สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบโทรสารผ่านเครือข่ายอินเตอร์เน็ต

INTERNET FACSIMILE SYSTEM

T061943

โดย

นางสาวชนิวิชญา รู้ทำนอง

นายชัชชัย ชมชูเวชช์

นายชัยมงคล ธรรมโหร

วท.
ว5154 ธ
2547

เลขหมู่.....
เลขทะเบียน.....61943
วัน,เดือน,ปี.....25 ก.ค. 2549

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ภาควิชา
วิศวกรรมโทรคมนาคม

11587003

b.11602301
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรสารผ่านเครือข่ายอินเตอร์เน็ต

INTERNET FACSIMILE SYSTEM

โดย

นางสาวชนิวิชญา รู้ทำนอง 44010089

นายชัชชัย ชมชูเวชช์ 44010097

นายชัยมงคล ธรรมโหร 44010105

อาจารย์ที่ปรึกษา

ดร.พิพัฒน์ พรหมมี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบโทรสารผ่านเครือข่ายอินเตอร์เน็ต**

The Internet Facsimile System

ผู้จัดทำ นางสาวชนิวิชญา รู้ทำนอง 44010089

นายชัชชัย ชมชูเวชช์ 44010097

นายชัยมงคล ธรรมโหร 44010105

..... อาจารย์ที่ปรึกษา

(ดร.พิพัฒน์ พรหมมี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรสารผ่านเครือข่ายอินเตอร์เน็ต

INTERNET FACSIMILE SYSTEM

โดย นางสาวชนิวิชญา รู้ทำนอง 44010089
นายชัชชัย ชมชูเวชช์ 44010097
นายชัยมงคล ธรรมโหร 44010105

อาจารย์ที่ปรึกษา ดร.พิพัฒน์ พรหมมี

บทคัดย่อ

ปริญญานิพนธ์นี้เสนอระบบการรับส่งโทรสารผ่านเครือข่ายอินเตอร์เน็ต โดยใช้หลักการของ Web-based E-mail และการลงทะเบียนบนอินเตอร์เน็ตโดยรับเอกสารเป็นแฟ้มเอกสารแนบ และส่งไปยังปลายทางที่เป็นเครื่องโทรสาร โดยมีปลายทางสำรองเป็นจดหมายอิเล็กทรอนิกส์หรือ SMS ผ่านโทรศัพท์มือถือ อีกทั้งระบบยังมีการแจ้งผลต่อผู้ส่งให้ทราบสถานะของการส่งโทรสารด้วย

ABSTRACT

The Internet Facsimile System is presented in this project. This project based on web-based E-mail, internet authentications and fax relay functions. The customer who is internet user have to registered to the E-mail server. The customer will sent the E-mail with document files as attachments. The system analyses the fax destination number then converted their attachments and sent them to the fax destination number. The alternated route are used in case of the first route is failure connected as well as E-mail or SMS. Moreover, System can also provide the feature of status updated to the customer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	**บทนำ**	**1**
	1.1 การกำหนดระยะทางคิดค่าบริการการเรียกทางไกล	1
	1.2 การเปรียบเทียบอัตราค่าบริการระหว่าง PSTN กับ Internet Fax System	2
บทที่ 2	**ทฤษฎีและหลักการ**	4
	2.1 โทรสาร (Facsimile (FAX))	4
	2.1.1 หลักการทำงานของโทรสาร	5
	2.1.2 การแบ่งกลุ่มของโทรสาร	5
	2.1.3 การเข้ารหัสข้อมูล	6
	2.1.4 ลำดับขั้นตอนในการรับส่งข้อมูลของโทรสาร	8
	2.2 โมเด็ม (MODEM)	10
	2.2.1 โมเด็มกับการสื่อสารข้อมูล	10
	2.2.2 โมเด็มสำหรับสายตรงและสายโทรศัพท์	11
	2.2.3 โมเด็มในยุคแรก	12
	2.2.4 องค์ประกอบในการสื่อสารข้อมูล	13
	2.2.5 การทำงานของเครือข่ายโทรศัพท์	14
	2.2.6 มาตรฐานของโมเด็มตาม CCITT (ITU-T) V-Series	18
	2.3 เครือข่ายคอมพิวเตอร์ (Computer Network)	20
	2.3.1 องค์ประกอบของเครือข่ายคอมพิวเตอร์ (Network Components)	20
	2.3.2 มาตรฐานอ้างอิงในการสื่อสารข้อมูล	24
	2.3.3 โปรโตคอล TCP/IP	29
	2.4 เว็บเซิร์ฟเวอร์	40
	2.5 เวิร์ดไวด์เว็บคืออะไร	41
	2.6 โฮมเพจ	42
	2.7 เอเอสพี	43
	2.8 การใช้งานโปรแกรม WINFAX PRO V.9	47
บทที่ 3	**การออกแบบและการสร้าง**	51
	3.1 ส่วนติดต่อกับผู้ใช้งาน	51
	3.2 ส่วนเซิร์ฟเวอร์หลัก	54
	3.3 ส่วนเซิร์ฟเวอร์ปลายทาง	56
บทที่ 4	**ผลการทดลอง**	59
	4.1 การทำงานของระบบ	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองใช้งานส่วน ติดต่อกับผู้ใช้งาน 59
4.3 การทดลองการใช้งานส่วนเซิร์ฟเวอร์กลาง 64
4.4 การทดลองการใช้งานส่วนเซิร์ฟเวอร์ปลายทาง 65
4.5 การทดลองวัดค่าประสิทธิภาพของระบบ 69
บทที่ 5 สรุปผลการทดลอง 77
5.1 สรุปผลการทดลอง 77
5.2 ปัญหาที่พบและวิธีแก้ไข 77
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 หลักการทำงานของโทรสารด้านส่ง 3

รูปที่ 2.2 หลักการทำงานของโทรสารด้านรับ 3

รูปที่2.3 ตัวอย่างการเข้ารหัสข้อมูลตามมาตรฐาน ITU-T(T.4) แบบ 1 มิติ 6

รูปที่ 2.4 แสดงลำดับในการติดต่อของโทรสาร 7

รูปที่ 2.5 โมเด็มแบบ Acoustic Couple 10

รูปที่ 2.6 ปลั๊กแบบ RJ-11 ใช้สำหรับต่อสายสัญญาณโทรศัพท์ 11

รูปที่ 2.7 แสดงการทำงานของระบบโทรศัพท์ 13

รูปที่ 2.8 แสดงเครือข่ายคอมพิวเตอร์ 18

รูปที่ 2.9 แสดงโดยผ่านการควบคุมเทอร์มินัลหรือตัวควบคุมระบบการสื่อสาร 19

รูปที่ 2.10 แสดงเครือข่ายย่อยส่วนของการสื่อสาร 20

รูปที่ 2.11 แสดงรูปแบบของเครือข่ายแบบจุดต่อจุด 21

รูปที่ 2.12 แสดงเครือข่ายแบบบัส 22

รูปที่ 2.13 แสดงสื่อสารข้อมูลจากระบบคอมพิวเตอร์หนึ่งไปยังอีกระบบหนึ่ง 23

รูปที่ 2.14 แสดงการแบ่ง OSI 7-Layer Model ในทางปฏิบัติ 23

รูปที่ 2.15 แสดงรูปแบบของข้อมูลแต่ละเลเยอร์ 26

รูปที่ 2.16 แสดงชั้นของโปรโตคอลแบบ TCP/IP 27

รูปที่ 2.17 แสดงการเชื่อมต่อเครือข่ายด้วยเราท์เตอร์ 29

รูปที่ 2.18 แสดงของข้อมูลของการส่งข้อมูลด้วย TCP/IP 30

รูปที่ 2.20 แสดงรูปแบบของ IP Datagram 30

รูปที่ 2.21 แสดงโปรโตคอลในชั้น Host-to-Host Layer และ Internetwork Layer 32

รูปที่ 2.22 แสดงโปรโตคอลของ TCP/IP Layer 32

รูปที่ 2.23 แสดงการส่งข้อมูลผ่านเราท์เตอร์ 34

รูปที่ 2.24 แสดงการแลกเปลี่ยนข้อมูลเกี่ยวกับเส้นทางการส่งผ่านข้อมูลระหว่าง Router 34

รูปที่ 2.25 แสดงการทำงานของ Link-State Routing Protocol 36

รูปที่2.26 แสดงการถ่ายโอนไฟล์ด้วยโปรโตคอล FTP 37

รูปที่2.27 พอร์ตสำหรับสัญญาณควบคุมและถ่ายโอนไฟล์ของโปรโตคอล FTP 38

รูปที่ 2.28แสดงภาพรวมของ wwwซึ่งเป็นบริการที่ได้รับความนิยมสูงสุดของอินเตอร์เน็ต 38

รูปที่ 2.29 แสดงการทำงานของ ASP 41

รูปที่ 2.30 แสดงองค์ประกอบในเอกสาร ASP ไฟล์หนึ่งๆ 43

รูปที่ 2.31 แสดงการติดตั้ง Winfax เป็นเครื่องพิมพ์เริ่มต้น(set as default) 45

รูปที่ 2.32 แสดงหน้าต่างโปรแกรม Winfax 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.33 แสดงหน้าต่างโปรแกรม Winfax ส่วนที่เป็นปกหน้า 47
รูปที่ 2.34 แสดงกล่องข้อความที่เป็นเอกสารแนบ 47
รูปที่ 2.35 แสดงหน้าต่างที่พร้อมจะส่งโทรสาร 48
รูปที่ 3.1 แสดงผังการทำงานของส่วนติดต่อกับผู้ใช้งาน 51
รูปที่ 3.2 แสดงผังการทำงานส่วนที่ทำหน้าที่ในการติดต่อกับเซิร์ฟเวอร์ปลายทาง 55
รูปที่ 3.3 ตัวอย่างการเข้ารหัสข้อมูลตามมาตรฐาน ITU-T (T.4)แบบ 1 มิติ 57
รูปที่ 3.3 ผังการทำงานส่วนเซิร์ฟเวอร์ปลายทาง 58
รูปที่ 4.1 แสดงหน้าLogin เพื่อให้ผู้ใช้กรอก Member name และ Password 60
รูปที่ 4.2 แสดงหน้าต่างลงทะเบียนสมาชิกใหม่ 60
รูปที่ 4.3 แสดงหน้าต่างเมื่อระบบได้รับข้อมูลการลงทะเบียนจากผู้ใช้งาน 61
รูปที่ 4.4 แสดงหน้าต่างสำหรับกรอกรายละเอียดต่าง ๆ ในการส่งโทรสาร 61
รูปที่ 4.5 แสดงหน้าต่างที่ระบบจะแสดงข้อความว่าได้ส่งข้อมูลต่าง ๆ เข้าสู่ระบบเรียบร้อยแล้ว 62
รูปที่ 4.6 แสดงหน้าผู้ดูแลระบบสามารถเพิ่ม credit ให้ผู้ใช้ที่ร้องขอ 62
รูปที่ 4.7 แสดงรายชื่อและข้อมูลของสมาชิกทั้งหมด 63
รูปที่ 4.8 แสดงหน้าของข้อมูลที่เคยส่งทั้งหมด 63
รูปที่ 4.9 แสดงหน้าต่างของเซิร์ฟเวอร์หลักขณะทำการส่งข้อมูลไปยังเซิร์ฟเวอร์ปลายทาง 64
รูปที่ 4.10 แสดงหน้าต่างของเซิร์ฟเวอร์หลักขณะทำการส่งข้อมูลไปยังเซิร์ฟเวอร์ปลายทาง 65
รูปที่ 4.11 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทาง 65
รูปที่ 4.12 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทางขณะรอการติดต่อจากเซิร์ฟเวอร์หลัก 66
รูปที่ 4.13 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทางขณะสร้างการเชื่อมต่อกับเซิร์ฟเวอร์หลัก 66
รูปที่ 4.14 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทางขณะรับข้อมูลจากเซิร์ฟเวอร์หลัก 67
รูปที่ 4.15 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทางขณะรอรับข้อมูลชุดต่อไปจากเซิร์ฟเวอร์หลัก 67
รูปที่ 4.16 แสดง SMS ที่ผู้รับโทรสารได้รับ 68
รูปที่ 4.17 แสดง SMS ที่ผู้ส่งโทรสารได้รับ 68
รูปที่ 4.18 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 1 หน้า 69
รูปที่ 4.19 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 2 หน้า 69
รูปที่ 4.20 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 3 หน้า 70
รูปที่ 4.21 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 4 หน้า 70
รูปที่ 4.22 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 5 หน้า 70
รูปที่ 4.23 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 6 หน้า 71
รูปที่ 4.24 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 7 หน้า 71
รูปที่ 4.25 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 8 หน้า 71
รูปที่ 4.26 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 9 หน้า 72
รูปที่ 4.27 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 10 หน้า 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.28 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 11 หน้า 72
รูปที่ 4.29 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 12 หน้า 73
รูปที่ 4.30 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 13 หน้า 73
รูปที่ 4.31 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 14 หน้า 73
รูปที่ 4.32 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 15 หน้า 74
รูปที่ 4.33 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 16 หน้า 74
รูปที่ 4.34 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 17 หน้า 74
รูปที่ 4.35 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 18 หน้า 75
รูปที่ 4.36 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 19 หน้า 75
รูปที่ 4.37 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 20 หน้า 75
รูปที่ 4.38 กราฟแสดงเวลาเฉลี่ยที่ใช้ในการส่งข้อมูล 76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันเทคโนโลยีทางด้านการสื่อสารมีการพัฒนาไปอย่างรวดเร็วมาก เครือข่ายอินเตอร์เน็ต (Internet) เป็นเทคโนโลยีที่มีการเจริญเติบโตอย่างรวดเร็ว ซึ่งจะเห็นได้จาก การเกิดขึ้นใหม่ของเว็บไซต์ (Website) มากมาย การตื่นตัวของการใช้งานพาณิชย์อิเล็กทรอนิกส์ (e-commerce) รวมทั้งการเพิ่มขึ้นของผู้ใช้ระบบเครือข่ายอินเตอร์เน็ต ทั้งนี้เนื่องจากเครือข่ายอินเทอร์เน็ต เป็นแหล่งข้อมูลขนาดใหญ่ และเป็นเครือข่ายที่มีการเชื่อมต่อกันทั่วโลก ทำให้เป็นช่องทางในการติดต่อสื่อสารกันได้ ซึ่งรูปแบบที่ใช้ในการสื่อสารสามารถใช้ได้หลายรูปแบบ ไม่ว่าจะเป็นข้อมูลอักษร ภาพ เสียง ฯลฯ ซึ่งสามารถติดต่อสื่อสารถึงกันได้อย่างรวดเร็ว

อย่างไรก็ดี การใช้งานระบบเครือข่ายอินเตอร์เน็ตภายในองค์การธุรกิจขนาดเล็กและขนาดย่อม มีข้อจำกัดต่างๆ ในหลายๆด้าน เช่นบุคลากรขาดความรู้ความสามารถในการใช้งานระบบเครือข่ายอินเตอร์เน็ตและคอมพิวเตอร์ หรือการที่ไม่สามารถเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้ตลอดเวลา

โทรสารเป็นอุปกรณ์สื่อสารที่ใช้เชื่อมต่อกับเครือข่ายโทรศัพท์ซึ่งมีใช้กันอย่างแพร่หลายในองค์กรธุรกิจต่างๆ ซึ่งมีข้อดีก็คือข่าวสารที่มาถึงสามารถพิมพ์ออกมาเป็นเอกสารได้ทันที บุคลากรส่วนใหญ่มีความชำนาญในการใช้งานอยู่แล้วและสามารถเปิดใช้งานได้ตลอดเวลา จากข้อดีดังกล่าวของโทรสารซึ่งสามารถกำจัดของการใช้งานระบบเครือข่ายอินเตอร์เน็ตภายในองค์การธุรกิจขนาดเล็กและขนาดย่อมได้

โครงงานนี้เสนอโปรแกรมที่ใช้ในการส่งโทรสารผ่านเครือข่ายอินเตอร์เน็ต ไปยังปลายทางที่เป็นเครื่องโทรสารเพื่อตอบสนองความต้องการดังได้กล่าวแล้วในข้างต้น

1.1 การกำหนดระยะทางคิดค่าบริการการเรียกทางไกล (distance call pricing plan)

เพื่อให้ผู้เช่าสามารถโทรศัพท์ทางไกลจ่ายค่าบริการเหมาะสมกับระยะทาง คือ ผู้ใช้ระยะใกล้จ่ายถูกกว่าผู้ใช้ระยะทางไกลกว่าในหน่วยเวลา เช่น นาที หรือ หน่วย ทราฟฟิก ในการคิดค่าบริการที่เป็นธรรม

การกำหนดระยะทางจะต้องลงตัวกับการกำหนดค่า K และการแปลรหัสปลายทางในประเทศ (รหัสพื้นที่ + รหัสชุมสาย) ตัวแรกบอกทิศทางสู่ภูมิภาคปลายทาง ตัวที่สองบอกระยะทางโดยสังเขป เช่น

รหัสปลายทางในประเทศ	ทิศทางส่งภูมิภาคปลายทาง
5X	ทิศทางส่งภูมิภาคปลายทางภาคเหนือ
“ 3	ทิศทางส่งภูมิภาคปลายทางภาคเหนือ ตอนบน
“ 6	ทิศทางส่งภูมิภาคปลายทาง ภาคเหนือ ตอนล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทศท. กำหนดระยะทางในโครงข่าย 4 ระดับ เป็น 6 ระยะ แต่โครงข่ายโทรศัพท์ทางไกลประหยัด 1234 ลดเหลือ 4 ระยะ และโทรศัพท์เคลื่อนที่เหลือ 3 ระยะดังแสดงในตารางที่ 1.1

ตารางที่ 1.1 การกำหนดระยะทางของ ทศท.

โครงข่ายPSTN		Y-tel 1234		โทรศัพท์เคลื่อนที่	
6 ระยะทาง	บาท/วินาที	4 ระยะทาง	บาท/วินาที	เขต โทรศัพท์เคลื่อนที่	บาท/นาที
0-50	3	0-50	2	เดียวกัน	3
51-100	6	51-100	4	ติดกัน	8
101-200	9	101-200	6	ไม่ติดกัน	12
201 ขึ้นไป	12	201 ขึ้นไป	8		

* ปรับลดเหลือ 12 บาท / นาที ตั้งแต่ 1 พ. ค . 2543

จากตาราง จะเห็นได้ว่าระยะทางเพิ่มขึ้น อัตราค่าบริการจะเพิ่มขึ้นตามระยะทาง ซึ่งแตกต่างจากInternet Fax System ซึ่งคิดอัตราค่าบริการในทุกพื้นที่ในอัตราค่าบริการพื้นที่เดียวกัน

1.2 **การเปรียบเทียบอัตราค่าบริการระหว่าง PSTN กับ Internet Fax System**

ระยะเวลา / ระยะทาง (กม.)	PSTN บาท / นาที	Internet Fax System บาท/ครั้ง
0 – 50	3	3
51 – 100	6	3
101 – 200	9	3
201 ขึ้นไป	12	3

ตารางที่ 1.2 เปรียบเทียบอัตราค่าบริการระหว่าง PSTN กับ Internet Fax System

หมายเหตุ: จากตารางอัตราค่าบริการของ PSTN อยู่ในช่วงเวลา 7.00น.-18.00น.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างเช่น บริษัทแห่งอยู่ที่กรุงเทพ ต้องการส่งโทรสารไปเชียงใหม่ 100แผ่นต่อเดือน โดยสมมุติว่าส่ง1แผ่นต่อ1ครั้ง ถ้าเป็นระบบ PSTN จะเสียค่าใช้จ่าย 300X12 = 3600 บาท แต่ถ้าเป็นInternet Fax System จะเสียค่าใช้จ่าย 300X3 = 900 บาท ทำให้ประหยัดได้ 2700 บาทต่อเดือน

จากตัวอย่างข้างต้น

ถ้าเราเป็นผู้ให้บริการคิดในส่วนของต้นทุน 900 บาท บวกกับค่าอุปกรณ์ประมาณ 900 บาทคิดเป็นต้นทุน 1800 บาท ประหยัดได้ถึง 1800 บาท ถ้าคืนกำไรแก่ผู้ใช้บริการ 500 บาท เรายังประหยัดค่าใช้บริการได้อีกถึง 1300 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 โทรสาร (Facsimile (FAX))

โทรสารเป็นเครื่องมือสื่อสารที่มีความสามารถในการส่งภาพจากต้นทางไปยังปลายทางถูกคิดค้นเมื่อปี ค.ศ. 1843 ก่อนที่จะมีการคิดค้นโทรพิมพ์ (Typewriter) 60 ปี โทรสารมีการพัฒนามาอย่างต่อเนื่องแต่ก็มักจะประสบปัญหาด้านโครงสร้างภายในของระบบการสื่อสารสมัยก่อน ในช่วงปี ค.ศ. 1920 โทรสารมีความสำคัญอย่างยิ่งในด้านงานข่าว โดยสำนักข่าวส่วนใหญ่นำมาใช้ในการส่งภาพข่าว

การใช้งานโทรสารใช้การส่งและรับเอกสาร ผ่านโครงข่ายของการสื่อสาร โดยใช้เทคนิคของการส่งข้อมูลตัวอักษรในแบบเดียวกัน และมีความเหมาะสมกับการส่งเอกสารที่ประกอบด้วยตัวหนังสืออย่างเดียว ส่วนเทคนิคอื่น ๆ ก็คือ การส่งภาพนิ่งของต้นฉบับในแบบอิเล็กทรอนิกส์ โดยจะมีแผนผังเป็นส่วนในการสร้างภาพนิ่งนี้ เพื่อใช้ในการสื่อสารผ่านโครงข่าย การบริการของโทรสารนี้จะรวมไปถึงตัวอักษร ภาพกราฟิก (Graphic) หรือรูปภาพ

ระบบโทรสารจะมีข้อดีเหนือกว่าการส่งอื่น ๆ ในรูปแบบของเอกสารหรือการส่งข้อความ การใช้งานของโทรสารจะช่วยประหยัดเวลาในการเตรียมเอกสารอีกด้วย เนื่องจากการส่งข่าวสารไม่จำเป็นต้องมาผ่านการส่งโดยกดจากแป้นพิมพ์ แต่จะใช้ตัวต้นฉบับของเอกสารนั้นส่งไปยังปลายทางได้เลย โดยต้นฉบับที่เป็นเอกสารนี้จะถูกสแกน (Scan) หาความสว่างและความมืดของเอกสาร แล้วจะเป็นแปลงสัญญาณอิเล็กทรอนิกส์ เพื่อส่งไปยังปลายทาง ระบบโทรสารได้ถูกพัฒนาขึ้นในปี ค.ศ. 1843 แต่เพิ่งถูกนำมาใช้งานในตอนกลางปี ค.ศ. 1970 และได้มีการเจริญอย่างรวดเร็ว

ได้มีการพัฒนาและการยอมรับมาตรฐาน CCITT เกี่ยวกับโทรสาร ทำให้โทรสารสามารถทำงานร่วมกันระหว่างประเทศได้ เทคโนโลยีของโทรสารที่มีความก้าวหน้าขึ้น มีความเร็วสูงขึ้น คุณภาพดีกว่าเดิม ขนาดของเครื่องเล็กลง ค่าใช้จ่ายของเครื่องและการส่งโทรสารลดลง และมีการใช้เครื่องที่ง่ายขึ้น

การคิดค้นระบบโทรสารแบบดิจิทัล ทำให้แต่ละหน้าของข่าวสารสามารถดำเนินการได้อย่างง่ายดาย โดยใช้คอมพิวเตอร์ทำให้สามารถเก็บข้อมูลข่าวสารลงในดิสก์ (Disk) หรือเทปแม่เหล็ก

โทรสารยังคงมีการพัฒนาเรื่อยมาจนปัจจุบัน การพัฒนาของระบบโทรสารแบบดิจิทัล ทำให้บทบาทของโทรสารได้ขยายวงกว้างออกไปในรูปแบบของดิจิทัล ในแง่ของข่าวสารที่สามารถส่งระหว่างเครื่องโทรสารด้วยกันได้โดยตรง โดยไม่ต้องผ่านขั้นตอนบนกระดาษก่อน เอกสารที่ต้องการส่งสามารถถูกเตรียมโดยการใช้โปรแกรมเขียนแบบเวิร์ดโปรเซสเซอร์ (Word Processor) ความสามารถในการส่งตัวหนังสือและกราฟิกร่วมกันได้ ดังนั้นผู้ใช้สามารถแก้ไขตัวหนังสือได้ รวมทั้งรูปภาพกราฟิกต่าง ๆ ก่อนที่จะมีการส่งโทรสารออกไปยังปลายทาง และเนื่องจากโทรสารสามารถนำมาต่อเข้ากับเครือข่ายโทรศัพท์ได้ทันที มีการใช้งานได้สะดวกรวดเร็วทำให้โทรสารเป็นอุปกรณ์สื่อสารที่มีการใช้กันอย่างแพร่หลายในสำนักงานและมีราคาที่ถูกลงอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 หลักการทำงานของโทรสาร

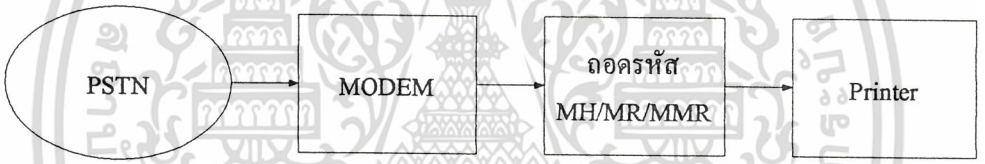
โดยทั่วไปโทรสารจะมีการทำงานโดยใช้ CCD (Charge Coupled Device) ในการสแกนภาพ โดยจำทำการเปลี่ยนภาพเป็นไฟฟ้า จากการตรวจสอบความเข้มของพิกเซล (Pixel) แต่ละพิกเซลทีละพิกเซล โดยในการสแกนใน 1 เส้นจะมีจำนวน 1728 พิกเซล (ตามมาตรฐาน CCITT T.30)

หลังจากได้ข้อมูลจากการสแกนโดย CCD แล้วจะนำไปผ่านอุปกรณ์แปลงสัญญาณแอนะล็อกเป็นดิจิทัล เพื่อนำเอาสัญญาณดิจิทัลที่ได้มาเข้ารหัสซึ่งสามารถทำได้ 3 รูปแบบ คือ

HM (Modified Huffman) , MR (Modified Reading) , MMR (Modified Modified Reading) ซึ่งจะช่วยในการลดขนาดของข้อมูลและช่วยให้สามารถส่งข้อมูลได้เร็วขึ้น

หลังจากนั้นจะส่งผ่านโมเด็มซึ่งจะแปลงสัญญาณกลับไปเป็นแอนะล็อกอีกครั้ง เพื่อส่งไปปลายทางผ่านเครือข่ายโทรศัพท์

รูปที่ 2.1 หลักการทำงานของโทรสารด้านส่ง



รูปที่ 2.2 หลักการทำงานของโทรสารด้านรับ

ทางด้านรับจะนำข้อมูลที่ได้ผ่านโมเด็มเพื่อทำการแปลงสัญญาณกลับมาเป็นสัญญาณดิจัล เพื่อนำไปถอดรหัสให้อยู่ในรูปพิกเซลขาวและดำ เพื่อให้อุปกรณ์การพิมพ์แสดงผลออกมา

2.1.2 การแบ่งกลุ่มของโทรสาร

ITU-T ได้แบ่งอุปกรณ์โทรสารสำหรับใช้บนโครงข่ายสาธารณะออกเป็น 4 แบบ (หรือกลุ่ม)

2.1.2.1) กลุ่มที่ 1 (Group 1 Facsimile) จะเป็นโทรสารแบบแอนะล็อก ที่มีความเร็วในการส่งต่ำ การส่งแบบ FM (Frequency Modulation) ในการแบ่งแยกระดับของสีเทาหลาย ๆ ระดับ ที่อยู่ระหว่างกลางของสีขาวกับสีดำ ซึ่งสามารถส่งเอกสารขนาด A4 ISO (210 x 297 มิลลิเมตร หรือ 8.25 x 11.7 นิ้ว) ที่ประมาณ 4 เส้น ต่อ มิลลิเมตร (100 เส้นต่อนิ้ว) ในเวลาประมาณ 6 นาที และมีข้อกำหนดต่าง ๆ ตามข้อกำหนด T.2 ของ CCITT

2.1.2.2) กลุ่มที่ 2 (Group 2 Facsimile) เป็นโทรสารแบบแอนะล็อก ที่มีการปรับปรุงขึ้นมาโดยใช้เทคนิคในการลดแบนด์วิดท์ลงเพื่อให้ได้ความเร็วที่เพิ่มขึ้นเป็น 2 เท่า หรือมากกว่าที่รายละเอียดของภาพเท่ากันกับในกลุ่มที่ 1 การส่งจะเป็นแบบ duo binary PM (Phase Modulation) และจะยังคงมีระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสีเทาระดับต่าง ๆ ในระหว่างที่สีขาวกับสีดำ ต้องสามารถทำการส่งเอกสาร A4 ได้ภายในเวลา 3 นาที และมีข้อกำหนดต่าง ๆ ตามข้อกำหนด T.3 ของ CCITT

2.1.2.3) กลุ่มที่ 3 (Group 3 Facsimile) เป็นโทรสารระบบแรกที่ใช้ระบบดิจิทัล ระบบนี้จะจัดให้มีค่าเพียงขาวกับดำเท่านั้น โดยจะมีการสุ่มสัญญาณที่ความหนาแน่น 200 พิกเซลต่อนิ้ว ทางแนวนอนขวางของกระดาษ โดยจะมีสแกน 100 หรือ 200 เส้นต่อหนึ่งนิ้ว ทางแนวตั้งของกระดาษ โทรสารกลุ่มที่ 3 จะใช้การเข้ารหัสดิจิทัลและมีการลดค่าเฉลี่ยของข่าวสารที่ซ้ำซ้อนในสัญญาณที่ได้จากการเข้ารหัสของข่าวสารบนเอกสารก่อนที่จะทำการมอดดูเลชั่น และถ้ามีการใช้โมเด็มในการส่งสัญญาณผ่านโครงข่ายโทรศัพท์แอนะล็อก เวลาในการส่งจะเร็วขึ้นเป็น 3 เท่า หรือมากกว่าเมื่อเปรียบเทียบกับโทรสารในกลุ่มที่ 2 ต้องสามารถทำการส่งเอกสาร A4 ได้ภายในเวลา 1 นาที และข้อกำหนดต่าง ๆ ตามข้อกำหนด T.4 ของ CCITT

2.1.2.4) กลุ่มที่ 4 (Group 4 Facsimile) เป็นโทรสารระบบดิจิทัล ดำ / ขาว เช่นเดียวกับกลุ่มที่ 3 แต่จะถูกเน้นสำหรับใช้บนโครงข่ายดิจิทัลที่มีความเร็วสูงขึ้นไปถึง 64 Kbps. และมีการจัดการเกี่ยวกับตรวจสอบความผิดพลาดของข้อมูลที่ได้รับ ความละเอียดสามารถใช้ได้จาก 200 ถึง 400 พิกเซล ต่อนิ้ว การส่งโทรสารกลุ่มที่ 4 จะใช้เวลาในการส่งลดลงเหลือเพียงไม่กี่วินาที มาตรฐานโทรสารกลุ่ม 4 เป็นการทำงานแบบดิจิทัล ซึ่งใช้งานร่วมกับระบบ ISDN

โทรสารในกลุ่มที่ 1 และ 2 ได้มีการกำหนดมาตรฐานในปี 1976 ส่วนกลุ่มที่ 3 ทำขึ้นในปี 1980 ส่วนในกลุ่มที่ 4 ได้ตั้งขึ้นในปี 1984

ปัจจุบันมีการใช้งานโทรสารทั่วไป จะใช้ตามมาตรฐานโทรสารกลุ่ม 3 โดยสามารถใช้การเข้ารหัสข้อมูลแบบดิจิทัล (มาตรฐานกลุ่ม 1 และ 2 มีระบบการทำงานแบบแอนะล็อก ส่วนมาตรฐานกลุ่ม 3 และ 4 มีระบบการทำงานแบบดิจิทัล) โดยมาตรฐานโทรสารกลุ่ม 3 จะมีความละเอียดในแนวตั้ง 3.85 เส้นต่อมิลลิเมตร และในแนวนอนจะมีความละเอียดในแนวนอน 1728 พิกเซลต่อความยาว 215 มิลลิเมตร เท่ากับ 8 พิกเซลต่อมิลลิเมตร อย่างไรก็ดีในมาตรฐาน T.4 ให้เพิ่มความละเอียดได้สูงถึง 15.4 เส้นต่อมิลลิเมตรในแนวตั้งและ 16 พิกเซลต่อมิลลิเมตรใน

2.1.3 การเข้ารหัสข้อมูล

ในการติดต่อสื่อสารด้วยโทรสาร รูปแบบของข้อมูลจะต้องใช้ตามมาตรฐาน CCITT(T.4) ดังนั้นเราจึงต้องทำการเปลี่ยนรูปแบบของข้อมูลที่จะทำการส่งโทรสารให้อยู่ในรูปแบบของข้อมูลที่ตรงตามมาตรฐานก่อนที่จะทำการส่งข้อมูลผ่านโมเด็ม

ในเครื่องโทรสารกลุ่ม 3 กำหนดให้มีการเข้ารหัสเพื่อให้ข้อมูลที่ต้องการส่งผ่านระบบสื่อสัญญาณมีจำนวนน้อยลง ช่วยให้การส่งเร็วขึ้น การเข้ารหัสมีให้เลือก 2 แบบ คือ 1 มิติ และ 2 มิติ แต่จะแสดงรายละเอียดการเข้ารหัสแบบ 1 มิติ ซึ่งใช้หลักการเข้ารหัสแบบโมดิฟายด์ฮัฟฟ์แมน (Modified Huffman) ซึ่งข้อมูลที่มีค่าความน่าจะเป็นสูง (เกิดขึ้นบ่อย) จะถูกแทนด้วยรหัสที่ความยาวน้อย ส่วนข้อมูลที่มีความน่าจะเป็นต่ำ (เกิดขึ้นน้อย) จะถูกแทนด้วยรหัสที่ความยาวมาก ซึ่งในมาตรฐาน CCITT T.4 ได้กำหนดรหัสที่ใช้ในการแทนมาอย่างแน่นอนแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดในการเข้ารหัสมีดังต่อไปนี้

1. การเข้ารหัสจะเริ่มต้นด้วยพิกเซลขาวก่อนเสมอ กรณีที่พิกเซลแรกเป็นดำ รหัสของพิกเซลขาวที่มีความยาวเท่ากับ 0 จะถูกส่งออกไปก่อน แล้วจึงตามด้วยพิกเซลดำ

2. ข้อมูลพิเซลขาวหรือดำสามารถมีความยาวได้ถึง 1728 พิกเซล ซึ่งเป็นความยาวสูงสุดสำหรับเส้นสแกนมาตรฐาน 1 เส้น

3. รหัสที่ใช้แทนข้อมูล มี 2 ชนิด คือ

- Terminating Code ใช้แทนพิกเซลที่มีความยาวตั้งแต่ 0 ถึง 63
- Makeup Code ใช้แทนพิกเซลที่มีความยาวเป็นจำนวนเท่าของ 64 จนถึง 1728

4. กรณีของพิกเซลที่มีความยาวตั้งแต่ 64 – 1728 จะแทนด้วย Makeup Code ซึ่งมีค่าเท่ากับหรือน้อยกว่าความยาวของพิกเซลก่อน โดยความยาวของพิกเซลที่เหลือจากการแทนด้วย Makeup Code จะมีค่าไม่เกิน 63 พิกเซล ซึ่งสามารถแทนค่าที่เหลือด้วย Terminating Code ได้

5. จุดสิ้นสุดของเส้นสแกน (EOL: End- Of – Line) เมื่อสิ้นสุดการสแกนข้อมูลครบ 1 เส้น ข้อมูลจะต้องตามด้วยรหัส EOL เพื่อแสดงจุดสิ้นสุดของเส้นสแกน นอกจากนี้รหัส EOL ยังใช้นำหน้าเส้นสแกนเส้นแรกของแต่ละหน้าด้วย

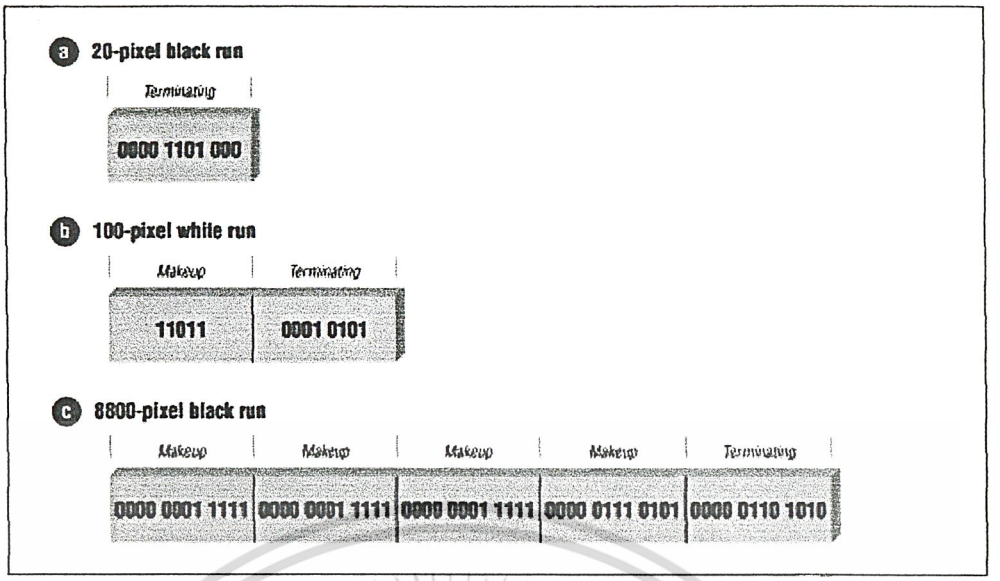
6. บิตเติม (Fill) เป็นบิตที่เติมขึ้นระหว่างบิตของข้อมูลกับบิตสิ้นสุด เพื่อให้เวลาในการส่งข้อมูลในเส้นสแกนนั้นไม่ต่ำกว่าเวลาต่ำสุดที่เครื่องพิมพ์ด้านรับสามารถทำงานได้ทัน

7. เมื่อจบการส่งข้อมูลแต่ละแผ่นจะต้องตามด้วยรหัส RTC (Return to Control) เพื่อเป็นการกลับเข้าสู่โหมดการควบคุมเครื่องโทรสาร

ตัวอย่างของการเข้ารหัสสามารถอธิบายได้ดังนี้

- สมมติมีข้อมูลภาพสีดำจำนวน 20 พิกเซล จะต้องแสดงอยู่ในรูปของข้อมูล Terminating Code สำหรับข้อมูลสีดำ ความยาว 20 พิกเซล ดังรูป 2.3 a
- สมมติมีข้อมูลภาพสีขาวจำนวน 100 พิกเซล จะต้องเข้ารหัสข้อมูลโดยการใช้ Makeup Code สำหรับข้อมูลสีขาว 64 พิกเซล ตามด้วย Terminating Code 36 พิกเซล (64 + 36 = 100) ดังรูป 2.3b
- สมมติมีข้อมูลภาพสีดำจำนวน 8800 พิกเซล จะต้องเข้ารหัสข้อมูลโดยการใช้ Makeup Code สำหรับข้อมูลสีดำ 2560 พิกเซล 3 ครั้ง จะได้ 7680 พิกเซล และใช้ Makeup Code สำหรับข้อมูลสีดำ 1088 พิกเซล ตามด้วย Terminating Code 32 พิกเซล (2560+2560+2560+1088+32 = 8800) ดังรูป 2.3c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.3 ตัวอย่างการเข้ารหัสข้อมูลตามมาตรฐาน CCITT(T.4) มีลักษณะการเข้ารหัสแบบ 1 มิติ

2.1.4 ลำดับขั้นตอนในการรับส่งข้อมูลของโทรสาร

ตามมาตรฐาน CCITT T.30 ได้จัดลำดับขั้นตอนการติดต่อในการรับส่งข้อมูลของโทรสารเป็น 5 เฟส (Phase) ประกอบด้วย

2.1.4.1 เฟส A : การเริ่มติดต่อ (Phase A : Call establish) การติดต่อถึงกันระหว่างกันของเครื่องโทรสารซึ่งจะเป็นแบบอัตโนมัติหรือไม่ก็ได้ และจะเสร็จสิ้นเมื่อโทรสารทั้ง 2 ฝ่ายสามารถติดต่อกันได้

เฟส A นี้อาจเรียกว่า ช่วงเรียก หลังจากฝ่ายรับยกหูขึ้น ฝ่ายเรียกจะส่งสัญญาณ CNG Beep (Calling Tone) ซึ่งมีความถี่ 1 kHz ส่วนฝ่ายรับจะตอบสนองโดยการส่งสัญญาณ CED (Calied Station Identification) ซึ่งมีความถี่ 2100 kHz

2.1.4.2 เฟส B : ขั้นตอนก่อนส่งข้อมูล (Phase B : Pre-Message Procedure) หลังจากทำการติดต่อกันได้แล้วจะทำการตรวจสอบคุณสมบัติต่าง ๆ ในการรับส่งข้อมูลให้ตรงกันโดยมี 2 ส่วน

<u>ส่วนการแสดงสถานะ</u> ทำการแจ้งสถานะและตรวจสอบสถานะต่าง ๆ ได้แก่

-มาตรฐานกลุ่มของโทรสาร

-สถานะความพร้อมในการรับข้อมูล

นอกจากนี้ยังมีสถานะอื่น ๆ ที่อยู่นอกเหนือมาตรฐานซึ่งสามารถใช้งานได้เพิ่มเติม เช่น หมายเลขต้นทาง ชื่อเครื่องโทรสารต้นทาง

<u>ส่วนการส่งคำสั่ง</u> เพื่อทำการตรวจสอบ ระบบจะส่ง DIS (Digital Identification Signal) เพื่อแสดงคุณสมบัติที่จะใช้งาน เมื่อปลายทางได้รับสัญญาณจะส่ง DCS (Digital Command Signal) กลับมาเพื่อแจ้งให้ทราบว่าจะสามารถใช้คุณสมบัติชนิดใดบ้าง เพื่อกำหนดในการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งนี้ โทรสารด้านผู้ถูกเรียกจะส่งสัญญาณควบคุมความเร็วสูงที่เรียกว่า เทรนนิ่ง (Training) เพื่อตรวจสอบระบบแล้วส่ง CFR (Confirmation to Request) กลับมาเพื่อยืนยันความพร้อมในการรับข้อมูล

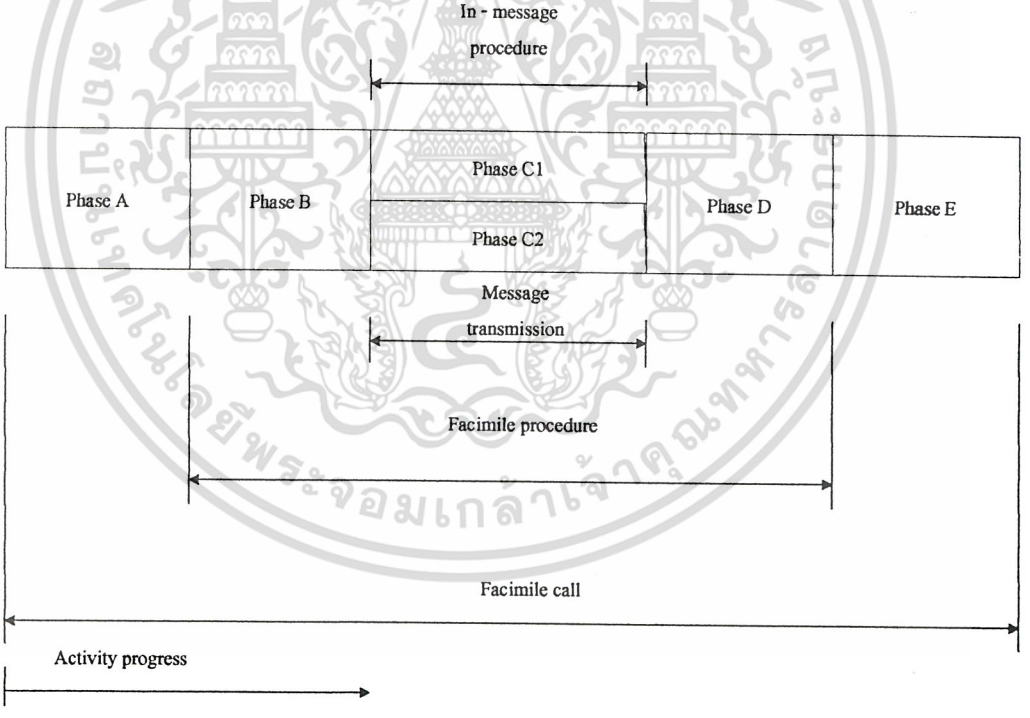
2.1.4.3 เฟส C : การส่งข้อมูล (Phase C : Message Transmission) แบ่งออกเป็น 2 ส่วน

2.2.4.3.1) เฟส C1 (Phase C1 : In-massage Procedure) เป็นการส่งสัญญาณควบคุมขณะที่ทำการส่งข้อมูล

2.2.4.3.2) เฟส C2 (Phase C2 : Message Transmission) ขั้นตอนในการส่งข้อมูลจริง โดยมีรูปแบบตามที่กำหนดในมาตรฐานที่ใช้แต่ละกลุ่ม

2.1.4.4 เฟส D : หลังการส่งข้อมูล (Phase D: Post – Message Procedure) หลังจากส่งข้อมูลเรียบร้อย ด้านผู้เรียกจะทำการส่ง RTC (Return to Control) เพื่อส่ง EOP (End of Procedure) เพื่อแสดงว่าจบการทำงาน

2.1.4.5 เฟส E : การเลิกการติดต่อ (Phase E : Call Release) เป็นการยกเลิกการติดต่อระหว่างโทรสาร ซึ่งสามารถทำได้โดยวิธีอัตโนมัติหรือไม่ก็ได้ โดยจะมีการส่ง DCN (Disconnect) ไปยังปลายทาง เพื่อยกเลิกการติดต่อ



รูปที่ 2.4 แสดงลำดับในการติดต่อของโทรสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โมเด็ม (MODEM)

โมเด็ม (MODEM) เป็นคำที่ย่อมาจากคำว่า Modulate – DEModulate ซึ่งเป็นหน้าที่การทำงานของโมเด็ม คือทำการมอดดูเลชั่น (Modulation) และ ดีมอดดูเลชั่น (Demodulation) สัญญาณ โดยโมเด็มจะทำหน้าที่แปลงสัญญาณดิจิทัล (Digital) จากคอมพิวเตอร์ที่ส่งมาทาง RS –232C ให้กลายเป็นสัญญาณแอนะล็อก (Analog) แล้วส่งออกไปตามสายส่ง ขบวนการนี้เราเรียกว่า การมอดดูเลต (Modulate) สัญญาณ ในทางกลับกัน เมื่อโมเด็มได้รับสัญญาณเข้ามาก็จะแปลงสัญญาณแอนะล็อกที่ได้รับกลับมาเป็นสัญญาณดิจิทัล แล้วจึงส่งให้คอมพิวเตอร์ในรูปของสัญญาณดิจิทัลผ่านทาง RS – 232C เช่นกัน ขบวนการแปลงสัญญาณกลับนี้ เรียกว่า ดีมอดดูเลต (Demodulate)

2.2.1 โมเด็มกับการสื่อสารข้อมูล

ข้อมูลที่ใช้อยู่ในระบบคอมพิวเตอร์ เป็นข้อมูลแบบดิจิทัล มีลักษณะเป็นสัญญาณแบบไม่ต่อเนื่อง คือเป็นสัญญาณของข้อมูล “0” และ “1” ส่วนสัญญาณไฟฟ้าอีกรูปแบบหนึ่งนั้น เป็นสัญญาณที่มีความต่อเนื่อง เรียกว่าสัญญาณแบบแอนะล็อก

ในการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ ในระยะทางที่ไม่ไกลมากนัก อาจใช้การรับส่งแบบอนุกรมแบบตามมาตรฐาน RS – 232C ซึ่งส่งข้อมูลดิจิทัลของคอมพิวเตอร์ไปตามสายจนถึงผู้รับ ในกรณีนี้จะสามารถรับส่งข้อมูลได้ไกลถึง 35 เมตร ตามคุณสมบัติของ RS – 232C ซึ่งขึ้นอยู่กับคุณภาพของสายส่งด้วย ถ้าสายส่งมีคุณภาพดีอาจส่งได้ไกลถึง 150 เมตร ที่ความเร็ว 9,600 บิต (Bit) ต่อวินาที สำหรับการส่งข้อมูลระยะไกลส่งข้อมูลแบบดิจิทัล โดยตรงอาจเกิดปัญหาหลายอย่าง ปัญหาที่สำคัญก็คือ สัญญาณรูปสี่เหลี่ยมของสัญญาณดิจิทัลเมื่อส่งไปไกล ๆ สัญญาณจะเพี้ยนหรือมีรูปร่างผิดไปจากเดิมได้ง่าย ทำให้สายส่งและวงจรรับส่งสัญญาณดิจิทัลต้องถูกออกแบบมาเป็นอย่างดี ราคาของสายส่งสัญญาณแบบดิจิทัลจึงมีราคาแพงกว่าสายส่งสัญญาณแบบแอนะล็อกมาก ในทางปฏิบัติเราอาจรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองเครื่องโดยใช้สัญญาณดิจิทัลผ่านสายส่งได้ ซึ่งทั้งสายส่งและวงจรเชื่อมต่อทั้งหมดเป็นแบบดิจิทัล แต่ค่าใช้จ่ายของระบบจะสูงมาก จึงไม่คุ้มกับการลงทุน วิธีที่จะสามารถหลีกเลี่ยงคือการส่งข้อมูลไปตามสายส่งในแบบแอนะล็อกแทน การทำเช่นนี้เราจำเป็นต้องใช้อุปกรณ์เข้าช่วยแปลงสัญญาณในการรับส่งข้อมูลทั้งสองด้าน ซึ่งเป็นที่มาของโมเด็มนั่นเอง

จากการที่โมเด็มแปลงสัญญาณดิจิทัลจากคอมพิวเตอร์ให้กลายเป็นสัญญาณแอนะล็อกในการรับส่งข้อมูลนี้เอง ถ้าโมเด็มแปลงสัญญาณออกมาอยู่ในรูปของเสียง ซึ่งเป็นสัญญาณแอนะล็อกแบบหนึ่ง เราก็สามารถรับส่งข้อมูลผ่านทางสายโทรศัพท์ได้ โมเด็มทั่ว ๆ ไป ที่เราใช้งานจะเป็นโมเด็มที่แปลงสัญญาณจากคอมพิวเตอร์ให้อยู่ในรูปคลื่นเสียงทั้งนั้น เมื่อโมเด็มได้รับข้อมูลดิจิทัลจากคอมพิวเตอร์มันจะเปลี่ยนในกลายเป็นสัญญาณแอนะล็อก จากนั้นก็นำสัญญาณแอนะล็อกที่ได้นี้เข้ากับสัญญาณพาหะ (Carrie Wave) แล้วส่งออกไปทางสายส่งข้อมูล

โดยปกติเมื่อโมเด็มติดต่อกันได้แล้วจะยังไม่สามารถรับส่งข้อมูลได้ทันที แต่จะต้องตกลงรายละเอียดของวิธีการรับส่งข้อมูลกันก่อน เรียกว่าการทำ handshaking ซึ่งมีขั้นตอนดังนี้คือ เมื่อโมเด็มปลายทางตอบรับแล้ว โมเด็มต้นทางจะทำการทดสอบสภาพสายก่อนว่าสามารถรับส่งข้อมูลได้ดีเพียงใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อที่จะได้เลือกความเร็วสูงที่สุดเท่าที่สายจะรับได้ จากนั้นจะทดสอบความเพี้ยนของสัญญาณต่อไป หลังจากขั้นตอนนี้ โมเด็มจะตกลงกับปลายทางได้ว่าจะใช้ความเร็วในการรับส่งข้อมูลเท่าไหร่ ใช้การผสมสัญญาณแบบไหนที่ความถี่เท่าใด ถ้าใช้ความเร็วสูงสุดตามที่กำหนดไม่ได้ โมเด็มก็จะทำการลดความเร็วลง และทดสอบสภาพสายใหม่ จนกระทั่งได้ความเร็วที่สามารถรับส่งข้อมูลได้อย่างถูกต้อง ซึ่งการลดความเร็วของโมเด็มจะลดลงไปเป็นขั้น ๆ ตามแต่ละมาตรฐานที่กำหนดไว้ จากนั้นโมเด็มจะตกลงกับปลายทางว่าจะใช้วิธีการตรวจสอบความผิดพลาด (Error Detection) แบบใด และใช้การลดขนาดข้อมูล (Data Compression) หรือไม่ ขั้นตอนในการทำ handshaking ของโมเด็มทั้งหมดจะใช้เวลาหลายสิบวินาที เมื่อเสร็จเรียบร้อยแล้วจึงจะสามารถรับส่งข้อมูลกันได้

2.2.2 โมเด็มสำหรับสายตรงและสายโทรศัพท์

โมเด็มแบ่งตามการใช้งานได้สองแบบคือ โมเด็มที่ใช้กับสายตรง (Leased Line) และโมเด็มที่ใช้กับสายโทรศัพท์ (Dial-up Line) โมเด็มที่ใช้กับสายตรงหรือสายเช่าจะส่งข้อมูลด้วยความเร็วสูงจนถึงสูงมาก (9,600 บิต ต่อวินาที ไปจนถึง 2 ล้านบิตต่อวินาที) ผ่านสายที่ลากตรงไปยังจุดหมายปลายทางตายตัว ซึ่งเป็นการติดต่อในลักษณะจุดถึงจุด (Point to Point) จะต่อไปยังจุดอื่น ๆ ไม่ได้ ส่วนมากจะเป็นการใช้ติดต่อส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่มีข้อมูลส่งไปมาเป็นจำนวนมาก เช่น เครือข่าย ATM ของธนาคาร เทอร์มินอล (Terminal) ของคอมพิวเตอร์ขนาดใหญ่ ซึ่งอาจจะเป็นมินิคอมพิวเตอร์ (Mini Computer) หรือเมนเฟรม (Mainframe) เป็นต้น การส่งข้อมูลมักจะส่งเป็นกลุ่มและมีซอฟต์แวร์ (Software) ควบคุมการรับส่งโดยเฉพาะ ข้อดีของโมเด็มแบบที่ใช้กับสายตรง คือส่งข้อมูลได้ที่ความเร็วสูง เนื่องจากสายส่งมีคุณภาพดี เหมาะกับงานส่งข้อมูลจำนวนมากระหว่างจุดสองจุด ข้อเสียก็คือไม่สมารถเปลี่ยนจุดรับข้อมูลไปตามที่ต่าง ๆ ได้ จึงขาดความคล่องตัว

โมเด็มแบบที่ใช้กับสายโทรศัพท์ จะรับส่งข้อมูลด้วยความเร็วตั้งแต่ 300 บิตต่อวินาที จนถึง 56, 000 บิตต่อวินาที ผ่านเครือข่ายโทรศัพท์ เราสามารถรับส่งข้อมูลไปยังที่ต่าง ๆ ได้ตามต้องการ โดยไม่กำหนดจุดรับข้อมูลตายตัวเหมือนอย่างโมเด็มสายตรง เครื่องคอมพิวเตอร์ส่วนบุคคล (PC) และบริการรับส่งข้อมูลต่าง ๆ จะใช้โมเด็มแบบนี้ในการทำงานเกือบทั้งหมด การรับส่งข้อมูลจะเป็นการรับส่งทีละหนึ่งตัวอักษร ไม่ส่งเป็นกลุ่มเรียกว่ารับส่งแบบ อะซิงโครนัส (Asynchronous) ซึ่งโดยปกติแล้วการส่งข้อมูลผ่านสายโทรศัพท์จะส่งด้วยความเร็ว 9,600 บิตต่อวินาทีถึง 28,800 บิตต่อวินาทีเท่านั้น

ความเร็วสูงสุดที่เชื่อถือได้ในการรับส่งข้อมูลของโมเด็มคือ 28,800 บิตต่อวินาที ถ้าใช้โมเด็มความเร็วสูงกว่านี้รับส่งข้อมูลผ่านสายโทรศัพท์เช่น 3,600 บิตต่อวินาที ไปจนถึง 56,000 บิตต่อวินาที อาจทำได้ในบางพื้นที่ แต่ถ้าส่งข้อมูลไกล ๆ ระหว่างจังหวัดหรือระหว่างประเทศ อาจจะมีความผิดพลาดสูง จึงถือความเร็ว 28,800 บิตต่อวินาที เป็นความเร็วสูงสุดสำหรับส่งข้อมูลผ่านสายโทรศัพท์ในปัจจุบัน ข้อดีของการใช้โมเด็มแบบนี้คือ มีความคล่องตัวสูง สามารถรับส่งข้อมูลไปยังที่ต่าง ๆ ได้ไม่จำกัดและไม่จำเป็นต้องจัดหาวงจรสายตรงมาเป็นพิเศษเพื่อส่งข้อมูล ข้อเสียของโมเด็มประเภทนี้คือความเร็วในการส่งข้อมูลต่ำกว่าโมเด็มแบบสายตรง และการส่งข้อมูลเป็นจำนวนมากไปยังจุดปลายจุดเดียวในระยะทางไกล ค่าโทรศัพท์ทางไกลอาจจะแพงกว่าการเช่าวงจรสายตรงมาใช้ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเด็มที่มีขายในท้องตลาดปัจจุบัน บางรุ่นอาจจะทำงานได้เฉพาะกับสายตรงหรือบางรุ่นอาจทำงานได้กับสายโทรศัพท์เท่านั้น โมเด็มที่ใช้กับสายตรงได้เพียงอย่างเดียวจะนำมาต่อใช้กับสายโทรศัพท์ไม่ได้และโมเด็มที่ใช้กับสายโทรศัพท์ได้อย่างเดียวก็จะนำมาใช้งานต่อกับสายตรงไม่ได้เช่นเดียวกัน ยกเว้นโมเด็มบางชนิด ซึ่งเลือกการทำงานในตัวได้ว่าจะต่อกับสายตรงหรือสายโทรศัพท์ โมเด็มที่ใช้งานได้ทั้งสองอย่างก็มักจะมีราคาแพงกว่าโมเด็มที่ใช้กับสายโทรศัพท์เพียงอย่างเดียว ในกรณีที่ไม่ต้องใช้การต่อเครื่องคอมพิวเตอร์ส่วนบุคคลเข้ากับเมนเฟรมหรือมินิคอมพิวเตอร์ การใช้โมเด็มแบบต่อสายโทรศัพท์ได้อย่างเดียวก็นับว่าเพียงพอแล้วสำหรับการรับส่งข้อมูลทั่วไป

2.2.3 โมเด็มในยุคแรก

โมเด็มรุ่นแรก ๆ ที่ใช้รับส่งข้อมูล จะทำหน้าที่แปลงสัญญาณจากคอมพิวเตอร์ ให้กลายเป็นความถี่เสียง แล้วส่งไปตามสายเท่านั้น ไม่มีความสามารถในการหมุนโทรศัพท์อัตโนมัติอย่างโมเด็มในปัจจุบัน โมเด็มแบบที่ไม่มีการต่อเข้ากับสายโทรศัพท์โดยตรง เราเรียกว่าโมเด็มแบบ Acoustic Coupler ซึ่งจะมีรูปร่างรองรับกับหูฟัง (Handset) ของโทรศัพท์แบบมาตรฐาน เมื่อคอมพิวเตอร์ส่งข้อมูลไปให้ผู้รับ Acoustic Coupler จะเปลี่ยนข้อมูลนั้นให้กลายเป็นสัญญาณเสียง แล้วส่งออกทางลำโพงของมัน หูฟังของเครื่องโทรศัพท์จะรับสัญญาณเสียงนี้ส่งไปให้ปลายทางอีกทีหนึ่ง เมื่อเครื่องโทรศัพท์ส่งสัญญาณเสียงให้ Acoustic Coupler มันก็จะรับสัญญาณเสียงนี้ผ่านไมโครโฟน และนำมาเปลี่ยนเป็นสัญญาณดิจิทัลให้คอมพิวเตอร์ต่อไป Acoustic Coupler มีข้อดีคือ มันไม่จำเป็นต้องต่อเข้ากับสายโทรศัพท์ ทำให้เราใช้ที่ไหนก็ได้ส่งข้อมูล เช่นโทรศัพท์สาธารณะ โทรศัพท์ในห้องพักของโรงแรม ฯลฯ ซึ่งที่ต่าง ๆ เหล่านี้บางครั้งเราจะไปตัดต่อสายโทรศัพท์เติมให้มาเข้ากับโมเด็มโดยตรงไม่ได้

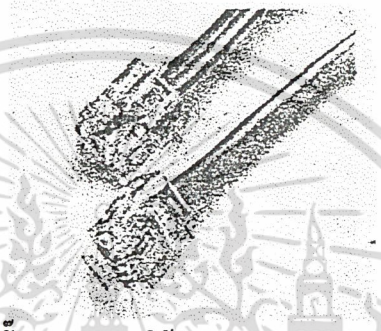
รูปที่ 2.5 โมเด็มแบบ Acoustic Couple

สำหรับข้อเสียของโมเด็มชนิดนี้ก็คือ รับส่งข้อมูลได้ช้า เนื่องจากความเร็วมาตรฐานของ Acoustic Coupler อยู่ที่ 300 บิตต่อวินาที เท่านั้น และจากการที่มันฟังเสียงจากเครื่องโทรศัพท์ ทำให้เสียงรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภายนอกเข้ารบกวนรับส่งข้อมูลได้ง่าย จึงไม่เหมาะที่จะใช้งานในสถานที่ที่มีเสียงดัง และข้อเสียสำคัญอีกข้อก็คือ ไม่สามารถใช้งานกับหูฟังที่มีขนาดและรูปร่างไม่ได้มาตรฐาน เช่นโทรศัพท์แบบแบน ๆ เป็นต้น

โมเด็มปัจจุบันจะรับส่งข้อมูลผ่านสายโทรศัพท์ โดยต่อสายโทรศัพท์เข้าตัวโมเด็มโดยตรงไม่ต้องส่งเสียงผ่าน Acoustic Coupler สายโทรศัพท์จะต่อเข้าโมเด็มโดยใช้ปลั๊กโทรศัพท์ที่เรียกว่า RJ-11 การต่อสายโทรศัพท์เข้ากับโมเด็มโดยตรงจะช่วยป้องกันสัญญาณรบกวนได้ดีขึ้น และทำให้ส่งข้อมูลได้เร็วกว่าแบบเก่า แต่การใช้งานในลักษณะนี้มีข้อเสียก็คือ ถ้าโทรศัพท์ที่ใช้ไม่มีปลั๊กแบบ RJ-11 เพื่อต่อเข้าสายโทรศัพท์ โมเด็มก็จะใช้สายโทรศัพท์นั้นรับส่งข้อมูลไม่ได้ ทำให้ยุ่งยากเวลาต้องการส่งข้อมูลจากนอกสถานที่ เช่น จากที่พักในโรงแรงเป็นต้น



รูปที่ 2.6 ปลั๊กแบบ RJ-11 ใช้สำหรับต่อสายสัญญาณโทรศัพท์

2.2.4 องค์ประกอบในการสื่อสารข้อมูล

จากการที่โมเด็มต่อกับคอมพิวเตอร์ผ่านทาง RS-232C และส่งข้อมูลไปยังเครื่องคอมพิวเตอร์อีกเครื่องหนึ่ง ดังนั้น องค์ประกอบในการรับส่งข้อมูล (Communications Parameter) จึงต้องตรงกันตลอดเส้นทางที่รับส่ง

องค์ประกอบดังกล่าวคือ ความเร็ว (Speed) จำนวนบิตของข้อมูล (Data Bit) จำนวนบิตเริ่มและบิตจบ (Start Bit , Stop Bit) การตรวจสอบพาริตีบิต (Parity Bit) และการเลือกใช้ Echo (Duplex) ในการรับส่ง

ความเร็ว คืออัตราการรับส่งข้อมูลเป็นจำนวนบิตต่อวินาที โมเด็มที่ด้านรับและคอมพิวเตอร์เครื่องรับจะต้องมีความเร็วในการรับส่งข้อมูลเท่ากัน ความเร็วที่ใช้กันทั่วไปมีตั้งแต่ 2400 4800 9600 19200 บิตต่อวินาที ไปจนถึง 38400 57600 หรือ 115000 บิตต่อวินาที ข้อสังเกตคือความเร็วนี้เป็นความเร็วในการส่งผ่านข้อมูลตลอดเส้นทาง แต่การส่งข้อมูลจริงในสายแต่ละช่วง เช่นระหว่างโมเด็มสองตัวอาจไม่สูงเท่านี้ เนื่องจากมีการใช้เทคโนโลยีในการย่อข้อมูลเข้ามาช่วยทำให้ปริมาณข้อมูลที่ต้องส่งลดลง

จำนวนบิตของข้อมูล คือ การกำหนดใช้จำนวนข้อมูลกี่บิตในการส่ง ปกติจะเลือกได้สองแบบ คือ 7 บิตต่อ 1 ตัวอักษร หรือ 8 บิตต่อ 1 ตัวอักษร การรับส่งข้อมูลภาษาไทยเราจำเป็นต้องใช้แบบ 8 บิตต่อตัวอักษรเท่านั้น เนื่องจากภาษาไทยเราใช้รหัสครบทั้ง 8 บิต ถ้าหากส่งไปในแบบ 7 บิตต่อตัวอักษร รหัสภาษาไทยจะถูกตัดบิตที่ 8 กลายเป็นตัวภาษาอังกฤษ ส่วนจำนวนบิตเริ่มและบิตจบนั้นกำหนดไว้ให้ตัวส่งแยกออกว่าข้อมูลจะเริ่มต้นเมื่อไหร่ และจบลงเมื่อใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตเริ่มต้นมักจะใช้ 1 บิตเสมอ และบิตจบข้อมูล จะมีสองแบบ คือ 1 บิต และ 2 บิต การใช้งานทั่วไปมักจะใช้แบบ 1 บิต

การตรวจสอบพาริตี เป็นการตรวจสอบความถูกต้องของการรับข้อมูลที่ส่งมา โดยสามารถเลือกให้ตรวจสอบว่าข้อมูลที่เป็น "1" เป็นจำนวนคู่ (Even) หรือจำนวนคี่ (Odd) หรือไม่ต้องตรวจสอบ (None) เช่นถ้าส่งข้อมูล11000001 ไปให้ผู้รับและมีการตรวจสอบพาริตีแบบจำนวนคู่ พาริตีในกรณีนี้จะมีค่าเป็น "1" เพื่อให้จำนวน "1" ทั้งหมดมีจำนวนเป็นเลขคู่ ถ้ากำหนดการตรวจสอบพาริตีเป็นจำนวนคี่ พาริตีก็จะเป็น "0" เพื่อให้จำนวน "1" ทั้งหมดมีจำนวนเป็นจำนวนคี่ และถ้าไม่มีการตรวจสอบเลย เครื่องส่งก็จะส่งบิตจบ ปิดท้ายข้อมูลทันที ไม่มีการส่งพาริตีไปให้ผู้รับ ส่วนมากถ้ารับส่งข้อมูลแบบ 7 บิต เรามักจะตั้งการตรวจสอบพาริตีเอาไว้ แต่ถ้าส่งรับข้อมูลแบบ 8 บิต ก็จะไม่มีการตรวจสอบพาริตีบิต สำหรับการเลือกใช้ Echo ในการรับส่งก็เป็นการเลือกให้สอดคล้องกับ Full Duplex หรือ Half Duplex นั่นเอง ถ้าเป็นการรับส่งแบบ Full Duplex ก็ต้องเลือกใช้ Echo Off และถ้าเป็นแบบ Half Duplex เราก็เลือกใช้ Echo On เมื่อองค์ประกอบทั้งหมดนี้ตรงกัน การรับส่งข้อมูลก็จะทำได้ถูกต้อง ถ้าส่วนใดส่วนหนึ่งไม่ว่าจะเป็นคอมพิวเตอร์ด้านส่ง โมเด็มด้านส่ง โมเด็มด้านรับหรือคอมพิวเตอร์ด้านรับ เพียงส่วนเดียว ใช้องค์ประกอบในการรับส่งข้อมูลผิดไป การรับส่งข้อมูลก็จะผิดพลาดได้ ทั้งผู้รับและผู้ส่งต้องตกลงกันให้แน่นอนว่าจะใช้องค์ประกอบในการรับส่งข้อมูลแต่ละตัวอย่างไร

2.2.5 การทำงานของเครือข่ายโทรศัพท์

ระบบโทรศัพท์ที่ใช้กันอยู่ในปัจจุบันนั้น เป็นระบบดิจิทัล ซึ่งได้รับการพัฒนามาจากระบบโทรศัพท์แบบเดิมที่เป็นระบบแอนะล็อก โทรศัพท์ในระบบแอนะล็อกนั้นจะใช้อุปกรณ์ไฟฟ้าและกลไกในการส่งเสียงพูดของเราจากต้นทางไปให้ผู้รับปลายทางผ่านอุปกรณ์สลับสายหรืออุปกรณ์สวิตชิ่ง (Switching) ตลอดเส้นทางทางผู้เรียกถึงผู้รับ สัญญาณเสียงจะถูกส่งออกไปในรูปของสัญญาณไฟฟ้าในแบบบแอนะล็อก และเปลี่ยนกลับเป็นสัญญาณเสียงที่ลำโพงของผู้รับ การใช้ระบบแอนะล็อกในโทรศัพท์มีข้อเสียคือ การทำงานช้าและอุปกรณ์สวิตชิ่งมีขนาดใหญ่รวมทั้งยังต้องใช้ไฟมากอีกด้วย ทกให้มีการพัฒนามาใช้โทรศัพท์ระบบดิจิทัลดังเช่นทุกวันนี้

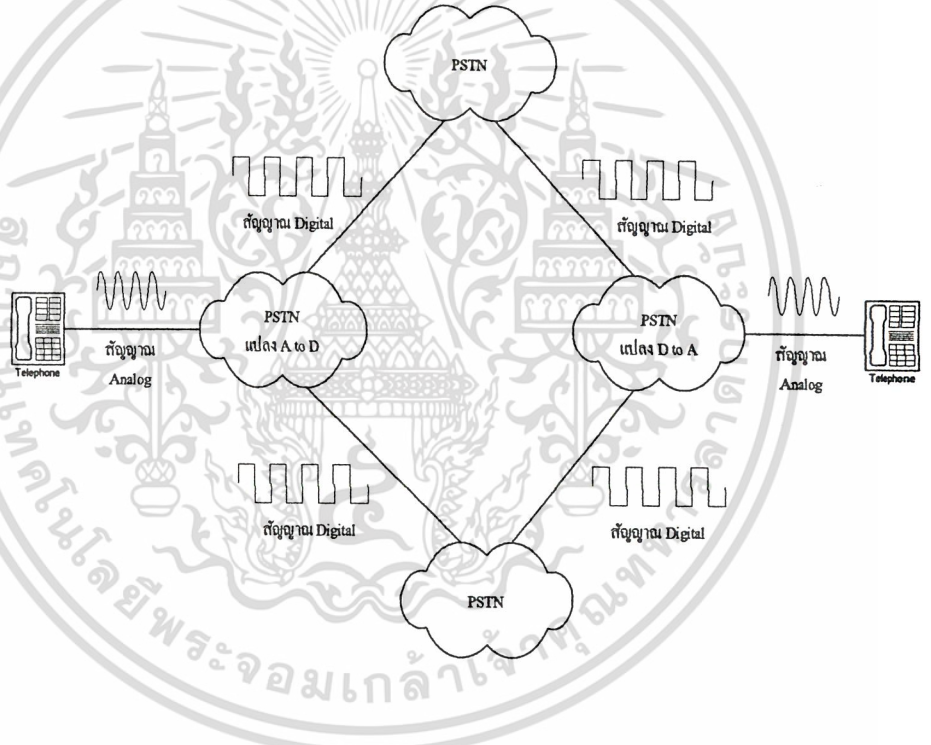
การทำงานของโทรศัพท์ระบบดิจิทัลจะมีความซับซ้อนกว่าระบบแอนะล็อกมาก เมื่อเรายกหูโทรศัพท์ขึ้นและกดหมายเลขปลายทาง สัญญาณจะถูกส่งไปให้ชุมสายโทรศัพท์เพื่อนำหมายเลขปลายทางมาแปลงเป็นรหัสดิจิทัล และให้รหัสนี้ค้นหาตำแหน่งของผู้รับปลายทางต่อไป ซึ่งการทำงานภายในชุมสายจะใช้ระบบดิจิทัลทั้งหมด โดยไม่มีระบบกลไกแบบเก่าเข้ามาเกี่ยวข้องเลย เรียกว่าระบบ Stored Program Control (SPC) ตัวชุมสายโทรศัพท์จะเปรียบเสมือนเครื่องคอมพิวเตอร์ขนาดใหญ่ ที่มีหน้าที่เชื่อมต่อผู้เรียกและผู้รับให้สนทนากันได้นั่นเอง ถ้าเครื่องปลายทางว่าง ชุมสายโทรศัพท์จะส่งสัญญาณเรียกให้กระดิ่งดังขึ้น เมื่อปลายทางรับสายแล้วทั้งผู้รับเรียกและผู้รับก็เริ่มสนทนากันได้

สัญญาณเสียงจากผู้พูดจะถูกแปลงเป็นสัญญาณเสียงแอนะล็อก ส่งไปให้ชุมสายโดยเครื่องโทรศัพท์ จากนั้นชุมสายโทรศัพท์จะเปลี่ยนสัญญาณนี้ให้เป็นสัญญาณดิจิทัล ขนาด 64 กิโลบิตต่อวินาที โดยใช้อุปกรณ์ Analog to Digital converter หรือ A to D (A/D) ซึ่งขบวนการนี้จะอ่านค่าของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณแอนะล็อกของเสียงพูด 8,000 ครั้งต่อวินาที โดย 1 ครั้งจะได้ค่าของสัญญาณออกมา 8 บิต (8,000 ครั้งต่อวินาที X 8 บิต = 64,000 บิตต่อวินาที) เราเรียกขบวนการแปลงสัญญาณนี้ว่า Pulse code Modulation (PCM) แล้วจึงส่งสัญญาณดิจิทัล 64 กิโลบิตต่อวินาทีนี้ไปให้ชุมสายโทรศัพท์ปลายทาง เมื่อถึงปลายทางสัญญาณดิจิทัลก็จะถูกเปลี่ยนกลับมาเป็นสัญญาณแอนะล็อกโดย Digital to Analog converter หรือ D to A (D/A) ส่งไปให้เครื่องรับปลายทางต่อไป เครื่องรับปลายทางก็จะนำสัญญาณแอนะล็อกนี้มาเปลี่ยนเป็นเสียงออกทางลำโพงของเครื่องรับ จะเห็นได้ว่าจากต้นทางถึงปลายทางเราให้สัญญาณดิจิทัลเกือบทั้งหมด เหลือเพียงช่วงสุดท้ายที่ต่อเข้ากับเครื่องโทรศัพท์เท่านั้นที่ยังใช้สัญญาณแอนะล็อกอยู่ ดังรูปที่ 2.7

ชุมสายโทรศัพท์ระบบดิจิทัลที่เราใช้งานอยู่ในปัจจุบัน จะมีขนาดเล็กกว่าชุมสายแอนะล็อกสมัยก่อนมาก เนื่องจากทำงานไม่มีส่วนของระบบกลไกเข้ามาเกี่ยวข้อง ทำให้มีน้ำหนักเบากว่าและกินไฟน้อยกว่า ความเร็วในการทำงานก็เร็วกว่าเดิม รวมทั้งมีบริการพิเศษเพิ่มเติม เช่นการประชุมทางโทรศัพท์ การรับสายซ้อน ฯลฯ อีกด้วย



รูปที่ 2.7 แสดงการทำงานของระบบโทรศัพท์

การรับส่งข้อมูลโดยใช้เสียงผ่านเครือข่ายโทรศัพท์นั้น เป็นวิธีที่น่าสนใจมาก เนื่องจากว่าโทรศัพท์มีอยู่ทุกทนทุกแห่ง ไม่ว่าจะเป็นที่บ้าน ที่ทำงาน หรือสถานที่ต่าง ๆ และเราสามารถใช้โทรศัพท์ติดต่อไปยังที่ต่าง ๆ ได้ทั่วโลก ราคาค่าบริการของโทรศัพท์ทางไกลในปัจจุบันก็ไม่แพงจนเกินไปนัก การรับส่งข้อมูลผ่านสายโทรศัพท์จึงทำได้ครอบคลุมพื้นที่กว้างขวางมากกว่าการใช้ระบบอย่างอื่น ๆ แต่เนื่องจากระบบโทรศัพท์ไม่ได้ถูกออกแบบมาให้ใช้รับส่งข้อมูลคอมพิวเตอร์หรือข้อมูลอื่น ๆ นอกจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับส่งเสียงพูดได้เท่านั้น ระบบโทรศัพท์จึงมีขีดจำกัดหลายอย่างที่ทำให้การรับส่งข้อมูลต้องทำด้วยความระมัดระวัง ขีดจำกัดของระบบโทรศัพท์ที่มีผลต่อการรับส่งข้อมูลก็คือ ช่วงแบนด์วิดท์ (Bandwidth) ของสัญญาณ ระบบกำจัดเสียงสะท้อนของโทรศัพท์ (Echo Suppressor) และเสียงรบกวนในสาย (Background Noise) ขีดจำกัดเหล่านี้ ทำให้การรับส่งข้อมูลทำได้ช้าและอาจผิดพลาดได้

ระบบการรับฟังเสียงของคนเรา สามารถรับรู้เสียงความถี่ได้ตั้งแต่ 20 Hz ถึง 20,000 Hz โดยประมาณ ถ้าความถี่สูงกว่าและต่ำกว่านี้ เราจะไม่ได้ยินเสียงนั้น เมื่อมีการคิดระบบโทรศัพท์ขึ้นมาครั้งแรกนั้น เทคโนโลยียังไม่สูงมากพอที่จะสร้างระบบโทรศัพท์ให้รับส่งเสียงได้เทียบเท่ากับที่หูคนได้ยิน การออกแบบจึงพิจารณาทางด้านเทคนิคและราคาที่พอยอมรับได้ในสมัยนั้น ผลที่ออกมาก็คือระบบโทรศัพท์สามารถรับส่งสัญญาณเสียงได้ตั้งแต่ความถี่ 300 Hz จนถึงประมาณ 3,400 Hz โดยมีแบนด์วิดท์ที่รับส่งได้ตั้งแต่ประมาณ 3,350 Hz ถึง 3,400 Hz ลบด้วยความถี่ต่ำ 350 Hz ถึง 300 Hz เหลือแบนด์วิดท์ที่รับส่งได้จริงประมาณ 3,000 Hz ซึ่งความถี่ช่วงดังกล่าวนี้มากพอสำหรับการพูดคุยของคนทั่วไปด้วยคุณภาพของเสียงดีพอสมควร แต่ไม่มากพอให้เราส่งข้อมูลคอมพิวเตอร์ได้อย่างสะดวกผ่านสายโทรศัพท์ เพราะในการบส่งข้อมูลทั่วไปนั้น ถ้าสายส่งมีแบนด์วิดท์สูงเท่าใด ก็จะสามารถรับส่งข้อมูลได้เร็วมากขึ้นเท่านั้น

2.2.5.1 ขีดจำกัดต่าง ๆ ของเครือข่ายโทรศัพท์

ขีดจำกัดข้อแรกก็คือช่วงความถี่ 300 Hz ถึง 3,400 Hz ที่สายโทรศัพท์สามารถส่งสัญญาณได้นี้นับว่าเป็นสายส่งที่มีแบนด์วิดท์แคบมาก เมื่อเรานำมาใช้รับส่งข้อมูล เพราะว่าตามธรรมดาแล้ว ความเร็วสูงสุดของการรับส่งข้อมูลจะมีค่าต่ำกว่าความถี่สูงสุดของสายส่งเสมอ ถ้าหากไม่ใช้เทคนิคพิเศษเข้าช่วย สายโทรศัพท์ก็จะรับส่งข้อมูลได้ไม่เกิน 3,000 บิตต่อวินาทีเท่านั้น ซึ่งในยุคแรกเริ่มของการรับส่งข้อมูลผ่านสายโทรศัพท์ จะมีความเร็วเพียง 300 บิตต่อวินาทีเท่านั้น และได้มีการพัฒนาต่อกันมาจนรับส่งได้ 1,200 ถึง 2,400 บิตต่อวินาที ตามลำดับ เมื่อเทคโนโลยีก้าวหน้าขึ้น และจากการที่แบนด์วิดท์ของสัญญาณจำกัดมากในสายโทรศัพท์นี้เอง ดังนั้นในการรับส่งข้อมูลที่มีความเร็วสูงกว่า 9,600 บิตต่อวินาที ผู้ผลิตโมเด็มจึงจำเป็นต้องใช้เทคนิคการเข้ารหัส และการผสมสัญญาณที่ซับซ้อนมากเพื่อแก้ปัญหาดังกล่าว ดังนั้นการรับส่งข้อมูลความเร็วสูงจึงถูกรบกวนได้ง่าย และมีอัตราการผิดพลาดในการรับส่งข้อมูลสูงขึ้นเป็นเงาตามตัว วงจรสายตรงนับว่าได้เปรียบการส่งผ่านระบบโทรศัพท์มากในข้อนี้ เนื่องจากสายส่งและวงจรมีคุณภาพดีกว่า ทำให้รับส่งข้อมูลความเร็วสูงได้อย่างไม่มีปัญหา ด้านขีดจำกัดของช่วงความถี่เหมือนที่เกิดขึ้นกับสายโทรศัพท์ทั่วไป

ขีดจำกัดข้อที่ 2 ของการส่งข้อมูลผ่ายล เขโทรศัพท์ก็คือ วงจรกำจัดเสียงสะท้อน ของระบบโทรศัพท์ เนื่องจากการส่งสัญญาณไฟฟ้าผ่านสายส่งเป็นระยะทางไกล ๆ นั้นสัญญาณที่ส่งออกไปจะมีส่วนหนึ่งสะท้านกลับมายังผู้ส่ง ซึ่งเราเรียกว่าเกิด Echo ขึ้น ปรากฏการณ์อันนี้เกิดขึ้นกับคลื่นทุกชนิด ไม่จำกัดว่าจะต้องเป็นสัญญาณไฟฟ้าเท่านั้น เมื่อมันเกิดขึ้นในสายโทรศัพท์ เราก็จะได้ยินเสียงตัวเองสะท้อนกลับมาในลักษณะของเสียงก้อง คล้าย ๆ กับเวลาที่เราพูดอยู่ในถ้านั้นเอง การเกิดเสียงสะท้อนนี้นอกจากจะทำให้การรับฟังโทรศัพท์มีคุณภาพต่ำลงแล้ว ยังทำให้การสนทนาเป็นไปอย่างลำบากอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบโทรศัพท์ใช้วงจรกำจัดเสียงสะท้อนเข้ามาช่วยแก้ปัญหาอันนี้ การทำงานของมันจะยอมให้สัญญาณจากผู้พูดเดินทางไปถึงผู้รับได้ทางเดียว สัญญาณที่สะท้อนกลับมาจะถูกกั้นเอาไว้ไม่ให้กลับมาหาผู้พูดได้ เมื่อผู้ฟังพูดสวนกลับไปวงจรกำจัดเสียงสะท้อนก็จะสลับทิศทางไปในทางตรงข้าม เพื่อให้สัญญาณจากผู้พูดถูกกั้นไม่ให้เสียงสะท้อนกลับมาหาตัวเองอีกเช่นเดียวกัน วิธีการนี้ทำให้เสียงสะท้อนถูกกำจัดออกไปได้ วงจรกำจัดเสียงสะท้อนจะรู้ว่าใครเป็นผู้พูดโดยดูจากความแรงของสัญญาณ ใช้กฎเกณฑ์ที่ว่าด้านที่พูดจะมีสัญญาณแรงกว่าด้านรับเสมอ และการสลับทิศทางของวงจรกำจัดเสียงสะท้อนจะทำด้วยความเร็วมากจนเราไม่รู้สึกผิดปกติเมื่อพูดและฟังธรรมดา ขณะที่เราสลับกันพูดโทรศัพท์ในการสนทนาทั่วไป วงจรกำจัดเสียงสะท้อนจะทำงานโดยสลับทิศทางตลอดเวลา เพื่อป้องกันเสียงสะท้อนไม่ให้เกิดขึ้น

ในขณะที่เราส่งข้อมูลผ่านสายโทรศัพท์ วงจรกำจัดเสียงสะท้อนจะทำให้การรับส่งข้อมูลมีปัญหามาก เนื่องจากการสลับทิศทางของวงจรกำจัดเสียงสะท้อนจะทำให้สัญญาณขาดหายไปในช่วงนั้น ถึงแม้ว่าจะเป็นช่วงเวลาสั้น ๆ ก็อาจทำให้ข้อมูลบางส่วนหายไปได้ นอกจากนี้การสลับทิศทางของวงจรกำจัดเสียงสะท้อนยังทำให้เสียเวลาในการส่งข้อมูลอีกด้วย เพราะมันต้องใช้ช่วงเวลาหนึ่งในการสลับทิศทางเสมอสิ่งเหล่านี้มีผลมาก เมื่อเราส่งข้อมูลผ่านสายโทรศัพท์ด้วยความเร็วสูง

โมเด็มปัจจุบันจะหยุดการทำงานของวงจรกำจัดเสียงสะท้อนได้ โดยส่งคลื่นเสียงความถี่ประมาณ 2,100 Hz ไปยังชุมสายโทรศัพท์นานประมาณครึ่งวินาที วงจรกำจัดเสียงสะท้อนก็จะหยุดสลับทิศทาง ทำให้การรับส่งข้อมูลเป็นไปได้อย่างไม่มีปัญหา

นอกจากนี้สัญญาณรบกวนในสายโทรศัพท์ ก็เป็นปัญหาอีกอันหนึ่งในการรับส่งข้อมูลปกติ สัญญาณรบกวนจะมีอยู่ระดับหนึ่งตลอดเวลาในสาย ซึ่งอาจจะเกิดจากคลื่นแม่เหล็กไฟฟ้าที่อื่นถูกเหนี่ยวนำเข้ามาในสาย หรือเกิดจากอุปกรณ์รับส่งของชุมสายโทรศัพท์เองก็ได้ เมื่อสัญญาณรบกวนมีขนาดไม่มากนักเทียบกับสัญญาณข้อมูลที่เราใช้รับส่ง เราก็ยังคงแยกออกว่าส่วนไหนคือข้อมูล และส่วนไหนคือสัญญาณรบกวน แต่ในระยะทางไกล ๆ สัญญาณข้อมูลที่ส่งออกไปจะมีกำลังอ่อนลงตามระยะทาง ทำให้ความแตกต่างระหว่างสัญญาณรับส่งและสัญญาณรบกวนน้อยลง การรับส่งข้อมูลจะทำได้ลำบากมากยิ่งขึ้น

โมเด็มที่รับส่งข้อมูลความเร็วไม่เกิน 9,600 บิตต่อวินาที มักไม่ค่อยมีปัญหาเรื่องของสัญญาณรบกวน เนื่องจากที่ความเร็วขนาดนี้การผสมสัญญาณในการรับส่งยังไม่ซับซ้อนมากนัก รูปคลื่นที่ส่งออกไปยังมีความแตกต่างกันชัดเจน เมื่อถูกรบกวนจากสัญญาณอื่น ๆ ก็ยังสามารถแยกออกได้ง่ายกว่าข้อมูลที่ส่งมาเป็นอย่างไร ปัญหาของสัญญาณรบกวนในสายจะมีผลกับโมเด็มความสูง เช่น 19,200 28,800 หรือ 33,600 บิตต่อวินาที ค่อนข้างมาก เพราะที่ความเร็วสูงโมเด็มจะต้องใช้เทคนิคการผสมสัญญาณที่ซับซ้อนมากเพื่อส่งข้อมูลไปในสายโทรศัพท์ซึ่งมีแบนด์วิดท์กว้างเพียง 3,000 Hz เท่านั้น รูปคลื่นของสัญญาณที่ส่งออกไปจึงต้องมีความแม่นยำสูง ผู้รับถึงจะรับข้อมูลได้ถูกต้อง สัญญาณรบกวนจะทำให้รูปคลื่นผิดไปจากเดิม ซึ่งอาจมีผลทำให้การรับข้อมูลผิดพลาดด้วย โมเด็มความเร็วสูงจึงจำเป็นต้องมีฮาร์ดแวร์ (Hardware) พิเศษมาทำหน้าที่ตัดสัญญาณรบกวนในสายออกโดยเฉพาะ ทำให้ราคาของโมเด็มแพงมากขึ้นอีก

61943
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตามปัญหาทั้งหมดนี้ได้รับการแก้ไขจนเป็นที่น่าพอใจแล้วระดับหนึ่งในปัจจุบัน การรับส่งข้อมูลผ่านสายโทรศัพท์ด้วยความเร็วสูงถึง 28,800 หรือ 33,600 บิตต่อวินาทีนั้น นับว่าเป็นการเอาชนะขีดจำกัดต่าง ๆ ดังกล่าวด้วยเทคโนโลยีสมัยใหม่ของเรา

2.2.6 มาตรฐานของโมเด็มตาม CCITT (ITU-T) V-Series

โมเด็มจำเป็นต้องมีมาตรฐานเช่นเดียวกับอุปกรณ์อื่น ๆ เพื่อให้ผู้ผลิตแต่ละบริษัทผลิตโมเด็มออกมาแล้วใช้รับส่งข้อมูลกับบริษัทอื่น ๆ ได้ มาตรฐานของโมเด็มจะแบ่งออกเป็นสองส่วน คือมาตรฐานในส่วนของฮาร์ดแวร์ที่โมเด็มใช้ และอีกส่วนหนึ่งคือมาตรฐานในส่วนของซอฟต์แวร์ที่ควบคุมการทำงานของโมเด็มหรือคำสั่งของโมเด็ม โดยมาตรฐานนี้กำหนดขึ้นจากองค์กรมาตรฐานสื่อสารสากล หรือ CCITT (International Telephone and Telegraph Consultative Committee) ที่ปัจจุบันเปลี่ยนชื่อมาเป็น ITU-T (International Telecommunication Union-Telecommunication) ซึ่งจะครอบคลุมเกี่ยวกับความเร็วในการรับส่งข้อมูล ความถี่ที่ใช้ และเทคนิคการผสมสัญญาณในสาย ฯลฯ

มาตรฐานทางการสื่อสารและรับส่งข้อมูล เนื่องจากมีสมาชิกเกือบทุกประเทศทั่วโลก ผู้ผลิตรายใหญ่จึงปฏิบัติตามมาตรฐานของ CCITT หรือ ITU-T ทำให้โมเด็มคนละยี่ห้อกันที่ใช้มาตรฐานเดียวกันสามารถรับส่งข้อมูลได้ทันที

ความเร็วในการรับส่งข้อมูลของโมเด็มนั้นกำหนดเป็น บิตต่อวินาที (bit per second, bps) หรือบิตเรท (Bit Rate) ซึ่งแตกต่างจากอัตราการเปลี่ยนแปลงของสัญญาณไฟฟ้าในสายส่งหรือที่เรียกกันว่า Baud Rate ในสมัยก่อนการรับส่งข้อมูลใช้เทคนิคการผสมสัญญาณแบบง่าย ๆ เช่นการเปลี่ยนแปลงความถี่ตามข้อมูล “0” และ “1” ที่ได้รับ อัตราการส่งข้อมูลและอัตราการเปลี่ยนแปลงของสัญญาณในสายส่งจึงมีค่าเท่ากัน หรือพูดอีกอย่างว่าสัญญาณรูปคลื่น 1 ลูก จะแทนข้อมูลเพียง 1 บิต เราจึงถือว่าอัตราการเปลี่ยนแปลงของสัญญาณในสายส่ง ก็คืออัตราการส่งข้อมูลนั่นเอง ต่อมาเทคนิคการผสมสัญญาณซับซ้อนมากขึ้นทำให้เราสามารถส่งข้อมูลได้มากขึ้นกว่าเดิม โดยอัตราการเปลี่ยนแปลงของสัญญาณในสายยังคงเท่าเดิม ดังนั้นเมื่อเราพูดว่าโมเด็มรับส่งข้อมูลที่ความเร็ว 1,200 Baud เราจะไม่ทราบเลยว่าโมเด็มนั้นรับส่งข้อมูลได้กี่บิตต่อวินาที เนื่องจากว่าถ้าโมเด็มผสมสัญญาณ 1 บิตต่อหนึ่งลูกคลื่น โมเด็มนั้นจะรับส่งข้อมูลที่ความเร็ว 1,200 บิตต่อวินาที หรือถ้าโมเด็มผสมสัญญาณ 4 บิตต่อหนึ่งลูกคลื่น ที่เปลี่ยนแปลงในสายส่งโมเด็มก็จะรับส่งข้อมูลได้เร็วถึง 4,800 บิตต่อวินาที โดยยังคงมีอัตราการเปลี่ยนแปลงสัญญาณในสายส่งเท่ากับ1,200 Baud เหมือนเดิม เพราะฉะนั้นเราจึงเลิกใช้คำว่า Baud Rate สำหรับบอกความเร็วการรับส่งข้อมูลของโมเด็มและหันมาใช้คำว่า บิตเรท หรือ อัตราการส่งข้อมูลเป็นบิตต่อวินาทีแทน ซึ่งสื่อความหมายได้เข้าใจตรงกันมากกว่า

มาตรฐานของโมเด็มที่ใช้อยู่ในปัจจุบัน เป็นไปตามที่องค์กรมาตรฐานสื่อสารสากล หรือ CCITT (ITU-T) เป็นผู้กำหนดขึ้น โดยมีชื่อเรียกแต่ละมาตรฐานของโมเด็มขึ้นต้นด้วยอักษร “V” และตามด้วยตัวเลข จึงมีการเรียกมาตรฐานอันนี้อีกชื่อหนึ่งว่า V-Series นอกจากมาตรฐานของโมเด็มแล้ว CCITT หรือ ITU-T ยังเป็นผู้กำหนดมาตรฐานทางการสื่อสารอื่น ๆ เช่น มาตรฐานของการสื่อสารผ่านดาวเทียม มาตรฐานของโทรสาร (Facsimile , Fax) มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลต่าง ๆ ทั้งในแบบดิจิทัลและแอนะล็อก รวมถึงมาตรฐานเกี่ยวกับระบบโทรศัพท์อีกด้วย มาตรฐานที่ CCITT หรือ ITU –T เป็นผู้กำหนด ได้รับการยอมรับทั่วโลก การติดต่อสื่อสารระหว่างประเทศจึงเป็นไปได้อย่างไม่มีปัญหา เนื่องจากทุก ๆ คนต่างก็ทำตามมาตรฐานเดียวกัน

มาตรฐานที่ขึ้นต้นด้วยอักษร "V " ไม่ใช้มาตรฐานของโมเด็มทั้งหมด บางมาตรฐานอาจหมายถึงการเชื่อมต่อแบบอื่น ๆ ก็ได้ เช่น V.24 เป็นมาตรฐานการรับส่งข้อมูลแบบอนุกรมเทียบได้กับ RS-232C นั่นเอง และ V.35 หมายถึงการรับส่งข้อมูลแบบอนุกรมความเร็วสูงเป็นต้น ในที่นี้จะขอยกตัวอย่างมาตรฐานของโมเด็ม (โดยรายละเอียดมาตรฐานของโมเด็มที่ใช้กันมากตาม CCITT (ITU-T) V-Series ตั้งแต่ความเร็วต่ำไปจนถึงความเร็วสูง และรายละเอียดของแต่ละมาตรฐาน อยู่ในส่วนของภาคผนวก)

V.90 พัฒนาจากมาตรฐาน V.32 bis ซึ่งใช้สำหรับโมเด็มความเร็ว 56,000 บิตต่อวินาที มาตรฐานนี้ ITU-T ประกาศออกใช้งาน เมื่อเดือนกันยายน ปี 1998 โดยอาศัยหลักการที่เครือข่ายดิจิทัลสามารถรองรับการจราจรทางเสียงที่มีความเร็ว 64 Kbps ต่อสายแต่ละเส้น และ การที่จะ digitize (แปลงสัญญาณแอนะล็อกเป็นดิจิทัล) สัญญาณเสียงแอนะล็อกด้วยที่มีความถี่ 3 KHz นั้น จำเป็นต้องทำการสุ่มสัญญาณ (Sampling) เสียงด้วยอัตราสองเท่าของความถี่นั้น โมเด็มความเร็ว 56 Kbps ใช้ประโยชน์จากข้อเท็จจริงที่ว่า AOL (American Online – ผู้ให้บริการ online รายใหญ่ที่สุดในประเทศสหรัฐ) และผู้ให้บริการ Internet รายใหญ่อื่น ๆ ล้วนแล้วแต่มีการเชื่อมโยงระบบดิจิทัลโดยตรงกับเครือข่ายโทรศัพท์อยู่แล้ว และไม่จำเป็นต้องทำการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล ด้วยในขั้นตอนการส่งขึ้น (Upstream) นี้จะช่วยสงวนรักษาแบนด์วิดท์ของช่องทางดิจิทัลโดยสมบูรณ์ตลอดทางไปยังจุดสุดท้าย ข้ามเครือข่ายโทรศัพท์ผู้ใช้ความเร็ว 56 Kbps หรือ ก็คือช่วงขาลง (Downstream)

