

การโปรแกรมจับภาพวิดีโอเพื่อสนับสนุนการสอนทางไกล

A DEVELOPMENT OF VIDEO CAPTURE PROGRAM FOR
DISTANCE LEARNING



ประวิทย์ ชีริชิตยงกูร

ประวิทย์ สุดสาคร

ภาสกร สมบุญ

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เลขหมู่.....
เลขทะเบียน..... 39685
วัน, เดือน, ปี..... 19 ส.ย. 2544

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

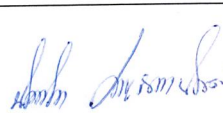

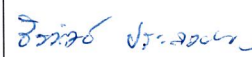

A DEVELOPMENT OF VIDEO CAPTURE PROGRAM
FOR DISTANCE LEARNING

PRAWIT DIRATHITAYANGKUL
PRAWIT SUDDAKORN
PASSKORN SOMBOON

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCES
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2000

หัวข้อปัญหาพิเศษ	การโปรแกรมจับภาพวิดีโอเพื่อสนับสนุนการสอนทางไกล A DEVELOPMENT OF VIDEO CAPTURE PROGRAM FOR DISTANCE LEARNING	
ชื่อนักศึกษา	นายประวิทย์ ชีร์จิตยางกูร	40056045
	นายประวิทย์ สุกสาคร	40056047
	นายภาสกร สมบุญ	40056059
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล	
	ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์	

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2543

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ดร. นันทิกา เบญจเทพานันท์	
กรรมการ	ผู้ช่วยศาสตราจารย์พัชรินทร์ เหมโชติ	
กรรมการและอาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล	
กรรมการและอาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์	



(ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ	การโปรแกรมจับภาพวิดีโอเพื่อสนับสนุนการสอนทางไกล	
ชื่อนักศึกษา	นายประวิทย์ ธีรจิตยางกูร	40056045
	นายประวิทย์ สุธสาคร	40056047
	นายภาสกร สมบุญ	40056059
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2543	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรัักษ์พงษ์	

บทคัดย่อ

ปัจจุบันโลกของการติดต่อสื่อสารทางด้านข้อมูลมีความสำคัญอย่างมากทั้งในการแข่งขันทางธุรกิจ การศึกษา และรวมไปถึงการดำเนินชีวิตประจำวัน เครือข่าย (Network) ก็เป็นเทคโนโลยีที่ถูกนำมาใช้กันอย่างแพร่หลาย เพื่อรองรับการเจริญเติบโตทางด้าน การสื่อสารข้อมูล ทั้งนี้เนื่องจากระบบเครือข่ายคอมพิวเตอร์ที่เชื่อมโยงไปทั่วโลกและเป็นเครือข่ายที่มีผู้ใช้งานมากที่สุดในโลกอีกด้วย และโดยเฉพาะอย่างยิ่งในประเทศไทยการใช้งานระบบเครือข่ายเริ่มขยายตัวอย่างรวดเร็วมีจำนวนผู้ใช้เพิ่มขึ้นเรื่อยๆ ทั้งภาครัฐ และเอกชน ทั้งนี้เนื่องจากบุคคลและหน่วยงานทั้งหลายสามารถใช้ประโยชน์จากระบบเครือข่ายนี้ได้อย่างมหาศาล

โครงการนี้เป็นการเขียนโปรแกรมประยุกต์เพื่อใช้ในการติดต่อสื่อสารมัลติมีเดียเพื่อสนับสนุนการเรียนการสอนโดยผ่านในระบบเครือข่าย โดยจะเป็นการส่งไฟล์ภาพเคลื่อนไหว โดยจะประกอบไปด้วยโปรแกรม 2 ส่วนคือ ส่วน Server ทำหน้าที่จับภาพและทำการตัดเป็นแพคเกจแล้วทำการส่งข้อมูลไป อีกส่วนทำหน้าที่ส่งคำสั่งขอเพื่อไปนำเอาไฟล์ภาพเคลื่อนไหวที่ทำการจับภาพอยู่ที่ตัว server ในขณะนั้นมาแสดงผลในส่วนที่สองเราเรียกส่วนนี้ว่า Client โดยการจัดการการสร้างโปรแกรมประยุกต์โปรแกรมนี้จะใช้โปรแกรม Visual Basic 6.0 เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรม

Special Project Title	A Development of Video Capture Program for Distance Learning	
Sudents	Mr.Prawit Dirathitayangkun	40056045
	Mr.Prawit Sudsakorn	40056047
	Mr.Passkorn Somboon	40056059
Degree	Bachelor 's Degree of Science	
Department	Mathematics and Computer Sciences, Faculty of Science	
Programme	Computer Sciences	
Academic Year	2000	
Special Project Advisor	Assistant Professor Teerawat Phakorbpon	
	Assistant Professor Praiboon Pantaragphong	

ABSTRACT

Currently, Communication technology is very important in business, academics and role of our life. Network system is technology that is distributed to support growth of communication technology because network computer system is associate all around the world and it makes too many people to use. The special reason in Thailand, the network computer system is very fast increment both government and private organization. Many organizations are use network computer system that makes more efficiently communication technology.

This project presents the program for multimedia communication to support distance learning. In this program consists 2 part: 1. Server, it makes to capture animation file (avi file) from digital camera then animation file are decomposed to many package and then send this package to client who ask for request. 2. Client, sends ask for request to server and receive package form server and show this package. This project using Visual Basic 6.0 for program development.

กิตติกรรมประกาศ

ปัญหาพิเศษนี้สำเร็จลงได้ด้วยดีก็เพราะความช่วยเหลือของอาจารย์ทุกท่าน โดยเฉพาะอย่างยิ่ง ผู้ช่วยศาสตราจารย์ธีรวัฒน์ ประกอบผล และ ผู้ช่วยศาสตราจารย์ไพบุลย์ พันธรักษ์พงษ์ ที่ได้ให้แนวทางในการวิจัย ตลอดจนคำปรึกษาอันก่อให้เกิดแนวคิดที่สามารถแก้ปัญหาต่างๆ ที่เกิดขึ้นระหว่างการทำงานวิจัย นอกจากนี้ยังช่วยแนะนำแนวทางในการดำเนินงานและตรวจทานแก้ไขด้วยความเอาใจใส่เป็นอย่างยิ่ง

ขอขอบคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ทุกท่านที่สนับสนุนในการใช้ห้องปฏิบัติการและอุปกรณ์ต่างๆ

ขอขอบคุณเพื่อน ๆ ที่คอยช่วยเหลือให้คำปรึกษาโดยเฉพาะอย่างยิ่งเพื่อนกลุ่มทเวลดเอร์ที่ที่เคยส่งน้ำส่งข้าวและชงน้ำชาให้ดื่มเกือบทุกวัน

ขอขอบคุณ คุณพ่อ คุณแม่ คุณลุง คุณป้า ญาติพี่น้องที่ทำให้กำลังใจและเป็นแรงบันดาลใจเสมอมา

คณะผู้จัดทำ

มีนาคม 2544

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT.....	II
กิตติกรรมประกาศ	III
สารบัญ.....	IV
สารบัญตาราง	IX
สารบัญภาพ	X
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	1
1.3 ขอบเขตการศึกษา	2
1.4 ขั้นตอนขั้นตอนและวิธีการดำเนินการศึกษา.....	2
1.4.1 การศึกษาและรวบรวมความต้องการของระบบงาน และเทคโนโลยีต่าง ๆ ที่ใช้ในการพัฒนา	2
1.4.2 การออกแบบโปรแกรม.....	2
1.4.3 การพัฒนาโปรแกรม.....	2
1.4.4 การทดสอบและปรับปรุงโปรแกรม.....	3
1.4.5 การจัดทำเอกสารประกอบการใช้งาน.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	4
2.1 ระบบเครือข่าย.....	4
2.1.1 ความหมายของระบบเครือข่ายและอินเทอร์เน็ตเวิร์คกิ้ง (Internet Working).....	4
2.1.2 สถาปัตยกรรมของอินเทอร์เน็ต.....	4
2.1.3 ข้อกำหนดรูปแบบของเกตเวย์ (Gateway Protocol).....	5

2.2 โมเดลอ้างอิง OSI	6
2.3 TCP/IP	9
2.3.1 TCP/IP คืออะไร.....	9
2.3.2 ส่วนประกอบ.....	10
2.3.2.1 กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocols).....	10
2.3.2.2 กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocols).....	10
2.3.2.3 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address).....	10
2.3.2.4 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางสื่อสาร ระหว่างเครือข่าย (Gateway Protocols) และสนับสนุนข้อมูลสถานะ	11
2.3.2.5 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Services).....	11
2.3.3 โครงสร้างของชุดโปรโตคอล TCP/IP.....	12
2.3.4 ข้อแตกต่างระหว่างชุดโปรโตคอล TCP/IP	13
2.3.4.1 ลำดับการติดต่อสื่อสารของชั้นเลเยอร์.....	13
2.3.4.2 การติดต่อสื่อสารระหว่างเครือข่ายหรืออินเทอร์เน็ต.....	13
2.3.4.3 การบริการการเชื่อมต่อการสื่อสาร (Connection Service).....	13
2.3.5 ลักษณะของการติดต่อ.....	13
2.3.5.1 Connection-Oriented.....	13
2.3.5.2 Connectionless หรือดาต้าแกรม.....	14
2.3.6 ความสัมพันธ์ของ TCP และ UDP กับอินเทอร์เน็ต.....	14
2.3.7 พอร์ตและซ็อกเก็ต (TCP Port and Socket).....	14
2.3.8 ข้อกำหนดรูปแบบ IP (Internet Protocol).....	15
2.3.9 ข้อกำหนดรูปแบบ ICMP (Internet Control Message Protocol).....	16
2.4 ไอพีแอดเดรส(IP Address).....	16
2.4.1 รหัสเลขประจำเครื่อง IP.....	16
2.5 ซ็อกเก็ต (Socket).....	18
2.5.1 เอพีไอ (API) ข้อกำหนดในการติดต่อสื่อสาร.....	18
2.5.1.1 Socket System Calls.....	18
2.5.1.2 Remote Procedure Call Facilities	18
2.5.2 ชนิดของ Socket	20
2.5.2.1 Stream Socket.....	20

2.5.2.2	Datagram Socket.....	20
2.5.2.3	Row Socket.....	20
2.5.3	Socket Address.....	21
2.5.4	Socket System Call	21
2.5.5	Application Protocol.....	24
2.5.5.1	Connectionless Application Protocol.....	24
2.5.5.1.1	Server System Call.....	24
2.5.5.1.2	Client System Call.....	26
2.5.5.2	Connection-Oriented Application Protocol.....	26
2.5.5.2.1	Server System Call.....	27
2.5.5.2.2	Client System Call.....	28
2.5.6	กระบวนการ System Call พื้นฐาน	29
2.5.6.1	Socket System Call.....	29
2.5.6.2	Bind system Call.....	30
2.5.6.3	Connect System Call.....	30
2.5.6.4	Listen System Call.....	31
2.5.6.5	Accept System Call.....	31
2.5.6.6	Write and Send System Call	31
2.5.6.7	Sendto System Call.....	32
2.5.6.8	Read and Recv System Call.....	32
2.5.6.9	RecvForm System Call	33
2.5.6.10	Close System Call.....	33
2.6	ระบบมัลติมีเดีย.....	34
2.6.1	สื่อที่เป็นองค์ประกอบสำคัญของมัลติมีเดีย.....	34
2.6.1.1	ภาพ.....	34
2.6.1.2	เสียง.....	34
2.6.1.3	การโต้ตอบซึ่งกันและกัน.....	35
2.6.2	ส่วนประกอบของระบบ.....	35
2.6.2.1	ฮาร์ดแวร์มัลติมีเดีย.....	35
2.6.2.2	ซอฟต์แวร์มัลติมีเดีย.....	36

2.6.3	เทคโนโลยีเกี่ยวกับมัลติมีเดีย.....	36
2.6.4	Microsoft Audio Video Interleave.....	37
บทที่ 3	การออกแบบการทำงาน.....	38
3.1	โปรแกรมด้านการส่งไฟล์.....	39
3.2	โปรแกรมด้านการรับไฟล์.....	40
บทที่ 4	การทดลองและพัฒนาโปรแกรม.....	41
4.1	เครื่องมือที่ใช้ในการพัฒนาโปรแกรม.....	41
4.2	ส่วนประกอบต่างๆ ของโปรแกรม.....	42
4.2.1	Server.....	42
4.2.1.1	Frame Commands.....	48
4.2.1.2	Frame Capture Stream Option.....	49
4.2.1.3	Frame Capture.....	50
4.2.1.4	Frame Capture Device Detail.....	50
4.2.2	Client.....	51
4.2.2.1	Frame System Network.....	53
4.2.2.2	ช่องกรอก File Name.....	54
4.2.2.3	ปุ่ม	54
4.2.2.4	ปุ่ม Get Video File.....	54
4.2.3	Menu Control.....	54
4.3	ขั้นตอนการใช้งานโปรแกรม.....	57
4.3.1	การทำงานของ Server.....	57
4.3.2	การทำงานของ Client	59
4.4	ส่วนสนับสนุนการทำงาน.....	60
4.5	ตัวอย่างผลการทดลองจากการทดสอบโปรแกรม.....	61
บทที่ 5	สรุปผลและข้อเสนอแนะ.....	63
5.1	สรุปผล.....	63
5.1.1	ปัญหาต่างๆ ที่พบระหว่างการพัฒนาโปรแกรม.....	63
5.1.2	ข้อจำกัดของตัวโปรแกรม.....	63
5.2	ข้อเสนอแนะ.....	64

บรรณานุกรม.....65

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดง OSI โมเดล.....	7
2.2 แสดงโครงสร้างของชุดโปรโตคอล TCP/IP เปรียบเทียบกับ OSI Model	12
2.3 แสดงหมายเลขพอร์ตที่ให้บริการต่าง ๆ.....	15

สารบัญรูป

รูปที่	หน้า
2.1 แสดงเครือข่ายอินเทอร์เน็ต.....	5
2.2 แสดงคลาสของไอพีแอดเดรส.....	17
2.3 แสดงการเปรียบเทียบระหว่าง Socket และ SUN RPC application programming	19
2.4 แสดงลักษณะทั้งสามแบบของ Socket	20
2.5 แสดงโครงสร้างของข้อมูลที่ใช้ในการกำหนดซ็อกเก็ตแอดเดรส.....	21
2.6 แสดงวิธีการแบบ connectionless application protocol โดยใช้ UDP.....	25
2.7 แสดงวิธีการแบบ Connection-oriented Application Protocol โดยใช้ TCP.....	28
3.1 แสดงการติดต่อกันระหว่าง Server และ Client.....	38
3.2 แสดงการทำงานของโปรแกรมด้าน Server.....	39
3.3 แสดงการทำงานของโปรแกรมด้าน Client.....	40
4.1 โพลีชาร์ทของ Server.....	45
4.2 โพลีชาร์ท ของ โปรแกรมย่อย runfile.....	46
4.3 แสดงหน้าจอหลัก.....	47
4.4 แสดงหน้าจอของ Server.....	48
4.5 แสดง Frame Commands.....	49
4.6 แสดง Frame Capture Stream Option.....	49
4.7 แสดง Frame Capture.....	50
4.8 แสดง Frame Detail.....	50
4.9 แสดง Frame Status.....	50
4.10 Flow Chart ของ Client.....	52
4.11 แสดงหน้าจอส่วนการทำงานของ Client.....	53
4.12 แสดง Frame Network System.....	54
4.13 แสดงส่วนควบคุมการเล่นไฟล์ภาพ.....	54
4.14 แสดง Menu Bar.....	55
4.15 แสดง File Menu.....	55
4.16 แสดง Status Menu.....	56

4.17 แสดง Video Menu.....	56
4.18 แสดง Tool Menu.....	56
4.19 แสดง About Menu.....	57
4.20 แสดงหน้าจอหลังทำการติดต่อกับกล้องดิจิทัล.....	58
4.21 แสดงหน้าจอขณะทำการ Capture.....	59
4.22 แสดง Status เมื่อ End.....	59
4.23 แสดงหน้าจอเมื่อได้รับไฟล์มาก็จะทำการเล่น.....	60
4.24 แสดง Server-U.....	61
4.25 แสดงการเปรียบเทียบหน้าจอ Server กับ Client ในขณะเวลาเดียวกัน.....	62

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันโลกของการติดต่อสื่อสารทางด้านข้อมูลมีความสำคัญอย่างมากทั้งในการแข่งขันทางธุรกิจ การศึกษา และรวมไปถึงการดำเนินชีวิตประจำวัน เครือข่าย (Network) ก็เป็นเทคโนโลยีที่ถูกนำมาใช้กันอย่างแพร่หลาย เพื่อรองรับการเจริญเติบโตทางด้าน การสื่อสารข้อมูล ทั้งนี้เนื่องจากระบบเครือข่ายคอมพิวเตอร์ที่เชื่อมโยงไปทั่วโลกและเป็นเครือข่ายที่มีผู้ใช้งานมากที่สุดในโลกอีกด้วย และโดยเฉพาะอย่างยิ่งในประเทศไทยการใช้งานระบบเครือข่ายเริ่มขยายตัวอย่างรวดเร็วมีจำนวนผู้ใช้เพิ่มขึ้นเรื่อยๆ ทั้งภาครัฐ และเอกชน ทั้งนี้เนื่องจากบุคคลและหน่วยงานทั้งหลายสามารถใช้ประโยชน์จากระบบเครือข่ายนี้ได้อย่างมหาศาล

เราจึงสามารถกล่าวได้ว่า เครือข่ายเป็นเทคโนโลยีทางด้านเครือข่ายที่ช่วยอำนวยความสะดวกและสื่อสารได้อย่างรวดเร็ว ทำให้ข้อมูลข่าวสารที่เกิดขึ้นอยู่ทั่วโลกสามารถรับรู้ได้ทั่วถึงกัน โดยไม่มีพรมแดนเสมือนย่อโลกให้เล็กลง นับว่าเป็นประโยชน์อย่างมากสำหรับยุคข้อมูลข่าวสารในปัจจุบัน การสื่อสารบนเครือข่ายนั้นมีการพัฒนาไปอย่างมาก ทำให้มีประสิทธิภาพในการทำงานสูงขึ้น ดังนั้นการที่เครื่องคอมพิวเตอร์จะติดต่อกับอีกเครื่องจึงเป็นไปได้ จึงได้มีการพัฒนาโปรแกรมต่างๆขึ้นมาเพื่อเพิ่มความสะดวกสบายในการใช้งาน ทั้งยังสามารถให้ประโยชน์ในด้านการศึกษาผ่านระบบเครือข่ายด้วย การคิดค้นที่จะทำการสร้างโปรแกรมส่งภาพวิดีโอที่เป็นไฟล์ภาพเคลื่อนไหวผ่านระบบไปยังส่วนต่างๆที่ทำการร้องขอจึงเกิดขึ้น เพื่อความสามารถในการนำข้อมูลจากต้นทางไปยังปลายทางที่ต้องการได้

เนื่องจากในปัจจุบันมีปัญหาทางการศึกษาเกิดขึ้นเพราะหลาย ๆ สาเหตุ เช่น จำนวนอาจารย์ผู้สอนมีอยู่อย่างจำกัด ปัญหาในการเดินทางของอาจารย์ผู้สอนและนักเรียน รวมทั้งอาจารย์ผู้สอนและนักเรียนต้องอยู่ในสถานที่เดียวกันและเวลาเดียวกันเท่านั้นจึงจะสามารถทำการสอนได้ ฯลฯ ด้วยเหตุผลหลายประการทำให้เกิดความคิดที่จะนำเอาเทคโนโลยีระบบเครือข่ายเข้ามาช่วยในเรื่องการกระจายภาพวิดีโอการสอน ณ ขณะนั้นไปยังสถานที่ต่าง ๆ ที่ผู้เรียนมีความสะดวกในการที่จะเข้าฟังการเรียนการสอนนั้นได้โดยอาศัยเทคโนโลยีเข้ามาช่วย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อเพิ่มประสิทธิภาพของการเรียนการสอนโดยอาจารย์ผู้สอนสามารถทำการสอนโดยมีการถ่ายทอดภาพการเรียนการสอนในขณะนั้นผ่านทางระบบเครือข่าย เพื่อที่จะให้นักเรียนที่ต้องการ

เข้ารับการสอนในรายวิชานั้น ๆ สามารถเข้าเรียนโดยไม่ต้องอยู่ในห้องเรียน โดยอาศัยเทคโนโลยีเข้ามาช่วย

1.3 ขอบเขตการศึกษา

ในปัญหาพิเศษนี้จะทำการสร้างโปรแกรมโดยสามารถแบ่งการทำงานออกได้เป็น 2 ส่วน คือ

1. โปรแกรมที่ทำการจับภาพการสอนผ่านทางกล้องดิจิตอลมาทำการเก็บแล้วตัดเป็นแพ็คเกจเพื่อรอรับการร้องขอแล้วทำการส่งไป (จะเป็นตัวที่อยู่ในห้องเรียนที่อาจารย์ทำการสอน)
2. ส่วนที่สองจะเป็นส่วนที่นักเรียนจะทำการร้องขอไฟล์วีดิโอการสอนที่ได้จากโปรแกรมส่วนที่ 1 โดยโปรแกรมส่วนที่สองนี้ต้องอยู่ในระบบเครือข่ายเดียวกับส่วนแรกด้วย และเมื่อได้ไฟล์วีดิโอการสอนแล้วจึงทำการแสดงผล

ในการพัฒนาโปรแกรมนั้นได้พัฒนาโดยใช้โปรแกรม Visual Basic 6.0 โดยที่โปรแกรมส่วนแรกจะทำการจับภาพการสอนที่อาจารย์กำลังสอนในขณะนั้นนำมาตัดเป็นแพ็คเกจเพื่อรอการร้องขอของส่วนโปรแกรมที่สอง แล้วโปรแกรมในส่วนแรกจะทำการส่งไฟล์ภาพของแพ็คเกจล่าสุดไปยังตัวโปรแกรมในส่วนที่สองทำการแสดงผลออกมา

1.4 ขั้นตอนและวิธีการดำเนินการศึกษา การพัฒนาโปรแกรมโดยผ่านระบบเครือข่ายมีหลักการทำงานดังนี้

1.4.1 การศึกษาและรวบรวมความต้องการของระบบงานทั้งเทคโนโลยีต่างๆที่ใช้ในการพัฒนา

- 1) ศึกษาการทำงานภายในระบบเครือข่าย
- 2) ศึกษาการเขียนโปรแกรมเกี่ยวกับการติดต่อไฟล์ภาพเคลื่อนไหวโดยใช้ Visual Basic 6.0
- 3) ศึกษาเรื่องการจัดการการติดต่อผ่าน Socket, IP address
- 4) ศึกษาโปรแกรมที่มีอยู่ในปัจจุบัน

1.4.2 การออกแบบโปรแกรม

- 1) ออกแบบโปรแกรม
- 2) ออกแบบรูปแบบการแสดงผล
- 3) ออกแบบโครงสร้างของโปรแกรม

1.4.3 การพัฒนาโปรแกรม

ทำการเขียนโปรแกรมตามรูปแบบและโครงสร้างที่วางไว้ในขั้นตอนการออกแบบโปรแกรม

1.4.4 การทดสอบและปรับปรุงโปรแกรม

ทำการทดสอบโปรแกรมและพยายามเพิ่มขีดความสามารถของโปรแกรมให้มีประสิทธิภาพดียิ่งขึ้น

1.4.5 การจัดทำเอกสารประกอบการใช้งาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 เพื่อเป็นโปรแกรมต้นแบบสำหรับการพัฒนาสื่อการสอนผ่านระบบเครือข่าย

1.5.2 เพื่อลดปัญหาจำนวนอาจารย์ผู้สอนที่มีไม่เพียงพอต่อความต้องการ

บทที่ 2

ทฤษฎีหรือหลักการ

2.1 ระบบเครือข่าย

การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายจำเป็นต้องมีความรู้พื้นฐานทางด้านระบบเครือข่ายพอสมควร ในหัวข้อนี้จะอธิบายหลักการของระบบเครือข่ายที่จำเป็นสำหรับการทำความเข้าใจในการพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายดังต่อไปนี้

2.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์ และอินเทอร์เน็ตเวิร์คกิ้ง(Internet Working)

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) คือ ระบบการเชื่อมต่อระหว่างระบบปลายทาง (End-System) ซึ่งระบบปลายทางเป็นระบบอิสระจากกัน (Autonomous) ระบบปลายทางสามารถเป็นได้ตั้งแต่ ไมโครคอมพิวเตอร์ (Microcomputer) ไปจนกระทั่งซูเปอร์คอมพิวเตอร์ (Supercomputer) ขนาดใหญ่ เพื่อจุดมุ่งหมายในการแลกเปลี่ยนข้อมูล และการแบ่งข้อมูลในการแบ่งปันทรัพยากรของระบบ เช่น ฟิล์ข้อมูล, เครื่องพิมพ์ (Printer), โมเด็ม (Modem) ตลอดจนการให้บริการฐานข้อมูลร่วม (Sharing database)

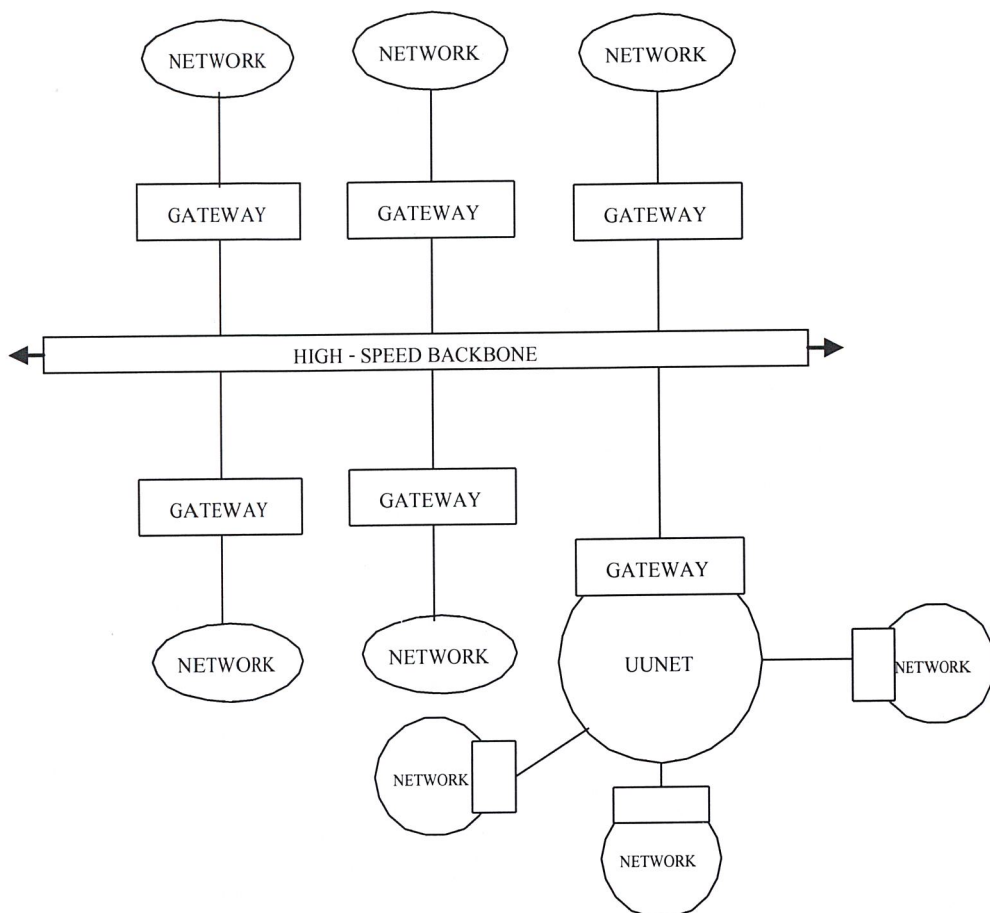
อินเทอร์เน็ตเวิร์คกิ้ง หรือ อินเทอร์เน็ต (Internet) คือ การเชื่อมต่อระบบเครือข่าย 2 เครือข่ายขึ้นไป ดังนั้นคอมพิวเตอร์บนระบบเครือข่ายหนึ่งก็สามารถติดต่อกับคอมพิวเตอร์บนระบบเครือข่ายอื่นๆได้

2.2.2 สถาปัตยกรรมของอินเทอร์เน็ต

คำกล่าวถึงอินเทอร์เน็ตว่าเป็นเครือข่ายที่ใหญ่ที่สุดในโลก เครือข่ายที่มีการต่อเสมือนกับใยแมงมุมครอบคลุมโลก อินเทอร์เน็ตเชื่อมโยงโลกเข้าด้วยกันอย่างไร้มิติ (Cyberspace) จากคำกล่าวเหล่านี้ทำให้เราควรรู้ว่าอินเทอร์เน็ตจัดการส่วนประกอบต่างๆอย่างไร จึงทำให้แต่ละส่วนสามารถรับ-ส่งข้อมูลและทำงานสัมพันธ์กันดีเป็นอย่างไร เราจึงจำเป็นต้องศึกษาสถาปัตยกรรมของอินเทอร์เน็ตดังรูปที่ 2.1

