ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

การพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้ส่งผ่านด้วยโปรโตคอล HTTP

SYSTEM DEVELOPMENT OF HTTP TRAFFIC SHAPING SYSTEM

H005979

โดย

บุญลือ จิตตคาม

BOONLUE JITTAKAM

อาจารย์ที่ปรึกษา

ผศ. อัครินทร์ คุณกิตติ

วพ.
บ561ก
2551

เลขหมู่.....
เลขทะเบียน.....05979
วัน,เดือน,ปี.....- 3 ก.พ. 2553

b.12178868.....
i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYSTEM DEVELOPMENT OF HTTP TRAFFIC SHAPING SYSTEM

BOONLUE JITTAKAM

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2009

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	การพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้ส่งผ่านด้วยโปรโตคอล HTTP
นักศึกษา	นายบุญลือ จิตตคาม
รหัสนักศึกษา	48066815
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2551
อาจารย์ที่ปรึกษา	ผศ. อัครินทร์ คุณกิตติ

บทคัดย่อ

ปัจจุบันระบบอินเทอร์เน็ตมีการใช้งานเพิ่มขึ้นอย่างรวดเร็วและถูกนำไปใช้ในการให้บริการหลากหลายรูปแบบทำให้ต้องมีการขยายวงจรสื่อสารเชื่อมต่อกับอินเทอร์เน็ตเพิ่มขึ้นอยู่ตลอดเวลาเพื่อให้สามารถตอบสนองการใช้งานได้ดี ดังนั้นในโครงการนี้จึงได้ออกแบบและพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้ส่งผ่านด้วยโปรโตคอล HTTP ด้วยโปรโตคอลนี้เป็นโปรโตคอลสำหรับการให้บริการเวิลด์ไวด์เว็บซึ่งถือเป็นบริการหลักที่ใช้งาน โดยจะอาศัยระบบปฏิบัติการ FreeBSD ในการพัฒนาระบบเป็นหลัก ด้วยระบบปฏิบัติการ FreeBSD มีความสามารถในการควบคุมปริมาณการส่งผ่านข้อมูลได้อย่างดีด้วยการใช้ระบบไอพีไฟล์วอลล์ (IPFW) และระบบ Dummynet ที่มีความสามารถในการควบคุมแบนด์วิดท์มาทำงานร่วมกับโปรแกรมที่พัฒนาขึ้นที่อาศัยการผลการทำงานของโปรแกรม tcpdump ที่มีความสามารถในการตรวจจับข้อมูลที่ส่งผ่านในระบบเครือข่ายมาวิเคราะห์เลือกเฉพาะข้อมูลของโปรโตคอล HTTP เท่านั้นและสร้างกฎของไอพีไฟล์วอลล์ สำหรับควบคุมทราฟฟิกให้ส่งผ่านในช่องของแบนด์วิดท์ที่กำหนดขึ้นโดยเฉพาะจึงทำให้สามารถควบแบนด์วิทของโปรโตคอล HTTP ได้เป็นรายเครื่องหรือเครือข่ายที่มีการติดต่อสื่อสารกัน และโปรแกรมการตรวจจับและสร้างกฎการควบคุมนี้พัฒนาด้วยโปรแกรมภาษา Perl

นอกจากนั้นการออกแบบและพัฒนาระบบในโครงการได้จัดทำระบบติดต่อควบคุมสำหรับผู้ใช้งานหรือยูสเซอร์อินเทอร์เฟสเป็นแบบเว็บเบส ด้วยโปรแกรมภาษาพีเอชพี ทำให้ผู้ใช้งานมีความสะดวกในการใช้งานไม่ต้องควบคุมระบบด้วยการป้อนคำสั่งเป็นรายการย่อยเข้าระบบ ทำให้ลดข้อผิดพลาดในการใช้งานแต่สามารถควบคุมแบนด์วิดท์ได้ตามความต้องการของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	System Development of HTTP Traffic Shaping System
Student	Mr. Boonlue Jittakam
Student ID.	48066815
Degree	Master of Science
Program	Information Science
Academic Year	2008
Advisor	Asst.Prof. Akharin Khunkitti

ABSTRACT

With the soaring popularity of the Internet, a wide variety of Internet-based application services have been developed and available online including bandwidth-hungry applications like peer-to-peer file sharing. Without managing the network resources properly, these bandwidth-hungry applications may use up almost all available bandwidth, not leaving enough for other applications. Efficiently managing and controlling network resources have become increasingly important in today's Internet world.

In this paper, we design and develop the bandwidth management system that can help network administrators to prioritize and control the network traffic, especially network traffic from HTTP protocol – a primary application layer protocol for web surfing, in order to optimize the use of available network bandwidth.

Written in Perl and running on FreeBSD, the HTTP traffic shaping system analyzes the HTTP traffic intercepted by tcpdump and uses IPFW and Dummynet to create rules for controlling the traffic from each PC or each network based on organization's policies or application requirements. A simple and easy-to-use web-based configuration manager, written in PHP, is also provided to help configuring the system without typing commands.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานเรื่อง การพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้ส่งผ่านด้วยโปรโตคอล HTTP นี้สำเร็จได้ด้วยความกรุณาจาก อาจารย์ที่ปรึกษา ผศ. อัครินทร์ คุณกิตติ ที่กรุณาให้คำปรึกษาแนะนำและตรวจสอบแก้ไขข้อบกพร่องของโครงงานนี้ ตลอดจนให้ความรู้ความคิดเห็นที่เป็นประโยชน์อย่างยิ่งในการจัดทำโครงการ

ขอขอบพระคุณคณาจารย์ทุกท่านที่คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ทุกท่านได้ประสิทธิประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้ข้าพเจ้า ซึ่งความรู้ต่างๆ ได้มีส่วนช่วยในการพัฒนาโครงการนี้จนประสบผลสำเร็จ

สุดท้ายข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา ครอบครัวและเพื่อนๆ ของข้าพเจ้าที่ร่วมเป็นกำลังใจและการสนับสนุนข้าพเจ้า ด้วยดีเสมอมา

นายบุญลือ จิตตคาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของการพัฒนาระบบ	2
1.3 ขอบเขตของการพัฒนาระบบ	3
1.4 ความต้องการของระบบ	3
1.5 ขั้นตอนการพัฒนาระบบ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	5
1.6 เครื่องมือที่ใช้ในการพัฒนาระบบ	5
บทที่ 2 ระบบควบคุมปริมาณทราฟฟิก	6
2.1 ระบบควบคุมปริมาณทราฟฟิก (Traffic Control)	6
2.2 โครงสร้างการทำงานของโปรโตคอล HTTP	8
2.3 เกตเวย์ (Gateway)	13
2.4 การใช้งานระบบปฏิบัติการ FreeBSD ทำงานเกตเวย์	13
2.5 หลักการทำงานของ IPFW	15
2.6 การควบคุมปริมาณทราฟฟิกด้วย Dummynet	19
2.7 การทำงานของโปรแกรม tcpdump	24
2.8 หลักการทำงานของโปรแกรมที่พัฒนา	26
2.9 กลไกและวิธีการเชื่อมต่อระหว่างผู้ใช้งานเพื่อการจัดการกับกฎของ IPFW	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การวิเคราะห์และออกแบบระบบ 30
3.1 ศึกษาความต้องการของระบบ 30
3.2 การวิเคราะห์และออกแบบระบบงาน 31
3.2.1 การออกแบบคอนเท็กซ์ไดอะแกรม 32
3.2.2 การออกแบบดาต้าไฟล์วไดอะแกรม 32
3.2.3 การออกแบบโครงสร้างของระบบเว็บอินเทอร์เฟส..... 39
3.2.4 การออกแบบฐานข้อมูล 41
3.2.5 พจนานุกรมข้อมูล 43
3.2.6 การออกแบบส่วนเชื่อมต่อกับผู้ใช้งาน 46
3.3 การวิเคราะห์และออกแบบระบบตรวจจับโปรโตคอล HTTP (http_sniffer)..... 48

บทที่ 4 การพัฒนาและทดสอบระบบ 50
4.1 สถาปัตยกรรมของระบบ 50
4.2 การพัฒนาระบบ 52
4.3 การทดสอบระบบ 52

บทที่ 5 บทสรุป 57
5.1 สรุปโครงการ 57
5.2 ข้อเสนอแนะ 58

บรรณานุกรม 59
ภาคผนวก. 60
ภาคผนวก ก. การติดตั้งและเตรียมความพร้อมของระบบ 60
ภาคผนวก ข. คู่มือการใช้งานระบบควบคุมปริมาณการสื่อสารที่ส่งผ่านด้วย
โปรโตคอล HTTP 84

ประวัติผู้เขียน 99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงคำสั่งของโปรโตคอล HTTP..... 12

3.1 เอนทิตีของระบบเว็บอินเทอร์เฟส..... 42

3.2 รายการตารางของระบบฐานข้อมูลระบบเว็บอินเทอร์เฟส..... 43

3.3 ตาราง Rule 44

3.4 ตาราง Queue 44

3.5 ตาราง Host 44

3.6 ตาราง Pipe 45

3.7 ตาราง Interface 45

3.8 ตาราง User 45

3.9 การทำงานของระบบเว็บอินเทอร์เฟส..... 46

4.1 องค์ประกอบของระบบฮาร์ดแวร์และซอฟท์แวร์ที่ใช้สำหรับการพัฒนาระบบ 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แสดงระบบเครือข่ายเดิมที่ใช้ต่อเชื่อมอินเทอร์เน็ต	7
2.2 แสดงเมื่อเพิ่มระบบควบคุมปริมาณทราฟฟิกในเครือข่ายเดิม	7
2.3 แสดงการติดต่อสื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์	9
2.4 แสดงรูปแบบทั่วไปของ request message	11
2.5 แสดงรูปแบบทั่วไปของ response message	12
2.6 แสดงการทำงานของ Dummynet	19
2.7 แสดงดาต้าโฟลว์การทำงานของ IPFW และ Dummynet ร่วมกับโปรแกรมที่พัฒนา	27
3.1 การสั่งงานระบบผ่านทางบรรทัดคำสั่ง (Command Line)	30
3.2 การสั่งงานระบบผ่านทางเว็บอินเทอร์เฟส	31
3.3 แสดงโครงสร้างทางกายภาพของระบบเครือข่ายเมื่อใช้งาน	32
3.4 คอนแท็กซ์ไดอะแกรมของระบบเว็บอินเทอร์เฟส	32
3.5 ดาต้าโฟลว์ไดอะแกรม 0 ของระบบเว็บอินเทอร์เฟส	33
3.6 ดาต้าโฟลว์ไดอะแกรม 1	34
3.7 ดาต้าโฟลว์ไดอะแกรม 3	35
3.8 ดาต้าโฟลว์ไดอะแกรม 4	36
3.9 ดาต้าโฟลว์ไดอะแกรม 5	36
3.10 ดาต้าโฟลว์ไดอะแกรม 6	38
3.11 ดาต้าโฟลว์ไดอะแกรม 8	38
3.12 แสดงโครงสร้างโดยรวมของระบบ	40
3.13 แสดงโครงสร้างของส่วนผู้ใช้งานหลัก	40
3.14 แสดงโครงสร้างของระบบควบคุมหลัก	41
3.15 แสดงโครงสร้างของระบบควบคุมทราฟฟิก	41
3.16 อีอาร์ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปรโตคอล HTTP	42
3.17 โฟลว์ชาร์ตการทำงานของระบบตรวจจับการส่งผ่านข้อมูลของโปรโตคอล HTTP	49
4.1 แสดงการเชื่อมต่อระบบเครือข่ายเมื่อนำระบบไปใช้งาน	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.2 แสดงการเชื่อมต่อระบบเครือข่ายสำหรับการทดสอบการทำงานจริง	53
4.3 แสดงการสร้าง pipe ทีมีขนาดแบนด์วิท 4Mbit/s	53
4.4 แสดงการApply ข้อมูล pipe ที่สร้างไปใช้งานจริง	53
4.5 แสดงการสร้าง queue ทีเชื่อมต่อการทำงานกับ pipe 1	54
4.6 แสดงการApply ข้อมูล queue ที่สร้างไปใช้งานจริง	54
4.7 แสดงการสร้างกฎการ Shaping โปรโตคอล HTTP	55
4.8 แสดงการApply ข้อมูลการ Shaping ที่สร้างไปใช้งานจริง	55
4.9 การตรวจสอบการทำงานของระบบก่อนดาวน์โหลดทดสอบ	56
4.10 แสดงการดาวน์โหลดข้อมูลด้วยโปรโตคอล HTTP	56
4.11 การตรวจสอบการทำงานของระบบหลังดาวน์โหลดทดสอบ	56
4.12 แสดงกราฟสถิติข้อมูลเครือข่ายหลังการดาวน์โหลดข้อมูล	56
ก-1 หน้าจอหลัก Welcome to FreeBSD!หลังจากบู๊ตเครื่องจากแผ่นติดตั้ง	60
ก-2 หน้าจอหลักหลังจากบู๊ตเครื่องจากแผ่นติดตั้งเพื่อให้เลือกประเทศที่ติดตั้ง	61
ก-3 หน้าจอเลือก System Console Keymap	61
ก-4 หน้าจอหลักของการติดตั้ง เลือก Standard	61
ก-5 คำอธิบายว่าถัดจากนี้เป็นการกำหนด Partition ของ Disk ในรูปแบบของ DOS(fdisk)	62
ก-6 หน้าจอภาพของคำสั่ง fdisk	62
ก-7 หน้าจอให้เลือกชนิดของการติดตั้ง boot selector	62
ก-8 หน้าจออธิบายว่าจะสร้าง BSD Partition ภายใน fdisk partition	63
ก-9 หน้าจอการสร้าง BSD Partition	63
ก-10 หน้าจอเลือกชุดของการติดตั้ง	63
ก-11 หน้าจอเลือกสื่อที่ใช้สำหรับการติดตั้งระบบ	64
ก-12 หน้าจอให้ยืนยันการติดตั้ง	64
ก-13 หน้าจอข้อความแจ้งว่าระบบปฏิบัติการ FreeBSD ติดตั้งเสร็จเรียบร้อย	64
ก-14 หน้าจอ เลือกว่าต้องการกำหนดค่าให้อุปกรณ์เน็ตเวิร์ค	64
ก-15 หน้าจอกำหนดค่าของอุปกรณ์เน็ตเวิร์ค	65
ก-16 หน้าจอให้เลือกว่าต้องการกำหนดให้รองรับ IPv6 หรือไม่	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก-17 หน้าจอให้เลือกว่าต้องการกำหนดใช้ DHCP กับอินเทอร์เฟส หรือไม่	65
ก-18 หน้าจอกำหนดค่าต่างของอินเทอร์เฟส le2	66
ก-19 หน้าจอให้เลือกว่าต้องการให้ อินเทอร์เฟส le2 เปิดใช้งาน หรือไม่	66
ก-20 หน้าจอให้เลือกว่าต้องการให้เครื่องที่ติดตั้งทำหน้าที่เป็นเกตเวย์ หรือไม่	66
ก-21 หน้าจอให้เลือกว่าต้องการกำหนดค่า inetd และ network service หรือไม่	66
ก-22 หน้าจอให้เลือกเปิดการใช้งาน SSH (Secure Shell) เพื่อใช้ในการทำรีโหมด Login	67
ก-23 หน้าจอให้เลือกว่าต้องเปิดใช้บริการ FTP หรือไม่	67
ก-24 หน้าจอให้เลือกว่าต้องเปิดใช้บริการ NFS Server หรือไม่	67
ก-25 หน้าจอให้เลือกว่าต้องเปิดใช้บริการ NFS Client หรือไม่	67
ก-26 หน้าจอให้เลือกว่าต้องกำหนดค่าของ system console หรือไม่	67
ก-27 หน้าจอให้เลือกว่าต้องกำหนดค่าของ time zone หรือไม่	68
ก-28 หน้าจอให้เลือกว่าต้องใช้ค่าเวลาของเครื่องหรือ ค่า UTC	68
ก-29 หน้าจอหน้าจอให้เลือก Time Zone	68
ก-30 หน้าจอให้เลือกประเทศใน เอเชีย	69
ก-31 หน้าจอให้ยืนยันการอ้างอิงเวลา ICT	69
ก-32 หน้าจอยืนยันต้องการให้โปรแกรม Binary ของ Linux สามารถใช้บน FreeBSD ได้	69
ก-33 หน้าจอให้ยืนยันต้องการใช้ PS/2, serial Mouse หรือไม่	70
ก-34 หน้าจอให้ยืนยันต้องให้แสดงรายการ FreeBSD package หรือไม่	70
ก-35 หน้าจอให้ยืนยันต้องเพิ่มทะเบียนผู้ใช้งานอื่น นอกจาก User root หรือไม่	70
ก-36 หน้าจอแจ้งให้กำหนดรหัสผ่านของผู้ดูแลระบบ (root)	70
ก-37 หน้าจอป้อนรหัสผ่านของ User Root	71
ก-38 หน้าจอให้ยืนยันว่าต้องการกลับหน้าจอ Configuration Menu หรือไม่	71
ก-39 หน้าจอหลักของการติดตั้งระบบ	71
ก-40 หน้าจอให้ยืนยันการออกจากการติดตั้ง	72
ก-41 หน้าจอการกำหนดค่าตัวเลือกการติดตั้งของ Apache	74
ก-42 หน้าจอการกำหนดค่าตัวเลือกการติดตั้งของ php5-extensions	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ข-1 หน้าจอภาพการลอกอินเข้าระบบ	84
ข-2 หน้าจอภาพอธิบายโครงสร้างการควบคุมระบบ	85
ข-3 หน้าจอภาพการเพิ่มชื่อทะเบียนผู้ใช้ระบบ	86
ข-4 หน้าจอภาพการแก้ไขชื่อทะเบียนผู้ใช้ระบบ	86
ข-5 หน้าจอภาพการลบชื่อทะเบียนผู้ใช้ระบบ	87
ข-6 หน้าจอภาพการกำหนดค่าของเน็ตเวิร์กอินเทอร์เฟส	87
ข-7 หน้าจอภาพการนำค่าอินเทอร์เฟสที่กำหนดไปใช้งานจริง	88
ข-8 หน้าจอภาพการเพิ่มค่ารายการเครื่องคอมพิวเตอร์	88
ข-9 หน้าจอภาพการแก้ไขค่ารายการเครื่องคอมพิวเตอร์	89
ข-10 หน้าจอภาพการลบค่ารายการเครื่องคอมพิวเตอร์	89
ข-11 หน้าจอภาพการสร้างรายการ pipe	90
ข-12 หน้าจอภาพการแก้ไขรายการ pipe	90
ข-13 หน้าจอภาพการลบรายการ pipe	91
ข-14 หน้าจอภาพการนำค่าของรายการ pipe ไปใช้งานจริง	91
ข-15 หน้าจอภาพการเพิ่มรายการของ queue	92
ข-16 หน้าจอภาพการแก้ไขรายการของ queue	92
ข-17 หน้าจอภาพการลบรายการของ queue	92
ข-18 หน้าจอภาพการนำค่าของรายการ queue ไปใช้งานจริง	93
ข-19 หน้าจอภาพการเพิ่มของรายการ HTTP Shaping	93
ข-20 หน้าจอภาพการแก้ไขของรายการ HTTP Shaping	92
ข-21 หน้าจอภาพการลบของรายการ HTTP Shaping	92
ข-22 หน้าจอภาพการนำค่าของรายการ HTTP Shaping ไปใช้งานจริง	92
ข-23 หน้าจอภาพการลบของรายการ Other Shaping	95
ข-24 หน้าจอภาพการลบของรายการ Other Shaping	95
ข-25 หน้าจอภาพการลบของรายการ Other Shaping	96
ข-26 หน้าจอภาพการลบของรายการ Other Shaping	96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

ข-27 หน้าจอภาพแสดงสถิติการใช้แบนด์วิทรายวัน..... 97

ข-28 หน้าจอภาพแสดงสถิติการใช้แบนด์วิทรายสัปดาห์..... 97

ข-29 หน้าจอภาพแสดงสถิติการใช้แบนด์วิทรายเดือน..... 97

ข-30 หน้าจอภาพแสดงสถิติการใช้แบนด์วิทรายปี..... 98

ข-31 หน้าจอภาพแสดงการ Reboot เครื่อง..... 98

ข-32 หน้าจอภาพแสดงการ Shutdown เครื่อง..... 98

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ปัจจุบันเครือข่ายอินเทอร์เน็ตมีการขยายเครือข่ายการให้บริการออกไปอย่างขว้างขวาง รวดเร็วโดยมีการสร้างเครือข่ายภายในสำนักงานหรือองค์กรในลักษณะของเครือข่ายท้องถิ่นแบบ LAN หรือที่นิยมเรียกว่าเครือข่ายอินทราเน็ต และทำการขยายเครือข่ายดังกล่าวโดยการเชื่อมต่อสู่เครือข่ายอินเทอร์เน็ต ซึ่งถือเป็นปัจจัยอันหนึ่งที่ให้ผู้ใช้งานระบบเครือข่ายภายในหรือเครือข่ายอินทราเน็ตสามารถเข้าถึงข้อมูลข่าวสารในระบบเครือข่ายอินเทอร์เน็ตได้สะดวกมากขึ้น ด้วยเหตุนี้ปัญหาที่ตามมาคือเครือข่ายส่วนที่ต่อเชื่อมกับเครือข่ายอินเทอร์เน็ตจำเป็นต้องมีวงจรสื่อสารความเร็วสูงที่มีขนาดความเร็วที่รองรับปริมาณการใช้งานได้เพียงพอกับปริมาณการใช้งานของผู้ใช้งานภายในเครือข่ายอินทราเน็ตขององค์การนั้นๆ ด้วยรูปแบบการให้บริการที่เกิดขึ้นใหม่ทุกวันในระบบอินเทอร์เน็ตจึงเกิดความต้องการเพื่อให้สามารถเข้าถึงบริการและการทำธุรกรรมต่างๆ ทำให้ปริมาณการใช้งานเครือข่ายอินเทอร์เน็ตเพิ่มขึ้นสูงตามประเภทบริการที่เพิ่มขึ้น ดังนั้นทรัพยากรของระบบเครือข่ายที่ทุกคนจำเป็นต้องใช้งานร่วมกันอย่างหนักคือระบบเครือข่ายในส่วนของการต่อเชื่อมกับเครือข่ายอินเทอร์เน็ต โดยทรัพยากรส่วนนี้ต้องมีการลงทุนซื้อบริการจากผู้ให้บริการอินเทอร์เน็ตที่เรียกกันว่า Internet Service Provider (ISP) การแก้ปัญหานี้โดยการเพิ่มขนาดแบนด์วิดท์หรือความเร็วในการต่อเชื่อมระบบอินเทอร์เน็ตสามารถทำได้แต่เป็นการแก้ปัญหาระยะสั้นคือในช่วงแรกแบนด์วิดท์อาจเพียงพอแต่เมื่อผู้ใช้รู้สึกว่าการเข้าถึงเครือข่ายอินเทอร์เน็ตเร็วขึ้นก็เพิ่มปริมาณการใช้ตามระบบเครือข่ายก็จะกลับมาช้าหรือตอบสนองการใช้งานไม่เพียงพอ ปัจจุบันหลายหน่วยงานหันไปใช้กลไกการจำกัดการเข้าถึงบริการบนเครือข่ายอินเทอร์เน็ตแทน โดยเลือกอนุญาตเฉพาะบริการที่มีความจำเป็นต้องใช้งานเพื่อรองรับกับภารกิจของหน่วยงานหรือองค์การและไม่อนุญาตบริการอื่นๆ ให้สามารถใช้งานได้ในช่วงเวลาทำการหรือตลอดเวลา กลไกการกำหนดนโยบายการใช้งานบริการอินเทอร์เน็ตในรูปแบบนี้เป็นความต้องการของหลายหน่วยงานแต่ต้องแลกกับการลงทุนหาอุปกรณ์ที่ทำหน้าที่กำหนดนโยบายการควบคุมนี้ได้โดยทั่วไปอุปกรณ์ประเภทนี้ที่มีจำหน่ายในท้องตลาดได้แก่
Bandwidth Shaper ซึ่งมีผลิตภัณฑ์ที่การทำงานมีประสิทธิภาพให้เลือกหลายยี่ห้อ แต่ราคาค่อนข่างสูง ถึงสูงมาก

ด้วยเหตุนี้โครงการนี้จึงศึกษาถึงแนวทางการพัฒนาระบบควบคุมปริมาณทราฟฟิกที่ส่งผ่านในระบบเครือข่ายบนอุปกรณ์ที่ทำหน้าที่เป็น Gateway ของระบบเครือข่ายเพื่อให้การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมปริมาณทราฟฟิกสามารถควบคุมได้จริง โดยเป้าหมายของการพัฒนามองทางเลือกในการนำซอฟต์แวร์ประเภท Open Source หรือ Freeware มาใช้งานเพื่อเป็นการลดต้นทุนและสามารถนำไปใช้งานได้ในวงกว้าง คือหน่วยงานใดมีความต้องการสามารนำไปประยุกต์หรือนำไปปรับใช้งานได้โดยง่าย ในที่นี้เลือกใช้ระบบปฏิบัติการ FreeBSD เป็นระบบหลักในการติดตั้งที่ชุดอุปกรณ์ที่ทำหน้าที่เป็น Gateway ของระบบเครือข่าย ระบบปฏิบัติการ FreeBSD มีความนิยมใช้งานกันมากในระบบเครือข่ายอินเทอร์เน็ตในหลายรูปแบบ เช่น ใช้งานเป็น Gateway, Firewall, Mail Server, Web Server, Proxy Server เป็นต้น ในการพัฒนาระบบควบคุมทราฟฟิกในครั้งนี้สนใจกลไกการควบคุมปริมาณทราฟฟิกการใช้งานบริการ เวิลไวด์เว็บ (www) ที่ใช้โปรโตคอล HTTP ในการสื่อสารข้อมูลเป็นหลักโดยมุ่งเน้นที่ตัวโปรโตคอล HTTP ที่ใช้ในการสื่อสารโดยไม่สนใจพอร์ตการให้บริการ คือตามปรกติบริการเวิลไวด์เว็บจะใช้งานผ่านพอร์ตบริการ TCP/80 ระบบปฏิบัติการ FreeBSD ที่ทำหน้าที่เป็น Gateway สามารถควบคุมการใช้งานบริการต่าง ๆ ผ่านพอร์ตบริการต่างๆ ได้ดี แต่ถ้าต้องการควบคุมบริการต่างๆ โดยที่สนใจในเนื้อหาของบริการนั้นจริงๆ จำต้องสร้างเครื่องมือหรือกลไกในการตรวจสอบการทำงานของการสื่อสารนั้นๆ ว่าเป็นรูปแบบที่ต้องการหรือไม่ ดังนั้นในการพัฒนาโครงการครั้งนี้จึงมุ่งเน้นในส่วนของการติดต่อสื่อสารผ่านโปรโตคอล HTTP ซึ่งถือเป็นโปรโตคอลในชั้น Application Layer

1.2 วัตถุประสงค์ของการพัฒนาระบบ

1. เพื่อศึกษาการทำงานของระบบปฏิบัติการ FreeBSD สำหรับการทำงานเป็น Gateway ของระบบเครือข่ายในการควบคุมปริมาณทราฟฟิกในระบบเครือข่ายว่ามีความสามารถหรือข้อจำกัดในการทำงานเพียงใด
2. พัฒนาระบบวิเคราะห์การตรวจจับทราฟฟิกเพื่อเสริมการทำงานของระบบปฏิบัติการ FreeBSD ให้สามารถทำงานกรองทราฟฟิกของโปรโตคอล HTTP เพื่อนำไปใช้สร้างกลไกการควบคุมทราฟฟิกประเภทเวิลไวด์เว็บได้
3. เพื่อเป็นแนวทางในออกแบบและการพัฒนาชุดควบคุมที่มีความสามารถในการควบคุมบริการที่ใช้โปรโตคอลอื่นๆ นอกจากการสื่อสารของโปรโตคอล HTTP
4. สร้างระบบต้นแบบที่สามารถใช้งานได้จริงบนระบบเครือข่าย โดยการติดตั้งและควบคุมการใช้งานไม่ซับซ้อนสำหรับผู้ใช้งานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของการพัฒนาระบบ

1. ทำการพัฒนาระบบเฉพาะส่วนที่มีความสามารถในการกรองและควบคุมปริมาณทราฟฟิกของโปรโตคอล HTTP เท่านั้น
2. ระบบที่พัฒนาเอื้ออำนวยในการพัฒนาชุดกรองทราฟฟิกของโปรโตคอลประเภทอื่นเข้ามาเสริมหรือทำงานร่วมกับระบบที่พัฒนาเสร็จแล้ว
3. พัฒนาและออกแบบให้เหมาะสมกับการใช้งานในระบบเครือข่ายที่มีการควบคุมปริมาณทราฟฟิกไม่ซับซ้อนในการใช้งาน

1.4 ความต้องการของระบบ

ระบบที่พัฒนามีความต้องการทางด้านซอฟต์แวร์และฮาร์ดแวร์ของระบบเมื่อนำไปใช้งานประกอบด้วยส่วนสำคัญ 2 ส่วนดังต่อไปนี้

1. เครื่องคอมพิวเตอร์เซิฟเวอร์ ที่มีเน็ตเวิร์กอินเทอร์เฟสจำนวน 2 ชุด ติดตั้งซอฟต์แวร์ดังนี้
   - ระบบปฏิบัติการ FreeBSD Version 7.1
   - Apache Web Server Version 2.2
   - ตัวแปลภาษา PHP Version 5.2
   - ตัวแปลภาษา Perl Version 5.2
   - โปรแกรมตรวจสอบการทำงานของระบบเครือข่าย SNMP
   - โปรแกรมเก็บสถิติข้อมูลเครือข่าย RRD Tools
   - โปรแกรมบริหารจัดการฐานข้อมูล MySQL 5.2
   - โปรแกรมที่พัฒนาขึ้นเพื่อใช้ในการตรวจจับการส่งผ่านข้อมูลของโปรโตคอล HTTP
2. เครื่องคอมพิวเตอร์ไคล์เอ็นท์ ติดตั้งซอฟต์แวร์ดังนี้
   - ระบบปฏิบัติการ Windows Xp
   - โปรแกรมเว็บราวเซอร์ Internet Explorer 7 หรือ Firefox 3.0

1.5 ขั้นตอนการพัฒนาระบบ

การพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้ส่งผ่านด้วยโปรโตคอล HTTP มีขั้นตอนการดำเนินงาน ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ขั้นตอนการวางแผน
    - กำหนดแผนงานย่อยและแผนงานโดยรวมของทั้งระบบเพื่อช่วยกำหนดขอบเขตการทำงานโดยรวม
    - กำหนดขอบเขตของระยะเวลาในแต่ละแผนงานเพื่อใช้ควบคุมระยะเวลาสิ้นสุดของโครงการ
2. ขั้นตอนการวิเคราะห์
    - ศึกษาระบบและกลไกการทำงานควบคุมปริมาณทราฟฟิกของระบบปฏิบัติการ FreeBSD เมื่อนำมาใช้งานจริงในระบบเครือข่ายว่ามีขั้นตอนการทำงานอย่างไร
    - ศึกษาถึงกลไกการทำงานของการส่งผ่านข้อมูลผ่านโปรโตคอล HTTP เพื่อพัฒนาโปรแกรมชุดซอฟต์แวร์กรอง ทราฟฟิกของโปรโตคอล HTTP สำหรับเป็นส่วนหนึ่งของระบบควบคุมทราฟฟิกที่สร้างขึ้น
    - ศึกษาเทคโนโลยีการควบคุมทราฟฟิกอื่นๆ ที่มีใช้งานปัจจุบันเพื่อนำมาปรับใช้หรือนำมาเปรียบเทียบเพื่อเป็นทางเลือกในการนำเทคโนโลยีเหล่านั้นมาใช้งาน
    - ออกแบบโครงสร้างหลักของระบบว่ามีองค์ประกอบที่สำคัญอะไรบ้าง
    - ออกแบบโครงสร้างโดยละเอียดของทั้งระบบ
    - ตรวจสอบความถูกต้องของผลงานที่ออกแบบ
3. ขั้นตอนการออกแบบ
    - ออกแบบฐานข้อมูลของระบบเว็บอินเตอร์เฟส
    - ออกแบบโครงสร้างหน้าจอและโปรแกรมในแต่ละส่วนของระบบเว็บอินเตอร์เฟส
    - ออกแบบโปรแกรมสำหรับวิเคราะห์ผลการตรวจจับโปรโตคอล HTTP
4. ขั้นตอนการพัฒนาระบบและติดตั้งระบบ
    - ติดตั้งซอฟท์แวรที่จำเป็นสำหรับการทำงานของระบบลงบนระบบปฏิบัติการ FreeBSD
    - พัฒนาโปรแกรมสำหรับวิเคราะห์ผลการตรวจจับโปรโตคอล HTTP
    - พัฒนาระบบเว็บอินเตอร์เฟสสำหรับใช้ในการควบคุมการทำงานของระบบ
    - ทดสอบการทำงานย่อยของและระบบ
    - ทดสอบการทำงานโดยรวมของทั้งระบบเพื่อหาข้อผิดพลาดและดำเนินการแก้ไข
5. ขั้นตอนการสนับสนุนและบำรุงรักษา
    - จัดทำคู่มือการใช้งานระบบ
    - ปรับปรุงระบบหากมีการแก้ไขการทำงานของระบบเมื่อนำไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจการทำงานของระบบปฏิบัติการ FreeBSD สำหรับการทำงานเป็น Gateway ของระบบเครือข่ายในการควบคุมปริมาณทราฟฟิกในระบบเครือข่ายว่ามีความสามารถแบ่งแยกบริการในระดับ Transport Layer ได้อย่างไร
2. ได้ชุดซอฟต์แวร์เสริมการทำงานของระบบปฏิบัติการ FreeBSD สำหรับกรองทราฟฟิกของโปรโตคอล HTTP เพื่อนำไปใช้สร้างกลไกการควบคุมทราฟฟิกประเภทเวิลไวด์เว็บได้
3. ได้แนวทางในออกแบบและการพัฒนาชุดควบคุมที่มีความสามารถในการควบคุมบริการที่ใช้โปรโตคอลอื่นๆ นอกจากการสื่อสารของโปรโตคอล HTTP
4. มีระบบต้นแบบที่สามารถใช้งานได้จริงบนระบบเครือข่าย โดยการติดตั้งและควบคุมการใช้งานไม่ซับซ้อนสำหรับผู้ใช้งานทั่วไป

1.7 เครื่องมือที่ใช้ในการพัฒนาระบบ

1. ฮาร์ดแวร์ เป็นเครื่องคอมพิวเตอร์ สำหรับใช้งานพัฒนาระบบและทำหน้าที่เป็นเครื่อง Gateway ประกอบด้วยเครื่องคอมพิวเตอร์โน้ตบุค (CPU Intel Centrino Mobile 1.86 GHz, RAM 1GB, Hard Disk 40GB) สำหรับพัฒนาและจำลองการทำงานของระบบทั้งหมด และเครื่องคอมพิวเตอร์แม่ข่าย สำหรับนำระบบที่พัฒนาไปทดสอบติดตั้งใช้งานจริง
2. ซอฟต์แวร์
    - ระบบปฏิบัติการ FreeBSD release 7.0 (ติดตั้งบนเครื่องที่ทำหน้าที่เป็น Gateway)
    - gcc Compiler (สำหรับคอมไพล์เคอร์เนล)
    - Apache Web Server 2.2.6-2
    - ตัวแปลภาษา PHP 5.0
    - โปรแกรมจัดการระบบฐานข้อมูล MySQL 5.0.45
    - โปรแกรมตรวจวัดสถิติข้อมูลเครือข่าย RRD Tools
    - โปรแกรมชุดคำสั่ง IPFW และ Dummynet
    - ระบบปฏิบัติการ Windows XP (ติดตั้งบนเครื่องที่ทดสบใช้บริการเวิลไวด์เว็บ)
    - VMware-workstation-6.0 (ใช้สร้างเครื่องคอมพิวเตอร์จำลองสำหรับการพัฒนาระบบ)
    - โปรแกรม Edit Plus
    - โปรแกรมเว็บบราวเซอร์ (Firefox 3.0 และ Internet Explorer 7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ระบบควบคุมปริมาณทราฟฟิก

2.1 ระบบควบคุมปริมาณทราฟฟิก (Traffic Control)

การควบคุมปริมาณทราฟฟิกในระบบเครือข่าย หรือ Traffic Control หรือ Traffic Shaping เป็นวิธีการหรือกระบวนการในการควบคุมปริมาณทราฟฟิกหรืออัตราการส่งข้อมูล หรือทั้งการควบคุมปริมาณทราฟฟิกและอัตราการส่งข้อมูลในระบบเครือข่าย โดยอาศัยกลไกการทำงานต่าง เช่น การกำหนดหรือจัดกลุ่มผู้ใช้งาน การกำหนดนโยบายการใช้งานระบบเครือข่าย การใช้ทฤษฎีแถวคอย (Queuing Theory) การรับประกันคุณภาพของบริการ (QoS) เป็นต้น

ข้อมูลทราฟฟิกที่ไหลเวียนหรือส่งผ่านในระบบเครือข่ายจะประกอบด้วยข้อมูลหลากหลายประเภทซึ่งในแต่ละหน่วยงานก็จะให้ความสำคัญกับข้อมูลในแต่ละประเภทไม่เหมือนกันโดยเหตุผลจะขึ้นอยู่กับเป้าหมายหลักของการให้บริการระบบสารสนเทศของหน่วยงานนั้น ๆ โดยทั่วไปสมารถแบ่งประเภทของทราฟฟิกได้ดังนี้

1 . ทราฟฟิกที่ต้องให้ความสำคัญพิเศษ (Sensitive Traffic) คือชนิดของข้อมูลหรือบริการที่หน่วยงานนั้น ๆ จำเป็นต้องใช้งานหรือต้องให้บริการกับบุคคลกรของหน่วยงานที่ต้องมีการรับประกันคุณภาพการให้บริการที่ต้องสมารถใช้งานได้ตลอดเวลา เช่น บริการเว็บ บริการจดหมายอิเล็กทรอนิกส์ บริการวอยซ์โอเวอร์ไอพี (VoIP) บริการวีดีโอสตีมมิ่ง เป็นต้น และในแต่ละชนิดของข้อมูลหรือบริการอาจจำเป็นต้องจัดลำดับความสำคัญด้วย

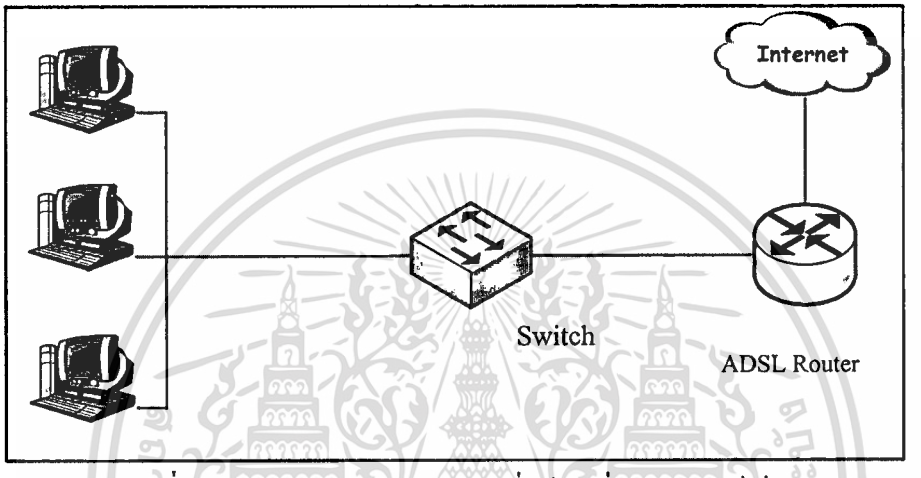
2. ทราฟฟิกที่ไม่ต้องรับประกันคุณภาพ (Best-Effort Traffic) ซึ่งเป็นข้อมูลหรือบริการที่ไม่มีความสำคัญกับหน่วยงานและอาจจัดสรรแบนด์วิดท์ส่วนที่เหลือจาก Sensitive Traffic ให้สามารถใช้บริการเหล่านี้ได้ ตัวอย่างบริการชนิดนี้ได้แก่ Peer-to-Peer เกมส์ออนไลน์ เป็นต้น .

3. ทราฟฟิกที่ไม่พึงประสงค์ (Undesired Traffic) คือข้อมูลหรือทราฟฟิกที่อาจเกิดจากการโจมตีระบบเครือข่าย หรือทราฟฟิกจากการแพร่กระจายของหนอนหรือไวรัสในระบบเครือข่าย ทราฟฟิกชนิดนี้นอกจากไม่พึงประสงค์แล้วยังต้องหากลไกการควบคุมและป้องกันด้วย

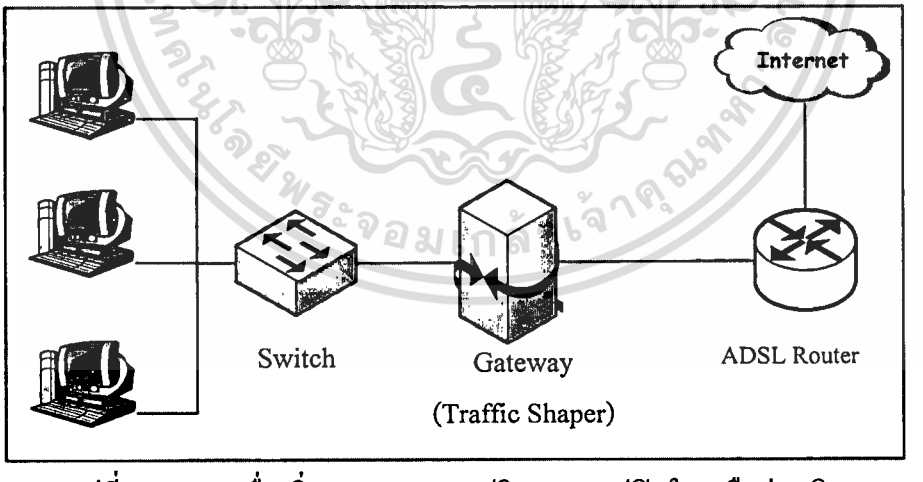
ระบบควบคุมปริมาณทราฟฟิกหรือ Traffic Control ที่พัฒนาขึ้นในครั้งนี้จะสนใจในส่วนของการควบคุมทราฟฟิกประเภท Sensitive Traffic เป็นหลัก โดยนำเอาระบบปฏิบัติการ FreeBSD ทำการติดตั้งลงบนเครื่องคอมพิวเตอร์ที่มีเน็ตเวิร์กอินเทอร์เฟส อย่างน้อย 2 ชุดต่อขั้นระหว่างเน็ตเวิร์กสองส่วนโดยสามารถที่จะส่งผ่านข้อในเครือข่ายทั้งสองส่วนหากันได้ในระดับเลเยอร์ 3 (Network Layer) โดยการทำงานในรูปแบบนี้เทียบได้กับการทำงานของอุปกรณ์เน็ตเวิร์กที่เรียกว่าเร้าเตอร์ (Router) ความสามารถเหล่านี้ระบบปฏิบัติการ FreeBSD มีความสามารถทำได้ดีโดย

เอกสารนี้มีลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพส่วนหนึ่งก็ขึ้นอยู่กับประสิทธิภาพของเครื่องคอมพิวเตอร์ที่นำมาใช้งานด้วย ข้อดีของการออกแบบระบบให้ทำงานในเกตเวย์ คือต้องแก้ไขโครงสร้างระบบเครือข่ายเดิมในเรื่องการกำหนด IP Address ของเครือข่ายทั้งสองส่วน ดังในรูปที่ 2.1 เป็นระบบเครือทั่วไปที่ต่อเชื่อมเครือข่ายภายในกับเครือข่ายอินเทอร์เน็ต ส่วนรูปที่ 2.2 แสดงการนำระบบควบคุมปริมาณทราฟฟิกที่พัฒนาขึ้นและทำงานในลักษณะเกตเวย์ต่อใช้งานในเน็ตเวิร์กเดิม โดยที่โครงสร้างของระบบเครือข่ายเดิมในส่วนอื่นไม่ต้องทำการแก้ไข



รูปที่ 2.1 แสดงระบบเครือข่ายเดิมที่ใช้ต่อเชื่อมอินเทอร์เน็ต



รูปที่ 2.2 แสดงเมื่อเพิ่มระบบควบคุมปริมาณทราฟฟิกในเครือข่ายเดิม

โดยความสามารถของระบบปฏิบัติการ FreeBSD ที่ทำงานเป็นเกตเวย์จะส่งผ่านข้อมูลระหว่าง Switch กับ ADSL ได้มีอุปกรณ์ Traffic Shaper ทำหน้าที่เป็นเกตเวย์ดังนั้นเมื่อเราต้องการให้อุปกรณ์ Traffic Shaper ชุดนี้ทำงานควบคุมปริมาณทราฟฟิกที่ส่งผ่านระหว่างระบบเครือข่ายทั้งสองจำเป็นต้องพัฒนาระบบเพิ่มความสามารถในการกรองข้อมูลที่ส่งผ่านในระบบเพื่อนำมาใช้กำหนดกลไกการควบคุมปริมาณทราฟฟิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงการนี้จะจัดทำชุดควบคุมปริมาณทราฟฟิกที่มีความสามารถควบคุมปริมาณขนาดของทราฟฟิกที่ต้องการพร้อมทั้งกรองเฉพาะโปรโตคอล HTTP ให้ไหลผ่านในท่อทราฟฟิก (pipe) ที่สร้างขึ้นเพื่อเป็นการเจาะจงในการควบคุมทราฟฟิกของโปรโตคอล HTTP โดยขบวนการทั้งหมดจะอาศัยการทำงานของระบบปฏิบัติการ FreeBSD เป็นหลัก และอาศัยการทำงานของ IPFW ซึ่งเป็นระบบไฟร์วอลล์ของ FreeBSD โดยทำหน้าที่เป็นชุดควบคุมการไหลเวียนของข้อมูลของทราฟฟิก และควบคุมการทำงานของระบบ Dummynet ที่มีหน้าที่ในการควบคุมทราฟฟิกของข้อมูลในระบบเครือข่าย และในขั้นตอนการกรองทราฟฟิกของโปรโตคอล HTTP จะมีการพัฒนาระบบตรวจจับและกรองข้อมูลของโปรโตคอล HTTP นำมาทำงานร่วมกับ IPFW ซึ่งเป็นคำสั่งการทำงานภายในของระบบปฏิบัติการ FreeBSD โดยขั้นตอนทั้งหมดต้องมีการปรับแก้และคอมไพล์เคอร์เนลของระบบปฏิบัติการ FreeBSD ใหม่เพื่อให้รองรับการทำงาน ซึ่งจะอธิบายหน้าที่และการทำงานรายละเอียดในแต่ละส่วนโดยละเอียดต่อไป

