โปรแกรมติดตามและบันทึกพฤติกรรมผู้ใช้งาน

USER BEHAVIOR MONITORING AND LOGGING PROGRAM

ณัชชา นิยมธรรมกิจ

สัภยา ธัญญสกุลกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมติดตามและบันทึกพฤติกรรมผู้ใช้งาน

USER BEHAVIOR MONITORING AND LOGGING PROGRAM

ผู้จัดทำ

1.) นางสาว ณัชชา นิยมธรรมกิจ รหัสนักศึกษา 47010212

2.) นาย สักยา ธัญญสกุลกิจ รหัสนักศึกษา 47010825

อาจารย์ที่ปรึกษา

(ผศ.ธนา หงษ์สุวรรณ)

โปรแกรมติดตามและบันทึกพฤติกรรมผู้ใช้งาน

นางสาวณัชชา นิยมธรรมกิจ 47010212
นายสัภยา ธัญญสกุลกิจ 47010825
ผศ.ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

บทคัดย่อ

การใช้งานอินเทอร์เน็ตในปัจจุบันมีความแพร่หลายมากขึ้น ไม่ว่าจะเป็นการใช้ตามบ้านเรือน หรือใช้ในเชิงธุรกิจ เนื่องจากอินเทอร์เน็ตเป็นทั้งแหล่งให้ความบันเทิง และให้ความรู้อันมหาศาล การใช้งานอินเทอร์เน็ตจะเกิดประโยชน์อย่างมาก ถ้าใช้ให้ถูกต้องตามกาลเทศะ แต่ในปัจจุบันการใช้งานอินเทอร์เน็ตในทางที่ไม่เหมาะสมยังมีอีกมาก ดังนั้นจึงเกิดแนวคิดในการสร้างโปรแกรมตรวจสอบและบันทึกการใช้งานของผู้ใช้งาน โดยมีจุดประสงค์เพื่อส่งเสริมให้การทำงานมีประสิทธิภาพสูงสุด

ปริญญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อสร้างโปรแกรมตรวจสอบและบันทึกพฤติกรรมการใช้งานอินเทอร์เน็ตในโปรโตคอลเอชทีทีพี และ Instant Messenger เป็นหลัก นอกจากนี้ยังมีการเทียบการใช้งานเป็นเปอร์เซ็นต์ทำให้เห็นภาพรวมได้มากขึ้น มีการเก็บบันทึกการใช้งาน เพื่อนำมารวบรวมเป็นสถิติและแสดงผลออกมาในรูปของกราฟ ในส่วนของการเก็บบันทึกรายละเอียดของการใช้งานต่างๆจะแยกออกเป็นไอพีและเวลาที่ใช้อย่างชัดเจน

USER BEHAVIOR MONITORING AND LOGGING PROGRAM

Ms. Natcha	Niyomthammakit	47010212
Mr. Sappaya	Tanyasakulkit	47010825
Asst. Prof. Thana	Hongsuwan	Advisor

Academic Year 2007

ABSTRACT

Nowadays internet using have been already widespread more and more even in home or business. Because internet is the great resources for entertainment and knowledge if it has been used in the right way. Anyway misuses of internet still out of control then this is the main point for building monitoring program that detects and logs internet using to files. The purposes are for maintaining the system efficiency.

User behavior monitoring is developed in order to detect and logging internet using behavior of the user especially in using HTTP and MSN. More over , this application can display usage traffic of each user compare by percentage to total traffic in the network. By logging these information to system , we have history files for examine , calculate and making statistical report such as graph.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี เนื่องจากคณะผู้จัดทำได้รับคำแนะนำ และการให้คำปรึกษาที่ดีจากผู้ช่วยศาสตราจารย์ธนา หงส์สุวรรณ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ท่านได้กรุณาสละเวลาให้คำปรึกษา เสนอแนะ และดูแลเอาใจใส่เป็นอย่างดี รวมทั้งแก้ไขข้อบกพร่องของโครงงานให้มีความถูกต้องและสมบูรณ์ยิ่งขึ้น พร้อมทั้งคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การอบรมสั่งสอนประสาทวิชาความรู้ รวมถึงขอขอบคุณห้องแล็บเน็ตเวิร์คที่เอื้อเฟื้อสถานที่ปฏิบัติงานให้แก่คณะผู้จัดทำ

ขอกราบขอบพระคุณบิดา มารดา ญาติพี่น้อง และเพื่อนๆทุกคน ที่คอยให้คำปรึกษา และเป็นกำลังใจที่ดีเสมอมา และสุดท้ายขอขอบพระคุณผู้ที่มีส่วนช่วยเหลือในการแนะนำ ติชมและให้กำลังใจในการจัดทำโครงงานทุกๆท่านมา ณ โอกาสนี้ด้วย

ณัชชา นิยมธรรมกิจ

สัภยา ธัญญสกุลกิจ

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

เรื่อง **หน้า**

สารบัญ (ต่อ)

สารบัญตาราง

ตาราง **หน้า**

สารบัญรูป

รูป **หน้า**

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบัน การใช้คอมพิวเตอร์เพื่อเพิ่มความสะดวกสบายในการปฏิบัติงานมีมากขึ้น ภายในองค์กรจึงมีการใช้บริการอินเทอร์เน็ต การใช้งานโปรแกรมสนทนา (MSN) ของผู้ปฏิบัติงาน เพื่อเพิ่มโอกาสให้กับธุรกิจ และส่งผลให้ธุรกิจสามารถขยายตัวและพัฒนาอย่างรวดเร็วอีกด้วย

เนื่องจากการให้อิสระในการใช้งานอินเทอร์เน็ตกับผู้ปฏิบัติงาน องค์กรจึงจำเป็นต้องเสี่ยงกับการใช้งานอินเทอร์เน็ตของผู้ปฏิบัติงาน ที่ผิดวัตถุประสงค์ขององค์กร ซึ่งเป็นสิ่งที่ควบคุมได้ยาก ดังนั้นการตรวจสอบการใช้งานอินเทอร์เน็ตที่ผิดวัตถุประสงค์ขององค์กรจึงมีความจำเป็นเพิ่มมากขึ้น และในปัจจุบันยังไม่มีโปรแกรมที่ใช้งานในลักษณะนี้โดยเฉพาะเจาะจง จึงทำให้เกิดแนวคิดในการสร้างโปรแกรมตรวจสอบและบันทึกพฤติกรรมผู้ใช้งาน เพื่อการตรวจสอบพฤติกรรมการใช้งาน โดยการเก็บข้อมูลในการใช้งานต่างๆผ่านเครือข่าย และเก็บบันทึกการสนทนาของโปรแกรมสนทนา (MSN) ของผู้ใช้งาน โดยสำหรับการเก็บข้อมูลและสถิติทั้งหมด จะมุ่งเน้นไปเพื่อการตรวจสอบดูว่าผู้ใช้งานมีการใช้งานในลักษณะใด มากน้อยอย่างไร เมื่อเทียบกับการใช้งานโดยรวมของระบบ โดยประเด็นสนใจไปในการใช้งานหลักๆตามองค์กร บริษัทต่างๆ เช่น โปรโตคอลเอชทีทีพี (HTTP) MSN และอื่นๆ

ปริญญานิพนธ์นี้จึงมุ่งเน้นศึกษาการใช้งานที่สำคัญของผู้ใช้งานแต่ละรายอย่างชัดเจน เพื่อเป็นการตรวจดูพฤติกรรม จึงได้มีการบันทึกการใช้งานบางส่วนลงล็อกไฟล์โดยละเอียด แยกตามไอพีแอดเดรส การใช้งาน และวันที่ เพื่อให้สามารถตรวจสอบย้อนหลังได้ตลอดเวลา และนอกจากนี้ยังเน้นให้ง่ายต่อการมองภาพรวมของการใช้งาน เช่นการแสดงเป็นกราฟการใช้งานซึ่งสามารถเลือกดูตามช่วงเวลาเฉพาะที่เราต้องการศึกษา ในส่วนของโปรแกรมสนทนา (MSN) จะทำการบันทึกคำสนทนาของผู้ใช้ไว้ด้วย

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้จัดทำขึ้นภายใต้วัตถุประสงค์หลัก 3 ประการ ได้แก่

1.) เพื่อศึกษารายละเอียดของโปรโตคอลทีซีพี/ไอพี

2.) เพื่อศึกษาแนวทางการตรวจสอบพฤติกรรมผู้ใช้งานเป็นรายบุคคล

3.) เพื่อสร้างระบบตรวจสอบและบันทึกพฤติกรรมของผู้ใช้งานเป็นรายบุคคลให้ชัดเจนมากยิ่งขึ้น

1.3 ขอบเขตของปริญญานิพนธ์

ขอบเขตของปริญญานิพนธ์นี้ ได้แก่

1.) การทำการดักจับและแสดงผลข้อมูล จะมีขอบเขตตามไอพีที่ได้ทำการตั้งค่าไว้เท่านั้น

2.) การทำงานจะสามารถทำได้ในกลุ่มผู้ใช้งานใน Collision Domain เดียวกันหรือเฉพาะในเครือข่ายที่ใช้อุปกรณ์เน็ตเวิร์คที่สามารถรีไดเร็คทราฟฟิค (traffic) ไปยังแต่ละพอร์ตได้เท่านั้น

3.) การทำงานสามารถแสดงผลได้ละเอียดขึ้นในเฉพาะบางโปรโตคอลที่สนใจเป็นพิเศษ คือ โปรโตคอลเอชทีทีพี (HTTP) และเอ็มเอสเอ็น (MSN) เท่านั้น

4.) การทำงานของโปรแกรมขึ้นอยู่กับประสิทธิภาพเครื่องที่ใช้งานเป็นสำคัญ

5.) การทำงานของการดักจับการสนทนาของโปรแกรมสนทนา (MSN) ใช้ได้เฉพาะภาษาอังกฤษเท่านั้น

1.4 ขั้นตอนการดำเนินงาน

1) ศึกษารายละเอียดเกี่ยวกับโปรโตคอลทีซีพี/ไอพี เบื้องต้น
2) ศึกษารายละเอียดการใช้งานไลบรารีที่มีความสำคัญต่อการทำงาน
3) ศึกษาเกี่ยวกับกระบวนการดักจับแพ็กเก็ต และเก็บรักษาแพ็กเก็ต
4) ออกแบบการตรวจจับและการแสดงผลของข้อมูลที่ดักจับมา
5) ออกแบบการทำงานและบันทึกพฤติกรรมการใช้งาน
6) ออกแบบวิธีการจัดเก็บล็อกไฟล์ (Log File)
7) พัฒนาระบบตรวจสอบ บันทึกพฤติกรรม และแสดงผลการใช้งาน
8) ทดสอบและปรับปรุงระบบตรวจสอบและบันทึกพฤติกรรมการใช้งาน

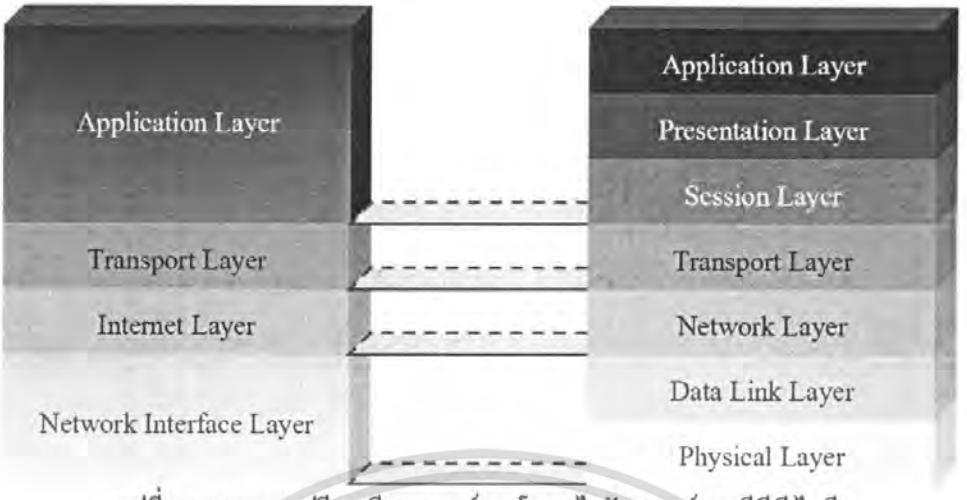
บทที่ 2
โปรโตคอลทีซีพี / ไอพี

2.1 ความเป็นมาของโพรโตคอลทีซีพี/ไอพี

ทีซีพี/ไอพี เป็นมาตรฐานการรับส่งข้อมูลระหว่างคอมพิวเตอร์สองระบบที่มีขึ้นเมื่อกระทรวงกลาโหมสหรัฐฯ หรือ Department Of Defense (DOD) ทำการทดลองในปี ค.ศ.1969 เชื่อมโยงคอมพิวเตอร์ทางทหารของแต่ละหน่วย ซึ่งเป็นคอมพิวเตอร์ต่างชนิดกันให้สามารถติดต่อรับส่งข้อมูลกันได้ โครงการนี้มีชื่อว่า Advanced Research Projects Agency Network หรือ ARPANET ซอฟต์แวร์ที่ใช้ควบคุมการรับส่งข้อมูลของ ARPANET ประกอบด้วยส่วนหลักๆ 2 ส่วน คือ ทีซีพี (Transmission Control Protocol หรือ TCP) และ ไอพี (Internet Protocol หรือ IP) ซึ่ง ทีซีพี มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างคอมพิวเตอร์ผู้รับและผู้ส่ง ให้ได้รับข้อมูลถูกต้องครบถ้วน ส่วนไอพีจะมีหน้าที่เลือกเส้นทางที่ใช้รับส่งข้อมูลผ่านระบบเครือข่าย และตรวจสอบที่แอดเดรสของผู้รับ เรียกว่าไอพีแอดเดรส (IP Address) ต่อมาในปี ค.ศ.1983 ทีซีพี/ไอพี ถูกกำหนดให้เป็นมาตรฐานการรับส่งข้อมูลของกระทรวงกลาโหมสหรัฐฯ และได้รวมเป็นส่วนหนึ่งของระบบปฏิบัติการยูนิกซ์ ส่งผลให้มีการใช้งานกันอย่างกว้างขวาง ในปัจจุบันใช้งานอยู่ในแทบทุกเครือข่าย ไม่ว่าจะเป็นเครือข่ายเฉพาะที่หรือเครือข่ายในบริเวณกว้างทีซีพี/ไอพีเชื่อมกลุ่มเครือข่ายย่อยเข้าด้วยกันเป็นเครือข่ายขนาดใหญ่ หรือ อินเทอร์เน็ต (Internet)

2.2 การเชื่อมต่อของโพรโตคอลทีซีพี/ไอพี (TCP/IP Linking)

ทีซีพี/ไอพี (TCP/IP หรือ Transmission Control Protocol/Internet Protocol) เป็นโพรโตคอลในการสื่อสารในระบบอินเทอร์เน็ตและอินทราเน็ต การทำงานของทีซีพี/ไอพีสามารถเปรียบเทียบกับโมเดลอ้างอิงโอเอสไอ (Open System Interconnection Reference Model: OSI) ตามมาตรฐานไอเอสโอ (International Organization for Standardization: ISO) ได้ดังรูปที่ 2-1



รูปที่ 2.1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี

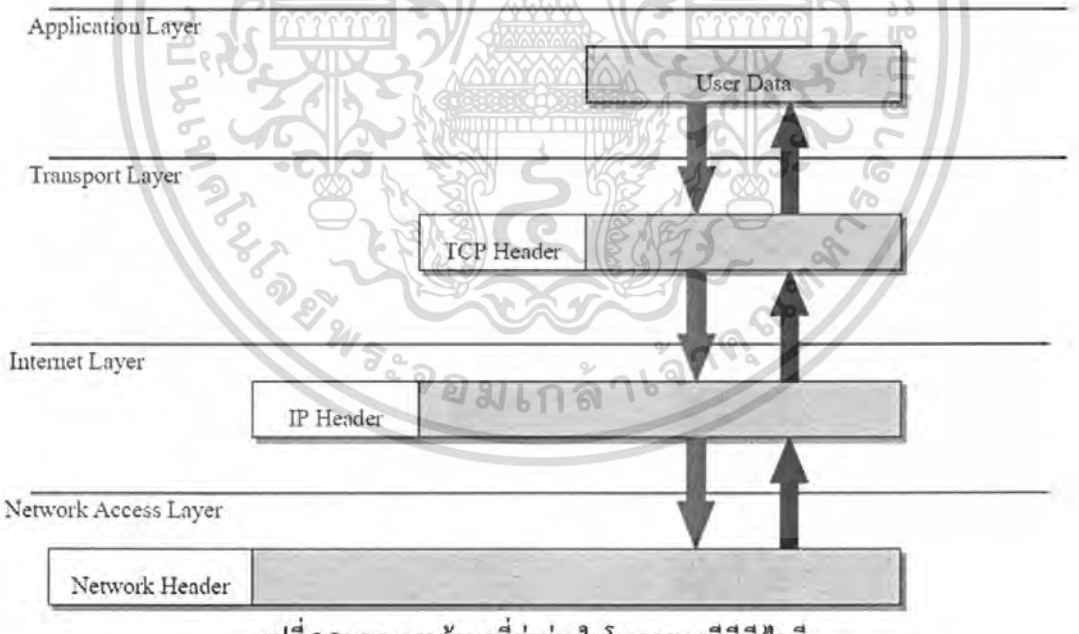
ในแต่ละระดับชั้นของทีซีพี/ไอพีมีการทำงานที่แตกต่างกัน ตั้งแต่การติดต่อกับแอพพลิเคชันจนกระทั่งแปลงเป็นสัญญาณส่งไปตามสายสัญญาณ ซึ่งการทำงานในแต่ละระดับชั้นของทีซีพี/ไอพี มีดังตารางที่ 2.1

ตารางที่ 2.1 การทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี

ชื่อระดับชั้น	หน้าที่
1. ชั้นแอพพลิเคชัน (Application Layer)	ชั้นนี้รองรับการทำงานของแอพพลิเคชันต่างๆ ที่ทำงานเป็นโพรเซสอยู่ในเครื่องต้นทางและปลายทาง โดยจัดการเชื่อมต่อระหว่างโพรเซส หรือแอพพลิเคชันที่อยู่ต่างเครื่องกัน โดยการทำงานของแอพพลิเคชันต่างๆมีการติดต่อกันตามแต่ละโพรโตคอลเฉพาะแล้วแต่แอพพลิเคชันที่ใช้งานซึ่งจะขอบริการจากชั้นทรานสปอร์ตอีกทีหนึ่ง
2. ชั้นทรานสปอร์ต (Transport Layer)	มีการสร้างการเชื่อมต่อกันระหว่างแอพพลิเคชันแบบ end-to-end โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต (port) หรือซ็อกเก็ต (Socket) ในชั้นนี้มีบริการหลักอยู่ 2 แบบ คือ Connection Oriented โดยเรียกผ่านโพรโตคอลทีซีพี (TCP: Transmission Control Protocol) และConnectionless ซึ่งเรียกผ่านโพรโตคอลยูดีพี (UDP: User Datagram Protocol) ซึ่งกล่าวถึงในหัวข้อถัดไป

ตารางที่ 2.1 (ต่อ) การทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี

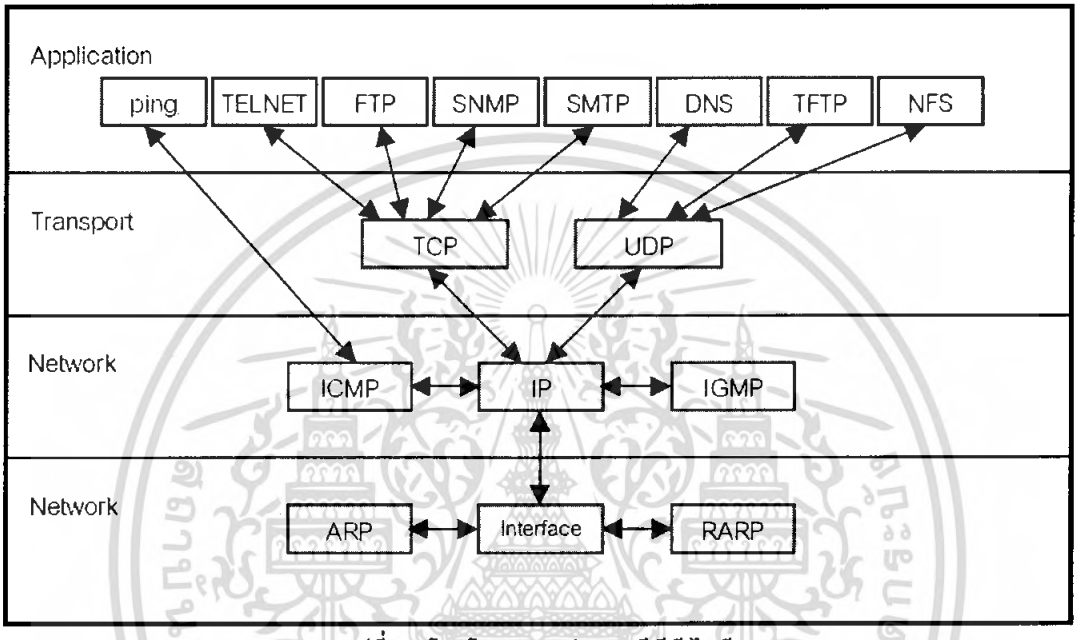
ชื่อระดับชั้น	หน้าที่
3. ชั้นอินเทอร์เน็ต (Internet Layer)	ชั้นนี้มีหน้านี้ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโพรโตคอลที่ทำงาน เป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆ ในอินเทอร์เน็ตคือ ไอพี (Internet Protocol: IP) ซึ่งกล่าวถึงในหัวข้อถัดไป นอกจากนี้ในชั้นนี้ยังมีโพรโตคอลทำงานอยู่ด้วยอีก 2 ชนิด คือ ไอซีเอ็มพี (Internet Control Message Protocol: ICMP) และเออาร์พี (Address Resolution Protocol: ARP)
4. ชั้นเน็ตเวิร์กอินเตอร์เฟส (Network Interface Layer)	ทำหน้าที่ในการแปลงข้อมูลให้อยู่ในรูปที่เหมาะสมกับเครือข่ายแต่ละแบบ ซึ่งแตกต่างกันออกไป และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่าย



รูปที่ 2.2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี

2.3 โพรโตคอลสแต็ก

การทำงานตามโปรแกรมประยุกต์หนึ่งๆไม่ได้ใช้โพรโตคอลพร้อมกันทั้งชุด หากแต่ใช้เพียงโพรโตคอลที่สัมพันธ์กันไปในแต่ละระดับชั้นของแบบอ้างอิง ตัวอย่างเช่นการใช้งานเทลเน็ต (Telnet) จะอาศัยทีซีพีและไอพี ตามลำดับ การซ้อนทับของโพรโตคอลจากระดับชั้นบนไปชั้นล่างเรียกว่าโพรโตคอลสแต็ก (Protocol Stack) ดังรูปที่ 2-3



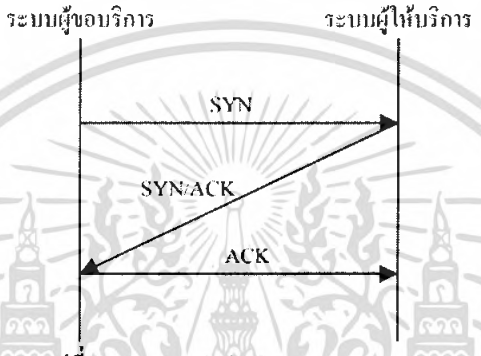
รูปที่ 2.3 โพรโตคอลสแต็กของทีซีพี/ไอพี

ไอพีซึ่งอยู่ในระดับชั้นเน็ตเวิร์คตามรูป เป็นแกนสำคัญของ โพรโตคอลสแต็ก เนื่องจากทั้งทีซีพี และ ยูดีพี ต้องใช้ไอพีเพื่อเลือกเส้นทางส่งแพ็กเก็ต ในระดับชั้นเน็ตเวิร์คยังมีไอซีเอ็มพีสนับสนุนการทำงานของไอพีเพื่อรายงานข้อผิดปกติที่เกิดขึ้นเนื่องจากการส่งแพ็กเก็ต และมีไอจีเอ็มพีดูแลการจัดกลุ่มโฮสต์ในเครือข่ายมัลติคาสต์ ระดับชั้นทรานสปอร์ตมี 2 โพรโตคอล ที่สำคัญคือ ทีซีพีและยูดีพี แอพลิเคชันจะเลือกใช้ทีซีพีหรือยูดีพีตามลักษณะงาน โพรโตคอลระดับล่างถัดจากไอพีได้แก่ โพรโตคอล ระดับเน็ตเวิร์คอินเตอร์เฟสซึ่งกำหนดการทำงานตามเทคโนโลยีเครือข่ายที่ใช้งาน ในระดับชั้นนี้มี โพรโตคอลในชุดของ ทีซีพี/ไอพี ทำหน้าที่สนับสนุนการทำงานอยู่สอง โพรโตคอล คือ เออาร์พี และ อาร์เออาร์พี ทั้งสองโพรโตคอลทำหน้าที่แปลงค่าระหว่างแอดเดรสไอพี กับ ฮาร์ดแวร์แอดเดรส

ในชุดโพรโตคอลทีซีพี/ไอพีนี้ มีโพรโตคอลหลักที่ขอกล่าวถึง 5 โพรโตคอล ได้แก่ โพรโตคอลทีซีพี โพรโตคอลยูดีพี ซึ่งทำงานในชั้นทรานสปอร์ต และโพรโตคอลไอพี โพรโตคอลเออาร์พีโพรโตคอลไอซีเอ็มพี ซึ่งทำงานในชั้นอินเทอร์เน็ต โดยมีรายละเอียดดังต่อไปนี้

2.4 โพรโตคอลทีซีพี (TCP : Transmission Control Protocol)

การทำงานที่สำคัญอย่างหนึ่งของโพรโตคอลทีซีพี คือ การทำ "3-way Handshake" ซึ่งเป็นกระบวนการเริ่มต้นในการสร้างการเชื่อมต่อในชั้นทรานสปอร์ต กล่าวคือ ในการติดต่อกันระหว่างระบบในเครือข่ายต้องมีการสร้างการเชื่อมต่อไปยังระบบที่ให้บริการก่อน โดยผู้ขอบริการส่งสัญญาณ SYN เพื่อขอบริการ จากนั้นผู้ให้บริการจะส่งสัญญาณ ACK เพื่อตอบรับการเชื่อมต่อที่ร้องขอมาจึงสามารถรับส่งข้อมูลกันได้ ดังรูปที่ 2-4



รูปที่ 2.4 แสดงการทำ 3-way Handshake

การเชื่อมต่อแบบ 3-way handshake นี้ เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่งและฝ่ายรับและการกำหนดค่าเริ่มต้นของพารามิเตอร์ต่างๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ 3-way handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับและส่งข้อมูลซึ่งกันและกันได้

ดังนั้นโพรโตคอลทีซีพีจึงเป็นโพรโตคอลที่มีการรับส่งข้อมูลแบบ "Connection Oriented" ทำให้การทำงานของทีซีพีมีความน่าเชื่อถือมากขึ้น หน้าที่การทำงานของทีซีพีในการรับส่งข้อมูลมีหน้าที่หลัก 6 ข้อคือ

1. ควบคุมการรับส่งข้อมูล (Basic Data Transfer)
2. ความน่าเชื่อถือในการรับส่งข้อมูล (Reliability)
3. ควบคุมการไหลของข้อมูล (Flow Control)
4. การทำมัลติเพล็ก (Multiplexing)
5. ควบคุมการเชื่อมต่อ (Connection)
6. ความปลอดภัยในการรับส่งข้อมูล (Security)

ส่วนประกอบของทีซีพีเฮดเดอร์

1. Source Port : เป็นหมายเลขพอร์ตของบริการที่เครื่องต้นทาง

2. Destination Port : เป็นหมายเลขพอร์ตของบริการเครื่องปลายทาง

3. Sequence Number : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลของเครื่องที่ต้องการขอส่งข้อมูล

4. Acknowledgement Number : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลที่ฝั่งรับข้อมูลปกติ ค่าของ Acknowledgement Number มีค่าเท่ากับ Sequence Number (ของอีกฝั่งหนึ่ง) + 1 เสมอ

5. Data Offset : เป็นตัวบอกค่าออฟเซตของข้อมูล เพราะทีซีพีนั้นไม่มีการกำหนดความยาวที่แน่นอนของข้อมูล จึงต้องมีออฟเซตเป็นตัวบอก

6. Flag : เป็นบิตที่บอกชนิดของข้อมูล ได้แก่

- URG : Urgent Pointer Field Significant - แสดง Urgent Pointer
- ACK : Acknowledgement Field Significant – แสดงการ Acknowledgement
- PSH : Push Function
- RST : Reset The Connection - แสดงเมื่อรีเซ็ทการเชื่อมต่อ
- SYN : Synchronize Sequence Number - หมายเลขแพ็กเก็ตที่ส่งแบบซิงโครนัส
- FIN : No more data from sender - แสดงว่าไม่มีข้อมูลที่ส่งจากผู้ส่งแล้ว

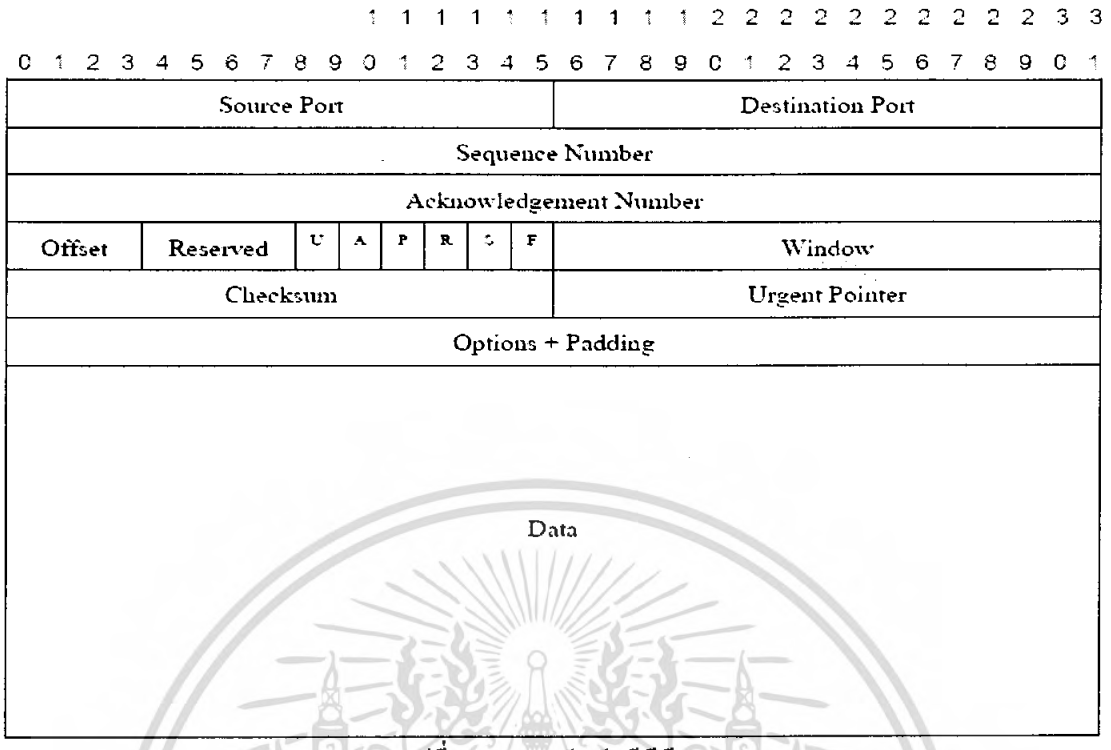
7. Window : เป็นเลขบอกจำนวนของอ็อกเต็ต (octet) ของข้อมูล จัดการในส่วนของ end-to-end flow control

8. Checksum : เป็นส่วนที่ตรวจสอบความถูกต้องของข้อมูล

9. Urgent Pointer : เป็นตัวชี้ตำแหน่งของ Urgent Data

10. Option and Padding : เป็นตัวบอกออปชันของโพรเซสที่ใช้ทีซีพี

11. Data : เนื้อข้อมูลที่ต้องการสื่อสาร มีขนาดได้ไม่ต่ำกว่า 5 32-บิตเวิร์ด (6 บิตแรกสงวนไว้และกำหนดให้เป็นศูนย์)



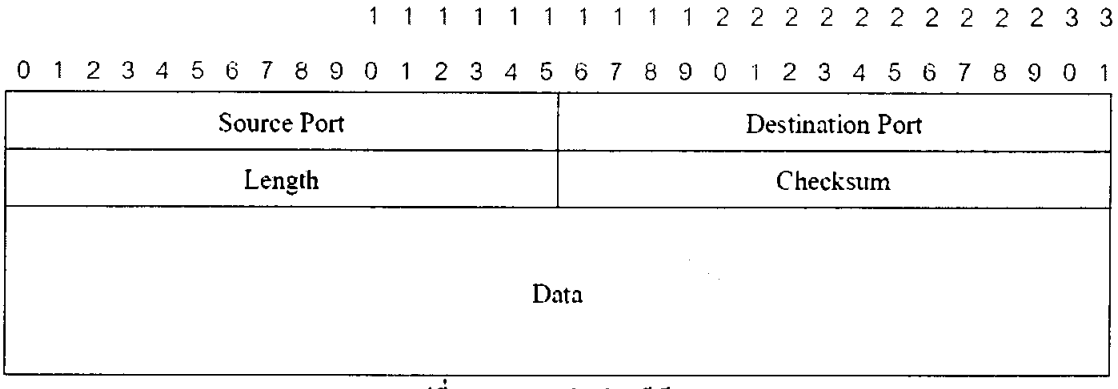
รูปที่ 2.5 แสดงแพ็กเก็ตทีซีพี

2.5 โพรโตคอลยูดีพี (UDP: User Datagram Protocol)

โพรโตคอลยูดีพีเป็นโพรโตคอลในการติดต่อสื่อสารในชั้นทรานสปอร์ต (Transport Layer) การทำงานคล้ายกับทีซีพีมาก คือ จัดการเกี่ยวกับการสื่อสารระหว่างเครื่อง แต่เป็นแบบ Connectionless คือ ทั้งฝ่ายส่งและฝ่ายรับไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน โดยไม่ต้องมีการแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโพรโตคอลทีซีพี และไม่มีการส่งสัญญาณตรวจสอบว่าข้อมูลถึงเครื่องปลายทางอย่างถูกต้องครบถ้วนในการส่งข้อมูลแต่ละครั้ง จึงไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล

<u>ส่วนประกอบของ UDP Frame</u>

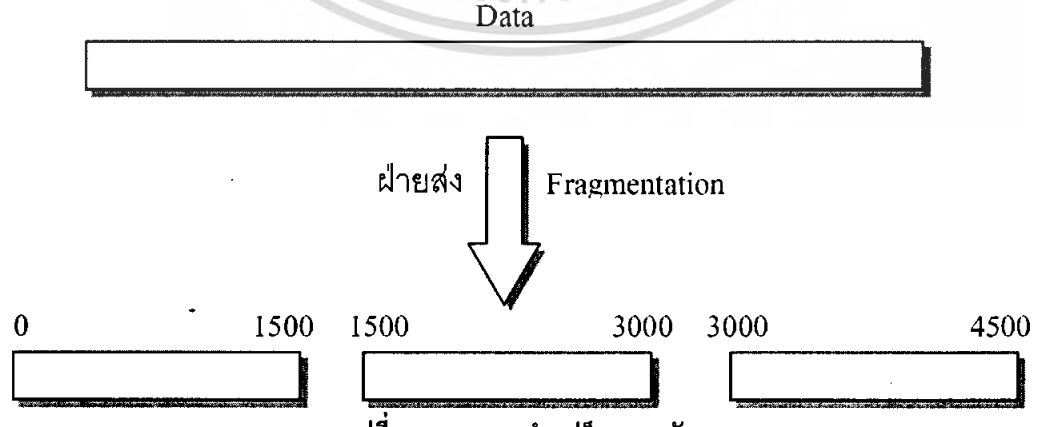
1. Source Port : เป็นค่าตัวเลข 16 บิต บอกพอร์ตของบริการที่เครื่องต้นทาง
2. Destination Port : เป็นค่าตัวเลข 16 บิต บอกพอร์ตของบริการที่เครื่องปลายทาง
3. Length : เป็นค่าตัวเลข 16 บิต บอกความยาวของข้อมูล
4. Checksum : เป็นค่าตัวเลข 16 บิต ตรวจสอบความถูกต้องของข้อมูลที่ส่ง



