สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

สถานีวิทยุบนระบบเครือข่ายอินเตอร์เน็ต

Internet Radio

โดย

นายพันกิจ ประทานพรทิพย์

นายพิพัฒน์ แซ่จึง

เลขหมู่.....
เลขทะเบียน 42221
วัน, เดือน, ปี 15 พ.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานีวิทยุบนระบบเครือข่ายอินเตอร์เน็ต

Internet Radio

โดย

นายพันกิจ ประทานพรทิพย์ 40010502

นายพิพัฒน์ แซ่จึง 40010515

อาจารย์ที่ปรึกษา

อ. นภัทร สระเอี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **สถานีวิทยุบนระบบเครือข่ายอินเตอร์เน็ต**

Internet Radio

ผู้จัดทำ

1. นายพันกิจ ประทานพรทิพย์ **40010502**

2. นายพิพัฒน์ แซ่จึง **40010515**

_____ อาจารย์ที่ปรึกษา

(อ. นภัทร สระเอี่ยม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานีวิทยุบนระบบเครือข่ายอินเตอร์เน็ต

Internet Radio

โดย นายพันกิจ ประทานพรทิพย์ 40010502
นายพิพัฒน์ แซ่จึง 40010515

อาจารย์ที่ปรึกษา อ. นภัทร สระเอี่ยม

บทคัดย่อ

เนื่องจากปัจจุบันในการจัดตั้งสถานีวิทยุขึ้นมาเป็นเรื่องยาก เพราะว่าความถี่คลื่นวิทยุถูกจับจองไว้มากแล้ว อีกทั้งงบค่าใช้จ่ายในการสร้างก็สูง โครงงานนี้ได้การสื่อสารผ่านทางอินเตอร์เนตมาประยุกต์ใช้เป็นสถานีวิทยุทางอินเตอร์เนต ซึ่งทำให้ประหยัดค่าใช้จ่ายได้มาก และทำให้ครอบคลุมพื้นที่ได้มากขึ้นด้วย

Abstract

It is difficult to have a new radio station nowadays because almost all available frequencies have been used and the price would be quiet expensive too. This project will applied the internet to solve this problem by having radio station on the internet that would be a lot more less expensive and could be global broadcasting.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีหรือหลักการ	2
2.1 อินเตอร์เน็ต (Internet)	2
2.2 OSI โมเดล	3
2.3 โปรโตคอลและไอพีแอดเดรส (IP Address)	6
2.3.1 โปรโตคอลคืออะไร	7
2.3.2 การอ้างอิงอุปกรณ์ในเครือข่าย	9
2.3.3 การจัดลำดับชั้นของเครือข่าย (Network Class)	14
2.4 โปรโตคอลทีซีพี/ไอพี (TCP/IP Protocol)	21
2.4.1 TCP/IP คืออะไร	22
2.4.2 ส่วนประกอบของ TCP/IP	22
2.4.3 สถาปัตยกรรม TCP/IP	24
2.4.4 กลไกของโปรโตคอล IP	25
2.5 พัลส์โค้ดมอดดูเลชั่น (Pulse Code Modulation)	27
2.5.1 แซมปลิ้ง (Sambling)	28
2.5.2 ควอนไตซิ่ง (Quantizing)	28
2.5.3 การเข้ารหัส (Coding)	29
2.6 ไฟล์สกุล .wav	30
2.7 วินโดวส์ ซ้อกเก็ต (Windows Sockets)	32
2.7.1 ซ้อกเก็ต คืออะไร	33
2.7.2 การได้รับ API ของวินโดว์ซ้อกเก็ต	33
2.8 ข้อมูลการ์ดเสียง (Sound Card)	34
2.8.1 การสังเคราะห์เสียง	36
2.8.2 คุณสมบัติของการ์ดเสียง	36
2.9 ความเป็นมาของเทคโนโลยีมัลติมีเดียใน Internet	36
2.9.1 Multicast Protocol	38
2.9.2 MBONE	39
2.9.3 Real-Time Streaming Protocol (RTSP)	41
บทที่ 3 การออกแบบและการสร้าง	43
3.1 ขั้นตอนการทำงานของโปรแกรม	43
3.2 การเขียนโปรแกรม	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

3.2.1 การเขียนส่วนตัวโปรแกรม 45

3.2.2 การเขียนโปรแกรมในส่วนของโมดูล 45

บทที่ 4 การทดลองและผลการทดลอง 46

4.1 การทดลองโปรแกรม 46

4.2 ผลการทดลอง 50

4.3 สรุปผลการทดลอง 55

บทที่ 5 บทสรุปและวิจารณ์ 56

5.1 บทวิจารณ์ 56

5.2 บทสรุป 56

5.3 ปัญหาที่พบจากการทดลอง 56

5.4 แนวทางในการพัฒนา 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 1.1 แสดงขอบข่ายโครงงาน 1

รูปที่ 2.1 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI model 4

รูปที่ 2.2 โปรโตคอล XNS ซึ่งเป็นต้นแบบของ IPX/SPX
เทียบกับมาตรฐาน OSI 7

รูปที่ 2.3 โครงสร้างของหมายเลข IPv6 แบบ multicast 13

รูปที่ 2.4 แสดงคลาสของไอพี 14

รูปที่ 2.5 แสดงเครือข่าย 3 เครือข่ายที่แต่ละเครือข่ายมี IP address ของตัวเอง 16

รูปที่ 2.6 แสดงการแบ่ง Subnet ออกเป็นเครือข่ายย่อย
โดยใช้ IP address ตัวเดียว 17

รูปที่ 2.7 แสดงสถาปัตยกรรมของ TCP/IP 25

รูปที่ 2.8 แสดงการ bind IP address 26

รูปที่ 2.9 แสดงเครื่องคอมพิวเตอร์ในเครือข่าย Ethernet
มีหมายเลข IP address แต่ละเครื่อง 26

รูปที่ 2.10 แสดงการแซมปลิ้งสัญญาณ 27

รูปที่ 2.11 กระบวนการเข้าและถอดรหัสของพัลส์โค้ดมอดดูเลชั่น 27

รูปที่ 2.12 แสดงการแปลงสัญญาณพัลส์แอมปลิจูดมอดดูเลชั่น
ให้เป็นตัวเลข (การจัดระดับ) 28

รูปที่ 2.13 การนำสัญญาณพัลส์แอมปลิจูดมอดดูเลชั่นมาเข้ารหัส 29

รูปที่ 2.14 โครงสร้างของไฟล์ .wav 30

รูปที่ 2.15 แสดงองค์ประกอบของส่วนหัวไฟล์ .wav
แบบ 8 บิต โมโน 31

รูปที่ 2.16 แสดงองค์ประกอบของส่วนหัวไฟล์ .wav
แบบ 8 บิต สเตอริโอ 31

รูปที่ 2.17 แสดงองค์ประกอบของส่วนหัวไฟล์ .wav
แบบ 16 บิต สเตอริโอ 32

รูปที่ 2.18 การติดต่อของวินโดว์แอพพลิเคชันผ่านเครือข่ายอินเตอร์เน็ต 34

รูปที่ 2.19 กระบวนการแปลงสัญญาณของการ์ดเสียง 35

รูปที่ 2.20 แสดงความสัมพันธ์ของข้อมูลมัลติมีเดีย 41

รูปที่ 3.1 แผนผัง (Flowchart) การทำงานของโปรแกรม 44

รูปที่ 4.1 แสดงโปรแกรมเริ่มแรก 46

รูปที่ 4.2 ทำการกดปุ่มตอบรับอัตโนมัติ 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.3 ด้านรับทำการใส่ค่าไอพีแอดเดรสที่ต้องการติดต่อ 47

รูปที่ 4.4 ไอพีแอดเดรสของด้านรับปรากฏขึ้นที่ด้านส่ง 48

รูปที่ 4.5 โปรแกรมถามว่าต้องการรับการร้องขอหรือไม่ 48

รูปที่ 4.6 โปรแกรมทางด้านส่งขณะพูดผ่านไมโครโฟน 49

รูปที่ 4.7 โปรแกรมทางด้านรับขณะรับฟังเสียงจากด้านส่ง 49

รูปที่ 4.8 แสดงการยกเลิกการติดต่อ 50

รูปที่ 4.9 แสดงเสียงพูดคำว่า “สวัสดีครับ” ของเครื่องด้านส่ง 51

รูปที่ 4.10 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง
Celeron 300 MHz RAM 32 MB 51

รูปที่ 4.11 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง
Celeron 366 MHz RAM 64 MB 51

รูปที่ 4.12 แสดงเสียงพูดคำว่า “เฮลโล” ของเครื่องด้านส่ง 52

รูปที่ 4.13 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง
Celeron 300 MHz RAM 32 MB 52

รูปที่ 4.14 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง
Celeron 366 MHz RAM 64 MB 52

รูปที่ 4.15 แสดงเสียงพูดคำว่า “ขอให้โชคดี” ของเครื่องด้านส่ง 53

รูปที่ 4.16 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง
Celeron 300 MHz RAM 32 MB 53

รูปที่ 4.17 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง
Celeron 366 MHz RAM 64 MB 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 อธิบายค่า Scope 13

ตารางที่ 2.2 บอกจำนวนเครื่องลูกข่ายที่มีและ IP address เริ่มต้น 15

ตารางที่ 2.3 แสดงการแบ่ง Host address ออกเป็น 2 ส่วนคือ Subnet และ Host address 17

ตารางที่ 2.4 แสดงการกำหมด Subnet Mask ของ Subnet ขนาด 3 บิต 18

ตารางที่ 2.5 แสดง Subnet ทั้งของ class C 18

ตารางที่ 2.6 แสดงความหมายของโปรโตคอลแต่ละชนิด 20

ตารางที่ 2.7 แสดงจำนวนเครือข่ายและหมายเลขเครือข่ายของ Network Class ของหน่วยงานที่ไม่ต้องการ เชื่อมต่อ Internet 21

ตารางที่ 2.8 รหัสไบนารีแบบต่าง ๆ 29

ตารางที่ 4.1 แสดงผลการทดลองคุณภาพเสียงของเครื่องคอมพิวเตอร์สเปคต่าง ๆ กัน 54

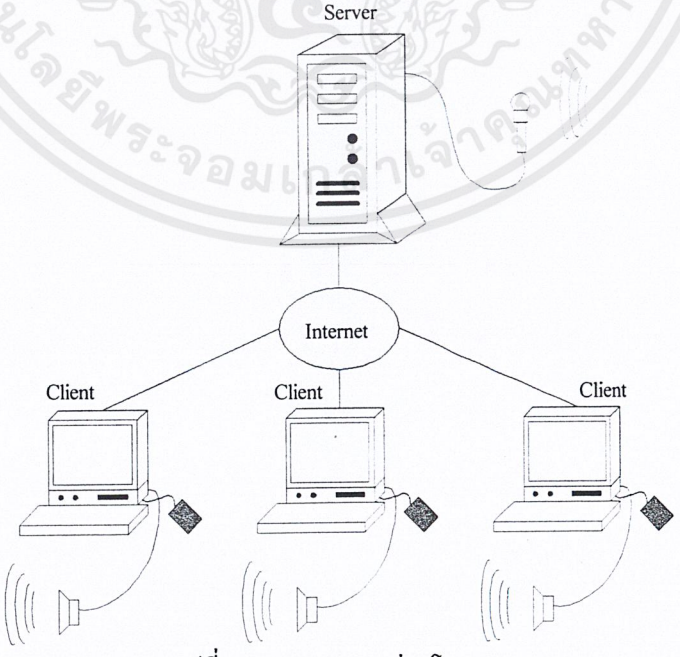
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันนี้ คอมพิวเตอร์มีการพัฒนาอย่างรวดเร็ว มีประสิทธิภาพสูงในการประมวลผลอีกทั้งระบบการสื่อสารก็ได้พัฒนาไปอย่างมาก สามารถส่งข้อมูลทั้งภาพ และเสียงได้ หรือที่เราเรียกกันว่ามัลติมีเดีย (Multimedia) ทำให้มีการพัฒนาโปรแกรมประยุกต์ (Software) ขึ้นมารองรับการใช้งานอย่างแพร่หลาย นอกจากนี้เทคโนโลยีทางการสื่อสารก็ได้พัฒนาทั้งรูปแบบและวิธีการไปอย่างรวดเร็ว อีกทั้งมีความหลากหลายมากยิ่งขึ้น ซึ่งอินเตอร์เน็ตก็เป็นรูปแบบหนึ่งของการสื่อสารข้อมูลที่ได้รับความนิยมอย่างสูงในขณะนี้ ดังนั้นโครงงานนี้จึงได้ทำการสร้างแอปปลิเคชัน (Application) เพื่อใช้ในเครือข่ายอินเตอร์เน็ตขึ้นมา

สถานีวิทยุบนเครือข่ายอินเตอร์เน็ต (Internet Radio) เป็นแอปปลิเคชันอีกอันหนึ่งที่ใช้ในการกระจายเสียงผ่านบนระบบเครือข่ายของอินเตอร์เน็ต โดยจะมีเครื่องคอมพิวเตอร์เครื่องหนึ่งซึ่งทำหน้าที่เป็นเซิร์ฟเวอร์ (Server) ทำการกระจายเสียงผ่านบนระบบเครือข่ายอินเตอร์เน็ตไปยังเครื่องคอมพิวเตอร์สำหรับผู้ใช้ ที่เรียกว่า ไคลเอ็นต์ (Client) โดยเซิร์ฟเวอร์จะทำการส่งข้อมูลเสียงไปแบบสตรีมมิ่ง (Streaming Audio) ซึ่งเป็นการส่งข้อมูลที่ทางด้านรับไม่จำเป็นต้องดาวน์โหลดไฟล์ (Download File) ข้อมูลทั้งหมดให้เสร็จก่อนแล้วจึงจะทำการเปิดไฟล์ (File) นั้นได้ กล่าวคือทางด้านรับสามารถเปิดไฟล์นั้นได้ในขณะที่ทำการดาวน์โหลดอยู่

ข้อดีของอินเตอร์เน็ตเรดิโอ (Internet Radio) คือ ค่าใช้จ่ายในการจัดตั้งไม่สูงมากเมื่อเปรียบเทียบกับการจัดตั้งสถานีคลื่นวิทยุแบบปกติ แต่สถานีวิทยุบนเครือข่ายอินเตอร์เน็ตมีข้อจำกัดอยู่ คือ เมื่อต้องการรับฟังจะต้องมีเครื่องคอมพิวเตอร์ที่ต่อกับเครือข่ายอินเตอร์เน็ตอยู่ด้วยจึงจะสามารถรับฟังได้



รูปที่ 1.1 แสดงขอบข่ายโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีหรือหลักการ

2.1 อินเตอร์เน็ต (Internet)

ระบบเครือข่ายคอมพิวเตอร์ และอินเตอร์เน็ตเวิร์คกิ้ง (Internetworking)

ระบบเครือข่ายคอมพิวเตอร์(Computer Network) คือระบบการเชื่อมต่อระหว่างระบบปลายทาง (End-System) ซึ่งระบบปลายทางเป็นอิสระต่อกัน (Autonomous) ระบบปลายทางสามารถเป็นได้ตั้งแต่ ไมโครคอมพิวเตอร์ไปจนกระทั่งถึงซุปเปอร์คอมพิวเตอร์ (Supercomputer) ขนาดใหญ่เพื่อจุดมุ่งหมาย ในการแลกเปลี่ยนข้อมูลและการแบ่งปันทรัพยากรของระบบ เช่น ไฟล์ (File) , เครื่องพิมพ์ (Printer) , โมเด็ม (Modem) , ตลอดจนการให้บริการฐานข้อมูลร่วม (Sharing Database)

อินเตอร์เน็ตเวิร์คกิ้ง หรืออินเตอร์เน็ต (Internet) คือการเชื่อมต่อของระบบเครือข่าย 2 เครือ ข่ายขึ้นไป ดังนั้นคอมพิวเตอร์บนระบบเครือข่ายหนึ่งก็สามารถติดต่อกับคอมพิวเตอร์บนระบบเครือข่าย อื่น ๆ ได้

องค์ประกอบของอินเตอร์เน็ต

อินเตอร์เน็ตเป็นเครือข่ายคอมพิวเตอร์ชนิดหนึ่งที่ใช้โปรโตคอล TCP/IP (Transmission Control Protocol /Internet Protocol) เป็นมาตรฐานการทำงานของระบบ ดังนั้นถ้ามีเครือข่ายคอมพิวเตอร์ ที่ใช้โปรโตคอล TCP/IP อยู่แล้วก็จะเป็นการสะดวกและง่ายต่อการเชื่อมต่อเข้ากับระบบอินเตอร์เน็ต ระบบการทำงานของเครือข่ายโปรโตคอล TCP/IP โดยเฉพาะสำหรับเครือข่ายอินเตอร์นั้นจะแบ่งกลุ่ม ของแฟกเตอร์หรือฟังก์ชั่นการทำงานออกเป็น 6 กลุ่มใหญ่ ๆ ซึ่งการติดตั้งอุปกรณ์ต่าง ๆ เองก็ต้องคำนึง ถึงแฟกเตอร์ 6 กลุ่มด้วยกันคือ

ชนิดของสถานี ระบบเครือข่าย TCP/IP ส่วนใหญ่จะประกอบด้วยเครื่องคอมพิวเตอร์ที่ทำหน้าที่ แตกต่างกันอยู่สองชนิด คือ เครื่องที่ทำหน้าที่ให้บริการที่เรียกว่า โฮสต์ (Host) หรือเซิร์ฟเวอร์ (Server) และเครื่องคอมพิวเตอร์สำหรับผู้ใช้ทั่วไปเรียกว่า เทอร์มินอล (Terminal) หรือไคลเอ็นต์ (Client) โดยที่ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นสถานีให้บริการนั้นจะเป็นเครื่องที่คอยให้บริการแก่ผู้ใช้ในด้านต่าง ๆ ไม่ว่าจะเป็นแหล่งเก็บรวบรวมข้อมูล (Data Sharing) การให้บริการโปรแกรมประยุกต์ต่างๆ (Application) หรือการให้บริการการใช้งานระบบประมวลผลกลาง (CPU Time Sharing) เป็นต้น ดังนั้นคุณสมบัติ โดยทั่วไปทั้งด้านฮาร์ดแวร์และด้านซอฟต์แวร์ของเครื่องโฮสต์ หรือเซิร์ฟเวอร์ จึงมีคุณสมบัติที่ดีกว่า เครื่องคอมพิวเตอร์สำหรับผู้ใช้มาก

ระบบไอพีแอดเดรส (IP Address)

การสื่อสารข้อมูลในระบบเครือข่ายคอมพิวเตอร์ เป็นการสื่อสารในลักษณะที่เฟรมข้อมูลของแ ละการสื่อสารเป็นคนกำหนดเส้นทางที่สื่อสารเอง คือเมื่อมีการร้องขอติดต่อสื่อสารข้อมูลของเครื่อ คอมพิวเตอร์คู่ใดเกิดขึ้น เครื่องคอมพิวเตอร์เครื่องนั้นก็จะทำการสร้างเฟรมข้อมูลขึ้นมาแล้วค่อยส่งออ ไปในระบบเครือข่าย โดยที่เฟรมข้อมูลนั้นจะมีส่วนของแอดเดรสที่อยู่ในส่วนอ้างอิงทำกับการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Header) ที่จะบอกว่า เฟรมข้อมูลนี้เป็นของเครื่องคอมพิวเตอร์เครื่องใดที่กำลังส่ง และจะส่งไปยังเครื่องคอมพิวเตอร์เครื่องใด

ดังนั้นการที่เครื่องคอมพิวเตอร์ต่างๆ จะติดต่อสื่อสารกันในระบบเครือข่ายโปรโตคอล TCP/IP จะต้องมีการกำหนดค่าไอพีแอดเดรสให้แต่ละสถานีที่จะสื่อสารกันด้วย นอกเหนือจากค่า MAC Address ที่มีอยู่ในแต่ละเครื่อง เพราะค่าไอพีแอดเดรสนั้นจะเป็นค่าอ้างอิงในเฟรมข้อมูลที่สื่อสารในเครือข่าย ซึ่งจะมี 2 ชนิด คือ

- ไอพีแอดเดรสต้นทาง (Source IP Address)
- ไอพีแอดเดรสปลายทาง (Destination IP Address)

ระบบโปรโตคอลหาเส้นทาง (IP Routing Protocol)

ลักษณะการติดต่อสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ใด ๆ ในระบบเครือข่ายอินเตอร์เน็ตนั้น โดยทั่วไปแล้วมี 2 ลักษณะ คือ

1. การเชื่อมต่อภายในเครือข่ายท้องถิ่น (LAN)
2. การเชื่อมต่อระหว่างเครือข่ายท้องถิ่นหนึ่งกับเครือข่ายท้องถิ่นอื่น โดยอาจมีการบริการของระบบเครือข่ายระยะไกล (WAN) เข้ามาเกี่ยวข้องด้วย

ในการเชื่อมต่อจำเป็นต้องใช้อุปกรณ์ที่เรียกว่า เราเตอร์ (Router) โดยเราเตอร์จะเกี่ยวข้องกับระบบทำงานที่เรียกว่า โปรโตคอลหาเส้นทาง (Routing Protocol) ซึ่งจะทำหน้าที่ตรวจสอบและจัดการเกี่ยวกับเส้นทางในการสื่อสารข้อมูลทั้งหมดของระบบ

ระบบชื่อกลุ่ม (Domain Name System)

มีการออกแบบระบบชื่อของสถานีบริการต่าง ๆ บนเครือข่ายอินเตอร์เน็ตในรูปลักษณะตัวอักษร เพื่ออำนวยความสะดวกต่อการใช้งานของยูสเซอร์ ระบบ DNS (Domain Name System) เป็นระบบซอร์ฟแวร์ที่ทำหน้าที่ในการจัดสรรและบริการในส่วนการเปรียบเทียบค่าระหว่างชื่อตัวอักษรกับค่าไอพีแอดเดรสของเครื่องสถานีต่าง ๆ บนอินเตอร์เน็ต

โปรแกรมประยุกต์บนเครือข่ายอินเตอร์เน็ต (Application)

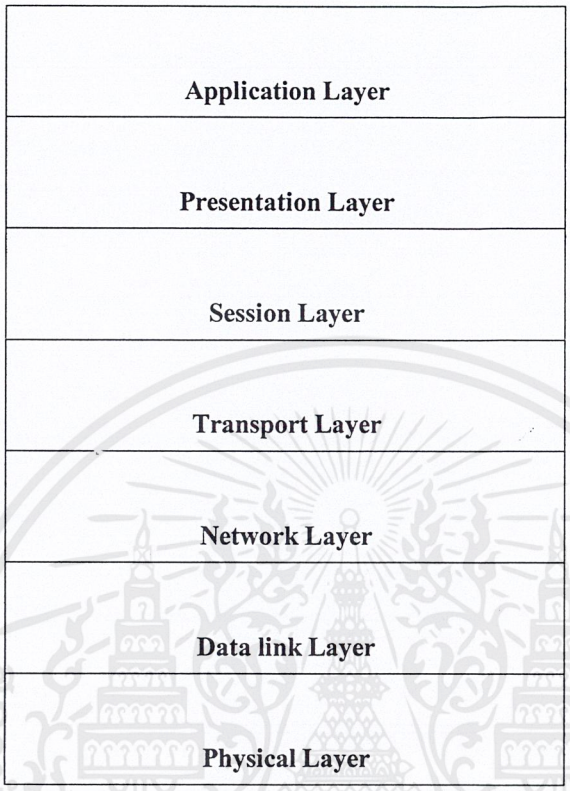
ระบบเครือข่ายอินเทอร์เน็ตเป็นระบบเครือข่ายขนาดใหญ่ที่มีการเชื่อมโยงกันทั่วโลก ดังนั้นการใช้งานโปรแกรมประยุกต์ต่าง ๆ บนระบบอินเตอร์เน็ตจึงมีลักษณะพิเศษแตกต่างจากการใช้งานบนระบบเครือข่ายท้องถิ่นทั่วไป คือจะมีโปรแกรมประยุกต์มากมายหลายชนิด เช่น ระบบจดหมายอิเล็กทรอนิกส์ (E-mail) ระบบข่าวสารรวม (Usenet) และระบบเครือข่ายใยแมงมุม (World-Wide-Web) เป็นต้น โดยที่แต่ละชนิดมีการใช้งานที่แตกต่างกันมาก ระบบความปลอดภัย (Security) มีหน้าที่ป้องกันไม่ให้มีการลักลอบเข้ามาใช้ หรือ ทำลายข้อมูลที่สำคัญ

2.2 OSI โมเดล

องค์กรมาตรฐานสากล ISO (International Organization for Standardization) ได้กำหนดมาตรฐานของเครือข่าย โดยจัดแบ่งกิจกรรมของเครือข่าย ออกเป็นงานย่อยๆ และกำหนดโมเดลแบ่งเป็นขั้นๆ ตามลำดับเรียกว่ามาตรฐาน OSI (Open System Interconnection) โดยที่จะแบ่งกิจกรรมที่ซับซ้อนในเครือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข่าย ออกเป็นงานย่อยๆ ก็จะช่วยในการออกแบบ และการใช้งานเครือข่ายรวมถึงการเชื่อมโยงกันเป็นไปได้ด้วยความสะดวก และมีวิธีการทำงานอยู่ในกรอบเดียวกัน (รูปที่ 2.1)



รูปที่ 2.1 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI model

ในแต่ละขั้นของ OSI model จะมีการติดต่อสื่อสารกันเป็นขั้นๆ ตามลำดับลงมาเช่น Application layer ก็จะติดต่อสื่อสารกับ Presentation Layer ตามลำดับไปจนถึงขั้นแรกสุดคือ Physical Layer

Application layer เป็นชั้นบนสุดของ โมเดลเป็นส่วนที่จะทำให้การติดต่อระหว่างเครือข่ายกับผู้ใช้เป็นไปได้ตามต้องการ ตัวอย่างแอปพลิเคชั่นของเครือข่ายเช่น ระบบ E-MAIL การโอนถ่ายข้อมูล (File transfer) การขอใช้ระบบคอมพิวเตอร์ในเครือข่ายเป็นต้น

Presentation Layer เป็นชั้นที่ทำหน้าที่ตกลงกับคอมพิวเตอร์อีกด้านหนึ่งว่า การรับส่งข้อมูลในระดับโปรแกรมประยุกต์จะมีขั้นตอนและข้อบังคับอย่างไร ข้อมูลที่ทำการรับส่งในชั้นนี้จะอยู่ในรูปแบบของข้อมูลระดับสูง ซึ่งอยู่ในรูปแบบของคำสั่งที่มีกฎ (Syntax) บังคับอย่างแน่นอน

Session Layer เป็นชั้นที่จัดการในเรื่อง "การติดต่อแต่ละครั้ง" หรือ session ให้ระบบคอมพิวเตอร์ทั้ง 2 ฝั่ง โดยทำหน้าที่ตั้งแต่เริ่มการติดต่อ ดูแลในการส่งผ่านข้อมูล ในการติดต่อครั้งนั้นๆ เป็นไปได้โดยไม่มีปัญหาจนถึงเลิกการติดต่อเมื่อเสร็จงาน

Transport Layer ทำหน้าที่ควบคุมปริมาณ และรายละเอียดวิธีการรับส่งข้อมูล ให้เป็นไปตามกำหนดที่ตั้งไว้ และจัดการให้การเชื่อมโยงเครือข่ายเป็นไปด้วยความราบรื่น Transport Layer จะเป็นชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้าย ที่จัดการเรื่องเส้นทางในการส่งข้อมูล และจัดการตรวจสอบความผิดพลาดของข้อมูล ซึ่งส่วนของ TCP (Transmission Control Protocol) ในโปรโตคอล TCP/IP ทำงานที่ระดับนี้

Network Layer ทำหน้าที่ควบคุมวิธีการส่งผ่านข้อมูลระหว่างเครือข่ายให้ถูกต้อง และเป็นไปตามเส้นทางที่กำหนด โดยจะจัดการส่งผ่าน packet ข้อมูล ผ่านอุปกรณ์ต่างๆ ไปยังเครือข่ายย่อยได้อย่างถูกต้องตามที่ต้องการ นอกจากนี้ยังจัดการดูแลเส้นทางในการส่งข้อมูล (Routing table) และกั้นหรือกรอง packet ข้อมูลที่ส่งไปยังเครือข่ายเดียวกันไม่ให้ข้ามไปยังเครือข่ายอื่น ซึ่งจะช่วยลดปริมาณข้อมูลที่จะวิ่งบนเครือข่ายได้ส่วนหนึ่ง โปรโตคอล IP และ IPX เป็นโปรโตคอลที่ทำงานอยู่ใน layer นี้

Data link Layer ทำหน้าที่เรียกใช้หรือกำหนดช่องทาง ในการส่งข้อมูลที่ถูกต้อง เช่น Ethernet, Token ring หรือ FDDI เป็นต้น รวมถึงการลำดับและอัตราการรับส่งข้อมูลหรือ flow control และสถานที่ที่จะส่งข้อมูลไป (address) ทั้งนี้ Data link layer จะเป็นชั้นแรกที่จัดการแปลงข้อมูลจาก bit ให้เป็น packet โดยจะมีการเพิ่มข้อมูลเพื่อตรวจสอบความถูกต้อง ในกรณีที่ส่งข้อมูลออกไป หรือในกรณีที่อ่านข้อมูลที่เข้ามาก็จะตรวจสอบผ่าน checksum เพื่อดูว่าข้อมูลที่จะได้รับมาถูกต้องครบถ้วน และถ้าได้รับ packet ข้อมูลที่ไม่ถูกต้องก็จะไม่เอาข้อมูลนั้นไปใช้งาน และจะบอกให้ต้นทางส่งข้อมูลเดิมมาใหม่

Physical Layer รับผิดชอบดูแลในรายละเอียดในการส่งข้อมูลในด้านฮาร์ดแวร์ เช่นการควบคุม Network Interface Card การส่งสัญญาณผ่านสายสัญญาณแบบต่างๆ การเชื่อมต่อเข้ากับเครือข่ายแบบต่างๆ โดยใช้ Physical Layer จะจัดสร้างสัญญาณทางไฟฟ้า,สัญญาณเสียง หรือ สัญญาณที่จำเป็นในการสื่อสารโดยตรงเนื่องจาก Network Layer เป็นชั้นที่โปรโตคอล IP , TCP/IP ทำงานอยู่จะกล่าวโดยละเอียดต่อไป

การใช้โปรแกรมข้อมูลร่วมกัน สามารถจะทำให้ User หลายๆคนใช้โปรแกรมข้อมูลร่วมๆกันได้ และเป็นการประหยัดเวลาในการติดตั้งโปรแกรม โดยทำการติดตั้งโปรแกรมใช้งานไว้ที่ไฟล์เซิร์ฟเวอร์ และเก็บข้อมูลไว้ที่ไฟล์เซิร์ฟเวอร์ เพียงตัวเดียว เครื่องเวิร์กสเตชันไม่จำเป็นต้องมีดิสก์ไดร์ฟ หรือฮาร์ดดิสก์ ก็สามารถใช้งานโปรแกรมและข้อมูลจากเครื่องไฟล์เซิร์ฟเวอร์ได้

การใช้ฮาร์ดแวร์ร่วมกัน ระบบเวิร์กเป็นระบบที่สามารถจะใช้อุปกรณ์จากฮาร์ดแวร์ร่วมกันได้ดี เช่น ระบบเน็ตเวิร์คของท่านมีเครื่องเวิร์กสเตชันอยู่ 12 เครื่อง แต่ละเครื่องมีงานที่ต้องพิมพ์บนเครื่องพิมพ์เลเซอร์ ถ้าไม่มีระบบเน็ตเวิร์คจะต้องใช้งบประมาณในการซื้อเครื่องพิมพ์อย่างน้อย 4-6 เครื่อง แต่สำหรับเน็ตเวิร์คแล้ว สามารถใช้เครื่องพิมพ์เพียง 1-2 เครื่องโดยการทำเป็น Print Server คอยรับงานพิมพ์จากเครื่องที่ 12 เครื่อง นอกจากนี้ยังใช้งานโมเด็มและฮาร์ดดิสก์ร่วมกันได้ด้วย

การกระจายการประมวลผล เป็นที่ทราบแล้วว่าโปรแกรมใช้งานต่างๆเช่น Excel, word, Data Base ฯลฯได้ถูกติดตั้งในฮาร์ดดิสก์ของเครื่องไฟล์เซิร์ฟเวอร์ ส่วนเครื่องเวิร์กสเตชัน จะมีเพียงจอภาพ ซีพียู และคีย์บอร์ด สำหรับทำงานเมื่อ User ของเวิร์กสเตชันเครื่องหนึ่งที่ต้องการ ใช้งานโปรแกรม Word เครื่องไฟล์เซิร์ฟเวอร์จะทำงาน Copy โปรมแกรม Word จากฮาร์ดดิสก์ของไฟล์เซิร์ฟเวอร์มาไว้ยังหน่วยความจำ ram ของเวิร์กสเตชันเครื่องนั้นต่อจากนี้เป็นหน้าที่ของ ซีพียู ภายในเครื่องเวิร์กสเตชันทำการ ram โปรแกรม Word ต่อไปเป็นการกระจายการประมวลผลอย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อสื่อสารแบบรวดเร็ว ด้วยมาตรฐาน IEEE 802.3 สามารถส่งข้อมูลด้วยอัตราเร็วความเร็ว 10 เมกะบิตต่อวินาที (โดยใช้หลักการส่งข้อมูลแบบ Base-band) ความเร็วในระดับนี้จะทำให้การปรับปรุง (Update) ข้อมูลเป็นไปอย่างรวดเร็วโดยเฉพาะการนำอิเล็กทรอนิกส์เมล์ (E-MAIL) มาใช้บนระบบเน็ตเวิร์ค ปัจจุบันระบบ Ethernet สามารถจะส่งข้อมูลได้ด้วยความเร็ว 10 จิกะบิตต่อนาที (Gbit / s)

<u>การติดต่อสื่อสารระหว่างเวิร์กสเตชัน</u>

ในขณะที่เครื่องเวิร์กสเตชันแต่ละเครื่องกำลังทำงานอยู่นั้น บังเอิญต้องมีการปิด (Shut Down) เครื่องไฟล์เซิร์ฟเวอร์เพื่อเปลี่ยนฮาร์ดดิสก์หรือ อุปกรณ์อื่นๆ สามารถส่งข้อความไปยังเครื่องเวิร์กสเตชันทุกเครื่องเพื่อเตือนให้เซฟข้อมูลภายใน 10 นาที เช่น Shut Down File Server Within 10 minute. Please save data now. ข้อความเหล่านี้จะปรากฏที่หน้าจอเครื่องเวิร์กสเตชันทุกเครื่องทันทีถ้าเวลาการเซฟน้อยไปจะขอเพิ่มเป็น 20 นาที เครื่องเวิร์กสเตชันสามารถจะส่งข้อความไปบอกไฟล์เซิร์ฟเวอร์ได้เช่น Please give me 20 minute for save data…OK