V.92 เป็นข้อกำหนดหรือมาตรฐานสำหรับ Dial-up โมเด็มความเร็ว 28,800 บิตต่อวินาที ใช้เทคนิคการผสมสัญญาณแบบใหม่ที่แตกต่างไปจากเดิมอย่างสิ้นเชิง คือใช้หลักการของการเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก (Digital to Analog converter) มาใช้แทนการผสมสัญญาณแบบ Trellis Encoding การรับส่งข้อมูลจะแปลงข้อมูลดิจิทัลครั้งละ 7 บิตส่งออกไป 8,000 ครั้งต่อวินาที (7x800=56,000 บิตต่อวินาที) แล้วส่งออกไปเป็นสัญญาณแอนะล็อกตามสายโทรศัพท์ โมเด็มชนิดนี้จะมีความเร็วการรับและส่งข้อมูลไม่เท่ากันทั้งด้าน คือด้านรับเข้ามาจะมีความเร็วสูง 56,000 บิตต่อวินาที แต่ด้านส่งออกไปจะมีความเร็วเพียง 28,800 บิตต่อวินาทีเท่านั้น ส่วนโมเด็มด้านส่งก็จะกลับกัน คือ ส่งข้อมูลที่ความเร็ว 56,000 บิตต่อวินาที และรับข้อมูลเข้ามาที่ความเร็ว 28,800 หรือ 36,600 บิตต่อวินาทีตามมาตรฐาน V.34 ซึ่งเป็นมาตรฐาน V.92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

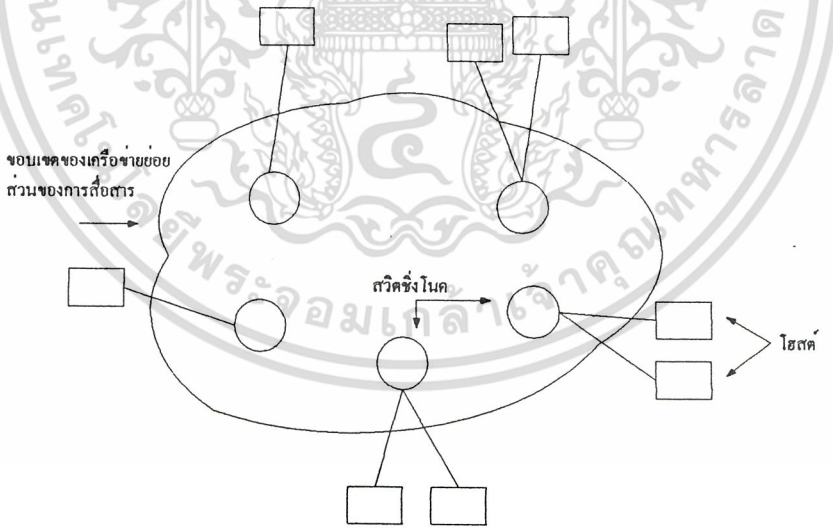
ที่กำหนดโดย ITU-T ประกอบด้วยคุณสมบัติใหม่ 3 ประการ ได้แก่ Quick Connect , Modem on Hold , V.PCM Upstream

มาตรฐานของโมเด็ม V-Series ที่กล่าวถึงทั้งหมดนี้ เป็นมาตรฐานที่พบเห็นได้ทั่วไป ซึ่งยังมีบางมาตรฐานไม่ได้นำมาพูดถึง เนื่องจากมีการใช้งานพิเศษเฉพาะงานเท่านั้น ส่วนมาตรฐานของโมเด็มตามแบบสหรัฐอเมริกา หรือที่เราเรียกว่า Bell Standard ปัจจุบันค่อย ๆ ลดความนิยมลง เนื่องจากประเทศต่าง ๆ ใช้ตามมาตรฐานของ CCITT หรือ ITU-T เป็นหลัก และในประเทศไทยก็ใช้มาตรฐานของ CCITT หรือ ITU-T เช่นกัน

2.3 เครือข่ายคอมพิวเตอร์ (Computer Network)

2.3.1 องค์ประกอบของเครือข่ายคอมพิวเตอร์ (Network Compunents)

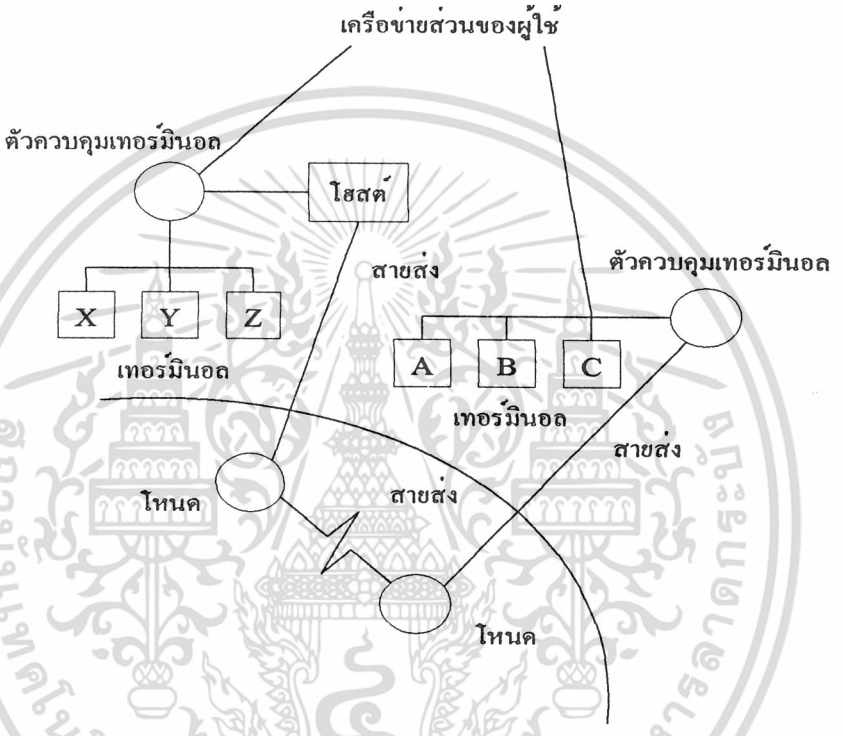
เครือข่ายคอมพิวเตอร์จะประกอบด้วยโหนด (node) 2 ชนิด คือ โหนดที่ทำหน้าที่ประมวลข้อมูล หรือ Computing node และ โหนดที่ทำหน้าที่ส่งข้อมูล หรือ Switching node ผ่านสายส่งจากต้นทางไปยังปลายทางได้ ดังแสดงในรูปที่ 2.8 ปกติแล้วเมื่อผู้ใช้เครือข่ายต้องการประมวลผลข้อมูลจะอาศัยเทอร์มินัลในการส่งข้อมูลไปประมวลผลที่คอมโหนดประมวลผล เช่น โฮสต์ (host) โดยที่ข้อมูลเหล่านั้นจะผ่านโหนดส่งข้อมูลต่าง ๆ ซึ่งจะส่งข้อมูลให้ถึงปลายทาง และหลังจากที่โฮสต์ประมวลผลเสร็จก็จะส่งผลผ่านสวิตชิงโหนดต่าง ๆ กลับมาให้ผู้ใช้ได้ ดังนั้นจึงมักแบ่งเครือข่ายคอมพิวเตอร์ออกเป็น 2 ส่วนใหญ่ ๆ ตามหน้าที่ของโหนดนั้น ๆ คือแบ่งเป็น เครือข่ายย่อยส่วนของผู้ใช้ (User Subnetwork) และเครือข่ายย่อยส่วนของการสื่อสาร (Communication Subnetwork)



รูป 2.8 แสดงเครือข่ายคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 เครือข่ายย่อยส่วนของผู้ใช้ (User Subnetwork) คือ ส่วนที่ผู้ใช้ต้องการส่งข้อมูลไปประมวลผลแล้วแสดงผลออกมาให้ทราบ เครือข่ายส่วนนี้ประกอบด้วยโฮสต์คอมพิวเตอร์ (Host computer) เทอร์มินัล (Terminal) และตัวควบคุมเทอร์มินัล (Terminal controllers) โดยที่โฮสต์จะมีทรัพยากรต่าง ๆ เช่น ฐานข้อมูล โปรแกรมประยุกต์ที่ผู้ใช้สามารถใช้งานได้เป็นต้น ผู้ใช้จะอาศัยเทอร์มินัลในการป้อนข้อมูลและแสดงผลที่ได้จากการประมวลผลนั่นเอง โดยผ่านการควบคุมเทอร์มินัลหรือตัวควบคุมระบบการสื่อสาร (Communication controller) ซึ่งทำหน้าที่ควบคุมการทำงานของกลุ่มของเทอร์มินัลดังแสดงในรูปที่ 2.9



รูป 2.9 แสดงโดยผ่านการควบคุมเทอร์มินัลหรือตัวควบคุมระบบการสื่อสาร

การใช้งานคอมพิวเตอร์ในลักษณะที่ผู้ใช้ป้อนข้อมูลผ่านเทอร์มินัลแล้ว การประมวลผลเกิดที่โฮสต์คอมพิวเตอร์ศูนย์กลาง มักจะถูกเรียกว่า ระบบโฮสต์-เทอร์มินัล (Host-Terminal) ซึ่งในกรณีที่มีผู้ใช้หลายคนและประมวลผลคอมพิวเตอร์พร้อม ๆ กัน จะทำให้การทำงานของระบบคอมพิวเตอร์ช้ามาก ดังนั้นในปัจจุบันจึงมีการใช้งานคอมพิวเตอร์ในลักษณะของผู้ใช้บริการ/ผู้ให้บริการ (Client/Serer) มากขึ้น ซึ่งเป็นรูปแบบหนึ่งของการประมวลผลแบบกระจาย กล่าวคือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์จะให้บริการ เช่น บริการเก็บแฟ้มข้อมูล (File Server) บริการพิมพ์งานออกเครื่องพิมพ์ (Print Server) บริการประมวลผลงานเฉพาะอย่าง (Computing Server) หรือบริการการติดต่อสื่อสารกับเครือข่าย (Communication Server) เป็นต้น ส่วนเครื่องคอมพิวเตอร์ของผู้ใช้บริการ (Client) นั้นก็จะมีส่วนประมวลผลเป็นของตัวเองโดยจะส่งคำขอใช้งาน (Request) ไป เมื่อเครื่องผู้ให้บริการ (Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานตามคำขอ ก็จะส่งผลลัพธ์ตอบกลับมาเครื่องของผู้ใช้บริการก็จะสามารถประมวลผลได้ด้วยตัวเอง ซึ่งการประมวลลักษณะนี้ถึงแม้ว่าจะมีผู้ใช้จำนวนมากก็ไม่ทำให้การทำงานของระบบช้าแต่อย่างใด

2.3.2 เครือข่ายย่อยส่วนของการสื่อสาร (Communication Subnetwork) จะประกอบด้วยโหนดหลายโหนดคิดต่อสื่อสารกันผ่านสายส่ง (Transmission Link) เพื่อจะส่งข้อมูลจากต้นทางถึงปลายทางได้อย่างถูกต้องและรวดเร็ว ตัวอย่างของเครือข่ายส่วนนี้ ได้แก่ เครือข่ายโทรศัพท์ เครือข่ายดาวเทียม เครือข่ายวิทยุ เป็นต้น และจากรูปที่ 2.10 จะแสดงให้เห็นว่าเครือข่ายย่อยส่วนของการสื่อสารจะประกอบด้วยโหนดเครือข่ายสายส่ง (Network node) และตัวแปลงสัญญาณ (Singnal Converters)

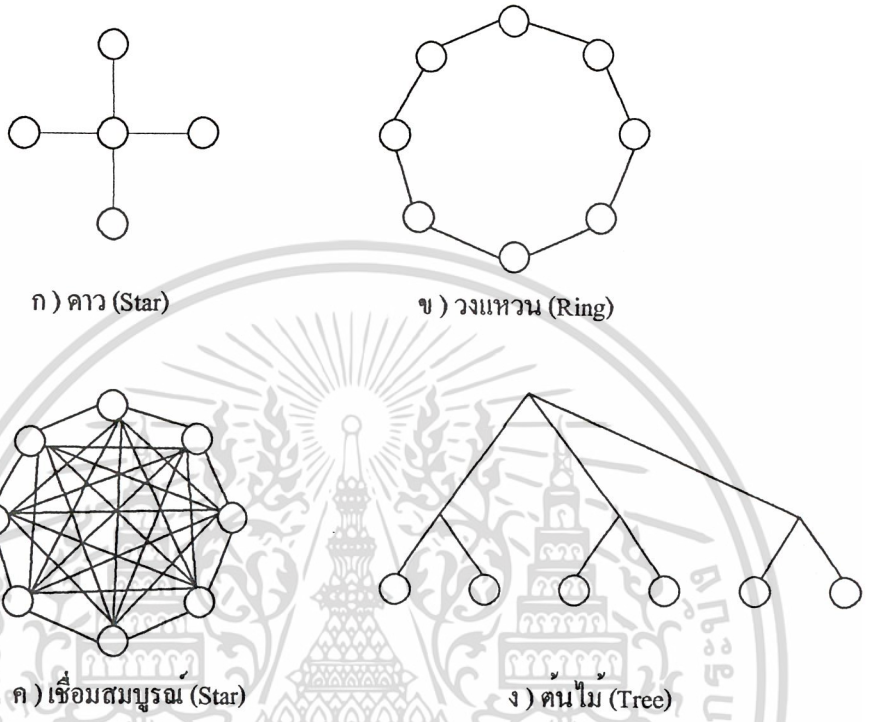
รูป 2.10 แสดงเครือข่ายย่อยส่วนของการสื่อสาร

สายส่งจะเป็นช่องการสื่อสาร (Communication Channels) หรือตัวกลางที่ใช้ในการส่งข้อมูล สายส่งอาจจะเป็นสายเคเบิลโทรศัพท์ โทรเลข วิทยุ หรือดาวเทียม ซึ่งอัตราการส่งข้อมูลมีตั้งแต่ 50 bps จนถึงระดับ 10^{14} bps ปกติแล้วมักจะมีการใช้สายส่งมากกว่าหนึ่งชนิดสำหรับโหนดที่มีติดกันเพื่อเพิ่มความเชื่อถือได้ของระบบ สำหรับตัวแปลงสัญญาณนั้นจะทำหน้าที่ ในการเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณที่เหมาะสมสำหรับการส่งข้อมูลผ่านสายส่ง เช่น เปลี่ยนสัญญาณดิจิตอล เป็นสัญญาณแสงเมื่อส่งข้อมูลผ่านเส้นใยนำแสง ตัวอย่างของตัวแปลงสัญญาณ เช่น โมเด็ม (Modem) ซึ่งใช้ในการเปลี่ยนสัญญาณดิจิตอลให้เป็นสัญญาณอนาลอก (Analog) เพื่อส่งข้อมูลผ่านวงจรโทรศัพท์ ส่วนโหนดเครือข่ายนั้นจะทำหน้าที่เป็นทั้งตัวเชื่อมประสานกับเครือข่ายย่อยส่วนของผู้ใช้และทำหน้าที่ส่งข้อมูลผ่านเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากต้นทางไปยังปลายทางได้อย่างถูกต้องรวดเร็วและมีประสิทธิภาพ ซึ่งบางทีจะเรียกโหนดเครือข่ายนี้ว่า Router โดยทั่วไปจะแบ่งชนิดเครือข่ายย่อยส่วนของการสื่อสารออกตามลักษณะการส่งข้อมูลระหว่างโหนดเครือข่ายโดยผ่านสายส่ง ออกเป็น 2 ชนิด คือ

1. แบบจุดต่อจุด (Point – to – Point channels)
2. แบบแพร่กระจายข้อมูล (Broadcast channels)



รูปที่ 2.11 แสดงรูปแบบของเครือข่ายแบบจุดต่อจุด

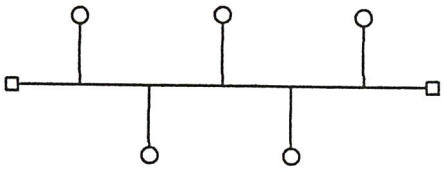
จากรูปที่ 2.11 แสดงรูปแบบของเครือข่ายแบบจุดต่อจุด ซึ่งจะเห็นว่าแต่ละสายส่งจะถูกใช้เชื่อมโยงเพื่อส่งข้อมูลระหว่างโหนด 2 โหนด และหากโหนดคู่ใดไม่มีสายเชื่อมโยงจะต้องช่วยส่งข้อมูลผ่านโหนดอื่น สำหรับเทคนิคในการส่งข้อมูลผ่านโหนดต่าง ๆ ของเครือข่ายแบบนี้ จะทำในลักษณะต่าง ๆ ได้แก่

- Circuit Switching เช่น ระบบโทรศัพท์ปัจจุบัน ซึ่งเมื่อมีการขอสร้างการติดต่อระหว่างต้นทางและปลายทาง เครือข่ายจะหาเส้นทางผ่านโหนดต่าง ๆ แล้วเชื่อมต่อสวิตช์ภายในโหนดเหล่านั้น เพื่อสร้างช่องสัญญาณให้ข้อมูลผ่านจากต้นทางถึงปลายทาง

- Packet Switching ซึ่งจะทำงานในลักษณะของการเก็บรอและส่งออก (Store and Forward) คือ เมื่อโหนดใดได้รับข้อมูลซึ่งเป็นลักษณะบล็อกเล็ก ๆ ที่เรียกว่า Packet เข้ามาแล้ว โหนดนั้นจะพิจารณาเลือกสายส่งเพื่อส่งข้อมูลออกให้ถึงปลายทางได้เร็วที่สุด แล้วจะนำไปเก็บรอไว้จนกว่าสายส่งนั้นว่างจึงส่งข้อมูลออก ไปยังโหนดอื่นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเครือข่ายย่อยส่วนของการสื่อสารแบบกระจายนั้น เป็นลักษณะที่โหนดเครือข่ายทั้งหมดจะใช้สายส่งหรือช่องสัญญาณหนึ่งร่วมกัน ดังแสดงในรูปที่ 2.12



รูป 2.12 แสดงเครือข่ายแบบบัส

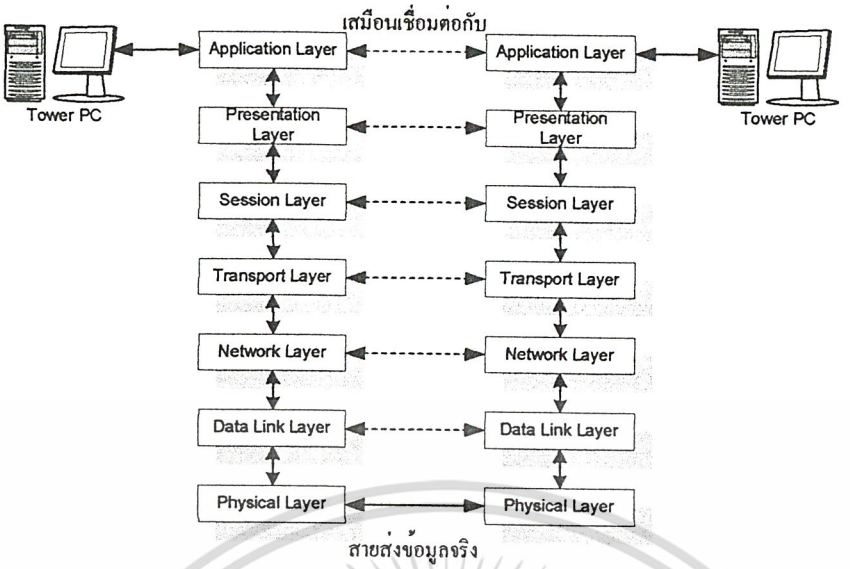
เมื่อโหนดใดส่งข้อมูลลงในช่องสัญญาณ ข้อมูลจะถูกแพร่ออกไป และถูกพบเห็นโดย ทุก ๆ โหนด ดังนั้นในกรณีที่ต้องการส่งข้อมูลให้แก่โหนดหนึ่งจะต้องใส่แอดเดรส (Address) ของโหนดนั้นลงไปในข้อมูลด้วย นอกจากนี้ในระบบแพร่กระจายข้อมูลนี้ยังสามารถส่งข้อมูลให้แก่ทุกโหนดพร้อมกันหรือกลุ่มของโหนดบางโหนดก็ได้ เนื่องจาระบบแพร่กระจายข้อมูลนี้มีการใช้ช่องสัญญาณร่วมกันระหว่างหลาย ๆ โหนด ดังนั้นจึงต้องมีวิธีควบคุมการใช้ช่องสัญญาณ เพื่อป้องกันไม่ให้โหนดต่าง ๆ ส่งข้อมูลออกมารชนกันเอง สำหรับวิธีการควบคุมการแบ่งใช้ช่องสัญญาณนี้ได้แก่ การจัดสรรช่องสัญญาณ (Allocation)

2.3.3 OSI Model : มาตรฐานอ้างอิงในการสื่อสารข้อมูล

เมื่อคอมพิวเตอร์ของเรามีการรับส่งข้อมูลกับคอมพิวเตอร์เครื่องอื่น ๆ การเชื่อมต่อระหว่างคอมพิวเตอร์หลาย ๆ เครื่องเข้าด้วยกันเป็นระบบเครือข่ายก็เกิดขึ้น อย่างไรก็ตามการเชื่อมต่อคอมพิวเตอร์คนละระบบหรือคนละยี่ห้อเป็นสิ่งที่ได้จากในยุคแรก ๆ ของการสื่อสารข้อมูล เนื่องจากขนาดมาตรฐานส่วนกลางที่จำเป็นต้องใช้ในการรับส่งข้อมูล ส่วนมากแต่ละบริษัทผู้จัดทำก็จะมีมาตรฐานของตนเองซึ่งเข้ากันไม่ได้กับรายอื่น ๆ ทำให้ผู้ใช้ต้องผูกติดกับผู้ผลิตแต่ละราย และเป็นขีดจำกัดในการเชื่อมต่อคอมพิวเตอร์คนละชนิดไม่ให้รับส่งข้อมูลกัน ระบบคอมพิวเตอร์ในยุคนั้นจึงถือว่าเป็นระบบปิด (Closed System) นั่นเอง

ปัญหานี้ทำให้หน่วยงานกำหนดมาตรฐานสากล คือ International Standards Organization หรือ ISO จัดการกำหนดโครงสร้างทั้งหมดที่จำเป็นต้องใช้ในการสื่อสารข้อมูลจากคอมพิวเตอร์ระบบหนึ่งไปยังคอมพิวเตอร์อีกระบบหนึ่งขึ้น จุดมุ่งหมายก็เพื่อเปิดช่องทางให้ข้อมูลที่เก็บอยู่ในระบบคอมพิวเตอร์หนึ่ง ๆ รับส่งไปยังคอมพิวเตอร์ที่เป็นระบบเดียวกันหรือต่างระบบได้อย่างอิสระ โดยไม่ขึ้นกับผู้ผลิตอย่างที่เป็นอยู่ในอดีตซึ่งเป็นการทำงานที่เรียกว่า ระบบเปิด (Open Systme) และเรียกโครงสร้างของมาตรฐานการรับส่งข้อมูลนี้ว่า Open System Interconnection หรือ ISO ซึ่งจัดทำขึ้นราวกลางปี ค.ศ. 1970 และใช้อ้างอิงกันมาจนถึงในยุคปัจจุบัน

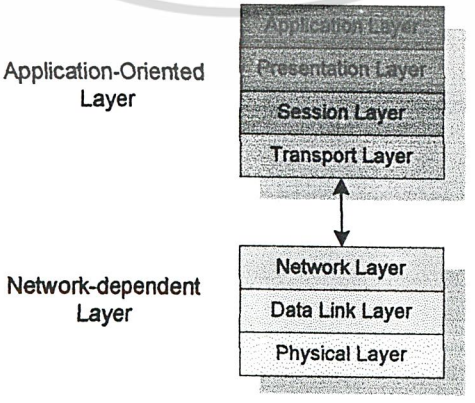
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.13 แสดงสื่อสารข้อมูลจากระบบคอมพิวเตอร์หนึ่งไปยังอีกระบบหนึ่ง

OSI กำหนดให้การสื่อสารข้อมูลจากระบบคอมพิวเตอร์หนึ่งไปยังอีกระบบหนึ่งแบ่งออกเป็น 7 ขั้นตอนย่อย ๆ ซึ่งคอมพิวเตอร์ทั้งสองระบบจะมีขั้นตอนทั้ง 7 นี้เหมือนกันทั้งสองฝั่ง เราเรียกการสื่อสารข้อมูลนี้ว่า OSI 7-Layer Reference Model ดังแสดงในรูปที่ 2.13 โดยโครงสร้างการสื่อสารข้อมูลที่กำหนดขึ้นมีคุณสมบัติ คือ

แต่ละชั้น (Layer) ของการสื่อสารข้อมูล จะเสมือนเชื่อมต่ออยู่กับชั้นที่เทียบเท่ากันของคอมพิวเตอร์อีกด้านหนึ่ง ส่วนการเชื่อมต่อกันจริง ๆ จะมีเพียงชั้นที่ 1 หรือ Layer 1 ซึ่งเป็นชั้นล่างสุดเท่านั้นที่มีการรับส่งข้อมูลเกิดขึ้นผ่านสายส่งข้อมูลระหว่างคอมพิวเตอร์ทั้งสองระบบ ส่วนชั้นอื่น ๆ จะไม่ได้เชื่อมต่อกันจริง เพียงแต่ทำงานเสมือนกับว่ามีการติดต่อรับส่งข้อมูลโดยใช้กลไกที่เรียกว่า Peer-to-Peer Protocol และแต่ละชั้นที่ทำหน้าที่รับส่งข้อมูลจะมีการติดต่อรับส่งข้อมูลกับชั้นที่อยู่ติดกับตัวเองเท่านั้นโดยอาศัย Interface Protocol จะติดต่อรับส่งข้อมูลข้ามกระโดดไปชั้นอื่น ๆ ในคอมพิวเตอร์ของตัวเองไม่ได้ โดยผู้ใช้ หรือ User จะติดต่อรับส่งข้อมูลผ่านทางชั้นที่ 7 ซึ่งอยู่ด้านบนสุดของ OSI 7-Layer Model เท่านั้น



รูป 2.14 แสดงการแบ่ง OSI 7-Layer Model ในทางปฏิบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางปฏิบัตินั้น OSI 7-Layer Model จะแบ่งออกเป็น 2 กลุ่มใหญ่ ๆ ดังรูปที่ 2.14 คือ กลุ่มแรกได้แก่ 4 ชั้นด้านบนคือชั้นที่ 7, 6, 5 และ 4 ทำหน้าที่เชื่อมต่อรับส่งข้อมูลระหว่างผู้ใช้กับซอฟต์แวร์โปรแกรมประยุกต์ ให้รับส่งข้อมูลกับฮาร์ดแวร์ที่อยู่ชั้นล่างได้ถูกต้อง เรียกว่า Application Oriented Layers ซึ่งจะเกี่ยวข้องกับซอฟต์แวร์เป็นหลัก โดย 4 ชั้นด้านบนนี้มักจะเป็นซอฟต์แวร์ของบริษัทหนึ่งรวมอยู่อย่างเบ็ดเสร็จในโปรแกรมเดียว จะแยกออกจากกันเป็นชั้น ๆ เพื่อใช้โปรแกรมของบริษัทอื่นได้ลำบาก หรือในบางกรณีก็อาจทำไม่ได้เลย ส่วนกลุ่มที่สองจะเป็นชั้นล่าง ได้แก่ ชั้นที่ 3, 2 และ 1 ทำหน้าที่เกี่ยวกับการรับส่งข้อมูลผ่านสายส่ง และควบคุมการรับส่งข้อมูล ตรวจสอบข้อผิดพลาด รวมทั้งเลือกเส้นทางที่ใช้ในการรับส่งข้อมูลที่จะเกี่ยวข้องกับฮาร์ดแวร์เป็นหลักเรียกว่า Network Dependent Layers สามารถแยกแต่ละชั้นออกจากกันได้โดยง่าย และใช้ผลิตภัณฑ์ของต่างบริษัทกันในแต่ละชั้นได้อย่างไม่มีปัญหา OSI 7-Layer Model ที่แบ่งการรับส่งข้อมูลระหว่างคอมพิวเตอร์สองระบบออกเป็น 7 ชั้นนั้น แต่ละชั้นมีชื่อเรียกและทำหน้าที่การทำงานดังนี้ คือ

ชั้นที่ 7 Application Layer

เป็นชั้นที่อยู่บนสุดของขบวนการส่งข้อมูลทำหน้าที่เชื่อมต่อผู้ใช้เข้ากับระบบคอมพิวเตอร์โดยรับคำสั่งต่าง ๆ จากผู้ใช้ให้ระบบคอมพิวเตอร์แปลความหมาย และทำงานตามคำสั่งที่ได้รับในระดับโปรแกรมประยุกต์ เช่น แปลความหมายของการกดปุ่มบนเมาส์ให้ เป็นต้น ซึ่งการแปลความหมายของการกดปุ่มบนเมาส์ให้เป็นคำสั่งในการคัดลอกข้อมูล หรือดึงข้อมูลมาแสดงผลบนจอภาพ เป็นต้น ซึ่งการแปลออกมาถูกต้องตามกฎที่ใช้ในระบบปฏิบัติการของคอมพิวเตอร์นั้น ๆ ตัวอย่างเช่น ถ้ามีการคัดลอกข้อมูลเกิดขึ้นในระบบ คำสั่งที่ใช้จะต้องสร้างไฟล์ได้ถูกต้อง มีชื่อไฟล์ยาวไม่เกินจำนวนที่ระบบปฏิบัติการใช้อยู่ และชื่อไฟล์ต้องประกอบด้วยตัวอักษรตามที่กำหนด ไม่มีตัวอักษรห้ามมาตั้งเป็นชื่อไฟล์ เป็นต้น สิ่งต่าง ๆ เหล่านี้จะเกิดขึ้นในชั้นที่ 7 ของการสื่อสารข้อมูลรวมทั้งฟังก์ชั่นในการเชื่อมต่อรับส่งข้อมูลระหว่างชั้นที่ 7 กับชั้นที่ 6 ด้วย

ชั้นที่ 6 Presentation Layer

เป็นชั้นที่ทำหน้าที่ตกลงกับคอมพิวเตอร์อีกด้านหนึ่งว่า การรับส่งข้อมูลในระดับโปรแกรมประยุกต์จะมีขั้นตอนและข้อบังคับอย่างไร ข้อมูลที่ทำการรับส่งกันในชั้นที่ 6 นี้จะอยู่ในรูปแบบของข้อมูลชั้นสูง ซึ่งอยู่ในรูปแบบของคำสั่งที่มีกฎ (Syntax) บังคับอย่างแน่นอน เช่น ในการคัดลอกข้อมูลก็จะมีขั้นตอนย่อยประกอบกัน คือ สร้างไฟล์ที่กำหนดขึ้นมาเสียก่อน จากนั้นจึงเปิดไฟล์แล้วทำการรับข้อมูลจากปลายทางมาเก็บลงในไฟล์ที่สร้างขึ้นใหม่นี้ โดยเนื้อหาของข้อมูลที่ทำการรับส่งระหว่างกัน ก็คือคำสั่งของขั้นตอนย่อย ๆ ข้างต้นนั่นเอง คำสั่งเหล่านี้จะต้องหมายถึงว่าจะให้ทำอะไรบ้างและถูกต้องตามกฎด้วย นอกจากนี้ในชั้นที่ 6 ยังทำหน้าที่แปลความหมายของคำสั่งที่ได้รับจากชั้นที่ 7 ให้เป็นคำสั่งระดับปฏิบัติการส่งในชั้นที่ 5 ต่อไปอีกด้วย

ชั้นที่ 5 Session Layer

ทำหน้าที่ควบคุมจังหวะในการรับส่งข้อมูลของคอมพิวเตอร์ทั้งสองด้านที่รับส่งแลกเปลี่ยนข้อมูลกันให้มีความสอดคล้องกัน (Synchronization) และกำหนดวิธีที่ใช้รับส่งข้อมูล เช่น อาจจะเป็นลักษณะ Half Duplex หรือ Full Duplex ซึ่งในชั้นที่ 5 นี้จะเป็นชั้นที่ใช้ควบคุมการรับส่งข้อมูลในลักษณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังกล่าว ข้อมูลที่รับส่งกันในชั้นที่ 5 นี้ จะอยู่ในรูปของ Dialog หรือประโยคของข้อมูลที่สนทนาโต้ตอบกันระหว่างด้านรับและด้านที่ส่งข้อมูล ไม่ได้มองเป็นคำสั่งอย่างในชั้นที่ 6 เช่น เมื่อผู้รับได้รับข้อมูลส่วนแรกของผู้ส่ง ก็จะโต้ตอบกลับไปให้ผู้ส่งรู้ว่าได้รับข้อมูลส่วนแรกเรียบร้อยแล้ว และพร้อมที่จะรับข้อมูลส่วนที่สองต่อไป คล้ายกับการสนทนาโต้ตอบกันระหว่างผู้รับและผู้ส่งนั่นเอง

ชั้นที่ 7 Transport Layer

ทำหน้าที่เชื่อมต่อการรับส่งข้อมูลระดับสูงของชั้นที่ 5 มาเป็นข้อมูลที่รับส่งในระดับฮาร์ดแวร์ เช่น แปลงค่าหรือชื่อของคอมพิวเตอร์ในเครือข่ายให้เป็น Network Address พร้อมทั้งเป็นชั้นที่ควบคุมการรับส่งข้อมูลจากปลายด้านส่งถึงปลายด้านรับข้อมูล ให้ข้อมูลมีการไหลลื่นตลอดเส้นทางตามจังหวะที่ควบคุมจากชั้นที่ 5 โดยในชั้นที่ 4 นี้ จะเป็นรอยต่อระหว่างการรับส่งข้อมูลของซอฟต์แวร์กับฮาร์ดแวร์ การรับส่งข้อมูลของระดับสูงจะถูกแยกจากฮาร์ดแวร์ที่ใช้รับส่งข้อมูลที่ชั้นที่ 4 นี้ และจะไม่มีส่วนใดผูกติดกับฮาร์ดแวร์ที่ใช้รับส่งข้อมูลในระดับล่าง ดังนั้นฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ควบคุมการรับส่งข้อมูลในระดับล่างลงไปจากชั้นที่ 4 สามารถสับเปลี่ยนและใช้ข้ามไปมากับซอฟต์แวร์รับส่งข้อมูลในระดับสูงที่อยู่ข้างบนได้ง่าย หน้าที่อีกประการหนึ่งของชั้นที่ 4 คือ การควบคุมคุณภาพของการรับส่งข้อมูลให้มีมาตรฐานในระดับที่ตกลงกันของทั้งสองฝ่าย และการตัดข้อมูลออกเป็นส่วนย่อย ๆ ให้เหมาะกับลักษณะการทำงานของฮาร์ดแวร์ที่ใช้ในเน็ตเวิร์ค เช่น หากชั้นที่ 5 ต้องการส่งข้อมูลที่มีความยาวมากเกินกว่าที่ระบบเครือข่ายจะส่งได้ ชั้นที่ 4 ก็จะทำหน้าที่ตัดข้อมูลออกเป็นส่วนย่อย ๆ แล้วส่งไปให้ผู้รับ ข้อมูลที่ได้รับปลายทางก็จะถูกนำมาต่อกันที่ชั้นที่ 4 ของด้านผู้รับและผู้ส่งให้ชั้นที่ 5 ต่อไป

ชั้นที่ 3 Network Layer

ทำหน้าที่เชื่อมต่อคอมพิวเตอร์ของด้านรับและด้านส่งเข้าหากันผ่านระบบเครือข่าย พร้อมทั้งเลือกหรือกำหนดเส้นทางที่จะใช้ในการรับส่งข้อมูลระหว่างกัน และส่งผ่านข้อมูลที่ได้รับไปยังอุปกรณ์ในเครือข่ายต่าง ๆ จนกระทั่งถึงปลายทาง ในชั้นที่ 3 นี้ ข้อมูลที่รับส่งกันจะอยู่ในรูปแบบของกลุ่มข้อมูลที่เรียกว่า Packet หรือ Frame ข้อมูลที่ชั้นที่ 4, 5, 6 และ 7 มองเห็นเป็นคำสั่งและ Dialog ต่าง ๆ นั้น จะถูกแปลงและผนึกรวมอยู่ในรูปของ Packet หรือ Frame ที่มีเพียงแอดเดรสของผู้รับ, ผู้ส่ง, ลำดับการรับส่งและส่วนของข้อมูลเท่านั้น ตัวเนื้อหาของข้อมูลจะไม่มีผลใด ๆ ในการรับส่งข้อมูลเข ไม่ว่าข้อมูลในระดับสูงจะเป็น วีดีโอ, ภาพ, เสียง หรือข้อมูลอื่นใดก็ตาม แต่ในชั้นที่ 3 จะมองข้อมูลทั้งหมดเป็น Packet หรือ Frame เท่านั้น หน้าที่อีกประการหนึ่งของชั้นที่ 3 นี้คือ การทำ Call Setup หรือเรียกติดต่อคอมพิวเตอร์ปลายทางก่อนการรับส่งข้อมูล และการทำ Call Clearing หรือยกเลิกการติดต่อเมื่อการรับส่งข้อมูลจบลงแล้ว ในกรณีที่การรับส่งข้อมูลนั้นต้องมีการติดต่อกันก่อน

ชั้นที่ 2 Datalink Layer

เป็นชั้นที่ทำหน้าที่เชื่อมต่อการรับส่งข้อมูลในระดับฮาร์ดแวร์ โดยเมื่อมีการสั่งให้รับข้อมูลจากในชั้นที่ 3 ลงมา ชั้นที่ 2 จะทำหน้าที่แปลคำสั่งนั้นให้เป็นคำสั่งควบคุมฮาร์ดแวร์ที่ใช้รับส่งข้อมูล ทำการตรวจสอบข้อผิดพลาดในการรับส่งข้อมูลของระดับฮาร์ดแวร์ และแก้ไขข้อผิดพลาดที่ตรวจพบนั้น ข้อมูลที่อยู่ในชั้นที่ 2 นี้ จะอยู่ในรูปของ Frame คือกลุ่มของข้อมูลที่มีรูปร่างตามข้อบังคับของฮาร์ดแวร์

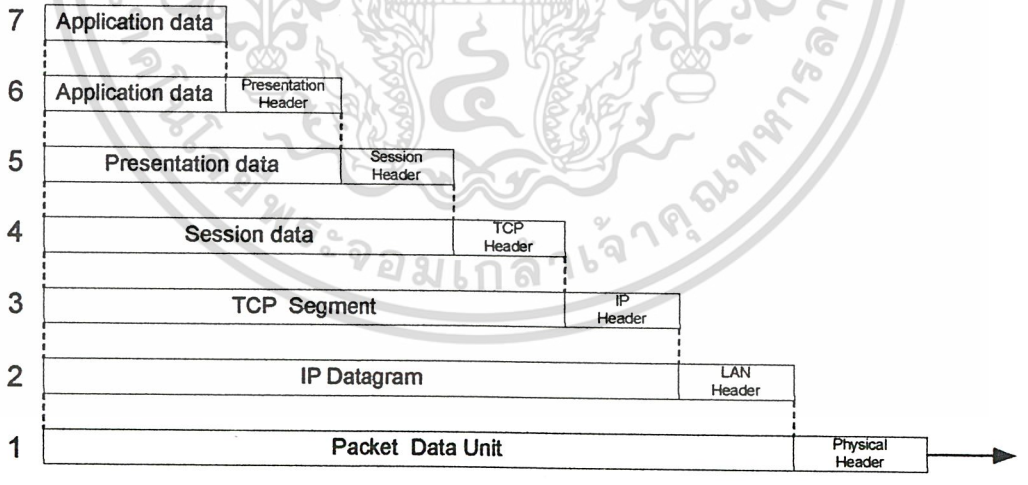
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ใช้ในการรับส่งข้อมูล เช่น ถ้าฮาร์ดแวร์ที่ใช้เป็น Ethernet LAN ข้อมูลก็จะมีรูปร่างของ Frame ตามที่ระบุไว้ในมาตรฐานของ Ethernet หากว่าฮาร์ดแวร์ที่ใช้รับส่งข้อมูลเป็นชนิดอื่น เช่น Token Ring LAN

ชั้นที่ 1 Physical Layer

เป็นชั้นล่างสุดของขั้นตอนในการรับส่งข้อมูลของ OSI 7-Layer Reference Model ซึ่งเป็นชั้นเดียวที่มีการเชื่อมต่อกันทางกายภาพระหว่างคอมพิวเตอร์สองระบบที่ทำการรับส่งข้อมูลกันในชั้นที่ 1 นี้จะกำหนดคุณสมบัติทางกายภาพของฮาร์ดแวร์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์ทั้งสองระบบ เช่น สายที่ใช้รับส่งข้อมูลจะเป็นแบบไหน, ความเร็วในการรับส่งข้อมูลเป็นเท่าไร เป็นต้น ข้อมูลในชั้นที่ 1 นี้จะมองเห็นเป็นการรับส่งข้อมูลทีละบิตเรียงกันต่อกันไป โดยไม่มีการพิจารณาเรื่องความหมายของข้อมูลเลย การรับส่งจะส่งข้อมูล "0" หรือ "1" ไปให้คอมพิวเตอร์ด้านรับส่งข้อมูลในระดับฮาร์ดแวร์เท่านั้น หากการรับส่งข้อมูลมีปัญหาเนื่องจากฮาร์ดแวร์จะเป็นหน้าที่ของชั้นที่ 1 นี้ในการตรวจสอบและแจ้งข้อผิดพลาดนั้นให้ชั้นอื่นๆ ที่อยู่เหนือขึ้นไปทราบ

ในการรับส่งข้อมูลใน OSI 7-Layer Model นั้น ข้อมูลจากชั้นบนสุด คือชั้นที่ 7 เมื่อถูกส่งลงไปในชั้นถัดลงไป ข้อมูลเดิมก็จะถูกผนึกรวมกับข้อมูลที่ใช้ควบคุมของแต่ละชั้นซ้อน ๆ กันเป็นลำดับเท่ากับจำนวนชั้นที่ผ่านลงไป ตัวอย่างเช่น Application Data เมื่อถูกส่งลงไปยังชั้นถัดไปก็จะถูกผนึกด้วย Application Header และทั้ง Application Header และ Application Data จะรวมกันเป็นข้อมูลของชั้นที่อยู่ถัดลงไปอีกเรื่อยๆ เป็นเช่นนี้จนกระทั่งถึงชั้นล่างสุดซึ่งเป็น Physical Layer ซึ่งเมื่อข้อมูลถูกส่งไปถึงปลายทาง ข้อมูลที่ได้รับจะถูกแยก Header ที่เพิ่มเข้ามานี้ออกทีละชั้น ซึ่งเป็นขบวนการย้อนกลับด้านส่ง จนกระทั่งถึงชั้นบนสุด จึงจะเป็นข้อมูล Application Data ให้ผู้รับตามต้องการ ดังแสดงในรูปที่ 2.15



รูป 2.15 แสดงรูปแบบของข้อมูลแต่ละเลเยอร์

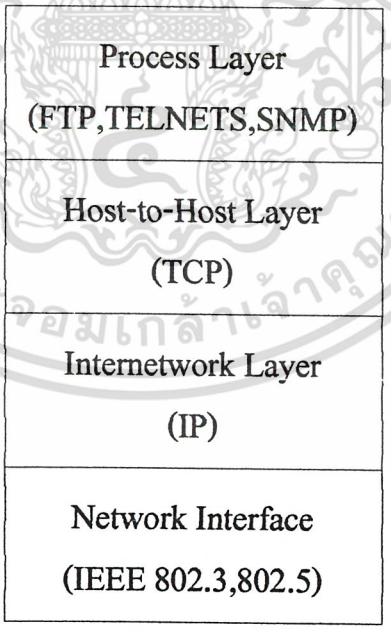
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 โปรโตคอล TCP/IP

โปรโตคอล TCP/IP เป็นโปรโตคอลที่สำคัญที่มีการใช้กันอย่างแพร่หลายตามการขยายตัวของอินเตอร์เน็ตและอินทราเน็ต โดยความจริงแล้ว TCP/IP เป็นกลุ่มของโปรโตคอลหลายตัวที่ประกอบกันเป็นชุดให้ใช้งานโดยมีชื่อเรียกว่า Transmission Control Protocol / Internet Protocol ตัวอย่างของโปรโตคอลในกลุ่มของ TCP/IP ที่เราพบและใช้งานบ่อย ๆ โดยผ่านทางแอพพลิเคชั่นต่าง ๆ เช่น Internet Protocol (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), Simple Mail Transfer Protocol (SMTP) และ Domain Name System (DNS) เป็นต้น โดยโปรโตคอลที่มีบทบาทสำคัญในการทำงานในเครือข่ายอินเตอร์เน็ต คือ IP (Internet Protocol) เนื่องจากเมื่อโปรโตคอลอื่น ๆ ต้องการส่งข้อมูลผ่านเครือข่ายอินเตอร์เน็ตนั้น จำเป็นต้องอาศัยการผนึกข้อมูล (Encapsulation) ไปกับโปรโตคอล IP ที่มีกลไกการระบุเส้นทาง (Route Service) ผ่าน Gateway เพื่อนำข้อมูลไปยังเครือข่ายหรือเครื่องปลายทางที่ถูกต้อง โดยกลไกการระบุเส้นทางจะทำงานที่โปรโตคอล IP เท่านั้น หรืออาจเรียกว่า IP เป็นโปรโตคอลที่มีความสามารถในการระบุเส้นทางของการส่งผ่านข้อมูลได้ (Routable)

2.3.4.1 สถาปัตยกรรมของ TCP/IP (TCP/IP Reference Model)

TCP/IP จะมีการแบ่งจำนวนขั้นตอนที่ใช้รับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองระบบออกเป็น 4 ชั้นเท่านั้น หรือเรียกว่าเป็น TCP/IP Stack โดยมีชื่อเรียกแตกต่างกันดังแสดงได้ดังรูปที่ 2.16



รูป 2.16 แสดงชั้นของโปรโตคอลแบบ TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Process Layer** จะเป็น Application Protocol ที่ทำหน้าที่เชื่อมต่อกับผู้ใช้และให้บริการต่าง ๆ เช่น FTP, Telnet เป็นต้น

- **Host-to-Host Layer** จะเป็น TCP หรือ UDP ที่ทำหน้าที่คล้ายกับชั้นที่ 4 ของ OSI 7-Layer Model คือ ควบคุมการรับส่งข้อมูลจากปลายด้านส่งถึงปลายด้านรับข้อมูล และตัดข้อมูลออกเป็นส่วนย่อยให้เหมาะกับเครือข่ายที่ใช้รับส่งข้อมูล รวมทั้งประกอบข้อมูลส่วนย่อย ๆ นี้เข้าด้วยกันเมื่อถึงปลายทาง

- **Internetwork Layer** ได้แก่ ส่วนของโปรโตคอล IP ซึ่งทำหน้าที่คล้ายกับชั้นที่ 3 ของ OSI 7-Layer Model คือเชื่อมต่อคอมพิวเตอร์เข้าระบบเครือข่ายที่อยู่ชั้นล่างลงไป และทำหน้าที่เลือกเส้นทางการรับส่งข้อมูลในลักษณะเครือข่ายต่าง ๆ จนไปถึงผู้รับข้อมูล ในชั้นนี้จะจัดการกับกลุ่มข้อมูลในลักษณะที่เรียกว่า Frame ในรูปแบบของ TCP/IP ที่เรารู้จักกันนั่นเอง

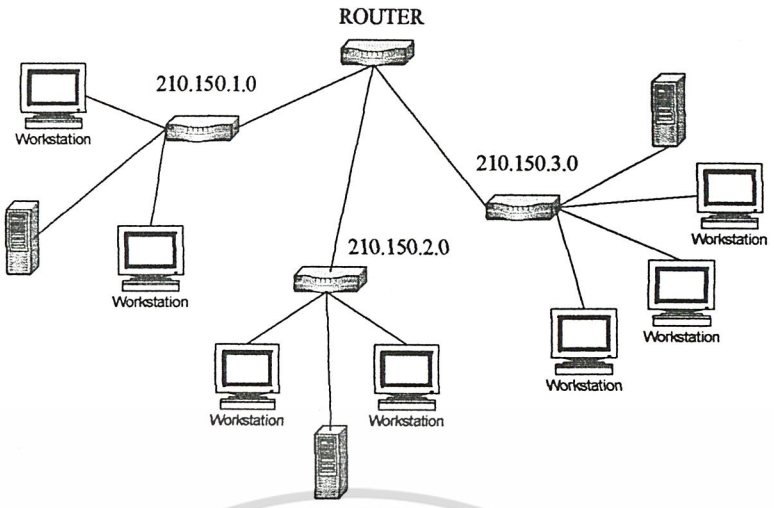
- **Network Interface** คือ ชั้นที่ควบคุมฮาร์ดแวร์รับส่งข้อมูลผ่านเครือข่าย ซึ่งเทียบได้กับชั้นที่ 1 และ 2 ของ OSI 7-Layer Model ในชั้นนี้จะทำหน้าที่เชื่อมต่อกับฮาร์ดแวร์ และควบคุมการรับส่งข้อมูลในระดับฮาร์ดแวร์ของเครือข่าย ซึ่งที่ใช้กันอยู่จะเป็นตามมาตรฐานของ IEEE

