

# Software ช่วยในการบริหารจัดการระบบ Proxy / Cache

## Management Software for Proxy / Cache



วัน เดือน ปี.....	25 S.A. 2549
เลขทะเบียน.....	01694
เลขเรียกหนังสือ.....	จท. ๗583 ๕ ๒549
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจธ."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	Software ช่วยในการบริหารจัดการระบบ Proxy / Cache
นักศึกษา	นายสันติ สิทธิเลิศพิศาล
อาจารย์ที่ปรึกษา	อาจารย์อัครินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

### บทคัดย่อ

โปรแกรม Squid เป็นโปรแกรม Proxy/Cache ที่ได้รับความนิยมมากที่สุด โปรแกรมหนึ่งซึ่งมีประสิทธิภาพในการทำงานสูง โครงการพัฒนาระบบงานนี้ได้ทำการศึกษาการทำงานของโปรแกรม วิธีการปรับเปลี่ยนค่าพารามิเตอร์ และวิธีการตรวจสอบประสิทธิภาพการทำงาน พบว่าการบริหารจัดการทั้งระบบมีความยุ่งยากโดยเฉพาะการแก้ไข Configuration File ซึ่งมีพารามิเตอร์จำนวนมากและมีความซับซ้อน จึงได้ทำการออกแบบและพัฒนาระบบงานที่ช่วยในการบริหารจัดการ Proxy/Cache ได้โดยง่ายจากส่วนกลาง โดยมีการทำงาน 3 ส่วนด้วยกัน คือการสั่งงานโปรแกรม Squid จาก System Command การแก้ไข Configuration File และการตรวจสอบประสิทธิภาพการทำงาน โดยในการสั่งงานโปรแกรมและการแก้ไข Configuration File จะอาศัยโปรแกรม Agent ซึ่งพัฒนาด้วย GNU C++ ทำงานที่ Proxy/Cache เพื่อคอยรับคำสั่งจากส่วนกลางแล้วทำงาน และมีโปรแกรมส่วนกลางซึ่งพัฒนาด้วย Microsoft Visual Basic คอยควบคุม ส่วนการตรวจสอบประสิทธิภาพการทำงานจะตรวจสอบด้วย SNMP โปรโตคอลโดยใช้โปรแกรม MRTG เป็นผลให้ผู้บริหารจัดการระบบสามารถบริหารจัดการ Proxy/Cache ได้โดยง่ายจากส่วนกลาง โดยที่ผู้บริหารไม่จำเป็นต้องมีความรู้ความชำนาญมากนัก

<b>Title</b>	Management Software for Proxy / Cache
<b>Student</b>	Mr. Santi Sittilerdpisan
<b>Advisor</b>	Mr. Akharin Khunkitti
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	2000

## ABSTRACT

Squid, one of the popular Proxy/Cache software, has very high working performance. This project study its working mechanism, method of changing the working parameter and checking its performance. System management is very complicate, especially in parameter changing of the configuration file. This project covers design and development of an application software for Squid central management. The application software works in 3 parts; Squid controlled via system command, configuration file controlling, and working performance checking. Proxy/Cache controlled and configuration file changing using agent program, developed by GNU C++, run on each Proxy/Cache client that waiting for a command controlled from the central software, which is an application program developed by Microsoft Visual Basic. Working performance is checked by SNMP protocol using program called MRTG. This results the system administrator can manage Proxy/Cache easily from the center with less experience.

## กิตติกรรมประกาศ

งานโครงการพัฒนาระบบงานบริหารจัดการระบบ Proxy/Cache ที่ได้จัดทำขึ้นนี้ ข้าพเจ้าได้รับการสนับสนุนจากบุคคลหลายฝ่าย ทำให้โครงการนี้สามารถเสร็จสิ้นลงไปได้ด้วยดี ข้าพเจ้าจึงขอขอบพระคุณบุคคลดังต่อไปนี้

1. บิดาและมารดา ที่คอยให้ความช่วยเหลือ คอยดูแลและให้กำลังใจ
2. อาจารย์อัครินทร์ คุณกิตติ อาจารย์ที่ปรึกษาโครงการ ที่ให้คำปรึกษาแนะนำและชี้แนะแนวทางต่างๆ เป็นอย่างดี
3. ศูนย์สารสนเทศ กรมโรงงานอุตสาหกรรม ที่ให้การสนับสนุนอุปกรณ์ในการจัดทำรายงานและอุปกรณ์ในการพัฒนาโปรแกรม รวมทั้งยินยอมให้ใช้เครื่อง Proxy/Cache ทั้งที่มีการใช้งานอยู่
4. คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และอาจารย์ทุกๆ ท่าน ที่ได้มอบความรู้ทางวิชาการต่างๆ เพื่อนำมาใช้พัฒนา
5. เพื่อนๆ รุ่น IS6(ภาคสมทบ) ที่คอยให้คำปรึกษาด้านการเขียนโปรแกรม
6. Webboard Pantip.com ที่เปิด โอกาสให้มีการถาม-ตอบปัญหา เกี่ยวกับโปรแกรม

สันติ สิทธิเลิศพิศาล

ผู้จัดทำ

# สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ .....	III
สารบัญ .....	IV
สารบัญตาราง .....	VI
สารบัญภาพ .....	VII
บทที่	
1. บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ของการพัฒนาระบบงาน .....	1
1.3 เป้าหมายของการพัฒนาระบบงาน .....	2
1.4 ขอบเขตของการศึกษาและพัฒนาระบบงาน .....	2
1.5 ขั้นตอนการศึกษาและพัฒนาระบบ .....	3
1.6 รายละเอียดของเครื่องคอมพิวเตอร์และเครื่องมือที่ใช้ในการพัฒนาระบบ .....	3
1.7 รายละเอียดของแต่ละบท .....	4
2. Proxy/Cache และทฤษฎีที่เกี่ยวข้อง	
2.1 ความหมายของ Proxy/Cache .....	6
2.2 การตรวจสอบความล้าสมัยของข้อมูลใน Cache .....	7
2.3 ประวัติความเป็นมาของ โปรแกรม Squid Proxy/Cache .....	8
2.4 หลักการทำงานของ โปรแกรม Squid Proxy/Cache .....	9
2.5 การติดตั้งโปรแกรม Squid .....	11
2.6 การ Configuration Parameter .....	12
2.7 การตรวจสอบประสิทธิภาพการทำงานของ โปรแกรม Squid .....	13
2.8 สรุปบท .....	15
3. การวิเคราะห์และออกแบบ	
3.1 ปัญหาของระบบในปัจจุบัน .....	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ความต้องการของระบบ .....	16
3.3 การออกแบบระบบการทำงาน .....	17
3.4 ส่วนประกอบของ โปรแกรม .....	18
3.5 กระบวนการทำงานของระบบ .....	21
3.6 ข้อกำหนด(Request-Reply)ในการติดต่อกับ Agent .....	24
3.7 การสร้าง Configuration Files ของโปรแกรม Mrtg .....	26
3.8 รูปแบบเพิ่มข้อมูลของโปรแกรม .....	27
3.9 ส่วนติดต่อผู้ใช้งาน(User Interface) .....	30
4. การพัฒนาระบบงาน	
4.1 การพัฒนาโปรแกรม Agent .....	34
4.2 การติดตั้งเครื่องมือ(Tools) ต่างๆ ที่ใช้ .....	35
4.3 การพัฒนาโปรแกรม Squid Admin .....	35
4.4 วิธีการทำงานกับโปรแกรม .....	36
4.5 ทดสอบการใช้งานกับระบบจริง .....	47
5. สรุปผลการค้นคว้าและพัฒนาระบบ	
5.1 ผลจากการพัฒนาระบบ .....	49
5.2 ประโยชน์ที่ได้รับ .....	49
5.3 อุปสรรคในการพัฒนาระบบ .....	50
5.4 ข้อเสนอแนะ .....	50
บรรณานุกรม .....	51
ภาคผนวก	
ภาคผนวก ก. ตัวอย่าง Configuration file ของโปรแกรม MRTG .....	53
ภาคผนวก ข. ตัวอย่างเพิ่มเก็บข้อมูลระบบ .....	56
ภาคผนวก ค. Source Code ของโปรแกรม Agent (Agent.C) .....	57
ภาคผนวก ง. วิธีการติดตั้งโปรแกรม .....	60
ภาคผนวก จ. คู่มือการใช้งานโปรแกรม .....	62
ประวัติผู้เขียน .....	70

# สารบัญตาราง

หน้า

ตารางที่

3.1 ค่าประสิทธิภาพการทำงานโดยโพรโตคอล SNMP .....	19
3.2 ความหมายของ Tag ในแฟ้มเก็บข้อมูลระบบ .....	28
3.3 การเปลี่ยนแปลงแฟ้มต้นแบบเมื่อนำไปใช้ .....	30
4.1 ความหมายใน Configuration File สำหรับ Agent .....	34
4.2 การแก้ไขโปรแกรม Mrtg .....	35



# สารบัญภาพ

หน้า

ภาพที่

2.1	การทำงานของ Proxy/Cache .....	9
2.2	Parent และ Sibling Cache .....	10
2.3	การใช้ ICP Packet .....	11
3.1	ภาพการเชื่อมต่อเครือข่ายของระบบ .....	17
3.2	ส่วนประกอบของโปรแกรมช่วยบริหารจัดการระบบ Proxy/Cache .....	18
3.3	แผนภาพ Context Diagram ของระบบ .....	21
3.4	แผนภาพแสดงการไหลของข้อมูลระดับที่ 1 (DFD Level 1) .....	22
3.5	แผนภาพแสดงการไหลของข้อมูลระดับที่ 2 (DFD Level 2) ของกระบวนการที่ 1 .....	23
3.6	แผนภาพแสดงการไหลของข้อมูลระดับที่ 2 (DFD Level 2) ของกระบวนการที่ 2 .....	23
3.7	แผนภาพแสดงการไหลของข้อมูลระดับที่ 2 (DFD Level 2) ของกระบวนการที่ 3 .....	24
3.8	แผนภาพแสดงขั้นตอนการติดต่อกับ Agent .....	25
3.9	โครงสร้างการเก็บข้อมูลของระบบ .....	27
3.10	หน้าจอหลักของโปรแกรม .....	30
3.11	สถานะของข้อมูล Configuration ที่ Management Station .....	31
3.12	สถานะของการทำงานของ Proxy/Cache .....	31
3.13	แถบเครื่องมือของหน้าจอหลัก .....	32
3.14	หน้าจอแสดงภาพความสัมพันธ์ .....	33
4.1	การเพิ่ม Component ต่างๆ เข้ามาใน Project .....	36
4.2	หน้าจอการสร้าง Project ใหม่ .....	37
4.3	หน้าจอการเพิ่ม Proxy/Cache เข้ามาใน โปรแกรม .....	38
4.4	หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลทั่วไป .....	39
4.5	หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลรายละเอียด .....	39
4.6	หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลความสัมพันธ์ .....	40
4.7	หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลชื่อ Acl .....	40
4.8	หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูล Access Control .....	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9	หน้าจอตัวเลือกของคำสั่ง Squid Command .....	41
4.10	หน้าจอการกำหนดค่าต่างๆ ของโปรแกรม Mrtg .....	42
4.11	หน้าจอแสดงประสิทธิภาพการทำงานปัจจุบันของ Proxy/Cache .....	43
4.12	หน้าจอแรกของการใช้ Browser เรียกดูข้อมูลจากโปรแกรม Mrtg .....	43
4.13	หน้าจอรายละเอียดของประสิทธิภาพในแต่ละค่า .....	44
4.14	หน้าจอแสดงภาพความสัมพันธ์ของ Proxy/Cache .....	45
4.15	หน้าจอป้อนข้อมูลความสัมพันธ์ระหว่าง Proxy to Proxy .....	45
4.16	หน้าจอป้อนข้อมูลขอบเขตของ IP Address .....	46
4.17	หน้าจอแสดงความสัมพันธ์ที่สร้างขึ้นทั้งหมด .....	46
4.18	หน้าจอแสดงการเปลี่ยนแปลงสีของภาพความสัมพันธ์ .....	47
4.19	ระบบ Proxy/Cache ของกรมโรงงานอุตสาหกรรม .....	47



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากการได้มีการนำเทคโนโลยี Proxy/Cache เข้ามาแก้ปัญหาค่าการสื่อสารที่คับคั่งในเครือข่าย Internet โดยการสำเนาข้อมูลที่มีการเรียกใช้จากภายนอกมาเก็บไว้ใน Cache มีการนำ Proxy/Cache หลายๆ เครื่องเข้ามาทำงานร่วมกัน แบ่งปันข้อมูลและทรัพยากรซึ่งกันและกัน ช่วยลดเวลาในการเข้าถึงข้อมูลและลดการใช้ Bandwidth จำนวนมากได้ ในปัจจุบัน Software ที่เป็นที่ยอมรับและใช้กันอย่างกว้างขวางมากที่สุดโปรแกรมหนึ่งก็คือ โปรแกรม Squid ซึ่งสามารถนำมาใช้โดยง่ายและไม่มีค่าใช้จ่ายใดๆ (Freeware) ซ้ำยังมีประสิทธิภาพในการทำงานสูง ส่วนการ Install ไม่ยุ่งยากมากนัก ดังนั้นจึงมีการใช้ Software นี้กับ Proxy/Cache อย่างแพร่หลาย ตามเครือข่ายขององค์กรต่างๆ ทั้งขนาดเล็กและขนาดใหญ่ แต่ Software นี้ยังมีข้อเสียอยู่ก็คือ การ Configuration และบริหารจัดการระบบ Proxy/Cache ทำได้ด้วยความยากลำบาก เพราะมีตัวแปรจำนวนมากที่จะต้องทำการปรับแต่งและยังต้องจัดการกับ Proxy/Cache หลายๆ เครื่องให้มีการทำงานสัมพันธ์ซึ่งกันและกัน ซึ่งหากการจัดการไม่ดีพอก็จะทำให้ประสิทธิภาพในการทำงานลดลง หรืออาจจะไม่ทำงานตามที่ผู้ใช้ต้องการ

การจัดการโปรแกรม Squid ในเครื่องๆ หนึ่ง จะต้องมีการสั่งงานโปรแกรม และมีการปรับปรุง Configuration ซึ่งประกอบไปด้วยตัวแปรจำนวนมาก ซึ่งผู้ที่ไม่มีความรู้หรือความชำนาญเพียงพอจะไม่สามารถจะทำการปรับปรุง Configuration ได้ เมื่อมีการเปลี่ยนแปลงใดๆ ในระบบเครือข่ายหรือเมื่อมีข้อผิดพลาดเกิดขึ้น

### 1.2 วัตถุประสงค์ของการพัฒนาระบบงาน

- 1.2.1 ช่วยอำนวยความสะดวกแก่ผู้บริหารจัดการระบบ Squid Proxy/Cache ในสั่งงานและแก้ไขปรับปรุง Configuration ของโปรแกรม ได้โดยไม่ต้องมีความรู้ความชำนาญมากนัก
- 1.2.2 สามารถตรวจสอบประสิทธิภาพการทำงานของ Squid Proxy/Cache ในแต่ละเครื่อง หรือนำมาเปรียบเทียบกันได้ ทั้งในขณะปัจจุบันและการทำงานที่ผ่านมา
- 1.2.3 สามารถบริหารจัดการระบบ Squid Proxy/Cache ทั้งหมดจากส่วนกลางที่จุดเดียว

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.4 ทำให้ Configuration ของ Squid Proxy/Cache ทั้งระบบมีการทำงานที่สัมพันธ์กัน

1.2.5 ช่วยเพิ่มประสิทธิภาพและคุณภาพของการให้บริการ

### 1.3 เป้าหมายของการพัฒนาระบบงาน

โครงการพัฒนาระบบงานนี้มีเป้าหมายเพื่ออำนวยความสะดวกแก่ผู้บริหารระบบ Proxy/Cache ในการบริหารจัดการโปรแกรม Squid ทั้งระบบได้โดยง่ายจากส่วนกลาง เช่นการสั่งงานโปรแกรม การแก้ไขปรับปรุง Configuration หรือการตรวจสอบประสิทธิภาพการทำงาน โดยที่ผู้บริหารไม่จำเป็นต้องมีความรู้ความชำนาญมากนัก

### 1.4 ขอบเขตของการศึกษาและพัฒนาระบบงาน

โครงการพัฒนาระบบงาน Software ช่วยในการบริหารจัดการระบบ Proxy/Cache นี้จะทำการศึกษาและพัฒนาระบบงานสำหรับโปรแกรม Squid ซึ่งเป็นโปรแกรม Proxy/Cache ที่ได้รับความนิยมมากที่สุดโปรแกรมหนึ่ง โดยการศึกษาจะมุ่งเน้นในเรื่องวิธีการปรับแต่งโปรแกรม Squid ในส่วนที่มีผลต่อประสิทธิภาพ การทำงานร่วมกันระหว่าง Proxy/Cache ในเครือข่ายและการตรวจสอบประสิทธิภาพในการทำงาน

ส่วนการพัฒนาโปรแกรมจะแบ่งออกเป็น 3 ส่วนด้วยกัน คือการสั่งงานโปรแกรม Squid จาก System Command การแก้ไข Configuration File และการตรวจสอบประสิทธิภาพการทำงาน โดยในการสั่งงานโปรแกรมและการแก้ไข Configuration File จะใช้วิธีการสั่งงานจากส่วนกลางและอาศัยโปรแกรม Agent ซึ่งมีการทำงานที่ฝั่ง Proxy/Cache ที่อยู่ในระบบทั้งหมด คอยรับคำสั่งแล้วส่งต่อการทำงานหรือทำการแก้ไข Configuration File ส่วนการตรวจสอบประสิทธิภาพการทำงานจะตรวจสอบด้วย SNMP โพรโตคอลผ่านทาง MIB Object ของโปรแกรม Squid โดยตรง

### 1.5 ขั้นตอนการศึกษาและพัฒนาระบบ

ได้กำหนดขั้นตอนในการศึกษาและการพัฒนาโปรแกรมไว้ดังต่อไปนี้

- ขั้นตอนที่ 1 ศึกษาการทำงานของระบบ Proxy/Cache โดยทั่วไป และศึกษาเรื่องราวเกี่ยวกับโปรแกรม Squid Proxy/Cache เพื่อที่จะนำมาใช้ในการออกแบบโปรแกรม
- ขั้นตอนที่ 2 ศึกษาการใช้เครื่องมือ(Tools) ต่างๆ ที่จำเป็น เพื่อนำมาประกอบการทำงานและช่วยพัฒนาโปรแกรม

ขั้นตอนที่ 3 วิเคราะห์และออกแบบการทำงานของโปรแกรม จากที่ได้ทำการศึกษาในขั้นตอนที่ผ่านมา เพื่อใช้ในการพัฒนาโปรแกรมต่อไป ภายใต้ขอบเขตของการศึกษาและพัฒนาระบบงานที่ได้กำหนดไว้

ขั้นตอนที่ 4 ทำการพัฒนา และทดสอบโปรแกรมในแต่ละส่วน(Module) ตามที่ได้ออกแบบไว้

ขั้นตอนที่ 5 ทดสอบ ตรวจสอบข้อผิดพลาดโดยรวม แล้วนำมาปรับแต่งเพื่อพร้อมจะนำไปใช้งานจริงต่อไป

## 1.6 รายละเอียดของเครื่องคอมพิวเตอร์และเครื่องมือที่ใช้ในการพัฒนาระบบ

เนื่องจากโปรแกรมระบบงานที่พัฒนานี้เป็นโปรแกรมบริหารจัดการเครื่อง Proxy/Cache ซึ่งใช้กับเครื่อง Proxy/Cache ได้หลายๆ เครื่องพร้อมๆกัน ดังนั้นเพื่อให้มีสภาพเหมือนกับการใช้งานจริงที่สุด จึงได้พัฒนาระบบงานนี้ที่หน่วยงานที่ทำงานอยู่ปัจจุบัน คือกรมโรงงานอุตสาหกรรม ซึ่งได้มีการใช้งาน Squid Proxy/Cache อยู่แล้ว 2 เครื่อง โดยได้ทำการจัดหาเครื่องบริหารจัดการ Proxy/Cache จากส่วนกลางเพิ่มขึ้นมาอีกเครื่องหนึ่ง ดังนั้นการพัฒนาระบบจึงใช้เครื่องคอมพิวเตอร์และเครื่องมือในการพัฒนาดังต่อไปนี้

### 1.6.1 เครื่องบริหารจัดการ Proxy/Cache จากส่วนกลาง (Management Station)

#### 1.6.1.1 คุณสมบัติด้าน Hardware

CPU	:	Pentium MMX 233 MHz
RAM	:	64 MB
Hard-Disk	:	8.2 GB
NIC	:	Novell NE2000 Compatible

#### 1.6.1.2 ระบบปฏิบัติการ (Operating System)

Windows NT 4.0 Thai Enable (Services Pack 6)

#### 1.6.1.3 ระบบเว็บเซิร์ฟเวอร์

IIS (Internet Information Server) Version 4

### 1.6.2 เครื่อง Proxy/Cache Server เครื่องที่ 1

#### 1.6.2.1 คุณสมบัติด้าน Hardware

CPU	:	Pentium MMX 233 MHz
RAM	:	64 MB
Hard-Disk	:	4.3 GB
NIC	:	Intel Pro 10/100 Mbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.6.2.2 ระบบปฏิบัติการ (Operating System)

Linux RedHat 6.0

### 1.6.2.3 Proxy Software

Squid Proxy/Cache Version 2.2.STABLE5

## 1.6.3 เครื่อง Proxy/Cache Server เครื่องที่ 2

### 1.6.3.1 คุณสมบัติด้าน Hardware

CPU : Pentium 100 MHz  
 RAM : 24 MB  
 Hard-Disk : 4.3 GB  
 NIC : Intel Pro 10/100 Mbps

### 1.6.3.2 ระบบปฏิบัติการ (Operating System)

Linux RedHat 6.0

### 1.6.3.3 Proxy Software

Squid Proxy/Cache Version 2.2.STABLE5

## 1.6.4 ระบบเครือข่ายสื่อสาร (Network)

Internet/Intranet Network ภายในองค์กรซึ่งอยู่ภายใน Local Area Network เดียวกัน (Fast Ethernet 10/100 UTP Cat 5) สามารถเชื่อมต่อออกไปยัง Internet ได้

## 1.6.5 เครื่องมือที่ใช้พัฒนา (Development Tools)

Microsoft Visual Basic 6.0 (Service Pack 3)

GNU C++ for Linux Version egcs-2.91.66

## 1.6.6 โปรแกรมสำเร็จรูปที่นำมาใช้ประกอบการทำงาน

Active Perl Version 5.005\_3 Build 522

MRTG (Multi Router Traffic Grapher) Version 2.8.12

Internet Explorer Version 4.0

Dart Snmp Tool Version 2.0

## 1.7 รายละเอียดของแต่ละบท

สำหรับเนื้อหาของเอกสารประกอบการพัฒนาระบบถัดไปจากบทนี้ จะมีการแบ่งเนื้อหาหลักๆ ออกเป็น 5 ส่วน โดยเริ่มจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 จะกล่าวถึง Proxy/Cache และทฤษฎีที่เกี่ยวข้อง ความหมายของ Proxy/Cache และโปรแกรม Squid ว่ามีการทำงานอย่างไร ทำไมจึงช่วยแก้ปัญหาการติดต่อสื่อสารในเครือข่ายอินเทอร์เน็ตที่มีความคับคั่งได้ และจะทำการบริหารจัดการได้อย่างไร

บทที่ 3 เป็นการวิเคราะห์และออกแบบ จากปัญหาและความต้องการของระบบ จนได้ส่วนประกอบของโปรแกรม วิธีการเก็บข้อมูลและออกแบบ User Interface

ต่อไป บทที่ 4 เป็นขั้นตอนการพัฒนาระบบงานตามที่ได้ออกแบบไว้ในบทที่ 3 และวิธีการทำงานกับโปรแกรม

ส่วนบทที่ 5 จะเป็นการสรุปผลการพัฒนาระบบงาน ประโยชน์ที่ได้รับ อุปสรรคของการพัฒนา และข้อเสนอแนะเพื่อเป็นประโยชน์ในการทำงาน



## บทที่ 2

### Proxy/Cache และทฤษฎีที่เกี่ยวข้อง

#### 2.1 ความหมายของ Proxy/Cache

ปัจจุบันเครือข่ายคอมพิวเตอร์ส่วนใหญ่มีการใช้โปรโตคอล TCP/IP ในการสื่อสารระหว่างเครื่องด้วยกัน และมักจะมีการเชื่อมต่อออกไปยังเครือข่ายอินเทอร์เน็ตอีกด้วย ซึ่งปัจจุบันในเครือข่ายอินเทอร์เน็ตมีคอมพิวเตอร์ที่เข้ามาเชื่อมต่อเพิ่มขึ้นอย่างมากมาและรวดเร็ว ซ้ำยังมีแนวโน้มที่จะมากขึ้นเป็นทวีคูณ ทำให้ปริมาณข้อมูลที่ผ่านเข้าออกในเครือข่ายมีจำนวนสูงขึ้นเรื่อยๆ เครือข่ายที่มีอยู่เดิมต้องแบกรับภาระในการบริการข้อมูลจำนวนมาก เกิดความล่าช้าหรือมีการสูญหายของข้อมูลเกิดขึ้นบ่อยครั้ง ถึงแม้ในปัจจุบันเทคโนโลยีของสายนำสัญญาณจะพัฒนาขึ้นจากเดิมมาก กล่าวคือสามารถที่จะส่งข้อมูลได้ด้วยความเร็วที่สูงขึ้นกว่าเดิมแล้วก็ตาม แต่ก็ยังไม่เพียงพอต่อความต้องการใช้ยู่ดี