2.2 โครงสร้างการทำงานของโปรโตคอล HTTP

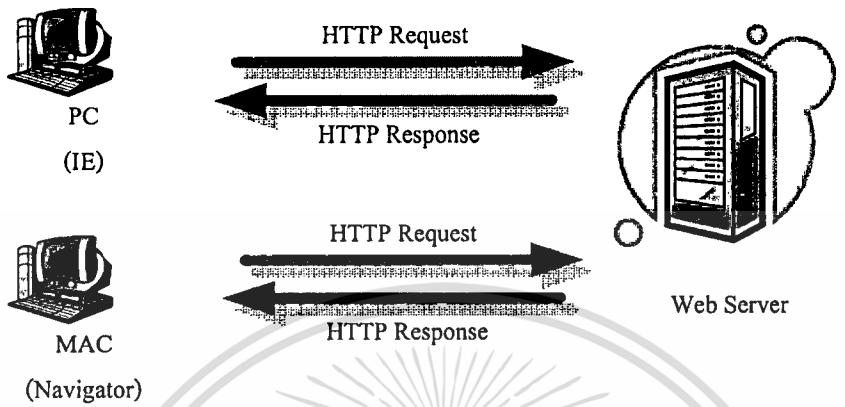
HTTP (Hyper Text Transfer Protocol) เป็นโปรโตคอลใน Application Layer ซึ่งเป็นหัวใจสำคัญของเว็บซึ่งประกอบด้วย 2 ส่วน คือ โปรแกรมไคลเอ็นต์ (Client program) และโปรแกรมเซิร์ฟเวอร์ (Server program) โดยอาจจะมีการทำงานบนระบบที่ต่างกัน มีการแลกเปลี่ยน HTTP message กันซึ่ง HTTP นั้นจะทำการกำหนดรูปแบบของข้อมูล (message) และ วิธีการแลกเปลี่ยนข้อมูลระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

เว็บเพ็จประกอบด้วยออบเจ็กต์ (Object) คือ ไฟล์ประเภทต่างๆ เช่น HTML, JPEG, GIF, Java applet, audio clip เป็นต้น ซึ่งไฟล์เหล่านี้จะมี URL เป็นตัวบอกที่อยู่ของไฟล์และจะเก็บอยู่ในรูปแบบของไฟล์ HTML สมมติว่าเว็บเพ็จประกอบด้วยไฟล์รูป JPEG 5 ไฟล์และเท็กซ์ไฟล์ HTML 1 ไฟล์ ดังนั้นเว็บเพ็จนี้จะ ประกอบด้วย 6 ออบเจ็กต์ ทำให้เวลาที่เว็บเพ็จจะอ้างถึงออบเจ็กต์ต่าง ๆ จึงต้องมีการใช้ URL ของแต่ละออบเจ็กต์นั่นเอง ในแต่ละ URL จะประกอบด้วย 2 ส่วน คือ Host name ของ server และ path ของออบเจ็กต์นั้น ๆ ตัวอย่างเช่น www.it.kmitl.ac.th/e-Leaning/picture.gif โดยมี www.it.kmitl.ac.th เป็น Host name และ ในส่วนของ /e-Leaning/picture.gif เป็น path name เป็นต้น

ส่วนบราวเซอร์ (Browser) คือโปรแกรมที่ทำหน้าที่ติดต่อระหว่างผู้ใช้ (user) กับเว็บเซิร์ฟเวอร์ (web server) บราวเซอร์ที่นิยมได้แก่ Microsoft Internet Explorer นอกจากนี้บราวเซอร์ยังทำหน้าที่ประมวลผล HTTP ของฝั่งไคลเอ็นต์ ในขณะที่ Web server เป็นที่เก็บออบเจ็กต์ต่างๆ ซึ่งในแต่ละออบเจ็กต์จะระบุที่อยู่โดย URL เว็บเซิร์ฟเวอร์ที่นิยมได้แก่ Apache, Microsoft Internet Information Server โดยที่เว็บเซิร์ฟเวอร์จะทำหน้าที่ประมวลผล HTTP ของฝั่งเซิร์ฟเวอร์ด้วย เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้ต้องการข้อมูลของเว็บเพ็จ (เช่น คลิก Hyperlink) บราวเซอร์จะส่ง HTTP request เพื่อขอรายละเอียดของเว็บเพ็จคือออบเจ็กต์ต่าง ๆ ไปยังเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ได้รับ request จะส่ง HTTP response message ซึ่งมีออบเจ็กต์ต่าง ๆ กลับไปด้วย



รูปที่ 2.3 แสดงการติดต่อสื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

จากรูปที่ 2.3 แสดงการติดต่อสื่อสารกันระหว่างไคลเอนต์ และเซิร์ฟเวอร์ เมื่อผู้ใช้ร้องขอเว็บเพ็จบราวเซอร์จะทำการส่ง HTTP request message ไปยังเซิร์ฟเวอร์ และเมื่อเซิร์ฟเวอร์ได้รับ request ก็จะส่ง HTTP response message และออบเจ็กต์ที่ร้องขอกลับไปยังไคลเอนต์ HTTP เป็นโปรโตคอลในชั้น Application layer โดยใช้บริการของโปรโตคอล TCP ดังนั้นบราวเซอร์ต้องทำการสร้างการเชื่อมต่อ (connection) โดยการสร้าง socket ขึ้นมาและติดต่อไปยังเซิร์ฟเวอร์ที่พอร์ตเบอร์ 80 ทางด้านเซิร์ฟเวอร์เมื่อได้รับคำร้องขอการต่อเชื่อมก็จะตอบกลับไปยังไคลเอนต์เพื่อเริ่มการส่ง message ต่อไป โดย message จะมีการแลกเปลี่ยนกันระหว่างบราวเซอร์กับเซิร์ฟเวอร์ และเมื่อทำการแลกเปลี่ยน message ทั้งหมดเสร็จเรียบร้อยแล้วก็จะทำการปิดการต่อเชื่อม (close connection) HTTP เป็นโปรโตคอลแบบ stateless คือจะไม่เก็บรักษาสถานะต่างๆ ของไคลเอนต์ไว้ เช่น ไคลเอนต์ ใดทำการร้องขอเว็บเพ็จมาเมื่อไร ร้องขออะไรบ้าง เป็นต้น

โปรโตคอล HTTP ในปัจจุบัน ได้พัฒนาขึ้นมาเป็นเวอร์ชัน 1.1 (จากเดิมคือ เวอร์ชัน 1.0) ซึ่งโปรแกรมบราวเซอร์ที่แพร่หลายทั่วไปนั้นจะสามารถรองรับโปรโตคอลในเวอร์ชั่นใหม่นี้ได้ และได้กำหนดไว้เป็นมาตรฐานใน RFC 2068 แล้ว โดยใน HTTP เวอร์ชัน 1.1 นี้ได้เพิ่มประสิทธิภาพทำงานให้สูงขึ้น และปรับปรุงในด้านต่างๆที่ทำให้มีความสามารถมากขึ้นดังนี้ ลดภาระของการเชื่อมต่อผ่านโปรโตคอล TCP และสามารถใช้ประสิทธิภาพของ TCP ได้อย่างเต็มที่ สามารถทำการบีบอัดข้อมูลที่รับส่งระหว่างเซิร์ฟเวอร์และไคลเอนต์ได้รองรับการทำงานแบบ virtual host หมายถึง เว็บเซิร์ฟเวอร์เครื่องหนึ่งๆ มีชื่อโดเมนมากกว่าหนึ่งชื่อได้ สามารถรองรับการทำงานได้หลายภาษา โอนไฟล์ข้อมูลเฉพาะบางส่วนได้ ซึ่งคุณสมบัตินี้จะมีประโยชน์มากในกรณีที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการโอนไฟล์ข้อมูลขนาดใหญ่และเกิดปัญหาขึ้นระหว่างการทำงาน ซึ่งโปรโตคอล HTTP 1.1 มีจุดเด่นที่สามารถตรวจสอบได้ และโอนไฟล์ข้อมูลต่อจากส่วนที่เคยโอนมาแล้ว

โครงสร้างข้อมูลของ HTTP ทั้ง HTTP/1.0 และ HTTP/1.1 กำหนดรูปแบบของ HTTP message ไว้ 2 แบบด้วยกันคือ HTTP request message และ HTTP response message

รูปแบบของ HTTP Request Message เป็นดังนี้

GET /index.html HTTP/1.1

Host: www.it.kmitl.ac.th

Connection: Close

User-agent: Mozilla/4.0

Accept-language: th

(Extra carriage return, line feed)

จะเห็นว่า Message ประกอบด้วย 5 บรรทัด โดยแต่ละบรรทัดจะจบด้วย carriage return, line feed ซึ่งจริงแล้วใน message อาจมีมากกว่า 5 บรรทัด หรือมีเพียง 1 บรรทัดก็ได้ โดยบรรทัดแรกใน HTTP request message เรียกว่า request line ส่วนบรรทัดอื่น ๆ เรียกว่า header line ใน request line จะ ประกอบด้วย 3 field ได้แก่ method field, URL field และ HTTP version field ซึ่ง HTTP request message ส่วนใหญ่จะใช้ GET method โดยถ้าใช้ GET method ข้อความที่ผู้ใช้กรอกแบบรับข้อมูลในเว็บเพ็จจะปรากฏข้อมูลอยู่ข้างหลัง URL และถ้ายาวเกินจะถูกตัดออกไป แต่ถ้าใช้ POST method ข้อความที่ผู้ใช้ กรอกจะปรากฏในส่วน body ของ request message ส่วนของ Header line มีความหมาย ดังนี้

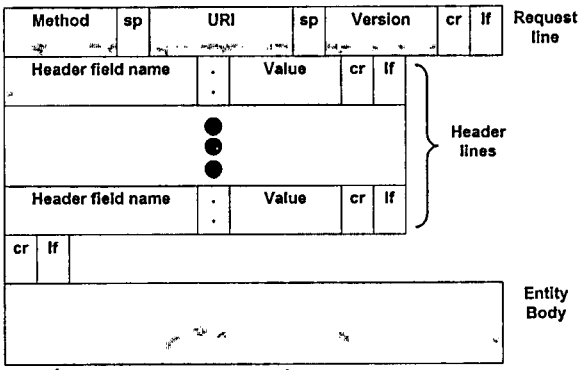
Host: ใช้ระบุชื่อเครื่อง (หรือ URL) ที่ทำการเก็บไฟล์ (Object ที่ต้องการ)

Connection: close นั้นบราวเซอร์จะบอกเซิร์ฟเวอร์ว่ามันไม่ต้องการใช้ persistent connection แต่ต้องการให้เซิร์ฟเวอร์ทำการปิด connection หลังจากเซิร์ฟเวอร์ส่งออบเจ็กต์ที่ร้องขอมายังไคลเอนต์แล้ว ดังนั้นจะพบว่าบราวเซอร์ที่สร้าง request message นี้ถูกประมวลผลโดย HTTP/1.1 แต่ไม่ต้องการใช้ persistent connection

User-agent: ใช้ทำการระบุชนิดของ browser ที่สร้างการร้องขอไปยังเซิร์ฟเวอร์ในกรณีนี้ user agent คือMozilla/4.0 ซึ่งก็คือ Netscape browser

Accept-language: ใช้บอก server ว่าชนิดของออบเจ็กต์ที่บราวเซอร์สามารถรับได้นั้นเป็นภาษาอะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงรูปแบบทั่วไปของ request message

จากรูป 2.4 แสดงรูปแบบทั่วไปของ request message ข้างต้น สังเกตได้ว่าหลังบรรทัดว่างจะตามด้วย "Entity body" โดย Entity Body จะไม่ใช้ใน GET method แต่จะใช้ใน POST method โดยที่ HTTP client จะใช้ POST method เมื่อผู้ใช้มีการใส่ข้อความเข้ามา ตัวอย่าง เมื่อผู้ใช้ ใส่คำที่ต้องการหาใน search engine เช่น Yahoo โดยใช้ POST method ซึ่งก็คือผู้ใช้ทำการร้องขอเว็บเพ็จจากเซิร์ฟเวอร์ ถ้าค่าใน Method field คือ POST ในส่วนของ Entity Body จะบรรจุค่าที่ผู้ใช้ใส่เข้ามา ส่วน HEAD method ก็คล้ายกับ POST method คือเมื่อเซิร์ฟเวอร์รับ request ด้วย HEAD method แล้ว server จะทำการตอบ HTTP message กลับมายังไคลเอนต์ โดยไม่ส่ง object ไคลเอนต์ ร้องขอกลับมาด้วย (จะส่งแค่ HTTP header กลับมา) HEAD method จะถูกใช้โดย HTTP server ที่พัฒนาขึ้นมาเพื่อทำการ debugging

รูปแบบของ HTTP Response Message เป็นดังนี้

HTTP/1.1 200 OK

Connection: Close

Date: Thu, 06 Aug 1998 10:00:15 GMT

Server: Apache/1.3.0(Unix)

Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT

Content-Length: 6821

Content-Type: text/html

Data Data Data Data Data Data .....

HTTP response message ประกอบด้วย 3 ส่วน ได้แก่ status line, header line และ Entity Body โดย Entity Body คือเนื้อของ message ซึ่งบรรจุออบเจ็กต์ ที่ไคลเอนต์ ร้องขอใน Status line มีทั้งหมด 3 field ได้แก่ protocol response field, status code และ status message จากตัวอย่างนี้ status line HTTP/1.1 200 OK จะบอกว่าเซิร์ฟเวอร์ใช้ HTTP/1.1 และทุกอย่าง OK หมายถึงเซิร์ฟเวอร์ พบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจ็กต์ที่ร้องขอแล้วและกำลังจัดส่งออบเจ็กต์ไปให้ไคลเอนต์ ในส่วนของ Header line server มีรายละเอียด ดังนี้

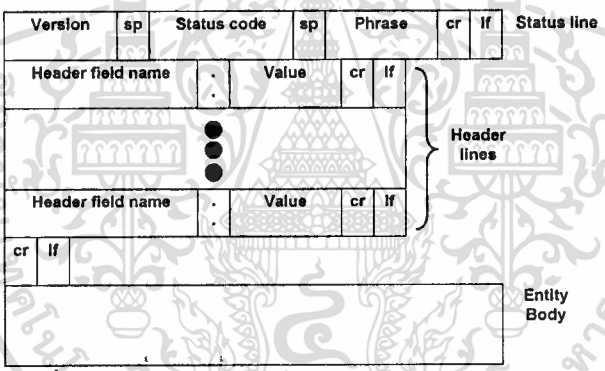
Connection : close header line เพื่อบอก ไคลเอนต์ว่าจะทำการปิด TCP connection ทันทีหลังจากทำการส่ง message เสร็จสิ้นแล้ว

Date : header line ระบุเวลาและวันที่ ที่เซิร์ฟเวอร์สร้างและส่ง HTTP response นี้สังเกตว่าเวลาดังกล่าว ไม่ใช่เวลาที่สร้างออบเจ็กต์หรือเวลาล่าสุดที่แก้ไขออบเจ็กต์ แต่เป็นเวลาที่เซิร์ฟเวอร์ดึงออบเจ็กต์ออกมาจากระบบไฟล์ แล้วใส่ออบเจ็กต์ไปใน response message และส่ง response message

Server : header line ระบุว่าใช้เว็บเซิร์ฟเวอร์อะไร ตามตัวอย่างเป็นการบอกไคลเอนต์ว่าใช้ Apache web server ซึ่งคล้าย User-agent: header line ใน HTTP request message

Content-Length : header line ระบุจำนวน byte ของออบเจ็กต์ที่จะส่งไป

Content-Type : header line ระบุว่าออบเจ็กต์ในส่วน entity body เป็น text ชนิด HTML



รูปที่ 2.5 แสดงรูปแบบทั่วไปของ response message

โปรโตคอล HTTP มีคำสั่งต่างๆ ไม่มากนัก เพื่อให้สามารถใช้งานได้อย่างสะดวกและรวดเร็ว โดยมีคำสั่งที่ใช้งานแพร่หลายอยู่ 3 คำสั่ง คือ GET HEAD และ POST รายละเอียดของคำสั่งมีดังนี้

ตารางที่ 2.1 แสดงคำสั่งของโปรโตคอล HTTP

คำสั่ง	รายละเอียด
GET	โดยทั่วไปการที่ไคลเอนต์ส่ง request message ไปที่เว็บเซิร์ฟเวอร์จะใช้ Method GET ในการร้องขอข้อมูล แต่ถ้าผู้ใช้มีการกรอกข้อความในแบบรับค่า ข้อความจะปรากฏอยู่ข้างหลัง URL ใน request message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 (ต่อ)

HEAD	คำสั่งนี้จะทำงานคล้ายกับคำสั่ง POST แต่เว็บเซิร์ฟเวอร์จะส่งข้อมูลให้เฉพาะ HTTP message กลับมายังไคลเอนต์ โดยไม่ส่งออบเจ็กต์ที่ร้องขอกลับมาด้วย คำสั่ง HEAD จะถูกใช้โดย HTTP server ที่พัฒนาขึ้นมาเพื่อทำการ debugging
POST	เป็นคำสั่งที่ทำหน้าที่ส่งข้อมูลจากไคลเอนต์ไปยังเว็บเซิร์ฟเวอร์เมื่อผู้ใช้งานกรอกข้อมูลลงใน แบบรับค่า ข้อความจะถูกใส่ลงในส่วนของ Entity Body ของ request message

2.3 เกตเวย์ (Gateway)

อุปกรณ์เกตเวย์เป็นอุปกรณ์เน็ตเวิร์กที่ใช้แก้ไขปัญหาในเรื่องสัญญาณที่วิ่งอยู่ในเครือข่ายมากเกินไปได้โดยจะจัดแบ่งเครือข่ายออกเป็นเครือข่ายย่อยหรือ Network Segment และจะทำการกลั่นกรองสัญญาณเท่าที่จำเป็นเพื่อส่งให้กับเครือข่ายย่อยที่ถูกต้องได้ ทำให้สัญญาณไม่มารบกวนกันหรือมีสัญญาณที่ไม่เกี่ยวข้องมาในเครือข่ายย่อย โดยไม่จำเป็น แต่ในทางกลับกัน ถ้ามีความจำเป็นต้องการสื่อสารกันข้ามเครือข่ายเป็นจำนวนมากแล้ว อุปกรณ์ Gateway ก็อาจกลายเป็นเสมือนคอขวดที่ทำให้เครือข่าย มีการทำงานช้าลงได้ แต่ข้อดีก็คือสามารถแยกการทำงานของเครือข่ายที่ร่วมกันเป็นส่วนๆ ได้ทำให้ช่วยลดทราฟฟิกข้อมูลที่ไม่จำเป็นเมื่อเทียบกับสวิทช์หรือฮับจะไม่สามารถทำได้

หลักการทำงานของ Gateway จะเหมือนกับการใช้อุปกรณ์เร้าเตอร์ต่อเชื่อมระหว่างเครือข่ายทั้งสองและควบคุมการส่งผ่านข้อมูลในระบบเลเยอร์ 3 (Network Layer) ดังนั้นจึงช่วยลดทราฟฟิกพิเศษที่เรียกว่า Broadcast Frame ระหว่าง 2 เครือข่ายลงได้รวมทั้งสามารถกำหนดนโยบายการสื่อสารกันระหว่างเครือข่ายได้ว่าต้องการให้เครื่องลูกข่ายใดบ้างสามารถสื่อสารกันได้

2.4 การใช้งานระบบปฏิบัติการ FreeBSD ทำงานเกตเวย์

การใช้ระบบปฏิบัติการ FreeBSD ทำงานเป็นบริดก์ ก่อนอื่นเราต้องเตรียมเครื่องคอมพิวเตอร์ที่มีเน็ตเวิร์ก อินเทอร์เฟสมากกว่า 1 ชุด (ตามโครงการนี้มีเน็ตเวิร์ก อินเทอร์เฟส 2 ชุด) จากนั้นก็ทำการติดตั้งระบบปฏิบัติการ FreeBSD รายละเอียดตามภาคผนวก ก. ที่ได้อธิบายขั้นตอนการติดตั้งโดยละเอียด โดยในการติดตั้ง FreeBSD เลือกติดตั้งชุดซอฟต์แวร์แบบ Minimal หรือ Standard ก็เพียงพอสำหรับการใช้งานเพราะไม่ต้องการให้ระบบใช้เนื้อที่ในการติดตั้งมากเกินความจำเป็น ขั้นตอนการกำหนดค่าให้ FreeBSD ทำงานเป็นเกตเวย์มี ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบรายชื่อเน็ตเวิร์กอินเทอร์เฟสในระบบเพื่อใช้ในการอ้างอิง ในที่นี้ประกอบด้วย 3 อินเทอร์เฟสหลัก คือ le0, le1 ส่วน lo0 เป็นอินเทอร์เฟส Loopback และ plip0 เป็นอินเทอร์เฟสแบบ PointToPoint

```
shaper# ifconfig
le0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:0c:29:12:0d:48
        media: Ethernet autoselect
le1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:0c:29:12:0d:52
        media: Ethernet autoselect
plip0: flags=108810<POINTOPOINT,SIMPLEX,MULTICAST,NEEDSGIANT> metric 0 mtu
1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x5
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
```

กำหนดหน้าที่ของอินเทอร์เฟสในที่จะกำหนดให้ le0 เป็นส่วนเชื่อมต่อกับ WAN หรืออุปกรณ์ที่ต่อเชื่อมออกสูระบบอินเทอร์เน็ต และ le1 สำหรับเชื่อมต่อกับระบบ LAN ภายในรวมทั้งต้องกำหนดไอพีแอดเดรสของเน็ตเวิร์กอินเทอร์เฟสทั้งสองด้วยโดยสามารถกำหนดได้ที่ไฟล์ /etc/rc.conf ดังนั้นต้องทำการแก้ไขไฟล์ดังกล่าวดังนี้

```
shaper# vi /etc/rc.conf
        firewall_enable="YES"
        firewall_type="OPEN"
        gateway_enable="YES"
        natd_enable="YES"
        natd_interface="le0"
        ifconfig_le0="inet 192.168.10.10  netmask 255.255.255.0"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ifconfig_le0="inet 192.168.10.10 netmask 255.255.255.0"

จากนั้นก็ทำการรีบูทระบบปฏิบัติการ FreeBSD ด้วยคำส่ง "shutdown –r now"

shaper# **shutdown –r now**

หลังจากเสร็จสิ้นตามขั้นตอนข้างต้นทราฟฟิกระหว่างอินเทอร์เฟส le0 กับ le1 ก็สามารถส่งผ่านหากันได้ โดยมีระบบปฏิบัติการ FreeBSD ทำหน้าที่เป็นเกตเวย์เชื่อมโยงระหว่างสองเครือข่าย

2.5 หลักการทำงานของ IPFW

ชุดคำสั่ง IPWIREWALL (IPFW) เป็นเครื่องมือสำหรับติดต่อกับผู้ใช้งานในการควบคุมกฎต่าง ๆ ของระบบไฟร์วอลล์ ซึ่ง IPFW เป็นชุดคำสั่งที่ติดตั้งอยู่ในเคอร์เนลของระบบปฏิบัติการ FreeBSD ที่มีความสามารถในการควบคุมทราฟฟิกได้ทั้งใน เลเยอร์ 2 (Data-Link Layer) เลเยอร์ 3 (Network Layer) และเลเยอร์ 4 (Transport Layer) โดยมีโครงสร้างของคำสั่งดังนี้

ipfw [-cq] add *rule*

ipfw [-acdefnNStT] [set *N*] {**list** | **show**} [*rule* | *first-last...*]

ipfw [-f | **-q] [set** *N*] **flush**

ipfw [-q] [set *N*] {**delete** | **zero** | **resetlog**} [*number* ...]

ipfw enable

{**firewall** | **altq** | **one_pass** | **debug** | **verbose** |

dyn_keepalive}

ipfw disable

{**firewall** | **altq** | **one_pass** | **debug** | **verbose** |

dyn_keepalive}

ipfw set [disable *number* ...] [**enable** *number* ...]

ipfw set move [rule] *number* **to** *number*

ipfw set swap *number number*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ipfw set show

ipfw table *number* **add** *addr*[*/masklen*] [*value*]

ipfw table *number* **delete** *addr*[*/masklen*]

ipfw table *number* **flush**

ipfw table *number* **list**

ipfw {**pipe** | **queue**} *number* **config** *config-options*

ipfw [**-s** [*field*]] {**pipe** | **queue**} {**delete** | **list** | **show**} [*number*...]

ipfw nat *number* **config** *config-options*

ipfw [**-cfnNqS**] [**-p** *preproc* [*preproc-flags*]] *pathname*

ในการใช้งานสามารถที่จะสร้างชุดคำสั่งในลักษณะโปรแกรมชุดคำสั่ง โดยประกอบด้วยคำสั่งย่อย ๆ เรียงลำดับไปแต่ละบรรทัด ซึ่งแต่ละบรรทัดถือเป็นหนึ่งชุดคำสั่งที่มีลำดับที่กำกับโดยที่ IPFW ยอมให้มีชุดคำสั่งได้ตั้งแต่ 1 – 65535 ชุดคำสั่ง และชุดคำสั่งที่ 65535 เป็นชุดคำสั่งที่ระบบ FreeBSD สร้างขึ้นไม่สมารถแก้ไขด้วยคำสั่งภายนอกได้ต้องกำหนดในขั้นตอนการคอมไพล์เคอร์เนลซึ่งชุดคำสั่งนี้จะเป็นการสั่งให้ระบบอนุญาตหรือไม่อนุญาตให้ทราฟฟิกไหลผ่านเน็ตเวิร์กอินเทอร์เฟสได้ นอกจากนี้ยังมีตัวเลือกพิเศษ หรือ Options ที่สามารถใช้งานร่วมกับ IPFW เพื่อเพิ่มความสามารถพิเศษในการใช้งานคำสั่งของ IPFW โดยมีรายละเอียดส่วนที่สำคัญดังนี้

-a ตัวเลือกนี้ใช้งานร่วมกับคำสั่ง show เพื่อแสดงเฉพาะค่าของตัวนับการใช้งาน

-b ตัวเลือกแสดงผลของคำสั่งหรือข้อคิดเห็น ไม่รวมรายละเอียดของคำสั่ง

-c กำหนดตัวเลือกนี้ เมื่อป้อนคำสั่ง จะแสดงกลับให้ดูในรูปแบบง่าย เช่น "ip from any to any"

-d กำหนดตัวเลือกนี้จะแสดง dynamic rules เพิ่มขึ้นคงที่ 1 ค่า

-e ถ้าใช้ร่วมกับ -d จะแสดงค่าเวลาหมดอายุของ dynamic rules

-f เมื่อเลือกตัวเลือกนี้ระบบจะไม่ถามการยืนยันคำสั่งแบบกลุ่ม เช่นการใช้คำสั่ง flush

-n ตัวเลือกนี้ใช้ตรวจสอบความถูกต้องของคำสั่งโดยไม่มีผลในการทำงานจริง

-N เมื่อกำหนดค่านี้จะสั่งให้แปลง Address และ Service Name ออกเป็นชื่อ

-q ตัวทำงานเช่นเดียวกับ -f ใช้ในการสั่งงานแบบโปรแกรมชุดคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคนิคการแพทย์ มหาวิทยาลัยมหิดล

-S ตัวเลือกนี้มีผลให้แสดงคำสั่ง แยกออกเป็นชุดที่เกี่ยวข้องกัน

-s [field] แสดงรายการ pipes เรียงตามปริมาณข้อมูล (total or current packets or bytes)

-t แสดงรายการล่าสุดที่ตรงกับค่าเวลาจาก ctime()

-T แสดงรายการล่าสุดที่ตรงกับค่าเวลาจากการประมวลผลก่อนหน้า

ตัวอย่างรูปแบบทั่วไปของไวยากรณ์การใช้งานคำสั่งของ IPFW

ipfw -q add deny src-ip 10.0.0.0/24,127.0.0.1/8

ipfw -q add deny src-ip 10.0.0.0/24, 127.0.0.1/8

ipfw "-q add deny src-ip 10.0.0.0/24, 127.0.0.1/8"

2.5.1 การใช้งานชุดคำสั่ง rule ของ IPFW

ชุดคำสั่งของ rule ใน IPFW มีรูปแบบและวิธีการใช้ในหลากกรณีได้แก่ ชุดคำสั่งสำหรับการเพิ่มกฎ ชุดคำสั่งสำหรับการลบกฎ ชุดคำสั่งสำหรับการแสดงกฎ เป็นต้น โดยมีรายละเอียดดังนี้

รูปแบบทั่วไปของชุดคำสั่งสำหรับการเพิ่มกฎต่าง ๆ เข้าไปในระบบมีดังนี้

ipfw add r-number **queue** q-number {**PROTOCOL**: ip | tcp | udp | icmp } **from** [address | any } [port-number] **to** {address | any } [port-number] [**via** {interface device | interface address}] [{in | out}] config-options

ipfw add คือการเพิ่มกฎเข้าไปในระบบโดยลำดับก่อนหลังจะพิจารณาจาก r-number

r-number คือหมายเลขลำดับที่ของกฎนั้นๆ ซึ่งใช้ประโยชน์ในการอ้างอิงหรือจัดลำดับก่อนหลังของกฎด้วยกฎที่มีหมายเลขลำดับที่ต่ำกว่าจะทำงานก่อนกฎที่หมายเลขลำดับสูงกว่า ด้วยการทำงานที่เรียงลำดับก่อนหลังในการเพิ่มกฎไม่จำเป็นต้องสร้างหลายเลขของกฎให้ต่อเนื่องติดกันคือสามารถเว้นได้เช่น กฎหมายเลข 100 200 300 เป็นต้นการกำหนดให้มีช่องว่างทำให้เราสามารถแทรกกฎเข้าไประหว่างกฎเดิมทั้งสองได้

q_number คือหลายเลขลำดับที่ของคิว (queue) เมื่ออยู่ในคำสั่งของ rule จะหมายถึงให้ส่งการทำงานไปยังหมายเลขคิวลำดับที่กำหนด ซึ่งใช้กรณีที่ต้องการควบคุมปริมาณทราฟฟิก

PROTOCOL คือการกำหนดชนิดของโปรโตคอลที่ต้องการใช้งานหรือควบคุม สำหรับกฎนั้นๆ โปรโตคอลที่สามารถเลือกได้คือ ip, tcp, udp, icmp หรือถ้าต้องการเลือกทั้งหมดสามารถใช้ all ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

05979

from {address | any} [port-number] เป็นการกำหนดหมายเลขโฮสต้นทางหรือเน็ตเวิร์กต้นทาง ของแพ็กเก็ตที่ส่งเข้ามายังระบบซึ่งประกอบด้วย 2 ส่วนคือ ส่วนที่ 1 {address | any} หมายถึงการกำหนดค่า ไอพีของเครื่องหรือเครือข่ายต้นทาง โดยถ้าค่า address ประกอบด้วยค่าไอพีอย่างเดียวจะหมายถึงโฮสเดียว (single host) แต่ถ้าใส่ subnet mask ก็จะพิจารณาค่าของ subnet mask ว่าควรเป็นโฮสเดียว หรือเครือข่าย และถ้าต้องการทั้งหมดหรือไม่เจาะจงใช้ any และส่วนที่ 2 port-number หมายถึงหมายเลขพอร์ตของโปรโตคอลที่กำหนด ซึ่งใช้อ้างอิงถึงประเภทของบริการ เช่น โปรโตคอล tcp พอร์ต 80 จะหมายถึงบริการเว็บ โดยถ้าไม่ระบบค่าของ port-number จะมายถึงพอร์ตใดๆ ก็ได้

to {address | any} เป็นการกำหนดหมายเลขโฮสต้นทางหรือเน็ตเวิร์กต้นทาง ของแพ็กเก็ตที่ส่งออกจากระบบ โดยรายละเอียดของการกำหนดส่วนของ {address | any} และ port-number มีวิธีการเช่นเดียวกับตัวเลือก **from** ก่อนหน้านี้

via {interface device | interface address}] [{in | out} เป็นการกำหนดเตเวิร์กอินเทอร์เฟสที่ต้องการใช้งานกับกฎนั้น ๆ ถ้าไม่กำหนดจะหมายถึงทุกอินเทอร์เฟส โดยค่าของ interface device คือชื่อของเน็ตเวิร์กอินเทอร์เฟส เช่น le0, le0 ส่วนตัวเลือก {in | out} จะเป็นการกำหนดทิศทางของไหลเวียนของทราฟฟิกโดย in คือ ทราฟฟิกที่เข้ามาที่ interface name ที่กำหนด และ out คือ ทราฟฟิกที่ออกจาก interface name ที่กำหนด

config-options เป็นตัวเลือกเพิ่มเติมในการกำหนดข้อจำกัดของปริมาณทราฟฟิก มีรูปแบบดังนี้ **below** number {K | M | G} ซึ่งเป็นการกำหนดปริมาณการส่งผ่านข้อมูลตามกฎนั้นได้ปริมาณสูงสุดไม่เกินค่าของ number ที่กำหนดโดยที่ค่าของ number สามารถกำหนดหน่วยได้เป็น K=Kilobyte, M=Megabyte, G=Gigabyte

รูปแบบทั่วไปของชุดคำสั่งสำหรับการลบกฎต่าง ๆ ที่สร้างไว้แล้วในระบบสามารถทำได้หลายกรณี ได้แก่

กรณีลบกฎหลายเลขที่กำหนดออกจากระบบมีรูปแบบการใช้คำสั่งดังนี้

ipfw delete rule number

เมื่อ number คือหมายเลขลำดับที่ของกฎที่ต้องการลบออกจากระบบ

กรณีลบกฎทั้งหมดออกจากระบบมีรูปแบบการใช้คำสั่งดังนี้

ipfw rule flush

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบทั่วไปของชุดคำสั่งสำหรับการแสดงผลกฎต่าง ๆ ที่มีในระบบมีรูปแบบดังนี้

ipfw rule number { **show** | **list** }

เมื่อ number คือหมายเลขลำดับที่ของกฎที่ต้องกาให้แสดงผลถ้าไม่มีการกำหนดจะหมายถึงแสดงผลของกฎทั้งหมดที่มีในระบบ โดยตัวเลือกหลังค่า number เป็นได้ทั้ง show หรือ list ซึ่งให้ผลเหมือนกัน

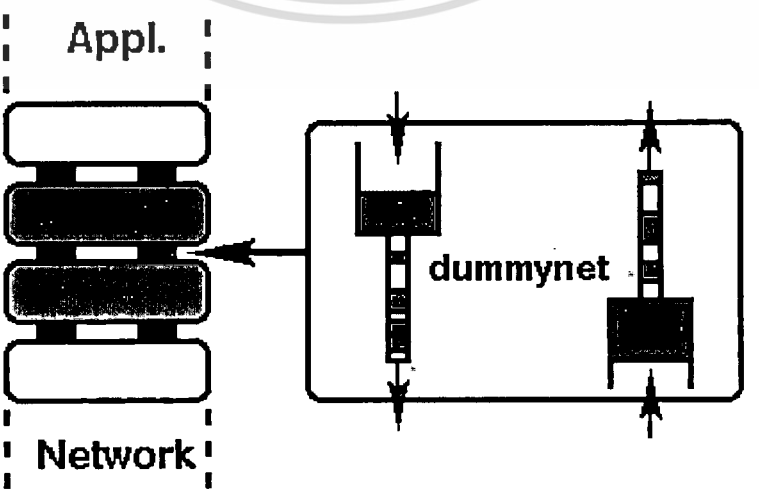
รูปแบบทั่วไปของชุดคำสั่งสำหรับลบค่าของตัวนับปริมาณข้อมูล (reset) มีรูปแบบดังนี้

ipfw zero [number1. number2.number3…]

เมื่อ number คือการระบุหมายเลขลำดับที่ของกฎที่ต้องการลบค่าตัวนับปริมาณข้อมูลถ้าหากไม่กำหนดค่าของ number จะหมายถึงการลบค่าตัวนับปริมาณข้อมูลของทุกๆ กฎในระบบ

2.6 การควบคุมปริมาณทราฟฟิกด้วย Dummynet

ภายในระบบปฏิบัติการ FreeBSD มีระบบที่ผังตัวอยู่ภายในทำหน้าที่ในการควบคุมปริมาณทราฟฟิกที่เรียกว่าระบบ Dummynet การทำงานของระบบDummynet จะทำงานที่ลำดับชั้นของโปรโตคอลระหว่างชั้นเครือข่าย (Network Layer) กับชั้นเชื่อมโยงข้อมูล (Data Link Layer) ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 แสดงการทำงานของ Dummynet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ Dummynet จะทำงานร่วมกับ IPFW ดังนั้นคำสั่งในการควบคุมปริมาณข้อมูลด้วย Dummynet จะสั่งงานผ่านคำสั่งของ IPFW และเมื่อต้องการใช้ความสามารถในเรื่องการควบคุมปริมาณการส่งผ่านข้อมูลด้วย Dummynet จำเป็นต้องคอมไพล์เคอร์เนลของระบบ FreeBSD ใหม่ โดยต้องเพิ่มตัวเลือก (option) ที่จำเป็นต่อการใช้งานดังนี้

```
# Configuretion for Traffic Shaper
options   IPFIREWALL                  # enable ipfirewall  (required for dummynet)
options   IPFIREWALL_FORWARD    # enable ipfirewall forward
options   IPFIREWALL_DEFAULT_TO_ACCEPT  # default firewall
                                                          # policy setting
options   IPFIREWALL_VERBOSE      # enable firewall output
options   IPFIREWALL_VERBOSE_LIMIT    # limit firewall output
options   DUMMYNET              # enable dummynet  operation
options   IPDIVERT                  # enable for divert  traffic
```

options IPFIREWALL เป็นตัวเลือกสำหรับการเปิดการใช้งานระบบ IPFIREWALL (IPFW) ของระบบปฏิบัติการ FreeBSD ซึ่งถือเป็นส่วนที่จำเป็นด้วยต้องใช้ในการควบคุมการทำงานของ Dummynet

options IPFIREWALL_FORWARD เป็นตัวเลือกในการกำหนดให้ระบบยอมส่งผ่านข้อมูลทราฟฟิกข้ามเน็ตเวิร์กอินเทอร์เฟส

options IPFIREWALL_DEFAULT_TO_ACCEPT เป็นตัวเลือกในการกำหนดค่าตั้งต้นของกฎของไฟร์วอลล์ว่าต้องการอนุญาตให้สามารถติดต่อสื่อสารกับอุปกรณ์ภายนอกอื่นๆ ในระบบเครือข่ายโดยถ้าไม่กำหนดตัวเลือกนี้ค่าตั้งต้นของกฎของไฟร์วอลล์จะไม่อนุญาตให้ติดต่อผ่านระบบเครือข่าย ซึ่งกฎในข้อนี้ก็คือกฎหมายเลขลำดับที่ 65535 และไม่สามารถแก้ไขได้ผ่านคำสั่งของ IPFW

options IPFIREWALL_VERBOSE เป็นตัวเลือกในการสั่งให้เก็บข้อมูลที่เกิดขึ้นของไฟร์วอลล์ลง log file ในรูปแบบของ system log file

options IPFIREWALL_VERBOSE_LIMIT =5 เป็นตัวเลือกที่ใช้ป้องการการสร้าง log file จำนวนมากๆ ที่ข้อมูลซ้ำกันได้ตามจำนวนที่กำหนดที่ยอมให้เกิดข้อมูลซ้ำกันต่อวินาที ในที่นี้คือ 5 ครั้งต่อวินาที่ ผลดีของการกำหนดค่านี้คือจะไม่ทำให้ log file มีขนาดใหญ่เกินความจำเป็น

options DUMMYNET เป็นตัวเลือกในการเปิดใช้งาน Dummynet ของ FreeBSD

options IPDIVERT เป็นตัวเลือกที่ยอมให้ IPFW ส่งข้อมูลผ่านไปยังโปรแกรมภายนอกได้โดยโปรแกรมนั้นต้องสามารถทำงานร่วมกับ IPFW ได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับขั้นตอนการคอมไพล์เคอร์เนลสามารถอ่านได้จาก ภาคผนวก ก ได้แสดงขั้นตอนและวิธีการการคอมไพล์เคอร์เนลอย่างละเอียด ส่วนชุดคำสั่งในการควบคุมการทำงานของ Dummynet ให้สามารถควบคุมปริมาณข้อมูลต่างๆ ผ่านทาง IPFW ประกอบด้วยชุดคำสั่งที่สำคัญดังนี้

2.6.1 การใช้งานชุดคำสั่ง pipe

ชุดคำสั่งในการสร้างไปท์ (pipe) เพิ่มมีรูปแบบดังนี้

ipfw pipe p-number **config** config-option

เมื่อ **p-number** คือหมายเลขลำดับที่ของไปท์ที่ต้องการเพิ่มหรือสร้างขึ้น โดยหมายเลขที่จะใช้เพื่อการอ้างอิงถึงในการใช้งาน

config-options คือการกำหนดคุณสมบัติของไปท์ในเรื่องต่างเช่น การกำหนดขนาดของแบนด์วิดท์ที่ต้องการใช้งาน การกำหนดอัตราการสูญหายของแพ็กเก็ต การกำหนดขนาดของคิว การกำหนดค่าหน่วงเวลาของแพ็กเก็ต มีรูปแบบดังนี้

bw number {bit/s | Kbit/s | Mbit/s | byte/s | Kbyte/s | Mbyte/s}

เมื่อ **bw** ซึ่งเป็นส่วนหนึ่งของ **config-options** คือแบนด์วิดท์ และ number คือขนาด หรือปริมาณของแบนด์วิดท์ของไปท์ที่ต้องการ ซึ่งหน่วยสามารถกำหนดเป็น Bit/s หรือ Byte/s ก็ได้

plr {Float 0-1}

เมื่อ **plr** ซึ่งเป็นส่วนหนึ่งของ **config-options** คือการกำหนดค่าของอัตราการสูญหายของแพ็กเก็ต โดยค่าที่สามารถกำหนดได้อยู่ในช่วง 0-1 ตัวอย่างกำหนดค่าเป็น plr 0.1 หมายความว่าจำนวนแพ็กเก็ตที่วิ่งผ่านไปท์ที่ยอมให้มีการสูญหายได้ 10%

queue {slots | size Kbyte}

เมื่อ **queue** ซึ่งเป็นส่วนหนึ่งของ **config-options** คือการกำหนดขนาดของคิวที่เชื่อมต่อมายังไปท์ นั้นๆ และสามารถกำหนดรูปแบบของคิวได้ 2 รูปแบบคือ กำหนดจำนวน slot ที่เชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มายังคิวโดยจำนวน slot จะเป็นการกำหนดจำนวนแพ็กเก็ตสูงสุดที่คิวสามารถรองรับได้ หรืออีกรูปแบบหนึ่งกำหนดเป็นขนาดของปริมาณข้อมูลสูงสุดที่คิวสมารถรองรับได้หน่วยเป็นกิโลไบท์ และถ้าไม่มีการกำหนดระบบจะกำหนดให้อัตโนมัติเป็นแบบ slot และมีจำนวน 50 slots

delay ms-delay

เมื่อ **delay** ซึ่งเป็นส่วนหนึ่งของ **config-options** คือการกำหนดค่าหน่วงเวลาของแพ็กเก็ตที่ส่งมายังไปท์ซึ่งค่าเวลาที่ถูกหน่วงจะมีค่าเป็น 1/1000 วินาที หรือ ms ถ้าหากไม่กำหนดค่าหน่วงเวลาจะมีค่าเป็น 0 ms

การลบไปท์ออกจากระบบสามารถทำได้ด้วยคำสั่งดังนี้

ipfw delete pipe p-number

เมื่อ **p-number** คือค่าของหลายเลขไปท์ที่ต้องการลบ

ถ้าต้องการลบรายการไปท์ในระบบทั้งหมดจะใช้ชุดคำสั่งดังนี้ (ผลของคำสั่งนี้นอกจากจะลบรายการไปท์ออกจากระบบแล้วยังมีผลในการลบค่าคิวที่ยังค้างอยู่ในระบบด้วย)

ipfw pipe flush

การแสดงรายการไปท์ในระบบสามารถใช้คำสั่งดังนี้

ipfw pipe p-number { list | show }

การใช้คำสั่งนี้ จะแสดงรายการไปท์หมายเลขที่กำหนดถ้าไม่กำหนดจะแสดงรายการไปท์ทั้งหมดออกมา พร้อมทั้งแสดงรายการคิวต่างที่เชื่อมต่ออยู่กับไปท์นั้นๆ ส่วนตัวเลือก list หรือ show ให้ผลเหมือนกัน

2.6.2 การใช้งานชุดคำสั่ง queue

การใช้งานชุดคำสั่งในการเพิ่มหรือสร้าง queue มีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ipfw queue q-number **config pipe** p-number config-options

เมื่อ q-number คือ หมายเลขคิวที่กำหนดขึ้นสำหรับใช้อ้างอิงการใช้งาน
p-number คือ การอ้างอิงถึงหมายลเขไปท์ที่คิวนั้นๆ จะไปเชื่อมต่อด้วย
config-options คือ การกำหนดคุณสมบัติของคิวโดยคุณสมบัติที่สำคัญคือ

weight {number 1-100}

เมื่อ weight เป็นค่า config-options ซึ่งคือค่าของน้ำหนักในการกำหนดความสำคัญของการส่งผ่านข้อมูลของคิวนั้นๆ ไปยังไปท์ เช่น มีคิวจำนวน 2 คิวส่งผ่าข้อมูลไปยังไปท์เดียวกันแต่มีค่าน้ำหนักของคิวไม่เท่ากันคิวที่มีค่านำหนักสูงกว่าจะสามารถส่งผ่านข้อมูลได้มากกว่า โดยค่าน้ำหนักที่เป็นไปได้คือ 1 ถึง 100

คำสั่งสำหรับการในการลบคิวออกจากระบบสามารถทำได้ดังนี้

ipfw delete queue q-number

เมื่อ q-number เป็นค่าของหมายเลขคิวที่ต้องการลบออกจากระบบ

ส่วนคำสั่งในการลบคิวทั้งหมดออกจากระบบสามารถทำได้ดังนี้ และควรระวังผลของคำสั่งนี้นอกจากจะลบคิวทั้งหมดออกจากระบบแล้วยังมีผลในการลบไปท์ทั้งหมดออกจากระบบด้วย

ipfw queue flush

การแสดงรายการทั้งหมดหรือบางส่วนในคิวและไปท์สามารถใช้ชุดคำสั่งดังนี้

ipfw queue q-number { **list | show**}

การใช้คำสั่งนี้ถ้าไม่ใส่ค่าตัวเลือก q-number ระบบจะแสดงรายการคิวและไปท์ทั้งหมดที่มีในระบบ ส่วนตัวเลือก list หรือ show จะได้ผลเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การทำงานของโปรแกรม tcpdump

โปรแกรม tcpdump เป็นโปรแกรมที่มีความสารถในการตรวจจับข้อมูลการสื่อสารบนระบบเครือข่ายโดยสามารถใช้งานได้ดีกับระบบปฏิบัติการประเภท UNIX เช่น SunOS, Solaris,HP-UX,Linux,Tru64,BSD เป็นต้น การใช้งานสามารถสั่งให้โปรแกรมทำงานด้วยคำสั่ง tcpdump จากนั้นโปรแกรมจำทำงานตลอดไปจนกว่าจะสั่งหยุดการทำงานด้วยคำสั่ง kill บนระบบปฏิบัติการของ UNIX หรือสั่งด้วยคำสั่ง control-C ผลลัพธ์จากการตรวจจับโดยปรกติจะแสดงออกทางหน้าจอภาพแต่ก็สามารถกำหนดให้บันทึกผลลงในแฟ้มข้อมูลได้ ซึ่งมีรูปแบบของคำสั่งในการใช้งานดังนี้

tcpdump [-AdDefIlLnNOpqRStuUvxX] [-c count]
[-C file_size] [-F file]
[-i interface] [-m module] [-M secret]
[-r file] [-s snaplen] [-T type] [-w file]
[-W filecount]
[-E spi@ipaddr algo:secret,...]
[-y datalinktype] [-Z user]
[-y datalinktype]
[expression]

ตัวเลือก (OPTIONS) ที่นิยมใช้สั่งงานของโปรแกรม tcpdump ที่สำคัญประกอบด้วย

-A แสดงข้อมูลที่ตรวจจับได้ในรูปแบบ ASCII เพื่อรองรับการตรวจจับข้อมูลของเว็บเพ็จ

-c ให้ทำงานตรวจจับจนจำนวนเพ็กเก็จได้เท่าที่กำหนด (count) จะจบการทำงาน

-C ก่อนที่จะบันทึกข้อมูลที่ตรวจจับได้ลงแฟ้มข้อมูลจะตรวจสอบขนากของแฟ้มข้อมูลปัจจุบันถ้ามีขนาดใหญ่กว่าที่กำหนด (-C file_size) จะปิดแฟ้มข้อมูลเดิมและเปิดแฟ้มข้อมูลใหม่

-D แสดงรายการชื่อเน็ตเวิร์กอินเตอร์เฟสที่สามารถตรวจจับได้ในระบบ

-e แสดงรายการ Link-Level Header บนผลของข้อมูลที่ตรวจจับได้

-F ใช้แฟ้มข้อมูล (file) ภายนอกเป็นข้อมูลที่ใช้ในเป็นเงื่อนไขการกรองข้อมูล

-i กำหนดชื่อเน็ตเวิร์กอินเตอร์เฟสที่ต้องการตรวจจับ

-l กำหนดให้แสดงผลการตรวจจับออกทางหน้าจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-L แสดงรายการ data link type ที่รู้จักหรือรองรับการทำงาน

-n ไม่ทำการแปลงค่า host address, port number เป็นชื่อ

-s เป็นการกำหนดขนาดของเพ็กเก็จที่ต้องการตรวจจับ โดยปรกติจะมีขนาด 68 ซึ่งเพียงพอต่อการใช้งานกับโปรโตคอล IP,ICMP,TCP,UDP

-t กำหนดให้ไม่แสดงค่าเวลาในผลลัพธ์ที่ตรวจจับได้

-Z ยกเลิกสิทธิ์ถ้ามีการสั่งงานด้วยรหัสผู้ใช้ root โดยเปลี่ยนสิทธิ์เป็นรหัสผู้ใช้ใหม่ที่กำหนด (-Z user)

การกำหนดเงื่อนไขการตรวจสอบ (expression) ที่นิยมใช้สั่งงานของโปรแกรม tcpdump ที่สำคัญประกอบด้วย

type ประกอบด้วย host, net, port, portrange

dir ประกอบด้วย src, dst, src หรือ dst, src และ dst

protocol ประกอบด้วย ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, lat, sca, moprc, mopdl, iso, esis, isis, icmp, icmp6, tcp และ udp

โดยที่ส่วนประกอบของเงื่อนไขทั้ง 3 ส่วนสามารถนำมาใช้งานร่วมกันเพื่อการกำหนดเงื่อนไขการกรองหรือตรวจจับข้อมูลที่ให้ตรงความต้องการมากที่สุด ที่ใช้งานทั่วไปมีดังนี้

dst host *host*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขเครื่อง (ip address) หรือชื่อเครื่องปลายทางตรงกับค่าของ *host*

src host *host*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขเครื่อง (ip address) หรือชื่อเครื่องต้นทางตรงกับค่าของ *host*

host *host*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขเครื่อง (ip address) หรือชื่อเครื่องต้นทางหรือปลายทางตรงกับค่าของ *host*

gateway *host*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขเครื่อง (ip address) หรือชื่อเครื่องที่ใช้ค่าของ *host* เป็นเกตเวย์

dst net *net*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขเครือข่ายปลายทางตรงกับค่าของ *net*

src net *net*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขเครือข่ายต้นทางตรงกับค่าของ *net*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

net *net*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขเครือข่ายต้นทางหรือปลายทางตรงกับค่าของ *net*

dst port *port*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขพอร์ตปลายทางตรงกับค่าของ *port*

src port *port*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขพอร์ตต้นทางตรงกับค่าของ *port*

port *port*

ผลการตรวจจับจะได้ข้อมูลที่มีหมายเลขพอร์ตต้นทางหรือปลายทางตรงกับค่าของ *port*

ตัวอย่างการใช้งานโปรแกรม tcpdump สำหรับตรวจจับข้อมูลของโปรโตคอล tcp โดยแสดงผลการตรวจจับผ่านทางหน้าจอภาพ

