

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย

THE APPLICATION PROGRAMME OF MONITORING  
COMPUTERS OVER THE NETWORK



เลขหมู่.....  
เลขทะเบียน..... 73329  
วัน,เดือน,ปี..... 12 ก.ค. 2550

b. 11790295  
i. ....

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต  
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE APPLICATION PROGRAMME OF MONITORING  
COMPUTERS OVER THE NETWORK



A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE  
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE  
FACULTY OF SCIENCE  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADEMIC YEAR 2006

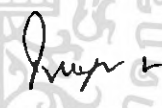
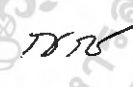
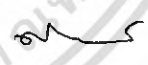
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หัวข้อปัญหาพิเศษ** โปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย  
 THE APPLICATION PROGRAMME OF MONITORING COMPUTERS  
 OVER THE NETWORK

**ชื่อนักศึกษา** นายธินะ รุ่งสว่าง 46050294  
 นางสาวรสรินทร์ รัชตะชัยอนันต์ 46050298

**ภาควิชา** คณิตศาสตร์และวิทยาการคอมพิวเตอร์  
**สาขาวิชา** วิทยาการคอมพิวเตอร์  
**อาจารย์ที่ปรึกษา** อ.ศังกรศรัณย์ ล่องชุมผล

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2549

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	รศ.ไพโรบลย์ พันธวิทย์พงษ์	
กรรมการ	ผศ.ดร.กรกช ประทุมรักษ์	
กรรมการและอาจารย์ที่ปรึกษา	อ.ศังกรศรัณย์ ล่องชุมผล	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	โปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย	
ชื่อนักศึกษา	นายธินณะ รุ่งสว่าง	46050294
	นางสาวรัชนีทร์ รัชตะชัยอนันต์	46050298
ปริญญา	วิทยาศาสตร์บัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2549	
อาจารย์ที่ปรึกษา	อ.ศังกรศรีณีย์ ล่องชุมผล	

### บทคัดย่อ

ปัญหาพิเศษนี้ เป็นการสร้างโปรแกรมที่เครื่องแม่ข่ายหรือเครื่องคอมพิวเตอร์ของอาจารย์ผู้สอน ใช้ตรวจสอบการแสดงผลบนเครื่องลูกข่ายหรือเครื่องคอมพิวเตอร์ของนักเรียน ในการเรียนการสอนภายในห้องเรียนคอมพิวเตอร์ เพื่อความสะดวกสบายในการตรวจสอบและการจัดการกับเครื่องคอมพิวเตอร์ลูกข่ายภายในเครือข่ายของห้องเรียน เช่น การตรวจสอบข้อมูลของเครื่องลูกข่าย, การดูภาพหน้าจอของเครื่องลูกข่าย ซึ่งสามารถทำได้ทีละหลายเครื่อง และการแพร์ภาพหน้าจอของเครื่องแม่ข่าย ปัญหาพิเศษนี้ยังช่วยป้องกันการใช้งานต่างๆ เช่น การลือกเมาส์และแผงแป้นอักขระ ซึ่งเป็นประโยชน์ต่อการเรียนการสอน ที่ทำให้นักเรียนสนใจการสอนจากอาจารย์ได้มากขึ้น นอกจากนี้ยังสามารถส่งข้อความเตือน ไปยังเครื่องลูกข่าย และสามารถส่งเพิ่มข้อมูลหากันภายในระบบเครือข่ายได้อีกด้วย

ปัญหาพิเศษนี้ใช้ภาษา C#.NET ในการพัฒนาโปรแกรม ซึ่งใช้ Microsoft Visual Studio 2005 เป็นเครื่องมือในการพัฒนาโปรแกรม และใช้โพรโทคอล TCP/IP ในการติดต่อระหว่างเครื่องแม่ข่ายและเครื่องลูกข่ายภายในระบบเครือข่าย นอกจากนี้ในการพัฒนาโปรแกรมจะใช้ซ็อกเก็ตเป็นตัวกลางในการสื่อสารข้อมูลระหว่างโปรแกรมและฮาร์ดแวร์ และใช้การติดต่อสื่อสารแบบไม่ประสานเวลา

Special Project Title	THE APPLICATION PROGRAMME OF MONITORING COMPUTERS OVER THE NETWORK	
Students	Mr. Tisana Roongsawang	46050294
	Miss Russarin Ruchatachayanon	46050298
Degree	Bachelor of Science	
Department	Mathematics and Computer Science, Faculty of Science	
Programme	Computer Science	
Academic Year	2006	
Special Project Advisor	Sungkornsarun Longchupole	

### ABSTRACT

This special project is a program development for server machine monitoring of client machines in classrooms. It can handle more than one machine for convenience to monitor and to manage client machines within a network system such as information checking and monitoring screen of client machines including broadcasting screen of the server machine. Moreover, it also supports to use any various shielded functions, for example, locking mouse and keyboard which is useful for studying and increasing attention of students. Furthermore, it has features of sending alert messages to client machines for server site and transferring some files for both sites in the network system.

This special problem uses TCP/IP for connecting server to clients in network and using socket for communication between program and hardware with asynchronous. The complement of C#.NET language with Microsoft Visual Studio 2005 in which is a tool for developing this program.

## กิตติกรรมประกาศ

ในการทำปัญหาพิเศษ เรื่องโปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย สามารถสำเร็จลุล่วงได้ด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ อาจารย์ดังกกรศรีธัญย์ ล่องชุมผล อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาดังกล่าว รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จได้ด้วยดี รวมทั้งเพื่อนๆ และน้องๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ  
มีนาคม 2549



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	.I
บทคัดย่อภาษาอังกฤษ .....	.II
กิตติกรรมประกาศ .....	.III
สารบัญ .....	.IV
สารบัญตาราง .....	VIII
สารบัญรูป .....	.IX
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา .....	2
1.3 สมมติฐานของการศึกษา .....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการศึกษา .....	2
1.5 ขอบเขตของการศึกษา .....	3
1.6 ขั้นตอนของการศึกษา .....	3
1.7 ข้อจำกัดของการศึกษา .....	4
บทที่ 2 การจัดการระบบเครือข่ายและทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ทีซีพี/ไอพี (TCP/IP หรือ Transmission Control Protocol/Internet Protocol).....	5
2.1.1 Encapsulation / Demultiplexing.....	5
2.1.1.1 ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer) .....	7
2.1.1.2 ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer) .....	7
2.1.1.3 ชั้นสื่อสารนำส่งข้อมูล (Transport Layer) .....	10
2.1.1.4 ชั้นสื่อสารการประยุกต์ (Application Layer) .....	17
2.2 การเขียนโปรแกรมซ็อกเก็ต (Socket Programming).....	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
2.2.1 Socket Programming with TCP.....	19
2.2.2 Socket Programming with UDP.....	19
2.2.3 Windows Socket.....	20
2.2.3.1 ภาพสถาปัตยกรรมของ Winsock.....	21
2.2.3.2 ความหมายของ Winsock.....	21
2.2.3.3 ระดับของ Winsock ในส่วนของ TCP/IP.....	22
2.3 Microsoft Visual C# .NET.....	22
2.3.1 Namespaces.....	23
2.4 .NET Framework.....	23
2.4.1 ความสำคัญ.....	24
2.4.2 เป้าหมายของการออกแบบและคุณสมบัติที่สำคัญ.....	24
2.4.2.1 Interoperability.....	24
2.4.2.2 Common Runtime Engine.....	24
2.4.2.3 Language Independence.....	25
2.4.2.4 Base Class Library (BCL).....	25
2.4.2.5 Simplified Deployment.....	25
2.4.2.6 Security.....	25
2.4.3 สถาปัตยกรรม .NET Framework.....	26
2.4.3.1 Common Language Infrastructure (CLI).....	26
2.4.3.2 Assemblies.....	27
2.4.3.3 Metadata.....	27
2.4.4 ความปลอดภัย.....	27
2.5 Windows Management Instrumentation (WMI).....	28
2.6 Window API.....	29

## สารบัญ(ต่อ)

	หน้า
2.6.1 ประเภทของ Win API.....	29
2.6.1.1 Base Services.....	29
2.6.1.2 Graphics Device Interface.....	29
2.6.1.3 User Inter.....	29
2.6.1.4 Common Dialog Box Library.....	30
2.6.1.5 Common Control Library.....	30
2.6.1.6 Windows Shell.....	30
2.6.1.7 Network Services.....	30
<b>บทที่ 3 การออกแบบโปรแกรม.....</b>	<b>32</b>
3.1 การออกแบบโปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย.....	32
3.2 Use Case Diagram.....	33
3.3 Class Diagram .....	34
3.3.1 Class T_Main.....	35
3.3.2 Class ClientInfo.....	37
3.3.3 Class T_BC.....	38
3.3.4 Class S_Main.....	39
3.3.5 Class SocketPacket.....	41
3.3.6 Class WindowAPI.....	41
3.4 Deployment Diagram .....	42
<b>บทที่ 4 การทำงานโปรแกรม.....</b>	<b>43</b>
4.1 โปรแกรมในเครื่องด้านแม่ข่าย.....	43
4.1.1 การดูข้อมูลเครื่องลูกข่าย.....	45
4.1.2 การเรียกดูภาพหน้าจอของเครื่องลูกข่าย.....	45

## สารบัญ(ต่อ)

	หน้า
4.1.3 การส่งภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่าย.....	.47
4.1.4 การล็อกเมาส์และแผงแป้นอักขระเครื่องลูกข่าย.....	.48
4.1.5 การส่งข้อความไปยังเครื่องลูกข่าย.....	.50
4.1.6 การรับ-ส่งเพิ่มข้อมูลไปยังเครื่องลูกข่าย.....	.51
4.2 โปรแกรมในเครื่องด้านลูกข่าย.....	.53
4.2.1 การรับ-ส่งเพิ่มข้อมูลไปยังเครื่องแม่ข่าย.....	.56
<b>บทที่ 5 สรุปผลการทำงานของโปรแกรมและข้อเสนอแนะ.....</b>	<b>.57</b>
5.1 สรุปผลการทำงานของโปรแกรม.....	.57
5.2 ข้อจำกัดของโปรแกรม.....	.58
5.3 ข้อเสนอแนะ.....	.59
5.3.1 การปรับปรุง.....	.59
5.3.1.1 ส่วนของการดูหน้าจอเครื่องลูกข่าย.....	.59
5.3.1.2 ส่วนของแพร์ภาพหน้าจอเครื่องแม่ข่าย.....	.59
5.3.1.3 ส่วนของการล็อกเมาส์และแผงแป้นอักขระ.....	.59
5.3.1.4 ส่วนของการส่งเพิ่มข้อมูล.....	.59
5.3.2 การต่อยอด.....	.59
5.3.2.1 ส่วนของการสื่อสาร.....	.59
5.3.2.2 ส่วนของการควบคุม.....	.59
<b>ภาคผนวก ก.....</b>	<b>.60</b>
<b>บรรณานุกรม.....</b>	<b>.78</b>

## สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงประเภทของ Flag .....	13
3.1 แสดงความหมายเมทอดของ Class T_Main.....	36
3.2 แสดงความหมายเมทอดของ Class ClientInfo.....	37
3.3 แสดงความหมายเมทอดของ Class T_BC.....	38
3.4 แสดงความหมายเมทอดของ Class S_Main.....	40
3.5 แสดงความหมายเมทอดของ Class WindowAPI.....	41



## สารบัญรูป

รูปที่	หน้า
2.1 ขั้นตอนการ Encapsulation และ Demultiplexing .....	6
2.2 โครงสร้าง ทีซีพี/ไอพี.....	7
2.3 ส่วนหัวของไอพี.....	8
2.4 ส่วนหัวของICMP.....	10
2.5 ส่วนหัวของยูดีพี.....	11
2.6 ส่วนหัวของทีซีพี.....	12
2.7 ลำดับขั้นตอนการส่งทีซีพี.....	14
2.8 ลำดับขั้นตอนที่โหนดสองแห่งพยายามสร้างการเชื่อมต่อระหว่างซ็อกเก็ตคู่เดียวกัน.....	14
2.9 เหตุการณ์ที่กลุ่มข้อมูล TPDU เป็นสำเนากลุ่มข้อมูลเก่า.....	17
2.10 โครงสร้างของ window socket.....	20
2.11 การสื่อสารผ่านซ็อกเก็ต.....	21
2.12 ระดับของซ็อกเก็ต.....	22
2.13 แสดงการทำงานของ CLI.....	26
2.14 แสดงโครงสร้างการทำงานของคลาส WMI.....	28
2.15 โครงสร้างของ Win32 API.....	31
3.1 แสดงการทำงานของระหว่างเครื่องลูกข่ายและแม่ข่าย.....	32
3.2 แสดงแผนภาพ use case diagram .....	33
3.3 แสดงแผนภาพความสัมพันธ์ของคลาสทั้งหมด.....	34
3.4 แสดงลักษณะประจำและเมทอดของ Class T_Main.....	35
3.5 แสดงลักษณะประจำและเมทอดของ Class ClientInfo.....	37
3.6 แสดงลักษณะประจำและเมทอดของ Class T_BC.....	38
3.7 แสดงลักษณะประจำและเมทอดของ Class S_Main.....	39
3.8 แสดงลักษณะประจำและเมทอดของ Class SocketPacket.....	41
3.9 แสดงลักษณะประจำและเมทอดของ Class WindowAPI.....	41

## สารบัญรูป(ต่อ)

รูปที่	หน้า
3.10 แสดงแผนภาพ Deployment diagram .....	42
4.1 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายขณะรอรับการติดต่อจากเครื่องลูกข่าย.....	43
4.2 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายเมื่อมีเครื่องลูกข่ายติดต่อเข้ามาแสดงใน tab INFO VIEW.....	44
4.3 หน้าต่างที่จะแสดงเมื่อเครื่องลูกข่ายหลุดจากการติดต่อ.....	44
4.4 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab THUMBNAIL VIEW.....	46
4.5 หน้าต่างที่จะแสดงเมื่อไม่มีการเลือกเครื่องลูกข่ายในการจัดการ.....	46
4.6 หน้าต่าง Thumbnail View.....	47
4.7 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab BROADCAST.....	48
4.8 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab LOCK MOUSE KEYBOARD.....	49
4.9 หน้าต่างเมื่อใส่ค่า Time Lock เกิน 8 ชั่วโมง.....	49
4.10 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab SEND MESSAGE.....	50
4.11 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab SEND FILE.....	52
4.12 หน้าต่าง OpenFileDialog.....	52
4.13 หน้าต่างเมื่อได้รับเพิ่มข้อมูลเรียบร้อยแล้ว.....	53
4.14 หน้าต่างโปรแกรมบนเครื่องลูกข่าย.....	53
4.15 หน้าต่างเมื่อใส่เลขที่อยู่ไอพีของเครื่องแม่ข่ายผิด.....	54
4.16 หน้าต่างเมื่อกดปุ่ม DISCONNECT.....	54
4.17 หน้าต่างเมื่อหลุดการติดต่อจากเครื่องแม่ข่าย.....	54
4.18 Message Box แสดงถึงผู้สอนเริ่มทำการแพรรูปหน้าจอแล้ว.....	55
4.19 หน้าต่างที่แสดงหน้าจอของเครื่องแม่ข่าย.....	55
4.20 หน้าต่างแสดงข้อความจากเครื่องแม่ข่ายชนิด Information .....	55
4.21 หน้าต่างแสดงข้อความจากเครื่องแม่ข่ายชนิด Warning.....	56
4.22 หน้าต่างแสดงข้อความจากเครื่องแม่ข่ายชนิด Question.....	56
4.23 หน้าต่างแสดงข้อความจากเครื่องแม่ข่ายชนิด Error.....	56
ก.1 แสดงหน้าต่าง download Microsoft .NET Framework 2.0.....	61

## สารบัญรูป(ต่อ)

รูปที่	หน้า
ก.2 แสดงขั้นตอนเริ่มกระบวนการติดตั้งของ Microsoft .NET Framework 2.0.....	62
ก.3 แสดงขั้นตอนพร้อมเริ่มการติดตั้งของ Microsoft .NET Framework 2.0.....	62
ก.4 แสดงหน้ายอมรับเงื่อนไข ข้อตกลงการใช้งาน Microsoft .NET Framework 2.0.....	63
ก.5 แสดงขั้นตอนดำเนินการติดตั้งของ Microsoft .NET Framework 2.0.....	63
ก.6 แสดงขั้นตอนเสร็จสิ้นการติดตั้งของ Microsoft .NET Framework 2.0.....	64
ก.7 แสดงขั้นตอนเริ่มกระบวนการติดตั้งโปรแกรมบนฝั่งแม่ข่าย.....	65
ก.8 แสดงขั้นตอนการตั้งค่าแฟ้มที่ติดตั้งโปรแกรมบนฝั่งแม่ข่าย.....	65
ก.9 แสดงขั้นตอนการตั้งค่า start menu ของโปรแกรมบนฝั่งแม่ข่าย.....	66
ก.10 แสดงขั้นตอนการเลือกติดตั้งสัณฐานปลัดบนเดสก์ทอปของโปรแกรมบนฝั่งแม่ข่าย.....	66
ก.11 แสดงข้อมูลการติดตั้งที่ได้ถูกกำหนดไปก่อนหน้าของโปรแกรมบนฝั่งแม่ข่าย.....	67
ก.12 แสดงสถานการณ์คัดลอกแฟ้มข้อมูลลงสู่เครื่องของโปรแกรมบนฝั่งแม่ข่าย.....	67
ก.13 แสดงขั้นตอนเสร็จสิ้นการลงโปรแกรมบนฝั่งแม่ข่าย.....	68
ก.14 แสดงขั้นตอนเริ่มกระบวนการติดตั้งโปรแกรมบนฝั่งลูกข่าย.....	69
ก.15 แสดงขั้นตอนการตั้งค่าแฟ้มที่ติดตั้งโปรแกรมบนฝั่งลูกข่าย.....	69
ก.16 แสดงขั้นตอนการตั้งค่า start menu ของโปรแกรมบนฝั่งลูกข่าย.....	70
ก.17 แสดงขั้นตอนการเลือกติดตั้งสัณฐานปลัดบนเดสก์ทอปของโปรแกรมบนฝั่งลูกข่าย.....	70
ก.18 แสดงข้อมูลการติดตั้งที่ได้ถูกกำหนดไปก่อนหน้าของโปรแกรมบนฝั่งลูกข่าย.....	71
ก.19 แสดงสถานการณ์คัดลอกแฟ้มข้อมูลลงสู่เครื่องของโปรแกรมบนฝั่งลูกข่าย.....	71
ก.20 แสดงขั้นตอนเสร็จสิ้นการลงโปรแกรมบนฝั่งลูกข่าย.....	72
ก.21 แสดงการถอนการติดตั้งจาก Start Menu ของโปรแกรมบนฝั่งแม่ข่าย.....	73
ก.22 แสดงการถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่ายจาก Add or Remove Programs.....	73
ก.23 แสดงการยืนยันว่าต้องการจะถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่ายจริงหรือไม่.....	74
ก.24 แสดงสถานะการถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่าย.....	74
ก.25 แสดงการยืนยันการถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่ายเสร็จสิ้น.....	75
ก.26 แสดงการถอนการติดตั้งโปรแกรมบนฝั่งลูกข่ายจาก Start Menu.....	75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

รูปที่	หน้า
ก.27 แสดงการถอนการติดตั้งโปรแกรมบนฝั่งลูกข่ายจาก Add or Remove Programs.....	76
ก.28 แสดงการยืนยันว่าต้องการจะถอนการติดตั้งโปรแกรมบนฝั่งลูกข่ายจริงหรือไม่.....	76
ก.29 แสดงสถานะการถอนการติดตั้งโปรแกรมบนฝั่งลูกข่าย.....	77
ก.30 แสดงการยืนยันการถอนการติดตั้งโปรแกรมบนฝั่งลูกข่ายเสร็จสิ้น.....	77



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา (Statement and significance of the problems)

ในปัจจุบันนี้ สถานศึกษาต่างๆ มากมายได้จัดให้มีการเรียนการสอนในห้องเรียนคอมพิวเตอร์ ซึ่งเป็นทางเลือกที่ดีที่สุดทางหนึ่งที่ได้เข้ามาแทนการสอนแบบเขียนบนกระดานหรือการฉายผ่านโปรเจคเตอร์ อีกทั้งสถานศึกษาโดยส่วนใหญ่จะจัดให้คอมพิวเตอร์ทุกเครื่องถูกเชื่อมต่อสัญญาณการสื่อสารเข้าด้วยกัน โดยผ่านเครือข่ายท้องถิ่น(LAN: Local Area Network) เพื่อความสะดวกในการติดต่อสื่อสาร การร่วมใช้แฟ้มข้อมูลหรือแชร์ไฟล์ การร่วมใช้อินเทอร์เน็ตหรือแชร์อินเทอร์เน็ต และง่ายต่อการจัดการต่างๆ

การเรียนการสอนในห้องเรียนคอมพิวเตอร์นั้น ผู้สอนจะสอนผ่านเครื่องคอมพิวเตอร์แม่ข่าย (Server) และนักเรียนจะเรียนรู้จากสิ่งที่ผู้สอนได้ปฏิบัติ จากการสังเกตภายในห้องเรียนคอมพิวเตอร์ จะเห็นได้ว่านักเรียนบางคนไม่ได้รับความรู้ได้อย่างเต็มที่ อาจมีสาเหตุหลายประการ เช่น นักเรียนกำลังแอบเล่นเกม หรือกำลังเล่นอินเทอร์เน็ต ซึ่งเป็นกิจกรรมที่อยู่นอกเหนือจากขอบเขตการเรียน จึงจำเป็นที่จะต้องมีการควบคุมเครื่องคอมพิวเตอร์ที่นักเรียนใช้อยู่ เพื่อให้การเรียนการสอนนั้นประสบความสำเร็จ

จากปัญหาที่ได้กล่าวมา ก่อให้เกิดหลักความคิดในการแก้ปัญหาเหล่านี้ โดยการสร้างโปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่ายขึ้นมา เพื่อช่วยเหลือให้ผู้สอนหรือจะเรียกได้ว่าเป็นผู้ดูแลระบบเครือข่ายนั้นมีความสะดวกในการตรวจสอบ และควบคุมเครื่องคอมพิวเตอร์ของนักเรียนแต่ละคนได้โดยผ่านเครื่องคอมพิวเตอร์แม่ข่ายหรือเครื่องของผู้สอนนั่นเอง อย่างเช่น การแพร่ภาพ(Broadcast) หน้าจอของเครื่องผู้สอนให้กับเครื่องของนักเรียน สามารถติดตามการทำงานของนักเรียนได้โดยเรียกดูหน้าจอเครื่องของนักเรียน ซึ่งผู้สอนสามารถให้ความช่วยเหลือกับนักเรียนได้ทันที อีกทั้งยังควบคุมการแจกหรือรวบรวมแฟ้มข้อมูลเอกสารงานจากนักเรียนได้โดยง่าย เป็นต้น

## 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา (Goal and objective)

จุดมุ่งหมายของการศึกษานี้ เพื่อนำเสนอโปรแกรมต้นแบบของโปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่ายท้องถิ่นที่ใช้ในห้องเรียนคอมพิวเตอร์ เพื่อความสะดวกสบายของผู้สอนที่สามารถตรวจสอบและควบคุมการทำงานของเครื่องลูกข่าย(Client) หรือเครื่องคอมพิวเตอร์ที่นักเรียนกำลังใช้ในการเรียนผ่านเครื่องคอมพิวเตอร์ของผู้สอน และเพื่อให้การเรียนการสอนผ่านเครือข่ายนั้นมีความสมบูรณ์และประสบความสำเร็จไปด้วยดี

## 1.3 สมมติฐานของการศึกษา (Hypothesis to be tested)

หากนำโปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่ายนี้ มาใช้ในห้องเรียนคอมพิวเตอร์แล้ว จะสามารถช่วยในเรื่องของการเพิ่มประสิทธิภาพของการเรียนการสอน หรือประสิทธิภาพในเรื่องของการเรียนการสอนอาจไม่เพิ่มขึ้น เนื่องจากผู้สอนอาจไม่ได้ใช้ประโยชน์จากโปรแกรมนี้ในการตรวจสอบอย่างเต็มที่ หรือนักเรียนไม่ให้ความร่วมมือหลังจากที่มีการดักเตือน

## 1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการศึกษา (Theory or concept of the study)

แนวความคิด คือ การสร้างโปรแกรมตรวจสอบและจัดการเครือข่ายแบบรวมศูนย์ในห้องเรียน โดยเครื่องของผู้สอนหรือผู้บรรยายจะมีสิทธิในการตรวจสอบ และควบคุมเครื่องลูกข่ายเท่านั้น เนื่องจากในปัญหาพิเศษนี้ได้มุ่งประเด็นไปที่การพัฒนาโปรแกรมสำหรับทำงานบนระบบปฏิบัติการ Microsoft Windows XP จึงทำการเลือกใช้ .NET Framework มาใช้ในการพัฒนาโปรแกรมเนื่องจาก .NET Framework มีคุณสมบัติในการติดต่อและทำงานร่วมกับระบบปฏิบัติการของทาง Microsoft ได้เป็นอย่างดี ส่วนในด้านการติดต่อสื่อสารในเครือข่ายจะใช้โพรโทคอล TCP/IP ในการติดต่อสื่อสาร เนื่องจากโพรโทคอลนี้สามารถทำงานใน OSI Model ซึ่งเป็นตัวแบบมาตรฐานของการติดต่อสื่อสารผ่านเครือข่ายได้อย่างมีประสิทธิภาพสูง สามารถตรวจสอบได้ว่าการส่งข้อมูลไปถึงเป้าหมายหรือไม่ นอกจากนี้ในการเขียนโปรแกรมจะใช้ซ็อกเก็ต(socket) เป็นตัวกลางในการสื่อสารข้อมูลระหว่างโปรแกรมและฮาร์ดแวร์(hardware) และใช้การติดต่อสื่อสารแบบไม่ประสานเวลา (asynchronous)

## 1.5 ขอบเขตของการศึกษา (Scope or limitation of the study)

ปัญหาพิเศษนี้มีขอบเขตของการศึกษา ดังนี้

1. โปรแกรมจะประกอบไปด้วย 2 ส่วน คือ โปรแกรมแม่ข่าย จะถูกติดตั้งในเครื่องคอมพิวเตอร์ของผู้สอนหรือผู้บรรยาย และโปรแกรมลูกข่าย จะถูกติดตั้งในเครื่องคอมพิวเตอร์ที่ผู้เรียนใช้
2. ความสามารถในการตรวจสอบเครื่องคอมพิวเตอร์ในเครือข่ายได้ ประกอบไปด้วย
  - หน้าจอในขณะนั้นของเครื่องลูกข่ายโดยสามารถดูได้มากกว่า 1 เครื่องในเวลาเดียวกัน
  - ข้อมูลเครื่องคอมพิวเตอร์ เช่น ชื่อผู้ใช้ในขณะนั้น ชื่อเครื่อง หมายเลข IP Address
  - แสดงสถานะของเครื่องลูกข่ายในขณะนั้นว่าติดต่อกับเครื่องเซิร์ฟเวอร์อยู่หรือไม่
3. ความสามารถในการควบคุมเครื่องคอมพิวเตอร์ในเครือข่ายได้ ประกอบไปด้วย
  - ทำการแพร์ภาพหน้าจอของเครื่องผู้สอนไปยังเครื่องผู้เรียนได้
  - ทำการล็อกแ่งเป็นอักขระ(keyboard) และเมาส์(mouse) ไม่ให้ใช้งานได้
  - การส่งข้อความหาเครื่องคอมพิวเตอร์ในเครือข่าย เช่น การตักเตือนนักเรียน การแจ้งข่าวสั้นๆ ให้ นักเรียน
4. สามารถรับ-ส่งเพิ่มข้อมูลระหว่างเครื่องผู้สอนและเครื่องผู้เรียนได้ เช่น ในกรณีผู้สอนได้สั่งงานหรือการบ้านแก่ผู้เรียน สามารถทำการส่งผ่านโปรแกรมไปยังเครื่องผู้เรียนทั้งหมดได้ทันที