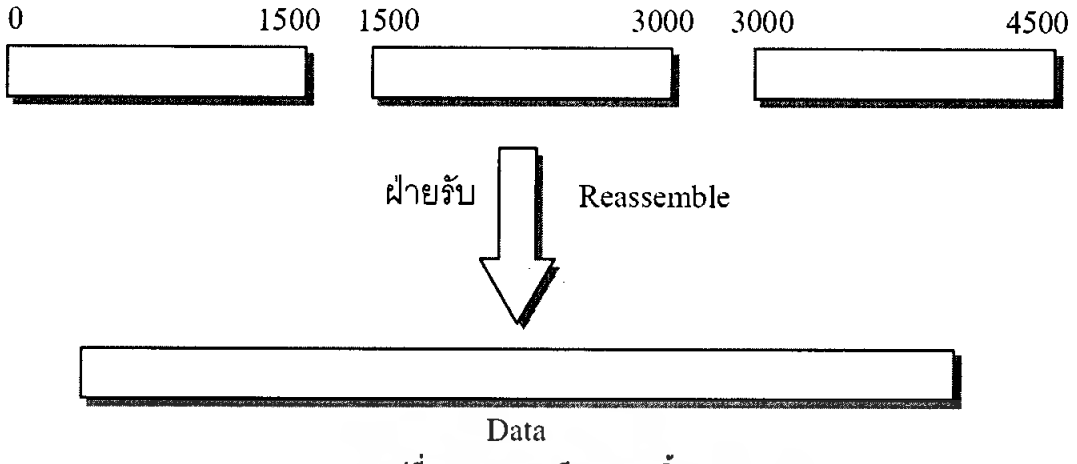
รูปที่ 2.6 แสดงแพ็กเก็ตยูดีพี

2.6 โพรโตคอลไอพี (IP: Internet Protocol)

โพรโตคอลไอพีเป็นโพรโตคอลที่จัดการเกี่ยวกับแอดเดรสของแต่ละแพ็กเก็ต เพื่อให้ส่งแพ็กเก็ตต่างๆ ไปยังเป้าหมายได้ถูกต้อง การทำงานของไอพีเป็นเพียงการส่งข้อมูลไปยังเครื่องเป้าหมายเท่านั้นไม่มีการส่งสัญญาณขอบริการ หรือสัญญาณให้บริการระหว่างกันเหมือนทีซีพี เรียกว่าการเชื่อมต่อแบบConnectionless ซึ่งระบบทั้งสองตั้งสมมติฐานว่าการเชื่อมต่อระหว่างกันไม่มีความผิดพลาดเกิดขึ้นแน่เนื่องจากมาตรฐานในเครือข่ายมีหลากหลาย ขนาดของแพ็กเก็ตในแต่ละมาตรฐานจึงมีความแตกต่างกันออกไป ทำให้การส่งข้อมูลระหว่างอุปกรณ์ในเครือข่ายนั้นอาจมีการแบ่งข้อมูลออกเป็นแพ็กเก็ตย่อยๆ ในระหว่างการส่ง เรียกว่า การทำแฟร็กเมนเตชัน (Fragmentation) เช่น แพ็กเก็ตของ FDDI มีขนาด 4,500 ไบต์ หากเครื่องปลายอยู่ในเครือข่าย Ethernet ซึ่งมีขนาดของแพ็กเก็ตสูงสุดเพียง 1,500 ไบต์ดังนั้นการส่งแพ็กเก็ตไปยังเครื่องปลายทางจึงต้องมีการแบ่งเป็นแพ็กเก็ตย่อย และเมื่อแพ็กเก็ตย่อยมาถึงเครื่องเป้าหมายก็จะมารวมกันเป็นแพ็กเก็ตเดิมที่มีขนาด 4,500 ไบต์อีกครั้ง เรียกการรวมกันนี้ว่า การรีแอสเซม-เบิ้ล (Reassemble) ซึ่งทำให้ได้ข้อมูลเหมือนที่ส่งมาจากเครื่องต้นทาง



รูปที่ 2.7 แสดงการทำแฟร็กเมนเตชัน



รูปที่ 2.8 แสดงการรีแอสเซมเบิ้ล

<u>ส่วนประกอบของแพ็กเก็ตไอพี</u>

1. version : เป็นค่าตัวเลข 4 บิต บอกเวอร์ชันของมาตรฐานไอพีที่ใช้ โดยปกติมีค่าเป็น 4 ซึ่งหมายถึง IPv4

2. Internet Header Length (IHL) : เป็นตัวบอกความยาวเฮดเดอร์ของไอพี

3. Type of Service : เป็นส่วนที่บอกการทำงานของแพ็กเก็ตที่ส่งว่าทำหน้าที่อะไร มีทั้งหมด 8 บิต โดย

Bit 0-2 : บอกรายละเอียดการทำงานของแพ็กเก็ตนั้นๆ

111 - Network Control

110 - Internetwork Control

101 - CRITIC / ECP

100 - Flash Override

011 - Flash

010 - Immediate

001 - Priority

000 - Routine

Bit 3 : บอกถึงลักษณะของดีเลย์

0 = Normal Delay - มีดีเลย์ปกติ

1 = Low Delay - มีดีเลย์ต่ำ

Bit 4 : บอกถึงประเภทของทรูพุต

0 = Normal Throughput – มีทรูพุตปกติ

1= High Throughput – มีทรูพุตสูง

Bit 5 : บอกถึงประเภทของความน่าเชื่อถือ

0 = Normal Reliability - มีความน่าเชื่อถือพอประมาณ

1 = High Reliability - มีความน่าเชื่อถือสูง

Bit 6-7 : กันไว้ใช้ในอนาคต

4. Total Length : มีขนาด 16 บิต บอกถึงความยาวในดาต้าแกรมของไอพี

5. Identification field : เป็นตัวเลข 16 บิต เป็นค่าประจำตัวของไอพีนั้น โดยโฮสต์ที่ส่งเป็นผู้กำหนด และเพิ่มค่าขึ้นหนึ่งเมื่อมีการส่งดาต้าแกรมของไอพีใหม่ ซึ่งใช้ในการประกอบกลับ

6. Flag : เป็นตัวเลข 3 bit บอกลักษณะของแพ็กเก็ตว่ามีการแฟร็กเมนต์หรือไม่

Bit 0 : สงวนไว้ ปกติเป็น 0

Bit 1 : 0 = บอกว่าแพ็กเก็ตมีการแตกแพ็กเก็ตย่อย

1 = บอกว่าแพ็กเก็ตไม่มีการแตกแพ็กเก็ตย่อย

Bit 2 : 0 = บอกว่าแพ็กเก็ตนั้นเป็นแพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

1 = บอกว่าแพ็กเก็ตนั้นยังไม่ใช่แพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

7. Fragment Offset : เป็นค่าตัวเลข 13 บิต บอกออปเซตของแฟร็กเมนต์เมื่อเทียบในดาต้าแกรม

8. Time To Live (TTL) : เป็นตัวเลข 8 บิต บอกช่วงเวลาของแพ็กเก็ตที่ยังอยู่ในเครือข่ายได้ โดยกำหนดค่าเป็นจำนวนเราท์เตอร์สูงสุดที่ดาต้าแกรมผ่านได้ ซึ่งโดยทั่วไปทีค่าระหว่าง 32 ถึง 64 และลดค่าลงเรื่อยๆ เมื่อผ่านเราเตอร์ เพื่อเป็นการป้องกันแพ็กเก็ตล้นเครือข่าย

9. Protocol : เป็นตัวเลข 8 bit บอกถึงโพรโตคอลที่อยู่เหนือขึ้นไป ว่าเป็นโพรโตคอลระดับสูงกว่าประเภทใด

10. Header Checksum : เป็นค่าตัวเลข 32 บิต ใช้ตรวจสอบความถูกต้องของเฮดเดอร์

11. Source Address : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องต้นทาง

12. Destination Address : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องปลายทาง

1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 Bits

Ver	IHL	Type of Service	Total Length	
Identifier			Flags	Fragment
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options + Padding				
Data				

รูปที่ 2.9 แสดงแพ็กเก็ตไอพี

บทที่ 3
โปรโตคอลเอ็มเอสเอ็นเอ็มเอส (MSNMS)

3.1 ความเป็นมาของเอ็มเอสเอ็นแมสเซ็นเจอร์

เอ็มเอสเอ็นแมสเซ็นเจอร์ (MSN Messenger) หรือที่เรียกกันว่าเอ็มเอสเอ็น (MSN) คือ โปรแกรมส่งข้อความข้าม ระบบเน็ทเวิร์ค แบบทันทีทันใด เรียกว่า IM (Instant Messenger) ถ้าเคย ใช้โปรแกรมไอซีคิว (ICQ) ไออาร์ซี (IRC) หรือ เพิร์ช (Pirch) ก็เข้าข่ายเป็นโปรแกรมประเภท เดียวกัน เหตุผลที่เอ็มเอสเอ็น (MSN) ได้รับความนิยมเป็นอย่างมาก เนื่องมาจากความง่ายของการ ใช้งาน เช่นเราเพียงแค่มีอีเมลล์ของฮอทเมลล์ (Hotmail) หรือ MSN ก็สามารถเข้าใช้งานเอ็มเอสเอ็น (MSN) ได้ทันที และยังมีการผนวกเข้ากับอีเมลล์ด้วย โดยที่ เมื่อใดก็ตามที่มีเมลล์ เข้ามาใหม่ โปรแกรมเอ็มเอสเอ็น (MSN) จะแจ้งให้ทราบทันที นอกจากนั้น ความเร็วของการ รับและส่ง ข้อความระหว่างกัน สามารถทำได้อย่างรวดเร็ว มีหน้าตาโปรแกรมที่สวยงาม และมีเวอร์ชั่นใหม่ ออกมาสม่ำเสมอ

รูปที่ 3.1 โปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์(หน้าต่างหลัก)

รูปที่ 3.2 โปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์(หน้าต่างย่อย)

จากรูปแสดงตัวอย่างโปรแกรมเอ็มเอสเอ็นซึ่งจะแบ่งออกเป็นสองส่วนย่อยๆ ดังนี้

1. หน้าต่างหลัก ที่หน้าต่างนี้จะแสดงชื่อของเพื่อนๆ ทั้งผู้ที่ออนไลน์ (Online) และผู้ที่อ็อฟไลน์ (offline) เวลาที่ต้องการพูดคุยกับเพื่อนคนไหน ก็สามารถดับเบิ้ลคลิ๊ก ที่ชื่อแล้ว หน้าต่างอีกอันจะแสดงขึ้นมา จากนั้นจะสามารถพิมพ์ข้อความส่งให้เพื่อนได้ทันที

2. หน้าต่างย่อย เป็นหน้าต่างที่เราสนทนากับเพื่อน ที่หน้าต่างนี้สามารถพิมพ์ข้อความโต้ตอบกับเพื่อนได้ทันที และยังสามารถให้โปรแกรมเอ็มเอสเอ็น (MSN) แสดงรูปภาพที่เราต้องการได้อีกด้วย โดยที่ทางฝั่งเพื่อนจะเห็นรูปดังกล่าวเช่นกัน อีกทั้งยังสามารถส่งไอคอนแสดงอารมณ์ต่างๆ เพื่อสื่ออารมณ์ และเพิ่มความสนุกสนานในการพูดคุยได้อีกด้วย

3.2 โปรโตคอลที่เกี่ยวข้อง

โปรโตคอลต่างๆที่เกี่ยวข้องกับการใช้งานโปรแกรมเอ็นเอสเอ็นแมสเซ็นเจอร์(MSN Messenger) ทั้งหมดมีดังนี้

- โปรโตคอลทีซีพี (TCP--Transmission Control Protocol) ใช้ในการทำ Three way handshake และใช้ในการรับส่งไฟล์ระหว่างโปรแกรมเอ็มเอสเอ็น

- โปรโตคอลเอชทีทีพี (HTTP--Hyper Text Transer Protocal) ใช้งานผ่านพอร์ทหมายเลข 80 เพื่อใช้ในการติดต่อกับหน้าเว็บไซต์ของเอ็มเอสเอ็นแมสเซ็นเจอร์

- โปรโตคอลเอ็มเอสเอ็นเอ็มเอส (MSNMS--Microsoft Network Messenger Service) ใช้งานผ่านพอร์ทหมายเลข 1863 ซึ่งโปรโตคอลนี้เป็นโปรโตคอลของโปรแกรมเอ็มเอสเอ็นซึ่งจะมี

คำสั่งต่างๆ เอาไว้ใช้งาน เช่น Command USR ซึ่งเป็นคำสั่งที่ใช้ในการยืนยันตัวตน (Authentication) สำหรับการลงชื่อเข้าใช้งานโปรแกรม (Sign in)

3.3 คำสั่งที่ใช้ในโปรโตคอลเอ็มเอสเอ็นเอ็มเอส (MSNMS)

3.3.1 Logon/Dispatch server

เป็นคำสั่งที่ใช้สำหรับส่งและรับค่าจาก server เมื่อได้ทำการติดต่อไปที่เครื่อง server ซึ่งจะมี command ดังต่อไปนี้

1). VER - Protocol version เป็นคำสั่งที่ใช้สำหรับตรวจสอบ version ของ protocol

2). CVR - Sends version information เป็นคำสั่งที่ใช้สำหรับตรวจสอบ version และข้อมูลของโปรแกรมที่ใช้งานอยู่

3). USR - Authentication command เป็นคำสั่งที่ใช้สำหรับพิสูจน์ตัวตน คือเป็นการตรวจสอบว่า username และ password ถูกต้องหรือเปล่า

4). XFR - Redirection to Notification server เป็นคำสั่งที่ใช้สำหรับ Redirection ไปที่ Notification server

3.3.2 Contact list/Settings/Initial synchronization commands

เป็นคำสั่งที่เกี่ยวกับ Contact list/Settings/Initial ต่างๆ ที่ได้ทำเอาไว้ ซึ่งจะมีคำสั่งดังต่อไปนี้

1). BLP - Initial settings

2). BPR - Initial settings download

3). GTC - Initial contact list/settings download เป็นคำสั่งที่ใช้สำหรับ download เกี่ยวกับ contact list และ setting ต่างๆ

4). ILN - Initial contact presence notification

5). LSG - Initial contact list download – Groups เป็นคำสั่งที่ใช้สำหรับ download group ที่ได้ทำการสร้างไว้ในเอ็มเอสเอ็น (MSN)

6). LST - Initial contact list download – Contacts เป็นคำสั่งที่ใช้สำหรับ download contacts ที่มีในเอ็มเอสเอ็น (MSN)ของผู้ใช้งาน

7). MSG - Initial profile download เป็นคำสั่งที่ใช้สำหรับ download profile ของผู้ใช้ที่ได้ทำการสร้าง profile ไว้

8). PRP - Initial settings download - Mobile settings and display name เป็นคำสั่งที่ใช้สำหรับ download display name ที่ได้ทำการตั้งไว้

3.3.3 Standard send/receive commands

เป็นคำสั่งที่เกี่ยวกับการส่งและรับค่าต่างๆ เช่นการ Add user, Create groups ซึ่งจะมีคำสั่งดังต่อไปนี้

1). ADC - Add users to your contact lists เป็นคำสั่งที่ใช้สำหรับการเพิ่มผู้ที่จะติดต่อด้วย (Add user) เข้ามาใน Contact list ของผู้ใช้

2). ADD - Add users to your contact lists (deprecated as of MSNP11) Add เป็นคำสั่งที่ใช้สำหรับเพิ่มผู้ที่จะติดต่อด้วย (Add user) เข้ามาใน Contact list ของผู้ใช้ แต่จะใช้สำหรับ Protocol MSNP11

3). ADG - Create groups เป็นคำสั่งที่ใช้สำหรับสร้างกลุ่ม (Create groups) ขึ้นมา

4). CHG - Change client's online status เป็นคำสั่งที่ใช้สำหรับเปลี่ยนสถานะของผู้ใช้งานไปเป็นสถานะต่างๆ

5). GCF– Unknown เป็นคำสั่งที่ใช้สำหรับบอกว่าการทำงานที่จับได้ โปรแกรมไม่รู้จัก

6). OUT - Gracefully logout เป็นคำสั่งที่ใช้สำหรับการเลิกใช้งานโปรแกรม (Sign out)

7). PNG - Client ping เป็นคำสั่งที่ใช้สำหรับส่งสัญญาณ (Ping) ไปที่เครื่อง server เพื่อตรวจสอบว่าเครื่อง server ยังทำงานอยู่

8). QNG - Server response to PNG เป็นคำสั่งที่ใช้สำหรับส่งสัญญาณ (Response) กลับไปที่เครื่อง client เมื่อเครื่อง client ทำการ Ping มา เพื่อที่จะบอกว่าเครื่อง server ยังทำงานอยู่

9). QRY - Response to CHL by client เป็นคำสั่งที่ใช้สำหรับส่งสัญญาณ (Response) ไปที่เครื่อง server

10). SBS – Unknown เป็นคำสั่งที่ใช้สำหรับบอกว่าการทำงานที่จับได้ โปรแกรมไม่รู้จัก

11). SYN - Begin synchronization/download contact list เป็นคำสั่งที่ใช้สำหรับทำการ synchronization/download contact list

12). REA - Change display name เป็นคำสั่งที่ใช้สำหรับแสดงชื่อหัวข้อ (display name) ที่ผู้ใช้ได้ทำการตั้งชื่อไว้ REG - Rename groups เป็นคำสั่งที่ใช้สำหรับเปลี่ยนชื่อกลุ่ม (Group)

13). REM - Remove contacts เป็นคำสั่งที่ใช้สำหรับลบผู้ติดต่อออกจากโปรแกรม ของผู้ใช้ (Remove contacts) RMG - Remove groups เป็นคำสั่งที่ใช้สำหรับลบกลุ่มออกจากโปรแกรม (Remove groups)

14). XFR - Opens new chat session on switchboard server เป็นคำสั่งที่ใช้สำหรับสร้างเส้นทางการพูดคุยบน switchboard ของ server

15). UBX - Inform you with a user PSM/Media เป็นคำสั่งที่ใช้สำหรับแสดงข้อความที่อยู่ด้านล่าง display name หรือใช้แสดงรายชื่อเพลง

3.3.4 Asynchronous commands

I). CHL - Client challenge (see MSNP8:Challenges) เป็นคำสั่งที่ใช้สำหรับเครื่อง client ส่ง challenge

2). FLN - Principal signed off เป็นคำสั่งที่ใช้สำหรับออกจากโปรแกรม (sign off)

3). NLN - Principal changed presence/signed on เป็นคำสั่งที่ใช้สำหรับแสดงว่ามีผู้ใช้ใน Contact list ออนไลน์ เข้ามา

4). RNG - Client invited to chat session เป็นคำสั่งที่ใช้สำหรับเชิญผู้ใช้ใน Contact list เข้ามาร่วมคุย

3.3.5 Switchboard

เป็นคำสั่งที่เกี่ยวกับการส่งและรับ หลังจากที่ติดต่อกับ switchboard ได้แล้ว ซึ่งจะมีคำสั่งดังต่อไปนี้

1). ANS - Log in to switchboard chat session using invitation เป็นคำสั่งที่ใช้สำหรับเข้าไปใน switchboard โดยการเชิญชวน

2). IRO - Defines which principals are in the current chat session เป็นคำสั่งที่ใช้สำหรับกำหนดความสำคัญของการสนทนา

3). USR - Log in to switchboard chat session after requesting session from NS เป็นคำสั่งที่ใช้สำหรับเข้าไปใน switchboard หลังจากที่ได้รับการร้องขอจาก NS

4). CAL - Invite a user to a chat session เป็นคำสั่งที่ใช้สำหรับเชิญผู้ใช้ใน Contact list เข้ามาร่วมคุย

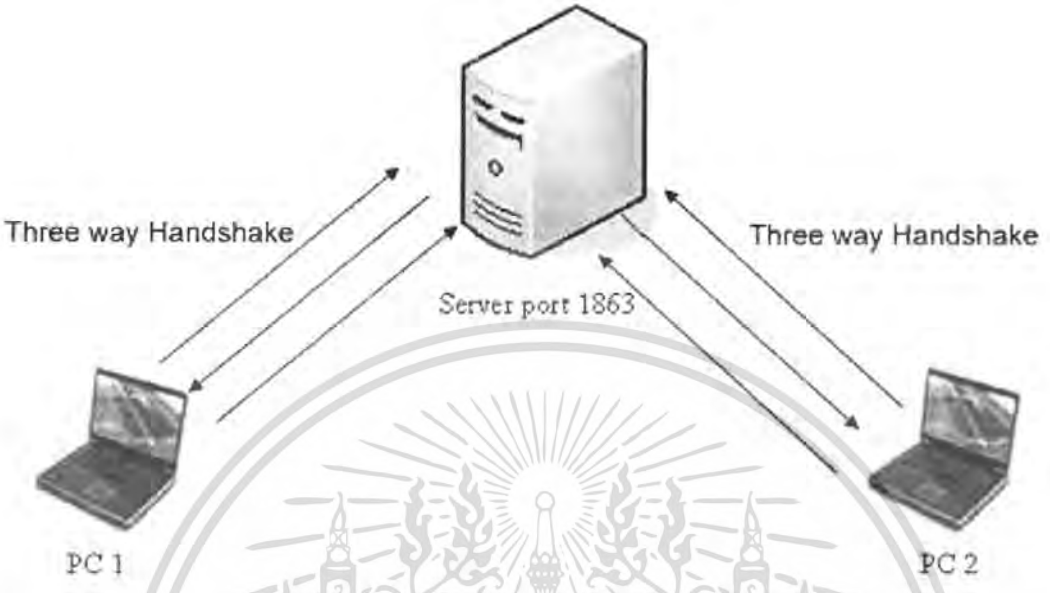
5). JOI - Response to CAL, when user connected successfully เป็นคำสั่งที่ใช้บอกว่าการ เชิญผู้ใช้ใน Contact list เข้ามาร่วมคุย ทำได้สำเร็จ

6). MSG - Used to send and receive messages in the chat session เป็นคำสั่งที่ใช้สำหรับการรับและการส่งข้อความ

7). BYE - Contact has left conversation เป็นคำสั่งที่ใช้สำหรับออกจากการสนทนา

8). OUT - Gracefully leave switchboard chat session เป็นคำสั่งที่ใช้สำหรับออกจาก switchboard

3.4 ขั้นตอนในการติดต่อไปที่เซิร์ฟเวอร์ (Server)



รูปที่ 3.3 แสดงขั้นตอนในการเชื่อมต่อไปยังเซิร์ฟเวอร์

เมื่อมีการใช้งานโปรแกรมเอ็มเอสเอ็น โปรแกรมจะทำการติดต่อไปยังเซิร์ฟเวอร์ เพื่อทำการลงชื่อใช้งาน ขั้นตอนในการเข้าใช้งานโปรแกรมเอ็มเอสเอ็น มีดังนี้

1). เมื่อทำการเปิดโปรแกรมเอ็มเอสเอ็น(MSN) ขึ้นมาแล้วทำการใส่บัญชีผู้ใช้งาน (username) และรหัสผ่าน (password) โปรแกรมจะทำการติดต่อไปที่เซิร์ฟเวอร์ด้วยการทำ Three way handshake

2). เมื่อทำการติดต่อได้สำเร็จแล้ว ก็จะทำการตรวจสอบเวอร์ชั่นของโปรโตคอลที่ใช้อยู่ว่าเป็นเวอร์ชั่นใด ถ้าเป็น MSN เวอร์ชั่น 7.5 จะใช้โปรโตคอล MSNP10, MSNP11, MSNP12 ส่วนถ้าเป็น MSN เวอร์ชั่น 8.1 จะใช้โปรโตคอล MSNP13, MSNP14, MSNP15

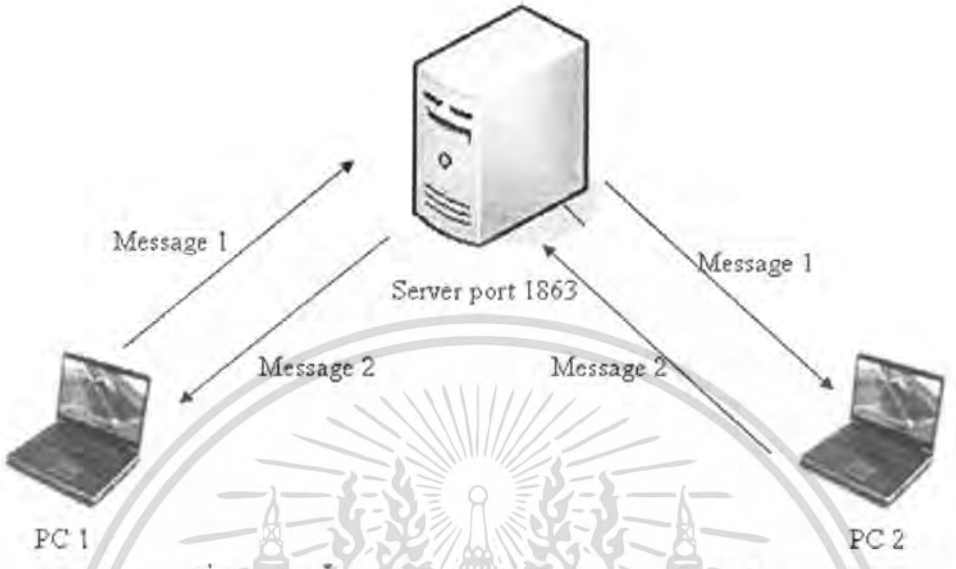
3). หลังจากนั้นจะทำการตรวจสอบเวอร์ชั่นของโปรแกรมที่ใช้งานอยู่

4). ว่าเป็นเวอร์ชั่น ไหน อย่างเช่น MSN 7.5, MSN 8.1

5). ที่เครื่องเซิร์ฟเวอร์จะเอาบัญชีผู้ใช้งาน (username) และรหัสผ่าน (password) ที่ได้ใส่ไว้ตอน แรก มาทำการยืนยันตัวตน (Authentication) ว่ามีสิทธิ์เข้ามาใช้งานหรือไม่

6). เมื่อผ่านขั้นตอนของการยืนยันตัวตน (Authentication) แล้วก็จะสามารถสนทนา หรือส่งไฟล์ได้

3.5 ขั้นตอนในการสนทนา



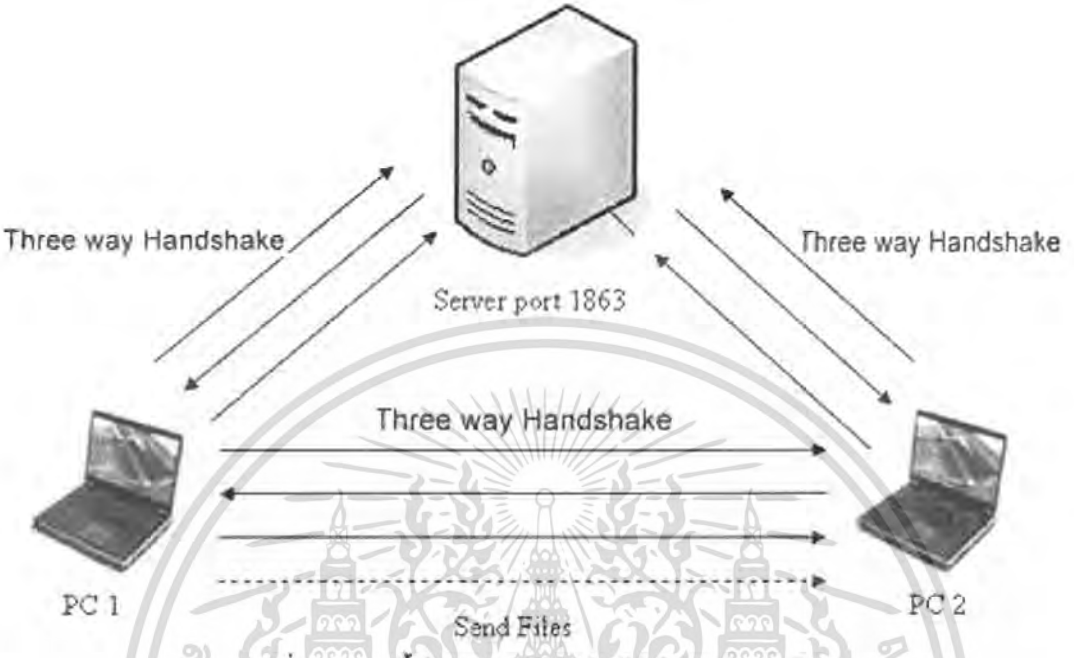
รูปที่ 3.4 แสดงขั้นตอนในการสนทนาผ่านโปรแกรมเอ็มเอสเอ็น

หลังจากผ่านขั้นตอนการยืนยันตัวตนกับเซิร์ฟเวอร์แล้ว จะสามารถเริ่มการสนทนาได้ โดยการสนทนาจะเป็นการสนทนาผ่านเซิร์ฟเวอร์อีกต่อหนึ่ง ขั้นตอนในการสนทนามีดังนี้

1). หลังจากผ่านขั้นตอนของการติดต่อกับเซิร์ฟเวอร์แล้ว เมื่อเครื่อง PC1 ต้องการสนทนากับเครื่อง PC2 เครื่อง PC1 จะส่งข้อความไปที่เซิร์ฟเวอร์และ เซิร์ฟเวอร์ก็จะส่งต่อข้อความของเครื่อง PC1 ไปให้เครื่อง PC2

2). ถ้าเครื่อง PC2 ต้องการสนทนากับเครื่อง PC1 ก็จะต้องทำแบบเดียว กัน คือ เครื่อง PC2 จะส่งข้อความไปที่เซิร์ฟเวอร์และ เซิร์ฟเวอร์ก็จะส่งต่อข้อความของเครื่อง PC2 ไปให้เครื่อง PC1 เช่นเดียวกัน

3.6 ขั้นตอนในการส่งไฟล์



รูปที่ 3.5 แสดงขั้นตอนการส่งไฟล์ผ่านโปรแกรมเอ็มเอสเอ็น

หลังจากผ่านขั้นตอนการยืนยันตัวตนกับเซิร์ฟเวอร์แล้ว จะสามารถเริ่มการสนทนา หรือการส่งไฟล์ได้ ในที่นี้จะกล่าวถึงการส่งไฟล์ผ่านโปรแกรมเอ็มเอสเอ็น โดยการส่งไฟล์นั้นจะเป็นแบบ Peer-To-Peer ซึ่งต่างกับการสนทนาที่ต้องผ่านเซิร์ฟเวอร์ก่อน ขั้นตอนในการส่งไฟล์มีดังนี้

1). ถ้าเครื่อง PC1 ต้องการส่งไฟล์ไปให้กับเครื่อง PC2 ในกรณีที่ไม่ได้ออนไลน์อยู่ จะต้องออนไลน์เข้ามาและ จะต้องผ่านการติดต่อกับเซิร์ฟเวอร์ก่อน โดยการทำ Three way handshake ตรวจสอบเวอร์ชั่นโปรโตคอล ตรวจสอบเวอร์ชั่นของโปรแกรม และทำการยืนยันตัวตน (Authentication) แต่ถ้าออนไลน์อยู่แล้ว ก็ไม่ต้องทำขั้นตอนนี้

2). เมื่อผ่านขั้นตอนแรกมาแล้ว เครื่อง PC1 จะต้องทำ Three way handshake กับเครื่อง PC2 อีกครั้งก่อนที่จะทำการส่งไฟล์

3). เมื่อผ่านขั้นตอนที่สองมาแล้ว เครื่อง PC1 ก็ตจะทำการส่งไฟล์ให้ กับเครื่อง PC2 โดยการส่งไฟล์นี้จะไม่ส่งผ่านเซิร์ฟเวอร์เหมือนกับการส่ง ข้อความสนทนา แต่จะส่งกันโดยตรงเป็นแบบ Peer-To-Peer ระหว่าง เครื่อง PC1 และ PC2

บทที่ 4
โปรโตคอลเอชทีทีพี (HTTP)

4.1 โปรโตคอลเอชทีทีพี (HTTP – Hypertext Transfer Protocol)

Hypertext Transfer Protocol (HTTP) คือโปรโตคอลที่เป็นรากฐานของเวิลด์ไวด์เว็บ (World Wide Web : WWW) และสามารถนำมาประยุกต์ใช้ได้กับแอพพลิเคชันจำพวกไคลเอ็นต์/เซิร์ฟเวอร์ใดๆ ที่มีลักษณะของความเป็นไฮเปอร์เท็กซ์ ชื่อของโปรโตคอลนี้อาจทำให้เกิดความเข้าใจผิดว่า HTTP เป็นโปรโตคอลที่ใช้ในการโอนย้ายไฮเปอร์เท็กซ์ แต่ความจริงแล้วมันเป็นโปรโตคอลสำหรับส่งผ่านข้อมูล ซึ่งมีความสามารถเพียงพอที่จะก่อให้เกิดลักษณะของไฮเปอร์เท็กซ์ ข้อมูลที่ถูกเคลื่อนย้ายผ่านโปรโตคอลนี้อาจเป็นเพียงข้อความธรรมดาๆ ,ไฮเปอร์เท็กซ์, ภาพ, เสียง หรือข้อมูลอื่นใดที่ช่วยให้ผู้ใช้เข้าถึงอินเทอร์เน็ตได้ ในการใช้งานเครือข่ายคอมพิวเตอร์ทั่วไปหรือในเครือข่ายอินเตอร์เน็ตก็ตามจะมีการส่งผ่านข้อมูลไป มาระหว่างเครื่องคอมพิวเตอร์หรือข้ามเครือข่ายออกไป ระบบคอมพิวเตอร์ที่เชื่อมต่ออยู่ในแต่ละเครือข่ายอาจจะใช้ฮาร์ดแวร์และซอฟต์แวร์ที่เหมือนกันหรือแตกต่างกันได้ ดังนั้นการที่จะทำให้สามารถส่งผ่านข้อมูลถึงกันและตีความได้อย่างได้อย่างถูกต้องจะต้องมีการกำหนดกลไกในการสื่อสารกันเสียก่อน เรียกว่าจะต้องกำหนดระเบียบวิธีในการติดต่อกันให้ตรงกัน เปรียบเหมือนกับการสื่อสารกันของมนุษย์เรา ถ้าเราต้องการจะติดต่อกับผู้คนต่างเชื้อชาติต่างภาษากันให้เข้าใจกันได้ถูกต้องตรงกันก็จะต้องตกลงกำหนดกันเสียก่อนว่า จะติดต่อกันอย่างไร ด้วยภาษาใดที่จะเข้าใจกันได้เช่น ปัจจุบันมีการใช้ภาษาอังกฤษเป็นภาษากลางในการติดต่อกันมาก ทำให้เราพูดได้ว่า ภาษาอังกฤษเปรียบเสมือนเป็นภาษามาตรฐานในการสื่อสารของมนุษย์ได้ ถ้าพูดในแง่การสื่อสารข้อมูล เราก็พูดได้ว่า ภาษาอังกฤษเป็นโปโตคอลในการสื่อสารของมนุษย์ที่มีการใช้งานอย่างแพร่หลาย เช่นเดียวกันกับโปรโตคอล TCP/IP เป็นโปรโตคอลหลักที่ใช้ในการสื่อสารข้อมูลในเครือข่ายอินเตอร์เน็ต

จากที่กล่าวมาพอจะพูดได้ว่าโปรโตคอล(Protocol) คือระเบียบวิธีที่กำหนดขึ้นสำหรับสื่อสารข้อมูล ให้สามารถส่งผ่านข้อมูลไปยังปลายทางได้อย่างถูกต้อง ซึ่งโปรโตคอลได้กำหนดสิ่งที่จำเป็นจะต้องมีดังต่อไปนี้

1). ชนิดของการตรวจสอบข้อผิดพลาด ที่จะใช้
2). วิธีบีบอัดข้อมูล (ถ้ามี)
3). วิธีที่ระบบที่ส่งข้อมูลรับรู้ว่ามันได้ส่งข้อมูลเสร็จแล้ว
4). วิธีที่ระบบที่รับข้อมูลรับรู้ว่ามันได้รับข้อมูลแล้ว

