

การพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์
A Development of Electronic Mail Capturing System



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2546
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์
นักศึกษา	นายกาน มะลิหอม
อาจารย์ที่ปรึกษา	ผศ. อัครินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

ปัจจุบัน มีการใช้งานอินเทอร์เน็ตอย่างแพร่หลาย ในด้านต่างๆ รวมทั้งการรับส่งจดหมายอิเล็กทรอนิกส์ เพราะสามารถทำได้สะดวก และมีการพัฒนาความสามารถมากขึ้นเรื่อย ปัญหาของการใช้งานจดหมายอิเล็กทรอนิกส์ที่พบได้บ่อยก็คือ มักจะมีจดหมายที่ไม่พึงประสงค์ เช่น ทำการส่งจดหมายที่มีเนื้อหา โฆษณาสินค้า หรือ ส่งจดหมายขยะมาพร้อมๆ กันเป็นจำนวนมาก ซึ่งอาจก่อให้เกิดความรำคาญ หรือ ก่อให้เกิดความเสียหายได้ กับทั้งตัวผู้ใช้ และระบบของผู้ให้บริการ จึงมีการสร้างระบบที่ทำหน้าที่รวบรวม ตัวอย่างของจดหมายอิเล็กทรอนิกส์เพื่อนำตัวอย่างดังกล่าวไปใช้ในกระบวนการกลั่นกรองแยกแยะจดหมายไม่พึงประสงค์ต่อไป โครงการพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ ที่จัดทำขึ้นนี้ พัฒนาโดยใช้ภาษาซีและมีการเรียกใช้งานไลบรารี PCAP ทำการเลือกดักจับข้อมูลบนเครือข่าย จากนั้นจึงทำการคัดกรองเลือกเฉพาะแพ็คเกจที่เกี่ยวข้องกับการรับส่งจดหมายอิเล็กทรอนิกส์แล้วจึงบันทึกลงฐานข้อมูล จากนั้นจึงอ่านข้อมูลจากฐานข้อมูลเพื่อนำข้อมูลที่ได้มาทำการประกอบและแปลงกลับเป็นรูปของจดหมายดั้งเดิม จากผลทดสอบของการทำงานของโปรแกรมสามารถทำการดักจับข้อมูลที่วิ่งบนเครือข่าย และคัดกรองเฉพาะข้อมูลที่เกี่ยวข้องกับจดหมายอิเล็กทรอนิกส์(SMTP) และทำการประกอบข้อมูลที่เกี่ยวข้องเข้าด้วยกัน เพื่อให้ได้จดหมายอิเล็กทรอนิกส์ที่คล้ายคลึงกับต้นฉบับ โดยผลของโครงการนี้ จะเป็นตัวอย่างของจดหมายที่ทำการส่งในเครือข่ายจริงเพื่อใช้ในการวิเคราะห์รูปแบบลักษณะของจดหมายอิเล็กทรอนิกส์ได้ต่อไป

Title	A Development of Electronic Mail Capturing System
Student	Mr. Karn Malihom
Advisor	Asst.Prof Akharin Khunkitti
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2003

Abstract

Today, Telecommunication trend was changed into internet. E-mail is the one of the most popular way to communicate through internet because it's easy and fast. There are some problem for using e-mail, such as junk mails, advertising mails and many security issues. For analysing them, the e-mail analysis system will just collect all of them for processing. A Development of Electronic Mail Capturing System was created for capturing all e-mail traffic. It's implemented using C language with PCAP libraly to capture data from network then filter only e-mail packets into database. An E-mail was created by integrating selected packet from the database. The experimental system can capture SMTP e-mails from network, in batch mode. The result will be used for e-mail analysis purposes.

กิตติกรรมประกาศ

โครงการพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ นี้สำเร็จลงได้ เนื่องจากความกรุณาของ ผศ. อัครินทร์ คุณกิตติ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการ ที่ช่วยให้คำแนะนำแนวคิดในการทำโครงการ ให้คำปรึกษาในด้านเนื้อหาที่เป็นประโยชน์ รวมทั้งให้ความช่วยเหลือในการตรวจสอบแก้ไขเอกสาร เรียบเรียงเอกสาร

ขอขอบคุณพนักงานเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศทุกท่าน ที่ช่วยอำนวยความสะดวกในด้านต่างๆ ทั้งสถานที่ อุปกรณ์ในการทำโครงการ ให้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณความคิดดีๆ และกำลังใจ จากครอบครัวและเพื่อนๆ นักศึกษาทุกท่าน ที่ช่วยกระตุ้น และผลักดันให้ทำโครงการนี้จนสำเร็จ

ผู้เขียนจึงขอขอบคุณทุกท่านมา ณ โอกาสนี้

กาน มะลิหอม
กุมภาพันธ์ 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ **iii** ษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII
บทที่	
1. บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการพัฒนาโครงการ	2
1.3 ขอบเขตของการพัฒนาโครงการ	2
1.4 ขั้นตอนในการพัฒนาระบบงาน	2
1.5 รายละเอียดของแต่ละบท	3
2. การดักจับจดหมายอิเล็กทรอนิกส์	4
2.1 จดหมายอิเล็กทรอนิกส์	4
2.1.1 องค์ประกอบของจดหมายอิเล็กทรอนิกส์	5
2.1.2 ขั้นตอนการทำงานของระบบการรับส่งจดหมายอิเล็กทรอนิกส์	7
2.1.3 Simple Mail Transfer Protocol (SMTP)	9
2.2 การดักจับข้อมูลบนระบบเครือข่าย	11
2.2.1 องค์ประกอบของโปรแกรมประเภทดักจับข้อมูล	12
2.3 รูปแบบวิธีการดักจับข้อมูลบนเครือข่าย	13
2.4 โพรโทคอล TCP/IP	15
2.4.1 Ethernet	17
2.4.2 IP (Internet Protocol)	17
2.4.3 TCP (Transmission Control Protocol)	20

สารบัญ (ต่อ)

	หน้า
2.5 กระบวนการส่งข้อมูลผ่านโปรโตคอล TCP/IP.....	23
2.5.1 การสร้างการเชื่อมต่อ.....	23
2.5.2 การถ่ายโอนข้อมูล.....	24
2.5.3 การจบการเชื่อมต่อ.....	24
3. การวิเคราะห์และออกแบบระบบ.....	26
3.1 การศึกษาความต้องการของระบบ.....	26
3.2 การออกแบบโครงสร้างการทำงาน.....	26
3.2.1 ขั้นตอนการทำงานโดยรวมของระบบทั้งหมด.....	27
3.2.2 ขั้นตอนการทำงานของส่วนดักจับข้อมูล.....	28
3.2.3 ขั้นตอนการทำงานของส่วนแปลข้อมูล.....	30
3.2.4 การออกแบบในส่วนของการจัดเก็บข้อมูล.....	31
3.3 การออกแบบในส่วนของดักจับข้อมูล.....	32
3.3.1 โครงสร้างในการเก็บข้อมูลของ PCAP Library.....	32
3.3.2 โครงสร้างในการเก็บข้อมูลของ Ethernet.....	33
3.3.3 โครงสร้างในการเก็บข้อมูลของ IP.....	33
3.3.4 โครงสร้างในการเก็บข้อมูลของ TCP.....	34
4. การพัฒนาระบบงานและการทดลอง.....	36
4.1 เครื่องมือที่ใช้ในการพัฒนาระบบงาน.....	36
4.2 ขั้นตอนการพัฒนาโปรแกรมส่วนการดักจับจดหมายอิเล็กทรอนิกส์.....	36
4.3 การขั้นตอนการพัฒนาโปรแกรมส่วนการดักจับจดหมายอิเล็กทรอนิกส์.....	37
4.3.1 การพัฒนาในส่วนของการดักจับและคัดกรองแพ็คเก็ต.....	37
4.3.2 การพัฒนาในส่วนของการเรียงเรียงและแปลข้อมูล.....	39
4.4 การทดลองระบบดักจับจดหมายอิเล็กทรอนิกส์.....	40

สารบัญ (ต่อ)

	หน้า
5. บทสรุปและข้อเสนอแนะ.....	43
5.1 สรุปผลการทำงานและปะโยชน์ที่ได้รับ.....	43
5.2 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ.....	43
บรรณานุกรม.....	45
ภาคผนวก.....	46
ประวัติผู้เขียน.....	51



สารบัญตาราง

ตารางที่	หน้า
2.1	แสดงถึงตัวอย่างคำสั่งของ SMTP ที่ใช้ในการติดต่อของฝั่งผู้ส่ง.....
2.2	แสดงถึงตัวอย่างของ result code บางส่วน
2.2	แสดงถึงตัวอย่างของ result code บางส่วน (ต่อ).....
3.1	แสดง Table ของข้อมูลที่เกี่ยวข้องในระบบงาน.....
3.1	แสดง Table ของข้อมูลที่เกี่ยวข้องในระบบงาน (ต่อ).....
4.1	แสดงรายละเอียดเหตุการณ์ที่เกิดขึ้น.....



สารบัญภาพ

ภาพที่	หน้า
2.1 สถาปัตยกรรมระบบแม่ข่ายใน TCP/IP	4
2.2 แสดงถึงตัวอย่างส่วนประกอบของจดหมาย.....	5
2.3 ส่วนประกอบของโปรแกรมดักจับข้อมูลทั่วไป.....	13
2.4 ภาพจำลองการทำงานของโปรแกรมดักจับข้อมูล.....	14
2.5 ภาพลำดับชั้นของชุดโปรโตคอล TCP/IP.....	16
2.6 ภาพการห่อหุ้มข้อมูลตามลำดับชั้นของโปรโตคอล.....	16
2.7 แสดงถึงโครงสร้างของ IP Packet Header.....	18
2.8 แสดงถึงโครงสร้างของ TCP Packet Header.....	21
2.9 ลักษณะการ 3-Ways handshakes ของ TCP.....	23
2.10 ขั้นตอนการถ่ายโอนข้อมูล.....	24
2.11 ขั้นตอนการยุติการเชื่อมต่อ.....	24
3.1 Context Diagram ของระบบดักจับจดหมายอิเล็กทรอนิกส์.....	27
3.2 DFD 0 ของระบบดักจับจดหมายอิเล็กทรอนิกส์.....	27
3.3 แสดงขั้นตอนการทำงานโดยรวมในส่วนของการดักจับจดหมายอิเล็กทรอนิกส์.....	28
3.4 แสดงขั้นตอนการคัดกรองแพ็คเก็ตที่ต้องการ.....	29
3.5 แสดงขั้นตอนการทำงานโดยรวมในส่วนของการแปลข้อมูล.....	30
3.6 แสดงขั้นตอนการแปลงข้อมูล.....	31
3.7 แสดงถึงโครงสร้างในการเก็บข้อมูลของ PCAP Library.....	32
3.8 แสดงถึงโครงสร้างในการเก็บข้อมูลของ Ethernet.....	33
3.9 แสดงถึงโครงสร้างในการเก็บข้อมูลของ IP.....	33
3.10 แสดงถึงโครงสร้างในการเก็บข้อมูลของ TCP.....	35
4.1 แสดงถึงการเชื่อมต่อเครือข่ายระหว่างการพัฒนาและที่ใช้ในการทดลอง.....	36
4.2 แสดงถึงชุดคำสั่งที่ใช้ในการตรวจสอบคุณสมบัติของการ์ดแลน.....	37
4.3 แสดงถึงชุดคำสั่งที่ใช้ในการคัดกรองแพ็คเก็ตต่างๆ.....	38
4.4 แสดงตัวอย่างโปรแกรมระหว่างการดักจับโดยไม่แสดง Payload.....	40
4.5 แสดงตัวอย่างข้อมูลชุดเดียวกับเทียบกับโปรแกรม Ethereal.....	41

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.6	แสดงตัวอย่างข้อมูลที่ได้จากการแปลง.....42
ก-1	แสดงถึงการเรียกใช้งาน ส่วนคัดจับจดหมายอิเล็กทรอนิกส์.....48
ก-2	แสดงตัวอย่างโปรแกรมระหว่างการดักจับ.....49
ก-3	แสดงถึงการเรียกใช้งาน แปลและเรียบเรียงข้อมูล.....49
ก-4	แสดงถึงไฟล์จดหมายอิเล็กทรอนิกส์ที่ทำการจัดเก็บ.....50
ก-5	แสดงตัวอย่างจดหมายอิเล็กทรอนิกส์ด้วย Text Editor.....50



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน อินเทอร์เน็ตเป็นช่องทางที่ใช้ในการติดต่อสื่อสาร ที่ได้รับความนิยมอย่างแพร่หลาย และมีความสำคัญเพิ่มมากขึ้นเป็นลำดับ ซึ่งอินเทอร์เน็ตนั้น ช่วยให้ผู้คนสามารถแลกเปลี่ยน ข้อมูลข่าวสาร ระหว่างกัน ได้อย่างรวดเร็ว ทั้งที่ระยะทางนั้นอยู่ห่างไกลออกไป อินเทอร์เน็ต เปรียบเสมือนเป็นคลังข้อมูลขนาดใหญ่ ที่มีข้อมูลสารสนเทศต่าง ๆ มากมายให้สามารถเลือกนำไปใช้อ้างอิง ศึกษาหาความรู้ ได้ไม่รู้จักจบ และยังมีเนื้อหาต่าง ๆ มากขึ้นเรื่อย ๆ

จดหมายอิเล็กทรอนิกส์ เป็นหนึ่งในบริการ ที่เกิดขึ้นมาพร้อมกับอินเทอร์เน็ต เพื่อใช้ในการส่งจดหมาย ไปหาผู้รับ ได้อย่างรวดเร็วผ่านทางเครือข่ายอินเทอร์เน็ต ซึ่งจดหมายอิเล็กทรอนิกส์ นั้น มีข้อได้เปรียบ เหนือกว่า จดหมายธรรมดาทั่วไปมากมาย ไม่ว่าจะเป็นความรวดเร็วในการส่งถึงผู้รับโดยไม่จำกัดระยะทาง ความสะดวกในการส่งหาผู้รับหลายคนในเวลาพร้อมกัน และ มีค่าใช้จ่ายในการส่งที่ถูกกว่ามาก และจดหมายอิเล็กทรอนิกส์ยังได้มีการพัฒนาเพิ่มเติมความสามารถ ความน่าสนใจมากกว่าแค่เพียงจดหมายธรรมดา ไม่ว่าจะเป็นการส่ง จดหมายที่ประกอบไปด้วยข้อมูล ภาพและเสียง ด้วยข้อได้เปรียบอันมากมาย การใช้งานจดหมายอิเล็กทรอนิกส์ จึงเติบโตขึ้นอย่างรวดเร็ว และมีผู้ใช้งานเป็นจำนวนที่สูงมาก และด้วยเหตุที่มีผู้ใช้งานที่สูงมาก จึงเป็นเสมือนเป้าหมายในการโจมตีก่อวินาศกรรม เช่น การส่งจดหมายที่มี ไวรัสปะปนมา ทำการส่งจดหมายที่มีเนื้อหา โฆษณาสินค้า หรือ ส่งจดหมายขยะมาพร้อมกันเป็นจำนวนมาก ซึ่งอาจก่อให้เกิดความรำคาญ หรือ ก่อให้เกิดความเสียหายได้ กับทั้งตัวผู้ใช้ และระบบของผู้ให้บริการ จึงต้องมีการสร้างกระบวนการในการกลั่นกรอง แยกแยะ จดหมายอิเล็กทรอนิกส์ที่ไม่พึงประสงค์ดังกล่าว ออกมา

จึงได้มีการจัดทำ โครงการ การพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ เพื่อใช้ในการสนับสนุนงาน ที่เกี่ยวกับการค้นคว้าวิจัยลักษณะของจดหมายอิเล็กทรอนิกส์ที่ไม่พึงประสงค์ โดยโครงการดังกล่าว จะเป็นการจัดทำโปรแกรมดักจับข้อมูลบนเครือข่าย โดยที่ทำการเลือกดักจับเฉพาะข้อมูลที่เป็นจดหมายอิเล็กทรอนิกส์ และทำการ เก็บรวบรวมจดหมายดังกล่าวให้เป็นระเบียบให้สามารถนำไปใช้งาน ต่อไปได้

1.2 วัตถุประสงค์ของโครงการพัฒนา

การพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ ผู้ศึกษาได้กำหนดวัตถุประสงค์ของโครงการไว้ดังนี้

- 1.2.1 เพื่อทำการศึกษารูปแบบการทำงาน ของระบบการรับส่งจดหมายอิเล็กทรอนิกส์ รวมทั้งขั้นตอนการรับส่งข้อมูลผ่านทางระบบเครือข่ายคอมพิวเตอร์
- 1.2.2 เพื่อทำการพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์
- 1.2.3 เพื่อศึกษาและทำความเข้าใจ เกี่ยวกับโครงสร้างข้อมูล ของโปรโตคอลต่างๆ ที่ใช้งานในระบบเครือข่ายคอมพิวเตอร์

1.3 ขอบเขตของการพัฒนาโครงการ

ระบบดักจับจดหมายอิเล็กทรอนิกส์ เป็นแอปพลิเคชันที่พัฒนาขึ้นเพื่อรวบรวมจดหมายอิเล็กทรอนิกส์ที่ทำการรับส่งบนเครือข่ายคอมพิวเตอร์ ซึ่งจะครอบคลุมการทำงานส่วนต่างๆ ดังนี้

- 1.3.1 สามารถทำการดักจับข้อมูลจดหมายอิเล็กทรอนิกส์ บนระบบเครือข่ายคอมพิวเตอร์ที่เป็นอีเทอร์เน็ตได้
- 1.3.2 ทำการจัดเก็บ ข้อมูลที่ดักจับได้เก็บบนระบบฐานข้อมูล เพื่อให้สามารถนำเข้าสู่กระบวนการแปลข้อมูลได้ง่าย
- 1.3.3 สามารถทำการแปลข้อมูลที่ทำการดักจับได้ เพื่อทำการแปลงกลับมาให้อยู่ในรูปแบบของจดหมายอิเล็กทรอนิกส์

1.4 ขั้นตอนในการพัฒนาระบบงาน

- 1.4.1 ศึกษาการรูปแบบและขั้นตอนการทำงานของระบบการรับส่งจดหมายอิเล็กทรอนิกส์ รวมทั้งขั้นตอนในการรับส่งข้อมูลบนเครือข่ายอีเทอร์เน็ต
- 1.4.2 ศึกษารายละเอียดในเรื่องของการพัฒนาโปรแกรมบนเครือข่ายคอมพิวเตอร์ และจัดเก็บข้อมูลบนฐานข้อมูล
- 1.4.3 วิเคราะห์และออกแบบระบบดักจับจดหมายอิเล็กทรอนิกส์
- 1.4.4 พัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์
- 1.4.5 ทดสอบการใช้งาน และปรับปรุงแก้ไขระบบที่พัฒนาแล้ว
- 1.4.6 สรุปผลการทดสอบจากการใช้งานที่เกิดขึ้น
- 1.4.7 จัดทำเอกสารประกอบโครงการ

1.5 รายละเอียดของแต่ละบท

- 1.5.1 บทที่ 2 : นำเสนอความรู้เกี่ยวกับกระบวนการทำงานของจดหมายอิเล็กทรอนิกส์ ส่วนประกอบต่างๆของจดหมาย โปรโตคอลที่เกี่ยวข้องกับการพัฒนา เครื่องมือหรือซอฟต์แวร์ที่ใช้ในการพัฒนาระบบงาน
- 1.5.2 บทที่ 3 นำเสนอเกี่ยวกับการออกแบบและพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ โดยวิเคราะห์ความต้องการของระบบเพื่อนำไปสู่กระบวนการพัฒนาระบบต่อไป
- 1.5.3 บทที่ 4 : นำเสนอวิธีการพัฒนาระบบและผลการทดลองใช้ระบบดักจับจดหมายอิเล็กทรอนิกส์เพื่อตรวจสอบว่าระบบที่พัฒนาขึ้นสามารถทำงานได้ตรงตามความต้องการหรือไม่
- 1.5.4 บทที่ 5 : สรุปผลการพัฒนาระบบงาน และข้อเสนอแนะ