เครือข่ายที่ต้องการเชื่อมโยงโดยตรงกับอินเทอร์เน็ตจะต่อกับแบคโบน ด้วยอุปกรณ์ที่เรียกว่า “เกตเวย์” (Gateway) ซึ่งเป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่ควบคุมการผ่านเข้า-ออกของข่าวสารระหว่างเครือข่ายกับแบค-โบน เกตเวย์ทุกตัวสามารถกำหนดการติดต่อกับเกตเวย์ตัวอื่น หรือเครือข่ายอื่นได้ โดยใช้ไอพีแอดเดรส (IP Address) ของเครือข่าย อ้างอิงถึงกัน เช่น อินเทอร์เน็ตประเทศไทยเป็นเกตเวย์ของเครือข่ายในกลุ่มติดต่อกับเกตเวย์ของ UUNET ที่รัฐเวอร์จิเนียประเทศสหรัฐอเมริกา

ข่าวสารจากเครือข่ายถูกส่งออกไปผ่านเกตเวย์เข้าสู่อินเทอร์เน็ต โดยที่เกตเวย์เป็นตัวเลือกทิศทางการเดินทางเพื่อไปยังปลายทางที่ต้องการ แต่ตามเส้นทางอาจจะต้องผ่านเกตเวย์อีกหลายตัวเพื่อรับช่วงส่งข่าวสารจนถึงที่หมาย ถึงแม้ว่าจะต้องเดินทางระยะไกลก็ตาม แต่ด้วยสายสื่อสารความเร็วสูงทำให้การส่งข่าวสารทำได้อย่างรวดเร็ว



รูปที่ 2.1 แสดงเครือข่ายอินเทอร์เน็ต

2.1.3 ข้อกำหนดรูปแบบของเกตเวย์ (Gateway Protocols)

เกตเวย์ต้องมีข้อมูลของเกตเวย์ตัวอื่นและรู้จักเครือข่ายปลายทาง เพื่อนำมาใช้ในการกำหนดเส้นทางที่ข่าวสารสามารถเดินทางไปถึงได้เร็วที่สุด เกตเวย์จะมีการแลกเปลี่ยนข้อมูลระหว่างกันเกี่ยวกับเส้นทาง รายละเอียดสถานะเครือข่าย และคุณสมบัติเครือข่ายย่อยที่ติดต่อเข้าสู่เครือข่ายใหญ่ตามลำดับชั้น จึงต้องมีการกำหนดรูปแบบพิเศษสำหรับเกตเวย์ขึ้นมา

ข้อกำหนดรูปแบบเกตเวย์แบ่งออกตามการใช้งาน ได้ดังนี้

1) IGP (Interior Gateway Protocol) ถูกนำไปใช้กับเกตเวย์ที่อยู่ในเครือข่ายลูก ติดต่อกับเครือข่ายลูกที่อยู่ภายในเครือข่ายแม่เดียวกัน หรือติดต่อกันระหว่างเครือข่ายแม่กับเครือข่ายลูก การ

เชื่อมต่อโยงเกตเวย์ประเภทนี้มักจะไม่ค่อยมีการเปลี่ยนแปลง จึงเรียกว่า ระบบอิสระจากกัน (Autonomous) หรือ Selfcomplete

2) EGP (External Gateway Protocol) ในเครือข่ายใหญ่ๆ การติดต่อกับเครือข่ายอื่นจะมีการเปลี่ยนแปลงอยู่เสมอ ตามการเปลี่ยนแปลงของเครือข่ายย่อยที่เชื่อมต่ออยู่เป็นจำนวนมาก จึงมีข้อกำหนดรูปแบบที่ใช้กับการสื่อสารระหว่างเกตเวย์ของเครือข่ายด้วย EGP

3) GGP (Gateway-to-Gateway Protocol) การเดินทางของข่าวสารระยะไกลบนแบคโบน อาจจะต้องผ่านเกตเวย์หลายตัวว่าจะถึงปลายทาง GGP เป็นข้อกำหนดรูปแบบของการสื่อสารระหว่างเกตเวย์บนแบคโบนเพื่อทำให้การจราจรบนแบคโบนไม่ติดขัด ข่าวสารเคลื่อนที่ไปได้รวดเร็ว

2.2 โมเดลอ้างอิง OSI

เพื่อลดปัญหาความยุ่งยากซับซ้อนในการติดต่อสื่อสารข้อมูล โครงสร้างของการสื่อสารข้อมูลภายในอุปกรณ์คอมพิวเตอร์ส่วนใหญ่จะถูกแบ่งเป็นชั้นๆ โดยแต่ละชั้นมีอิสระไม่ขึ้นต่อกันทำให้การเปลี่ยนแปลงบริการของชั้นใดๆ ไม่ก่อปัญหาต่อบริการชั้นอื่น การเพิ่มเติมการบริการใหม่ทำได้โดยไม่ต้องเปลี่ยนแปลงโปรแกรมระบบเดิมและสิ่งสำคัญก็คือ การทำระดับชั้นนั้นทำให้ตัวโปรแกรมมีขนาดเล็ก สามารถระบุส่วนที่จะต้องปรับปรุงได้แน่นอน ไม่ต้องวิตกกังวลถึงโปรแกรมส่วนอื่น ทำให้การพัฒนาประสิทธิภาพของระบบทำได้ง่ายและดียิ่งขึ้น

ISO (International Standardization for Organization) ซึ่งเป็นองค์กรที่จัดตั้งขึ้นมาเพื่อดูแลและส่งเสริมตลอดจนกำหนดมาตรฐานของการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์ ได้พัฒนาสถาปัตยกรรมข้อกำหนดรูปแบบสำหรับเครือข่าย ซึ่งเป็นลักษณะระบบเปิดที่เรียกว่า 'Open System Interconnection Model' (OSI) โดยโมเดล OSI นี้มีลักษณะเป็นสถาปัตยกรรมแบบระบบเปิด (Open System) เพราะมุ่งที่จะให้ระบบคอมพิวเตอร์ในหลายๆรูปแบบที่แตกต่างกันสามารถเชื่อมต่อกันได้ OSI โมเดลได้แบ่งโปรโตคอล (Protocol) ในการสื่อสารออกเป็น 7 เลเยอร์ (Layer) ซึ่งโปรโตคอล คือชุดหรือข้อตกลงในการติดต่อ

ตาราง 2.1 OSI โมเดล

ระดับชั้น	ชื่อระดับชั้น	ระดับหน้าที่
7	Application	Upper Layers
6	Presentation	
5	Session	
4	Transport	Lowery Layers
3	Network	
2	Data Link	
1	Physical	

ข้อกำหนดรูปแบบแต่ละระดับชั้นของ OSI มีการระบุหน้าที่ต้องปฏิบัติแน่นอน โดยจะต้องเป็นตามกลุ่มการบริการ ดังนี้

1) ระดับชั้น 7 Application Layer เป็นเลเยอร์ชั้นบนสุดของรูปแบบ OSI ซึ่งเป็นชั้นที่ติดต่อกันระหว่างผู้ใช้โดยตรงซึ่งได้แก่ โฮสต์คอมพิวเตอร์ เทอร์มินัลหรือคอมพิวเตอร์ส่วนบุคคล (PC) เป็นต้น แอปพลิเคชันในเลเยอร์นี้ สามารถนำเข้าหรือออกจากระบบเครือข่ายได้โดยไม่ต้องสนใจจะมีขั้นตอนการทำงานอย่างไร เพราะจะมีชั้นพีรีเซนเตชันเป็นผู้รับผิดชอบอยู่แล้ว ในรูปแบบ OSI นี้ชั้นแอปพลิเคชันจะมีการติดต่อกับชั้นพีรีเซนเตชันโดยตรง

2) ระดับชั้น 6 Presentation Layer คอยรวบรวมข้อความและแปลงรหัสหรือแปลงรูปแบบของข้อมูลที่ได้รับมาจากระดับชั้น 7 เพื่อให้เป็นรูปแบบการสื่อสารเดียวกัน เพื่อช่วยลบลปัญหาต่างๆที่อาจเกิดขึ้นกับผู้ใช้งานในระบบ

3) ระดับชั้น 5 Session Layer ทำหน้าที่เชื่อมโยงระหว่างผู้ใช้คอมพิวเตอร์เครื่องอื่นๆโดยผู้ใช้จะใช้คำสั่งหรือข้อความที่กำหนดไว้ป้อนเข้าไปในระบบ ในการสร้างการเชื่อมโยงนี้ผู้ใช้จะต้องกำหนดรหัสตำแหน่งของจุดหมายปลายทางที่ต้องการติดต่อสื่อสารด้วย ในบางครั้งเครือข่ายอาจมีชั้นเซสชันและชั้นทรานสปอร์ตรวมเป็นชั้นเลเยอร์ชั้นเดียวกัน

4) ระดับชั้น 4 Transport Layer บางครั้งเรียกว่าชั้น เครื่องต่อเครื่อง (Host-to-Host) และจากเลเยอร์ชั้นที่ 4 ถึงชั้นที่ 7 นี้เรียกรวมกันว่า เลเยอร์เอนทูเอน (End-to-End) ในเลเยอร์ชั้นทรานสปอร์ต (Transport) นี้จะเป็นการสื่อสารระหว่างต้นทางและปลายทางกันจริงๆสามารถแบ่งการบริการเป็น 2 กรณีคือ

กรณีหนึ่ง ส่งข้อมูลของระดับชั้น 5 โดยนำมาใส่ข้อมูลของปลายทาง เช่น ที่อยู่แล้วส่งออกไปโดยแบ่งข่าวสารยาวๆให้เป็นหน่วยที่เล็กลง เรียกว่าแพคเกจ (Packet) หรือเฟรม (Frame)

กรณีที่สอง เป็นบริการที่ทำการถอดรหัสข้อมูลที่รับเข้ามาพิจารณาตัดสินว่ามีความสัมพันธ์เกี่ยวข้องกับข้อมูลที่ส่งออกไปหรือไม่ อย่างไร

เลเยอร์ชั้น ทรานสปอร์ต จะทำหน้าที่ตรวจสอบว่าข้อมูลที่ส่งมาจากชั้นเลเยอร์ที่ 5 นั้น ไปถึงปลายทางจริงหรือไม่ ดังนั้นการกำหนดตำแหน่งของข้อมูลจึงสำคัญในชั้นนี้

5) ระดับชั้น 3 Network Layer เป็นชั้นที่ออกแบบหรือกำหนดเส้นทางการเดินทางของข้อมูลที่ส่ง-รับในการส่งผ่านข้อมูลระหว่างต้นทางและปลายทาง ซึ่งเป็นที่แน่นอนว่าในการสื่อสารข้อมูลผ่านเครือข่ายการสื่อสารจะต้องมีเส้นทางการรับ-ส่ง ข้อมูลมากกว่า 1 เส้นทาง ดังนั้นในชั้นนี้จะมีหน้าที่เลือกเส้นทางที่ใช้เวลาในการสื่อสาร น้อยที่สุดและระยะทางสั้นที่สุดด้วย

6) ระดับชั้น 2 Data Link Layer เป็นเสมือนผู้ควบคุมการรวมบิตเข้าเป็นตัวอักษรและจัดข้อมูลให้เป็นแพ็คเกจ รวมทั้งตรวจสอบความผิดพลาดในการส่งข้อมูล ถ้าผู้รับได้รับข้อมูลถูกต้องก็จะส่งสัญญาณยืนยันกลับมาว่าได้รับข้อมูลแล้ว เรียกว่า สัญญาณ ACK (Acknowledge) ให้กับผู้ส่ง แต่ถ้าผู้ส่งไม่ได้รับสัญญาณ ACK หรือได้รับสัญญาณ NAK (Negative Acknowledge) กลับมา ผู้ส่งอาจจะทำการส่งข้อมูลไปให้ใหม่ อีกหน้าที่หนึ่งของชั้นนี้ คือ ป้องกันไม่ให้เครื่องส่งทำการส่งข้อมูลเร็วจนเกินขีดความสามารถของเครื่องผู้รับข้อมูลจะรับได้

7) ระดับชั้น 1 Physical Layer เป็นระดับชั้นการติดต่อระหว่างอุปกรณ์กับสื่อกลางของเครือข่ายจริงๆ การส่งข้อมูลระดับชั้นนี้ข้อมูลมีลักษณะเป็นบิต สื่อกลางที่ใช้ อาจจะเป็นสายโทรศัพท์ไฟเบอร์ออปติก ไมโครเวฟหรือ ดาวเทียม ตามระบบที่ใช้ โดยมาตรฐานสำหรับเลเยอร์ชั้นนี้จะกำหนดว่าแต่ละคอนเนคเตอร์ เช่น RS-232-C มีกี่พิน แต่ละพินทำหน้าที่อะไรบ้าง ใช้สัญญาณไฟกี่โวลต์ เทคนิคการมัลติเพล็กซ์แบบต่างๆ ก็จะถูกกำหนดอยู่ในชั้นเลเยอร์นี้

ในการสื่อสารกันระหว่างระบบหรือระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ โปรโตคอลแบบหนึ่งก็จะถูกใช้

สำหรับการสื่อสารระหว่างชั้น N ของคอมพิวเตอร์เครื่องหนึ่งกับชั้น N เดียวกันของคอมพิวเตอร์อีกเครื่องหนึ่ง ซึ่งในระหว่างการสื่อสารข้อมูลกันจริงๆ นั้น โปรโตคอลชั้น N ของอุปกรณ์ทั้งสองเครื่องจะสื่อสารกันผ่านชั้นล่างซึ่งเป็นกายภาพหรือวัตถุ (สื่อกลางการสื่อสาร) แต่โดยแนวความคิดของการสื่อสารข้อมูลแล้วถือว่าชั้นที่ N ของเครื่องหนึ่งกำลังติดต่อโดยตรงกับชั้น N ของอีกเครื่องหนึ่ง ซึ่งเรียกรูปแบบนี้ว่า การสื่อสารแบบเสมือนจริง (Virtual Communication)

โปรโตคอลของในเลเยอร์แต่ละชั้นจะแตกต่างกันออกไป แต่อย่างไรก็ตามการที่เครื่องคอมพิวเตอร์หลายๆเครื่องจะติดต่อสื่อสารกันได้ในแต่ละเลเยอร์ของแต่ละเครื่องจะต้องใช้โปรโตคอลแบบเดียวกัน หรือถ้าใช้โปรโตคอลต่างชนิดกันก็ต้องมีอุปกรณ์ หรือซอฟต์แวร์ที่สามารถแปลงโปรโตคอลที่ต่างกันนั้นให้มีรูปแบบอย่างเดียวกันเพื่อเชื่อมโยงให้คอมพิวเตอร์ทั้ง 2 เครื่องสามารถติดต่อกันได้

ในหนึ่งชั้นของเลเยอร์ไม่ได้กำหนดว่าจะต้องมีเพียงหนึ่งโปรโตคอลเท่านั้นที่อยู่ระดับเลเยอร์เดียวกันและในทางตรงข้าม ชุดของโปรโตคอลใดๆอาจจะมีมากกว่าหนึ่งเลเยอร์ประกอบกันเป็นข้อกำหนดของระบบเครือข่ายเรียกว่าชุดโปรโตคอล (Protocol Suit) เช่น ชุดโปรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นต้น ประโยชน์ในการแบ่งเป็นเลเยอร์ คือ กำหนดการติดต่อระหว่างเลเยอร์ทำโดยไม่ต้องคำนึงถึงการเปลี่ยนแปลงในเลเยอร์ใดๆที่ติดกัน

2.3 TCP/IP

อินเทอร์เน็ตนับได้ว่าเป็นเครือข่ายบนเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่นเชื่อมโยงเข้ามาใช้งาน หรือเป็นศูนย์กลางเชื่อมโยงเครือข่ายคอมพิวเตอร์อื่นๆ แต่ปัญหาที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันก็คือ แต่ละเครือข่ายใช้คอมพิวเตอร์ต่างชนิด ต่างยี่ห้อ และระบบปฏิบัติการที่ต่างกัน มาตรฐานของ TCP/IP จึงถูกใช้เป็นกุญแจสำคัญในการแก้ไขปัญหาเหล่านี้ โดยกลายเป็นระบบเปิดที่สมบูรณ์แบบที่มีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่เครื่องพีซี จนถึงเครื่องเมนเฟรม และไม่จำกัดระบบปฏิบัติการที่ใช้ TCP/IP จึงเป็นมาตรฐานที่ทั่วโลกยอมรับ มีอุปกรณ์และซอฟต์แวร์ผลิตออกมาสับสนุน TCP/IP มากมาย ดังนั้นจึงนับได้ว่า TCP/IP เป็นหัวใจของอินเทอร์เน็ตเลยทีเดียว

2.3.1 TCP/IP คืออะไร

TCP/IP เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Networking Protocol) จัดทำเพื่อใช้เป็นกฎเกณฑ์ให้เครื่องคอมพิวเตอร์ใช้งานร่วมกัน ในลักษณะของระบบเปิด (Open System) คือไม่ว่าจะเป็นคอมพิวเตอร์ชนิดใดหรือระบบใดก็ตาม จะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันได้

TCP/IP เป็นการกำหนดรูปแบบการสื่อสารระหว่างซอฟต์แวร์ การจัดการโอนย้ายข้อมูล การแสดงสถานะของเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย ตลอดจนกฎระเบียบต่างๆที่กำหนดให้ทำเมื่อเกิดความผิดพลาด หรือต้องทำเพื่อป้องกันไม่ให้เกิดความผิดพลาด

TCP/IP เกิดจากการนำข้อกำหนดของรูปแบบต่างๆกันมาใช้ร่วมกัน TCP และ IP ต่างก็เป็นรูปแบบหนึ่งของชุดข้อกำหนดนี้ (แต่เรียกชุดข้อกำหนดรูปแบบนี้ว่า TCP/IP) ถูกออกแบบมาเพื่อใช้รับส่งหรือโอนย้ายข้อมูลระหว่างคอมพิวเตอร์ที่อยู่บนระบบเครือข่ายเดียวกัน หรือต่างเครือข่ายกันก็ได้ และมีการจัดเตรียมข้อมูลสถานะของเครือข่ายขึ้นได้ภายในตัวข้อกำหนดรูปแบบเอง ในการสร้างซอฟต์แวร์ของระบบเครือข่ายจะใช้ TCP/IP เป็นส่วนสนับสนุนได้ทั้งระบบเครือข่ายเฉพาะบริเวณ (Local Area Network) และเครือข่ายบริเวณกว้าง (Wide Area Network) ไม่ได้ใช้งานเฉพาะกับอินเทอร์เน็ตเท่านั้น

2.3.2 ส่วนประกอบ

จากที่กล่าวมาแล้วข้างต้นว่า TCP/IP ประกอบไปด้วยชุดข้อกำหนดรูปแบบต่างๆซึ่งแบ่งเป็นกลุ่มได้ดังนี้

2.3.2.1 กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocols) ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูลระหว่างคอมพิวเตอร์สองเครื่อง แบ่งย่อยออกได้เป็นสองชนิด คือ

1) TCP (Transmission Control Protocol) เป็นการบริการแบบ Connection Based Service ซึ่งคอมพิวเตอร์ด้านผู้รับและผู้ส่งต้องต่อกันอยู่ตลอดเวลาในระหว่างการสื่อสารถ้าเปรียบเทียบกับคล้ายกับระบบโทรศัพท์ที่ต้องติดต่อกันให้ได้ก่อนจะพูดคุยกันได้

2) UDP (User Datagram Protocol) เป็นการให้บริการแบบ Connection Service คอมพิวเตอร์ด้านผู้ส่งไม่จำเป็นต้องติดต่อกับด้านผู้รับก่อน เพียงรู้ที่อยู่ของด้านผู้รับแล้วใส่ที่อยู่นั้นไปกับข้อมูลที่ส่งออก ข้อมูลจะเดินทางตามเส้นทางต่างๆเพื่อไปถึงปลายทางตามที่อยู่ คล้ายกับการส่งจดหมายที่ไปรษณีย์ จะส่งให้ตามที่อยู่ที่กำหนดของจดหมาย โดยผู้ส่งและผู้รับไม่ต้องติดต่อกัน

2.3.2.2 กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocol) ทำหน้าที่พิจารณาเส้นทางที่ดีที่สุดที่ใช้ส่งข้อมูลและถ้ามีข้อมูลเป็นจำนวนมากหรือมีขนาดใหญ่ กลุ่มข้อมูลรูปแบบนี้ก็จะทำการแบ่งย่อยข้อมูลให้มีขนาดเหมาะสมแล้วส่งออกไป เมื่อถึงผู้รับปลายทาง กลุ่มข้อมูลนี้ก็จะทำหน้าที่ตรงข้าม คือรวบรวมข้อมูลย่อยให้ถูกต้องก่อนการแสดงผล กลุ่มข้อกำหนดรูปแบบกลุ่มนี้ ประกอบด้วย

- 1) IP (Internet Protocol) เป็นการกำหนดรูปแบบการส่งข้อมูล
- 2) ICMP (Internet Control Message Protocol) เป็นข้อกำหนดรูปแบบของข้อมูลเกี่ยวกับสถานะของ IP เช่นข่าวสารความผิดพลาดและผลกระทบต่อเส้นทางเมื่อมีการเปลี่ยนแปลงฮาร์ดแวร์ในเครือข่าย
- 3) RIP (Routing Information Protocol) ข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่ดีที่สุด เหมาะสมกับข้อมูลมากที่สุด
- 4) OSPF (Open Shortest Path First) ข้อกำหนดรูปแบบอีกประเภทหนึ่งที่ใช้ตัดสินเลือกเส้นทาง โดยพิจารณาจากเส้นทางที่สั้นที่สุดก่อน

2.3.2.3 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address) ทำหน้าที่พิจารณาที่อยู่ของเครือข่ายและเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นลักษณะตัวเลขหรือชื่อก็ตาม เพื่อความถูกต้องของข้อมูลที่จะไปยังผู้รับปลายทาง โดยที่ไม่ว่าเครือข่ายจะใหญ่โตสักเพียงใดหรือมีเครื่องคอมพิวเตอร์จำนวนมากก็ตาม ที่อยู่จะต้องไม่ซ้ำกัน กลุ่มข้อกำหนดรูปแบบ กลุ่มนี้มีดังนี้

1) ARP (Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เพื่อไม่ให้เกิดที่อยู่ซ้ำกัน

2) DNS (Domain Name System) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เมื่อรู้ชื่อของเครือข่ายหรือเครื่องคอมพิวเตอร์ เพราะการใช้งานจริงนั้นใช้เพียงที่อยู่ที่เป็นตัวเลข แต่ระบบชื่อจัดทำขึ้นเพื่อให้สะดวกต่อการใช้งานของผู้ใช้

3) RARP (Reverse Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เช่นเดียวกับARP แต่จะทำตรงข้ามกับ ARP

2.3.2.4 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางการสื่อสารระหว่างเครือข่าย (Gateway Protocols) และสนับสนุนข้อมูลสถานะ เพื่อนำไปใช้เลือกเส้นทางที่เหมาะสม ข้อกำหนดรูปแบบเหล่านี้ประกอบด้วย

1) EGP (Exterior Gateway Protocol) ข้อกำหนดรูปแบบนี้จะทำการถ่ายโอนข้อมูลเส้นทางกันระหว่างเกตเวย์ กับเครือข่ายภายนอกเพื่อทำการสื่อสาร

2) GGP (Gateway-to-Gateway Protocol) เป็นข้อกำหนดรูปแบบที่ทำงานถ่ายโอนข้อมูลเส้นทางกันระหว่างเกตเวย์กับเกตเวย์

3) IGP (Interior Gateway Protocol) ข้อกำหนดรูปแบบที่ถ่ายโอนข้อมูลเส้นทางกันภายในเครือข่ายเดียวกัน

2.3.2.5 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Services) ผู้ใช้สามารถใช้ข้อกำหนดรูปแบบได้โดยตรง ข้อกำหนดรูปแบบนี้ประกอบด้วย

1) BOOTP (BOOT Protocol) เมื่อผู้ใช้เปิดเครื่องคอมพิวเตอร์บนเครือข่ายให้เริ่มทำงานข้อกำหนดรูปแบบนี้จะอ่าน โปรแกรมควบคุมการทำงานจากคอมพิวเตอร์ให้บริการ (Server Computer) มาให้

2) FTP (File Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนไฟล์ข้อมูลระหว่างคอมพิวเตอร์ซึ่งอาจจะอยู่บนเครือข่ายเดียวกันหรือต่างเครือข่ายกันก็ได้

3) TELNET เป็นข้อกำหนดรูปแบบ ที่ให้บริการเกี่ยวกับการควบคุมการติดต่อระยะทางไกล (Remote Login)

กลุ่มข้อกำหนดรูปแบบอื่นที่นอกเหนือจากกลุ่มที่จัดไว้และบริการที่สำคัญๆจัดทำไว้ให้บนเครือข่ายที่สนใจมีดังนี้

1) NFS (Network File System) เป็นข้อกำหนดรูปแบบที่ทำให้ผู้ใช้คอมพิวเตอร์เครื่องหนึ่งสามารถเข้าไปดูไฟล์ข้อมูลและใช้งานไฟล์ข้อมูลซึ่งอยู่ในคอมพิวเตอร์เครื่องอื่นได้

2) NIS (Network Information Services) เป็นข้อกำหนดรูปแบบที่ให้บริการกับ User Accounts ข้ามเครือข่าย เช่น Logins และ Password

3) RPC (Remote Procedure Call) ข้อกำหนดรูปแบบที่อำนวยความสะดวกให้กับโปรแกรมประยุกต์ที่ใช้งานกับการควบคุมระยะทางไกล

SMTP (Simple Mail Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนจดหมายอิเล็กทรอนิกส์ (Electronic Mail) ระหว่างคอมพิวเตอร์

4) SNMP (Simple Network Management Protocol) ข้อกำหนดรูปแบบที่ให้บริการข่าวสารต่างๆที่แสดงสถานะของเครือข่ายและอุปกรณ์ที่ต่ออยู่บนเครือข่าย

2.3.3 โครงสร้างของชุดโปรโตคอล TCP/IP

ตาราง 2.2 โครงสร้างของชุดโปรโตคอล TCP/IP เปรียบเทียบกับ OSI Model

OSI Model	TCP/IP (Internet)
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Network Interface
Physical	Physical

1) Application Layer ในชั้นนี้ประกอบด้วยโปรแกรมประยุกต์ที่ใช้ในเครือข่าย เช่น โปรแกรมส่งถ่ายข้อมูล (File-Transfer Program) และอาจกล่าวได้ว่าโปรโตคอล TCP/IP ก็คือโปรโตคอลในชั้นแอปพลิเคชันรวมกับชั้นพีรีเซนต์ชันของ OSI โมเดลนั่นเอง

2) Transport Layer ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายเปรียบเทียบกับชั้นเซสชันร่วมกับ ทรานสปอร์ตเลเยอร์นั่นเอง โดยโปรโตคอล TCP/IP มีซ็อกเก็ต (Socket) เป็นจุดปลาย (End-Point) ในการสื่อสาร ซึ่งซ็อกเก็ตนี้ประกอบไปด้วยหมายเลขของคอมพิวเตอร์ และหมายเลขพอร์ต (Port) ของเครื่องที่ต้องการส่งข้อมูลไปถึง ในชั้นนี้มีการรับรองให้ถึงที่หมาย และลำดับข้อมูลที่ส่งโดยปราศจากข้อมูลซ้ำซ้อน โดยในชั้นนี้มีโปรโตคอลหลัก 2 ตัว คือ TCP และ UDP

3) Internet Layer ในชั้นนี้มีการกำหนดค่าแกรม (Datagram) และทำการหาเส้นทางการส่ง การทำงานในชั้นนี้จะเป็นแบบ Connectionless เนื่องจากไม่มีการเชื่อมต่อระหว่างต้นทางกับปลายทางก่อนค่าแกรมแต่ละตัวสามารถเลือกเส้นทางไปโดยอิสระ และไม่มีการรับประกันความถูกต้องข้อมูลหรือลำดับการส่ง

4) Network Interface Physical Layer ทำหน้าที่ควบคุมตัวกลางที่ใช้สื่อสารข้อมูลและรูปแบบการเชื่อมต่อในทางกายภาพ ชั้นนี้จะแบ่งข้อมูลออกเป็นส่วนๆ เรียกว่า เฟรม หรือ แพคเกจและส่งข้อมูลที่ไประบายไปยังปลายทางที่เชื่อมต่อกันอยู่บนเครือข่ายเดียวกัน

2.3.4 ข้อแตกต่างระหว่างชุดโพรโตคอล TCP/IP

2.3.4.1 ลำดับการติดต่อสื่อสารของชั้นเลเยอร์ ในรูปแบบ OSI นั้นจะกำหนดลำดับชั้นการสื่อสารที่เป็นลำดับขั้นตอนการติดต่อที่แน่นอน โดยเฉพาะการอินเตอร์เฟสระหว่างชั้นเลเยอร์ ซึ่งทำให้รูปแบบ OSI สามารถเป็นระบบเปิดสำหรับระบบเครือข่ายคอมพิวเตอร์ทั่วไป เพราะไม่ว่าจะมีการเปลี่ยนแปลงโพรโตคอลในเลเยอร์ชั้นใดก็ตามจะไม่มีกระทบต่อการสื่อสารเลเยอร์ชั้นถัดไป ในขณะที่ชุดโพรโตคอล TCP/IP จะไม่มีการกำหนดรูปแบบการติดต่อที่ตายตัว เพื่อให้ผู้ออกแบบเครือข่ายมีอิสระสามารถเปลี่ยนแปลงโครงสร้างของเครือข่ายได้ง่ายขึ้น

