โปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตวิชาที่เรียน

AN APPLICATION FOR WARNING USING A COMPUTER ABOUT CLASSROOM MISBEHAVIOR

นายศุภักษร ฟองจางวาง รหัส 51362152

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2554

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน โปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตวิชาที่เรียน
ผู้ดำเนินโครงงาน นายศุภักษร ฟองจางวาง รหัส 51362152
ที่ปรึกษาโครงงาน ดร.สุรเดช จิตประไพกุลศาล
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2554

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวรอนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

.....ประธานกรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)

.....กรรมการ
(ดร.วรลักษณ์ คงเด่นฟ้า)

.....กรรมการ
(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ
(อาจารย์เศรษฐา ตั้งค้าวานิช)

ชื่อหัวข้อโครงงาน	โปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตวิชาที่เรียน
ผู้ดำเนินโครงงาน	นายศุภักษร ฟองจางวาง รหัส 51362152
ที่ปรึกษาโครงงาน	ดร.สุรเดช จิตประไพกุลศาล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

.....

บทคัดย่อ

โครงงานนี้เป็นการพัฒนาโปรแกรมสำหรับช่วยแจ้งเตือนการเข้าเว็บไซต์ที่ไม่อนุญาตให้เข้า ซึ่งอยู่นอกขอบเขตวิชาที่เรียน และเพื่อใช้เป็นเครื่องมือสำหรับอาจารย์ผู้สอนในการตรวจจับการเข้าเว็บไซต์ที่ไม่เหมาะสมของผู้เรียนในระหว่างการเรียนการสอน โดยมีการพัฒนาโปรแกรม 2 โปรแกรม ประกอบด้วยโปรแกรมที่เครื่องแม่ข่าย ซึ่งมีหน้าที่สั่งงานโปรแกรมที่เครื่องลูกข่ายและแจ้งเตือนการเข้าเว็บไซต์ที่ไม่เหมาะสมให้กับอาจารย์ผู้สอน และโปรแกรมที่เครื่องลูกข่าย ซึ่งมีหน้าที่ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า โดยใช้วิธีการดักจับและวิเคราะห์แพ็คเกต (Packet) และส่งข้อมูลการแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่าย

ผลที่ได้จากการพัฒนาและทดสอบโปรแกรม คือ โปรแกรมสามารถตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าแล้วแจ้งเตือนออกทางหน้าจอพร้อมกับแจ้งเตือนด้วยเสียงเป็นคำพูดได้ นอกจากนี้โปรแกรมยังสามารถบันทึกการเข้าเว็บไซต์ทั้งหมดลงฐานข้อมูลเพื่อนำมาวิเคราะห์ได้และสามารถตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานได้อีกด้วย

Project title	An application for warning using a computer about classroom misbehavior.
Name	Mr. Supuksorn Fongjangwang ID: 51362152
Project advisor	Dr. Suradet Jitprapaikulsarn
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2011

Abstract

The main purpose of this project is to develop a program for instructors to detect the misuse of Internet by students during class time. Two programs were developed: server and client. The server controls the client works and warns the instructor about students who visit inappropriate website. The client detects the surfing behavior by capturing and analyzing packets and sending message to the server program.

Based on our experiments, the two programs work very well together detecting misuse of Internet by students. Once the misuse is detected, the sound alarm goes off by the program warn the instructors about the misbehavior. In addition, the program can record the URL of the website visited by student for further analysis. The client program can also detect the usage of inappropriate programs by students.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์เล่มนี้สำเร็จลุล่วงมาได้นั้น เนื่องจากความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงงาน คือ ดร.สุรเดช จิตประไพกุลศาล ที่กรุณาให้คำปรึกษา คำแนะนำ และความรู้ในการพัฒนาโปรแกรม การทดสอบโปรแกรม พร้อมทั้งเสนอแนวทางแก้ไขปัญหาที่เกิดขึ้นตลอดระยะเวลาการทำโครงงาน ผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ทั้งนี้ ขอขอบพระคุณอาจารย์ซึ่งเป็นคณะกรรมการทั้ง 3 ท่าน ได้แก่ ดร.วรลักษณ์ คงเด่นฟ้า อาจารย์ภาณุพงศ์ สอนคม และอาจารย์เศรษฐา ตั้งค้าวาณิชย์ ที่ช่วยให้คำแนะนำ และแนวทางการแก้ไขโครงงานนี้ให้มีความสมบูรณ์ยิ่งขึ้น และการทำโครงงานนี้จะเสร็จสมบูรณ์ได้นั้น ต้องขอขอบคุณ คุณไตรรงค์ เรืองพิพัฒน์ เจ้าหน้าที่ดูแลห้องปฏิบัติการคอมพิวเตอร์ EN616 ที่อำนวยความสะดวกในการใช้ห้องเพื่อทดสอบโปรแกรมเป็นระยะเวลาเกือบ 1 เดือน

สุดท้ายนี้ขอขอบพระคุณบิดา มารดาซึ่งเป็นกำลังใจอันสำคัญยิ่งในการทำโครงงานนี้ให้สำเร็จลุล่วงด้วยดี

นายศุภักษร ฟองจางวาง

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

หน้า

สารบัญรูป

หน้า

สารบัญรูป (ต่อ)

หน้า

สารบัญรูป (ต่อ)

หน้า

สารบัญรูป (ต่อ)

หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันการใช้คอมพิวเตอร์มีบทบาทและมีความสำคัญต่อชีวิตประจำวันของคนเราเป็นอย่างมาก ในการอำนวยความสะดวกในชีวิตประจำวันหลายๆ ด้าน โดยเฉพาะอย่างยิ่งการใช้คอมพิวเตอร์ในสถาบันการศึกษา ซึ่งส่วนใหญ่ได้มีการเชื่อมต่อคอมพิวเตอร์เข้ากับเครือข่ายอินเทอร์เน็ตเพื่อใช้ในการศึกษา ค้นคว้าหาความรู้ให้เกิดความสะดวกและรวดเร็วมากขึ้น แต่เมื่อมีการนำคอมพิวเตอร์เชื่อมต่อเข้ากับเครือข่ายอินเทอร์เน็ตมาใช้ในการเรียนการสอนแล้ว พบว่าผู้เรียนไม่ตั้งใจเรียนเท่าที่ควร เนื่องจากในระหว่างที่ผู้สอนกำลังดำเนินการสอนโดยมีการใช้งานคอมพิวเตอร์ในการช่วยสอนนั้น ผู้เรียนได้ให้ความสนใจต่อสิ่งอื่นๆ ที่อยู่นอกขอบเขตของวิชาที่เรียน เช่น การเปิดเว็บไซต์ที่ไม่เหมาะสม ซึ่งอาจทำให้ให้ผู้เรียนไม่เข้าใจในเนื้อหาที่เรียน และอาจทำให้ผลการเรียนตกต่ำได้

ดังนั้น ผู้จัดทำจึงได้คิดพัฒนาโปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตวิชาที่เรียนนี้ขึ้นมา เพื่อช่วยแก้ปัญหาที่ผู้เรียนเข้าเว็บไซต์ที่ไม่เหมาะสมขณะที่ผู้สอนกำลังสอนอยู่ โดยโปรแกรมจะคอยตรวจจับเว็บไซต์ที่มีการเข้าทั้งหมดและตรวจสอบว่าเป็นเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือไม่ เมื่อตรวจสอบได้ว่ามีการเข้าเว็บไซต์ที่ไม่อนุญาตให้เข้า โปรแกรมก็จะแจ้งเตือนที่หน้าจอพร้อมทั้งส่งเสียงเตือนเป็นคำพูดได้ว่าข้อมูลการแจ้งเตือนนี้มาจากเครื่องไหน และผู้เรียนกำลังเข้าเว็บไซต์อะไรอยู่ในขณะนั้น โดยที่ผู้สอนไม่จำเป็นต้องคอยตรวจสอบที่หน้าจออยู่ตลอดเวลา

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อพัฒนาโปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตวิชาที่เรียน

1.2.2 เพื่อให้ผู้สอนนำโปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตวิชาที่เรียนไปใช้เป็นเครื่องมือในการเรียนการสอน

1.3 ขอบข่ายของโครงงาน

1.3.1 ขอบข่ายของโปรแกรมที่เครื่องแม่ข่าย

1.3.1.1 โปรแกรมสามารถค้นหาหมายเลขไอพีและชื่อเครื่องของเครื่องลูกข่ายทั้งหมดได้

1.3.1.2 โปรแกรมสามารถสั่งโปรแกรมที่เครื่องลูกข่ายให้ตรวจจับและหยุดตรวจจับเว็บไซต์ได้

1.3.1.3 โปรแกรมสามารถตรวจสอบสถานะของเครื่องลูกข่ายและสถานะของโปรแกรมที่เครื่องลูกข่ายได้

1.3.1.4 โปรแกรมสามารถแจ้งเตือนด้วยข้อความที่หน้าจอและแจ้งเตือนด้วยเสียงได้

1.3.1.5 โปรแกรมสามารถส่งไฟล์นามสกุล .txt ไปยังเครื่องลูกข่ายได้

1.3.1.6 โปรแกรมสามารถบันทึกข้อมูลการเข้าเว็บไซต์ลงฐานข้อมูลได้

1.3.2 ขอบข่ายโปรแกรมที่เครื่องลูกข่าย

1.3.2.1 โปรแกรมสามารถตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าได้

1.3.2.2 โปรแกรมสามารถตรวจจับเว็บไซต์สำหรับบันทึกข้อมูลเพื่อนำมาวิเคราะห์ได้

1.3.2.3 โปรแกรมสามารถส่งข้อมูลการแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่ายได้

1.3.3.4 โปรแกรมสามารถรับคำสั่งจากโปรแกรมที่เครื่องแม่ข่ายและตอบกลับได้

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาความเป็นไปได้ของโครงงานและกำหนดขอบเขตของโครงงาน

1.4.2 ศึกษาการทำงานของข่ายงานบริเวณเฉพาะที่ (LAN) และการทำงานของโปรโตคอลต่างๆ ที่เกี่ยวข้อง

1.4.3 ศึกษาการเขียนโปรแกรมแบบ Socket Programing

1.4.4 ศึกษาการเขียนโปรแกรมแบบ Multi-Threading

1.4.5 ออกแบบและพัฒนาโปรแกรมตามเป้าหมายที่วางไว้

1.4.6 ทดสอบการทำงานของโปรแกรม รวมทั้งแก้ไขข้อผิดพลาดต่างๆ ที่เกิดขึ้น

1.4.7 สรุปผลและจัดทำรายงานโครงงาน

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

รายละเอียด	ปี 2554							ปี 2555	
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ศึกษาความเป็นไปได้ของโครงงานและกำหนดขอบเขตของโครงงาน	↔								
2. ศึกษาการทำงานของข่ายงานบริเวณเฉพาะที่ (LAN) และการทำงานของโปรโตคอลต่างๆ ที่เกี่ยวข้อง		↔							
3. ศึกษาการเขียนโปรแกรมแบบ Socket Programing			↔						
4. ศึกษาการเขียนโปรแกรมแบบ Multi-Threading				↔					
5. ออกแบบและพัฒนาโปรแกรมตามเป้าหมายที่วางไว้				↔	→	→	→		
6. ทดสอบการทำงานของโปรแกรมรวมทั้งแก้ไขข้อผิดพลาดต่างๆ ที่เกิดขึ้น							↔	→	
7. สรุปผลและจัดทำรายงานโครงงาน								↔	→

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้โปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตของวิชาที่เรียน สำหรับใช้ในการเรียนการสอนที่มีการใช้งานคอมพิวเตอร์

1.6.2 ผู้เรียนเข้าเว็บไซต์ในระหว่างการเรียนการสอนที่เหมาะสมมากขึ้น เนื่องจากรู้ว่ามีโปรแกรมตรวจจับติดตั้งอยู่ที่เครื่อง

1.7 งบประมาณของโครงงาน

1.7.1 ค่าจัดทำรายงาน 500 บาท

1.7.2 ค่าถ่ายเอกสาร 500 บาท

รวม 1,000 บาท

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 อุปกรณ์เครือข่าย (Network Devices)

2.2.1 ฮับ (HUB) [1]

ฮับทำงานในระดับเลเยอร์ (layer) 1 ซึ่งเป็นเลเยอร์เกี่ยวข้องกับเรื่องของการส่งสัญญาณออกไปสู่สื่อกลางที่ใช้ในการสื่อสาร (Media) รวมไปถึงเรื่องของการเข้ารหัสสัญญาณ เพื่อที่จะส่งออกไปเป็นค่าต่างๆในทางไฟฟ้า และเป็นเลเยอร์ที่กำหนดถึงการเชื่อมต่อต่างๆ ที่เป็นไปในทางกายภาพ (Physical) ฮับจะทำงานในลักษณะของการทวนสัญญาณ หมายถึงว่า จะทำการทำซ้ำสัญญาณนั้นอีกครั้ง ซึ่งเป็นคนละอย่างกับการขยายสัญญาณ พอทำแล้วก็จะส่งออกไปยังเครือข่ายที่เชื่อมต่ออยู่ โดยจะมีหลักว่า จะส่งออกไปยังทุกๆ พอร์ต (Port) ยกเว้นพอร์ตที่เป็นตัวส่งสัญญาณออกมา และเมื่อปลายทางแต่ละจุดรับข้อมูลไปแล้ว ก็จะต้องพิจารณาข้อมูลที่ได้มาว่าข้อมูลนั้นส่งมาถึงตัวเองหรือไม่ ถ้าหากไม่ใช่ข้อมูลที่จะส่งมาถึงตัวเอง ก็จะไม่รับข้อมูลที่ส่งมานั้น การทำงานในระดับนี้ ถ้าดูในส่วนของตัวฮับเองนั้น จะเห็นได้ว่าตัวของฮับนั้น เวลาส่งข้อมูลออกไปจะไม่มีการพิจารณาข้อมูล MAC Address ของเลเยอร์ 2 หรือ IP Address ซึ่งเป็นของเลเยอร์ 3 เลย

2.2.2 เราเตอร์ (Router) [2]

เราเตอร์ทำงานในเลเยอร์ (layer) ที่ 3 ของ OSI Model เป็นอุปกรณ์ที่ใช้เชื่อมต่อเครือข่าย 2 เครือข่ายหรือมากกว่าเข้าด้วยกัน ไม่ว่าจะเป็นการเชื่อม LAN เข้ากับ LAN หรือแม้แต่เชื่อม LAN เข้ากับ WAN ก็ตาม เราเตอร์สามารถกรองข้อมูลได้เช่นเดียวกับบริดก์ (Bridge) แต่มีความสามารถมากกว่าตรงที่สามารถหาเส้นทางในการส่งแพ็คเก็ต (Packet) ไปยังเครื่องปลายทางได้สั้นที่สุด เราเตอร์ทำงานที่ระดับ Network layer ของแบบจำลอง (OSI Model)โดยใช้ Logical Address หรือ Network layer Address ซึ่งก็คือ Address ที่ตั้งด้วยซอฟแวร์ตามที่ผู้ใช้แต่ละเครื่องจะตั้งขึ้นให้โปรโตคอล (Protocol) ในระบบ Network layer รู้จัก ในการส่งข้อมูลผ่านโปรโตคอลของเครือข่ายชนิดต่างๆ ไม่ว่าจะเป็น IPX, TCP/IP หรือ Apple Talk ซึ่งเป็นโปรโตคอลที่ทำงานใน Network layer การกำหนด Network Address ทำได้โดยผู้ดูแลระบบเครือข่ายนั้น ทำให้สามารถแก้ไขเปลี่ยนแปลงได้ง่าย

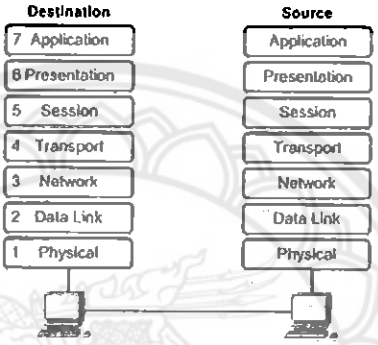
2.2.3 สวิตซ์ (Switch) [3]

สวิตซ์มีอยู่ด้วยกัน 2 ชนิด คือ layer-2 Switch และ layer-3 Switch ดังรายละเอียดต่อไปนี้ layer-2 Switch หรือ L2 Switch ก็คือ บริดก์ (Bridge) แต่เป็นบริดก์ที่มีอินเตอร์เฟซ (Interface) ในการเชื่อมต่อกับเซกเมนต์ (Segment) มากขึ้นทำให้สามารถแบ่งเครือข่าย LAN ออกเป็นเซกเมนต์ย่อยๆ เพื่อประโยชน์ในการบริหารจัดการเครือข่ายได้ดียิ่งขึ้น และประสิทธิภาพในการทำงานของ L2 Switch ก็สูงกว่าบริดก์ ทำให้ในปัจจุบันนิยมใช้งาน L2 Switch ส่วน L3 Switch ก็คือ เราเตอร์ (Router) ที่ได้รับการปรับปรุงให้มีประสิทธิภาพสูงขึ้น แต่มีราคาถูกลง โดย L3 Switch นี้จะสามารถจัดการกับเครือข่ายที่มีเซกเมนต์มากๆ ได้ดีกว่าเราเตอร์ในระดับของเลเยอร์ (layer) 2 ซึ่งเป็นการทำงานในระดับของ Data Link layer

ในกรณีของอีเทอร์เน็ต (Ethernet) นั้น ก็จะมีความเกี่ยวข้องกับเรื่องของเฟรม (Frame) และ MAC Address LLC Switch เป็นอุปกรณ์ที่มีหลักการในการทำงานคือจะส่งข้อมูลจากพอร์ต (Port) หนึ่ง ไปยังปลายทางที่เฉพาะเจาะจงเท่านั้น ข้อมูลนั้นจะไม่ถูกส่งออกไปยังพอร์ตอื่นๆ ยกเว้นมีความจำเป็นในบางกรณี เช่น ข้อมูลที่ส่งไม่มีผู้รับที่เชื่อมต่ออยู่ในสวิตซ์ของตัวเอง หรือข้อมูลที่ต้องส่งนั้นเป็นข้อมูลที่ต้องส่งออกไปในลักษณะของ Broadcast หรือ Multicast การที่พอร์ตใดๆ จะส่งข้อมูลถึงกันนั้น สวิตซ์ก็จะทำการตรวจสอบ MAC Address ของอุปกรณ์ที่เชื่อมต่อกันอยู่ และมีการทำตาราง (Table) เอาไว้เพื่อเก็บข้อมูลเหล่านี้ และเมื่อเวลามีการส่งข้อมูลระหว่างกัน ก็จะเอา MAC Address ปลายทางที่อยู่ในส่วนหัว (Header) ของเฟรมมาเทียบกับตารางที่ตัวเองมีอยู่ ซึ่งถ้าหากว่ามีข้อมูล MAC Address ตรงกับที่มีอยู่ในตาราง และได้มีการบันทึกเอาไว้ว่าเป็นของอุปกรณ์ที่เชื่อมต่ออยู่กับพอร์ตไหน สวิตซ์ก็จะทำการส่งข้อมูลไปยังพอร์ตนั้นทันที

2.2 OSI Model [4]

OSI model เป็นแบบจำลองมาตรฐานในการสื่อสารที่กำหนดขึ้นมาโดยองค์การมาตรฐานสากล (International Standard Organization - ISO) และมีชื่อเรียกว่า Open System Interconnection Model (OSI model) โดยมีวัตถุประสงค์ เพื่อให้ผู้ผลิตฮาร์ดแวร์หรือซอฟต์แวร์ใดๆ ใช้เป็นโครงสร้างอ้างอิงในการสร้างอุปกรณ์ให้สามารถทำงานร่วมกันได้อย่างดีบนระบบเครือข่าย แบบจำลอง OSI จะแบ่งการทำงานของระบบเครือข่ายออกเป็น 7 ชั้น คือ



รูปที่ 2.1 แบบจำลอง OSI 7 เลเยอร์ (layer)

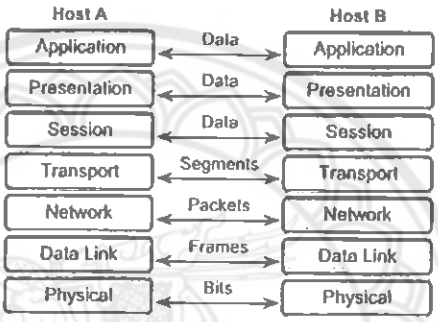
แต่ละชั้นของแบบการสื่อสารข้อมูลเรียกว่าเลเยอร์ (layer) ประกอบด้วยเลเยอร์ย่อยๆ ทั้งหมด 7 เลเยอร์ แต่ละชั้นทำหน้าที่รับส่งข้อมูลกับชั้นที่อยู่ติดกับตัวเองเท่านั้นจะไม่ติดต่อกระโดดข้ามไปยังชั้นอื่นๆ เช่น เลเยอร์ 6 จะติดต่อกับเลเยอร์ 5 และเลเยอร์ 7 เท่านั้นและการส่งข้อมูลจะไล่จากเลเยอร์7 ลงมาจนถึงเลเยอร์ 1 ซึ่งเป็นชั้นที่มีการเชื่อมต่อทางกายภาพ จากนั้นข้อมูลจะถูกส่งไปยังเครื่องผู้รับปลายทางโดยเริ่มจากเลเยอร์ 1 ข้อมูลก็จะถูกถอดรหัส และส่งขึ้นไปตามเลเยอร์ 1 จนถึง เลเยอร์7 ก็จะประกอบกลับมาเป็นข้อมูล ส่งให้แอพพลิเคชัน (application) นำไปใช้งานหรือแสดงผลต่อไป

การส่งผ่านข้อมูลระหว่างชั้น

เมื่อคอมพิวเตอร์ A ต้องการส่งข้อมูลไปยังคอมพิวเตอร์ B จะมีกระบวนการทำงานต่างๆ ตามลำดับดังนี้

ข้อมูลจากเลเยอร์ 7 6 และ 5 จะถูกนำมาหั่นเป็นท่อนๆ แล้วใส่ข้อมูลบางอย่างเพิ่มเข้าไปในส่วนหัว เรียกว่า header เพื่อใช้ในการบันทึกข้อมูลที่จำเป็น เช่น หมายเลขพอร์ต (Port address) ต้นทางและหมายเลขพอร์ตปลายทาง กลายมาเป็นก้อนข้อมูล (Segment) ในเลเยอร์ 4 ซึ่งเรียกว่า TCP Segment จากนั้นข้อมูลเลเยอร์ 4 จะถูกส่งผ่านลงไปยังเลเยอร์ 3 และจะถูกใส่ header อีกซึ่งเป็นการเพิ่ม header เป็นชั้นๆ เรียกว่า การ Encapsulate ซึ่งในส่วนนี้จะเหมือนกับการเอาเอกสารใส่ซอง

จดหมายแล้วจ่าหน้าซองระบุผู้ส่งและผู้รับ คือเป็นการบันทึกหมายเลขไอพี (IP Address) ของโฮส (Host) ต้นทางและโฮสปลายทางไว้ด้วย เมื่อการ encapsulate เสร็จสิ้นจะได้ก้อนข้อมูลที่เรียกว่า Packet จากนั้น Packet ของข้อมูลจะถูกส่งผ่านไปยังระดับล่างอีก คือส่งไปให้เลเยอร์ 2 ในชั้นนี้ข้อมูลจะถูกใส่ header เพิ่มเข้าไปที่ส่วนหัวเพื่อเก็บ MAC Address ของต้นทางและปลายทาง และยังมีการใส่ข้อมูลต่อเพิ่มเข้าไปในส่วนหางด้วย ข้อมูลที่ต่อเพิ่มไปในส่วนหางนี้เรียกว่า Trailer จึงรวมกันกลายเป็นก้อนข้อมูลของเลเยอร์ 2 ที่เรียกว่าเฟรม (Frame) จากนั้น Frame ข้อมูลจะถูกแปลงให้กลายเป็นบิตของข้อมูลเพื่อส่งไปตามสื่อ เช่น สาย UTP, Fiber optic ต่อไป การส่งสัญญาณทางไฟฟ้าไปตามสื่อต่างๆ นี้ เป็นการทำงานในระดับเลเยอร์ 1 เรียกว่า Physical layer



รูปที่ 2.2 การส่งข้อมูลผ่านระหว่างเลเยอร์ (layer)

เลเยอร์ 7 Application layer เป็นเลเยอร์ที่อยู่บนสุดของขบวนการรับส่งข้อมูล ทำหน้าที่ติดต่อกับผู้ใช้เพื่อคอยรับคำสั่งจากผู้ใช้ โดยมีบริการต่างๆ คอยให้บริการ เช่น บริการการส่งไฟล์อาจจะใช้โปรโตคอล FTP บริการเว็บเพ็จใช้โปรโตคอล HTTP บริการส่งอีเมลล์ใช้โปรโตคอล SMTP เป็นต้น

เลเยอร์ 6 Presentation layer เป็นเลเยอร์ที่ทำหน้าที่ตกลงกับคอมพิวเตอร์อีกด้านหนึ่งในชั้นเดียวกันว่าการรับส่งข้อมูลในระดับโปรแกรมประยุกต์จะมีขั้นตอนและข้อบังคับอย่างไร จุดประสงค์หลักของเลเยอร์นี้คือ กำหนดรูปแบบของการสื่อสาร เช่น ASCII Text, Unicode, EBCDIC รวมถึงการเข้ารหัส (Encryption) และบีบอัดข้อมูล (Data compression) เพื่อให้มีขนาดเล็กลงก็รวมอยู่ในเลเยอร์ด้วย

เลเยอร์ 5 Session layer เป็นเลเยอร์ที่ควบคุมการสื่อสารจากต้นทางไปยังปลายทาง และคอยควบคุมช่องทางการสื่อสารในกรณีที่มีหลายๆ โปรเซสต้องการรับส่งข้อมูลพร้อมๆ กันบนเครื่องเดียวกัน (ทำงานคล้ายๆ เป็นหน้าต่างคอยสลับเปิดให้ข้อมูลเข้าออกตามหมายเลขพอร์ตที่กำหนด) ควบคุมจังหวะในการรับส่งข้อมูลของทั้ง 2 ด้านให้มีความสอดคล้องกัน(Synchronization) ตัวอย่างโปรโตคอลในชั้นนี้ คือ RPC, NetBIOS, Windows socket, NFS เป็นต้น

เลเยอร์ 4 Transport layer เป็นเลเยอร์ที่มีหน้าที่หลักในการแบ่งข้อมูลในเลเยอร์บนให้พอเหมาะกับการจัดส่งไปใน layer ล่าง ซึ่งการแบ่งข้อมูลนี้เรียกว่า Segmentation และทำหน้าที่ประกอบรวมข้อมูลต่างๆ ที่ได้รับมาจากเลเยอร์ล่าง (Reassembly) คอยให้บริการตรวจสอบและแก้ไขปัญหาเมื่อเกิดข้อผิดพลาดขึ้นระหว่างการส่ง ทำหน้าที่ยืนยันว่าข้อมูลได้ถูกส่งไปถึงยังเครื่องปลายทางและได้รับข้อมูลถูกต้องเรียบร้อยแล้ว การส่งข้อมูลในเลเยอร์นี้จัดอยู่ในระดับ End-to-End หน่วยของข้อมูลที่ถูกแบ่งแล้วนี้เรียกว่า เซกเมนต์ (Segment) ตัวอย่างของโปรโตคอลในชั้นนี้คือ TCP, UDP เป็นต้น

เลเยอร์ 3 Network layer เป็นเลเยอร์ที่มีหน้าที่หลักในการส่งแพ็คเกต (Packet) จากเครื่องต้นทางให้ไปถึงปลายทางด้วยความพยายามที่ดีที่สุด เลเยอร์นี้จะกำหนดให้มีการตั้ง Logical Address ขึ้นมาเพื่อใช้ระบุตัวตน ตัวอย่างของโปรโตคอล นี้ เช่น IP และ Logical Address ที่ใช้คือหมายเลขไอพี (IP Address) นั่นเอง เลเยอร์นี้ส่วนใหญ่เกี่ยวข้องกับอุปกรณ์ฮาร์ดแวร์ คือ เราเตอร์นั่นเอง โดยจะมีการเลือกเส้นทางที่ดีที่สุดในการส่งข้อมูล การส่งข้อมูลในเลเยอร์นี้จัดอยู่ในระดับ Source-to-Destination หน่วยของเลเยอร์นี้คือ Packet ตัวอย่างของโปรโตคอลในชั้นนี้คือ IP, IPX, Apple talk

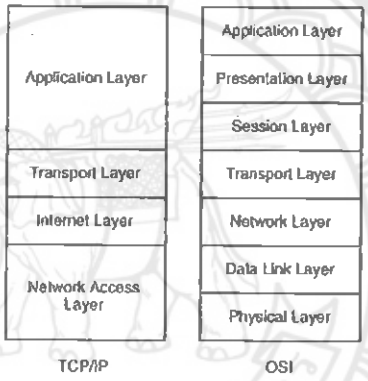
เลเยอร์ 2 Data Link layer รับผิดชอบในการส่งข้อมูลบนเครือข่าย (Network) แต่ละประเภท เช่น Ethernet, Token ring, FDDI หรือบน WAN ต่างๆ โดยหน่วยย่อยข้อมูลในเลเยอร์นี้เรียกว่า เฟรม (Frame) ใน Network แบบ Ethernet layer นี้จะมีการระบุหมายเลข (Address) ของเครื่องต้นทางกับปลายทางด้วย Hardware Address ที่เรียกว่า MAC Address เป็นหมายเลขที่ฝังมากับอุปกรณ์นั้นเลยไม่สามารถเปลี่ยนเองได้ การส่งข้อมูลในเลเยอร์นี้จัดอยู่ในระดับ Node-to-Node ตัวอย่างของโปรโตคอลและเทคโนโลยีในการเชื่อมต่อในเลเยอร์นี้ คือ Ethernet, Token Ring, Frame relay, FDDI, HDLC, ATM เป็นต้น

เลเยอร์ 1 Physical layer เป็นการกล่าวถึงข้อกำหนดมาตรฐานคุณสมบัติทางกายภาพของฮาร์ดแวร์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์ทั้ง 2 ระบบ สัญญาณทางไฟฟ้าและการเชื่อมต่อต่างๆ ของสายเคเบิล ต่างๆ เช่น สายที่ใช้รับส่งข้อมูลเป็นแบบไหน ข้อต่อหรือปลั๊กที่ใช้มีมาตรฐานอย่างไร ใช้ไฟกี่โวลต์ ความเร็วในการรับส่งเป็นเท่าไร สัญญาณที่ใช้รับส่งข้อมูลมีมาตรฐานอย่างไร เลเยอร์ 1 นี้จะมองเห็นข้อมูลเป็นการรับ-ส่งที่ละบิต (Bit) เรียงต่อกันไปโดยไม่มีการพิจารณาเรื่องความหมายของข้อมูลเลย

2.3 Internet Protocol Suite [5]

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้ TCP/IP เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

2.3.1 การแบ่งชั้นของ TCP/IP



รูปที่ 2.3 การแบ่ง TCP/IP ออกเป็น 4 เลเยอร์ (layer) เปรียบเทียบกับ OSI model

ในแต่ละเลเยอร์จะมีหน้าที่ ดังนี้

1. Network access layer ทำหน้าที่ควบคุมการรับส่งข้อมูลในระดับฮาร์ดแวร์ของเครือข่าย รับผิดชอบการรับส่งข้อมูลในระดับกายภาพ จนถึงการแปลความจากสัญญาณไฟฟ้าเป็นข้อมูลทางคอมพิวเตอร์ โดยจะจัดการการเชื่อมต่อแบบ Ethernet, PPP, xDSL, ISDN เป็นต้นโปรโตคอลในเลเยอร์นี้ ได้แก่ ARP, RARP เป็นต้น เทียบได้กับ Physical layer และ Data Link layer ของ OSI model

2. Network layer ทำหน้าที่รับข้อมูลจากชั้น Transport layer ค้นหาและเลือกเส้นทางระหว่างผู้รับและผู้ส่ง โปรโตคอลในเลเยอร์นี้ได้แก่ IP, ICMP, IGMP เป็นต้น เทียบได้กับ Network layer ของ OSI Model

3. Transport layer รับผิดชอบการรับส่งข้อมูลระหว่างปลายด้านส่งและด้านรับข้อมูล และส่งข้อมูลขึ้นไปให้ Application layer นำไปใช้งานต่อ โปรโตคอลในเลเยอร์นี้ ได้แก่ TCP, UDP เทียบได้กับ Session layer และ Transport layer ของ OSI Model

4. Application layer เป็นเลเยอร์ที่แอพพลิเคชั่นเรียกโปรโตคอลระดับล่างๆลงไป เพื่อให้บริการต่างๆ เช่น FTP, SMTP, Telnet, HTTP, POP เป็นต้น เทียบได้กับ Session layer, Presentation layer และ Application layer ของ OSI model

2.3.2 โครงสร้างของโปรโตคอล TCP/IP

เนื่องจาก TCP/IP เป็นชุดของโปรโตคอลประกอบด้วยโปรโตคอลหลายตัวทำงานร่วมกันในเลเยอร์ต่างๆ และมีหน้าที่แตกต่างกันออกไป ได้แก่

TCP (Transmission Control Protocol) อยู่ใน Transport layer ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล และมีกลไกความคุมการ รับส่งข้อมูลให้มีความถูกต้อง (Reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (Connection-Orient)

UDP (User Datagram Protocol) อยู่ใน Transport layer ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล แต่ไม่มีกลไกความคุมการรับ ส่งข้อมูลให้มีเสถียรภาพและเชื่อถือได้ โดยปล่อยให้เป็นหน้าที่ของแอพพลิเคชั่นเลเยอร์ แต่ UDP มีข้อได้เปรียบในการส่งข้อมูลได้ทั้งแบบ unicast, multicast และ broadcast อีกทั้งยังทำการติดต่อสื่อสารได้เร็วกว่า TCP เนื่องจาก TCP ต้องเสีย overhead ให้กับขั้นตอนการสื่อสารที่ทำให้ TCP มีความน่าเชื่อถือในการรับส่งข้อมูลนั่นเอง

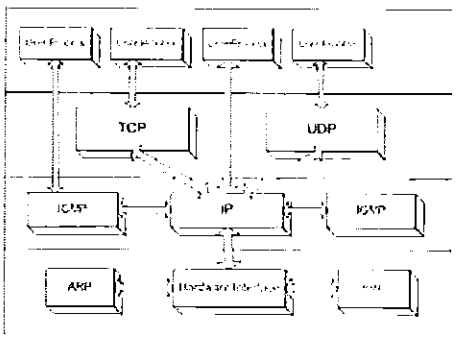
IP (Internet Protocol) อยู่ใน Internetwork layer เป็นโปรโตคอลหลักในการสื่อสารข้อมูล มีหน้าที่ค้นหาเส้นทางระว่างผู้รับและผู้ส่ง โดยใช้หมายเลขไอพี (IP Address) ซึ่งมีลักษณะเป็นเลขสี่ชุด แต่ละชุดมีค่าตั้งแต่ 0 - 255 เช่น 172.17.3.12 ในการอ้างอิงโฮสต่างๆ และกลไกการค้นหาเส้นทาง (Route) เพื่อส่งต่อข้อมูลไปจนถึงจุดหมายปลายทาง

ICMP (Internet Control Message Protocol) อยู่ใน Internetwork layer มีหน้าที่ส่งข่าวสารและแจ้งข้อผิดพลาดให้แก่ IP

IGMP (Internet Group Management Protocol) อยู่ใน Network layer ทำหน้าที่ในการส่ง UDP ดาต้าแกรมไปยัง กลุ่มของโฮส หรือ โฮสหลายๆตัวพร้อมกัน

ARP (Address Resolution Protocol) อยู่ใน Link layer ทำหน้าที่เปลี่ยนระหว่างหมายเลขไอพี (IP Address)ให้เป็นแอดเดรสของ Network Interface เรียกว่า MAC Address ในการติดต่อระหว่างกัน MAC Address คือหมายเลขประจำของ Hardware Interface ซึ่งในโลกนี้จะไม่มี MAC Address ที่ซ้ำกัน มีลักษณะเป็นเลขฐาน 16 ยาว 6 ไบต์ เช่น 23:43:45:AF:3D:78 โดย 3 ไบต์แรกจะเป็นรหัสของผู้ผลิต และ 3 ไบต์หลังจะเป็นรหัสของผลิตภัณฑ์

RARP (Reverse ARP) อยู่ใน Link layer เช่นกัน แต่ทำหน้าที่กลับกันกับ ARP คือ เปลี่ยนระหว่างแอดเดรสของ Network Interface ให้เป็น IP Address



รูปที่ 2.4 ความสัมพันธ์ระหว่างโปรโตคอลต่างๆใน TCP/IP

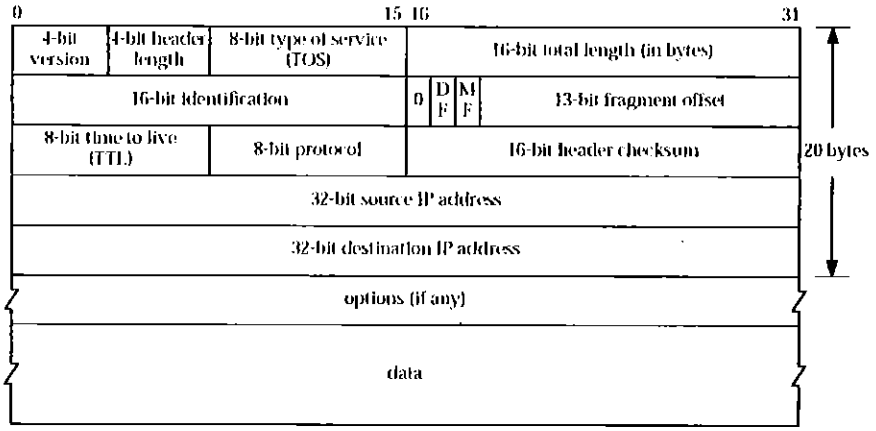
2.3.3 IP Protocol

IP เป็นโปรโตคอลที่ทำหน้าที่รับภาระในการนำข้อมูลไปส่งยังผู้รับ ที่เชื่อมต่ออยู่ในระบบเน็ตเวิร์ค ซึ่งทั้งสองฝั่งอาจอยู่คนละเน็ตเวิร์คกันก็ได้ โปรโตคอลอื่นๆ ในระดับ Network layer ขึ้นไปทั้ง TCP, UDP, ICMP ต่างก็ต้องอาศัยโปรโตคอล IP ในการรับส่งข้อมูลทั้งสิ้น