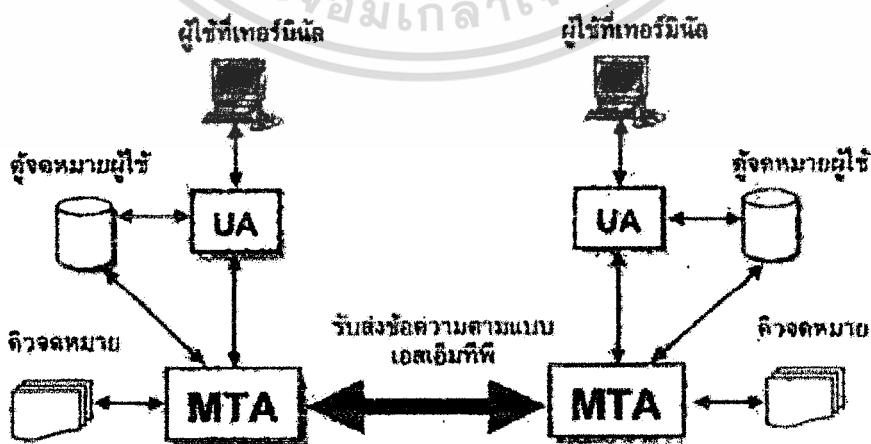
บทที่ 2

การดักจับจดหมายอิเล็กทรอนิกส์

2.1 จดหมายอิเล็กทรอนิกส์

จดหมายอิเล็กทรอนิกส์ ซึ่งนิยม เรียกเพียงสั้นๆว่า อีเมล (e-mail) เป็นโปรแกรมประยุกต์ที่แพร่หลายในระบบเครือข่าย ประโยชน์ของ อีเมลช่วยให้ผู้ใช้ ส่งและรับข้อความข้ามเครือข่ายได้ TCP/IP มีโปรโตคอลสนับสนุนการรับส่งเมลหลายโปรโตคอล แต่โปรโตคอลที่นิยมใช้ในอินเทอร์เน็ตคือ เอสเอ็มทีพี (SMTP : Simple Mail Transfer Protocol) หน้าที่ของ SMTP ก็คือการกำหนดกรรมวิธีและแบบแผนการนำส่งข้อความระหว่างผู้รับและผู้ส่ง SMTP อาศัย TCP เพื่อลำเลียงจดหมายผ่านทางพอร์ต 25

ระบบเมลที่ใช้ใน TCP/IP มีองค์ประกอบสองส่วนคือ ยูสเซอร์เอเจนต์ (UA : User Agent) และ เอ็มทีเอ (MTA : Mail Transfer Agent) เป็นชื่อที่นำมาจากระบบ X.400 ซึ่งเป็นมาตรฐานนานาชาติกำหนดการนำส่งเมล UA เป็น โปรแกรมที่ทำการติดต่อกับผู้ใช้และอำนวยความสะดวกให้ผู้เขียน แก้ไข และส่งจดหมาย รวมทั้งการเปิดอ่านจดหมายที่ได้รับ และจัดเก็บจดหมายเพื่อนำมาใช้ภายหลัง ส่วน MTA ทำหน้าที่หาเส้นทางและส่งจดหมายไปถึงปลายทาง การติดต่อระหว่าง MTA ใช้ TCP พอร์ต 25 โดยการแลกเปลี่ยนคำสั่งตามรูปแบบที่กำหนดใน RFC 822 รูปที่ 2.1 แสดงถึง สถาปัตยกรรมทั่วไปของระบบเมล



รูปที่ 2.1 สถาปัตยกรรมระบบเมลใน TCP/IP

การจัดแบ่งออกเป็น UA และ MTA มีข้อดีคือ แยกงานของทั้งสองส่วนให้เป็นอิสระจากกัน หน้าที่ของ UA เน้นการทำงานกับผู้ใช้เพื่อให้ผู้ใช้อ่านเขียนจดหมายได้อย่างสะดวกโดยไม่ต้องยุ่งเกี่ยวกับการทำงานระดับล่างของโปรโตคอล ส่วน MTA ทำงานตาม SMTP เช่นการตรวจสอบความถูกต้องของ ที่อยู่ผู้รับผู้ส่ง รวมทั้งการหาเส้นทางและนำส่งจดหมายไปยังปลายทาง

2.1.1 องค์ประกอบของจดหมายอิเล็กทรอนิกส์

รูปแบบของจดหมายอิเล็กทรอนิกส์นั้น มีลักษณะคล้ายคลึงกับ จดหมายธรรมดาทั่วไป นั่นคือ ส่วนหลักของจดหมาย เรียกว่า Body และมีส่วนที่เป็นหัวจดหมาย Header ซึ่งจะบอกถึงรายละเอียดของผู้ส่งและผู้รับ และทั้งหมดก็อยู่บนซองจดหมาย ส่วนจดหมายอิเล็กทรอนิกส์นั้นก็ประกอบไปด้วยสามส่วน เช่นเดียวกันคือ Body Header และ Envelope



รูปที่ 2.2 แสดงถึงตัวอย่างส่วนประกอบของจดหมาย

2.1.1.1 The Envelope ส่วนที่เรียกว่า Envelope นั้นคือส่วนที่ MTA(Mail Transfer Agent) มีหน้าที่คอยตรวจสอบว่าอย่างน้อยต้องประกอบไปด้วย ข้อมูลที่สำคัญในการส่งต่อจดหมาย หรือส่งกลับจดหมาย ถูกเรียกใช้งานโดย MTA ในการตรวจหาเส้นทางในการส่งต่อ การตรวจสอบข้อผิดพลาดของจดหมาย เมื่อจดหมายมีการส่งต่อจากที่หนึ่งไปสู่อีกที่หนึ่ง Envelope นั้นจะถูกแยกออกจาก ตัวของจดหมาย โปรโตคอล ที่ใช้ในการที่การรับส่งจดหมายคือ SMTP(Simple Mail Transfer Protocol)(RFC 821) ซึ่งได้มีการกำหนดว่า ข้อมูลใน Envelope ที่จะทำการส่งไปนั้นมีอะไรบ้าง เพราะใน SMTP ได้มีการอธิบายถึง ขั้นตอนของคำสั่งในการแลกเปลี่ยนเมลระหว่าง MTA ซึ่งขั้นตอนเหล่านั้นมีดังนี้

- ที่อยู่ผู้ส่ง ซึ่งใช้คำสั่ง Mail From:
- ที่อยู่ผู้รับ ซึ่งใช้คำสั่ง Rcpt To:
- ส่วนที่เป็นเนื้อความ ซึ่งใช้คำสั่ง Data:
- วิธีการส่ง(ใน RFC 821 ได้มีการระบุวิธีการส่งอยู่สองแบบ)
 - Sending จดหมายส่งโดยตรงสู่เครื่องของผู้ใช้
 - Mailing จดหมายส่งไปยังตู้จดหมาย

ทั้งหมดนี้คือข้อมูลอย่างน้อยที่สุด เพื่อที่จะทำการระบุเพื่อให้สามารถส่งจดหมายไปถึงผู้รับได้

2.1.1.2 The Header รูปแบบของส่วน Header นั้นถูกระบุใน RFC 822 โดยประกอบไปด้วยฟิลด์ต่างๆแต่ละฟิลด์แยกกันคนละบรรทัด ส่วนประกอบสำคัญที่ระบุไว้เป็นมาตรฐานมีสี่ชนิดคือ Date Fields, Source Fields, Destination Fields และ Optional Fields

- Date Fields คือส่วนที่บ่งบอกเวลาในการสร้างจดหมายซึ่งประกอบไปด้วยส่วนย่อยดังนี้
 - Date คือเวลาที่แท้จริงเมื่อจดหมายมีการยืนยันที่จะส่ง ซึ่งได้ทำการรวมไว้เป็นส่วนหนึ่งของจดหมาย
- Source Fields คือส่วนที่บ่งบอกว่าจดหมายมาจากไหน ซึ่งมีอยู่ด้วยกันสามประเภทคือ
 - Trace Fields ซึ่งจะแสดงถึงเส้นทางว่าจดหมายมาจากที่ใด ประกอบด้วยสองส่วนคือ
 - Received เป็นส่วนที่ถูกเพิ่มเข้าไปโดย MTA แต่ละตัวที่จดหมายทำการส่งผ่านมา เนื้อหาส่วนนี้จะระบุถึงชื่อโฮสต์ล่าสุด เวลาที่รับล่าสุด
 - Return-Path คือส่วนที่ระบุถึงเส้นทางในการตอบกลับของจดหมายตามลำดับของผู้ส่ง ซึ่งส่วนนี้จะถูกเพิ่มโดยระบบส่งตัวสุดท้ายที่ทำการส่งจดหมายนั้น
 - Originator Fields ระบุถึงผู้สร้างจดหมาย
 - Authentic Originator Fields คือส่วนที่ระบุถึงตัวตนจริงๆของผู้สร้างจดหมาย คนที่ส่งคนแรก
 - Reply-To ระบุถึงที่หมายที่จดหมายตอบกลับจะส่งไป
 - Resent Fields มีลักษณะคล้ายคลึงกับ Originator fields
- Destination Fields ฟิลด์ที่ระบุถึงปลายทางของจดหมาย
 - To ระบุให้ผู้รับคือใคร
 - Cc ระบุถึงผู้รับจดหมายคนอื่นๆโดยที่จดหมายจะทำการส่งไปถึง เหมือนกับ To ผู้รับสามารถมองเห็นชื่อผู้รับคนอื่นๆได้
 - Bcc ใช้ในการระบุชื่อของผู้รับโดยจะทำการซ่อนชื่อของผู้รับคนอื่นๆในช่องนี้ไว้
- Optional Fields คือข้อมูลที่เพิ่มเติมเข้ามาในส่วนของ Header เช่น
 - Message-ID ระบุถึง uniqueID ของจดหมายโดย MTA ตัวแรกที่ได้รับจดหมายเป็นผู้รวมเข้าไปในส่วนหนึ่งของจดหมาย
 - In-Reply-To ระบุข้อมูลเกี่ยวกับจดหมายสำหรับการตอบจดหมายกลับ

- Keywords คือส่วนที่เป็น Keyword ที่ใช้ในการอ้างถึงตัวจดหมาย ซึ่งถูกใช้บ่อยมากสำหรับเครือข่ายประเภทข่าว
- Encrypted ประกอบไปด้วยข้อมูลของกระบวนการเข้ารหัส (ถ้าจดหมายนั้นมีการเข้ารหัส)

2.1.1.3 The Body คือส่วนของเนื้อหาของจดหมาย ใน RFC822 นั้นได้มีการระบุไว้ว่าข้อความในส่วนที่เป็น body นั้นจะต้องอยู่ในรูปแบบของ 7-bit ASCII ทำให้เป็นข้อจำกัดอย่างมากที่ข้อมูลที่ส่งมากับจดหมายนั้นเป็นได้เพียงแค่ข้อความธรรมดา หรือหากต้องการส่งข้อมูลที่เป็นไบนารีจะต้องอาศัยโปรแกรมอย่างเช่น uuencode เพื่อแปลงข้อมูลไบนารีเป็น ASCII แล้วจึงไปพร้อมกับจดหมาย

2.1.1.4 MIME (Multipurpose Internet Mail Extensions) เป็นข้อกำหนดขยายการทำงานของอินเทอร์เน็ตเมลให้สามารถรับส่งข้อมูลไบนารีได้โดยไม่ต้องใช้โปรแกรมแปลงข้อมูล วิธีการของ MIME คือกำหนดเฮดเดอร์เพิ่มเติมขึ้นมาเพื่อบ่งบอกว่าอ็อบเจกต์ในจดหมายมีลักษณะเฉพาะอย่างไร เฮดเดอร์ที่ใช้ใน MIME มีดังนี้คือ

- MIME-Version : กำหนดรุ่นของ MIME
- Content-Type : กำหนดแบบอ็อบเจกต์
- Content-transfer-encoding : กำหนดว่าอ็อบเจกต์เข้ารหัสโดยใช้วิธีใด
- Content-ID : รหัสสำหรับอ้างอิงอ็อบเจกต์
- Content-Description : คำอธิบายลักษณะอ็อบเจกต์

2.1.2 ขั้นตอนการทำงานของ การรับส่งจดหมายอิเล็กทรอนิกส์

ระบบการรับส่งจดหมายอิเล็กทรอนิกส์เป็นไปตามลักษณะของ Client-Server โปรแกรมของทั้งสองฝั่งทำงานร่วมกันเพื่อที่จะทำการขนส่งจดหมายจากฝั่งผู้ส่งไปยังผู้รับเมื่อผู้ใช้ส่งจดหมาย โปรแกรมบนเครื่องผู้ส่งจะทำหน้าที่เป็น Client จะทำหน้าที่ติดต่อไปยัง Mail Server บนเครื่องของผู้รับและเริ่มทำการถ่ายข้อมูลของจดหมาย Server จะทำการบันทึกจดหมายลงในตู้รับของผู้รับ ส่วนที่ทำหน้าที่ในการส่งผ่านจดหมายระหว่าง Server นี้เราเรียกว่า Mail Transfer Agent (MTA) เมื่อ MTA ได้รับจดหมายที่จะทำการส่งมาจากผู้ส่งตัว MTA จะมีกระบวนการตรวจสอบตัวจดหมายก่อนดังนี้

1. ทำการวิเคราะห์ ส่วนที่เป็น Header ของตัวจดหมายว่ามีข้อมูลที่สำคัญครบถ้วนหรือไม่ (หรืออาจจะทำการเปรียบเทียบกับฟิลด์ต่างๆใน envelope ถ้าจดหมายนั้นมีส่วนของ envelope)

2. ถ้ามีส่วนใดส่วนหนึ่งหายไปหรือไม่ครบถ้วนจาก header ตัว MTA จะทำการสร้างฟิลด์นั้น ฟิลด์ที่ MTA จำเป็นต้องสร้างเช่น From, To, Message-ID, Date
 3. ถ้ายังไม่มีในส่วนของ envelope ตัว MTA จะทำการสร้าง envelope ขึ้นมาจากข้อมูลของ Header ซึ่งจะใช้ข้อมูลในส่วนของ From, To
- เมื่อทำการตรวจสอบตัวจดหมายเรียบร้อยแล้ว จึงจะสามารถทำการ Forwarding Mail ไปยัง MTA ตัวอื่นๆ ได้ กระบวนการในการ Forwarding ทุกๆ MTA จะทำกระบวนการนี้เหมือนกัน ทั้งหมดตั้งแต่ต้นทางไปจนถึงปลายทาง กระบวนการเหล่านั้นประกอบไปด้วย

- **Routing** MTA แต่ละตัวต้องทำการตรวจสอบจดหมายที่ได้รับและทำการตัดสินใจว่า จะต้องส่งต่อไปยัง MTA ตัวไหนเป็นตัวถัดไป ข้อมูลที่ใช้ในการตัดสินใจก็คือส่วนที่เป็น envelope ของจดหมาย โดยส่วนที่เป็น Header นั้นไม่ควรมีการเปลี่ยนแปลง ยกเว้นในกรณีที่ ตัวจดหมายนั้นมีการส่งออกมาจาก โดเมนย่อยที่ไม่สามารถเข้าถึงได้จากภายนอก รูปแบบของผลลัพธ์ในการค้นหาเส้นทางสามารถแสดงได้ตามรูปแบบดังนี้ <next host, next address, delivery method> ซึ่งระบุถึงโฮสต์ถัดไปที่จดหมายควรทำการส่งต่อไป ที่อยู่ที่ถูกระบุลงใน envelope และวิธีการส่ง
- **Exception Handling** ในบางครั้ง MTA ไม่สามารถทำการส่งจดหมายไปยังปลายทางได้ด้วยในหลายหลายสาเหตุ หนึ่งในเหตุผลนั้นเช่น การพิมพ์ที่อยู่ของผู้รับผิดไป หรือ โฮสต์นั้นดาวน์ MTA ต้องหาวิธีในการจัดการกับจดหมายนั้นโดยไม่ทำให้สูญหาย ในกรณีที่ผู้ส่งพิมพ์ที่อยู่ของผู้รับผิดไปนั้น จดหมายควรมีการส่งคืนกลับไปยังผู้ส่ง ถ้าโฮสต์ปลายทางเกิดดาวน์ MTA ควรจะทำการเก็บจดหมายนั้นและพยายามส่งใหม่ในภายหลัง
- **Transactions** กรณีผิดพลาดในบางอย่างเช่นเกิด การเชื่อมต่อขาดหายไประหว่างที่ทำการส่งจดหมายตัว MTA ควรจะมีการจัดการที่ฉลาด เช่น การทำเครื่องหมายในตัวจดหมายที่ทำการส่งไปและแน่ใจว่าจดหมายนั้นถึงมือผู้รับเรียบร้อยแล้ว
- **Flow Control** คือในส่วนของการติดต่อระหว่าง MTA กับ MTA ข้างเคียง ซึ่งต้องมีการติดต่อสื่อสารเพื่อให้ทราบถึงสถานะของ MTA ตัวถัดไป ว่าไม่มีการติดต่อกับที่อื่นในเวลาเดียวกัน ซึ่งจะสามารถช่วยให้รองรับการทำงานได้ทันทีโดยไม่ทำงานอื่นที่ไม่จำเป็น โดยในบางโปรแกรม จะพยายามส่งจดหมายเดิมออกมาเป็นกลุ่มในเวลาเดียวกัน เพื่อให้แน่ใจว่าจดหมายจะไม่ซ้ำ MTA ต้องมีการตรวจสอบ ในบางกรณีที่จดหมายทำการส่งออกไปได้ซ้ำ ซึ่งสาเหตุอาจเกิดมาจากคุณภาพในการเชื่อมต่อของ MTA ในแต่ละที่ ทำให้ส่งจดหมายออกไปได้ซ้ำ จึงทำให้จดหมายอื่นๆ ทำการส่งออกไปได้ซ้ำด้วย MTA จึงต้องมีการ

จัดสรรคิว (mail queue) เพื่อที่จะสามารถรับประกันได้ว่าจดหมายจะไม่ตกหล่นหรือถูก
แข่งคิว

2.1.3 Simple Mail Transfer Protocol (SMTP)

SMTP คือ ข้อตกลงมาตรฐานในการส่งจดหมายอิเล็กทรอนิกส์ระหว่างโฮสต์บนเครือข่าย
TCP/IP ซึ่งมีการกำหนดไว้ใน RFC 821 จดหมายที่มีการส่งข้อมูลผ่าน SMTP นั้นต้องมีการทำตาม
รูปแบบที่กำหนดไว้ใน RFC 822 ซึ่ง SMTP ก็คือ ข้อกำหนด สำหรับ MTA ใช้ในการส่งจดหมาย
ออกไปยัง MTA อื่นๆ ตามหลักการที่กล่าวมาในหัวข้อก่อนหน้านี้ ส่วนที่จะกล่าวเพิ่มเติมคือใน
ส่วนของ ตัวอย่างของคำสั่งที่มีใช้กันใน SMTP

ในแต่ละขั้นตอนการทำงานของ SMTP ประกอบไปด้วย ชุดของคำสั่งและผลลัพธ์ที่
ได้ตอบกลับมา ระหว่าง SMTP ของฝั่งส่งและฝั่งรับ เมื่อ SMTP ฝั่งส่งเริ่มทำการเชื่อมต่อผ่านทาง
TCP จะมีชุดคำสั่งส่งออกไป ในทางที่เชื่อมต่อแต่ละคำสั่งที่ส่งออกไปก็จะมีคำตอบรับ กลับมา
หนึ่งครั้งเสมอ ใน ตารางที่ 2.1 นั้นได้แสดงให้เห็นถึงคำสั่งที่มีการใช้งานในการรับส่งจดหมายของ
SMTP ซึ่งแต่ละคำสั่งจะประกอบไปด้วย ข้อความเพียงหนึ่งบรรทัด คำสั่งข้างล่างเป็นคำสั่งที่ SMTP
รู้จัก

