

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

การพัฒนาระบบควบคุมปริมาณการจราจรของเครือข่ายของ FreeBSD

DEVELOPMENT FOR NETWORK TRAFFIC SHAPING ON
FREEBSD



โดย

สรารุธ สิงห์กุล

SARAWUTH SINGKUL

อาจารย์ที่ปรึกษา

ผศ.อักรินทร์ คุณกิตติ

กพ.
๙๓๕๓
๒๕๕๑

เลขหมู่.....
เลขทะเบียน... 05420
วัน,เดือน,ปี... 1 1 ส.ย. 2551

b. 120922๗๗
i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ภาคเรียนที่ 1 ปีการศึกษา 2551
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**DEVELOPMENT FOR NETWORK TRAFFIC SHAPING ON
FREEBSD**



**A SYSTEM DEVELOPMENT PROJECT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECNOLOGY**

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
1/ 2008

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2008

FACULTY OF INFORMATION TECHNOLOGY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	การพัฒนาระบบควบคุมปริมาณการจราจรของเครือข่ายของ FreeBSD
นักศึกษา	นายสรารุช สิงห์กุล
รหัสนักศึกษา	49066807
ปริญญา	วิทยาศาสตร์มหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2551
อาจารย์ที่ปรึกษา	ผศ.อัครินทร์ คุณกิตติ

บทคัดย่อ

การควบคุมปริมาณการจราจรของเครือข่าย เป็นการบริหารจัดการแบนวิดของระบบเครือข่าย ตามนโยบายของแต่ละองค์กรเพื่อให้เกิดประสิทธิภาพในการใช้งานสูงสุด แต่ในปัจจุบันตัวอุปกรณ์ที่ทำหน้าที่นี้โดยเฉพาะยังมีราคาที่สูง ดังนั้นจึงเกิดความต้องการในการพัฒนาระบบที่มีราคาต่ำแต่ยังคงประสิทธิภาพที่ดี ฟรีบีเอสดี เป็นระบบปฏิบัติการฟรีซอฟต์แวร์ที่มีประสิทธิภาพสูง และมีเครื่องมือที่ใช้ในการจัดการแบนวิด คือ PF (Packet Filter) และ ALTQ (Alternate Queuing) ซึ่งใช้หลักการจัดการระบบแถวคอยแบบต่างๆ ในการทำงาน ทั้งส่วนขาเข้าและขาออกจากระบบ แต่การใช้งานฟรีบีเอสดีนั้น จะเป็นการป้อนคำสั่งเข้าสู่ระบบ ทำให้ไม่เหมาะสมกับการใช้งาน จากผู้ใช้งานที่ไม่มีความเชี่ยวชาญในฟรีบีเอสดี ดังนั้นจึงได้ทำการวิเคราะห์และออกแบบระบบ โดยใช้ UML Model นำเสนอสิ่งที่ได้วิเคราะห์ เพื่อนำมาใช้พัฒนาระบบให้เกิดการใช้งานที่ง่ายขึ้นจากเดิม และใช้ภาษา PHP ในการพัฒนาร่วมกับ Apache เว็บเซิร์ฟเวอร์ ส่วนผลการพัฒนานั้น ระบบที่ได้พัฒนาสามารถทำการจัดการแบนวิดได้ตามหลักการจัดการระบบแถวคอยที่ได้ศึกษามา โดยสามารถควบคุมได้สองแบบคือ Class Base Queuing และ Priority Queuing ซึ่งในอนาคตต่อไปจะได้ใช้เป็นแนวทางในการพัฒนา เพื่อให้สามารถรองรับระบบแถวคอยในรูปแบบต่างๆ ได้มากยิ่งขึ้น

Title	Network Traffic Shaping on FreeBSD
Student	Mr. Sarawuth Singkul
Student ID.	49066807
Degree	Master of Science
Programme	Information Science
Academic Year	2008
Advisor	Asst.Prof. Akharin Khunkitti

ABSTRACT

Network Traffic Control is bandwidth management and control of network traffic for better efficiency with company policy. FreeBSD is open source software that have tools are PF (Packet Filter) and ALTQ (Alternate Queuing) for internal and external bandwidth control with queuing theory such as class base and priority queue algorithm. In analysis and design phase, it represents the system functionality and requirement with UML model. For developing the user interface, web base technology and PHP language programming for Apache web server are used. Lastly the system successes for basic function but can not support all of queue types and advance features. It is challenge for next research.

กิตติกรรมประกาศ

โครงการพัฒนาระบบนี้ ผู้เขียนได้ทำการพัฒนาระบบจนเสร็จสมบูรณ์ โดยได้รับความช่วยเหลือจากหลายท่าน โดยเฉพาะอย่างยิ่งท่านอาจารย์ที่ปรึกษา, ท่านอาจารย์หลายๆ ท่าน รวมถึงหัวหน้างานที่คอยให้คำปรึกษา คำแนะนำ ข้อเสนอแนะและชี้แนะข้อบกพร่องต่างๆ รวมทั้งเพื่อนร่วมงานที่คอยแบ่งปันความรู้ความเข้าใจในเทคโนโลยีที่ใช้ในโครงการ เพื่อให้โครงการสำเร็จ ลุล่วงได้ด้วยดี

ขอขอบคุณคุณพ่อ คุณแม่ ที่คอยให้คำปรึกษา กำลังใจ และความรักที่มีให้ตลอดเวลา



สราวุธ สิงห์กุล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VI
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของโครงการ.....	1
1.2 เป้าหมายในการพัฒนาระบบ.....	3
1.3 ขอบเขตในการพัฒนาระบบ.....	3
1.4 องค์ประกอบของระบบงาน.....	4
1.5 ขั้นตอนในการพัฒนาระบบ.....	4
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	5
บทที่ 2 การควบคุมปริมาณการจราจรของระบบเครือข่าย.....	6
2.1 องค์ประกอบของ Traffic Shaping.....	6
2.2 ระบบแถวคอย.....	7
2.3 การควบคุมปริมาณการจราจรของเครือข่ายด้วยพีริเอสดี.....	11
2.3.1 พีริเอสดี.....	11
2.3.2 การจัดการ PF.....	12
2.3.3 ALTQ.....	13
2.3.4 การเปิดใช้งาน ALTQ.....	13
2.3.5 การออกแบบใช้งาน ALTQ.....	14
2.3.6 การใช้ PF เพื่อปรับแต่ง ALTQ.....	17
บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	21
3.1 องค์ประกอบโดยรวมของระบบ.....	21
3.2 ลักษณะการทำงานของระบบ.....	22
3.3 Use Case Model.....	25
3.3.1 Use Case Diagram.....	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

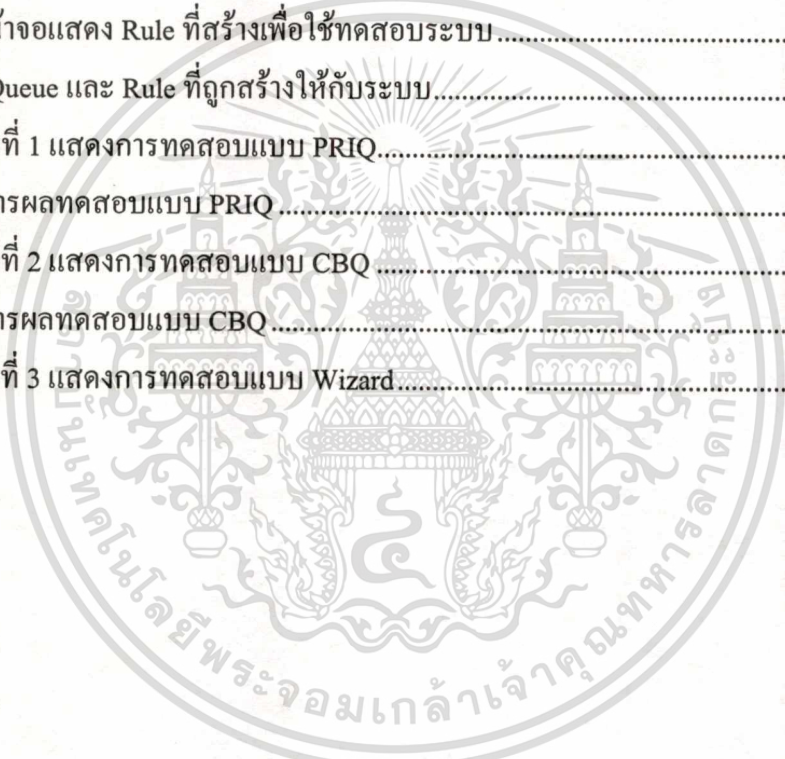
	หน้า
3.3.2 Use Case Description	26
3.3.3 Activity Diagram.....	28
3.3.4 Class Diagram	30
3.3.5 Sequence Diagram	31
บทที่ 4 การพัฒนาโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย.....	33
4.1 การวางแผนปฏิบัติงาน	33
4.2 การวางแผนโครงสร้างของระบบ.....	34
4.3 การเริ่มต้นการใช้งาน PF และ ALTQ.....	34
4.4 การใช้งาน PF และ ALTQ.....	36
4.5 การพัฒนาโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย	37
4.6 ผลการทดสอบโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย	43
บทที่ 5 บทสรุปและแนวทางในการพัฒนาในอนาคต.....	46
5.1 สรุปผลการพัฒนา.....	46
5.2 ข้อจำกัดของระบบ.....	46
5.3 สรุปแนวทางในการพัฒนาในอนาคต.....	47
บรรณานุกรม.....	48
ภาคผนวก ก	49
ภาคผนวก ข	72
ประวัติผู้เขียน	78

สารบัญรูป

รูปที่	หน้า
1.1 แสดงตัวอย่างบริการต่างๆ ที่ใช้งานผ่านอินเทอร์เน็ตของบริษัทอาบีโก้ จำกัด(มหาชน) ระหว่างเดือน มิถุนายน – สิงหาคม 2551.....	2
2.1 องค์ประกอบของ Traffic Shaping	6
2.2 FIFO.....	8
2.3 PQ	8
2.4 SFQ.....	9
2.5 CBQ.....	9
2.6 Service curve based QoS model	10
2.7 RED	10
2.8 RED packet drop probability as a function of the average queue size	11
2.9 การใช้งาน FreeBSD	12
2.10 Alternate Queuing Architecture.....	13
3.1 องค์ประกอบของ Traffic Shaper System	21
3.2 องค์ประกอบภายในของส่วน Server	22
3.3 แสดงการทำงานของ PF Framework	23
3.4 แสดงการใช้งาน Queue ร่วมกัน.....	23
3.5 แสดงการใช้งานแบบแยก Queue ทำงาน	23
3.6 Diagram แสดงลำดับการทำงานของระบบ	24
3.7 Diagram แสดงลำดับการตั้งค่าระบบ	24
3.8 แสดง Use Case Diagram ระบบการจัดการ Traffic Shaper	25
3.9 Activity Diagram การกำหนดค่าให้ระบบ	29
3.10 Activity Diagram การกำหนดค่าให้ระบบ แบบ Wizard Mode	30
3.11 Class Diagram ของระบบ Traffic Shaper.....	31
3.12 Sequence Diagram ของ Create new rule Use Case	32
3.13 Sequence Diagram ของ Modify rule Use Case	32
4.1 การวางแผน โครงสร้างของระบบ.....	34
4.2 แสดงการเก็บไฟล์ระบบที่ได้จากการกำหนดค่าผ่าน Web Base Interface	38

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3 แสดงหน้าจอการเรียกใช้งานโปรแกรมในส่วน Login.....	38
4.4 แสดงหน้าจอหน้าจอเมนูหลักของโปรแกรม	39
4.5 แสดงหน้าจอการกำหนดค่าพื้นฐานเบื้องต้นให้ Interface	39
4.6 แสดงหน้าจอการกำหนดค่าต่างๆ ของ Queue.....	40
4.7 แสดงหน้าจอแสดง Queue ที่สร้างเพื่อใช้ทดสอบระบบ	41
4.8 แสดงหน้าจอการกำหนดค่าต่างๆ ของ Rule.....	41
4.9 แสดงหน้าจอแสดง Rule ที่สร้างเพื่อใช้ทดสอบระบบ	42
4.10 แสดง Queue และ Rule ที่ถูกสร้างให้กับระบบ.....	42
4.11 ตัวอย่างที่ 1 แสดงการทดสอบแบบ PRIQ.....	43
4.12 แสดงการผลทดสอบแบบ PRIQ	43
4.13 ตัวอย่างที่ 2 แสดงการทดสอบแบบ CBQ	44
4.14 แสดงการผลทดสอบแบบ CBQ.....	44
4.15 ตัวอย่างที่ 3 แสดงการทดสอบแบบ Wizard.....	45



บทที่ 1

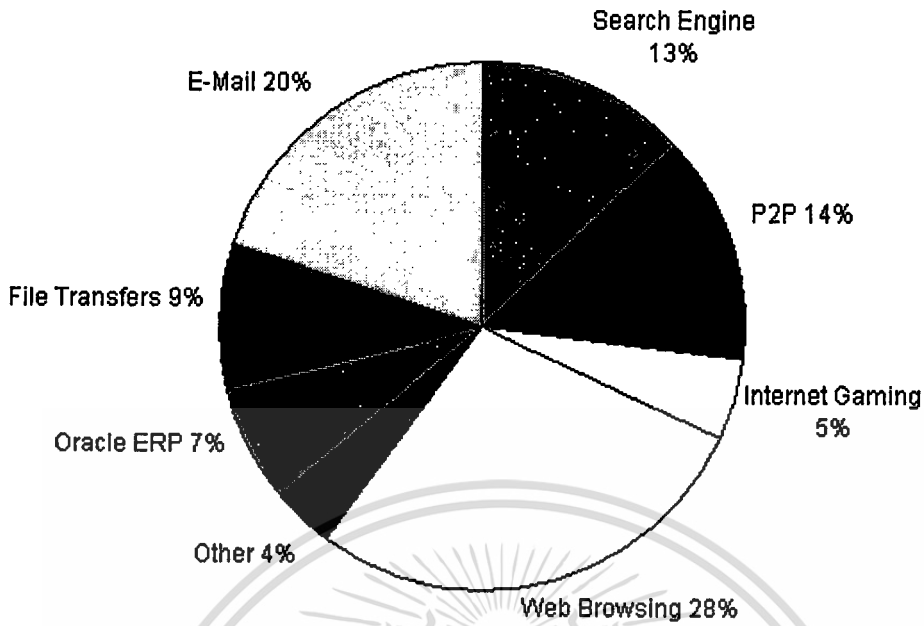
บทนำ

1.1 ความเป็นมาของโครงการ

ในปัจจุบันอินเทอร์เน็ตได้เข้ามามีบทบาทที่สำคัญต่อการดำเนินกิจกรรมต่างๆ ขององค์กรเป็นอย่างมาก โดยมีสาเหตุที่สำคัญคือ ความต้องการในการสื่อสารข้อมูลต่างๆ ขององค์กร ให้มีประสิทธิภาพสูงที่สุด ซึ่งอินเทอร์เน็ตถือเป็นเครื่องมือที่ช่วยการทำงานในการรับ-ส่งข้อมูลทางอิเล็กทรอนิกส์ ที่มีความรวดเร็ว และประหยัดค่าใช้จ่ายขององค์กรได้เป็นอย่างดี ในภาวะที่ค่าใช้จ่ายอื่นๆ มีอัตราการเติบโตที่สูงขึ้น จากราคาน้ำมันในยุคปัจจุบัน ทำให้เกิดการศึกษาค้นคว้าและวิจัยเทคโนโลยีใหม่ๆ เพื่อใช้ประโยชน์จากช่องทางอินเทอร์เน็ต เช่นการเปลี่ยนจากระบบรับ-ส่งเอกสารปกติเป็นระบบไร้เอกสาร หรือเปลี่ยนมาเป็นเอกสารทางอิเล็กทรอนิกส์แทน รวมทั้งการทำธุรกรรมต่างๆ ซึ่งต่อไปนี้ไม่จำเป็นต้องเดินทางไปติดต่อด้วยตนเองและสามารถดำเนินกิจกรรมต่างๆ ได้จากทั่วทุกมุมโลก

เมื่อการใช้งานอินเทอร์เน็ตของแต่ละองค์กรมีอัตราการใช้งานเพิ่มสูงขึ้นอย่างต่อเนื่อง สิ่งที่องค์กรต้องพิจารณาในลำดับถัดมาคือ ค่าบริการในการใช้งานอินเทอร์เน็ต ซึ่งก็ต้องมีการเพิ่มขึ้นตามอัตราการใช้งานของแต่ละองค์กร เพื่อให้เพียงพอต่อการใช้งานของทุกคนในองค์กร เพราะหากไม่มีการเพิ่มอัตราค่าบริการรับ-ส่งข้อมูลทางอินเทอร์เน็ตแล้ว องค์กรก็จะประสบปัญหาในการใช้งานอินเทอร์เน็ต คืออัตราการรับ-ส่งข้อมูลเต็ม มีผลทำให้การใช้งานระบบโดยรวมช้าลง แต่การเพิ่มอัตราค่าบริการรับ-ส่งข้อมูลให้สูงขึ้นก็เป็นเพียงการแก้ปัญหาที่ปลายเหตุ เพราะจากการศึกษาพฤติกรรมของผู้ใช้งานอินเทอร์เน็ตแล้ว จะพบว่าผู้ใช้แต่ละคนพยายามใช้งานให้มากที่สุด ตรวจจับที่อินเทอร์เน็ตยังสามารถให้บริการได้อย่างรวดเร็ว ซึ่งทำให้เกิดการแข่งขันกันเข้าใช้งาน และทำให้เกิดการเต็มในที่สุด และหากระบบหรือบริการนั้นมีความสำคัญสูง อาจทำให้เกิดความเสียหายต่อองค์กรได้ ดังนั้นเมื่อเราทำการศึกษาและวิเคราะห์การใช้งานอินเทอร์เน็ต ในรายละเอียดแล้วจะพบว่า ข้อมูลที่สื่อสารกันผ่านช่องทางอินเทอร์เน็ตนั้น ประกอบไปด้วยหลายบริการ เช่น เว็บ, อิเล็กทรอนิกส์เมล, โปรแกรมต่างๆ ที่ให้บริการทางธุรกิจ ดังรูปที่ 1.1 จะเห็นว่าบริการที่เกี่ยวข้องกับการทำงานหลักขององค์กรมีการใช้งานเพียง 14 % และอีก 53 % เป็นข้อมูลที่เกี่ยวข้องกับความบันเทิงและนันทนาการเป็นส่วนใหญ่ ทำให้ทุกองค์กรพยายามที่จะหาวิธีหรืออุปกรณ์ เพื่อมาช่วยในการจัดการหรือควบคุมการใช้งานช่องทางให้เกิดประสิทธิภาพสูงที่สุด และจัดลำดับความสำคัญของบริการหรือความสำคัญของผู้ใช้งาน ตามความต้องการขององค์กรได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดงตัวอย่างบริการต่างๆ ที่ใช้งานผ่านอินเทอร์เน็ตของบริษัทอาบีโก้ จำกัด(มหาชน) ระหว่างเดือน มิถุนายน – สิงหาคม 2551

จากปัญหาที่กล่าวมาทำให้การควบคุมปริมาณการจราจรของระบบเครือข่าย ในปัจจุบันจึงเป็นที่นิยมเป็นอย่างมาก แต่ละองค์กรพยายามที่จะหาวิธีจัดการที่ประหยัดที่สุด ดังนั้นในอันดับแรกจะมองไปที่การจัดการบนอุปกรณ์เราเตอร์ หรืออุปกรณ์ที่ทำหน้าที่เป็นเกตเวย์ของระบบอินเทอร์เน็ตเพื่อทำหน้าที่นี้ด้วย แต่จะพบปัญหา คือประสิทธิภาพการทำงานโดยรวมจะต่ำ เพราะตัวอุปกรณ์ต้องทำหน้าที่หลายอย่าง และการทำงานโดยรวมจะไม่ตรงกับความต้องการโดยตรงขององค์กร เนื่องจากหน้าที่การทำงานพื้นฐานเท่านั้น อันดับที่สอง องค์กรจะมองหาอุปกรณ์ที่ทำหน้าที่เฉพาะด้าน เช่น TrafficShaper ของบริษัท Packeteer แต่เป็นอุปกรณ์จากต่างประเทศที่มีราคาที่สูงมาก จึงไม่เหมาะกับองค์กรขนาดเล็กและกลางที่จะลงทุน ดังนั้นจึงได้มีการศึกษาหาวิธีต่างๆ เพื่อนำมาพัฒนาระบบ เช่นการประยุกต์ใช้งาน ฟรีบีเอสดี ซึ่งเป็นซอฟต์แวร์ฟรีที่ไม่เสียค่าลิขสิทธิ์ในการใช้งาน แต่จะมีส่วนที่ติดต่อกับผู้ใช้งานและคำสั่งต่างๆ ที่ใช้ในการปรับแต่งระบบ ยังมีการจัดการที่ยุ่งยากและซับซ้อน สำหรับผู้ที่ไม่ได้ศึกษาในรายละเอียดของการทำงาน ดังนั้นจึงได้มีแนวคิดที่จะพัฒนาให้ได้อุปกรณ์ที่มีประสิทธิภาพในการทำงานสูงและตรงกับความต้องการขององค์กรมากที่สุด ในราคาที่เหมาะสมกับองค์กรภายในประเทศ

1.2 เป้าหมายในการพัฒนาระบบ

เนื่องจากการใช้งานระบบควบคุมปริมาณการจราจรของระบบเครือข่าย ของฟรีบีเอสดี ที่มีอยู่เดิมเป็นการทำงานโดยการป้อนคำสั่งผ่านไฟล်ระบบ และเรียกใช้งานผ่านระบบ Command Line จึงทำให้เกิดความซับซ้อนและยุ่งยากต่อการเรียกใช้งาน ดังนั้นระบบที่จะมีการพัฒนาขึ้นมาใหม่จะเป็นระบบที่มีการรองรับการใช้งาน โดยที่ผู้ใช้ผู้ใช้สามารถจะเรียกใช้งานระบบผ่านเว็บ และทำการติดตั้งคำสั่งต่างๆ โดยการป้อนข้อมูลที่จำเป็นเท่านั้น และไม่จำเป็นต้องทราบรูปแบบของคำสั่ง

1.3 ขอบเขตในการพัฒนาระบบ

1. ติดตั้งระบบปฏิบัติการ FreeBSD เพื่อเป็นระบบปฏิบัติการของระบบควบคุมปริมาณการจราจรของระบบเครือข่าย
2. ติดตั้งระบบ Packet Filter บน FreeBSD เพื่อทำหน้าที่เป็นระบบควบคุมปริมาณการจราจรของระบบเครือข่าย
3. ติดตั้ง Apache Web Server เพื่อเป็นส่วนติดต่อกับผู้ใช้งานผ่านเว็บ
4. ติดตั้ง Open SSL เพื่อใช้ในการเข้ารหัสในส่วนติดต่อกับผู้ใช้งานผ่านเว็บ
5. ติดตั้ง Super User Do Package เพื่อการใช้งานคำสั่งที่ root สามารถสั่งงานได้ผ่านเว็บ
6. พัฒนาโปรแกรมระบบควบคุมปริมาณการจราจรของระบบเครือข่าย โดยที่โปรแกรมมีความสามารถในการทำงานดังนี้
 - สามารถป้อนข้อมูลที่จำเป็นสำหรับการสร้างกฎของระบบควบคุมปริมาณการจราจรของระบบเครือข่าย ซึ่งระบุได้ถึง IP Source, IP Destination
 - สามารถเพิ่มหรือลบกฎที่ทำการสร้างขึ้นได้
 - สามารถแสดงกฎที่สร้างขึ้นใหม่และกฎที่มีอยู่เดิมได้
 - สามารถแสดงผลการทำงานของระบบควบคุมปริมาณการจราจรของระบบเครือข่ายได้
 - สามารถแสดงผลของ Log การทำงานของกฎเพื่อใช้วิเคราะห์การทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 องค์ประกอบของระบบงาน

ระบบงานประกอบด้วยองค์ประกอบต่าง ๆ ดังต่อไปนี้

1.4.1 เครื่องคอมพิวเตอร์เซิร์ฟเวอร์ ระบบปฏิบัติการ FreeBSD ทำหน้าที่ตรวจสอบข้อมูลที่ผ่านเข้าออก เพื่อควบคุมปริมาณการจราจรของระบบเครือข่าย ซึ่งได้รับการเตรียมความพร้อมดังนี้

- ติดตั้งระบบปฏิบัติการ FreeBSD version 6.2 เพื่อรองรับการตรวจสอบข้อมูลที่ผ่านเข้าออกและควบคุมปริมาณการจราจรของระบบเครือข่าย
- ติดตั้งซอฟต์แวร์ควบคุมปริมาณการจราจรของระบบเครือข่ายโดยเลือกใช้ โปรแกรม Packet Filter
- ติดตั้งส่วนให้บริการเว็บเซิร์ฟเวอร์สำหรับเป็นส่วนติดต่อกับผู้ใช้เพื่อใช้ในการควบคุมระบบ โดยเลือกใช้ Apache เวอร์ชัน 2.2.8
- ติดตั้งซอฟต์แวร์ภาษา โดยเลือกใช้ PHP เวอร์ชัน 5.1.6
- ติดตั้ง Super User Do Package เวอร์ชัน 1.6.9
- ติดตั้งโปรแกรม Open SSL เวอร์ชัน 2.2

1.4.2 เครื่องคอมพิวเตอร์ไคลเอ็นท์ ที่ใช้ควบคุมปริมาณการจราจรของระบบเครือข่าย โดยผ่านทางเว็บ ซึ่งได้รับการเตรียมความพร้อมดังนี้

- ติดตั้งระบบปฏิบัติการ Windows XP Professional
- ติดตั้งโปรแกรมเว็บเบราว์เซอร์ สำหรับติดต่อกับเว็บเซิร์ฟเวอร์ เพื่อควบคุมโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย

1.5 ขั้นตอนในการพัฒนาระบบ

ประกอบไปด้วยขั้นตอนต่าง ๆ ดังนี้

1.5.1 ศึกษาความเป็นไปได้ ในการพัฒนาระบบควบคุมปริมาณการจราจรของระบบเครือข่าย ของ FreeBSD เพื่อกำหนดขอบเขตของปัญหาและวางแผนวิธีการพัฒนาโปรแกรม รวมถึงกำหนดเป้าหมายในการพัฒนาโครงการ โดยศึกษา ดังนี้

- ศึกษาวิธีการติดตั้งและการใช้งานซอฟต์แวร์ต่าง ๆ ที่ใช้สำหรับการสร้างระบบควบคุมปริมาณการจราจรของระบบเครือข่าย และระบบเว็บเซิร์ฟเวอร์ เพื่อรองรับการติดต่อจากภายนอก ได้แก่ FreeBSD, PF, Apache Web Server, PHP Extension, Super User Do, Shell Script Command

- ศึกษาเทคโนโลยีการควบคุมปริมาณการจราจรของระบบเครือข่ายโดยใช้ FreeBSD
- ศึกษาวิธีการสร้างกฎสำหรับควบคุมปริมาณการจราจรของระบบเครือข่ายให้มีประสิทธิภาพ
- ศึกษาการใช้งานโปรแกรมภาษา PHP เพื่อพัฒนาโปรแกรมประยุกต์ ในส่วนของการติดต่อกับผู้ใช้งาน

1.5.2 การวิเคราะห์และออกแบบ

ทำการวิเคราะห์และออกแบบระบบรวมถึงกำหนดความต้องการของโครงการพัฒนาระบบ โดยได้ทำการออกแบบให้ระบบสามารถเพิ่มหรือลบกฎที่ทำการสร้างและป้อนข้อมูลที่จำเป็น สำหรับการสร้างกฎของการควบคุมปริมาณการจราจรของระบบเครือข่าย ซึ่งระบุได้ถึง IP Source, Port Source, IP Destination, Port Destination, Protocol, Direction, Interface และแสดงผลการตรวจสอบการทำงานของไฟร์วอลล์

1.5.3 การพัฒนาและทดสอบ

- ทำการติดตั้งระบบควบคุมปริมาณการจราจรของระบบเครือข่ายและทดสอบกฎต่างๆ และตรวจสอบผลการทำงานของระบบ
- ทำการพัฒนาโปรแกรมและทดสอบการทำงานของโปรแกรมในฟังก์ชันต่างๆ

1.5.4 การทดลองใช้งานและปรับปรุงแก้ไข

นำโปรแกรมมาทดลองใช้งานและปรับปรุงแก้ไขเพื่อให้สามารถใช้งานได้ถูกต้องและง่ายยิ่งขึ้น

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้พัฒนาความรู้ความเข้าใจเรื่องการทำงานพื้นฐานของการควบคุมปริมาณการจราจรของระบบเครือข่าย
2. ได้พัฒนาความรู้ความเข้าใจเรื่องการทำงานของระบบแลกดอยด้วย PF
3. ได้พัฒนาความรู้ความสามารถในการวิเคราะห์ ออกแบบและพัฒนาระบบงานและสามารถนำไปใช้ประโยชน์ต่อการทำงานในอนาคตได้
4. ได้โปรแกรมประยุกต์ที่ผู้ดูแลระบบหรือผู้ใช้งานทั่วไป สามารถเรียกใช้งานและทำการแก้ไขการทำงานกฎต่าง ๆ ของระบบควบคุมปริมาณการจราจรของระบบเครือข่ายได้โดยสะดวก รวดเร็ว และง่ายต่อการใช้งานยิ่งขึ้นในลักษณะการทำงานแบบเว็บ