ปัจจุบันโปรโตคอลในการสื่อสารข้อมูลมีอยู่หลายโปรโตคอลนอกเหนือจาก TCP/IP คล้ายกับภาษาต่างๆในโลกนี้ ที่นอกจากภาษาอังกฤษแล้วยังมีภาษาจีน ญี่ปุ่น ฝรั่งเศส เยอรมัน และอื่นอีกมากมาย ในด้านของโปรโตคอลสื่อสารข้อมูลก็เช่นกัน ซึ่งได้มีการออกแบบโปรโตคอลอื่นๆขึ้นมาช้างานอีกมาก เช่น โปรโตคอลIPX/SPX, โปรโตคอล NetBIOS และ โปรโตคอล AppleTalk เป็นต้น

เนื้อหาของปริญญานิพนธ์ฉบับนี้จะเริ่มต้นจากการอธิบายแนวความคิดและการดำเนินงานของโปรโตคอลเอชทีทีพี (HTTP) หลังจากนั้นจึงพิจารณาเข้าไปในรายละเอียดบางอย่าง โดยอิงกับเอชทีทีพีเวอร์ชัน 1.1 (RFC 2068) ทั้งนี้เราได้สรุปความหมายของคำศัพท์ต่างๆ ที่ปรากฏอยู่ในข้อกำหนดของโปรโตคอลเอชทีทีพี ไว้ดังนี้

ตารางที่ 4.1 สรุปความหมายของคำศัพท์ที่เกี่ยวข้องกับโปรโตคอลเอชทีทีพี (HTTP)

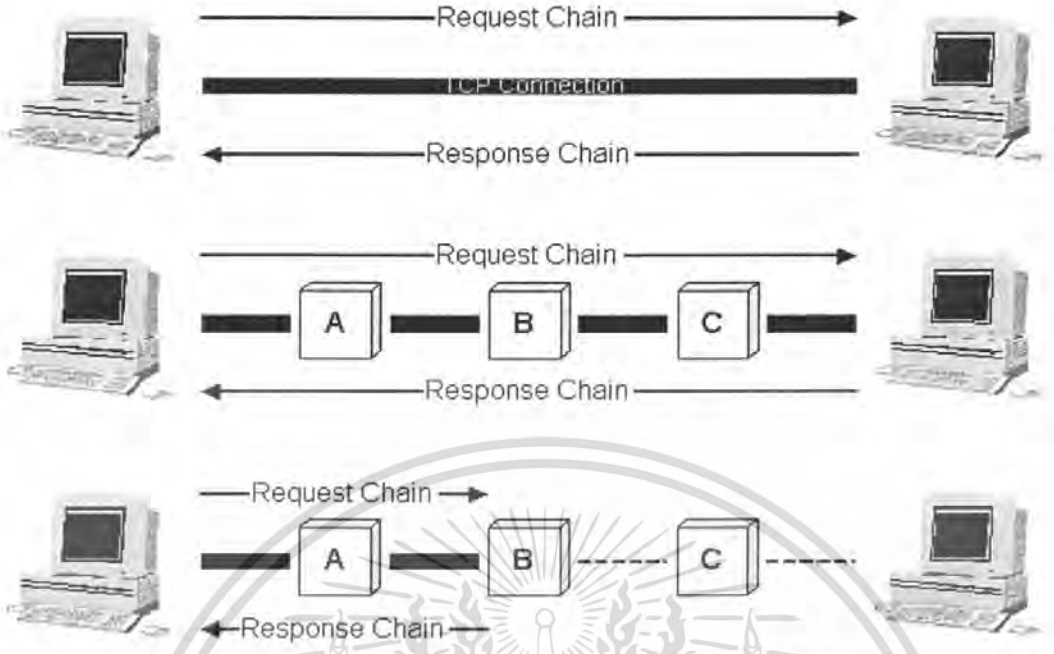
คำศัพท์	ความหมาย
แคช (Cache)	
ไคลเอ็นต์ (Client)	โปรแกรมที่สร้างการเชื่อมต่อขึ้นเพื่อส่งคำร้องขอ (Request) ไป
การเชื่อมต่อ (Connection)	วงจรเสมือนที่ถูกสร้างขึ้นระหว่างโปรแกรม 2 โปรแกรม เพื่อให้โปรแกรมสามารถสื่อสารกันได้
เอ็นติตี้ (Entity)	
เกตเวย์ (Gateway)	
แมสเซส (Message)	
ออริจิ้นเซิร์ฟเวอร์ (Origin Server)	เซิร์ฟเวอร์ที่มีรีซอร์สอยู่ หรือรีซอร์สถูกสร้างขึ้น
พร็อกซี่ (Proxy)	
รีซอร์ส (Resource)	ออบเจ็กต์ของข้อมูลหรือบริการ ที่สามารถอ้างถึงได้ด้วย URI
เซิร์ฟเวอร์ (Server)	โปรแกรมที่ยอมรับการเชื่อมต่อเพื่อให้บริการต่อคำร้องขอต่างๆ โดยส่งคำตอบสนอง (Response) กลับไป
ทันเนล (Tunnel)	
ยูสเซอร์เอเจนต์ (User Agent)	ไคลเอ็นต์ที่เริ่มต้นส่งคำร้องขอ โดยทั่วไปหมายถึง บราวเซอร์, เอดิเตอร์, สไปเดอร์ หรือเครื่องมือใดๆ ของผู้ใช้ปลายทาง

4.2 ภาพโดยรวมของโปรโตคอลเอชทีทีพี (HTTP)

เอชทีทีพี (HTTP) เป็นโปรโตคอลแบบไคลเอ็นต์/เซิร์ฟเวอร์ในลักษณะ transaction-oriented คือมีการติดต่อระหว่างโปรแกรม 2 โปรแกรม ซึ่งโดยทั่วไปได้แก่เว็บบราวเซอร์และเว็บเซิร์ฟเวอร์ เพื่อให้มีความน่าเชื่อถือเอชทีทีพี (HTTP) จึงใช้ประโยชน์จากโปรโตคอลทีซีพี (TCP) แต่ถึงกระนั้น เอชทีทีพี (HTTP) ก็เป็นโปรโตคอลที่ "ปราศจากสถานะ" กล่าวคือ การติดต่อในแต่ละครั้งเป็นอิสระต่อกัน โดยการเชื่อมต่อระหว่างไคลเอ็นต์และเซิร์ฟเวอร์จะถูกสร้างขึ้นมาใหม่สำหรับการติดต่อในแต่ละครั้ง และถูกตัดขาดจากกันทันทีที่การติดต่อเสร็จสิ้นสมบูรณ์ ถึงแม้ว่าข้อกำหนดของเอชทีทีพี (HTTP) จะไม่ได้ระบุความสัมพันธ์ในแบบหนึ่งต่อหนึ่งระหว่างการติดต่อและช่วงเวลาของการเชื่อมต่อเช่นนี้ไว้ก็ตามที

คุณสมบัติ "ปราศจากสถานะ" ดังกล่าวของโปรโตคอลเอชทีทีพี (HTTP) นี้เหมาะสมต่อการนำมาประยุกต์ใช้เป็นอย่างยิ่ง การใช้งานเว็บบราวเซอร์นั้นโดยปกติเกี่ยวข้องกับการรับเอากลุ่มของเว็บเพจและเอกสารเข้ามา ซึ่งการดำเนินการตรงนี้เกิดขึ้นรวดเร็วมาก โดยเว็บเพจและเอกสารเหล่านี้อาจมาจากเซิร์ฟเวอร์ที่แตกต่างกันไป

คุณลักษณะที่สำคัญอีกประการหนึ่งของโปรโตคอล HTTP ก็คือ ความยืดหยุ่นในแง่ของรูปแบบที่มันสามารถจัดการได้ เมื่อไคลเอ็นต์ส่งคำร้องขอไปยังเซิร์ฟเวอร์ ไคลเอ็นต์อาจระบุรายการของรูปแบบต่างๆ ที่มันสามารถจัดการได้ไปให้เซิร์ฟเวอร์ด้วย ฝ่ายเซิร์ฟเวอร์เองก็จะตอบสนองกลับมาด้วยรูปแบบที่เหมาะสม ยกตัวอย่างเช่น บราวเซอร์ lynx ซึ่งเป็นบราวเซอร์ที่ทำงานภายใต้เท็กซ์โหมดของยูนิกซ์นั้นไม่สามารถจัดการกับรูปภาพได้ เว็บเซิร์ฟเวอร์จึงไม่จำเป็นต้องส่งรูปภาพใดๆ ที่ปรากฏอยู่บนเว็บเพจไปให้ การเตรียมการเช่นนี้ป้องกันมิให้เกิดการส่งข้อมูลที่ไม่จำเป็น และยังเป็นหลักสำคัญสำหรับการเพิ่มเติมรูปแบบตามข้อกำหนดที่จะถูกสร้างขึ้นใหม่ให้เป็นมาตรฐานในอนาคตอีกด้วย

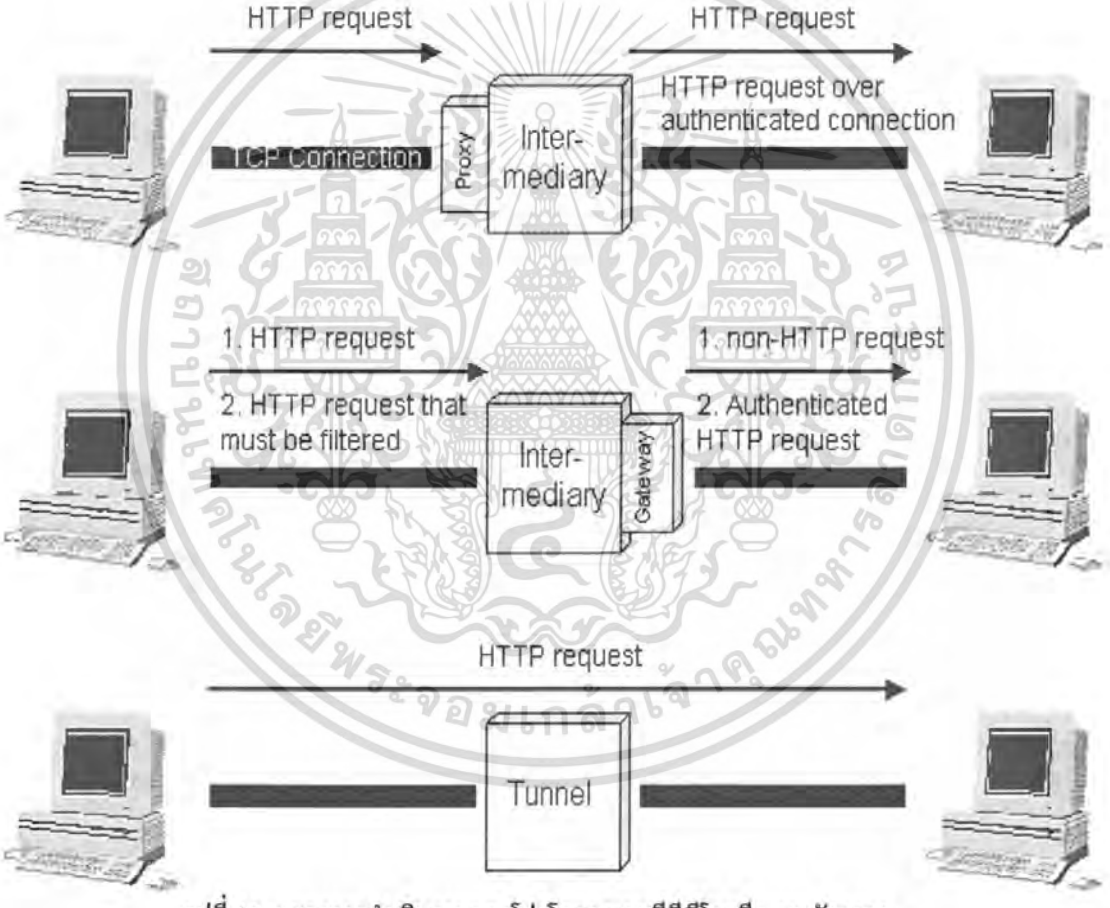


รูปที่ 4.1 แสดงตัวอย่างการดำเนินงานของโปรโตคอลเอชทีทีพี

แสดงให้เห็นตัวอย่างการดำเนินงานของ HTTP กรณีที่ง่ายที่สุดกรณีหนึ่งเกิดขึ้นเมื่อยูสเซอร์เอเจนต์สร้างการเชื่อมต่อโดยตรงกับออริจิ้นเซิร์ฟเวอร์ ยูสเซอร์เอเจนต์ (User Agent) คือโปรแกรมที่เริ่มต้นส่งคำร้องขอ เช่น เว็บบราวเซอร์ที่รันอยู่ทางฝั่งผู้ใช้ เป็นต้น ส่วน ออริจิ้นเซิร์ฟเวอร์ (Origin Server) คือเซิร์ฟเวอร์ที่มีรีซอร์สอยู่ เช่น เว็บเซิร์ฟเวอร์ที่เก็บโฮมเพจไว้ เป็นต้น ในกรณีนี้ไคลเอ็นต์เป็นฝ่ายเปิดการเชื่อมต่อแบบทีซีพี (TCP) ระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ จากนั้นไคลเอ็นต์จะส่งคำร้องขอไปยังเซิร์ฟเวอร์ โดยคำร้องขอประกอบด้วยคำสั่งอย่างใดอย่างหนึ่ง หรือที่เรียกว่าเมธ็อด (method), URL, แมสเซสที่มีลักษณะคล้าย MIME (Multipurpose Internet Mail Extensions) ซึ่งบรรจุพารามิเตอร์ต่างๆ ของการร้องขอในครั้งนั้น, ข้อมูลเกี่ยวกับไคลเอ็นต์ และข้อมูลเพิ่มเติมอื่นๆ เมื่อเซิร์ฟเวอร์ได้รับคำร้องขอแล้ว มันจะพยายามปฏิบัติตามสิ่งที่ไคลเอ็นต์ร้องขอมา แล้วส่งคำตอบสนองกลับคืนไป คำตอบสนองประกอบด้วยข้อมูลสถานะ, รหัสความสำเร็จหรือความผิดพลาด, แมสเซสที่มีลักษณะคล้าย MIME ซึ่งบรรจุข้อมูลต่างๆ ของเซิร์ฟเวอร์, ข้อมูลที่เกี่ยวข้องกับการตอบสนอง และในบางครั้งจะรวมถึงส่วนที่เป็นเนื้อหา (body content) ด้วย หลังจากนั้นการเชื่อมต่อแบบทีซีพี (TCP) จะถูกปิดลงไป

ตัวอย่างถัดไปของ รูปที่ 4.1 แสดงให้เห็นกรณีที่ไม่มีการเชื่อมต่อแบบทีซีพี (TCP) โดยตรงระหว่างยูสเซอร์เอเจนต์กับออริจิ้นเซิร์ฟเวอร์ แต่จะมีระบบตัวกลาง (intermediate system) ที่มีการเชื่อมต่อแบบทีซีพี (TCP) ระหว่างระบบที่อยู่ติดกันในทางตรรกะ ระบบตัวกลางเหล่านี้ทำหน้าที่เสมือนตัวถ่ายทอด คำร้องขอที่สร้างขึ้นโดยไคลเอ็นต์จะถูกส่งผ่านไปยังระบบตัวกลางต่างๆ จนกระทั่งถึงเซิร์ฟเวอร์ในท้ายที่สุด และคำตอบสนองจากเซิร์ฟเวอร์ก็จะถูกส่งผ่านระบบตัวกลางกลับมายังไคลเอ็นต์ด้วยเช่นกัน

ระบบตัวกลางที่ถูกระบุไว้ในข้อกำหนดของเอชทีทีพี (HTTP) จำแนกออกเป็น 3 รูปแบบ คือ พร็อกซี่, เกตเวย์ และทันเนล ดังรายละเอียดในรูปที่ 4.2



รูปที่ 4.2 แสดงการดำเนินงานของโปรโตคอลเอชทีทีพีโดยมีระบบตัวกลาง

4.3 คำสั่งของโปรโตคอลเอชทีทีพี (HTTP)

โปรโตคอลเอชทีทีพีมีคำสั่งต่าง ๆ ไม่มากนัก เพื่อให้สามารถใช้งานได้อย่างสะดวกและรวดเร็ว โดยมีคำสั่งที่ใช้งานแพร่หลายอยู่เพียง 3 คำสั่ง คือ GET , HEAD และ POST ส่วนคำสั่งอื่นอีก 4 คำสั่งคือ PUT DELETE LINK และ UNLINK มีให้ใช้งานเช่นกัน แต่ไม่เป็นที่นิยมมากนัก รายละเอียดของคำสั่งของโปรโตคอลเอชทีทีพีมีดังนี้

ตารางที่ 4.2 แสดงคำสั่งของโปรโตคอลเอชทีทีพี (HTTP)

คำสั่ง	รายละเอียด
GET	ให้อ่านข้อมูลจากเว็บเซิร์ฟเวอร์และส่งไปยังไคลเอนต์โดยมีรูปแบบดังนี้ GET <URL> HTTP/1.0 ตัวอย่างเช่น ต้องการให้เว็บเซิร์ฟเวอร์ส่งไฟล์ sale.html จากโดเมน www.netcorp.com ไปยังไคลเอนต์จะใช้รูปแบบของคำสั่ง GET ดังนี้ GET www.netcorp.com/sale.html/HTTP/1.0 นอกจากนี้คำสั่ง GET ยังสามารถกำหนดเงื่อนไขให้อ่านข้อมูลจากเว็บเซิร์ฟเวอร์ เฉพาะที่มีการเปลี่ยนแปลงแก้ไขได้ด้วย
HEAD	คำสั่งนี้จะทำงานคล้ายกับคำสั่ง GET แต่เว็บเซิร์ฟเวอร์ จะส่งข้อมูลกลับมาให้เฉพาะในรายละเอียดของ metadata หรือข้อมูลในเฮดเดอร์เท่านั้น ส่วนข้อมูลที่เป็น HTML จะไม่ถูกส่งมาด้วย ซึ่งคำสั่ง HEAD นี้ใช้เพื่อทดสอบว่าข้อมูลตาม URL นั้น ๆ มีการเปลี่ยนแปลงหรือไม่เท่านั้น
POST	เป็นคำสั่งที่ตรงข้ามกับคำสั่ง GET และ HEAD โดยทำหน้าที่ส่งข้อมูลจากไคลเอนต์ไปยังเซิร์ฟเวอร์ แต่โดยปกติแล้วจะส่งข้อมูลจากไคลเอนต์ไปยังเซิร์ฟเวอร์นั้นจะไม่ค่อยมีการใช้งาน นอกจากในกรณีที่ HTML ทำงานในลักษณะที่ให้ผู้ใช้กรอกข้อมูลลงตามแบบฟอร์ม (เช่น รายละเอียดส่วนตัวของผู้ใช้งาน) และส่งข่อมูลนี้กลับมาเก็บ
PUT	เป็นคำสั่งที่ทำงานเหมือนกับคำสั่ง POST แต่ไม่เป็นที่นิยม
DELETE	เพื่อให้ไคลเอนต์สั่งเว็บเซิร์ฟเวอร์ลบ URL ที่กำหนดไว้ออกจากเซิร์ฟเวอร์แต่ไม่เป็นที่นิยมใช้มากนัก เนื่องจากเว็บเซิร์ฟเวอร์ทั่วไปมักจะทำงานในแบบอ่านข้อมูลได้เท่านั้น (read-only)
LINK	เป็นคำสั่งที่เชื่อม URL ที่ต้องการไปยังเว็บเซิร์ฟเวอร์อื่น
UNLINK	ยกเลิกคำสั่ง LINK ให้กลับมาใช้เซิร์ฟเวอร์เดิมตามที่กำหนดไว้ใน URL

4.4 สถานะการทำงานของเอชทีทีพี (HTTP)

โปรโตคอลเอชทีทีพีได้กำหนดรหัสแสดงสถานะการทำงานของโปรโตคอล โดยแบ่งกลุ่มของรหัสสถานะออกไว้เป็น 5 กลุ่มคือ

ตารางที่ 4.3 แสดงสถานการณ์ทำงานของโปรโตคอลเอชทีทีพี (HTTP)

รหัสสถานะ	ประเภท	รายละเอียด
100 –199	Informational	เป็นรหัสสถานะกลุ่มที่เปิดให้โปรแกรมประยุกต์ต่าง ๆ กำหนดใช้งานได้เอง
200-299	Successful	กลุ่มรหัสที่แสดงว่าการทำงานเสร็จแล้ว
300-399	Redirection	กลุ่มรหัสนี้จะใช้ภายในโปรโตคอล HTTP เอง โดยเป็นการทำงานที่ต่อเนื่องมาจากโปรเซสก่อนหน้านี้ ซึ่งไคลเอนต์เป็นผู้สั่งงาน
400-499	Client Error	ใช้แสดงปัญหาที่เกิดขึ้นกับไคลเอนต์
500-599	Server Error	ใช้แสดงปัญหาที่เกิดขึ้นกับเซิร์ฟเวอร์

รหัสแสดงสถานะในแต่ละตัว จะนำหน้าด้วยตัวเลข 3 หลักและตามด้วยตัวอักษร ซึ่งรหัสในกลุ่ม 100-199 จะเปิดกว้างให้ผู้พัฒนาโปแกรมประยุกต์สามารถกำหนดค่าขึ้นมาใช้งานเอง ส่วนรายละเอียดของรหัสในกลุ่มอื่น ๆ มีดังนี้

ตารางที่ 4.4 แสดงรหัสสถานะของโปรโตคอลเอชทีทีพี (HTTP)

รหัสสถานะ	รายละเอียด
200 OK	การทำงานสำเร็จเรียบร้อย
201 Created	คำสั่ง POST ทำงานเสร็จสมบูรณ์
202 Accepted	ได้รับคำสั่งให้ทำงานเรียบร้อย แต่ไม่ต้องมีการตอบกลับ
300 Multiple Choices	ถ้าค้นหาและพบแหล่งข้อมูลที่ต้องการหลายแห่ง เซิร์ฟเวอร์จะตอบกลับไปทั้งหมดเพื่อให้ไคลเอนต์เลือกแหล่งข้อมูลที่ต้องการเองได้
301 Moved Permanently	URL ที่ร้องขอได้ถูกย้ายไปที่อื่นแล้ว ดังนั้นการร้องขอให้งาน URL จะต้องเปลี่ยนเป็นแอดเดรสใหม่
302 Moved Temporarily	URL ที่ร้องขอมาได้ถูกย้ายไปที่อื่นชั่วคราว
304 Not Modify	ใช้แสดงสถานะเมื่อใช้คำสั่ง GET ที่กำหนดเงื่อนไขเฉพาะเว็บไซต์ที่มีการเปลี่ยนแปลง ส่วนเว็บไซต์ที่ไม่มีการเปลี่ยนแปลงจะแสดงด้วยสถานะนี้
400 Bad Request	คำสั่งจากไคลเอนต์ไม่ถูกต้อง
401 Unauthorized	ปฏิเสธการทำงานจากไคลเอนต์ที่ไม่ได้รับอนุญาต
403 Forbidden	เซิร์ฟเวอร์ไม่อนุญาตให้ใช้งาน หรือไคลเอนต์มีสิทธิ ในการใช้งานไม่เพียงพอ
404 Not Found	ไม่พบเซิร์ฟเวอร์ตาม URL ที่กำหนด
500 Internal Server Error	เซิร์ฟเวอร์มีปัญหา
501 Not Implemented	เซิร์ฟเวอร์ไม่รองรับคำสั่งที่สั่งไป
502 Bad Gateway	Proxy Server รับคำสั่งไม่ถูกต้องจากเว็บเซิร์ฟเวอร์
503 Service Unavailable	เซิร์ฟเวอร์กำลังทำงานอื่นอยู่ ไม่สามารถให้บริการได้ในขณะนี้

4.5 พร็อกซี่ (Proxy)

พร็อกซี่ทำงานในฐานะเซิร์ฟเวอร์เมื่อโต้ตอบกับไคลเอ็นต์ และทำงานในฐานะไคลเอ็นต์เมื่อโต้ตอบกับเซิร์ฟเวอร์ การใช้งานพร็อกซี่เกิดขึ้นใน 2 สถานการณ์ต่อไปนี้

4.5.1 ตัวกลางที่ดูแลเรื่องความปลอดภัย (Security Intermediary)

ไคลเอ็นต์และเซิร์ฟเวอร์อาจถูกแยกจากกันด้วยตัวกลางที่คอยดูแลในเรื่องความปลอดภัย อย่างเช่นไฟร์วอลล์ (firewall) โดยมีพร็อกซี่อยู่ทางฝั่งไคลเอ็นต์ของไฟร์วอลล์ โดยทั่วไปแล้วไคลเอ็นต์ถือเป็นส่วนหนึ่งของเน็ตเวิร์กที่ได้รับการปกป้องจากไฟร์วอลล์ ในขณะที่เซิร์ฟเวอร์นั้นถือเป็นส่วนที่อยู่ภายนอก ซึ่งในกรณีนี้เซิร์ฟเวอร์จะต้องพิสูจน์ตนเอง (authenticate) ต่อไฟร์วอลล์เพื่อสร้างการเชื่อมต่อกับพร็อกซี่ขึ้น พร็อกซี่จะได้รับคำตอบสนองก็ต่อเมื่อคำตอบสนองเหล่านั้นผ่านมาจากไฟร์วอลล์แล้วเท่านั้น

4.5.2 เวอร์ชั่นของเอชทีทีพี(HTTP)ที่แตกต่างกัน

เมื่อไคลเอ็นต์และเซิร์ฟเวอร์รันเอชทีทีพี (HTTP) เวอร์ชั่นที่แตกต่างกัน เราสามารถใช้พร็อกซี่เพื่อดำเนินการแปลงสิ่งต่างๆ ที่จำเป็นสำหรับ 2 เวอร์ชันนั้น

โดยสรุปแล้ว พร็อกซี่ก็คือเอเจนต์ที่ทำหน้าที่ส่งต่อ รับคำร้องขอสำหรับออบเจ็กต์ของ URL หนึ่งๆ ปรับปรุงคำร้องขอ และส่งต่อคำร้องขอนั้นไปยังเซิร์ฟเวอร์ที่ระบุไว้ใน URL

4.6 เกตเวย์ (Gateway)

เกตเวย์คือเซิร์ฟเวอร์ที่ปรากฏต่อไคลเอ็นต์ราวกับว่าเป็นออริจินเซิร์ฟเวอร์ โดยปฏิบัติงานอยู่ทางเซิร์ฟเวอร์ซึ่งไม่สามารถสื่อสารกับไคลเอ็นต์ได้โดยตรง ใช้งานใน 2 สถานการณ์ต่อไปนี้

4.6.1 ตัวกลางที่ดูแลเรื่องความปลอดภัย (Security Intermediary)

ไคลเอ็นต์และเซิร์ฟเวอร์อาจถูกแยกจากกันด้วยตัวกลางที่คอยดูแลในเรื่องความปลอดภัย อย่างเช่นไฟร์วอลล์ (firewall) โดยมีเกตเวย์อยู่ทางฝั่งเซิร์ฟเวอร์ของไฟร์วอลล์ ตามปกติเซิร์ฟเวอร์จะถูกเชื่อมต่อจากเน็ตเวิร์กที่ปกป้องด้วยไฟร์วอลล์ โดยไคลเอ็นต์นั้นถือเป็นส่วนที่อยู่ภายนอกของเน็ตเวิร์ก ซึ่งในกรณีนี้ไคลเอ็นต์จะต้องพิสูจน์ตนเองต่อเกตเวย์ หลังจากนั้นจึงจะสามารถส่งผ่านคำร้องขอไปยังเซิร์ฟเวอร์ได้

4.6.2 เซิร์ฟเวอร์ที่ไม่ใช่เอชทีทีพี

เว็บบราวเซอร์มีความสามารถที่จะติดต่อไปยังเซิร์ฟเวอร์ของโปรโตคอลอื่นๆ นอกเหนือจากเอชทีทีพี (HTTP)ได้ เช่น FTP และ Gopher เป็นต้น ซึ่งเกตเวย์เองก็มีความสามารถนี้ โดยเมื่อไคลเอ็นต์ส่งคำร้องขอแบบเอชทีทีพีไปยังเกตเวย์เซิร์ฟเวอร์ เกตเวย์เซิร์ฟเวอร์จะติดต่อไปยังเซิร์ฟเวอร์ FTP หรือ Gopher เพื่อนำผลลัพธ์ที่ต้องการกลับมา ผลลัพธ์เหล่านี้จะถูกปรับเปลี่ยนให้อยู่ในรูปแบบที่เหมาะสมสำหรับเอชทีทีพีแล้วส่งต่อไปยังไคลเอ็นต์

4.7 ทันเนล (Tunnel)

สิ่งที่แตกต่างจากพร็อกซี่และเกตเวย์ก็คือ ทันเนลไม่ได้ดำเนินการใดๆ กับคำร้องและคำตอบสนองของเอชทีทีพี (HTTP)เลย แต่ทันเนลเป็นเพียงจุดส่งต่อที่อยู่ระหว่างการเชื่อมต่อแบบ TCP แมสเซสของเอชทีทีพี (HTTP) จะถูกส่งผ่านไปมาโดยไม่ได้รับการเปลี่ยนแปลงแก้ไขใดๆ ราวกับว่าเป็นการเชื่อมต่อโดยตรงระหว่างยูสเซอร์เอเจนต์และออริจิ้นเซิร์ฟเวอร์ เราใช้ทันเนลเมื่อต้องการให้มีระบบตัวกลางระหว่างไคลเอ็นต์และเซิร์ฟเวอร์โดยที่ระบบนั้นไม่จำเป็นต้องเข้าใจเนื้อหาของแมสเซส ตัวอย่างของทันเนลก็คือ ไฟร์วอลล์ซึ่งไคลเอ็นต์หรือเซิร์ฟเวอร์ภายนอกเน็ตเวิร์กที่ได้รับการปกป้องนั้น สามารถสร้างการเชื่อมต่อและรักษาการเชื่อมต่อเพื่อการติดต่อสื่อสารในแบบเอชทีทีพี (HTTP) แคช (Cache) กลับไปยังรูปที่ 1ส่วนล่างสุดของรูปนั้นแสดงให้เห็นตัวอย่างของการใช้แคช แคชถือเป็นสิ่งอำนวยความสะดวกที่สามารถเก็บคำร้องขอและคำตอบสนองก่อนหน้านี้ไว้เพื่อการจัดการกับคำร้องขอใหม่ หากคำร้องขอที่เข้ามาใหม่เหมือนกับคำร้องขอที่ได้จัดเก็บไว้ในแคชแล้ว แคชจะส่งคำตอบสนองที่ได้จัดเก็บไว้ แทนที่จะต้องไปเข้าถึงรีซอร์สที่ระบุใน URL แคชสามารถทำงานได้ทั้งบนไคลเอ็นต์ เซิร์ฟเวอร์ หรือระบบตัวกลางใดๆ ที่ไม่ใช่ทันเนล ในรูปดังกล่าวนั้น ระบบตัวกลาง B ทำหน้าที่เป็นแคชสำหรับการติดต่อ เพื่อที่คำร้องขอครั้งใหม่จากไคลเอ็นต์จะไม่ต้องเดินทางไปยังออริจิ้นเซิร์ฟเวอร์ แต่จะถูกจัดการโดย B ได้ทันที มิใช่ว่าการติดต่อทุกอย่างจะใช้ประโยชน์จากแคชได้ ทั้งนี้ไคลเอ็นต์หรือเซิร์ฟเวอร์สามารถควบคุมได้ว่าจะให้การติดต่อใดใช้แคช และจำกัดให้ใช้เป็นระยะเวลาเท่าใด

4.8 แมสเซส (Messages)

วิธีที่อธิบายการทำงานของ HTTP ได้ดีที่สุดก็คือการอธิบายถึงแต่ละองค์ประกอบของแมสเซส HTTP โดย HTTP ประกอบด้วยแมสเซส 2 ประเภทคือ คำร้องขอจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์ และคำตอบสนองจากเซิร์ฟเวอร์ไปยังไคลเอ็นต์ รูปที่ 3แสดงให้เห็นโครงสร้างทั่วไปของแมสเซส ซึ่งสามารถเขียนให้เป็นทางการในรูปของ BNF (Backus-Naur Form) ได้ดังนี้

```
HTTP-Message = Simple-Request | Simple-Response | Full-Request | Full-Response
Full-Request = Request-Line
*( General-Header | Request-Header | Entity-Header )
CRLF
[ Entity-Body ]
Full-Response = Status-Line
*( General-Header | Response-Header | Entity-Header )
CRLF
[ Entity-Body ]
Simple-Request = "GET" SP Request-URL CRLF
Simple-Response = [ Entity-Body ]
```

รูปที่ 4.3 เมสเซสของโปรโตคอลเอชทีทีพี

Request line
General header
Request header หรือ response header
Entity header
Entity body

รูปที่ 4.4 บล็อกบรรจุข้อมูลของเมสเซส

แมสเซส Simple-Request และ Simple-Response ได้ถูกกำหนดไว้ใน HTTP/0.9 สำหรับคำร้องขอนั้นเป็นเพียงคำสั่ง GET ง่ายๆ ที่ระบุถึง URL หนึ่งๆ ส่วนคำตอบสนองก็คือบล็อคที่บรรจุข้อมูลซึ่งถูกระบุใน URL นั้น คำร้องขอและคำตอบสนองที่สมบูรณ์จะใช้ประโยชน์จากฟิลด์ต่างๆ ต่อไปนี้

ตารางที่ 4.5 ความหมายของฟิลล์ต่างๆในเมจเซส

ชนิดของแมสเซส	ความหมาย
Request-Line	ระบุประเภทของแมสเซสและรีซอร์สที่ต้องการ
Response-Line	แสดงข้อมูลสถานะที่เกี่ยวข้องกับคำตอบสนองนั้น
General-Header	ประกอบด้วยฟิลด์ที่เกี่ยวข้องกับทั้งแมสเซสร้องขอและแมสเซสตอบสนอง แต่ไม่เกี่ยวข้องกับเอ็นติตี้ที่ส่งไป
Request-Header	บรรจุข้อมูลเกี่ยวกับคำร้องขอและไคลเอ็นต์
Response-Header	บรรจุข้อมูลเกี่ยวกับคำตอบสนอง
Entity-Header	บรรจุข้อมูลเกี่ยวกับรีซอร์สที่ถูกระบุจากคำร้องขอและข้อมูลเกี่ยวกับเอ็นติตี้

ส่วนเฮดเดอร์ทุกประเภทของเอชทีทีพี (HTTP) ประกอบด้วยกลุ่มของฟิลด์ ตามรูปแบบของ RFC 822 แต่ละฟิลด์จะเริ่มที่ต้นบรรทัดเสมอ โดยประกอบด้วยชื่อฟิลด์ ตามด้วยเครื่องหมายเซมิโคลอนและค่าของฟิลด์ ถึงแม้ว่ากลไกการติดต่อจะไม่ซับซ้อน แต่เอชทีทีพี (HTTP) ก็ได้กำหนดให้มีฟิลด์และพารามิเตอร์อยู่เป็นจำนวนมาก ดังแสดงไว้ใน ตารางที่ 2 ต่อไปเราจะมาพิจารณาฟิลด์ในส่วนเฮดเดอร์ทั่วไป หลังจากนั้นจึงเป็นเรื่องของเฮดเดอร์คำร้องขอ, เฮดเดอร์คำตอบสนอง และเอ็นติตี้ ตามลำดับ

ตารางที่ 4.6 เฮดเดอร์ของแมสเซสทุกชนิด

แมสเซสทุกชนิด			
ฟิลด์ในส่วนเฮดเดอร์ทั่วไป		ฟิลด์ในส่วนเฮดเดอร์ของเอ็นติตี้	
Cache-Control	Keep-Alive	Allow	Derived-From
Connection	MIME-Version	Content-Encoding	Expires
Data	Pragma	Content-Language	Last-Modified
Forwarded	Upgrade	Content-Length	Link
		Content-MDS	Title
		Content-Range	Transfer-Encoding
		Content-Type	URI-Header
		Content-Version	extension-header

ตารางที่ 4.7 เฮดเดอร์ของแมสเซสร้องขอ

แมสเซสร้องขอ			
เมธ็อดร้องขอ		ฟิลด์ในส่วนเฮดเดอร์ร้องขอ	
OPTIONS	MOVE	Accept	If-Modified-Since
GET	DELETE	Accept-Charset	Proxy-Authorization
HEAD	LINK	Accept-Encoding	Range
POST	UNLINK	Accept-Language	Referer
PUT	TRACE	Authorization	Unless
PATCH	WRAPPED	From	User-Agent
COPY	extension-method	Host	