โปรโตคอล IP มีความสามารถในการค้นหาเส้นทางจากผู้รับไปยังผู้ส่ง มีกลไกที่ชาญฉลาดในการค้นหาเส้นทาง สามารถค้นหาเส้นทางได้ไปถึงผู้รับได้เอง หากมีเส้นทางที่สามารถไปได้ แต่ไม่ได้ติดต่อระหว่างผู้รับกับผู้ส่งโดยตรง และไม่มีการยืนยันว่าข้อมูลถึงผู้รับจริงหรือไม่ ทั้งนี้อาจเกิดจากหลายสาเหตุ เช่น ที่อยู่ของผู้รับไม่มีการเชื่อมต่ออยู่ในระบบอินเทอร์เน็ต กล่าวได้ว่าโปรโตคอล IP มีหน้าที่ในการค้นหาเส้นทางเท่านั้น ไม่มีการยืนยันผลสำเร็จในการส่งข้อมูล หากเกิดข้อผิดพลาดในการส่งข้อมูล แม้ว่าจะมีการส่ง ICMP Message กลับมารายงานข้อผิดพลาด แต่ก็รับประกันไม่ได้อยู่ดีว่า ICMP Message จะกลับมาถึงเรียบร้อยหรือไม่ ด้วยเหตุนี้ จึงถือว่า IP เป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (Reliable)

IP Header

เมื่อข้อมูลถูกส่งลงมาจากชั้น Transport layer สู่ชั้น Network layer กระบวนการ encapsulation ของ IP protocol จะทำการเพิ่มส่วน header ลงไป header ของ IP datagram มีขนาด 20-32 ไบต์ มีส่วนประกอบต่างๆ ดังแสดงในรูป



รูปที่ 2.5 IP Header

บิตที่ 0-3	Version	มีขนาด 4 บิตเป็นเวอร์ชั่น (version) ของ IP ปัจจุบันค่านี้ถูกกำหนดให้เป็น 4
บิตที่ 4-7	Length	มีขนาด 4 บิตเป็นค่าความยาวของ header นี้ โดยปกติจะเป็น 5 หมายความว่า 5*32 บิต = 20 ไบต์
บิตที่ 8-15	Type of service	เป็นข้อมูลขนาด 8 บิต ปัจจุบันไม่ได้ใช้งานแล้ว
บิตที่ 16-31	Total length	เป็นฟิลด์ที่บอกจำนวนไบต์ทั้งหมดของ IP datagram ด้วยขนาด 16 บิตทำให้ datagram มีขนาดสูงสุดไม่เกิน 65535 ไบต์ และมีขนาดเล็กสุดไม่ต่ำกว่า 512 ไบต์
บิตที่ 32-47	Identification	ใช้ในกรณีที่มีการแบ่งดาต้าแกรม (Datagram) ออกเป็นแฟรกเมนต์ (fragment) เมื่อนำกลับมารวมกันใหม่จะได้รู้ว่ามาจากดาต้าแกรมเดียวกัน
บิตที่ 48-50	Flag	ใช้ในกรณีที่มีการแบ่งข้อมูลออกเป็นแฟรกเมนต์ มีความหมายดังนี้ บิต 0: reserved เป็น 0 เสมอ บิต 1 (DF) 0 = May Fragment, 1 = Don't Fragment บิต 2 (MF) 0 = Last Fragment, 1 = More Fragments.
บิตที่ 51-63	fragment offset	เป็นส่วนระบุข้อมูลที่ใช้แยกรวมข้อมูล เพื่อให้ข้มูลที่ถูกแยกออกเป็นแฟรกเมนต์กลับมารวมกันได้อย่างถูกต้องตามลำดับ
บิตที่ 64-71	Time to Live (TTL)	เป็นจำนวนครั้งสูงสุดที่ดาต้าแกรมนี้จะถูกส่งผ่านครือข่ายไปยังปลายทางได้ เพื่อ ป้องกันไม่ให้ดาต้าแกรมถูกเราต์ (Route)ไปเรื่อยๆอย่างไม่สิ้นสุด ปกติค่านี้จะเริ่มต้นที่ 32 และจะถูกลดค่า

ลงทีละ 1 เมื่อมีการเราต์ จนค่านี้มีค่าเป็น 0 ก็จะไม่ถูกเราต์อีกต่อไป

บิตที่ 72-79 Protocol เป็นข้อมูลที่ระบุโปรโตคอลที่ส่งดาต้าแกรมนี้มา ตัวอย่างโปรโตคอลที่ใช้บ่อยๆ ได้แก่

โปรโตคอล	ค่าในฟิลด์ Protocol	อธิบาย
ICMP	1	Internet Control Message Protocol
TCP	6	Transmission Control Protocol
UDP	17	User Datagram Protocol

บิตที่ 80-95 Header checksum เป็นส่วนตรวจสอบความถูกต้องของข้อมลใน header โดยไม่เกี่ยวกับส่วนข้อมูลที่อยู่ภายใน payload ค่านี้จะถูกคำนวณใหม่ทุกครั้งที่มีการเปลี่ยนแปลงข้อมูลใน Header (เช่น TTL ที่มีการเปลี่ยนแปลงทุกครั้งที่ IP datagram ถูกส่งผ่านเราเตอร์)

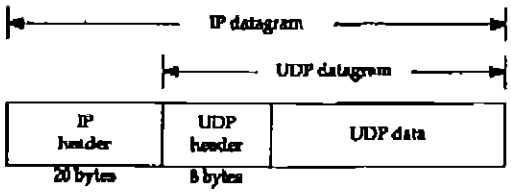
บิตที่ 86-127 Source IP Address คือ IP Address ของผู้ส่งดาต้าแกรม

บิตที่ 128-163 Destination IP Address คือ IP Address ของผู้รับดาต้าแกรมไม่แน่นอน

Option มีขนาดข้อมูลไม่แน่นอน ใช้สำหรับกำหนดค่าพารามิเตอร์ปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการนำไปใช้งาน ขึ้นอยู่กับ Option Padding มีข้อมูลว่างเปล่า ใช้เป็นส่วนเติมเต็มของฟิลด์ Optionให้ครบ 32 ไบต์

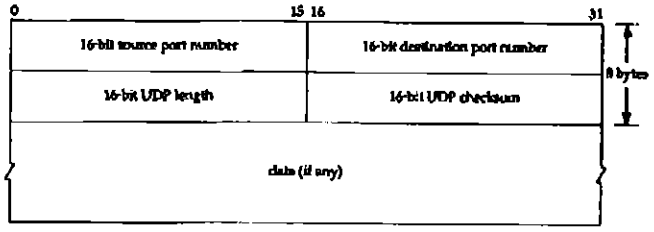
2.3.4 UDP Protocol

UDP เป็นโปรโตคอลที่ถูกออกแบบมาให้ทำหน้าที่รับส่งข้อมูลโดยมีขั้นตอนการทำงานไม่ซับซ้อนและทำงานได้รวดเร็ว แต่มีจุดด้อยคือไม่มีความน่าเชื่อถือ (Unreliable) และเป็นการสื่อสารแบบไม่ต่อเนื่อง (Connectionless) โปรโตคอล UDP ทำงานในชั้น Transport layer ซึ่งจะต้องพึ่งพาโปรโตคอล IP ในการรับส่งข้อมูล



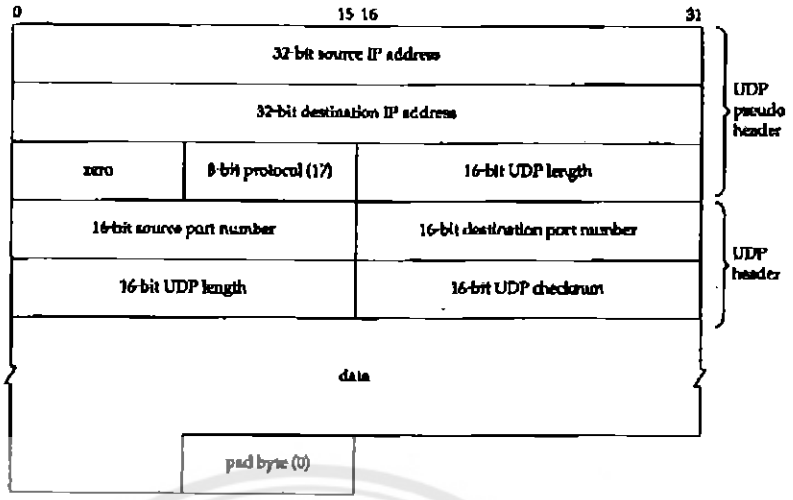
รูปที่ 2.6 IP Datagram ของ UDP

UDP Header



รูปที่ 2.7 UDP Header

บิตที่ 0-15	Source port number	หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรม (Datagram) นี้ มีความยาว 16 บิต
บิตที่ 16-31	Destination port number	หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับ ดาต้าแกรม มีความยาว 16 บิตเช่นกัน
บิตที่ 32-47	UDP length	ความยาวของดาต้าแกรม ทั้งส่วน Header และ data หมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
บิตที่ 48-63	Checksum	เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย Checksum เป็นเลข 16 บิตถูกคำนวณด้วยวิธี one's complement โดยนำ Pseudo Header และข้อมูลทั้งหมดใน UDP Datagram มาคำนวณ Pseudo Header เป็นข้อมูลที่อยู่ในส่วนของ IP Header ประกอบด้วยฟิลด์ Source IP Address, Destination IP Address, Zero, Protocol, UDP Length ดังแสดงในรูป



รูปที่ 2.8 Pseudo Header และ UDP Header

หากค่า Checksum ที่คำนวณออกมาเป็น 0 ค่า checksum จะถูกกำหนดเป็น 1 ทั้งหมดแทน (มีค่าเท่ากันในระบบ 1's complement) ทั้งนี้เพราะในบางแอพพลิเคชั่นที่ไม่ต้องการตรวจสอบค่า checksum ในระดับ UDP จะกำหนดค่านี้เป็น 0 (disable checksum)

2.3.5 TCP Protocol

TCP เป็นโปรโตคอลที่ใช้สื่อสารระหว่างโฮสที่มีความน่าเชื่อถือ จะเห็นได้ว่าโปรโตคอลในระดับ IP หรือแม้กระทั่ง UDP จะสนใจ ข้อมูลเพียง 1 ดาต้าแกรม (Datagram) กลไกของโปรโตคอลจะมีหน้าที่ตรวจสอบความถูกต้องเพียงเฉพาะดาต้าแกรมนั้นๆ เมื่อจะทำ การส่งดาต้าแกรมใหม่ก็จะถือว่าเป็นข้อมูลชุดใหม่ที่ไม่มีความสัมพันธ์ใดๆ กับข้อมูลดาต้าแกรมอื่น (การสื่อสาร 1 ครั้ง จึงใช้เพียง 1 ดาต้าแกรม) แต่สำหรับ TCP แล้วจะเห็นว่าข้อมูลนั้นเป็น stream คือมีความสัมพันธ์ต่อเนื่องกัน มีกลไกในการตรวจสอบทั้งด้านส่ง และด้านรับเพื่อให้แน่ใจว่าสามารถสื่อสารกันได้จริงจึงจะมีการส่งรับข้อมูลเกิดขึ้น ตลอดจนการยกเลิกการติดต่อก็มีกลไกสำหรับแจ้งให้อีกฝั่งทราบ ทำให้การสื่อสารด้วย TCP จึงเสมือนว่าทั้ง 2 ฝ่ายคือฝ่ายรับและฝ่ายส่งได้ทำการต่อสายเน็ตเวิร์กถึงกัน (Connected) ตลอดเวลาที่การรับส่งข้อมูลจนกระทั่งการสื่อสารทั้งหมดเสร็จสิ้นจึงจะทำการยกเลิกการเชื่อมต่อนั้นเสีย

จุดเด่นประการสำคัญของ TCP ที่กล่าวถึงอยู่เสมอ คือ ความมีเสถียรภาพและความถูกต้องของการสื่อสารซึ่งมีความเชื่อถือได้สูง คุณสมบัติที่ทำให้ TCP มีข้อดีดัง กล่าวคือ

1. ข้อมูลที่จะส่งผ่าน TCP จะถูกนำมาแตกย่อยออกเป็นส่วนๆ ให้มีขนาดเหมาะสมสำหรับการส่งข้อมูล โดย TCP มีกลไกในการพิจารณาว่าขนาดเท่าใดจะทำให้การรับ-ส่งนั้นมีประสิทธิภาพและน่าเชื่อถือสูงสุด โดยข้อมูลแต่ละชุดที่แบ่งออกและ ทำการส่งโดย TCP แต่ละครั้งจะเรียกว่า TCP segment

2. ในการส่งข้อมูลแต่ละครั้ง TCP จะมีการจับเวลาไว้เสมอ เพื่อรอการตอบรับจากผู้รับว่าได้รับข้อมูลถูกต้อง หากหมดเวลาแล้วไม่มีการตอบรับ TCP จะถือว่าข้อมูลไปไม่ถึงและทำการแก้ปัญหาที่เกิดขึ้น เช่น ยกเลิกการติดต่อ ส่งข้อมูลซ้ำ ทำให้ application ทราบสถานะการส่งข้อมูลตลอดเวลา

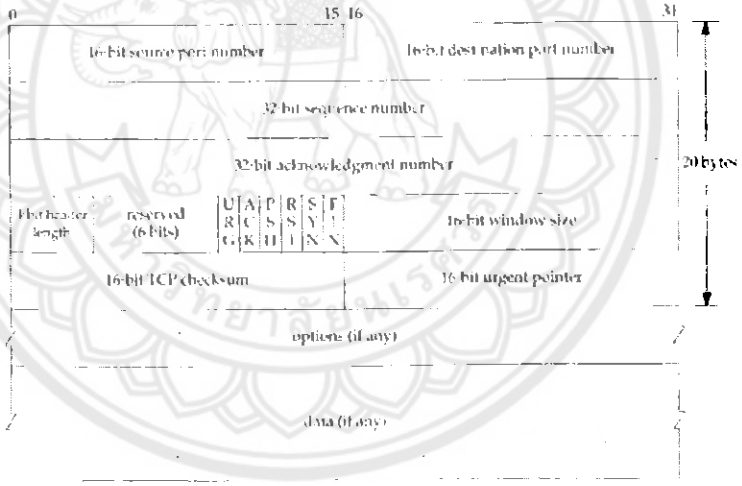
3. TCP มี Checksum ซึ่งจะครอบคลุมทั้ง TCP header และ TCP data เพื่อเป็นการป้องกันและตรวจสอบว่าข้อมูล ที่ส่งมานั้นถูกต้อง และไม่ได้ถูกแก้ไขระหว่างทาง หาก TCP ได้รับข้อมูลที่ทำการตรวจสอบกับ checksum แล้วปรากฏว่า มีความผิดพลาดเกิดขึ้น TCP จะทิ้งข้อมูลที่ได้รับและจะไม่ทำการตอบรับข้อมูลนั้นกลับไปยังผู้ส่ง คือ ถือเสมือนว่าไม่ได้รับข้อมูลนั้น เพื่อให้ทางฝ่ายผู้ส่งทำการส่งใหม่หรือหาข้อบกพร่องและพยายามแก้ไขตามแต่แอพพลิเคชั่นทางฝ่ายผู้ส่งเห็น สมควร

4. เนื่องจาก TCP อาศัย IP ในการส่งข้อมูล ซึ่ง IP เองอาจจะถูกแฟรกเมนต์ (Fragment) ได้ และทำให้ข้อมูลที่ถูกแฟรกเมนต์นั้นส่งถึงปลายทางในลำดับที่ไม่ถูกต้องได้ หน้าที่ของ TCP เมื่อรับข้อมูลที่แฟรกเมนต์มานั้นจะต้องนำข้อมูลแต่ละส่วนมาประกอบ รวมกันให้ถูกต้องสมบูรณ์ก่อนจะส่งไปยัง Application layer ต่อไป

5. การส่ง - รับข้อมูลด้วย IP อาจจะมีกรณีที่ IP Datagram นั้นถูกส่งซ้ำขึ้นได้ TCP ที่รับข้อมูลซ้ำดังกล่าวจะต้องทราบว่า เป็น IP Datagram ที่ซ้ำและไม่นำข้อมูลไปใช้งาน

6. TCP มีกลไกควบคุมการไหลของข้อมูล (Flow control) โดยการควบคุมนี้จะต้องอาศัยลำดับของการรับส่งที่ถูกต้อง และสัมพันธ์กันทั้ง 2 ฝั่ง ในขณะเดียวกันข้อมูลที่ส่งนั้นจะต้องอาศัย IP หลายดาต้าแกรมจึงจะได้รับข้อมูลครบทั้งหมด ดังนั้นในการรับข้อมูลทางฝ่ายรับจึงต้องเตรียมบัฟเฟอร์ไว้จำนวนหนึ่งเพื่อรอรับข้อมูลและรวบรวมข้อมูลทั้งชุดให้อยู่ใน บัฟเฟอร์ก่อนที่จะทำการจัดเรียงข้อมูล ตรวจสอบความถูกต้องแล้วจึงส่งต่อไปยังแอพพลิเคชั่น ด้วยเหตุผลดังกล่าวจะเห็น ได้ว่าขนาดของข้อมูลมิได้ถูกจำกัดที่ขนาดของดาต้าแกรมใดๆ ข้อมูลที่ส่งอาจจะมีขนาดใหญ่มากอยู่ในหลายดาต้าแกรม ก็เป็นได้ ดังนั้นเพื่อป้องกันการส่งข้อมูลขนาดใหญ่เร็วเกินไปจนทำให้ทางฝ่ายรับไม่มีหน่วยความจำเพียงพอที่จะเป็น บัฟเฟอร์ที่พักข้อมูล การส่งข้อมูลจึงถูกจำกัดโดยจะอนุญาตให้ทำการส่งข้อมูลได้ เท่าที่ฝ่ายรับมีบัฟเฟอร์เพียงพอเท่านั้น

TCP Header



รูปที่ 2.9 TCP Header

Source port number หมายถึง พอร์ตที่โฮสต้นทางใช้ในการสื่อสารกันของเซสชั่น (Session) นี้ และ TCP/IPจะใช้พอร์ตนั้นไป ตลอดตราบใดที่การสื่อสารในเซสชั่นนี้ยังไม่ยุติลง โดยทั่วไปพอร์ตนี้จะเรียกว่า "ไคลเอนต์พอร์ต" คือพอร์ตที่ไคลเอนต์เปิดขึ้น มาเพื่อรอการตอบรับจากเซิร์ฟเวอร์ (พิจารณาจากทิศทางของแพ็กเก็ต (Packet) ที่ส่งมาจากไคลเอนต์ไปยังเซิร์ฟเวอร์) ไคลเอนต์พอร์ต จะมีหมายเลขไม่แน่นอนและเปลี่ยนไปทุกครั้งที่มีการเริ่มการเชื่อมต่อใหม่ เป็นพอร์ตที่ถูกเปิดไว้ในระยะเวลาสั้นๆ (Ephemeral Port) ค่าที่เป็นไปได้ของพอร์ตนี้ขึ้นอยู่กับการ

จัดสรรของระบบปฏิบัติการ ในการกำหนดขอบเขตของพอร์ตเหล่านี้ส่วนใหญ่จะมีค่า อยู่ในช่วง 1024 - 5000

Destination port number หมายถึง หมายเลขพอร์ตบนโฮสปลายทางที่โฮสต้นทางต้องการติดต่อด้วย โดยนัยแล้วจะหมายถึงแอพพลิเคชั่นที่ให้บริการอยู่พอร์ตนั้นที่โฮสปลายทางนั่นเอง พอร์ตนั้นจะเรียกอีกอย่างหนึ่งว่า "เซิร์ฟเวอร์พอร์ต" หมายเลขพอร์ตที่เปิดไว้จะขึ้นอยู่กับแอพพลิเคชั่นที่ให้บริการ โดยทั่วไปแอพพลิเคชั่นแต่ละประเภทจะมีหมายเลขพอร์ตเป็นมาตรฐานสำหรับให้ไคลเอนต์ได้เรียกใช้บริการ

Sequence number เป็นฟิลด์ (field) ที่ระบุถึงหมายเลขลำดับที่ใช้อ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อให้ทั้ง 2 ฝ่ายจะได้รับทราบตรงกันว่าเป็นข้อมูลของชุดใด การนำไปใช้งานจะได้ไม่ปะปนกัน และมีลำดับที่ถูกต้อง เนื่องจากการสื่อสารข้อมูลผ่าน TCP นั้นจังหวะและลำดับเป็นส่วนสำคัญของโปรโตคอลไม่ยิ่งหย่อนไปกว่าข้อมูลใน TCP header รวมไปถึง การที่ข้อมูลในแต่ละ TCP segment อาจจะถูกทำการแฟรกเมนต์ (Fragment) ในเลเยอร์ของ IP ถัดลงไป ทำให้ข้อมูลถูกแบ่งออกและส่งไปในลำดับที่ไม่เรียงกัน หากไม่มีจุดอ้างอิงของข้อมูลก็จะไม่สามารถอ่านข้อมูลกลับใหม่ได้อย่างสมบูรณ์และถูกต้อง การส่งข้อมูลและการตอบรับจะใช้ฟิลด์ (field) นี้เป็นตัวยืนยันระหว่างกันเสมอ

Acknowledge number ทำหน้าที่เช่นเดียวกับ Sequence number ต่างกันตรงที่เป็น Sequence number ซึ่งในการตอบรับ กล่าวคือ เนื่องจาก Sequence number ที่ใช้ในการอ้างอิงนั้นผู้ที่เริ่มส่งข้อมูลจะเป็นผู้กำหนดเลขขึ้น มาและส่งไปพร้อมกับการสร้างการเชื่อมต่อครั้งใหม่แต่สำหรับฝ่ายที่ถูกติดต่อก็จำเป็นต้องกำหนดหมายเลขสำหรับใช้อ้างอิง ในการตอบรับเช่นกัน ค่าที่อยู่ใน Acknowledge number ก็คือหมายเลขที่ใช้อ้างอิงในการตอบรับนี้

Header length โดยปกติความยาวของ TCP header จะเท่ากับ 20 ไบต์ แต่ถ้าหากมีการใช้ Option อาจจะทำให้ ขนาดของเฮดเดอร์ (header) ยาวขึ้นตามข้อมูลที่ต้องเพิ่มมาจาก Option นั้น แต่ทั้งหมดแล้วจะไม่เกิน 60 ไบต์

Flag เป็นข้อมูลในระดับบิตที่ใช้เป็นตัวบอกคุณสมบัติของ TCP segment ที่กำลังส่งอยู่นั้น และใช้เป็นตัวควบคุมจังหวะ การรับส่งข้อมูลด้วย ซึ่ง Flag ทั้งหมดมีอยู่ 6 บิต แต่ละบิตมีชื่อและมีความหมายดังนี้

URG ใช้บอก ความหมายว่า เป็นข้อมูลด่วนและมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent pointer)

ACK แสดงว่าข้อมูลในฟิลด์ Acknowledge number นำมาใช้งานได้

DSH เพื่อแจ้งให้ผู้รับข้อมูลทราบว่า ควรจะส่งข้อมูล segment นี้ไปยังโปรเซสที่กำลังรออยู่ทันที

RST ใช้ในกรณีที่เกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น โฮสมีปัญหา ให้เริ่มต้นสื่อสารกันใหม่

SYN ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง

FIN ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

Window size เป็นขนาดของการรับ-ส่งข้อมูลในแต่ละครั้งที่ทางฝ่ายผู้รับจะสามารถรับได้ เนื่องจากในการรับข้อมูลนั้น ทางผู้รับจะต้องจัดเตรียมหน่วยความจำในการพักข้อมูลที่มากจาก TCP และทำการดีมัลติเพลกซ์ (Demultiplex) ออกมา หากไม่มีการตกลง ถึงขนาดที่ทางฝ่ายรับสามารถรับได้ ก็จะทำให้การสื่อสารข้อมูลไม่สมดุล และฝ่ายรับอาจจะประมวลผลทัน ซึ่งจะส่งผลให้ต้องส่ง ข้อมูลซ้ำหลายครั้ง

Checksum เป็นฟิลด์ที่ใช้ในการตรวจสอบความถูกต้องของข้อมูลใน TCP เซกเมนต์ใช้ระบุหมายเลข Sequence number ของ TCP เซกเมนต์ล่าสุดที่อยู่ในโหมด Urgent

Urgent pointerเป็นข้อมูลเพิ่มเติมซึ่งจะอยู่ใน TCP header เมื่อมีการตั้งค่า option บางอย่างที่ต้องการข้อมูลเพิ่มเติมซึ่งไม่มีใน TCP header เช่น MSS, Strict Route

2.4 เครื่องมือ และภาษาที่ใช้ในการพัฒนา

2.4.1 ดอตเน็ตเฟรมเวิร์ก (.NET Framework) [6]

คือแพลตฟอร์มสำหรับพัฒนาซอฟต์แวร์สร้างขึ้นโดยไมโครซอฟท์ โดยรองรับภาษาดอตเน็ตที่หลากหลายภาษา ซึ่งมีไลบรารีเป็นจำนวนมากสำหรับการเขียนโปรแกรม รวมถึงบริหารการดำเนินการของโปรแกรมบนดอตเน็ตเฟรมเวิร์ก โดยไลบรารีนั้นได้รวมถึงส่วนต่อประสานกับผู้ใช้ การเชื่อมต่อฐานข้อมูล วิทยาการเข้ารหัสลับ อัลกอริทึม การเชื่อมต่อเครือข่ายคอมพิวเตอร์ และการพัฒนาเว็บแอพพลิเคชัน ปัจจุบันมีการพัฒนาออกมาเป็นเวอร์ชันที่ 4 แล้ว โดย .NET Framework นั้นมีส่วนประกอบภายในแบ่งออกเป็น 3 ชั้นใหญ่ๆ คือ

1. Programming Language เป็นรูปแบบของภาษาที่ออกแบบมาเพื่อให้สามารถทำงานในสภาวะที่เป็น .NET ได้โดยที่ทาง Microsoft ได้เปิดตัวภาษาหลักๆ ที่จะใช้ในการพัฒนาบน .NET นี้ 3 ภาษา คือวิชวลเบสิกดอทเน็ต (VB.NET) วิชวลซีชาร์ป (VC#) และวิชวลซีพลัสพลัส (VC++)

2. Base Classes Library โดย Library นั้นเปรียบเสมือนชุดคำสั่งสำเร็จรูปย่อยๆที่เพิ่มเข้ามา ซึ่งส่วนใหญ่จะเป็นชุดคำสั่งที่ต้องใช้งานอยู่เป็นประจำ ดังนั้นจึงมีผู้คิดค้นเครื่องอำนวยความสะดวกในการเขียน ซึ่งภายในระบบ .NET จะสร้างสิ่งที่เรียกว่าเป็น Library พื้นฐานขึ้น ทำให้ไม่ว่าจะใช้ภาษาใดในการพัฒนาโปรแกรม ก็สามารถที่จะเรียกใช้ Library ที่เป็นตัวเดียวกันได้หมด

3. Common Language Runtime (CLR) นับเป็นสิ่งสำคัญแทบจะที่สุดของระบบ .NET เลยก็ว่าได้ เพราะ CLR ที่ว่านี้มีหน้าที่ในการรันโปรแกรมที่เราคอมไพล์ขึ้นมา ซึ่งโค๊ดที่คอมไพล์นั้นจะกลายเป็นภาษาที่เรียกว่า Intermediate language (IL) ซึ่งเป็นโค๊ดกลางไม่ขึ้นอยู่กับ Platform ของ OS หรือ Hardware ใดๆ และจะถูกรันขึ้นมาโดยการจัดการและควบคุมโดย CLR นั่นเอง

ข้อดีของ .NET Framework นั้นพอจะสรุปออกมาได้เป็นข้อๆ ดังนี้

1. เป็นระบบที่มี Library ที่เป็นมาตรฐานเดียวกัน เนื่องจากมี Library ที่เป็นมาตรฐานเดียวกันทั้งหมดทำให้เราไม่ต้องกังวลว่าภาษาที่ใช้เขียนนั้นมี Library ตัวนั้นตัวนี้หรือไม่ รวมทั้งไม่ต้องคอยกังวลว่าถ้าใช้ Library ของภาษาหนึ่งแล้วอีกภาษาหนึ่งจะไม่มี Library ตัวนั้น

2. ไม่ขึ้นกับระบบปฏิบัติการ (OS) เนื่องจากระบบปฏิบัติการที่แต่ละบุคคล หรือองค์กรใช้นั้นย่อมไม่เหมือนกัน แต่ภายใน .NET Framework จะไม่มีปัญหานี้ ขอเพียงแค่มีระบบ .NET Framework ก็จะทำให้สามารถใช้งานโปรแกรมต่างๆได้ ซึ่งเป็นข้อดีตรงที่เราจะสามารถใช้โปรแกรมต่างๆได้ทุกระบบปฏิบัติการ

3. ใช้ในการพัฒนาได้ทุกภาษา ทำให้เราไม่ต้องคอยมาศึกษาภาษาใหม่ๆ เมื่อต้องการสร้างโปรแกรมในแต่ละครั้ง นอกจากนั้นเรายังสามารถเลือกใช้ภาษาที่เราถนัดที่สุดในการพัฒนาโปรแกรมต่างๆ ได้ด้วย

4. มีการควบคุมสิ่งแวดล้อมในการทำงานเป็นอย่างดี เนื่องจากเป็นระบบที่เป็นมาตรฐาน ทำให้การควบคุม จัดสรรระบบต่างๆ ทำได้ง่ายขึ้น ไม่ว่าจะเป็นการจัดสรรหน่วยความจำด้านการใช้งานเครื่องก็มีความรวดเร็วมากขึ้น ลดโอกาสที่เครื่องจะล่มได้เป็นอย่างดี

ดังนั้นโปรแกรมที่เขียนบนดอตเน็ตเฟรมเวิร์ก จะทำงานบนสภาพแวดล้อมที่บริหารโดย Common Language Runtime (CLR) ซึ่งเป็นส่วนหนึ่งในดอตเน็ตเฟรมเวิร์ก โดย CLR นั้นเตรียมสภาพแวดล้อมเสมือน ทำให้ผู้พัฒนาไม่ต้องคำนึงถึงความสามารถที่แตกต่างระหว่างหน่วยประมวลผลต่างๆ และ CLR ยังให้บริการด้านกลไกระบบความปลอดภัย การบริหารหน่วยความจำ และ Exception handling ดอตเน็ตเฟรมเวิร์กนั้นออกแบบมาเพื่อให้การพัฒนาซอฟต์แวร์ง่ายขึ้น รวดเร็วขึ้น และปลอดภัยขึ้นกว่าเดิม

2.4.2 ภาษาซีชาร์ป (C# Programming Language)

เป็นภาษาโปรแกรมเชิงวัตถุทำงานบนดอตเน็ตเฟรมเวิร์ค พัฒนาโดยบริษัทไมโครซอฟท์และมี Anders Hejlsberg เป็นหัวหน้าโครงการ โดยมีรากฐานมาจากภาษาซีพลัสพลัสและภาษาอื่นๆ (โดยเฉพาะภาษาเดลไฟและจาวา) มาปรับปรุงเพื่อให้มีความเป็น OOP อย่างสมบูรณ์ที่สุด ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) โดยจะคอมไพล์โค้ดเป็น Intermediate language (IL) หรือโค้ดกลาง ที่จะต้องไปคอมไพล์อีกครั้งด้วย Common Language Runtime (CLR) ใน .Net Framework บนเครื่องที่จะไปทำงานอีกครั้งหนึ่ง

ลักษณะคำสั่ง ชื่อของชนิดตัวแปรต่างๆ จะคล้ายกับภาษา C++ แต่รูปแบบไวยากรณ์จะคล้ายกับภาษา Java มาก รูปแบบของภาษา C# คือ OOP จะมีการมองทุกอย่างเป็นวัตถุ และสืบทอดต่อกันมาด้วยการสร้างคลาส คลาสทุกคลาสที่ไม่ถูกสืบทอดจากคลาสอื่นจะถือว่าคลาสนั้นสืบทอดมาจากคลาสชื่อ Object ซึ่งก็คือคลาสทุกคลาสที่ถูกสร้างมาจะถือว่าเป็นวัตถุอย่างหนึ่ง และเมื่อนำคลาสนั้นไปให้คลาสอื่นสืบทอด คลาสที่ได้รับการสืบทอดก็จะสืบทอดความเป็น Object มาด้วย โดยปัจจุบันภาษาซีชาร์ปพัฒนาจนถึงเวอร์ชัน 4.0 และเป็นภาษามาตรฐานที่รองรับโดย ECMA และ ISO

2.4.3 ไมโครซอฟท์ วิชวลสตูดิโอ (Microsoft Visual Studio)

คือ IDE (Integrated Development Environment) พัฒนาขึ้นโดยไมโครซอฟท์ ซึ่งเป็นเครื่องมือที่ช่วยนักพัฒนาซอฟต์แวร์พัฒนาโปรแกรมคอมพิวเตอร์ เว็บไซต์ เว็บแอพพลิเคชัน และเว็บเซอร์วิส ระบบที่รองรับการทำงานนั้นมีไมโครซอฟท์วินโดวส์ พ็อคเกตพีซี (Pocket PC) สมาร์ทโฟน (Smart Phone) และเว็บเบราว์เซอร์ (Web browser) ในปัจจุบันวิชวลสตูดิโอนั้นสามารถใช้ภาษาโปรแกรมที่เป็นภาษาดอตเน็ตในโปรแกรมเดียวกัน เช่น VB.NET, C++, C#, J# เป็นต้น สำหรับวิชวลสตูดิโอ 2010 ซึ่งเป็นรุ่นล่าสุดได้แบ่งเป็นรุ่น ดังต่อไปนี้

Visual Studio Express Edition

Visual Studio Premium Edition

Visual Studio Professional Edition

Visual Studio Test Professional Edition

Visual Studio Team System

โดยในรุ่น Express Edition นั้นสามารถดาวน์โหลดมาใช้งานได้ฟรี ไม่เสียค่าใช้จ่าย แต่จะถูกจำกัดบางคุณลักษณะไม่ให้ใช้งานได้

2.4.4 SQLite [7]

SQLite เป็นฐานข้อมูลฉบับกระเป๋าทำนองเดียวกับ MS Access สิ่งที่แตกต่างกันคือ ฟรี ติดตั้งง่าย ไม่จำกัดระบบปฏิบัติการทั้งวินโดวส์ (Windows) แมค (MAC OS) และลินุกซ์ (Linux) สำหรับฐานข้อมูลของ SQLite เป็นลักษณะไฟล์ข้อมูลธรรมดา กล่าวคือ เก็บข้อมูลไว้ในไฟล์เพียงไฟล์เดียวเช่นเดียวกับ *.mdb หรือ *.ACCDB ของ MS Access และ *.mdf ของ SQL Server ดังนั้นเพื่อไม่ให้สับสนก็ควรตั้งชื่อนามสกุลของไฟล์ที่ไม่ไปชนกับฐานข้อมูลตระกูลอื่น ยกตัวอย่างเช่น .db, .dat, .sdb, .s3db เป็นต้น SQLite เหมาะกับแอพพลิเคชันแบบ Standalone แต่สามารถนำไปประยุกต์ใช้งานได้หลากหลาย เช่น ดิกชินนารี รายการสินค้า โปรแกรมแบบสอบถาม การเก็บข้อมูลที่ต้องการส่งเป็นไฟล์ข้อมูลทางเมล์หรือมือถือ เป็นต้น นอกจากนี้ยังเปิดเผย Source code สำหรับการนำพัฒนาต่ออีกด้วย

บทที่ 3

วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงวิธีการดำเนินงานเพื่อพัฒนาโปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตวิชาที่เรียน โดยจะมีการพัฒนาโปรแกรมทั้ง 2 โปรแกรมด้วยกัน คือ โปรแกรมที่เครื่องแม่ข่าย และโปรแกรมที่เครื่องลูกข่าย ซึ่งในความจริงแล้วเครือข่ายคอมพิวเตอร์หนึ่งอาจจะมีการใช้ Server เป็นศูนย์กลางในการเชื่อมต่อกับอินเทอร์เน็ต ทำให้ไม่จำเป็นต้องมีโปรแกรมที่เครื่องลูกข่ายทุกเครื่อง แต่จะมีโปรแกรมที่คอยตรวจจับเว็บไซต์บนเครื่อง Server เพียงเครื่องเดียวแทน โดยส่วนใหญ่แล้ว เครื่อง Server จะไม่อยู่ในห้องเรียน แต่จะอยู่อีกที่ๆ หนึ่งที่มีการรักษาความปลอดภัย เพราะฉะนั้นการที่จะไปติดตั้งโปรแกรมที่เครื่อง Server และสั่งให้โปรแกรมตรวจจับเว็บไซต์นั้นดูเป็นสิ่งที่ไม่เหมาะสมนัก หรือถ้าหากว่ามีการใช้เครื่อง Server มาใช้ในการตรวจจับเว็บไซต์โดยเฉพาะนั้น ก็ถือว่าเป็นการใช้ทรัพยากรไม่คุ้มค่าเท่าที่ควร เพราะเอามาใช้ตรวจจับเว็บไซต์อย่างเดียว นอกจากนี้ การนำโปรแกรมไปติดตั้งบนเครื่อง Server ยังทำให้ไม่สามารถตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานได้อีกด้วย

ดังนั้นเพื่อเป็นการใช้ทรัพยากรที่มีอยู่แล้ว และเพื่อความสะดวกในการติดตั้งโปรแกรม จึงจำเป็นต้องมีโปรแกรมที่เครื่องลูกข่ายทุกเครื่อง

3.1..แนวคิดในการออกแบบ

3.1.1 พัฒนาโปรแกรมที่ทำงานในลักษณะ Client – Server

3.1.2 โปรแกรมที่เครื่องแม่ข่ายมีหน้าตาสำหรับติดต่อกับผู้ใช้งาน ส่วนโปรแกรมที่เครื่องลูกข่ายไม่มีหน้าตาสำหรับติดต่อกับผู้ใช้งาน

3.1.3 โปรแกรมที่เครื่องแม่ข่ายจะควบคุมให้โปรแกรมที่เครื่องลูกข่ายทำงานหรือหยุดทำงานได้ตามต้องการ

3.1.4 โปรแกรมที่เครื่องลูกข่ายจะตรวจจับการเข้าเว็บไซต์หรือการใช้งานโปรแกรมและส่งข้อมูลการแจ้งเตือนมายังโปรแกรมที่เครื่องแม่ข่าย

3.1.5 โปรแกรมที่เครื่องแม่ข่ายจะแจ้งเตือนที่หน้าจอพร้อมทั้งแจ้งเตือนด้วยเสียงเป็นคำพูด