2.3.4.2 การติดต่อสื่อสารระหว่างเครือข่ายหรืออินเทอร์เน็ต คือการติดต่อสื่อสารข้อมูลระหว่างระบบคอมพิวเตอร์ 2 ระบบที่ไม่สามารถติดต่อสื่อสารกันได้ในผ่านทางเครือข่ายการสื่อสารข้อมูลเพียงเครือข่ายเดียวได้คืออาศัยเครือข่ายตั้งแต่ 2 เครือข่ายขึ้นไปในการติดต่อสื่อสารกัน และเครือข่ายเหล่านี้อาจจะมีลักษณะของเครือข่ายที่ต่างกันก็ได้ ความแตกต่างในเรื่องของอินเทอร์เน็ตระหว่างชุดโพรโตคอล TCP/IP กับรูปแบบ OSI ก็คือ ในชุดโพรโตคอล TCP/IP จะใช้โพรโตคอลสำหรับอินเทอร์เน็ตที่เรียกว่า โพรโตคอล IP (Internet Protocol) ซึ่งในรูปแบบ OSI จะเรียกว่า โพรโตคอลสำหรับการอินเทอร์เน็ตว่า โพรโตคอลเน็ตเวิร์ค

2.3.4.3 การบริการการเชื่อมต่อการสื่อสาร (Connection Service) ในชุดโพรโตคอล TCP/IP นั้นจะมีการบริการการเชื่อมต่อการสื่อสารระหว่างต้นทางและปลายทาง 2 แบบ คือการบริการแบบ Connectionless และ แบบ Connection-Oriented ส่วนในรูปแบบ OSI จะให้ความสำคัญเฉพาะกับการบริการแบบ Connection-Oriented เท่านั้น โพรโตคอลควบคุมการจัดการสื่อสารในชุดโพรโตคอล TCP/IP จะใช้โพรโตคอล TCP เป็นโพรโตคอลสำหรับควบคุมการสื่อสาร กำหนดตำแหน่งต้นทางและปลายทาง และอื่นๆกับข้อมูล ซึ่งในรูปแบบ OSI นั้นจะแบ่งแยกการควบคุมการสื่อสารออกจากกันโดยใช้โพรโตคอลเซสชันและโพรโตคอลทรานสปอร์ตตามลำดับ

2.3.5 ลักษณะของการติดต่อ แบ่งออกเป็น 2 ชนิด คือ

2.3.5.1 Connection-Oriented คือ การติดต่อที่ต้องมีการเชื่อมโปรเซส (Process) ที่จะมีการส่งหรือรับข้อมูลซึ่งใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่างโปรเซส ถึงแม้ว่าข้อมูลนี้อาจจะผ่าน Packet-Switching Network บริการชนิดนี้ส่วนมากจะใช้ในกรณีที่มีข่าวสารต้องการมากกว่าหนึ่งข่าวสาร ดังนั้นสามารถแบ่งขั้นตอนการทำงานออกเป็น

2.3.5.1.1 ขั้นการสร้างการติดต่อ (Connection Establishment)

2.3.5.1.2 ขั้นการส่งผ่านข้อมูล(Data Transfer)

2.3.5.1.3 ขั้นยกเลิกการติดต่อ (Connection Termination)

2.3.5.2 Connectionless หรือ **ดาต้าแกรม** คือ จะไม่มีขั้นการสร้างการติดต่อ และขั้นการยกเลิกการติดต่อ แต่จะมีขั้นการส่งผ่านข้อมูลเพียงอย่างเดียว โดยข้อมูลซึ่งเรียกว่าดาต้าแกรมจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นอยู่กับดาต้าแกรมอื่น

2.3.6 ความสัมพันธ์ของ TCP และ UDP กับอินเทอร์เน็ต

ตามสถาปัตยกรรมของชุดข้อกำหนดรูปแบบ TCP/IP ระดับชั้น ทรานสปอร์ต จะมีบริการการส่งข้อมูลอยู่ 2 ประเภท คือ TCP และ UDP

ความแตกต่างของ TCP กับ UDP อยู่ที่วิธีการสื่อสารระหว่างเครื่องคอมพิวเตอร์สองเครื่อง TCP ต้องสร้างการติดต่อให้ได้ก่อนแล้วจึงส่งข้อมูลและยกเลิกเส้นทางที่ต่อกันเมื่อจบการส่งข้อมูล ส่วน UDP จะไม่มีการสร้างการติดต่อ แต่จะใช้ไอพีแอดเดรสของคอมพิวเตอร์ปลายทางใส่รวมกับข้อมูลแล้วส่งออกไป ข้อมูลจะเดินตามไอพีแอดเดรสด้วยเส้นทางที่ไม่แน่นอน ขึ้นอยู่กับความเหมาะสมจนกว่าจะถึงคอมพิวเตอร์ปลายทาง

TCP เป็นวิธีการสื่อสารที่มีความเชื่อมั่นได้มากกว่า UDP ทั้งการส่งและการตอบรับข่าวสาร วิธีของ UDP จะไม่มีการรับประกันว่าข่าวสารจะไปถึงผู้รับ แต่ UDP มีวิธีการที่จะตรวจสอบโดยให้เครื่องคอมพิวเตอร์ปลายทางส่งข่าวสารการตอบรับกลับมาเมื่อได้รับข่าวสารที่ส่งไปให้ ทางด้านผู้ส่งจะตั้งเวลาวานานเท่าใดควรมีการตอบรับกลับมา ถ้าเกินเวลา (Time-Out) แล้วยังไม่มีการตอบรับกลับมาก็สันนิษฐานได้ว่าข่าวสารสูญหาย UDP ก็จะทำการส่งข่าวสารเดิมให้ใหม่ เครือข่ายใหญ่เช่นอินเทอร์เน็ตซึ่งมีเครื่องคอมพิวเตอร์บนเครือข่ายเป็นจำนวนมาก การใช้ TCP ที่ต้องสร้างการติดต่อและยกเลิกอยู่บ่อยๆจึงเป็นไปได้ในทางปฏิบัติ ดังนั้นอินเทอร์เน็ตจึงใช้การส่งข่าวสารด้วย UDP แทน

การบริการที่ดำเนินการด้วยชุดข้อกำหนดรูปแบบ TCP/IP ต้องออกแบบการให้บริการเป็น TCP หรือ UDP อย่างใดอย่างหนึ่งเท่านั้น การบริการอย่างเดียวกันจะให้บริการทั้งสองประเภทไม่ได้ เช่น การ Loginsจากระยะไกล (Telnet) และการโอนย้ายไฟล์ข้อมูล FTP ใช้การให้บริการแบบ TCP ส่วน TFTP (Trivial File Transfer Protocol) ใช้การให้บริการแบบ UDP

2.3.7 พอร์ตและซ็อกเก็ต (TCP Port and Socket)

การใช้บริการของ TCP ทำได้โดยผ่านทางพอร์ต โดยมีการกำหนดหมายเลขให้กับบริการต่าง ๆ การบริการที่ต่างกันจะมีการใช้หมายเลขพอร์ตไม่เหมือนกัน

พอร์ตถูกกำหนดตามชนิดของการบริการ เช่นต้องการสื่อสารกันด้วย Telnet เครื่องคอมพิวเตอร์ต้องการจะส่งข้อความต้องการออกทางพอร์ตหมายเลข 23 ที่จัดไว้สำหรับ Telnet ไปยังเครื่องคอมพิวเตอร์ที่ต้องการติดต่อด้วยถ้าเครื่องคอมพิวเตอร์ปลายทางนั้นต้องการยอมรับการติดต่อ การสื่อสาร Telnet ก็จะเริ่มขึ้น ในเวลาถ้ามีความต้องการสื่อสารด้วย Telnet จากเครื่องคอมพิวเตอร์อื่นเข้ามา จะไม่สามารถทำการติดต่อได้เนื่องจากพอร์ตหมายเลข 23 สำหรับบริการ Telnet ถูกใช้งานอยู่ TCP พอร์ตเป็นพอร์ตทาง logical ไม่ใช่ พอร์ตที่อยู่ทางด้านหลังของเครื่องคอมพิวเตอร์ ใช้คอนเน็คเตอร์ (Connector) บน backbone เพื่อให้สามารถติดต่อสื่อสารบริการหลายประเภทได้พร้อม ๆ กัน

ซ็อกเก็ตเป็นหมายเลขกำหนดวงจรซึ่งทางเข้า-ออกของระดับชั้น TCP ของเครื่องคอมพิวเตอร์บนเครือข่าย หมายเลขซ็อกเก็ตประกอบด้วยหมายเลข IP Address กับหมายเลขพอร์ตรวมกัน หมายเลขซ็อกเก็ตใช้สำหรับการสื่อสารคอมพิวเตอร์ที่อยู่ต่างเครือข่ายกัน เนื่องจาก IP Address ของเครือข่ายจะไม่ซ้ำกันถึงแม้ว่าหมายเลขพอร์ตจะซ้ำกันได้

เครื่องคอมพิวเตอร์บนเครือข่ายจะสร้างตารางพอร์ตขึ้นมาเก็บรายการของพอร์ตที่มีอยู่เพื่อใช้ในการตรวจสอบว่าพอร์ตที่ต้องการนั้นสามารถใช้บริการได้หรือไม่ และพอร์ตมีสถานะเป็นอย่างไรการที่พอร์ตอยู่ในรายการตารางพอร์ตของแต่ละเครื่อง ร้องขอการสื่อสารซึ่งกันและกันเรียกว่า 'Binding' ของพอร์ตและวิธีที่ทำให้พอร์ตที่ถูกใช้อยู่สามารถใช้สื่อสารได้อีกจากการขอใช้บริการเดียวกันเรียกว่า 'Multiplexing'

ตาราง 2.3 หมายเลขพอร์ตที่ให้บริการต่าง ๆ

TCP Port	บริการ
20	FTP Data
21	FTP Control
23	Telnet
25	Simple Mail Transfer Protocol
53	Domain Name Server
69	Trivial File Transfer Protocol
79	Finger

2.3.8 ข้อกำหนดรูปแบบ IP (Internet Protocol)

IP เป็นข้อกำหนดรูปแบบหนึ่งในชุดข้อกำหนดรูปแบบ TCP/IP ลักษณะการให้บริการเป็นแบบ Connectionless เช่นเดียวกับ UDP มีหน้าที่หลักเกี่ยวกับการจัดการ คาด้าแกรม ระหว่างเครื่องคอมพิวเตอร์เช่น ที่อยู่ปลายทาง วิธีการหาเส้นทางที่ดีที่สุดที่จะไปถึงปลายทาง และมีการจัดเตรียมวิธีการแก้ปัญหาที่อาจจะเกิดขึ้นในกรณีต่าง ๆ เป็นต้นนอกจากนี้ยังมีหน้าที่แบ่งย่อยคาด้าแกรมที่มี

ขนาดใหญ่เกินกว่าที่ IP Message กำหนดไว้ (ประมาณ 64 kilobyte) และรวบรวม ค้าแกรมย่อย ๆ กลับมาเป็น Message ให้เหมือนเดิมที่เครื่องปลายทาง การแบ่งย่อยละรวบรวมค้าแกรมมีอยู่ 4 วิธีคือ

- 1) Segmentation วิธีแบ่งค้าแกรมขนาดใหญ่เป็นค้าแกรมย่อย
- 2) Reassemble วิธีรวบรวมค้าแกรมย่อยให้กลับมาอยู่ในสภาพเดิม
- 3) Concatenation วิธีรวมค้าแกรมให้เป็น block
- 4) Separation วิธีแบ่ง block ให้เป็นค้าแกรม

2.3.9 ข้อกำหนดรูปแบบ ICMP (Internet Control Message Protocol)

ปัญหาหนึ่งที่เกิดขึ้นได้ง่ายกับค้าแกรม ก็คือ ค้าแกรมเดินทางผิดเส้นทาง ทำให้สูญหายหรือบางส่วนข่าวสารเสียหาย เครื่องคอมพิวเตอร์ด้านส่งต้องรู้เงื่อนไขที่จะทำให้เกิดความผิดพลาดต่าง ๆ บนเครือข่ายเพื่อให้ ค้าแกรมถูกส่งออกไปอย่างถูกต้อง

ICMP จะรายงานความผิดพลาดในระบบให้กับไอพีพิเศษ ICMP มี header เหมือนกับข่าวสารไอพีอื่น ๆ โดยถูกส่งรวมไปกับข่าวสารที่อยู่เครื่องปลายทาง ซึ่งเครื่องปลายทางจะส่งรายงานความผิดพลาดกลับมาให้เครื่องส่วนที่อยู่ทางด้านส่งจะรู้ว่าเกิดอะไรขึ้นบ้างในการเดินทางซึ่งข้อมูลในรายงานจะถูกนำมาใช้แก้ปัญหาคือต่อไป

2.4 ไอพีแอดเดรส (IP Address)

เครื่องคอมพิวเตอร์ที่ต่อเชื่อมกับอินเทอร์เน็ตหรือเครือข่ายที่ใช้ TCP/IP มีอยู่เป็นจำนวนมาก เครื่องคอมพิวเตอร์แต่ละเครื่องต้องสามารถระบุ หรืออ้างอิงได้โดยไม่เกิดความซ้ำซ้อนกับเครื่องคอมพิวเตอร์อื่น มิฉะนั้นแล้วข่าวสารที่เครือข่ายรับมาไม่สามารถส่งไปให้กับเครื่องคอมพิวเตอร์ที่ต้องการข่าวสารนั้นได้จึงต้องมีการจัดระบบที่ดี เครือข่ายอินเทอร์เน็ตหรือเครือข่ายที่ใช้ TCP/IP ได้ออกแบบจัดการระบบตรงส่วนนี้ไว้ใช้แล้ว

2.4.1 รหัสเลขประจำเครื่อง IP

เครือข่ายอินเทอร์เน็ตใช้รหัสหมายเลขมากำหนดให้แต่ละเครือข่าย และเครื่องคอมพิวเตอร์ภายในเครือข่ายที่เชื่อมโยงเรียกรหัสหมายเลขนี้ว่า อินเทอร์เน็ตแอดเดรส (Internet Address)

ไอพีแอดเดรส ประกอบด้วยเลขฐานจำนวน 32 บิต แบ่งออกเป็น 4 ส่วนแต่ละส่วนมี 8 บิต เมื่อดูเฉพาะแต่ละส่วนเป็นเลขฐานสิบจะได้เลขจำนวน 256 ค่า ไม่ซ้ำกัน (0-255) ไอพีแอดเดรสจะนำเอาหมายเลขทั้ง 4 ส่วนมารวมกันโดยแยกส่วนด้วยจุด ดังนั้นหมายเลขทั้งหมดที่เป็นไปได้ โดยที่ค่าไม่ซ้ำกัน คือ 256^4 หรือ 4,294,967,296 จำนวน มีค่าหมายเลขจาก 000.000.000.000 จนถึง 255.255.255.255 หมายเลขเหล่านี้เองที่อินเทอร์เน็ตใช้กำหนดให้กับเครือข่ายและเครื่องคอมพิวเตอร์เพื่อใช้อ้างอิง

ไอพีแอดเดรสบางหมายเลขสงวนไว้ใช้ ด้วยจุดหมายกรณีพิเศษทำให้ไอพีแอดเดรสที่ใช้ งานทั่วไปลดลงจากจำนวนที่เป็นไปได้ ความหมาย ไอพีแอดเดรสจะแบ่งได้เป็น 2 กลุ่มดังนี้

1) กลุ่มที่ใช้เป็นรหัสประจำบนเครือข่าย

2) กลุ่มที่ใช้เป็นรหัสประจำเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่าย (Host Computer) ไอพีแอดเดรสในกลุ่มรหัสประจำเครื่องคอมพิวเตอร์สามารถซ้ำกันได้ แต่กลุ่มรหัสประจำเครือข่ายจะซ้ำกันไม่ได้ ดังนั้นรหัสเครื่องที่ซ้ำกันจึงไม่มีผลต่อการอ้างอิงถึง

นอกจากนี้เพื่อความเหมาะสมในการกำหนด ไอพีแอดเดรสให้กับผู้ขอทางผู้บริหารอินเทอร์เน็ตบ่งคลาสนของผู้ขอ ๆ ไอพีแอดเดรสตามขนาดของเครือข่าย เพื่อให้ทรัพยากรส่วนนี้ถูกใช้อย่างคุ้มค่าที่สุด องค์กรขนาดใหญ่ก็จะจัดให้อยู่ในคลาสที่สามารถกำหนด ไอพีแอดเดรสให้กับเครื่องคอมพิวเตอร์ในเครือข่ายได้มาก การแบ่งคลาสจะแบ่งได้

รูป 2.2 แสดงคลาสของไอพีแอดเดรส

Class A	0	Network ID (7)			Host ID (24)	
Class B	1	0	Network ID (14)		Host ID (16)	
Class C	1	1	0	Network ID (21)	Host ID (8)	
Class D	1	1	1	0	Multicast Address	
Class E	1	1	1	1	0	Reserved

การกำหนดหมายเลขของเครือข่ายและเครื่องคอมพิวเตอร์นอกจากจะแบ่งไอพีแอดเดรสเป็นคลาสทั้ง 5 ประเภทแล้วยังมีข้อกำหนดปลีกย่อยอีกหลายประการที่ผู้วางระบบต้องรู้ไว้ เพื่อที่จะกำหนดแอดเดรสใช้งานได้อย่างถูกต้องไม่ผิดพลาดได้แก่ ไอพีแอดเดรสที่เป็น '0' ทุกบิต จะไม่ใช้งานทั่วไป แต่จะนำไปใช้กับอุปกรณ์หาเส้นทาง (Router) เพื่อกำหนด 'Default Route'

ไอพีแอดเดรสในแต่ละส่วน คือ ส่วนหมายเลขเครือข่ายและส่วนหมายเลขประจำเครื่องคอมพิวเตอร์จะเป็น '0' หรือ '1' ทุกบิตไม่ได้

- ถ้าส่วนของหมายเลขประจำเครื่องคอมพิวเตอร์เป็น '0' ทุกบิตหมายถึง หมายเลขเครือข่ายนั้นใช้งานร่วมกับอุปกรณ์หาเส้นทางเพื่อบอกให้เครื่องคอมพิวเตอร์รู้ว่าตัวเองอยู่ในเครือข่ายใด

- ส่วนหมายเลขประจำเครื่องคอมพิวเตอร์เป็น '1' ทุกบิตนั้น หมายถึง หมายเลขเครือข่ายนั้นใช้สำหรับการกระจายข่าวภายในเครือข่าย (Broadcast Address)

- ถ้าไอพีแอดเดรสของทั้งสองส่วนนั้นเป็น '1' ทุกบิต หมายถึง แอดเดรสที่ใช้การจ่าย
ข่าวหรืออีกนัยหนึ่งเครื่องคอมพิวเตอร์ที่ส่งแอดเดรสนี้เข้าสู่ระบบไม่ทราบว่าตนเองอยู่เครือข่ายใด

3) ไอพีแอดเดรสของคลาส A หมายเลข 127.0.0.0 จะสงวนไว้ใช้สงวนเฉพาะอย่าง เช่น
IPC (Internet – Process Communication)

จากรูป 2.2 จะแสดงประเภทของ ไอพีแอดเดรสและข้อกำหนดปลีกย่อยต่าง ๆ เราสามารถ
ทราบได้ว่าเครือข่ายขององค์กรถูกจัดอยู่ใน Class ใด โดยดูจากหมายเลขของ 8-บิตแรกซ้ายมือสุด
ดังนี้

Class-A: ไอพีแอดเดรสอยู่ในช่วง 1 ถึง 126

Class-B: ไอพีแอดเดรสอยู่ในช่วง 127 ถึง 1191

Class-C: ไอพีแอดเดรสอยู่ในช่วง 192 ถึง 233

2.5 ซ็อกเก็ต (Socket)

2.5.1 เอพีไอ (API) ข้อกำหนดในการติดต่อสื่อสาร

TCP/IP เป็น Software ในการเชื่อมโยงเครือข่ายชนิดหนึ่งที่สนับสนุน โปรแกรมประยุกต์
ในการอินเทอร์เน็ตสำหรับการติดต่อสื่อสารบนอินเทอร์เน็ต ซึ่งจะอธิบายชนิดของ API ในที่นี้เพียง
2 ชนิด

2.5.1.1 Socket System Calls

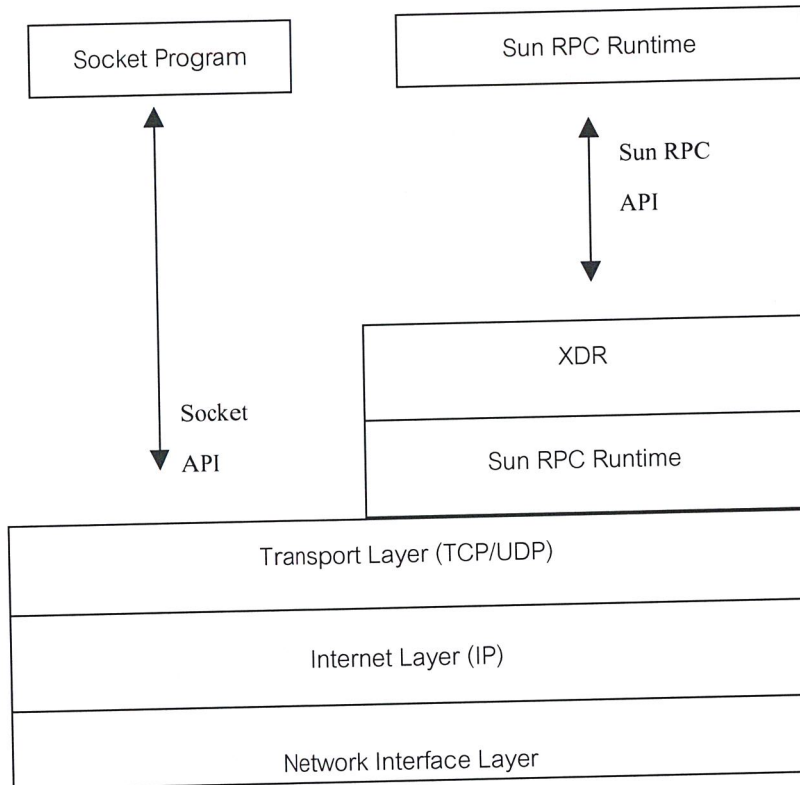
ซ็อกเก็ต เข้าถึงข้อกำหนดในการติดต่อสื่อสารของโครงข่าย ซึ่งโดยทั่วไปจะเกี่ยวข้อง
กับการเปลี่ยนแปลง BSD ของ UNIX ซึ่งเป็นอุปกรณ์อันดับแรกของ ซ็อกเก็ต API การอินเทอร์เน็ต
ของซ็อกเก็ตได้จัด API สำหรับการติดต่อสื่อสารของโครงข่ายซึ่งเป็น API ที่ใกล้เคียงกับการทำงาน
ของ I/O มาก ๆ

การอินเทอร์เน็ตของซ็อกเก็ตเป็นเครื่องมือที่ใช้งานอย่างกว้างขวางของระบบปฏิบัติการ
UNIX และระบบปฏิบัติการอื่น ๆ โปรแกรมที่เข้าถึงซ็อกเก็ตสามารถที่จะใช้งานเป็นเครื่องมือ
สำหรับการติดต่อสื่อสารบนโครงข่ายอินเทอร์เน็ต โดยอยู่ในสภาพแวดล้อมที่แตกต่างกันได้ติดต่อ
สื่อสารกันได้ดีกับ Host ทุกชนิดซ็อกเก็ต API โดยทั่วไปมักใช้สำหรับโครงข่ายที่ใช้งานบน
TCP/IP และก็สามารถใช้งานได้ดีกับโครงข่ายแบบอื่น ๆ

2.5.1.2 Remote Procedure Call Facilities

Remote Procedure Call (RPC) เป็นชนิดหนึ่งของ API ซึ่งยอมให้ผู้เขียน โปรแกรมกับ
โครงข่ายใช้ Procedure Call ที่มีโครงสร้างอย่างง่าย ๆ เพื่อที่จะควบคุมกันระหว่าง Procedure ของ
Host ที่มีการทำงานแตกต่างกันบนอินเทอร์เน็ต สิ่งอำนวยความสะดวกของ RPC ที่ใช้กันอย่างกว้าง
ขวางบนสถานะ TCP/IP ได้แก่

- SUN RPC
- Hewlett-Packard RPC



รูปที่ 2.3 แสดงการเปรียบเทียบระหว่าง Socket และ SUN RPC application

programming

Socket Interface เป็นการเริ่มพัฒนาของหลาย ๆ BSD ของระบบปฏิบัติการ UNIX โดยการใช้Socket Interface จะมีอยู่ 2-โปรแกรมประยุกต์ คือ

- 1) การทำงานที่ Logical System
- 2) การทำงานที่ Remote System

ซึ่งสามารถติดต่อกับ Host อื่น ๆ ได้ด้วยวิธีมาตรฐาน Socket Interface ที่ใช้กันอย่างกว้างขวางคือ API ที่ใช้งานบน TCP/IP

Socket API เป็นอุปกรณ์ที่ใช้สำหรับการเชื่อมต่อระหว่างผู้ขอใช้บริการและผู้ให้บริการระหว่าง 2 ส่วนประกอบ ประยุกต์ที่ทำงานบนระบบคอมพิวเตอร์ที่ต่างกัน

การติดต่อระหว่างผู้ขอใช้บริการและผู้ให้บริการแต่ละครั้ง จะเรียก Function ขึ้นมาจัดการเชื่อมต่อระหว่างผู้ขอใช้บริการและผู้ให้บริการนั้น ๆ การเชื่อมต่อระหว่าง 2 ส่วนประกอบ (component) ประยุกต์เป็นพื้นฐานบนการใช้โครงสร้างข้อมูลที่เรียกว่า Socket เพื่อเป็นการบริการการติดต่อสื่อสารโดยใช้ TCP/IP

ส่วนประกอบของผู้ใช้บริการและผู้ให้บริการ (Client-Server Component) จะเรียก ฟังก์ชันที่ใช้ส่งและรับข้อมูลบนอินเทอร์เน็ต ซึ่งคล้ายกับวิธีการเรียก ฟังก์ชันที่ใช้กับไฟล์ทั่ว ๆ ไป

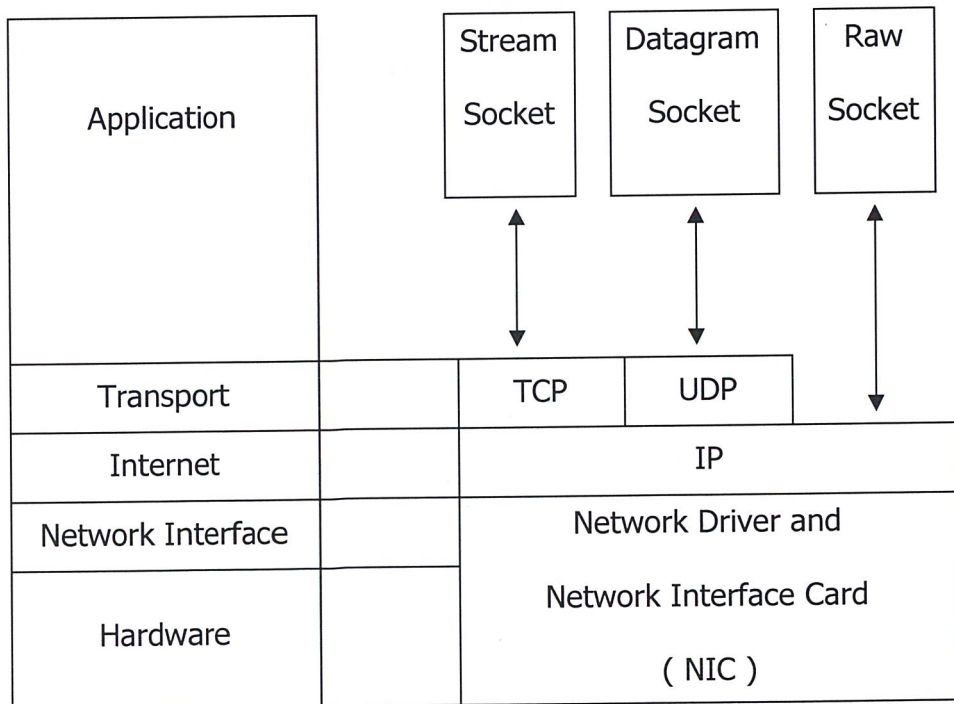
Socket API สามารถจะใช้งานได้ ใน Protocol อื่น ๆ ที่นอกเหนือจาก TCP/IP ได้ แต่ ในที่นี่จะอธิบายเฉพาะ Protocol TCP/IP เท่านั้น

