

ระบบจัดการแบนด์วิดท์สำหรับเครือข่าย LAN ผ่านเว็บ
LAN Bandwidth Management via Web

โดย

จักรพงษ์ ศรีประสม

รหัสประจำตัว 461713003

อาจารย์ที่ปรึกษา

อ.ปุลลฑรา สร้อยสน

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงงาน 2

ภาคเรียนที่ 2 ปีการศึกษา 2549

หลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ

มหาวิทยาลัยธุรกิจบัณฑิตย์

ชื่อหัวข้อ	ระบบจัดการแบนด์วิดท์สำหรับเครือข่าย LAN ผ่านเว็บ
ชื่อนักศึกษา	นายจักรพงษ์ ศรีประสม
อาจารย์ที่ปรึกษา	อาจารย์ปณิตตรา สร้อยสน
ระดับการศึกษา	วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ
ปีการศึกษา	2549

บทคัดย่อ

โครงการนี้เป็นโครงการเพื่อการควบคุมการใช้งาน **Bandwidth** ของเครือข่าย LAN เพื่อให้ผู้ใช้ที่ใช้งานอินเทอร์เน็ตภายในเครือข่ายเดียวกันได้รับช่องสัญญาณ **Bandwidth** อย่างเสมอภาคกัน โดยที่ผู้ใช้งานแต่ละรายจะมี **User Account** ของแต่ละคนที่ใช้ในการเข้าระบบและเพื่อตรวจสอบสิทธิ ในการใช้งานต่าง ๆ โดยโครงการนี้ได้ใช้ระบบปฏิบัติการ **FreeBSD** เป็น **Server** และใช้โปรแกรม **Squid** เป็นโปรแกรมในการบริหารจัดการเรื่องการแบ่ง **Bandwidth** และ **Block** เว็บไซต์ โดยใช้ **Web Application** ที่เขียนด้วยภาษา **PHP** เป็นตัวควบคุมการสั่งงาน

กิตติกรรมประกาศ

โครงการ ระบบจัดการแบนด์วิดท์สำหรับเครือข่าย LAN ผ่านเว็บ ที่ได้จัดทำขึ้นมาสำเร็จ ล่วงได้เนื่องจากได้รับการสนับสนุนร่วมมือ และการให้คำแนะนำปรึกษาในแนวทางต่างๆ จากหลายบุคคล ทำให้โครงการนี้สำเร็จลุล่วงได้ตามเป้าหมายที่ได้วางไว้ ผู้จัดทำใคร่ขอขอบคุณบุคคลต่างๆ ดังนี้

- อ.ปณตธา สร้อยสน สำหรับการเป็นที่ปรึกษาโครงการ และให้คำแนะนำต่างๆ ในการทำโครงการนี้ จนสำเร็จ
- อ.วรภัทร ไพรีเกรง สำหรับท่านอาจารย์วรภัทร นี้คงเป็นที่ประทับใจของ ข้าพเจ้าไปตลอด ท่านเป็นทั้ง อาจารย์ที่ปรึกษา และเป็นตัวอย่างของการดำเนินชีวิต ท่านได้ให้ความรู้ ให้คำแนะนำ และเทคนิคต่างๆ เสมอมาตลอดระยะเวลา 3ปีที่ข้าพเจ้าได้อยู่ในคณะเทคโนโลยีสารสนเทศแห่งนี้
- อ.วิลาวัลย์ อินทร์ชำนาญ สำหรับการสอนให้ข้าพเจ้ารู้จักการวางแผนในการทำโครงการ และท่านยังได้ให้คำแนะนำที่ดีเสมอมาสำหรับการออกแบบ UML
- อ.ผกาพรรณ ว่องวุฒิพรชัย สำหรับคำแนะนำต่างๆ ทั้งที่เกี่ยวข้องกับโครงการและไม่เกี่ยวข้องกับโครงการ ข้อเสนอแนะต่างๆ ที่ท่านได้ให้แนวทางในการทำโครงการ
- คุณพ่อ และ คุณแม่ สำหรับท่านทั้ง 2 นี้คงจะกล่าวคำว่าขอบคุณออกมาเป็นคำพูดได้ไม่รู้จบ แต่สำหรับโครงการนี้ท่านได้ให้ กำลังใจ และเป็นแรงบันดาลใจให้ข้าพเจ้าต่อสู้กับอุปสรรคต่างๆ เพื่อที่จะได้นำความรู้ความสามารถ และประสบการณ์ นำไปดำเนินชีวิตต่อ
- และบุคคลอื่นๆ ที่ไม่ได้กล่าวมา ณ ที่นี้

จึงใคร่ขอขอบพระคุณบุคคลดังกล่าวข้างต้นมา ณ โอกาสนี้

นายจักรพงษ์ ศรีประสม

ผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
กิตติกรรมประกาศ.....	II
สารบัญ.....	III
สารบัญรูป.....	V
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของการพัฒนาระบบ.....	1
1.3 ขั้นตอนการดำเนินงาน และระยะเวลาการดำเนินงาน.....	2
1.4 ขอบเขตของระบบงาน.....	3
1.5 เครื่องมือที่ใช้ในการพัฒนา.....	3
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 FreeBSD.....	5
2.2 PHP.....	7
2.3 MYSQL.....	9
2.4 Squid.....	10
2.5 Client-serve.....	11
2.6 File Server.....	12
2.7 Database Server.....	13
2.8 Transaction Server.....	13
2.9 Groupware Server.....	13
2.10 Web Server.....	14

สารบัญ (ต่อ)

หน้า

211 Firewall.....	14
212 Network Address Translation(NAT).....	15
213 Telnet หรือ SSH.....	16
214 WWW (World Wide Web)	17
3 การออกแบบโครงสร้างระบบ.....	18
31 การสำรวจความต้องการของระบบ.....	18
32 แนวทางการพัฒนาระบบ.....	18
33 การออกแบบการทำงานของระบบ	19
4 ผลการพัฒนาระบบ	22
41 ขั้นตอนการทำงานของเครื่องลูกข่าย	22
42 ขั้นตอนการทำงานในส่วนของผู้ดูแลระบบ.....	24
43 การออกแบบ Interface ในส่วนของผู้ดูแลระบบ	25
5 บทสรุปและข้อเสนอแนะ.....	34
51 บทสรุป.....	34
52 ข้อเสนอแนะ	34
53 การประยุกต์ใช้งาน	35
ภาคผนวก.....	37
ภาคผนวก ก. คู่มือสำหรับผู้ใช้งาน.....	38
ภาคผนวก ข. คู่มือสำหรับผู้ดูแลระบบ.....	42
บรรณานุกรม.....	50
ประวัติผู้แต่ง.....	51

สารบัญรูป

รูปที่	หน้า
1.1	ระยะเวลาการดำเนินงาน..... 2
3.1	USE CASE DIAGRAM..... 20
3.2	STATE DIAGRAM..... 21
4.1	แสดงวิธีการกรอก IP และ Port ของ Server..... 23
4.2	แสดงหน้าต่างสำหรับกรอก User name และ Password..... 23
4.3	หน้าจอเข้าสู่ระบบของผู้ดูแลระบบ 25
4.4	หน้าจอของเมนูหลักของระบบ..... 25
4.5	หน้าจอหลัก ของเมนู Block เว็บไซต์ 26
4.6	หน้าจอแสดง URL ที่ต้องการ Block group 56 Kbps 26
4.7	หน้าจอแสดง URL ที่ต้องการ Block สำหรับผู้ใช้งานทั้งหมดในระบบ 27
4.8	หน้าจอสำหรับลบบรายชื่อผู้ใช้งาน 27
4.9	หน้าจอสำหรับเพิ่มรายชื่อผู้ใช้งาน..... 28
4.10	หน้าจอแสดงผู้ที่กำลังใช้งาน 28
4.11	หน้าจอหลักของ รายชื่อผู้ใช้งานทั้งหมดในระบบ..... 29
4.12	หน้าจอแสดงรายชื่อของผู้ใช้งานในกลุ่ม 56 Kbps..... 29
4.13	หน้าจอสำหรับเลือกวันที่ เก็บสถิติต่างๆ..... 30
4.14	หน้าจอแสดงรายละเอียดหลัก ของการเก็บสถิติ..... 30
4.15	หน้าจอแสดงรายชื่อเว็บไซต์ที่ถูกใช้งานมากที่สุด..... 31
4.16	หน้าจอแสดงรายชื่อเว็บไซต์และผู้ที่ใช้ใช้งาน..... 31
4.17	หน้าจอแสดงรายการที่เคย Download..... 32
4.18	หน้าจอแสดงรายชื่อของเว็บไซต์ที่มีปัญหาไม่สามารถเข้าได้..... 32
4.19	หน้าจอแสดงรายการทั้งหมดของ ผู้ใช้งาน 33
ก. 1	หน้าจอแสดงการ Login สำหรับผู้ใช้ระบบ 38
ก. 2	หน้าจอแสดงตัวอย่างการ Login..... 39
ก. 3	หน้าจอแสดงเมื่อเข้าระบบได้สำเร็จ 39

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก. 4 หน้าจอแสดงการ Block เว็บไซต์สำหรับผู้ใช้งานกลุ่ม 56 Kbps	42
ก. 5 หน้าจอแสดงการ Block เว็บไซต์สำหรับผู้ใช้งานทั้งหมดในระบบ	43
ข. 1 หน้าจอ Login เข้าสู่ระบบ	45
ข. 2 หน้าจอหลักของการจัดการระบบ	45
ข. 3 Add User	46
ข. 4 แสดง User ทั้งหมด.....	47
ข. 5 แสดงรายชื่อ User กลุ่ม 512 Kbps	47
ข. 6 หน้าจอการลบ User	48
ข. 7 หน้าจอสำหรับเลือกกลุ่มที่ต้องการ Block	49
ข. 8 แสดง URL ที่ Block	49
ข. 9 หน้าจอแสดง Online User	50
ข. 10 หน้าจอแสดงวันที่เก็บสถิติ.....	51
ข. 11 หน้าจอแสดงรายละเอียดของสถิติ.....	51

บทที่ 1

บทนำ

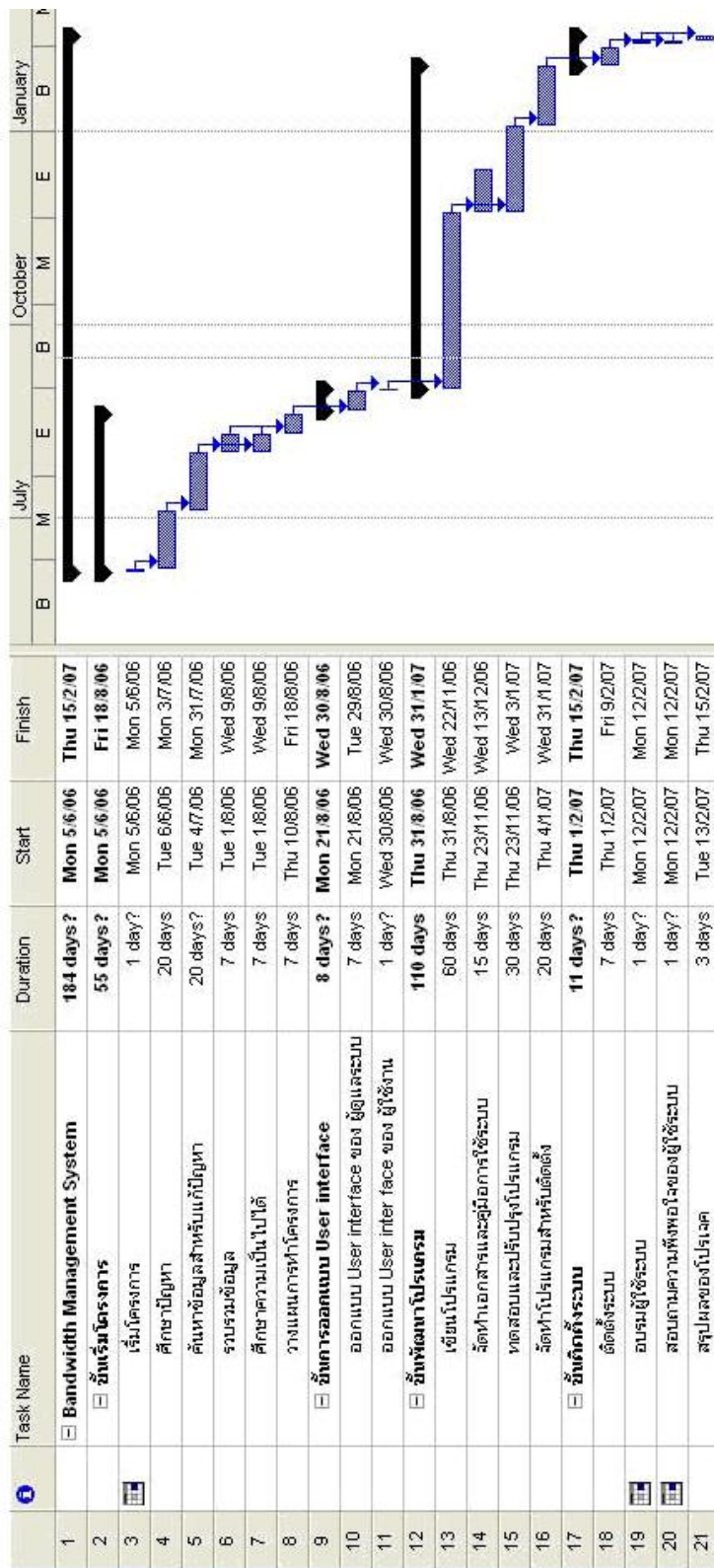
1.1 ความเป็นมาของปัญหา

เนื่องจากปัจจุบัน อินเทอร์เน็ต ได้เข้ามามีบทบาทในการดำเนินชีวิตมากขึ้น สังเกตได้จากการเพิ่มของอัตราการใช้อินเทอร์เน็ต ความเร็วสูง เช่น **ADSL (Asymmetric Digital Subscriber Line)** มากขึ้นจากเดิม นอกจากนี้ตามสถานที่ต่าง ๆ ยังมีบริการเช่าสัญญาณ อินเทอร์เน็ต ไวไฟให้ใช้เช่าบริการอย่างสะดวก รวมถึงหอพัก อพาร์ทเมนต์ ก็จะมี อินเทอร์เน็ต ไวไฟให้บริการ ดังนั้นหากหอพักใดมีผู้ใช้งาน อินเทอร์เน็ต พร้อมกันหลายคนก็จะทำให้ความเร็วของ อินเทอร์เน็ต นั้นช้าลง เนื่องจาก แบนด์วิดท์ ถูกใช้งานมาก และหากมีผู้ใช้บางรายทำการ **Download** ไฟล์ ก็จะทำให้แบนด์วิดท์ ที่ใช้งานนั้นสูงกว่าบุคคลอื่น ทำให้บุคคลอื่นใช้งาน อินเทอร์เน็ต ได้ช้าลงไปจากเดิม ดังนั้นผู้จัดทำโครงการจึงได้คิดวิธีที่จะทำให้ทุกคนที่ใช้งาน อินเทอร์เน็ต ได้ความเร็วในการใช้งานเท่าๆกัน

1.2 วัตถุประสงค์ของการพัฒนาระบบ

เพื่อแบ่งแบนด์วิดท์ให้กับผู้ใช้ทุกคนได้ใช้ช่องสัญญาณของแบนด์วิดท์ ได้เท่ากัน โดยใช้เซิร์ฟเวอร์ ที่มีระบบปฏิบัติการ **FreeBSD** และใช้โปรแกรม **Squid** เป็นตัวควบคุมและจัดการแบนด์วิดท์ และ **Block** เว็บไซต์ที่ไม่เหมาะสม อีกทั้งเพื่อความสะดวกและสะดวกของผู้ควบคุมจึงได้ใช้ **Web Application** ที่เขียนด้วยภาษา **PHP (Personal Home Page Tool)** เป็นตัวควบคุมการสั่งงานของโปรแกรม **Squid** (ที่ใช้งานยาก) ให้สามารถใช้งานได้อย่างง่ายดาย