ตารางที่ 4.8 เฮดเดอร์ของแมสเซสตอบสนอง

แมสเซสตอบสนอง			
รหัสตอบสนอง		**ฟิลด์ในส่วนเฮดเดอร์ตอบสนอง**	
Continue	Payment Required	Location	
Switching Protocols	Forbidden	Proxy-Authenticate	
OK	Not Found	Public	
Created	Method Not Allowed	Retry-After	
Accepted	None Acceptable	Server	
Non-Authoritative	Proxy Authentication	WWW-Authenticate	
Information	Required		
No Content	Request Timeout		
Reset Content	Conflict		
Partial Content	Gone		
Multiple Content	Length Required		
Moved Permanently	Unless True		
Moved Temporarily	Internal Server Error		
See Other	Not Implemented		
Not Modified	Bad Gateway		
Use Proxy	Service Unavailable		
Bad Request	Gateway Timeout		
Unauthorized	extension code		

4.8.1 ฟิลด์ที่อยู่ในส่วนเฮดเดอร์ทั่วไป

ฟิลด์ในส่วนนี้สามารถใช้ได้ทั้งในแมสเซสร้องขอและแมสเซสตอบสนอง ฟิลด์เหล่านี้ใช้ได้กับแมสเซสทั้ง 2 ประเภท โดยบรรจุข้อมูลที่ไม่ได้เกี่ยวข้องกับเอ็นติตี้โดยตรง ประกอบด้วยฟิลด์ดังต่อไปนี้

ตารางที่ 4.9 ฟิลล์ที่ดูในเฮดเดอร์ทั่วไป

เฮดเดอร์	ความหมาย
Cache-Control	ใช้ควบคุมกลไกการทำแคชสำหรับชุดคำร้องขอ/คำตอบสนอง จุดมุ่งหมายก็เพื่อป้องกันมิให้เกิดความขัดแย้งระหว่างแคชกับคำร้องขอหรือคำตอบสนอง
Connection	บรรจุรายการของคีย์เวิร์ดและชื่อฟิลด์ในส่วนเฮดเดอร์ซึ่งเกี่ยวเนื่องกับการเชื่อมต่อแบบ TCP ระหว่างผู้ส่งกับผู้รับที่ไม่ใช่ทันเนล
Date	วันและเวลาที่แมสเซสถูกสร้างขึ้น
Forwarded	ใช้โดยเกตเวย์และพร็อกซี่ เพื่อบ่งชี้ขั้นตอนกลางของคำร้องขอหรือคำตอบสนอง แต่ละเกตเวย์และพร็อกซี่ที่จัดการกับแมสเซสอาจผนวกฟิลด์ Forwarded นี้เพื่อแจ้ง URL ของตนเอง
Keep-Alive	จะปรากฏฟิลด์นี้ถ้ามีคีย์เวิร์ด keep-alive อยู่ในฟิลด์ Connection ที่ได้รับเข้ามา ฟิลด์นี้ทำหน้าที่แสดงข้อมูลต่อผู้ที่ร้องขอให้มีการเชื่อมต่อ ฟิลด์นี้อาจระบุช่วงเวลาสูงสุดที่ผู้ส่งจะเปิดการเชื่อมต่อรอไว้สำหรับคำร้องขอถัดไป หรือจำนวนคำร้องขอสูงสุดที่อนุญาตให้มีได้ในการเชื่อมต่อนั้น
MIME-Version	ระบุให้รู้ว่าแมสเซสสอดคล้องกับเวอร์ชันใดของ MIME
Pragma	บรรจุคำสั่งควบคุมที่เกี่ยวข้องกับชุดคำร้องขอ/คำตอบสนองหนึ่งๆ
Upgrade	ใช้ในคำร้องขอเพื่อระบุถึงโปรโตคอลอื่นๆ ที่ไคลเอ็นต์รองรับและต้องการใช้ หรือใช้ในคำตอบสนองเพื่อระบุโปรโตคอลที่ถูกใช้

4.8.2 แมสเซสร้องขอ (Request Messages)

แมสเซสร้องขอที่สมบูรณ์ประกอบด้วยบรรทัดสถานะ ตามด้วยเฮดเดอร์ทั่วไป, เฮดเดอร์คำร้องขอ และเฮดเดอร์ของเอ็นติตี้ หลังจากนั้นจึงเป็นส่วนของเอ็นติตี้ซึ่งจะมีหรือไม่มีก็ได้

เมธ็อดร้องขอ (Request Method)

แมสเซสร้องขอที่สมบูรณ์จะเริ่มต้นด้วยส่วน Request-Line เสมอ ซึ่งมีรูปแบบดังนี้

```
Request-Line = Method SP Request-URL SP HTTP-Version CRLF
```

รูปที่ 4.5 แมสเซสร้องขอที่สมบูรณ์

พารามิเตอร์ Method เป็นตัวระบุคำสั่งของการร้องขอที่แท้จริง หรือที่นิยมเรียกกันใน HTTP ว่าเมธ็อดนั่นเอง สำหรับ Request-URL คือ URL ของรีซอร์สที่ถูกร้องขอ และ HTTP-Version คือ หมายเลขเวอร์ชันของ HTTP ที่ผู้ส่งใช้ HTTP/1.1 กำหนดให้มีเมธ็อดร้องขอดังต่อไปนี้

ตารางที่ 4.10 เมธ็อดร้องขอของโปรโตคอลเอชทีทีพี

เมธ็อด	ความหมาย
OPTIONS	ใช้สอบถามข้อมูลเกี่ยวกับตัวเลือกที่ใช้ได้ในคำร้องขอ/คำตอบสนองซึ่งระบุด้วย URL นี้
GET	ร้องขอข้อมูลที่ระบุใน URL โดยส่งกลับมาในส่วนของเอ็นติตี้
HEAD	มีความหมายเหมือนเมธ็อด GET ยกเว้นคำตอบสนองที่เซิร์ฟเวอร์ส่งกลับคืนมานั้นต้องไม่มีส่วนของเอ็นติตี้ แต่ฟิลด์ต่างๆ ในส่วนเฮดเดอร์ของคำตอบสนองจะเหมือนกัน เมธ็อดนี้ช่วยให้ไคลเอ็นต์ทราบข้อมูลเกี่ยวกับรีซอร์สนั้นโดยไม่จำเป็นต้องมีการส่งเอ็นติตี้มา
POST	ร้องขอเอ็นติตี้ในฐานะที่เป็น subordinate ของ URL ที่ระบุ เอ็นติตี้ที่ส่งมานั้นเป็น subordinate ของ URL ในลักษณะเดียวกันกับที่ไฟล์เป็น subordinate ของไดเร็กทอรีที่บรรจุไฟล์นั้นไว้ ทำนองเดียวกับบทความข่าวที่เป็น subordinate ของกลุ่มข่าว (Newsgroup) และทำนองเดียวกับเรคอร์ดที่เป็น subordinate ของฐานข้อมูล
PUT	ร้องขอเอ็นติตี้ แล้วนำไปเก็บไว้ภายใต้ URL ที่กำหนด ซึ่งอาจเป็นรีซอร์สใหม่ของ URL ใหม่ หรือแทนที่รีซอร์สเดิมด้วย URL ที่มีอยู่แล้ว
PUT	ร้องขอเอ็นติตี้ แล้วนำไปเก็บไว้ภายใต้ URL ที่กำหนด ซึ่งอาจเป็นรีซอร์สใหม่ของ URL ใหม่ หรือแทนที่รีซอร์สเดิมด้วย URL ที่มีอยู่แล้ว
PATCH	คล้ายคลึงกับเมธ็อด PUT แต่เอ็นติตี้จะเก็บรายการความแตกต่างจากเนื้อหาของรีซอร์สเดิมที่ระบุใน URL
COPY	ร้องขอให้มีการคัดลอกรีซอร์สที่ระบุด้วย URL ใน Request-Line ไปยังตำแหน่งที่ระบุด้วยฟิลด์ URL-Header ในส่วน Entity-Header ของแมสเซส

ตารางที่ 4.10 (ต่อ) เมธ็อดร้องขอของโปรโตคอลเอชทีทีพี

เมธ็อด	ความหมาย
MOVE	ร้องขอให้มีการย้ายรีซอร์สที่ระบุด้วย URL ใน Request-Line ไปยังตำแหน่งที่ระบุด้วยฟิลด์ URL-Header ในส่วน Entity-Header ของแมสเซส เมธ็อดนี้เทียบเท่ากับการใช้เมธ็อด COPY แล้วตามด้วยเมธ็อด DELETE
DELETE	ร้องขอให้ออริจิ้นเซิร์ฟเวอร์ลบรีซอร์สที่ระบุด้วย URL ใน Request-Line
LINK	สร้างลิงค์ 1 ลิงค์หรือมากกว่าจากรีซอร์สที่ระบุใน Request-Line โดยลิงค์ต่างๆ จะถูกกำหนดไว้ในฟิลด์ Link ใน Entity-Header
UNLINK	ลบลิงค์ 1 ลิงค์หรือมากกว่าจากรีซอร์สที่ระบุใน Request-Line โดยลิงค์ต่างๆ จะถูกกำหนดไว้ในฟิลด์ Link ใน Entity-Header
TRACE	ร้องขอให้เซิร์ฟเวอร์ส่งสิ่งต่างๆ ที่ได้รับมา กลับไปในรูปของเอ็นติตี้ของคำตอบสนอง เมธ็อดนี้มีจุดมุ่งหมายเพื่อการทดสอบและวิเคราะห์เป็นสำคัญ
WRAPPED	อนุญาตให้ไคลเอ็นต์ส่งคำร้องในแบบ encapsulated 1 คำร้องหรือมากกว่าคำร้องอาจได้รับการเข้ารหัสไว้หรือได้รับการประมวลผลอย่างใดอย่างหนึ่ง ซึ่งเซิร์ฟเวอร์จะต้องแกะคำร้องนั้นออกมาแล้วประมวลผลไปตามความเหมาะสม
Extension-method	อนุญาตให้มีการกำหนดเมธ็อดขึ้นมาใหม่โดยไม่ต้องปรับเปลี่ยนโปรโตคอล อย่างไรก็ตาม เราไม่สามารถทึกทักได้ว่าผู้รับรู้จักกับเมธ็อดเหล่านั้น

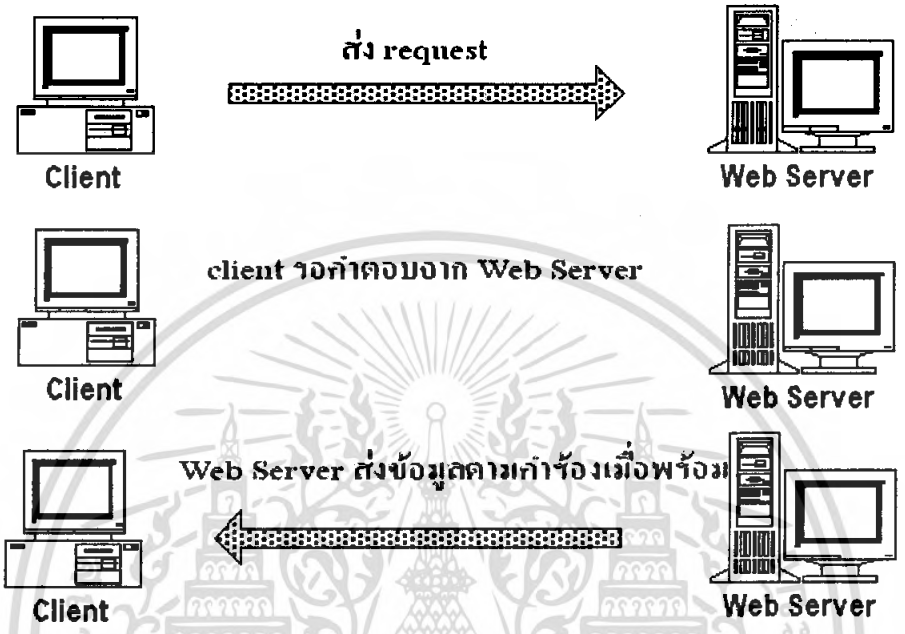
4.8.3 ฟิลด์ในส่วนเฮดเดอร์ของคำร้องขอ

ฟิลด์ในส่วนเฮดเดอร์ของคำร้องขอทำหน้าที่เป็นส่วนเพิ่มเติมของคำร้องขอ โดยจัดเตรียมข้อมูลและพารามิเตอร์ที่เกี่ยวข้องกับคำร้องขอนั้น ฟิลด์ต่างๆ ต่อไปนี้ถูกกำหนดไว้ใน HTTP/1.1

ตารางที่ 4.11 ฟิลด์ในส่วนเฮดเดอร์ของคำร้องขอ

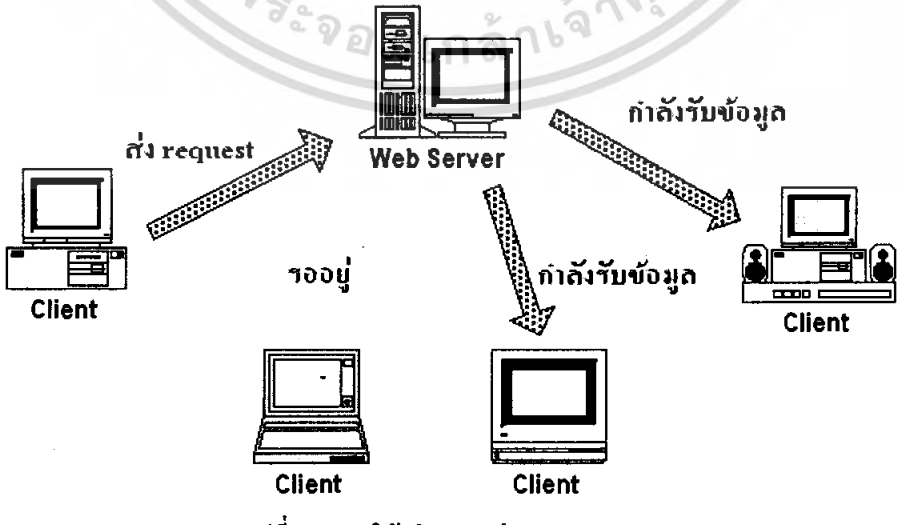
ฟิลด์	เฮดเดอร์
Accept	ประเภทของสื่อและช่วงค่าที่ยอมรับได้ในคำตอบสนองของคำร้องนั้น
Accept-Charset	ชุดอักขระที่ยอมรับได้ในคำตอบสนอง
Accept-Encoding	รูปแบบการเข้ารหัสเนื้อหาในส่วนของเอ็นติตี้ที่ยอมรับได้ การเข้ารหัสนั้นมีจุดมุ่งหมายหลักเพื่อการบีบอัดเอกสารและการเข้ารหัสเพื่อความลับของข้อมูล โดยทั่วไปแล้วรีซอร์สจะถูกเก็บอยู่ในส่วนของการเข้ารหัสนี้ และจะถูกถอดรหัสก่อนนำไปใช้จริง
Accept-Language	จำกัดกลุ่มของภาษาที่ต้องการให้ใช้ในคำตอบสนอง
Authorization	บรรจุค่าที่เรียกว่าเป็น "หนังสือรับรอง" ซึ่งไคลเอ็นต์ใช้ยืนยันตนเองต่อเซิร์ฟเวอร์
From	อีเมลล์แอดเดรสของผู้ควบคุมยูสเซอร์เอเจนต์ที่ใช้ร้องขอ
Host	ระบุโฮสต์ของรีซอร์สที่ร้องขอไป
If-Modified-Since	ใช้กับเมธ็อด GET ส่วนนี้จะรวมถึงพารามิเตอร์ที่ระบุวัน/เวลาด้วย โดยรีซอร์สจะถูกส่งมาให้ก็ต่อเมื่อรีซอร์สนั้นได้รับการปรับปรุงภายหลังจากวัน/เวลาที่ระบุ ฟิลด์นี้ช่วยให้การอัพเดทแคชเป็นไปอย่างมีประสิทธิภาพ โดยกลไกการทำแคชสามารถส่งแมสเซสที่ใช้เมธ็อด GET ไปยังออริจิ้นเซิร์ฟเวอร์เป็นระยะๆ และจะได้รับแมสเซสตอบสนองขนาดเล็กๆ กลับคืนมาจนกว่าจะมีการปรับปรุงรีซอร์ส
Proxy-Authorization	อนุญาตให้ไคลเอ็นต์ระบุตนเองต่อเซิร์ฟเวอร์ได้เมื่อจำเป็น
Range	เตรียมการไว้ใช้กับเมธ็อด GET ในอนาคต โดยไคลเอ็นต์สามารถร้องขอส่วนใดส่วนหนึ่งของรีซอร์สที่ระบุได้
Referer	คือ URL ของรีซอร์สซึ่ง Request-URL ได้รับมา
Unless	มีหน้าที่คล้ายกับฟิลด์ If-Modified-Since แต่ด้วยความแตกต่าง 2 ประการ คือ (1) ไม่ถูกจำกัดอยู่กับเมธ็อด GET และ (2) การเปรียบเทียบจะกระทำกับค่าของฟิลด์ใดๆ ใน Entity-Header ก็ได้ แทนที่จะเป็นค่าของวัน/เวลา
User-Agent	บรรจุข้อมูลเกี่ยวกับยูสเซอร์เอเจนต์ที่สร้างคำร้องขอนี้ขึ้นมา ฟิลด์นี้ใช้เพื่อบันทึกเก็บเป็นสถิติไว้

โปรโตคอลเอชทีทีพี (HTTP) นี้วิ่งอยู่บนโปรโตคอลทีซีพีไอพี (TCP/IP) อีกชั้นหนึ่ง รูปแบบการทำงานจะไม่มีการจองสาย โดย client จะเรียกข้อมูลจาก server โดยการส่ง request ไป แล้วจะตัดการติดต่อทันที จากนั้นจะรอจนกระทั่ง server ส่งข้อมูลมาให้



รูปที่ 4.6 การร้องขอข้อมูลจากเซิร์ฟเวอร์

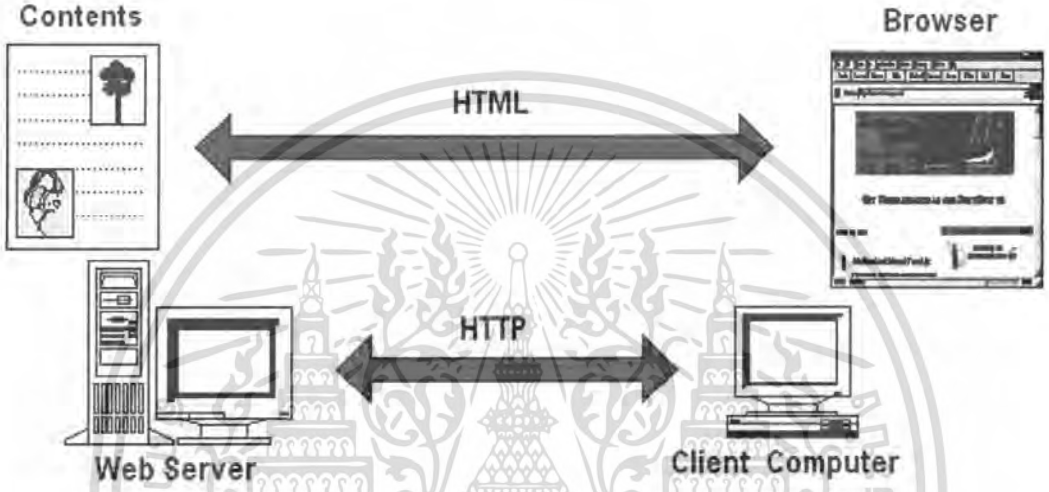
ประโยชน์ของการทำงานแบบไม่จองสายของโปรโตคอลเอชทีทีพี (HTTP) ทำให้ WWW server สามารถให้บริการ client ได้หลายๆ คนพร้อมๆ กัน การสื่อสารของ WWW จึงมีประสิทธิภาพมากขึ้น



รูปที่ 4.7 การให้บริการลูกข่ายจำนวนมาก

4.9 ความสัมพันธ์ ระหว่าง HTTP กับ HTML

HTTP คือโปรโตคอลที่ใช้สื่อสารระหว่าง client computer กับ server computer ทำให้ทั้งสองเครื่องรู้ว่าจะจัดการส่งข้อมูลไปอย่างไร ส่วน HTML คือสื่อภาษาที่ทำให้เอกสารหรือ contents ที่อยู่บนเครื่อง server computer เมื่อถูกส่งมาที่ client computer แล้วจะนำไปแสดงได้อย่างไร เราเรียกซอฟท์แวร์ที่ใช้แสดงนี้ว่า Browser



รูปที่ 4.8 ความสัมพันธ์ ระหว่าง HTTP กับ HTML

4.9.1 ข้อดีของการแยกชั้นการทำงานระหว่าง HTTP กับ HTML

1. Contents
   - พัฒนาบนเครื่องแบบใดก็ได้ เช่น PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray etc.
   - มีเครื่องมือช่วยในการพัฒนามากมาย
2. Web Server
   - เครื่องที่ใช้เป็น Web Server เป็นเครื่องใดๆ ก็ได้ เช่น PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray
   - ในแต่ละ Platform มี โปรแกรม Web Server ให้เลือกมากมาย
3. Client Computer
   - เครื่องที่ใช้เป็น Client Computer เป็นเครื่องใดๆ ก็ได้ เช่น PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray, TV with Set-Top Box, Pen Computer etc.
4. Browser
   - โปรแกรม Browser มีให้เลือกใช้มากมายบน PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray, TV with Set-Top Box, Pen Computer etc.

บทที่ 5
การทำงานของระบบตรวจสอบและบันทึกพฤติกรรมผู้ใช้งาน

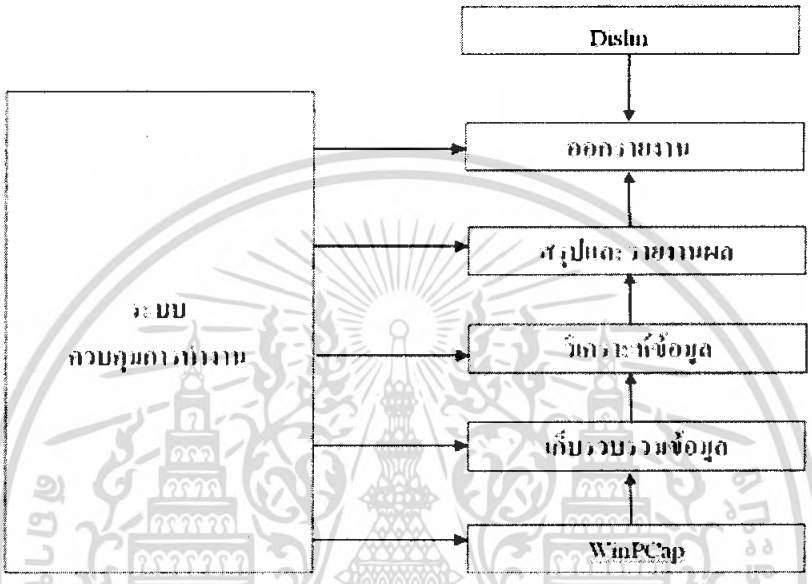
5.1 ที่มาของโครงงาน

เนื่องจากการได้ไปค้นคว้าและทดลองใช้โปรแกรมจำพวก Sniffer เป็นจำนวนมาก เช่น Ethereal , Sniphere , Traffic watch , Packetyzer เป็นต้น เหล่านี้จะเน้นไปที่การทำการดักจับข้อมูลที่เป็นแพ็กเก็ตขึ้นมา โดยที่ไม่มีการประมวลผลสถิติการใช้งานของผู้ใช้มีการออกรายงานเป็นกราฟบ้างแต่ออกจากแพ็กเก็ตโดยตรง ไม่ได้ผ่านการบันทึกลงไฟล์ที่ได้มีการคำนวณและบันทึกไว้ นอกจากนี้ก็ไม่มีการเก็บล็อกที่ละเอียด โดยทั่วไปจะเป็นการเก็บล็อกเป็นแพ็กเก็ตธรรมดา โดยไม่มีการตีความหรือวิเคราะห์ข้อมูลแต่อย่างใด ดังนั้นจึงเกิดความคิดที่จะทำโปรแกรมดักจับข้อมูลที่สามารถล่วงรู้พฤติกรรมของผู้ใช้งาน สามารถบันทึกข้อมูลการใช้งานได้เพื่อให้สามารถตรวจสอบย้อนหลังได้ว่า มีใคร ทำอะไร เมื่อใด กับใคร

จุดประสงค์ของการใช้งานโปรแกรมนั้นจะเน้นไปในการตรวจสอบเป็นส่วนใหญ่ในส่วนของการใช้งานส่วนหลักๆ เช่น การใช้เว็บไซต์ (HTTP) และการใช้โปรแกรมสนทนา (MSN) จะมีการเก็บล็อกที่ละเอียดเป็นพิเศษกว่าการใช้งานโดยทั่วไป สามารถแสดงการใช้งานต่างๆแบ่งตามไอพีอย่างชัดเจน รวมทั้งยังแสดงการใช้งานเป็นเปอร์เซนต์เทียบกับปริมาณการใช้งานเครือข่ายทั้งหมดด้วย นอกจากนี้มีการแสดงสถิติการใช้งานรวมของเครือข่ายทั้งหมด และสรุปเป็นกราฟโดยแบ่งตามไอพีแอดเดรสด้วย

5.2 การทำงานของระบบ

ระบบตรวจสอบและบันทึกพฤติกรรมผู้ใช้งาน พัฒนาขึ้นจากไลบรารี WinPCap ร่วมกับ Microsoft Visual Studio 2005 โดยไลบรารี่ดังกล่าวจะทำงานอยู่ในชั้นล่างสุดของโปรแกรม การทำงานของไลบรารีจะทำหน้าที่ควบคุมการทำงานของการ์ดอีเทอร์เน็ต

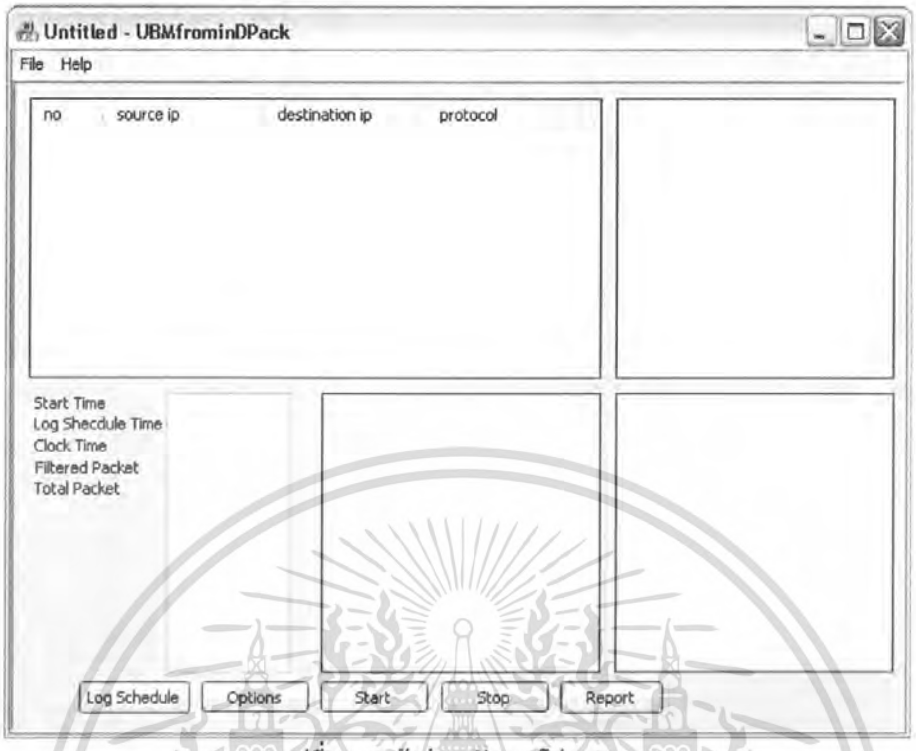


รูปที่ 5.1 โครงสร้างของระบบ

ไลบรารี Winpcap จะทำหน้าที่ในการดักจับแพ็กเก็ตขึ้นมาจากอีเทอร์เน็ตการ์ด โดยเราจะทำการถอดเฮดเดอร์ต่างๆ ขึ้นมาเพื่อใช้งานในส่วนของการเก็บรวมข้อมูล วิเคราะห์แยกประเภทของข้อมูลแต่ละประเภท จากนั้นจึงนำมาสรุปเป็นสถิติ แสดงผลและเก็บข้อมูลไว้ในล็อกไฟล์ ส่วนต่างๆทั้งหมดจะถูกควบคุมโดยโปรแกรมซึ่งพัฒนาบน Microsoft Visual Studio 2005 เป็นภาษา C++ ผ่านUser Interface โดยการทำงานทั้งหมดมีดังนี้

5.2.1 ระบบควบคุมการทำงาน

ในที่นี้หมายถึงโปรแกรมที่ถูกพัฒนาขึ้นด้วยภาษา C++ ซึ่งผู้ใช้งานจะควบคุมการทำงานทั้งหมด ผ่านทาง User Interface



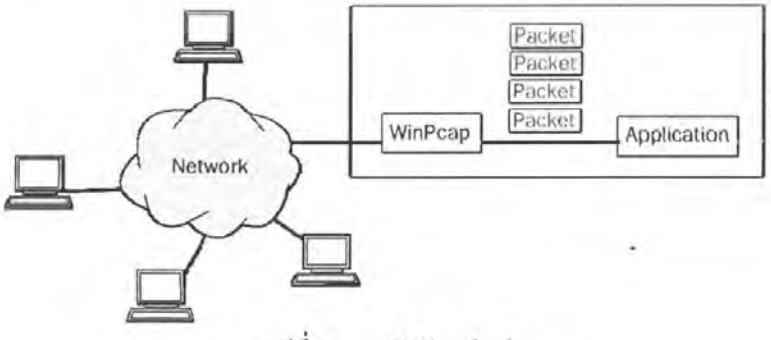
รูปที่ 5.2 หน้าต่างหลักของโปรแกรม

5.2.2 WinPCap

WinPCap เป็นไลบรารี่ที่ช่วยในการดักจับแพ็กเก็ตทั้งหมดขึ้นมาจากการ์ดอีเทอร์เน็ต โดยเราจะใช้ฟังก์ชัน

```
int pcap_next_ex
( pcap_t *  p, struct pcap_pkthdr **  pkt_header,
                const u_char **  pkt_data )
```

เพื่อทำการดักจับแพ็กเก็ตขึ้นมาจากอินเตอร์เฟสที่ได้เลือกไว้ จากนั้นส่วนเฮดเดอร์ของแพ็กเก็ตจะถูกเก็บไว้ในตัวแปร pkt_header และส่วนของข้อมูลจะเก็บไว้ที่ pkt_data เพื่อนำไปประมวลผลต่อไป



รูปที่ 5.3 การดักจับแพ็กเก็ต

5.2.3 เก็บรวบรวมข้อมูล

หลังจากโปรแกรมจะทำการเก็บรวบรวมแพ็กเก็ตที่ถูกดักจับขึ้นมาทั้งหมด เราก็จะได้ตัวแปรที่เก็บส่วนของเฮดเดอร์และข้อมูลของแต่ละแพ็กเก็ตไว้ เราจะทำการแตกส่วนของแพ็กเก็ตดาต้าออกมา เราสามารถแตกส่วนต่างๆของแพ็กเก็ตดาต้าได้ตามลำดับของ OSI Layer ดังนี้

ส่วนที่ 1 ส่วนของ Ethernet Header

ส่วนที่ 2 ส่วนของ IP Header

ส่วนที่ 3 ส่วนของ TCP Header

ส่วนที่ 4 ส่วนของ payload

รูปที่ 5.4 การเก็บรวบรวมข้อมูล

no	source ip	destination ip	protocol
000206	202.44.5.18	161.246.5.99	TCP
000205	202.44.5.18	161.246.5.99	TCP
000204	161.246.5.99	202.44.5.18	TCP
000203	202.44.5.18	161.246.5.99	TCP
000202	161.246.5.99	202.44.5.18	TCP
000201	202.44.5.18	161.246.5.99	TCP
000200	161.246.5.99	202.44.5.18	TCP
000199	202.44.5.18	161.246.5.99	TCP
000198	202.44.204.85	161.246.5.99	TCP
000197	161.246.5.99	202.44.5.18	TCP
000196	202.44.5.18	161.246.5.99	TCP
000195	161.246.5.99	202.44.5.18	TCP

รูปที่ 5.5 แพ็กเก็ตที่ดักจับได้

5.2.4 วิเคราะห์ข้อมูล

โปรแกรมจะทำการจำแนกแพ็กเก็ตที่ดักจับได้ออกเป็นหมวดหมู่ โดยเราจะใช้ในส่วนของพอร์ตต้นทางและปลายทางในเฮดเดอร์ของทีซีพีเลเซอร์ โดยจำแนกแพ็กเก็ตออกเป็น 3 กลุ่มใหญ่ๆ คือ แพ็กเก็ตที่ใช้โปรโตคอล HTTP (พอร์ต 80) , แพ็กเก็ตที่ใช้โปรโตคอล MSN (พอร์ต 1863) และแพ็กเก็ตที่ใช้โปรโตคอลอื่นๆ

หลังจากที่เราได้ทำการแตกเฮดเดอร์ออกมาจนถึงชั้นทีซีพีแล้ว แล้วจะนำส่วนของพอร์ตต้นทางและพอร์ตปลายทางมาเข้าฟังก์ชันในการแยกประเภทของแพ็กเก็ตจากนั้นจึงนำแพ็กเก็ตแต่ละประเภทไปเก็บรวบรวมข้อมูลของโปรโตคอลนั้นๆ

5.2.4.1 โครงสร้างการวิเคราะห์ข้อมูลและสรุปผล

สำหรับในส่วนการรวบรวมสถิตินี้เราได้ทำการสร้างโครงสร้าง (Struct) ขึ้นมาเพื่อรองรับการเก็บค่าสถิติการใช้งานต่างๆของแต่ละไอพี ประกอบด้วยโครงสร้างที่สำคัญๆ ได้แก่

5.2.4.1.1 Ethernet header

จะประกอบด้วยข้อมูลในส่วน เฮดเดอร์ที่สำคัญของเลเยอร์สองหรือ ชั้นอีเทอร์เน็ต

```
struct sniff_ethernet {
    u_char  ether_dhost[ETHER_ADDR_LEN];    /* destination host address */
    u_char  ether_shost[ETHER_ADDR_LEN];    /* source host address */
    u_short ether_type;                     /* IP? ARP? RARP? etc */
};
```

รูปที่ 5.6 โครงสร้างที่ใช้เก็บเฮดเดอร์ของชั้นอีเทอร์เน็ต

5.2.4.1.2 IP header

จะประกอบด้วยข้อมูลในส่วน เฮดเดอร์ที่สำคัญของเลเยอร์สามหรือ ชั้นเน็ตเวิคซึ่งข้อมูลที่สำคัญๆได้แก่ หมายเลขไอพีต้นทางหรือปลายทาง ชนิดของโปรโตคอล ความยาวรวมของเฮดเดอร์

```
struct sniff_ip {
        u_char  ip_vhl;                 /* version << 4 | header length >> 2 */
        u_char  ip_tos;                 /* type of service */
        u_short ip_len;                 /* total length */
        u_short ip_id;                  /* identification */
        u_short ip_off;                 /* fragment offset field */
        #define IP_RF 0x8000            /* reserved fragment flag */
        #define IP_DF 0x4000            /* dont fragment flag */
        #define IP_MF 0x2000            /* more fragments flag */
        #define IP_OFFMASK 0x1fff       /* mask for fragmenting bits */
        u_char  ip_ttl;                 /* time to live */
        u_char  ip_p;                   /* protocol */
        u_short ip_sum;                 /* checksum */
        struct  in_addr ip_src,ip_dst;  /* source and dest address */
};
```