บทที่ 2

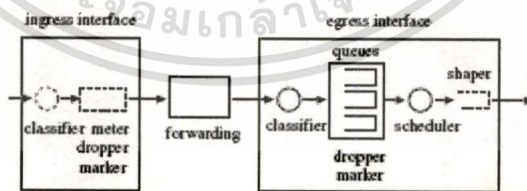
การควบคุมปริมาณการจราจรของระบบเครือข่าย

การควบคุมปริมาณการจราจรของระบบเครือข่ายหรือที่รู้จักกันในชื่อ Traffic shaping นั้นเป็นการควบคุมปริมาณทราฟฟิกที่จะถูกส่งเข้าไปในระบบเครือข่าย เพื่อให้เกิดการใช้งาน bandwidth ให้มีประสิทธิภาพและเหมาะสม โดยสามารถแบ่งออกเป็นหลายประเภท ตามการประยุกต์ใช้งานระบบแถวคอย (queue disciplines) แบบต่างๆ ซึ่งเป็นการจัดการทราฟฟิกทางด้านขาออก (Output) โดยจะได้อธิบายในหัวข้อถัดไป

สำหรับลักษณะการทำ Traffic shaping ที่เข้าใจได้ง่ายคือการจำกัดอัตราการส่งข้อมูล (Rate Limiting) ของทุกๆ ทราฟฟิก ให้มีอัตราเดียวกันหมดซึ่งจะเป็นรูปแบบที่ไม่มีควมยืดหยุ่น ดังนั้นลักษณะที่พบส่วนมากคือการแยกประเภทของทราฟฟิก (Traffic Classification) ตาม Port number หรือ Protocol number ทำให้สามารถจัดการ Rate limiting ของแต่ละบริการได้อย่างอิสระ เช่น กำหนดให้ Voice service ได้รับ Rate limiting มากกว่า File sharing service เป็นต้น

2.1 องค์ประกอบของ Traffic Shaping

องค์ประกอบต่างๆของการทำ Traffic shaping ภายในอุปกรณ์ Router จะเห็นว่าเมื่อ Packet เข้ามาที่ส่วนขาเข้า (ingress interface) ซึ่งทำหน้าที่ตรวจสอบ Packet หลังจากนั้นจะถูกส่งต่อ (Forwarding) ไปที่ส่วนขาออก (egress interface) ซึ่งเป็นส่วนหลักของการทำ queuing ในระบบ Traffic shaping ดังรูปที่ 2.1



รูปที่ 2.1 องค์ประกอบของ Traffic Shaping

สำหรับหน้าที่ของแต่ละส่วนมีดังต่อไปนี้

- Classifier ทำหน้าที่แยกประเภทของ packet ต่างๆที่ผ่านเข้ามาตาม packet header เช่น Address หรือ Port number โดย packet ที่ตรงกับ rule ที่กำหนดก็จะถูกนำไปประมวลผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

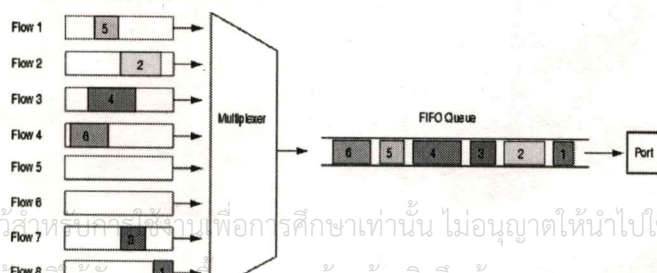
- Meter ทำหน้าที่วัดและเก็บจำนวน packet หรือค่าของ bandwidth (traffic stream) โดยค่าที่วัดได้จะถูกเก็บไว้ถูกนำไปใช้งานในส่วนอื่นๆ
- Markers ทำหน้าที่บันทึกค่าพิเศษลงไป ใน packet header เช่น priority, congestion, ชนิดของ application เป็นต้น
- Droppers ทำหน้าที่ drop packet ที่ทิ้งไปในกรณีที่ queue เกิดเต็มหรือมีความคับคั่ง (Congestion) ในเครือข่ายเกิดขึ้น
- Queues ทำหน้าที่เป็น buffer ในการเก็บ packet ซึ่งมีขนาดจำกัด โดย queue มีด้วยกันหลายรูปแบบ (queue discipline) ซึ่งจะกล่าวถึงในหัวข้อถัดไป
- Scheduler ทำหน้าที่เลือก packet ที่ต้องส่งจาก queue
- Shaper ทำหน้าที่หน่วงเวลา packet เพื่อเป็นการจำกัด peak rate สูงสุดที่จะส่งออกได้ โดยมีขนาด buffer ที่จำกัดและ packet ที่ถูก delay จะถูกทิ้งถ้า buffer เต็ม

2.2 ระบบแถวคอย

การทำ Queuing เป็นพื้นฐานที่สำคัญของการศึกษาการทำ Traffic shaping โดยในปัจจุบันมีด้วยกันหลากหลายแบบเป็นอย่างมาก โดยแต่ละแบบจะมีจุดเด่นที่แตกต่างกัน ขึ้นอยู่กับการนำไปประยุกต์ ใช้งานกับระบบที่กำลังศึกษาอยู่นั้น ว่ามีความเหมาะสมมากน้อยเพียงใด ดังนั้นในหัวข้อนี้จะนำเสนอ องค์ประกอบ และวิธีการทำ queuing แบบต่างๆ

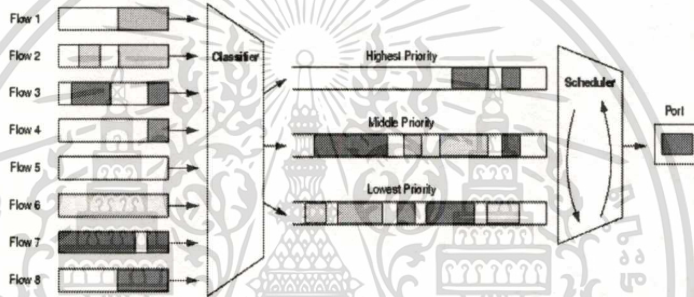
Queuing Disciplines คือรูปแบบของการจัดการ Queue โดยวิธีการต่างๆ กัน ซึ่งมีเป้าหมายสำคัญเพื่อจัดการ Traffic ที่จะส่งเข้าไปในระบบเครือข่าย เช่น bandwidth, delay, congestion avoidance โดยทั่วไปจะเป็นการทำงานร่วมกันของส่วนต่างๆ ที่อยู่ภายในฝั่งขาออก (egress interface) คือโครงสร้างของ queue, scheduler, dropper ซึ่งในปัจจุบัน Queue จะมีด้วยกันหลายแบบเพื่อให้เหมาะกับการนำไปใช้งานได้หลายรูปแบบ ดังต่อไปนี้

- FIFO (Fist-In-Fist-Out) เป็นการทำให้ queue discipline แบบที่เข้าใจง่ายที่สุด โดย packet ที่เข้าก่อนจะได้รับการบริการก่อน ตามลำดับ จากรูปที่ 2.2 เห็นว่า packet จะเรียงกันแบบแถวเดียวดังนั้นหาก buffer ภายใน queue นั้นเกิดเต็ม packet ที่เข้ามาถัดไปก็จะถูก drop ทิ้งไป ซึ่งวิธีนี้หาก Service ใดสามารถเข้าถึง queue ได้เร็วกว่าและทำการส่งข้อมูลแบบต่อเนื่อง จะทำให้ service อื่นๆ ไม่สามารถใช้งานในตอนนั้นได้



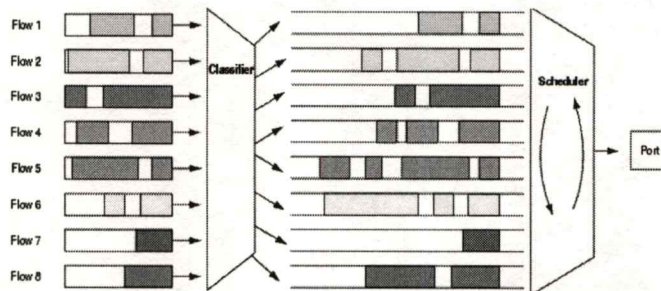
รูปที่ 2.2 FIFO

- PQ (Priority Queuing) เป็น queue discipline แบบมีการแบ่ง queue ออกเป็นหลาย queues ดังรูปที่ 3 โดยแต่ละ Queue จะมีการกำหนดระดับความสำคัญ (Priority level) ที่แตกต่างกัน แถวใดมีระดับความสำคัญที่สูงกว่า จะได้รับการบริการก่อนจนเสร็จ หลังจากนั้นแถวที่มีค่าความสำคัญต่ำกว่า จึงจะได้รับการบริการตามลำดับต่อไป แต่ถ้า queue ใดที่มี priority เท่ากันจะเวียนกันส่ง packet ใน queue สลับกันไปมาโดยใช้วิธีการที่เรียกว่า Round robin ในการส่ง packet ออกไป



รูปที่ 2.3 PQ

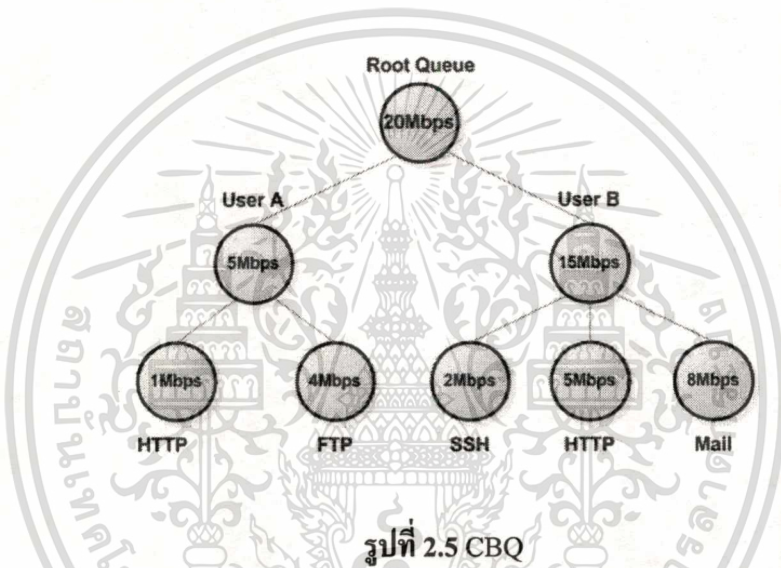
- SFQ (Stochastic Fairness Queuing) เป็น queue discipline ที่ทำการแบ่ง queue ออกเป็นหลายๆ queues ตามการแยกประเภทของ packet หรือ service ที่เข้ามาดังรูปที่ 2.4 และจะใช้วิธีการที่เรียกว่า Round robin เพื่อให้ packet ที่ส่งออกไป มีการกระจายกันอย่างยุติธรรม (Fair) ดังนั้น packet ประเภทใดที่ส่งมากเกินไปขนาดของ buffer ของแต่ละ queue ก็จะมี packet ประเภทนั้นทิ้งไปเท่านั้น ทำให้ packet ชนิดอื่นไม่ถูก drop ไปด้วย



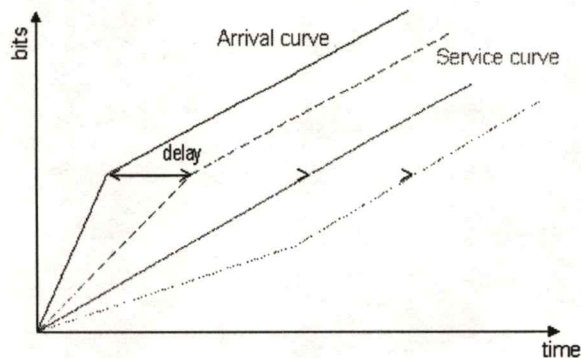
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 SFQ

- CBQ (Class Based Queuing) เป็น queue discipline ที่ทำการแบ่ง Queue ออกเป็นหลายๆ Queue class ตามปริมาณ bandwidth ที่แต่ละ queue ต้องการหรือสามารถจัดสร้างเป็นกลุ่ม (Class) สำหรับผู้ใช้หลายระดับ ในลักษณะที่เป็นลำดับชั้นได้ (Hierarchically) ดังรูปที่ 2.5 โดยสามารถสร้างกลุ่มลูก (Child class) ให้ผู้ใช้งานแต่ละราย เพื่อบริหารจัดการการจราจรภายในกลุ่มแม่ (Parent class) ย่อยลงไปอีกได้และสามารถนำ bandwidth ที่เหลือจากผู้ใช้งานหนึ่งไปให้กับผู้ใช้รายหนึ่งได้ เช่น Child class สามารถจะกำหนดให้ยืม (Borrow) bandwidth จาก parent class ได้ถ้าหากมี bandwidth เหลือ และยังมีคุณสมบัติพิเศษ (Option) เพิ่มเติมคือกำหนดค่าความสำคัญ (Priority) ได้อีกด้วย จึงเป็นวิธีการที่สามารถใช้ Bandwidth ได้อย่างคุ้มค่า อีกวิธีการหนึ่ง

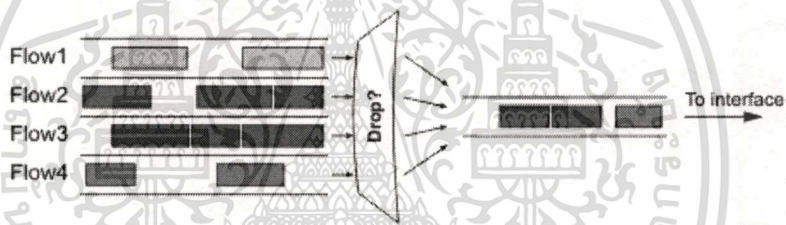


- HFSC (Hierarchical Fair Service Curve) เป็น queue discipline ที่มีการทำงานคล้ายกับ CBQ แต่จะต่างกันที่ HFSC จะใช้ Service curve based QoS model ร่วมด้วย ดังรูปที่ 2.6 เพื่อทำการจัดสรร Bandwidth (Link sharing) และรับประกันบริการประเภท Real time ไปด้วย (guaranteed real-time services) ซึ่งถือเป็นคุณสมบัติเฉพาะของ HFSC ในการแยก (Decouple) ระหว่าง Delay และ Bandwidth ออกจากกัน

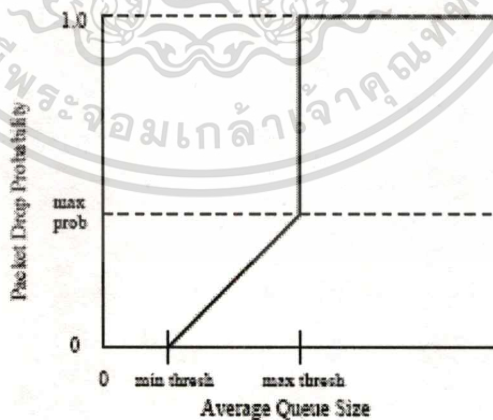


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับวงการศึกษาเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่วากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RED (Random Early Detection) เป็น queue discipline เพื่อใช้ทำกระบวนการหลีกเลี่ยงความคับคั่ง (Congestion avoidance algorithm) ของระบบเครือข่าย ซึ่งใช้จัดการกับ queue ต่างๆ ที่สร้างขึ้นมารับประกันว่าข้อมูลที่อยู่ใน queue ต่างๆ จะไม่เกิดการเต็ม ดังรูปที่ 7 โดย RED จะมีการคำนวณค่าเฉลี่ยขนาดของแถว (Average queue size) และนำไปเปรียบเทียบกับค่าที่ใช้ตรวจสอบต่ำสุด และสูงสุด (Minimum and maximum thresholds) ของแถว ถ้าค่า average queue size มีค่าต่ำกว่า minimum thresholds จะไม่มี packet ใดๆ ที่จะถูก drop แต่ถ้าค่าเฉลี่ยของแถวมีค่าสูงกว่าค่า maximum thresholds จะทำให้ packet ที่เข้ามาทั้งหมดจะถูก drop และถ้าค่าเฉลี่ยของแถวมีค่าระหว่างช่วง minimum และ maximum thresholds packet จะถูก drop โดยการคำนวณค่าความน่าจะเป็นจากค่า average queue size โดยส่วนมากค่าความน่าจะเป็นจะมีค่าใกล้เคียงกับค่า maximum thresholds ดังรูปที่ 8 ซึ่งการจัดการ Queue ของ TCP traffic เป็นตัวอย่างของการใช้ RED ในการทำ Congestion control



รูปที่ 2.7 RED



รูปที่ 2.8 RED packet drop probability as a function of the average queue size

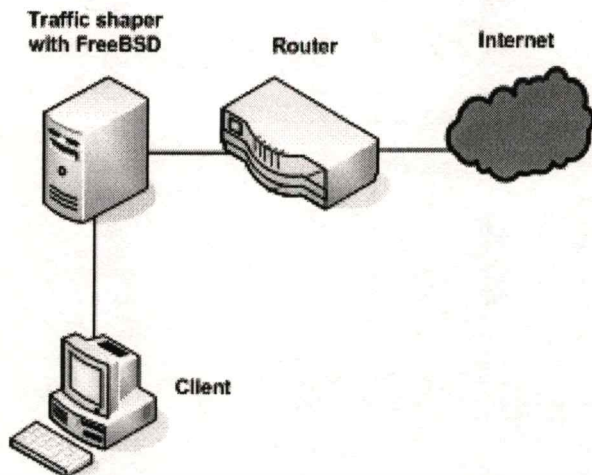
2.3 การควบคุมปริมาณการจราจรของเครือข่ายด้วยฟรีบีเอสดี

FreeBSD (Free Berkeley Software Distribution) เป็นระบบปฏิบัติการ Unix ชั้นสูงที่มีเสถียรภาพสูง และยังสามารถ Download มาใช้งานได้ฟรี โดยไม่มีค่าใช้จ่ายภายใต้ลิขสิทธิ์แบบ BSD (<http://www.freebsd.org>) และสามารถทำงานได้บนหลายๆ Platforms สำหรับการควบคุมปริมาณการจราจรของเครือข่ายบนระบบปฏิบัติการ FreeBSD ในปัจจุบันมีด้วยกันสองวิธีการคือ ALTQ (Alternate Queuing) และ DUMMYNET โดย ALTQ จะใช้เครื่องมือที่เรียกว่า PF (Packet Filter) ส่วน DUMMYNET จะใช้เครื่องมือที่เรียกว่า IPFW (IP Firewall) สำหรับในสัมมนา 1 นี้จะนำเสนอการจัดการ PF ร่วมกับ ALTQ disciplines แบบต่างๆ ในการควบคุมปริมาณการจราจรของเครือข่าย

2.3.1 ฟรีบีเอสดี

FreeBSD คือ ซอฟต์แวร์เสรีซึ่งเป็นระบบปฏิบัติการที่เหมือนยูนิกซ์ (Unix-like) สืบทอดมาจาก AT&T UNIX ผ่านทางสายของ Berkeley Software Distribution (BSD) คือ 386BSD และ 4.4BSD โดย FreeBSD รองรับการทำงานบน CPU ตระกูลหลักๆ หลายตระกูลด้วยกัน นอกจากตระกูล X86 ของ Intel ที่ใช้กันอย่างกว้างขวางแล้วยังมี DEC Alpha, UltraSPARC ของ Sun Microsystems, Itanium (IA-64), AMD64 และ PowerPC ส่วนของตระกูลรองได้แก่คอมพิวเตอร์สถาปัตยกรรมแบบ PC-98 การรองรับสำหรับตระกูล ARM และ MIPS กำลังอยู่ระหว่างการพัฒนา จุดเด่นที่สำคัญของ FreeBSD คือประสิทธิภาพและ เสถียรภาพ

ปัจจุบันระบบปฏิบัติการนี้ได้พัฒนาถึง เวอร์ชัน 7.0 ซึ่งถือเป็นเวอร์ชันล่าสุดที่เราสามารถดาวน์โหลดมาทดลองใช้ได้ฟรี ซึ่ง FreeBSD สามารถรองรับการทำงานหนัก ๆ (Heaviest loads) เพื่อตอบสนองการทำงานที่คึกคักงานต่าง ๆ ที่ User เป็นผู้สั่งการเข้ามาในเวลาเดียวกัน ปัจจุบันในประเทศไทยมีหลาย ๆ องค์กรนำระบบปฏิบัติการ FreeBSD นี้ไปใช้งานหลากหลายรูปแบบ อาทิ นำไปสร้างเป็นเครื่องให้บริการ (Server) ต่างๆ เช่น Web Server, Mail Server, FTP Server, Internet Gateway, Radius, IM (Instant Messaging), NAT, Firewall เป็นต้น สำหรับรายละเอียดต่างๆ สามารถศึกษาเพิ่มเติมได้ที่ <http://www.freebsd.org>



รูปที่ 2.9 การใช้งาน FreeBSD

2.3.2 การจัดการ PF

PF (Packet filter) เป็นเครื่องมือสำหรับการจัดการ Firewall ซึ่งรองรับกับระบบปฏิบัติการ FreeBSD ตั้งแต่ Version 5.3 เป็นต้นไป สามารถเลือกใช้งานเพื่อ การควบคุมปริมาณการจราจรของเครือข่ายได้ โดยผ่านทาง ALTQ disciplines แบบต่างๆ โดยปกติ PF จะถูกเปิดใช้งานอยู่แล้วแต่การใช้งาน ALTQ จะต้องทำการ Compile kernel ใหม่เพื่อเปิดใช้งาน

การเปิดใช้งานและกำหนดค่าให้กับ PF โดยการกำหนดค่าในไฟล์ /etc/rc.conf ดังนี้

```

gateway_enable="YES"
pf_enable="YES"
pf_rules="/etc/pf.conf"
pf_flags=" "
pflog_enable="YES"
pflog_logfile="/var/log/pflog"
pflog_flags=" "
  
```

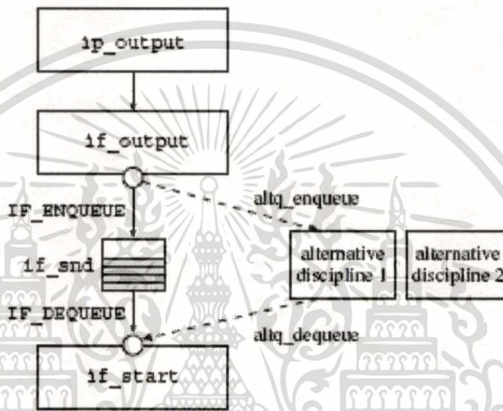
สำหรับคำสั่งที่สั่งให้ PF ทำงานคือ

```
# pfctl -e
```

โดยปกติ PF จะทำการอ่านกฎ (Rule) ต่างๆที่ถูกกำหนดจาก configuration ไฟล์ที่ชื่อว่า /etc/pf.conf ซึ่งการกำหนด rule ต่างๆของ ALTQ ก็จะถูกกำหนดในไฟล์นี้ด้วยเช่นกัน

2.3.3 ALTQ

ALTQ (Alternate Queuing) เป็น framework ที่มีในระบบปฏิบัติการ FreeBSD เป็นการประยุกต์ใช้งานการทำ Queuing disciplines ในรูปแบบต่างๆ ผ่านทาง PF โดยก่อนการใช้งาน ALTQ นั้นควรจะตรวจสอบระบบ Hardware ต่างๆ กับคู่มือ (Manual page) ที่ <http://www.freebsd.org> ก่อนเนื่องจากอุปกรณ์บางรุ่น เช่น Network card driver รุ่นที่ใช้งานนี้อาจจะไม่รองรับการทำงานได้ รูปที่ 10 จะเป็นการแสดงส่วนประกอบของ ALTQ โดยจะทำงานในส่วนขาออกของระบบ โดยปกติหากไม่มีการตั้งค่า ALTQ ให้กับระบบจะเป็นแบบ FIFO โดยอัตโนมัติ



รูปที่ 2.10 Alternate Queuing Architecture

2.3.4 การเปิดใช้งาน ALTQ

การปรับแต่ง Kernel เพื่อ Activate ALTQ ทำได้โดยการแก้ไขไฟล์ GENERIC สำหรับใช้ปรับแต่ง Kernel ของ FreeBSD ซึ่งอยู่ใน `/usr/src/sys/i386/conf`

ในตัวอย่างนี้จะคัดลอกไฟล์ GENERIC มาเป็นไฟล์ที่ชื่อ MYSHAPER สำหรับใช้ปรับแต่ง Kernel โดยใช้คำสั่งต่อไปนี้

```
# cd /usr/src/sys/i386/conf
# cp GENERIC MYSHAPER
```

ทำการแก้ไขไฟล์ MYSHAPER และเพิ่มคำสั่งต่างๆ เข้าไปด้านล่างของไฟล์ เพื่อรองรับการทำ ALTQ disciplines แบบต่างๆ ดังนี้

```
device pf
```

เอกสารนี้เป็นเอกสารที่ device pflog ับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

device pfsync
options ALTQ
options ALTQ_CBQ
options ALTQ_RED
options ALTQ_RIO
options ALTQ_HFSC
options ALTQ_PRIQ
options ALTQ_NOPCC

```

เมื่อแก้ไขเสร็จแล้ว ให้ทำการ Compile kernel ใหม่โดยใช้คำสั่งดังนี้

```

# /usr/sbin/config MYSHAPER
# cd ../compile/MYSHAPER
# make cleandepend
# make depend
# make
# make install

```

2.3.5 การออกแบบใช้งาน ALTQ

เมื่อเราเข้าใจถึงวิธีการทำ Queuing disciplines แบบต่างๆแล้ว ต่อไปจะเสนอตัวอย่างในการออกแบบด้วยวิธี PRIQ และ CBQ ซึ่งวิธี CBQ นี้เป็นแบบที่มีความยืดหยุ่นสูงและได้รับความนิยมเป็นอย่างมาก ดังนี้

- **ALTQ_PRIQ**

โครงสร้างของ PRIQ เป็นแบบแบนราบ (flat) คือ โดยการกำหนด Root queue ที่รวม Bandwidth ทั้งหมดและกำหนด Queue ย่อย (Sub queue) ภายใต้อัน Root queue แต่ไม่สามารถกำหนดให้มี Queue ย่อย อยู่ภายใน Queue นั้น ได้อีก เหมือนกับ CBQ ดังนั้นจึงทำให้มี Queue เพียงแค่ระดับเดียวเท่านั้น

ตัวอย่างที่ 2.3.5.1

Root Queue (2Mbps)

Queue A (priority 1)

Queue B (priority 2)

Queue C (priority 3)

เอกสารนี้เป็นเอกสารที่... เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ PRIQ เป็นแบบแบนราบ (flat) คือ โดยการกำหนด Root queue ที่รวม Bandwidth ทั้งหมดและกำหนด Queue ย่อย (Sub queue) ภายใต้ Root queue แต่ไม่สามารถกำหนดให้มี Queue ย่อย อยู่ภายใน Queue นั้น ได้อีก เหมือนกับ CBQ ดังนั้นจึงทำให้มี Queue เพียงแค่ระดับเดียวเท่านั้น

ตัวอย่างที่ 2.3.5.1

Root Queue (2Mbps)

Queue A (priority 1)

Queue B (priority 2)

Queue C (priority 3)

ในตัวอย่างนี้มีการกำหนด Root queue มี Bandwidth รวมเท่ากับ 2 Mbps โดยมี 3 queue ย่อยโดย Queue C จะได้รับการบริการก่อนจนหมดทุก packet ที่อยู่ใน Queue หรือ Queue ว่าง จึงจะมาให้บริการ Queue ถัดไปคือ Queue B และ Queue A ตามลำดับ สำหรับการให้บริการภายในแต่ละ Queue จะเป็นการจัดการแบบ FIFO

ดังนั้นในการใช้งาน PRIQ จะต้องมีการวางแผนก่อนให้ดี เพราะ PRIQ จะให้บริการ priority ที่สูงกว่าก่อนเสมอ ถ้าเกิด priority ที่สูงกว่านั้นมีการส่ง packet แบบต่อเนื่องไปตลอด (Constant stream) Queue ที่มี priority ที่ต่ำกว่า packet ภายใน queue นั้นก็จะถูก Delay หรือ Drop ไปตลอด จนกว่า queue ที่มี priority สูงกว่าจะส่ง packet จนหมด

- **ALTO_CBQ [7]**

CBQ จะจัดการ queue แบบลำดับชั้น โดยส่วนบนสุดจะถูกเรียกว่า Root queue ซึ่งจะเป็นเป็นตัวที่กำหนดขนาดรวมของ bandwidth ที่มีที่จะใช้ในการแบ่ง ส่วน Child queue จะถูกสร้างภายใต้ root queue และจะถูกกำหนด bandwidth ที่แบ่งมาจาก root queue

ตัวอย่างที่ 2.3.5.2

Root Queue (2Mbps)

Queue A (1Mbps)

Queue B (500Kbps)

Queue C (500Kbps)

ในตัวอย่างนี้ Root Queue มี Bandwidth รวมเท่ากับ 2 megabits per second (Mbps) โดยถูกแบ่งให้กับ 3 Child queues คือ A, B และ C ซึ่งแต่ละ queue ได้รับ bandwidth ไม่เท่ากัน

ตัวอย่างที่ 2.3.5.3

ในการจัดการแบบลำดับชั้นสามารถที่จะขยาย Queue ภายใน Child queue ได้อีก โดย bandwidth รวมของ child queue จะต้องไม่เกินขนาดของ parent queue ดังตัวอย่างด้านล่าง

Root Queue (2Mbps)

User A (1Mbps)

SSH (50Kbps)

Bulk (950Kbps)

User B (1Mbps)

Audio (250Kbps)

Bulk (750Kbps)

HTTP (100Kbps)

Other (650Kbps)

ตัวอย่างที่ 2.3.5.4

ในแต่ละ Queue ยังสามารถกำหนดให้มีการยืม Bandwidth จาก parent queue ได้ถ้ายังว่างและไม่ถูกใช้จาก child queue อื่นๆ โดยสามารถกำหนดได้ดังนี้

Root Queue (2Mbps)

User A (1Mbps)

SSH (100Kbps)

FTP (900Kbps, borrow)

User B (1Mbps)

การทำงานถ้า Traffic ใน ftp queue ต้องการให้ bandwidth มากกว่า 900 Kbps และ traffic ใน User A มี bandwidth เหลือ เพราะ SSH queue ใช้ bandwidth น้อยกว่าที่ได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น FTP queue จะยืม bandwidth จาก User A แต่ถ้า SSH queue ต้องการใช้ bandwidth เพิ่มขึ้นก็จะได้รับคืนโดยอัตโนมัติ

ตัวอย่างที่ 2.3.5.5

CBO ยังสามารถที่จะกำหนดระดับความสำคัญ (Priority level) ให้แต่ละ Queue ได้ด้วย โดยใน parent queue เดียวกัน Queue ที่มี priority ที่สูงกว่าจะได้รับสิทธิก่อน queue ที่มี priority ต่ำกว่าในกรณีที่เครือข่ายมีความหนาแน่น (network congested) แต่ถ้า Queue ใดที่มี priority เท่ากันจะได้รับบริการแบบ Round robin ดังตัวอย่างด้านล่าง

Root Queue (2Mbps)

User A (1Mbps, priority 1)

SSH (100Kbps, priority 5)

FTP (900Kbps, borrow, priority 3)

User B (1Mbps, priority 1)

การทำงาน CBQ จะให้บริการ User A และ User B แบบ Round robin ส่วน queue อื่นๆ เช่นในกรณีที่ User A เริ่มทำงาน CBQ จะให้สิทธิกับ SSH queue ก่อน FTP queue ในช่วงที่เครือข่ายเกิดความหนาแน่น แต่ SSH queue และ FTP queue จะไม่เปรียบเทียบ priority กับ User A และ User B เนื่องจากไม่ได้อยู่ในระดับชั้นเดียวกันกับตัวมัน

ตัวอย่างที่ 2.3.5.6

แสดงการออกแบบ CBQ ร่วมกับการทำ RED เพื่อให้มีการทำ Congestion control ดังตัวอย่างด้านล่าง

Root Queue (2Mbps)

User A (1Mbps)

SSH (40%, RED)

FTP (60%, borrow, RED)

User B (1Mbps)

2.3.6. การใช้ PF เพื่อปรับแต่ง ALTQ

เนื่องจาก PF เป็นเครื่องมือที่ใช้ปรับแต่ง ALTQ ดังนั้นในหัวข้อนี้จะกล่าวถึงคำสั่งเพื่อใช้ PF ปรับแต่ง queues แบบต่างๆ ที่เราได้ออกแบบมา ซึ่งจะผ่านไฟล์ของ Firewall rule คือ `/etc/pf.conf` โดยจะแบ่งออกเป็น 2 ส่วนหลัก คือส่วนที่ 1 เป็นคำสั่งในการจัดการ queues ซึ่งจะประกอบด้วยคำสั่ง `altq on` และคำสั่ง `queue` กับส่วนที่ 2 ใช้กำหนด traffic ให้กับ queues ที่สร้างขึ้นมา ดังนี้

1. คำสั่ง ALTQ ON

เป็นคำสั่งในการเปิดการทำงาน (Enable) ALTQ ร่วมกับ Interface และใช้สำหรับสร้าง root queue ขึ้นมาด้วย โดยมีรูปแบบคำสั่งดังนี้