มีความพยายามมากมายที่จะหาทางแก้ปัญหาที่เกิดขึ้นนี้ แนวความคิดหนึ่งที่น่าสนใจคือ การนำข้อมูลที่ต้องใช้บ่อย ๆ มาเก็บไว้ให้ใกล้ผู้ใช้งานที่สุด เพื่อลดปริมาณการติดต่อกับเครือข่ายภายนอก ทำให้เครือข่ายภายนอกมีความคล่องตัวสูงขึ้น อีกแนวความคิดหนึ่งที่มีการนำมาประยุกต์จนสามารถใช้งานได้จริงคือ การสำเนาข้อมูลที่มีการเรียกใช้จากภายนอกมาเก็บไว้ใน Cache เมื่อมีความต้องการใช้ข้อมูลเดิมอีกครั้งในภายหลัง จะนำสำเนาที่เก็บไว้ใน Cache ส่งให้แทน การทำงานเช่นนี้เหมาะสำหรับการติดต่อที่ไม่ต้องการการโต้ตอบแบบทันทีทันใด เช่น HTTP, FTP หรือ GOPHER การทำงานในลักษณะเช่นนี้จำเป็นต้องอาศัยตัวกลางเชื่อมต่อระหว่าง Client และ Host Server ซึ่งจะเรียกว่า Proxy

เมื่อ Client ต้องการดึงข้อมูลจากเครือข่ายภายนอก จะมีการติดต่อไปยัง Proxy/Cache ก่อน เพื่อตรวจสอบว่ามีสำเนาข้อมูลเก็บไว้ใน Cache หรือไม่ ถ้ามีก็จะส่งข้อมูลนั้นกลับไปยัง Client ที่ติดต่อมา แต่ถ้าไม่มี Proxy/Cache ก็จะติดต่อไปยัง Web Site ดันฉบับ แล้วดึงข้อมูลมาทำสำเนาเก็บไว้ เพื่อประโยชน์ในการร้องขอข้อมูลครั้งต่อไป แล้วก็ส่งต่อไปยัง Client ที่ติดต่อมาอีกทีหนึ่ง แต่ระบบนี้ยังมีข้อเสียอยู่คือบางครั้งข้อมูลที่ได้อาจมีความล้าสมัย ดังนั้นจึงต้องมีกระบวนการในการตรวจสอบด้วย

## 2.2 การตรวจสอบความล้าสมัยของข้อมูลใน Cache

เมื่อไรก็ตามที่มีการทำสำเนาจากข้อมูลต้นฉบับก็จะมีปัญหาในการจัดการ ถ้าข้อมูลต้นฉบับมีการเปลี่ยนแปลงหลังจากที่ได้มีการทำสำเนาเก็บไว้ มันจะกลายมาเป็นข้อมูลที่พ้นสมัย(out of date) ในทันที(เรียกกันว่าปัญหา Consistency) ความล้าสมัยของสำเนาข้อมูลถูกกำหนดเป็นช่วงเวลาตั้งแต่ข้อมูลต้นฉบับเปลี่ยนไป และมักจะแปรผันไปตามสัดส่วนของอายุของข้อมูล

เป็นไปได้ยากที่จะติดต่อกับที่มาว่าสำเนาข้อมูลนี้ล้าสมัยแล้วหรือยัง ถ้าให้ระบบเครือข่าย และ Host ที่ให้บริการไม่มีการเปลี่ยนแปลงใดๆ เนื่องจาก Hypertext Transfer Protocol ในรุ่นก่อนๆ มีเพียงแต่การร้องขอแบบ 'GET' เท่านั้น ซึ่งจะได้รับข้อมูลมาทั้งหมด เหมือนกับขบวนการร้องขอข้อมูลแล้วนำมาเปรียบเทียบกับสำเนา

การปรับปรุงสิ่งแรกก็คือเพิ่มการร้องขอแบบ 'HEAD' ซึ่งจะมีการส่งข้อมูลสรุปเกี่ยวกับข้อมูลที่ร้องขอเช่นเวลาที่เปลี่ยนแปลงครั้งล่าสุด ถ้าข้อมูลนั้นแสดงให้เห็นว่าสำเนามีความล้าสมัยมันจะต้องสร้าง Connection ไปยังเครื่องต้นฉบับเพิ่มขึ้นอีกเพื่อร้องขอข้อมูลแบบ 'GET' ข้อดีของนี้ทำให้มีการนำการร้องขอในแบบมีเงื่อนไขมาใช้ ซึ่งเรียกกันว่า 'If-Modified-Since GET' หรือ 'IMS GET' การร้องขอแบบนี้จะมีการนำเวลาของสำเนาข้อมูลใส่ไว้ในการร้องขอด้วย แล้วคาดหวังว่าการตอบกลับจะเป็นเพียง 'unchanged' หรือการส่งข้อมูลมาใหม่ ดังนั้นทั้งการตรวจสอบและการแก้ไขข้อมูลสามารถทำได้ในการติดต่อเพียง Connection เดียวเท่านั้น ปัจจุบัน Web Server โดยส่วนใหญ่สนับสนุนการร้องขอแบบ 'IMS GET' แต่การตรวจสอบแบบนี้ยังคงต้องมีการติดต่อไปยัง Host ที่เป็นต้นฉบับ ดังนั้นจึงอาจจะต้องใช้เวลานานสำหรับ Site ที่อยู่ในระยะไกล แต่มันจะช่วยลดปริมาณความต้องการใช้ระบบเครือข่ายเมื่อข้อมูลไม่มีการเปลี่ยนแปลง

อย่างไรก็ตามมันยังไม่เป็นที่น่าพอใจ โดยเฉพาะข้อมูลขนาดเล็กที่อยู่ในระยะไกล การสร้าง Connection ไปยังเครื่องต้นฉบับ อาจใช้เวลามากกว่าการส่งข้อมูลเสียอีก ถึงแม้ว่าจะมีการตรวจสอบความล้าสมัยด้วยการร้องขอแบบมีเงื่อนไข (IMS GET) มันก็ยังใช้เวลาเกือบจะเท่ากับการที่ Client ร้องขอข้อมูลจากเครื่องต้นฉบับโดยตรง จากกรณีเช่นนี้จุดประสงค์ของการใช้ Cache จะสูญเสียไป ดังนั้นทางที่เป็นไปได้ผู้ใช้งานอาจต้องมีการยอมรับข้อมูลที่มีความล้าสมัยอยู่บ้าง เพื่อความรวดเร็วในการตอบสนองในทันทีทันใดจาก Cache แต่ถ้าผู้ใช้งานรู้สึกว่าข้อมูลมีความล้าสมัยก็ยังสามารถที่จะบังคับให้ Cache ทำการดึงข้อมูลจากเครื่องต้นฉบับมาใหม่ได้เช่นกัน

ถ้า Cache ไม่ได้มีการตรวจสอบความล้าสมัยในทุกๆ การร้องขอข้อมูล จะต้องมีวิธีการที่จะตัดสินว่าเมื่อไร ที่จะต้องมีการตรวจสอบ วิธีที่ง่ายที่สุดคือ ตรวจสอบเมื่อช่วงระยะเวลาหนึ่งผ่านไป (ในทางปฏิบัติจะตรวจในครั้งแรกที่มีการร้องขอข้อมูล หลังจากได้เลยช่วงเวลาที่กำหนด) ดังนั้น

จึงเป็นการยอมให้ผู้บริหาร Web Cache สามารถกำหนดความถี่มากที่สุดเป็นเวลาที่น่าเชื่อถือ ตัวอย่างเช่น กำหนดให้ Cache ต้องไม่นำข้อมูลที่มีอายุมากเกินไปกว่า 12 ชั่วโมงมาใช้

อีกวิธีการหนึ่งเป็นวิธีที่พยายามจะสะท้อนถึงความหลากหลายของข้อมูลบน Web ที่บ้างก็เปลี่ยนเป็นรายชั่วโมง บ้างก็ไม่เคยมีการเปลี่ยน เมื่อ Cache เก็บข้อมูลมันจะกำหนดอายุเวลาของข้อมูลเอาไว้ (time to live หรือ TTL) ข้อมูลนี้จะถูกนำมาใช้โดยมีต้องมีการตรวจสอบกับต้นฉบับจนกระทั่งผ่านพ้นเวลานี้ ใน HTTP Protocol สามารถกำหนด Timestamp Headers ซึ่งนำมาใช้ในการกำหนดค่า TTL ได้ แต่โดยส่วนใหญ่ไม่ได้ถูกนำมาใช้หรือไม่อยู่ใน Web Server ทั้งหมด ดังนั้นสิ่งที่นำมาใช้กันอย่างกว้างขวางก็คือเวลาที่มีการเปลี่ยนแปลงครั้งล่าสุด (Last-Modify) โดยเราจะกำหนด TTL ให้เป็นเปอร์เซ็นต์ของอายุข้อมูลในปัจจุบัน ตัวอย่างเช่น กำหนดให้ TTL มีค่าเป็น 10% ของข้อมูลซึ่งมีการเปลี่ยนแปลงครั้งล่าสุดท้ายเมื่อ 10 วันก่อน ดังนั้นสำเนาข้อมูลจะไม่ถูกตรวจสอบจนกว่าจะผ่าน 1 วันให้หลัง มีค่าอีกสองค่าที่โดยปรกติจะนำมาใช้ในการคำนวณหา TTL ด้วยก็คือค่ามากที่สุดที่จะปกป้องสำเนาข้อมูลให้สามารถนำมาใช้ได้โดยไม่ต้องมีการตรวจสอบ และค่า TTL โดยปริยายซึ่งจะใช้สำหรับข้อมูลที่ไม่ทราบแน่ชัดถึงวันที่มีการเปลี่ยนแปลงครั้งล่าสุด

วิธีการใช้ TTL จะเป็นการเพิ่มความสามารถในการจัดการความถี่ได้ทั้งสองทางด้วยกันคือ ข้อมูลที่ไม่มีการเปลี่ยนแปลงจะไม่ถูกตรวจสอบโดยไม่จำเป็น และข้อมูลที่มีการเปลี่ยนแปลงเสมอจะไม่ถี่จนเกินไปนัก อย่างไรก็ตามผู้ใช้งานมักจะต้องการให้มีการประกันเวลาสูงที่สุดที่ข้อมูลมีความถี่ด้วย

โปรแกรม Web Cache โดยส่วนใหญ่ ยอมให้มีการกำหนดค่าพารามิเตอร์ของความถี่ที่ต่างกันได้ตามรูปแบบ URL ซึ่งโดยทั่วไปจะมีข้อแตกต่างระหว่างเอกสารที่เป็นข้อความและรูปภาพ เอกสารที่เป็นข้อความมักจะมีขนาดเล็กและมักมีการเปลี่ยนแปลงบ่อย ขณะที่เอกสารที่เป็นรูปภาพมีขนาดใหญ่และเปลี่ยนแปลงน้อยกว่า ความสามารถเช่นนี้สามารถนำไปใช้กับการรองรับ Protocol อื่นๆ ได้อีกด้วย เช่น FTP หรือ Gopher ซึ่งจะไม่ปรากฏอายุของข้อมูล

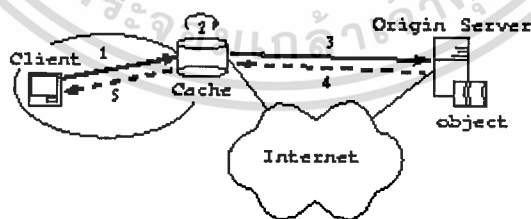
### 2.3 ประวัติความเป็นมาของโปรแกรม Squid Proxy/Cache

ในช่วงแรกของระบบอินเทอร์เน็ต มักจะใช้ UNIX เป็นโครงสร้างพื้นฐาน การพัฒนาโปรแกรมลักษณะที่ช่วยลดการติดต่อไปยังภายนอกโดยเก็บข้อมูลไว้ใน Proxy/Cache ที่อยู่ใกล้ๆ จึงพัฒนาอยู่บนระบบปฏิบัติการ UNIX เป็นส่วนใหญ่ และเนื่องจากเป็นระบบเปิดจึงมีการพัฒนาโปรแกรม Proxy/Cache ออกมามากมาย เช่น CERN, Apache, Netscape Proxy Server เป็นต้น รวมไปถึง Squid ซึ่งแต่ละโปรแกรมก็มีความสามารถและจุดเด่นแตกต่างกันออกไป

โปรแกรม Squid ได้มีกำเนิดจาก Internet Research Task Force Research Group on Resource Discovery (IETF-RD) ซึ่งได้พัฒนาโปรแกรม Harvest เพื่อให้มีการใช้ Network ของแต่ละ Server อย่างมีประสิทธิภาพ โปรแกรมสามารถใช้กับระบบปฏิบัติการ SunOS และ Digital Unix สามารถรองรับโปรโตคอล HTTP, FTP และ GOPHER ได้ มีการเปิดเผย Source Code แจกจ่ายเผยแพร่ เพื่อให้มีการทดสอบและปรับปรุงโปรแกรมอย่างต่อเนื่อง จึงมีผู้นำไปใช้กับระบบปฏิบัติการอื่นๆ ด้วยอย่างแพร่หลาย เช่น AIX, FreeBSD, HP-UX, Linux and IRIX ต่อมาโปรแกรม Harvest ถูกนำไปใช้ในโครงการ Proxy/Cache ของ National Laboratory for Applied Network Research (NLNR) จึงได้เปลี่ยนชื่อมาเป็น Squid มีการพัฒนาเพิ่มขีดความสามารถ เช่น ทำให้สามารถใช้กับโปรโตคอล HTTPS และ SSL ได้ ซึ่งในปัจจุบันเป็น Version 2.3

#### 2.4 หลักการทำงานของโปรแกรม Squid Proxy/Cache

โปรแกรม Squid Proxy/Cache ทำหน้าที่ในการเป็นตัวกลางในการติดต่อ ซึ่งทุกๆ การร้องขอของ Client ควรจะกระทำผ่านโปรแกรมนี้ เมื่อได้รับการร้องขอข้อมูล โปรแกรมจะทำการตรวจสอบว่ามีข้อมูลนั้นอยู่ภายใน Cache (อาจเป็น Memory Cache หรือ Disk Cache) หรือไม่ก่อน หากมีก็จะทำการตรวจสอบความทันสมัย(Consistency) ของข้อมูล จาก TTL (Time To Live) ถ้าข้อมูลนั้นเป็นข้อมูลที่หมดอายุก็จะทำการร้องขอข้อมูลต่อไปยัง Server ด้วยการร้องขอแบบ IMS GET (If Modify Since GET) ซึ่งจะใช้ Packet ในการรับ-ส่ง น้อยกว่าแบบ HEAD เพราะสามารถตรวจสอบและรับข้อมูลโดยการร้องขอได้ในครั้งเดียว แต่ถ้าไม่พบก็จะร้องขอแบบ GET เช่นเดียวกับ Browser ธรรมดา การทำงานแสดงดังภาพที่ 2.1



ภาพที่ 2.1 การทำงานของ Proxy/Cache

ข้อมูลที่ได้รับมาอาจจะถูกเก็บไว้เป็น Object Cache พร้อมกับอายุของข้อมูล(TTL) เพื่อใช้ในการตรวจสอบการร้องขอในครั้งต่อไป ลักษณะโครงสร้างการเก็บข้อมูลของ Object ใน Disk จะอยู่ในรูป File โดยที่ชื่อ File มีการเข้ารหัส Hash Function เก็บอยู่ใน Path 2 ถ้าดัชนีซึ่งสามารถปรับเปลี่ยนขนาดได้ (ดูรายละเอียดวิธีการปรับแต่งในข้อ 5.) เมื่อเนื้อที่เก็บข้อมูลเต็มก็จะทำการลบเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

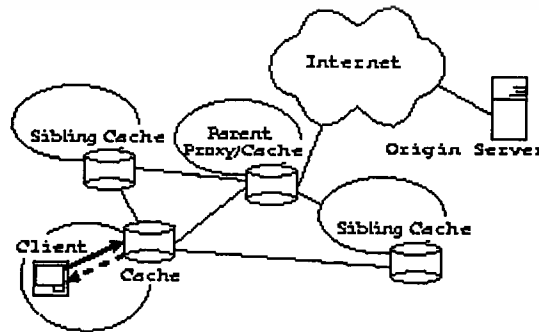
ข้อมูลเก่าๆ ที่มีการใช้น้อย (โดยดูจาก LRU Expiration Age) เพื่อให้มีเนื้อที่เหลือพอสำหรับเก็บข้อมูลใหม่ต่อไป

### 2.3.1 การให้บริการกับ Client

Client สามารถส่งการร้องขอข้อมูลผ่าน Squid Proxy/Cache ด้วยโปรโตคอล HTTP คล้ายกับการติดต่อไปยัง Server โดยตรง ต่างกันก็เพียงแต่ต้องส่งค่า URL เต็มๆ ไม่ใช่แค่ส่วน Item ที่ต้องการ เพราะโปรแกรม Squid จะนำมันไปใช้ในการร้องขอต่ออีกทีหนึ่ง นอกจากนั้นแล้ว Squid ยังรองรับการร้องขอด้วยโปรโตคอล ICP มีการส่ง Packet แบบ UDP ซึ่งขนาดเล็กกว่า และสามารถใช้มันในการตรวจสอบการให้บริการและใช้เลือกเครื่องที่จะติดต่อทำการรับ-ส่งข้อมูล ซึ่งจะเป็นประโยชน์ต่อการทำ Load Balancing ที่จะกล่าวถึงต่อไปอีกด้วย โปรแกรม Squid สามารถเลือกที่จะให้บริการให้กับ Client ผ่านทางโปรโตคอล HTTP หรือ ICP ได้ โดยกำหนดจาก Access Control List (ดูรายละเอียดวิธีการปรับแต่งในข้อ 5.)

### 2.3.2 การทำงานร่วมกันใน Network

Squid สามารถทำงานร่วมกันในเครือข่ายเพื่อแบ่งปันภาระ และลดการใช้พื้นที่ Disk ในแต่ละ Proxy/Cache ทำให้การทำงานมีประสิทธิภาพยิ่งขึ้น สามารถให้บริการกับ Client จำนวนมากๆ ได้ โดยส่วนใหญ่แล้วโครงสร้างการทำงานจะเป็นแบบ Hierarchy แบ่งเป็นหลายๆ ระดับชั้น เช่น Local, Regional หรือ National Proxy/Cache โดยที่ Parent Proxy/Cache มักจะเป็นเครื่องที่อยู่ใกล้กับข้อมูลมากกว่าเมื่อเทียบกับอีกเครื่องหนึ่ง ซึ่งหากไม่พบข้อมูลในที่ใดเลย ก็จะทำการติดต่อไปยัง Parent อีกทอดหนึ่ง และเนื่องจาก Parent Proxy/Cache มักมีภาระในการทำงานสูงจากการให้บริการกับ Client จำนวนมากกว่า ดังนั้นส่วนใหญ่จึงทำการสำเนาข้อมูลที่ได้มาจาก Parent เก็บไว้ด้วย แต่อย่างไรก็ตามสามารถกำหนดลงไปว่า ไม่ให้มีการเก็บข้อมูลก็ได้ (ดูรายละเอียดวิธีการปรับแต่งในข้อ 2.5.)

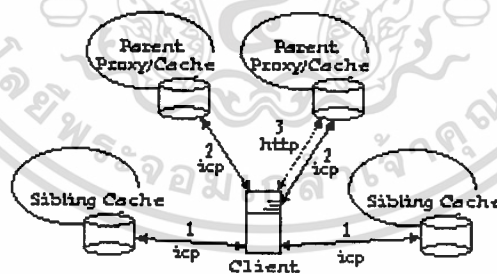


ภาพที่ 2.2 Parent และ Sibling Cache

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Sibling Cache จะเป็นเครื่องที่อยู่ใกล้เคียงกันและอยู่ในระดับเดียวกัน ซึ่งอาจมีข้อมูลที่ต้องการเก็บอยู่ก็ได้ ดังนั้นเพื่อที่จะลดภาระการทำงานของ Parent Proxy/Cache ก่อนที่จะติดต่อไปยัง Parent จะทำการค้นหาข้อมูลที่อยู่ใน Sibling Cache ก่อน ด้วย ICP โพรโตคอล หากพบก็จะทำการนำข้อมูลส่งต่อไปให้กับ Client ที่ทำการร้องขอในทันที โดยอาจจะเก็บหรือไม่เก็บข้อมูลไว้ในเครื่องตนเองก็ได้ ขึ้นอยู่กับว่าต้องการลด Traffic ในการสื่อสารระหว่าง Proxy/Cache หรือต้องการประหยัดเนื้อที่เก็บข้อมูลใน Disk ตัวอย่างดังภาพที่ 2.2

Squid Proxy/Cache สามารถมีทั้ง Parent หรือ Sibling ได้มากกว่า 1 เครื่อง ซึ่งในการทำงาน Squid จะใช้ ICP Packet ในการตรวจสอบและค้นหาข้อมูลจาก Sibling ก่อน แล้วก็ Parent ตามลำดับ โดยส่ง Packet ออกไปพร้อมๆ กัน แล้วตรวจหาเวลาในการตอบสนอง แล้วเลือกติดต่อกับ Proxy/Cache เครื่องที่มีข้อมูลที่ต้องการและมีการตอบรับเร็วที่สุด (แสดงในภาพที่ 2.3) ซึ่งเวลาในการตอบสนองจะขึ้นอยู่กับความเร็วและภาระงานของแต่ละเครื่อง รวมไปถึงผลจากระยะทางและความคับคั่งของเครือข่ายด้วย จึงเป็นแนวทางที่ดีในการทำ Load Balancing ในเครือข่าย แต่หากไม่พบข้อมูลที่ต้องการอีก Squid จะทำการติดต่อไปยัง Parent ที่ตอบรับเร็วที่สุดถึงแม้จะไม่มีข้อมูลที่ต้องการ แต่ก็ยังเป็นเส้นทางที่ใกล้กับข้อมูลมากกว่า อย่างไรก็ตามหากไม่มีการตอบสนองใดๆ จาก Parent Proxy/Cache ซึ่งเป็นแนวทางสุดท้ายแล้ว โปรแกรม Squid จะเลือกทำการร้องขอข้อมูลจากแหล่งของข้อมูลโดยตรง



ภาพที่ 2.3 การใช้ ICP Packet

## 2.5 การติดตั้งโปรแกรม Squid

ขั้นตอนการติดตั้งไม่ยุ่งยากมากนัก ก่อนอื่นจะต้องทำการ Download Source Archive File จากเว็บไซต์ <http://www.squid-cache.org/> ซึ่งจะเป็น File ที่มีการบีบอัดข้อมูลเอาไว้อยู่ในรูป `squid-x.y.z-src.tar.gz` (ตัวอย่างเช่น `squid-2.2.STABLE5-src.tar.gz`) แล้วนำมาขยายออกเป็น File Install ด้วยคำสั่ง `"tar -xzf squid-2.2.STABLE5-src.tar.gz"` ใน Version 2 นี้จะต้องทำการปรับแต่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมด้วย Configure Script ก่อน ซึ่งจะใช้คำสั่ง `./configure [option]` ซึ่งใน [Option] อาจระบุให้มีการ Enable SNMP หรือ Option อื่นๆ อีกก็ได้ เช่น `./configure --enable-snmp` (สามารถอ่านวิธีการเซต Option ต่างๆ ได้ด้วยคำสั่ง `./configure --help`) แล้วจึง Install ด้วยคำสั่ง `make install` หลังจากนั้นให้ทำการปรับแต่งการทำงานโดยการแก้ไข File `squid.conf` ตามความเหมาะสมของการใช้งานของเรา ซึ่งโดยปรกติจะเก็บไว้อยู่ที่ `/usr/local/squid/etc/` (ดูรายละเอียดวิธีการปรับแต่งในข้อ 5.) เมื่อเสร็จแล้วจะต้องทำการสร้าง Swap directories สำหรับเก็บข้อมูล โดยใช้คำสั่ง `/usr/local/squid/bin/squid -z` เท่านั้น โปรแกรม squid ก็พร้อมที่จะเริ่มทำงานแล้ว ด้วยการใช้คำสั่ง `/usr/local/squid/bin/squid` (อาจจะต้องมีการใช้ Option `-f` ตามด้วย Path Name ของ `squid.conf` กรณีที่ต้องการเปลี่ยนชื่อหรือตำแหน่งที่เก็บ Configuration File) แต่หากมีการแก้ไข File `squid.conf` ในภายหลังอีก จะต้องทำการสั่งให้โปรแกรม squid อ่าน Configuration File ใหม่เพื่อให้เกิดการเปลี่ยนแปลงที่แก้ไขไปเป็นผล ด้วยคำสั่ง `/usr/local/squid/bin/squid -k reconfigure` หรืออาจใช้วิธีการหยุดแล้วเริ่มทำงานใหม่ก็ได้เช่นกัน

## 2.6 การ Configuration Parameter

การเปลี่ยน Parameter ในการทำงานของโปรแกรม Squid สามารถทำได้โดยการแก้ไข File `squid.conf` ซึ่งจะถูกรับอ่านในตอนเริ่มต้นทำงานหรือด้วยการสั่งให้โปรแกรม squid อ่านค่าใหม่ดังที่ได้กล่าวไปแล้วข้างต้น โดยไม่ต้องทำการ Compile โปรแกรมใหม่ ใน File นี้ เป็น Text File ซึ่งประกอบไปด้วย Tag และ Parameter หลากๆ ตัวอยู่ด้วยกัน Parameter โดยส่วนใหญ่จะมีค่า Default Value แต่หากต้องการเปลี่ยนแปลงค่า ก็เพียงแค่ Uncomment ด้วยการลบเครื่องหมาย `#` ที่หน้าบรรทัดนั้นออก แล้วทำการแก้ไขเปลี่ยนแปลงค่า เช่น ต้องการเปลี่ยน HTTP Service Port จากเดิม 3128 เป็น Port 8080 ก็ให้ทำการแก้ไข บรรทัดที่เขียนว่า `#http_port 3128` แก้ไขเป็น `http_port 8080`

Parameter Option ใน `squid.conf` แบ่งออกเป็นหลายๆ ส่วนด้วยกัน แต่จะกล่าวถึงเฉพาะส่วนหลักๆ ที่สำคัญ ดังนี้

*“Network Options”* เป็นการระบุถึงการติดต่อโปรแกรม Squid โดยผ่านทางเครือข่าย เช่น หมายเลข Port ของการติดต่อด้วย TAG: `http_port` และ `icp_port` ตามลำดับ

*“Options Which Affect The Neighbor Selection Algorithm”* เป็นการกำหนดถึงการทำงานร่วมกันในเครือข่าย ซึ่งประกอบไปด้วยข้อมูลของ Cache เครื่องอื่นๆ ที่จะติดต่อกับ เช่น ชื่อเครื่อง ชนิดของ Cache ว่าเป็น Parent หรือ Sibling รวมทั้ง HTTP และ ICP Port ที่จะใช้ติดต่อ หรือ Option อื่นๆ เช่น ต้องการเพียงแค่ส่งผ่านข้อมูลโดยไม่ต้องเก็บไว้ใน Cache (proxy-only) โดยจะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TAG: *cache\_peer* เช่น *"cache\_peer sib.foo.net sibling 3128 3130 proxy-only"* รวมไปถึงการบังคับให้มีการส่ง UDP Query ไปยังเครื่องใดเครื่องหนึ่งเท่านั้นหากเป็น Domain ที่ระบุ ด้วย TAG: *cache\_peer\_domain*