```
shaper# tcpdump -i le0 -lnx -s 1024 tcp
## ผลของการตรวจจับ
  18:55:20.480533 IP 192.168.1.10.1249 > 192.168.1.100.80: . ack 324 win 63918
        0x0000:  4500 0028 a90e 4000 8006 ce02 c0a8 010a
        0x0010:  c0a8 0164 04e1 0050 ebcf 9d84 50d1 c42a
        0x0020:  5010 f9ae 8ee5 0000 0000 0000 0000
```

-i le0 ตรวจจับข้อมูลที่ส่งผ่านทางเน็ตเวิร์กอินเตอร์เฟสชื่อ le0

-lnx กำหนดให้แสดงผลข้อมูลออกทางหน้าจอภาพ โดยไม่แปลงค่าหมายเลขไอพีหรือหมายเลขพอร์ตเป็นชื่อเครื่องหรือชื่อพอร์ต

-s 1024 กำหนดขนาดของข้อมูลที่ตรวจจับมีขนาด 1024 ไบท์

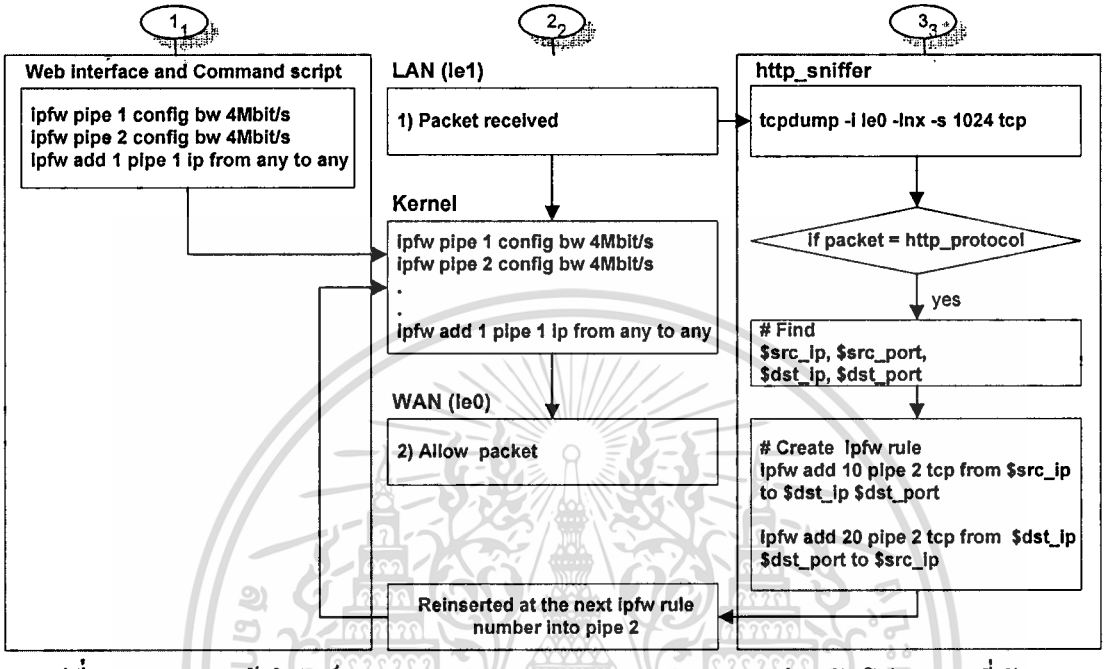
tcp กำหนดให้ตรวจจับและกรองผลลัพธ์เฉพาะข้อมูลของโปรโตคอล tcp

2.8 หลักการทำงานของโปรแกรมที่พัฒนา

โปรแกรมที่พัฒนาขึ้นเป็นการพัฒนาระบบที่อาศัยการทำงานพื้นฐานของโปรแกรม tcpdump ที่มีความสามารถในการดักจับข้อมูลที่ส่งผ่านในระบบเครือข่ายแล้วนำผลลัพธ์ของการดักจับไปแปลความหมายเพื่อหาความหมายโดยเลือกเฉพาะส่วนของข้อมูลของการติดต่อสื่อสารของโปรโตคอล HTTP ในไปหาค่าหมายเลขไอพีและพอร์ตของสารสื่อสารทั้งส่วนต้นทางและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปลายทางเพื่อนำไปสร้างกฎของ IPFW เพื่อใช้ในการควบคุมแบนด์วิดท์ ซึ่งโปรแกรมที่พัฒนาขึ้นจะใช้โปรแกรมภาษา Perl โดยทำการคอมไพล์เป็นไบนารีที่สามารถทำงานได้อย่างอิสระจากตัวแปลภาษา Perl และ สามารถทำงานร่วมกับระบบปฏิบัติการ FreeBSD ได้เป็นอย่างดี



รูปที่ 2.7 แสดงดาต้าโฟลว์การทำงานของ IPFW และ Dummynet ร่วมกับโปรแกรมที่พัฒนา

จากรูปที่ 2.7 แสดงดาต้าโฟลว์การทำงานของ IPFW ร่วมกับ http_sniffer ที่พัฒนาขึ้น โดยมีองค์ประกอบการทำงานที่สำคัญ 3 ส่วนประกอบด้วย ส่วนที่ 1 เป็นส่วนของชุดโปรแกรมเว็บอินเทอร์เฟสที่ทำหน้าที่รับการสั่งงานจากผู้ใช้ระบบแปลงเป็นโปรแกรมสคริป ส่วนที่ 2 เป็นส่วนของเคอร์เนลของระบบ FreeBSD ที่รองรับการทำงานของทั้งระบบ และส่วนที่ 3 เป็นการทำงานของโปรแกรมที่พัฒนาขึ้นไว้ทำหน้าที่ดักจับข้อมูลของโปรโตคอล HTTP เพื่อสร้างคำสั่งในการควบคุมแบนด์วิทของระบบ ซึ่งมีขั้นตอนการทำงานโดยรวมดังนี้คือ เมื่อมีแพ็กเก็ตไหลเข้ามาในระบบทางอินเทอร์เฟส le1 จะถูกส่งเข้าในส่วนของเคอร์เนลเพื่อควบคุมทิศทางการไหลเวียนของข้อมูลด้วยกฎของ IPFW ในที่นี้จะมีกฎ "ipfw add 1 pipe 1 ip from any to any" ซึ่งทำหน้าที่ควบคุมแพ็กเก็ตทั้งหมด (from any) ที่จะไหลไปปลายทางที่ใดก็ตาม (to any) จะถูกส่งจากอินเตอร์เฟส le1 ไปยังอินเตอร์เฟส le0 ผ่านทาง pipe 1 โดยมีการควบคุมขนาดของแบนด์วิดท์ด้วยคำสั่ง "ipfw pipe 1 config bw 4Mbit/s" ให้มีขนาดของแบนด์วิดท์เป็น 4 Mbit/s ในขณะเดียวกันโปรแกรม http_sniffer จะทำงานอยู่เบื้องหลัง (background process) ทำหน้าที่คอยตรวจจับข้อมูลของโปรโตคอล HTTP ซึ่งมีรายละเอียดการทำงานภายในประกอบด้วย 2 ส่วนที่สำคัญคือ ส่วนที่ 1 ทำหน้าที่ดักจับข้อมูลของโปรโตคอล TCP ทั้งหมดที่ส่งผ่านในระบบเครือข่ายโดยอาศัยผลลัพธ์จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรม tcpdump ที่คอยตรวจจับการทำงานและส่งผลการดักจับหรือส่งผลลัพธ์การทำงานไปยังส่วนที่ 2 ที่ทำหน้าที่แปลความหมายของข้อมูลและคัดเลือกเอาเฉพาะส่วนของข้อมูลของโปรโตคอล HTTP มาทำการหาหมายเลขไอพีและพอร์ตบริการของทั้งต้นทางและปลายทาง ซึ่งจะทำให้ทราบว่ามีคู่สนทนาหมายเลขไอพีคู่ใดที่กำลังติดต่อกันอยู่ จากนั้นนำผลไปสร้างเป็นกฎของไฟร์วอลล์ IPFW แทรกก่อนหน้ากฎเดิม (ipfw add 1 pipe 1 ip from any to any) ที่เคยมีโดยกำหนดให้ข้อมูลการสื่อสารของโปรโตคอล HTTP ใช้ช่องสื่อสารที่ 2 หรือ pipe 2 ที่สร้างด้วยคำสั่ง “ipfw pipe 2 config bw 4Mbit/s” ที่มีขนาดของแบนด์วิดท์เป็น 4Mbit/s ส่งผลให้ทราฟฟิกเดิมที่สื่อสารผ่านทางช่องทาง pipe 1 เปลี่ยนไปใช้ช่องทางการสื่อสาร pipe 2 แทน

การควบคุมแบนด์วิดท์ด้วยวิธีนี้มีความแตกต่างจากเดิมที่ใช้ระบบปฏิบัติการ FreeBSD ด้วยการทำงานของ IPFW และ Dummynet ที่ทำหน้าที่ควบคุมแบนด์วิดท์ ซึ่งการควบคุมจะแยกประเภทของบริการจากหมายเลขพอร์ตที่ใช้ในการติดต่อสื่อสารเป็นหลัก เช่น การควบคุมการสื่อสารของโปรโตคอล HTTP จะควบคุมที่พอร์ต TCP/80 เป็นต้น ซึ่งในโครงการที่พัฒนาขึ้นที่จะเสริมความสามารถในการควบคุมแบนด์วิดท์ของโปรโตคอล HTTP โดยการตรวจสอบจากข้อมูลที่ส่งผ่านจริงในระบบเครือข่ายว่าเป็นการติดต่อสื่อสารของโปรโตคอล HTTP ก่อนสร้างกฎในการควบคุมโดยกฎมีความสามารถในการปรับตัวตามสภาพการสื่อสารจริงของโปรโตคอล HTTP จึงทำให้ระบบปฏิบัติการ FreeBSD สามารถควบคุมทราฟฟิกได้ถึงระดับเลย์เยอร์ 7 หรือ Application Layer เฉพาะส่วนของระบบงานที่ติดต่อสื่อสารด้วยโปรโตคอล HTTP

2.9 กลไกและวิธีการเชื่อมต่อระหว่างผู้ใช้งานเพื่อการจัดการกับกฎของ IPFW

กลไกการทำงานและการสั่งการของผู้ใช้งานผ่านทางระบบสคริปไฟล์ของระบบปฏิบัติการ FreeBSD เป็นเงื่อนไขสำคัญที่ทำให้ผู้ใช้งานรู้สึกเป็นเรื่องยากในการนำเอาระบบไปใช้งานต่อ ดังนั้นเพื่อให้ผู้ใช้งานรู้สึกใช้งานง่ายขึ้นจำเป็นต้องหากลไกภายนอกมาเชื่อมต่อระหว่างผู้ใช้งานกับโปรแกรมสคริป กลไกหรือวิธีการที่น่าสนใจและนิยมใช้กันทั่วไปคือให้ผู้ใช้งานสั่งงานผ่านเว็บอินเทอร์เฟส ด้วยระบบปฏิบัติการ FreeBSD นั้นก็รองรับการทำงานเป็นเว็บเซิร์ฟเวอร์อยู่แล้ว ดังนั้นจึงต้องเลือกต่อว่าโปรแกรมภาษาที่รองรับการทำงานแบบเว็บและสามารถส่งต่อคำสั่งไปยังระบบปฏิบัติการได้ ในที่นี้จึงเลือกโปรแกรมภาษาพีเอสพี (PHP) มาใช้งานร่วมกับโปรแกรมเว็บเวิร์ฟเวอร์ของ Apache ซึ่งโปรแกรมทั้งสองชุดเป็นประเภทโอเพ็นซอร์ส (Open Source) และไม่มีค่าใช้จ่ายในการจัดหา โปรแกรมภาษาพีเอสพีมีฟังก์ชั่นการทำงานที่สามารถส่งคำสั่งตรงไปยังระบบปฏิบัตการได้ ที่เรียกว่า System Command โดยหลักๆ มี 2 ฟังก์ชันคือ

1.ฟังชั่น system() นำมาใช้ในการส่งคำสั่งไปยังระบบปฏิบัติการ FreeBSD และไม่ต้องการให้เว็บเซิร์ฟเวอร์แสดงผลการทำงานจากผลของการใช้คำสั่งกลับมาที่หน้าจออินเทอร์เฟสของเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์ตัวอย่างการการสั่งงานคำสั่งของ IPFW ผ่านทางฟังก์ชั่น system() ของพีเอสพีในการสร้างกฎต่างๆ ของ IPFW

system("ipfw add pipe 1 ip from any to any in recv le0");

2. ฟังก์ชั่น shell_exec() ใช้กรณีที่ต้องการส่งคำสั่งไปยังระบบปฏิบัติการและต้องการนำผลของคำสั่งที่เกิดขึ้นกลับมาแสดงผลทางหน้าจอเว็บอินเทอร์เฟส เช่นการใช้คำสังให้แสดงรายการกฎต่างที่ใช้งานอยู่ของ IPFW

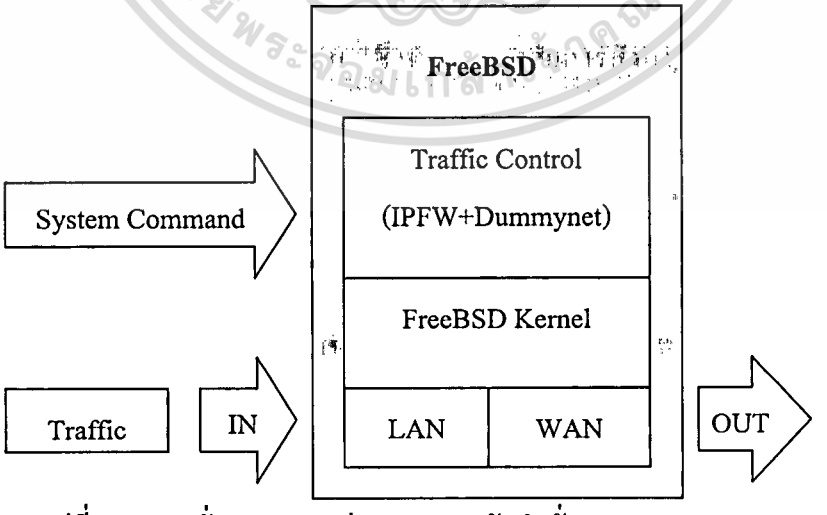
shell_exec("ipfw show");

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบระบบ

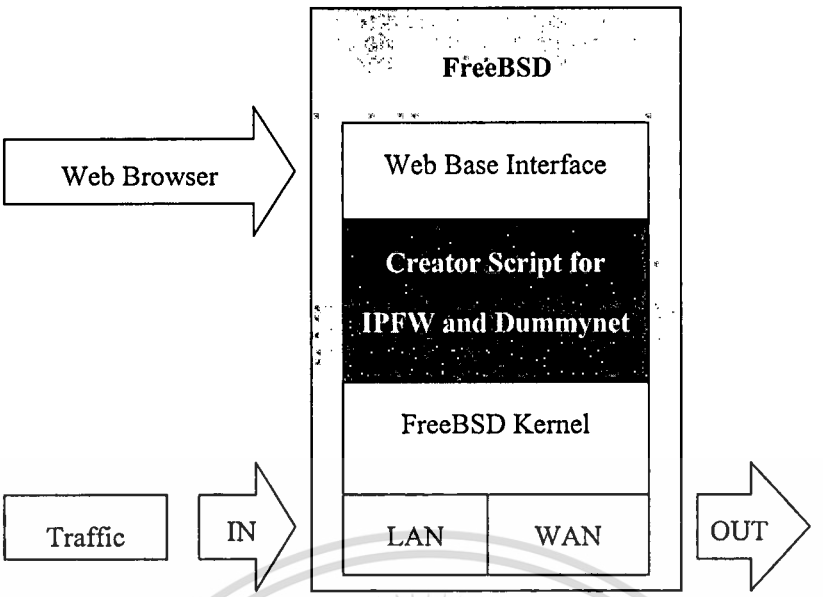
3.1 ศึกษาความต้องการของระบบ

ระบบควบคุมปริมาณการส่งผ่านข้อมูลด้วยโปรโตคอล HTTP ออกแบบและพัฒนาขึ้นโดยอาศัยระบบปฏิบัติการ FreeBSD และชุดคำสั่งภายในได้แก่ระบบ IPFW, IPDIVERT, Dummynet และชุดโปรแกรมที่พัฒนาขึ้น ซึ่งทั้งหมดอาศัยการสั่งการผ่านระบบบรรทัดคำสั่ง (Command Line) ดังแสดงในรูปที่ 3.1 โดยอาจจะรวมคำสั่งจำนวนหลายๆ รายการเป็นชุดคำสั่งเพื่อสั่งงานในครั้งเดียวได้หลายคำสั่งแต่ในขั้นตอนการสร้างและแก้ไขต้องอาศัยทักษะของผู้ดูแลระบบซึ่งมีโอกาสเกิดข้อผิดพลาดได้ง่าย โครงการนี้จึงพัฒนาส่วนติดต่อกับผู้ใช้งานเพิ่มขึ้นให้มีขั้นตอนและวิธีการที่ไม่ซับซ้อนเข้าใจง่าย โดยอาศัยเทคโนโลยีของเว็บเบสอินเทอร์เฟสดังแสดงในรูปที่ 3.2 จะมีส่วนของติดต่อกับผู้ใช้เพิ่มขึ้นมาในลักษณะเว็บเบสอินเทอร์เฟสเพื่อให้ผู้ใช้โตตอบหรือสั่งงานระบบพร้อมทั้งบันทึกรายการสั่งการทั้งหมดเก็บไว้ในฐานข้อมูลสำหรับนำมาแก้ไขปรับปรุงได้โดยง่าย จากนั้นจะมีส่วนที่ทำหน้าที่เชื่อมต่อระหว่างเว็บอินเทอร์เฟสกับระบบปฏิบัติการ FreeBSD โดยหน้าที่หลักคือสร้างชุดคำสั่งมีมาจากฐานข้อมูลเปลี่ยนเป็นชุดคำสั่งสำหรับสั่งการบนระบบในลักษณะบรรทัดคำสั่ง ซึ่งระบบสร้างชุดคำสั่งจะมีการตรวจสอบรูปแบบของข้อมูลให้อยู่ในรูปแบบที่ถูกต้องจะทำให้ลดข้อผิดพลาดได้มาเมื่อเทียบกับวิธีการแบบเดิม



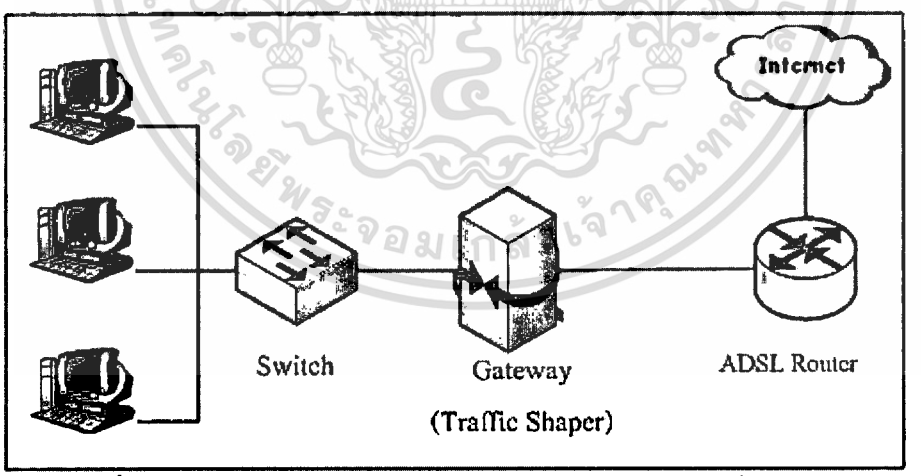
รูปที่ 3.1 การสั่งงานระบบผ่านทางบรรทัดคำสั่ง (Command Line)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การสั่งงานระบบผ่านทางเว็บอินเทอร์เฟส

ระบบควบคุมปริมาณการส่งผ่านข้อมูลด้วยโปรโตคอล HTTP ที่ออกแบบและพัฒนาเมื่อนำไปต่อใช้งานในระบบเครือจะมีวิธีการต่อเชื่อมดังแสดงในรูปที่ 3.3 ซึ่งอุปกรณ์ที่จัดทำขึ้นจะทำหน้าที่เป็นระบบเกตเวย์ของระบบเครือข่ายโดยจะเหมาะสำหรับการใช้งานในระบบเครือข่ายที่มีขาดไม่ใหญ่และการใช้งานไม่ซับซ้อน



รูปที่ 3.3 แสดงโครงสร้างทางกายภาพของระบบเครือข่ายเมื่อใช้งาน

3.2 การวิเคราะห์และออกแบบระบบงานเว็บอินเทอร์เฟส

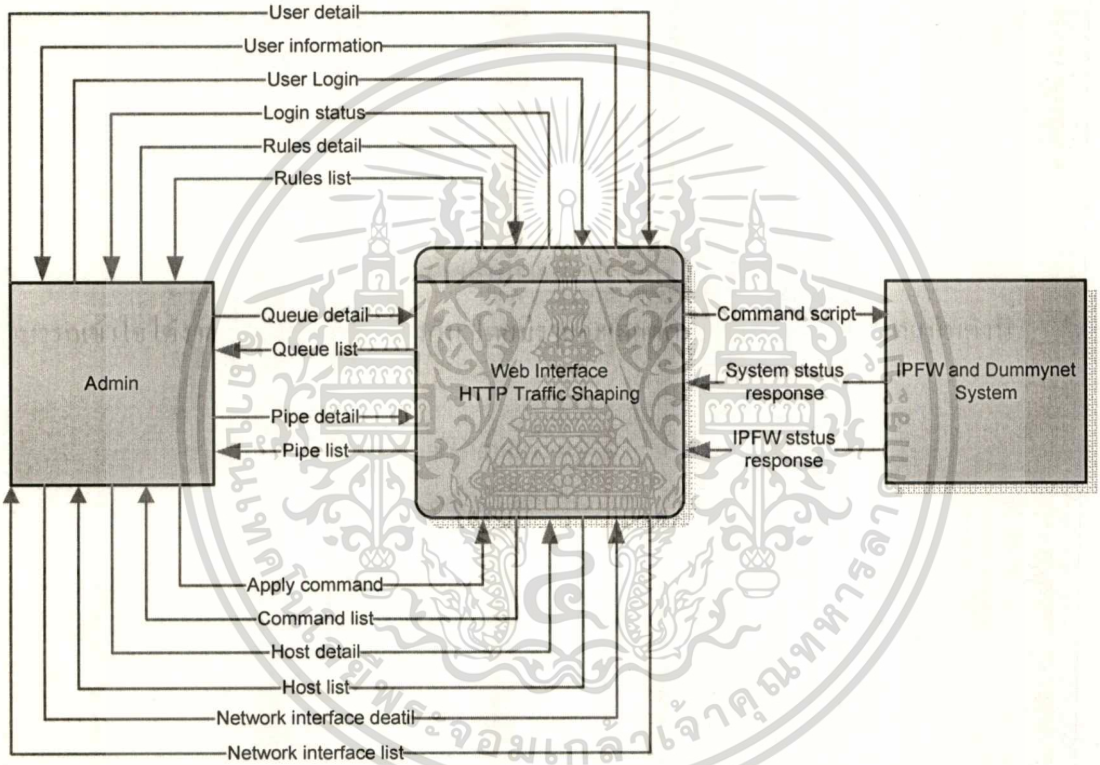
ในการวิเคราะห์และออกแบบระบบควบคุมปริมาณการส่งผ่านข้อมูลด้วยโปรโตคอล HTTP จะอาศัยแนวทางการออกแบบโดยใช้แผนภาพหรือสัญลักษณ์ในรูปแบบทั่วไปที่นิยมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาวิเคราะห์และออกแบบระบบเพื่อช่วยสื่อสารและถ่ายทอดแนวคิดในการออกแบบระบบให้สามารถเข้าใจได้ง่ายและสื่อสารได้ตรงกัน โดยมีรายละเอียดการวิเคราะห์และออกแบบดังนี้

3.2.1 การออกแบบคอนเท็กซ์ไดอะแกรมของระบบเว็บอินเทอร์เฟส

แผนภาพคอนเท็กซ์ไดอะแกรมดังแสดงในรูปที่ 3.4 เป็นแผนภาพโดยรวมของระบบที่ช่วยให้มองเห็นภาพความสัมพันธ์ขององค์ประกอบต่างๆ ของระบบตลอดจนขอบเขตการทำงานของระบบว่ามีการทำงานโดยร่วมอย่างใดและเกี่ยวเนื่องกับส่วนต่างอย่างใด



รูปที่ 3.4 คอนเท็กซ์ไดอะแกรมของระบบเว็บอินเทอร์เฟส

3.2.2 การออกแบบดาต้าไฟลว์ไดอะแกรมของระบบเว็บอินเทอร์เฟส

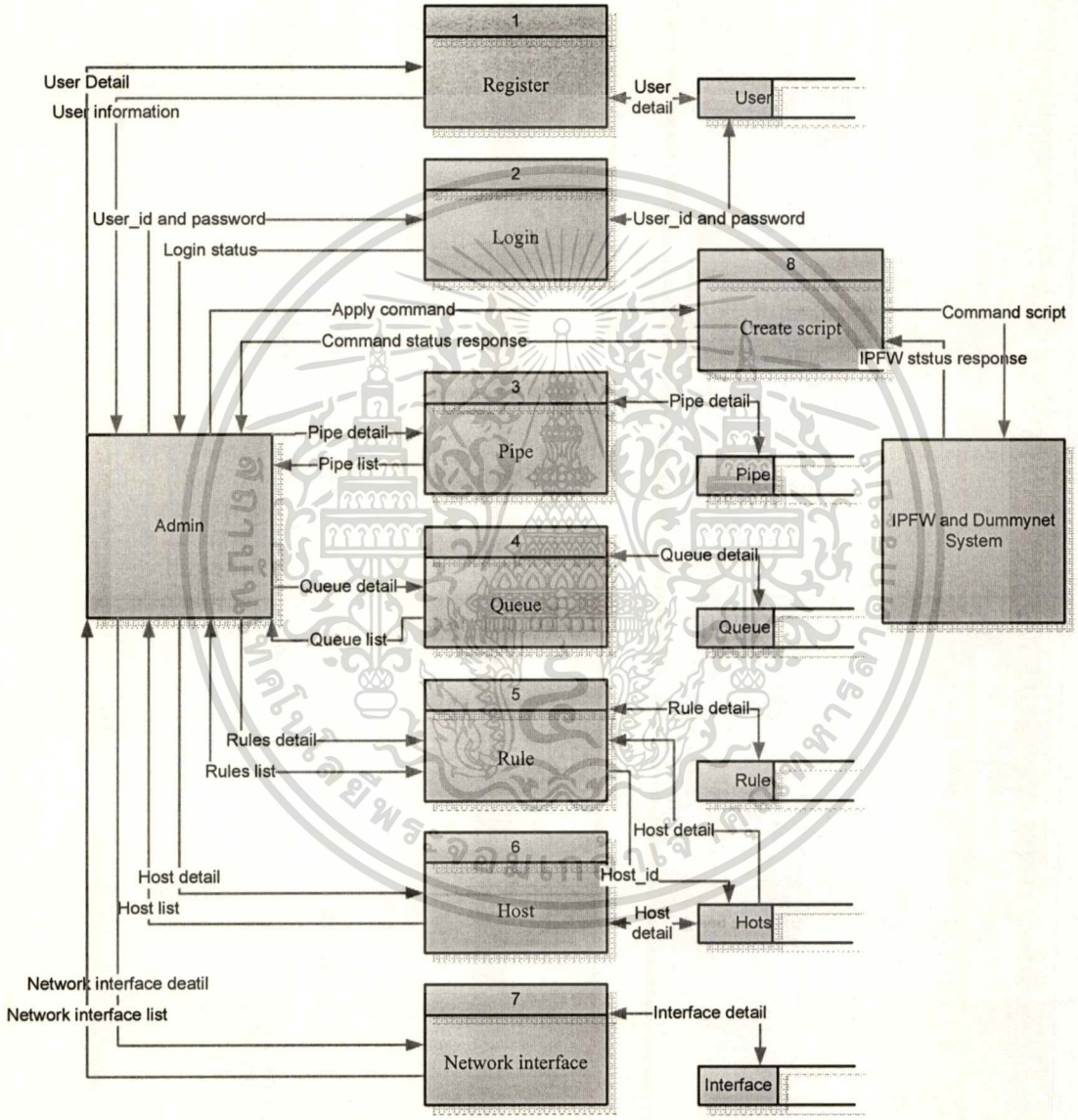
ดาต้าโฟลว์ไดอะแกรมเป็นแผนภาพที่อธิบายรายละเอียดการทำงานของระบบควบคุมปริมาณการส่งผ่านข้อมูลด้วยโปรโตคอล HTTP โดยจะมีรายละเอียดมากขึ้นจากแผนภาพคอนเท็กซ์ไดอะแกรม ดังแสดงในรูปที่ 3.5 เป็นแผนภาพดาต้าโฟลว์ไดอะแกรมระดับ 0 ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปรโตคอล HTTP ซึ่งจะมีรายละเอียดของโปรเซสต่างในระบบทั้งหมดที่จำเป็นได้แก่โปรเซส Register, Login, Pipe, Queue, Rule, Host, Network interface และ Create script โดยรูปแบบที่นำเสนอสามารถเข้าใจได้โดยง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของเดต้าโฟลว์ไดอะแกรม 0 ของระบบควบคุมปริมาณการส่งผ่านข้อมูลด้วยโปรโตคอล HTTP ของโปรเซสมีรายละเอียดที่สามารถอธิบายได้ดังนี้

ส่วนของเอนทิตีภายนอก (External Entity) ประกอบด้วย

- Admin หมายถึงผู้ดูแลและควบคุมระบบ
- IPFW and Dummynet system หมายถึงระบบควบคุมแบนด์วิดท์ของระบบ FreeBSD



รูปที่ 3. 5 ดาต้าโฟลว์ไดอะแกรม 0 ของระบบเว็บอินเทอร์เฟส

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- Register หมายถึง กระบวนการสร้างและแก้ไขข้อมูลของผู้ดูแลระบบ
- Login หมายถึง กระบวนการในการพิสูจน์ตัวตนของผู้ดูแลระบบที่จะเข้ามาใช้งานระบบ

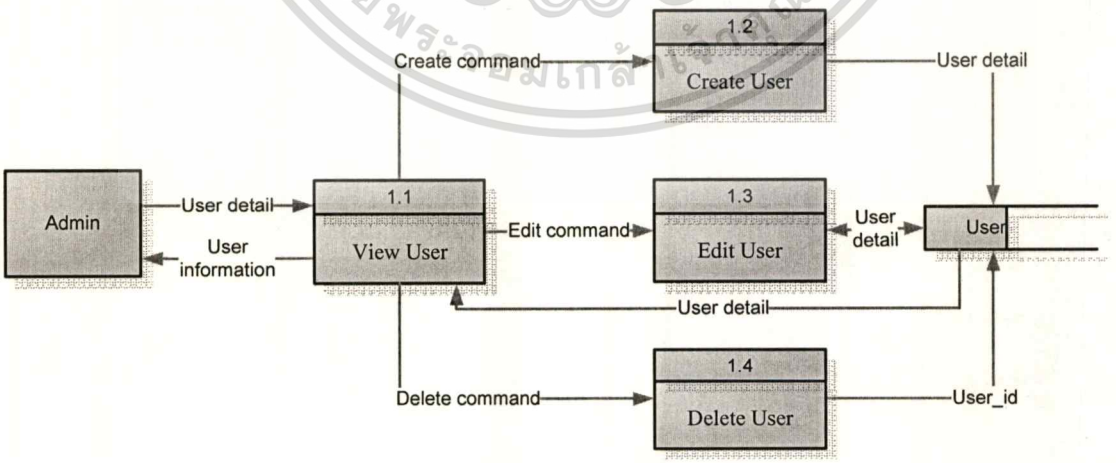
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Pipe หมายถึงกระบวนการในการสร้างและแก้ไขข้อมูลของไปท์ (pipe)ในระบบ
- Queue หมายถึงกระบวนการในการสร้างและแก้ไขข้อมูลคิวในระบบ
- Rule หมายถึงกระบวนการในการสร้างและแก้ไขข้อมูลกฎต่างของระบบ IPFW ในระบบ
- Host หมายถึงกระบวนการในการสร้างและแก้ไขข้อมูลรายการชื่อเครื่องและเครือข่ายเพื่อใช้ประกอบการสร้างกฎต่างๆ
- Network Interface หมายถึงกระบวนการในการสร้างและแก้ไขข้อมูลไอพีแอดเดรสของเน็ตเวิร์กอินเทอร์เฟสของระบบ

ส่วนของที่เก็บข้อมูล (Data store) ประกอบด้วย

- User หมายถึง ที่เก็บข้อมูลของผู้ดูแลและควบคุมระบบ
- Pipe หมายถึง ที่เก็บข้อมูลคอนฟิกของไปท์
- Queue หมายถึง ที่เก็บข้อมูลคอนฟิกของคิว
- Rule หมายถึง ที่เก็บข้อมูลคอนฟิกของกฎต่างๆ
- Host หมายถึง ที่เก็บข้อมูลรายการของชื่อโฮสและเครือข่ายที่ใช้งาน
- Interface หมายถึง ที่เก็บข้อมูลคอนฟิกไอพีแอดเดรสของเน็ตเวิร์กอินเทอร์เฟส

สำหรับรายละเอียดการทำงานภายในของแต่ละโปรเซสจะขออธิบายรายละเอียดเฉพาะส่วนที่สำคัญๆ ได้แก่กระบวนการย่อยของโปรเซส User, Pipe, Queue, Rule, Host และ Create script โดยมีรายละเอียดของแต่ละกระบวนการดังนี้



รูปที่ 3. 6 ดาต้าโฟลว์ไดอะแกรม 1

รายละเอียดทิศทางการไหลของข้อมูล ดาต้าโฟลว์ไดอะแกรม 1 ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปร โตคอล HTTP ดังรายละเอียดในรูปที่ 3.6 สามารถอธิบายรายละเอียดของโปรเซสและข้อมูลได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

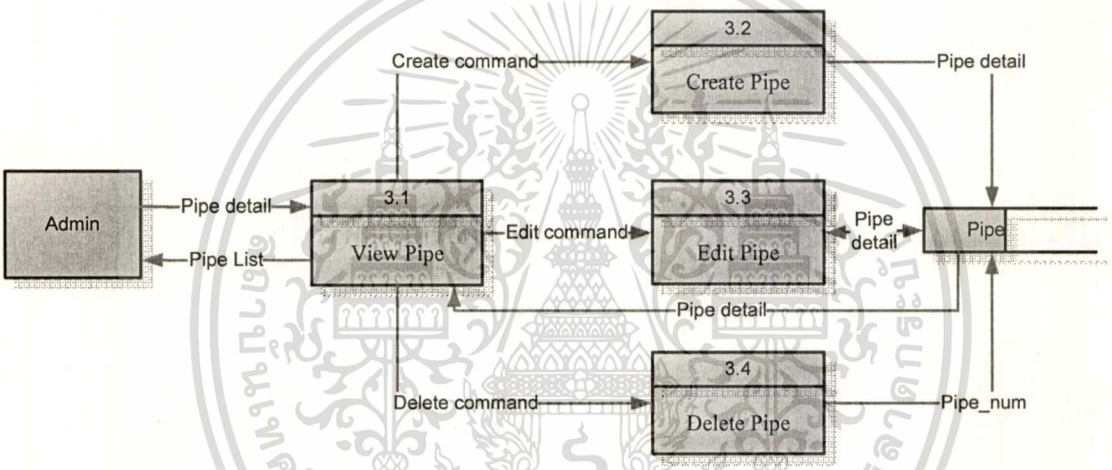
- Admin หมายถึง ผู้ดูแลหรือผู้ควบคุมระบบ

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- View User หมายถึง การแสดงรายละเอียดของผู้ดูแลหรือควบคุมระบบ
- Create User หมายถึง การสร้างรายการของผู้ดูแลหรือควบคุมระบบ
- Edit User หมายถึง การแก้ไขรายการของผู้ดูแลหรือควบคุมระบบ
- Delete User หมายถึง ลบข้อมูลของผู้ดูแลหรือควบคุมระบบ

ส่วนของที่เก็บข้อมูล (Data store) ประกอบด้วย

- User หมายถึง ที่เก็บข้อมูลของผู้ดูแลหรือควบคุมระบบ



รูปที่ 3. 7 ดาต้าโฟลว์ไดอะแกรม 3

รายละเอียดทิศทางการไหลของข้อมูล ดาต้าโฟลว์ไดอะแกรม 3 ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปรโตคอล HTTP ดังรายละเอียดในรูปที่ 3.7 สามารถอธิบายรายละเอียดของโปรเซสและข้อมูลได้ดังนี้

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- Admin หมายถึง ผู้ดูแลหรือผู้ควบคุมระบบ

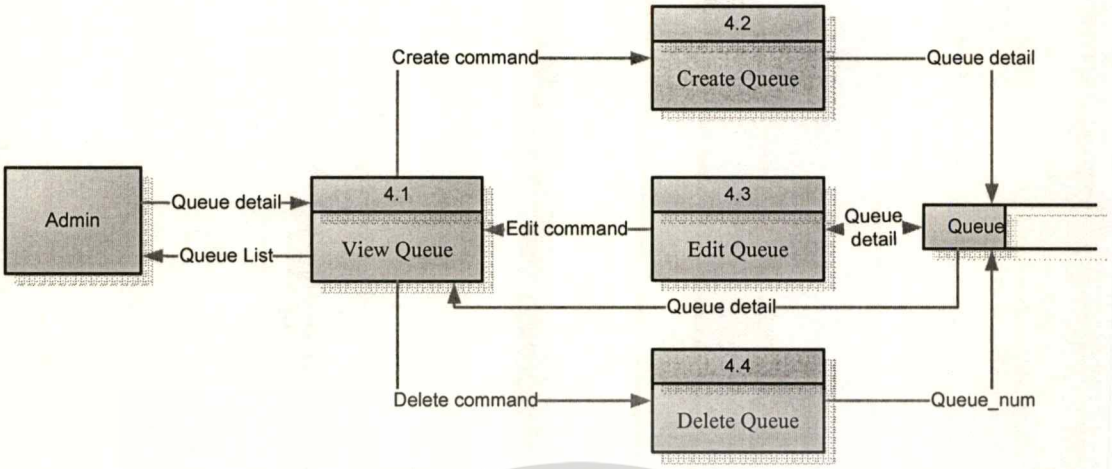
ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- View Pipe หมายถึง การแสดงรายละเอียดข้อมูลของไปท์ในระบบ
- Create Pipe หมายถึง การสร้างรายการข้อมูลของไปท์เข้าในระบบ
- Edit Pipe หมายถึง การแก้ไขรายการข้อมูลของไปท์ในระบบ
- Delete Pipe หมายถึง ลบข้อมูลของไปท์ในระบบ

ส่วนของที่เก็บข้อมูล (Data store) ประกอบด้วย

- Pipe หมายถึง ที่เก็บข้อมูลของไปท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 8 ดาต้าโฟลว์ไดอะแกรม 4

รายละเอียดทิศทางการไหลของข้อมูล ดาต้าโฟลว์ไดอะแกรม 4 ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปรโตคอล HTTP ดังรายละเอียดในรูปที่ 3.8 สามารถอธิบายรายละเอียดของโปรเซสและข้อมูลได้ดังนี้

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

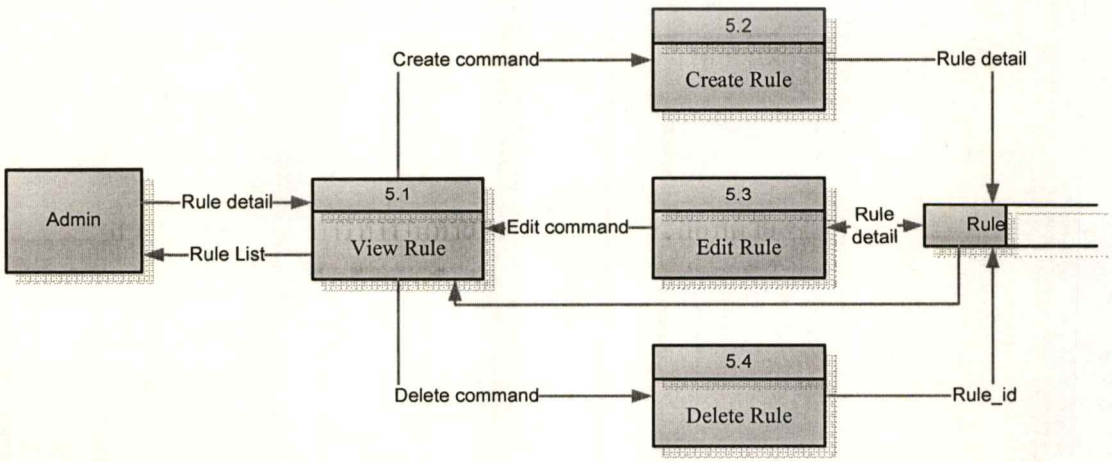
- Admin หมายถึง ผู้ดูแลหรือผู้ควบคุมระบบ

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- View Queue หมายถึง การแสดงรายละเอียดข้อมูลของคิวในระบบ
- Create Queue หมายถึง การสร้างรายการข้อมูลของคิวเข้าในระบบ
- Edit Queue หมายถึง การแก้ไขรายการข้อมูลของคิวในระบบ
- Delete Queue หมายถึง ลบข้อมูลของคิวในระบบ

ส่วนของที่เก็บข้อมูล (Data store) ประกอบด้วย

- Queue หมายถึง ที่เก็บข้อมูลของคิว



รูปที่ 3. 9 ดาต้าโฟลว์ไดอะแกรม 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดทิศทางการไหลของข้อมูล ดาต้าโฟลว์ไดอะแกรม 5 ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปรโตคอล HTTP ดังรายละเอียดในรูปที่ 3.9 สามารถอธิบายรายละเอียดของโปรเซสและข้อมูลได้ดังนี้

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- Admin หมายถึง ผู้ดูแลหรือผู้ควบคุมระบบ

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- View Rule หมายถึง การแสดงรายละเอียดข้อมูลของกฎต่างๆ ในระบบ
- Create Rule หมายถึง การสร้างรายการข้อมูลของกฎต่างๆ เข้าในระบบ
- Edit Rule หมายถึง การแก้ไขรายการข้อมูลของกฎต่างๆ ในระบบ
- Delete Rule หมายถึง ลบข้อมูลของกฎต่างๆ ในระบบ

ส่วนของที่เก็บข้อมูล (Data store) ประกอบด้วย

- Rule หมายถึง ที่เก็บข้อมูลของกฎต่างๆ

รายละเอียดทิศทางการไหลของข้อมูล ดาต้าโฟลว์ไดอะแกรม 6 ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปรโตคอล HTTP ดังรายละเอียดในรูปที่ 3.10 สามารถอธิบายรายละเอียดของโปรเซสและข้อมูลได้ดังนี้

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- Admin หมายถึง ผู้ดูแลหรือผู้ควบคุมระบบ

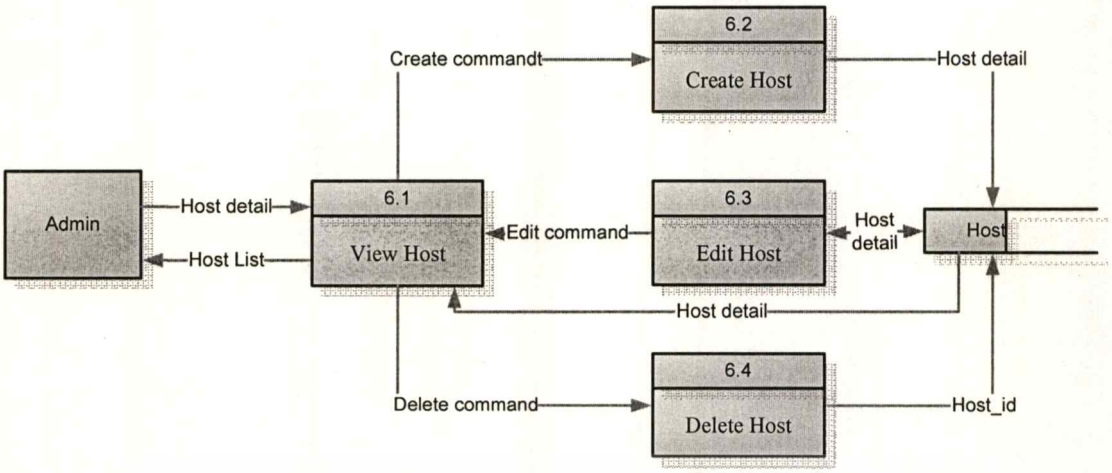
ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- View Host หมายถึง การแสดงรายละเอียดข้อมูลของชื่อเครื่องและชื่อเครือข่ายในระบบ
- Create Host หมายถึง การสร้างรายการข้อมูลของชื่อเครื่องและชื่อเครือข่าย เข้าในระบบ
- Edit Host หมายถึง การแก้ไขรายการข้อมูลของชื่อเครื่องและชื่อเครือข่าย ในระบบ
- Delete Host หมายถึง ลบข้อมูลของชื่อเครื่องและชื่อเครือข่าย ในระบบ

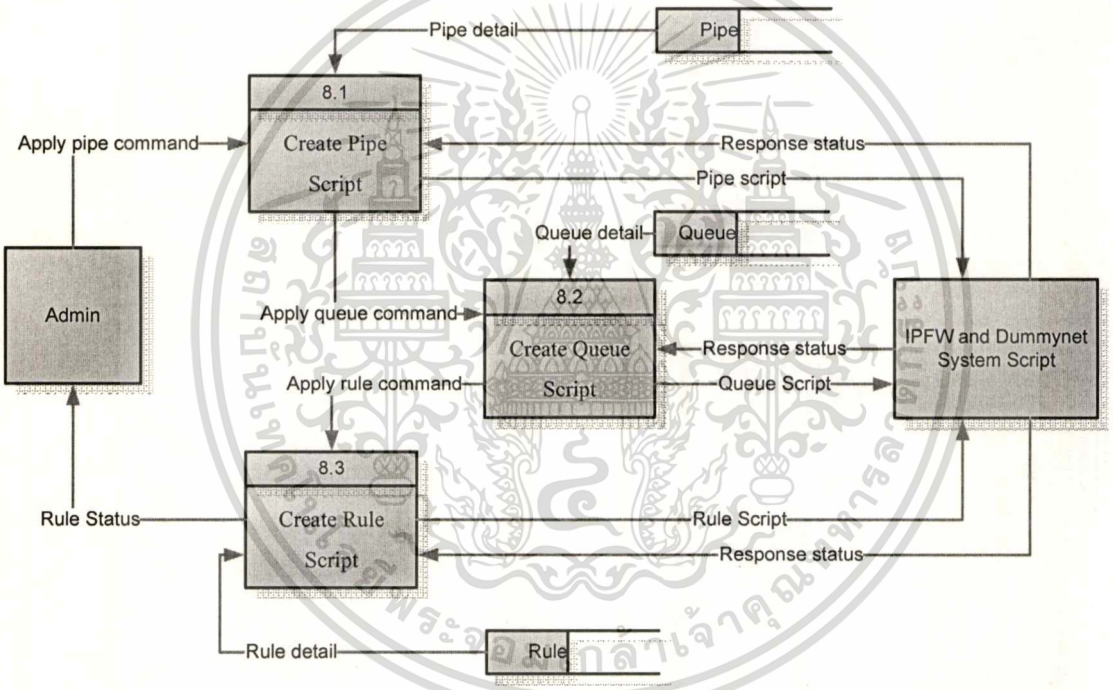
ส่วนของที่เก็บข้อมูล (Data store) ประกอบด้วย

- Host หมายถึง ที่เก็บข้อมูลของชื่อเครื่องและชื่อเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 10 ดาต้าโฟลว์ไดอะแกรม 6



รูปที่ 3. 11 ดาต้าโฟลว์ไดอะแกรม 8

รายละเอียดทิศทางการไหลของข้อมูล ดาต้าโฟลว์ไดอะแกรม 8 ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปรโตคอล HTTP ดังรายละเอียดในรูปที่ 3.11 สามารถอธิบายรายละเอียดของโปรเซสและข้อมูลได้ดังนี้

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

- Admin หมายถึง ผู้ดูแลหรือผู้ควบคุมระบบ
- IPFW and Dummynet system หมายถึงระบบควบคุมแบนด์วิดท์ของระบบ FreeBSD

ส่วนของกระบวนการทำงานภายใน (Process) ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Create Pipe Script หมายถึง การสร้างคำสั่งแบบบรรทัดหรือ System Command เพื่อสั่งการทำงานของระบบปฏิบัติการ FreeBSD สำหรับการสร้างไปท์
- Create Queue Script หมายถึง การสร้างคำสั่งแบบบรรทัดหรือ System Command เพื่อสั่งการทำงานของระบบปฏิบัติการ FreeBSD สำหรับการสร้างคิว
- Create Rule Script หมายถึง การสร้างคำสั่งแบบบรรทัดหรือ System Command เพื่อสั่งการทำงานของระบบปฏิบัติการ FreeBSD สำหรับการสร้างกฎต่างๆ

ส่วนของที่เก็บข้อมูล (Data store) ประกอบด้วย

- Pipe หมายถึง ที่เก็บข้อมูลของไปท์
- Queue หมายถึง ที่เก็บข้อมูลของคิว
- Rule หมายถึง ที่เก็บข้อมูลของกฎต่างๆ

3.2.3 การออกแบบโครงสร้างของระบบเว็บอินเทอร์เฟส

โครงสร้างของระบบโดประกอบด้วยส่วนสำคัญหลักดังแสดงในรูปที่ 3.12 ซึ่งมีรายละเอียดทั้งระบบดังนี้

Login คือการพิสูจน์ตัวต้นของผู้ควบคุมหรือผู้ดูแลระบบ

Home คือระบบหน้าจอหลักเมื่อผ่านการ Login เข้าระบบ

User Profile คือข้อมูลรายละเอียดเกี่ยวกับผู้ควบคุมหรือผู้ดูแลระบบ

- Create User สร้างผู้ควบคุมหรือผู้ดูแลระบบใหม่
- Edit User แก้ไขรายละเอียดผู้ควบคุมหรือผู้ดูแลระบบ
- Delete User ลบรายการผู้ควบคุมหรือผู้ดูแลระบบ
- Change Password เปลี่ยนรหัสผ่าน

System Control คือระบบควบคุมเกี่ยวกับระปฏิบัติการและข้อมูลทั่วไป

Network Interface คือรายละเอียดเกี่ยวกับเน็ตเวิร์กอินเทอร์เฟส

- Edit IP Address การแก้ไขรายละเอียดเกี่ยวกับเน็ตเวิร์กอินเทอร์เฟส
- Host คือข้อมูลรายการโฮสและเน็ตเวิร์กที่ต้องการควบคุม
- Add Host เพิ่มรายการโฮสและเน็ตเวิร์ก
- Edit Host แก้ไขรายการโฮสและเน็ตเวิร์ก
- Delete Host ลบรายการโฮสและเน็ตเวิร์ก

Traffic Control คือระบบควบคุมทราฟฟิกของทั้งระบบ

Pipe Configuration คือการจัดการค่าคอนฟิกของระบบไปท์

- Add Pipe การเพิ่มรายการไปท์
- Edit Pipe การแก้ไขรายการไปท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

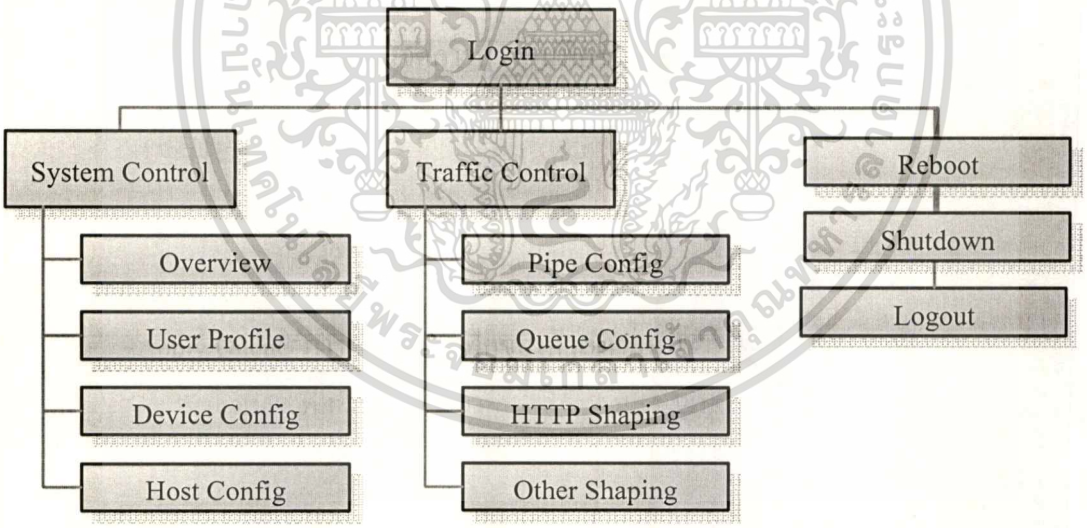
- Delete Pipe การลบรายการไปท์
- Show Pipe แสดงรายการไปท์
- Apply Pipe การส่งค่าคอนฟิกของไปท์เข้าระบบ

Queue Configuration คือการจัดการค่าคอนฟิกของระบบคิว

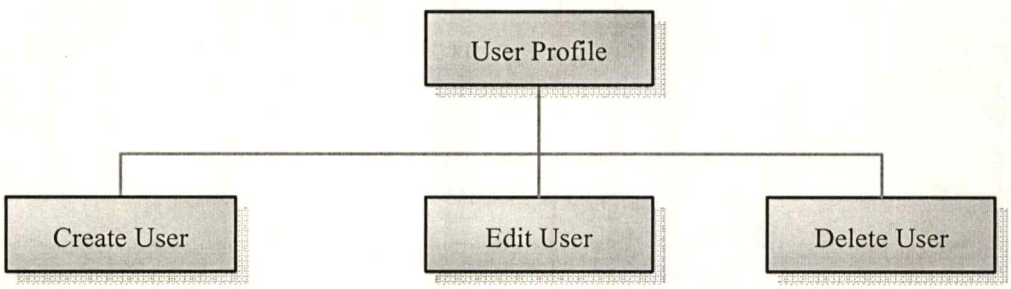
- Add Queue การเพิ่มรายการคิว
- Edit Queue การแก้ไขรายการคิว
- Delete Queue การลบรายการคิว
- Show Queue แสดงรายการคิว
- Apply Queue การส่งค่าคอนฟิกของคิวเข้าระบบ

Rule Configuration คือการจัดการค่าคอนฟิกของระบบกฎต่างๆ

- Add Rule การเพิ่มรายการกฎต่างๆ
- Edit Rule การแก้ไขรายการกฎต่างๆ
- Delete Rule การลบรายการกฎต่างๆ
- Show Rule แสดงรายการกฎต่างๆ
- Apply Rule การส่งค่าคอนฟิกของกฎต่างๆ เข้าระบบ

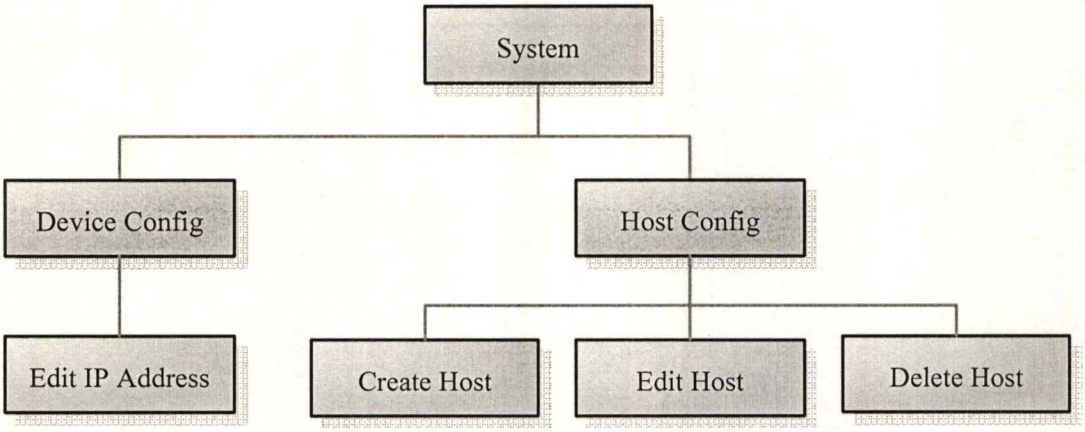


รูปที่ 3. 12 แสดงโครงสร้างโดยรวมของระบบ

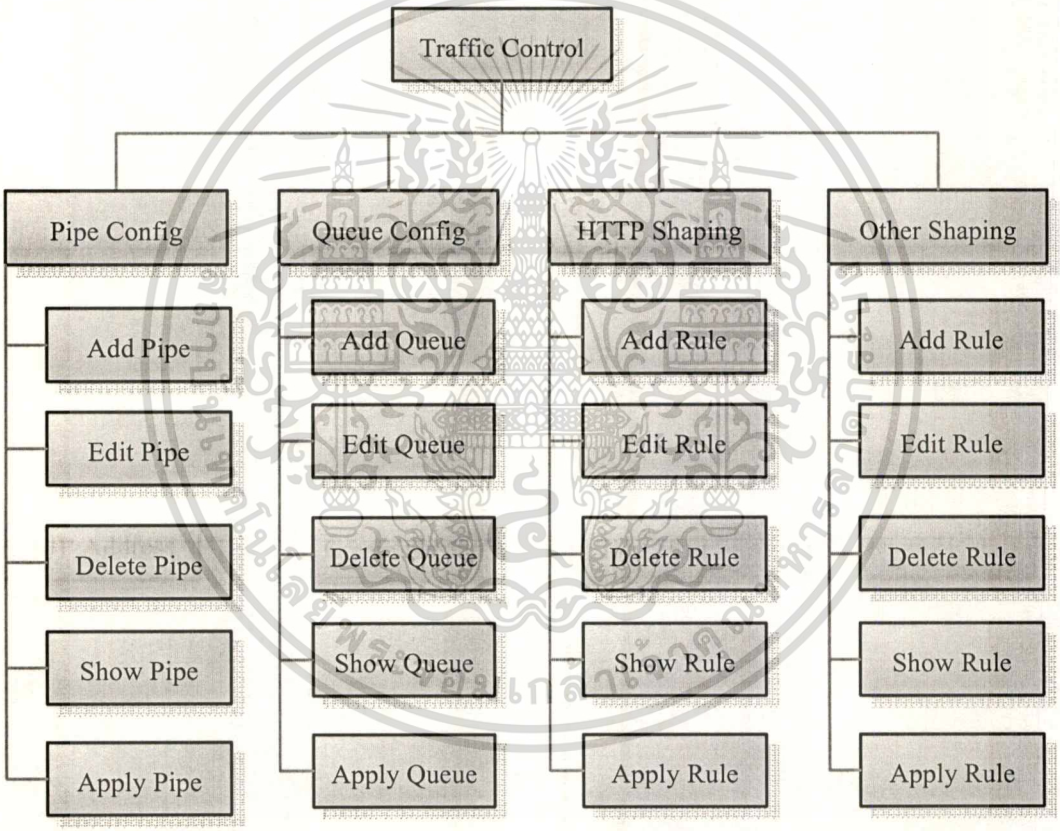


รูปที่ 3. 13 แสดงโครงสร้างของส่วนผู้ใช้งานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 14 แสดงโครงสร้างของระบบควบคุมหลัก



รูปที่ 3. 15 แสดงโครงสร้างของระบบควบคุมทราฟฟิก

3.2.4 การออกแบบฐานข้อมูล

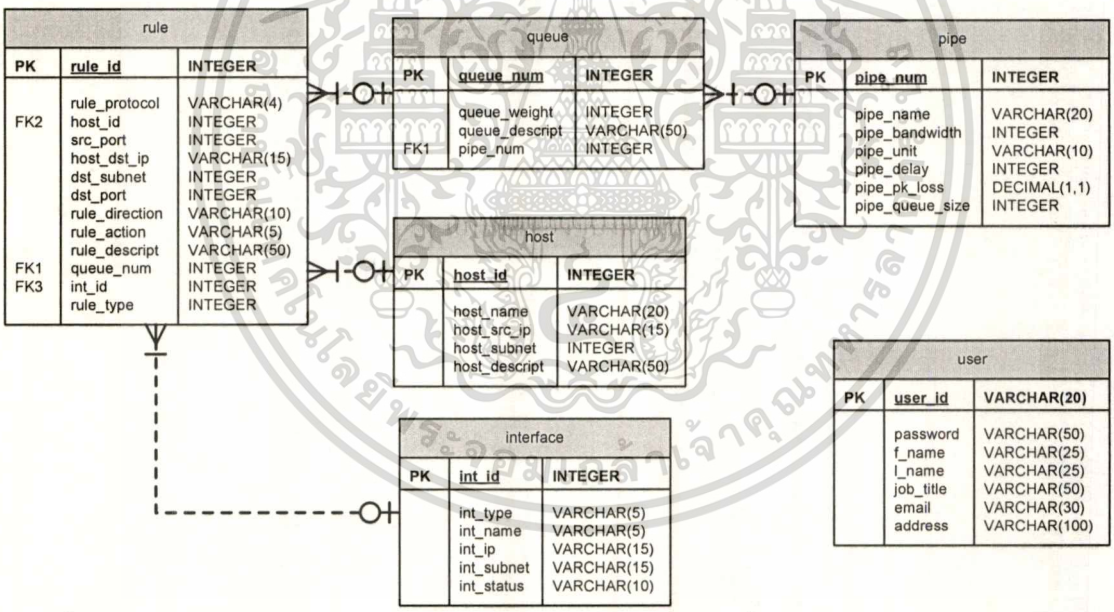
การออกแบบระบบฐานข้อสำหรับระบบควบคุมปริมาณการส่งผ่านข้อมูลโวยโปรโตคอล HTTP ก็เพื่อใช้สำหรับเก็บค่าคอนฟิกต่างๆ ที่สำคัญของระบบเพื่อให้เห็นรายละเอียดความสัมพันธ์ของฐานข้อมูลจะอาศัยแผนภาพอีอาร์ไดอะแกรมในการซึ่งมีรายละเอียดของความสัมพันธ์ของเอนทิตีต่างๆ ดังแสดงในตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 เอนทิตีของระบบเว็บอินเทอร์เฟส

ลำดับที่	เอนทิตี	คำอธิบายเอนทิตี
1	User	เอนทิตีของผู้ใช้งาน
2	Pipe	เอนทิตีของการบริหารจัดการไปท์
3	Queue	เอนทิตีของของการบริหารจัดการคิว
4	Rule	เอนทิตีของของการบริหารจัดการกฎต่างๆ
5	Host	เอนทิตีของโฮสและเครือข่าย
6	Interface	เอนทิตีของเน็ตเวิร์กอินเทอร์เฟส

จากตารางเอนทิตีของระบบควบคุมปริมาณการส่งผ่านข้อมูลโวยโปรโตคอล HTTP ตามตารางที่ 3.1 นำมาสร้างแผนภาพอีอาร์ไดอะแกรมเพื่อแสดงความสัมพันธ์ของเอนทิตีต่างดังแสดงในรูปที่ 3.16



รูปที่ 3. 16 อีอาร์ของระบบการควบคุมปริมาณการสื่อสารข้อมูลที่ส่งผ่านด้วยโปรโตคอล HTTP

สำหรับความสัมพันธ์ระหว่างเอนทิตีแต่ละตัวของระบบจะมีความสัมพันธ์กันดังนี้

- Rule กับ Host มีความสัมพันธ์กันแบบ M:1 หมายถึง รายการของโฮส 1 รายการสามารถที่จะนำไปใช้ได้ในหลายๆ กฎ แต่กฎสามารถมีรายการของโฮสได้เพียงรายการเดียวในแต่ละบทบาทหน้าที่ คือ Source หรือ Destination

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Rule กับ Interface มีความสัมพันธ์กันแบบ M:1 หมายถึง รายการของอินเทอร์เฟส 1 รายการสามารถมีอยู่ในรายการของกฎได้มากกว่า 1 กฎ แต่กฎสามารถมีรายการของอินเทอร์เฟสได้เพียงรายการเดียวเท่านั้น
- Rule กับ Queue มีความสัมพันธ์กันแบบ M:1 หมายถึง รายการของคิว 1 รายการสามารถใช้งานกับกฎได้หลายกฎ แต่กฎแต่ละข้อสามารถใช้งานหรืออ้างอิงรายการคิวได้เพียงรายการเดียว
- Queue กับ Pipe มีความสัมพันธ์กันแบบ M:1 หมายถึง รายการของไปท์ 1 รายการสามารถใช้งานหรือถูกอ้างอิงจากคิวได้หลายรายการ

3.2.5 พจนานุกรมข้อมูล

จากอีอาร์ไดอะแกรมที่ได้นั้นสามารถแปลงค่าเอนทิตีเป็นฐานข้อมูลเชิงสัมพันธ์โดยนำเสนอผ่านพจนานุกรมข้อมูลที่สามารถแสดงรายละเอียดต่างที่ต้องมีในฐานข้อมูลเพื่อการทำงานสมบูรณ์

ตารางที่ 3.2 รายการตารางของระบบฐานข้อมูลระบบเว็บอินเทอร์เฟส

ชื่อตาราง	คำอธิบายตาราง
Rule	ตารางที่เก็บข้อมูลของการกำหนดค่าของกฎ
Queue	ตารางที่เก็บข้อมูลการกำหนดค่าของคิว
Pipe	ตารางที่เก็บข้อมูลการกำหนดค่าของไปท์
Host	ตารางที่เก็บข้อมูลการกำหนดค่ารายการของโฮส
Interface	ตารางที่เก็บข้อมูลการกำหนดค่าให้อินเทอร์เฟส
User	ตารางที่เก็บข้อมูลผู้ใช้งานที่ควบคุมระบบ

จากแผนภาพอีอาร์ในรูปที่ 3.16 และตารางที่ 3.2 จะมีจำนวนตารางที่ต้องการใช้งานในฐานข้อมูลจำนวน 6 ตารางและมี 5 ตารางที่ต้องทำงานให้มีความสัมพันธ์กับเพื่อให้การทำงานเป็นไปตามความต้องการต้องทำการกำหนดคุณสมบัติต่างๆ ของแต่ละตารางเช่น ชื่อฟิลด์ ชนิดของฟิลด์ ขนาดของฟิลด์ และข้อมูลของฟิลด์ที่จำเป็นต้องใช้อ้างอิงถึงกันระหว่างตาราง ซึ่งมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ตาราง Rule

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาว	คีย์	ตารางอ้างอิง
rule_id	รหัสประจำกฎ	Integer	5	PK	
rule_protocol	โปรโตคอลที่ใช้	Varchar	4		
host_id	รหัสเครื่องต้นทาง	Integer	5	FK	**Host**
src_port	พอร์ตต้นทาง	Integer	5		
host_dst_ip	รหัสเครื่องปลายทาง	Varchar	15		
dst_port	พอร์ตปลายทาง	Integer	5		
rule_direction	ทิศทางของกฎ	Varchar	10		
rule_action	ผลการกระทำของกฎ	Varchar	5		
queue_num	รหัสประจำคิว	Integer	5	FK	**Queue**
int_id	รหัสประอินเทอร์เฟส	Integer	5	FK	**Interface**
rule_type	ชนิดของกฎที่จัดเก็บ	Integer	5		
rule_descript	คำอธิบายของกฎ	Varchar	50		

ตารางที่ 3.4 ตาราง Queue

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาว	คีย์	ตารางอ้างอิง
queue_num	รหัสประจำคิว	Integer	5	PK	
queue_weight	ค่าน้ำหนักของคิว	Integer	3		
pipe_num	รหัสของไปท์ที่ใช้งาน	Integer	5	FK	**Pipe**
queue_descript	คำอธิบายเกี่ยวกับคิว	Varchar	50		

ตารางที่ 3.5 ตาราง Host

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาว	คีย์	ตารางอ้างอิง
host_id	รหัสประจำโฮส/เครือข่าย	Integer	5	PK	
host_name	ชื่อโฮส/เครือข่าย	Varchar	20		
host_src_ip	หมายเลขเครือข่ายไอพี	Varchar	15		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาว	คีย์	ตารางอ้างอิง
host_subnet	หมายเลขซับเน็ต	Varchar	15		
host_descript	คำอธิบายโฮส/เครือข่าย	Varchar	50		

ตารางที่ 3.6 ตาราง Pipe

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาว	คีย์	ตารางอ้างอิง
pipe_num	รหัสประจำไปท์	Integer	5	PK	
pipe_name	ชื่อไปท์	Varchar	20		
pipe_bandwidth	ขนาดแบนด์วิดท์ของไปท์	Integer	5		
pipe_unit	หน่วยของแบนด์วิดท์	Varchar	10		
pipe_delay	ดีเลย์ของไปท์	Integer	10		
pipe_pk_loss	อัตราแพ็กเก็ตสูญหาย	Decimal	1,1		
pipe_queue_size	ขนาดของคิวในไปท์	Integer	10		

ตารางที่ 3.7 ตาราง Interface

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาว	คีย์	ตารางอ้างอิง
int_id	รหัสอินเทอร์เฟส	Integer	5	PK	
int_name	ชื่ออินเทอร์เฟส	Varchar	5		
int_type	ชนิดของอินเทอร์เฟส	Varchar	5		
int_ip	ไอพีของอินเทอร์เฟส	Varchar	15		
int_subnet	ไอพีซับเน็ตมาร์ค	Varchar	15		
int_status	สถานะของอินเทอร์เฟส	Varchar	10		

ตารางที่ 3.8 ตาราง User

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาว	คีย์	ตารางอ้างอิง
user_id	ชื่อรหัสผู้ใช้งาน	Varchar	20	PK	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	ความยาว	คีย์	ตารางอ้างอิง
passwors	รหัสผ่านผู้ใช้งาน	Varchar	50		
f_name	ชื่อ ตัว	Varchar	25		
l_name	ชื่อ สกุล	Varchar	25		
job_title	ชื่อตำแหน่งปัจจุบัน	Varchar	50		
email	ที่อยู่อีเมลย์ของผู้ใช้งาน	Varchar	30		
address	ที่อยู่ผู้ใช้งาน	Varchar	100		

3.2.6 การออกแบบส่วนเชื่อมต่อกับผู้ใช้งาน

การออกแบบส่วนเชื่อมต่อกับผู้ใช้งานถือเป็นส่วนสำคัญที่จะทำให้ผู้ใช้สามารถใช้ระบบได้อย่างถูกต้องและมีประสิทธิภาพ วัตถุประสงค์หลักของการทำงานในส่วนนี้คือการติดต่อประสานงานกันระหว่างผู้ใช้งานกับระบบงานที่ใช้ควบคุมการทำงานของระบบควบคุมปริมาณการสื่อสารที่ใช้การส่งผ่านข้อมูลด้วยโปรโตคอล HTTP ระบบที่พัฒนาขึ้นนี้จะเป็นลักษณะของเว็บอินเทอร์เฟสที่ผู้ใช้งานเข้าใจง่ายเพียงแต่ทราบขั้นตอนการทำงานของระบบก็สามารถใช้งานได้ตัวระบบออกแบบและพัฒนาด้วยโปรแกรมภาษา PHP ที่มีการทำงานร่วมกับดาต้าเบส MySQL และโปรแกรมจัดการด้านเว็บเซิร์ฟเวอร์คือ Apache และระบบงานทั้งหมดทำงานอยู่บนระบบปฏิบัติการ FreeBSD