Peer to Peer เป็นระบบที่เครื่องเวิร์กสเตชันทุกเครื่องบนระบบเน็ตเวิร์กมีฐานะเทียมเท่ากันคือทุกเครื่องสามารถจะใช้ไฟล์ในเครื่องอื่นได้ และสามารถใช้เครื่องอื่นมาใช้ไฟล์ของตนเองได้เช่นกันระบบ Peer to Peer มีลักษณะการทำงานแบบ Distributed system โดยกระจายทรัพยากรต่างๆไปสู่เวิร์กสเตชันอื่นๆแต่จะมีปัญหาเรื่องการรักษาความปลอดภัย เพราะข้อมูลที่เป็นความลับจะถูกส่งออกไปสู่เวิร์กสเตชันอื่นเช่นกัน

โปรแกรมที่มีความสามารถทาง Peer to Peer และเป็นที่รู้จักกันดีคือ Windows for Workgroup และ Personal NetWare

Client/Server เป็นการทำงานแบบ Distributed Processing หรือการประมวลผลแบบกระจาย โดยจะแบ่งกันประมวลผลระหว่างเครื่อง เซิร์ฟเวอร์กับเครื่องเวิร์กสเตชันและเมื่อใดที่เครื่องเวิร์กสเตชันต้องการผลลัพธ์ของข้อมูลบางส่วน จะมีการเรียกใช้ไปยังเครื่องเซิร์ฟเวอร์นำเฉพาะข้อมูลบางส่วนเท่านั้นส่งกลับมาให้ที่เครื่องเวิร์กสเตชันเพื่อทำการคำนวณข้อมูลต่อไป

2.3 โปรโตคอลและไอพีแอดเดรส (IP Address)

ในการใช้งานเครือข่ายคอมพิวเตอร์ทั่วไปหรือในเครือข่ายอินเตอร์เน็ตก็ตาม จะมีการส่งข้อมูลไปมาระหว่างเครื่องคอมพิวเตอร์หรือข้ามเครือข่ายออกไป ระบบคอมพิวเตอร์ที่เชื่อมต่ออยู่ในแต่ละเครือข่ายอาจจะใช้ฮาร์ดแวร์และซอฟต์แวร์ที่เหมือนกันหรือแตกต่างกันได้ ดังนั้นการที่ทำให้สามารถส่งผ่านข้อมูลถึงกันและตีความได้อย่างถูกต้องตรงกันจะต้องมีการกำหนดกลไกในการสื่อสารกันเสียก่อน เรียกว่าจะต้องกำหนดระเบียบวิธีในการติดต่อกันให้ตรงกัน เปรียบเสมือนกับการสื่อสารกันของมนุษย์เรา ถ้าเราต้องการจะติดต่อกับผู้คนต่างเชื้อชาติให้เข้าใจกันได้ถูกต้องตรงกันก็จะต้องตกลงกันเสียก่อนว่า จะติดต่อกันอย่างไร ด้วยภาษาใดที่จะเข้าใจกันได้เช่น ปัจจุบันมีการใช้ภาษาอังกฤษเปรียบเสมือนเป็นภาษามาตรฐานในการสื่อสารกันของมนุษย์ได้ ถ้าพูดในแง่การสื่อสารข้อมูล เราก็พูดได้ว่า ภาษาอังกฤษเป็นโปรโตคอลในการสื่อสารข้อมูลของมนุษย์ที่มีการใช้งานอย่างแพร่หลาย เช่นเดียวกันกับโปรโตคอล TCP/IP เป็นโปรโตคอลหลักที่ใช้ในการสื่อสารข้อมูลในเครือข่ายอินเตอร์เน็ต

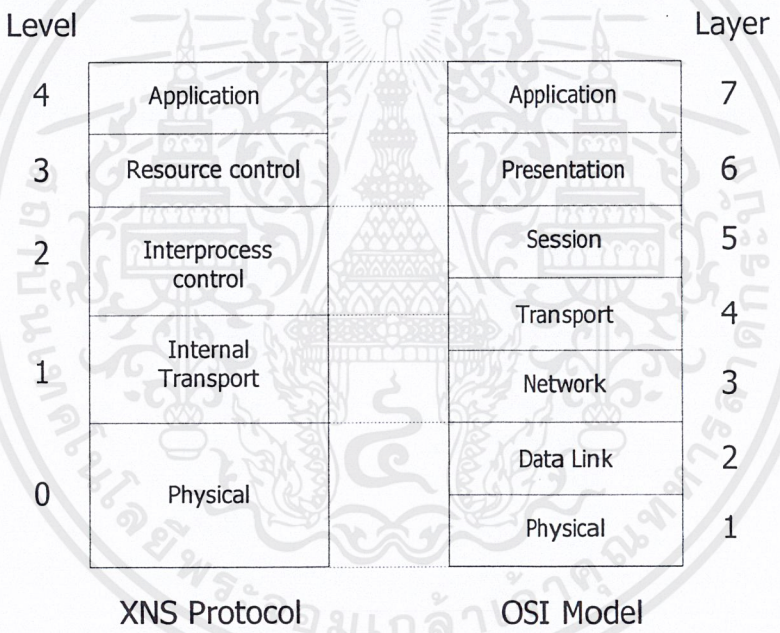
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 โปรโตคอลคืออะไร

จากที่กล่าวมาแล้วพอสรุปได้ว่า โปรโตคอล (Protocol) คือระเบียบวิธีที่กำหนดขึ้นมาสำหรับการสื่อสารข้อมูล ให้สามารถส่งผ่านข้อมูลไปยังปลายทางได้อย่างถูกต้อง ในปัจจุบันโปรโตคอลในการสื่อสารข้อมูลก็มีอยู่หลายโปรโตคอลนอกเหนือจาก TCP/IP คล้ายกับภาษาต่างๆในโลกนี้ ที่นอกจากภาษาอังกฤษแล้วก็ยังมีภาษาจีน ญี่ปุ่น ฝรั่งเศส เยอรมัน และอื่นๆอีกมากมาย ในด้านของโปรโตคอลสื่อสารข้อมูลก็เช่นกัน ซึ่งได้มีการออกแบบโปรโตคอลอื่นๆขึ้นมาใช้งานอีกมาก เช่น โปรโตคอล IPX/SPX, โปรโตคอล NetBIOS และ โปรโตคอล AppleTalk เป็นต้น

<u>โปรโตคอล IPX/SPX</u>

โปรโตคอล IPX/SPX พัฒนาขึ้นโดยบริษัท Novell ซึ่งเป็นผู้พัฒนาระบบปฏิบัติการการเครือข่าย Netware กำเนิดของโปรโตคอล IPX/SPX นี้มีการพัฒนาขึ้นมาจากโปรโตคอลที่ชื่อว่า XNS พัฒนาขึ้นโดยบริษัท Xerox Corporation ที่ Palo Alto Research Center (PARC) อีกทีหนึ่ง โครงสร้างการทำงานของโปรโตคอล XNS เมื่อเทียบกับมาตรฐาน OSI จะมี ลักษณะดังในรูปที่ 2.2



รูปที่ 2.2 โปรโตคอล XNS ซึ่งเป็นต้นแบบของ IPX/SPX เทียบกับมาตรฐาน OSI

โปรโตคอล IPX/SPX แบ่งออกเป็นโปรโตคอลหลัก 2 โปรโตคอลคือ Internetwork Packet Exchange (IPX) และ Sequenced Packet Exchange (SPX) โดยโปรโตคอล IPX ทำหน้าที่ในระดับ network layer ตามมาตรฐาน OSI มีกลไกการส่งผ่านข้อมูลแบบ connectionless, unreliable หมายความว่าเมื่อมีการส่งข้อมูล โดยไม่ต้องการเชื่อมต่อกันระหว่าง host กับเครื่องที่ติดต่อกันอย่างถาวร (host หมายถึงเครื่องเซิร์ฟเวอร์ที่ให้บริการใดๆในเครือข่าย) และไม่ต้องการรอสัญญาณยืนยันการรับข้อมูลจากปลายทาง โดยตัวโปรโตคอลจะพยายามส่งข้อมูลนั้นไปยังปลายทางให้ดีที่สุด สำหรับโปรโตคอล SPX ทำหน้าที่ในระดับ transport layer ตามมาตรฐาน OSI โดยส่งผ่านข้อมูลในแบบตรงข้ามกับโปรโตคอล IPX คือ ต้องมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อเชื่อมโยงกันก่อนและมีการส่งผ่านข้อมูลที่เชื่อถือได้ ด้วยการตรวจสอบสัญญาณยืนยันการรับส่งข้อมูลจากปลายทาง

คอนเน็กชั่นเลส (Connectionless) และ คอนเน็กชั่นโอเรียน (Connection-oriented)

การทำงานของโปรโตคอลในแบบ Connectionless คือ ขบวนการติดต่อกันระหว่างระบบสองระบบที่ผู้ส่งข้อมูลไม่ต้องตรวจสอบว่าผู้รับพร้อมรับข้อมูลหรือไม่ และการส่งข้อมูลไปให้นั้นไม่มีการตรวจสอบว่าผู้รับได้รับข้อมูลหรือไม่ ส่วนการทำงานของโปรโตคอลในแบบ Connection-oriented จะเป็นการเชื่อมต่อส่งข้อมูล โดยทั้งสองด้านจะต้องติดต่อกันโดยมีขบวนการยืนยันการติดต่อระหว่างกันขึ้นมาก่อน จากนั้นจึงจะเริ่มแลกเปลี่ยนข้อมูลกันได้ และในทางกลับกันเมื่อเลิกติดต่อก็จะมีการส่งสัญญาณในการสอบทานความถูกต้องของข้อมูล เพื่อให้ทราบแน่ชัดว่ารับส่งข้อมูลนั้นสำเร็จหรือไม่

โปรโตคอลเน็ตไบออส (NetBIOS)

โปรโตคอล NetBIOS หรือ Network Basic Input/Output System นั้นความจริงแล้วไม่ได้เป็นโปรโตคอลที่แท้จริง แต่สามารถเรียกว่าเป็นโปรโตคอลที่สื่อสารข้อมูลได้ เพราะมีการกำหนดระเบียบวิธีในการสื่อสารข้อมูลไว้เช่นกัน ซึ่งที่จริงแล้ว NetBIOS เป็นไลบรารีของกลุ่มคำสั่งควบคุมระบบเครือข่ายหรือ API (Application Programming Interface) ที่ออกแบบมาให้โปรแกรมเมอร์สามารถสร้างแอพพลิเคชันโปรแกรมด้านเครือข่ายได้ง่าย โดยไม่ต้องเขียนโปรแกรมส่วนควบคุมเครือข่ายเองให้ยุ่งยาก ซึ่งโปรแกรมเมอร์จะสามารถเรียกใช้งานชุดคำสั่งต่างๆในการจัดการเครือข่ายได้เป็นมาตรฐาน ถ้าเทียบมาตรฐาน OSI แล้ว NetBIOS จะทำงานที่ระดับ session layer แต่ไม่มีส่วนในการระบุเส้นทางส่งผ่านข้อมูล (route service) ดังนั้นจึงทำให้ NetBIOS ไม่สามารถใช้งานเป็นโปรโตคอลสำหรับเชื่อมโยงเครือข่ายที่แตกต่างกันได้

ตัวชุดไลบรารีของกลุ่มคำสั่งควบคุมระบบเครือข่ายของ NetBIOS นี้มุ่งเน้นให้ใช้งานในระดับเครือข่ายคอมพิวเตอร์ส่วนบุคคลหรือพีซีเท่านั้น โดยบริษัท Sytek เป็นผู้ริเริ่มพัฒนา NetBIOS ขึ้นราวปี ค.ศ. 1984 จากนั้นถูกบริษัทไมโครซอฟท์ซื้อไปพัฒนาต่อ เพื่อให้ใช้งานกับกลุ่มโปรแกรมด้านเครือข่ายของไมโครซอฟท์เอง ในปัจจุบันก็มีการใช้โปรโตคอล NetBIOS ในลักษณะของ workgroup computing กันมาก โดยจะเชื่อมต่อเครือข่ายของเครื่องที่ใช้ระบบปฏิบัติการ Windows เป็นหลัก ช่วยให้ผู้ใช้งานสามารถแชร์เครื่องพิมพ์หรือฮาร์ดดิสก์ได้ง่าย

โปรโตคอลแอปเปิ้ลทอล์ก (AppleTalk)

จุดเริ่มต้นของโปรโตคอล AppleTalk เกิดขึ้นราวปี ค.ศ. 1983 และ 1984 ซึ่งช่วงนั้นบริษัท Apple Computer ต้องการออกแบบชุดโปรโตคอลสื่อสารข้อมูลของตนเองขึ้น เพื่อใช้เชื่อมโยงเครือข่ายของเครื่องแมคอินทอช (Macintosh) และสามารถแชร์อุปกรณ์ต่างๆ เช่น เครื่องพิมพ์เลเซอร์แบบ PostScript ที่ใช้งานเป็นเครือข่ายได้ นอกจากนี้ยังขยายไปสู่การเชื่อมโยงเป็นเครือข่ายของเซิร์ฟเวอร์, เครื่องพิมพ์, Gateway และ Router ของผู้ผลิตรายอื่นๆด้วย ต่อจากนั้นเครื่องแมคอินทอชและอุปกรณ์ประกอบต่างๆที่บริษัท Apple ผลิตออกมาก็ได้มีการเพิ่มส่วนของฮาร์ดแวร์และซอฟต์แวร์ให้ให้สามารถรองรับโปรโตคอล AppleTalk ได้ รวมถึงระบบปฏิบัติการ MacOS รุ่นใหม่ๆ ก็ได้มีการผนวกฟังก์ชันให้รองรับโปรโตคอล AppleTalk ได้เช่นกัน ทำให้กลุ่มผู้ใช้เครื่อง แมคอินทอชสามารถเชื่อมโยงกันเป็นเครือข่ายได้ง่ายโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ต้องไปหาซื้ออุปกรณ์หรือซอฟต์แวร์พิเศษอื่นเพิ่มเติมอีก จึงทำให้การใช้งานโปรโตคอล AppleTalk ได้แพร่หลายอย่างมากในหมู่ผู้ใช้เครื่องของ Apple

โปรโตคอล AppleTalk ถูกออกแบบมาให้ทำงานเป็นเครือข่ายในแบบ peer-to-peer ซึ่งถือว่าเครื่องทั้งหมดที่เชื่อมต่ออยู่ในเครือข่ายสามารถเป็นเซิร์ฟเวอร์ได้ทุกเครื่องโดยไม่ต้องจัดให้บางเครื่องทำหน้าที่เป็นเซิร์ฟเวอร์ที่ให้บริการโดยเฉพาะขึ้นมา ต่อมาในเดือนมิถุนายน ค.ศ. 1989 บริษัท Apple ได้พัฒนาโปรโตคอล AppleTalk ให้สนับสนุนเครือข่ายที่ใหญ่มากขึ้นได้ สามารถมีเครื่องลูกข่ายและอุปกรณ์ที่เชื่อมต่อในเครือข่ายได้มากกว่าเดิม เรียกว่าเป็นโปรโตคอล AppleTalk Phase 2 นอกจากนี้ยังเพิ่มโปรโตคอลที่ทำให้สามารถเชื่อมต่อกับเครือข่ายแบบ Ethernet และ Token Ring ได้ โดยบริษัท Apple เรียกโปรโตคอลของตนว่า EtherTalk และ TokenTalk ตามลำดับ

<u>โปรโตคอลทีซีพี/ไอพี (TCP/IP)</u>

โปรโตคอล TCP/IP เป็นชื่อเรียกของชุดโปรโตคอลที่สำคัญ มีการใช้งานกันอย่างแพร่หลายตามการขยายตัวของอินเตอร์เน็ต/อินทราเน็ต ความจริงแล้วโปรโตคอล TCP/IP เป็นกลุ่มของโปรโตคอลหลายตัวที่ประกอบกันเป็นชุดให้ใช้งาน โดยมีคำเต็มว่า Transmission Control Protocol /Internet Protocol ซึ่งจากชื่อเต็มทำให้เราเห็นว่าอย่างน้อยก็มีโปรโตคอลประกอบกันทำงานร่วมกัน 2 โปรโตคอลคือ TCP/IP

ตัวอย่างของกลุ่มโปรโตคอลในชุดของ TCP/IP ที่เราพบและใช้งานบ่อยๆ (ส่วนใหญ่เราจะไม่ได้ใช้งานตรงๆ แต่ใช้งานผ่านแอพพลิเคชันต่างๆ หรือใช้งานโดยทางอ้อม) เช่น Internet Protocol (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), Transport Control Protocol (TCP), Simple Mail Transfer Protocol (SMTP) และ Domain Name System (DNS) เป็นต้น

โปรโตคอลที่มีบทบาทสำคัญในการทำงานในเครือข่ายอินเตอร์เน็ต คือ Internet Protocol (โปรโตคอล IP) เนื่องจาก เมื่อโปรโตคอลอื่นๆที่ต้องการส่งผ่านข้อมูลข้ามเครือข่ายในอินเตอร์เน็ตนั้น จะต้องอาศัยการผนึกข้อมูล (encapsulation) ไปกับโปรโตคอล IP ที่มีกลไกการระบุเส้นทาง (route service) ผ่าน Gateway หรือ Router เพื่อนำข้อมูลไปยังเครือข่ายและเครื่องปลายทางที่ถูกต้อง เนื่องจากกลไกการระบุเส้นทางจะทำงานที่โปรโตคอล IP เท่านั้น และด้วยเหตุนี้เราจึงเรียกว่า IP ว่าเป็นโปรโตคอลที่มีความสามารถระบุเส้นทางการส่งผ่านของข้อมูลได้ (routable)

ก่อนที่จะทำความเข้าใจเรื่องของกลไกและโครงสร้างของโปรโตคอล TCP/IP การระบุเส้นทางในการส่งผ่านข้อมูลและการผนึกข้อมูล จะขอกล่าวถึงเรื่องพื้นฐานของเครือข่ายและส่วนประกอบอื่นๆที่มีอยู่ในเครือข่ายเสียก่อน

2.3.2 การอ้างอิงอุปกรณ์ในเครือข่าย

แนวความคิดหลักของระบบเครือข่ายคอมพิวเตอร์ก็คือ การเชื่อมโยงอุปกรณ์เข้าด้วยกัน ไม่ว่าจะเป็นเครื่องเซิร์ฟเวอร์ที่ให้บริการ (หรือบางที่เรียกว่า host) และอุปกรณ์ในเครือข่ายอื่นๆ เช่น Router, เครื่องพิมพ์ เพื่อให้สามารถแชร์การใช้อุปกรณ์ร่วมกันได้ หรือสามารถส่งผ่านข้อมูลไปมาระหว่างกันได้อย่างถูกต้อง เมื่อมีการเชื่อมต่อกันแล้วก็จำเป็นต้องมีการกำหนดหรือระบุเลขหมายของอุปกรณ์ทุกชิ้นทุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดในเครือข่าย เพื่อให้อ้างอิงได้โดยไม่ซ้ำกัน เพราะถ้าซ้ำกันแล้วการรับส่งข้อมูลอาจจะไม่ถึงมือของผู้รับปลายทางได้อย่างถูกต้อง เลขหมายดังกล่าวจะเรียกได้ว่า แอดเดรส (address) หรือเลขหมายประจำตัวที่มีข้อกำหนดเป็นมาตรฐาน ซึ่งในการใช้งานโปรโตคอล TCP/IP ที่เชื่อมโยงเครือข่ายอินเตอร์เน็ตนี้ เลขหมายที่ใช้อ้างถึงกันจะใช้เป็นเลขที่เรียกว่า IP address (Internet Protocol address) หรือ "หมายเลข IP"

ไอพีแอดเดรส (IP address)

หมายเลข IP address ถูกกำหนดขึ้นมาให้เป็นหมายเลขอ้างอิงประจำตัวของอุปกรณ์ต่างๆที่เชื่อมต่ออยู่ในเครือข่ายอินเตอร์เน็ต โดยการกำหนดหมายเลข IP address ให้แต่ละเครื่องหรือแต่ละอุปกรณ์นี้จะต้องไม่ซ้ำกัน ซึ่งหมายเลข IP address นี้จะไม่ถูกผูกติดกับตัวฮาร์ดแวร์แต่อย่างใด จึงสามารถกำหนดใหม่หรือแก้ไขเปลี่ยนแปลงได้เมื่อมีการเปลี่ยนตัวฮาร์ดแวร์ ทั้งนี้เนื่องจากเป็นการกำหนดด้วยซอฟต์แวร์ แตกต่างกับหมายเลข MAC address (Media Access Control address) ซึ่งเป็นหมายเลขประจำตัวของอุปกรณ์ที่ต่ออยู่ในเครือข่าย ค่า MAC address จะถูกกำหนดจากบริษัทผู้ผลิตอุปกรณ์ตั้งแต่เริ่มผลิต เช่น อุปกรณ์ Network Interface Card (NIC) จะมีค่า MAC address ประจำตัวที่ไม่ซ้ำกันและไม่สามารถแก้ไขได้ ค่า MAC address เป็นการระบุค่าอ้างอิงของอุปกรณ์ฮาร์ดแวร์ในระดับล่างสุด (Physical Layer) ของกลไกการรับส่งข้อมูลภายในเครือข่าย ถ้าจะใช้หมายเลข MAC address สำหรับระบุอ้างอิงกันในเครือข่ายแล้วจะเกิดปัญหามาก เมื่อมีการเปลี่ยนหรือย้ายเครื่องต้องทำการกำหนดระบบเครือข่ายใหม่ (configuration) นอกจากนี้ยังจดจำได้ยากกว่า ตัวอย่างของหมายเลข MAC address คือ 08:0a:0e:12:b5:05 การที่หมายเลข IP address ถูกใช้อ้างอิงในการติดต่อกันด้วยโปรโตคอล TCP/IP เพราะการใช้หมายเลข IP address จะยืดหยุ่นและคล่องตัวกว่า

การทำงานของโปรโตคอล IP จำเป็นต้องอาศัยหมายเลข IP address นี้เพื่อระบุและอ้างถึงอุปกรณ์ต่างๆที่ต่ออยู่ในเครือข่ายไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์, เมล์เซิร์ฟเวอร์, อุปกรณ์ Router ฯลฯ หมายเลข IP address จะเป็นค่าตัวเลขขนาด 32 บิต ถูกแบ่งออกเป็นส่วนละ 8 บิต รวมเป็น 4 ส่วนและคั่นแต่ละส่วนด้วยเครื่องหมายจุด (.) ดังนั้นค่าตัวเลขในแต่ละส่วนจะมีได้ตั้งแต่ 0 ถึง 255 (2^8) ตัวอย่างของ IP address เช่น 205.144.78.1 หรือ 10.0.0.1 เป็นต้น นอกจากนี้ IP address บางหมายเลขหรือบางช่วงจะมีการใช้งานในลักษณะความหมายและหน้าที่พิเศษออกไปในการทำงานของโปรโตคอล TCP/IP เช่นหมายเลข IP ที่ 127.0.0.1 เป็นหมายเลข IP ที่ใช้ทำหน้าที่เป็น loop back address คือใช้กำหนดค่า loop back หรือแอดเดรสย้อนกลับให้กับอุปกรณ์นั้น

ลูปแบกแอดเดรส (Loop back address)

Loop back address เป็นแอดเดรสที่กำหนดขึ้นเพื่อใช้ในงานที่ต้องการให้ process หนึ่งติดต่อกับ process อื่นๆในเครื่องเดียวกันผ่าน IP address หรืออาจจะกล่าวโดยง่ายว่า Process ในเครื่องจะติดต่อกันระหว่าง process เองได้โดยผ่ายกลไก Loop back interface ซึ่งมีการกำหนดหมายเลข IP address พิเศษให้เป็น Loop back address ดังนี้

127.0.0.0	Loop back สำหรับเครือข่าย class A
191.255.0.0	Loop back สำหรับเครือข่าย class B
223.255.255.0	Loop back สำหรับเครือข่าย class C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นถ้าเราพบค่า IP address เป็น 127.0.0.2 หรือ 191.255.0.1 หรือ 223.255.255.111 ให้คิดว่าเป็นการกำหนดค่า loop back ในเครือข่าย class A,B,C ตามลำดับ

ค่า IP address ที่กำหนดเป็น loop back นี้จะถูกผนึกหรือ bind (กำหนดการอ้างอิงกับฮาร์ดแวร์) เข้ากับ loop back network interface ของอุปกรณ์นั้นๆ เช่น ค่า IP address ที่ 127.0.0.1 สำหรับในระบบ Windows NT จะถือว่าเป็นค่าที่ใช้เป็น loop back ปกติที่ระบบรู้จักและกำหนดไว้ให้ใช้งาน ดังนั้นเมื่อมีการส่งผ่านข้อมูลไปที่หมายเลข IP 127.0.0.1 จะไม่มีการส่งข้อมูลออกไปที่เครือข่าย แต่จะย้อนกลับมาที่ IP ต้นทางโดยกลไกการ loop back ของ loop back driver นั่นเอง

ค่าของ IP address จะถูกกำหนดออกเป็น 2 ความหมายคือ ค่าของหมายเลขอุปกรณ์ในเครือข่าย (host address) และค่าของหมายเลขเครือข่าย (network address) ตัวอย่างเช่นมีเครื่องเว็บเซิร์ฟเวอร์เชื่อมต่ออยู่ในเครือข่าย 2 เครื่อง โดยแต่ละเครื่องมี IP address ประจำตัวคือ 205.144.78.2 และ 205.144.78.3 ตามลำดับ เครื่องทั้งสองมีค่าของเลขหมายเครือข่ายเหมือนกันคือ 205.144.78 แสดงว่าเครื่องทั้งสองต่อเชื่อมอยู่ในเครือข่ายเดียวกัน บนสายสัญญาณที่เชื่อมโยงเส้นเดียวกันแต่มีหมายเลขแระจำตัวเครื่องที่แตกต่างกันคือ 2 และ 3 ตามลำดับ

เพื่อไม่ให้การกำหนดแจกจ่ายค่า IP address ซ้ำซ้อนกัน จึงมีหน่วยงานกลางทำหน้าที่กำหนด IP address และแจกจ่ายเลขหมายให้แต่ละองค์กรได้ใช้งาน คือ หน่วยงาน InterNIC หรือ Internet Network Information Center เป็นผู้ดูแลฐานข้อมูลการแจกจ่าย IP address และการใช้งานให้เกิดประโยชน์สูงสุด บริษัทหรือองค์กรที่ต้องการ IP address ในการกำหนดใช้งานเพื่อเชื่อมระบบเครือข่ายของตนเข้าสู่เครือข่ายอินเตอร์เน็ต จะต้องสมัครเป็นสมาชิกขององค์กร InterNIC เสียก่อนโดยจะมีการเก็บค่าบำรุงเป็นรายปี และแบ่งระดับของสมาชิกเป็น องค์กรขนาดใหญ่, องค์กรขนาดกลางและองค์กรขนาดเล็ก ตามลำดับ ค่าบำรุงสมาชิกก็จะจ่ายไม่เท่ากันเมื่อบริษัทหรือองค์กรได้เป็นสมาชิกแล้วก็สามารถแจ้งความจำนงขอ IP address มาใช้งานต่อไป สำหรับประเทศไทย บริษัทใดที่ต้องการเชื่อมต่อเครือข่ายของตนเข้าสู่อินเตอร์เน็ต จะต้องติดต่อกับบริษัทผู้ให้บริการอินเตอร์เน็ต (Internet Service Provider หรือ ISP) แทน และสามารถขอ IP address ที่ ISP ได้ทันที เพราะทาง ISP ได้ขอ IP address เอาไว้สำหรับแจกจ่ายให้กับลูกค้าของตนแล้ว

<u>ไอพีเวอร์ชั่น 4 (Ipv4) และ ไอพีเวอร์ชั่น 6 (Ipv6)</u>

หมายเลข IP address ทุกค่าจะมีรูปแบบที่เขียนให้เข้าใจเหมือนกันคือ เป็นตัวเลข 4 ชุดคั่นด้วยจุด เพื่อให้อ่านและจดจำได้ง่าย IP address มีขนาด 32 บิต เช่น 204.283.255.20 เป็นต้น ในปัจจุบันเรียกรูปแบบดังกล่าวนี้ว่า IP version 4 หรือ IPv4 ซึ่งมีการใช้งานกันทั่วไปและถูกใช้หมดไปเรื่อย ๆ จนอาจจะไม่เพียงพอสำหรับใช้งานในอนาคตอันใกล้นี้ เนื่องจาก IPv4 มีขนาดเพียง 32 บิต ทำให้การกำหนดค่า IP address ทำได้ไม่มากพอกับอุปกรณ์และเครือข่ายใหม่ๆ ที่เชื่อมต่อเข้ามาในเครือข่ายอินเตอร์เน็ตจะมีบทบาทมากขึ้นจนทุกครัวเรือนจะมีส่วนเกี่ยวข้องด้วย ดังนั้น IP address ที่มีอยู่จะไม่เพียงพอให้ใช้งานกันทุกครัวเรือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาหลักของ IPv4

IPv4 ที่ใช้งานกันอยู่มีปัญหาที่ต้องปรับปรุงแก้ไขใหม่หลายประการ ทำให้ต้องมีการพัฒนา IP รุ่นใหม่ที่มีคุณสมบัติดีกว่าเดิม แต่ปัญหาหลักที่นับว่าเป็นเรื่องสำคัญคือ

1. หมายเลข IP address ที่มีให้ใช้งานใกล้จะหมด เนื่องจาก IPv4 มีขนาด 32 บิต ทำให้สามารถกำหนดค่า IP address ได้ไม่เพียงพอสำหรับอนาคตทำให้ไม่สามารถขยายเครือข่ายอินเตอร์เน็ตได้อีกต่อไป
2. ข้อมูลการระบุเส้นทางการส่งผ่านข้อมูลหรือ routing table ที่ถูกกำหนดไว้ใน Router หลักที่เชื่อมโยงเครือข่ายหลักของอินเตอร์เน็ตมีขนาดใหญ่ขึ้นมาก และเพิ่มขึ้นเรื่อย ๆ ทำให้มีปัญหาในการตรวจสอบเส้นทางในการส่งข้อมูลและทำให้ Router หลักทำงานได้ช้าลง

เพื่อแก้ปัญหาดังกล่าว หน่วยงาน IETF (Internet Engineering Task Force) ได้เริ่มหาทางแก้ไขในช่วง ค.ศ. 1990 เป็นต้นมา โดยมีการออกแบบโปรโตคอล IP และหมายเลข IP address ขึ้นใหม่ โดยเรียกว่า IP Next Generation (IPng) และต่อมาได้เปลี่ยนชื่อใหม่อย่างเป็นทางการว่า IPv6 (Internet Protocol Version 6) ซึ่งได้เพิ่มคุณสมบัติและความสามารถใหม่ ๆ เข้าไป และแก้ไขปัญหาจำนวน IP address ที่ขาดแคลนในอนาคตด้วย โดย IPv6 มีขนาด 128 บิต ซึ่งช่วยขยายจำนวน IP address ได้อย่างพอเพียง ทำให้สามารถกำหนดค่า IP ได้มากกว่าเดิมถึง 2^{96} เท่า ตัวโปรโตคอล IP ได้มีการปรับปรุงส่วน header ให้สนับสนุนการประมวลผลจาก Router ได้เร็วขึ้น มีการเพิ่มกลไกการตรวจสอบการใช้งาน (authentication) และการตรวจสอบความถูกต้องของข้อมูลดีกว่าเดิม ช่วยเพิ่มความปลอดภัยของข้อมูลได้มากขึ้น รองรับการใช้งานแอพพลิเคชั่นที่ต้องการให้มีการส่งผ่านข้อมูลแบบพิเศษได้ เช่น การใช้งานแอพพลิเคชั่นแบบ real time การส่งข้อมูลภาพและเสียงผ่านเครือข่ายอินเตอร์เน็ต เป็นต้น และสุดท้าย IPv6 ยังได้รับการออกแบบให้สามารถใช้งานกับอุปกรณ์เดิมที่ใช้กับ IPv4 ได้ ปัจจุบันการจัดสรรค่า IPv6 ยังไม่ได้เริ่มต้น แต่คาดว่าจะมีการใช้งานในเร็ววันนี้ สำหรับ IPv5 ไม่ได้นำออกมาใช้งาน เพราะใช้เป็นชุดพัฒนาและทดลองเท่านั้น และข้ามไปเป็น IPv6 แทน

ในการติดต่อโดยโปรโตคอล IPv6 สามารถกำหนดลักษณะการทำงานได้ 3 แบบ คือ unicast, multicast และ anycast ซึ่งมีรายละเอียดดังนี้

ไอพีเวอร์ชั่น 6 ยูนิแคส (IPv6 Unicast)

หมายเลข IPv6 แบบ unicast คือ การกำหนดเลขหมาย IP address ให้กับเครื่องใช้เซิร์ฟเวอร์ที่ให้บริการ หรือกำหนดให้อุปกรณ์ที่เชื่อมต่ออยู่ในเครือข่ายอินเตอร์เน็ตเหมือนปกติ แต่แบ่งเป็น 3 ลักษณะคือ

1. Provider – based address ใช้กำหนดให้กับผู้ให้บริการอินเตอร์เน็ต หรือบริษัท ISP ซึ่งข้อมูลของ IP address จะมีส่วนของ Provider ID และ Subscribe ID อยู่ด้วยเพื่อบอกให้ทราบว่าเป็นของ ISP ใด

2. Site – local – use address เป็นการกำหนดหมายเลข IPv6 ให้ใช้งานกับองค์กรที่ต้องการใช้โปรโตคอล TCP/IP แต่ไม่ได้เชื่อมต่อเข้ากับเครือข่ายอินเตอร์เน็ต เรียกว่าเป็นการใช้งานแบบ

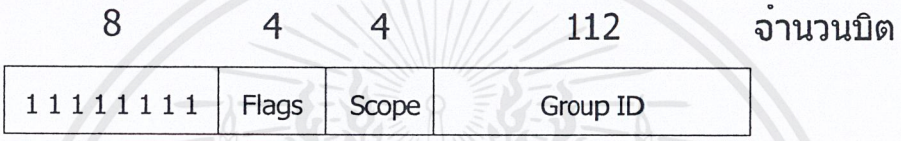
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินทราเน็ตในเบื้องต้นและต้องการเชื่อมต่อเข้าอินเตอร์เน็ตได้ในอนาคต ซึ่งจะจัดการแก้ไขข้อมูลระบบผ่านผู้ดูแลระบบได้ทันที ไม่จำเป็นต้องกำหนดค่าหมายเลข IP ให้ใหม่ทีละเครื่อง