2.3.4.2 IP ADDRESS

IP Address เป็นหมายเลขที่ถูกกำหนดขึ้นมาเป็นหมายเลขอ้างอิงประจำตัวของอุปกรณ์ต่าง ๆ ที่เชื่อมต่ออยู่ในเครือข่ายอินเตอร์เน็ต โดยอุปกรณ์แต่ละชิ้นจะมี IP Address ไม่ซ้ำกัน แต่อย่างไรก็ตามหมายเลขนี้จะไม่ถูกผูกติดกับตัวอุปกรณ์ สามารถกำหนดหรือแก้ไขเปลี่ยนแปลงได้ ทั้งนี้เนื่องจากเป็นการกำหนดด้วยซอฟต์แวร์ แตกต่างกับหมายเลข MAC Address (Media Access Control Address) ซึ่งเป็นหมายเลขประจำตัวของอุปกรณ์ที่ต่ออยู่ในเครือข่าย ค่า MAC Address จะถูกกำหนดมาจากบริษัทผู้ผลิตอุปกรณ์ตั้งแต่ต้นโดยจะมีค่าไม่ซ้ำกันและไม่สามารถแก้ไขได้ ค่า MAC Address เป็นการระบุค่าอ้างอิงของอุปกรณ์ในระดับล่างสุด (Physical Layer) ของกลไกการรับส่งข้อมูลผ่านเครือข่าย

การทำงานของ Internet Protocol (IP) จำเป็นต้องอาศัย IP Address ในการระบุและอ้างถึงอุปกรณ์ต่าง ๆ ที่ต่ออยู่ในเครือข่าย หมายเลข IP Address จะเป็นค่าตัวเลขขนาด 32 บิต ถูกแบ่งออกเป็นส่วนละ 8 บิต รวมเป็น 4 ส่วนและคั่นด้วยเครื่องหมายจุด (.) ดังนั้นตัวเลขแต่ละส่วนจะมีได้ตั้งแต่ 0 ถึง 255 (2^8) โดยค่าของ IP Address จะถูกกำหนดออกเป็น 2 ความหมาย คือ ค่าของหมายเลขอุปกรณ์เครือข่าย (Host Address) และค่าของหมายเลขเครือข่าย (Network Address) ตัวอย่างเช่น มีเครื่องเว็บเซิร์ฟเวอร์เชื่อมต่ออยู่ในเครือข่าย 2 เครื่อง โดยแต่ละเครื่องมี IP Address ประจำตัวคือ 205.144.78.2 และ 205.144.78.3 จะเห็นได้ว่าเครื่องทั้งสองมีค่า 205.144.78 เหมือนกัน คือมีค่าของหมายเลขเครือข่ายเดียวกัน บนสายสัญญาณที่เชื่อมโยงเส้นเดียวกันแต่มีหมายเลขประจำเครื่องที่แตกต่างกัน คือ 2 และ 3 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

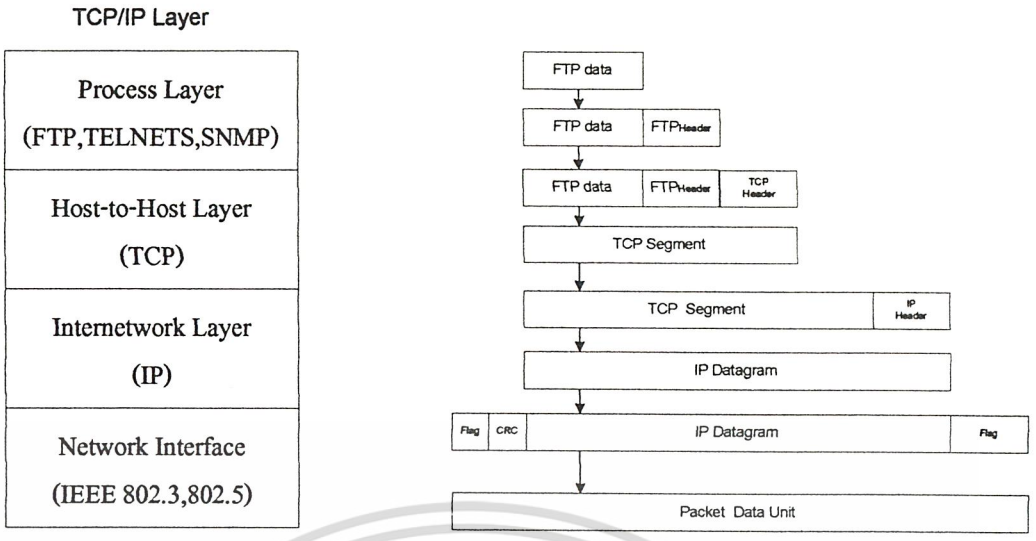


รูป 2.17 แสดงการเชื่อมต่อเครือข่ายด้วยเราท์เตอร์

ในการรับส่งข้อมูลผ่านเครือข่ายอินเตอร์เน็ตนั้น ตัวข้อมูลที่จะส่งจะถูกทำให้มีขนาดเล็กลงโดยแบ่งออกเป็นส่วนย่อย ๆ เรียกว่า Data Packet หรือ Datagram ซึ่งการแบ่งข้อมูลเป็นส่วนย่อยนี้จะช่วยให้เครือข่ายสามารถส่งข้อมูลได้อย่างราบรื่นไม่ติดขัด อีกทั้งยังทำให้สามารถทำการตรวจทานความถูกต้องของข้อมูลที่ปลายทางและแก้ไขข้อผิดพลาดได้อย่างมีประสิทธิภาพด้วย และที่สำคัญช่องสัญญาณที่ใช้ในการเชื่อมต่อในระบบเครือข่ายมีจำกัดและเป็นสิ่งที่ต้องใช้ร่วมกัน เมื่อมีอุปกรณ์ใดส่งข้อมูล อุปกรณ์อื่น ๆ ก็ต้องรอให้การส่งข้อมูลนั้นเสร็จก่อน ดังนั้นการแบ่งข้อมูลเป็นส่วนย่อยจะเป็นการแบ่งเวลาให้กับผู้ใช้งานรายอื่น ๆ ได้

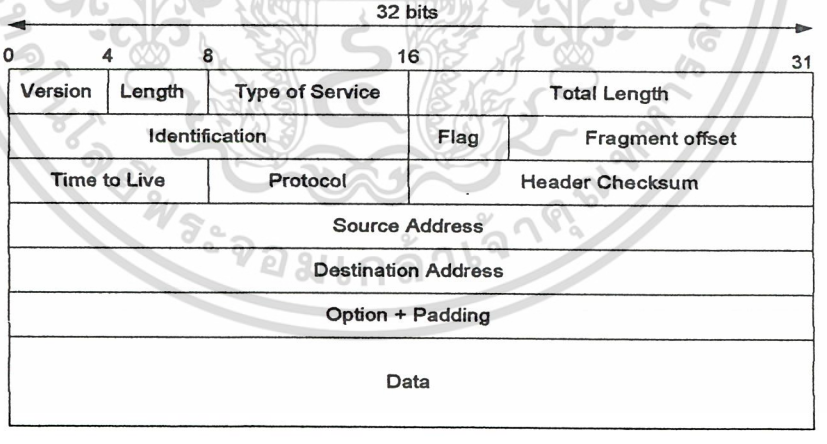
โดยตัวของข้อมูลที่ถูกแยกออกเป็น Data Packet หรือ Datagram จะมีลักษณะเป็นข้อมูลแบบต่อเนื่อง (Stream Byte) คือมีการลำดับก่อนหลังของข้อมูล เพื่อให้ประกอบข้อมูลย่อยคืนสู่สภาพเดิมได้อย่างถูกต้อง และมีรูปแบบที่แน่นอน คือ Datagram จะประกอบด้วยส่วนของ Header และส่วนของตัวข้อมูล (Body) โดยในส่วนหัวของข้อมูลจะประกอบด้วยข้อมูลต่าง ๆ ที่ระบุที่อยู่ปลายทางที่จะต้องส่งข้อมูลไป, หมายเลขต้นทางที่ส่งข้อมูลมา, ค่าความยาวของ Datagram นี้ และข้อมูลอื่น ๆ โดย Datagram นี้จะถูกผนึก (Encapsulation) เข้าไปกับโปรโตคอล IP เรียกร่วมกันเป็น IP Datagram นับว่าเป็นกลไกสำคัญของการส่งข้อมูลด้วย TCP/IP โดยขบวนการที่ใช้จะมีขั้นตอนคร่าว ๆ ดังรูปที่ 2.18 ซึ่งเป็นตัวอย่างของการเรียกใช้โปรแกรมแอพพลิเคชั่น FTP (File Transfer Protocol) ร่วมกับ TCP/IP ในการส่งผ่านข้อมูลไปในเครือข่ายแบบ Ethernet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดงของข้อมูลของการส่งข้อมูลด้วย TCP/IP

หลังจากขบวนการ Encapsulation นั้น เราจะพบว่าข้อมูลในรูปของ Datagram จะถูกแปลงเป็น Ethernet Frame หรือเฟรมข้อมูลในรูปแบบอื่น ๆ ตามลักษณะการเชื่อมต่อทางกายภาพ เช่น Ethernet หรือ Token-Ring เป็นต้น และเพื่อให้ข้อมูลสามารถส่งออกสู่เครือข่ายและข้ามเครือข่ายไปสู่อินเตอร์เน็ตได้ ตัวข้อมูลที่ถูกแปลงออกมาเป็น IP Datagram นี้จะประกอบด้วย ส่วน IP Header ที่มีขนาด 32 ไบต์ และส่วนเนื้อข้อมูลที่เรียกว่า Payload โดยจะมีขนาดขึ้นกับข้อมูล ทำให้ IP Datagram มีขนาดไม่แน่นอน และมีลักษณะตามรูปที่ 2.19



รูปที่ 2.20 แสดงรูปแบบของ IP Datagram

ส่วนของ IP Header มีการแบ่งย่อยเพื่อระบุพารามิเตอร์ในการใช้งานต่าง ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Version มีขนาด 4 บิต กำหนดเป็น 4 เมื่อใช้มาตรฐาน IPv4 และเป็น 6 เมื่อใช้เป็น IPv6 (มาตรฐานที่จะใช้งานในอนาคต)

- Length มีขนาด 4 บิต กำหนดความยาวของ IP Header

- Type of Service มีขนาด 8 บิต บอกให้ทราบว่าจะดำเนินการกับข้อมูลนี้อย่างไรเช่น Low delay, High Throughout เป็นต้น

- Total Length มีขนาด 16 บิต กำหนดความยาวสุทธิของ IP Datagram เป็นจำนวน ไบต์ ซึ่งมีค่าไม่เกิน 2^{16} หรือ 65,535 ไบต์

- Identification เป็นข้อมูลที่บอกให้ทราบว่า IP Datagram นี้มาจากที่ใด

- Flag และ Fragment Offset เป็นส่วนข้อมูลที่ใช้ระบุการแยกและรวมข้อมูล เพื่อให้ข้อมูลที่ถูกแยกออกเป็นข้อมูลย่อย (Fragment) สามารถกลับมารวมกันใหม่ตามลำดับได้อย่างถูกต้อง

- Time เป็นข้อมูลแสดงจำนวนเวลามากที่สุดของ IP Datagram นี้สามารถส่งผ่านเครือข่ายไปยังปลายทางได้ มีหน่วยเป็นวินาที ปกติเป็น 32 โดยเมื่อ IP Datagram ถูกส่งผ่าน Router ตัวหนึ่งก็จะถูกลดค่า Time ลงไปหนึ่ง ทำให้สามารถใช้ค่านี้นับจำนวนเครือข่ายที่ IP Datagram นี้ถูกส่งผ่านมาได้ เรียกว่า Hop Count

- Protocol เป็นการระบุโปรโตคอลที่ทำงานในชั้น (Layer) ข้างบนที่ผนึกลงมาใน IP Datagram

- Header Checksum เป็นส่วนของข้อมูลที่ใช้ตรวจสอบความถูกต้องของข้อมูลในส่วนของ IP Header เท่านั้น โดยเมื่อเกิดความผิดพลาดของข้อมูล IP Datagram นั้นจะถูกยกเลิกไปไม่นำมาใช้งาน

- Source / Destination IP Address เป็นส่วนที่เก็บหมายเลข IP Address ข้างต้นทางและปลายทางตามลำดับ

- Option มีขนาดไม่แน่นอนใช้เก็บค่าพารามิเตอร์ส่วนประกอบปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการใช้งาน

- Padding ทำหน้าที่เป็นส่วนเติมข้อมูลให้ IP Header ครบ 32 ไบต์

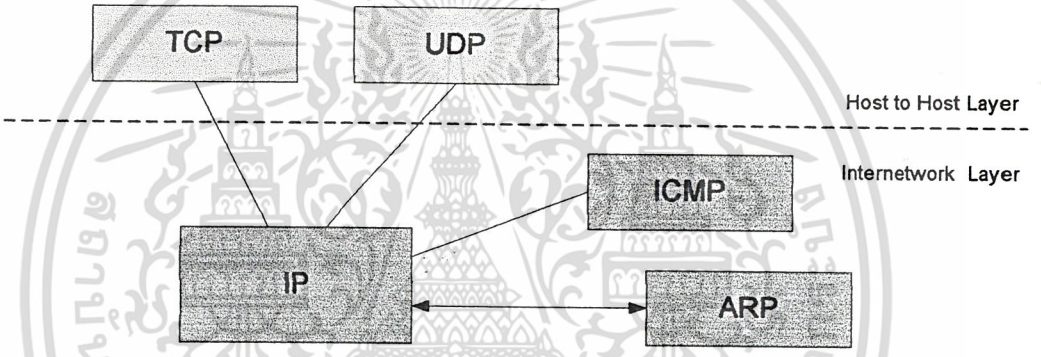
2.3.4.3 กลไกการทำงานของโปรโตคอล TCP

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ Stream Oriented Protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อย ๆ ก่อน (Datagram) แล้วจึงส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป ก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูลย่อยประกอบกลับเป็นข้อมูลทั้งหมดได้ และจะแยกข้อมูลที่ไม่ถูกต้องออก การติดต่อระหว่างกันต้องเป็นแบบ Connection-Oriented คือ ต้องกมีการสร้างช่องทางการติดต่อ (Session) ทั้ง 2 ด้านเสียก่อน แล้วจึงทำการรับส่งข้อมูลไปได้พร้อมกัน (Full Duplex) คล้ายกับการใช้โทรศัพท์

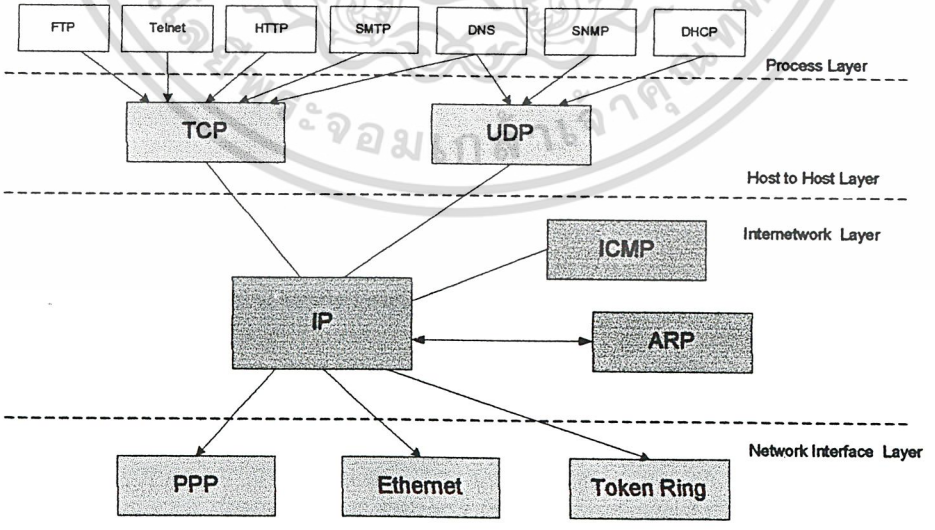
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีแอพพลิเคชั่นต้องการส่งผ่านข้อมูลจะใช้โปรโตคอลที่เหมาะสมในชั้นของ Process Layer ติดต่อไปจากนั้นจะมีการสร้างช่องส่งข้อมูลผ่านพอร์ต (Port) ที่กำหนดเพื่อส่งข้อมูลไปยังโปรโตคอล TCP ที่ทำงานอยู่ในชั้น Host-to-Host Layer ในระหว่างการรับส่งข้อมูลนี้โปรโตคอล TCP จะเพิ่มกระบวนการสอบทานข้อมูลเพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณตรวจสอบทานข้อมูล (Acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น

โดยนอกจากโปรโตคอล TCP แล้วยังมีโปรโตคอล UDP (User Datagram Protocol) ที่จะทำงานในลักษณะที่แตกต่างจากโปรโตคอล TCP คือจะทำการติดต่อสื่อสารแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (Connectionless-Oriented) และไม่มีการตรวจสอบความถูกต้องครบถ้วนของข้อมูลในการรับส่งด้วย จึงทำให้โปรโตคอล UDP ใช้ทรัพยากรของระบบน้อยกว่าโปรโตคอล TCP เนื่องจากการทำงานที่ต่างกันทำให้โปรโตคอลทั้งสองแบบมีแอพพลิเคชั่นที่เรียกใช้งานต่างกันดังรูปที่ 2.21



รูป 2.21 แสดงโปรโตคอลในชั้น Host-to-Host Layer และ Internetwork Layer



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.5 กลไกการทำงานของโปรโตคอล IP

โปรโตคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host Layer เพื่อส่งข้ามไปยังเครือข่ายใด ๆ ได้อย่างถูกต้อง โดยจะทำงานแบบ Packet Switching คือมีการส่งผ่านข้อมูลผ่านอุปกรณ์ Router หรือ Gateway ในระบบเครือข่ายจนกว่าจะถึงปลายทางที่กำหนดโดย IP Address เมื่อข้อมูลไปถึงปลายทางที่กำหนดโดย IP Address แล้วจะมีกลไกการแปลงหมายเลข IPAddress ให้เป็นหมายเลขประจำตัวอุปกรณ์ (MAC Address) ที่ถูกต้องอีกทีหนึ่งด้วยโปรโตคอล ARP (Address Resolution Protocol) นอกจากนี้ยังมีการใช้งานโปรโตคอล ICMP (Internet Control Message Protocol) ควบคู่ไปด้วย โดยจะมีหน้าที่ในการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งข้อความที่โปรโตคอล ICMP ส่งนั้นแบ่งออกได้เป็น 2 แบบคือ ICMP Error Message หรือข้อมูลความแจ้งความผิดพลาด และ ICMP Query หรือข้อความเรียกขอข้อมูลเพิ่ม ดังรูปที่ 3.16 แสดงความสัมพันธ์ของโปรโตคอลที่ใช้ร่วมกับ IP

ในการส่งผ่านข้อมูล หรือ IP Datagram ไปยังเครือข่ายอินเตอร์เน็ตนั้น โปรโตคอล IP จะทำการพิจารณาว่าปลายทางในการส่งนั้นภายในเครือข่ายของตนเองหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยตรวจสอบว่าหมายเลขเครือข่าย (Network Address) ถ้าหมายเลขเครือข่ายของ IP Address เหมือนกับของต้นทางหรือไม่ ถ้าตรงกันแสดงว่าเป็นการส่งข้อมูลภายในเครือข่ายเดียวกัน แต่ถ้าต่างกันแสดงว่าต้องส่งข้อมูลไปยังปลายทางที่อยู่คนละเครือข่ายกัน ซึ่งจะมีกลไฟที่ต่างกัน ดังนี้

การส่งข้อมูลภายในเครือข่ายเดียวกัน มีกลไกดังต่อไปนี้

1. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARS เพื่อแปลงหมายเลข IP Address ปลายทางให้เป็นค่าความหมายเลขประจำตัวของอุปกรณ์ เช่น MAC Address เป็นต้น

2. เมื่อโปรโตคอล IP ได้รับหมายเลขประจำตัวอุปกรณ์แล้ว ก็จะทำการส่งข้อมูลนั้นไปยังอุปกรณ์ที่ระบุไว้

การส่งข้อมูลข้ามเครือข่าย มีกลไกดังต่อไปนี้

1. โปรโตคอล IP ตรวจสอบพบว่าหมายเลข IP Address ปลายทางอยู่คนละเครือข่ายกัน โปรโตคอล IP ก็จะทำการอ่านค่า IP Address ของ Router เพื่อเตรียมส่งข้อมูลไปที่ Router แทน (กำหนดเป็น Default Router)

2. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARS เพื่อแปลงหมายเลข IP Address ปลายทางให้เป็นค่าหมายเลขประจำตัวของอุปกรณ์

3. โปรโตคอล IP ส่งข้อมูล IP Datagram ไปยัง Router ที่กำหนดไว้ จากนั้น Router ก็จะทำการส่งผ่านข้อมูลข้ามเครือข่ายต่อไป

2.3.4.6 การส่งข้อมูลโดยอาศัย Router และ Routing Algorithm

อุปกรณ์ Router ทำหน้าที่ในการเชื่อมโยงเครือข่ายที่อยู่ห่างไกลเข้าด้วยกัน ไม่ว่าเครือข่ายนั้นจะต่างกันหรือเหมือนกันทางด้านกายภาพก็ตาม การเชื่อมโยงนี้มีได้หลายลักษณะ โดยส่วนใหญ่แล้วจะทำการเชื่อมโยงกันโดยผ่านบริการ Wide Area Network Services (WAN Services) เช่น ISDN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

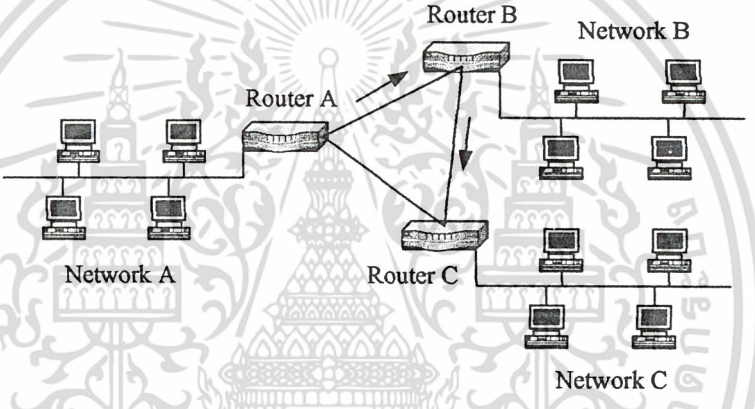
(Integrated Services Digital Network), Frame Relay, Point-to-Point Leased Circuit, X.25 เป็นต้น โดยมีบทบาทการทำงานอย่างคร่าว ๆ ดังต่อไปนี้

- เมื่ออุปกรณ์ Router ได้รับ IP Datagram มาจะทำการตรวจสอบความถูกต้องของข้อมูลก่อน ด้วยขบวนการ Checksum จนกว่าจะได้ข้อมูลที่ถูกต้อง

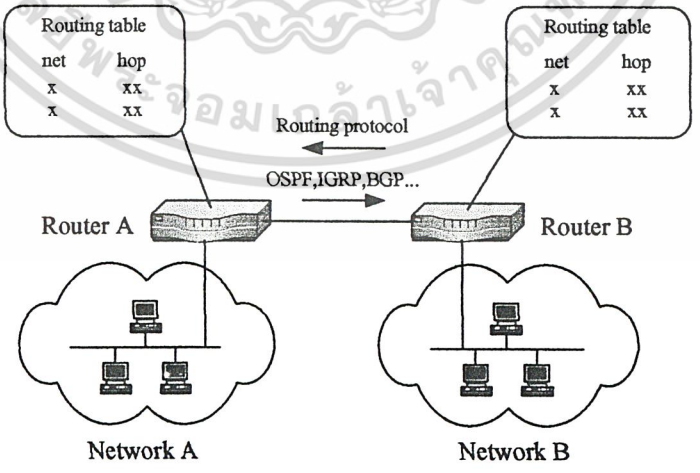
- ต่อมา Router จะทำการพิจารณาเส้นทางในการส่งข้อมูล ด้วยขบวนการ Router Algorithm

- เมื่อทราบเส้นทางที่จะทำการส่งข้อมูลแล้ว ก็จะทำการนำข้อมูลใส่ลงในลำต้น (Queue) เพื่อรอการส่งต่อไป

- นอกจากนี้ Router จะทำการรับส่งข้อมูล Routing Table ระหว่าง Router ด้วยกันเอง เพื่อเป็นการแลกเปลี่ยนข้อมูลเกี่ยวกับเส้นทางการส่งผ่านข้อมูลเป็นระยะ ๆ ตามเวลาที่กำหนด โดยในการแลกเปลี่ยนนั้นจะอาศัยโปรโตคอลที่เรียกว่า Routing Protocol



รูป 2.23 แสดงการส่งข้อมูลผ่านเราท์เตอร์



รูป 2.24 แสดงการแลกเปลี่ยนข้อมูลเกี่ยวกับเส้นทางการส่งผ่านข้อมูลระหว่าง Router

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

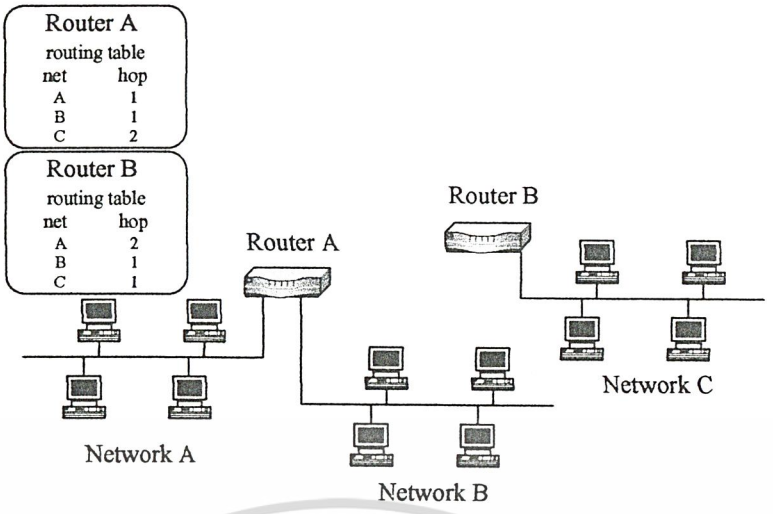
ขบวนการพิจารณาเส้นทางของการส่งผ่านข้อมูล หรือที่เรียกว่า Routing Algorithm มีบทบาทในการปรับปรุงข้อมูลใน Routing Table ให้ใหม่อยู่เสมอ ตามสภาวะที่มีการเปลี่ยนแปลงไปของเครือข่าย รวมทั้งยังมีบทบาทในการพิจารณาว่า การส่งข้อมูลนั้นจะเป็นไปอย่างมีประสิทธิภาพตามเงื่อนไขที่กำหนดไว้หรือไม่ ช่วยให้ไม่พบกับปัญหาการส่งข้อมูลไปถึงปลายทาง ดังนั้นในการติดต่อระหว่าง Router รวมถึงวิธีคิดพิจารณาเส้นทางการส่งผ่านข้อมูลที่สามารถทำได้หลายแบบ ซึ่งจะกำหนดได้เป็นโปรโตคอลสำหรับการพิจารณาเส้นทางการผ่านข้อมูลหรือ Routing Protocol ซึ่งจะทำงานบนโปรโตคอล IP อีกทีหนึ่ง โดยโปรโตคอล IP สามารถผนึกข้อมูลหรือโปรโตคอลที่จะระบุเส้นทางส่งผ่านข้อมูลเข้าไปได้

Routing Algorithm มีหลายประเภท ซึ่งแต่ละประเภทมีวิธีการคำนวณพิจารณาหาเส้นทางและวิธีปรับปรุงข้อมูลระหว่าง Router ด้วยกันที่แตกต่างกันด้วย ดังนั้นถ้าต้องการให้สามารถเชื่อมโยงเครือข่ายระหว่างกันได้ และสามารถส่งข้อมูลข้ามเครือข่ายได้อย่างถูกต้องแล้วละก็ อุปกรณ์ Router ที่ใช้เชื่อมต่อกันนั้นจำเป็นต้องใช้งาน Routing Protocol ที่เหมือนกันด้วย โดยสามารถแบ่งประเภทของ Routing Algorithm ได้เป็น 2 ประเภทใหญ่ ๆ คือ

1. Interior Routing Protocol มักใช้งานกับเครือข่ายขนาดเล็กที่มีเครือข่ายย่อยเชื่อมต่อเป็นสมาชิกอยู่ โดยใช้เป็นเส้นทางการติดต่อแลกเปลี่ยนข้อมูลภายในกลุ่มสมาชิกด้วยกัน เช่น บริษัทผู้ให้บริการอินเตอร์เน็ต (ISP) จะมีการเชื่อมต่อเครือข่ายย่อย ๆ คือเครือข่ายของลูกค้าแบบองค์กรที่เชื่อมต่อเครือข่ายของตนเข้ากับเครือข่ายของ ISP ทำให้เครือข่ายของลูกค้าอยู่ในเครือข่าย อินเตอร์เน็ตได้ โดยในการติดต่อส่งผ่านข้อมูลอุปกรณ์ Router จะทำการเปลี่ยนแปลงข้อมูลของ Routing Table เพื่อให้ทราบว่าเส้นทางใดเป็นเส้นทางในการส่งผ่านข้อมูล ตัว Router ของบริษัทผู้ให้บริการ (ISP) จะทำหน้าที่เป็น Router หลักอยู่ในระบบที่เรียกว่า Autonomous System (AS) ที่จะทำหน้าที่เชื่อมคยงไปสู่ระบบภายนอกและออกสู่ระบบอินเตอร์เน็ตต่อไป โดย AS นี้จะทำการเชื่อมโยงระหว่าง Router หลักของผู้ใช้บริการเข้าด้วยกัน Router หลักของ ISP จะมีหมายเลขประจำตัวหรือ AS Number ที่ไม่ซ้ำกัน และมีโปรโตคอลบางชนิดที่ใช้งานค่าหมายเลข AS Number ในการแลกเปลี่ยนสถานะของเครือข่ายและข้อมูล Routing Table โดยมีแบ่งย่อยอีกได้เป็นอีกหลายแบบ ตัวอย่างเช่น

1.1) Distance-Vector Routing Protocol มีลักษณะที่สำคัญคือ ในแต่ละ Router จะมีข้อมูล Routing Table เอาไว้พิจารณาเส้นทางการส่งข้อมูล โดยพิจารณาจากระยะทางที่ข้อมูลจะไปถึงปลายทางเป็นหลักโดยจะนับเป็นจำนวน Hop ดังรูปที่ 3.19 โดยการส่งในลักษณะของ Distance Vector Routing Protocol จะเลือกหาเส้นทางที่ดีที่สุดและมีการคำนวณตาม Routing Algorithm เพื่อให้ได้ผลลัพธ์ออกมาซึ่งมักจะเป็นเส้นทางที่มีจำนวน Hop น้อยที่สุด อุปกรณ์ Router ที่เชื่อมต่อกันจะทำการปรับปรุงข้อมูลใน Routing Table อยู่เป็นระยะ ๆ ด้วยการ Broadcast ข้อมูลทั้งหมดใน Routing Table ไปในเครือข่ายตามระยะเวลาที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.25 แสดงการทำงานของ Link-State Routing Protocol

1.2) Link-State Routing Protocol มีกลไกการทำงาน คือ อุปกรณ์ Router จะทำการ Broadcast ข้อมูลการเชื่อมต่อของเครือข่ายตนเองไปให้ตัวอุปกรณ์อื่น ๆ ทราบ ข้อมูลนี้เรียกว่า Link-State ซึ่งเกิดจากการคำนวณโดยตัวอุปกรณ์ที่ทำการคำนวณ จะทำการคำนวณหาค่าเชื่อมต่อโดยการพิจารณาให้ Router ของตนเองเป็นหลักในการสร้าง Routing Table ขึ้นมา ดังนั้นข้อมูล Link-State ที่ส่งออกไปในเครือข่ายจะเป็นข้อมูลที่บอกเส้นทางที่ดีที่สุดของตนเอง โดยไม่สนใจอุปกรณ์อื่น บางครั้งเรียกว่าเป็น Intradomain Routing Protocol ดังรูปที่ 2.25

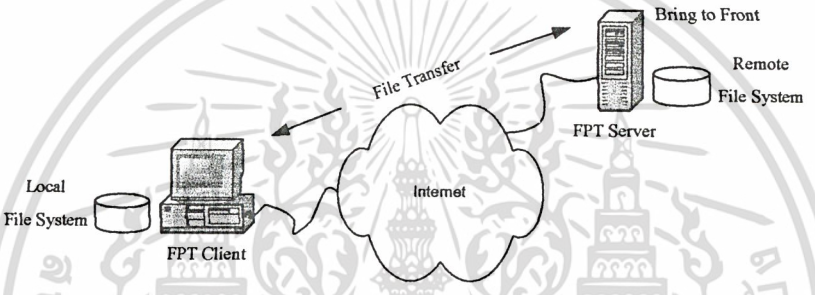
2. Exterior Routing Protocol เป็นการเชื่อมต่อเครือข่ายภายใน เช่น ISP เข้ากับเครือข่ายอินเตอร์เน็ตนั้น จำเป็นต้องมีการเชื่อมต่อแบบ Exterior Routing Protocol และอาศัยหมายเลข AS Number ในการติดต่อ ถ้าอาศัยหลักการของ Interior Routing Protocol คือให้ Router ในเครือข่ายทุกตัวในอินเตอร์เน็ตส่งค่า Routing Table ออกไปในเครือข่ายนั้นไม่สามารถทำได้เพราะช่องสัญญาณที่มีอยู่ไม่เพียงพอ และส่งผลให้เกิดการติดขัดได้ บางครั้งเราเรียกโปรโตคอล Exterior Routing Protocol ว่าเป็น Interdomain Routing Protocol ซึ่งมีลักษณะดังรูปที่ โดยหลักการทำงานจะใช้ในลักษณะของ Distance-Vector Routing Protocol และ Link-State Roting Protocol เช่นกัน แต่จะใช้การแลกเปลี่ยนข้อมูลระหว่าง Router หลักที่มีหมายเลข AS Number แทน ประกอบกับต้องมีการพิจารณาปัจจัยอื่น ๆ ประกอบด้วย เช่น ขนาดของช่องสัญญาณ (Bandwidth) จำนวน Delay ที่เกิดขึ้นในช่องสัญญาณ, ความหนาแน่นของช่องสัญญาณ และความน่าเชื่อถือของช่องสัญญาณ (Reliable) เป็นต้น ตัวอย่างโปรโตคอลที่ใช้งานได้แก่ RIP (Routing Information Protocol), IGRP (Interior Gateway Routing Protocol), OSPF (Open Shortest Path First), EGP (Exterior Gateway Protocol) และ BGP (Border Gateway Protocol) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.7 การถ่ายโอนไฟล์ : FTP

FTP (File Transfer Protocol) เป็นโปรโตคอลสำหรับการถ่ายโอนไฟล์ระหว่างสองเครื่อง โปรโตคอล FTP นั้นมีมาพร้อมกับอินเตอร์เน็ทในสมัยแรกๆและยังเป็นโปรโตคอลที่ยังเป็นที่นิยมในปัจจุบัน FTP ถูกอธิบาย RFC 959 ในดังรูปที่2.26 ซึ่งแสดงลักษณะการถ่ายโอนไฟล์ระหว่างเครื่องด้วย FTP

โดยทั่วไปเมื่อมีผู้ใช้ต้องการที่จะถ่ายโอนไฟล์ระหว่างเครือข่าย ผู้ใช้ก็จะเปิดโปรแกรม FTP ซึ่งสิ่งที่ผู้ใช้ต้องระบุในการเชื่อมต่อครั้งแรกคือ ชื่อหรือที่อยู่ของ FTP เซิร์ฟเวอร์พร้อมทั้งชื่อล็อกอินและรหัสผ่านหลังจากนั้นไคลเอนท์จะสร้างการเชื่อมต่อ TCP กับเซิร์ฟเวอร์ และส่งข้อมูลเกี่ยวกับล็อกอินเพื่อเซิร์ฟเวอร์จะได้ตรวจสอบสิทธิ์ของผู้ใช้ และถ้าตรวจสอบสิทธิ์ผ่านผู้ใช้ก็สามารถอัพโหลดไฟล์ หรือดาวน์โหลดไฟล์ระหว่างเครื่องของผู้ใช้และเซิร์ฟเวอร์ได้

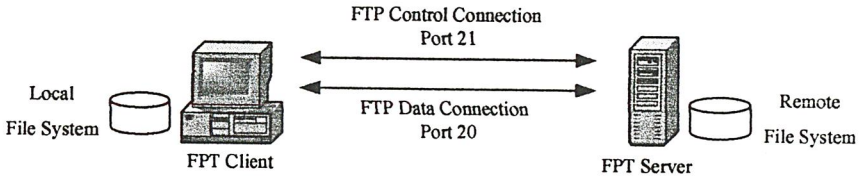


รูปที่2.26 แสดงการถ่ายโอนไฟล์ด้วยโปรโตคอล FTP

โปรโตคอล FTP และ HTTP มีหลายอย่างที่เหมือนกัน เช่น ทั้งสองเป็นโปรโตคอลสำหรับถ่ายโอนไฟล์และนอกจากนี้ใช้การเชื่อมต่อแบบ TCP เหมือนกัน อย่างไรก็ตามทั้งสองโปรโตคอลมีข้อแตกต่างที่สำคัญคือ โปรโตคอล FTPจะใช้การเชื่อมต่อ TCP ที่ขนานกันสองการเชื่อมต่อการเชื่อมต่อแรกใช้สำหรับการควบคุมการถ่ายโอนไฟล์ (Control Connection) ส่วนการเชื่อมต่อที่สองจะใช้สำหรับการถ่ายโอนข้อมูลหรือไฟล์ (Data Connection)

ช่องการเชื่อมต่อควบคุมนั้นจะใช้สำหรับการส่งข้อมูล หรือคำสั่งที่ใช้สำหรับควบคุมการถ่ายโอนไฟล์ระหว่างโฮสต์ เช่น ชื่อล็อกอิน รหัสผ่าน คำสั่งสำหรับการเปลี่ยนไดเร็คทอรี หรือ คำสั่งสำหรับการอัพโหลดไฟล์ (put) และคำสั่งสำหรับการดาวน์โหลดไฟล์ (get) เป็นต้น ส่วนช่องการเชื่อมต่อข้อมูลนั้นก็ใช้สำหรับการถ่ายโอนไฟล์ ส่วนโปรโตคอล HTTP นั้นจะใช้การเชื่อมต่อเดียวสำหรับทั้งการรับส่งข้อมูล การควบคุมและไฟล์เว็บเพจ รูปที่ 2.27 แสดงช่องทางการเชื่อมต่อของ FTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

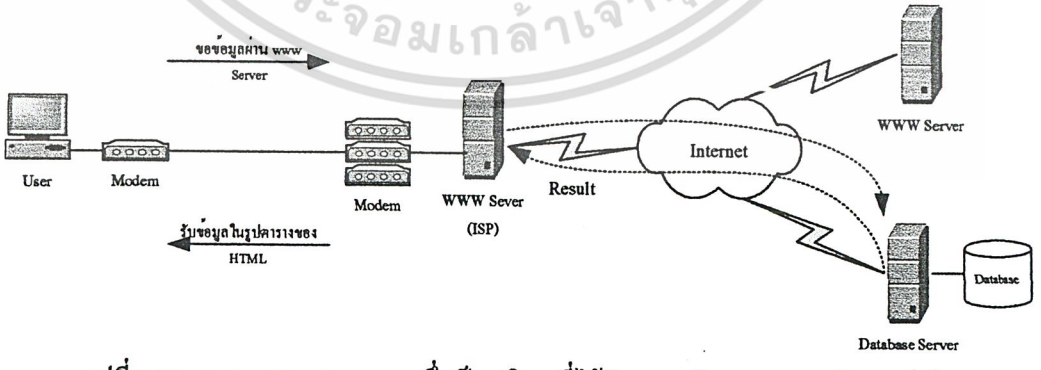


รูปที่2.27 พอร์ตสำหรับสัญญาณควบคุมและถ่ายโอนไฟล์ของโปรโตคอล FTP

กระบวนการถ่ายโอนไฟล์ด้วย FTP นั้นจะเริ่มจากไคลเอนท์สร้างการเชื่อมต่อ TCP กับทางฝั่งเวิร์ฟเวอร์ผ่านทางพอร์ต21 ซึ่งการเชื่อมนี้เป็นช่องสำหรับการส่งข้อมูลการควบคุมการถ่ายโอนไฟล์เมื่อสร้างการเชื่อมต่อสำหรับต่อไปยังฝั่งไคลเอนท์ก็จะส่งข้อมูลล็อกอิน เช่น ชื่อผู้ใช้และรหัสผ่านไปให้ทางเซิร์ฟเวอร์ตรวจสอบสิทธิ์ เมื่อเซิร์ฟเวอร์ตรวจสอบสิทธิ์ผ่านไคลเอนท์ก็สามารถดาวน์โหลด หรืออัพโหลดไฟล์ได้ ซึ่งมีขั้นตอนดังนี้ เช่น เมื่อไคลเอนท์ต้องการดาวน์โหลดไฟล์ไคลเอนท์ก็จะส่งข้อมูลเกี่ยวกับไฟล์นั้นไปให้ทางฝั่งเซิร์ฟเวอร์ผ่านทางพอร์ต21 เมื่อเซิร์ฟเวอร์ได้รับการร้องขอก็จะสร้างการเชื่อมต่อใหม่โดยใช้พอร์ต20 กับทางฝั่งไคลเอนท์ หลังจากนั้นไฟล์ที่ร้องขอนั้นก็จะถูกถ่ายโอนผ่านทางพอร์ต 20 และเมื่อถ่ายโอนไฟล์เสร็จก็จะปิดการเชื่อมต่อที่พอร์ต 20 และเมื่อมีการร้องขอใหม่ ซึ่งอาจจะเป็นการอัพโหลดหรือการดาวน์โหลดไฟล์ก็จะสร้างการเชื่อมต่อ TCP ผ่านพอร์ต20ใหม่ แต่ช่องการเชื่อมต่อสำหรับการควบคุมนั้นยังคงสภาพไว้จนกว่าฝั่งไคลเอนท์จะยกเลิก

2.4 เว็บเซิร์ฟเวอร์

เว็บเซิร์ฟเวอร์ คือ แอพพลิเคชั่นทำหน้าที่รับ และประมวลผลเอกสาร ที่ถูกร้องขอจากผู้ใช้บริการอินเตอร์เน็ท จากนั้นเว็บเซิร์ฟเวอร์ก็จะส่งเอกสารกลับไปแสดงผลให้ผู้ใช้บริการผ่านบราวเซอร์ นอกจากเว็บเซิร์ฟเวอร์จะถูกนำมาให้บริการในอินเตอร์เน็ตแล้ว แต่อาจมีการประยุกต์ให้นำมาใช้กับเครือข่ายภายในองค์กร หรือ อินเตอร์เน็ตได้เช่นกัน



รูปที่ 2.28แสดงภาพรวมของ wwwซึ่งเป็นบริการที่ได้รับความนิยมสูงสุดของอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เดิมนั้นเว็บเซิร์ฟเวอร์มักจะอยู่ในเครื่องคอมพิวเตอร์ ยูนิกซ์ ที่มีประสิทธิภาพสูง และราคาแพง ต่อมา เมื่ออินเตอร์เน็ตขยายความนิยมมาสู่ผู้ใช้ พีซี ทำให้มีการพัฒนาซอฟแวร์ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์บนพีซี ซึ่งสามารถรันได้ทั้งวินโดวส์ 95/98 และ วินโดวส์เอ็นที เซิร์ฟเวอร์/Workstation ตัวอย่างเช่น

- NCSA Web Server จาก NCSA เป็นเว็บเซิร์ฟเวอร์ที่สามารถดาวน์โหลดได้ฟรีจากเว็บไซต์ที่ให้บริการดาวน์โหลดฟรีทั่วไป

- Net Server จาก Netscape เป็น เวิลด์ ไวด์ เว็บ (World Wide Web: www) เซิร์ฟเวอร์ (Server) ที่มีความสามารถรองรับ จาวา (JAVA) ได้อย่างเต็มรูปแบบ

- ออราเคิลเว็บเซิร์ฟเวอร์ (Oracle Web Server) จากออราเคิล เป็นเว็บเซิร์ฟเวอร์ที่เน้นความสามารถด้านการติดต่อกับเซิร์ฟเวอร์ฐานข้อมูลโดยเฉพาะการใช้งานร่วมกับระบบจัดการฐานข้อมูลของ ออราเคิล ปัจจุบันเราอาจได้ยินชื่อที่ตั้งขึ้นใหม่เป็น แอพพลิเคชั่น เซิร์ฟเวอร์ (Application Server) (คือ ออราเคิล เว็บเซิร์ฟเวอร์ ตัวเดิมที่ถูกพัฒนาขึ้นมา)

- Personal Web Server จากไมโครซอฟท์เป็นเว็บเซิร์ฟเวอร์ที่สามารถใช้งานได้กับ วินโดวส์ 95/98 หรือ วินโดวส์เอ็นที เวิร์คสเตชั่น และรองรับการใช้งานร่วมกับ เอเอสพี (Active Server Page) โดยเราสามารถดาวน์โหลดมาใช้ได้ฟรีจากเว็บไซต์ของไมโครซอฟท์ แต่ถ้าใช้งาน วินโดวส์ 98 ก็สามารถเลือกติดตั้งได้ ซึ่งมักใช้ในการทดสอบเว็บเพจ หรือ แอพพลิเคชั่นอินเตอร์เน็ต ก่อนจะนำไปใช้งานจริง

- IIS (Internet Information Server) จากไมโครซอฟท์ เป็นอินเตอร์เน็ต เซิร์ฟเวอร์ เวอร์ชั่น 2 ที่แถมมากับ วินโดวส์ เอ็นที เซิร์ฟเวอร์ 4.0 มีความสามารถให้บริการได้ทั้ง เวิลด์ ไวด์ เว็บ, เอฟ ที พี (FTP: File Transfer Protocol) และ Gopher ส่วนเวอร์ชั่นที่ใช้งานกับเอเอสพีได้จะเป็นเวอร์ชั่น 3.0 ขึ้นไป สำหรับบทนี้ จะแสดงการใช้งานกับเวอร์ชั่น 4.0

2.5 เวิลด์ ไวด์ เว็บ คืออะไร

เมื่อไม่กี่ปีก่อนหน้านี้มีผู้สนใจใช้งานอินเตอร์เน็ตไม่มากนัก เนื่องจากการใช้บริการอินเตอร์เน็ตไม่ว่าจะเป็น การค้นหาข่าวสารข้อมูล การรับส่งจดหมายอิเล็กทรอนิกส์ การสำเนาไฟล์ ฯลฯ จะอยู่ในรูปแบบของตัวอักษร (Text Mode) เท่านั้น ไม่มีการแสดงที่เป็น รูปภาพ เสียง และไม่มีตัวอักษรแบบต่างๆ ปรากฏให้เห็นแต่อย่างใด นอกจากนี้ผู้ใช้ต้องเรียนรู้และจดจำคำสั่งคอมพิวเตอร์มากมาย เช่น ต้องเรียนรู้คำสั่งเบื้องต้นของยูนิกซ์ (UNIX) เนื่องจากเมื่อมีการเรียกใช้อินเตอร์เน็ต เครื่องคอมพิวเตอร์ของผู้ใช้จะถูกเปลี่ยนให้เป็นเทอร์มินอล ของโฮสต์คอมพิวเตอร์ที่ให้บริการอินเตอร์เน็ต และโฮสต์ส่วนมากจะทำงานอยู่ภายใต้ระบบปฏิบัติการยูนิกซ์ ดังนั้นผู้ใช้จึงจะต้องเรียนรู้คำสั่งเบื้องต้นของยูนิกซ์ (เช่นเดียวกับการเรียนรู้คำสั่งดอส (DOS) บนเครื่องพีซี) เพื่อทำการป้องกันคำสั่งที่เป็นตัวอักษรด้วยตัวเองให้โฮสต์คอมพิวเตอร์ทำงานตามที่เราต้องการใช้

