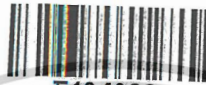


สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบจัดการเครื่องคอมพิวเตอร์ตั้งโต๊ะภายในห้องปฏิบัติการคอมพิวเตอร์

DESKTOP MANAGEMENT FOR COMPUTER LABORATORY



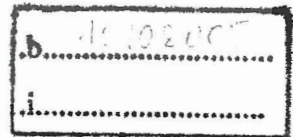
T104002



เลขหมู่.....

เลขทะเบียน 104002

วัน,เดือน,ปี 28 ต.ค. 2552



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของ **องค**การศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรม **ร**วมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยี **ย**พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2551

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจัดการเครื่องคอมพิวเตอร์ตั้งโต๊ะภายในห้องปฏิบัติการคอมพิวเตอร์

DESKTOP MANAGEMENT FOR COMPUTER LABORATORY

ผู้จัดทำ

1. นางสาวกรรณก เตชะรัตนยืนยง รหัสนักศึกษา 48010015
2. นายรัฐศักดิ์ น้อมบุญสงศรี รหัสนักศึกษา 48010735
3. นายสนรยา ตันเล่ง รหัสนักศึกษา 48010929



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจัดการเครื่องคอมพิวเตอร์ตั้งโต๊ะภายในห้องปฏิบัติการคอมพิวเตอร์

นางสาวกรกนก เตชะรัตนยืนยง 48010015

นายรัฐศักดิ์ น้อมบุญสงศรี 48010735

นายสนรยา ตันเล่ง 48010929

อาจารย์เกียรติณรงค์ ทองประเสริฐ อาจารย์ที่ปรึกษา
ปีการศึกษา 2551

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ได้นำเสนอซอฟต์แวร์ระบบบริหารจัดการเครื่องคอมพิวเตอร์ โดยออกแบบมาเพื่อรองรับการบริหารจัดการระบบคอมพิวเตอร์ภายในห้องปฏิบัติการคอมพิวเตอร์ ซึ่งจุดมุ่งหมายของโครงการนี้คือ สามารถจะนำซอฟต์แวร์นี้ไปใช้งานในห้องปฏิบัติการคอมพิวเตอร์ โดยสามารถใช้งานได้อย่างง่ายดาย พร้อมด้วยฟังก์ชันการใช้งานที่หลากหลายและครอบคลุม ทำให้การบริหารจัดการเครื่องคอมพิวเตอร์ภายในปฏิบัติการคอมพิวเตอร์เป็นไปอย่างมีประสิทธิภาพ

ซอฟต์แวร์นี้ใช้ภาษา C# ในการพัฒนา ซึ่งมีการออกแบบให้ใช้งานได้สะดวก โดยมีการแสดงตำแหน่งของเครื่องในห้องปฏิบัติการ ตรวจสอบสถานะการทำงานของเครื่องลูกข่าย สามารถจะควบคุมการใช้โปรแกรม ณ ขณะนั้น การใช้พอร์ต USB และการใช้อินเทอร์เน็ต และยังสามารถดูฮาร์ดแวร์และซอฟต์แวร์ของเครื่องลูกข่ายได้อีกด้วย

DESKTOP MANAGEMENT FOR COMPUTER LABORATORY

Ms. Kornkanok Techaratanayuenyong 48010015

Mr. Rattasak Nombumsongsri 48010735

Mr. Sontaya Tanleng 48010929

Mr. Keitinarong Thongprasert Advisor

Academic Year 2008

ABSTRACT

This thesis presents about software that is a computer management. It is designed to manage the computer in computer laboratory. A purpose of this project is to be used in computer laboratory that can use conveniently with many functions that implementing the desktop management in computer laboratory work efficiently.

This software develops on C# language and designs to use easily. It can show the computer location in computer laboratory, check the client status and it can control the program using, USB port using and Internet using. In addition, it can monitor the client's hardware and software.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ด้วยคำแนะนำ คำปรึกษาจากหลายๆฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษา ที่ให้โอกาสให้ข้าพเจ้าได้ทำปริญญานิพนธ์ฉบับนี้ คอยให้คำแนะนำ และความช่วยเหลือเสมอมา คือ ท่านอาจารย์เกียรติณรงค์ ทองประเสริฐ ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง พร้อมทั้งขอขอบพระคุณคณาจารย์ทุกท่านมา ณ ที่นี้ด้วย

ขอขอบพระคุณ ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้จัดเตรียมสิ่งอำนวยความสะดวกต่างๆ เพื่อให้การศึกษาและพัฒนาโปรแกรมเป็นไปได้ด้วยความสะดวกและรวดเร็ว

ขอขอบพระคุณ ชุมชนคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ที่ให้สถานที่ในการศึกษาและพัฒนาโปรแกรม พร้อมทั้งสมาชิกชุมชนทุกคน ที่คอยให้กำลังใจเสมอมา

และสุดท้ายนี้ขอขอบพระคุณ บุคคลที่สำคัญที่สุดในชีวิตของข้าพเจ้า ที่ทำให้ข้าพเจ้ามีวันนี้ คือ บิดา มารดา และบุคคลในครอบครัว อันเป็นที่เคารพรัก ซึ่งได้เลี้ยงดูและสั่งสอนข้าพเจ้าเป็นอย่างดี คอยให้คำปรึกษาต่างๆ พร้อมให้โอกาสทางด้านการศึกษาอย่างเต็มที่ คอยให้กำลังใจ และความรักเสมอมา ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

นางสาวกรรณก เตชะรัตนยืนยง

นายรัฐศักดิ์ น้อมบุญส่งศรี

นายสนธยา ต้นเล่ง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตของโครงการ.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	3
2.1 Network Protocols.....	3
2.1.1 Basic Protocols.....	4
2.1.2 Internet Protocols.....	4
2.2 Sockets.....	4
2.2.1 Overview.....	4
2.2.2 Interface Functionality.....	5
2.2.3 Socket Types.....	6
2.2.4 Working with Sockets in .Net.....	6
2.2.5 Working with Sockets.....	7
2.3 Stream Sockets (Connection Oriented Socket).....	8
2.3.1 ข้อควรระวังในการใช้งาน Stream Socket.....	9
2.4 Datagram Sockets (Connectionless Sockets).....	12
2.4.1 การสร้าง Connectionless Socket.....	12
2.4.2 การใช้งาน Connect () ร่วมกับ Connectionless Socket.....	13
2.4.3 ข้อดีของ Connectionless Sockets.....	14
2.4.4 ข้อดีของ Connectionless Sockets.....	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4.5 ข้อควรระวังในการใช้งาน Connectionless Socket.....	14
2.5 ตัวอย่าง Desktop Management ที่มีใช้ในปัจจุบัน.....	16
2.5.1 StarCat 5.0 Enterprise.....	16
2.5.2 IBM Tivoli Software.....	17
บทที่ 3 การออกแบบและพัฒนา.....	19
3.1 Requirement.....	19
3.1.1 Server Requirement.....	19
3.1.2 Client Requirement.....	19
3.2. Use Case Diagram.....	20
3.3 Class Diagram.....	20
3.3.1 เครื่องลูกข่าย.....	20
3.3.2 เครื่องเซิร์ฟเวอร์.....	21
3.4 Sequence Diagram.....	22
3.4.1 การติดต่อและส่งคำสั่งระหว่างเซิร์ฟเวอร์กับเครื่องลูกข่าย.....	22
3.4.2 การแสดงสถานะของเครื่องลูกข่าย.....	22
3.4.3 การเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย.....	23
3.4.4 การเรียกดูโปรเซสของเครื่องลูกข่าย.....	23
3.4.5 การควบคุมการใช้งานพอร์ต USB.....	24
3.4.6 การควบคุมการใช้งานแอปพลิเคชัน.....	24
3.4.7 การควบคุมการเข้าเว็บไซต์.....	25
3.5 State Diagram.....	25
3.5.1 การติดต่อและส่งคำสั่งระหว่างเซิร์ฟเวอร์กับเครื่องลูกข่าย.....	25
3.5.2 การแสดงสถานะของเครื่องลูกข่าย.....	26
3.5.3 การเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย.....	26
3.5.4 การเรียกดูโปรเซสของเครื่องลูกข่าย.....	27
3.5.5 การควบคุมการใช้งานพอร์ต USB.....	27
3.5.6 การควบคุมการใช้งานแอปพลิเคชัน.....	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.5.7 การควบคุมการเข้าเว็บไซต์.....	28
3.5.8 การตั้งค่า IP Address.....	29
บทที่ 4 การทดลองและผลการทดลอง.....	30
4.1 เครื่องเซิร์ฟเวอร์.....	30
4.1.1 หน้าต่างควบคุมของเซิร์ฟเวอร์.....	30
4.1.2 การแสดงสถานะของเครื่องลูกข่าย.....	30
4.1.3 การเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย.....	31
4.1.4 การเรียกดูโพสเซสของเครื่องลูกข่าย.....	33
4.1.5 การควบคุมการใช้พอร์ต USB.....	34
4.1.6 การควบคุมการใช้งานแอปพลิเคชัน.....	35
4.1.7 การควบคุมการใช้งานเว็บไซต์.....	36
4.1.8 การกำหนด IP Address ให้กับเครื่องลูกข่าย.....	37
4.1.9 การประกาศไปถึงเครื่องลูกข่าย.....	38
4.1.10 การสั่งปิดเครื่องลูกข่าย.....	40
4.2 เครื่องลูกข่าย.....	40
บทที่ 5 บทวิจารณ์และสรุป.....	41
5.1 บทสรุป.....	41
5.2 วิจารณ์สิ่งที่ได้จากโครงงาน.....	41
5.3 ปัญหาอุปสรรคและแนวทางแก้ไข.....	41
5.4 แนวทางการพัฒนาต่อ.....	42
บรรณานุกรม.....	43
ภาคผนวก คู่มือการใช้งานซอฟต์แวร์.....	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 e.ErrorCode ที่สำคัญ.....	16



สารบัญรูป

	หน้า
รูปที่ 2.1 OSI-TCP/IP Protocol Stack.....	3
รูปที่ 2.2 Diagram ขั้นตอนการสร้าง Passive Sockets.....	8
รูปที่ 2.3 Diagram ขั้นตอนการสร้าง Active Sockets.....	8
รูปที่ 2.4 แบบจำลอง TCP Buffer.....	9
รูปที่ 3.1 Use Case Diagram.....	20
รูปที่ 3.2 Class Diagram ของเครื่องลูกข่าย.....	20
รูปที่ 3.3 Class Diagram ของเครื่องเซิร์ฟเวอร์.....	21
รูปที่ 3.4 Sequence Diagram แสดงการติดต่อและส่งคำสั่งระหว่างเซิร์ฟเวอร์กับเครื่องลูกข่าย.....	22
รูปที่ 3.5 Sequence Diagram แสดงสถานะของเครื่องลูกข่าย.....	22
รูปที่ 3.6 Sequence Diagram แสดงการเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย.....	23
รูปที่ 3.7 Sequence Diagram แสดงการเรียกดูโปรเซสของเครื่องลูกข่าย.....	23
รูปที่ 3.8 Sequence Diagram แสดงการควบคุมการใช้งานพอร์ต USB.....	24
รูปที่ 3.9 Sequence Diagram แสดงการควบคุมการใช้งานแอปพลิเคชัน.....	24
รูปที่ 3.10 Sequence Diagram แสดงการควบคุมการเข้าเว็บไซต์.....	25
รูปที่ 3.11 State Diagram แสดงการติดต่อและส่งคำสั่งระหว่างเซิร์ฟเวอร์กับเครื่องลูกข่าย.....	25
รูปที่ 3.12 State Diagram แสดงสถานะของเครื่องลูกข่าย.....	26
รูปที่ 3.13 State Diagram แสดงการเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย.....	26
รูปที่ 3.14 State Diagram แสดงการเรียกดูโปรเซสของเครื่องลูกข่าย.....	27
รูปที่ 3.15 State Diagram แสดงการควบคุมการใช้งานพอร์ต USB.....	27
รูปที่ 3.16 State Diagram แสดงการควบคุมการใช้งานแอปพลิเคชัน.....	28
รูปที่ 3.17 State Diagram แสดงการควบคุมการเข้าเว็บไซต์.....	28
รูปที่ 3.18 State Diagram แสดงการตั้งค่า IP Address.....	29
รูปที่ 4.1 หน้าต่างควบคุมของเครื่องเซิร์ฟเวอร์.....	30
รูปที่ 4.2 เครื่องลูกข่ายมีสถานะเปิด.....	30
รูปที่ 4.3 เครื่องลูกข่ายมีสถานะปิด.....	31
รูปที่ 4.4 เครื่องลูกข่ายที่ยังไม่ได้กำหนด IP Address.....	31
รูปที่ 4.5 การดู IP Address, MAC Address และ Computer Name ของเครื่องลูกข่าย.....	31
รูปที่ 4.6 การเลือก “Show Hardware Information”.....	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.7 ไอคอนการ “Show Hardware Information”	32
รูปที่ 4.8 หน้าต่าง Hardware Information	33
รูปที่ 4.9 การเลือก “Show Running Process”	33
รูปที่ 4.10 หน้าต่าง Processes Information	34
รูปที่ 4.11 การกำหนดการใช้งานพอร์ต USB เครื่องลูกข่าย	34
รูปที่ 4.12 ไอคอน USB	35
รูปที่ 4.13 สัญลักษณ์การไม่อนุญาตให้ใช้งานพอร์ต USB	35
รูปที่ 4.14 การกำหนดแอปพลิเคชันที่ไม่อนุญาตให้ใช้งาน	35
รูปที่ 4.15 การควบคุมการใช้งานแอปพลิเคชันที่ละเครื่อง	36
รูปที่ 4.16 การกำหนดเว็บไซต์ที่ไม่อนุญาตให้ใช้งาน	36
รูปที่ 4.17 การควบคุมการใช้งานเว็บไซต์ที่ละเครื่อง	37
รูปที่ 4.18 การเลือกตั้งค่าเครื่องลูกข่าย	37
รูปที่ 4.19 หน้าต่างตั้งค่าเครื่องลูกข่าย	38
รูปที่ 4.20 การส่งข้อความ ไปยังเครื่องลูกข่ายนั้นๆ	38
รูปที่ 4.21 หน้าต่างการประกาศของเครื่องลูกข่ายนั้นๆ	39
รูปที่ 4.22 ไอคอนการประกาศข้อความไปยังเครื่องลูกข่ายทุกเครื่อง	39
รูปที่ 4.23 หน้าต่างการประกาศของเครื่องลูกข่าย	39
รูปที่ 4.24 ข้อความที่แสดงบนเครื่องลูกข่าย	40
รูปที่ 4.25 การสั่งปิดเครื่องลูกข่าย	40
รูปที่ ก.1 หน้าต่างเพื่อเริ่มการติดตั้งซอฟต์แวร์ DMServer	44
รูปที่ ก.2 หน้าต่างสำหรับกำหนดพาร์ตของไฟล์เดอร์	45
รูปที่ ก.3 หน้าต่างแสดงข้อความพร้อมการติดตั้งซอฟต์แวร์	45
รูปที่ ก.4 หน้าต่างสถานะในการติดตั้งซอฟต์แวร์	46
รูปที่ ก.5 หน้าต่างแสดงการติดตั้งซอฟต์แวร์เสร็จสิ้น	46
รูปที่ ก.6 หน้าต่างเพื่อเริ่มการติดตั้งซอฟต์แวร์ ConsoleClientSetup	47
รูปที่ ก.7 หน้าต่างสำหรับกำหนดพาร์ตของไฟล์เดอร์	47
รูปที่ ก.8 หน้าต่างแสดงข้อความพร้อมการติดตั้งซอฟต์แวร์	48
รูปที่ ก.9 หน้าต่างสถานะในการติดตั้งซอฟต์แวร์	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ ก.10 หน้าต่างแสดงการติดตั้งซอฟต์แวร์เสร็จสิ้น.....	49
รูปที่ ก.11 เครื่องลูกข่ายเปิดอยู่.....	49
รูปที่ ก.12 เครื่องลูกข่ายปิดอยู่.....	50
รูปที่ ก.13 เครื่องลูกข่ายที่ยังไม่ได้ตั้งค่า.....	50
รูปที่ ก.14 หน้าต่างหลักของโปรแกรม.....	50
รูปที่ ก.15 Title Bar.....	51
รูปที่ ก.16 Menu Bar.....	51
รูปที่ ก.17 Tool Bar.....	51
รูปที่ ก.18 ส่วนแสดงส่วนแสดง IP Address.....	51
รูปที่ ก.19 ส่วนแสดงตำแหน่งของเครื่องลูกข่าย.....	52
รูปที่ ก.20 ส่วนแสดง Transparent Mode.....	52
รูปที่ ก.21 Client Properties.....	52
รูปที่ ก.22 การเลือก “Show Hardware Information”.....	53
รูปที่ ก.23 หน้าต่าง Hardware Information.....	53
รูปที่ ก.24 การเลือก “Show Running Process”.....	55
รูปที่ ก.25 หน้าต่าง Processes Information.....	55
รูปที่ ก.26 ไอคอน “Website”.....	56
รูปที่ ก.27 หน้าต่าง Restricted Object.....	56
รูปที่ ก.28 สัญลักษณ์มีการไม่อนุญาตให้ใช้เว็บไซต์.....	56
รูปที่ ก.29 ไอคอน “Application”.....	57
รูปที่ ก.30 สัญลักษณ์มีการไม่อนุญาตให้ใช้โปรแกรม.....	57
รูปที่ ก.31 ไอคอน “USB”.....	57
รูปที่ ก.32 สัญลักษณ์การไม่อนุญาตให้ใช้พอร์ต USB.....	57
รูปที่ ก.33 ไอคอนการส่งข้อความประกาศ.....	58
รูปที่ ก.34 หน้าต่างการประกาศของเครื่องลูกข่าย.....	58
รูปที่ ก.35 การส่งข้อความ ไปยังเครื่องลูกข่ายอื่นๆ.....	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

จากที่ผ่านมามหาวิทยาลัยของการเรียนในห้องปฏิบัติการคอมพิวเตอร์ จะมีนักศึกษาบางคนไม่สนใจในการเรียน ในช่วงที่อาจารย์กำลังบรรยายให้ฟัง ก็มักจะใช้งานคอมพิวเตอร์ทำอย่างอื่น เช่น เข้าเว็บไซต์ต่างๆ หรือไม่ก็นั่งเล่นเกมส์ ทำให้ตัวนักศึกษาเองไม่เข้าใจเนื้อหาที่สอน และทำงานที่ได้รับมอบหมายไม่เสร็จทันเวลา และหลังจากเลิกใช้คอมพิวเตอร์แล้ว นักศึกษาบางรายอาจจะไม่ปิดเครื่องคอมพิวเตอร์ ทำให้ต้องเดินตรวจว่าปิดเครื่องคอมพิวเตอร์ครบแล้วหรือยัง จึงได้มีความคิดว่า จะทำอย่างไรที่จะให้อาจารย์สามารถที่จะควบคุมการใช้งานเครื่องคอมพิวเตอร์ได้ และตรวจเช็คได้ว่าตอนนี้เครื่องคอมพิวเตอร์ทำงานอะไรอยู่ เครื่องปิดหรือเปิดอยู่ ทำให้สะดวกสำหรับอาจารย์ผู้สอน และการควบคุมให้นักศึกษาอยู่ในระเบียบ

ในปัจจุบันมีโปรแกรมจัดการเครื่องคอมพิวเตอร์ หรือที่เรียกว่า Desktop Management ซึ่งเป็นโปรแกรมที่ใช้จัดการเครื่องคอมพิวเตอร์ในระบบเครือข่ายในบริษัท สำนักงาน เพื่อควบคุมการทำงานของเครื่องคอมพิวเตอร์ สามารถตรวจสอบว่าตอนนี้เครื่องคอมพิวเตอร์กำลังทำงานอะไรอยู่ หรือหากเครื่องมีปัญหาผู้ดูแลระบบก็สามารถที่จะแก้ไขได้โดยไม่ต้องเดินมาด้วยตัวเอง ด้วยเหตุนี้จึงได้จัดทำโปรแกรม Desktop Management อย่างง่ายขึ้นมาใช้งานภายในห้องปฏิบัติการคอมพิวเตอร์ โดยจะมีฟังก์ชันการทำงานที่จำเป็น ไม่เทียบเท่ากับโปรแกรม Desktop Management ที่มีในตลาด เพื่ออำนวยความสะดวกในการใช้งานในห้องปฏิบัติการคอมพิวเตอร์แก่อาจารย์ผู้สอน

เมื่อโปรแกรมนี้มาใช้งานในห้องปฏิบัติการคอมพิวเตอร์ คาดว่าจะทำให้การใช้งานภายในห้องปฏิบัติการจะสะดวกสบายขึ้นสำหรับอาจารย์ผู้สอน ทั้งในด้านการควบคุมให้นักศึกษาไม่ใช้งานเครื่องคอมพิวเตอร์นอกเหนือไปจากการเรียนการสอน เพราะสามารถที่จะตรวจสอบได้ว่าแต่ละเครื่องคอมพิวเตอร์กำลังทำงานอะไรอยู่บ้าง อีกทั้งเมื่อเลิกใช้ห้องปฏิบัติการก็สามารถตรวจสอบได้ว่าเครื่องไหนยังเปิดอยู่บ้าง และสามารถสั่งปิดผ่านเครื่อง Server ได้ทันที ซึ่งจะเห็นได้ว่าอาจารย์ผู้สอนจะมีความสะดวกสบายมากขึ้นอย่างเห็นได้ชัด

1.2 วัตถุประสงค์ของโครงการ

1.2.1 พัฒนาโปรแกรม Desktop Management

1.2.2 ศึกษาการเขียน โปรแกรมผ่านเครือข่าย

1.2.3 เพื่อพัฒนาทักษะการเขียน โปรแกรมแบบ Object Oriented

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 ได้รับความรู้ ความเข้าใจเกี่ยวกับการเขียน โปรแกรมผ่านเครือข่าย
- 1.3.2 ได้รับความรู้ ความเข้าใจเกี่ยวกับการเขียน โปรแกรมด้วยภาษา C#
- 1.3.3 สามารถนำโปรแกรมที่พัฒนามาใช้ในห้องปฏิบัติการคอมพิวเตอร์ได้
- 1.3.4 ได้รับความรู้ ความเข้าใจเกี่ยวกับการเขียน โปรแกรมแบบ Object Oriented

1.4 ขอบเขตของโครงการ

- 1.4.1 ศึกษาภาษาที่ใช้ในการเขียน โปรแกรม
- 1.4.2 ศึกษา Desktop Management ที่มีใช้อยู่ในปัจจุบัน
- 1.4.3 ออกแบบ และพัฒนา โปรแกรมที่ใช้สามารถทำ Desktop Management อย่างง่ายได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 Network Protocols

OSI จะเป็นการกำหนดรูปแบบของ Protocol Layers จุดประสงค์ของแต่ละ layer และวิธีการทำงานร่วมกันของแต่ละ layer เพื่อให้เห็นเป็นรูปธรรมมากยิ่งขึ้น ในรูป 2.1 จะเป็นการนำเอาชุดของ Protocol ที่ใช้งานจริง ในที่นี้คือ TCP/IP Protocol Stack มาเปรียบเทียบกับ OSI Model

Application Layer						
Presentation Layer	HTTP	FTP	SMTP	RIP	DNS	
Session Layer						
Transport Layer		TCP		UDP		
Network Layer	ICMP		IP	IGMP		
Data Link Layer	ETHERNET, ATM, Etc.					
Physical Layer						

รูปที่ 2.1 OSI-TCP/IP Protocol Stack

อาจกล่าวได้ว่า TCP/IP Protocol Stack เป็นรูปแบบอย่างง่ายของ OSI Model กล่าวคือสามารถมองเป็น 4 Layers ได้ โดยในชุดของ TCP/IP Protocol จะพบว่า Protocol ในชั้นบนสุด (HTTP, FTP, SMTP, RIP, DNS, etc.) นั้นจะไม่สามารถจัดอยู่ใน Layer เพียง Layer เดียวได้ ทั้งนี้เพราะ Functions การทำงานของ Protocol เหล่านั้นครอบคลุมทั้ง 3 Layers ของ OSI Model

สำหรับ Network Protocol นี้ยังสามารถแบ่งออกเป็นประเภทต่าง ๆ ตามรูปแบบการทำงานและจุดประสงค์การใช้งานได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 Basic Protocols

หากดูในรูป 2.1 แล้วจะพบว่า Basic Protocol เหล่านี้จะเทียบเคียงได้กับการทำงานใน Layer 3 และ 4 ซึ่งเป็น Layer สำคัญมีหน้าที่หลักในการควบคุมดูแลการสื่อสารข้อมูล โดย Basic Protocols นี้ได้แก่ IP, TCP, UDP, ICMP และ IGMP

2.1.2 Internet Protocols

เป็น Protocol ในระดับบนสุดของ TCP/IP Protocol Stack เป็นชุดของ Protocol ที่นิยมใช้งานกันอย่างแพร่หลายบน Internet ได้แก่ FTP, HTTP และ HTTPS

2.2 Sockets

2.2.1 Overview

โดยทั่วไปในการติดต่อสื่อสารที่เป็นที่นิยมใช้กันในปัจจุบัน จะกระทำบนพื้นฐานของ Protocol TCP/IP และชุดของ Software ซึ่งทำงานกับ Packets ตามมาตรฐานของ TCP/IP (ซึ่งต่อไปจะเรียกว่า TCP/IP Protocol Software) นั้นจะถูกสร้างขึ้นมาเป็นส่วนหนึ่งของตัวระบบปฏิบัติการ ดังนั้น เมื่อ Application Software ใด ๆ ที่ต้องการสื่อสารผ่านระบบเครือข่ายตามมาตรฐาน TCP/IP Protocol จะต้องติดต่อกับระบบปฏิบัติการเพื่อขอรับบริการ