เมื่อมีการส่งคำสั่ง SMTP ต่างๆ ไปแล้วนั้นก็จะมีการตอบกลับถึงสถานะของคำสั่ง ว่า
สามารถปฏิบัติตามได้หรือไม่ ตามตารางที่ 2.2 เป็นตัวอย่างบางส่วนของสถานะของคำสั่งที่กระทำ
ซึ่งแสดงด้วยตัวเลขสามหลักและบางครั้งอาจตามด้วยรายละเอียด ซึ่งชนิดของสถานะตอบกลับ
สามารถแยกได้เป็นสี่รูปแบบคือ

- Positive Completion คือสามารถทำตามชุดคำสั่งต่างๆ ได้อย่างสมบูรณ์และรองรับคำสั่งชุด
ใหม่
- Positive Intermediate reply คือ รับทราบคำสั่งดังกล่าว แต่อาจจะยังไม่ได้ทำตามคำสั่งนั้น
หรือ ค้างเพื่อรอรับรายละเอียดอื่นๆ ผู้ส่งต้องส่งคำสั่งอื่นๆ มาเพื่อบ่งรายละเอียด
- Transient Negative Completion reply คือ ไม่ยอมรับคำสั่งและการกระทำที่ร้องขอไม่
เกิดขึ้น อย่างไรก็ตาม นั่นเป็นแค่ชั่วคราวอาจร้องขอคำสั่งเดิมกลับมาอีก
- Permanent Negative Completion reply คือ ไม่ยอมรับคำสั่งและคำสั่งนั้นไม่ได้กระทำ

ตารางที่ 2.1 แสดงถึงตัวอย่างคำสั่งของ SMTP ที่ใช้ในการติดต่อของฝั่งผู้ส่ง

ข้อความ	รูปแบบของคำสั่ง	รายละเอียด
HELO	HELO<SP><domain><CRLF>	ทำการเริ่มต้นการส่ง
MAIL	MAIL<SP>FROM:<reverse-path><CRLF>	บ่งบอกต้นทางของจดหมาย
RCPT	RCPT<SP>TO:<forward-path><CRLF>	บ่งบอกจุดหมายปลายทาง
DATA	DATA<CRLF>	ข้อมูลในจดหมาย
RSET	RSET<CRLF>	ยกเลิกรายการ
NOOP	NOOP<CRLF>	ไม่มีการทำงาน
QUIT	QUIT<CRLF>	ปิดการเชื่อมต่อ
SEND	SEND<SP>FROM:<reverse-path><CRLF>	ส่งจดหมายไปปลายทาง
SOML	SOML<SP>FROM:<reverse-path><CRLF>	ส่งจดหมายไปปลายทาง
SAML	SAML<SP>FROM:<reverse-path><CRLF>	ส่งจดหมายไปปลายทางและไปผู้จดหมาย
VRIFY	VRIFY<SP><string><CRLF>	ยืนยันชื่อผู้ใช้
EXPN	EXPN<SP><string><CRLF>	ส่งคืนสมาชิกภาพของ mailing list
HELP	HELP[<SP><string>]<CRLF>	สอบถามหาคู่มือ ช่วยเหลือ
TURN	TURN<CRLF>	เปลี่ยนบทบาทผู้รับผู้ส่ง

<CRLF> = carriage return, line feed
<SP> = ช่องว่าง

ตารางที่ 2.2 แสดงถึงตัวอย่างของ result code บางส่วน

Code	Description
Positive Completion Reply	
211	System status or system help reply
214	Help message (Information on how to use the receiver or the meaning of a particular non standard command)
220	<domain> Service Ready
221	<domain>Service closing transmission channel
250	Requested mail action okay
251	User not local will forward to <forward-path>
Transient Negative Completion Reply	
421	<domain> Service not available, losing transmission channel (This may be a reply to any command)
450	Requested mail action not taken
451	Requested action aborted
452	Requested action not taken

ตารางที่ 2.2 (ต่อ)	
Permanent Negative Completion Reply	
500	Syntax error, command unrecognized (This may include errors such as command line too long)
501	Syntax error in parameters or arguments
503	Bad sequence of commands
504	Command parameter not implemented

2.2 การดักจับข้อมูลบนระบบเครือข่าย

โปรแกรมประเภทดักจับ (sniffer) เป็นเครื่องมือสำหรับการดักอ่านข้อมูลที่สื่อสารอยู่บนเน็ตเวิร์กเพื่อให้ได้มาซึ่งข้อมูลของผู้อื่นที่ไม่ใช่ของตนเอง ลักษณะการนำไปใช้งานใกล้เคียงกับการดักฟังโทรศัพท์ ซึ่งจะทำให้เครื่องที่ต่อพ่วงอยู่นั้นสามารถได้ยินการสนทนาได้ตอบที่เกิดขึ้นโดยที่เจ้าของไม่รู้ตัว การนำโปรแกรมดักจับข้อมูลมาใช้ ทำให้ความน่าเชื่อถือในความปลอดภัยของการสื่อสารในรูปแบบอิเล็กทรอนิกส์ลดต่ำลงไปมาก ภัยคุกคามอันเนื่องมาจากการถูกดักอ่านข้อมูล โดยโปรแกรมดักจับข้อมูลนั้นสามารถป้องกันและตรวจสอบได้ยาก เนื่องจาก โปรแกรมที่ทำงานอยู่นั้นไม่ได้ส่งผลกระทบใดๆกับเน็ตเวิร์ก จึงมีโอกาสน้อยมากที่จะทราบได้ว่า มีการดักจับข้อมูลเกิดขึ้น แต่ถึงแม้ว่า โปรแกรมประเภทดักจับข้อมูลนี้จะสามารถ ก่อให้เกิดผลร้ายมากกว่าผลดีก็ตาม โปรแกรมประเภทดักจับข้อมูลนี้ ก็สามารถนำไปใช้ประโยชน์ได้ในหลายด้าน เช่น

- **Network Analyzer** เป็นการนำข้อมูลที่ดักจับได้ไปทำการประมวลผล วิเคราะห์คุณสมบัติต่างๆ ของเครือข่ายได้หลายแง่มุม ไม่ว่าจะเป็นการกระจายการใช้งานของเครือข่ายของโปรโตคอลต่างๆ เช่น HTTP, SMTP, FTP เพื่อที่จะทำให้ผู้ดูแลระบบได้ทราบว่า เครือข่ายของตนนั้น ถูกนำไปใช้งานในด้านใดๆบ้าง มากน้อยเพียงใด จะทำให้สามารถจัดการเครือข่ายได้อย่างเหมาะสม หรือในด้านการใช้งานเครือข่าย ในช่วงเวลาต่างๆ เวลาใดมีผู้ใช้งานมากน้อยเพียงใด จะได้มีการจัดสรรแบนด์วิธได้อย่างเหมาะสม
- **Network Debugging Tools** ในบางครั้งการหาสาเหตุความผิดปกติของแอปพลิเคชันต่างๆ จำเป็นที่จะต้องสืบค้นลึกลงไปถึงเนื้อข้อมูลที่รับส่งกันจริงๆ บนเครือข่ายเพื่อหาสาเหตุว่า เหตุใดการทำงานของแอปพลิเคชัน จึงไม่สามารถทำงานได้ตามปกติ การได้เห็นข้อมูลดิบที่รับส่งกันจริง บนเครือข่ายจะสามารถทำให้การวิเคราะห์และแก้ไขปัญหาทำได้อย่างรวดเร็วและถูกต้องมากขึ้น โดยเฉพาะในกรณีที่มีการใช้เครื่องมือในระดับเน็ตเวิร์กเข้ามาเกี่ยวข้อง เช่นมีการส่งข้อมูลผ่านไฟร์วอลล์แล้วมีปัญหา หรือการทดสอบ ACL (Access

Control List) ของเราเตอร์เป็นต้น ซึ่งปัญหาในกรณีนี้ หากไม่มีโปรแกรมประเภทดักจับข้อมูล เพื่อนำข้อมูลมาวิเคราะห์ อาจต้องใช้เวลาในการแก้ปัญหา

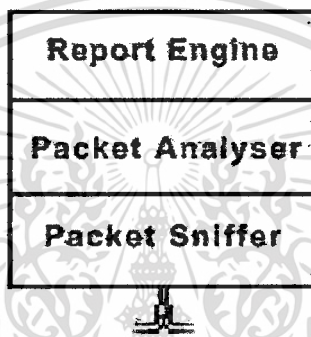
- **Packet Monitoring** การศึกษาเทคนิคของ โปรโตคอลในระดับเน็ตเวิร์กจำเป็นต้องเห็นข้อมูลที่สื่อสารกันอยู่จริง จึงจะเข้าใจและสามารถเห็นภาพได้ Packet Monitoring จะเป็นเครื่องมือที่น่าแพ็คเก็ต มาแสดงให้เห็นในรูปแบบต่างๆ ได้ เพื่อให้ผู้ศึกษาเกี่ยวกับเครือข่าย ใช้ในการศึกษาโปรโตคอลได้ดี
- **Intrusion Detection System** ระบบตรวจจับการบุกรุก มีพื้นฐานมาจากการดักอ่านข้อมูล เพียงแต่เป็นการประยุกต์การวิเคราะห์รูปแบบการบุกรุกเข้ากับอ่านดักอ่านข้อมูลบนเครือข่ายแบบเรียลไทม์ ทำให้สามารถรู้ได้ว่ามีกิจกรรมใด บนเครือข่ายที่มีแนวโน้มว่าจะเป็น การบุกรุกหรือทำอันตรายผู้อื่น
- **Other Security Tools** นอกจากนี้ โปรแกรมดักจับข้อมูลยังสามารถนำไปประยุกต์ใช้ในงานรักษาความปลอดภัยอย่างกว้างขวาง เช่น เพื่อการบันทึกร่องรอยของแฮกเกอร์ จับตาการใช้งานทรัพยากรที่มีความสำคัญ ทดสอบความแข็งแรงของไฟร์วอลล์ เป็นต้น

2.2.1 องค์ประกอบของโปรแกรมประเภทดักจับข้อมูล

โปรแกรมดักจับข้อมูลที่จะสามารถทำงานได้นั้น จะต้องมียุทธศาสตร์ประกอบพื้นฐาน 4 ส่วนคือ

1. **Hardware** ซึ่งเป็นอุปกรณ์ทางอิเล็กทรอนิกส์ต่างๆที่สามารถดักอ่านสัญญาณจากเน็ตเวิร์กเข้ามาได้ และสามารถนำสัญญาณที่ได้ส่งต่อไปประมวลผลออกมาเป็นข้อมูลทางคอมพิวเตอร์ได้ มีหน้าที่หลักคือจัดการกับการรับข้อมูลใน ชั้น Physical เช่น ระดับแรงดัน สัญญาณรบกวน การแก้ไขข้อผิดพลาดของสัญญาณ อุปกรณ์นี้โดยทั่วไปก็คือ การ์ดเครือข่าย (Network Interface Card) ที่ใช้ในปัจจุบันนั่นเอง
2. **Driver** เป็นโปรแกรมระดับล่างที่ควบคุมการดักข้อมูลของฮาร์ดแวร์ และนำสัญญาณที่ได้ไปเก็บไว้ใน Buffer เป็นข้อมูลดิบรอการประมวลผลต่อไป
3. **Buffer** เป็นหน่วยความจำที่ใช้พักข้อมูลจากการดักมาได้ของ Driver โดยจะทำการจัดเก็บเพียงชั่วคราว และหมุนเวียนข้อมูลใหม่เข้ามาเสมอเมื่อมีข้อมูลใดปรากฏขึ้นบน Network กัดไกการนำข้อมูลจาก driver มาเก็บไว้ยัง buffer นี้จะเป็นตัวบ่งบอกสมรรถนะของการดักข้อมูลของโปรแกรมดักจับข้อมูลนั้นๆว่า จะสามารถดักข้อมูลได้ความเร็วสูงสุดเท่าใด หากกระบวนการนำข้อมูลจัดเก็บเป็นไปอย่างล่าช้า ย่อมทำให้โปรแกรมดักจับข้อมูลนั้นๆ ไม่สามารถดักจับข้อมูลบนเครือข่ายได้ทัน ทำให้ต้องปล่อยข้อมูลนั้นทิ้งไป

4. **Software** เพื่อทำหน้าที่จัดการข้อมูลที่ได้รับเข้ามาโดยการประมวลผลตามวัตถุประสงค์ของการดักอ่านข้อมูลนั้น เนื่องจากข้อมูลดิบที่ทำการดักอ่านขึ้นมาได้นั้นจะเป็นข้อมูลในระดับต่ำ คือ DataLink layer ซึ่งจะมีข้อมูลที่ยังไม่ได้ผ่านกระบวนการ Demultiplexing และจัดรูปแบบให้เข้าใจได้ สิ่งที่จะได้จะเป็นข้อมูลในเลขฐานสอง 0 กับ 1 จำนวนมหาศาลที่ต้องการ การแปลความหมาย อีกประการหนึ่งคือข้อมูลที่ดักจับมาได้ เป็นข้อมูลที่ได้มาจากการสื่อสารของ ทุกๆ โฮสต์ ที่ใช้ สายสัญญาณนั้นอยู่พร้อมๆ กัน จึงยังไม่มีแยกแยะว่า เป็นการสื่อสารประเภทใด และเป็นการส่งจากโฮสต์ ใด



รูปที่ 2.3 ส่วนประกอบของโปรแกรมดักจับข้อมูลทั่วไป

2.3 รูปแบบวิธีการดักจับข้อมูลบนเครือข่าย

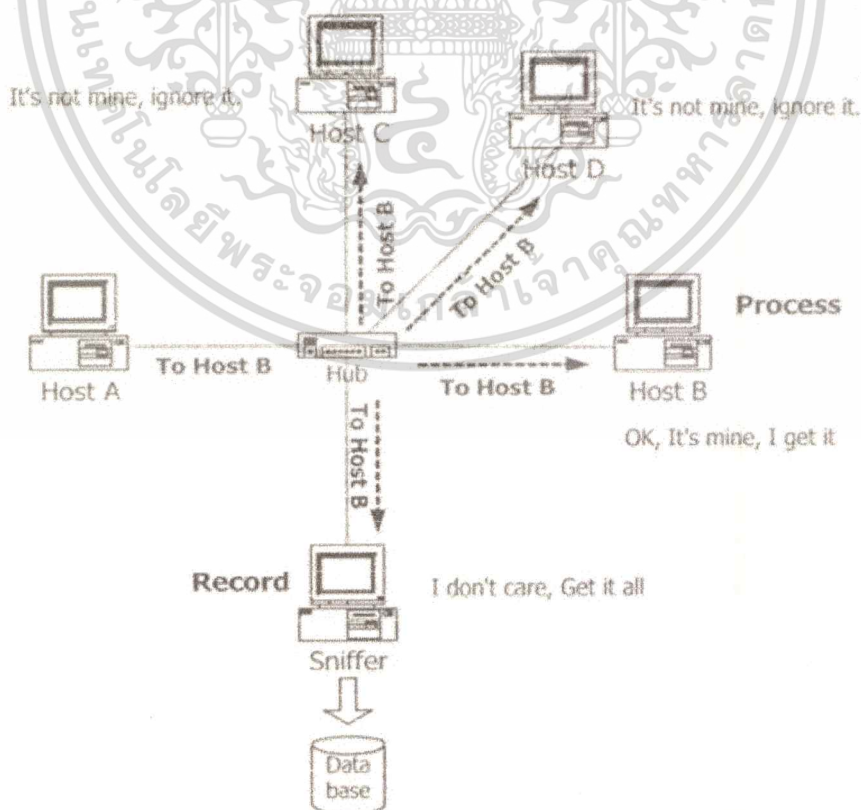
การที่โปรแกรมดักจับข้อมูลสามารถทำการดักอ่านข้อมูลที่อยู่บนเครือข่ายได้นั้น มีสาเหตุที่สำคัญคือ ด้วยลักษณะของโปรโตคอลอีเทอร์เน็ตที่ใช้หลักการกระจายของข้อมูลไปยังทุกโฮสต์ที่อยู่ในเครือข่ายและอาศัยโฮสต์แต่ละตัวทำหน้าที่จำแนกการสื่อสารของตนเอง นั่นหมายความว่าข้อมูลทุกแพ็คเก็ตที่ใช้สื่อสารกันนั้น ได้ถูกส่งไปทุกโฮสต์ ซึ่งจะได้รับพร้อมกันและเหมือนกัน เพียงแต่การที่จะสื่อสารกันได้อย่างถูกต้องนั้น โฮสต์แต่ละตัวจะต้องมีกระบวนการที่สามารถเรียนรู้ได้ว่าข้อมูลแพ็คเก็ตใดเป็นของตนเอง และข้อมูล แพ็คเก็ตใดไม่ใช่

ดังนั้นทุกๆแพ็คเก็ตที่ทำการส่งกระจายไปในเครือข่ายจะต้องมีหมายเลขระบุชัดเจน คือ MAC Address ซึ่งจะเป็นสิ่งที่บอกว่า แพ็คเก็ตใดมาจากฮาร์ดแวร์ใดในเครือข่าย MAC Address จะเป็นหมายเลขเฉพาะตามฮาร์ดแวร์ทุกชนิดที่ใช้การสื่อสารโดยโปรโตคอลอีเทอร์เน็ต และในทางทฤษฎีแล้วฮาร์ดแวร์ทุกชนิดจะไม่มี MAC Address ที่ซ้ำกัน โดยทั่วไป MAC Address จะถูกกำหนดตายตัวอยู่ใน ROM ของฮาร์ดแวร์และโดยทั่วไปจะไม่สามารถเปลี่ยนแปลงได้โดยซอฟต์แวร์ การใช้งานของฮาร์ดแวร์นั้นๆ โดยปกติแล้วใครเวอร์จะถูกกำหนดให้ปฏิบัติตามโปรโตคอล อย่างเข้มงวด คือ

- ใหัรับข้อมูลที่มี MAC Address เป็นของตนเองเท่านั้น
- ใหัส่งข้อมูลโดยใช้ MAC Address ของตนเองเท่านั้น

ดังนั้นหากใช้ ไดรเวอร์ตามปกติที่มากับฮาร์ดแวร์ แล้วเครื่องคอมพิวเตอร์ก็จะทำงานอยู่ในระเบียบเรียบร้อย และไม่สามารถวุ่นวายกับข้อมูลของผู้อื่นได้ แต่อย่างไรก็ตามไดรเวอร์ก็เป็นเพียงโปรแกรมประเภทหนึ่งเท่านั้นที่ทำหน้าที่จัดการกับการสื่อสารข้อมูลในระดับต่ำของฮาร์ดแวร์กับระบบปฏิบัติการ โปรแกรมดักจับข้อมูลมีการใช้งานไดรเวอร์ที่พิเศษกว่า คือทำการเปลี่ยนแปลงโหมดในการทำงานของการ์ดเครือข่าย ให้สามารถทำการรับข้อมูลของผู้อื่นเข้ามาได้โดยไม่มีการปิดกั้น ซึ่งเราเรียกโหมดการทำงานนี้ว่า Promiscuous Mode เป็นโหมดที่ทำให้ฮาร์ดแวร์อ่านข้อมูลดิบทั้งหมดบนเครือข่ายเข้ามาในเครื่องคอมพิวเตอร์ของตนเอง โดยไม่สนใจว่าจะเป็นข้อมูลของเครื่องใด ส่งให้เครื่องใด

ปัจจัยที่สำคัญที่ทำให้โปรแกรมดักจับข้อมูลสามารถดักอ่านข้อมูลของผู้อื่นได้ ก็คือ การใช้สื่อกลางของเครือข่ายร่วมกัน ไม่ว่าจะเป็น 10BaseT, 10Base5, 100BaseTx ล้วนแต่เป็นเทคโนโลยีที่มีการใช้งานสื่อสัญญาณร่วมกัน นั่นหมายความว่า เครื่องทุกเครื่องที่ใช้ที่เชื่อมต่อโดยตรงกับสื่อสัญญาณเดียวกันนั้นย่อมได้รับข้อมูลเหมือนกันและทั่วถึงกัน