3.2 ความสามารถของโปรแกรม

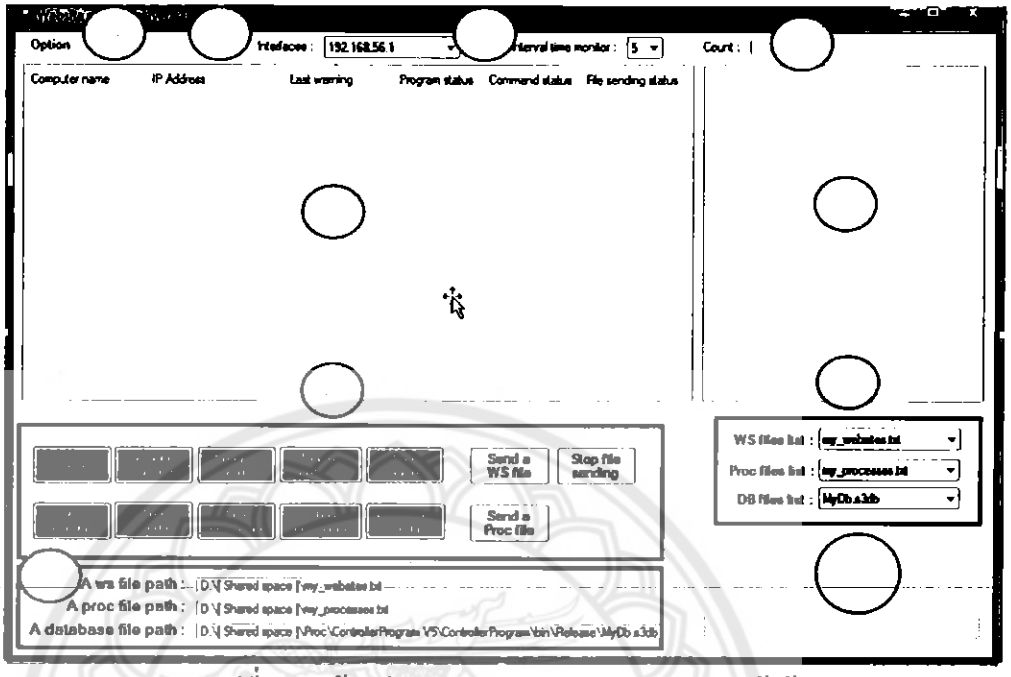
<u>ความสามารถของโปรแกรมที่เครื่องแม่ข่าย</u>

1. โปรแกรมสามารถค้นหาหมายเลขไอพีและชื่อเครื่องภายในวงแลนเดียวกันได้
2. โปรแกรมสามารถตรวจสอบสถานะของเครื่องลูกข่ายและโปรแกรมที่เครื่องลูกข่ายได้
3. โปรแกรมสามารถสั่งให้โปรแกรมที่เครื่องลูกข่ายทำงานและหยุดทำงานตามที่ต้องการได้
4. โปรแกรมสามารถส่งไฟล์ที่เก็บรายชื่อเว็บไซต์หรือโปรแกรมไปยังเครื่องลูกข่ายได้
5. โปรแกรมสามารถดาวน์โหลดรูปภาพที่บันทึกไว้ในขณะเข้าเว็บไซต์หรือใช้งานโปรแกรมที่ไม่อนุญาตจากเครื่องลูกข่ายได้
6. โปรแกรมสามารถบันทึกข้อมูลการเข้าเว็บไซต์ลงฐานข้อมูลและสามารถสรุปข้อมูลการบันทึกเว็บไซต์ทั้งหมดได้
9. โปรแกรมสามารถแจ้งเตือนด้วยเสียงและสามารถปิด เปิดเสียงการแจ้งเตือนได้

<u>ความสามารถของโปรแกรมที่เครื่องลูกข่าย</u>

1. โปรแกรมสามารถตรวจจับเว็บไซต์และแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่ายเมื่อมีการเข้าเว็บที่ไม่อนุญาตได้
2. โปรแกรมสามารถตรวจจับโปรแกรมและแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่ายเมื่อมีการใช้งานโปรแกรมที่ไม่อนุญาตได้
3. โปรแกรมสามารถตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์และแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่ายได้

3.3 การออกแบบส่วนติดต่อกับผู้ใช้ (Graphic User Interface)



รูปที่ 3.1 หน้าตาโปรแกรม WMS สำหรับติดต่อกับผู้ใช้

หมายเลข 1 คือ เมนู Option เป็นเมนูที่เอาไว้สำหรับเปิด ปิดการแจ้งเตือนด้วยเสียง

หมายเลข 2 คือ ช่องสำหรับเลือกหมายเลขไอพีของคอมพิวเตอร์ที่อยู่ภายใน Subnet เดียวกันกับเครื่องลูกข่าย

หมายเลข 3 คือ ช่องสำหรับเลือกเวลา มีหน่วยเป็นนาที เพื่อสั่งให้โปรแกรมที่เครื่องลูกข่ายตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานทุกๆ n นาที

หมายเลข 4 คือ จำนวนเครื่องลูกข่ายที่ค้นหาได้

หมายเลข 5 คือ ส่วนแสดงข้อมูลเกี่ยวกับเครื่องลูกข่าย

- คอลัมน์ Computer name คือ คอลัมน์แสดงสถานะของเครื่องลูกข่าย
- คอลัมน์ IP Address คือ คอลัมน์แสดงหมายเลขไอพีของเครื่องลูกข่าย
- คอลัมน์ Last warning คือ คอลัมน์แสดงสถานะของการแจ้งเตือนครั้งล่าสุด
- คอลัมน์ Program status คือ คอลัมน์แสดงสถานะของโปรแกรมที่เครื่องลูกข่าย
- คอลัมน์ Command status คือ คอลัมน์แสดงสถานะของการสั่งงานต่างๆ
- คอลัมน์ File sending status คือ คอลัมน์แสดงสถานะของการส่งไฟล์

หมายเลข 6 คือ ส่วนที่แสดงข้อมูลการแจ้งเตือนจากเครื่องลูกข่ายทุกเครื่อง

หมายเลข 7 คือ ปุ่มที่เอาไว้สั่งงานโปรแกรม

- ปุ่ม Scan คือ ปุ่มค้นหาหมายเลขไอพีและชื่อเครื่องลูกข่ายทั้งหมดที่อยู่ภายใน Subnet เดียวกัน
- ปุ่ม Computer status คือ ตรวจสอบสถานะของเครื่องลูกข่าย
- ปุ่ม Program status คือ ตรวจสอบสถานะของโปรแกรมที่เครื่องลูกข่าย
- ปุ่ม Monitor process คือ สั่งให้โปรแกรมที่เครื่องลูกข่ายตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน
- ปุ่ม Stop monitor คือ สั่งให้โปรแกรมที่เครื่องลูกข่ายหยุดตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน
- ปุ่ม Start capture คือ ปุ่มสั่งให้โปรแกรมที่เครื่องลูก
- ปุ่ม Stop capture คือ ปุ่มสั่งให้โปรแกรมที่เครื่องลูก
- ปุ่ม Start collecting คือ ปุ่มสั่งให้โปรแกรมที่เครื่องลูก
- ปุ่ม Stop collecting คือ ปุ่มสั่งให้โปรแกรมที่เครื่องลูก
- ปุ่ม Get image capture คือ ปุ่มดาวน์โหลดรูปภาพหน้าจอที่ตรวจจับได้จากเครื่องลูกข่ายพร้อมกับเปิดรูปภาพขึ้นมาแสดง
- ปุ่ม Send WS file คือ ปุ่มส่งไฟล์ที่เก็บรายชื่อเว็บไซต์ไปยังเครื่องลูกข่าย
- ปุ่ม Send Proc file คือ ปุ่มส่งไฟล์ที่เก็บรายชื่อโปรแกรมไปยังเครื่องลูกข่าย
- ปุ่ม Stop file sending คือ ปุ่มสั่งให้โปรแกรมที่เครื่องแม่ข่ายหยุดการส่งไฟล์

หมายเลข 8 คือ ส่วนที่แสดงไฟล์ที่ถูกเพิ่มเข้ามาในโปรแกรม เมื่อคลิกขวาจะแสดงเมนู เช่น เพิ่มไฟล์ ลบไฟล์ เปิดโฟลเดอร์ที่เก็บไฟล์

หมายเลข 9 คือ ส่วนที่แสดงตำแหน่งที่อยู่ของไฟล์

หมายเลข 10 คือ แถบแสดงความก้าวหน้าของการทำงาน (Progress Bar)

3.4 การออกแบบโปรแกรมด้วยภาษา UML (Unified Modeling Language)

การออกแบบโปรแกรมนั้นจะใช้ UML (Unified Modeling Language) ซึ่งเป็นภาษาที่ใช้ในการแสดงแบบจำลองการทำงานของระบบ โดยจะแสดงการออกแบบทั้ง 4 มุมมอง ดังตารางที่ 3.1

ตารางที่ 3.1 ตารางแสดง System Overview ที่ใช้ในการออกแบบโปรแกรม

	Dynamic	**Static**
External	Use case Diagram	Component Diagram
Internal	Activity Diagram	Class Diagram

แต่ละมุมมองมีความหมายดังต่อไปนี้

External-Dynamic	แสดงการติดต่อภายนอกที่สามารถมองเห็นได้
External-Static	แสดงการโครงสร้างและส่วนประกอบต่างๆ ของระบบที่ติดต่อภายนอก
Internal-Dynamic	แสดงพฤติกรรมและสถานะต่างๆ ของระบบ
Internal-Static	แสดงโครงสร้างและส่วนประกอบต่างๆ ภายในระบบ

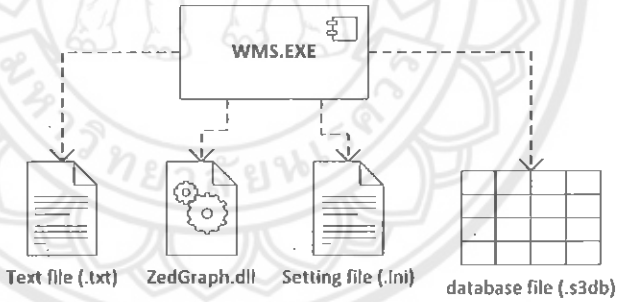
หมายเหตุ โปรแกรม WMS หมายถึง โปรแกรมที่ทำงานบนเครื่องแม่ข่าย โดย WMS ย่อมาจาก Website monitor server

โปรแกรม WMC หมายถึง โปรแกรมที่ทำงานบนเครื่องลูกข่าย โดย WMC ย่อมาจาก Website monitor client

3.4.1 การออกแบบด้วย Component Diagram

3.4.1.1 Component Diagram ของโปรแกรม WMS

โปรแกรม WMS จะทำงานได้อย่างสมบูรณ์นั้นจะต้องประกอบด้วยส่วนต่างๆ ดังรูปที่ 3.2



รูปที่ 3.2 Component ของโปรแกรม WMS

คำอธิบาย

1. Text file (.txt) คือ ไฟล์ที่เอาไว้เก็บตำแหน่งที่อยู่ของไฟล์เก็บรายชื่อเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือไฟล์เก็บรายชื่อโปรแกรมที่ไม่อนุญาตให้ใช้งาน ซึ่งตำแหน่งที่อยู่ของไฟล์นั้นจะถูกใช้ในการส่งไฟล์ไปยังเครื่องลูกข่าย การลบไฟล์ และการเปิดโฟลเดอร์ที่เก็บไฟล์

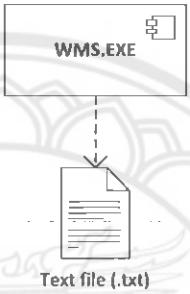
2. ZedGraph.dll คือ ไลบรารีสำหรับใช้ในการวาดกราฟ เพื่อใช้ในการแสดงผลในการสรุปข้อมูลเว็บไซต์ที่ได้บันทึกไว้

3. Setting file (.ini) คือ ไฟล์ที่เอาไว้เก็บค่าตั้งต้นของโปรแกรม โดยจะถูกเรียกใช้เมื่อโปรแกรมเริ่มต้นทำงาน

4. database file (.s3db) คือ ไฟล์ฐานข้อมูลที่ใช้เก็บข้อมูลการเข้าเว็บไซต์สำหรับนำมาวิเคราะห์

3.4.1.2 Component Diagram ของโปรแกรม WMC

โปรแกรม WMC จะทำงานได้อย่างสมบูรณ์นั้นจะต้องประกอบด้วยส่วนต่างๆ ดังรูป 3.3



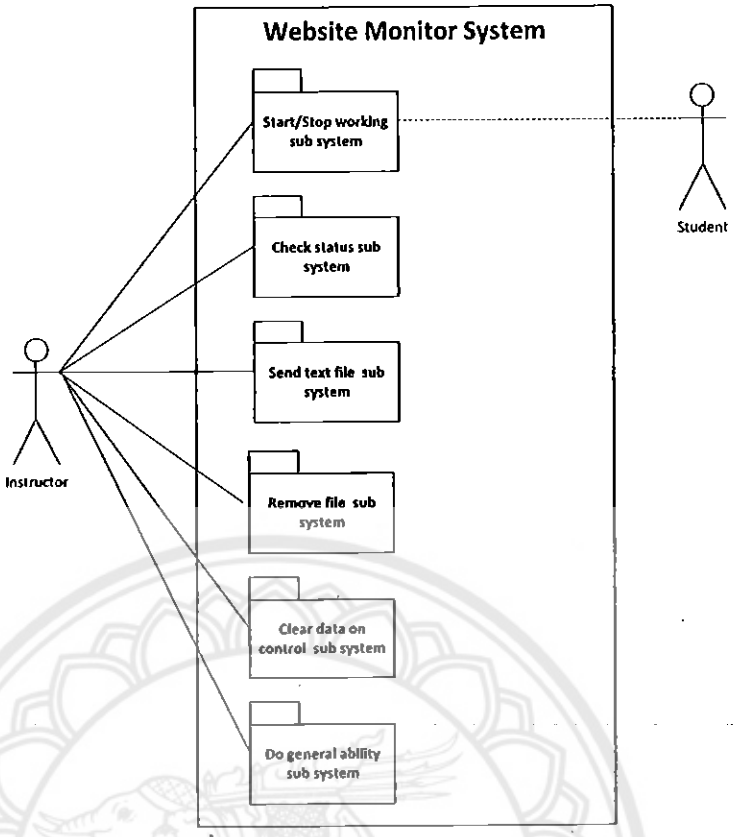
รูปที่ 3.3 Component ของโปรแกรม WMC

คำอธิบาย

1. Text file (.txt) คือ ไฟล์ที่เก็บรายชื่อเว็บไซต์สำหรับใช้ในการตรวจจับการเข้าเว็บไซต์ที่ไม่อนุญาตให้เข้าและรายชื่อโปรแกรมที่ไม่อนุญาตให้ใช้งาน ซึ่งใช้ในการตรวจสอบว่าเว็บไซต์ที่ตรวจจับได้ตรงกับรายชื่อเว็บไซต์ที่อยู่ในไฟล์นี้หรือไม่ หรือถ้าเป็นการตรวจจับโปรแกรมก็จะตรวจสอบว่าโปรแกรมที่ตรวจจับได้นั้นตรงกับรายชื่อโปรแกรมที่อยู่ในไฟล์นี้หรือไม่

3.4.2 การออกแบบด้วย Use case Diagram

Use case Diagram แสดงให้เห็นถึงการใช้งานโปรแกรมของผู้ใช้ว่าสามารถใช้งานอะไรได้บ้าง ดังรูปที่ 3.4



รูปที่ 3.4 Use case Diagram ของระบบ

<u>คำอธิบาย</u>

ระบบนี้มีชื่อว่า Website monitor system จะเห็นได้ว่า Use case Diagram นี้จะมีการแบ่งออกเป็นระบบย่อย เนื่องจากว่ามีข้อจำกัดในเรื่องพื้นที่ และตัวระบบมี Use case ค่อนข้างเยอะ อาจะทำให้ยากต่อการทำความเข้าใจ โดยแต่ละระบบย่อยนั้นจะมี Use case อยู่ภายใน ซึ่งจะอธิบายอีกครั้งหนึ่ง สามารถอธิบายหน้าที่ของแต่ละระบบย่อยได้ ดังนี้

1. Start/Stop working sub system คือ ระบบย่อยที่มีหน้าที่ในการสั่งและหยุดการทำงาน ซึ่งการทำงานที่ว่านี้ก็คือ สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า สั่งให้โปรแกรม WMCตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน และสั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์ ส่วนหยุดก็คือ สั่งให้โปรแกรมหยุดการทำงานอย่างที่ได้กล่าวมา ซึ่งจะมีผู้ใช้งานโดยตรงคือ อาจารย์ผู้สอน ส่วนนักเรียนนั้นจะมีความเกี่ยวข้องในระบบนี้ด้วยเพราะว่าเป็นผู้กระทำทางอ้อมที่ทำให้การตรวจจับต่างๆ นั้นสมบูรณ์

2. Check status sub system คือ ระบบย่อยที่มีหน้าที่ในการตรวจสอบสถานะ ซึ่งมีทั้งสถานะของเครื่องลูกข่าย และสถานะของโปรแกรม WMC ซึ่งจะถูกใช้งานโดยอาจารย์ผู้สอน

3. Send text file sub system คือ ระบบย่อยที่มีหน้าที่ในการส่งไฟล์นามสกุล .txt ไปยังเครื่องลูกข่าย โดยเป็นไฟล์ที่เก็บรายชื่อเว็บไซต์ที่ไม่อนุญาตให้เข้า และเป็นไฟล์ที่เก็บรายชื่อโปรแกรมที่ไม่อนุญาตให้ใช้งาน ซึ่งจะถูกใช้งานโดยอาจารย์ผู้สอน

4. Remove file sub system คือ ระบบย่อยที่มีหน้าที่ในการลบไฟล์นามสกุล .txt และไฟล์ฐานข้อมูลนามสกุล .s3db ออกจากโปรแกรม WMS ซึ่งจะถูกใช้งานโดยอาจารย์ผู้สอน

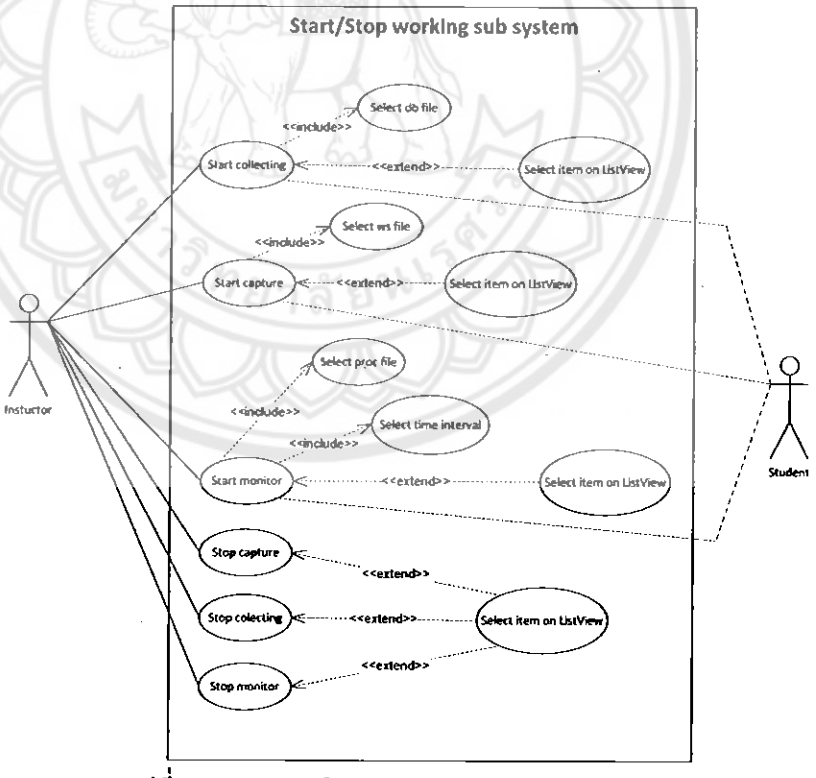
5. Clear data on control sub system คือ ระบบย่อยที่มีหน้าที่ในการลบข้อมูลที่อยู่บนส่วนแสดงผล ซึ่งจะถูกใช้งานโดยอาจารย์ผู้สอน

6. Do general ability sub system คือ ระบบย่อยที่ประกอบด้วย Use case ที่ไม่สามารถจัดให้อยู่ในระบบอื่นๆ ได้ ซึ่งจะได้อธิบายอย่างละเอียดอีกครั้งหนึ่ง

จากการที่ได้อธิบายหน้าที่ของแต่ละระบบย่อยไปแล้วนั้น สามารถที่จะอธิบายรายละเอียดของแต่ละ Use case แต่ละระบบย่อยได้ ดังนี้

3.4.2.1 Use case ใน Start/Stop working sub system

Use case ในระบบนี้ แสดงดังรูปที่ 3.5



รูปที่ 3.5 Use case ใน Start/Stop working sub system

คำอธิบาย

1. Start capture คือ Use case ที่มีหน้าที่สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า โดยจะทำงานได้อย่างสมบูรณ์นั้น จะต้องทำ Use case ที่ชื่อว่า Select ws file ด้วย ซึ่งหมายถึงการเลือกไฟล์ที่เก็บรายชื่อเว็บไซต์ที่ไม่อนุญาตให้ใช้งาน ส่วน Use case ที่ชื่อว่า Select item on ListView นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะสั่งให้ทุกๆ เครื่องตรวจจับ แต่ถ้าทำก็หมายความว่าจะสั่งให้ตรวจจับเฉพาะเครื่องที่เลือก

2. Start monitor คือ Use case ที่มีหน้าที่สั่งให้โปรแกรม WMC ตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน โดยจะทำงานได้อย่างสมบูรณ์นั้น จะต้องทำ Use case ที่ชื่อว่า Select proc file ด้วย ซึ่งหมายถึงการเลือกไฟล์ที่เก็บรายชื่อโปรแกรมที่ไม่อนุญาตให้ใช้งาน และทำ Use case ที่ชื่อว่า Select time interval ด้วย ซึ่งหมายถึงการเลือกเวลาที่จะให้โปรแกรม WMC ตรวจสอบทุกๆ กี่นาทีส่วน Use case ที่ชื่อว่า Select item on ListView นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะสั่งให้ทุกๆ เครื่องตรวจจับ แต่ถ้าทำก็หมายความว่าจะสั่งให้ตรวจจับเฉพาะเครื่องที่เลือก

3. Start collecting คือ Use case ที่มีหน้าที่สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์ โดยจะทำงานได้อย่างสมบูรณ์นั้น จะต้องทำ Use case ที่ชื่อว่า Select db file ด้วย ซึ่ง Select db file คือ การเลือกไฟล์ที่ฐานข้อมูลนามกสุล .s3db ส่วน Use case ที่ชื่อว่า Select item on ListView นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะสั่งให้ทุกๆ เครื่องตรวจจับ แต่ถ้าทำก็หมายความว่าจะสั่งให้ตรวจจับเฉพาะเครื่องที่เลือก

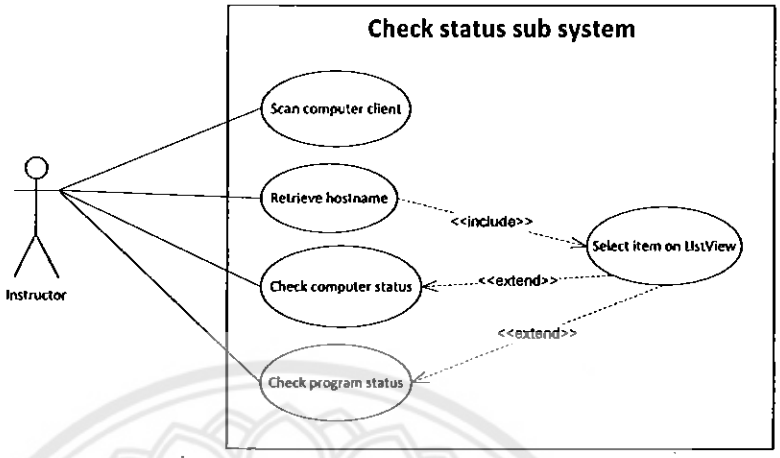
4. Stop capture คือ Use case ที่มีหน้าที่สั่งให้โปรแกรม WMC หยุดตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า โดย Use case ที่ชื่อว่า Select item on ListView นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะสั่งให้ทุกๆ เครื่องหยุดตรวจจับ แต่ถ้าทำก็หมายความว่าจะสั่งให้หยุดตรวจจับเฉพาะเครื่องที่เลือก

5. Stop monitor คือ Use case ที่มีหน้าที่สั่งให้โปรแกรม WMC หยุดตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน โดย Use case ที่ชื่อว่า Select item on ListView นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะสั่งให้ทุกๆ เครื่องหยุดตรวจจับ แต่ถ้าทำก็หมายความว่าจะสั่งให้หยุดตรวจจับเฉพาะเครื่องที่เลือก

6. Stop collecting คือ Use case ที่มีหน้าที่สั่งให้โปรแกรม WMC หยุดตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์ โดย Use case ที่ชื่อว่า Select item on ListView นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะสั่งให้ทุกๆ เครื่องหยุดตรวจจับ แต่ถ้าทำก็หมายความว่าจะสั่งให้หยุดตรวจจับเฉพาะเครื่องที่เลือก

3.4.2.2 Use case ใน Check status sub system

Use case ในระบบนี้ แสดงดังรูปที่ 3.6



รูปที่ 3.6 Use case ใน Check status sub system

คำอธิบาย

1. Scan computer client คือ Use case ที่มีหน้าที่ในการค้นหาหมายเลขไอพีและชื่อเครื่องลูกข่ายที่อยู่ในวงแลนเดียวกัน

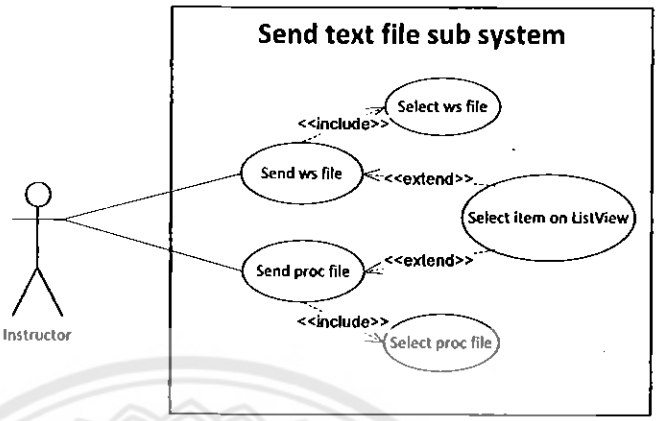
2. Retrieve hostname คือ Use case ที่มีหน้าที่ส่งข้อความไปบอกให้โปรแกรม WMC ส่งชื่อเครื่องของตัวเองกลับมา ในกรณีที่ไม่ทราบชื่อเครื่องหลังจากที่ทำ Use case ที่ชื่อว่า Scan computer client เสร็จ โดยจะต้องทำ Use case ที่ชื่อว่า Select item on ListView ด้วย เพื่อเลือกหมายเลขไอพีปลายทางที่ต้องการส่งข้อความไป

3. Check computer status คือ Use case ที่มีหน้าที่ในการตรวจสอบสถานะของเครื่องลูกข่ายว่ายังทำงานอยู่หรือไม่ ส่วน Use case ที่ชื่อว่า Select item on ListView นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะตรวจสอบสถานะของทุกเครื่อง แต่ถ้าทำก็หมายความว่าจะสั่งให้ตรวจสอบสถานะเฉพาะเครื่องที่เลือก

4. Check program status คือ Use case ที่มีหน้าที่ตรวจสอบสถานะของโปรแกรม WMC ว่ากำลังทำงานอะไรอยู่หรือพร้อมทำงานหรือไม่ โดย Use case ที่ชื่อว่า Select item on ListView นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะตรวจสอบสถานะของทุกๆ เครื่อง แต่ถ้าทำก็หมายความว่าจะตรวจสอบสถานะเฉพาะเครื่องที่เลือก

3.4.2.3 Use case ใน Send text file sub system

Use case ในระบบนี้ แสดงดังรูปที่ 3.7



รูปที่ 3.7 Use case ใน Send text file sub system

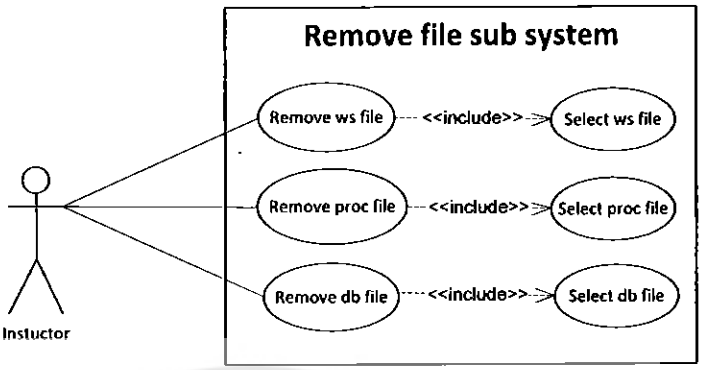
คำอธิบาย

1. Send ws file คือ Use case ที่มีหน้าที่ในการส่งไฟล์นามสกุล .txt ซึ่งเป็นไฟล์ที่เก็บรายชื่อเว็บไซต์ที่ไม่อนุญาตให้เข้าไปยังเครื่องลูกข่าย โดยต้องทำ Use case ที่ชื่อว่า Select ws file ด้วยซึ่ง หมายถึง การเลือกไฟล์ที่เก็บรายชื่อเว็บไซต์ที่ต้องการจะส่งไป ส่วน Use case ที่ชื่อว่า Select item on ListViw นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะส่งไฟล์ไปยังทุกๆ เครื่อง แต่ถ้าทำก็จะส่งไฟล์ไปยังเฉพาะเครื่องที่เลือก

2. Send ws file คือ Use case ที่มีหน้าที่ในการส่งไฟล์นามสกุล .txt ซึ่งเป็นไฟล์ที่เก็บรายชื่อโปรแกรมที่ไม่อนุญาตให้ใช้งานไปยังเครื่องลูกข่าย โดยต้องทำ Use case ที่ชื่อว่า Select proc file ด้วยซึ่ง หมายถึง การเลือกไฟล์ที่เก็บรายชื่อโปรแกรมที่ต้องการจะส่งไป ส่วน Use case ที่ชื่อว่า Select item on ListViw นั้นเป็นทางเลือกว่าจะทำหรือไม่ทำก็ได้ ถ้าไม่ทำก็หมายความว่าจะส่งไฟล์ไปยังทุกๆ เครื่อง แต่ถ้าทำก็จะส่งไฟล์ไปยังเฉพาะเครื่องที่เลือก

3.4.2.4 Use case ใน Remove file sub system

Use case ในระบบนี้ แสดงดังรูปที่ 3.8



รูปที่ 3.8 Use case ใน Remove file sub system

คำอธิบาย

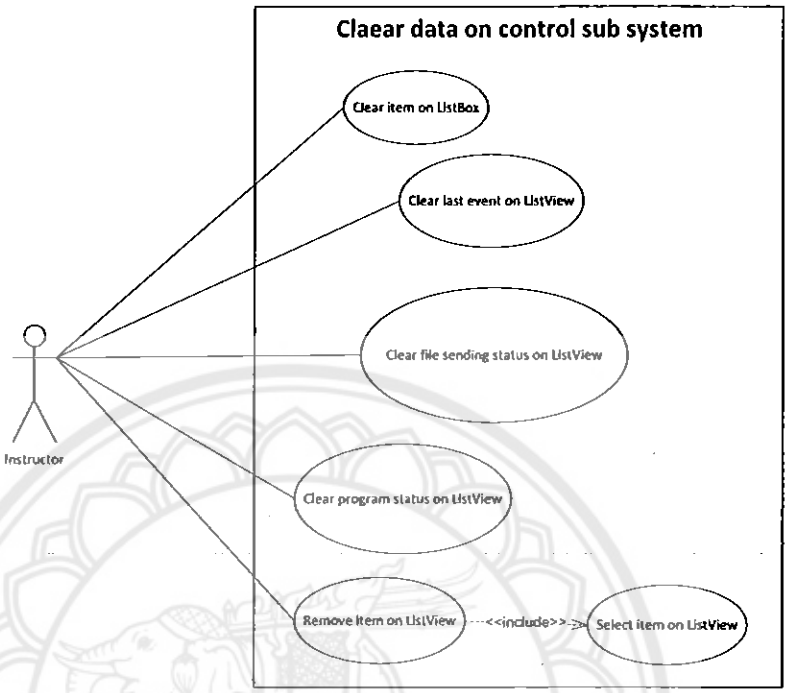
1. Remove ws file คือ Use case ที่มีหน้าที่ในลบไฟล์นามสกุล .txt ซึ่งเป็นไฟล์ที่เก็บรายชื่อเว็บไซต์ที่ไม่อนุญาตให้เข้า โดยต้องทำ Use case ที่ชื่อว่า Select ws file ด้วย ซึ่งหมายถึงการเลือกไฟล์ที่เก็บรายชื่อเว็บไซต์ที่ต้องการจะลบ

2. Remove proc file คือ Use case ที่มีหน้าที่ในลบไฟล์นามสกุล .txt ซึ่งเป็นไฟล์ที่เก็บรายชื่อโปรแกรมที่ไม่อนุญาตให้ใช้งาน โดยต้องทำ Use case ที่ชื่อว่า Select proc file ด้วย ซึ่งหมายถึงการเลือกไฟล์ที่เก็บรายชื่อโปรแกรมที่ต้องการจะลบ

3. Remove db file คือ Use case ที่มีหน้าที่ในการลบไฟล์นามสกุล .s3db ซึ่งเป็นไฟล์ที่ฐานข้อมูลที่เก็บข้อมูลเว็บไซต์ที่ได้บันทึกไว้ โดยต้องทำ Use case ที่ชื่อว่า Select db file ด้วย ซึ่งหมายถึงการเลือกไฟล์ฐานข้อมูลที่ต้องการจะลบ

3.4.2.5 Use case ใน Clear data on control sub system

Use case ในระบบนี้ แสดงดังรูปที่ 3.9



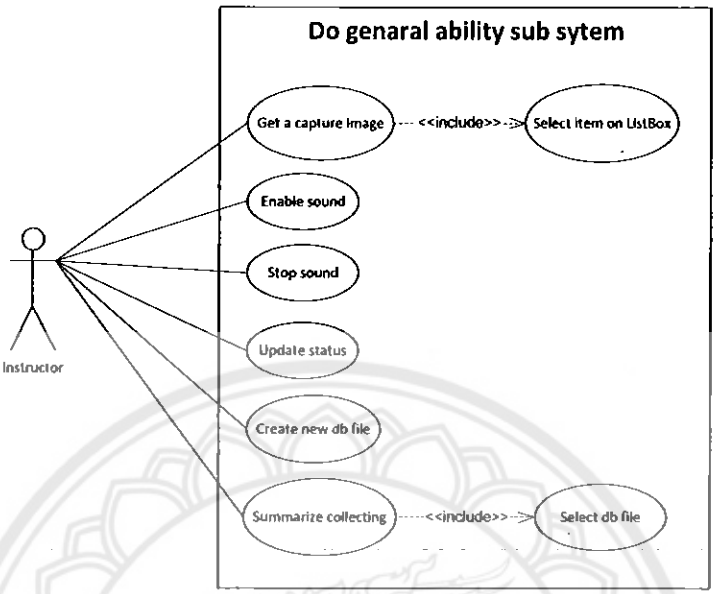
รูปที่ 3.9 Clear data on control sub system

<u>คำอธิบาย</u>

1. Clear item on ListBox คือ Use case ที่มีหน้าที่ลบรายการข้อมูลที่แสดงอยู่บน ListBox
2. Clear last event on ListView คือ Use case ที่มีหน้าที่ลบรายการข้อมูลที่คอลัมน์ Last event ของ ListView
3. Clear file sending status on ListView คือ Use case ที่มีหน้าที่ลบรายการข้อมูลที่คอลัมน์ File sending status ของ ListView
4. Clear program status on ListView คือ Use case ที่มีหน้าที่ลบรายการข้อมูลที่คอลัมน์ Program status ของ ListView
5. Remove item on ListView คือ Use case ที่มีหน้าที่ลบรายการที่แสดงอยู่บน ListView โดยจะต้องทำ Use case ที่ชื่อว่า Select item on ListView ด้วย ซึ่งหมายถึงการเลือกรายการที่ต้องการจะลบ

3.4.2.6 Use case ใน Do general ability sub system

Use case ในระบบนี้ แสดงดังรูปที่ 3.10



รูปที่ 3.10 Do general ability sub system

คำอธิบาย

1. Get capture image คือ Use case ที่มีหน้าที่ในในการดาวน์โหลดรูปภาพจากเครื่องลูกข่ายมาแสดงผล ซึ่งเป็นรูปภาพบันทึกภาพหน้าจอขณะตรวจจับได้ว่ามีการเข้าเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือโปรแกรมที่ไม่อนุญาตให้ใช้งาน โดยต้องทำ Use case ที่ชื่อว่า Select item on ListBox ด้วย ซึ่งหมายถึงการเลือกรายการที่ต้องการจะดาวน์โหลดรูปภาพ

2. Enable sound คือ Use case ที่มีหน้าที่เปิดความเสียงแจ้งเตือนให้ทำงาน

3. Stop sound คือ Use case ที่มีหน้าที่ในการหยุดการแจ้งเตือนด้วยเสียง ซึ่งไม่ใช่การหยุดอย่างถาวร แต่เป็นการหยุดให้ส่งเสียงในคิวทั้งหมด ณ ช่วงเวลานั้นๆ

4. Update status คือ Use case ที่มีหน้าที่อัพเดตสถานะของเครื่องลูกข่าย โดยจะทำการตรวจสอบเครื่องลูกข่ายที่มีอยู่ในรายการก่อน จากนั้นจะทำการตรวจสอบเครื่องลูกข่ายที่ไม่อยู่ในรายการ

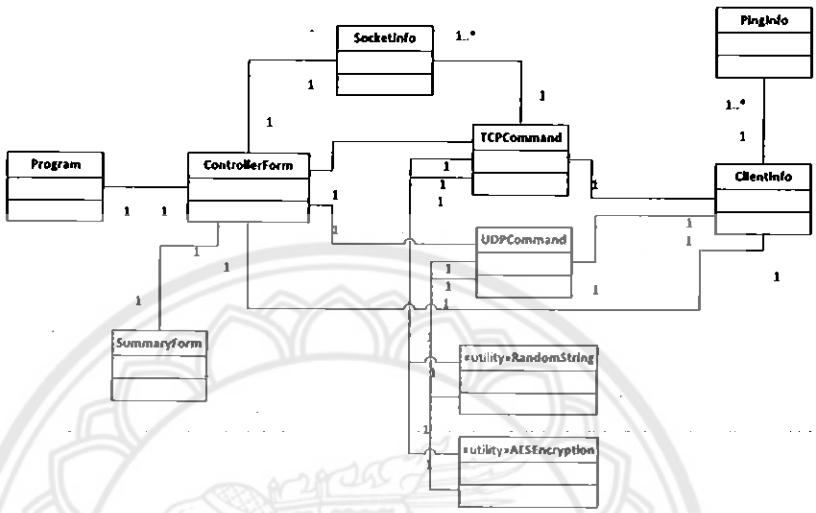
5. Create new db file คือ Use case ที่มีหน้าที่ในการสร้างไฟล์ฐานข้อมูลใหม่

6. Summarize collecting คือ Use case ที่มีหน้าที่สรุปผลการบันทึกข้อมูลเกี่ยวกับเว็บไซต์ที่ได้บันทึกไว้ในฐานข้อมูล โดยต้องทำ Use case ที่ชื่อว่า Select db file ด้วย ซึ่งหมายถึงการเลือกไฟล์ฐานข้อมูลที่ต้องการจะสรุปผลการบันทึก