สำหรับทางผู้พัฒนามาตรฐาน TCP/IP Protocol ไม่ต้องการให้ TCP/IP Protocol ใช้งานได้กับ Platform ใดเป็นการเฉพาะ หรือใช้งานได้กับเพียงระบบปฏิบัติการระบบใดระบบหนึ่ง ดังนั้น ผู้พัฒนาจึงมีความระมัดระวังในการกำหนดมาตรฐานมิให้มีการอ้างอิงข้อมูลการเชื่อมต่อภายใน ซึ่งเป็นรูปแบบเฉพาะของ Platform ใด Platform หนึ่ง หรืออ้างอิงวิธีการเชื่อมต่อกับ Application Software ในลักษณะหรือรูปแบบเฉพาะของระบบปฏิบัติการจากผู้ผลิตรายใดรายหนึ่ง และด้วยแนวคิดในการออกแบบที่ไม่ยึดติดกับ Platform หรือระบบปฏิบัติการใด ๆ นี้ทำให้อาจกล่าวได้ว่า Protocol TCP/IP มีลักษณะเป็น Loosely Specified Protocol Software Interface

จากที่กล่าวมาข้างต้นสามารถกล่าวโดยสรุปได้ว่า “มาตรฐาน TCP/IP นั้นมิได้ระบุรายละเอียดเกี่ยวกับการเรียกใช้งาน TCP/IP Protocol Software ของ Application Software หากแต่มีเพียงข้อเสนอแนะเกี่ยวกับ functions ที่ควรจะมีในชุดของ TCP/IP Protocol Software เท่านั้น ส่วนของรายละเอียดที่นอกเหนือจากนั้นจะเปิดกว้างให้นักออกแบบระบบกำหนดและลงรายละเอียดเอง”

ข้อดีของการออกแบบ TCP/IP ให้เป็นแบบ Loosely Specified Protocol Software Interface นั้นคือ ความ Flexible และ Tolerance ทั้งนี้เพราะการไม่กำหนดรายละเอียดปลีกย่อยทำให้นักออกแบบระบบสามารถที่จะใส่ TCP/IP ลงในระบบปฏิบัติการได้โดยง่ายไม่ว่าระบบปฏิบัติการนั้นจะเป็นระบบปฏิบัติการแบบง่าย ๆ อย่างที่ใช้กันใน Embedded System ไปจนถึงระบบปฏิบัติการที่สลับซับซ้อนอย่างในระบบปฏิบัติการที่ใช้ในเครื่อง Super Computer เป็นต้น นอกจากนั้นด้วยการที่ไม่กำหนดวิธีการเชื่อมต่อกับ Application Software ทำให้นักออกแบบระบบสามารถเลือกใช้เอกสารที่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักผู้ดำเนินการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบในการเชื่อมต่อได้อย่างอิสระ โดยสามารถใช้ได้ทั้งแบบ Procedural หรือ Message-Passing เป็นต้น แต่ทั้งนี้ Loosely Specified Software Interface นั้นก็ยังมีข้อด้อยเช่นกันกล่าวคือการทำงานที่หนักออกแบบระบบมีอิสระในการใส่รายละเอียดในส่วนการเชื่อมต่อ จึงทำให้เกิดความหลากหลายของรูปแบบการเชื่อมต่อ ซึ่งนั่นทำให้การเขียน Application Program เพื่อการใช้งาน TCP/IP มีความยุ่งยากและซับซ้อนมากยิ่งขึ้น อีกทั้งยังทำให้ Application Program นั้นยากแก่การ Port ไปยังระบบปฏิบัติการหรือ Platform อื่น ๆ

แม้ว่านักออกแบบระบบจะสามารถใส่ส่วนการเชื่อมต่อกับซอฟต์แวร์ (Software Interface) ใดๆก็ได้ แต่ในทางปฏิบัติกลับมีรูปแบบของ TCP/IP Software Interface อยู่เพียงไม่กี่แบบเท่านั้น และ TCP/IP Software Interface ที่นิยมมีอยู่เพียง 2 รูปแบบเท่านั้นคือ

2.2.1.1 Socket Interface เป็น TCP/IP Software Interface ที่พัฒนาขึ้น โดย University of California at Berkley ซึ่งในตอนต้นได้พัฒนารูปแบบการเชื่อมต่อแบบนี้เพื่อใช้กับระบบปฏิบัติการ Berkley Unix

2.2.1.2 Transport Layer Interface หรือ TLI เป็น TCP/IP Software Interface ที่พัฒนาโดย AT&T เพื่อใช้งานกับระบบปฏิบัติการ System V Unix

2.2.2 Interface Functionality

แม้ว่าทางผู้พัฒนา TCP/IP Protocol Software Interface จะไม่ได้มีการกำหนดรูปแบบหรือวิธีการใด ๆ ในการเชื่อมต่อ Application Software กับ TCP/IP Protocol Software แต่ทางผู้พัฒนาเองก็ได้มีข้อเสนอแนะเกี่ยวกับ Functions ต่าง ๆ ที่จำเป็น เป็นต้นว่า Interface จะต้องสามารถรองรับการทำงานต่าง ๆ ตาม concept ดังต่อไปนี้

- Allocate local resources for communication
- Specify local and remote communication end-points
- Initiate a connection (client side)
- Wait for an incoming connection (server side)
- Send or receive data
- Determine when data arrives
- Generate urgent data
- Handle urgent data
- Terminate a connection gracefully
- Handle connection terminate from the remote site
- Abort communication
- Handle error condition or a connection abort
- Release local resources when communication finishes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจาก Concept ในการทำงานดังที่ได้กล่าวมาแล้วข้างต้น ทางผู้พัฒนามาตรฐาน TCP/IP Protocol ได้สร้าง Conceptual Interface ขึ้นมาเพื่อเป็นแนวทางให้นักออกแบบระบบสามารถพัฒนาซอฟต์แวร์ของตนได้ง่ายและเร็วขึ้น โดย Conceptual Interface ที่ทางผู้พัฒนามาตรฐาน TCP/IP Protocol ได้สร้างขึ้นนั้นจะอยู่ในรูปของ Set of Procedures and Functions เหตุที่ Conceptual Interface อยู่ในรูปแบบของ Procedures และ Functions นั้น เนื่องจากในช่วงเวลาที่พัฒนา Conceptual Interface ระบบปฏิบัติการต่าง ๆ ในยุคนั้นนิยมใช้วิธีการซึ่งเรียกกันว่า Procedural Mechanism ในการเคลื่อนย้ายการควบคุม จาก Application Software ไปยังระบบปฏิบัติการ ดังนั้นจึงอาจกล่าวโดยสรุปเกี่ยวกับ Conceptual Interface ได้ดังนี้ “Conceptual Interface ซึ่งถูกกำหนดโดยผู้พัฒนามาตรฐาน TCP/IP นั้นจะไม่ระบุรูปแบบของการแทนข้อมูล หรือ รายละเอียดในการเขียนโปรแกรม หากแต่แสดงเพียงตัวอย่างหนึ่งของการเชื่อมต่อ ซึ่งมีความเป็นไปได้ในทางปฏิบัติที่ระบบปฏิบัติการจะสามารถให้บริการแก่ Application Software ที่ต้องการใช้งาน TCP/IP Protocol”

2.2.3 Socket Types

Sockets สามารถแบ่งออกเป็นประเภทตามลักษณะการรับส่งข้อมูลผ่านเครือข่ายได้เป็น 3 ประเภท ดังนี้

- Stream Sockets หรือ Connection Oriented Sockets
- Datagram Sockets หรือ Connectionless Sockets
- Raw Sockets

2.2.4 Working with Sockets in .Net

ใน .Net การทำงานที่เกี่ยวข้องกับ Sockets จะต้องเรียกใช้งาน Namespace ที่ชื่อว่า System.Net.Sockets ซึ่งเป็น Namespace ที่บรรจุ Classes ต่าง ๆ ที่สนับสนุนการทำงานกับ Sockets โดย Classes ที่เกี่ยวข้องกับการทำงานหลัก ๆ กับ Sockets มีดังนี้

- MulticastOption
- NetworkStream
- TcpClient
- TcpListener
- UdpClient
- SocketException
- Socket

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Class Socket จะมี function พื้นฐานต่าง ๆ ที่จำเป็นสำหรับการสร้าง Application ซึ่งใช้งาน Sockets สำหรับ Properties ที่สำคัญสำหรับ System.Net.Sockets.Socket ได้แก่

- AddressFamily
- Available
- Blocking
- Connected
- LocalEndPoint
- ProtocolType
- RemoteEndPoint
- SocketType

และ Method ที่สำคัญของ System.Net.Sockets.Socket มีดังนี้

- Accept ()
- Bind ()
- Close ()
- Connect ()
- GetSocketOption ()
- IOControl ()
- Listen ()
- Receive ()
- Poll ()
- Select ()
- Send ()
- SetSocketOption ()
- Shutdown ()

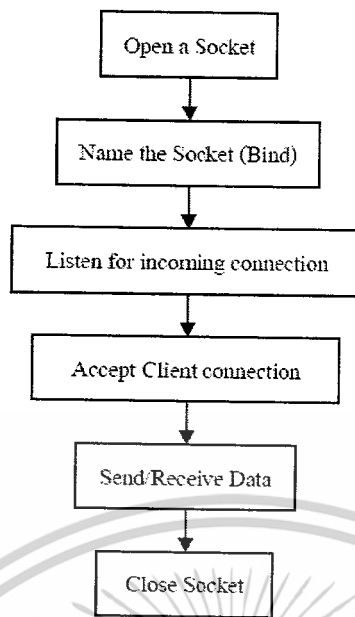
2.2.5 Working with Sockets

2.2.5.1 Passive Sockets

Passive Sockets หรือ Sockets ซึ่งใช้ใน Server Application นั้นมีขั้นตอนการสร้าง

ดังนี้

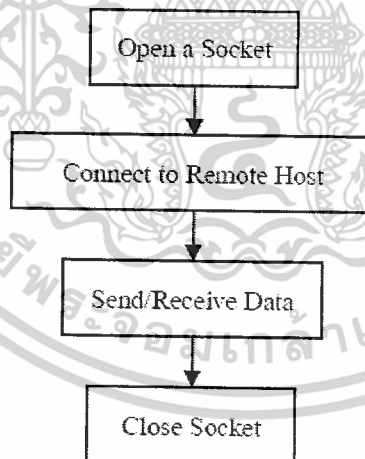
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 Diagram ขั้นตอนการสร้าง Passive Sockets

2.2.5.2 Active Sockets

Active Sockets หรือ Socket ประเภทที่ใช้งานใน Client Application มีขั้นตอนการสร้างดังนี้



รูปที่ 2.3 Diagram ขั้นตอนการสร้าง Active Sockets

2.3 Stream Sockets (Connection Oriented Socket)

Stream Socket หรือ Connection Oriented Socket เป็น Socket ที่ใช้ Protocol TCP ในการสื่อสารข้อมูล โดยลักษณะการทำงานจะแยกออกเป็น 2 ฝั่งคือ Server และ Client ซึ่งขั้นตอนการเปิด Socket ใน Application ของทั้ง 2 ฝั่งนั้นจะมีความแตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 ข้อควรระวังในการใช้งาน Stream Socket

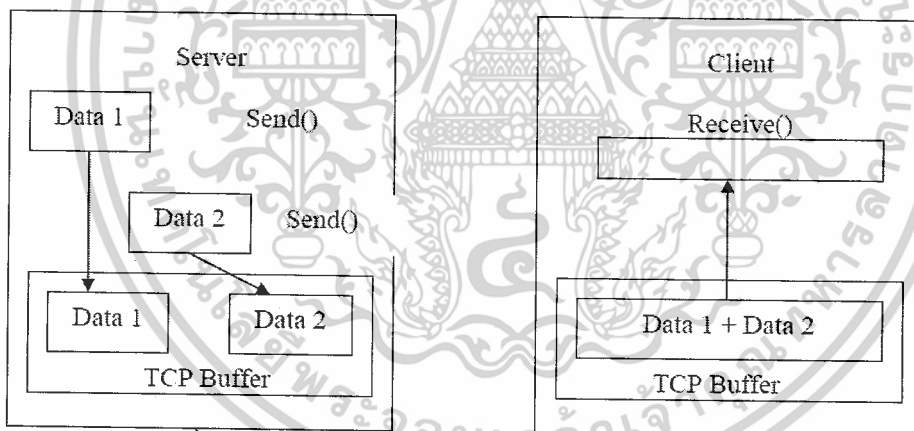
ในการใช้งาน Stream Socket หรือ Connection Oriented Socket นั้น มีข้อควรระวังอยู่ 2 ประการ ได้แก่

- การจัดการ buffer ในโปรแกรมไม่ดีพอ (Improper buffer handling)
- การจัดการข้อมูลที่ส่งผ่านระหว่างเครือข่ายไม่ดีพอ (Improper message handling)

โดยข้อควรระวังทั้ง 2 ข้อดังกล่าวนี้สามารถแก้ไขได้โดยใช้เทคนิคในการเขียนโปรแกรมเข้ามาช่วยได้ ดังนี้

2.3.1.1 ปัญหาในการจัดการ Buffer

สำหรับ Buffer ในการทำงานกับ Stream Socket นั้นจะแบ่งเป็น 2 ส่วน ส่วนแรกเป็น Buffer ของระบบ (เป็น Buffer ของ TCP Protocol) ซึ่งเป็น Buffer ที่จะรับเอาข้อมูลซึ่งส่งผ่านโครงข่ายไว้เพื่อรอโปรแกรมใช้คำสั่ง Receive() เพื่ออ่านออกไป สำหรับ Buffer ในส่วนที่ 2 นั้นจะเป็น Buffer ซึ่งเกิดจากการที่นักเขียนโปรแกรมสร้างตัวแปรขึ้นมาเพื่อให้รองรับข้อมูลที่อ่านจาก TCP Buffer โดยคำสั่ง Receive() ซึ่งปัญหา Improper Buffer Handling นั้นจะเกิดกับ Buffer ส่วนที่ 2 นี้แน่นอนโดยทั่วไปโครงสร้างของ TCP Buffer นั้นจะมีลักษณะดังนี้



รูปที่ 2.4 แบบจำลอง TCP Buffer

สำหรับในฝั่งของ Server นั้น TCP Buffer จะมีไว้เพื่อรองรับการ re-transmit ในกรณีที่ Packet ที่ส่งไปนั้นเกิดการสูญหายไปนในโครงข่าย โดยข้อมูลใน Buffer จะถูกลบออกไปก็ต่อเมื่อฝั่ง Client ตอบกลับมาว่าได้รับข้อมูลนั้นแล้วตามกลไกของ TCP Protocol ส่วน TCP Buffer ในฝั่งของ Client นั้นมีไว้เพื่อรับเอาข้อมูลที่ส่งมาทางโครงข่ายไว้รอการนำออกไปใช้งาน โดยคำสั่ง Receive()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการอ่านข้อมูลจาก TCP Buffer โดยใช้คำสั่ง Receive () นั้น ปริมาณของข้อมูล ซึ่งอ่านในแต่ละครั้งจะขึ้นอยู่กับ 2 ปัจจัย ได้แก่

- ขนาดของ data buffer ซึ่งใส่เป็น parameter ให้กับคำสั่ง Receive ()
- ขนาดของตัวแปรซึ่งใช้เป็น data buffer ซึ่งใส่เป็น parameter ให้กับคำสั่ง Receive ()

ในกรณีของโปรแกรมตัวอย่างทั้ง 2 โปรแกรมนั้นได้กำหนดตัวแปร data ให้เป็นแบบ byte array และมีขนาด 1024 ไบต์ และในการใช้งานคำสั่ง Receive () ในโปรแกรมตัวอย่างทั้ง 2 นั้น parameter เดียวที่ใส่ให้กับคำสั่ง Receive ได้แก่ ตัวแปรซึ่งใช้เป็น data buffer ในที่นี้คือตัวแปร data ดังนั้นในโปรแกรมตัวอย่างทั้ง 2 อาจกล่าวได้ว่าปริมาณในการอ่านข้อมูลจาก TCP Buffer แต่ละครั้งจะมีขนาดมากที่สุดไม่เกินครั้งละ 1024 ไบต์ หรือก็คือ ขนาดของตัวแปรที่ใช้เป็น data buffer นั้นคือขนาด default ของการอ่านข้อมูลจาก TCP Buffer โดยคำสั่ง Receive () นั่นเอง

เมื่อขนาดของ Data buffer มีผลกระทบต่อกรอ่านข้อมูลจาก TCP Buffer ในแต่ละครั้งดังที่กล่าวมาแล้วจะได้ว่าถ้าการกำหนดขนาดของ data buffer ไม่เหมาะสมก็จะมีผลต่อการทำงานของโปรแกรมได้ โดยในการกำหนดขนาดของ data buffer ไม่เหมาะสมนั้นกล่าวได้ 2 นัยคือ กำหนดขนาดไว้มากเกินไป และกำหนดขนาดไว้น้อยเกินไป

2.3.1.2 ปัญหาด้านการจัดการข้อมูลที่ส่งผ่านระหว่างเครือข่าย

ในการสื่อสารข้อมูลระหว่างเครือข่ายนั้นสิ่งที่จะเกิดขึ้นได้คือ การสูญหายหรือความล่าช้าของข้อมูลที่ส่งจากต้นทางไปยังปลายทาง แม้ว่า TCP Protocol นั้นจะรับประกันในเรื่องของการส่งข้อมูลว่าจะไม่มีเรื่องของการสูญหายระหว่างกรรับส่ง หรือหากเกิดการสูญหายของข้อมูลขึ้น ข้อมูลชุดที่สูญหายนั้นก็จะถูกส่งใหม่ (retransmit) แต่ทั้งนี้แม้ข้อมูลจะไม่สูญหายแต่ TCP Protocol นั้นเมื่อรับข้อมูลเข้าสู่ TCP Buffer แล้วนั้นจะทำการเรียงลำดับข้อมูล แล้วรวมข้อมูลที่รับมานั้นเข้าด้วยกัน (ในกรณีที่ข้อมูลนั้นมีขนาดใหญ่แล้วถูกแบ่งออกเป็น ส่วน ๆ แล้วทยอยส่งมาทีละส่วน) หรือในกรณีที่ข้อมูลมีขนาดเล็กก็จะนำข้อมูลนั้นมาเก็บเรียงต่อกันไปภายใน TCP Buffer โดยไม่มีการทำเครื่องหมายเพื่อระบุถึงขอบเขตของข้อมูลแต่ละชุด

จะเห็นได้ว่า TCP Protocol นั้นไม่มีการป้องกันขอบเขตของแต่ละชุดข้อมูล (Unprotected Message Boundary) ดังนั้นในการเขียนโปรแกรมเพื่อสื่อสารข้อมูลด้วย TCP Protocol จึงต้องออกแบบโปรแกรมให้สามารถรองรับปัญหานี้ด้วย สำหรับวิธีการแก้ปัญหาดังกล่าวนั้น โดยทั่วไปนิยมใช้กันอยู่ 3 วิธี ได้แก่

- การจำกัดขนาดของแต่ละชุดข้อมูลที่จะส่งให้มีขนาดเท่ากัน
- การส่งขนาดของชุดข้อมูลไปพร้อมกันกับชุดข้อมูลนั้น
- การใส่สัญลักษณ์เพื่อบอกขอบเขตของข้อมูลแต่ละชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในแต่ละวิธีการที่ใช้แก้ปัญหานั้นก็ยังมีทั้งข้อดีและข้อเสียในตัวเอง ดังที่จะได้กล่าวถึงต่อไป

1. การจำกัดขนาดของแต่ละชุดข้อมูลที่จะส่งให้มีขนาดเท่ากัน

เป็นวิธีการแก้ปัญหาที่ง่ายที่สุด เพราะเมื่อขนาดของข้อมูลที่จะส่งแต่ละชุดมีขนาดเท่ากัน โปรแกรมจะทราบได้ทันทีว่าข้อมูลมาครบแล้วหรือยัง โดยการตรวจสอบขนาดของชุดข้อมูลนั้นนั่นเอง นอกจากนี้แม้ชุดข้อมูลมากกว่า 1 ชุดมาถึงพร้อม ๆ กันก็ยังสามารถแยกชุดข้อมูลออกจากกันได้โดยการนับจำนวนไบต์ในข้อมูลที่รับเข้ามานั้นเทียบกับขนาดมาตรฐานของชุดข้อมูลปกติ

2. การส่งขนาดข้อมูลไปพร้อมกันกับชุดข้อมูลนั้น

จากหัวข้อที่แล้วการส่งข้อมูลด้วยการจำกัดขนาดของข้อมูลที่จะส่งให้มีขนาดเท่ากันทุก ๆ packet นั้นจะสามารถแก้ปัญหาในเรื่องของ message boundaries ได้ แต่ข้อเสียของวิธีการดังกล่าวคือ ถ้าข้อมูลที่จะส่งนั้นมีขนาดที่ยาวมาก ก็จำเป็นที่จะต้องแบ่งข้อมูลนั้นออกเป็น ส่วน ๆ ที่เท่า ๆ กัน และทำการส่งข้อมูลออกไปครั้งละ 1 ชุด ซึ่งการส่งข้อมูลออกไปหลาย ๆ ชุด แทนที่จะเป็นชุดเดียวยาว ๆ เช่นที่ยกตัวอย่างมาจะมีผลให้เกิดการสิ้นเปลือง bandwidth ของระบบสื่อสารโดยใช้เหตุด้วยเช่นกัน

เพื่อแก้ปัญหานี้จึงได้มีการคิดค้นวิธีการที่จะให้สามารถส่งชุดข้อมูลที่มีขนาดต่างกันก็ได้ แต่การจะส่งชุดข้อมูลซึ่งมีขนาดแตกต่างกันไปได้นั้น ทางฝั่งรับจะต้องทราบขนาดของชุดข้อมูลที่จะส่งไปด้วย ซึ่งก็สามารถทำได้โดยการส่งขนาดของชุดข้อมูลไปรวมไปกับชุดข้อมูลที่จะส่งนั้น ตัวอย่างเช่น

9message 1

ในที่นี้ขนาดของชุดข้อมูลที่จะส่งไปด้วยคือ 9 สำหรับตัวข้อมูลคือ message 1 แต่การใช้ตัวอักษรเพื่อบอกขนาด ถ้าชุดข้อมูลมีขนาดใหญ่ก็ต้องใช้หลายไบต์ในการบอกขนาด (1 ตัวอักษรต่อ 1 ไบต์) เพื่อลดปริมาณข้อมูลที่จะส่ง โดยทั่วไปจึงไม่ใช้ตัวอักษรในการบอกขนาดแต่จะใช้ข้อมูลชนิดไบต์ในการบอกขนาดแทนเช่นถ้าใช้ 2 ไบต์แรกของชุดข้อมูล จะได้ว่าชุดข้อมูลนั้นสามารถมีขนาดได้มากที่สุด 216 ไบต์ หรือ 64 กิโลไบต์นั่นเอง

3. การใส่สัญลักษณ์เพื่อบอกขอบเขตของข้อมูลแต่ละชุด

เป็นวิธีการแก้ไขปัญหากับ Message boundaries วิธีสุดท้าย เป็นวิธีการแก้ปัญหาโดยการใส่สัญลักษณ์ที่ได้กำหนดไว้ล่วงหน้าเพื่อบอกจุดสิ้นสุดของข้อมูลแต่ละชุด เมื่อรับข้อมูลเข้ามาโปรแกรมจะต้องทำการตรวจข้อมูลนั้นทีละตัวอักษรเพื่อหาสัญลักษณ์บอกจุดสิ้นสุด สำหรับวิธีการนี้มีข้อดีอยู่ 2 ประการคือ ต้องกำหนดสัญลักษณ์ที่ใช้ระบุขอบเขตของชุดข้อมูล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะต้องไม่ซ้ำกันกับข้อมูลที่จะส่ง และการที่จะต้องตรวจสอบข้อมูลที่ได้รับเข้ามาทีละ 1 ตัวอักษร ก็จะทำให้เกิด Overhead ขึ้น

ใน C# มี Classes ซึ่งช่วยในการทำงานในลักษณะของการใส่สัญลักษณ์เพื่อบอกรูปแบบของข้อมูล โดยที่นักเขียนโปรแกรมไม่ต้องเขียน code เพื่อทำการตรวจสอบตัวอักษรทีละตัวเพื่อหาสัญลักษณ์

Classes ซึ่งช่วยทำงานดังกล่าวได้แก่ Class ชื่อ NetworkStream ซึ่งทำหน้าที่เป็นส่วนเชื่อมต่อกับ Socket, Class ชื่อ StreamReader และ StreamWriter ทำหน้าที่อ่าน/เขียนข้อมูลชนิด text ที่ได้จาก NetworkStream

2.4 Datagram Sockets (Connectionless Sockets)

Connectionless Sockets เป็น Socket ซึ่งใช้งานกับ UDP Protocol โดยที่การส่งข้อมูลโดยผ่าน Connectionless Socket/UDP นี้ไม่จำเป็นต้องสร้างเส้นทางการเชื่อมต่อเสมือน กล่าวเมื่อต้องการจะส่งข้อมูลโปรแกรมก็จะทำการส่งข้อมูลนั้นออกไปเลย โดยไม่มีการตรวจสอบสถานะเส้นทางไปยังฝั่งที่จะรับ ดังนั้นการสูญหายของข้อมูลจึงอาจเกิดขึ้นได้

จากคุณลักษณะของ UDP ที่ไม่รับประกันในเรื่องของข้อมูลสูญหายระหว่างทางและการเรียงลำดับของข้อมูล ดังนั้นในโปรแกรมที่ใช้งานกับ Connectionless Socket/UDP จะต้องมีส่วนของโปรแกรมที่ใช้จัดการเกี่ยวกับปัญหาดังกล่าว จึงอาจกล่าวได้ว่า Connectionless Socket นั้นหากมองในด้านการรับและส่งข้อมูลสามารถทำได้ง่ายและมีกระบวนการที่ซับซ้อนน้อยกว่า Connection Oriented Socket เป็นอย่างมาก หากถ้ามองด้านการจัดการข้อมูลซึ่งได้รับเข้ามาแล้วจะพบว่าต้องมีกระบวนการรองรับที่ซับซ้อนมากกว่าการใช้งาน Connection Oriented Socket เนื่องจากการจัดการเกี่ยวกับข้อมูลที่รับเข้ามาหลาย ๆ อย่าง ไม่มีการรองรับโดยตัว UDP Protocol