```
altq on interface scheduler bandwidth bw qlimit qlim tbrsize size queue { queue_list }
```

- **interface** คือ Network interface ที่จะใช้ทำ queuing
- **scheduler** คือ queuing discipline ที่ ALTQ รองรับ ซึ่ง interface 1 ตัวรองรับได้เพียง 1 queuing discipline เท่านั้น
- **bw** คือ จำนวน bandwidth ทั้งหมดที่สามารถใช้ใน discipline นี้ได้ โดยมีหน่วยเป็น b (bit/sec), Kb, Mb, Gb หรือเป็นเปอร์เซ็นต์ของ Interface bandwidth ก็ได้
- **qlim (option)** คือจำนวนของ packet สูงสุดที่จะรับได้ใน queue โดยปกติจะมีค่าเท่ากับ 50
- **size (option)** คือจำนวนของ token bucket regulator มีหน่วยเป็น byte โดยปกติจะมีค่าเท่ากับ Interface bandwidth
- **queue_list** คือ child queues ต่างๆที่อยู่ใน root queue นี้

ตัวอย่างที่ 2.3.6.1

```
altq on fxp0 cbq bandwidth 2Mb queue {std, ssh}
```

จากตัวอย่างเป็นการใช้งาน CBQ บน Interface fx0 โดยมี bandwidth ทั้งหมดเท่ากับ 2 Mbps และมี 2 child queues คือ std และ ssh

2. คำสั่ง Queue

จะเป็นการกำหนดคุณสมบัติของ child queue โดยมีรูปแบบคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
queue name [on interface] bandwidth bw [priority pri] [qlimit qlim] scheduler \
(sched_options) { queue_list }
```

- **name** คือชื่อของ queue ซึ่งจะต้องตรงกับ queue list ที่ altq on
- **interface** (option) คือ interface ที่ queue นี้ใช้งานโดยปกติจะกำหนดให้กับทุก interface
- **bw** คือ จำนวน bandwidth ทั้งหมดที่สามารถใช้ใน discipline นี้ได้โดยมีหน่วยเป็น b (bit/sec), Kb, Mb, Gb หรือเป็นเปอร์เซ็นต์ของ Parent queue bandwidth ก็ได้ โดยรองรับกับ queuing discipline แบบ CBQ และ HFSC เท่านั้น ถ้าไม่ได้กำหนดจะมีค่าเท่ากับ 100% ของ Parent queue
- **pri** คือค่า priority ของ queue สำหรับ CBQ จะมีค่าระหว่าง 0 - 7 และ PRIQ จะมีค่าระหว่าง 0 - 15 โดย 0 จะมี priority ต่ำสุด หากไม่ได้กำหนดจะมีค่าเท่ากับ 1
- **qlim** (option) คือจำนวนของ packet สูงสุดที่จะรับได้ใน queue โดยปกติจะมีค่าเท่ากับ 50
- **scheduler** คือ queuing discipline ที่ ALTQ รองรับ และต้องตรงกับ root queue
- **sched_options** (option) คือการกำหนดคุณสมบัติพิเศษให้กับ scheduler เพิ่มเติม ดังนี้
 - **default** เป็นการกำหนด queue ให้กับ packet ต่างๆ ที่ไม่ตรงกับ Rule ใดๆ ที่กำหนดให้ผ่าน queue นี้
 - **red** เป็นการเปิดใช้งานการทำ RED
 - **rio** เป็นการเปิดใช้งานการทำ RED with input/output โดย RED จะสามารถใช้ค่า average queue lengths และ threshold หลายค่าสำหรับแต่ละบริการของ QoS
 - **ecn** เป็นการเปิดให้บริการ Explicit Congestion Notification เป็นการแสดงถึงความคับคั่งของระบบเครือข่าย
 - **borrow** เป็นการกำหนดให้ queue สามารถยืม bandwidth จาก parent queue ได้โดยรองรับกับ queuing discipline แบบ CBQ และ HFSC เท่านั้น
- **queue_list** คือ child queues ต่างๆที่ถูกสร้างอยู่ภายใน queue นี้โดยรองรับกับ queuing discipline แบบ CBQ และ HFSC เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.3.6.2

```
queue std bandwidth 80% cbq(default)
queue ssh bandwidth 20% { ssh_login, ssh_bulk }
queue ssh_login bandwidth 25% priority 4 cbq(ecn)
queue ssh_bulk bandwidth 75% cbq(borrow red)
```

จากตัวอย่างเป็นการสร้าง Child queue ของ Root queue ตัวอย่างที่แล้ว โดย std queue จะได้รับ bandwidth 80 % ของ root queue (หรือ 1600 Kbps) และถูกตั้งเป็น default queue ด้วย ส่วน ssh queue จะได้รับ bandwidth 20 % ของ root queue (หรือ 400 Kbps) และมี 2 child queue ภายในคือ ssh_login และ ssh_bulk โดย ssh_login มี priority ที่สูงกว่า ssh_bulk และใช้ ECN ส่วน ssh_bulk สามารถยืม bandwidth จาก ssh queue ได้และมีการทำ RED ด้วย

3. การกำหนด Traffic ให้กับ Queue

การกำหนด Traffic ให้กับ queue จะใช้งานร่วมกับคำสั่งในส่วน filter rule ของ Firewall เช่น

```
pass out on fxp0 from any to any port 22
```

ซึ่งจะเป็นการกำหนดกฎให้ทุก Traffic สามารถผ่าน fx0 ทางขาออกได้ ถ้าใช้ port 22 และจากคำสั่งด้านล่างเป็นกำหนด Traffic ให้กับ queue ssh ดังนี้

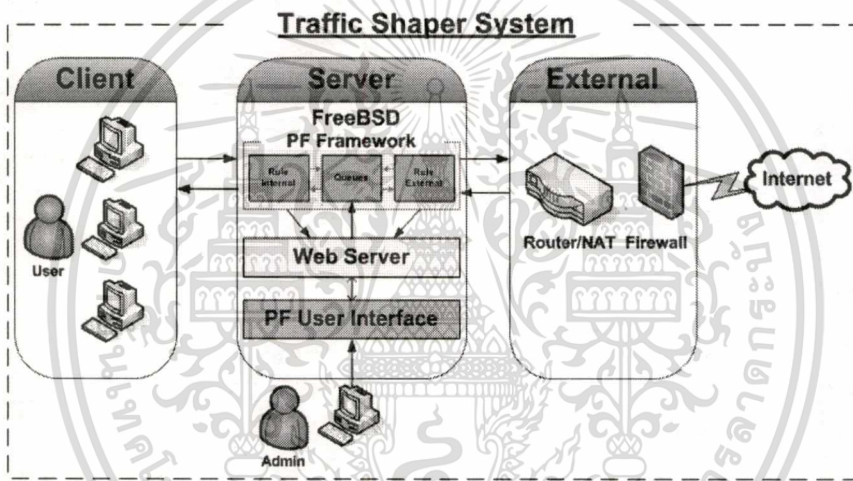
```
pass out on fxp0 from any to any port 22 queue ssh
```

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 องค์ประกอบโดยรวมของระบบ

ในการนำระบบไปใช้งานนั้น โดยส่วนมากเราจะนำไปใช้งานกับ ระบบเครือข่ายที่มีการใช้งาน Internet อยู่แล้วเป็นพื้นฐาน ซึ่งโดยปกติจะใช้การทำ NAT (Network Address Translation) และ Firewall อยู่แล้ว ดังนั้นเพื่อให้สะดวกในการใช้งาน เราจึงออกแบบให้ สามารถต่อพ่วงกับระบบเดิมเพื่อสร้างส่วนใหม่ ที่มีการทำ Traffic Shaping ขึ้นมา ดังรูปที่ 3.1



รูปที่ 3.1 องค์ประกอบของ Traffic Shaper System

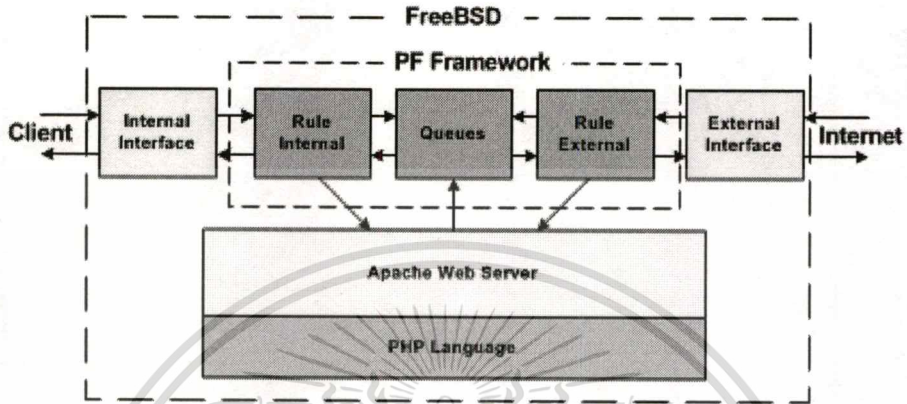
ซึ่งประกอบด้วยส่วนประกอบต่างๆดังต่อไปนี้

1. เครื่องคอมพิวเตอร์ Client ติดตั้งระบบปฏิบัติการ Windows XP และโปรแกรมเว็บเบราว์เซอร์ สำหรับติดต่อกับเว็บเซิร์ฟเวอร์ เพื่อควบคุมการทำงานของระบบ
2. เครื่องคอมพิวเตอร์ Server ติดตั้งระบบปฏิบัติการ FreeBSD และติดตั้งซอฟต์แวร์ต่างๆ ดังนี้
 - PF Framework ทำหน้าที่ ตรวจสอบข้อมูลที่วิ่งผ่านเข้าออกและการกระทำต่อข้อมูลนั้น, การกำหนด Queue รูปแบบต่างๆ เพื่อจัดการ Bandwidth
 - Apache Web Server ทำหน้าที่ ให้บริการเว็บ สำหรับเป็นส่วนติดต่อกับผู้ใช้งาน เพื่อใช้ในการควบคุมระบบ
 - PHP Language ทำหน้าที่ประมวลผลคำสั่งจากผู้ใช้งานผ่านเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. External เป็นอุปกรณ์ Router ที่เชื่อมต่อกับระบบภายนอกคือ Internet ซึ่งอาจจะมีการใช้งานในส่วนของการทำ Firewall และ NAT (Network Address Translation) ด้วยก็ได้

จากส่วนประกอบทั้ง 3 ส่วน เราจะพิจารณาการทำงานของส่วน Server ซึ่งจะเป็นส่วนที่เราจะทำการพัฒนาเพื่อนำมาใช้งาน ดังรูปที่ 3.2



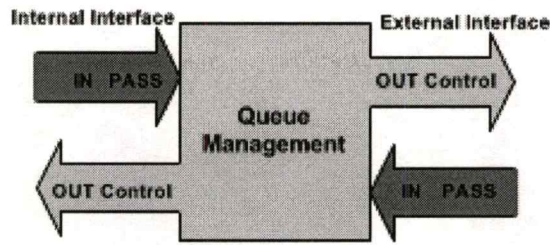
รูปที่ 3.2 องค์ประกอบภายในของส่วน Server

3.2 ลักษณะการทำงานของระบบ

จากองค์ประกอบของระบบที่ผ่านมา เราจะทำการวิเคราะห์ลักษณะการทำงานของส่วน PF Framework ซึ่งถือเป็นส่วนที่สำคัญที่สุดของการทำงาน เพราะเป็นส่วนตรวจสอบข้อมูลที่วิ่งผ่านเข้าออกและการกระทำต่อข้อมูลนั้นรวมทั้ง การกำหนด Queue รูปแบบต่างๆ โดยเราจะแบ่งออกเป็น 3 ส่วนย่อยคือ

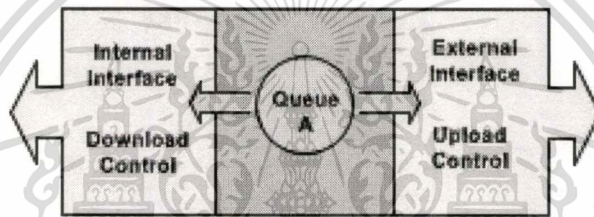
1. **Internal Interface** ส่วนนี้จะเป็นส่วนของระบบเครือข่ายภายใน จะมีหน้าที่รับข้อมูลจากภายในเข้าสู่ Queue เพื่อส่งออกภายนอกทาง External Interface
2. **External Interface** ส่วนนี้จะเป็นส่วนของระบบเครือข่ายภายนอก (Internet) จะมีหน้าที่รับข้อมูลจากภายนอกเข้าสู่ Queue เพื่อส่งออกภายในทางผ่าน Internal Interface
3. **Queue Management** ส่วนนี้จะทำหน้าที่จัดการและประมวลผล Queue ตามที่ได้กำหนดรูปแบบ และกฎต่างๆ ไว้ เพื่อส่งออก (Out) ให้กับทั้ง 2 ส่วน ในรูปที่ 3.3 แสดงให้เห็นการทำงาน จะเห็นว่าการจัดการ Queue จะเกิดที่ส่วนขาออก ของ Internal และ External Interface เท่านั้น ส่วนขาเข้า จะทำหน้าที่เพียงส่ง Packet เข้ามาไว้ใน Queue เท่านั้นเอง ดังนั้นการ Download จะถูกควบคุมโดยขาออก ของ Internal Interface และการ Upload จะถูกควบคุมโดยฝั่งขาออกของ External Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

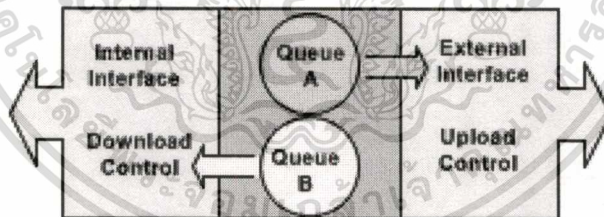


รูปที่ 3.3 แสดงการทำงานของ PF Framework

ในการออกแบบ Queue นั้นเราสามารถออกแบบให้ Queue ที่เราออกแบบนั้นสามารถนำไปใช้งานกับทั้ง 2 Interface ก็ได้ ดังรูปที่ 3.4 หรือหากเราต้องการออกแบบให้แต่ละ Interface ใช้งาน Queue คนละแบบกันก็ได้ เพื่อความสะดวก ดังรูปที่ 3.5

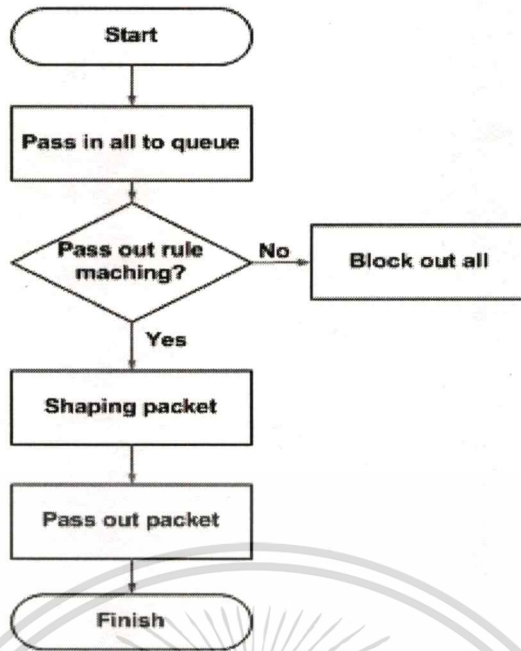


รูปที่ 3.4 แสดงการใช้งาน Queue ร่วมกัน



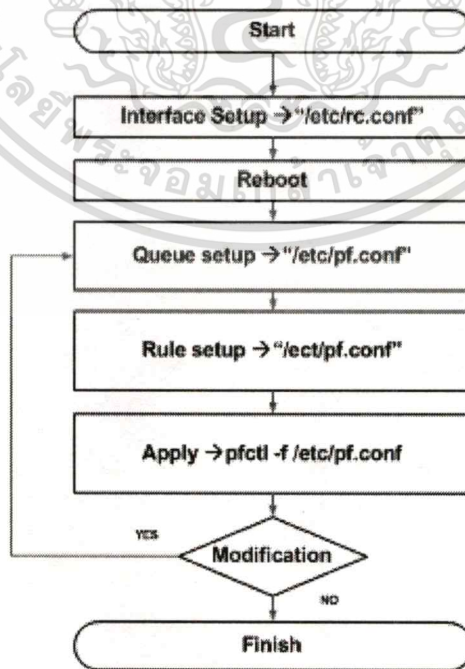
รูปที่ 3.5 แสดงการใช้งานแบบแยก Queue ทำงาน

สรุปการทำงาน ดังรูปที่ 3.6 ด้านล่าง จะเป็นการแสดงให้เห็นการทำงาน เมื่อมีข้อมูลเข้ามาที่ระบบทั้ง Internal หรือ External ข้อมูลจะผ่านเข้ามาโดย Default และเก็บอยู่ใน Queue เพื่อรอการส่งออก ซึ่งส่วนส่งออกนี้จะทำการตรวจสอบกับ Rule ที่เรากำหนดให้ (Administrator) ว่าลักษณะข้อมูลตรงกันหรือไม่ หากไม่ตรงก็จะ Block ข้อมูลนั้นทิ้งไป แต่หากตรงก็จะความีการเรียกใช้ Queue แบบใดเพื่อจัดการข้อมูลในการส่งออก



รูปที่ 3.6 Diagram แสดงลำดับการทำงานของระบบ

ส่วนรูปที่ 3.7 จะเป็นลำดับการตั้งค่าให้กับระบบ โดยในลำดับแรกจะต้องตั้งค่า Network ให้กับ Interface ก่อนทั้ง Internal และ External ซึ่งต้องทำการ Reboot ระบบด้วย หลังจากนั้นจึงทำการจัดการ Queue และตรวจสอบข้อมูล (Packet) ผ่านไฟล์ “/etc/pf.conf” และ Apply โดยคำสั่ง pfctl -f



รูปที่ 3.7 Diagram แสดงลำดับการตั้งค่าระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

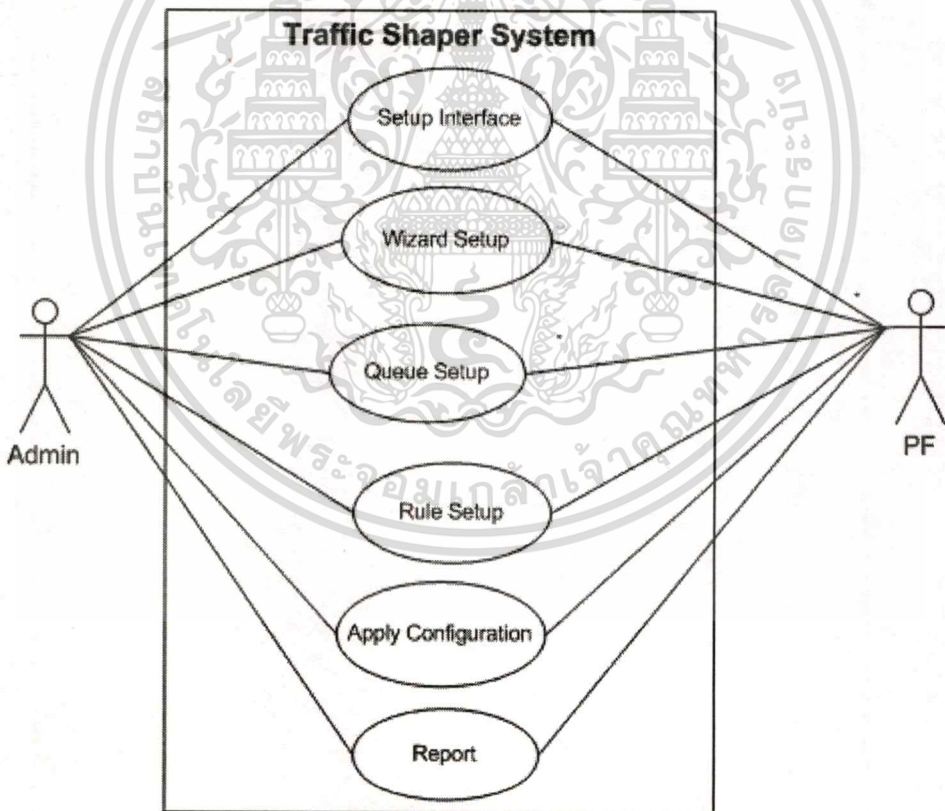
3.3 Use Case Model

Use Case Model เป็นการออกแบบระบบโดยใช้ UML Model เพื่อให้เห็นภาพรวมของระบบ และเป็นเอกสารอ้างอิง ในการพัฒนาระบบขึ้นมา โดย Model แรกที่เราจะนำเสนอในที่นี้คือ Use Case Diagram และ Use Case Description

Use Case Diagram

เป็นแผนภาพที่ใช้ที่แสดงปฏิสัมพันธ์ระหว่าง ระบบงานและสิ่งที่อยู่นอกระบบงาน Use Case Diagram ของระบบ Traffic Shaper System ประกอบด้วย

- Actor ในระบบนี้จะมี 2 Actor คือ Admin ซึ่งจะเป็นผู้จัดการกับระบบทั้งหมด และ PF
- Use Case ประกอบด้วย 6 Use Case ดังนี้คือ Setup Interface, Wizard Setup, Queue Setup, Rule Setup, Apply Configuration และ Report



รูปที่ 3.8 แสดง Use Case Diagram ระบบการจัดการ Traffic Shaper

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use Case Description

จะเป็นการแสดงรายละเอียดของแต่ละ Use Case ที่ได้นำเสนอไปแล้ว เพื่อให้เข้าใจชัดเจนยิ่งขึ้น ดังตัวอย่างด้านล่าง

- **Use Case: Setup Interface**

Brief Description: Admin logon เข้าสู่ระบบและต้องการกำหนดค่า Network Configuration ของ Internal/External Interface ให้กับระบบ

Actor: Admin

Precondition: ป้อนรหัสผ่านถูกต้อง

Basic Flows:

1. Admin เรียกใช้ Web page และป้อนรหัสผ่านเพื่อ Log on.
2. เลือก Tab Interface Setup และกำหนดค่าของระบบ Network ให้กับระบบ
3. บันทึกและ Restart ระบบอีกครั้ง เพื่อเสร็จสิ้นกระบวนการ

Alternative Flow:

1a. รหัสไม่ถูกต้อง: ระบบให้ใส่รหัสใหม่ พร้อมแจ้งข้อความผิดพลาด

Post condition: ระบบเครือข่ายจะเปลี่ยนแปลงตามที่กำหนด

- **Use Case: Wizard Setup**

Brief Description: Admin logon เข้าสู่ระบบและต้องการกำหนดค่าของระบบแบบ Wizard

Actor: Admin

Precondition: ป้อนรหัสผ่านถูกต้อง

Basic Flows:

1. Admin เรียกใช้ Web page และป้อนรหัสผ่านเพื่อ Log on.
2. เลือก Tab Wizard Setup และกำหนดค่าให้กับระบบตามลำดับขั้นตอน
3. Restart ระบบอีกครั้ง เพื่อเสร็จสิ้นกระบวนการ

Alternative Flow:

1a. รหัสไม่ถูกต้อง: ระบบให้ใส่รหัสใหม่ พร้อมแจ้งข้อความผิดพลาด

Post condition: ระบบจะเปลี่ยนแปลงตามที่กำหนด

- **Use Case: Queue Setup**

Brief Description: Admin logon เข้าสู่ระบบและต้องการสร้าง Queue ขึ้นใหม่ให้กับระบบ

Actor: Admin

Precondition: ป้อนรหัสผ่านถูกต้อง, ต้องกำหนดค่า Internal/External Interface ให้เรียบร้อย

Basic Flows:

1. Admin เรียกใช้ Web page และป้อนรหัสผ่านเพื่อ Log on.
2. เลือก Tab Queue Management และ Create Queue
3. เลือก Interface และชนิดของ Queue ที่จะใช้งานรวมทั้งกำหนดรายละเอียดต่างๆของ Queue
4. บันทึกเพื่อเสร็จสิ้นกระบวนการ

Alternative Flow:

- 1a. รหัสไม่ถูกต้อง: ระบบให้ใส่รหัสใหม่ พร้อมแจ้งข้อความผิดพลาด
- 2a. สร้าง Queue มากกว่าจำนวน Interface: ระบบจะแจ้งข้อความผิดพลาด

Post condition: Queue จะถูกสร้างและเชื่อมโยงกับ Interface นั้น

- **Use Case: Rule Setup**

Brief Description: Admin logon เข้าสู่ระบบและต้องการสร้าง Rule ขึ้นใหม่ให้กับระบบ

Actor: Admin

Precondition: ป้อนรหัสผ่านถูกต้อง, ต้องทำการสร้าง Queue ให้เรียบร้อย

Basic Flows:

1. Admin เรียกใช้ Web page และป้อนรหัสผ่านเพื่อ Log on.
2. เลือก Tab Rule Management และ Create Rule
3. เลือก Interface และชนิดของบริการ หรือเงื่อนไขในการทำงานของ Rule นี้
4. บันทึกเพื่อเสร็จสิ้นกระบวนการ

Alternative Flow:

- 1a. รหัสไม่ถูกต้อง: ระบบให้ใส่รหัสใหม่ พร้อมแจ้งข้อความผิดพลาด

- **Use Case: Report**

Brief Description: Admin logon เข้าสู่ระบบและต้องการตรวจสอบสถานการณ์ทำงานของระบบ

Actor: Admin

Precondition: ป้อนรหัสผ่านถูกต้อง

Basic Flows:

1. Admin เรียกใช้ Web page และป้อนรหัสผ่านเพื่อ Log on.
2. เลือก Tab System Monitor
3. เลือก ออก เมื่อเสร็จสิ้นกระบวนการ

Alternative Flow:

- 1a. รหัสไม่ถูกต้อง: ระบบให้ใส่รหัสใหม่ พร้อมแจ้งข้อความผิดพลาด

- **Use Case: Apply Configuration**

Brief Description: Admin logon เข้าสู่ระบบและต้องการตรวจสอบสถานการณ์ทำงานของระบบ

Actor: Admin

Precondition: ป้อนรหัสผ่านถูกต้อง

Basic Flows:

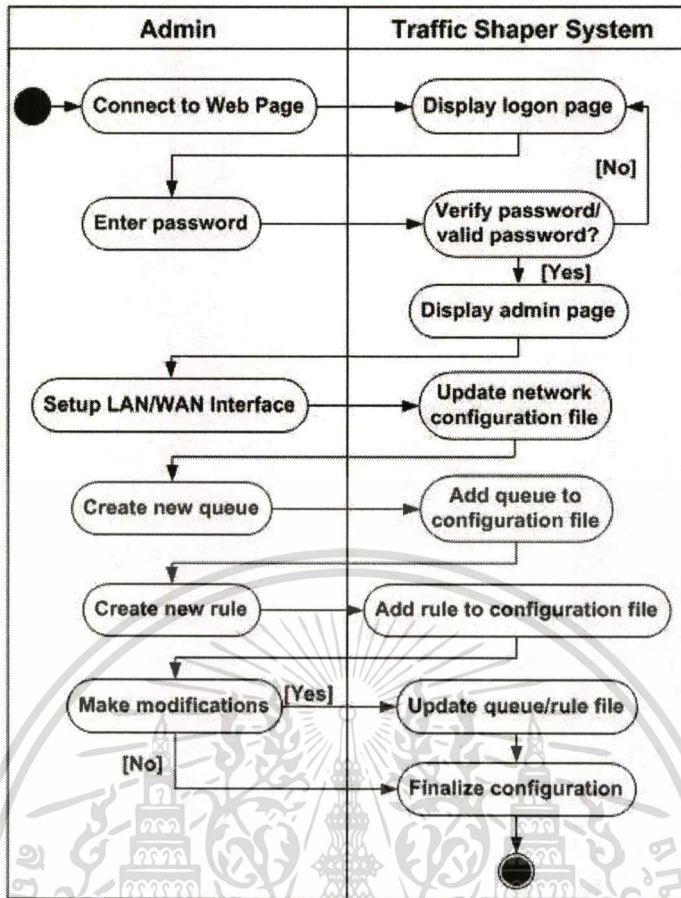
1. Admin เรียกใช้ Web page และป้อนรหัสผ่านเพื่อ Log on.
2. เลือก Tab Apply
3. เลือก ออก เมื่อเสร็จสิ้นกระบวนการ