2.5.2 ชนิดของ Socket Socket ที่ใช้กันอยู่ในการติดต่อสื่อสารบน TCP/IP จะมีอยู่สามแบบด้วยกันคือ

2.5.2.1 Stream Socket คือ Socket ที่ใช้โปรโตคอล TCP ชั้น Transport Layer ในการติดต่อ Socket ชนิดนี้จะใช้การติดต่อแบบ Connection Oriented ซึ่งการส่งรับข้อมูลแบบนี้สามารถเชื่อถือได้

2.5.2.2 Datagram Socket คือ Socket ที่ใช้โปรโตคอล UDP ใน Transport Layer ในการติดต่อ Datagram Socket จะเชื่อถือไม่ค่อยได้ รูปแบบของข้อมูลที่ส่งจะถูกแบ่งออกเป็น ส่วน ๆ โดยแต่ละอันจะถูกส่งไปยังปลายทางโดยอิสระจากกัน

2.5.2.3 Raw Socket คือ Socket ที่อยู่ภายใต้ชั้น IP และ ICMP ซึ่งก็เกิดชนิดนี้จะใช้เฉพาะ การส่งที่มีจุดมุ่งหมายพิเศษ



TCP/IP Architectural Layers

TCP/IP Components and Protocols

รูป 2.4 แสดงลักษณะทั้งสามแบบของ Socket

2.5.3 Socket Address

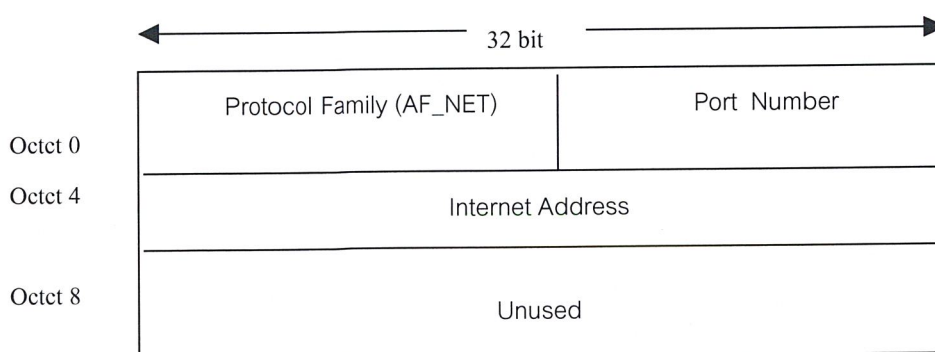
ระบบของซ็อกเก็ตโดยทั่ว ๆ ไปจะอ้างอิงโครงสร้างข้อมูลแบบพอยน์เตอร์ (Pointer) ที่ประกอบไปด้วยข้อมูลต่าง ๆ ซึ่งก็คือ Socket Address นั่นเอง

สำหรับโปรโตคอล TCP/IP จะมีโครงสร้างของ Socket Address ดังนี้

1) Family เป็นตัวเลขจำนวนเต็ม 16 บิต ที่ชี้เฉพาะโปรโตคอลเช่น TCP/IP คือ AF_INET, UNIX คือ AF_UNIX

2) Port เป็นเลขจำนวนเต็ม 16 บิต ที่ระบุหมายเลขพอร์ต สำหรับใช้ในกระบวนการต่าง ๆ สำหรับผู้ให้บริการ หมายเลขพอร์ตที่ใช้จะถูกกำหนดโดยตัวผู้ให้บริการเองซึ่งผู้ขอใช้บริการที่ต้องการติดต่อกับผู้ให้บริการ โดยใช้โปรแกรมนั้นต้องรู้หมายเลขพอร์ตที่ใช้ติดต่อจึงสามารถติดต่อกันได้

3) Address เป็นเลขจำนวนเต็ม 32 บิต ซึ่งภายในคือ ไอพีแอดเดรสที่อยู่ในรูปแบบของเลขฐานสองในการทำงานเพื่อติดต่อกันผู้ขอใช้บริการต้องรู้ไอพีแอดเดรสของผู้ให้บริการจึงสามารถทำการติดต่อกันได้และแต่ละครั้งของการติดต่อของผู้ขอใช้บริการ ผู้ให้บริการต้องมีการกำหนดซ็อกเก็ตแอดเดรส ซึ่งซ็อกเก็ตแอดเดรสก็หมายรวมถึง ซ็อกเก็ตแอดเดรสของผู้ให้บริการ และซ็อกเก็ตของผู้ขอใช้บริการ กระบวนการติดต่อของผู้ใช้บริการและผู้ให้บริการต้องมีการชี้เฉพาะไปที่โปรโตคอลใน Transport Layer เพื่อใช้ในการติดต่อสื่อสาร รูป 2.5 จะแสดงความหมายของชนิดของข้อมูลและ โครงสร้างของข้อมูลที่จะใช้ในการกำหนดซ็อกเก็ตแอดเดรส



รูป 2.5 แสดงโครงสร้างของข้อมูลที่ใช้ในการกำหนดซ็อกเก็ตแอดเดรส

2.5.4 Socket System Call

ซ็อกเก็ต API ให้ Socket System Call เป็นโปรแกรมประยุกต์ที่ทำหน้าที่เรียกบริการการร้องขอที่จะทำการติดต่อสื่อสาร Socket System Call พื้นฐานที่ใช้กันทั่วไปอธิบายได้ดังนี้

2.5.4.1 accept. ถูกใช้ใน connection-oriented application protocol ที่ใช้ TCP โดยใช้ใน ขบวนการผู้ให้บริการซึ่งผู้ให้บริการจะรอรับการร้องขอเพื่อติดต่อจากผู้ใช้บริการ

2.5.4.2 bind. ถูกใช้ร่วมกับ Endpoint identifier (internet address และหมายเลข port) ซึ่ง เลขจำนวนเต็มเป็นตัวอธิบายในการเปิดซ็อกเก็ต

2.5.4.3 close. ใช้ในการปิดซ็อกเก็ต โดยในรายละเอียดจะคล้ายกับการใช้วิธีปิดไฟล์

2.5.4.4 connect. ถูกใช้ใน connection-oriented application protocol ที่ใช้ TCP โดยขบวนการผู้ใช้บริการ จะต้องสร้างการติดต่อแบบ TCP กับผู้ให้บริการ

2.5.4.5 gethostbyaddr. จะใช้เมื่อต้องการข้อมูลเกี่ยวกับ โฮสต์ซึ่งรู้อินเทอร์เน็ตแอดเดรส แล้ว

2.5.4.6 gethostid. ถูกใช้เพื่อให้ได้อินเทอร์เน็ตแอดเดรสของ Local Host

2.5.4.7 gethostbyname. ใช้เพื่อให้ได้อินเทอร์เน็ตแอดเดรสของ remote host ซึ่งเรารู้ host name

2.5.4.8 gethostname. ถูกใช้เพื่อให้ได้ชื่อของ Local Host

2.5.4.9 getpeername. ถูกใช้เพื่อให้ได้ Endpoint address ซึ่งเกี่ยวข้องกับขบวนการด้านรีโมท

2.5.4.10 getsockname. ถูกใช้เพื่อให้ได้ชื่อที่ซ็อกเก็ตให้มา

2.5.4.11 getprotobyname. ใช้เพื่อให้ได้ค่าเลขจำนวนเต็มซึ่งเป็นค่าของ โปรโตคอลที่จะบอกชื่อของ โปรโตคอล

2.5.4.12 gerservbyname. ถูกใช้เพื่อให้ได้ค่าเลขจำนวนเต็มซึ่งเป็นค่าที่ให้ชื่อของการบริการของโปรแกรม ประยุกต์บน Application Layer ของโมเดล TCP/IP ที่รู้จักกันดี

2.5.4.13 getsockopt. ถูกใช้เพื่อให้ได้พารามิเตอร์ของซ็อกเก็ต

2.5.4.14 listen. ใช้ในการติดต่อแบบ connection-oriented application protocol ที่ใช้ TCP โดยเป็นขบวนการของผู้ให้บริการที่แสดงให้เห็นว่าผู้ให้บริการจะยอมรับติดต่อจากผู้ใช้บริการ เมื่อมีการร้องขอเพื่อที่จะสร้างการติดต่อแบบ TCP

2.5.4.15 read. ถูกใช้เพื่อที่จะรับข้อมูลจากซ็อกเก็ต

2.5.4.16 readv. เป็นอีกตัวหนึ่งของ read เพื่อที่จะรับข้อมูลใส่ในบัฟเฟอร์ที่ไม่ติดกัน

2.5.4.17 recv. เป็นอีกตัวเลือกหนึ่งของ read เพื่อที่จะใช้รับข้อมูลที่กำลังเข้ามาจากซ็อกเก็ตและใช้แฟลทมาเป็นข้อกำหนดเพิ่มเข้าไปด้วย

2.5.4.18 recvfrom. ใช้กับ connectionless application protocol ที่ใช้แบบ UDP เพื่อที่จะรับ user datagram จาก datagram socket เพื่อที่จะให้ได้ที่อยู่ซ็อกเก็ตของผู้ส่ง

2.5.4.19 recvmsg. ใช้กับ connectionless application protocol ที่ใช้แบบ UDP เพื่อที่จะรับ user datagram จาก **datagram socket** โดยในบัฟเฟอร์จะประกอบด้วยส่วนหัวของ user datagram ซึ่งก็คือส่วนหนึ่งของข้อมูลของ user datagram

2.5.4.20 select. ถูกใช้เพื่อให้ขบวนการรอคอย(wait) สมบูรณ์

2.5.4.21 send. ใช้ในอีกทางเลือกหนึ่งของ write เพื่อที่จะส่งข้อมูลบน socket ใช้เงื่อนไขของแฟลทเพิ่มมาด้วย

2.5.4.22 sendmsg. ใช้กับ connectionless application protocol ที่ใช้แบบ UDP เพื่อที่จะส่ง user datagram จาก **datagram socket** โดยในบัฟเฟอร์จะประกอบด้วยส่วนหัวของ user datagram ซึ่งก็คือส่วนหนึ่งของข้อมูลของ user datagram

2.5.4.23 sendto. ใช้กับ connectionless application protocol ที่ใช้แบบ UDP เพื่อที่จะส่งข้อมูลบน datagram socket และระบุ socket address ของ remote process, sendto จะถูกเรียกมาเพื่อให้สัมพันธ์กับการสร้างการติดต่อในเวลาที่จะส่งข้อมูล

2.5.4.24 setsockopt. ใช้กำหนดค่าสำหรับพารามิเตอร์ต่าง ๆ ของซ็อกเก็ต

2.5.4.25 shutdown. ใช้กับ connection-oriented application protocol ที่ใช้แบบ TCP โดย shutdown จะทำการยกเลิกการติดต่อที่ TCP สร้างขึ้น

2.5.4.26 socket. ใช้เพื่อเริ่มต้นโครงสร้างทางข้อมูลของทางซ็อกเก็ต ทำการแยกแยะโปรโตคอลที่ใช้งานสำหรับการติดต่อสื่อสารในชั้น transport layer และได้ค่าจำนวนเต็มที่สามารถบอกรายละเอียดได้โดยหลังจากที่ระบบการเรียกของซ็อกเก็ตสามารถใช้ตัวเลขนั้นสามารถใช้อ้างอิงถึงซ็อกเก็ตได้

2.5.4.27 write. ใช้เพื่อส่งข้อมูลบนซ็อกเก็ต

2.5.4.28 writev. เป็นอีกทางเลือกหนึ่งของ write เพื่อที่จะส่งข้อมูลจากบัฟเฟอร์ที่ไม่ติดกัน

2.5.5 Application Protocol

สามารถมีซ็อกเก็ต API ได้ทั้งสองแบบ คือทั้ง Datagram Socket และ Stream Socket ที่จะใช้บอกในการออกแบบโปรแกรม โดยมีโปรโตคอลติดต่อแบ่งเป็น 2 ประเภทดังนี้

2.5.5.1 Connectionless Application Protocol

เป็นโปรแกรมประยุกต์ของผู้ใช้บริการและผู้ให้บริการที่ใช้ Connectionless Application Protocol จะใช้กับ Datagram Socket เพื่อที่จะติดต่อสื่อสารบนการติดต่อ UDP ดังแสดงในรูป 2.6 แสดงให้เห็นถึงลำดับของ Socket System Call ที่ใช้แบบ Connectionless Application Protocol การทำแบบนี้จะไม่มี การตรวจสอบความผิดพลาดและการส่งซ้ำ (Retransmission) เนื่องจากเป็นการบริการแบบ UDP

พิจารณากระบวนการของผู้ให้บริการจะใช้ Socket, Bind System Call และรับข้อมูลโดยใช้ Recvfrom ซึ่งเป็น System Call ทั้ง 5 กระบวนการนี้ต้องการการเชื่อมต่อที่สมบูรณ์ระหว่างกระบวนการผู้ให้บริการ และกระบวนการผู้ให้บริการสำหรับ Connectionless Application Protocol นี้การเชื่อมต่อจะไม่สมบูรณ์จนกระทั่งถึงเวลาที่ กระบวนการผู้ให้บริการได้ส่งข้อมูลไปยังกระบวนการผู้ให้บริการแล้วจริงๆ

พิจารณากระบวนการของผู้ให้บริการจะใช้ Socket, Bind, Sendto System Call เพื่อให้การเชื่อมต่อระหว่างกระบวนการผู้ให้บริการและกระบวนการผู้ให้บริการนั้นสมบูรณ์ ซึ่งการเชื่อมต่อจะสมบูรณ์เมื่อเวลาที่ผู้ให้บริการ ส่งค่าตัวแปรอันดับแรกถึงผู้ให้บริการ

2.5.5.1.1 Server System Call

1) ผู้ให้บริการจะสั่งให้ socket system call เข้าถึงข้อมูลของโปรโตคอลใน 5 tuple ที่สัมพันธ์กัน ซึ่งในที่สุดแล้วก็จะถูกสร้างขึ้นเพื่อสนับสนุนการติดต่อสื่อสารระหว่างกระบวนการผู้ให้บริการผู้ให้บริการใน Connectionless Application จะเห็นว่า Socket System Call เาจะจงที่จะใช้ UDP

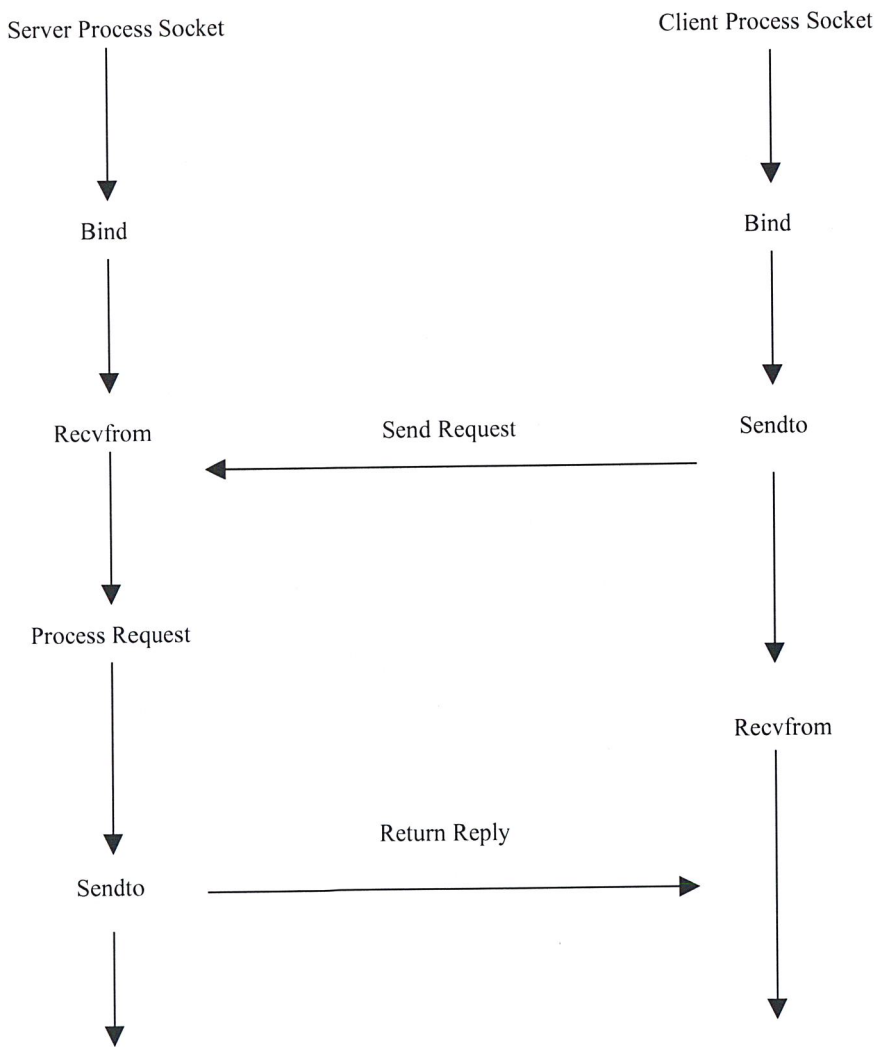
2) ผู้ให้บริการจะสั่งให้ bind system call (ระบุที่อยู่ของซ็อกเก็ตของผู้ให้บริการ) ทำแทนในส่วนที่เหลือของข้อมูลข่าวสารที่กำลังซ่อมแซม server's half-association, โดย half-association ของผู้ให้บริการบรรจุไปด้วยโปรโตคอลที่มีลักษณะเฉพาะตัวในตอนนี้อินเทอร์เน็ตแอดเดรส และหมายเลขพอร์ตของผู้ให้บริการกำลังทำงานอยู่

3) ผู้ให้บริการจะสั่งให้ทำ recvfrom system call เพื่อให้ผู้ให้บริการรับข้อมูลจากผู้ให้บริการที่กำลังเข้ามา, recvfrom system call เป็นสาเหตุให้ผู้ให้บริการ ขบวนการรอคอย จนกระทั่งข้อมูลจากผู้ให้บริการ มาถึงซึ่งมันก็คือความต้องการที่จะขอใช้บริการจากผู้ให้บริการ

4) เมื่อ user datagram จากผู้ให้บริการมาถึง, ผู้ให้บริการจะทำขบวนการ request process ซึ่งจะเห็นว่าหัวข้อมูลของ user datagram ที่ส่งมาจะประกอบไปด้วยอินเทอร์เน็ต

แอดเดรส และ หมายเลขพอร์ตของผู้ให้บริการ ข่าวสารเหล่านี้จะจัดหาหนทางไปให้ผู้ให้บริการจนกระทั่ง 2 ชั้นสุดท้ายของข้อมูลถูกร่องขอเพื่อให้ 5 tuple สมบูรณ์ ซึ่งจะสัมพันธ์ระหว่างด้าน ผู้ขอใช้บริการและผู้ให้บริการ

5) ผู้ให้บริการจะตอบกลับไปยังผู้ขอใช้บริการ 1 ครั้งหรือมากกว่า โดยใช้คำสั่ง sendto system call ซึ่งจะทำให้การส่ง user datagram กลับไปหาผู้ขอใช้บริการ, sendto system call จะใช้ socket address ของผู้ขอใช้บริการเป็นตัวอ้างอิงเพื่อที่จะตกลงตอบรับการติดต่อ



รูปที่ 2.6 วิธีการแบบ connectionless application protocol โดยใช้ UDP

2.5.5.1.2 Client System Call

- 1) ผู้ขอใช้บริการจะสั่งให้ socket system call เข้าถึงข้อมูลโปรโตคอลสำหรับ half-association หนึ่งใน connectionless application protocol จะเห็นว่า socket system call จะใช้ UDP
- 2) ผู้ขอใช้บริการจะสั่งให้ bind system call ระบุที่อยู่ของซ็อกเก็ตของผู้ขอใช้บริการแล้ว ซึ่งเต็มไปด้วยส่วนที่เหลือของข้อมูลข่าวสารที่มีลักษณะ half-association ของผู้ขอใช้บริการ, ขบวนการที่ผู้ขอใช้บริการเป็นชนิดที่ร้องขอ networking software ที่จะเป็นแหล่งของอินเทอร์เน็ตแอดเดรสของผู้ขอใช้บริการและจัดหาหมายเลขพอร์ตที่ไม่ได้ใช้เอามาใช้ชั่วคราว
- 3) ผู้ขอใช้บริการจะสั่งให้ sendto system call ทำการส่ง user datagram ไปยังผู้ให้บริการ ทางด้านผู้ขอใช้บริการต้องรู้ที่อยู่ซ็อกเก็ตของผู้ให้บริการและที่อยู่ซ็อกเก็ตนี้ต้องถูกอ้างอิงใน sendto system call
- 4) ถ้าผู้ขอใช้บริการคาดหวังว่าจะได้รับข้อมูลจากผู้ให้บริการ มันจะต้องสั่งให้เรียก recvfrom ซึ่งเป็นสาเหตุให้ผู้ขอใช้บริการคอยจนกระทั่งข้อมูลจากผู้ให้บริการมาถึง
- 5) หลังจากนั้นขบวนการติดต่อระหว่างผู้ขอใช้บริการและผู้ให้บริการจะสิ้นสุดลง

2.5.5.2 Connection-Oriented Application Protocol

เป็นโปรแกรมประยุกต์ผู้ขอใช้บริการและผู้ให้บริการที่ใช้ Connection-Oriented Application Protocol จะใช้กับ Stream Socket เพื่อจะติดต่อสื่อสารบน TCP ดังรูป 2.7

การส่งข้อมูลโดยใช้ Connection-Oriented อันดับแรกผู้ให้บริการจะเตรียมตัวให้พร้อมด้วยตัวของมันเอง และรอการร้องขอการติดต่อจากผู้ขอใช้บริการ ส่วนผู้ขอใช้บริการจะร้องขอการติดต่อและส่งข้อมูลไปยังผู้ให้บริการภายใต้การทำงานของ TCP จะมีการตรวจสอบความผิดพลาด มีการส่งข้อมูลซ้ำใหม่และมีการควบคุมการไหลของข้อมูล

พิจารณากระบวนการของผู้ให้บริการจะใช้ Socket, Bind System Calls ต่อมาหากมีการติดต่อจะใช้ Accept System Call เพื่อยอมรับการติดต่อนั้น วิธี Connection-Oriented Application Protocol จะสมบูรณ์เมื่อสร้างการเชื่อมต่อสำเร็จ

พิจารณากระบวนการของผู้ขอใช้บริการจะใช้ Socket, Connect System Calls จะเห็นว่า มี Three way Handshake ซึ่งจะเริ่มเมื่อผู้ขอใช้บริการ Connect System Call ตามด้วยการแลกเปลี่ยนข้อมูลระหว่างผู้ขอใช้บริการและผู้ให้บริการ ดังนั้นการติดต่อนี้จะสมบูรณ์เมื่อเวลาที่ผู้ขอใช้บริการใช้ Connect System Call

2.5.5.2.1 Server System Calls

1) ผู้ให้บริการจะสั่งให้ Server System Call เข้าถึงข้อมูลของโปรโตคอลใน 5-tuple ที่สัมพันธ์กันซึ่งในที่สุด แล้วก็จะถูกสร้างเพื่อสนับสนุนการติดต่อสื่อสารระหว่างขบวนการผู้ขอใช้บริการและขบวนการผู้ให้บริการใน Connection-Oriented Application จะเห็นว่า Server System Call จะใช้ TCP

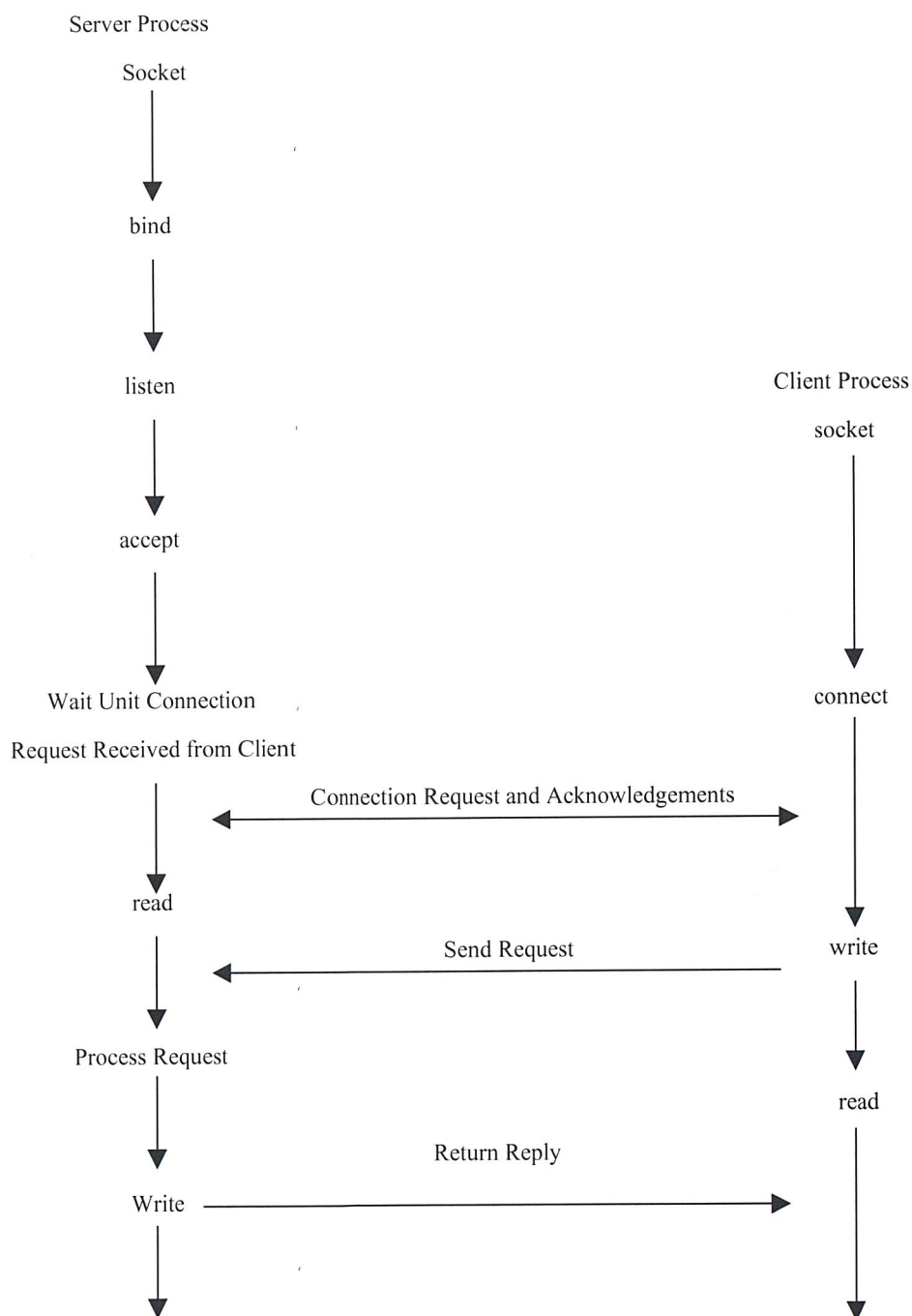
2) ผู้ให้บริการจะสั่งให้ Bind System Call (ระบุที่อยู่ซ็อกเก็ตของผู้ให้บริการ) ทำแทนในส่วนที่เหลือของข้อมูล ข่าวสารที่กำลังซุ่มแซม Server's half-association โดย half-association ของผู้ให้บริการบรรจุไปด้วยโปรโตคอลที่มีลักษณะเฉพาะตัว ในตอนนี้อินเทอร์เน็ต แอดเดรสและหมายเลขพอร์ตของผู้ให้บริการกำลังทำงานอยู่

3) ผู้ให้บริการจะสั่งให้ทำ Listen System Call ซึ่งแสดงว่าผู้ให้บริการกำลังรอการร้องขอจากผู้ขอใช้บริการเพื่อสร้างการติดต่อ TCP

4) ผู้ให้บริการจะสั่งให้ทำ Accept System Call เมื่อมีการร้องขอการสร้างการเชื่อมต่อ TCP ที่ได้รับจาก Client Accept System Call จะอ้างอิงซ็อกเก็ตแอดเดรสที่เกิดจากซ็อกเก็ตแอดเดรสของกระบวนการผู้ขอใช้บริการ เมื่อมีการร้องขอสำหรับการเชื่อมต่อที่ได้รับจริงๆ

5) ผู้ให้บริการจะสั่งให้ทำ Read System Call หรือ Recv System Call เพื่ออนุญาตให้ผู้ให้บริการรับข้อมูลจากผู้ขอใช้บริการบนการเชื่อมต่อ TCP

6) ผู้ให้บริการสามารถสั่งให้ทำ Write System Call หรือ Send System Call เพื่อที่จะอนุญาตให้ผู้ให้บริการส่งข้อมูลไปยังผู้ขอใช้บริการได้



รูป 2.7 วิธีการแบบ Connection-oriented Application Protocol โดยใช้ TCP

2.5.5.2.1 Client System Calls

1) ผู้ขอใช้บริการจะสั่งให้ Socket System Call เข้าถึงข้อมูลของโปรโตคอลสำหรับ Half – association หนึ่งใน Connection-oriented Application Protocol จะเห็นว่า Socket System Call จะใช้ TCP