## 1.6 ขั้นตอนของการศึกษา (Process of the study)

ปัญหาพิเศษนี้มีขั้นตอนของการศึกษา ดังนี้

1. ศึกษาระบบ Computer Network Management
2. ศึกษาการทำงานของ protocol TCP/IP
3. ศึกษาการทำงานของ Winsock library และ socket connection
4. ศึกษาการทำงานของ Remote connection บน Windows XP
5. ศึกษาการทำงานของโปรแกรม Microsoft Visual Studio 2005 และ .net framework
6. ออกแบบโปรแกรม
7. พัฒนาโปรแกรม
8. ทดสอบการทำงานของโปรแกรมในสภาพแวดล้อมจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ปรับปรุงและแก้ไขปัญหาโปรแกรมที่เกิดขึ้น
10. สรุปการทำงานและรวบรวมข้อมูลในการทำปัญหาพิเศษทั้งหมดมาดำเนินการจัดทำเป็นเอกสาร

### 1.7 ข้อจำกัดของการศึกษา (Limitation of the study)

เนื่องจากปัญหาพิเศษนี้พัฒนาตามทฤษฎีที่ได้ศึกษามาทำให้มีข้อจำกัดของการศึกษา ดังนี้

1. โปรแกรมจะสามารถทำงานได้เฉพาะบนระบบปฏิบัติการ Windows XP เท่านั้น
2. การตรวจสอบและควบคุมบนระบบเครือข่ายต้องกระทำผ่านทางเครื่องที่เป็นแม่ข่ายเท่านั้น
3. สามารถเชื่อมต่อคอมพิวเตอร์ในเครือข่ายได้ไม่เกิน 50 เครื่อง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและแนวความรู้ต่างๆ ที่เกี่ยวข้อง

#### 2.1 ทีซีพี/ไอพี (TCP/IP หรือ Transmission Control Protocol/Internet Protocol)

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโพรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้อัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โพรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

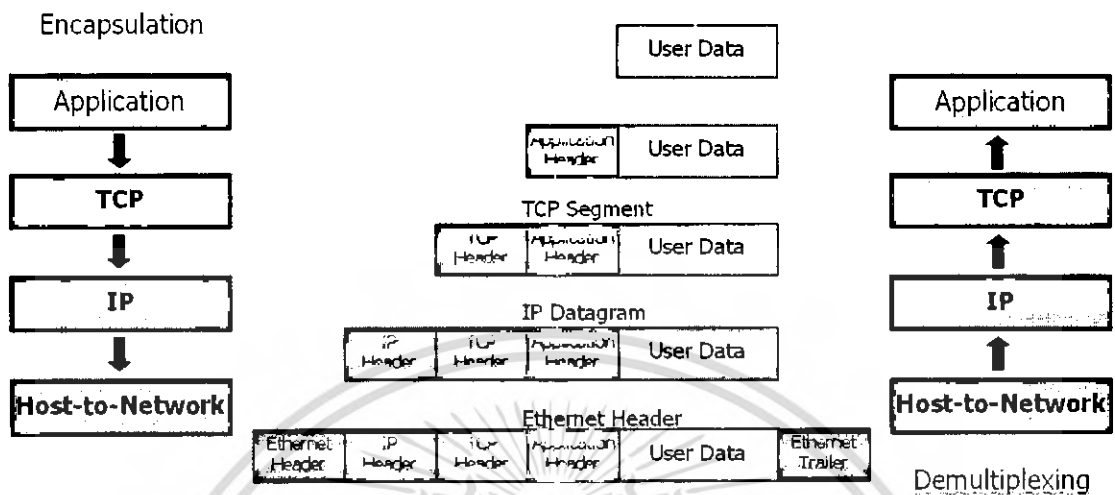
ชุดโพรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้ครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเทอร์เน็ต ทำให้ทีซีพี/ไอพีเป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

ทีซีพี/ไอพีมีจุดประสงค์ของการสื่อสารตามมาตรฐาน สามประการคือ

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่จุดต่อ (node) กลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบทันทีหรือการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

##### 2.1.1 Encapsulation / Demultiplexing

การส่งข้อมูลผ่านในแต่ละเลเยอร์ แต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมากับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่า Header ภายในส่วนหัวข้อมูลจะบรรจุข้อมูลที่สำคัญของโพรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็เกิดกระบวนการทำงานย้อนกลับคือ โพรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็นส่วนหัวข้อมูลก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing

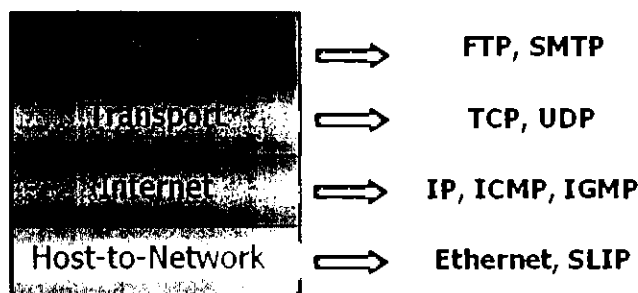


รูปที่ 2.1 ขั้นตอนการ Encapsulation และ Demultiplexing

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละเลเยอร์มีชื่อเรียกแตกต่างกัน ดังนี้

- ข้อมูลที่มาจากผู้ใช้หรือก็คือข้อมูลที่ผู้ใช้เป็นผู้ป้อนให้กับโปรแกรมประยุกต์ (Application) เรียกว่าข้อมูลผู้ใช้ (User Data)
- เมื่อโปรแกรมประยุกต์ได้รับข้อมูลจากผู้ใช้ก็จะนำมาประกอบกับส่วนหัวของโปรแกรมประยุกต์ เรียกว่าข้อมูลโปรแกรมประยุกต์ (Application Data) และส่งต่อไปยังโพรโตคอลทีซีพี
- เมื่อโพรโตคอลทีซีพีได้รับข้อมูลโปรแกรมประยุกต์ก็จะนำมารวมกับส่วนหัวข้อมูลของโพรโตคอลทีซีพี เรียกว่าทีซีพีเชกเมนต์ และส่งต่อไปยังโพรโตคอลไอพี
- เมื่อโพรโตคอลไอพีได้รับทีซีพีเชกเมนต์ก็จะนำมารวมกับส่วนหัวข้อมูลของโพรโตคอลไอพีเรียกว่าไอพีเดทาแกรม (IP Datagram) และส่งต่อไปยังชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)
- ในระดับชั้นโฮสต์-เครือข่ายจะนำไอพีเดทาแกรมมาเพิ่มการแก้ไขความผิดพลาด (Error Correction) และตัวบ่งชี้ (flag) เรียกว่าอีเทอร์เน็ตเฟรม (Ethernet Frame) ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

ในแต่ละชั้นของโครงสร้างทีซีพี/ไอพีสามารถอธิบายได้ดังนี้



รูปที่ 2.2 โครงสร้างทีซีพี/ไอพี

#### 2.1.1.1 ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

โพรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้ เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสารไอพีมาแล้วส่งไปยังจุดต่อที่ระบุไว้ในเส้นทางเดินข้อมูลทางด้านผู้รับก็จะทำงานในทางกลับกัน คือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร

#### 2.1.1.2 ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer)

ใช้ประเภทของระบบการสื่อสารที่เรียกว่า ระบบเครือข่ายแบบสลับช่องสื่อสารระดับกลุ่มข้อมูล (packet-switching network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่ากลุ่มข้อมูล (Packet) สามารถไหลจากจุดต่อผู้ส่งไปตามจุดต่อต่างๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ หากว่ามีการส่งกลุ่มข้อมูลออกมาเป็นชุดโดยมีจุดหมายปลายทางเดียวกันในระหว่างการเดินทางในเครือข่าย กลุ่มข้อมูลแต่ละตัวในชุดนี้ก็จะไปอิสระแก่กันและกัน ดังนั้นกลุ่มข้อมูลที่ส่งไปถึงปลายทางอาจจะไม่เป็นไปตามลำดับก็ได้

##### 1) IP (Internet Protocol)

ไอพีเป็นโพรโตคอลในระดับชั้นเครือข่ายทำหน้าที่จัดการเกี่ยวกับเลขที่อยู่และข้อมูล และควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของกลุ่มข้อมูล ซึ่งกลไกในการหาเส้นทางของไอพีจะมีความสามารถในการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางได้ระหว่างการส่งข้อมูล และมีระบบการแยกและประกอบเดตาแกรม (datagram) เพื่อรองรับการส่งข้อมูลระดับเส้นทางเชื่อมโยงข้อมูลที่มีขนาด MTU (Maximum

Transmission Unit) ที่แตกต่างกัน ทำให้สามารถนำไอพีไปใช้บนโพรโตคอลอื่นได้หลากหลาย เช่น อีเทอร์เน็ต, วงแหวนโทเค็น หรือ Apple Talk

การเชื่อมต่อของไอพีเพื่อทำการส่งข้อมูล จะเป็นแบบไม่ต่อเนื่องหรือเกิดเส้นทางการเชื่อมต่อในทุกๆ ครั้งของการส่งข้อมูล 1 เดททาแกรม โดยจะไม่ทราบถึงข้อมูลเดททาแกรมที่ส่งก่อนหน้าหรือส่งตามมา แต่การส่งข้อมูลใน 1 เดททาแกรม อาจจะมีการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (fragmentation) และถูกนำไปรวมเป็นเดททาแกรมเดิมเมื่อถึงปลายทาง

4-bit Version	Header Length	8-bit Type of Service	16-bit Total Length in Byte	
16-bit Identification		3-bit Flag	16-bit Fragment Checksum	
8-bit Time to Live (TTL)	8-bit Protocol		16-bit Header Checksum	
32-bit Source IP Address				
32-bit Destination IP Address				

รูปที่ 2.3 ส่วนหัวของไอพี

ส่วนหัวของไอพี โดยปกติจะมีขนาด 20 ไบต์ยกเว้นในกรณีที่มีการเพิ่มทางเลือกบางอย่าง เขตข้อมูลส่วนหัวของไอพีส่วนต่างๆ จะมีความหมายดังนี้

- Version : หมายเลขรุ่นของโพรโตคอล ที่ใช้งานในปัจจุบันคือ รุ่นที่ 4 (IPv4) และรุ่นที่ 6 (IPv6)
- Header Length : ความยาวของส่วนหัว โดยทั่วไปถ้าไม่มีส่วนทางเลือก จะมีค่าเป็น 5 (5\*32 บิต)
- Type of Service (TOS) : ใช้เป็นข้อมูลสำหรับอุปกรณ์จัดเส้นทาง (router) ในการตัดสินใจเลือกเส้นทางข้อมูลในแต่ละเดททาแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว

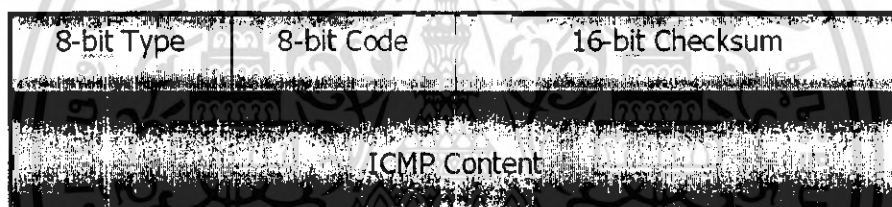
- d. Length : ความยาวทั้งหมดเป็นจำนวนไบต์ของเดทาแกรม ซึ่งด้วยขนาด 16 บิตของเขตข้อมูล จะหมายถึงความยาวสูงสุดของเดทาแกรม คือ 65535 ไบต์ (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดในลิงค์เลเยอร์ และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทาง โปรแกรมประยุกต์ส่วนใหญ่จะมีขนาดของเดทาแกรมไม่เกิน 512 ไบต์
- e. Identification : เป็นหมายเลขของเดทาแกรมในกรณีที่มีการแยกเดทาแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มีการระบุตัวเดียวกันมารวมกัน
- f. Flag : ใช้ในกรณีที่มีการแยกเดทาแกรม
- g. Fragment offset : ใช้ในการกำหนดตำแหน่งข้อมูลในเดทาแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง
- h. Time to live (TTL) : กำหนดจำนวนครั้งที่มากที่สุดที่เดทาแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และอุปกรณ์จัดเส้นทางสุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิดการหมดเวลารอในระหว่างการส่งข้อมูล
- i. Protocol : ระบุโพรโตคอลที่ส่งในเดทาแกรม เช่น ทีซีพี , ยูดีพี หรือ ICMP
- j. Header checksum : ใช้ในการตรวจสอบความถูกต้องของข้อมูลใน ส่วนหัว
- k. Source IP address : หมายเลขไอพี ของผู้ส่งข้อมูล
- l. Destination IP address : หมายเลขไอพี ของผู้รับข้อมูล
- m. Data : ข้อมูลจากโพรโตคอลระดับบน

## 2) ICMP (Internet Control Message Protocol)

ICMP เป็นโพรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานะภาพของเดทาแกรมในกรณีที่เกิดปัญหาเกี่ยวกับเดทาแกรม เช่น อุปกรณ์จัดเส้นทางไม่สามารถส่งเดทาแกรมไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยังโฮสต์ต้นทาง

เพื่อรายงานข้อผิดพลาดที่เกิดขึ้น อย่างไรก็ตามก็ไม่มีอะไรรับประกันได้ว่าสาร ICMP ที่ส่งไปจะถึงผู้รับจริงหรือไม่ หากมีการส่งเดทากแกรมออกไปแล้วไม่มีสาร ICMP แจ้งความผิดพลาดกลับมา ก็แปลความหมายได้สองกรณีคือ ข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย หรืออาจจะมีปัญหา ในการสื่อสาร ทั้งการส่งเดทากแกรม และสาร ICMP ที่ส่งกลับมาก็มีปัญหาระหว่างทางก็ได้ ICMP จึงเป็นโพรโตคอลที่ไม่มีความน่าเชื่อถือ ซึ่งจะเป็นหน้าที่ของโพรโตคอลในระดับสูงกว่าชั้นเครือข่ายในการจัดการให้การสื่อสารนั้นๆ มีความน่าเชื่อถือ

ในส่วนของสาร ICMP จะประกอบด้วยแบบชนิด (type) ขนาด 8 บิต Checksum ขนาด 16 บิต และส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตามแบบชนิด และคำสั่ง ดังรูป



รูปที่ 2.4 ส่วนหัวของ ICMP

### 2.1.1.3 ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

แบ่งเป็นโพรโตคอล 2 ชนิดตามลักษณะ ลักษณะแรกเรียกว่า Transmission Control Protocol (TCP) เป็นแบบที่มีการกำหนดช่วงการสื่อสาร ตลอดระยะเวลาการสื่อสาร (connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte stream ที่ไว้ใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่าสาร (message) ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำสารมาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิมที่ซีพีพียังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่งส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

โพรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง มีการตรวจสอบความถูกต้องของข้อมูลแต่

จะไม่มี การแจ้งกลับไปยังผู้ส่ง จึงถือได้ว่าไม่มีการตรวจสอบความถูกต้องของข้อมูล อย่างไรก็ตาม วิธีการนี้มีข้อดีในด้านความเร็วในการส่งข้อมูล จึงนิยมใช้ในระบบ รับ-ให้บริการ (client/server system) ซึ่งมีการสื่อสารแบบ ถาม/ตอบ (request/reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่ง เสียงทางอินเทอร์เน็ต

#### 1) UDP : (User Datagram Protocol)

เป็นโพรโตคอลที่อยู่ในชั้นสื่อสารนำส่งข้อมูล เมื่อเทียบกับโมเดลไอ-เอลไอ (OSI) โดยการส่งข้อมูลของยูดีพี (UDP) นั้นจะเป็นการส่งครั้งละ 1 ชุด ข้อมูล เรียกว่ายูดีพีเดทาแกรม ซึ่งจะไม่มีความสัมพันธ์กันระหว่างเดทาแกรม และจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล

กลไกการตรวจสอบโดย checksum ของยูดีพีนั้นเพื่อเป็นการ ป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหาก เกิดเหตุการณ์ดังกล่าว ปลายทางจะรู้ว่ามิใช่ข้อมูลผิดพลาดเกิดขึ้น แต่มันจะเป็น การตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของยูดีพี หากพบว่า Checksum เกิดข้อผิดพลาดก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มี การแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิด ข้อผิดพลาดในระดับไอพี เช่น ส่งข้อมูลไปไม่ถึง, หมดเวลา ผู้ส่งจะได้รับสาร ข้อผิดพลาดจากระดับไอพี เป็นสารข้อผิดพลาด ICMP แต่เมื่อข้อมูลส่งถึง ปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของยูดีพีเองจะไม่มีที่ยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด

16-bit Source Port	16-bit Destination Port
Lenght	Checksum

รูปที่ 2.5 ส่วนหัวของ ยูดีพี

มีรายละเอียด ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- a. Source Port Number : หมายเลขช่องทางต้นทางที่ส่งเดทาแกรมนี้
- b. Destination Port Number : หมายเลขช่องทางปลายทางที่จะเป็นผู้รับเดทาแกรม
- c. UDP Length : ความยาวของเดทาแกรมทั้งส่วนหัว และข้อมูลนั้น หมายความว่า ค่าที่น้อยที่สุดในเขตข้อมูลนี้คือ 8 ซึ่งเป็นขนาดของส่วนหัว
- d. Checksum : เป็นตัวตรวจสอบความถูกต้องของเดทาแกรมยูดีพี และจะนำข้อมูลบางส่วนในส่วนหัวของไอพีมาคำนวณด้วย

## 2) TCP : (Transmission Control Protocol)

อยู่ในชั้นสื่อสารนำส่งข้อมูล เช่นเดียวกับยูดีพีทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล ซึ่งมีความสามารถและรายละเอียดมากกว่ายูดีพี โดยเดทาแกรมของทีซีพี จะมีความสัมพันธ์ต่อกัน และมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง และมีการสื่อสารอย่างเป็นกระบวนการ

16-bit Source Port Number				16-bit Source Destination Port				
32-bit Sequence Number								
32-bit Acknowledgment Number								
Header Length	6-Bit Reserved	URG	ACK	PUSH	RESET	SYN	FIN	16-bit Windows Size
16-bit TCP Checksum				16-bit Urgent Pointer				
TCP Option								
Data								

รูปที่ 2.6 ส่วนหัวของทีซีพี

มีรายละเอียด ดังนี้

- a. Source Port Number : หมายเลขช่องทางต้นทางที่ส่งเดทาแกรมนี้
- b. Destination Port Number : หมายเลขช่องทางปลายทางที่จะเป็นผู้รับเดทาแกรม
- c. Sequence Number : เขตข้อมูลที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการระบุว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง
- d. Acknowledgment Number : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ
- e. Header Length : โดยปกติความยาวของส่วนหัวที่ซีพี จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ตัวเลือก แต่ต้องไม่เกิน 60 ไบต์
- f. Flag : เป็นข้อมูลระดับบิตที่อยู่ในส่วนหัวของทีซีพี โดยใช้เป็นตัวบอกคุณสมบัติของกลุ่มข้อมูลที่ซีพีขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง ตัวบ่งชี้ มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังนี้

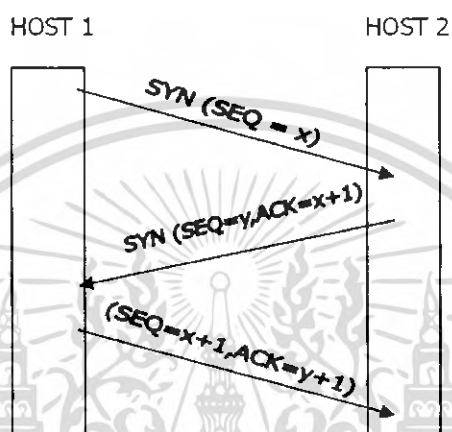
ตารางที่ 2.1 แสดงประเภทของ ตัวบ่งชี้

Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent pointer)
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (reset) เนื่องจากในกรณีที่เกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น โสสัดมีปัญหา ให้เริ่มต้นสื่อสารกันใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

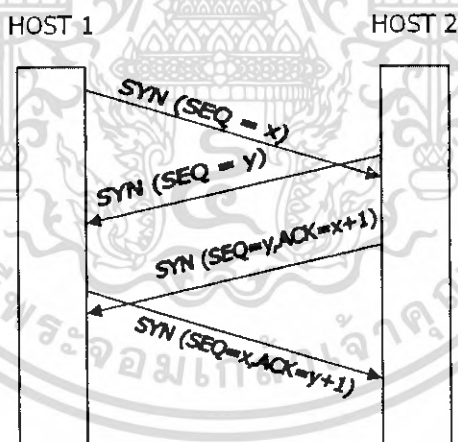
ตัวบ่งชี้ ในส่วนหัวของทีซีพี มีความสำคัญในการกำหนดการทำงานของทีซีพีเซกเมนต์ เนื่องจากข้อมูลในส่วนหัวของทีซีพี จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน ตัวบ่งชี้ จะเป็นตัวกำหนดว่าให้ใช้งานเขตข้อมูลไหน เช่น เขตข้อมูล Acknowledgment number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ

แต่จะมีข้อมูลในเขตข้อมูล ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ ซึ่งถ้าไม่มีตัวบ่งชี้เป็นตัวกำหนดก็อาจจะมีการนำข้อมูลมาใช้ และก่อให้เกิดความผิดพลาดได้

### 2.1) การสื่อสารของทีซีพี



รูปที่ 2.7 ลำดับขั้นตอนการส่ง ทีซีพี



รูปที่ 2.8 ลำดับขั้นตอนที่โฮสต์สองแห่งพยายามสร้างการเชื่อมต่อระหว่างซ็อกเก็ตคู่เดียวกัน

เมื่อเซกเมนต์ CONNECT (SYN = "1" และ ACK = "0") เดินทางมาถึง Entity TCP ที่โฮสต์ปลายทางจะค้นหากระบวนการตามหมายเลขช่องทางที่กำหนดในเขตข้อมูลช่องทางปลายทาง ซึ่งถ้าหากไม่พบก็จะตอบปฏิเสธด้วยเซกเมนต์ที่มี RST = "1" กลับไปยังผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งเซกเมนต์ CONNECT ของผู้ส่งจะถูกส่งต่อไปยังโพรเซส ตามช่องทางที่ระบุซึ่งอาจจะตอบรับหรือตอบปฏิเสธก็ได้ ถ้ากระบวนการนั้นต้องการสื่อสารด้วยก็จะส่งเซกเมนต์ตอบรับกลับไป รูปที่ 2.7 แสดงลำดับขั้นตอนการส่งที่ซีพีเซกเมนต์ในการสร้างการเชื่อมต่อในสภาวะปกติระหว่างผู้ส่งและผู้รับ

ในกรณีที่โฮสต์สองแห่งพยายามสร้างการเชื่อมต่อระหว่างซ็อกเก็ตคู่เดียวกันจะเกิดเป็นลำดับขั้นตอนแสดงในรูปที่ 2.8 ผลสุดท้ายจะมีการเชื่อมต่อเกิดขึ้นเพียงหนึ่งช่องทางเท่านั้น เนื่องจากการเชื่อมต่อในแต่ละช่องทาง จะถูกกำหนดขึ้นโดยใช้หมายเลขซ็อกเก็ตผู้ส่งและผู้รับ ถ้าการเชื่อมต่อลำดับแรกสำเร็จก็就会被บันทึกไว้ในตารางการสื่อสาร เช่น  $(x, y)$  ถ้าการเชื่อมต่อลำดับที่สองสำเร็จในเวลาต่อมา ข้อมูลนี้ก็จะถูกบันทึกไว้ที่เดียวกันคือ  $(x, y)$

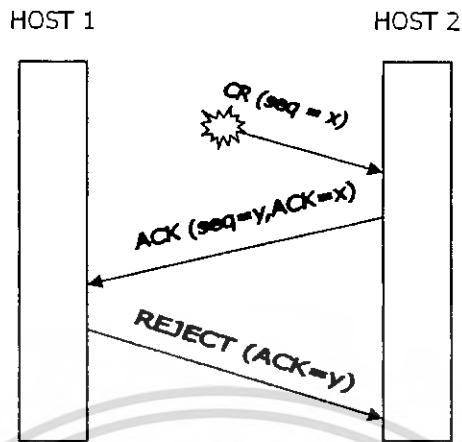
ขั้นตอนในการสร้างการเชื่อมต่อ และการยกเลิกสามารถเขียนอธิบายด้วยเครื่องสถานะจำกัด (Finite state machine) ที่มีการทำงาน 11 สถานะ ในแต่ละสถานะจะมีเหตุการณ์บางอย่างที่เป็นไปได้ซึ่งจะได้รับการตอบสนองด้วยการกระทำที่เหมาะสม ในทางตรงกันข้าม เหตุการณ์ที่เป็นไปไม่ได้จะกลายเป็นข้อผิดพลาดที่จะต้องรายงานให้ทราบ

การเชื่อมต่อเริ่มต้นจากสถานะ CLOSED เมื่อเรียกใช้บริการ LISTEN หรือ CONNECT ก็จะมีการเปลี่ยนสถานะไปจากเดิม และถ้าอีกฝ่ายต้องการเชื่อมต่อด้วย การเชื่อมต่อก็จะเกิดขึ้นและย้ายไปอยู่ในสถานะ ESTABLISHED คือการเชื่อมต่อสมบูรณ์ และเมื่อยกเลิกการติดต่อก็จะกลับไปสู่สถานะ CLOSED อย่างเดิม

## 2.2) การเริ่มต้นการสื่อสารของทีซีพี โดยใช้การบันทึกเวลา แบบ Three-way handshake

Three-way Handshake เป็นวิธีการส่งกลุ่มข้อมูลที่ สามารถช่วยแก้ปัญหาในเรื่องกลุ่มข้อมูลซ้ำซ้อนได้ดี แต่วิธีนี้จำเป็นต้องสร้างช่องสื่อสารให้ได้ก่อนที่จะเริ่มรับ-ส่งข้อมูล อย่างไรก็ตามกลุ่มข้อมูลควบคุมที่ใช้ในการต่อรองค่าตัวแปรสำหรับการสื่อสารต่างๆ อาจเกิดการตกร้างอยู่ในระบบได้ ทำให้การกำหนดค่าหมายเลขลำดับมีปัญหาไปด้วย เช่นการสร้างช่องสื่อสารระหว่างโฮสต์ 1 และ โฮสต์ 2 เริ่มจาก โฮสต์ 1 ขอเริ่มการเชื่อมต่อด้วยการส่งกลุ่มข้อมูล CR (Connection Request) ไปยังโฮสต์ 2 ซึ่งจะมีค่าตัวแปรต่างๆสำหรับการสื่อสารรวมทั้งหมายเลขลำดับ และหมายเลขของสื่อสารไปด้วย ผู้รับคือโฮสต์ 2 ก็จะส่ง ACK (Acknowledge) กลับมายังโฮสต์ 1 แต่ถ้ากลุ่มข้อมูลจากผู้ส่งเกิดสูญหายระหว่างทางและสำเนาข้อมูลที่ยังตกร้างอยู่ ระบบเกิดเดินทางไปถึงผู้รับในภายหลังก็จะทำให้การสร้างช่องสื่อสารใช้การไม่ได้ เนื่องจากมีค่าตัวแปรต่างๆ ไม่ตรงกัน

การใช้ Three-way handshake เป็นการไม่บังคับให้ผู้ส่งและผู้รับข้อมูลจะต้องกำหนดค่าเริ่มต้นของหมายเลขลำดับ เป็นเลขเดียวกัน ทำให้สามารถนำวิธีนี้มาใช้ร่วมกับวิธีการจัดจังหวะการทำงานให้พร้อมกัน (Synchronization) แบบต่างๆได้ แทนที่จะเป็นการใช้วิธีการบันทึกเวลา ดังรูปที่ 2.8 แสดงขั้นตอนการเริ่มต้นการทำงานจากโฮสต์ 1 ไปยังโฮสต์ 2 สมมติให้โฮสต์ 1 เลือกหมายเลขลำดับเป็น "x" และส่งกลุ่มข้อมูล CONNECTION REQUEST ไปยังโฮสต์ 2 โฮสต์ 2 ตอบรับด้วยกลุ่มข้อมูล CONNECTION ACCEPTED ซึ่งจะยอมรับหมายเลขลำดับ "x" พร้อมกับประกาศหมายเลขลำดับ "y" ที่เป็นของตนเอง จากนั้นโฮสต์ 1 ก็จะตอบรับค่าตัวเลือกของโฮสต์ 2 ผ่านทางเซตข้อมูลสำหรับการควบคุมในกลุ่มข้อมูลข้อมูลแรกที่ส่งมา