3.4.3 การออกแบบด้วย Class Diagram

3.4.3.1 Class Diagram ของโปรแกรม WMS

โปรแกรม WMS เมื่อมองในระดับ Class แล้ว สามารถแสดงให้เห็นถึงความสัมพันธ์ระหว่างคลาสได้ดังรูปที่ 3.11



รูปที่ 3.11 Class Diagram ของโปรแกรม WMS

<u>คำอธิบาย</u>

Class Diagram นี้ไม่ได้แสดงให้เห็นถึง Attribute และ Method เนื่องจากว่าพื้นที่จำกัด และไม่ค่อยสะดวกในการทำความเข้าใจ โดยสามารถอธิบายการติดต่อกันของคลาสแต่ละคลาสได้ดังนี้

1. คลาส Program เป็นคลาสที่ใช้ในการรันโปรแกรม โดยจะเรียกให้คลาส Controller Form ทำงาน

2. คลาส ControllerForm เป็นคลาสที่มีหน้าที่แสดงผลส่วนติดต่อกับผู้ใช้ (GUI) รับคำสั่งจากผู้ใช้และจะเรียกใช้คลาส SocketInfo คลาส ClientInfo คลาส TCPCommand คลาส UDPCommand และคลาส SummaryForm ให้ทำงานตามคำสั่งของผู้ใช้

3. คลาส TCPCommand เป็นคลาสที่มีหน้าที่เกี่ยวกับการส่งไฟล์ไปยังเครื่องลูกข่าย การดาวน์โหลดรูปภาพจากเครื่องลูกข่าย โดยใช้โปรโตคอล TCP และจะเรียกใช้ คลาส SocketInfo คลาส ClientInfo คลาส RandomString และคลาส AESEncryption นอกจากนี้ยังมีการเรียกใช้คลาส ControllerForm เพื่ออัพเดตข้อมูลการแสดงผลแก่ผู้ใช้อีกด้วย

4. คลาส UDPCommand เป็นคลาสที่มีหน้าที่เกี่ยวกับส่งข้อความหรือคำสั่งไปยังโปรแกรม WMC รับข้อความตอบกลับจากโปรแกรม WMC และรับข้อความแจ้งเตือนต่างๆ โดยจะ

เรียกใช้คลาส คลาส ClientInfo คลาส RandomString และคลาส AESEncryption นอกจากนี้ยังมีการเรียกใช้คลาส ControllerForm เพื่ออัพเดตข้อมูลการแสดงผลแก่ผู้ใช้อีกด้วย

5. คลาส SocketInfo เป็นคลาสที่ถูกเรียกใช้จากคลาส TCPCommand โดยจะข้อมูลที่ใช้ในการทำงานของคลาส TCPCommand เพื่อช่วยให้การทำงานของคลาส TCPCommand สมบูรณ์มากขึ้น นอกจากนี้ยังถูกเรียกใช้โดยคลาส ControllerForm ในการเก็บข้อมูลเพื่อส่งให้คลาส TCPCommand ทำงานต่อไป

6. คลาส ClientInfo เป็นคลาสที่ทำหน้าที่ในการค้นหาหมายเลขไอพีและชื่อเครื่องลูกข่ายภายในวงแลนเดียวกัน รวมทั้งเก็บข้อมูลของเครื่องลูกข่ายด้วย โดยจะมีเรียกใช้คลาส PingInfo เพื่อเก็บข้อมูลเกี่ยวกับการค้นหาหมายเลขไอพี ช่วยให้การค้นหาหมายเลขไอพีมีความสมบูรณ์มากขึ้น และมีการเรียกใช้คลาส ControllerForm เพื่ออัพเดตข้อมูลการแสดงผลแก่ผู้ใช้

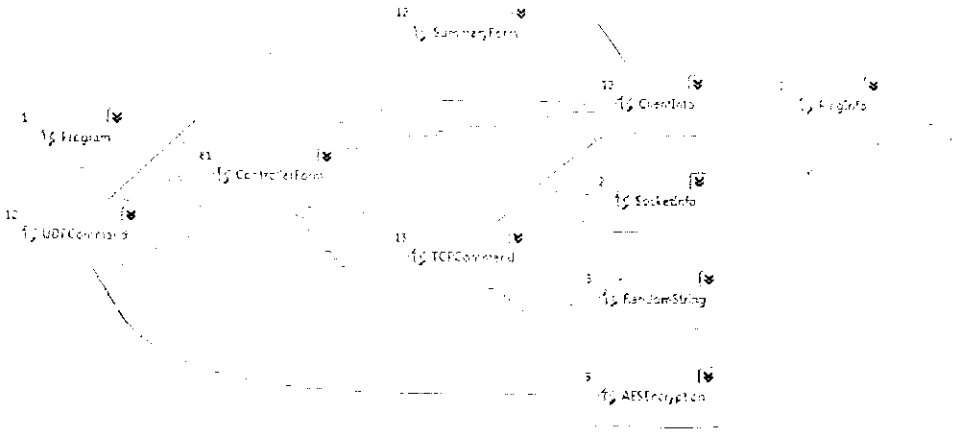
7. คลาส PingInfo เป็นคลาสที่ใช้ในเก็บข้อมูลเกี่ยวกับหมายเลขไอพี ซึ่งจะถูกเรียกใช้จากคลาส ClientInfo เพื่อช่วยให้การทำงานของคลาส ClientInfo มีความสมบูรณ์มากขึ้น

8. คลาส RandomString เป็นคลาสแบบ Static มีหน้าที่ในการสุ่มข้อมูลแบบ String ตามความยาวที่กำหนด เพื่อเอาไปใช้ในการเติมด้านหน้าของข้อความที่จะส่งไปยังโปรแกรม WMC ก่อนที่จะมีการเข้ารหัส เพื่อให้มีความซับซ้อนมากยิ่งขึ้น และจะถูกเรียกใช้โดยคลาส TCPCommand และคลาส UDPCommand

9. คลาส AESEncryption เป็นคลาสแบบ Static มีหน้าที่ในการเข้ารหัสข้อความด้วยอัลกอริทึม AES โดยข้อความที่ว่าจะเป็นข้อความที่ใช้ในการติดต่อสื่อสารกันกับโปรแกรม WMC เพื่อให้มีความปลอดภัยในเรื่องของการถูกเจาะระบบมากยิ่งขึ้น และจะถูกเรียกใช้โดยคลาส TCPCommand และคลาส UDPCommand

10. คลาส SummaryForm เป็นคลาสที่มีหน้าที่ในการวิเคราะห์ข้อมูลเว็บไซต์ที่ได้บันทึกไว้ โดยจะแสดงความถี่การเข้าของแต่ละเว็บไซต์ได้บันทึกไว้ เพื่อนำมาใช้ในการตัดสินใจว่าเว็บไหนไม่ควรจะอนุญาตให้เข้าในระหว่างการเรียนการสอน และจะถูกเรียกใช้โดยคลาส ControllerForm

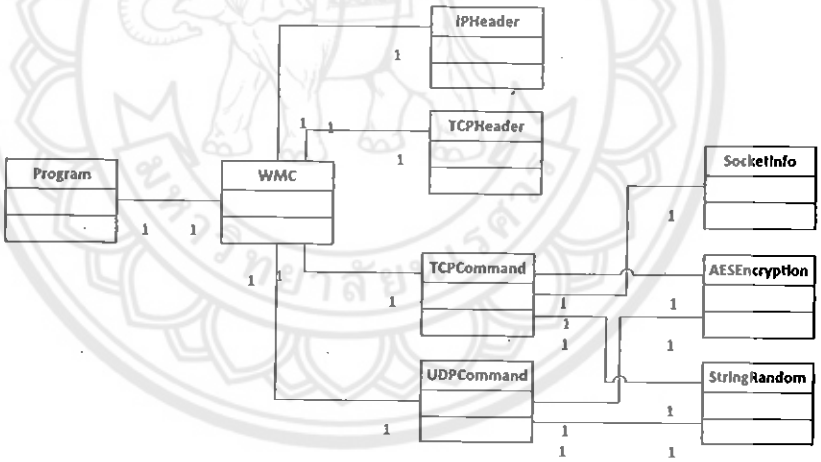
จากรูปที่ 3.11 สามารถมองได้อีกรูปแบบหนึ่ง ดังรูปที่ 3.12 โดยตัวเลขตรงมุมซ้ายมือ คือ จำนวน Method ของคลาสนั้นๆ



รูปที่ 3.12 Dependency Class ของโปรแกรม WMS

3.4.3.2 Class Diagram ของโปรแกรม WMC

โปรแกรม WMC เมื่อมองในระดับ Class แล้ว สามารถแสดงให้เห็นถึงความสัมพันธ์ระหว่างคลาสได้ดังรูปที่ 3.13



รูปที่ 3.13 Class Diagram ของโปรแกรม WMC

<u>คำอธิบาย</u>

Class Diagram นี้ไม่ได้แสดงให้เห็นถึง Attribute และ Method เนื่องจากว่าพื้นที่จำกัด และไม่ค่อยสะดวกในการทำความเข้าใจ โดยสามารถอธิบายการติดต่อกันของคลาสแต่ละคลาสได้ ดังนี้

1. คลาส Program เป็นคลาสที่ใช้ในการรันโปรแกรม โดยจะเรียกให้คลาส WMC ทำงาน

2. คลาส WMC เป็นคลาสที่มีหน้าที่ในการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า ตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน และตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์ โดย

จะเรียกใช้คลาสคลาส IPHeader คลาส TCPHeader คลาส TCPCommand และคลาส UDPCommand

3. คลาส IPHeader เป็นคลาสที่มีหน้าที่ในการสร้าง IP Packet จากข้อมูลที่ได้จากชั้น IP layer ซึ่งอยู่ในรูปแบบ Byte ให้อยู่ในรูปแบบที่ใช้งานได้

4. คลาส TCPHeader เป็นคลาสที่มีหน้าที่ในการถอด IP Header เพื่อสร้างเป็น TCP Packet ให้อยู่ในรูปแบบที่ใช้งานได้

5. TCPCommand เป็นคลาสที่มีหน้าที่ในการรับไฟล์นามสกุล .txt จากเครื่องแม่ข่าย ซึ่งเป็นไฟล์เก็บรายชื่อเว็บไซต์ที่ไม่อนุญาตให้เข้า และเก็บรายชื่อโปรแกรมที่ไม่อนุญาตให้ใช้งาน และมีหน้าที่ในการส่งไฟล์รูปภาพนามสกุล .JPG ไปยังเครื่องแม่ข่ายด้วย จะเรียกใช้นคลาส RandomString และคลาส AESEncryption

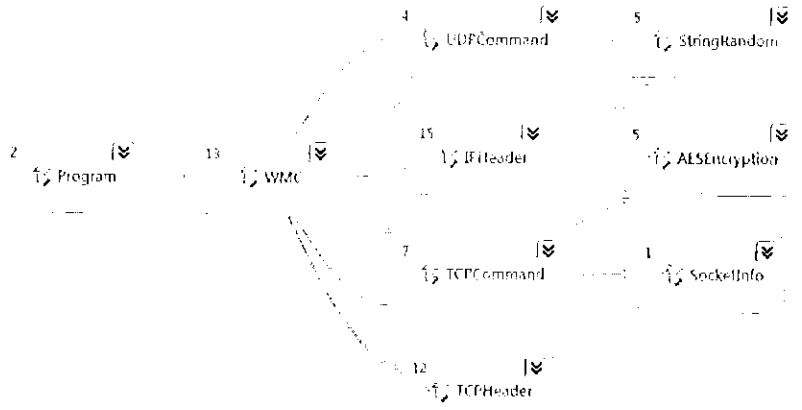
6. UDPCommand เป็นคลาสที่มีหน้าที่ในการรับข้อความซึ่งเป็นคำสั่งให้ทำงาน และส่งข้อความตอบกลับไปยังโปรแกรม WMS นอกจากนี้ยังมีหน้าที่ในการส่งข้อความแจ้งเตือนอีกด้วย จะเรียกใช้คลาส RandomString และคลาส AESEncryption

7. SocketInfo เป็นคลาสที่เอาไว้เก็บข้อมูลในการรับไฟล์นามสกุล .txt จากเครื่องแม่ข่าย ถูกเรียกใช้จากคลาส TCPCommand เพื่อช่วยให้คลาส TCPCommand ทำงานได้อย่างสมบูรณ์มากขึ้น

8. คลาส RandomString เป็นคลาสแบบ Static มีหน้าที่ในการสุ่มข้อความแบบ String ตามความยาวที่ต้องการ เพื่อเอาไปเติมด้านหน้าของข้อความตอบกลับหรือข้อความแจ้งเตือนก่อนการเข้ารหัส เพื่อให้มีความซับซ้อนยิ่งขึ้น โดยจะถูกเรียกใช้จากคลาส TCPCommand และคลาส UDPCommand

9. คลาส AESEncryption เป็นคลาสแบบ Static มีหน้าที่ในการเข้ารหัสข้อความก่อนที่จะส่งไปยังโปรแกรม WMS โดยข้อความที่ว่าคือข้อความตอบกลับ และข้อความการแจ้งเตือน เพื่อให้มีความปลอดภัยจากการถูกโจมตีหรือถูกกลั่นแกล้งจากผู้ไม่ประสงค์ดีได้ระดับหนึ่ง โดยจะถูกเรียกใช้จากคลาส TCPCommand และคลาส UDPCommand

จากรูปที่ 3.13 สามารถมองได้อีกรูปแบบหนึ่ง ดังรูปที่ 3.14 โดยตัวเลขตรงมุมซ้ายมือ คือ จำนวน Method ของคลาสนั้นๆ



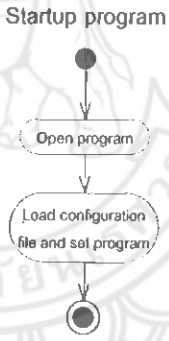
รูปที่ 3.14 Dependency Class ของโปรแกรม WMC

3.4.4 การออกแบบด้วย **Activity Diagram**

3.4.4.1 **Activity Diagram** ของโปรแกรม **WMS**

Activity Diagram จะออกแบบเป็นส่วนๆ ดังนี้

1. Activity Diagram ของการเริ่มต้นรันโปรแกรม

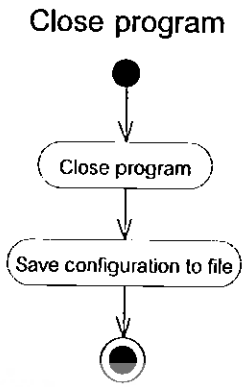


รูปที่ 3.15 Activity Diagram ของการเริ่มต้นรันโปรแกรม

คำอธิบาย

จากรูปที่ 3.14 เมื่อมีการเปิดโปรแกรม โปรแกรมจะเริ่มต้นด้วยการโหลดข้อมูลการตั้งค่าโปรแกรม จากไฟล์ จากนั้นจะเอาข้อมูลเหล่านี้ไปกำหนดค่าโปรแกรมเพื่อให้พร้อมใช้งาน

2. Activity Diagram ของการเริ่มต้นรันโปรแกรม WMS

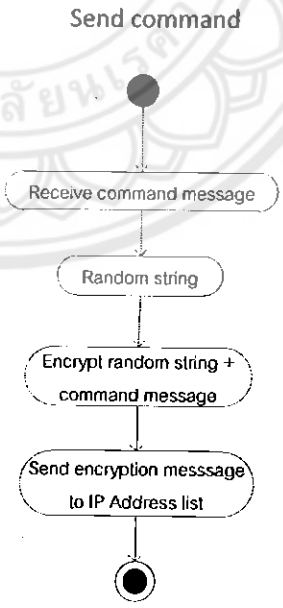


รูปที่ 3.16 Activity Diagram ของการปิดโปรแกรม WMS

<u>คำอธิบาย</u>

จากรูปที่ 3.16 เมื่อมีการปิดโปรแกรม โปรแกรมจะจดจำการใช้งานเกี่ยวกับไฟล์ไว้ โดยการบันทึกชื่อไฟล์ ตำแหน่งที่อยู่ของไฟล์ไว้ เพื่อที่ว่าครั้งต่อไปเมื่อเปิดโปรแกรมมาข้อมูลเหล่านี้จะยังคงอยู่

3. Activity Diagram ของการส่งคำสั่ง

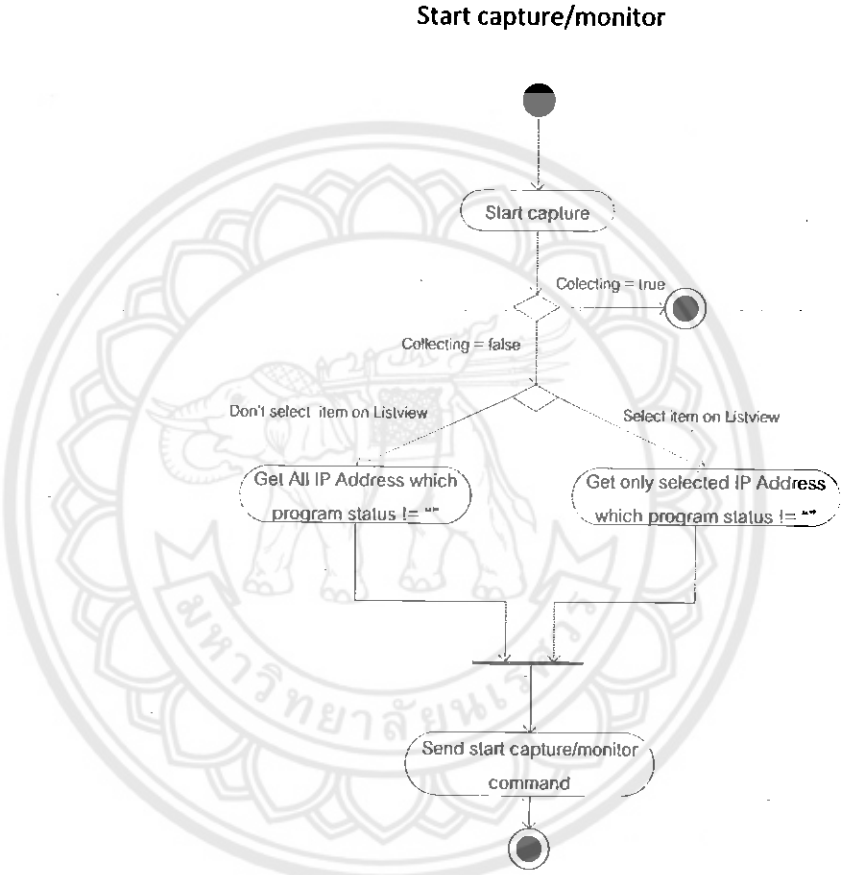


รูปที่ 3.17 Activity Diagram ของการส่งคำสั่ง

คำอธิบาย

จากรูปที่ 3.17 เมื่อมีการรับข้อความมา จะมีการสุ่มข้อความตามความยาวที่กำหนดไว้ จากนั้นทำการเข้ารหัสข้อความที่สุ่มได้บวกกับข้อความที่รับเข้ามา จากนั้นส่งไปยังหมายเลขไอพีของเครื่องลูกข่ายที่เก็บอยู่ใน List

4. Activity Diagram ของการสั่งให้ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือโปรแกรมที่ไม่อนุญาตให้ใช้งาน



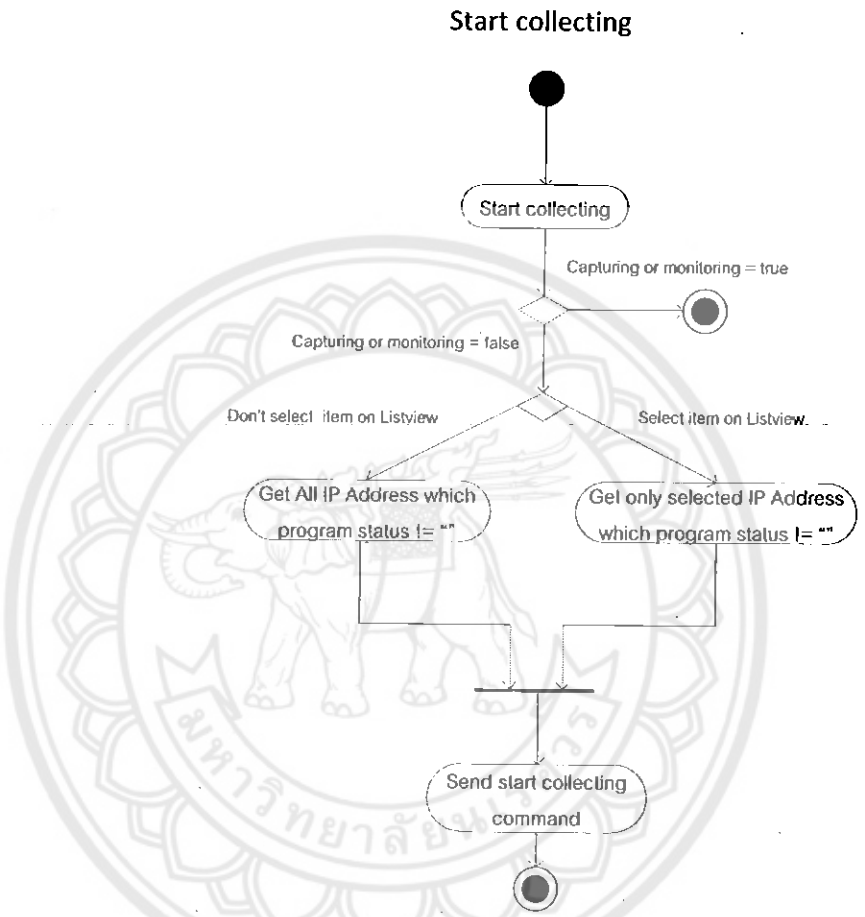
รูปที่ 3.18 Activity Diagram ของการสั่งให้ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือโปรแกรมที่ไม่อนุญาตให้ใช้งาน

คำอธิบาย

จากรูปที่ 3.18 เมื่อมีการสั่งให้ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน ถ้าโปรแกรมกำลังตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์อยู่ ก็จะจบการทำงานทันที แต่ถ้าไม่ใช่ โปรแกรมก็จะพิจารณาว่ามีการเลือกหมายเลขไอพีที่แสดงอยู่บน ListView หรือไม่ ถ้ามี โปรแกรมก็จะเก็บเอาหมายเลขไอพีเฉพาะที่เลือก แล้วส่งไปคำสั่งไป

ยังหมายเลขไอพีเหล่านั้น แต่ถ้าไม่ใช่ โปรแกรมจะส่งคำสั่งไปยังหมายเลขไอพีทั้งหมดที่แสดงอยู่บน ListView

5. Activity Diagram ของการสั่งให้ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์



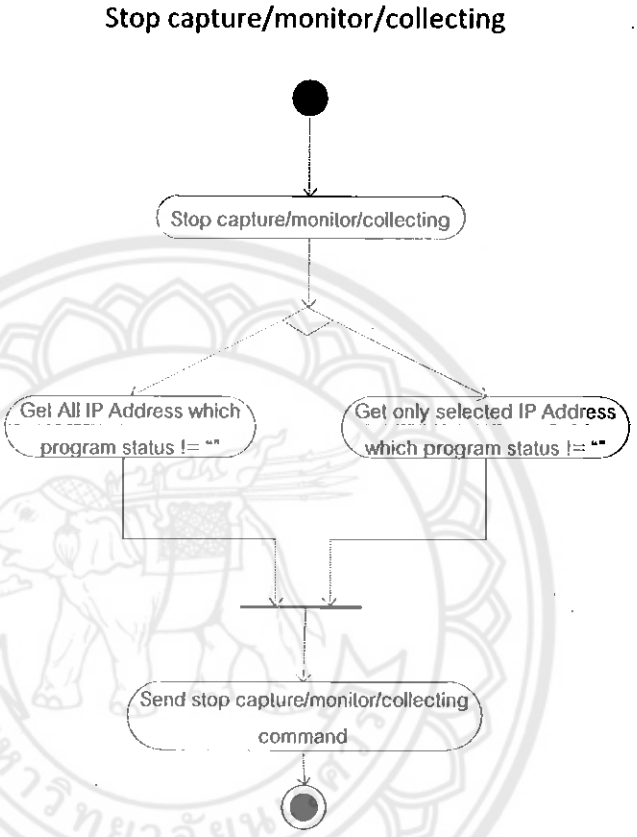
รูปที่ 3.19 Activity Diagram ของการสั่งให้ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์

<u>คำอธิบาย</u>

จากรูปที่ 3.19 เมื่อมีการสั่งให้ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์ ถ้าโปรแกรมกำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือโปรแกรมที่ไม่อนุญาตให้ใช้งานอยู่ ก็จะจบการทำงานทันที แต่ถ้าไม่ใช่ โปรแกรมก็จะพิจารณาว่ามีการเลือกหมายเลขไอพีที่แสดงอยู่บน ListView หรือไม่ ถ้ามี โปรแกรมก็จะเก็บเอาหมายเลขไอพีเฉพาะที่เลือก แล้วส่งไปคำสั่งไปยัง

หมายเลขไอพีเหล่านั้น แต่ถ้าไม่ใช่ โปรแกรมจะส่งคำสั่งไปยังหมายเลขไอพีทั้งหมดที่แสดงอยู่บน ListView

6. Activity Diagram ของการสั่งหยุดการตรวจจับทั้งหมด

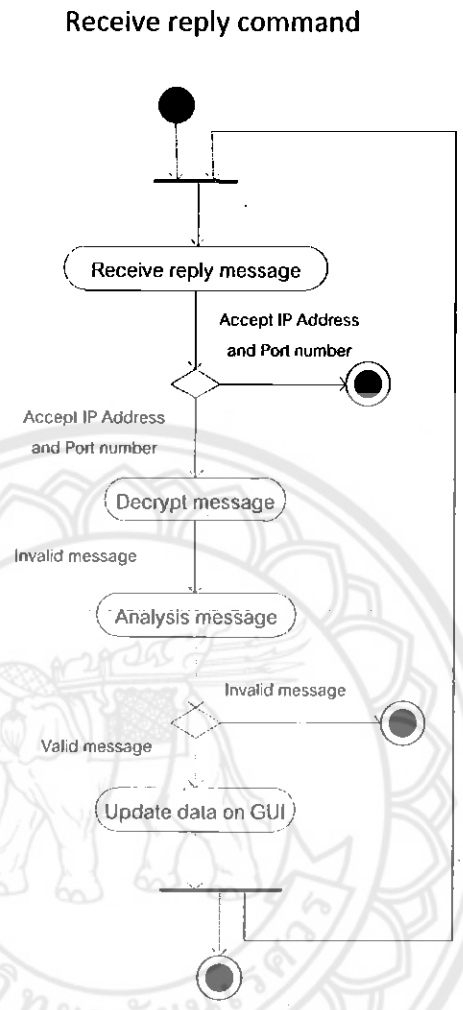


รูปที่ 3.20 Activity Diagram ของการสั่งให้หยุดการตรวจจับทั้งหมด

คำอธิบาย

จากรูปที่ 3.20 เมื่อมีการสั่งให้หยุดการตรวจจับอย่างใดอย่างหนึ่ง โปรแกรมจะพิจารณาว่ามีการเลือกหมายเลขไอพีที่แสดงอยู่บน ListView โดยที่คอลัมน์ Program status ไม่มีค่าว่างหรือไม่ ถ้ามี โปรแกรมก็จะเก็บเอาหมายเลขไอพีเฉพาะที่เลือก แล้วส่งไปคำสั่งไปยังหมายเลขไอพีเหล่านั้น แต่ถ้าไม่ใช่ โปรแกรมจะส่งคำสั่งไปยังหมายเลขไอพีทั้งหมดที่แสดงอยู่บน ListView

7. Activity Diagram ของการรับข้อความตอบกลับ

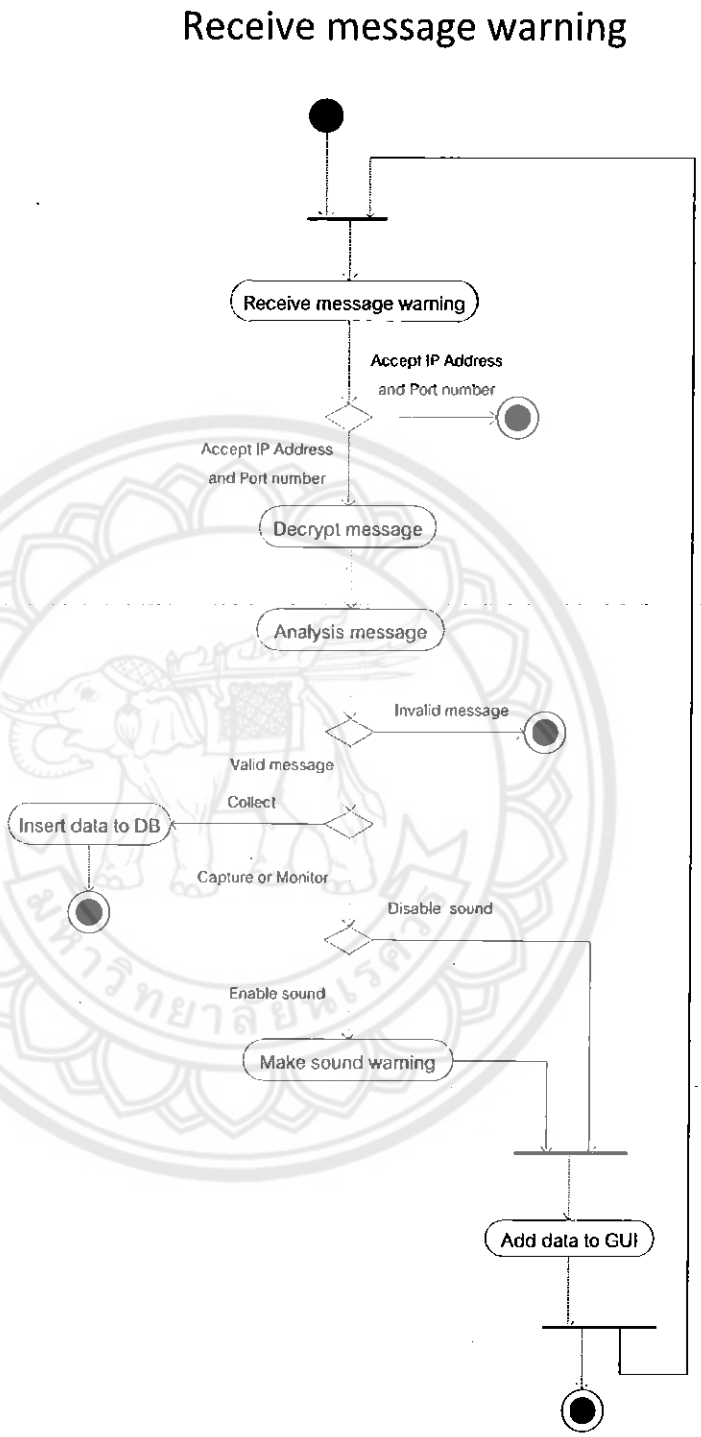


รูปที่ 3.21 Activity Diagram ของการรับข้อความตอบกลับ

คำอธิบาย

จากรูปที่ 3.21 เมื่อมีข้อความตอบกลับมา โปรแกรมจะตรวจสอบว่ายอมรับหมายเลขไอพีและหมายเลขพอร์ตของเครื่องลูกข่ายที่ส่งมานี้หรือไม่ ถ้าไม่ยอมรับ จะจบการทำงาน แต่ถ้ายอมรับโปรแกรมจะถอดรหัสข้อความที่ได้รับ จากนั้นจะแยกแยะข้อความว่าคือข้อความตอบกลับคำสั่งอะไร ถ้าพบว่าข้อความส่งมาไม่ตรงกับเงื่อนไขที่กำหนดไว้ จะจบการทำงาน แต่ถ้าตรงกับเงื่อนไขที่กำหนดไว้ก็จะอัพเดตข้อมูลบน ListView แล้วก็จะไปรอรับข้อความตอบกลับไปเรื่อยๆ

8. Activity Diagram ของการรับข้อความแจ้งเตือน

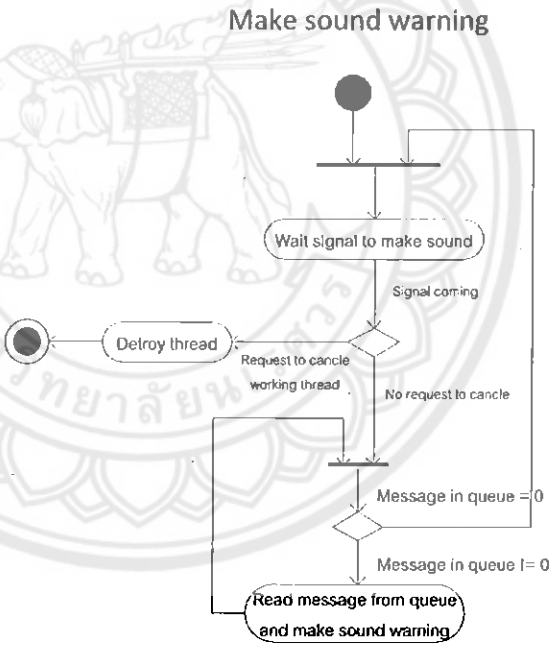


รูปที่ 3.22 Activity Diagram ของการรับข้อความแจ้งเตือน

คำอธิบาย

จากรูปที่ 3.22 เมื่อมีข้อความแจ้งเตือนส่งมา โปรแกรมจะตรวจสอบว่ายอมรับหมายเลขไอพีและหมายเลขพอร์ตของเครื่องลูกข่ายที่ส่งมานี้หรือไม่ ถ้าไม่ยอมรับ จะจบการทำงาน แต่ถ้ายอมรับ โปรแกรมจะถอดรหัสข้อความที่ได้รับ จากนั้นจะแยกแยะข้อความว่าคือข้อความตอบกลับคำสั่งอะไร ถ้าพบว่าข้อความส่งมาไม่ตรงกับเงื่อนไขที่กำหนดไว้ จะจบการทำงาน แต่ถ้าตรงกับเงื่อนไขที่กำหนดไว้ก็พิจารณาต่อว่าถ้าเป็นข้อความแจ้งเตือนสำหรับบันทึกข้อมูลการเข้าเว็บไซต์ ก็จะบันทึกข้อมูลลงฐานข้อมูล แต่ถ้าเป็นข้อความแจ้งเตือนสำหรับเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือโปรแกรมที่ไม่อนุญาตให้ใช้งาน ก็จะพิจารณาต่อว่าถ้าโปรแกรมได้เปิดการแจ้งเตือนไว้ ก็จะแจ้งเตือนด้วยเสียง แล้วอัพเดตข้อมูลบน ListView และ ListBox แต่ถ้ามีการปิดการแจ้งเตือนด้วยเสียงไว้ ก็จะอัพเดตข้อมูลบน ListView และ ListBox แล้วก็จะไปรอรับข้อความแจ้งเตือนไปเรื่อยๆ

9. Activity Diagram ของการแจ้งเตือนด้วยเสียง



รูปที่ 3.23 Activity Diagram ของการแจ้งเตือนด้วยเสียง

คำอธิบาย

จากรูปที่ 3.23 เมื่อมีสัญญาณให้แจ้งเตือนด้วยเสียงส่งมา โปรแกรมก็จะตรวจสอบว่ามีการยกเลิกการแจ้งเตือนด้วยเสียงหรือไม่ ถ้ามีก็จบการทำงานทันที แต่ถ้าไม่ใช่ โปรแกรมจะตรวจสอบว่ามีข้อความที่อยู่ใน Queue หรือไม่ ถ้าไม่มีก็จะไปรอรับข้อสัญญาณการแจ้งเตือนด้วยเสียงใหม่ แต่

ถ้ามี โปรแกรมจะทำการแจ้งเตือนด้วยเสียงตามข้อความที่อยู่ใน Queue จนกว่าจะไม่มีข้อความเหลืออยู่ใน Queue เมื่อ Queue ว่างเปล่าแล้ว จะกลับไปรอรับสัญญาณการแจ้งเตือนด้วยเสียงใหม่

10. Activity Diagram ของการสรุปข้อมูลเว็บไซต์ที่ได้บันทึกไว้ในฐานข้อมูล

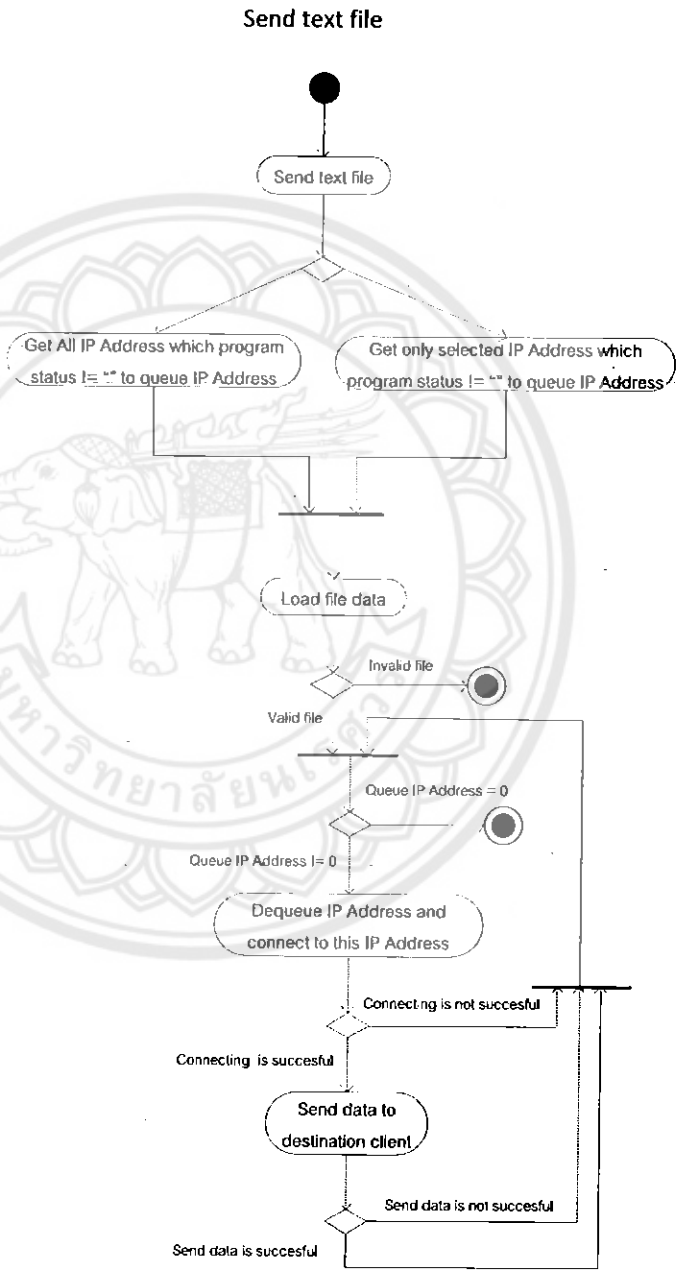
รูปที่ 3.24 Activity Diagram ของการสรุปข้อมูลเว็บไซต์ที่ได้บันทึกไว้ในฐานข้อมูล