การออกแบบหน้าจอของระบบจะแบ่งการทำงานเป็นชุดกลุ่มของคำสั่งและกระบวนการทำงานที่เกี่ยวเนื่องกันไว้เป็นหมวดหมู่ดังแสดงรายละเอียดตามตารางที่ 3.9

ตารางที่ 3.9 เมนูการทำงานของระบบเว็บอินเทอร์เฟส

เมนูหลัก	ชื่อโปรแกรม
1. System Control	Overview User Profile - Create User - Edit User - Delete User - Change Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 (ต่อ)

เมนูหลัก	ชื่อโปรแกรม
	Network Interface - Edit Network Interface Host - Create Host Information - Edit Host Information - Delete Host Information - View Host Information
2. Traffic Control	Pipe Config - Add Pipe Information - Edit Pipe Information - Delete Pipe Information - Show Pipe Information - Apply Pipe Information Queue Config - Add Queue Information - Edit Queue Information - Delete Queue Information - Show Queue Information - Apply Queue Information HTTP Shaping - Add Rule Information - Edit Rule Information - Delete Rule Information - Show Rule Information - Apply Rule Information Other Shaping - Add Rule Information - Edit Rule Information - Delete Rule Information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

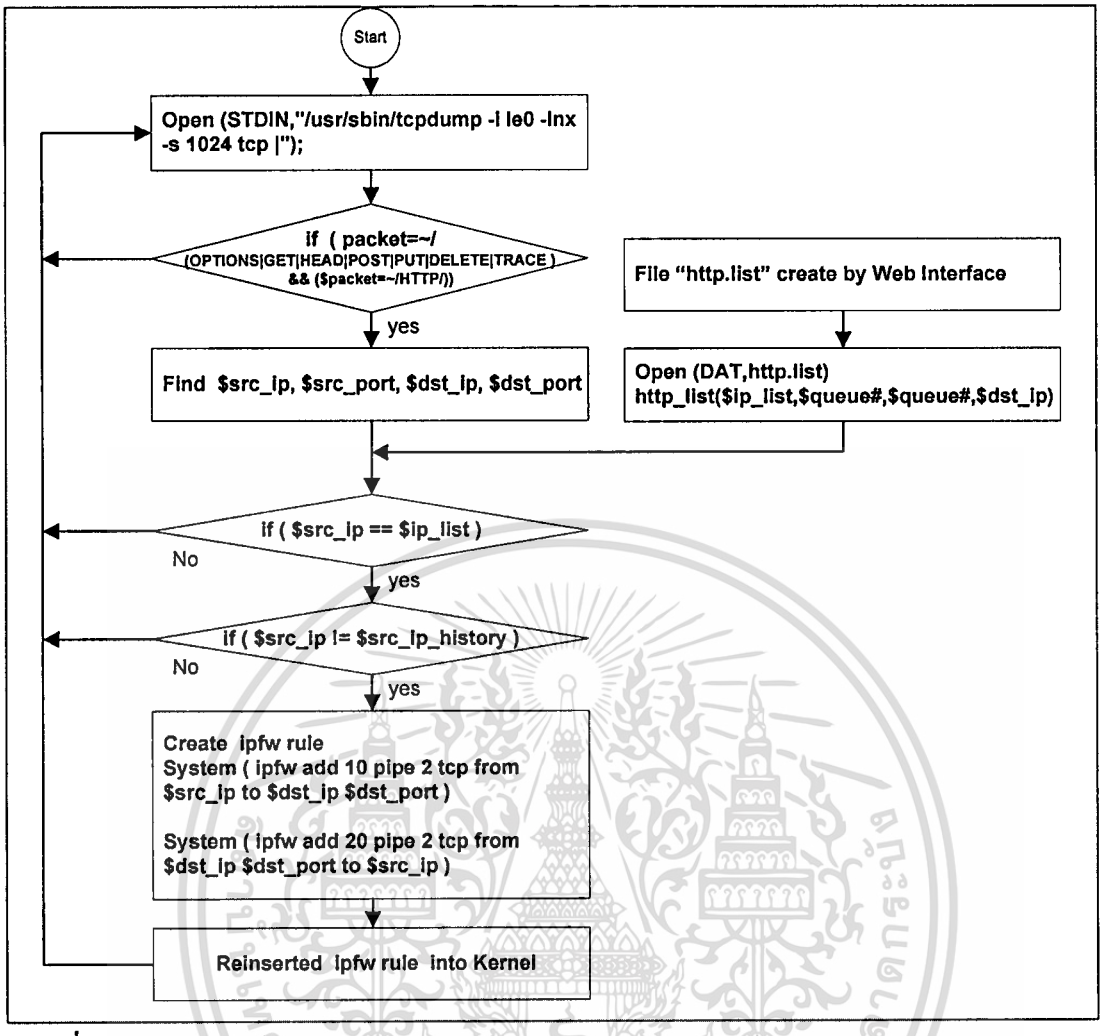
ตารางที่ 3.9 (ต่อ)

เมนูหลัก	ชื่อโปรแกรม
	- Show Rule Information - Apply Rule Information
3. Bandwidth	WAN Traffic Statistic - Daily - Weekly - Monthly - Yearly
4. Reboot …	Reboot system
5. Shutdown …	Shutdown system
6. Logout	System Logout

3.3 การวิเคราะห์และออกแบบระบบตรวจจับโปรโตคอล HTTP (http_sniffer)

ระบบตรวจจับการสื่อสารหรือส่งผ่านข้อมูลของโปรโตคอล HTTP มีหน้าที่หลักคือค้นหาว่าการสื่อสารที่ส่งผ่านในระบบเครือข่ายมีการสื่อสารของโปรโตคอล HTTP หรือไม่ เมื่อตรวจพบต้องหาค่าของไอพีแอดเดรสพร้อมพอร์ตบริการของเครื่องต้นทางและปลายทางออกมาเพื่อใช้เป็นองค์ประกอบในการสร้างกฎการควบคุมแบนด์วิทของโปรโตคอลนี้ จากรูปที่ 3.17 แสดงโฟลว์ชาร์ตการทำงานของระบบทั้งหมดซึ่งมีขั้นตอนดังนี้คือ เมื่อเริ่มทำงานระบบจะรับผลของข้อมูลที่ตรวจจับได้จากโปรแกรม tcpdump โดยอาศัยการกำหนด การทำงานของ STDOUT ที่เป็นจอภาพของโปรแกรม tcpdump เป็นข้อมูลขาเข้า (STDIN) ของระบบที่พัฒนาเมื่อได้ข้อมูลของแพ็กเก็ตแล้วนำข้อมูลทั้งหมดมาตรวจสอบหาค่าข้อความ OPTIONS หรือ GET หรือ HEAD หรือ POST หรือ PUT หรือ DELETE หรือ TRACE ซึ่งข้อความทั้งหมดเป็นชื่อ Method หรือวิธีการรองขอข้อมูลของโปรโตคอล HTTP และข้อความ HTTP/ ซึ่งเป็นส่วนของเวอร์ชั่น โดยทั้งหมดรวมอยู่ในข้อมูลของ HTTP request message ดังแสดงในรูปที่ 2.4 เมื่อตรวจสอบเงื่อนไขว่ามีองค์ประกอบทั้งสองส่วนครบก็ประมวลผลหาค่าของไอพีแอดเดรสพร้อมพอร์ตของเครื่องต้นทางและปลายทางแล้วนำค่าไอพีแอดเดรสของเครื่องต้นทางไปเปรียบเทียบกับข้อมูลของไฟท์ http.list ที่สร้างขึ้นจากระบบเว็บอินเทอร์เฟส ซึ่งมีรายละเอียดข้อมูลภายในคือ Src_IP_Address, xmit_queue,revc_queue ตัวอย่างเช่น 192.168.10.11,1,1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 17 โฟลว์ชาร์ตการทำงานของระบบตรวจจับการส่งผ่านข้อมูลของโปรโตคอล HTTP

ถ้าผลการเปรียบเทียบไม่ตรงแสดงว่าระบบไม่ต้องการควบคุมแบนด์วิทของเครื่องต้นทางเครื่องนี้ แต่ถ้าตรงกันก็ทำการตรวจสอบเปรียบเทียบกับข้อมูลเดิมในอดีตว่าเคยควบคุมไอพีแอดเดรสต้นทางเครื่องนี้หรือยัง ถ้ายังก็ทำการสร้างกฎควบคุมการสื่อสารของการติดต่อชุดนี้ การตรวจสอบนี้ก็เพื่อไม่ให้มีการสร้างกฎซ้ำซ้อนด้วยข้อเท็จจริงการสื่อสารระหว่างเครื่องสองเครื่องมีการส่งข้อมูลติดต่อกันมากกว่าหนึ่งแพ็กเก็ต เมื่อโปรแกรมสร้างชุดคำสั่งของการควบคุมโดยอาศัยข้อมูลที่ตรวจจับได้ในเรื่องไอพีแอดเดรสพร้อมพอร์ตของเครื่องต้นทางปลายทางและข้อมูลหมายเลข queue ทั้งขารับและขาส่ง ก็จะสั่งงานตรงเข้าระบบด้วยชุดคำสั่งของ IPFW โดยไม่มีการสร้างชุดคำสั่งให้ชุดคำสั่งใน crontab เรียกใช้งานเหมือนในระบบเว็บอินเทอร์เฟส เมื่อจบการทำงานระบบก็จะวนการทำงานซ้ำรอบเดิมเช่นนี้เลยไปจนครบ 30 นาทีระบบจะสั่งผ่าน crontab ให้มีการ restart การทำงานของโปรแกรมเพื่อเป็นการลบค่ากฎเก่าที่ค้างในระบบออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การพัฒนาและทดสอบระบบ

การพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้การส่งผ่านข้อมูลด้วยโปรโตคอล HTTP เป็นการพัฒนาระบบขึ้นใหม่ทั้งหมดและได้คำนึงถึงการใช้งานของผู้ใช้ระบบต้องสามารถเข้าถึงการทำงานของระบบได้โดยง่ายและตัวระบบงานต้องนำไปติดตั้งร่วมกับตัวอุปกรณ์ที่ทำหน้าที่เป็นเกตเวย์ของระบบซึ่งส่วนมากถูกต่อเชื่อมใช้งานในระบบเครือข่ายท้องถิ่นแบบ LAN ทำให้การเข้าถึงระบบสามารถใช้เครื่องลูกข่ายเครื่องใดๆ ในระบบเครือข่ายเข้าบริหารจัดการระบบได้โดยไม่จำเป็นต้องติดตั้งระบบซอฟต์แวร์ที่เครื่องลูกข่ายที่ใช้ในการควบคุมระบบ

4.1 สถาปัตยกรรมของระบบ

สถาปัตยกรรมของระบบควบคุมปริมาณการสื่อสารที่ใช้การส่งผ่านข้อมูลด้วยโปรโตคอล HTTP มีแนวทางในการติดตั้งใช้งานระบบโดยมีองค์ประกอบที่สำคัญคือ ระบบฮาร์ดแวร์และซอฟต์แวร์ที่ใช้สำหรับการพัฒนาระบบ และระบบการเชื่อมต่อเครือข่าย โดยมีรายละเอียดดังดังต่อไปนี้

ตารางที่ 4.1 องค์ประกอบของระบบฮาร์ดแวร์และซอฟต์แวร์ที่ใช้สำหรับการพัฒนาระบบ

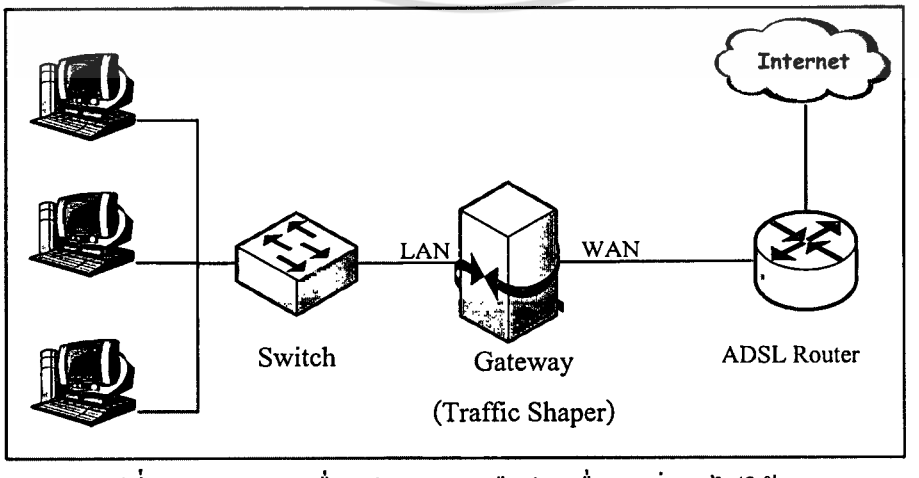
เครื่องคอมพิวเตอร์แม่ข่ายสำหรับ Gateway จำนวน 1 เครื่อง	
หน่วยประมวลผล	Intel Pentium 4, Clock speed ไม่น้อยกว่า 3.0 GHz
หน่วยความจำหลัก	ขนาดไม่ต่ำกว่า 1GB
ฮาร์ดดิสก์	ความจุไม่ต่ำกว่า 40GB
Network Interface	10/100 Ethernet Interface (RJ-45) จำนวน 2 ชุด
เครื่องคอมพิวเตอร์สำหรับผู้ใช้งาน (PC Client) จำนวน 1 เครื่อง	
หน่วยประมวลผล	Intel Pentium 4, Clock speed ไม่น้อยกว่า 3.0 GHz
หน่วยความจำหลัก	ขนาดไม่ต่ำกว่า 1GB
ฮาร์ดดิสก์	ความจุไม่ต่ำกว่า 40GB
Network Interface	10/100 Ethernet Interface (R-J45) จำนวน 1 ชุด
จอภาพแสดงผล	จอภาพ LCD ขนาดไม่ต่ำกว่า 14 นิ้ว จำนวน 1 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 (ต่อ)

ซอฟต์แวร์	
ระบบปฏิบัติการ	FreeBSD 7.0 สำหรับเครื่องแม่ข่าย (Gateway) Windows XP Professional สำหรับเครื่องลูกข่าย
ระบบจัดการฐานข้อมูล	MySQL 5.0
ระบบบริหารจัดการผ่านเว็บ	สำหรับเครื่องแม่ข่าย (Gateway) - Apache Web Server 2.2 - PHP 5.1 - PERL 5.0 - ชุดคำสั่ง IPFW - โปรแกรมเก็บสถิติการใช้งานเครือข่าย RRD Tools สำหรับเครื่องคอมพิวเตอร์ของผู้ใช้งาน - โปรแกรมเว็บบราวเซอร์ (IE 7) - โปรแกรม Edit Plus (สำหรับการพัฒนาระบบ) - โปรแกรม Dreamweaver (สำหรับการพัฒนาระบบ)

การต่อเชื่อมต่อระบบเครือข่ายเมื่อนำระบบไปใช้งานจริงจะมีลักษณะดังแสดงในรูปที่ 4.1 โดยส่วนที่พัฒนาในครั้งนี้คืออุปกรณ์ที่ทำหน้าที่เป็น Gateway (Traffic Shaper) และสำหรับผู้ควบคุมระบบสามารถเข้าถึงระบบด้วยเครื่องคอมพิวเตอร์เครื่องใดๆ ในระบบเครือข่ายก็สามารถทำได้โดยใช้โปรแกรมเว็บบราวเซอร์ใส่ URL ติดต่อไปที่หมายเลขไอพีที่ระบุว่าเป็นเกตเวย์ของเครื่องลูกข่ายหรือไอพีแอดเดรสของอินเทอร์เฟส LAN ของเครื่อง Gateway (Traffic Shaper)



รูปที่ 4.1 แสดงการเชื่อมต่อระบบเครือข่ายเมื่อนำระบบไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การพัฒนาระบบ

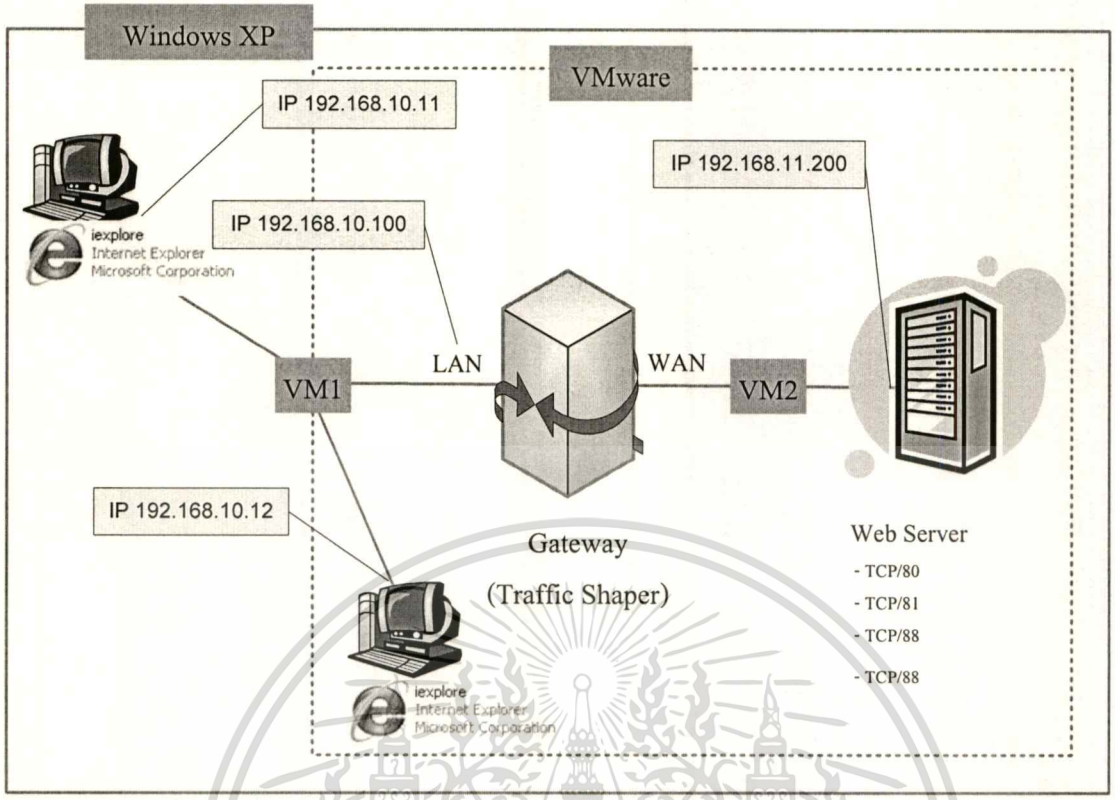
การพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้การส่งผ่านข้อมูลด้วยโปรโตคอล HTTP ได้ใช้โปรแกรมภาษา PHPในการพัฒนาระบบเว็บอินเทอร์เฟสและใช้ระบบจัดการฐานข้อมูล MySQL สำหรับเก็บค่าคอนฟิกและรายละเอียดที่สำคัญของระบบ โดยระบบที่พัฒนาขึ้นมีความสามารถหรือหน้าที่หลักๆ ดังนี้

1 ระบบสามารถป้อนข้อมูล การสร้าง แก้ไขและลบข้อมูลทะเบียนผู้ใช้ระบบได้
2 ระบบสามารถป้อนข้อมูล การสร้าง แก้ไขและลบข้อมูลเครื่องคอมพิวเตอร์หรือชุดไอพีของระบบเครือข่ายที่ใช้สำหรับติดต่อด้วยไม่ว่าจะติดต่อในลักษณะต้นทางหรือปลายทาง
3 ระบบสามารถป้อนข้อมูล การสร้าง แก้ไขข้อมูลไอพีแอดเดรสของเครื่อง Gateway ได้
4 ระบบสามารถป้อนข้อมูล การสร้าง แก้ไขและลบข้อมูลการบริหารจัดการทราฟฟิกของระบบเครือข่ายในเรื่องการกำหนด คิว ไปท์ และกฎต่างๆ ในระบบ ได้อย่างถูกต้องตามหลักไวยากรณ์ของคำสั่งนั้นๆ
5 ระบบมีความสามารถในการบริหารจัดการหรือกำหนดการใช้งานแบนด์วิดท์ของเครื่องลูกข่ายเป็นรายเครื่องหรือกลุ่มเครือข่ายได้ และยังสามารถกำหนดขนาดแบนด์วิดท์สำหรับผู้ใช้แต่ละกลุ่มได้อย่างอิสระต่อกัน
6 ระบบต้องมีความสามารถในการกรองแพ็กเก็ตเฉพาะข้อมูลของโปรโตคอล HTTP และสามารถจัดขนาดแบนด์วิดท์ได้

4.3 การทดสอบระบบ

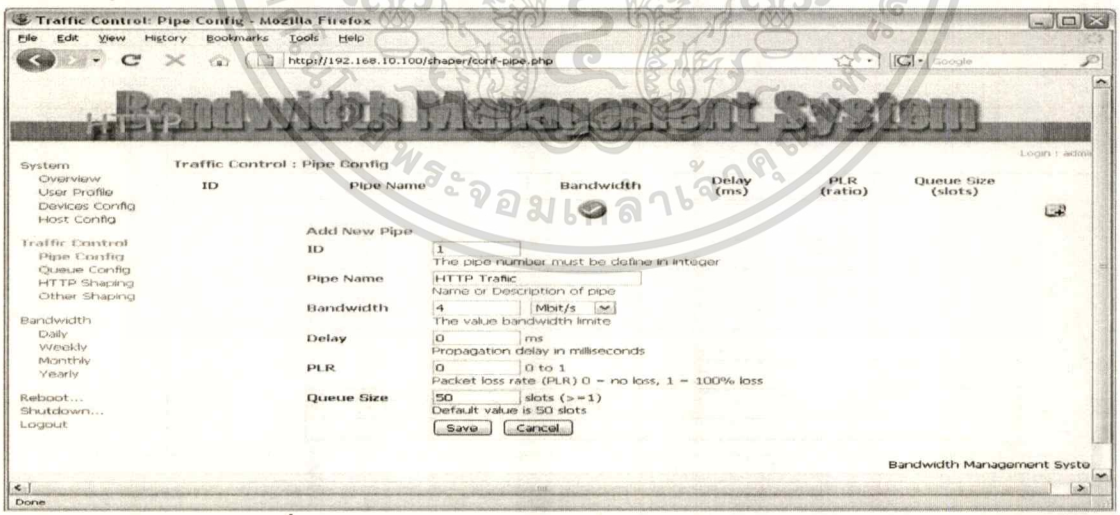
การทดสอบระบบควบคุมปริมาณการสื่อสารที่ใช้การส่งผ่านข้อมูลด้วยโปรโตคอล HTTP ได้จำลองการทำงานของระบบโดยอาศัยซอฟต์แวร์ VMware Workstation สร้างระบบเครือข่ายจำลองเสมือนจริงโดยมีองค์ประกอบดังรูปที่ 4.2 เพื่อทดสอบการใช้งานระบบเสมือนจริงในการติดต่อใช้งานกับเครื่องเว็บเซอร์ฟเวอร์ ผ่านพอร์ตบริการ TCP/80, TCP/81 และ TCP/88 รวมทั้งทดสอบการใช้งานบริการดาวน์โหลดข้อมูล (HTTP) ซึ่งใช้พอร์ตบริการ TCP/88 ในการทดสอบครั้งนี้กำหนดให้ pipe ที่ใช้ในการดาวน์โหลดข้อมูล มีขนาด 4Mbit/s และทดสอบดาวน์โหลดข้อมูลว่าใช้แบนด์วิทสูงสุดเท่าที่กำหนดหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



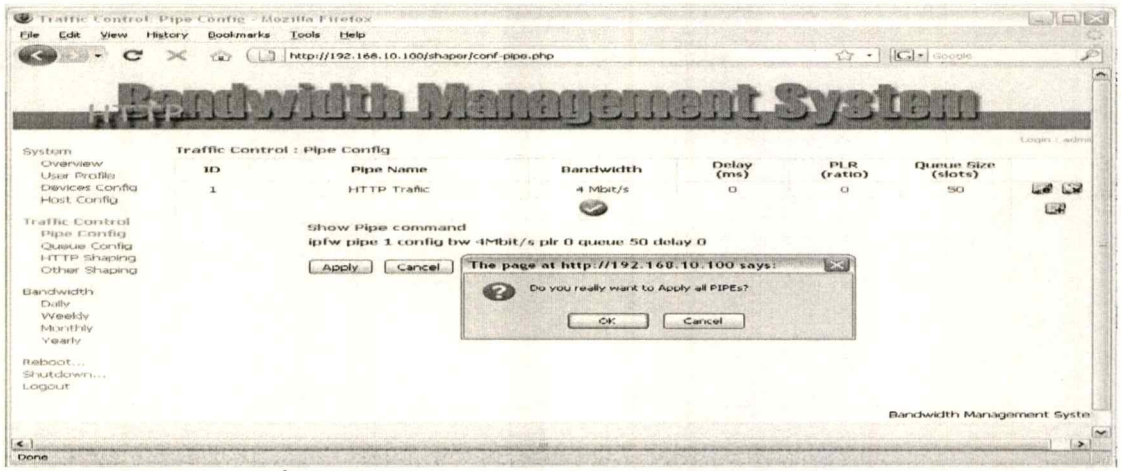
รูปที่ 4.2 แสดงการเชื่อมต่อระบบเครือข่ายสำหรับการทดสอบการทำงานจริง

ขั้นตอนที่ 1 กำหนดหรือสร้าง pipe หมายเลข 1 ให้มีขนาดของแบนด์วิทเป็น 4Mbit/s ดังแสดงในรูปที่ 4.3 และ รูปที่ 4.4



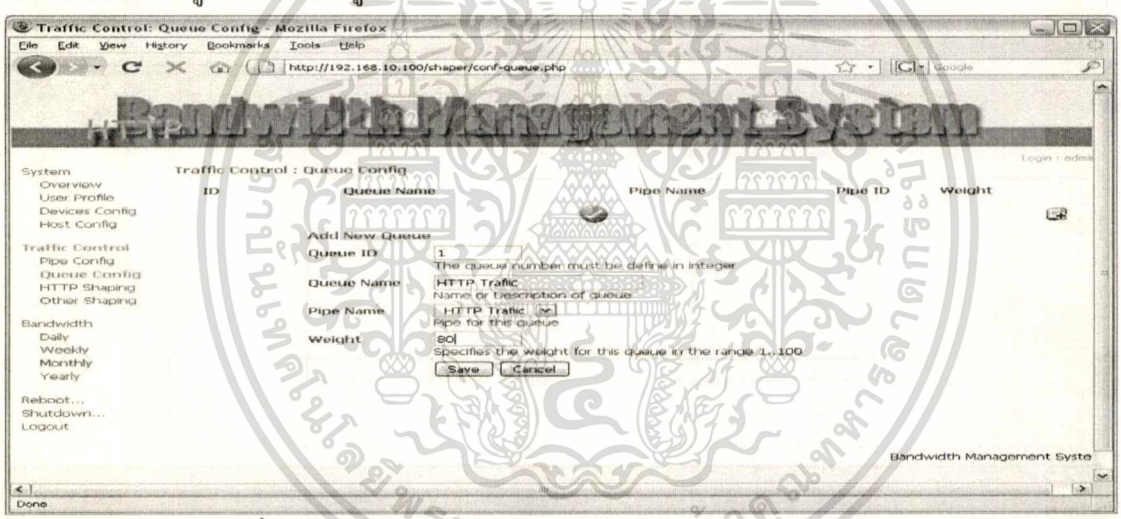
รูปที่ 4.3 แสดงการสร้าง pipe ทีมีขนาดแบนด์วิท 4Mbit/s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

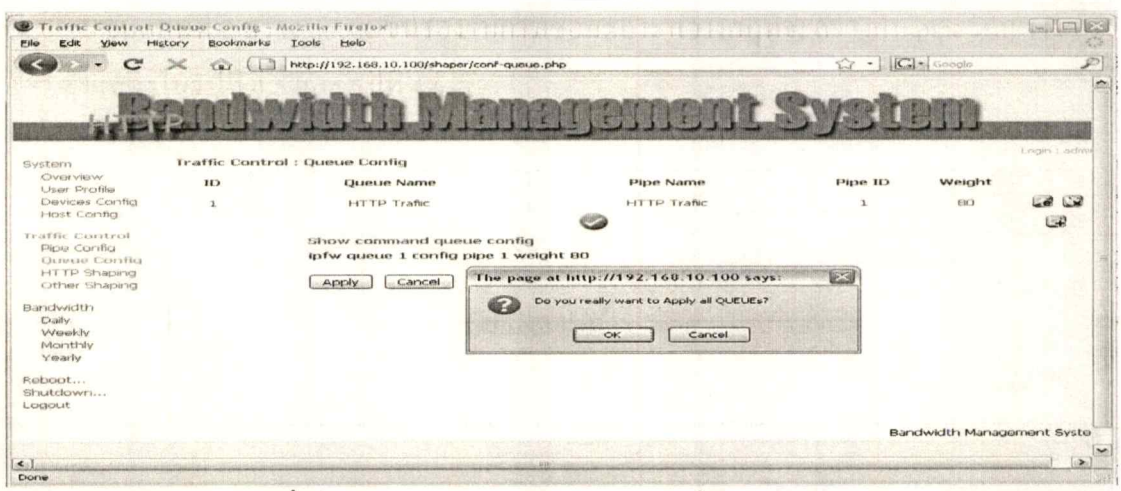


รูปที่ 4.4 แสดงการApply ข้อมูล pipe ที่สร้างไปใช้งานจริง

ขั้นตอนที่ 2 สร้าง queue ขึ้นรองรับการเชื่อมต่อระหว่างกฎที่จะสร้างกับ pipe ที่สร้างไว้แล้ว ดังแสดงในรูปที่ 4.5 และ รูปที่ 4.6



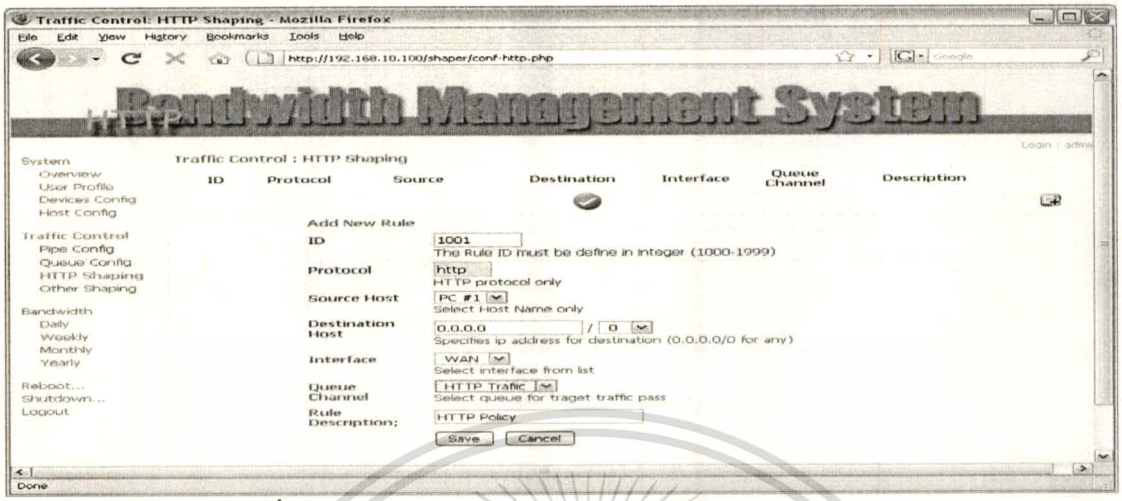
รูปที่ 4.5 แสดงการสร้าง queue ทีเชื่อมต่อการทำงานกับ pipe 1



รูปรูปที่ 4.6 แสดงการApply ข้อมูล queue ที่สร้างไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

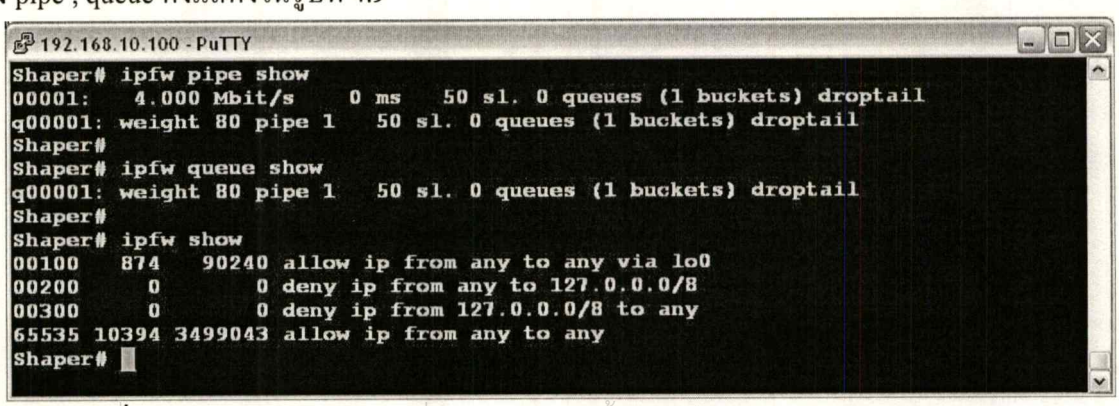
ขั้นตอนที่ 3 สร้างกฎของการ Shaping โปรโตคอล HTTP ดังแสดงในรูปที่ 4.7 และ รูปที่ 4.8



รูปที่ 4.7 แสดงการสร้างกฎการ Shaping โปรโตคอล HTTP

รูปที่ 4.8 แสดงการApply ข้อมูลการ Shaping ที่สร้างไปใช้งานจริง

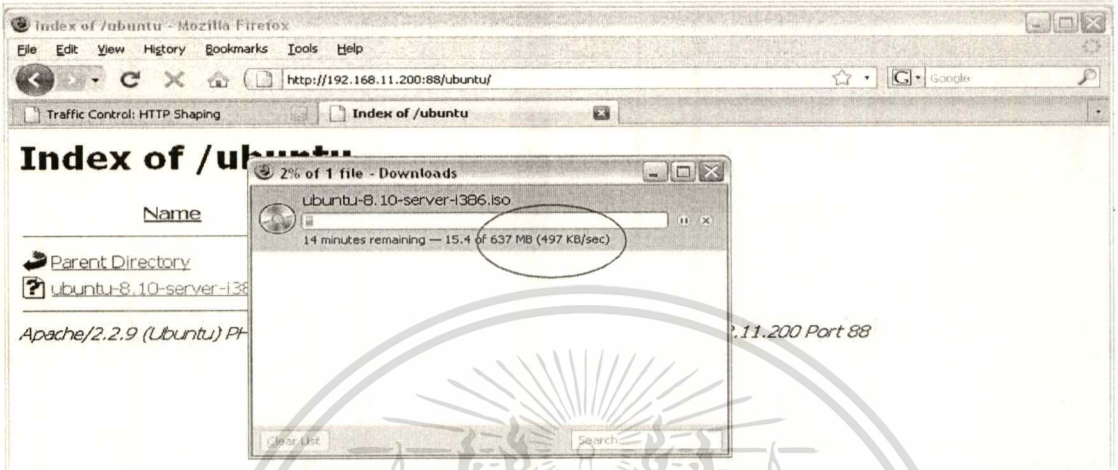
ขั้นตอนที่ 4 ตรวจสอบการทำงานของระบบว่ามีการกำหนดค่าต่างๆ ตามที่กำหนดไว้ในเรื่องของ pipe , queue ดังแสดงในรูปที่ 4.9



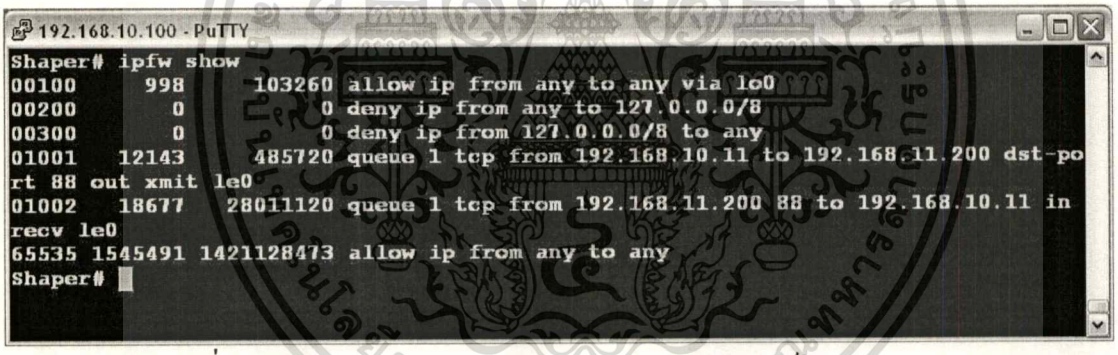
รูปที่ 4.9 การตรวจสอบการทำงานของระบบก่อนดาวน์โหลดทดสอบ

เอกสารนี้เป็นเอกสาร[illegible]ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

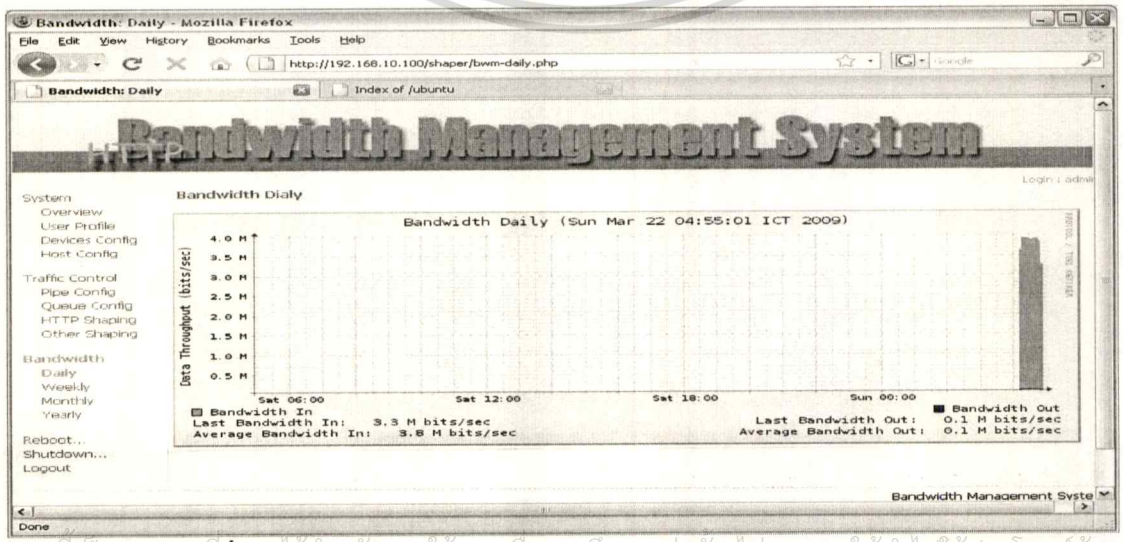
ขั้นตอนที่ 5 ทดสอบการดาวน์โหลดข้อมูลด้วยโปรโตคอล HTTP ดังแสดงในรูปที่ 4.10 ผลการดาวน์โหลดได้ความเร็ว 497 KB/sec หรือประมาณ 3.98Mbit/s และเมื่อตรวจสอบคำสั่งของ IPFW ได้ผลดังแสดงในรูปที่ 4.11 คือมีการเพิ่มกฎการควบคุมให้กับระบบ และรูปที่ 4.12 แสดงกราฟสถิติข้อมูลเครือข่ายหลังการดาวน์โหลดข้อมูล สามารถอ่านค่าได้เฉลี่ย 3.9 MBits/sec



รูปที่ 4.10 แสดงการดาวน์โหลดข้อมูลด้วยโปรโตคอล HTTP



รูปที่ 4.11 การตรวจสอบการทำงานของระบบหลังดาวน์โหลดทดสอบ



รูปที่ 4.12 แสดงกราฟสถิติข้อมูลเครือข่ายหลังการดาวน์โหลดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุปโครงการ

การนำระบบเทคโนโลยีสารสนเทศเข้ามาช่วยในการปฏิบัติงานในปัจจุบันถือเป็นเรื่องจำเป็นอย่างยิ่งเพราะระบบเทคโนโลยีสารสนเทศและระบบเครือข่ายอินเทอร์เน็ตช่วยให้หลายหน่วยงานสามารถดำเนินธุรกิจได้อย่างราบรื่น โครงการพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้การส่งผ่านข้อมูลด้วยโปรโตคอล HTTP ที่จัดทำขึ้นโดยอาศัยการทำงานของระบบปฏิบัติการ FreeBSD เป็นหลัก ซึ่งเป็นระบบที่มีกลไกการควบคุมการใช้งานด้านระบบเครือข่ายในเรื่องการบริหารจัดการแบนด์วิดท์ ด้วยชุดคำสั่งของ IPFW และการทำงานของ Dummynet ที่มีความสามารถในการกำหนดปริมาณแบนด์วิดท์ของการสื่อสารของเครื่องลูกข่ายได้เป็นรายเครื่องและเลือกประเภทบริการที่ต้องการจะควบคุมหรือจัดการ แต่ประเภทบริการที่ IPFW และ Dummynrt สามารถทำได้เป็นบริการที่บริหารจัดการผ่านเน็ตเวิร์กเลเยอร์ในลำดับชั้นที่ 4 คือ Transport Layer คือควบคุมบริการด้วยการแบ่งแยกจากพอร์ตบริการเช่น เว็บพอร์ต TCP/80 ระบบเมล์พอร์ต TCP/25 แต่ปัจจุบันพอร์ตบริการเหล่านี้ยังสามารถสร้างบริการอื่นๆ ขึ้นมาใช้งานได้ทำให้เกิดการสูญเสียแบนด์วิดท์ของระบบเครือข่ายไปในส่วนที่ไม่จำเป็น โครงการนี้จึงพัฒนาระบบการตรวจสอบการส่งผ่านข้อมูลที่คอยตรวจจับโปรโตคอล HTTP มาทำงานร่วมกับ IPFW และ Dummynet ในการควบคุมแพ็กเก็ตของข้อมูลในชั้น Application Layer พร้อมกับควบคุมขนาดของแบนด์วิดท์ของ Application Protocol ที่สนใจ โครงการนี้ได้พัฒนาขึ้นโดยอาศัยพื้นฐานการทำงานของโปรแกรม tcpdump ที่มีความสามารถในการตรวจจับข้อมูลที่มีการส่งผ่านในระบบเครือข่ายให้สามารถทำงานร่วมกับ IPFW ของระบบปฏิบัติการ FreeBSD โปรแกรมที่พัฒนาจะกรองผลการตรวจจับของ tcpdump เฉพาะส่วนของโปรโตคอล HTTP นำมาหาค่าของหมายเลขไอพีและหมายเลขพอร์ต ต้นทางและปลายทางของคู่การติดต่อสื่อสารเพื่อนำไปปรับปรุงในกฎของ IPFW ในส่วนของไปท์ที่สร้างขึ้นโดยกำหนดขนาดของแบนด์วิดท์ที่ต้องการควบคุม ส่งผลให้เราสามารถควบคุมทราฟฟิกของการติดต่อสื่อสารก่อนที่แพ็กเก็ตข้อมูลของโปรโตคอล HTTP จะถูกส่งผ่านออกจากชุดเกตเวย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

โครงการพัฒนาระบบควบคุมปริมาณการสื่อสารที่ใช้การส่งผ่านข้อมูลด้วยโปรโตคอล HTTP เป็นระบบที่สร้างขึ้นด้วยความต้องการที่จะควบคุมการใช้งานแบนด์วิดท์ให้เหมาะสมกับระบบงานจริงหรือความต้องการกาใช้งานจริง ซึ่งโครงการนี้สนใจเฉพาะการควบคุมทราฟฟิกของโปรโตคอล HTTP ซึ่งความเป็นจริงมีประเภทบริการที่เปิดให้บริการมากมาย จึงหวังว่าจะมีผู้อาศัยแนวคิดนี้ในการจัดทำระบบ Traffic Shaper ที่สามารถใช้งานได้จริงกับเครือข่ายขนาดใหญ่โดยการพัฒนาซอฟต์แวร์จากระบบซอฟต์แวร์ประเภท Open Source เพื่อเป็นการส่งเสริมการใช้งานซอฟต์แวร์ประเภทนี้อันจะส่งผลให้หน่วยงานต่างๆ ประหยัดงบประมาณในการจัดหาอุปกรณ์ประเภท Traffic Shaper และได้ใช้งานระบบเครือข่ายได้ตรงความต้องการที่แท้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

วิรินทร์ เมฆประดิษฐสิน. 2547. คัมภีร์ระบบเครือข่ายแบบฉบับอาจารย์วิรินทร์ เล่ม 1. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

Basic Traffic Shaping with dummynet. [Online]. Available URL: http://www.afp548.com/article.php?story=20060214081244545

FreeBSD Man Pages. [Online]. Available URL: http://www.freebsd.org/cgi/man.cgi

Luigi Rizzo. Dummynet. [Online]. Available URL: http://info.iet.unipi.it/~luigi/ip_dummynet/

Nick Rogness. A How-To Guide for running snort_inline on FreeBSD. [Online]. Available URL: http://freebsd.rogness.net/snort_inline/

Nick Rogness. A Network Tutorial/How-To Guide for the FreeBSD OS. [Online]. Available URL: http://freebsd.rogness.net/snort_inline/

The FreeBSD Documentation Project. 1999. FreeBSD Handbook. [Online]. Available URL: http://docs.FreeBSD.org/doc/.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

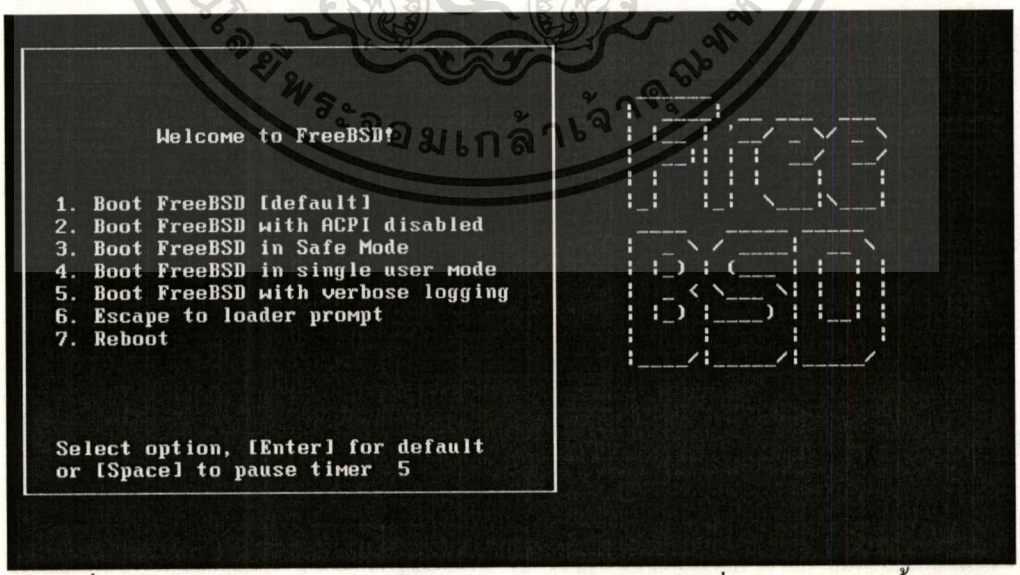
ภาคผนวก ก.
การติดตั้งและเตรียมความพร้อมของระบบ

1 การติดตั้งระบบปฏิบัติการ FreeBSD

ก่อนการติดตั้งระบบปฏิบัติการ FreeBSD เราต้องเตรียมเครื่องคอมพิวเตอร์ ที่มีการติดตั้งเน็ตเวิร์คอินเทอร์เฟสการ์ดจำนวน 3 ชุด เพื่อจำลองให้เครื่องคอมพิวเตอร์ทำงานเป็นบริดก์ที่มีอินเทอร์เฟสการ์ดขาเข้า ขาออก และอินเทอร์เฟสการ์ดสำหรับควบคุมการทำงาน ซึ่งตัวระบบปฏิบัติการ FreeBSD สามารถจำลองการทำงานเป็นอุปกรณ์เน็ตเวิร์คแบบบริดก์ได้ โดยจะทำหน้าที่ในการ Forward Packet จากอินเทอร์เฟสการ์ดขาเข้า ไปยังอินเทอร์เฟสการ์ดขาออก ซึ่งในการติดตั้งมีรายละเอียดดังต่อไปนี้

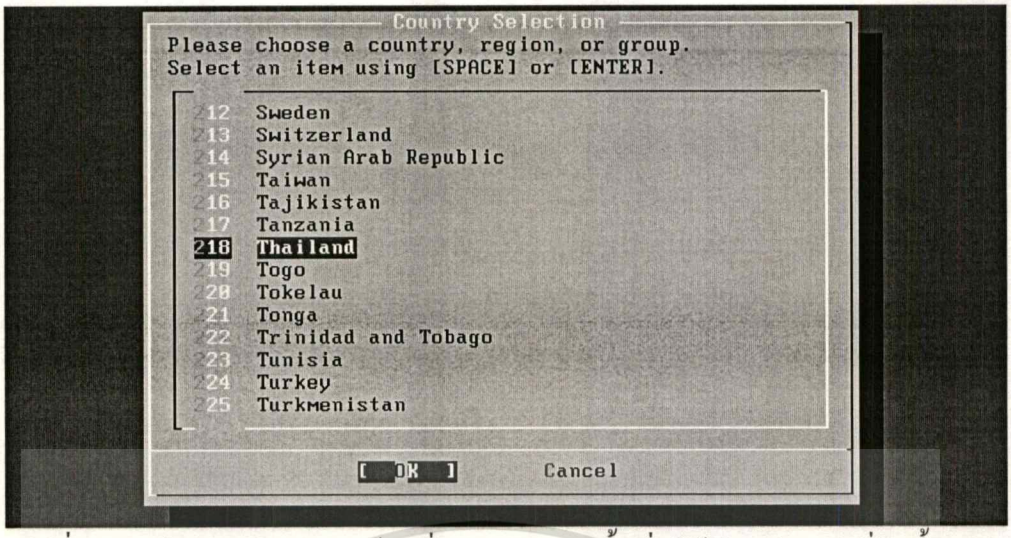
1.1. เตรียมแผ่นซีดีระบบปฏิบัติการ FreeBSD ซึ่งสามารถดาวโหลดได้ที่ "http://www.freebsd.org/" ซึ่งสามารถเลือกดาวโหลดไฟล์ชนิด ISO โดยแนะนำให้ดาวโหลดไฟล์ 7.0-RELEASE-i386-disc1.iso, 7.0-RELEASE-i386-disc2.iso, 7.0-RELEASE-i386-disc2.iso แล้วนำไปเขียนลงแผ่นซีดีรอม ในการพัฒนาระบบงานครั้งนี้ใช้ระบบปฏิบัติการ FreeBSD Release 7.0

1.2. บู๊ตเครื่องคอมพิวเตอร์ด้วยแผ่นติดตั้ง FreeBSD แผ่นที่ 1 หน้าจอภาพแสดงผลทั้งรูปที่ ข-1 จากนั้นรอจนหน้าเปลี่ยนเป็นรูปที่ ข-2 จากนั้นเลือกประเทศที่ติดตั้งใช้งานในที่นี้คือ Thailand และกดแป้น Enter ที่ตัวเลือก OK



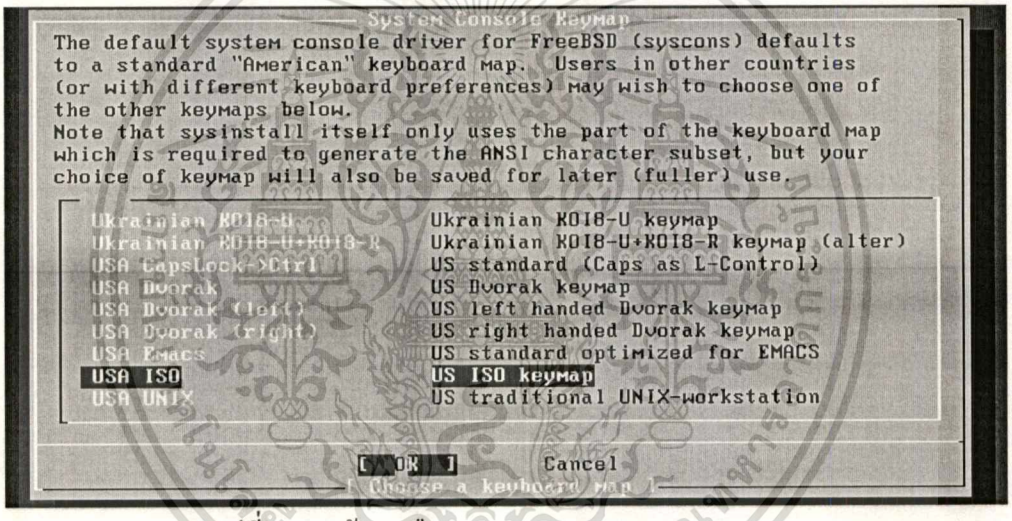
รูปที่ ก-1 หน้าจอหลัก Welcome to FreeBSD!หลังจากบู๊ตเครื่องจากแผ่นติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



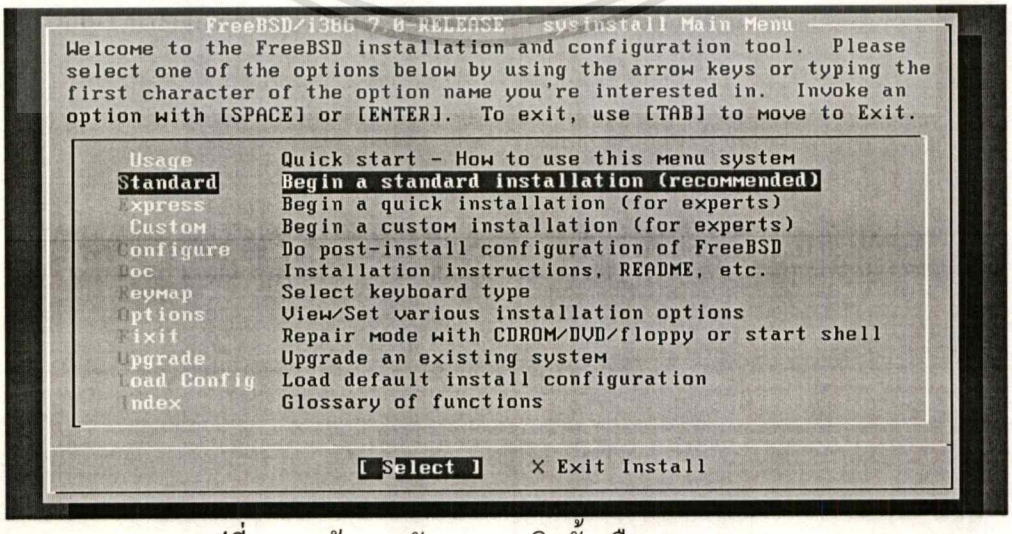
รูปที่ ก-2 หน้าจอหลักหลังจากบู๊ตเครื่องจากแผ่นติดตั้งเพื่อให้เลือกประเทศที่ติดตั้ง

1.3. จากนั้นเลือก System Console Keymap เป็น US ISO keymap



รูปที่ ก-3 หน้าจอเลือก System Console Keymap

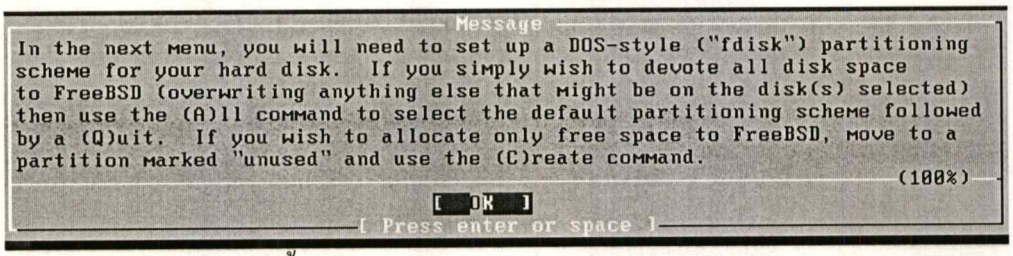
1.4. จากนั้นจะเข้าหน้าจอหลักของการเริ่มติดตั้ง เลือกการติดตั้งแบบ Standard



รูปที่ ก-4 หน้าจอหลักของการติดตั้ง เลือก Standard

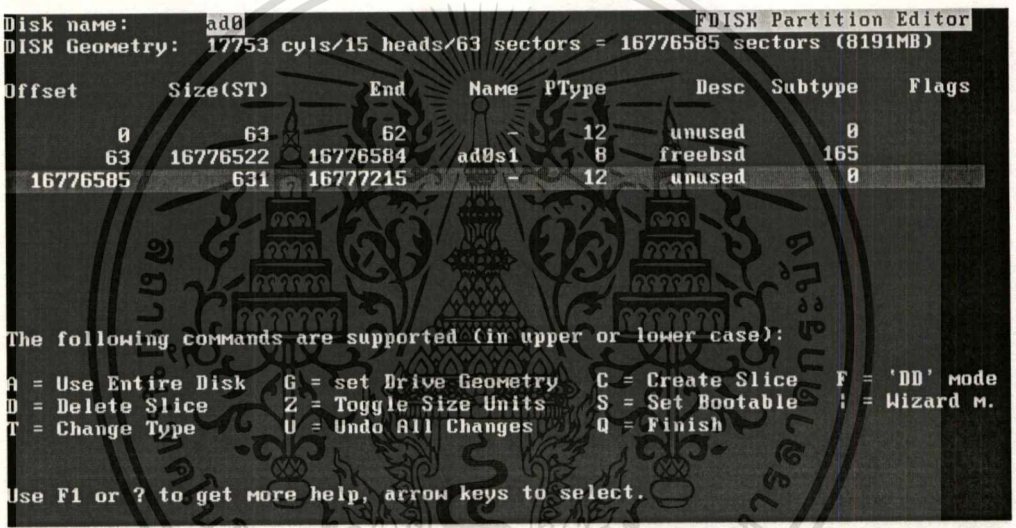
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5. คำอธิบายว่าถัดจากนี้เป็นการกำหนด Partition ของ Disk ในรูปแบบของ DOS (fdisk) ให้เลือก OK



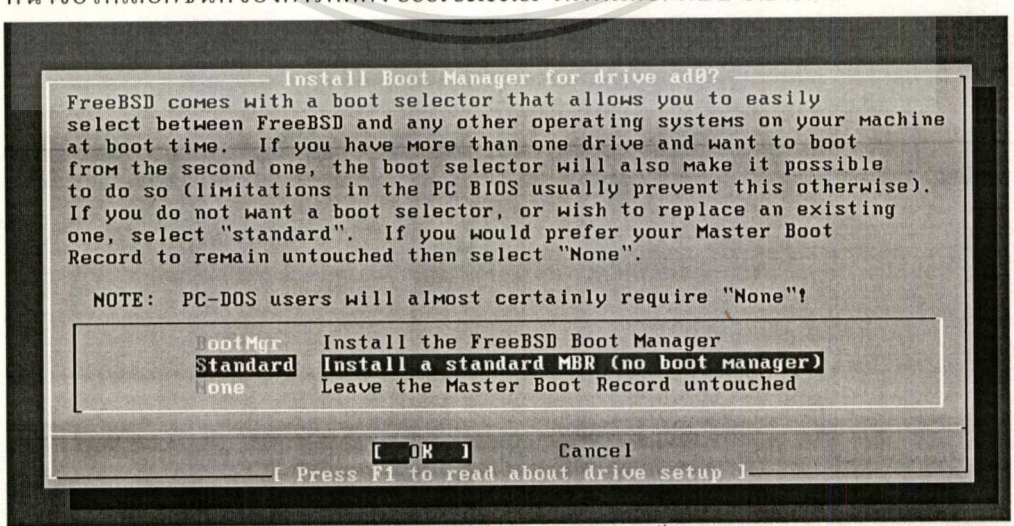
รูปที่ ก-5 คำอธิบายว่าถัดจากนี้เป็นการกำหนด Partition ของ Disk ในรูปแบบของ DOS (fdisk)

1.6. หน้าจอ fdsik ให้เลือก A คือ ใช้พื้นที่ Disk ทั้งหมด จากนั้นเลือก Q เพื่อจบการทำงานและบันทึก Partition



รูปที่ ก-6 หน้าจอภาพของคำสั่ง fdisk

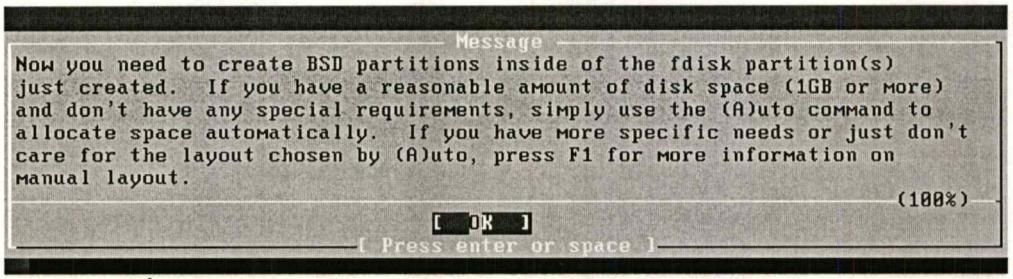
1.7. หน้าจอให้เลือกชนิดของการติดตั้ง boot selector ในที่นี้เลือกแบบ Standard



รูปที่ ก-7 หน้าจอให้เลือกชนิดของการติดตั้ง boot selector

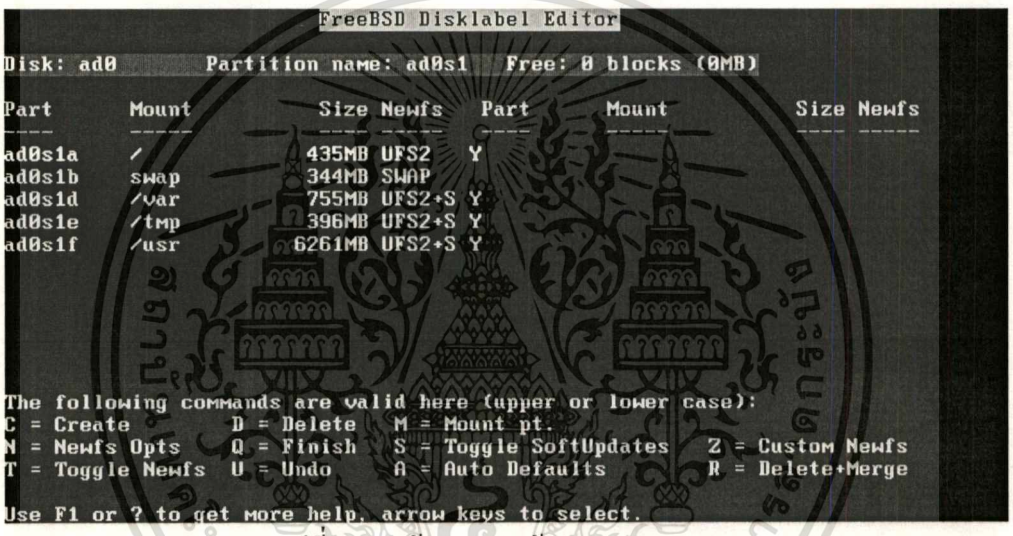
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.8. หน้าจออธิบายว่าจะสร้าง BSD Partition ภายใน fdisk partition ที่สร้างไว้ เลือก OK



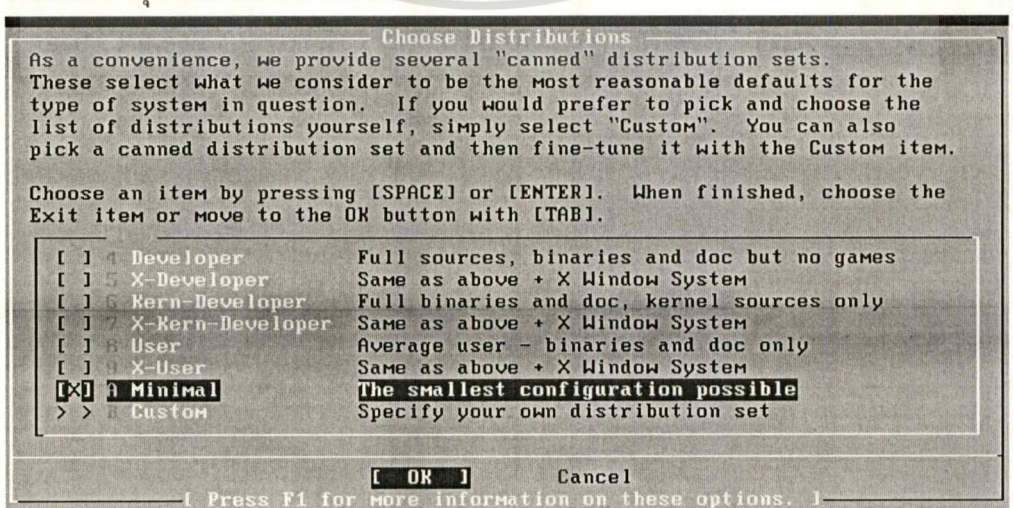
รูปที่ ก-8 หน้าจออธิบายว่าจะสร้าง BSD Partition ภายใน fdisk partition

1.9. หน้าจอการสร้าง BSD Partition เลือก A เพื่อให้สร้าง Partition Default ให้อัตโนมัติ จานั้น เลือก Q เพื่อจบการทำงาน



รูปที่ ก-9 หน้าจอการสร้าง BSD Partition

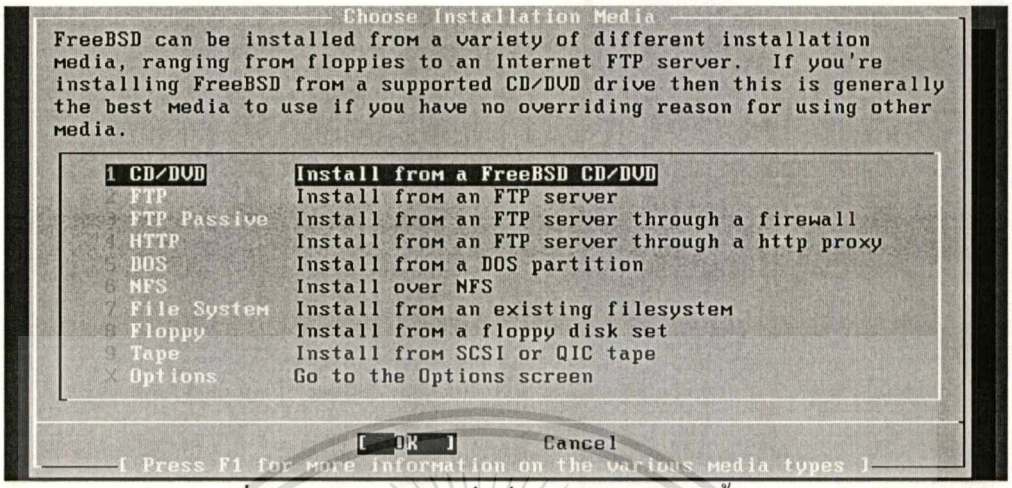
1.10. หน้าจอเลือกชุดของการติดตั้ง ในที่นี้เลือกแบบ Minimal เพื่อให้ตัวระบบปฏิบัติการมีขนาดเล็กที่สุดซึ่งเพียงพอสำหรับการใช้งาน



รูปที่ ก-10 หน้าจอเลือกชุดของการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.11.หน้าจอเลือกสื่อที่ใช้สำหรับการติดตั้งระบบ ในที่นี้เลือกสื่อเป็นแบบ CD/DVD จากนั้นเลือก OK

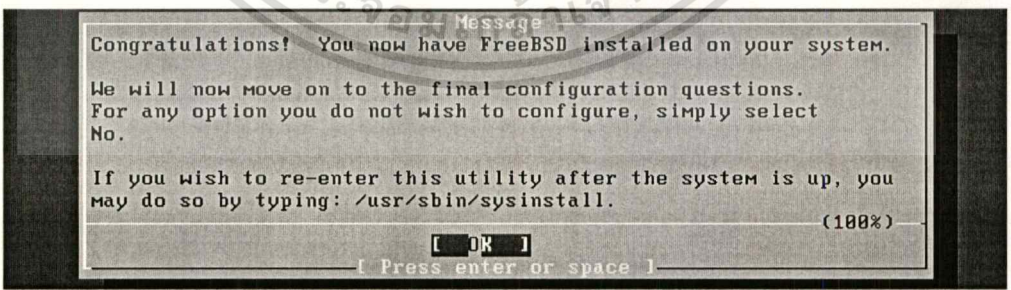


รูปที่ ก-11 หน้าจอเลือกสื่อที่ใช้สำหรับการติดตั้งระบบ

1.12.หน้าจอให้ยืนยันการติดตั้ง ในที่นี้เลือก Yes

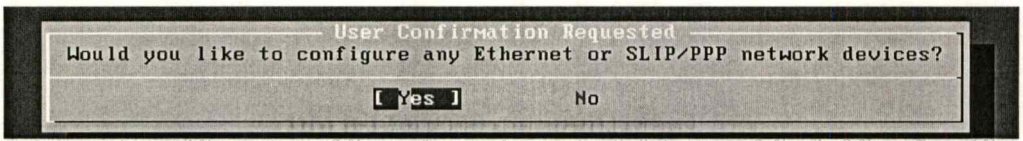
รูปที่ ก-12 หน้าจอให้ยืนยันการติดตั้ง

1.13.หน้าจอข้อความแจ้งว่าระบบปฏิบัติการ FreeBSD ติดตั้งเสร็จเรียบร้อย จากนั้นเลือก OK



รูปที่ ก-13 หน้าจอข้อความแจ้งว่าระบบปฏิบัติการ FreeBSD ติดตั้งเสร็จเรียบร้อย

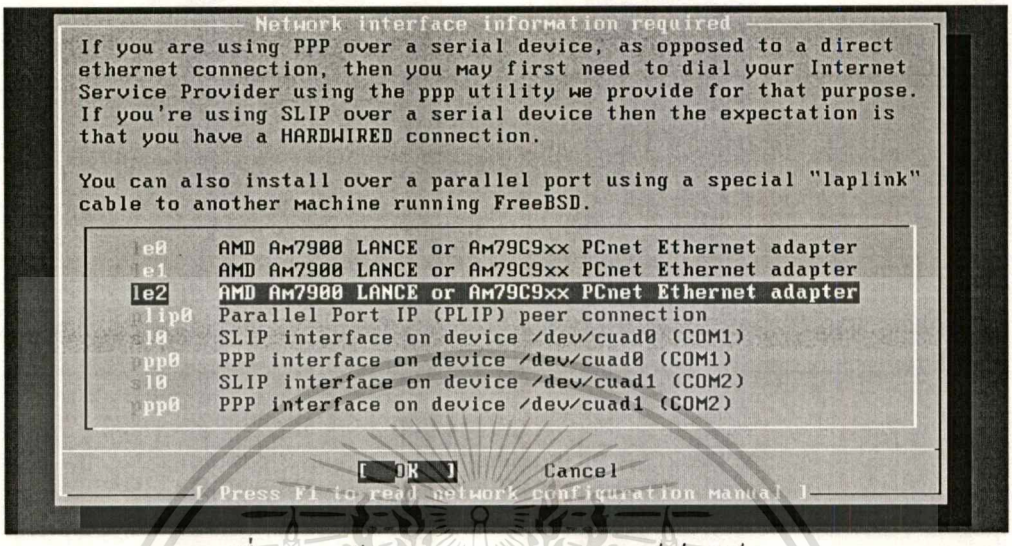
1.14.หน้าจอ เลือกว่าต้องการกำหนดค่าให้อุปกรณ์เน็ตเวิร์ค เลือก Yes



รูปที่ ก-14 หน้าจอ เลือกว่าต้องการกำหนดค่าให้อุปกรณ์เน็ตเวิร์ค

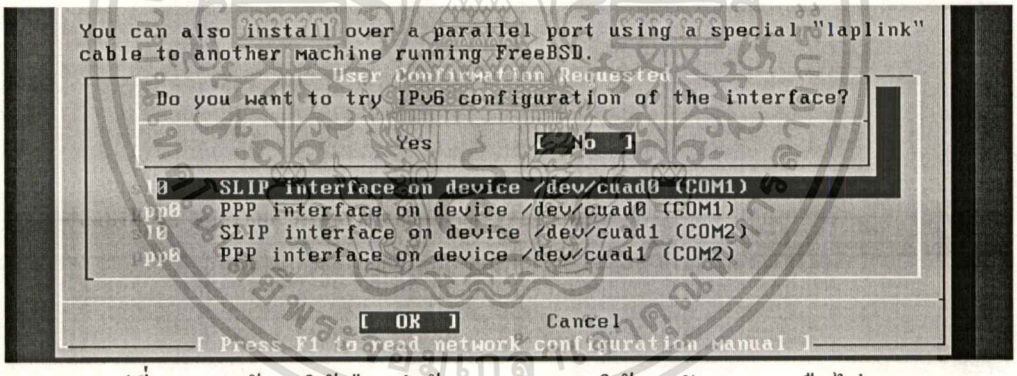
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.15.หน้าจอกำหนดค่าของอุปกรณ์เน็ตเวิร์ค ในที่นี้เครื่องที่ติดตั้งมี อีเทอร์เน็ตการ์ด 3 ชุด คือ le0 le1 และ le2 เลือกกำหนดค่าให้อีเทอร์เน็ตการ์ด le2 สำหรับเชื่อมต่อเข้าควบคุมระบบ ส่วนอีก 2 อีเทอร์เน็ตการ์ดใช้สำหรับทำเป็นอินเทอร์เฟสขาเข้าและขาออกของทราฟฟิก



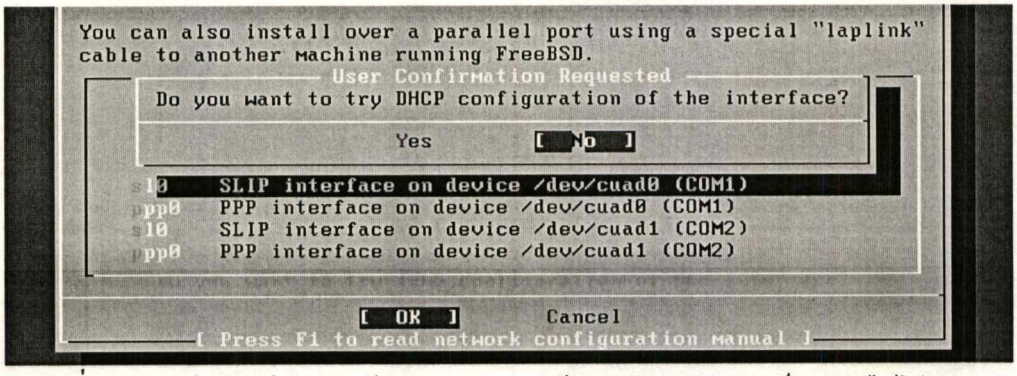
รูปที่ ก-15 หน้าจอกำหนดค่าของอุปกรณ์เน็ตเวิร์ค

1.16.หน้าจอให้เลือกว่าต้องการกำหนดให้รองรับ IPv6 หรือไม่ ในที่นี้เลือก No



รูปที่ ก-16 หน้าจอให้เลือกว่าต้องการกำหนดให้รองรับ IPv6 หรือไม่

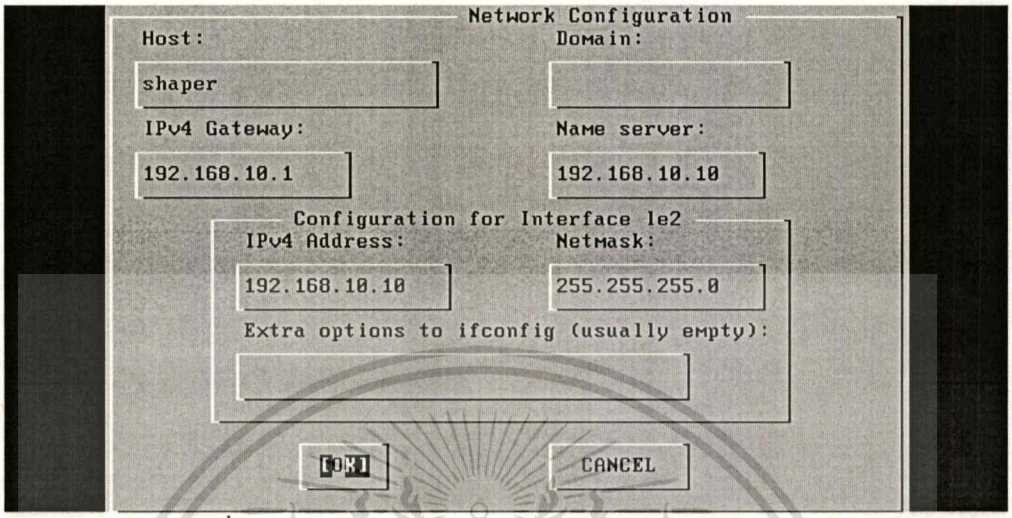
1.17.หน้าจอให้เลือกว่าต้องการกำหนดใช้ DHCP กับอินเทอร์เฟส หรือไม่ ในที่นี้เลือก No



รูปที่ ก-17 หน้าจอให้เลือกว่าต้องการกำหนดใช้ DHCP กับอินเทอร์เฟส หรือไม่

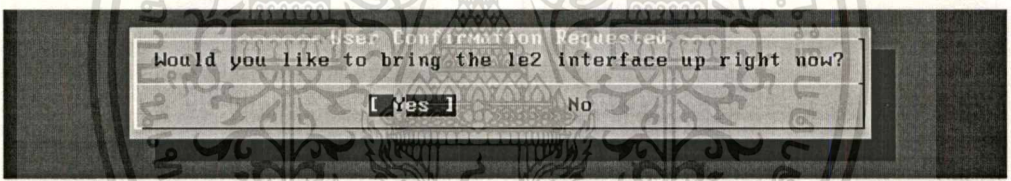
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.18.หน้าจอกำหนดค่าต่างของอินเทอร์เฟส le2 โดยกำหนดค่าดังนี้ Host : shaper, IPv4 Gateway : 192.168.10.1 Name Server : 192.168.10.10, IPv4 Address : 192.168.10.10, Netmask : 255.255.255.0 จากนั้นเลือก OK



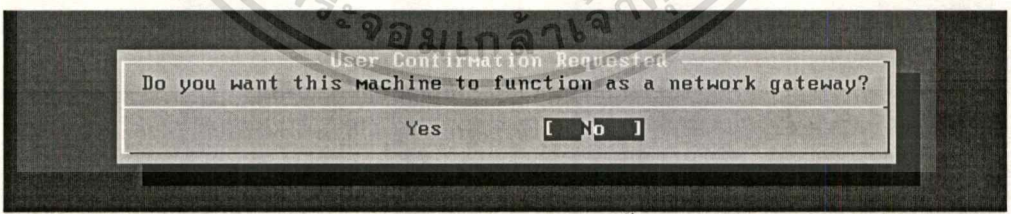
รูปที่ ก-18 หน้าจอกำหนดค่าต่างของอินเทอร์เฟส le2

1.19.หน้าจอให้เลือกว่าต้องการให้ อินเทอร์เฟส le2 เปิดใช้งาน หรือไม่ จากนั้นเลือก Yes



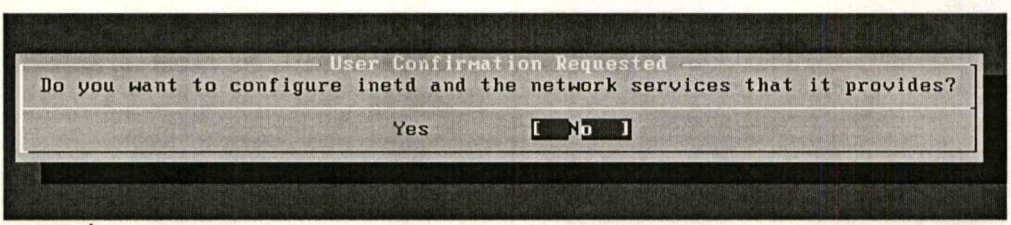
รูปที่ ก-19 หน้าจอให้เลือกว่าต้องการให้ อินเทอร์เฟส le2 เปิดใช้งาน หรือไม่

1.20.หน้าจอให้เลือกว่าต้องการให้เครื่องที่ติดตั้งทำหน้าที่เป็นเกตเวย์ หรือไม่ ในที่นี้เลือก No



รูปที่ ก-20 หน้าจอให้เลือกว่าต้องการให้เครื่องที่ติดตั้งทำหน้าที่เป็นเกตเวย์ หรือไม่

1.21.หน้าจอให้เลือกว่าต้องการกำหนดค่า inetd และ network service หรือไม่ ในที่นี้เลือก No



รูปที่ ก-21 หน้าจอให้เลือกว่าต้องการกำหนดค่า inetd และ network service หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

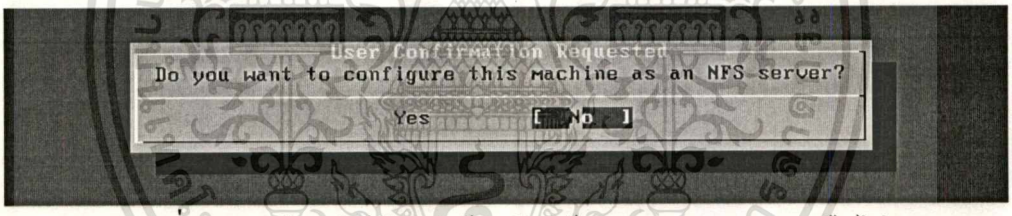
1.22.หน้าจอให้เลือกว่าต้องเปิดการใช้งาน SSH (Secure Shell) เพื่อใช้ในการทำรีโหมด Login หรือไม่ ในที่นี้เลือก Yes

รูปที่ ก-22 หน้าจอให้เลือกเปิดการใช้งาน SSH (Secure Shell) เพื่อใช้ในการทำรีโหมด Login หรือไม่

1.23.หน้าจอให้เลือกว่าต้องเปิดใช้บริการ FTP หรือไม่ ในที่นี้เลือก No

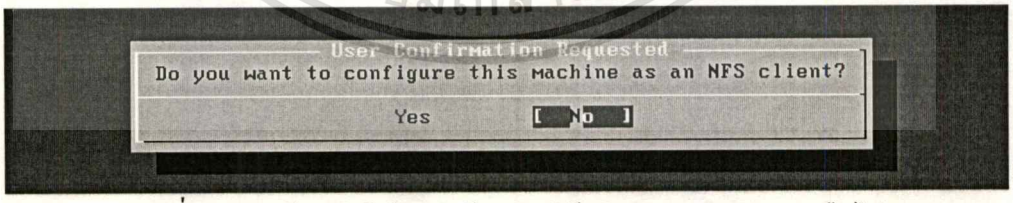
รูปที่ ก-23 หน้าจอให้เลือกว่าต้องเปิดใช้บริการ FTP หรือไม่

1.24.หน้าจอให้เลือกว่าต้องเปิดใช้บริการ NFS Server หรือไม่ ในที่นี้เลือก No



รูปที่ ก-24 หน้าจอให้เลือกว่าต้องเปิดใช้บริการ NFS Server หรือไม่

1.25.หน้าจอให้เลือกว่าต้องเปิดใช้บริการ NFS Client หรือไม่ ในที่นี้เลือก No



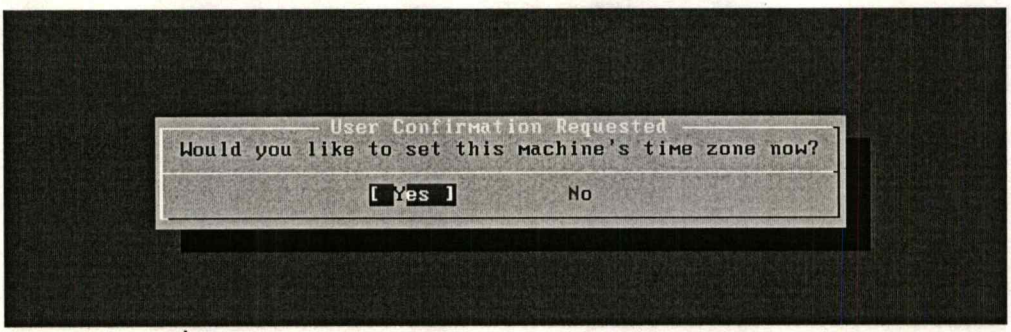
รูปที่ ก-25 หน้าจอให้เลือกว่าต้องเปิดใช้บริการ NFS Client หรือไม่

1.26.หน้าจอให้เลือกว่าต้องกำหนดค่าของ system console หรือไม่ ในที่นี้เลือก No

รูปที่ ก-26 หน้าจอให้เลือกว่าต้องกำหนดค่าของ system console หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.27.หน้าจอให้เลือกว่าต้องกำหนดค่าของ time zone หรือไม่ ในที่นี้เลือก Yes

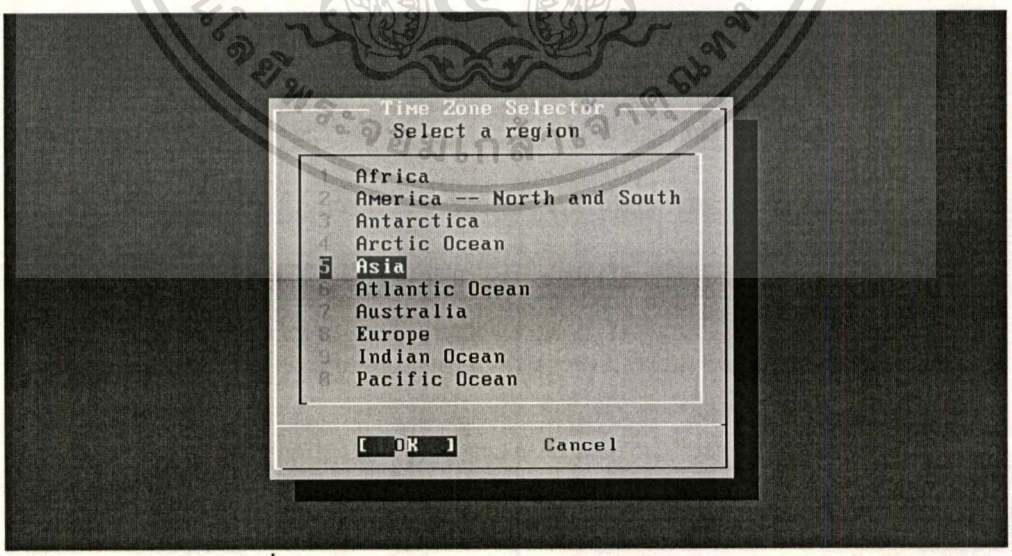


รูปที่ ก-27 หน้าจอให้เลือกว่าต้องกำหนดค่าของ time zone หรือไม่

1.28.หน้าจอให้เลือกว่าต้องใช้ค่าเวลาของเครื่องหรือ ค่า UTC ในที่นี้เลือก Yes

รูปที่ ก-28 หน้าจอให้เลือกว่าต้องใช้ค่าเวลาของเครื่องหรือ ค่า UTC

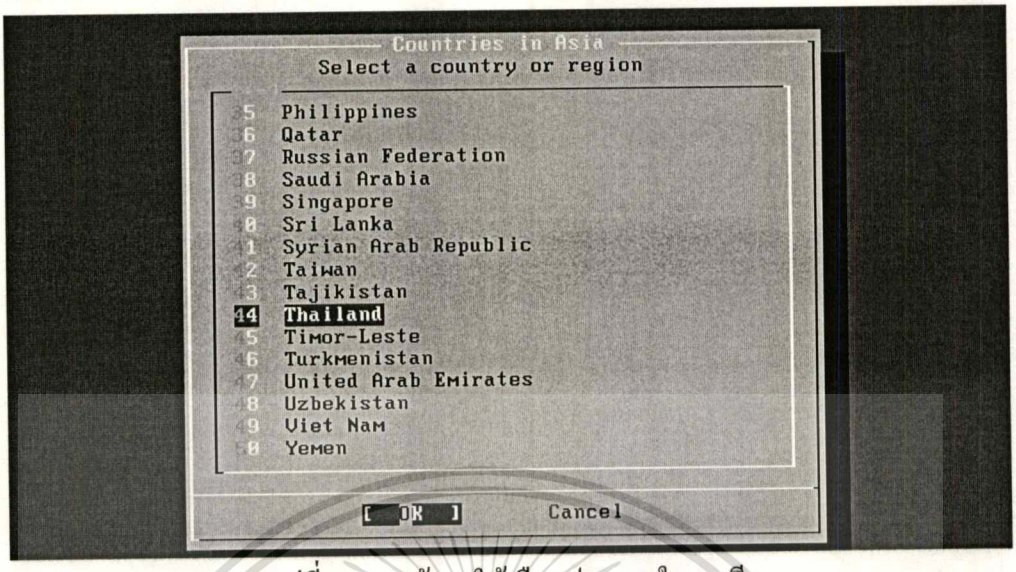
1.29.หน้าจอให้เลือก Time Zone ในที่นี้เลือก Asia จากนั้นเลือก OK



รูปที่ ก-29 หน้าจอหน้าจอให้เลือก Time Zone

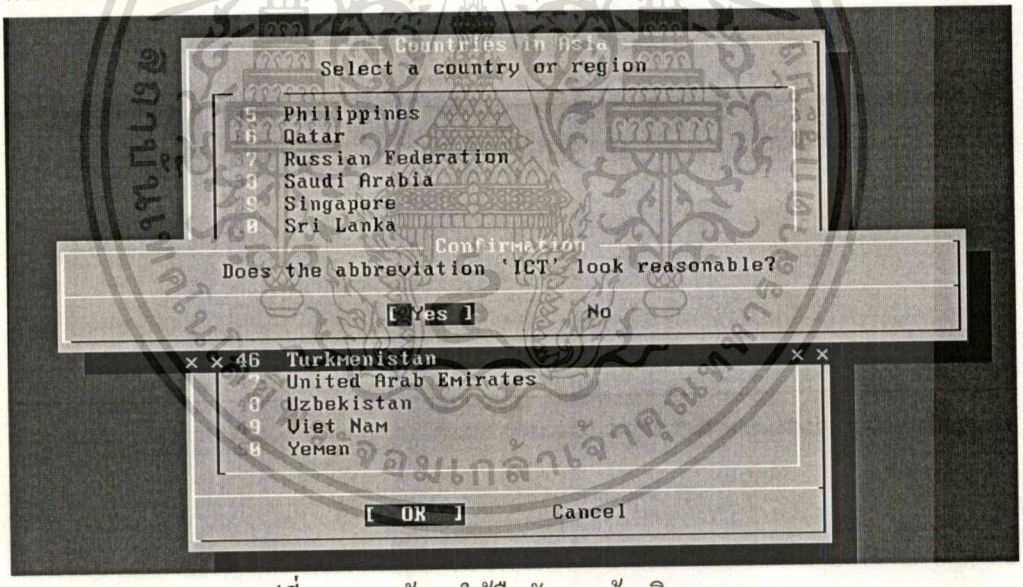
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.30.หน้าจอให้เลือกประเทศใน เอเชีย ในที่นี้เลือก Thailand จากนั้นเลือก OK



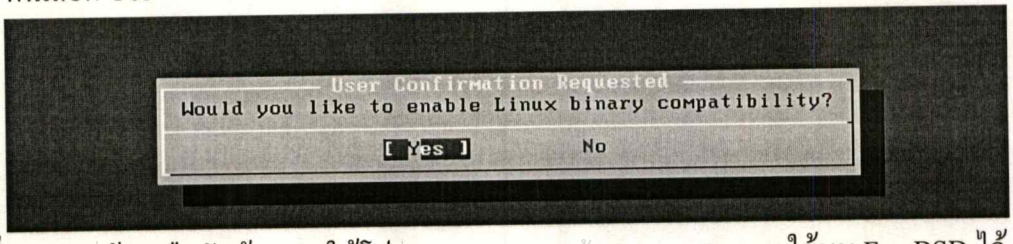
รูปที่ ก-30 หน้าจอให้เลือกประเทศใน เอเชีย

1.31.หน้าจอให้ยืนยันการอ้างอิงเวลา ICT ในที่นี้เลือก Yes



รูปที่ ก-31 หน้าจอให้ยืนยันการอ้างอิงเวลา ICT

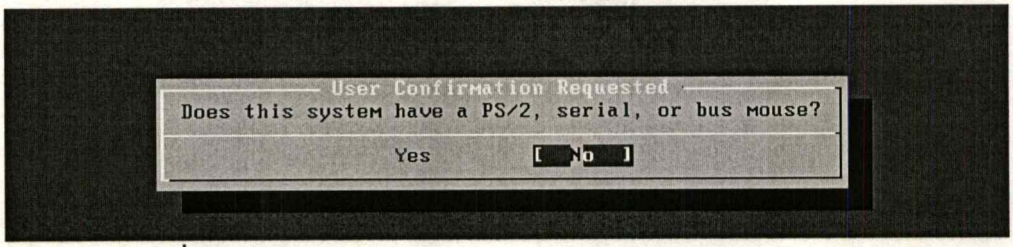
1.32.หน้าจอให้ยืนยันต้องการให้โปรแกรม Binary ของ Linux สามารถใช้บน FreeBSD ได้ ในที่นี้เลือก Yes



รูปที่ ก-32 หน้าจอยืนยันต้องการให้โปรแกรม Binary ของ Linux สามารถใช้บน FreeBSD ได้

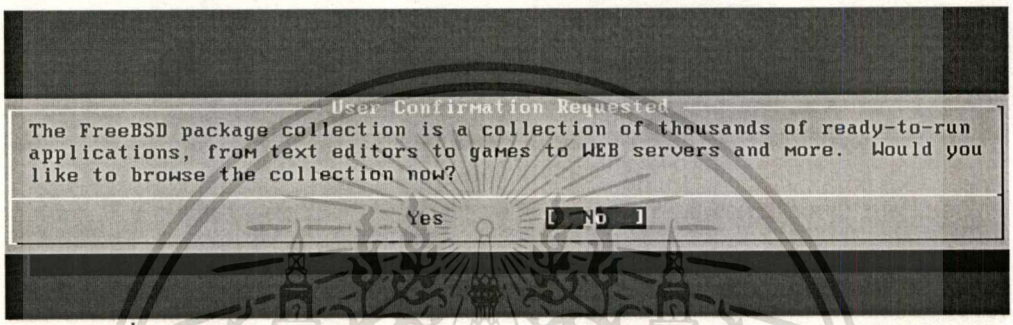
เอกสารนี้ [illegible] เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.33.หน้าจอให้ยืนยันต้องการใช้ PS/2, serial Mouse หรือไม่ ในที่นี้เลือก No



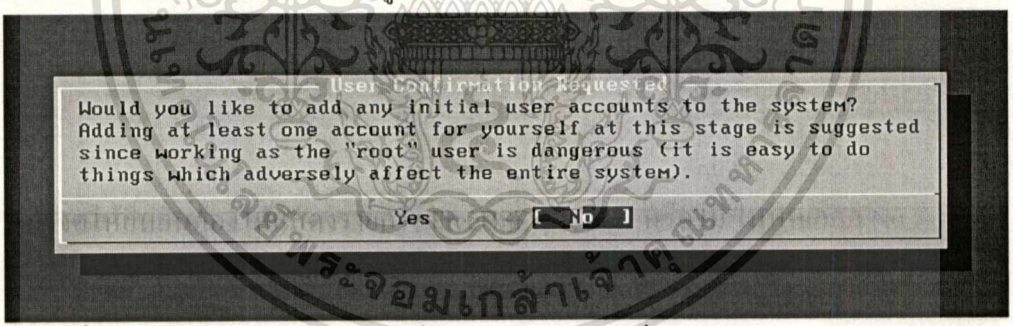
รูปที่ ก-33 หน้าจอให้ยืนยันต้องการใช้ PS/2, serial Mouse หรือไม่

1.34.หน้าจอให้ยืนยันต้องให้แสดงรายการ FreeBSD package หรือไม่ ในที่นี้เลือก No



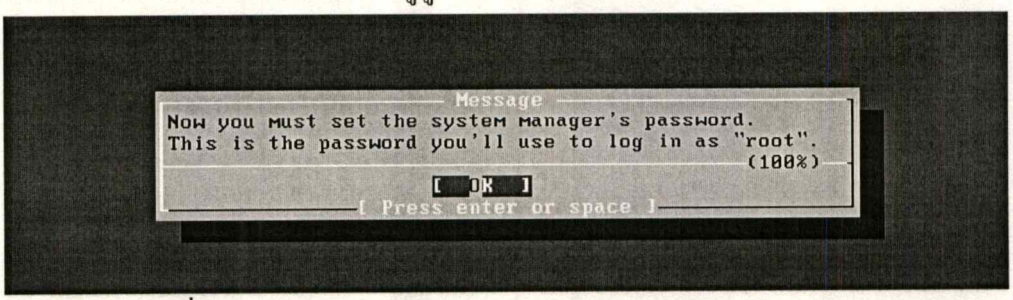
รูปที่ ก-34 หน้าจอให้ยืนยันต้องให้แสดงรายการ FreeBSD package หรือไม่

1.35.หน้าจอให้ยืนยันต้องเพิ่มทะเบียนผู้ใช้งานอื่น นอกจาก User root หรือไม่ ในที่นี้เลือก No



รูปที่ ก-35 หน้าจอให้ยืนยันต้องเพิ่มทะเบียนผู้ใช้งานอื่น นอกจาก User root หรือไม่

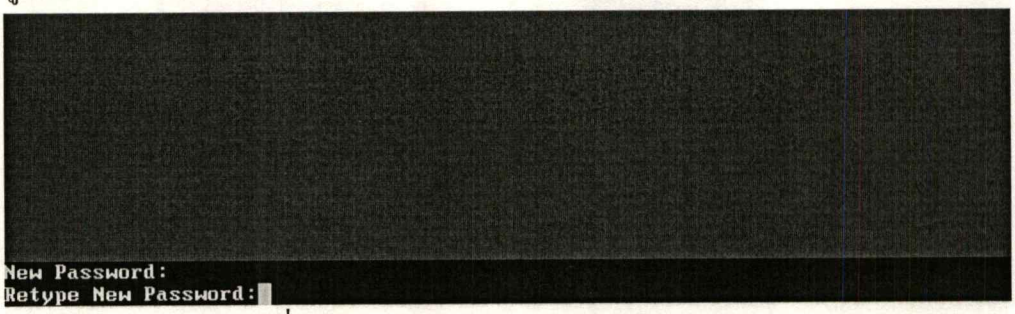
1.36.หน้าจอแจ้งให้กำหนดรหัสผ่านของผู้ดูแลระบบ (root) ในที่นี้เลือก OK



รูปที่ ก-36 หน้าจอแจ้งให้กำหนดรหัสผ่านของผู้ดูแลระบบ (root)

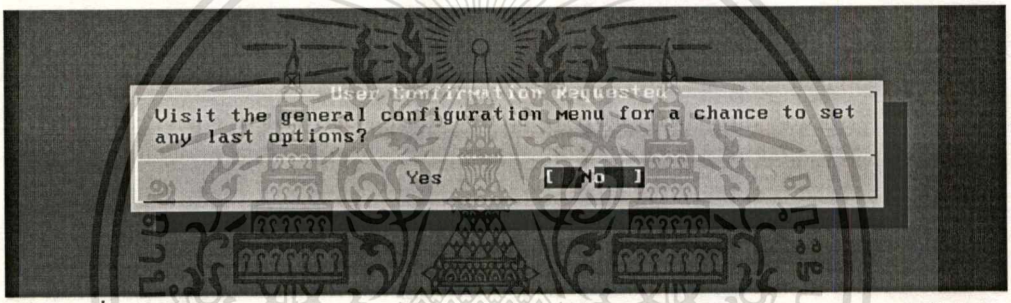
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.37.หน้าจอป้อนรหัสผ่านของ User Root ในที่นี้ให้ป้อนรหัสผ่าน 2 ครั้งเพื่อยืนยันความถูกต้อง



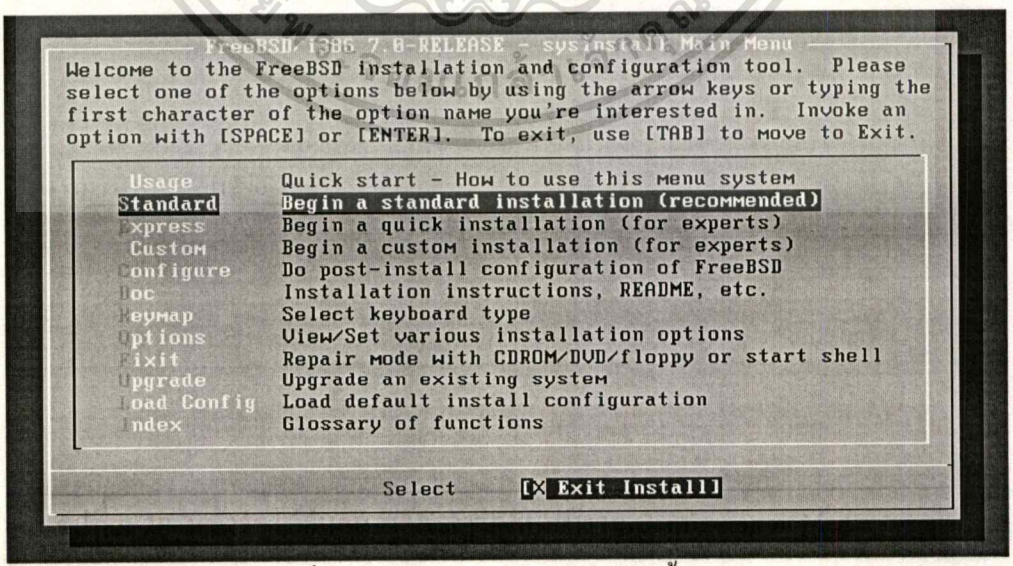
รูปที่ ก-37 หน้าจอป้อนรหัสผ่านของ User Root

1.38.หน้าจอให้ยืนยันว่าต้องการกลับหน้าจอ Configuration Menu เพื่อกำหนดค่าตัวเลือกอื่นหรือไม่ ในที่นี้เลือก No



รูปที่ ก-38 หน้าจอให้ยืนยันว่าต้องการกลับหน้าจอ Configuration Menu หรือไม่

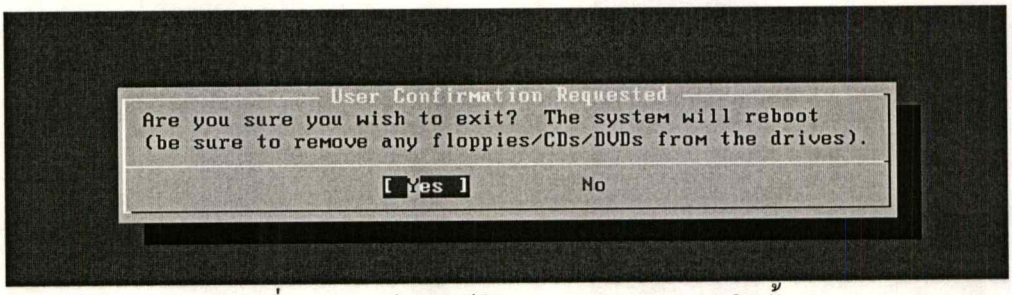
1.39.หน้าจอหลักของการติดตั้งระบบ เมื่อกำหนดค่าต่างๆ เรียบร้อยระบบจะกลับมาที่หน้าจอการติดตั้งระบบหลัก ซึ่งในที่นี้ได้ทำการติดตั้งระบบเรียบร้อยแล้ว จึงเลือก Exit Install



รูปที่ ก-39 หน้าจอหลักของการติดตั้งระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.40.หน้าจอให้ยืนยันการออกจากการติดตั้ง ในที่นี้เลือก Yes จากนั้นระบบจะทำการ reboot เป็นอันจบขั้นตอนการติดตั้ง



รูปที่ ก-40 หน้าจอให้ยืนยันการออกจากการติดตั้ง

2 การคอมไพล์เคอร์เนลของ FreeBSD ให้รับการใช้งาน Dummynet และ IPDIVERT

โดยปรกติระบบเคอร์เนลของ FreeBSD จะรองรับการทำงานของ IPFW อยู่แล้วเมื่อเราต้องการใช้ Dummynet และ ipdivert จำเป็นต้องทำการคอมไพล์เคอร์เนลเพิ่มเติม โดยมีขั้นตอนดังต่อไปนี้

1. ย้ายไดเร็กทรอรี่เข้าไปที่เก็บไฟล์เคอร์เนล และคัดลอกไพล์ GENERIC ให้เป็นชื่อที่ต้องการเพื่อป้องการการเสียหายของไฟล์คอนฟิกของเคอร์เนลต้นฉบับ ในที่นี้จะคัดลอกไฟล์ GENERIC เป็นชื่อ SHAPER