2) ผู้ขอใช้บริการจะสั่งให้ Connect System Call ที่อ้างอิงชื่อเกิดแอดเดรสของผู้ขอใช้บริการที่จะสร้างการเชื่อมต่อ TCP กับผู้ให้บริการ Connect System Call เกิดได้จากการมีข้อมูลไหลกลับ และอันดับ 4 ระหว่างกระบวนการผู้ขอใช้บริการและกระบวนการผู้ให้บริการเพื่อที่จะ

สร้างกระบวนการ Three way Handshake ขึ้นอย่างสมบูรณ์เพื่อที่จะต้องการสร้างการติดต่อ ข้อมูลที่แลกเปลี่ยนระหว่างกระบวนการ Handshake รวมทั้งข้อมูลที่ต้องการเพื่อที่จะสร้างการเชื่อมต่อที่สมบูรณ์ระหว่างผู้ขอใช้บริการและผู้ให้บริการ ขบวนการผู้ขอใช้บริการต้องรู้ซ็อกเก็ตแอดเดรสของกระบวนการผู้ให้บริการ ในการออกคำสั่ง Connect System Call และเป็นชนิดที่ร้องขอ Networking software ที่จะเป็นแหล่งของอินเทอร์เน็ตแอดเดรสของผู้ขอใช้บริการและจัดหาหมายเลขพอร์ตที่ไม่ได้ใช้เอามาใช้ชั่วคราว

3) ผู้ขอใช้บริการจะสั่งให้ Write System Call หรือ Send System Call เพื่อที่จะส่งข้อมูลไปยังผู้ให้บริการบนการเชื่อมต่อ TCP ที่ถูกสร้างขึ้น

4) ผู้ขอใช้บริการสามารถสั่งให้ Read System Call หรือ Recv System Call เพื่อที่จะรับข้อมูลที่เข้ามาจากผู้ให้บริการ

5) หลังจากนั้นขบวนการติดต่อกันระหว่างผู้ขอใช้บริการและผู้ให้บริการจะสิ้นสุดลง

2.5.6 กระบวนการ System Call พื้นฐาน

System Call ในหัวข้อนี้ใช้บน TCP/IP และจะให้ค่าผลลัพธ์เป็น Integer ซึ่งค่านี้จะบอกได้ถึงความผิดพลาดที่เกิดขึ้นในการทำงานของซ็อกเก็ตนั้น

2.5.6.1 Socket System Call

ใช้สำหรับสร้างซ็อกเก็ตและสร้างการติดต่อระหว่าง 2 กระบวนการติดต่อกันมีการกำหนดรูปแบบดังนี้

Socket (family , type , protocol)

- family : แสดง Family ของโปรโตคอลที่ใช้ในการติดต่อ เช่น โปรโตคอล TCP/IP คือ AF_INET
- type : แสดงชนิดของซ็อกเก็ตที่ใช้ เช่น Stream Socket คือ SOCK_STREAM
- protocol : ซึ่งเฉพาะ โปรโตคอลที่ใช้ในการติดต่อทุกๆ ไปแล้วจะกำหนดให้เท่ากับศูนย์

ตัวอย่าง

การสร้าง Stream Socket สำหรับ Connection – Oriented Application Protocol โดยใช้ TCP

```
stream_sockfd=socket (AF_INET,SOCK_STREAM,0);
```

การสร้าง Datagram Socket สำหรับ Connectionless Application Protocol โดยใช้ UDP

```
dgram_sockfd=socket (AF_INET, SOCK_DGRAM,0);
```

2.5.6.2 Bind System Call

จะสร้างแอดเดรสและพอร์ตสำหรับการติดต่อ โดยที่ผู้ให้บริการจะบันทึกอินเทอร์เน็ทแอดเดรส และหมายเลขพอร์ตด้วยซอฟต์แวร์ของการติดต่อสื่อสาร TCP/IP ส่วนกระบวนการของผู้ขอใช้บริการที่มีการติดต่อแบบ Connectionless Application Protocol ทั่วๆ ไปจะใช้ Bind เพื่อบันทึกอินเทอร์เน็ทแอดเดรสและสร้างหมายเลขพอร์ตชั่วคราวที่ผู้ให้บริการสามารถใช้ส่งข้อมูลกลับมาที่กระบวนการผู้ขอใช้บริการ

ผู้ขอใช้บริการ สามารถร้องขอซอฟต์แวร์ของการติดต่อ TCP/IP เพื่อจะให้อินเทอร์เน็ทแอดเดรสของผู้ขอใช้บริการเอง และสร้างหมายเลขพอร์ตชั่วคราวที่ยังไม่ได้มีการใช้ระหว่างการติดต่อ แต่ในการติดต่อแบบ Connection-Oriented Application Protocol กระบวนการ ผู้ขอใช้บริการไม่จำเป็นที่จะต้องใช้ Bind ซึ่งถ้าผู้ขอใช้บริการไม่เรียกใช้ Bind

Connect System Call จะถูกเรียกใช้โดยอัตโนมัติเพื่อให้กระบวนการ Binding สมบูรณ์รูปแบบแสดงได้ดังนี้

```
bind ( socket , local_address , address_length );
```

- socket : แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call
- local address : แสดง local แอดเดรสที่ bind สร้างขึ้น
- address length : เป็นเลขจำนวนเต็มแสดงความยาวของแอดเดรสในหน่วยไบต์

ตัวอย่าง

```
bind ( sockfd,( struct sockaddr * )&own_addr,sizeof( own_addr ) );
```

2.5.6.3 Connect System Call

หลังจากมีการสร้างซ็อกเก็ตแอดเดรสแล้ว ซ็อกเก็ตเป้าหมายจะทำการติดต่อโดยใช้ Connect System Call ซึ่งทั่วๆ ไปแล้ว จะออกโดยกระบวนการผู้ขอใช้บริการ ในการติดต่อแบบ Connection-Oriented Application Protocol ซึ่งจะมีกระบวนการ Three way Handshake รวมทั้งแลกเปลี่ยนอัตโนมัติทั้งข้อมูลที่จำเป็นและเชื่อมต่ออย่างสมบูรณ์ระหว่างกระบวนการทั้งสอง แสดงรูปแบบดังนี้

```
connect ( socket , destination_address , address_length )
```

- socket : แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call
- destination address : แสดงแอดเดรสปลายทางที่ต้องการติดต่อ
- address length : เป็นเลขจำนวนเต็มแสดงความยาวของแอดเดรสปลายทางในหน่วยไบต์

ตัวอย่าง

```
connect ( sockfd,( struct sockaddr * )&serv_addr,sizeof( serv_addr ) );
```

2.5.6.4 Listen System Call

Listen System Call ออกโดยกระบวนการผู้ให้บริการในการติดต่อแบบ Connection-Oriented Application Protocol โดยใช้โปรโตคอล TCP โดยจะใช้ Listen System Call เพื่อบอกแก่ซอฟต์แวร์ในการติดต่อแบบ TCP/IP ซึ่งกระบวนการผู้ให้บริการจะยอมรับ (Accept) การร้องขอจากกระบวนการผู้ขอใช้บริการ สำหรับการสร้างการติดต่อบน TCPรูปแบบแสดงดังนี้

```
listen ( socket , queue_length )
```

- socket : แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call
- queue length : เป็นตัวเลขที่ชี้ให้เห็นว่า มีการร้องขอการติดต่อเพื่อรอคอยกระบวนการตอบรับจำนวนเท่าใด ตัวแปรนี้มีค่ามากที่สุดเท่ากับ 5

ตัวอย่าง listen (sockfd,5);

2.5.6.5 Accept System Call

ส่วนใหญ่แล้ว Accept System Call จะถูกออกโดยกระบวนการทางผู้ให้บริการใน Connection-Oriented Application Protocol เป็นผลให้กระบวนการทางผู้ให้บริการต้องรอการร้องขอที่จะเข้ามาจากผู้ขอใช้บริการ รูปแบบแสดงดังนี้

```
accept ( socket ,client_address,address_length )
```

- socket :แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call
- client address : แอดเดรสของผู้ขอใช้บริการ ที่ต้องการทำการติดต่อ
- address length : เป็นเลขจำนวนเต็มแสดงความยาวของแอดเดรสของผู้ขอใช้บริการในหน่วยไบต์

ผลที่ได้จะสร้างโครงสร้างข้อมูลซ็อกเก็ตใหม่ และคืนค่าตัวแปรของซ็อกเก็ตที่สามารถใช้หลังจากอ้างอิงโครงสร้างข้อมูลของซ็อกเก็ตใหม่

ตัวอย่าง

```
newsockfd = accept ( sockfd , (struct sockaddr * )&cli_addr,&clilen );
```

2.5.6.6 Write and Send System Call

Write and Send System Call ใช้สำหรับส่งข้อมูลไปยังอีกฝ่ายหนึ่ง Write และ Send คล้ายกันแต่ Send System Call จะประกอบด้วยตัวแปรที่มากกว่า มีรูปแบบดังนี้

```
write ( socket,bueffer, length )
```

```
send ( socket ,buffer_address,length,flags )
```

- socket : แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call
- buffer address : เป็นบัพเฟอร์ที่ใช้เก็บข้อมูลในการส่ง
- length : จำนวนไบต์ที่กำหนดเพื่อใช้ในการส่ง

- flags : เป็นตัวที่ใช้กำหนดเจาะจงทางเลือกพิเศษสำหรับการส่ง
ผลที่ได้จาก System Call นี้คือ จำนวน ไบต์ที่ส่งได้

ตัวอย่าง

```
nwritten = write ( sockfd,buffer,noctets );
nwritten = send ( sockfd,buffer,noctets,sendflag );
```

2.5.6.7 Sendto System Call

Sendto System Call คล้ายกับ Send และ Write System Call แต่จะใช้กับการติดต่อแบบ Connectionless Application Protocol บน UDP Sendto System Call จะสร้างการเชื่อมต่อระหว่าง 2 กระบวนการ ซึ่งการติดต่อนี้จะสมบูรณ์เมื่อข้อมูลได้ถูกส่งออกไปแล้ว มีรูปแบบดังนี้

```
sendto ( socket,buffer_address,length,flags,destination,address_length )
```

- socket : แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call
- buffer address : เป็นบัฟเฟอร์ที่ใช้เก็บข้อมูลในการส่ง
- length : จำนวนไบต์ที่กำหนดเพื่อใช้ในการส่ง
- flags : เป็นตัวที่ใช้กำหนดเจาะจงทางเลือกพิเศษสำหรับการส่ง
- destination address : แสดงแอดเดรสปลายทางที่ต้องการติดต่อ
- address length : เป็นเลขจำนวนเต็มแสดงความยาวของแอดเดรสปลายทางในหน่วยไบต์

ผลที่ได้จาก System Call นี้คือ จำนวนไบต์ที่ส่งได้จริง

ตัวอย่าง

```
nwritten = sendto ( sockfd ,buffer,noctets,sendflag,(struct,sockaddr * )
&serv_addr,sizeof(serv_addr));
```

2.5.6.8 Read and Recv System Call

Read and Recv System Call ใช้สำหรับการยอมรับ (Accept) ข้อมูลที่เข้ามาจากอีกฝ่ายหนึ่ง โดยที่ Recv System Call ทำหน้าที่คล้ายกับ Read System Call แต่มีตัวแปรที่มากกว่ารูปแบบแสดงดังนี้

```
read ( socket , buffer,length )
```

```
recv ( socket , buffer_address ,length,flags )
```

- socket : แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call
- buffer address : เป็นบัฟเฟอร์ที่ใช้เก็บข้อมูลในการรับ
- length : จำนวนไบต์ที่กำหนดเพื่อใช้ในการรับ
- flags : เป็นตัวที่ใช้กำหนดเจาะจงทางเลือกพิเศษสำหรับการรับ

ผลที่ได้จาก System Call นี้คือ จำนวน ไบต์ที่รับได้จริง
ตัวอย่าง

```
nread = read ( sockfd,buffer,bufflength );
nread = recv ( sockfd,buffer,bufflength,recvflag );
```

2.5.6.9 Recvform System Call

Recvform System Call คล้ายกับ Recv และ Read System Call แต่จะใช้กับการติดต่อแบบ Connectionless Application Protocol ที่ใช้ UDP Recvform System Call จะสร้างการเชื่อมต่อระหว่าง 2 กระบวน-การซึ่งการติดต่อนี้จะสมบูรณ์เมื่อได้รับข้อมูลเข้ามาเรียบร้อยแล้ว มีรูปแบบดังนี้

```
recvform ( socket,buffer_address,length,flags,source_address,address_length )
```

- socket : แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call
- buffer address : เป็นบัฟเฟอร์ที่ใช้เก็บข้อมูลในการรับ
- length : จำนวนไบต์ที่กำหนด เพื่อใช้ในการรับ
- flags : เป็นตัวที่ใช้กำหนดเจาะจงทางเลือกพิเศษสำหรับการรับ
- source address : แสดงแอดเดรสที่ต้องการส่งข้อมูล
- address length : เป็นเลขจำนวนเต็มแสดงความยาวของแอดเดรสในหน่วยไบต์

ผลที่ได้จาก System Call นี้คือ จำนวน ไบต์ที่รับได้จริง

ตัวอย่าง

```
nwritten = recvform ( sockfd,buffer,bufflength,recvflag, (struct,sockaddr * )
&serv_addr,&length );
```

2.5.6.10 Close System Call

การเรียกใช้ Close System Call นั้นจะทำหลังจากมีเชื่อมต่อที่สมบูรณ์แล้ว การเรียกใช้ System Call นี้ จะเกิดผลดีในการจัดเตรียมเนื้อที่ในหน่วยความจำ ในการติดต่อแบบ Connection Oriented Application Protocol ที่ใช้ TCP จะเรียก Close System Call หลังการติดต่อสมบูรณ์แล้วโดยที่มั่นใจว่าข้อมูลต่างๆที่ต้องการส่งนั้น ได้ถูกส่งออกไปเรียบร้อยแล้วก่อนที่การติดต่อจะสิ้นสุดลง มีรูปแบบดังนี้

```
close (socket)
```

- socket : แสดงถึงซ็อกเก็ตที่กำหนดขึ้นใน Socket System Call

2.6 ระบบมัลติมีเดีย

การให้คำจำกัดความของคำว่า มัลติมีเดีย ในขณะนี้ยังไม่มีที่สิ้นสุด ทั้งนี้เนื่องมาจากการพัฒนาเทคโนโลยีต่างๆยังไม่มีที่สิ้นสุด จึงขออนุญาตอย่างกว้างๆว่า คือ การรวบรวมสื่อต่างๆที่มีองค์ประกอบสำคัญใหญ่ๆ 3 อย่าง คือ ภาพ เสียง การโต้ตอบซึ่งกันและกัน

2.6.1 สื่อที่เป็นองค์ประกอบสำคัญของมัลติมีเดีย

2.6.1.1 ภาพ

เป็นสื่อที่มนุษย์เรานิยมใช้มาตั้งแต่อดีตและนิยมใช้มาจนถึงปัจจุบัน ความจริงแล้วสื่อที่เป็นเสียงนั้น เป็นสื่อแรกที่มนุษย์รู้จักและใช้ แต่เสียงนั้นมีข้อจำกัดอยู่ที่ระยะทางและภาษา การสื่อสารภายในเผ่าเดียวกันก็ไม่ใช่ปัญหาแน่ แต่หากจะต่างเผ่าออกไป การติดต่อสื่อสารระหว่างกันจะเริ่มยุ่งยาก หากสังเกตร่องรอยที่ถูกบันทึกไว้ โดยมนุษย์ในยุคแรกๆจะเป็นภาพวาดมา

ภาพในระบบมัลติมีเดียแบ่งออกเป็นประเภทใหญ่ๆได้ดังนี้ คือ

- 1) ตัวอักษร (Text)
- 2) รูปภาพ (Image)

และสามารถแบ่งออกตามลักษณะของภาพเป็นประเภทใหญ่ๆได้อีกคือ ภาพนิ่งและภาพเคลื่อนไหว

ถ้าหากพิจารณาถึงที่มาของภาพแล้ว สามารถจำแนกออกเป็น 2 ลักษณะใหญ่ๆ คือ

1) ภาพจากการจำลองแบบ (Imitated Image) ได้แก่ ภาพที่ถ่ายแบบมาจากของจริง โดยอาศัยเครื่องมือช่วยในการจำลองแบบ เช่น กล้องถ่ายภาพนิ่ง กล้องวิดีโอ เครื่องอ่านภาพ (Scanner) เป็นต้น

2) ภาพจากการสร้าง (Creative Image) เป็นภาพที่สร้างขึ้นจากจินตนาการของมนุษย์ เช่น ภาพวาด ภาพการ์ตูน เป็นต้น

เหตุที่เราไม่แยกตัวอักษรกับภาพออกจากกัน เพราะถ้ามองในส่วนของการใช้งานคอมพิวเตอร์ในปัจจุบันตัวอักษรจัดเป็นภาพจากการสร้างโดยที่การสร้างแบบตัวอักษร (Font) จะมีลักษณะการสร้างในเชิงรูปภาพ

2.6.1.2 เสียง

เสียงนับได้ว่าเป็นสื่อแรกที่มนุษย์ใช้ และเสียงหรือภาษาพูดก็เป็นสื่อที่มีความสำคัญไม่น้อยไปกว่าภาษาเขียน การผลิตภาษาเชิงภาพจะมีขั้นตอนยุ่งยากกว่าภาษาเสียง ซึ่งอาศัยหูเป็นสื่อกลางเพียงอย่างเดียวก็ใช้งานได้แล้ว อย่างไรก็ตาม เสียงก็มีข้อจำกัดที่ลดลงไปมากกว่าในอดีต เนื่องจากการพัฒนาสื่อทางเสียงได้รับการพัฒนาไปเร็วกว่าสื่อทางภาพ และที่สำคัญการพัฒนาสื่อทางเสียง จะไม่ร้ายแรงเท่ากับความผิดพลาดในเชิงภาพ ทั้งนี้ เพราะความผิดพลาดทางเสียงจะมีผลแค่เพียงทำให้ความชัดเจนของเสียงลดลง หรือเพี้ยนไปจากเดิม เช่น หูมกเกินไป แห่ลมกเกินไป

เป็นต้น แต่ถ้าเป็นความผิดพลาดในลักษณะของสื่อทางภาพแล้ว อาจถึงขั้นทำให้ใช้ไม่ได้เลยทีเดียว ระบบมัลติมีเดียจึงเป็นการประสานกันระหว่างสื่อทางภาพกับสื่อทางเสียงนั่นเอง

2.6.1.3 การโต้ตอบซึ่งกันและกัน

คำว่า โต้ตอบกันได้ (Interactive) หมายถึง แสดงผลการทำงานว่าถูกต้องหรือไม่ ใช้ได้หรือไม่ได้ในขณะนั้น ระบบมัลติมีเดียต้องมีการโต้ตอบกันได้จึงจะสมบูรณ์ จากลักษณะนี้เองที่ทำให้ขอบเขตของมัลติมีเดียกว้างขวางออกไป จำแนกได้ดังนี้

1) การตอบโต้กันได้ในลักษณะตัวเลือก มีการจำลองสถานการณ์ (Simulation) จากปัญหาหนึ่งไว้หลายรูปแบบ โดยพิจารณาจากเงื่อนไขที่แตกต่างกัน หากสามารถจำลองสถานการณ์ได้มาก และซับซ้อนครอบคลุมได้มากเท่าไร จะทำให้ระบบมัลติมีเดียนี้สมบูรณ์มากขึ้นเท่านั้น

2) การตอบโต้กันได้ในลักษณะการติดต่อสื่อสาร ใช้สมรรถนะของการคมนาคมที่ทันสมัยเข้าช่วย ผลลัพธ์ที่ได้จะปรับเปลี่ยนตามเหตุการณ์ที่เปลี่ยนไป เช่น เดิมถ้าต้องการทราบข้อมูลเพียงว่า มีโรงแรมอะไรบ้างเท่านั้น แต่ปัจจุบันเมื่อเราเลือกโรงแรมใดแล้ว เราจะทราบถึงขนาดว่า โรงแรมนั้นมีห้องพักกี่ห้อง และมีห้องว่างหรือไม่ด้วย เป็นต้น

ขอบเขตของระบบมัลติมีเดีย ได้รับการพัฒนากว้างขวางมากขึ้น ทำให้ข้อจำกัดในการใช้งานมีน้อยลง เดิมคิดกันแต่เพียงว่าระบบมัลติมีเดียจะเหมาะสำหรับงานนำเสนอเท่านั้น แต่ความจริงเราสามารถนำมัลติมีเดียมาประยุกต์ใช้งานได้มากมาย ได้แก่ ประยุกต์ใช้ในการศึกษา ด้านธุรกิจ ด้านการโฆษณา ด้านการให้ข้อมูลตามสถานที่ต่างๆ เป็นต้น

2.6.2 ส่วนประกอบของระบบมัลติมีเดีย

สิ่งที่จำเป็นสำหรับระบบมัลติมีเดีย แยกออกเป็น 2 ส่วน คือ

2.6.2.1 ฮาร์ดแวร์มัลติมีเดีย องค์ประกอบที่สำคัญของระบบมัลติมีเดีย คือ ภาพ เสียง และการตอบโต้กันได้ ส่วนที่เป็นฮาร์ดแวร์จึงต้องจำแนกออกไปตามองค์ประกอบดังกล่าว โดยมีคอมพิวเตอร์เป็นส่วนสำคัญคอมพิวเตอร์ที่ใช้งานในระบบมัลติมีเดียต้องแยกจากกัน ระหว่างการสร้างงานจำเป็นต้องใช้คอมพิวเตอร์ที่มีสมรรถนะสูง ถ้าเป็นระดับเวิร์คสเตชันจะดีมาก ซึ่งจะเป็นตัวเชื่อมโยงไปหาฮาร์ดแวร์อื่นๆ

1) ฮาร์ดแวร์สำหรับงานด้านภาพ อุปกรณ์ที่สำคัญและจำเป็นนอกจากคอมพิวเตอร์ ได้แก่ เครื่องอ่านภาพ (Scanner) ใช้สำหรับงานสำเนาภาพจากต้นฉบับที่เป็นภาพนิ่ง มีหลายแบบ หลายระดับ มีลักษณะเป็นภาพสองมิติ กล้องถ่ายภาพเชิงตัวเลข (Digital Camera) เป็นกล้องถ่ายภาพธรรมดาที่มีการเปลี่ยนส่วนรับภาพที่เป็นฟิล์มเป็นตัวเปลี่ยนสัญญาณภาพมาเป็นสัญญาณเชิงตัวเลข หรือเรียกว่า CCD (Charge Couple Device) ใช้ทำสำเนาภาพนิ่งได้ทั้ง 2 มิติ และ 3 มิติ กล้องถ่ายภาพวิดีโอ (Video Camera) ใช้งานถ่ายภาพเคลื่อนไหว และใช้สัญญาณเชิงตัวเลขและใช้ CCD เช่นเดียวกับกล้องถ่ายภาพเชิงตัวเลข การ์ดแปลงสัญญาณอนาล็อกเป็นสัญญาณเชิงตัวเลข (Digital

Card) เนื่องจากเครื่องเล่นวีดีโอและกล้องถ่ายภาพวีดิทัศน์ ส่วนใหญ่ให้สัญญาณแบบอนาล็อก การจะนำสัญญาณจากอุปกรณ์ดังกล่าวไปใช้งานกับคอมพิวเตอร์ จึงจำเป็นต้องแปลงสัญญาณจากอนาล็อกให้เป็นสัญญาณเชิงตัวเลข

2) ฮาร์ดแวร์สำหรับงานด้านเสียง ในส่วนของระบบเสียงนั้นเดิมเป็นแบบอนาล็อก แต่ในปัจจุบันได้มีการประยุกต์ และพัฒนาระบบเสียงให้ใช้สัญญาณเชิงตัวเลขได้ ฮาร์ดแวร์ที่ใช้สำหรับงานด้านเสียงจึงแทบจะประยุกต์ใช้กับระบบมัลติมีเดียได้ทันที ทั้งโดยตรงและโดยอ้อม อุปกรณ์มีดังนี้ เครื่องเล่นซีดี (CD-ROM) ถ้าไม่ต้องการบันทึกเสียงลงในคอมพิวเตอร์ก็สามารถจะแยกใช้ระบบเสียงจากภายนอกได้ โดยอาศัยการควบคุมการทำงาน วิธีที่สะดวกที่สุดจะเป็นการใช้เครื่องเล่นซีดี-รอม ที่ในปัจจุบันเล่นได้ทั้งระบบเสียงและระบบอ่านข้อมูลคอมพิวเตอร์ เพียงแต่ต้องบันทึกลงแผ่นซีดีมาก่อนจากนั้นจึงมาเปิดใช้งาน โดยอาศัยคำสั่งจากตัวเครื่องคอมพิวเตอร์เป็นต้น ความคมชัดที่หนึ่ง เสียงสำเร็จรูป (Clip Sound) เป็นระบบเสียงสำเร็จรูปในลักษณะไฟล์สัญญาณเชิงตัวเลขสามารถใช้งานได้ทันที ส่วนใหญ่จะเป็นเสียงเพลงหรือเอฟเฟ็กต์ต่างๆ

2.6.2.2 ซอฟต์แวร์มัลติมีเดีย แยกออกเป็น 2 ตอน คือ ตอนสร้างกับตอนใช้งาน

1) ซอฟต์แวร์สร้างงานระบบมัลติมีเดีย แยกออกเป็นสองส่วนคือ ซอฟต์แวร์สำหรับสร้างภาพและเสียง กับซอฟต์แวร์จัดระบบ

- ซอฟต์แวร์สำหรับสร้างภาพและเสียงมักมีการใช้งานและเป็นทีคุ้นเคยของผู้ใช้คอมพิวเตอร์อยู่แล้ว มีการตกแต่ง ตัดต่อ คัดแปลง สร้างใหม่ ซึ่งแยกกันทำเป็นขั้นเป็นตอนได้

- ซอฟต์แวร์จัดระบบมัลติมีเดีย เป็นซอฟต์แวร์ที่ควบคุมงานส่วนต่างๆ มาจัดลำดับ เพื่อให้มีการตอบโต้กันได้ เช่น บอกให้รู้ว่าถึงตอนนี้จะมีภาพและต่อไปจะมีเสียง หรือถ้ากดตรงนี้จะมีการพ่นภาพนี้ออกมา หรือมีเสียงนั้นเสียงนี้เป็นต้น ซอฟต์แวร์ประเภทนี้ยังไม่เป็นที่คุ้นเคยมากนัก ทำให้ยังมีน้อย ราคาจึงสูงและที่สำคัญยังต้องใช้กับคอมพิวเตอร์ที่มีสมรรถนะสูงด้วย

2) ซอฟต์แวร์ใช้งานระบบมัลติมีเดีย ระบบมัลติมีเดีย นั้นเกิดจากการสร้างขึ้นมา โดยผ่านทางระบบคอมพิวเตอร์ ทำให้ไม่เป็นการยากนักหากจะใช้งาน แต่มีข้อจำกัดอยู่ที่ความง่ายในการใช้งาน

2.6.3 เทคโนโลยีที่เกี่ยวข้องกับมัลติมีเดีย

เนื่องจากระบบคอมพิวเตอร์มัลติมีเดียเป็นการรวบรวมเทคโนโลยีหลายอย่างเข้าด้วยกัน เพื่อให้เกิดความสมบูรณ์ในการทำงาน เทคโนโลยีเหล่านั้น ได้แก่

1) การพัฒนาเทคโนโลยีในการบันทึกข้อมูลการทำงานของมัลติมีเดียประกอบไปด้วยภาพและเสียง

2) การพัฒนาด้านระบบคอมพิวเตอร์เครือข่าย สิ่งที่ระบบคอมพิวเตอร์มัลติมีเดียเข้าไปมีบทบาทร่วมกับระบบคอมพิวเตอร์เครือข่าย เช่น การติดต่อสื่อสารด้วยระบบ Electronics Mail ซึ่ง

เดิมเป็นการติดต่อที่เป็นลักษณะ Text Base เท่านั้น เป็นการนำสองเทคโนโลยีมารวมช่วยกัน ทำให้การติดต่อสื่อสารในระบบเครือข่ายคอมพิวเตอร์ทำได้ทั้งที่เป็นภาพและเสียง

3) การพัฒนาเทคนิคการย่อขนาดข้อมูล การย่อข้อมูลที่มีประสิทธิภาพเป็นปัจจัยสำคัญอย่างหนึ่งในการทำงานของระบบคอมพิวเตอร์มัลติมีเดีย

4) การพัฒนาไมโครคอมพิวเตอร์ การทำงานของคอมพิวเตอร์มัลติมีเดียเป็นการทำงานที่เกี่ยวข้องกับข้อมูลในปริมาณมหาศาล กระบวนการย่อและขยายขนาดข้อมูลจะต้องเกิดอย่างรวดเร็วและมากพอที่จะทำให้การติดต่อส่งข้อมูลระหว่างหน่วยความจำและอุปกรณ์ต่างๆ ไม่เกิดการหยุดชะงัก เพราะถ้าเกิดเหตุการณ์เช่นนี้จะทำให้การแสดงผลทั้งภาพและเสียงอาจเพี้ยนไปจากของจริงได้