*"Options Which Affect The Cache Size"* เป็นการกำหนดถึงการใ้ Memory ของโปรแกรม Squid ในการทำงาน ด้วย TAG: *cache\_mem* และขนาดของข้อมูลขนาดใหญ่ที่สุดที่โปรแกรม Squid จะทำการเก็บไว้ ด้วย TAG: *Maximum\_object\_size*

*"Logfile Pathnames and Cache Directories"* เป็นส่วนของการระบุ ชื่อ File และ Path ของ Log File ต่างๆ เช่น TAG: *cache\_access\_log* แล้วส่วนนี้ยังประกอบด้วย ตำแหน่ง Path ที่ใช้ในการเก็บข้อมูล Cache ขนาดของพื้นที่ Disk ในหน่วย kB ทั้งหมดภายใต้ Path นั้น จำนวนของ Sub Directory ในลำดับชั้นที่ 1 และ 2 ตามลำดับ ด้วย TAG: *cache\_dir* ซึ่งสามารถมีตำแหน่งที่ใช้เก็บข้อมูลหลายที่หรือหลาย Partition ได้โดยเขียนเรียงไว้ในบรรทัดถัดไป ตัวอย่างเช่น *"cache\_dir /usr/local/squid/cache 100 16 256"*

*"Access Controls"* เป็นส่วนควบคุมการเข้าใช้งาน ซึ่งจะมีการตรวจสอบสิทธิในการใช้ ในส่วนแรกจะเป็นการกำหนดชื่อและคุณลักษณะของ Access Control List สำหรับการอ้างถึงในภายหลัง โดยใช้ TAG: *acl* ในรูปแบบ *"acl aclname acltype string1 .."* สามารถเขียนได้หลายบรรทัดตามจำนวนชื่อ ตัวอย่างเช่น *"acl kmitl src 168.246.0.0/255.255.0.0"* ส่วนต่อมาเป็นการบ่งบอกถึงการอนุญาตหรือไม่ ในการใช้โปรโตคอล HTTP ติดต่อเข้ามา ด้วยการใช้ TAG: *http\_access* ในรูปแบบ *"http\_access allow|deny aclname ..."* ซึ่งการใช้งานจะต้องทำการอนุญาตให้กับผู้ใช้ที่จะติดต่อกับเครื่องนี้ เช่น *"http\_access allow kmitl"* ส่วนถัดมาเป็น TAG: *icmp\_access* ซึ่งจะใช้งานคล้ายๆ กัน เพียงแต่เป็นการอนุญาตให้ติดต่อด้วยโปรโตคอล ICP การอนุญาตของทั้ง 2 ส่วนนี้ ลำดับของการเรียงบรรทัดมีความสำคัญจากบนลงล่างตามลำดับ

*"Miscellaneous"* เป็นส่วน Parameter อื่นๆ เช่น การติดต่อโปรแกรม Squid ด้วยโปรโตคอล SNMP ว่าจะติดต่อผ่านทาง Port ใด ด้วยการใช้ TAG: *snmp\_port* และอนุญาตให้ใครบ้างด้วย TAG: *snmp\_access* ตามหลักของ Access Control List

## 2.7 การตรวจสอบประสิทธิภาพการทำงานของโปรแกรม Squid

เราสามารถตรวจสอบประสิทธิภาพการทำงานของ Squid Proxy/Cache ได้หลายวิธี แต่วิธีที่เป็นที่นิยมมีด้วยกัน 3 วิธี คือ การตรวจวิเคราะห์จาก Log File, ตรวจสอบจากโปรแกรม Cache Manager ของ Squid และการตรวจสอบจาก MIB Object ของ Squid ด้วย SNMP โปรโตคอล

ในขณะที่ทำงาน โปรแกรม Squid จะทำการบันทึกเหตุการณ์และสถานะการทำงานไว้ใน Log File ตลอดเวลา เช่น ข้อมูลการทำงานทั่วไป(cache.log), ข้อมูลการใช้งาน(access.log) หรือ ข้อมูล Object ที่เก็บไว้(store.log) ซึ่งเราสามารถนำ File เหล่านี้มาวิเคราะห์การทำงานของโปรแกรม ตลอดเวลาที่ผ่านไป ได้ ซึ่งโดยปกติจะทำการวิเคราะห์วันละ 1 ครั้งในช่วงที่มีการใช้งานน้อย โดยมี โปรแกรมที่นิยมคือ Calamaris ซึ่งเขียนด้วย Perl

โปรแกรม Cache Manager เป็น CGI โปรแกรมที่ผนวกมากับโปรแกรม squid การใช้โปรแกรมนี้จะต้องนำ File cachemgr.cgi มาไว้ใน PATH ที่สามารถใช้งาน CGI ได้ (ขึ้นอยู่กับ Configuration ของโปรแกรม Web Server ที่ใช้) โปรแกรมนี้สามารถตรวจสอบประสิทธิภาพการทำงานของ Squid Proxy/Cache ในขณะนั้นได้ว่ามีสถานะเป็นอย่างไร เช่นสถิติการใช้ Memory หรือ CPU รวมไปถึงสถานะการทำงานของโปรแกรมที่ผ่านมาโดยไม่ต้องพึ่ง Log File

การใช้ SNMP โปรโตคอล ตรวจสอบ MIB Object ของ Squid จะคล้ายๆ กับการตรวจสอบด้วยโปรแกรม Cache Manager ซึ่งสามารถส่ง Packet SNMP ในการสอบถามข้อมูลการทำงานที่เราสนใจ จากโปรแกรมใดๆ ก็ได้ที่สามารถทำงานร่วมกับ SNMP ตัวอย่างเช่น โปรแกรม MRTG ซึ่งเขียนด้วย Perl โดยสามารถนำค่าที่ตรวจสอบได้มาแสดงผลในรูปภาพ

การตรวจสอบด้วย SNMP นี้จำเป็นต้องปรับแต่งโปรแกรม Squid ด้วย Configure Script ให้มีการ Enable SNMP (จากที่ได้กล่าวไปแล้วในหัวข้อ 2.6) การตรวจสอบจะติดต่อผ่าน Default Port 3401 ด้วย Community String “squid” หรือ “public” ด้วยหมายเลข OID ‘1.6.1.4.1.3495.1’ หรือในชื่อ “iso.org.dod.internet.private.enterprises.nlamr” ในปัจจุบันโปรแกรม Squid แบ่ง MIB Object ออกเป็น 5 กลุ่มย่อยด้วยกัน คือ cacheSystem, cacheConfig, cachePerf, cacheNetwork และ cacheMesh

“cacheSystem” จะเป็นข้อมูลเกี่ยวกับระบบของโปรแกรม Squid คือ ขนาดของ Virtual Memory, เนื้อที่ Disk ที่ใช้ และเวลาทำงานตั้งแต่เริ่มโปรแกรม ส่วน “cacheConfig” จะเป็นส่วนที่เกี่ยวกับการ Configuration โปรแกรม ตัวอย่างเช่น Software Version, ขนาดของ Memory สูงสุดที่ได้ตั้งไว้ เป็นต้น

ส่วนต่อมา “cachePerf” จะเป็นส่วนที่แสดงถึงประสิทธิภาพการทำงาน(Performance) ของโปรแกรม Squid โดยจะแบ่งออกเป็น 2 ส่วนย่อย คือ ส่วนแรก cacheSysPerf เป็นการบอกถึงประสิทธิภาพของระบบ เช่น การใช้งาน Memory การใช้งาน CPU รวมไปถึงจำนวนข้อมูลที่เกิดข้อผิดพลาด และส่วนที่ 2 คือ cacheProtoAggergateStats เป็นการรายงานสถิติการใช้งานด้วยโปรโตคอล HTTP และ ICP ซึ่งส่วนใหญ่จะมี Syntax เป็นแบบ Counter 32 bits

ส่วนถัดมา “cacheNetwork” ถูกออกแบบมาสำหรับการใช้ในกลุ่มของ Proxy/Cache ประกอบด้วย 3 ส่วนคือ cacheIpCache, cacheFqdnCache และ cacheDNS และส่วนสุดท้าย “cacheMesh” ใช้สำหรับการดูข้อมูลของเครื่องที่เข้ามาติดต่อด้วย มีสองส่วนเช่นกัน คือ ข้อมูลและสถานะของแต่ละ Cache Peer (Parent และ Sibling) และข้อมูลของแต่ละ Client ที่ติดต่อเข้ามา โดยจะอ้างถึงด้วย IP Address ต่อท้าย OID

## 2.8 สรุปบท

Proxy/Cache สามารถช่วยแก้ไขปัญหาคอขวดใน Internet Network ได้เป็นอย่างดี โดยการเป็นตัวกลางในการติดต่อ และสำเนาข้อมูลที่มีการใช้บ่อยเก็บไว้ใน Cache เพื่อลดปริมาณการติดต่อไปยังเครือข่ายภายนอก แต่มีข้อเสียคือข้อมูลอาจมีความล้าสมัยเนื่องจากข้อมูลต้นฉบับมีการเปลี่ยนแปลงหลังจากที่ทำสำเนาเก็บไว้ จึงต้องมีการตรวจสอบความถูกต้องของข้อมูลด้วย

Squid เป็นโปรแกรม Proxy/Cache ตัวหนึ่งที่มีการใช้งานกันอย่างแพร่หลาย สามารถส่งการร้องขอด้วยโปรโตคอล HTTP คล้ายกับการติดต่อไปยัง Server โดยตรง อีกทั้งยังรองรับการร้องขอด้วยโปรโตคอล ICP ทำให้สามารถทำงานร่วมกันระหว่าง Proxy/Cache ด้วยกันใน Network หลายๆ เครื่องเพื่อเพิ่มประสิทธิภาพในการทำงานได้ดียิ่งขึ้น

การติดตั้งโปรแกรมเพื่อใช้งานไม่ยุ่งยากนักเนื่องจากจะมี Script สำเร็จรูปสำหรับจัดการทั้งหมด เพียงแต่ต้องทำการ Configure Parameter ต่างๆ เพื่อให้โปรแกรมทำงานได้ตามวัตถุประสงค์ ซึ่ง Parameter ก็มีอยู่หลายตัวด้วยกัน เราสามารถตรวจสอบประสิทธิภาพการทำงานของโปรแกรม Squid Proxy/Cache ได้หลายวิธีด้วยกัน วิธีหนึ่งก็คือการตรวจสอบโดยใช้โปรโตคอล SNMP ซึ่งเป็นมาตรฐานและใช้งานง่าย

## บทที่ 3

### การวิเคราะห์และออกแบบ

#### 3.1 ปัญหาของระบบในปัจจุบัน

โปรแกรม Squid สามารถนำมาใช้ได้กับหน่วยงานที่มีการใช้ Internet และต้องการให้มีการใช้งานเครือข่ายอย่างมีประสิทธิภาพ ในหน่วยงานขนาดเล็กอาจมีปัญหาคาดผู้ที่มีความรู้ที่จะมาดำเนินการในส่วนนี้ หรือกรณีที่หน่วยงานมีขนาดใหญ่ระบบเครือข่ายมีความซับซ้อนและห่างไกลกัน การบริหารจัดการกันเองในแต่ละแผนก อาจทำให้ระบบมีประสิทธิภาพทำงานไม่ดีเท่าที่ควร

โปรแกรม Squid Proxy/Cache นั้นการเปลี่ยนแปลง Parameter ในการทำงานทำได้ยาก เพราะต้องทำการแก้ไข File "*squid.conf*" ที่อยู่ใน Proxy/Cache แต่ละเครื่อง ซึ่ง File นี้ประกอบไปด้วย Tag และ Parameter อยู่ด้วยกันหลายตัวรวมกันเกือบ 2000 บรรทัด ทำให้การค้นหาสิ่งที่ต้องการแก้ไขทำได้ช้า อีกทั้งมีความซับซ้อนเพราะบาง Tag อาจมีได้หลายบรรทัด ตัวอย่างเช่น "*acl*" หรือ "*cache\_dir*" ซึ่งการเปลี่ยนแปลงนี้บางครั้งจะต้องมีความสัมพันธ์กับเครื่องอื่นๆ ที่อยู่ในระบบด้วย ดังนั้นจึงมีความยุ่งยากพอสมควรถ้าหากมีการเปลี่ยนแปลงเกิดขึ้น

ส่วนการตรวจสอบประสิทธิภาพการทำงาน ถ้าเป็นการวิเคราะห์จาก Log File จะต้องมีการใช้โปรแกรมประกอบสำหรับการวิเคราะห์และต้องใช้กำลังของเครื่อง Proxy/Cache ในการทำงานเป็นอย่างมาก ถ้าเป็นการตรวจสอบด้วย Cache Manager ที่ผนวกมากับโปรแกรม Squid สามารถใช้งานได้ดี แต่การติดต่อผู้ใช้ส่วนใหญ่จะเป็นในลักษณะข้อความซึ่งอ่านเข้าใจได้ยาก ส่วนการตรวจสอบด้วย SNMP จำเป็นต้องปรับแต่งโปรแกรม Squid ตั้งแต่ตอนติดตั้งโปรแกรม แต่สามารถใช้งานร่วมกับโปรแกรมอื่นได้ง่ายเพราะเป็นมาตรฐาน

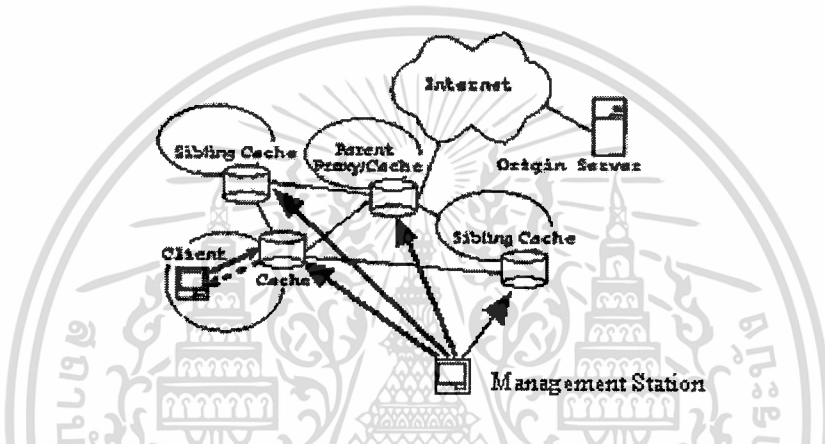
#### 3.2 ความต้องการของระบบ

ต้องการสิ่งที่ช่วยอำนวยความสะดวก ในการบริหารจัดการระบบ Squid Proxy/Cache ทั้งหมดจากส่วนกลางที่จุดเดียว โดยที่ผู้บริหารไม่จำเป็นต้องมีความรู้ความชำนาญมากนัก เช่น การเรียกดูและปรับปรุงแก้ไขค่า Configuration ต่างๆ ของ Proxy/Cache การตรวจสอบประสิทธิภาพการทำงานในแต่ละเครื่องหรือเมื่อนำมาเปรียบเทียบกัน การสั่งให้โปรแกรมหยุดหรือเริ่มการทำงาน และการแสดงภาพความสัมพันธ์ของ Proxy/Cache ทั้งระบบเพื่อให้เกิดความเข้าใจและนำไปปรับ

ปรุ่งแก้ไข ทั้งนี้เพื่อให้การบริหารจัดการระบบ Squid Proxy/Cache ทำได้ง่ายและเป็นไปอย่างมีประสิทธิภาพช่วยเพิ่มคุณภาพของการให้บริการให้ดียิ่งขึ้น

### 3.3 การออกแบบระบบการทำงาน

จากปัญหาและความต้องการของระบบ จำเป็นต้องมีศูนย์กลางในการบริหารจัดการขึ้นมาเครื่องหนึ่ง(Management Station) สำหรับทำหน้าที่ปรับปรุ่งแก้ไขค่า Configuration ต่างๆ และคอยตรวจสอบประสิทธิภาพการทำงาน ของ Proxy/Cache ทั้งระบบ ดังภาพที่ 3.1



ภาพที่ 3.1 ภาพการเชื่อมต่อเครือข่ายของระบบ

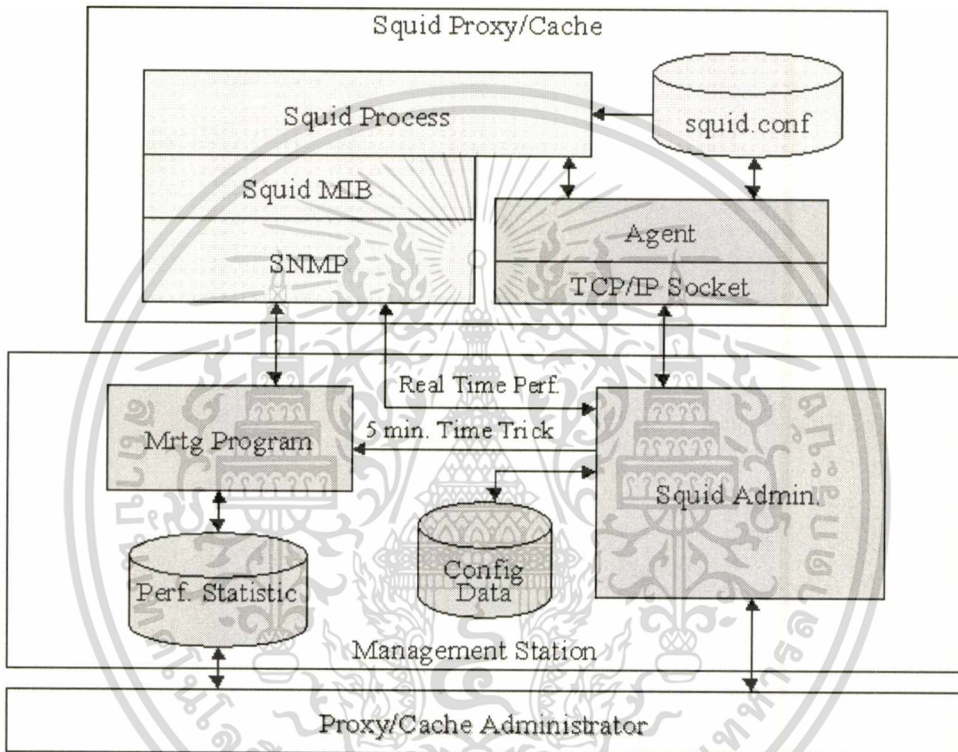
การบริหารจัดการทำจากศูนย์กลางที่ Management Station โดยส่งคำสั่งควบคุมไปยัง Proxy/Cache ที่กระจายอยู่ตามที่ต่างๆ แต่เนื่องจาก โปรแกรม Squid Proxy/Cache ยังไม่เปิดให้มีการควบคุมจากเครื่องที่อยู่ไกลออกไป ดังนั้นจึงต้องมีโปรแกรมส่วนหนึ่งอยู่ที่ฝั่ง Proxy/Cache ที่คอยรับคำสั่งจาก Management Station แล้วส่งการทำงานต่อไปยัง โปรแกรม Squid อีกทีหนึ่ง

เพื่อให้การบริการของ Proxy/Cache เป็นไปอย่างต่อเนื่อง ในขณะที่ผู้บริหารจัดการกระทำการเปลี่ยนแปลง Configuration ที่ Management Station ในตอนแรก จะยังไม่มีผลต่อ Proxy/Cache ทันที เป็นผลให้ Proxy/Cache ยังสามารถให้บริการกับผู้ใช้เหมือนเดิม ต่อเมื่อผู้บริหารจัดการพอใจในการเปลี่ยนแปลงแล้วจึงจะสั่งให้การเปลี่ยนแปลงนั้นเป็นผลอีกครั้งหนึ่ง

การตรวจสอบประสิทธิภาพการทำงานจากศูนย์กลาง ทำให้ทราบถึงภาระการทำงานของทั้งระบบได้ และสามารถทำการปรับเปลี่ยนได้ทันที ทำให้การบริการเป็นไปอย่างต่อเนื่อง

### 3.4 ส่วนประกอบของโปรแกรม

จากการออกแบบการทำงาน โปรแกรมบริหารจัดการระบบ Proxy/Cache จะต้องประกอบด้วยโปรแกรม 2 ส่วน โดยส่วนหนึ่งจะทำงานอยู่บนเครื่อง Proxy/Cache ซึ่งอาจมีได้หลายเครื่อง คอยส่งต่อคำสั่งการทำงานไปยังโปรแกรม Squid และอีกส่วนหนึ่งเป็นโปรแกรมสำหรับการบริหารจัดการระบบจากส่วนกลาง แสดงดังภาพที่ 3.2



ภาพที่ 3.2 ส่วนประกอบของโปรแกรมช่วยบริหารจัดการระบบ Proxy/Cache

โดยในส่วนแรกจะมี Agent เป็นส่วนที่ติดต่อกับโปรแกรม Squid เพื่ออ่านค่า Configuration จาก File "squid.conf" ส่งให้กับ Management Station หรือในทางกลับกัน ใช้เขียนค่า Configuration ลงไปตามที่ได้รับคำสั่งมา อีกทั้งยังสามารถสั่งการให้โปรแกรม Squid ทำการเปลี่ยนแปลงการทำงานโดยอ่านค่าจาก Configuration ที่เขียนไว้ หรือสั่งให้โปรแกรม Squid ทำงานตามต้องการ เช่น ทำการสร้าง Swap Directory หยุดหรือเริ่มการทำงาน ส่วนนี้จะมีข้อตกลงในการติดต่อกับ Squid Administrator ในแบบ Client-Server

ส่วนที่สองสามารถแบ่งได้เป็นอีก 2 ส่วนย่อย ส่วนย่อยที่หนึ่งได้นำโปรแกรม Mrtg ที่มีจุดเด่นในด้านการแสดงผลของประสิทธิภาพในรูปแบบกราฟ มาใช้ประกอบในการเก็บสถิติการทำงานของโปรแกรม Squid Proxy/Cache ด้วยโปรโตคอล SNMP ทุกๆ 5 นาที โปรแกรมนี้จะนำค่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถิติของประสิทธิภาพ ที่เก็บได้มาสร้างเป็นกราฟประกอบด้วย Html File ซึ่งสามารถเรียกดูได้ด้วย Internet Browser ทั่วไปในภายหลัง ค่าประสิทธิภาพที่สนใจในที่นี้มีทั้งหมด 18 ค่า ดังตารางที่ 3.1

ตารางที่ 3.1 ค่าประสิทธิภาพการทำงาน โดยโปรโตคอล SNMP

ชื่อ Object	OID	ชนิดข้อมูล	คำอธิบาย
CacheServerRequests	1.3.6.1.4.1.3495.1.3.2.1.10	Integer 32 bit	จำนวน Request
CacheServerErrors	1.3.6.1.4.1.3495.1.3.2.1.11	Integer 32 bit	จำนวน Errors
CacheServerInKb	1.3.6.1.4.1.3495.1.3.2.1.12	Counter32 bit	จำนวนข้อมูลเข้าจาก Server ในหน่วย Kb
CacheServerOutKb	1.3.6.1.4.1.3495.1.3.2.1.13	Counter32 bit	จำนวนข้อมูลออกไป Server ในหน่วย Kb
CacheClientHttpRequests	1.3.6.1.4.1.3495.1.3.2.1.1	Counter32 bit	จำนวน Http Requests โดย Client
CacheHttpHits	1.3.6.1.4.1.3495.1.3.2.1.2	Counter32 bit	จำนวน Http Hits
CacheHttpErrors	1.3.6.1.4.1.3495.1.3.2.1.3	Counter32 bit	จำนวน Http Errors
CacheICPpktsSent	1.3.6.1.4.1.3495.1.3.2.1.8	Counter32 bit	จำนวน Packet ICP ที่ส่งออก
CacheICPpktsRecv	1.3.6.1.4.1.3495.1.3.2.1.9	Counter32 bit	จำนวน Packet ICP ที่ได้รับ
CacheICPkbSent	1.3.6.1.4.1.3495.1.3.2.1.6	Counter32 bit	จำนวนข้อมูล ICP ที่ส่งออกในหน่วย Kb
CacheICPkbRecv	1.3.6.1.4.1.3495.1.3.2.1.7	Counter32 bit	จำนวนข้อมูล ICP ที่ได้รับในหน่วย Kb
CacheHttpOutKb	1.3.6.1.4.1.3495.1.3.2.1.5	Counter32 bit	จำนวนข้อมูล Http ที่ส่งออกในหน่วย Kb
CacheHttpInKb	1.3.6.1.4.1.3495.1.3.2.1.4	Counter32 bit	จำนวนข้อมูล Http ที่ได้รับในหน่วย Kb
CacheCurrentSwapSize	1.3.6.1.4.1.3495.1.3.2.1.14	Counter32 bit	ขนาดของ Swap Directory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ Object	OID	ชนิดข้อมูล	คำอธิบาย
CacheNumObjCount	1.3.6.1.4.1.3495.1.3.1.7	Counter32 bit	จำนวนของ Object
CacheCpuUsage	1.3.6.1.4.1.3495.1.3.1.5	Integer 32 bit	เวลาที่ cpu ใช้เป็นวินาที
CacheMemUsage	1.3.6.1.4.1.3495.1.3.1.4	Integer 32 bit	จำนวน system memory ที่ถูกจองโดย cache
CacheSysPageFaults	1.3.6.1.4.1.3495.1.3.1.1	Counter32 bit	จำนวน system page faults

ในการทำงานจะต้องมีการสร้าง Configuration ของโปรแกรม Mrtg ที่เหมาะสมในแต่ละ Proxy/Cache แล้วต้องอาศัยโปรแกรม Perl ในการทำงานในทุกๆ 5 นาที ซึ่งทั้งหมดนี้จะจัดการโดยส่วนย่อยที่สองคือ Squid Admin ทั้งหมด นอกจากนี้ยังสามารถแยกการทำงานของส่วนย่อยนี้ไว้ที่เครื่องอื่นก็ได้ในกรณีที่ต้องการลดภาระการทำงานของ Management Station