```
shaper# cd /usr/src/sys/i386/conf
shaper# cp GENERIC SHAPER
```

2. แก้ไขค่าคอนฟิกของเคอร์เนล SHAPER ให้รองรับการทำงาน IPFW Dummynet และ IPDIVERT โดยใช้คำสั่ง vi SHAPER เพื่อแก้ไขค่าต่าง ๆ ดังต่อไปนี้

```
shaper# vi SHAPER
cpu          I486_CPU
cpu          I586_CPU
cpu          I686_CPU
ident        SHAPER
# Configuretion for Traffic Shaper
options   IPFIREWALL              # enable ipfirewall (required for dummynet)
options   IPFIREWALL_FORWARD      # enable ipfirewall forward
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
options   IPFIREWALL_DEFAULT_TO_ACCEPT  # default firewall policy
options   IPFIREWALL_VERBOSE            # enable firewall output
options   IPFIREWALL_VERBOSE_LIMIT=5    # limit firewall output
options   DUMMYNET                      # enable dummynet operation
options   HZ=1000                       # set the timer granularity
options   IPDIVERT                      # enable for divert traffic
# if_bridge for network bridge device
device    if_bridge
```

3. เมื่อแก้ไขค่าคอนฟิกของเคอร์เนล SHAPER เรียบร้อยให้ทำการบันทึกผลและออกจากโปรแกรม vi (ใช้คำสั่ง :wq!) จากนั้นทำการคอมไพล์เคอร์เนลและติดตั้งระบบเคอร์เนลของ FreeBSD ใหม่ โดยมีคำสั่งและขั้นตอนดังต่อไปนี้