5) การพัฒนาของจอภาพ

6) การพัฒนาอุปกรณ์ป้อนข้อมูล

7) การพัฒนาซอฟต์แวร์ ส่วนหนึ่งที่ทำให้โลกคอมพิวเตอร์มัลติมีเดียเป็นจริงก็คือ การพัฒนาซอฟต์แวร์ให้มีประสิทธิภาพสูงและมีการใช้งานได้ง่ายขึ้น การพิจารณาเลือกซอฟต์แวร์เพื่อมาทำงานด้านมัลติมีเดียอาจพิจารณาได้จาก

- ความง่ายในการใช้งาน
- ความสามารถในการนำเสนองาน
- ความสามารถในการติดต่อกับผู้ใช้
- ความสามารถในการใช้ฟังก์ชันในการคำนวณ
- ความสามารถในการใช้งานร่วมกับโปรแกรมอื่นๆ
- มีLibrary สนับสนุนการทำงาน
- ความสามารถในการทำเอกสารประกอบโปรแกรม
- ความสามารถในการส่งแอปพลิเคชันที่เสร็จแล้วให้ผู้ใช้

2.6.4 Microsoft Audio Video Interleave (AVI)

เป็น Format ของไฟล์ที่ใช้ในการ Capture, แก้ไข, PlayBack ลำดับของภาพและเสียง โดยทั่วไปไฟล์ AVI จะเก็บ stream ของข้อมูลหลากหลายรูปแบบ แต่ส่วนใหญ่จะเก็บข้อมูลของภาพและเสียงเป็นหลัก ไฟล์ AVI พื้นฐานจะเก็บเฉพาะลำดับของรูปภาพ จนถึงไฟล์ AVI แบบพิเศษที่สามารถเพิ่มเติม Control Track หรือ MIDI Track จะทำให้สามารถควบคุมอุปกรณ์ภายนอก เช่น เครื่องบันทึก Video Disc ได้ และ MIDI Track ทำให้สามารถเล่นเสียง MIDI เป็น Background ได้ แต่ต้องใช้โปรแกรมเฉพาะจึงจะสามารถใช้ความสามารถเหล่านี้ได้ โปรแกรมประยุกต์ทั่วไปไม่สามารถใช้ความสามารถเหล่านี้ได้ จึงแสดงผลเป็นภาพและเสียงได้เท่านั้น

บทที่ 3

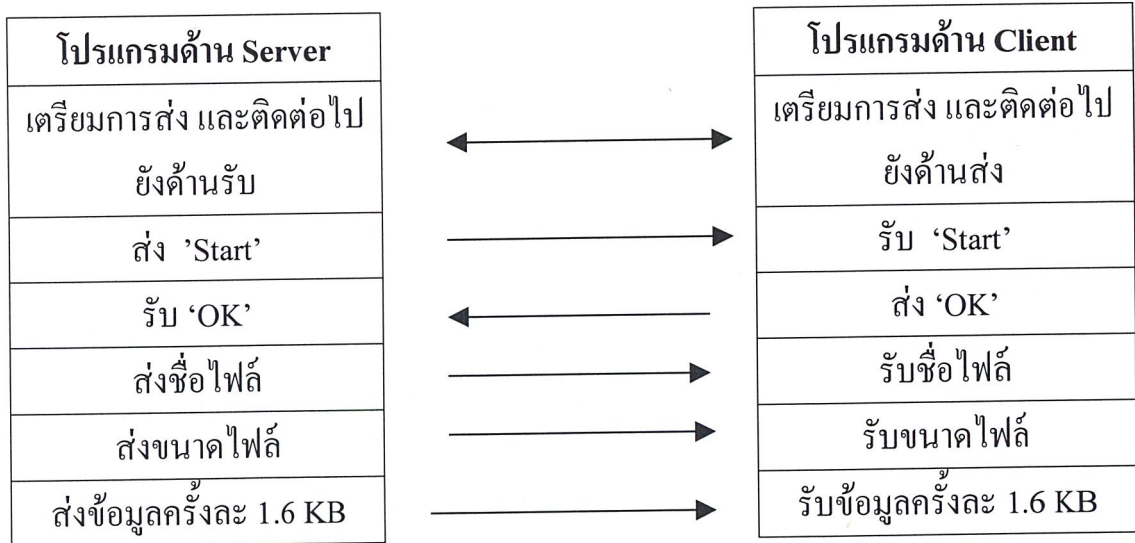
การออกแบบการทำงาน

โปรแกรมที่สร้างขึ้นมาจากโครงการนี้จะสามารถแสดงได้เป็นไดอะแกรมการรับส่งข้อมูลดังรูปในงานจะมีโปรแกรมหลักที่จะทำการเรียกโปรแกรมย่อย ๆ ซึ่งแต่ละโปรแกรมย่อย ๆ นั้นจะทำหน้าที่แตกต่างกันตามงานที่ได้รับ โดยในบทนี้จะทำการอธิบายลำดับขั้นตอนการทำงานของโปรแกรมย่อยที่สำคัญและเกี่ยวข้องกับโครงการนี้พร้อมทั้งแสดงผังงานของแต่ละส่วนโปรแกรมย่อยเหล่านี้ (Flowchart) เพื่ออธิบาย

โปรแกรมมีอยู่ 2 ส่วน

- 1) โปรแกรมที่ทำการส่ง (Server)
- 2) โปรแกรมที่ทำการรับ (Client)

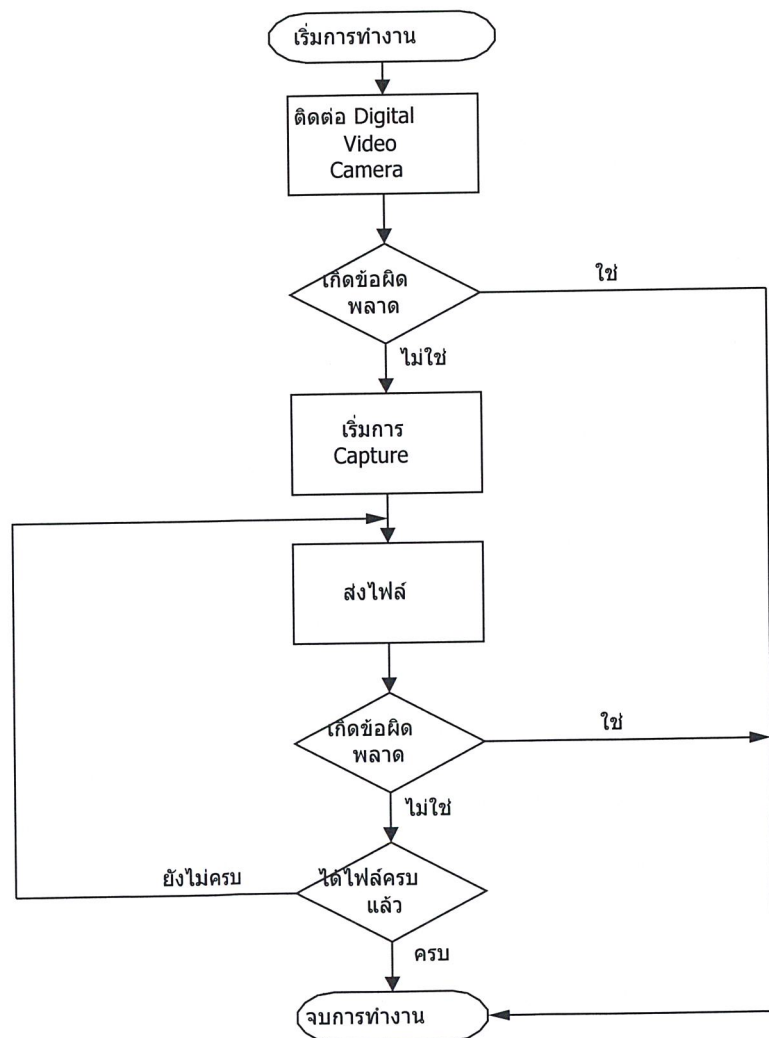
เนื่องจากการทำงานของโปรแกรมเกี่ยวข้องกับการส่งไฟล์และรับไฟล์ผ่านเครือข่าย โปรแกรมทั้งในด้านการรับและการส่งต้องทำงานควบคู่กัน



รูปที่ 3.1 แสดงการติดต่อกันระหว่าง Server และ Client

3.1 โปรแกรมด้าน Server

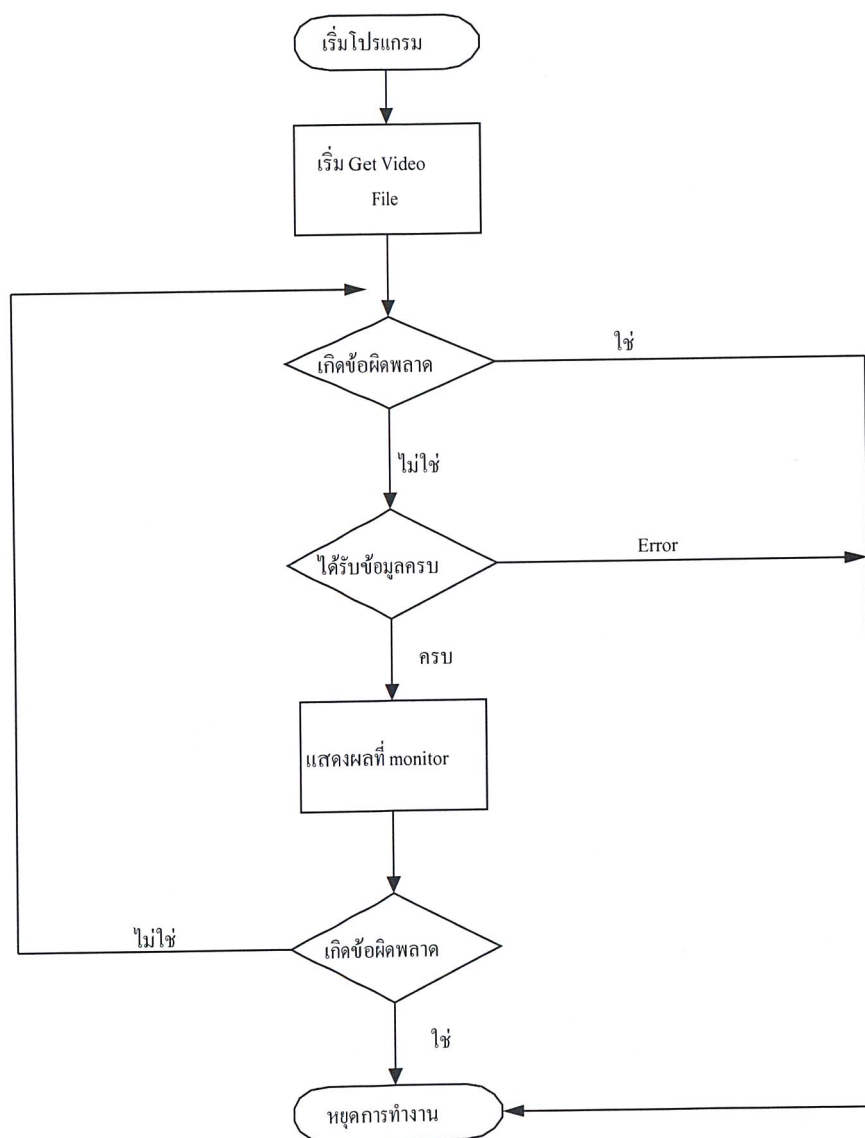
จะทำการ Connect กับกล้องวิดีโอ ก่อน หลังจากนั้นก็จะทำการ Capture ไฟล์ภาพวิดีโอ โดยที่แต่ละไฟล์จะเก็บไฟล์ภาพวิดีโอไว้ประมาณ 20 วินาที หลังจากครบ 20 วินาทีแล้ว ก็จะทำการเปลี่ยนไฟล์ใหม่ โดยจะทำการบวกเลขลำดับของไฟล์ไปอีก 1 ค่า เช่น จากไฟล์แรกที่ Save ชื่อ Test.avi ไว้ที่ c:\TuiTonMax\ ไฟล์ต่อไปก็จะเปลี่ยนเป็น Test1.avi ซึ่งเราจะเก็บประมาณ 10 ไฟล์ (คือเก็บถึง Test9.avi) หลังจากทีเลขลำดับไฟล์มีค่าเป็น 10 แล้วก็จะวนมาเก็บทับกับไฟล์ Test1.avi ใหม่ซึ่งจะทำการเก็บอย่างนี้ไปเรื่อยๆ จนกว่าจะสั่งให้หยุดการ Capture หรือ หยุดการติดต่อกับกล้อง โดยที่เราสามารถแก้ไขชื่อไฟล์ที่ต้องการเก็บได้โดยเปลี่ยนชื่อไฟล์ที่ Textbox หรือถ้าต้องการให้ Save ทับไฟล์ที่มีอยู่แล้วก็ให้ทำการ Browse หาชื่อไฟล์ที่ต้องการ



รูปที่ 3.2 แสดงแผนภาพการทำงาน

3.2 โปรแกรมด้าน Client

จะทำการรับไฟล์ภาพการสอนจากเครื่อง Server โดยจะทำการโหลดไฟล์ที่เป็นไฟล์ล่าสุดที่เครื่อง Server ทำการ Capture เอาไว้และเมื่อได้ไฟล์ภาพการสอนจนครบทั้งไฟล์แล้วก็จะทำการเล่นและในขณะที่ทำการเล่นไฟล์ที่ได้รับมานั้นก็จะทำการติดต่อไปยังเครื่อง Server เพื่อทำการดึงไฟล์ต่อไปมาทันที และเมื่อไฟล์การสอนที่ทำการเล่นอยู่นั้นเล่นเสร็จสิ้นก็จะทำการเล่นไฟล์ตัวต่อไปทันที (หากไฟล์ลำดับต่อไปยังโหลดไม่เสร็จสมบูรณ์ก็จะทำการรอกจนกว่าไฟล์นั้นจะได้รับจนครบแล้วจึงเล่นต่อไป)



รูป 3.3 flow chart ตัว Client

บทที่ 4

การทดลองและพัฒนาโปรแกรม

การทดลองและการพัฒนาโปรแกรมมีประโยชน์มากเนื่องจากหากเราเขียนโปรแกรมขึ้นมาแล้วไม่มีการทดลองมันก็อาจทำให้เกิดปัญหาต่าง ๆ มากมาย หรือหากไม่มีการพัฒนามันก็จะไม่มีประโยชน์อันใดเลยที่จะทำการสร้างโปรแกรมนั้นเพราะแสดงว่าโปรแกรมไม่มีส่วนช่วยให้เกิดความสะดวกสบาย ฉะนั้นในการเขียนโปรแกรมใดนั้นต้องมีการทดลองและพัฒนาไปด้วยจึงจะเกิดประโยชน์ดังที่ผู้จัดทำตั้งใจ

4.1 เครื่องมือที่ใช้ในการพัฒนาโปรแกรม

สภาพแวดล้อมในการพัฒนาโปรแกรมตัวนี้ส่วนที่เราจะลืมไม่ได้ก็คือ ตัวอุปกรณ์ที่ใช้ (Hardware) เนื่องจากแต่ละโปรแกรมนั้นจะสามารถทำงานได้เต็มที่ก็ต่อเมื่อสภาพแวดล้อมต่าง ๆ นั้นภายในตัวเครื่องสามารถที่จะสนับสนุนการทำงานได้อย่างเต็มที่ อย่างน้อยที่สุดต้องสามารถรองรับได้เพียงพอต่อความต้องการของตัวโปรแกรม ในที่นี้การพัฒนาโปรแกรมนั้นจะอาศัยเครื่องที่มีอยู่คั้งนั้นสภาวะแวดล้อมต่ำที่สุดในการพัฒนานั้นก็จะยึดเอาส่วนนี้เป็นหลัก

ในการทำงานของตัวโปรแกรมนั้นดังที่กล่าวไปแล้วว่าประกอบไปด้วย 2 ส่วน คั้งนั้นตัวอุปกรณ์จึงแบ่งเป็น 2 ทางคือ Server กับ Client

4.1.1 ฮาร์ดแวร์และซอฟต์แวร์สำหรับเซิร์ฟเวอร์

- CPU Intel Pentium III 500 MHz
- RAM 12 8 MB
- Network card
- Windows 98 se
- กล้องวีดีโอติดจอของ One Point Multimedia รุ่น V6010

4.1.2 ฮาร์ดแวร์และซอฟต์แวร์สำหรับไคลเอนต์

- CPU Intel Pentium III 500 MHz
- RAM 12 8 MB
- Network card
- Windows 98 se

การทำงานต่าง ๆ นั้นอยู่ภายใน Local Network

4.2 ส่วนประกอบต่างๆ ของโปรแกรม

การทำงานของโปรแกรมจะเริ่มต้นที่หน้าจอหลัก การทำงานให้ผู้ใช้ทำการเลือกความต้องการให้คอมพิวเตอร์ที่ใช้งานนั้นเป็นระบบใด โดยเราสามารถเลือกได้เพียง 2 ระบบการทำงานอันได้แก่

4.2.1 Server จะเป็นส่วนที่มีการติดต่อกับตัวกล้องวิดีโอจิจิตอลและเป็นส่วนที่ทำการบันทึกไฟล์จากกล้องวิดีโอจิจิตอลมาทำการแบ่งแยกออกเป็นไฟล์ย่อยแล้วทำการส่งเมื่อมีการร้องขอจากทางฝั่ง Client

การทำงานของ Server เป็นไปตาม Procedure ที่จัดการมีดังนี้

OptServer_Click

เมื่อเรา Click เลือกสถานะ Server โปรแกรมจะทำการเรียก Procedure OptServer_Click ซึ่งจะทำการตรวจสอบไฟล์ที่ชื่อ Count.txt ใน Folder ที่ชื่อ C:\TuiTonMax ว่ามีการสร้างแล้วหรือยัง ถ้าไม่ปรากฏ จะทำการสร้างไฟล์ที่ชื่อ Count.txt แล้วทำการเก็บลงใน Folder ที่ชื่อ C:\TuiTonMax หลังจากนั้นจะทำการเปิดไฟล์ Count.txt ขึ้นมาเพื่อทำการบันทึกค่า “0” ลงในไฟล์ Count.txt เพื่อเป็นค่า Default ที่แสดงให้เห็นว่ายังไม่เริ่มทำการ Capture หรือ ไฟล์แรกที่ทำ การ Capture ยังไม่เสร็จ ซึ่งถ้ามีเครื่อง Client มาร้องขอไฟล์จะรู้ว่าเครื่อง Server ยังไม่สามารถส่งไฟล์ไปให้ได้

CmdConnect_Click

เราจะกดปุ่ม Connect เพื่อทำการติดต่อกับกล้องวิดีโอ โดยถ้าเรามีกล้องหลายตัว เราสามารถเลือกที่จะติดต่อกับกล้องได้ตามชื่อ Device ที่มีแสดงใน ComboBox ซึ่งถ้าเลือก Device ที่ไม่ได้ต่อกับเครื่องก็จะไม่สามารถทำงานได้

CmdCap_Click

เมื่อกดปุ่ม Capture จะทำการเรียก Procedure CmdCap_Click ซึ่งจะทำการตรวจสอบค่าของ เวลาที่ใช้ในการ Capture ซึ่งมี Default เป็น 5 วินาที แต่ถ้าต้องการเปลี่ยนแปลงเวลาที่จะ Capture ก็ให้ใส่ค่าที่ช่อง Time Limit (s) โดยจะเก็บค่าเป็นวินาที แล้วจะไปทำการแปลงหน่วยจากวินาทีเป็นมิลลิวินาทีแล้วไปเก็บในตัวแปรที่ชื่อ TimeLimit แล้วทำการส่งค่าไปให้ Component Timer1 โดยทำการกำหนดค่า Interval ของ Timer1 เพื่อให้มีการรอตตามค่า TimeLimit แล้วค่อยเรียก Procedure Timer1_Timer เพื่อจะทำงานต่อไป ส่วนชื่อไฟล์วิดีโอจะมี Default เป็นชื่อ Test.avi เก็บอยู่ใน Folder ที่ชื่อ C:\TuiTonMax แต่ถ้าต้องการเปลี่ยนชื่อไฟล์ก็สามารถ Browse หาชื่อไฟล์อื่นได้

หรือ ถ้าเราทราบชื่อไฟล์ และ Path ที่ต้องการบันทึกไฟล์นั้นๆ ก็สามารถพิมพ์ลงใน TextBox ได้เช่นกัน แล้วทำการเริ่ม Capture ไฟล์นั้น

Timer1_Timer

หลังจากครบตามเวลาที่กำหนดให้กับ Timer1 แล้วจะทำการ Set ค่า Interval ของ Timer1 ให้มีค่าเป็น 0 เพื่อไม่ให้เกิดการวนซ้ำตามค่าของ Interval หลังจากนั้นจะทำการตรวจสอบว่ามีการกดปุ่ม End Capture หรือไม่

ถ้าไม่มีการกดปุ่ม End Capture จะสั่งให้หยุดการ Capture ไฟล์นั้น แล้วทำการแสดงรายละเอียดของไฟล์ที่เพิ่ง Capture เสร็จใน Capture Stream Options Frame ดังนี้ คือ Total Frames, Time, Total และ Frames Dropped หลังจากนั้นจะทำการบันทึกชื่อไฟล์ที่เพิ่ง Capture เสร็จรวมทั้ง Path ที่เก็บ เช่น ไฟล์ที่ทำการ Capture เสร็จ ชื่อ Test.avi เก็บใน Folder ชื่อ TuiTonMax ใน Drive C ก็จะบันทึกว่า C:\TuiTonMax\Test.avi ลงใน Count.txt ซึ่งจะทำการบันทึกทับข้อความเดิมเพื่อป้องกันความผิดพลาดที่ Client จะมาดึงไฟล์ผิดไป หลังจากนั้นจะทำการเรียก Procedure runfile เมื่อ Procedure runfile ทำงานเสร็จแล้ว Return กลับมาที่ Timer1_Timer ก็จะทำการกำหนดค่า Interval ของ Timer2 เป็น 2000 เพื่อให้มีการรอ 2 วินาทีแล้วค่อยเรียก Procedure Timer2_Timer ทำงานต่อไป

ถ้ามีการกดปุ่ม Disconnect โดยไม่ได้กดปุ่ม End Capture ก็จะทำการสร้างไฟล์ Count.txt ใน C:\TuiTonMax ขึ้นใหม่

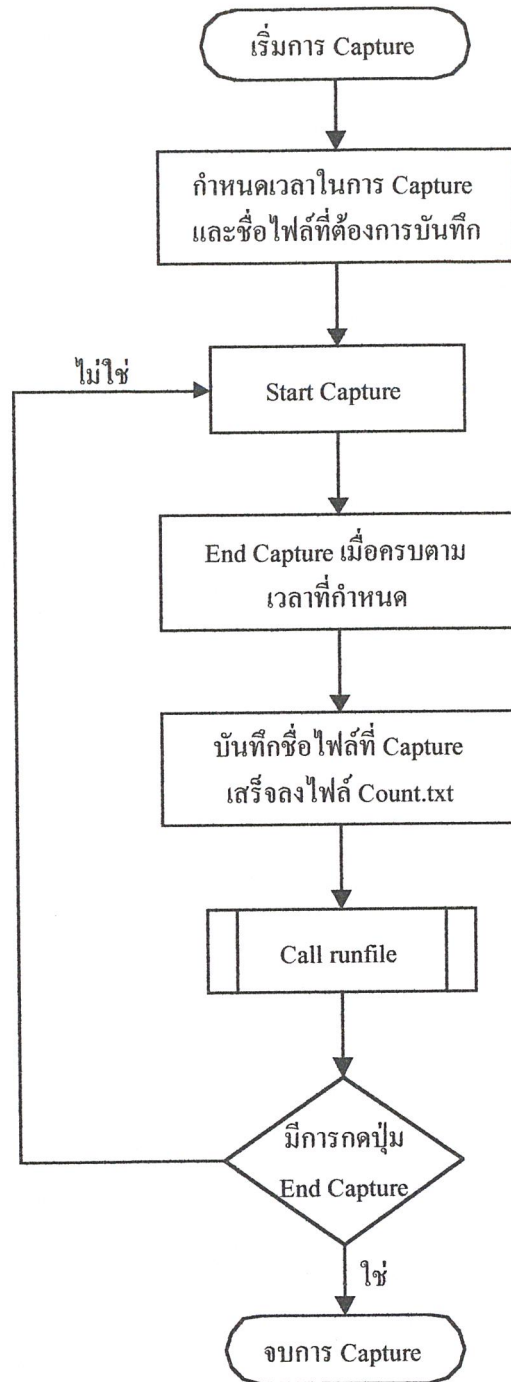
runfile

จะเป็น Procedure ที่ทำการเปลี่ยนชื่อไฟล์ โดยการใส่เลขต่อท้ายไฟล์ โดยจะทำการตรวจสอบเลขไฟล์จากตัวแปรชื่อ fcount ก่อนว่ามีค่าเท่ากับ 6 แล้วหรือไม่ ถ้าใช่จะทำการเปลี่ยนค่า fcount ให้มีค่าเท่ากับ 1 แต่ถ้าไม่ใช่ก็จะทำการเพิ่มค่า fcount ขึ้นไปอีก 1 หลังจากนั้นจะทำการตัดเอาอักษรตัวสุดท้ายของไฟล์นั้นมาตรวจว่าเป็นตัวเลขหรือไม่ ถ้าเป็นตัวเลขก็จะทำการตัดอักษรตัวนั้นทิ้ง เช่น ถ้าไฟล์ชื่อ Test ก็จะไม่ทำอะไร แต่ถ้าเป็น Test1 จะเห็นว่าตัวอักษรตัวสุดท้ายเป็น 1 ก็ทำการตัดเอา 1 ทิ้งคงเหลือแค่ Test หลังจากนั้นจะทำการรวมชื่อไฟล์นั้น กับ fcount และ ".avi" แล้วทำการกำหนดค่าให้กับ EdtFile ซึ่งเป็น TextBox เช่น จาก Test1 เราตัดเหลือ Test ก็จะนำไปรวมกับ fcount ที่มีค่าเท่ากับ 2 จะได้เป็น C:\TuiTonMax\Test2.Avi นี้ก็คือชื่อไฟล์ถัดไปที่จะทำการบันทึก หลังจากนั้นก็จะทำการตรวจสอบจำนวนไฟล์ที่ทำการบันทึกไว้ทั้งหมดว่ามีจำนวนเป็นกี่เท่าของเวลาที่ทำการ Capture +1 ถ้าเป็นจำนวนคู่จะทำการกำหนดค่า Interval ของ Timer4 เป็น 2000 เพื่อให้มีการรอ 2 วินาที แต่ถ้าเป็นจำนวนคี่จะทำการกำหนดค่า Interval ของ Timer4 เป็น 2000 เพื่อให้มีการรอ 1 วินาที ที่ต้องทำเช่นนี้ เนื่องจากบางครั้งเครื่อง Client มาร้องขอไฟล์ไม่ทัน ทำให้ไฟล์

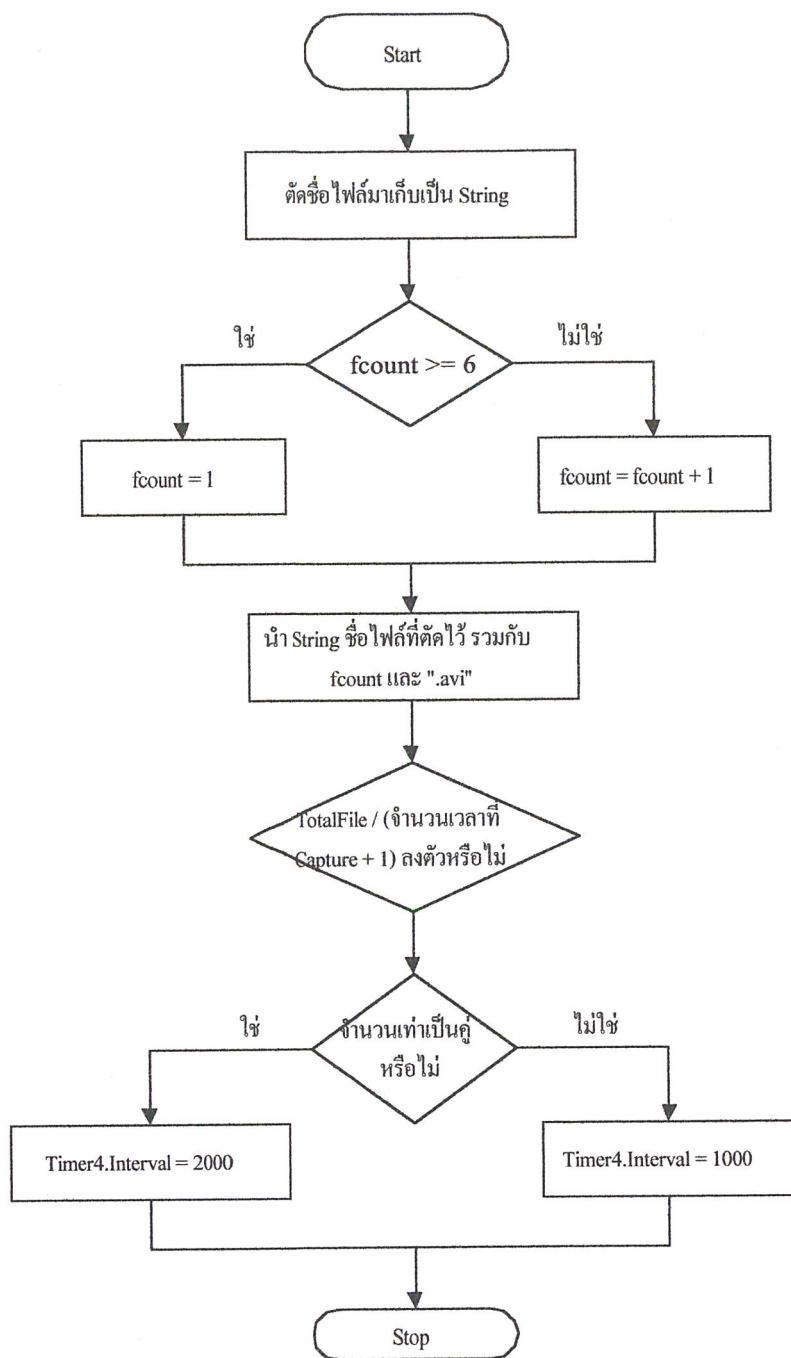
ที่ส่งไปให้กับเครื่อง Client อาจไม่เรียงตามลำดับ ซึ่งการทำเช่นนี้จะช่วยลดการกระโดดของไฟล์ได้บ้าง