1.3 ขั้นตอนการดำเนินงาน และระยะเวลาการดำเนินงาน



รูปที่ 1.1 ระยะเวลาการดำเนินงาน

1.4 ขอบเขตของระบบงาน

1. สร้างระบบเพื่อแบ่งแบนด์วิดท์ให้กับผู้ใช้งาน
2. **Block** เว็บไซต์บางแห่งเพื่อไม่ให้ผู้ใช้เข้าใช้งานได้
3. มีระบบตรวจสอบ **username** และ **password** ก่อนการเข้าใช้งาน อินเทอร์เน็ต

1.5 เครื่องมือที่ใช้ในการพัฒนา

1. ระบบปฏิบัติการ **FreeBSD**
2. เครื่องมือที่ใช้ในส่วนของการพัฒนาเว็บ **PHP, squid, MySQL, Apache, SARG**
3. **PuTTY** เป็นเครื่องมือในการ **Remote login** เข้าไปใช้งานเครื่องเซิร์ฟเวอร์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. สร้างความสะดวกให้กับผู้ดูแลระบบ
2. เพื่อให้ผู้ใช้งาน สามารถใช้งาน อินเทอร์เน็ต ได้ด้วยความเร็วคงที่
3. ป้องกันเพื่อไม่ให้บุคคลที่ไม่ได้รับ **Username** และ **password** เข้าใช้งาน อินเทอร์เน็ต ได้
4. มีระบบจัดการความเร็วในการใช้งาน อินเทอร์เน็ต ตามที่ผู้ใช้ต้องการได้เช่น **56k, 128k, 256k, 512k**
5. **block** เว็บไซต์ที่ไม่พึงประสงค์ได้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

21 FreeBSD

21.1 คุณลักษณะและจุดเด่นของ FreeBSD

FreeBSD (ฟรีบีเอสดี) คือซอฟต์แวร์เสรีซึ่งเป็นระบบปฏิบัติการที่เหมือนยูนิกซ์(Unix-like) สืบทอดมาจาก **AT&T UNIX** ผ่านทางสายของ **Berkeley Software Distribution (BSD)** คือ **386BSD** และ **44BSD**. **FreeBSD** รองรับการทำงานบน **CPU** ตระกูลหลัก ๆ หลายตระกูลด้วยกันนอกจากตระกูล **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 โลโก้ดั้งเดิมและตัวสัญลักษณ์ที่ใช้ (**Mascot**) ของโครงการ **FreeBSD** คือตัวดีมอนสีแดง (**Red demon**) โดยมี มาร์แชล เคิร์ก แมคคูลิก (**Marshall Kirk McKusick**) เป็นเจ้าของลิขสิทธิ์

การพัฒนา **FreeBSD** เป็นแบบเบ็ดเสร็จทั้งระบบปฏิบัติการ กล่าวคือทั้งเคอร์เนล, ยูเซอร์แลนด์ยูทิลิตี้เช่น **shell**, และ **ดีไวซ์ไดรเวอร์** อยู่ในทรินของระบบควบคุมเวอร์ชันของซอร์สโค้ด (**CVS**) เดียวกัน ซึ่งแตกต่างจาก **Linux** ที่มีการพัฒนาเฉพาะส่วนของเคอร์เนลโดยบุคคลกลุ่มหนึ่ง ส่วนของยูเซอร์แลนด์ยูทิลิตี้พัฒนาโดยกลุ่มอื่นๆ เช่น กลุ่มในโครงการของ **GNU** และนำมารวมเข้าด้วยกันกับโปรแกรมประยุกต์กลายเป็นดิสทริบิวท์ชันซึ่งนำมาเผยแพร่ให้ผู้ใช้ได้ใช้กัน

FreeBSD ได้รับการยกย่องว่าเป็นระบบปฏิบัติการที่มีชื่อเสียงทางด้านเสถียรภาพและความอึด (แต่ไม่อืด) จึงเป็นระบบปฏิบัติการที่ใช้รันเว็บเซิร์ฟเวอร์อย่างแพร่หลายข้อยืนยันนี้ดูได้จากรายงานอ็อปไทม์ (เวลาจากการรีบูตครั้งล่าสุด). ในรายการ **50** อันดับของเว็บเซิร์ฟเวอร์ที่มีอ็อปไทม์นานที่สุดก็มี **FreeBSD** และ **BSD/OS** ปรากฏอยู่เป็นจำนวนมาก สิ่งนี้เป็นตัวบ่งบอกถึงความมั่นคงของ **FreeBSD** ว่าตลอดเวลาการปฏิบัติงานอันยาวนานนี้นอกจากจะไม่มีการแครชแล้ว ยังไม่มีจำเป็นต้องมีการอัปเดตเคอร์เนลแต่อย่างใด (หลังจากอัปเดตเคอร์เนลจำเป็นต้องรีบูต)

21.2 ความเป็นมาของFreeBSD

Jordan Hubbard ได้เขียนถึงความเป็นมาของ FreeBSD ไว้ที่หน้าเว็บไซต์ http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/history.html เอาไว้ว่าโครงการ FreeBSD เกิดขึ้นประมาณต้นปี 1993 ส่วนหนึ่งสืบเนื่องมาจากพัฒนาการของ "Unofficial 386BSD Patchkit" จากผู้ร่วมงาน 3 คน คือ Nate Williams, Rod Grimes และ Jordan Hubbard เป้าหมายเดิมของการสร้างโปรแกรม 386BSD เพื่อแก้ปัญหามากมายที่ patchkit แก้ไม่ได้ ชื่อโครงการนี้แต่เดิมคือ "386BSD 0.5" หรือ "386BSD Interim" นั่นเอง 386BSD เดิมเป็นระบบปฏิบัติการของ Bill Jolitz's แต่โปรแกรมนี้มีจุดอ่อนหลายอย่างและไม่ได้พัฒนาต่อเนื่อง กล่าวคือ patchkit มีขนาดใหญ่มากขึ้นทุกวัน และทำงานได้ช้า (patchkit swell) พวกเขาจึงช่วยแก้กันปัญหา 386BSD แต่แผนปฏิบัติการครั้งนี้ต้องหยุดชะงักเพราะ Bill Jolitz เขาตัดสินใจถอนตัวไม่สนับสนุนดำเนินการแก้ปัญหาในเรื่องนี้ โดยผู้ร่วมงานทั้ง 3 คนก็ยังไม่ทราบเหตุการณ์ตัดสินใจของ Bill Jolitz จนถึงบัดนี้

ในส่วนของผู้ร่วมงานเห็นว่าถ้าดำเนินการแก้ปัญหาสำเร็จจะมีประโยชน์มากแม้ว่า Bill จะไม่สนับสนุนพวกเขาก็ตามเลยดำเนินการต่อไป และเปลี่ยนชื่อโครงการว่า "FreeBSD" โดยความร่วมมือของ David Greenman การดำเนินการเบื้องต้นโดยการกำหนดเป้าชัดเจน หลังจากนั้นโครงการจึงเป็นรูปร่างเป็นจริงขึ้นมา Jordan Hubbard กล่าวว่าเขาคิดต่อกับ Walnut Creek CDROM เพื่อมุ่งหวังปรับปรุงการปฏิบัติงานของ FreeBSD แก้ข้อบกพร่องในการใช้ร่วมกับอินเทอร์เน็ต Walnut Creek CDROM ไม่เพียงแต่สนับสนุนความคิด FreeBSD บน CD เท่านั้น แต่ดำเนินการสนับสนุนโครงการด้วยการให้สามารถใช้งานกับอินเทอร์เน็ตได้เร็วมากขึ้น ถ้าหากว่าไม่มี Walnut Creek CDROMs มาช่วย FreeBSD จะถูกลืมไป ชื่อเสียงการยอมรับน่าจะไม่มีจนถึงทุกวันนี้ การจัดทำหน่วย CDROM (and general net-wide) ครั้งแรกคือ FreeBSD 1.0 ตั้งแต่ธันวาคม 1993 ชุด CDROM นี้พัฒนามาจาก 4.3BSD-Lite ("Net/2") ซึ่งเป็นที่ซื้อที่มาจาก U.C. Berkeley มีส่วนประกอบได้มาจาก 386BSD Free Software Foundation จัดได้ว่าค่อนข้างประสบความสำเร็จทีเดียว ต่อมาอีกชุดหนึ่งคือ FreeBSD 1.1 ออกมาในเดือนพฤษภาคม 1994 ในช่วงเวลาเดียวกันเริ่มเกิดปัญหาขึ้นเมื่อ Novell and U.C. Berkeley มีการดำเนินการฟ้องร้องเรื่องสถานภาพตามกฎหมาย Berkeley Net/2 tape เงื่อนงำโดยสถานภาพในแง่กฎหมายตาม U.C. Berkeley's ซึ่งได้รับสัมปทานส่วนใหญ่ Net/2 และเป็นทรัพย์สินของ Novell ซึ่งได้รับสิทธิบัตรมาจาก AT&T สิ่งที่ Berkeley ได้รับมาเป็นการตอบแทนก็คือ Novell's ซึ่ง 4.4BSD-Lite ปล่อกออกจำหน่าย เมื่อมีการจำหน่ายออกจริง จึงได้มีการประกาศว่า ลักษณะที่ไม่สะดวกใช้ มีการแนะนำให้ใช้ Net/2 แทนได้ รวมทั้ง FreeBSD งานทั้งโครงการดำเนินมาถึงปลายเดือนกรกฎาคม 1994 จึงมีการหยุดจำหน่ายผลิตภัณฑ์ที่

พัฒนามาจาก **Net/2** ภายใต้งื่อนไขนั้น มีข้อตกลงให้จำหน่ายเป็นครั้งสุดท้ายก่อนสิ้นสุดการจำหน่ายตามสัญญา, ตัวที่ออกมาจำหน่ายหลังสุดคือ **FreeBSD 1.1.5.1**. ก่อนที่จะไม่ให้จำหน่ายต่อไปอีก **FreeBSD** ได้ดำเนินการพัฒนาใหม่จากเดิมที่ใหม่สุดแต่ยังไม่ สมบูรณ์ซึ่งเรียกว่า **44BSD-Lite bits** มันปรากฏเป็รยวขึ้นเพราะ **Berkeley's CSRG** ตัดส่วนที่เทะอะทะออกไปเพราะสะดวกในการปฏิบัติงานได้สะดวกขึ้น (**due to various legal requirements**) และโดยข้อเท็จจริงที่ว่า **Intel port of 44** ไม่สมบูรณ์แบบเอาเลย โครงการนี้ต้องใช้เวลาจนถึงพฤศจิกายน 1994 ส่วนที่ไม่สมบูรณ์นี้ ต่อจากนั้นจึงส่งตัวนี้ส่ง **FreeBSD 2.0** ทางอินเทอร์เน็ตและ **CDROM** (ในปลายเดือน ธันวาคม) แม้ว่าจะมีข้อบกพร่องอยู่บ้าง (**FreeBSD 2.0.5**) ในการ **release** ครั้งนี้นับว่าประสบความสำเร็จ ตามมาด้วยการ **release** ต่ออีกครั้งหนึ่ง (**FreeBSD 2.0.5**) ในเดือนมิถุนายน 1995. **FreeBSD 2.1.5** เปิดตัวในเดือน สิงหาคม 1996 และเป็นที่ยอมรับสมควรในบรรดา **ISP** และ วงการค้า เป็นผลให้มีการออกชุดใหม่ **FreeBSD 2.1.7.1** ออกในเดือน กุมภาพันธ์ 1997 และก่อให้เกิดการพัฒนา **21-STABLE** ขึ้น ซึ่งเป็นในช่วงการดูแลให้โปรแกรมสามารถใช้งานได้ดี เน้นแก้ไขในด้านการรักษาระบบความปลอดภัย และการแก้ปัญหาข้อผิดพลาดของโปรแกรม (**RELENG_2_1_0**). **FreeBSD 2.2** แยกย่อยมาจากการพัฒนา **mainline ("-CURRENT")** ในเดือนพฤศจิกายน 1996 ให้เป็น **RELENG_2_2 branch** และมีการออกชุดที่สมบูรณ์ (**2.2.1**) ในเดือนเมษายน 1997 ที่มีการ **releases** ต่อมารวมกับ **2.2 branch** มีในช่วง **the summer** และ **fall of '97** อันสุดท้าย (**2.2.8**) ในเดือน พฤศจิกายน 1998 การเปิดตัวออกมาเป็นทางการของ **FreeBSD 3.0** ในเดือนตุลาคม 1998 และถือเป็นการสิ้นสุดการการพัฒนาในรุ่น **2.2 branch**

โครงการใหญ่มีการเริ่มต้นอีกครั้งหนึ่งในวันที่ 20 เดือนมกราคม 1999 นำไปสู่โครงการย่อย คือ **4.0-CURRENT** และ **3.X-STABLE branches** จาก **3.X-STABLE** คือ 3.1 เปิดตัวเมื่อวันที่ 15 กุมภาพันธ์ 1999, 3.2 เมื่อ 15 พฤษภาคม 1999, 3.3 เมื่อ 16 กันยายน 1999, 3.4 เมื่อ 20 พฤศจิกายน 1999, และ 3.5 เมื่อ 24 มิถุนายน 2000, หลังจากนั้นไม่กี่วันมีการแก้ไขจาก **3.X-STABLE 3.5.1** เพื่อทำงานรักษาความปลอดภัยร่วมกับ **Kerberos**. และนี่ถือว่าการจัดเป็นการเปิดตัวครั้งสุดท้ายของ **3.X-STABLE 3.X branch** มีการออกโครงการใหม่อีกโครงการหนึ่งเมื่อ 13 มีนาคม 2000 ก่อให้เกิด **4.X-STABLE branch** มีการ **releases 4.X-STABLE branch** ออกมาหลายชุด เริ่มจากเดือนมีนาคม 2000 และออกชุดสุดท้ายคือ **4.11-RELEASE** เมื่อเดือนมกราคม 2005 ชุดที่รอคอยกันมาคือ **5.0-RELEASE** ออกมาเมื่อวันที่ 19 เดือนมกราคม 2003 เป็นงานที่พัฒนาต่อเนื่องกันมาเกือบ 3 ปีการเปิดตัวครั้งนี้จัดว่าเป็นการเริ่ม **FreeBSD** ยุคใหม่ที่การออกแบบมาเพื่อใช้กับ **advance multiprocessor** และ สนับสนุน **UltraSPARC** และ **ia64 platforms** ตามมาด้วยการออกรุ่น 5.1 ในเดือนมิถุนายน 2003 ตามมาด้วย **5.2.1-RELEASE** ซึ่งออกมาเมื่อเดือนกุมภาพันธ์ 2004 ในเดือน

สิงหาคม 2004 5.3-RELEASE และกำลังตามมาด้วย 5.4 ซึ่งเป็นการเริ่มต้น releases ชุด 5-STABLE branch จนถึงปัจจุบัน โครงการที่พัฒนาต่อเนื่องยาวนานกันมาใน 6X-CURRENT (trunk) branch ซึ่งเป็นโครงการใหญ่ และ SNAPSHOT releases เป็น 6X ในแบบ CDROM และทางอินเทอร์เน็ต ซึ่งติดตามได้จาก snapshot server ปัจจุบันนี้ เวอร์ชันล่าสุดขณะที่ผู้ทำโครงการนี้ได้จัดทำโครงการนี้คือ FreeBSD 6.1-STABLE

2.2 PHP

PHP ย่อมาจากคำว่า “Personal Home Page Tool” เป็นการเขียนโค้ดโปรแกรมที่เก็บและทำงานฝั่งเซิร์ฟเวอร์ (Server-Side Script) ซึ่งรูปแบบในการเขียนคำสั่งการทำงานนั้นจะมีลักษณะคล้ายกับภาษา Perl หรือภาษา C และสามารถที่จะใช้ร่วมกับภาษา HTML ได้อย่างมีประสิทธิภาพ ทำให้รูปแบบเว็บเพจของเรามีลูกเล่นมากขึ้น