จนกระทั่งมีบริการที่เรียกว่า เวิลด์ ไวด์ เว็บ หรือเครือข่ายใยแมงมุมเกิดขึ้น ทำให้ความนิยมการใช้อินเตอร์เน็ตสูงขึ้นเป็นทวีคูณ เนื่องจาก เวิลด์ ไวด์ เว็บ เป็นบริการอันหนึ่งที่มีอยู่ในอินเตอร์เน็ต ทำให้การใช้งานอินเตอร์เน็ตเป็นเรื่องที่ง่ายขึ้น ผู้ใช้ไม่ต้องจดจำคำสั่งของยูนิกซ์อีกต่อไป การอ่านและค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข่าวสารข้อมูล ทำได้โดยการกดปุ่มบนเมาส์เพียงอย่างเดียว นอกจากนี้ ข่าวสาร ข้อมูล หรือเอกสารที่เก็บอยู่ในอินเตอร์เน็ตจะอยู่ในรูปแบบของสื่อผสม (Multimedia) ที่เรียกว่า "เว็บเพจ" (web page) อันประกอบไปด้วยอักษรข้อความที่มีรูปแบบต่างๆ รูปภาพ เสียง ภาพเคลื่อนไหว วีดีโอ และไฮเปอร์เท็กซ์ (Hypertext) ซึ่งเป็นการแสดงข้อความที่มีการเชื่อมโยงถึงกันได้เหมือนกับการเรียกใช้ Help ในโปรแกรมวินโดวส์ โดยผู้ใช้สามารถเรียกดูเอกสารหนึ่งจากอีกเอกสารหนึ่งได้ ซึ่งเป็นลักษณะการเชื่อมต่อที่คล้ายกับใยแมงมุม จึงมีการตั้งชื่อบริการนี้ว่าเป็นเครือข่ายใยแมงมุม

ส่วนประกอบของ เวิลด์ ไวด์ เว็บ

การจะใช้บริการ เวิลด์ ไวด์ เว็บ ได้นั้น จำเป็นต้องมีส่วนประกอบเหล่านี้

- แหล่งข้อมูลหรือเว็บไซต์
- โปรแกรมเว็บบราวเซอร์

1. แหล่งข้อมูลหรือเว็บไซต์

เว็บไซต์ หรือ เว็บเซิร์ฟเวอร์ คือ ระบบคอมพิวเตอร์ที่เป็นแหล่งเก็บเว็บเพจที่ผู้ใช้บริการสามารถเรียกดูเว็บเพจที่เก็บอยู่ในเว็บไซต์นั้นได้ ซึ่งเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเว็บไซต์อาจจะใช้ระบบปฏิบัติการยูนิกซ์ หรือวินโดวส์ เอ็นที ก็ได้ และจะต้องมีการติดตั้งโปรแกรมจัดการ ที่จะทำให้เครื่องคอมพิวเตอร์เครื่องนั้นทำหน้าที่เป็นเว็บไซต์ ตัวอย่างเช่น ในกรณีที่เครื่องคอมพิวเตอร์เครื่องนั้นใช้ระบบปฏิบัติการวินโดวส์ เอ็นที จะมีซอฟท์แวร์เว็บเซิร์ฟเวอร์ให้เลือกใช้

2. โปรแกรมเว็บบราวเซอร์

เป็นโปรแกรมหรือซอฟท์แวร์ที่ใช้ในการเข้าสู่ เวิลด์ ไวด์ เว็บ จะเปิดดูเว็บเพจที่เก็จอยู่ในเว็บไซต์ใดๆ ตัวอย่างของ โปรแกรมเว็บบราวเซอร์ ที่มีให้ดาวน์โหลดใช้กันได้ฟรีได้แก่ โปรแกรม Netscape Navigator จากบริษัท Netscape Communications และโปรแกรม Microsoft Internet Explorer จากบริษัทไมโครซอฟท์ ผู้ผลิตโปรแกรมวินโดวส์ที่ผู้คนรู้จักกันดี เป็นต้น ซึ่งโปรแกรมเว็บบราวเซอร์ส่วนใหญ่นั้นจะทำงานร่วมกันกับโปรแกรมวินโดวส์ และนอกจากจะใช้เพื่อดูเว็บเพจจากเว็บไซต์ใดๆแล้ว หลายโปรแกรมยังมีความสามารถอื่นๆ เช่น บริการส่งจดหมายอิเล็กทรอนิกส์ การค้นหาข้อมูล การโอนถ่ายโปรแกรมด้วย FTP เป็นต้น

2.6 โฮมเพจ

โฮมเพจ (Home Page) จะหมายถึงหน้าแรกของเว็บเพจทั้งหมดที่ผู้ใช้บริการอินเตอร์เน็ตจะพบเมื่อมีการเข้าไปยังเว็บไซต์ใดๆ โฮมเพจเปรียบเสมือนกับสารบัญ และคำนำที่เจ้าของเว็บไซต์สร้างขึ้นเพื่อจะใช้ประชาสัมพันธ์องค์กรของตนว่าให้บริการในสิ่งใดบ้าง นอกจากนี้แล้วภายในโฮมเพจก็อาจมีเอกสารข้อความอื่นๆที่เชื่อมโยงต่อจากโฮมเพจนั้นๆได้อีก ที่เรียกว่า เว็บเพจ ซึ่งโฮมเพจหนึ่งๆก็อาจมีการเชื่อมกับเว็บเพจอื่นๆเป็นจำนวนมากก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 เอเอสพี (Active Sever Page : ASP)

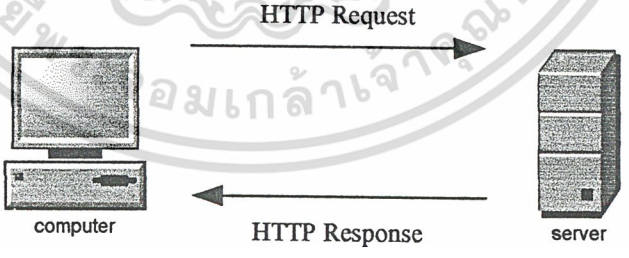
เอเอสพี เป็นคำที่ย่อมาจาก Active Sever Pages ซึ่งเป็นโปรแกรมที่พัฒนาขึ้นโดยบริษัทไมโครซอฟท์ เพื่อใช้งานทางด้านอินเตอร์เน็ต โดย เอเอสพี จะทำหน้าที่ตีความเอกสารที่เขียนด้วยภาษาสคริปท์ เช่น VB สคริปท์ โดยที่มี เอเอสพี tag (คือคำสั่งที่มีเครื่องหมาย <%%> กำกับอยู่ซึ่งบราวเซอร์ทั่วไป เช่น Netscape Navigator หรือ Internet Explorer ไม่สามารถนำไปแสดงผล จากนั้นจึงสร้างผลลัพธ์เป็นเอกสาร HTML อันเป็นเอกสารที่ประกอบด้วย HTML tag ต่างๆ (คือ คำสั่งที่มีเครื่องหมาย < >) กำกับอยู่ซึ่งบราวเซอร์ทั่วไปสามารถนำไปสร้างเป็นเว็บเพจขึ้นเพื่อใช้แสดงผลได้

การทำงานของโปรแกรม เอเอสพี จะเกิดขึ้นเฉพาะทางฝั่งเซิร์ฟเวอร์เท่านั้น เราจึงเรียกว่าเป็นการทำงานแบบ เซิร์ฟเวอร์ไซด์ จากนั้นผลลัพธ์ที่ได้จะถูกส่งไปให้เว็บเซิร์ฟเวอร์ แล้วเว็บเซิร์ฟเวอร์ก็จะส่งเอกสารดังกล่าวต่อไปยังบราวเซอร์อีกทีหนึ่ง เมื่อบราวเซอร์ได้รับเอกสารนั้นแล้ว บราวเซอร์ก็จะสามารถแสดงผลได้ถูกต้องครบถ้วน โดยการทำงานของบราวเซอร์ทางฝั่งของผู้ใช้นี้เราเรียกว่าเป็นการทำงานแบบ ไคลเอนไซด์ (client side)

2.7.1 ขบวนการทำงานของ เอเอสพี

เพื่อได้ศึกษา เอเอสพี ในระดับเบื้องลึกลงไป ก่อนอื่นขอให้ลองทำความเข้าใจพื้นฐานขบวนการทำงานของ เอเอสพี (เอเอสพี model)

การทำงานทั้งหมดจะเริ่มจาก บราวเซอร์ร้องขอเอกสาร HTML ไปยังเว็บเซิร์ฟเวอร์ผ่านทาง HTTP (HTTP request) โดยที่เอกสารที่ขอไปจะเป็นแฟ้มข้อมูลที่มีนามสกุลเป็น ASP (เช่น search.ASP ฯลฯ) เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอดังกล่าว ก็จะส่งเอกสารนั้นไปให้ เอเอสพีตีความจากนั้น เอเอสพีก็จะสร้างเอาสาร HTML ส่งกลับไปให้เซิร์ฟเวอร์เพื่อส่งต่อไปยังบราวเซอร์และใช้แสดงผลทางฝั่งผู้ใช้ต่อไป (HTTP response) ซึ่งการทำงานของ เอเอสพีนี้แทบไม่แตกต่างไปจากหลักการทำงาน โปรแกรม CGI (Common Gateway Interface) จนอาจกล่าวได้ว่า เอเอสพี ก็เป็นโปรแกรม CGI ประเภทหนึ่งเช่นกัน



รูปที่ 2.29 แสดงการทำงานของ ASP

การเขียนโปรแกรมเพื่อที่จะสร้างเอกสารที่จะทำงานกับ เอเอสพี นั้น (ในที่นี้ขอเรียกว่า เอกสารเอสเอพี)ไม่จำเป็นต้องอาศัยโปรแกรมเฉพาะในการเขียน เราสามารถนำ โปรแกรมประเภท text editor มา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้งานได้ทันที เช่น โปรแกรม Notepad ฯลฯ หรือจะใช้โปรแกรมที่เขียนเอกสาร เอเอสพี โดยเฉพาะก็ได้ เช่น Visual InteDev เป็นต้น

เอกสารเอเอสพี แตกต่างกับเอกสาร HTML ทั่วไปตรงที่มีส่วนของคำสั่ง เอเอสพี อยู่ในเอกสารด้วย โดยทั่วไปหากเรานำเอกสาร HTML มาเปลี่ยนเป็นเอกสาร เอเอสพี เลยก็ทำได้ นั้นคือวิธีการสร้างเอกสาร เอเอสพี แบบง่ายๆ เช่น เราสามารถเปลี่ยนเอกสาร HTML ที่ชื่อ index.html ไปเป็น index.asp ได้เลย โดยที่เมื่อโปรแกรม เอเอสพี ตีความส่วนใดของเอกสารที่มี HTML tag กำกับอยู่ ก็จะไม่เกิดการเปลี่ยนแปลงใดๆในเอกสารนั้นเลย แต่หากว่าส่วนใดมี เอเอสพี tag กำกัล เอเอสพี ก็จะเปลี่ยนแปลงเอกสารส่วนดังกล่าวไปอยู่ในรูปข้อความทั่วไปหรือเป็น HTML tag แทน เช่น หากในเอกสารมีคำสั่งนี้

```
<br><%response.write("Hello"&now) %>
```

ก็จะถูกเปลี่ยนเป็น...

```
<br>Hello แล้วตามด้วยวันและเวลาปัจจุบัน
```

2.7.2 เอกสาร ASP

เอกสาร ASP คือ เท็กซ์ไฟล์ที่บรรจุเอาคำสั่งสคริปต์ต่างๆ (อาจเขียนด้วยภาษาเดียวหรือ หลายภาษาผสมกันก็ได้) แทรกอยู่กับแท็ก HTML และสคริปต์ที่ทำงานฝั่งไคลเอนต์ เมื่อบราวเซอร์เรียกใช้งานเอกสาร ASP เอกสาร ASP นั้น ก็จะถูกแปล โดย ASP Interpreter ที่เซิร์ฟเวอร์ และถูกสั่งให้ทำงานที่เว็บเซิร์ฟเวอร์ โดยผลลัพธ์ที่ได้จากการรันแอพพลิเคชั่น ASP จะถูกส่งมายังบราวเซอร์ที่เรียกใช้ในรูปแบบเอกสาร HTML

เอกสาร ASP ประกอบด้วย

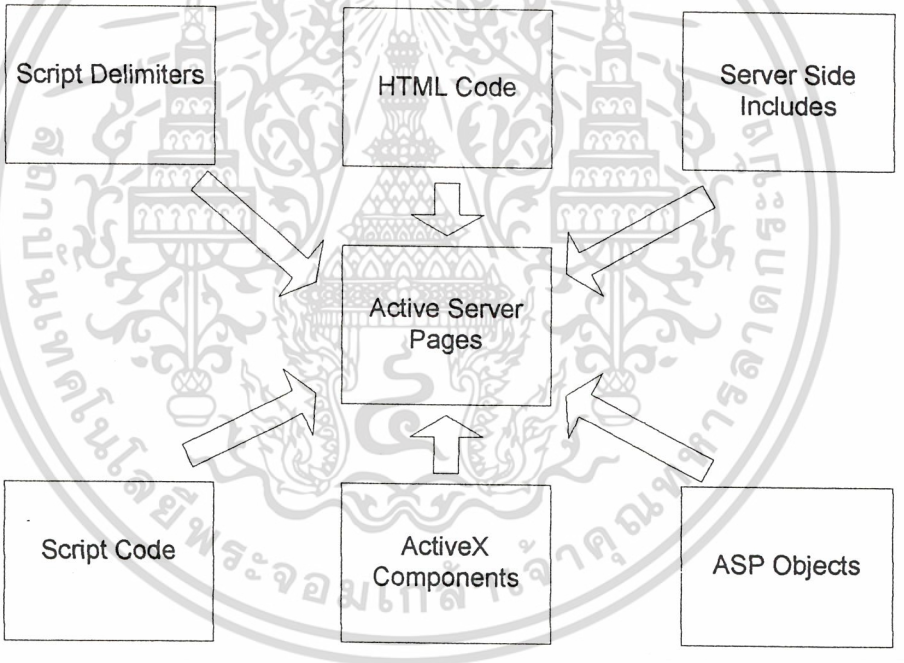
1. Server Cide Include (อาจมีหรือไม่ก้อได้)
2. HTML Code
3. Script delimiters คือ เครื่องหมายที่แยกโค้ด ASP ออกจากโค้ด HTML
4. Script Code เช่น VBScript ,Jscript, JaVaScript เป็นต้น
5. ActiveX Components
6. ActiveX Objects

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 ความต้องการของระบบในการใช้งาน ASP

ในการที่จะพัฒนาและใช้งานแอพพลิเคชั่น ASP นั้น มีความต้องการด้าน Software และ Hardware ดังนี้

1. หน่วยประมวลผล (CPU) ควรจะเป็น Pentium ขึ้นไป
2. Ram อย่างต่ำ 32 เมกะไบต์
3. พื้นที่ว่างใน HardDisk อย่างน้อย 150 เมกะไบต์
4. ระบบปฏิบัติการ Windows NT Server 4.0 , Windows NT Workstation 4.0 หรือ Windows 95/98/ME และต้องมีการติดตั้ง โปรโตคอล TCP/IP ด้วย
5. โปรแกรม Web Server ที่สนับสนุน ASP เช่น Internet Information Server 3.0/4.0 สำหรับ Windows NT และ Windows 2000 หรือ Personal Web Server สำหรับ Windows 95/98/ME
6. ถ้ามีการพัฒนาระบบฐานข้อมูล เราต้องมีโปรแกรมฐานข้อมูลที่รองรับ ODBC ด้วย เช่น Microsoft Access หรือ Microsoft SQL Server เป็นต้น



รูปที่ 2.30 แสดงองค์ประกอบในเอกสาร ASP ไฟล์หนึ่งๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.4 การแบ่งแยกโค้ด ASP ออกจากโค้ด HTML

เนื่องจากในเอกสาร ASP นั้น โค้ด ASP จะถูกแทรกอยู่กับโค้ด HTML การที่ตัวแปลโค้ด ASP (ASP Interpreter) ที่เว็บเซิร์ฟเวอร์จะสามารถแยกออกว่าส่วนไหนเป็นโค้ด HTML และส่วนไหนที่จะเป็นโค้ด ASP ต้องอาศัยเครื่องหมายที่ใช้แบ่งแยก ที่เราเรียกว่า ASP Delimiters โดย ASP Deilmiters มีอยู่ 3 รูปแบบ ซึ่งมีประโยชน์และความเหมาะสมในการใช้งานต่างๆกันดังนี้

1. เครื่องหมายที่ใช้กำหนดขอบเขตของโค้ด ASP ที่แทรกอยู่ตามโค้ด HTML

โค้ดที่เป็น ASP จะต้องอยู่ระหว่างเครื่องหมาย <% และ %> โดยมีรูปแบบการใช้งานดังนี้

```
<%
<โค้ด คำสั่งการทำงานของ ASP>
%>
```

2. เครื่องหมายที่ใช้แสดงค่าของนิพจน์หรือตัวแปร

สำหรับค่าที่ต้องการแสดงออกให้ผู้ใช้เห็น หรือต้องการให้ปรากฏในโค้ด HTML นั้น สามารถทำได้ง่ายๆ โดยการใส่ค่าตัวแปรหรือนิพจน์ที่ต้องการแสดงระหว่างเครื่องหมาย <%= และ %> โดยมีรูปแบบการใช้งานดังนี้

```
<%=<สิ่งที่ต้องการแสดง>%>
```

3. การกำหนดขอบเขตของ ASP โดยการระบุภาษาที่จะใช้

วิธีนี้จะใช้แอตทริบิวต์ RUNAT=SERVER แทรกในแท็ก <Script> เพื่อเป็นการบ่งบอกว่าสคริปต์ที่อยู่ระหว่างแท็ก <Script >และ </Script> ต่อไปนี้จะทำงานที่ฝั่งเซิร์ฟเวอร์เท่านั้น ซึ่งจะไม่แสดงโค้ดให้เห็นในฝั่งไคลเอนต์ด้วย

เนื่องจาก ASP นั้น อนุญาตให้เราสามารถเขียนสคริปต์คำสั่งได้หลากหลายภาษา (ซึ่งเราอาจเลือกใช้ภาษาใดภาษาหนึ่ง หรือใช้ผสมกันหลายภาษาก็ได้) ในกรณีที่เราไม่ได้ระบุภาษาที่เราจะใช้เขียน ASP จะถือว่าเราใช้ VB Script โดยปริยาย แต่ถ้าในกรณีที่เราต้องการใช้ภาษาอื่นที่ไม่ใช่ภาษา VB Script เราต้องระบุภาษาที่เราต้องการใช้แทรกเพิ่มเติม โดยแทรกแอตทริบิวต์ LANGUAGE = ภาษาที่เราจะใช้ภายในแท็ก <Script > ที่ใช้ในการแบ่งแยกโค้ด ASP ออกจากโค้ด HTML ด้วยมีรูปแบบการใช้งานดังนี้

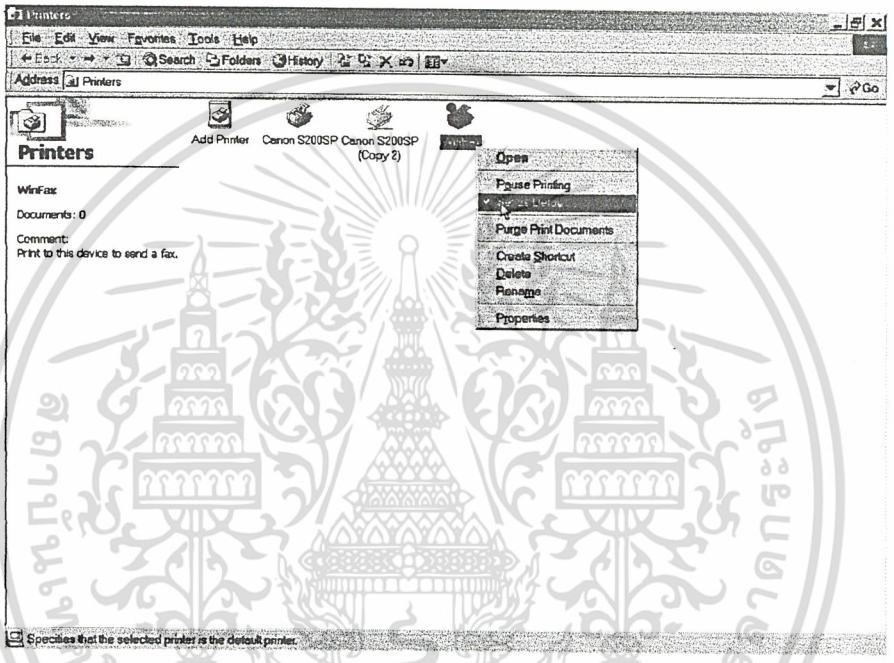
```
< Script RUNAT = SERVER LANGUAGE =language>
    < โค้ดคำสั่งการทำงาน >
</Script>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 การใช้งานโปรแกรม WINFAX PRO V.9

โปรแกรมนี้ สามารถทำการรับ และส่งโทรสารได้ ในลักษณะเดียวกับการใช้เครื่องโทรสารทุกประการ

โดยเมื่อเราติดตั้งโปรแกรม Winfax ระบบปฏิบัติการจะมอง โปรแกรม Winfax เปรียบเสมือนเครื่องพิมพ์เครื่องหนึ่ง ผู้ใช้สามารถเรียกโปรแกรม Winfax โดยใช้คำสั่ง print แต่ผู้ใช้ต้องทำการติดตั้ง Winfax เป็นเครื่องพิมพ์เริ่มต้น(set as default) ดังแสดงในรูป 2.28



รูปที่ 2.31 แสดงการติดตั้ง Winfax เป็นเครื่องพิมพ์เริ่มต้น(set as default)

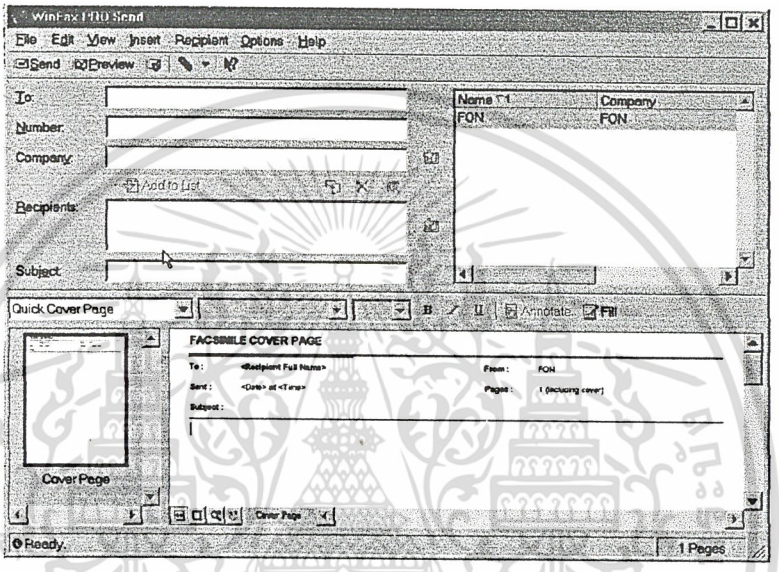
2.8.1 การรับ โทรสาร ผ่านโปรแกรม WINFAX PRO V.9

โดยที่การรับ โทรสาร จะทำได้โดยเรียกโปรแกรมนี้ให้ทำงานไว้ และทำการตั้งให้มีการรับโทรสาร อัตโนมัติ (หรือจะรับเองแบบ manual ก็ได้) เมื่อมีเสียงกระดิ่งโทรศัพท์ดัง โปรแกรมก็จะทำการรับสายโทรศัพท์นั้น และส่งสัญญาณ การเชื่อมต่อ โทรสาร ทันที เพื่อทำการรับ โทรสาร หลังจากนั้น หากทำการตั้งค่าให้พิมพ์ออกทางเครื่องพิมพ์อัตโนมัติไว้ โปรแกรมก็จะ ทำการพิมพ์ โทรสาร ออกทางเครื่องพิมพ์ได้ทันที เหมือนกันเครื่อง โทรสาร ทั่ว ๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 การส่ง โทรสาร ผ่านโปรแกรม WINFAX PRO V.9

ในส่วนของการส่ง โทรสาร มีหลักการทำงานง่าย ๆ คือ ตัวโปรแกรม จะทำงานเหมือนกับเป็นเครื่องพิมพ์ เครื่องหนึ่งที่อยู่ในระบบ Windows ดังนั้น แทนที่เราจะสั่งให้ทำการพิมพ์หน้ากระดาษ ออกทางเครื่องพิมพ์ ก็เปลี่ยนเป็นการพิมพ์ให้ออกทางโปรแกรม Winfax นี้แทน โดยผลลัพธ์ที่ได้ คือแผ่นโทรสาร ที่เครื่อง โทรสาร ปลายทาง การใช้งานแบบนี้ จะติดขัดปัญหาตรงที่ เราไม่สามารถส่ง โทรสาร จากกระดาษธรรมดาได้ (อาจจะทำได้โดยใช้เครื่องสแกน และสั่งให้ส่งออกทาง โทรสาร)

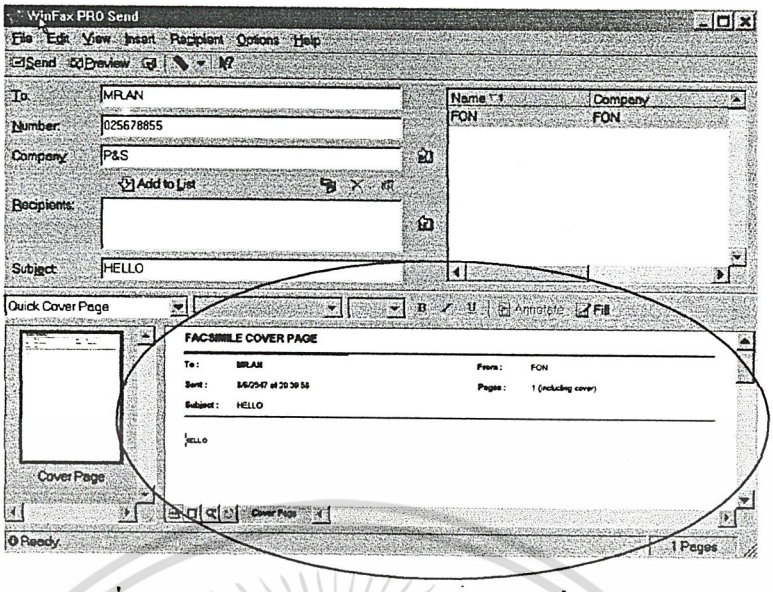


รูปที่ 2.32 แสดงหน้าต่างโปรแกรม Winfax

การส่งโทรสารผ่านโปรแกรม Winfax มีขั้นตอนดังนี้

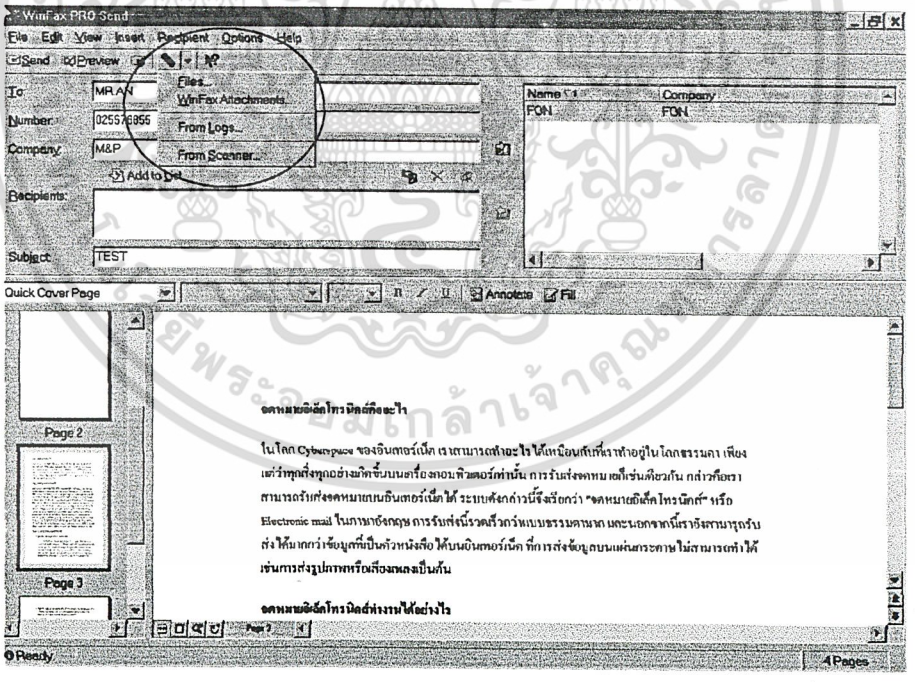
1. พิมพ์ชื่อผู้รับปลายทางในช่อง To
2. พิมพ์หมายเลขโทรสารปลายทางของผู้รับในช่อง NUMBER
3. พิมพ์ชื่อบริษัทปลายทางในช่อง Company
4. พิมพ์ชื่อเรื่องในช่อง Subject
5. พิมพ์ข้อความที่ต้องการส่งในช่องที่ให้พิมพ์ข้อความ ดังรูปที่ 2.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.33 แสดงหน้าต่างโปรแกรม Winfax ส่วนที่เป็นปกหน้า

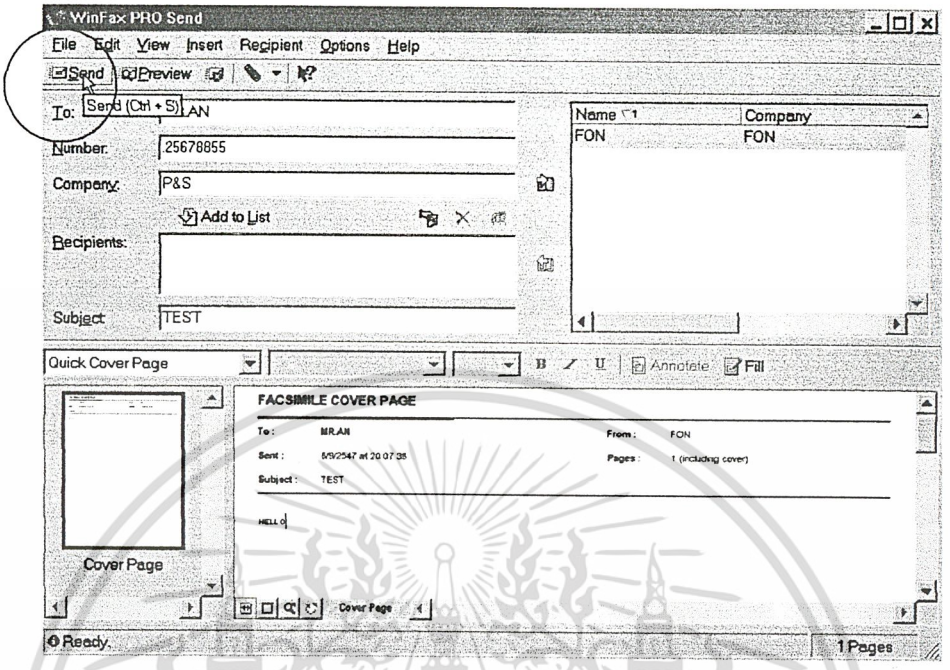
5. ในกรณีที่ข้อมูลที่ต้องการส่งเป็นเอกสารแนบ สามารถทำได้โดยการกดปุ่ม Attatched file ดังรูปที่ 2.33ข้อความในเอกสารแนบจะถูกแสดงในกล่องข้อความ



รูปที่ 2.34 แสดงกล่องข้อความที่เป็นเอกสารแนบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.เมื่อทำการกรอกข้อมูลเรียบร้อยแล้วกดปุ่ม sent เพื่อทำการส่งโทรสารไปยังปลายทาง



รูปที่ 2.35 แสดงหน้าต่างที่พร้อมจะส่งโทรสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้าง

โครงงานนี้พัฒนาขึ้นโดยการเขียนโปรแกรมภาษา ASP (Active Server Pages) และภาษาเดลไฟ เวอร์ชั่น 6 (Borland Delphi Version 6) บนระบบปฏิบัติการไมโครซอฟท์วินโดว์ XP (Microsoft Windows XP) เพื่อติดต่อกับโปรแกรม Winfax

โครงสร้างของโครงงานนี้จะเชื่อมต่อระหว่าง 2 เครือข่าย คือ เครือข่ายอินเตอร์เน็ตและเครือข่าย โทรศัพท์ (PSTN) ผ่านคอมพิวเตอร์ที่มีโปรแกรมที่ทำหน้าที่รับข้อมูลที่ผู้ใช้ต้องการส่งโทรสาร แล้ว ประมวลผลเพื่อติดต่อไปยังเซิร์ฟเวอร์ปลายทางโดยผ่านเครือข่ายอินเตอร์เน็ต จากนั้นเซิร์ฟเวอร์ปลายทาง ก็ส่งข้อมูลต่อไปยังโปรแกรม Winfax เพื่อให้โปรแกรม Winfax ทำหน้าที่ในการเข้ารหัสข้อมูลตาม มาตรฐาน ITU-T (T.4) และส่งข้อมูลต่อไปยังเครื่องโทรสารโดยผ่านโครงข่ายโทรศัพท์(PSTN) ต่อไป

โครงงานนี้ประกอบด้วยโปรแกรมส่วนย่อยต่าง ๆ ดังนี้

1. ส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้
2. ส่วนติดต่อกับเซิร์ฟเวอร์ปลายทาง (เซิร์ฟเวอร์หลัก)
3. ส่วนเซิร์ฟเวอร์ปลายทาง

3.1 ส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้

ส่วนรับข้อมูลจากผู้ใช้ทาง อินเทอร์เน็ท โดยมีหน้าแรกรับ Username และ Password ของผู้ใช้ ถ้ายังไม่เป็นสมาชิกต้องสมัครเป็นสมาชิกก่อนข้อมูลสมาชิกจะถูกเก็บไว้ในฐานข้อมูล เมื่อ Username และ Password ถูกต้องจะอนุญาตให้ใช้ระบบ ในหน้าถัดมาเป็นหน้าให้กรอก Fax Parameter ถ้าค่า Fax Parameter ไม่ถูกต้องจะต้องพิมพ์ใหม่ค่า Fax Parameter จะเก็บไว้ใน Database เพื่อรอการนำไปใช้ต่อไป ส่วนของ loginจะใช้ function Validator ในการตรวจสอบการกรอกข้อมูลว่าได้กรอกข้อมูลหรือไม่ ถ้าไม่กรอกข้อมูลจะแจ้งเตือน

```
function Validator() {
        if (document.FormAddress.mname.value == "") {
         alert("กรุณากรอกข้อมูลลงในช่อง  Member Name  ด้วยครับ.");
         document.FormAddress.mname.focus();
         return (false);
        }
```

จากนั้นจะนำ user name กับ password ไปดูว่ามีในฐานข้อมูลหรือไม่ ถ้ามีจะสามารถเข้าระบบได้ ส่วนการรับสมาชิก ข้อมูลของสมาชิกจะเก็บไว้ในฐานข้อมูล

```
s="insert into member values('" & mname & "','" & password & "','" & fname & "','" & lname
& "','" & address & "','" & tel & "','" & email & "')"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดตัวแปร s เป็นตัวแทรกข้อมูลทั้งหมดลงตาราง

sql="select mname from member where mname='" & Mname & "'"

กำหนดตัวแปร sql เป็นตัวเลือกช่องแรกของข้อมูลที่จะนำลงฐานข้อมูล

set conn=server.createobject("adodb.connection")

conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" & Server.MapPath ("\indb2\member\data.mdb")

สร้างออบเจคต์ connection ที่ชื่อ conn โดยออบเจกต์ จะทำการเรียกเมคถอด open เพื่อติดต่อกับฐานข้อมูล **data.mdb**

set rs=server.createobject("adodb.recordset")

rs.open sql,conn,1,3

สร้างออบเจกต์ RecordSet ที่ชื่อ rs โดยออบเจกต์ rs จะทำการเรียกเมคถอด open เพื่อดึงข้อมูลจากออบเจกต์ conn โดยการเลือก record mname ในฐานข้อมูล data.mdb

if rs.eof then

set rs2=server.createobject("adodb.recordset")

rs2.open s,conn,1,3

ถ้า mname ยังไม่มีใครใช้ ให้สร้างออบเจกต์ RecordSet ที่ชื่อ rs2 แล้วกรอกข้อมูลสมาชิก

ส่วนการส่งโทรสาร เมื่อผู้ใช้ เข้าระบบแล้วจะมาที่หน้าส่งโทรสาร จะให้กรอกข้อมูล ชื่อผู้รับ บริษัท หมายเลขโทรสาร เบอร์มือถือผู้รับ เบอร์มือถือผู้ส่ง หัวข้อ ข้อความ และ ไฟล์แนบ จากนั้นจะทำการเก็บข้อมูลลงฐานข้อมูลทำเช่นเดียวกับการรับสมาชิก แต่ในส่วนนี้จะมีการ Upload file เพิ่มเข้ามา การ การ Upload file ใช้ออบเจกต์ SoftArtisans.FileUp

Set oFileUp = Server.CreateObject("SoftArtisans.FileUp")

โดยจะเก็บชื่อไฟล์ไว้ในฐานข้อมูล ซึ่งจะเป็นตัวชี้ที่อยู่ของไฟล์

หน้าของผู้ดูแลระบบ ผู้ดูแลระบบสามารถเพิ่ม จำนวนการส่งของแต่ละผู้ใช้ว่าผู้ใช้แต่ละคนสามารถส่งได้กี่ครั้ง อีกทั้งยังสามารถดูข้อมูลที่เคยส่งไปแล้วทั้งหมด และรายชื่อสมาชิกทั้งหมดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.1 แสดงผังการทำงานของส่วนติดต่อกับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนเซิร์ฟเวอร์หลัก

ส่วนที่เซิร์ฟเวอร์หลัก ทำหน้าที่ในการตรวจสอบข้อมูลในฐานข้อมูล และส่งข้อมูลต่าง ๆ ที่รับมาจากผู้ใช้งานไปยังเซิร์ฟเวอร์ปลายทาง โดยในการส่งเอกสารแนบนั้นจะใช้โพรโตคอล FTP

การทำงานจะเริ่มโดยการตรวจสอบข้อมูลในฐานข้อมูล ถ้ามีข้อมูลเข้ามาใหม่จะทำการเก็บค่าพารามิเตอร์ของโทรสาร ข้อความและเอกสารแนบ จากนั้นจะทำการวิเคราะห์เลขหมายปลายทางแล้วทำการติดต่อไปยังเซิร์ฟเวอร์ปลายทางเพื่อส่งพารามิเตอร์ของโทรสาร ข้อความและเอกสารแนบ เพื่อให้ดำเนินการในขั้นตอนต่อไป

ตัวอย่างการเขียนโปรแกรมในส่วนเซิร์ฟเวอร์หลัก

ส่วนแรกคือการดึงข้อมูลมาจากตารางฐานข้อมูล ผ่านทาง Borland Database Engine (BDE) โดยจะใช้คอมโพเนนท์ Table คลาส TTable ที่มีอยู่แล้วในเดลไฟ ตัวอย่างสำหรับคำสั่งที่ใช้ในส่วนของการดึงข้อมูลมีดังนี้

table1.fieldbyname('faxnumber').asstring; ใช้สำหรับการดึงข้อมูลในฟิลด์ faxnumber มาจากตาราง table1 โดยมีการกำหนดชนิดของข้อมูลประเภท String

table1.Close; ใช้สำหรับการสั่งให้ปิดตาราง table1

table1.open; ใช้สำหรับการสั่งให้เปิดตารางขึ้นมาใช้งาน

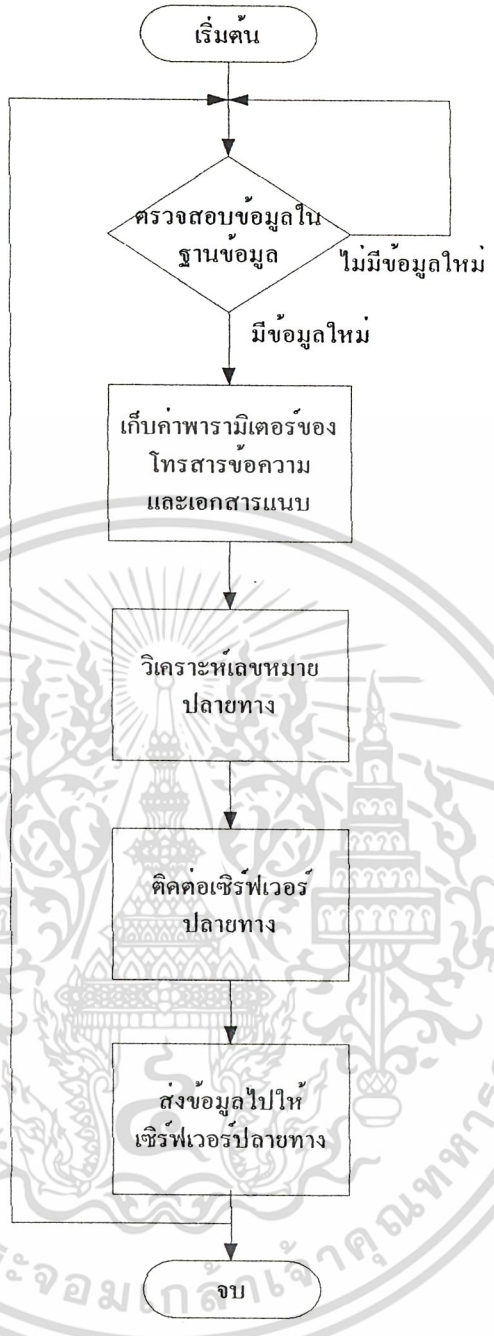
table1.delete; ใช้สำหรับการสั่งลบเรคคอร์ดปัจจุบัน

ส่วนที่สอง เป็นส่วนส่งไฟล์ข้อมูล จะใช้คอมโพเนนท์ NMFTP คลาส TNMFTP ที่สำหรับการเรียกใช้งานโปรโตคอล FTP ในการส่งไฟล์ข้อมูลไปยังเครื่องผู้รับเรียกว่าเครื่อง FTP Server โดยเริ่มต้นจะต้องระบุแอดเดรสของเครื่องโฮสต์ปลายทางก่อน แล้วตามด้วย User ID และ password สุดท้ายต้องทำการสร้างการเชื่อมต่อด้วยคำสั่ง NMFTP1.Connect;

```
NMFTP1.Host := 'host address';
NMFTP1.UserID := 'user id';
NMFTP1.Password := 'password';
NMFTP1.Connect;
```

ส่วนสุดท้ายคือส่วนที่ใช้สำหรับการส่งข้อความ จะใช้คอมโพเนนท์ ServerSocket คลาส TServerSocket โดยเครื่องที่ใช้รับข้อมูลจะใช้คอมโพเนนท์ ClientSocket การทำงานเริ่มโดยเครื่องเซิร์ฟเวอร์รอฟัง (listen) สัญญาร้องขอการติดต่อด้วยโปรโตคอล TCP/IP (TCP/IP connections) จากเครื่องไคลเอนท์ และจะทำการสร้างการเชื่อมต่อ (connection) เมื่อได้รับสัญญาณร้องขอแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.2 แสดงผังการทำงานส่วนที่ทำหน้าที่ในการติดต่อกับเซิร์ฟเวอร์ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

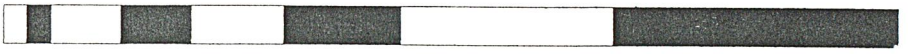
3.3 ส่วนเซิร์ฟเวอร์ปลายทาง

ในส่วนนี้จะทำการรับค่า Fax Parameter และรับข้อความหรือเอกสารแนบที่ต้องการส่งจากเซิร์ฟเวอร์หลัก เมื่อรับข้อมูลมาแล้วเซิร์ฟเวอร์ปลายทางก็จะทำการจัดรูปแบบข้อความที่รับมาได้ จากนั้นจะทำการเข้ารหัส และติดต่อไปยังโมเด็มเพื่อส่งโทรสารไปยังปลายทางต่อไป ซึ่งสามารถอธิบายโดยละเอียดดังนี้

1. เซิร์ฟเวอร์หลักส่งข้อมูลมาให้
2. เซิร์ฟเวอร์ปลายทางนี้จะทำการวิเคราะห์หา Fax Parameter ข้อความและชื่อเอกสารแนบ จากข้อมูลที่เซิร์ฟเวอร์หลักส่งมาให้ โดย Fax Parameter ประกอบด้วย
   2.1 Fax number หมายถึงหมายเลขโทรสารปลายทาง
   2.2 Send date time หมายถึงเวลาที่ทำการส่งโทรสาร
   2.3 Recipient name หมายถึงชื่อผู้รับโทรสาร
   2.4 Recipient company หมายถึงชื่อบริษัทของผู้รับโทรสาร
   2.5 Subject หมายถึง หัวข้อเรื่องที่ส่ง
3. ส่วนข้อความที่ต้องการส่ง ข้อความนั้นสามารถเป็นได้หลายรูปแบบ เช่น ข้อความเอกสารเวิร์ด ข้อความที่เป็นรูปภาพ เป็นต้น หรือผู้ส่งสามารถพิมพ์ข้อความที่ต้องการส่ง ในหน้าต่างข้อความ (Message)
4. นำค่า Fax Parameter และข้อความจัดรูปแบบ โดยแบ่งเป็นส่วนของปกหน้า และส่วนของข้อความ
5. ทำการเข้ารหัสข้อมูล โดยรูปแบบข้อมูลจะใช้มาตรฐานการติดต่อโทรสารที่กำหนดโดย ITU-T(T.4) มาตรฐาน T.4 มีลักษณะการเข้ารหัสแบบ 1 มิติ (Modified Huffman)และแบบ 2 มิติ (Modified Read) ซึ่งในโครงงานนี้ใช้การเข้ารหัสแบบ 1 มิติ หลักการเข้ารหัสจะทำได้โดยใช้หลักการ Run Length Code ซึ่งสามารถเทียบการเข้ารหัสจากตารางมาตรฐานที่ กำหนดโดย ITU-T(แสดงไว้ในภาคผนวก) ซึ่งมีการแบ่งเป็น 2 ประเภท คือ Make-up code และ Terminating code โดยMake-up code จะมีจำนวนพิกเซลของข้อมูลมากกว่า 63 พิกเซลขึ้นไป และTerminating code จะมีจำนวนพิกเซลของข้อมูล 0- 63 พิกเซล โดยถ้ามีจำนวนพิกเซลมากกว่า 64 พิกเซล จะต้องส่งข้อมูลของ Make-up code ก่อนแล้วจึงส่งข้อมูลจำนวนพิกเซลของข้อมูล Terminating code ตามไปจนครบจำนวนพิกเซล

ตัวอย่างของการเข้ารหัสสามารถอธิบายได้ดังนี้ สมมติมีข้อมูลภาพ 1 เส้น ยาว 1728 พิกเซล ซึ่งประกอบด้วยพิกเซลต่าง ๆ เรียงตามภาพ จะสามารถแบ่งได้ตามมาตรฐานเป็น Make-up codeและ Terminating code และเมื่อเทียบกับตารางรหัสแล้วสามารถแปลงเป็นไบนารี่ (Binary) ได้ดังรูป จากนั้นต้องตามด้วยรหัส EOL จะได้ข้อมูลครบ 1 เส้น โดยก่อนการส่งข้อมูลจะมีการสลับบิตจาก MSB เป็น LSB ของทุก ๆ ไบต์ข้อมูล เนื่องจากการส่งข้อมูลของโมเด็มจะส่งข้อมูลจาก MSB ก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนพิกเซล	31 33 103 100 127 169 328 837
ภาพ 1 เส้น	
สี/พิกเซล	W31 + B33 + W64+W39 + B64+B36 + W64+W63 + B128+B41 +W320+W8 + B832+B5
MH Code	00011010 000001101011 11011 00101000 0000001111 000011010100 11011 00110100 000011001000 000001101101 00110110 10011 0000001001101 0000001111

รูปที่ 3.3 ตัวอย่างการเข้ารหัสข้อมูลตามมาตรฐาน ITU-T (T.4) มีลักษณะการเข้ารหัสแบบ 1 มิติ

6. ทำการติดต่อกับโมเด็ม ละสั่งงานให้โมเด็มทำการหมุนโทรศัพท์ไปยังเครื่องโทรสารปลายทางและส่งข้อมูลที่ได้ทำการเข้ารหัสแล้ว ไปยังเครื่องโทรสารปลายทางโดยใช้คำสั่ง AT ในการสั่งให้โมเด็มทำงาน โดยใช้มาตรฐานคำสั่ง AT FAX Class 2 การทำงานของโมเด็มจะต้องเป็นไปตามมาตรฐานของ ITU-T T.30 ซึ่งมีการแบ่งการติดต่อกันของโทรสารกลุ่ม 3 ออกเป็นขั้นตอนต่างๆ 5 เฟส เฟส A – เฟส E

โดยเมื่อสั่งให้โมเด็มทำงาน โมเด็มจะทำการส่งสัญญาณตามมาตรฐาน T.30 ออกไปติดต่อกับเครื่องโทรสารปลายทาง และเมื่อโทรสารปลายทางส่งสัญญาณตอบรับกลับมายังโมเด็ม โมเด็มจะส่งข้อมูลตอบรับมาหาคอมพิวเตอร์ในรูปแบบของ คำสั่ง AT ตอบรับมายังเครื่องคอมพิวเตอร์

เฟส A : การเริ่มติดต่อ จะเป็นการสั่งโมเด็มให้หมุนหมายเลขโทรศัพท์ไปยัง โทรสารปลายทางด้วยคำสั่ง ATDT แล้วตามด้วยหมายโทรศัพท์ไปยัง โทรสารปลายทาง เมื่อปลายทางรับสายจะส่งสัญญาณ CNG Beep ไปยังปลายทาง ปลายทางจะตอบรับพร้อมส่งสัญญาณ CED (Called Station Identification) กลับมา โมเด็มจะตอบรับพร้อมส่งข้อมูล +FCON มายังคอมพิวเตอร์