3. Link – local – use address ใช้กำหนดให้กับเครือข่ายที่เชื่อมต่อยู่ในอินเตอร์เน็จตลอดเวลาขององค์กรหรือบริษัทใด ๆ

ไอพีเวอร์ชั่น 6 มัลติแคส (IPv6 Multicast)

การส่งข้อมูลแบบ multicast ก็คือการส่งข้อมูลจากที่ ๆ หนึ่งให้กับกลุ่มของผู้รับใด ๆ กลุ่มหนึ่งที่กำหนดไว้ตามเงื่อนไข เรียกว่าการส่งข้อมูลแบบ one – to – many ซึ่งจะแตกต่างกับการส่งข้อมูลแบบ broadcast ที่ส่งข้อมูลแบบ one – to – all คือส่งให้กับทุกคนที่อยู่ในเครือข่าย ซึงการที่ขนาดของ IPv6 มีมากกว่า IPv4 จึงทำให้หมายเลข IP ที่ทำหน้าที่ multicast มีจำนวนมากกว่าด้วยเช่นกัน โดยโครงสร้างของหมาย IPv6 แบบ multicast จะเป็นดังรูปที่ 2.4



รูปที่ 2.3 โครงสร้างของหมายเลข IPv6 แบบ multicast

ตามรูปโครงสร้างหมายเลข IPv6 address แบบ multicast มีการกำหนดให้ไบต์ (8 บิต) แรกเป็น 1 ทั้งหมด 4 บิตต่อมาเป็นข้อมูล Flag และอีก 4 บิตต่อมาเป็นข้อมูลระบุชนิดของการ multicast ตามลักษณะของกลุ่มผู้ใช้งาน โดยค่า scope และคำอธิบายดูได้จากตาราง

ค่าฐาน 16	คำอธิบาย
0	สำรองไม่มีการใช้งาน
1	Node-Local-Scope
2	Link-Local-Scope
3,4	ไม่มีการกำหนด
5	Site-Local-Scope
6,7	ไม่มีการกำหนด
8	Organization-Local-Scope
9-E	ไม่มีการกำหนด
F	สำรองไม่มีการใช้งาน

ตารางที่ 2.1 อธิบายค่า Scope

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอพีเวอร์ชั่น 6 เอนี่แคส (IPv6 Anycast)

การกำหนดเลขหมาย IP address เป็นประเภท anycast นี้เป็นเรื่องที่เกิดขึ้นใหม่ ซึ่งหมายเลข IP ที่ทำหน้าที่ anycast ใน IPv6 จะคล้ายกับ multicast ที่สามารถกำหนดค่าเลขหมายให้มีจุดเชื่อมต่อได้มากว่าหนึ่งหมายเลข การส่งข้อมูลไปยังหมายเลข IP ที่เป็น anycast จะเหมือนกับการส่งข้อมูลให้กับทุกคนในกลุ่มของ Interface เดียวกันซึ่งหมายเลข IP ที่เป็น anycast นี้จะถูกกำหนดมาจากเลขหมาย multicast address อีกทีหนึ่ง

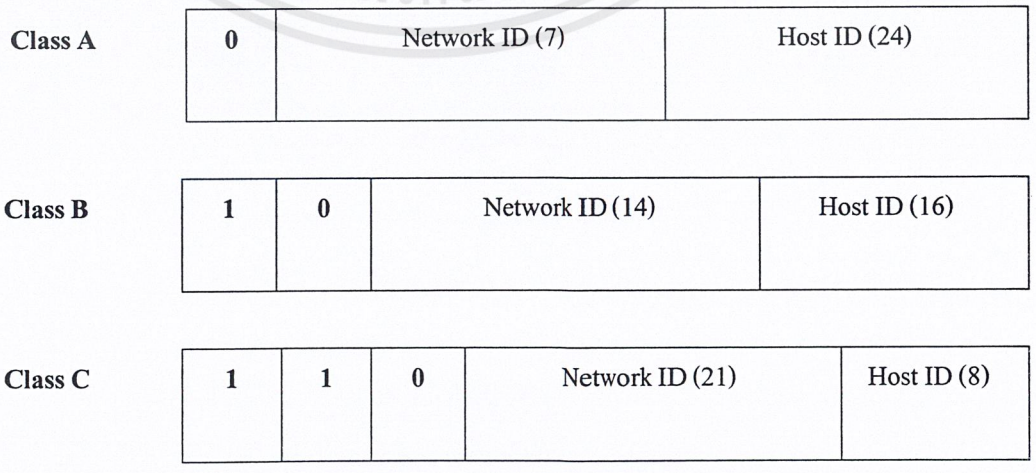
2.3.3 การจัดลำดับชั้นของเครือข่าย (Network Class)

การกำหนด IP address ไม่สามารถกำหนดขึ้นได้ตามใจชอบ แต่มีระเบียบวิธีแบ่งและการกำหนดที่ชัดเจนเป็นมาตรฐาน ในหมายเลข IP address จะถูกแบ่งเป็น 4 ส่วนโดยคั่นด้วยเครื่องหมายจุด ซึ่งสามารถแยกเป็น 2 ส่วนย่อยคือ ส่วนแรกเป็นหมายเลขของเครือข่าย (network address) และส่วนที่สองเป็นหมายเลขของเครื่องลูกข่าย (host address) ทั้งนี้การแบ่งส่วนจะเป็นไปตามการแบ่งระดับชั้นของเครือข่าย เรียกว่า network class ซึ่งการกำหนดให้มี network class นี้ก็เพื่อให้สามารถแจกจ่าย IP address ให้กับเครือข่ายต่างๆ ได้อย่างเหมาะสม เพราะในแต่ละเครือข่ายมักจะแตกต่างกัน บางเครือข่ายก็มีจำนวนเครื่องลูกข่ายมาก บางเครือข่ายมีน้อยแต่มีเครือข่ายย่อยๆ ในเครือข่ายหลักจำนวนมาก ฉะนั้นถ้าไม่มีการจัดลำดับของเครือข่ายให้ดี หมายเลข IP adress ก็จะถูกใช้งานอย่างสิ้นเปลืองและใช้งานไม่ได้เต็มจำนวนที่มี ลำดับของเครือข่ายแบ่งได้เป็น 5 ลำดับคือ class A, B, C, D และ E

ในแต่ละ network class หมายเลข IP ทั้ง 32 บิตจะถูกกำหนดเป็นหมายเลขของเครือข่ายและหมายเลขของเครื่องลูกข่าย โดยมีเงื่อนไขดังนี้คือ

1. **Class A** IP address บิตแรกของไบต์แรกสุดจะเป็น 0 เสมอ
2. **Class B** IP address 2 บิตแรกของไบต์แรกสุดจะเป็น 1 และ 0 เสมอ
3. **Class C** IP address 3 บิตแรกของไบต์แรกสุดจะเป็น 1, 1 และ 0 เสมอ

การแบ่งหมายเลขเครือข่ายและหมายเลขเครื่องลูกข่ายจะเป็นดังรูปที่ 2.5



รูปที่ 2.4 แสดงคลาสของไอพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นแต่ละ network class จะมีเครื่องลูกข่าย (host address) ที่เป็นสมาชิกในเครือข่ายของตนได้ตามตาราง

class	จำนวนเครื่องลูกข่ายที่มีได้	IP address เริ่มต้น
A	$2^{24} = 16,777,214$	0-127
B	$2^{16} = 65,536$	128-191
C	$2^{8} = 256$	192-223
D	-	224-239
E	-	240-255

ตารางที่ 2.2 บอกจำนวนเครื่องลูกข่ายที่มีและ IP address เริ่มต้น

- **Class A** เป็นหมายเลข IP address ที่เริ่มตั้งแต่ 0-127 จะกำหนดให้กับเครือข่ายที่มีขนาดใหญ่ เพราะ 1 เครือข่ายสามารถมีเครื่องลูกข่ายได้กว่า 16 ล้านเครื่อง หมายเลข IP adress จะเป็นลักษณะ net.host.host.host ตัวอย่างเช่น IP address ของ class A เป็น 121.7.23.3 หมายถึงเครือข่ายเลขที่ 121 หมายเลขเครื่องคือ 7.23.3
- **Class B** เป็น class A ที่เริ่มตั้งแต่ 128-191 จะกำหนดให้กับเครือข่ายที่มีขนาดใหญ่เช่น กันแต่เล็กกว่า class A ในแต่ละเครือข่ายของ class B สามารถมีเครื่องลูกข่ายได้ 2^{16} - จำนวนแอดเดรสที่ใช้ควบคุมระบบเครือข่าย = 64,516 เครื่อง หมายเลข IP address จะเป็นลักษณะ net.net.host.hots ตัวอย่างเช่น ค่า IP address ของ Class B เป็น 137.103.210.2 หมายถึงเครือข่ายเลขที่ 137.103 หมายเลขเครื่องคือ 210.2
- **Class C** เป็น IP address ที่เริ่มตั้งแต่ 192-223 จะกำหนดให้กับเครือข่ายที่เป็นองค์กรทั่วไป ซึ่งส่วนใหญ่จะเป็นองค์กรขนดกลางถึงเล็ก ในแต่ละเครือข่ายมีเครื่องลูกข่ายไม่เกิน 2^{8} - จำนวนแอดเดรสที่ใช้ควบคุมระบบเครือข่าย = 254 เครื่อง หมายเลข IP address จะเป็นลักษณะ net.net.net.host ตัวอย่างเช่น ค่า IP address ของ class C เป็น 202.182.255.3 หมายถึง เครือข่ายเลขที่ 202.192.255 หมายเลขเครื่องคือ 3
- **Class D** เป็นการกำหนดหมายเลข IP address สำรองไว้สำหรับส่งข้อมูลแบบ multicast ซึ่งจะไม่มีการแจกจ่ายให้ใช้งานทั่วไป
- **Class E** เป็นหมายเลข IP address พิเศษที่ใช้สำหรับงานทดสอบและพัฒนา ไม่มีการกำหนดให้ใช้งานทั่วไป

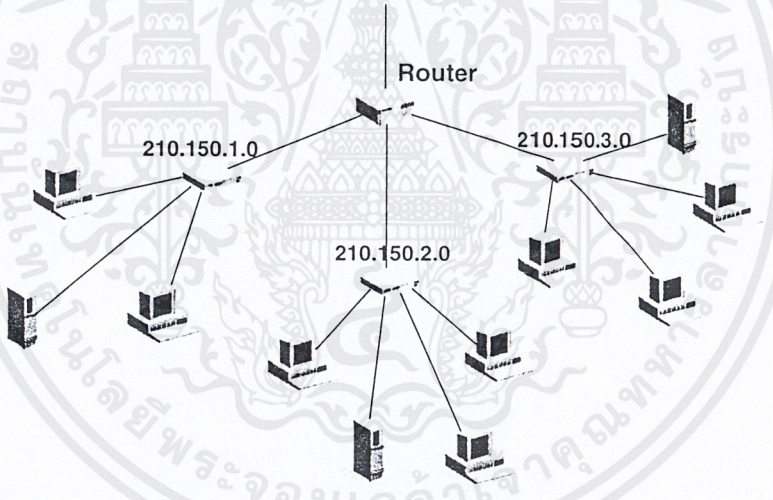
เราอาจจำหมายเลข IP address ได้ง่ายๆ คือในเครือข่าย class A จะมีหมายเลข IP ตัวแรกอยู่ระหว่าง 0-127 ดังนั้นค่าที่ตามมาจะเป็นหมายเลขของเครื่องส่วนเครือข่าย class B หมายเลข IP หน้าสุดจะเป็น 128 – 191 ดังนั้น ค่า IP สองตัวแรกจะเป็นหมายเลขเครือข่ายและค่า IP สองตัวหลังจะเป็นหมายเลขเครื่องสุดท้าย เครือข่ายใน class C จะมีหมายแรก IP หน้าสุดเป็น 192 – 223 ดังนั้นค่า IP ตัวหลังสุดจะเป็นหมายเลขเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซับเน็ต (Subnet)

ในปัจจุบันการกำหนดหมายเลข IP ของเครือข่ายประเภท class A และ class B นั้น ไม่มีการกำหนดให้แล้ว เนื่องจากแทบไม่มีเครือข่ายใดที่มีความจำเป็นต้องใช้ address มากขนาดนั้น คงเหลือแต่ class C เท่านั้นที่กำหนดให้แต่ละบริษัทหรือหน่วยงานต่าง ๆ และถึงแม้ว่าจะเป็นเครือข่ายที่ใช้หมายเลข IP ประเภท class C ก็ตามส่วนมากก็ไม่มีใครเชื่อมต่อเครื่องลูกข่ายมากถึง 254 เครื่อง ในหนึ่งเครือข่ายหากทุกๆ เครือข่ายมีอุปกรณ์หรือเครื่องติดตั้งใช้งานในเครือข่ายนั้นเพียงไม่กี่เครื่อง แต่จำเป็นต้องกำหนดเลขหมาย IP ให้ใช้งานในเครือข่ายนั้น 254 เครื่อง จะเกิดปัญหามีหมายเลข IP ที่ไม่ได้ถูกใช้งานเป็นจำนวนมาก และไม่สามารถนำหมายเลข IP ที่ไม่ได้ใช้งานนี้ไปให้ผู้อื่นใช้ได้ ดังนั้นการทำ Subnet หรือ Sub Network จึงเกิดขึ้น เพื่อแบ่งเครือข่ายออกเป็นเครือข่ายย่อยๆ และทำให้การกำหนดใช้งานหมายเลข IP ที่ได้รับมาสามารถแบ่งออกเป็นส่วนๆ เหมาะสมกับจำนวนอุปกรณ์ในแต่ละเครือข่ายได้ ทั้งยังแบ่งหมายเลข IP address ส่วนที่ไม่ได้ใช้ให้หน่วยงานอื่นหรือเครือข่ายอื่นได้อีกด้วย

ตัวอย่างเช่น หากหน่วยงานของเรามีเครือข่ายอยู่ 3 เครือข่าย แต่ละเครือข่ายก็มีหมายเลข IP address class C เป็นของตัวเองคือ 210.150.1.0 และ 210.120.2.0 และ 210.150.3.0 เครือข่ายทั้งหมดก็จะมีรูปร่างหน้าตาดังแสดงในรูป ที่ 2.5



รูปที่ 2.5 แสดงเครือข่าย 3 เครือข่ายที่แต่ละเครือข่ายมี IP address ของตัวเอง

แต่ละเครือข่าย เช่น 210.150.1.0 จะสามารถเชื่อมต่อเครื่องลูกข่ายได้สูงสุดถึง 254 เครื่อง เนื่องจากใช้ 8 บิตสุดท้ายเป็นตัวกำหนด Host address ซึ่งจำนวนข้อมูล 8 บิตนี้จะอ้างอิงตัวเลขแตกต่างกันได้ 256 address หักออก 2 address คือ 0 กับ 255 ซึ่งจองไว้สำหรับควบคุมการทำงานเครือข่าย ถ้าหากว่าแต่ละเครือข่ายเชื่อมต่อกับคอมพิวเตอร์ไม่กี่เครื่องหมายเลข IP address ที่เหลือก็จะเสียเปล่าๆ และจะแบ่งให้ผู้อื่นนำไปใช้ก็ไม่ได้เนื่องจาก อุปกรณ์ Router เมื่อพบ IP address 210.150.1.x จะไม่สนใจข้อมูลตัวสุดท้ายทางขวามือสุด แต่จะส่งข้อมูลให้ระบบเครือข่าย โดยดูที่ IP address 24 บิต ซ้ายมือคือดูที่หมายเลขประจำเครือข่าย (Network address) ในที่นี้คือ 210.150.1 เท่านั้น เนื่องจาก class C แบ่งการอ้างอิง IP address 24 บิตซ้ายมือสำหรับเป็น Network address และ 8 บิตขวามือเป็นหมายเลขเครื่อง (Host address) การทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

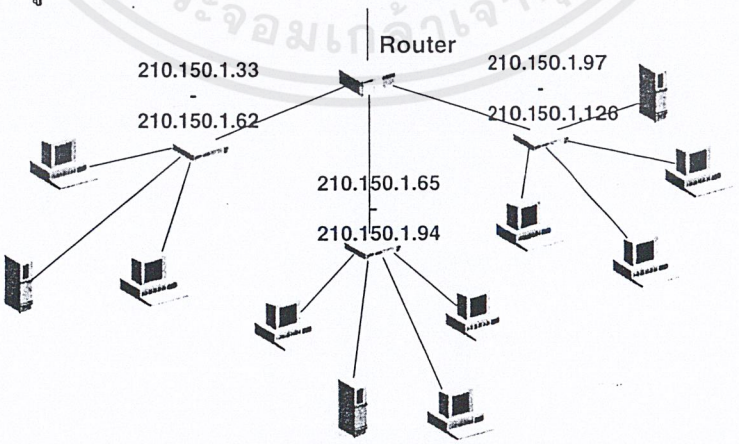
Subnet จะช่วยให้การกำหนดหมายเลข IP address ถูกแบ่งออกเป็นเครือข่ายย่อย ๆ ที่ใช้ IP address ได้อย่างมีประสิทธิภาพ

จากตัวอย่างข้างต้นหากเรานำ Host address จำนวน 8 บิตมาแบ่งออกเป็น 2 ส่วน เพื่อแบ่งเครือข่ายออกเป็นเครือข่ายย่อย ๆ เช่นใช้ข้อมูล 3 บิตทางซ้ายมือเป็นตัวกำหนดเครือข่ายย่อย และใช้ 5 บิตทางขวามือเป็นตัวกำหนด Host address ดังนั้นในหมายเลข IP address 1 ชุด เราสามารถแบ่งเครือข่ายย่อยได้ 8 เครือข่ายโดยในแต่ละเครือข่ายจะมีเครื่องลูกข่ายได้ถึง 32 เครื่อง เราเรียกการแบ่งเครือข่ายย่อยนี้ว่า Subnet ดังแสดงในตารางต่อไปนี้

IP address ของเครือข่าย 24 บิต	Host address เดิมขนาด 8 บิต	
(Network address)	Subnet address 3 บิต	Host address ใหม่ 5 บิต
210.150.1	000	00000 ถึง 11111
210.150.1	001	00000 ถึง 11111
210.150.1	010	00000 ถึง 11111
210.150.1	011	00000 ถึง 11111
210.150.1	100	00000 ถึง 11111
210.150.1	101	00000 ถึง 11111
210.150.1	110	00000 ถึง 11111
210.150.1	111	00000 ถึง 11111

ตารางที่ 2.3 แสดงการแบ่ง Host address ออกเป็น 2 ส่วนคือ Subnet และ Host address

แต่ในทางปฏิบัติ Subnet address 000 และ 111 กับ Host address 00000 และ 11111 จะถูกจองไว้สำหรับใช้ในการควบคุม ดังนั้น หากเราแบ่ง 3 บิตซ้ายมือเป็น Subnet และ 5 บิตขวามือเป็น Host address เราจะได้เครือข่ายย่อยทั้งหมด 6 เครือข่ายย่อย และแต่ละเครือข่ายจะมีเครื่องลูกข่ายได้สูงสุด 30 เครื่อง ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 แสดงการแบ่ง Subnet ออกเป็นเครือข่ายย่อย โดยใช้ IP address ตัวเดียว คือ 210.150.1.0 (ในรูปแสดงเพียง 3 เครือข่ายจากทั้งหมด 6 เครือข่ายย่อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42221

ซึ่งจากตัวอย่างเราจะเห็นว่าเราใช้หมายเลข IP address เพียงค่าเดียวคือ 210.151.1.0 มาแบ่งออกเป็นเครือข่ายย่อยๆ หรือ Subnet โดยในตัวอย่างนี้เราแบ่งออกเป็น 6 Subnet สูงสุด แต่ละ Subnet จะเชื่อมต่อเครื่องลูกข่ายได้สูงสุด 30 เครื่อง ทำให้เราสามารถกำหนดใช้งานหมายเลข IP address ได้อย่างมีประสิทธิภาพ และไม่เสีย IP address โดยไม่จำเป็น แต่ในการแบ่งเครือข่ายย่อยนี้อุปกรณ์ Router จะต้องทราบข้อมูลว่าการแบ่ง Subnet ของเรานั้น ใช้ข้อมูลกี่บิตเป็น Subnet และใช้ข้อมูลกี่บิตเป็น Host address จึงจะส่งข้อมูลให้เครือข่ายได้อย่างถูกต้อง เราเรียกข้อมูลการแบ่ง Subnet นี้ว่า "Subnet Mask" ซึ่งในจะกำหนดให้ค่าของ Network address และ Subnet address มีค่าเป็น "1" ส่วน Subnet Mask ของ Host address จะมีค่าเป็น "0" ดังแสดงในตารางต่อไปนี้

	Network address	Subnet	Host address
IP address	11010010.10010110.00000001	xxx	xxxxx
Subnet Mask	11111111.11111111.11111111	111	00000
Subnet Mask เลขฐานสิบ	255.255.255	224	

ตารางที่ 2.4 แสดงการกำหมด Subnet Mask ของ Subnet ขนาด 3 บิต

เมื่อเรากำหนดค่าของ Subnet Mask ได้แล้ว และใส่ค่านี้ลงในอุปกรณ์ Router ตัว Router ก็จะรับส่งข้อมูลในเครือข่ายได้อย่างถูกต้อง เนื่องจากอุปกรณ์ Router จะตรวจสอบหมายเลข IP address ว่าเป็นของเครือข่ายใดโดยใช้ข้อมูลถึง 27 บิต แทนที่จะเป็น 24 บิตตามเดิมโดยใช้ Subnet Mask นี้เข้าช่วย

Class C Subnet

เนื่องจากหมายเลข IP address ที่เราใช้งานกันอยู่ทุกวันนี้จะเป็นเครือข่าย Class C ก็คือการกำหนด Host address หรือเครื่องลูกข่าย ดังนั้นการ Subnet ใน Class C ก็คือการแบ่งข้อมูล 8 บิตของ Host address นี้ออกเป็น Subnet address และ Host address ใหม่นั่นเอง ซึ่งใน Class C นี้เราจะแบ่ง Subnet ได้ทั้งหมด 5 แบบดังตารางดังต่อไปนี้

จำนวนเครือข่ายย่อย (Subnet)	จำนวนเครื่องลูกข่าย (Host Address)	Subnet Mask เลขฐานสิบ	จำนวนบิตที่ใช้เป็น Subnet
2	62	255.255.255.192	2
6	30	255.255.255.224	3
14	14	255.255.255.240	4
30	6	255.255.255.248	5
62	2	255.255.255.252	6

ตารางที่ 2.5 แสดง Subnet ทั้งของ class C

ซึ่งหากไม่มีการทำ Subnet ค่า Subnet Mask จะเป็น 255.255.255.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Subnet 0 คือทุกบิตของ Subnet มีค่าเป็น “0” ทั้งหมด และ Subnet 7 ที่ทุกบิตมีค่าเป็น “1” ทั้งหมดจะถูกจองไว้ใช้สำหรับกรควบคุมระบบเครือข่าย และ Subnet ที่มีจำนวนเครือข่ายย่อยเท่ากับหนึ่งจะใช้งานไม่ได้เนื่องจากไม่มีจำนวนบิตเหลือให้นำไปใช้ทำ Subnet ดังนั้นจาก Host address ขนาด 8 บิต จึงเหลือนำมาใช้ทำ Subnet ได้เพียง 5 แบบเท่านั้น เพราะถูกหักออกไป 3 Subnet จากเหตุผลดังกล่าว

ในแต่ละ Subnet ก็จะมีการจอง Host address ไว้ 2 ตำแหน่ง คือ Host address ที่มีค่าเป็น “0” หมดทุกบิต เอาไว้ใช้สำหรับเป็นหมายเลขประจำเครือข่าย (Network Address) และ Host address ที่มีค่าเป็น “1” หมดทุกบิตจะถูกจองไว้ใช้สำหรับเป็น Broadcast address ดังนั้นในทุก ๆ แบบของ Subnet จำนวนเครื่องลูกข่ายจึงถูกหักออก 2 address คือ address ที่เป็น “0” ทั้งหมดทุกบิตและที่เป็น “1” หมดทุกบิต เหลือจำนวนเครื่องลูกข่ายสูงสุดดังที่แสดงในตารางข้างต้น

เราจะเห็นได้ว่าถ้าเราแบ่งจำนวนเครือข่ายย่อยออกไม่กี่เครือข่าย แต่ละเครือข่ายย่อยก็จะมีจำนวนเครื่องลูกข่ายได้มาก ในทางกลับกันหากเราแบ่งจำนวนเครือข่ายย่อยออกไปมากๆ แต่ละเครือข่ายย่อยก็จะมีเครื่องลูกข่ายได้ไม่กี่เครื่องเท่านั้น ในทางปฏิบัติจำนวนเครือข่ายย่อย 2 เครือข่าย หรือ 64 เครือข่ายจึงไม่เหมาะสมในการวางระบบ เนื่องจากเครือข่ายย่อย 2 เครือข่ายนั้นน้อยเกินไปและขยายเครือข่ายย่อยอีกไม่ได้ ส่วนจำนวนเครือข่ายย่อย 62 เครือข่ายนั้นก็มากเกินไปและแต่ละเครือข่ายย่อยก็มีเครื่องลูกข่ายได้เพียง 2 เครื่องเท่านั้น ในการใช้งานจริงก็น่าจะเป็นการแบ่งเครือข่ายย่อยออกเป็น 6 เครือข่าย โดยแต่ละเครือข่ายย่อมมีลูกข่าย 30 เครื่อง หรือแบ่งออกเป็นเครือข่ายย่อย 14 เครือข่าย และแต่ละเครือข่ายย่อยมีลูกข่าย 14 เครื่องเท่านั้นสำหรับการทำ Subnet ของ Class C นี้

<u>การแบ่ง IP address แบบไม่ลง Class พอดี</u>

การแบ่ง IP address แบบไม่ลง Class พอดีหรือเรียกว่าแบบไม่มี Class (Classless Inter – Domain Router หรือ CIDR) คือการกำหนดหมายเลข IP address แบบไม่คำนึงถึงการแบ่งตาม network class เช่น กำหนดหมายเลข IP address ใน class C ให้ส่วนของหมายเลขหรือเครือข่ายเป็น 26 บิต (ปกติ 24 บิต) ทำให้จำนวนเครื่องลูกข่ายหรือ Host address ลดลงไป เหตุผลที่มีการกำหนด IP address เป็น CIDR นี้ก็เพราะต้องการให้มีการใช้หมายเลข IP address ได้มากที่สุด มักใช้กำหนดให้กับองค์กรหรือบริษัทที่มีเครือข่ายขนาดเล็ก มีเครื่องลูกข่ายไม่มากต่อเชื่อมเข้ากับเครือข่ายอินเตอร์เน็ต ถ้ากำหนดหมายเลข IP address ให้กับองค์กรนั้นทั้ง class C ก็จะมีหมายเลข IP address ที่ไม่ถูกใช้งานจำนวนมาก ในการกำหนดหมายเลข IP แบบ CIDR จะมีการเขียนในรูปแบบของเครื่องหมาย “/” คั่นเพื่อบอกให้ทราบว่าส่วน network address เป็นกี่บิต เช่น 202.185.255.32/26

<u>หมายเลขอ้างอิงเครือข่ายและ Broadcast address</u>

ในการกำหนดหมายเลข IP address ให้กับอุปกรณ์และเครื่องเซิร์ฟเวอร์ให้บริการ จะต้องมีการกำหนดค่าหมายเลข IP address พิเศษให้กับเครือข่ายนั้นด้วย ดังนี้

- กำหนดหมายเลขประจำเครือข่าย (network address)
- กำหนดหมายเลข IP สำหรับการส่งข้อมูลให้กับอุปกรณ์ทั้งหมดของเครือข่ายนั้น (broadcast address) ใช้เพื่อส่งข้อมูลให้กับอุปกรณ์ทั้งหมดของเครือข่ายนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขประจำเครือข่าย คือ หมายเลข IP address ที่ใช้อ้างอิงเครือข่ายนั้นโดยไม่เจาะจงถึงเครื่องลูกข่ายภายใน ซึ่งโดยทั่วไปการกำหนดหมายเลขของเครือข่ายทำได้โดยกำหนดให้ค่าหมายเลข host เป็น 0 เช่น เครือข่าย 172.10.0.0

หมายเลข IP address สำหรับส่งข้อมูล broadcast ของเครือข่ายนั้นๆ คือ หมายเลข IP address ที่กำหนดค่าหมายเลข host เป็น 255 ทั้งหมด จากตัวอย่าง หมายเลข IP address ที่ 172.10.255.255 จะเป็นหมายเลข network broadcast address ของเครือข่ายหมายเลข 172.10.0.0 และเครือข่ายนี้มี IP address ที่จะจัดสรรให้เป็นหมายเลข IP address ประจำเครื่องลูกข่ายได้ตั้งแต่หมายเลข IP ที่ 172.10.0.1 ถึง 172.10.255.254 เป็นต้น

นอกจากนี้ได้มีการกำหนดให้หมายเลข IP ที่ 255.255.255.255 เป็นหมายเลข IP สำหรับการ broadcast ทั่วไปไม่เฉพาะเจาะจงกับเครือข่ายใดเครือข่ายหนึ่งอีกด้วย

มัลติแคสแอดเดรส (Multicast Address)

จากการแบ่งเครือข่ายด้วยหมายเลข IP address เป็นชั้นต่างๆ ของเครือข่ายเป็น class A, B, C, D และ E นั้น เครือข่ายใน class D ถูกกำหนดให้เป็นหมายเลข IP address สำรองเพื่อใช้งานแบบ multicast กล่าวคือหมายเลข IP ของ class D ตั้งแต่ 224.0.0.0 ถึง 239.255.255.255 จะทำหน้าที่เป็น multicast address โดยแบ่งเป็นกลุ่มย่อย ๆ ซึ่งสมาชิกในกลุ่มจะรับข้อมูลที่ถูกส่งออกมาให้ในระหว่างเครือข่าย ตัวอย่างเช่นโปรโตคอล RIPv2 ใช้หมายเลข IP ที่ 224.0.0.9 ทำหน้าที่เป็น multicast address ของกลุ่มตนเอง คือกลุ่มอุปกรณ์ Gateway หรือ Router ที่ใช้โปรโตคอล RIPv2 ในการติดต่อระหว่างกัน กลุ่มอุปกรณ์อื่นหรืออุปกรณ์ที่ใช้การติดต่อกันด้วยโปรโตคอลอื่นก็จะไม่สนใจข้อมูลที่ถูกส่งเป็น broadcast ออกมา ดังนั้นในการส่งข้อมูลถึงกันในกลุ่มจะติดต่อด้วย IP address นี้เท่านั้น ตัวอย่างของค่า multicast address ที่กำหนดไว้จะดูได้ในตารางต่อไปนี้

โปรโตคอล	Multicast address	ความหมาย
RIPv2	224.0.0.9	กลุ่มอุปกรณ์ Router ที่ใช้โปรโตคอล RIPv2 รับข้อมูล broadcast
OSPF	224.0.0.5	กลุ่มอุปกรณ์ Router ที่ใช้โปรโตคอล OSPF รับข้อมูล broadcast
	224.0.0.6	ใช้เฉพาะบาง Router ที่ใช้โปรโตคอล OSPF รับข้อมูล broadcast
Multicast ทั่วไป	224.0.0.1	ทุกระบบในเครือข่ายนั้น รับข้อมูล broadcast

ตารางที่ 2.6 แสดงความหมายของโปรโตคอลแต่ละชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IP address ที่ออกสู่อินเตอร์เน็ตไม่ได้

เนื่องจากความนิยมในการใช้โปรโตคอล TCP/IP มีมากขึ้นทั้งการใช้งานในเครือข่ายส่วนตัว (private network/intranet) และในเครือข่ายอินเตอร์เน็ต หน่วยงาน InterNIC ได้มีการกำหนดให้หมายเลขประจำเครือข่ายบางอันเป็นหมายเลขพิเศษสำรองเอไว้ใช้งาน ในวัตถุประสงค์แบบเครือข่ายส่วนตัวที่ไม่สามารถใช้งานในอินเตอร์เน็ตได้ และกำหนดให้หมายเลขเครือข่ายนั้นเป็นแบบ non – routable คือไม่สามารถรับส่งข้อมูลออกสู่อินเตอร์เน็ตได้ ซึ่งหมายเลขเครือข่ายเฉพาะนี้จะใช้ในการกำหนดหมายเลข IP address ให้กับเครือข่ายขององค์กรหรือหน่วยงานที่ไม่ต้องการเชื่อมต่อเข้าสู่อินเตอร์เน็ต โดยหมายเลขเครือข่ายดังกล่าวมีดังนี้

Network Class	จำนวนเครือข่ายที่กำหนดได้	หมายเลขเครือข่าย
A	1	10.0.0.0
B	16	172.16.0.0 – 173.31.0.0
C	256	192.168.0.0 – 192.168.255.0

ตารางที่ 2.7 แสดงจำนวนเครือข่ายและหมายเลขเครือข่ายของ Network Class ของหน่วยงานที่ไม่ต้องการเชื่อมต่อ Internet

ดังนั้นในเครือข่าย class C ที่มีการกำหนดหมายเลขเครือข่ายเป็น 192.168.0.0 และมีเครื่องลูกข่ายเป็นหมายเลข IP ตั้งแต่ 192.168.0.1 ถึง 192.168.0.254 นั้น เราจะรู้ได้ทันทีว่าเครือข่ายนี้ใช้งานโดยไม่มีการเชื่อมต่อออกสู่อินเตอร์เน็ต ซึ่งการกำหนดหมายเลขเครือข่ายในกลุ่ม non–routable network นี้ก็เพื่อให้แน่ใจได้ว่าเครือข่ายขององค์กรนั้นจะไม่สามารถส่งผ่านข้อมูลใด ๆไปยังอินเตอร์เน็ตได้ และผู้ใช้อินเตอร์เน็ตจากภายนอกก็จะไม่สามารถติดต่อเรียกใช้ข้อมูลหรือเครื่องเซิร์ฟเวอร์ที่ให้บริการใด ๆ ในเครือข่ายนี้ได้เช่นกัน

2.4 โปรโตคอลทีซีพี/ไอพี (TCP/IP Protocol)