2.4.1 การสร้าง Connectionless Socket

ในการสร้าง Connectionless Socket นั้นมีกระบวนการอยู่ 2 ขั้นตอนคือ

- สร้าง Socket Object
- Bind the socket to a local IPEndPoint

หลังจากเสร็จสิ้น 2 ขั้นตอนนี้แล้ว Socket ที่สร้างขึ้นนั้นสามารถใช้รับ-ส่งข้อมูลได้ทันที ซึ่งจะพบว่าขั้นตอนน้อยกว่าการสร้าง Connection Oriented Socket และกระบวนการต่างๆ ที่ใช้เพื่อการเชื่อมต่อโดย TCP นั้นไม่มีความจำเป็นเลยสำหรับการใช้งาน UDP นอกจากนี้สำหรับโปรแกรมทำการสื่อสารโดยใช้ Connectionless Socket และทำหน้าที่เป็นผู้ส่งเพียงอย่างเดียวนั้นสามารถตัดขั้นตอนการ bind socket ออกไปได้ เพียงสร้าง Socket แล้วส่งข้อมูลทาง Socket ที่สร้างขึ้นได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาเนื่องจากลักษณะการจัดการข้อมูล/การเชื่อมต่อที่แตกต่างออกไปของ UDP คำสั่ง Send() และ Receive() จึงไม่สามารถนำมาใช้งานกับ Connectionless Socket ได้ หากแต่ต้องใช้คำสั่งที่ถูกสร้างขึ้นเพื่อรองรับการเชื่อมต่อชนิดนี้คือ SendTo() และ ReceiveFrom()

2.4.1.1 SendTo()

เป็น method ซึ่งใช้ในการส่งข้อมูลไปยังเครื่องปลายทาง โดยการใช้ SendTo() นั้นสามารถใส่ parameter ได้ในหลายลักษณะ ดังนี้

SendTo(byte[] data, EndPoint Remote)

รูปแบบข้างต้นเป็นรูปแบบอย่างง่ายของ SendTo() คือส่งข้อมูลในรูปแบบของ byte array ไปยังปลายทางซึ่งถูกระบุไว้ในตัวแปร Remote นั้นเอง

SendTo(byte[] data, SocketFlags Flags, EndPoint Remote)

สำหรับรูปแบบนี้จะรับ parameter SocketFlags เพิ่มเข้ามา ซึ่ง SocketFlags นี้จะใช้เพื่อระบุ UDP Socket Option ที่จะเลือกใช้งาน

SendTo(byte[] data, int Size, SocketFlags Flags, EndPoint Remote)

ในรูปแบบนี้จะมีการระบุจำนวนไบต์ที่ต้องการจะส่งเพิ่มเข้ามา โดยจำนวนไบต์ที่ต้องการจะส่งนั้นจะระบุลงในตัวแปรชนิด integer ชื่อ Size

SendTo(byte[] data, int Offset, int Size, SocketFlags Flags, EndPoint Remote)

ในรูปแบบสุดท้ายของ SendTo() นี้จะสามารถระบุตำแหน่ง offset ใน byte array data เพื่อกำหนดตำแหน่งที่จะเริ่มทำการส่งผ่านข้อมูลผ่านตัวแปรชนิด integer ชื่อ Offset นั้นเอง

2.4.1.2 ReceiveFrom()

เป็น Method ที่ใช้สำหรับการรับข้อมูล โดยจะรับ parameter 2 ตัวได้แก่ byte array ที่ทำหน้าที่เป็น data buffer และ EndPoint ของเครื่องปลายทาง โดยมีรูปแบบของการใช้งาน ReceiveFrom() เป็นดังนี้

ReceiveFrom(byte[] data, ref EndPoint Remote)

สำหรับการใช้งาน ReceiveFrom() นั้นมีข้อสังเกตที่ parameter ตัวที่ 2 จะพบว่าค่าที่จะส่งให้มันเป็นแบบ reference กล่าวคือสิ่งที่ส่งไปนั้นจะเป็นเพียง address ตำแหน่งที่เก็บ EndPoint Object โดยที่ ReceiveFrom() นั้นจะทำการใส่ค่า EndPoint ของฝ่ายส่งที่ตำแหน่ง address ที่ส่งมาเป็นค่า parameter ของ ReceiveFrom()

2.4.2 การใช้งาน Connect() ร่วมกับ Connectionless Socket

ในบางครั้งโปรแกรมฝั่ง client นั้นต้องการติดต่อกับ server เพียงเครื่องเดียว ดังนั้นการที่จะต้องระบุ EndPoint ของเครื่องปลายทางในทุก ๆ ครั้งที่ส่งข้อมูล หรือการที่จะต้องเตรียมตัวแปรไว้รองรับค่า EndPoint ฝั่งส่งเพื่อใช้กับคำสั่ง ReceiveFrom() นั้นค่อนข้างจะเป็นสิ่งที่ไม่จำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน C# เมื่อต้องการใช้งาน Connectionless Socket โดยใช้รับ – ส่งข้อมูลกับเครื่องปลายทางเพียงเครื่องเดียวนั้น สามารถใช้คำสั่ง Connect() เข้ามาช่วยได้ ซึ่งจะทำให้การทำงานต่างๆ ลดความยุ่งยากลง

2.4.3 ข้อดีของ Connectionless Sockets

เมื่อเปรียบเทียบ Connectionless Sockets กับ Connection Oriented Sockets (UDP vs. TCP) จะได้ว่า Connectionless Sockets นั้นมีข้อดีกว่าดังต่อไปนี้

- Lack of Handshaking Signals
- No Support for sessions
- No guarantee of sequenced data delivery
- Reduced security
- Lack of flow control

2.4.4 ข้อดีของ Connectionless Sockets

ด้วยข้อกำหนดของ UDP ให้ message ทุก message ที่รับเข้ามานั้นไม่สูญเสียขอบเขตของตัว message ดังเช่นที่เกิดขึ้นใน TCP (ส่วนหนึ่งเนื่องจาก UDP ไม่ต้องมี Buffer ในการรับส่งและ Socket แบบ Connectionless นี้เมื่อถูกสร้างขึ้นมาแล้วสามารถรับข้อมูลจากเครื่องใดก็ได้ ดังนั้นลักษณะเฉพาะของตัว UDP Packet เองจะต้องบ่งบอกถึงผู้ส่งและขอบเขตของ message นั้น) โดยความสามารถในการระบุขอบเขตของ message นี้สามารถทดลองได้โดยเรียกใช้งานตัวอย่างโปรแกรม 4.5 และ 4.6 แล้วสังเกตผลที่ได้ นอกจากนี้ UDP ยังมีข้อดีที่เหนือกว่า TCP ในเรื่องต่างๆ ดังนี้ เมื่อใช้ Connect() จะทำให้สามารถใช้คำสั่ง Send() และ Receive() ได้อีกด้วยโดยที่การสื่อสารต่าง ๆ จะยังคงใช้ UDP Packet เช่นเดิม การใช้งาน Connect()

- Low overhead requirements
- Speed
- Topology Support
- Smaller header size

2.4.5 ข้อควรระวังในการใช้งาน Connectionless Socket

ในการใช้งาน Connectionless Socket มีข้อพึงระวังอยู่ 2 ข้อด้วยกันคือ

- การสูญหายของข้อมูลเนื่องจากข้อจำกัดบางประการของ method ชื่อ ReceiveFrom()
- Detecting and allowing for lost packets

ซึ่งข้อพึงระวังดังกล่าวหากในการเขียน โปรแกรมเพื่อสื่อสารข้อมูลผ่าน Connectionless Socket แล้วไม่มีการระวัง (aware) ที่ดีพอ ก็อาจทำให้เกิดการทำงานของ โปรแกรมที่ผิดเพี้ยนไปจากวัตถุประสงค์ที่ตั้งไว้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5.1 การป้องกันข้อมูลสูญหาย

ในโปรแกรมจะเป็นการกำหนด Read Buffer โดยกำหนดเป็น byte array หากข้อมูลที่ได้รับเข้ามาทาง socket มีขนาดน้อยกว่าหรือเท่ากับขนาดที่กำหนด โปรแกรมก็จะทำงานได้อย่างราบรื่น หากข้อมูลที่ได้รับเข้ามาทาง socket นั้นมีขนาดใหญ่ขนาดที่กำหนด สิ่งที่เกิดขึ้นคือ Exception Error ซึ่ง Exception Error ดังกล่าวจะเกิดเมื่อคำสั่ง ReceiveFrom() ทำการอ่านข้อมูลเข้ามาไว้ใน Read Buffer แล้วปรากฏว่าไม่สามารถอ่านข้อมูลทั้งหมดมาใส่ไว้ใน Read Buffer ได้ กล่าวคือยังมีข้อมูลค้างอยู่ที่ไม่สามารถนำมาได้นั่นเอง ReceiveFrom() ก็จะทำการเตือนระบบเพื่อให้รับทราบถึงสถานการณ์ดังกล่าว และเนื่องจาก UDP ไม่มี Buffer แบบ TCP ดังนั้นข้อมูลใดซึ่งอ่านออกมาไม่หมดก็จะหายไป ดังนั้นเมื่อมีการตรวจพบการสูญหายในลักษณะดังกล่าวจึงต้องให้อีกฝั่งที่ส่งข้อมูลมาทำการส่งข้อมูลใหม่เท่านั้น

สำหรับปัญหาในเรื่องของ Exception Error นั้นสามารถแก้ไขได้โดยการนำเอา Try...catch มาประยุกต์ใช้งาน

นอกจากการทำ Try...Catch แล้ว เมื่อเกิด Exception (Buffer Overflow) ขึ้นจะทำการเพิ่มขนาด buffer ให้ใหญ่ขึ้นไปอีก 10 bytes โดยอัตโนมัติ ซึ่งจะสามารถทำได้โดยการกำหนดขนาดของ byte array ให้อยู่ในรูปของตัวแปร i

2.4.5.2 การป้องกัน Packets Lost

ปัญหาหลักอย่างหนึ่งของการทำงานกับ Connectionless Socket ก็คือการการสูญหายของข้อมูลที่ส่งไปบนเน็ตเวิร์ค ทั้งนี้เนื่องจากการส่งข้อมูลนั้นไม่มีการสร้าง virtual connection ทำให้ไม่สามารถทราบได้ว่า packet นั้นส่งถึงปลายทางแล้วหรือไม่ ดังนั้นจึงได้มีการคิดค้นวิธีการเพื่อชดเชยข้อด้อย/แก้ปัญหของ UDP ในส่วนนี้ขึ้น

- การใช้ Sockets Time-outs

ดังนั้นสำหรับ โปรแกรมบางลักษณะซึ่งการสูญหายของ Packet มีผลกระทบต่อโปรแกรมนั้นเป็นอย่างมาก จึงมีการออกแบบวิธีการทำงานของโปรแกรมเพื่อแก้ปัญหาดังกล่าว โดยการสร้างกระบวนการรับส่งให้ เป็นไปในลักษณะที่ว่า เมื่อมีการส่งข้อมูลออกไป หากเครื่องปลายทางได้รับข้อมูลนั้นแล้วจะทำการสร้าง packet เพื่อตอบรับกลับมา (Acknowledge) หากไม่มีการตอบกลับมาในช่วงเวลาที่กำหนดก็จะทำการส่งข้อมูลนั้นซ้ำอีกครั้งหนึ่ง โดยการจะทำเช่นนั้นได้มีขั้นตอนดังนี้

- กำหนดค่า time-outs ให้กับ Socket ที่สร้างขึ้นดังนี้ *SetSocketOption (Socket - OptionLevel.Socket,SocketOptionName.ReceiveTimeout, 1234)*

โดย parameter ตัวแรกจะบ่งบอกถึงชนิดของ Socket Option ที่ต้องการทำงานด้วย สำหรับ parameter ตัวที่ 2 SocketOptionName จะใช้สำหรับระบุ Option ที่ต้องการจะกำหนดค่า และสุดท้ายตัวเลขจะเป็นค่าที่กำหนดให้สำหรับ Option นั้น ๆ ซึ่งในบรรทัดข้างต้นก็จะเป็นการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดค่า Time Out ให้กับ Socket นั้นเอง การทำงานที่เกิดขึ้นเมื่อมีการกำหนด time-outs ให้กับ Socket คือ เมื่อมีการเรียกใช้งาน ReceiveFrom() โปรแกรมจะรอรับข้อมูลเป็นเวลา (หน่วยเป็น มิลลิวินาที)ตามที่กำหนดไว้ใน SetSocketOption() ข้างต้น หาก ReceiveFrom() ไม่ได้รับข้อมูล ภายในเวลาดังกล่าว จะมีการสร้าง Exception ขึ้น

- การรอรับค่า Exception

จากหัวข้อที่ผ่านมาเมื่อมี Exception เกิดขึ้นย่อมแสดงถึงเหตุการณ์ Time-Outs ของ Socket ดังนั้นจึงต้องมีการตั้ง Try...Catch ขึ้นมาเพื่อรอรับ Exception ที่จะเกิดขึ้น

- Retransmission

เมื่อสามารถตรวจจับการสูญหายของข้อมูลได้โดยการตั้งค่า Time-Outs และ Catch Exception ซึ่งเกิดจากการ time-outs ได้ก็อาจสร้าง method ขึ้นมาเพื่อ retransmission ซึ่งสูญหายนั้นอีกครั้งนอกจากนั้นหาก Retransmission แล้วยังเกิด Time-Outs หรือ Error อื่นก็จะทำการ Retry อีกทั้งหมด 4 ครั้ง โดย e.ErrorCode ที่เกิดขึ้นจาก Exception นั้นที่สำคัญมีดังนี้

ตารางที่ 2.1 e.ErrorCode ที่สำคัญ

Error Code	Description	Error Code	Description
10013	Permission Denied	10042	Bad protocol option
10014	Bad Address	10043	Protocol not support
10022	Invalid Argument	10048	Address already in used
10024	Too many open socket	10050	Network is down
10035	Resource temp. unavailable	10051	Network is unreachable
10036	Operation now in progress	10054	Connection reset by peer
10038	Socket op. on a non Socket	10056	Soc. is already connected
10039	Destination Address Require	10057	Socket is not connected
10040	Message too long	10064	Host is down
10041	Protocol wrong type	11001	Host not found

2.5 ตัวอย่าง Desktop Management ที่มีใช้ในปัจจุบัน

2.5.1 StarCat 5.0 Enterprise

สตาร์แคท เอนเตอร์ไพรส์ เป็น โปรแกรมระบบบริหารเครื่องคอมพิวเตอร์ที่ถูกออกแบบมา เพื่อรองรับการ บริหารระบบคอมพิวเตอร์ภายในองค์กรทุกขนาด ไม่ว่าจะเป็้องค์กรขนาดเล็กที่มี เครื่องคอมพิวเตอร์ตั้งแต่ 50 เครื่อง ไปจนถึงองค์กรขนาดใหญ่ที่มีเครื่องคอมพิวเตอร์หลายพันเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถใช้งานได้อย่างง่ายดาย ด้วยฟังก์ชันที่ใช้งานที่หลากหลายและครอบคลุมการทำงานเพื่อการบริหารระบบสารสนเทศภายในองค์กรได้อย่างมีประสิทธิภาพ

เนื่องจากสตาร์แคท เป็นโปรแกรมระบบบริหารเครื่องคอมพิวเตอร์ที่ใช้ทรัพยากรระบบต่ำ ไม่ว่าจะเป็นทรัพยากรระบบของเครื่องคอมพิวเตอร์ที่ใช้สำหรับติดตั้งโปรแกรมสตาร์แคท ในส่วนเซิร์ฟเวอร์ ส่วนของเอเจนต์ และการใช้ทรัพยากรระบบการสื่อสารข้อมูลระหว่างกัน ในเครือข่ายที่ต่ำอีกด้วย ทำให้สตาร์แคทสามารถติดตั้งได้ในทุกระบบเครือข่ายโดยไม่จำเป็นต้องปรับปรุงระบบเครือข่ายคอมพิวเตอร์เดิมขององค์กร และ ไม่ทำให้เกิดผลกระทบต่อประสิทธิภาพการทำงานของระบบคอมพิวเตอร์และระบบเครือข่ายขององค์กร อีกทั้งยังสามารถทำงานได้บนเครื่องคอมพิวเตอร์ส่วนบุคคลทุกระดับ

โปรแกรมระบบบริหารเครื่องคอมพิวเตอร์ สตาร์แคท เอนเตอร์ไพรส์ มีการปรับปรุงและพัฒนาอย่างต่อเนื่อง เพื่อให้มั่นใจได้ว่า สตาร์แคท สามารถใช้งานได้อย่างครอบคลุม ตรงตามความต้องการของทุกองค์กรในการบริหารงานระบบคอมพิวเตอร์ เพื่อประสิทธิภาพสูงสุดในการใช้งานโปรแกรมระบบบริหารเครื่องคอมพิวเตอร์อย่างแท้จริง

คุณลักษณะของ StarCat 5.0 Enterprise Edition:

- การควบคุมเครื่องลูกข่ายระยะไกล
- การติดตั้งซอฟต์แวร์ระยะไกล
- การตรวจสอบรายการของฮาร์ดแวร์
- การบริหารผู้ใช้งานระบบ
- การตรวจสอบการใช้งานซอฟต์แวร์
- การควบคุมเครื่องด้วยนโยบายด้านสารสนเทศ
- การตั้งประมวลผลคำสั่งระยะไกล
- การป้องกันการเข้าควบคุมเครื่องลูกข่าย
- การส่งข้อมูล
- การตรวจสอบไฟล์ดูแลทรัพยากรเครื่องลูกข่าย
- การตรวจสอบทรัพยากรของเครื่องคอมพิวเตอร์
- รายงานระบบ
- ความปลอดภัย

2.5.2 IBM Tivoli Software

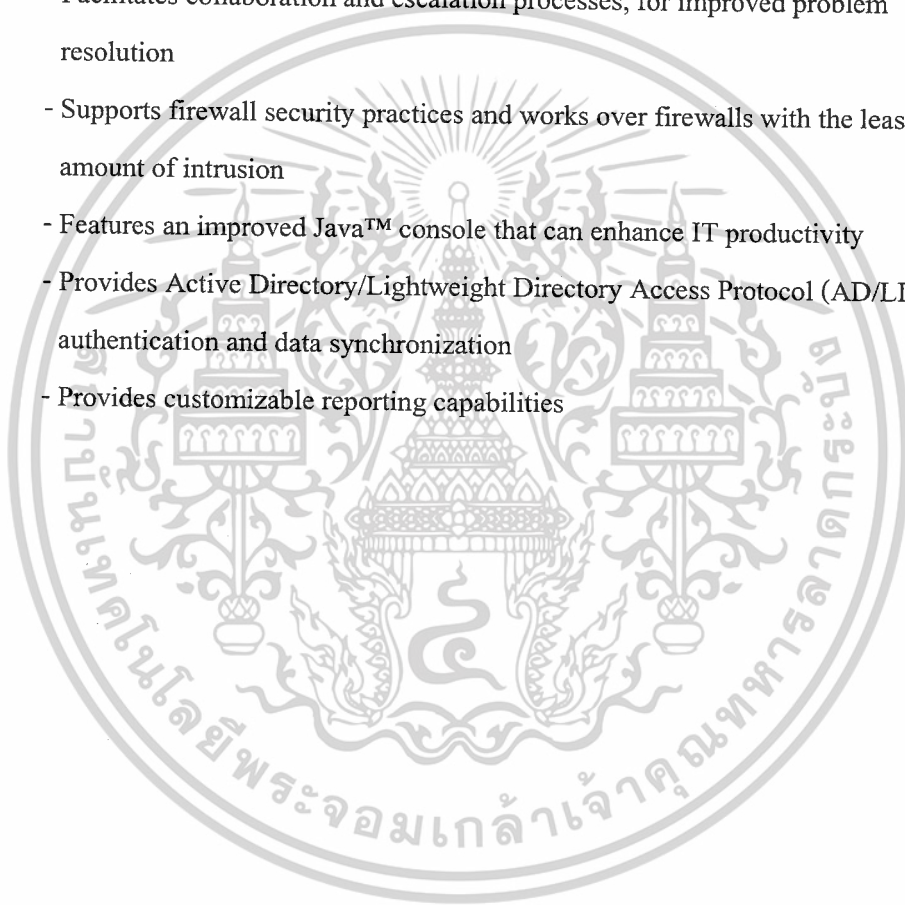
IBM มีผลิตภัณฑ์ที่เป็นโปรแกรมระบบบริหารเครื่องคอมพิวเตอร์จำนวนมาก ภายใต้ชื่อว่า Tivoli เพื่อให้เหมาะสมกับการใช้งานของแต่ละองค์กร อาทิเช่น IBM® Tivoli® Remote Control ซึ่งเป็นโปรแกรมระบบบริหารเครื่องคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อรองรับการควบคุมเครื่องลูกข่ายระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

104002
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณลักษณะของ Tivoli Remote Control:

- Secures firewall traversal through a new proxy component
- Enables organizations to remotely manage thousands of PCs and servers from a central location
- Allows PC problems to be resolved remotely, helping to reduce expensive onsite visits by IT staff
- Enhances security with centralized logging and session video recordings
- Offers new session modes, including Guidance, File Transfer and Chat
- Facilitates collaboration and escalation processes, for improved problem resolution
- Supports firewall security practices and works over firewalls with the least amount of intrusion
- Features an improved Java™ console that can enhance IT productivity
- Provides Active Directory/Lightweight Directory Access Protocol (AD/LDAP) authentication and data synchronization
- Provides customizable reporting capabilities



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและพัฒนา

3.1 Requirement

3.1.1 Server Requirement

3.1.1.1 Inventory Management การบริหารจัดการรายการทรัพย์สินของระบบ

- สามารถแสดงรายละเอียดฮาร์ดแวร์ของเครื่องลูกข่าย เช่น Main board Serial Number, Main board Model, Sound Card, Memory Banks, BIOS version เป็นต้น
- สามารถแสดงรายละเอียดซอฟต์แวร์ของเครื่องลูกข่าย เช่น Installed Software, Target Directory, Product Key
- สามารถแสดงรายละเอียดข้อมูลทั่วไปของเครื่องลูกข่าย OS Version, Mac Address, System Path, Asset Serial numbers, OS CD Key เป็นต้น

3.1.1.2 System Control Function คำสั่งที่ใช้ควบคุมระบบ

- สามารถดูสถานะการทำงานของเครื่องลูกข่าย
- สามารถปิด/รีสตาร์ทของเครื่องลูกข่าย
- สามารถควบคุมการใช้พอร์ต USB ของเครื่องลูกข่าย
- สามารถควบคุมการใช้อินเทอร์เน็ตของเครื่องลูกข่าย
- สามารถควบคุมการใช้โปรแกรมเครื่องลูกข่าย

3.1.1.3 System report รายงานของระบบ

- รายงานระบบที่แสดงข้อมูลทางด้านฮาร์ดแวร์
- รายงานระบบที่แสดงข้อมูลทางด้านซอฟต์แวร์
- รายงานระบบที่แสดงข้อมูลทางด้านระบบปฏิบัติการเครื่องคอมพิวเตอร์

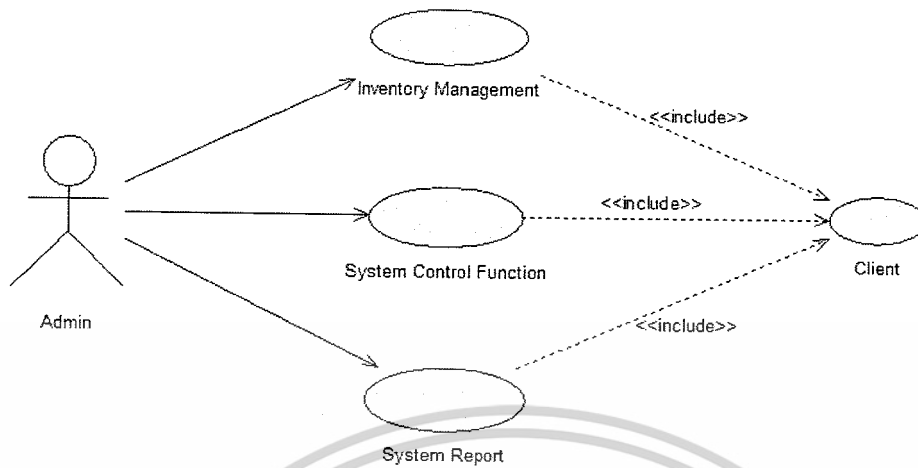
3.3.2 Client Requirement

3.3.2.1 Program สามารถ run เมื่อเปิดเครื่อง

3.3.2.2 ฝังตัวอยู่ใน Service ของ Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