Alternative Flow:

- 1a. รหัสไม่ถูกต้อง: ระบบให้ใส่รหัสใหม่ พร้อมแจ้งข้อความผิดพลาด

3.3.3 Activity Diagram

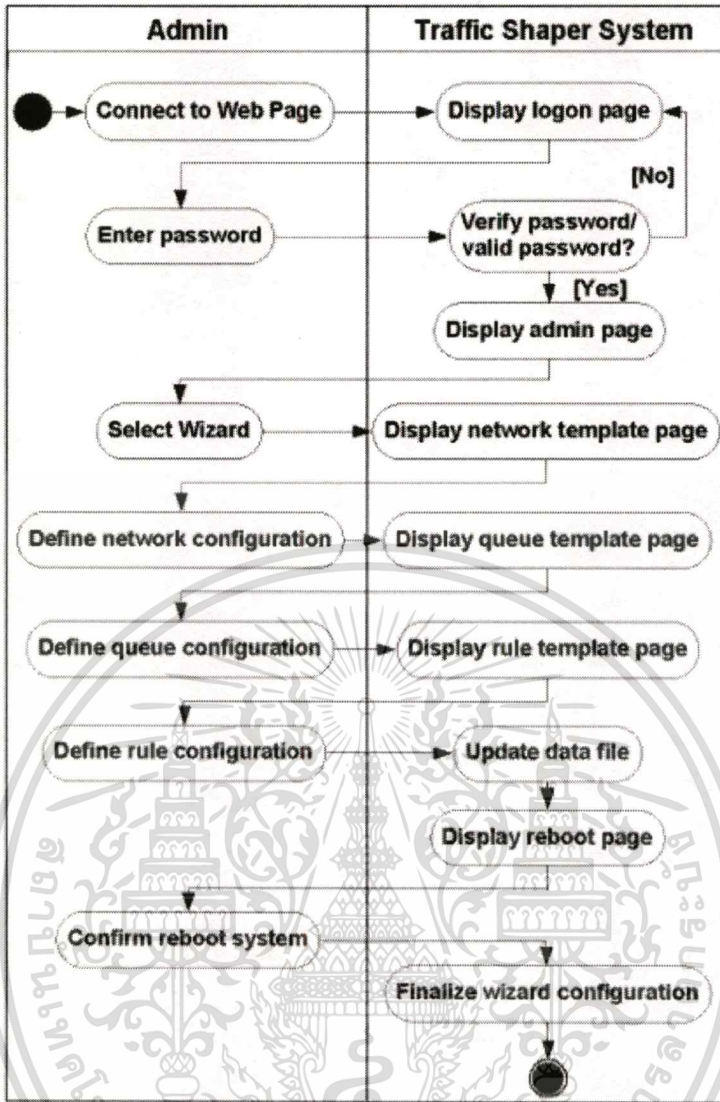
Activity Diagram เป็น Diagram ที่ใช้ในการแสดงลำดับการทำงาน (Workflow) ของ Admin ในการกำหนดการทำงานให้กับระบบ Traffic Shaper System โดยจะแบ่งการทำงานออกเป็น 2 ฟิ่ง (Swim lanes) คือฟิ่งของ Admin และฟิ่งของระบบ ดังรูปที่ 3.9



รูปที่ 3.9 Activity Diagram การกำหนดค่าให้ระบบ

จากรูปที่ 3.9 จะเป็นการแสดงเหตุการณ์ ในการที่ Administrator ได้ทำการกำหนดค่าให้กับระบบ โดยเริ่มต้นจะเป็นการเรียก Web page จากระบบ และป้อน Password เพื่อให้ระบบทำการตรวจสอบ เมื่อตรวจสอบแล้ว จะทำการเลือกส่วน Setup LAN/WAN Interface เพื่อตั้งค่า แล้วจึงไปสร้าง Queue และ Rule ต่อไป หากไม่ต้องการแก้ไข ก็จะเสร็จสิ้นกระบวนการกำหนดค่าให้ระบบ

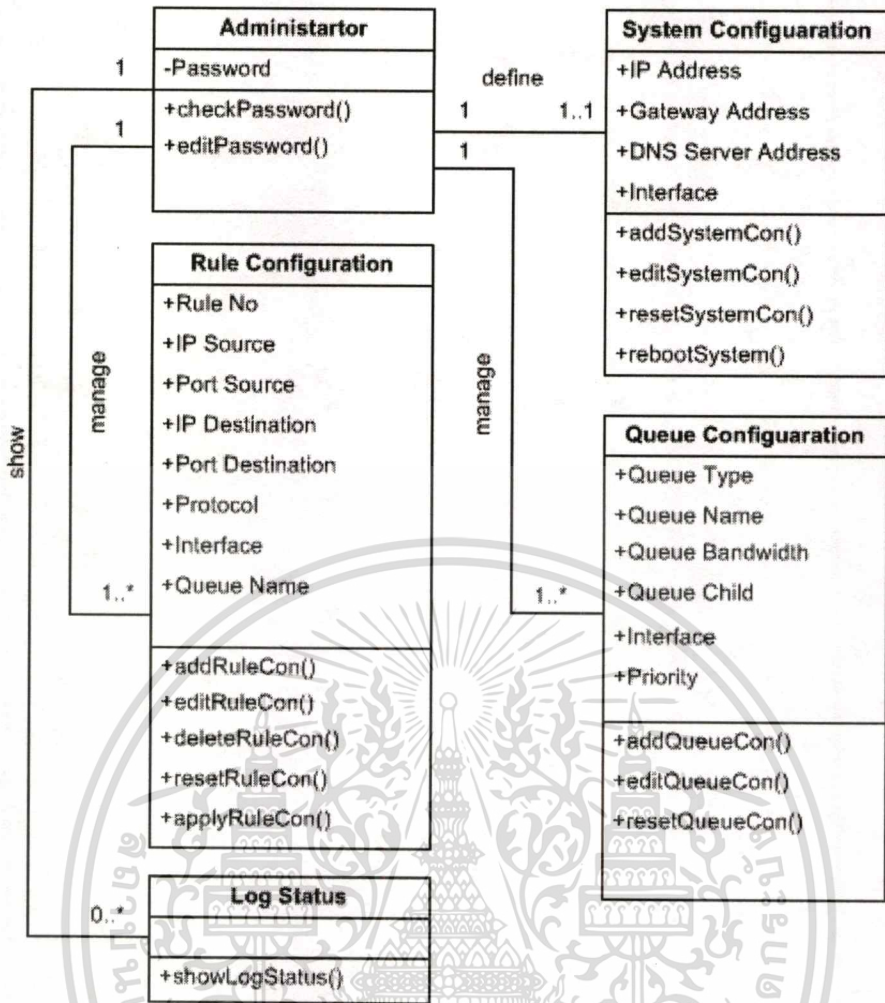
รูปที่ 3.10 ด้านล่างจะเป็นการกำหนดค่าให้ระบบในแบบ Wizard Mode ซึ่งจะทำให้เกิดความสะดวกในการใช้งานมากยิ่งขึ้น เพราะไม่ต้องเข้าไปในแต่ละเมนู เพื่อตั้งค่าให้แต่ละส่วน โดยระบบจะทำการแสดง Template page แต่ละขั้นตอนให้อัตโนมัติ เพื่อให้การตั้งค่าต่อเนื่องจนเสร็จสิ้นกระบวนการ ทำให้การใช้งานที่สะดวกขึ้น



รูปที่ 3.10 Activity Diagram การกำหนดค่าให้ระบบ แบบ Wizard Mode

3.3.4 Class Diagram

เป็นแผนภาพที่ใช้แสดง Class และความสัมพันธ์ระหว่าง Class ของระบบที่สนใจ (Problem Domain) ดังรูปที่ 3.11 แสดง Class Diagram ของระบบ Traffic Shaper ซึ่งประกอบไปด้วย 4 Class ดังต่อไปนี้ Administrator, System Configuration, Rule Configuration, Queue Configuration และ Log Status

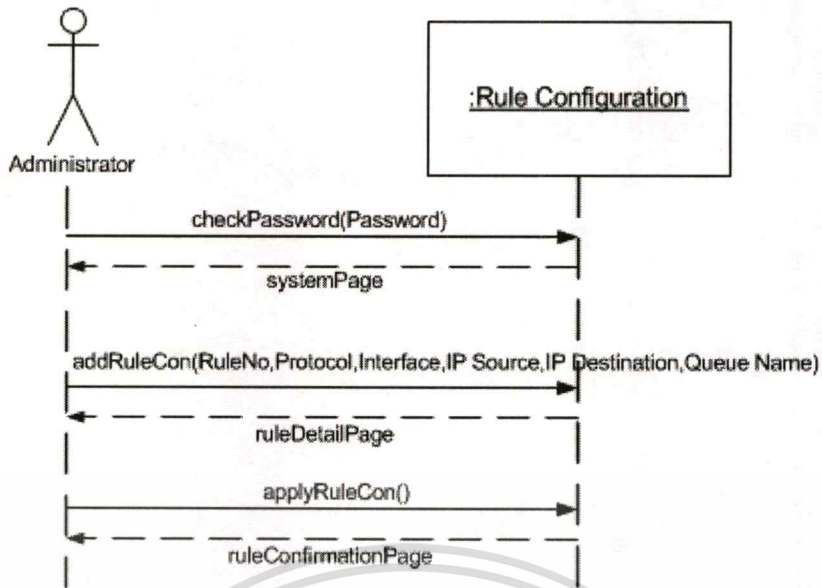


รูปที่ 3.11 Class Diagram ของระบบ Traffic Shaper

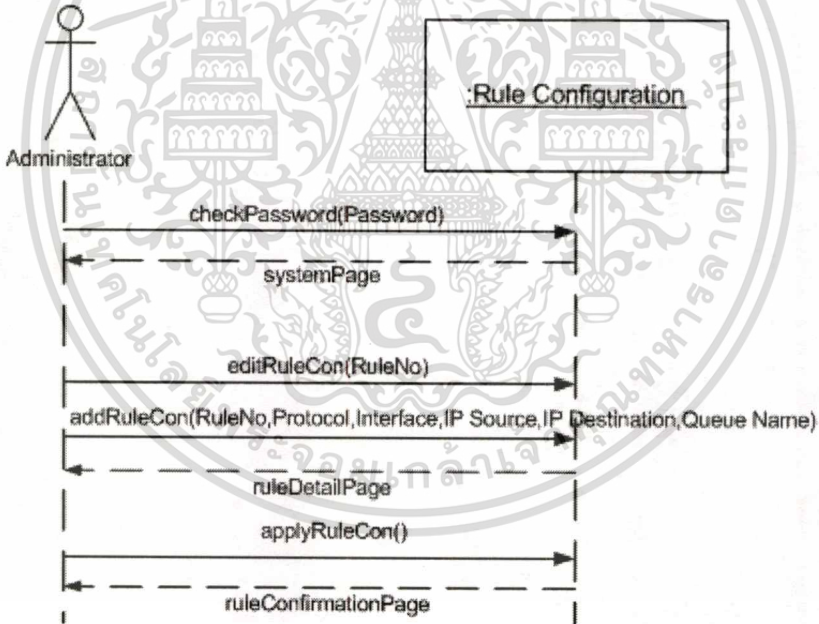
3.3.5 Sequence Diagram

เป็นแผนภาพที่ใช้อธิบายการทำงานของ Use Case เพื่อแสดงถึงขั้นตอนการทำงานและลำดับของการสื่อสาร (Message) ระหว่าง Object ที่ตอบโต้กัน

Sequence Diagram จะแสดงอยู่ในรูปแบบ 2 มิติ โดยเส้นประแนวตั้ง (Lifeline) จะนำเสนอในด้านเวลา ส่วนเส้นแนวนอน (Message) จะนำเสนอเกี่ยวกับการโต้ตอบกันระหว่าง Object หรือ Class ต่างๆ



รูปที่ 3.12 Sequence Diagram ของ Create new rule Use Case



รูปที่ 3.13 Sequence Diagram ของ Modify rule Use Case

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การพัฒนาโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย

ในการพัฒนาโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย ได้มีการกำหนดแผนการพัฒนาดังนี้

4.1 การวางแผนปฏิบัติงาน

ได้เลือกใช้ระบบปฏิบัติการและซอฟต์แวร์ดังต่อไปนี้ (สามารถอ่านรายละเอียดลำดับขั้นตอนในการติดตั้งและแหล่งที่มาของโปรแกรม ได้จากภาคผนวก (ก))

1. ระบบปฏิบัติการเลือกใช้ FreeBSD 6.2

เป็นระบบปฏิบัติการยูนิกซ์ ที่ได้รับความนิยมอย่างแพร่หลาย และไม่ต้องเสียค่าใช้จ่าย โดยระบบปฏิบัติการ FreeBSD นั้นเป็นระบบปฏิบัติการที่มีความปลอดภัยสูงกว่าระบบปฏิบัติการแบบ Open Source ทั่วไป

2. ซอฟต์แวร์ในการควบคุมปริมาณการจราจรของระบบเครือข่ายเลือกใช้ PF

เป็นซอฟต์แวร์ที่สามารถทำงานกับระบบปฏิบัติการ FreeBSD โดยที่ PF จะทำงานอยู่ใน เคอร์เนลของระบบปฏิบัติการ ซึ่งมีส่วนที่ติดต่อกับผู้ใช้งานแบบ Command Line โดยผู้ใช้งานจะต้องทำการแก้ไขและสั่งงานด้วยคำสั่ง

3. เว็บเซิร์ฟเวอร์ที่รองรับการทำงานของ PHP และ SSL เลือกใช้ Apache เวอร์ชัน 2.2.8

Apache เป็นซอฟต์แวร์สำหรับให้บริการเว็บเซิร์ฟเวอร์ เพื่อติดต่อกับผู้ใช้งานผ่านทางโปรโตคอล HTTP ในกรณีใช้งานในแบบมีการเข้ารหัสข้อมูล จะใช้งานผ่านโปรโตคอล HTTPS

4. ซอฟต์แวร์ภาษา เลือกใช้ PHP เวอร์ชัน 5.1.6

PHP เป็น scripting language ที่ทำงานร่วมกับ HTML โดยสามารถเขียนแทรกเข้าไปใน HTML โดย PHP เป็น Open Source ที่สามารถใช้ได้กับหลายๆระบบปฏิบัติการ ไม่ว่าจะเป็น Windows, UNIX, Linux

5. ติดตั้ง Super User Do Package เวอร์ชัน 1.6.9

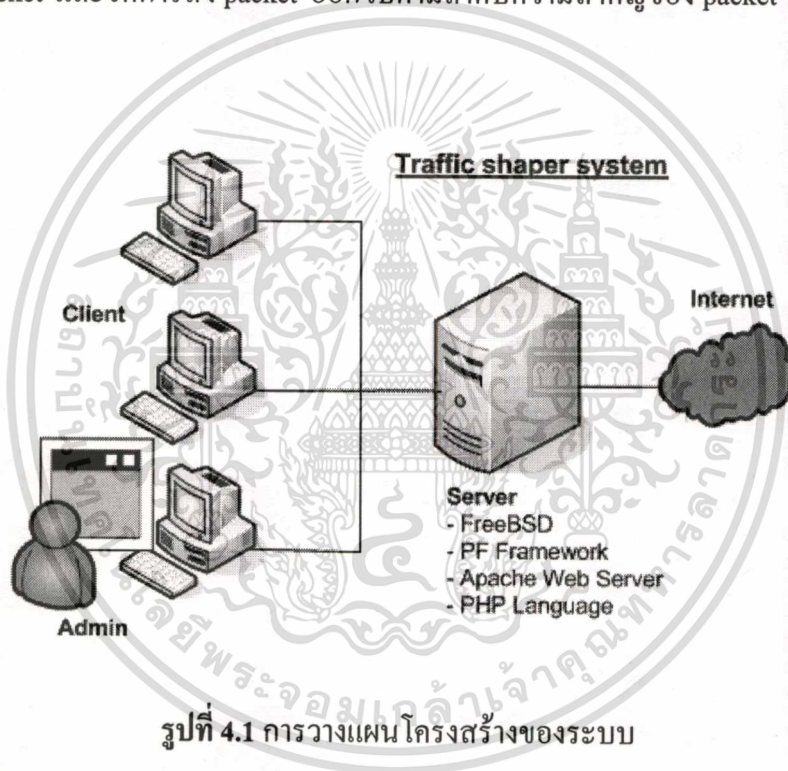
Super User Do Package เป็นซอฟต์แวร์ที่อนุญาตให้ผู้ใช้งานจากเว็บ สามารถใช้คำสั่งระบบต่างๆได้ผ่านทางเว็บ

6. ติดตั้งโปรแกรม Open SSL เวอร์ชัน 2.2

Open SSL เป็นซอฟต์แวร์ที่ใช้ในการเข้ารหัสข้อมูล ของการสื่อสารผ่านทางเว็บ เพื่อให้เกิดความปลอดภัยในการใช้งาน ใช้งานผ่าน โพรโทคอล HTTPS

4.2 การวางแผนโครงสร้างของระบบ

โครงสร้างของระบบที่จะนำเอาระบบควบคุมปริมาณการจราจรของระบบเครือข่ายมาใช้งาน ได้มีการออกแบบลักษณะการใช้งาน โดย จะทำการติดตั้งระบบควบคุมปริมาณการจราจรของระบบเครือข่ายบนเครื่องที่ทำหน้าที่เป็น Gateway ของระบบเครือข่ายโดยที่เครื่องนี้จะทำหน้าที่ตรวจสอบ packet และจัดการส่ง packet ออกไปตามลำดับความสำคัญของ packet ที่ได้กำหนดไว้ ดังรูปที่ 4.1



รูปที่ 4.1 การวางแผน โครงสร้างของระบบ

4.3 การเริ่มต้นการใช้งาน PF และ ALTQ

ก่อนที่จะเริ่มใช้งานซอฟต์แวร์ PF และ ALTQ ต้องทำการเปิดการใช้งานซึ่งอยู่ในเคอร์เนลของ FreeBSD ก่อน โดยปกติ PF จะถูกเปิดใช้งานอยู่แล้วแต่การใช้งาน ALTQ จะต้องทำการ Compile kernel ใหม่เพื่อเปิดใช้งาน ดังนี้

1. กำหนดค่าให้กับ PF โดยการกำหนดค่าในไฟล์ /etc/rc.conf ดังนี้

```
gateway_enable="YES"
```

```
pf_enable="YES"
```

```
pf_rules="/etc/pf.conf"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pf_flags=" "
pflog_enable="YES"
pflog_logfile="/var/log/pflog"
pflog_flags=" "
```

- เปิดการทำงานของ PF โดยใช้คำสั่งดังนี้

```
# pfctl -e
```

- คัดลอกไฟล์ Kernel เพื่อใช้ในการปรับแต่งเพื่อเปิดการทำงานของ ALTQ ดังนี้

```
# cd /usr/src/sys/i386/conf
# cp GENERIC MYSHAPER
```

- ทำการแก้ไขไฟล์ MYSHAPER และเพิ่มคำสั่งต่างๆ เข้าไปด้านล่างของไฟล์ เพื่อรองรับการทำ ALTQ disciplines แบบต่างๆ ดังนี้

```
device pf
device pflog
device pfsync
options ALTQ
options ALTQ_CBQ
options ALTQ_RED
options ALTQ_RIO
options ALTQ_HFSC
options ALTQ_PRIQ
options ALTQ_NOPCC
```

- เมื่อแก้ไขเสร็จแล้ว ให้ทำการ Compile kernel ใหม่โดยใช้คำสั่งดังนี้

```
# /usr/sbin/config MYSHAPER
# cd ../compile/MYSHAPER
# make cleandepend
# make depend
# make
# make install
```

- สิ้นสุดขั้นตอนติดตั้งให้ทำการรีบูตเครื่องใหม่เพื่อใช้งาน PF และ ALTQ

4.4 การใช้งาน PF และ ALTQ

การใช้งานระบบจะผ่าน Command Line Interface ของ FreeBSD ซึ่งจะเป็นการกระทำผ่าน Configuration file ที่เกี่ยวข้องโดยจะมีขั้นตอนดังต่อไปนี้

1. Interface Setup โดยการแก้ไข File “/etc/rc.conf” ของทั้ง 2 Interface (WAN/LAN) ดังคำสั่งด้านล่าง และหลังการ Setup เสร็จต้องมีการ Reboot ระบบใหม่

```
ifconfig_dc0="inet 192.168.5.1 netmask 255.255.255.0"
```

```
ifconfig_rl0="inet 192.168.1.20 netmask 255.255.255.0"
```

2. Queue Management โดยการแก้ไข File “/etc/pf.conf” เพื่อกำหนด Queue ให้แต่ละ Interface แยกกัน ดังคำสั่งด้านล่าง

```
# control download on internal interface
```

```
altq on dc0 bandwidth 1Mb cbq queue {std,www,ftp}
```

```
queue std bandwidth 100Kb cbq(default)
```

```
queue www bandwidth 100Kb
```

```
queue ftp bandwidth 800Kb
```

```
# control upload on external interface
```

```
altq on rl0 bandwidth 512Kb cbq queue {low,medium,high}
```

```
queue low bandwidth 20% cbq(default)
```

```
queue medium bandwidth 30%
```

```
queue high bandwidth 50%
```

3. Rule Management โดยการแก้ไข File “/etc/pf.conf” เหมือนการกำหนด Queue โดยต้องใส่หลังจากการกำหนด Queue แล้วเท่านั้น และกำหนดให้ Input ผ่านเข้ามาได้ทั้ง 2 Interface ดังคำสั่งด้านล่าง

```
# Pass in all interface
```

```
pass in all
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pass control download

pass out log on dc0 proto tcp from any to any queue ftp

Pass control upload

pass out log on r10 proto tcp from any to any flags S/SA keep state queue medium

4. Apply คำสั่งในการเรียกใช้งานค่าที่กำหนดให้กับระบบคือ

pfctl -f /etc/pf.conf

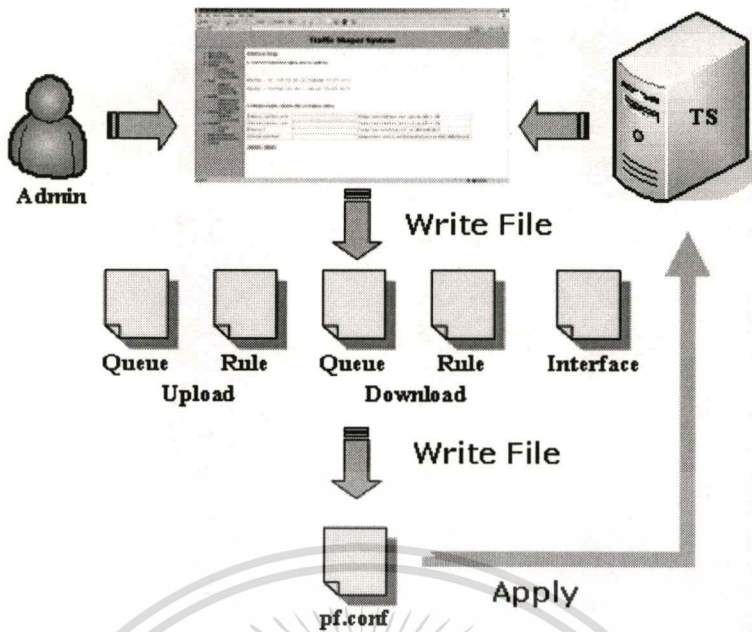
4.5 การพัฒนาโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย

การพัฒนาโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย พัฒนาโดยโปรแกรมภาษา PHP โดยให้โปรแกรมที่พัฒนามานั้นทำงานผ่านเว็บเบราว์เซอร์ โดยโปรแกรมจะมีหน้าที่ดังนี้

1. สามารถป้อนข้อมูลที่จำเป็นสำหรับการสร้างกฎของระบบควบคุมปริมาณการจราจรของระบบเครือข่าย ซึ่งระบุได้ถึง IP Source, IP Destination, Port Destination,
2. สามารถเพิ่มหรือลบกฎที่ทำการสร้างขึ้นได้
3. สามารถสำรองข้อมูลกฎเดิมก่อนที่จะทำการแก้ไขเปลี่ยนแปลงได้
4. สามารถแสดงผลการทำงานของระบบควบคุมปริมาณการจราจรของระบบเครือข่ายได้
5. สามารถแสดงผลของ Log การทำงานของกฎเพื่อใช้วิเคราะห์การทำงานของระบบ โดยมีรูปแบบการทำงานดังนี้

4.5.1 ไฟล์ระบบ

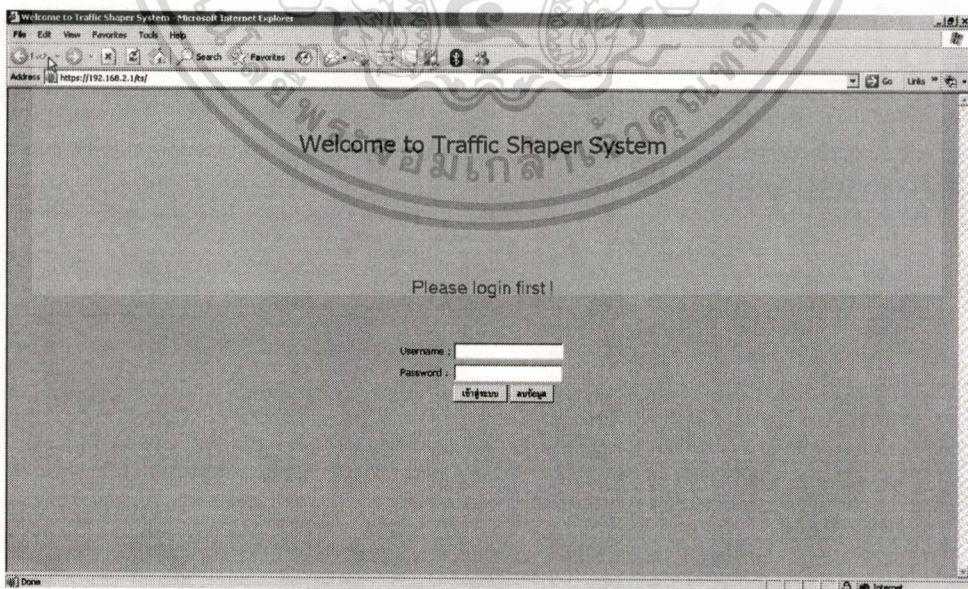
หลักการออกแบบการเก็บไฟล์ระบบที่ได้จากการกำหนดค่าผ่าน Web Base Interface แสดงดังรูปที่ 4.2 เมื่อผู้ดูแลระบบป้อนค่าต่างๆ พร้อมส่งค่าให้กับระบบแล้ว ระบบจะทำการเขียนไฟล์เพื่อบันทึกค่าลงในไฟล์ที่เกี่ยวข้องในแต่ละส่วน และจะทำการรวมกันเป็นไฟล์ระบบที่สมบูรณ์เพื่อส่งให้กับ PF ทำการประมวลผลคำสั่งต่อไป



รูปที่ 4.2 แสดงการเก็บ ไฟล์ระบบที่ได้จากการกำหนดค่าผ่าน Web Base Interface

4.5.2 ส่วนติดต่อกับผู้ใช้งาน

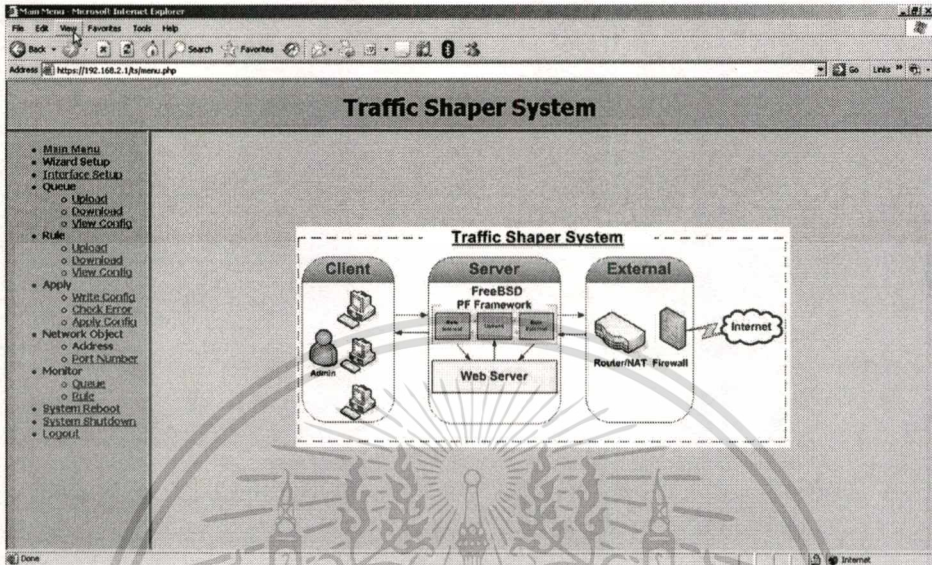
ส่วนติดต่อกับผู้ใช้งาน ในขั้นตอนแรกต้องทำการ Login เข้าสู่ระบบ โดยเปิดเว็บเบราว์เซอร์ และพิมพ์ <https://IP-Address/ts> ในช่อง Address ซึ่งในครั้งแรกใช้ชื่อรหัสผู้ใช้เป็น admin และรหัสผ่านเป็น “admin” จากนั้นระบบจะตรวจสอบข้อมูลผู้ใช้ ดังรูปที่ 4.3



รูปที่ 4.3 แสดงหน้าจอการเรียกใช้งานโปรแกรมในส่วน Login

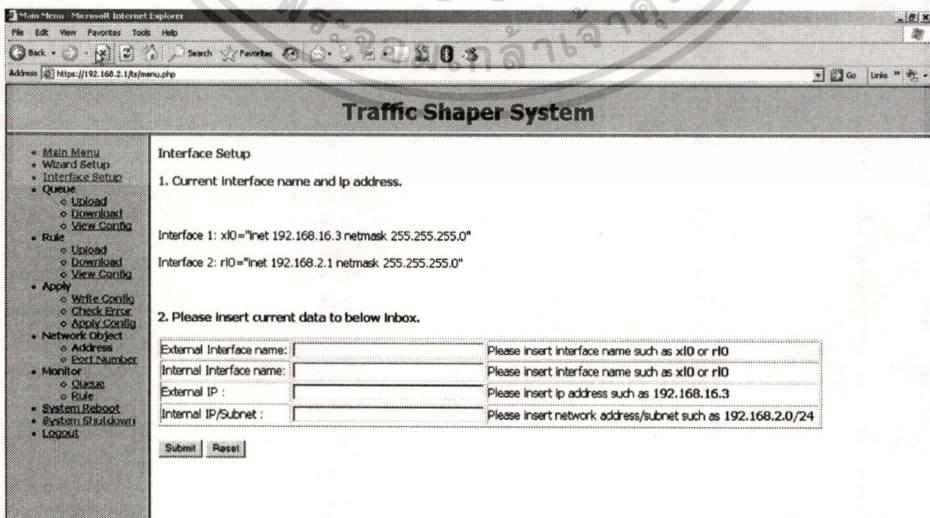
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 ระบบจะทำการแสดงหน้าจอเมนูหลัก ของโปรแกรม ซึ่งจะแบ่งการทำงาน ออกเป็น 3 ส่วนคือ ส่วนบนจะแสดงชื่อระบบ, ส่วนซ้ายจะแสดงรายการต่างๆ ของเมนู ที่สามารถ ทำงานได้ของระบบ และ ส่วนสุดท้ายคือส่วนทำงาน เพื่อป้อนค่าต่างๆ ให้กับระบบ ดังรูปที่ 4.4



รูปที่ 4.4 แสดงหน้าจอหน้าจอนำเสนอเมนูหลักของโปรแกรม

ขั้นตอนที่ 3 ต้องทำการกำหนดค่าพื้นฐานเบื้องต้น ให้กับระบบก่อน โดยให้เลือกที่ Interface Setup ในส่วนเมนูคำสั่ง โดยระบบจะแสดงรายละเอียดของ Interface ของระบบ ให้ป้อนค่าที่แสดงเพื่อกำหนดให้กับระบบ ดังรูปที่ 4.5

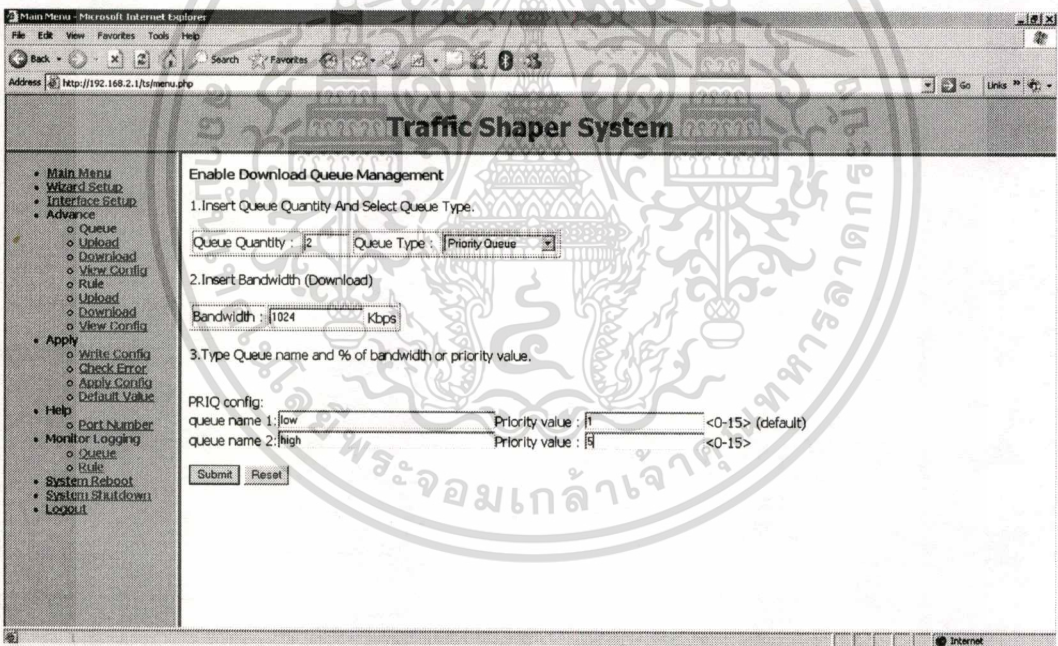


รูปที่ 4.5 แสดงหน้าจอการกำหนดค่าพื้นฐานเบื้องต้นให้ Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 4 จะเป็นการกำหนดรายละเอียดของ Queue ที่ต้องการสร้างให้ตรงกับความต้องการ โดยจะแบ่งออกเป็น 2 ส่วนคือ Upload และ Download ดังแสดงในรูปที่ 4.6 ซึ่งจะต้องกำหนดค่าต่างๆ ให้กับระบบดังนี้

- **Bandwidth** จะเป็นการกำหนดค่า bandwidth ทั้งหมดที่สามารถใช้งานได้ ในส่วน Upload หรือ Download
- **Queue Type** จะเป็นการเลือกรูปแบบของ Queue ที่ต้องการตามการใช้งาน โดยสามารถเลือกได้ระหว่าง Class Base Queue หรือ Priority Queue
- **Queue Config** จะเป็นการกำหนดรายละเอียดของ Queue ที่เลือก โดย Class Base Queue จะต้องกำหนดชื่อของ Queue ใน Class ต่างๆ และเปอร์เซ็นต์ที่กำหนดให้แต่ละ Class ในส่วน Priority Queue จะต้องกำหนดชื่อของ Queue ในแต่ละระดับความสำคัญ และใส่ค่าระดับความสำคัญ



รูปที่ 4.6 แสดงหน้าจอการกำหนดค่าต่างๆ ของ Queue

Traffic Shaper System

- Main Menu
- Wizard Setup
- Interface Setup
- Queue
 - Upload
 - Download
 - View Config
- Rule
 - Upload
 - Download
 - View Config
- Apply
 - Write Config
 - Check Error
 - Apply Config
- Network Object
 - Address
 - Port Number
- Monitor
 - Queue
 - Rule
- System Reboot
- System Shutdown
- Logout

Upload Queue Config