รูปที่ 2.9 เหตุการณ์ที่กลุ่มข้อมูล TPDU เป็นสำเนาของกลุ่มข้อมูลเก่า

สมมติว่า ได้เกิดปัญหาการสูญหายของกลุ่มข้อมูลใน ขณะที่สำเนาของกลุ่มข้อมูลที่ค้างในระบบเดินทางไปถึงผู้รับแทน รูปที่ 2.9 แสดงเหตุการณ์ที่กลุ่มข้อมูล TPDU เป็นสำเนาของกลุ่มข้อมูลเก่าที่ ฟังจะเดินทางไปถึงโฮสต์ 2 โดยที่โฮสต์ 1 ไม่ทราบ โฮสต์ 2 ก็ จะทำงานตามปกติคือจะตอบรับด้วยการส่งกลุ่มข้อมูล CONNECTION ACCEPTED TPDU กลับมา ที่โฮสต์ 1 ซึ่งโฮสต์ 1 จะสามารถ ตรวจสอบได้ว่า หมายเลขลำดับโฮสต์ 2 ตอบกลับมานั้นเป็น หมายเลขลำดับที่ได้เลิกใช้ไปแล้ว จึงมีการส่งกลุ่มข้อมูล REJECT กลับมายังโฮสต์ 2 เพื่อบอกยกเลิกการทำงาน จะเห็นว่าวิธีการนี้ อาศัยการสื่อสารผ่านกลุ่มข้อมูล 3 ตัวซึ่งเป็นที่มาของคำว่า "การจับมือร่วมสามขั้นตอน" ผลสุดท้าย ทั้งโฮสต์ 1 และโฮสต์ 2 ก็จะไม่มีการ สร้างช่องสื่อสารขึ้นมาจากข้อมูลในสำเนาของกลุ่มข้อมูลเก่าแต่อย่างใด

#### 2.1.1.4 ชั้นสื่อสารการประยุกต์ (Application Layer)

มีโพรโตคอลสำหรับสร้างจอตอร์มินัลเสมือน เรียกว่าเทลเน็ต (TELNET) โพรโตคอลสำหรับการจัดการแฟ้มข้อมูล เรียกว่า FTP และโพรโตคอลสำหรับการ ให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP โดยโพรโตคอลสำหรับสร้างจอตอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่องโฮสต์ที่อยู่ไกลออกไป โดยผ่าน อินเทอร์เน็ต และสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่องโฮสต์นั้น โพรโตคอลสำหรับการจัดการแฟ้มข้อมูลช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่อง อื่น ที่อยู่ในระบบเครือข่ายหรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใดๆก็ได้ โพรโตคอล

สำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

## 2.2 การเขียนโปรแกรมซ็อกเก็ต (Socket Programming)

การพัฒนาโปรแกรมเพื่อให้สื่อสารกันในเครือข่ายนั้น จะต้องพัฒนาโปรแกรมประยุกต์ 2 โปรแกรมด้วยกัน คือ โปรแกรมที่ทำหน้าที่เป็นลูกข่าย และโปรแกรมที่ทำหน้าที่เป็นแม่ข่าย โดยเมื่อดำเนินงาน โปรแกรมทั้ง 2 แล้ว จะอ่านและส่งข้อมูลกันผ่านทางซ็อกเก็ต ซึ่งเรียกโปรแกรมประยุกต์ เช่นนี้ว่า "Client-server application" โดยจะแบ่งออกเป็น 2 ประเภทคือ

### 1) Client-Server ที่ทำงานโดยใช้โพรโตคอลที่ได้นิยามไว้ใน RFC

ผู้พัฒนาโปรแกรมลูกข่าย และแม่ข่ายจะต้องพัฒนาให้โปรแกรมทำงานตามรูปแบบที่กำหนดไว้ในอาร์เอฟซี หรือ เอกสารขอความเห็น (RFC : Request For Comment) เดียวกัน เพื่อให้ลูกข่าย และแม่ข่ายสามารถสื่อสารกันได้อย่างถูกต้อง (ตามอาร์เอฟซีเดียวกัน) เช่น ถ้าต้องการพัฒนาเอฟทีพี (FTP) (RFC959) ผู้ที่พัฒนาโปรแกรมลูกข่ายเอฟทีพีจะต้องให้โปรแกรมทำงานตามข้อกำหนดใน RFC959 อย่างถูกต้อง (หมายเลขช่องทาง, รูปแบบสารที่ทำการรับ-ส่ง) เช่นเดียวกับกับผู้พัฒนาโปรแกรมแม่ข่ายก็ต้องให้โปรแกรมทำงานตามข้อกำหนดใน RFC959 เพื่อที่จะให้ลูกข่ายและแม่ข่ายทำงานร่วมกันได้อย่างถูกต้อง

### 2) Proprietary Client-Server application

ในการพัฒนาโปรแกรมลักษณะนี้ ผู้พัฒนาไม่ต้องอ้างอิงตามอาร์เอฟซี แต่นั่นหมายถึง จะต้องพัฒนาทั้งโปรแกรมประยุกต์ลูกข่าย และโปรแกรมประยุกต์แม่ข่ายเพราะไม่ได้อ้างอิงตามอาร์เอฟซีซึ่งเป็นมาตรฐานสากล ผู้พัฒนาจะต้องพัฒนาโปรแกรมให้ลูกข่าย และแม่ข่ายทำงานร่วมกันได้อย่างถูกต้องตามที่ผู้พัฒนานั้นกำหนดขึ้นเองแต่ผู้พัฒนาต้องระวังไม่ให้ใช้ช่องทางที่รู้จักดี (Well-known port) ที่ได้กำหนดไว้ใน RFCs

ผู้พัฒนาจะต้องทำการเลือกใช้โพรโตคอลการสื่อสารระหว่างทีซีพี และยูดีพีโดยที่ทีซีพีเป็นโพรโตคอลแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร ซึ่งให้ความเชื่อถือได้ระหว่างการส่ง-รับข้อมูลระหว่างระบบปลายทาง และยูดีพีเป็นโพรโตคอลแบบการติดต่อไม่ต่อเนื่อง ซึ่งไม่ได้ให้การรับประกันว่าข้อมูลจะส่งถึงกันหรือไม่

### 2.2.1 Socket Programming with TCP

ในการสื่อสารถึงกันจะต้องประกอบด้วยกระบวนการ (ทั้งลูกข่ายและแม่ข่าย) และชื่อเกิดของกระบวนการที่ทำการแลกเปลี่ยนสารกัน

โปรแกรมลูกข่ายจะต้องทำการส่งสารเพื่อขอการติดต่อกับโปรแกรมแม่ข่าย ในการที่จะให้โปรแกรมแม่ข่ายรับการติดต่อได้นั้น โปรแกรมแม่ข่ายจะต้องดำเนินงานกระบวนการเพื่อรอรับสารไว้ก่อนล่วงหน้า โดยกระบวนการในแม่ข่ายจะสร้างที่ซีพีชื่อเกิด (welcoming socket) เพื่อรอรับสารจากโปรแกรมลูกข่ายที่ส่งมา โดยในการเริ่มต้นการขอการติดต่อนั้น ลูกข่ายจะทำการสร้างชื่อเกิด (เพื่อขอการเชื่อมต่อ) มายังแม่ข่าย เมื่อแม่ข่ายได้รับแล้ว แม่ข่ายจะทำการสร้างชื่อเกิด (connection socket) ขึ้นใหม่ เพื่อที่จะใช้ติดต่อกับลูกข่ายนั้นๆ

### 2.2.2 Socket Programming with UDP

จะเห็นว่า ที่ซีพี หลังจากการแฮนด์เชค (handshake) แล้ว ทั้ง 2 กระบวนการ (กระบวนการลูกข่าย และกระบวนการแม่ข่าย) เสมือนว่าจะได้ท่อ (pipe) ในการสื่อสารกันแล้วจะให้ท่อนี้ตลอดจนกว่าจะปิดการเชื่อมต่อ การส่งข้อมูลที่เพียงแต่เขียนข้อมูลลงไปในท่อ และส่ง โดยไม่ต้องใส่ข้อมูลเลขที่อยู่ไอพีไปกับข้อมูล (ใช้เฉพาะตอนสร้างการเชื่อมต่อ) เนื่องจากที่ซีพีจะจัดการให้ รวมทั้งจัดการกับลำดับในการรับ-ส่งให้ถูกต้อง

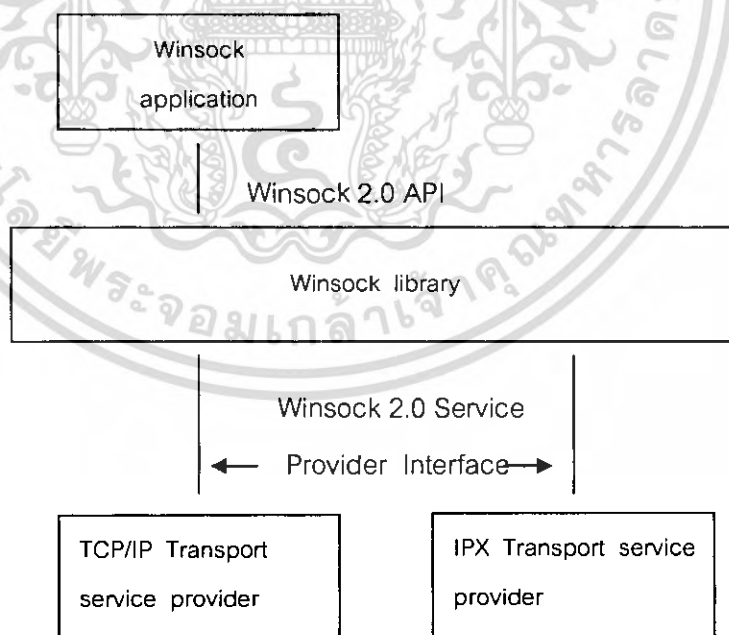
ยูดีพีแตกต่างกับที่ซีพีในหลายๆด้าน อย่างแรก ยูดีพีเป็นบริการแบบการติดต่อไม่ต่อเนื่อง (ไม่มีการแฮนด์เชค ซึ่งจะมีการสร้างท่อระหว่าง 2 กระบวนการ) เพราะว่ายูดีพีไม่มีการสร้างท่อ เมื่อกระบวนการต้องการส่งข้อมูลไปยังกระบวนการอื่น กระบวนการที่ทำการจะต้องแนบที่อยู่ปลายทางไปกับชุดข้อมูลด้วยจะเปรียบเทียบกับบริการรถแท็กซี่ โดยเมื่อคนขึ้นรถแท็กซี่ก็ต้องบอกจุดหมายปลายทางให้คนขับรถแท็กซี่ทราบ อีกทั้งไม่รับประกันว่าข้อมูลที่ส่งจะถึงปลายทางและลำดับที่ได้รับจะถูกต้องหรือไม่ เราสามารถสรุปความแตกต่างในการพัฒนาโปรแกรมระหว่างการใช้ ที่ซีพี และยูดีพีได้ดังนี้

- ยูดีพีไม่ต้องการ handshaking ระหว่าง 2 กระบวนการ ดังนั้นแม่ข่ายไม่ต้องมี welcoming socket
- ไม่ต้องใช้กระแสข้อมูลผูกติดกับชื่อเกิด
- ในการส่งข้อมูลแต่ละชุด จะต้องมีเลขที่อยู่ไอพี และหมายเลขช่องทางของเครื่องปลายทางรวมกับข้อมูลที่ส่งด้วย

### 2.2.3 Windows Socket

Winsock รุ่นปัจจุบันคือรุ่นที่ 2 ที่นำออกมาใช้งานตั้งแต่ปีค.ศ. 1993 การทำงานของ Winsock จะคล้ายคลึงกับ Berkeley Socket ในยูนิกซ์ (Unix) ซึ่งออกแบบมาเพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรมที่เกี่ยวข้องกับเครือข่าย

เป้าหมายหลักของการพัฒนา Winsock คือเปิดให้มีการพัฒนาโปรแกรมด้านเครือข่ายให้สามารถทำงานร่วมกับโพรโตคอลทีซีพี/ไอพีจากผู้ผลิตรายใดก็ได้ การทำเช่นนี้มีความจำเป็นเนื่องจากการทำงานร่วมกับทีซีพี/ไอพีนั้น ผู้ผลิตแต่ละรายจะมีเทคนิคเฉพาะของตนเอง แต่ความสามารถของ Winsock รุ่นแรก (รุ่นที่ 1.1) กลไกของการขนส่งข้อมูลจะถูกจำกัดโดยการทำงานของทีซีพี/ไอพี ซึ่งได้พัฒนาต่อมาในรุ่นที่ 2 ให้สามารถทำงานกับโพรโตคอลเพื่อการส่งข้อมูลอื่นๆ เช่น เอทีเอ็ม (ATM : Asynchronous Transfer Mode) และ IPX/SPX (Internet Packet Exchange/Sequenced Package Exchange) ได้ กลไกการทำงานในการขนส่งข้อมูลที่แม้จะแตกต่างกัน ก็สามารถใช้งานได้พร้อมกันโดยโปรแกรมที่แตกต่างกัน ในรุ่นที่ 2 มีความสามารถในการเข้ากันได้กับรุ่นก่อนหน้าเพื่อช่วยให้โปรแกรมที่ทำงานบนรุ่นที่ 1.1 สามารถทำงานได้โดยไม่ต้องมีการดัดแปลงใดๆ



รูปที่ 2.10 โครงสร้างของ window socket

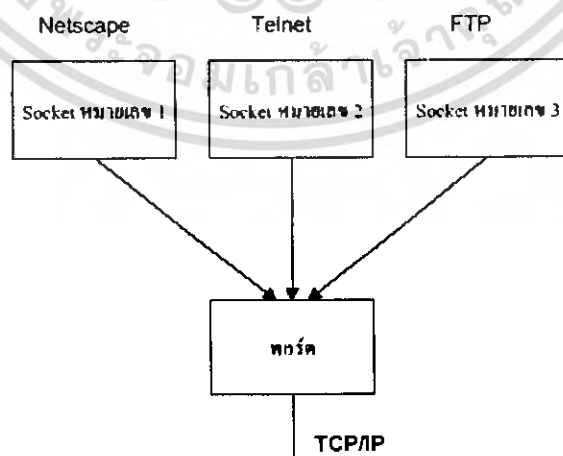
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.3.1 ภาพสถาปัตยกรรมของ Winsock

บริษัทไมโครซอฟท์และอินเทลจัดเตรียมซอฟต์แวร์ที่รู้จักกันโดยทั่วไปว่า Winsock Dynamic Link Library (Winsock DLL) ซึ่งในทางปฏิบัติจริงนั้น DLL มีหลายรุ่นที่ตามรุ่นที่ของวินโดวส์ โดยในแต่ละรุ่นที่ก็จะมีชื่อเรียกที่ต่างกันไป เอพีไอ หรือส่วนต่อประสานโปรแกรมประยุกต์ (API: Application Programming Interface) ของ Winsock DLL มีที่ต่างกันไปสองแบบนั่นคือ ที่ชั้นบนสุดจะเป็นส่วนของ Winsock 2.0 API เพื่อประโยชน์ของโปรแกรมประยุกต์ต่างๆ ในส่วนล่างจะเป็น Winsock Service ที่เอื้อประโยชน์แก่ผู้พัฒนาซอฟต์แวร์ระบบที่ต้องการต่อขยาย หรือเพิ่มเติมกลไกการทำงานในการส่งข้อมูลแบบอื่นๆ

### 2.2.3.2 ความหมายของ Winsock

การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์บนระบบอินเทอร์เน็ต เช่นการ เพลเน็ตเข้าไปในระบบยูนิกซ์ ระบบจะต้องสามารถรองรับการทำงานแบบหลายผู้ใช้ ได้นั้นหมายความว่า ระบบจะต้องสามารถมีการติดต่อสื่อสารกับผู้ใช้ ได้พร้อมๆ กัน หรือ เมื่อมีการใช้งานโปรแกรมที่เกี่ยวกับอินเทอร์เน็ต เช่น Netscape, เพลเน็ต, เอพีไอ พี ซึ่งโปรแกรมต่างๆเหล่านี้จะต้องทำงานแยกกันโดยอิสระ ดังนั้นจึงได้เกิด Winsock ขึ้นมา นั่นก็คือ Winsock สามารถที่จะสร้างช่องทางสื่อสารขึ้นมาได้หลายๆ ช่องทาง (ขึ้นอยู่กับเวอร์ชันของ Winsock) และแต่ละช่องทางสื่อสารสามารถที่จะส่งข้อมูลได้ โดยไม่ขึ้นกับช่องทางสื่อสารอื่นๆ ดังรูป

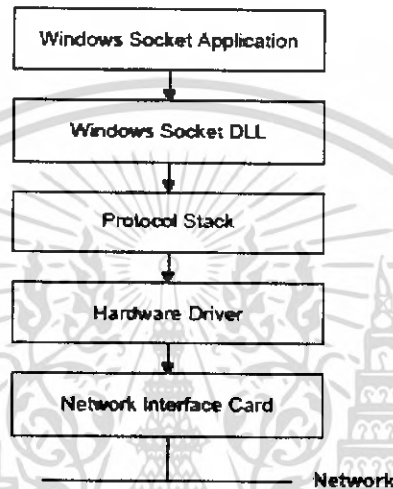


รูปที่ 2.11 การสื่อสารผ่านซ็อกเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.3.3 ระดับของ Winsock ในส่วนของ TCP/IP

Winsock ทำหน้าที่เป็นตัวกลางในการรับส่งข้อมูลจากโปรแกรมประยุกต์ กับทีซีพี/ไอพี จากนั้นทีซีพี/ไอพีจึงส่งข้อมูลลงไปในระบบอินเทอร์เน็ต ดังรูป



รูปที่ 2.12 ระดับของ ซ็อกเก็ต

## 2.3 Microsoft Visual C# .NET

ภาษา C# เป็นภาษาที่ถูกออกแบบมาเพื่อรองรับการทำงานในแพลตฟอร์มที่รองรับ .NET โดยมีแนวของภาษาเป็นแบบของการเขียนโปรแกรมเชิงวัตถุสมัยใหม่ (Modern Object Oriented Programming) โดยที่จุดยืนของภาษา C# จะอยู่ที่การอาศัยไวยากรณ์ที่ปรับปรุงมาจากภาษา C/C++ ร่วมกับความง่ายของภาษา Visual Basic และลักษณะการเขียนโปรแกรมเชิงวัตถุของภาษา Java โดยแนวความคิดของการเขียนโปรแกรมแบบโปรแกรมเชิงวัตถุสมัยใหม่เกิดจากการที่บริษัท ไมโครซอฟต์พัฒนาคลาส (CLASS) ต้นแบบต่างๆที่เรียกว่า Base Class Library แล้วนำมาจัดหมวดหมู่ให้เป็นระเบียบ เมื่อต้องการเรียกใช้งานคลาสใดก็จะอาศัยระบบเนมสเปซ (Name Spaces System) เข้ามาช่วยในการระบุคลาสต้นแบบต่างๆ เพื่อให้ผู้พัฒนาสามารถนำอ็อบเจกต์ต่างๆ ที่อยู่ในคลาสนั้นๆ ออกมาใช้งานได้ นอกจากนั้นภาษา C# ยังสามารถเรียกใช้ส่วนโปรแกรม (Components) ซึ่งเป็นชิ้นส่วนทางซอฟต์แวร์มาประกอบเป็นโปรแกรม ทำให้ลดเวลาในการพัฒนาโปรแกรมได้เป็นอย่างดี

### 2.3.1 Namespaces

ระบบเนมสเปซทำหน้าที่รวบรวมอ็อบเจกต์ต้นแบบต่างๆ เข้าด้วยกันแล้วแบ่งออกเป็นหมวดหมู่เป็นสัดส่วนแล้วกำหนดให้เป็นอ็อบเจกต์หลักของระบบ เพื่อให้ผู้พัฒนาโปรแกรมสะดวกในการเรียกใช้เนมสเปซชั้นบนสุดในภาษา C# เรียกว่า System มีเพียงตัวเดียวประกอบด้วย เนมสเปซย่อยๆ สืบทอดลงมามากมายมหาศาล แยกเป็นกลุ่มๆ เป็นชั้นๆ โดยที่แต่ละลำดับชั้นจะคั่นด้วยเครื่องหมายจุด (.) เช่น System.Windows.Forms, System.Data, System.Web เป็นต้น เมื่อใดก็ตามที่เราต้องการเรียกใช้งานอ็อบเจกต์ต้นแบบใดๆ เราต้องระบุชื่อเนมสเปซเพื่อให้การเรียกใช้งานอ็อบเจกต์ดังกล่าวไม่เกิดข้อผิดพลาดขึ้นมา

## 2.4 .NET Framework



คือกรอบการทำงานของการเขียนโปรแกรมที่บริษัทไมโครซอฟต์คิดขึ้นมา เพื่อรองรับการติดต่อ-สื่อสาร เพื่อแลกเปลี่ยนข้อมูลระหว่างกันหรือแลกเปลี่ยนข้อมูลระหว่างแพลตฟอร์มให้มีความสมบูรณ์ยิ่งขึ้น โดยอาศัยภาษา XML (Extensible Markup Language) ทำหน้าที่เป็นตัวกลางในการแลกเปลี่ยนข้อมูลระหว่างแพลตฟอร์มเพิ่มของฐานข้อมูล

Microsoft .NET Framework เป็นซอฟต์แวร์แบบส่วนประกอบ (Software component) ที่ถูกสร้างขึ้นโดยบริษัทไมโครซอฟต์ โดยได้รับแรงบันดาลใจจากความสำเร็จของบริษัทซัน (Sun) ใน Java Enterprise Edition (J2EE) API .NET Framework สามารถติดตั้งลงไปยังระบบปฏิบัติการของไมโครซอฟต์ได้เท่านั้น (ในขณะนี้) และทางไมโครซอฟต์ก็หวังว่า .NET Framework นี้จะเป็นส่วนสำคัญสำหรับผู้พัฒนาโปรแกรมบนระบบปฏิบัติการของทางไมโครซอฟต์ทั้งในปัจจุบันและอนาคต .NET Framework มีส่วนประกอบที่สำคัญคือ pre-coded solution ที่โปรแกรมส่วนใหญ่ต้องมีการเรียกใช้และการจัดการ การกระทำของโปรแกรมที่ถูกเขียนมาโดยเฉพาะของภาษา .NET

Pre-coded solution จะรู้จักกันในชื่อของเนมสเปซหรือคลาสไลบรารี (class library) หรือ ฟังก์ชัน ซึ่งมีส่วนของคำสั่งที่ครอบคลุมการเขียนโปรแกรมในด้านต่างๆ เป็นวงกว้าง เช่น การสร้างส่วนติดต่อผู้ใช้, การเข้าถึงข้อมูล, การเข้ารหัส, อัลกอริทึมทางตัวเลข, การติดต่อสื่อสารบนเครือข่าย

ผู้พัฒนาโปรแกรมจะนำฟังก์ชันเหล่านี้ไปผนวกกับคำสั่งโปรแกรมของตัวเองเพื่อสร้างเป็นโปรแกรมประยุกต์

#### 2.4.1 ความสำคัญ

สำหรับนักพัฒนาซอฟต์แวร์ .NET Framework เป็นก้าวแห่งความเปลี่ยนแปลงครั้งสำคัญ เพราะ .NET Framework ได้มีการจัดการคุณสมบัติและความรับผิดชอบของระบบปฏิบัติการ ซึ่งในสมัยก่อนภาษาแต่ละภาษาต้องทำการจัดการเอง จุดเด่นของ .NET Framework ที่สำคัญประกอบไปด้วย

- รับประกันได้ว่า framework นี้จะสามารถเรียกใช้ได้ทุกครั้งที่ต้องการในทุกภาษาของ .NET
- คุณสมบัติในการติดต่อและเรียกใช้ framework จะไม่ขึ้นกับภาษาของโปรแกรม ทุกภาษาจะมีวิธีการติดต่อที่เหมือนกัน
- ไม่ว่าจะเรียกใช้ framework ในภาษาได้ ก็จะได้ผลลัพธ์ที่เหมือนกัน
- สามารถทำงานร่วมกับระบบปฏิบัติการจัดการควบคุมพฤติกรรมของโปรแกรม เพื่อเหตุผลในด้านความปลอดภัยของระบบปฏิบัติการ
- ช่วยลดความซับซ้อนและข้อจำกัดในการติดต่อระหว่างโปรแกรมต่อโปรแกรม แม้จะเขียนโดยภาษาของ .NET ที่ต่างกัน (เช่น VB.net กับ C#.net เป็นต้น)

#### 2.4.2 เป้าหมายของการออกแบบและคุณสมบัติที่สำคัญ

##### 2.4.2.1 Interoperability

เนื่องจากในปัจจุบันมีคลังโปรแกรม (library) ประเภท COM (Component Object Model) ได้ถูกสร้างขึ้นมาก่อนเป็นจำนวนมากแล้ว .NET Framework สามารถทำงานร่วมกันระหว่างคำสั่งโปรแกรมที่สร้างขึ้นใหม่กับคลังโปรแกรมที่มีอยู่เดิมได้

##### 2.4.2.2 Common Runtime Engine

การแปลโปรแกรม (compile) ของ .NET Framework จะทำการแปลภาษาของการโปรแกรมไปเป็นภาษากลาง (intermediate language) หรือที่รู้จักในชื่อ Common Intermediate Language (CIL) ซึ่งทาง ไมโครซอฟต์ เรียกว่า (Microsoft

Intermediate Language หรือ MSIL หรือ IL) ซึ่งในขั้นตอนการแปลง IL นี้ไปเป็นรหัสเครื่อง (machine code) จะใช้การแปลโปรแกรมแบบ Just-in-Time Compilation (JIT) ซึ่งหลักการทำงานทั้งหมดนี้ เรียกว่า Common Language Infrastructure (CLI) แต่สำหรับไมโครซอฟต์ที่ได้นำวิธีนี้ไปใช้กับ CIL จะเรียกว่า Common Language Runtime (CLR)

#### 2.4.2.3 Language Independence

.NET Framework จะมี Common Type System (CTS) ซึ่งทำหน้าที่นิยามแบบชนิดข้อมูล (data type) และรูปแบบของการเขียนโปรแกรม โดย CTS จะทำงานร่วมกับ CLR เพื่อรองรับการทำงานที่ไม่จำกัดทางด้านภาษาทางโปรแกรมนี้

#### 2.4.2.4 Base Class Library (BCL)

บางครั้งจะถูกเรียกว่า Framework Class Library (FCL) เป็นแหล่งรวบรวมคลาสซึ่งถูกห่อหุ้ม (encapsulate) ฟังก์ชันต่างๆไว้ให้ผู้พัฒนาโปรแกรมได้ใช้งาน

#### 2.4.2.5 Simplified Deployment

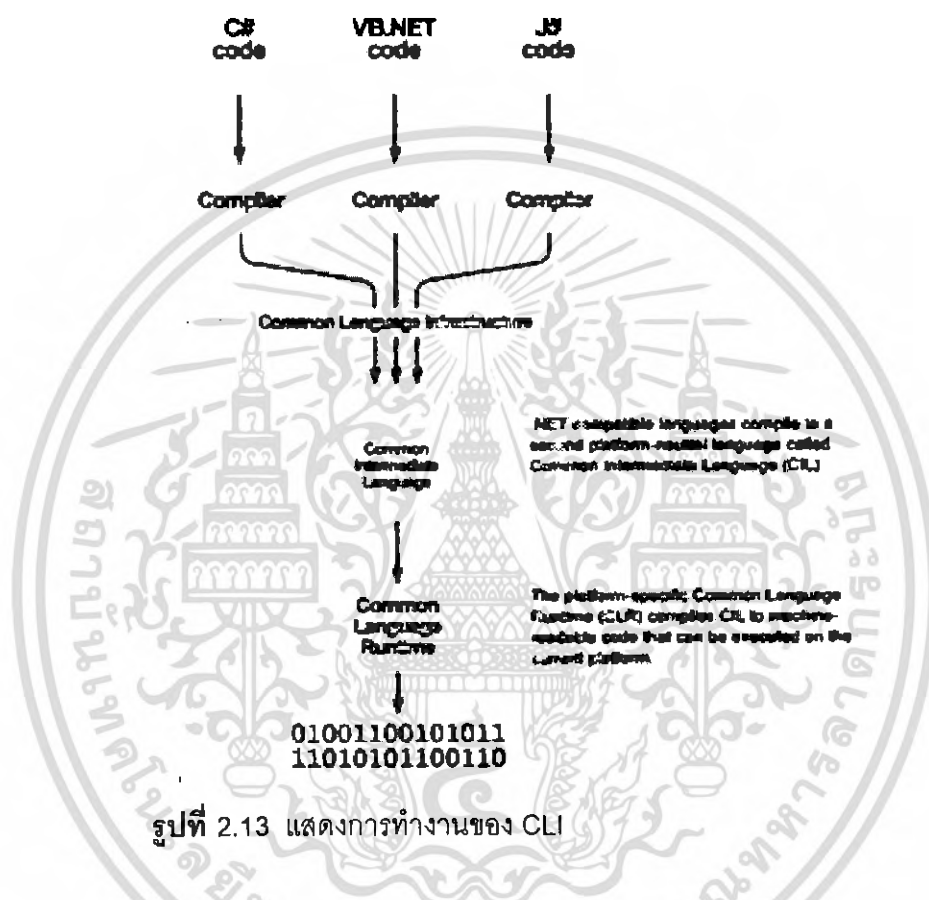
.NET Framework สามารถจัดการช่วยเหลือในการติดตั้งโปรแกรมประยุกต์ เช่นการตั้งค่า Registry, การกระจายแฟ้ม, การเชื่อมต่อและจัดการกับแฟ้ม DLL (Dynamic Link Libraries)