ในหัวข้อนี้จะกล่าวถึงลักษณะการทำงานของอินเตอร์เน็ต ซึ่งประกอบไปด้วยเครื่องเกี่ยวกับ TCP/IP ,การกำหนดชื่อ และเลข IP อินเตอร์เน็ตนับว่าเป็นเครือข่ายบนเครือข่าย ที่เปิดโอกาสให้เครื่องข่ายคอมพิวเตอร์อื่นๆ เชื่อมโยงเข้ามาใช้งาน หรือเป็นศูนย์กลางเชื่อมโยงเครือข่ายคอมพิวเตอร์อื่นๆอีก แต่ปัญหาที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันคือ แต่ละเครือข่ายใช้คอมพิวเตอร์ต่างชนิด ต่างยี่ห้อ และต่างระบบปฏิบัติการ มาตรฐานของ TCP/IP จึงถูกใช้เป็นกุญแจสำคัญในการแก้ปัญหาเหล่านี้ โดยจะกลายเป็นระบบเปิดที่สมบรูณ์แบบที่มีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่พีซีจนถึงเมนเฟรม และไม่จำกัดระบบปฏิบัติการที่ใช้ TCP/IP จึงเป็นมาตรฐานที่ทั่วโลกยอมรับ มีอุปกรณ์และซอฟต์แวร์ผลิตออกมาสนับสนุน TCP/IP มากมาย ดังนั้นจึงนับได้ว่า TCP/IP เป็นหัวใจของอินเตอร์เน็ตเลยทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 TCP/IP คืออะไร

TCP/IP เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Networking Protocol) จัดทำขึ้นเพื่อให้เป็นกฎเกณฑ์ให้เครื่องคอมพิวเตอร์ใช้งานร่วมกัน ในลักษณะของระบบเปิด (Open System) คือไม่ว่าจะเป็นเครื่องคอมพิวเตอร์ชนิดใดหรือระบบใดก็ตาม จะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันได้ เช่น เครื่องคอมพิวเตอร์ของ Digital Equipment ซึ่งเป็นระบบ Mini ติดต่อสื่อสารกับ Compaq ซึ่งเป็น PC ได้ เมื่อดำเนินการด้วย TCP/IP

TCP/IP เป็นการกำหนดรูปแบบการสื่อสารระหว่างซอฟต์แวร์ การจัดการโอนย้ายข้อมูล การแสดงสถานะของเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย ตลอดจนกฎระเบียบต่างๆที่กำหนดให้ทำ เมื่อเกิดการผิดพลาดหรือต้องทำเพื่อป้องกันไม่ให้เกิดการผิดพลาด

TCP/IP เกิดจากการนำข้อกำหนดของรูปแบบต่างๆกันมาใช้ร่วมกัน TCP และ IP ต่างก็เป็นรูปแบบหนึ่งของชุดข้อกำหนดนี้ (แต่เรียกชุดกำหนดรูปแบบนี้ว่า TCP/IP) ถูกออกแบบมาเพื่อใช้รับส่งหรือโอนย้ายข้อมูลระหว่างคอมพิวเตอร์ที่อยู่บนระบบเครือข่ายเดียวกันหรือต่างเครือข่ายกันก็ได้และมีการจัดเตรียมข้อมูลสถานะของเครือข่ายขึ้นได้ภายในดังข้อกำหนดรูปแบบเอง ในการสร้างซอฟต์แวร์ของระบบเครือข่ายจะใช้ TCP/IP เป็นส่วนสนับสนุนได้ที่งระบบเครือข่ายเฉพาะบริเวณ (Local Area Network) และเครือข่ายบริเวณกว้าง (Wide Area Network) ไม่ได้ใช้งานเฉพาะกับอินเตอร์เน็ตเท่านั้น

2.4.2 ส่วนประกอบของ TCP/IP

จากที่กล่าวมาแล้วข้างต้นว่า TCP/IP ประกอบไปด้วยชุดข้อกำหนดรูปแบบต่างๆซึ่งแบ่งเป็นกลุ่มได้ดังนี้

<u>กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocol)</u>

ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูลระหว่างคอมพิวเตอร์ 2 เครื่อง แบ่งย่อยออกได้เป็น 2 ชนิด

1.TCP (Transmission Control Protocol) เป็นการบริการแบบ Connection Based Service ซึ่งคอมพิวเตอร์ด้านผู้รับและผู้ส่งต้องต่อถึงกันอยู่ตลอดเวลาในระหว่างการสื่อสาร ถ้าเปรียบเทียบก็คล้ายกับระบบโทรศัพท์ที่ต้องติดต่อถึงกันให้ได้ก่อนจะพูดคุยกันได้

2.UDP (User Datagram Protocol) เป็นการให้บริการแบบ Connections Service คอมพิวเตอร์ด้านส่งไม่จำเป็นต้องติดต่อกับด้านผู้รับก่อน เพียงรู้ที่อยู่ของด้านผู้รับแล้วใส่ที่อยู่นั้นไปกับข้อมูลที่ส่งออกไป ข้อมูลจะเดินตามเส้นทางต่างๆเพื่อไปถึงปลายทางที่อยู่ คล้ายกับการส่งจดหมายที่ไปรษณีย์ จะส่งให้ตามที่อยู่ที่จ่าหน้าซองจดหมายโดยผู้รับและผู้ส่งไม่ได้ติดต่อถึงกัน

<u>กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocol)</u>

ทำหน้าที่พิจารณาเส้นทางที่ดีที่สุดในการส่งข้อมูลและถ้ามีข้อมูลเป็นจำนวนมากหรือมีขนาดใหญ่ กลุ่มข้อกำหนดรูปแบบนี้ก็จะทำการแบ่งย่อยข้อมูลให้มีขนาดเหมาะสม แล้วส่งออกไป เมื่อถึงผู้รับปลายทาง กลุ่มข้อกำหนดนี้ก็จะทำหน้าที่ตรงข้าม คือรวบรวมข้อมูลย่อยให้ถูกต้องก่อนการแสดงผล กลุ่มข้อกำหนดรูปแบบกลุ่มนี้ประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. IP (Internet Protocol) เป็นข้อกำหนดรูปแบบการส่งข้อมูล
2. ICMP (Internet Control Message Protocol) เป็นข้อกำหนดรูปแบบของข้อมูลข่าวสารเกี่ยวกับสถานะของ IP เช่น ข่าวสารความผิดพลาดและผลกระทบต่อเส้นทางเมื่อมีการเปลี่ยนแปลงฮาร์ดแวร์ในเครือข่าย
3. RIP (Routing Information Protocol) ข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่ดีที่สุด เหมาะสมกับข้อมูลมากที่สุด
4. OSPF (Open Shortest Path First) ข้อกำหนดรูปแบบอีกประเภทหนึ่งที่ใช้ตัดสินเลือกเส้นทางโดยพิจารณาจากเส้นทางที่สั้นที่สุดก่อน

<u>กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address)</u>

ทำหน้าที่พิจารณาที่อยู่ของเครือข่ายและเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นลักษณะตัวเลขหรือชื่อก็ตาม เพื่อความถูกต้องของข้อมูลที่จะไปยังผู้รับปลายทาง โดยที่ไม่ว่าเครือข่ายจะใหญ่โตเพียงใด หรือมีเครื่องคอมพิวเตอร์จำนวนมากมายก็ตาม ที่อยู่จะต้องไม่ซ้ำกัน กลุ่มข้อกำหนดรูปแบบกลุ่มมีดังนี้

1. ARP (Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เพื่อไม่ให้เกิดที่อยู่ซ้ำกัน
2. DNS (Domain Name System) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เมื่อรู้ชื่อของเครือข่ายหรือเครื่องคอมพิวเตอร์ เพราะการใช้งานจริงนั้นใช้เพียงที่อยู่ที่เป็นตัวเลข แต่ระบบที่ชื่อจัดทำขึ้นเพื่อให้สะดวกต่อการใช้งานของผู้ใช้
3. RARP (Reverse Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เช่นเดียวกับ ARP แต่จะทำตรงข้ามกับ ARP

<u>กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางการสื่อสารเกี่ยวกับเครือข่าย (Gateway Protocol)</u>

สนับสนุนข้อมูลสถานะเพื่อนำไปใช้เลือกเส้นทางที่เหมาะสม ข้อกำหนดรูปแบบเหล่านี้ประกอบด้วย

1. EGP (Exterior Gateway Protocol) ข้อกำหนดรูปแบบนี้จะทำการถ่ายโอน ข้อมูลเส้นทางกันระหว่าง Gateway กับเครือข่ายภายนอกเพื่อทำการสื่อสาร
2. GGP (Gateway-to-Gateway Protocol) เป็นข้อกำหนดรูปแบบที่ทำงานถ่ายโอนข้อมูลเส้นทางกันระหว่าง Gateway กับ Gateway
3. IGP (Interior Gateway Protocol) ข้อกำหนดรูปแบบที่ถ่ายโอนข้อมูลเส้นทางกันระหว่างภายในเครือข่ายเดียวกัน

Gateway เป็นอุปกรณ์หรือเครื่องคอมพิวเตอร์ที่ทำงานเสมือนประตูสื่อสาร เป็นช่องสัญญาณเข้าหรือออกไปยังระบบสื่อสารอื่น หรือภายในเครือข่ายเดียวกันติดต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Services)

ผู้ใช้สามารถใช้ข้อกำหนดรูปแบบได้โดยตรง ข้อกำหนดรูปแบบนี้ประกอบด้วย

1. BOOTP (BOOT Protocol) เมื่อผู้ใช้เปิดเครื่องคอมพิวเตอร์บนเครือข่ายเริ่มทำงานข้อกำหนดแบบรูปนี้จะอ่านโปรแกรมควบคุมการทำงานจากเครื่องคอมพิวเตอร์ให้บริการ (Service Computer) มาให้
2. FTP (File Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนไฟล์ข้อมูลระหว่างเครื่องคอมพิวเตอร์ ซึ่งอาจจะอยู่บนเครือข่ายเดียวกันหรือต่างเครือข่ายกันก็ได้
3. TELNET เป็นข้อกำหนดรูปแบบที่ใช้บริการเกี่ยวกับการควบคุมการติดต่อระยะทางไกล (Remote Login)

กลุ่มข้อกำหนดรูปแบบอื่นที่นอกเหนือจากกลุ่มที่จัดไว้และบริการที่สำคัญๆ จัดทำไว้ให้บนระบบเครือข่ายที่สนใจมีดังนี้

- NFS (Network File System) เป็นข้อกำหนดรูปแบบที่ทำให้ผู้ใช้คอมพิวเตอร์เครื่องหนึ่งสามารถเข้าไปดูไฟล์ข้อมูลและใช้งานไฟล์ข้อมูล ซึ่งอยู่ในคอมพิวเตอร์เครื่องอื่นได้
- NIS (Network Information Services) เป็นข้อกำหนดรูปแบบที่ให้บริการกับ User Accounts ข้ามเครือข่าย เช่น Logins และ Password
- RPC (Remote Procedure Call) ข้อกำหนดรูปแบบที่อำนวยความสะดวกให้กับโปรแกรมประยุกต์ที่ใช้งานกับการควบคุมระยะทางไกล
- SMTP (Simple Mail Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนจดหมายอิเล็กทรอนิกส์ (Electronic Mail) ระหว่างคอมพิวเตอร์
- SNMP (Simple Network Management Protocol) ข้อกำหนดรูปแบบที่ให้บริการข่าวสารต่างๆ ที่แสดงสถานะของเครือข่ายและอุปกรณ์ที่ต่ออยู่บนเครือข่าย

2.4.3 สถาปัตยกรรม TCP/IP

TCP/IP ออกแบบเป็นชุดกำหนดรูปแบบ แบ่งการบริหารออกเป็นกลุ่มๆ ภายในชุดข้อกำหนดรูปแบบเช่น กลุ่มบริการผู้ใช้ กลุ่มการขนส่ง และกลุ่มเกี่ยวกับเครือข่าย ดังกล่าวมาแล้ว จากกลุ่มต่างๆที่แบ่งไว้นำไปพัฒนาเป็นสถาปัตยกรรมซึ่งมีลักษณะเป็นชั้น (Layer) ตามกลุ่มการให้บริการ

สถาปัตยกรรมแบบระดับชั้น มีข้อดีอยู่หลายประการ แต่ละชั้นมีอิสระไม่ขึ้นต่อกัน ทำให้การเปลี่ยนแปลงการบริการของชั้นใดๆ ไม่ก่อปัญหากับการบริการในชั้นอื่น การเพิ่มเติมการบริการใหม่ทำได้โดยไม่ต้องเปลี่ยนแปลงโปรแกรมระบบเดิมและสิ่งสำคัญที่สุดก็คือการทำระดับชั้นนั้นทำให้ตัวโปรแกรมมีขนาดเล็ก สามารถระบุส่วนที่จะต้องปรับปรุงได้อย่างแน่นอน ไม่ต้องวิตกกังวลถึงโปรแกรมส่วนอื่นทำให้การพัฒนาประสิทธิภาพของระบบทำได้ง่ายและดียิ่งขึ้น (รูปที่ 2.7)

ชุดข้อกำหนดรูปแบบ TCP/IP เมื่อนำไปใช้งานจะอยู่ระหว่างระดับชั้นที่เป็นฮาร์ดแวร์กับระดับชั้นที่เป็นซอฟต์อื่น เมื่อติดตั้งกับบางระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application	
TCP/IP	User Interface (Windows,X,etc.)
Operating System	
Hardware	

รูปที่ 2.7 แสดงสถาปัตยกรรมของ TCP/IP

TCP/IP เมื่อนำไปใช้งานกับระบบWindows ต้องเพิ่ม Drivers พิเศษเข้าไปด้วยเพื่อให้ Windows รู้จักข่าวสารที่เป็นแพ็กเกจ (Packet) และส่งแพ็กเกจเข้าสู่ระบบเครือข่ายได้ แต่ถ้าเป็นซอฟต์แวร์ที่ใช้ชุดข้อกำหนดรูปแบบ TCP/IP อยู่แล้วข่าวสารจากโปรแกรมประยุกต์จะผ่านมายังระดับชั้น TCP/IP ทำการประมวลส่งเข้าเครือข่ายโดยให้ระบบปฏิบัติการทำการส่ง

2.4.4 กลไกของโปรโตคอล IP

ในการส่งผ่านข้อมูล หรือ IP datagram ไปยังเครือข่ายอินเตอร์เน็ตนั้น โปรโตคอล IP จะทำหน้าที่พิจารณาว่าปลายทางในการส่ง IP datagram นั้นจะเป็นภายในเครือข่ายของตนเองหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยการพิจารณานี้ โปรโตคอล IP จะตรวจสอบจากค่า IP address ปลายทางว่าส่วนที่เป็นค่าหมายเลขเครือข่าย (network address) จะเหมือนกับค่าหมายเลขเครือข่ายของ IP address ต้นทางหรือไม่ ถ้าค่าตรงกันแสดงว่าการส่งข้อมูลอยู่ภายในเครือข่ายเดียวกัน แต่ถ้าค่าต่างกัน แสดงว่าต้องส่งข้อมูลไปยังปลายทางที่อยู่คนละเครือข่ายกัน

<u>การส่งข้อมูลภายในเครือข่ายเดียวกัน</u> มีกลไกดังนี้

1. โปรโตคอล IP จะเรียกใช้บริการ โปรโตคอล ARP (Address Resolution Protocol) เพื่อแปลงค่า IP ปลายทางให้เป็นค่าหมายเลขฮาร์ดแวร์ เช่น MAC address

2. โปรโตคอล IP ได้รับค่าหมายเลขฮาร์ดแวร์แล้ว ก็จะส่งข้อมูลนั้นไปยังฮาร์ดแวร์ที่ระบุไว้

<u>การส่งข้อมูลข้ามเครือข่าย</u> มีกลไกดังนี้

1. โปรโตคอล IP ตรวจสอบพบว่าหมายเลข IP address ปลายทางอยู่คนละเครือข่ายกัน โดยโปรโตคอล IP จะอ่าน IP address ของ Router เพื่อเตรียมส่งข้อมูลไปที่ Router แทน ซึ่งในที่นี้จะมีการกำหนดเป็น default router

2. โปรโตคอล IP จะเรียกใช้บริการ โปรโตคอล ARP เพื่อแปลงค่า IP address ของ Router ให้เป็นค่าหมายเลขฮาร์ดแวร์

3. โปรโตคอล IP ส่งข้อมูล IP datagram ไปยัง Router ที่กำหนดไว้ จากนั้น Router ส่งข้อมูลข้ามเครือข่ายไปตามขั้นตอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล IP จะรู้ได้อย่างไรว่าเครือข่ายดังกล่าวมีการเชื่อมต่อ Router อยู่และมีค่า IP อะไร ซึ่งในเรื่องนี้ผู้ใช้จะต้องกำหนดค่าที่เรียกว่า default Router หรือ default Gateway เสียก่อน ว่ามีค่า IP address อะไร โดยสามารถสอบถามได้จากผู้ดูระบบ สำหรับกลไกการส่งผ่านข้อมูลต่อจาก Router ไปยังเครื่องปลายทางก็จะมีกลไกเดียวกัน

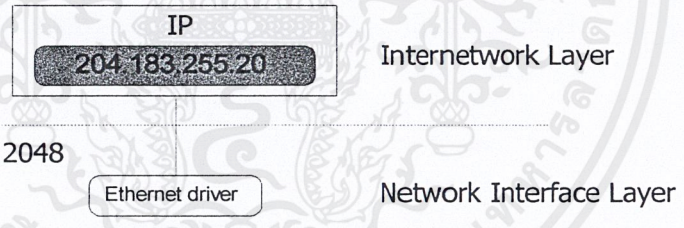
การกำหนด ให้กับอุปกรณ์

มีคำถามอยู่ว่าเราจำเป็นต้องกำหนดหมายเลข IP address ให้กับอุปกรณ์ทุกชิ้นในเครือข่ายหรือไม่ คำตอบคือ ไม่จำเป็นต้องกำหนดทั้งหมดก็ได้ แต่มีหลักอยู่ว่า เราจะต้องกำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อเข้าสู่เครือข่ายทุกจุด จุดเชื่อมต่อหรือ interface อาจจะหมายถึง Network Interface Card (การ์ด LAN) ที่ติดตั้งในเครื่องเซิร์ฟเวอร์หรือ WAN port, Ethernet port ที่ Router ใช้เชื่อมต่อเข้ากับเครือข่าย เป็นต้น การกำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อนี้ทำให้เราเข้าใจได้ว่าในบางอุปกรณ์ที่มีจุดเชื่อมต่อเข้าเครือข่ายมากกว่าหนึ่งจุด จะต้องกำหนดหมายเลข IP address ให้ครบ

การ Bind IP address

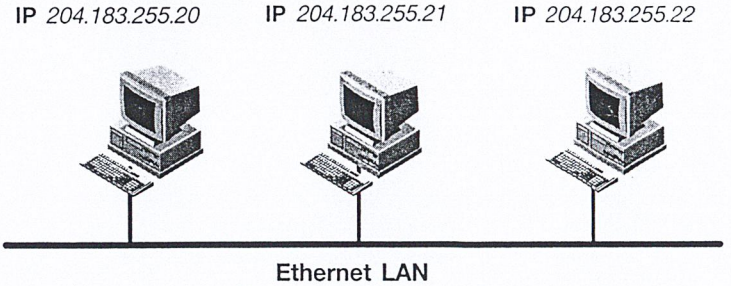
เมื่อได้กำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อ เช่น LAN card แล้วที่เครื่องเซิร์ฟเวอร์จะต้องมีการ bind หรือผนวกค่า IP address ดังกล่าวเข้ากับ Ethernet driver เพื่ออ้างอิงหมายเลข IP กับฮาร์ดแวร์ ให้ทำหน้าที่ติดต่อส่งข้อมูลในระดับ network interface ได้ต่อไป

ดังตัวอย่างในรูป แสดงการ bind IP address หมายเลข 204.183.255.20 เข้ากับ Ethernet driver ซึ่งเป็น network interface driver ทำให้ IP สามารถสื่อสารกับเครือข่ายได้ ส่วนหมายเลข 2048 เป็น interface identifier



รูปที่ 2.8 แสดงการ bind IP address

จากรูปจะแสดงค่า bind IP address 204.183.255.2.0 เข้ากับ Ethernet driver (ในกรณีนี้ใช้เครือข่ายแบบ Ethernet) โปรโตคอล IP จะใช้ค่า IP address นี้ในการติดต่อกันและผ่านฮาร์ดแวร์ที่ถูก bind ไว้อีกต่อหนึ่ง ค่าหมายเลขฮาร์ดแวร์ก็ได้แก่ MAC address ที่มีประจำอยู่บน LAN card ซึ่งจะไม่ได้ใช้งานอ้างอิงโดยตรง แต่จะผ่านหมายเลข IP address แทน



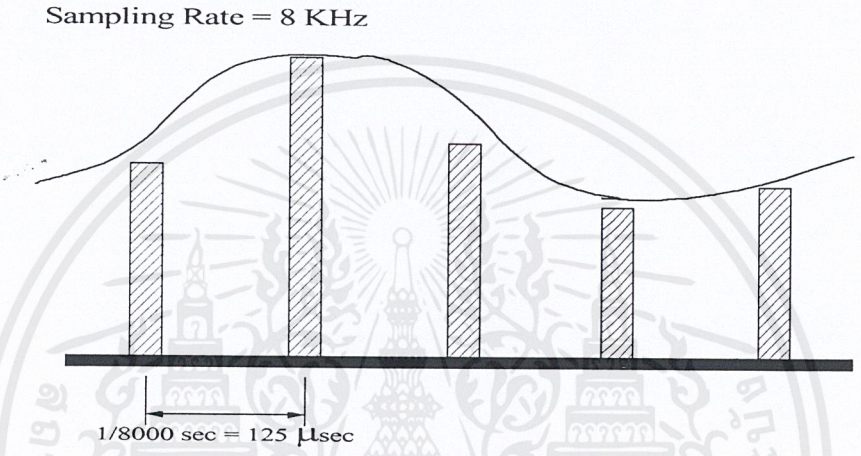
รูปที่ 2.9 แสดงเครื่องคอมพิวเตอร์ในเครือข่าย Ethernet มีหมายเลข IP address แต่ละเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

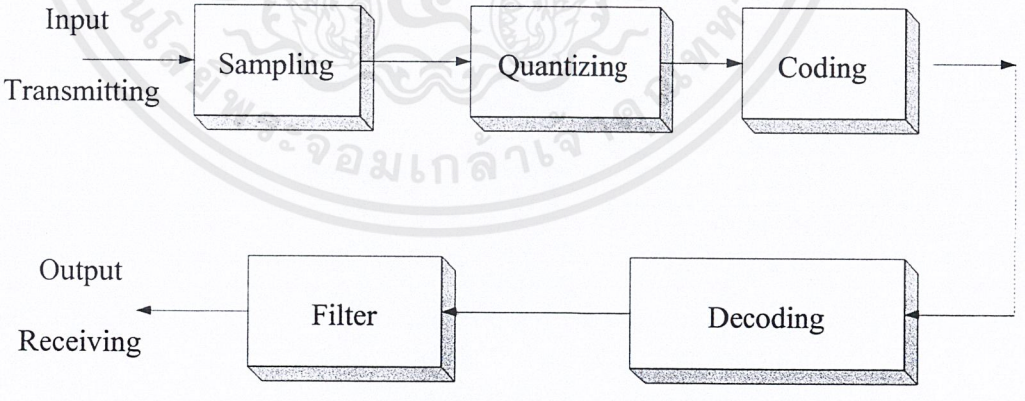
จากรูปตัวอย่าง เป็นรูปเครือข่าย Ethernet อย่างง่าย จะเห็นว่ามีการกำหนดหมายเลข IP address ให้กับ LAN card ที่เป็นจุดเชื่อมต่อเข้าเครือข่ายทุกจุด แต่เนื่องจากเครื่องทุกเครื่องมีจุดเชื่อมต่อเพียงจุดเดียว ทำให้เราสามารถอ้างอิงเครื่องกับหมายเลข IP address นั้นได้ตรงกัน

2.5 พัลส์โค้ดมอดดูเลชั่น (Pulse Code Modulation)

ในปัจจุบันระบบพัลส์โค้ดมอดดูเลชั่น ถูกนำมาใช้อย่างกว้างขวางโดยเฉพาะสัญญาณเสียงและในระยะหลัง ๆ นี้ การพัฒนาระบบนี้เพื่อการนำไปใช้กับสัญญาณภาพก็มีบทบาทขึ้นด้วย ดังนั้นในบทนี้จะกล่าวถึงขั้นตอนของกระบวนการ พัลส์โค้ดมอดดูเลชั่น อย่างคร่าว ๆ



รูปที่ 2.10 แสดงการแซมปลิ้งสัญญาณ



รูปที่ 2.11 กระบวนการเข้าและถอดรหัสของพัลส์โค้ดมอดดูเลชั่น

จากรูปที่ 2.11 เป็นการแสดงขั้นตอนการประมวลเพื่อให้ได้รับพัลส์โค้ดมอดดูเลชั่น อย่างกว้าง ๆ ก็คือการเข้ารหัส (Coding) และการนำสัญญาณไปแปลงกลับซึ่งเรียกว่า การถอดรหัส (Decoding) เพื่อให้ได้สัญญาณเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

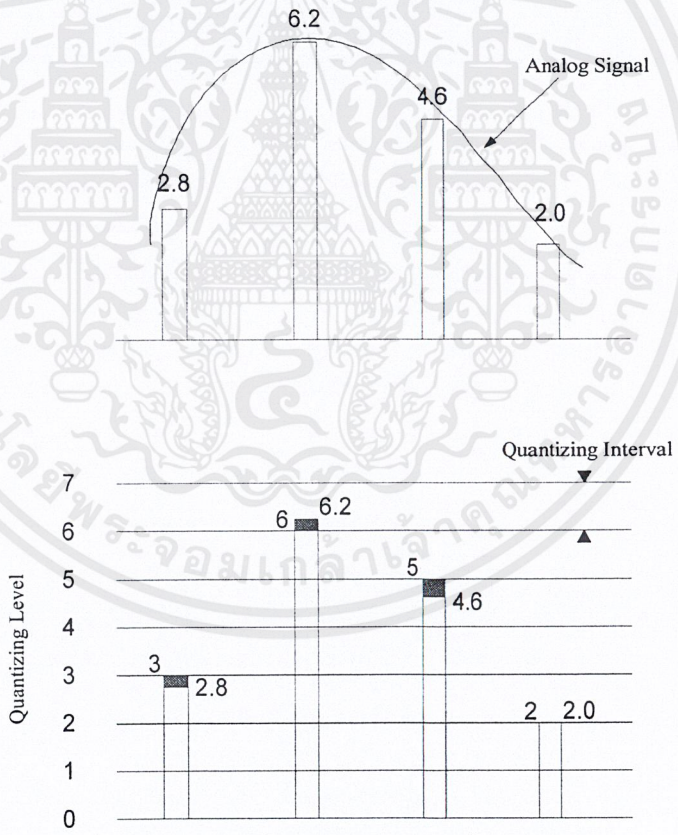
2.5.1 แซมปลิ้ง (Sambling)

แซมปลิ้ง คือการทำสัญญาณซึ่งมีค่าต่อเนื่องให้เป็นดิสครีท (Discrete) ในช่วงเวลาที่เท่า ๆ กัน โดยใช้ทฤษฎีแซมปลิ้งที่ว่า ถ้าเก็บแซมปลิ้งด้วยอัตรา 2 เท่า หรือ มากกว่าความถี่สูงสุดของสัญญาณอนาล็อกแล้ว จะสามารถทำให้สัญญาณเดิมกลับคืนมาได้อย่างถูกต้อง สัญญาณที่ผ่านการแซมปลิ้งแล้วจะได้สัญญาณ พัลส์แอมปลิจูดมอดดูเลชั่น (PAM : Pulse Amplitude Modulation)

2.5.2 ควอนไตซิ่ง (Quantizing)

ขบวนพัลส์แอมปลิจูดมอดดูเลชั่น ที่ผ่านการแซมปลิ้งแล้ว ยังถือว่าเป็นสัญญาณอนาล็อกอยู่ คือมันจะมีขนาดเปลี่ยนอย่างต่อเนื่องไปกับเวลา การจัดระดับคือ กระบวนการเปลี่ยนขนาดของพัลส์แอมปลิจูดมอดดูเลชั่น เหล่านั้นให้เป็นตัวเลขเป็นขั้น ๆ ตามค่าที่กำหนดไว้

แอมปลิจูดของแซมปลิ้งทุกตัวจะถูกจัดให้เป็นระดับ ซึ่งเรียกว่า ควอนไตซิ่งลีเวล (Quantizing Level) โดยมีระยะห่างระหว่างระดับข้างเคียง เรียกว่า ควอนไตซิ่งอินเตอร์วอล (Quantizing Interval) กรณีนี้เรียกว่า การจัดระดับแบบยูนิฟอร์ม (Uniform) หรือ แบบลิเนียร์ (Linear)

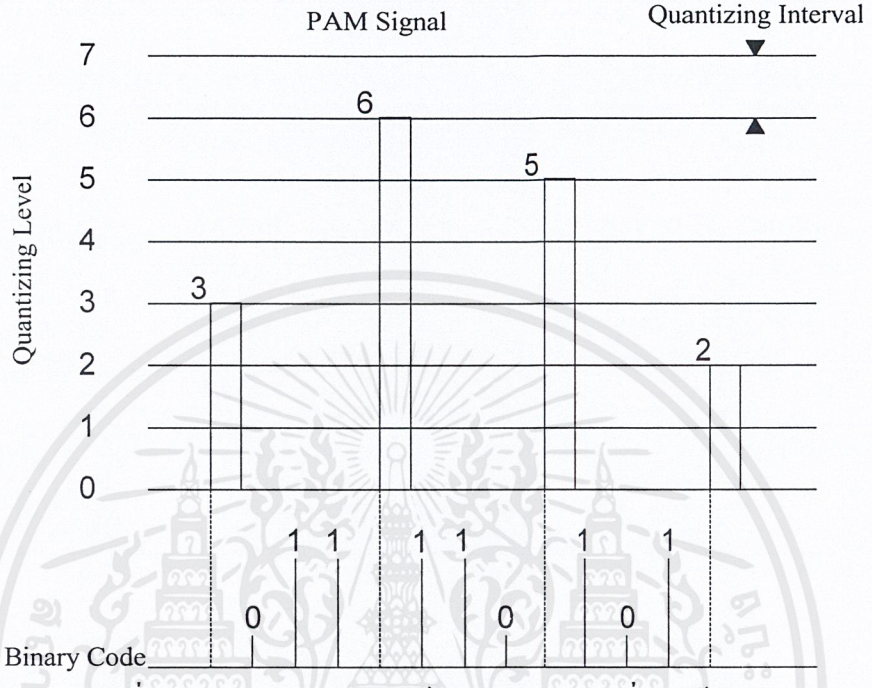


รูปที่ 2.12 แสดงการแปลงสัญญาณพัลส์แอมปลิจูดมอดดูเลชั่นให้เป็นตัวเลข (การจัดระดับ)

ขนาดของแซมปลิ้งทุกตัวจะแสดงด้วยค่าระดับควอนไตซ์ที่ใกล้เคียงที่สุด จะเห็นว่าสัญญาณพัลส์แอมปลิจูดมอดดูเลชั่น ที่ถูกจัดระดับแล้วนี้จะเป็นเพียงค่าโดยประมาณเท่านั้น ดังนั้นส่วนเกินและส่วนขาดจากการจัดระดับจึงเป็นค่าผิดพลาดระหว่างสัญญาณเดิมและค่าที่ได้จัดระดับซึ่ง ค่าผิดพลาดนี้เรียกว่า ควอนไตซิ่งน้อยส์ (Quantizing Noise)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการให้ทำควอนไตซิ่งน้อยส์นี้ลดลงในเบื้องต้นคือการลดควอนไตซิ่งอินเตอร์วอลให้แคบลง เช่นถ้าลดควอนไตซิ่งอินเตอร์วอลลงครึ่งหนึ่ง ปริมาณขอควอนไตซิ่งน้อยส์จะลดลง ¼ และการลดควอนไตซิ่งอินเตอร์วอลจะสอดคล้องกับการเพิ่มจำนวนบิทอีก 1 บิท นั่นคือควอนไตซิ่งน้อยส์จะลดลง 1 dB ทุก ๆ การเพิ่ม 1 บิท



รูปที่ 2.13 การนำสัญญาณพัลส์แอมปลิจูดมอดดูเลชั่นมาเข้ารหัส

2.5.3 การเข้ารหัส (Coding)

ระดับการควอนไตซ์	รหัสธรรมดา	รหัสแบบเกรย์	รหัสแบบสมมาตร
0	000	000	011
1	001	001	010
2	010	011	001
3	011	010	000
4	100	110	100
5	101	111	101
6	110	101	110
7	111	100	111

ตารางที่ 2.8 รหัสไบนารีแบบต่าง ๆ

หลังจากขบวนพัลส์แอมปลิจูดมอดดูเลชั่น ได้ผ่านการจัดระดับมาแล้ว จะต้องเปลี่ยนขนาดเหล่านั้นเป็นรหัสไบนารี กรณีที่เป็นการส่งสัญญาณเสียงทางโทรศัพท์ จะถูกเปลี่ยนเป็นรหัส 8 บิท ซึ่งสามารถแสดงค่าแอมปลิจูดได้ 2^8 ระดับ ระบบการเข้ารหัสมีหลายแบบ แต่ส่วนมากจะใช้กัน 3 แบบ ดังแสดงไว้ในตาราง