รูปที่ 2.4 ภาพจำลองการทำงานของโปรแกรมดักจับข้อมูล

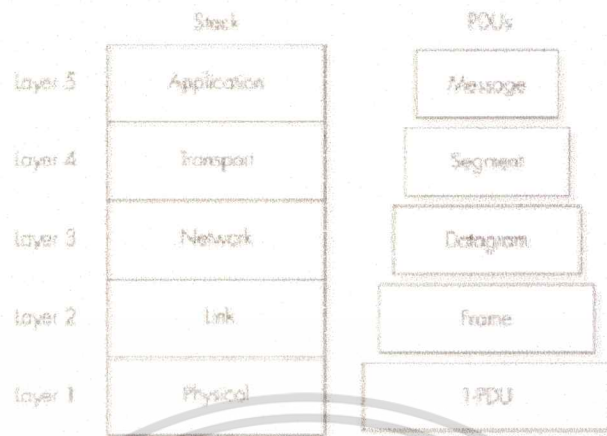
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป มีโฮสต์ทั้งหมด 4 เครื่องต่อเข้าโดยใช้ฮับร่วมกัน นอกจากนี้ ยังมีเครื่องที่ทำมีการรันโปรแกรมดักจับข้อมูลร่วมใช้งานในฮับเดียวกัน เครื่อง A ทำการส่งข้อมูลไปยังเครื่อง B ซึ่งแท้จริงแล้ว เมื่อ เครื่อง A ส่งข้อมูลออกไปนั้น ข้อมูลได้กระจายไปยังทุกๆเครื่องที่เชื่อมต่อเข้ากับฮับตัวเดียวกันนั้น รวมถึงเครื่องที่ทำการใช้งานโปรแกรมดักข้อมูลด้วย โดยเครื่องต่างๆ จะมีปฏิกิริยาดังนี้

- เครื่อง B เมื่อ ได้รับข้อมูลจะทำการตรวจสอบ MAC Address หากพบว่าเป็นข้อมูลที่ส่งมาหาตนเอง ก็จะทำการอ่านและส่งต่อไปให้ระบบปฏิบัติการนำไปประมวลผล
- เครื่อง C และ D เครื่องทั้งสองซึ่งมีการ์ดเครือข่ายทำงานอยู่ในโหมดการทำงานปกติ เมื่อได้ทำการตรวจสอบ MAC Address ของข้อมูลแล้ว พบว่าไม่ใช่ข้อมูลที่จะทำการส่งให้ตนเอง ดังนั้นเครื่องทั้งสองจะไม่สนใจ ข้อมูลชุดนี้
- Sniffer เครื่องนี้ นั้นได้มีการรันโปรแกรมดักจับข้อมูลไว้ ซึ่งทำให้การ์ดเครือข่ายนั้นทำงานอยู่ใน Promiscuous Mode ซึ่งก็คือสนใจข้อมูลทุกชนิด โดยไม่สนใจว่าเป็นของเครื่องใด แล้วทำการเก็บไว้ใน buffer เพื่อนำไปใช้ประโยชน์ต่อไป

2.4 โพรโทคอล TCP/IP

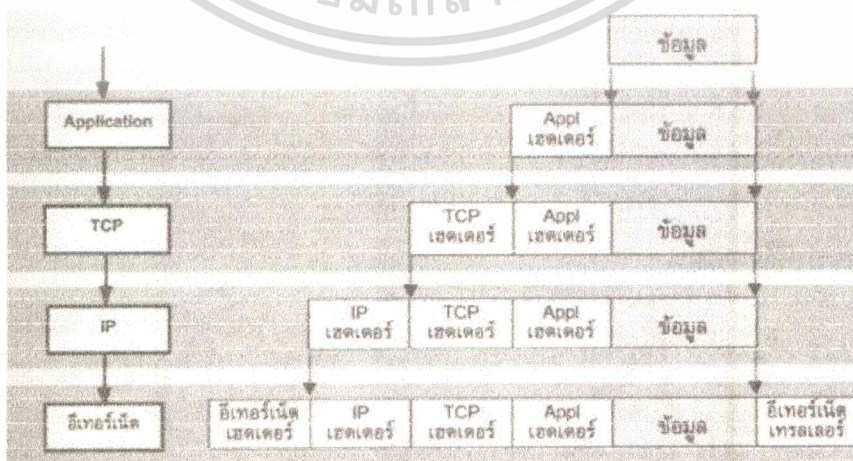
TCP/IP เป็นโปรโตคอลที่ใช้งานอย่างแพร่หลายในแทบทุกเครือข่าย โดยผ่านการออกแบบให้เป็นอิสระจาก ชนิดของเครื่องคอมพิวเตอร์ ฮาร์ดแวร์และระบบปฏิบัติการ มีความน่าเชื่อถือสูง ชื่อ TCP/IP มีที่มาจากโปรโตคอลสองโปรโตคอล คือ TCP (Transmission Control Protocol) และ IP (Internet Protocol) TCP มีหน้าที่หลักๆคือการทำหน้าที่รับประกันความถูกต้องในการลำเลียงข้อมูล ส่วน IP มีหน้าที่หลักคือ การกำหนดแอดเดรส จัดแบ่งขนาดข้อมูลให้พอเหมาะและเลือกเส้นทางในการส่งข้อมูล ในการใช้งานในระบบเครือข่าย ไม่ได้มีเพียงแค่สองโปรโตคอล นี้เท่านั้น หากแต่ยังมีโปรโตคอลสนับสนุนอีกเป็นจำนวนมาก รวมกันเป็นชุดโปรโตคอล TCP/IP



รูปที่ 2.5 ภาพลำดับชั้นของชุดโปรโตคอล TCP/IP

โปรโตคอลแต่ละชั้นล้วนมีหน้าที่เกี่ยวข้องในการส่งผ่านข้อมูลจากต้นทางไปยังปลายทาง ข้อมูลจะถูกส่งผ่านจากโปรโตคอลระดับบนสุด จากต้นทางไปยังระดับล่าง จนกระทั่งข้อมูลถูกแปลงให้อยู่ในรูปสัญญาณไฟฟ้า แล้วเดินทางผ่านเครือข่ายไปยังปลายทาง โปรโตคอลระดับล่างสุดที่เครื่องปลายทางแล้วส่งผ่านไปยังโปรโตคอลระดับบนสุดต่อไป

ในระหว่างที่ข้อมูลผ่านโปรโตคอลในแต่ละชั้นนั้น จะมีการผนวกข้อมูลที่เกี่ยวข้องกับการทำงานประจำโปรโตคอล ซึ่งเรียกว่า Protocol Header เข้ากับข้อมูล ส่วนเฮดเดอร์และตัวข้อมูลจากระดับบนจะถูกส่งผ่านไปยังระดับล่าง โปรโตคอลในระดับล่างจะมองเฮดเดอร์และข้อมูลนั้นรวมกันเปรียบเสมือนเป็นข้อมูลชุดเดียวกัน ข้อมูลเดิมเมื่อมาถึงชั้นล่าง จึงมีเฮดเดอร์หุ้มเป็นชั้นๆ ซึ่งเรียกกระบวนการนี้ว่า การเ็นแคปซูล (Encapsulation)



รูปที่ 2.6 ภาพการห่อหุ้มข้อมูลตามลำดับชั้นของโปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเครื่องปลายทางได้รับข้อมูล ก็จะดำเนินการไปตามลำดับชั้น โปรโตคอลประจำชั้นจะทำการถอดส่วนเฮดเดอร์ออกและส่งส่วนที่เหลือไปยังชั้นบนถัดไป เฮดเดอร์ที่ถูกถอดออกนี้จะเหลือเฉพาะข้อมูลเมื่อถึงชั้นบนสุด กระบวนการนี้เรียกว่า การดีแคปซูลเลต (Decapsulation)

2.4.1 Ethernet

อีเทอร์เน็ต โดยทั่วไปหมายถึง มาตรฐานการสื่อสารข้อมูล ที่ใช้สำหรับการรับส่งข้อมูลบน LAN (Local Area Network) โดยใช้วิธีการเข้าถึงข้อมูลแบบ CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ซึ่งทำให้อุปกรณ์บน LAN สามารถใช้สายรับ-ส่งสัญญาณร่วมกัน และรับส่งข้อมูลพร้อมๆ กันได้ด้วยความเร็ว 10Mbps โดยมีแอคเครสในการระบุต้นทาง-ปลายทาง ในการรับส่งข้อมูลขนาด 48 บิต ในเวลาต่อมาได้มีการพัฒนาและรวบรวมมาตรฐานต่างๆ ที่คล้ายๆ กัน เข้าด้วยกัน ทำให้โครงสร้างของอีเทอร์เน็ต ดูยุ่งยากและมีความซับซ้อนมากขึ้น เพราะมีทั้งที่ยังอิงจาก RFC 894 และ RFC 1042 ซึ่งโพรโทคอลที่กำหนดทั้งสอง มิได้ขัดแย้งกัน แต่ก็มีจุดที่ต่างกันเล็กน้อย คือ ตาม RFC 894 นั้นจะมีการใช้ข้อมูลไบต์ที่ 12-13 เป็นไบต์ที่ระบุชนิดของโปรโตคอลที่ถูกเอ็นแคปซูลเลตอยู่ แต่สำหรับ RFC 1042 นั้น ในไบต์ที่ 12-13 กลับใช้เป็นที่ระบุความยาวของ Ethernet Frame

ข้อมูลที่รับส่งกันในชั้นของ Link Layer นั้น จำเป็นต้องมีอีเทอร์เน็ตแอคเครสของทั้งต้นทางและปลายทางชื่อ แอคเครสที่ใช้ในชั้นนี้ เรียกว่า MAC Address ซึ่งเป็นแอคเครสที่ถูกกำหนดมาตั้งแต่กระบวนการผลิตของฮาร์ดแวร์ ที่จะไม่ซ้ำกัน

2.4.2 IP (Internet Protocol)

ไอพี เป็นโปรโตคอลในระดับชั้นเน็ตเวิร์กทำหน้าที่กำหนดรูปแบบแอคเครสประจำเครื่องเพื่อนำส่งข้อมูล นอกจากนี้ยังทำหน้าที่ในการเลือกเส้นทางส่งข้อมูลและแบ่งข้อมูลให้เหมาะสมกับโปรโตคอลระดับล่าง ไอพีมีรูปแบบการส่งเดทาแกรมสองรูปแบบ คือ แบบ unreliable และแบบ connectionless

ความหมายของ unreliable คือ ไอพีไม่มีกลไกการรับประกันว่าเดทาแกรมที่ส่งไปจะถึงปลายทางได้สำเร็จ ไอพีให้บริการถ้าเลียงเดทาแกรมอย่างดีที่สุด หากมีความผิดปกติใดเกิดขึ้นระหว่างการลำเลียงส่ง เช่น บัฟเฟอร์ของเราเตอร์ระหว่างทางเต็ม สิ่งที่ไอพีดำเนินการคือ ทิ้งเดทาแกรมนั้นทิ้งไปแล้วรายงานสาเหตุของปัญหากลับไปด้วยโปรโตคอลไอซีเอ็มพี (ICMP : Internet Control Message Protocol) ส่วนความหมายของ connectionless คือ จะไม่มีสถานะ การเชื่อมต่อ

ระหว่างต้นทางกับปลายทาง เพื่อกำหนดเส้นทางในการส่งข้อมูล ไอพีเดทาแกรมแต่ละชิ้นเป็นอิสระจากกัน และมีโอกาสไปถึงปลายทางได้โดยไม่เรียงลำดับ

Version	IHL	Type service	Total lenght	
Identification			DM	Fragment offset
Time to live	Protocol		Header checksum	
Source address				
Destination addresses				
Options (0 or more words)				
Beginning of data				

รูปที่ 2.7 แสดงถึง โครงสร้างของ IP Packet Header

ไอพีเดทาแกรมเฮดเดอร์ โดยปกติจะมีขนาด 20 ไบต์ ยกเว้นในกรณีที่มีการเพิ่มเติมอปชัน มีโครงสร้างตามรูปที่ 2.7 สำหรับข้อมูลแต่ละส่วนในเฮดเดอร์มีความหมายดังนี้

- Version Number บิตที่ 0-3 แสดงรุ่นของโปรโตคอล รุ่นที่ใช้งานอยู่ในปัจจุบันมีค่า 4
- Header Length บิตที่ 4-7 บอกความยาวของไอพีเดทาแกรมเฮดเดอร์โดยนับจาก version ไปจนถึงไบต์สุดท้ายของก่อนที่จะถึงข้อมูล หน่วยนับความยาวจะบอกเป็นจำนวนเท่าของ 4 ไบต์ (32 บิตเวิร์ด) โดยทั่วไปถ้าไม่มีออปชันส่วนนี้จะมีค่าเป็น 5 หมายความว่าความยาวข้อมูลมีขนาด 5 จะหมายถึงไอพีเฮดเดอร์มีขนาด 20 ไบต์
- TOS (type of service) บิตที่ 8-15 มีขนาด 8 บิต ปัจจุบันไม่ได้ใช้งานแล้ว ถูกออกแบบมาเพื่อเก็บค่าของ Minimize Delay, Maximize Throughput, Maximize reliability, Minimize Monetary cost เพื่อใช้เป็นตัวบ่งชี้ให้เราเตอร์ตัดสินใจในการเลือกข้อมูลแต่ละเดทาแกรม
- Total length บิตที่ 16-31 มีขนาด 16 บิต บอกความยาวทั้งหมดของเดทาแกรม โดยมีหน่วยนับเป็นไบต์เนื่องจากฟิลด์นี้มีขนาด 16 บิต แสดงว่าขนาดความยาวข้อมูลของ ไอพีเดทาแกรมจะมีขนาดได้สูงสุด 65535 ไบต์ ขึ้นอยู่กับขนาด MTU โดยทั่วไปแล้ว ถึงแม้ว่าเราจะสามารถส่งข้อมูลใน 1 เดทาแกรมได้สูงสุดถึง 65535 ไบต์ แต่เมื่อข้อมูลถูกส่งลงไป ในโปรโตคอลระดับล่างก็จะถูกแบ่งย่อยก่อนทำการส่งจริงอยู่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Identification บิตที่ 32-47 ขนาด 16 บิต เป็นหมายเลขของเดทาแกรมที่ส่งในกรณีที่มีการกระจายเดทาแกรมและนำกลับมารวมกันใหม่จะรู้ได้ว่ามาจากเดทาแกรมเดียวกัน
- Flags บิตที่ 48-50 ขนาด 3 บิต ใช้ในกรณีที่มีการแยกย่อยข้อมูลของเดทาแกรม
- Fragmentation Offset บิตที่ 51-63 ขนาด 13 บิต ใช้ในการกำหนดตำแหน่งของข้อมูลใน 1 เดทาแกรม ที่ถูกแยกย่อยกลับมาต่อเรียงกันในตำแหน่งข้อมูลที่ต้องการ
- Time-to-live (TTL) บิตที่ 64-71 ขนาด 8 บิต ใช้ในการบ่งบอกจำนวนครั้งสูงสุดที่เดทาแกรมนี้อาจเดินทางผ่านได้ หรืออีกความหมายหนึ่งก็คือการกำหนดอายุของเดทาแกรม ซึ่งมีค่าได้สูงสุด $2^8 - 1$ คือ 255 ซึ่งเครื่องต้นทางจะทำการกำหนดค่านี้ไว้ และเมื่อข้อมูลผ่านเราเตอร์ 1 ตัว ค่านี้จะถูกลดลง 1 ซึ่งเมื่อ TTL มีค่า 0 แสดงว่า เดทาแกรมนั้นหมดอายุ เราเตอร์ก็จะครอบข้อมูลชุดนั้นทิ้งไป
- Protocol บิตที่ 72-79 ขนาด 8 บิต ฟังก์ชันบ่งบอกชนิดของโปรโตคอลระดับบนที่เอ็นแคปซูลเลตอยู่ในเดทาแกรม ว่าเป็นชนิดอะไร เช่น TCP, UDP, ICMP
- Header Checksum บิตที่ 80-95 ขนาด 16 บิต ใช้ในการตรวจสอบความถูกต้องของข้อมูลเฉพาะในส่วนหัวของ Datagram เท่านั้น Host ทุกเครื่องเมื่อรับ Datagram มาแล้วจะต้องตรวจสอบข้อนี้ ถ้าค่าที่คำนวณได้ไม่ตรงกับค่าที่อ่านได้จากข้อมูลนี้ก็แสดงว่าเกิดข้อผิดพลาดขึ้นในระหว่างการรับ-ส่งข้อมูล ซึ่งจำเป็นจะต้องครอบ Datagram นี้แล้วหาทางแก้ไขวิธีอื่นต่อไป ถ้าผลการคำนวณตรงกับค่าที่อ่านได้ ก็จะต้องคำนวณค่านี้ใหม่ (เพราะค่าของ Time to live เปลี่ยนไปแล้ว) และเก็บค่าใหม่ไว้ในเขตข้อมูลนี้เพื่อรอการทำงานอื่นต่อไป
- Source IP Address บิตที่ 96-127 ขนาด 32 บิต บรรจุ IP Address ของผู้ส่งข้อมูลเดทาแกรม
- Destination IP Address บิตที่ 128-163 ขนาด 32 บิต บรรจุ IP Address ปลายทางผู้รับข้อมูลเดทาแกรม
- Options ขนาดไม่คงที่ ใช้สำหรับกำหนดข่าวสารเพิ่มเติมสำหรับเดทาแกรม ค่าที่ใช้ในปัจจุบันจะเกี่ยวข้องกับการรักษาความปลอดภัย และการบันทึกผลลัพธ์จากการทำงานของคำสั่ง traceroute หรือ ping
- Data (payload) ในส่วนนี้ของ IP ประกอบไปด้วย ข้อมูลในชั้นของ Transport Layer Segment (TCP หรือ UDP)

2.4.3 TCP (Transmission Control Protocol)

TCP เป็นโปรโตคอลที่ให้บริการชนิดที่ต้องมีการเชื่อมต่อ และรับประกันความเชื่อถือในการส่งข้อมูล โดยทำการตรวจสอบเซกเมนต์ที่ผิดปกติ แล้วทำการส่งซ้ำใหม่ รวมทั้งจัดลำดับให้ถูกต้องก่อนส่งไปยังโปรแกรมระดับบนที่ใช้ข้อมูล ทีซีพีให้บริการจัดการด้านความน่าเชื่อถือของข้อมูล การจัดลำดับ การควบคุมปริมาณการรับ-ส่งข้อมูล โดยมีรายละเอียดที่สำคัญๆดังต่อไปนี้

- ข้อมูลที่จะส่งผ่าน TCP จะถูกนำมาแตกย่อยออกเป็นส่วนๆให้มีขนาดเหมาะสมสำหรับการส่ง โดย TCP จะเป็นตัวพิจารณาว่าขนาดเท่าใดจะทำให้การรับส่ง มีประสิทธิภาพและน่าเชื่อถือสูงสุด โดยข้อมูลแต่ละชุดที่แบ่งออกและทำการส่งโดย TCP แต่ละครั้งจะเรียกว่า TCP เซกเมนต์
- ในการส่งข้อมูลแต่ละครั้งของ TCP จะมีการจับเวลาไว้เสมอเพื่อรอให้อีกฝั่งหนึ่งตอบยืนยันการรับข้อมูลกลับมา หากถึงเวลากำหนดที่ข้อมูลควรจะถึงปลายทางและตอบยืนยันกลับมา แต่ยังไม่มีการตอบกลับมา TCP จะถือว่าข้อมูลที่ส่งไปนั้นไม่ถึงปลายทางและจะทำการส่งข้อมูลซ้ำ
- TCP มี Checksum ซึ่งจะครอบคลุมทั้ง เฮดเดอร์และ ส่วนที่เป็นข้อมูล เพื่อเป็นการป้องกันและตรวจสอบว่าข้อมูลที่ส่งมานั้นถูกต้อง และ ไม่ได้ถูกแก้ไขระหว่างทาง หาก TCP ได้รับข้อมูลที่ทำการตรวจสอบกับ Checksum แล้วปรากฏว่ามีความผิดพลาดเกิดขึ้น TCP จะทิ้งข้อมูลนั้นและไม่ทำการตอบรับข้อมูลนั้นกลับไปยังผู้ส่ง คือให้เสมือนว่าไม่ได้รับข้อมูลนั้น เพื่อให้ฝ่ายส่ง ทำการส่งซ้ำกลับมาใหม่
- TCP มีกลไกการควบคุมการไหลของข้อมูล (Flow Control) โดยการควบคุมนี้จะต้องอาศัยลำดับของการรับส่งข้อมูลที่ถูกต้องและสัมพันธ์กันทั้งสองฝั่ง ในขณะที่เดียวกันข้อมูลที่ส่งนั้นจะต้อง อาศัย IP Datagram จึงจะ ได้ข้อมูลครบทั้งหมด ดังนั้นในการรับข้อมูลทางฝ่ายรับจึงต้องเตรียมบัฟเฟอร์ไว้จำนวนหนึ่งเพื่อรองรับข้อมูลและรวบรวมข้อมูลทั้งหมดให้อยู่ในบัฟเฟอร์ก่อนที่จะทำการจัดเรียงข้อมูล ตรวจสอบความถูกต้องแล้วจึงส่งต่อไปยังแอปพลิเคชัน ดังนั้นเพื่อป้องกันการส่งข้อมูลขนาดใหญ่หรือเร็วเกินกว่าที่บัฟเฟอร์จะรองรับได้ การส่งข้อมูลจึงถูกจำกัดโดยจะอนุญาตให้ทำการส่งข้อมูลได้เท่าที่ฝ่ายรับมีบัฟเฟอร์เพียงพอ