Timer1_Timer

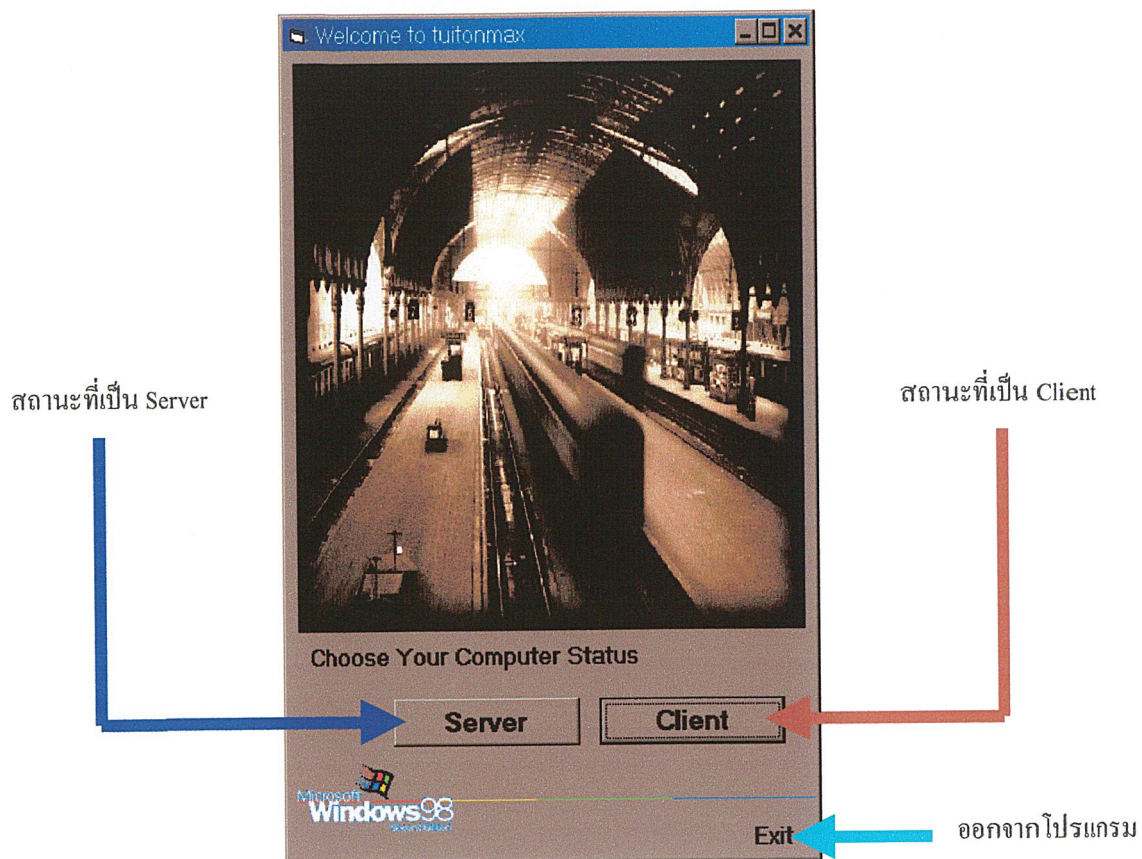
หลังจากครบตามเวลาที่กำหนดให้กับ Timer2 แล้วจะทำการ Set ค่า Interval ของ Timer2 ให้มีค่าเป็น 0 เพื่อไม่ให้เกิดการวนซ้ำตามค่าของ Interval หลังจากนั้นจะทำการเรียก Procedure CmdCap_Click เพื่อให้ทำการ Capture ไฟล์ถัดไป โดยจะวนทำเช่นนี้ไปเรื่อยๆ จนกว่าจะกดปุ่ม End Capture หรือ Disconnect ก็จะหยุดการ Capture



รูปที่ 4.1 โฟล์ชาร์ท ของ Server



รูปที่ 4.2 โฟลว์ชาร์ท ของโปรแกรมย่อย runfile



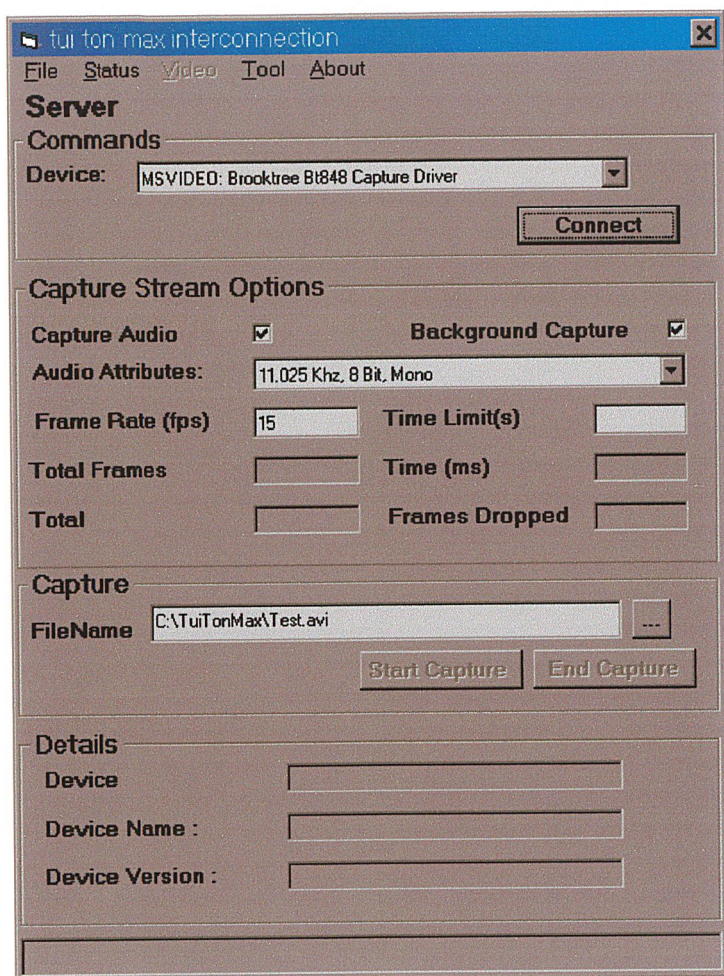
รูปที่ 4.3 แสดงหน้าจอหลัก

จากรูปที่ 4.3 เราต้องทำการเลือกว่าจะเป็นสถานะอะไรเพื่อที่จะสามารถเข้าไปดำเนินการทำงานต่อไป

จะเห็นว่าหากไม่ต้องการเข้าทำงานในส่วนของโปรแกรมนี้สามารถเลือกออกจากโปรแกรมโดยอาศัยกดตรงคำว่า "Exit" ก็จะสามารถออกจากโปรแกรมได้ทันที

จากตัวโปรแกรมเราจะเห็นได้ว่าสามารถแบ่งโปรแกรมเป็น 2 ส่วนคือ ส่วน Server กับ ส่วน Client ซึ่งเราจะกล่าวเป็นลำดับ

เมื่อเรากดปุ่ม Server จะปรากฏหน้าจอการทำงานในส่วนของ Server ดังรูปที่ 4.4

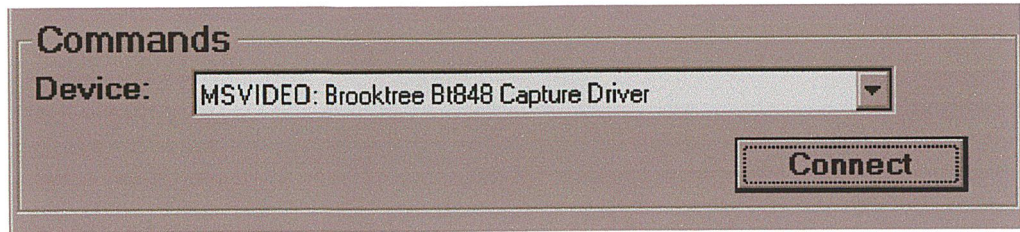


รูปที่ 4.4 แสดงหน้าจอของ Server

หน้าจอในส่วนของ Server นี้จะมีการแบ่งการควบคุมออกเป็นส่วนๆ เพื่อแยกให้เห็นการทำงานได้อย่างเด่นชัด โดยมีส่วนการควบคุมดังนี้

4.2.1.1 Frame Commands

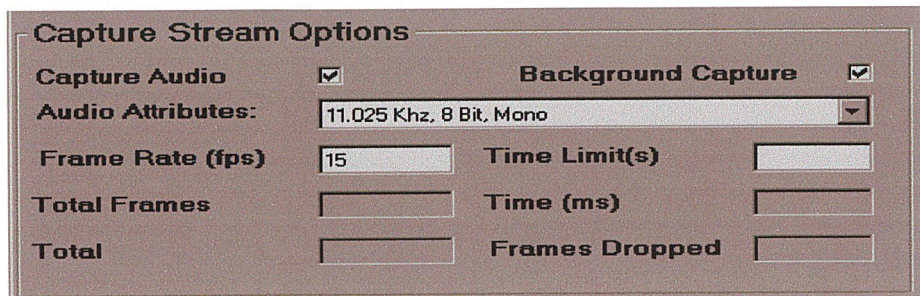
เป็นส่วนที่ทำการแสดง Device ของตัวกล้องวิดีโอจิจิตอลว่ามีอะไรบ้าง โดยจะเก็บเป็น Dropdown listbox โดยจะแสดงกล่องที่มีการติดตั้งไว้ในตัวเครื่องคอมพิวเตอร์ทั้งหมด ผู้ใช้สามารถที่จะเลือกใช้ตัวไหนก็ได้ ที่ปุ่ม Connect นั้นจะเป็นส่วนที่ทำหน้าที่ติดต่อกับกล้องจิจิตอลให้แสดงภาพเคลื่อนไหว Frame Command แสดงดังรูปที่ 4.5



รูปที่ 4.5 แสดง Frame Commands

4.2.1.2 Frame Capture Stream Option

เป็นส่วนที่ทำการบอกรายละเอียดการทำงานของกล้องดิจิทัลขณะที่ทำการ Capture ภาพเคลื่อนไหว ดังที่เห็นได้จากรูปที่ 4.6



รูปที่ 4.6 แสดง Frame Capture Stream

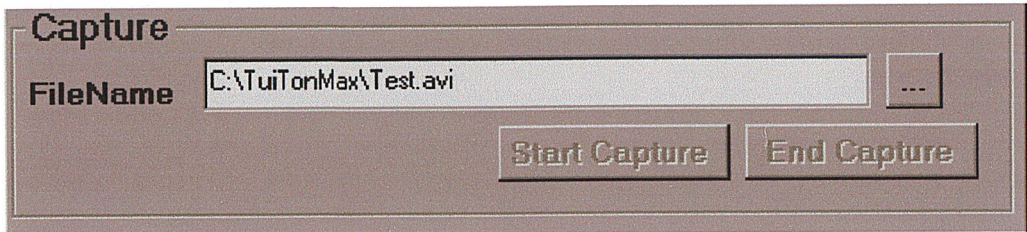
รูปที่ 4.6 ประกอบด้วย

- 1) Capture Audio เลือกเพื่อทำการเก็บเสียงไปพร้อมกับการเก็บไฟล์ภาพ
- 2) Background Capture
- 3) Audio Attributes แสดงลักษณะของไฟล์เสียงที่ทำการเก็บขณะบันทึกภาพ
- 4) Frame Rate (fps) เป็นจำนวนเฟรมในหนึ่งวินาทีที่จะทำการ Capture ภาพ
- 5) Time Limit (s) เวลาในการ Capture ในแต่ละครั้ง
- 6) Total Frame จำนวน Frame ทั้งหมดที่เราทำการ Capture
- 7) Time (ms) เวลาในการ capture ทั้งหมด
- 8) Total จำนวน Frame ทั้งหมดที่ทำการ Capture ได้จริง
- 9) Frame Dropped

จากรูป 4.6 เราสามารถที่จะดูสถานะของผลการทำงานของการ capture ในฝั่ง Server ได้โดย อาศัย Frame Capture Stream Option นี้ได้

4.2.1.3 Frame Capture

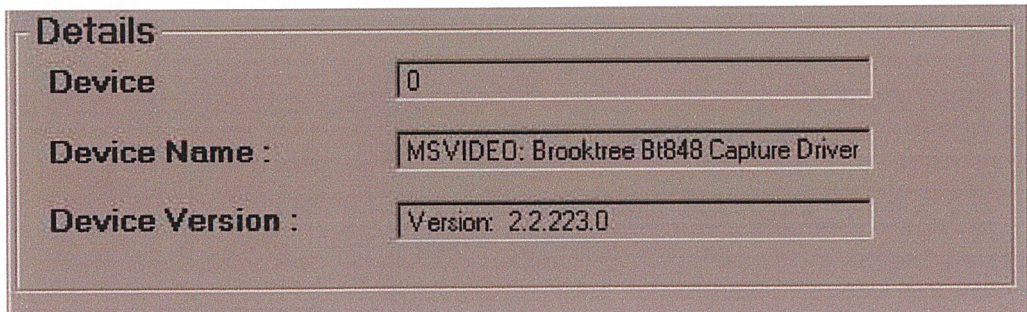
เป็นส่วนที่ทำการเริ่มการ Capture และ สิ้นสุดการ Capture จะมีส่วนที่เป็น Text box เพื่อให้กรอกชื่อไฟล์และตำแหน่งที่จะเก็บไฟล์นั้นตามที่เรต้องการ หรือทำการเลือกโดยการกดปุ่มทางขวามือก็ได้ ดังรูปที่ 4.7



รูปที่ 4.7 แสดง Frame Capture

4.2.1.4 Frame Device Detail

เป็นส่วนที่ทำการบอกว่าตัวกล้องวิดีโอดิจิตอลนั้นมี Index ที่เท่าไร มีชื่ออย่างไร และมี Version ที่เท่าไร ดังรูปที่ 4.8



รูปที่ 4.8 แสดง Frame Detail

4.2.1.5 Frame Status

เป็นส่วนที่อยู่ด้านล่างสุดจะเป็นส่วนที่แสดงสถานะของกล้องดิจิตอลในขณะนั้น ดังรูปที่ 4.9 เป็นสถานะขณะที่กล้องดิจิตอลกำลังทำการ capture ภาพการสอน



รูปที่ 4.9 Frame Status

4.2.2 Client จะเป็นส่วนที่ส่งคำร้องขอไปยังส่วน Server เพื่อที่จะรับไฟล์ภาพเคลื่อนไหวจากกล้องวิดีโอที่ติดจติดมาทำการแสดงผลที่หน้าจอทางฝั่งของ Client

การทำงานของ Client เป็นไปตามลำดับดังนี้

OptClient_Click

จะทำการลบ Folder ที่ชื่อ TuiTonMax ใน Drive C เสียก่อนเพื่อให้แน่ใจว่าใน TuiTonMax ยังไม่มีไฟล์ใดๆ อยู่ เพราะว่าถ้ามีไฟล์ Count.txt อยู่ใน Folder นี้จะทำให้เวลาไปถึงไฟล์ Count.txt ไม่สามารถทำได้ หลังจากนั้นก็จะทำการสร้าง Folder ที่ชื่อ TuiTonMax เก็บไว้ใน Drive C ขึ้นมาใหม่ เพื่อใช้เป็นที่เก็บไฟล์ที่ได้จากเครื่อง Server

sendfile_Click

เมื่อต้องการร้องขอไฟล์การสอนจากเครื่อง Server ให้ทำการกดปุ่มนี้ โดยจะทำการติดต่อกับเครื่อง Server ตาม IP Address, User Name และ Password ที่อยู่ใน Network System Frame ซึ่งทางฝั่ง Client ต้องทำการกรอกค่าต่างๆ ข้างต้นให้ครบถ้วนก่อนที่จะกดปุ่มนี้ มิฉะนั้นจะขึ้น Dialog แสดงว่า "ติดต่อกับ Server ไม่ได้"

โดยการทำงานจะเริ่มที่ฝั่ง Client จะทำการร้องขอไฟล์ Count.txt จาก Server เสียก่อน โดยจะมาทำการเก็บที่ C:\TuiTonMax หลังจากที่ได้ไฟล์ Count.txt มาครบแล้วก็จะทำการอ่านค่าใน Count.txt มาเก็บในตัวแปร Lineoftext แล้วก็ทำการลบไฟล์ Count.txt ที่อยู่ใน C:\TuiTonMax เสีย ถ้า Lineoftext มีค่าเป็น 0 ก็会上 Dialog แสดงว่า "ไฟล์ยังไม่ถูกสร้าง" แต่ถ้าไม่ใช่ก็จะทำการกำหนดค่าใน Lineoftext ให้กับตัวแปร OldFileCount แล้วก็ทำการร้องขอไฟล์ตามชื่อไฟล์ที่อยู่ใน Lineoftext จนกว่าจะรับไฟล์จนครบจึงจะทำการเรียก Procedure TestSend ทำงานต่อ

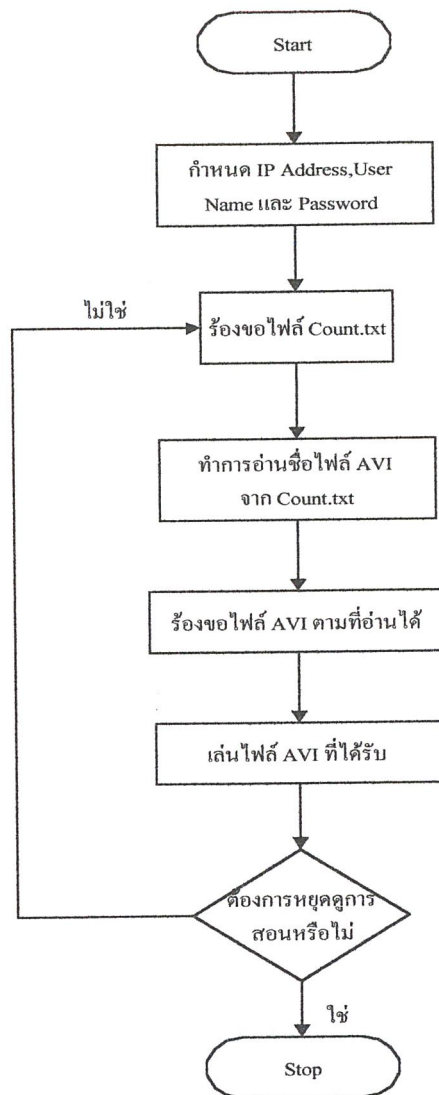
TestSend

จะทำการเล่นไฟล์ที่ได้รับมาและทำการเรียก Procedure GetNextFile ทำงานต่อไป

GetNextFile

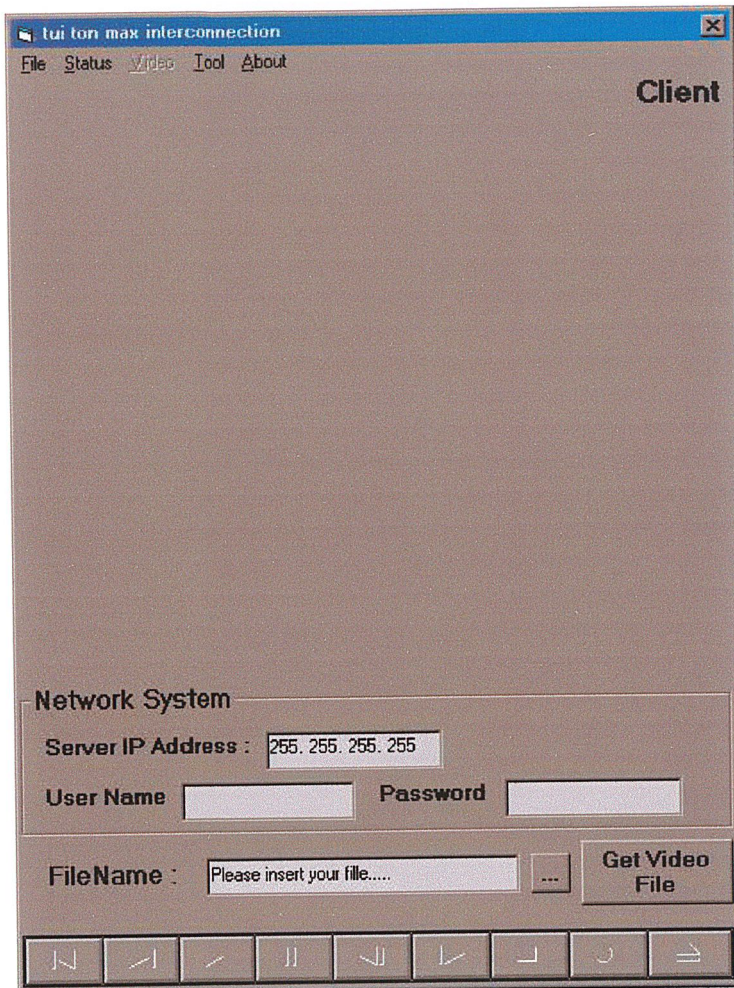
จะทำการร้องขอไฟล์ Count.txt จาก Server เสียก่อน โดยจะมาทำการเก็บที่ C:\TuiTonMax หลังจากที่ได้ไฟล์ Count.txt มาครบแล้วก็จะทำการอ่านค่าใน Count.txt มาเก็บในตัวแปร Lineoftext แล้วทำการตรวจสอบดูว่าตรงกับไฟล์ก่อนหน้านี้หรือไม่ ถ้าตรงกันจะทำการกำหนดค่า Interval ของ Timer4 เป็น 1500 หลังจากครบแล้วค่อยเรียก Procedure GetNextFile ทำงานต่อไป แต่ถ้าไม่ตรงกันก็จะทำการร้องขอไฟล์นั้นมา เมื่อได้มาไฟล์มาครบแล้วก็จะเรียก

TestSend แล้วทำงานวนอย่างนี้ไปเรื่อยๆ จนกว่าฝั่ง Client จะกดปุ่ม Stop หรือเกิดปัญหาที่ฝั่ง Server



รูปที่ 4.10 โฟลว์ชาร์ต ของ Client

เมื่อเรากดปุ่ม Client เราจะเข้ามาทำงานในหน้าจอของ Client ซึ่งเป็นส่วนของการส่ง Request ไปยังเครื่องที่ทำการ Capture การสอนให้เครื่องนั้นทำการส่งไฟล์ภาพการสอนมาให้ นอกจากนั้นยังสามารถที่จะเลือกเล่นไฟล์ภาพวิดีโออื่นได้ด้วย



รูปที่ 4.11 แสดงหน้าจอส่วนการทำงานของ Client

จากรูปที่ 4.11 จะมีส่วนควบคุมการทำงานของเครื่อง Client ประกอบด้วย

4.2.2.1 Frame System Network ใน Frame Network System ประกอบด้วย

4.2.2.1.1 Server IP Address เป็นส่วนที่ทำการใส่ค่า IP Address ของ Server ที่ต้องการติดต่อเพื่อส่งคำร้องขอให้เครื่อง Server ทำการส่งไฟล์ภาพการสอณาให้ ดังรูปที่ 4.12

4.2.2.1.2 User Name เป็นส่วนที่ให้ User ทำการกรอกชื่อที่มีการ Assign ไว้ ดังรูปที่ 4.12

4.2.2.1.3 Password เป็น Password ของ User ที่ทำการ Assign ไว้ที่เครื่อง server ดังรูปที่ 4.12

4.2.2.2 ช่องกรอก File Name

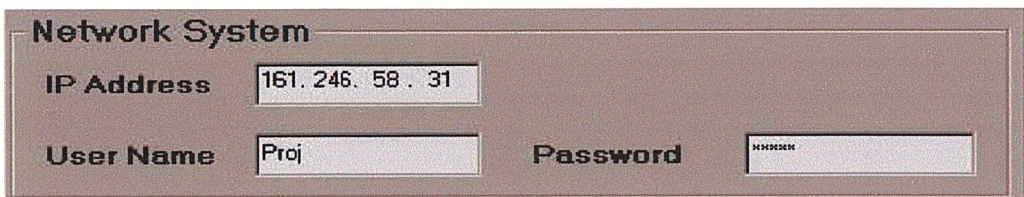
เป็นส่วนที่ผู้ใช้งานสามารถกรอกชื่อไฟล์ภาพ (avi ไฟล์) อื่นๆที่ต้องการจะเล่นนอกเหนือจากไฟล์ภาพการสอนได้ตามต้องการ ดังรูปที่ 4.13

4.2.2.3 ปุ่ม ...

เป็นปุ่มที่ใช้ในการ Browse หาชื่อไฟล์ภาพอื่นๆที่ต้องการเล่นนอกเหนือจากไฟล์ภาพการสอนในกรณีที่ไม่ทราบได้เรียกทอรี่ที่แน่นอน ดังรูปที่ 4.13

4.2.2.4 ปุ่ม Get Video File

จะเป็นตัวที่ส่งคำร้องขอเพื่อไปเอาไฟล์ภาพการสอนจาก Server ที่ได้ระบุไว้ใน Frame System Network ดังรูปที่ 4.13



Network System

IP Address: 161.246.58.31

User Name: Proj Password: XXXXXX

รูปที่ 4.12 Frame Network System



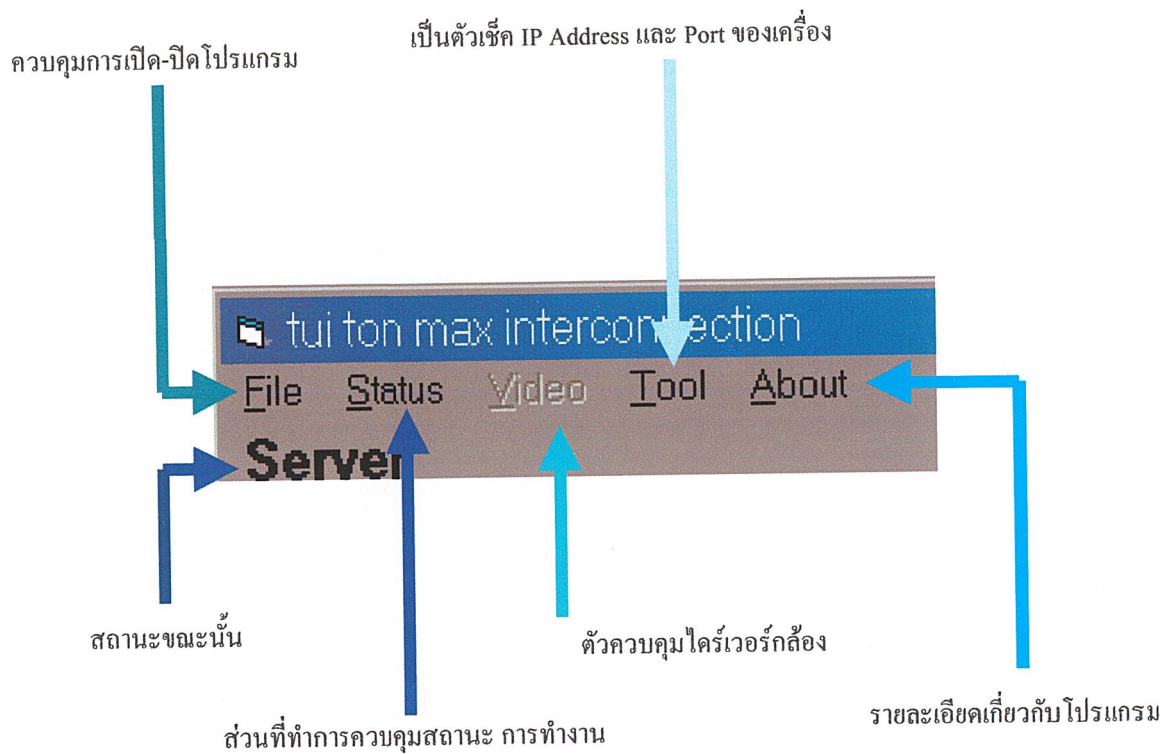
FileName: Please insert your file....