#### 2.4.2.6 Security

.NET Framework อนุญาตให้คำสั่งโปรแกรมสามารถทำงานได้ในขณะที่มีระดับการรักษาความปลอดภัยที่ต่างกัน (trust level) โดยไม่ต้องทำการแบ่ง sandbox (ลักษณะการรักษาความปลอดภัยของคอมพิวเตอร์แบบหนึ่ง)

## 2.4.3 สถาปัตยกรรม .NET Framework

### 2.4.3.1 Common Language Infrastructure (CLI)



รูปที่ 2.13 แสดงการทำงานของ CLI

CLI เป็นส่วนประกอบที่มีความสำคัญมาก ในสถาปัตยกรรมของ .NET Framework เป็นการทำให้ผู้พัฒนาไม่จำเป็นต้องพึ่งพารายภาษาใดเพียงภาษาหนึ่งเท่านั้น รวมถึงการทำงานต่างๆที่ไม่ต้องขึ้นกับภาษา เช่น component, exception handling, garbage collection, security, interoperability CLI ของ ไมโครซอฟต์ จะถูกเรียกว่า CLR ซึ่ง CLI นี้จะประกอบไปด้วย 5 ส่วนหลักคือ

- Common Type System (CTS)
- Common Language Specification (CLS)
- Common Intermediate Language (CIL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Just-in-Time Compiler (JIT)
- Virtual Execution System (VES)

#### 2.4.3.2 Assemblies

MSIL code จะถูกจัดการแปลงเป็นแฟ้มแบบ Portable Executable (PE) เช่นแฟ้ม .exe, .dll ผ่านตัวแอสเซมเบลอร์ (assembler) ที่ชื่อ .NET assemblies ซึ่งแอสเซมเบลอร์จะประกอบไปด้วยเมทาเดตา (metadata) ของการทำแอสเซมบลี, ชื่อ, หมายเลข รุ่น, public key ค่าหลักๆจะเป็นเมทาเดตา และชื่อของแอสเซมบลี ส่วนค่าอื่น ๆ นั้นจะไว้ใช้เพิ่มเติมเป็นทางเลือก

#### 2.4.3.3 Metadata

CIL ทั้งหมดใน .NET Framework จะถูกระบุไว้ใน .NET Metadata CLR จะทำการตรวจสอบข้อมูลจากเมทาเดตา ว่าเมท็อดที่ถูกเรียกนั้นเป็นเมท็อดที่ต้องการโดยทั่วไปแล้ว เมทาเดตาจะถูกสร้างโดยคอมไพเลอร์ของภาษาแต่อย่างไรก็ตาม นักพัฒนาโปรแกรมสามารถเพิ่มข้อมูลในเมทาเดตาเพิ่มเติมได้ด้วย

#### 2.4.4 ความปลอดภัย

.NET Framework มีมาตรการรักษาความปลอดภัยของตัวเอง โดยจะประกอบไปด้วย 2 ส่วนคือ

1. Code Access Security (CAS) ใช้หลักการตรวจสอบค่าที่เรียกว่า "evidence" ค่านี้จะเป็นค่าเฉพาะของแอสเซมบลีแต่ละตัว CAS ใช้ค่านี้เพื่อตรวจสอบสิทธิการใช้คำสั่งโปรแกรมแต่ละส่วนเมท็อด CLR จะทำการ stack walk ตรวจสอบสิทธิของแต่ละเมท็อด หากมีส่วนไหนก็ตามที่แอสเซมบลี ไม่ได้รับสิทธิ security exception ก็จะถูกสั่งให้ทำงาน

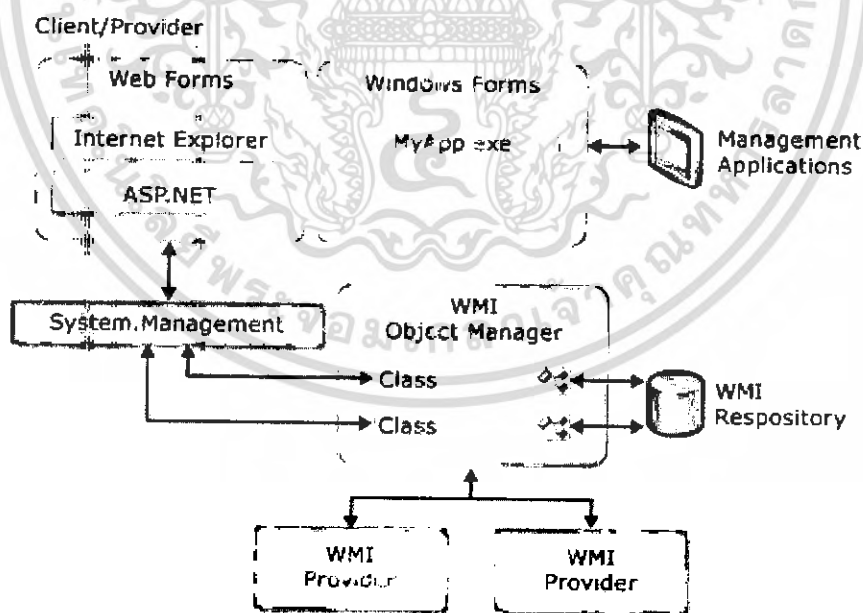
2. Validation และ Verification ในการทำ validation CLR จะตรวจสอบว่าแอสเซมบลีมี เมทาเดตา, CLI และตารางภายใน มีความถูกต้องหรือไม่ ส่วน verification จะเป็นการตรวจสอบว่าคำสั่งโปรแกรมที่สร้างขึ้นนั้นไม่มีความเสี่ยง โดยจะมีอัลกอริทึมที่คอยยับยั้ง แต่บางครั้งการ verification ก็จะไม่ถูกทำ เช่นกรณีที่แอสเซมบลีได้รับการอนุญาตให้ข้ามขั้นตอนนี้ หรือเป็นกรณีที่คำสั่งโปรแกรมถูกติดตั้งในเครื่องท้องถิ่น (local machine) นั้นเอง

## 2.5 Windows Management Instrumentation (WMI)

Windows Management Instrumentation เป็นคลาสหนึ่งของเนมสเปซ System.Management ของ .NET Framework ที่ช่วยในการจัดการกับข้อมูลระดับ enterprise แบบอัตโนมัติ

WMI ทำงานได้โดยอาศัยการติดตั้งคลาสแบบแผนที่เขาไว้ก่อน โดยคลาสเหล่านี้จะทำหน้าที่ตรวจสอบ และตั้งค่ากับโปรแกรม, ระบบ, ส่วนประกอบของเครือข่าย และฮาร์ดแวร์ในระบบ enterprise เช่น คลาส Win32\_Process จะมีข้อมูล กระบวนการ ต่างๆทั้งหมดที่ทำงานอยู่ของเครื่องคอมพิวเตอร์, คลาส Win32\_LogicalDisk จะมีข้อมูลของอุปกรณ์ดิสต่างๆทั้งหมด เช่น ฮาร์ดดิส, ดีวีดี-รอม เป็นต้น การดึงข้อมูลจาก WMI คลาสสามารถทำได้จากการใช้ WMI WQL Query Language ซึ่งเป็นภาษาที่เป็นซับเซตของภาษา SQL โดยสิ่งที่จะสามารถดึงข้อมูลจาก WMI จะประกอบไปด้วย

- Windows Forms
- Web Forms/ASP.NET
- โปรแกรมด้านการจัดการระบบ เช่น Management Operations Manager (MOM), Systems Management Server (SMS) เป็นต้น



รูปที่ 2.14 แสดงโครงสร้างการทำงานของคลาส WMI

## 2.6 Window API

Window API หรือที่รู้จักในชื่อ Win API, Win32 API เป็นเอพีไอหรือกล่าวได้ว่าเป็นส่วนติดต่อสำหรับโปรแกรมหรือกลุ่มคลังโปรแกรมที่ถูกสร้างขึ้นมาเพื่อรองรับการทำงานหนึ่ง ซึ่งสำหรับ Win API นี้ก็คือโปรแกรมสำหรับติดต่อไปยังแฟ้มระบบของกลุ่มระบบปฏิบัติการวินโดวส์ โดยเฉพาะการติดต่อในระดับชั้นล่างของระบบปฏิบัติการ คือการติดต่อกับฮาร์ดแวร์ต่างๆผ่านทาง Windows Driver Foundation ซึ่ง Win API ก็คือฟังก์ชันในการควบคุมการทำงานของระบบปฏิบัติการ Win32 จะถูกเก็บไว้ใน ..\Windows\System32 (ใน Windows XP) และ ..\Windows\System (ใน Windows 95,98,me) ซึ่งอยู่ในรูปของแฟ้ม ".DLL" ฟังก์ชันเหล่านี้จะถูกระบบปฏิบัติการเรียกใช้ขณะที่ระบบปฏิบัติการกำลังทำงานอยู่ เราสามารถดึงฟังก์ชันเหล่านี้มาใช้ในการพัฒนาโปรแกรมให้มีความสามารถเหมือนกับระบบปฏิบัติการได้

Win API ได้รับการออกแบบมาสำหรับใช้ทำงานกับภาษา C/C++ แต่อย่างไรก็ดี เนื่องจากภาษา C# เป็นภาษาที่ได้รับการพัฒนาโดยมีพื้นฐานของภาษา C/C++ เป็นทุนเดิม จึงสามารถใช้งาน Win API ได้อย่างดี ประกอบกับ .NET Framework เองก็รองรับการทำงานกับ Win API อยู่แล้ว จึงทำให้การทำงานสามารถทำได้อย่างราบรื่น

### 2.6.1 ประเภทของ Win API

Win API จะประกอบไปด้วย 7 ประเภทหลักคือ

#### 2.6.1.1. Base Services

เป็นส่วนที่เตรียมการติดต่อกับทรัพยากรพื้นฐาน ที่มีในระบบของวินโดวส์ เช่น แฟ้มระบบ (files system), อุปกรณ์ (devices), กระบวนการ (processes), สายใย (threads) ฟังก์ชันการทำงานเหล่านี้จะถูกควบคุมโดยแฟ้ม kernel.exe, kernel32.dll, advapi32.dll

#### 2.6.1.2. Graphics Device Interface

เป็นส่วนที่เตรียมตัวการติดต่อกับข้อมูลการแสดงผลทางด้านกราฟิกต่างๆ เช่น จอภาพ, เครื่องพิมพ์ เป็นต้น ซึ่งการทำงานเหล่านี้จะอยู่ถูกควบคุมโดยแฟ้ม gdi32.dll

#### 2.6.1.3. User Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นส่วนที่เตรียมตัวการติดต่อกับการสร้าง และจัดการหน้าต่างวินโดวส์ และการควบคุมพื้นฐาน เช่น ปุ่มกด และแถบเลื่อนหน้าจอ, ค่าที่รับเข้ามาจากแผงแป้นอักขระและเมาส์ และฟังก์ชันต่างๆที่เกี่ยวข้องกับ GUI (Graphical User Interface) ของวินโดวส์การทำงานเหล่านี้จะอยู่ถูกควบคุมโดยแฟ้ม user32.dll แต่ใน Windows XP การควบคุมพื้นฐานจะอยู่ภายใต้การควบคุมจากแฟ้ม comctl32.dll กับ Common Control Library

#### 2.6.1.4. Common Dialog Box Library

เป็นส่วนที่เตรียมตัวการติดต่อกับกล่องข้อความมาตรฐานต่างๆ เช่น หน้าต่างเปิดแฟ้ม และบันทึกแฟ้ม, หน้าต่างเลือกสี, หน้าต่างเลือกชนิดตัวอักษรที่ใช้ เป็นต้น การทำงานเหล่านี้จะอยู่ถูกควบคุมโดยแฟ้ม comdlg32.dll ซึ่งจะทำงานร่วมกับ User Interface API

#### 2.6.1.5. Common Control Library

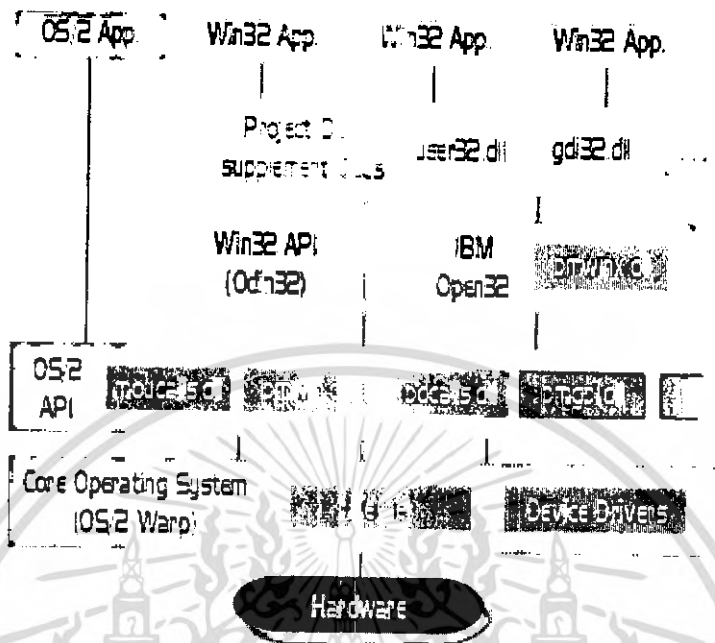
เป็นส่วนที่เตรียมตัวการติดต่อกับการควบคุมขั้นสูง ที่ถูกจัดเตรียมไว้โดยระบบปฏิบัติการ เช่นแถบสถานะ, แถบแสดงความก้าวหน้า, แถบเครื่องมือ และ จุดตั้งระยะ เป็นต้น การทำงานเหล่านี้จะอยู่ถูกควบคุมโดยแฟ้ม comctl32.dll ซึ่งจะทำงานร่วมกับ User Interface API

#### 2.6.1.6. Windows Shell

เป็นส่วนที่ช่วยให้โปรแกรมสามารถติดต่อกับฟังก์ชัน shell ของระบบปฏิบัติการได้ รวมถึงการเปลี่ยนแปลงและเพิ่มประสิทธิภาพการทำงาน การทำงานเหล่านี้จะอยู่ถูกควบคุมโดยแฟ้ม shlwapi.dll และ shell32.dll ซึ่งจะทำงานร่วมกับ User Interface API

#### 2.6.1.7. Network Services

เป็นส่วนช่วยในการทำงานกับระบบเน็ตเวิร์กของระบบปฏิบัติการ ให้สามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งภายใน API จะประกอบไปด้วยการติดต่อกับ โพรโทคอล ที่สำคัญๆ เช่น NetBIOS, Winsock, NetDDE, RPC เป็นต้น



รูปที่ 2.15 โครงสร้างของ Win32 API

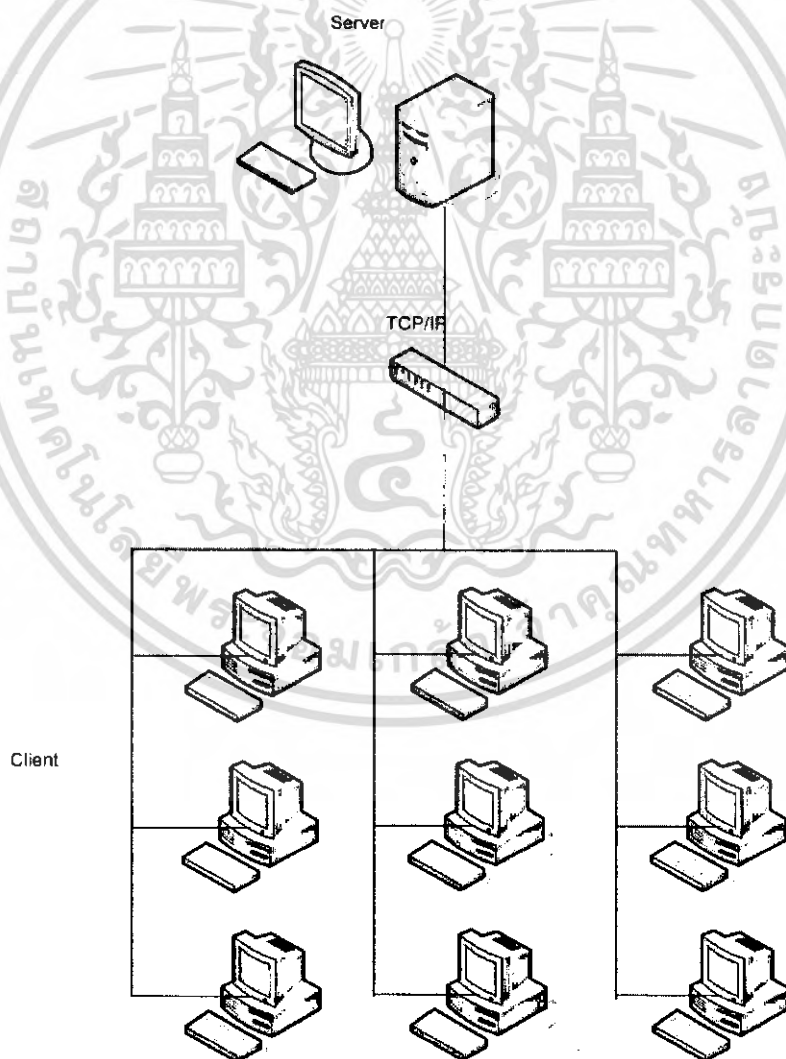
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

## การออกแบบโปรแกรม

### 3.1 การออกแบบโปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย

การออกแบบโปรแกรมประกอบด้วย 2 ส่วน คือ ส่วนที่ทำหน้าที่เป็นแม่ข่าย และส่วนที่ทำหน้าที่เป็นลูกข่าย โดยเครื่องแม่ข่ายและเครื่องลูกข่ายสามารถติดต่อกันได้โดยผ่านโพรโทคอล TCP/IP (Transmission Control Protocol/Internet Protocol) ซึ่งเครื่องลูกข่ายต้องทราบถึงเลขที่อยู่ไอพี (IP address) ของเครื่องแม่ข่ายเพื่อใช้ในการอ้างอิงการติดต่อ

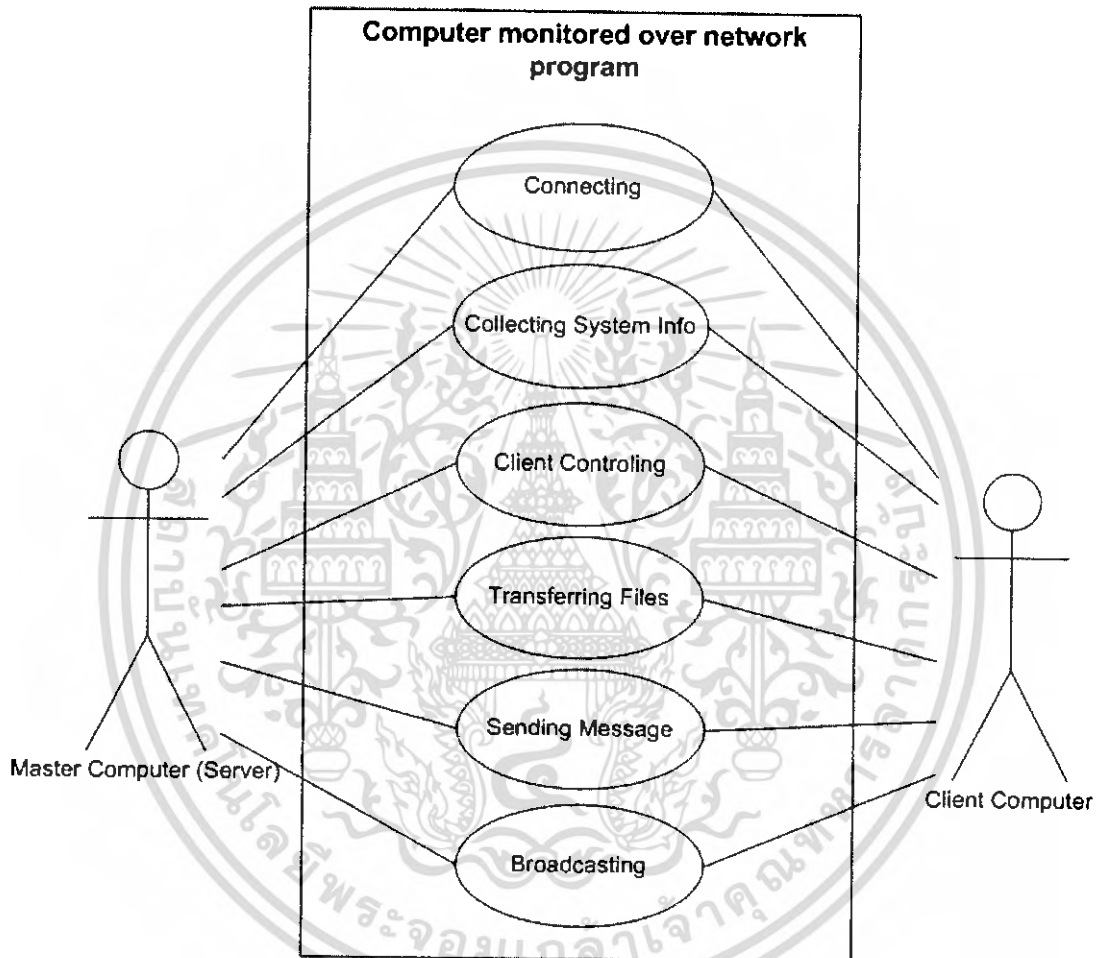


รูปที่ 3.1 แสดงการทำงานของระหว่างเครื่องลูกข่ายและแม่ข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 Use Case Diagram

เป็นแผนภาพที่แสดงถึงพฤติกรรมหรือภาพรวมของระบบ ในที่นี้จะมีเครื่องแม่ข่าย และ เครื่องลูกข่าย เป็นผู้ที่มีปฏิสัมพันธ์(actor) กับระบบ

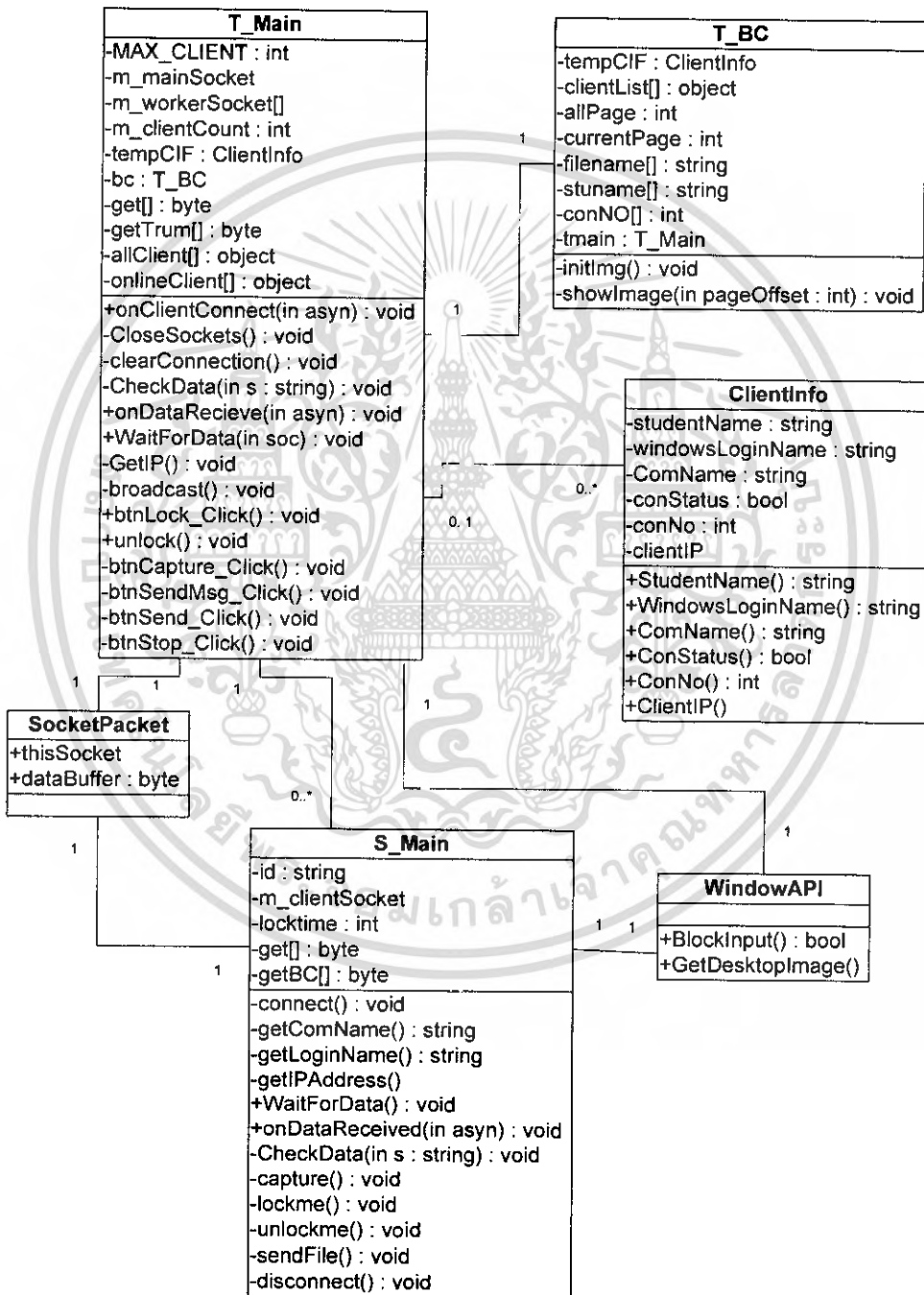


รูปที่ 3.2 แสดงแผนภาพ use case diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 Class Diagram

เป็นแผนภาพที่แสดงความสัมพันธ์ของระหว่างคลาส และแสดงลักษณะประจำ(attribute) และเมทอด(method) ที่สำคัญ

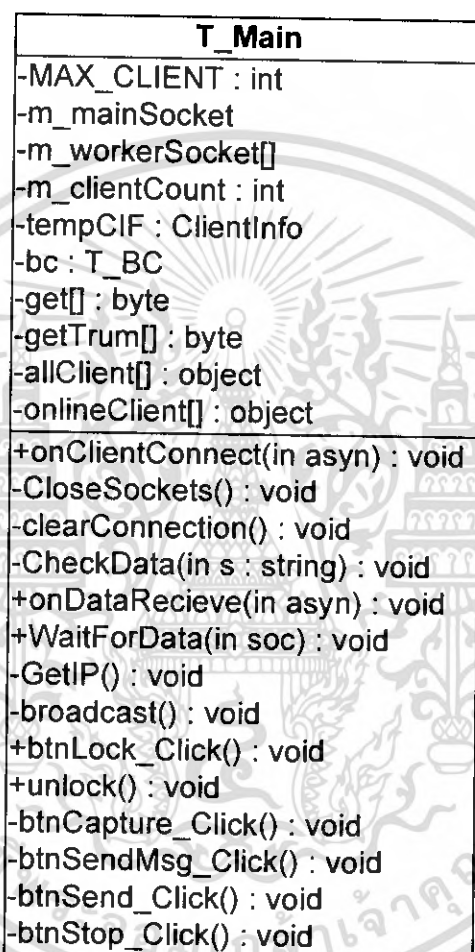


รูปที่ 3.3 แสดงแผนภาพความสัมพันธ์ของคลาสทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.1 Class T\_Main