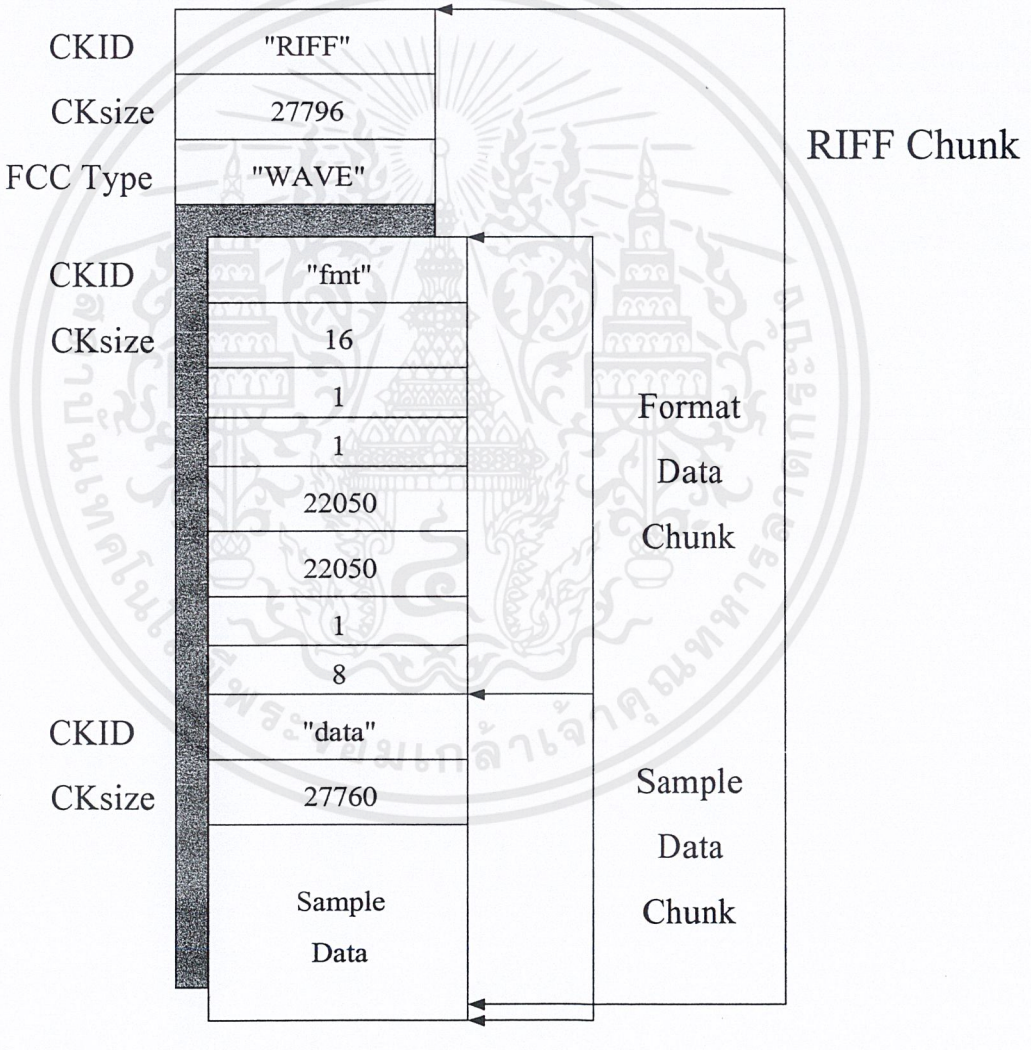
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ไฟล์สกุล .wav

<u>รูปแบบการบันทึกคลื่นเสียง</u>

สำหรับการบันทึก 1 ช่องสัญญาณเสียงทั่วไปจะบันทึกเป็นโหมดโมโน และถ้าต้องการบันทึกเป็น 2 ช่องสัญญาณเสียงแยกออกเป็น ช่องสัญญาณเสียงข้างซ้ายและขวาจะบันทึกในโหมดสเตอริโอ ซึ่งเหมาะกับการบันทึกเสียงดนตรี

เสียงพูดจะใช้การแซมปลิ้งที่ความถี่ 11025 Hz ส่วนความถี่แซมปลิ้งที่ 22050 Hz จะให้คุณภาพใกล้เคียงกับเทปบันทึกเสียง และความถี่แซมปลิ้งที่ 44100 Hz จะเป็นความถี่ที่คุณภาพเสียงเดียวกับเพลงที่บันทึกในแผ่น CD ขนาดของแต่ละแซมเปิลจะมีสองระดับ คือ 8 บิต และ 16 บิต



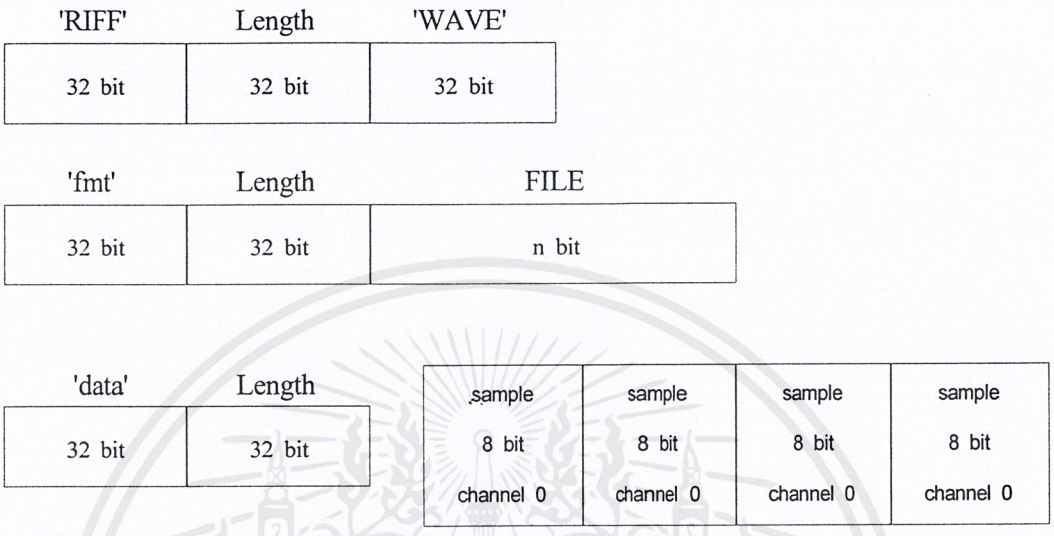
รูปที่ 2.14 โครงสร้างของไฟล์ .wav

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของไฟล์ .wav มีอยู่ 4 ประเภท คือ

1.ไฟล์ 8 บิตโมโน (8-bit Mono)

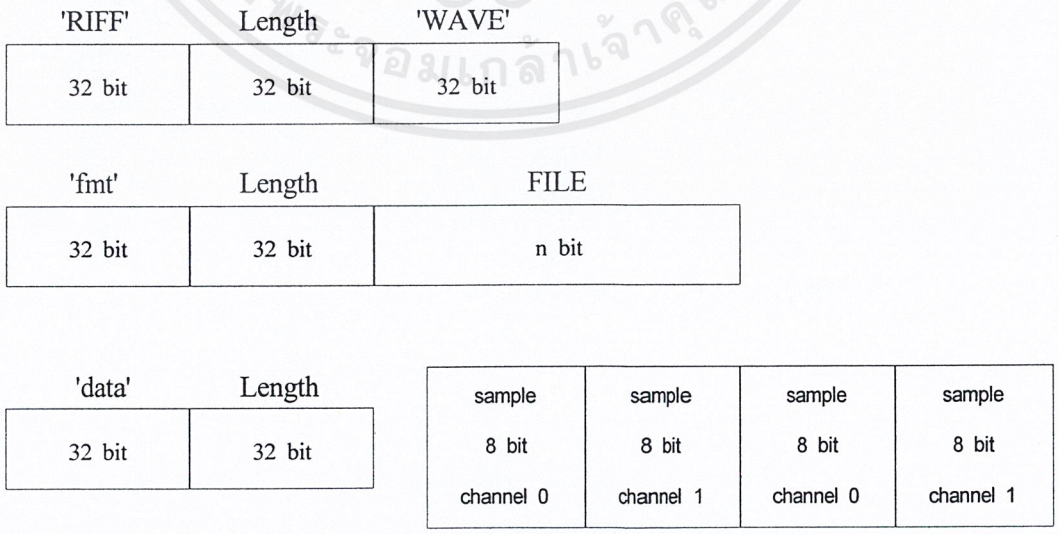
ตัวอย่างแบบ 8 บิตโมโนนี้ ไบท์ (byte) ทั้งหมดถูกเก็บเรียงตามลำดับโดยช่องสัญญาณที่ 0 (Channel 0) จะถูกนำมาใช้สำหรับตัวอย่าง



รูปที่ 2.15 แสดงองค์ประกอบของส่วนหัวไฟล์ .wav แบบ 8 บิต โมโน

2. ไฟล์ 8 บิต สเตอริโอ (8-bit Stereo)

ตัวอย่างแบบสเตอริโอนี้ ช่องสัญญาณ 0 คือ ช่องสัญญาณทางด้านซ้าย และช่องสัญญาณที่ 1 คือ ช่องสัญญาณทางด้านขวา ข้อมูลจะเก็บอยู่ตามรูปแบบนี้ คือ 8 บิต สำหรับช่องสัญญาณที่ 0 , 8 บิต สำหรับช่องสัญญาณที่ 1 , 8 บิต สำหรับช่องสัญญาณที่ 0 , 8 บิต สำหรับช่องสัญญาณที่ 1, ... ซึ่งหมายความว่า 1 แซมเปิล ประกอบด้วย 2 ไบท์ ไบท์แรกสำหรับช่องสัญญาณทางซ้าย และอีกไบท์สำหรับช่องสัญญาณทางขวา



รูปที่ 2.16 แสดงองค์ประกอบของส่วนหัวไฟล์ .wav แบบ 8 บิต สเตอริโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

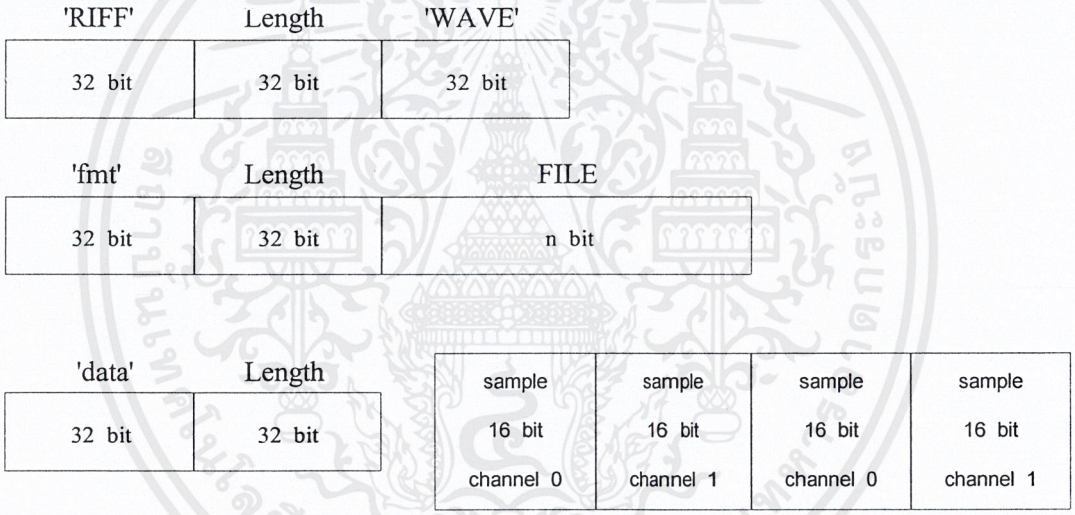
3.ไฟล์ 16 บิต โมโน (16-bit Mono)

การแสดงตัวอย่างแบบ 16 บิต โมโน ในหน่วยความจำนี้ ถูกใช้สำหรับการบันทึกแต่ละแซมเปิลแบบเรียงลำดับของไบท์ข้อมูลเหมือนกับแบบ 8 บิต โมโน

4.ไฟล์ 16 บิต สเตอริโอ (16-bit Stereo)

ตัวอย่างแบบ 16 บิต สเตอริโอ ต้องการหน่วยความจำจำนวนมากสำหรับแต่ละแซมเปิล ซึ่งต้องการถึง 4 ไบท์ (Byte) โดย 2 ไบท์สำหรับช่องสัญญาณทางซ้าย และอีก 2 ไบท์สำหรับช่องสัญญาณทางขวา ลำดับของการเรียงไบท์ข้อมูล คือ 1 ไบท์ต่ำ (Low byte) และ 1 ไบท์สูง (High byte) สำหรับแต่ละช่องสัญญาณ

ในการเก็บแบบแซมเปิลขนาด 8-bit ค่าของแต่ละไบท์ อยู่ระหว่าง 0 ถึง 255 ดังนั้น ค่า 128 แสดงค่ากลางของรูปคลื่นแบบดิจิตอล (median of the digitized waveform) สำหรับตัวอย่างแบบ 16-bit พิสัยของค่าอยู่ระหว่าง -32768 ถึง 32767 ในกรณีนี้ค่า 0 เป็นค่ากลาง



รูปที่ 2.17 แสดงองค์ประกอบของส่วนหัวไฟล์ .wav แบบ 16 บิต สเตอริโอ

2.7 วินโดวส์ ซ้อกเก็ต (Windows Sockets)

กุญแจที่จะนำไปสู่การเขียนโปรแกรมบนอินเตอร์เน็ต (Internet Programming) โดยใช้วินโดว์คือ "ซ้อกเก็ต" (Sockets) โดยทั่วไปแล้วซ้อกเก็ตหมายถึง เซต เซตหนึ่ง (a set) ของฟังก์ชันที่ใช้ทำการติดต่อแบบสองทาง (Two-way Communication) โดยใช้โปรโตคอล TCP/IP หรือโปรโตคอลอื่นๆ

ส่วนถ้าหากเราพิจารณาคำว่าวินโดว์ ซ้อกเก็ต (Windows Sockets) แล้วจะหมายความถึง API มาตรฐานที่ถูกพัฒนาโดยกลุ่มบริษัทกว่า 20 บริษัทและถูกใช้เป็นมาตรฐานการเปิดเครือข่ายเพื่อจัดการการติดต่อโดยใช้โปรโตคอล TCP/IP (ถึงแม้ว่าจะมีการเผื่อไว้สำหรับใช้กับโปรโตคอลอื่นภายหลังก็ตาม) ถ้าหากเราต้องการเขียนโปรแกรมบนอินเตอร์เน็ตสำหรับวินโดว์แล้ว เราจะต้องรู้จักซ้อกเก็ตและ วินโดว์ซ้อกเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 ซ้อกเก็ต คืออะไร

ซ้อกเก็ตมาจากระบบปฏิบัติการยูนิกซ์ (UNIX) และโดยเฉพาะอย่างยิ่งมาจากเบิร์กเลย์ยูนิกส์ (Berkley UNIX : BSD) ช่วงประมาณปีคริสตศักราช 1980 ซ้อกเก็ตหนึ่งซ้อกเก็ตคือชิ้นส่วนหนึ่ง (a piece) ของซอฟต์แวร์ (Software) ซึ่งสามารถรับและส่งข้อมูลบนเครือข่าย TCP/IP ได้ส่วนของข่าวสารข้อมูล (Information) เกี่ยวกับซ้อกเก็ตที่สำคัญมีอยู่ 3 ส่วนคือ

1. IP Address ซ้อกเก็ตกำลังติดต่อ
2. พอร์ตที่ซ้อกเก็ตใช้ติดต่อกับแอดเดรสนั้น
3. ชนิดของซ้อกเก็ต

สิ่งหนึ่งที่เกี่ยวกับพอรต์ที่จำเป็นต้องทำความเข้าใจให้ถูกต้องนั่นคือ พอร์ตในที่นี้ไม่เหมือนกับพอร์ตขนานหรือพอร์ตอนุกรม พอร์ตเหล่านี้ไม่ได้เป็นฮาร์ดแวร์ (Hardware) ใดๆที่งสิ้นพอร์ตในที่นี้เป็นเพียงสภาวะ (Convention) ที่ถูกใช้โดยโปรแกรมที่รันอยู่บนเครื่องที่กำลังทำการติดต่อสื่อสาร โปรโตคอล TCP/IP จะให้เครื่องหนึ่งสามารถติดต่อกับหลายๆเครื่องในขณะเดียวกันได้ ยกตัวอย่างเช่น เราสามารถเปิดเวบบราวเซอร์ให้ โหลดเพจขนาดใหญ่ๆขณะเดียวกันก็ให้นิวส์รีดเดอร์จัดเรียงและอินเดกซิ่งบทความขนาดใหญ่ และโปรแกรม FTP ก็กำลังดาวน์โหลดไฟล์ขนาดใหญ่ไฟล์หนึ่ง และทันใดนั้นก็มีอีเมล์ส่งมาถึงพอดี เราก็เลยไปอ่านเมล์โดยปล่อยให้มีการดาวน์โหลด,อินเดกซิ่งและอื่นๆต่อไป สภาวะต่างๆของการติดต่อบนอินเตอร์เน็ตระหว่างเครื่องของเรากับเครื่องอื่น สภาวะเหล่านี้จะเกิดขึ้นบนพอร์ตที่แตกต่างกัน บริการมาตรฐานบนอินเตอร์เน็ต เช่น เวบนิวส์ FTP และเมล์ เหล่านี้จะมีพอร์ตมาตรฐานของแต่ละบริการ

ซ้อกเก็ตอยู่ 2 ชนิดได้แก่ สตรีมซ้อกเก็ต (stream socket) และดาต้าแกรมซ้อกเก็ต (Datagram socket) ซึ่งทั้ง 2 ชนิดใช้กับโปรโตคอลหลัก 2 ประเภทของ TCP/IP คือ TCP และ UDP โปรโตคอล TCP มีกระบวนการคล้ายกับการที่เครื่องสองเครื่องสร้างการติดต่อหนึ่งช่องการติดต่อ (from a connection) จากนั้นข้อมูลจะถูกแลกเปลี่ยนผ่านช่องการติดต่อนี้ เมื่อเสร็จก็จะปิดช่องการติดต่อนั้น ในทางตรงข้ามโปรโตคอล UDP จะไม่เกี่ยวข้องกับการสร้างการติดต่อเครื่องหนึ่งส่งข้อมูลให้ เครื่องอื่นโดยไม่สร้างช่องทางการติดต่อ สตรีมซ้อกเก็ตใช้สร้างโปรโตคอล TCP และถูกใช้กับงานที่จะต้องแลกเปลี่ยนข้อมูลจำนวนมากๆ หรืองานที่เน้นว่าลำดับก่อนหลังของข้อมูลที่เข้ามามีความสำคัญ ตัวอย่างของการใช้ TCP ที่เก่าแก่ซึ่งก็ใช้ สตรีมซ้อกเก็ตนั่นคือ โปรโตคอล FTP แต่ดาต้าแกรมซ้อกเก็ตใช้สร้างโปรโตคอล UDP และถูกใช้กับงานที่มีการแลกเปลี่ยนข้อมูลน้อยๆ หรืองานที่ไม่สนใจลำดับก่อนหลังของข้อมูล ตัวอย่างของการใช้งาน UDP ที่เก่าแก่เห็นจะเป็นคล็อกอัพเดทเตอร์ (clock updater) ซึ่งจะบรอดคาสท์เวลาของระบบไปสู่เครื่องอื่นๆ

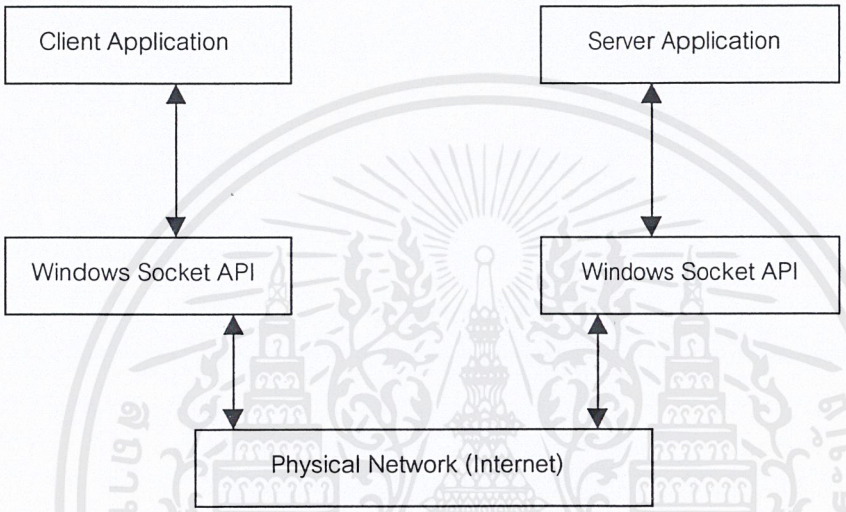
2.7.2 การได้รับ API ของวินโดว์ซ้อกเก็ต

เมื่อมีคนพูดถึง API ของวินโดว์ซ้อกเก็ต คนๆนั้นจะหมายถึงข้อตกลงที่เป็นที่ยอมรับของทุกๆบริษัท ซึ่งก็คือชื่อของฟังก์ชั่นต่างๆ และหน้าที่การทำงานของฟังก์ชั่นเหล่านี้ยกตัวอย่างเช่นข้อตกลงระบุว่า มีฟังก์ชันหนึ่งชื่อ connect() ซึ่งทำหน้าที่ “สร้างช่องการติดต่อระหว่างซ้อกเก็ตที่กำหนด”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตามเราไม่สามารถที่จะเอ็กซิคิวต์ "ข้อตกลง" ได้ เราต้องมีซอสโค้ด หรือโค้ดที่คอมไพล์แล้ว หลายบริษัทจึงจัดหาไดนามิกไลบารี่เรียกว่า WINSOCK.DLL ซึ่งทำให้เราสามารถเรียกฟังก์ชันได้ทุกฟังก์ชันที่ระบุไว้ในข้อตกลงดังกล่าว ตัวอย่างเช่นชุดของโปรแกรมซึ่ง โดยทั่วไปเรียกว่าทรัมเป็ด (Trumpet) ของปีเตอร์ แทตแทม (Peter Tattam) จะมีไฟล์ WINSOCK.DLL ซึ่งจะมีฟังก์ชันเหล่านี้และฟังก์ชันอื่นที่ใช้ในการสร้างการเชื่อมต่อกับอินเตอร์เน็ต แต่ถ้าเป็นวินโดว์ 95 และวินโดว์ NT จะมี WINSOCK.DLL ต่างกันเล็กน้อย

API ของวินโดว์ซ้อกเก็ตจะเป็นตัวที่อยู่ระหว่างแอพพลิเคชันกับเน็ตเวิร์ก



รูปที่ 2.18 การติดต่อของวินโดว์แอพพลิเคชันผ่านเครือข่ายอินเตอร์เน็ต

รูปนี้แสดงการติดต่อของวินโดว์แอพพลิเคชันสองแอพพลิเคชันคืออินเตอร์เน็ตไคลเอนต์กับอินเตอร์เน็ตเซิร์ฟเวอร์ ซึ่งทั้งสองติดต่อผ่านเครือข่ายอินเตอร์เน็ตทางวินโดว์ซ้อกเก็ต เป็นที่แน่นอนว่าโปรแกรมที่เราติดต่ออยู่อาจไม่เป็นวินโดว์แอพพลิเคชัน ซ้อกเก็ตสร้างกระบวนการติดต่อแบบเดียวกันในหลายระบบปฏิบัติการ

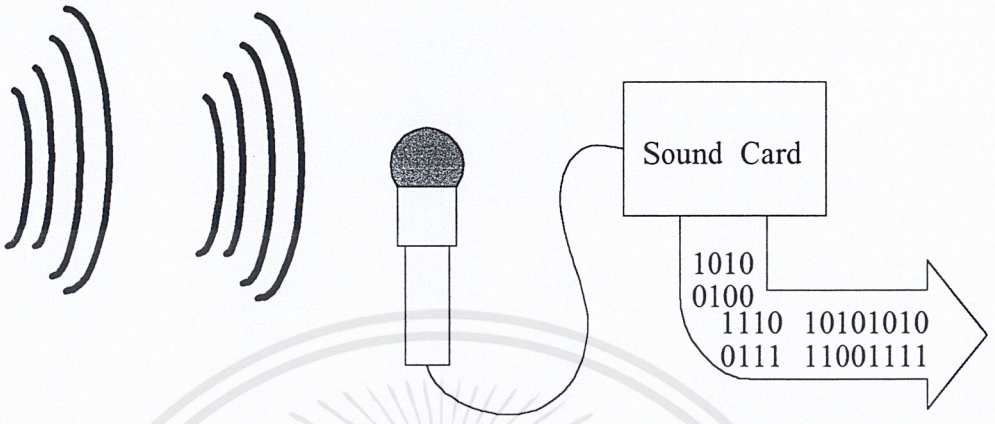
2.8 ข้อมูลการ์ดเสียง (Sound Card)

ต้นตำรับของเสียงสังเคราะห์เกิดจากนายโรเบิร์ต มูก เป็นผู้ริเริ่มนำเอาเสียงสังเคราะห์มาใช้โดยการรวมคีย์บอร์ดเข้ากับวงจรอิเล็กทรอนิกส์ เพื่อให้ได้เสียงที่แตกต่างจากออร์แกนอิเล็กทรอนิกส์ นายโรเบิร์ต ใช้วงจรออสซิลเลเตอร์เป็นวงจรกรองสัญญาณ และวงจรขยายสำหรับสังเคราะห์เสียงดนตรีต่างๆ เพิ่มเติมนอกเหนือจากเปียโน และออร์แกน

คุณสมบัติของเสียงโดยทั่วไป มักเป็นสัญญาณอนาล็อก ที่มีค่าขนาดของสัญญาณต่อเนื่องกันไป โดยสามารถแทนค่าขนาดของสัญญาณด้วยแรงดันไฟฟ้า และความถี่เสียงด้วยความดังมากกว่าสัญญาณที่มีความถี่ สูงกว่า แต่ถ้าเป็นสัญญาณดิจิตอล จะเก็บและประมวลผลไว้ในลักษณะของตัวเลขฐาน 2 คือเลข 0 และ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นการเก็บเสียงในคอมพิวเตอร์ จึงต้องแปลงสัญญาณเสียง ให้เป็นสัญญาณดิจิตอลก่อนเพื่อนำไปเก็บไว้ในหน่วยความจำของคอมพิวเตอร์ เมื่อต้องการรับฟังเสียงที่บันทึกไว้ก็จะนำข้อมูลดิจิตอลมาแปลงกลับให้เป็นสัญญาณอนาล็อกเสียก่อน (รูปที่ 2.19)



รูปที่ 2.19 กระบวนการแปลงสัญญาณของการ์ดเสียง

กระบวนการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล ในการ์ดเสียงจะเป็นหน้าที่ของชิพ ADC โดยจะทำหน้าที่วัดขนาดของสัญญาณที่คลื่นเสียงตามระยะเวลาที่คงที่ เรียกว่า อัตราแซมปิ้ง (Sampling Rate) และเรียกเวลาในการแปลงขนาดของสัญญาณเป็นข้อมูลตัวเลข เพื่อนำไปประมวลผลของชิพ ADC ว่าคาบเวลาของการแซมปลิ้ง (Sampling Time) ค่าตัวเลขที่ได้เป็นเลขฐานสิบ ส่วนเอาท์พุทจริงจากชิพ ADC เป็นเลขฐาน 2

ดังนั้นจะต้องมีกระบวนการแปลงเลขฐานสิบเป็นเลขฐานสอง ค่าที่เก็บจะเริ่มจากเลขศูนย์ตามด้วยเลขฐานสองต่างๆ กรรมวิธีนี้เป็นกระบวนการมอดูเลชั่นแบบหนึ่งเรียกว่า พัลส์โค้ดมอดูเลชั่น (PCM: Pulse Code Modulation)

ความแม่นยำในการวัดขนาดของสัญญาณอนาล็อกกำหนดได้ โดยจำนวนบิตของเลขฐานสอง ในระบบดิจิตอลของชิพ ADC ถ้าหากใช้ชิพขนาด 8 บิต จะได้ขนาดของสัญญาณที่วัดได้ช่วงละ 256 ค่า และถ้าเพิ่มเป็น 12 บิตจะได้รายละเอียดเพิ่มเป็น 4096 ค่า และเมื่อเพิ่มขึ้นเป็น 16 บิต จะมีรายละเอียดเพิ่มเป็น 65536 ค่า และยิ่งเพิ่มความละเอียดมากขึ้นก็ต้องใช้เนื้อที่ของหน่วยความจำมากขึ้นและการแปลงสัญญาณข้อมูลดิจิตอลกลับมาเป็นสัญญาณอนาล็อก (Digital to Analog Converter: DAC) โดยใช้ความเร็วในการแปลงกลับเท่ากับการแซมปลิ้ง เอาท์พุทที่ได้จะมีลักษณะเป็นขั้นบันไดต่อเนื่องกันไป จากนั้นต้องผ่านกระบวนการแปลงคลื่นขั้นบันไดให้เป็นคลื่นปกติ โดยการใช้วงจรกรองความถี่ชิพ ADC และ DACใช้งานควบคู่กันเพื่อการบันทึก และสังเคราะห์เสียงออกมา เป็นผลให้ระดับเสียงแบบสเตอริโอ ใช้หน่วยความจำที่บรรจุข้อมูล เป็นสองเท่าของระบบโมโน โดยปกติแล้วอัตราการแซมปลิ้งสัญญาณที่เหมาะกับการ์ดเสียงที่ติดตั้งชิพ ADC และ DAC จะใช้ค่า 5.0125 และ 11.025 KHz ทำให้คุณภาพเสียงเท่ากับเสียงจากโทรศัพท์ ส่วนการแซมปลิ้ง 16 บิตเป็นขนาดมาตรฐานในวงการค้าของเครื่องเล่นซีดี และงานบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียงระบบดิจิตอลออดิโอเทป (Digital Audio Tape: DAT) เครื่องเล่นซีดีมีอัตราการแซมปลิ้งที่44.1 KHz และ DAT จะใช้ 48 KHz การ์ดอะแดปเตอร์เสียงส่วนใหญ่ใช้ 8 บิต มีอัตราส่วนการแซมปลิ้งที่ 22.05 KHz

2.8.1 การสังเคราะห์เสียง

การสังเคราะห์เสียงดิจิตอลชนิดเอฟเอ็ม สามารถเลียนเสียงต่างได้มากมาย แต่ไม่ทั้งหมด เสียงที่มีฮาร์โมนิค เปลี่ยนแปลงรวดเร็วจะไม่สามารถทำการสังเคราะห์เสียงได้ แต่บันทึกเสียงเหล่านั้นไว้แล้วแปลงเป็นข้อมูลดิจิตอลจากนั้นมาเล่นกลับ ภายหลังจากการสังเคราะห์เสียงต่างๆด้วยการบันทึกลงซีดีทำได้โดยนำเสียงดนตรีแต่ละชิ้นมาทำการแซมปลิ้งเสียงลงแผนซีดี 16 บิต และเก็บไว้ด้วยความเร็ว 44.1 KHz โดยการปรับเปลี่ยนอัตราการแซมปลิ้งทำให้พตช์เปลี่ยน และสร้างโน๊ตเสียงใหม่ๆได้อีกมากมาย หรือถ้าเปลี่ยนอัตราความเร็วในการเล่นเหลือเพียงครึ่งหนึ่ง หรือเพิ่มความเร็วเป็นสองเท่าก็ได้ซึ่งเป็นผลให้เพิ่มหรือลดระดับเสียงสูงจนถึงต่ำสุดของเสียงอีกหนึ่งอ๊อคเตป (Octave)

2.8.2 คุณสมบัติของการ์ดเสียง

การ์ดเสียงโดยทั่วไปมีข้อกำหนดไว้ 5 ข้อคือ

1. สามารถเล่นไป-กลับ โดยใช้ข้อมูลที่เก็บไว้ในรูปพัลส์โค้ดมอดูเลชั่น
2. สามารถสังเคราะห์เสียงได้ 128 เสียง
3. ส่วนมิกเซอร์ (Mixer) เสียงจะทำหน้าที่ควบคุมแหล่งจ่ายเสียง และระดับสัญญาณเสียงที่ลำโพง ในบางการ์ดอาจมีวงจรควบคุมเสียงทุ่ม และเสียงแหลมเพิ่มเติมด้วย
4. การบันทึกเสียงจะบันทึกเสียงดนตรี หรือเสียงต่างๆ ในรูปของพัลส์โค้ดมอดูเลชั่นอาจบันทึกเป็นแบบสเตอริโอ หรือโมโน และสามารถควบคุมอัตราการแซมปลิ้งที่เหมาะสมกับความคมชัดของเสียงได้
5. ต้องมีอินพุทและเอาท์พุทเพื่อให้สามารถใช้การ์ดเสียงกับเครื่องดนตรีมิดี้ภายนอกได้ปกติ การ์ดจะมีแอดเดรสสำหรับต่อกับจอยสติ๊ก, แอดเดรสสำหรับต่อกับซีดีรอมไดร์ฟ การที่มีแอดเดรสจอยสติ๊กช่วยให้แน่ใจว่าสามารถต่อกับคอมพิวเตอร์ที่คอมแพตติเบิลกันได้ด้วย แอดเดรสอินพุท และเอาท์พุทของ 8086 มักมีคอนเนคเตอร์หลายขาที่เรียกว่า เฮดเดอร์ (Header) เพื่ออินเตอร์เฟสกับซีดีรอมได้ บางทีก็แบ่งการ์ดเล็กต่างหากแล้วมีขาสำหรับเสียงต่อกับเมนบอร์ด

2.9 ความเป็นมาของเทคโนโลยีมัลติมีเดียใน Internet

ในยุคของอินเตอร์เน็ตเมื่อคอมพิวเตอร์ติดต่อสื่อสารกันได้ทั่วโลก การนำมัลติมีเดียมาใช้รับส่งข้อมูลข้อมูลก็เป็นสิ่งที่ทุกคนต้องการ เนื่องจากตัวเครื่องคอมพิวเตอร์นั้นสามารถใช้งานข้อมูลมัลติมีเดียได้อยู่แล้ว แต่ว่าเครือข่ายของอินเตอร์เน็ตนั้นไม่ได้ถูกออกแบบมาให้รองรับการส่งข้อมูลในรูปแบบของมัลติมีเดีย ทำให้โลกของมัลติมีเดียกับอินเตอร์เน็ตยังไปด้วยกันไม่ได้ ซึ่งข้อจำกัดของการใช้งานมัลติมีเดีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บนอินเตอร์เน็ตก็คือ ความเร็วในการรับส่งข้อมูลผ่านอินเตอร์เน็ต และโปรโตคอลที่จะใช้รับส่งข้อมูลมัลติมีเดียนั่นเอง