```
shaper# config SHAPER
Kernel build directory is ../compile/SHAPER
Don't forget to do ``make cleandepend && make depend''
shaper# cd ../compile/SHAPER
shaper# make cleandepend ; make depend ; make ; make install
```

4. รอจนระบบคอมไพล์เคอร์เนลและติดตั้งระบบเสร็จสมบูรณ์ จากนั้นให้รีบูทระบบใหม่ โดยก่อนการรีบูทระบบควรตรวจสอบรายละเอียดของเคอร์เนลที่ใช้งานปัจจุบัน ซึ่งใช้คำสั่งดังต่อไปนี้

```
shaper# uname -a
FreeBSD Bridge.lab.com 7.0-RELEASE FreeBSD 7.0-RELEASE
#0: Sun Feb 24 19:59:52 UTC 2008
root@logan.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  i386
shaper# shutdown -r now
```

5. เมื่อระบบบูทเสร็จเรียบร้อยควรตรวจสอบรายละเอียดของเคอร์เนลที่ใช้งานปัจจุบันอีกครั้งว่าเป็นเคอร์เนลที่คอมไพล์ล่าสุดหรือไม่โดยดูที่วันเดือนปีที่คอมไพล์เคอร์เนล คำสั่งที่ใช้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
shaper# uname -a
FreeBSD shaper.lab.com 7.0-RELEASE FreeBSD 7.0-RELEASE
#1: Sat Sep 13 12:10:51 UTC 2008
root@shaper.lab.com:/usr/src/sys/i386/compile/SHAPER  i386
```

3 การติดตั้งระบบเว็บเซอร์ฟเวอร์ Apache

การติดตั้งเว็บเซอร์ฟเวอร์ Apache บนระบบ FreeBSD สามารถทำได้โดยการติดตั้งชุดซอฟต์แวร์ ports tree ซอฟต์แวร์ทั้งหมดจะเก็บอยู่ในไฟล์ \7.0-RELEASE\ports\ports.tgz และเมื่อติดตั้ง port tree เหล่านี้จะเก็บอยู่ที่ /usr/ports เมื่อกำหนดค่า Options ของชุดซอฟต์แวร์ (make config) และสั่งติดตั้ง (make install clean) ระบบจะไปดาวน์โหลดข้อมูลที่จำเป็นในการติดตั้งทั้งหมดผ่านทางอินเทอร์เน็ต ในการติดตั้งชุดซอฟต์แวร์ Apache สามารถทำได้ดังนี้

shaper# **cd /usr/ports/www/apache22**

shaper# **make config**

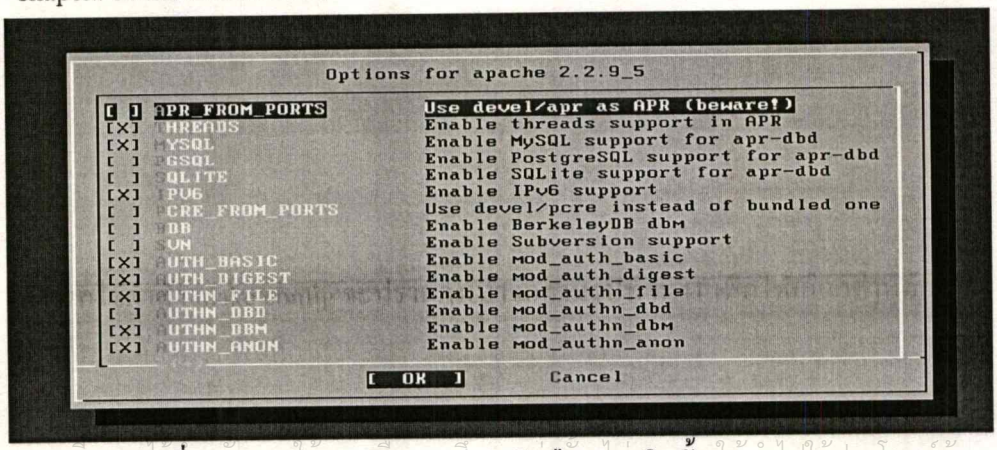
หลังจากใช้คำสั่ง make config จะปรากฏหน้าจอตามรูปที่ ก-41 เพื่อให้เลือกค่าที่ต้องการติดตั้ง ในที่นี้ตัวเลือกที่ต้องการเพิ่มคือ

[X] THREADS Enable threads support in APR

[X] SSL Enable mod_ssl

จากนั้นเลือก OK เพื่อบันทึกค่าตัวเลือก จากนั้นสั่งติดตั้งซอฟต์แวร์ด้วยคำสั่ง make install clean ระบบจะเชื่อมต่ออินเทอร์เน็ตเพื่อดาวน์โหลดข้อมูลที่จำเป็นสำหรับการติดตั้ง

shaper# **make install clean**



รูปที่ ก-41 หน้าจอการกำหนดค่าตัวเลือกการติดตั้งของ Apache

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อระบบติดตั้งเว็บเซอร์ฟเวอร์เรียบร้อยแล้วทำการเพิ่มคำสั่งให้ระบบ FreeBSD สั่งให้บริการเว็บเซอร์ฟเวอร์ Apache เริ่มทำงานอัตโนมัติเมื่อเปิดเครื่องโดยเพิ่มคำสั่งลงในไฟล์ /etc/rc.conf ดังนี้

```
shaper# vi /etc/rc.conf
apache22_enable="YES"
```

4 การติดตั้งระบบจัดการฐานข้อมูล MySQL

การติดตั้งระบบจัดการฐานข้อมูล MySQL บนระบบปฏิบัติการ FreeBSD สมารถติดตั้งผ่าน port tree เช่นเดียวกับ Apache Web Server โดยมีขั้นตอนและคำสั่งดังนี้

```
shaper# cd /usr/ports/databases/mysql50-server
shaper# make WITH_CHARSET=tis620  WITH_XCHARSET=all \
        WITH_COLLATION=tis620_thai_ci  install clean
```

เมื่อรอจนระบบติดตั้ง MySQL เรียบร้อยแล้วทำการเพิ่มคำสั่งให้ระบบ FreeBSD สั่งให้บริการเว็บเซอร์ฟเวอร์ MySQL เริ่มทำงานอัตโนมัติเมื่อเปิดเครื่องโดยเพิ่มคำสั่งลงในไฟล์ /etc/rc.conf ดังนี้

```
shaper# vi /etc/rc.conf
mysql_enable="YES"
```