รูปที่ 5.7 โครงสร้างที่ใช้เก็บเฮดเดอร์ของโปรโตคอลไอพี

5.2.4.1.3 TCP header

จะประกอบด้วยข้อมูลในส่วน เฮดเดอร์ที่สำคัญของเลเยอร์สี่หรือ ชั้นทรานสปอร์ตซึ่งข้อมูลที่สำคัญๆ ได้แก่ พอร์ตต้นทางหรือปลายทาง

```
struct sniff_tcp {
        u_short th_sport;                       /* source port */
        u_short th_dport;                       /* destination port */
        tcp_seq th_seq;                         /* sequence number */
        tcp_seq th_ack;                         /* acknowledgement number */
        u_char  th_offx2;                       /* data offset, rsvd */
#define TH_OFF(th)      (((th)->th_offx2 & 0xf0) >> 4)
        u_char  th_flags;
        #define TH_FIN  0x01
        #define TH_SYN  0x02
        #define TH_RST  0x04
        #define TH_PUSH 0x08
        #define TH_ACK  0x10
        #define TH_URG  0x20
        #define TH_ECE  0x40
        #define TH_CWR  0x80
        #define TH_FLAGS
(TH_FIN|TH_SYN|TH_RST|TH_ACK|TH_URG|TH_ECE|TH_CWR)
        u_short th_win;                         /* window */
        u_short th_sum;                         /* checksum */
        u_short th_urp;                         /* urgent pointer */
};
```

รูปที่ 5.8 โครงสร้างที่ใช้เก็บเฮดเดอร์ของโปรโตคอลทีซีพี

5.2.4.1.4 Statistic collector

จะเป็นส่วนที่ช่วยในการเก็บข้อมูลในส่วน เฮดเดอร์ที่สำคัญแต่ละไอพี

```
struct sum_port
{
        u_short packethttp;                 /* HTTP packet count */
        u_short packetmsn;                  /*MSN packet count */
        u_short packetother;                /* Other packet count */
        u_short total;                      /* Total packet count */
        int payload;                        /* Total payload count */
        vector<string> chatlist;            /* MSN chat list */
        vector<string> URL_list;            /* Web list */

};
```

รูปที่ 5.9 โครงสร้างที่ใช้เก็บข้อมูลโดยสรุป

5.2.4.1.5 MSN map

จะเป็นส่วนช่วยในการจัดการการทำงานของโปรแกรมสนทนาเอ็มเอสเอ็น

```
struct MSN
{
        CString sip;                        /* source IP */
        string serverip[10];                /*Server IP */
        string semail;                      /* source Email*/
        string demail[10];                  /* Destination Email */
        int NoOfServIp;                     /*Server IP count */

};
```

รูปที่ 5.10 โครงสร้างที่ใช้เก็บข้อมูลของโปรโตคอลเอ็มเอสเอ็น(MSN)

หลังจากที่เราได้ค่าของพอร์ตต้นทางและปลายทางแล้ว เราจะทำการแยกเข้าฟังก์ชันเพื่อใช้ในการเก็บข้อมูลนั้นๆแยกเป็น

1. การเก็บรวมรวมข้อมูลของโปรแกรมสนทนา
2. การเก็บรวบรวมข้อมูลของโปรโตคอลเอชทีทีพี (HTTP)

นอกเหนือจากโปรโตคอลสองตัวนี้ เราจะทำการเก็บค่าการใช้งานเป็นจำนวนแพ็กเก็ตที่อยู่ในสถานการใช้งานแบบโปรโตคอลอื่น

5.2.4.2 การเก็บรวมรวมข้อมูลของโปรแกรมสนทนา

เนื่องจากการทำงานคร่าวๆของโปรแกรมเอ็มเอสเอ็นอธิบายได้ดังนี้

1). ผู้ใช้งานทำการยืนยันตัวตนเข้าใช้งาน

2). เมื่อมีการส่งข้อความใดๆในการสนทนา จะต้องมีการติดต่อกับเซิร์ฟเวอร์ของระบบก่อน โดยเซิร์ฟเวอร์จะทำการติดต่อกันให้โดยเป็นตัวกลาง ดังนั้นในการสนทนาจะมีฝ่ายหนึ่งเป็นไอพีของเซิร์ฟเวอร์

3). หากไม่มีการใช้งานนานเกินช่วงเวลาที่เซิร์ฟเวอร์กำหนด หรือมีการใช้งานระบบมาก จะมีการส่งแมสเซสเพื่อยกเลิกการติดต่อ หรือบางครั้งอาจจะไม่มีการส่งข้อความใดๆเลย เพียงแต่เมื่อเริ่มการสนทนาใหม่ จะต้องทำการติดต่อกับเซิร์ฟเวอร์อีกครั้งเพื่อขอเซิร์ฟเวอร์ไอพี (Server IP) ที่ใช้ในการติดต่อใหม่

5.2.4.2.1 ตัวอย่าง แพ็กเก็ตที่ได้รับจากการสนทนา

1). การยืนยันเพื่อเข้าใช้งาน

USR 119 OK immi_87@hotmail.com ||%20iMmI%20||

เป็นการที่เซิร์ฟเวอร์ตอบรับการเข้าใช้งานของผู้ใช้งานที่มีอีเมลล์คือ immi_87@hotmail.com และมีชื่อที่ใช้แสดงในการสนทนาว่า || iMmI ||

2). การเข้าร่วมสนทนาจากคู่สนทนา (ผู้ใช้งานภายในเครือข่ายเป็นผู้ริเริ่มการสนทนา)

JOI oak_apache@hotmail.com ||%20OAK_APACHE%20||

เป็นการตอบรับการสนทนาจากคู่สนทนาว่าได้เข้าร่วมการสนทนาหรือมีการสร้างเซสชั่น (Session) กับเราแล้วจากผู้ใช้งานอีเมลล์ oak_apache@hotmail.com และมีชื่อที่ใช้แสดงในการสนทนาว่า || OAK_APACHE ||

3). การส่งข้อความที่สนทนา

MSG 129 N 140

แมสเซสแบบนี้จะถูกจับได้จากฝั่งผู้ส่งข้อความการสนทนา หมายเลข จะเป็นตัวบอกถึงความยาวของข้อความ

MSG atom_harrypotter@hotmail.com

แมสเซสแบบนี้จะเกิดขึ้นกับฝั่งที่เป็นผู้รับ หมายถึงว่าได้รับข้อความจากผู้ส่งที่มีอีเมลล์ชื่อ atom_harrypotter@hotmail.com

4). การเข้าร่วมสนทนาจากคู่สนทนา (ผู้ใช้งานภายนอกเครือข่ายเป็นผู้ริเริ่มการสนทนา)

ANS 89 aoh@hotmail.com

แมสเซสแบบนี้แสดงให้เห็นว่าเราได้ตอบรับการเข้าร่วมสนทนาหรือมีการสร้างเซสชั่น (Session) กับปลายทางที่มีอีเมล์ว่า aoh@hotmail.com

5). การยกเลิกหรือหมดช่วงเวลาที่ได้ติดต่อกับ server เอาไว้

BYE flatron_k@hotmail.com

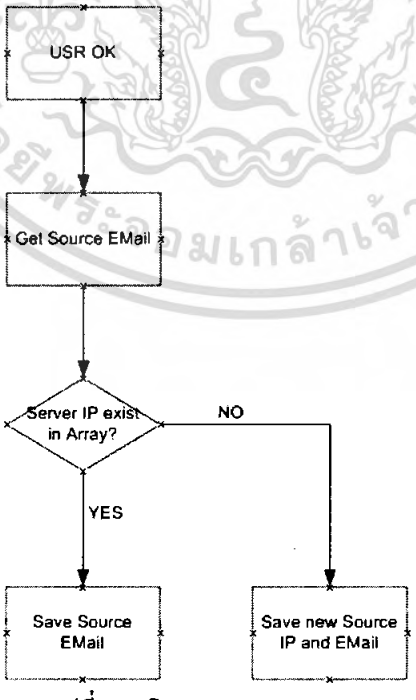
แสดงให้เห็นว่า ผู้ใช้งานอีเมลล์ flatron_k@hotmail.com ได้ยุติหรือหมดเซสชั่นการสนทนากับเราแล้ว

ดังนั้นเราจะทำการตรวจสอบหา แพ็กเก็ตที่มีแมสเซสเหล่านี้ในการรวบรวมข้อมูลของคู่สนทนาเพื่อทำการแยกแยะการสนทนาว่าเกิดขึ้นระหว่างใครกับใคร

5.2.4.2.2 กระบวนการการทำงานของการเก็บรวบรวมข้อมูลคู่สนทนา

แบ่งได้ตามประเภทของแมสเซสที่ได้รับโดยเราจะเก็บข้อมูลเอาไว้ใน Struct MSN ดังที่ได้กล่าวไปแล้ว

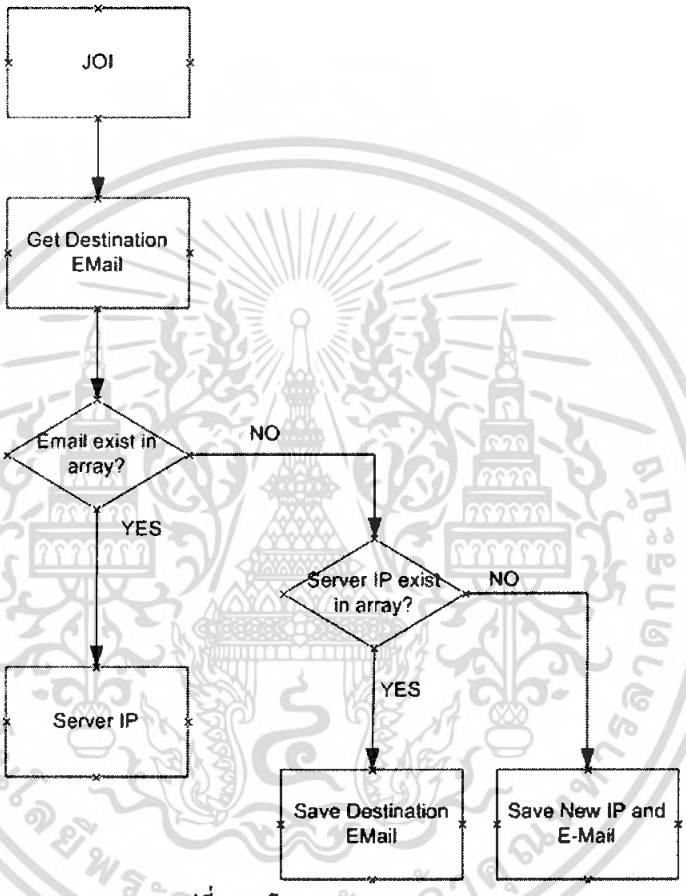
1. USR OK Message



รูปที่ 5.11 โพรเซสของ User OK Message

ในแพ็กเก็ตที่มีแมสเซสยืนยันตัวตนเข้าทำงาน ในส่วนนี้เราจะได้ตัวอีเมล์ต้นทางของผู้เข้าใช้งานหรือผู้ใช้งานที่อยู่ในเครือข่ายที่เราดักจับ เราจะต้องทำการตรวจสอบว่ามีไอพีของเซิร์ฟเวอร์นั้นบันทึกอยู่หรือยัง ถ้ามีแล้วก็ทำงานบันทึกอีเมล์ลงไปให้ตรงกับตำแหน่งของเซิร์ฟเวอร์ไอพีนั้นๆ แต่ถ้ายังไม่มีก็บันทึกข้อมูลใหม่ทั้งหมด

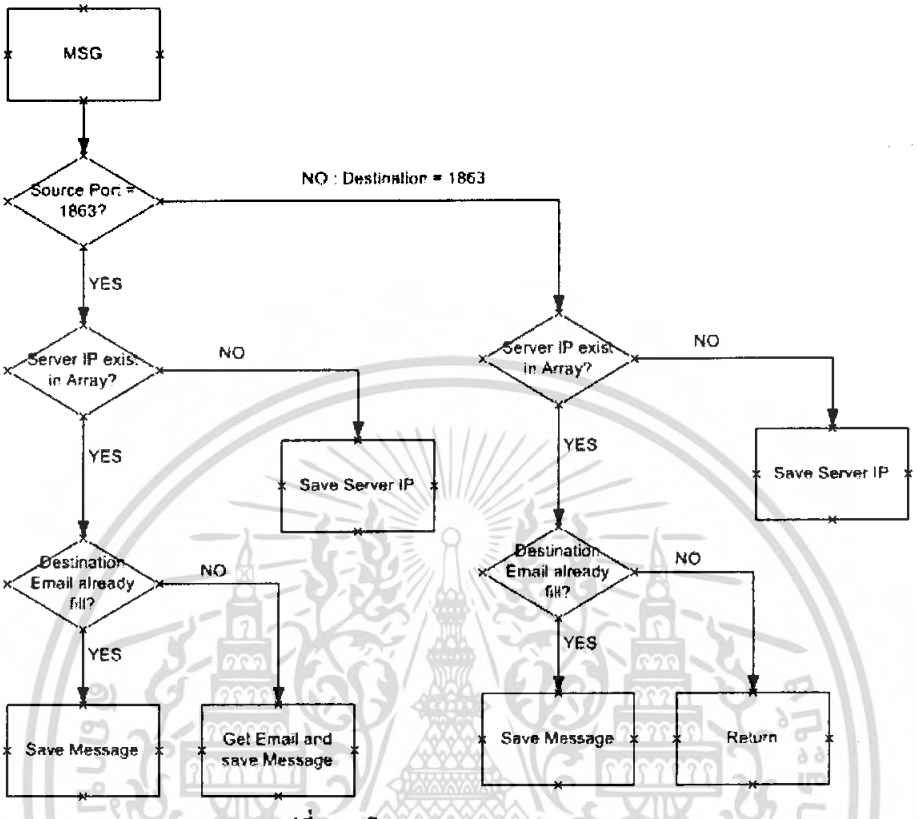
2. JOI Message



รูปที่ 5.12 โพรเซสของ JOI Message

ในแพ็กเก็ตที่มีแมสเซสตอบรับการเข้าสนทนาเมื่อมีการส่งแมสเซสไปจากผู้ใช้งานในเครือข่ายเรา เราจะได้ข้อมูลของอีเมล์ปลายทางที่เป็นคู่สนทนากับเรา เราก็ทำเช่นเดิมคือทำการตรวจสอบใน Struct MSN ว่ามีอีเมล์นั้นบันทึกอยู่หรือไม่ ถ้ามีแล้วเราก็ทำการบันทึกเซิร์ฟเวอร์ไอพีลงไปอีกครั้งเพราะในบางกรณีอาจมีการเปลี่ยนเซิร์ฟเวอร์ไอพีอย่างกระทันหันจากเซิร์ฟเวอร์ แต่ถ้ายังไม่มีอีเมล์อยู่เราก็ตรวจสอบว่ามีเซิร์ฟเวอร์ไอพีอยู่หรือไม่เนื่องจากในบางแพ็กเก็ตเราเก็บข้อมูลได้แต่เซิร์ฟเวอร์ไอพี ถ้ามีเซิร์ฟเวอร์ไอพีอยู่แล้วก็ทำการบันทึกอีเมล์ปลายทาง แต่ถ้าไม่มีก็บันทึกใหม่ทั้งหมด

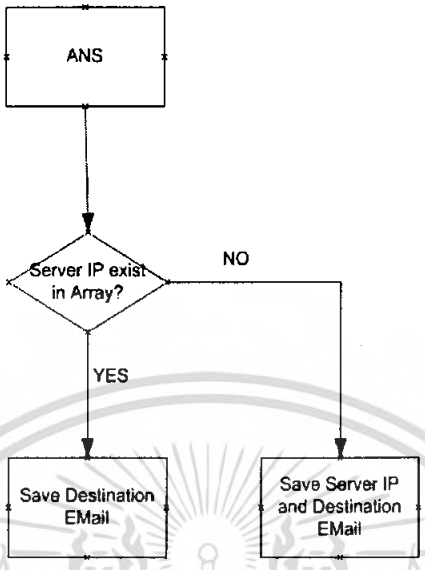
3. MSG Message



รูปที่ 5.13 โพรเซสของ MSG Message

เนื่องจากดังที่ได้อธิบายไว้แล้วว่า MSG Message มีสองแบบคือ แบบที่ส่งมาจากเซิร์ฟเวอร์ และแบบที่ส่งมาจากผู้สนทนาไปยังเซิร์ฟเวอร์ ถ้ามีพอร์ตต้นทางเป็น 1863 แสดงว่าเป็นแบบแรก แต่ถ้ามีพอร์ตปลายทางเป็น 1863 จะเป็นแบบที่สอง จากนั้นก็เข้ากระบวนการทำงานคล้ายกับรูปแบบการทำงานข้ออื่นๆต่อไป

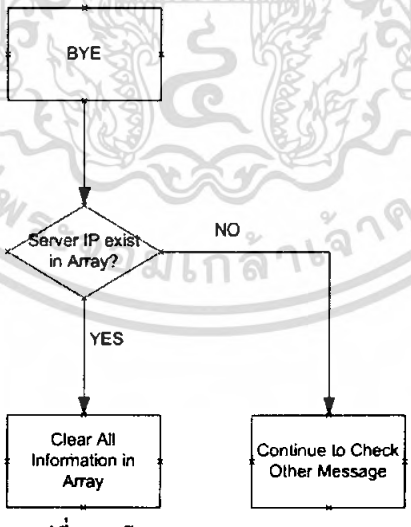
4. ANS Message



รูปที่ 5.14 โพรเซสของ ANS Message

มีวิธีการทำงานเช่นเดียวกับ USR OK Message

5. BYE Message



รูปที่ 5.15 โพรเซสของ BYE Message

แมสเซสแบบนี้เมื่อเราตรวจสอบพบเซิร์ฟเวอร์ไอพีนั้นๆอยู่ใน Struct ของเราแล้ว เราจะทำการลบข้อมูลทั้งหมด ณ ตำแหน่งเดียวกันทิ้งไป เพราะเซสชั่นการสนทนานั้นได้จบลงแล้ว

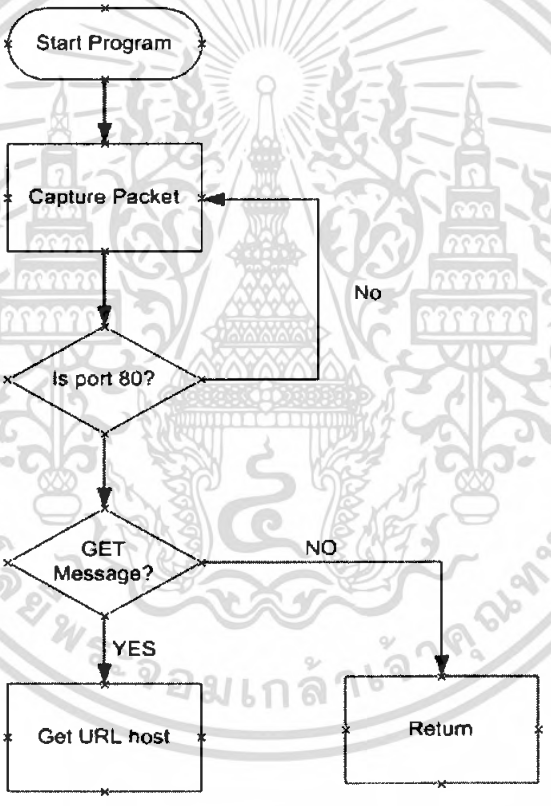
5.2.4.3 การเก็บรวบรวมข้อมูลของโปรโตคอลเอชทีทีพี (HTTP)

ในการเรียกใช้งานหรือเข้าสู่เว็บไซต์ใดๆนั้น ในการเรียกหน้าเว็บขึ้นมาจะใช้เอชทีทีพีแมสเซส (HTTP message) GET ตัวอย่างเช่น

GET / HTTP/1.1

Host: www.dek-d.com

ในที่นี้หมายถึงมีการเรียกหน้าเว็บไซต์ชื่อ www.dek-d.com ขึ้นมาหรือในบางกรณีก็ใช้ แมสเซส GET ในการเรียกส่วนประกอบต่างๆของหน้าเว็บขึ้นมา ดังนั้นการทำงานเราจะทำการตรวจสอบหาเอชทีทีพีแพ็กเก็ตที่มีส่วนประกอบของดาต้าที่มีคำว่า GET / HTTP เป็นหลัก แล้วเก็บ host ซึ่งเป็นชื่อเว็บไซต์มาเก็บไว้

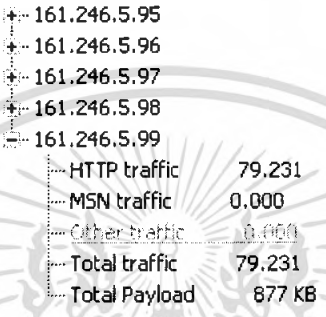


รูปที่ 5.16 โพรเซสของ Start Program

กระบวนการทำงานจะเริ่มจากการตรวจสอบว่ามีพอร์ตต้นทางหรือปลายทางเป็น 80 หรือไม่ ถ้าเป็นเราก็จะทำการตรวจสอบว่าเป็น GET แมสเซสหรือไม่ ถ้าเราก็จะเก็บค่ายูอาร์แอล เอาไว้

5.3 สรุปและรายงานผล

ส่วนนี้โปรแกรมจะทำการแสดงผลการดักจับโดยแบ่งออกเป็น 3 กลุ่ม คือ การแสดงผลการใช้งานโปรโตคอลเอชทีทีพี การแสดงผลการใช้งานโปรโตคอลเอ็มเอสเอ็นและการแสดงผลการใช้งานโปรโตคอลอื่นๆ ในส่วนของการรายงานผล จะมีการรายงานผลอยู่ 2 แบบ คือ การรายงานผลในรูปกราฟ โปรแกรมสามารถสร้างกราฟโดยใช้ไลบรารี DISLIN ช่วยในการสร้างกราฟ และการรายงานผลในรูปของเอกสารเอ็กเซลล์ (Excel File)

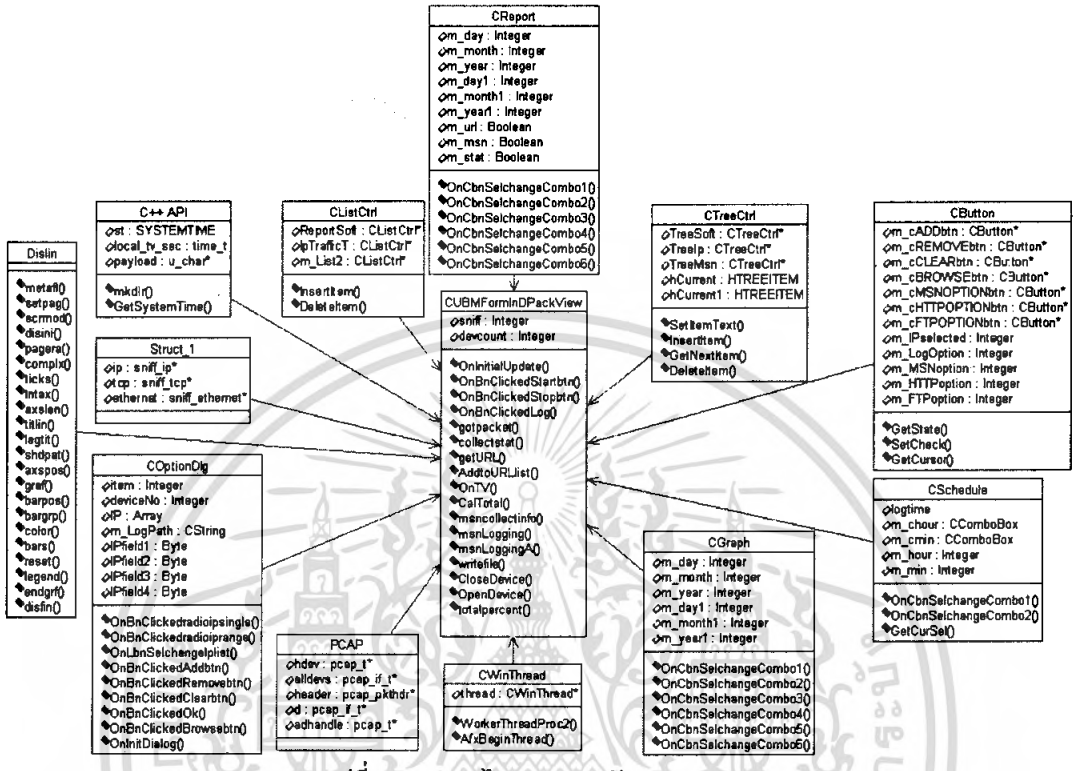


รูปที่ 5.17 ข้อมูลการใช้งานทั่วไป

รูปที่ 5.18 ข้อมูลการใช้งานเว็บไซต์

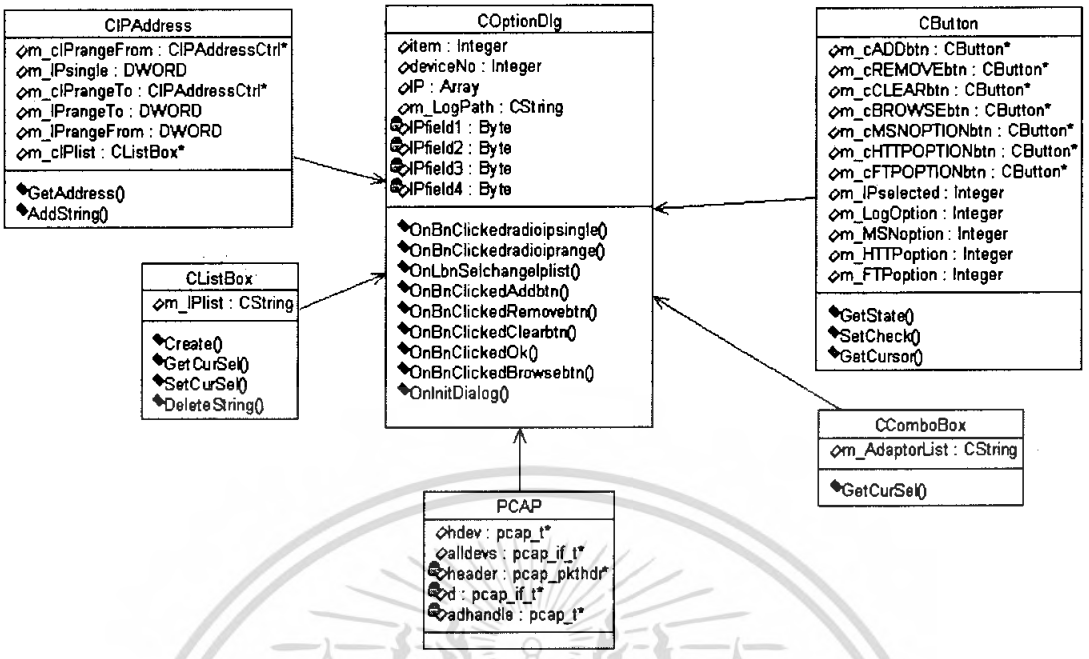
5.4 การออกแบบระบบ

5.4.1 คลาสไดอะแกรมของระบบ



รูปที่ 5.19 คลาสไดอะแกรมหลักของระบบ

1) CUBMFormInDPackView ทำหน้าที่ควบคุมการแสดงผลในหน้าต่างหลัก
2) CListCtrl ทำหน้าที่แสดงผลลัพธ์ในรูปแบบลิสต์
3) CTreeCtrl ทำหน้าที่แสดงผลลัพธ์ในรูปแบบโครงสร้างต้นไม้
4) CButton ทำหน้าที่ควบคุมการทำงานของปุ่มควบคุมต่างๆ
5) C++API ทำหน้าที่เรียกดูเวลาปัจจุบันของระบบ
6) COptionDlg ทำหน้าที่ในการสร้างหน้าต่างการตั้งค่าของผู้ใช้และส่งการตั้งค่ากลับไปยังโปรแกรมหลัก
7) CSchedule ทำหน้าที่ในการตั้งค่าช่วงเวลาในการดักจับแพ็กเกจ และบันทึกข้อมูล
8) CGraph ทำหน้าที่ในการตั้งค่าช่วงเวลาในการสร้างกราฟ
9) CReport ทำหน้าที่ในการตั้งค่าช่วงเวลาที่ต้องการออกรายงาน
10) CWinThread ทำหน้าที่ในการเรียกใช้งานการดักจับแพ็กเกจ
11) Struct_1 ประกอบด้วยโครงสร้างข้อมูลต่างๆเพื่อช่วยในการดักจับแพ็กเกจ
12) PCAP ประกอบด้วยโครงสร้างข้อมูลต่างๆเพื่อช่วยในการดักจับแพ็กเกจ
13) DISLIN ประกอบด้วยโครงสร้างข้อมูลต่างๆเพื่อช่วยในการสร้างกราฟ



รูปที่ 5.20 คลาสไดอะแกรมของคลาสที่ใช้ในการตั้งค่า

1). COptionDlg ทำหน้าที่ในการสร้างหน้าต่างการตั้งค่าของผู้ใช้และส่งการตั้งค่ากลับไปยังโปรแกรมหลัก

2). CIPAddress ทำหน้าที่ในการรับค่าไอพีแอดเดรสจากผู้ใช้และแสดงผล

3). CListBox ทำหน้าที่ในการแสดงค่าไอพีแอดเดรสที่ผู้ใช้เลือก

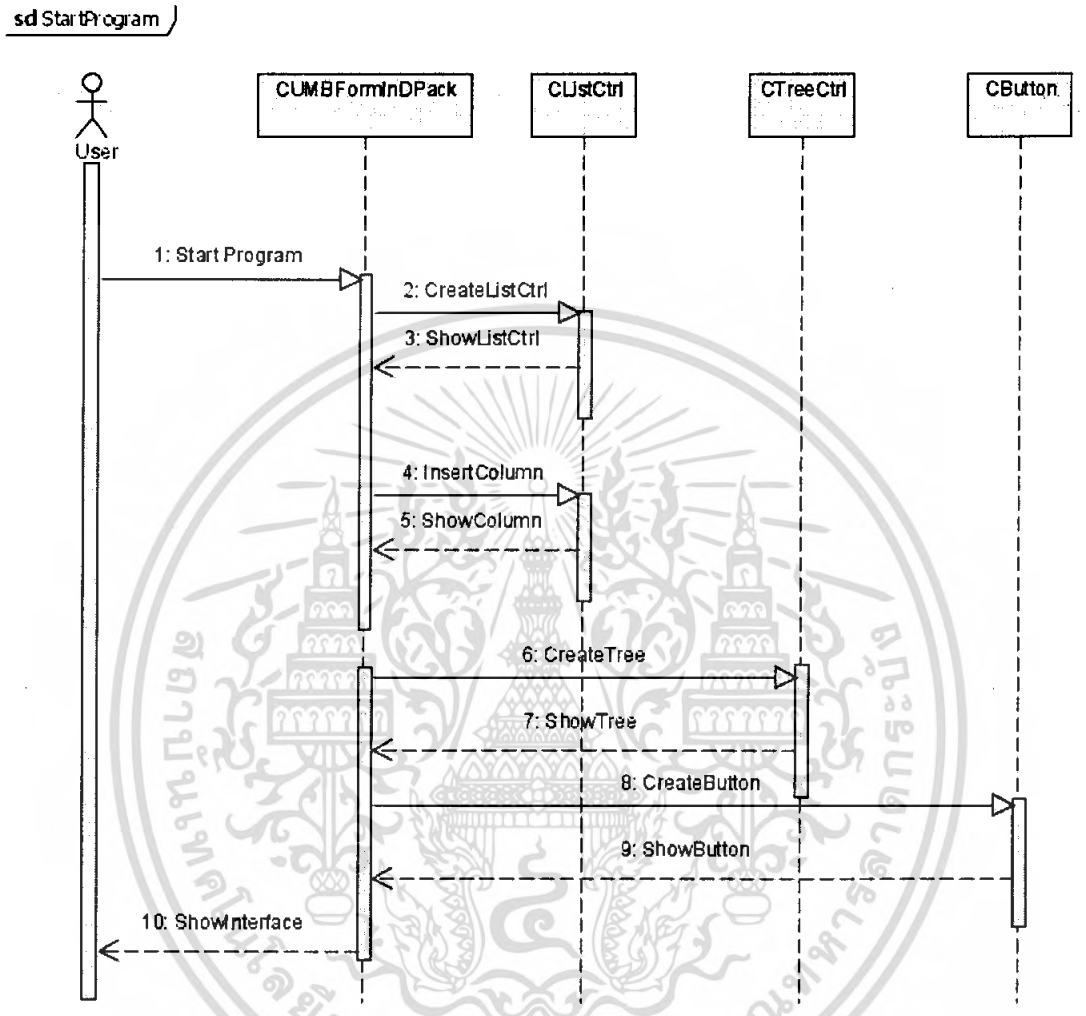
4). CButton ทำหน้าที่ควบคุมการทำงานของ Button และ Radio Button

5). CComboBox ทำหน้าที่ในการระบุอุปกรณ์เน็ตเวิร์คที่ผู้ใช้เลือกใช้งาน

6). PCAP ประกอบด้วยโครงสร้างข้อมูลต่างๆเพื่อช่วยในการดักจับแพ็กเกจ

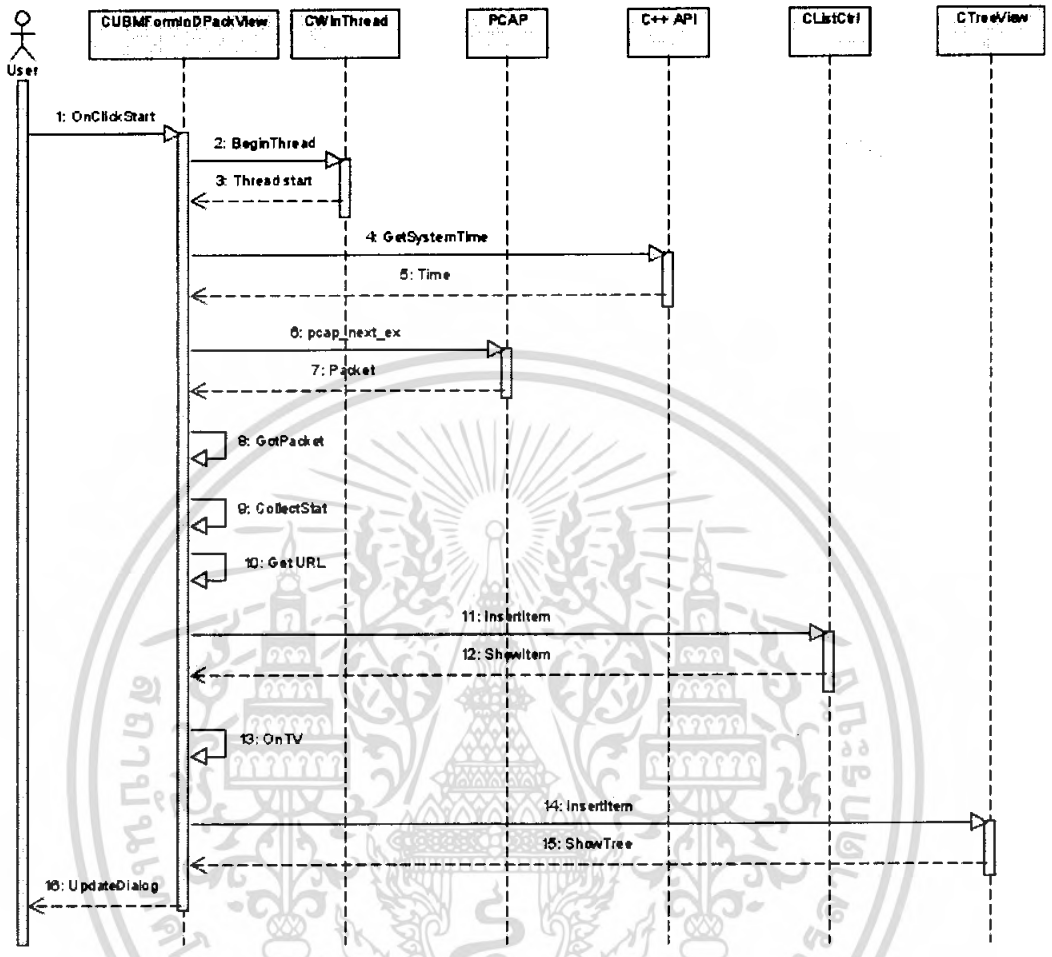
5.4.2 ซีเควนซ์ไดอะแกรมของระบบ

5.4.2.1 ซีเควนซ์ไดอะแกรมเมื่อเปิดใช้งานโปรแกรม

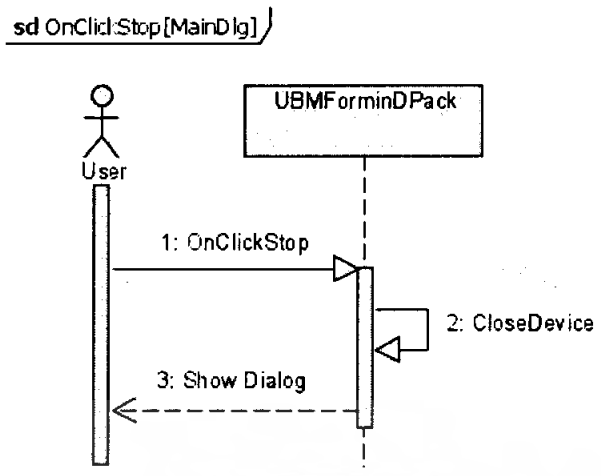


รูปที่ 5.21 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานเริ่มเปิดโปรแกรม