ส่วนย่อยที่สองคือ Squid Admin เป็นส่วนที่มีความสำคัญมากที่สุด ออกแบบให้ทำงานที่ Management Station อยู่บนระบบปฏิบัติการ Windows เนื่องจากต้องการเน้นในเรื่องการติดต่อผู้ใช้ในแบบ GUI เพื่อให้ผู้ใช้สามารถเข้าใจและทำการเปลี่ยนแปลงแก้ไขค่า Configuration ได้โดยง่าย ส่วนนี้มีหน้าที่ในการทำงานหลายอย่างด้วยกัน ดังนี้

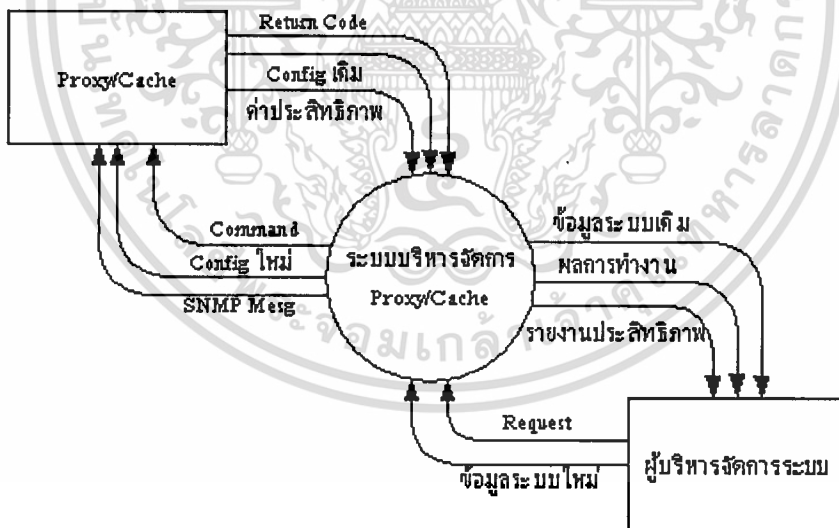
1. ติดต่อควบคุม Agent ที่อยู่บน Proxy/Cache แต่ละเครื่อง โดยการส่งคำสั่งไปในเครือข่าย TCP/IP ในลักษณะ Client-Server ซึ่งจะต้องมีข้อกำหนดไว้ล่วงหน้า(Request-Reply) การติดต่อสามารถมีได้หลายอย่าง เช่น การอ่านค่า Configuration เดิม การเขียนค่า Configuration ค่าใหม่ หรือการสั่งให้โปรแกรม Squid ทำงานตามที่ต้องการ
2. การนำข้อมูล Configuration จาก Agent ที่ได้มาจัดเก็บไว้ เพื่อเตรียมสำหรับการแก้ไขเปลี่ยนแปลงต่อไป
3. สามารถทำการ Backup หรือเรียกดูข้อมูลการ Configuration ทั้งระบบที่บันทึกเก็บไว้ได้
4. ทำการแก้ไขเปลี่ยนแปลงค่า Configuration ของ Proxy/Cache ตามที่ต้องการ หรืออาจสร้างขึ้นใหม่โดยไม่สนใจค่าเดิมเลยก็ได้
5. สามารถสร้าง Configuration ของโปรแกรม Mrtg สำหรับ Proxy/Cache เครื่องใดๆ แล้วสั่งให้โปรแกรม Mrtg ทำงานทุก 5 นาที เพื่อเก็บสถิติของประสิทธิภาพได้โดยอัตโนมัติ
6. สามารถแสดงประสิทธิภาพการทำงานของ Proxy/Cache ในขณะนั้นได้ โดยการติดต่อด้วยโปรโตคอล SNMP ไปยังโปรแกรม Squid โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

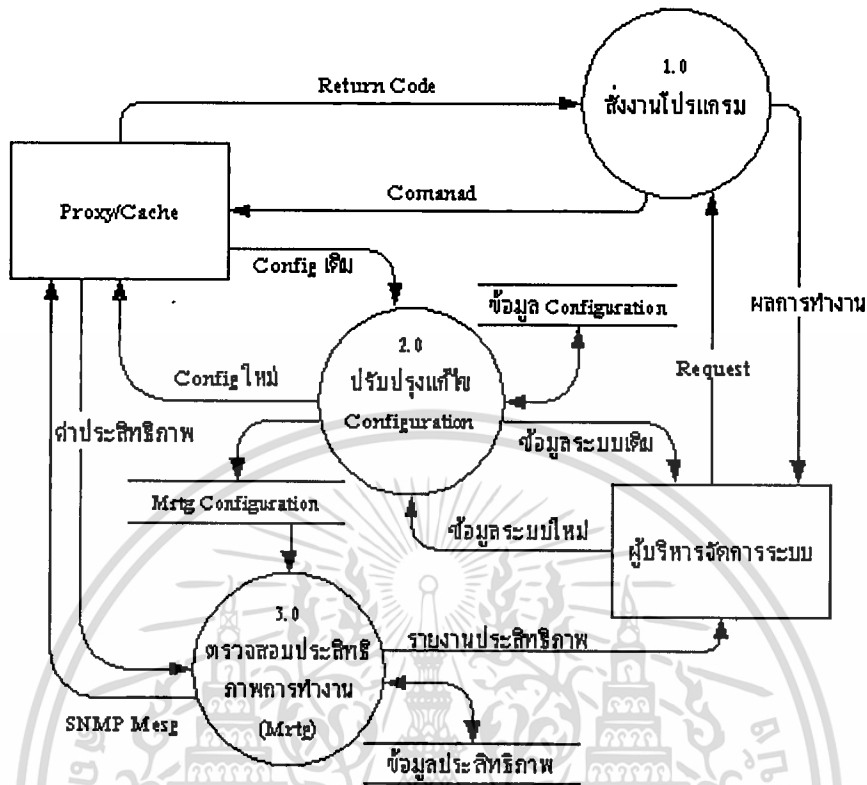
7. สร้างภาพแสดงความสัมพันธ์ของ Proxy/Cache ทั้งระบบ ทำให้ผู้ใช้สามารถเข้าใจและสามารถทำการเปลี่ยนแปลง Configuration ได้โดยง่าย เช่น วิธีการลากแล้วปล่อย(Drag & Drop) ภาพแสดงความสัมพันธ์นี้ ผู้ใช้สามารถทำการลากเปลี่ยนแปลงตำแหน่งของ Proxy/Cache เพื่อความเหมาะสม สามารถเปลี่ยนแปลงสีต่างๆ ได้ตามต้องการ สามารถเพิ่มหรือลดความสัมพันธ์ระหว่าง Proxy/Cache ที่มีต่อกันได้โดยง่าย
8. ผู้ใช้สามารถเปลี่ยนแปลงเพิ่มเติมขอบเขตการให้บริการของ Proxy/Cache (Access Control) ได้จากในภาพแสดงความสัมพันธ์ในข้อ 6 และแสดงให้เห็นเป็นภาพที่เข้าใจได้ง่าย การเปลี่ยนแปลงสามารถทำได้ด้วยวิธีการลากแล้วปล่อยเช่นเดียวกัน

### 3.5 กระบวนการทำงานของระบบ

จากที่ได้ทำการออกแบบไว้ข้างต้น สามารถนำมาเขียนเป็น Context Diagram ดังภาพที่ 3.3 โดย Context Diagram จะอธิบายให้ถึงระบบโดยรวมว่ามีการเชื่อมโยงติดต่อกับสิ่งที่อยู่ภายนอกคือ Proxy/Cache และผู้บริหารจัดการระบบ และมีการแตกออกเป็นกระบวนการย่อยในแผนภาพการไหลของข้อมูลระดับที่ 1 (Data Flow Diagram Level 1) ในภาพที่ 3.4



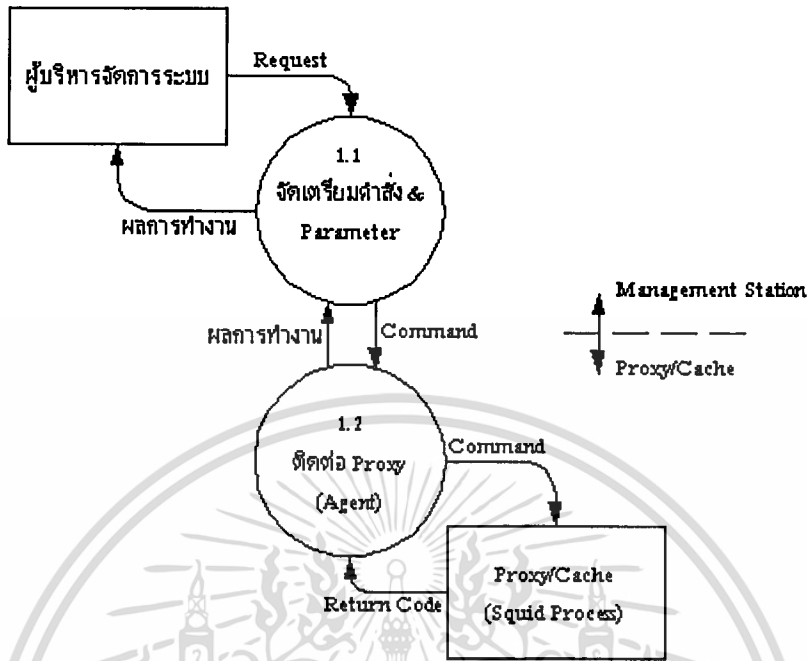
ภาพที่ 3.3 แผนภาพ Context Diagram ของระบบ



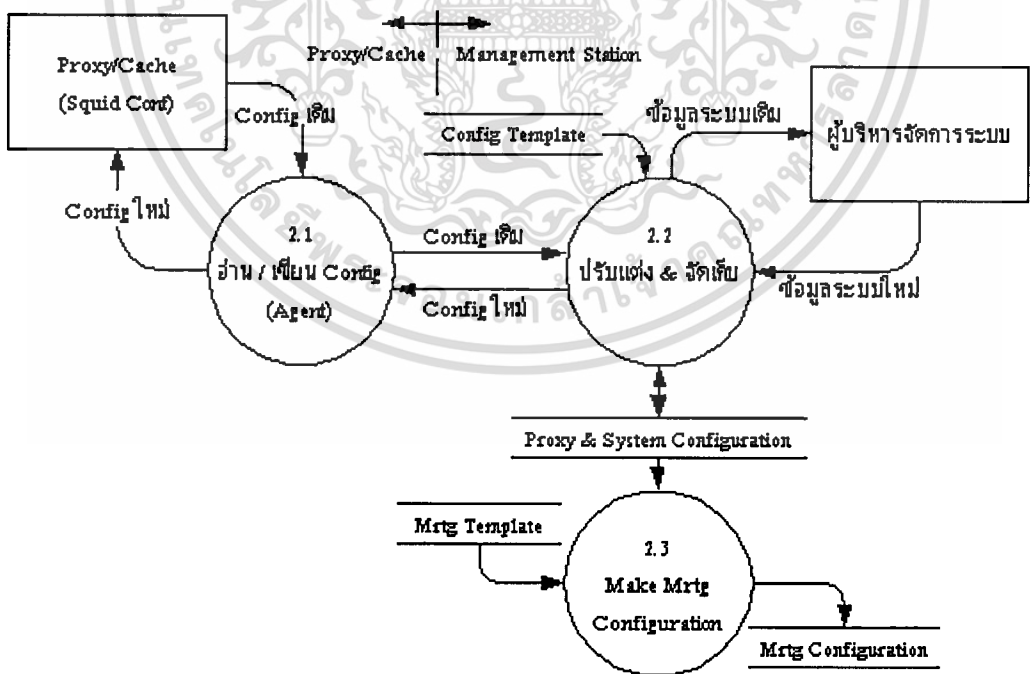
ภาพที่ 3.4 แผนภาพแสดงการไหลของข้อมูลระดับที่ 1 (DFD Level 1)

จากแผนภาพการไหลของข้อมูลระดับที่ 1 ระบบนี้สามารถแบ่งการทำงานออกเป็น 3 ส่วน คือ กระบวนการที่ 1 การสั่งงานโปรแกรมที่ Proxy/Cache ให้มีการทำงานตามคำสั่งโดยผู้บริหารจัดการระบบ กระบวนการที่ 2 การปรับปรุงแก้ไข Configuration ของ Proxy/Cache ซึ่งจะทำให้การแก้ไขข้อมูลที่ Management Station ก่อนที่จะทำการส่งไปยัง Proxy/Cache และในส่วนนี้จะมีการสร้าง Configuration ของ MRTG เพื่อใช้ในการตรวจสอบประสิทธิภาพในการทำงานของ Proxy/Cache ในกระบวนการที่ 3 อีกด้วย

ในกระบวนการที่ 1 จะเริ่มจากมีการสั่งงานจากผู้บริหารจัดการระบบ ส่งการร้องขอไปยังกระบวนการที่ 1.1 เพื่อจัดเตรียมคำสั่ง(Command) และ Parameter ตามที่ผู้ใช้ต้องการ ก่อนที่จะมีการส่งต่อไปให้ Agent ในกระบวนการที่ 1.2 เพื่อให้ Proxy/Cache ทำงานตามคำสั่งแล้วจัดการแปลง Return Code เป็นผลการทำงานส่งกลับไปให้ผู้บริหารจัดการระบบ แสดงดังภาพที่ 3.5



ภาพที่ 3.5 แผนภาพแสดงการไหลของข้อมูลระดับที่ 2 (DFD Level 2) ของกระบวนการที่ 1

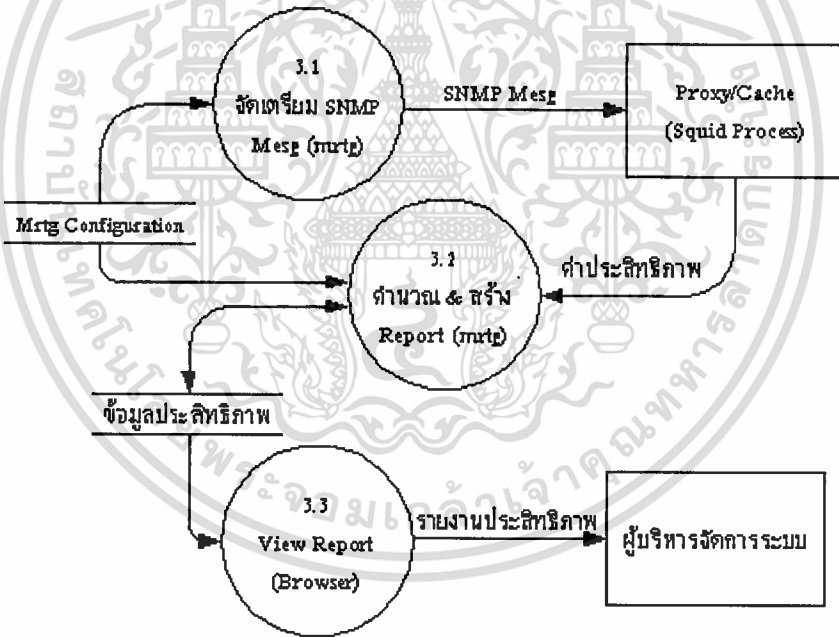


ภาพที่ 3.6 แผนภาพแสดงการไหลของข้อมูลระดับที่ 2 (DFD Level 2) ของกระบวนการที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกระบวนการที่ 2 สามารถอ่าน Configuration จาก Proxy/Cache ในกระบวนการที่ 2.1 มาเก็บไว้ที่ Management Station เพื่อทำการเรียกดูหรือแก้ไข โดยกระบวนการที่ 2.2 ซึ่งขบวนการนี้สามารถทำการแก้ไข Configuration ของ Proxy/Cache ได้พร้อมๆ กันหลายเครื่องได้ โดยเริ่มต้นอาจนำมาจาก Proxy/Cache โดยตรงหรือจาก Template ที่สร้างไว้ เมื่อแก้ไขเรียบร้อยแล้วก็จะทำการส่งค่ากลับไปยัง Proxy/Cache ต่อไป ส่วนกระบวนการที่ 2.3 จะทำการสร้าง MRTG Configuration จาก Mrtg Template และข้อมูล Proxy/Cache ที่ได้ปรับแต่งไว้ เพื่อใช้ในการตรวจสอบประสิทธิภาพของ Proxy/Cache ต่อไป แสดงดังภาพที่ 3.6

ในกระบวนการที่ 3 เป็นการทำงานของโปรแกรม Mrtg ซึ่งจะทำการอ่าน Mrtg Configuration จากที่ได้สร้างไว้จากขบวนการที่ 2 มาใช้ในการส่ง SNMP ไปยัง Proxy/Cache เพื่ออ่านค่าประสิทธิภาพแล้วมาทำการคำนวณและสร้างเป็นข้อมูลประสิทธิภาพเก็บไว้ เพื่อให้ผู้บริหารจัดการระบบเข้ามาเรียกดูได้ในภายหลัง แสดงดังภาพที่ 3.7

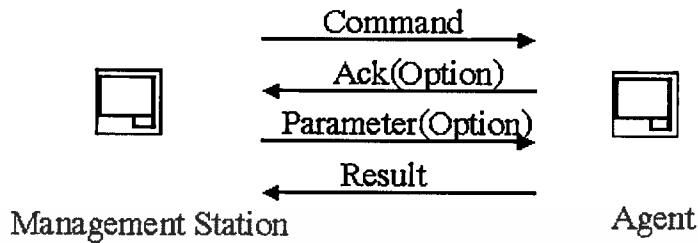


ภาพที่ 3.7 แผนภาพแสดงการไหลของข้อมูลระดับที่ 2 (DFD Level 2) ของกระบวนการที่ 3

**3.6 ข้อกำหนด (Request-Reply) ในติดต่อกับ Agent**

การติดต่อจะเป็นแบบ Client & Server โดยผ่าน TCP/IP Socket ที่หมายเลข Port ที่ผู้ใช้สามารถกำหนดเองได้ โดยเริ่มต้น Management Station จะเป็นฝ่ายร้องขอการติดต่อไปยัง Agent เมื่อ Agent ได้รับก็จะทำการตรวจสอบคำสั่ง หากเป็นคำสั่งที่ต้องการ Parameter เพิ่มเติมจะทำการตอบรับ(Ack Message) ไปยัง Management Station (Null Character) เพื่อรอ Parameter ที่จะส่งตามเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มา เมื่อได้รับข้อมูลครบถ้วนแล้วจึงทำงานตามคำสั่ง และส่งผลลัพธ์กลับไปยัง Management Station แล้วจึงทำการตัดการติดต่อ ดังภาพที่ 3.8



ภาพที่ 3.8 แผนภาพแสดงขั้นตอนการติดต่อกับ Agent

ซึ่งรายละเอียดของคำสั่งและ Parameter ที่ใช้ในการติดต่อกับ Agent มีดังนี้

1. การอ่านข้อมูล Configuration ทั้งหมดจาก Proxy/Cache ไปยัง Management Station  
คำสั่งที่ส่งไปยัง Agent ข้อความ *"READ\_ALL"*  
ค่าที่รับกลับ ข้อมูล Configuration ทั้งหมด ตามด้วยข้อความ *"END\_FILE"*
2. การเขียนข้อมูล Configuration ที่ Proxy/Cache ใหม่ทั้งหมดโดยนำมาจาก Management Station  
คำสั่งที่ส่งไปยัง Agent ข้อความ *"WRITE\_ALL"*  
Parameter ที่ส่ง ข้อมูล Configuration ทั้งหมด ตามด้วยข้อความ *"END\_FILE"*  
ค่าที่รับกลับ จำนวนบรรทัดของข้อมูล Configuration
3. การตรวจสอบหาเวลาครั้งสุดท้ายที่ข้อมูล Configuration มีการเปลี่ยนแปลง  
คำสั่งที่ส่งไปยัง Agent ข้อความ *"FILE\_STAT"*  
ค่าที่รับกลับ เวลาที่มีการเปลี่ยนแปลงข้อมูล Configuration ครั้งหลังสุด
4. การอ่านข้อมูล Configuration เฉพาะ Tag ที่กำหนดจาก Proxy/Cache ไปยัง Management Station  
คำสั่งที่ส่งไปยัง Agent ข้อความ *"READ\_PARA"*  
Parameter ที่ส่ง ชื่อ Tag ที่ต้องการ  
ค่าที่รับกลับ ค่า Parameter ของ Tag นั้น
5. การเพิ่มบรรทัดใหม่เข้าไปใน Configuration File ที่ Proxy/Cache  
คำสั่งที่ส่งไปยัง Agent ข้อความ *"ADD\_LINE"*  
Parameter ที่ส่ง *"Tag"* ที่เพิ่ม ตามด้วยบรรทัดที่ต้องการเพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่รับกลับ หมายเลขบรรทัดที่เพิ่ม

6. การเปลี่ยนแปลงเฉพาะบางบรรทัดใน Configuration File ที่ Proxy/Cache

คำสั่งที่ส่งไปยัง Agent ข้อความ *"REPL\_LINE"*

Parameter ที่ส่ง ข้อความบรรทัดที่ต้องการลบออก ตามด้วยข้อความบรรทัดใหม่ที่ต้องการเพิ่ม  
ค่าที่รับกลับ หมายเลขบรรทัดที่มีการเปลี่ยนแปลง

7. การลบบรรทัดใน Configuration File ที่ Proxy/Cache

คำสั่งที่ส่งไปยัง Agent ข้อความ *"DEL\_LINE"*

Parameter ที่ส่ง ข้อความบรรทัดที่ต้องการลบ

ค่าที่รับกลับ หมายเลขบรรทัดที่ลบ

8. การสั่งให้ Squid Process เปลี่ยนแปลงการทำงานตาม Configuration ใหม่

คำสั่งที่ส่งไปยัง Agent ข้อความ *"REST\_SQUID"*

ค่าที่รับกลับ ค่า Return Code ของคำสั่ง *"squid -k reconfigure"*

9. การสั่งให้ Squid Process ทำงานตามที่ต้องการด้วย Option ต่างๆ ของโปรแกรม Squid

คำสั่งที่ส่งไปยัง Agent ข้อความ *"SQUID\_CMD"*

Parameter ที่ส่ง ข้อความ *<Command Option>*

ค่าที่รับกลับ ค่า Return Code ของคำสั่ง *"squid <Command Option>"*

10. การสั่งให้ Proxy/Cache ทำงานตามที่ต้องการด้วยคำสั่งใดๆ ที่กำหนด

คำสั่งที่ส่งไปยัง Agent ข้อความ *"SYSTEM\_CMD"*

Parameter ที่ส่ง ข้อความ *<Command Option>*

ค่าที่รับกลับ ค่า Return Code ของคำสั่ง *"<Command>"*

11. เมื่อรับคำสั่งใดๆ ที่ไม่ถูกต้องตามที่กำหนดข้างต้น

ค่าที่รับกลับ ข้อความ *"Error Unknown Command"*

### 3.7 การสร้าง Configuration Files ของโปรแกรม Mrtg

โดยปรกติแล้วโปรแกรม Mrtg ถูกออกแบบมาให้ใช้งานการตรวจสอบเครือข่าย เช่น ปริมาณข้อมูลเข้าและออกจาก Router ที่ Interface ต่างๆ โดยใช้โปรโตคอล SNMP แล้วรายงานผลการตรวจสอบในรูปแบบกราฟ โดยใช้ Browser ในการเรียกดู การใช้งานจะต้องทำการสร้าง Configuration Files ซึ่งสามารถทำได้โดยง่าย ด้วยใช้ Script Program ที่ชื่อว่า "Indexmaker" แต่หากนำโปรแกรม Mrtg นี้มาใช้ในการตรวจสอบ OID ค่าอื่นๆ จะต้องทำการเขียน Configuration Files

ขึ้นใหม่เอง โดย File นี้จะต้องมีการระบุชื่อ OID, Community String, Host Address, Snmp Port ตามที่ต้องการ รวมทั้งจะต้องกำหนดรายละเอียดของการแสดงผลในแต่ละ OID ตัวอย่างเช่น

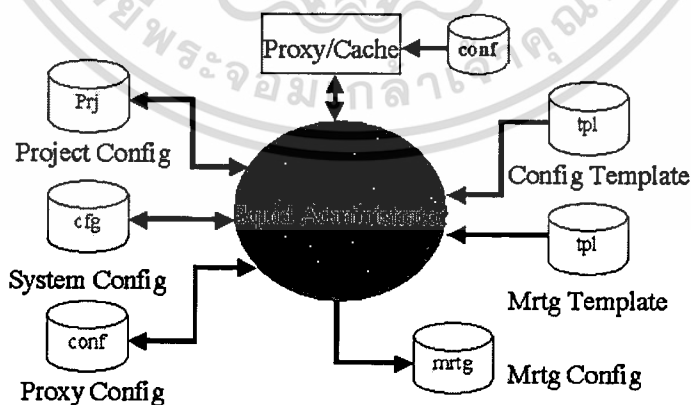
```
Target[wallHttpInKb]: cacheHttpInKb&cacheHttpInKb:public@wall.diw.go.th:3401
MaxBytes[wallHttpInKb]: 100000
Title[wallHttpInKb]: cacheHttpIn
Options[wallHttpInKb]: perhour
PageTop[wallHttpInKb]: <h1>cacheHttpIn @ wall</h1>
YLegend[wallHttpInKb]: Bytes per hour
ShortLegend[wallHttpInKb]: B/h
kMG[wallHttpInKb]: k,M,G,T,P,
LegendI[wallHttpInKb]: HttpIn
LegendO[wallHttpInKb]:
Legend1[wallHttpInKb]: cacheHttp In in Bytes per Hour
Legend2[wallHttpInKb]:
```

ซึ่งในที่นี้เป็นการตรวจสอบค่า cacheHttpInKb ที่เครื่อง "wall.diw.go.th" และ Port "3401" โดยใช้ Community String "Public" กำหนดค่าที่แสดงเป็นค่าเฉลี่ยต่อชั่วโมง ในหน่วย Bytes per hour

ในที่นี้จะมี OID ที่ต้องตรวจสอบทั้งหมด 18 ค่า ต่อ Proxy/Cache 1 เครื่อง นอกจากนั้น จะต้องกำหนดค่า Option ต่างๆ เช่นตำแหน่งของที่เก็บข้อมูลและสีของกราฟ ที่ส่วนหัวของ Files อีกด้วย ตัวอย่างของ Configuration File ของโปรแกรม Mrtg ได้แสดงอยู่ในภาคผนวก ก.