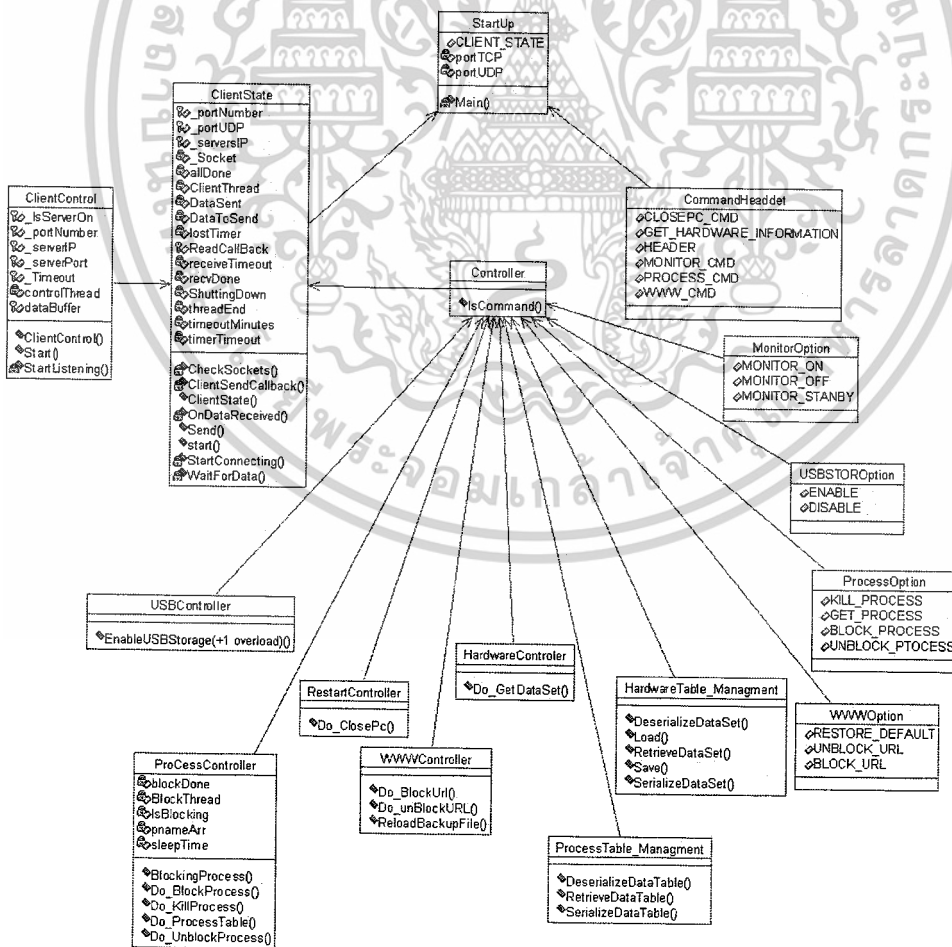
3.2. Use Case Diagram



รูปที่ 3.1 Use Case Diagram

3.3 Class Diagram

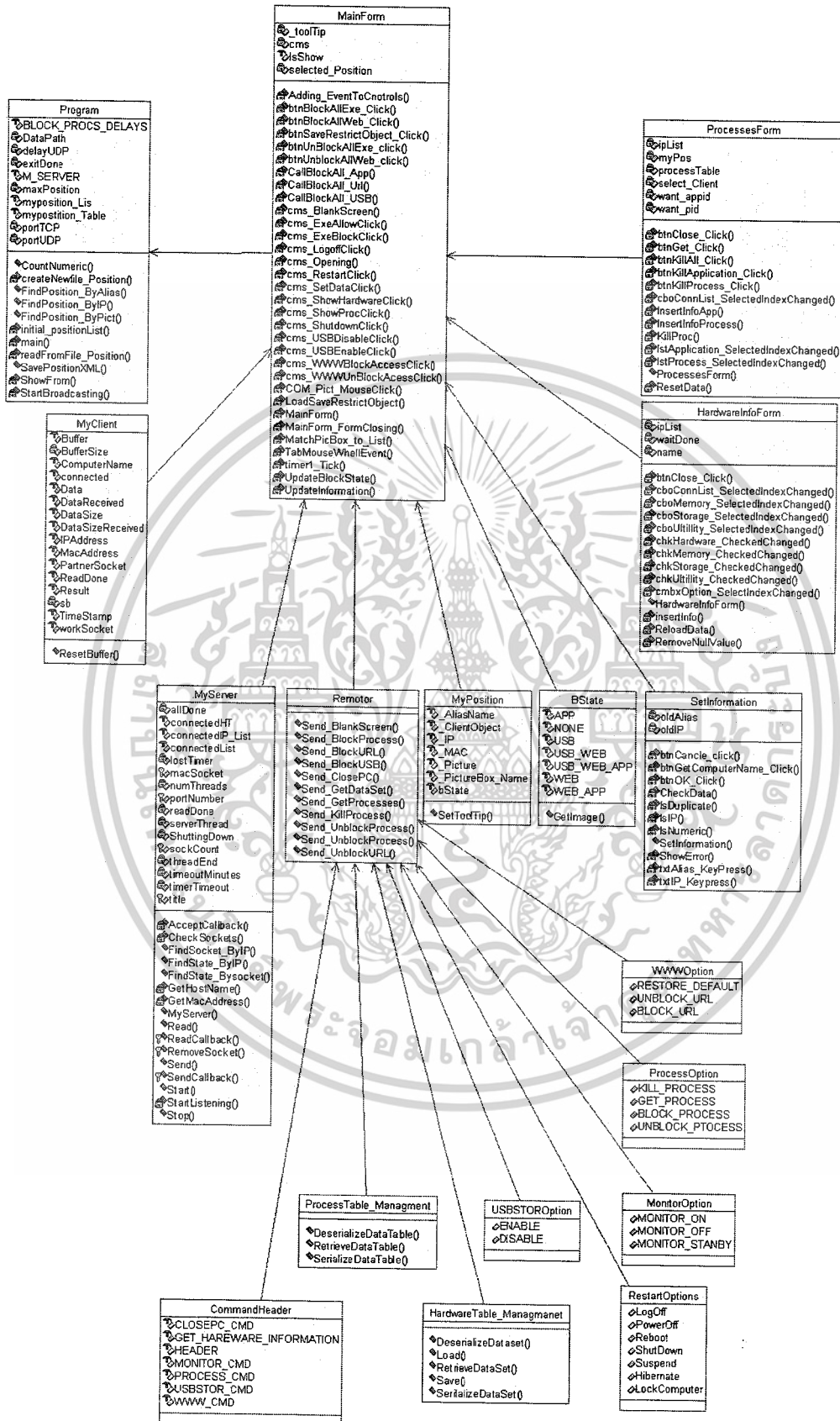
3.3.1 เครื่องลูกข่าย



รูปที่ 3.2 Class Diagram ของเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 เครื่องเซิร์ฟเวอร์

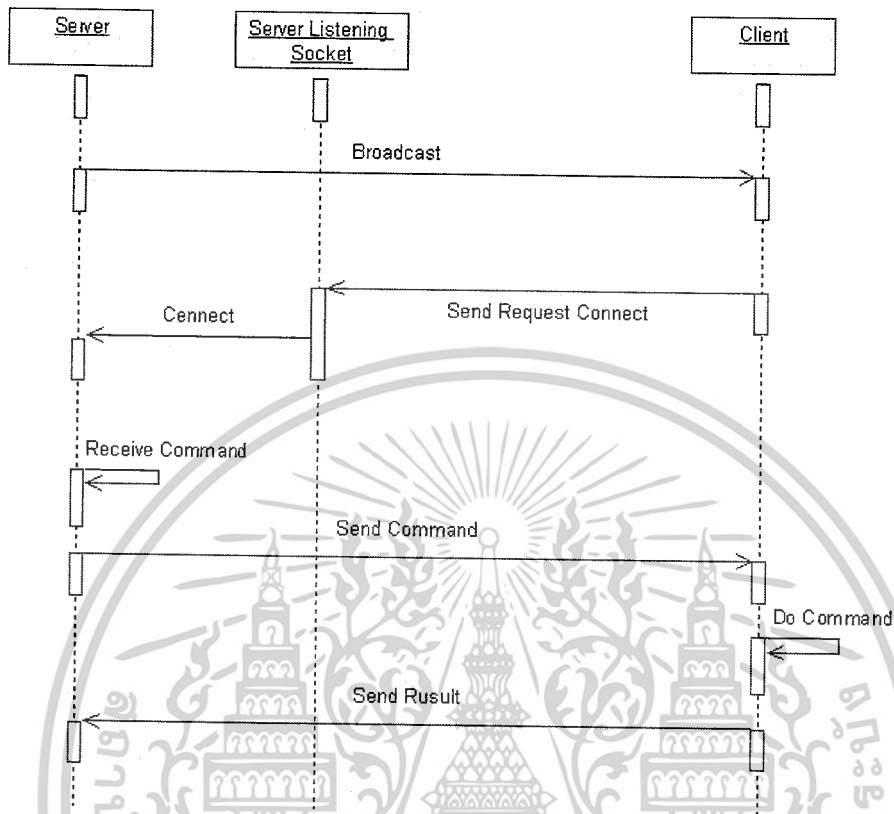


รูปที่ 3.3 Class Diagram ของเครื่องเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

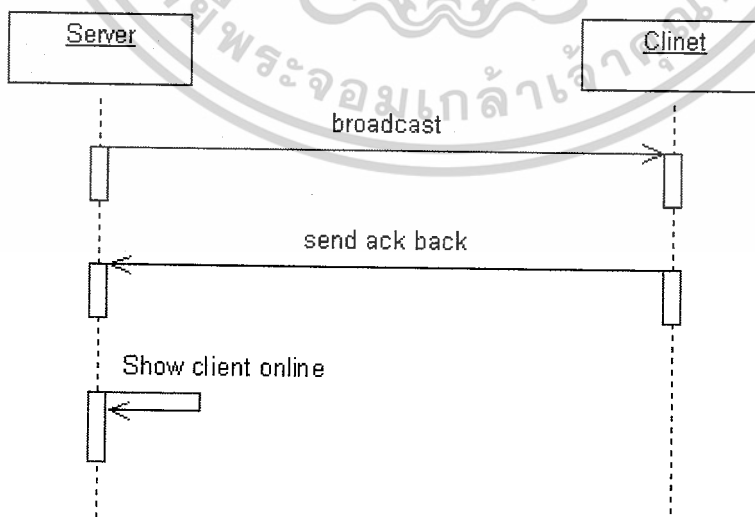
3.4 Sequence Diagram

3.4.1 การติดต่อและส่งคำสั่งระหว่างเซิร์ฟเวอร์กับเครื่องลูกข่าย



รูปที่ 3.4 Sequence Diagram แสดงการติดต่อและส่งคำสั่งระหว่างเซิร์ฟเวอร์กับเครื่องลูกข่าย

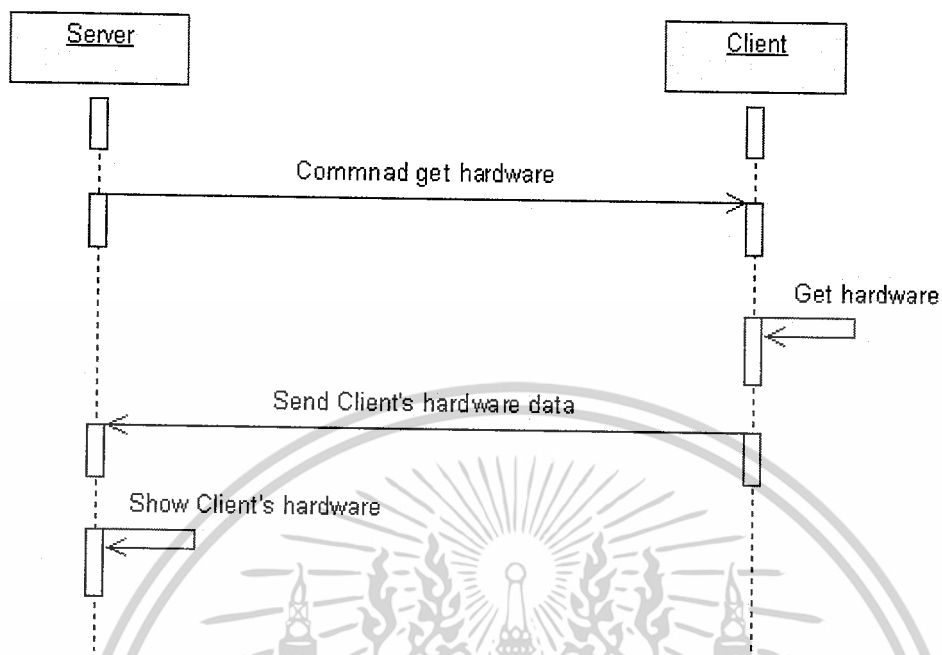
3.4.2 การแสดงสถานะของเครื่องลูกข่าย



รูปที่ 3.5 Sequence Diagram แสดงสถานะของเครื่องลูกข่าย

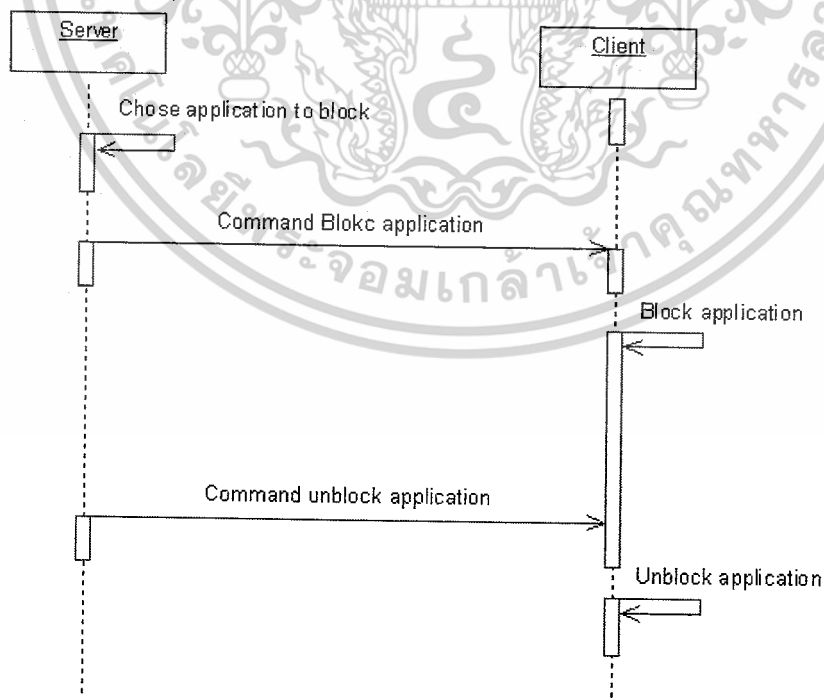
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 การเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย



รูปที่ 3.6 Sequence Diagram แสดงการเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย

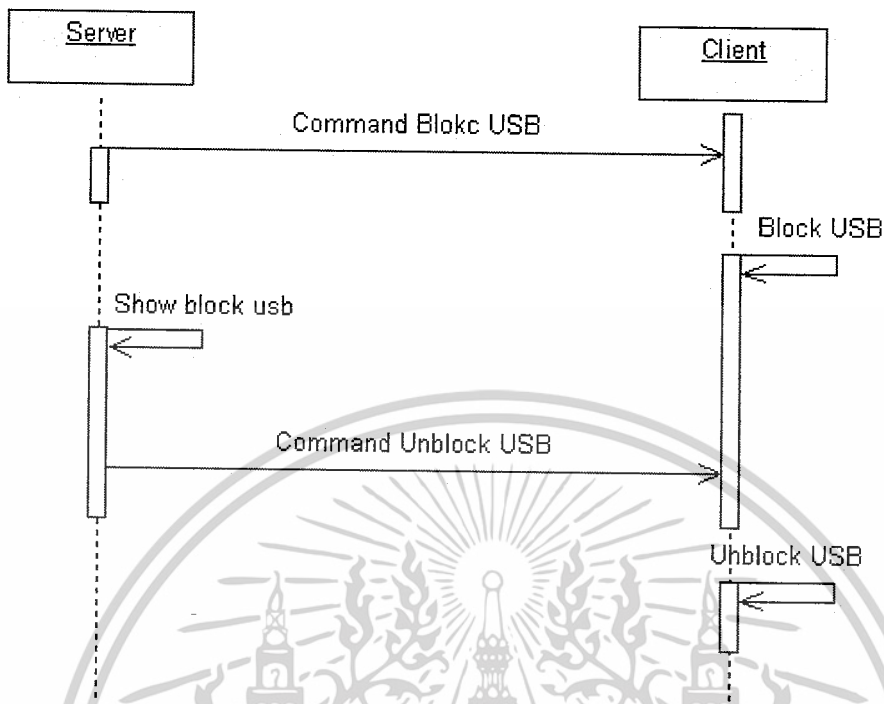
3.4.4 การเรียกดูโปรเซสของเครื่องลูกข่าย



รูปที่ 3.7 Sequence Diagram แสดงการเรียกดูโปรเซสของเครื่องลูกข่าย

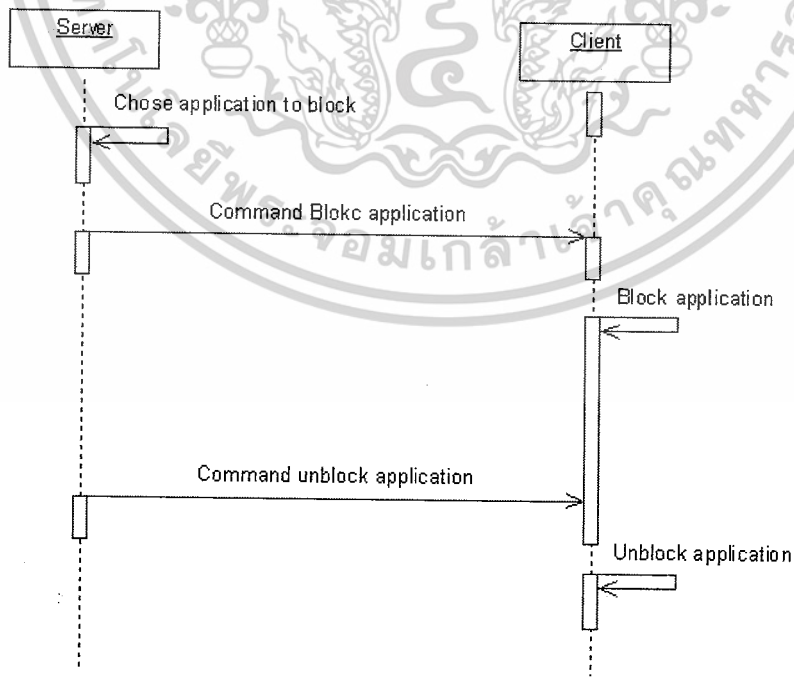
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 การควบคุมการใช้งานพอร์ต USB



รูปที่ 3.8 Sequence Diagram แสดงการควบคุมการใช้งานพอร์ต USB

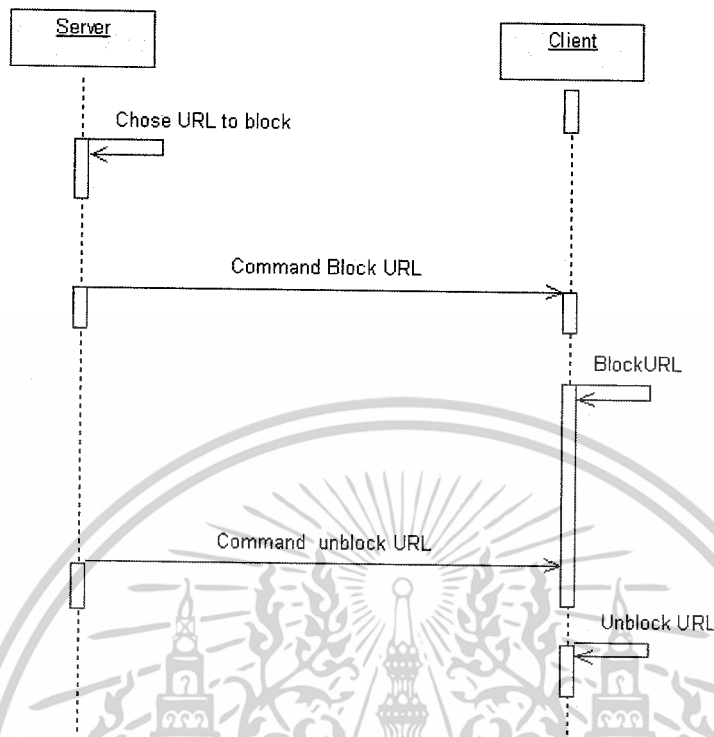
3.4.6 การควบคุมการใช้งานแอปพลิเคชัน



รูปที่ 3.9 Sequence Diagram แสดงการควบคุมการใช้งานแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

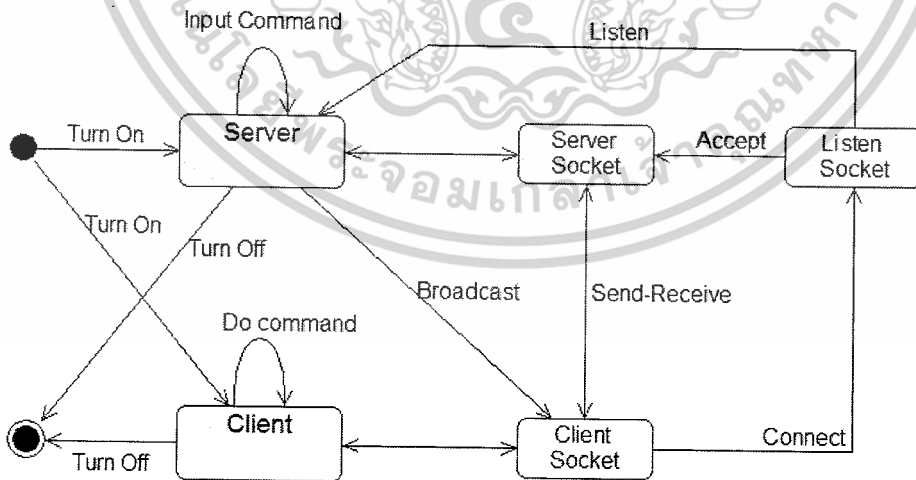
3.4.7 การควบคุมการเข้าเว็บไซต์



รูปที่ 3.10 Sequence Diagram แสดงการควบคุมการเข้าเว็บไซต์

3.5 State Diagram

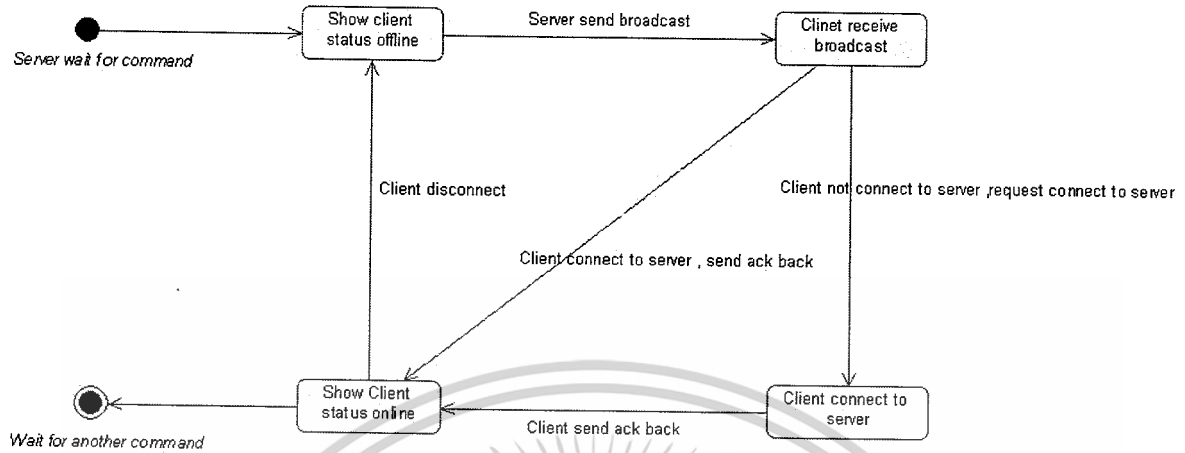
3.5.1 การติดต่อและส่งคำสั่งระหว่างเซิร์ฟเวอร์กับเครื่องลูกข่าย



รูปที่ 3.11 State Diagram แสดงการติดต่อและส่งคำสั่งระหว่างเซิร์ฟเวอร์กับเครื่องลูกข่าย

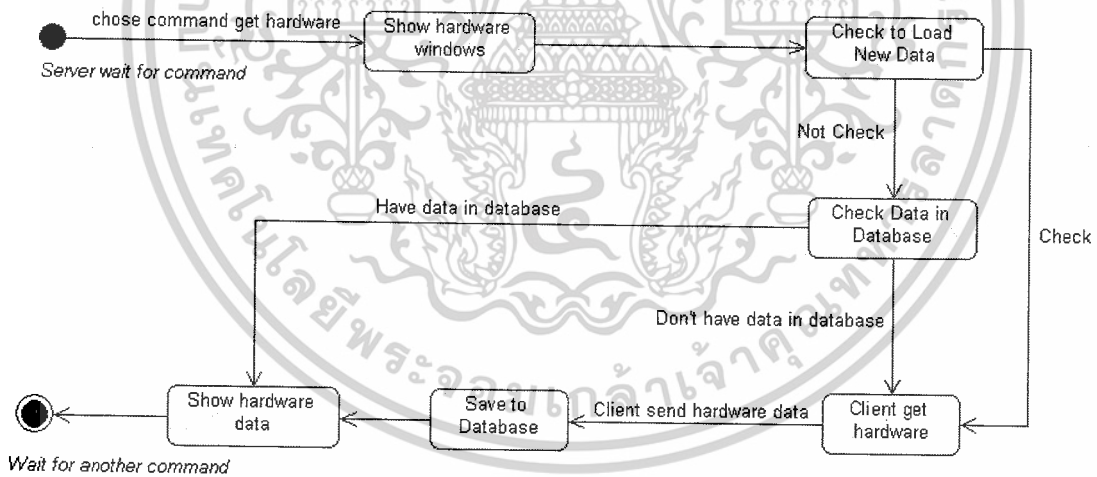
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 การแสดงสถานะของเครื่องลูกข่าย



รูปที่ 3.12 State Diagram แสดงสถานะของเครื่องลูกข่าย

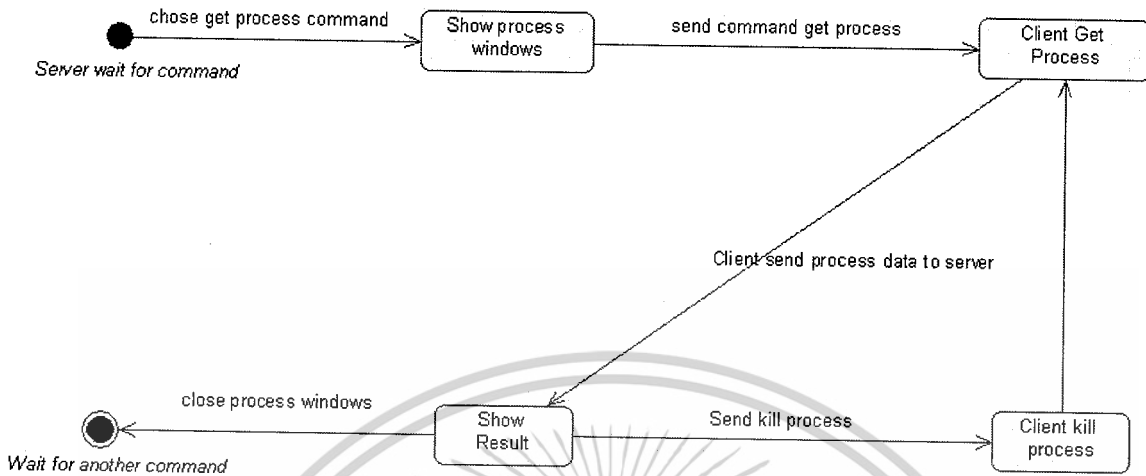
3.5.3 การเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย



รูปที่ 3.13 State Diagram แสดงการเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย

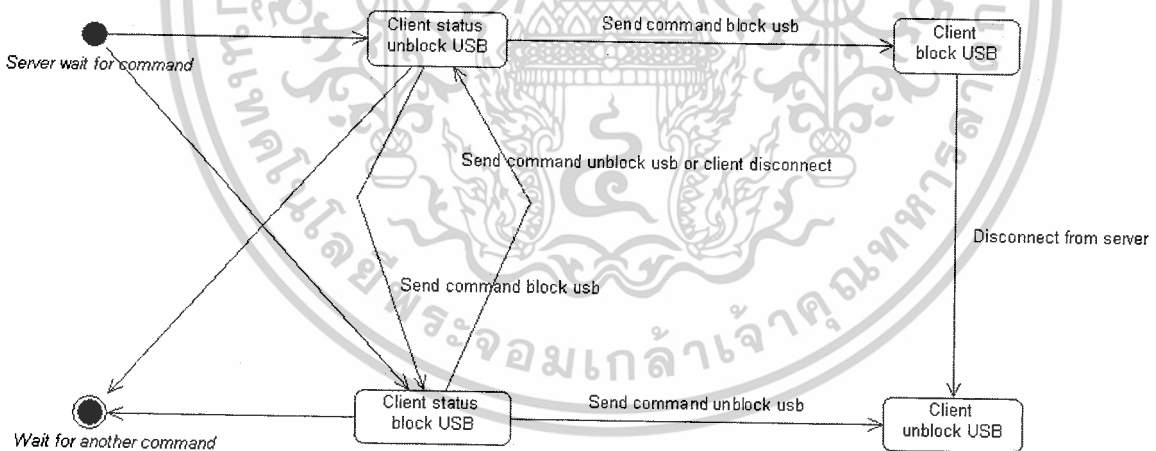
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 การเรียกดูโพรเซสของเครื่องลูกข่าย



รูปที่ 3.14 State Diagram แสดงการเรียกดูโพรเซสของเครื่องลูกข่าย

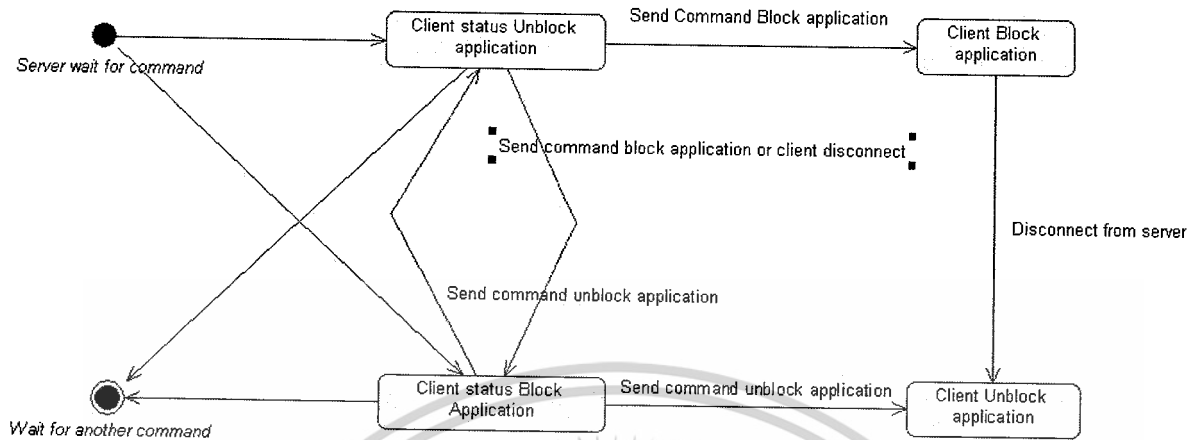
3.5.5 การควบคุมการใช้งานพอร์ต USB



รูปที่ 3.15 State Diagram แสดงการควบคุมการใช้งานพอร์ต USB

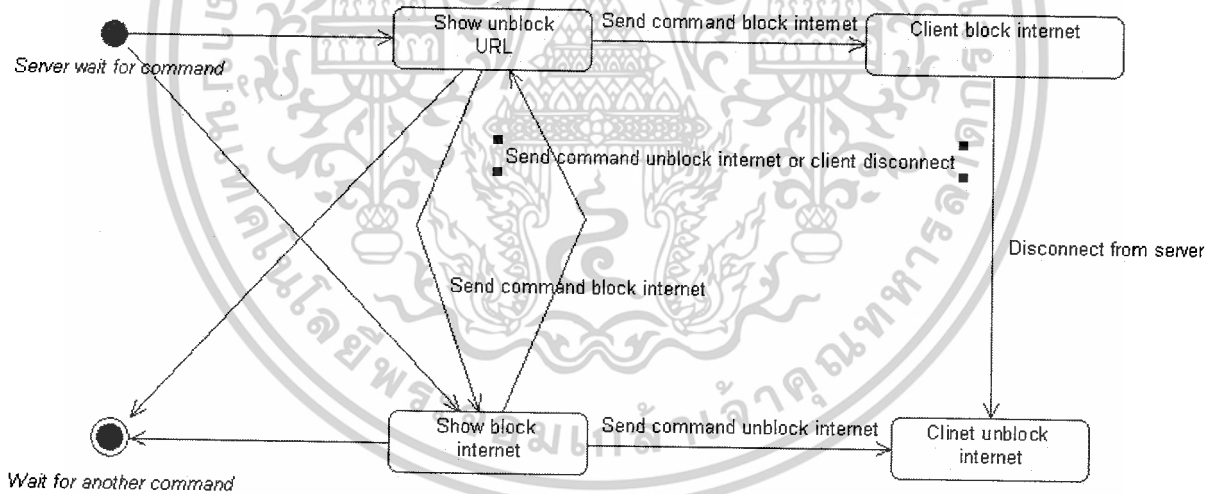
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6 การควบคุมการใช้งานแอปพลิเคชัน



รูปที่ 3.16 State Diagram แสดงการควบคุมการใช้งานแอปพลิเคชัน

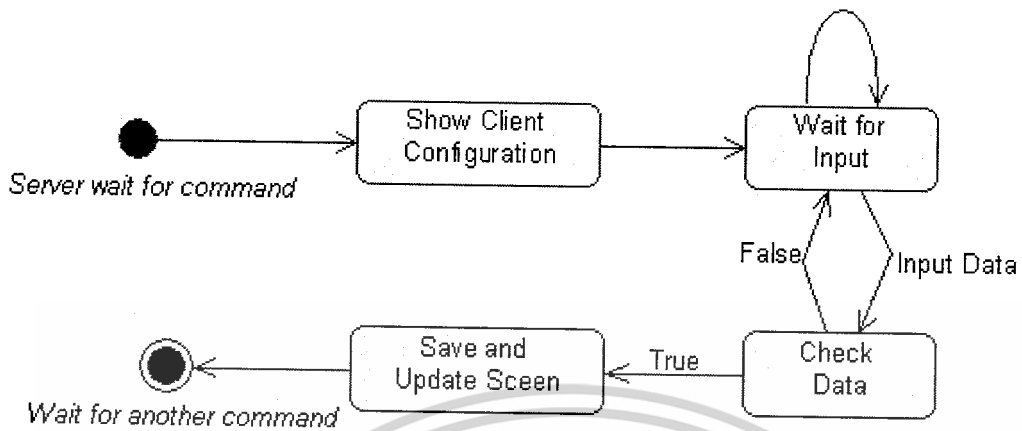
3.5.7 การควบคุมการเข้าเว็บไซต์



รูปที่ 3.17 State Diagram แสดงการควบคุมการเข้าเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.8 การตั้งค่า IP Address



รูปที่ 3.18 State Diagram แสดงการตั้งค่า IP Address



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

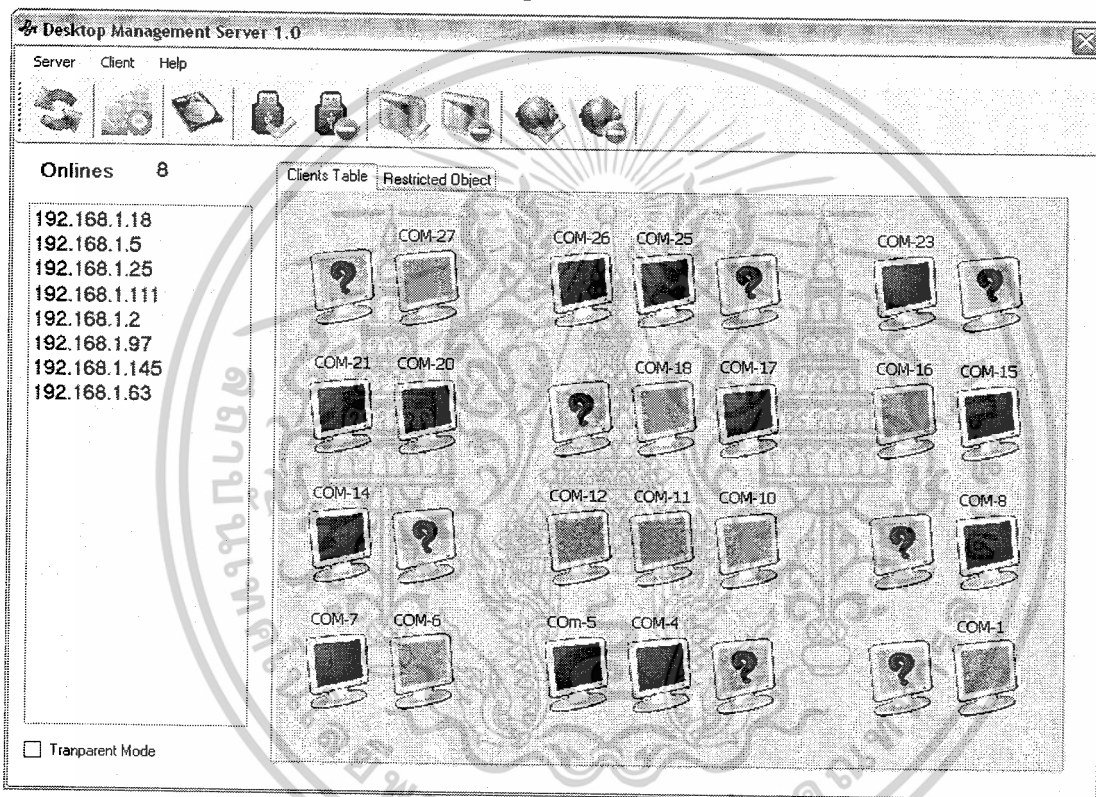
การทดลองและผลการทดลอง

จากการพัฒนาโปรแกรม ได้ผลการทดลองดังต่อไปนี้

4.1 เครื่องเซิร์ฟเวอร์

4.1.1 หน้าต่างควบคุมของเซิร์ฟเวอร์

ระบบมีหน้าต่างที่ใช้ในการควบคุมเครื่องลูกข่าย



รูปที่ 4.1 หน้าต่างควบคุมของเครื่องเซิร์ฟเวอร์

4.1.2 การแสดงสถานะของเครื่องลูกข่าย

ระบบสามารถแสดงว่าเครื่องลูกข่ายนั้นมีสถานะอะไรอยู่ คือ เปิดเครื่อง ปิดเครื่อง หรือยังไม่มีการกำหนด IP Address ให้กับเครื่องลูกข่ายนั้น ดังรูปที่ 4.2, รูปที่ 4.3 และรูปที่ 4.4 ตามลำดับ ซึ่งแสดงไว้ในรูปแบบการจัดโต๊ะคอมพิวเตอร์ในห้องปฏิบัติการ



รูปที่ 4.2 เครื่องลูกข่ายมีสถานะเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

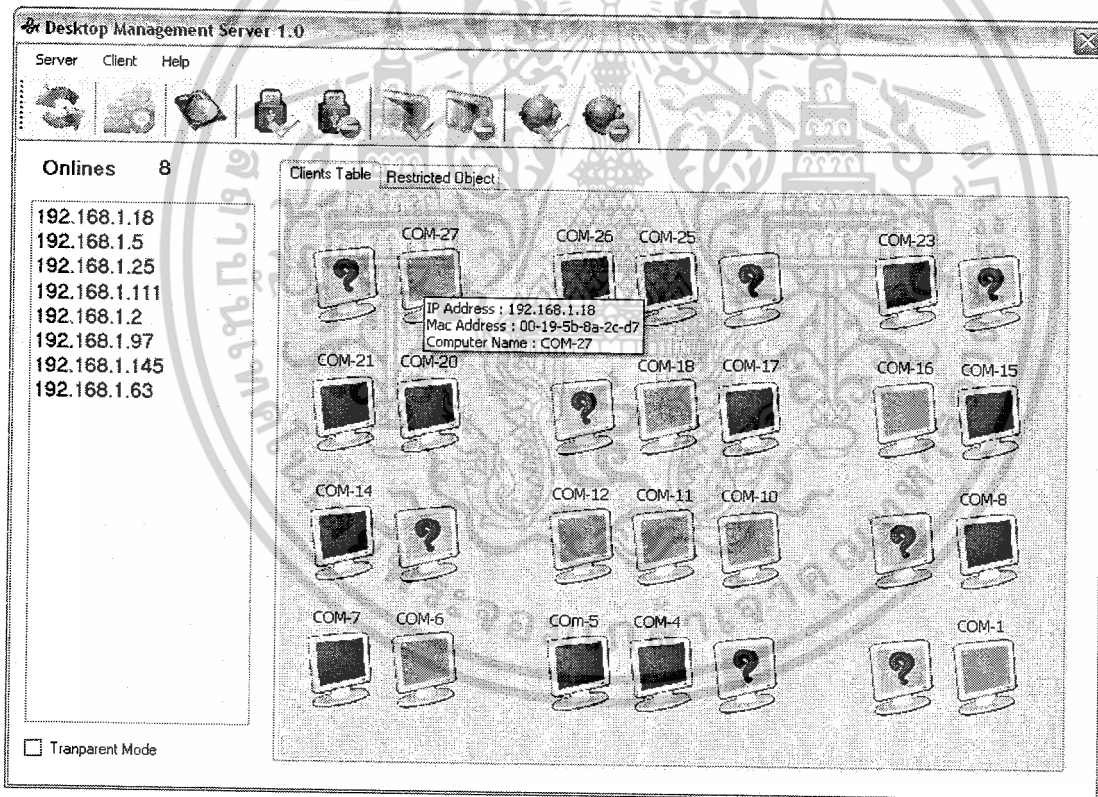


รูปที่ 4.3 เครื่องลูกข่ายมีสถานะปิด



รูปที่ 4.4 เครื่องลูกข่ายที่ยังไม่ได้กำหนด IP Address

สามารถดูได้ว่าเครื่องลูกข่ายนั้นๆ มี IP Address, MAC Address และ Computer Name อะไร โดยวางเมาส์ไว้ที่เครื่องลูกข่าย ดังรูปที่ 4.5

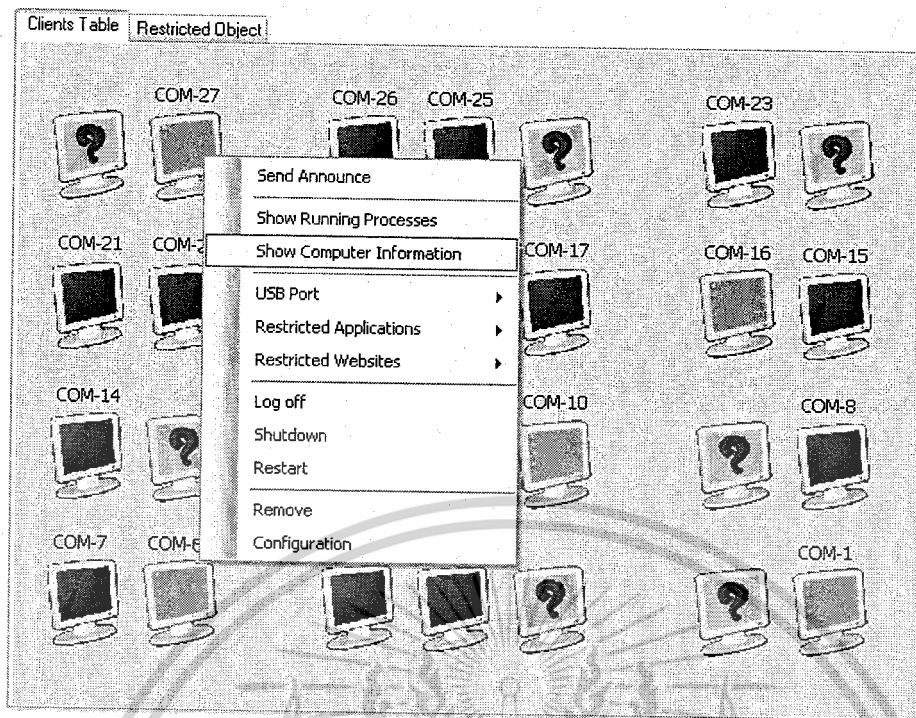


รูปที่ 4.5 การดู IP Address, MAC Address และ Computer Name ของเครื่องลูกข่าย

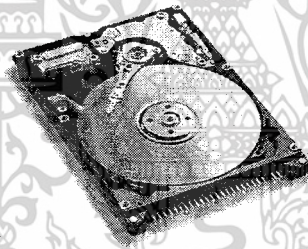
4.1.3 การเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย

ระบบสามารถเรียกดูฮาร์ดแวร์ของเครื่องลูกข่าย โดยเลือกเครื่องลูกข่าย คลิกขวา เลือก “Show Information” ดังรูปที่ 4.6 หรือเลือกไอคอนที่แถบเครื่องมือด้านบน ดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



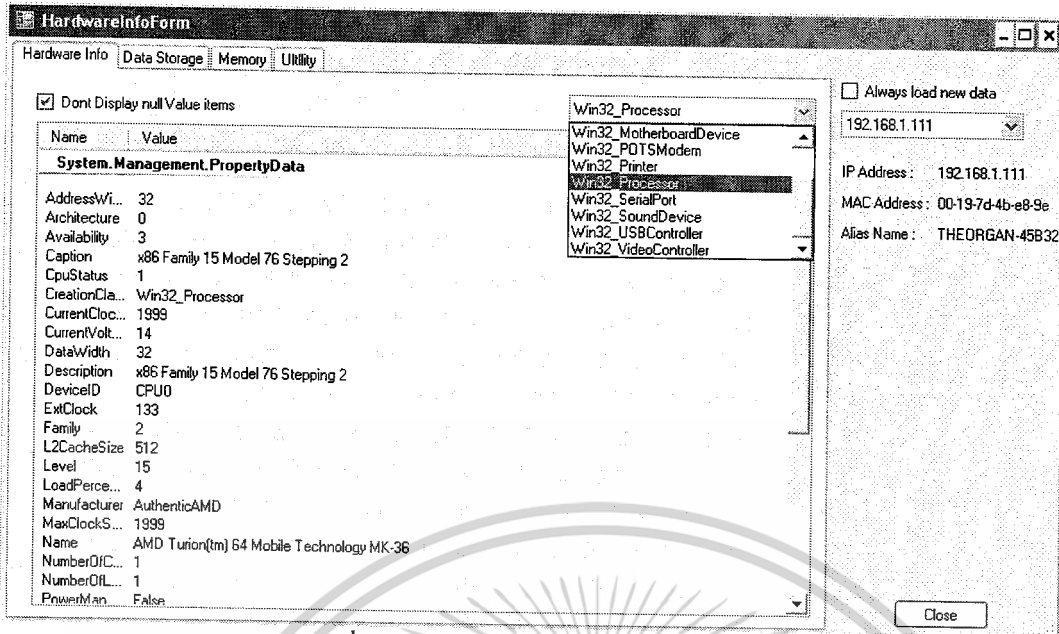
รูปที่ 4.6 การเลือก “Show Hardware Information”



รูปที่ 4.7 ไอคอนการ “Show Hardware Information”

หน้าต่าง Hardware Information ปรากฏขึ้น ดังรูปที่ 4.8 ถ้าเลือก Always load new data จะเป็นการเรียกข้อมูลของเครื่องลูกข่ายใหม่ทุกครั้งและจะเก็บข้อมูลลงดาต้าเบส ทั้งนี้ถ้าไม่เลือก Always load new data ก็สามารดูข้อมูลของเครื่องลูกข่ายที่อยู่ในดาต้าเบสได้ หากข้อมูลนั้นไม่เคยมีการเรียกดูก็จะเรียกข้อมูลจากเครื่องลูกข่ายทันที ดังนั้นเราจึงสามารถดูข้อมูลของเครื่องลูกข่ายทั้งเครื่องที่เปิด หรือปิดอยู่ก็ได้ แต่ถ้าเครื่องปิดอยู่นั้นไม่เคยมีการดูข้อมูลก็จะไม่สามารถดูข้อมูลได้

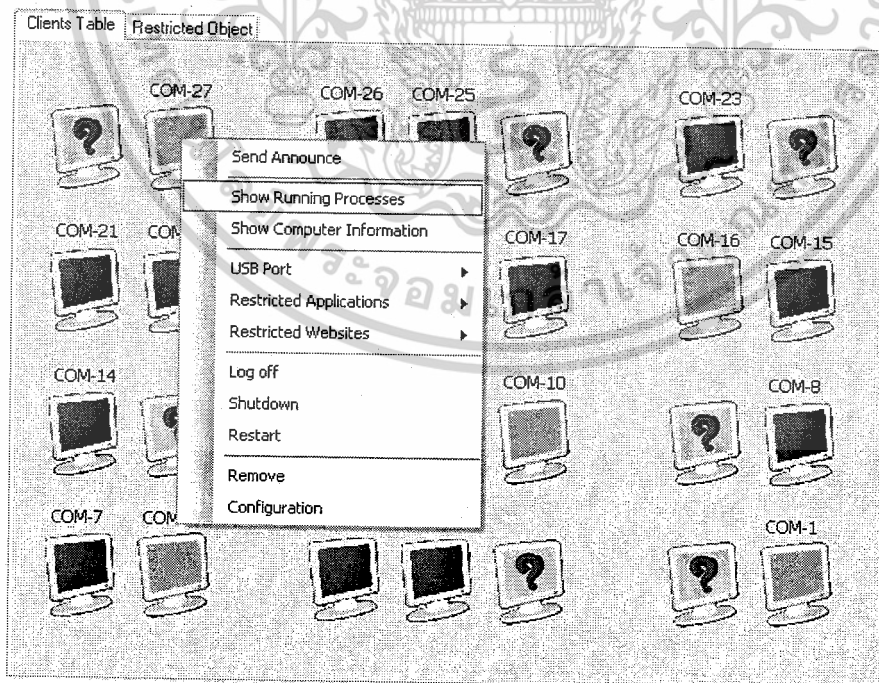
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 หน้าต่าง Hardware Information

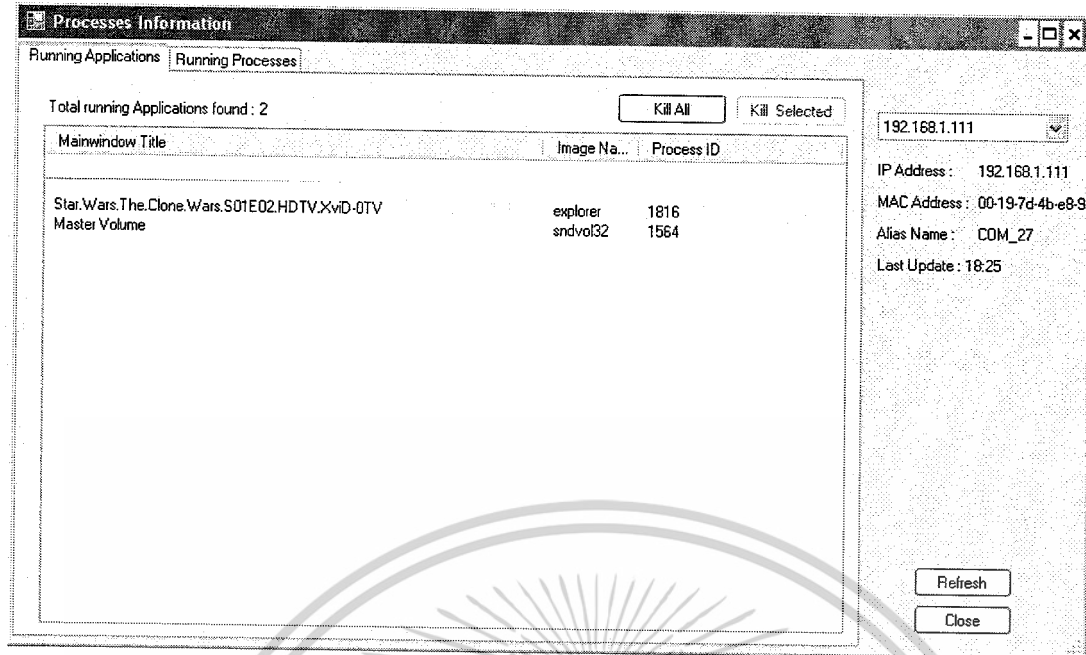
4.1.4 การเรียกดูโพรเซสของเครื่องลูกข่าย

ระบบสามารถเรียกดูโพรเซสของเครื่องลูกข่ายว่าในขณะนั้นมีโพรเซสอะไรทำงานอยู่บ้าง และสามารถดูแอปพลิเคชันที่เครื่องลูกข่ายใช้งานอยู่ ดังนั้นผู้ดูแลระบบหรืออาจารย์ผู้สอนสามารถปิดแอปพลิเคชันที่นอนเหนือจากการเรียนการสอน โดยคลิกขวา เลือก “Show Running Processes” ดังรูปที่ 4.9 และหน้าต่าง Application Information จะปรากฏดังรูป 4.10



รูปที่ 4.9 การเลือก “Show Running Process”