จากนั้นกำหนดรหัสผ่านของ USER : root (รหัสผู้ใช้สำหรับจัดการฐานข้อมูล MySQL) ให้กับฐานข้อมูล MySQL โดยกำหนดรหัสผ่าเป็น "@root" โดยก่อนจะกำหนดรหัสผ่านต้องสั่งให้ MySQL เริ่มให้บริการ ซึ่งใช้คำสั่งดังนี้

```
shaper# /usr/local/bin/mysql_install_db
shaper# chown -R mysql:mysql /var/db/mysql
shaper# /usr/local/bin/mysqld_safe &
[1] 1530
shaper# Starting mysqld daemon with databases from /var/db/mysql
shaper# mysqladmin -u root password '@root'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

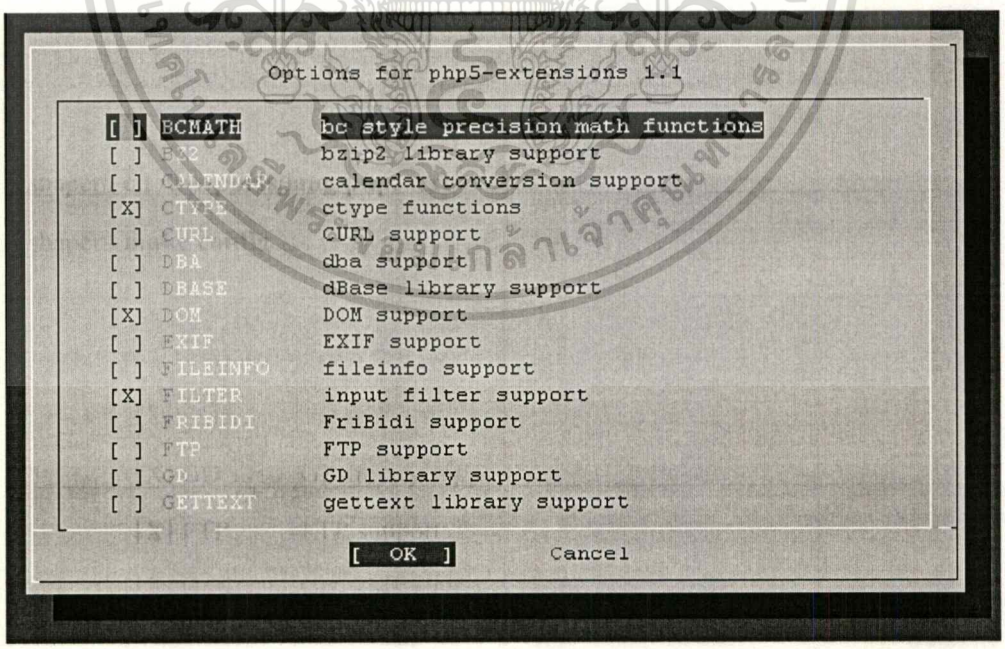
5 การติดตั้งตัวแปลภาษา PHP

การติดตั้งตัวแปลภาษา PHP บนระบบปฏิบัติการ FreeBSD สมารถติดตั้งผ่าน port tree การติดตั้ง PHP ก็เพื่อให้รองรับการทำงานและการสั่งงานระบบผ่านหน้าเว็บเพ็จหรือเว็บเซอร์ฟเวอร์ โดยในการติดตั้ง PHP ต้องให้รองรับการทำงานร่วมกับฐานข้อมูล MySQL ได้ด้วย โดยมีขั้นตอนและคำสั่งดังนี้

```
shaper# cd /usr/ports/lang/php5-extensions
shaper# make config
```

เมื่อปรากฏหน้าจอภาพตามรูปที่ ก-42 เลือก Options ของการติดตั้ง PHP ที่จำเป็นดังนี้

```
[X] DBA      dba support
[X] GD       GD library support
[X] FTP      FTP support
[X] GMP      GNU MP support
[X] MYSQL    MySQL database support
```



รูปที่ ก-42 หน้าจอการกำหนดค่าตัวเลือกการติดตั้งของ php5-extensions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อแก้ไขค่า Options เรียบทำการบันทึกผลโดยเลือกคำสั่ง OK แล้วติดตั้งระบบด้วยคำสั่ง make install clean และรอจนระบบดาวน์โหลดข้อมูลการติดตั้งเรียบร้อย

```
shaper# make install clean
```

เมื่อติดตั้งโปรแกรมตัวแปลภาษา PHP เสร็จเรียบร้อยจากนั้นต้องปรับแต่งค่าในไฟล์ควบคุมของ PHP (php.ini) เพื่อให้รอบรับการทำงานตามที่ต้องการซึ่งมีขั้นตอนดังนี้

```
shaper# cp /usr/local/etc/php.ini-dist /usr/local/etc/php.ini
shaper# vi /usr/local/etc/php.ini
register_globals = On
default_charset = "tis-620"
upload_tmp_dir ="/tmp"
session.save_path = "/tmp"
```

6 การติดตั้งโปรแกรมที่พัฒนาขึ้น

การติดตั้งโปรแกรมที่พัฒนาขึ้นลงบนระบบ FreeBSD สามารถทำได้โดยคัดลอกไฟท์จากซีดีรอมไปไว้ที่ไดเร็คทรอรี่ที่ใช้งานด้วยคำสั่งดังนี้

```
shaper# mount /cdrom
shaper# cd /cdrom
shaper# tar –xvzf http_shaper.tgz /
```

หลังใช้คำสั่ง tar แล้วโปรแกรมที่พัฒนาทั้งหมดจะถูกติดตั้งลงบนระบบโดยมีโฟลเดอร์หลักๆ ที่ติดตั้งโปรแกรมคือ /usr/local/shaper จะเป็นส่วนของชุดโปรแกรมคำสั่งในการตรวจสอบการทำงานของระบบ และโฟลเดอร์ /usr/local/www/apache22/data จะเป็นส่วนของโปรแกรมเว็บอินเทอร์เฟสทั้งหมด นอกจากนั้นยังต้องเพิ่มรายการคำสั่งลงใน /etc/crontab ดังนี้

```
# RRD Graph create every 5 minutes
*/5   *    *    *    *    root   /usr/local/shaper/rrd_graph
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# init_shaper restart every 30 minutes
*/30  *    *    *    *    root   /usr/local/shaper/init_shaper restart
# Apply all Rule every 1 minute
*     *    *    *    *    root   /usr/local/shaper/apply_rule
```

7 ชุดคำสั่งของโปรแกรมที่พัฒนาขึ้น

7.1 โปรแกรม http_sniffer.pl

ชุดโปรแกรมที่พัฒนาขึ้นใช้โปรแกรมภาษา Perl ในการพัฒนาให้ทำงานร่วมกับโปรแกรม tcpdump เพื่อทำหน้าที่ตรวจสอบข้อมูลของการติดต่อสื่อสารของโปรโตคอล HTTP เมื่อพัฒนาโปรแกรมเรียบร้อยแล้วก็ทำการคอมไพล์ชุดโปรแกรมทั้งหมดเป็นไบนารีที่สามารถทำงานได้อิสระด้วยคำสั่ง perlcc –o http_sniffer http_sniffer.pl ผลที่ได้จะได้โปรแกรม http_sniffer เพื่อนำไปใช้งาน

```
#!/usr/bin/perl
system("/usr/local/shaper/flush_http_rule");
$i=1000;
#-------------------------------------------------------------------------
%history=("session",$i);
%List_queue_1=($src_ip,$q_id_1);
%List_queue_2=($src_ip,$q_id_2);
%List_dst=($src_ip,$dts_ip);
%List_int=($src_ip,$int);
$data_file="/usr/local/shaper/current_cmd/http.list";

$|=1;
open (STDIN,"/usr/sbin/tcpdump -i le0 -lnx -s 1024 tcp |");
while (<>) {
   if (/^\S/) {
 while (($packet=~/(OPTIONS|GET|HEAD|POST|PUT|DELETE|TRACE|WWW-
Authenticate|Authorization).+/g) && ($packet=~/HTTP/)) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# print "S:$client : $sport -> D:$host : $dport \t$&\n";
 open(DAT, $data_file) || die("Could not open file!");
     @raw_data=<DAT>;
    foreach $line (@raw_data)
     {  ($src_ip,$q_id_1,$q_id_2,$dst_ip,$int)=split(",",$line);
        $List_queue_1{$src_ip}=$q_id_1;
        $List_queue_2{$src_ip}=$q_id_2;
      $List_dst{$src_ip}=$dst_ip;
      $List_int{$src_ip}=$int;
# print  $src_ip.",".$q_id_1.",".$q_id_2.",".$dst_ip.",".$int."\n";
     }
  $session = "$client-$host:$dport";
# print $List_queue_1{"192.168.1.11"}."queue---\n";
# print $client;
# print $List_queue_1{$client}."queue---\n";
  if (exists $List_queue_1{$client}) {
  if (exists $history{$session}) {
#           print "Rule exists\n";
   } else {
     $i=$i+1;
     $history{$session} =$i;
#   print "$session \n";
 $cmd_1 = "ipfw add $i queue $List_queue_1{$client} tcp from $client to $host \
      $dport out xmit $List_int{$client}";
 $i=$i+1;
 $cmd_2 = "ipfw add $i queue $List_queue_2{$client} tcp from $host $dport to \
     $client in recv $List_int{$client}";
  system($cmd_1);
  system($cmd_2);
  if ($i == '2000') {
        system("/use/local/shaper/flush_http_rule");
        $i=1000;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
   }
  }
 }
 undef $client; undef $sport; undef $host; undef $dport; undef $packet;
($client,$sport,$host,$dport) = /(\d+\.\d+\.\d+\.\d+)+\.(\d+)+ >
(\d+\.\d+\.\d+\.\d+)+\.(\d+)/
if /P \d+:\d+\((\d+)\)/ && $1 > 0;
   }
   next unless $client && $host;
   s/\s+//;
   s/([0-9a-f]{2})\s?/chr(hex($1))/eg;
   tr/\x1F-\x7E\r\n//cd;
   $packet .= $_;
}
#-------------------------------------------------------
```

7.2 ชุดคำสั่งเชลสครีป init_shaper

ชุดคำสั่งเชลสตรีป init_shaper ใช้สำหรับสั่งงานโปรแกรม http_sniffer ให้ทำงานหรือหยุดการทำงานโดยมีวิธีการสั่งงานเช่น init_shaper start สั่งให้โปรแกรม http_sniffer ทำงาน และ init_shaper stop สั่งให้โปรแกรม http_sniffer หยุดการทำงาน เป็นต้น

```
#! /bin/sh
. /etc/rc.subr
#--------------------------------------
case "$1" in
 start)
      /usr/local/shaper/http_sniffer &
      ;;
 stop)
      killall -9 http_sniffer > /dev/null && echo 'http_sniffer stopped.'
      killall -9 tcpdump > /dev/null && echo 'tcpdump stopped.'
```

เอกสารนี้เป็นเอกสาร[illegible]ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ;;
restart)
    $0 stop
    $0 start
    ;;
status)
ps -auxww | egrep http_sniffer | egrep -v "($0|egrep)"
ps -auxww | egrep tcpdump | egrep -v "($0|egrep)"
;;
*)
    echo "Usage: `basename $0` {start|stop|restart|status}" >&&
    ;;
esac
exit 0
```

7.3 ชุดคำสั่งเชลสคริป apply_rule

ชุดคำสั่งเชลสตรีป apply_rule ใช้สำหรับตรวจสอบว่าระบบเว็บอินเทอร์เฟสมีการสร้างชุดคำสั่งขึ้นใหม่หรือไม่ โดยคอยตรวจสอบชุดคำสั่งที่โฟลเดอร์ /usr/local/shaper/new_cnd ซึ่งคำสั่ง apply_rule จะถูกสั่งงานผ่าน crontab ให้คอยตรวจสอบทุกนาทีตลอดเวลาที่เปิดเครื่องและเมื่อพบชุดคำสั่งใหม่แล้วจะสั่งให้ทำงานแล้วย้ายที่เก็บไปไว้ที่ /usr.local/shaper/current_cmd

```
#!/usr/local/bin/bash
# PIPE
if [ -f /usr/local/shaper/new_cmd/pipe.cmd ]; then
  for i in { `ipfw pipe show | grep /s | cut -f1 -d:` }
  do
   /sbin/ipfw pipe delete $i
  done
  /usr/local/shaper/new_cmd/pipe.cmd
  mv /usr/local/shaper/new_cmd/pipe.cmd /usr/local/shaper/current_cmd/
fi
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# QUEUE
if [ -f /usr/local/shaper/new_cmd/queue.cmd ]; then
  for i in { `ipfw queue show | grep ^q | cut -f1 -d: | cut -f2 -d"q"` }
  do
   /sbin/ipfw queue delete $i
  done
  /usr/local/shaper/new_cmd/queue.cmd
  mv /usr/local/shaper/new_cmd/queue.cmd /usr/local/shaper/current_cmd/
fi
# RULE
if [ -f /usr/local/shaper/new_cmd/rule.cmd ]; then
  for i in { `ipfw show | grep ^02 | cut -f1 -d" "` }
  do
   /sbin/ipfw delete $i
  done
  /usr/local/shaper/new_cmd/rule.cmd
  mv /usr/local/shaper/new_cmd/rule.cmd /usr/local/shaper/current_cmd/
fi
# HTTP
if [ -f /usr/local/shaper/new_cmd/http.cmd ]; then
  for i in { `ipfw show | grep ^01 | cut -f1 -d" "` }
  do
   /sbin/ipfw delete $i
  done
  /usr/local/shaper/new_cmd/http.cmd
  mv /usr/local/shaper/new_cmd/http.cmd /usr/local/shaper/current_cmd/
fi
# Shutdown
if [ -f /usr/local/shaper/new_cmd/shutdown.cmd ]; then
  cd /usr/local/shaper/new_cmd
  /usr/local/shaper/new_cmd/shutdown.cmd;rm shutdown.cmd
fi
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# Reboot
if [ -f /usr/local/shaper/new_cmd/reboot.cmd ]; then
  cd /usr/local/shaper/new_cmd
  /usr/local/shaper/new_cmd/reboot.cmd;rm reboot.cmd
fi
```