ทำหน้าที่เป็นคลาสหลักของแม่ข่าย ควบคุมการทำงานหลักต่างๆ ทั้งหมดของแม่ข่าย



รูปที่ 3.4 แสดงลักษณะประจำและเมทอดของ Class T\_Main

ตารางที่ 3.1 แสดงความหมายเมทอดของ Class T\_Main

เมทอด	ความหมาย
onClientConnect(IAsyncResult asyn)	ทำการเชื่อมต่อระหว่างเครื่องแม่ข่ายและเครื่องลูกข่ายเมื่อมีการติดต่อเข้ามา
CloseSockets()	ทำการปิดการเชื่อมต่อของซ็อกเก็ต
clearConnection()	ทำการตัดการเชื่อมต่อกับเครื่องลูกข่ายและปรับสถานะที่แสดงบนหน้าจอเครื่องแม่ข่าย
CheckData(String s)	ทำการตรวจสอบข้อมูลกลุ่มข้อมูล(packet) ที่เข้ามาว่าเป็นประเภทอะไร
onDataRecieve(IAsyncResult asyn)	ทำการรับข้อมูลเข้ามาแล้วตรวจสอบข้อมูล จะทำงานร่วมกับเมทอด CheckData()
WaitForData(Socket soc)	ทำการรอข้อมูลเข้ามา เปิดกลุ่มข้อมูลก่อนจะส่งไปให้เมทอด onDataRecieve() ต่อไป
GetIP()	ทำการค้นหาและแสดงเลขที่อยู่ไอพีของเครื่องแม่ข่าย
broadcast()	ทำการแพร่ภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่ายที่ถูกเลือก
btnLock_Click ()	ทำการล็อกแ่งเป็นอักขระและเมาส์ของเครื่องลูกข่ายที่ถูกเลือก
unlock()	ทำการปลดล็อกเครื่องลูกข่ายที่ถูกล็อกอยู่
btnCapture_Click()	ทำการจับภาพหน้าจอของเครื่องลูกข่ายที่ถูกเลือกส่งมาแสดงผลยังเครื่องแม่ข่าย
btnSendMsg_Click()	ทำการส่งข้อความจากเครื่องแม่ข่ายไปยังเครื่องลูกข่ายที่ถูกเลือก
btnSend_Click()	ทำการส่งเพิ่มข้อมูลที่ต้องการจากเครื่องแม่ข่ายไปยังเครื่องลูกข่ายที่ถูกเลือก
btnStop_Click()	ทำการหยุดบริการของแม่ข่าย เครื่องลูกข่ายทั้งหมดจะถูกตัดการเชื่อมต่อออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 Class ClientInfo

ทำหน้าที่เก็บข้อมูลของเครื่องลูกข่ายแต่ละเครื่องที่เข้ามาติดต่อกับแม่ข่าย

<b>ClientInfo</b>
-studentName : string
-windowsLoginName : string
-ComName : string
-conStatus : bool
-conNo : int
-clientIP
+StudentName() : string
+WindowsLoginName() : string
+ComName() : string
+ConStatus() : bool
+ConNo() : int
+ClientIP()

รูปที่ 3.5 แสดงลักษณะประจำและเมทอดของ Class ClientInfo

ตารางที่ 3.2 แสดงความหมายเมทอดของ Class ClientInfo

เมทอด	ความหมาย
StudentName()	ทำการรับและตั้งค่าให้กับตัวแปร studentName
WindowsLoginName()	ทำการรับและตั้งค่าให้กับตัวแปร windowsLoginName
ComName()	ทำการรับและตั้งค่าให้กับตัวแปร comName
ConStatus()	ทำการรับและตั้งค่าให้กับตัวแปร conStatus
ConNo()	ทำการรับและตั้งค่าให้กับตัวแปร conNo
ClientIP()	ทำการรับและตั้งค่าให้กับตัวแปร clientIP

### 3.3.3 Class T\_BC

ทำหน้าที่แสดงข้อมูลหน้าจอของเครื่องลูกข่าย ซึ่งจะสามารถแสดงหน้าจอของหลายๆ เครื่องได้พร้อมกัน

T_BC
-tempCIF : ClientInfo
-clientList[] : object
-allPage : int
-currentPage : int
-filename[] : string
-stuname[] : string
-conNO[] : int
-tmain : T_Main
-initlmg() : void
-showlmg(in pageOffset : int) : void

รูปที่ 3.6 แสดงลักษณะประจำและเมทอดของ Class T\_BC

ตารางที่ 3.3 แสดงความหมายเมทอดของ Class T\_BC

เมทอด	ความหมาย
initlmg()	เตรียมค่าการแสดงผล เช่น ภาพหน้าจอและชื่อเครื่องลูกข่ายที่ถูกเลือกให้แสดง
showlmg(int pageOffset)	แสดงรูปหน้าจอของเครื่องลูกข่ายและจัดการเกี่ยวกับการเปลี่ยนหน้าการแสดงผล

### 3.3.4 Class S\_Main

ทำหน้าที่เป็นคลาสหลักของเครื่องลูกข่าย คอยติดต่อกับแม่ข่าย และทำงานหลักทั้งหมดของเครื่องฝั่งลูกข่าย

<b>S_Main</b>
-id : string
-m_clientSocket
-locktime : int
-get[] : byte
-getBC[] : byte
-connect() : void
-getComName() : string
-getLoginName() : string
-getIPAddress()
+WaitForData() : void
+onDataReceived(in asyn) : void
-CheckData(in s : string) : void
-capture() : void
-lockme() : void
-unlockme() : void
-sendFile() : void
-disconnect() : void

รูปที่ 3.7 แสดงลักษณะประจำและเมทอดของ Class S\_Main

ตารางที่ 3.4 แสดงความหมายเมทอดของ Class S\_Main

เมทอด	ความหมาย
Connect()	ทำการเชื่อมต่อไปยังเครื่องแม่ข่าย
getComName()	ทำการดึงข้อมูลชื่อเครื่องคอมพิวเตอร์จากข้อมูลในระบบปฏิบัติการ
getLoginName()	ทำการดึงข้อมูลชื่อผู้ใช้นั้นจากข้อมูลในระบบปฏิบัติการ
getIPAddress()	ทำการดึงข้อมูลเลขที่อยู่ไอพีจากข้อมูลในระบบปฏิบัติการ
WaitForData()	ทำการรอข้อมูลเข้ามา เปิดช็อกเก็ตก่อนจะส่งไปให้เมทอด onDataRecieve() ต่อไป
onDataRecieve(IAsyncResult asyn)	ทำการรับข้อมูลเข้ามาแล้วตรวจสอบข้อมูล จะทำงานร่วมกับเมทอด CheckData()
CheckData(string s)	ทำการตรวจสอบข้อมูลกลุ่มข้อมูลที่เข้ามาว่าเป็นประเภทอะไร
capture()	ทำจับภาพหน้าจอสำหรับส่งไปให้เครื่องแม่ข่าย
lockme()	ทำการล็อกแฉงเป็นอักขระและเมาส์ของเครื่องลูกข่าย
unlockme()	ทำการปลดล็อกแฉงเป็นอักขระและเมาส์ของเครื่องลูกข่าย
sendFile()	ทำการส่งแฟ้มข้อมูลจากเครื่องลูกข่ายไปยังเครื่องแม่ข่าย เช่น แฟ้มข้อมูลการบ้าน เป็นต้น
disconnect()	ยกเลิกการเชื่อมต่อกับแม่ข่ายและปิดช็อกเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5 Class SocketPacket

เป็นคลาสที่สร้างซ็อกเก็ต และระบุขนาดของซ็อกเก็ต

SocketPacket
+thisSocket
+dataBuffer : byte

รูปที่ 3.8 แสดงลักษณะประจำของ Class SocketPacket

### 3.3.6 Class WindowAPI

ทำหน้าที่ติดต่อกับ Win32 API สำหรับการทำงานของฟังก์ชันการล็อกแฉงแป้นอักขระและเมาส์ และฟังก์ชันการจับภาพหน้าจอ

WindowAPI
+BlockInput() : bool
+GetDesktopImage()

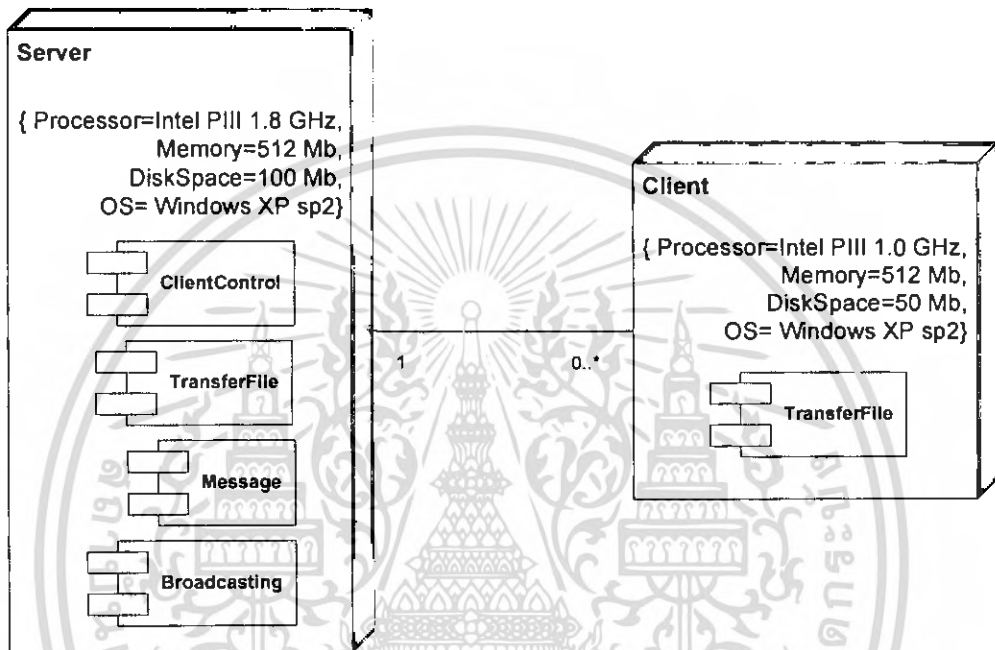
รูปที่ 3.9 แสดงลักษณะประจำและเมทอดของ Class WindowAPI

ตารางที่ 3.5 แสดงความหมายเมทอดของ Class WindowAPI

เมทอด	ความหมาย
BlockInput()	ทำการติดต่อกับระบบเพิ่มข้อมูลที่เกี่ยวข้องกับการทำงานของแป้นอักขระและเมาส์
GetDesktopImage()	ทำการติดต่อกับระบบเพิ่มข้อมูลที่เกี่ยวข้องกับการทำงานของการแสดงผลบนหน้าจอ

### 3.4 Deployment Diagram

เป็นแผนภาพที่แสดงความสัมพันธ์ระหว่างฮาร์ดแวร์ หรือจำลองสภาพแวดล้อมการทำงานจริงของโปรแกรม



รูปที่ 3.10 แสดงแผนภาพ Deployment diagram

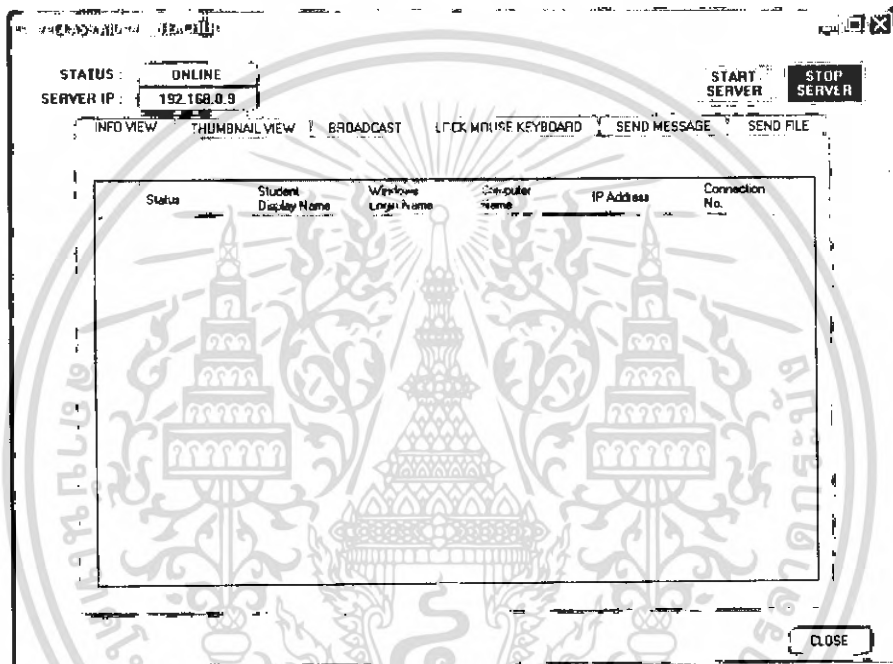
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทำงานของโปรแกรม

#### 4.1 โปรแกรมในเครื่องด้านแม่ข่าย

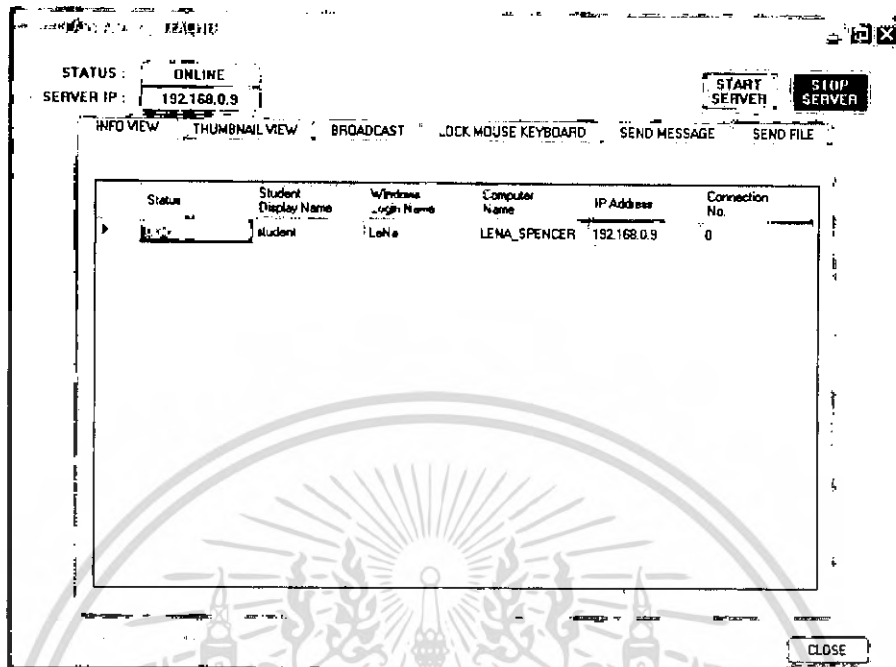
เมื่อเปิดโปรแกรมในเครื่องด้านแม่ข่าย แล้วหน้าต่างจะแสดงผล ดังรูปที่ 4.1



รูปที่ 4.1 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายขณะรอรับการติดต่อจากเครื่องลูกข่าย

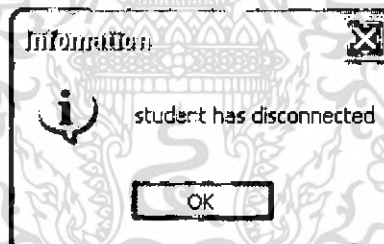
- เมื่อกดปุ่ม [START SERVER] แล้วเครื่องแม่ข่ายจะเปลี่ยนสถานะภาพ (STATUS) เป็นออนไลน์ (ONLINE) ที่สามารถรอรับการติดต่อจากเครื่องลูกข่ายได้
- เมื่อกดปุ่ม [STOP SERVER] แล้วเครื่องแม่ข่ายจะเปลี่ยนสถานะภาพเป็นออฟไลน์ (OFFLINE) ซึ่งเครื่องลูกข่ายจะไม่สามารถติดต่อเข้ามายังเครื่องแม่ข่าย
- กดปุ่ม [CLOSE] เพื่อปิดโปรแกรม
- เมื่อมีเครื่องลูกข่ายติดต่อเข้ามายังเครื่องด้านแม่ข่าย แล้วหน้าต่างจะแสดงผล ดังรูปที่ 4.2
- หากเครื่องลูกข่ายหลุดจากการติดต่อ จะปรากฏหน้าต่าง ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายเมื่อมีเครื่องลูกข่ายติดต่อเข้ามา แสดงใน tab

INFO VIEW



รูปที่ 4.3 หน้าต่างที่จะแสดงเมื่อเครื่องลูกข่ายหลุดจากการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.1 การดูข้อมูลเครื่องลูกข่าย

เมื่อเลือก Tab [INFO VIEW] ซึ่งเป็นหน้าโดยปริยาย(default) ที่จะแสดงเมื่อเปิดโปรแกรม ดังรูปที่ 4.2 ซึ่งเป็นส่วนที่ไว้ดูข้อมูลต่างๆ ของเครื่องลูกข่ายที่ติดต่อ ประกอบด้วยส่วนต่างๆ ดังนี้

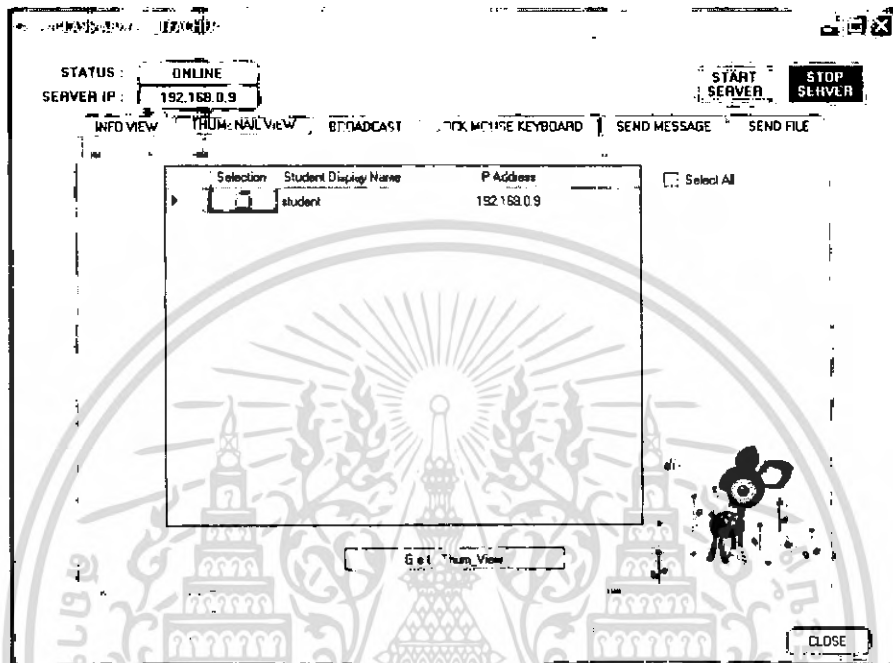
1. Status คือ สถานะปัจจุบันของเครื่องลูกข่ายที่ติดต่อเข้ามาว่าออนไลน์ หรือออฟไลน์อยู่ ซึ่งสถานะออฟไลน์นั้นจะเกิดขึ้นได้ก็ต่อเมื่อต้องมีการติดต่อเข้ามาแล้ว แต่ภายหลังจากนั้นได้ตัดการเชื่อมต่อออกไป
2. Student Display Name คือ ชื่อของนักเรียนจากเครื่องลูกข่ายที่ติดต่อ
3. Windows Login Name คือ ชื่อผู้ใช้ที่ลงบันทึกเข้า(login) กับวินโดวส์ (Windows) ในขณะนั้น
4. Computer Name คือ ชื่อเครื่องคอมพิวเตอร์ของเครื่องลูกข่ายที่ติดต่อ
5. IP Address คือ เลขที่อยู่ไอพีของเครื่องลูกข่ายที่ติดต่อ
6. Connection No. คือ หมายเลขการติดต่อ ซึ่งทางด้านแม่ข่ายจะเป็นคนสร้างให้

#### 4.1.2 การเรียกดูภาพหน้าจอของเครื่องลูกข่าย

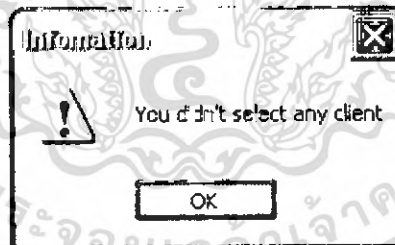
เมื่อเลือก Tab [THUMBNAIL VIEW] หน้าต่างจะแสดงผลดังรูปที่ 4.4 ซึ่งเป็นส่วนที่สามารถเลือกดูภาพหน้าจอในขณะนั้นของเครื่องลูกข่ายที่ติดต่อ ประกอบด้วยส่วนต่างๆ ดังนี้

1. Selection คือ ส่วนที่ให้ผู้ใช้งานสามารถเลือกได้ว่าจะดูภาพหน้าจอของเครื่องลูกข่ายใดบ้าง ซึ่งมี Check box [Select All] เพื่อความสะดวกในการเลือกเครื่องลูกข่ายทั้งหมด
2. หากไม่มีการเลือกหรือคลิกในแนว check box [selection] เครื่องลูกข่าย ในการจัดการกับเครื่องลูกข่ายนั้นจะปรากฏหน้าต่าง ดังรูปที่ 4.5
3. Student Display Name คือ ชื่อของนักเรียนจากเครื่องลูกข่ายที่ติดต่อ
4. IP Address คือ เลขที่อยู่ไอพีของเครื่องลูกข่ายที่ติดต่อ
5. ปุ่ม [Get Thum\_View] เมื่อคลิกแล้ว จะแสดงหน้าต่าง Thumbnail View ดังรูปที่ 4.6 ซึ่งเป็นหน้าต่างที่แสดงภาพหน้าจอและชื่อนักเรียนของเครื่องลูกข่าย ได้ทีละ 6 เครื่อง มีปุ่ม next เพื่อดูภาพหน้าจอของเครื่องลูกข่าย 6 เครื่องถัดไป และปุ่ม prev เพื่อย้อนกลับมาดูภาพหน้าจอของเครื่องลูกข่าย 6 เครื่องที่แล้ว

และปุ่ม [Refresh] เมื่อคลิกแล้วจะจับภาพหน้าจอและแสดงภาพหน้าจอของเครื่องลูกข่ายอีกครั้ง

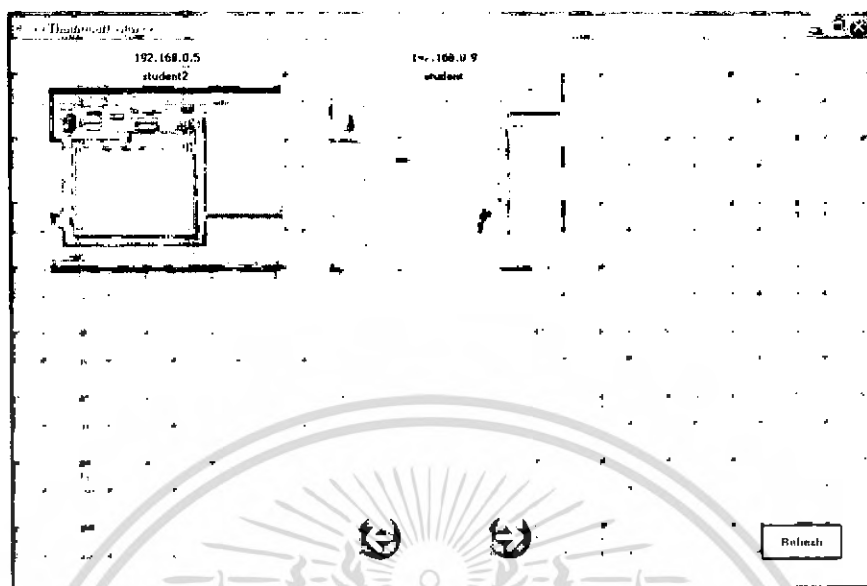


รูปที่ 4.4 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab THUMBNAIL VIEW



รูปที่ 4.5 หน้าต่างที่จะแสดงเมื่อไม่มีการเลือกเครื่องลูกข่ายในการจัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



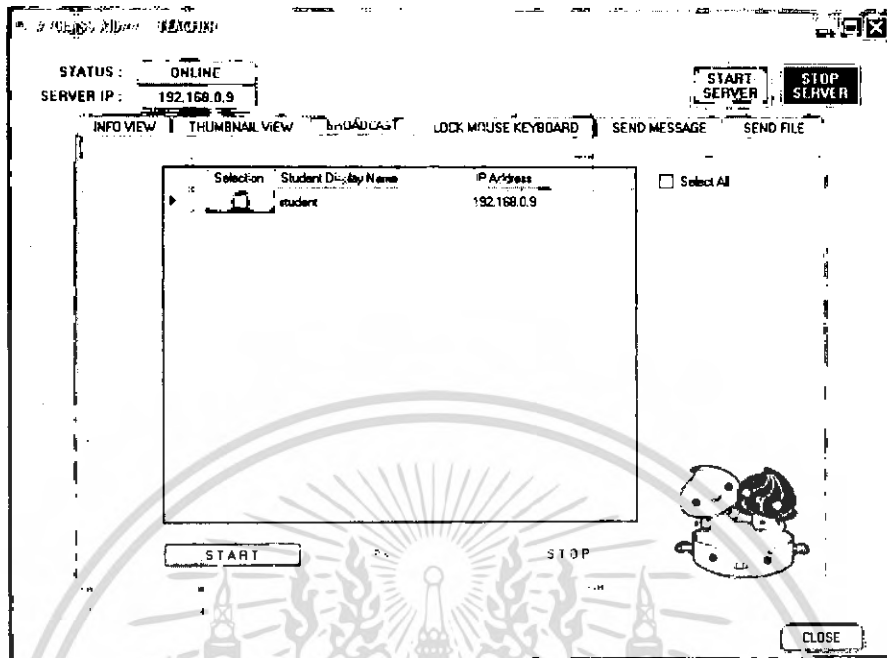
รูปที่ 4.6 หน้าต่าง Thumbnail View

#### 4.1.3 การส่งภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่าย

เมื่อเลือก Tab [BROADCAST] หน้าต่างจะแสดงผลดังรูปที่ 4.7 ซึ่งเป็นส่วนที่ไว้ให้ผู้ใช้แพร่ภาพหน้าจอ(broadcast) ของเครื่องแม่ข่าย ไปยังเครื่องลูกข่ายที่ติดต่อกันที่ถูกลเลือก ประกอบด้วยส่วนต่างๆ ดังนี้

1. Selection คือ ส่วนที่ให้ผู้ใช้งานสามารถเลือกได้ว่า จะส่งภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่ายใดบ้าง ซึ่งมี Check box [Select All] เพื่อความสะดวกในการเลือกเครื่องลูกข่ายทั้งหมด
2. หากไม่มีการเลือกหรือคลิกในแนว check box [selection] เครื่องลูกข่าย ในการจัดการกับเครื่องลูกข่ายนั้นจะปรากฏหน้าต่าง ดังรูปที่ 4.5
3. Student Display Name คือ ชื่อของนักเรียนจากเครื่องลูกข่ายที่ติดต่อกัน
4. IP Address คือ เลขที่อยู่ไอพีของเครื่องลูกข่ายที่ติดต่อกัน
5. ปุ่ม [START] เมื่อคลิกแล้ว จะเริ่มส่งภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่าย
6. ปุ่ม [PAUSE] เมื่อคลิกแล้ว จะหยุดส่งภาพหน้าจอของเครื่องแม่ข่าย ซึ่งเครื่องลูกข่ายจะแสดงค้างอยู่ในภาพสุดท้าย
7. ปุ่ม [STOP] เมื่อคลิกแล้ว จะจบการส่งภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab BROADCAST