```
altq on $ext_if bandwidth 256Kb cbq queue { everyone, staff, senior, executive, manager }
queue everyone bandwidth 30% cbq(default)
queue staff bandwidth 30%
queue senior bandwidth 20%
queue executive bandwidth 10%
queue manager bandwidth 10%
```

Download Queue Config

```
altq on $int_if bandwidth 512Kb cbq queue { dengineer, daccount, dmarketing, ddesign, dother }
queue dengineer bandwidth 50% cbq(default)
queue daccount bandwidth 30%
queue dmarketing bandwidth 10%
queue ddesign bandwidth 5%
queue dother bandwidth 5%
```

รูปที่ 4.7 แสดงหน้าจอแสดง Queue ที่สร้างเพื่อใช้ทดสอบระบบ

ขั้นตอนที่ 5 จะเป็นการกำหนดรายละเอียดของ Rule ที่ต้องการสร้างให้ตรงกับ Queue ที่กำหนด โดยจะแบ่งออกเป็น 2 ส่วนคือ Upload และ Download ดังแสดงในรูปที่ 4.8 ซึ่งจะต้องกำหนดค่าต่างๆ ให้กับระบบดังนี้

- **From** จะเป็นการกำหนด IP Address ต้นทาง
- **To** จะเป็นการกำหนด IP Address ปลายทาง
- **Protocol** จะเป็นการกำหนด Protocol ปลายทางที่จะใช้งาน
- **Port Number** จะเป็นการกำหนด Port Number ปลายทางที่จะใช้งาน

Traffic Shaper System

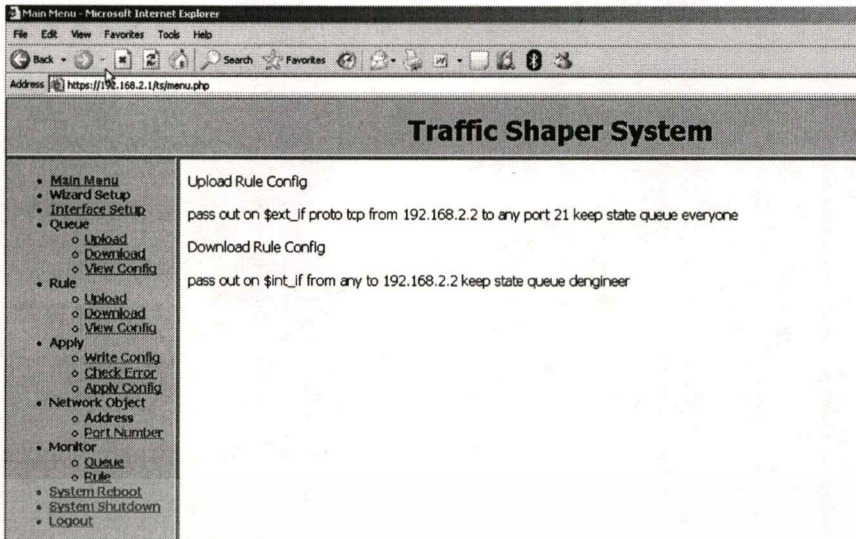
- Main Menu
- Wizard Setup
- Interface Setup
- Queue
 - Upload
 - Download
 - View Config
- Rule
 - Upload
 - Download
 - View Config
- Apply
 - Write Config
 - Check Error
 - Apply Config
- Network Object
 - Address
 - Port Number
- Monitor
 - Queue
 - Rule
- System Reboot
- System Shutdown
- Logout

Rule Upload Management

From :	<input type="text"/>	any "or" ip address
To :	<input type="text"/>	any "or" ip address
Protocol :	None	blank "or" select protocol
Port Number :	<input type="text"/>	blank "or" insert port number
Queue :	everyone	Select queue name to assign

รูปที่ 4.8 แสดงหน้าจอการกำหนดค่าต่างๆ ของ Rule

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงหน้าจอแสดง Rule ที่สร้างเพื่อใช้ทดสอบระบบ

ขั้นตอนที่ 6 จากนั้นเลือก Apply Menu กำหนดให้มีผลทำงานกับระบบ โดยจะเป็นการเรียกใช้คำสั่งของ PF ผ่านคำสั่ง PHP Language System Command ดังนี้

```
pfctl -f /usr/local/www/apache22/data/ts/pf.con
```

ซึ่งเมื่อตรวจสอบโดยใช้คำสั่งของ FreeBSD คือ pf-s queue และ pf-s rule ได้ผลดังรูปที่ 4.10

```
ts# pfctl -s queue
queue root_lnc0 bandwidth 1.02Mb priority 0 cbq( mrr root ) (ssh, mmm)
queue ssh bandwidth 204.00Kb cbq( borrow default )
queue mmm bandwidth 819.20Kb cbq( borrow )
ts# pfctl -s rule
pass in all
pass out all
pass out on lnc0 inet proto tcp from 192.168.2.1 port = ssh to 192.168.2.2 keep
state queue ssh
pass out on lnc0 inet proto tcp from 192.168.2.1 port = http to 192.168.2.2 keep
state queue mmm
ts#
```

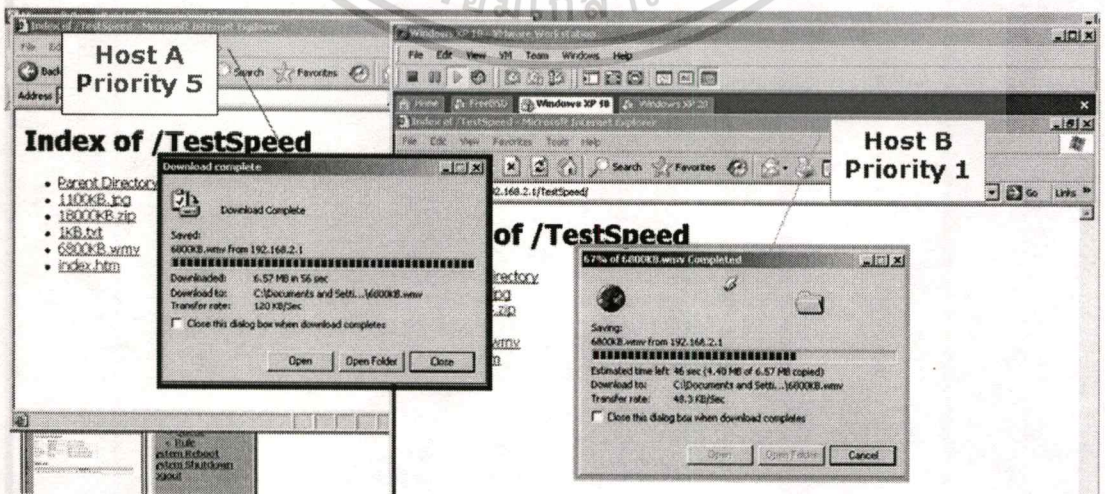
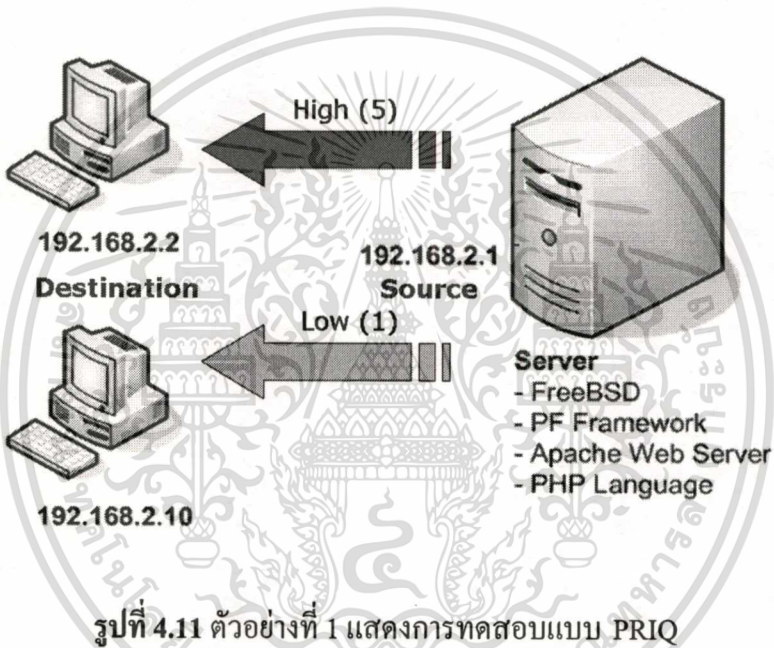
รูปที่ 4.10 แสดง Queue และ Rule ที่ถูกสร้างให้กับระบบ

หมายเหตุ รายละเอียดและวิธีการใช้งานโปรแกรมได้จัดไว้ในภาคผนวก ข

4.6 ผลการทดสอบโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย

การทดสอบการทำงานของโปรแกรมควบคุมปริมาณการจราจรของระบบเครือข่าย จะเริ่มทำการทดสอบโดยวิธีการ Download และ Upload ข้อมูลจาก Server ที่ให้บริการ โดยใช้โปรแกรม SSH Secure File Transfer และ Web Browser ซึ่งจะทำการเปรียบเทียบอัตราการรับ-ส่งข้อมูล ก่อนและหลังการกำหนด Queue ทั้ง 2 แบบ และ Rule ให้กับระบบ เพื่อดูผลของการควบคุมอัตราการรับ-ส่งข้อมูล ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 1 จะเป็นการทดสอบ Download ข้อมูลจาก Server โดยมีการสร้าง Queue แบบ PRIQ ไว้ 2 Queue และทำการ Download ข้อมูลพร้อมกันจะได้ผลทดลองดังรูปที่ 4.11

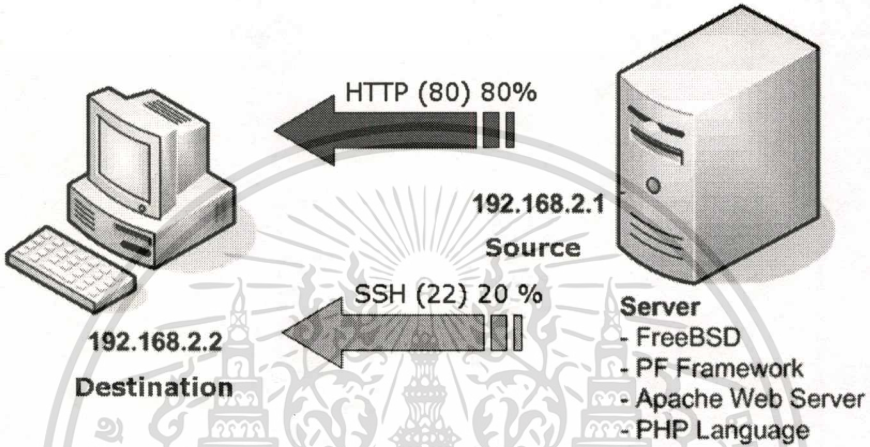


รูปที่ 4.12 แสดงการผลทดสอบแบบ PRIQ

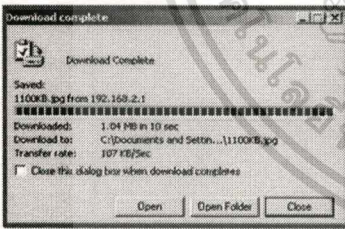
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายกเว้นกรณีอื่น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.12 จะแสดงผลการทดสอบการสร้าง Queue แบบ PRIQ โดยจะประกอบด้วย High Queue และ Low Queue ซึ่งจากผลการทดสอบเครื่องที่มี PRIQ ที่เป็น High จะได้รับการบริการก่อนจนเสร็จเป็นลำดับแรก

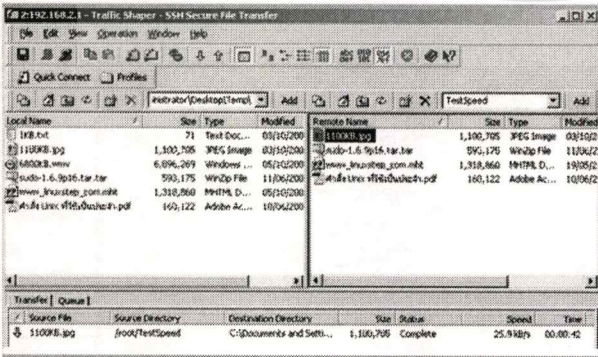
ตัวอย่างที่ 2 จะเป็นการทดสอบ Download ข้อมูลจาก Server โดยมีการสร้าง Queue แบบ CBQ ไว้ 2 Queue และทำการ Download ข้อมูลพร้อมกันโดยใช้บริการคนละชนิดคือ HTTP และ SSH จะได้ผลทดลองดังรูปที่ 4.13



รูปที่ 4.13 ตัวอย่างที่ 2 แสดงการทดสอบแบบ CBQ



Service HTTP
BW=80%=800Kbps
Speed= 107KBpsx8= 856kbps

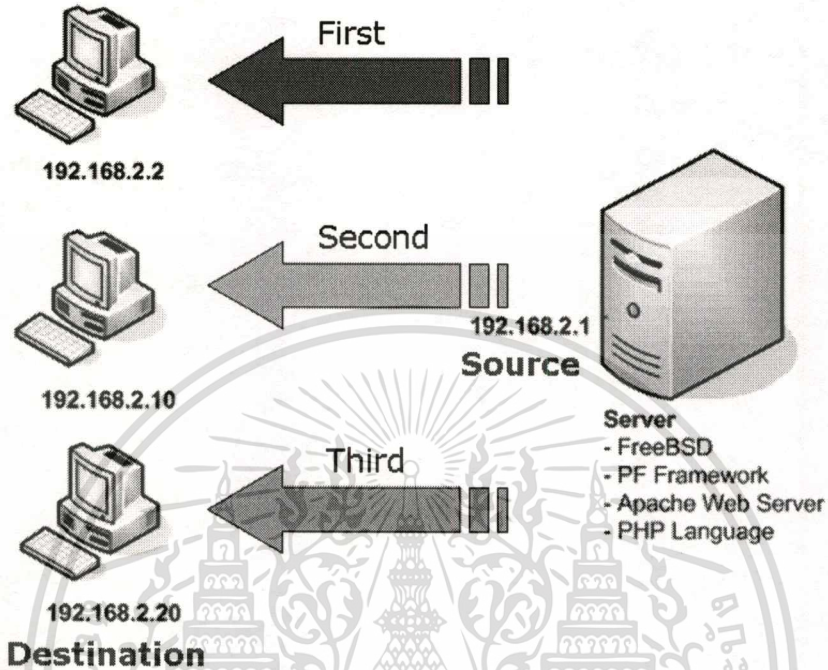


Service SSH
BW=20%=200Kbps
Speed= 25.9KBpsx8= 207Kbps

รูปที่ 4.14 แสดงการผลทดสอบแบบ CBQ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 3 จะเป็นการกำหนดค่าแบบง่ายให้กับระบบโดยใช้ Wizard Setup ซึ่งระบบจะมีการสร้าง Queue มาตรฐาน 3 Queue คือ First, Second และ Third เพื่อให้เกิดการใช้งานที่ง่าย ดังรูปที่ 4.15



รูปที่ 4.15 ตัวอย่างที่ 3 แสดงการทดสอบแบบ Wizard

สรุปผลการทำงานของโปรแกรมควบคุมปริมาณการจราจรของเครือข่าย สามารถจะทำการควบคุมปริมาณการจราจรของเครือข่ายได้อย่างถูกต้อง ตามกฎที่ได้สร้างไว้ โดยผู้ใช้สามารถควบคุมการทำงานได้อย่างสะดวก ตามความต้องการ อย่างไม่จำกัดตามผู้ใช้ควรคำนึงถึงนโยบายการใช้งานขององค์กรเป็นหลัก ว่าส่วนใด หรือบริการใดที่จะควบคุม เพื่อให้เกิดประสิทธิภาพสูงสุด

บทที่ 5

บทสรุปและแนวทางการพัฒนาในอนาคต

5.1 สรุปผลการพัฒนา

จากผลการทดสอบระบบควบคุมปริมาณการจราจรของเครือข่าย ที่ได้พัฒนานี้ ทำให้เราได้เข้าใจวิธีและกระบวนการทำงานของการจัดการแบนวิด ทั้งในส่วนรับและส่ง รวมทั้งเข้าใจการทำงานของการจัดการ Queue ในรูปแบบต่างๆ เพื่อนำมาประยุกต์ใช้งานในโครงการนี้ ผ่านทาง PF Framework ของฟรีบีเอสดี ซึ่งจะเห็นว่า PF เป็นเครื่องมือที่มีประสิทธิภาพที่ดีอย่างยิ่ง และยังไม่มีค่าใช้จ่ายในด้านลิขสิทธิ์ ในการนำมาใช้งานภายในองค์กรต่างๆ ทำให้แต่ละองค์กรไม่ต้องเสียค่าใช้จ่ายในการซื้ออุปกรณ์ที่มีราคาสูงมาใช้งาน สำหรับการพัฒนาในส่วนติดต่อกับผู้ใช้งานระบบทางผู้พัฒนามีข้อจำกัดทางการพัฒนาพอสมควร เนื่องจากผู้พัฒนาไม่มีประสบการณ์ในการพัฒนาระบบในแบบ Web Base Application และยังคงเขียนคำสั่งเพื่อเรียกใช้คำสั่งของระบบฟรีบีเอสดี ผ่าน Web Base Interface อีกด้วย ซึ่งต้องอาศัยความชำนาญทั้งทางด้าน โปรแกรมมิ่งและคำสั่งระบบของฟรีบีเอสดี ความถนัดกันจึงจะสามารถสร้าง User Interface ที่มีคุณภาพที่ดีได้ จึงมีผลทำให้คำสั่งระบบบางอย่างที่มีความซับซ้อน ผู้พัฒนาไม่สามารถเขียน User Interface สั่งงานได้อย่างมีประสิทธิภาพหรือเกิดข้อผิดพลาดระหว่างการทำงานบางประการได้ ดังนั้นทางผู้พัฒนาจึงได้พยายามพัฒนาในแต่ละส่วนให้มีความซับซ้อนน้อยที่สุด เช่น Wizard Setup เพื่อช่วยให้ใช้งานพบปัญหาระหว่างการใช้น้อยที่สุด

5.2 ข้อจำกัดของระบบ

เนื่องจากโปรแกรมควบคุมปริมาณการจราจรของเครือข่ายนี้ ได้ถูกพัฒนาขึ้นเพื่อให้ผู้ใช้งาน สามารถใช้งานได้โดยสะดวก ดังนั้นการทำงานกับ Queue และ Rule ที่ซับซ้อน อาจจะทำให้ผู้ใช้งานเกิดความสับสนในการใช้งานได้ ผู้พัฒนาจึงได้จัดเตรียม Queue พื้นฐานมาให้ 2 ชนิดคือ Class Base Queue และ Priority Queue และการแก้ไข Queue และ Rule ต้องสร้างแทนที่ไฟล์เดิมเท่านั้น ส่วนการ Monitor จะแสดงเป็นลักษณะ Text Base ทำให้การตรวจสอบอาจจะเข้าใจได้ยากจากข้อความที่ปรากฏ ดังนั้นถ้าผู้ใช้ไม่มีพื้นฐานด้านระบบเครือข่ายมาก่อน อาจจะทำให้เข้าใจได้ยาก จากการแสดงผล

5.3 สรุปแนวทางในการพัฒนาในอนาคต

แนวทางในการพัฒนาในอนาคต จะเป็นการเพิ่มความสามารถในส่วนที่เป็นข้อจำกัดของระบบ ที่ได้กล่าวมาแล้ว คือ การเพิ่มจำนวนและรูปแบบ Queue แบบต่างๆ ได้มากกว่าเดิม รวมทั้งค่า Option ต่างๆ ที่มีผลกับการจัดการ Queue และสามารถแก้ไขเปลี่ยนแปลงค่า Queue และ Rule ต่างๆ ได้ง่ายขึ้นกว่าเดิม พร้อมทั้งความสามารถที่จะตรวจสอบ Monitor ระบบได้ในรูปแบบที่ง่ายขึ้น เพื่อง่ายต่อการตรวจสอบการทำงานต่างๆ ต่อไป



บรรณานุกรม

กิตติพงษ์ สุวรรณราช. 2548, **รู้จักกับระบบปฏิบัติการ FreeBSD.**

<http://bsd.psru.ac.th/microcom/micro235.htm>.

วิกิพีเดีย สารานุกรมเสรี. 2007, **FreeBSD.**

<http://th.wikipedia.org/w/index.php?title=FreeBSD&oldid=721509>.

Flickenger R., Belcher M., Canessa E., Zennaro M. 2006, **How To Accelerate Your Internet.**

<http://www.eslared.org.ve/articulos/bwmo-ebook.pdf>.

FreeBSD Foundation. 2007, **The OpenBSD Packet Filter (PF) and ALTQ.**

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls-pf.html.

Hui Zhang. 1999, **Packet Hierarchical Packet Schedulers.**

<http://www.cs.cmu.edu/~hzhang/HFSC/main.html>.

Kenjiro Cho. 1999, **Managing traffic with ALTQ.**

http://www.usenix.org/events/usenix99/full_papers/cho/cho.pdf.

Kenjiro Cho. 2001, **The Design and Implementation of the ALTQ Traffic Management System.** <ftp://ftp.sonydsl.co.jp/pub/kjc/papers/dissertation.ps.gz>.

Leonardo Balliache. 2002, **Linux Queuing Disciplines.**

<http://www.opalsoft.net/qos/DS-20.htm>.

OpenBSD Foundation. 2007, **Packet Queueing and Prioritization.**

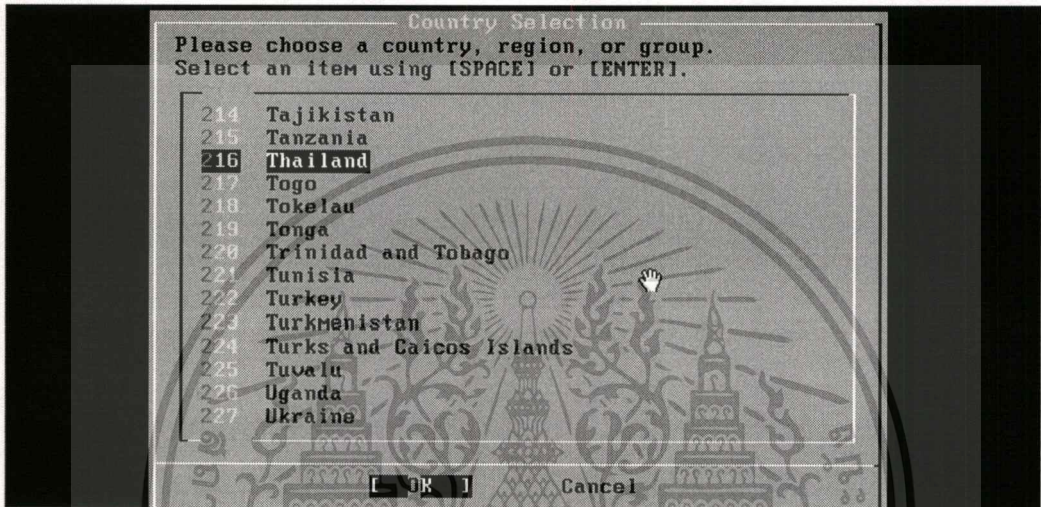
<http://www.openbsd.org/faq/pf/queueing.html>.

ภาคผนวก ก

คู่มือการติดตั้งระบบ

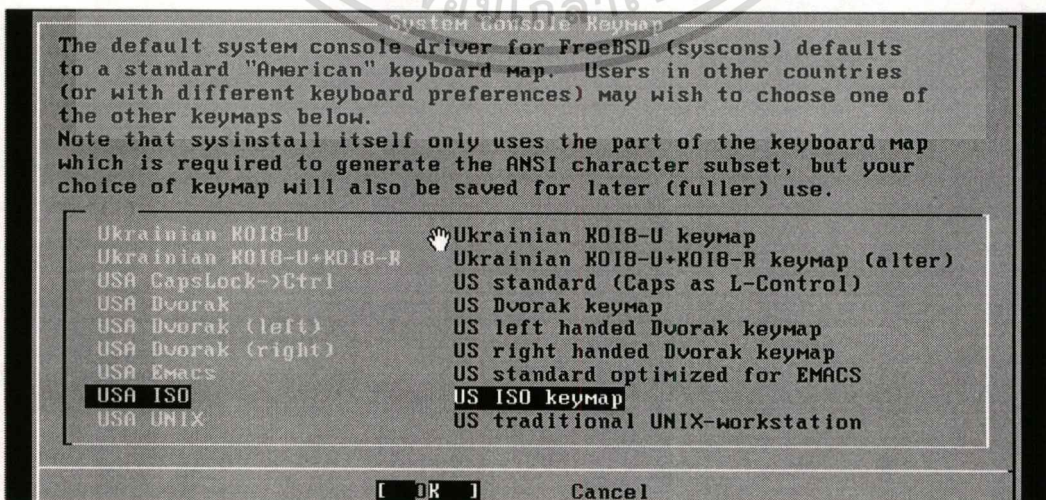
การติดตั้งระบบปฏิบัติการฟรีบีเอสดี เวอร์ชัน 6.2

1. เริ่มต้นด้วยการบูทเครื่องเซิร์ฟเวอร์จากแผ่น FreeBSD 6.2 Installation Disc 1 รอจนปรากฏหน้าจอภาพให้เลือกประเทศ (ลำดับที่ 216 Thailand) แล้วกดปุ่ม Enter เพื่อทำงานต่อ ดังรูปที่ 1



รูปที่ 1 เลือกประเทศ ลำดับที่ 216 Thailand

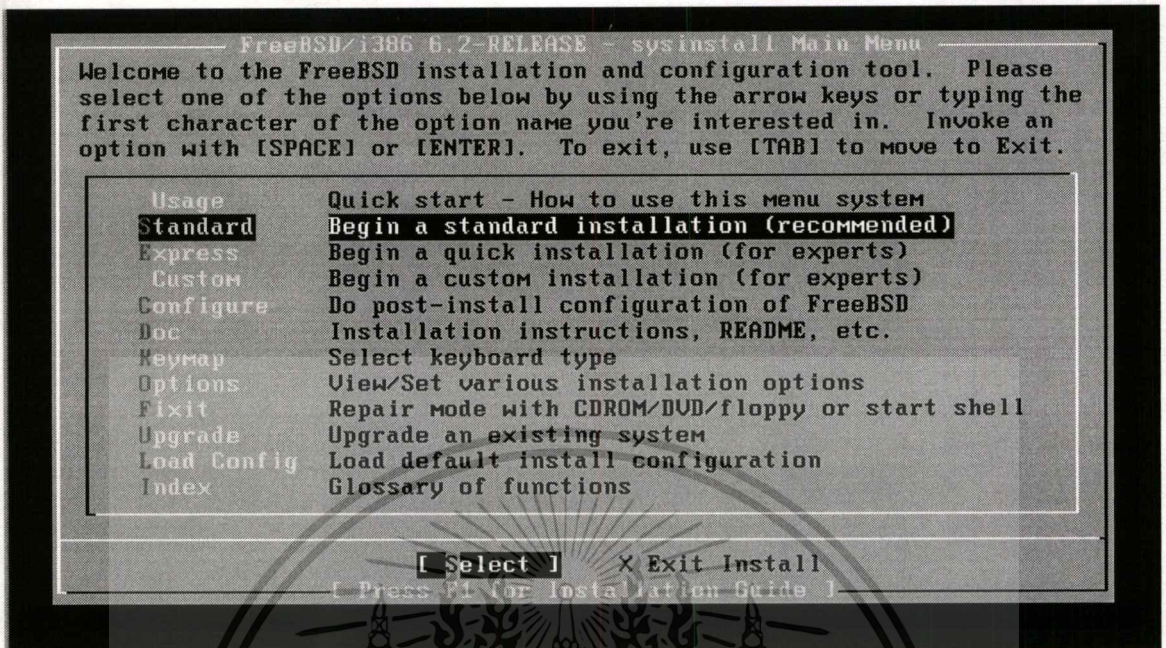
2. เลือกคีย์บอร์ดเพื่อใช้งาน ในที่นี้จะเลือกเป็น USA ISO ซึ่งถือว่าเป็น Keyboard มาตรฐานที่เราสามารถพิมพ์สั่งงานระบบได้ ดังรูปที่ 2



รูปที่ 2 เลือกคีย์บอร์ดที่ใช้งาน

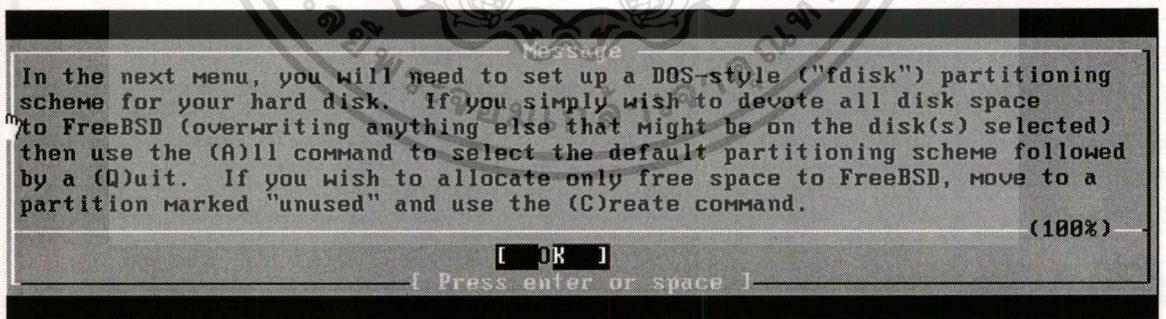
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือก Standard และกดปุ่ม Enter เพื่อเริ่มการติดตั้งระบบปฏิบัติการ FreeBSD ดังรูปที่ 3



รูปที่ 3 เลือก Standard Installation

4. โปรแกรมติดตั้งระบบปฏิบัติการ FreeBSD จะแจ้งให้เราทราบว่า จะจัดการกับโครงสร้างของ Partition ของ Hard disk ของเรา เมื่ออ่านคำแนะนำเสร็จแล้วก็สามารถกดปุ่ม Enter เพื่อเริ่มการจัดการกับ Partition ดังรูปที่ 4



รูปที่ 4 แนะนำเกี่ยวกับการจัดการ Partition

5. เราสามารถสร้าง Partition ได้โดยการกดปุ่มตัวอักษร C หรือมีความหมายเป็น Create เพื่อสร้าง Partition ใหม่ จากนั้นระบบจะแสดงขนาดของ Block Size ออกมาให้เราทราบ ซึ่งก็ขอให้ยอมรับว่า Block Size ทั้งหมด เนื่องจากเราต้องการสร้าง Partition ใหม่นี้เป็นโครงสร้างที่ใช้กับเอกสารระบบปฏิบัติการ FreeBSD ให้เรากดปุ่ม Enter ดังรูปที่ 5 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Disk name:      dd0
DISK Geometry: 12483 cyls/16 heads/63 sectors = 12582864 sectors (6143MB)

Offset      Size(ST)      End      Name      PType      Desc      Subtype      Flags
-----
0           12582912      12582911      -         12         unused      0

Value Required
Please specify the size for new FreeBSD slice in blocks
or append a trailing 'M' for megabytes (e.g. 20M).
12582912

The follow [ OK ] Cancel

A = Use Entire Disk  G = set Drive Geometry  C = Create Slice  F = 'DD' mode
D = Delete Slice     Z = Toggle Size Units   S = Set Bootable  ; = Wizard m.
T = Change Type      U = Undo All Changes   Q = Finish

Use F1 or ? to get more help, arrow keys to select.