2.2.1 ประวัติ PHP

แรกเริ่มนั้น PHP ได้ถูกคิดค้นโดยนาย Rasmus Lerdorf ตั้งแต่ปี ค.ศ. 1994 แต่เป็นเวอร์ชันที่ไม่เป็นทางการเพราะใช้ทดสอบกับเว็บเพจของเขาเองเท่านั้น โดยนาย Rasmus Lerdorf ได้ใช้ PHP ในการเก็บข้อมูลสถิติผู้เข้าชมเว็บของตนเอง

ต่อมา PHP เวอร์ชันแรก นั้นได้ถูกพัฒนาและเผยแพร่ให้กับอื่นที่ต้องการใช้ศึกษาในปี ค.ศ. 1995 ซึ่งถูกเรียกว่า “Personal Home Page Tool” ซึ่งเป็นที่มาของคำว่า PHP นั้นเอง ซึ่งในระแวกเวลานั้น PHP ยังไม่มีความสามารถอะไรที่โดดเด่นมากมาย จนกระทั่งเมื่อประมาณกลางปี 1995 Rasmus ได้คิดค้นและพัฒนาให้ PHP/FI หรือ PHP เวอร์ชัน 2 ให้มีความสามารถจัดเก็บเกี่ยวกับแบบฟอร์ม ข้อมูลที่ถูกสร้างมาจากภาษา HTML และสนับสนุนการติดต่อกับโปรแกรมจัดการระบบฐานข้อมูล mSQL จึงทำให้ PHP เริ่มถูกใช้มากขึ้นอย่างรวดเร็ว และเริ่มมีผู้สนับสนุนการใช้งาน PHP มากขึ้น โดยในปี ค.ศ. 1996 PHP ถูกนำไปใช้ประมาณ 15,000 เว็บไซต์ทั่วโลก และเพิ่มจำนวนขึ้นเรื่อยๆ เป็น 50,000 เว็บไซต์

นอกจากนี้ในราวกลางปี ค.ศ. 1997 PHP ได้มีการเปลี่ยนแปลงและถูกพัฒนาจากเจ้าของเดิมคือ Rasmus ซึ่งพัฒนาอยู่คนเดียวมาเป็นทีมงาน โดยมีนาย Zeev Suraski และ Andi Gutmans ทำการวิเคราะห์พื้นฐานของ PHP/FI และได้นำโค้ดมาพัฒนาใหม่เป็น PHP เวอร์ชัน 3 ซึ่งมีความสมบูรณ์มากขึ้น

ในกลางปี ค.ศ. 1999 PHP เวอร์ชัน 3 หรือ PHP3 สามารถทำงานกับ C2's StrongHold WebServer และ Red Hat Linux ได้

ต่อมาในปี ค.ศ. 2000 PHP เวอร์ชัน 4 พร้อม **Zend scripting engine** และความสามารถที่ทำงานกับเว็บเซิร์ฟเวอร์ยี่ห้ออื่นได้ นอกเหนือจากเว็บเซิร์ฟเวอร์ **Apache** ทำให้ **PHP4** มีความสมบูรณ์มากยิ่งขึ้น โดยในปลายปี 2000 นั้นมีเว็บไซต์ ที่ใช้ **PHP** รวมกันทั่วโลกมากถึง **3,300,000** เว็บไซต์ และ **PHP4** ก็กลายเป็น **PHP** เวอร์ชันที่มีผู้นิยมใช้มากที่สุด โดยในปัจจุบัน (06/2005) **PHP4** ได้ถูกได้ถูกพัฒนามาจนถึงเวอร์ชัน **4.3.10**

PHP เวอร์ชันต่อไปคือ **PHP5** เริ่มต้นออกเวอร์ชันทดสอบ (**Beta 1**) ตั้งแต่กลางปี ค.ศ. 2003 และพัฒนาเป็นตัวเต็มเมื่อประมาณกลางปี ค.ศ. 2004 ปัจจุบัน (06/2005) **PHP5** ได้พัฒนามาถึงเวอร์ชัน **5.0.4** แล้ว

2.2.2 ลักษณะเด่นของ PHP

ความเร็วในการพัฒนาโปรแกรม เพราะว่า **PHP** เป็นสรีปต์แบบ **Embedded** คือสามารถแทรกแทรกกร่วมกับภาษา **HTML** ได้อย่างอิสระและหากพัฒนาโค้ดไว้ในรูปแบบของ **Class** ที่เขียนขึ้นเพียงครั้งเดียวแล้วเรียกใช้ได้ตลอด ทำให้สะดวกรวดเร็วในการพัฒนาโปรแกรมต่าง ๆ

2.2.3 PHP เป็นโค้ดแบบเปิดเผย (Open Source)

คำว่า (**Open Source**) ว่าเป็นไปแล้วก็มีความหมายเหมือนกับของฟรีนั่นเอง เนื่องจาก **PHP** มีกลุ่มของผู้ใช้งานอยู่เป็นจำนวนมากทั่วโลก และมีเว็บไซต์อยู่เป็นจำนวนมากที่เป็นแหล่งรวบรวมซอสโค้ดโปรแกรม หรือจะเป็นบทความต่าง ๆ ทำให้ผู้ใช้มีมือใหม่ ๆ หรือผู้ที่ต้องการศึกษา สามารถค้นหาซอสโค้ดมาเป็นแนวทางในการพัฒนาได้ง่ายขึ้น

2.2.4 การบริหารหน่วยความจำ (Memory Usage)

มีการใช้งานหน่วยความจำที่ดี กล่าวคือ **PHP** จะไม่เรียกใช้หน่วยความจำตลอดเวลา ทำให้เซิร์ฟเวอร์ไม่จำเป็นต้องมีทรัพยากรมากนัก

2.2.5 อิสระต่อระบบปฏิบัติการ

เว็บแอปพลิเคชันที่ถูกสร้างขึ้นมาสามารถที่จะรันได้หลายระบบปฏิบัติการ ไม่ว่าจะเป็น **UNIX, FreeBSD, Linux** หรือ **Windows** เป็นต้น

2.3 MYSQL

MySQL เป็นฐานข้อมูลแบบ **open source** ที่ได้รับความนิยมในการใช้งานสูงสุด โปรแกรมหนึ่งบนเครื่องให้บริการ มีความสามารถในการจัดการกับฐานข้อมูลด้วยภาษา **SQL (Structures Query Language)** อย่างมีประสิทธิภาพ มีความรวดเร็วในการทำงาน รองรับการทำงานจากผู้ใช้หลายๆ คนและหลายๆ งานได้ในขณะเดียวกัน

MySQL ถูกพัฒนาขึ้นโดย **MySQL AB** โดยมีลิขสิทธิ์การใช้งาน **2** แบบ นั่นคือ ผู้ดูแลระบบสามารถใช้งานซอฟต์แวร์ **MySQL** ได้โดยไม่มีค่าใช้จ่ายใดๆ ภายใต้ลิขสิทธิ์ของ **GNU General Public License (<http://www.gnu.org/licenses/>)** หรืออาจเลือกใช้แบบที่มีลิขสิทธิ์ทางการค้าของ **MySQL AB** ซึ่งเป็นผู้ผลิตและพัฒนาซอฟต์แวร์โดยตรงก็ได้ หากไม่ต้องการเกี่ยวข้องกับข้อตกลงเรื่อง **GPL** รายละเอียดเพิ่มเติมเกี่ยวกับโปรแกรม **MySQL** สามารถหาข้อมูลได้จาก <http://www.mysql.com> คำอธิบายเพิ่มเติมเกี่ยวกับหน้าที่ ความสามารถและการทำงานของโปรแกรม **MySQL** มีดังต่อไปนี้

- **MySQL** ถือเป็นระบบจัดการฐานข้อมูล (**DataBase Management System (DBMS)**) ฐานข้อมูลมีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูล การที่จะเพิ่มเติม เข้าถึงหรือประมวลผลข้อมูลที่เก็บในฐานข้อมูลจำเป็นจะต้องอาศัยระบบจัดการฐานข้อมูล ซึ่งจะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลทั้งสำหรับการใช้งานเฉพาะ และรองรับการทำงานของแอปพลิเคชันอื่นๆ ที่ต้องการใช้งานข้อมูลในฐานข้อมูล เพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก **MySQL** ทำหน้าที่เป็นทั้งตัวฐานข้อมูล และระบบจัดการฐานข้อมูล
- **MySQL** เป็นระบบจัดการฐานข้อมูลแบบ **relational** ฐานข้อมูลแบบ **relational** จะทำการเก็บข้อมูลทั้งหมดในรูปแบบของตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์เพียงไฟล์เดียว ทำให้ทำงานได้รวดเร็วและมีความยืดหยุ่น นอกจากนั้น แต่ละตารางที่เก็บข้อมูลสามารถเชื่อมโยงเข้าหากันทำให้สามารถรวมหรือจัดกลุ่มข้อมูลได้ตามต้องการ โดยอาศัยภาษา **SQL** ที่เป็นส่วนหนึ่งของโปรแกรม **MySQL** ซึ่งเป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล

- **MySQL** แจกจ่ายให้ใช้งานแบบ **open source**

นั่นคือ ผู้ใช้งาน **MySQL** ทุกคนสามารถใช้งานและปรับแต่งการทำงานได้ตามต้องการ สามารถดาวน์โหลดโปรแกรม **MySQL** ได้จากอินเทอร์เน็ตและนำมาใช้งานโดยไม่มีค่าใช้จ่ายใดๆ

2.4 Squid

Squid โปรแกรม **Proxy Server** ที่มีประสิทธิภาพสูง มีคุณสมบัติที่น่าสนใจมากมาย สามารถช่วยให้การใช้แบนด์วิดท์ของเครือข่ายเป็นไปอย่างคุ้มค่า เรื่องราวของ **Squid** ยังมีสิ่งที่จะต้องศึกษาอีกมากมาย นับตั้งแต่ระบบเครือข่ายอินเทอร์เน็ต ได้เริ่มเข้ามามีบทบาทในการสื่อสาร เป็นเสมือนแหล่งข้อมูลจำนวนมหาศาล และช่องทางการดำเนินกิจกรรมต่าง ๆ มากมาย ความต้องการที่จะเข้าสู่อินเทอร์เน็ต และใช้บริการจากเว็บไซต์จึงเป็นสิ่งที้องค์กร และหน่วยงานต่าง ๆ มีความจำเป็นต้องจัดวางระบบเครือข่ายของตนเองให้สามารถบริการนี้แก่ผู้ใช้งานภายในองค์กรอย่างมีประสิทธิภาพ การใช้อินเทอร์เน็ตร่วมกันด้วยคอร์เน็ทชันเดียว ด้วยเทคนิคต่าง ๆ จึงถูกนำมาใช้ ไม่ว่าจะเป็นเทคนิคการ **NAT (Network Address Translation)** หรือเทคนิคการทำ **Proxy Service** ทั้งนี้ด้วยความต้องการที่จะลดค่าใช้จ่ายเกี่ยวกับค่าบริการอินเทอร์เน็ต และค่าใช้จ่ายในการเชื่อมต่อสัญญาณนั่นเอง รูปแบบการแชร์อินเทอร์เน็ตดังกล่าว สามารถกระทำได้หลายวิธี ทั้งนี้ขึ้นอยู่กับชนิดของระบบสื่อสารที่เลือกใช้ การลงทุนด้านอุปกรณ์ ลักษณะการใช้งาน และประสิทธิภาพที่ต้องการ เช่น หากเป็นองค์กรขนาดใหญ่ที่จำเป็นต้องใช้ **IP Address** จริง และมีการเชื่อมต่อแบบถาวร ก็น่าจะลงทุนเช่าสาย **Leased Line** และใช้เราเตอร์เป็นอุปกรณ์เชื่อมต่อ แต่ถ้าเป็นร้านบริการอินเทอร์เน็ตขนาดเล็กก็อาจจะใช้โมเด็มแบบ **Dial up** หรือบริการ **ADSL** และแชร์อินเทอร์เน็ตด้วยอุปกรณ์ประเภท **IP Sharer** ก็เพียงพอแล้ว อย่างไรก็ตามความเร็วของการใช้งานอินเทอร์เน็ตร่วมกันเช่นนี้จะขึ้นอยู่กับจำนวนผู้ใช้งาน และประเภทของบริการที่เรียกใช้เข้าสู่อินเทอร์เน็ต เนื่องจากแบนด์วิดท์ของช่องสัญญาณมีจำกัดนั่นเอง ยังมีจำนวนการใช้งานมากเท่าไรก็ยิ่งทำให้ความเร็วเฉลี่ยที่ผู้ใช้งานแต่ละคนได้รับก็ยิ่งลดลงเท่านั้น จึงจำเป็นต้องหาวิธีการที่จะประหยัดการใช้งานช่องทางสื่อสารให้มากที่สุด วิธีการหนึ่งที่จะช่วยให้งานบริการประเภท **Web Service** ได้ก็คือ การใช้งาน **Proxy Server** หมายถึงอุปกรณ์ในระบบเครือข่ายที่รับเอาการร้องขอใช้บริการ (**request**) จากเครื่องลูกข่าย ได้แก่ โปรแกรมเว็บเบราว์เซอร์ หรือ โปรแกรม **FTP Client** แล้วส่งผ่านการร้องขอนั้นไปยังเซิร์ฟเวอร์ปลายทางในเครือข่ายอินเทอร์เน็ต ดังนั้น **Proxy Server** จึงเปรียบเสมือนตัวแทนของเครื่องลูกข่ายที่อยู่ภายในระบบและเป็นตัวกลางระหว่างเครือข่ายภายในกับเครือข่ายภายนอก โดยมี

ภาระหน้าที่ที่ถูกกำหนดให้รับผิดชอบ แตกต่างกันไปตามความต้องการของผู้ออกแบบระบบ เช่น **Firewall Proxy** ทำหน้าที่รักษาความปลอดภัยให้แก่ระบบ เป็นต้น

ระบบปฏิบัติการ **FreeBSD** มีโปรแกรม **Proxy Server** ที่มีประสิทธิภาพสูงมาก คือ **Squid** ซึ่งเป็นโปรแกรมประเภท **Proxy Caching Server** สำหรับการให้บริการ **Web Caching Service** คือ จะคอยรับคำร้องขอบริการจากเครื่องลูกข่าย และส่งผ่านไปยังเซิร์ฟเวอร์ปลายทางที่เหมาะสม ข้อมูลต่าง ๆ ที่ผ่านเข้ามาจะถูกสำเนาเก็บไว้ในหน่วยความจำแคช และดิสก์ ดังนั้นเมื่อมีการร้องขอข้อมูลซ้ำอีกในครั้งต่อมาจะ สามารถนำข้อมูลในแคชมาให้บริการได้รวดเร็วกว่าการติดต่อไปยังเซิร์ฟเวอร์โดยตรง ช่วยให้เกิดการใช้ช่องทางสื่อสารข้อมูลลงได้ นอกจากนี้ **Squid** ยังมีคุณสมบัติเป็น **Firewall Proxy** อีกด้วย อย่างไรก็ตาม **Squid** เป็นเพียง **Web Only Cache** หมายถึง จะทำงานกับโปรโตคอล **HTTP** โดยเฉพาะเท่านั้น จะไม่สามารถแคชข้อมูลจากโปรโตคอลอื่น ๆ เช่น **RealAudio** หรือ **FTP** ได้ ยกเว้นกรณีที่ **FTP** นั้นทำงานโดยโปรโตคอล **HTTP** ซึ่งมีเพียงส่วนน้อย และสำหรับโปรโตคอล **SSL** แล้ว **Squid** สามารถทำการแคชการบริการได้ การใช้งานที่ต้องเข้าสู่เว็บไซต์ที่ใช้ **SSL** เช่น เว็บไซต์บริการอีเมลล์ฟรีต่าง ๆ จึงไม่มีปัญหาแต่อย่างใด

2.5 Client-server

Server คือ เครื่องที่ทำหน้าที่ในการให้บริการแก่ **Client**

Client คือ เครื่องที่ไปขอใช้บริการอย่างใดอย่างหนึ่งจาก **Server**

- เป็นระบบที่ทำงานในรูปแบบของความสัมพันธ์ของ **Process** ที่อยู่บนต่างเครื่องกัน
- มีขอบข่ายการทำงานที่แบ่งแยกกันอย่างชัดเจน คือ **process** ที่ใช้เป็นตัวให้บริการอยู่ในเครื่องหนึ่ง ในขณะที่ **Client Process** เป็นตัวใช้บริการจะอยู่อีกเครื่องหนึ่ง