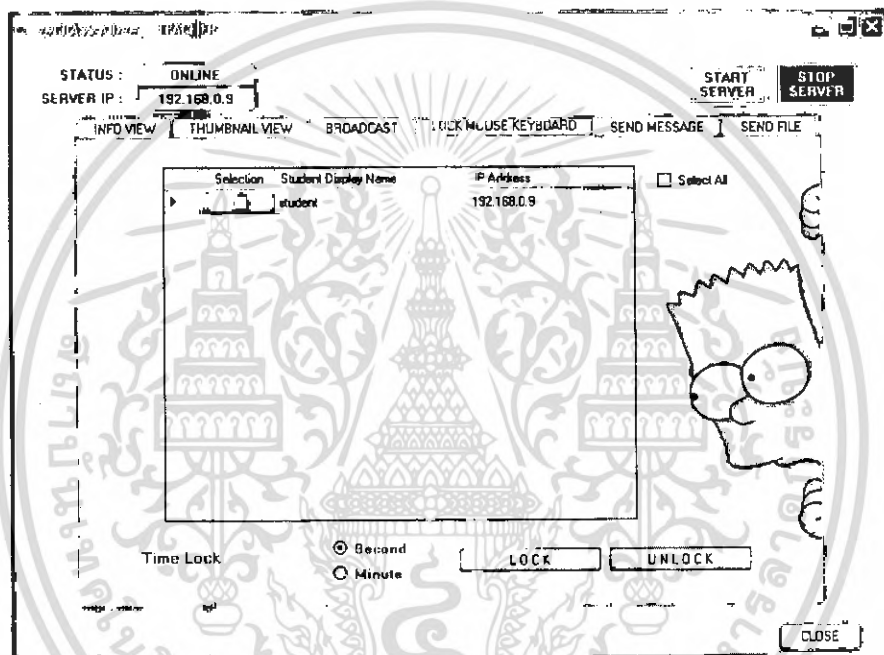
#### 4.1.4 การล็อกเมาส์และแผงแป้นอักขระของเครื่องลูกข่าย

เมื่อเลือก Tab [LOCK MOUSE KEYBOARD] หน้าต่างจะแสดงผลดังรูปที่ 4.8 ซึ่งเป็นส่วนที่ไว้ล็อกเมาส์และแผงแป้นอักขระของเครื่องลูกข่ายที่ติดต่อที่ถูกเลือก ประกอบด้วยส่วนต่างๆ ดังนี้

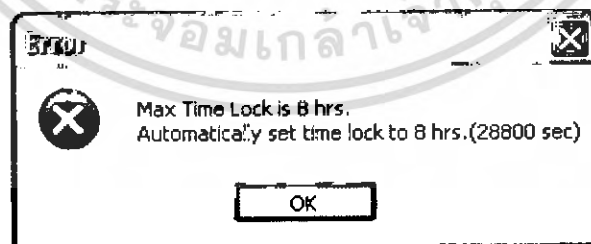
1. Selection คือ ส่วนที่ให้ผู้ใช้งานสามารถเลือกได้ว่าจะส่งภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่ายใดบ้าง ซึ่งมี Check box [Select All] เพื่อความสะดวกในการเลือกเครื่องลูกข่ายทั้งหมด
2. หากไม่มีการเลือกหรือคลิกในแนว check box [selection] เครื่องลูกข่าย ในการจัดการกับเครื่องลูกข่ายนั้นจะปรากฏหน้าต่าง ดังรูปที่ 4.5
3. Student Display Name คือ ชื่อของนักเรียนจากเครื่องลูกข่ายที่ติดต่อ
4. IP Address คือ เลขที่อยู่ไอพีของเครื่องลูกข่ายที่ติดต่อ
5. Time Lock คือ ผู้ใช้สามารถใส่เวลาที่ จะทำการล็อก ที่เป็นวินาที หรือ นาที แต่หากไม่ได้ใส่ค่าลงไปจะเป็นการล็อกจนกว่าจะได้รับการปลดล็อก หรือจะปลดล็อกเองหากครบ 8 ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หากใส่ค่า Time Lock มากกว่า 8 ชั่วโมง หรือ 28800 วินาที หรือ 480 นาที จะปรากฏหน้าต่างดังรูปที่ 4.9 ซึ่งจะกำหนดค่า Time Lock อัตโนมัติเป็น 8 ชั่วโมง
7. ปุ่ม [LOCK] เมื่อคลิกแล้ว จะส่งคำสั่งให้เครื่องลูกข่ายล็อกเมาส์และแผงแป้นอักขระตามเงื่อนไขของเวลาที่ส่งไป
8. ปุ่ม [UNLOCK] เมื่อคลิกแล้ว จะส่งคำสั่งให้เครื่องลูกข่ายปลดล็อกเมาส์และแผงแป้นอักขระ



รูปที่ 4.8 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab LOCK MOUSE KEYBOARD



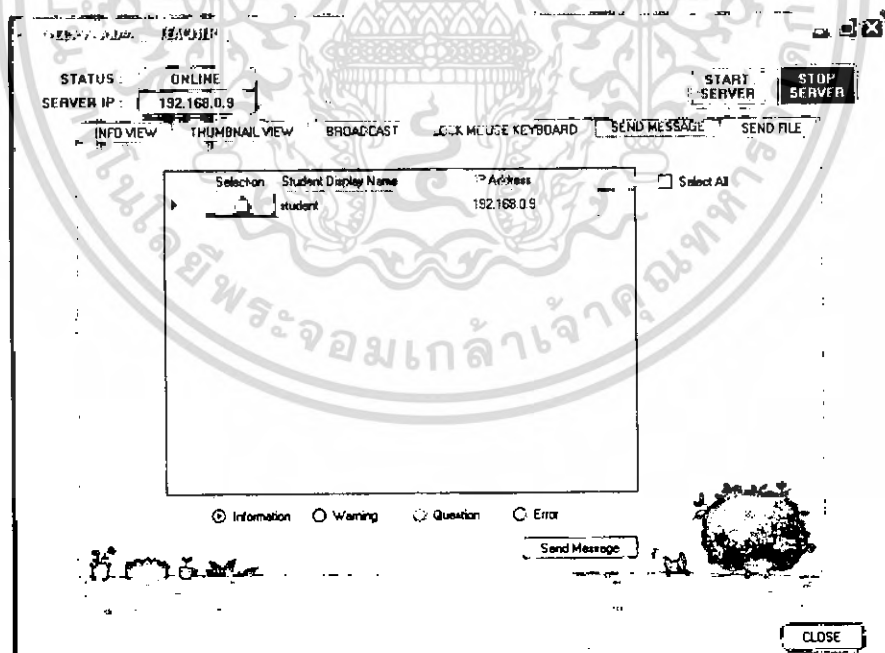
รูปที่ 4.9 หน้าต่างเมื่อใส่ค่า Time Lock เกิน 8 ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.5 การส่งข้อความไปยังเครื่องลูกข่าย

เมื่อเลือก Tab [SEND MESSAGE] หน้าต่างจะแสดงผลดังรูปที่ 4.10 ซึ่งเป็นส่วนที่ไว้ส่งข้อความเตือนไปยังเครื่องลูกข่ายที่ติดต่อที่ถูกเลือก ประกอบด้วยส่วนต่างๆ ดังนี้

1. Selection คือ ส่วนที่ให้ผู้ใช้งานสามารถเลือกได้ว่าจะส่งภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่ายใดบ้าง ซึ่งมี Check box [Select All] เพื่อความสะดวกในการเลือกเครื่องลูกข่ายทั้งหมด
2. หากไม่มีการเลือกหรือคลิกในแนว check box [selection] เครื่องลูกข่าย ในการจัดการกับเครื่องลูกข่ายนั้นจะปรากฏหน้าต่าง ดังรูปที่ 4.5
3. Student Display Name คือ ชื่อของนักเรียนจากเครื่องลูกข่ายที่ติดต่อ
4. IP Address คือ เลขที่อยู่ไอพีของเครื่องลูกข่ายที่ติดต่อ
5. Radio Buttons คือ ปุ่มที่ให้ผู้ใช้งานเลือกว่าจะส่งข้อความเตือนเป็นชนิดใด (Information, Warning, Question และ Error)
6. Text Box คือ ช่องที่ให้ผู้ใช้งานใส่ข้อความที่จะแสดงทางฝั่งเครื่องลูกข่าย
7. ปุ่ม [Send Message] เมื่อคลิกแล้ว จะส่งข้อความไปยังเครื่องลูกข่าย

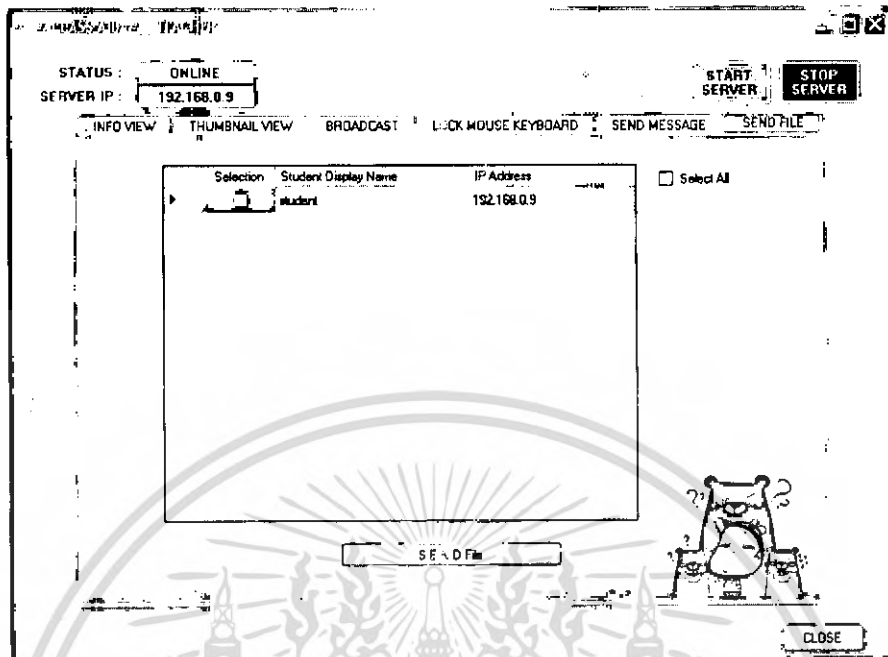


รูปที่ 4.10 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab SEND MESSAGE

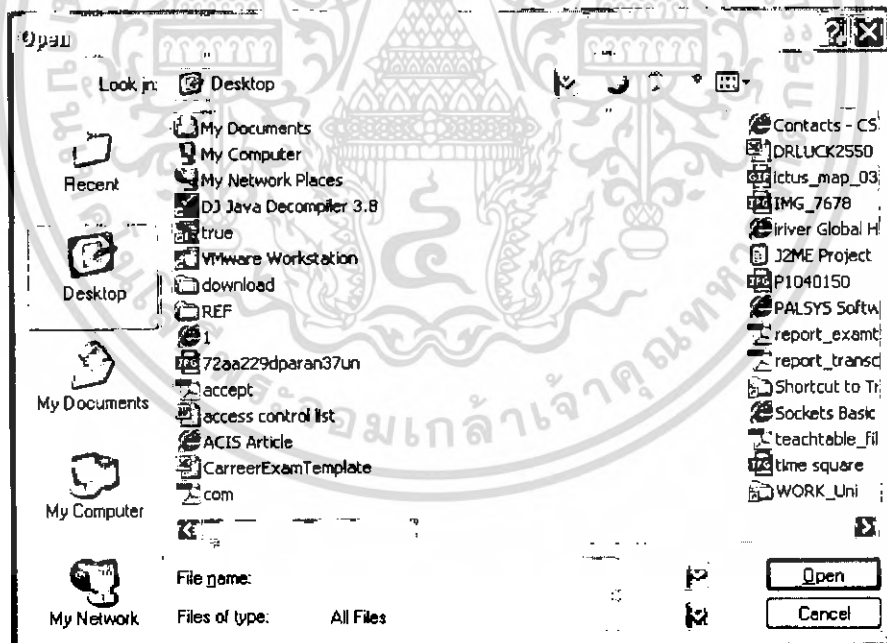
#### 4.1.6 การรับ-ส่งแฟ้มข้อมูลไปยังเครื่องลูกข่าย

เมื่อเลือก Tab [SEND FILE] หน้าต่างจะแสดงผลดังรูปที่ 4.11 ซึ่งเป็นส่วนที่ไว้ส่งแฟ้มข้อมูลต่างๆ เช่น แฟ้มข้อความ, แฟ้มรูปภาพ, แฟ้มเสียง, แฟ้มวีดิทัศน์ หรือแฟ้มที่ถูกบีบอัดขนาด เป็นต้น ไปยังเครื่องลูกข่ายที่ติดต่อที่ถูกเลือก ประกอบด้วยส่วนต่างๆ ดังนี้

1. Selection คือ ส่วนที่ให้ผู้ใช้งานสามารถเลือกได้ว่าจะส่งภาพหน้าจอของเครื่องแม่ข่ายไปยังเครื่องลูกข่ายใดบ้าง ซึ่งมี Check box [Select All] เพื่อความสะดวกในการเลือกเครื่องลูกข่ายทั้งหมด
2. หากไม่มีการเลือกหรือคลิกในแนว check box [selection] เครื่องลูกข่าย ในการจัดการกับเครื่องลูกข่ายนั้นจะปรากฏหน้าต่าง ดังรูปที่ 4.5
3. Student Display Name คือ ชื่อของนักเรียนจากเครื่องลูกข่ายที่ติดต่อ
4. IP Address คือ เลขที่อยู่ไอพีของเครื่องลูกข่ายที่ติดต่อ
5. ปุ่ม [SEND File] เมื่อคลิกแล้ว จะปรากฏหน้าต่าง OpenFileDialog ดังรูปที่ 4.12 เพื่อเลือกแฟ้มข้อมูลที่ต้องการส่ง เมื่อเลือกได้แล้วจะทำการส่งแฟ้มข้อมูลนั้นไปยังเครื่องลูกข่าย ณ สารบบ(directory) ../File/ ของทางฝั่งเครื่องลูกข่าย
6. ในการรับแฟ้มข้อมูล เมื่อได้รับแฟ้มข้อมูลเรียบร้อยแล้ว จะปรากฏหน้าต่าง ดังรูปที่ 4.13 ซึ่งจะบอกชื่อแฟ้มข้อมูลที่รับด้วย

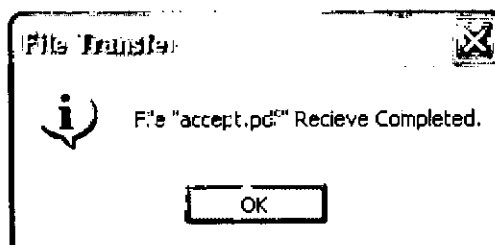


รูปที่ 4.11 หน้าต่างโปรแกรมบนเครื่องแม่ข่ายใน tab SEND FILE



รูปที่ 4.12 หน้าต่าง OpenFileDialog

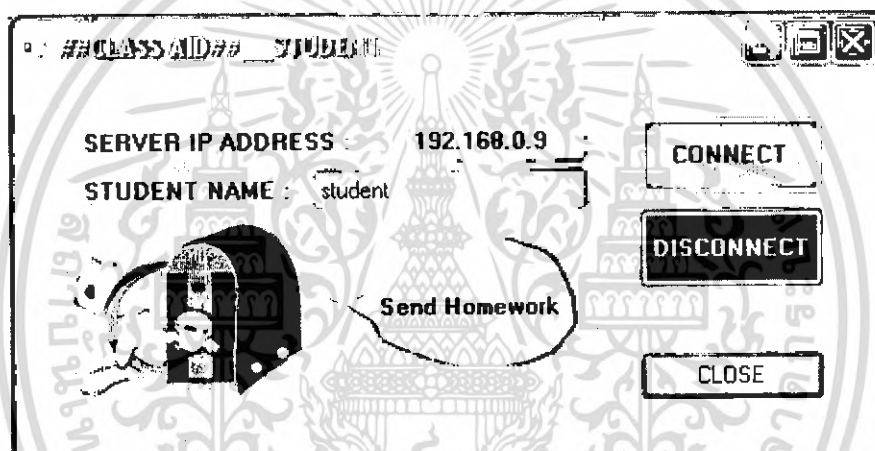
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 หน้าต่างเมื่อได้รับเพิ่มข้อมูลเรียบร้อยแล้ว

## 4.2 โปรแกรมในเครื่องด้านลูกข่าย

เมื่อเปิดโปรแกรมในเครื่องด้านลูกข่าย แล้วหน้าต่างจะแสดงผล ดังรูปที่ 4.14



รูปที่ 4.14 หน้าต่างโปรแกรมบนเครื่องลูกข่าย

- SERVER IP ADDRESS Text box ไว้ให้ผู้ใช้อกรอกเลขที่อยู่ไอพีของเครื่องแม่ข่ายที่ต้องการติดต่อ
- STUDENT NAME Text box ไว้ให้ผู้ใช้อกรอกชื่อของนักเรียนที่เป็นผู้ใช้
- ปุ่ม [CONNECT] เพื่อเชื่อมต่อกับเครื่องแม่ข่ายที่มีเลขที่อยู่ไอพี ดังใน SERVER IP ADDRESS และส่งชื่อนักเรียนไป ดังใน STUDENT NAME Text box
  - หากใส่เลขที่อยู่ไอพีของเครื่องแม่ข่ายผิด จะปรากฏหน้าต่าง ดังรูปที่ 4.15



รูปที่ 4.15 หน้าต่างเมื่อใส่เลขที่อยู่ไอพีของเครื่องแม่ข่ายผิด

- กดปุ่ม [DISCONNECT] เพื่อยกเลิกการเชื่อมต่อกับเครื่องแม่ข่าย และจะปรากฏหน้าต่างดังรูปที่ 4.16



รูปที่ 4.16 หน้าต่างเมื่อกดปุ่ม DISCONNECT

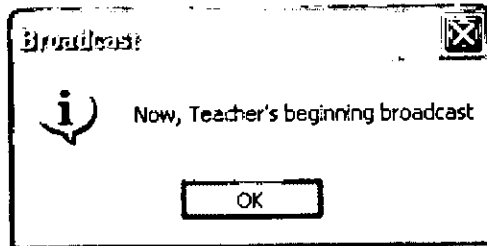
- เมื่อแม่ข่ายปิดการติดต่อ หรือหลุดจากการติดต่อ จะปรากฏหน้าต่างดังรูปที่ 4.17



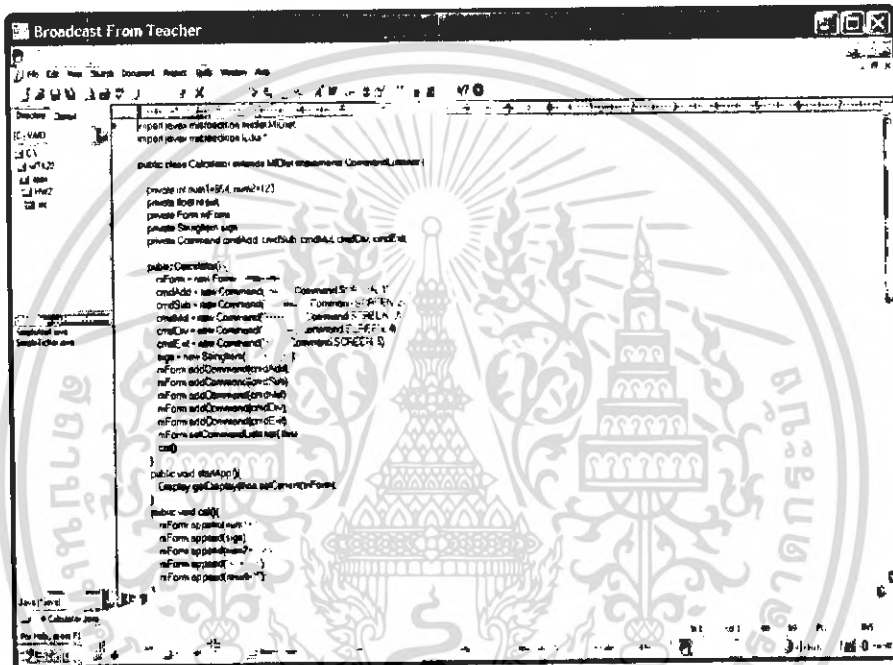
รูปที่ 4.17 หน้าต่างเมื่อหลุดการติดต่อจากเครื่องแม่ข่าย

- เมื่อทางฝั่งแม่ข่ายได้ทำการแพร์ภาพหน้าจอของตนมายังเครื่องลูกข่าย ทางฝั่งลูกข่ายจะปรากฏหน้าต่าง Message Box ว่าผู้สอนเริ่มทำการแพร์ภาพหน้าจอแล้ว ดังรูปที่ 4.18 เมื่อเครื่องลูกข่ายกด OK จะปรากฏหน้าต่าง ดังรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

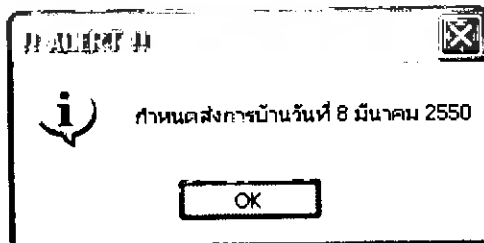


รูปที่ 4.18 Message Box แสดงถึงผู้สอนเริ่มทำการแพร่ภาพหน้าจอแล้ว



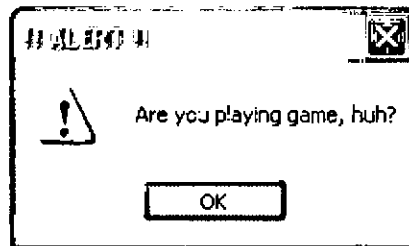
รูปที่ 4.19 หน้าต่างที่แสดงหน้าจอของเครื่องแม่ข่าย

- เมื่อทางฝั่งแม่ข่ายได้ทำการส่งข้อความเตือนมาให้เครื่องลูกข่าย ทางฝั่งลูกข่ายจะปรากฏหน้าต่างดังรูปที่ 4.20 – 4.23

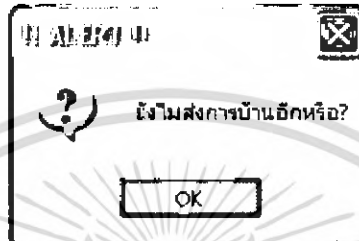


รูปที่ 4.20 หน้าต่างแสดงข้อความจากเครื่องแม่ข่ายชนิด Information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 หน้าต่างแสดงข้อความจากเครื่องแม่ข่ายชนิด Warning



รูปที่ 4.22 หน้าต่างแสดงข้อความจากเครื่องแม่ข่ายชนิด Question



รูปที่ 4.23 หน้าต่างแสดงข้อความจากเครื่องแม่ข่ายชนิด Error

- กดปุ่ม [CLOSE] เพื่อปิดโปรแกรม

#### 4.2.1 การรับ-ส่งแฟ้มข้อมูลไปยังเครื่องแม่ข่าย

เป็นส่วนที่ไว้ส่งไฟล์ต่างๆ เช่น แฟ้มข้อความ, แฟ้มรูปภาพ, แฟ้มเสียง, แฟ้มวีดิทัศน์ หรือแฟ้มที่ถูกบีบอัดย่อขนาด เป็นต้น ไปยังเครื่องแม่ข่ายที่ติดต่อบริการ ประกอบด้วยส่วนต่างๆ ดังนี้

1. ปุ่ม [Send Homework] เมื่อคลิกแล้ว จะปรากฏหน้าต่าง OpenFileDialog ดังรูปที่ 4.12 เพื่อเลือกแฟ้มข้อมูลที่ต้องการส่ง เมื่อเลือกได้แล้วจะทำการส่งแฟ้มข้อมูลนั้นไปยังเครื่องแม่ข่าย ณ สาระบบ ../File/ ของทางฝั่งเครื่องลูกข่าย
2. ในการรับแฟ้มข้อมูล เมื่อได้รับแฟ้มข้อมูลเรียบร้อยแล้ว จะปรากฏหน้าต่าง ดังรูปที่ 4.13 ซึ่งจะบอกชื่อแฟ้มข้อมูลที่รับด้วย

## บทที่ 5

### สรุปผลการทำงานของโปรแกรมและข้อเสนอแนะ

#### 5.1 สรุปผลการทำงานของโปรแกรม

การทำงานของระบบจะทำการตรวจสอบ และจัดการการทำงานของเครื่องลูกข่ายหรือเครื่องคอมพิวเตอร์ที่นักเรียนใช้ในการเรียนบนเครือข่ายท้องถิ่น เพื่อให้การเรียนการสอนผ่านระบบเครือข่ายคอมพิวเตอร์ท้องถิ่นนั้นมีความสมบูรณ์ ประสบความสำเร็จ และเป็นไปอย่างมีประสิทธิภาพ ปัญหาพิเศษนี้จัดทำขึ้น เพื่อสร้างเครื่องมือหรือโปรแกรมที่สามารถนำไปใช้ในห้องเรียนคอมพิวเตอร์ได้จริง ซึ่งจากการทดสอบการใช้งานโปรแกรมของปัญหาพิเศษเรื่อง โปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย สามารถสรุปผลการทดสอบได้ดังนี้

1. ส่วนของการเรียกดูข้อมูล สามารถเรียกดูข้อมูลต่างๆ ของเครื่องลูกข่าย ได้แก่ Window Login Name, Computer Name และ IP Address ซึ่งเป็นข้อมูลที่เพียงพอต่อการใช้งานในห้องเรียนคอมพิวเตอร์
2. ส่วนของการจัดการ สามารถจัดการกับเครื่องลูกข่ายที่มีในระบบ ได้แก่ การจับภาพหน้าจอของเครื่องลูกข่าย, การแพร่ภาพหน้าจอ(broadcast) ของเครื่องแม่ข่าย และการล็อกห้ามใช้งานเมาส์และแผงแป้นอักขระ เพื่อทำการห้ามใช้งานคอมพิวเตอร์ชั่วคราว
3. ส่วนของการส่งข้อความ สามารถส่งข้อความจากเครื่องแม่ข่ายไปยังเครื่องลูกข่ายได้เพียงฝั่งเดียว โดยข้อความนั้นอาจเป็น การแจ้งข้อมูลประชาสัมพันธ์, การแจ้งเตือน, การตั้งคำถาม หรือการแจ้งข้อผิดพลาด เป็นต้น
4. ส่วนของการรับ-ส่งเพิ่มข้อมูล สามารถรับ-ส่งเพิ่มข้อมูลได้ทั้งจากทางฝั่งเครื่องแม่ข่ายและจากทางฝั่งเครื่องลูกข่ายได้

## 5.2 ข้อจำกัดของโปรแกรม

1. โปรแกรมจะสามารถทำงานได้เฉพาะบนระบบปฏิบัติการ Windows XP เท่านั้น
2. ใช้ได้เฉพาะกับเครื่องที่มีการ์ด LAN เท่านั้น
3. เครื่องลูกข่ายต้องทำการเปิดโปรแกรมเพื่อติดต่อเข้ามายังเครื่องแม่ข่ายก่อน จึงจะสามารถตรวจสอบและจัดการกับเครื่องลูกข่ายเครื่องนั้นได้
4. การตรวจสอบและการจัดการกับเครื่องลูกข่ายบนระบบเครือข่ายนั้น ต้องกระทำผ่านทางเครื่องที่เป็นแม่ข่ายเท่านั้น
5. สามารถเชื่อมต่อคอมพิวเตอร์ในเครือข่ายได้ไม่เกิน 50 เครื่อง
6. ในการล็อกเมาส์และแผงแป้นอักขระ ได้กำหนดเวลาดล็อกนานที่สุดไว้ที่ 8 ชั่วโมง เนื่องจากการเรียนการสอนภายในห้องเรียนคอมพิวเตอร์ในวันนั้น มีเวลาทำการเพียง 8 ชั่วโมง
7. ในการส่งข้อความจากเครื่องฝั่งแม่ข่ายไปยังเครื่องฝั่งลูกข่ายนั้น สามารถส่งได้มากที่สุด 32767 ตัวอักษร
8. ในการส่งเพิ่มข้อมูลทั้งจากฝั่งแม่ข่ายและจากฝั่งลูกข่ายนั้น ขนาดเพิ่มข้อมูลที่ส่งไม่ควรเกิน 100 เมกกะไบต์ เนื่องจากข้อจำกัดของหน่วยความจำ ในขณะที่เขียนข้อมูล ณ ปลายทาง จำเป็นต้องเขียนข้อมูลลงหน่วยความจำหลักก่อน
9. ในการส่งเพิ่มข้อมูลจากเครื่องลูกข่ายมากกว่า 1 เครื่องนั้น ทางฝั่งแม่ข่ายจะต้องรอให้การรับเพิ่มข้อมูลจากเครื่องลูกข่ายหนึ่งนั้นเสร็จสมบูรณ์ก่อน แล้วจึงรับเพิ่มข้อมูลจากเครื่องลูกข่ายที่สองได้