7.4 ชุดคำสั่งเชลสคริป flush_http_rule

ชุดคำสั่งเชลสตรีป flush_http_rule ใช้สำหรับลบรายการของกฎที่โปรแกรม http_sniffer สร้างขึ้น โดยเชลสตรีปนี้จะถูกเรียกใช้งานภายในจากโปรแกรม http_sniffer ดังนั้นทุกครั้งที่ crontab สั่ง init_shaper restart ในทุก 30 นาทีเชลสตรีปจะถูกเรียกใช้งานเพื่อลบกฎในระบบ

```
#!/usr/local/bin/bash
# Delete Rule 1000-2000
  for i in { `ipfw show | grep ^01 | cut -f1 -d" "` }
  do
    /sbin/ipfw delete $i
  done
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

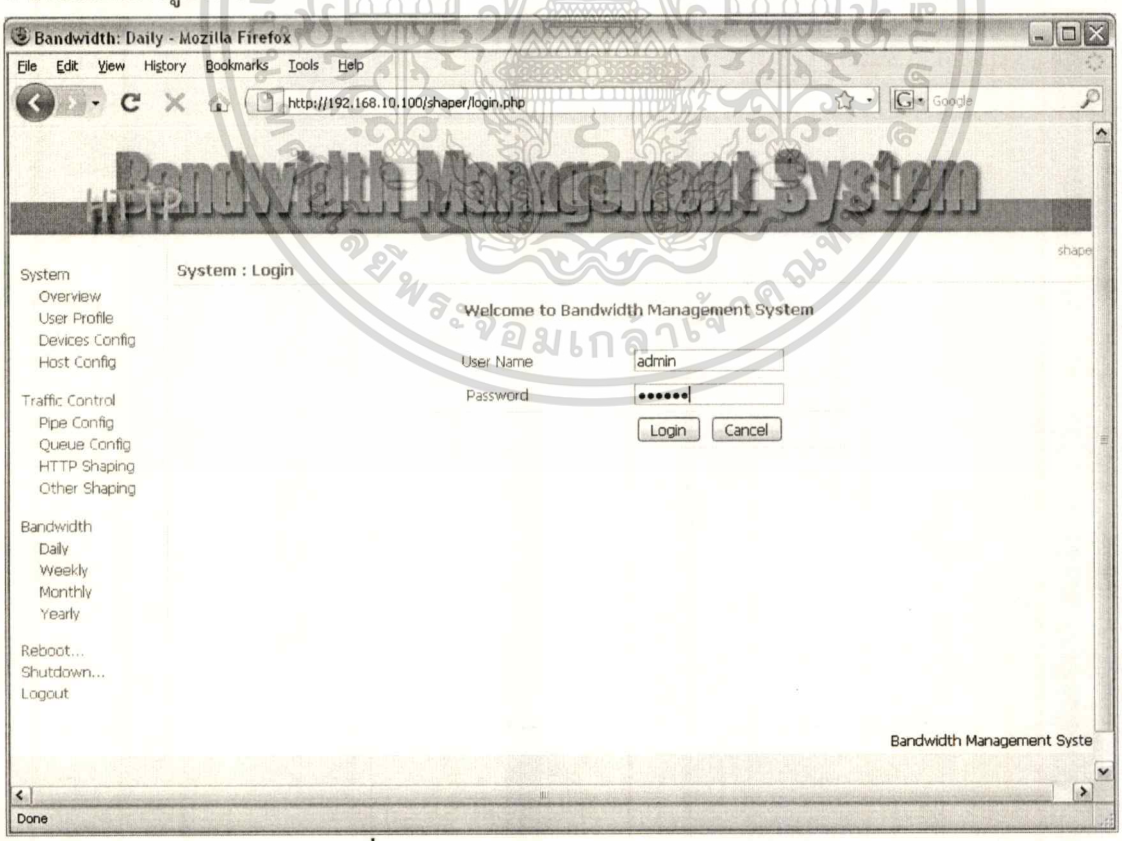
คู่มือการใช้งานระบบควบคุมปริมาณการสื่อสารที่ส่งผ่านด้วยโปรโตคอล HTTP

การใช้งานระบบควบคุมปริมาณการสื่อสารที่ส่งผ่านด้วยโปรโตคอล HTTP

การใช้งานระบบควบคุมปริมาณการสื่อสารที่ส่งผ่านด้วยโปรโตคอล HTTP ได้มีการออกแบบและพัฒนาระบบเว็บอินเทอร์เฟสขึ้นรองรับการทำงานในการควบคุมระบบทั้งหมดเพื่อให้เกิดความสะดวกในการนำไปใช้งาน โดยมีรายละเอียดการใช้งานดังต่อไปนี้

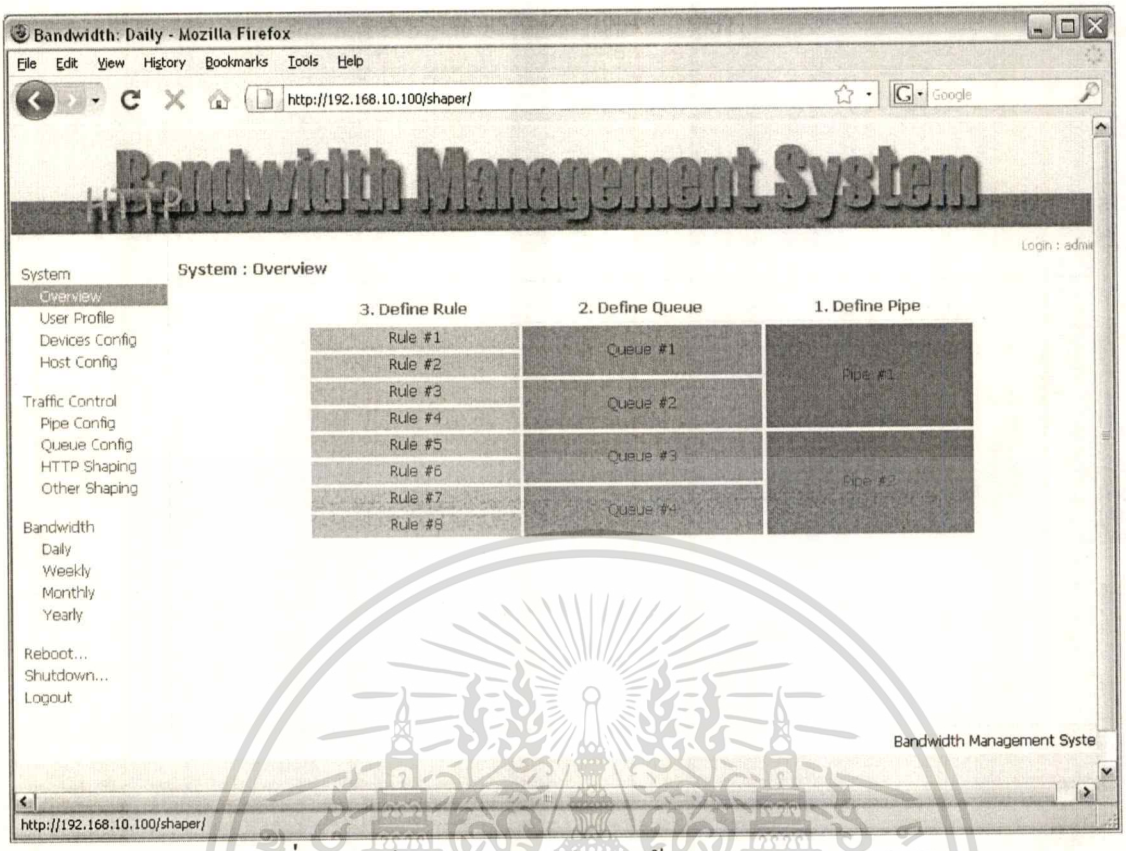
1. การกำหนดค่าที่สำคัญของระบบ

เมื่อใช้บราวเซอร์เชื่อมต่อเข้าระบบควบคุมจะพบหน้าจอภาพของการควบคุมสิทธิ์ด้วยการลอกอินเข้าระบบดังรูปที่ ข-1 เมื่อใส่ชื่อ ลอกอินและรหัสผ่านจากนั้นคลิกปุ่ม Login จะเข้าสู่หน้าจอภาพดังรูปที่ ข-2



รูปที่ ข-1 หน้าจอภาพการลอกอินเข้าระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

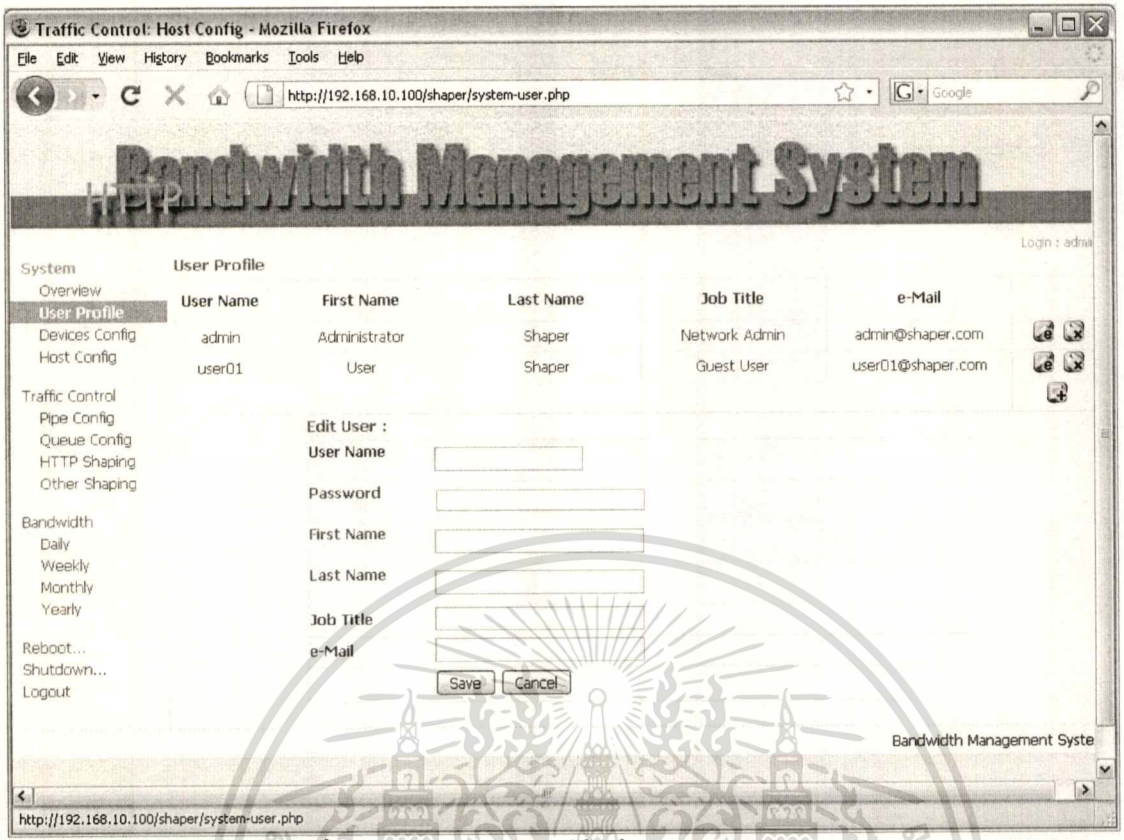


รูปที่ ข-2 หน้าจอภาพอธิบายโครงสร้างการควบคุมระบบ

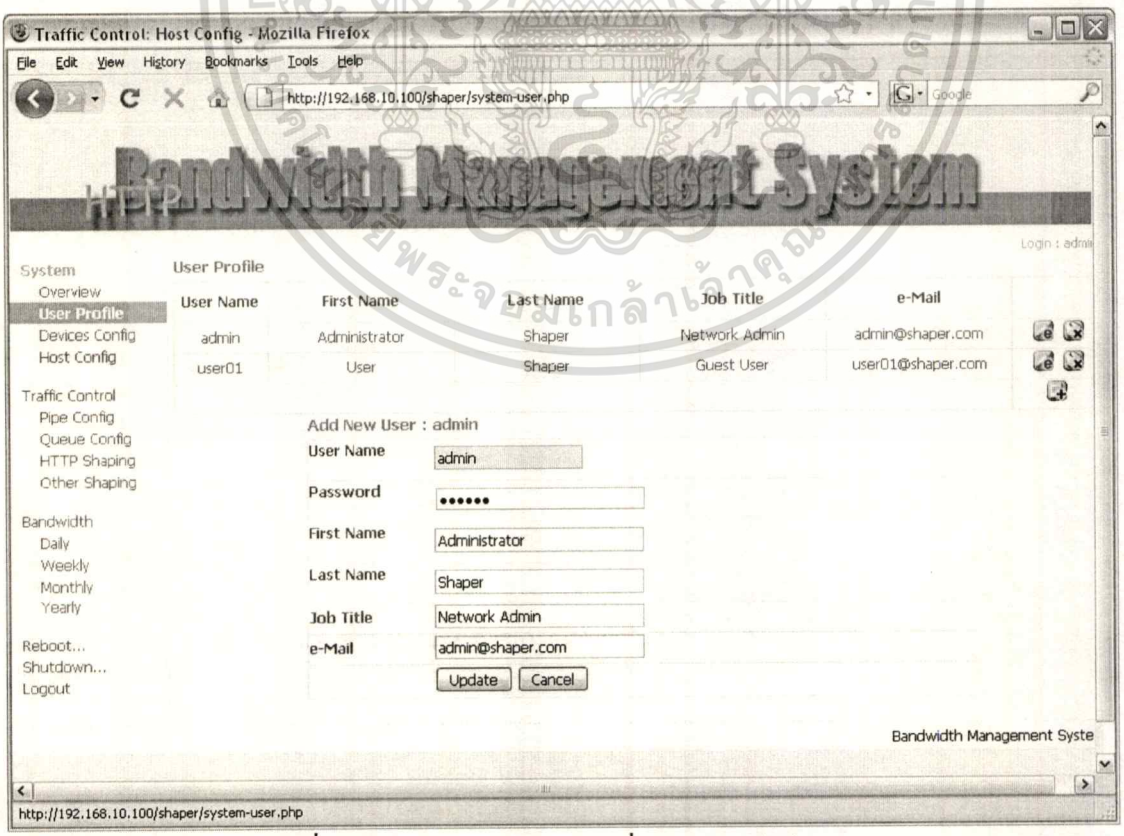
รูปที่ ข-2 เป็นการอธิบายโครงสร้างการควบคุมระบบโดยระบบออกแบบให้มีการสร้าง pipe ที่รองรับการสื่อสารสำหรับการควบคุมแบนด์วิทระบบเป็นอันดับแรกจากนั้นก็สร้างระบบ queue เชื่อมต่อเข้ากับระบบ pipe สุดท้ายเป็นการสร้างกฎต่างๆ ที่ต้องการใช้ในการควบคุมระบบ โดยกฎในที่นี้หมายรวมทั้งกฎของการควบคุมการทำงานของโปรโตคอล HTTP และกฎของการควบคุมทราฟฟิกประเภทอื่นๆ

การจัดการทะเบียนผู้ใช้ระบบมีหน้าจอภาพของการกำหนดทะเบียนชื่อผู้ใช้งานระบบที่ประกอบด้วยส่วนย่อยได้แก่ การเพิ่มทะเบียนชื่อผู้ใช้ระบบ ดังรูปที่ ข-3 การเรียกใช้หน้าจอภาพทำได้โดยคลิกที่ จากนั้นป้อนค่าข้อมูลที่จำเป็นสำหรับระบบจากนั้นก็จัดเก็บข้อมูลโดยคลิกที่ปุ่ม Save ถัดมาเป็นการแก้ไขทะเบียนชื่อผู้ใช้ระบบคลิกที่ จะได้หน้าจอภาพดังรูปที่ ข-4 ซึ่งจะนำรายละเอียดทั้งหมดของผู้ใช้ระบบขึ้นมาให้แก้ไข โดยยกเว้นชื่อผู้ใช้ระบบ (User Name) ที่ไม่สามารถแก้ไขได้ด้วยเป็นคีย์หลักของระบบ และถ้าต้องการลบทะเบียนชื่อผู้ใช้ระบบสามารถคลิกที่ปุ่ม จะปรากฏของการยืนยันการลบข้อมูลออกจากระบบดังรูปที่ ข-5 โดยถ้าต้องการยืนยันการลบข้อมูลก็คลิกปุ่ม OK ถ้าต้องการยกเลิกคลิกปุ่ม Cancel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

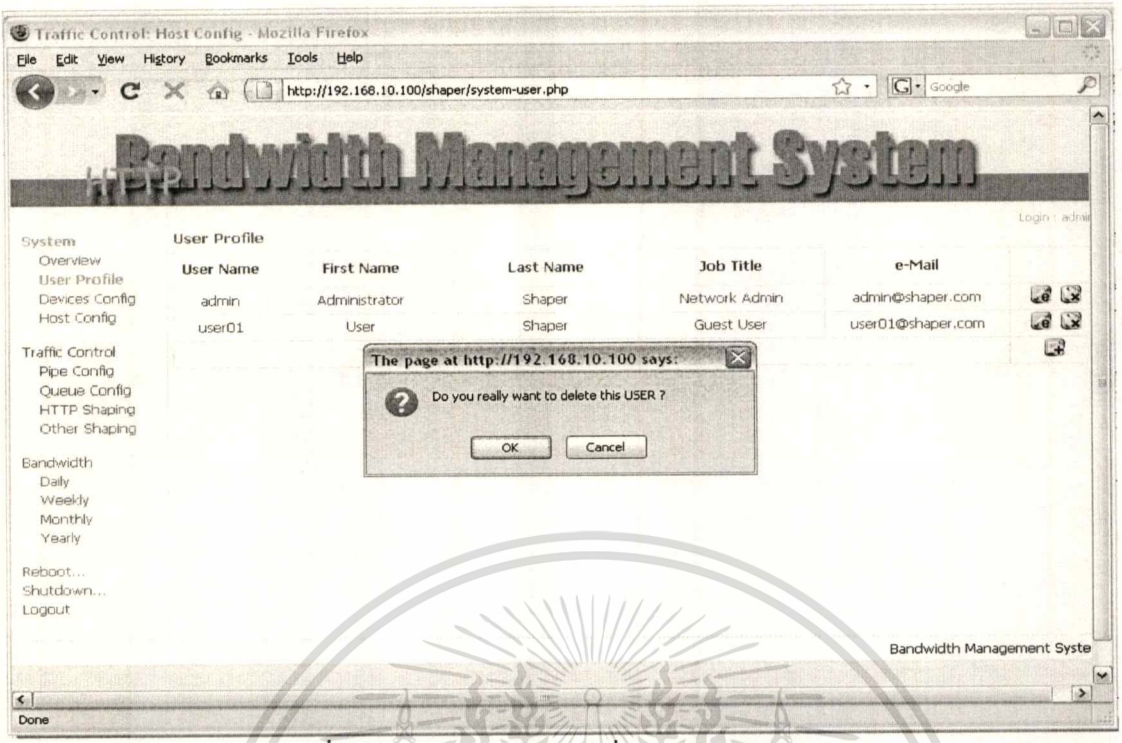


รูปที่ ข-3 หน้าจอภาพการเพิ่มชื่อทะเบียนผู้ใช้ระบบ



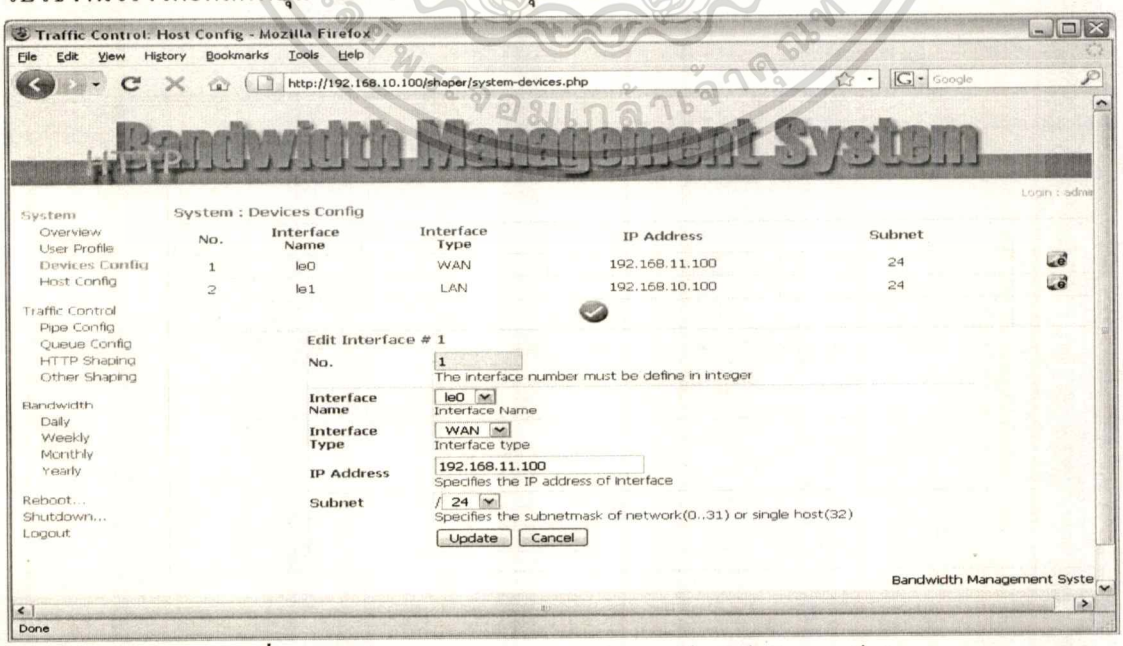
รูปที่ ข-4 หน้าจอภาพการแก้ไขชื่อทะเบียนผู้ใช้ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



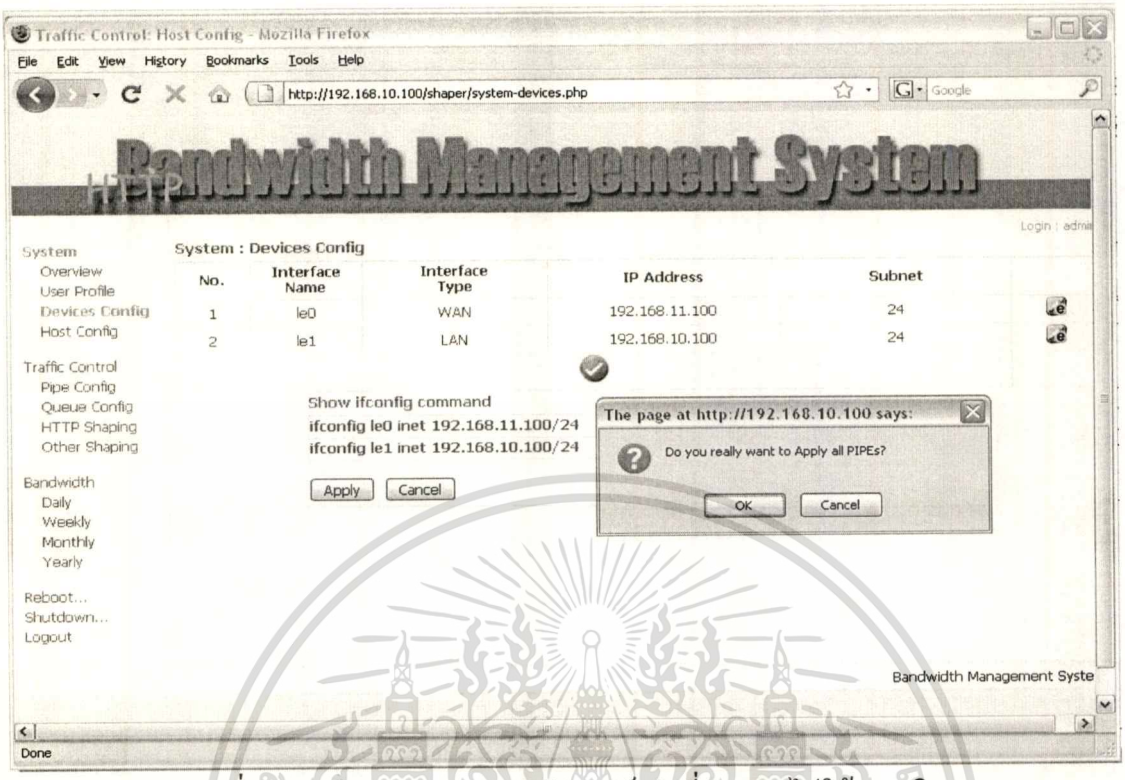
รูปที่ ข-5 หน้าจอภาพการลบชื่อทะเบียนผู้ใช้ระบบ

การจัดการเน็ตเวิร์กอินเทอร์เฟสของระบบสามารถคลิกที่ปุ่ม ที่หน้าจอภาพ ดังรูปที่ ข-6 เพื่อแก้ไขค่าต่างของระบบที่สำคัญได้แก่ค่าไอพีแอดเดรส (IP Address) ชื่ออินเทอร์เฟส (Interface Name) โดยต้องเป็นชื่อจริงที่มีในระบบโดยทั่งไประบบปฏิบัติการ FreeBSD จะกำหนดชื่อเป็น le0,le1 ซึ่งในที่นี้จะกำหนดไว้แล้วโดยเลือกจากระบบที่มี รูปที่ ข-7 เป็นการนำค่าที่กำหนดไปใช้งานจริงโดยคลิกที่ปุ่ม จากนั้นคลิกปุ่ม OK



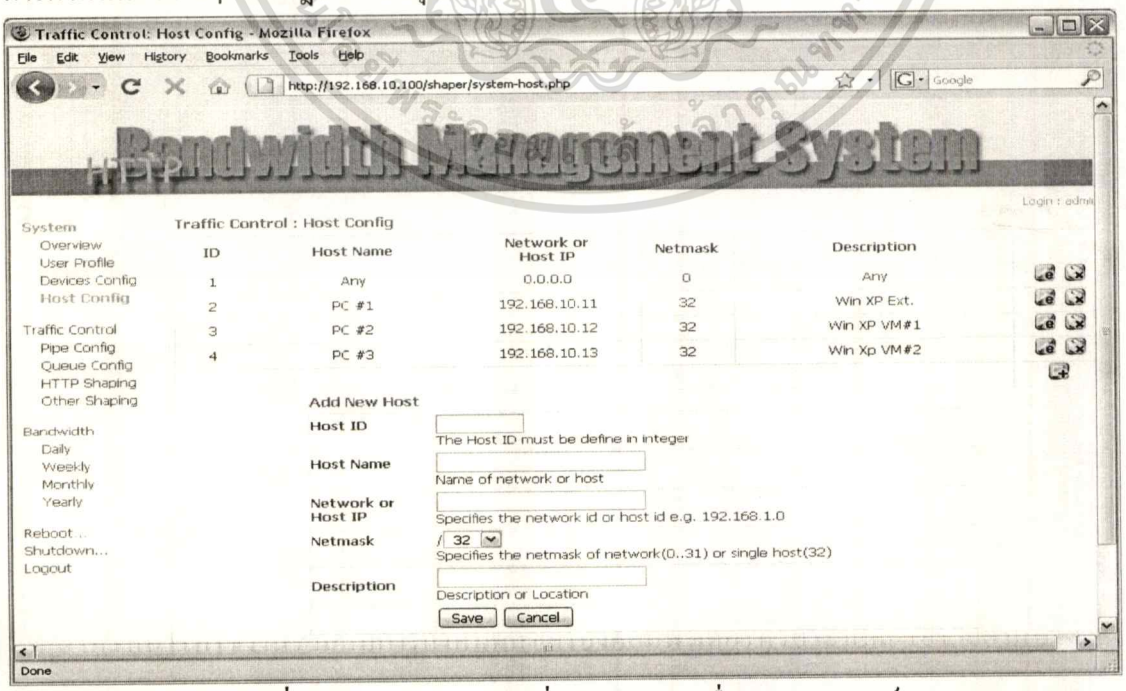
รูปที่ ข-6 หน้าจอภาพการกำหนดค่าของเน็ตเวิร์กอินเทอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



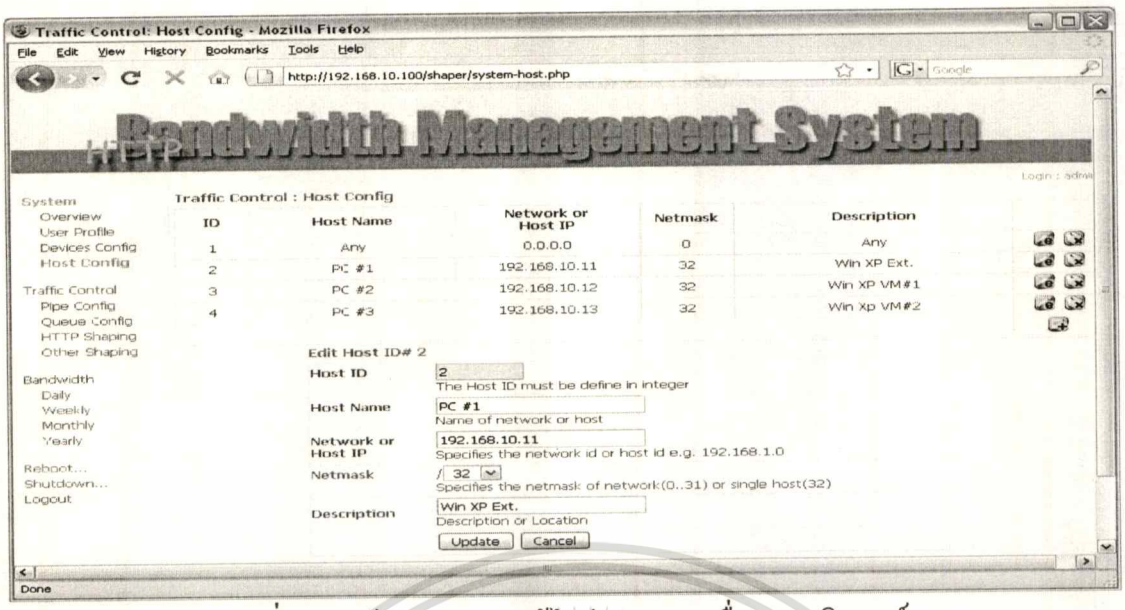
รูปที่ ข-7 หน้าจอภาพการนำค่าอินเทอร์เฟสที่กำหนดไปใช้งานจริง

การกำหนดทะเบียนรายการเครื่องคอมพิวเตอร์และระบบเครือข่ายที่ต้องการติดต่อใช้งานสำหรับนำไปใช้งานรวมกับการกำหนดค่าเครื่องคอมพิวเตอร์หรือระบบเครือข่ายต้านทางของการกำหนดค่าต่างๆ ของกฎการควบคุมแบนด์วิทของระบบ

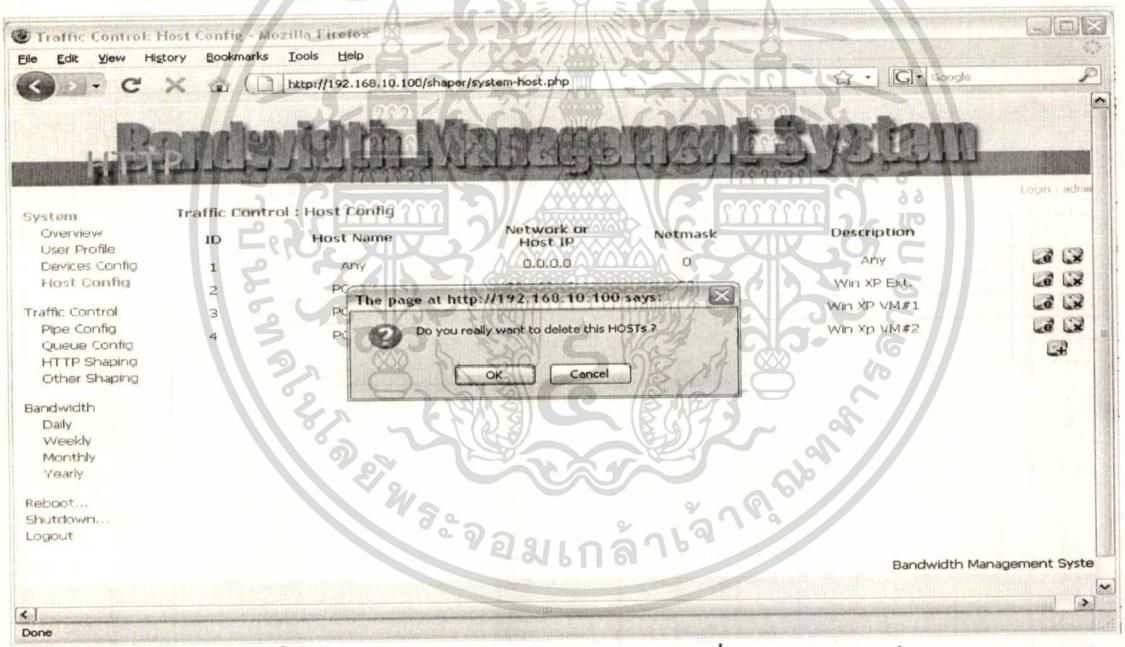


รูปที่ ข-8 หน้าจอภาพการเพิ่มค่ารายการเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-9 หน้าจอภาพการแก้ไขค่ารายการเครื่องคอมพิวเตอร์



รูปที่ ข-10 หน้าจอภาพการลบค่ารายการเครื่องคอมพิวเตอร์

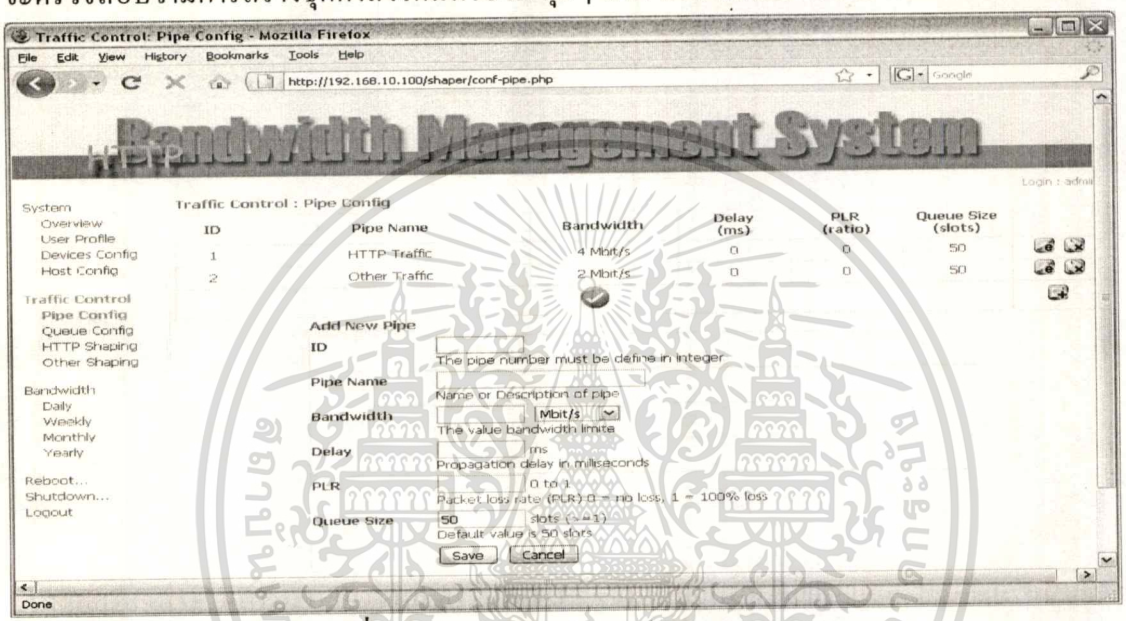
2. การควบคุมทราฟฟิกของระบบ

การควบคุมทราฟฟิกของระบบเป็นขั้นตอนการกำหนดค่าที่สำคัญต่างๆ ที่จำเป็นต่อการควบคุมโดยถือเป็นหัวใจหลักของระบบควบคุมทั้งหมดโดยประกอยด้วย

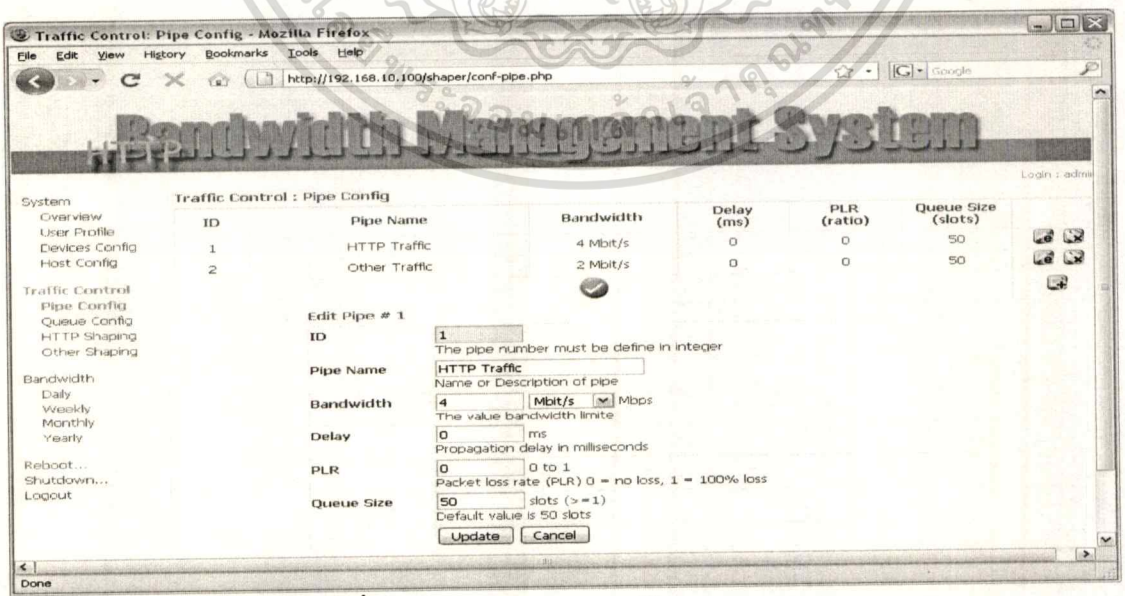
การกำหนดขนาดของแบนด์วิทที่ต้องการควบคุมหรือการกำหนดขนาดแบนด์วิทของ pipe โดยระบบสามารถกำหนดจำนวน pipe ที่ต้องการใช้งานได้มากกว่าหนึ่ง pipe ซึ่งการทำงานของ pipe แต่ละชุดจะมีความเป็นอิสระต่อกัน โดยมีขั้นตอนดังนี้

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข-11 เป็นหน้าจอภาพการสร้างรายการ pipe เข้าไปในระบบโดยค่าที่สำคัญที่ต้องกำหนดคือค่าของ ID และขนาดของแบนด์วิทที่ต้องการควบคุม ส่วนรูปที่ ข-12 เป็นการแก้ไขค่าของ pipe ที่กำหนดไว้แล้ว ส่วนรูปที่ ข-13 เป็นการลบรายการ pipe ที่ไม่ต้องการออกจากระบบ เมื่อต้องการนำค่าหรือคุณสมบัติต่างของ pipe ไปใช้งานจริง สามารถคลิกที่ปุ่ม จะปรากฏรายการของชุดคำสั่งที่ใช้ในการสั่งงานระบบ ดังรูปที่ ข-14 จากนั้นคลิกปุ่ม Apply และยืนยันคำสั่งด้วยการคลิกปุ่ม OK ระบบจะทำการสร้างชุดคำสั่งไปรอไว้ให้ชุดคำสั่งใน crontab เรียกใช้งานโดย corntab จะตรวจสอบว่ามีการสร้างชุดคำสั่งใหม่หรือไม่ ทุกๆ นาที ดังนั้นภายในหนึ่งนาทีระบบจะทำงาน

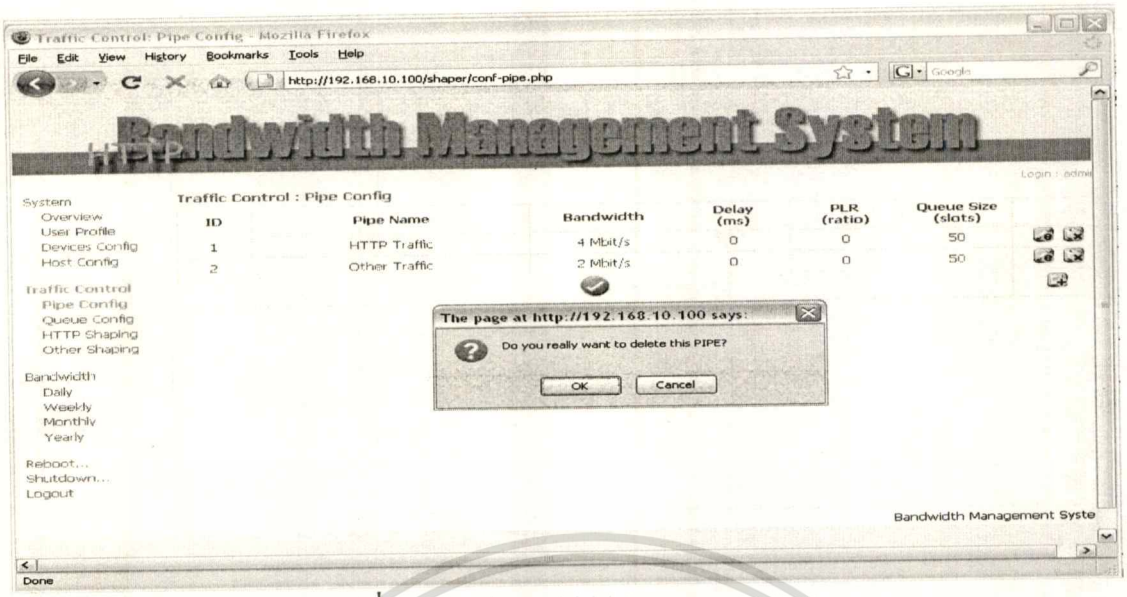


รูปที่ ข-11 หน้าจอภาพการสร้างรายการ pipe



รูปที่ ข-12 หน้าจอภาพการแก้ไขรายการ pipe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



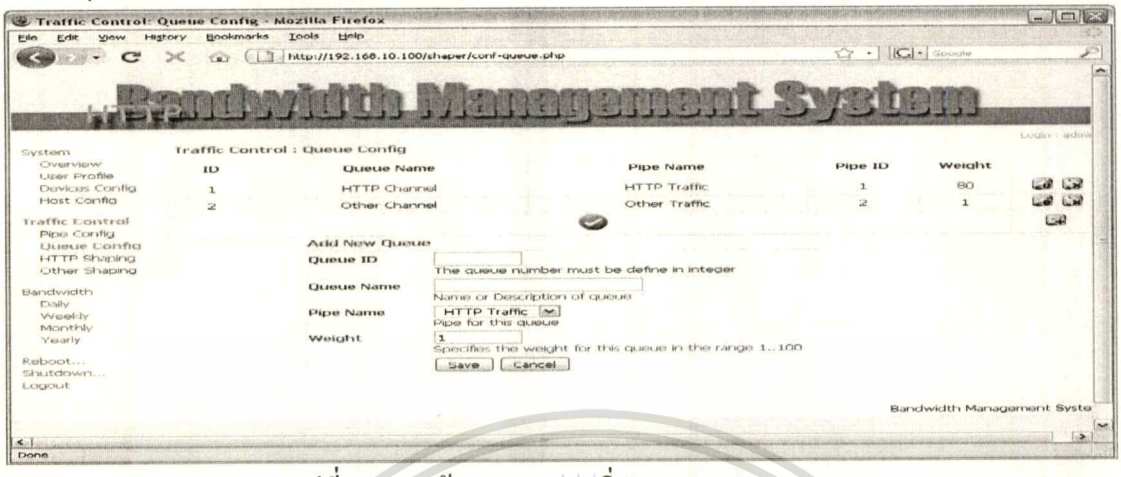
รูปที่ ข-13 หน้าจอภาพการลบรายการ pipe

รูปที่ ข-14 หน้าจอภาพการนำค่าของรายการ pipe ไปใช้งานจริง

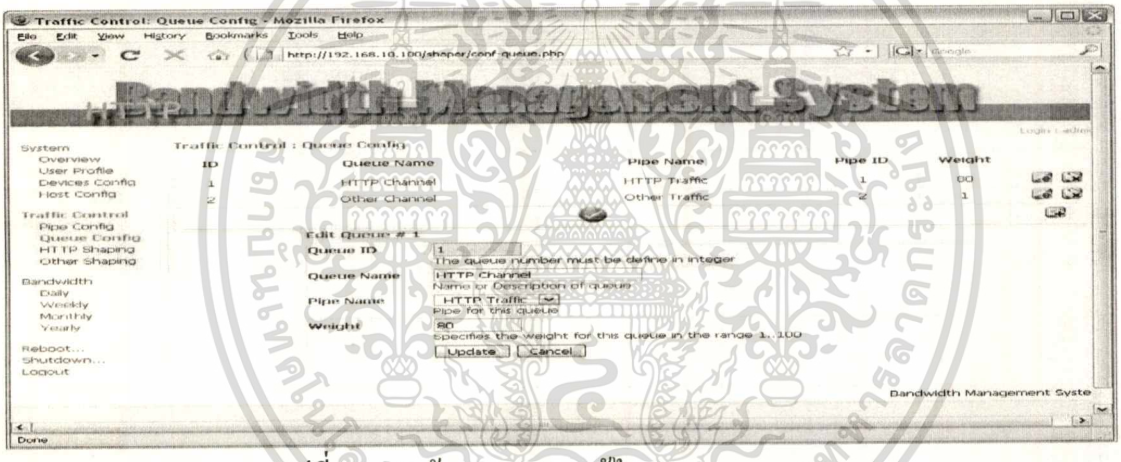
การกำหนดค่าของแถวคอยหรือการกำหนดค่า queue ที่จะต่อเชื่อมกับระบบ pipe ดังแสดงในรูปที่ ข-15 เป็นการเพิ่มรายการของ queue โดยส่วนที่สำคัญคือการกำหนดให้ queue ที่สร้างขึ้นเชื่อมต่อกับระบบ pipe รายการใดซึ่งระบบจะนำรายการ pipe ที่สร้างไว้ขึ้นมาให้เลือก และเมื่อบันทึกรายการไปแล้วสามารถเรียกกลับมาทำการแก้ไขใหม่ได้ดังแสดงดังรูปที่ ข-16 จะแสดงรายการทั้งหมดที่สำคัญของ queue ที่สามารถแก้ไขได้ และถ้าต้องการลบรายการ queue ออกจากระบบสามารถทำได้ ดังแสดงในรูปที่ ข-17 โดยถ้าต้องการยืนยันการลบคลิก OK จากนั้นเป็นส่วนที่สำคัญคือการนำค่ารายการของ queue ไปใช้งานจริง ดังแสดงในรูปที่ ข-18 จะแสดงรายการคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

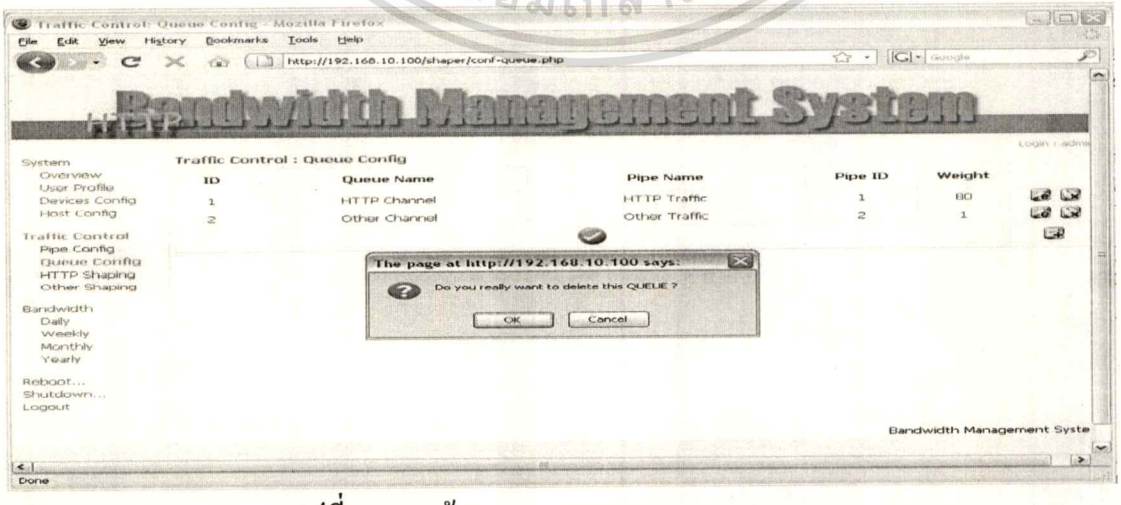
ที่ระบบแปลงจากรายการของ queue ที่กำหนด โดยถ้าต้องการสั่งให้ทำงานตามรายการของ queue คลิกที่ปุ่ม Apply และยืนยันด้วยคลิก OK



รูปที่ ข-15 หน้าจอภาพการเพิ่มรายการของ queue

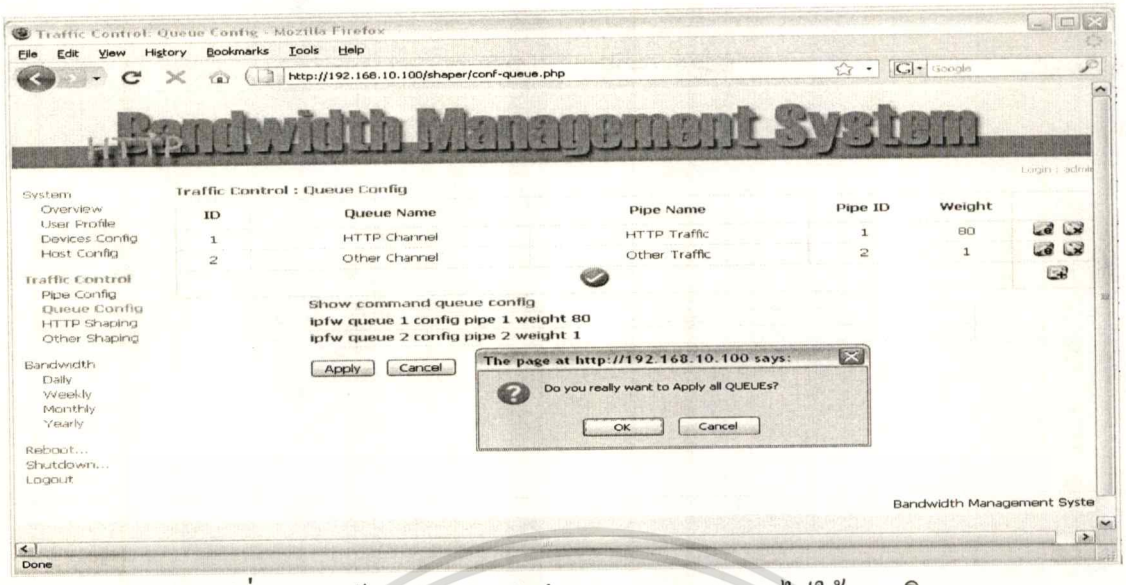


รูปที่ ข-16 หน้าจอภาพการแก้ไขรายการของ queue



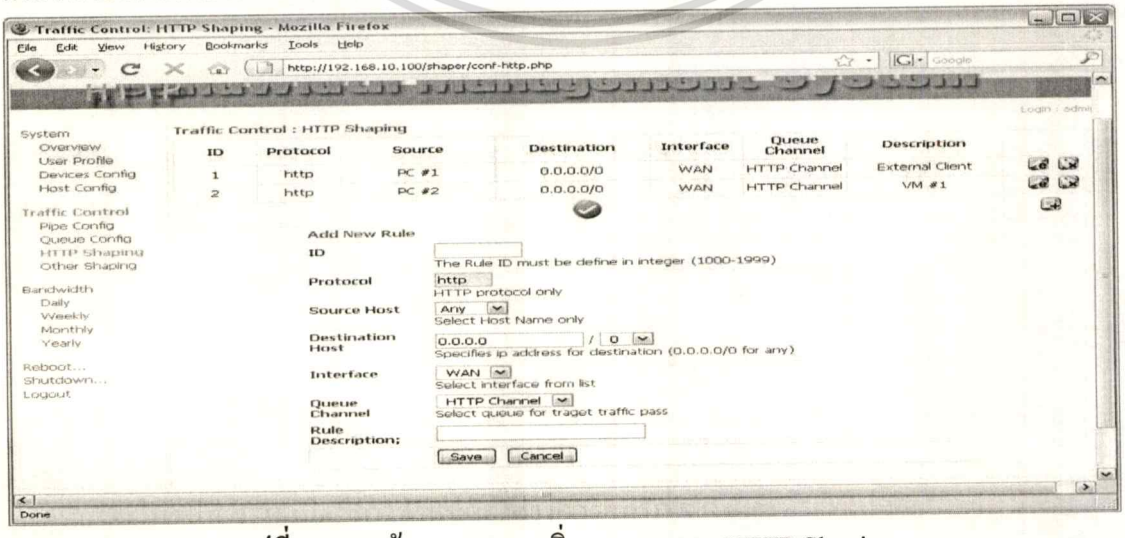
รูปที่ ข-17 หน้าจอภาพการลบรายการของ queue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



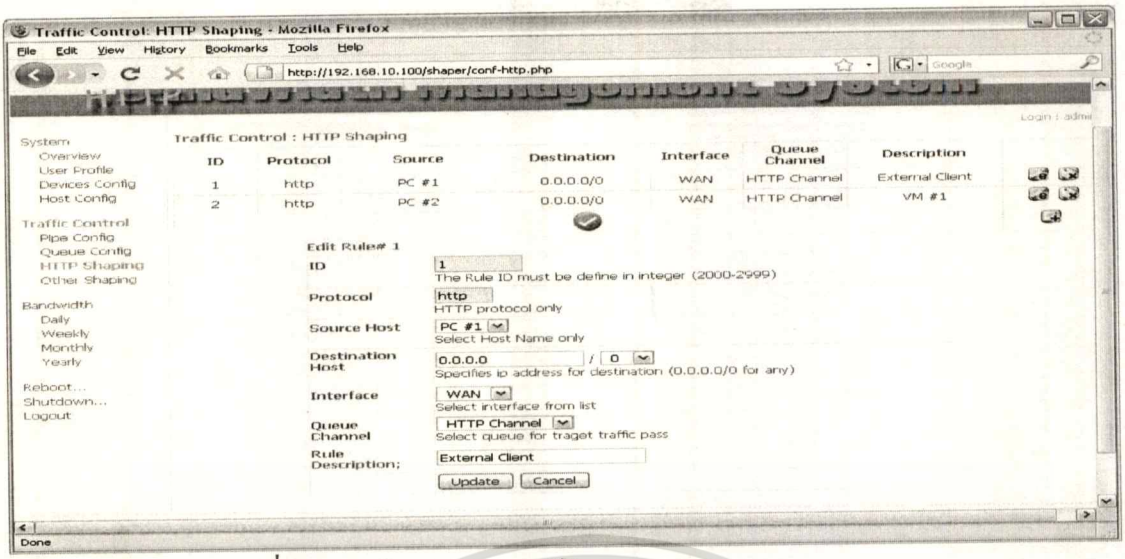
รูปที่ ข-18 หน้าจอภาพการนำค่าของรายการ queue ไปใช้งานจริง

การกำหนดค่าของการควบคุมทราฟฟิกของโปรโตคอล HTTP ดังแสดงในรูปที่ ข-19 เป็นการสร้างรายการควบคุมใหม่ส่วนที่ต้องกำหนดคือ Source Host หรือรายการเครื่องต้นทางที่ต้องการควบคุมแบนด์วิท และช่องทางที่ต้องการใช้ในการควบคุมหรือหมายเลข queue ที่เชื่อมต่ออยู่กับ pipe ที่ต้องการควบคุมซึ่งรายการของ queue จะมีรายการให้เลือกจากรายการที่สร้างไว้ เมื่อกรอกรายการเรียบร้อยแล้วจึงบันทึก (Save) และเมื่อต้องการแก้ไขค่าที่กำหนดไว้แล้วสามารถแก้ไขได้ดังแสดงในรูปที่ ข-20 แสดงการเรียกรายการ ID ขึ้นมาแก้ไขค่า หรือถ้าต้องการลบค่าสมารถทำได้ดังแสดงในรูปที่ ข-21 เมื่อมีการกำหนดค่าต่างๆ เรียบร้อยต้องการนำค่าไปสั่งการใช้งานจริงต้องคลิกที่ เพื่อแสดงรายการชุดคำสั่งที่ต้องการสั่งงานดังแสดงในรูปที่ ข-22 และเมื่อต้องการสั่งก็คลิกที่ Apply และยืนยันการสั่งงานระบบจะสร้างชุดคำสั่งไปรอให้ชุดคำสั่งใน crontab สั่งงานตามคำสั่งที่ต้องการภายในเวลาหนึ่งนาที

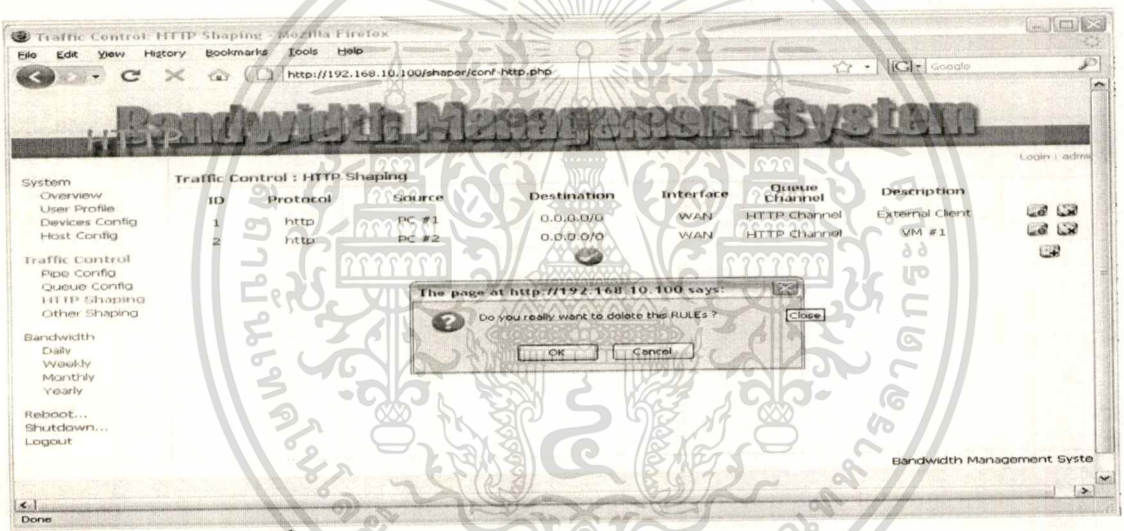


รูปที่ ข-19 หน้าจอภาพการเพิ่มของรายการ HTTP Shaping

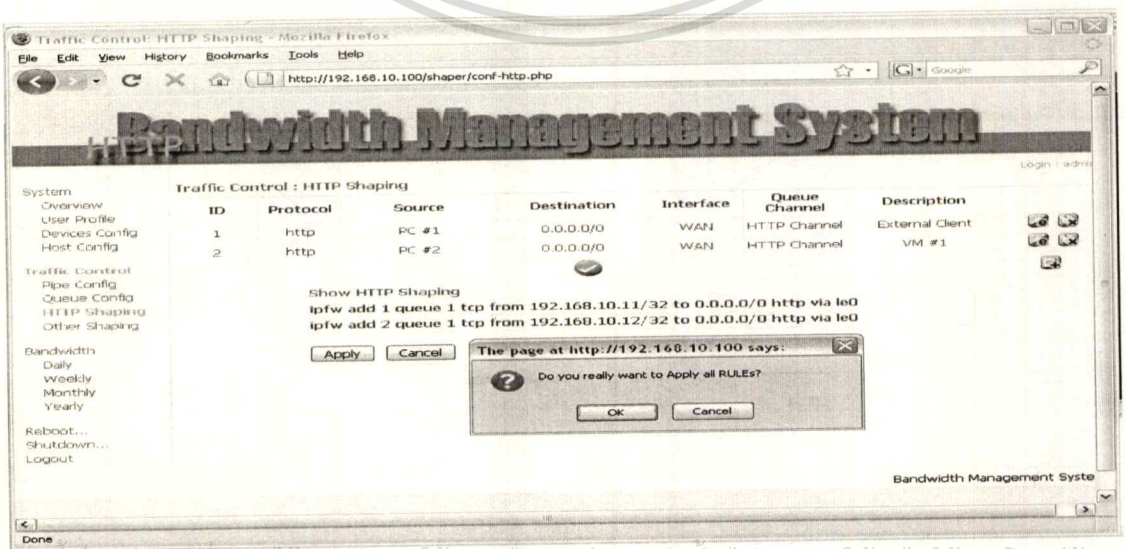
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-20 หน้าจอภาพการแก้ไขของรายการ HTTP Shaping



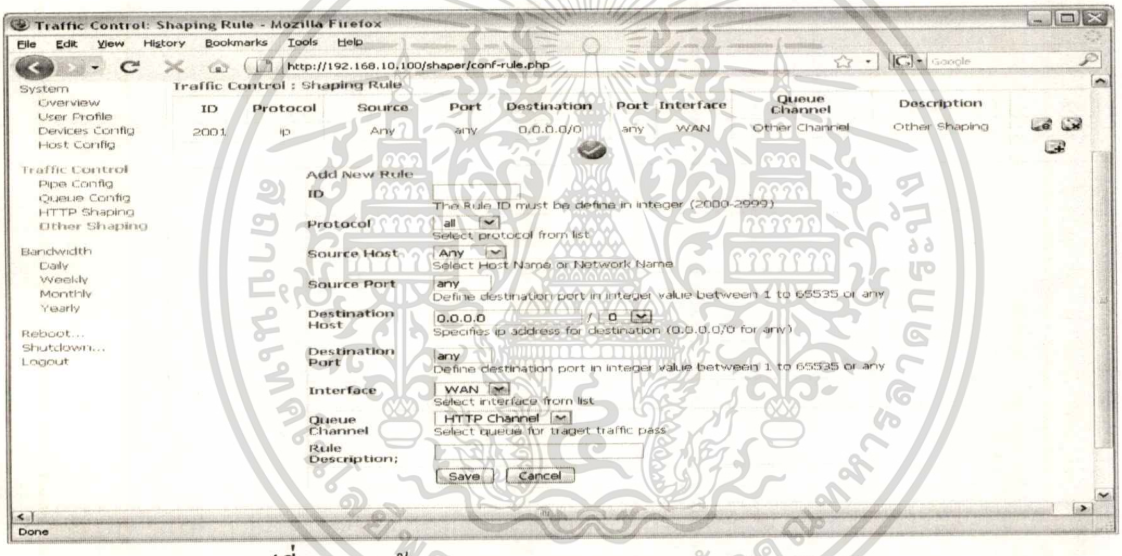
รูปที่ ข-21 หน้าจอภาพการลบของรายการ HTTP Shaping



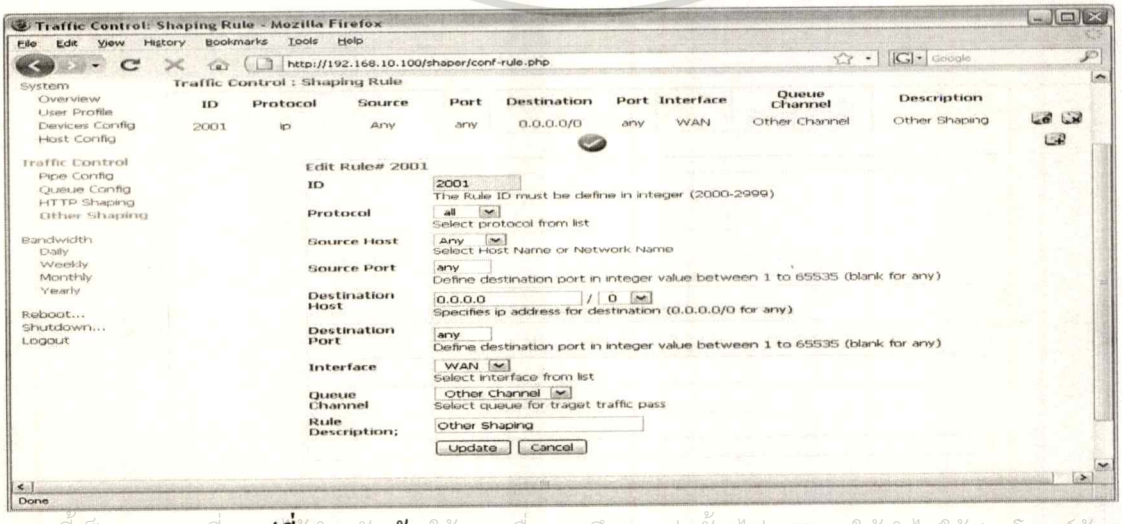
รูปที่ ข-22 หน้าจอภาพการนำค่าของรายการ HTTP Shaping ไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบนด์วิทของการสื่อสารประเภทอื่นๆ ระบบได้ออกแบบให้รองรับการควบคุมแบนด์วิทของการติดต่อสื่อสารในรูปแบบอื่นๆ นอกจากการควบคุมแบนด์วิทของการสื่อสารของโปรโตคอล HTTP โดยมีขั้นตอนการสร้างกฎดังแสดงในรูปที่ ข-23 ที่สำคัญคือการกำหนดเครื่องต้นทางซึ่งเลือกจากรายการเครื่องที่กำหนดไว้และหมายเลขพอร์ตปลายทางซึ่งสำคัญในการควบคุมแบนด์วิทชนิดนี้เพราะพอร์ตปลายทางเป็นการกำหนดประเภทของบริการที่ต้องการควบคุม จากนั้นต้องเลือกรายการ Queue Channel ที่ต้องการส่งต่อทราฟฟิกไปยัง pipe ที่ต้องการควบคุม เมื่อมีการกำหนดค่าต่างๆ เรียบร้อยต้อการนำค่าที่กำหนดไปใช้งานคลิก ระบบจะแสดงรายการข้อกำหนดต่างๆ ของกฎทั้งหมดดังแสดงในรูปที่ ข-26 และเมื่อต้องการนำไปใช้งานจริงคลิกที่ปุ่ม Apply และคลิก OK เพื่อยืนยันระบบจะสร้างชุดคำสั่งไปรอให้ชุดคำสั่งใน crontab สั่งงานตามคำสั่งที่ต้องการภายในเวลาหนึ่งนา

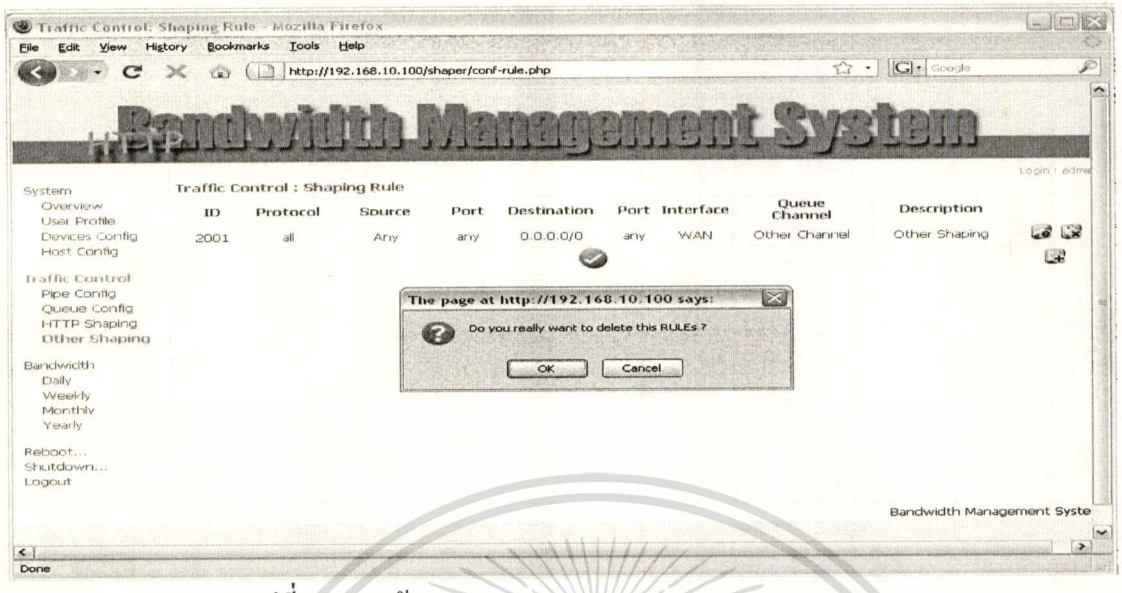


รูปที่ ข-23 หน้าจอภาพการลบของรายการ Other Shaping

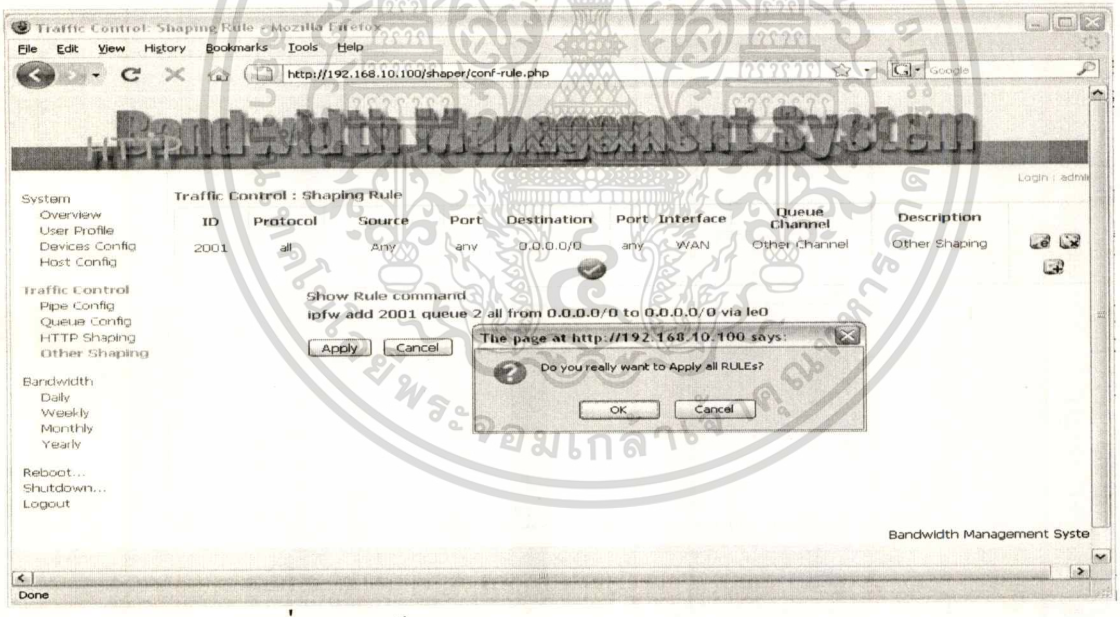


รูปที่ ข-24 หน้าจอภาพการลบของรายการ Other Shaping

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-25 หน้าจอภาพการลบของรายการ Other Shaping

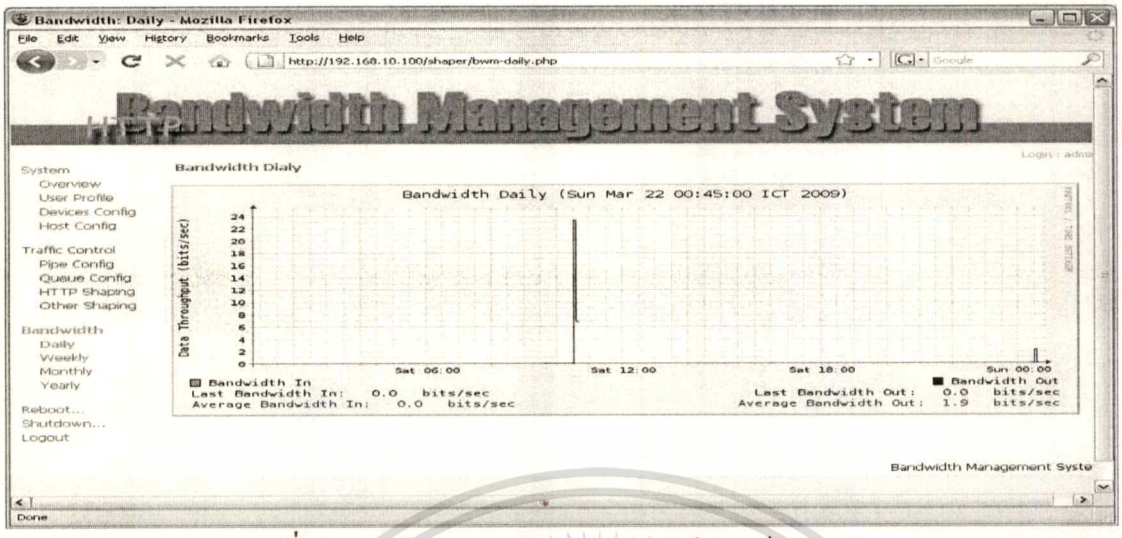


รูปที่ ข-26 หน้าจอภาพการลบของรายการ Other Shaping

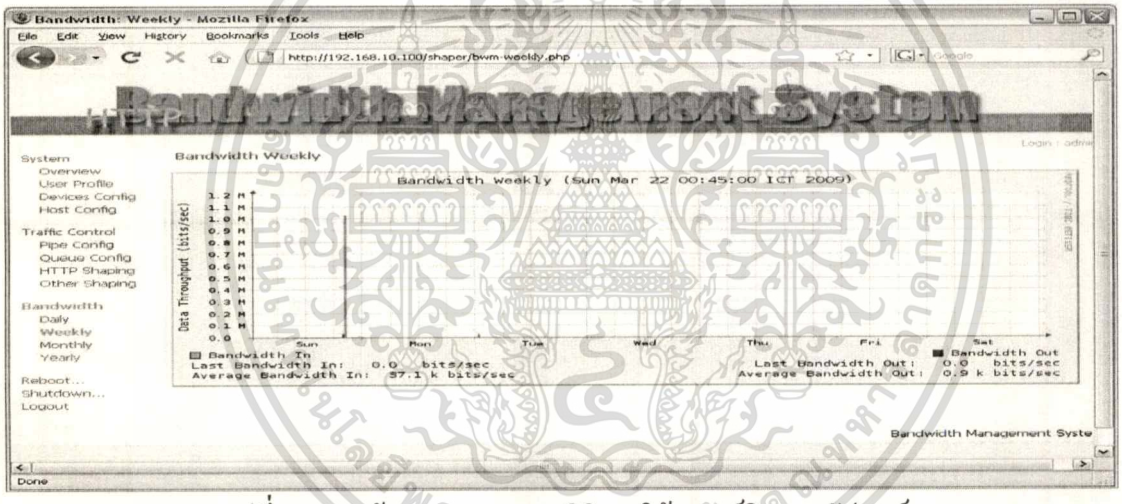
3. การตรวจสอบการใช้งานแบนด์วิท

ระบบที่พัฒนาได้นำเอาระบบการตรวจวัดสถิติการใช้งานทราฟฟิกในระบบเครือข่าย ที่ชื่อว่า RRD Tools มาใช้ในการตรวจวัดค่าสถิติการใช้งานของอินเทอร์เฟส WAN โดยมีการแบ่งรายการแสดงผลการตรวจวัดเป็น รายวัน(Daily) ดังแสดงในรูปที่ ข-27 รายสัปดาห์(Weekly) ดังแสดงในรูปที่ ข-28 รายเดือน(Monthly) ดังแสดงในรูปที่ ข-29 รายปี(Yearly) ดังแสดงในรูปที่ ข-30

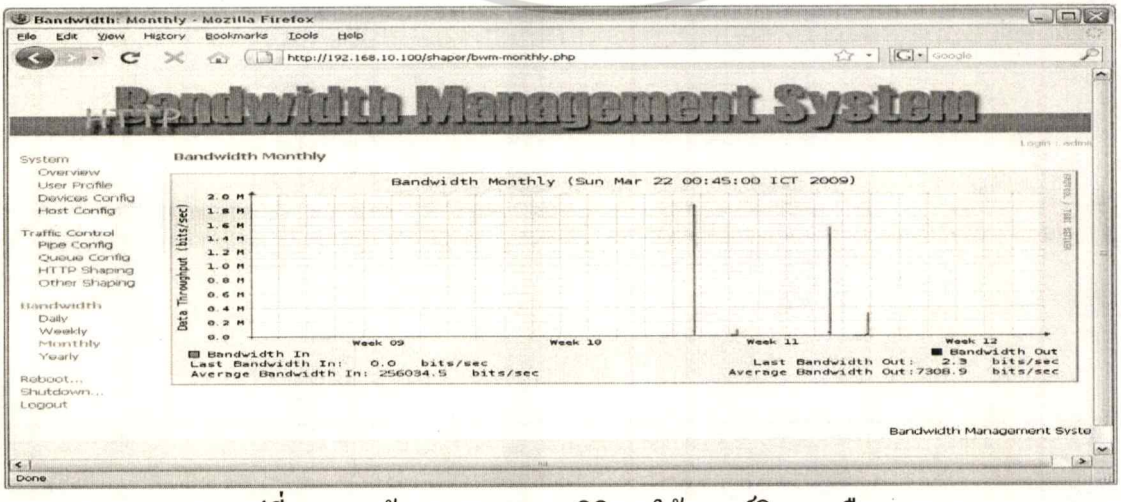
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-27 หน้าจอภาพแสดงสถิติการใช้แบนด์วิทรายวัน

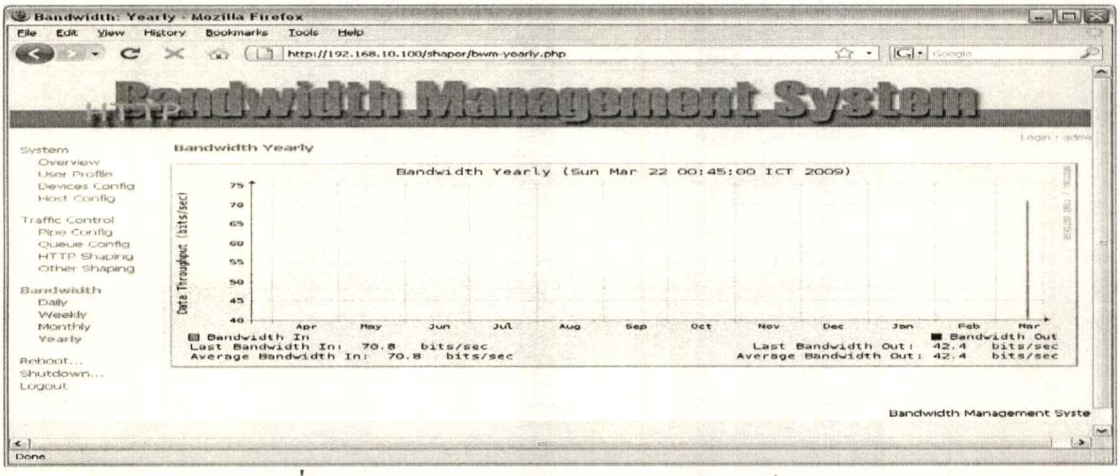


รูปที่ ข-28 หน้าจอภาพแสดงสถิติการใช้แบนด์วิทรายสัปดาห์



รูปที่ ข-29 หน้าจอภาพแสดงสถิติการใช้แบนด์วิทรายเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

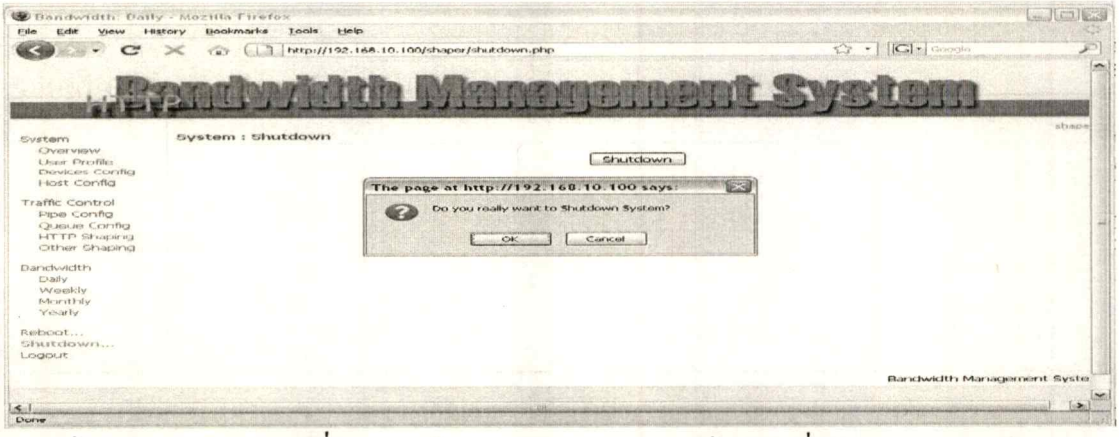


รูปที่ ข-30 หน้าจอภาพแสดงสถิติการใช้แบนด์วิทรายปี

4. การสั่งงานปิด-เปิดและปิดเครื่องเซิร์ฟเวอร์

เป็นการสั่งงานการ Reboot เครื่อง หรือปิดเครื่องเซิร์ฟเวอร์ (Shutdown) ผ่านทางเว็บอินเทอร์เฟส ซึ่งระบบจะมีการยืนยันการสั่งงานดังแสดงในรูปที่ ข-31 และ รูปที่ ข-32

รูปที่ ข-31 หน้าจอภาพแสดงการ Reboot เครื่อง



รูปที่ ข-32 หน้าจอภาพแสดงการ Shutdown เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายบุญลือ จิตตคาม
สถานที่เกิด	พระนครศรีอยุธยา
การศึกษา	ระดับปริญญาตรี
	คบ. (คอมพิวเตอร์ศึกษา)
	คณะครุศาสตร์
	มหาวิทยาลัยราชภัฏพระนครศรีอยุธยา
ประสบการณ์การทำงาน	เจ้าพนักงานเครื่องคอมพิวเตอร์ ชำนาญงาน
	ทำหน้าที่บริหารจัดการระบบคอมพิวเตอร์และเครือข่าย
	ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร
	สำนักงานปลัดกรทรวงการคลัง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้