<u>คำอธิบาย</u>

จากรูปที่ 3.24 เมื่อต้องการสรุปข้อมูลเว็บไซต์ที่ได้บันทึกไว้ โปรแกรมจะตรวจสอบก่อนว่าไฟล์ฐานข้อมูลที่เลือกมีอยู่จริงหรือไม่ ถ้าไม่มีอยู่จริงก็จะจบการทำงาน แต่ถ้ามีอยู่จริงก็จะคิวรีข้อมูลเว็บไซต์ที่แตกต่างกันจากฐานข้อมูลมาเก็บไว้ใน DataTable จากนั้นอ่านข้อมูลที่อยู่ใน DataTable ไว้ใน List จากนั้นคิวรีเพื่อนับจำนวนความถี่ของเว็บไซต์แต่ละเว็บไซต์ โดยใช้ชื่อ

เว็บไซต์ที่อยู่ใน List แล้วเก็บข้อมูลที่ได้ไว้ใน DataTable จากนั้นอ่านข้อมูลจาก DataTable มาเก็บไว้ใน Dictionary ทำการจัดเรียงข้อมูลใน Dictionary จากเว็บไซต์ที่มีความถี่น้อยไปหามาก จากนั้นอ่านข้อมูลจาก Dictionary เพื่อนำไปสร้างเป็นกราฟรูปวงกลม และแสดงข้อมูลบน ListView ต่อไป

11. Activity Diagram ของการสรุปข้อมูลเว็บไซต์ที่ได้บันทึกไว้ในฐานข้อมูล

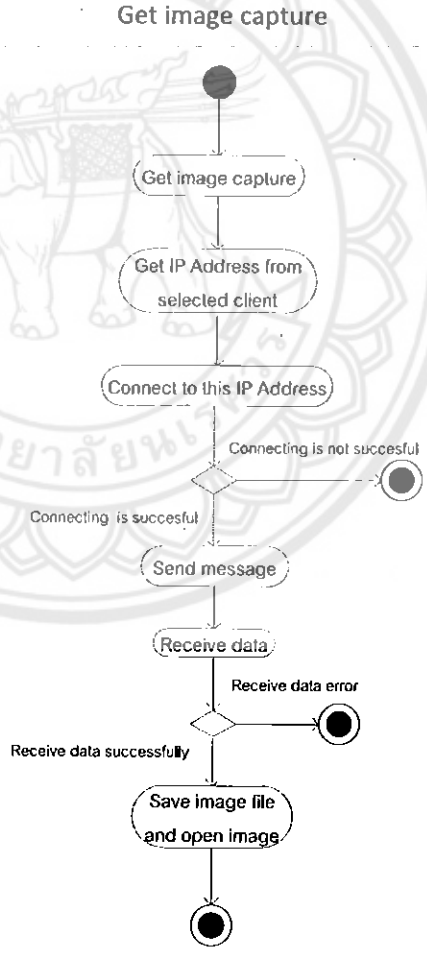


รูปที่ 3.25 Activity Diagram ของการสรุปข้อมูลเว็บไซต์ที่ได้บันทึกไว้ในฐานข้อมูล

<u>คำอธิบาย</u>

จากรูปที่ 3.25 เมื่อต้องการส่งไฟล์ โปรแกรมจะตรวจสอบดูว่ามีการเลือกหมายเลขไอพีที่อยู่บน ListView โดยที่คอลัมน์ Program status ไม่มีค่าว่างหรือไม่ ถ้ามีก็จะเก็บเฉพาะหมายเลขไอพีที่เลือกไว้ใน List ถ้าไม่มีก็จะเก็บหมายเลขไอพีทั้งหมดที่อยู่บน ListView ไว้ใน Queue จากนั้นอ่านข้อมูลจากไฟล์ที่ต้องการส่ง โดยจะตรวจสอบว่าเป็นข้อมูลที่ถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็จบการทำงาน แต่ถ้าถูกต้องก็จะตรวจสอบว่า List มีหมายเลขไอพีอยู่หรือไม่ ถ้าไม่มีก็จะจบการทำงาน ถ้ามีก็จะเอาข้อมูลออกจาก Queue จากนั้นทำการเชื่อมต่อไปยังหมายเลขไอพีแต่ละหมายเลข ถ้าการเชื่อมต่อสำเร็จก็จะส่งข้อมูลไป แต่ถ้าไม่สำเร็จก็จะทำการเชื่อมต่อใหม่กับหมายเลขไอพีที่เหลืออยู่ เมื่อส่งข้อมูลเสร็จแล้วหรือไม่เสร็จก็ตาม โปรแกรมจะไปทำการเชื่อมต่อหมายเลขไอพีที่เหลืออยู่เพื่อส่งข้อมูลจนกว่าคิวจะว่างเปล่า

12. Activity Diagram ของการดาวน์โหลดรูปภาพจากเครื่องลูกข่าย

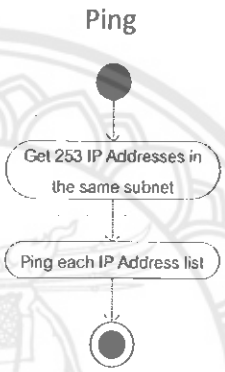


รูปที่ 3.26 Activity Diagram ของการดาวน์โหลดรูปภาพจากเครื่องลูกข่าย

<u>คำอธิบาย</u>

จากรูปที่ 3.26 เมื่อต้องการดาวน์โหลดรูปภาพจากเครื่องลูกข่าย โปรแกรมจะหาหมายเลขไอพีจากรายการที่เลือกบน ListView หรือ ListBox จากนั้นทำการเชื่อมต่อไปยังเครื่องลูกข่ายที่มีหมายเลขไอพีนั้นๆ เมื่อเชื่อมต่อไม่สำเร็จจะจบการทำงาน แต่ถ้าเชื่อมต่อสำเร็จจะส่งข้อความไปว่าต้องการดาวน์โหลดรูปภาพ จากนั้นรอรับข้อมูลจากเครื่องลูกข่าย เมื่อรับข้อมูลเสร็จแล้วก็จะบันทึกข้อมูลเป็นรูปภาพ แล้วเปิดรูปภาพขึ้นมาแสดง แต่ถ้ารับข้อมูลไม่สำเร็จก็จะจบการทำงาน

13. Activity Diagram ของการตรวจสอบสถานะของเครื่องลูกข่าย

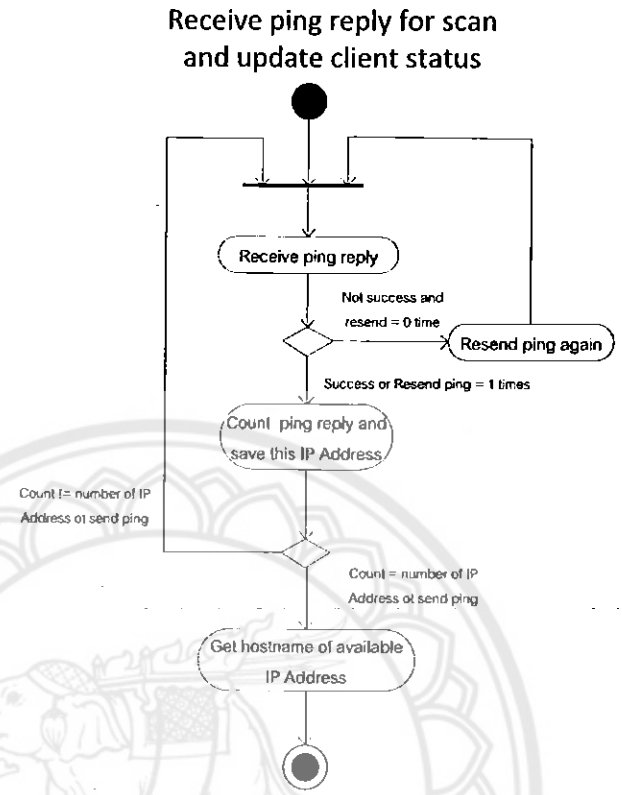


รูปที่ 3.27 Activity Diagram ของการตรวจสอบสถานะของเครื่องลูกข่าย

<u>คำอธิบาย</u>

จากรูปที่ 3.27 เมื่อต้องการตรวจสอบสถานะของเครื่องลูกข่าย โปรแกรมจะเก็บหมายเลขไอพีจำนวน 253 หมายเลข ไว้ใน List จากนั้นก็จะทำการส่ง ICMP Request ไปยังหมายเลขไอพีทั้งหมดที่อยู่ใน List

14. Activity Diagram ของการตรวจสอบสถานะของเครื่องลูกข่าย



รูปที่ 3.28 Activity Diagram ของการตรวจสอบสถานะของเครื่องลูกข่าย

<u>คำอธิบาย</u>

จากรูปที่ 3.28 เมื่อมี ICMP Reply ตอบกลับมา ถ้าการตอบกลับมานั้นสำเร็จหรือจำนวนครั้งในการส่ง ICMP Request ใหม่เท่ากับหนึ่ง ก็จะทำการนับจำนวนหมายเลขไอพีที่ตอบกลับมาและเก็บหมายเลขไอพีนี้ไว้ใน List ถ้าการตอบกลับมานั้นไม่สำเร็จ ก็จะทำการส่ง ICMP Request ไปใหม่อีกรอบ พร้อมกับนับจำนวนครั้งของการส่งใหม่นี้เพิ่มขึ้นอีกหนึ่งครั้ง ถ้าจำนวนครั้งการตอบกลับยังไม่เท่ากับจำนวนหมายเลขไอพีที่ส่ง ICMP Request ไป ก็กลับไปรอรับ ICMP Reply ใหม่อีกรอบ แต่ถ้ามีการตอบกลับมาครบแล้ว ก็จะไปค้นหาชื่อเครื่องของหมายเลขไอพีที่ตอบกลับมานี้ต่อไป

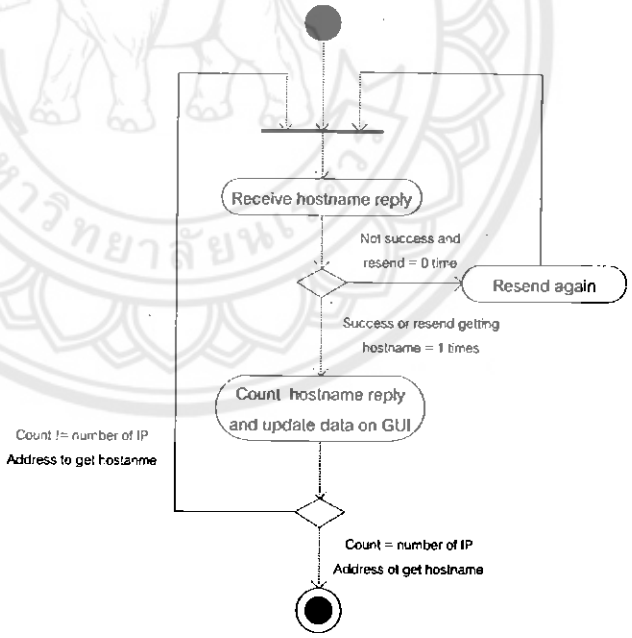
15. Activity Diagram ของการร้องขอชื่อเครื่องลูกข่าย

รูปที่ 3.29 Activity Diagram ของการร้องขอชื่อเครื่องลูกข่าย

คำอธิบาย

จากรูปที่ 3.29 จะทำการร้องขอชื่อเครื่องลูกข่าย โดยการส่ง LLMNR Query ไปยังหมายเลขไอพีทั้งหมดที่อยู่ใน List ซึ่งได้จากการขั้นตอนการตรวจสอบสถานะของเครื่องลูกข่าย

16. Activity Diagram ของการรับการตอบการร้องขอชื่อเครื่องลูกข่าย



รูปที่ 3.30 Activity Diagram ของการรับการตอบการร้องขอชื่อเครื่องลูกข่าย

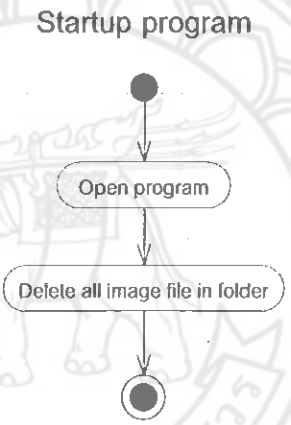
คำอธิบาย

จากรูปที่ 3.30 เมื่อมีการตอบกลับมาจากการร้องขอชื่อเครื่องลูกข่าย ถ้าการตอบกลับมานั้นสำเร็จหรือจำนวนครั้งการส่งการร้องขอเท่ากับหนึ่ง ก็จะนับจำนวนการตอบกลับมา และอัพเดตข้อมูลการแสดงผลบน GUI แต่ถ้าไม่สำเร็จก็จะทำการส่งการร้องขอไปใหม่พร้อมกับเพิ่มจำนวนการขอไปอีกหนึ่ง ถ้าจำนวนครั้งการตอบกลับมาเท่ากับจำนวนหมายเลขไอพีที่ส่งไปก็จะเสร็จสิ้นการทำงาน แต่ถ้ายังไม่เท่าก็จะกลับไปรอรับการตอบกลับมาใหม่

3.4.4.2 Activity Diagram ของโปรแกรม WMC

Activity Diagram จะออกแบบเป็นส่วนๆ ดังนี้

1. Activity Diagram ของการเริ่มต้นรันโปรแกรม WMC

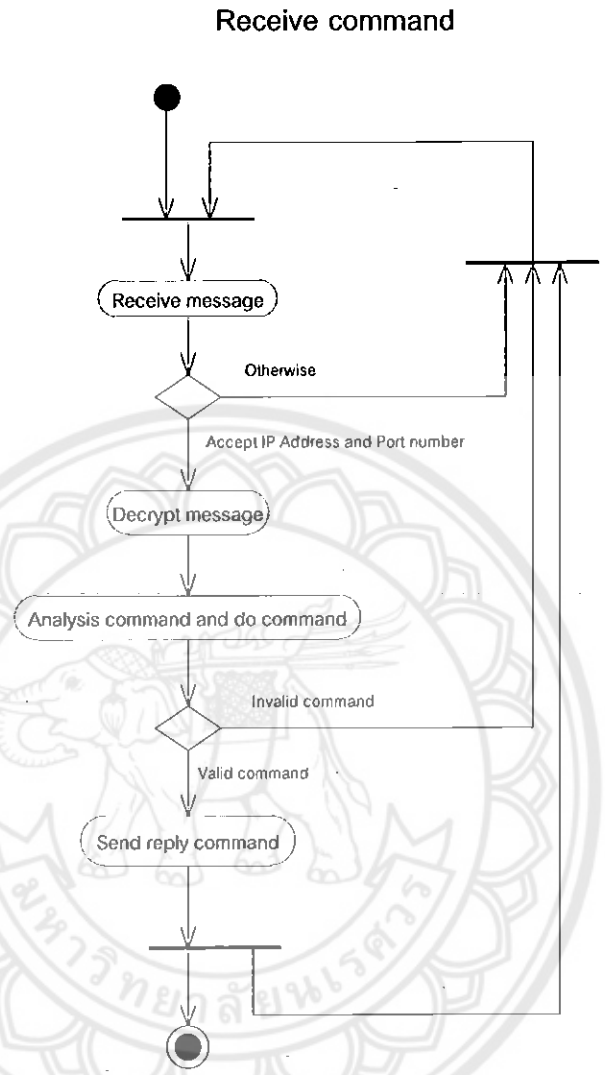


รูปที่ 3.31 Activity Diagram ของการเริ่มต้นรันโปรแกรม

คำอธิบาย

จากรูปที่ 3.31 เมื่อมีการเปิดโปรแกรม โปรแกรมจะเริ่มต้นด้วยการลบรูปภาพที่อยู่ในโฟลเดอร์ที่เก็บทั้งหมด จากนั้นโปรแกรมทำงานโดยการรอรับคำสั่งจากโปรแกรม WMS ต่อไป

2. Activity Diagram ของการรับคำสั่งจากโปรแกรม WMS

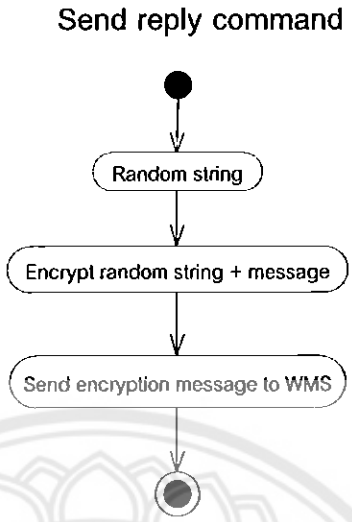


รูปที่ 3.32 Activity Diagram ของการรับคำสั่งจากโปรแกรม WMS

<u>คำอธิบาย</u>

จากรูปที่ 3.32 เมื่อมีการรับข้อความที่ส่งมา จะทำการตรวจสอบว่าถูกส่งมาจากเครื่องแม่ข่ายจริงหรือไม่ โดยการตรวจสอบหมายเลขไอพีและหมายเลขพอร์ตว่าตรงกับที่มีอยู่หรือไม่ ถ้าไม่ตรงก็จะกลับไปรอรับข้อความใหม่ แต่ถ้าตรงก็จะถอดรหัสข้อความ แยกแยะข้อความ ถ้าข้อความที่ส่งมาตรงกับเงื่อนไขที่กำหนดไว้ก็จะส่งข้อความตอบกลับไปยังโปรแกรม WMS จากนั้นจะกลับไปรอรับข้อความใหม่ต่อไป แต่ถ้าไม่ตรงก็จะกลับไปรอรับข้อความใหม่เช่นกัน

3. Activity Diagram ของการส่งข้อความตอบกลับไปยังโปรแกรม WMS

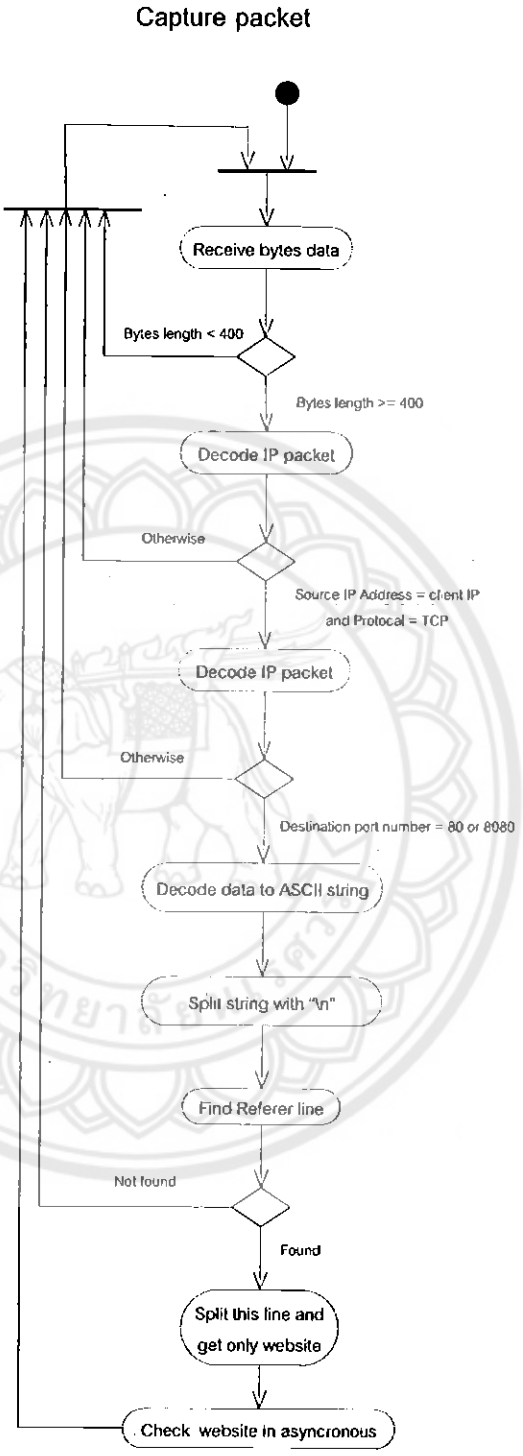


รูปที่ 3.33 Activity Diagram ของการรับคำสั่งจากโปรแกรม WMS

คำอธิบาย

จากรูปที่ 3.33 เริ่มจากทำการสุ่มข้อความตามยาวที่กำหนดไว้ จากนั้นทำการเข้ารหัสข้อความที่สุ่มได้บวกกับข้อความที่ต้องการส่ง จากนั้นทำการส่งข้อความที่ได้จากการเข้ารหัสนี้ไปยังโปรแกรม WMS

4. Activity Diagram ของการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า



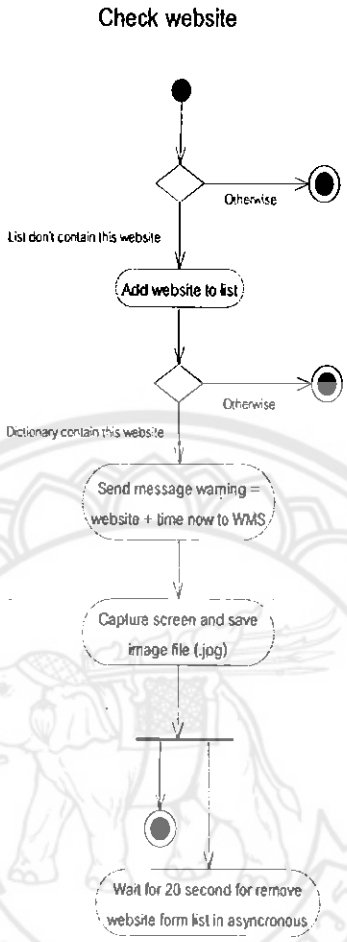
รูปที่ 3.34 Activity Diagram ของการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า

<u>คำอธิบาย</u>

หลักการทำงาน คือ โปรแกรมจะตรวจจับแพ็คเกต (Packet) เฉพาะตอนส่งออก โดยเป็นแพ็คเกตของโปรโตคอล TCP ที่มีหมายเลขพอร์ตปลาย คือ 80 หรือ 8080 และบรรจุ HTTP Message ที่เป็นคำสั่ง HTTP GET ซึ่งภายใน HTTP Message จะมีบรรทัดที่ชื่อ Referer: ตามด้วยชื่อ URL ซึ่งบรรทัดนี้จะเก็บ URL เพื่อให้ทราบว่าการร้องขอ URL ใหม่นี้มาจาก URL อะไร เราจะเอาเฉพาะชื่อโดเมนหรือเว็บไซต์เท่านั้น โดยการตัดส่วนที่ไม่จำเป็นของ URL ทิ้ง จากนั้นก็จะนำไปตรวจสอบกับเว็บไซต์ที่เก็บอยู่ใน List ถ้ายังไม่มีเว็บไซต์อยู่ใน List ก็จะเพิ่มเว็บไซต์นี้เข้าไป จากนั้นจึงไปตรวจสอบกับ Dictionary ซึ่ง Dictionary นี้ ได้มาจากการที่โปแกรม WMS สั่งให้ตรวจจับเว็บไซต์ แล้วโปรแกรมจะไปโหลดรายชื่อเว็บไซต์จากไฟล์แล้วนำมาเก็บใน Dictionary เมื่อตรวจสอบว่ามีชื่อเว็บไซต์ที่ตรงกันก็จะส่งข้อมูลการแจ้งเตือนไปยังโปรแกรม WMS การเก็บเว็บไซต์ไว้ใน List นี้ เพื่อไม่ต้องการให้มีการตรวจสอบกับเว็บไซต์ที่อยู่ใน Dictionary ที่ถี่เกินไปโดยไม่จำเป็น เพราะว่าการเข้าเว็บไซต์หนึ่งๆ จะมีการส่ง HTTP GET หลายครั้ง ทำให้ตรวจจับเว็บไซต์ชื่อเดิมได้หลายครั้ง เพื่อแก้ปัญหานี้จึงเก็บเว็บไซต์ไว้ใน List ไว้ 20 วินาที จึงค่อยทำการลบออกเพื่อยอมให้มีการตรวจสอบเว็บไซต์นี้กับ Dictionary อีกครั้ง

จากรูปที่ 3.34 เมื่อมีการรับข้อมูลจากชั้น IP layer มาแล้ว จะตรวจสอบว่าขนาดของข้อมูลในหน่วยไบต์น้อยกว่า 400 ไบต์หรือไม่ ถ้าน้อยกว่าก็จะกลับไปรอรับข้อมูลใหม่ แต่ถ้ามากกว่าหรือเท่ากับ จะทำการแปลงข้อมูลให้อยู่ในรูปของ IP Packet ที่สามารถเข้าใจได้ แล้วทำการตรวจสอบหมายเลขไอพีต้นทางว่าตรงกับหมายเลขไอพีของตัวเองและเป็น Packet ของโปรโตคอล TCP หรือไม่ ถ้าไม่ใช่ก็กลับไปรอรับข้อมูลใหม่ ถ้าใช่ก็จะถอด IP Header ออก จะได้ TCP Packet จากนั้นตรวจสอบว่าหมายเลขพอร์ตปลายทาง คือ 80 หรือ 8080 หรือไม่ ถ้าไม่ใช่ก็กลับไปรอรับข้อมูลใหม่ ถ้าใช่ก็จะถอดรหัสแอสกี้ (ASCII Code) ของข้อมูลใน TCP Packet ซึ่งเป็น HTTP Message จากนั้นค้นหาบรรทัด Referer: และตัดข้อความเอาแต่ชื่อเว็บไซต์ จากนั้นก็จะส่งชื่อเว็บไซต์นี้ไปตรวจสอบในแบบ Asynchronous ต่อไป จากนั้นกลับไปรอรับข้อมูลใหม่ต่อไป แต่ถ้าไม่เจอบรรทัด Referer: ก็จะกลับไปรอรับข้อมูลใหม่

5. Activity Diagram ของการตรวจสอบเว็บไซต์ที่ไม่อนุญาตให้เข้า



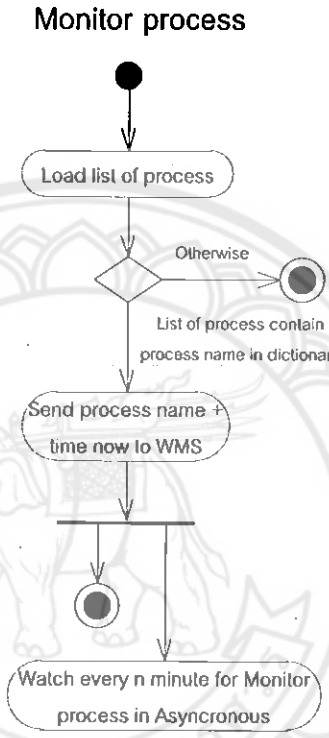
รูปที่ 3.35 Activity Diagram ของการตรวจสอบเว็บไซต์ที่ไม่อนุญาตให้เข้า

คำอธิบาย

จากรูปที่ 3.35 เมื่อมีการรับเว็บไซต์ที่ส่งมา จะตรวจสอบว่าเว็บไซต์มีอยู่ใน List หรือไม่ ถ้ามีก็จบการทำงาน แต่ถ้ายังไม่มีก็จะเพิ่มเว็บไซต์นี้เก็บไว้ใน List จากนั้นตรวจสอบกับ Dictionary ว่าเว็บไซต์นี้เป็นเว็บที่ไม่อนุญาตให้เข้าหรือไม่ ถ้าไม่ใช่ก็จบการทำงาน แต่ถ้าใช่ก็จะส่งข้อความแจ้งเตือนไปยังโปรแกรม WMS และทำการบันทึกภาพหน้าจอของเครื่องลูกข่ายขณะนั้นเก็บไว้ เพื่อใช้เป็นหลักฐานยืนยันว่ามีการเข้าเว็บไซต์นี้จริงๆ จากนั้นรอคอยเป็นเวลา 20 วินาทีในแบบ Asynchronous เพื่อที่จะลบเว็บไซต์ออกจาก List. เหตุผลที่ต้องมีการบันทึกภาพหน้าจอไว้นั้นเพราะว่าโปรแกรมไม่สามารถตรวจจับเว็บไซต์ที่กำลังเข้าชมอยู่ได้อย่างแม่นยำ เนื่องจากว่าบางเว็บไซต์หรือบาง URL มีการไปอ้างอิงหรือไปดึงข้อมูลจากที่อื่นๆ ทำให้การเข้าชมเว็บไซต์หนึ่งๆ อาจจะทำให้การแจ้งเตือนเว็บไซต์ที่ไม่อนุญาตให้เข้านั้นไม่ถูกต้องตามความเป็นจริงได้ เช่น ไม่

อนุญาตให้เข้าเว็บไซต์ facebook.com แต่พอเข้าเว็บไซต์ f0nt.com ก็จะทำให้มีการแจ้งเตือนว่ามีการเข้าเว็บไซต์ facebook.com เพราะเว็บไซต์ f0nt.com มีการไปอ้างอิงหรือไปดึงข้อมูลจากเว็บไซต์ facebook.com มาใช้งานหรือแสดงผล

6. Activity Diagram ของการตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน



รูปที่ 3.36 Activity Diagram ของการตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน

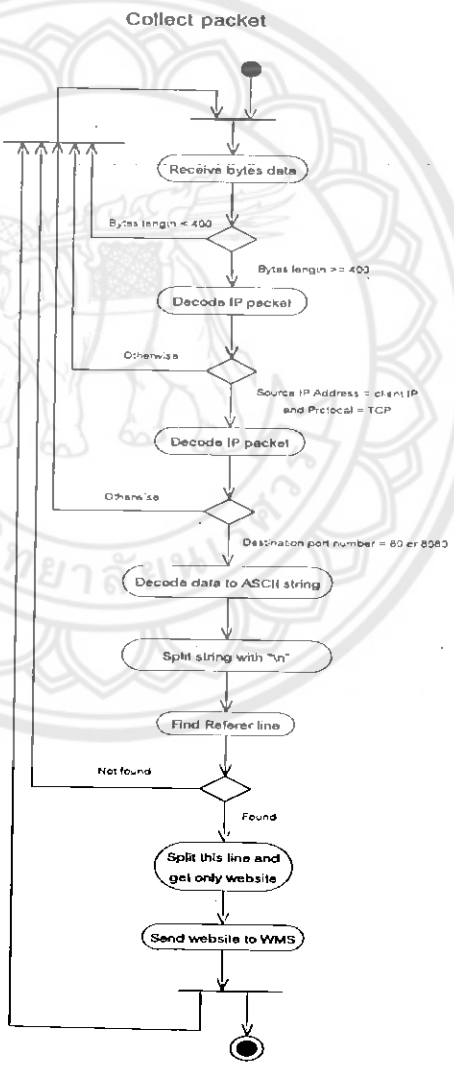
คำอธิบาย

หลักการทำงาน คือ โปรแกรมจะมีการแยกเธรด (Thread) ทำงาน เพื่อคอยตรวจสอบรายชื่อโปรเซสที่ทำงานอยู่ในขณะนั้นทุกๆ n นาที ตามที่สั่งไว้ โดยจะดึงรายชื่อโปรเซสที่ทำงานอยู่ในขณะนั้นมาทั้งหมด แล้วนำแต่ละชื่อโปรเซสนี้ไปตรวจสอบกับ List ก่อน ถ้าไม่มีชื่อโปรเซสอยู่ใน List ก็ให้เพิ่มชื่อโปรเซสนี้ลงไปใน List จากนั้นจึงไปตรวจสอบกับชื่อโปรเซสที่อยู่ใน Dictionary ซึ่งรายชื่อโปรเซสใน Dictionary นี้ ได้มาจากการที่โปรแกรม WMS สั่งให้ตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน แล้วโปรแกรมจะไปโหลดรายชื่อโปรเซสจากไฟล์มาเก็บไว้ใน Dictionary ถ้าชื่อโปรเซสมีอยู่ใน Dictionary ก็จะส่งข้อมูลการแจ้งเตือนไปยังโปรแกรม WMS การตรวจสอบกับ List ก่อนนั้น เป็นการป้องกันปัญหาการที่มีชื่อโปรเซสซ้ำกันหลายชื่อ เช่น โปรแกรม Chrome ที่จะ

ทำให้โปรแกรมส่งการแจ้งเตือนไปยังโปรแกรม WMS หลายครั้งโดยไม่จำเป็น เพราะถ้าชื่อโปรเซสที่นำมาตรวจสอบมีอยู่ใน List แล้ว แสดงว่าได้ส่งข้อมูลการแจ้งเตือนไปแล้ว

จากรูปที่ 3.36 เริ่มแรกจะทำการดึงข้อมูลเกี่ยวกับโปรเซสทุกตัวที่ทำงานอยู่ในขณะนั้น จากนั้นทำการตรวจสอบโปรเซสแต่ละตัวว่ามีชื่อตรงกับชื่อโปรเซสที่มีอยู่ใน Dictionary หรือไม่ ถ้าไม่ตรงก็จะทำการตรวจสอบไปเรื่อยๆ จนกว่าจะครบทุกโปรเซส แต่ถ้าตรงก็จะส่งข้อความแจ้งเตือนไปยังโปรแกรม WMS โดยทำการตรวจสอบจนกว่าจะครบทุกโปรเซส เมื่อครบแล้วก็จะทำงานในแบบ Asynchronous เพื่อตรวจสอบโปรเซสใหม่อีกครั้งทุกๆ n นาทีต่อไป

7. Activity Diagram ของการรับคำสั่งจากโปรแกรม WMS

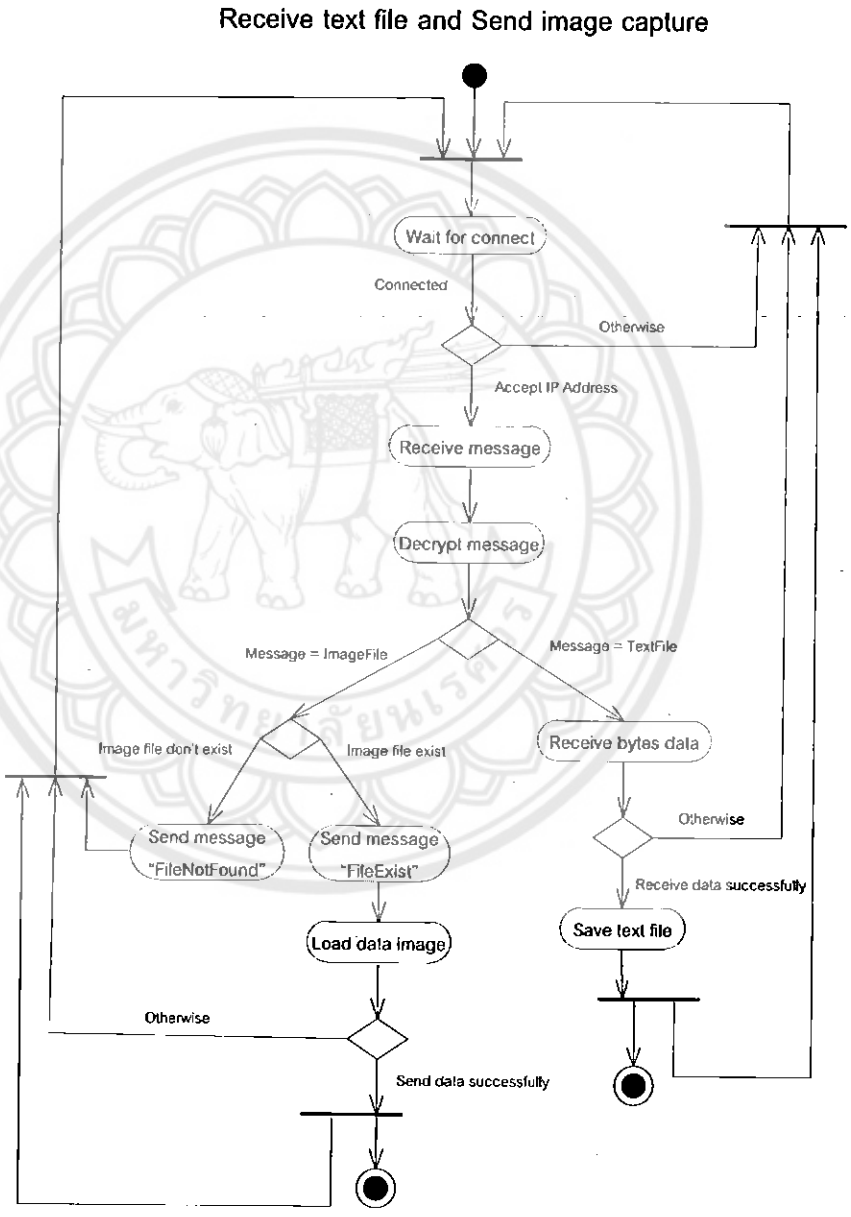


รูปที่ 3.37 Activity Diagram ของการตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์

<u>คำอธิบาย</u>

จากรูปที่ 3.37 มีขั้นตอนการตรวจจับเว็บไซต์เหมือนกับคำอธิบายของ Activity Diagram ของการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า แต่ต่างกันตรงที่เมื่อได้ชื่อเว็บไซต์มาแล้วจะทำการส่งข้อมูลไปยังโปรแกรม WMS เลย และกลับไปรอรับข้อมูลใหม่

8. Activity Diagram ของการรับไฟล์นามสกุล .txt และการส่งรูปภาพ



รูปที่ 3.38 Activity Diagram ของการรับไฟล์นามสกุล .txt และการส่งรูปภาพ

คำอธิบาย

จากรูปที่ 3.38 เมื่อเครื่องแม่ข่ายทำการเชื่อมต่อเข้ามา ก็จะตรวจสอบว่าหมายเลขไอพีต้นทางเป็นหมายเลขไอพีของเครื่องแม่ข่ายที่เก็บไว้จริงหรือไม่ ถ้าไม่จริงก็กลับไปรอรับการเชื่อมต่อใหม่ ถ้าจริงก็จะรับข้อความที่ส่งมาและทำการถอดรหัสข้อความ ถ้าข้อความที่ส่งมาเท่ากับ "TextFile" ก็จะรับข้อมูลที่ส่งมา ถ้าการรับข้อมูลสำเร็จก็จะบันทึกข้อมูลลงไฟล์ แล้วกลับไปรอรับการเชื่อมต่อใหม่ แต่ถ้าไม่สำเร็จก็จะกลับไปรอรับการเชื่อมต่อใหม่เช่นกัน ถ้าข้อความที่ส่งมาเท่ากับ "ImageFile" จะตรวจสอบว่าชื่อไฟล์ที่แนบมากับข้อความมีอยู่จริงหรือไม่ ถ้าไม่มีอยู่จริงก็จะส่งข้อความ "FileNotFound" ไปให้โปรแกรม WMS แล้วกลับไปรอรับการเชื่อมต่อใหม่ แต่ถ้ามีอยู่จริงก็จะส่งข้อความ "FileExist" ไปให้โปรแกรม WMS จากนั้นจะอ่านข้อมูลของรูปภาพแล้วส่งไปให้โปรแกรม WMS ถ้าการส่งข้อมูลไฟล์รูปภาพนี้สำเร็จหรือไม่สำเร็จก็ตาม ก็จะกลับไปรอรับการเชื่อมต่อใหม่เช่นกัน