นอกจากนั้นเพื่อความสะดวกในการเรียกดูข้อมูลประสิทธิภาพ จะสร้าง Html File สำหรับ Proxy/Cache ขึ้นมาด้วยอีก File หนึ่ง เพื่อให้ผู้ใช้สามารถคลิกเลือกดูรายละเอียดได้ตามต้องการ

### 3.8 รูปแบบแฟ้มข้อมูลของโปรแกรม



ภาพที่ 3.9 โครงสร้างการเก็บข้อมูลของระบบ

โปรแกรมนี้ได้ออกแบบการเก็บข้อมูลอยู่ในรูป File ทั้งหมด เนื่องจากเดิม Configuration เก็บอยู่ในรูป Files อยู่ก่อนแล้ว โดยจะใช้รูปแบบเดิมในการเก็บข้อมูล Proxy/Cache แต่ละเครื่อง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Proxy Config ในภาพที่ 3.9) นอกจากนั้นจะต้องมีเพิ่มเก็บข้อมูลของระบบอีกด้วย(System Config) และเพื่อความสะดวกมีการจะกำหนด Folder สำหรับเก็บข้อมูลไว้ที่เดียวกันทั้งหมด รวมเป็น 1 โครงการ(Project Config) ซึ่งเพิ่มเก็บข้อมูลมีรายละเอียดดังนี้

1. เพิ่มเก็บข้อมูลโครงการ จะมีนามสกุลเป็น ".prj" เพิ่มนี้จะใช้เก็บข้อมูล Folder ในการทำงาน (คือตำแหน่งที่เก็บข้อมูลอื่นๆ ทั้งหมด) และยังมีชื่อและตำแหน่งเพิ่มเก็บข้อมูลระบบอีกด้วย รูปแบบในการเก็บคือ

WorkPath [Working Directory]

ConfFile [Path Name of System Configuration Files]

2. เพิ่มเก็บข้อมูลระบบ จะมีนามสกุลเป็น ".cfg" เพิ่มนี้ได้ถูกออกแบบให้มีลักษณะคล้ายกับการเก็บข้อมูล Configuration ของโปรแกรม Squid Proxy/Cache ก็จะมีลักษณะเป็น Tag แล้วตามด้วย Parameter ต่างๆ ในแต่ละบรรทัดโดยใช้ “ “ เป็นตัวแบ่ง โดยลำดับของบรรทัดไม่มีความสำคัญ ซึ่งมีรายละเอียดของ Tag แสดงดังตารางที่ 3.2 พร้อมกับแสดงตัวอย่างของเพิ่มนี้ในภาคผนวก ข.

ตารางที่ 3.2 ความหมายของ Tag ในเพิ่มเก็บข้อมูลระบบ

TAG	รายละเอียด
Mrtg_work_dir	เป็น Directory การทำงานของโปรแกรม Mrtg
Mrtg_prog	ชื่อและ Path ของโปรแกรม mrtg
Mrtg_perf_url	Url ของการใช้ Browser ดูประสิทธิภาพ
Mrtg_community	Community String ที่ใช้ในการติดต่อด้วยโปรโตคอล SNMP
Mrtg_run	ความต้องการให้โปรแกรม Mrtg ทำงานทุก ๆ 5 นาที ซึ่งมีค่าเป็น True หรือ False เท่านั้น
Picture_size	ขนาดความกว้างและความสูงของภาพแสดงความสัมพันธ์ ซึ่งจะเปลี่ยนแปลงตามที่ใช้ต้องการ
Picture_BG_color	สีพื้นของภาพแสดงความสัมพันธ์ซึ่งเก็บเป็นเลขฐาน 16
Picture_RP_color	สีฉลากของ Proxy/Cache ในภาพแสดงความสัมพันธ์ซึ่งเก็บเป็นเลขฐาน 16
Picture_RC_color	สีฉลากของ Client ในภาพแสดงความสัมพันธ์ซึ่งเก็บเป็นเลขฐาน 16
Picture_RL_color	สีฉลากของเส้นความสัมพันธ์ในภาพแสดงความสัมพันธ์ซึ่งเก็บเป็นเลขฐาน 16

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้วยการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TAG	รายละเอียด
Cache	ข้อมูลทั่วไปของ Proxy/Cache ซึ่งข้อมูลนี้จะใช้บอกถึงชื่อ Proxy/Cache ที่มีอยู่ในระบบทั้งหมดอีกด้วย
Cache_position	ตำแหน่งที่ตั้งของ Proxy/Cache ในภาพแสดงความสัมพันธ์
Cache_relation	ข้อมูลความสัมพันธ์ระหว่าง Proxy to Proxy ซึ่งจะประกอบด้วยชื่อ Proxy/Cache ทั้ง 2 ชื่อ ตามด้วยแบบของความสัมพันธ์ (parent หรือ sibling) หมายเลข Port และ Option ต่างๆ
Client	ข้อมูลชื่อและ IP Address ของขอบเขตการให้บริการของ Proxy/Cache
client_position	ตำแหน่งที่ตั้งของ Client ในภาพแสดงความสัมพันธ์
client_relation	ข้อมูลความสัมพันธ์ระหว่าง Client to Proxy ซึ่งจะประกอบด้วยชื่อ Client ชื่อ Proxy/Cache ตามด้วยแบบของความสัมพันธ์ (http หรือ icp) และหมายเลข Port

3. เพิ่มเก็บข้อมูล Configuration ของแต่ละ Proxy/Cache ซึ่งใช้ชื่อตามชื่อของ Proxy/Cache และมีนามสกุลเป็น ".conf" เพิ่มนี้มีรูปแบบเช่นเดียวกับ Configuration ของ Proxy/Cache ซึ่งพร้อมที่จะนำไปใช้เป็น Configuration File ของโปรแกรม Squid ได้ทันที แต่สิ่งที่ควรระวังคือ Text File ในระบบ Unix จะแตกต่างจากระบบของ Windows อยู่บ้าง

นอกจากเพิ่มเก็บข้อมูลข้างต้นแล้ว โปรแกรมยังมีแม่แบบ(Template) สำหรับการนำไปใช้งานอีก 4 แม่แบบด้วยกันคือ

1. แม่แบบสำหรับเพิ่มเก็บข้อมูลระบบที่สร้างขึ้นใหม่ จะใช้ชื่อเป็น "cfg.tpl"
2. แม่แบบสำหรับการเพิ่ม Proxy/Cache ใหม่เข้ามาในระบบโดยการนำค่า Default ของโปรแกรมมาใช้ ซึ่งเพิ่มนี้จะใช้ชื่อเป็น "conf.tpl"
3. แม่แบบสำหรับการสร้าง Configuration ของโปรแกรม Mrtg (ใช้ชื่อเป็น "mrtg.tpl")
4. แม่แบบสำหรับการสร้าง Html Document ของโปรแกรม Mrtg (ใช้ชื่อเป็น "html.tpl")

ซึ่งการใช้งานแม่แบบนี้ ออกแบบให้มีการสร้างเพิ่มขึ้นใหม่โดยทำการคัดลอกจากแม่แบบ ไปเก็บไว้ในที่ที่เหมาะสม และโดยเฉพาะ "mrtg.tpl" และ "html.tpl" นั้นจะต้องมีการเปลี่ยนแปลงข้อมูลภายในแม่แบบ ก่อนที่จะนำไปใช้เป็น Configuration ของโปรแกรม MRTG ดังตารางที่ 3.3

### ตารางที่ 3.3 การเปลี่ยนแปลงเพิ่มเติมแบบเมื่อนำไปใช้

ค่าเดิม	ค่าที่เปลี่ยนใหม่
<!--mrtg_work_dir>	ค่าของ mrtg_work_dir ซึ่งนำมาจากเพิ่มเก็บข้อมูลระบบ
<!--community_string>	ค่าของ mrtg_community ซึ่งนำมาจากเพิ่มเก็บข้อมูลระบบ
<!--host_name>	ค่าของ Host Name ซึ่งเป็นชื่อของ Proxy/Cache ที่เลือก
<!--host_address>	ค่าของ Host Address ของ Proxy/Cache ที่เลือก
<!--snmp_port>	ค่าของ snmp_port ของ Proxy / Cache ที่เลือก

### 3.9 ส่วนติดต่อผู้ใช้งาน (User Interface)

ส่วนติดต่อกับผู้ใช้งานได้ออกแบบให้สามารถใช้งานได้ง่าย (User Friendly) ผู้ใช้สามารถสั่งงานได้จาก Poll-Down เมนู แถบเครื่องมือหรือแม้กระทั่ง Pop-up เมนู และจะมี Status Bar แสดงสถานะการทำงานที่ส่วนล่างของโปรแกรม แสดงดังภาพที่ 3.10



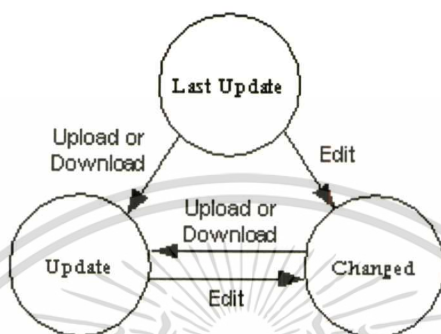
ภาพที่ 3.10 หน้าจอหลักของโปรแกรม

ในส่วนกลางของโปรแกรมจะเป็นช่องแสดงรายการของเครื่อง Proxy/Cache ที่เราสนใจในระบบทั้งหมด โดยแบ่งออกเป็น 5 ช่อง ซึ่งมีรายละเอียดดังนี้

1. ช่อง Proxy / Cache จะเป็นแสดงชื่อที่ใช้เรียก Proxy/Cache (ชื่อ Alias) ซึ่งเราจะใช้ชื่อนี้สำหรับการติดต่อกับโปรแกรม
2. ช่อง Config เป็นช่องที่แสดงสถานะการเปลี่ยนแปลงข้อมูล Configuration ของ Proxy/Cache ที่ Management Station โดยจะมีสถานะเป็นดังภาพที่ 3.11
  - "Last Update" เกิดในกรณีที่โปรแกรมขาดการติดต่อกับ Proxy/Cache เช่นการปิดโปรแกรม ข้อมูล Configuration ที่อยู่ ณ Management Station จะเป็นข้อมูลท้ายสุดที่มีการติดต่อกัน

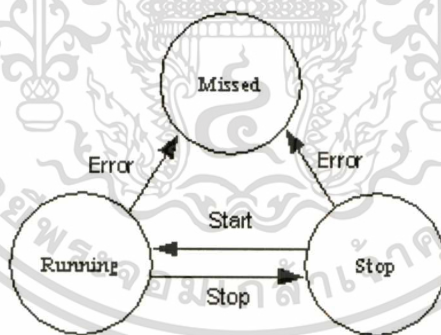
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- "Update" แสดงว่าข้อมูล Configuration ที่อยู่ ณ. Management Station มีความทันสมัย เป็นข้อมูลเดียวกับข้อมูล Configuration ใน Proxy/Cache โดยการ Upload หรือ Download
- "Changed" แสดงว่าข้อมูล Configuration ที่อยู่ ณ. Management Station มีการแก้ไขเปลี่ยนแปลงไปจากเดิม



ภาพที่ 3.11 สถานะของข้อมูล Configuration ที่ Management Station

3. ช่อง Status เป็นช่องที่แสดงสถานะการทำงานของ Proxy/Cache ซึ่งจะมีการเปลี่ยนแปลงสถานะเป็นดังภาพที่ 3.12



ภาพที่ 3.12 สถานะการทำงานของ Proxy/Cache

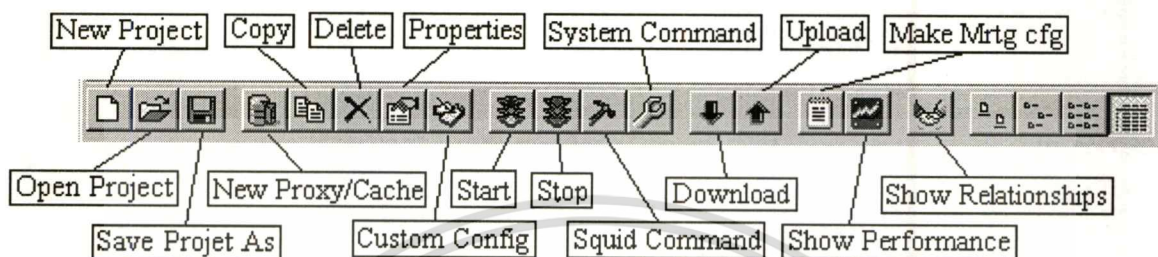
- "Running" หมายถึงขณะนั้นเครื่อง Proxy/Cache มีการให้บริการอยู่
- "Stop" หมายถึงเครื่อง Proxy/Cache ได้มีการหยุดให้บริการ
- "Missed" หมายถึงไม่สามารถติดต่อไปยัง Proxy/Cache โดยผ่าน โปรแกรม Agent ได้

4. ช่อง Host Address จะเป็นข้อมูลชื่อของ Proxy/Cache (Host Name) ที่สามารถอ้างอิงได้ด้วยระบบ Domain Name System หรืออาจใช้เป็น IP Address ของ Proxy/Cache ก็ได้

5. ช่อง Port เป็นข้อมูลหมายเลข Port ที่จะใช้ติดต่อกับ Proxy / Cache โดยผ่าน โปรแกรม Agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนแถบเครื่องมือจะมีลักษณะเป็นปุ่มรูปภาพที่มีความหมาย และผู้ใช้มีความคุ้นเคยอยู่ก่อนแล้ว เพื่อให้ผู้ใช้สามารถใช้งานคำสั่งได้ด้วยความรวดเร็ว โดยไม่จำเป็นต้องใช้ Pull-Down เมนูหรือ Pop-up เมนู ซึ่งแต่ละปุ่มมีความหมายดังภาพที่ 3.13

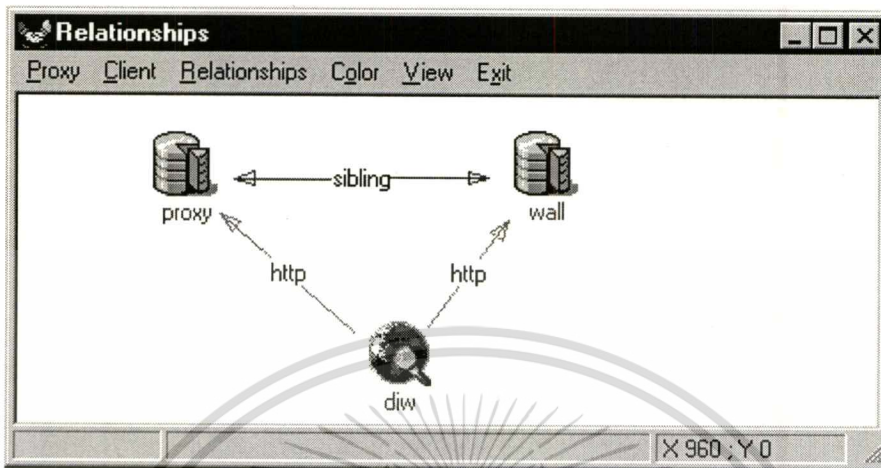


ภาพที่ 3.13 แถบเครื่องมือของหน้าจอหลัก

แถบเครื่องมือนี้สามารถแบ่งได้เป็น 7 ส่วนด้วยกัน คือ ส่วนแรกสำหรับการทำงานกับโครงการ(Project) ส่วนที่สองเป็นการปรับปรุงแก้ไขข้อมูลระบบ ส่วนที่สามเป็นการสั่งการทำงานกับ Proxy/Cache ส่วนที่สี่เป็นการรับส่งข้อมูล Configuration ส่วนที่ห้าเป็นการทำงานร่วมกับโปรแกรม Mrtg ในการตรวจสอบประสิทธิภาพ ส่วนที่หกเป็นภาพแสดงความสัมพันธ์ และส่วนสุดท้ายเป็นการปรับเปลี่ยนมุมมองในช่องแสดงรายการ

ส่วนการแสดงความสัมพันธ์ จะออกแบบให้ผู้ใช้สามารถทำความเข้าใจกับระบบได้ง่าย และสามารถทำการเปลี่ยนแปลง Configuration จากในภาพได้ทันที โดย Proxy/Cache จะแทนด้วยรูปภาพพร้อมชื่อด้านล่าง ส่วนความสัมพันธ์จะแสดงด้วยลูกศรมิติศทางพร้อมมีฉลากแสดงชนิดของความสัมพันธ์กำกับ ส่วนขอบเขตการให้บริการของ Proxy/Cache จะแทนด้วยภาพลูกโลก ดังภาพที่ 3.14

เมื่อ User ทำการเพิ่มหรือลดความสัมพันธ์ จะมีการเปลี่ยนแปลง Configuration ในแต่ละ Proxy/Cache ที่เกี่ยวข้อง โดยจะเปลี่ยนในส่วนของ Access Control List และ cache\_peer ตามความเหมาะสม



ภาพที่ 3.14 หน้าจอแสดงภาพความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การพัฒนาระบบงาน

เมื่อเสร็จสิ้นขั้นตอนการวิเคราะห์และออกแบบแล้ว ขั้นตอนต่อไปคือการนำเอาสิ่งที่ออกแบบไว้มาพัฒนาเป็นโปรแกรมจริง ซึ่งมีรายละเอียดดังต่อไปนี้

#### 4.1 การพัฒนาโปรแกรม Agent

โปรแกรมนี้มีการทำงานที่ Proxy/Cache ซึ่งอยู่บนระบบปฏิบัติการ Unix จึงใช้โปรแกรม GNU C++ ในการพัฒนา โดยเริ่มพัฒนาบน Linux ลักษณะโปรแกรมจะเป็นการติดต่อกับเครือข่ายและทำงานกับการเปลี่ยนแปลง File โดยส่วนใหญ่ จึงใช้ Module เกี่ยวกับ Socket, File Stream และ String เข้ามาพร้อมด้วย (Source Code แสดงอยู่ในภาคผนวก ค.)

ในการแก้ไข Configuration File ของ Squid ผู้พัฒนามีความต้องการให้มีการเปลี่ยนแปลงน้อยที่สุดเช่น ลำดับของบรรทัดและการ Comment ต่างๆ โปรแกรมนี้จะมี Configuration File ของการทำงานเช่นกัน ซึ่งสามารถระบุได้ในตอนเริ่มต้นโปรแกรมที่ Command Line และมีค่า Default เป็น file "agent.conf" ตัวอย่างเช่น "#agent agent.conf" ลักษณะของ File นี้เป็น TAG และ Parameter เช่นกัน แสดงดังตารางที่ 4.1

ตารางที่ 4.1 ความหมายใน Configuration File สำหรับ Agent

TAG	ความหมาย
squid_bin	Path Name ของ โปรแกรม squid ที่จะใช้ Run "Squid Command"
squid_conf	Path Name ของ Configuration File ของโปรแกรม squid ที่จะทำการแก้ไขต่อไป
admin_host	ชื่อของ Host ที่ Agent ยอมให้มีการติดต่อด้วย หากกำหนดเป็น "any" จะหมายถึงเครื่องใดก็ได้ และระบบจะมีความปลอดภัยลดลง
agent_port	หมายเลข Port ที่ใช้ในการติดต่อ (ค่า Default=2023)

## 4.2 การติดตั้งเครื่องมือ(Tools) ต่างๆ ที่ใช้

การพัฒนาในขั้นตอนนี้ได้ทำการ Download โปรแกรม Active Perl และ Mrtg เพื่อนำมาติดตั้งบนเครื่อง Management Station โดยสามารถ Download ได้จาก Website "<http://www.activestate.com>" และ "<http://www.mrtg.org>" ตามลำดับ สำหรับการติดตั้งโปรแกรม Active Perl สามารถทำได้ทันทีโดยทำตามขั้นตอนที่โปรแกรมแนะนำ ส่วนโปรแกรม Mrtg เมื่อ Download มาจะอยู่ในรูปแฟ้ม ".zip" จึงต้องทำการเปิดด้วยโปรแกรม Winzip จากนั้นจึงนำ Files ใน Folder "run" ซึ่งเป็นส่วนของโปรแกรม Mrtg ที่ต้องใช้ Extract ไปเก็บไว้ (ในที่นี้จะเก็บไว้ใน Folder "D:\Perl\mrtg") โปรแกรม Mrtg นี้จะต้องมีการแก้ไขเล็กน้อย โดยการเปลี่ยนแปลง File "mrtg" ซึ่งเป็น Text File ดังตารางที่ 4.2

ตารางที่ 4.2 การแก้ไขโปรแกรม Mrtg

ข้อความเดิม	แก้ไขเป็นข้อความใหม่
\$main::OS = 'UNIX';	#\$main::OS = 'UNIX';
#\$main::OS = 'NT';	\$main::OS = 'NT';
\$main::binpath = "";	\$main::binpath = "D:\perl\mrtg";

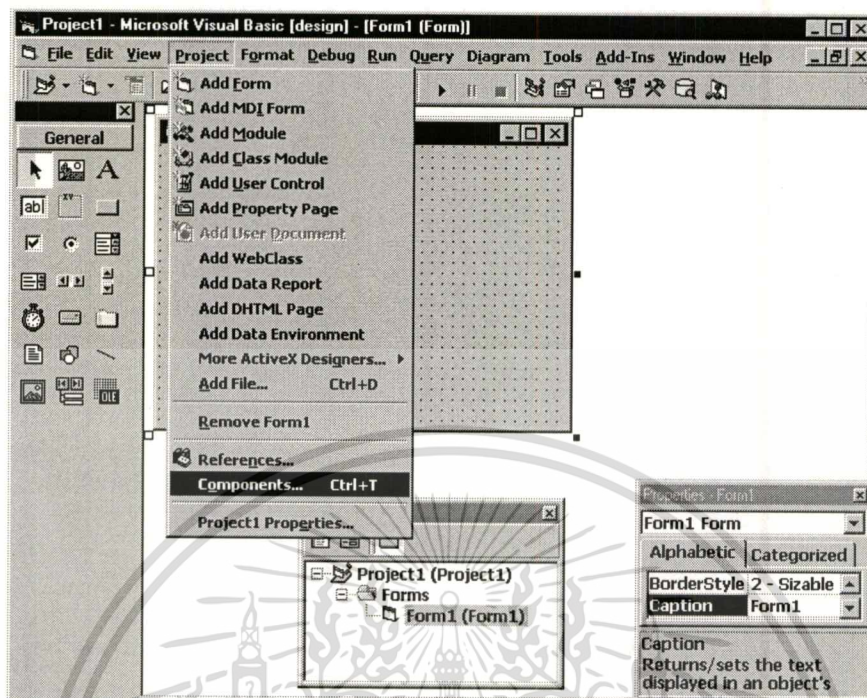
## 4.3 การพัฒนาโปรแกรม Squid Admin

ส่วนนี้มีการเน้นเรื่องการติดต่อผู้ใช้งานที่เป็น GUI สามารถเข้าใจได้ง่าย ได้ใช้โปรแกรม Microsoft Visual Basic Version 6.0 (Service Pack3) ในการพัฒนา และเนื่องจากโปรแกรมนี้อาจต้องมีส่วนของการตรวจสอบประสิทธิภาพด้วย SNMP โพรโทคอล จึงได้นำ Component "SNMP tool" เข้ามาติดตั้งใช้ประกอบด้วย ซึ่ง Component นี้สามารถ Download ได้ที่ Website "<http://www.dart.com>"

หลังจากติดตั้ง Component เรียบร้อยแล้ว จึงทำการสร้าง Project ใหม่แบบ "Standard EXE" ด้วย Microsoft Visual Basic ซึ่งต้องทำการเพิ่ม Component เข้ามาใน Project ด้วยโดยใช้เมนู "Project" → "Components..." แสดงดังภาพที่ 4.1 แล้วเลือกทำเครื่องหมายที่ Component ที่ต้องการ ซึ่งนอกจาก "Dart Snmp Controls" แล้วยังต้องมีการเพิ่ม Component ดังต่อไปนี้ "Microsoft Common Dialog Control 6.0", "Microsoft Windows Common Controls 6.0" และ "Microsoft Winsock Control 6.0"

หลังจากนั้นจึงเริ่มทำการลาก Control ต่างๆ มาไว้ใน Form พร้อมเขียน Code ตามที่ได้ ออกแบบการทำงานไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.1 การเพิ่ม Component ต่างๆ เขามาใน Project

ขั้นตอนต่อไปเป็นเขียนโปรแกรม โดยการลาก Control ต่างๆ ลงบน Form แล้วเขียน Code ตามที่ได้ออกแบบไว้ และหากต้องการเพิ่ม Form ใหม่เข้ามาใน Project สามารถใช้เมนู "Project" → "Add Form" (โปรแกรมที่พัฒนานี้มี Form ทั้งหมด 16 Form โดยมี "FormAdmin" เป็น Form หลัก และมี "RelaForm" เป็น Form สำหรับแสดงภาพความสัมพันธ์)

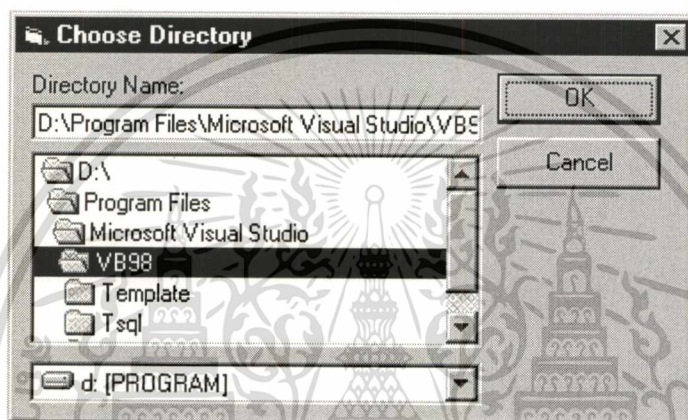
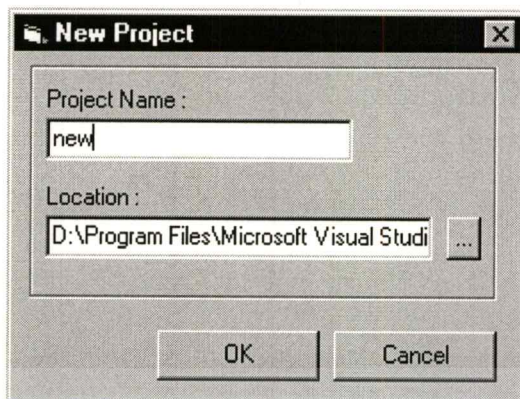
#### 4.4 วิธีการทำงานกับโปรแกรม