Get Video File

รูปที่ 4.13 ส่วนควบคุมการเล่นไฟล์

4.2.3 Menu Control

จากทั้งในส่วนของ Server กับ ส่วนที่เป็น Client นั้นมีส่วนที่เหมือนกันก็คือส่วนของ Menu bar ซึ่งจะเป็นส่วนในการควบคุมการทำงานของโปรแกรมอีกแบบหนึ่ง ดังมีรายละเอียดดังรูปที่ 4.14

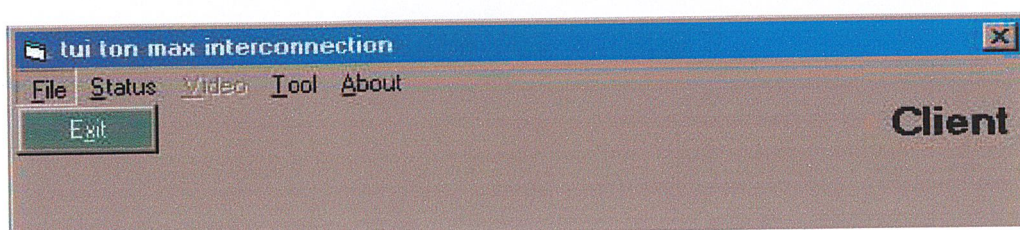


รูปที่ 4.14 แสดง Menu Bar

ในแต่ละ Menu ก็จะมีการทำงานที่แตกต่างกันไปตามเรื่องที่เกี่ยวข้องกับมัน

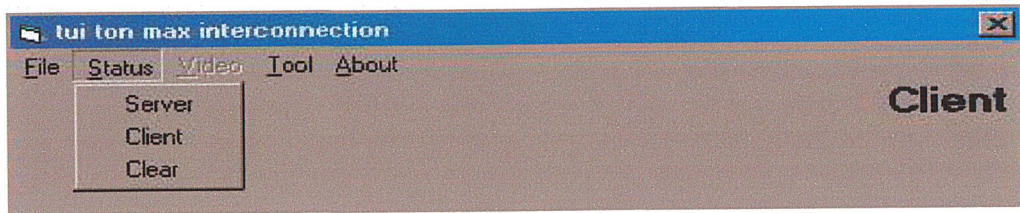
- 1) File เป็นส่วนควบคุมโปรแกรมมีการทำงานอย่างเดียวคือการออกจากโปรแกรม ดัง

รูปที่ 4.15



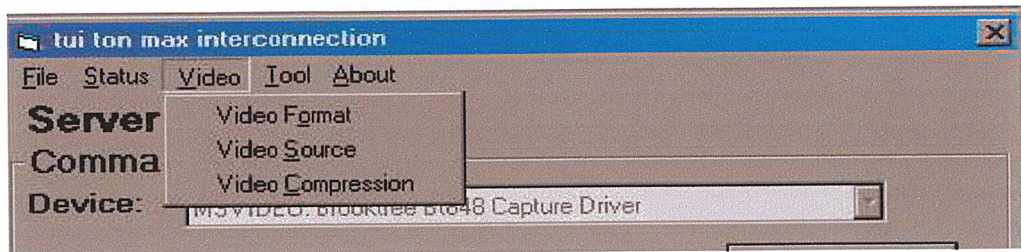
รูปที่ 4.15 File Menu

- 2) Status เป็นการเลือกสถานะว่าต้องการให้สถานะเป็น Server หรือต้องการให้เป็น Client หรือต้องการเริ่มต้นการทำงานของระบบใหม่ก็ให้เลือกที่ Clear ดังรูปที่ 4.16



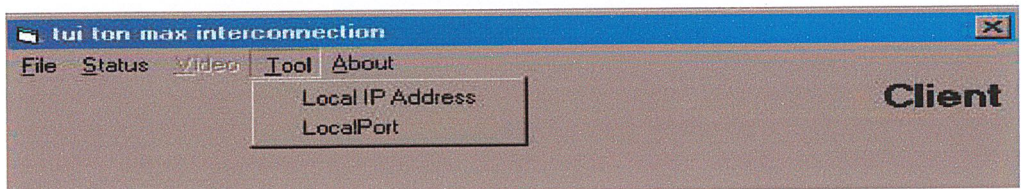
รูปที่ 4.16 Status Menu

3) Video ส่วนที่ทำการติดต่อกับตัวไดร์เวอร์ของกล้องวิดีโอดิจิทัลประกอบด้วย Video Source, Video Format, Video Compression อยู่ที่ฝั่ง Server เพียงอย่างเดียว ดังรูปที่ 4.17



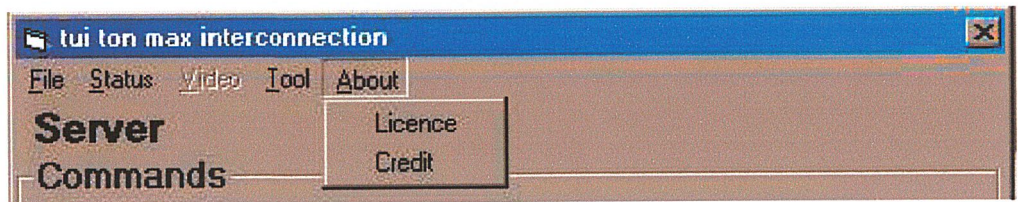
รูปที่ 4.17 Video Menu

4) Tool มีไว้เพื่อเป็นตัวบอกให้เครื่องรู้ว่าเรามี IP อะไร และ Port อะไร ดังรูปที่ 4.18



รูปที่ 4.18 Tool Menu

5) About เป็นส่วนที่จะแสดงชื่อคณะผู้จัดทำและเป็นส่วนที่แสดงลิขสิทธิ์ ดังรูปที่ 4.19



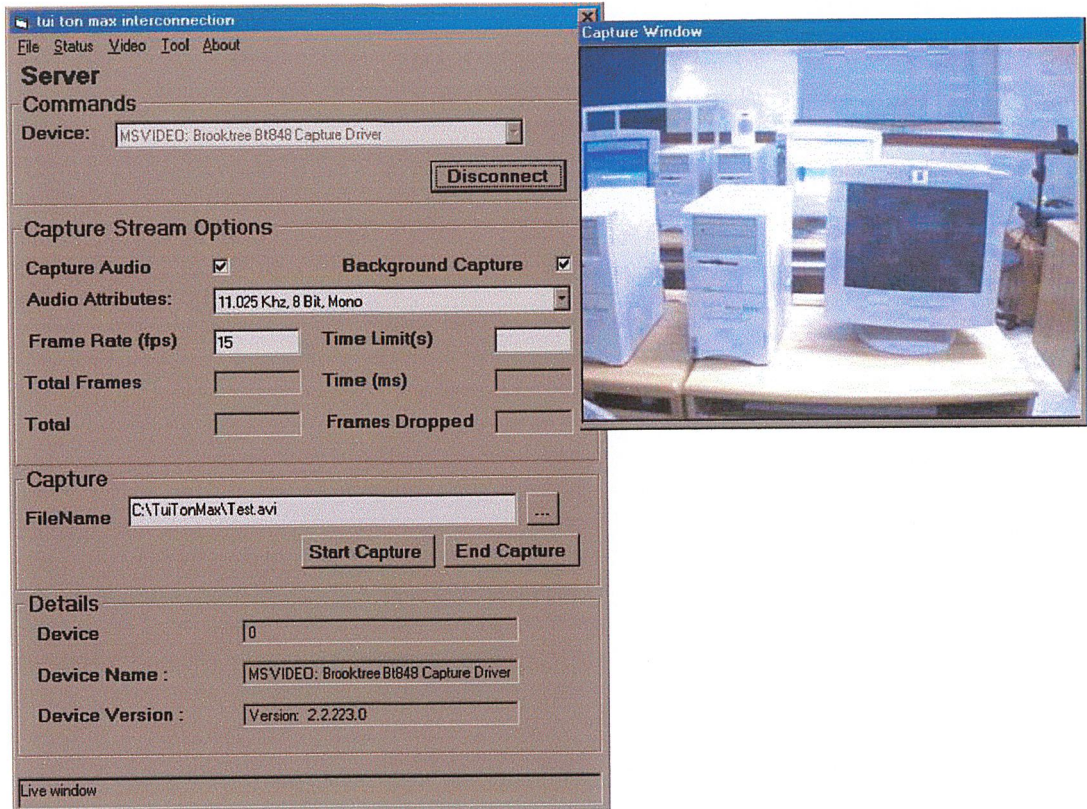
รูปที่ 4.19 About Menu

4.3 ขั้นตอนการใช้โปรแกรม

ในการใช้งาน โปรแกรมนั้นสามารถใช้งานได้แบ่งเป็น 2 ลักษณะคือ เป็น Server (เป็นส่วนที่ใช้ในการ Capture ภาพการสอน) และ เป็น Client (เป็นส่วนที่ใช้ในการรับไฟล์จากเครื่อง Server มาแสดงผล) ได้ดังนี้

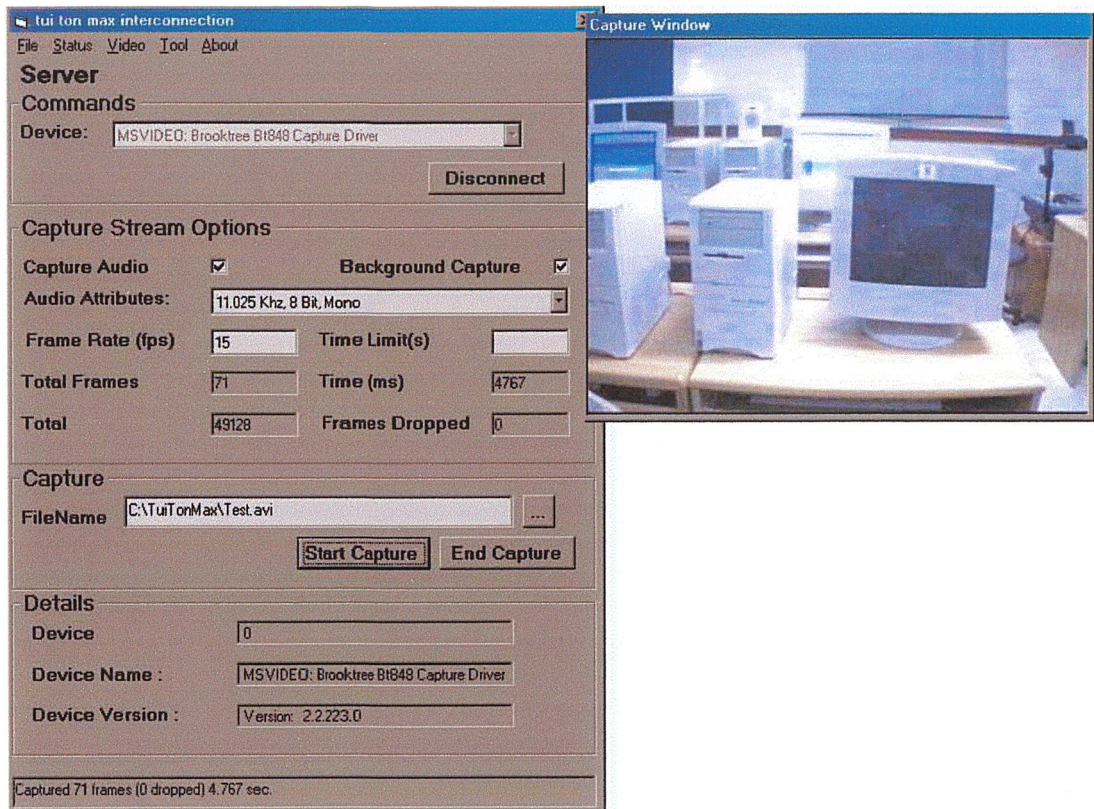
4.3.1 การทำงานของ Server

เริ่มจากปุ่ม Server ที่หน้าจอหลัก (รูปที่ 4.3) เพื่อเข้าสู่การทำงานของ Server ซึ่งจะปรากฏหน้าจอตั้งรูปที่ 4.4 หลังจากนั้นให้ทำการกดปุ่ม Connect เพื่อทำการติดต่อกับกล้องดิจิทัลก็จะได้หน้าจอปรากฏตั้งรูปที่ 4.20 ซึ่งจะมีหน้าต่างของภาพที่ Capture ได้จากกล้องปรากฏเพิ่มขึ้นมา และ ปุ่ม Connect จะกลายเป็น ปุ่ม Disconnect แทนเพื่อใช้ในการยกเลิกการติดต่อกับกล้องดิจิทัล ส่วนในช่องของ Status ก็จะปรากฏคำว่า Live Window ขึ้น และใน Frame Detail ก็จะมีปรากฏรายละเอียดของกล้องดิจิทัล



รูปที่ 4.20 หน้าจอหลังทำการติดต่อกับกล้องดิจิทัล

เมื่อต้องการที่จะเริ่ม Capture ก็ให้ทำการกดปุ่ม Start Capture ก็จะได้หน้าจอดังรูปที่ 4.21



รูปที่ 4.21 หน้าจอขณะทำการ Capture

จากรูปจะเห็นว่าเมื่อเรา Start Capture ที่แถบสถานะเราจะเห็นสถานะการ Capture เพื่อที่เราจะได้รู้ว่าตอนนี้ระบบทำงานไปถึงขนาดไหนแล้ว เราสามารถดูได้จากตรงส่วนนี้ และเมื่อเราต้องการที่จะเลิกการ Capture ก็อาศัยปุ่ม End Capture เมื่อ End Capture ที่แถบ Status นั้นจะแสดงผลว่าทั้งหมดเราสามารถที่จะ Capture อะไรได้บ้าง ลักษณะจะเป็นดังรูปที่ 4.22

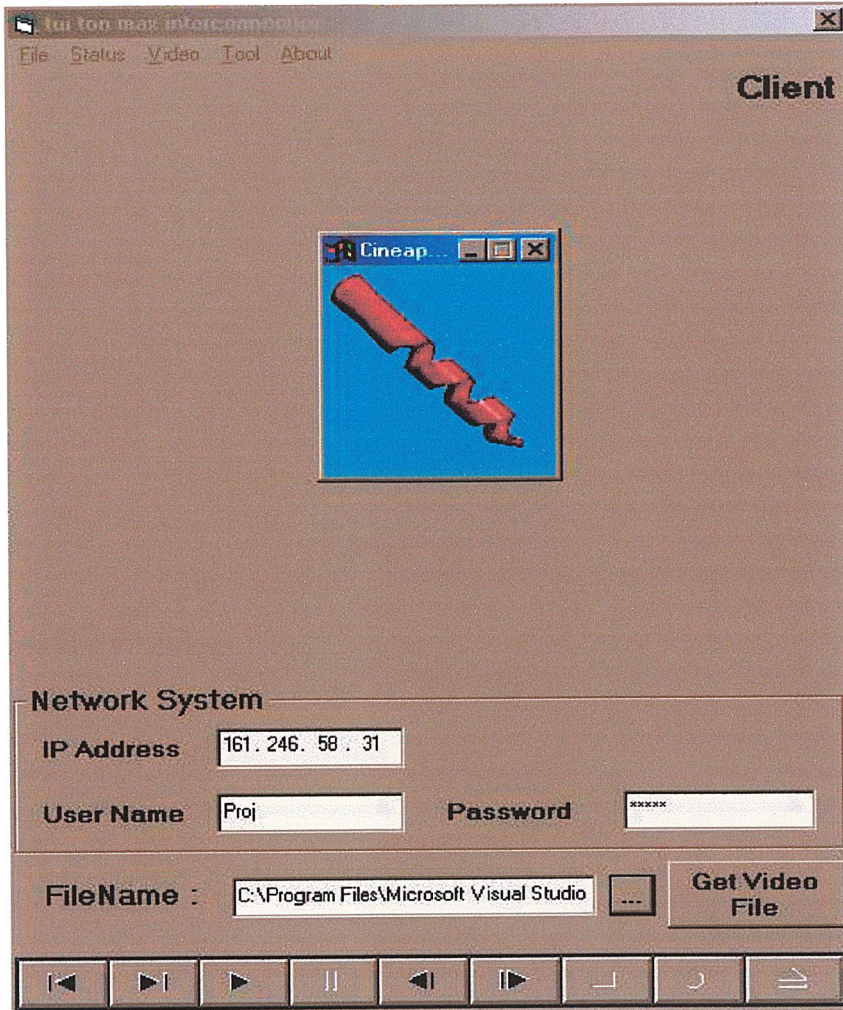
Captured 6.489 sec. 97 frames (0 dropped) (14.948 fps). 71552 audio bytes (11.025 sps)

รูปที่ 4.22 Status เมื่อ End

4.3.2 การทำงานของ Client

การทำงานในส่วน Client นั้นเมื่อเราต้องการจะรับไฟล์ภาพจาก Server เราเพียงกรอกข้อมูล IP Address ของ Server ที่เราต้องการติดต่อ และกรอกค่า User Name และ Password (User Name และ Password จะต้องได้รับการ Assign ไว้ที่เครื่อง Server ก่อน) แล้วทำการกดปุ่ม Get Video File เท่านั้น โปรแกรมก็จะทำการติดต่อกับเครื่อง Server ให้ทำการส่งไฟล์ภาพการสอนที่

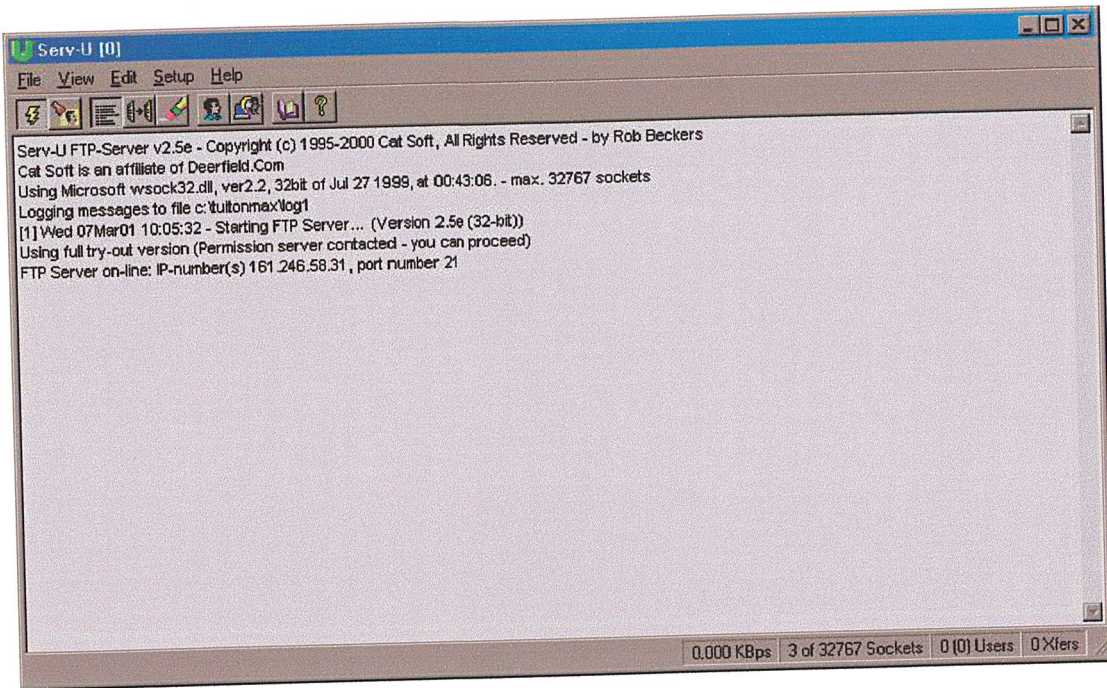
เครื่อง Server ทำการ Capture เสร็จแล้วมาให้ เมื่อเครื่อง Client ได้รับไฟล์ภาพมาครบถ้วนแล้วก็จะทำการเล่นไฟล์ภาพนั้นขึ้นมาบนหน้าจอ ดังรูปที่ 4.23



รูปที่ 4.23 เมื่อได้รับไฟล์มาก็จะทำการเล่น

4.4 ส่วนที่ทำการสนับสนุนโปรแกรม

โปรแกรมในปัญหาพิเศษนี้ได้มีการอาศัยโปรแกรมตัวหนึ่งเป็นตัวช่วยในการทำงานช่วยการจัดการส่งไฟล์จากเครื่อง Server ไปยังเครื่อง Client โดยที่โปรแกรมตัวนี้จะมีการเซตไว้ที่เครื่อง Server คือ โปรแกรม Server-U ซึ่งโปรแกรม Server-U ได้แสดงไว้ดังรูปที่ 4.24



รูปที่ 4.24 Server-U

4.5 ตัวอย่างผลการทดลองจากการทดสอบโปรแกรม

จากการทดสอบจะเห็นว่าการทำงานจะเข้าใกล้ที่จะเป็น Real - Time เนื่องจากการส่งข้อมูลมาแสดงผลที่ตัว Client นั้น Server ต้องทำการเก็บไฟล์ให้ได้ไฟล์แรกก่อนถึงจะสามารถที่จะให้ตัว Client สามารถเข้าไปร้องขอข้อมูลได้ฉะนั้นจากเหตุผลตรงที่ตัว Server ต้องทำการเก็บไฟล์ให้ได้ไฟล์หนึ่งก่อน ฉะนั้นต้องเสียเวลาในช่วงนั้นไปจากการทดลองจะกำหนดไว้ให้เป็น 5 วินาที ดังนั้นการทำงานของตัว Client จะช้ากว่าตัว Server อยู่ 5 วินาที (เราสามารถกำหนดเวลารอใหม่ได้แต่จากการทดลอง 5 วินาทีเป็นเวลาที่เหมาะสมที่สุด) จะเห็นได้จากภาพที่เราจับได้จากหน้าจอ Server กับ Client ในขณะที่ทำงานเวลาเดียวกัน

จากรูปจะเห็นได้ว่าทางฝั่งซ้ายมือเป็นหน้าจอทางฝั่ง Server และทางขวามือเป็นหน้าจอทางฝั่ง Client การทดลองครั้งนี้เป็นการทดลองโดยนับถอยหลังจาก 5 - 0 โดยอาศัยนับด้วยนิ้วมือจะเห็นได้ดังรูปที่ 4.25



ก. ทางฝั่งเซิร์ฟเวอร์

ข. ทางฝั่งไคลเอนต์

รูปที่ 4.25 แสดงการเปรียบเทียบหน้าจอ Server กับ Client ในขณะเวลาเดียวกัน

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

เนื่องจากการเรียนการสอนในปัจจุบันมีปัญหาและอุปสรรคมากมาย จึงเกิดแนวความคิดที่จะประยุกต์เอาเครื่องคอมพิวเตอร์และระบบเครือข่ายมาใช้ร่วมกัน โดยนำเอาความสามารถของทั้งสองสิ่งมาพัฒนาเป็นโปรแกรม เพื่อนำไปช่วยสนับสนุนการเรียนการสอนให้มีประสิทธิภาพดียิ่งขึ้น

5.1.1 ปัญหาต่างๆ ที่พบระหว่างพัฒนาโปรแกรม ดังต่อไปนี้

1) ในการพัฒนาโปรแกรม Component ที่สำคัญคือตัว CapturePRO ซึ่งเป็น Component ของทางบริษัท Pegasus จำกัด โดยที่ทางบริษัทมีตัว Demo Versions ให้บุคคลที่ต้องการทดลองใช้ Component ดังกล่าวดาวน์โหลดไปทดลองใช้ หากต้องการตัวเต็มต้องทำการซื้อผ่านทางบริษัท โดยที่ปัญหาพิเศษนี้ได้ใช้ Component ตัวนี้เป็นหลัก ดังนั้นความสามารถบางอย่างของตัว Component นี้จึงไม่สามารถใช้ได้อย่างเต็มประสิทธิภาพอาจเป็นเพราะทางบริษัทมีการจำกัดความสามารถของตัว Component นี้ไว้เพื่อให้ผู้ที่ต้องการซื้อติดต่อกับทางบริษัทเอง ดังนั้นการพัฒนาโปรแกรมจึงประสบปัญหา คือ ไม่สามารถใช้ event, method และ Functions ของตัว CapturePRO ได้อย่างเต็มที่ แต่เราก็สามารถที่จะดำเนินการพัฒนาโปรแกรมให้เกิดความถูกต้องและสมบูรณ์ที่สุดเท่าที่สามารถทำได้

2) เนื่องจากไฟล์ภาพวิดีโอที่เก็บมีขนาดใหญ่ ทำให้ต้องมีการวนมาบันทึกทับไฟล์เดิม เพื่อประหยัดเนื้อที่ในเครื่องที่ทำการ Capture

3) ไฟล์ภาพวิดีโอที่จะส่งมีขนาดใหญ่เนื่องจากไม่ได้ผ่านการบีบอัดข้อมูล เนื่องจากจุดประสงค์หลักของการพัฒนาโปรแกรมนั้นเพื่อให้เป็น Real Time ถ้าหากมีการบีบอัดข้อมูลจะทำให้เสียเวลามากกว่าเดิมทำให้ผิดจุดประสงค์หลักไป ดังนั้นจึงไม่มีการบีบอัดข้อมูลก่อนที่จะส่ง ทำให้เวลาที่ทาง Client มาดึงไฟล์ไปเล่นนั้นจะใช้เวลานาน

5.1.2 ข้อจำกัดของตัวโปรแกรม

ในการสร้างโปรแกรมทุกโปรแกรมจะมีข้อจำกัดของแต่ละโปรแกรม และในโปรแกรมนี้อีกเช่นเดียวกันมีข้อจำกัดต่าง ๆ อันเกิดจากการพัฒนาโปรแกรมเพื่อความถูกต้องของการทำงาน

1) ในการเริ่มต้นการทำงานได้ในแต่ละครั้ง ส่วนที่ทำงานเป็น Server ต้องมีการเปิดก่อน และต้องเปิด Server-U ด้วย และเครื่องที่ทำงานส่วน Client จะสามารถทำงานได้ก็ต่อเมื่อส่วนที่เป็น Server และ Server-U ทำงานก่อน

2) เนื่องจากเครื่องที่ใช้เป็นตัว Capture มีประสิทธิภาพไม่สูงนักจึงไม่สามารถรองรับการทำงานหลายอย่างได้พร้อมกัน ทั้งนี้เพราะเครื่องที่ใช้เป็นตัว Capture ต้องทำงานทั้งการ Capture ไฟล์ภาพการสอนและส่งไฟล์ภาพการสอนไปยังเครื่องที่ทำการร้องขอในเวลาเดียวกัน

3) ไม่สามารถส่งไฟล์ภาพวิดีโอได้ทัน เนื่องจากประสิทธิภาพของ Hardware จึงอาจทำให้การรับไฟล์ที่จะแสดงผลไม่เป็นไปตามลำดับ

5.2 ข้อเสนอแนะ

5.2.1 ต้องการระบบเน็ตเวิร์คที่สามารถถ่ายโอนข้อมูลได้เร็ว

5.2.2 ถ้าต้องการเพิ่มประสิทธิภาพในการทำงานของโปรแกรม ควรให้มีส่วนที่ทำการบีบอัดข้อมูลก่อน ซึ่งจะทำให้ไฟล์ภาพวิดีโอมีขนาดเล็กลง ดังนั้นเราอาจจะต้องเพิ่มประสิทธิภาพเครื่องให้มีความสามารถมากขึ้น

5.2.3 ควรใช้ตัว Component CapturePRO ที่เป็นตัวที่สั่งซื้อแล้ว แทนที่จะเป็นตัว Demo Version เพราะจะได้ทั้งวิธีการใช้รวมทั้งรายละเอียดเกี่ยวกับโปรแกรม ทำให้สามารถใช้ตัว Component นี้ได้อย่างมีประสิทธิภาพ

บรรณานุกรม

- [1] นิสิตรา สืบประสงค์ และวิไลรัตน์ พิธรากร. 2541. “การสื่อสารมัลติมีเดียผ่านเครือข่ายอินเทอร์เน็ต”. ปรินญาณิพนธ์ ปรินญาณวิศกรรมบัณฑิต สาขาโทรคมนาคม,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- [2] Ed Tittel and James Michael Stewart. 1997. **Intranet Bible**. IDG Books Worldwide, Inc
- [3] ชาริน สิทธิธรรมขารี. 2543. “คู่มือการเขียนโปรแกรม Microsoft Visual Basic 6.0”.
กรุงเทพฯ : ซัคเซสมิเดีย
- [4] กิตติ ภัคดีวัฒนะกุล และจำลองครูอุสาหะ. 2543. “Microsoft Visual Basic 6.0 ฉบับโปรแกรมเมอร์”. พิมพ์ครั้งที่ 7. กรุงเทพฯ : คอมแอนด์คอนซัลท์
- [5] Pegasus Software, LLC. www.pegasustools.com