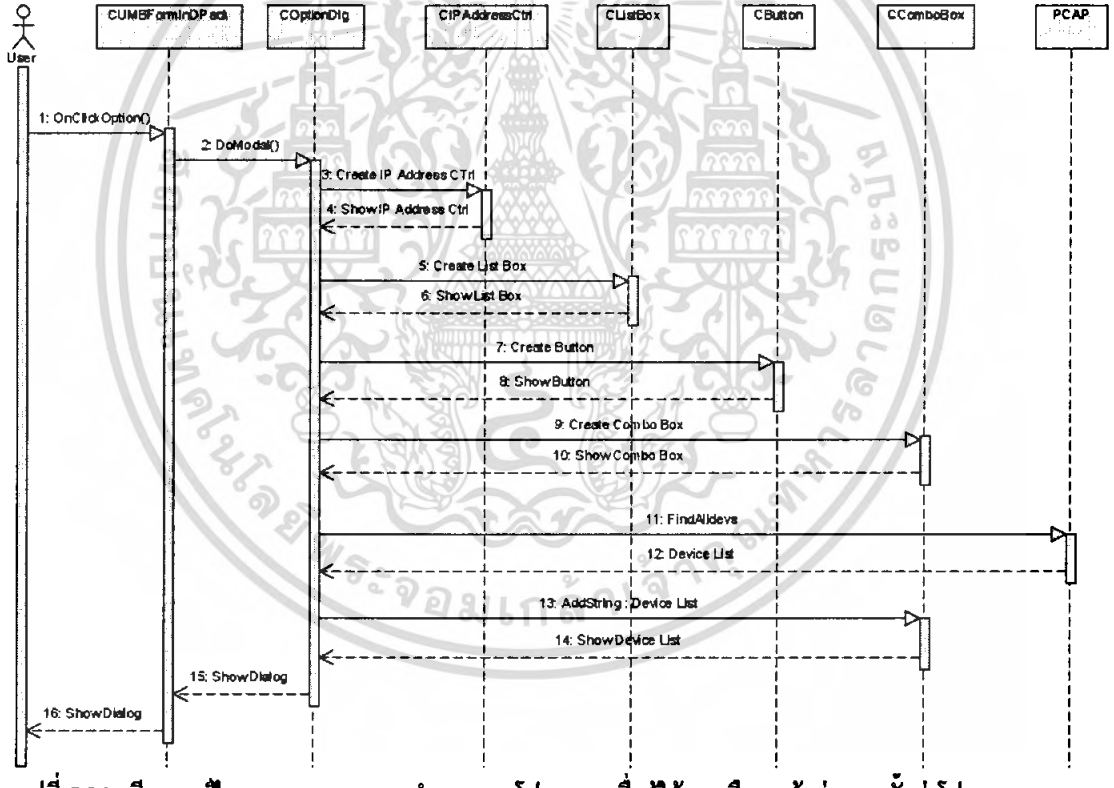
5.4.2.2 ซีเควนซ์ไดอะแกรมแสดงการใช้งานในหน้าต่างหลัก



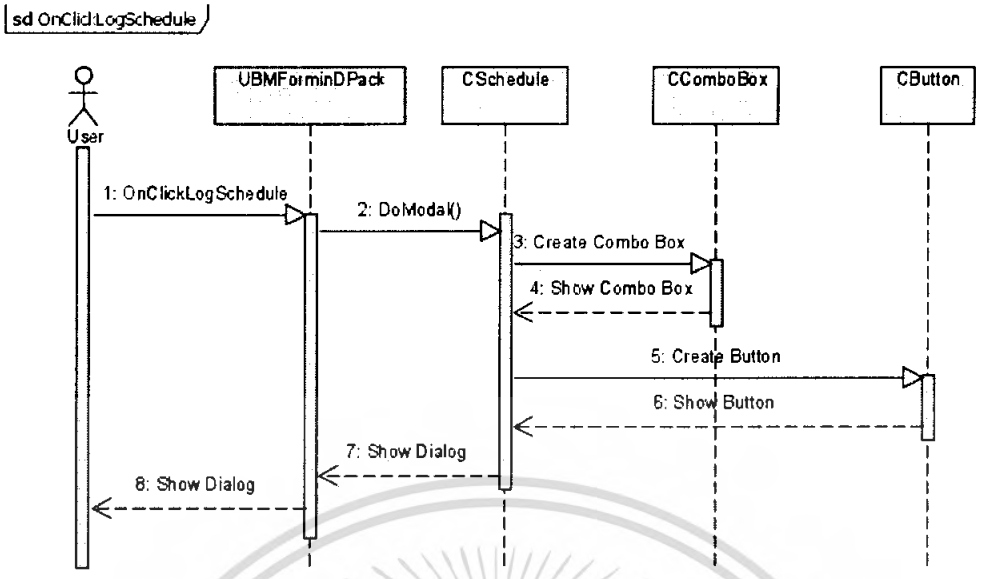
รูปที่ 5.22 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานกดปุ่มให้โปรแกรมเริ่มทำงาน



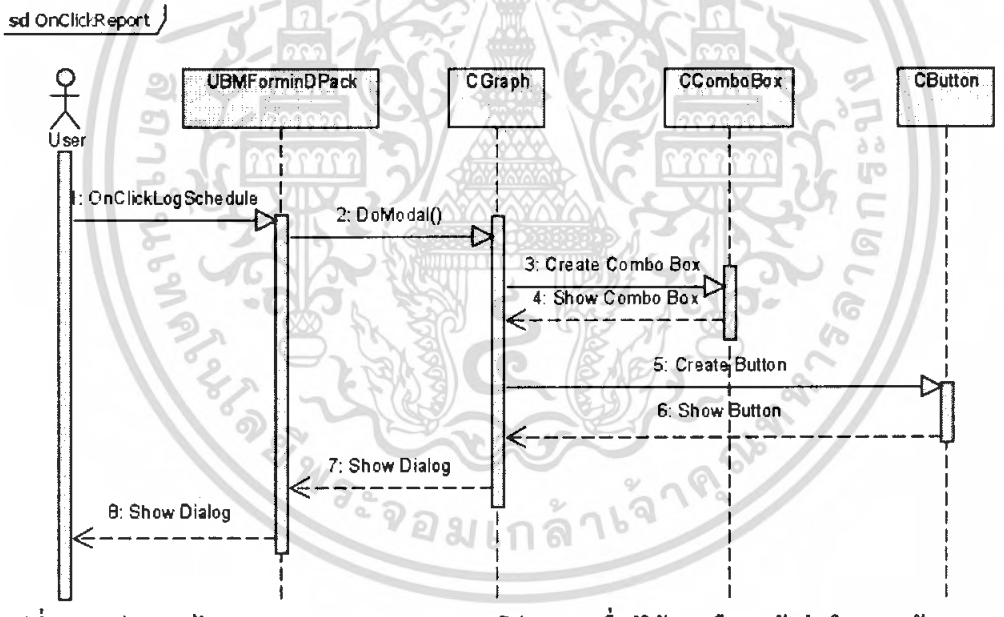
รูปที่ 5.23 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานกดปุ่มให้โปรแกรมหยุดทำงาน



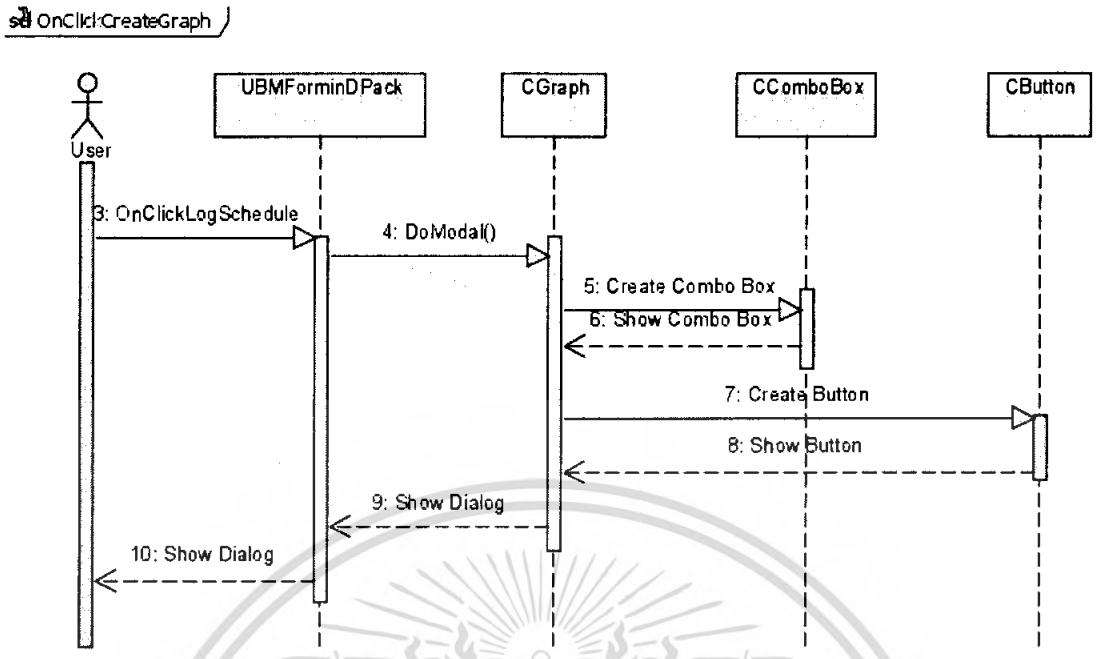
รูปที่ 5.24 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานเลือกหน้าต่างการตั้งค่าโปรแกรม



รูปที่ 5.25 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานเลือกระยะเวลาในการเก็บล็อกไฟล์

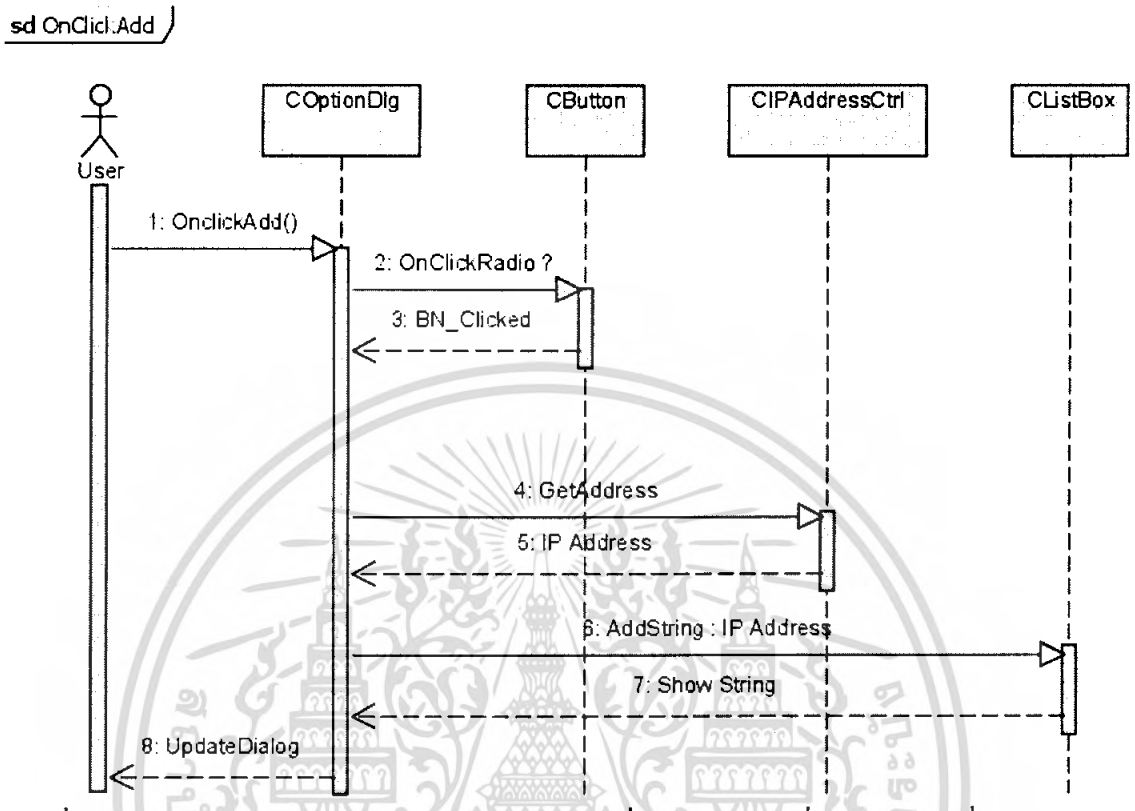


รูปที่ 5.26 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานเลือกหน้าต่างในการสร้างรายงาน

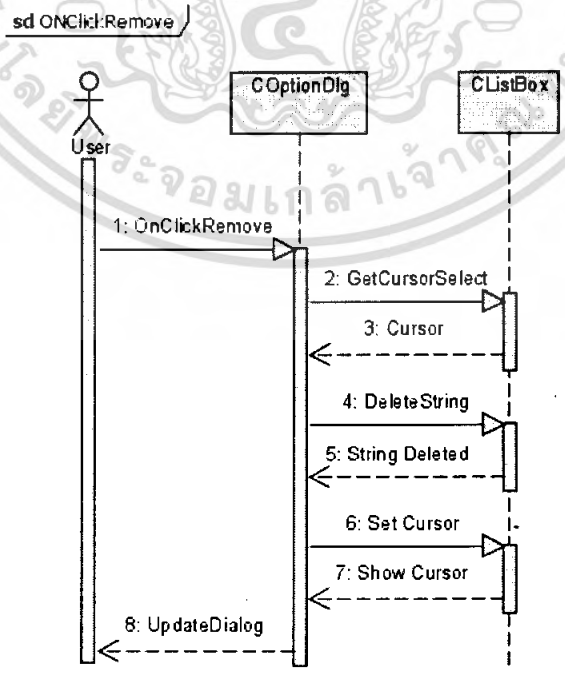


รูปที่ 5.27 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานเลือกหน้าต่างในการสร้างกราฟ

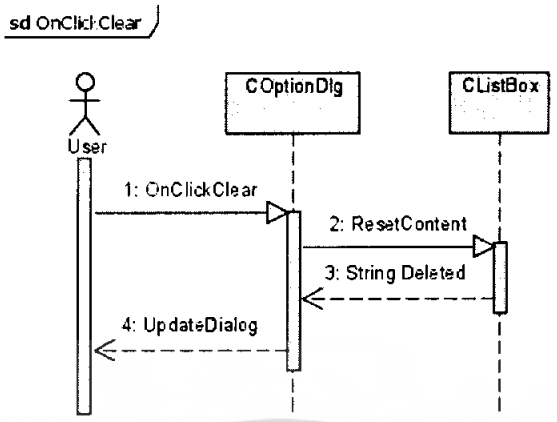
5.4.2.3 ซีเควนซ์ไดอะแกรมแสดงการใช้งานในหน้าต่างออฟชั่น



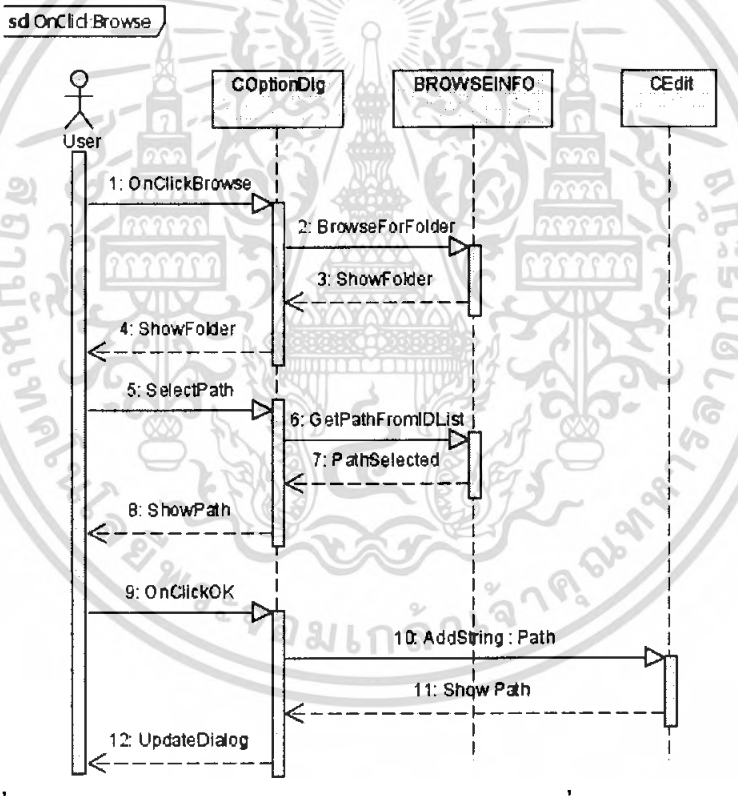
รูปที่ 5.28 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานทำการเพิ่มไอพีแอดเดรสที่ต้องการ



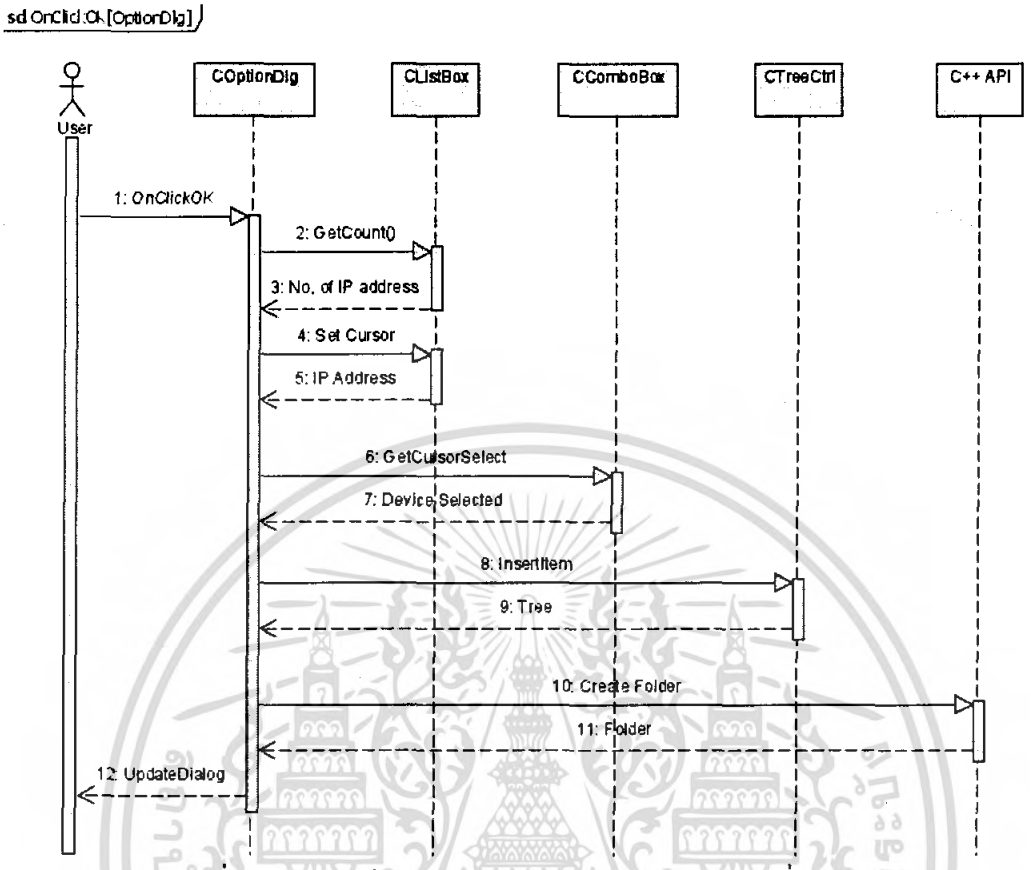
รูปที่ 5.29 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานทำการลบไอพีแอดเดรสที่ไม่ต้องการ



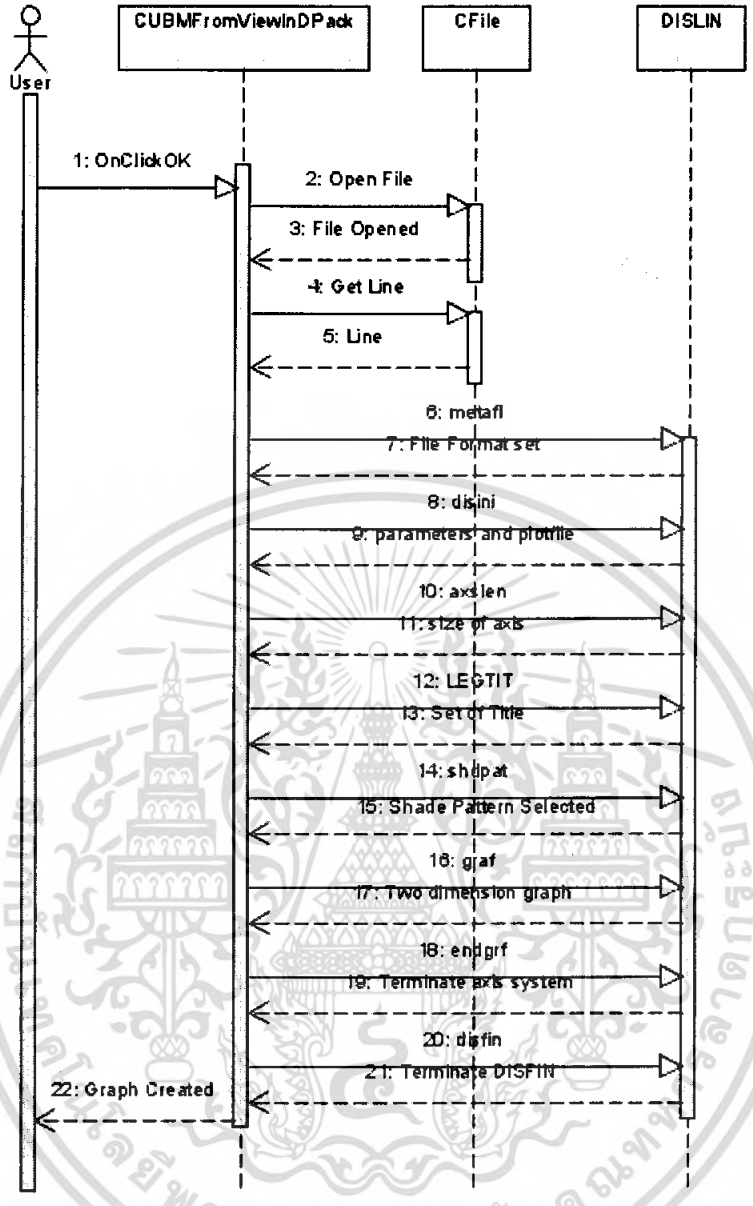
รูปที่ 5.30 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานทำการลบไอพีแอดเดรสทั้งหมด



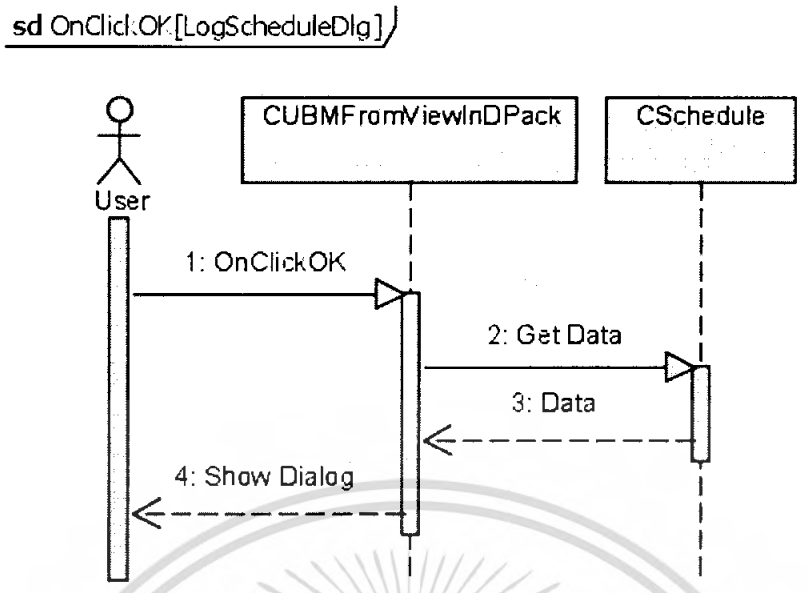
รูปที่ 5.31 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานทำการเลือกตำแหน่งที่ต้องการเก็บข้อมูลการดักจับ



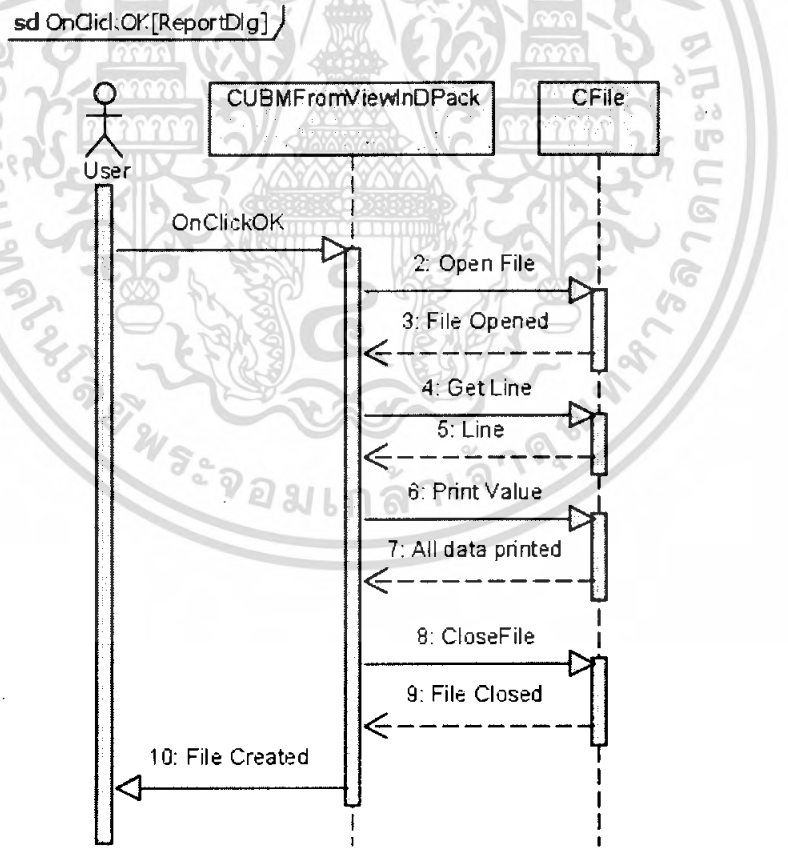
รูปที่ 5.32 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานทำการตั้งค่าในหน้าต่างออฟชั่นเสร็จสิ้น



รูปที่ 5.33 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานต้องการให้โปรแกรมสร้างกราฟ



รูปที่ 5.34 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานทำการตั้งระยะเวลาในการเก็บล็อกไฟล์



รูปที่ 5.35 ซีเควนซ์ไดอะแกรมแสดงการทำงานของโปรแกรมเมื่อผู้ใช้งานต้องการให้โปรแกรมสร้างไฟล์รายงาน

บทที่ 6
ผลการทดลอง

เราจะทำการทดลองตรวจสอบการใช้งานของไอพี 161.246.5.97 – 161.246.5.98 โดยจะทำการบันทึกล็อกไฟล์ที่ ณ เวลา 22 นาฬิกา 35 นาที โดยมีลำดับขั้นตอนการใช้งานดังนี้

6.1 ขั้นตอนการทำการดักจับแพ็กเก็ต

6.1.1 ทำการตั้งค่าการดักจับ

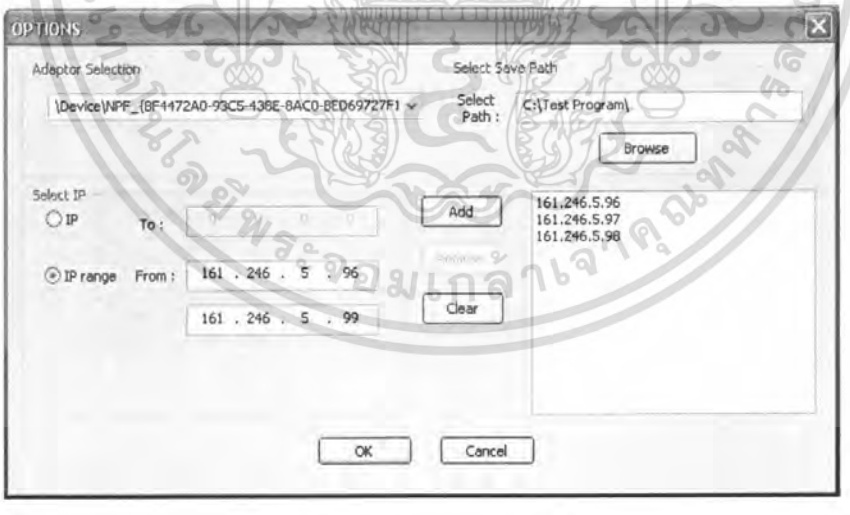
ทำการตั้งค่าการดักจับ โดยเข้าไปที่ปุ่ม Options แล้วทำตามขั้นตอนดังนี้

1). ทำการตั้งค่าของอินเตอร์เฟสที่ต้องการใช้งาน

2). ทำการตั้งค่าช่วงของไอพีที่ต้องการทำการดักจับ

3). ทำการเลือกไดเรคทอรี่ที่ต้องการบันทึกล็อกไฟล์

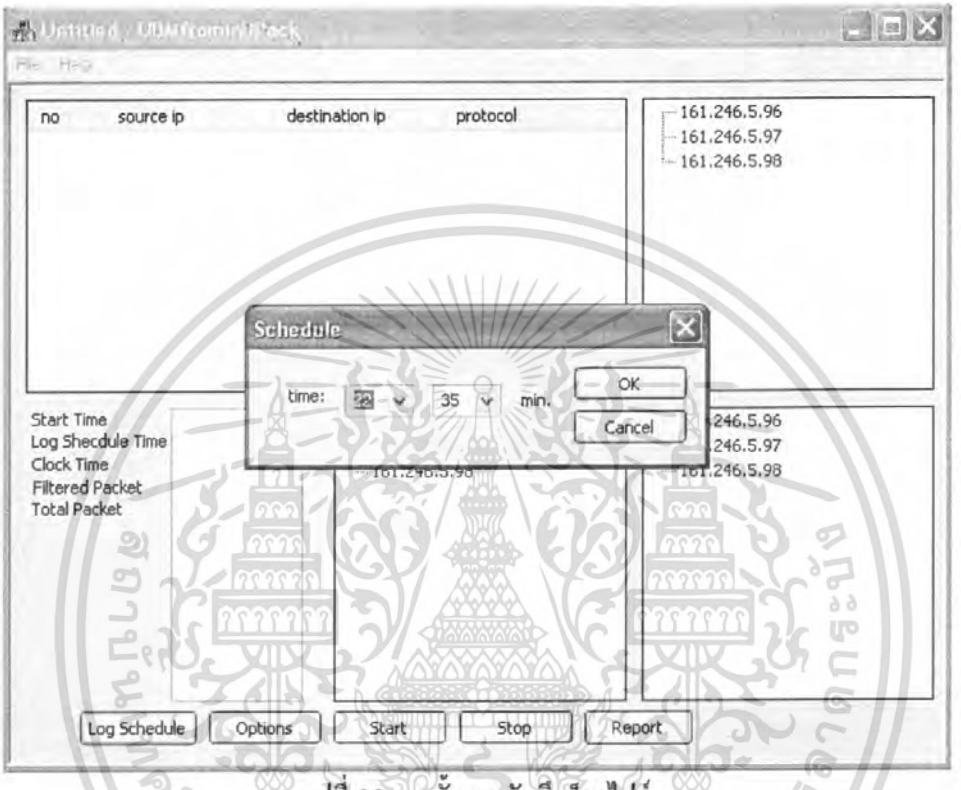
4). กด OK เพื่อตกลง



รูปที่ 6.1 หน้าต่างแสดงการตั้งค่าดักจับ

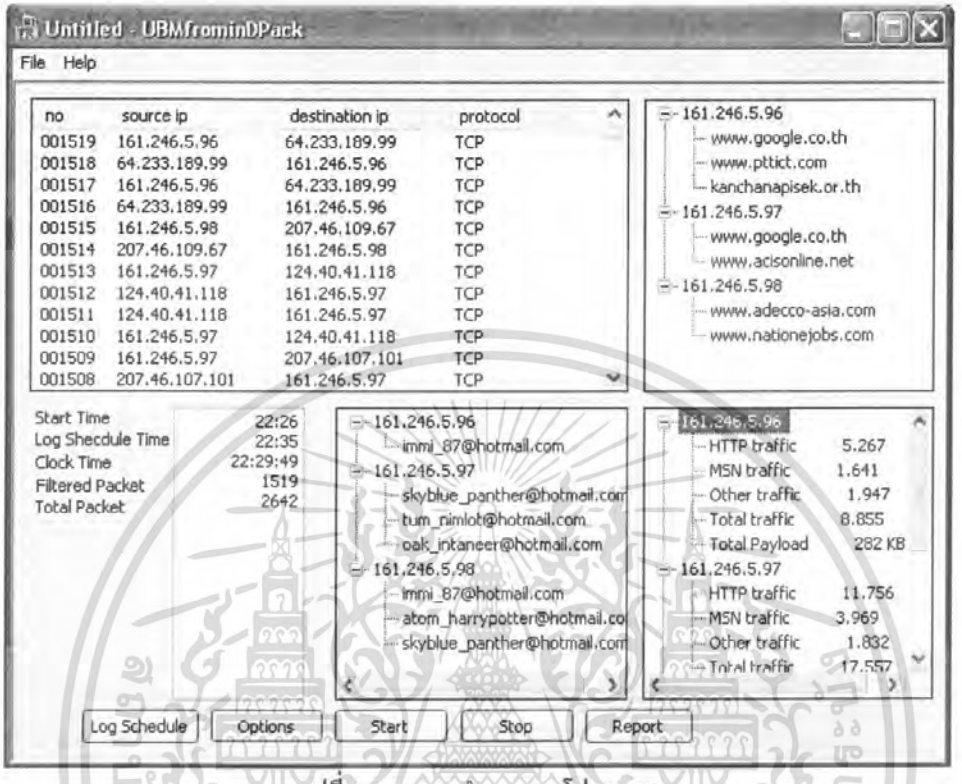
6.1.2 ทำการตั้งเวลาที่ต้องการให้เก็บบันทึกลงไฟล์

โดยปกติจะเลือกช่วงเวลาที่เครือข่ายมีการใช้งานน้อยที่สุด เพราะอาจจะมีการล่าช้าของการทำงาน ทำให้อาจจะมีการดร็อปแพ็กเก็ตที่อินเตอร์เฟสทิ้งไปบางส่วนถ้าทำงานไม่ทัน