3.5 การออกแบบฐานข้อมูล

ฐานข้อมูลมีไว้สำหรับบันทึกข้อมูลเกี่ยวกับการเข้าเว็บไซต์ทั้งหมด สำหรับนำมาวิเคราะห์ว่าเว็บไซต์ไหนบ้างที่ไม่ควรอนุญาตให้เข้าในระหว่างการเรียนการสอน

รูปที่ 3.39 การออกแบบตาราง (table) ในการเก็บข้อมูลการเข้าเว็บไซต์ต่างๆ

คำอธิบาย

ฟิลด์ number คือ ฟิลด์เก็บลำดับการบันทึก

ฟิลด์ computerName คือ ฟิลด์เก็บชื่อเครื่องลูกข่าย

ฟิลด์ website คือ ฟิลด์เก็บชื่อเว็บไซต์

ฟิลด์ date คือ ฟิลด์เก็บวัน เดือน ปีที่มีการเข้าเว็บไซต์

ฟิลด์ time คือ ฟิลด์เก็บเวลาที่มีการเข้าเว็บไซต์

บทที่ 4

การทดสอบและผลการทดสอบ

4.1 การทดสอบในระดับ System Test ของโปรแกรม WMS

System Test คือ การทดสอบโปรแกรมโดยรวมทั้งหมดว่าตรงกับความต้องการที่กำหนดไว้หรือไม่ โดยสามารถแบ่งประเภทได้ 2 ประเภท คือ

1. Functional คือ การทดสอบเกี่ยวกับหน้าที่การทำงานของโปรแกรม
2. Non – Functional คือ การทดสอบที่ไม่เกี่ยวกับการทำงานของโปรแกรม

ตารางที่ 4.1 ตารางแสดงการทดสอบในระดับ System Test ของโปรแกรม WMS

ประเภทการทดสอบ	ชื่อการทดสอบ
Functional	ค้นหาหมายเลขไอพีและชื่อเครื่องลูกข่ายทั้งหมดที่ทำงานอยู่ในเครือข่ายส่วนตัว (LAN) เดียวกัน
	ตรวจสอบสถานะของคอมพิวเตอร์ลูกข่าย
	ตรวจสอบสถานะของโปรแกรม WMC
	สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าใช้งาน
	สั่งให้โปรแกรม WMC หยุดตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าใช้งาน
	สั่งให้โปรแกรม WMC ตรวจจับโปรแกรมที่ไม่อนุญาตให้เข้าใช้งาน
	สั่งให้โปรแกรม WMC ตรวจจับโปรแกรมไม่อนุญาตให้เข้าใช้งาน
	สั่งให้โปรแกรม WMC หยุดตรวจจับโปรแกรมที่ไม่อนุญาตให้เข้าใช้งาน
	สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์ที่เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์
	ให้โปรแกรม WMC หยุดตรวจจับเว็บไซต์ที่เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์
	สั่งให้โปรแกรม WMC ส่งเทกซ์ไฟล์ (Text file) ไปยังเครื่องลูกข่าย
	สั่งให้โปรแกรม WMC ดาวน์โหลดไฟล์รูปภาพจากเครื่องลูกข่าย
	สรุปข้อมูลการเข้าเว็บไซต์ที่ได้บันทึกไว้ในฐานข้อมูล

ตารางที่ 4.1 (ต่อ) ตารางแสดงการทดสอบในระดับ System Test ของโปรแกรม WMS

ประเภทการทดสอบ	ชื่อการทดสอบ
Non - Functional	ทดสอบประสิทธิภาพการทำงานของโปรแกรม (Performance)
	ทดสอบความน่าเชื่อถือของโปรแกรม (Reliability)
	ทดสอบด้านความปลอดภัยของโปรแกรม (Security)

หมายเหตุ โปรแกรม WMS หมายถึง โปรแกรมที่ทำงานบนเครื่องแม่ข่าย โดย WMS ย่อมาจาก Website monitor server

โปรแกรม WMC หมายถึง โปรแกรมที่ทำงานบนเครื่องลูกข่าย โดย WMC ย่อมาจาก Website monitor client

คุณสมบัติของเครื่องลูกข่ายที่ใช้ในการทดสอบ

- ซีพียู: Intel Core 2 Duo E4500 2.19 GHz
- หน่วยความจำ: 1.97 GB
- หน่วยเก็บข้อมูลหลัก: Seagate 250GB ความเร็ว 7200 RPM
- ระบบปฏิบัติการ: Microsoft Windows XP SP3 X86
- ข้อมูลการเชื่อมต่อ: Ethernet ความเร็ว 100 Mbps โดยเชื่อมต่อกับ HUB

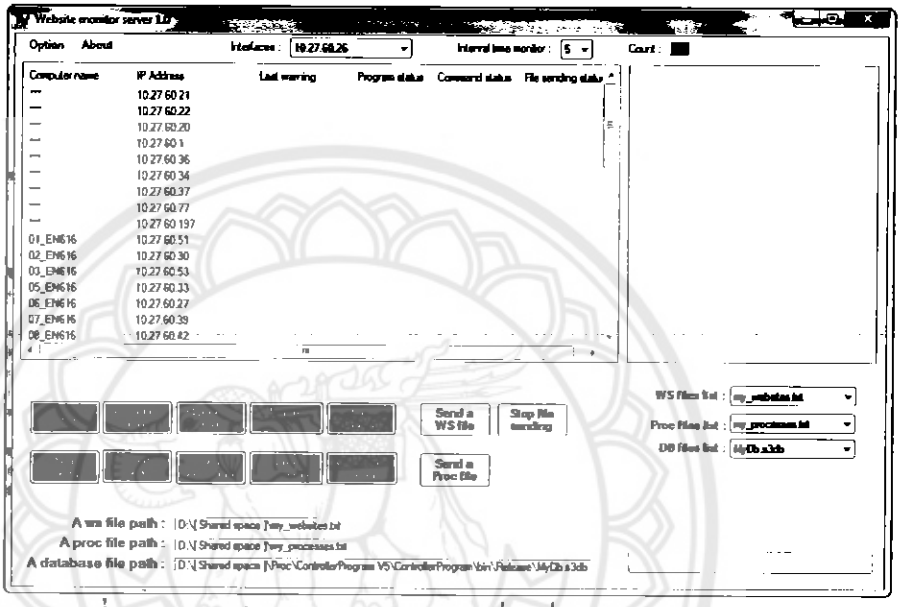
คุณสมบัติของเครื่องแม่ข่ายที่ใช้ในการทดสอบ

- ซีพียู: Intel Core i3 350 2.13 GHz
- หน่วยความจำ: 3.5 GB
- หน่วยเก็บข้อมูลหลัก: Western Digital 500GB ความเร็ว 7200 RPM
- ระบบปฏิบัติการ: Microsoft Windows 7 SP1 X86
- ข้อมูลการเชื่อมต่อ: Ethernet ความเร็ว 100 Mbps โดยเชื่อมต่อกับ HUB

4.1.1 ค้นหาหมายเลขไอพีและชื่อเครื่องลูกข่ายทั้งหมดที่ทำงานอยู่ในเครือข่ายส่วนตัว (LAN) เดียวกัน

วิธีการทดสอบ คือ คลิกที่ปุ่ม Scan แล้วรอการตอบกลับมาจากเครื่องลูกข่าย

กรณีที่ 1 ถ้าโปรแกรมค้นหาหมายเลขไอพีได้ แต่ไม่สามารถหาชื่อเครื่องของหมายเลขไอพีนั้นได้ จะแสดงเครื่องหมาย *** และหมายเลขไอพี ที่คอลัมน์ Computer name และ IP Address ตามลำดับ ดังรูปที่ 4.1



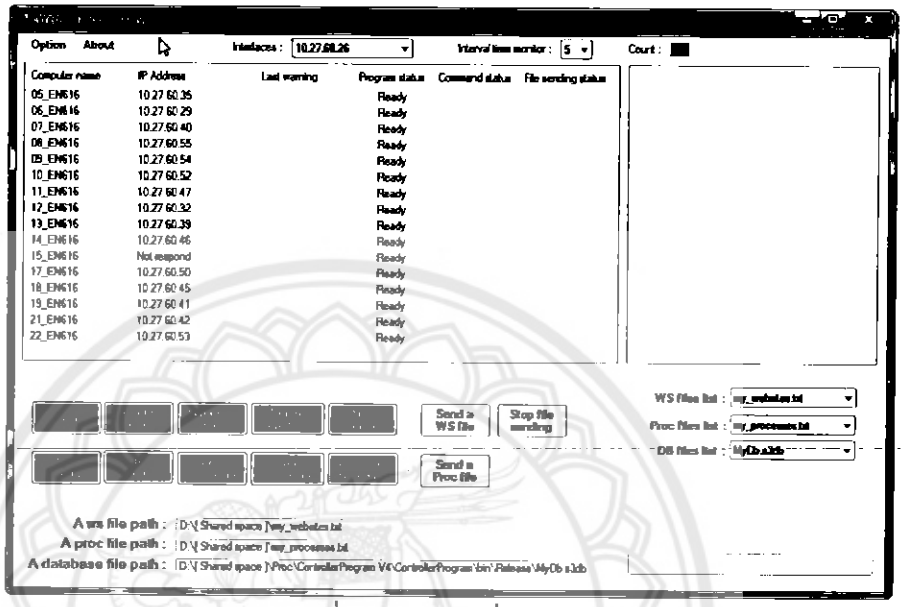
รูปที่ 4.1 ผลการค้นหาหมายเลขไอพีและชื่อเครื่องภายใน Subnet เดียวกัน

กรณีที่ 2 ถ้าโปรแกรมค้นหาหมายเลขไอพีได้ และสามารถหาชื่อเครื่องของหมายเลขไอพีนั้นได้ จะแสดงเครื่องชื่อเครื่อง และหมายเลขไอพีที่คอลัมน์ Computer name และ IP Address ตามลำดับ ดังรูปที่ 4.1

4.1.2 ตรวจสอบสถานะของคอมพิวเตอร์ลูกข่าย

วิธีการทดสอบ คือ คลิกที่ปุ่ม Computer status แล้วรอการตอบกลับมาจากเครื่องลูกข่าย

กรณีที่ 1 ถ้าไม่มีการตอบกลับมาจากหมายเลขไอพีใดๆ จะแสดงคำว่า Not respond ที่คอลัมน์ IP Address ดังรูปที่ 4.2



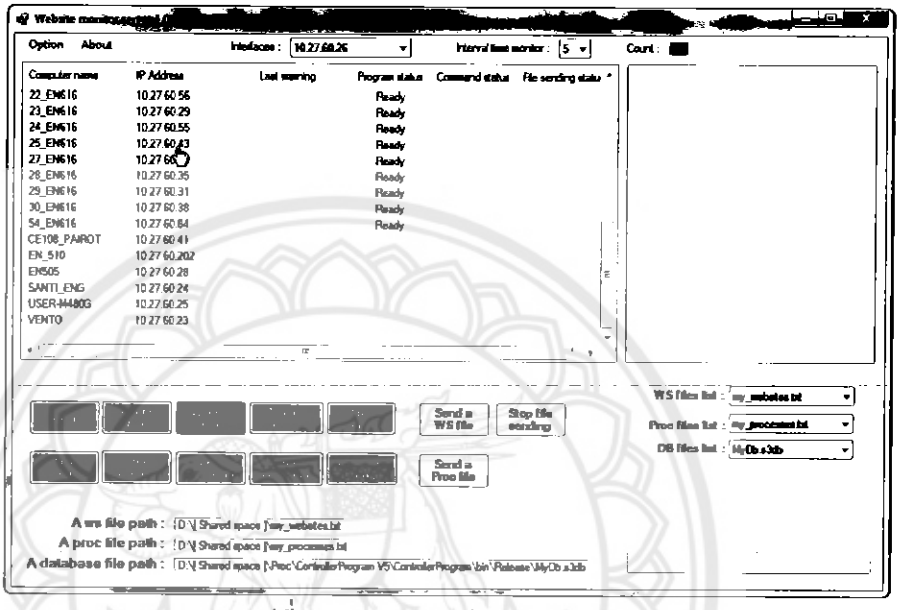
รูปที่ 4.2 สถานะเครื่องลูกข่าย

กรณีที่ 2 ถ้ามีการตอบกลับมาจากหมายเลขไอพีใดๆ จะแสดงหมายเลขไอพีนั้นๆ ที่คอลัมน์ IP Address ดังรูปที่ 4.2

4.1.3 ตรวจสอบสถานะของโปรแกรม WMC

วิธีการทดสอบ คือ คลิกที่ปุ่ม Program status แล้วรอการตอบกลับมาจากโปรแกรม WMC

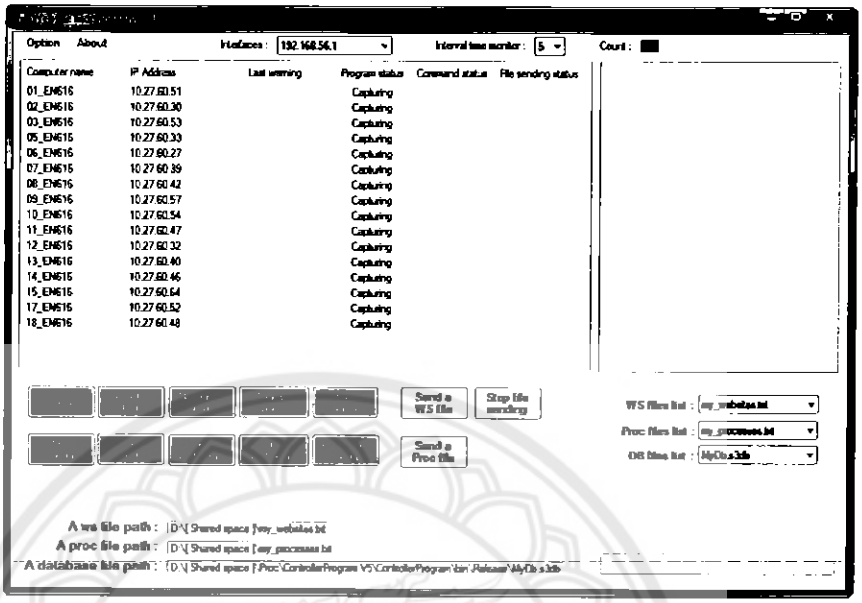
กรณีที่ 1 ถ้าเครื่องลูกข่ายใดๆ ไม่ทำงานหรือโปรแกรม WMC ไม่ทำงาน จะไม่แสดงข้อความใดๆ ที่คอลัมน์ Program status เลย ดังรูปที่ 4.3



รูปที่ 4.3 สถานะของโปรแกรม WMC

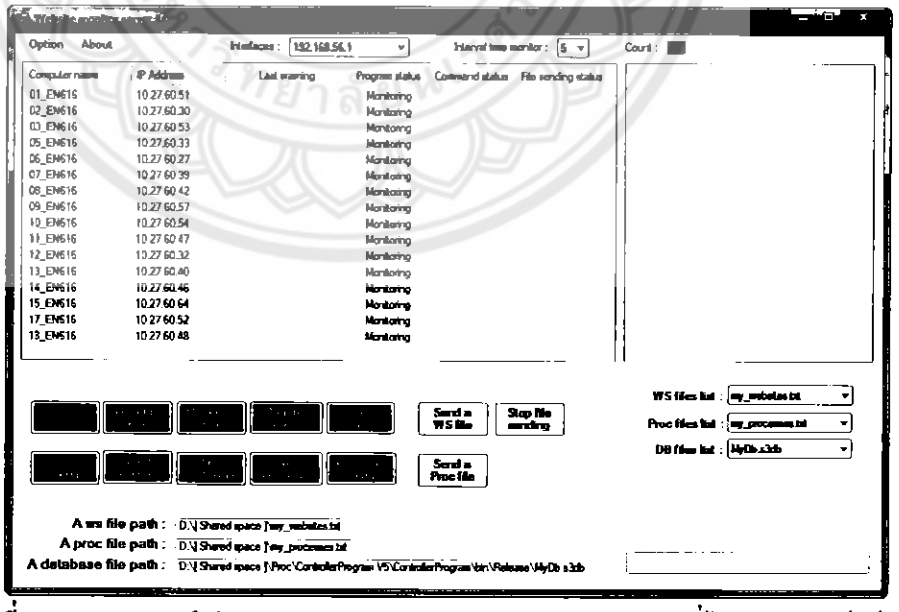
กรณีที่ 2 ถ้าโปรแกรม WMC ยังไม่ได้สั่งให้ทำงานอะไร จะแสดงคำว่า Ready ที่คอลัมน์ Program status ดังรูปที่ 4.3

กรณีที่ 3 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า จะแสดงคำว่า Capturing ที่คอลัมน์ Program status ดังรูปที่ 4.4



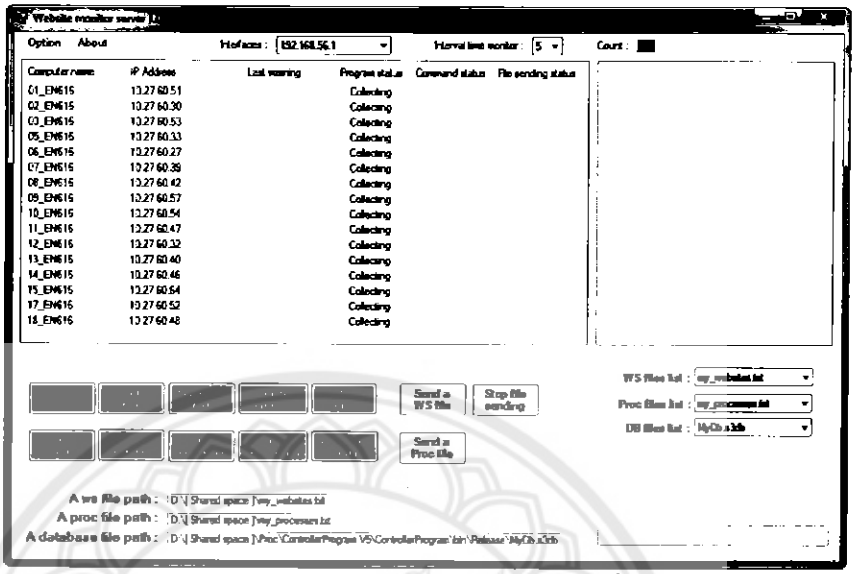
รูปที่ 4.4 แสดงสถานะของโปรแกรม WMC ขณะกำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า

กรณีที่ 4 ถ้าโปรแกรม WMC กำลังตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน จะแสดงคำว่า Monitoring ที่คอลัมน์ Program status ดังรูปที่ 4.5



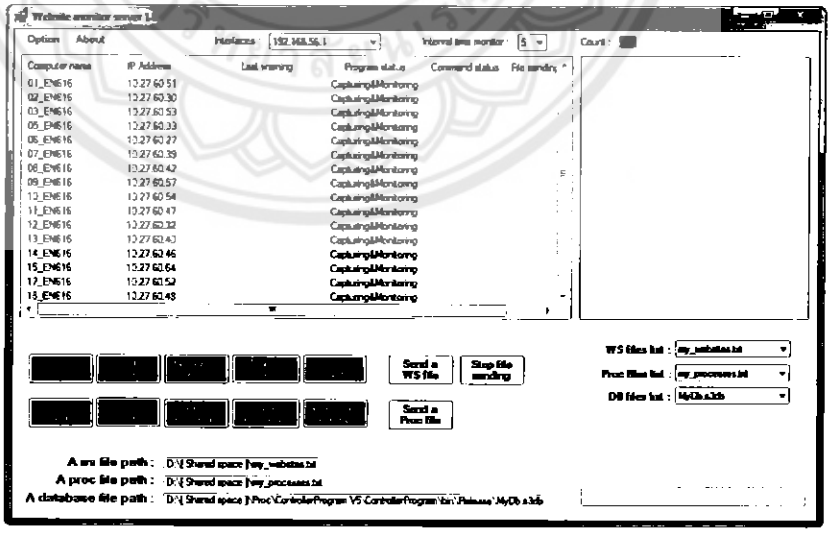
รูปที่ 4.5 สถานะของโปรแกรม WMC ขณะกำลังตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน

กรณีที่ 5 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์ จะแสดงคำว่า Collecting ที่คอลัมน์ Program status ดังรูปที่ 4.6



รูปที่ 4.6 สถานะของโปรแกรม WMC ขณะกำลังตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์

กรณีที่ 6 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งาน จะแสดงคำว่า Capturing&Monitoring ที่คอลัมน์ Program status ดังรูปที่ 4.7

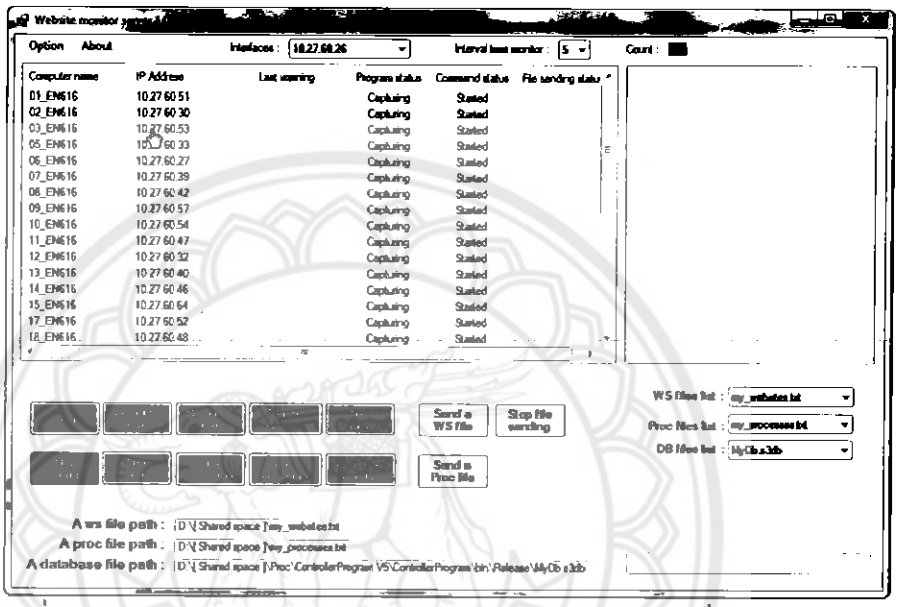


รูปที่ 4.7 สถานะของโปรแกรม WMC ขณะกำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งาน

4.1.4 สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าใช้งาน

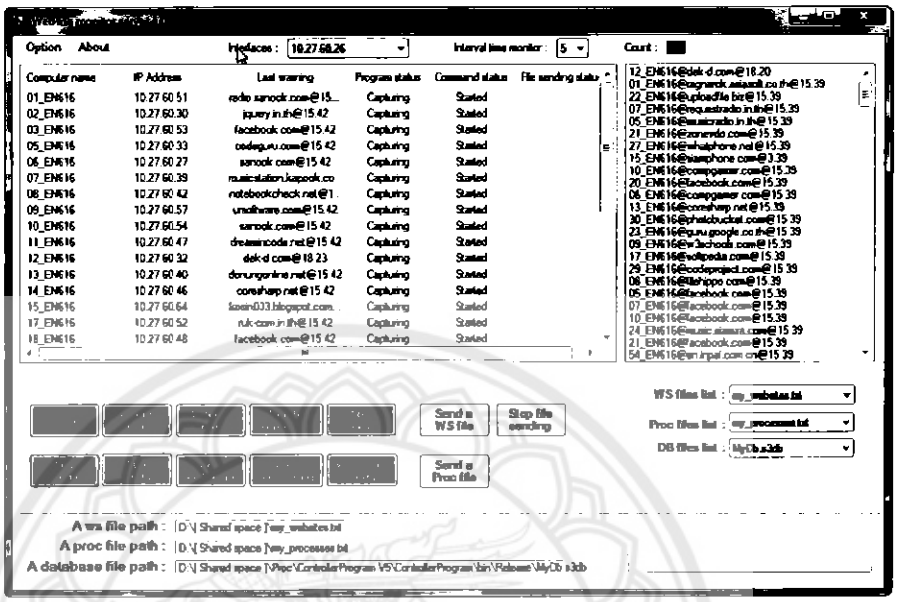
วิธีการทดสอบ คือ เลือกไฟล์ที่เก็บรายชื่อเว็บไซต์ที่ช่อง WS file list จากนั้นคลิกที่ปุ่ม Start capture แล้วรอการตอบกลับมาจากโปรแกรม WMC

กรณีที่ 1 ถ้าโปรแกรม WMC ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าใช้งานแล้ว จะแสดงคำว่า Capturing และ Started ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.8



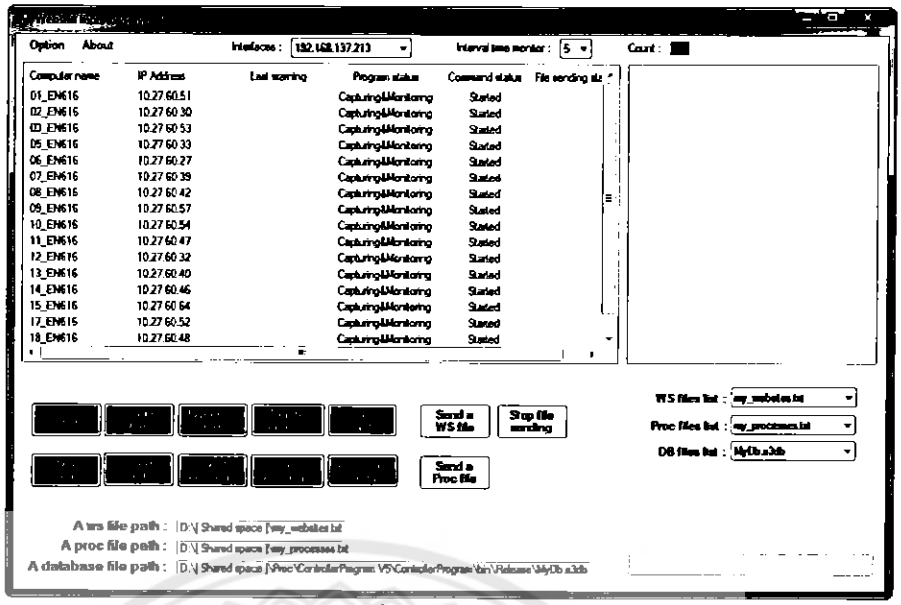
รูปที่ 4.8 สถานะของโปรแกรม WMC ขณะกำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า

กรณีที่ 2 เหมือนกับ กรณีที่ 1 แต่มีการแจ้งเตือนจากโปรแกรม WMC เพิ่มเข้ามา จะแสดงชื่อเครื่อง@ชื่อเว็บไซต์@เวลาที่เข้าเว็บ ที่ช่องด้านขวามือ และแสดง ชื่อเว็บไซต์@เวลาที่เข้าที่คอลัมน์ Last event ด้วย ดังรูปที่ 4.9



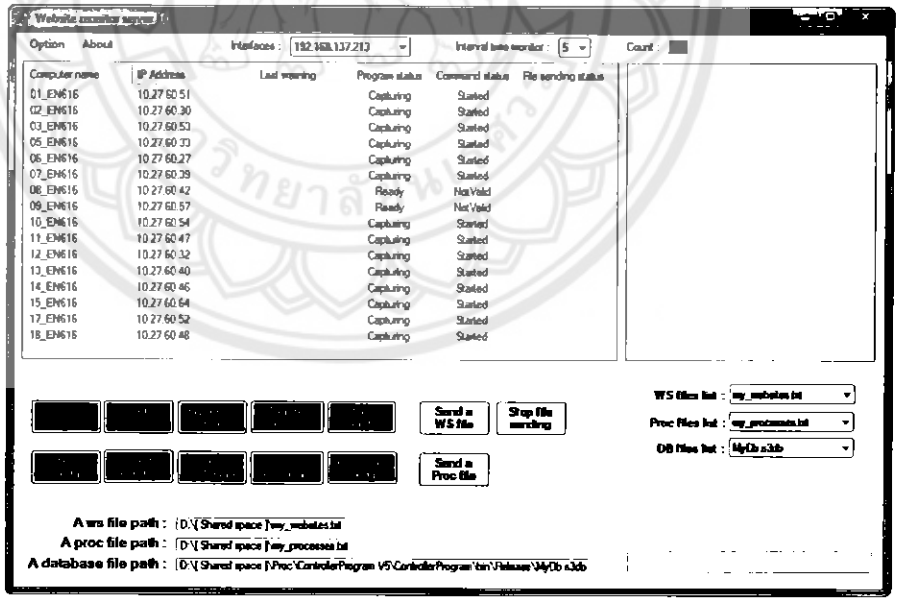
รูปที่ 4.9 การรับข้อมูลแจ้งเตือนการเข้าเว็บไซต์ที่ไม่อนุญาตให้เข้าจากเครื่องลูกข่ายหลายเครื่อง

กรณีที่ 3 ถ้าโปรแกรม WMC กำลังตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานอยู่ และเมื่อตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าแล้วจะแสดงคำว่า Capturing&Moniotoring และคำว่า Started ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.10



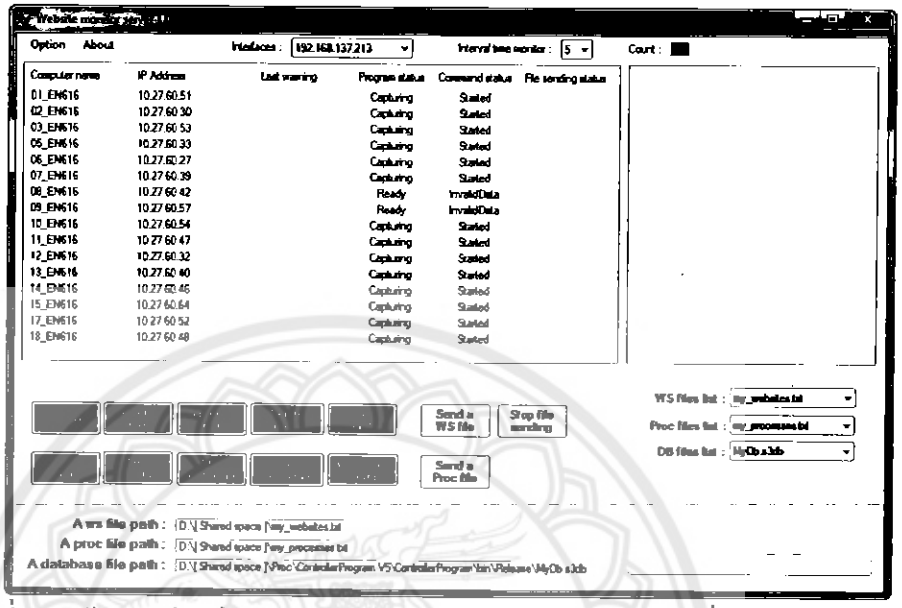
รูปที่ 4.10 ผลการแจ้งเตือนการเข้าเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งานจากเครื่องลูกข่ายหลายเครื่อง

กรณีที่ 4 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์อยู่ จะแสดงคำว่า NotValid ที่คอลัมน์ Command status ดังรูปที่ 4.11



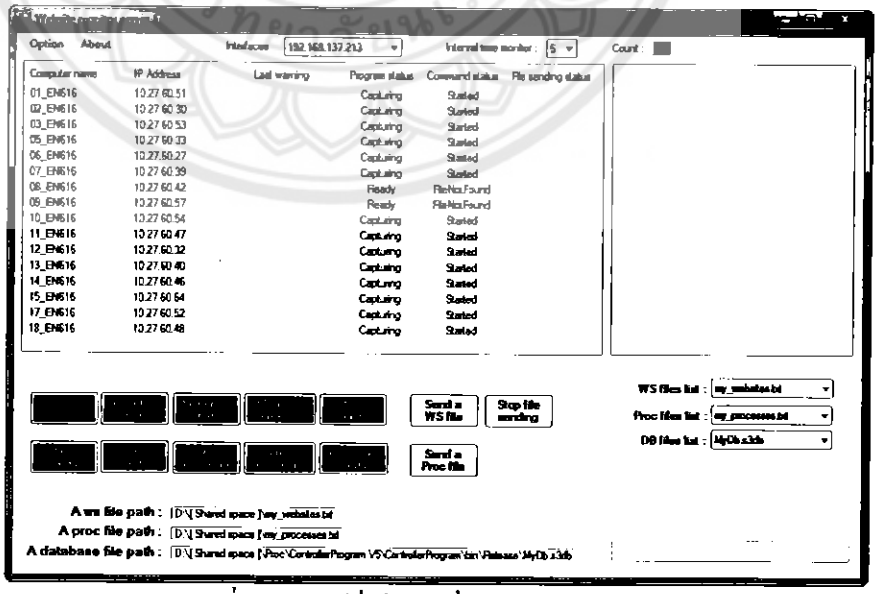
รูปที่ 4.11 แสดงผลการไม่อนุญาตให้ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าพร้อมกันการตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์

กรณีที่ 5 ถ้าโปรแกรม WMC ตรวจพบว่าข้อมูลในไฟล์ไม่สามารถนำมาใช้ในการตรวจสอบเว็บไซต์ที่ไม่อนุญาตให้เข้าได้ จะแสดงคำว่า InvalidData ที่คอลัมน์ Command status ดังรูปที่ 4.12



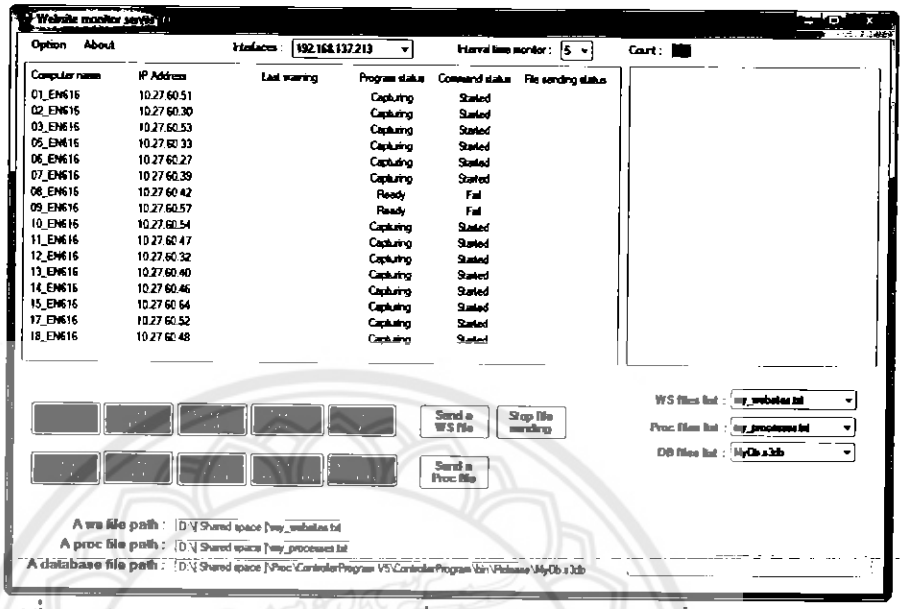
รูปที่ 4.12 ข้อมูลในไฟล์ไม่สามารถนำมาใช้ในการตรวจสอบเว็บไซต์ที่ไม่อนุญาตให้เข้าได้

กรณีที่ 6 ถ้าโปรแกรม WMC ตรวจพบว่าไฟล์ที่เก็บรายชื่อเว็บไซต์ไม่มีอยู่จริง จะแสดงคำว่าที่คอลัมน์ Command status ดังรูปที่ 4.13



รูปที่ 4.13 ไฟล์ที่เก็บรายชื่อเว็บไซต์ไม่มีอยู่จริง

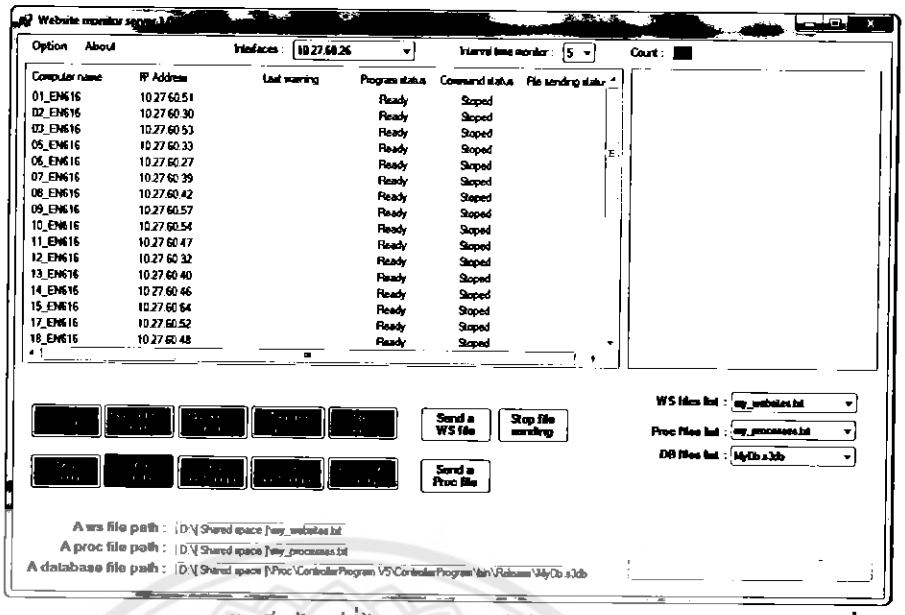
กรณีที่ 7 ถ้าโปรแกรม WMC ไม่สามารถเริ่มการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าได้ จะแสดงคำว่า Fail ที่คอลัมน์ Command status ดังรูปที่ 4.14



รูปที่ 4.14 โปรแกรม WMC ไม่สามารถเริ่มการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าได้

4.1.5 สั่งให้โปรแกรม WMC หยุดตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า

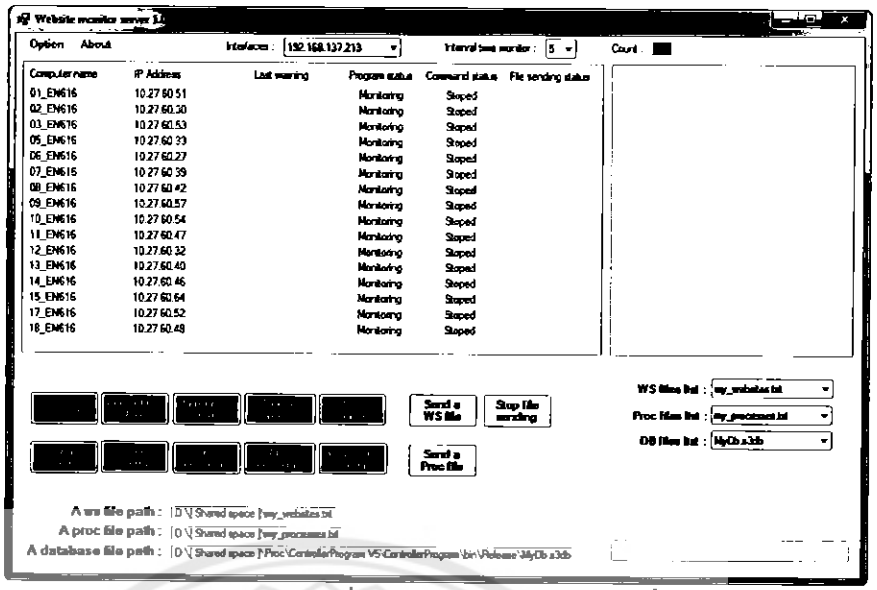
วิธีการทดสอบ คือ คลิกที่ปุ่ม Stop capture แล้วรอการตอบกลับมาจากโปรแกรม WMC กรณีที่ 1 ถ้าโปรแกรม WMC หยุดตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า จะแสดงคำว่า Ready และ Stoped ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.15