ทีซีพีเตทาแกรมเฮดเดอร์ จะเริ่มต้นระบุที่หมายเลขพอร์ตต้นทางและปลายทาง แต่ที่จริงแล้วข้อมูลอีกส่วนหนึ่งที่ใช้ในการสื่อสารคือ IP Address ของต้นทางและปลายทางก็ต้องระบุด้วยเช่นกัน แต่ได้ถูก Encapsulate ไว้ในเลขอร์ของ IP และค่าของ IP Address ทั้งคู่จะอยู่ใน IP Header

ส่วนหัวของ IP Address และหมายเลขพอร์ตนั้นจะเรียกว่า ซ็อกเก็ต (Socket) ซึ่งในการสื่อสารแต่ละครั้งจะต้องมีทั้งซ็อกเก็ตของต้นทางและปลายทางข้อมูล จึงจะสามารถรับส่งข้อมูลไปถูกที่และถูกแอปพลิเคชัน รายละเอียดของ TCP Header มีดังนี้

Source Port		Destination Port	
Sequence Number			
Acknowledgement Number			
Offset	Reserv	Code	Windows
Checksum		Urgent Pointer	
Beginning of data			

รูปที่ 2.8 แสดงถึงโครงสร้างของ TCP Packet Header

- Source Port คือหมายเลขพอร์ตของผู้ส่ง มีขนาด 16 บิต
- Destination Port คือหมายเลขพอร์ตของผู้รับมีขนาด 16 บิต
- Sequence Number ขนาด 32 บิต เป็นข้อมูลที่ระบุถึงหมายเลขลำดับที่ใช้อ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อให้ทั้งสองฝ่ายจะได้รับทราบตรงกันว่าเป็นข้อมูลของชุดใด การนำไปใช้งานจะได้ไม่ปะปนกัน และมีลำดับที่ถูกต้อง เนื่องจากการสื่อสารข้อมูลผ่าน TCP นั้นจึงหะและลำดับเป็นส่วนสำคัญของโปรโตคอลไม่ยิ่งหย่อนไปกว่าข้อมูลใน TCP Header รวมไปถึงการที่ข้อมูลในแต่ละเซกเมนต์ อาจจะถูกทำการแยกย่อยในเลเยอร์ชั้น IP ทำให้ข้อมูลถูกแบ่งออกและส่งไปในลำดับที่ไม่เรียงกัน หากไม่มีจุดอ้างอิงของข้อมูลก็จะไม่สามารถอ่านข้อมูลกลับใหม่ได้อย่างสมบูรณ์และถูกต้อง การส่งข้อมูลจะใช้ ฟิลด์นี้เป็นตัวยืนยันระหว่างกันเสมอ
- Acknowledgement Number ทำหน้าที่เช่นเดียวกับ Sequence Number ต่างกันตรงที่ Sequence Number ซึ่งใช้ในการตอบรับ กล่าวคือ เนื่องจาก Sequence Number ที่ใช้ในการอ้างอิงข้อมูลนั้น ผู้ส่งข้อมูลจะเป็นผู้กำหนดหมายเลขขึ้นมาและส่งไปพร้อมกับการสร้างการเชื่อมต่อครั้งใหม่ แต่สำหรับฝ่ายที่ถูกติดต่อ ก็จำเป็นต้องกำหนดหมายเลขสำหรับใช้อ้างอิงในการตอบรับเช่นกัน ค่าที่อยู่ใน Acknowledgement Number ก็คือหมายเลขที่ใช้ในการอ้างอิงในการตอบรับนี้ ปกติค่าในทั้ง Sequence Number และ Acknowledgement

Number จะต้องพิจารณาประกอบกับ Flag จึงจะสามารถแปลความหมายของ TCP Segment ได้อย่างสมบูรณ์

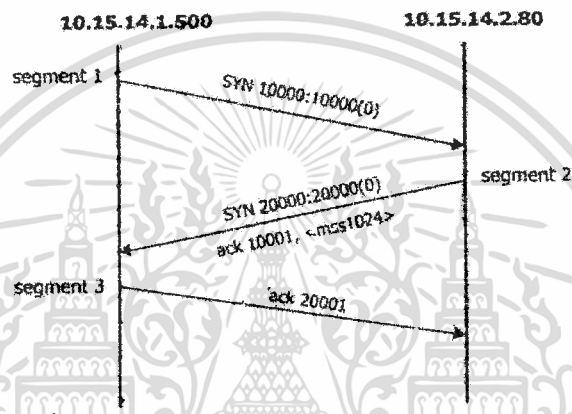
- Offset มีขนาด 4 บิต บอกถึงตำแหน่งเริ่มต้นของไบต์ข้อมูลหรืออีกนัยหนึ่งใช้บอกขนาดของ TCP Header ตัวเลขนี้มีหน่วยเป็นจำนวนเท่าของ 4 ไบต์เช่นเดียวกับที่ใช้ใน IP Datagram Header ซึ่งโดยปกติโดยทั่วไปมีขนาด 20 ไบต์
- Reserved(RSV) ขนาด 4 บิต สำรองไว้ใช้ในอนาคต
- Flag Code เป็นข้อมูลในระดับบิตที่ใช้เป็นตัวบ่งบอกคุณสมบัติของ TCP Segment ที่กำลังส่งอยู่นั้น และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูล ซึ่ง Flag ทั้งหมดนั้นมีอยู่ด้วยกัน 6 บิต แต่ละบิตจะมีความหมายต่าง ๆ กันดังนี้
 - URG ถ้าบิตนี้เป็น 1 แสดงว่า Urgent Pointer บรรจุตำแหน่งข้อมูลที่ต้องรีบดำเนินการเร่งด่วนก่อน
 - ACK ถ้าบิตนี้เป็น 1 แสดงว่าเซกเมนต์เป็นเซกเมนต์ตอบรับ โดยตอบอ้างอิงเลขลำดับตามที่กำหนดในฟิลด์ Acknowledgement Number
 - DSH ถ้าบิตนี้เป็น 1 แสดงว่าทันทีที่ปลายทางได้รับข้อมูลนี้ ต้องรีบนำส่งเซกเมนต์นั้นๆ ไปยังโปรโตคอลระดับถัดไปที่ทันทีโดยไม่ต้องรอให้บัฟเฟอร์เต็ม
 - RST ถ้าบิตนี้เป็น 1 แสดงว่าให้ยกเลิกการเชื่อมต่อนี้ เนื่องจากอาจมีข้อผิดพลาดเกิดขึ้น หากต้องการส่งข้อมูลอีก ต้องเริ่มการเชื่อมต่อใหม่
 - SYN ถ้าบิตนี้เป็น 1 แสดงว่าขอเริ่มต้นการเชื่อมต่อและเมื่อการเชื่อมต่อสำเร็จ บิตนี้จะถูกกำหนดให้เป็น 0 หลังจากนั้นจึงสามารถส่งข้อมูลผ่านไปมาหากันได้
 - FIN ถ้าบิตนี้เป็น 1 แสดงว่าส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ
- Window size ขนาด 16 บิต เป็นขนาดของการรับส่งข้อมูลในแต่ละครั้งที่ทางฝ่ายผู้รับจะสามารถรับได้ เนื่องจากในการรับข้อมูลนั้น ทางผู้รับจะต้องจัดเตรียมหน่วยความจำในการพักข้อมูลทีมาจาก TCP หากไม่มีการตกลงถึงขนาดที่ทางฝ่ายรับสามารถรับได้ ก็จะทำให้การสื่อสารข้อมูลไม่สมดุลและฝ่ายรับอาจจะประมวลผลไม่ทัน ซึ่งทำให้ฝ่ายส่งอาจต้องส่งข้อมูลซ้ำหลายครั้ง
- Checksum มีขนาด 16 บิต ใช้ในการตรวจสอบความถูกต้องของข้อมูล โดยคำนวณทั้งส่วนของเฮดเดอร์และข้อมูล
- Urgent pointer ขนาด 16 บิต ใช้ระบุหมายเลข Sequence Number ของ TCP Segment ลำดับที่อยู่ใหม่ urgent
- Option ขนาดสามารถแปรเปลี่ยนได้ ใช้กำหนดงานเพิ่มเติมให้กับ TCP ซึ่งจะมีหรือไม่ก็ได้

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่สามารถนำออกเผยแพร่ได้โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 กระบวนการส่งข้อมูลผ่านโปรโตคอล TCP/IP

ก่อนที่จะ TCP จะสามารถรับส่งข้อมูลกันได้ จะต้องมีการสถาปนา (Establishment) การเชื่อมต่อหรือมีการสร้าง Connection เกิดขึ้นก่อน เปรียบเสมือนการต่อสองสายของทั้งสองฝั่งให้เชื่อมถึงกัน ซึ่งโปรโตคอล TCP ได้กำหนดขั้นตอนในการเริ่มสร้าง Connection ไว้ดังนี้

2.5.1 การสร้างการเชื่อมต่อ



รูปที่ 2.9 ลักษณะการ 3-Ways handshakes ของ TCP

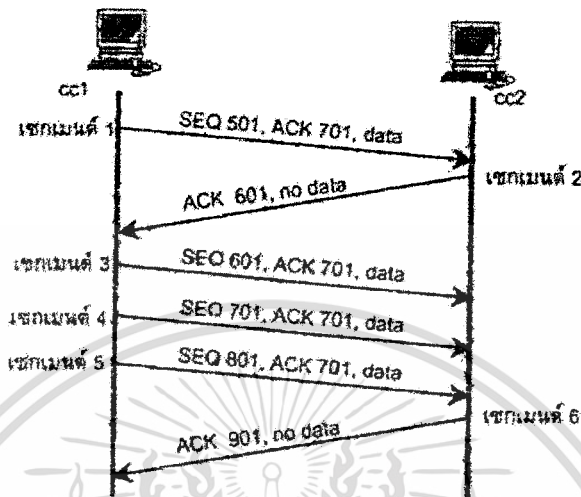
1. เครื่องต้นทางทำการส่ง เซกเมนต์ ที่มี SYN Flag เป็น 1 ระบุหมายเลขพอร์ตที่ต้องการติดต่อบนเครื่องปลายทางและระบุหมายเลขลำดับของข้อมูล (ISN : Initial Sequence Number)
2. เครื่องปลายทางเมื่อได้รับเซกเมนต์ตามข้อที่ 1 ก็จะตอบกลับด้วยการเพิ่มค่า ISN ที่ได้รับขึ้นอีก 1 พร้อมทั้งระบุหมายเลขลำดับ (ISN) ของตนเองและทำการเปิด Flags SYN กับ ACK
3. เครื่องต้นทางเมื่อได้รับการตอบกลับของข้อ 2 ก็จะมีการตอบรับไปอีกครั้ง โดยเพิ่มค่า ISN ของเครื่องปลายทางขึ้นอีก 1 และเปิดค่า ACK Flags

กระบวนการทั้งสามข้างต้นเรียกว่า 3-Ways Handshakes ซึ่งเมื่อผ่านกระบวนการนี้แล้ว จะเปรียบเสมือนว่า เครื่องต้นทางและเครื่องปลายทางได้ทำการเชื่อมต่อกันเรียบร้อยแล้ว

2.5.2 การถ่ายโอนข้อมูล

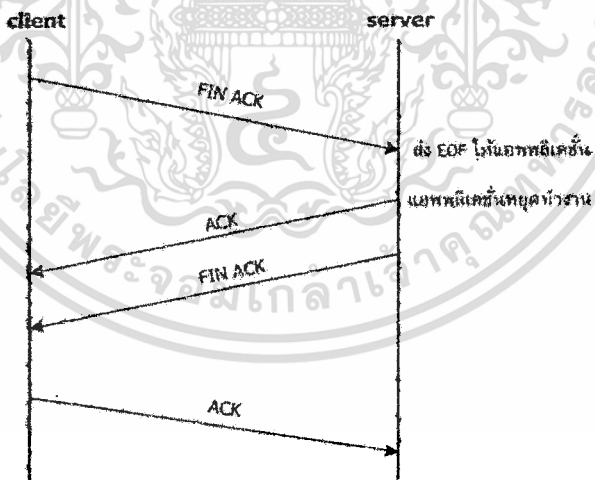
การขนถ่ายข้อมูลจะเริ่มขึ้นหลังจากการสถาปนาการเชื่อมต่อสำเร็จ เซกเมนต์แรกที่ เครื่องต้นทางทำการส่งไปยังปลายทาง บรรจุข้อมูลไบต์ 501 ถึง 600 และใช้เลขตอบรับ 701 เมื่อเครื่องปลายทางรับข้อมูลเรียบร้อยแล้ว ก็จะตอบกลับด้วยเซกเมนต์ตอบรับ ที่มีเลขตอบรับที่กำหนดข้อมูลที่คาดว่าจะรับต่อไป ซึ่งคือ 601

เครื่องต้นทางสามารถส่ง ข้อมูลไปได้อย่างต่อเนื่อง ตามภาพ คือ 601ม 701 และ 801 เครื่องปลายทางสามารถตอบกลับมาในคราวเดียวกัน



รูปที่ 2.10 ขั้นตอนการถ่ายโอนข้อมูล

2.5.3 การจบการเชื่อมต่อ



รูปที่ 2.11 ขั้นตอนการยุติการเชื่อมต่อ

หลังจากการรับส่งข้อมูลได้ยุติลง จะต้องทำขั้นตอนยุติการรับ-ส่งข้อมูล เปรียบเสมือนการถอดสายเชื่อมที่ต่ออยู่ระหว่างเครื่องต้นทางและเครื่องปลายทาง การสิ้นสุดการเชื่อมต่อโดยสมบูรณ์มี 4 ขั้นตอน ดังนี้

1. เครื่องต้นทางทำการส่ง ISN ที่มี Flags FIN ACK ที่เป็น 1 ไปยังปลายทาง
2. เครื่องปลายทาง ทำการตอบรับ ISN และบวกค่า ISN อีก 1 พร้อมกับ ACK เป็น 1
3. เครื่องต้นทางทำการส่ง ISN พร้อมกับ FIN ACK Flags ไปยังต้นทาง
4. เครื่องต้นทางตอบรับการยุติการเชื่อมต่อ ด้วย ISN +1 พร้อมกับ ACK Flags



บทที่ 3

การวิเคราะห์และออกแบบระบบ

ในบทนี้ จะกล่าวถึงการวิเคราะห์และออกแบบระบบดักจับจดหมายอิเล็กทรอนิกส์ โดยจะเป็นการกล่าวถึงภาพรวมของการออกแบบ ที่ประกอบด้วยการศึกษาความต้องการของระบบและนำข้อมูลที่ได้มาทำการวิเคราะห์ เพื่อสร้างแนวทางในการออกแบบระบบให้สอดคล้องกับความต้องการ โดยจะนำเสนออยู่ในรูปแบบของ ไคอะแกรมแบบต่างๆ

3.1 การศึกษาความต้องการของระบบ

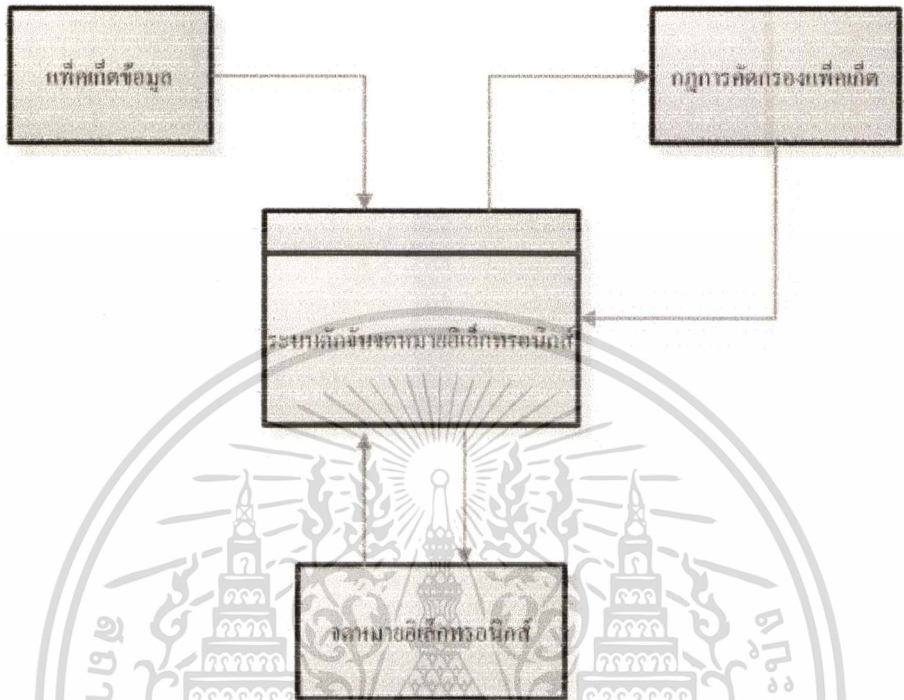
จากการกำหนดวัตถุประสงค์ และขอบเขตในการพัฒนาระบบ จึงสามารถกำหนดความต้องการเพื่อให้ระบบที่พัฒนาสามารถทำงานเป็นไปตาม วัตถุประสงค์ที่กำหนดดังต่อไปนี้ได้

1. ระบบที่พัฒนาต้องสามารถทำการดักจับข้อมูลบนเครือข่ายอินเทอร์เน็ตได้และข้อมูลที่ทำการดักจับได้ ต้องสามารถกรองและนำข้อมูลที่เกี่ยวข้องกับจดหมายอิเล็กทรอนิกส์มาใช้งานเท่านั้น
2. ข้อมูลที่ผ่านการกรองแล้ว ระบบต้องสามารถทำการแยกแยะความหมายส่วนประกอบ ต่างๆของข้อมูลเพื่อทำการ จัดเก็บลงฐานข้อมูลได้
3. ระบบสามารถทำการประกอบข้อมูลที่สัมพันธ์กัน เพื่อแปลให้กลับมาอยู่ในรูปแบบของจดหมายอิเล็กทรอนิกส์ได้