รูปที่ 6.2 การตั้งเวลาบันทึกล็อกไฟล์

6.1.3 กด Start เพื่อเริ่มการทำงาน



รูปที่ 6.3 ผลการทำงานของโปรแกรม

6.2 ขั้นตอนการแสดงรายงาน

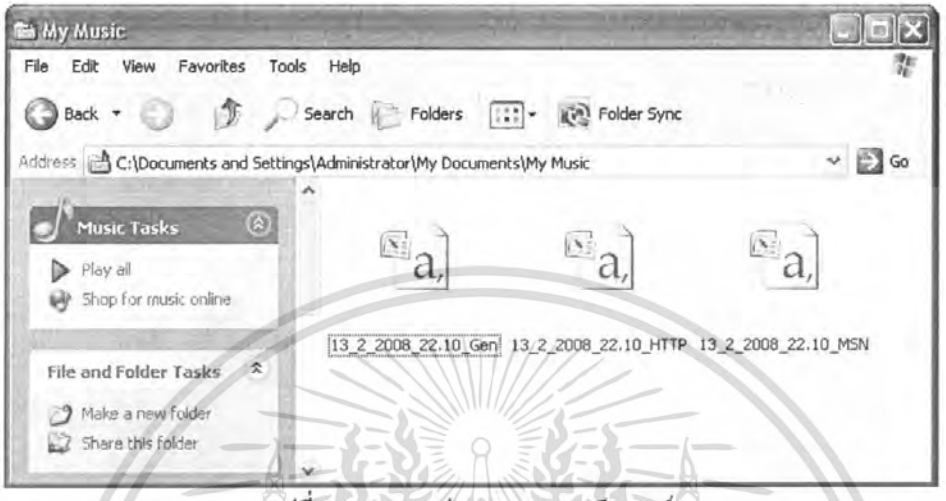
1. ไปที่ปุ่ม Report
2. ทำการเลือกประเภทของรายงานที่ต้องการ แบ่งได้เป็น 4 แบบ คือ
    1). URL List เป็นรายงานแสดงการใช้งานเว็ปไซด์ โดยประมวลผลเป็นรายเดือน (ตั้งค่าเฉพาะเดือนหรือปีที่ต้องการ)
    2). MSN List เป็นรายงานแสดงการใช้งานเอ็มเอสเอ็น โดยประมวลผลเป็นรายเดือน (ตั้งค่าเฉพาะเดือนหรือปีที่ต้องการ)
    3). General Statistic เป็นรายงานแสดงเปอร์เซ็นต์การใช้งานระบบเป็นรายวัน (ตั้งค่าวัน เดือนหรือปีที่ต้องการ)
    4). Graphเป็นการแสดงกราฟแสดงเปอร์เซ็นต์การใช้งานระบบเป็นรายวันแต่สามารถแสดงเปรียบเทียบได้ภายในเดือนเดียวกันเท่านั้น(ตั้งค่าช่วงของวันในเดือนหรือปีที่ต้องการ)
3. ทำการเลือกไดเรคทอรี่ที่ได้มีการบันทึกล็อกไฟล์เอาไว้
4. ทำการเลือกไดเรคทอรี่ที่ต้องการบันทึกรายงานที่ต้องการ

ทำการเลือกช่วงของวัน เดือน ปี ให้เหมาะสมกับการออกรายงานแต่ละประเภทดังรูป

รูปที่ 6.4 การตั้งค่ารายงาน

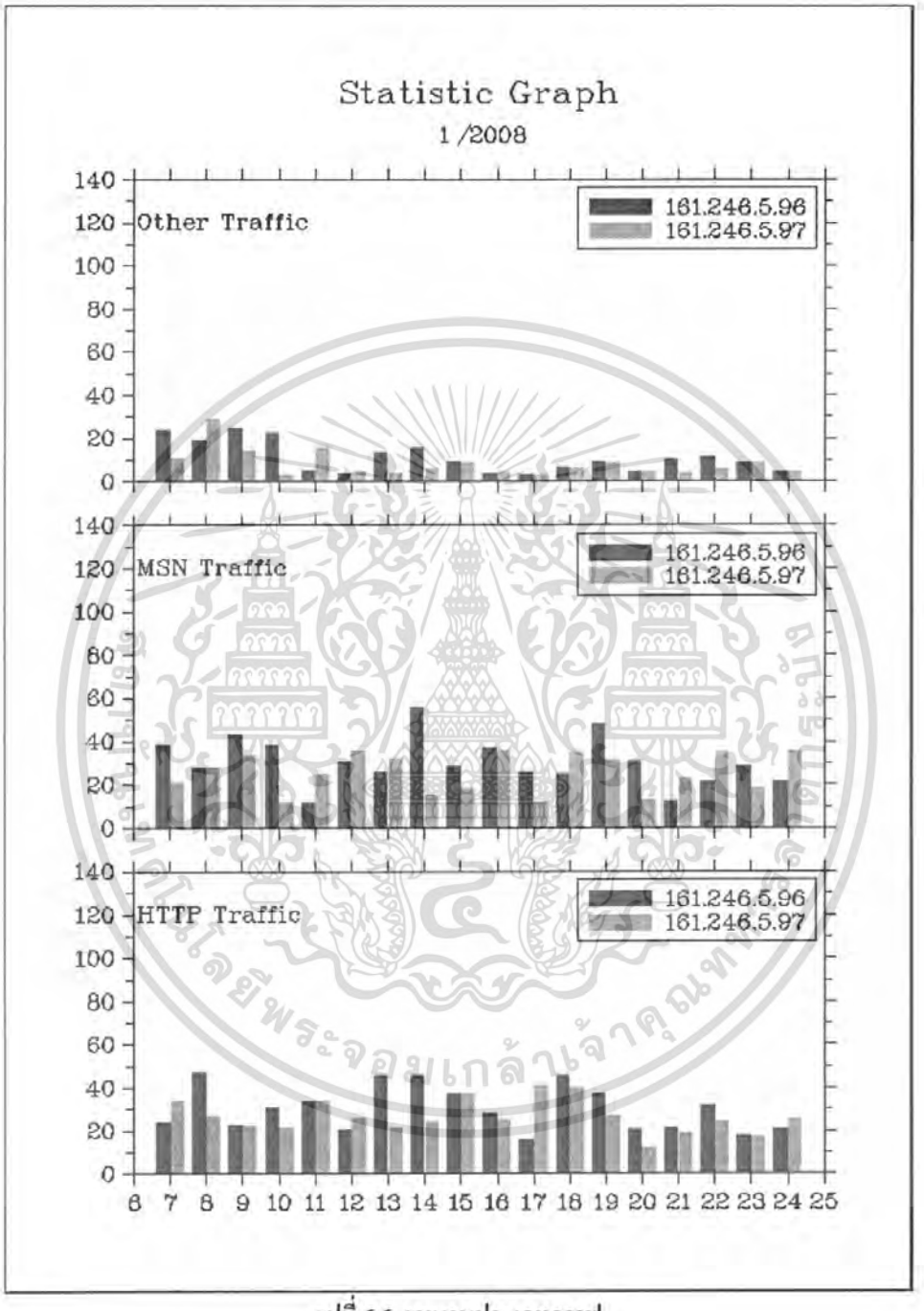
6.3 รายงานประเภทต่างๆ

6.3.1 รายงานประเภทเอกสารเอ็กเซลล์



รูปที่ 6.5 รายงานประเภทเอกสารเอ็กเซลล์

6.3.2 รายงานประเภทกราฟ



รูปที่ 6.6 รายงานประเภทกราฟ

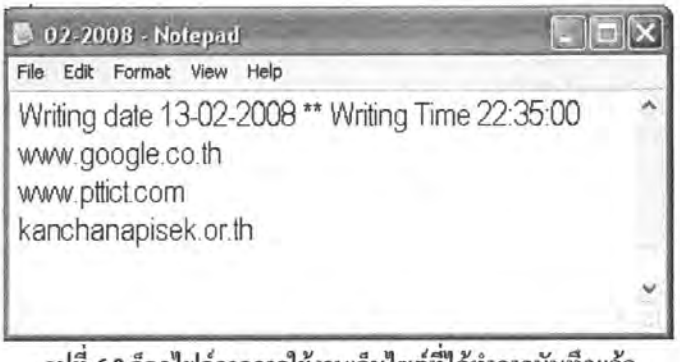
6.4 ล็อกไฟล์

6.4.1 ล็อกไฟล์ของโปรโตคอลเอ็มเอสเอ็น

รูปที่ 6.7 หน้าต่างการใช้งานโปรแกรมเอ็มเอสเอ็น

รูปที่ 6.8 ล็อกไฟล์จากการใช้งานโปรแกรมเอ็มเอสเอ็นที่ได้ทำการบันทึกแล้ว

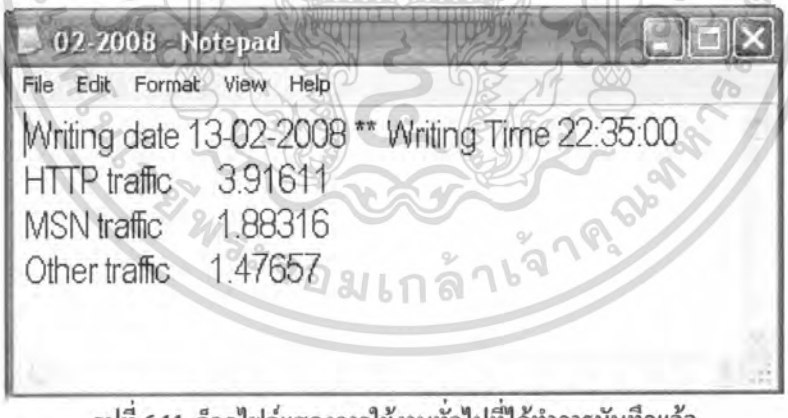
6.4.2 ล็อกไฟล์ของโปรโตคอลเอชทีทีพี



รูปที่ 6.9 ล็อกไฟล์จากการใช้งานเว็บไซต์ที่ได้ทำการบันทึกแล้ว

6.4.3 ล็อกไฟล์ของการใช้งานทั่วไป

รูปที่ 6.10 การใช้งานเว็บไซต์



รูปที่ 6.11 ล็อกไฟล์แสดงการใช้งานทั่วไปที่ได้ทำการบันทึกแล้ว

บทที่ 7
วิจารณ์และสรุปผล

7.1 ผลสรุป

โครงงานนี้ ได้จัดทำขึ้นเพื่อทำการบันทึกข้อมูลการใช้งานต่างๆของผู้ใช้งานเพื่อการตรวจสอบทั้ง ณ เวลาปัจจุบันและตรวจสอบย้อนหลัง โปรแกรมที่ได้จัดทำขึ้นสามารถทำงานได้ค่อนข้างน่าพอใจ ในการตรวจสอบการใช้งานโปรโตคอลเอชทีทีพี (HTTP) และเอ็มเอสเอ็น (MSN) ทำให้เราสามารถได้รู้ว่ามีการเข้าใช้งานเว็บไซต์ใด หรือสนทนากับใคร มากน้อยเท่าไร สามารถตรวจดูได้ว่าผู้ใช้งานมีปริมาณการใช้งานเครือข่ายเมื่อเทียบกับเครือข่ายรวมเทียบเป็นกี่เปอร์เซ็นต์ นอกจากนี้ยังสามารถออกรายงานได้ทำให้สามารถนำผลที่ได้บันทึกไว้ไปวิเคราะห์ต่อไปได้ แต่อย่างไรก็ตามโครงงานยังมีส่วนที่สามารถจะพัฒนาได้อีกมาก

7.2 ปัญหาและอุปสรรค

1). เนื่องจากผู้ทำวิจัยไม่มีความรู้พื้นฐานเกี่ยวกับการเขียนโปรแกรมในด้านนี้ทำให้ต้องใช้เวลาศึกษาเป็นเวลานาน

2). ผู้ทำวิจัยต้องใช้เวลาในการศึกษาและใช้งานไลบรารี Winpcap เป็นเวลานานเพราะเป็นไลบรารีที่ที่มีผลกระทบกับการพัฒนาโปรแกรมขึ้นมาอย่างมาก

3). ในเครือข่ายที่มีปริมาณการสื่อสารเป็นจำนวนมาก อาจทำให้การดักจับมีความคลาดเคลื่อน

4). การโปรแกรมขั้นต้นที่ได้จัดทำเป็นการใช้งานบัฟเฟอร์ถ้ามีปริมาณผู้ใช้งานมากอาจทำให้บัฟเฟอร์เต็ม

5). ในส่วนของการทำงาน หากทำงานในการดักจับทั้งแบบปกติและ MSN ไปพร้อมๆกันอาจจะกินทรัพยากรมาก

6). การทำงานของเอมเอสเอ็น บางกรณีจะไม่มี message มาครบตามกฎทำให้ทำงานได้ไม่ครบถ้วน

7). ในกรณีที่ระบบมีปริมาณ traffic มาก อาจจะทำให้มีการดรอปบางแพ็กเก็ตทิ้ง อาจทำให้มีการขาดข้อมูลบางส่วนที่สำคัญ ทำให้โปรแกรมทำงานได้ไม่สมบูรณ์

7.3 แนวทางในการพัฒนาต่อ

7.3.1 ส่วนของการรวบรวมข้อมูลของระบบ

- เพิ่มเติมโปรโตคอลที่ทำการตรวจสอบให้เหมาะสมกับการทำงานที่ต้องการ
- เพิ่มเติมส่วนของการรวบรวมข้อมูลปริมาณ payload ของแต่ละรายผู้ใช้งาน
- เพิ่มรายละเอียดในการบันทึกข้อมูลลงในล็อกไฟล์ เพื่อให้สามารถเก็บได้ละเอียดเฉพาะส่วนที่ต้องการหรือเน้นเป็นพิเศษ
- อาจพัฒนาการเก็บรวบรวมข้อมูลลงฐานข้อมูล ถ้ามีปริมาณการใช้งานมาก

7.3.2 ส่วนของการแสดงผล

- เพิ่มเติมความสามารถในการกรองการแสดงผลเฉพาะที่ต้องการเช่น การกรองจากคำ หรือไอพีที่ต้องการ เป็นต้น

7.3.3 ส่วนของการออกรายงาน

- เพิ่มเติมชนิดของรายงานเช่น เอกสารธรรมดา, pdf หรือ excel เป็นต้น
- เพิ่มความสามารถในการออกรายงานแบบกราฟให้มีหลายรูปแบบได้ เช่น กราฟวงกลม กราฟเส้น
- พัฒนารูปแบบการรายงานผลให้เป็นไปตามความต้องการ

บรรณานุกรม

นิรุธ อำนวยศิลป์. 2548. Visual C++ and MFC Programming. กรุงเทพฯ : ดวงกลมสมัย.

David Chapman. Teach Yourself Visual C++ 6 in 21 days.

A Division of Macmillan Computer Publishing, 1998.

"ArcObjects Online". [Online].

Available : http://edndoc.esri.com/arcobjects/8.3/default.asp?URL=/arcobjects/8.3/gettingstarted/vcppenv.htm#character_strings

"Charting Library C Chart in MFC C++ CPP Win32Win64". [Online].

Available : http://www.gigasoft.com/chartinglibrary.html

Chris Water. "Packetyzer". [Online].

Available : http://packetyzer.cvs.sourceforge.net/packetyzer/. 2003

"CodeProject: A Multiline Header Control Inside a CListCtrl". [Online].

Available : http://www.codeproject.com/KB/combobox/headerctrlex.aspx

"CodeProject: Beginners Guide to Dialog Based". [Online].

Available : http://www.codeproject.com/KB/dialog/dialogapptute2.aspx

"Data Conversion Reference". [Online].

Available : http://www.zegelin.com/computers_files/ref/DataConversionReference.htm

"DotMSN - .NET Messenger library". [Online].

Available : http://www.xihsolutions.net/dotmsn/

Eran Aharonovich. "Network Sniffer" [Online].

Available : www.codeproject.com/tools/Sniffer.asp?df=100&forumid=79319&exp=0&select=1003670. 2004

"E-SNIFF : The Embeded Ethernet Packet Sniffer"

Available : http://cegt201.bradley.edu/projects/proj2007/dsniff/docs.html

"FunctionX - Visual C++ - Property Sheet". [Online].

Available : http://functionx.com/visualc/articles/propsheet1.htm

Ilya Solnyshkin. " Sniffer80 - HTTP Sniffer .NET". [Online].

Available : http://www.codeprojcct.com/tools/sniffcr80.asp 2003

"Input/Output with files". [Online].

Available : http://www.cplusplus.com/doc/tutorial/files.html

maszup. "Simple Packet Sniffer in VC++ .Net" [Online].

Available : www.codeproject.com/Purgatory/Sznyfer.asp 2005.

"Max Planck Institute for Solar System Research". [Online].

Available : http://www.mps.mpg.de/dislin/downloads.html. 2007

"Microsoft". [Online].

Available : http://support.microsoft.com/kb/308407

"Microsoft Source Code Control Interface (MSSCCI)". [Online].

Available : http://alinconstantin.members.winisp.net/webdocs/scc/MSSCCI.htm

"Microsoft visual C++". [Online].

Available : http://www.functionx.com/visualc/.

"MSN Messenger Protocol - Notification - Authentication". [Online].

Available : http://www.hypothetic.org/docs/msn/notification/authentication.php . 2003

"MSN Messenger Protocol - Switchboard - Example Session". [Online].

Available : http://www.hypothetic.org/docs/msn/switchboard/example_session.php .

2003

"Obtaining CPU Time Used in a C++ Program". [Online].

Available : http://www.intranet.csupomona.edu/~hnriley/www/timing_cpu_cpp.html

"Programming with pcap". [Online].

Available : http://www.tcpdump.org/pcap.htm

"Retrieving Conversations from MSN Messenger". [Online].

Available : http://www.codeproject.com/KB/cpp/msnchattext.aspx

"SecureSphere". [Online].

Available : http://www.securesphere.net/main.php?win=l&nav=n. 2005

"Sniff'em Tool of the Trade" . [Online].

Available : http://www.sniff-em.com/. 2005

"The Boost Graph Library". [Online].

Available : http://www.boost.org/libs/graph/doc/index.html

"The TCP/IP Guide - HTTP Request Message Format". [Online].

Available : http://www.tcpipguide.com/free/t_HTTPRequestMessageFormat.htm

"Visual C++ Developer Center". [Online].

Available : http://msdn2.microsoft.com/en-us/library/482ck6x8(VS.80).aspx

"VelocityReviews". [Online].

Available : http://www.velocityreviews.com/forums/t286206-how-to-get-the-cpu-time-with-vc-win.html

"WinDump". [Online].

Available : http://www.winpcap.org/windump/install/default.htm. 2007

"WinPcap : The Windows Packet Capture Library". [Online].

Available : http://www.winpcap.org. 2007

"[Winpcap-users] how to open the IP packet data". [Online].

Available : http://www.winpcap.org/pipermail/winpcap-users/2006-February/000667.html

"[Winpcap-users] filtering traffic using payload contents". [Online].

Available : http://www.winpcap.org/pipermail/winpcap-users/2006-April/000901.html

"[Winpcap-users] Retriving the data from UDP packet". [Online].

Available : http://www.winpcap.org/pipermail/winpcap-users/2007-September/002104.html

"Wiretapped". [Online].

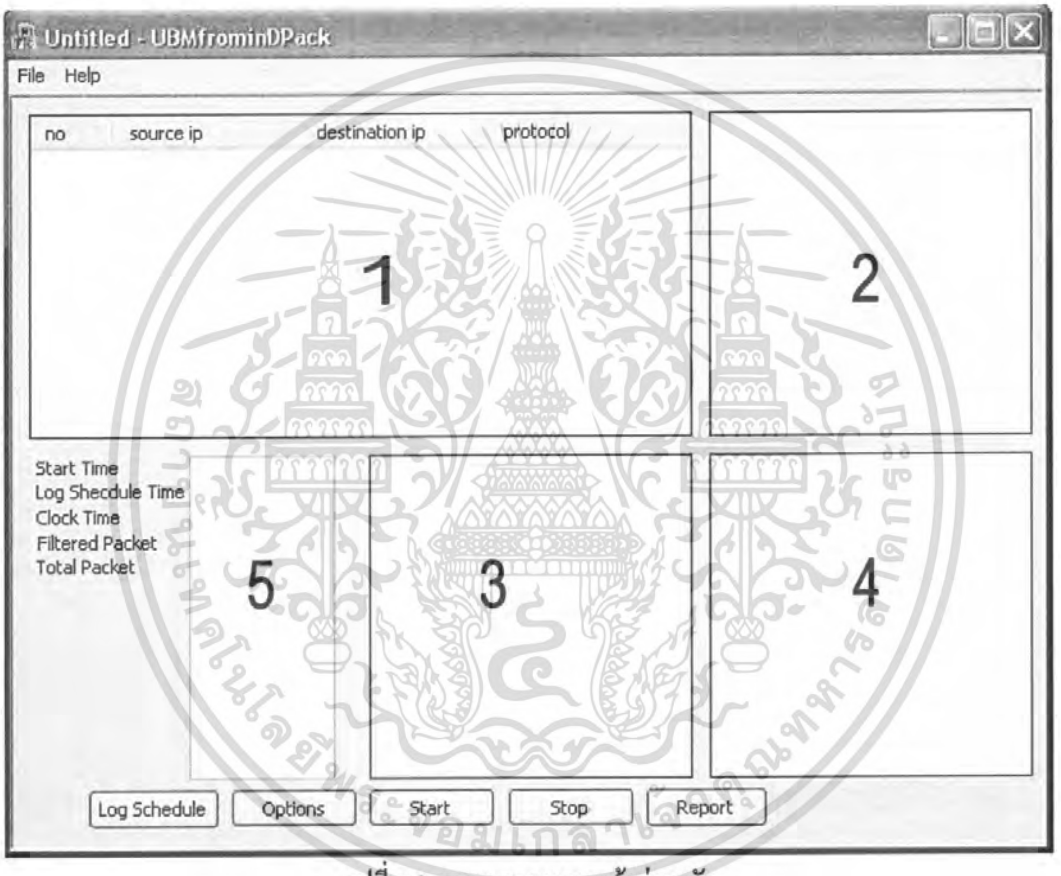
Available : http://www.wiretapped.net/indexes/packet-capture.html.

ภาคผนวก ก.

ตัวอย่างการทำงานของระบบ

ก.1 ตัวอย่างการทำงานในเบื้องต้นของ User Interface

ก.1.1 การแสดงผลบนหน้าต่างหลัก



รูปที่ ก.1 การแสดงผลบนหน้าต่างหลัก

ในหน้าต่างนี้จะทำการแสดงผลการทำงานของโปรแกรมทั้งหมดโดยจะประกอบด้วยส่วนที่ใช้แสดงผลทั้งหมด 5 ส่วนดังนี้

1). ส่วนที่ 1

ส่วนนี้จะทำการแตกแพ็กเก็ตออกมาแสดงให้เห็นถึง source IP, destination IP และprotocol ของแต่ละแพ็กเก็ต

no	source ip	destination ip	protocol
000206	202.44.5.18	161.246.5.99	TCP
000205	202.44.5.18	161.246.5.99	TCP
000204	161.246.5.99	202.44.5.18	TCP
000203	202.44.5.18	161.246.5.99	TCP
000202	161.246.5.99	202.44.5.18	TCP
000201	202.44.5.18	161.246.5.99	TCP
000200	161.246.5.99	202.44.5.18	TCP
000199	202.44.5.18	161.246.5.99	TCP
000198	202.44.204.85	161.246.5.99	TCP
000197	161.246.5.99	202.44.5.18	TCP
000196	202.44.5.18	161.246.5.99	TCP
000195	161.246.5.99	202.44.5.18	TCP

รูปที่ ก.2 แพ็กเก็ตที่ดักจับได้

2). ส่วนที่ 2

ส่วนนี้จะแสดงรายชื่อเว็บที่เข้าใช้งานแยกตามไอพีที่ทำการดักจับ

รูปที่ ก.3 เว็บไซต์ที่เข้าใช้งาน

3). ส่วนที่ 3

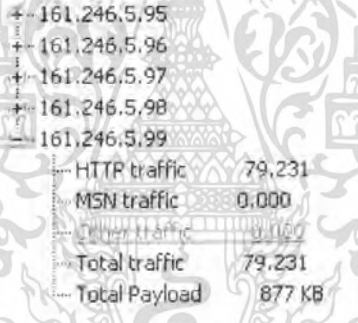
ส่วนนี้จะแสดงรายชื่อคู่สนทนาของโปรแกรมเอ็มเอสเอ็นแยกตามไอพีแอดเดรสที่ทำการดักจับ

รูปที่ ก.4 รายชื่อคู่สนทนาของแต่ละไอพีแอดเดรส

4). ส่วนที่ 4

ส่วนนี้จะแสดงผลรวมของการใช้งานเครือข่ายแยกตามไอพีที่ดักจับ โดยผลรวมของการใช้งานเครือข่ายจะประกอบด้วย

- เปอร์เซ็นต์การใช้งานของพอร์ต 80 (HTTP Traffic) เทียบกับปริมาณการใช้งานในเครือข่ายทั้งหมด
- เปอร์เซ็นต์การใช้งานของพอร์ต 1863 (MSN Traffic) เทียบกับปริมาณการใช้งานในเครือข่ายทั้งหมด
- เปอร์เซ็นต์การใช้งานของพอร์ตนอกเหนือจากนี้ (Other Traffic) เทียบกับปริมาณการใช้งานในเครือข่ายทั้งหมด
- เปอร์เซ็นต์การใช้งานรวมของไอพี (Total Traffic)
- ปริมาณของข้อมูล (Payload)

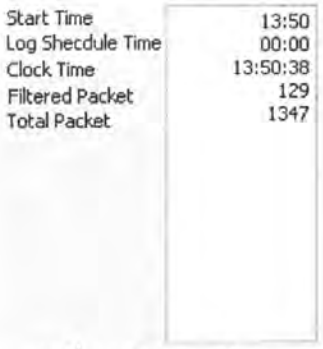


รูปที่ ก.5 การใช้งานเครือข่ายทั้งหมด

5). ส่วนที่ 5

ส่วนนี้จะประกอบด้วย

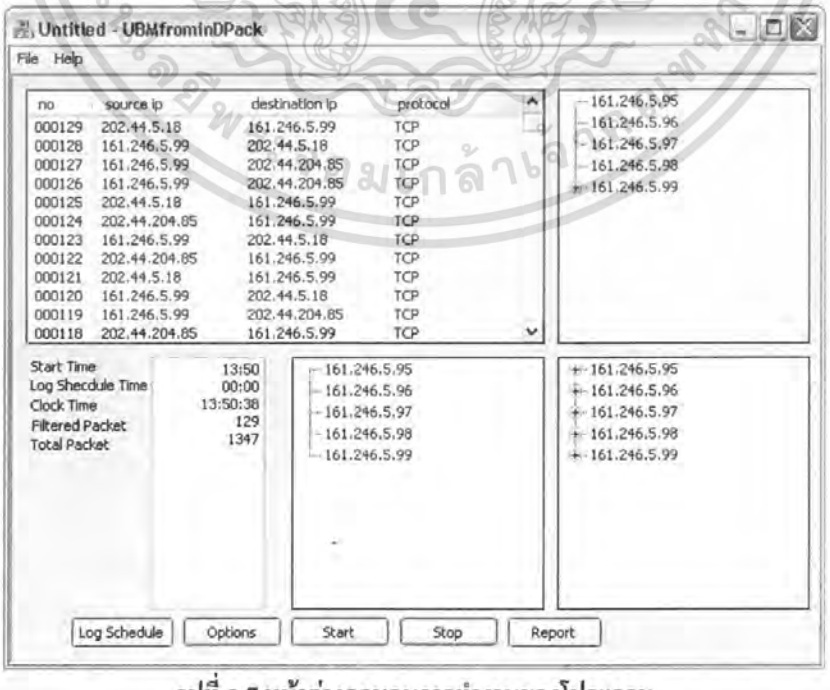
- เวลาที่โปรแกรมเริ่มทำงาน (Start Time)
- เวลาที่ทำการตั้งไว้เพื่อให้ทำการบันทึกล็อกไฟล์ (Log Schecdule Time)
- เวลาปัจจุบันที่มีการใช้งาน (Clock Time)
- ปริมาณแพ็กเก็ตที่ถูกกรองออกมาตามไอพีที่ต้องการดักจับ (Filtered Packet)
- ปริมาณแพ็กเก็ตทั้งหมดภายในเครือข่ายที่ทำการดักจับ (Total Packet)



รูปที่ ก.6 ข้อมูลการดักจับ

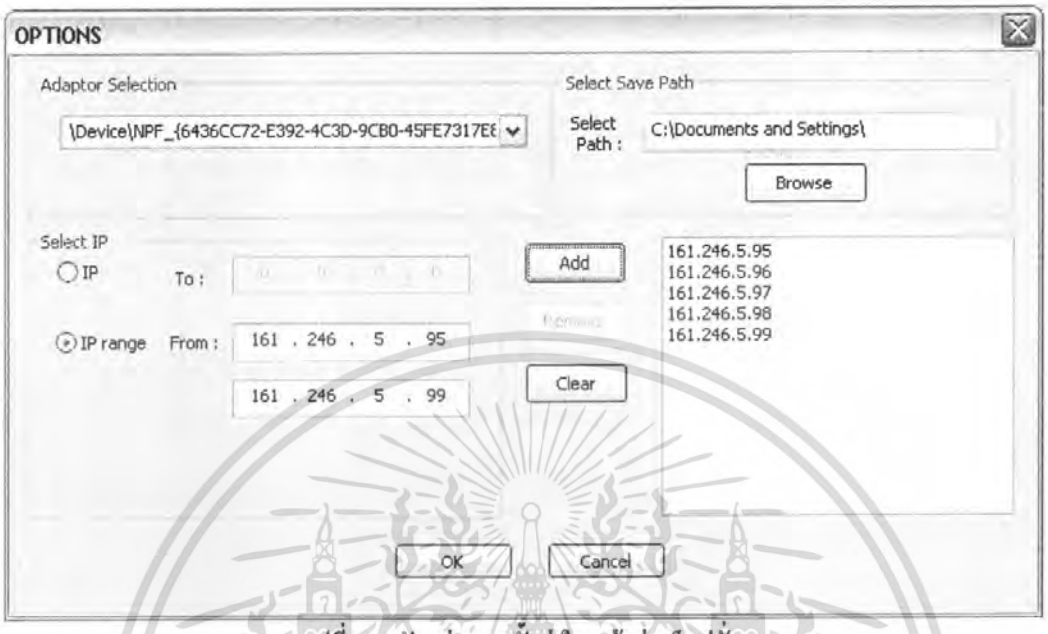
สำหรับการควบคุมการทำงานของโปรแกรมทั้งหมดจะทำงานที่หน้านี้เป็นหน้าหลัก โดยจะมีปุ่มควบคุมต่างๆดังนี้

1. ปุ่ม Start เพื่อทำการเริ่มทำการดักจับแพ็กเก็ต
2. ปุ่ม Stop เพื่อทำการหยุดการทำงาน
3. ปุ่ม Options เพื่อทำการตั้งค่าการต่างๆที่ใช้ในการดักจับ ได้แก่
   - การเลือกใช้อินเตอร์เฟสที่ใช้ในการดักจับ (Adaptor Selection)
   - การเลือกไอพีที่ต้องการทำการดักจับ (IP Capturing)
   - การเลือกไดเร็คทอรี่ที่ต้องการทำการบันทึกล็อกไฟล์
4. ปุ่ม Log Schedule เพื่อทำการตั้งเวลาที่ต้องการบันทึกล็อกไฟล์
5. ปุ่ม Report เพื่อทำการเลือกแสดงหรือออกรายงานตามรายไอพีที่ใช้งานและช่วงเวลา



รูปที่ ก.7 หน้าต่างควบคุมการทำงานของโปรแกรม

ก.1.2 การตั้งค่าในหน้าต่างอ็อฟชั่น



รูปที่ ก.8 ตัวอย่างการตั้งค่าในหน้าต่างอ็อฟชั่น

ในหน้าต่างนี้เป็นการตั้งค่าการทำงานเริ่มต้นของโปรแกรมเกือบทั้งหมดดังที่ได้กล่าวไปแล้ว

ส่วนที่ 1 การเลือกใช้อินเตอร์เฟส (Adaptor Selection)

ส่วนที่ 2 การเลือกไดเร็คทอรี่ที่ต้องการทำการบันทึกล็อกไฟล์ (Select Save Path)

ส่วนที่ 3 การเลือกไอพีที่ต้องการทำการดักจับ (Select IP)

ในส่วนของการกำหนดไอพีนั้นจะทำได้สองแบบคือ

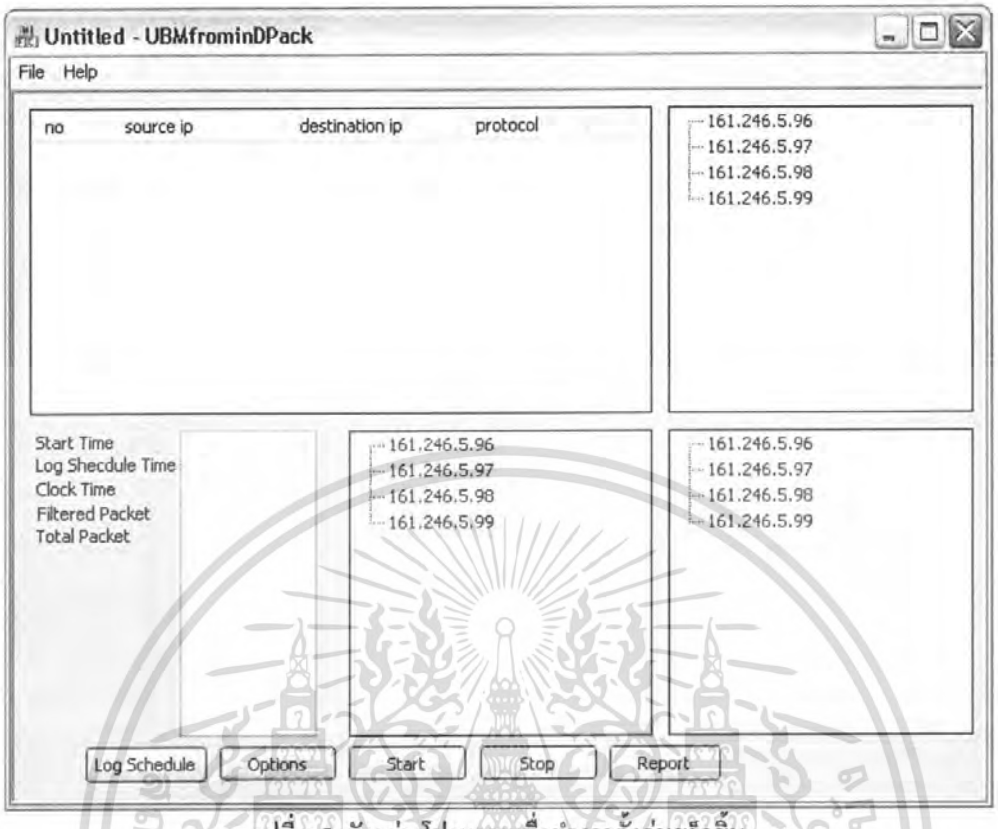
1). เลือกที่ Radio Button ที่ชื่อ IP

จะสามารถกำหนดไอพีได้ทีละ 1 ไอพี โดยเมื่อทำการพิมพ์หมายเลขไอพีลงในช่องที่จัดเตรียมไว้แล้วให้กดปุ่ม Add เพื่อทำการแสดงไอพีนั้นบนรายการ (IP List) ทางด้านขวามือ

2). เลือกที่ Radio Button ที่ชื่อ IP range

จะสามารถทำการกำหนดไอพีเป็นช่วงได้ โดยทำการใส่ไอพีเริ่มต้นทางช่องด้านบน และใส่ไอพีสุดท้ายในช่องด้านล่างจากนั้นกดปุ่ม Add เพื่อทำการแสดงรายการเช่นเดียวกับข้อแรก

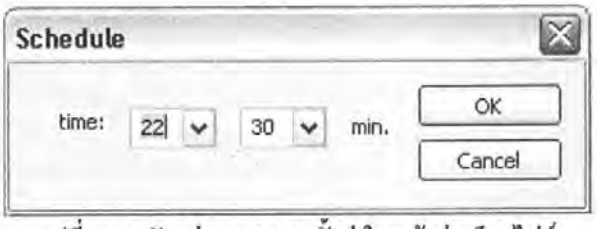
ในกรณีที่ต้องการลบรายชื่อไอพีทั้งหมดออกจากรายการสามารถใช้ปุ่ม Clear ออกได้ หรือถ้าหากต้องการลบออกบางไอพีก็ทำการกดที่ไอพีนั้นๆแล้วกดปุ่ม Remove