รูปที่ 4.15 การหยุดการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือหยุดตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานหรือหยุดตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์

กรณีที่ 2 ถ้าโปรแกรม WMC ไม่ได้ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าอยู่ จะแสดงคำ Stoped ที่คอลัมน์ Command status ดังรูปที่ 4.14

กรณีที่ 3 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งานอยู่ จะแสดงคำ Monitoring และ Stoped ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.16

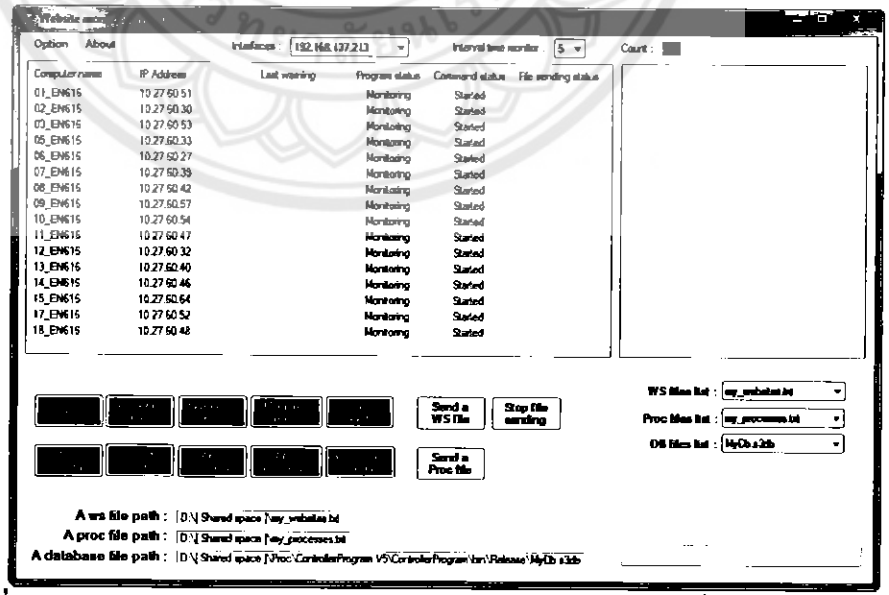


รูปที่ 4.16 สถานะของโปรแกรม WMC เมื่อหยุดการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าขณะที่การตรวจจับโปรแกรมกำลังทำงานอยู่

4.1.6 สั่งให้โปรแกรม WMC ตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน

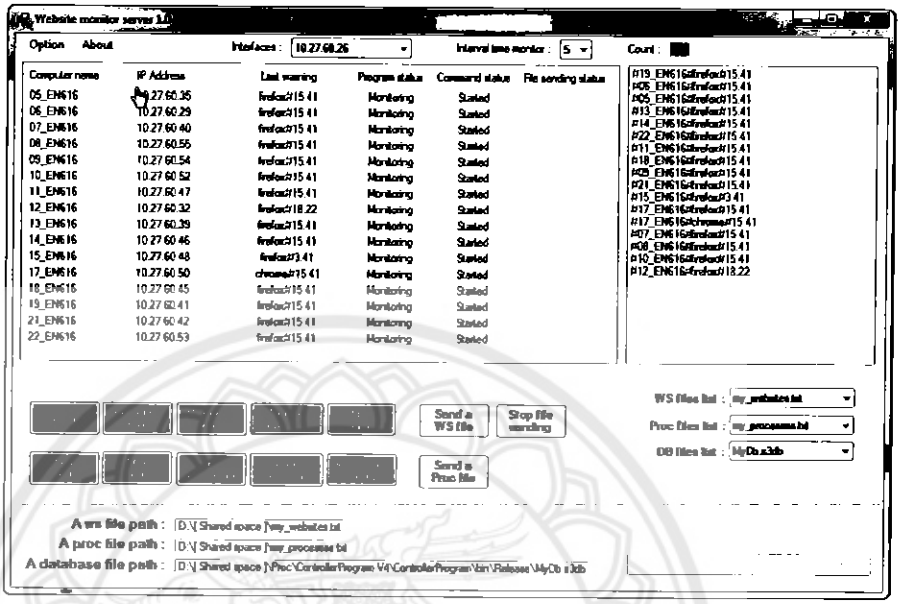
วิธีการทดสอบ คือ เลือกไฟล์ที่เก็บรายชื่อเว็บไซต์ที่ช่อง Proc file list เลือกจำนวนนาที จากนั้นคลิกที่ปุ่ม Monitor process แล้วรอการตอบกลับมาจากโปรแกรม WMC

กรณีที่ 1 โปรแกรม WMC ตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานแล้ว จะแสดงคำว่า Monitoring และ Started ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.17



รูปที่ 4.17 สถานะของโปรแกรม WMC ขณะกำลังตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน

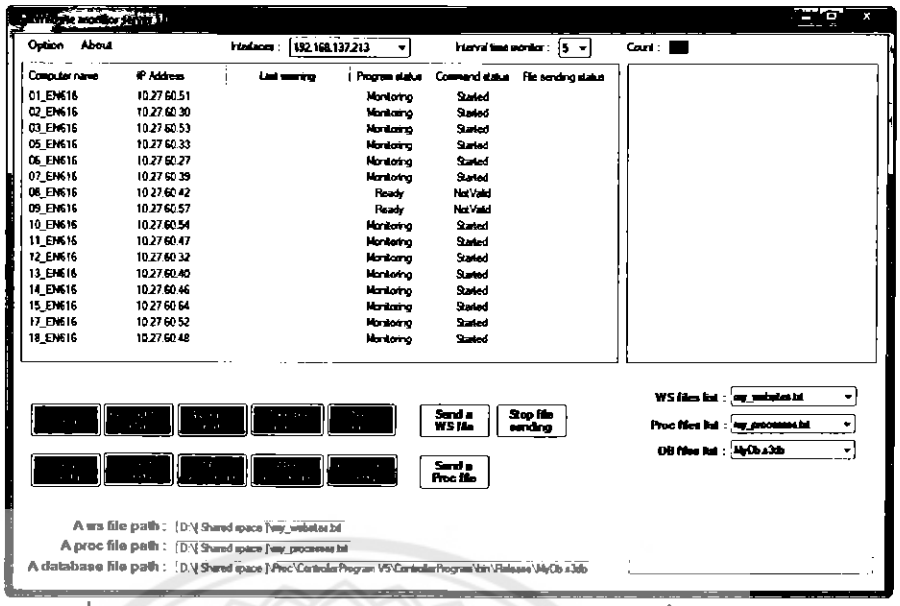
กรณีที่ 2 เหมือนกับ กรณีที่ 1 แต่มีการแจ้งเตือนจากโปรแกรม WMC เพิ่มเข้ามา จะแสดง # ชื่อเครื่อง #ชื่อโปรแกรม # เวลาที่ใช้งาน ที่ช่องด้านขวามือ และแสดง ชื่อโปรแกรม@เวลาที่ใช้งาน ที่คอลัมน์ Last event ด้วย ดังรูปที่ 4.18



รูปที่ 4.18 สถานะของโปรแกรม WMC ขณะรับการแจ้งเตือนการตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน

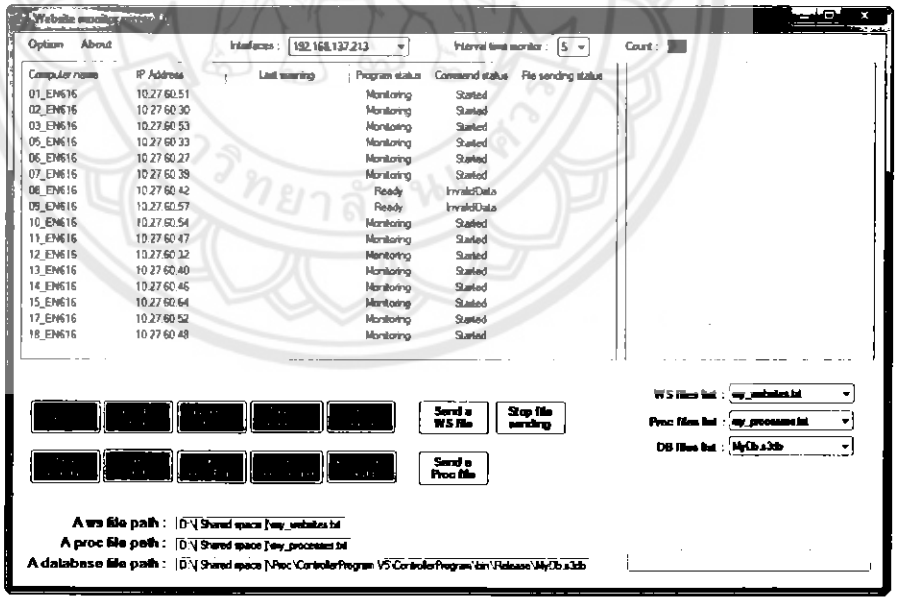
กรณีที่ 3 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าอยู่ และตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานแล้วจะแสดงคำว่า Capturing&Moniotoring และคำว่า Started ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.10

กรณีที่ 4 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์อยู่ จะแสดงคำว่า NotValid ที่คอลัมน์ Command status ดังรูปที่ 4.19



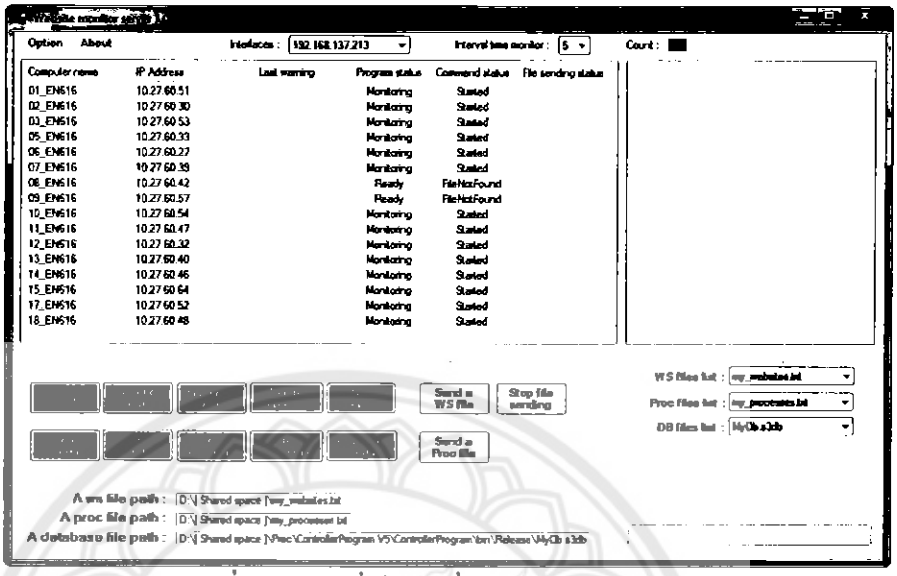
รูปที่ 4.19 โปรแกรม WMS ไม่ยอมให้ตรวจจับโปรแกรมที่ไม่อนุญาตให้เข้า

กรณีที่ 5 ถ้าโปรแกรม WMC ตรวจพบว่าข้อมูลในไฟล์ไม่สามารถนำมาใช้ในการตรวจสอบโปรแกรมที่ไม่อนุญาตให้ใช้งานได้ จะแสดงคำว่า InvalidData **ที่คอลัมน์** Command status ดังรูปที่ 4.20



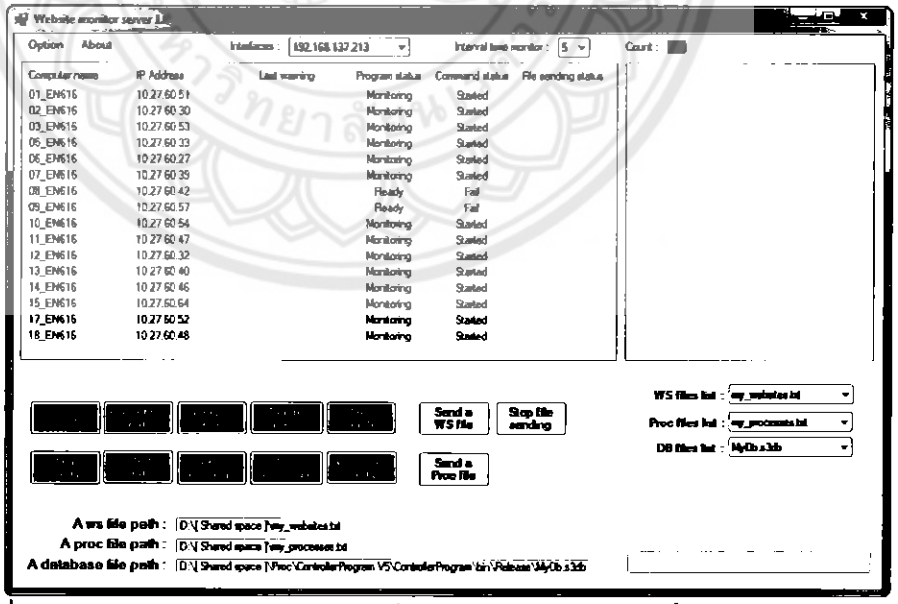
รูปที่ 4.20 ข้อมูลในไฟล์ไม่สามารถนำมาใช้ในการตรวจสอบโปรแกรมที่ไม่อนุญาตให้ใช้งานได้

กรณีที่ 6 ถ้าโปรแกรม WMC ตรวจพบว่าไฟล์ที่เก็บรายชื่อโปรแกรมไม่มีอยู่จริง จะแสดงคำว่า FileNotFound ที่คอลัมน์ Command status ดังรูปที่ 4.21



รูปที่ 4.21 ไฟล์ที่เก็บรายชื่อโปรแกรมไม่มีอยู่จริง

กรณีที่ 7 ถ้าโปรแกรม WMC ไม่สามารถเริ่มการตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานได้ จะแสดงคำว่า Fail ที่คอลัมน์ Command status ดังรูปที่ 4.22



รูปที่ 4.22 โปรแกรม WMC ไม่สามารถเริ่มการตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานได้

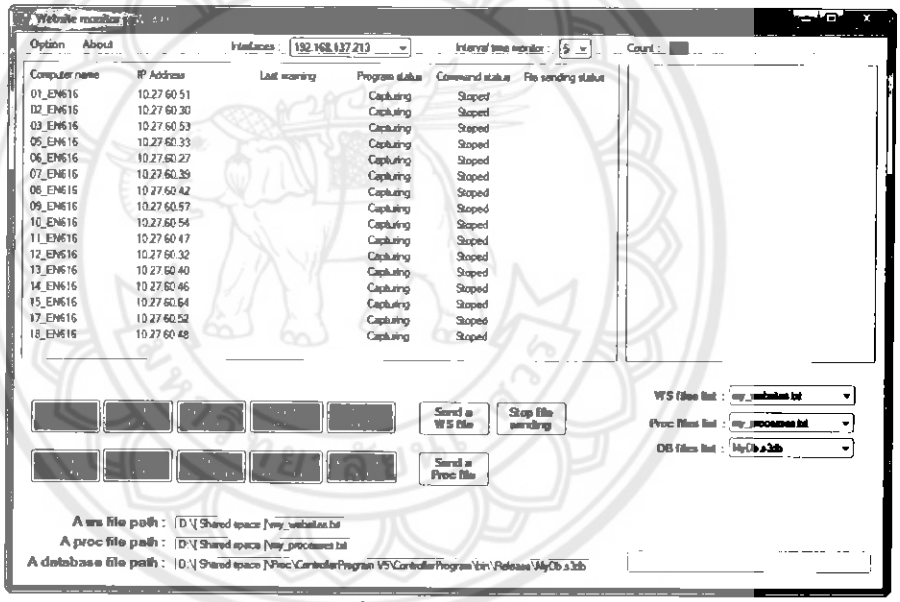
4.1.7 สั่งให้โปรแกรม WMC หยุดตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน

วิธีการทดสอบ คือ คลิกที่ปุ่ม Stop Monitor แล้วรอการตอบกลับมาจากโปรแกรม WMC

กรณีที่ 1 ถ้าโปรแกรม WMC หยุดตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานแล้ว จะแสดงคำว่า Ready และ Stoped ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.14

กรณีที่ 2 ถ้าโปรแกรม WMC ไม่ได้ตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานอยู่ จะแสดงคำ Stoped ที่คอลัมน์ Command status ดังรูปที่ 4.14

กรณีที่ 3 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งานอยู่ จะแสดงคำ Capturing และ Stoped ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.23

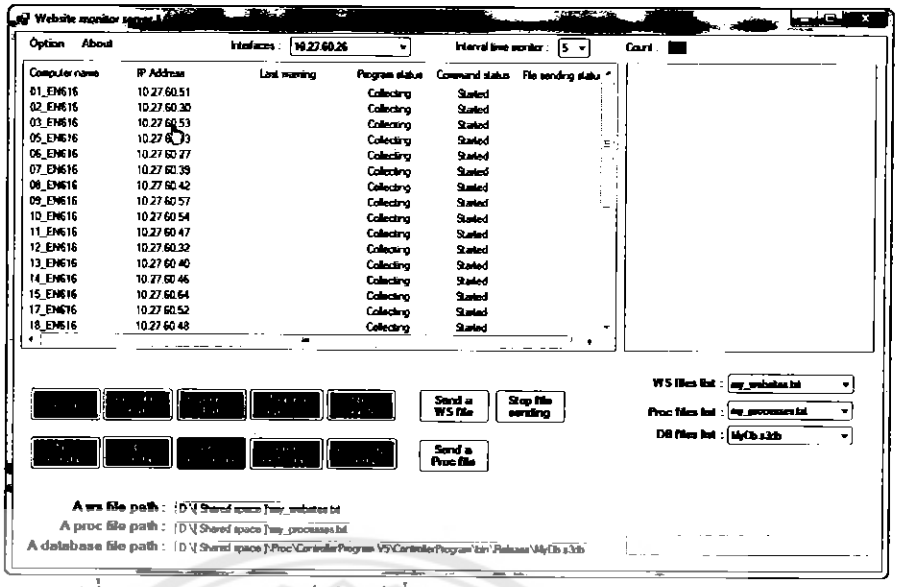


รูปที่ 4.23 สถานะของโปรแกรม WMC เมื่อหยุดการตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งาน ขณะที่การตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้ากำลังทำงานอยู่

4.1.8 สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์

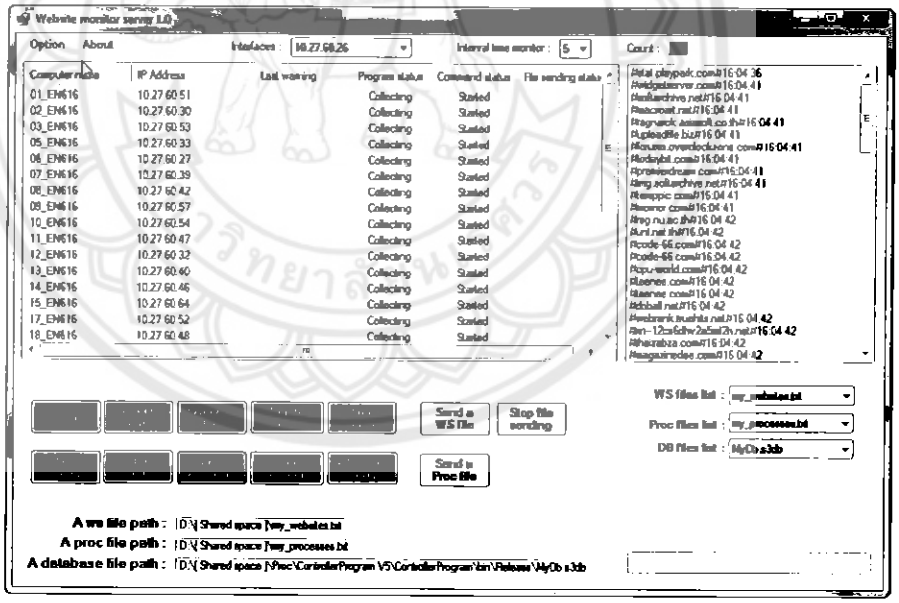
วิธีการทดสอบ คือ คลิกปุ่ม Start collecting แล้วรอการตอบกลับมาจากโปรแกรม WMC

กรณีที่ 1 โปรแกรม WMC ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์แล้ว จะแสดงคำว่า Collecting และ Started ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.24



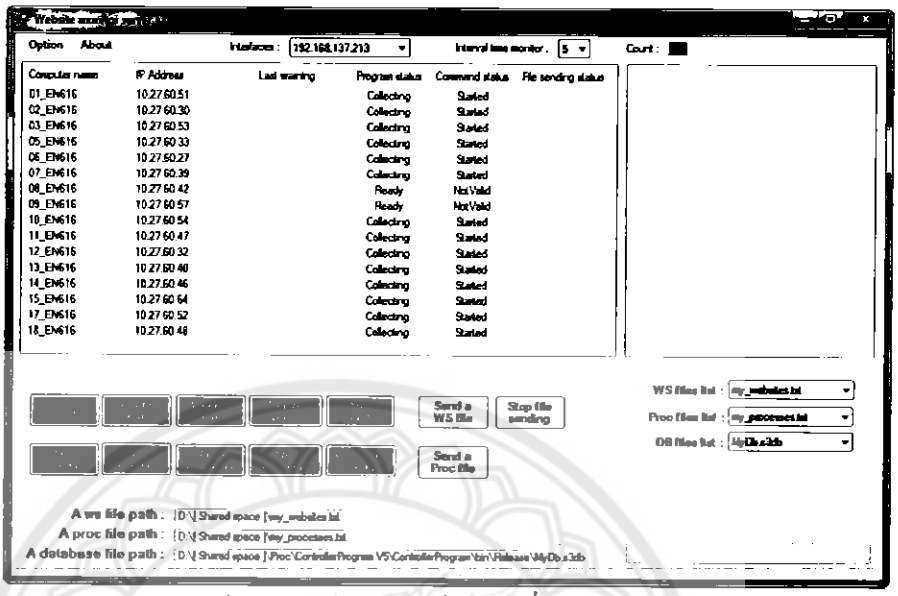
รูปที่ 4.24 การตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์

กรณีที่ 2 เหมือนกับ กรณีที่ 1 แต่มีการแจ้งเตือนจากโปรแกรม WMC เพิ่มเข้ามา จะแสดง #ชื่อเว็บไซต์# เวลาที่เข้า ที่ช่องด้านขวามือ ดังรูปที่ 4.25



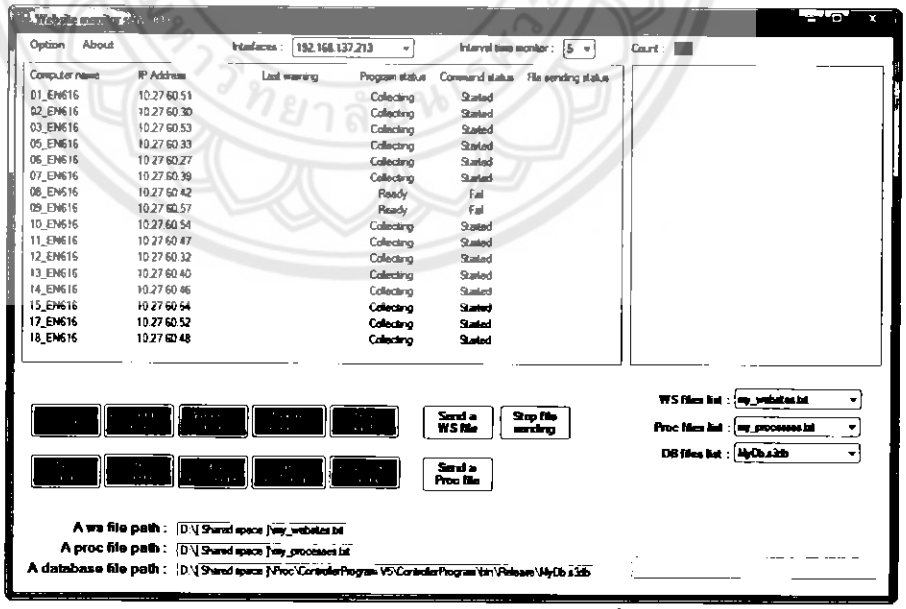
รูปที่ 4.25 การรับการแจ้งเตือนการตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมา วิเคราะห์จากเครื่องลูกข่ายหลายเครื่อง

กรณีที่ 3 ถ้าโปรแกรม WMC กำลังตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าหรือโปรแกรมที่ไม่อนุญาตให้ใช้งานอยู่ จะแสดงคำว่า NotValid ที่คอลัมน์ Command status ดังรูปที่ 4.26



รูปที่ 4.26 โปรแกรม WMS ไม่ยอมให้ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์

กรณีที่ 4 ถ้าโปรแกรม WMC ไม่สามารถเริ่มการตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์ได้ จะแสดงคำว่า Fail ที่คอลัมน์ Command status ดังรูปที่ 4.27



รูปที่ 4.27 โปรแกรม WMS ไม่สามารถเริ่มการตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์ได้

4.1.9 ให้โปรแกรม WMC หยุดตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์

วิธีการทดสอบ คือ คลิกที่ปุ่ม Stop Collecting แล้วรอการตอบกลับมาจากโปรแกรม WMC

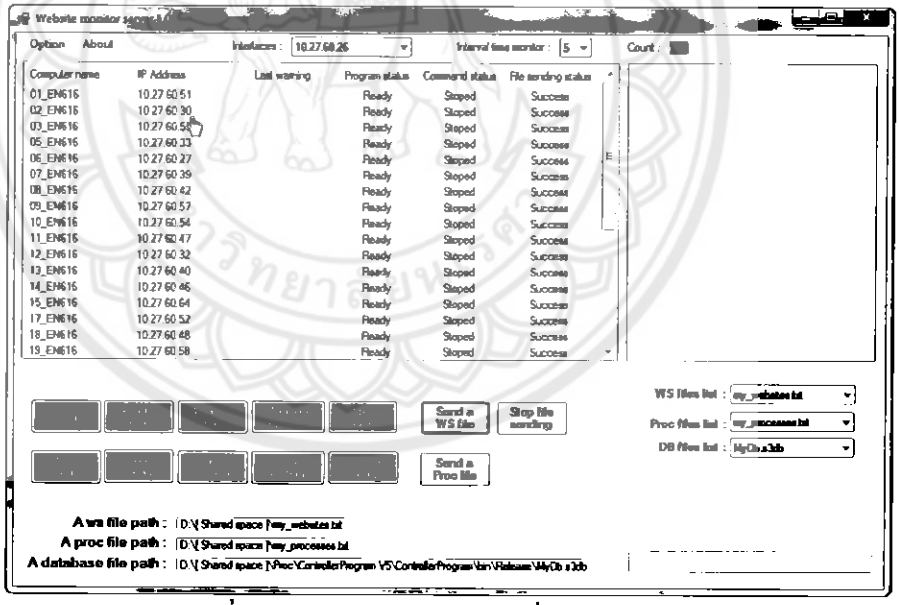
กรณีที่ 1 ถ้าโปรแกรม WMC หยุดตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์แล้ว จะแสดงคำว่า Ready และ Stoped ที่คอลัมน์ Program status และ Command status ตามลำดับ ดังรูปที่ 4.14

กรณีที่ 2 ถ้าโปรแกรม WMC ไม่ได้ตรวจจับโปรแกรมที่ไม่อนุญาตให้ใช้งานอยู่ จะแสดงคำ Stoped ที่คอลัมน์ Command status ดังรูปที่ 4.14

4.1.10 สั่งให้โปรแกรม WMC ส่งเทกซ์ไฟล์ (Text file) ไปยังเครื่องลูกข่าย

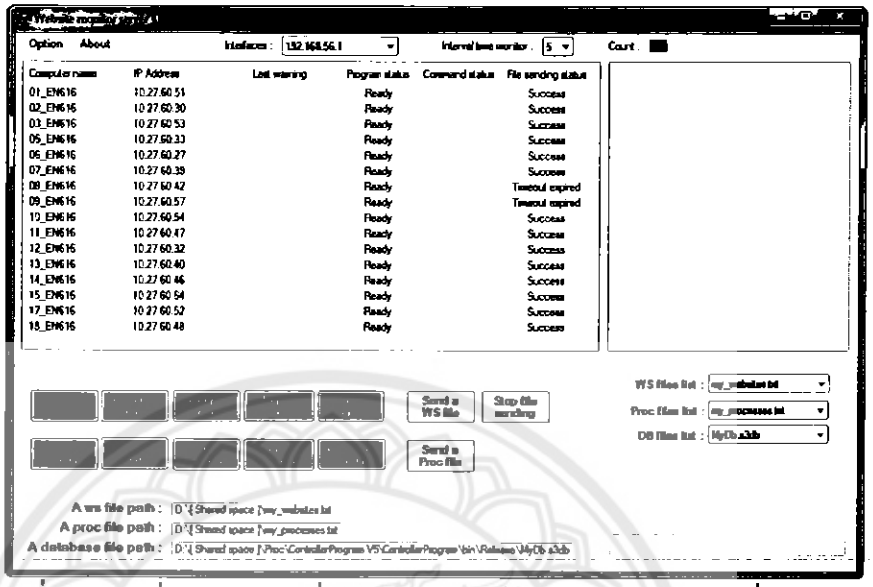
วิธีการทดสอบ คือ เลือกไฟล์ที่จะส่ง จากนั้นคลิกที่ปุ่ม Send a WS file แล้วรอผลจากการส่งไฟล์

กรณีที่ 1 ถ้าโปรแกรมส่งไฟล์ไปยังเครื่องลูกข่ายเครื่องนั้นสำเร็จ จะแสดงคำว่า Success ที่คอลัมน์ File sending status ดังรูปที่ 4.28



รูปที่ 4.28 ผลการส่งไฟล์ไปยังเครื่องลูกข่ายสำเร็จ

กรณีที่ 2 ถ้าโปรแกรมยังไม่เชื่อมต่อกับเครื่องลูกข่ายใดๆ นานเกินกว่า 3 วินาที จะแสดงคำว่า Timeout expired ที่คอลัมน์ File sending status ดังรูปที่ 4.29

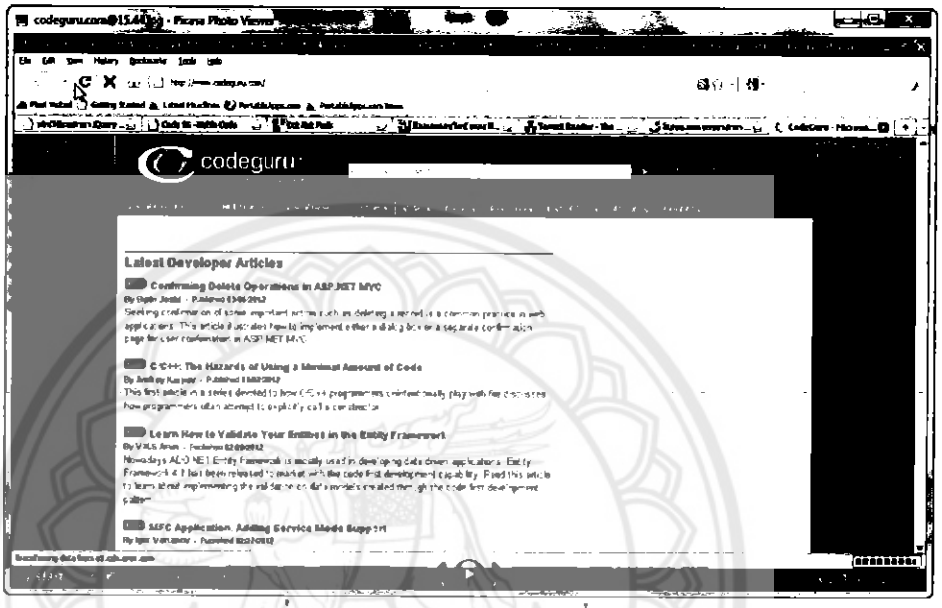


รูปที่ 4.29การเชื่อมต่อไปยังเครื่องลูกข่ายไม่สำเร็จ เพราะหมดเวลาตามที่กำหนดไว้

4.1.11 สั่งให้โปรแกรม WMC ดาวน์โหลดไฟล์รูปภาพจากเครื่องลูกข่าย

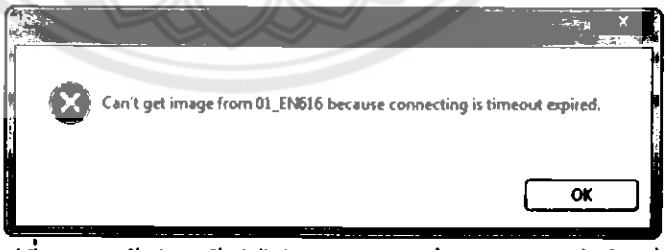
วิธีการทดสอบ คือ เลือกรายการที่จะดาวน์โหลดที่ช่องด้านขวา โดยต้องไม่ขึ้นต้นด้วยเครื่องหมาย # จากนั้นคลิกที่ปุ่ม Get image capture แล้วรอดูรูป

กรณีที่ 1 ถ้าโปรแกรมดาวน์โหลดรูปภาพสำเร็จ จะแสดงรูปภาพที่ได้ดาวน์โหลดมา ดังรูปที่ 4.30



รูปที่ 4.30 รูปภาพหน้าจอของเครื่องลูกข่าย

กรณีที่ 2 ถ้าโปรแกรมยังไม่เชื่อมต่อกับเครื่องลูกข่ายใดๆ นานเกินกว่า 3 วินาที จะแสดงหน้าต่างแจ้งว่าเชื่อมต่อไม่สำเร็จเพราะหมดเวลา ดังรูปที่ 4.31

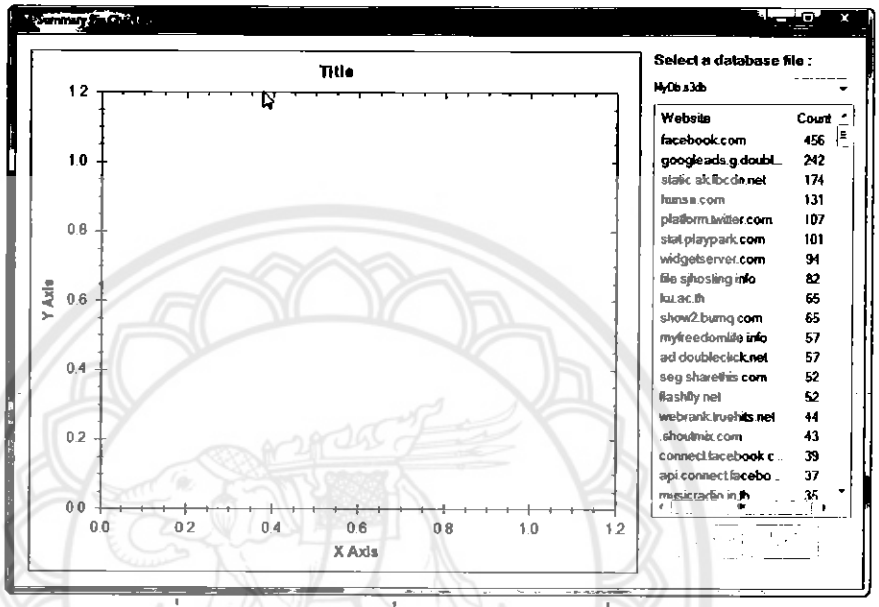


รูปที่ 4.31 หน้าต่างแจ้งว่าไม่สามารถดาวน์โหลดรูปภาพได้สำเร็จ

4.1.12 สรุปข้อมูลการเข้าเว็บไซต์ที่ได้บันทึกไว้ในฐานข้อมูล

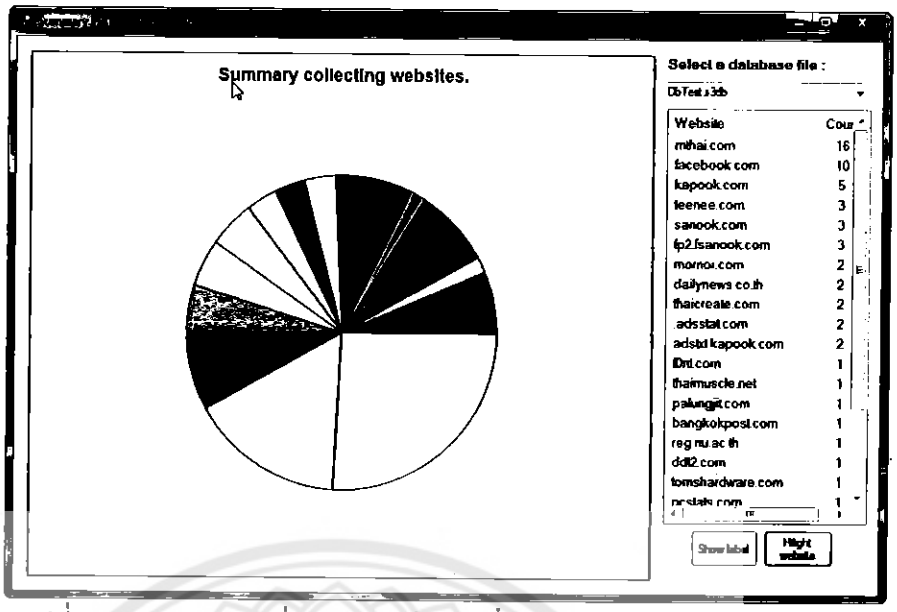
วิธีการทดสอบ คือ คลิกขวาที่ไฟล์ฐานข้อมูล จากนั้นเลือก Summarize collecting จากนั้นเลือกไฟล์ฐานข้อมูลที่ต้องการสรุป

กรณีที่ 1 ถ้าฐานข้อมูลมีเว็บไซต์ที่บันทึกไว้มากกว่า 148 เว็บไซต์ จะไม่สามารถแสดง Pie chart ได้ เนื่องจากว่าจำนวนสีไม่พอใช้ แต่จะแสดงความถี่ของแต่ละเว็บไซต์ ดังรูปที่ 4.32



รูปที่ 4.32 ผลสรุปความถี่ของแต่ละเว็บไซต์ที่ได้บันทึกไว้

กรณีที่ 2 ถ้าฐานข้อมูลมีเว็บไซต์ที่บันทึกไว้น้อยกว่าหรือเท่ากับ 148 เว็บไซต์ จะแสดง Pie chart และแสดงความถี่ของแต่ละเว็บไซต์ ดังรูปที่ 4.33



รูปที่ 4.33 ผลสรุปความถี่ของแต่ละเว็บไซต์ที่ได้บันทึกไว้พร้อมกราฟรูปวงกลม

4.1.13 ทดสอบประสิทธิภาพการทำงานของโปรแกรม (Performance)