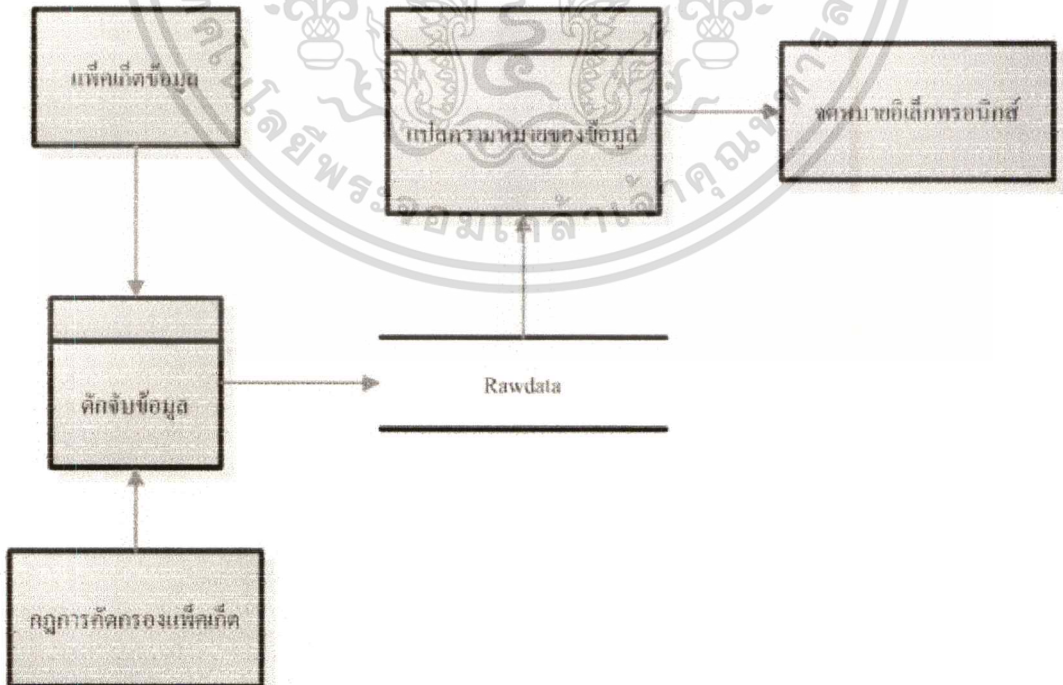
3.2 การออกแบบโครงสร้างการทำงาน

เมื่อได้ทำการศึกษาถึงความต้องการของระบบแล้ว ขั้นตอนต่อไป คือการออกแบบขั้นตอนการทำงานของระบบเพื่อให้สามารถรองรับ ความต้องการได้ จึงเขียนเป็นผังแสดงขั้นตอนการทำงานดังนี้

3.2.1 ขั้นตอนการทำงานโดยรวมของระบบทั้งหมด



รูปที่ 3.1 Context Diagram ของระบบดักจับจดหมายอิเล็กทรอนิกส์



รูปที่ 3.2 DFD 0 ของระบบดักจับจดหมายอิเล็กทรอนิกส์

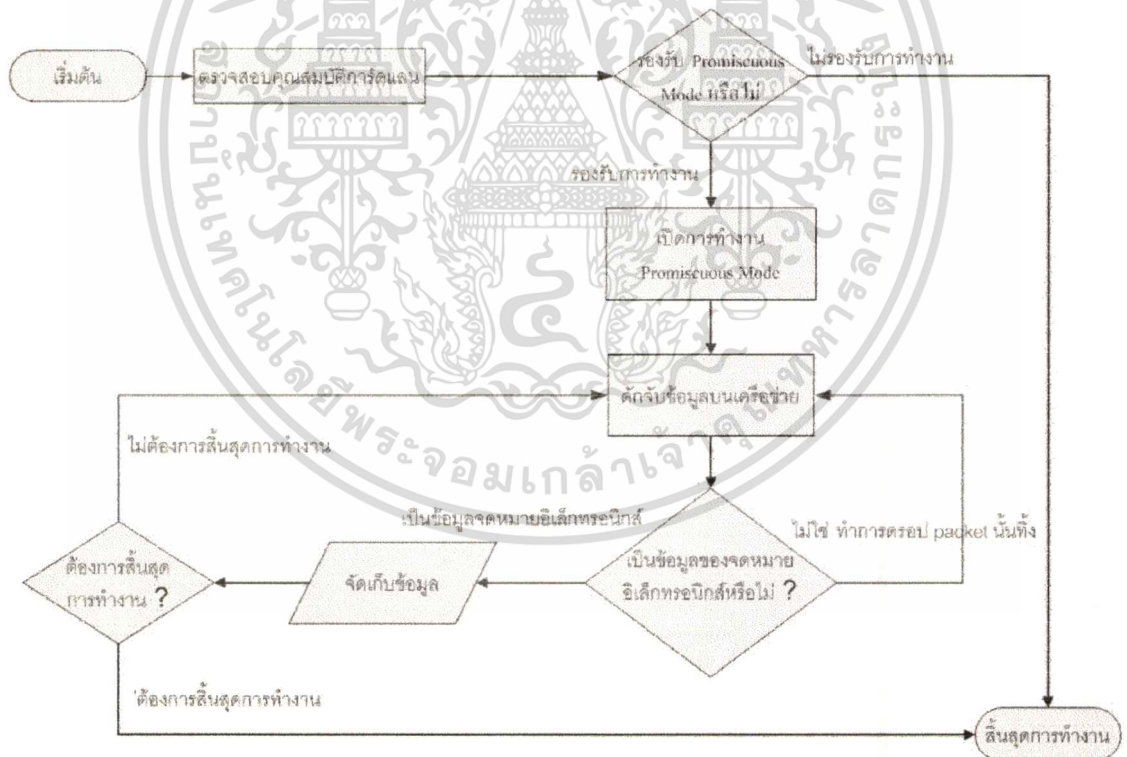
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ขั้นตอนการทำงานของส่วนดักจับข้อมูล

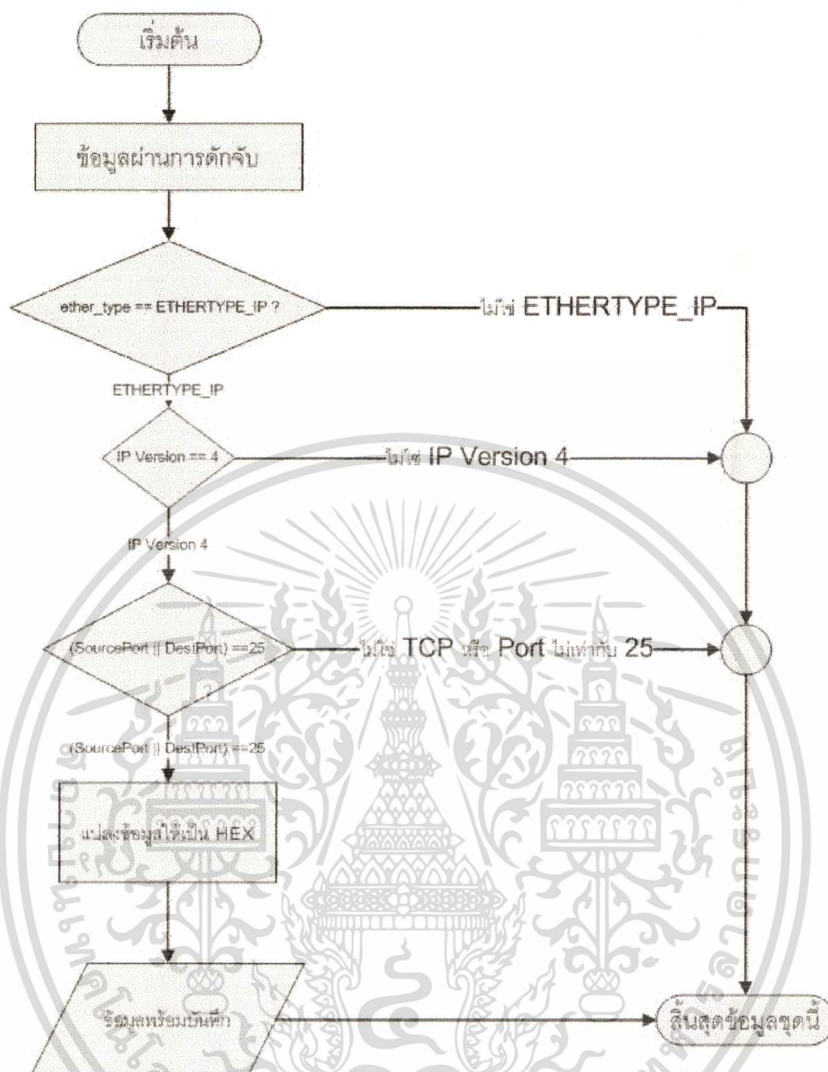
ขั้นตอนที่ 1 เมื่อระบบเริ่มทำงาน ระบบจะทำการตรวจสอบ การ์ดเครือข่าย ว่ามีอยู่หรือรองรับการทำงานใน Promiscuous Mode หรือไม่ ซึ่งในขั้นตอนนี้ หาก การ์ดเครือข่ายที่ใช้งานอยู่ไม่สามารถรองรับได้ โปรแกรมก็จะไม่สามารถทำงานได้

ขั้นตอนที่ 2 เปิดการทำงานการ์ดเครือข่าย ให้อยู่ใน Promiscuous Mode เพื่อให้เครื่องสนใจข้อมูลทุกอย่างที่ ส่งบนสายสัญญาณ

ขั้นตอนที่ 3 พิจารณาข้อมูลที่วิ่งอยู่บนเครือข่าย โดยจะพิจารณาเฉพาะ ข้อมูลที่เกี่ยวข้องกับจดหมายอิเล็กทรอนิกส์ เท่านั้น โดยทำการบันทึกลงในฐานข้อมูล เพื่อให้่ายต่อการใช้งาน เมื่อถึงขั้นตอนการแปลงข้อมูล ส่วนข้อมูลที่ไม่เกี่ยวข้องกับจดหมายอิเล็กทรอนิกส์ โปรแกรมจะไม่สนใจข้อมูลนั้น โดยโปรแกรมจะมีการทำงานแบบวนรอบ จนกว่าจะมีการยกเลิกการทำงานโดยผู้ใช้ หรือเกิดความผิดพลาดขึ้น



รูปที่ 3.3 แสดงขั้นตอนการทำงานโดยรวมในส่วนของการดักจับจดหมายอิเล็กทรอนิกส์



รูปที่ 3.4 แสดงขั้นตอนการคัดกรองแพ็กเก็ตที่ต้องการ

จากรูป เมื่อมีดักจับแพ็กเก็ตบนเครือข่ายได้ โปรแกรมจะทำการกรองแพ็กเก็ตต่างๆ โดยอาศัยทฤษฎีการส่งข้อมูลบนอีเทอร์เน็ต นั่นคือแพ็กเก็ตข้อมูลจะถูกห่อหุ้มด้วยเฮดเดอร์ต่างๆ ตามลำดับชั้นที่แพ็กเก็ตนั้นๆวิ่งผ่าน ซึ่งมีหลักเกณฑ์ดังนี้

- แพ็กเก็ต ต้องเป็นข้อมูลประเภทอีเทอร์เน็ตหากไม่ใช่จะทำการครอบแพ็กเก็ตนั้นทิ้งไป
- หากข้อมูลเป็นประเภทอีเทอร์เน็ต ก็จะทำการตรวจรายละเอียดภายในว่า ข้อมูลที่บรรจุมาด้วยนั้น เป็นข้อมูลประเภท IP หรือไม่ (เนื่องจาก SMTP นั้นรับส่งโดยใช้โปรโตคอล TCP/IP) หากรายละเอียดภายในไม่ใช่ IP ก็จะทำการครอบแพ็กเก็ตนั้นทิ้ง
- หากเป็นข้อมูลประเภท IP ก็จะทำการตรวจหมายเลขรุ่นของ IP ว่าเป็นรุ่น 4 หรือไม่ เนื่องจากรูปแบบของ IP รุ่นอื่น เช่น รุ่น 6 มีรูปแบบภายในแตกต่างกันทำให้ไม่สามารถ

เอกสารนี้เป็นเอกสารเข้าใจความหมายได้ หากเป็น IP รุ่นที่ 4 ก็จะทำการตรวจเช็ค ในส่วนของ TCP ต่อไป
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

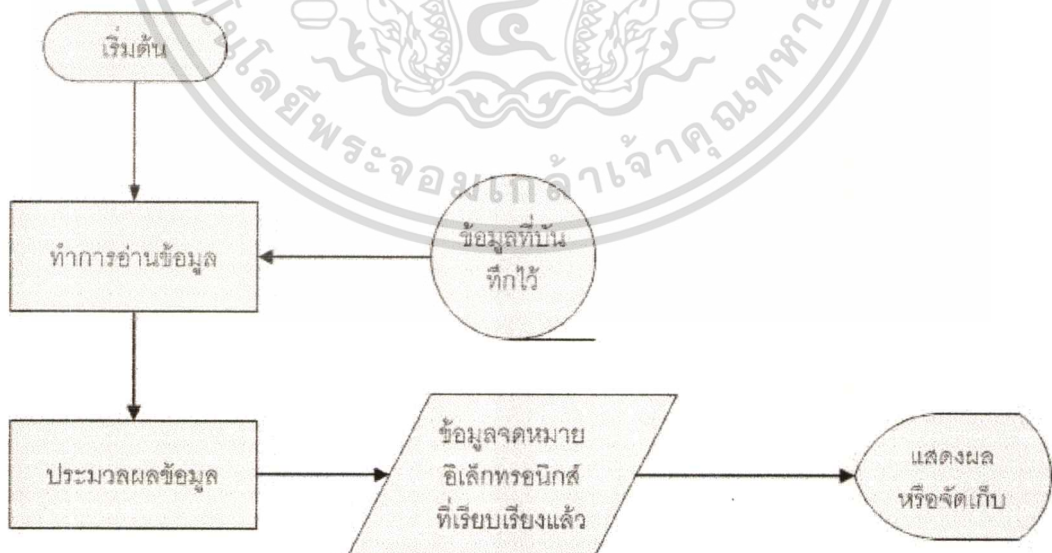
- ในส่วนของ TCP จะทำการตรวจหมายเลขพอร์ต ต้นทางหรือปลายทาง ว่า มีหมายเลขใดตรงกับหมายเลขของ SMTP หรือไม่ (เนื่องจาก SMTP ใช้ พอร์ตหมายเลข 25 เป็นมาตรฐาน) หากทางใดทางหนึ่งมีหมายเลข เป็น 25 จะมีความหมายว่า เป็นแพ็คเก็ตที่ติดต่อกับ SMTP จึงสามารถนำแพ็คเก็ตนั้นมาพิจารณา ขั้นต่อไปแพ็คเก็ตที่ได้ในส่วนนี้ จะเป็นแพ็คเก็ตที่เกี่ยวข้องกับ SMTP ทั้งสิ้น กระบวนการต่อไป คือ กระบวนการบันทึกข้อมูลแพ็คเก็ตนั้นๆ โดยข้อมูลที่จะทำการบันทึกนั้น ควรแปลงให้อยู่ในรูปฐาน 16 เพื่อง่ายต่อการเก็บและนำมาใช้งาน

3.2.3 ขั้นตอนการทำงานของส่วนแปลงข้อมูล

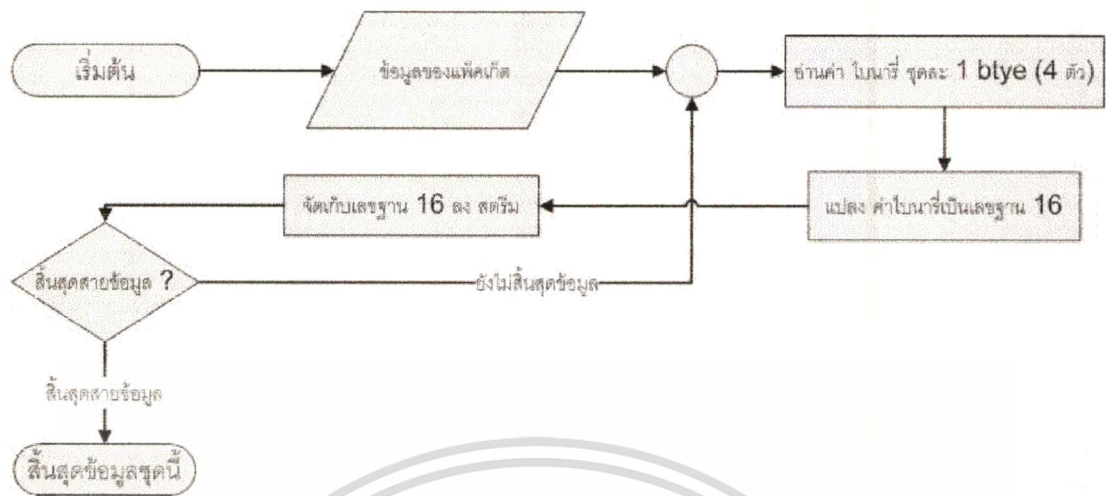
ขั้นตอนที่ 1 ทำการอ่านข้อมูลจากฐานข้อมูล โดย ในขั้นตอนการเลือกนี้ จะทำการเลือกข้อมูลที่เกี่ยวข้องสัมพันธ์กันหรือ เป็นข้อมูลชุดเดียวกัน

ขั้นตอนที่ 2 ทำการแปลงข้อมูลที่ดึงมาจากฐานข้อมูลในรูปแบบข้อมูลดิบ ให้ออกมาอยู่ในรูปของจดหมายอิเล็กทรอนิกส์ที่สามารถอ่านเข้าใจได้

ขั้นตอนที่ 3 ทำการจัดเก็บจดหมายอิเล็กทรอนิกส์แต่ละฉบับที่ทำการประมวลผลเรียบร้อยแล้วลงสู่ไฟล์หรือแสดงผล



รูปที่ 3.5 แสดงขั้นตอนการทำงานโดยรวมในส่วนของการแปลงข้อมูล



รูปที่ 3.6 แสดงขั้นตอนการแปลงข้อมูล

ในกระบวนการจัดเก็บข้อมูลนั้น ต้องมีการแปลงรูปแบบของแพ็คเก็ตจาก ข้อมูลไบนารี ให้มาอยู่ในรูปแบบฐาน 16 (Hex) ซึ่งจะมีการแปลงข้อมูลเฉพาะในส่วน payload ซึ่งก็คือ ข้อมูลที่แพ็คเก็ตนั้นๆบรรจุมา หลักการในการแปลงนั้นคือ จะมีการแบ่งค่าไบนารีออกเป็นชุดต่างๆ ชุดละ 4 ตัว จากนั้นจึงทำการแปลงเลขไบนารี โดยเทียบค่ากับเลขฐาน 16 เช่น 1000_2 สามารถแปลงเป็น 8_{16} จากนั้นจึงทำการบันทึกข้อมูล

3.2.4 การออกแบบในส่วนของการจัดเก็บข้อมูล

ในการออกแบบในส่วนนี้ เป็นการออกแบบการจัดเก็บข้อมูล โดยข้อมูลที่จัดเก็บนั้น จะเก็บเฉพาะข้อมูลที่สำคัญ ที่ใช้ในการแปลความหมายของแพ็คเก็ตให้สามารถกลับมาอยู่ในรูปที่อ่านเข้าใจได้ โดยข้อดีของการใช้ฐานข้อมูลนั้นคือ สามารถทำงานได้รวดเร็ว ง่ายต่อการจัดรูปแบบการจัดเก็บ รวมถึง ง่ายต่อการเลือกใช้ข้อมูล

ตารางที่ 3.1 แสดง Table ของข้อมูลที่เกี่ยวข้องในระบบงาน

Attribute	Description	Type	Length	Key
no	หมายเลขลำดับของแพ็คเก็ต	Int	11	PK
IPSource	IP Address ของต้นทาง	text	-	
IPDest	IP Address ของปลายทาง	text	-	
ACK	Acknowledgement Number	text	-	
SEQ	Sequence Number	text	-	
payload_length	ความยาวของข้อมูล	int	10	

ตารางที่ 3.1 แสดง Table ของข้อมูลที่เกี่ยวข้องในระบบงาน (ต่อ)				
raw	ส่วนของข้อมูล	longtext	-	
SourcePort	หมายเลขพอร์ตต้นทางในการติดต่อ	text	-	
DestPort	หมายเลขพอร์ตปลายทางในการติดต่อ	text	-	
time	เวลาเมื่อแพ็คเก็ตนั้นมาถึง	text		

3.3 การออกแบบในส่วนของการจับข้อมูล

ในการจับแพ็คเก็ตข้อมูล นั้นได้มีการใช้ libpcap หรือ library for packet capture คือ library ที่ใช้ในการ sniff packet ในระบบ UNIX ซึ่งแพ็คเก็ตที่ได้ในการจับข้อมูลนั้นมีโครงสร้างดังนี้

3.3.1 โครงสร้างในการเก็บข้อมูลของ PCAP Library

```
struct pcap_pkthdr {
    struct timeval ts;
    bpf_u_int32 caplen;
    bpf_u_int32 len;
};
```