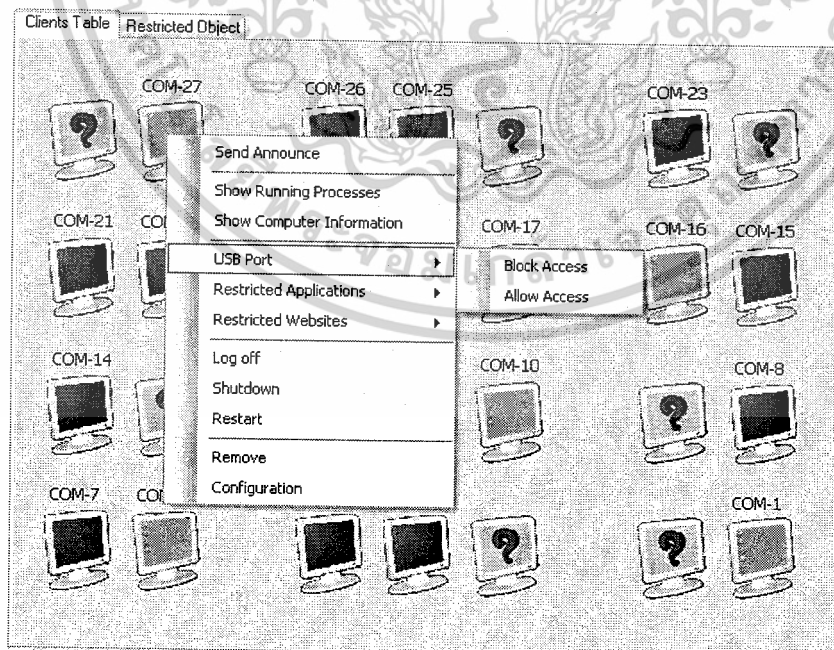
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 หน้าต่าง Processes Information

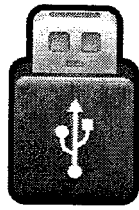
4.1.5 การควบคุมการใช้พอร์ต USB

ระบบสามารถควบคุมการใช้พอร์ต USB ของเครื่องลูกข่าย โดยมีการอนุญาตให้เครื่องลูกข่ายสามารถใช้งานพอร์ต USB หรือไม่อนุญาตให้เครื่องลูกข่ายใช้งานพอร์ต USB สามารถกำหนดได้ที่แต่ละเครื่องของเครื่องลูกข่าย ดังรูปที่ 4.11 และสามารถกำหนดพร้อมกันทุกเครื่อง โดยเลือกไอคอน USB ซึ่งอยู่ตรงแถบเครื่องมือ ดังรูปที่ 4.12



รูปที่ 4.11 การกำหนดการใช้งานพอร์ต USB เครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ไอคอน USB

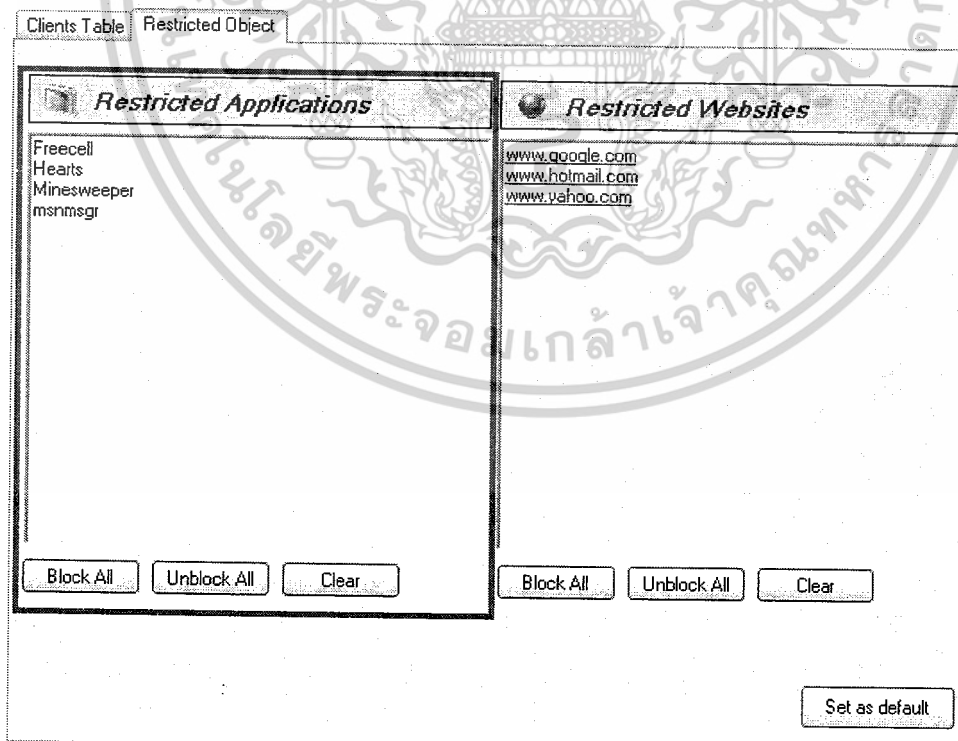
เมื่อเครื่องลูกข่ายถูกกำหนดไม่อนุญาตให้ใช้งานพอร์ต USB ที่เครื่องลูกข่ายจะมีสัญลักษณ์ดังรูปที่ 4.13 ปรากฏที่เครื่องลูกข่ายนั้นๆ



รูปที่ 4.13 สัญลักษณ์การไม่อนุญาตให้ใช้งานพอร์ต USB

4.1.6 การควบคุมการใช้งานแอปพลิเคชัน

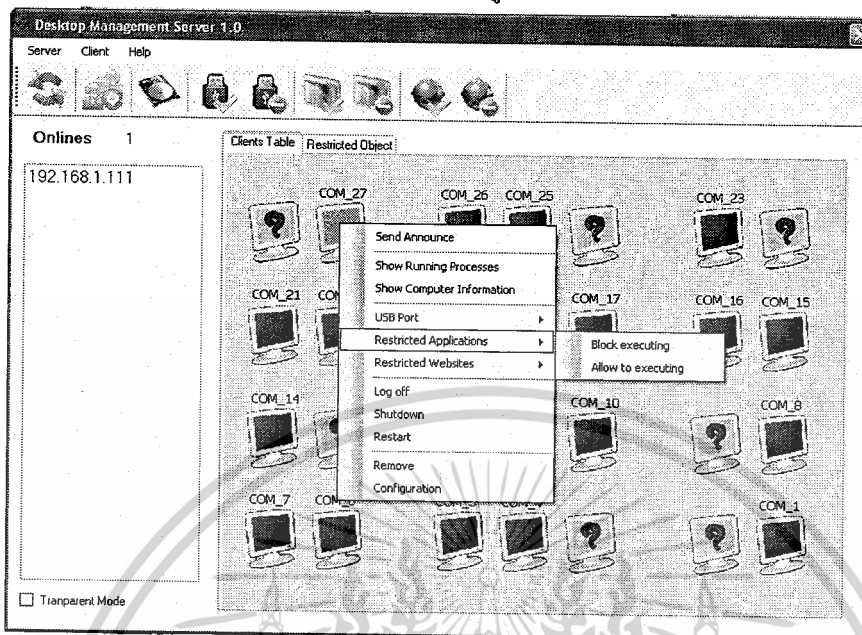
ระบบสามารถควบคุมการใช้งานแอปพลิเคชัน โดยไม่อนุญาตให้ใช้แอปพลิเคชันตามที่ผู้ดูแลระบบหรืออาจารย์ผู้สอนกำหนดโดยสามารถกำหนดที่ Restricted Object ในส่วนของ Restricted Applications ดังรูปที่ 4.14



รูปที่ 4.14 การกำหนดแอปพลิเคชันที่ไม่อนุญาตให้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

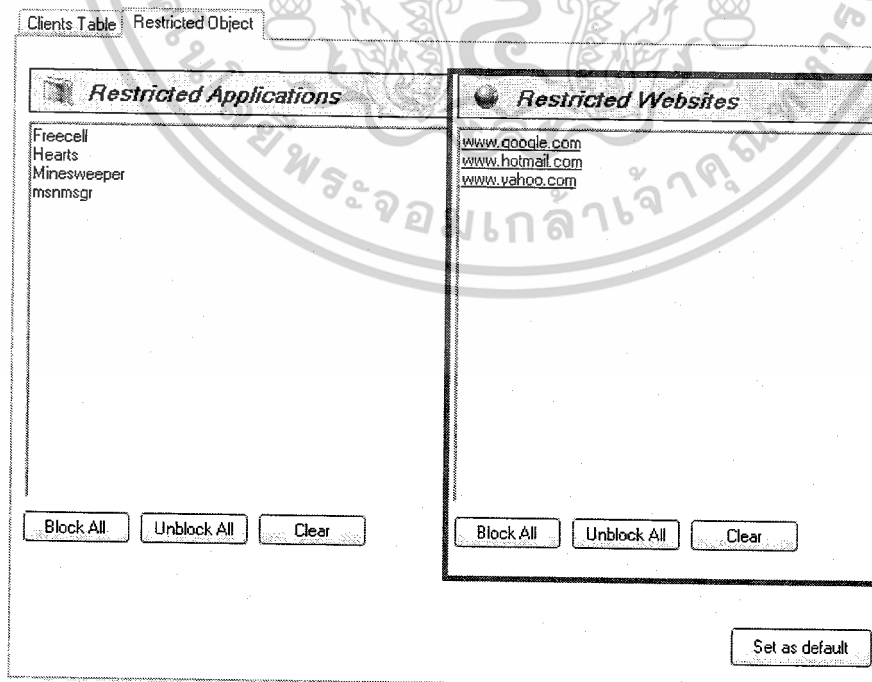
หากต้องการควบคุมการใช้งานแอปพลิเคชันที่เครื่องสามารถ สามารถคลิกขวาที่เครื่องลูกข่ายเครื่องนั้น แล้วเลือก Restricted Application ดังรูปที่ 4.15



รูปที่ 4.15 การควบคุมการใช้งานแอปพลิเคชันที่เครื่อง

4.1.7 การควบคุมการใช้งานเว็บไซต์

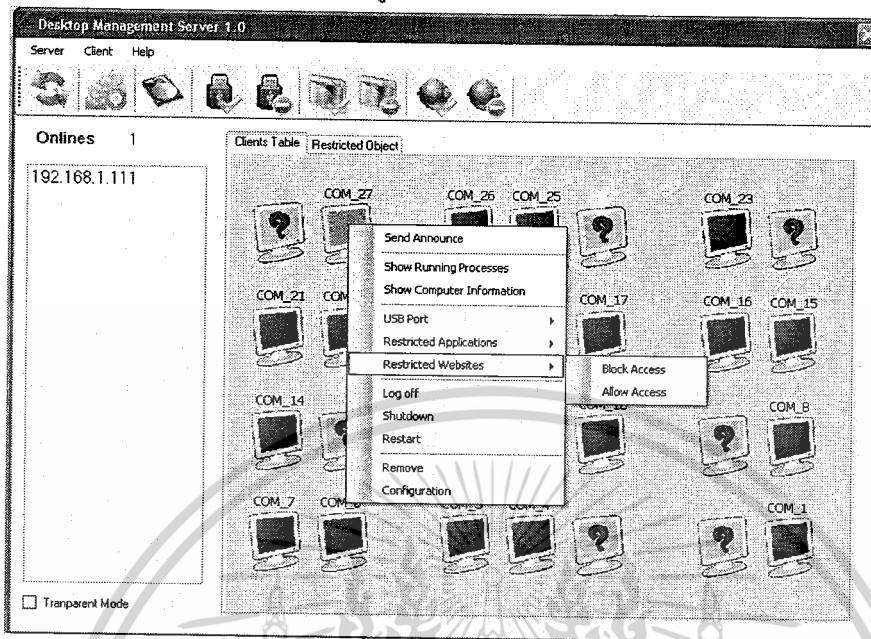
ระบบสามารถควบคุมการเข้าเว็บไซต์ โดยไม่อนุญาตให้เข้าเว็บไซต์ตามที่คุณดูแลระบบหรืออาจารย์ผู้สอนกำหนด โดยสามารถกำหนดที่ Restricted Object ในส่วนของ Restricted Websites ดังรูปที่ 4.16



รูปที่ 4.16 การกำหนดเว็บไซต์ที่ไม่อนุญาตให้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

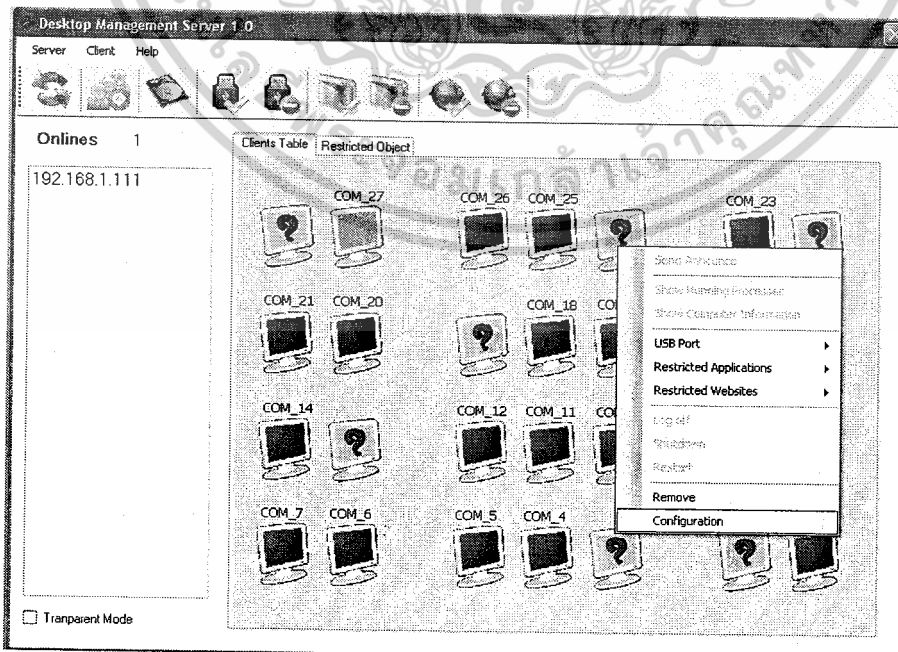
หากต้องการควบคุมการใช้งานเว็บไซต์ที่ละเครื่องสามารถ สามารถคลิกขวาที่เครื่องลูกข่าย เครื่องนั้น แล้วเลือก Restricted Website ดังรูปที่ 4.17



รูปที่ 4.17 การควบคุมการใช้งานเว็บไซต์ที่ละเครื่อง

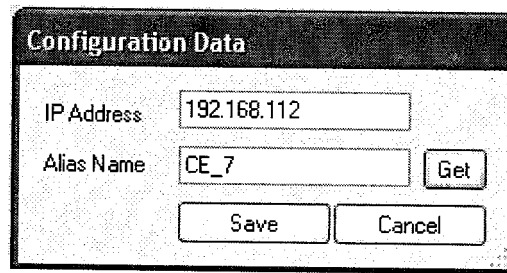
4.1.8 การกำหนด IP Address ให้กับเครื่องลูกข่าย

ระบบจะแสดงสถานะของเครื่อง ได้ก็ต่อเมื่อมีการกำหนด IP Address ให้กับเครื่องลูกข่าย เพื่อให้ทราบว่าในขณะนั้น IP Address นี้มีในระบบหรือไม่ ถ้ามี IP Address นี้ในระบบแสดงว่า เครื่องนั้นกำลังเปิด โดยเลือกเครื่องลูกข่ายที่จะกำหนดดังรูปที่ 4.18 และหน้าต่างในการตั้งค่าจะปรากฏดังรูปที่ 4.19



รูปที่ 4.18 การเลือกตั้งค่าเครื่องลูกข่าย

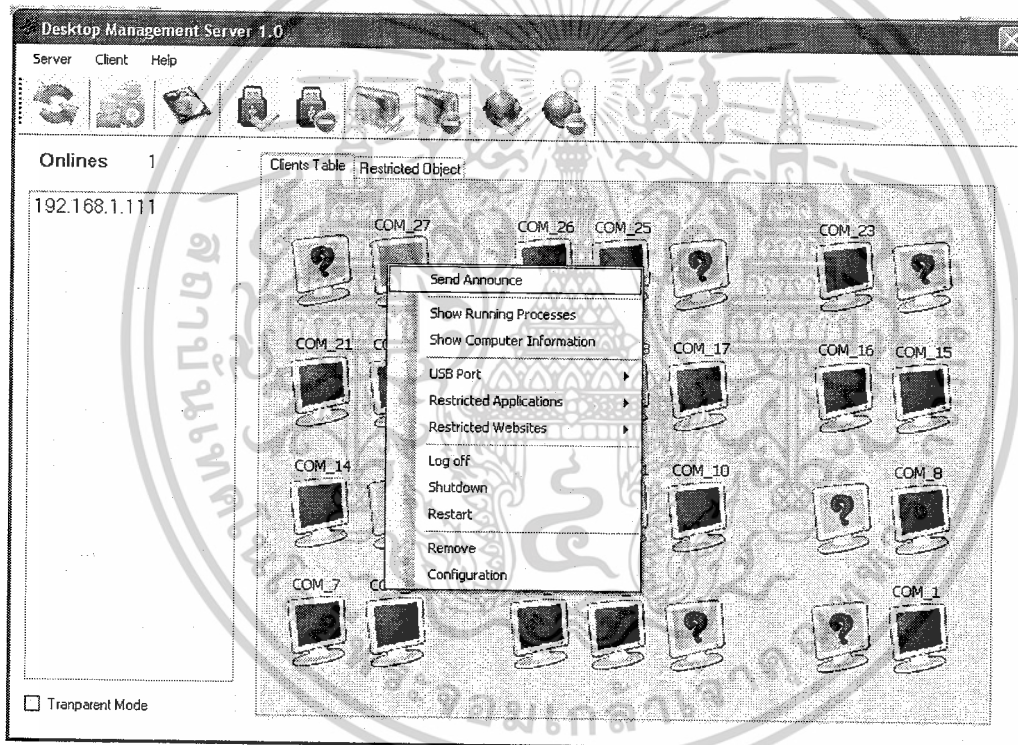
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 หน้าต่างตั้งค่าเครื่องลูกข่าย

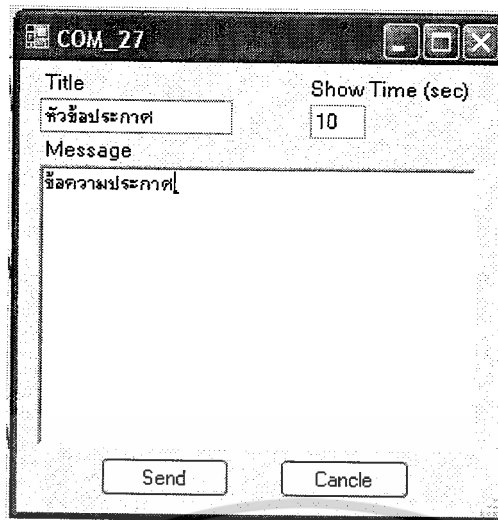
4.1.9 การประกาศไปยังเครื่องลูกข่าย

ระบบสามารถส่งข้อความประกาศไปยังเครื่องลูกข่ายได้ ดังรูปที่ 4.20 และจะมีหน้าต่างดังรูปที่ 4.21 ปრაกฏขึ้นเพื่อให้กรอกหัวข้อและข้อความที่ต้องการประกาศ และสามารถเลือกระยะเวลาที่จะให้ข้อความนั้นแสดง



รูปที่ 4.20 การส่งข้อความไปยังเครื่องลูกข่ายนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

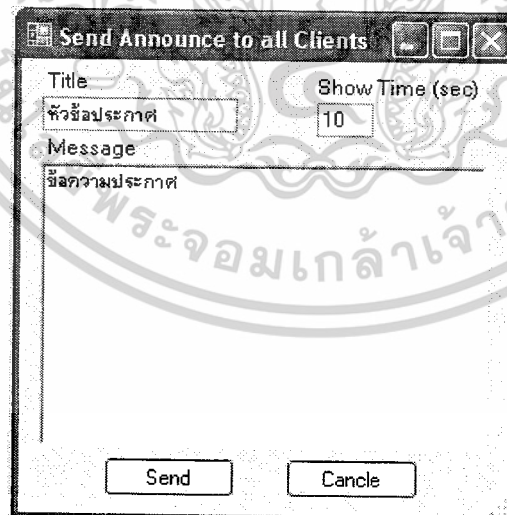


รูปที่ 4.21 หน้าต่างการประกาศของเครื่องลูกข่ายนั้นๆ

หากต้องการประกาศไปยังเครื่องลูกข่ายทุกเครื่องสามารถเลือกไอคอนดังรูปที่ 4.22 และจะมีหน้าต่างดังรูปที่ 4.23 ปรากฏขึ้น



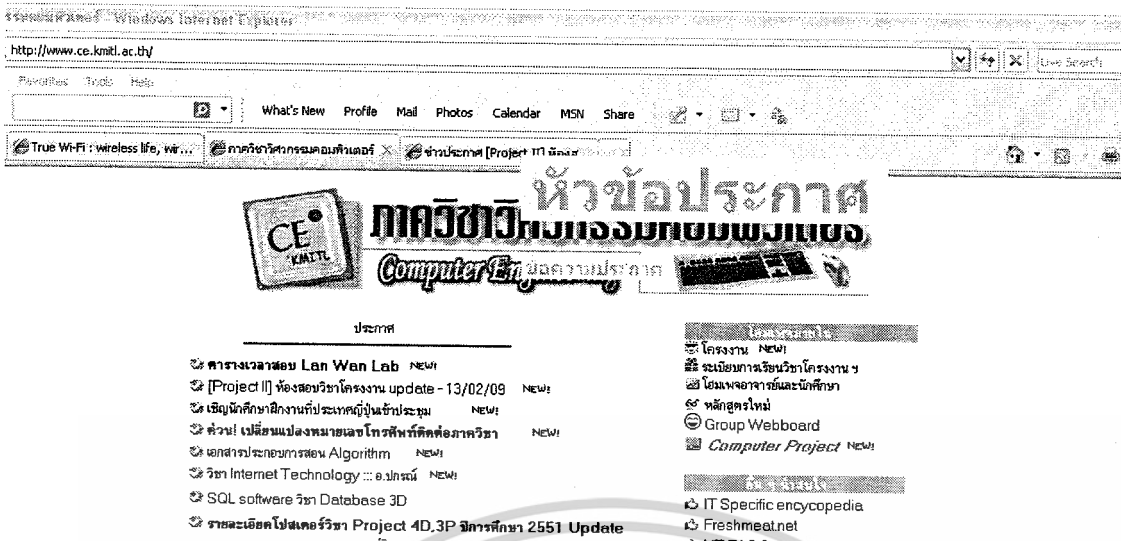
รูปที่ 4.22 ไอคอนการประกาศข้อความไปยังเครื่องลูกข่ายทุกเครื่อง



รูปที่ 4.23 หน้าต่างการประกาศของเครื่องลูกข่าย

เมื่อประกาศข้อความไปยังเครื่องลูกข่าย ณ เครื่องลูกข่ายจะปรากฏข้อความดังรูปที่ 4.24

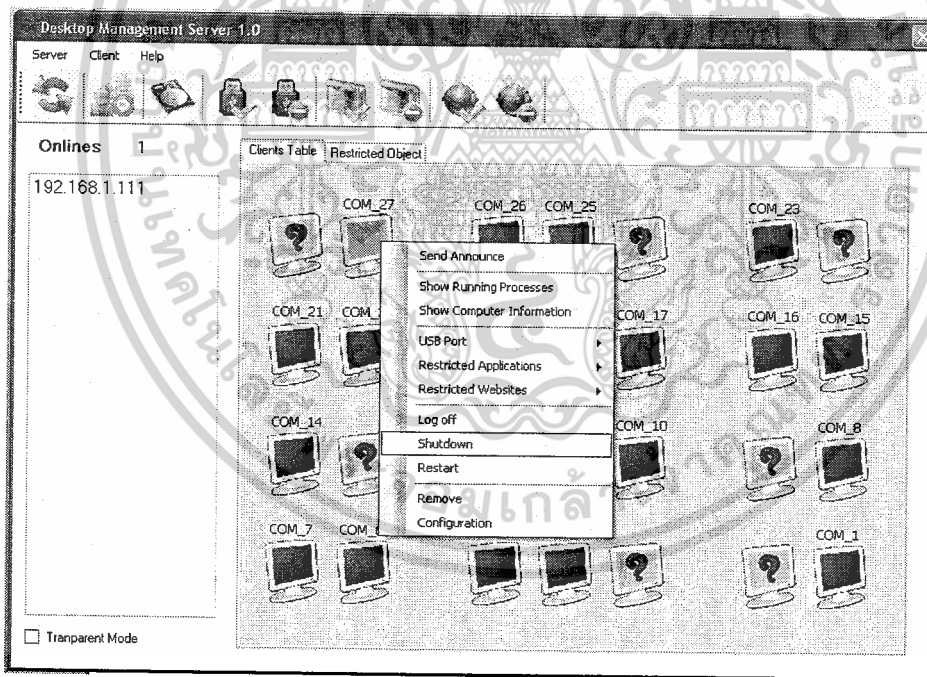
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 ข้อความที่แสดงบนเครื่องลูกข่าย

4.1.10 การสั่งปิดเครื่องลูกข่าย

ระบบสามารถสั่งปิดเครื่องลูกข่ายได้ นอกจากนี้ยังสามารถสั่งรีสตาร์ท และถือออกออฟเครื่องลูกข่ายได้อีกด้วย



รูปที่ 4.25 การสั่งปิดเครื่องลูกข่าย

4.2 เครื่องลูกข่าย

4.2.1 ระบบสามารถ run โปรแกรมขึ้นมาเมื่อเปิดเครื่องได้

4.2.2 ระบบสามารถฟังตัว Service ของ window ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทสรุป

Desktop Management คือ เครื่องมือที่ทำให้เราจัดการและควบคุมเครื่องลูกข่ายจำนวนมากผ่าน Network ได้อย่างสะดวกและง่ายดาย เสมือนกำลังนั่งอยู่ ณ เครื่องลูกเครื่องนั้น เช่น การที่เราสามารถเรียกดูหน้าจอเครื่องลูกข่ายแบบ Real-Time ว่าในขณะนั้นผู้ใช้กำลังทำอะไรอยู่ และสามารถที่จะควบคุมได้, สั่งเปิด-ปิดเครื่อง, สั่งบล็อกโปรแกรมต่างๆ เป็นต้น

เนื่องจากโปรแกรมตัวนี้ได้เริ่มพัฒนา จึงยังมีฟังก์ชันต่างๆ ไม่ครบเท่ากับโปรแกรม Desktop Management อย่าง StarCat หรือ Tivoli โดยตัวโปรแกรมจะมีฟังก์ชันสำคัญๆ เช่น ส่วนที่เกี่ยวกับการจัดการ โพรเซสของเครื่องลูกข่าย, ส่วนที่เกี่ยวกับข้อมูลฮาร์ดแวร์ของเครื่องลูกข่าย หรือแม้แต่การสั่ง Log off, Shut down, Restart เครื่องลูกข่ายได้

การติดต่อระหว่างเครื่องเซิร์ฟเวอร์กับเครื่องลูกข่ายนั้นจะใช้ Socket ซึ่งเป็นแบบ Stream Socket หรือ TCP Socket ในการติดต่อระหว่างกัน และจะให้เครื่องเซิร์ฟเวอร์ส่งคำสั่งไปยังเครื่องลูกข่ายเพื่อควบคุมการทำงานของเครื่องลูกข่ายได้

5.2 วิจารณ์สิ่งที่ได้จากโครงการ

ฝึกการทำงานร่วมกันเป็นกลุ่มก้อน เป็นทีมงาน ได้ฝึกฝนการเขียนโปรแกรมด้วยภาษา C# และได้พัฒนาทักษะการเขียนโปรแกรมแบบ object oriented programming ได้ฝึกการออกแบบ UML ได้ฝึกการใช้งานแบบ Command line ได้ความรู้ทางด้าน Network Programming

5.3 ปัญหาอุปสรรคและแนวทางแก้ไข

เนื่องจากโปรเจกต์นี้ต้องใช้ความรู้ ความเข้าใจในหลายๆด้าน ทั้งด้าน Network Programming, Service ของ Windows, ฟังก์ชันต่างๆ ที่เกี่ยวข้องกับตัว windows ซึ่งการเขียนติดต่อกับ Windows เป็นเรื่องที่ยาก รวมทั้งต้องศึกษาภาษา C# ที่ใช้ในการพัฒนาโปรแกรมอีก และโครงการนี้เป็นโครงการที่ทำเป็นซอฟต์แวร์ จึงต้องศึกษาอย่างละเอียด คิดให้รอบรอบรองรับการทำงานหลายๆด้าน และต้องทำให้สามารถใช้งานได้สะดวกด้วย จึงทำให้มีปัญหาของโปรแกรมบางจุดที่คาดไม่ถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางการพัฒนาต่อ

เขียนพัฒนาฟังก์ชันที่น่าสนใจ และมีความจำเป็นสำหรับ Desktop Management ให้ครบถ้วนสมบูรณ์มากยิ่งขึ้น เช่น การ Remote Desktop สามารถลบไฟล์เมื่อเครื่องถูกข่ายใช้งานเสร็จเรียบร้อย และตรวจสอบแก้ไขปัญหาของโปรแกรมที่มีให้ม้น้อยที่สุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ว้จันพงส์ เกษมศิริ. 2551. เอกสารประกอบการสอนวิชา 01074201 Network Programming

กรุงเทพฯ :ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Robert H. Davis II. 2003. **Interact With Desktop when Installing Windows Service.** [Online].