เป็นการทดสอบว่าโปรแกรมสามารถรองรับการส่งข้อมูลจากเครื่องลูกข่ายในเวลาใกล้เคียงกันได้กี่เครื่อง และมีการแสดงข้อมูลการแจ้งเตือนได้ครบทุกเครื่องหรือไม่

วิธีการทดสอบ คือ ทดสอบกับเครื่องลูกข่ายทั้งหมด 35 เครื่อง โดยเครื่องลูกข่ายแต่ละเครื่องมีการเข้าเว็บไซต์แบบสุ่มจำนวน 115 เว็บไซต์ ทุกๆ 30 วินาที

ตารางที่ 4.2 ตารางแสดงผลการทดสอบประสิทธิภาพการทำงานของโปรแกรม WMS

การทำงาน	จำนวนเครื่องที่ทดสอบ	ทดสอบ 1 ชั่วโมง	ทดสอบ 2 ชั่วโมง
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า	35	ผ่าน	ผ่าน
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งาน	35	ผ่าน	ผ่าน
ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์	35	ผ่าน	ผ่าน

4.1.14 ทดสอบความน่าเชื่อถือของโปรแกรม (Reliability)

เป็นทดสอบว่าโปรแกรมมีการใช้ซีพียูและหน่วยความจำเป็นปริมาณที่มากน้อยแค่ไหน อยู่ในระดับที่ยอมรับได้หรือไม่ รวมถึงโปรแกรมสามารถทำงานได้อย่างยาวนาน โดยที่ไม่ทำให้เครื่องคอมพิวเตอร์ล่ม หรือเกิดหน่วยความจำรั่ว (Memory leak) หรือไม่

วิธีการทดสอบ คือ ทดสอบกับเครื่องลูกข่ายทั้งหมด 35 เครื่อง โดยเครื่องลูกข่ายแต่ละเครื่องมีการเข้าเว็บไซต์แบบสุ่มจำนวน 115 เว็บไซต์ ทุกๆ 30 วินาที โดยสังเกตการการใช้ซีพียูและหน่วยความจำของทุกๆ เครื่อง และจะเอาปริมาณการใช้ซีพียูและหน่วยความจำที่สูงที่สุดจากปริมาณการใช้ซีพียูและหน่วยความจำของเครื่องทั้งหมดมาใช้ในการสรุป

ตารางที่ 4.3 ตารางแสดงผลการใช้หน่วยความจำของโปรแกรม WMS

การทำงาน	จำนวนเครื่องที่ทดสอบ	ปริมาณการใช้หน่วยความจำ	
		ขณะไม่ได้ทำงาน (MB)	ขณะรับการแจ้งเตือน (MB)
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า	35	ไม่เกิน 15 MB	ไม่เกิน 20 MB
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งานพร้อมกัน	35	ไม่เกิน 15 MB	ไม่เกิน 20 MB
ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์	35	ไม่เกิน 15 MB	ไม่เกิน 20 MB

ตารางที่ 4.4 ตารางแสดงผลการใช้ซีพียูของโปรแกรม WMS

การทำงาน	จำนวนเครื่องที่ทดสอบ	ปริมาณการใช้ซีพียู	
		ขณะไม่ได้ทำงาน (%)	ขณะรับการแจ้งเตือน (%)
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า	35	0 %	ไม่เกิน 10 %
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งานพร้อมกัน	35	0 %	ไม่เกิน 10 %
ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์	35	0 %	ไม่เกิน 10 %

4.1.15 ทดสอบด้านความปลอดภัยของโปรแกรม (Security)

4.1.15.1 ทดสอบการดักจับข้อมูลระหว่างการส่งข้อมูลไปยังเครื่องลูกข่ายด้วยโปรแกรม **Wireshark**

วิธีการทดสอบ: ส่งข้อความที่มีการเข้ารหัสไปยังโปรแกรม WMC คือ T9v2Szx1ZUVR/tce6aE0gYTKep6xxrpr6vNVaISqJl4OzN3B8ckDtJRAPdfyanQQ ซึ่งเป็นคำสั่งในการตรวจสอบสถานะโปรแกรม WMC

ผลที่คาดว่าจะได้รับ: ไม่สามารถอ่านข้อความเป็นภาษาที่เข้าใจได้ เนื่องจากมีการเข้ารหัส

ผลที่ได้รับ: ดักจับข้อความ T9v2Szx1ZUVR/tce6aE0gYTKep6xxrpr6vNVaISqJl4OzN3B8ckDtJRAPdfyanQQ ได้ แต่ไม่สามารถถอดรหัสได้เนื่องจากไม่รู้คีย์ที่ใช้เข้ารหัส

4.1.15.2 ทดสอบการกลั่นแกล้งโดยพยายามส่งข้อมูลการแจ้งเตือนการเข้าเว็บไซต์ที่ไม่อนุญาตมายังโปรแกรม WMS

วิธีการทดสอบ: ส่งข้อมูลการแจ้งเตือนที่ได้จากการดักจับไปยังเครื่องแม่ข่าย คือJ0Pm1Ts0iMUG4EusTBMfG4TJyT+yp0dEk6PzAmKET+I= ซึ่งเป็นข้อมูลการแจ้งเตือนการเข้าเว็บไซต์ที่ไม่อนุญาตให้เข้า

ผลที่คาดว่าจะได้รับ: โปรแกรม WMS ไม่รับข้อมูลการแจ้งเตือน เพราะว่าถูกส่งมาจากหมายเลขไอพีที่ไม่มีอยู่ในรายการ

ผลที่ได้รับ: โปรแกรม WMS ไม่อัพเดตการแสดงผลใดๆ

4.2 การทดสอบในระดับ System Test ของโปรแกรม WMC

โปรแกรม WMC ในความจริงแล้วจะไม่มีหน้าตาส่วนติดต่อกับผู้ใช้งาน แต่ในผลการทดสอบนี้จะแสดงให้เห็น เนื่องจากว่าง่ายต่อการแสดงผล

คุณสมบัติเครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ

-ซีพียู: AMD Athlon X2 4400e 3.2 GHz

-หน่วยความจำ: 2.0 GB

-หน่วยเก็บข้อมูลหลัก: Western Digital 640GB ความเร็ว 7200 RPM

-ระบบปฏิบัติการ: Microsoft Windows 7 SP1 X86

-ข้อมูลการเชื่อมต่อ: Ethernet ความเร็ว 100 Mbps โดยเชื่อมต่อกับ Switch ความเร็วอินเทอร์เน็ต 6 Mbps

ตารางที่ 4.5 ตารางแสดงการทดสอบในระดับ System Test ของโปรแกรม WMC

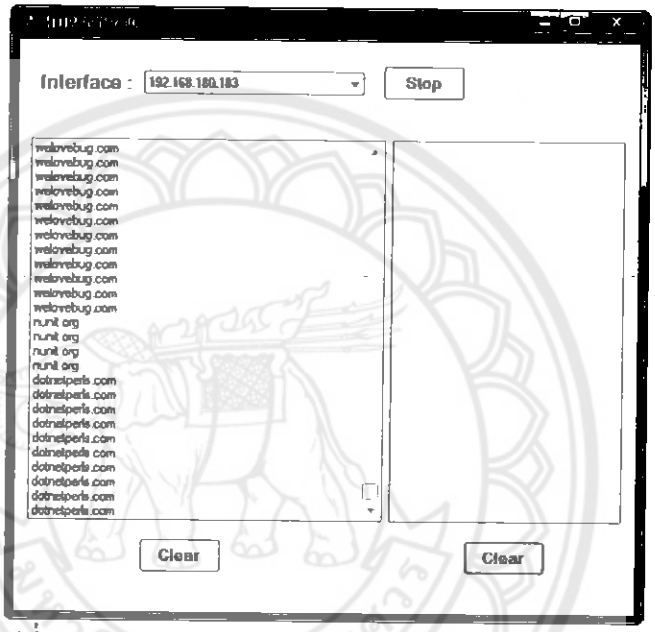
ประเภทการทดสอบ	ชื่อการทดสอบ
Functional	ทดสอบการตรวจจับเว็บไซต์จำนวน 115 เว็บไซต์
	ทดสอบการตรวจจับเว็บไซต์ที่ไม่อนุญาตจำนวน 115 เว็บไซต์
Non - Functional	ทดสอบประสิทธิภาพการทำงานของโปรแกรม (Performance)
	ทดสอบความน่าเชื่อถือของโปรแกรม (Reliability)
	ทดสอบความปลอดภัยของโปรแกรม (Security)

4.2.1 ทดสอบการตรวจจับเว็บไซต์จำนวน 115 เว็บไซต์

วิธีการทดสอบ: ทดสอบโดยการเข้าเว็บไซต์จำนวน 115 เว็บไซต์ที่แตกต่างกันทั้งหมด

ผลที่คาดว่าจะได้รับ: โปรแกรม WMC สามารถตรวจจับได้ว่ามีการเข้าเว็บไซต์จำนวน 115 เว็บไซต์ที่ทดสอบได้ทั้งหมด

ผลที่ได้รับ โปรแกรม WMC สามารถตรวจจับได้ว่ามีการเข้าเว็บไซต์จำนวน 115 เว็บไซต์ที่ทดสอบได้ทั้งหมด ดังรูปที่ 4.34 โดยช่องด้านซ้ายมือแสดงชื่อเว็บไซต์ที่มีการตรวจจับได้ในการเข้าเว็บไซต์แต่ละเว็บไซต์



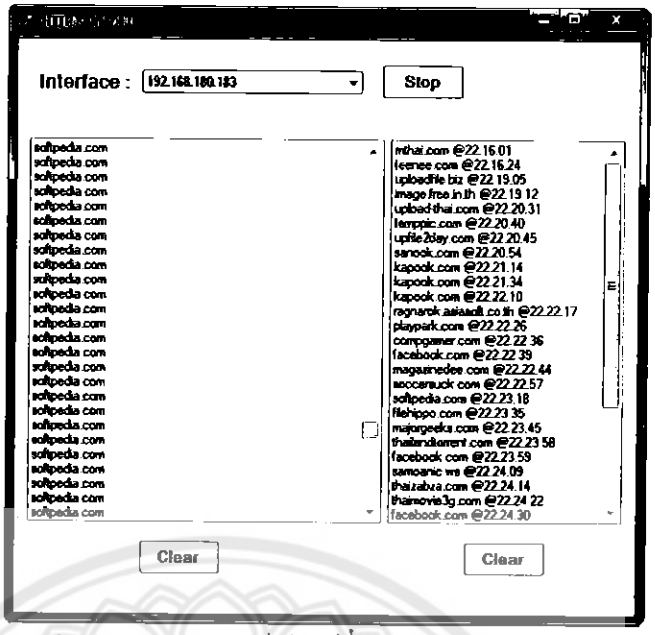
รูปที่ 4.34 ตัวอย่างผลการตรวจจับเว็บไซต์จำนวน 115 เว็บไซต์

4.2.2 ทดสอบการตรวจจับเว็บไซต์ที่ไม่อนุญาตจำนวน 115 เว็บไซต์

วิธีการทดสอบ: ทดสอบโดยการเข้าเว็บไซต์จำนวน 115 เว็บไซต์ที่แตกต่างกันทั้งหมด โดยเป็นเว็บไซต์ที่ไม่อนุญาตเข้า

ผลที่คาดว่าจะได้รับ: โปรแกรม WMC สามารถตรวจจับได้ว่ามีการเข้าเว็บไซต์จำนวน 115 ที่ไม่อนุญาตให้เข้าได้ทั้งหมด

ผลที่ได้รับ: โปรแกรม WMC สามารถตรวจจับได้ว่ามีการเข้าเว็บไซต์จำนวน 115 ที่ไม่อนุญาตให้เข้าได้ทั้งหมด ดังรูปที่ 4.35 โดยช่องด้านซ้ายมือแสดงชื่อเว็บไซต์ที่มีการตรวจจับได้ในการเข้าเว็บไซต์แต่ละเว็บไซต์ ส่วนช่องด้านขวามือแสดง ชื่อเว็บไซต์@เวลาที่เข้า ซึ่งเป็นเว็บไซต์ที่ไม่อนุญาตให้เข้า



รูปที่ 4.35 ตัวอย่างผลการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าจำนวน 115 เว็บไซต์

4.2.3 ทดสอบประสิทธิภาพการทำงานของโปรแกรม (Performance)

เป็นการทดสอบว่าโปรแกรมใช้เวลาในการวิเคราะห์แต่และแพ็คเกต (Packet) ในการโหลดเว็บไซต์หนึ่งๆ ทันกับการที่บราวเซอร์โหลดหน้าเว็บเพื่อจนเสร็จหรือไม่ ถ้าสามารถวิเคราะห์แต่และแพ็คเกตเสร็จก่อนที่บราวเซอร์จะโหลดเสร็จหรือเสร็จเกือบจะพร้อมกันพอดี แสดงว่ามีประสิทธิภาพการทำงานที่รวดเร็วและยอมรับได้ แต่ถ้าโปรแกรมใช้เวลาในการวิเคราะห์นานเกินกว่าบราวเซอร์โหลดหน้าเว็บเสร็จแล้ว แสดงว่าโปรแกรมมีประสิทธิภาพการทำงานที่ช้า ซึ่งจะช้ามากหรือน้อยนั้นขึ้นอยู่กับเวลาที่นับหลังจากที่บราวเซอร์โหลดเสร็จแล้วจนถึงเวลาที่โปรแกรมหยุดการวิเคราะห์แพ็คเกต

วิธีการทดสอบ: เปิดเว็บไซต์ แล้วจับเวลาการวิเคราะห์แพ็คเกตตั้งแต่บราวเซอร์เริ่มโหลดจนบราวเซอร์โหลดเสร็จสิ้น โดยทดสอบกับเว็บไซต์จำนวน 10 เว็บไซต์

ตารางที่ 4.6 ตารางแสดงผลทดสอบประสิทธิภาพการทำงานของโปรแกรม WMC

ชื่อเว็บไซต์	จำนวนแพ็คเกตที่ทำการวิเคราะห์	เวลาที่ใช้		
		เสร็จก่อนบราวเซอร์ (วินาที)	เสร็จพร้อมบราวเซอร์พอดี	เสร็จช้ากว่าบราวเซอร์ (วินาที)
www.mthai.com	302		✓	
www.kapook.com	638		✓	
www.teenee.com	565		✓	
www.hardwaresecrets.com	72		✓	
www.bangkokpost.com	214		✓	
www.dailynews.co.th	124		✓	
www.tomshardware.com	95		✓	
www.mornor.com	68		✓	
www.student-weekly.com	42		✓	
www.palungjit.com	41		✓	

4.2.4 ทดสอบความน่าเชื่อถือของโปรแกรม (Reliability)

เป็นการทดสอบว่าโปรแกรมมีการใช้ซีพียูและหน่วยความจำเป็นปริมาณที่มากน้อยแค่ไหน อยู่ในระดับที่ยอมรับได้หรือไม่ รวมถึงโปรแกรมสามารถทำงานได้อย่างยาวนาน โดยที่ไม่ทำให้เครื่องคอมพิวเตอร์ล่ม หรือเกิดหน่วยความจำรั่ว (Memory leak) หรือไม่

วิธีทำการทดสอบ คือ เครื่องลูกข่ายแต่ละเครื่องมีการเข้าเว็บไซต์แบบสุ่มจำนวน 115 เว็บไซต์ ทุกๆ 30 วินาที โดยสังเกตการการใช้ซีพียูและหน่วยความจำของทุกๆ เครื่อง และเอาปริมาณการใช้ซีพียูและหน่วยความจำที่สูงที่สุดจากปริมาณการใช้ซีพียูและหน่วยความจำของเครื่องทั้งหมดมาใช้ในการสรุป

ตารางที่ 4.7 ตารางแสดงผลการใช้หน่วยความจำของโปรแกรม WMC

การทำงาน	ปริมาณการใช้หน่วยความจำ	
	ขณะไม่ได้ทำงาน (MB)	ขณะตรวจจับเว็บไซต์ (MB)
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า	ไม่เกิน 15 MB	ไม่เกิน 35 MB
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งานพร้อมกัน	ไม่เกิน 15 MB	ไม่เกิน 35 MB
ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์	ไม่เกิน 15 MB	ไม่เกิน 35 MB

ตารางที่ 4.8 ตารางแสดงผลการใช้ซีพียูของโปรแกรม WMC

การทำงาน	ปริมาณการใช้ซีพียู	
	ขณะไม่ได้ทำงาน (%)	ขณะรับการแจ้งเตือน (%)
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้า	0 %	ไม่เกิน 10 %
ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าและโปรแกรมที่ไม่อนุญาตให้ใช้งานพร้อมกัน	0 %	ไม่เกิน 10 %
ตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับนำมาวิเคราะห์	0 %	ไม่เกิน 10 %

4.2.5 ทดสอบด้านความปลอดภัยของโปรแกรม (Security)

4.2.5.1 ทดสอบการดักจับข้อมูลระหว่างการส่งข้อมูลไปยังเครื่องแม่ข่าย

วิธีการทดสอบ: ดักจับข้อมูลการส่งจากเครื่องลูกข่ายไปยังเครื่องแม่ขายด้วยโปรแกรม Wireshark

ผลที่คาดว่าจะได้รับ: ไม่สามารถอ่านข้อความเป็นภาษาที่เข้าใจได้ เนื่องจากมีการเข้ารหัส

ผลที่ได้รับ: ข้อความที่ดักจับได้ คือ T9v2Szx1ZUVR/tce6aE0gYTKep6xxrpr6vNVaISqJl4OzN3B8ckDtJRAPdfyanQQ แต่ไม่สามารถถอดรหัสได้เนื่องจากไม่รู้คีย์ที่ใช้เข้ารหัส

4.2.5.2 ทดสอบการกลั่นแกล้งโดยพยายามปลอมตัวว่าเป็นเครื่องแม่ข่ายแล้วส่งข้อความมาเพื่อสั่งหยุดการตรวจจับเว็บไซต์

วิธีการทดสอบ: ทดสอบส่งข้อความที่มีการเข้ารหัส ซึ่งเป็นคำสั่งที่สั่งหยุดการตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าไปยังเครื่องลูกข่าย คือ Y6rizcp6J83vR8mD12ecY/lpMmumIoy+QXCg/nkoO2A=

ผลที่คาดว่าจะได้รับ: โปรแกรม WMC ไม่สนใจข้อความที่ส่งมา และยังคงตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าต่อไป เพราะว่าหมายเลขไอพีที่ส่งมาไม่ใช่หมายเลขไอพีของเครื่องแม่ข่าย

ผลที่ได้รับ: โปรแกรม WMC ยังคงตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าต่อไป

4.3 สรุปผลการทดสอบโปรแกรม

ตารางที่ 4.9 ตารางแสดงผลการทดสอบในระดับ System Test ของโปรแกรม WMS

ชื่อการทดสอบ	ผลการทดสอบ
ค้นหาหมายเลขไอพีและชื่อเครื่องลูกข่ายทั้งหมดที่ทำงานอยู่ในเครือข่ายส่วนตัว (LAN) เดียวกัน	ผ่าน
ตรวจสอบสถานะของคอมพิวเตอร์ลูกข่าย	ผ่าน
ตรวจสอบสถานะของโปรแกรม WMC	ผ่าน
สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าใช้งาน	ผ่าน
สั่งให้โปรแกรม WMC หยุดตรวจจับเว็บไซต์ที่ไม่อนุญาตให้เข้าใช้งาน	ผ่าน
สั่งให้โปรแกรม WMC ตรวจจับโปรแกรมที่ไม่อนุญาตให้เข้าใช้งาน	ผ่าน
สั่งให้โปรแกรม WMC หยุดตรวจจับโปรแกรมที่ไม่อนุญาตให้เข้าใช้งาน	ผ่าน
สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์ที่เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์	ผ่าน
สั่งให้โปรแกรม WMC ตรวจจับเว็บไซต์ที่เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์	ผ่าน
ให้โปรแกรม WMC หยุดตรวจจับเว็บไซต์ที่เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์	ผ่าน
สั่งให้โปรแกรม WMC ส่งเทกซ์ไฟล์ (Text file) ไปยังเครื่องลูกข่าย	ผ่าน
สั่งให้โปรแกรม WMC ดาวน์โหลดไฟล์รูปภาพจากเครื่องลูกข่าย	ผ่าน
ทดสอบประสิทธิภาพการทำงานของโปรแกรม (Performance)	ผ่าน
ทดสอบความน่าเชื่อถือของโปรแกรม (Reliability)	ผ่าน
ทดสอบด้านความปลอดภัยของโปรแกรม (Security)	ผ่าน

ตารางที่ 4.10 ตารางแสดงการทดสอบในระดับ System Test ของโปรแกรม WMC

ชื่อการทดสอบ	ผลการทดสอบ
ทดสอบการตรวจจับเว็บไซต์จำนวน 115 เว็บไซต์	ผ่าน
ทดสอบการตรวจจับเว็บไซต์ที่ไม่อนุญาตจำนวน 115 เว็บไซต์	ผ่าน
ทดสอบประสิทธิภาพการทำงานของโปรแกรม (Performance)	ผ่าน
ทดสอบความน่าเชื่อถือของโปรแกรม (Reliability)	ผ่าน
ทดสอบความปลอดภัยของโปรแกรม (Security)	ผ่าน

บทที่ 5

สรุปผลการดำเนินงาน

โครงงานนี้มีวัตถุประสงค์เพื่อพัฒนาโปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตที่เรียน และเพื่อให้ผู้สอนนำโปรแกรมช่วยแจ้งเตือนการใช้งานคอมพิวเตอร์นอกขอบเขตวิชาที่เรียนไปใช้เป็นเครื่องมือในการเรียนการสอน

ความรู้ที่ใช้ในการทำโครงงานนี้ ประกอบด้วย

- ความรู้เกี่ยวอุปกรณ์ทางเครือข่าย เช่น ฮับ (Hub) สวิตซ์ (Switch)
- ความรู้เกี่ยวกับ OSI Model และ TCP/IP Protocol Suite
- ความรู้เกี่ยวภาษา C# .NET และ .NET Framework
- ความรู้เกี่ยวกับการใช้งานฐานข้อมูล SQLite
- ความรู้เกี่ยวกับการเขียนโปรแกรมแบบ Socket Programming, Multi - Threading Programming และการเขียนโปรแกรมแสดงผลแบบ GUI ของภาษา C# .NET

การออกแบบโปรแกรมนั้น มีการพัฒนา 2 โปรแกรมด้วยกัน คือ

1. โปรแกรมที่เครื่องแม่ข่าย ซึ่งจะคอยควบคุมและสั่งการให้ โปรแกรมที่เครื่องลูกข่ายทำงาน โดยมีหน้าตาสำหรับติดต่อกับผู้ใช้
2. โปรแกรมที่เครื่องลูกข่าย ซึ่งจะคอยรับคำสั่งการทำงานจากโปรแกรมที่เครื่องแม่ข่ายและแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่าย โดยไม่มีหน้าตาสำหรับติดต่อกับผู้ใช้

โปรแกรมทั้ง 2 โปรแกรม ออกแบบด้วยภาษา UML ซึ่งประกอบด้วย

- Use case diagram แสดงให้เห็นว่าผู้ใช้สามารถใช้งานอะไรของโปรแกรมได้บ้าง
- Component diagram แสดงให้เห็นถึงส่วนประกอบภายนอกโปรแกรมเรียกใช้
- Activity diagram แสดงให้เห็นถึงขั้นตอนการทำงานภายในของโปรแกรม
- Class diagram แสดงให้เห็นถึงความสัมพันธ์และการติดต่อกันระหว่างคลาสภายในโปรแกรม

การทดสอบโปรแกรมนั้น เป็นการทดสอบในระดับ System Test ซึ่งเป็นการทดสอบโปรแกรมโดยรวมทั้งหมดว่าสามารถทำงานได้ตามที่กำหนดไว้หรือไม่ โดยทำการทดสอบทั้ง 2 โปรแกรม สามารถแบ่งประเภทการทดสอบได้ดังนี้

- Functional คือ การทดสอบเกี่ยวกับหน้าที่การทำงานของโปรแกรม
- Non – Functional คือ การทดสอบที่ไม่เกี่ยวกับการทำงานของโปรแกรม

5.1 สรุปผลการทดสอบ

จากการทสอบโปรแกรมในบทที่ 4 นั้น สามารถสรุปผลการดำเนินการได้ดังนี้

<u>ความสามารถของโปรแกรมที่เครื่องแม่ข่าย</u>

1. โปรแกรมสามารถค้นหาหมายเลขไอพีและชื่อเครื่องภายในวงแลนเดียวกันได้
2. โปรแกรมสามารถตรวจสอบสถานะของเครื่องลูกข่ายและโปรแกรมที่เครื่องลูกข่ายได้
3. โปรแกรมสามารถสั่งให้โปรแกรมที่เครื่องลูกข่ายทำงานและหยุดทำงานตามที่ต้องการได้
4. โปรแกรมสามารถส่งไฟล์ที่เก็บรายชื่อเว็บไซต์หรือโปรแกรมไปยังเครื่องลูกข่ายได้
5. โปรแกรมสามารถดาวน์โหลดรูปภาพที่บันทึกไว้ในขณะเข้าเว็บไซต์หรือใช้งานโปรแกรมที่ไม่อนุญาตจากเครื่องลูกข่ายได้
6. โปรแกรมสามารถบันทึกข้อมูลการเข้าเว็บไซต์ลงฐานข้อมูลและสามารถสรุปข้อมูลการบันทึกเว็บไซต์ทั้งหมดได้
9. โปรแกรมสามารถแจ้งเตือนด้วยเสียงและสามารถปิด เปิดเสียงการแจ้งเตือนได้

<u>ความสามารถของโปรแกรมที่เครื่องลูกข่าย</u>

1. โปรแกรมสามารถตรวจจับเว็บไซต์และแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่ายเมื่อมีการเข้าเว็บที่ไม่อนุญาตได้
2. โปรแกรมสามารถตรวจจับโปรแกรมและแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่ายเมื่อมีการใช้งานโปรแกรมที่ไม่อนุญาตได้
3. โปรแกรมสามารถตรวจจับเว็บไซต์เพื่อบันทึกข้อมูลสำหรับการนำมาวิเคราะห์และแจ้งเตือนไปยังโปรแกรมที่เครื่องแม่ข่ายได้

5.2 ปัญหาและอุปสรรค

ตารางที่ 5.1 ตารางแสดงปัญหาและอุปสรรค แนวทางแก้ไข

ปัญหาและอุปสรรค	แนวทางแก้ไข
1. เกิดปัญหาในเรื่องของ Thread มีการอ่านและเขียนข้อมูลในเวลาเดียวกัน ซึ่งอาจทำให้สถานะของข้อมูลไม่ถูกต้องตามที่ควรจะเป็น หรือทำให้โปรแกรมเกิดข้อผิดพลาดได้	1. ศึกษาหลักการของ Thread Synchronization ให้เข้าใจ รวมทั้งการเขียนโปรแกรมลักษณะนี้ใน C# .NET โดยศึกษาจากเว็บต่างๆ ที่เกี่ยวข้อง รวมทั้งซอร์สโค้ด (Source code) ของผู้อื่น
2. การทดสอบโปรแกรมกับหลายๆ เครื่อง มีความยุ่งยากพอสมควร ถ้ามีการแก้ไขโปรแกรมที่เครื่องลูกข่ายใหม่ก็ต้องนำไปติดตั้งใหม่บนคอมพิวเตอร์หลายๆ เครื่องทำให้ต้องเสียเวลาไปพอสมควร	2. ลองทดสอบกับคอมพิวเตอร์จำนวนน้อยๆ ก่อน พอมั่นใจว่าโปรแกรมทำงานได้อย่างถูกต้องแล้วค่อยเพิ่มจำนวนเครื่องขึ้นไป หรือเขียนโปรแกรมสำหรับส่งไฟล์ไปยังเครื่องลูกข่ายโดยเฉพาะขึ้นมา
3. การเขียนโปรแกรมแบบ Socket Programming ใน C#.NET มีการกำหนด Event ซ้ำซ้อนกันทำให้ Event มีการสร้าง Thread หลาย Thread และมีการใช้ข้อมูลร่วมกัน ส่งผลให้โปรแกรมมีข้อผิดพลาดที่ยากในการแก้ไข เนื่องจากขาดความเข้าใจอย่างแท้จริง	3. ศึกษากระบวนการเกิด Event ให้ถ่องแท้ และต้องระมัดระวังไม่ให้เกิด Event ซ้ำซ้อนกัน โดยตรวจสอบโค้ด (Code) อย่างรอบคอบ และศึกษาจากเว็บต่างๆ ที่เกี่ยวข้อง รวมทั้งซอร์สโค้ด (Source code) ของผู้อื่น
4. ขาดความรู้ในการโปรแกรมให้ทำงานแบบ Asynchronous เพื่อให้โปรแกรมทำงานได้หลายๆ อย่างพร้อมกัน	4. ศึกษาวิธีการเขียนโปรแกรมแบบ Asynchronous ใน C# .NET โดยศึกษาจากเว็บต่างๆ ที่เกี่ยวข้อง รวมทั้งซอร์สโค้ด (Source code) ของผู้อื่น

ตารางที่ 5.1 ตารางแสดงปัญหาและอุปสรรค แนวทางแก้ไข (ต่อ)

ปัญหาและอุปสรรค	แนวทางแก้ไข
5. การใช้งาน Thread ร่วมกับการแสดงผลแบบ GUI มีปัญหาเรื่อง Thread อื่นที่ไม่ใช่ Thread ที่สร้าง GUI ไม่สามารถเข้าถึง Control บน Windows Form ได้โดยตรง	5. ศึกษาการเขียนโปรแกรมเกี่ยวกับการทำงานหลายอย่างพร้อมกันโดยใช้ Thread ร่วมกับ GUI ใน C# .NET โดยศึกษาจากเว็บต่างๆ ที่เกี่ยวข้อง รวมทั้งซอร์สโค้ด (Source code) ของผู้อื่น

5.3 ความต้องการของโปรแกรม

1. จำเป็นต้องติดตั้ง .Net Framework 4.0 สำหรับใช้ในการรันโปรแกรม
2. โปรแกรมสามารถทำงานได้บนระบบปฏิบัติการตั้งแต่ Windows XP ขึ้นไป

5.4 ข้อจำกัดของโปรแกรม

1. โปรแกรมไม่สามารถตรวจจับเว็บไซต์ที่กำลังเข้าชมอยู่ได้อย่างแม่นยำ เนื่องจากว่าบางเว็บไซต์หรือบาง URL มีการไปอ้างอิงหรือไปดึงข้อมูลจากที่อื่นๆ ทำให้การเข้าชมเว็บไซต์หนึ่งๆ อาจจะทำให้การแจ้งเตือนเว็บไซต์ที่ไม่อนุญาตให้เข้านั้นไม่ถูกต้องตามความเป็นจริงได้ เช่น ไม่อนุญาตให้เข้าเว็บไซต์ facebook.com แต่พอเข้าเว็บไซต์ f0nt.com ก็จะทำให้มีการแจ้งเตือนว่ามีการเข้าเว็บไซต์ facebook.com เพราะเว็บไซต์ f0nt.com มีการไปอ้างอิงหรือไปดึงข้อมูลจากเว็บไซต์ facebook.com มาใช้งานหรือแสดงผล

5.5 ข้อแตกต่างเมื่อเทียบกับโปรแกรมอื่นๆ ที่คล้ายกัน

1. โปรแกรมตรวจจับเฉพาะชื่อเว็บไซต์เท่านั้น ไม่ได้ตรวจจับ URL แต่โปรแกรมอื่นๆ เช่น PC Activity Monitor และ Real Spy Monitor นั้นจะตรวจจับทั้ง URL ซึ่งไม่ตรงกับความต้องการที่ต้องการเพียงแค่ชื่อเว็บไซต์เท่านั้นสำหรับการนำไปแจ้งเตือน

2. โปรแกรมสามารถแจ้งเตือนด้วยเสียงได้แบบ Real-Time แต่โปรแกรมอื่นๆ เช่น PC Activity Monitor และ Real Spy Monitor นั้นไม่สามารถทำได้ซึ่งไม่ตรงกับความต้องการ

3. โปรแกรมนี้พัฒนาขึ้นมาตามความต้องการใช้งานจริงๆ แต่โปรแกรมอื่นๆ เช่น PC Activity Monitor และ Real Spy Monitor มีฟีเจอร์การใช้งานที่หลากหลายมาก ซึ่งไม่ค่อยตรงกับความต้องการและยังมีค่าใช้จ่ายในการซื้อซอฟแวร์แบบถูกลิขสิทธิ์ที่ค่อนข้างสูง ถ้านำมาใช้กับคอมพิวเตอร์หลายๆ เครื่อง

5.6 ข้อเสนอแนะ

1. เนื่องจากว่าโปรแกรม WMC ไม่สามารถซ่อนตัวจากการถูกพบเห็นได้ จึงสามารถที่จะถูกปิดโดยผู้ใช้คอมพิวเตอร์ได้ แต่ถ้ามีการจำกัดบัญชีผู้ใช้งานเป็น Standard user ใน Windows 7 หรือ Limited accountใน Windows XP แล้วรันโปรแกรม WMC ในรูปแบบของ Windows Service ก็จะสามารถป้องกันการถูกปิดได้ แต่ข้อเสีย คือ การบันทึกภาพหน้าจอขณะใช้งานจะไม่สามารถใช้งานได้ ซึ่งจะส่งผลต่อการตรวจสอบว่าผู้ใช้งานได้ทำการเข้าเว็บไซต์นั้นจริงๆ หรือไม่

5.7 ความรู้ที่ต้องมีในการพัฒนาต่อ

ผู้ที่สนใจพัฒนาโปรแกรมนี้ต่อ ควรจะมีความรู้ดังต่อไปนี้

1. ความรู้เกี่ยวกับ TCP/IP Protocol suite
2. การเขียนโปรแกรมแบบ Socket Programming ของภาษา C# .NET สำหรับใช้ในการตรวจจับเว็บไซต์ การส่งข้อมูลระหว่างเครื่องแม่ข่ายและเครื่องลูกข่าย การรับ – ส่งไฟล์ และการค้นหาหมายเลขไอพีและชื่อเครื่องลูกข่าย
3. การเขียนโปรแกรมแบบ Multi–Threading Programming หรือ Asynchronous Programming ของภาษา C# .NET สำหรับใช้ในการทำงานหรือแสดงผลผ่าน GUI หลายๆ อย่างพร้อมกัน

5.8 แนวทางในการพัฒนาต่อในอนาคต

1. ผู้จัดทำพบว่าการตรวจจับเว็บไซต์นั้น นอกจากว่าจะใช้วิธีการดักจับแพ็คเกต (Packet) แล้วยังมีอีกวิธีการหนึ่งที่โปรแกรมอื่นๆ ใช้ เช่น PC Activity Monitor และ Real Spy Monitor ผู้จัดทำคิดว่าน่าจะใช้ Win32API ในการตรวจหาข้อความที่อยู่ใน Address bar ของ Browser เช่น Internet Explorer, Firefox, Chrome และ Opera ซึ่งวิธีนี้จะทำให้การตรวจจับเว็บไซต์ได้เที่ยงตรงและลดการใช้งานซีพียูและหน่วยความจำได้พอสมควร แต่ต้องศึกษาการใช้งาน Win32API ที่ค่อนข้างเข้าใจยากพอสมควร รวมทั้งแหล่งข้อมูลออนไลน์เกี่ยวกับการตรวจจับด้วยวิธีนี้ไม่มีเลย

2. พัฒนาโปรแกรมให้สามารถตรวจจับการเล่นเกมแฟลช (Flash) บนเว็บไซต์ได้ รวมทั้งการแชต (Chat) ผ่านโปรแกรมแชต เช่น Windows Live Messenger (WLM) โดยเกมแฟลชนั้นอาจจะตรวจจับได้จากการวิเคราะห์หมายเลขพอร์ตปลายทาง (Destination port number) ส่วนโปรแกรมแชตอย่าง Windows Live Messenger นั้น อาจจะวิเคราะห์ได้จากชนิดของโปรโตคอลที่ใช้

3. พัฒนาโปรแกรม WMC ให้สามารถซ่อนตัวจากการถูกพบเห็นได้ เพื่อป้องกันการถูกปิดจากผู้ใช้งาน ผู้จัดทำเสนอว่าต้องใช้ Win32API ซึ่งโดยปกแล้วฟังก์ชันการซ่อนตัวนี้ไม่มีให้ใช้งานอย่างแน่นอน จำเป็นต้องใช้เทคนิคอื่นๆ ที่ดูคล้ายกับการเจาะระบบ (Hack) ระบบปฏิบัติการ Windows โดยเอามาประยุกต์ร่วมกันกับ Win32API

เอกสารอ้างอิง

[1] **Switch กับ hub แตกต่างกันอย่างไร.** สืบค้นเมื่อ 15 กรกฎาคม 2554,
จาก http://cha-uat.net/oldweb/freebsd/all/index.php?page=network4

[2] **เราเตอร์ (Router).** สืบค้นเมื่อ 15 กรกฎาคม 2554,
จาก http://www.bothong.ac.th/Te43102/pag10b3.html

[3] **สวิตซ์ (Switch).** สืบค้นเมื่อ 15 กรกฎาคม 2554,
จาก http://www.nkatc.svec.go.th/pictures/e-learning/E-Learning3/unit3.html

[4] วิชชุลดา คำอ้าย. **OSI Model.** สืบค้นเมื่อ 15 กรกฎาคม 2554,
จาก http://mymint.tripod.com/report5.html

[5] **โปรโตคอล TCP/IP.** สืบค้นเมื่อ 15 กรกฎาคม 2554,
จาก http://www.it.co.th/networkdetail.php?n_id=21

[6] **NET คืออะไรและมีบทบาทอย่างไร.** สืบค้นเมื่อ 20 กรกฎาคม 2554,
จาก http://cpe.kmutt.ac.th/previousproject/2004/6/2.2.htm

[7] **จับฉ่าย SQLite.** สืบค้นเมื่อ 20 กรกฎาคม 2554, จาก
http://my.thaifox.net/modules.php?name=News&file=categories&op=newindex&catid=2

วัจนพงศ์ เกษมศิริ. (2551). **เอกสารประกอบการสอนวิชา 01074201 Network Programming.**
กรุงเทพฯ: ภาควิชาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

สุนทริน วงศ์ศิริกุล และชัยวัฒน์ สิทธิกร โอฬารกุล. (2550).
การพัฒนาโมดุลสำหรับการเขียนโปรแกรมเชิง วัตถุด้วย UML 2.0. พิมพ์ครั้งที่ 1.
กรุงเทพฯ: บริษัทซัคเซส มีเดีย จำกัด.

Hypertext Transfer Protocol (HTTP). สืบค้นเมื่อ 20 กรกฎาคม 2554,
จาก http://staff.cs.psu.ac.th/noi/cs344-481/group11_Http/HTTP.htm

Richard Blum. (2546). **C# Network Programming.** United States of America: SYBEX.

Joseph Albahari. (27 เมษายน 2554). **Threading in C#.**
United States of America: O'Reilly Media.