### 5.3 ข้อเสนอแนะ

การพัฒนาโปรแกรมตรวจสอบเครื่องคอมพิวเตอร์ผ่านระบบเครือข่ายนั้น สามารถปรับปรุงคุณลักษณะเดิมที่มีอยู่ให้ดีขึ้น หรือนำโปรแกรมไปพัฒนาให้มีคุณลักษณะใหม่เพิ่มเติม เพื่อเป็นประโยชน์เพิ่มขึ้นในการเรียนการสอนภายในห้องเรียนคอมพิวเตอร์ ดังเช่น

#### 5.3.1 การปรับปรุง

##### 5.3.1.1 ส่วนของการดูหน้าจอเครื่องลูกข่าย

- ปรับปรุงให้เป็นการดูหน้าจอแบบเวลาจริง(real-time) หรือเป็นการเล่นภาพขณะรับต่อเนื่อง(streaming video)
- ปรับปรุงให้มีการบันทึกความเคลื่อนไหวของเครื่องลูกข่ายไว้ดูภายหลังได้

##### 5.3.1.2 ส่วนของการแพร่ภาพหน้าจอเครื่องแม่ข่าย

- ปรับปรุงให้เป็นการแพร่ภาพแบบเวลาจริง หรือเป็นการเล่นภาพขณะรับต่อเนื่อง

##### 5.3.1.3 ส่วนของการล็อกเมาส์และแผงแป้นอักขระ

- ปรับปรุงไม่ให้เครื่องลูกข่ายสามารถเปิด Window Task Manager ได้

##### 5.3.1.4 ส่วนของการส่งแฟ้มข้อมูล

- ปรับปรุงให้สามารถส่งแฟ้มข้อมูลได้ที่ละหลายๆ แฟ้มข้อมูลได้
- ปรับปรุงให้สามารถกำหนดตำแหน่งปลายทางที่จะส่งแฟ้มข้อมูลไปได้

#### 5.3.2 การพัฒนาคุณลักษณะใหม่เพิ่มเติม

##### 5.3.2.1 ส่วนของการสื่อสาร

- เพิ่มการสนทนาโต้ตอบด้วยตัวหนังสือ(Chat) แบบเวลาจริง
- เพิ่มการสนทนาโต้ตอบด้วยเสียง(Audio Chat) แบบเวลาจริง

##### 5.3.2.2 ส่วนของการควบคุม

- เพิ่มคุณสมบัติให้เครื่องแม่ข่ายสามารถควบคุมการเคลื่อนไหวเมาส์ของเครื่องลูกข่ายได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

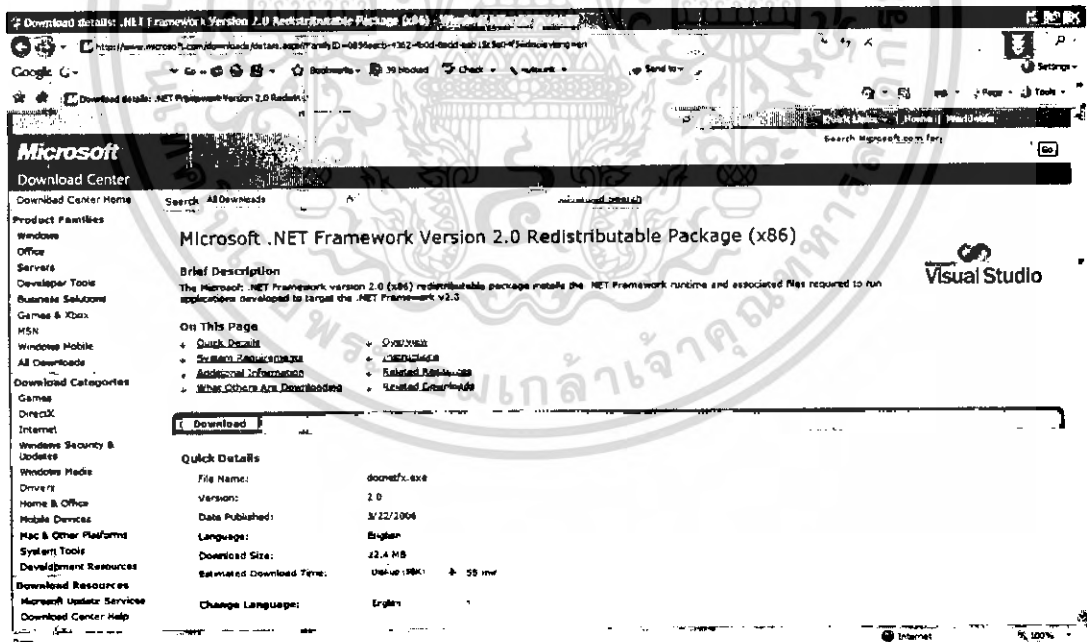
## ภาคผนวก ก การติดตั้งโปรแกรม

### 1. คุณสมบัติขั้นต่ำของเครื่องคอมพิวเตอร์ที่ต้องการ

1. ระบบปฏิบัติการ Microsoft Windows XP Professional
2. หน่วยประมวลผล CPU 1.0 GHz
3. หน่วยความจำหลักขนาด 256 MB
4. หน่วยความสำรองต้องมีพื้นที่ว่างเหลือไม่ต่ำกว่า 50 MB
5. มี Network การ์ด

### 2. การติดตั้ง Microsoft .NET framework 2.0

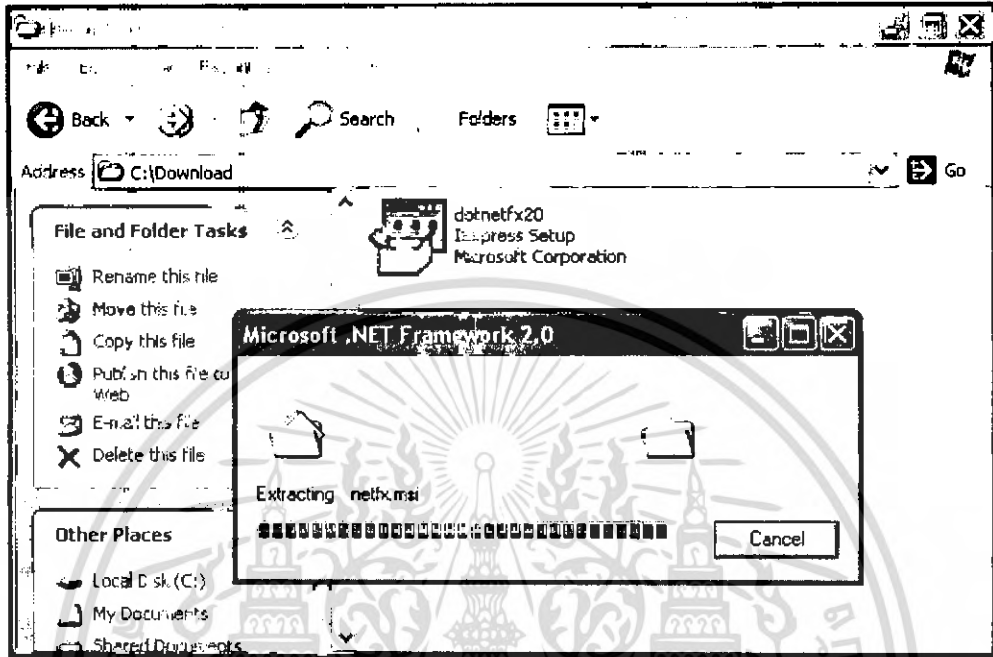
- 2.1 ทำการหาเพิ่มข้อมูลชื่อ dotnetfx2.exe ซึ่งสามารถ download ได้จากเว็บ <http://www.microsoft.com> หรือติดตั้งได้จากใน folder runtime ภายในแผ่น CD-Rom



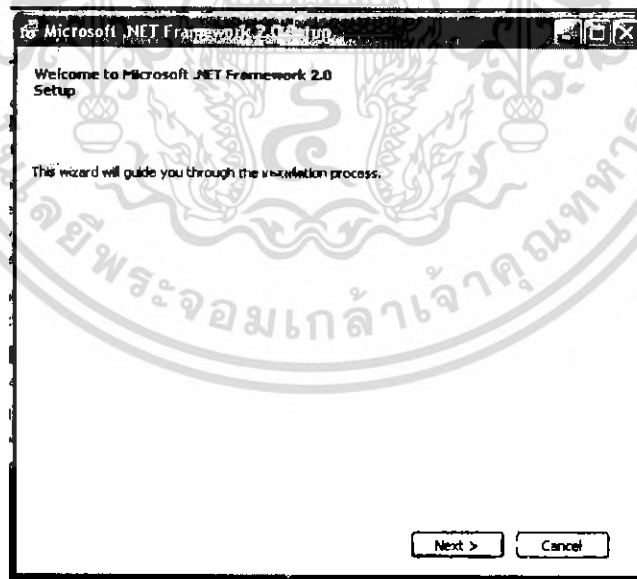
รูปที่ ก.1 แสดงหน้าต่าง download Microsoft .NET Framework 2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ทำการดับเบิลคลิกที่แฟ้มข้อมูล dotnetfx20.exe เพื่อเริ่มขั้นตอนการติดตั้ง



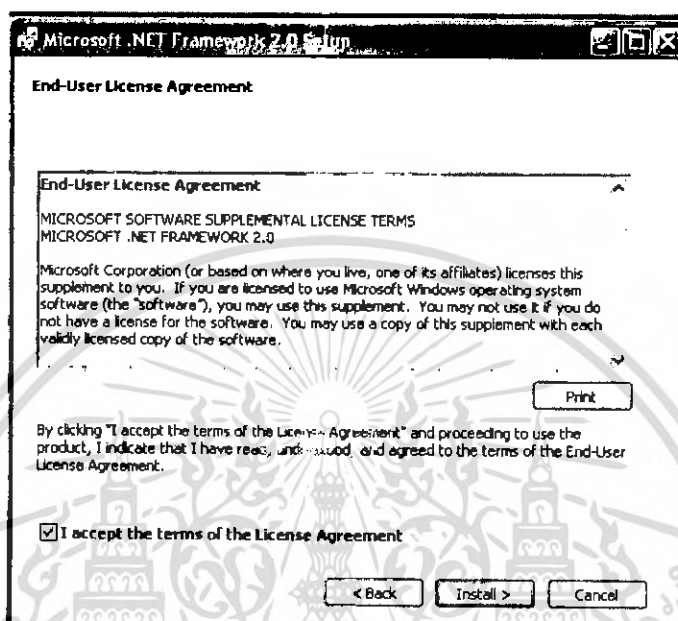
รูปที่ ก.2 แสดงขั้นตอนเริ่มกระบวนการติดตั้งของ Microsoft.NET Framework 2.0



รูปที่ ก.3 แสดงขั้นตอนพร้อมเริ่มการติดตั้งของ Microsoft.NET Framework 2.0

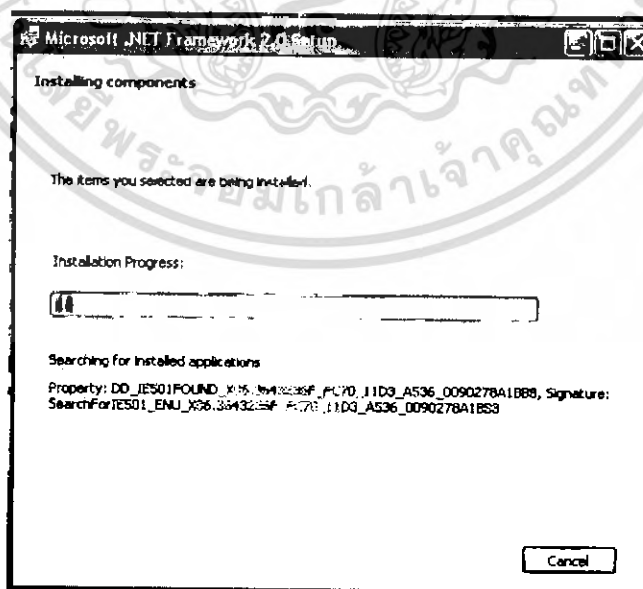
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ทำการกดปุ่ม Next เพื่อไปสู่ขั้นตอนต่อไป ในหน้าต่างไปให้เลือก "I'm accept term of the License Agreement" แล้วกดปุ่ม install เพื่อเริ่มการติดตั้ง



รูปที่ ก.4 แสดงหน้ายอมรับเงื่อนไข ข้อตกลงการใช้งาน Microsoft .NET Framework 2.0

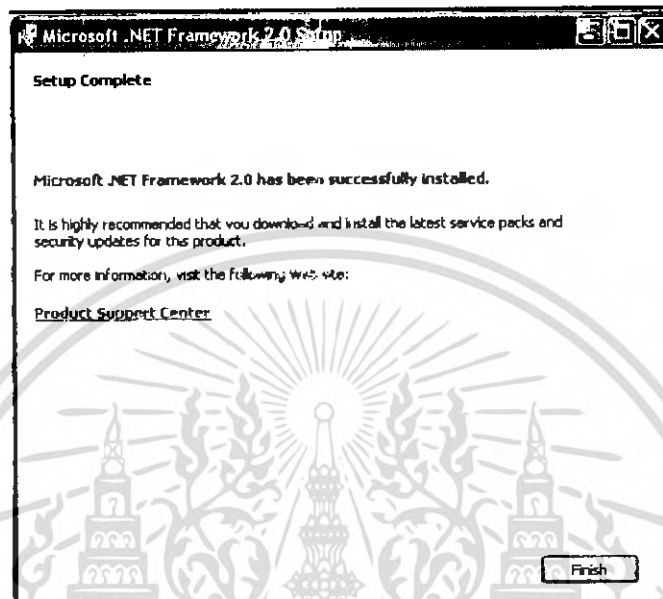
2.4 โปรแกรมจะเริ่มทำการติดตั้ง



รูปที่ ก.5 แสดงขั้นตอนดำเนินการติดตั้งของ Microsoft .NET Framework 2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เมื่อโปรแกรมติดตั้งเสร็จสิ้นจะแจ้งหน้าขึ้นมาดังรูป ให้กด Finish เพื่อจบขั้นตอนการติดตั้งอย่างสมบูรณ์



รูปที่ ก.6 แสดงขั้นตอนเสร็จสิ้นการติดตั้งของ Microsoft .NET Framework 2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

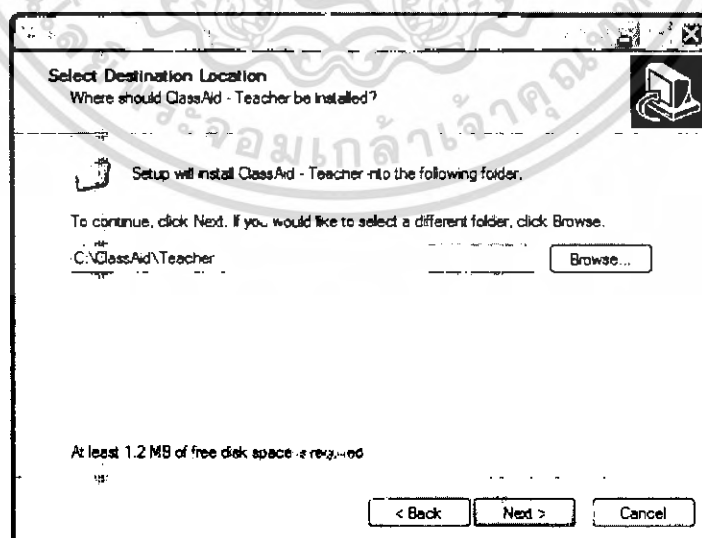
### 3. การติดตั้งโปรแกรมบนฝั่งเซิร์ฟเวอร์

- 3.1 ทำการดับเบิลคลิกที่เพิ่มข้อมูล Teacher Setup.exe เพื่อเริ่มการติดตั้ง หลังจากแสดงหน้าต่างตอนรับการติดตั้งให้กด Next เพื่อดำเนินการไปยังขั้นตอนต่อไป



รูปที่ ก.7 แสดงขั้นตอนเริ่มกระบวนการติดตั้งโปรแกรมบนฝั่งแม่ข่าย

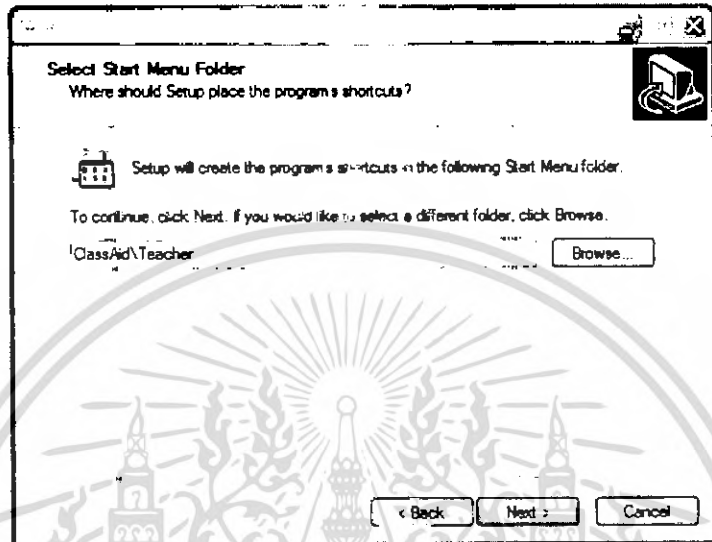
- 3.2 ตั้งค่าเพิ่มที่โปรแกรมจะติดตั้งลงไป โดยค่ามาตรฐานจะเป็น C:\ClassAid\Teacher แล้วทำการกด Next



รูปที่ ก.8 แสดงขั้นตอนการตั้งค่าเพิ่มที่ติดตั้งโปรแกรมบนฝั่งแม่ข่าย

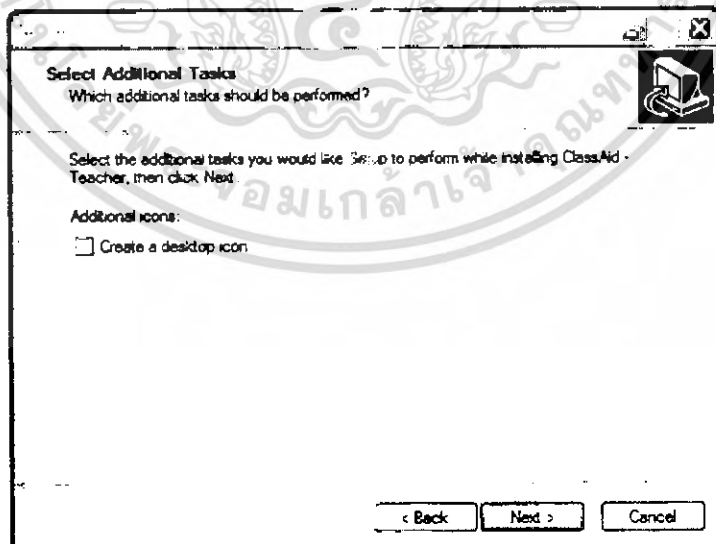
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3.3 ตั้งค่าการติดตั้ง start menu ของโปรแกรม โดยค่ามาตรฐานจะเป็น ClassAid\Teacher แล้วทำการกด Next



รูปที่ ก.9 แสดงขั้นตอนการตั้งค่า start menu ของโปรแกรมบนฝั่งแม่ข่าย

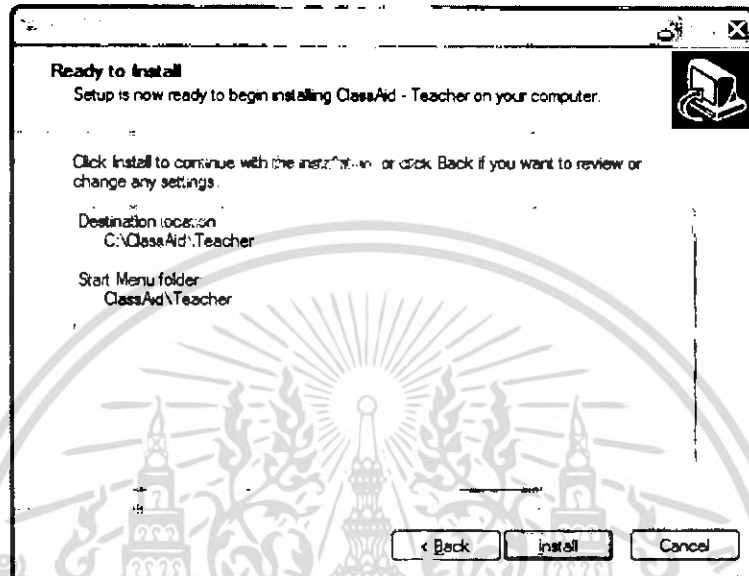
- 3.4 ทำการเลือกว่าต้องการจะสร้างสัญลักษณ์รูป(icon) ลัดบนเดสก์ทอปหรือไม่ ค่ามาตรฐานคือ ไม่ทำการสร้าง แล้วกด Next



รูปที่ ก.10 แสดงขั้นตอนการเลือกติดตั้งสัญลักษณ์รูปลัดบนเดสก์ทอปของโปรแกรมบนฝั่งแม่ข่าย

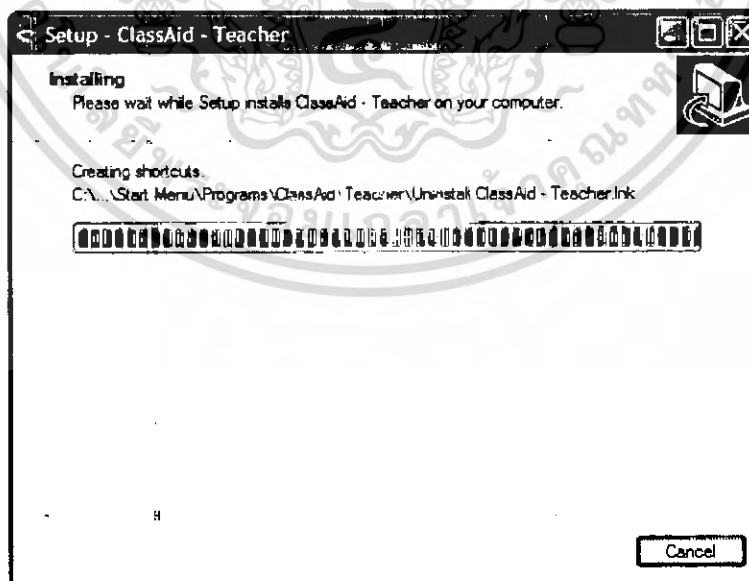
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ทำการตรวจสอบข้อมูลการติดตั้งที่ได้ถูกตั้งค่าไว้ก่อนหน้านี้ หากค่าถูกต้องให้ทำการกด Install เพื่อเริ่มกระบวนการคัดลอกเพิ่มข้อมูลลงสู่เครื่อง



รูปที่ ก.11 แสดงข้อมูลการติดตั้งที่ได้ถูกกำหนดไปก่อนหน้านี้ของโปรแกรมบนฝั่งแม่ข่าย

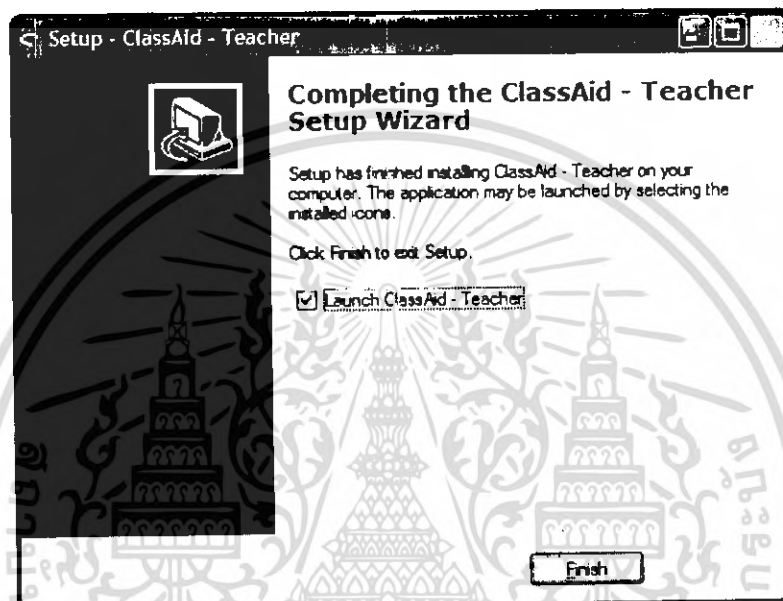
3.6 โปรแกรมติดตั้งจะทำการคัดลอกเพิ่มข้อมูลลงสู่เครื่อง



รูปที่ ก.12 แสดงสถานการณ์คัดลอกเพิ่มข้อมูลลงสู่เครื่องของโปรแกรมบนฝั่งแม่ข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3.7 เมื่อการคัดลอกข้อมูลสำเร็จ โปรแกรมติดตั้งจะทำการแจ้งให้ทราบ และสามารถเลือกได้ว่าจะทำการเริ่มการทำงานของโปรแกรมฝั่งแม่ข่ายเลยหรือไม่ หน้าต้องการเริ่มให้เลือกถูกที่หน้าข้อความ "Launch ClassAid - Teacher" แล้วกด finish เพื่อจบกระบวนการติดตั้ง

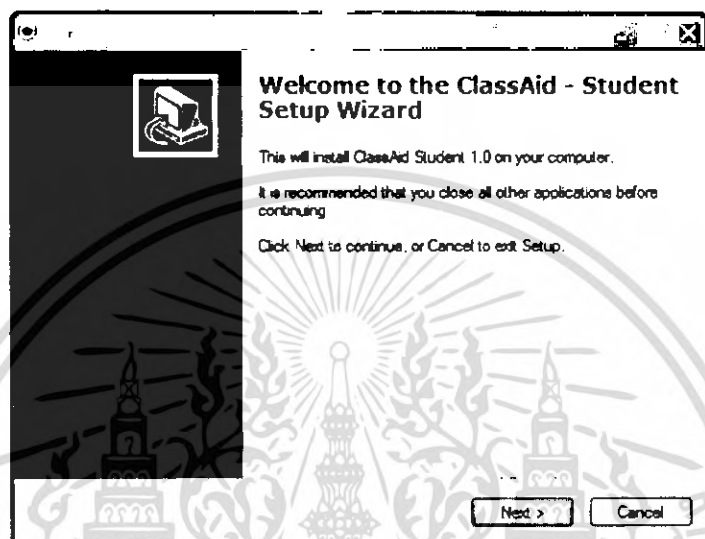


รูปที่ ก.13 แสดงขั้นตอนเสร็จสิ้นการลงโปรแกรมบนฝั่งแม่ข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

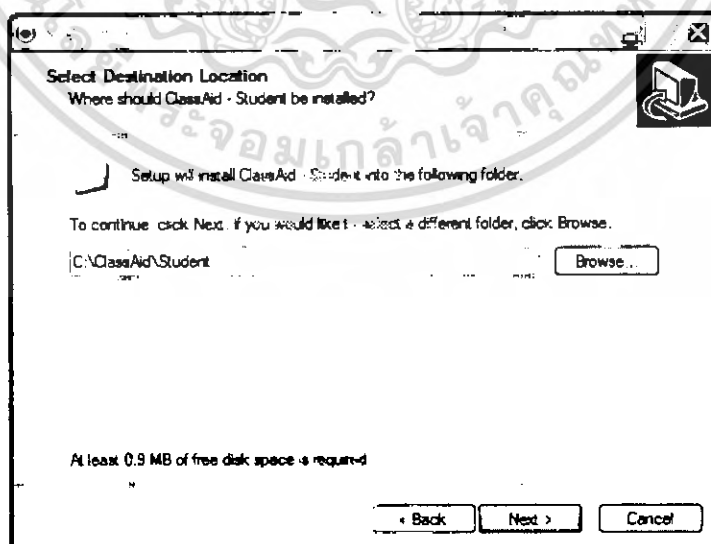
#### 4. การติดตั้งโปรแกรมบนฝั่งลูกข่าย

- 4.1 ทำการดับเบิลคลิกที่เพิ่มข้อมูล Student Setup.exe เพื่อเริ่มการติดตั้ง หลังจากแสดงหน้าจอตอนรับการติดตั้งให้กด Next เพื่อดำเนินการไปยังขั้นตอนต่อไป