Available : <http://www.codeproject.com/KB/install/cswindowsservicedesktop.aspx>

Alireza. Shirazi. 2008. **How To Get Hardware Information.** [Online].

Available : <http://www.codeproject.com/KB/system/GetHardwareInformation.aspx>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

คู่มือการใช้งานซอฟต์แวร์

ก.1 แนะนำเบื้องต้น

ก.1.1 ส่วนประกอบของซอฟต์แวร์

ประกอบด้วยส่วนประกอบหลักของระบบ 2 ส่วน ได้แก่

ก.1.1.1 เซิร์ฟเวอร์ เป็นส่วนที่ใช้ในการควบคุมระบบ ซึ่งเป็นส่วนที่ทำหน้าที่ควบคุมเครื่องลูกข่ายทั้งหมดในระบบ และจัดเก็บข้อมูลทั้งอุปกรณ์ฮาร์ดแวร์ ซอฟต์แวร์ จากเครื่องลูกข่าย โดยมีหน้าที่ในการควบคุม เพื่อให้ผู้ดูแลหรือผู้ใช้งานระบบทำการตรวจสอบ ฝึมอง สั่งงาน หรือจัดการเครื่องลูกข่าย

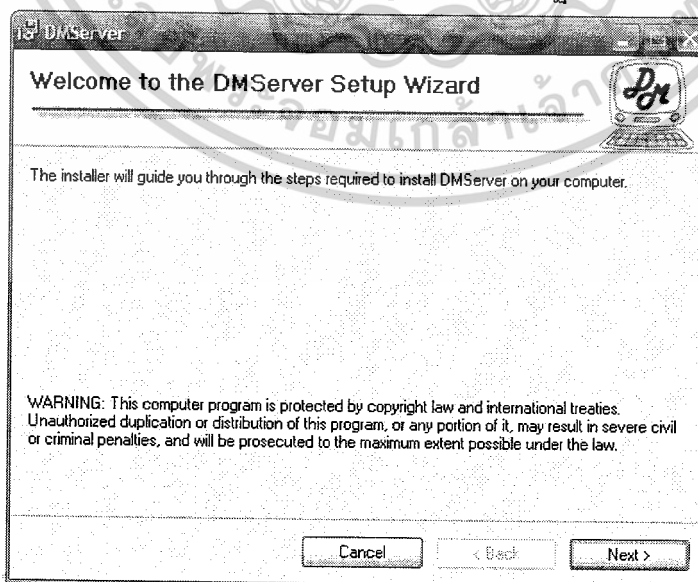
ก.1.1.2 เครื่องลูกข่าย เพื่อใช้ในการรองรับการสั่งงานจากส่วนอื่นๆในระบบ โดยมีหน้าที่ในการอ่านข้อมูลระบบของเครื่องลูกข่ายเพื่อส่งไปยังเซิร์ฟเวอร์ และรับคำสั่งจากเซิร์ฟเวอร์ที่ทำการสั่งงาน แล้วดำเนินการตามคำสั่งที่ได้รับ ซึ่งโดยปกติเครื่องลูกข่ายจะทำงานแบบไม่แสดงสถานะ

ก.2 การติดตั้งโปรแกรม

ก.2.1 การติดตั้งโปรแกรมที่เครื่องเซิร์ฟเวอร์

1. ทำการติดตั้งซอฟต์แวร์

โดยการรันไฟล์สำหรับติดตั้งที่ชื่อว่า setup.exe จะปรากฏหน้าต่างเริ่มการติดตั้ง

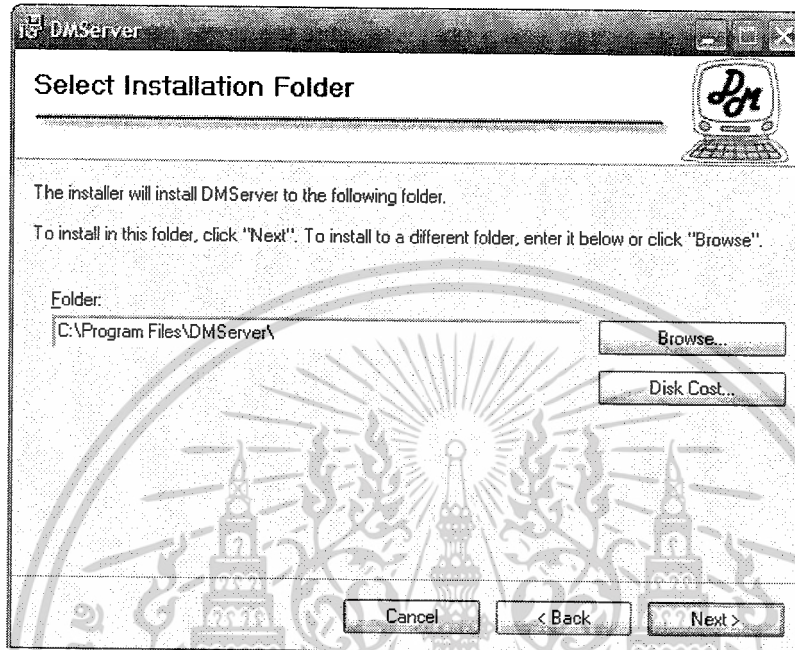


รูปที่ ก.1 หน้าต่างเพื่อเริ่มการติดตั้งซอฟต์แวร์ DMServer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

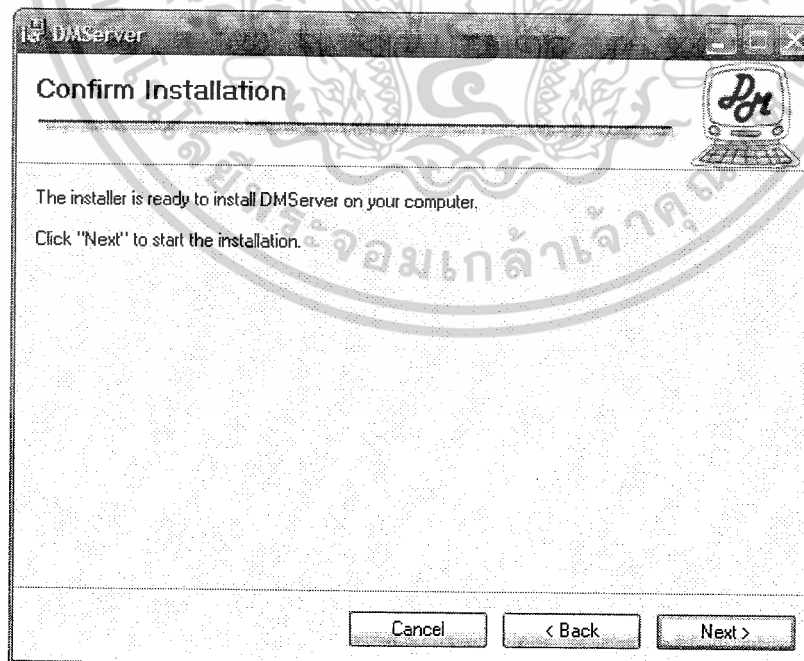
2. คลิก “Next” เพื่อทำการติดตั้งต่อ

จะปรากฏหน้าต่างกำหนดพาร์ทของไฟล์เดอร์ที่ต้องการจัดเก็บไฟล์ โดยปกติซอฟต์แวร์ จะทำการกำหนดพาร์ทของไฟล์เดอร์ให้โดยอัตโนมัติ หากต้องการเปลี่ยน ให้กดปุ่ม “Browse...” เพื่อเลือกพาร์ทใหม่ตามต้องการ



รูปที่ ก.2 หน้าต่างสำหรับกำหนดพาร์ทของไฟล์เดอร์

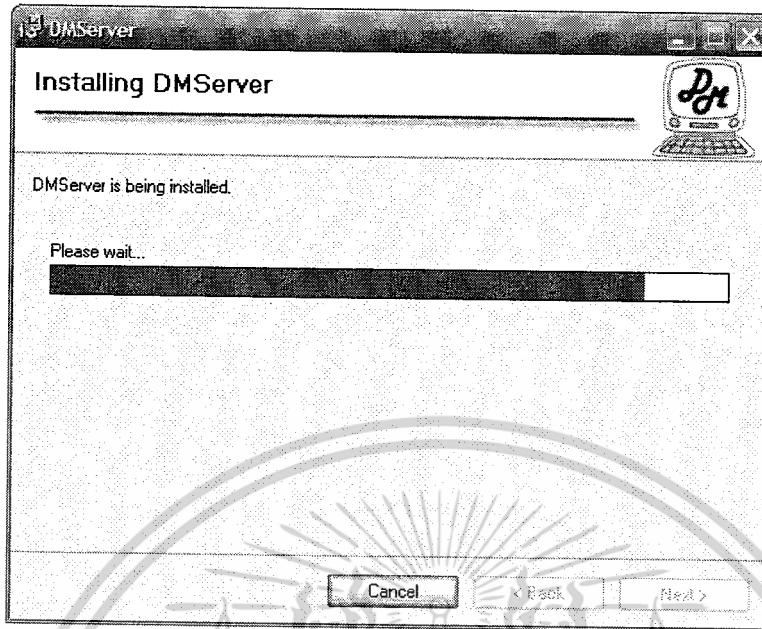
3. คลิก “Next” จะปรากฏหน้าต่างแสดงความพร้อมติดตั้ง



รูปที่ ก.3 หน้าต่างแสดงความพร้อมการติดตั้งซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ให้คลิก “Next” เพื่อเริ่มทำการติดตั้ง พร้อมแสดงสถานะในการติดตั้ง

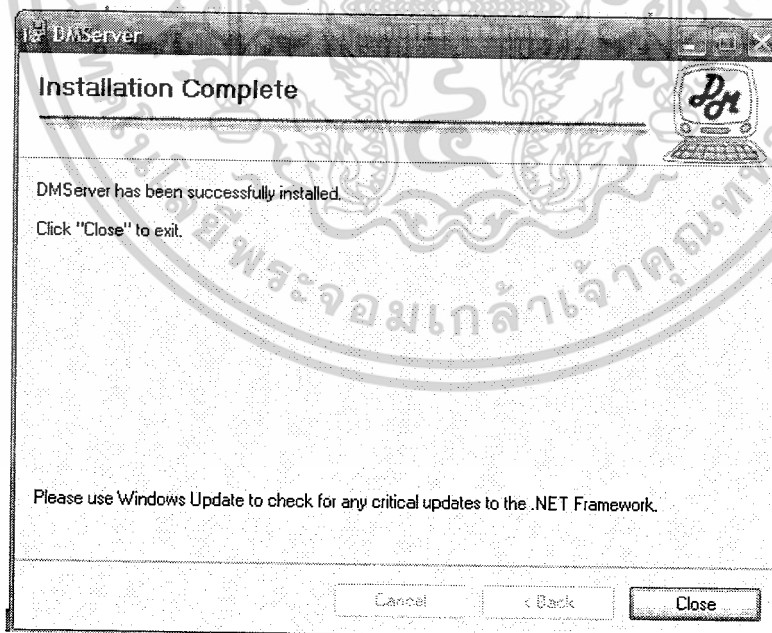


รูปที่ ก.4 หน้าต่างสถานะในการติดตั้งซอฟต์แวร์

5. เมื่อการติดตั้งเสร็จสิ้น

จะปรากฏหน้าต่างแสดงข้อความการติดตั้งสมบูรณ์ ให้คลิก “Close” เพื่อสิ้นสุดการ

ติดตั้ง



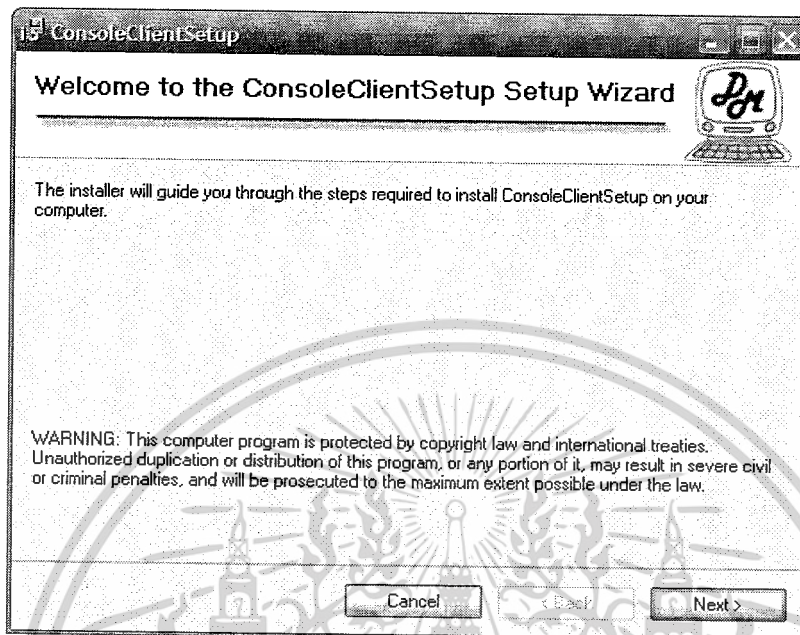
รูปที่ ก.5 หน้าต่างแสดงการติดตั้งซอฟต์แวร์เสร็จสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2.2 การติดตั้งโปรแกรมที่เครื่องลูกข่าย

1. ทำการติดตั้ง ซอฟต์แวร์

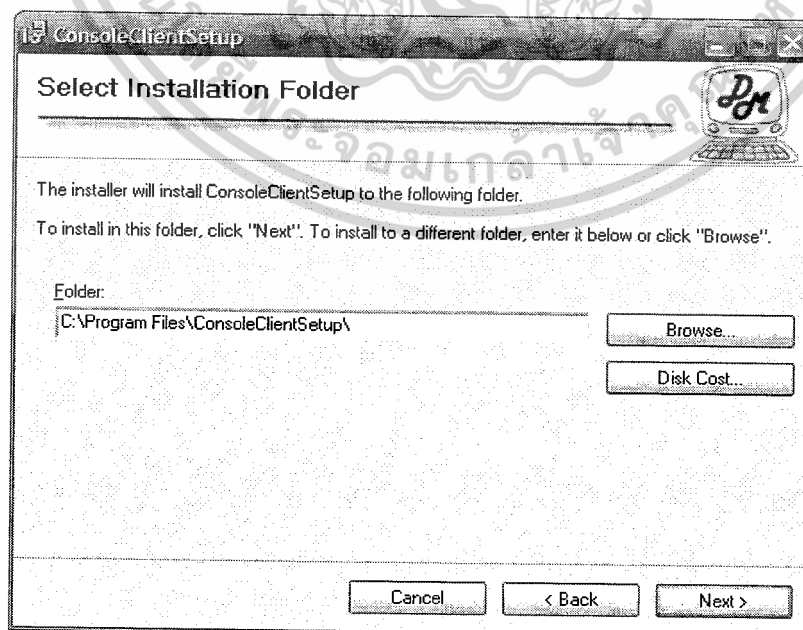
โดยการรันไฟล์สำหรับติดตั้งที่ชื่อว่า setup.exe จะปรากฏหน้าต่างเริ่มการติดตั้ง



รูปที่ ก.6 หน้าต่างเพื่อเริ่มการติดตั้งซอฟต์แวร์ ConsoleClientSetup

2. คลิก "Next" เพื่อทำการติดตั้งต่อ

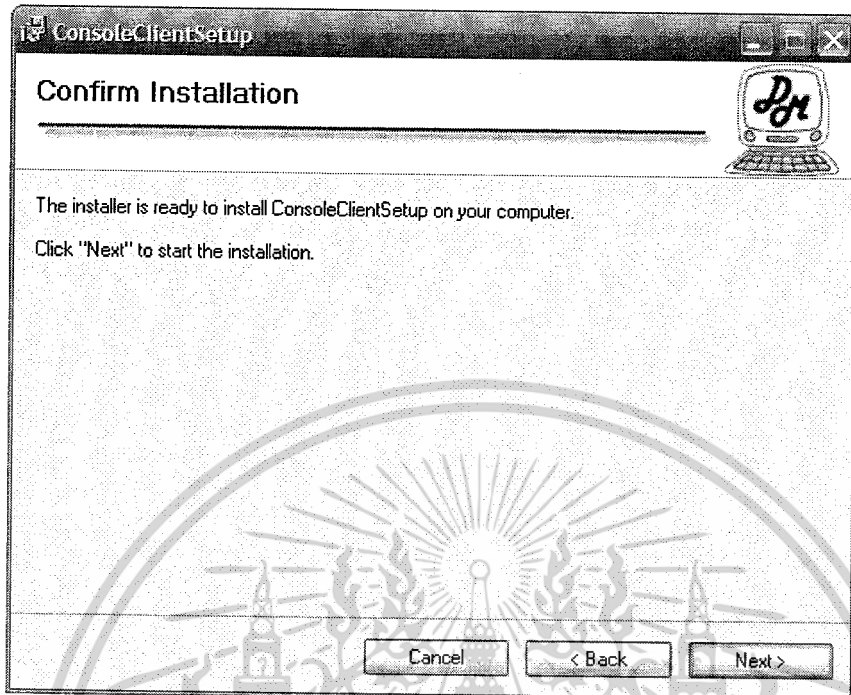
จะปรากฏหน้าต่างกำหนดพาร์ทของโฟลเดอร์ที่ต้องการจัดเก็บไฟล์ โดยปกติซอฟต์แวร์ จะทำการกำหนดพาร์ทของโฟลเดอร์ให้โดยอัตโนมัติ หากต้องการเปลี่ยน ให้กดปุ่ม "Browse..." เพื่อเลือกพาร์ทใหม่ตามต้องการ



รูปที่ ก.7 หน้าต่างสำหรับกำหนดพาร์ทของโฟลเดอร์

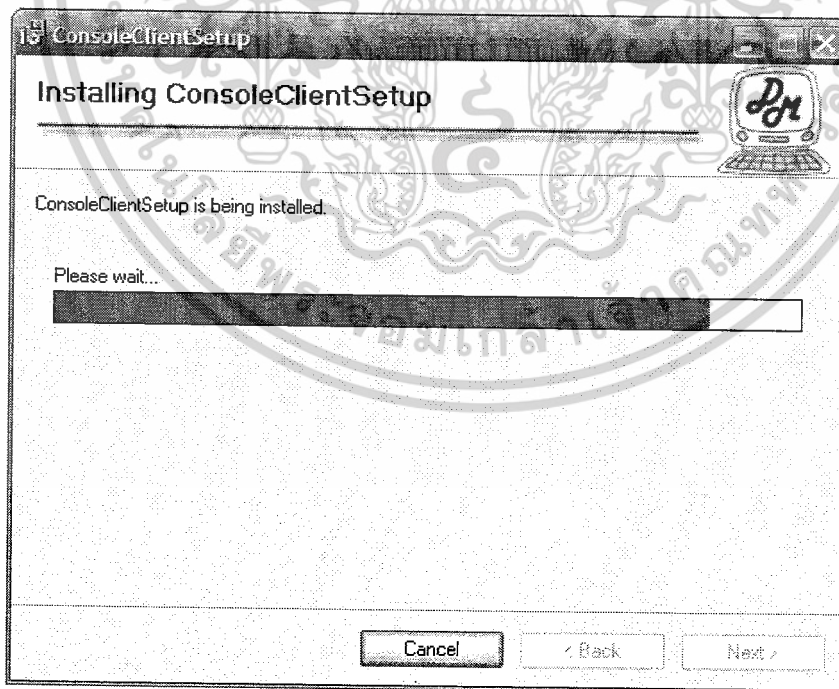
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิก “Next” จะปรากฏหน้าต่างแสดงข้อความพร้อมติดตั้ง



รูปที่ ก.8 หน้าต่างแสดงข้อความพร้อมการติดตั้งซอฟต์แวร์

4. ให้คลิก “Next” เพื่อเริ่มทำการติดตั้ง พร้อมแสดงสถานะในการติดตั้ง

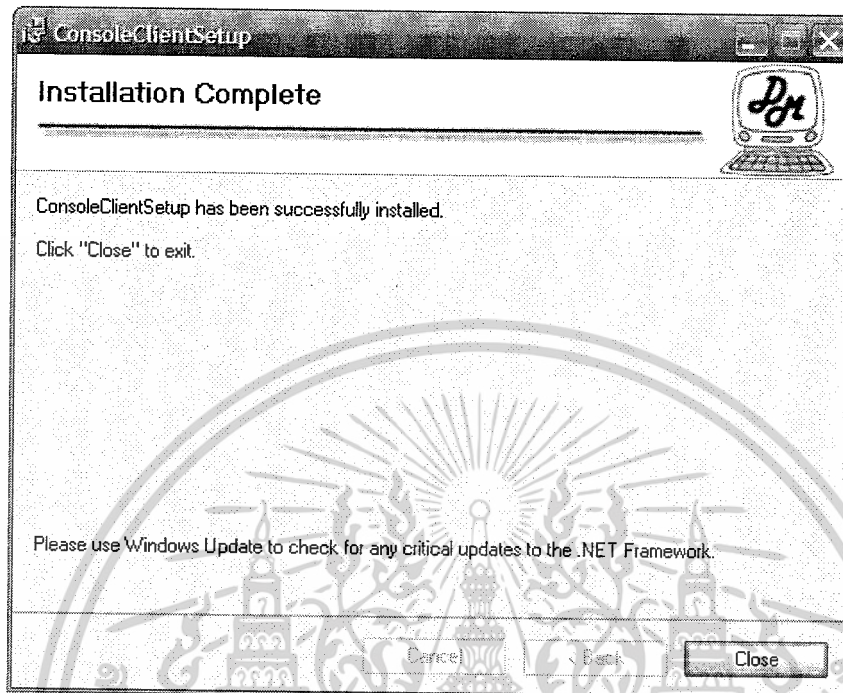


รูปที่ ก.9 หน้าต่างสถานะในการติดตั้งซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เมื่อการติดตั้งเสร็จสิ้น

จะปรากฏหน้าต่างแสดงข้อความการติดตั้งสมบูรณ์ ให้คลิก“Close” เพื่อสิ้นสุดการติดตั้ง



รูปที่ ก.10 หน้าต่างแสดงการติดตั้งซอฟต์แวร์เสร็จสิ้น

ก.3 การใช้งานโปรแกรม

ก.3.1 สัญลักษณ์ของเครื่องลูกข่าย

เมื่อมีการตรวจพบเครื่องลูกข่ายในระบบ ที่หน้าต่างของเครื่องเซิร์ฟเวอร์ จะแสดงสัญลักษณ์ของเครื่องลูกข่ายที่ตรวจพบ โดยสัญลักษณ์ของเครื่องลูกข่ายจะแบ่งตามสถานะของเครื่องลูกข่าย ดังนี้

ก.3.1.1 เครื่องลูกข่ายเปิดอยู่ (ออนไลน์) คือเครื่องลูกข่ายที่มีการเปิดใช้งานอยู่ มีหน้าจอเป็นสีฟ้า



รูปที่ ก.11 เครื่องลูกข่ายเปิดอยู่

ก.3.1.2 เครื่องลูกข่ายปิดอยู่ (ออฟไลน์) คือ เครื่องลูกข่ายที่ไม่ได้ใช้งานอยู่ มีหน้าจอเป็นสีดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.12 เครื่องลูกข่ายปิดอยู่

ก.3.1.3 เครื่องลูกข่ายที่ยังไม่ได้ตั้งค่า คือ เครื่องลูกข่ายที่ยังไม่ได้กำหนด IP Address มีหน้าจอเป็นสีเหลือง