ในตอนเริ่มต้นจะต้องมีการสร้าง Project ขึ้นใหม่ โดยใช้เมนูคำสั่ง "New Project" จาก Pull-Down เมนู หรือจากแถบเครื่องมือ ซึ่งการสร้าง Project ใหม่จะมีการสร้างเพิ่มข้อมูลโครงการขึ้นใหม่ โดยจะต้องมีการระบุชื่อของโครงการ และตำแหน่งที่ใช้เก็บข้อมูล ดังภาพที่ 4.2 ซึ่งตำแหน่งที่ใช้เก็บข้อมูลสามารถคปุม่ "... เพื่อ Browse เลือก Folder ที่ต้องการได้

และในกรณีที่มี Project เดิมอยู่สามารถใช้เมนูคำสั่ง "Open Project" เพื่อที่จะทำการแก้ไข ข้อมูลที่ทำการบันทึกเก็บไว้ได้

หลังจากนั้นหากต้องการเพิ่ม Proxy/Cache เข้ามาในโปรแกรมสามารถทำได้โดยการใช้เมนูคำสั่ง "New Proxy" ซึ่งการเพิ่มนี้จะมีการสร้างเพิ่มเก็บข้อมูล Configuration ของแต่ละ Proxy/Cache ขึ้นใหม่ โดยผู้ใช้สามารถเลือกการนำข้อมูลจาก 3 แหล่งด้วยกันคือ

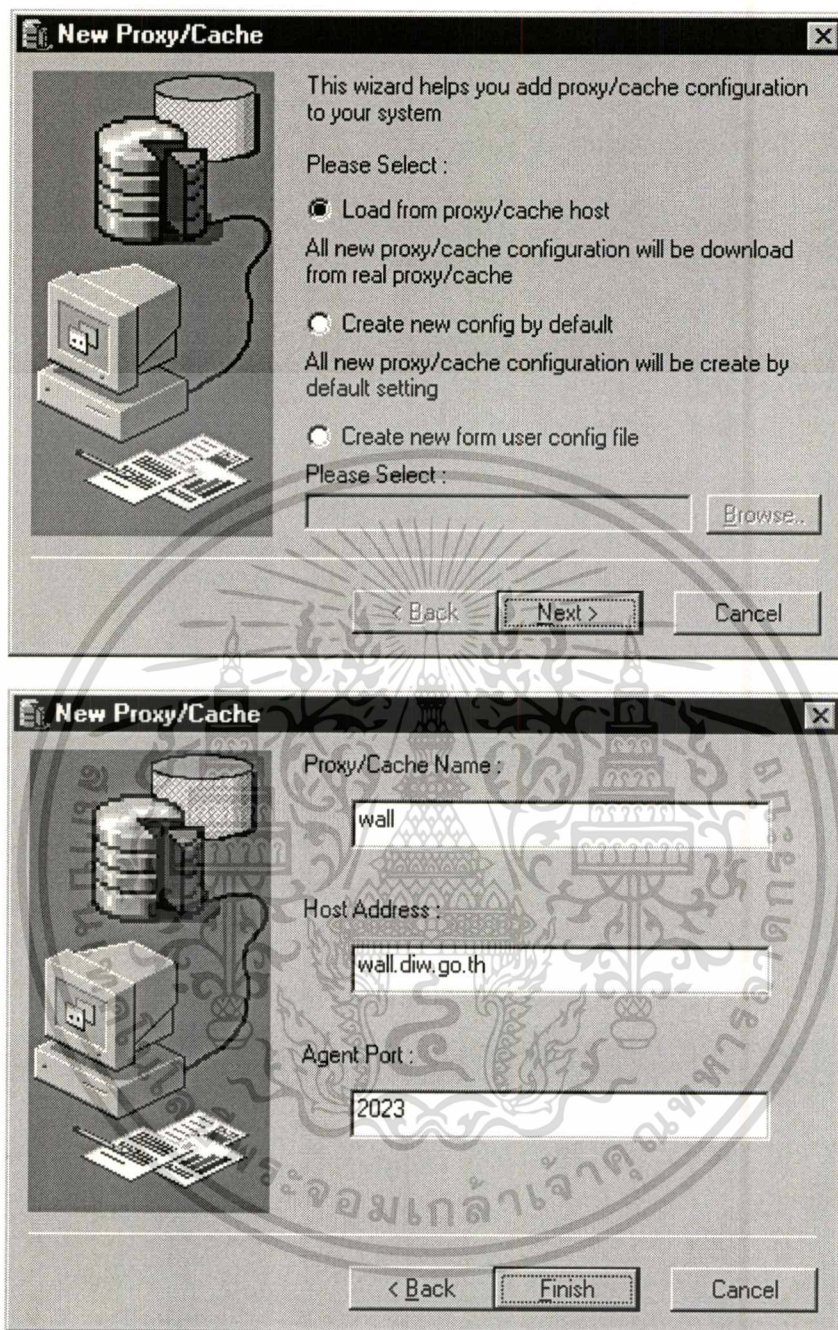
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.2 หน้าจอการสร้าง Project ใหม่

1. การ Download ข้อมูล Configuration จาก Proxy/Cache
2. การนำค่า Default ของโปรแกรมมาใช้
3. การนำเข้าจาก File ตามที่ผู้ใช้งานกำหนด

ผู้ใช้งานจะต้องมีการระบุชื่อ Proxy/Cache และ Host Address ซึ่งแน่นอนจะต้องไม่ซ้ำกับเครื่องอื่นๆ ที่มีอยู่เดิม อีกทั้งยังต้องมีการระบุ Agent Port ที่จะใช้ในการติดต่อสื่อสารอีกด้วย การทำงานนี้จะเป็นแบบ "Wizard" หากต้องการยกเลิกสามารถกดที่ปุ่ม "Cancel" ได้ดังภาพที่ 4.3 เมื่อการทำงานเสร็จสิ้นจะมี Proxy/Cache เพิ่มเข้ามาช่องแสดงรายการที่ส่วนกลางของโปรแกรม ซึ่งพร้อมที่จะใช้บริหารจัดการต่อไป



ภาพที่ 4.3 หน้าจอการเพิ่ม Proxy/Cache เข้ามาในโปรแกรม

ขั้นตอนต่อไปเราสามารถเรียกดูหรือทำการแก้ไขเปลี่ยนแปลงเพิ่มข้อมูล Configuration ของ Proxy/Cache ได้โดยใช้เมนูคำสั่ง "Properties" ซึ่งได้นำ Tag ที่สำคัญมารวบรวมแสดงเป็น Dialog ซึ่งสามารถแบ่งได้เป็น 5 ส่วน คือข้อมูลทั่วไป ข้อมูลรายละเอียด ข้อมูลความสัมพันธ์ การกำหนดชื่อ Acl และสุดท้าย Access Control ดังภาพที่ 4.4 – ภาพที่ 4.8 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AcI Name | Access Control

General | Detail | Cache Peer

Proxy/Cache Name :

wall

Host Address :

wall.diw.go.th

Agent Port :

2023

OK Cancel Apply

ภาพที่ 4.4 หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลทั่วไป

AcI Name | Access Control

General | Detail | Cache Peer

http\_port : 3128 icp\_port : 3130

snmp\_port : 3401

cache\_mem : 8 MB

cache\_dir

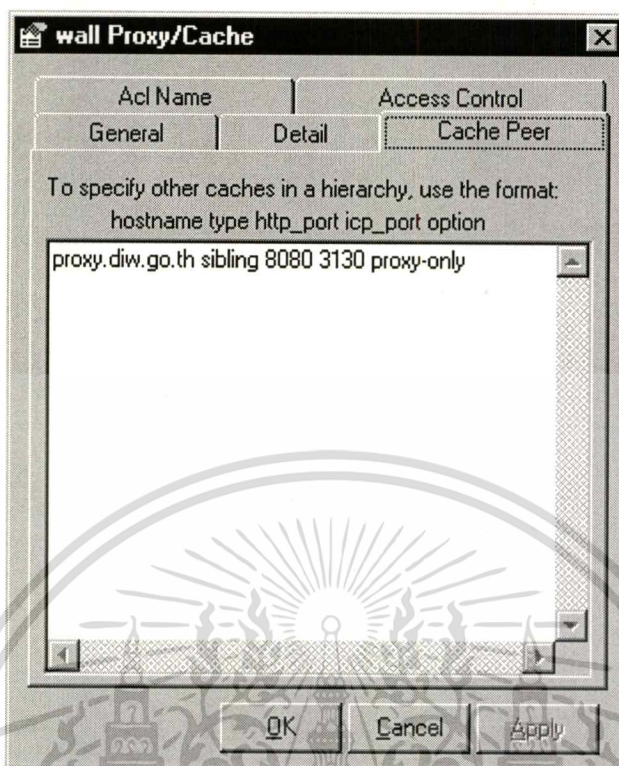
Usage: Directory-Name	Mbytes	Level-1	Level-2
/var/spool/squid	100	16	256

maximum\_object\_size : 4096 KB

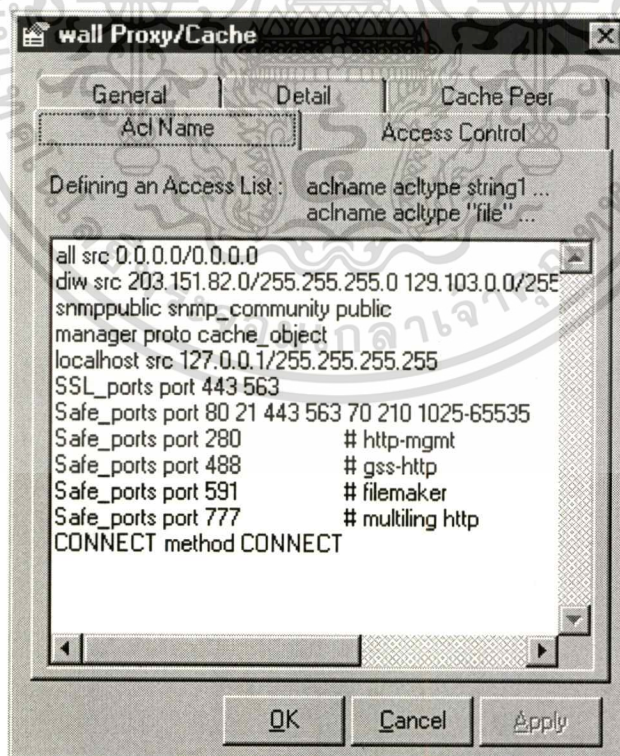
OK Cancel Apply

ภาพที่ 4.5 หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

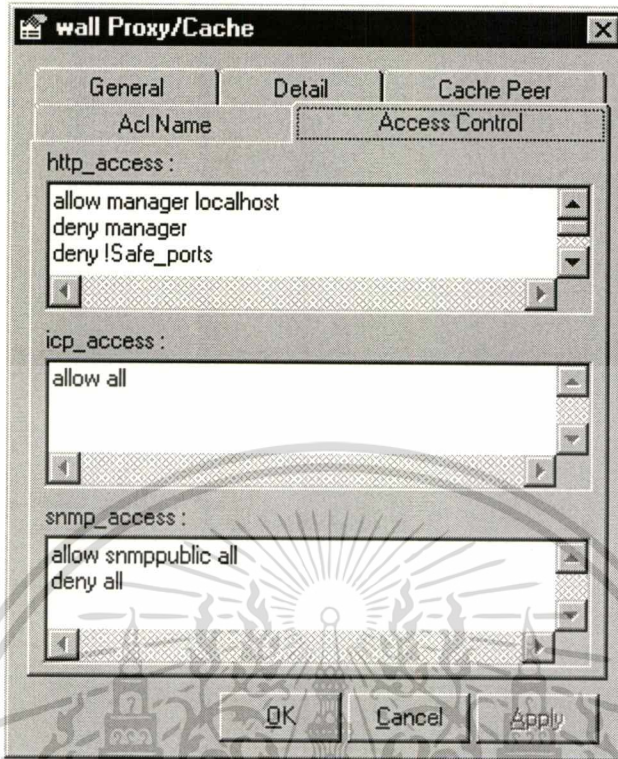


ภาพที่ 4.6 หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลความสัมพันธ์



ภาพที่ 4.7 หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลชื่อ Acl

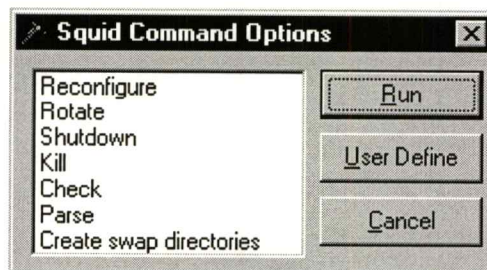
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.8 หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูล Access Control

ในช่องที่เป็นหมายเลข Port จะใส่ได้แต่เพียงตัวเลขเท่านั้น เมื่อมีการแก้ไขเปลี่ยนแปลงในช่องใดๆ ตัวอักษรจะเปลี่ยนเป็นสีแดงเพื่อให้ผู้ใช้สามารถจำได้ และหากผู้ใช้ต้องการเปลี่ยนแปลงข้อมูล Configuration ใน Tag อื่นๆ ด้วยตนเอง สามารถทำได้ด้วยเมนูคำสั่ง "Custom Config" ซึ่งเมนูคำสั่งนี้จะทำการแก้ไขข้อมูล Configuration โดยผ่าน โปรแกรม Notepad หรือ Wordpad ของ Windows

เมนูคำสั่ง "Start" และ "Stop" ใช้สำหรับการสั่งให้ Proxy/Cache เริ่มและหยุดการให้บริการตามลำดับ ส่วนเมนูคำสั่ง "Squid Command" ใช้ในการสั่งให้โปรแกรม Squid ทำงานตามต้องการ โดยจะมีตัวเลือกให้หรือผู้ใช้สามารถกำหนดเองก็ได้ ดังภาพที่ 4.9



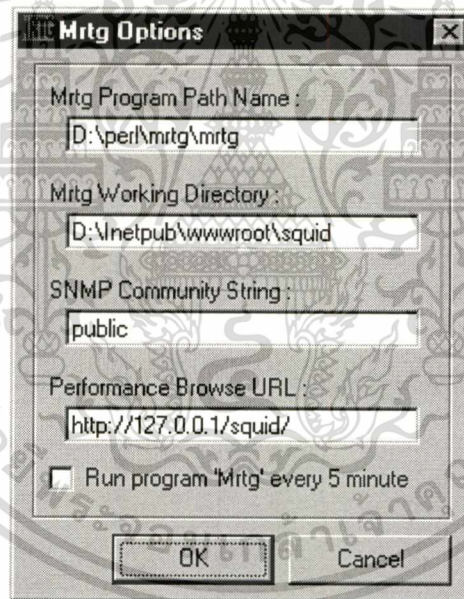
ภาพที่ 4.9 หน้าจอตัวเลือกของคำสั่ง Squid Command

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้งานสามารถกำหนดให้ Proxy/Cache ทำงานคำสั่งใดๆ ก็ได้ตามต้องการด้วยคำสั่ง "System Command" ซึ่งผู้ใช้งานต้องมีความเข้าใจในการทำงานคำสั่งนั้นเป็นอย่างดี เพราะผู้ใช้งานทราบผลแต่เพียงว่าการทำงานนั้นเป็นสำเร็จหรือไม่

เมื่อทำการแก้ไขข้อมูล Configuration ของ Proxy/Cache เสร็จเรียบร้อยแล้ว สามารถนำ Configuration นั้นไปติดตั้งใช้งานด้วยการใช้เมนูคำสั่ง "Upload" แต่ในทางตรงข้ามหากต้องการนำเข้าสู่ข้อมูลจาก Proxy/Cache ใหม่ก็สามารถใช้คำสั่ง "Download" ได้

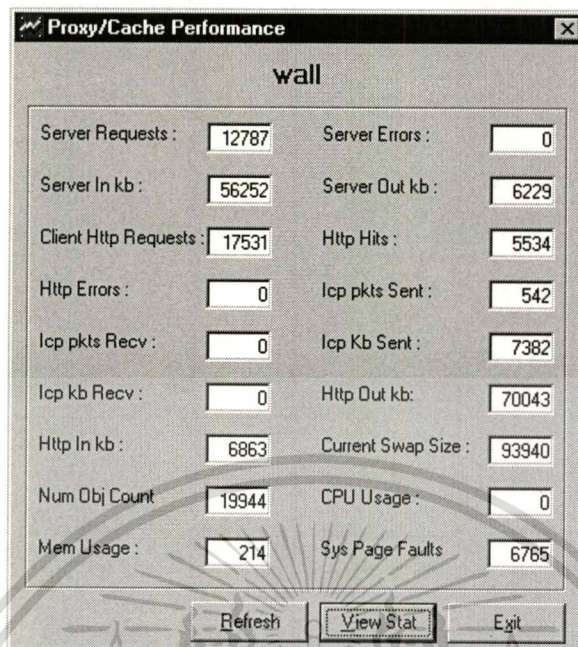
การตรวจสอบประสิทธิภาพการทำงานของ Proxy/Cache เราสามารถใช้โปรแกรมนี้ควบคุมการทำงานของโปรแกรม Mrtg โดยจะต้องมีการกำหนดค่าต่างๆ จากเมนู "Mrtg Option" ดังภาพที่ 4.10 เสียก่อน ส่วนขั้นถัดไปก็คือการสร้าง Config ของ Mrtg ด้วยเมนูคำสั่ง "Make Mrtg Config" ซึ่งจะทำการสร้างเพิ่มข้อมูลที่ต้องการและนำไปเก็บไว้ในที่ที่เหมาะสม (ใน Mrtg Working Directory)



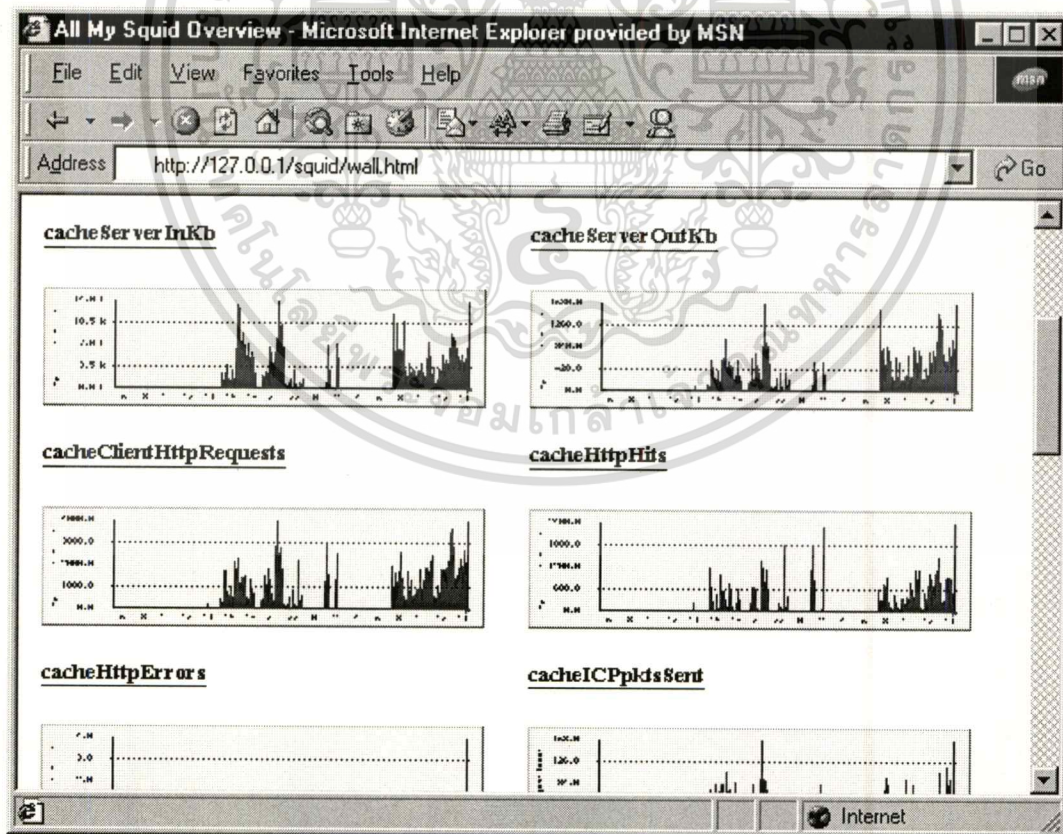
ภาพที่ 4.10 หน้าจอการกำหนดค่าต่างๆ ของโปรแกรม Mrtg

สามารถตรวจสอบประสิทธิภาพการทำงานปัจจุบันของ Proxy/Cache โดยใช้เมนูคำสั่ง "Show Performance" ซึ่งแสดงเป็น Dialog ในภาพที่ 4.11 และหากต้องการดูสถิติในรูปแบบกราฟที่สามารถเข้าใจได้ง่าย ก็สามารถใช้คำสั่ง "View Stat" ในการเรียก Browser ดูข้อมูลสถิติจากโปรแกรม Mrtg ซึ่งหน้าแรกจะเป็นสถิติโดยรวม(แสดงในภาพที่ 4.12) ซึ่งสามารถคลิกเลือกดูรายละเอียดของประสิทธิภาพแต่ละค่าได้ (แสดงในภาพที่ 4.13)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

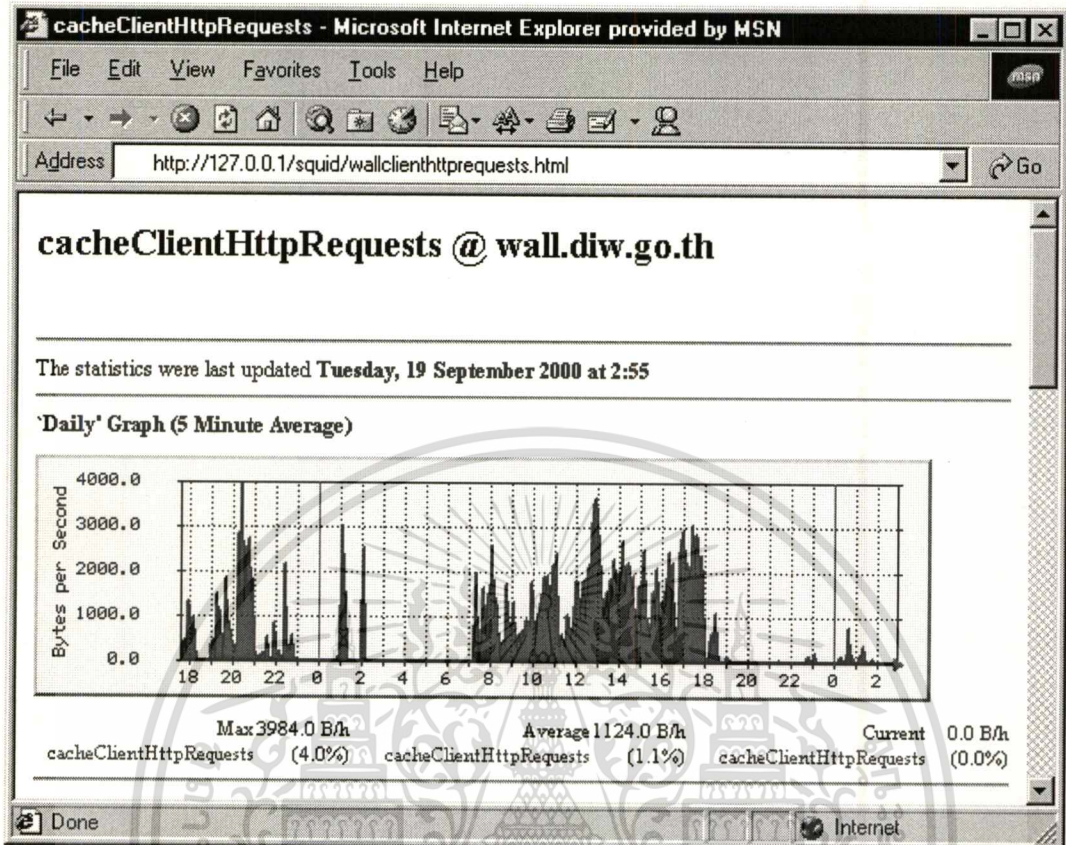


ภาพที่ 4.11 หน้าจอแสดงประสิทธิภาพการทำงานปัจจุบันของ Proxy/Cache



ภาพที่ 4.12 หน้าจอแรกของการใช้ Browser เรียกดูข้อมูลจากโปรแกรม Mrtg

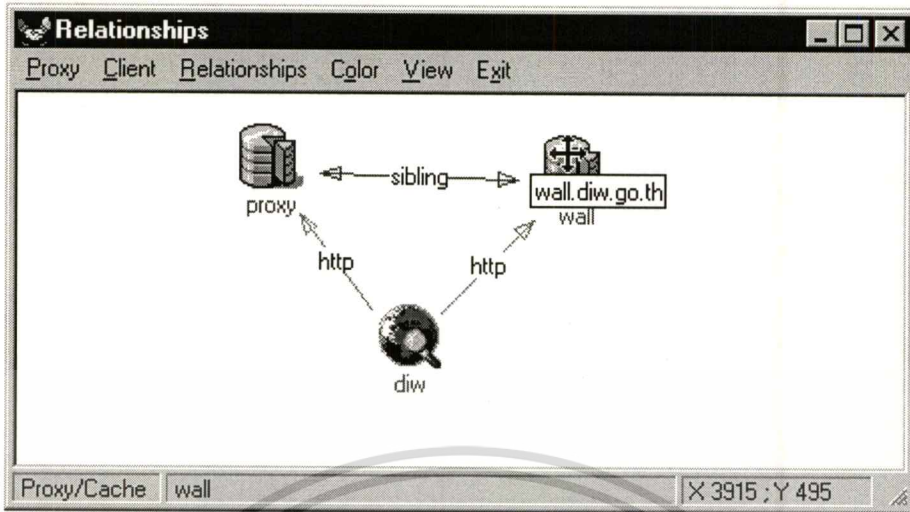
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.13 หน้าจอรายละเอียดของประสิทธิภาพในแต่ละค่า