เฟส B : ขั้นตอนก่อนส่งข้อมูล หลังจากทำการติดต่อกัน ได้แล้วจะทำการตรวจสอบคุณสมบัติต่างในการรับและส่งข้อมูลให้ตรงกัน โทรสารจะทำการแจ้งสถานะและตรวจสอบสถานะต่างๆ โดยโมเด็มจะส่งชื่อเครื่องโทรสาร ด้วยคำสั่ง AT+FDID= " string" และส่งสัญญาณ DIS (Digital Identification Signal) เช่น AT+FDIS = 0,5,0,2,0,0,0,4 เมื่อปลายทางได้รับสัญญาณ จะส่ง DCS (Digital Command Signal) กลับมาเพื่อแจ้งว่าจะสามารถใช้คุณสมบัติชนิดใดบ้าง โมเด็มก็จะทำการตอบกลับ +FDCS: โทรสารด้านผู้ถูกเรียกจะส่งสัญญาณความเร็วสูงที่เรียกว่า เทรนนิ่ง (Training) เพื่อตรวจสอบระบบแล้วส่ง CFR (Confirmation to Request) กลับมาเพื่อยืนยันความพร้อมในการรับข้อมูลเพื่อพร้อมรับข้อมูลแล้ว โมเด็มจะส่งข้อมูล ติดต่อมายังคอมพิวเตอร์

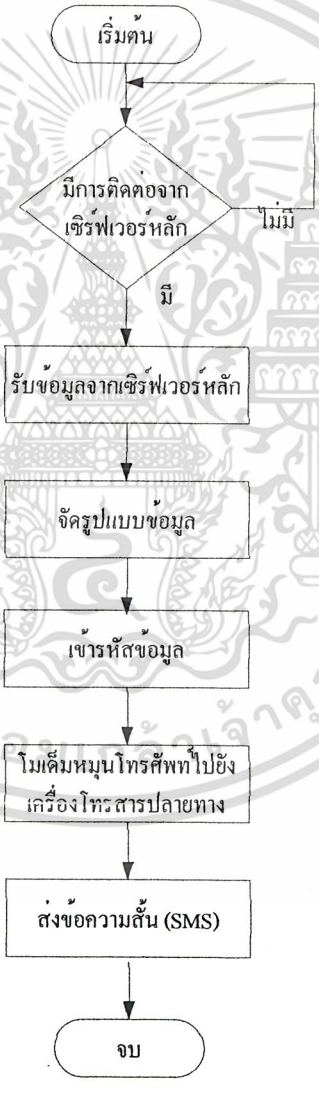
เฟส C : การส่งข้อมูล ทำการส่งข้อมูล ทำการส่งข้อมูลออกไปหลังจากได้รับสัญญาณติดต่อจากโมเด็มแล้วจะทำการส่งข้อมูล ที่ได้เตรียมไว้แล้วไปยังเครื่องโทรสารปลายทาง และหลังจากข้อมูลถูกส่งไปครบทั้งหมดแล้วจะตามด้วย <DCE><ETX> เพื่อแจ้งให้ปลายทางทราบว่าจบข้อมูลที่ต้องการส่งแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟส D : หลังจากส่งข้อมูลเรียบร้อยแล้วด้านผู้เรียกจะทำการส่ง RTC (Return to Control) และส่ง EOP (End of Procedure) เพื่อแสดงว่าจบการทำงาน ด้วยคำสั่ง AT+FET = 2 และรอรับสัญญาณ MCF (Message Confirmation) ซึ่งโมเด็มจะส่งข้อมูลกลับมา +FPTS:1 หมายความว่าเอกสารที่ได้ปรกติ ไม่มีข้อผิดพลาด

เฟส E : การเลิกการติดต่อ การยกเลิกการติดต่อระหว่างโทรการ โดยส่งสัญญาณ DCN (Disconnect) เพื่อทำการวางสาย และรอสัญญาณตอบกลับจากโมเด็ม +FHNG: 0 เป็นการสิ้นสุดการทำงานส่งโทรสารของส่วนติดต่อกับโมเด็ม

7. ทำการส่งข้อความสั้น (SMS) ไปยังผู้ส่งเพื่อแจ้งสถานะของการส่งโทรสาร และส่งไปยังผู้รับว่ามีโทรสารถูกส่งมาให้ผู้รับ โดยการส่งข้อความสั้นนั้นจะใช้ AT command ในการติดต่อกับโมเด็ม



รูปที่ 3.4 ผังการทำงานส่วนเซิร์ฟเวอร์ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 การทำงานของระบบ

การเริ่มต้นการทำงาน ผู้ใช้จะต้องเป็นสมาชิกของระบบ จากนั้นต้องกำหนดรายละเอียดต่าง ๆ ให้ครบถ้วน เช่นกรอกรายชื่อผู้รับ เบอร์โทรสารของผู้รับ ข้อความที่ต้องการส่ง ระบุเอกสารแนบที่ต้องการส่ง เป็นต้น เมื่อผู้ใช้งานกรอกรายละเอียดต่าง ๆ ครบแล้ว จะต้องกดปุ่ม SEND เพื่อส่งข้อมูลให้ระบบดำเนินการต่อไป

ระบบจะรับข้อมูลที่เป็นรายละเอียดต่าง ๆ เกี่ยวกับการส่งโทรสารจากผู้ใช้มาเก็บไว้ในฐานข้อมูลของระบบที่เซิร์ฟเวอร์หลัก (Main Server) จากนั้นระบบจะดึงข้อมูลจากฐานข้อมูลมาตรวจสอบหมายเลขโทรสารของผู้รับ แล้ววิเคราะห์ว่าต้องส่งข้อมูลเหล่านั้นไปที่เซิร์ฟเวอร์ปลายทางใด เช่น หมายเลขโทรสารของผู้รับปลายทางคือ 053123456 ระบบจะส่งข้อมูลไปที่เซิร์ฟเวอร์ปลายทางรหัส 053 (รหัสพื้นที่ของจังหวัดเชียงราย เชียงใหม่ ลำพูน และแม่ฮ่องสอน)

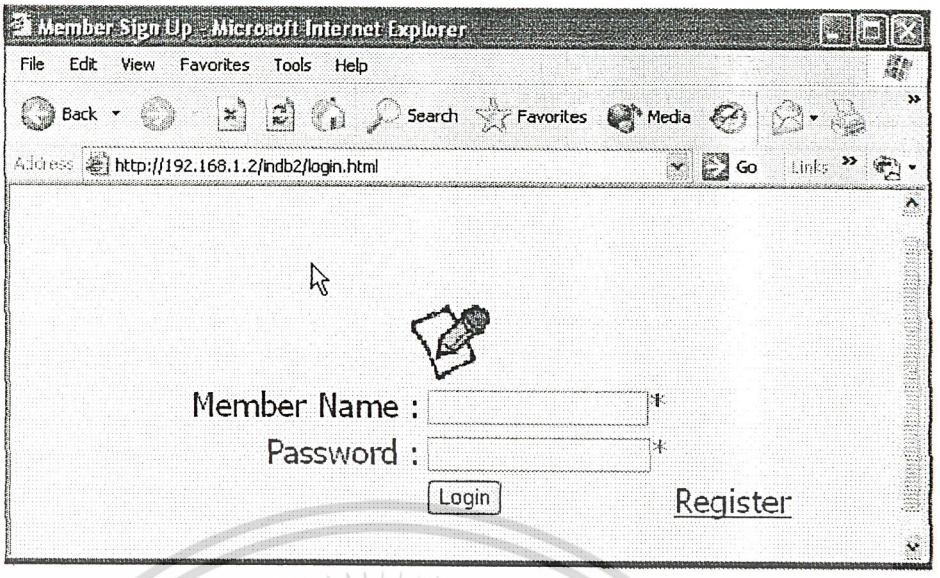
หลังจากที่หาเซิร์ฟเวอร์ปลายทางได้แล้ว เซิร์ฟเวอร์หลักก็จะเริ่มทำการติดต่อกับเซิร์ฟเวอร์ปลายทางนั้น เมื่อเซิร์ฟเวอร์ปลายทางได้รับการติดต่อจากเซิร์ฟเวอร์หลักก็จะส่งสัญญาณตอบรับกลับมายังเซิร์ฟเวอร์หลัก เซิร์ฟเวอร์หลักก็จะเริ่มต้นการส่งข้อมูลออกไปเซิร์ฟเวอร์ปลายทาง

เมื่อเซิร์ฟเวอร์ปลายทางได้รับข้อมูลจากเซิร์ฟเวอร์หลัก ก็จะทำการจัดรูปแบบข้อมูลเพื่อส่งไปเข้ารหัส และติดต่อกับโมเด็มเพื่อให้โมเด็มทำหน้าที่หมุนโทรศัพท์เพื่อส่งข้อมูลไปยังเครื่องโทรสารปลายทางต่อไป

4.2 การทดลองใช้งานส่วน ติดต่อกับผู้ใช้งาน

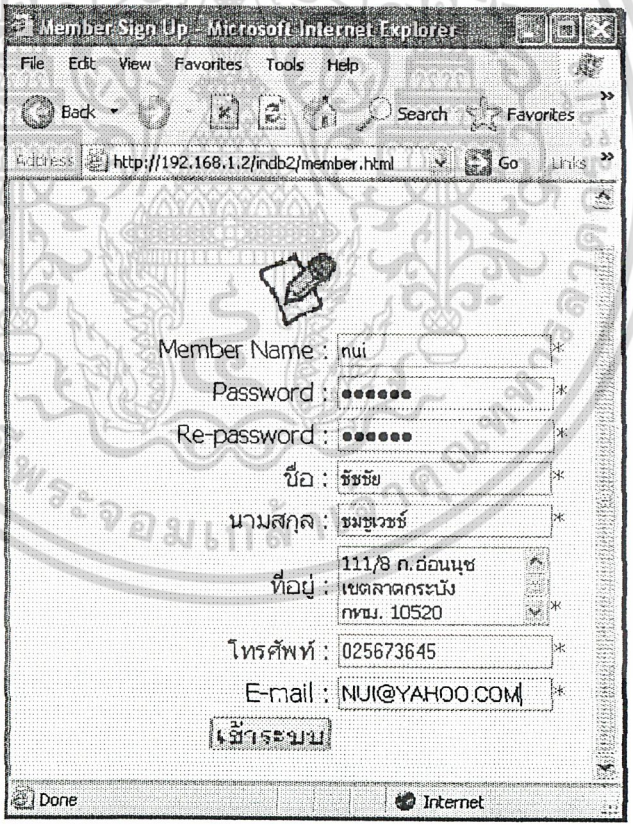
โปรแกรมส่วนนี้เขียนขึ้นโดยใช้ ASP และใช้ Microsoft Access เป็นฐานข้อมูลในการเก็บข้อมูลที่เกี่ยวข้องกับการส่งโทรสาร สมมติว่านายชัชชัยต้องการส่งโทรสารไปเชียงใหม่ ที่หมายเลขโทรสาร 053123456 นายชัชชัยต้อง ทำการกรอกชื่อสมาชิกและรหัสผ่านก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงหน้าLogin เพื่อให้ผู้ใช้กรอก Member name และ Password

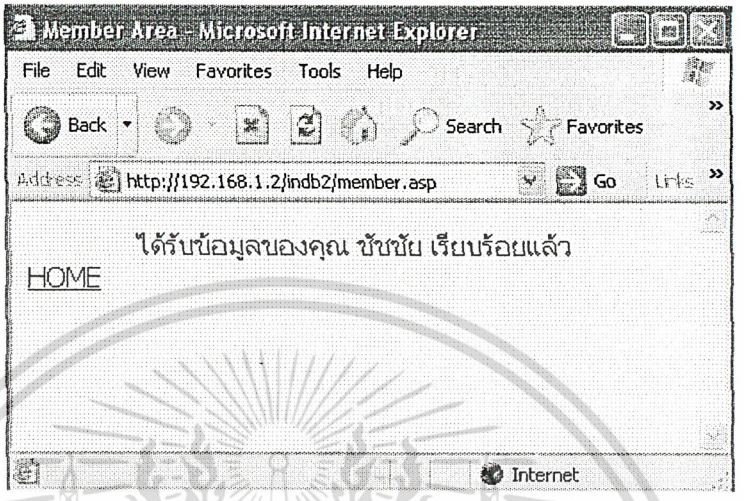
ในกรณีที่ผู้ใช้งานยังไม่ได้ทำการลงทะเบียนเพื่อเป็นสมาชิกให้กดปุ่ม Register เพื่อทำการลงทะเบียน ซึ่งจะปรากฏหน้าลงทะเบียนดังรูปที่ 4.2



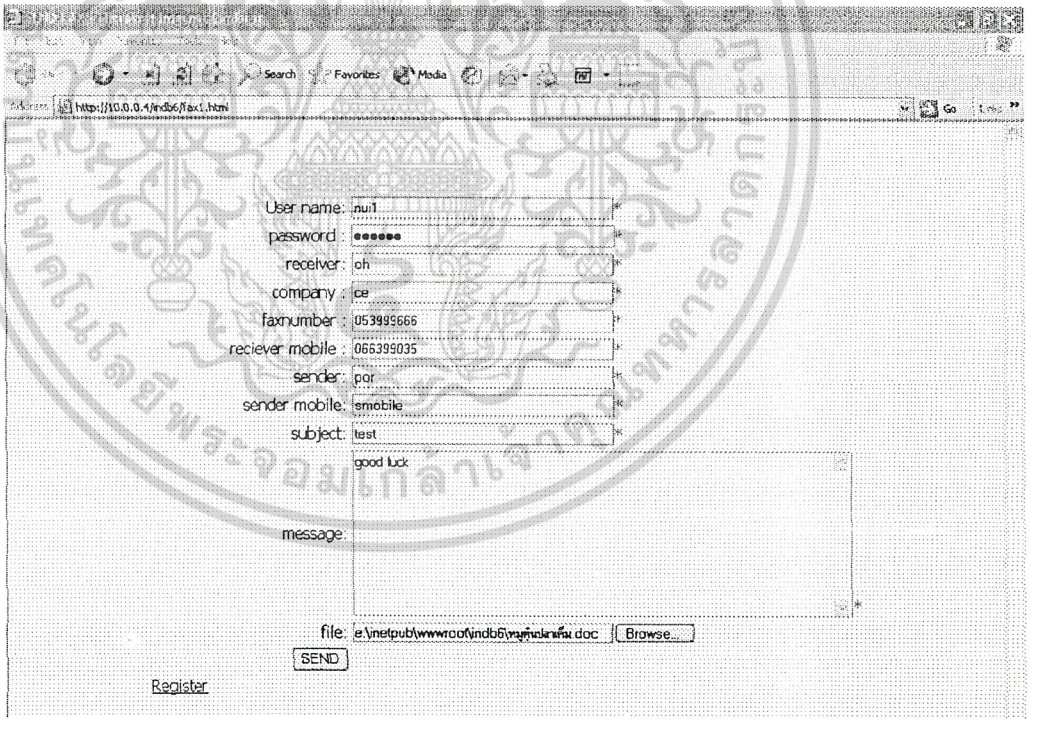
รูปที่ 4.2 แสดงหน้าต่างลงทะเบียนสมาชิกใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการลงทะเบียนสมาชิกใหม่เสร็จเรียบร้อยแล้วกดปุ่ม เข้าระบบ เพื่อเข้าสู่หน้า Login อีกครั้ง หลังจากนั้นก็กรอก Member name และ Password เพื่อเข้าสู่หน้ากรอกรายละเอียดต่าง ๆ ในการส่งโทรสาร ดังแสดงในรูปที่ 4.4



รูปที่ 4.3 แสดงหน้าต่างเมื่อระบบได้รับข้อมูลการลงทะเบียนจากผู้ใช้งาน

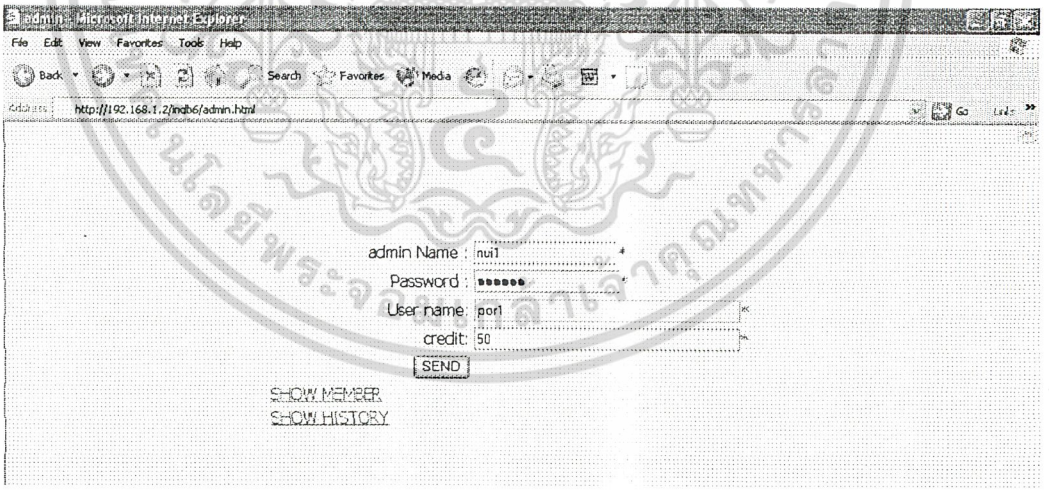


รูปที่ 4.4 แสดงหน้าต่างสำหรับกรอกรายละเอียดต่างๆ ในการส่งโทรสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

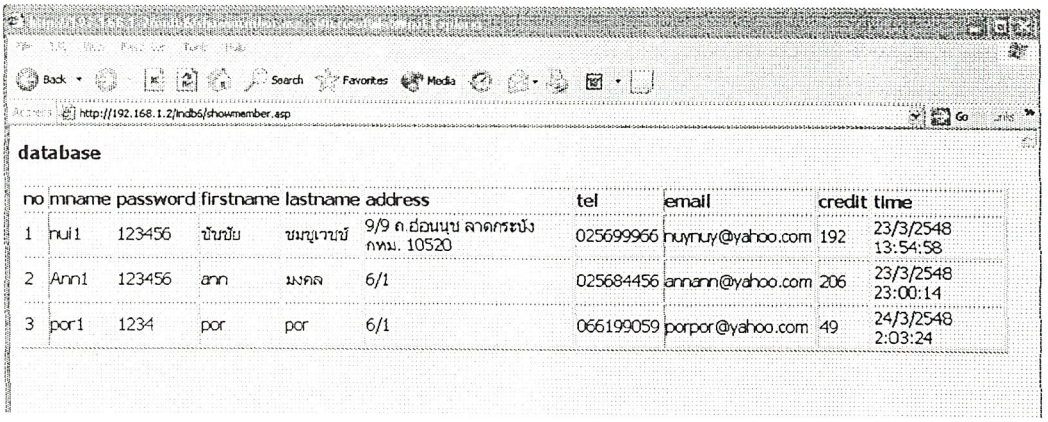
เมื่อกรอกรายละเอียดๆ ในการส่งโทรสารเสร็จแล้วกดปุ่ม Send เพื่อให้ระบบทำการส่งข้อมูลเก็บไว้ในฐานข้อมูลดังรูป 4.6 เมื่อผู้ใช้ส่งโทรสารแล้วจะแสดงข้อความว่าได้ส่งข้อมูลต่าง ๆ เข้าสู่ระบบเรียบร้อยแล้ว ระบบจะทำการรับข้อมูลต่าง ๆ มาเก็บไว้ฐานข้อมูลของระบบ

รูปที่ 4.5 แสดงหน้าต่างที่ระบบจะแสดงข้อความว่าได้ส่งข้อมูลต่าง ๆ เข้าสู่ระบบเรียบร้อยแล้ว



รูปที่ 4.6 แสดงหน้าผู้ดูแลระบบสามารถเพิ่ม credit ให้ผู้ใช้ที่ร้องขอ

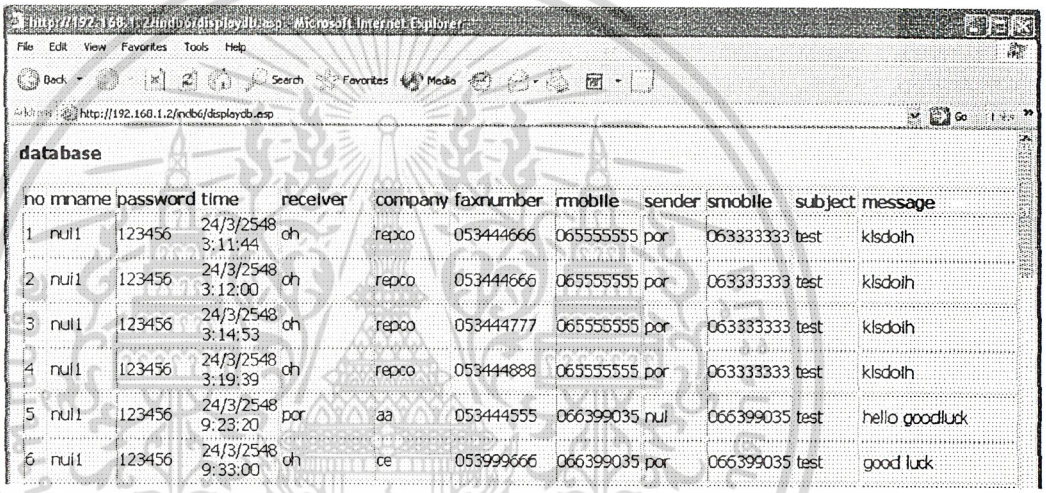
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



database

no	mname	password	firstname	lastname	address	tel	email	credit	time
1	nui1	123456	ขัขยัย	ชมชูเวชช์	9/9 ถ.อ่อนนุช ลาดกระบัง กทม. 10520	025699966	nuynuy@yahoo.com	192	23/3/2548 13:54:58
2	Ann1	123456	ann	มงคล	6/1	025684456	annann@yahoo.com	206	23/3/2548 23:00:14
3	por1	1234	por	por	6/1	066199059	porpor@yahoo.com	49	24/3/2548 2:03:24

รูปที่ 4.7 แสดงรายชื่อและข้อมูลของสมาชิกทั้งหมด



database

no	mname	password	time	receiver	company	faxnumber	rmobile	sender	smobile	subject	message
1	nui1	123456	24/3/2548 3:11:44	oh	repco	053444666	065555555	por	063333333	test	klsdoih
2	nui1	123456	24/3/2548 3:12:00	oh	repco	053444666	065555555	por	063333333	test	klsdoih
3	nui1	123456	24/3/2548 3:14:53	oh	repco	053444777	065555555	por	063333333	test	klsdoih
4	nui1	123456	24/3/2548 3:19:39	oh	repco	053444888	065555555	por	063333333	test	klsdoih
5	nui1	123456	24/3/2548 9:23:20	por	aa	053444555	066399035	nui	066399035	test	hello goodluck
6	nui1	123456	24/3/2548 9:33:00	oh	ce	053999666	066399035	por	066399035	test	good luck

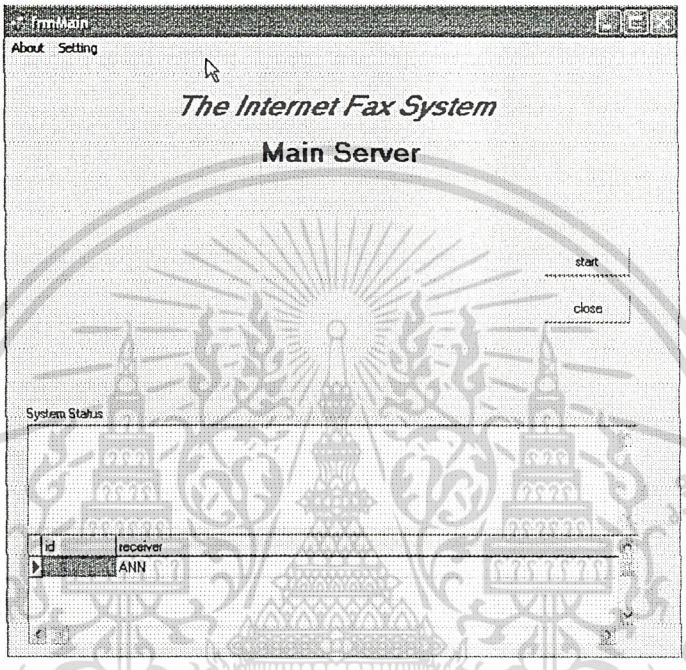
รูปที่ 4.8 แสดงหน้าของข้อมูลที่เคยส่งทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองการใช้งานส่วนเซิร์ฟเวอร์กลาง

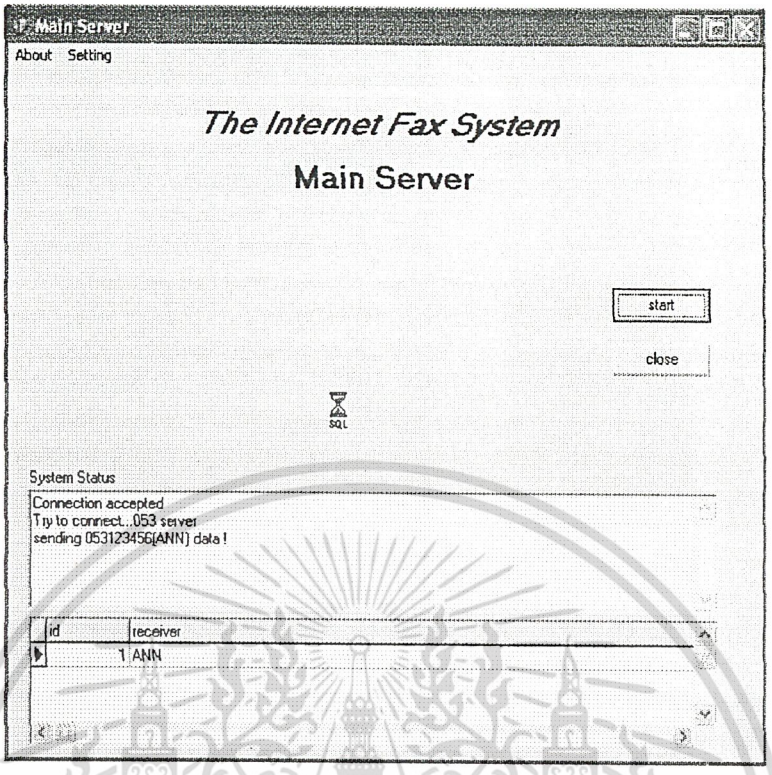
โปรแกรมในส่วนนี้เขียนขึ้นโดยภาษาเดลไฟเวอร์ชั่น 6 เริ่มต้นระบบทำการตรวจสอบจำนวนข้อมูลจากฐานข้อมูล จากนั้นระบบจะดึงข้อมูลจากฐานข้อมูลมาตรวจสอบหมายเลขโทรสารของผู้รับ แล้ววิเคราะห์และส่งข้อมูลเหล่านั้นต่อไปที่เซิร์ฟเวอร์ปลายทาง

โปรแกรมจะดึงฐานข้อมูลที่นายชัชชัยส่งมา แล้วทำการตรวจสอบหมายเลขโทรสารของผู้รับ แล้ววิเคราะห์และส่งข้อมูลเหล่านั้นต่อไปที่เซิร์ฟเวอร์ปลายทาง



รูปที่ 4.9 แสดงหน้าต่างของเซิร์ฟเวอร์หลักขณะที่มีข้อมูลในฐานข้อมูล 1 เรคอร์ด

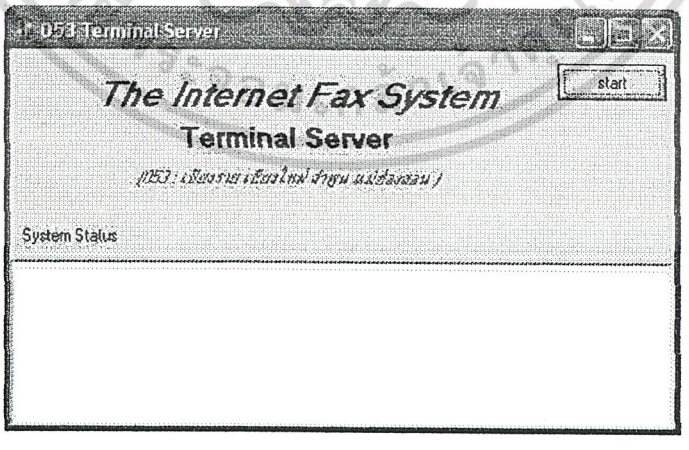
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงหน้าต่างของเซิร์ฟเวอร์หลักขณะทำการส่งข้อมูลไปยังเซิร์ฟเวอร์ปลายทาง

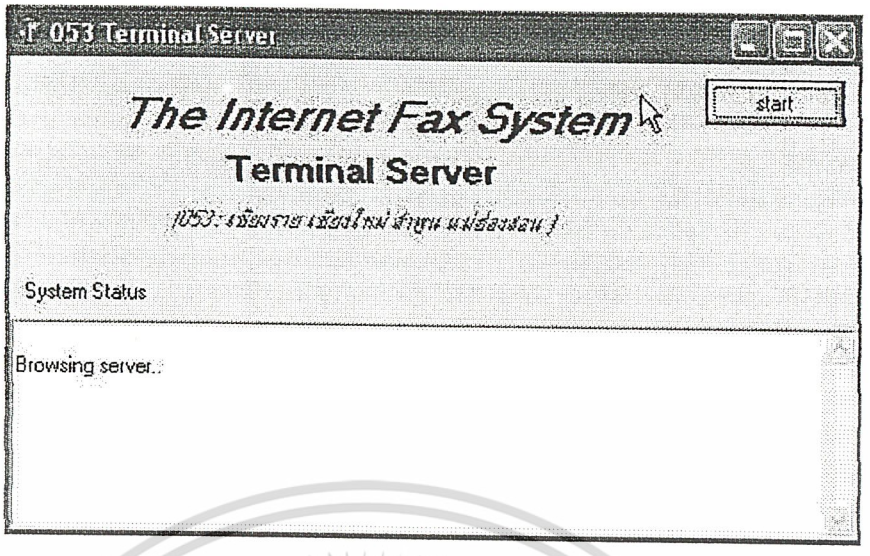
4.4 การทดลองการใช้งานส่วนเซิร์ฟเวอร์ปลายทาง

เริ่มต้นทำงานระบบจะตรวจสอบสัญญาณจากเซิร์ฟเวอร์หลัก เมื่อเซิร์ฟเวอร์ปลายทางได้รับการติดต่อจากเซิร์ฟเวอร์หลักก็จะส่งสัญญาณตอบรับกลับไปยังเซิร์ฟเวอร์หลัก และสร้างการเชื่อมต่อกับเซิร์ฟเวอร์หลัก

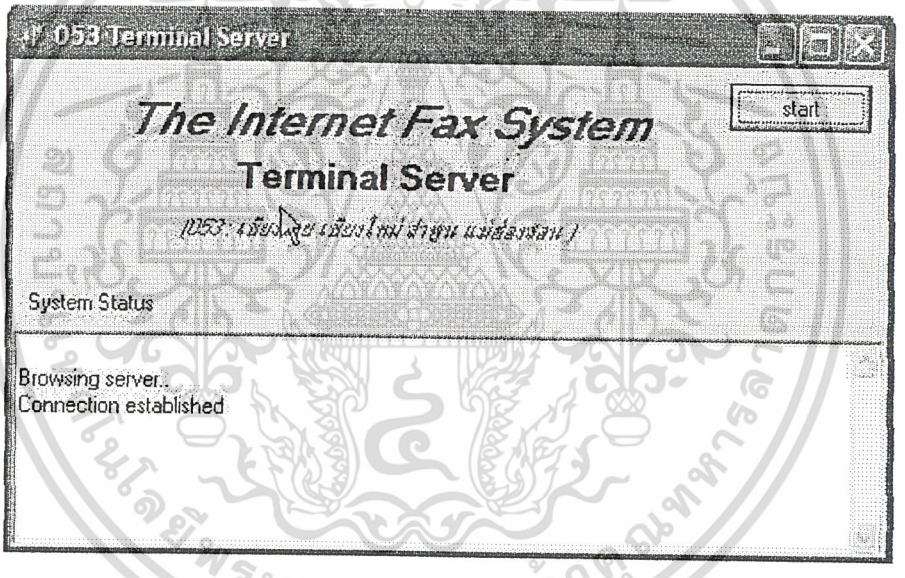


รูปที่ 4.11 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

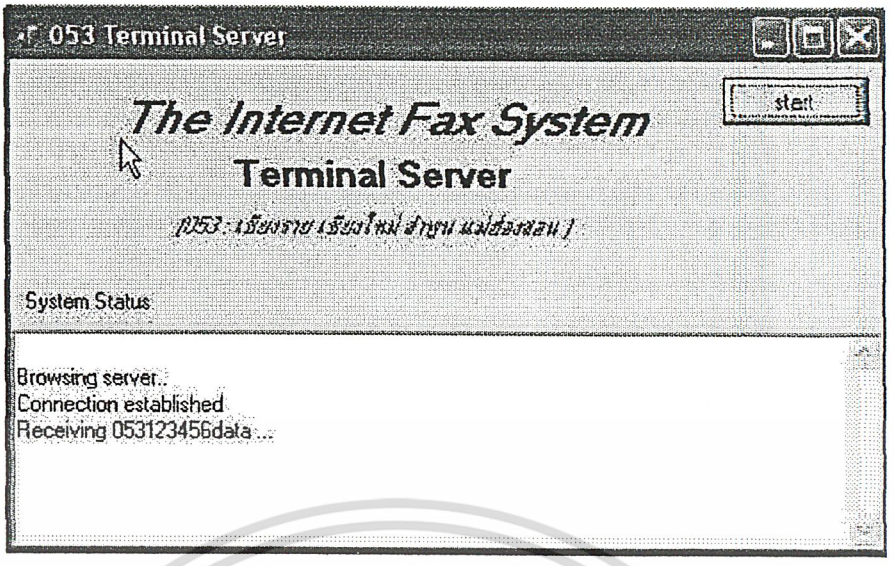


รูปที่ 4.12 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทางขณะรอการติดต่อจากเซิร์ฟเวอร์หลัก



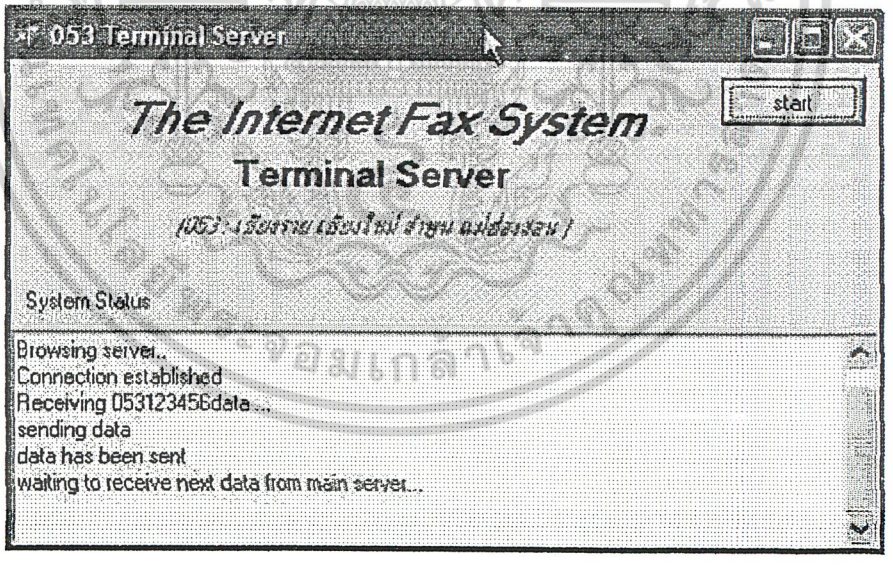
รูปที่ 4.13 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทางขณะสร้างการเชื่อมต่อกับเซิร์ฟเวอร์หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



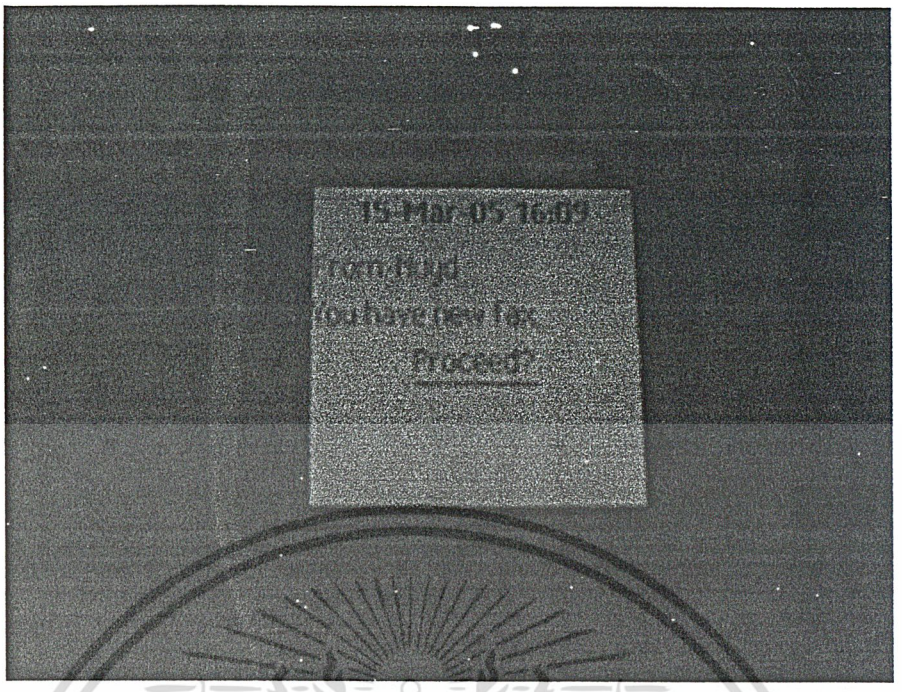
รูปที่ 4.14 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทางขณะรับข้อมูลจากเซิร์ฟเวอร์หลัก

เมื่อเซิร์ฟเวอร์ปลายทางได้รับข้อมูลจากเซิร์ฟเวอร์หลัก ก็จะทำการจัดรูปแบบข้อมูล และเข้ารหัสข้อมูลและสุดท้ายก็จะทำการติดต่อกับโมเด็มเพื่อทำการส่งโทรสารไปยังปลายทางต่อไป

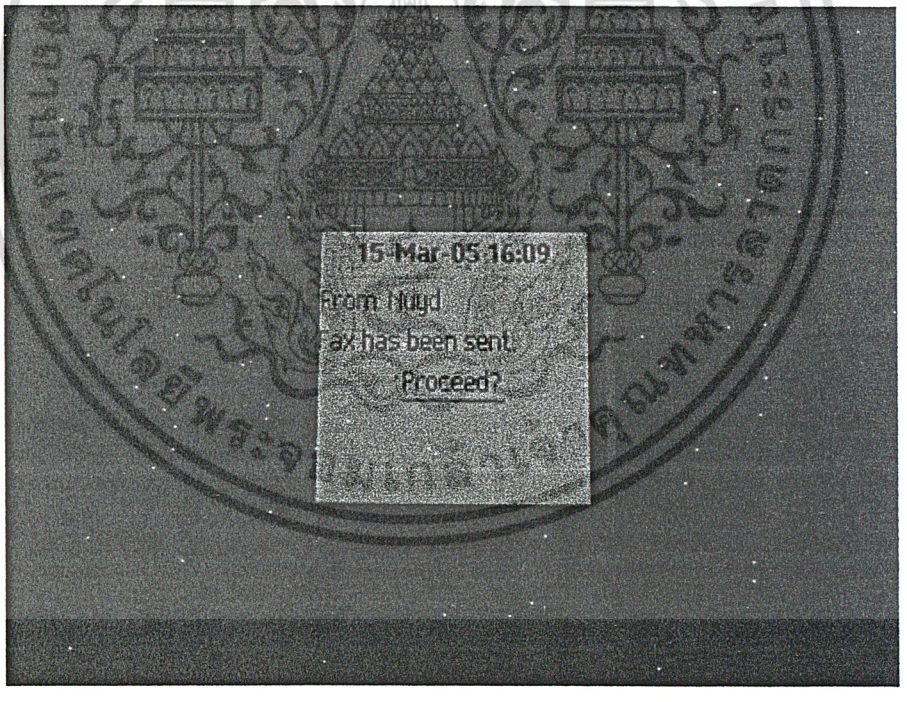


รูปที่ 4.15 แสดงหน้าต่างของเซิร์ฟเวอร์ปลายทางขณะรอรับข้อมูลชุดต่อไปจากเซิร์ฟเวอร์หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.16 แสดง SMS ที่ผู้รับโทรสารได้รับ

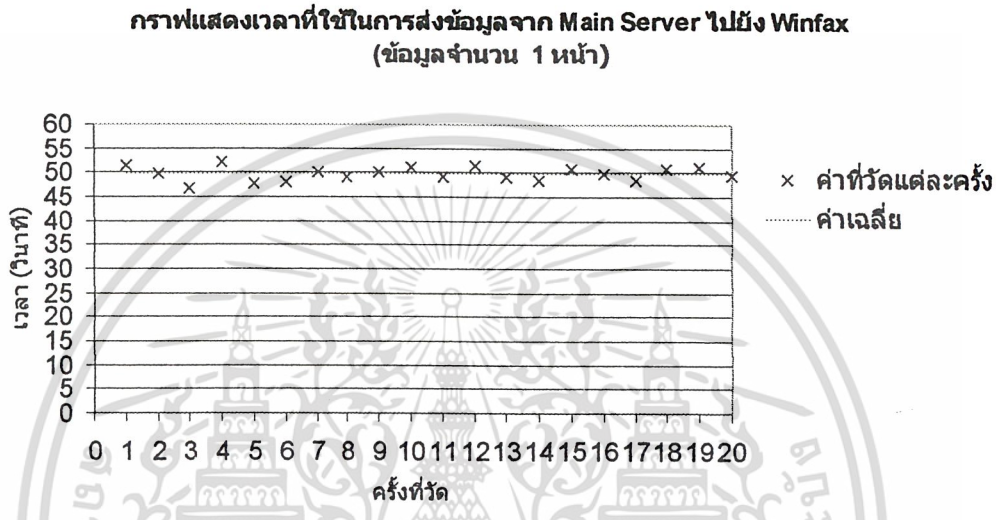


รูป 4.17 แสดง SMS ที่ผู้ส่งโทรสารได้รับ

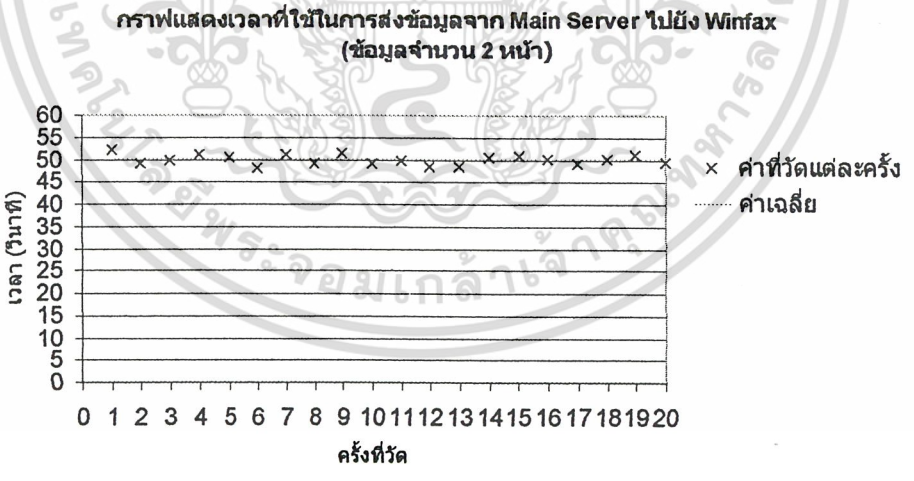
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองวัดค่าประสิทธิภาพของระบบ

จากการทดลองวัดค่าประสิทธิภาพของระบบ โดยวัดเวลาในการส่งข้อมูลจากเซิร์ฟเวอร์หลักถึงโมเด็มที่เครื่องเซิร์ฟเวอร์ปลายทาง โดยข้อมูลที่ทำการส่งมีขนาด (จำนวนหน้า) ที่แตกต่างกัน ตั้งแต่ข้อมูลจำนวน 1 หน้าไปถึงข้อมูลจำนวน 20 หน้า พบว่าค่าเวลาที่ใช้ในการส่งข้อมูลจากเซิร์ฟเวอร์หลักถึงโมเด็มที่เครื่องเซิร์ฟเวอร์ปลายทางที่วัดได้มีค่าเฉลี่ยเท่ากับ 49.55 วินาที หมายความว่าเวลาที่ใช้ในการให้บริการสำหรับการส่งข้อมูลแต่ละครั้งมีค่าประมาณ 50 วินาที โดยผลการทดลองเป็นดังนี้

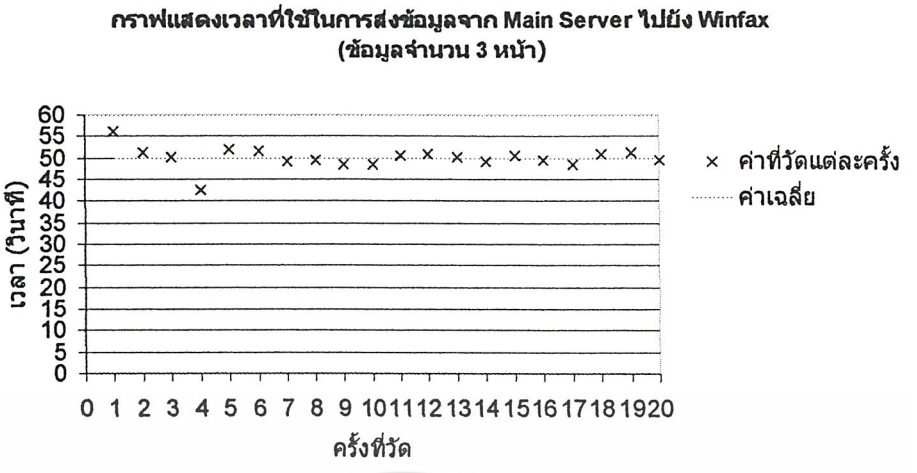


รูปที่ 4.18 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 1 หน้า

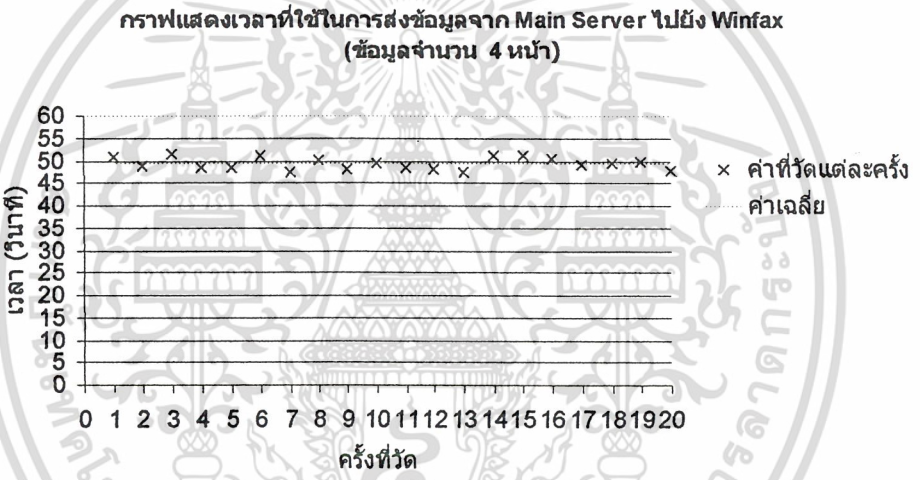


รูปที่ 4.19 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 2 หน้า

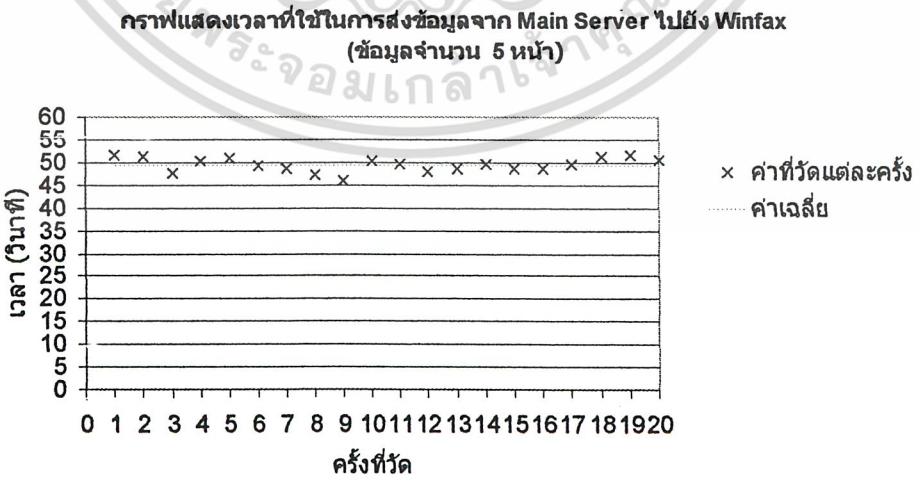
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 3 หน้า