```

รูปที่ 5 เลือกขนาดของ Partition ทั้งหมด

6. ให้กำหนดค่าชนิดของ Partition ให้กำหนดเป็น 165 ซึ่งเป็นมาตรฐานที่ใช้ในระบบปฏิบัติการ FreeBSD ให้กดปุ่ม Enter ดังรูปที่ 6

```

Disk name:      dd0
DISK Geometry: 12483 cyls/16 heads/63 sectors = 12582864 sectors (6143MB)

Offset      Size(ST)      End      Name      PType      Desc      Subtype      Flags
-----
Value Required
Enter type of partition to create:

Pressing Enter will choose the default, a native FreeBSD
slice (type 165). Other popular values are 6 for a
DOS FAT partition, 131 for a Linux ext2fs partition, or
138 for a Linux swap partition.

Note: If you choose a non-FreeBSD partition type, it will not
be formatted or otherwise prepared, it will simply reserve space
for you to use another tool, such as DOS format, to later format
and actually use the partition.

The fo [ OK ] Cancel

A = Us mode
D = De rd m.
T = Ch
165

```

รูปที่ 6 เลือก Partition Type

7. โปรแกรมจะแสดงโครงสร้างของ Hard Disk ที่เราได้กำหนดและแบ่ง Partition ให้เราทราบ ซึ่งดูรายละเอียดต่าง ๆ เรียบร้อยแล้วให้กดปุ่ม Q ดังรูปที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Disk name:      ad0
DISK Geometry: 12483 cyls/16 heads/63 sectors = 12582864 sectors (6143MB)
FDISK Partition Editor

Offset      Size(ST)      End      Name      PType      Desc      Subtype      Flags
   0         63          62       -         12         unused     0
   63      12582801    12582863  ad0s1     8          freebsd    165
12582864     48      12582911  -         12         unused     0

The following commands are supported (in upper or lower case):

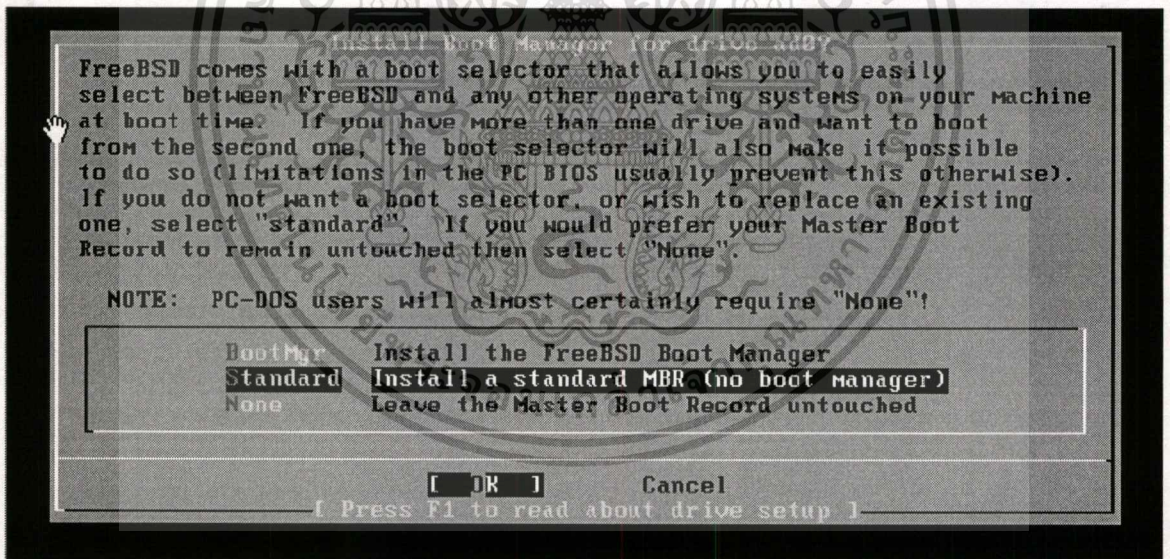
A = Use Entire Disk      G = set Drive Geometry      C = Create Slice      F = 'DD' mode
D = Delete Slice        Z = Toggle Size Units      S = Set Bootable      ; = Wizard m.
T = Change Type         U = Undo All Changes       Q = Finish