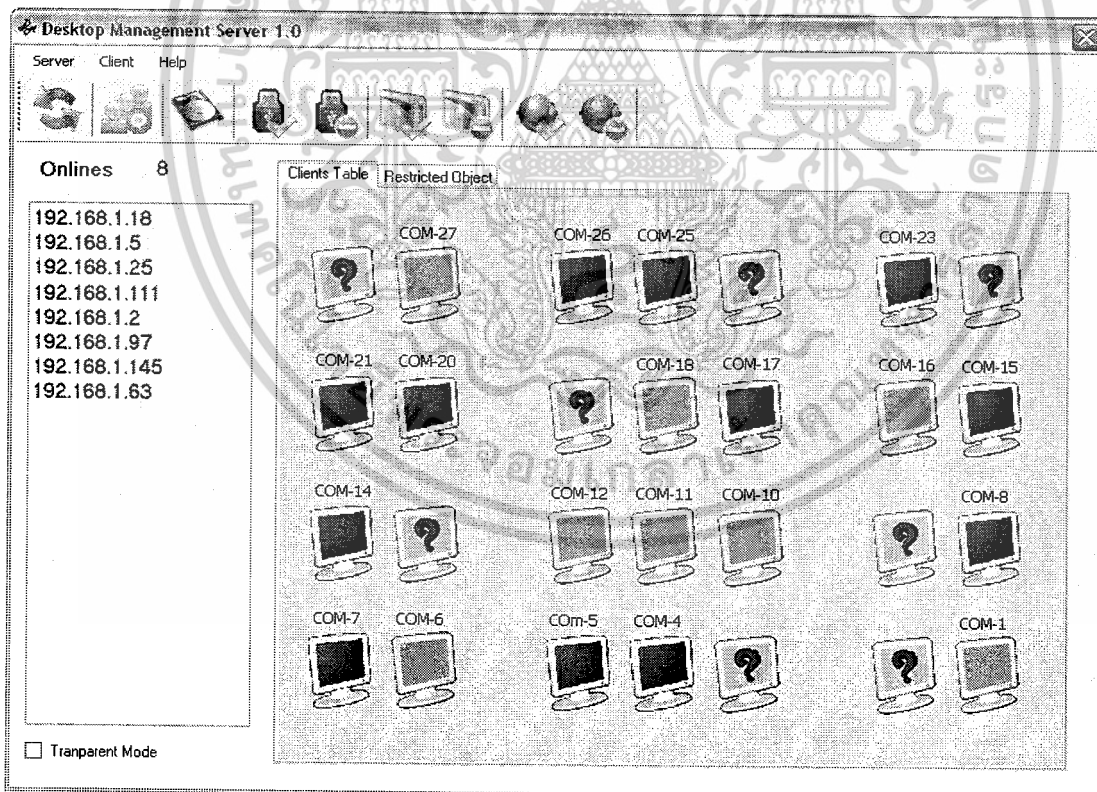


รูปที่ ก.13 เครื่องลูกข่ายที่ยังไม่ได้ตั้งค่า

ก.3.2 หน้าต่างควบคุม

ก.3.2.1 หน้าต่างหลักของโปรแกรม

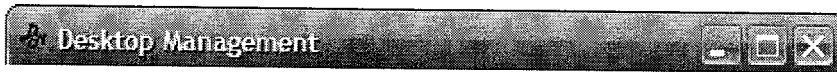
เป็นหน้าต่างหลักที่ใช้ในการควบคุม ดูแลเครื่องลูกข่าย



รูปที่ ก.14 หน้าต่างหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Title Bar** แสดงชื่อหน้าต่างควบคุม



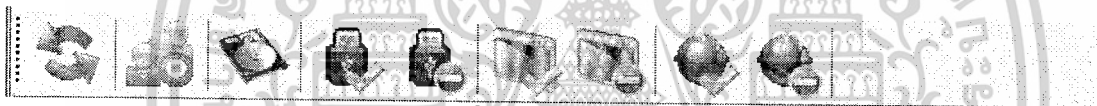
รูปที่ ก.15 Title Bar

- **Menu Bar** แถบเมนู ประกอบด้วยชุดเมนูย่อยซึ่งอยู่ภายใต้เมนูหลัก ที่ใช้สำหรับการเข้าไปใช้เมนูต่างๆ



รูปที่ ก.16 Menu Bar

- **Tool Bar** แถบเครื่องมือ เป็นชุดเครื่องมือสั่งงาน โดยอยู่ในรูปแบบ Icon เครื่องมือต่างๆ สามารถอำนวยความสะดวกในการใช้งานให้ทำงานได้รวดเร็วยิ่งขึ้น



รูปที่ ก.17 Tool Bar

- ส่วนแสดง IP Address เป็นส่วนที่แสดงว่า IP อะไรกำลังออนไลน์ และมีจำนวนที่ออนไลน์ทั้งหมดเท่าไร

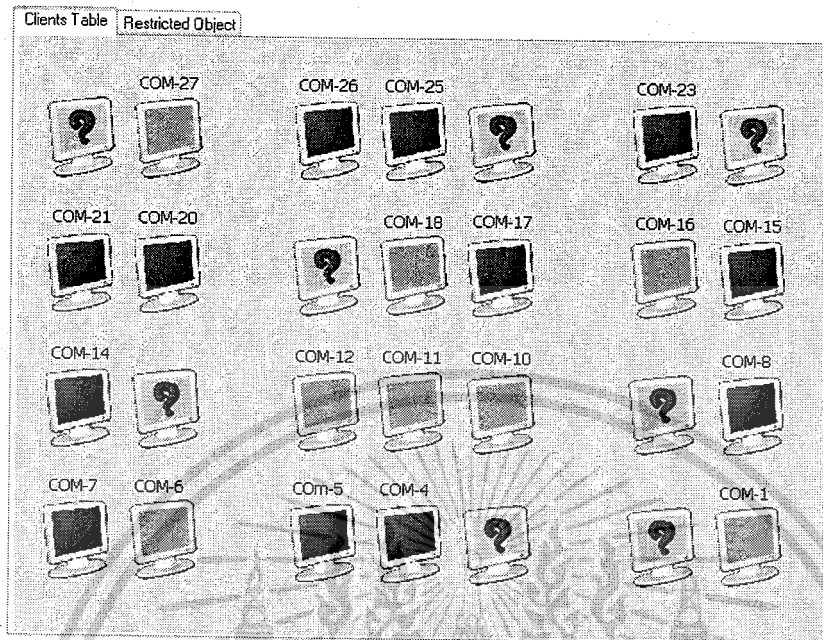
Onlines 8

192.168.1.18
192.168.1.5
192.168.1.25
192.168.1.111
192.168.1.2
192.168.1.97
192.168.1.145
192.168.1.63

รูปที่ ก.18 ส่วนแสดงส่วนแสดง IP Address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนแสดงตำแหน่งของเครื่องลูกข่าย เป็นส่วนที่แสดงว่าเครื่องลูกข่ายที่ตำแหน่งนั้นๆ มีสถานะอย่างไร



รูปที่ ก.19 ส่วนแสดงตำแหน่งของเครื่องลูกข่าย

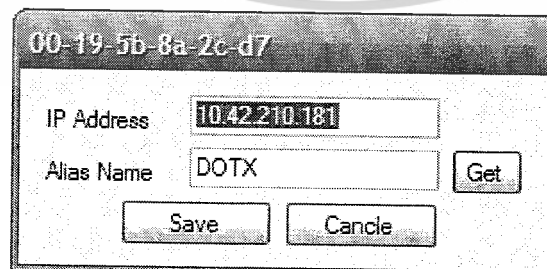
- ส่วนการทำ Transparent Mode

Transparent Mode

รูปที่ ก.20 ส่วนแสดง Transparent Mode

ก.2.2.2 หน้าต่างกำหนดค่าเครื่องลูกข่าย

ป้อน IP Address และเลือก “Get” เพื่อดึงค่า Alias Name จากเครื่องลูกข่าย และกด “Save” เพื่อบันทึก



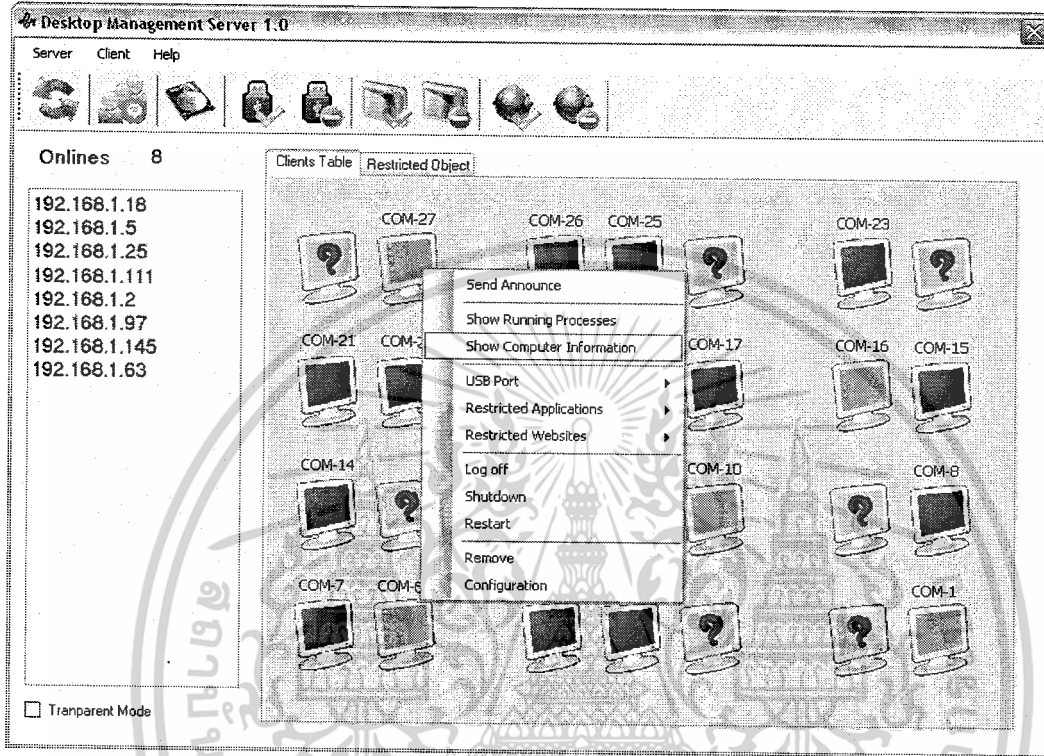
รูปที่ ก.21 Client Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

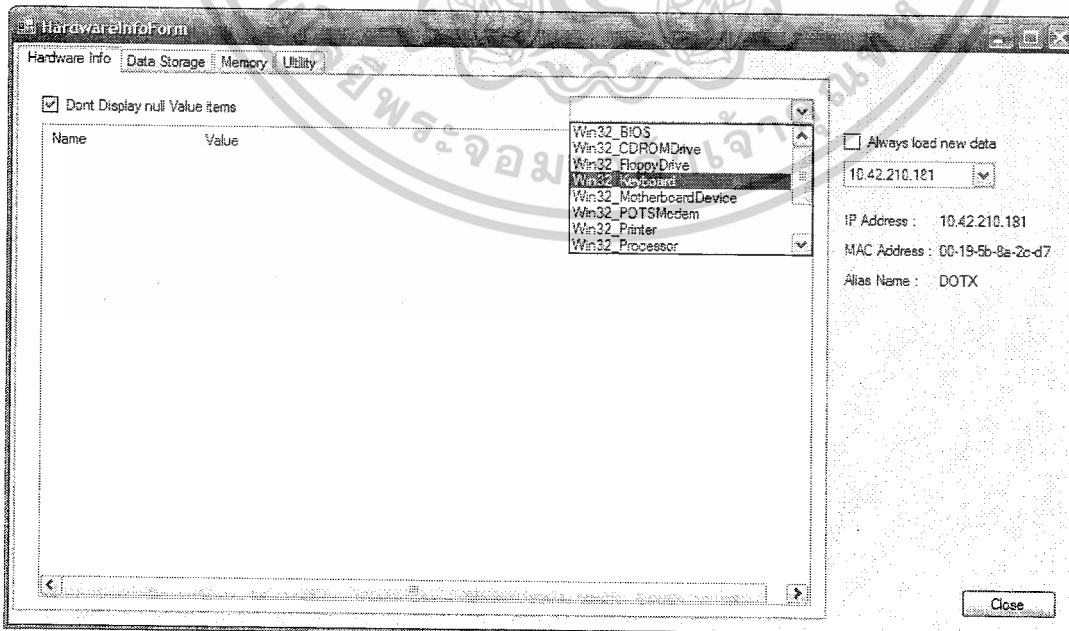
ก.4 การใช้คำสั่งต่างๆของโปรแกรม

ก.4.1 Inventory Management Module

สำหรับดูฮาร์ดแวร์ และซอฟต์แวร์ของเครื่องลูกข่าย โดยสามารถดูได้จากการคลิกขวาเครื่องลูกข่ายที่ต้องการ แล้วเลือก “Show Hardware Information” ดังรูปที่ ก.22



รูปที่ ก.22 การเลือก “Show Hardware Information”



รูปที่ ก.23 หน้าต่าง Hardware Information

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง Hardware Information จะปรากฏขึ้นดังรูปที่ ก.23 ซึ่งสามารถเลือกได้ว่าจะให้แสดงค่าต่างๆดังนี้

ก.4.1.1 Hardware Info

- BIOS
- CD ROM Drive
- Floppy Drive
- Keyboard
- Motherboard Device
- POTS Modem
- Printer
- Processor
- Serial Port
- Sound Device
- USB Controller
- Video Controller

ก.4.1.2 Storage

- Disk Drive
- Disk Partition
- Logical Disk

ก.4.1.3 Memory

- Cache Memory
- Logical Memory Configuration
- Physical Memory

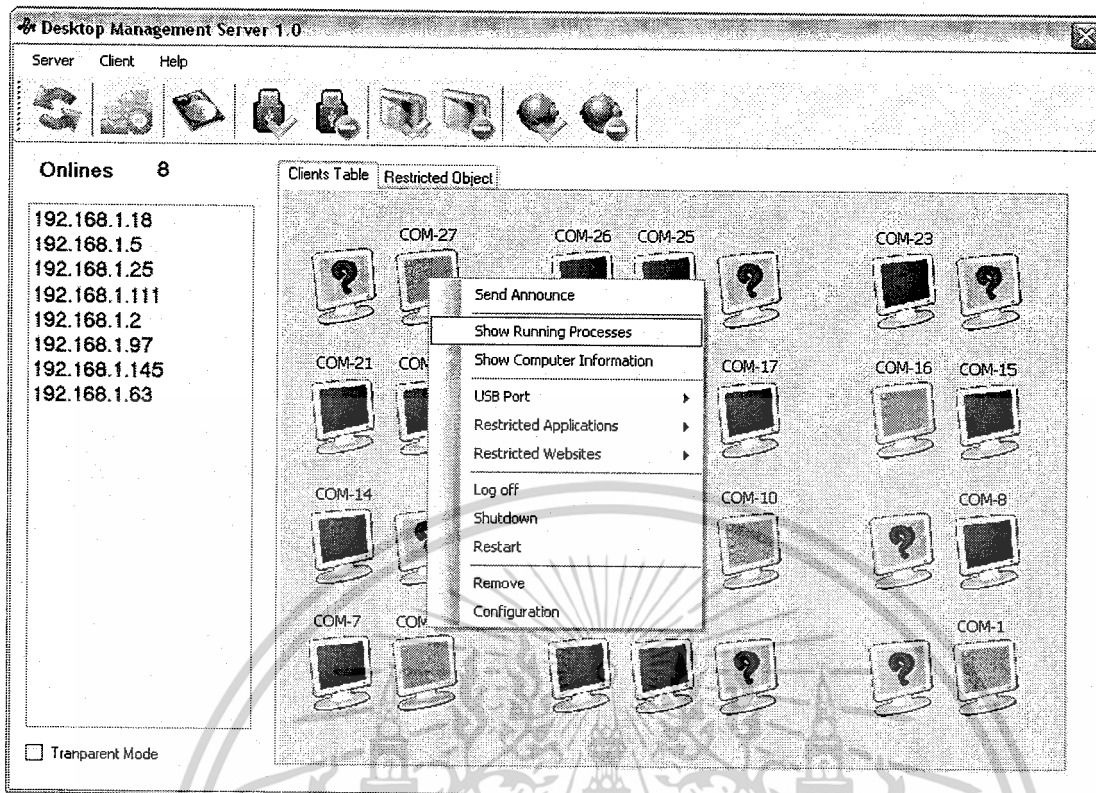
ก.4.1.6 Utility

- Account
- Network Adapter
- Product
- Service

ก.4.2 Processes Management Module

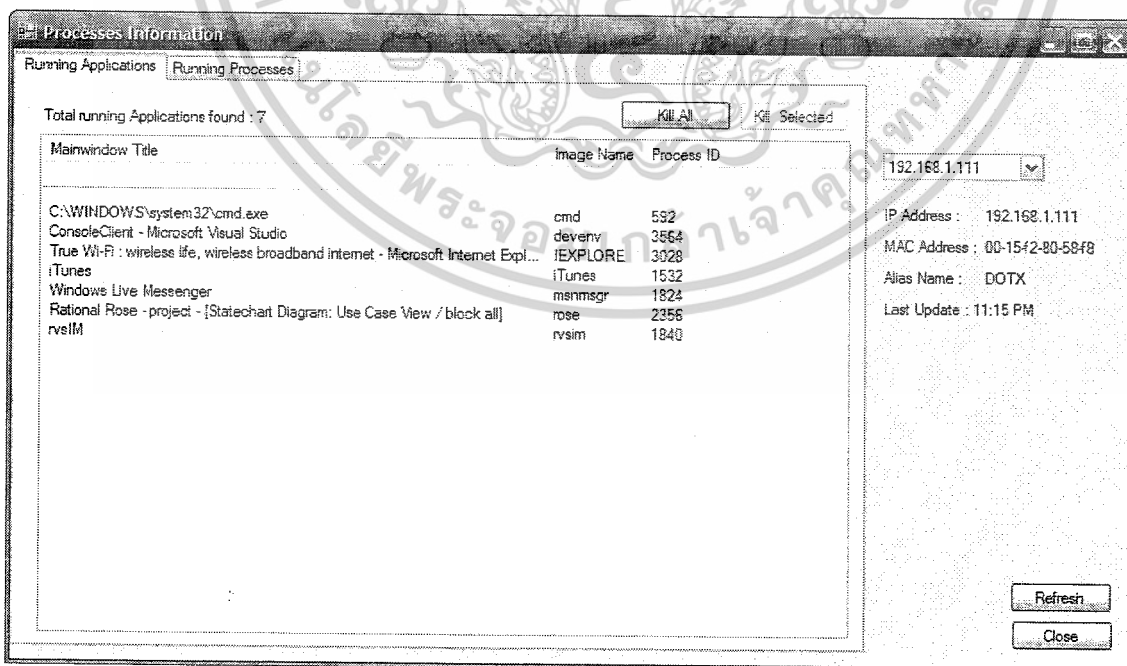
สำหรับจัดการ โพรเซสของเครื่องลูกข่าย โดยสามารถดูได้จากการคลิกขวาเครื่องลูกข่ายที่ต้องการ แล้วเลือก “Show Running Process” ดังรูปที่ ก.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.24 การเลือก “Show Running Process”

หน้าต่าง Processes Information จะปรากฏขึ้นดังรูปที่ ก.25 ซึ่งสามารถที่จะปิดโปรแกรมที่ทำงานอยู่ได้



รูปที่ ก.25 หน้าต่าง Processes Information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

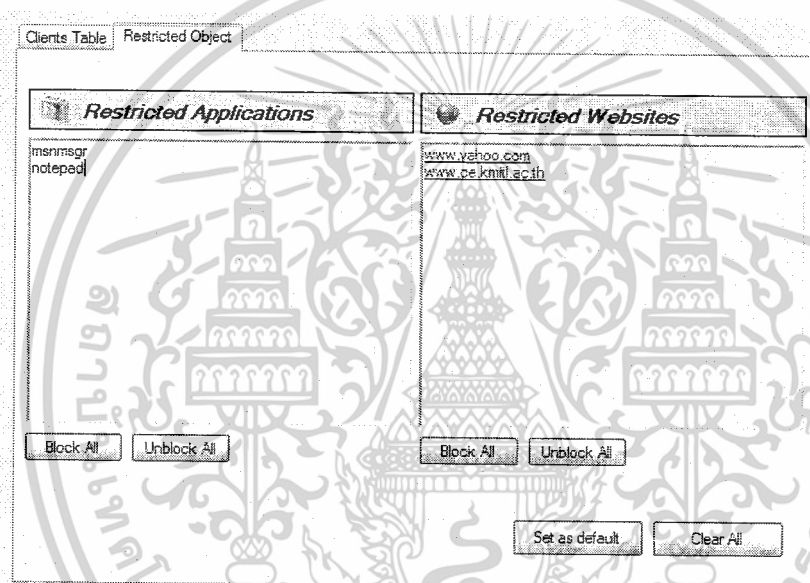
ก.4.3 Web Control Module

สำหรับควบคุมการเข้าเว็บไซต์ของเครื่องลูกข่ายให้ไม่สามารถเข้าเว็บไซต์ตามรายชื่อที่กำหนด หรือไม่สามารถเข้าเว็บไซต์ได้ทั้งหมด



รูปที่ ก.26 ไอคอน “Website”

สามารถกำหนดได้ว่าไม่อนุญาตให้เข้าเว็บไซต์อะไร โดยกำหนดที่ Restricted Websites



รูปที่ ก.27 หน้าต่าง Restricted Object

เมื่อมีการกำหนด “Restricted Website” จะมีสัญลักษณ์ปรากฏขึ้นที่เครื่องลูกข่าย เพื่อแสดงว่าเครื่องลูกข่ายเครื่องนั้น ได้มีการไม่อนุญาตให้ใช้โปรแกรม

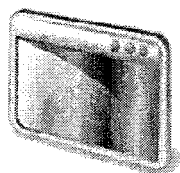


รูปที่ ก.28 สัญลักษณ์มีการไม่อนุญาตให้ใช้เว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.4.4 Application Control Module

สำหรับการควบคุมการใช้โปรแกรมที่เครื่องลูกข่ายให้ไม่สามารถใช้งานโปรแกรมตามรายชื่อโปรแกรมที่กำหนด



รูปที่ ก.29 ไอคอน “Application”

สามารถกำหนดได้ว่าไม่อนุญาตให้เข้าเว็บไซต์อะไร โดยกำหนดที่ Restricted Application ดังรูปที่ ก.27 และเมื่อมีการกำหนด “Restricted Application” จะมีสัญลักษณ์ปรากฏขึ้นที่เครื่องลูกข่าย เพื่อแสดงว่าเครื่องลูกข่ายเครื่องนั้นได้มีการไม่อนุญาตให้ใช้โปรแกรม



รูปที่ ก.30 สัญลักษณ์มีการไม่อนุญาตให้ใช้โปรแกรม

ก.4.5 USB Port Control Module

สำหรับควบคุมการใช้งานพอร์ต USB ที่เครื่องลูกข่ายให้สามารถใช้งาน หรือ ไม่สามารถใช้งานพอร์ต USB โดยกำหนดที่ไอคอน “USB” (กำหนดการใช้พอร์ต USB ทุกเครื่อง) ดังรูปที่ ก.31 หรือเลือกเครื่องลูกข่ายที่ต้องการและคลิกขวา เลือก “USB Port”



รูปที่ ก.31 ไอคอน “USB”

เมื่อเลือก “Block USB” จะมีสัญลักษณ์ปรากฏขึ้นที่เครื่องลูกข่าย เพื่อแสดงว่าเครื่องลูกข่ายเครื่องนั้นได้ถูกกำหนดไม่อนุญาตให้ใช้พอร์ต USB



รูปที่ ก.32 สัญลักษณ์การไม่อนุญาตให้ใช้พอร์ต USB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.4.6 Send Announce

สำหรับการส่งข้อความถึงเครื่องลูกข่าย สามารถส่งหาเครื่องลูกข่ายได้ทั้งหมด โดยเลือกไอคอนดังรูปที่ ก.33



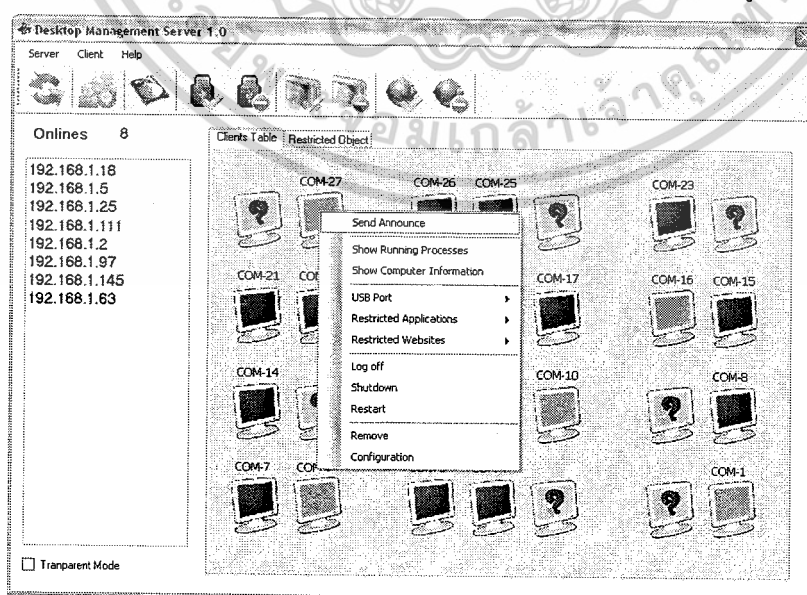
รูปที่ ก.33 ไอคอนการส่งข้อความประกาศ

จะมีหน้าต่างให้กรอกข้อมูลที่ต้องการประกาศ ซึ่งสามารถกำหนดระยะเวลาในการแสดงข้อความในช่อง Show Time ดังรูปที่ ก.34



รูปที่ ก.34 หน้าต่างการประกาศของเครื่องลูกข่าย

และสามารถส่งข้อความประกาศไปยังเครื่องลูกข่ายแต่ละเครื่องได้ ดังรูป ก.35



รูปที่ ก.35 การส่งข้อความไปยังเครื่องลูกข่ายนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้