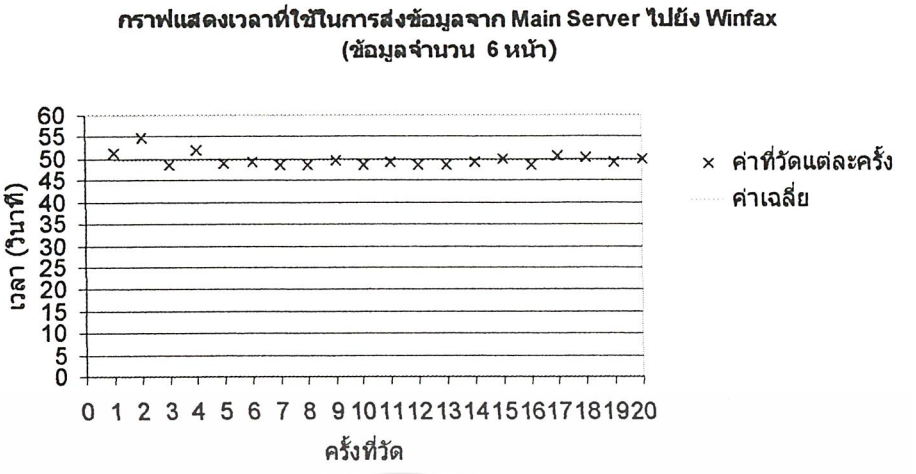


รูปที่ 4.21 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 4 หน้า



รูปที่ 4.22 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 5 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

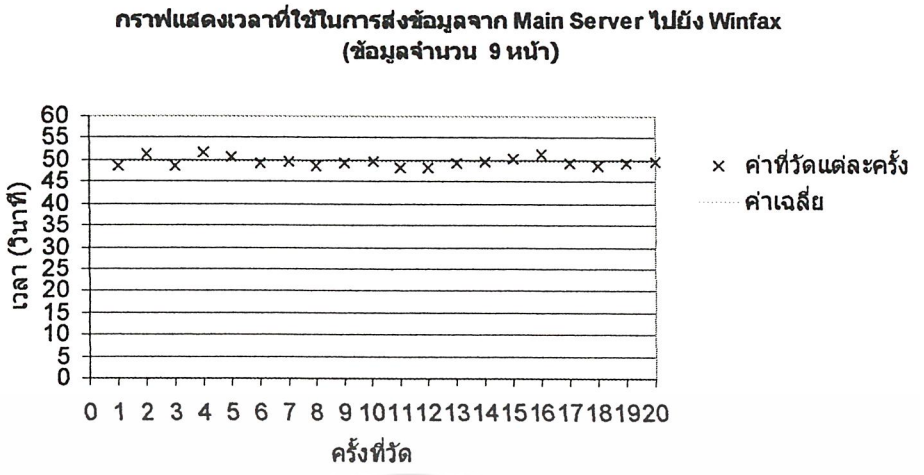


รูปที่ 4.23 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 6 หน้า

รูปที่ 4.24 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 7 หน้า

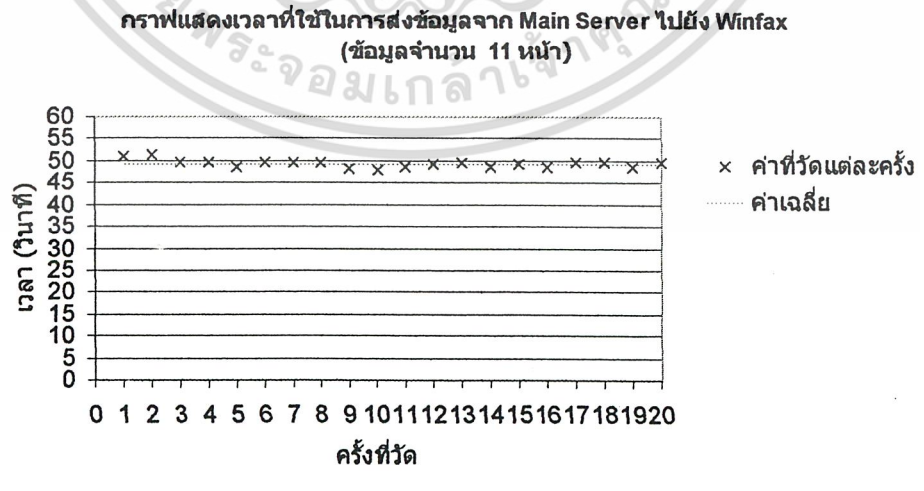
รูปที่ 4.25 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 8 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



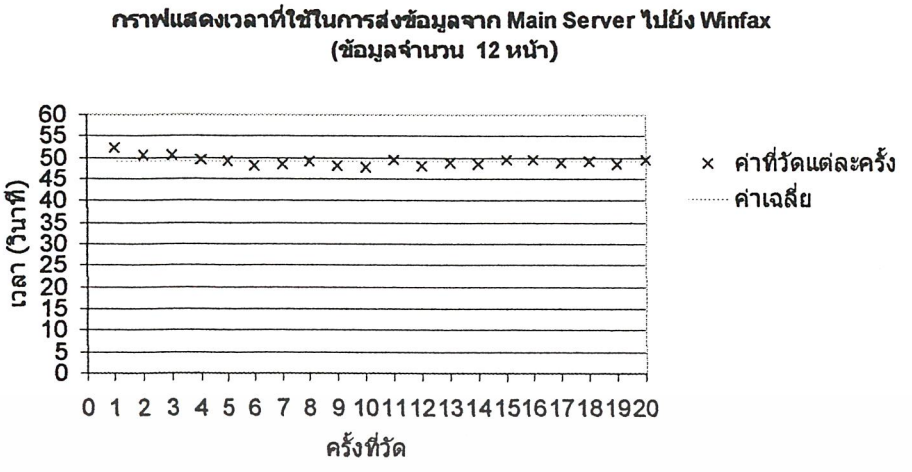
รูปที่ 4.26 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 9 หน้า

รูปที่ 4.27 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 10 หน้า

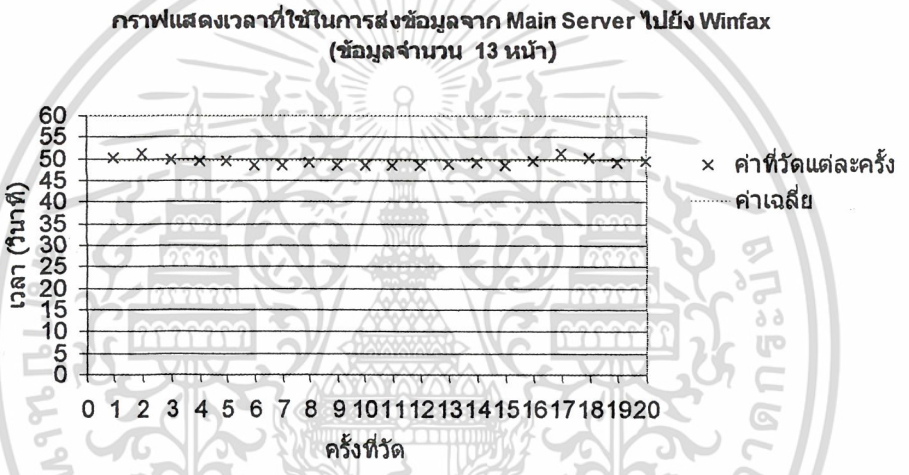


รูปที่ 4.28 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 11 หน้า

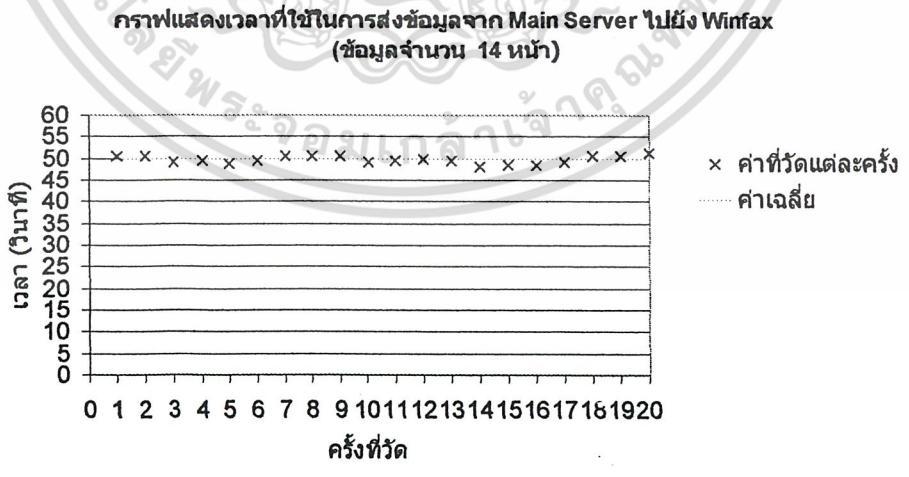
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 12 หน้า

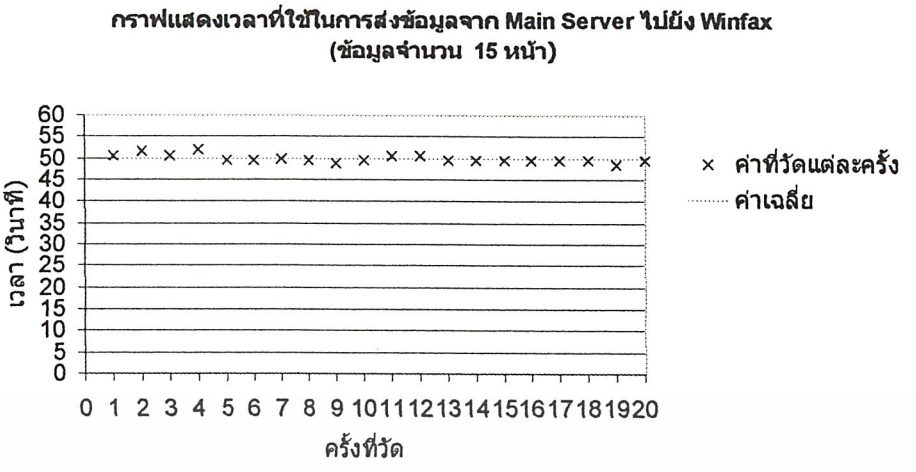


รูปที่ 4.30 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 13 หน้า

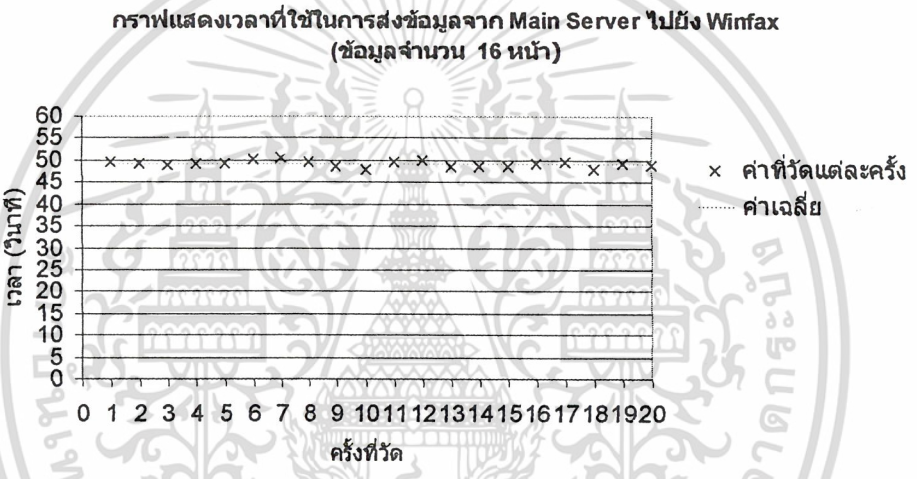


รูปที่ 4.31 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 14 หน้า

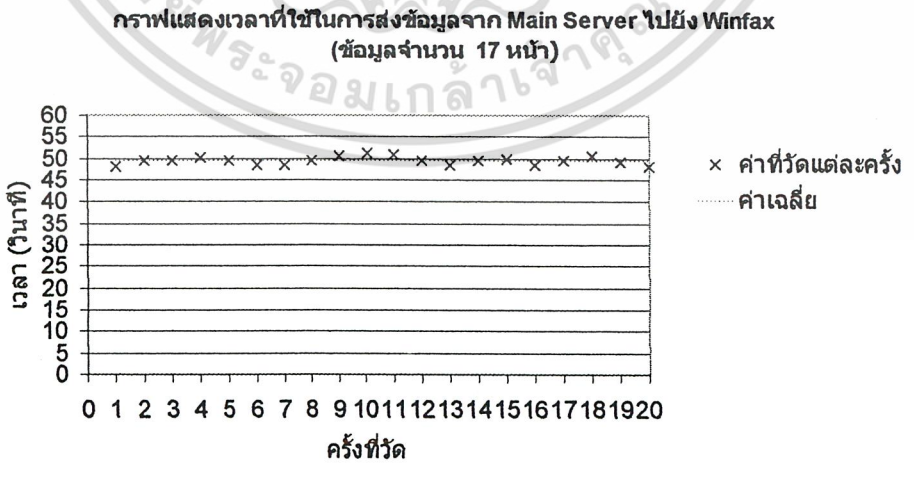
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 15 หน้า

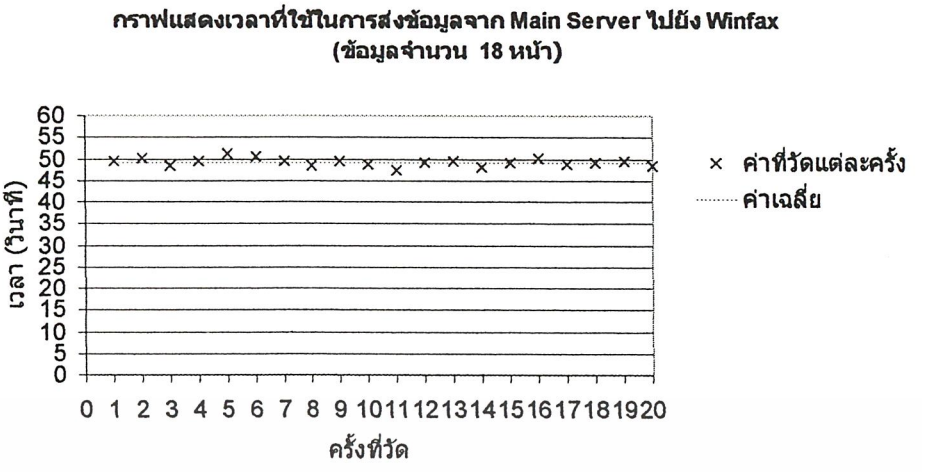


รูปที่ 4.33 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 16 หน้า

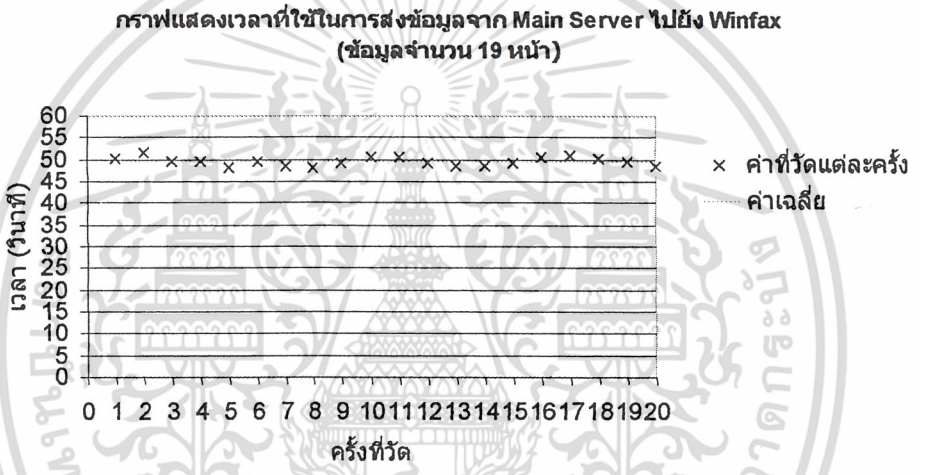


รูปที่ 4.34 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 17 หน้า

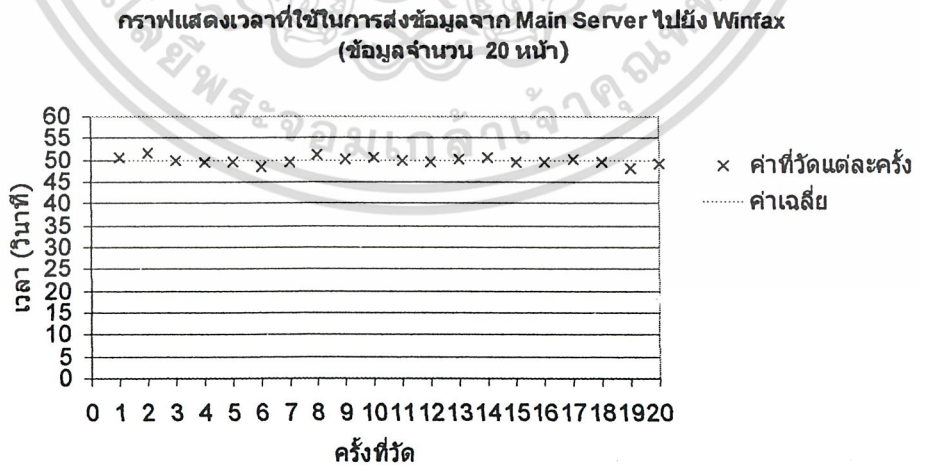
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.35 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 18 หน้า

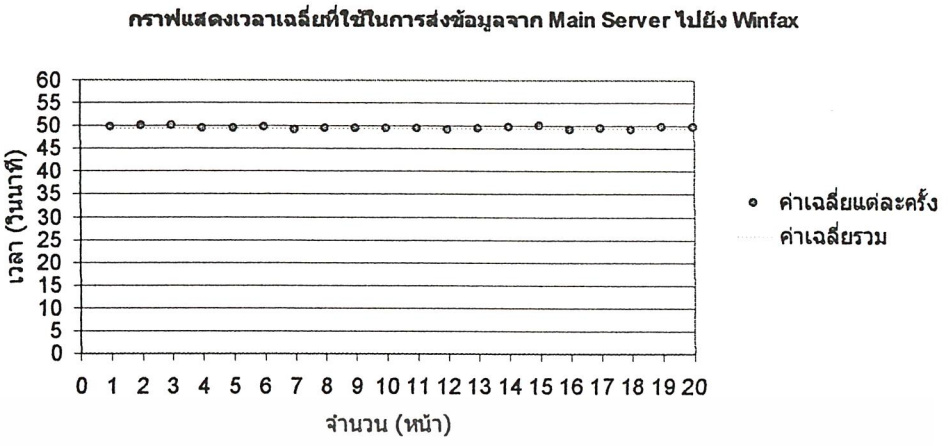


รูปที่ 4.36 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 19 หน้า



รูปที่ 4.37 กราฟแสดงเวลาที่ใช้ในการส่งข้อมูลจำนวน 20 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.38 กราฟแสดงเวลาเฉลี่ยที่ใช้ในการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทดลองการส่งโทรสารผ่านระบบอินเตอร์เน็ต โดยผ่านโปรแกรมที่ทำหน้าที่รับข้อมูลที่ต้องการส่งโทรสารจากผู้ใช้ แล้วประมวลผลเพื่อติดต่อไปยังเซิร์ฟเวอร์ปลายทางโดยผ่านเครือข่ายอินเตอร์เน็ต จากนั้นเซิร์ฟเวอร์ปลายทางก็ส่งข้อมูลต่อไปยังกระบวนการเข้ารหัสข้อมูลตามมาตรฐาน ITU-T(T.4) เมื่อเสร็จขั้นตอนดังกล่าวก็จะสั่งงานให้โมเด็มหมุนโทรศัพท์เพื่อติดต่อไปยังเครื่องโทรสารปลายทางต่อไป พบว่าสามารถทำงานได้ถูกต้องและให้ผลเป็นที่น่าพอใจคือ สามารถส่งข้อมูลความที่เป็นทั้งข้อความที่ผู้ใช้พิมพ์เข้ามา และข้อความที่มาจากไฟล์แนบได้อย่างถูกต้อง

5.2 ปัญหาที่พบและวิธีแก้ไข

1. ไม่สามารถทำการส่งโทรสารจากโปรแกรม Winfax ไปยังปลายทางได้ สามารถแก้ไขโดยการติดตั้ง Winfax เป็นเครื่องพิมพ์เริ่มต้น (set as default) และใช้คำสั่ง print เรียกโปรแกรม Winfax ขึ้นมา เพราะระบบปฏิบัติการจะมองโปรแกรม Winfax เปรียบเสมือน เครื่องพิมพ์เครื่องหนึ่ง ผู้ใช้สามารถเรียกโปรแกรม Winfax โดยใช้คำสั่ง printได้
2. ปัญหาเกี่ยวกับเทคนิคการเขียนโปรแกรมเรียก ใช้งานคอมโพเนนท์ DdeWinFaxConv ของโปรแกรม Winfax ไม่สามารถส่งโทรสารได้อัตโนมัติ แก้ไขโดยการกำหนด Properties ของคอมโพเนนท์ ในส่วนของ ConnectMode ให้เป็น ddeAutomatic , DdeService เป็น FAXMNG 32 , DdeTopic เป็น TRANSMIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Run Length	White bits	Black bits	Run Length	White bits	Black bits
0	00110101	0000110111	32	00011011	000001101010
1	000111	010	33	00010010	000001101011
2	0111	11	34	00010011	000011010010
3	1000	10	35	00010100	000011010011
4	1011	011	36	00010101	000011010100
5	1100	0011	37	00001110	000011010101
6	1110	0010	38	00010111	000011010110
7	1111	00011	39	00101000	000011010111
8	10011	000101	40	00101001	000001101100
9	10100	000100	41	00101010	000001101101
10	00111	0000100	42	00101011	000011011010
11	01000	0000101	43	00101100	000011011011
12	001000	0000111	44	00101101	000001010100
13	000011	00000100	45	00000100	000001010101
14	110100	00000111	46	00000101	000001010110
15	110101	000011000	47	00001010	000001010111
16	101010	0000010111	48	00001011	000001100100
17	101011	0000011000	49	01010010	000001100101
18	0100111	0000001000	50	01010011	000001010010
19	0001100	00001100111	51	01010100	000001010011
20	0001000	00001101000	52	01010101	000000100100
21	0010111	00001101100	53	00100100	000000110111
22	0000011	00000110111	54	00100101	000000111000
23	0000100	00000101000	55	01011000	000000100111
24	0101000	00000010111	56	01011001	000000101000
25	0101011	00000011000	57	01011010	000001011000
26	0010011	000011001010	58	01011011	000001011001
27	0100100	000011001011	59	01001010	000000101011
28	0011000	000011001100	60	01001011	000000101100
29	00000010	000011001101	61	00110010	000001011010
30	00000011	000001101000	62	001110011	000001100110
31	00011010	000001101001	63	00110100	000001100111

ตารางที่ 1 แสดงรหัส Terminating Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Run Length	White bits	Black bits
64	11011	000000111
128	10010	00011001000
192	010111	000011001001
256	0110111	000001011011
320	00110110	000000110011
384	00110111	000000110100
448	01100100	000000110101
512	01100101	0000001101100
576	01101000	0000001101101
640	01100111	0000001001010
704	011001100	0000001001011
768	011001101	0000001001100
832	011010010	0000001001101
896	101010011	0000001110010
960	011010100	0000001110011
1024	011010101	0000001110100
1088	011010110	0000001110101
1152	011010111	0000001110110
1216	011011000	0000001110111
1280	011011001	0000001010010
1344	011011010	0000001010011
1408	011011011	0000001010100
1472	010011000	0000001010101
1536	010011001	0000001011010
1600	010011010	0000001011011
1664	011000	0000001100100
1728	010011011	0000001100101
1792	00000001000	00000001000
1856	00000001100	00000001100
1920	00000001101	00000001101
1984	000000010010	000000010010
2048	000000010011	000000010011
2112	000000010100	000000010100
2170	000000010101	000000010101
2240	000000010110	000000010110
2304	000000010111	000000010111
2368	000000011100	000000011100
2432	000000011101	000000011101
2496	000000011110	000000011110
2560	000000011111	000000011111
EOL	000000000001	000000000001

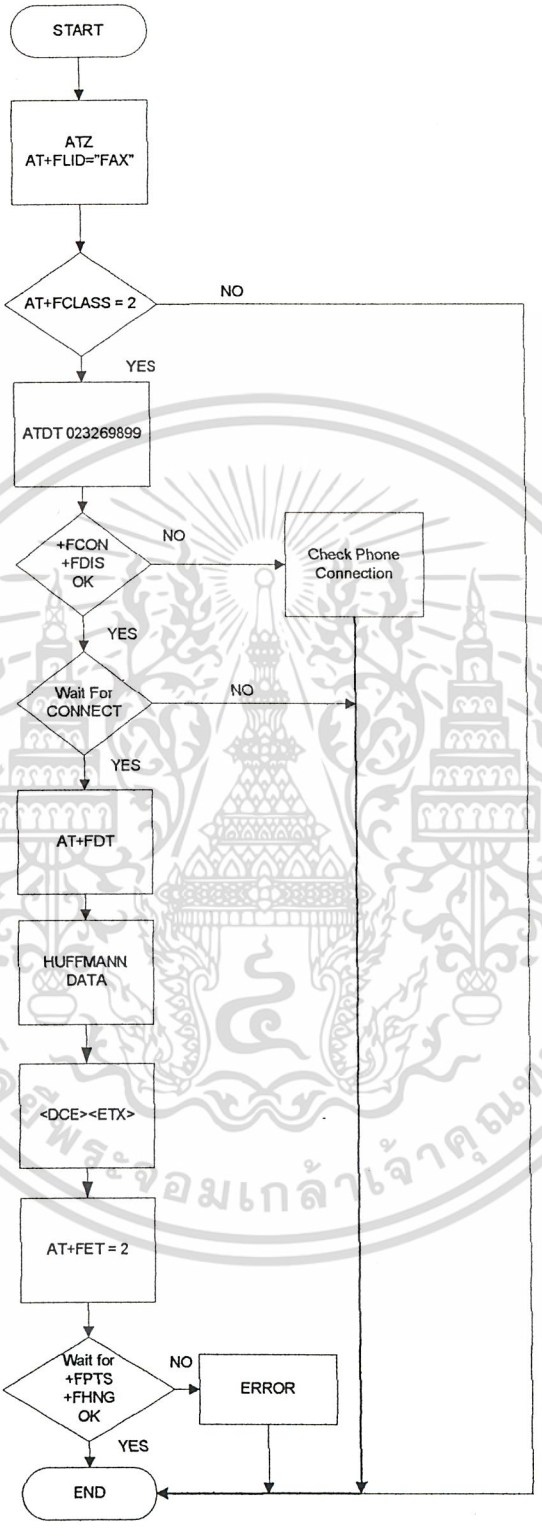
ตารางที่ 2 แสดงรหัส Make-up Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง AT COMMAND

ATZ	รีเซ็ตโมเด็ม
AT+FLID = “string”	กำหนดชื่อของโมเด็ม
AT+FCLASS = n	การเลือก FAX CLASS ของโมเด็มที่ใช้ในการทำงาน
	เช่น n = 2 กำหนดเป็น FAX CLASS 2
ATDT	คำสั่งหมุนโมเด็ม
AT+FCON	โมเด็มจะแสดงผลเพื่อบอกให้รู้ว่าการติดต่อ
	ระหว่างเครื่องโทรสารประสบความสำเร็จ
AT+FDIS	กำหนดพารามิเตอร์แสดงความสามารถของโมเด็ม ในโหมดส่งข้อมูล
AT+FDT	เป็นคำสั่งแจ้งโมเด็มว่าจะเริ่มส่งข้อมูลในเฟส C
AT+FET = n	คำสั่งบอกการสิ้นสุดของหน้ากระดาษ
	n = 0 มีอีก1หน้าแต่เป็นเอกสารเดียวกัน
	n = 1 สิ้นสุดเอกสารแต่มีเอกสารอื่นตามมาอีก
	n = 2 ไม่มีหน้ากระดาษหรือเอกสารเพิ่มเติม
AT+FPTS = n	แสดงสถานะการถ่ายโอนข้อมูล
	n = 1 เอกสารที่ได้ปรกติ
	n = 2 เอกสารที่ได้รับผิดพลาด ยังคงมีการเรียก
	n = 3 เอกสารที่ได้ปรกติ ยังคงมีการเรียก
AT+FHNG : nnn	แสดงผลตอบสนองการเรียก
	nnn = 00 สิ้นสุดการเรียก
	nnn = 10 การส่งข้อมูลผิดพลาดใน เฟส A
	nnn = 20 การส่งข้อมูลผิดพลาดใน เฟส B
	nnn = 40 การส่งข้อมูลผิดพลาดใน เฟส C
	nnn = 50 การส่งข้อมูลผิดพลาดใน เฟส D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงผังการทำงานของโมเด็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคิดค่าบริการ

Charging

คำนำ

การบริการโทรศัพท์ในอดีตโดยกองโทรศัพท์กรมไปรษณีย์โทรเลขหรือองค์การโทรศัพท์แห่งประเทศไทยตาม พ.ร.บ 2497 ในลักษณะผูกขาดเพราะผู้ให้บริการจะต้องฝึกหัดบุคลากรทางเทคโนโลยีเฉพาะทางให้แก่ช่วงเฉพาะด้านรับผิดชอบเพื่อให้บริการได้ 24 ชม. ดังนั้นองค์กรผู้ให้บริการดังกล่าวจึงมีสิทธิอันชอบธรรมที่จะกำหนดระเบียบบริหารจัดการขึ้นมาได้ ได้แก่ การกำหนดค่าบริการขั้นต่ำ ในฐานะผู้จัดหาอุปกรณ์โครงข่ายและบริการ

(1) เลือกหน่วยที่วัด

(2) วิธีวัดทราฟฟิกได้บริการ

หลักการที่นำมากำหนดค่าบริการ

7.2.1 หลักการกำหนดค่าบริการ มีดังนี้

- ค่าใช้จ่ายในการดำเนินการ
- ดอกเบี้ยเงินที่ลงทุน
- ค่าใช้จ่ายในการให้บริการสาธารณะในรอบปีงบประมาณ
- การเสื่อมค่า
- ค่าวิจัยและพัฒนา
- การลงทุน

องค์ประกอบดังกล่าวนี้องค์กรนำไปกำหนดค่าบริการขั้นต่ำ

ขบวนการคิดค่าบริการด้วยหน่วยทราฟฟิก (traffic unit price procedure)

ขบวนการคิดค่าบริการด้วยหน่วยทราฟฟิก คือ ขบวนการที่จะหารายได้เข้าองค์กรขึ้นกับจำนวนหน่วยทราฟฟิก (traffic unit)

เวลาสนทนา (conversation time)

เวลาเป็นนาทีของเวลาสนทนา คือ หน่วยทราฟฟิก (traffic unit)

เวลาสนทนา คือ ระยะเวลาที่ทราฟฟิกได้บริการผ่าน (เริ่มต้นถึงสิ้นสุด กล่าวคือ

- ขณะที่จับสภาวะสัญญาณตอบ (สัญญาณในทิศทางกลับหลัง) ณ จุด ที่เครื่องบันทึกเวลาการเรียนเริ่มต้น
- ขณะที่จับสภาวะทิศทางไปหน้าปลอด (clear forward signal) แสดงว่าการเรียกสิ้นสุด ณ จุดเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราคงที่ (flat rate)

บริการที่ผู้ใช้ได้รับสิทธิโดยไม่จำกัดจำนวนการเรียกโทรศัพท์ภายในพื้นที่ท้องถิ่นที่กำหนดไว้ด้วยอัตราคงที่

ในเมืองที่เปิดให้บริการโทรศัพท์โดยมีสมาชิกขั้นต้นจำนวนหนึ่งด้วยอัตราคงที่อาจเป็นรายเดือน ละ 100 บาท ผู้ให้บริการจึงจะสามารถให้บริการได้เป็นต้น

อัตราข่าวสาร (message rate)

อัตราข่าวสาร คือ เกณฑ์ที่ใช้วัดเพื่อคิดค่าบริการการเรียกท้องถิ่น ไม่คำนึงถึงระยะเวลาสนทนาและระยะทาง ด้วยการนับจำนวนครั้ง ผู้เช่าต้องจ่ายเพิ่มจากค่าบำรุงรักษารายเดือน

การกำหนดระยะทางคิดค่าบริการการเรียกทางไกล (distance call pricing plan)

เพื่อให้ผู้เช่าสามารถโทรศัพท์ทางไกลจ่ายค่าบริการเหมาะสมกับระยะทาง แนะนำ คือ ผู้ใช้ระยะใกล้จ่ายถูกกว่าผู้ใช้ระยะทางไกลกว่าในหน่วยเวลา เช่น นาที หรือ หน่วย ทราฟฟิก ในการคิดค่าบริการที่เป็นธรรม

การกำหนดระยะทางจะต้องลงตัวกับการกำหนดค่า K และการแปลรหัสปลายทางในประเทศ (รหัสพื้นที่ + รหัสชุมสาย) ตัวแรกบอกทิศทางสู่ภูมิภาคปลายทาง ตัวที่สองบอกระยะทางโดยสังเขป เช่น

รหัสปลายทางในประเทศ	ทิศทางส่งภูมิภาคปลายทาง	
5X	ทิศทางส่งภูมิภาคปลายทาง	เหนือ
“ 3	ทิศทางส่งภูมิภาคปลายทาง	“ ตอนบน
“ 6	ทิศทางส่งภูมิภาคปลายทาง	“ ตอนล่าง

ทศท. กำหนดระยะทางในโครงข่าย 4 ระดับ เป็น 6 ระยะ แต่โครงข่ายโทรศัพท์ทางไกลประหยัด 1234 ลดเหลือ 4 ระยะ และโทรศัพท์เคลื่อนที่เหลือ 3 ระยะดังแสดงในตารางที่ ก.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 การกำหนดระยะทางของ ทศท.

โครงข่ายPSTN		Y-tel 1234		โทรศัพท์เคลื่อนที่	
6 ระยะทาง	บาท/วินาที	4 ระยะทาง	บาท/วินาที	เขต โทรศัพท์เคลื่อนที่	บาท/นาที
0-50	3	0-50	2	เดียวกัน	3
51-100	6	51-100	4	ติดกัน	8
101-200	9	101-200	6	ไม่ติดกัน	12
201-300	12	201-	8		
301-500	15 *				
500-	18 *				

* ปรับลดเหลือ 12 บาท / นาที ตั้งแต่ 1 พ. ค . 2543

การกำหนดระยะเวลาทราฟฟิกและระยะเวลาคิดค่าบริการการเรียกทางไกล

(period of traffic and toll calls pricing plan)

องค์กรสามารถกำหนดค่าบริการระหว่างระยะเวลาทราฟฟิกสูง (period of heavy traffic) และระยะเวลาทราฟฟิกเบาลง (period of light traffic)

วงจรระหว่างประเทศ ระยะเวลาคิดค่าบริการไว้ดังนี้

(1) ระยะทางสนทนา แนะนำขั้นต่ำเต็ม 3 นาที

(2) ขึ้นกับขบวนของมิเตอร์พัลซ์ที่จ่าย ณ จุดเริ่มต้นของแต่ละระยะเวลา 3 นาที จำนวนของพัลซ์ในขบวนตามระยะทาง

วงจรในประเทศ คิดค่าบริการไว้ดังนี้

(1) 3+3 นาที หรือ 3+1 นาที

(2) นาทีต่อนาที

(3) ขบวนพัลซ์ตามระยะทาง

การกำหนดยุทธศาสตร์ด้านระยะเวลาทราฟฟิกมาใช้เพื่อจูงใจให้ผู้เช่าลดการใช้ในระยะเวลาทราฟฟิกสูง โดยเฉพาะในชั่วโมงใช้โทรศัพท์สูงสุด (busy hour) เพราะการจัดหาอุปกรณ์ชุมสายและโครงข่ายอยู่ในหลักการให้สามารถตอบสนองการเรียกในชั่วโมงใช้โทรศัพท์สูงสุดโดยสูญเสียน้อยที่สุด **“ดังนั้นระยะเวลาทราฟฟิกสูง” คิดค่าบริการสูงหรือปกติ ส่วนระยะเวลาทราฟฟิกเบาลง คิดค่าบริการลดลงตามลำดับ ดังแสดงในตารางที่ ก.2**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.2 การกำหนดระยะเวลาทราฟฟิกและอัตราของทั้ง ทศท. และ กสท.

ทศท.		กสท.	
ระยะเวลาทราฟฟิก	อัตรา	ระยะเวลาทราฟฟิก	อัตรา
07.00 – 18.00 น. สูง	กลางวัน	07.00 – 21.00 น.	ปกติ (standard rate)
18.00 – 22.00 น. เบาลง	ค่ำ (กลางวัน / 2)	21.00 – 24.00 น. 05.00 – 07.00 น.	ประหยัด (20 %) (economy rate)
			ลด (30 %) (reduced rate)
22.00 น.- 24.00น. เบาบาง	ดึก (กลางวัน / 3)	24. 00 – 05.00 น.	

อัตราค่าบริการก้าวหน้าของไทย (progress pricing plan of Thailand)

อัตราค่าบริการในการเรียกท้องถิ่น

การเรียกท้องถิ่นนับครั้งในการเรียกสำเร็จ (call attempt , complete) เป็น 1 ครั้ง ดังกล่าวแล้ว โดยไม่คำนึงระยะทางและระยะเวลาสนทนา ผู้เช่าต้องจ่ายเพิ่มจากค่าบำรุงรักษารายเดือนตามจำนวนครั้งที่ใช้

อัตราค่าบริการทั้งตามระยะเวลาทราฟฟิกและระยะทางเรียกทางไกลของทศท.

(double with period of traffic and distance call pricing plan of TOT)

การนำยุทธศาสตร์ มาผสมผสานของ ทศท. เพื่อให้ผู้ใช้มีทางเลือกมาที่จะประหยัดด้วยการเลือกใช้ระยะเวลาทราฟฟิกใด และเลือกประเภทค่าเช่ารายเดือนแบบใดที่ผู้ใช้เห็นว่าเป็นธรรม ดังแสดงในตารางที่ ก.3 และ ก.4 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.3 อัตราค่าบริการโทรศัพท์ทางไกลตามระยะเวลาทราฟฟิกของทศท.

ระยะเวลาทราฟฟิก/อัตรา ระยะทาง (กม.)	07.00 – 18.00 น. บาท / นาที	18.00 – 22.00 น. บาท / นาที	22.00 – 07.00 น. บาท / นาที
0 – 50	3	1.50	1.00
51 – 100	6	3.00	2.00
101 – 200	9	4.50	3.00
201 – 300	12	0.00	4.00
301 – 500	15*	7.50	5.00
501	18*	9.00	6.00

* ปรับลดเหลือเท่ากับระยะทางไม่เกิน 300 กม. ทุกระยะเวลาทราฟฟิกตั้งแต่ 1 พ.ค. 2543

ตารางที่ ก.4 ประเภทค่าเช่ารายเดือน

ประเภทแบบ	ค่าเช่ารายเดือน บาท	อัตราท้องถิ่น บาท / ครั้ง	อัตราโทรศัพท์ทางไกล บาท / วินาที
1	100.00	3	ตามตารางที่ 7.3
2	120.00	3	3, 6, 9 และ 12
3	90.00	1 บาท / นาที	3, 6 , 9 และ 12

หมายเหตุ : แบบที่ 2 และ 3 อัตรา 15 และ 18 เหลือ 12 บาท / นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราค่าบริการโทรศัพท์ทางไกลประหยัด 1234

อัตราค่าบริการโทรศัพท์ทางไกลประหยัด 1234 มีการลดราคาในวันหยุดและวันนักขัตฤกษ์ให้แก่ผู้ใช้อีกด้วย

ตารางที่ 7.5 อัตราค่าบริการ Y- tel 1234

ระยะเวลาทราฟฟิก / อัตรา แบบ	จันทร์ - ศุกร์		
	07.00- 18.00 น. บาท / นาที	18.00 – 22.00 น. บาท / นาที	22.00 – 07.00 น. บาท /นาที
0 – 50	2.00	1.00	0.75
51 – 100	4.00	2.00	1.50
101 – 200	6.00	3.00	2.25
201	8.00	4.00	3.00

อัตราค่าบริการเสริม SPC

บริการเสริม SPC ของ ทศท. มี 8 บริการ ผู้ใช้สามารถยื่นคำร้องขอใช้บริการได้ มีค่าบริการดังนี้

- บริการละ 30 บาท / เลขหมาย / เดือน
- ขอใช้ 3 บริการ / เลขหมาย / 80 บาท / เดือน
- ขอใช้เกิน 3 บริการขึ้นไปคิดค่าบริการที่ 4 ขึ้นไปบริการละ 25 บาท / เดือน

อัตราคาบริการประชุมด้วยภาพผ่านวงจรโทรศัพท์ (teleconference rate)

การประชุมด้วยภาพผ่านวงจรโทรศัพท์ แบ่งระยะทางและกำหนดอัตราดังแสดงในตารางที่ ก.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.6 อัตราค่าบริการ teleconference

อัตรา / ระยะทาง	ชั่วโมงแรก บาท	ทุกชั่วโมงเพิ่ม บาท
0 – 125	2, 400	1000
126 – 200	3, 000	1300
201 – 350	3, 600	1500
351 – 600	4,800	2000
601 – 900	6,000	25000
901-	7,200	3000

อัตราค่าบริการโทรศัพท์พื้นฐานนอกข่ายสาย

โทรศัพท์พื้นฐานที่สายไปไม่ถึงต้องใช้อุปกรณ์วิทยุเชื่อมโยงแทนการใช้คู่สายมีอัตราค่าบริการเช่นเดียวกับแบบที่ 3 ในตารางที่ ก.4

อัตราค่าบริการโทรศัพท์พื้นฐานใช้นอกสถานที่ (PCT)

โทรศัพท์พื้นฐานที่ผู้เช่าต้องการใช้นอกสถานที่พกพาเครื่อง PCT มีอัตราค่าบริการดังนี้

- ในเขตพื้นที่บริการ 3 / บาท 2 นาทีแรกและ 1.50 บาท / นาทีต่อไป
- ในเขตเดียวกัน , ระหว่างโซนติดกันและโซนไม่ติดกันอัตราตามโทรศัพท์เคลื่อนที่

อัตราค่าบริการโทรศัพท์ทางไกลระหว่างประเทศของ ทศท.

ทศท. ให้บริการโทรศัพท์ทางไกลไปยังประเทศเขตแดนติดกันกับประเทศไทยระบบอัตโนมัติ ISD/007 อัตราค่าบริการดังแสดงในตารางที่ ก.7

ตารางที่ ก.7 อัตรา ISD/007

ชื่อประเทศ	รหัสประเทศ	อัตราปกตินาทีละ		อัตราปกตินาทีละ (อาทิตย์)	
		07.00-18.00 น.	18.00-07.00 น.	07.00-18.00 น.	18.00-07.00น.
มาเลเซีย	60	30.00	20.00	15	10.00
เมียนม่า	95	30.00	24.00	24	24.00
กัมพูชา	855	30.00	24.00	24	24.00
ลาว	856	18.00	18.00	6	6.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อผ่านพนักงาน (101) ตัวอย่างเช่นไปมาเลเซีย อัตราปกติ 90 บาท / นาทีแรกต่อไป นาทีละ 30 บาท อัตราชายแดน ลดลงกึ่งหนึ่ง

อัตราค่าบริการโทรศัพท์ทางไกลระหว่างประเทศของ กสท.

ระบบการเรียกตรงอัตโนมัติ (ISD/001) คิดทั้งตามระยะทางไปแต่ละทวีป และระยะเวลาทราฟฟิก โดยคิดขั้นต้นเต็มนาทีแรกต่อไปคิดทุก 6 วินาที ที่เพิ่มดังแสดงในตารางที่ ก.7

ตารางที่ ก.7 อัตรา ISD/001

ระยะเวลาทราฟฟิก / ทวีป	อัตรานาทีแรก			ทุก 6 วินาทีที่เพิ่ม		
	ปกติ บาท	ประหยัด บาท	ลด บาท	ปกติ บาท	ประหยัด บาท	ลด บาท
เอเชีย	40	32	28	4.00	3.20	2.80
อาเซียน	30	24	24	3.00	2.40	2.40
ต.ออกกลาง	42	34	30	4.20	3.40	3.00
ยุโรป	36	29	25	3.60	2.90	2.50
ออสเตรเลีย	22	18	18	3.20	1.80	1.80
อาฟริกา	50	40	35	5.00	4.00	3.50
อเมริกา	50	40	35	5.00	4.00	3.50

หมายเหตุ (1) แต่ละทวีปมียกเว้น กล่าวคือ ราคาถูกกว่าในตาราง

(2) การเรียกผ่านพนักงานคิด 3 นาทีแรกมีอัตราบริการทั้งเจาะจง ไม่เจาะจงและเก็บเงินปลายทาง เกินคิดเป็นนาทีและบวกค่าพนักงานต่อ

(3) รายชื่อประเทศและรหัสประเทศที่โครงข่าย กสท. อำนวยในภาคผนวกที่ 7 ส่วนรายชื่อประเทศและรหัสประเทศดูได้จาก ITU T, Rec E.163 Annex A

ข่าวสารการเรียกโทรศัพท์ทั้งทางไกลระหว่างประเทศ

การที่องค์กรผู้ให้บริการจะออกใบเรียกเก็บค่าบริการการเรียกโทรศัพท์ทั้งทางไกลและระหว่างประเทศได้ แนะนำให้บันทึกข่าวสารเพื่อการดังกล่าวและหลักฐานทางบัญชี ดังนี้

- วัน, เวลา ที่เรียก
- เลขหมายในประเทศผู้เรียก
- **เลขหมายในประเทศผู้รับ**
- **ชนิด (เก็บเงินปลายทาง, บัตรเครดิตหรือจ่ายโดยบุคคลที่สาม)**
- **เลขบัตรเครดิต หรือเลขบัญชีเงินฝากออมทรัพย์**
- **ชนิดค่าบริการ (ปกติ, ลด, บุคคล หรือสถานี)**
- **ระยะเวลาคิดค่าบริการ (นาที) หรือเวลาสนทนา**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการชำระค่าบริการ

การแจ้งข่าวสารก็เพื่อให้องค์กรสามารถบริหารจัดการตามความประสงค์ และเป็นธรรมแก่ลูกค้าและยังต้องอำนวยความสะดวกให้แก่ลูกค้าในการชำระบริการด้วยวิธีต่างๆ

การชำระค่าบริการของ ทศท.

องค์กร เช่น ทศท. จัดวิธีการชำระอำนวยให้แก่ลูกค้า 7 วิธีดังนี้

1) ชำระที่ สนง. บริการทุกแห่ง
2) ชำระด้วยวิธีหักบัญชีฝากเงินออกทรัพย์
3) ชำระด้วยวิธีหักบัญชีบัตรเครดิต
4) ชำระทางอินเตอร์เน็ตเว็บไซต์ www.tot.web.net
5) ชำระที่ร้านสะดวกซื้อด้วยเงินสด
6) ชำระด้วยเงินสดที่ธนาคารที่ ทศท. มีข้อตกลงกัน
7) ชำระทางไปรษณีย์ลงทะเบียน

การชำระค่าบริการของ กสท.