ความเร็วในการรับส่งข้อมูลระหว่างคอมพิวเตอร์ของเราจะถูกจำกัดอยู่ที่ความเร็วของโมเด็มที่ใช้ ซึ่งในปัจจุบันโมเด็มได้รับการพัฒนามาจนมีความเร็วสูงสุดอยู่ที่ 56 Kbps หรือสามารถรับส่งข้อมูลได้เกือบ 56,000 บิตต่อวินาที และอีกมาตรฐานหนึ่งที่เราใช้กันมากจะสามารถรับส่งข้อมูลได้ 33.6 Kbps หรือ 33,600 บิตต่อวินาที แม้จะมีความเร็วสูงกว่าโมเด็มในสมัยก่อนมาก แต่เมื่อเราต้องการรับส่งข้อมูลทั้งภาพเคลื่อนไหวและเสียงผ่านอินเตอร์เน็ต ความเร็วของโมเด็มที่ใช้กันอยู่นับว่าช้ามาก เนื่องจากข้อมูลมัลติมีเดียที่มีทั้งภาพเคลื่อนไหวและเสียงนั้น ต้องการความเร็วในการรับส่งข้อมูลอย่างน้อย 50 ล้านบิตต่อวินาที จึงจะแสดงภาพเคลื่อนไหวเต็มจอพร้อมกับเสียง โดยมีคุณภาพใกล้เคียงกับวิดีโอเทปได้ แม้ว่าเราจะมีเทคนิคการบีบอัดอย่าง MPEG เพื่อลดขนาดข้อมูลของภาพเคลื่อนไหวและเสียงลงได้ 10 เท่าแล้วก็ตาม เราก็ยังต้องการความเร็วในการรับส่งข้อมูลถึง 5 ล้านบิตต่อวินาที ซึ่งยังเร็วกว่าที่โมเด็มในปัจจุบันทำได้ถึงเกือบ 100 เท่าทีเดียว

เมื่อความเร็วในการรับส่งข้อมูลของโมเด็มยังเป็นตัวจำกัดการรับส่งข้อมูลผ่านอินเตอร์เน็ตอยู่ เราจึงมีทางเลือกในการรับส่งข้อมูลมัลติมีเดียได้ 2 ทางคือ ลดคุณภาพของภาพและเสียงของข้อมูลมัลติมีเดียลง เพื่อให้รับส่งผ่านความเร็วของโมเด็มได้ ซึ่งวิธีนี้เราก็พอจะเห็นการใช้งานบนเว็ปไซต์ต่างๆอยู่ในปัจจุบัน คือขนาดของภาพจะเล็กลงเพื่อให้ข้อมูลของภาพลดลงเมื่อเทียบกับภาพเต็มจอ พร้อมๆกับการแสดงผลจำนวนภาพต่อวินาทีก็ลดลง จาก 25 ภาพต่อวินาที ก็ลดลงเหลือ 20 หรือ 10 ภาพต่อวินาที เป็นต้น เมื่อบีบอัดข้อมูลแล้วก็พอที่จะส่งข้อมูลมัลติมีเดียไปแสดงผลได้ด้วยความเร็ว 33.6 Kbps หรือ 56 Kbps แต่อย่างไรก็ตามการรับส่งข้อมูลมัลติมีเดียด้วยการลดคุณภาพ และขนาดของภาพลงนี้ แม้ว่าจะทำให้การใช้งานอินเตอร์เน็ตสามารถแสดงข้อมูลมัลติมีเดียได้ แต่คุณภาพก็นับว่ายังห่างไกลจากข้อมูลมัลติมีเดียที่เราได้จาก CD-ROM , Video-CD หรือ DVD มาก

อีกวิธีหนึ่งก็คือใช้การรับส่งข้อมูลเร็วขึ้น โดยใช้อุปกรณ์รับส่งข้อมูลชนิดอื่นที่ไม่ใช่โมเด็ม เช่น ใช้บริการ ISDN (Integrated Services Digital Network) ที่สามารถรับส่งข้อมูลได้ 128 Kbps ไปจนถึง 2 Mbps หรือ 2 ล้านบิตต่อวินาที หรือใช้บริการ ADSL (Asymmetric Digital Subscriber Line) ที่สามารถรับส่งข้อมูลได้เร็วถึง 5 ล้านบิตต่อวินาที ซึ่งอุปกรณ์รับส่งข้อมูลความเร็วสูงเหล่านี้กำลังเผยแพร่ออกสู่ตลาด โดยจะเข้ามาทดแทนโมเด็มที่เราใช้อยู่ในปัจจุบัน คาดว่าภายในไม่กี่ปีข้างหน้าการเชื่อมต่อรับส่งข้อมูลระหว่างคอมพิวเตอร์ของเรากับอินเตอร์เน็ตจะมีความเร็วในระดับ 2 Mbps หรือ 2 ล้านบิตต่อวินาทีเป็นมาตรฐาน ส่วนความเร็วที่ต่ำกว่านี้จะเป็นการใช้โมเด็มรับส่งข้อมูลอย่างในปัจจุบัน แต่จะเป็นการใช้งานผ่านอุปกรณ์ไร้สาย เช่น โทรศัพท์เคลื่อนที่ หรือใช้กับสถานที่ที่อยู่ห่างไกลออกไป ซึ่งเครือข่ายความเร็วสูงยังไม่มีให้บริการในพื้นที่นั้นๆ เท่านั้น

เนื่องจากข้อมูลมัลติมีเดียต้องใช้ความเร็วในการรับส่งที่สูงมาก เมื่อเรายังไปไม่ถึงจุดที่ความเร็วในการรับส่งข้อมูลระดับ 2 Mbps การใช้งานมัลติมีเดียผ่านอินเตอร์เน็ตของเราจึงยังจำกัดอยู่มาก คือรับส่งข้อมูลได้ภาพเคลื่อนไหวขนาดเล็ก และภาพเคลื่อนไหวไม่สม่ำเสมอนัก ทั้งยังต้องใช้เวลาในการรับส่งข้อมูลนานมาก เช่น การรับข้อมูลภาพเคลื่อนไหวเพียงไม่กี่นาทีจะต้องใช้เวลานานนับเป็นชั่วโมง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากความเร็วในการรับส่งข้อมูลจะเป็นตัวจำกัดการใช้งานมัลติมีเดียในอินเตอร์เน็ตแล้ว โปรโตคอลที่ใช้รับส่งข้อมูลของอินเตอร์เน็ต ยังไม่เหมาะกับการใช้งานในลักษณะกระจายข้อมูลมัลติมีเดียอีกด้วย คือโปรโตคอลต่างๆที่ใช้รับส่งข้อมูลผ่านอินเตอร์เน็ตนั้น มีจุดประสงค์ที่จะให้รับส่งข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่งได้อย่างถูกต้องเท่านั้น ในกรณีที่ต้องการส่งข้อมูลมัลติมีเดียผ่านไปให้ผู้รับหลายๆคนพร้อมกัน เราจะต้องส่งข้อมูลไปให้ผู้รับโดยแยกเป็นคนละชุดกัน เพราะข้อมูลมัลติมีเดียที่มีปริมาณข้อมูลมากอยู่แล้ว จะยิ่งทวีคูณขึ้นไปอีกทำให้การรับส่งข้อมูลผ่านอินเตอร์เน็ตช้าลงกว่าเดิมและเป็นการรับส่งข้อมูลซ้ำซ้อนโดยไม่จำเป็น เนื่องจากโปรโตคอลที่ใช้รับส่งข้อมูลในปัจจุบัน ยังไม่สามารถรับส่งข้อมูลในลักษณะกระจายจากจุดหนึ่งไปยังผู้รับหลายๆจุดพร้อมกันได้ ซึ่งการใช้โปรโตคอลในการรับส่งข้อมูลในแบบกระจาย (multicast) และรับส่งข้อมูลได้อย่างต่อเนื่อง (streaming) จะทำให้การรับส่งข้อมูลมัลติมีเดียบนอินเตอร์เน็ตได้ผลดีกว่าการใช้โปรโตคอลแบบเดิมที่ใช้กันอยู่

เพื่อแก้ไขปัญหาโปรโตคอลในการรับส่งข้อมูลมัลติมีเดียผ่านอินเตอร์เน็ต เราจำเป็นต้องนำโปรโตคอลอีกชนิดหนึ่งมาใช้รับส่งข้อมูลแทน Internet Protocol แบบเดิมที่มีการรับส่งข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่งเท่านั้น โดยการออกแบบโปรโตคอลใหม่ที่มีลักษณะการรับส่งข้อมูลจากจุดหนึ่งไปยังผู้รับหลายๆจุดได้เรียกว่า Multicast Protocol ซึ่งเหมือนกันการกระจายเสียงของสถานีวิทยุหรือสถานีโทรทัศน์นั่นเอง แต่ Multicast Protocol ก็เป็นเพียงการรับส่งข้อมูลจากจุดหนึ่งไปให้ผู้รับหลายๆจุดเท่านั้นยังไม่ใช่การแก้ปัญหาการรับส่งข้อมูลอย่างต่อเนื่องตามที่ข้อมูลมัลติมีเดียต้องการ จึงๆได้มีการพัฒนาโปรโตคอลรับส่งข้อมูลอีกชนิดหนึ่งขึ้นมา เรียกว่า Streaming Protocol ทำให้การรับส่งข้อมูลมัลติมีเดียดีขึ้นมาก เมื่อมาถึงจุดนี้ก็เหลือเพียงแต่รอให้ความเร็วในการรับส่งข้อมูลระหว่างคอมพิวเตอร์ของเรากับอินเตอร์เน็ตเร็วขึ้นสู่ระดับ 2 Mbps เท่านั้น การใช้งานมัลติมีเดียบนอินเตอร์เน็ตก็จะมีคุณภาพเท่าเทียมกับการอ่านข้อมูลจาก CD-ROM , Video-CD หรือ DVD เลยทีเดียว ซึ่งโปรโตคอลในแบบ Multicast และ Streaming ที่ใช้อยู่ในปัจจุบันนั้นมีรายละเอียดดังต่อไปนี้

2.9.1 Multicast Protocol

Multicast Protocol ได้รับการพัฒนาขึ้นมาเพื่อใช้รับส่งข้อมูลผ่านอินเตอร์เน็ต ในลักษณะการกระจายข้อมูลจากจุดหนึ่งไปยังผู้รับหลายๆจุด หรือที่เรียกว่า Broadcast โดยเริ่มใช้ตั้งแต่เดือนพฤษภาคม ปี 1993 เรียกว่า RFC 1458 มีเป้าหมายหลักคือ การรับส่งข้อมูลภาพผ่านอินเตอร์เน็ตไปยังผู้รับหลายๆคน ทั้งในลักษณะการรับส่งแบบต่อเนื่องตลอดเวลาและแบบไม่ต่อเนื่อง เพื่อกระจายข้อมูลภาพให้ผู้รับนำไปใช้วิเคราะห์ข้อมูล , แสดงภาพประกอบรายงาน และใช้ในกิจกรรมการประชุมทางโทรภาพ (video conference) ซึ่งโปรโตคอลที่ใช้รับส่งข้อมูลภาพในแบบกระจายนี้ จะช่วยแก้ปัญหาในการรับส่งข้อมูลแบบจุดต่อจุดของโปรโตคอลแบบเก่าได้ และมีกลไกในการควบคุมให้สามารถรับส่งข้อมูลแบบต่อเนื่องได้ดียิ่งขึ้น

การรับส่งข้อมูล multicast จะใช้โปรโตคอล IP/Multicast เป็นส่วนขยายของ IP โปรโตคอลปกติ ซึ่งโปรโตคอลนี้จะไม่มีกลไกในการกำหนด address ของผู้รับ เครื่องคอมพิวเตอร์ที่ต้องการรับข้อมูลจะส่งคำขอรับข้อมูล multicast โดยใส่ IP ของเครื่องผู้รับลงในกลุ่มที่จะรับข้อมูล multicast นั้นๆ เมื่อข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

multicast ถูกส่งออกมาถึงเครือข่ายผู้รับ ก็จะได้รับข้อมูลเหมือน IP โปรโตคอลปกตินั่นเอง การรับข้อมูล multicast จะถูกควบคุมโดย Router ในเครือข่ายเป็นหลัก เมื่อคอมพิวเตอร์นั้นๆไม่ต้องการรับข้อมูล multicast อีกต่อไป ก็จะส่งคำสั่งลบ IP ของตัวเองออกจากกลุ่มที่จะรับข้อมูล multicast นั้น กลไกการขอรับข้อมูลและยกเลิกการรับข้อมูล multicast นี้จะเกิดขึ้นตลอดเวลาในระบบเครือข่าย ซึ่งจะกลไกที่เกี่ยวข้องอีกหลายส่วนประกอบคือ Multicast Group Address Assignment , Group Setup , Membership Maintenance , Error Recovery และ Flow Control โดยปกติเครือข่ายจะมีการตรวจสอบกลุ่มผู้รับเป็นระยะๆ ว่ามีใครเพิ่มขึ้นมาหรือมีใครถูกลบออกจากกลุ่มบ้าง ซึ่งอาจจะตรวจทุกหนึ่งนาที สองนาที หรือตามเวลาที่ตั้งไว้

2.9.2 MBONE

MBONE เป็นเครือข่ายที่ซ้อนอยู่บนอินเตอร์เน็ตในลักษณะของ Virtual Private Network หรือ VPN เพื่อใช้เครือข่ายของ MBONE (Multicast Backbone) ในการรับส่งข้อมูลแบบกระจาย หรือ multicast ผ่านอินเตอร์เน็ต เนื่องจากโดยทั่วไปแล้วเครือข่ายเกือบทั้งหมดของอินเตอร์เน็ต ยังไม่สามารถรับส่งข้อมูลจากจุดหนึ่งไปให้ผู้รับหลายๆจุดในแบบ multicast ได้ เพียงแต่สามารถรับส่งข้อมูลจากผู้ส่งไปถึงผู้รับในลักษณะจุดต่อจุดที่เรียกว่า unicast เท่านั้น เมื่อเราต้องการรับส่งข้อมูลในลักษณะกระจาย หรือ multicast ผ่านอินเตอร์เน็ต อุปกรณ์เครือข่ายที่เชื่อมต่ออยู่ เช่น Router จะไม่สามารถรับข้อมูล multicast มากระจายส่งให้ผู้รับได้ เนื่องจาก Router ส่วนใหญ่จะเป็นการรับส่งข้อมูลแบบจุดต่อจุด คือต้องมี address ของผู้รับจึงจะสามารถส่งข้อมูลนั้นต่อไปให้ผู้รับได้อย่างถูกต้อง แต่ข้อมูลที่เป็น multicast จะไม่มีการระบุ address ของผู้รับ คือข้อมูลจะถูกส่งไปตามเครือข่าย ใครต้องการรับข้อมูลนั้นก็จะดึงข้อมูลที่ถูกส่งผ่านมาเก็บไปใช้งานเอาเอง โดยไม่มีการเจาะจง address ของผู้รับ ดังนั้น MBONE จึงเข้ามาทำหน้าที่ในการกระจายข้อมูล multicast นี้เอง โดยทำตัวเป็นเครือข่ายซ้อนอยู่บนอินเตอร์เน็ต เมื่อได้รับข้อมูล multicast มาก็จะกระจายข้อมูลนั้นไปให้ผู้รับตามต้องการผ่านเครือข่ายของอินเตอร์เน็ตโดยจะใช้เทคนิคที่เรียกว่า Tunneling ผนึกข้อมูล multicast นั้น แล้วแปลงให้อยู่ในรูปข้อมูล IP ปกติ ส่งไปให้ผู้รับทุกคนที่ต้องการ

จากหลักการการทำงานของ MBONE จะเห็นว่าเมื่อมาตรฐานของโปรโตคอล IP/Multicast มีการใช้งานอย่างแพร่หลาย การใช้ MBONE เพื่อผนึกข้อมูล multicast ส่งผ่านอินเตอร์เน็ตก็จะหมดความจำเป็นลงไป เนื่องจากเราสามารถรับส่งข้อมูลในแบบ multicast ผ่านอินเตอร์เน็ตได้โดยตรง แต่ MBONE ก็คงจะไม่หายไปจากการใช้งานของเรา เนื่องจาก MBONE ก็จะใช้เครือข่ายที่มีอยู่ให้บริการข้อมูล multicast ต่อไป เพียงแต่ปรับไปใช้ IP/Multicast แทนการใช้ Tunneling ในการรับส่งข้อมูลเท่านั้น

MBONE เกิดขึ้นในปี ค.ศ. 1992 โดยทีมของ Internet Engineering Task Force (IETF) ได้เสนอแนวคิดในการใช้ซอฟแวร์รับส่งข้อมูล multicastผ่านเครือข่ายอินเตอร์เน็ตโดยใช้ Virtual Private Network เพื่อให้ข้อมูลสามารถส่งไปยังผู้รับหลายๆคนได้ โดยไม่จำเป็นต้องรอ IP/Multicast โปรโตคอล และไม่ต้องปรับปรุงแก้ไขฮาร์ดแวร์ที่มีใช้อยู่ในเครือข่ายอินเตอร์เน็ต ก็คือการใช้ซอฟต์แวร์เข้าช่วยนั่นเอง ซึ่งเครือข่ายอินเตอร์เน็ต , อุปกรณ์เครือข่าย , Router และอื่นๆก็ยังคงใช้ของเดิมที่มีอยู่ เมื่อใช้ซอฟต์แวร์ที่กำหนดแล้ว เครือข่าย MBONE ก็จะสามารถส่งข้อมูลให้ผู้รับจำนาน 2 คน , 10 คน หรือเป็น 100 คนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมือนการกระจายข้อมูล โดยใช้เทคนิคการผนึกข้อมูล multicast ส่งผ่าน Tunneling ไปในเครือข่ายที่เป็น unicast ของอินเตอร์เน็ตได้อย่างไม่มีปัญหา

เมื่อข้อมูล multicast ที่ถูกผนึกอยู่ ถูกส่งไปยังอุปกรณ์รุ่นใหม่ที่สามารถรองรับการกระจายข้อมูล multicast ได้ เช่น multicast router ข้อมูลก็จะถูกกระจายส่งไปให้ผู้รับตามปกติ ซึ่งอุปกรณ์ที่สามารถรับส่งข้อมูล multicast ได้นี้ก็เริ่มออกสู่ตลาดเพิ่มมากขึ้นเรื่อยๆ ดังนั้นความแตกต่างของ MBONE กับ IP/Multicast ก็คือ MBONE เป็นการใช้งาน multicast ชั่วคราวในการรับส่งข้อมูลแบบกระจายผ่านอินเตอร์เน็ตโดยใช้ซอฟต์แวร์เข้าช่วย ซึ่งจะหมดความจำเป็นลงเมื่อ IP/Multicast ออกใช้งานอย่างสมบูรณ์นั่นเอง ซึ่ง IP/Multicast จะเป็นกลไกสำคัญในการรับส่งกระจายข้อมูลภาพ และเสียงในอินเตอร์เน็ต สำหรับการใช้งานมัลติมีเดียที่จะเปลี่ยนโฉมหน้าการใช้อินเตอร์เน็ตให้สมบูรณ์แบบยิ่งขึ้น ซึ่ง MBONE นับเป็นเครือข่ายที่ใช้งานได้จริง ๆ ของการรับส่งข้อมูลในแบบกระจาย

ปัจจุบัน MBONE มีเครือข่ายมากกว่า 1,700 เครือข่าย ครอบคลุม 20 ประเทศทั่งโลก ซึ่งแม้ว่าเครือข่ายของ MBONE จะไม่กว้างขวางครอบคลุมเหมือนอินเตอร์เน็ต แต่ก็เป็นเครือข่ายขนาดใหญ่ที่สามารถรองรับการส่งข้อมูลมัลติมีเดียแบบกระจายไปยังปลายทางหลายๆ แห่งได้ และมีขนาดเทียบเท่ากับเครือข่ายของอินเตอร์เน็ตทั้งหมดที่มีอยู่ในปี ค.ศ. 1990 เลยทีเดียว

และเนื่องจากข้อมูลประเภท multicast ถูกกระจายไปทั่วทั้งเครือข่าย เพื่อให้ผู้รับได้ข้อมูลอย่างทั่วถึงนั้น ถ้าหากไม่มีการควบคุมข้อมูลนี้เลย ก็จะพบว่าข้อมูลเหล่านี้ถูกส่งซ้ำไปซ้ำมาอยู่ในเครือข่ายตลอดเวลา ซึ่งนอกจากจะไม่มีประโยชน์แล้วยังทำให้เครือข่ายหนาแน่นไปด้วยข้อมูลอีกด้วย MBONE จึงกำหนดอายุของข้อมูลที่จะกระจายไปในเครือข่ายโดยใช้ตัวนับ (counter) ที่เรียกว่า Time to Live (TTL) เมื่อข้อมูลถูกรับส่งผ่านเครือข่าย ตัวนับ TTL ก็จะถูกลดจำนวนลงไปทีละหนึ่ง เมื่อค่าของ TTL เป็น 0 ข้อมูลนั้นก็จะไม่ถูกส่งกระจายไปยังผู้อื่นอีก ซึ่งค่า TTL นี้สามารถกำหนดได้ โดยหากต้องการกระจายข้อมูลให้ทั่วถึงอาจตั้งค่า TTL สูงถึง 200 แต่ถ้าเป็นข้อมูลที่ใช้ในกลุ่มย่อยๆอาจตั้งค่า TTL เอาไว้เพียง 10 เท่านั้น TTL จึงเป็นส่วนควบคุมข้อมูล multicast ใน MBONE ไม่ให้ถูกส่งกระจายอย่างไม่มีขีดจำกัด และป้องกันเครือข่ายเต็มอีกด้วย

การใช้งาน MBONE จะคล้ายกับการใช้งานมัลติมีเดียอื่นๆ คือจะสามารถประยุกต์ใช้ในการประชุมผ่านอินเตอร์เน็ต (video conference), เขียนข้อความลงบนกระดานที่ผู้ร่วมประชุมสามารถมองเห็นได้ทุกคนจากจอภาพของแต่ละคน , สามารถกระจายข้อมูลภาพและเสียงผ่านอินเตอร์เน็ตให้ผู้รับจำนวนมาก ๆ , สามารถกระจายเสียงผ่านทางอินเตอร์เน็ตเหมือนกับสถานีวิทยุ , ไปจนถึงการถ่ายทอดเหตุการณ์ต่างๆ เช่น ถ่ายทอดสดการแข่งขันฟุตบอลให้ผู้รับผ่านทางอินเตอร์เน็ตได้ เหล่านี้เป็นต้น

อย่างไรก็ตาม ปัจจุบันเทคโนโลยีการใช้มัลติมีเดียมีการนำ Streaming Protocol มาใช้รับส่งข้อมูล ซึ่งมีหลักการทำงานแตกต่างกันจาก MBONE และ IP/Multicast คือ Streaming Protocol จะใช้เซิร์ฟเวอร์ที่ตั้งขึ้นมาโดยเฉพาะทำการรับส่งข้อมูลในลักษณะต่อเนื่องจากผู้ส่งไปให้ผู้รับโดยตรง ผู้ใช้งานจะไม่รู้สึกถึงความแตกต่างระหว่างการใช้ MBONE กับ Streaming Protocol แต่ข้อมูลที่วิ่งไปมาในเครือข่ายอินเตอร์เน็ตจะน้อยกว่าการใช้ MBONE เนื่องจาก Streaming Protocol จะรับส่งข้อมูลจากผู้ส่งไปให้ผู้รับเท่านั้น ข้อมูลจึงไม่กระจายไปทั่วทั้งเครือข่าย และการส่งข้อมูลอย่างต่อเนื่องพร้อมกับแสดงผลข้อมูลทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

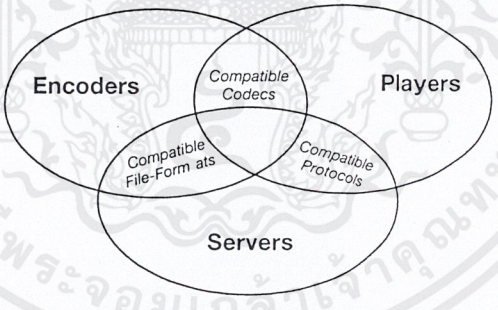
ทีโดยไม่ต้องรอให้รับข้อมูลทั้งหมดจบเสียก่อน ทำให้เราสามารถรับส่งภาพและเสียงในแบบทันทีทันใด โดยไม่ต้องรอนาน หรือเปลืองเนื้อที่บนฮาร์ดดิสก์ (Hard disk) ในการเก็บข้อมูลดังกล่าว ซึ่งเทคโนโลยีของ Streaming Protocol ก็เป็นอีกวิธีหนึ่งให้เราใช้งานมัลติมีเดียผ่านอินเตอร์เน็ตได้อย่างมีประสิทธิภาพ

2.9.3 Real-Time Streaming Protocol (RTSP)

Real - Time Streaming Protocol หรือ RTSP เป็นโปรโตคอลที่ใช้รับส่งข้อมูลมัลติมีเดียระหว่างเซิร์ฟเวอร์กับคอมพิวเตอร์ปลายทาง ซึ่งจะทำการรับส่งข้อมูลอย่างต่อเนื่องผ่านอินเตอร์เน็ต โดยตัวเซิร์ฟเวอร์ด้านผู้ส่งสามารถส่งข้อมูลไปให้ผู้รับปลายทางเพียงคนเดียว หรือจะส่งให้ผู้รับหลายๆคนในลักษณะเป็นกลุ่มก็ได้ ซึ่ง RTSP ถูกกำหนดให้เป็นโปรโตคอลที่นำไปใช้ในอินเตอร์เน็ตโดย Internet Engineering Task Force (IETF) ในเดือนเมษายน ปี ค.ศ. 1998 เรียกว่า RFC 2326

RTSPเป็นการกำหนดมาตรฐานโปรโตคอลที่สำคัญมากในการรับส่งข้อมูลมัลติมีเดียผ่านอินเตอร์เน็ต เนื่องจากการรับส่งข้อมูลมัลติมีเดียในแบบต่อเนื่องนั้นจะมีส่วนต่างๆที่เกี่ยวโยงกันถึง 3 ส่วน คือ เซิร์ฟเวอร์ที่เก็บข้อมูล , Encoder ที่ใช้เข้ารหัสข้อมูล และผู้รับข้อมูล หรือ Player ดังในรูป

ตัวเข้ารหัสข้อมูลหรือ Encoder นั้นจะต้องเข้ารหัสข้อมูลมัลติมีเดียเก็บลงในไฟล์ โดยมีฟอร์แมตที่เซิร์ฟเวอร์เรียกใช้งานได้ และเมื่อเซิร์ฟเวอร์ต้องการส่งข้อมูลนี้ไปให้ผู้รับ ก็จะต้องใช้โปรโตคอลรับส่งข้อมูลอย่างต่อเนื่องที่ผู้รับเข้าใจ และสามารถรับข้อมูลได้อย่างถูกต้อง จากนั้นเมื่อได้รับข้อมูลมาแล้วก็จะต้องถอดรหัสข้อมูลออกมาแสดงผลได้ โดยใช้มาตรฐานเดียวกับตัวเข้ารหัส การทำงานทั้งหมดจึงผูกกันตามที่แสดงในรูป ซึ่ง RTSP จะอยู่ในส่วนของโปรโตคอลที่ใช้รับส่งข้อมูลจากเซิร์ฟเวอร์ไปให้ผู้รับนั่นเอง



รูปที่ 2.20 แสดงความสัมพันธ์ของข้อมูลมัลติมีเดีย

แม้ว่า RTSP จะมีความสำคัญในการรับส่งข้อมูลมัลติมีเดียผ่านอินเตอร์เน็ตก็ตาม แต่ก็ไม่ใช่สิ่งเดียวที่ทำให้การรับส่งข้อมูลสมบูรณ์ได้เรายังต้องการฟอร์แมตมาตรฐานของไฟล์ที่ใช้เก็บข้อมูลมัลติมีเดียอีกด้วย เช่น Active Streaming Format (ASF) ของไมโครซอฟท์ , QuickTime หรืออื่นๆเพื่อเก็บข้อมูล รวมถึงมาตรฐานการเข้ารหัสข้อมูล เช่น MPEG สำหรับใช้เข้ารหัสข้อมูลมัลติมีเดียเก็บลงไฟล์อีกด้วย

RTSP จะทำหน้าที่ควบคุมการรับส่งข้อมูลมัลติมีเดียอย่างต่อเนื่องระหว่างเซิร์ฟเวอร์ที่เก็บข้อมูลกับคอมพิวเตอร์ผู้รับข้อมูล โดยมีไฟล์ฟอร์แมตสำหรับเก็บข้อมูล เช่น ASF , QuickTime หรืออื่นๆ ซึ่งการสร้างไฟล์ข้อมูลมัลติมีเดียนี้ก็ได้มาจากการนำข้อมูลมาทำการเข้ารหัสโดยใช้ตัวเข้ารหัส เช่น MPEG นั่นเอง ซึ่งในปัจจุบันซอฟต์แวร์ที่ใช้เล่นข้อมูลมัลติมีเดียก็ใช้ RTSP ในการรับส่งข้อมูลทั้งนั้น เช่น QuickTime Version 4 , Real System G2 และ Windows Media Player เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราอาจสงสัยว่าทำไมไม่ใช้ Hyper Text Transfer Protocol หรือ HTTP ในการรับส่งข้อมูลมัลติมีเดียผ่านอินเตอร์เน็ต เหตุผลก็คือ HTTP เป็นโปรโตคอลที่ถูกออกแบบมาให้รับส่งข้อมูลผ่าน TCP ซึ่งเน้นหนักไปทางความเชื่อถือได้ในการรับส่งข้อมูล (คือข้อมูลไม่สูญหายระหว่างทาง) โดยไม่สนใจเวลาที่ใช้รับส่งข้อมูล ดังนั้นเมื่อนำ HTTP มาใช้รับส่งข้อมูลมัลติมีเดียที่ต้องการความต่อเนื่องของข้อมูลก็จะเกิดปัญหาขึ้น คือข้อมูลที่ได้รับอาจหายไปเป็นช่วงๆ จึงต้องใช้ RTSP แทน เพื่อให้รับส่งข้อมูลได้อย่างต่อเนื่องและสามารถรับส่งข้อมูลในลักษณะการกระจาย (multicast) ได้ ทำให้ RTSP เหมาะสำหรับการรับส่งข้อมูลภาพ และเสียงผ่านอินเตอร์เน็ตมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

โครงงานนี้ทางกลุ่มได้เขียนโปรแกรมรับส่งเวปไฟล์ที่ได้จากการพูดผ่านไมโครโฟน โปรแกรมนี้สามารถใช้ได้ทั้งทางด้านส่งและด้านรับ โดยการรับส่งเวปไฟล์ระหว่างทางด้านส่งและด้านรับนั้นจะใช้โปรโตคอล TCP/IP เป็นโปรโตคอลสำหรับรับส่งข้อมูล โดยโครงงานนี้สร้างขึ้นโดยใช้ Visual Basic 6.0 เป็นเครื่องมือในการสร้าง

ในบทนี้จะอธิบายลำดับขั้นตอนการทำงานของโปรเจคทั้งหมดในโครงงาน พร้อมทั้งแผนผังการทำงาน (Flowchart) ของโปรแกรม และอธิบายฟังก์ชั่น (Function) ต่างๆ ที่ใช้ในโปรแกรม การรับส่งเวปไฟล์ระหว่างทางด้านส่งและด้านรับนั้นเป็นการรับส่งแบบสตรีมมิ่ง โดยทางด้านรับจะได้ยินเสียงในเวลาเดียวกันกับที่ทางด้านส่งพูด

3.1 ขั้นตอนการทำงานของโปรแกรม

เนื่องจากโปรแกรมที่เขียนขึ้นใช้เป็นได้ทั้งทางด้านส่งและทางด้านรับ จึงขออธิบายขั้นตอนการทำงานของโปรแกรมทั้งทางด้านส่งและทางด้านรับไปพร้อม ๆ กันดังนี้ (ดูรูปประกอบรูปที่ 3.1)

1) เริ่มแรก เมื่อทางด้านส่งและด้านรับเปิดโปรแกรมขึ้นมา โปรแกรมทางด้านส่งจะรอรับการร้องขอการติดต่อจากทางด้านรับที่พอร์ต (Port) 701

2) เมื่อทางด้านรับ ทำการร้องขอการติดต่อเข้ามา โปรแกรมจะถามว่าต้องการที่จะรับการร้องขอนี้หรือไม่ ถ้าต้องการรับการติดต่อโปรแกรมจะทำงานต่อในขั้นต่อไป แต่ถ้าไม่แล้วโปรแกรมจะวนกลับไปทำงานขั้นตอนที่ 1 อีกครั้ง

3) เมื่อทางด้านส่งรับการร้องขอแล้ว ก็จะทำการเก็บไอพีแอดเดรสของด้านที่ร้องขอเข้ามาไว้ในคอนเน็กชั่นลิสต์ (Connection List)

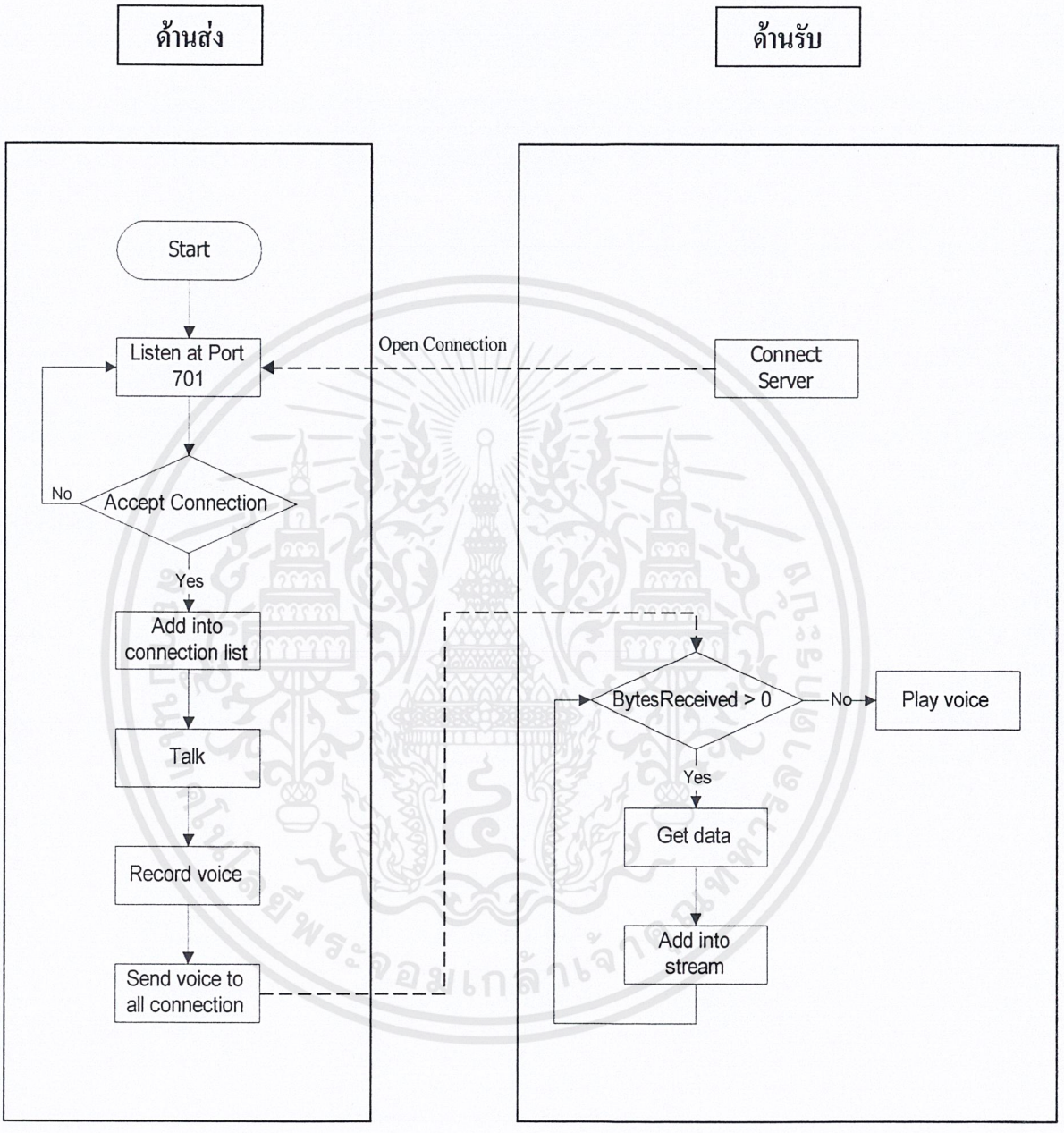
4) เมื่อทางด้านส่งกดปุ่มพูด (Talk) ค้างไว้ โปรแกรมจะทำการบันทึกเสียงไว้ในหน่วยความจำของเครื่องคอมพิวเตอร์