รูปที่ ก.9 ตัวอย่างโปรแกรมเมื่อทำการตั้งค่าเสร็จสิ้น

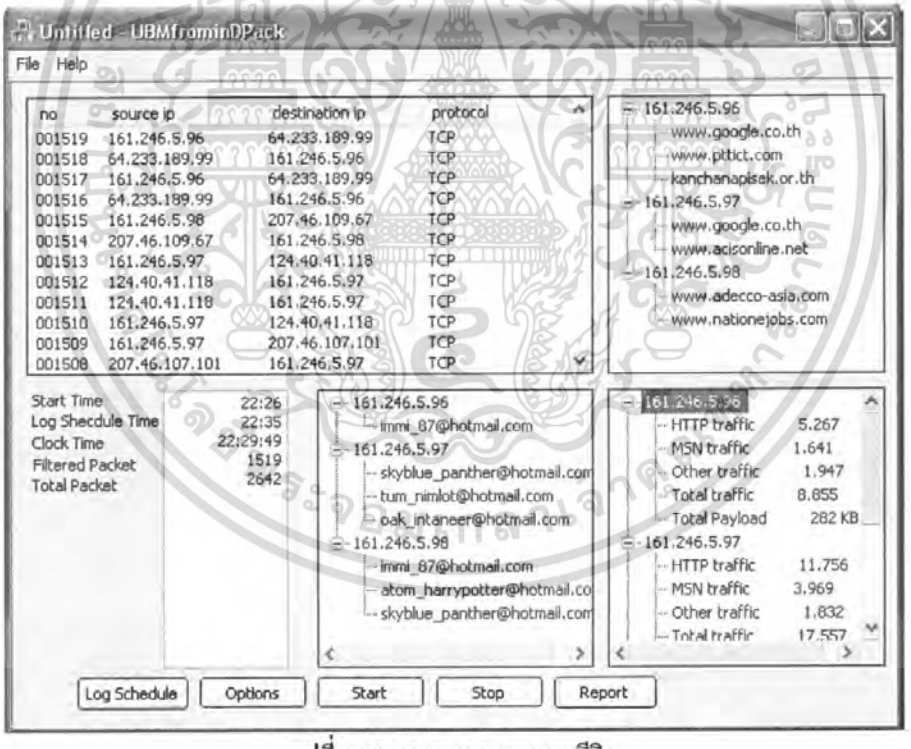
เมื่อผู้ใช้งานทำการตั้งค่าต่างๆในหน้าต่าง Options เสร็จสิ้น โปรแกรมจะแสดงไอพีแอดเดรสทั้งหมดที่ผู้ใช้งานต้องการดักจับแพ็กเก็ตในหน้าต่างแสดงผล หลังจากนั้นถ้าผู้ใช้ต้องการตั้งเวลาการบันทึกล็อกไฟล์ให้ทำการตั้งค่าได้ที่ปุ่ม Log Schedule

ก.1.3 การตั้งค่าในหน้าต่างล็อกไฟล์



รูปที่ ก.10 ตัวอย่างแสดงการตั้งค่าในหน้าต่างล็อกไฟล์

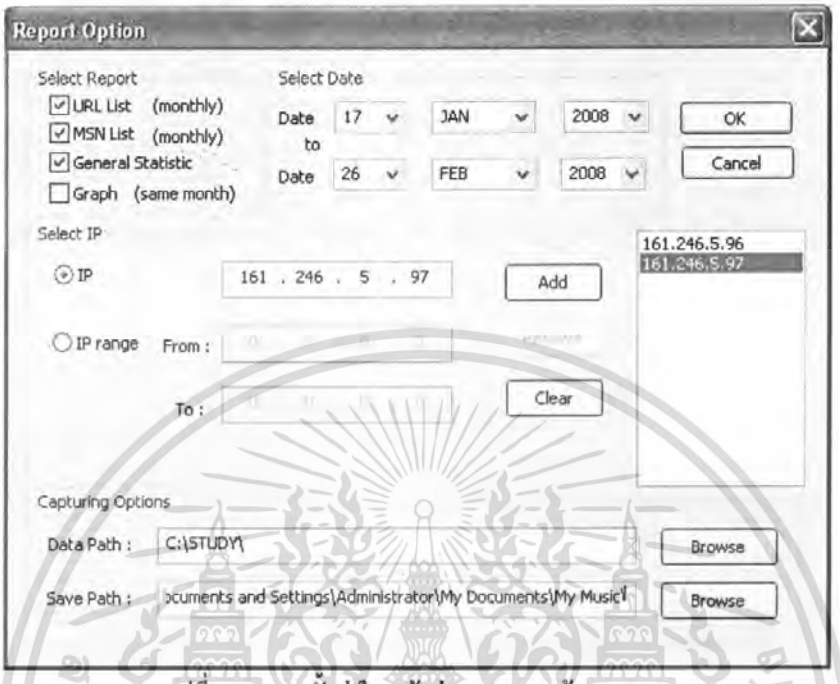
หน้าต่างนี้จะให้ผู้ใช้งานทำการตั้งเวลาให้โปรแกรมทำการเก็บการดักจับทั้งหมดลงล็อกไฟล์ ว่าโปรแกรมจะทำการจัดเก็บ ณ เวลาใด โดยทั่วไปแล้วผู้ใช้งานจะทำการจัดเก็บล็อกไฟล์ในเวลาที่มีความคับคั่งของเครือข่ายน้อยที่สุด ถ้าไม่มีการตั้งค่า ค่าเวลาจะอยู่ที่ 00:00 หรือเวลาเที่ยงคืนเป็นค่าดีฟอลล์



รูปที่ ก.11 การแสดงผลแบบทรีวิว

สำหรับการแสดงผลจะทำงานแบบทรี คือเราสามารถที่จะทำการกดเพื่อขยายรายละเอียดของแต่ละไอพีออกมาดูได้

ก.1.4 การตั้งค่าในหน้าต่างของการสร้างรายงาน



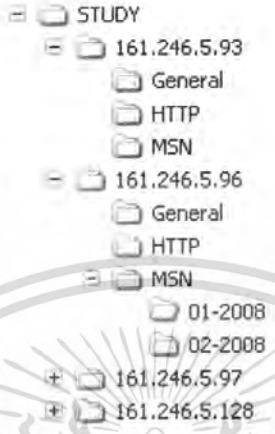
รูปที่ ก.12 การตั้งค่าในหน้าต่างของการสร้างรายงาน

หน้าต่างนี้จะทำการออกรายงานต่างๆทั้งหมดไม่ว่าจะเป็นกราฟหรือการออกเอกสารในรูปแบบของเอกสารเอ็กเซลล์ (ไฟล์นามสกุล .csv) โดยจะมีหน้าต่างที่ทำการตั้งค่าได้ดังนี้

1. Select Report ทำการเลือกรายงานที่ต้องการซึ่งประกอบด้วย
   - URL List คือการออกไฟล์เอกเซลในส่วนของการเข้าใช้งานเว็บ
   - MSN List คือการออกไฟล์เอกเซลในส่วนของการงานเอ็มเอสเอ็น
   - General Statistic คือการออกไฟล์เอกเซลแสดงเปอร์เซ็นต์การใช้งานต่างๆ
   - Graph คือการสร้างกราฟตามเปอร์เซ็นต์แสดงการใช้งาน
2. Select Date ทำการเลือกช่วงเวลาที่ต้องการแสดงรายงาน
   - Select IP ทำการเลือกไอพีที่ต้องการออกรายงาน โดยมีการทำงานเหมือนกันการเลือกไอพีที่ต้องการทำการดักจับในหน้าต่าง Options
   - Data Path ทำการเลือกไดเร็คทอรี่ที่เราได้ทำการบันทึกล็อกไว้เพื่อที่จะดึงข้อมูลมาใช้ในการออกรายงาน
   - Save Path ทำการเลือกไดเร็คทอรี่ที่เราต้องการที่จะบันทึกรายงานเอาไว้

ก.2 การเก็บล็อกของการใช้งานและไฟล์รายงาน

ก.2.1 ล็อกไฟล์



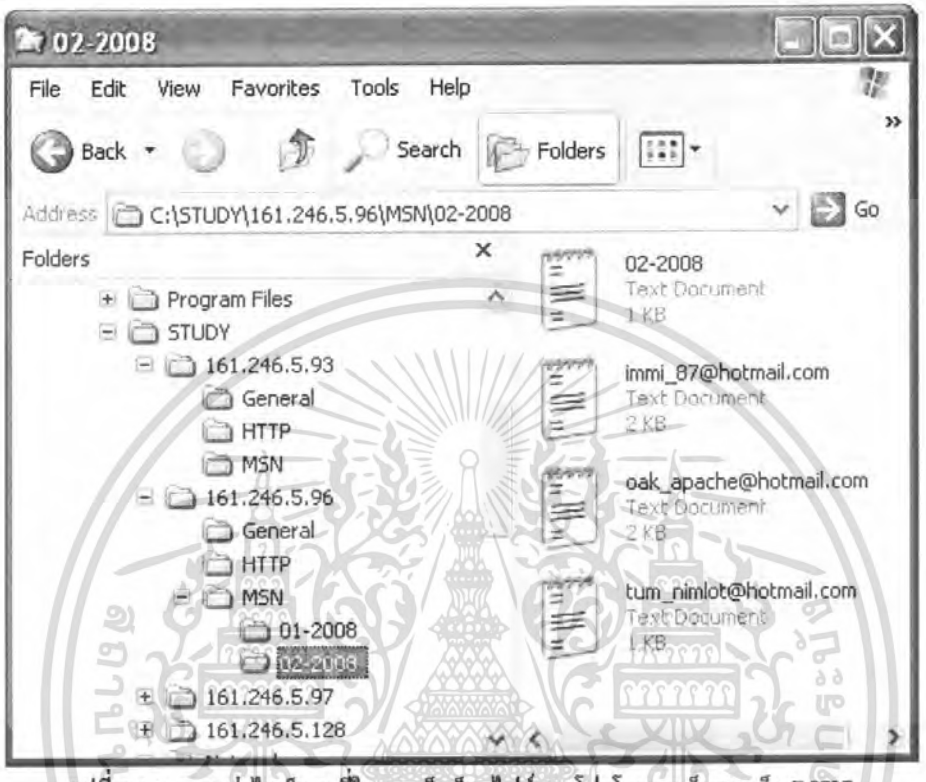
รูปที่ ก.13 การจัดเก็บล็อกไฟล์

การจัดเก็บล็อกไฟล์จะจัดเก็บไว้ที่ไดเร็คทอรี่ที่ได้ทำการเลือกไว้ในตอนตั้งค่าโปรแกรม ดังรูปได้ทำการบันทึกล็อกเอาไว้ที่ C:\STUDY\ จะเห็นได้ว่า จะมีโฟลเดอร์ของทุกๆไอพีที่เราได้ทำการดักจับเอาไว้ แยกย่อยลงไปจะแยกตามโปรโตคอล

ในส่วนของโปรโตคอล HTTP และ General Statistic นั้น จะมีไฟล์ย่อยๆแบ่งตามเดือนที่ได้ทำการจัดเก็บดังรูป

รูปที่ ก.14 การแบ่งไดเร็คทอรี่ในการเก็บล็อกไฟล์ของโปรโตคอลเอชทีทีพี

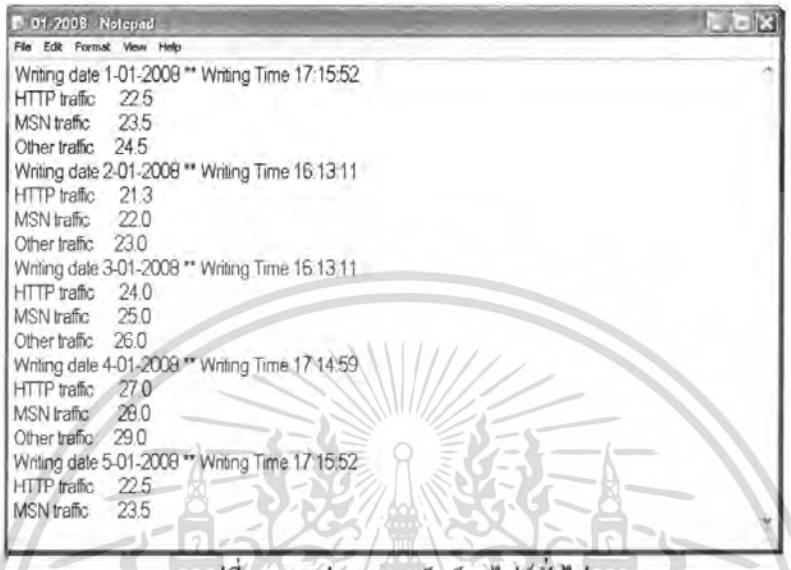
แต่ในส่วนของการจัดเก็บโปรโตคอลเอ็มเอสเอ็น (MSN) จะมีการแบ่งโฟลเดอร์ย่อยเข้าไปเป็นเดือนก่อนแล้วจึงมีการจัดเก็บไฟล์ ซึ่งมีสองแบบคือ ไฟล์สนทนา กับไฟล์ที่เก็บรายชื่อคู่สนทนา



รูปที่ ก.15 การแบ่งไดเร็กทอรี่ในการเก็บล็อกไฟล์ของโปรโตคอลเอ็มเอสเอ็น(MSN)

การจัดเก็บล็อกไฟล์จึงแบ่งออกเป็น 4 หมวด ดังนี้

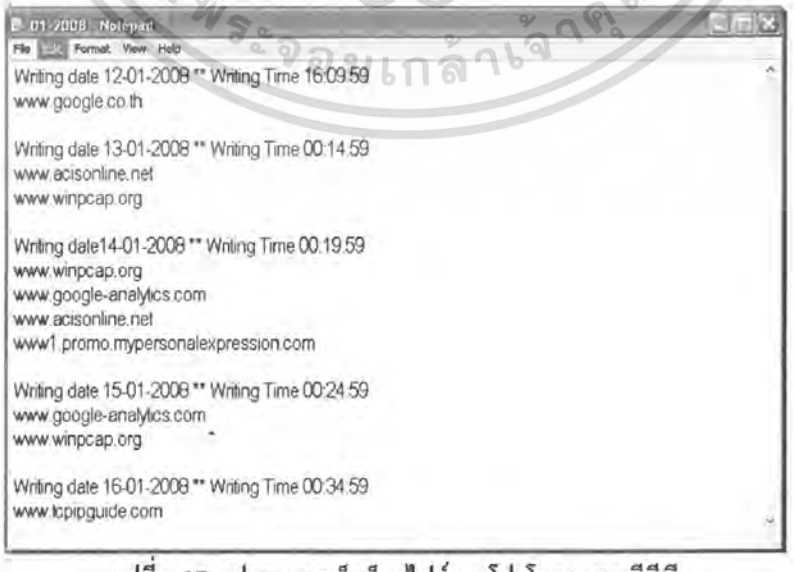
ก.2.1.1 ล็อกไฟล์ทั่วไป



รูปที่ ก.16 รูปแบบการเก็บล็อกไฟล์ทั่วไป

การจัดเก็บล็อกไฟล์ทั่วไป มีรูปแบบการจัดเก็บโดยแบ่งตามวันที่ที่ทำการจัดเก็บ แสดงรายละเอียดว่าไอพีแอดเดรสนั้นๆ มีการใช้งานโปรโตคอลใดบ้าง เทียบเป็นเปอร์เซ็นต์ได้มากน้อยเท่าใด

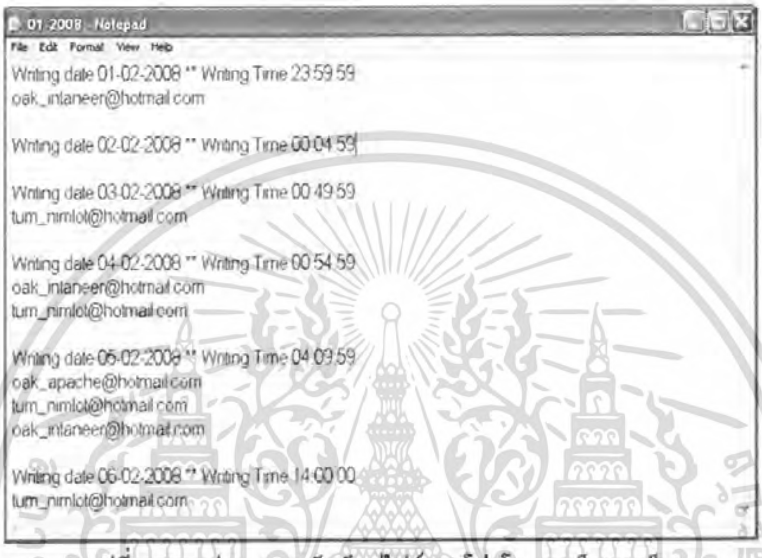
ก.2.1.2 ล็อกไฟล์ของโปรโตคอลเอชทีทีพี



รูปที่ ก.17 รูปแบบการเก็บล็อกไฟล์ของโปรโตคอลเอชทีทีพี

การจัดเก็บล็อกไฟล์ของโปรโตคอลเอชทีทีพี จะแบ่งตามวันที่ใช้งานเช่นกัน โดยจะแสดงรายละเอียดว่าไอพีแอดเดรสนั้นๆ มีการเข้าใช้งานในเว็บไซต์ใดบ้าง

ก.2.1.3 ล็อกไฟล์ของโปรโตคอลเอ็มเอสเอ็น



รูปที่ ก.18 รูปแบบการเก็บล็อกไฟล์ของโปรโตคอลเอ็มเอสเอ็น

การจัดเก็บล็อกไฟล์ของโปรโตคอล MSN จะแบ่งตามวันที่ใช้งานเช่นกัน โดยจะแสดงรายละเอียดว่าไอพีแอดเดรสนั้นๆ มีการใช้สนทนากับอีเมลล์แอดเดรสใดบ้าง

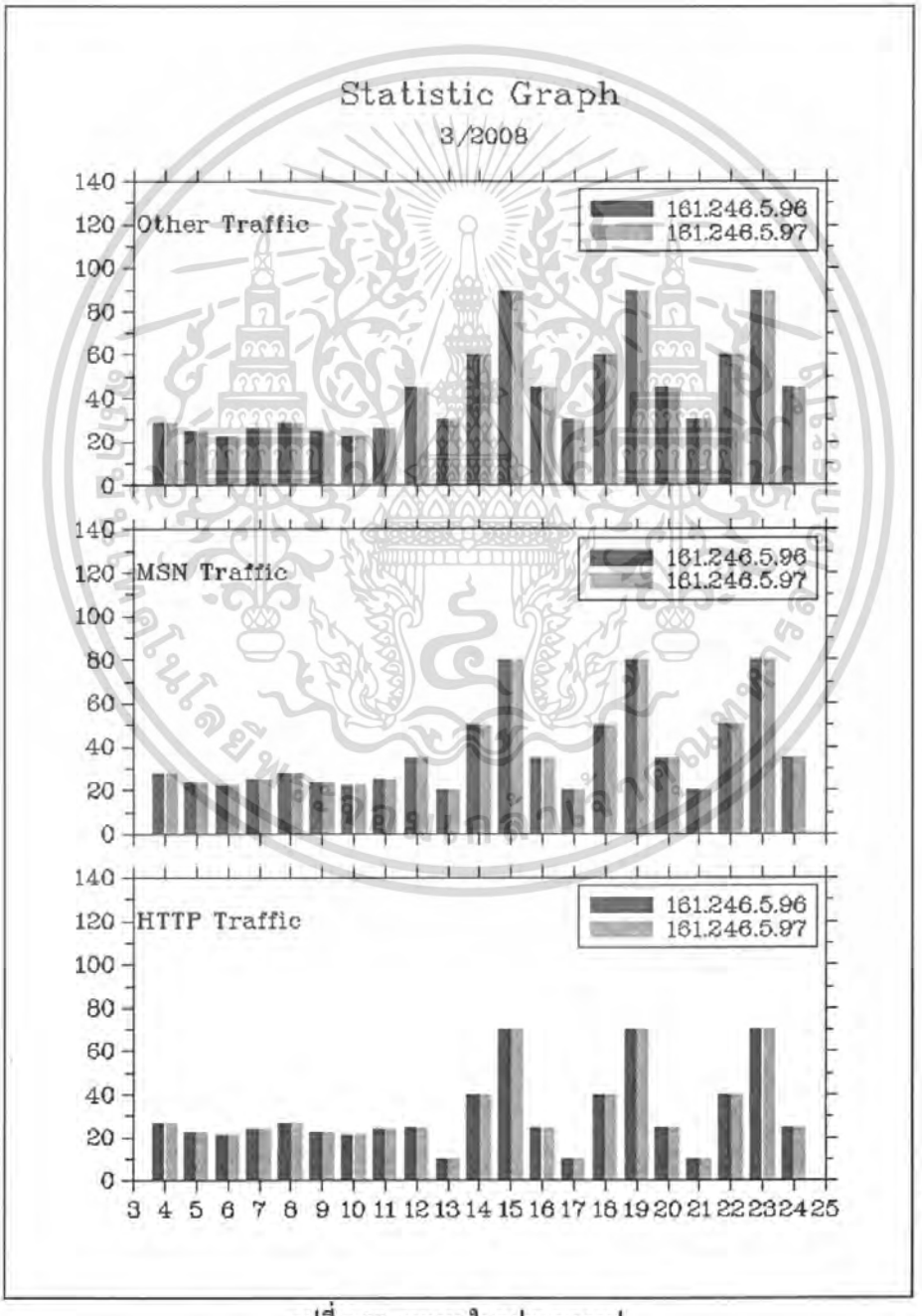
ก.2.1.4 ล็อกไฟล์ของการสนทนาผ่านโปรแกรมเอ็มเอสเอ็น

รูปที่ ก.19 รูปแบบการเก็บล็อกไฟล์ของการสนทนาเอ็มเอสเอ็น

ก.2.2 ไฟล์รายงาน

การจัดทำไฟล์รายงานนั้นจะมีอยู่ 2 ชนิดด้วยกัน คือ ไฟล์รายงานในรูปของกราฟเปรียบเทียบ และไฟล์รายงานในรูปของเอกสารเอ็กเซลล์ (Excel File) ซึ่งสามารถเรียกดูและทำความเข้าใจได้ง่ายกว่าการเปิดดูล็อกไฟล์โดยทั่วไป

ก.2.2.1 รายงานในรูปกราฟ



รูปที่ ก.20 รายงานในรูปแบบกราฟ

รายงานในรูปแบบกราฟจะทำการเปรียบเทียบการใช้งานในโปรโตคอลต่างๆ ของแต่ละไอพีแอดเดรส ในช่วงเวลาที่ผู้ใช้งานเลือกไว้ โดยโปรแกรมจะทำการสร้างกราฟทั้งสิ้น 3 กราฟ ได้แก่

1. กราฟแสดงเปอร์เซ็นต์การใช้งานโปรโตคอล HTTP
2. กราฟแสดงเปอร์เซ็นต์การใช้งานโปรโตคอล MSN
3. กราฟแสดงเปอร์เซ็นต์การใช้งานโปรโตคอลอื่นๆ

ในแกน X จะเป็นช่วงของวันที่ ที่ได้ทำการกำหนดมาในหน้าของการตั้งค่าการออกรายงาน (Report) ส่วนในแกน Y จะเป็นเปอร์เซ็นต์การใช้งาน หัวของเอกสารจะแสดงเดือนว่ากราฟที่ได้นำเสนอเป็นของเดือนและปีอะไร

ถ้ามีการแสดงผลมากกว่า 1 ไอพี แต่ละไอพีจะถูกแยกให้เห็นชัดเจนโดยใช้สีที่แตกต่างกันตามที่ระบุไว้ทางด้านขวาของแต่ละกราฟ

ก.2.2.2 รายงานในรูปเอกสารเอ็กเซลล์ (Excel File)

รายงานในรูปแบบเอกสารเอ็กเซลล์ โปรแกรมจะทำการรวบรวมข้อมูลต่างๆมาใส่ลงในเอกสารเอ็กเซลล์ โดยตัวอย่างรายงานแบบเอกสารเอ็กเซลล์ 3 แบบ ได้แก่

1). รายงานแสดงการใช้งานโปรโตคอล HTTP

35	161.246.5.96	ม.ค.-08	www.google.co.th	1
36	161.246.5.96	ม.ค.-08	www.prototypejs.org	1
37	161.246.5.96	ก.พ.-08	www.dek-d.com	1
38	161.246.5.96	ก.พ.-08	hits.truehits.in.th	2
39	161.246.5.96	ก.พ.-08	ivs.truehits.in.th	2
40	161.246.5.96	ก.พ.-08	image.dek-d.com	1
41	161.246.5.96	ก.พ.-08	www.mail.yahoo.com	2
42	161.246.5.96	ก.พ.-08	www.pantip.com	1
43	161.246.5.96	ก.พ.-08	www.google.co.th	4
44	161.246.5.96	ก.พ.-08	www.prototypejs.org	3
45	161.246.5.96	ก.พ.-08	getahead.org	1
46	161.246.5.96	ก.พ.-08	www.foundstone.com	1
47	161.246.5.96	ก.พ.-08	sdc.mcafee.com	1
48	161.246.5.96	ก.พ.-08	www.pandora.com	1
49	161.246.5.96	ก.พ.-08	www.xbox.com	1
50	161.246.5.96	ก.พ.-08	www.foxsports.com	1
51	161.246.5.96	ก.พ.-08	ccna-cisco-academy.blogspot	2
52	161.246.5.96	ก.พ.-08	www.blogger.com	2
53	161.246.5.96	ก.พ.-08	www.wieistmeineip.de	2
54	161.246.5.96	ก.พ.-08	www.pttict.com	2
55	161.246.5.96	ก.พ.-08	kanchanapisek.or.th	1
56	161.246.5.96	ก.พ.-08	www.msgpluslive-update.net	1
57	161.246.5.96	ก.พ.-08	dnl-us4.kaspersky-labs.com	1
58	161.246.5.96	ก.พ.-08	www.text-link-ads.com	1
59	161.246.5.96	ก.พ.-08	www.mypagerank.net	1
60	161.246.5.97	ม.ค.-08	www.google.co.th	1
61	161.246.5.97	ม.ค.-08	www.acisonline.net	2
62	161.246.5.97	ม.ค.-08	www.winpcap.org	6
63	161.246.5.97	ม.ค.-08	www.google-analytics.com	6
64	161.246.5.97	ม.ค.-08	www1.promo.mypersonalexpr	2
65	161.246.5.97	ม.ค.-08	www.tcpipguide.com	2

รูปที่ ก.21 รายงานในรูปแบบเอกสารเอ็กเซลล์แสดงการใช้งานโปรโตคอลเอชทีทีพี

ไอพี	เดือน/ปี	รายชื่อเว็บไซต์	ความถี่ที่เข้าใช้งานต่อเดือน

รูปที่ ก.22 รายละเอียดของแต่ละฟิลล์ในไฟล์รายงานโปรโตคอลเอชทีทีพี

รายงานในรูปแบบเอกสารเอ็กเซลล์ จะแสดงรายละเอียดว่า ในแต่ละเดือนแต่ละไอพีแอดเดรส ว่ามีการเข้าใช้งานเว็บไซต์ใดบ้าง จำนวนกี่ครั้ง โดยจะแสดงผลเรียงลำดับตามไอพีแอดเดรส

2). รายงานแสดงการใช้งานโปรโตคอล MSN

1	161.246.5.96	ม.ค.-08	oak_intaneen@hotmail.com	5
2	161.246.5.96	ม.ค.-08	tum_nimlot@hotmail.com	6
3	161.246.5.96	ม.ค.-08	immi_87@hotmail.com	7
4	161.246.5.96	ก.พ.-08	tum_nimlot@hotmail.com	4
5	161.246.5.96	ก.พ.-08	immi_87@hotmail.com	7
6	161.246.5.96	ก.พ.-08	oak_apache@hotmail.com	4
7	161.246.5.96	ก.พ.-08	tum_nimlot@hotmail.com	4
8	161.246.5.96	ก.พ.-08	immi_87@hotmail.com	7
9	161.246.5.96	ก.พ.-08	oak_apache@hotmail.com	4
10	161.246.5.97	ม.ค.-08	oak_intaneen@hotmail.com	5
11	161.246.5.97	ม.ค.-08	tum_nimlot@hotmail.com	6
12	161.246.5.97	ม.ค.-08	oak_apache@hotmail.com	4
13	161.246.5.97	ม.ค.-08	worrachaii@hotmail.com	1
14	161.246.5.97	ม.ค.-08	atom_harrypotter@hotmail.com	2
15	161.246.5.97	ม.ค.-08	juneveryhot@hotmail.com	1
16	161.246.5.97	ก.พ.-08	oak_intaneen@hotmail.com	5
17	161.246.5.97	ก.พ.-08	tum_nimlot@hotmail.com	6
18	161.246.5.97	ก.พ.-08	oak_apache@hotmail.com	4
19	161.246.5.97	ก.พ.-08	worrachaii@hotmail.com	1
20	161.246.5.97	ก.พ.-08	atom_harrypotter@hotmail.com	2
21	161.246.5.97	ก.พ.-08	juneveryhot@hotmail.com	1

รูปที่ ก.23 รายงานในรูปแบบเอกสารเอ็กเซลล์แสดงการใช้งานโปรโตคอลเอ็มเอสเอ็น

ไอพี	เดือน/ปี	รายชื่ออีเมลล์ที่มีการสนทนา	ความถี่ที่ใช้งานต่อเดือน

รูปที่ ก.24 รายละเอียดของแต่ละฟิลล์ในไฟล์รายงานโปรโตคอลเอ็มเอสเอ็น

รายงานในรูปแบบเอกสารเอ็กเซลล์ จะแสดงรายละเอียดว่า ในแต่ละเดือนแต่ละไอพีแอดเดรส ว่ามีการเข้าใช้งานโปรโตคอล MSN กับอีเมลล์ใดบ้าง จำนวนกี่ครั้ง โดยจะแสดงผลเรียงลำดับตามไอพีแอดเดรส

3). รายงานแสดงการใช้งานโดยรวม

66	161.246.5.96	29/3/2008	48.4	32.9	18.7
67	161.246.5.96	30/3/2008	36.27	19.82	43.91
68	161.246.5.96	31/3/2008	52.21	28.87	18.92
69	161.246.5.96	1/4/2008	62.12	31.47	6.41
70	161.246.5.96	2/4/2008	78	5.28	16.72
71	161.246.5.96	3/4/2008	42.37	28.95	28.68
72	161.246.5.96	4/4/2008	58.75	23.21	18.04
73	161.246.5.96	5/4/2008	12.54	35.25	52.21
74	161.246.5.97	25/1/2008	10.39	29.85	59.76
75	161.246.5.97	26/1/2008	55.68	37	7.32
76	161.246.5.97	27/1/2008	19.92	42.28	37.8
77	161.246.5.97	28/1/2008	22.54	37.95	39.51
78	161.246.5.97	29/1/2008	54.21	45.2	0.59
79	161.246.5.97	30/1/2008	34.82	42.39	22.79
80	161.246.5.97	31/1/2008	45.28	34.41	20.31
81	161.246.5.97	1/2/2008	36.58	47.29	16.13

รูปที่ ก.25 รายงานในรูปแบบเอกสารเอ็กเซลล์แสดงการใช้งานโดยรวม

ไอพี	วัน/เดือน/ปี	เปอร์เซ็นต์การใช้งาน โปรโตคอล HTTP	เปอร์เซ็นต์การใช้งาน โปรโตคอล MSN	เปอร์เซ็นต์การใช้งาน โปรโตคอลอื่นๆ

รูปที่ ก.26 รายละเอียดของแต่ละฟิลล์ในไฟล์รายงานโปรโตคอลอื่นๆ

รายงานในรูปแบบเอกสารเอ็กเซลล์ จะแสดงรายละเอียดว่า ในแต่ละเดือนแต่ละไอพีแอดเดรส ว่ามีการเข้าใช้งานโปรโตคอลใดบ้าง แบ่งเป็น โปรโตคอล HTTP โปรโตคอล MSN และโปรโตคอลอื่นๆ จำนวนเท่าใด โดยจะแสดงผลเรียงลำดับตามไอพีแอดเดรส

ภาคผนวก ข.

การทำงานร่วมกับไลบราลี **Dislin**

ข.1. การเก็บค่าที่ต้องการแสดงผล

ในการแสดงผลเป็นรูปแบบกราฟแท่งโดยใช้ไลบราลี Dislin นั้น เราจะต้องทำการเก็บค่าไว้ในตัวแปรที่เป็นอาเรย์สามมิติเช่น

float a[x][y][z];

คือ ค่าที่เก็บเป็น float

ค่า x เป็นจำนวนของผู้ใช้งาน หรือรายการที่ต้องการแสดง

ค่า y เป็นประเภทที่ต้องการแสดง

ค่า z เป็นวันที่ของค่านั้นๆ

ตัวอย่างเช่น

ถ้าเรามีล็อกไฟล์ คือ

IP a.a.a.a

Writing date 1-02-2008 ** Writing Time 17:15:52

HTTP traffic 22.5

MSN traffic 23.5

Other traffic 24.5

Writing date 2-02-2008 ** Writing Time 16:13:11

HTTP traffic 15.3

MSN traffic 18.0

Other traffic 26.0

รูปที่ ข.1 ตัวอย่างล็อกไฟล์ของไอพีแอดเดรส a.a.a.a

IP b.b.b.b

Writing date 1-02-2008 ** Writing Time 17:15:52

HTTP traffic 28.5

MSN traffic 13.5

Other traffic 14.5

Writing date 2-02-2008 ** Writing Time 16:13:11

HTTP traffic 21.3

MSN traffic 22.0

Other traffic 23.0

รูปที่ ข.2 ตัวอย่างล็อกไฟล์ของไอพีแอดเดรส b.b.b.b

จะมีตัวอย่างดังต่อไปนี้

a[0][0][0] = 22.5

หมายถึง อาเรย์ของไอพีแรก (IP a.a.a.a) ประเภทที่จะแสดงประเภทแรก (HTTP) วันที่ 1

a[0][1][0] = 23.5

หมายถึง อาเรย์ของไอพีแรก (IP a.a.a.a) ประเภทที่จะแสดงประเภทที่สอง(MSN) วันที่ 1

a[0][2][0] = 24.5

หมายถึง อาเรย์ของไอพีแรก (IP a.a.a.a) ประเภทที่จะแสดงประเภทที่สาม(Other) วันที่ 1

a[0][0][1] = 15.3

หมายถึง อาเรย์ของไอพีแรก (IP a.a.a.a) ประเภทที่จะแสดงประเภทแรก (HTTP) วันที่ 2

a[0][1][1] = 18.0

หมายถึง อาเรย์ของไอพีแรก (IP a.a.a.a) ประเภทที่จะแสดงประเภทที่สอง(MSN) วันที่ 2

a[0][2][1] = 26.0

หมายถึง อาเรย์ของไอพีแรก (IP a.a.a.a) ประเภทที่จะแสดงประเภทที่สาม(Other) วันที่ 2

a[1][0][0] = 28.5

หมายถึง อาเรย์ของไอพีที่สอง (IP b.b.b.b) ประเภทที่จะแสดงประเภทแรก (HTTP) วันที่ 1

a[1][1][0] = 13.5

หมายถึง อาเรย์ของไอพีที่สอง (IP b.b.b.b) ประเภทที่จะแสดงประเภทที่สอง(MSN) วันที่ 1

a[1][2][0] = 14.5

หมายถึง อาเรย์ของไอพีที่สอง(IP b.b.b.b) ประเภทที่จะแสดงประเภทที่สาม (Other) วันที่ 1

a[1][0][1] = 21.3

หมายถึง อาเรย์ของไอพีที่สอง (IP b.b.b.b) ประเภทที่จะแสดงประเภทแรก (HTTP) วันที่ 2

a[1][1][1] = 22.0

หมายถึง อาเรย์ของไอพีที่สอง (IP b.b.b.b) ประเภทที่จะแสดงประเภทที่สอง(MSN) วันที่ 2

a[1][2][1] = 23.0

หมายถึง อาเรย์ของไอพีที่สอง(IP b.b.b.b) ประเภทที่จะแสดงประเภทที่สาม (Other) วันที่ 2

การเรียกใช้ให้แสดงกราฟจะใช้ฟังก์ชัน

bars (float *xray, float *y1ray, float *y2ray, int n);

float *xray คือตำแหน่งเริ่มต้นในแกน x เช่น

x = {0,1,2,3}

float *y2ray คือตำแหน่งเริ่มต้นในแกน y ณ ตำแหน่ง x

y = {0,0,0,0}

float *y1ray คือตำแหน่งความสูงในแกน y ณ ตำแหน่ง x

เวลาเรียกใช้ตัวแปรของเรา เราจะเรียกแค่ 2 มิติ คือ a[x][y] โดยเมื่อเราเรียกฟังก์ชันนี้ มันจะทำการสร้างกราฟให้โดยไล่ตามค่าของฟิลด์ให้ หรือจะไล่วันทั้งหมดให้เองอัตโนมัติตามจำนวนในค่า n เช่นถ้าเราต้องการสร้างกราฟของไอพี a.a.a.a โดยให้แสดงข้อมูลประเภท HTTP Traffic เราจะเรียกใช้งานว่า a[0][0]

int n จำนวนของแท่งกราฟที่ต้องการแสดง

ในส่วนของฟังก์ชันอื่นๆที่มีการใช้งานก็ใช้ตามคู่มือการใช้งานปกติ