มองตามลักษณะการทำงานและลักษณะการประยุกต์ใช้ได้ดังนี้

File Server

Database Server

Transaction Server

Groupware Server

Object Server

Web Application Server

2.6 File Server

- **First wave Client/Server**
- **Client** ขอใช้บริการข้อมูลที่ **Server** คือ **File** ที่อาจเป็น **Program** หรือ **Data**
- เมื่อ **Client** ส่งสัญญาณ **Request** แล้ว **Server** จะ **Copy** ส่ง **File** ที่ขอให้
- **Process** ส่วนใหญ่เกิดขึ้นที่ **Client** เป็นหลัก **Server** มีหน้าที่แค่บริการ **File**
- เป็น **Model** ที่ง่ายและราคาถูก ตัวอย่างเช่น **PC LAN** โดย **Netware Netware** ไม่ได้มีหน้าที่ **run program** เป็นเพียงผู้ที่เก็บ **program** และแจกจ่ายให้ **user** แต่ละคนนำไปใช้ ระบบจะ **Protect** ไม่ให้สูญหาย และจะดูแล้วว่าข้อมูลใดเป็นของใคร สิทธิการเข้าใช้งานเป็นอย่างไร
- **Print Server** หรือ **Fax Server** เป็นกลุ่มที่ทำงานแบบ **File Server** เช่นกัน เพราะเป็นการส่งผ่านข้อมูลลักษณะ **File** เช่นเดียวกัน
- **Server search file** เดี่ยว : เพียง **program** เดียวก็ทำงานได้ อันเป็นลักษณะของ **Dos**
- **Server search file** เป็นกลุ่ม : ต้องได้ไฟล์ทั้งระบบจึงจะทำงานได้ ซึ่งเป็นระบบการทำงาน ของ **Windows - Data File**
- **End User Data** คือ **data** ที่ **user** สร้างเอง และเก็บไว้ใช้งานเอง - **Database Data** คือ **data** ที่ **share** ให้ **user** หลายคนเรียกใช้งานได้ ลักษณะการใช้งานจะเป็นฐานข้อมูลแบบ **ISAM** ซึ่งต้องสร้าง **Index Files** ขึ้นมาเพื่อให้ง่ายในการสืบค้นและเพื่อเรียงลำดับข้อมูล และการเข้าใช้งานในแต่ละครั้งต้องทำการ **Lock** ไว้จนกว่าจะใช้งานเสร็จ

2.7 Database Server

- **Second wave Client/Server**
- เหมาะกับ **Relational Database Model** แต่ไม่เหมาะกับ **VSAM** หรือ **ISAM Model**
- **Client** จะเรียกโดยการส่ง **SQL request message** ไปยัง **Server** จากนั้น **DBMS** จะทำการประมวลผลคำสั่ง คั้นหรือประมวลผลข้อมูลใน **Database** แล้วส่งผลลัพธ์กลับไปยัง **Client**
- มีปริมาณ **Traffic** น้อยมากใน **Network** เพราะเป็นการส่งผ่านแค่คำสั่งกับผลลัพธ์ถึงกันเท่านั้น
- ภาระทั้งหมดตกอยู่ที่ **Server** เนื่องจากต้องรองรับงานจากทุก **client** บางช่วงเวลาที่มีการส่ง **command** จาก **client** หนักๆจะทำให้ **Server** ไม่สามารถประมวลผลได้ทัน ทำให้ระบบ **Crash**
- ในระดับ **Centralized Host SQL Command** ก็จะถูกสร้างขึ้นที่ **Server** ซึ่งจะต่างจากใน **PC Client** ที่จะส่งมาเป็น **Command** เนื่องจาก **Client** เป็น **Terminal** ที่ไม่มี **CPU** และ **Memory** ทุกตัวอักษรที่กดจะถูกส่งไปที่ **Host** ทั้งหมด และจะเริ่มประมวลผลทันทีที่ **Submit** คำสั่ง
- การตอบสนองจะเป็นไปในลักษณะ **Interactive** แต่ไม่สามารถคาดหมายเวลาในการตอบสนองได้ เนื่องจากขึ้นอยู่กับช่วงเวลานั้นว่า

2.8 Transaction Server

- สร้างขึ้นมาเพื่อแก้ปัญหา **Crash** ของ **Server** ในช่วงเวลา **Busy** หนักๆ
- แบ่งกลุ่ม **Transaction Server** เป็นสองกลุ่มคือ
- **TP Monitor (TP-Heavy)** เหมาะสำหรับงานที่มีปริมาณ **Transaction** มากๆ
- **Store Procedure (TP-Lite)** เหมาะสำหรับงานที่มีปริมาณ **Transaction** ไม่มาก
- ลักษณะงานของ **Transaction Server** คือ รองรับ **Transaction** ทั้งหมดจาก **User** แล้วส่งไป **Process** ที่ **Server** ในอัตราสูงสุดไม่เกินปริมาณนี้ **Server** สามารถจะรับได้ในแต่ละช่วงเวลา

2.9 Groupware Server

- ใช้งานกับ **Data** ชนิดที่ไม่มีโครงสร้าง (**Un-Structure**) หรือกึ่งโครงสร้าง (**Semi-Structure**)
- รองรับงานจำพวก **Email** , **Scheduling** , **Conferencing** , **Workflow** หรือบางครั้งก็เป็น **Multimedia** หรืองานที่มีลำดับการทำงานไม่แน่นอน

Object Server

- เป็นเทคโนโลยี **Client/Server** ในยุคต่อไป
- **1 Application** เกิดจากการนำ **object** หลากๆตัวที่มี **relation** ต่อกันและใช้วิธี **Object Oriented Programming Language** ในการเขียน
- **1 Object** คือ **Unit** อิสระที่สามารถ **process application** เองได้ ถ้าต้องการสร้างโปรแกรมใหม่ก็เพียงแต่นำ **Object** ที่มีอยู่แล้วมาสร้างใหม่ได้
- เป็นตัวรวบรวม **object** ทั้งหมด และมี **ORB (Object Request Broker)** จะเป็นตัวเชื่อมต่อระหว่าง **Client** กับ **Server** โดยเมื่อ **Client** จะติดต่อไปหา **Object** ก็จะติดต่อไปหา **ORB** เพื่อบอกว่าต้องการ **Object** ใดๆ **ORB** ก็จะจัดสรรให้ตามต้องการ
- **OMG : Object Management Group** เป็นกลุ่มงานที่กำหนด **frame work** ของการสร้าง **object** เพื่อกำหนดว่าบริษัทผู้ผลิต **object** จะออกมาอย่างไรให้เข้ากันได้ **100%**
- **COBRA : Common Object Broker Adapter** เป็นแกนกลางที่ทำให้ **object** ที่มาจากบริษัทที่ต่างกันสามารถแลกเปลี่ยนหรือใช้ร่วมกันได้
- **Component** : คือชิ้นส่วนที่สามารถนำไปใช้ได้หลายๆอย่าง แต่มันจะมีหน้าที่ของมันเอง โดยเราจะนำเอา **Component** เหล่านี้มาสร้างเป็น **Application**

2.10 Web Server

- **www** เป็น **Application Level Protocol** ซึ่งเดิมเป็นระบบ **Web-Client** เท่านั้น กล่าวคือ **Client** จะมี **browser** ที่จะ **request** ไปยัง **host** ที่ต้องการ แล้ว **host** ก็จะทำการส่ง **html** กลับมายัง **client** เพื่อ **process** แล้วแสดงผลออกมา
- ปัจจุบันพัฒนาเป็น **Front-End Application Client**

2.11 Firewall

Firewall คือ ฮาร์ดแวร์และซอฟต์แวร์ที่องค์กรต่างๆมีไว้เพื่อป้องกันเครือข่ายคอมพิวเตอร์ภายในของตนจากอันตรายที่มาจากเครือข่ายคอมพิวเตอร์ภายนอก เช่น ผู้บุกรุก หรือ **Hacker** **Firewall** จะอนุญาตให้เฉพาะข้อมูลที่มีคุณลักษณะตรงกับเงื่อนไขที่กำหนดไว้ผ่านเข้าออกระบบเครือข่ายภายในเท่านั้น อย่างไรก็ดี **Firewall** นั้นไม่สามารถป้องกันอันตรายที่มาจากอินเทอร์เน็ตได้ทุกรูปแบบ ไวรัสนี้เป็นหนึ่งในนั้น ดังนั้นจึงไม่สามารถรับรองได้ว่าความปลอดภัยหรือความลับของข้อมูลจะมีอยู่ร้อยเปอร์เซ็นต์ถึงแม้ว่าจะมีการใช้ **Firewall** แล้วก็ตาม

เครื่องบริการรักษาความปลอดภัยของข้อมูล มีหลายที่หลายด้าน เช่น กรองแพ็กเกจ ฝ้าตรวจ ตรวจสอบการใช้แบนด์วิดท์หรือเก็บข้อมูลที่ **local host** ใช้งานบ่อย ไว้ให้ **local host** อื่น ๆ เรียกใช้ด้วยความเร็ว ซึ่งหลักการของ **firewall** ที่สำคัญมีดังนี้

1. ให้บริการเฉพาะที่ต้องการเปิด
2. ให้บริการใครบ้าง
3. ให้สมาชิกในแต่ละกลุ่ม สามารถใช้ได้เฉพาะที่เหมาะสม
4. กำหนดความปลอดภัยอย่างไร ให้แต่ละบริการ

211.1 Firewall ประกอบด้วย

1. **Package filters** เป็นหน้าที่ของ **router** ทำหน้าที่กรอง **IP** (อินเทอร์เน็ต Protocol) **TCP** (Transmission Control Protocol) และ **UDP** (User Datagram Protocol) ถ้าข้อมูลที่ส่งมาไม่เป็นไปตามกฎ ก็จะเข้าเครือข่ายไม่ได้
2. **Proxy server** ทำให้เครื่องภายนอกทั้งหมดไม่รู้จักเครื่องภายในเครือข่าย เพราะทุกกิจกรรมต้องผ่านการแปลง **IP** ของ **Proxy server**

212 Network Address Translation (NAT)

จากการที่ทุกหน่วยงานของมหาวิทยาลัย ได้ให้ความร่วมมือในการปรับเปลี่ยนมาใช้ **MUC-Net Address** เพื่อความมั่นคงและเสถียรภาพของระบบคอมพิวเตอร์ในมหาวิทยาลัย ในฉบับนี้เราจะมารู้จักกับ **NAT (Network Address Translation)** ที่เปลี่ยน **MUC-Net Address** เป็น **IP Address** เพื่อใช้งาน อินเทอร์เน็ต โดยผู้ใช้ระบบเครือข่าย อินเทอร์เน็ต ไม่ทราบ **IP Address** จริงที่เรากำลังใช้งานอยู่เป็นอย่างไร

NAT เป็นมาตรฐานหนึ่งใน อินเทอร์เน็ต ที่ทำให้เราสามารถใช้ **IP Address** ชุดหนึ่งเป็น **IP** ภายใน และอีกชุดหนึ่งเป็น **IP** ภายนอก เวลาเราจะส่งข้อมูลไปใน อินเทอร์เน็ต ข้อมูลจะถูกตัดแบ่งเป็นส่วนๆ ซึ่งเราเรียกว่า **IP Packet** ต้องระบุที่อยู่ผู้ส่งและที่อยู่ผู้รับลงใน **IP Packet** ด้วย เพื่อให้ **Router** ในระบบ อินเทอร์เน็ต รู้ว่าจะส่งไปโดยใช้เส้นทางใด ดังนั้น **NAT** จะเปลี่ยนที่อยู่ของผู้ส่งภายในเมื่อจะส่งออกไป อินเทอร์เน็ต และเปลี่ยนที่อยู่ผู้รับภายในเมื่อได้รับจาก อินเทอร์เน็ต นั่นเอง เรามักใช้ **Private IP Address** ซึ่งเป็นหมายเลข **IP Address** ที่ไม่ใช้ในระบบ อินเทอร์เน็ต เป็น **IP Address** ภายในเครือข่าย เช่น **Private IP Address** ของ **Class A** ได้แก่ หมายเลข **10.0.0.0**, **Class B** ได้แก่ หมายเลข **172.16.0.0 - 172.31.0.0** และ **Class C** ได้แก่ หมายเลข **192.168.0.0 - 192.168.255.0** เป็นต้น

2121 จุดประสงค์ของการใช้ NAT

1. ใช้เพื่อแปลง **IP Address** ภายในระบบเครือข่ายหลาย ๆ **Address** ออกไประบบภายนอกเครือข่ายเป็น **1 IP Address** ทำให้ประหยัด **IP Address** เนื่องจากปัจจุบันเราใช้ **IP Address Version 4** ซึ่งมีรูปแบบ **x.x.x.x** ตัวอย่างเช่นระบบเครือข่ายของมหาวิทยาลัยมหิดล ได้รับ **IP Address Class C** จาก **APNIC (Asia Pacific Network Information Centre)** จำนวนจำกัด โดย **1 ชุด IP Address Class C** สามารถติดตั้งให้เครื่องคอมพิวเตอร์ได้สูงสุดเพียง **254 เครื่อง** (โดย **28 = 256** ซึ่ง **x.x.x.0** เป็นหมายเลขของ **Network** และ **x.x.x.255** เป็นหมายเลขของ **Broadcast**) จึงต้องใช้ **NAT** เพื่อรองรับการติดตั้งเครื่องคอมพิวเตอร์ได้มากขึ้น
2. ใช้เพื่อประโยชน์ทางด้านความปลอดภัย เพราะเมื่อ **IP Packet** ออกไปในระบบอินเทอร์เน็ต แล้วคนอื่นที่อยู่ใน อินเทอร์เน็ต จะไม่รู้ว่ามีมาจากเครื่องคอมพิวเตอร์เครื่องใด ทำให้ป้องกันจากผู้บุกรุกภายนอกได้ ซึ่งระบบเครือข่ายของมหาวิทยาลัยมหิดลใช้เป็นเหตุผลหลักในการใช้ **NAT**
3. ใช้เพื่อทำ **Load Sharing** ในกรณี **Server** เช่นบางครั้งเราต้องการให้บริการงานแก่ผู้ใช้จากเครื่อง **Server** หลายเครื่อง โดยผู้ใช้ไม่จำเป็นต้องรู้ว่างานที่จะทำ ต้องขอบริการจากเครื่อง **Server** ใด และถ้าเครื่อง **Server** บางตัวในระบบที่ทำ **Load Sharing** ไม่สามารถให้บริการได้ ระบบจะให้บริการจาก **Server** ตัวอื่นแทน

213 Telnet หรือ SSH

เครื่องมือพื้นฐาน ที่ใช้สำหรับติดต่อกับเครื่อง **Server** ที่เป็น **UNIX** หรือ **LINUX** เพื่อใช้เข้าไปควบคุมการทำงานของเครื่อง หรือใช้อ่าน **mail** หรือใช้ปรับปรุง **homepage** หรือใช้เรียกโปรแกรมประมวลผลใด ๆ หรือใช้พัฒนาโปรแกรมและใช้งานในเครื่องนั้น เป็นต้น เพราะระบบ **UNIX** หรือ **LINUX** จะยอมให้ผู้ใช้สร้าง **application** ด้วย **Compiler** ภาษาต่าง ๆ ได้อย่างอิสระ และเชื่อมต่อกับ อินเทอร์เน็ต ได้อย่างมีประสิทธิภาพ เพราะ อินเทอร์เน็ต เริ่มต้นมาจากระบบ **UNIX** นี้เอง ประโยชน์อย่างหนึ่งที่ผู้ใช้โปรแกรม **Telnet** มักคุ้นเคย คือการใช้โปรแกรม **PINE** ซึ่งมีอยู่ใน **Telnet** สำหรับรับ-ส่ง **mail** และมีผู้ใช้อีกมากที่ไม่รู้ตัว ว่าตนเองกำลังใช้งาน **UNIX** อยู่ ทั้ง ๆ ที่ใช้ **PINE** ติดต่อกันอยู่ทุกวัน เดิมทีระบบ **UNIX** ไม่มีโปรแกรม **PINE** แต่มีนักศึกษาที่มหาวิทยาลัย **WASHINGTON University** เพราะใช้ง่ายกว่าการใช้คำสั่ง **mail** ในการรับ-ส่งมาก แต่ผู้ใช้ที่ใช้ **E-Mail** กับเครื่อง **UNIX** หรือ **LINUX** ซึ่งใช้ตามมาตรฐาน **IMAP** มักเป็นกลุ่มนักศึกษาในมหาวิทยาลัย ที่มหาวิทยาลัยได้ให้บริการ **E-Mail** ซึ่งมีข้อจำกัดบางประการ และหลายมหาวิทยาลัย เช่นกันที่ทำฐานข้อมูล **Mail** ใน **UNIX** และให้บริการ **Mail** ผ่าน **browser** ได้ ซึ่งเป็นหลักการที่ผู้