รูปที่ 3.7 แสดงถึง โครงสร้างในการเก็บข้อมูลของ PCAP Library

โดยที่ สามารถอธิบาย ส่วนต่างๆของ โครงสร้าง ได้ดังนี้

- struct timeval ts เป็นโครงสร้างตัวแปรในการเก็บค่า ของเวลา โดยเวลาที่เก็บนั้นจะเป็นเวลาที่ แพ็คเก็ต เกิด มาถึงที่การ์ดแลน
- caplen เป็นตัวแปรในการเก็บขนาดของส่วนที่ต้องการแสดง
- len เป็นตัวแปรในการเก็บขนาดของแพ็คเก็ตข้อมูล

3.3.2 โครงสร้างในการเก็บข้อมูลของ Ethernet

```
Struct ether_header
{
    u_int8_t ether_dhost[ETH_ALEN];
    u_int8_t ether_shost[ETH_ALEN];
    u_int16_t ether_type;
} __attribute__((packed));
```

รูปที่ 3.8 แสดงถึงโครงสร้างในการเก็บข้อมูลของ Ethernet

โดยที่ สามารถอธิบาย ส่วนต่างๆของ โครงสร้างได้ดังนี้

- ether_dhost เป็นตัวแปรที่ใช้ในการเก็บค่า MAC Address ของเครื่องปลายทาง
- ether_shost เป็นตัวแปรที่ใช้ในการเก็บค่า MAC Address ของเครื่องต้นทาง
- ether_type เป็นตัวแปรที่ใช้ในการเก็บค่า ชนิดของ โปรโตคอลที่ใช้ในการส่งข้อมูล
เช่น IP มีค่า 0x0800 ARP มีค่า 0x0806 เป็นต้น

3.3.3 โครงสร้างในการเก็บข้อมูลของ IP

```
struct my_ip{
    u_int8_t ip_vhl;
#define IP_V(ip)((ip->ip_vhl & 0xf0) >>4)
#define IP_HL(ip) ((ip->ip_vhl & 0x0f)
    u_int8_t ip_tos;
    u_int16_t ip_len;
    u_int16_t ip_id;
    u_int16_t ip_off;
#define IP_DF 0x4000
#define IP_MF 0x2000
#define IP_OFFMASK 0x1fff
```

```

u_int8_t ip_ttl;

u_int8_t ip_p;

u_int16_t ip_sum;

struct in_addr ip_src;

        struct in_addr ip_dst;

};

```

รูปที่ 3.9 แสดงถึงโครงสร้างในการเก็บข้อมูลของ IP

โดยที่ สามารถอธิบาย ส่วนต่างๆที่สำคัญของโครงสร้างได้ดังนี้

- #define IP_V(ip) (((ip)->ip_vhl & 0xf0) >> 4) เป็นฟังก์ชันเพื่อหาค่า version ของ IP โดยได้ค่ามาจากการแปลงค่าของ ip_vhl ซึ่งเป็นตัวแปรในการเก็บขนาดความยาวของเฮดเดอร์
- #define IP_HL(ip) ((ip)->ip_vhl & 0x0f) เป็นฟังก์ชันเพื่อหาค่า ความยาวของเฮดเดอร์
- ip_len เป็นตัวแปรที่ใช้ในการเก็บความยาวของข้อมูลในส่วน payload ของ IP
- ip_off เป็นตัวแปรที่ใช้ในการเก็บค่า offset ของ IP
- ip_ttl เป็นตัวแปรที่ใช้ในการเก็บค่า TTL (Time To Live) ซึ่งใช้ในการบ่งบอกอายุของแพ็คเก็ต
- ip_p เป็นตัวแปรที่ใช้ในการเก็บค่าโปรโตคอล ที่ใช้ใน TCP เช่น TCP, UDP
- in_addr ip_src เป็นตัวแปร โครงสร้างที่ใช้ในการเก็บค่า IP Address ของเครื่องต้นทาง
- in_addr ip_dst เป็นตัวแปร โครงสร้างที่ใช้ในการเก็บค่า IP Address ของเครื่องปลายทาง

3.3.4 โครงสร้างในการเก็บข้อมูลของ TCP

```

struct tcphdr {

        u_int16_t      th_sport;

        u_int16_t      th_dport;

        tcp_seq        th_seq;

        tcp_seq        th_ack;

```

```

        u_int8_t th_offx2;
#define TH_OFF(th)      (((th)->th_offx2 & 0xf0) >> 4)
        u_int8_t 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_ECN_ECHO     0x40
#define TH_CWR          0x80
        u_int16_t      th_win;
        u_int16_t      th_sum;
        u_int16_t      th_urp;
};

```

รูปที่ 3.10 แสดงถึงโครงสร้างในการเก็บข้อมูลของ TCP

โดยที่ สามารถอธิบาย ส่วนต่างๆที่สำคัญของโครงสร้าง ได้ดังนี้

- th_sport เป็นตัวแปรที่ใช้ในการเก็บค่าของ พอร์ตต้นทาง
- th_dport เป็นตัวแปรที่ใช้ในการเก็บค่าของ พอร์ตปลายทาง
- th_seq เป็นตัวแปรที่ใช้ในการเก็บค่าของ Sequence Number ซึ่งมีความจำเป็นในการรับส่งและเรียงลำดับของข้อมูล
- th_ack เป็นตัวแปรที่ใช้ในการเก็บค่าของ Acknowledgement Number ซึ่งมีความจำเป็นในการรับส่งและเรียงลำดับของข้อมูล
- #define TH_OFF(th) (((th)->th_offx2 & 0xf0) >> 4) เป็นฟังก์ชันที่ใช้ในการแปลงค่าเพื่อหาขนาดของ TCP
- th_flags เป็นตัวแปรที่ใช้ในการเก็บค่าของแฟล็กต่างๆของ TCP เช่น SYS,FIN เป็นต้น

บทที่ 4

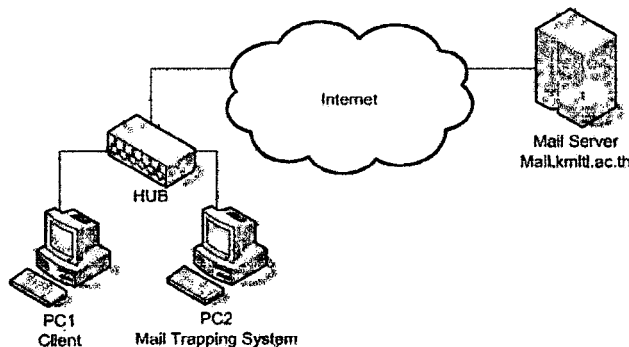
การพัฒนาระบบงานและการทดลอง

ในบทนี้จะกล่าวถึงขั้นตอนต่างๆในการพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ ซึ่งประกอบได้ด้วยขั้นตอนย่อยหลายๆขั้นตอน เช่นขั้นตอนในการกรองเฉพาะข้อมูลที่ต้องการ ขั้นตอนในการเรียบเรียงข้อมูล วิธีการทำงานในแต่ละขั้นตอน จากนั้นจะกล่าวถึง การทดลองการใช้งาน เพื่อแสดงผลลัพธ์ที่ได้จากการใช้โปรแกรม เพื่อเป็นแนวทางปรับปรุงและพัฒนาต่อไป

4.1 เครื่องมือที่ใช้ในการพัฒนาระบบงาน

- Aujata-IDE ใช้สำหรับสร้างและแก้ไขโปรแกรม
- gcc ใช้สำหรับ compile โปรแกรมที่เขียนด้วยภาษาซี
- libpcap-0.7.2 เป็นไลบรารี ที่ใช้สำหรับการดักจับข้อมูลบนเครือข่าย
- MySQL-4.2.1 ใช้สำหรับเก็บข้อมูลที่ดักจับได้
- Webmin ใช้สำหรับการปรับค่าต่างๆ ของ Server
- Microsoft Word 2000 ใช้สำหรับการสร้างเอกสาร
- เครื่องที่ใช้ในการพัฒนาระบบ ประกอบด้วย CPU AMD-1000MHz , หน่วยความจำ 768 MB, ฮาร์ดดิสก์ 40 GB การ์ดเครือข่าย SMC 1211TX (10/100 Mbps) บนระบบปฏิบัติการ Linux TLE 5.0

4.2 รูปแบบการเชื่อมต่อเครือข่ายระหว่างการพัฒนาและที่ใช้ในการทดลอง



เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ในการอื่นโดยไม่ได้รับอนุญาต
รูปที่ 4.1 แสดงถึงการเชื่อมต่อเครือข่ายระหว่างการพัฒนาและที่ใช้ในการทดลอง

4.3 การพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์

ระบบดักจับจดหมายอิเล็กทรอนิกส์ มีการออกแบบแยกเป็นสองโมดูลการทำงาน คือส่วนแรก เป็นส่วนที่มีหน้าที่ในการดักจับแพ็คเก็ตต่างๆ ที่รับ-ส่งบนเครือข่ายอีเทอร์เน็ตที่ ระบบทำการเชื่อมต่ออยู่โดยภายในจะประกอบไปด้วย ส่วนของการคัดกรองแพ็คเก็ตที่ต้องการ ส่วนที่สองคือส่วนที่มีหน้าที่ในการรวบรวมแพ็คเก็ตที่เกี่ยวข้องที่ได้มาจาก โมดูลแรก มาทำการเรียบเรียงให้กลายเป็นจดหมายอิเล็กทรอนิกส์

4.3.1 การพัฒนาในส่วนของการดักจับและคัดกรองแพ็คเก็ต

ระบบในส่วนนี้ จะประกอบได้ด้วยกระบวนการย่อยๆ หลายๆ ส่วนทำงานประกอบกัน กระบวนการเหล่านั้นคือ กระบวนการดักจับแพ็คเก็ต กระบวนการคัดกรองแพ็คเก็ต กระบวนการจัดเก็บข้อมูล

- กระบวนการดักจับแพ็คเก็ต ในการพัฒนาในส่วนนี้ จะเริ่ม ตั้งแต่การตรวจสอบคุณสมบัติของการ์ดแลน ว่ารองรับ Promiscuous Mode หรือไม่โดยใช้ชุดคำสั่งด้านล่าง

```
dev = pcap_lookupdev(errbuf); <----- ทำการตรวจสอบการ์ดแลนที่มี
if(dev == NULL)
{ printf("%s\n",errbuf); }
descr = pcap_open_live(dev,BUFSIZ,-1,-1,errbuf); <----- ทำการเปิดคุณสมบัติ Promiscuous
if(descr == NULL)
{printf("pcap_open_live(): %s\n",errbuf); exit(0);}
```

รูปที่ 4.2 แสดงถึงชุดคำสั่งที่ใช้ในการตรวจสอบคุณสมบัติของการ์ดแลน

หากตรวจพบว่าไม่รองรับ โปรแกรมจะไม่สามารถทำงานต่อได้ หากตรวจพบว่ารองรับ โปรแกรมจะทำการเปิดคุณสมบัติ Promiscuous Mode และเริ่มทำการดักจับข้อมูลที่วิ่งผ่านสาย สัญญาณ

- กระบวนการคัดกรองแพ็คเก็ต ข้อมูลที่สามารถดักจับผ่านสายสัญญาณได้นั้น ประกอบไปด้วยข้อมูลที่หลากหลาย ซึ่งข้อมูลที่ระบบนี้สนใจจะเป็นข้อมูลที่เกี่ยวข้องกับ SMTP นั่นคือข้อมูลที่ใช่โปรโตคอล TCP ในการรับส่ง และใช้พอร์ตหมายเลข 25 ซึ่งก่อนหน้าที่จะทำการตรวจสอบโปรโตคอล TCP จะต้องทำการตรวจสอบโปรโตคอลอื่นๆ ตามลำดับขั้นก่อนหน้านั้นคือ อีเทอร์เน็ต และ ไอพี

```

eptr = (struct ether_header *)packet;

if(ntohs(eptr->ether_type)==ETHERTYPE_IP)
{
    ip = (struct my_ip *) (packet + sizeof(struct ether_header));
    version = IP_V(ip);
    if(version==4) <----- ตรวจสอบ IP Version 4
    {
        addr = ip->ip_src;
        IPtemp = inet_ntoa(addr);
        SourceIP = (char *)malloc(strlen(IPtemp)+1);
        strcpy(SourceIP,IPtemp);

        addr = ip->ip_dst;
        IPtemp = inet_ntoa(addr);
        DestIP = (char *)malloc(strlen(IPtemp)+1);
        strcpy(DestIP,IPtemp);

        protocol = ntohs(ip->ip_p);
        ip_length = ntohs(ip->ip_len);

        tcp = (struct tcphdr *) (packet + sizeof(struct ether_header)+ sizeof(struct my_ip));
        tcp_length = TH_OFF(tcp) * 4;
        SourcePort = ntohs(tcp->th_sport);
        DestPort = ntohs(tcp->th_dport);

        if( (DestPort == Port) || (SourcePort == Port) ) <----- ตรวจสอบหมายเลขพอร์ต
        {

```

รูปที่ 4.3 แสดงถึงชุดคำสั่งที่ใช้ในการคัดกรองแพ็คเก็ตต่างๆ

- กระบวนการจัดเก็บข้อมูล หลังจากผ่านการคัดกรองแพ็คเก็ต แพ็คเก็ตที่ได้จะเป็นแพ็คเก็ตที่เกี่ยวข้องกับจดหมายอิเล็กทรอนิกส์ กระบวนการถัดไปคือการจัดเก็บข้อมูลที่ผ่านการคัดกรองแล้ว เพื่อที่จะสามารถนำไปใช้งานในโมดูลถัดไปซึ่งคือการแปลความหมายของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การพัฒนาในส่วนของการเรียบเรียงและแปลข้อมูล

การทำงานในส่วนของการเรียบเรียงและแปลข้อมูลนี้ เป็นการทำงานที่เป็นอิสระจาก จาก ส่วนของการดักจับ ข้อมูล เพราะ โปรแกรมในส่วนนี้ ทำหน้าที่ในการ นำข้อมูลดิบที่ถูกดักจับได้ใน ส่วนแรก มาทำการเรียบเรียงและแปลงให้อยู่ในรูปแบบที่สามารถอ่านเข้าใจได้และแสดงผล

- การดึงข้อมูลที่เกี่ยวข้องจากฐานข้อมูล ในการติดต่อสื่อสารผ่านทาง TCP นั้น หมายเลขพอร์ตจะเป็นสิ่งสำคัญในการระบุถึงโปรเซส ของการเชื่อมต่อนั้น เพื่อให้สามารถทราบได้ว่า แพ็คเก็ตนั้นๆจะมีการส่งไปยังโปรเซสใดๆ โดย ในการสร้างการเชื่อมต่อแบบ TCP เครื่องต้นทางจะส่งคำร้องขอเชื่อมต่อไปยังผู้ให้บริการโดยใช้ พอร์ตปลายทางที่เป็นเลขมาตรฐาน ส่วน พอร์ตต้นทางนั้นทางไคลเอนต์จะเป็นผู้สุ่มหมายเลขขึ้นมา เพื่อให้ผู้ให้บริการสามารถอ้างอิง ได้ ดังนั้นในกระบวนการแปลงแพ็คเก็ตต่างๆที่ได้จากการดักจับ จึงใช้ข้อมูลจากหมายเลข พอร์ตเพื่ออ้างอิงถึงโปรเซสของจดหมายฉบับต่างๆ โดยการพิจารณานี้ จะเป็นการพิจารณา ข้อมูลหมายเลขพอร์ตควบคู่ไปกับหมายเลข IP Address ของทั้งต้นทาง, ปลายทาง และเวลา

ตารางที่ 4.1 แสดงรายละเอียดแพ็คเก็ตที่สัมพันธ์กัน

IPSource	IPDest	SourcePort	DestPort	time
161.246.34.187	161.246.49.147	25	3651	04-02-04-20:20:42:437858
161.246.49.147	161.246.34.187	3651	25	04-02-04-20:20:42:439704
161.246.34.187	161.246.49.147	25	3651	04-02-04-20:20:42:440670
161.246.49.147	161.246.34.187	3651	25	04-02-04-20:20:42:444412
161.246.34.187	161.246.49.147	25	3651	04-02-04-20:20:42:499250
161.246.49.147	161.246.34.187	3651	25	04-02-04-20:20:42:500556
161.246.34.187	161.246.49.147	25	3651	04-02-04-20:20:42:567426
161.246.49.147	161.246.34.187	3651	25	04-02-04-20:20:42:570021
161.246.34.187	161.246.49.147	25	3651	04-02-04-20:20:42:573487
161.246.49.147	161.246.34.187	3651	25	04-02-04-20:20:42:580315
161.246.49.147	161.246.34.187	3651	25	04-02-04-20:20:42:581476
161.246.49.147	161.246.34.187	3651	25	04-02-04-20:20:42:582008
161.246.49.147	161.246.34.187	3651	25	04-02-04-20:20:42:679130
161.246.34.187	161.246.49.147	25	3651	04-02-04-20:20:42:718097

- การแปลงข้อมูลในรูป Hex Dump เป็น ASCII ข้อมูลในส่วนของ payload ที่จะเป็นเนื้อหาในส่วนต่างๆของจดหมายนั้น ได้ผ่านกระบวนการแปลงเป็น Hex Dump จากโปรแกรมในส่วนดักจับ ก่อนที่จะทำการบันทึกข้อมูลเพื่อความสะดวกในการเก็บ ดังนั้นในกระบวนการนี้ จะเป็นการนำ Hex Dump เหล่านั้นมาทำการแปลงรูปเป็นรหัส ASCII เพื่อให้สามารถอ่านเข้าใจได้ โดยใช้การเทียบรหัส Hex เข้ากับตาราง ASCII เพื่อให้ได้ตัวอักษรออกมา

4.4 การทดลองระบบดักจับจดหมายอิเล็กทรอนิกส์

ในการทดลองนี้ ใช้รูปแบบการเชื่อมต่อ ดังรูป 4.1 โดยทำการเปิดโปรแกรมดักจับข้อมูลที่ PC2 และทำการส่งจดหมาย โดยใช้ PC1

```

Source Port : 32853      Dest Port : 25
SEQ :2624674745 ACK :2332722961
IP length :612  TCP length :32
payload_length :560
*****
No : 8
IP Source : 161.246.49.160      IP Dest : 203.107.131.199
Source Port : 32853      Dest Port : 25
SEQ :2624675305 ACK :2332722961
IP length :55  TCP length :32
payload_length :3
*****
No : 9
IP Source : 203.107.131.199      IP Dest : 161.246.49.160
Source Port : 25      Dest Port : 32853
SEQ :2332722961 ACK :2624675308
IP length :80  TCP length :32
payload_length :28

```

รูปที่ 4.4 แสดงตัวอย่างโปรแกรมระหว่างการดักจับโดยไม่แสดง Payload

q	n	Source	Destination	Protocol	Info
3	1	203.107.131.199	161.246.49.160	SMTP	Response: 220 devmail.ksc.th.com ESMTP
3	1	161.246.49.160	203.107.131.199	SMTP	Command: EHLO karn.it.kmitl.ac.th
3	1	203.107.131.199	161.246.49.160	SMTP	Response: 250-devmail.ksc.th.com
1	1	161.246.49.160	203.107.131.199	SMTP	Command: MAIL From:<root@karn.it.kmitl.ac.th>
2	1	203.107.131.199	161.246.49.160	SMTP	Response: 250 ok
4	1	161.246.49.160	203.107.131.199	SMTP	Command: RCPT To:<malihom3@web.co.th>
3	1	203.107.131.199	161.246.49.160	SMTP	Response: 250 ok
7	1	161.246.49.160	203.107.131.199	SMTP	Message Body
3	1	161.246.49.160	203.107.131.199	SMTP	Message Body
1	1	203.107.131.199	161.246.49.160	SMTP	R----- 250 ok 1670440402 -- 250550
Transmission Control Protocol, Src Port: 32982 (32982), Dst Port: 25 (25), Seq: 126026388, Ack: 4					
Simple Mail Transfer Protocol					
Command: MAIL					
Request parameter: From:<root@karn.it.kmitl.ac.th>					
<pre> 0000 00 05 32 8e 66 02 00 e0 29 66 f8 35 08 00 45 00 ..2.f...)f.5..E. 0010 00 5a 9d c3 40 00 40 06 7a 11 a1 f6 31 a0 cb 6b .Z..@. z...1.k 0020 83 c7 80 d6 00 19 07 83 02 94 00 44 6b f5 80 18 Dk... 0030 16 d0 e8 ba 00 00 01 01 08 0a 00 01 9f ca 05 d7 0040 3a 21 4d 41 49 4c 20 46 72 6f 6d 3a 3c 72 6f 6f :!MAIL From:<roo 0050 74 40 6b 61 72 6e 2e 63 74 2e 6b 6d 63 74 6c 2e t@karn.it.kmitl. 0060 61 63 2e 74 68 3e 0d 0a ac.th>.. </pre>					