5) โปรแกรมจะทำการส่งเสียงที่ได้บันทึกไว้นั้นไปยังทางด้านรับ โดยการตัดเป็นช่วง ๆ ส่งไปพร้อม ๆ กับบันทึกเสียงพูดต่อไปด้วย

6) ส่วนทางด้านรับนั้น ก็จะทำการเช็คว่ามีข้อมูลเสียงส่งมาหรือไม่ ถ้ามีก็จะรับข้อมูลเสียงนั้นไว้แล้วเก็บรวมข้อมูลเสียงนั้นจนครบช่วงหนึ่ง พอถึงช่วงที่ไม่มีข้อมูลเสียง โปรแกรมก็จะทำการเล่นข้อมูลเสียงช่วงที่เก็บไว้นั้นก่อน ในขณะเดียวกันก็รอรับข้อมูลชุดต่อไปที่จะส่งมาด้วย

แผนผังการทำงานคร่าว ๆ ของโปรแกรม สามารถดูได้จากรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แผนผัง (Flowchart) การทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การเขียนโปรแกรม

การเขียนโปรแกรมของโครงงานนี้นั้น ทางกลุ่มเขียนขึ้นโดยใช้โปรแกรม Visual Basic 6.0 ซึ่งตัวโปรแกรมจะประกอบด้วย 3 ส่วน คือ

1. ส่วนที่เป็นตัวโปรแกรมทั้งหมด คือ ส่วนฟอร์ม (Form) ของโปรแกรม

2. ส่วนของโมดูล (Module) ประกอบไปด้วย ไฟล์ CHAT_DEF.BAS , CHAT_FUN.BAS , TCP_DEFS.BAS , TCP_FUNC.BAS

3. ส่วนเพิ่มเติม (Related Documents) คือ ไฟล์ voice.res เป็นรีซอร์ซเกี่ยวกับเสียง (Voice Resource)

3.2.1 การเขียนส่วนตัวโปรแกรม

ฟังก์ชัน (Function) ต่าง ๆ ที่ใช้ในโปรแกรมนั้น ในที่นี้จะเสนอเฉพาะฟังก์ชันสำคัญ ๆ ที่เกี่ยวกับการส่งข้อมูลเสียงแบบสตีมมิ่ง (Streaming) ซึ่งมีดังต่อไปนี้

- wStream.Playing เป็นฟังก์ชันที่ใช้แสดงสถานะว่ากำลังเล่นไฟล์เวปอยู่
- wStream.PlayDeviceFree เป็นฟังก์ชันที่ใช้ตรวจเช็คสถานะอุปกรณ์สำหรับเล่นไฟล์เวปว่าว่างอยู่รึไม่
- wStream.Recording เป็นฟังก์ชันที่ใช้แสดงสถานะว่ากำลังทำการบันทึกเสียงอยู่
- wStream.RecDeviceFree เป็นฟังก์ชันที่ใช้ตรวจเช็คสถานะอุปกรณ์สำหรับบันทึกเสียงพูดเป็นเวปไฟล์
- wStream.PlayWave เป็นฟังก์ชันที่ใช้สั่งให้เล่นเวปไฟล์
- wStream.RecordWave เป็นฟังก์ชันที่ใช้สั่งให้บันทึกเสียงที่พูดผ่านไมค์เป็นเวปไฟล์
- wStream.StreamInQueue เป็นฟังก์ชันที่ใช้ในการกำหนดพอร์ต (Port) ที่จะใช้ในการสตีมมิ่งตามลำดับ
- wStream.RemoveStreamFromQueue เป็นฟังก์ชันที่ใช้ยกเลิกการใช้พอร์ตที่ใช้ในการสตีมมิ่งออกจากลำดับนั้น

3.2.2 การเขียนโปรแกรมในส่วนของโมดูล

โมดูล เป็นส่วนที่ใช้สำหรับประกาศค่าคงที่ และฟังก์ชันต่างๆที่จะนำไปอ้างถึงในส่วนของตัวโปรแกรมทั้งหมด ประกอบด้วย 4 ส่วน ดังนี้คือ

1) CHAT_Defs เป็นส่วนที่ใช้กำหนดค่าคงที่ต่าง ๆ ที่ใช้ในตัวโปรแกรม เช่น เวปฟอร์แมต (Wave Format) , รูปต่าง ๆ ที่ใช้ในตัวโปรแกรม เป็นต้น

2) CHAT_Functions เป็นส่วนที่ใช้กำหนดฟังก์ชั่นต่าง ๆ ที่ใช้ในโปรแกรม

3) TCP_Defs เป็นส่วนค่าคงที่เกี่ยวกับทีซีพีพอร์ต (TCP Port) เช่น วอยซ์พอร์ต (VoicePort) ซึ่งเราใช้พอร์ต 701 และนัลพอร์ต (NullPort) ซึ่งก็คือพอร์ตที่เราไม่ใช้ กำหนดเป็นพอร์ต 0 เป็นต้น

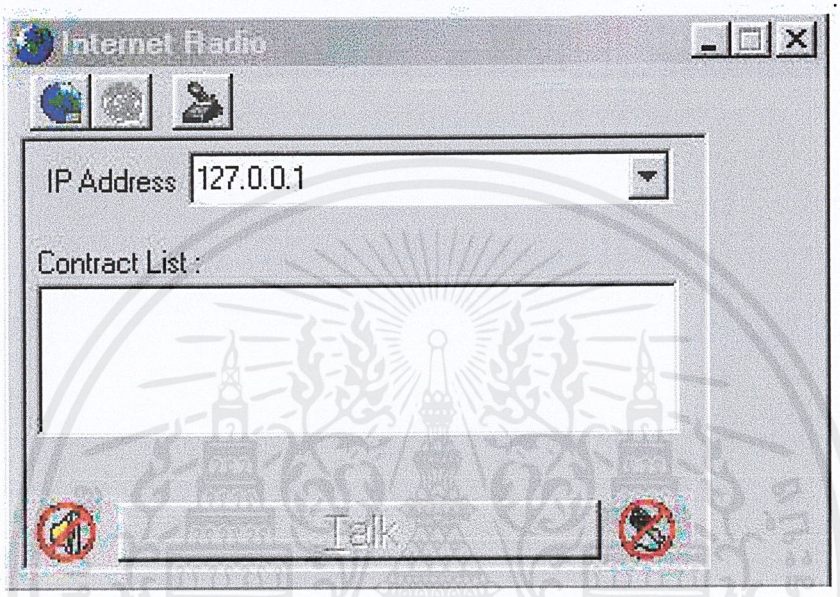
4) TCP_Functions เป็นส่วนที่ใช้กำหนดฟังก์ชั่นต่าง ๆ ที่ใช้ในติดต่อระหว่างเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 การทดลองโปรแกรม

1) ขั้นตอนแรก ทางด้านส่งและรับ จะต้องเปิดโปรแกรมอินเตอร์เน็ตเรดิโอ (Internet Radio) ทั้งสองด้าน เมื่อทำการเรียกโปรแกรมขึ้นมา จะพบหน้าต่างโปรแกรมแสดงดังรูปที่ 4.1



รูปที่ 4.1 แสดงโปรแกรมเริ่มแรก

จากรูป อธิบายได้ดังนี้คือ

1.1) ปุ่มที่เป็นรูปโลกที่อยู่ด้านซ้ายสุดนั้นเป็นปุ่มที่ใช้สำหรับเรียก (Call) ไปยังเครื่องคอมพิวเตอร์ที่เราต้องการติดต่อด้วย

1.2) ปุ่มถัดมาที่อยู่ติดกันนั้นเป็นปุ่มที่ใช้สำหรับยกเลิกการติดต่อ (Hang Up) กับเครื่องคอมพิวเตอร์ที่เราทำการเลือกจากคอนแทรกลิสต์ (Contract List) ซึ่งในขณะนี้ปุ่มนี้จะยังทำงานไม่ได้ เพราะยังไม่มีการติดต่อเกิดขึ้น

1.3) ปุ่มรูปโทรศัพท์ เมื่อกดปุ่มนี้แล้ว โปรแกรมจะทำการตอบรับการร้องขอเพื่อติดต่อแบบอัตโนมัติ (Auto Answering)

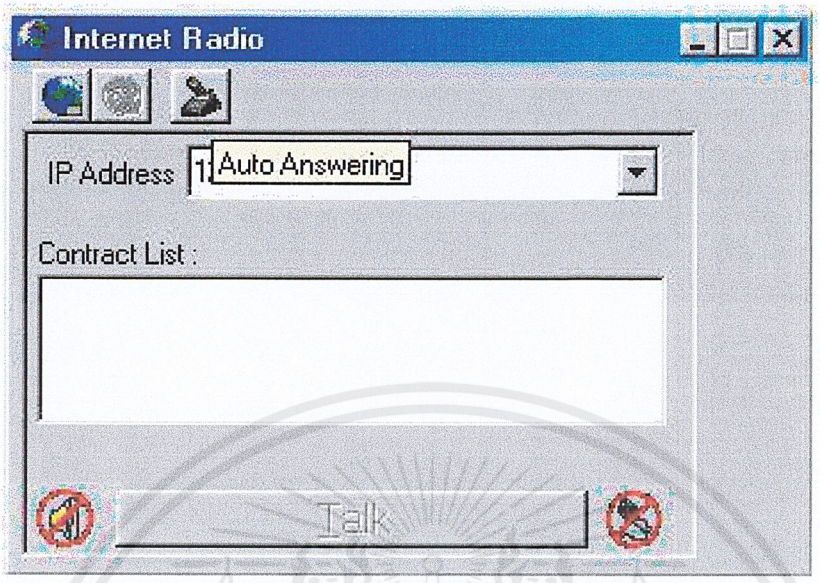
1.4) ช่องด้านบน เป็นช่องสำหรับใส่ค่าไอพีแอดเดรสของเครื่องคอมพิวเตอร์ที่เราต้องการจะทำการติดต่อด้วย ซึ่งได้ทำการจะตั้งค่าให้เป็น 127.0.0.1 ถือเป็นค่าพื้นฐาน โดยผู้ใช้สามารถเปลี่ยนแปลงได้ตามต้องการ ว่าต้องการติดต่อไปที่ไหน

1.5) คอนแทรกลิสต์ (Contract List) เป็นช่องที่แสดงค่าไอพีแอดเดรสของเครื่องคอมพิวเตอร์ที่ทำการติดต่ออยู่ในขณะนั้น

1.6) ปุ่มทอล์ค (Talk) เป็นปุ่มที่ต้องกดเมื่อต้องการพูดเพื่อส่งไปยังเครื่องคอมพิวเตอร์ที่ต่ออยู่

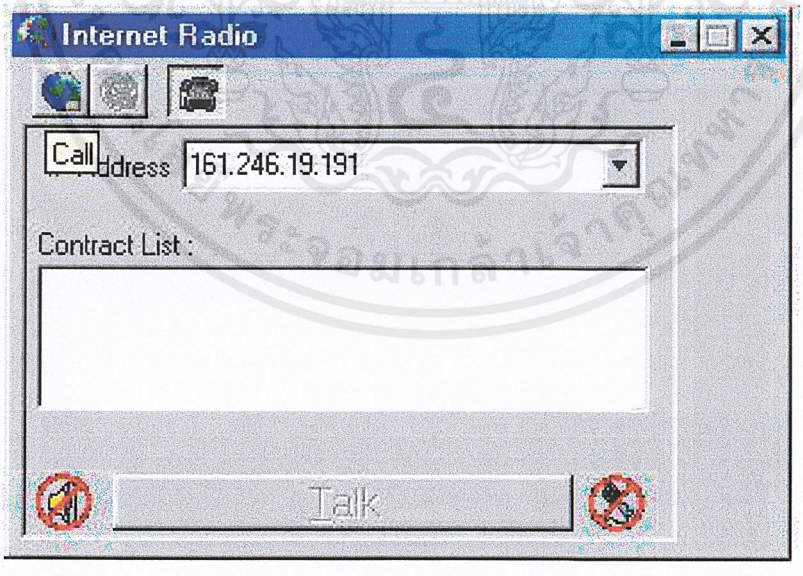
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) กดปุ่มตอบรับอัตโนมัติ เพื่อให้โปรแกรมทำการตอบรับการร้องขอโดยอัตโนมัติ ดังรูปที่ 4.2



รูปที่ 4.2 ทำการกดปุ่มตอบรับอัตโนมัติ

3) **ด้านรับ** ทำการใส่ค่าไอพีแอดเดรสที่ต้องการจะติดต่อ แล้วเลื่อนเมาส์ (Mouse) ไปคลิ้ก (Click) ที่ปุ่มเรียก (Call) ที่อยู่ซ้ายมือสุด



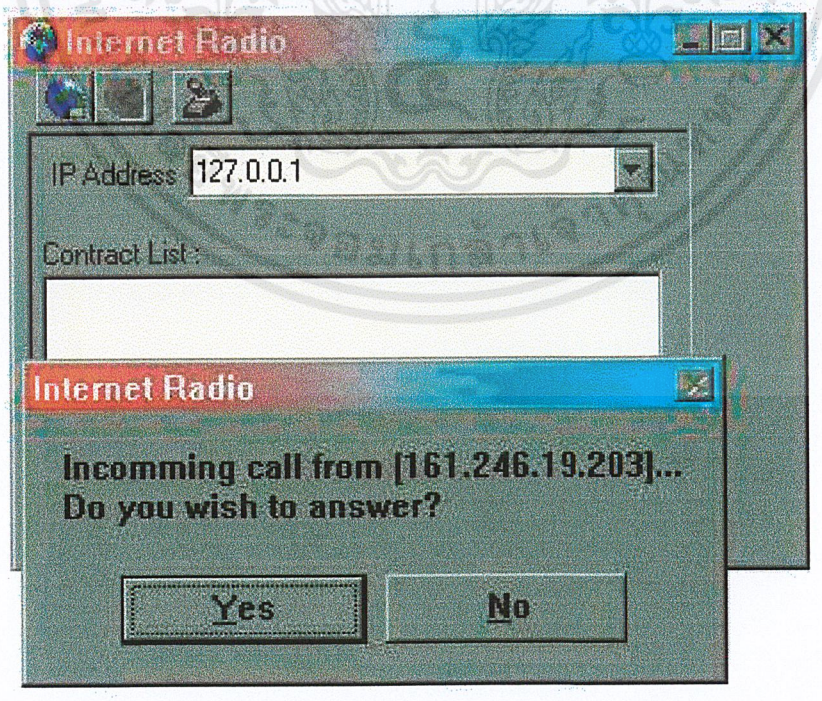
รูปที่ 4.3 ด้านรับทำการใส่ค่าไอพีแอดเดรสที่ต้องการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) **ด้านส่ง** ถ้าได้ทำการกดปุ่มตอบรับอัตโนมัติไว้แล้ว ไอพีแอดเดรสของเครื่องคอมพิวเตอร์ที่ร้องขอเข้ามาจะปรากฏขึ้นในคอนแทรกลิสต์ทันที ดังรูป

รูปที่ 4.4 ไอพีแอดเดรสของด้านรับปรากฏขึ้นที่ด้านส่ง

5) **ด้านส่ง** ถ้าทางด้านส่งไม่ได้ทำการกดปุ่มตอบรับอัตโนมัติไว้ เมื่อมีการร้องขอการติดต่อเข้ามา โปรแกรมจะถามว่าเราต้องการที่จะติดต่อกับเครื่องที่ร้องขอเข้ามาหรือไม่ แสดงดังรูป



รูปที่ 4.5 โปรแกรมถามว่าต้องการรับการร้องขอหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) **ด้านส่ง** เมื่อเลื่อนเมาส์มาไว้ที่ปุ่ม Talk แล้วกดค้างไว้ ก็จะสามารถพูด หรือ ส่งเสียงผ่านไมโครโฟนส่งไปยังด้านรับได้ โดยที่ปุ่ม Talk จะเปลี่ยนเป็น Talking ขณะที่กดค้างไว้ และโปรแกรมจะแสดงรูปไมโครโฟนรับเสียง โดยแสดงดังรูป

รูปที่ 4.6 โปรแกรมทางด้านส่งขณะพูดผ่านไมโครโฟน

7) **ด้านรับ** ด้านรับจะได้ยินเสียงที่ส่งมาพร้อมๆ กันกับด้านส่ง โดยปุ่ม Talk ของทางด้านรับจะเปลี่ยนเป็น Playing ขณะที่รับฟังเสียงจากด้านส่ง และโปรแกรมจะแสดงรูปลำโพงส่งเสียง โดยแสดงดังรูป

รูปที่ 4.7 โปรแกรมทางด้านรับขณะรับฟังเสียงจากด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) **ด้านรับ** เมื่อต้องการที่จะยกเลิกการติดต่อกับด้านส่ง ให้คลิ้กเลือกไอพีแอดเดรสของทางด้านส่งในคอนแทรกลิสต์ แล้วเลื่อนเมาส์ไปยังปุ่มยกเลิกการติดต่อ (Hang Up) เมื่อกดปุ่มก็จะยกเลิกการติดต่อระหว่างเครื่องคอมพิวเตอร์ไป แสดงดังรูป

รูปที่ 4.8 แสดงการยกเลิกการติดต่อ

4.2 ผลการทดลอง

ทางกลุ่มได้ทำการทดลองโดยใช้เครื่องคอมพิวเตอร์ 3 เครื่อง ซึ่งต่อกันอยู่ในเครือข่ายเดียวกัน เครื่องคอมพิวเตอร์ทั้ง 3 เครื่องมีสเปคเครื่องดังต่อไปนี้

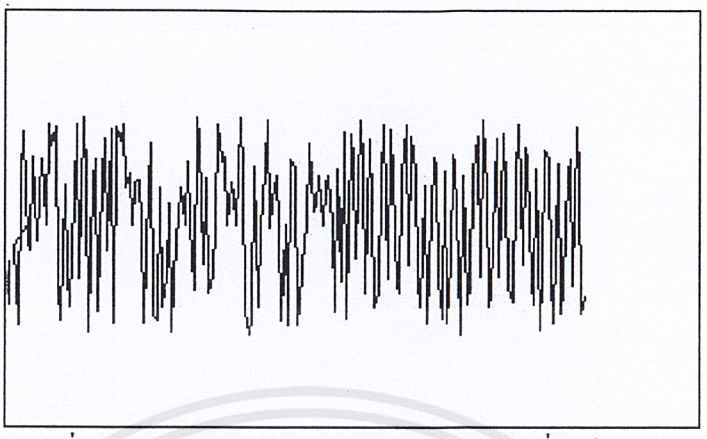
- Celeron 300 MHz RAM 32 MB
- Celeron 366 MHz RAM 64 MB
- Pentium II 400 MHz RAM 128 MB

กำหนดให้เครื่อง Pentium II 400 MHz เป็นเครื่องส่ง แล้วให้เครื่อง Celeron 300 และ 366 MHz เป็นเครื่องรับ โดยเราจะได้ทำการแปลงเสียงพูดที่เป็นเวปไฟล์ให้ออกมาเป็นรูปกราฟเพื่อดูความแตกต่างของเสียงที่ได้ยินของทางด้านส่งและด้านรับ และเครื่องคอมพิวเตอร์ทั้งสามเครื่องใช้การ์ดเสียงชนิดเดียวกัน คือ การ์ดเสียง Yamaha PCI Audio มีแซมปลิ้งบิตสูงสุด (Maximum Sampling Bits) 16 บิต และแซมปลิ้งเรตสูงสุด (Maximum Sampling Rate) เท่ากับ 44.1 kHz

ทางกลุ่มได้ทำการทดลอง แล้วได้ผลการทดลองออกมาดังต่อไปนี้

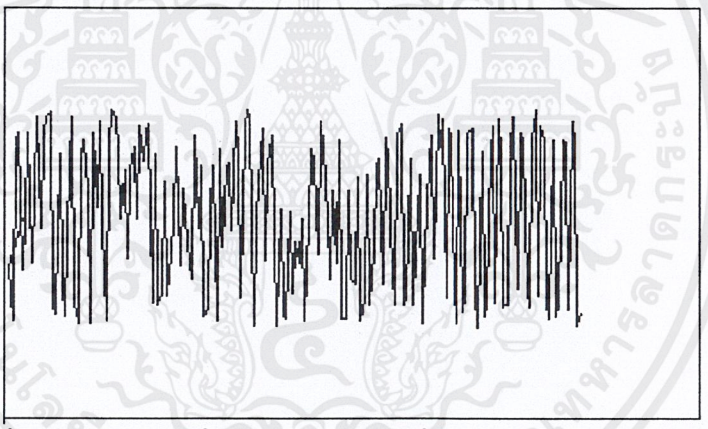
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ทางกลุ่มได้ทดลองพูดคำว่า “สวัสดีครับ” ผ่านไมโครโฟนทางด้านส่ง (เครื่อง Pentium II 400 MHz RAM 128 MB) แล้วทำการแปลงเป็นรูปกราฟออกมา ได้รูปกราฟออกมาดังนี้



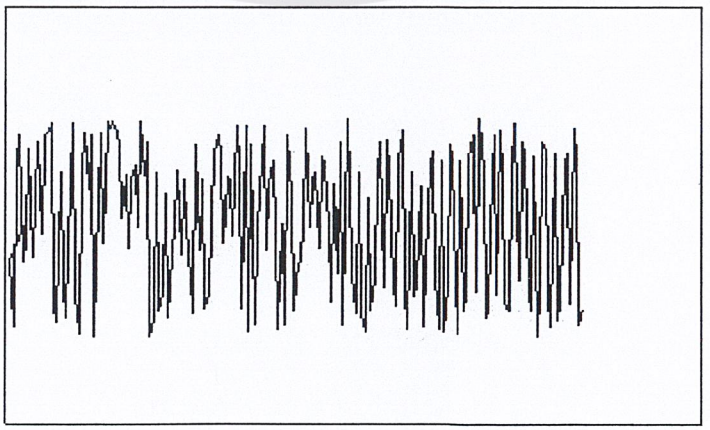
รูปที่ 4.9 แสดงเสียงพูดคำว่า “สวัสดีครับ” ของเครื่องด้านส่ง

2) เมื่อทำการส่งเสียงไปยังทางด้านรับเครื่องแรก (เครื่อง Celeron 300 MHz RAM 32 MB) จะได้ยินเสียงดังรูป



รูปที่ 4.10 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง Celeron 300 MHz RAM 32 MB

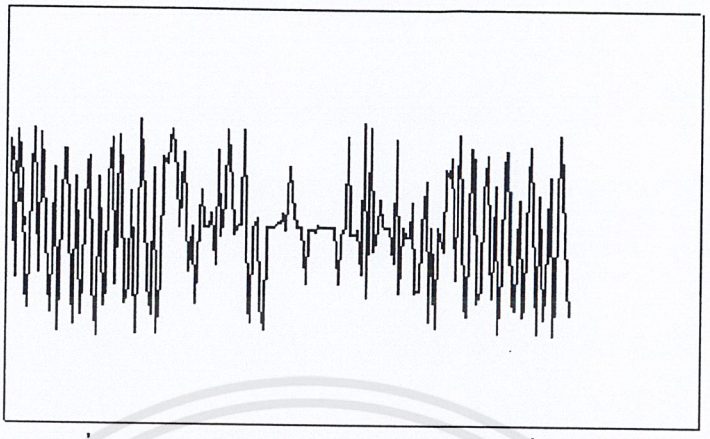
3) และเสียงที่ได้ยินทางด้านรับอีกเครื่องคือ เครื่อง Celeron 366 MHz RAM 64 MB เป็นดังนี้



รูปที่ 4.11 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง Celeron 366 MHz RAM 64 MB

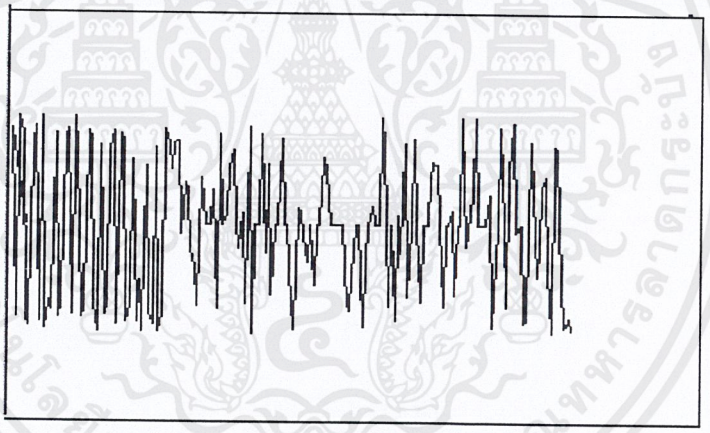
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ทำการทดลองพูดคำว่า “เฮลโล” ผ่านไมโครโฟนทางด้านส่ง (เครื่อง Pentium II 400 MHz RAM 128 MB) แล้วทำการแปลงเป็นรูปกราฟออกมา ได้รูปกราฟออกมาดังนี้



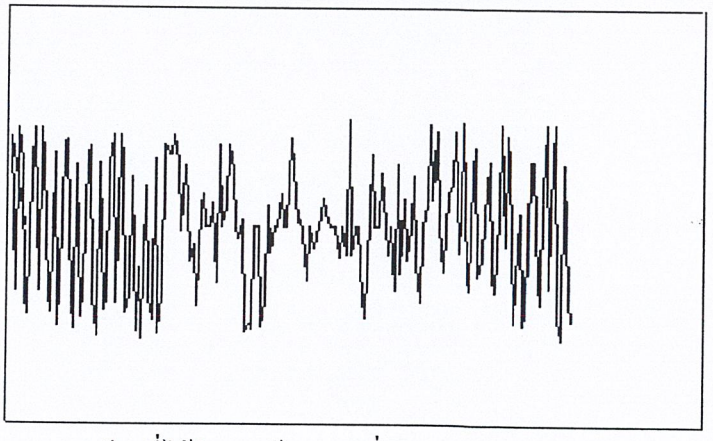
รูปที่ 4.12 แสดงเสียงพูดคำว่า “เฮลโล” ของเครื่องด้านส่ง

5) เมื่อทำการส่งเสียงไปยังทางด้านรับเครื่องแรก (เครื่อง Celeron 300 MHz RAM 32 MB) จะได้ยินเสียงดังรูป



รูปที่ 4.13 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง Celeron 300 MHz RAM 32 MB

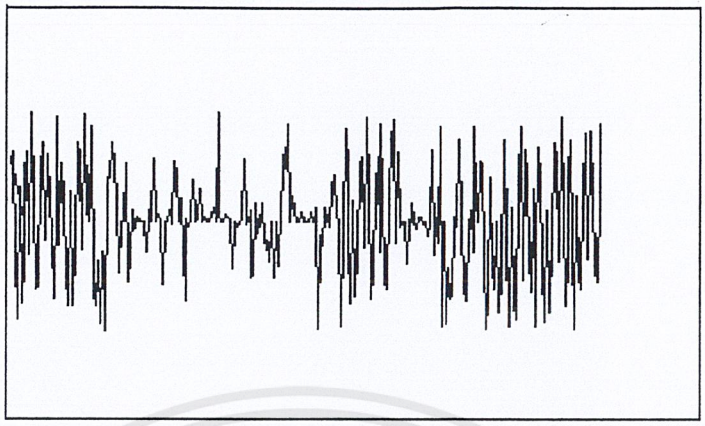
6) และเสียงที่ได้ยินทางด้านรับอีกเครื่องคือ เครื่อง Celeron 366 MHz RAM 64 MB เป็นดังนี้



รูปที่ 4.14 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง Celeron 366 MHz RAM 64 MB

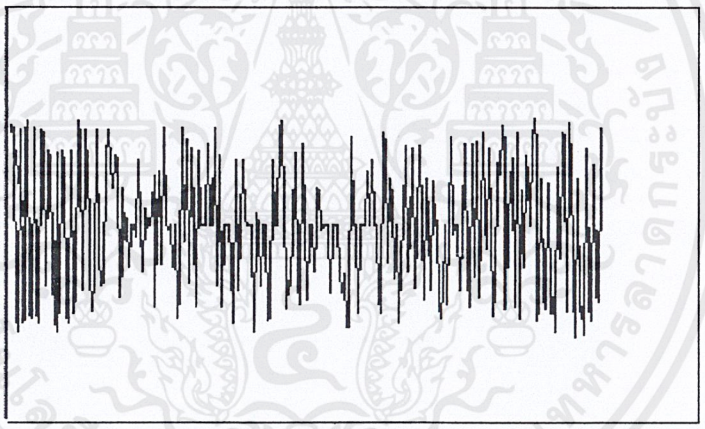
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) ทำการทดลองพูดคำว่า “ขอให้โชคดี” ผ่านไมโครโฟนทางด้านส่ง (เครื่อง Pentium II 400 MHz RAM 128 MB) แล้วทำการแปลงเป็นรูปกราฟออกมา ได้รูปกราฟออกมาดังนี้



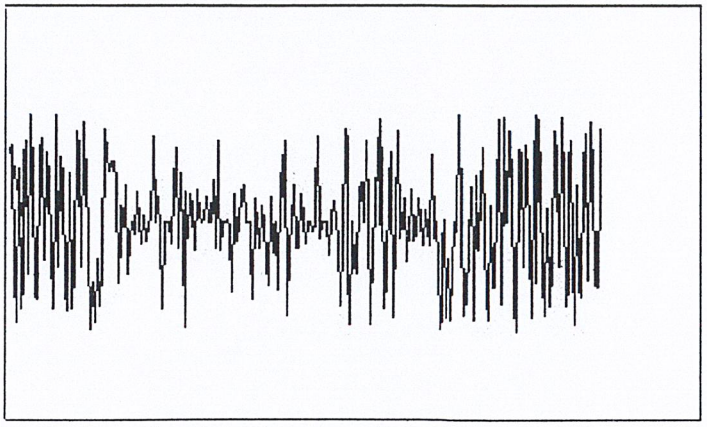
รูปที่ 4.15 แสดงเสียงพูดคำว่า “ขอให้โชคดี” ของเครื่องด้านส่ง

8) เมื่อทำการส่งเสียงไปยังทางด้านรับเครื่องแรก (เครื่อง Celercn 300 MHz RAM 32 MB) จะได้ยินเสียงดังรูป



รูปที่ 4.16 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง Celeron 300 MHz RAM 32 MB

9) และเสียงที่ได้ยินทางด้านรับอีกเครื่องคือ เครื่อง Celeron 366 MHz RAM 64 MB เป็นดังนี้



รูปที่ 4.17 แสดงเสียงที่ได้ยินทางด้านรับเครื่อง Celeron 366 MHz RAM 64 MB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองที่ผ่านมา จะพบว่า เมื่อทำการส่งเสียงผ่านเครือข่ายแล้ว จะเกิดสัญญาณรบกวนขึ้น ซึ่งจะทำให้เกิดเสียงก้อง เสียงสะท้อน (Echo) ขึ้นที่ด้านรับ โดยเครื่อง Celeron 366 MHz RAM 64 MB จะได้ยินเสียงที่มีคุณภาพเสียงที่ดีกว่าเครื่อง Celeron 300 MHz RAM 32 MB

นอกจากนี้ทางกลุ่มยังได้ทำการทดลองแบบเดียวกันกับเครื่องคอมพิวเตอร์ที่มีสเปคเครื่องต่างๆ กันอีกหลายเครื่อง เพื่อต้องการหาว่า สเปคของเครื่องคอมพิวเตอร์ขั้นต่ำ (Minimum Requirement) เพื่อที่จะทำให้เสียงที่ได้ยินมีคุณภาพดีพอที่จะสื่อสารกันได้ โดยได้ผลการทดลองดังตารางต่อไปนี้

สเปคด้านส่ง	สเปคด้านรับ	Performance Quality
Pentium I –166 MHz RAM – 32 MB	P – 166 MHz	Low
	Celeron – 300 MHz	Low
	Celeron – 366 MHz	Low
	Pentium II – 400 MHz	Low
	Pentium III – 500 MHz	Low
Celeron – 300 MHz RAM – 32 MB	P – 166 MHz	Low
	Celeron – 300 MHz	Low
	Celeron – 366 MHz	Low
	Pentium II – 400 MHz	Fair
	Pentium III – 500 MHz	Fair
Celeron – 366 MHz RAM – 64 MB	P – 166 MHz	Low
	Celeron – 300 MHz	Low
	Celeron – 366 MHz	Fair
	Pentium II – 400 MHz	Fair
	Pentium III – 500 MHz	Fair
Pentium II – 400 MHz RAM – 128 MB	P – 166 MHz	Low
	Celeron – 300 MHz	Low
	Celeron – 366 MHz	Fair
	Pentium II – 400 MHz	Good
	Pentium III – 500 MHz	Very good
Pentium III – 500 MHz RAM – 128 MB	P – 166 MHz	Low
	Celeron – 300 MHz	Fair
	Celeron – 366 MHz	Good
	Pentium II – 400 MHz	Very good
	Pentium III – 500 MHz	Very good

ตารางที่ 4.1 แสดงผลการทดลองคุณภาพเสียงของเครื่องคอมพิวเตอร์สเปคต่าง ๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 สรุปผลการทดลอง