ให้บริการ **mail** ฟรีหลายแห่งใช้กันอยู่ สำหรับโปรแกรม **Telnet** ผู้ต้องการใช้บริการ ไม่จำเป็นต้องไป **download** เพราะเครื่องที่ทำการติดตั้ง **TCP/IP** จะติดตั้งโปรแกรม **telnet.exe** ไว้ในห้อง **c:\windows** เพื่อให้ผู้ใช้สามารถเรียกใช้ได้อยู่แล้ว แต่ปัญหาหนักอยู่ที่วิธีการใช้ เพราะระบบ **UNIX** เป็นการทำงานใน **Text mode** เป็นหลัก การจะใช้คำสั่งต่าง ๆ ผู้ใช้จะต้องเรียนรู้มาก่อน จึงจะใช้งานได้ ในระดับที่ฟังตนเองได้ มิเช่นนั้นก็จะเหมือน คนตาบอดเดินอยู่กลางถนน หากใช้คำสั่งไม่ถูกต้อง อาจทำให้เกิดปัญหาทั้งกับตนเอง และระบบได้

214 WWW (World Wide Web)

บริการที่ต้องใช้โปรแกรม **Browser** เช่น **Fire Fox, Netscape, อินเทอร์เน็ต Explorer, Opera** หรือ **Neoplanet** เพื่อเปิดดูข้อมูลจากเว็บไซต์ (**Website**) หรือ โฮมเพจ (**Homepage**) จะได้ข้อมูลในลักษณะเป็นตัวอักษร ภาพ เสียง และภาพเคลื่อนไหว ในลักษณะสื่อผสม รวมทั้งการสั่งประมวลผล และตอบสนองแบบอินเทอร์แอ็กทีฟ (**Interactive**) บริการนี้ได้รับการพัฒนาอย่างรวดเร็ว จนนำมาใช้งานอย่างหลากหลาย เช่น ชมภาพยนตร์ ฟังเพลงออนไลน์ เล่นเกม ค้นข้อมูล ประมวลผล ข้อมูลทางธุรกิจ ทำข้อสอบ การส่งเมลล์ ติดต่อซื้อขาย ชำระเงินด้วยบัตรเครดิต หรือส่งโปสท์การ์ด เป็นต้น

บทที่ 3

การออกแบบโครงสร้างระบบ

3.1 การสำรวจความต้องการของระบบ

การสำรวจความต้องการของระบบนี้ผู้จัดทำโครงนี้ได้รวบรวมข้อมูลโดยการ สัมภาษณ์ จากผู้ใช้งานอินเทอร์เน็ตโดยตรง โดยรวบรวมปัญหาต่าง ๆ ที่เกิดขึ้นสำหรับการใช้งานอินเทอร์เน็ต ในเครือข่าย LAN (Local Area Network) ซึ่งผู้จัดทำได้นำเอาปัญหาเหล่านั้นมาวิเคราะห์ว่าจะแก้ไข ปัญหาที่เกิดขึ้นกับเครือข่าย LAN นั้นอย่างไร โดยจากการสัมภาษณ์นั้นปัญหาส่วนใหญ่ที่เกิดขึ้นก็คือ การใช้งานอินเทอร์เน็ตได้ช้าเพราะผู้ใช้บางราย **download** ไฟล์บางอย่าง และ การเข้าเว็บไซต์ ที่ไม่เหมาะสม จึงเป็นที่มาของการทำโครงการนี้

3.2 แนวทางการพัฒนาระบบ

การพัฒนาระบบนี้ได้นำหลักการหลักการของ SDLC มาใช้ในการวิเคราะห์และออกแบบ ระบบการจัดการความเร็วในการใช้งานอินเทอร์เน็ตสำหรับเครือข่าย LAN ซึ่งมีขั้นตอนในการ วิเคราะห์ดังนี้

1. กำหนดปัญหา

เนื่องจากการใช้งานอินเทอร์เน็ตสำหรับเครือข่าย LAN ในปัจจุบันผู้ใช้งานบางราย มักจะทำการ **Download** ไฟล์บางอย่างจากอินเทอร์เน็ตทำให้ช่องสัญญาณแบนด์วิดท์ถูกใช้ ไปเป็นจำนวนมากทำให้ผู้ใช้งาน ที่ไม่ได้ทำการ **download** นั้นใช้งานอินเทอร์เน็ตได้ช้าลง หรือหากมีผู้ **Download** หลาย ๆ คนในเครือข่าย LAN นั้น อาจจะทำให้ผู้ใช้ ที่เข้ามาใช้งาน ภายหลัง อาจจะไม่สามารถใช้งานอินเทอร์เน็ตได้

2. วิเคราะห์ปัญหาและศึกษาความเป็นไปได้

เนื่องจากปัญหาที่เกิดขึ้นนั้นทำให้เกิดความไม่เสมอภาคกัน ในการใช้งานอินเทอร์เน็ต เพราะทุกคนที่อยู่ในเครือข่าย LAN เดียวกันก็สมควรที่จะได้ใช้งานช่องสัญญาณของ แบนด์วิดท์ เท่าๆกัน

3 ออกแบบระบบ

การออกแบบระบบการจัดการความเร็วในการใช้งานอินเทอร์เน็ตสำหรับเครือข่าย LAN นี้ได้นำ การออกแบบระบบโดยใช้ ภาษา **UML** ซึ่งเป็นภาษาที่กำลังได้รับความนิยมเป็นอย่างสูงในขณะนี้สำหรับผู้ที่ทำหน้าที่ในการออกแบบระบบเพราะเป็นโมเดลมาตรฐานที่ใช้หลักการออกแบบ **OOP** ซึ่งเหมาะสมกับการเขียนโปรแกรมในปัจจุบันนี้

4 พัฒนาระบบ

ในการพัฒนาระบบการจัดการความเร็วในการใช้งานอินเทอร์เน็ตสำหรับเครือข่าย LAN นี้ได้ใช้ ระบบปฏิบัติการ **FreeBSD** มาเป็น เซิร์ฟเวอร์ ในการบริหารจัดการ แบนด์วิดท์ และ ทำการกรอง เว็บไซต์ ที่ไม่พึงประสงค์ เพื่อไม่ให้ **User** ในระบบของเราเข้าใช้งาน และได้พัฒนาระบบให้มีความง่ายในการใช้งานมากยิ่งขึ้นโดยใช้การ ควบคุมจาก **Web Application** ในการควบคุมระบบปฏิบัติการ **FreeBSD** นี้ซึ่งใช้ภาษา **PHP** มาพัฒนา

5 ทดสอบระบบ

การทดสอบระบบนั้นได้มีการทดสอบว่าระบบปฏิบัติการ **FreeBSD** นี้สามารถทำการ จัดสรรแบนด์วิดท์และ **blockweb** ที่ไม่พึงประสงค์ได้จริง

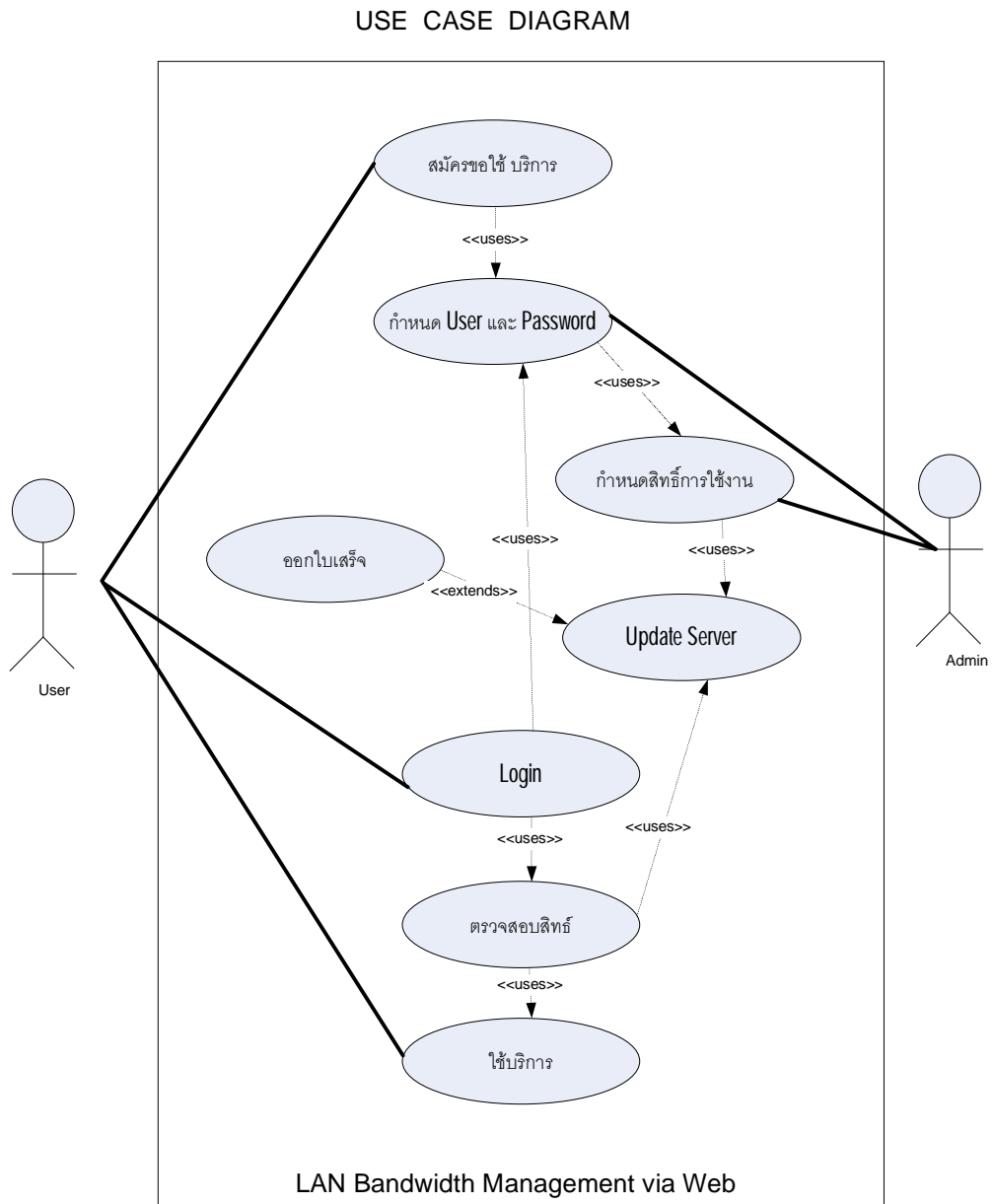
6 ซ่อมบำรุง

สำหรับการซ่อมบำรุงนั้นหากมีความผิดพลาดในส่วนตัว ก็จะทำการปรับปรุงระบบให้สามารถทำงานได้อย่างมีประสิทธิภาพ

3.3 การออกแบบการทำงานของระบบ

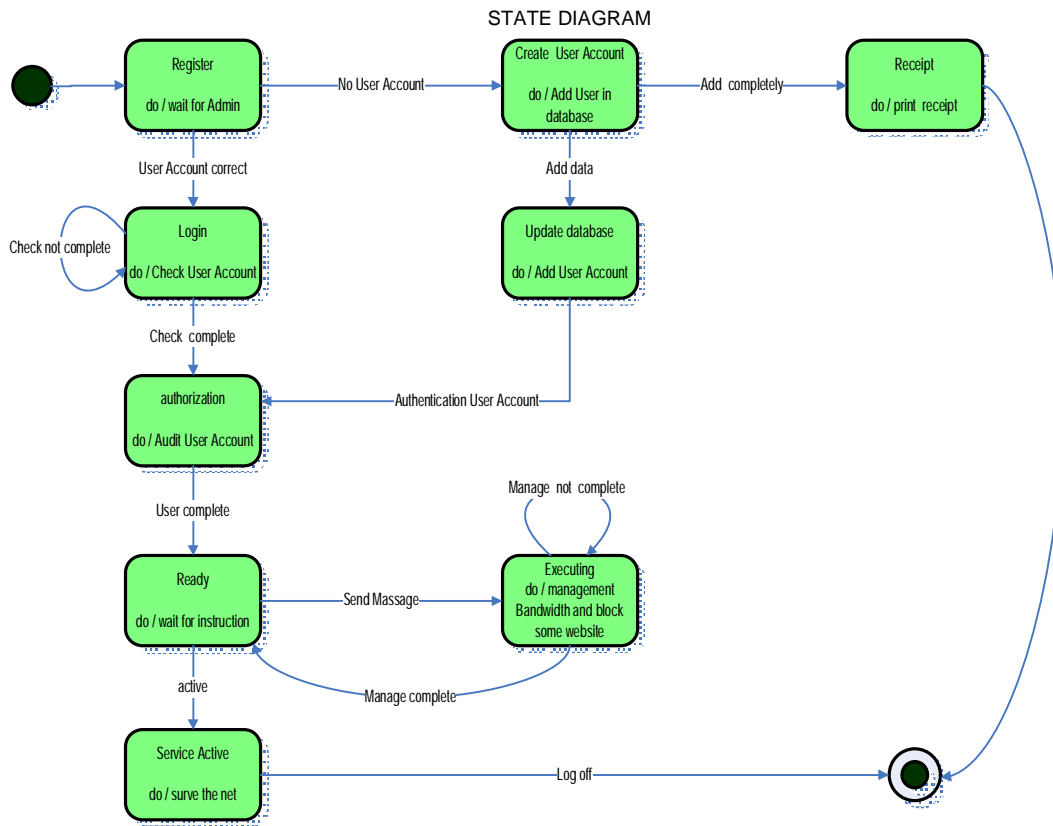
ขั้นตอนการออกแบบระบบการจัดการความเร็วในการใช้งานอินเทอร์เน็ตสำหรับเครือข่าย LAN โดยใช้ **UML** มีวิธีดังนี้