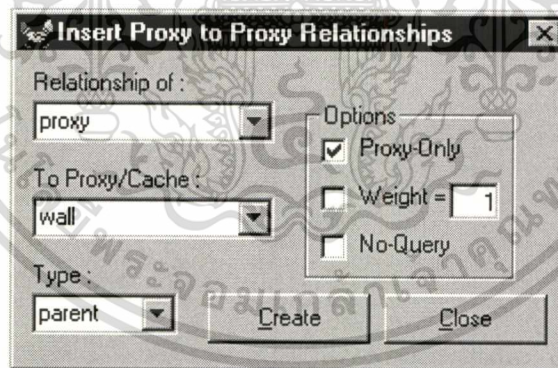
การสร้างหรือดูภาพแสดงความสัมพันธ์ของ Proxy/Cache สามารถทำได้โดยใช้เมนูคำสั่ง "Show Relationships" จะมีฟอร์มสำหรับการจัดการทั้งหมด เราสามารถใช้วิธี Drag & Drop ในการเคลื่อนย้ายวัตถุและการเพิ่มความสัมพันธ์ได้ วัตถุจะมีชื่อกำกับอยู่ด้านล่างและความสัมพันธ์จะแสดงด้วยลูกศรสีต่างๆ พร้อมฉลาก(Label) แสดงชนิดของความสัมพันธ์ด้วย ในภาพนี้และสามารถใช้ Mouse ชี้ที่วัตถุเพื่อดูรายละเอียดที่แถบแสดงสถานะด้านล่าง แสดงดังภาพที่ 4.14

การใช้งานเริ่มต้นจะต้องมีการเพิ่มวัตถุเข้ามาในภาพ ซึ่งจะมีวัตถุอยู่ 2 ชนิดด้วยกันก็คือ "Proxy" และ "Client" โดยที่ "Proxy" จะหมายถึงเครื่อง Proxy/Cache ที่อยู่ในระบบซึ่งให้บริการแก่ผู้ใช้งานในขอบเขตของ IP Address ที่กำหนด ในที่นี้จะแทนด้วย "Client" หลังจากนั้นเราจึงสามารถสร้างความสัมพันธ์ระหว่าง "Proxy to Proxy" หรือระหว่าง "Client to Proxy/Cache" ได้



ภาพที่ 4.14 หน้าจอแสดงภาพความสัมพันธ์ของ Proxy/Cache

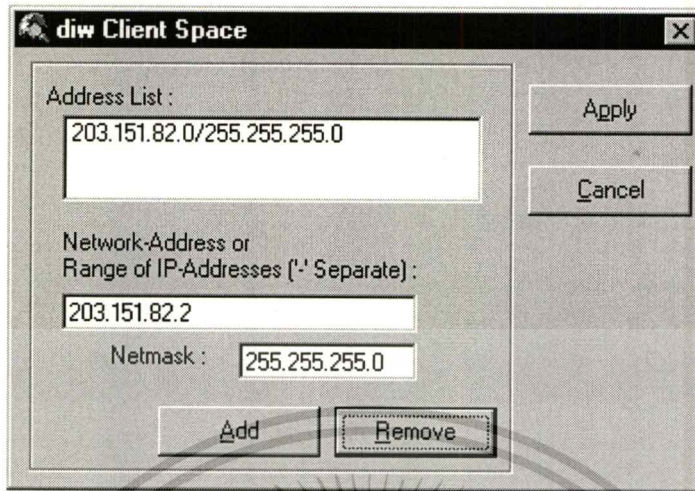
ความสัมพันธ์มีด้วยกัน 2 ชนิด ชนิดแรกคือความสัมพันธ์ระหว่าง Proxy to Proxy การสร้างความสัมพันธ์ชนิดนี้จะมี Dialog สำหรับป้อนข้อมูลดังภาพที่ 4.15 ความสัมพันธ์มีด้วยกัน 2 แบบคือ "parent" และ "sibling" สามารถเลือกได้ในช่อง Type และมี Options ให้เลือกโดยการกาเครื่องหมาย "√" ในช่องที่ต้องการ สำหรับรายละเอียดได้อธิบายไว้แล้วในทฤษฎีที่เกี่ยวข้อง



ภาพที่ 4.15 หน้าจอป้อนข้อมูลความสัมพันธ์ระหว่าง Proxy to Proxy

ความสัมพันธ์ชนิดที่สองเป็นความสัมพันธ์ระหว่าง Client to Proxy เป็นการยินยอมให้บริการของ Proxy/Cache ในขอบเขตของ IP Address กำหนด(ในที่นี้จะเรียกว่า Client Space) โดยสามารถสร้างขอบเขตขึ้นจากเมนู Client → Insert ในการกำหนดขอบเขตจะมี Dialog สำหรับป้อนข้อมูล IP Address และ Netmask ดังภาพที่ 4.16 หลังจากนั้นจึงสร้างความสัมพันธ์ด้วยวิธี Drag & Drop จาก Client ไปยัง Proxy ซึ่งจะมี 2 แบบด้วยกันคือ "http" และ "icp" ตามการใช้งานของผู้ใช้ ซึ่งโดยทั่วไปจะเป็นแบบ "http"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.16 หน้าจอป้อนข้อมูลขอบเขตของ IP Address

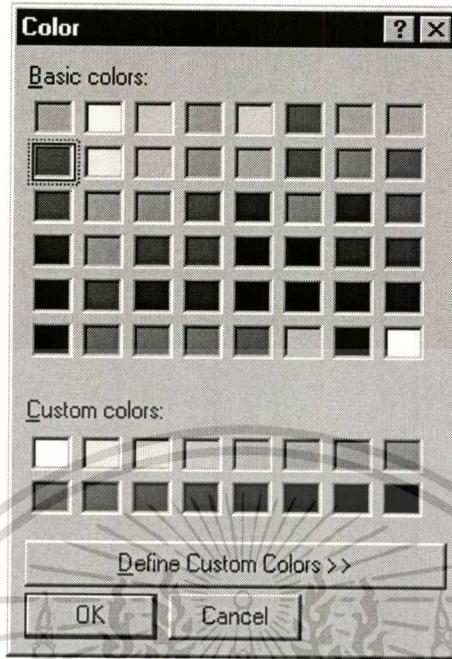
โปรแกรมนี้สามารถแสดงความสัมพันธ์ที่สร้างขึ้นทั้งหมดจากเมนูคำสั่ง Show All และหากผู้ใช้มีความต้องการยกเลิกความสัมพันธ์ที่สร้างขึ้น ก็สามารถเลือกความสัมพันธ์แล้วกดปุ่มคำสั่ง "Delete" ได้ แสดงดังภาพที่ 4.17 และในการยกเลิกความสัมพันธ์อาจทำได้โดยการใช้เมนู Pop-up จากการคลิกขวาที่ฉลากแสดงชนิดของความสัมพันธ์ได้เช่นกัน



ภาพที่ 4.17 หน้าจอแสดงความสัมพันธ์ที่สร้างขึ้นทั้งหมด

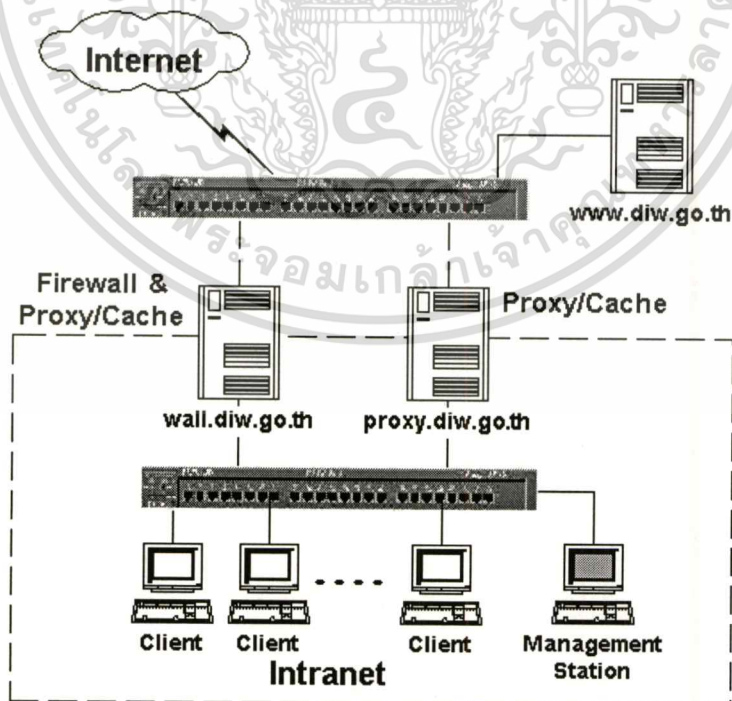
นอกจากนั้น ในภาพแสดงความสัมพันธ์ยังเปิดโอกาสให้ผู้ใช้สามารถเปลี่ยนแปลงสีต่างๆ ได้ตามความต้องการ เช่น สีพื้นหรือสีของฉลากต่างๆ เพื่อความเหมาะสมโดยใช้เมนูคำสั่ง "Color" โปรแกรมจะแสดง Dialog ให้ผู้ใช้เลือกสีที่ต้องการได้ ดังภาพที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.18 หน้าจอแสดงการเปลี่ยนแปลงสีของภาพความสั้มพันธ์

#### 4.5 การทดสอบใช้งานกับระบบจริง



ภาพที่ 4.19 ระบบ Proxy/Cache ของกรมโรงงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ทำการทดสอบการทำงานของทำงานของโปรแกรมนี้กับระบบ Proxy/Cache ของกรมโรงงานอุตสาหกรรม กระทรวงอุตสาหกรรม ซึ่งใช้ Squid Proxy/Cache จำนวน 2 เครื่องในการให้บริการกับเจ้าหน้าที่ภายในประมาณ 700 คน เครื่องหนึ่งทำหน้าที่เป็น Firewall ที่คอยปกป้องและให้บริการเครือข่ายภายในด้วย ส่วนอีกเครื่องทำหน้าที่เป็น Proxy/Cache เพียงอย่างเดียวเท่านั้นและให้บริการกับหน่วยงานภายในกระทรวงอีกด้วย แสดงดังภาพที่ 4.19

ในการทดสอบได้ใช้โปรแกรมบริหารจัดการ Proxy/Cache สร้าง Project ขึ้นมาใหม่ แล้วทำการเพิ่ม Proxy/Cache ด้วย Configuration จากค่า Default ของโปรแกรมทั้งสองเครื่อง กำหนดชื่อเครื่องให้ชื่อเป็น “wall” และ “proxy” ตามลำดับ เสร็จแล้วทำการเปลี่ยนแปลงแก้ไข ค่า http\_port ของเครื่อง wall ให้มีค่าเป็น “8080” และได้กำหนดค่าใน Mrtg Options ให้มีการทำงานโปรแกรม Mrtg ทุกๆ 5 นาทีด้วย

ในภาพความสัมพันธ์ ได้สร้างความสัมพันธ์ ระหว่าง Proxy to Proxy จาก “proxy” ไป “wall” ในแบบ Sibling และ จาก “wall” ไป “proxy” ในแบบ Sibling เช่นกัน แต่มี Option “proxy-only” เสร็จแล้วจึงสร้าง Client โดยใช้ชื่อ “diw” ซึ่งจะมีของเขตของ IP Address เป็น 203.151.82.0/255.255.255.0 แล้วกำหนดความสัมพันธ์ระหว่าง Client to Proxy ไปยังเครื่องทั้งสองในแบบ http ด้วย เมื่อเสร็จเรียบร้อยแล้วจะได้ภาพเช่นเดียวกับ ภาพที่ 4.14

หลังจากนั้นจึงทำการตรวจสอบความถูกต้องของข้อมูลใน Configuration File ที่ Management Station ปรากฏว่ามีความถูกต้องตรงตามต้องการ ทำการ Upload ข้อมูลกลับไปยัง Proxy/Cache ทั้งสอง แล้วทำการสั่งให้ Proxy/Cache หยุดทำงานแล้วเริ่มใหม่ เมื่อเสร็จสิ้นขบวนการ ปรากฏว่า Proxy/Cache ทั้งสองเครื่องสามารถให้บริการได้ตามปกติ

จากนั้นสักครู่ ได้ทำการตรวจสอบประสิทธิภาพของโปรแกรมพบว่าการส่งข้อมูล ICP ระหว่างเครื่องจำนวนหนึ่ง แต่ก็ยังไม่สามารถวิเคราะห์ได้ว่าการทำงานดีขึ้นหรือไม่ เนื่องจากมีเวลาในการทดสอบน้อย และการเปลี่ยนแปลงของระบบ Proxy/Cache จะต้องใช้ระยะเวลาช่วงหนึ่ง

## บทที่ 5

### สรุปผลการพัฒนาระบบงาน

#### 5.1 ผลจากการพัฒนาระบบ

การพัฒนา Software ช่วยในการบริหารจัดการระบบ Proxy/Cache นี้สามารถสรุปได้ดังนี้

##### 5.3.1 การศึกษาการทำงานของ Proxy/Cache โปรแกรม Squid

ผลการศึกษาพบว่า Proxy/Cache มีประโยชน์อย่างยิ่งในการช่วยลดปริมาณการติดต่อไปยังเครือข่ายภายนอก ในกรณีที่มีการสื่อสารข้อมูลซ้ำๆ กัน อีกทั้งยังสามารถทำงานร่วมกันระหว่าง Proxy/Cache เพื่อเพิ่มประสิทธิภาพการทำงานได้ แต่ยังมีปัญหาในการบริหารจัดการอยู่ เช่น การ Configuration และการตรวจสอบประสิทธิภาพ

##### 5.3.1 การวิเคราะห์และออกแบบ

ได้ทำการออกแบบระบบให้มีการทำงานจากส่วนกลาง โดยจะมีโปรแกรม Agent ทำงานอยู่ที่ Proxy/Cache ในการส่งต่อคำสั่งต่างๆ จาก Management Station ส่วนการตรวจสอบประสิทธิภาพจะใช้โปรแกรม Mrtg โดยผ่าน SNMP โพรโทคอล และจะมีโปรแกรม Squid Admin เป็นส่วนติดต่อผู้ใช้งานในการบริหารจัดการระบบทั้งหมดที่ส่วนกลาง

##### 5.3.1 การพัฒนาระบบงาน

ได้ทำการพัฒนาโปรแกรม Agent ที่สามารถส่งต่อการควบคุม Proxy/Cache จากส่วนกลาง และพัฒนาโปรแกรม Squid Admin สามารถนำมาใช้บริหารจัดการระบบ Squid Proxy/Cache ได้โดยง่าย เช่น การสั่งงานโปรแกรม การแก้ไขปรับปรุง Configuration และการตรวจสอบประสิทธิภาพ และโปรแกรมสามารถทำงานได้กับระบบจริง

#### 5.2 ประโยชน์ที่ได้รับ

หลังจากพัฒนาโปรแกรมนี้อแล้ว ทำให้เราสามารถบริหารจัดการระบบ Proxy/Cache ได้ง่ายขึ้น โดยสามารถทำการเปลี่ยนแปลงแก้ไข Configuration ได้อย่างรวดเร็วโดยที่ไม่จำเป็นต้องมีความรู้ความชำนาญเกี่ยวกับโปรแกรมมากนัก สามารถตรวจสอบและเก็บสถิติของประสิทธิภาพการทำงานของ Proxy/Cache แต่ละเครื่องแล้วนำมาเปรียบเทียบกันได้ โปรแกรมนี้จะเป็นประโยชน์อย่างมากกับหน่วยงานที่ใช้โปรแกรม Squid Proxy/Cache โดยเฉพาะกรณีที่มีเครื่อง Proxy/Cache จำนวนมากหรือ Network มีความซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 อุปสรรคในการพัฒนาระบบ

- 5.3.1 การทำงานของโปรแกรมนี้ออกแบบให้ สามารถทำงานกับเครื่อง Proxy/Cache ได้หลายเครื่อง และใช้กับระบบปฏิบัติการ Unix ได้หลายชนิด แต่ในการพัฒนามีเครื่องสำหรับทดสอบเพียง 2 เครื่องเท่านั้น และยังเป็นระบบปฏิบัติการ Linux ทั้งสองเครื่อง
- 5.3.2 เนื่องจากเครื่อง Proxy/Cache ที่นำมาใช้พัฒนา มีการให้บริการกับผู้ใช้จำนวนมาก ในบางช่วงเวลาจึงไม่สามารถทำการทดสอบการทำงานของโปรแกรมได้
- 5.3.3 เนื่องจากผู้พัฒนา ไม่มีความคุ้นเคยกับการเขียนโปรแกรมด้วย Microsoft Visual Basic มาก่อน จึงค่อนข้างจะใช้เวลามากในการพัฒนา

### 5.4 ข้อเสนอแนะ

- 5.4.1 เนื่องจากโปรแกรมในส่วน Management Station นี้มีการทำงานแก้ไขเพิ่มข้อมูลที่มีขนาดใหญ่จำนวนมาก จึงควรที่จะใช้เครื่องที่มี Disk I/O ความเร็วสูง
- 5.4.2 ก่อนทำการแก้ไขข้อมูล Configuration ควรมีการทำข้อมูล Backup ข้อมูลเก็บไว้ชุดหนึ่งก่อน เพราะ Configuration เดิมอาจมีการทำงานที่ดีกว่า Configuration ที่แก้ไขใหม่ก็ได้ วิธีการอาจทำได้โดยการใช้เมนูคำสั่ง "Save Project As" แล้วกำหนด Folder ที่เก็บข้อมูลเป็น Folder ใหม่ (การกำหนดให้เป็น Folder เดิม ข้อมูลเก่าอาจ ถูกลบทิ้ง)
- 5.4.3 Proxy/Cache มีผู้ใช้บริการจำนวนมาก การแก้ไขใดๆ ที่ Proxy/Cache ควรกระทำในช่วงที่มีการใช้งานน้อย
- 5.4.4 ควรกำหนดให้มี Management Station เพียงเครื่องเดียว และกำหนดให้ Agent ติดต่อกับเครื่องนี้เท่านั้น เพื่อป้องกันการบุกรุกจากผู้ประสงค์ร้าย (Hacker)

## บรรณานุกรม

กิตติ ภัคดีวัฒนกุล และ จำลอง ทรูอุตสาหะ . 2543 . Visual Basic 6 ฉบับโปรแกรมเมอร์ . พิมพ์ครั้งที่ 7 . กรุงเทพฯ : ไทยเจริญการพิมพ์

สันติ ศรีลาศักดิ์ และ วรุฒิ เทียงธรรม . 2542 . เจาะประเด็นงานเขียนโปรแกรมบนนี้ . กรุงเทพฯ : ซีเอ็ดยูเคชั่น

Andrew Cormack University of Wales, Cardiff. 2000 . “Web Caching” . [Online] . Available : <http://www.jisc.ac.uk/acn/caching.html>

Duane Wessels . 2000 . SQUID Frequently Asked Questions . [Online] . Available : <http://www.squid-cache.org/Doc/FAQ/FAQ.html>

Tobias Oetiker . 2000 . Multi Router Traffic Grapher . [Online] . Available : <http://www.mrtg.org>

## ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก.

## ตัวอย่าง Configuration File ของโปรแกรม MRTG

```

WorkDir: D:\inetpub\wwwroot\squid
Interval: 5

Options[^]: growright,
Colours[^]: VIOLET#aa00aa,DARK
VIOLET#660066,DARK RED#990033,DARK#3333ff

Target[wallServerRequests]:
cacheServerRequests&cacheServerRequests:public@wall.diw.go.th:3401
MaxBytes[wallServerRequests]: 100000
Title[wallServerRequests]: cacheServerRequests
Options[wallServerRequests]: gauge
PageTop[wallServerRequests]:
<h1>cacheServerRequests @ wall</h1>
YLegend[wallServerRequests]:
LegendI[wallServerRequests]: cacheServerRequests
LegendO[wallServerRequests]:
Legend1[wallServerRequests]: cacheServerRequests
Legend2[wallServerRequests]:

Target[wallServerErrors]:
cacheServerErrors&cacheServerErrors:public@wall.diw.go.th:3401
MaxBytes[wallServerErrors]: 100000
Title[wallServerErrors]: cacheServerErrors
Options[wallServerErrors]: gauge
PageTop[wallServerErrors]: <h1>cacheServerErrors @ wall</h1>
YLegend[wallServerErrors]:
LegendI[wallServerErrors]: cacheServerErrors
LegendO[wallServerErrors]:
Legend1[wallServerErrors]: cacheServerErrors
Legend2[wallServerErrors]:

Target[wallServerInKb]:
cacheServerInKb&cacheServerInKb:public@wall.diw.go.th:3401
MaxBytes[wallServerInKb]: 100000000
Title[wallServerInKb]: cacheServerInKb
Options[wallServerInKb]: perhour
PageTop[wallServerInKb]: <h1>cacheServerInKb @ wall</h1>
YLegend[wallServerInKb]:
LegendI[wallServerInKb]: cacheServerInKb
LegendO[wallServerInKb]:
Legend1[wallServerInKb]: cacheServerInKb
Legend2[wallServerInKb]:

Target[wallServerOutKb]:
cacheServerOutKb&cacheServerOutKb:public@wall.diw.go.th:3401
MaxBytes[wallServerOutKb]: 100000000
Title[wallServerOutKb]: cacheServerOutKb
Options[wallServerOutKb]: perhour

PageTop[wallServerOutKb]: <h1>cacheServerOutKb @ wall</h1>
YLegend[wallServerOutKb]:
LegendI[wallServerOutKb]: cacheServerOutKb
LegendO[wallServerOutKb]:
Legend1[wallServerOutKb]: cacheServerOutKb
Legend2[wallServerOutKb]:

Target[wallClientHttpRequests]:
cacheClientHttpRequests&cacheClientHttpRequests:public@wall.diw.go.th:3401
MaxBytes[wallClientHttpRequests]: 100000
Title[wallClientHttpRequests]:
cacheClientHttpRequests
Options[wallClientHttpRequests]: perhour
PageTop[wallClientHttpRequests]: <h1>cacheClientHttpRequests @ wall</h1>
YLegend[wallClientHttpRequests]:
LegendI[wallClientHttpRequests]:
cacheClientHttpRequests
LegendO[wallClientHttpRequests]:
Legend1[wallClientHttpRequests]:
cacheClientHttpRequests
Legend2[wallClientHttpRequests]:

Target[wallHttpHits]:
cacheHttpHits&cacheHttpHits:public@wall.diw.go.th:3401
MaxBytes[wallHttpHits]: 100000
Title[wallHttpHits]: cacheHttpHits
Options[wallHttpHits]: perhour
PageTop[wallHttpHits]: <h1>cacheHttpHits @ wall</h1>
YLegend[wallHttpHits]:
LegendI[wallHttpHits]: cacheHttpHits
LegendO[wallHttpHits]:
Legend1[wallHttpHits]: cacheHttpHits
Legend2[wallHttpHits]:

Target[wallHttpErrors]:
cacheHttpErrors&cacheHttpErrors:public@wall.diw.go.th:3401
MaxBytes[wallHttpErrors]: 100000
Title[wallHttpErrors]: cacheHttpErrors
Options[wallHttpErrors]: perhour
PageTop[wallHttpErrors]: <h1>cacheHttpErrors @ wall</h1>
YLegend[wallHttpErrors]:
LegendI[wallHttpErrors]: cacheHttpErrors
LegendO[wallHttpErrors]:
Legend1[wallHttpErrors]: cacheHttpErrors
Legend2[wallHttpErrors]:

Target[wallICPpktsSent]:
cacheICPpktsSent&cacheICPpktsSent:public@wall.diw.go.th:3401

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MaxBytes[wallICPpktsSent]: 100000  
 Title[wallICPpktsSent]: cacheICPpktsSent  
 Options[wallICPpktsSent]: perhour  
 PageTop[wallICPpktsSent]: <h1>cacheICPpktsSent @ wall</h1>  
 YLegend[wallICPpktsSent]: pkts per hour  
 ShortLegend[wallICPpktsSent]: p/h  
 LegendI[wallICPpktsSent]: ICPpktsSent  
 LegendO[wallICPpktsSent]:  
 Legend1[wallICPpktsSent]: cacheICPpkts Sent in Packets per Hour  
 Legend2[wallICPpktsSent]:

Target[wallICPpktsRecv]:  
 cacheICPpktsRecv&cacheICPpktsRecv:public@wall.diw.go.th:3401  
 MaxBytes[wallICPpktsRecv]: 100000  
 Title[wallICPpktsRecv]: cacheICPpktsRecv  
 Options[wallICPpktsRecv]: perhour  
 PageTop[wallICPpktsRecv]: <h1>cacheICPpktsRecv @ wall</h1>  
 YLegend[wallICPpktsRecv]: pkts per hour

ShortLegend[wallICPpktsRecv]: p/h  
 LegendI[wallICPpktsRecv]: ICPpktsRecv  
 LegendO[wallICPpktsRecv]:  
 Legend1[wallICPpktsRecv]: cacheICPpkts Receive in Packets per Hour  
 Legend2[wallICPpktsRecv]:

Target[wallICPkbSent]:  
 cacheICPkbSent&cacheICPkbSent:public@wall.diw.go.th:3401  
 MaxBytes[wallICPkbSent]: 100000  
 Title[wallICPkbSent]: cacheICPkbSent  
 Options[wallICPkbSent]: perhour  
 PageTop[wallICPkbSent]: <h1>cacheICPkbSent @ wall</h1>  
 YLegend[wallICPkbSent]: Bytes per hour  
 ShortLegend[wallICPkbSent]: B/h  
 kMG[wallICPkbSent]: k,M,G,T,P,  
 LegendI[wallICPkbSent]: ICPkbSent  
 LegendO[wallICPkbSent]:  
 Legend1[wallICPkbSent]: cacheICP Sent in Bytes per Hour  
 Legend2[wallICPkbSent]:

Target[wallICPkbRecv]:  
 cacheICPkbRecv&cacheICPkbRecv:public@wall.diw.go.th:3401  
 MaxBytes[wallICPkbRecv]: 100000  
 Title[wallICPkbRecv]: cacheICPkbRecv  
 Options[wallICPkbRecv]: perhour  
 PageTop[wallICPkbRecv]: <h1>cacheICPkbRecv @ wall</h1>  
 YLegend[wallICPkbRecv]: Bytes per hour  
 ShortLegend[wallICPkbRecv]: B/h  
 kMG[wallICPkbRecv]: k,M,G,T,P,  
 LegendI[wallICPkbRecv]: ICPkbRecv  
 LegendO[wallICPkbRecv]:  
 Legend1[wallICPkbRecv]: cacheICP Receive in Bytes per Hour  
 Legend2[wallICPkbRecv]:

Target[wallHttpOutKb]:  
 cacheHttpOutKb&cacheHttpOutKb:public@wall.diw.go.th:3401  
 MaxBytes[wallHttpOutKb]: 100000  
 Title[wallHttpOutKb]: cacheHttpOut  
 Options[wallHttpOutKb]: perhour  
 PageTop[wallHttpOutKb]: <h1>cacheHttpOut @ wall</h1>  
 YLegend[wallHttpOutKb]: Bytes per hour  
 ShortLegend[wallHttpOutKb]: B/h  
 kMG[wallHttpOutKb]: k,M,G,T,P,  
 LegendI[wallHttpOutKb]: HttpOut  
 LegendO[wallHttpOutKb]:  
 Legend1[wallHttpOutKb]: cacheHttp Out in Bytes per Hour  
 Legend2[wallHttpOutKb]:

Target[wallHttpInKb]:  
 cacheHttpInKb&cacheHttpInKb:public@wall.diw.go.th:3401  
 MaxBytes[wallHttpInKb]: 100000  
 Title[wallHttpInKb]: cacheHttpIn  
 Options[wallHttpInKb]: perhour  
 PageTop[wallHttpInKb]: <h1>cacheHttpIn @ wall</h1>  
 YLegend[wallHttpInKb]: Bytes per hour  
 ShortLegend[wallHttpInKb]: B/h  
 kMG[wallHttpInKb]: k,M,G,T,P,  
 LegendI[wallHttpInKb]: HttpIn  
 LegendO[wallHttpInKb]:  
 Legend1[wallHttpInKb]: cacheHttp In in Bytes per Hour  
 Legend2[wallHttpInKb]:

Target[wallCurrentSwapSize]:  
 cacheCurrentSwapSize&cacheCurrentSwapSize:public@wall.diw.go.th:3401  
 MaxBytes[wallCurrentSwapSize]: 100000  
 Title[wallCurrentSwapSize]: cacheCurrentSwapSize  
 Options[wallCurrentSwapSize]: perhour  
 PageTop[wallCurrentSwapSize]: <h1>cacheCurrentSwapSize @ wall</h1>  
 YLegend[wallCurrentSwapSize]:  
 LegendI[wallCurrentSwapSize]:  
 cacheCurrentSwapSize  
 LegendO[wallCurrentSwapSize]:  
 Legend1[wallCurrentSwapSize]:  
 cacheCurrentSwapSize  
 Legend2[wallCurrentSwapSize]:

Target[wallNumObjCount]:  
 cacheNumObjCount&cacheNumObjCount:public@wall.diw.go.th:3401  
 MaxBytes[wallNumObjCount]: 100000  
 Title[wallNumObjCount]: cacheNumObjCount  
 Options[wallNumObjCount]: perhour  
 PageTop[wallNumObjCount]: <h1>cacheNumObjCount @ wall</h1>  
 YLegend[wallNumObjCount]:  
 LegendI[wallNumObjCount]: cacheNumObjCount  
 LegendO[wallNumObjCount]:  
 Legend1[wallNumObjCount]: cacheNumObjCount

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Legend2[wallNumObjCount]:

Target[wallCpuUsage]:

cacheCpuUsage&cacheCpuUsage:public@wall.diw.go.th:3401

MaxBytes[wallCpuUsage]: 100000

Title[wallCpuUsage]: cacheCpuUsage

Options[wallCpuUsage]: gauge

PageTop[wallCpuUsage]: <h1>cacheCpuUsage @ wall</h1>

ShortLegend[wallCpuUsage]: s

YLegend[wallCpuUsage]: Seconds

LegendI[wallCpuUsage]: cacheCpuUsage

LegendO[wallCpuUsage]:

Legend1[wallCpuUsage]: cacheCpuUsage

Legend2[wallCpuUsage]:

Target[wallMemUsage]:

cacheMemUsage&cacheMemUsage:public@wall.diw.go.th:3401

MaxBytes[wallMemUsage]: 100000

Title[wallMemUsage]: cacheMemUsage

Options[wallMemUsage]: gauge

kMG[wallMemUsage]: k,M,G,T,P,

PageTop[wallMemUsage]: <h1>cacheMemUsage @ wall</h1>

ShortLegend[wallMemUsage]: B

YLegend[wallMemUsage]: Bytes

LegendI[wallMemUsage]: cacheMemUsage

LegendO[wallMemUsage]:

Legend1[wallMemUsage]: cacheMemUsage

Legend2[wallMemUsage]:

Target[wallSysPageFaults]:

cacheSysPageFaults&cacheSysPageFaults:public@wall.diw.go.th:3401

MaxBytes[wallSysPageFaults]: 100000

Title[wallSysPageFaults]: cacheSysPageFaults

Options[wallSysPageFaults]: perhour

PageTop[wallSysPageFaults]:

<h1>cacheSysPageFaults @ wall</h1>

ShortLegend[wallSysPageFaults]:

YLegend[wallSysPageFaults]:

LegendI[wallSysPageFaults]: cacheSysPageFaults

LegendO[wallSysPageFaults]:

Legend1[wallSysPageFaults]: cacheSysPageFaults

Legend2[wallSysPageFaults]:



## ภาคผนวก ข. ตัวอย่างเพิ่มเก็บข้อมูลระบบ

#This is configuration file for squidadmin.exe