รูปที่ ก.14 แสดงขั้นตอนเริ่มกระบวนการติดตั้งโปรแกรมบนฝั่งลูกข่าย

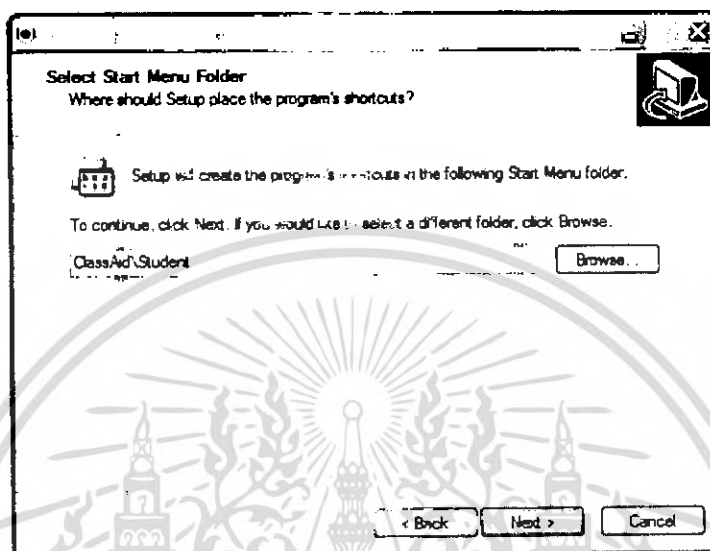
- 4.2 ตั้งค่าเพิ่มที่โปรแกรมจะติดตั้งลงไป โดยค่ามาตรฐานจะเป็น C:\ClassAid\Student แล้วทำการกด Next



รูปที่ ก.15 แสดงขั้นตอนการตั้งค่าเพิ่มที่ติดตั้งโปรแกรมบนฝั่งลูกข่าย

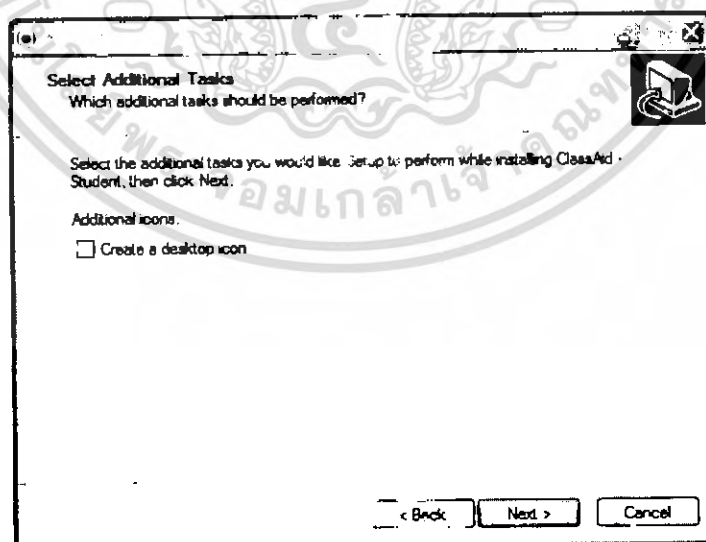
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.3 ตั้งค่าการติดตั้ง start menu ของโปรแกรม โดยค่ามาตรฐานจะเป็น ClassAid\Student แล้วทำการกด Next



รูปที่ ก.16 แสดงขั้นตอนการตั้งค่า start menu ของโปรแกรมบนฝั่งลูกข่าย

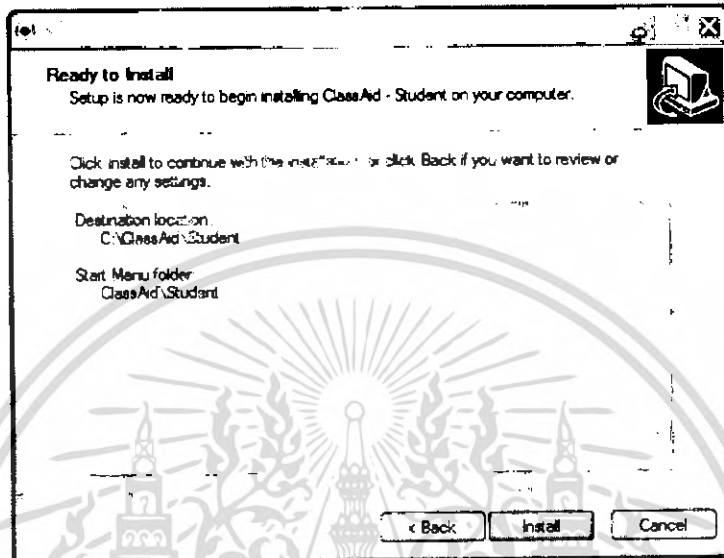
- 4.4 ทำการเลือกว่าต้องการจะสร้างสัญลักษณ์เดสก์ท็อปหรือไม่ ค่ามาตรฐานคือ ไม่ทำการสร้าง แล้วกด Next



รูปที่ ก.17 แสดงขั้นตอนการเลือกติดตั้งสัญลักษณ์เดสก์ท็อปของโปรแกรมบนฝั่งลูกข่าย

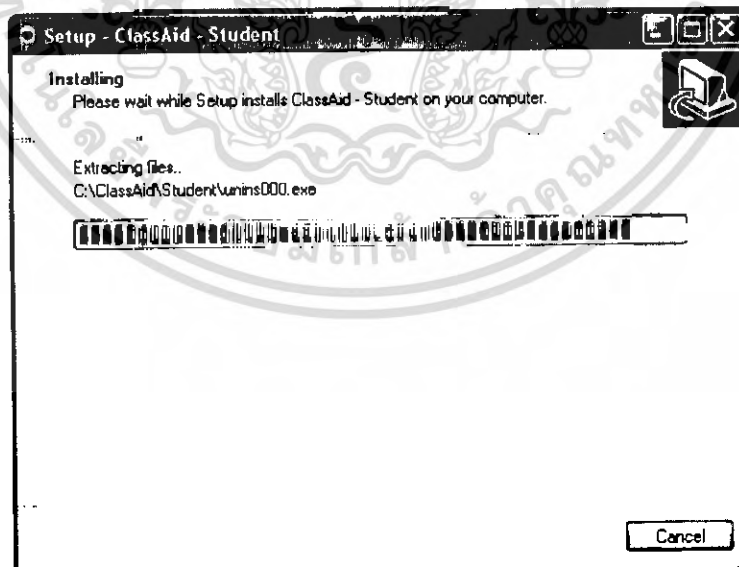
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ทำการตรวจสอบข้อมูลการติดตั้งที่ได้ถูกตั้งค่าไว้ก่อนหน้านี้ หากค่าถูกต้องให้ทำการกด Install เพื่อเริ่มกระบวนการคัดลอกเพิ่มข้อมูลลงสู่เครื่อง



รูปที่ ก.18 แสดงข้อมูลการติดตั้งที่ได้ถูกกำหนดไปก่อนหน้านี้ของโปรแกรมบนฝั่งลูกข่าย

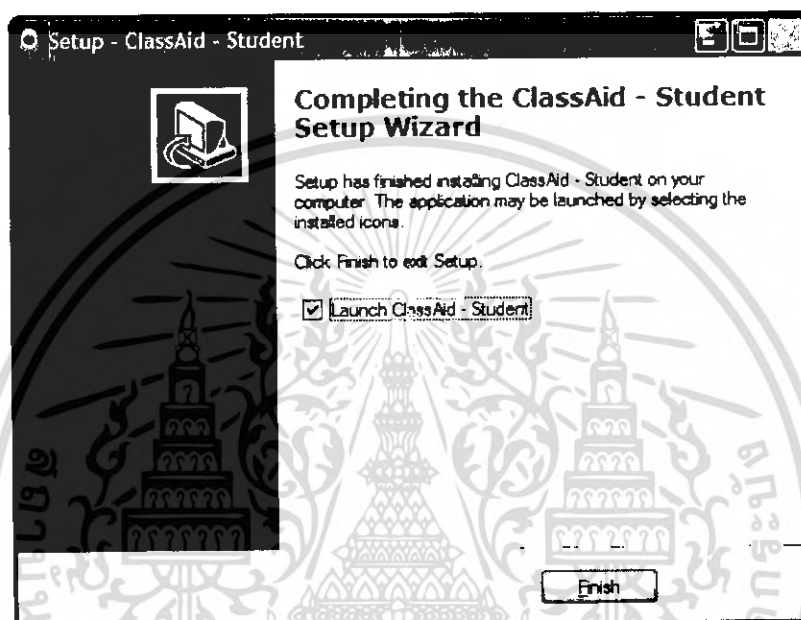
4.6 โปรแกรมติดตั้งจะทำการคัดลอกเพิ่มข้อมูลลงสู่เครื่อง



รูปที่ ก.19 แสดงสถานการณ์คัดลอกเพิ่มข้อมูลลงสู่เครื่องของโปรแกรมบนฝั่งลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.7 เมื่อการคัดลอกข้อมูลสำเร็จ โปรแกรมติดตั้งจะทำการแจ้งให้ทราบ และสามารถเลือกได้ว่าจะทำการเริ่มการทำงานของโปรแกรมฝังลูกข่ายเลยหรือไม่ หน้าต้องการเริ่มให้เลือกถูกที่หน้าข้อความ "Launch ClassAid - Student" แล้วกด finish เพื่อจบกระบวนการติดตั้ง

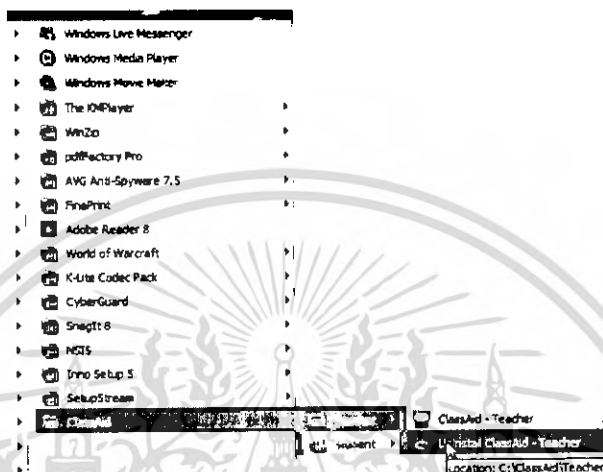


รูปที่ ก.20 แสดงขั้นตอนเสร็จสิ้นการลงโปรแกรมบนฝังลูกข่าย

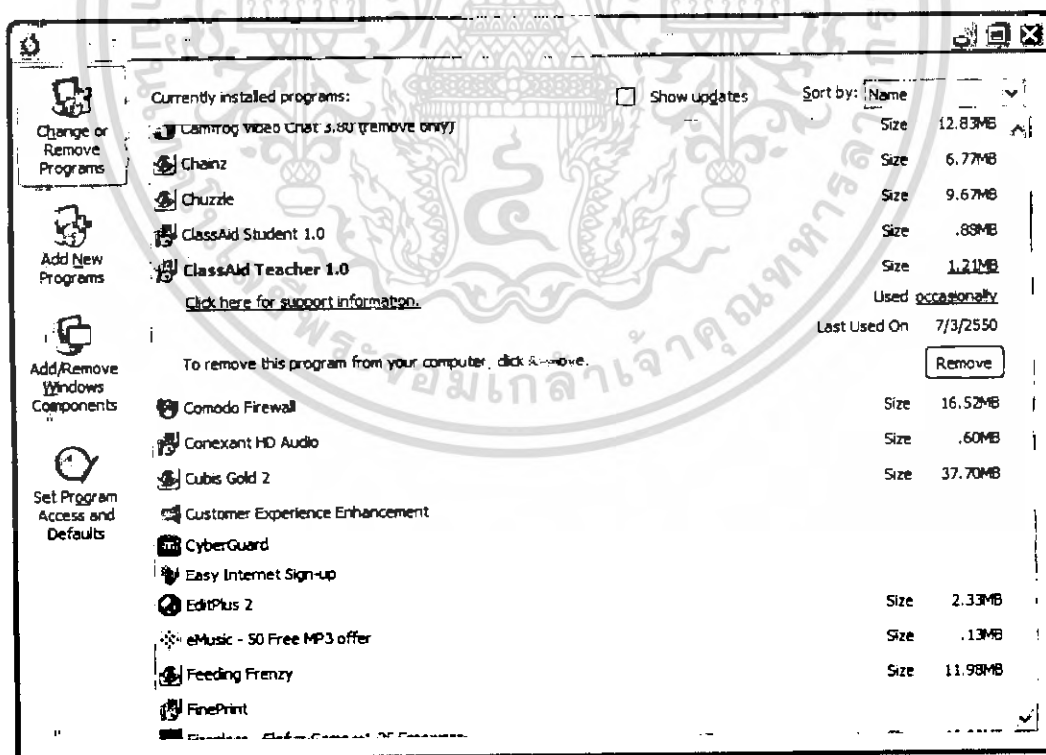
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5. การถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่าย

5.1 เลือก Uninstall ClassAid – Teacher จาก start menu หรือ Remove จาก Add or Remove Programs ใน Control Panel



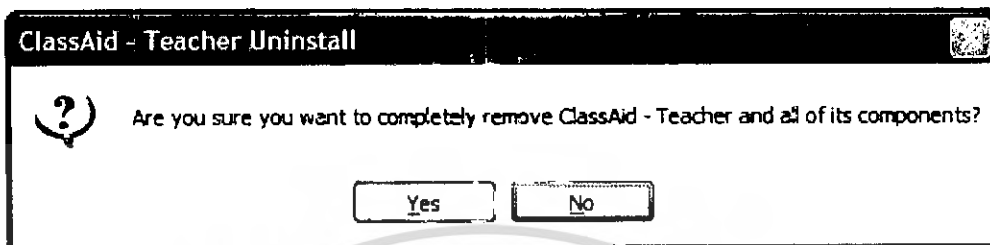
รูปที่ ก.21 แสดงการถอนการติดตั้งจาก Start Menu ของโปรแกรมบนฝั่งแม่ข่าย



รูปที่ ก.22 แสดงการถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่ายจาก Add or Remove Programs

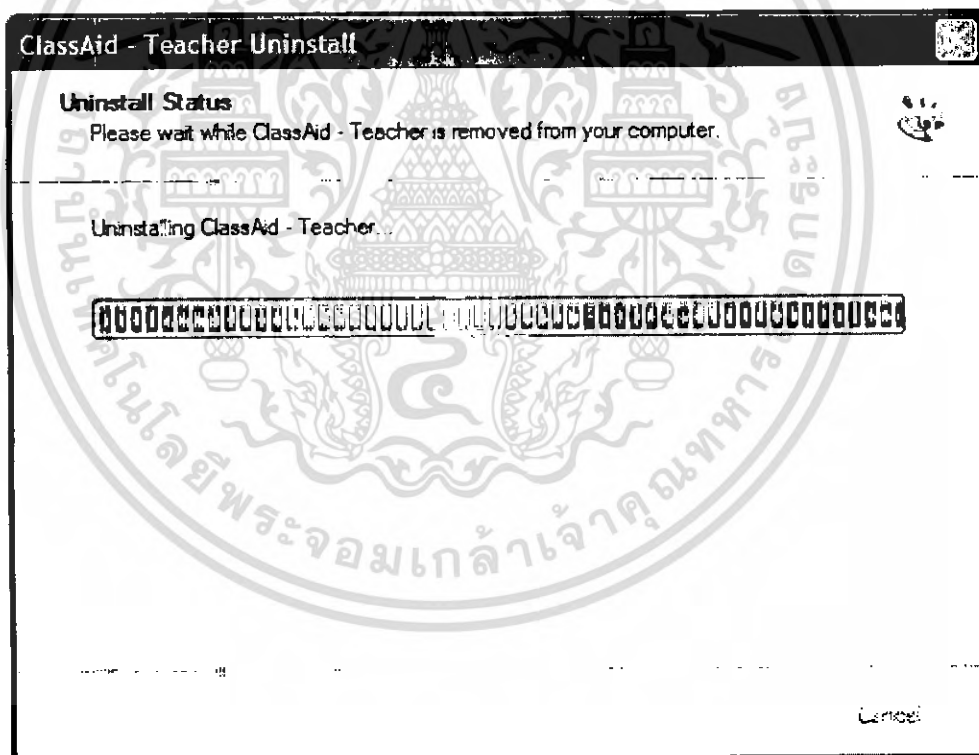
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 โปรแกรมติดตั้งจะทำการถามยืนยันว่าต้องการจะถอนการติดตั้งจริงหรือไม่ ให้ทำการเลือก Yes



รูปที่ ก.23 แสดงการยืนยันว่าต้องการจะถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่ายจริงหรือไม่

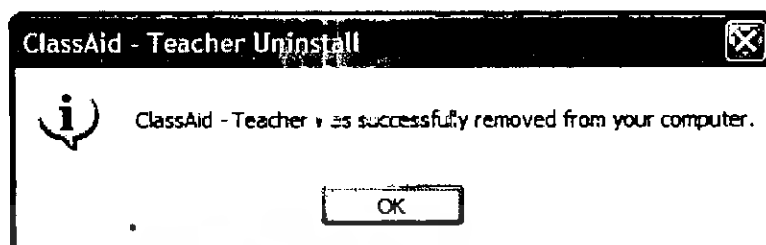
5.3 โปรแกรมจะดำเนินการถอนการติดตั้ง



รูปที่ ก.24 แสดงสถานะการถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

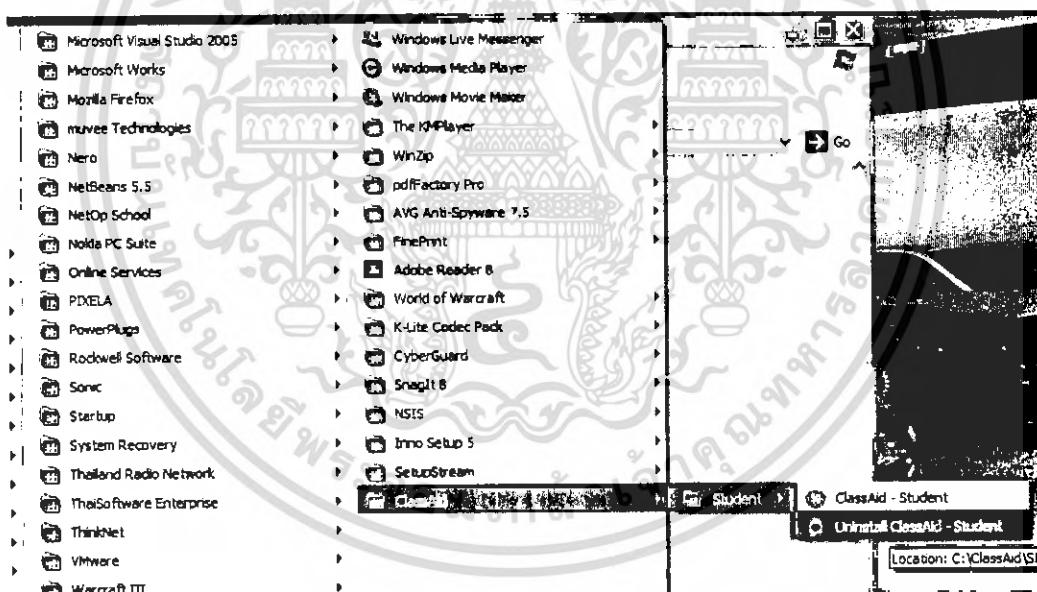
5.4 เมื่อการถอนการติดตั้งสำเร็จ โปรแกรมจะแจ้งผลการดำเนินงาน



รูปที่ ก.25 แสดงการยืนยันการถอนการติดตั้งโปรแกรมบนฝั่งแม่ข่ายเสร็จสิ้น

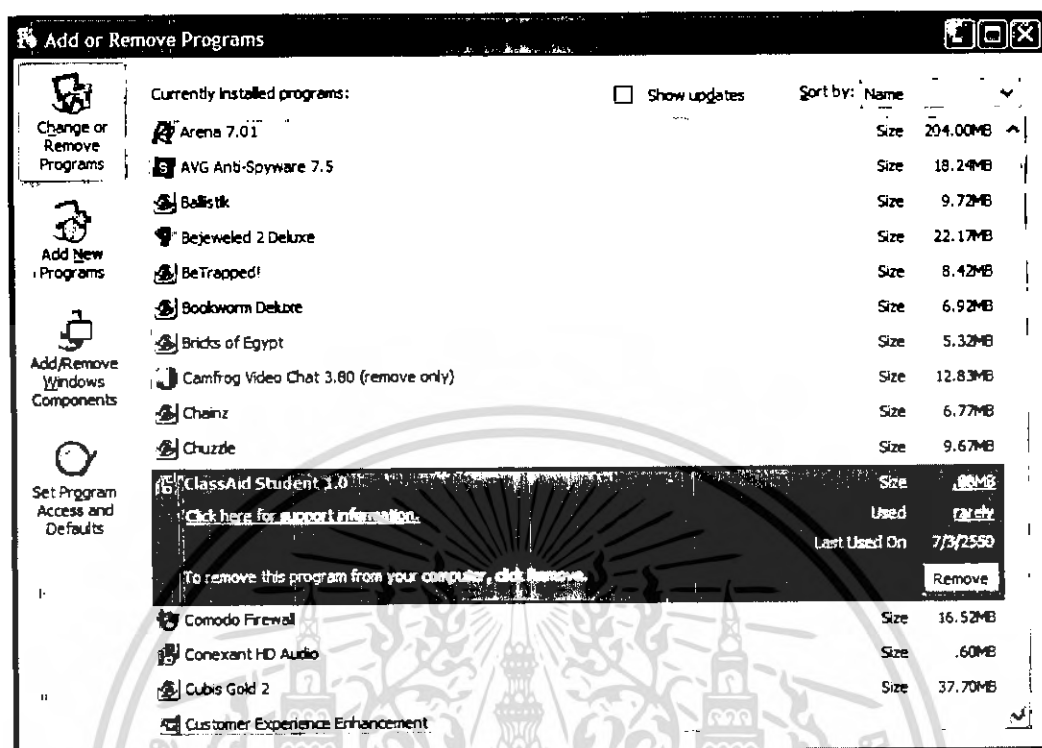
## 6. การถอนการติดตั้งโปรแกรมบนฝั่งลูกข่าย

6.1 เลือก Uninstall ClassAid – Student จาก start menu หรือ Remove จาก Add or Remove Programs ใน Control Panel



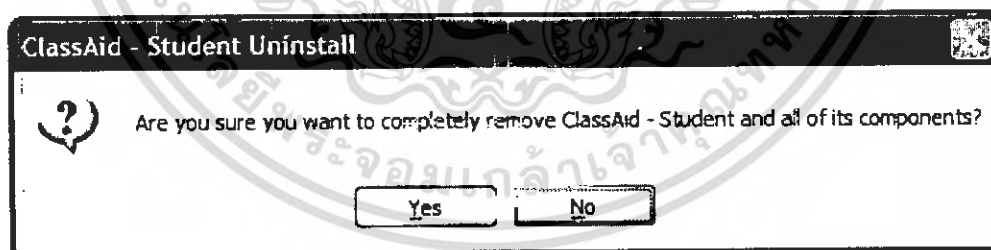
รูปที่ ก.26 แสดงการถอนการติดตั้งโปรแกรมบนฝั่งลูกข่ายจาก Start Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.27 แสดงการถอนการติดตั้งโปรแกรมบนฝั่งลูกข่ายจาก Add or Remove Programs

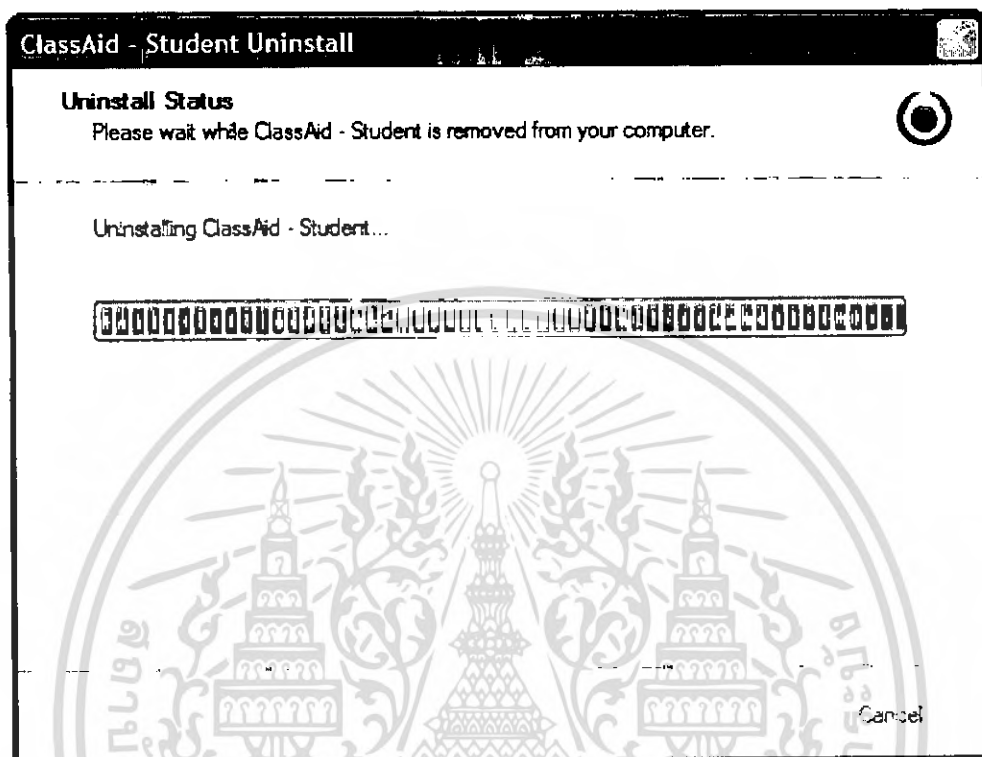
6.2 โปรแกรมติดตั้งจะทำการถามยืนยันว่าต้องการจะถอนการติดตั้งจริงหรือไม่ ให้ทำการเลือก Yes



รูปที่ ก.28 แสดงการยืนยันว่าต้องการจะถอนการติดตั้งโปรแกรมบนฝั่งลูกข่ายจริงหรือไม่

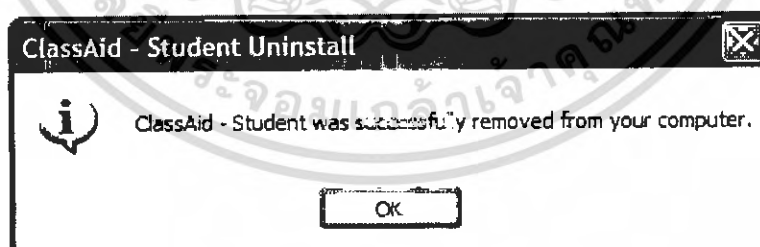
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 โปรแกรมจะดำเนินการถอนการติดตั้ง



รูปที่ ก.29 แสดงสถานะการถอนการติดตั้งโปรแกรมบนฝั่งลูกข่าย

### 6.4 เมื่อการถอนการติดตั้งสำเร็จ โปรแกรมจะแจ้งผลการดำเนินงาน



รูปที่ ก.30 แสดงการยืนยันการถอนการติดตั้งโปรแกรมบนฝั่งลูกข่ายเสร็จสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม

## หนังสือ

- [1] เรืองไกร รังสิพล. เจาะระบบ TCP/IP : จุดอ่อนของโปรโตคอลและวิธีป้องกัน. กรุงเทพฯ : บริษัท โปร วิชั่น จำกัด. 2001.
- [2] ชีรศักดิ์ ฐานกุล. กะเทาะเปลือกวายร้าย Trojan ด้วย Visual Basic. กรุงเทพฯ : ซีเอ็ด ยูเคชั่น. 2549.
- [3] จตุชัย แพงจันทร์ และอนุชิต วุฒิพรพงษ์. เจาะระบบ Network ฉบับสมบูรณ์. พิมพ์ครั้งที่ 2. นนทบุรี : ไอดีซีฯ. 2547.
- [4] Marcia J. Horton. Visual C# 2005 HOW TO PROGRAM. 2nd Ed. New Jersey : Pearson Education, Inc. 2006.

## เว็บไซต์

<http://msdn2.microsoft.com>

<http://en.wikipedia.org>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้