USE CASE DIAGRAM



รูปที่ 31 USE CASE DIAGRAM

STATE DIAGRAM



รูปที่ 3.2 STATE DIAGRAM

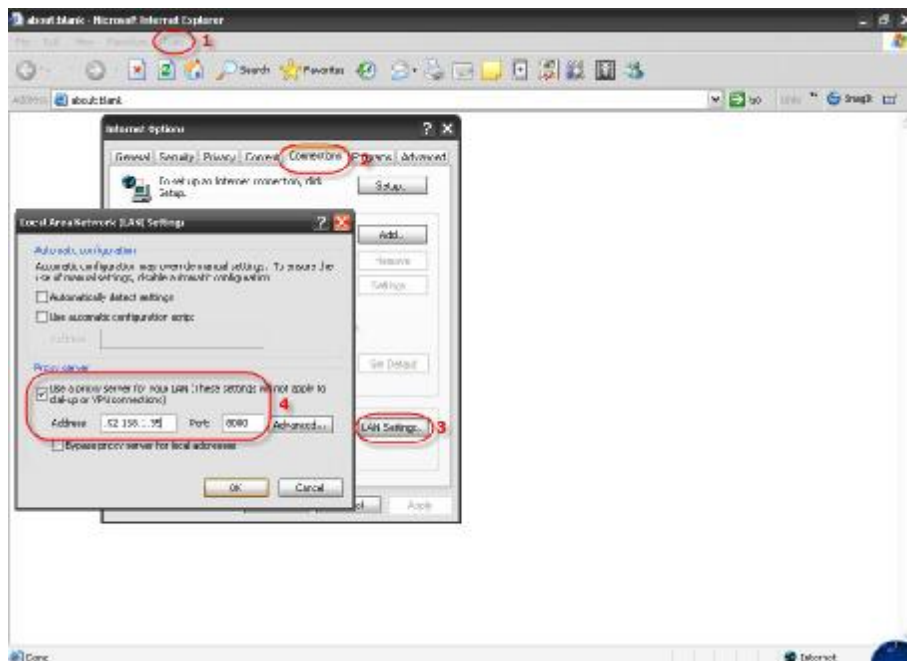
บทที่ 4

ผลการพัฒนาระบบ

ผลการพัฒนาระบบควบคุมความเร็วในการใช้งานอินเทอร์เน็ตสำหรับเครือข่าย LAN ผ่านเว็บไซต์ ได้แบ่งระบบออกเป็น 2 ส่วนด้วยกันคือ ส่วนของผู้ใช้งาน และส่วนของผู้ดูแลระบบ โดยในส่วนของผู้ดูแลระบบนั้นจะทำงานในส่วนของเครื่องเซิร์ฟเวอร์เพื่อบริหารจัดการกับระบบ และในส่วนของผู้ใช้งานนั้นจะทำงานที่เครื่องลูกข่าย ซึ่งทั้ง 2 ส่วนนี้สามารถใช้งานได้โดยง่าย

4.1 ขั้นตอนการทำงานของเครื่องลูกข่าย

1. เครื่องลูกข่ายทุกเครื่องต้องทำการใส่ค่าของ **Proxy** และ **Port** ที่ **Web Browser** ของเครื่องลูกข่ายโดยค่า **Proxy** และ **Port number** นั้นจะถูกกำหนดโดยผู้ดูแลระบบและใส่ไว้ให้เรียบร้อยแล้วดังตัวอย่างรูปที่ 4.1
2. เมื่อผู้ใช้ต้องการใช้งานอินเทอร์เน็ต โดยได้มีการเรียกใช้ **Web Browser** ระบบจะให้ผู้ใช้งาน กรอก **Username** และ **Password** ลงไปดังรูปที่ 4.2 หากผู้ใช้งาน กรอก **Username** และ **Password** ได้ถูกต้องระบบจะอนุญาต ให้ใช้งานได้ทันที แต่หากผู้ใช้งาน กรอก **Username** และ **Password** ผิด ระบบก็จะให้ ผู้ใช้งานได้กรอก **Username** และ **Password** ใหม่ หากกรอกผิดเกิน 3 ครั้งระบบจะมีการแจ้งเตือน
3. เมื่อผู้ใช้งานสามารถเข้าระบบได้แล้วก็จะสามารถใช้งานอินเทอร์เน็ตได้ตามสิทธิของกลุ่มผู้ใช้งานคนนั้น



รูปที่ 41 แสดงวิธีการกรอก IP และ Port ของ เซิร์ฟเวอร์



รูปที่ 42 แสดงหน้าต่างสำหรับกรอก Username และ Password

4.2 ขั้นตอนการทำงานในส่วนของผู้ดูแลระบบ

1. เมื่อผู้ดูแลระบบต้องการจัดการกับระบบของตน ก็จะต้องเรียกใช้งาน **Web Browser** โดยใส่ **IP** ของเครื่องเซิร์ฟเวอร์ ที่ ช่อง **Address Bar** ให้ถูกต้อง
2. และเมื่อ กรอก **Address** ของ **IP** ได้ถูกต้องแล้ว โปรแกรมจะให้ผู้ดูแลระบบทำการกรอก **Username** และ **Password** เนื่องจากการเข้ามาใช้งานนี้สามารถที่จะเพิ่มชื่อของผู้ใช้งานหรือลบชื่อผู้ใช้งานได้ จึงเป็นการป้องกันไม่ให้ผู้อื่นที่ไม่ใช่ ผู้ดูแลระบบเข้ามาในส่วนนี้ได้
3. เมื่อผู้ดูแลระบบเข้าระบบแล้ว จะพบกับ เมนูต่างๆ ดังนี้

- Block	- Online User
- Delete User	- All User
- Add User	- Statistic

โดยเมื่อดังกล่าวสามารถทำงานได้ดังนี้

Block เมื่อผู้ดูแลระบบเห็นว่าเว็บไซต์ไหนที่ไม่เหมาะสมกับการใช้งานในองค์กรของท่าน ก็สามารถใส่ ชื่อ ของ **URL** ดังกล่าวนั้นได้ เพื่อป้องกันไม่ให้ ผู้ใช้งานเข้าใช้งาน **URL** ดังกล่าวโดยสามารถแยกการ **Block** ได้เป็นกลุ่ม ของผู้ใช้งาน หรือ ทั้งหมดของผู้ใช้งานได้

Delete User เมื่อต้องการลบ ชื่อของผู้ใช้งานออกจากระบบ ผู้ควบคุมระบบ สามารถที่จะทำการลบได้โดยกรอกชื่อของผู้ใช้งานนั้นๆ

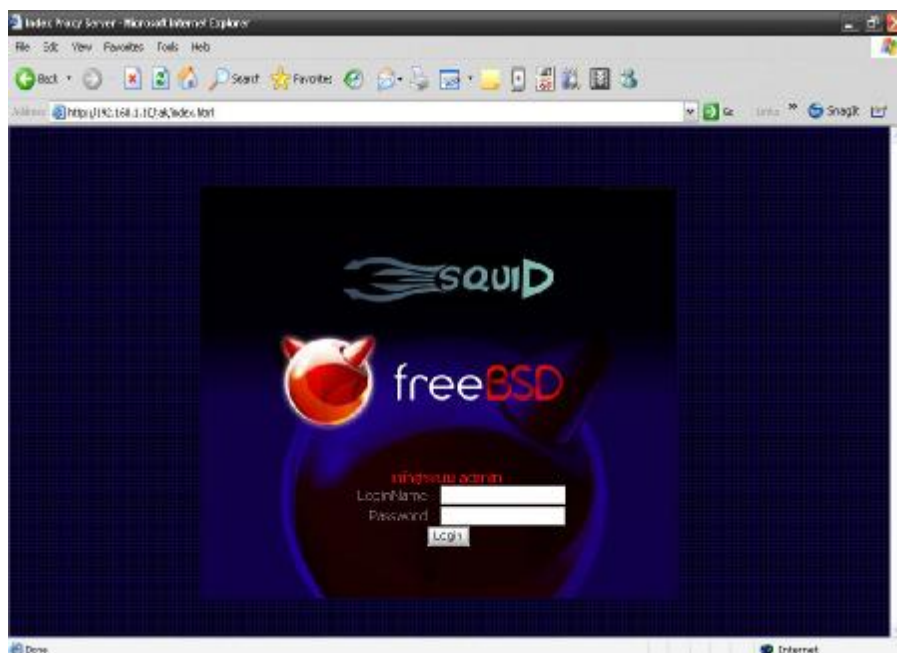
Add User เมื่อมีผู้ใช้งานเพิ่ม ผู้ดูแลระบบสามารถ เพิ่มชื่อของผู้ใช้งาน และรหัสผ่านพร้อมกับ ระบุว่า ผู้ใช้งานคนนั้นอยู่ในกลุ่มไหนได้โดยง่าย

Online User ส่วนนี้จะทำการตรวจสอบว่าในขณะนั้นมีผู้ใช้งานใดที่กำลังใช้งานอยู่ พร้อมทั้งบอกได้ด้วยว่ากำลังใช้งาน **URL** ใด

All User เมื่อต้องการดูรายชื่อของผู้ใช้งานทั้งหมดที่มีอยู่ในระบบ ก็สามารถดูได้โดยง่าย อีกทั้งยังสามารถ ดูผู้ใช้งานว่าอยู่ในกลุ่มไหนได้อย่างสะดวก

Statistic เป็นการดูสถิติ ต่างๆ ว่า ผู้ใช้งานใดที่ **Download** ไฟล์มากที่สุด เข้าเว็บไซต์ที่ได้ **Block** ไว้ หรือดูว่า เว็บไซต์ใดที่มีผู้ใช้งานเข้าใช้งานมากที่สุด รวมทั้งผู้ใช้งานแต่ละรายเข้าเว็บไซต์ใดบ้าง เพื่อเก็บข้อมูลไว้สำหรับ แก้ปัญหาต่างๆ ของระบบต่อไป

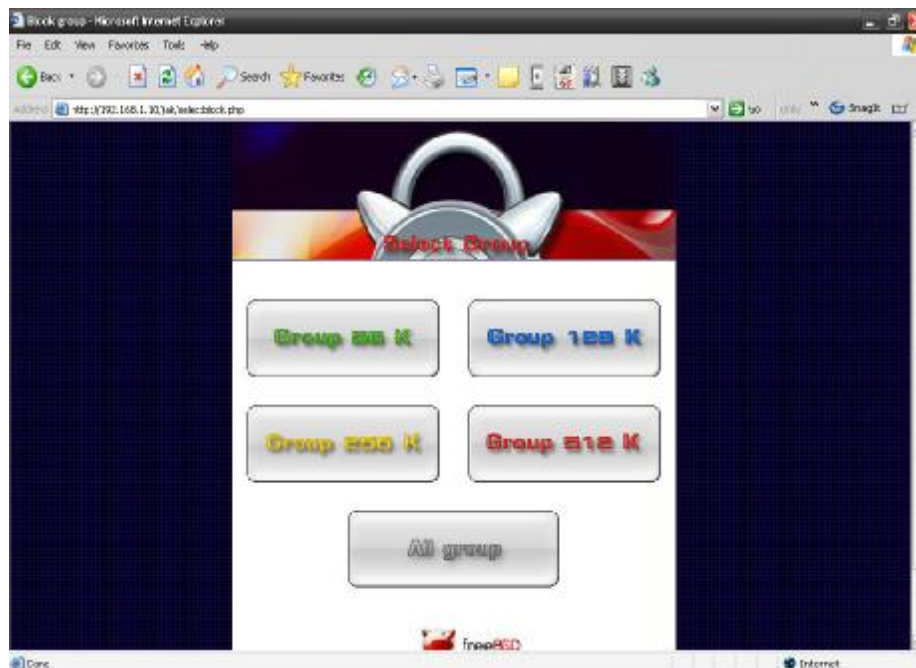
4.3 การออกแบบ Interface ในส่วนของผู้ดูแลระบบ



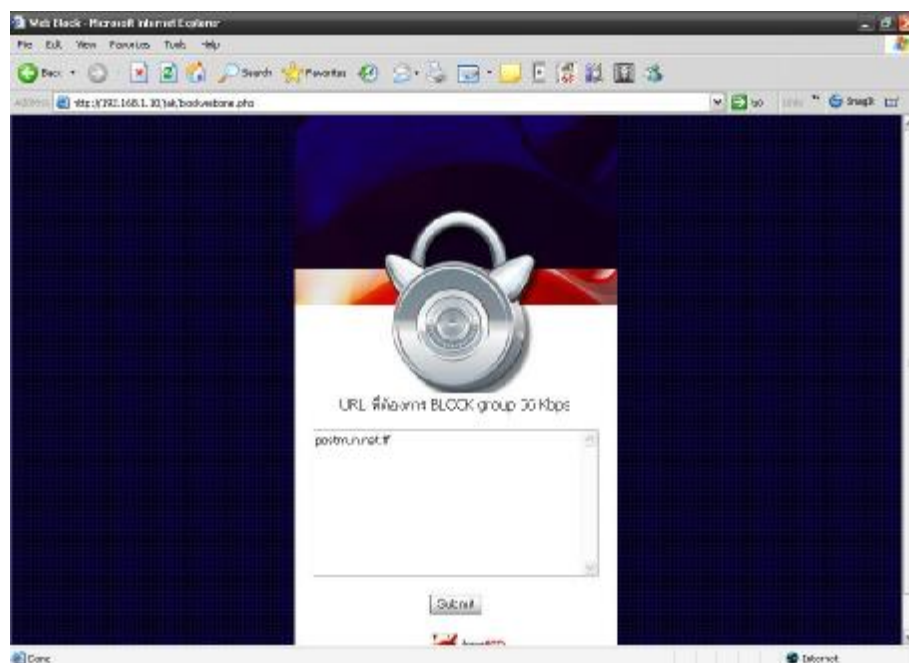
รูปที่ 4.3 หน้าจอเข้าสู่ระบบของผู้ดูแลระบบ



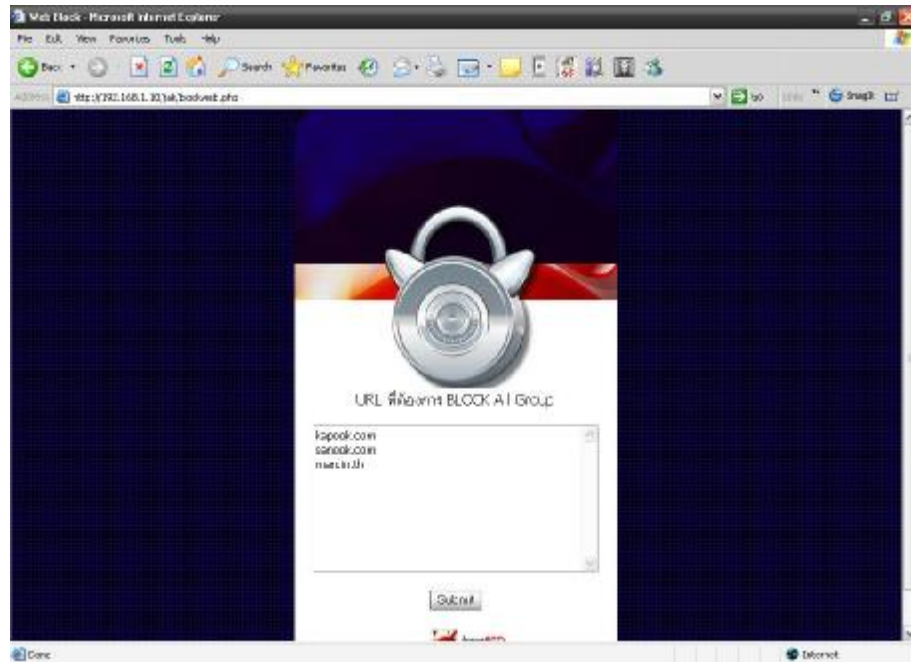
รูปที่ 4.4 หน้าจอของเมนูหลักของระบบ



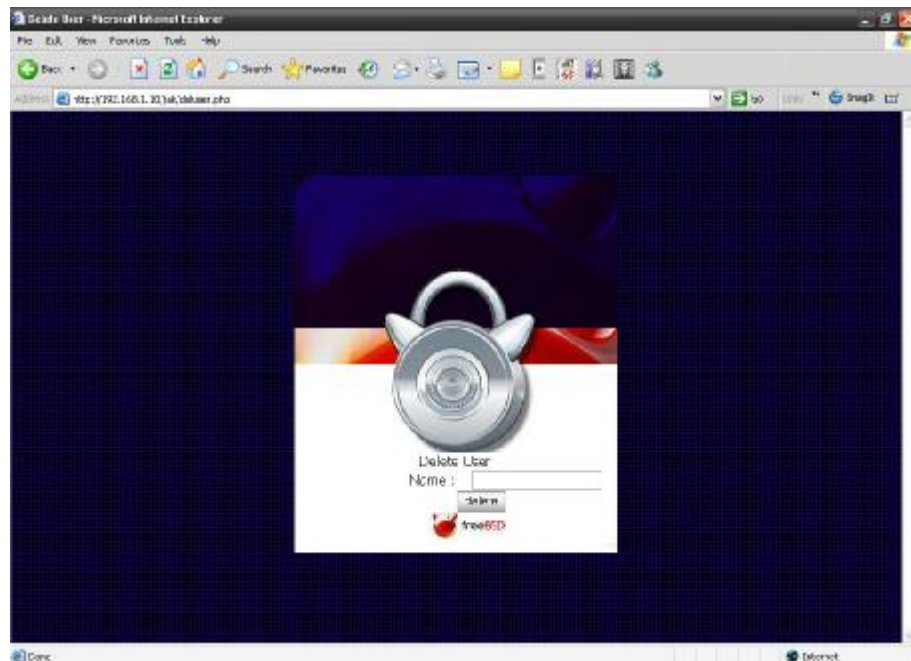
รูปที่ 45 หน้าจอหลัก ของเมนู Blockเว็บไซต์