รูปที่ 4.5 แสดงตัวอย่างข้อมูลชุดเดียวกับเทียบกับโปรแกรม Ethereal

รูป 4.4 เป็นตัวอย่างการทดลองโปรแกรมดักจับข้อมูล โดยทำการแสดงผลเพียงรายละเอียดของแต่ละแพ็คเก็ตที่ดักจับได้ โดยไม่แสดงในส่วนของ payload สำหรับรูป 4.5 เป็นการเปรียบเทียบความถูกต้องของการดักจับ โดยทำการรันโปรแกรม Ethereal ควบคู่ไปด้วย เพื่อเปรียบเทียบข้อมูลที่ดักจับได้จากโปรแกรมทั้งสอง ว่าตรงกันหรือไม่ ซึ่งผลที่ได้พบว่า มีความตรงกัน ทุกส่วน

```

220 devmail.ksc.th.com ESMTP
EHLO karn.it.kmitl.ac.th
250-devmail.ksc.th.com
250-PIPELINING
250 8BITMIME
MAIL From:<root@karn.it.kmitl.ac.th>
250 ok
RCPT To:<mailihom3@mweb.co.th>
DATA
250 ok
354 go ahead
Received: from karn.it.kmitl.ac.th (localhost.localdomain [127.0.0.1])
    by karn.it.kmitl.ac.th (8.12.8/8.12.8) with ESMTP id i1ADJZVE005466
    for <mailihom3@mweb.co.th>; Tue, 10 Feb 2004 20:19:41 +0700
Received: (from root@localhost)
    by karn.it.kmitl.ac.th (8.12.8/8.12.8/Submit) id i1ADJTkL005463
    for mailihom3@mweb.co.th; Tue, 10 Feb 2004 20:19:29 +0700
Date: Tue, 10 Feb 2004 20:19:29 +0700
From: root <root@karn.it.kmitl.ac.th>
Message-Id: <200402101319.i1ADJTkL005463@karn.it.kmitl.ac.th>
To: mailihom3@mweb.co.th
Subject: Test mail for Project

This is a test mail for test Project
.
250 ok 1076419058 qp 30501
QUIT

```

รูปที่ 4.6 แสดงตัวอย่างข้อมูลที่ได้จากการแปลง

จากรูป 4.6 เป็นการแปลงข้อมูลในรูปแบบ Hex เป็น ASCII ซึ่งจะทำให้สามารถอ่านได้เข้าใจ โดยข้อมูลที่ได้เป็นข้อมูลการติดต่อระหว่าง Mail Server เพื่อทำการส่งจดหมายระหว่างกัน ซึ่งมีการทำงานแบบใช้รหัสแอสกีเอ็นวีที (NVT)

โดยในการกระบวนการเรียงเรียงและแปลงข้อมูลนี้ โปรแกรมจะเริ่มทำงานที่เรคคอร์ดแรกของฐานข้อมูล และทำงานจนสิ้นสุดฐานข้อมูล จดหมายอิเล็กทรอนิกส์แต่ละฉบับที่มีการข้อมูลเรียบร้อยแล้ว จะจัดเก็บอยู่ในรูปแบบไฟล์ ที่สามารถเปิดอ่านได้ โดย Text Editor

บทที่ 5

บทสรุปและข้อเสนอแนะ

เนื้อหาในบทนี้จะเป็นบทสรุปของการพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ พร้อมทั้งข้อเสนอแนะสำหรับผู้สนใจนำระบบงานไปทำการพัฒนาต่อไป

5.1 สรุปผลการทำงานและประโยชน์ที่ได้รับ

จากการพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ และทำการทดลองพบว่า โปรแกรมที่มีหน้าที่ในการดักจับจดหมายอิเล็กทรอนิกส์ สามารถทำการตรวจพบและดักจับแพ็คเกจข้อมูลที่มีความเกี่ยวข้องกับการรับส่งจดหมายอิเล็กทรอนิกส์ระหว่างผู้ให้บริการผ่านทางโปรโตคอล SMTP (Simple Mail Transfer Protocol) ได้ โดยจะทำการพักข้อมูลที่ดักจับได้นั้นลงบนฐานข้อมูล เพื่อความสะดวกแก่โปรแกรมในส่วนของการเรียงเรียงและแปลความหมาย

ส่วนของการเรียงเรียงและแปลความหมายนั้น จากการทดลองพบว่า สามารถทำการแปลงข้อมูลให้กลับมามีรูปแบบของ ASCII ที่สามารถอ่านเข้าใจได้ โดยตัวโปรแกรมอาศัยลักษณะเฉพาะตัวของการติดต่อกันแบบ รหัสแอสกีเอ็นวีที (NVT) นั่นคือ โคลเอ็นต์จะทำการส่งคำสั่งไปยังผู้ให้บริการ ผู้ให้บริการจะตอบกลับเป็นรหัสตัวเลข บอกถึงสถานะของผู้ให้บริการ ดังนั้น โคลเอ็นต์จึงไม่สามารถส่งข้อมูลมายังผู้ให้บริการได้หาก ยังไม่ทราบสถานะของผู้ให้บริการ ทำให้ข้อมูลที่หลุดผ่านทางโปรโตคอลนี้ มีการเรียงลำดับข้อมูลแล้ว

5.2 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ

ในการพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์นี้ มีปัญหาและอุปสรรคในการพัฒนาบางประการ อันส่งผลถึงคุณภาพและความสามารถของระบบ ดังนี้

- โปรแกรมทำการพัฒนาในระบบปฏิบัติการลินุกซ์ ซึ่งเป็นระบบปฏิบัติการที่มีความหลากหลายของโมดูลการทำงาน โดยบางโมดูลมีการพัฒนาออกมาเป็นหลายรุ่น ซึ่งก่อให้เกิดปัญหาความเข้ากันได้ ระหว่างการพัฒนา
- เครื่องมือในการพัฒนาโปรแกรมบนระบบปฏิบัติการลินุกซ์ มีน้อย และมีความสามารถน้อยกว่าบนระบบปฏิบัติการวินโดวส์

- เพื่อให้การพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์ มีความสามารถมากขึ้นและสมบูรณ์มากยิ่งขึ้น จึงควรมีการนำระบบไปทำการพัฒนาต่อ โดยมีข้อเสนอแนะดังต่อไปนี้
- ระบบในปัจจุบันมีข้อจำกัดในด้านประสิทธิภาพการทำงาน คือ ทำงานแยกกันระหว่างส่วนของการดักจับและส่วนของการแปลข้อมูล
- ควรมีการปรับปรุงอัลกอริทึมในส่วนของการแปลข้อมูลจดหมายให้มีความเป็นมาตรฐาน หรือรองรับรูปแบบที่หลากหลายมากยิ่งขึ้น
- ปรับปรุงในส่วนของการแสดงผลข้อมูลให้มีความสวยงาม และเข้าใจได้ง่ายยิ่งขึ้น
- นำระบบดักจับจดหมายอิเล็กทรอนิกส์ มาประยุกต์ใช้กับงานในส่วนอื่น เพื่อก่อให้เกิดประโยชน์มากยิ่งขึ้น เช่น การพัฒนาระบบการตรวจจับจดหมายไม่พึงประสงค์ เช่น สแปมเมล์ หรือ จดหมายขยะ โดยอาศัยตัวอย่างของจดหมายต่างๆ จากระบบดักจับจดหมายอิเล็กทรอนิกส์นี้
- ระบบดักจับจดหมายอิเล็กทรอนิกส์นี้ ทำหน้าที่ดักจับและคัดกรองแพ็คเกจที่เกี่ยวข้องมาทำการเรียบเรียงและแปลความหมาย แต่ระบบไม่ได้สนใจการทำงานในส่วนของ TCP และ SMTP ที่มีกระบวนการควบคุมการส่งผิดพลาด การส่งซ้ำ ทำให้ระบบที่พัฒนานี้ ข้อมูลที่ได้ลดความน่าเชื่อถือลงไป

บรรณานุกรม

กาน มะลิหอม. 2545. ระบบการรับส่งจดหมายอิเล็กทรอนิกส์. บทความสัมภาษณ์ 2.

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

กาน มะลิหอม. 2545. การพัฒนาระบบดักจับจดหมายอิเล็กทรอนิกส์. บทความสัมภาษณ์ 2.

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

สุรศักดิ์ สงวนพงษ์. 2545. สถาปัตยกรรมและโปรโตคอลที่ซีพี/ไอพี. กรุงเทพฯ : ซีเอ็ดดูเคชั่น.

Libpcap packet capture. 2547. **Packet Capture With libpcap and other Low Level Network**

Tricks. [Online]. Available : <http://www.cse.nau.edu/~mc8/Socket/Tutorials/section1.html>

Manpage of PCAP. 2547. **PCAP.** [Online]. Available: www.tcpdump.org/pcap3_man.html.

Stevens, W. Richard. 2533. **Unix Network Programming.** New Jersey : Prentice Hall.

Stevens, W. Richard. 2537. **TCP/IP illustrated.** Boston : Addison-Wesley.

Volkerding, Patrick. 2541. **LINUX configuration and installation.** CA : M&T Books



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งและใช้งาน ระบบดักจับจดหมายอิเล็กทรอนิกส์

1. การติดตั้งระบบระบบดักจับจดหมายอิเล็กทรอนิกส์

ระบบระบบดักจับจดหมายอิเล็กทรอนิกส์ประกอบด้วยการทำงานสองส่วน คือส่วนที่ทำหน้าที่ในการดักจับข้อมูลเพื่อคัดกรองข้อมูลที่เกี่ยวข้องกับจดหมายอิเล็กทรอนิกส์จัดเก็บลงฐานข้อมูล และส่วนที่สองคือส่วนที่ทำหน้าที่อ่านข้อมูลจากฐานข้อมูลเพื่อเรียบเรียงให้กลับเป็นจดหมายอิเล็กทรอนิกส์ซึ่งจำเป็นต้องมีการติดตั้ง โปรแกรมซึ่งมีขั้นตอนต่างๆดังนี้

- 1.1 ติดตั้ง libraly pcap ซึ่งเป็นไลบรารีที่ใช้ในการดักจับข้อมูลบนระบบปฏิบัติการลินุกซ์ ก่อนทำการติดตั้งผู้ใช้ต้องทำการ login เป็น root แล้วจึงใช้คำสั่ง ดังนี้ที่ shell prompt

```
$:// tar xvfz libpcap-0.7.2
```

- 1.2 ติดตั้ง mysql เป็นโปรแกรมจัดการฐานข้อมูล ซึ่งผู้ใช้ ใช้ในการพัฒนา mysql ในปัจจุบันมีด้วยกันหลายรุ่นและหลายโมดูล โมดูลที่จำเป็นประกอบไปด้วย

- MySQL-client-x.x.x.i386.rpm
- MySQL-devel-x.x.x.i386.rpm
- MySQL-server-x.x.x.i386.rpm

วิธีการติดตั้ง ใช้คำสั่งดังต่อไปนี้

```
$://rpm -ivh MySQL-client-x.x.x.i386.rpm
```

ในการใช้งาน mysql นี้จำเป็นต้องมี account ในการเข้าใช้งาน ซึ่งขึ้นอยู่กับข้อกำหนดของผู้ติดตั้ง

2. การแก้ไขและคอมไพล์โปรแกรม

หากต้องการเปลี่ยนแปลงแก้ไข ในส่วนของ Sourcecode ต้องมีการคอมไพล์ใหม่เสียก่อน ซึ่งกระบวนการในการคอมไพล์มีดังต่อไปนี้

- Configuration ก่อนทำการคอมไพล์ต้องมีการใช้คำสั่งนี้ เพื่อให้คอมไพเลอร์ทำการตรวจสอบ ตำแหน่งต่างๆที่ทำการติดตั้งไลบรารีไว้
- Make หลังจากทำการ configuration แล้ว คำสั่ง make คือคำสั่งที่ใช้ในการคอมไพล์ Source ของโปรแกรม โดยในที่นี้ คือการใช้ gcc ซึ่งเป็นคอมไพเลอร์ของภาษาซี โดยมีออปชั่นในการคอมไพล์คือ `-lpcap -L/usr/lib/mysql -lmysqlclient` ซึ่งเป็นการระบุมีการใช้ไลบรารีดังกล่าวในการคอมไพล์ หลังจากคำสั่งนี้ จะไปโปรแกรมที่สามารถทำงานได้

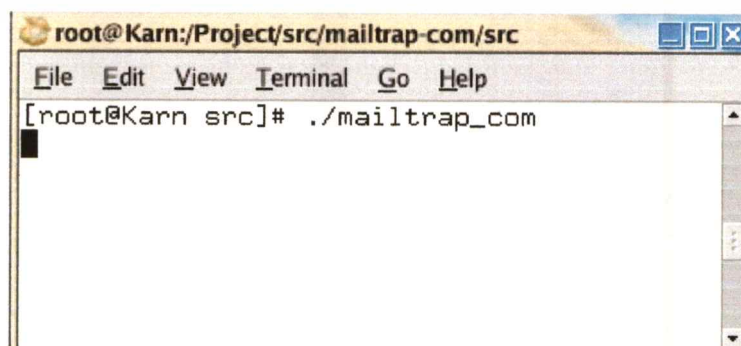
3. การใช้งานระบบระบบดักจับจดหมายอิเล็กทรอนิกส์

จากที่กล่าวข้างต้น ระบบดักจับจดหมายอิเล็กทรอนิกส์ประกอบไปด้การทำงานสองส่วน ซึ่งแต่ละส่วนมีการทำงานอิสระจากกัน ดังนั้น จึงต้องมีเรียกใช้งานแยกกัน ซึ่งก่อนหน้าที่จะทำการใช้งาน ต้องมีการ กำหนดถึง user password ของฐานข้อมูล mysql ให้ถูกต้องด้วย

- ส่วนดักจับข้อมูล เรียกใช้งานโดยใช้คำสั่ง

```
$: // ./mailtrap-com
```

ก่อนการใช้งานต้องมีการ login เป็น root เสียก่อน แล้วจึงเรียกใช้ตามคำสั่งด้านบนซึ่งหลังจากโปรแกรมเริ่มทำงาน โปรแกรมจะทำการรันเพื่อทำการพิจารณาข้อมูลต่างๆที่วิ่งบนเครือข่าย หากมีข้อมูลที่เกี่ยวข้องกับ smtp โปรแกรมจะรายงานผลออกมา และจัดเก็บข้อมูลลงในฐานข้อมูล



รูปที่ ก-1 แสดงถึงการเรียกใช้งาน ส่วนดักจับจดหมายอิเล็กทรอนิกส์

```

Source Port : 32853      Dest Port : 25
SEQ : 2624674745 ACK : 2332722961
IP length : 612  TCP length : 32
payload_length : 560
*****
No : 8
IP Source : 161.246.49.160      IP Dest : 203.107.131.199
Source Port : 32853      Dest Port : 25
SEQ : 2624675305 ACK : 2332722961
IP length : 55  TCP length : 32
payload_length : 3
*****
No : 9
IP Source : 203.107.131.199      IP Dest : 161.246.49.160
Source Port : 25      Dest Port : 32853
SEQ : 2332722961 ACK : 2624675308
IP length : 80  TCP length : 32
payload_length : 28

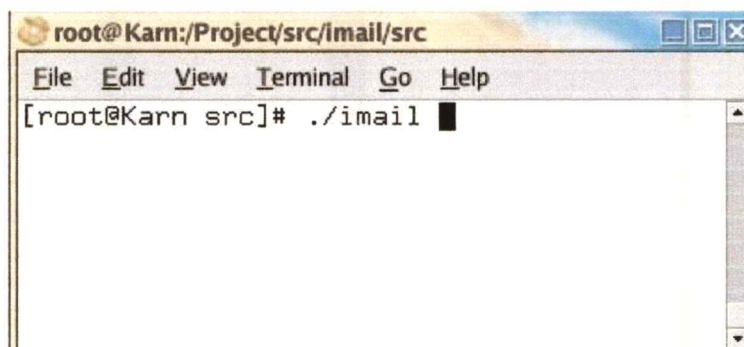
```

รูปที่ ก-2 แสดงตัวอย่าง โปรแกรมระหว่างการดักจับ

- ส่วนเรียงเรียงข้อมูล เมื่อมีการรวบรวมข้อมูลที่ดักจับได้ลงในฐานข้อมูล โปรแกรมส่วนนี้จัดทำหน้าที่ดึงข้อมูลที่สัมพันธ์กัน เพื่อนำไปทำการแปลให้ได้ข้อมูลของจดหมายอิเล็กทรอนิกส์ เรียกใช้งาน โดยใช้คำสั่ง

`$:// ./imail`

ก่อนการใช้งานต้องมีการ login เป็น root เสียก่อน แล้วจึงเรียกใช้ตามคำสั่งด้านบนซึ่งผลของการทำงาน จะได้ตัวอย่างของจดหมายอิเล็กทรอนิกส์ ซึ่งอยู่ในรูปของไฟล์ข้อความชื่อ “mail-data”



รูปที่ ก-3 แสดงถึงการเรียกใช้งาน แปลและเรียงเรียงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

root@Karn:/Project/src/imap/src
File Edit View Terminal Go Help
[root@Karn src]# ls
buff      imap-hex2dec.o      main.c      Makefile.in
hex2dec.c  imap-main.o         main.o      process_query.c
hex2dec.o  imap-process_query.o Makefile     process_query.o
imap       mail-data            Makefile.am rawdata
[root@Karn src]#

```

รูปที่ ก-4 แสดงถึงไฟล์จดหมายอิเล็กทรอนิกส์ที่ทำการจัดเก็บ

```

L | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17
*****
220 devmail.ksc.th.com ESMTP
EHLO kam.it.kmitl.ac.th
250-devmail.ksc.th.com
250-PIPELINING
250 8BITMIME
MAIL From:<root@kam.it.kmitl.ac.th>
250 ok
RCPT To:<malihom3@mweb.co.th>
DATA
250 ok
354 go ahead
Received: from kam.it.kmitl.ac.th (localhost.localdomain [127.0.0.1])
        by kam.it.kmitl.ac.th (8.12.8/8.12.8) with ESMTP id i1AG05Gg004648
        for <malihom3@mweb.co.th>; Tue, 10 Feb 2004 23:00:05 +0700
Received: (from root@localhost)
        by kam.it.kmitl.ac.th (8.12.8/8.12.8/Submit) id i1AG055t004646
        for malihom3@mweb.co.th; Tue, 10 Feb 2004 23:00:05 +0700
Date: Tue, 10 Feb 2004 23:00:05 +0700
From: root <root@kam.it.kmitl.ac.th>
Message-Id: <200402101600.i1AG055t004646@kam.it.kmitl.ac.th>
To: malihom3@mweb.co.th
Subject: Test

Test Test Test

250 ok 1076428531 qp 23312
QUIT
221 devmail.ksc.th.com
*****

```

รูปที่ ก-5 แสดงตัวอย่างจดหมายอิเล็กทรอนิกส์ด้วย Text Editor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายกาน มะลิหอม
วันเกิด	วันศุกร์ที่ 18 กรกฎาคม พ.ศ. 2523
สถานที่เกิด	เขตลาดกระบัง จ.กรุงเทพมหานคร
ประวัติการศึกษา	
ระดับประถมศึกษา	โรงเรียนมินบุรีศึกษา จ.กรุงเทพมหานคร
ระดับมัธยมศึกษา	โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า จ.กรุงเทพมหานคร
ระดับปริญญาตรี	สถาบันเทคโนโลยีราชมงคล วิทยาเขต คลองหก จ.ปทุมธานี