องค์กร เช่น กสท. จัดวิธีการชำระค่าบริการอำนวยให้แก่ลูกค้า 4 วิธี

1) ชำระโดยตรงที่

- กองรายได้ สนง.ใหญ่ กสท. ถนนแจ้งวัฒนะ
- กองการเงินและผลประโยชน์อาคารชั้น 1 สนง. ใหญ่ กสท.
- อาคารไปรษณีย์กลาง ถนนเจริญกรุง
- ที่ทำการไปรษณีย์ สนง. การสื่อสารโทรคมนาคม หรือ สนง. บริการโทรคมนาคมสาขา
- เฉพาะเงินสดที่ธนาคารพาณิชย์ที่ กสท. มีข้อตกลงกัน

2) ชำระทางไปรษณีย์ (เฉพาะกรุงเทพและปริมณฑล)

3) ชำระทางโทรศัพท์

4) ชำระโดยวิธีหักโอนบัญชีเงินฝากธนาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

login.html

```
<html>
<head>
      <title>Member Sign Up</title>
</head>
<SCRIPT language=Javascript>
  function Validator() {

       if (document.FormAddress.mname.value == "") {
         alert("กรุณากรอกข้อมูลลงในช่อง Member Name ด้วยครับ.");
         document.FormAddress.mname.focus();
         return (false);
       }

       if (document.FormAddress.password.value == "") {
         alert("กรุณากรอกข้อมูลลงในช่อง Password ด้วยครับ.");
         document.FormAddress.password.focus();
         return (false);
       }
            return (true);
}
</SCRIPT>

<body>
 <form name="FormAddress" onsubmit="return Validator()"
action="login.asp" method="post">
<br>
<br>
<br>
<center>
<img src="image/sign_up_rd.gif" width="51" height="50"><br>
<table border="0">
<tr><td width="200" align="right">Member Name :</td><td>
  <input type="text" name ="mname" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">Password :</td><td>
  <input type="password" name="password" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>

<tr><td width="200" align="right"></td><td><input type="submit"
value="Login"</td></tr>
</table>
<br>

</center>
</form>
</body>
</html>
```

login.asp

```
<% response.buffer=true %>


<html>
<head>
      <title>Member Login</title>
</head>

<body>
<%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mname=request("mname")
password=request("password")


sql="select mname,password from member where mname='" & Mname & "'
and password='" & password & "' "

set conn=server.createobject("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath ("\indb6\upload.mdb" )


set rs=server.createobject("adodb.recordset")

rs.open sql,conn,1,3


if rs.eof then
      response.write("<center>Login ไม่ผ่าน</center>")

else
      session("x")=mname

        response.redirect("fax1.html")
end if
%>

</body>
</html>


member.html
<html>
<head>
      <title>Member Sign Up</title>
</head>
<script language="Javascript">
function checkform(FormAddress) {

       if (FormAddress.mname.value == "") {
         alert("กรุณากรอกข้อมูลลงในช่อง Member Name ด้วยครับ.");
         FormAddress.mname.focus();
         return (false);
       }

         if (FormAddress.mname.value.length < 4) {
         alert("กรุณากรอกข้อมูลลงในช่อง Member Name ให้มากกว่า 3ตัว ด้วยครับ.");
         FormAddress.mname.focus();
         return (false);
       }



       if (FormAddress.password.value == "") {
         alert("กรุณากรอกข้อมูลลงในช่อง Password ด้วยครับ.");
         FormAddress.password.focus();
         return (false);
       }

        if (FormAddress.password.value.length < 4) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          alert("กรุณากรอกข้อมูลลงในช่อง Password ให้มากกว่า 3ตัว ด้วยครับ.");
          FormAddress.password.focus();
          return (false);
        }

 if (FormAddress.repassword.value == "") {
          alert("กรุณากรอกข้อมูลลงในช่อง Re-Password' ด้วยครับ.");
          FormAddress.repassword.focus();
          return (false);
        }

        if (FormAddress.password.value !=
document.FormAddress.repassword.value) {
          alert("กรุณากรอก Password และ Re-Password ให้ตรงกัน");
          FormAddress.repassword.focus();
          return (false);
        }

        if (FormAddress.firstname.value == "") {
          alert("กรุณากรอกชื่อด้วยครับ.");
          FormAddress.firstname.focus();
          return (false);
        }

        if (FormAddress.lastname.value == "") {
          alert("กรุณากรอกนามสกุลด้วยครับ.");
         FormAddress.lastname.focus();
          return (false);
       }


          if (FormAddress.address.value== "") {
          alert("กรุณากรอกที่อยู่ด้วยครับ.");
          FormAddress.address.focus();
          return (false);
        }

          if (FormAddress.tel.value== "") {
          alert("กรุณากรอกหมายเลขโทรศัพท์ด้วยครับ.");
         FormAddress.tel.focus();
          return (false);
        }

        if (FormAddress.tel.value.length != 9) {
          alert("กรุณากรอกหมายเลขโทรศัพท์ให้ครบ 9 หมายเลขด้วยครับ.");
         FormAddress.tel.focus();
          return (false);
        }
        if (FormAddress.email.value == "") {
          alert("กรุณากรอก E-mail ด้วยครับ.");
          FormAddress.email.focus();
          return (false);
        }
        return (true);
}
</SCRIPT>

<body>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<form name="FormAddress" onsubmit="return checkform(this)"
action="member.asp" method="post">
<br>
<br>
<br>
<center>
<img src="image/sign_up_rd.gif" width="51" height="50"><br>
<table border="0">
<tr><td width="0" align="right">Member Name :</td><td>
  <input type="text" name ="mname" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">Password :</td><td>
  <input type="password" name="password" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width="200" align="right">Re-password :</td><td>
  <input type="password" name="repassword" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">ชื่อ:</td><td>
  <input type="text" name ="firstname" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">นามสกุล:</td><td>
  <input type="text" name ="lastname" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">ที่อยู่:</td><td>
  <textarea name ="address" rows="3" cols="20"></textarea><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">โทรศัพท์:</td><td>
  <input type="text" name ="tel" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">E-mail :</td><td>
  <input type="text" name ="email" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width="200" align="right">  <input type="image"
src="image/bt_signup.gif" width="81" height="22"></td></tr>
</table>

</center>
</form>
</body>
</html>
```

member.asp

```
<html>
<head>
      <title>Member Area</title>
</head>

<body>
<%



Set conn = Server.CreateObject("ADODB.Connection")
Set rs = Server.CreateObject("ADODB.Recordset")

mname=request("mname")


sql="select mname from member where mname='" & mname & "'"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
set conn1=server.createobject("adodb.connection")
conn1.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath ("\indb6\upload.mdb" )
set rs1=server.createobject("adodb.recordset")
rs1.open sql,conn1,1,3


'--- Open the connection

conn.Open "DSN=connString"



if rs1.eof then
                rs.Open "member", conn, 2,3
                rs.AddNew
                rs.Fields("mname").Value = request.Form("mname")
                rs.Fields("password").Value =
request.Form("password")

            rs.Fields("firstname").Value = request.Form("firstname")
                rs.Fields("lastname").Value =
request.Form("lastname")
            rs.Fields("address").Value = request.Form("address")
            rs.Fields("tel").Value = request.Form("tel")
            rs.Fields("email").Value = request.Form("email")

      rs.Update
      response.write("<center>ได้รับข้อมูลของคุณ" & mname & " เรียบร้อยแล้ว</center>
")
else
      response.write("<center>มีคนใช้แล้ว กรุณาป้อน Member Name ใหม่</center>")
end if
%>

<td><a href="fax1.html">HOME</a></td>
</body>
</html>


fax1.html

<html>
<head>
      <title>SEND FAX</title>
</head>

<script language="Javascript">
function checkform(FormAddress) {


         if (FormAddress.mname.value == "") {
          alert("กรุณากรอกข้อมูลลงในช่อง User name ด้วยครับ.");
          FormAddress.mname.focus();
          return (false);
        }

         if (FormAddress.mname.value.length < 4) {
          alert("กรุณากรอกข้อมูลลงในช่อง User name ให้มากกว่า 3ตัว ด้วยครับ.");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  FormAddress.mname.focus();
  return (false);
}


 if (FormAddress.password.value == "") {
  alert("กรุณากรอกข้อมูลลงในช่อง password ด้วยครับ.");
  FormAddress.password.focus();
  return (false);
}
 if (FormAddress.password.value.length < 4) {
  alert("กรุณากรอกข้อมูลลงในช่อง Password ให้มากกว่า 3ตัว ด้วยครับ.");
  FormAddress.password.focus();
  return (false);
}



if (FormAddress.receiver.value == "") {
  alert("กรุณากรอกข้อมูลลงในช่อง receiver ด้วยครับ.");
  FormAddress.receiver.focus();
  return (false);
}
 if (FormAddress.company.value == "") {
  alert("กรุณากรอกข้อมูลลงในช่อง company ด้วยครับ.");
  FormAddress.company.focus();
  return (false);
}
 if (FormAddress.faxnumber.value == "") {
  alert("กรุณากรอกข้อมูลลงในช่อง faxnumber ด้วยครับ.");
  FormAddress.faxnumber.focus();
  return (false);
}



 if (FormAddress.rmobile.value == "") {
  alert("กรุณากรอกข้อมูลลงในช่องreciever mobile ด้วยครับ.");
  document.FormAddress.rmobile.focus();
  return (false);
}


 if (FormAddress.rmobile.value.length != 9) {
  alert("กรุณากรอกข้อมูลลงในช่องreciever mobile ให้ครบ9ตัวเลข ด้วยครับ.");
  FormAddress.rmobile.focus();
  return (false);
}



  if (FormAddress.sender.value == "") {
  alert("กรุณากรอกข้อมูลลงในช่องsender ด้วยครับ.");
  FormAddress.sender.focus();
  return (false);
}
  if (FormAddress.smobile.value == "") {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        alert("กรุณากรอกข้อมูลลงในช่อง sender mobile ด้วยครับ.");
        FormAddress.smobile.focus();
        return (false);
      }


        if (FormAddress.smobile.value.length != 9) {
        alert("กรุณากรอกข้อมูลลงในช่อง sender mobile ให้ครบ9ตัวเลข ด้วยครับ.");
        FormAddress.smobile.focus();
        return (false);
      }

        if (FormAddress.subject.value == "") {
        alert("กรุณากรอกข้อมูลลงในช่อง subject ด้วยครับ.");
        FormAddress.subject.focus();
        return (false);
      }
        if (FormAddress.message.value == "") {
        alert("กรุณากรอกข้อมูลลงในช่องmessage ด้วยครับ.");
        FormAddress.message.focus();
        return (false);
      }
return (true);
}
</SCRIPT>


<body>


<form name="FormAddress" onsubmit="return checkform(this)"
action="formresp2.asp"   ENCTYPE="MULTIPART/FORM-DATA"
method="post">
<br>
<br>
<br>
<center>

<table border="0">

<tr><td width="200" align="right">User name:</td><td>
  <input type="text" name ="mname" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">password :</td><td>
  <input type="password" name ="password" size=40"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width="200" align="right">receiver:</td><td>
  <input type="text" name ="receiver" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">company :</td><td>
  <input type="text" name ="company" size=40"><FONT
      color=#ff0000>*</FONT></td></tr>

<tr><td width ="200" align="right">faxnumber :</td><td>
  <input type="text" name ="faxnumber" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">reciever mobile :</td><td>
  <input type="text" name="rmobile" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width="200" align="right">sender:</td><td>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  <input type="text" name="sender" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>

<tr><td width="200" align="right">sender mobile:</td><td>
  <input type="text" name="smobile" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>


<tr><td width ="200" align="right">subject:</td><td>
  <input type="text" name ="subject" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>

<tr><td width ="200" align="right">message:</td><td>
  <textarea name ="message" rows="10" cols="80"></textarea><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">file:</td><td>
  <input type="file" name ="myFile" size="40"></td></tr>
<tr><td align="right"><input type="submit" value="SEND" ></td></tr>
<tr><td><a href="member.html">Register</a></td>
</table>

</center>
</form>
</body>
</html>

formresp2.asp
<% @Language=VBSCRIPT %>

<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>

<BODY>

<%

dim time1
time1 = Now

'--- Declarations
      Dim oFileUp
      Dim oUploadedFile
      Dim oConn
      Dim oRS

Const adOpenDynamic = 2
      Const adLockOptimistic = 3


      Set oFileUp = Server.CreateObject("SoftArtisans.FileUp")


      oFileUp.Path = Server.MapPath(Application("vroot") & "/indb6")


      If Not oFileUp.Form("myFile").IsEmpty Then

            '--- Create ADODB Connection and Recordset
            Set oConn = Server.CreateObject("ADODB.Connection")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Set oRS = Server.CreateObject("ADODB.Recordset")

            '--- Open the connection

            oConn.Open "DSN=connString"
            '--- For simplicity, get a reference to the uploaded file
            Set oUploadedFile = oFileUp.Form("myFile")

            '--- Save the file
            oUploadedFile.Save


set rs = Server.CreateObject("ADODB.Command")
rs.activeconnection = "connString"


mname=oFileUp.form("mname")
password=oFileUp.form("password")

rs.commandtext= "select * from member where " &  "mname='" &
oFileUp.form("mname") & "' and "& "password='" &
oFileUp.form("password") &"' "
rs.commandtype=1
set rs =rs.execute

s="select credit from member where mname='" & Mname & "' and
password='" & password & "' "
set conn=server.createobject("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath ("\indb6\upload.mdb" )
set rs1=server.createobject("adodb.recordset")
rs1.open s,conn,1,3


if  rs.eof then
      response.write("<center>Login ไม่ผ่าน</center>")
elseif rs1("credit")<1 then
        response.write("<center>credit ไม่ผ่าน</center>")
else
      session("x")=mname


    rs1("credit")=rs1("credit")-1
       rs1.update

            '--- Open the recordset and add a new record
            oRS.Open "UploadTable2", oConn, adOpenDynamic,
adLockOptimistic
            oRS.AddNew

            '--- Insert the text values into the recordset

            oRS.Fields("mname").Value = oFileUp.Form("mname")
                oRS.Fields("password").Value =
oFileUp.Form("password")

            oRS.Fields("receiver").Value = oFileUp.Form("receiver")
                oRS.Fields("company").Value = oFileUp.Form("company")
            oRS.Fields("faxnumber").Value = oFileUp.Form("faxnumber")
            oRS.Fields("rmobile").Value = oFileUp.Form("rmobile")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            oRS.Fields("sender").Value = oFileUp.Form("sender")
            oRS.Fields("smobile").Value = oFileUp.Form("smobile")
            oRS.Fields("subject").Value = oFileUp.Form("subject")
            oRS.Fields("message").Value = oFileUp.Form("message")

                oRS.Fields("FilePath").Value =
oUploadedFile.ServerName

            '--- Update the recordset
            oRS.Update

            '--- The file is saved, display a confirmation message
            'Response.Write("The file was successfully saved on the
filesystem to: <b>" & oUploadedFile.ServerName & "</B>, and file
information was entered into the database at: <B>" &
Application("dbpath") & "</B>")

            '--- Close our ADO objects
            oRS.Close
            oConn.Close
                Response.Write("The fax has send. ")


        end if
      Else
            '--- Create ADODB Connection and Recordset
            Set oConn = Server.CreateObject("ADODB.Connection")
            Set oRS = Server.CreateObject("ADODB.Recordset")

            '--- Open the connection

            oConn.Open "DSN=connString"
            '--- For simplicity, get a reference to the uploaded file
            'Set oUploadedFile = oFileUp.Form("myFile")

            '--- Save the file
            'oUploadedFile.Save

set rs = Server.CreateObject("ADODB.Command")
rs.activeconnection = "connString"


mname=oFileUp.form("mname")
password=oFileUp.form("password")

rs.commandtext= "select * from member where " &  "mname='" &
oFileUp.form("mname") & "' and "& "password='" &
oFileUp.form("password") &"' "
rs.commandtype=1
set rs =rs.execute

s="select credit from member where mname='" & Mname & "' and
password='" & password & "' "
set conn=server.createobject("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath ("\indb6\upload.mdb" )
set rs1=server.createobject("adodb.recordset")
rs1.open s,conn,1,3

if  rs.eof then
      response.write("<center>Login ไม่ผ่าน</center>")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elseif rs1("credit")<1 then
        response.write("<center>credit ไม่พอ</center>")
else
      session("x")=mname

        rs1("credit")=rs1("credit")-1
        rs1.update

            '--- Open the recordset and add a new record
            oRS.Open "UploadTable2", oConn, adOpenDynamic,
adLockOptimistic
            oRS.AddNew

            '--- Insert the text values into the recordset


                oRS.Fields("mname").Value = oFileUp.Form("mname")
                oRS.Fields("password").Value =
oFileUp.Form("password")

            oRS.Fields("receiver").Value = oFileUp.Form("receiver")
                oRS.Fields("company").Value = oFileUp.Form("company")
            oRS.Fields("faxnumber").Value = oFileUp.Form("faxnumber")
            oRS.Fields("rmobile").Value = oFileUp.Form("rmobile")
            oRS.Fields("sender").Value = oFileUp.Form("sender")
            oRS.Fields("smobile").Value = oFileUp.Form("smobile")
            oRS.Fields("subject").Value = oFileUp.Form("subject")
            oRS.Fields("message").Value = oFileUp.Form("message")


'--- Update the recordset
            oRS.Update


            '--- Close our ADO objects
            oRS.Close
            oConn.Close

              Response.Write("The fax has send. ")


          end if
      End If


      '--- Destroy objects
      Set oFileUp = Nothing
      Set oRS = Nothing
      Set oConn = Nothing

dim secs
secs = DateDiff("s",time1,Now)

%>
<br>
Total time = <%=secs%> sec.
<br>
<td><a href="fax.html">BACK</a></td>

</BODY>
</HTML>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

admin.html

```
<html>
<head>
      <title>admin</title>
</head>

<SCRIPT language=Javascript>

  function Validator() {

        if (document.FormAddress.admin.value == "") {
          alert("กรุณากรอกข้อมูลลงในช่อง Member Name  ด้วยครับ.");
          document.FormAddress.admin.focus();
          return (false);
        }

        if (document.FormAddress.password.value == "") {
          alert("กรุณากรอกข้อมูลลงในช่อง Password  ด้วยครับ.");
          document.FormAddress.password.focus();
          return (false);
        }
            return (true);
}
</SCRIPT>

<body>


<br>
<center>
<table>

<form name="FormAddress" onsubmit="return Validator()"
action="admin.asp"      method="post">
<br>
<br>
<br>
<center>

<table border="0">


<tr><td width="200" align="right">admin Name :</td><td>
  <input type="text" name ="admin" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">Password :</td><td>
  <input type="password" name="password" size="20"><FONT
      color=#ff0000>*</FONT></td></tr>


<tr><td width="200" align="right">User name:</td><td>
  <input type="text" name ="mname" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>

<tr><td width ="200" align="right">credit:</td><td>
 <input type="text" name ="credit" size="40"><FONT
      color=#ff0000>*</FONT></td></tr>


<br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<tr><td width ="200" align="right"><input type="submit" value="SEND"
></td></tr>
<tr><td><a href="showmember.asp">SHOW MEMBER</a></td></tr>
<tr><td><a href="displaydb.asp">SHOW HISTORY</a></td></tr>

</table>
<br>

</center>
</form>
</body>
</html>
```

admin.asp

```
<html>
<head>
      <title>Member Area</title>
</head>

<body>
<%




admin=request("admin")
password=request("password")

mname=request("mname")
credit=request("credit")


sql1="select admin,password from admin where admin='" & admin & "'
and password='" & password & "' "



sql="select credit from member where mname='" & Mname & "'  "
set conn1=server.createobject("adodb.connection")
conn1.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath ("\indb6\upload.mdb" )
set rs1=server.createobject("adodb.recordset")


set conn=server.createobject("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath ("\indb6\upload.mdb" )
set rs=server.createobject("adodb.recordset")

rs.open sql1,conn,1,3


if rs.eof then
      response.write("<center>Login ไม่ผ่าน</center>")

else
      session("x")=mname
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
rs1.open sql,conn1,1,3


'--- Open the connection

'conn.Open "DSN=connString"



 rs1.Fields("credit").Value =
rs1.Fields("credit").Value+request.Form("credit")

      rs1.Update
response.write("<center>ได้รับข้อมูลของคุณ" & mname & " เรียบร้อยแล้ว</center> ")

end if
%>

<td><a href="admin.html">HOME</a></td>
</body>
</html>

displaydb.asp

<% Set ObjDB = Server.CreateObject("ADODB.Connection")
ObjDB.open"connString"

SQLString = "select*from UploadTable2"
set ObjRecordSet = Server.CreateObject("ADODB.Recordset")
ObjRecordSet.Open SQLString, ObjDB,,2

Col=ObjRecordSet.fields.count-1

%>
<html>
<body>
<h3>database</h3>
<table border=1>

<tr><%for i=0 to Col %>
<td><b><%=ObjRecordSet(i).name %></b></td>
<% next %></tr>
<%temp=1%>
<%do while not ObjRecordSet.eof %>
    <tr><td><%=temp%></td>
    <%for i=1 to Col%>
    <td><%= ObjRecordSet(i)%></td>
    <% next%></tr>
    <%ObjRecordSet.movenext
    temp=temp+1
loop%>
</table>

<%ObjRecordSet.close
ObjDB.close
set ObjRecordSet=nothing
set ObjDB=nothing%>
showmember.asp

<% Set ObjDB = Server.CreateObject("ADODB.Connection")
ObjDB.open"connString"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SQLString = "select*from member"
set ObjRecordSet = Server.CreateObject("ADODB.Recordset")
ObjRecordSet.Open SQLString, ObjDB,,2

Col=ObjRecordSet.fields.count-1

%>

<html>
<body>
<h3>database</h3>
<table border=1>

<tr><%for i=0 to Col %>
<td><b><%=ObjRecordSet(i).name %></b></td>
<% next %></tr>
<%temp=1%>
<%do while not ObjRecordSet.eof %>
    <tr><td><%=temp%></td>
    <%for i=1 to Col%>
    <td><%= ObjRecordSet(i)%></td>
    <% next%></tr>
    <%ObjRecordSet.movenext
    temp=temp+1
loop%>
</table>

<%ObjRecordSet.close
ObjDB.close
set ObjRecordSet=nothing
set ObjDB=nothing%>


unit mainserver;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DB, DBTables, ScktComp,
ExtCtrls,
  Menus;

type
        TfrmMain = class(TForm)
        Button1: TButton;
        Table1: TTable;
        DataSource1: TDataSource;
        DBGrid1: TDBGrid;
        Memo1: TMemo;
        ClientSocket1: TClientSocket;
        ServerSocket1: TServerSocket;
        Button3: TButton;
        MainMenu1: TMainMenu;
        About1: TMenuItem;
        N1: TMenuItem;
        N2: TMenuItem;
        exit: TMenuItem;
        Setting1: TMenuItem;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Set1: TMenuItem;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        ComboBox1: TComboBox;
        Label4: TLabel;
        Label5: TLabel;
        Timer1: TTimer;
        maindB: TDatabase;
        Label6: TLabel;
        procedure Button1Click(Sender: TObject);
        procedure FormCreate(Sender: TObject);
        procedure FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
        procedure ServerSocket1Accept(Sender: TObject;
        Socket: TCustomWinSocket);
        procedure Button3Click(Sender: TObject);
        procedure exitClick(Sender: TObject);
        procedure N1Click(Sender: TObject);
        procedure Set1Click(Sender: TObject);
        procedure Timer1Timer(Sender: TObject);
    procedure ServerSocket1ClientRead(Sender: TObject;
      Socket: TCustomWinSocket);
  private
        serviceType : integer;
        connectSocket : TCustomWinSocket;Socket: TCustomWinSocket;
        procedure InitVar;
        procedure Echo( content : shortString );
        procedure ReleaseVar;
    { Private declarations }
  public

    { Public declarations }
  end;

var
  frmMain: TfrmMain;
  attach_file,remote_file:string;
  SendDateTime:TDateTime;b : shortString;

fax_no,local_num,name,addr,subject,company,recipient,sDate,sTime,f_ty
pe,r_email,s_ender,s_email,dbname,memo:string;
  procedure callform1(attach_file:string;remote_file:string);
  procedure
callform2(attach_file:string;remote_file:string;fax_no:string;r_email
:string);
  function file_type(attach_file:string):string;

  type TMyRec= record
       fax_no,local_num,name,addr,subject,company,info:string;
  end;
  var MyData:TmyRec;

implementation
 uses main_ftp, showAbout, setting;

{$R *.dfm}
procedure TfrmMain.ReleaseVar;
begin
        frmMain.Release;   frmMain:=nil;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TfrmMain.Button1Click(Sender: TObject);
var j,i,total,m:integer;  portNo,n : integer;
   s : shortString; p : pchar; signal_confirm:string;
   check,fax_status:boolean;
   tempname:string;sname:array[1..9]of string;
    a,comma,data_num,c:integer;
begin
  table1.Close;
  table1.open;
  showmessage('Please make sure terminal servers are connected');
  //-------------------------------------------------------------
   sDate    := (FormatDateTime('mm/dd/yy hh:mm:ss',Now));

  m:=0;
   check:=true;
   table1.Close;
   table1.open;
 while check=true do
     begin
          while not table1.Eof do
             begin
                  timer1.Enabled:=false;
                  fax_status:=table1.FieldByName('Check').AsBoolean;
                  if fax_status = false then
                     begin
                       name:='';
                       fax_no:='';
                       subject:='';
                       company:='';
                       attach_file:='';
                       r_email:='';
                       s_ender:='';
                       s_email:='';
                       memo:= '';
//-------------------------------get parameter from table(db) ---------
-----------------------------------


name:=table1.fieldbyname('receiver').asstring;

fax_no:=table1.fieldbyname('faxnumber').asstring;

subject:=table1.fieldbyname('subject').asstring;

company:=table1.fieldbyname('company').asstring;

attach_file:=table1.fieldbyname('filepath').asstring;

r_email:=table1.fieldbyname('rmobile').asstring;

s_ender:=table1.fieldbyname('sender').asstring;

s_email:=table1.fieldbyname('smobile').asstring;
                       memo:=table1.fieldbyname('message').asstring;

                       sTime :=
FormatDateTime('hh:mm:ss',SendDateTime);
                       sDate :=
FormatDateTime('mm/dd/yy',SendDateTime);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        if attach_file='' then
                                attach_file:='d:\noFile.doc';
                        f_type:=file_type(attach_file);

                        s:=dbname;
                        local_num:='';  //clear old local number
                         //scan for local number
                        for i:=1 to 3 do
                            begin
                                local_num:= local_num+fax_no[i];
                            end;//for

                        //check for Bangkok
                        if (local_num[2])='2' then local_num:='02';

                        // assign terminal server
                        case strtoint(local_num) of

053:Mydata2.addr:=Form5.terminal1.Text;

044:Mydata2.addr:=Form5.terminal2.Text;
                                02:
Mydata2.addr:=Form5.terminal3.Text;
                                end;   //end case
                        portNo:=4000;
                        serviceType:=1;
                        ServerSocket1.port:=portNo;
                        ServerSocket1.Open;

                        callform1('d:\signalFile.doc','signalFile');
                        Memo1.Lines.Add('Try to
connect...'+local_num+' server');


                        n:=0;
                        c:=0;
                        while( connectSocket=nil ) do
                        Application.ProcessMessages;

callform2(attach_file,'attach',fax_no,r_email);

dbname:='"'+fax_no+'","'+name+'","'+company+'","'+subject+'","'

+attach_file+'","'+r_email+'","'+s_ender+'","'+s_email+'","'+memo+'"'
;
                        s:=dbname;
                        p:=@s[1];
                        connectSocket.SendBuf( p^, length( s ) );
                        memo1.Lines.Add('sending '+
fax_no+'('+s_ender +') data !');
                        table1.Edit;
                        table1.FieldValues['check']:=true;
                        table1.Post;
                        while  b<>'ok' do
                        application.ProcessMessages;
                        sleep(10000);
                        table1.Next;
                        connectSocket:=nil;
                        Application.ProcessMessages;
                      end
                      else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    table1.Next;//end if
                 application.ProcessMessages;
              timer1.Enabled:=true;
            end;// end eof
            timer1.Enabled:=true;
            application.ProcessMessages;
            check:=true;

     end;//while check

end; // end procedure

procedure TfrmMain.FormCreate(Sender: TObject);
begin
         InitVar;
end;

procedure callform1(attach_file:string;remote_file:string);
begin
        form1:=nil;
     if (form1=nil) then
     begin
       with form1 do
            begin
               MyData2.local:= attach_file;
               MyData2.remote:=remote_file+'.doc';
            end;
        application.CreateForm(TForm1,form1);
     end;
end;

procedure
callform2(attach_file:string;remote_file:string;fax_no:string;r_email
:string);
begin
        form1:=nil;
     if (form1=nil) then
     begin
         with form1 do
             begin
               MyData2.local:= attach_file;
               MyData2.remote:='attach'+fax_no+r_email+'.'+f_type;
             end;
        application.CreateForm(TForm1,form1);
     end;
end;
//---------------------------------------------------------------------
-
procedure TfrmMain.InitVar;
begin
        serviceType:=0; connectSocket:=nil;
end;

procedure TfrmMain.Echo( content : shortString );
begin
        memo1.lines.add( content );
end;

procedure TfrmMain.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ReleaseVar;
        canClose:=true;
end;

procedure TfrmMain.ServerSocket1Accept(Sender: TObject;
  Socket: TCustomWinSocket);
var   s : shortString; p : pchar;
begin
        s:=dbname;
        Echo( 'Connection accepted' );
        connectSocket:=Socket;
end;

function file_type(attach_file:string):string;
var filename,filetype : string;
    l,a,len: integer;
begin
        filename := attach_file;
        len:=length(filename);
        for l:= 1 to len do
           begin
                if filename[l]='.' then
                for a:= 1 to len-l  do
                    filetype := filetype+ filename[l+a];
           end;
        result:=filetype;
end;

procedure TfrmMain.Button3Click(Sender: TObject);
begin
        self.Close;
end;

procedure TfrmMain.exitClick(Sender: TObject);
begin
        self.Close;
end;

procedure TfrmMain.N1Click(Sender: TObject);
begin
       application.CreateForm(TForm3,form3);
       Form3.Show;
end;

procedure TfrmMain.Set1Click(Sender: TObject);
begin
       application.CreateForm(TForm5,form5);
       Form5.Show;
end;

procedure TfrmMain.Timer1Timer(Sender: TObject);
var  update_db:integer;select:string;
begin
        select:=combobox1.Text;
        case combobox1.ItemIndex of
             0:update_db:=30*1000;
             1:update_db:=60*1000;
             2:update_db:=90*1000;
             3:update_db:=120*1000;
             4:update_db:=150*1000;
             5:update_db:=180*1000;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            6:update_db:=300*1000;

        end;
        timer1.Interval:=update_db;
        table1.Close;
        table1.Open;

end;

procedure TfrmMain.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
var  pc : pchar; size : integer;
begin
  fillchar( b, sizeof( shortString ), 0 ); pc:=@b[1];
  size:=socket.ReceiveBuf( pc^, sizeof( shortString ) );
  b[0]:=chr(size);
  Echo( 'Receiving cofirm from terinal ' );
  echo(b);

end;

end.
*****************************************************************
*****************************************************************
unit main_ftp;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, ComCtrls, StdCtrls, ExtCtrls, Psock, NMFtp, DB, DBTables,
Grids,
  DBGrids;

type
  TForm1 = class(TForm)
    Memo1: TMemo;
    StatusBar1: TStatusBar;
    NMFTP1: TNMFTP;

    procedure NMFTP1Disconnect(Sender: TObject);
    procedure NMFTP1PacketSent(Sender: TObject);
    procedure NMFTP1TransactionStart(Sender: TObject);
    procedure NMFTP1TransactionStop(Sender: TObject);
    procedure NMFTP1Failure(var Handled: Boolean; Trans_Type:
TCmdType);
    procedure NMFTP1Success(Trans_Type: TCmdType);
    procedure FormCreate(Sender: TObject);

 private
    { Private declarations }
  public
    { Public declarations }
  end;
 type TMyRec2= record
   local,remote,addr:string;
   end;
   var MyData2:TmyRec2;
var
  Form1: TForm1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
implementation
 uses call_form_u6;
{$R *.dfm}


procedure TForm1.NMFTP1Disconnect(Sender: TObject);
begin
        Memo1.Lines.Add('Disconnected');
end;



procedure TForm1.NMFTP1PacketSent(Sender: TObject);
begin
        StatusBar1.SimpleText := IntToStr(NMFTP1.BytesSent)+' bytes
sent out of '
                          +IntToStr(NMFTP1.BytesTotal);

end;


procedure TForm1.NMFTP1TransactionStart(Sender: TObject);
begin
        Memo1.Lines.Add('Transaction Start');
end;


procedure TForm1.NMFTP1TransactionStop(Sender: TObject);
begin
        Memo1.Lines.Add('Transaction Stop');
end;

procedure TForm1.NMFTP1Failure(var Handled: Boolean; Trans_Type:
TCmdType);
begin
  case Trans_Type of
    cmdAppend   : Memo1.Lines.Add('UploadAppend failure');
    cmdDoCommand: Memo1.Lines.Add('DoCommand failure');
  end;
end;

procedure TForm1.NMFTP1Success(Trans_Type: TCmdType);
begin
  case Trans_Type of
     cmdAppend   : Memo1.Lines.Add('UploadAppend success');
     cmdDoCommand: Memo1.Lines.Add('DoCommand success');
  end;
end;


procedure TForm1.FormCreate(Sender: TObject);
var
  LocalFile,
  RemoteFile: String;
  F: File of Byte;
  FSize: Integer;
  begin
      if NMFTP1.Connected then
        NMFTP1.Disconnect
      else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            begin
                 NMFTP1.Host := Mydata2.addr;
                 NMFTP1.UserID := 'nui';
                 NMFTP1.Password := '123456';
                 NMFTP1.Connect;
            end;
       with Mydata2 do
          begin
           LocalFile := local;
           NMFTP1.Mode(MODE_ASCII);
           AssignFile(F, LocalFile);
           Reset(F);
           FSize := FileSize(F);
           CloseFile(F);
           NMFTP1.Allocate(FSize);
           NMFTP1.UploadAppend(LocalFile, remote);
          end;

end;

end.


unit terminal_confirm_u;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ExtCtrls, ScktComp, ComCtrls, DdeMan,
DdeWinFaxConv,
  jpeg, DB, DBTables, Grids, DBGrids;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Memo1: TMemo;
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    Table1: TTable;
    Database1: TDatabase;
    DataSource1: TDataSource;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
//    procedure FormClose(Sender: TObject; var Action: TCloseAction);
   procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);


  private
    procedure InitVar;
   procedure ReleaseVar;
    { Private declarations }
  public
    { Public declarations }
  end;

var
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Form1: TForm1;
  type
PExceptionRecord = ^TExceptionRecord;
TExceptionRecord = record
  ExceptionCode: Cardinal;
  ExceptionFlags: Cardinal;
  ExceptionRecord: PExceptionRecord;
  ExceptionAddress: Pointer;
  NumberParameters: Cardinal;
  ExceptionInformation: array[0..14] of Cardinal;
end;
var ExceptionRecord: PExceptionRecord;

implementation
uses tcp_winfax_u, ftp_u;
{$R *.dfm}

//---------------------------------------------------------------------
-
//---------------------------------------------------------------------
-

procedure TForm1.Button1Click(Sender: TObject);
var filename,attach,file_signal:string;check:boolean;
begin
 //       try
        memo1.lines.add( 'Server Start up!');
        check:=true;
          while check do
            begin
                file_signal:='c:\documents and
settings\signalFile.doc';
                while not fileExists(file_signal)do
                    application.ProcessMessages;
                with form2 do
                    begin
                      tcp_param.filename:='c:\documents and
settings\';
                    end;
                application.createForm(TForm2,Form2);

              //  deletefile(filename);
               // end; //end if
               application.ProcessMessages;
               deletefile(file_signal);
               check:=true;
            end; //end while
  //          except on EStackoverflow do ExceptionAddress;
   //       end;
end;

procedure TForm1.FormCreate(Sender: TObject);
var file_signal:string;
begin
    //  InitVar;
end;

procedure callform3(attach_file:string;remote_file:string);
begin
      frmFtp:=nil;
      if (frmFtp=nil) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      begin
           with form1 do
             begin
                MyData2.local:= attach_file;
                MyData2.remote:=remote_file;
             end;
     application.ProcessMessages;
     application.CreateForm(TfrmFtp,frmFtp);
     frmFtp.Show;
     //sleep(1000);
     frmFtp.Hide;

     end;
end;
//---------------------------------------------------------------------
-

procedure TForm1.Button2Click(Sender: TObject);
begin

      self.Close;
end;
{
procedure TForm1.FormClose(Sender: TObject; var Action:
TCloseAction);
var
      Busy :boolean;
begin
      busy:=false;
      if Busy then Exit else Busy := true;
      application.ProcessMessages;
      Busy:=false;
end;
 }
procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
var
      Busy :boolean;
begin
         busy:=false;
      if Busy then Exit else Busy := true;

        ReleaseVar;
        CanClose:=true;
         Busy:=false;
end;

procedure TForm1.InitVar;
begin
end;

procedure TForm1.ReleaseVar;
begin
        Application.ProcessMessages;
        table1.Active:=false;
        database1.Close;
        form1.Close;
end;

end.
*******************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*******************************************************
unit tcp_winfax_u;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ComCtrls, DdeMan, DdeWinFaxConv, ScktComp,
CPort;

type
  TForm2 = class(TForm)
    ClientSocket1: TClientSocket;
    ServerSocket1: TServerSocket;
    DdeWinFaxConv1: TDdeWinFaxConv;
    Memo1: TMemo;
    richedit1: TRichEdit;
    edSendDateTime: TEdit;
    ComPort1: TComPort;
    procedure FormCreate(Sender: TObject);
    procedure ClientSocket1Connect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ClientSocket1Error(Sender: TObject; Socket:
TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
    procedure ClientSocket1Lookup(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ClientSocket1Read(Sender: TObject; Socket:
TCustomWinSocket);
    private
    serviceType : integer;
    connectSocket : TCustomWinSocket; attach:string;
    procedure InitVar;
    procedure ReleaseVar;
    procedure Echo( content : shortString );
        { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;
  type Ttcp_param2=record
       filename,attach,file_signal:string;
       end;
       var  tcp_param:Ttcp_param2;
       dbname,mobile_number,mobile_number1,mobile_number2:string;
  function file_type(attach_file:string):string;

implementation
 uses terminal_confirm_u, sms_u;
{$R *.dfm}

procedure TForm2.InitVar;
begin
        serviceType:=0;
        connectSocket:=nil;
end;

procedure TForm2.ReleaseVar;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin

end;

procedure TForm2.Echo( content : shortString );
begin
        Form1.memo1.lines.add( content );
end;

procedure TForm2.FormCreate(Sender: TObject);
var hostName : shortString; portNo : integer;
begin
        edSendDateTime.Text := FormatDateTime('mm/dd/yy
hh:mm:ss',Now);
        hostName:='192.168.1.2';
        serviceType:=2;
        ClientSocket1.Host:=hostName;
        ClientSocket1.Port:=4000;
        ClientSocket1.Open;
end;

procedure TForm2.ClientSocket1Connect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
        Echo( 'Connection established' );
end;

procedure TForm2.ClientSocket1Error(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
        Echo( ':ERROR' );
        Echo( '  Error code = '+IntToStr( errorCode ) );
end;

procedure TForm2.ClientSocket1Lookup(Sender: TObject;
  Socket: TCustomWinSocket);
begin
        Echo( 'Browsing server..' );
end;

procedure TForm2.ClientSocket1Read(Sender: TObject;
  Socket: TCustomWinSocket);
var s ,b: shortString; p : pchar;
    size,a,comma,data_num: integer;
    recipient,sDate,
sTime,sRecipient,f_type,attach2,sms_num1,sms_num2:string;
    tempname:string;sname:array[1..9]of string;
    SendDateTime:TDateTime ;
    Str_r,Str1_r,Str2_r,Str3_r,Str4_r,Str5_r: String;
    Str_s,Str1_s,Str2_s,Str3_s,Str4_s,Str5_s: String;

begin
        s:='';
        fillchar( s, sizeof( shortString ), 0 ); p:=@s[1];
        size:=socket.ReceiveBuf( p^, sizeof( shortString ) );
        s[0]:=chr(size);
 //----------------------- scan for receipient --------------------
        a:=1;
      data_num:=10;
      comma:=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      dbname:=s;
     //
dbname:='"'+fax_no+'","'+name+'","'+company+'","'+subject+'","'
      //
+attach_file+'","'+r_email+'","'+s_ender+'","'+s_email+'","'+memo+'"'
;
    //name:='"","","a3","a4","","a6","a7","";
    while comma<data_num do
    begin
        if dbname[a]='"' then
           if dbname[a+1]='"' then
                tempname:=''
        else
           while (dbname[a+1]<>'"') and (dbname[a+2]<>',') do
           begin
                tempname:=tempname+dbname[a+1];
                a:=a+1
           end;
           sname[comma]:=tempname;
           tempname:='';
           a:=a+3;
           comma:=comma+1;

    end;

        SendDateTime := StrToDateTime(edSendDateTime.Text);
        sTime := FormatDateTime('hh:mm:ss',SendDateTime);
        sDate := FormatDateTime('mm/dd/yy',SendDateTime);
        sRecipient:='recipient("' +  sname[1]          + '","'+
                                     stime             +'","' +
                                     sdate             +'","' +
                                     sname[2]          + '","' +
                                     sname[3]          + '","' +
                                     sname[4]          +   '")';
    form1.Table1.Append;
    form1.Table1.FieldValues['Fax Number']:=sname[1];
    form1.Table1.FieldValues['Receiver Name']:=sname[2];
    form1.Table1.FieldValues['Sender Name']:=sname[7];
    form1.Table1.Post;
 //----------------------- end scan for receipient ------------------
--
 //----------------------- attach file ------------------------------
--
  Echo( 'Receiving  '+sname[1]+'  data ... ' );
  attach:=tcp_param.filename+'attach'+sname[1]+sname[6]+'.';
  f_type:=file_type(sname[5]);
  attach2:=attach+f_type;

 //------------------------ end attach file -------------------------
------
 //------------------------ send data to winfax ---------------------
----------
   with DdeWinfaxConv1.SendFaxParams do
     begin
        recipient:=sRecipient;
        Textmessage:='......';
        SendDateTime := StrToDateTime(edSendDateTime.Text);
        Attach:=attach2;
        richedit1.Text:='Message:>>'+sname[9];
     end;
     DdeWinFaxConv1.Execute;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      RichEdit1.Print(Caption);


      Application.ProcessMessages;
      Echo( 'sending data to winfax..' );

      Echo( 'data has been sent to winfax.' );
      Echo( 'sending SMS..' );
    //  mobile_number1:= '065496490';
      mobile_number1:=sname[6];
      mobile_number2:=sname[8];
       sms_num1:= mobile_number1[3]+
mobile_number1[2]+mobile_number1[5]

+mobile_number1[4]+mobile_number1[7]+mobile_number1[6]
                    +mobile_number1[9]+mobile_number1[8];



     sms_num2:= mobile_number2[3]+
mobile_number2[2]+mobile_number2[5]

+mobile_number2[4]+mobile_number2[7]+mobile_number2[6]
                    +mobile_number2[9]+mobile_number2[8];


  ////////////////////////////////////////////////
  Str_r  :=  'AT+CMGS=30' ;
  Str1_r := '0691661900190211000A9166';
  Str2_r := sms_num1;
  Str3_r := '0004AA11596F7520686176652O6E6577206661782E';

  ComPort1.Open;
    Str_r := Str_r + #13#10;
    Str1_r := Str1_r+Str2_r+Str3_r + #26#10;
  ComPort1.WriteStr(Str_r);
  sleep(1000);
  ComPort1.WriteStr(Str1_r);

   sleep(6000);

   Str_s  :=  'AT+CMGS=31' ;
  Str1_s := '0691661900190211000A9166';
  Str2_s := sms_num2;
  Str3_s := '0004AA12466178206861732062656560073656E742E';

  ComPort1.Open;
    Str_s := Str_s + #13#10;
    Str1_s := Str1_s+Str2_s+Str3_s + #26#10;
  ComPort1.WriteStr(Str_s);
  sleep(1000);
  ComPort1.WriteStr(Str1_s);
  sleep(3000);
  comport1.Close;
  ////////////////////////////////////////////////


   sleep(50000);
   echo('SMS has sent to sender and receiver!');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   Echo( 'waiting to receive next data from main server...' );

   b:='ok'; p:=@b[1];
   clientSocket1.Socket.SendBuf( p^, length( b ) );
   sname[1]:='';sname[2]:='';sname[7]:='';

end;
 //------------------------ end send data to winfax -----------------
--------------

function file_type(attach_file:string):string;
var filename,filetype : string;
    l,a,len: integer;
   begin
    filename := attach_file;
    len:=length(filename);
    for l:= 1 to len do
    begin
        if filename[l]='.' then
           for a:= 1 to len-l  do
           filetype := filetype+ filename[l+a];
    end;
    result:=filetype;
end;
end.
*******************************************************
*******************************************************
unit DdeWinFaxConv;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  DdeMan;

const
  WINFAX_SERVICE = 'FAXMNG32';
  WINFAX_TOPIC = 'TRANSMIT';
  WINFAX_ITEM = 'sendfax';

type
//  ELinkWinFaxFailure = class(Exception);

  TSendFaxParams = class(TPersistent)
  private
    FPhoneNumber:string;
    FSendDateTime:TDateTime;
    FRecipientName:string;
    FRecipientCompany:string;
    FSubject:string;
    FAttach:string;
    FTextmessage:string;
     FRecipient:string;
    { Private declarations }
  public
    constructor Create;
    destructor Destroy; override;
    { Public declarations }
  published
    property PhoneNumber:string read FPhoneNumber write FPhoneNumber;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    property SendDateTime:TDateTime read FSendDateTime write
FSendDateTime;
    property RecipientName:string read FRecipientName write
FRecipientName;
    property RecipientCompany:string read FRecipientCompany write
FRecipientCompany;
    property Subject:string read FSubject write FSubject;
     property Textmessage:string read FTextmessage write
FTextmessage;
     property Attach:string read FAttach write FAttach;
     property Recipient:string read FRecipient write FRecipient;
    { Published declarations }
  end;

  TDdeWinFaxConv = class(TDdeClientConv)
  private
    FIncludeSendDateTime:boolean;
    FSendFaxParams:TSendFaxParams;


    { Private declarations }
  protected
    { Protected declarations }
  public
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override;

    procedure Execute;
    { Public declarations }
  published
    property IncludeSendDateTime:boolean read FIncludeSendDateTime
write FIncludeSendDateTime;

    property SendFaxParams:TSendFaxParams read FSendFaxParams write
FSendFaxParams;

    { Published declarations }
  end;

procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('Added', [TDdeWinFaxConv]);
end;

constructor TSendFaxParams.Create;
begin
     inherited Create;
     FPhoneNumber := '';
     FSendDateTime := Now;
     FRecipientName := '';
     FRecipientCompany := '';
     FSubject := '';
     FAttach:='';
     FTextmessage:='';
end;

destructor TSendFaxParams.Destroy;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
     inherited Destroy;
end;

constructor TDdeWinFaxConv.Create(AOwner: TComponent);
begin
     inherited Create(AOwner);
     FIncludeSendDateTime := False;
     FSendFaxParams := TSendFaxParams.Create;
end;

destructor TDdeWinFaxConv.Destroy;
begin
     FSendFaxParams.Free;
     inherited Destroy;
end;



procedure TDdeWinFaxConv.Execute;
var
   sRecipient, sNumber, sDate,
sTime,sAttach,sShow,stextmessage:string;
begin
     // Create & send the WinFax "recipient" DDE command


     with FSendFaxParams do
     begin

          stextmessage:='fillcoverpage("'+Textmessage+'")';
          RecipientName:=RecipientName +'@'+ RecipientCompany;

          if IncludeSendDateTime then begin
             sTime := FormatDateTime('hh:mm:ss',SendDateTime);
             sDate := FormatDateTime('mm/dd/yy',SendDateTime);
          end
          else begin
             sTime := '';
             sDate := '';
          end;

          sRecipient :=recipient;
          sAttach:='attach("'+Attach+'")';
          end;
      PokeData(WINFAX_ITEM,PChar(sRecipient));          //To send fax
parameter
      PokeData(WINFAX_ITEM,PChar(sAttach));             //To attach
file
      PokeData(WINFAX_ITEM,PChar(stextmessage));        //To fill
text to comment on coverpage
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ ดร.พิพัฒน์ พรหมมี อาจารย์ที่ปรึกษาที่ให้คำแนะนำและให้คำปรึกษาที่มีค่ายิ่ง ขอขอบคุณทุกคำชี้แนะเกี่ยวกับเทคนิคการเขียนโปรแกรม ขอขอบคุณกำลังใจจากเพื่อน ๆ ทุกคน และที่สำคัญที่สุดคือขอกราบขอบพระคุณ คุณพ่อ และ คุณแม่ ที่พวกเราเคารพรัก และทำให้มีพวกเราในวันนี้ ถ้ามีข้อผิดพลาดประการใดทางผู้จัดทำต้องขออภัยมา ณ ที่นี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ประพนธ์ อัศวภาณุวัฒน์, Delphi Episode II เทคนิคและการเขียนโปรแกรมด้วยเดลไฟ, กรุงเทพฯ:ซีเอ็ดยูเคชั่น, 2543

ประสิทธิ์ วิทยธีราภรณ์,เทคนิคการสื่อสารด้วยโมเด็ม, กรุงเทพฯ: ฟิสิกส์เซ็นเตอร์,2542

กนก กุสุมาลย์นุกูล ไกรวุฒิ มั่นเสถียรสิน, คู่มือการเขียนโปรแกรม Delphi 4, กรุงเทพฯ : ซัคเซสมีเดีย, 2541

สัจจะ จรัสรุ่งรวีวร, เริ่มต้นอย่างมืออาชีพด้วยDelphi7 ฉบับสมบูรณ์, นนทบุรี :อินโฟเพรส,2546

ธวัชชัย สุริยะทองธรรม , E-Commerce Application ด้วย ASP, 4, กรุงเทพฯ : ซัคเซสมีเดีย, 2541

ศ.มนูญ สุขเกษม, ทราฟฟิกและโครงข่ายโทรศัพท์, กรุงเทพ:คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้