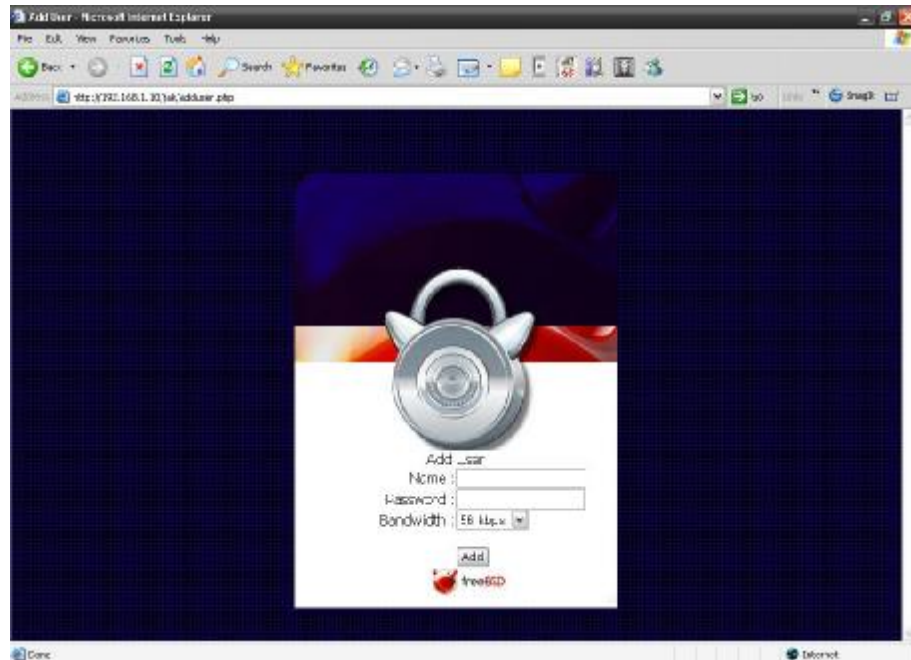
รูปที่ 46 หน้าจอ แสดง URL ที่ต้องการ Block group 56 Kbps



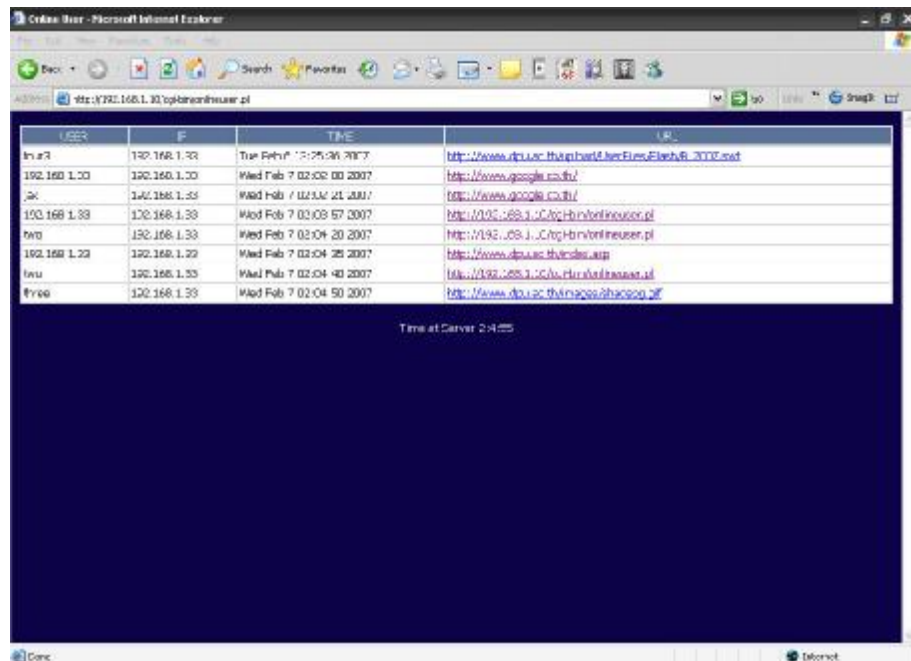
รูปที่ 47 หน้าจอ แสดง URL ที่ต้องการ Block สำหรับผู้ใช้งานทั้งหมดในระบบ



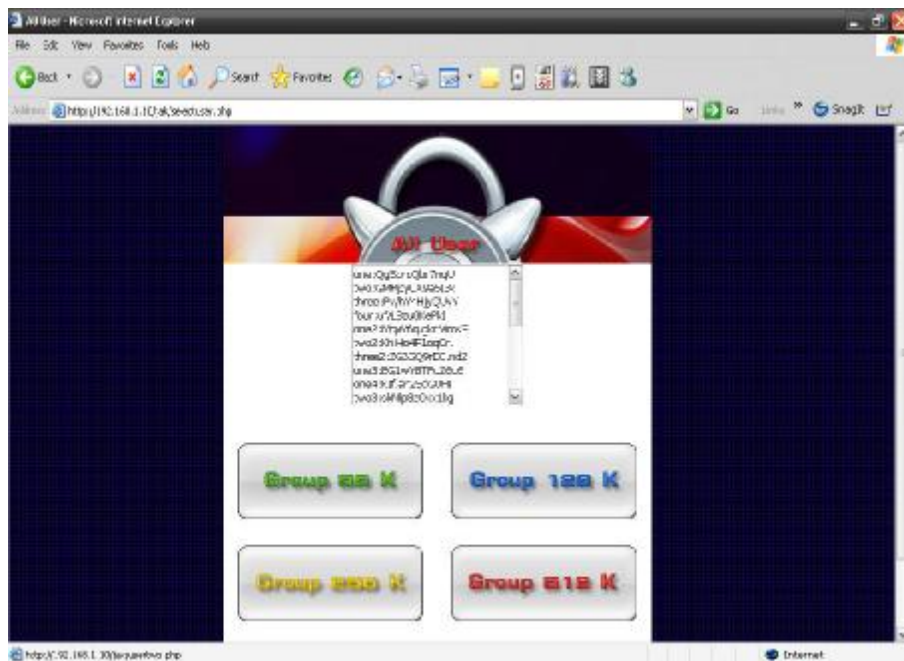
รูปที่ 48 หน้าจอสำหรับลบรายชื่อผู้ใช้งาน



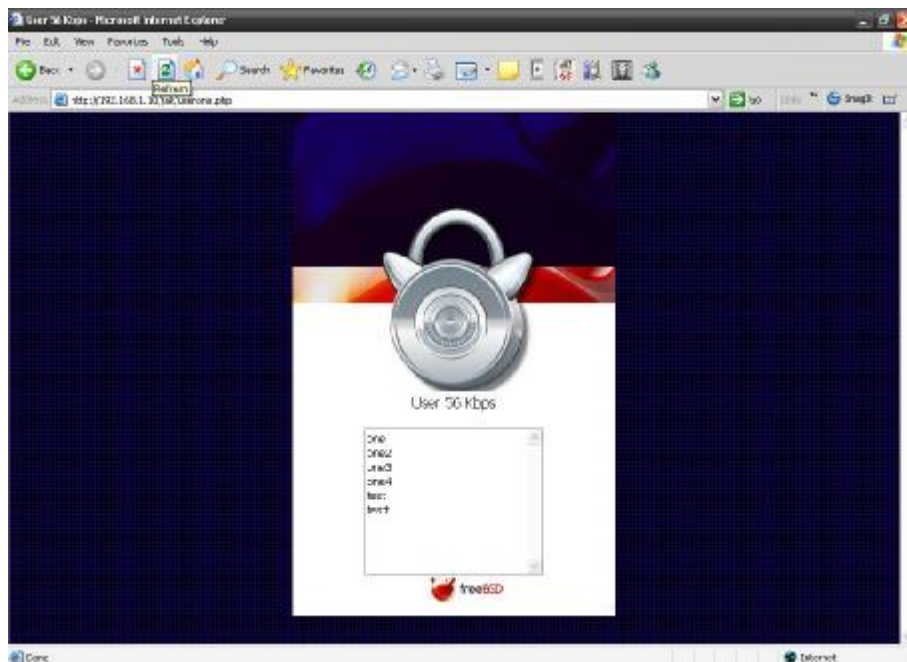
รูปที่ 49 หน้าจอสำหรับเพิ่มรายชื่อผู้ใช้งาน



รูปที่ 410 หน้าจอแสดงผู้ที่กำลังใช้งาน



รูปที่ 411 หน้าจอหลักของ รายชื่อผู้ใช้งานทั้งหมดในระบบ



รูปที่ 412 หน้าจอแสดงรายชื่อของผู้ใช้งานในกลุ่ม 56 Kbps

Squid Analysis Report Generator

Squid User Access Report

FILE/URI/URL	CREATION DATE	BYTES	OBJECTS	AVERAGE
2007Jan05-23:13:40:03	Sat Jan 5 20:16:28 ICT 2007	4	37,4196	9.3596
21Jan2007-01:14:21:01	Fri Jan 5 1:14:21 ICT 2007	12	39,2596	9.5696
21Jan2007-03:Feb 2:00T	Tue Feb 6 12:02:18 ICT 2007	11	39,2596	9.5696
21Jan2007-01:Feb 2:00T	Sat Feb 3 13:49:23 ICT 2007	8	34,3196	7.0296
21Jan2007-01:Feb 2:00T	Sat Feb 3 0:02:52:45 ICT 2007	0	54,2196	7.0296
21Jan2007-09:Feb 2:00T	Sun Feb 1 1:18:29 ICT 2007	12	58,1196	8.0096
21Jan2007-21:Jan 20:00T	Thu Jan 31 21:16:34 ICT 2007	9	39,5496	4.7296
05Jan2007-01:Jan 20:00T	Fri Jan 4 0:02:59:01 ICT 2007	26	44,796	891.8796
05Jan2007-03:Jan 20:00T	Fri Jan 4 21:16:25 ICT 2007	4	47,2796	7.0796
05Jan2007-03:Jan 20:00T	Sat Jan 5 20:40:27 ICT 2007	4	37,4196	9.3596

Generated by squid-2.6.0-1.388-0502281 at 13:48:00/07-10-42

รูปที่ 413 หน้าจอสำหรับเลือกวันที่ เก็บสถิติต่างๆ

Squid Analysis Report Generator

Squid User Access Report

File: 21Jan2007-09:Feb 2:00T

URI: 01:Feb 2:00T

Report

Time: 01:Feb 2:00T

Device: 01:Feb 2:00T

Without Cache: 01:Feb 2:00T

URI	METHOD	CONNECT	BYTES	OBJECTS	DISK-IO	ELAPSED TIME	MISSES	% HIT	
1	GET	23:29	20,429	29,3996	1,4196	99,6196	1:01:11:11	1:21:02:26	281.9%
2	GET	1:14	8,0619	14,1196	1,1196	99,6196	1:01:20:25	1:17:21:44	99.6%
3	GET	3:01	8,0619	14,1196	1,1196	99,6196	1:01:11:11	1:02:11:44	50.7%
4	HEAD	9:56	8,0619	11,2696	1,1196	99,6196	1:01:20:25	1:02:11:44	451.2%
5	GET	1:28	14,1196	14,1196	1,1196	99,6196	1:01:20:25	1:11:08:9	140.2%
6	HEAD	4:2	2,0159	14,1196	1,1196	99,6196	1:01:11:11	1:02:11:44	100.0%
7	GET	2	2,0159	14,1196	1,1196	100,0000	1:01:11:11	4:2:49:2	100.0%
8	GET	2	1,1196	1,1196	1,1196	100,0000	1:01:11:11	4:11	100.0%
9	GET	4	1,1196	1,1196	1,1196	100,0000	1:01:11:11	4:2	100.0%
10	GET	4	1,1196	1,1196	1,1196	100,0000	1:01:11:11	2:02:0	100.0%
11	GET	1	2,0159	1,1196	1,1196	100,0000	1:01:11:11	2	100.0%
12	GET	1	2,0159	1,1196	1,1196	100,0000	1:01:11:11	4:11	100.0%
TOTAL	CONNECT	96:024	1,146	8,0619	8,0619	100,0000	1:01:11:11	1:02:11:44	
AVERAGE	URI	4:564					8:0619	1:02:11:44	

Generated by squid-2.6.0-1.388-0502281 at 13:48:00/07-10-42

รูปที่ 414 หน้าจอแสดงรายละเอียดหลัก ของการเก็บสถิติ

Squid User Access Report
Period: 21-Aug-2007-07Feb-2007
Top 200 sites

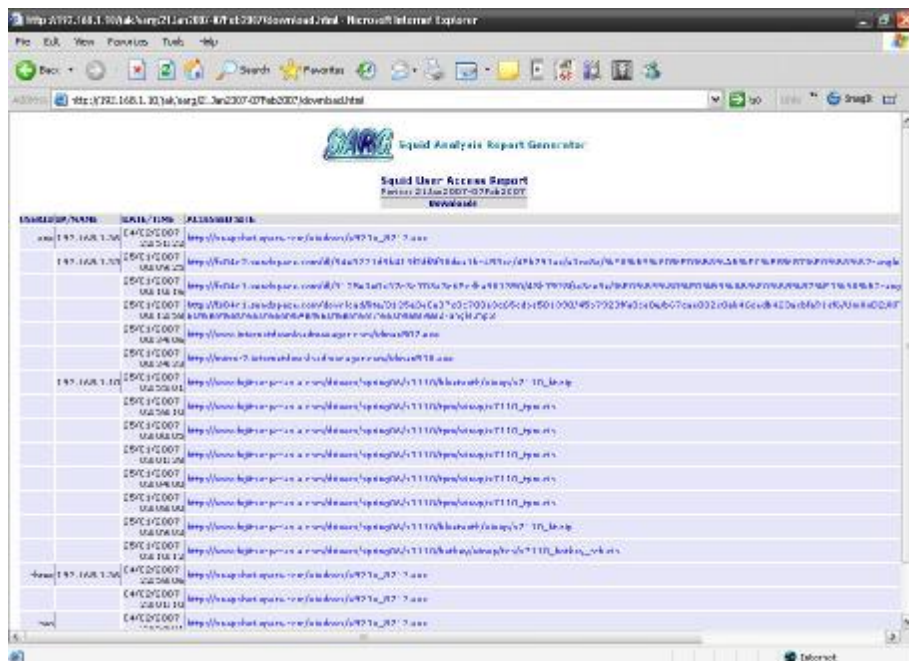
RANK	ACCESSED SITE	CONNECT ITEMS	TIME	
1	www.dyna.com	3,258	4,228	0:00:01
2	www.sca.thaiair.com	341	366,374	0:07:00
3	192.168.1.1	248	1,248	0:01:43
4	www.104.com	200	105,724	0:01:06
5	www.thairaid.com	223	19,148	0:02:19
6	www.kcp.com	192	2,188	0:00:39
7	www.thairaid.com	188	366,374	0:00:33
8	postnarcotics.com	143	1,288	0:02:50
9	ad.thairaid.com	132	136,014	0:00:00
10	www.thairaid.com	118	43,114	0:00:26
11	www.thairaid.com	83	162,734	0:02:33
12	www.thairaid.com	77	238,124	0:00:39
13	www.thairaid.com	70	120,034	0:01:05
14	www.thairaid.com	70	288,114	0:01:09
15	www.thairaid.com	67	267,554	0:00:17
16	www.thairaid.com	67	67,344	0:00:08
17	www.thairaid.com	47	240,544	0:00:19
18	www.thairaid.com	40	5,414	0:00:06
19	www.thairaid.com	40	421,114	0:00:21
20	www.thairaid.com	40	134,254	0:00:13
21	www.thairaid.com	40	202,214	0:00:48
22	www.thairaid.com	40	100,454	0:00:27
23	www.thairaid.com	32	83,034	0:00:12
24	www.thairaid.com	39	94,214	0:01:10
25	www.thairaid.com	33	71,814	0:00:13
26	www.thairaid.com	31	62,124	0:00:59
27	www.thairaid.com	31	16,734	0:00:10