จากผลการทดลองที่ได้ จะพบว่า เสียงที่ได้ยินทางด้านรับนั้น องค์ประกอบหนึ่งขึ้นอยู่กับประสิทธิภาพของเครื่องคอมพิวเตอร์ ทั้งของทางด้านส่งและด้านรับ โดยจากการทดลอง เครื่องคอมพิวเตอร์ด้านส่งที่สามารถส่งเสียงที่มีคุณภาพดีนั้น ควรใช้เครื่อง Celeron 366 MHz ที่มี RAM 64 MB ขึ้นไป จึงจะสามารถส่งเสียงที่มีคุณภาพดีได้ ส่วนเครื่องคอมพิวเตอร์ทางด้านรับที่ทำให้ได้ยินเสียงที่มีคุณภาพดีนั้น ควรใช้เครื่อง Celeron 300 MHz ที่มี RAM 32 MB ขึ้นไป จึงจะได้ยินเสียงที่มีคุณภาพดี จะเห็นว่าเครื่องคอมพิวเตอร์ด้านส่งควรจะมีประสิทธิภาพสูงเพื่อที่จะสามารถส่งเสียงที่มีคุณภาพได้ ส่วนเครื่องด้านรับมีประสิทธิภาพพอใช้ก็สามารถรับฟังเสียงที่ชัดเจนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์

5.1 บทวิจารณ์

โครงงานนี้ทางกลุ่มได้ศึกษาโปรแกรมและเก็บข้อมูลเกี่ยวกับสถานีวิทยุบนระบบเครือข่ายอินเตอร์เน็ต ซึ่งเป็นเทคโนโลยีค่อนข้างใหม่ในประเทศไทย โดยในต่างประเทศได้มีการใช้งานกันอย่างกว้างขว้าง ซึ่งคาดว่าในอนาคตประเทศไทยจะมีการนำมาใช้อย่างกว้างขวางต่อไป เพราะมีข้อดีค่อนข้างมาก แต่ก็ยังมีข้อเสียคือคุณภาพเสียงยังดีไม่เท่ากับสถานีคลื่นวิทยุทั่วไป ซึ่งนั่นก็ขึ้นอยู่กับประสิทธิภาพของเครื่องคอมพิวเตอร์และระบบเครือข่ายด้วย

สถานีวิทยุบนอินเตอร์เน็ตเป็นการแพร่กระจายเสียงผ่านระบบเครือข่ายโดยเป็นการรับส่งข้อมูลเสียงแบบสตรีมมิ่ง (Streaming Audio) ซึ่งทางกลุ่มได้ทำการเขียนโปรแกรม เพื่อรับส่งเวปไฟล์ (Wave File) ซึ่งได้จากการพูดผ่านไมโครโฟน โดยเมื่อทางด้านส่งพูดแล้วทางด้านรับจะได้ยินเสียงนั้นพร้อมๆ กัน ในเวลาเดียวกันกับที่ด้านส่งพูด

5.2 บทสรุป

จากการทดลองโดยใช้เครื่องคอมพิวเตอร์ที่ต่อกันอยู่ในระบบเครือข่ายเดียวกัน แล้วให้เครื่องหนึ่งเป็นด้านส่งและอีกเครื่องหนึ่งเป็นด้านรับนั้น ผลการทดลองโดยการพูดผ่านไมโครโฟนทางด้านส่งแล้วทางด้านรับจะได้ยินเสียงค่อนข้างชัดเจน แต่จะได้ยินเสียงพูดช้ากว่าทางด้านส่งประมาณหนึ่งวินาที ซึ่งอาจเป็นเพราะประสิทธิภาพของเครื่องคอมพิวเตอร์ที่ใช้ทดลองไม่ดีพอ และคุณภาพของเสียงที่ได้ยินก็ขึ้นอยู่กับประสิทธิภาพของเครื่องคอมพิวเตอร์ การ์ดเสียง ไมโครโฟน และลำโพงที่ใช้ด้วย

5.3 ปัญหาที่พบจากการทดลอง

การทดลองยังไม่เคยได้ทดสอบผ่านโมเด็มและเครือข่ายคอมพิวเตอร์ที่อื่น ๆ เนื่องจากเป็นการทดสอบระหว่างเครื่องคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายเดียวกัน จึงยังไม่สามารถระบุผลการทดลองโปรแกรม โดยที่เครื่องคอมพิวเตอร์ทางด้านส่งและด้านรับต่ออยู่คนละระบบเครือข่ายกัน หรือต่อผ่านระบบเครือข่ายอินเตอร์เน็ตได้

5.4 แนวทางในการพัฒนา

โปรแกรมที่ทางกลุ่มได้เขียนขึ้นนี้ ยังไม่สามารถส่งไฟล์เพลงที่เป็นเวปไฟล์ได้ เนื่องจากเป็นไฟล์ที่มีขนาดใหญ่มาก จึงต้องใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงมาก นอกจากนี้คุณภาพของเสียงที่ได้ยินยังไม่ดีมากนัก จึงควรพัฒนาให้คุณภาพเสียงดียิ่งขึ้น และเสียงที่ได้ยินทางด้านรับยังช้ากว่าทางด้านส่งอยู่เล็กน้อย ทั้งหมดนี้เป็นข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาโปรแกรมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรม (ไฟล์ chat.frm)

```
Option Explicit
Public CLOSINGAPPLICATION As Boolean                    ' Application status flag
Public wStream As Object
Dim cnt As Integer
Dim Iport1 As Integer

'------------------------------------------------------------
Private Sub cmdTalk_Click()                             ' Activates Audio PlayBack
'------------------------------------------------------------
  Dim rc As Long                                        ' Return Code Variable
  Dim iPort As Integer                                  ' Local Port
  Dim itm As Integer                                    ' Current listitem
'------------------------------------------------------------
  If (Not wStream.Playing And wStream.PlayDeviceFree And _
    Not wStream.Recording And wStream.RecDeviceFree) Then   ' Validate Audio Device Status
    wStream.Playing = True                              ' Turn Playing Status On
    cmdTalk.Caption = "&Playing"                        ' Modify Button Status Caption
    Screen.MousePointer = vbHourglass                   ' Set Pointer To HourGlass

    iPort = wStream.StreamInQueue
    Iport1 = iPort
    Do While (iPort <> NULLPORTID)                      ' While socket ports have data to playback
      inLight.Picture = ImgIcons.ListImages(speakON).Picture   ' Flash playback image
      inLight.Refresh                                   ' Repaint picture image

      For itm = 0 To ConnectionList.ListCount - 1       ' Search for listitem currently playing sound data
        If (ConnectionList.ItemData(itm) = iPort) Then  ' If a match is found...
          ConnectionList.TopIndex = itm                 ' Set that listitem to top of listbox
          ConnectionList.Selected(itm) = True           ' Select listitem to show who is currently talking...
          Exit For                                      ' Quit listitem search
        End If
      Next                                              ' Check next listitem
      If cnt = 1 Then cnt = 3
      If cnt = 0 Then cnt = 1
      rc = wStream.PlayWave(Me.hWnd, iPort)             ' Play wave data in iPort...
      Call wStream.RemoveStreamFromQueue(iPort)         ' Remove PortID From PlayWave Queue
      iPort = wStream.StreamInQueue

      inLight.Picture = ImgIcons.ListImages(speakOFF).Picture   ' Show done talking image...
      inLight.Refresh                                   ' Repaint image...
    Loop                                                ' Search for next socket in playback queue

    ConnectionList.TopIndex = 0                         ' Reset top image...
    If (ConnectionList.ListCount > 0) Then
      ConnectionList.Selected(0) = True                 ' Deselect previously listitem
      ConnectionList.Selected(0) = False                ' Deselect currently selected listitem
    End If
    Screen.MousePointer = vbDefault                     ' Set Pointer To Normal
    cmdTalk.Caption = "&Talk"                           ' Modify Button Status Caption
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        wStream.Playing = False                                    ' Turn Playing Status Off
    End If
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------
Private Sub cmdTalk_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
' Activates Audio Recording...
'-------------------------------------------------------------
    Dim rc As Long                                                 ' Return Code Variable
'-------------------------------------------------------------
    If (Not wStream.Playing And _
        Not wStream.Recording And _
            wStream.RecDeviceFree And _
            wStream.PlayDeviceFree) Then                           ' Check Audio Device Status
        wStream.Recording = True                                   ' Set Recording Flag
        cmdTalk.Caption = "&Talking"                               ' Update Button Status To "Talking"
        Screen.MousePointer = vbHourglass                          ' Set Hourglass
        outLight.Picture = ImgIcons.ListImages(mikeON).Picture     ' Show outgoing message image
        outLight.Refresh                                           ' Repaint image

        rc = wStream.RecordWave(Me.hWnd, TCPSocket)                ' Record voice and send to all connected sockets

        outLight.Picture = ImgIcons.ListImages(mikeOFF).Picture    ' Show done image
        outLight.Refresh                                           ' Repaint image
        Screen.MousePointer = vbDefault                            ' Reset Mouse Pointer
        cmdTalk.Caption = "&Talk"                                  ' Reset Button Status

        If Not wStream.Playing And _
            wStream.PlayDeviceFree And _
            wStream.RecDeviceFree Then                             ' Is Audio Device Free?
            Call cmdTalk_Click                                     ' Active Playback Of Any Inbound Messages...
        End If
    End If
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

Private Sub cmdTalk_MouseUp(Button As Integer, Shift As Integer, x As Single, Y As Single)
    wStream.Recording = False                                      ' Stop Recording
End Sub

Private Sub connectionlist_Click()
    Tools.Buttons(tbHANGUP).Enabled = True
End Sub

'-------------------------------------------------------------
Private Sub ConnectionList_DblClick()
'-------------------------------------------------------------
    Dim MemberID As String                                         ' (Server)(TCPidx)(RemotelP)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Dim Idx As Long                                          ' TCP idx
'-------------------------------------------------------------
    If (ConnectionList.Text = "") Then Exit Sub
    MemberID = ConnectionList.List(ConnectionList.MultiSelect)   ' Get The Conversation MemberID String From List Box

    Call GetIdxFromMemberID(TCPSocket, MemberID, Idx)        ' Get TCP idx From Member ID
    Call RemoveConnectionFromList(TCPSocket(Idx), ConnectionList)  ' Clear ListBox Entry(s)...
    Call Disconnect(TCPSocket(Idx))                          ' Disconnect Socket Connection
    Unload TCPSocket(Idx)                                    ' Destroy socket instance

    cmdTalk.Enabled = (ConnectionList.ListCount > 0)         ' Enable/Disable Talk Button...
    Tools.Buttons(tbHANGUP).Enabled = (ConnectionList.Text <> "")
    If Not cmdTalk.Enabled Then
      inLight.Picture = ImgIcons.ListImages(speakNO).Picture
      outLight.Picture = ImgIcons.ListImages(mikeNO).Picture
    End If
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------


'-------------------------------------------------------------
Private Sub Form_Load()
'-------------------------------------------------------------
    Dim rc As Long                                           ' Return Code Variable
    Dim Idx As Long                                          ' Current TCP idx variable
    Dim TCPidx As Long                                       ' Newly created TCP idx value
'-------------------------------------------------------------
    CLOSINGAPPLICATION = False                               ' Set status to not closing
    Call InitServerList(txtServer)                           ' Get Common Servers List
    txtServer.Text = txtServer.List(0)                       ' Display First Name In The List
    'imgStatus = ImgIcons.ListImages(phoneHungUp).Picture    ' Change Icon To Phone HungUp

    Set wStream = CreateObject("WaveStreaming.WaveStream")
    Call wStream.InitACMCodec(WAVE_FORMAT_GSM610, TIMESLICE)
'   Call wStream.InitACMCodec(WAVE_FORMAT_ADPCM, TIMESLICE)
'   Call wStream.InitACMCodec(WAVE_FORMAT_MSN_AUDIO, TIMESLICE)
'   Call wStream.InitACMCodec(WAVE_FORMAT_PCM, TIMESLICE)

    cmdTalk.Enabled = False                                  ' Disable Until Connect
    Tools.Buttons(tbHANGUP).Enabled = (ConnectionList.Text <> "")
    inLight.Picture = ImgIcons.ListImages(speakNO).Picture
    outLight.Picture = ImgIcons.ListImages(mikeNO).Picture

    Call Listen(TCPSocket(0))                                ' Listen For TCP Connection
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'------------------------------------------------------------
Private Sub Form_Unload(Cancel As Integer)
'------------------------------------------------------------
    Dim Idx As Long                                 ' TCP socket index
    Dim Socket As Winsock ' TCP socket control
'------------------------------------------------------------
    'Call wStream.PlayWave1(Me.hWnd, Iport1, 2)     ' Play wave data in iPort...
    CLOSINGAPPLICATION = True                       ' Set status flag to closing...
    For Each Socket In TCPSocket                    ' For each socket instance
      Call Disconnect(Socket)                       ' Close connection/listen
    Next                                            ' Next Cntl
    Set wStream = Nothing
    End                                             ' End Program
'------------------------------------------------------------
End Sub
'------------------------------------------------------------


'------------------------------------------------------------
Private Sub TCPSocket_Close(Index As Integer)
' Closing Current TCP Connection...
'------------------------------------------------------------
    Call RemoveConnectionFromList(TCPSocket(Index), ConnectionList)   ' Remove Connection From List
    Call Disconnect(TCPSocket(Index))                                 ' Close Port Connection...

    cmdTalk.Enabled = (ConnectionList.ListCount > 0)                  ' Enable/Disable Talk Button...
    If Not cmdTalk.Enabled Then
      inLight.Picture = ImgIcons.ListImages(speakNO).Picture
      outLight.Picture = ImgIcons.ListImages(mikeNO).Picture
    End If

    Tools.Buttons(tbHANGUP).Enabled = (ConnectionList.Text <> "")
    If cmdTalk.Enabled Then
      'imgStatus = ImgIcons.ListImages(phoneHungUp).Picture           ' Show Phone HungUp Icon...
    End If

    Unload TCPSocket(Index)                                           ' Destroy socket instance
'------------------------------------------------------------
End Sub
'------------------------------------------------------------


'------------------------------------------------------------
Private Sub TCPSocket_Connect(Index As Integer)
' TCP Connection Has Been Accepted And Is Open...
'------------------------------------------------------------
    Call AddConnectionToList(TCPSocket(Index), ConnectionList)        ' Add New Connection To List

    'imgStatus = ImgIcons.ListImages(phoneRingIng).Picture            ' Show Phone Ringing Icon
    Call ResPlaySound(RingOutId)
    'imgStatus = ImgIcons.ListImages(phoneAnswered).Picture           ' Show Phone Answered Icon
    cmdTalk.Enabled = True                                            ' Enabled For Connection...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Tools.Buttons(tbHANGUP).Enabled = (ConnectionList.Text <> "")
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------


'-------------------------------------------------------------
Private Sub TCPSocket_ConnectionRequest(Index As Integer, ByVal requestID As Long)
' Accepting Inbound TCP Connection Request...
'-------------------------------------------------------------
    Dim rc As Long
    Dim Idx As Long
    Dim RemHost As String
'-------------------------------------------------------------
    If (TCPSocket(Index).RemoteHost <> "") Then
        RemHost = UCase(TCPSocket(Index).RemoteHost)
    Else
        RemHost = UCase(TCPSocket(Index).RemoteHostIP)
    End If

    If (Tools.Buttons(tbAUTOANSWER).Value = tbrUnpressed) Then
        rc = MsgBox("Incomming call from [" & RemHost & "]..." & vbCrLf & _
                "Do you wish to answer?", vbYesNo)                  ' Prompt user to answer...
    Else
        rc = vbYes
    End If

    If (rc = vbYes) Then
        Idx = InstanceTCP(TCPSocket)                                ' Instance TCP Control...
        If (Idx > 0) Then                                           ' Validate that control instance was created...
            TCPSocket(Idx).LocalPort = 0                            ' Set local port to 0, in order to get next available port.
            Call TCPSocket(Idx).Accept(requestID)                   ' Accept connection
            Call AddConnectionToList(TCPSocket(Idx), ConnectionList)    ' Add New Connection To List

            'imgStatus = ImgIcons.ListImages(phoneRingIng).Picture      ' Show Phone Ringing Icon
            Call ResPlaySound(RingInId)
            'imgStatus = ImgIcons.ListImages(phoneAnswered).Picture     ' Show Phone Answered Icon
            cmdTalk.Enabled = True                                  ' Enabled For Connection...
            Tools.Buttons(tbHANGUP).Enabled = (ConnectionList.Text <> "")
        End If
    End If
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------


'-------------------------------------------------------------
Private Sub TCPSocket_DataArrival(Index As Integer, ByVal BytesTotal As Long)
' Incomming Buffer On...
'-------------------------------------------------------------
    Dim rc As Long                                                  ' Return Code Variable
    Dim WaveData() As Byte ' Byte array of wave data
    Dim WaveData1() As Byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Static ExBytes(MAXTCP) As Long                  ' Extra bytes in frame buffer
    Static ExData(MAXTCP) As Variant                ' Extra bytes from frame buffer
    Dim ExData1(MAXTCP) As String
    Dim i As Long
    Dim x As Long


'-------------------------------------------------------------


With wStream
  If (TCPSocket(Index).BytesReceived > 0) Then      ' Validate that bytes where actually received
    Do While (TCPSocket(Index).BytesReceived > 0)   ' While data available...
      If (ExBytes(Index) = 0) Then                  ' Was there leftover data from last time
        If (.waveChunkSize <= TCPSocket(Index).BytesReceived) Then  ' Can we get and entire wave buffer of data
          Call TCPSocket(Index).GetData(WaveData, vbByte + vbArray, .waveChunkSize) ' Get 1 wave buffer of data
          i = UBound(WaveData)
          x = InstrBt(WaveData, 255)
          'If i > 324 Then
            'Label3.Caption = "One " + Str(x)
            'Label1.Caption = "One " + Str(WaveData(i))
          Call .SaveStreamBuffer(Index, WaveData)   ' Save wave data to buffer
          Call .AddStreamToQueue(Index)             ' Queue current stream for playback
        Else
          ExBytes(Index) = TCPSocket(Index).BytesReceived  ' Save Extra bytes
          Call TCPSocket(Index).GetData(ExData(Index), vbByte + vbArray, ExBytes(Index)) ' Get Extra data
        End If
      Else
        Call TCPSocket(Index).GetData(WaveData, vbByte + vbArray, .waveChunkSize - ExBytes(Index)) ' Get leftover bits
        ExData(Index) = MidB(ExData(Index), 1) & MidB(WaveData, 1)  ' Sync wave bits...
        WaveData1 = ExData(Index)
        i = UBound(WaveData1)
        x = InstrBt(WaveData1, 255)
        'Label3.Caption = "Two " + Str(x)
        'Label1.Caption = "Two " + Str(WaveData1(i))
        Call .SaveStreamBuffer(Index, WaveData1)    ' Save the current wave data to the wave buffer

        Call .AddStreamToQueue(Index)               ' Queue the current wave stream
        ExBytes(Index) = 0                          ' Clear Extra byte count
        ExData(Index) = ""                          ' Clear Extra data buffer
      End If
    Loop                                            ' Look for next Data Chunk

    If (Not .Playing And .PlayDeviceFree And _
      Not .Recording And .RecDeviceFree) Then       ' Check Audio Device Status
      Call cmdTalk_Click                            ' Start PlayBack...
    End If
  End If
End With
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub TCPSocket_Error(Index As Integer, ByVal Number As Integer, Description As String, ByVal Scode As Long, ByVal Source As String, ByVal
HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
TCPSocket(Index).Close                                          ' Close down socket

   Debug.Print "TCPSocket_Error: Number:", Number
   Debug.Print "TCPSocket_Error: Scode:", Hex(Scode)
   Debug.Print "TCPSocket_Error: Source:", Source
   Debug.Print "TCPSocket_Error: HelpFile:", HelpFile
   Debug.Print "TCPSocket_Error: HelpContext:", HelpContext
   Debug.Print "TCPSocket_Error: Description:", Description
   Call DebugSocket(TCPSocket(Index))
End Sub

'------------------------------------------------------------
Private Sub Tools_ButtonClick(ByVal Button As Button)
'------------------------------------------------------------
   Dim rc As Long                                               ' Return Code Variable
   Dim Idx As Long                                              ' TCP Socket control index
   Dim LocalPort As Long                                        ' LocalPort Setting
   Dim RemotePort As Long                                       ' RemotePort Setting
'------------------------------------------------------------
   Select Case Button.Index
   Case tbCALL
      Idx = InstanceTCP(TCPSocket)                              ' Instance TCP Control...

      If (Idx > 0) Then                                         ' Did control instance get created???
         Button.Enabled = False                                 ' Disable Connect Button
         ConnectionList.Enabled = False                         ' Disable connection list box

         On Error Resume Next
         If Not Connect(TCPSocket(Idx), txtServer.Text, VOICEPORT) Then     ' Attempt to connect
            Unload TCPSocket(Idx)                               ' Connect failed unload control instance
         End If

         ConnectionList.Enabled = True                          ' Renable connection list box
         Button.Enabled = True                                  ' Enable Connect Button
      End If
   Case tbHANGUP
      ConnectionList_DblClick
   Case tbAUTOANSWER
      If (Button.Value = tbrPressed) Then
         Button.Image = phoneHungUp
      Else
         Button.Image = phoneAnswered
      End If
   End Select
End Sub

'------------------------------------------------------------
Private Sub txtServer_KeyPress(KeyAscii As Integer)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'-----------------------------------------------------------
    Dim Conn As Long                                    ' Index counter
'-----------------------------------------------------------
    If (KeyAscii = vbKeyReturn) Then                    ' If Return Key Was Pressed...
      For Conn = 0 To txtServer.ListCount               ' Search Each Entry In ListBox
        If (UCase(txtServer.Text) = UCase(txtServer.List(Conn))) Then Exit Sub
      Next                                              ' If Found Then Exit
      txtServer.AddItem UCase(txtServer.Text)           ' Add Server To List
    End If
'-----------------------------------------------------------
End Sub
'-----------------------------------------------------------
Function InstrBt(dt, chrb) As Long

Dim i As Long
Dim x As Long
i = LenB(dt) - 1
For x = 0 To i
   If dt(x) = chrb Then
      InstrBt = x
      Exit Function
   End If
Next x

End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนโมดูล CHAT_Defs (ไฟล์ CHAT_DEF.BAS)

```
Option Explicit

' Application User Defined Types...
' Sound Format
Public Const WAVE_FORMAT_PCM = &H1                          ' Microsoft Windows PCM Wave Format
Public Const WAVE_FORMAT_ADPCM = &H11                       ' ADPCM Wave Format
Public Const WAVE_FORMAT_IMA_ADPCM = &H11                   ' IMA ADPCM Wave Format
Public Const WAVE_FORMAT_DVI_ADPCM = &H11                   ' DVI ADPCM Wave Format
Public Const WAVE_FORMAT_DSPGROUP_TRUESPEECH = &H22         ' DSP Group Wave Format
Public Const WAVE_FORMAT_GSM610 = &H31                      ' GSM610 Wave Format
Public Const WAVE_FORMAT_MSN_AUDIO = &H32                   ' MSN Audio Wave Format

Public Const TIMESLICE = 0.2                                ' Time Slicing 1/5 Second

' Application Constants...
Public Const NoOfRings = 1                                  ' Number Of Times In/Out Bound Calls Ring...

Public Const phoneHungUp = 3                                ' Hangup Status Icon...
Public Const phoneRingIng = 2                               ' Ringing Status Icon...
Public Const phoneAnswered = 1                              ' Answered Status Icon...
Public Const mikeNO = 6
Public Const mikeOFF = 7
Public Const mikeON = 8
Public Const speakNO = 9
Public Const speakOFF = 10
Public Const speakON = 11

Public Const RingInId = 101                                 ' Ringing InBound Sound...
Public Const RingOutId = 102                                ' Ringing OutBound Sound...

' Toolbar constants...
Public Const tbCALL = 2
Public Const tbHANGUP = 3
Public Const tbAUTOANSWER = 5

'== flag values for wFlags parameter ================================
Public Const SND_SYNC = &H0                                 ' play synchronously (default)
Public Const SND_ASYNC = &H1                                ' play asynchronously
Public Const SND_NODEFAULT = &H2                            ' don't use default sound
Public Const SND_MEMORY = &H4                               ' lpszSoundName points to a memory file
Public Const SND_LOOP = &H8                                 ' loop the sound until next sndPlaySound
Public Const SND_NOSTOP = &H10                              ' don't stop any currently playing sound

'== MCI Wave API Declarations ==========================================
Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" (ByVal SoundData As Any, ByVal uFlags As Long) As Long
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโมดูล CHAT_Functions (ไฟล์ CHAT_FUN.BAS)

```
Option Explicit
Public Sub InitServerList(ServerList As ComboBox)
  ' Populate Server List Box...
  ServerList.AddItem "127.0.0.1"
End Sub

'-------------------------------------------------------------
Public Sub DebugSocket(TCPSocket As Winsock)
' Prints Information In A TCP Socket, For Debugging TCP Events...
'-------------------------------------------------------------
  Debug.Print "TCPSocket.RemoteHost", TCPSocket.RemoteHost
  Debug.Print "TCPSocket.RemoteHostIP", TCPSocket.RemoteHostIP
  Debug.Print "TCPSocket.RemotePort", TCPSocket.RemotePort
  Debug.Print "TCPSocket.LocalHostName", TCPSocket.LocalHostName
  Debug.Print "TCPSocket.LocalIP", TCPSocket.LocalIP
  Debug.Print "TCPSocket.LocalPort", TCPSocket.LocalPort
  Debug.Print "TCPSocket.State", TCPSocket.State
  Debug.Print "=============================================================="
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------
Public Sub ResPlaySound(ResourceId As Long)
' Uses Sound Play Sound To Play Back PreRecorded WaveFiles
'-------------------------------------------------------------
  Dim sndBuff As String
'-------------------------------------------------------------
  sndBuff = StrConv(LoadResData(ResourceId, "WAVE"), vbUnicode)
  Call sndPlaySound(sndBuff, SND_SYNC Or SND_MEMORY)
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------
Public Sub AddConnectionToList(Socket As Winsock, ConnList As ListBox)
' Adds A Connection Reference To A ListBox - [(Server)(LocalPort)(RemotePort)]
'-------------------------------------------------------------
  Dim MemberID As String                          ' Connection Reference Variable
'-------------------------------------------------------------
  ' Create MemberID From HostName and RemoteIP
  MemberID = Socket.RemoteHostIP & "  [" & _
        Format(Socket.RemotePort, "0") & "] - [" & _
        Format(Socket.LocalPort, "0") & "]"

  ConnList.AddItem MemberID                       ' Add New Member To List
  ConnList.ItemData(ConnList.NewIndex) = Socket.Index
'-------------------------------------------------------------
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'-----------------------------------------------------

'-----------------------------------------------------
Public Sub RemoveConnectionFromList(Socket As Winsock, ConnList As ListBox)
' Removes A Connection Reference From A ListBox
'-----------------------------------------------------
  Dim Conn As Long                                    ' Connection Array Element Variable
  Dim MemberID As String                              ' Connection Reference Variable
'-----------------------------------------------------
  ' Create MemberID From HostName and RemoteIP
  MemberID = Socket.RemoteHostIP & "  [" & _
        Format(Socket.RemotePort, "0") & "] - [" & _
        Format(Socket.LocalPort, "0") & "]"

  For Conn = 0 To ConnList.ListCount - 1              ' Search Each Member In List
    If (ConnList.List(Conn) = MemberID) Then          ' Look For MemberID In List
      ConnList.RemoveItem Conn                        ' Remove MemberID From List
    End If
  Next                                                ' Next Connection
'-----------------------------------------------------
End Sub
'-----------------------------------------------------

'-----------------------------------------------------
Public Sub GetIdxFromMemberID(Sockets As Variant, MemberID As String, Index As Long)
'-----------------------------------------------------
  Dim Idx As Long                                     ' Socket cntl index
  Dim LocPortID As Long                               ' Local Port ID
  Dim RemPortID As Long                               ' Remote Port ID
  Dim RemoteIP As String                              ' Remote Host IP address
  Dim sStart As Long                                  ' Substring begin position
  Dim sEnd As Long                                    ' Substring end postition
  Dim Socket As Winsock                               ' Winsock socket
'-----------------------------------------------------
  sStart = 1
  sEnd = InStr(1, MemberID, " ") - 1                  ' Get end of remote ip address
  If (sEnd > 1) Then
    RemoteIP = Mid(MemberID, sStart, sEnd)            ' Get remote host ip address
    sStart = InStr(sEnd, MemberID, "[") + 1           ' Get start of remote port
    If (sStart > 1) Then                              ' If Start found
      sEnd = InStr(sStart, MemberID, "]") - 1         ' Get end of remote port
      If (sEnd > 2) Then                              ' If end found
        RemPortID = Val(Mid(MemberID, sStart, sEnd))  ' Get RemotePort
        sStart = InStr(sEnd, MemberID, "[") + 1       ' Get start of local port
        If (sStart > 1) Then
          sEnd = InStr(sStart, MemberID, "]") - 1     ' Get end of local port
          If (sEnd > 2) Then                          ' If End Found
            LocPortID = Val(Mid(MemberID, sStart, sEnd))  ' Extract local port
            For Each Socket In Sockets
              If ((Socket.RemoteHostIP = RemoteIP) And _
                (Socket.RemotePort = RemPortID) And _
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                (Socket.LocalPort = LocPortID) And _
                (Socket.Index > 0)) Then                    ' Was a match found???
                Index = Socket.Index                        ' Save and return index
                Exit Sub                                    ' Done... exit
              End If
            Next
          End If
        End If
      End If
    End If
  End If
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโมดูล TCP_Defs (ไฟล์ TCP_DEFS.BAS)

```
Option Explicit

'== TCP Port Array Processing Const.s ==========================
Public Const MINTCP = 1                    ' Minimum index for tcpsocket control instance
Public Const MAXTCP = 32                   ' Maximum index for tcpsocket control instance
Public Const VOICEPORT = 701               ' Voice chat Port To Listen On...
Public Const NULLPORTID = 0                ' Null Port ID - A Port ID That Will Never Be Used...
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโมดูล TCP_Functions (ไฟล์ TCP_FUNC.BAS)

```
Option Explicit
'-----------------------------------------------------------
Public Function InstanceTCP(TCPArray As Variant) As Long
'-----------------------------------------------------------
   Dim Ind As Long                                  ' Array Index Var...
'-----------------------------------------------------------
   InstanceTCP = -1                                 ' Set Default Value
   On Error GoTo InitControl                        ' IF Error Then Control Is Available

   For Ind = MINTCP To MAXTCP                       ' For Each Member In TCPArray() + 1
      If (TCPArray(Ind).Name = "") Then             ' If Control Is Not Valid Then..
      End If                                        ' ..A Runtime Error Will Occure
   Next                                             ' Search Next Item In Array
'-----------------------------------------------------------
InitControl:                                        ' Initialize New Control
'-----------------------------------------------------------
   On Error GoTo ErrorHandler                       ' Enable Error Handling...

   If ((Ind >= MINTCP) And (Ind <= MAXTCP)) Then    ' Check to make sure index value is with in range
      Load TCPArray(Ind)                            ' Create New Member In TCPArray
      InstanceTCP = Ind                             ' Return New TCPctl Index
   End If

   Exit Function                                    ' Exit
'-----------------------------------------------------------
ErrorHandler:                                       ' Handler
'-----------------------------------------------------------
   Debug.Print Err.Number, Err.Description          ' Debug Errors
   Resume Next                                      ' Ignore Error And Continue
'-----------------------------------------------------------
End Function
'-----------------------------------------------------------

'-----------------------------------------------------------
Public Function Connect(Socket As Winsock, RemHost As String, RemPort As Long) As Boolean
' Attempts To Open A Socket Connection
'-----------------------------------------------------------
   Connect = False                                  ' Set default return code
   Call CloseListen(Socket)                         ' Stop Listening On LocalPort
   Socket.LocalPort = 0                             ' Not necessary, but done just in case
   Call Socket.Connect(RemHost, RemPort)            ' Connect To Server

   Do While ((Socket.State = sckConnecting) Or _
         (Socket.State = sckConnectionPending) Or _
         (Socket.State = sckResolvingHost) Or _
         (Socket.State = sckHostResolved) Or _
         (Socket.State = sckOpen))                  ' Attempting To Connect...
      DoEvents                                      ' Post Events
   Loop                                             ' Keep Waiting
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Connect = (Socket.State = sckConnected)                    ' Did Socket Connect On Port...
'-------------------------------------------------------------
End Function
'-------------------------------------------------------------


'-------------------------------------------------------------
Public Function Listen(Socket As Winsock) As Long
' Starts Listening For A Remote Connection Request On The LocalPort ID
'-------------------------------------------------------------
   If (Socket.State <> sckListening) Then                      ' Is Socket Already Listening
      If (Socket.LocalPort = 0) Then                           ' If local port is not initialized then...
         Socket.LocalPort = VOICEPORT                          ' Set standard application port
      End If
      Call Socket.Listen                                       ' Listen On Local Port...
   End If
'-------------------------------------------------------------
End Function
'-------------------------------------------------------------


'-------------------------------------------------------------
Public Function CloseListen(Socket As Winsock) As Long
' Stops Listening On A LocalPort For A Remote Connection Request...
'-------------------------------------------------------------
   If (Socket.State = sckListening) Then                       ' Is Socket Listening?
      Socket.Close                                             ' Close Listen
   End If
'-------------------------------------------------------------
End Function
'-------------------------------------------------------------


'-------------------------------------------------------------
Public Sub Disconnect(Socket As Winsock)
' Disconnects A Remote WinSock TCP/IP Connection If Connected
'-------------------------------------------------------------
   If (Socket.State <> sckClosed) Then                         ' Is Socket Already Closed?
      Socket.Close                                             ' Close Socket
   End If
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. สัจจะ จรัสรุ่งรวีวร “คู่มือการสร้างแอพพลิเคชันด้วย Visual Basic 6 Basic&Advanced”, สำนักพิมพ์อินโฟเพรส , พ.ศ. 2542
2. ธนพล ฉันจรัสวิชัย “การออกแบบและสร้างฐานข้อมูลด้วย Visual Basic 6.0” , บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , พ.ศ. 2543
3. สุวัฒน์ ปุณณชัยยะ , ต้น ตัณฑ์สุทธิวงศ์ , สุพจน์ ปุณณชัยยะ “เปิดโลกของ TCP/IP และโปรโตคอลของอินเตอร์เน็ต” , บริษัท โปรวิชั่น จำกัด , พ.ศ. 2543
4. ประชา ตระการศิลป์ , “การพัฒนาระบบงานไคลเอนต์/เซิร์ฟเวอร์” , ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ , พ.ศ. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้