Use F1 or ? to get more help, arrow keys to select.

```

รูปที่ 7 แสดง โครงสร้างของ Partition

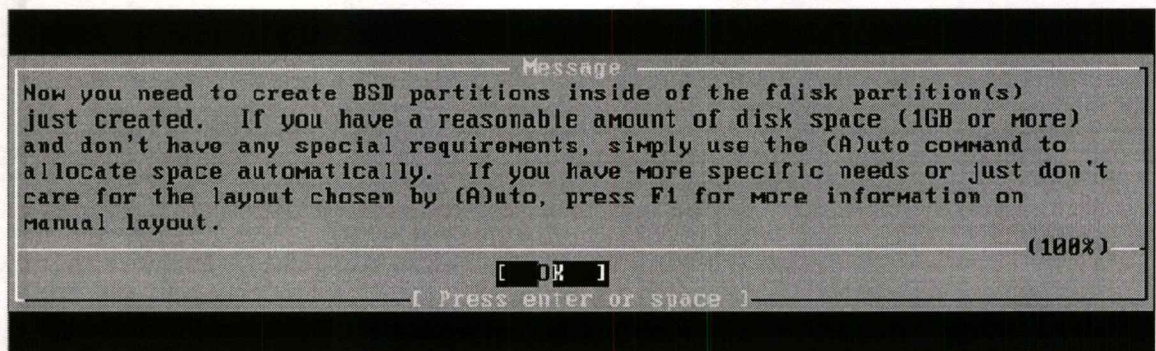
8. เลือก Boot Manager ของ Hard Disk (ad0) เป็น Standard ดังรูปที่ 8



รูปที่ 8 เลือก Boot Manager

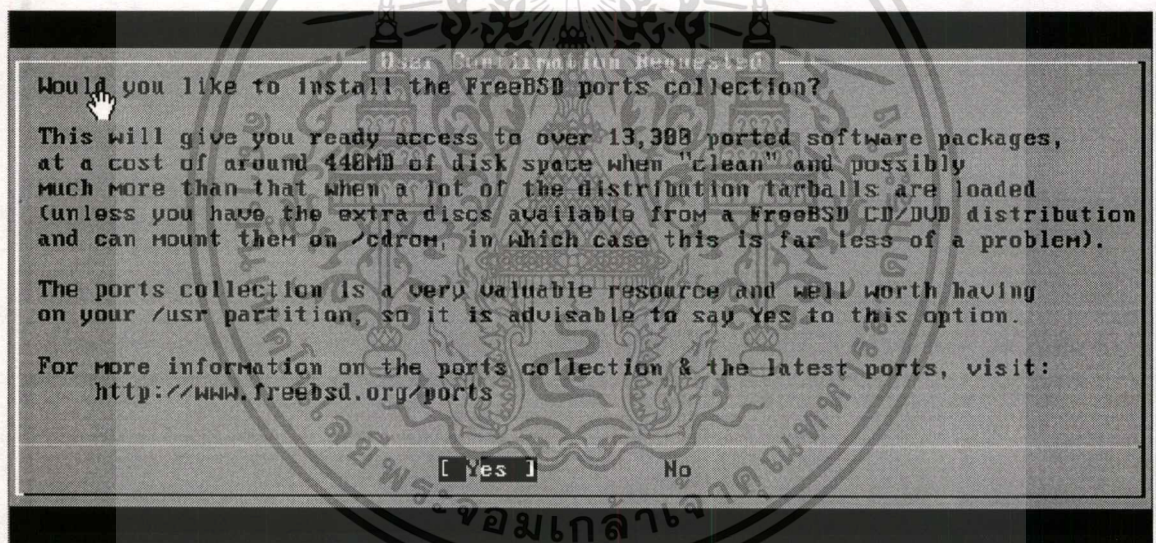
9. โปรแกรมติดตั้งจะแสดงคำแนะนำเพื่อบอกให้เราทราบเกี่ยวกับการแบ่ง Partition ออกเป็นส่วนต่าง ๆ โปรแกรมก็ได้แนะนำให้เราแบ่ง Partition ง่าย ๆ โดยการกดปุ่ม A หรือ Automatic ก็จะเป็นอันเสร็จกระบวนการตอนนี้ หากดำเนินการเกี่ยวกับ Partition เรียบร้อยแล้วก็สามารถกดปุ่ม Q เพื่อเสร็จสิ้นการแบ่ง Partition ในส่วนนี้ ดังรูปที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



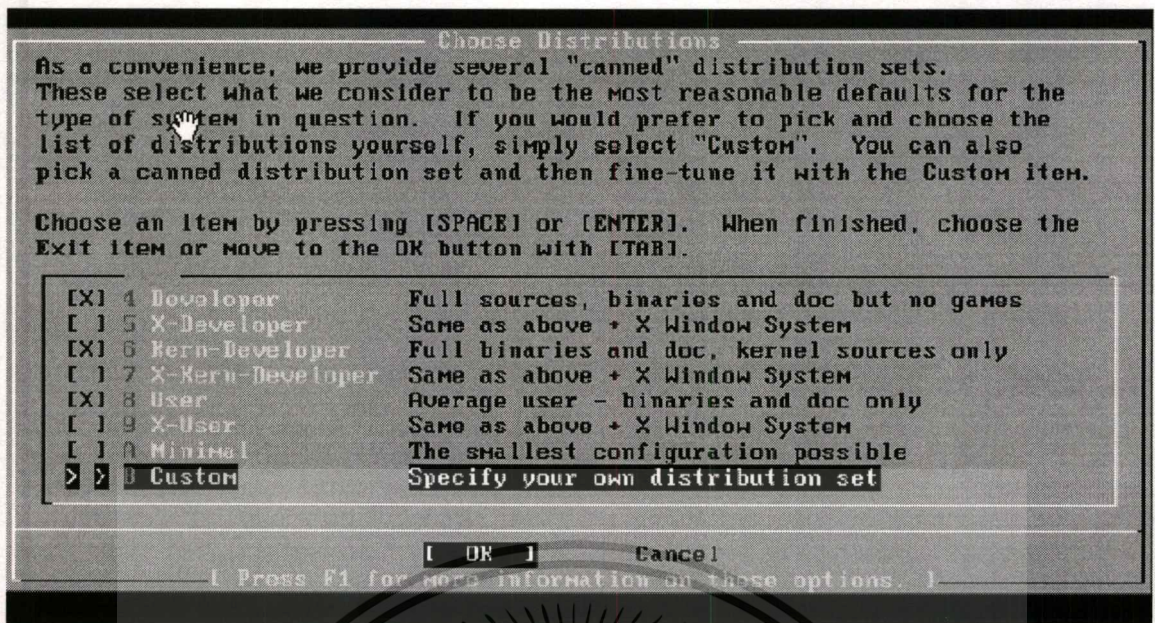
รูปที่ 9 แสดงคำแนะนำในการแบ่ง Partition

10. โปรแกรมติดตั้งจะแสดงรายละเอียดให้เราทราบว่า ในระบบปฏิบัติการ FreeBSD นั้นมีโปรแกรมหรือ Packages ต่าง ๆ ให้เราได้เลือกติดตั้ง เมื่ออ่านคำแนะนำเสร็จแล้วก็กดปุ่ม Enter ดังรูปที่ 10



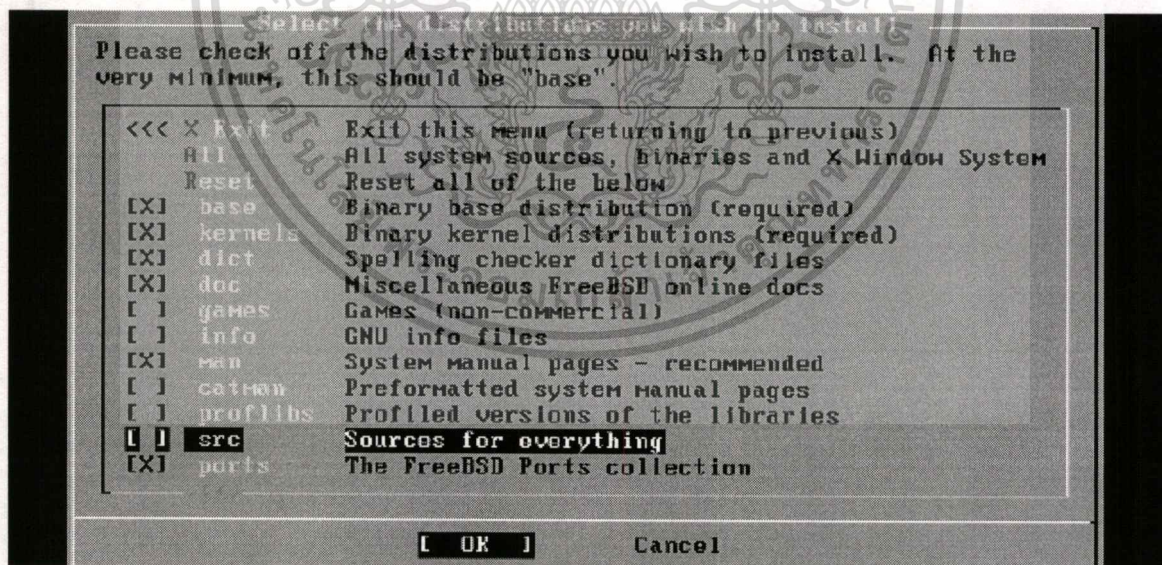
รูปที่ 10 คำแนะนำเกี่ยวกับ Packages ต่าง ๆ

11. โปรแกรมจะให้เราเลือกประเภทของ Distributions ที่เราต้องการติดตั้ง ได้แก่ตัวเลือกที่ 4, 6, 8 และ B Custom เท่านั้น ดังรูปที่ 11



รูปที่ 11 เลือก Distributions เพื่อใช้งาน

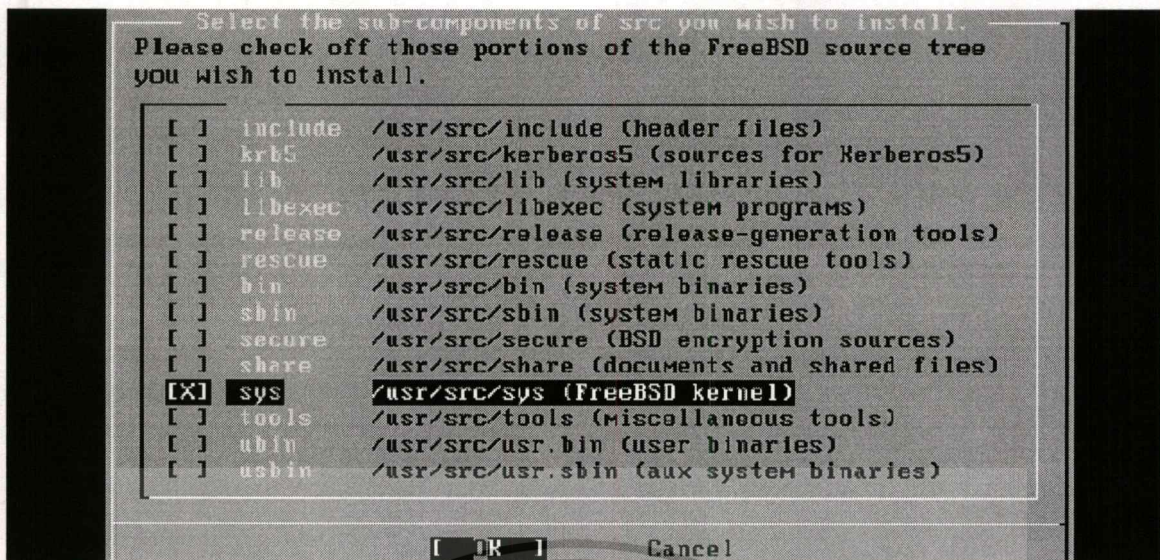
12. เมื่อเราเลือกตัวเลือก B Custom แล้ว ก็จะปรากฏหน้าจอให้เราเลือกตัวเลือกย่อยอีก ในที่นี้จะเลือกตัวเลือก [] src โดยการกดปุ่ม Space Bar เพื่อเป็นการเลือก ดังรูปที่ 12



รูปที่ 12 เลือกตัวเลือก src ที่อยู่ใน B Custom

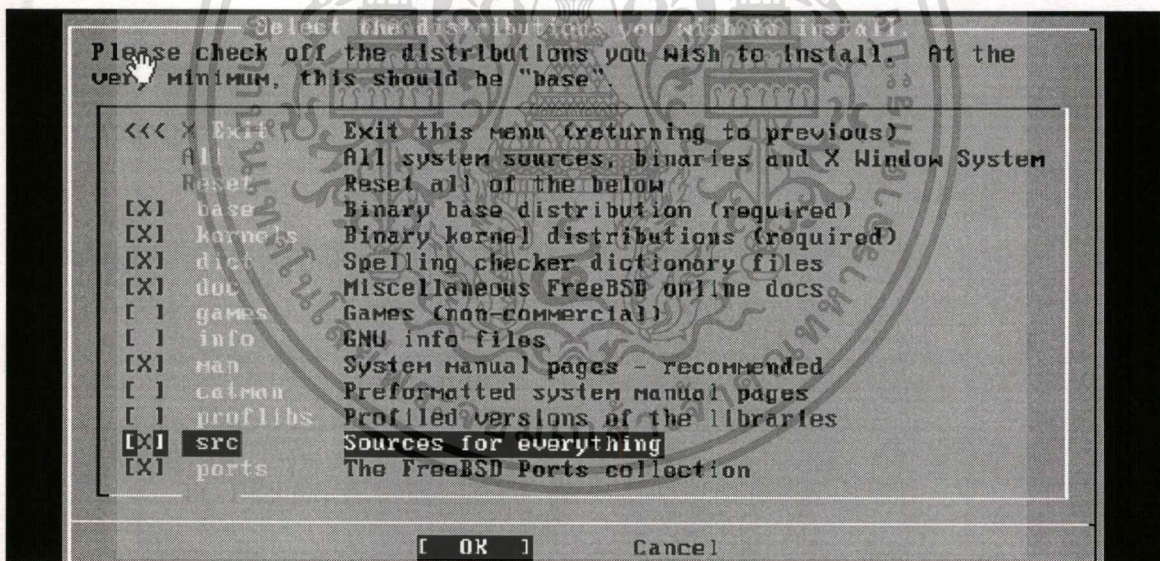
13. ให้เลือกตัวเลือก [] sys โดยการกดปุ่ม Space Bar เพื่อเลือก ซึ่งตัวเลือกนี้จะเป็น FreeBSD Kernel จากนั้นให้กดปุ่ม TAB มาที่ OK ดังรูปที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13 เลือกตัวเลือก sys (FreeBSD Kernel)

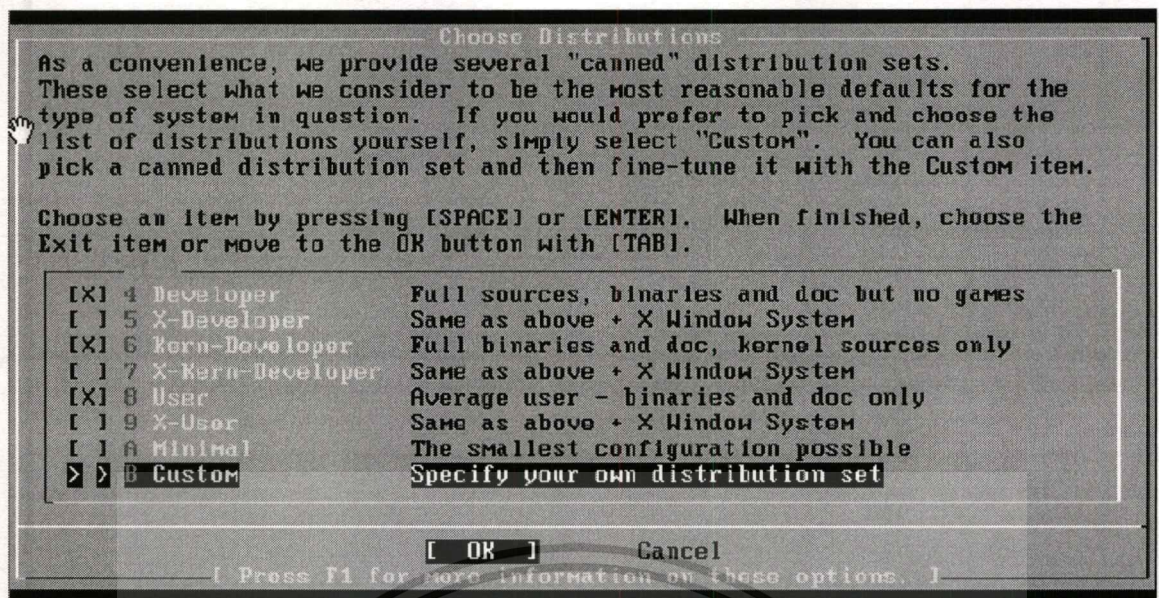
14. กดปุ่ม TAB มาที่ OK อีกครั้งเพื่อเป็นการย้อนกลับ ไปเมนูก่อนหน้านี้ ดังรูปที่ 14



รูปที่ 14 การกดปุ่ม TAB เพื่อย้อนกลับเมนู

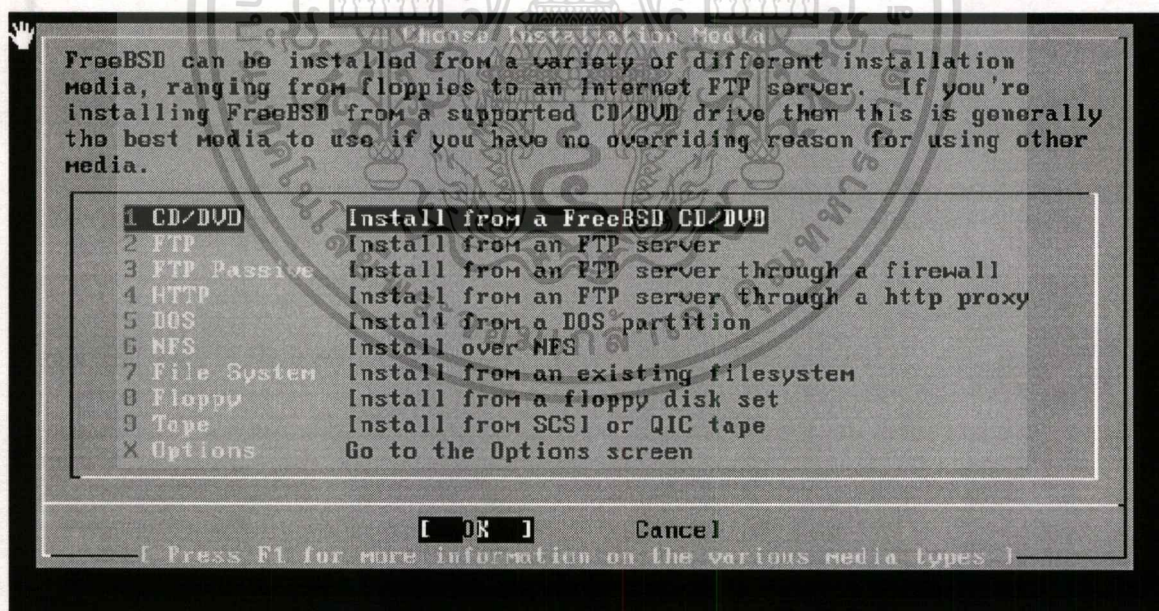
15. กดปุ่ม TAB มาที่ OK เพื่อย้อนกลับ ไปเมนูก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 15 กดปุ่ม TAB เพื่อย้อนกลับไปเมนูก่อนหน้านี้

16. โปรแกรมติดตั้งจะให้เราเลือกว่าต้องการติดตั้งระบบปฏิบัติการ FreeBSD จากสื่อประเภทใด ในที่นี้แนะนำเป็นตัวเลือกที่ 1 คือ CD/DVD แล้วกดปุ่ม Enter ดังรูปที่ 16



รูปที่ 16 เลือกสื่อที่ใช้ในการติดตั้ง FreeBSD

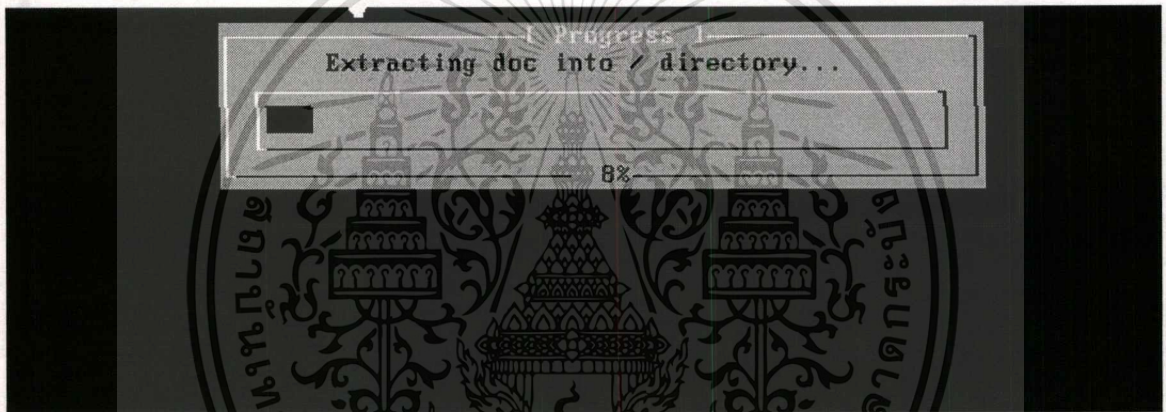
17. ตอบ Yes เพื่อทำการติดตั้งติดตั้งต่อไป ดังรูปที่ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



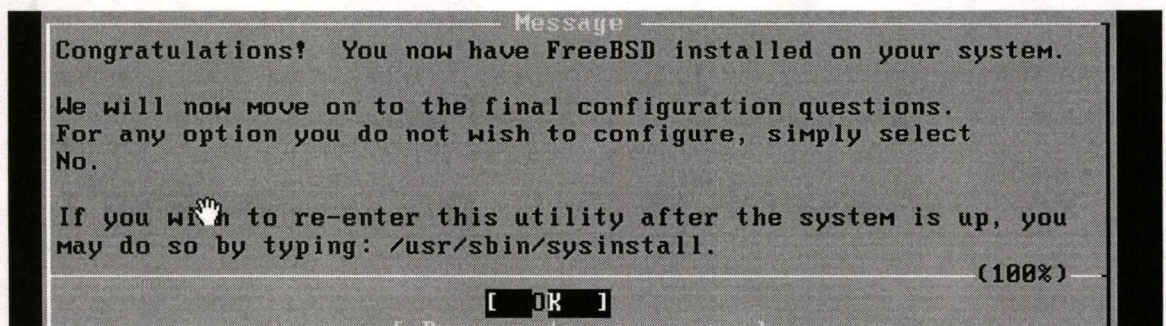
รูปที่ 17 โปรแกรมถามความแน่ใจในการติดตั้ง

18. หลังจากที่ยืนยันการติดตั้งระบบปฏิบัติการ FreeBSD แล้วโปรแกรมจะเริ่มทำการ Format Hard Disk ของเครื่องเซิร์ฟเวอร์แล้วติดตั้งโปรแกรมต่าง ๆ ไปตามโครงสร้างที่ได้กำหนดไว้ ดังรูปที่ 18



รูปที่ 18 โปรแกรมเริ่ม Format และ ติดตั้ง FreeBSD

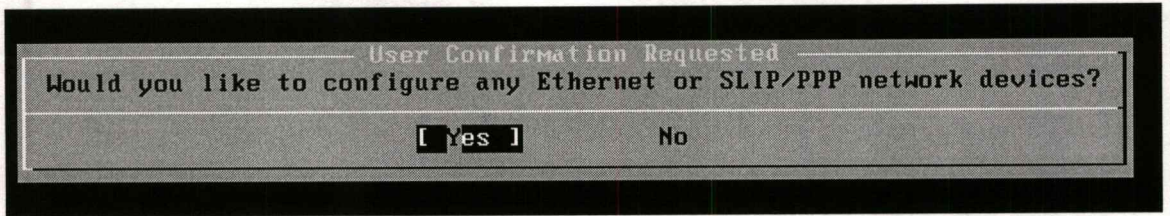
19. เมื่อโปรแกรมได้ติดตั้ง Source ต่าง ๆ ลงในโครงสร้างต่าง ๆ เรียบร้อยแล้ว ก็จะแสดงหน้าจอแสดงความยินดีให้เราทราบว่า ได้ทำการติดตั้งระบบปฏิบัติการ FreeBSD ให้เราเรียบร้อยแล้ว ดังรูปที่ 19



รูปที่ 19 หน้าจอแสดงความยินดีเมื่อติดตั้งเสร็จ

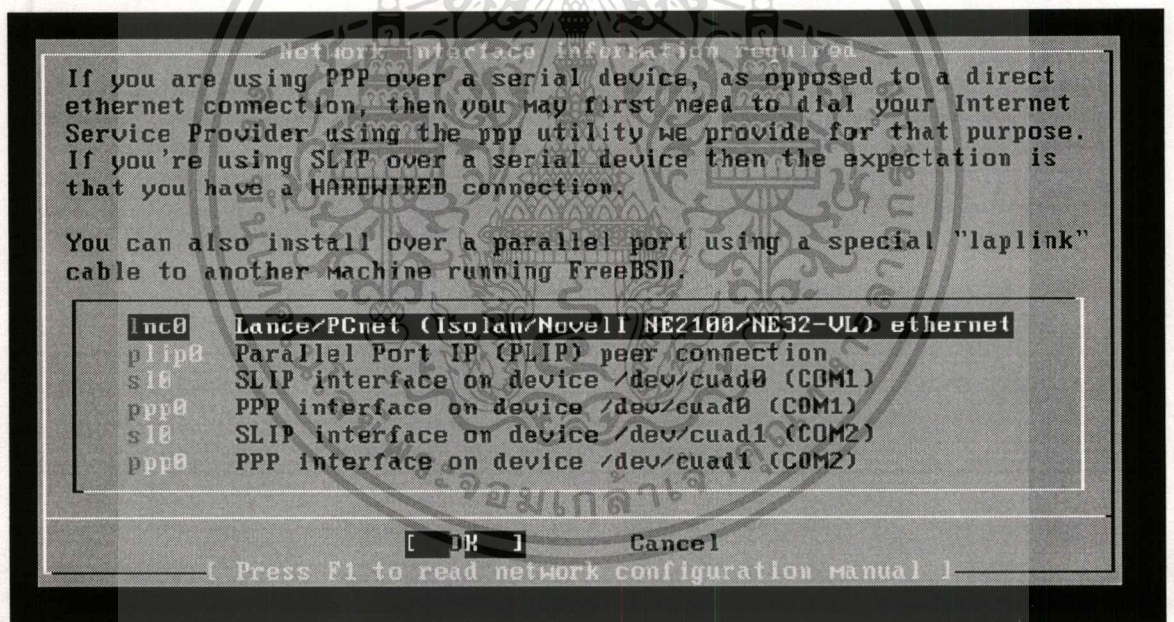
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20. โปรแกรมจะให้เรากำหนดค่าของเครือข่าย หรือที่เราเรียกว่า Network Devices ในที่นี้เราจะต้องกำหนดค่าให้กับอุปกรณ์เครือข่ายของเราโดยการตอบ Yes ดังรูปที่ 20



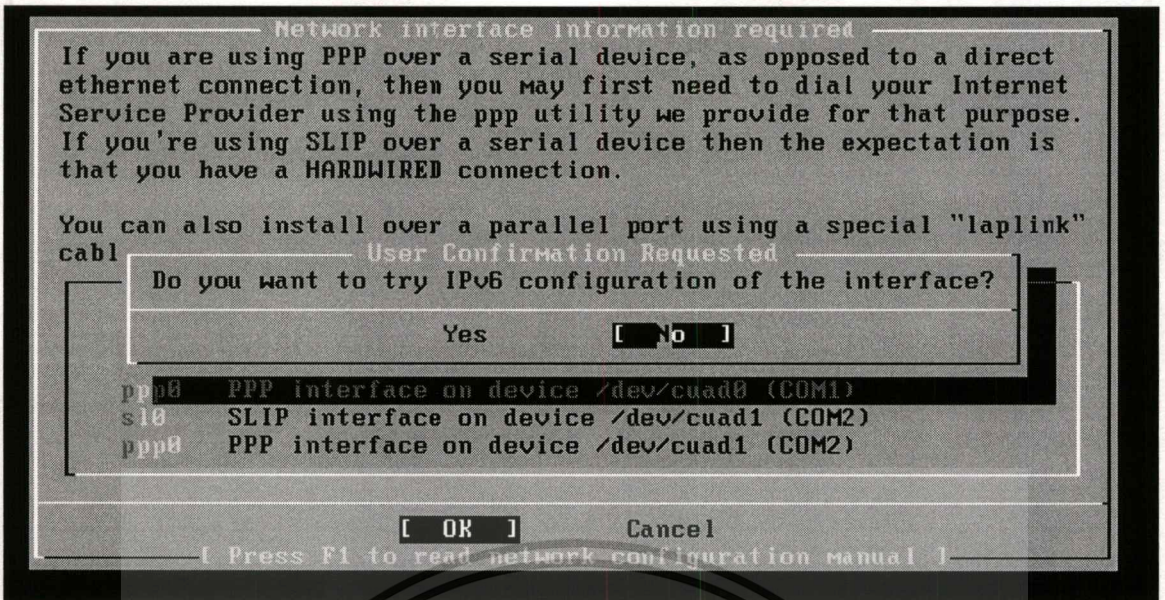
รูปที่ 20 กำหนดค่าเครือข่าย Network Devices

21. เลือกอุปกรณ์เครือข่าย Network Devices ในที่นี้เราจะเลือก lnc0 (เป็น LAN Card ที่เราใช้ในการติดตั้งนี้ แต่ในความเป็นจริง เราอาจจะพบเป็นชื่ออุปกรณ์อื่น ๆ ก็ได้ขึ้นอยู่กับยี่ห้อเช่น rl0, fxp0) กดปุ่ม Enter เพื่อทำงานต่อไป ดังรูปที่ 21



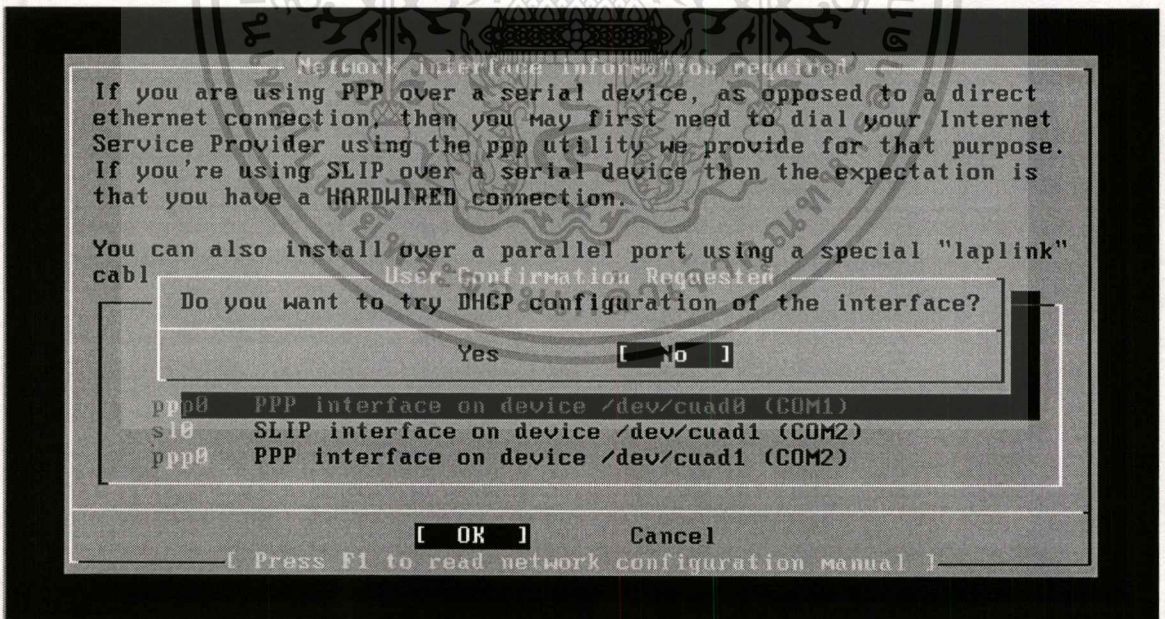
รูปที่ 21 เลือกอุปกรณ์ Network Devices

22. เราต้องการใช้งาน IPv6 (Internet Address Version 6 คือ มาตรฐานใหม่ของ IP address ที่มีขนาด 128 Bits) ในที่นี้เราจะตอบ No เนื่องจากยังไม่ได้ใช้ IPv6 ในการทำงาน ดังรูปที่ 22



รูปที่ 22 ไม่เลือก IPv6 Interface

23. ต้องการปรับแต่งหรือใช้บริการ DHCP หรือไม่ ในที่นี้ตอบ No เพื่อเราจะทำการกำหนดค่า IP address ให้กับอุปกรณ์เครือข่ายของเราเอง ดังรูปที่ 23



รูปที่ 23 ไม่ได้ปรับแต่งหรือใช้บริการ DHCP

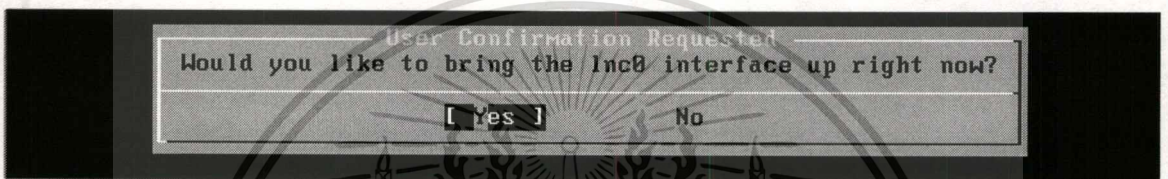
24. กำหนดค่าให้กับอุปกรณ์เครือข่าย จะต้องกำหนดให้สอดคล้องกับเครือข่ายที่ใช้งานอยู่สามารถ

อธิบายความหมายของค่าต่าง ๆ ได้ดังนี้

เอกสารฉบับเอกสารที่สงวนไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

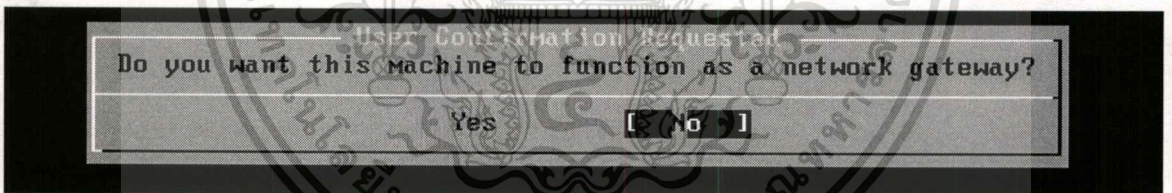
- Host : ชื่อเซิร์ฟเวอร์
- Domain : ชื่อ Domain ของเครื่องเซิร์ฟเวอร์
- IPv4 Gateway: หมายเลข IP Address ของ Gateway ที่ออกสู่ Internet
- Name Server : หมายเลข IP Address ของ DNS Server ขององค์กร หากไม่มีก็ให้กำหนด เป็นค่าของ DNS ของผู้ให้บริการหรือ ISP (Internet Service Providers)
- IPv4 address : หมายเลข IP Address ของเครื่องเซิร์ฟเวอร์
- Netmask : หมายเลข Netmask ของเครื่องเซิร์ฟเวอร์

25. ตอบ Yes เพื่อกำหนดให้อุปกรณ์เครือข่ายทำงาน ดังรูปที่ 25



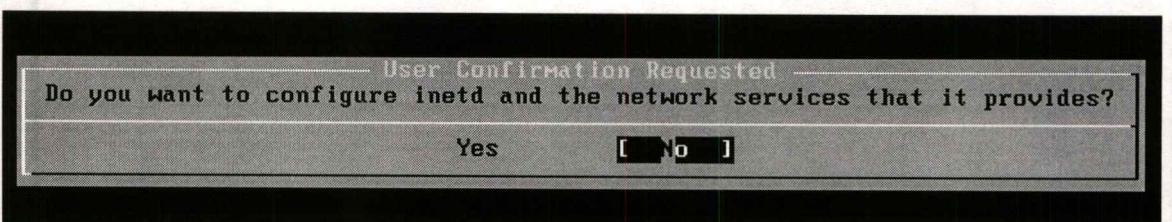
รูปที่ 25 กำหนดให้อุปกรณ์เครือข่ายทำงาน

26. ตอบ Yes เพื่อให้เครื่องเซิร์ฟเวอร์นี้เป็น Network Gateway ดังรูปที่ 26



รูปที่ 26 กำหนดค่า Network Gateway

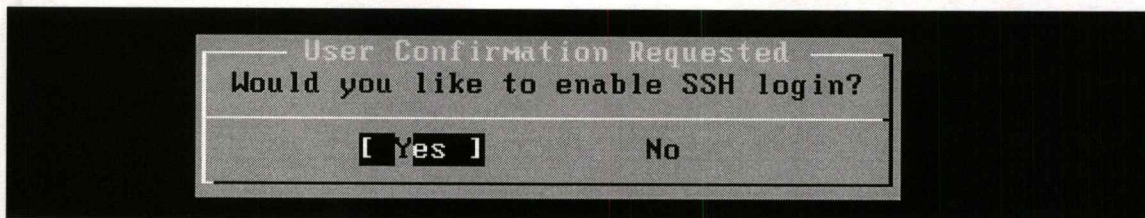
27. ตอบ No เพราะไม่ต้องการที่จะแก้ไขและปรับแต่งค่าของ inetd และ Network Service ดังรูปที่ 27



รูปที่ 27 การปรับแต่งค่า inetd (Internet Daemon)

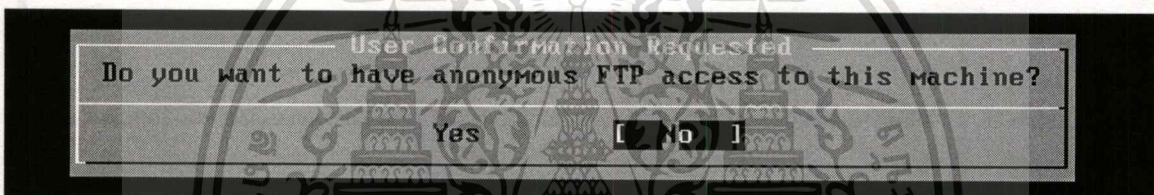
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

28. ตอบ Yes เพื่อเปิดบริการ SSH Login เนื่องจากเมื่อเราติดตั้งระบบปฏิบัติการ FreeBSD เรียบร้อยแล้ว เราสามารถที่จะ Remote เข้ามาได้จากภายนอก คล้าย ๆ กับบริการ Telnet แต่ SSH นั้นจะมีการเข้ารหัสข้อมูลจากเครื่องคอมพิวเตอร์ที่ Remote เข้ามายังเซิร์ฟเวอร์ ดังรูปที่ 28



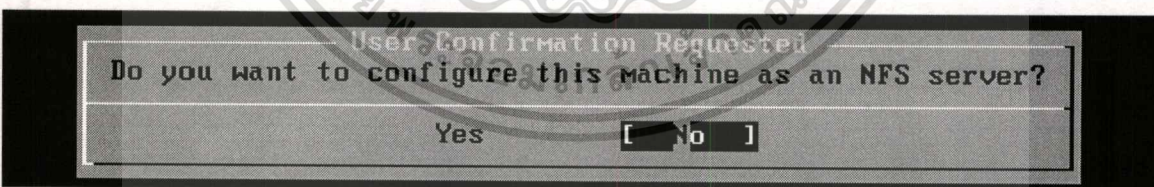
รูปที่ 28 เปิดบริการ SSH Login

29. ตอบ No เพื่อการปิดบริการ Anonymous FTP Service ดังรูปที่ 29



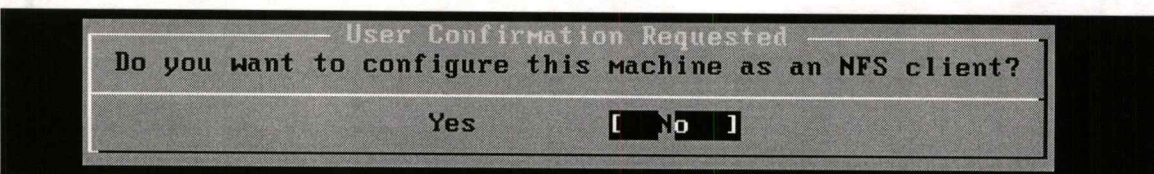
รูปที่ 29 การปิดบริการ Anonymous FTP Service

30. ตอบ No เพื่อการปิดบริการ NFS Server ดังรูปที่ 30



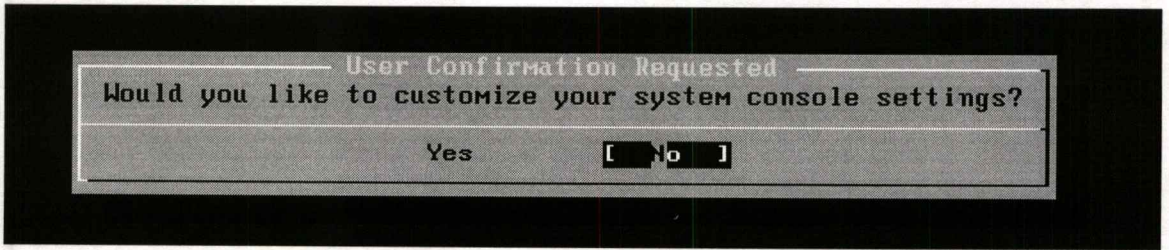
รูปที่ 30 ปิดบริการ NFS Server

31. ตอบ No เพื่อการปิดบริการ NFS Client ดังรูปที่ 31



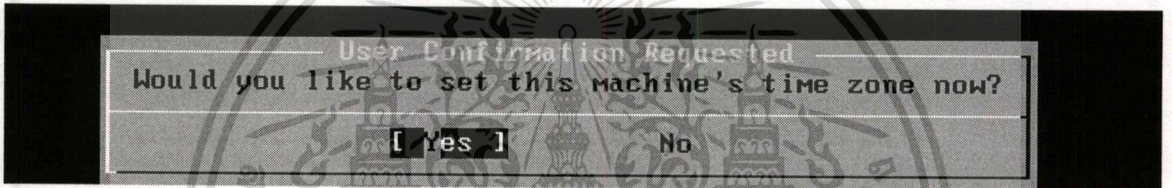
รูปที่ 31 ปิดบริการ NFS Client

32. ตอบ No เพื่อไม่ปรับแต่งค่าของ Console Setting ดังรูปที่ 32



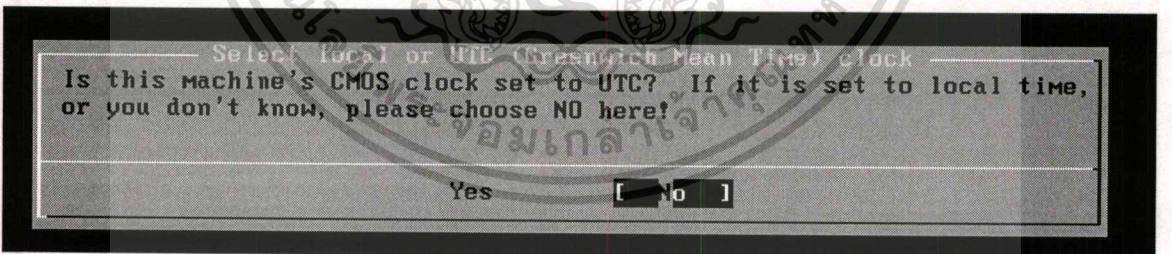
รูปที่ 32 ไม่ปรับแต่งค่าของ Console Setting

33. ระบบจะถามว่าต้องการปรับแต่งค่าของ Time Zone หรือไม่ ในที่นี้ให้ตอบ Yes เพื่อปรับแต่งค่าของ Time Zone ดังรูปที่ 33

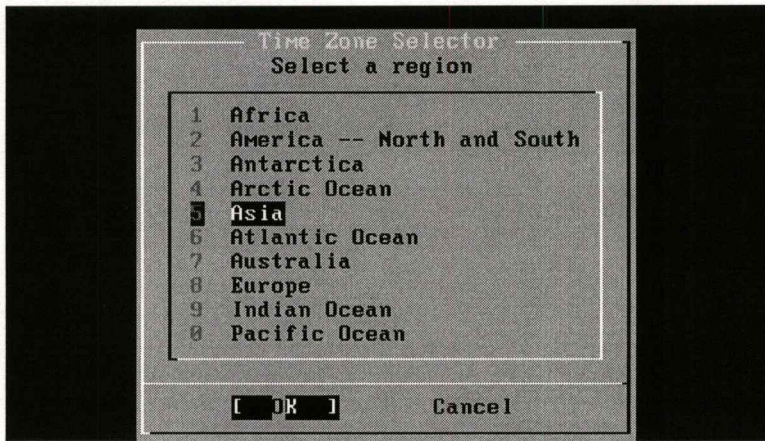


รูปที่ 33 ปรับแต่งค่าของ Time Zone

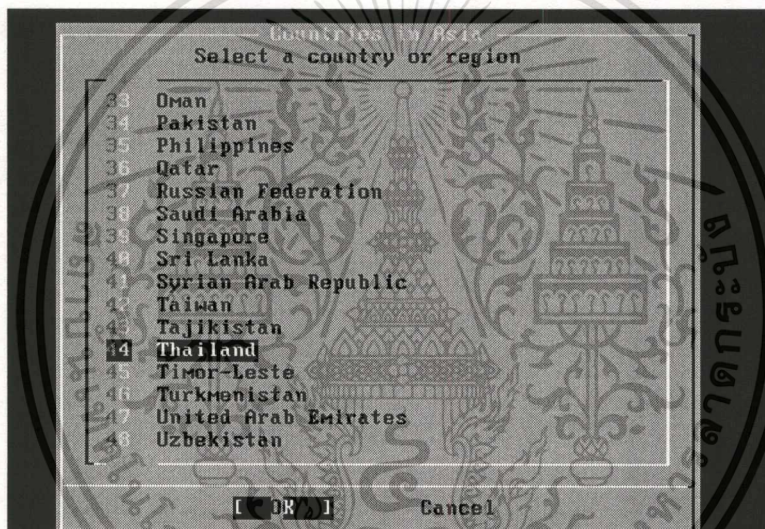
34. ตอบ No เพื่อจะได้ปรากฏตัวช่วยในการเลือก Time Zone ดังรูปที่ 34.1, 34.2, 34.3 และ 34.4



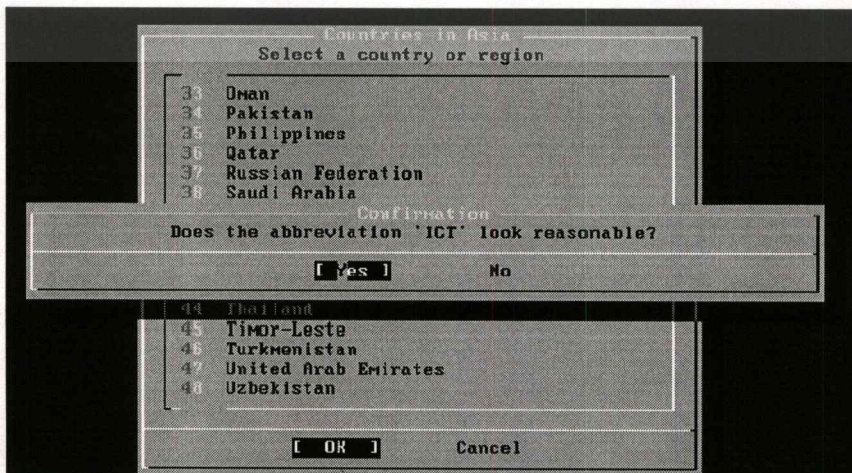
รูปที่ 34.1 ปรับเวลามาตรฐาน UTC



รูปที่ 34.2 ปรับเวลามาตรฐาน UTC

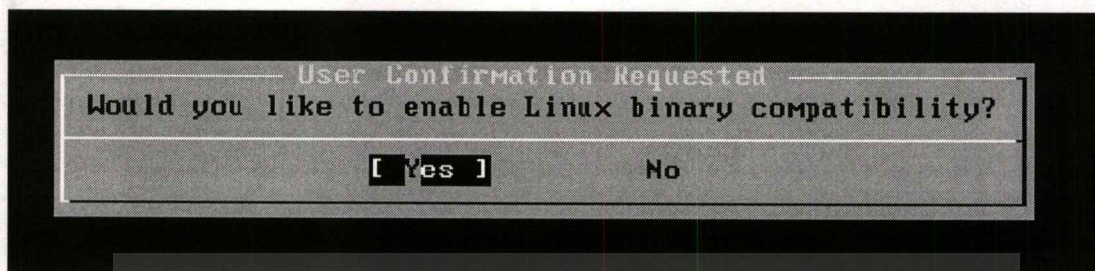


รูปที่ 34.3 ปรับเวลามาตรฐาน UTC



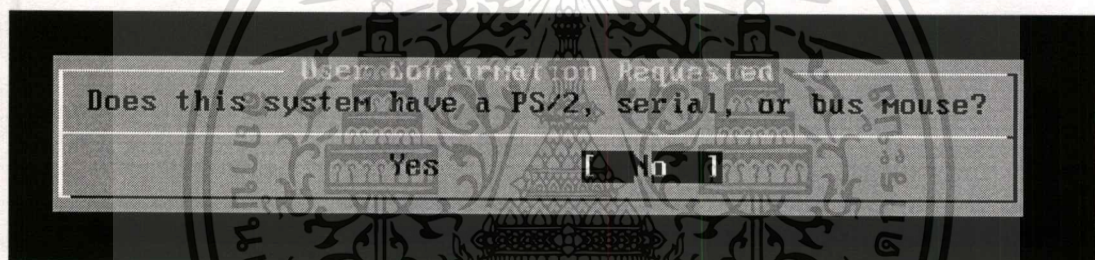
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคคลที่ทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 34.4 ปรับเวลามาตรฐาน UTC
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

35. ระบบถามว่าต้องการอนุญาตให้ โปรแกรมของระบบปฏิบัติการ Linux สามารถนำมาใช้กับ FreeBSD ได้หรือไม่ ให้เลือก Yes เพราะเราจะได้นำโปรแกรมของ Linux มาใช้งานร่วมกับ FreeBSD ได้ ดังรูปที่ 35



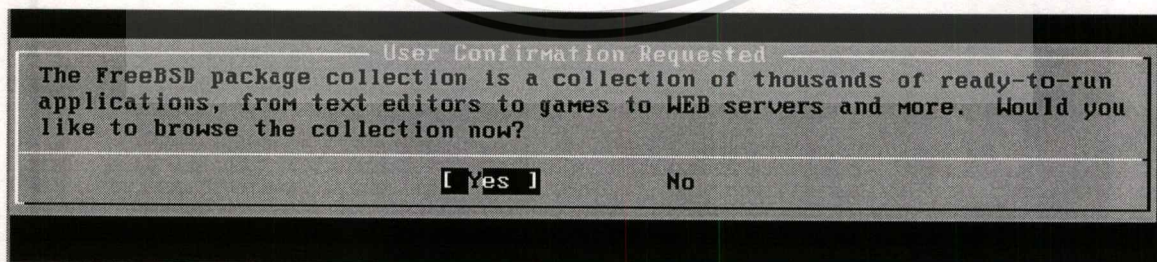
รูปที่ 35 การยอมรับให้โปรแกรม Linux ทำงานบน FreeBSD

36. ตอบ No เนื่องจากการใช้งานเซิร์ฟเวอร์นี้ไม่ได้ใช้ Mouse ดังรูปที่ 36



รูปที่ 36 ไม่มีการใช้ Mouse ในระบบ

37. ตอบ No เพราะไม่มีการติดตั้ง FreeBSD Packages เพิ่มเติม ดังรูปที่ 37

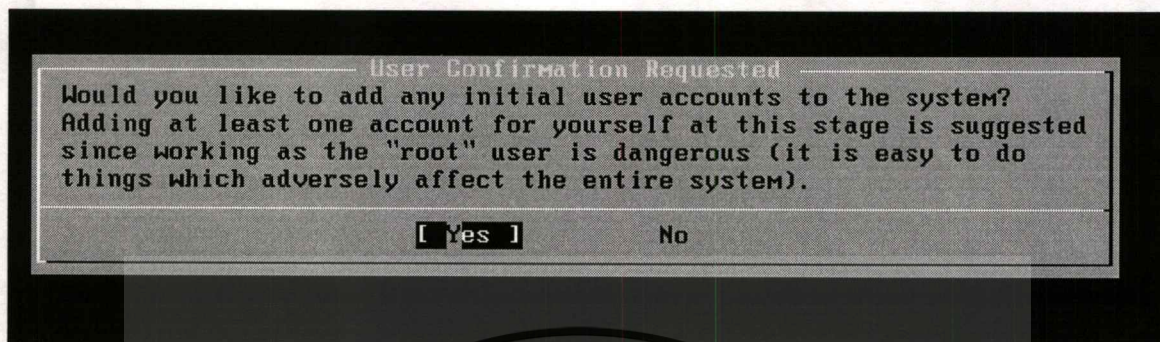


รูปที่ 37 การติดตั้ง FreeBSD Packages เพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

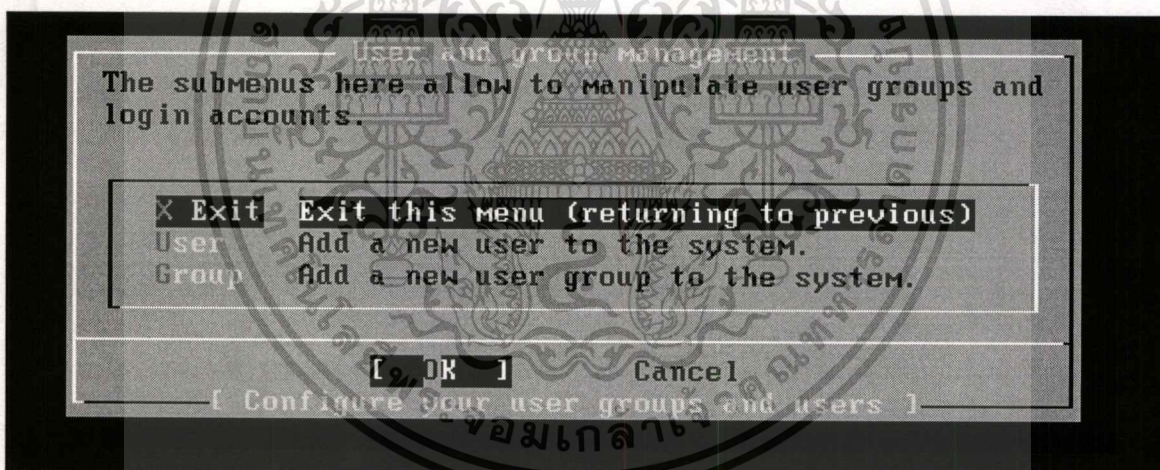
38. ระบบจะให้เราจัดการกับ Username และ Group รวมถึงการจัดการกับ root account (เป็น Account ที่เป็นผู้บริหารระบบ) ซึ่งเราสามารถที่จะจัดการได้จากเมนูนี้ ให้เลือกตอบ Yes ดังรูปที่

38.1



รูปที่ 38.1 เลือกจัดการกับ User และ Group

ให้เลือกคำสั่ง Exit เพื่อไม่เพิ่ม User Name และ Group ดังรูปที่ 38.2



รูปที่ 38.2 เลือก Exit เพื่อออกจากการทำงาน

จากนั้นจะปรากฏหน้าจอแนะนำให้เรากำหนดรหัสผ่านให้กับผู้ดูแลระบบที่เราเรียกว่า root โดยจะให้เราพิมพ์รหัสผ่านเข้าไปให้เหมือนกันสองครั้ง ซึ่งการพิมพ์รหัสผ่านนี้จะไม่มี การแสดงตัวอักษรใด ๆ บนหน้าจอภาพ ขอให้พิมพ์ให้ถูกต้องแล้วกดปุ่ม Enter ผ่านได้เลย ดังรูปที่ 38.3 และ

38.4



รูปที่ 38.3 คำแนะนำในการกรอกรหัสผ่านของ root



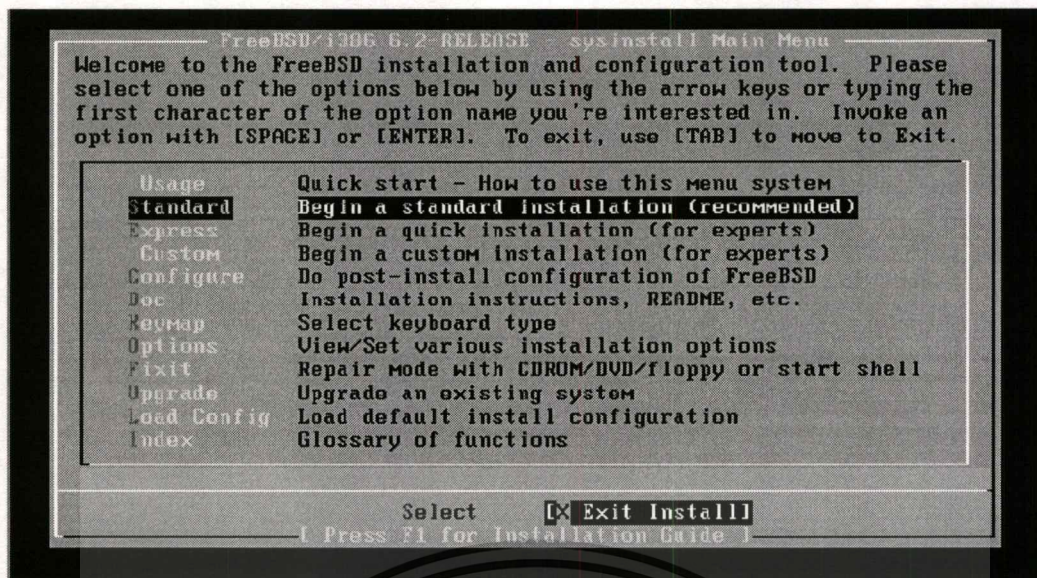
รูปที่ 38.4 กำหนดรหัสผ่านให้เหมือนกัน 2 ครั้ง

39. เลือก No เพราะไม่ต้องการกลับไปปรับแต่งค่าใด ๆ ดังรูปที่ 39



รูปที่ 39 ไม่เลือกปรับแต่งค่าต่าง ๆ ของระบบ

40. จากนั้นจะกลับมาที่เมนูหน้าแรกของการติดตั้งระบบปฏิบัติการ FreeBSD ให้เราเลือกปุ่ม Exit Install แล้ว Enter เพื่อเสร็จสิ้นการติดตั้ง



รูปที่ 41 เลือกเมนู Exit Install

41. โปรแกรมจะแนะนำให้เราเอาแผ่นติดตั้งระบบปฏิบัติการ FreeBSD ออกจาก CD-Rom Drive หรือ DVD Drive แล้วเลือกตัวเลือก Yes



รูปที่ 42 นำแผ่นติดตั้งออกจาก CD-Rom Drive

เมื่อนำแผ่นติดตั้งระบบปฏิบัติการ FreeBSD ออกจาก CD-Rom Drive แล้ว ระบบจะทำการรีบูทเครื่องเซิร์ฟเวอร์ จากนั้นก็เป็นการสิ้นสุดการติดตั้งระบบปฏิบัติการ FreeBSD หลังจากทีระบบปฏิบัติการ FreeBSD ได้ทำการบูทเสร็จเรียบร้อยแล้ว เราจะสามารถ Login เข้าระบบโดยใช้ root account แล้วใส่รหัสผ่านที่ได้กำหนดไว้ในขั้นตอนติดตั้งระบบ ดังรูปที่ 43