รูปที่ 415 หน้าจอแสดงรายชื่อเว็บไซต์ที่ถูกใช้งานมากที่สุด

Squid User Access Report
Period: 21-Aug-2007-07Feb-2007
Sites & bytes

RANK	ACCESSED SITE	BYTES
1	192.168.1.1	1,248,000,000 bytes
2	192.168.1.1	1,248,000,000 bytes
3	192.168.1.1	1,248,000,000 bytes
4	192.168.1.1	1,248,000,000 bytes
5	192.168.1.1	1,248,000,000 bytes
6	192.168.1.1	1,248,000,000 bytes
7	192.168.1.1	1,248,000,000 bytes
8	192.168.1.1	1,248,000,000 bytes
9	192.168.1.1	1,248,000,000 bytes
10	192.168.1.1	1,248,000,000 bytes
11	www.thairaid.com	1,248,000,000 bytes
12	www.thairaid.com	1,248,000,000 bytes
13	www.thairaid.com	1,248,000,000 bytes
14	www.thairaid.com	1,248,000,000 bytes
15	www.thairaid.com	1,248,000,000 bytes
16	www.thairaid.com	1,248,000,000 bytes
17	www.thairaid.com	1,248,000,000 bytes
18	www.thairaid.com	1,248,000,000 bytes
19	www.thairaid.com	1,248,000,000 bytes
20	www.thairaid.com	1,248,000,000 bytes
21	www.thairaid.com	1,248,000,000 bytes
22	www.thairaid.com	1,248,000,000 bytes
23	www.thairaid.com	1,248,000,000 bytes
24	www.thairaid.com	1,248,000,000 bytes
25	www.thairaid.com	1,248,000,000 bytes
26	www.thairaid.com	1,248,000,000 bytes
27	www.thairaid.com	1,248,000,000 bytes

รูปที่ 416 หน้าจอแสดงรายชื่อเว็บไซต์และผู้ที่ใช้ใช้งาน



รูปที่ 417 หน้าจอแสดงแสดงรายการที่เคย Download



รูปที่ 418 หน้าจอแสดงรายชื่อของเว็บไซต์ที่มีปัญหาไม่สามารถเข้าได้

http://192.168.1.30/squid21Jan07-07Feb07/AccessStats.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Dec 11:30 Search Favorites

Address http://192.168.1.30/squid21Jan07-07Feb07/AccessStats.html

Squid Analysis Report Generator

Squid User Access Report
 Period: 21Jan2007-07Feb2007
 Date: 07Feb
 User: root@192.168.1.30
 User Report

ACCESSED SITE	REQUESTS	BYTES	%BYTES	IN CACHE	OUT	ELAPSED TIME	KB/SEC	%TMR
http://www.google.com	20	5,340	74.93%	0.00%	19.91%	00:03:14	194.126	2.28%
http://www.google.co.uk	332	1,330	14.92%	3.84%	14.16%	11:39:22	3,762,399	49.97%
http://www.google.com	2	360,030	5.42%	0.00%	0.00%	00:02:29	14,943	1.73%
http://www.google.com	180	316,790	1.73%	11.14%	88.86%	00:00:34	34,221	0.46%
http://www.google.com	40	72,390	1.07%	0.00%	0.00%	00:10:21	701,328	10.82%
http://www.google.com	3	49,120	0.73%	0.44%	19.36%	00:00:22	22,884	0.48%
http://www.google.com	10	10,410	0.23%	0.00%	0.00%	00:00:09	9,000	0.11%
http://www.google.com	3	11,980	0.18%	3.62%	16.38%	00:00:07	7,196	0.09%
http://www.google.com	28	9,720	0.14%	0.00%	0.00%	00:10:10	913,411	10.74%
http://192.168.1.1	2	9,420	0.14%	0.00%	0.00%	00:00:00	316	0.00%
http://www.google.com	13	5,740	0.09%	39.12%	60.87%	00:00:00	89	0.00%
http://www.google.com	2	3,810	0.05%	0.00%	0.00%	00:00:10	13,129	0.17%
http://www.google.com	1	3,740	0.05%	0.00%	0.00%	00:00:06	4,113	0.07%
http://www.google.com	4	3,120	0.03%	23.28%	74.74%	00:00:10	10,221	0.14%
http://www.google.com	2	2,230	0.03%	100.00%	0.00%	00:00:10	13,140	0.16%
http://www.google.com	1	1,480	0.02%	0.00%	0.00%	00:00:22	20,820	0.42%
http://www.google.com	4	1,410	0.02%	0.00%	0.00%	00:00:10	12,321	0.17%
http://www.google.com	4	1,410	0.02%	100.00%	0.00%	00:00:00	490	0.01%
http://www.google.com	4	1,410	0.02%	100.00%	0.00%	00:00:00	391	0.01%
http://www.google.com	4	1,280	0.02%	0.00%	0.00%	00:01:10	72,381	0.86%
http://www.google.com	1	1,180	0.02%	0.00%	0.00%	00:00:10	10,079	0.13%
http://www.google.com	2	870	0.01%	0.00%	0.00%	00:00:00	244	0.00%
http://www.google.com	3	830	0.01%	0.00%	0.00%	00:00:00	863	0.01%
http://www.google.com	1	820	0.01%	0.00%	0.00%	00:00:00	129	0.00%
http://www.google.com	1	810	0.01%	0.00%	0.00%	00:00:00	7,866	0.07%

Internet

รูปที่ 419 หน้าจอแสดงรายการทั้งหมดของผู้ใช้งาน

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ระบบที่พัฒนาขึ้นมาสามารถลดปัญหาการใช้งาน แบนด์วิดท์ ได้เป็นอย่างดีเนื่องจากโปรแกรม **Squid** มีลักษณะการทำงานเป็นแบบ **proxy cache server** ส่งผลดีให้ระบบด้วยการเก็บเว็บไซต์ที่เคยไปเยี่ยมชมมาแล้วไว้ใน **hard drive** หากมีเครื่องในเครือข่ายที่เรียกเว็บไซต์เดิมอีกจะมาถึงขอมูลจาก **hard drive** ก่อน ทำให้ไม่มีการใช้ แบนด์วิดท์ สำหรับการเรียกเว็บไซต์เดิมๆ อีกทั้งยังมีระบบตรวจสอบ **Username** และ **Password** ก่อนเข้าใช้งาน และแบ่ง ผู้ใช้งานออกเป็นกลุ่มอย่างชัดเจน รวมทั้งสามารถ **Block** เว็บไซต์ที่ไม่พึงประสงค์ได้อีกด้วย ซึ่งการพัฒนาระบบครั้งนี้ส่งผลให้ข้าพเจ้าได้รับความรู้ในการวางแผนการปฏิบัติงาน รวมถึงความรู้ด้านเครือข่าย และการใช้งาน โปรแกรม **Opensource** ต่างๆ

5.2 ข้อเสนอแนะ

1. ควรทำให้ เซิร์ฟเวอร์ เป็น **Gateway** ด้วยเพราะจากการทดสอบระบบจัดการแบนด์วิดท์สำหรับเครือข่าย **LAN** ผ่านเว็บ นั้นพบว่าถ้าต้องการให้ระบบทำงานได้อย่างเต็มประสิทธิภาพนั้นสมควรที่จะมีออกแบบระบบเครือข่ายโดยให้ เซิร์ฟเวอร์ ที่ติดตั้งระบบจัดการแบนด์วิดท์สำหรับเครือข่าย **LAN** ผ่านเว็บ นั้นทำหน้าที่เป็น **Gateway** ของระบบด้วย เพราะถ้าเราให้อุปกรณ์อื่นทำหน้าที่เป็น **Gateway** จะส่งผลให้เวลาที่มีการ **Download** ไฟล์ต่างๆ โดยใช้โปรแกรม ช่วย **Download** เช่น **Flashget** หรือ **Internet Download Manager** จะส่งผลให้การควบคุมต่างๆ ของระบบนั้นไม่สามารถทำงานได้ เพราะตัวโปรแกรมดังกล่าวจะทำการเรียก **Gateway** โดยอัตโนมัติ จึงทำให้เวลาที่ **Download** ไฟล์จะใช้ แบนด์วิดท์ อย่างสูงทำให้ผู้ใช้งานท่านอื่นใช้งาน ได้ช้าลง
2. ระบบจัดการแบนด์วิดท์สำหรับเครือข่าย **LAN** ผ่านเว็บนี้จำเป็นต้อง กรอกค่า **Proxy** และ **Port** ให้ถูกต้อง จึงจะสามารถทำงานได้ และหากมีผู้ใช้งาน ได้ปรับเปลี่ยนค่าของ **Proxy** หรือ เอาค่าของ **Proxy** ออก จะทำให้ผู้ใช้งานนั้นไม่ถูกจำกัดสิทธิการใช้งาน สำหรับวิธีการแก้ไขสามารถทำได้โดยดูรายละเอียดตามข้อ 1

3. การพัฒนาระบบ ให้ผู้ใช้งาน สามารถ แก้ไข รหัสผ่านของตัวเองได้เพื่อความสะดวกในการใช้งานของผู้ใช้งานเอง และเป็นการป้องกันการลืมหืมนัดด้วย แต่เนื่องจากระบบนี้ไม่ได้เก็บรายชื่อของผู้ใช้งานลงฐานข้อมูลทำให้ การแก้ไข รหัสผ่านของผู้ใช้งานนั้นทำได้ยาก เพราะรายชื่อของผู้ใช้งานนั้นถูกเก็บอยู่ใน ไฟล์ของ **Apache** ซึ่งเก็บเป็นแบบ **Text file** ซึ่งผู้จัดทำโครงการนี้ ยังขาดความชำนาญในการใช้งานโปรแกรม **Apache** จึงแก้ปัญหาข้างต้นนี้ด้วยการ ลบชื่อของผู้ใช้งานออกแล้ว เพิ่มใหม่โดยใช้รหัสผ่าน ตามที่ผู้ใช้ต้องการ เป็นการแก้ไขปัญหาดังกล่าวเบื้องต้น
 4. การออกแบบระบบจัดการแบนด์วิดท์สำหรับเครือข่าย **LAN** ผ่านเว็บนี้ เป็นการออกแบบโดย **fix** รูปแบบของเว็บ อย่างแน่นอนโดย **interface** ไม่สามารถทำการแก้ไข หรือเพิ่มกลุ่มของผู้ใช้งานได้ ซึ่งในการใช้งานจริง ควรมีการปรับเปลี่ยนให้สามารถแก้ไข เพิ่มหรือลบกลุ่มผู้ใช้งานได้อย่างสะดวกโดยผ่าน **interface** นี้
- นอกจากนั้นระบบจัดการแบนด์วิดท์สำหรับเครือข่าย **LAN** ผ่านเว็บ นี้ยังสามารถพัฒนาต่อได้อีกมาก ซึ่งก็ขึ้นอยู่กับความต้องการของผู้ดูแลระบบเอง อีกทั้ง ซอฟต์แวร์ ที่ใช้ในการพัฒนาระบบในครั้งนี้ก็เป็น โปรแกรมประเภท **Opensource** ทั้งสิ้นจึงทำให้ประหยัด

5.3 การประยุกต์ใช้งาน

จากการพัฒนาระบบจัดการแบนด์วิดท์สำหรับเครือข่าย **LAN** ผ่านเว็บ นั้นสามารถประยุกต์ใช้งานได้หลาย องค์กรที่ต้องการจัดสรรแบนด์วิดท์ ให้กับผู้ใช้งานในระบบเครือข่าย ภายในของตนเอง เช่น หอพัก, โรงแรม, โรงพยาบาล, **LAB** ในมหาวิทยาลัย เป็นต้น

โดยในที่นี้ขอยกตัวอย่างการประยุกต์ใช้งานระบบจัดการแบนด์วิดท์สำหรับเครือข่าย **LAN** ผ่านเว็บ ในมหาวิทยาลัยธุรกิจบัณฑิต ดังตัวอย่างต่อไปนี้

เมื่อมีชั่วโมงการเรียนการสอนในห้อง **LAB** คอมพิวเตอร์ จะพบว่านักศึกษาหลายท่านใช้อินเทอร์เน็ต ในด้านที่ไม่เหมาะสม เช่น ใช้ในการ **chat** เพื่อพูดคุย, การเข้าเว็บไซต์ที่ไม่เหมาะสม, **download** ไฟล์ที่ไม่เหมาะสม ซึ่งเป็นเรื่องยากในการป้องกันดังกล่าว หรือหากแก้ปัญหาโดยการถอดสาย **LAN** ออกจากระบบ ก็จะทำให้ให้นักศึกษาที่ต้องการศึกษาข้อมูลเพิ่มเติมจาก อินเทอร์เน็ต ไม่สามารถทำได้

ดังนั้นหากนำระบบจัดการแบนด์วิดท์สำหรับเครือข่าย **LAN** ผ่านเว็บ นี้มาใช้ก็จะเป็นทางแก้ไขที่สะดวก เพราะก่อนการเข้าใช้งาน อินเทอร์เน็ต นักศึกษาทุกคนจะต้องทำการ ใส่ **Username** และ **password** เพื่อเข้าระบบ และระบบจะทำการเก็บสถิติ การใช้งานของผู้ใช้แต่ละรายเอาไว้ อีกทั้งยัง **Block** เว็บไซต์ที่ไม่เหมาะสมได้อีกด้วย จากเหตุผลดังกล่าวจึงถือว่าการ

ประยุกต์ใช้งานที่สามารถปรับเปลี่ยนได้ตามความต้องการของผู้ดูแลระบบ อีกทั้งระบบจัดการแบนด์วิดท์สำหรับเครือข่ายLAN ผ่านเว็บนี้ยังสามารถประยุกต์ใช้งานได้อีกมากมาย ซึ่งก็ขึ้นอยู่กับความต้องการขององค์กร นั้นๆ

บรรณานุกรม

- กิตติพงษ์ สุวรรณราช. 2547. การบริหารและจัดการเครือข่ายอินเทอร์เน็ตด้วยระบบปฏิบัติการ FreeBSD. นนทบุรี: บริษัท ออฟเซ็ท เพรส จำกัด.
- กิตติพงษ์ สุวรรณราช. 2548. ระบบปฏิบัติการ FreeBSD และการประยุกต์ใช้งาน. นนทบุรี: บริษัท ออฟเซ็ท เพรส จำกัด.
- กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2546. UML วิเคราะห์และออกแบบระบบเชิงวัตถุ. กรุงเทพฯ: เลทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด.
- กิตติศักดิ์ เจริญโกถานนท์. 2548. คู่มือเรียนเขียนเว็บอีคอมเมิร์ซด้วย PHP5. กรุงเทพฯ: บริษัท ซัคเซส มีเดีย จำกัด.
- กิตติ ภัคดีวัฒนะกุล. 2547. คัมภีร์ PHP. กรุงเทพฯ: บริษัท เลทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด
- บัณฑิต จามรฤติ. 2544. คู่มือระบบยูนิกซ์ FreeBSD เล่ม1. กรุงเทพฯ: บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน).
- บัณฑิต จามรฤติ. 2549. คู่มือระบบยูนิกซ์ FreeBSD เล่ม2. กรุงเทพฯ: บริษัท บัณฑิตเพรส จำกัด
- วิกิพีเดีย. 2006. FreeBSD. [Online] Available: <http://th.wikipedia.org/wiki/FreeBSD>.

ประวัติผู้แต่ง

ชื่อ - นามสกุล	นายจักรพงษ์ ศรีประสม
วัน / เดือน / ปีเกิด	29 ตุลาคม 2527
สถานที่เกิด	นครสวรรค์
ประวัติการศึกษา	มัธยมศึกษา โรงเรียนตากดีประชาสรรค์ วิทย์-คณิต อุดมศึกษา มหาวิทยาลัยธุรกิจบัณฑิตย์ คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