```

Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
6a:e3:e3:7d:e4:e9:66:3a:13:68:df:7b:d1:6b:c1:e9 root@bsd02.thaibsd.com
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
5a:f8:de:32:a8:0f:59:a1:7d:8f:7f:81:16:9a:5f:f3 root@bsd02.thaibsd.com
Generating public/private rsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_rsa_key.
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub.
The key fingerprint is:
76:4b:19:c5:4d:02:87:e6:d9:b8:6f:f9:e4:5f:70:5a root@bsd02.thaibsd.com
Starting sshd.
Starting cron.
Local package initialization:.
Additional TCP options:.
Starting background file system checks in 60 seconds.

Mon Feb 26 16:31:17 ICT 2007

FreeBSD/i386 (bsd02.thaibsd.com) (ttyv0)

login: root
Password:

```

รูปที่ 43 การ Login เข้าสู่ระบบ เมื่อเสร็จสิ้นกระบวนการติดตั้ง FreeBSD

การติดตั้ง Apache Web Server เวอร์ชัน 2.2

1. ให้เราเข้าไปยังไดเรกทอรี /usr/ports/www/apache22 แล้วทำการติดตั้ง Apache Web Server ด้วยคำสั่งดังนี้

```
# cd /usr/ports/www/apache22
```

```
# make install
```

2. โปรแกรมจะเริ่มติดตั้งแล้วจะปรากฏหน้าจอภาพ ถามเกี่ยวกับ Options เสริมของ gettext ในที่นี้ให้เรากดปุ่ม Tab มาที่ OK แล้วกดปุ่ม Enter เพื่อทำงานต่อไป

3. ให้รอนกว่าระบบปฏิบัติการ FreeBSD จะทำการคอมไพล์ (Compile) Apache จนเรียบร้อยก็จะปรากฏเครื่องหมาย Prompt (#) อีกครั้ง จากนั้นให้เราทำการปรับแต่ง Apache ดังนี้

```
# vi /usr/local/etc/apache22/httpd.conf
```

เพิ่มบรรทัดคำสั่งต่อไปนี้ลงไป

```
DirectoryIndex index.php index.html
```

```
AddDefaultCharset tis-620
```

```
AddType application/x-httpd-php .php
```

```
AddType application/x-httpd-php-source .phps
```

4. การที่เราจะสั่งให้ Apache Web Server ทำงานทุกครั้งที่มีการบูทเครื่องก็สามารถทำได้โดยการเพิ่มคำสั่ง apache22_enable="YES" ในไฟล์ /etc/rc.conf ดังนี้

```
# vi /etc/rc.conf
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งโปรแกรม Open SSL เวอร์ชัน 2.2

1. ทำการสร้าง Key สำหรับ SSL ด้วยคำสั่งด้านล่าง

```
# cd /root
# mkdir ssl
# cd ssl
# openssl genrsa -des3 -out server.key 1024
# openssl req -new -key server.key -out server.csr
# openssl x509 -req -days 365 -in server.csr \
? -signkey server.key \
? -out server.crt
```

2. ทำการ Copy Certificate ไปยังไคลเอนต์หรือปลายทาง

```
# cd /usr/local/etc/apache2/
# mkdir ssl.key
# mkdir ssl.crt
# cd ssl.crt
# cp /root/ssl/server.crt ./
# chmod 400 server.crt
# cd ..
# cd ssl.key
# cp /root/ssl/server.key ./
# cp server.key server.key.org
# openssl rsa -in server.key.org -out server.key
# chmod 400 server.crt
```

3. เพิ่มคำสั่งสำหรับ Start Server Apache ในไฟล์ /etc/rc.conf ดังนี้

```
apache2_enable="YES"
apache2_flags="-DSSL"
```

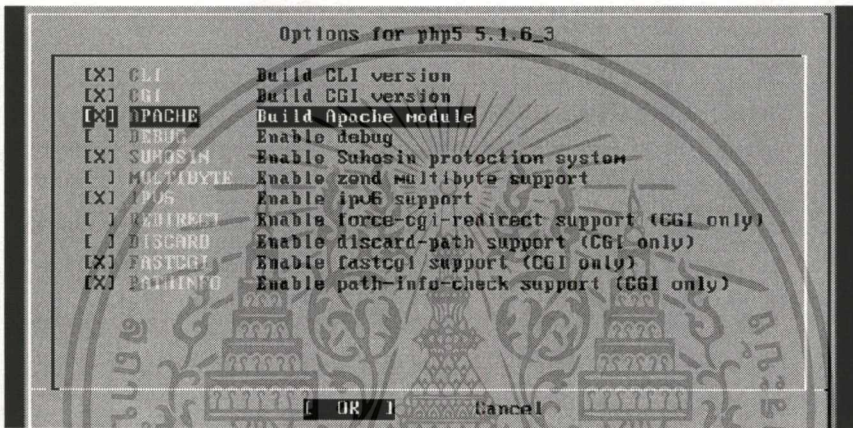
การติดตั้งภาษา PHP 5

1. เริ่มการติดตั้งโดยการเข้าไปยังไครเร็กทอรี `/usr/ports/lang/php5` ซึ่งจะเป็นภาษา PHP เวอร์ชัน 5 ซึ่งจำเป็นต่อการเขียนโปรแกรมภาษา PHP อย่างมากในปัจจุบัน เราสามารถติดตั้งได้โดยการใส่คำสั่งด้านล่าง

```
# cd /usr/ports/lang/php5
```

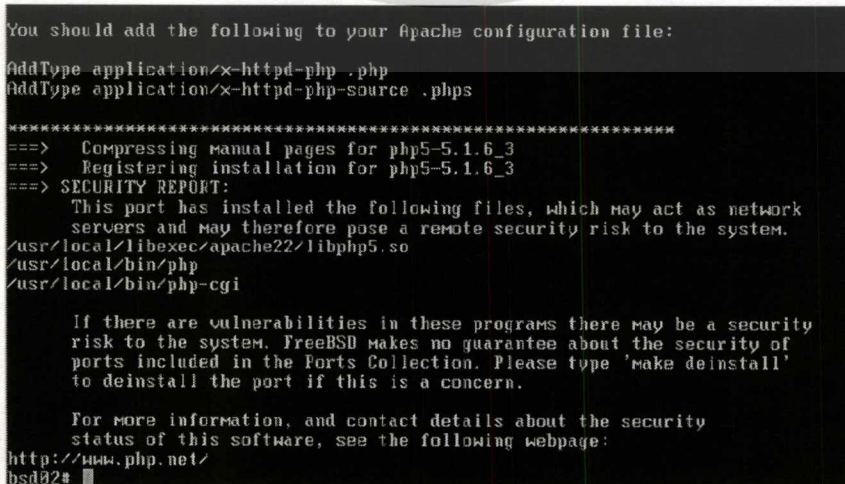
```
# make install
```

2. เลือก Options ของ Apache ซึ่งจะเป็น โมดูล (Modules) ที่จำเป็นต้องใช้งานเพื่อให้ Apache Web Server รับรู้ภาษา PHP ใหม่นี้ด้วย จากนั้นให้กดปุ่ม TAB มาที่ OK แล้ว Enter ดังรูปที่ 1



รูปที่ 1 การเลือก Apache Modules ของ PHP เพิ่มเติม

3. เมื่อโปรแกรมภาษา PHP ได้ทำการติดตั้งและคอมไพล์ (Compile) เสร็จเรียบร้อยแล้ว จะปรากฏหน้าจอ ดังรูปที่ 2 เพื่อให้ตรวจสอบว่าการมีการเพิ่มบรรทัดคำสั่ง `AddType` ทั้ง 2 บรรทัดในไฟล์ `httpd.conf`



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2 ภาษา PHP ถูกติดตั้งเรียบร้อยแล้ว

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หากเราต้องการทดสอบนั้นก็สามารถทำได้โดยการเขียนโปรแกรมภาษา PHP เพื่อทดสอบ ตามขั้นตอนนี้

```
# cd /usr/local/www/data
# vi test.php
```

เขียนคำสั่งเพิ่มหนึ่งบรรทัด ดังนี้

```
<? phpinfo(); ?>
```

หลังจากนั้นให้เปิดโปรแกรม Internet Explorer เรียก URL ตาม IP address ของเครื่อง เช่น <http://192.168.2.1/test.php> เป็นต้น หากทำได้ถูกต้องก็จะเขียนรายละเอียดของภาษา PHP ให้เราได้ทราบ เป็นการเสร็จสิ้นการติดตั้งภาษา PHP

การติดตั้ง Super User Do Package เวอร์ชัน 1.6.9

SUDO (Super User do) เป็น Package นี้ที่เราสามารถติดตั้งเพิ่มเติมเพื่อให้ user ทั่วไปมีสิทธิ์ในการใช้งานคำสั่งที่ root สามารถสั่งงานได้เช่น PF, Reboot, Shutdown และอื่นๆ เพราะเนื่องจากโดยปรกติแล้ว user ทั่วไปจะไม่มีสิทธิ์ในการสั่งงานคำสั่งเหล่านี้ โดยมีวิธีการติดตั้งดังนี้

1. เข้าสู่ไดเรกทอรี `cd /usr/ports/security/sudo` และใช้คำสั่ง `Make install`

```
# cd /usr/ports/security/sudo
# make install
# reboot
```

2. ปรับค่าต่างๆ โดยใช้คำสั่ง `visudo` และเพิ่มข้อมูล ด้านล่างดังนี้

```
www ALL=NOPASSWD
```

ภาคผนวก ข

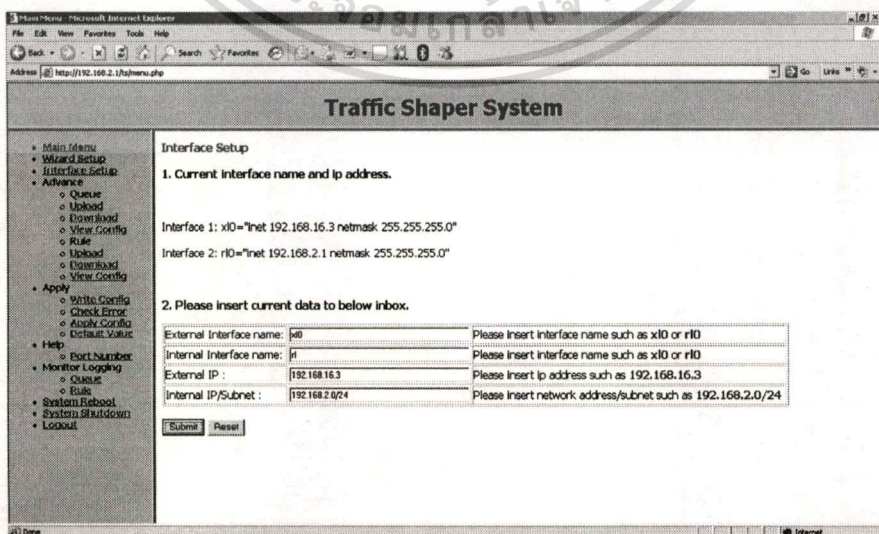
คู่มือการใช้งานระบบ

1. การเข้าใช้งานระบบ Web Admin

- เปิด Internet Explorer และพิมพ์ `https://<Server IP Address>/ts`
- ป้อน User Name: `<admin>` และ Password: `<admin>`
- เลือกเข้าสู่ระบบ



2. เมนูหลักของ Web Admin



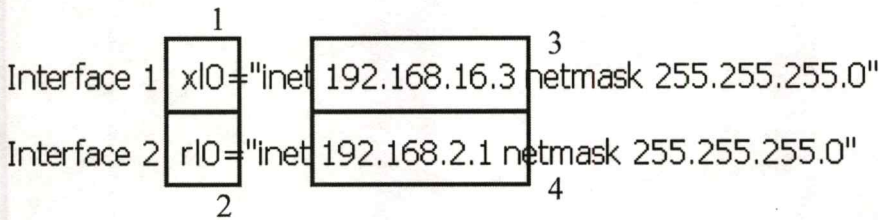
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Main Menu	แสดงหน้าเมนูหลัก
Wizard Setup	เลือกการติดตั้งสำหรับการใช้งานแบบง่ายและรวดเร็ว
Interface Setup	เลือกการกำหนดค่าอุปกรณ์เครือข่าย (NIC)
Queue	เลือกการกำหนดค่า Queue
Upload	กำหนดค่า Queue ส่วน Upload
Download	กำหนดค่า Queue ส่วน Download
View Config	แสดงการกำหนดค่า Queue
Rule	เลือกการกำหนดค่า Rule
Upload	กำหนดค่า Rule ส่วน Upload
Download	กำหนดค่า Rule ส่วน Download
View Config	แสดงการกำหนดค่า Rule
Apply	กำหนดการทำงานของระบบ
Write Config	บันทึกการกำหนดค่าให้กับระบบ
Check Error	ตรวจสอบค่าผิดพลาดของไฟล์ระบบ
Apply Config	กำหนดการทำงานให้ระบบ
Network Object	แสดงรายละเอียดของบริการ
Address	แสดงหมายเลขไอพี
Port Number	แสดงหมายเลขพอร์ต
Monitor	แสดงการทำงาน
Queue	แสดงการทำงานของ Queue
Rule	แสดงการทำงานของ Rule
System Reboot	รีบูทระบบใหม่
System Shutdown	ปิดการทำงาน
Logout	ออกจากระบบ Web Admin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Wizard Setup

- กำหนดค่าต่างๆ ของ Interface ลงใน Inbox



Wizard Setup

Step 1: Interface Setup

Interface 1: Interface 1: x10="inet 192.168.16.3 netmask 255.255.255.0"

Interface 2: Interface 2: r10="inet 192.168.2.1 netmask 255.255.255.0"

Please insert current data to below inbox.

External Interface name:	1	Please insert interface name such as x10 or r10
Internal Interface name:	2	Please insert interface name such as x10 or r10
External IP :	3	Please insert ip address such as 192.168.16.3
Internal IP/Subnet :	4	Please insert network address/subnet such as 192.168.2.0/24

- กำหนดค่า Upload และ Download Bandwidth สูงสุดลงใน Inbox

Bandwidth = 1024/512 Kbps < Download=1024Kbps/Upload=512Kbps>

Step 2: Bandwidth Setup

Upload Speed:	Kbps
Download Speed:	Kbps

- การกำหนดค่า Policy ให้กับระบบ จะแบ่งเป็น 3 ระดับดังนี้

First	ความสำคัญระดับสูงสุด
Second	ความสำคัญระดับกลาง
Third	ความสำคัญระดับต่ำ

Step 3: Traffic Policy Setup**Upload Policy**

Priority	From : <any "or" ip address>	To : <any "or" ip address>
First:	<input type="text"/>	<input type="text"/>
Second:	<input type="text"/>	<input type="text"/>
Third:	<input type="text"/>	<input type="text"/>

Download Policy

Priority	From : <any "or" ip address>	To : <any "or" ip address>
First:	<input type="text"/>	<input type="text"/>
Second:	<input type="text"/>	<input type="text"/>
Third:	<input type="text"/>	<input type="text"/>

- การกำหนดค่า Policy ให้กับระบบ

From	กำหนดค่า IP Address หรือกลุ่มของเครือข่ายต้นทาง
To	กำหนดค่า IP Address หรือกลุ่มของเครือข่ายปลายทาง

4. Advance Setup**4.1 PRIQ Queue Setup****Enable Download Queue Management**

1. Insert Queue Quantity And Select Queue Type.

Queue Quantity : Queue Type :

2. Insert Bandwidth (Download)

Bandwidth : Kbps

3. Type Queue name and % of bandwidth or priority value.

PRIQ config:

queue name 1: Priority value : <0-15> (default)

queue name 2: Priority value : <0-15>

4.2 CBQ Queue Setup

Enable Download Queue Management

1. Insert Queue Quantity And Select Queue Type.

Queue Quantity : Queue Type :

2. Insert Bandwidth (Download)

Bandwidth : kbps

3. Type Queue name and % of bandwidth or priority value.

CBQ config:

queue name 1: Bandwidth % (borrow) (default)
queue name 2: Bandwidth % (borrow)

Row Config	กำหนดจำนวน Queue ที่ต้องการ
Queue Type	เลือกชนิดของ Queue ระหว่าง CBQ หรือ PRIQ
Bandwidth	กำหนดค่า Bandwidth
Queue Name	กำหนดชื่อของ Queue
Bandwidth 'or' Priority	กำหนดปริมาณ Bandwidth หรือระดับความสำคัญ

5. Rule Setup

Rule Upload Management

From :	<input type="text"/>	any "or" ip address
To :	<input type="text"/>	any "or" ip address
Protocol	<input type="text" value="None"/>	blank "or" select protocol
Port Number :	<input type="text"/>	blank "or" insert port number
Queue :	<input type="text" value="voice"/>	Select queue name to assign

Rule data
pass out on \$ext_if from asdf to asdf keep state queue 11
pass out on \$ext_if proto tcp from any to 10.10.10.254 port 8080 keep state queue voice

From	กำหนดค่า IP Address หรือกลุ่มของเครือข่ายต้นทาง
To	กำหนดค่า IP Address หรือกลุ่มของเครือข่ายปลายทาง
Protocol	เลือกชนิดของ Protocol ระหว่าง TCP หรือ UDP
Port Number	เลือกหมายเลขพอร์ต
Queue	เลือก Queue ที่ต้องการสำหรับ Rule นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายสรารุช สิงห์กุล
วเดือนปีเกิด	30 มกราคม 2521
สถานที่เกิด	จังหวัดยโสธร
วุฒิการศึกษาระดับปริญญาตรี	วศบ. (วิศวกรรมไฟฟ้า) สาขา วิศวกรรมคอมพิวเตอร์
สถานที่สำเร็จการศึกษา	คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร
ปีที่สำเร็จการศึกษา	2545