```
picture_size 6615 3450
picture_BG_color &HFFFFFF
picture_RP_color &HFF
picture_RC_color &H408000
picture_RL_color &HFF0000
```

```
mrtg_work_dir D:\inetpub\wwwroot\squid
mrtg_prog D:\perl\mrtg\mrtg
mrtg_perf_url http://127.0.0.1/squid/
mrtg_community public
mrtg_run False
```

```
#cache proxy adress agent_port
cache_wall wall.diw.go.th 2023
cache_proxy proxy.diw.go.th 2023
```

```
#cache_position proxy X Y
cache_position wall 3705 255
cache_position proxy 1005 270
```

```
#cache_relation from_proxy to_proxy type http_port icp_port options
cache_relation wall proxy sibling 3128 3130
cache_relation proxy wall sibling 3128 3130 proxy-only
```

```
#client space_name src
client diw 203.151.82.0/255.255.255.0
```

```
#client_position space_name X Y
client_position diw 2610 1680
```

```
#client_relation space_name proxy type port
client_relation diw proxy http 3130
client_relation diw wall http 3130
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก.

## Source Code ของโปรแกรม Agent (agent.C)

```

#include "sock.h"
#include <fstream.h>
#include <string.h>
#include <sys/stat.h>

#define MAXLINE 4000

struct fline {char line[1024];};
fline squidline[MAXLINE];
int all_line;

char* readval(char *arg){
char *buf,*buf2,buf3[1024];
int i;
for(i=1;i<=all_line;i++){
buf=strstr(squidline[i].line,arg);
if(buf!=NULL&&buf==squidline[i].line){
buf2= buf+strlen(arg)+1;
return buf2;}
}
return "";}

void search(char *arg){
char *buf;
int i;
for(i=1;i<=all_line;i++){
buf=strstr(squidline[i].line,arg);
if(buf!=NULL&&buf==squidline[i].line){
cout << buf << "\n";}
}
}

int replace1(char *arg1,char *arg2){
char *buf,*buf2;
int i;
for(i=1;i<=all_line;i++){
buf=strstr(squidline[i].line,arg1);
if(buf!=NULL&&buf==squidline[i].line){
buf2=buf+strlen(arg1);
buf=strcat(arg2,buf2);
strcpy(squidline[i].line,buf);
return i;}
}
return 0;
}

int addline(char *arg1,char *arg2){
int i;
char *buf,*buf2;
for(i=all_line++;i>=1;i--){
buf=strstr(squidline[i].line,arg1);
if(buf!=NULL&&buf==squidline[i].line){
strcpy(squidline[i+1].line,arg2);
return i+1;
} else squidline[i+1].line = squidline[i].line;
}

int delline(char *arg1){
int i;
char *buf;
for(i=1;i<=all_line;i++){
buf=strstr(squidline[i].line,arg1);
if(buf!=NULL&&buf==squidline[i].line){
strcpy(squidline[i].line,arg2);
return 1;
}
}
return 0;
}

void loadfile(char *arg){
int i=0;
char buf[1024];
/* read file into array */
ifstream in(arg);
if(!in) {
cerr << "file not found\n";
}
while(in) {
in.getline(buf,1024);
strcpy(squidline[++i].line,buf);
}
in.close();
all_line=i-1;
}

void savefile(char *arg){
int i;
ofstream out(arg);
for(i=1;i<=all_line;i++){
out << squidline[i].line << "\n";
}
out.close();
}

main( int argc, char* argv[])
{
char msg[1024],*buf2,buf[1024],bufe[1024],opt1
[30];
char socknm[80],cmd[255],squid_path[255],
squid_conf[255],*admin_host;
int i=0, j, port, nsid, rc;

if(argc>1) loadfile(argv[1]); else loadfile
("agent.conf");
strcpy(squid_path,readval("squid_bin"));
strcpy(squid_conf,readval("squid_conf"));
admin_host=readval("admin_host");
buf2=readval("agent_port");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (strcmp(admin_host,"any")) admin_host = NULL;
port=atoi(buf2);
if (port<=0) port=2023;

loadfile(squid_conf);

/* create and Bind stream socket */
sock sp( port!=1 ? AF_INET : AF_UNIX,
SOCK_STREAM );
if (!sp.good()) return 1;
if (sp.bind(admin_host,port) < 0) return 2;

while (strcmp(cmd,"END_PRG")!=0) {

/* accept a connection request from a client socket */
if ((nsid = sp.accept(0, 0)) < 0) return 1;

/* read from a client socket */
if ((rc=sp.read(buf, sizeof(buf), 0, nsid)) < 0) return
5;
(void)sscanf(buf,"%s %s",cmd,opt1);

if (strcmp(cmd,"READ_ALL")==0) {
for(i=1;i<=all_line;i++){
strcpy(msg,squidline[i].line);
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
}
strcpy(msg,"END_FILE");
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout << cmd << " " << all_line <<" LINE" <<endl;
}

else if (strcmp(cmd,"WRITE_ALL")==0) {
FILE * fstrm;
fstrm = fopen(squid_conf,"w");
char buff[2048];
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buff, sizeof(buff), 0, nsid)) < 0) return
5;
while(strcmp(buff,"END_FILE")!=0) {
fputs(buff,fstrm);
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buff, sizeof(buff), 0, nsid)) < 0)
return 5;
}
fclose(fstrm);
loadfile(squid_conf);
sprintf(msg,"%d",all_line);
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<" " << all_line <<endl;
}

else if (strcmp(cmd,"FILE_STAT")==0) {
struct stat fs;
stat("squid.conf",&fs);
sprintf(msg,"%d",fs.st_mtime);
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<" " << msg <<endl;
}

else if (strcmp(cmd,"READ_PARA")==0) {
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buf, sizeof(buf), 0, nsid)) < 0) return
5;
strcpy(msg,readval(buf));
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<" " << buf <<endl;
}

else if (strcmp(cmd,"ADD_LINE")==0) {
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buf, sizeof(buf), 0, nsid)) < 0) return
5;
bufe=buf;
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buf, sizeof(buf), 0, nsid)) < 0) return
5;
sprintf(msg,"%d",addline(bufe,buf));
savefile(squid_conf);
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<" " << msg <<endl;
}

else if (strcmp(cmd,"REPL_LINE")==0) {
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buf, sizeof(buf), 0, nsid)) < 0) return
5;
bufe=buf;
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buf, sizeof(buf), 0, nsid)) < 0) return
5;
sprintf(msg,"%d",replace1(bufe,buf));
savefile(squid_conf);
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<" " << msg <<endl;
}

else if (strcmp(cmd,"DEL_LINE")==0) {
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buf, sizeof(buf), 0, nsid)) < 0) return
5;
sprintf(buf,"%d",delline(buf));
savefile(squid_conf);
strcpy(msg,buf);
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<" " << msg <<endl;
}

else if (strcmp(cmd,"REST_SQUID")==0) {
buf2=buf;
strcpy(buf2,squid_path);
strcat(buf2," -k reconfigure");
rc=system(buf2);
if(rc==0) strcpy(msg,"Finish");
else { sprintf(buf,"Error %d",rc); strcpy(msg,buf); }
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<" " << msg <<endl;
}

else if (strcmp(cmd,"SQUID_CMD")==0) {
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buf, sizeof(buf), 0, nsid)) < 0) return
5;
buf2=bufe;
strcpy(buf2,squid_path);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

strcat(buf2," -f ");
strcat(buf2,squid_conf);
strcat(buf2," ");
strcat(buf2,buf);
rc=system(buf2);
if(rc==0) strcpy(msg,"Finish");
else sprintf(msg,"Error %d",rc);
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<" "<< buf <<endl;
}

else if (strcmp(cmd,"SYSTEM_CMD")==0) {
if (sp.write("\0",1,0,nsid)<0) return 6;
if ((rc=sp.read(buf, sizeof(buf), 0, nsid) < 0) return
5;
rc=system(buf);
if(rc==0) strcpy(msg,"Finish");
else { sprintf(buf,"Error %d",rc); strcpy(msg,buf); }
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;

cout<< cmd <<" "<< msg <<endl;
}

else if (strcmp(cmd,"END_PRG")==0) {
strcpy(msg,"Finish");
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout<< cmd <<endl;
}

else {
strcpy(msg,"Error Unknown Command");
if (sp.write(msg,strlen(msg)+1,0,nsid)<0) return 6;
cout << msg <<endl;
}
sp.shutdown;
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ง. วิธีการติดตั้งโปรแกรม

ในระบบโปรแกรมช่วยในการบริหารจัดการระบบ Proxy/Cache จะประกอบด้วยโปรแกรม 2 ส่วนคือ ส่วนโปรแกรม Agent ทำงานที่ Proxy/Cache ซึ่งอยู่บนระบบปฏิบัติการ Unix และส่วน squidadmin ทำงานที่ส่วนกลาง(Management Station) ซึ่งอยู่บนระบบปฏิบัติการ Windows

ส่วนของ Agent สามารถทำงานได้กับเครื่อง Unix ทั่วไปที่สามารถใช้โปรแกรม Squid และส่วนของ squidadmin อาจใช้กับเครื่อง PC ที่มีคุณลักษณะและโปรแกรมดังนี้

1. CPU Pentium ความเร็วไม่ต่ำกว่า 100 MHz
2. HD ไม่ต่ำกว่า 1 GB
3. RAM ไม่น้อยกว่า 16 MB
4. Windows NT 4.0 หรือ Windows 98
5. Microsoft Internet Information Server หรือ Personal Web Server
6. Active Perl version 5.22 ขึ้นไป

โดยในส่วนการติดตั้งโปรแกรมสามารถแบ่งได้เป็น 2 ส่วน คือ

### ส่วนที่ 1 การติดตั้งโปรแกรม Agent ที่ Proxy/Cache

เมื่อได้เพิ่มข้อมูลโปรแกรมชื่อ Agent.zip แล้วให้ทำการแตกเพิ่มข้อมูลลงใน Directory ตามที่ต้องการ ซึ่งจะประกอบได้ด้วยเพิ่มข้อมูล 3 เพิ่มดังนี้ คือ “agent.C” , “agent.conf” และ “sock.h”

เมื่อได้เพิ่มข้อมูลครบแล้ว ให้ทำการ Compile โปรแกรมด้วย C++ Compiler โดยการใช้คำสั่ง “g++ agent.C -o agent” จากนั้นให้เข้าไปแก้ไขในส่วนของค่าตัวแปรที่เกี่ยวข้อง ใน File Agent.conf โดยใช้ Editor ของ Unix ดังนี้

1. ค่า Port ที่ต้องการให้โปรแกรมทำงาน
2. Path Name ของโปรแกรม squid
3. Path Name ของ Configuration file ของโปรแกรม squid
4. ชื่อของ Host ที่จะทำการบริหารจัดการ(ถ้ากำหนดเป็น “any” จะหมายถึงเครื่องใดก็ได้)

เมื่อเสร็จสิ้นแล้วโปรแกรม Agent ก็พร้อมทำงานด้วยการใช้คำสั่ง “agent s” โดยผู้ใช้ที่มีสิทธิในการแก้ไข Configuration File ของ Squid และสามารถสั่งงานโปรแกรม squid ได้ หรืออาจจะใส่ไว้ใน Start-Up Script ของ Unix เลขก็ได้

## ส่วนที่ 2 การติดตั้งโปรแกรม Squid Admin ที่ Management Station

เมื่อได้เพิ่มข้อมูลโปรแกรมชื่อ squidadmin.zip แล้วให้ทำการแตกเพิ่มข้อมูลลงใน Directory ตามที่ต้องการ ซึ่งจะประกอบได้ด้วยเพิ่มข้อมูล 5 แฟ้มดังนี้คือ “squidadmin.exe” , “prj.tpl” , “conf.tpl” , “mrtg.tpl” และ “html.tpl”

ทำการติดตั้งโปรแกรม Mrtg โดยนำแฟ้ม “mrtg-2.8.12.zip” มาแตกลงใน Directory ตามที่ต้องการ ตัวอย่างเช่น “C:\mrtg” แล้วทำการแก้ไขแฟ้ม “mrtg” ซึ่งเป็น Text File ใน Directory “run” ให้เป็นไปตามตารางที่ 4.2 ดังที่กล่าวไปแล้วในบทที่ 4

ทำการติดตั้ง Component ชื่อ “PowerTCP Snmp Tool” ตามขั้นตอนของโปรแกรม ซึ่งโปรแกรมนี้สามารถ Download ได้จาก Website “<http://www.dart.com>” โดยจะเป็นแฟ้มชื่อ “SnmpTool.exe”

เมื่อเสร็จสิ้นแล้วโปรแกรม squidadmin ก็พร้อมที่จะทำงานโดยการเรียกใช้ squidadmin.exe จาก Directory ที่เก็บโปรแกรม (จะต้องทำการ Run Agent ที่ Proxy/Cache ก่อน)

## ภาคผนวก จ.

### คู่มือการใช้งานโปรแกรม

Software ช่วยในการบริหารจัดการระบบ Proxy/Cache นี้ประกอบด้วยโปรแกรม 2 ส่วนที่ต้องทำงานร่วมกันคือ ส่วนของโปรแกรมที่ Proxy/Cache ซึ่งอยู่บนระบบปฏิบัติการ Unix และส่วนการบริหารที่ส่วนกลาง(Management Station) ซึ่งอยู่บนระบบปฏิบัติการ Windows ในการทำงานจะต้องทำการ Run โปรแกรม Agent ที่ Proxy/Cache ให้ครบเสียก่อน แล้วจึงมาใช้งานที่ Management Station

#### จ.1 เริ่มต้นใช้งานโปรแกรม

ในตอนเริ่มต้นจะต้องมีการสร้าง Project ขึ้นใหม่ โดยใช้เมนูคำสั่ง "New Project" จาก Pull-Down เมนู หรือจากแถบเครื่องมือ โดยจะต้องมีการระบุชื่อของโครงการ และตำแหน่งที่ใช้เก็บข้อมูล ดังภาพที่ จ.1

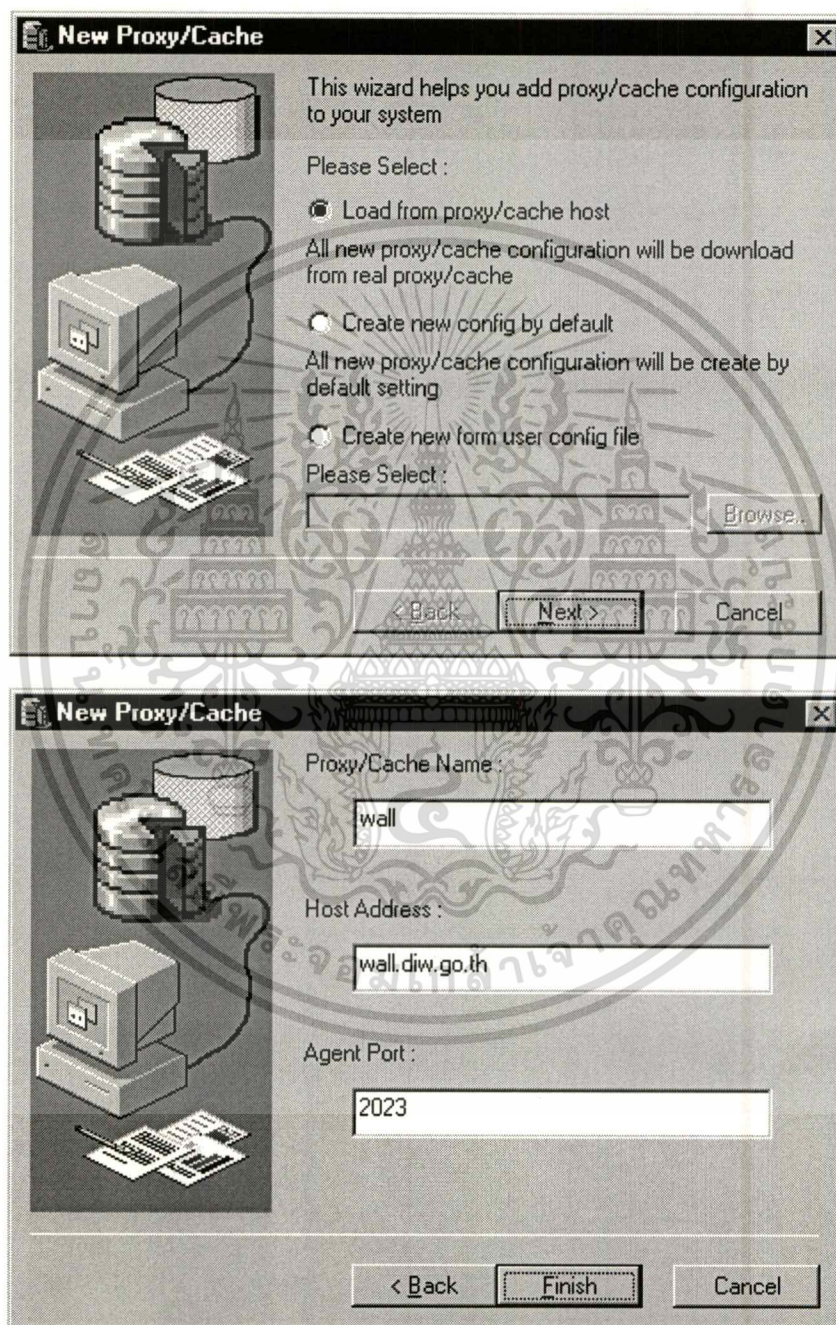


ภาพที่ จ.1 หน้าจอการสร้าง Project ใหม่

หลังจากนั้นให้ทำการเพิ่ม Proxy/Cache ในระบบทั้งหมดเข้ามาในโปรแกรม โดยการใช้เมนูคำสั่ง "New Proxy" ซึ่งผู้ใช้สามารถเลือกการนำข้อมูลได้จาก 3 แหล่งด้วยกันคือ

1. การ Download ข้อมูล Configuration จาก Proxy/Cache โดยตรง
2. การนำค่า Default ของโปรแกรมมาใช้
3. การนำเข้าจาก File ตามที่ผู้ใช้กำหนด

ผู้ใช้งานจะต้องมีการระบุชื่อ Proxy/Cache และ Host Address ซึ่งจะต้องไม่ซ้ำกัน พร้อมทั้งระบุ Agent Port ที่จะใช้ในการติดต่อสื่อสารด้วย การทำงานนี้จะเป็นแบบ "Wizard" หากต้องการยกเลิกสามารถกดที่ปุ่ม "Cancel" ได้ดังภาพที่ จ.2

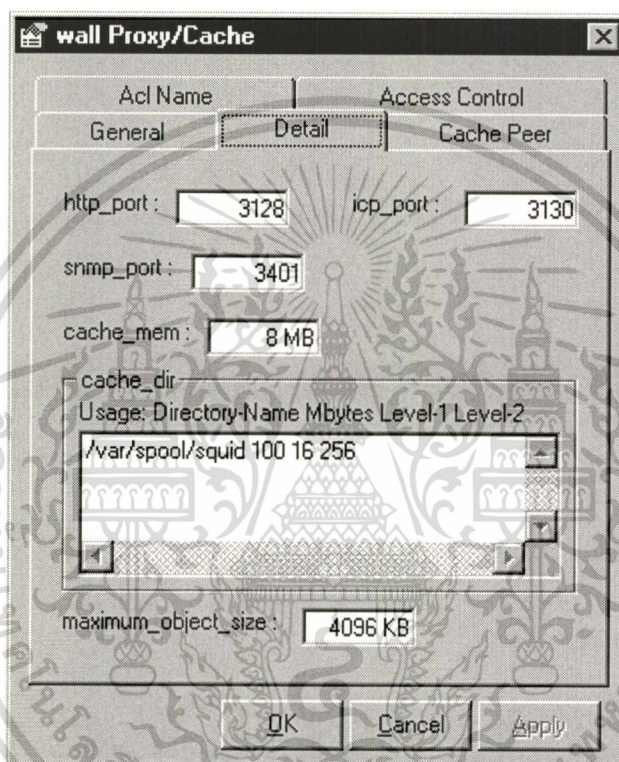


ภาพที่ จ.2 หน้าจอการเพิ่ม Proxy/Cache เข้ามาในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## จ.2 การแก้ไข Configuration ของ Proxy/Cache

เมื่อเพิ่ม Proxy/Cache ในระบบครบทั้งหมดแล้ว ให้ทำการเรียกดูหรือแก้ไขเปลี่ยนแปลง Configuration ของ Proxy/Cache แต่ละเครื่องให้เป็นไปตามต้องการ เช่น หมายเลข Port หรือ ขนาดของ Cache Directory โดยใช้เมนูคำสั่ง "Properties" โดยเฉพาะค่าใน Tab ข้อมูลรายละเอียด (Detail) ดังภาพที่ จ.3



ภาพที่ จ.3 หน้าจอการเรียกดูหรือทำการแก้ไขข้อมูลรายละเอียด

ส่วนการแก้ไขความสัมพันธ์ระหว่าง Proxy/Cache พร้อมกับ Access Control สามารถทำได้โดยอัตโนมัติ จากภาพความสัมพันธ์ ซึ่งจะกล่าวถึงในภายหลัง แต่ผู้ใช้ก็สามารถแก้ไขเพิ่มเติมจากในส่วนนี้ได้เช่นกัน

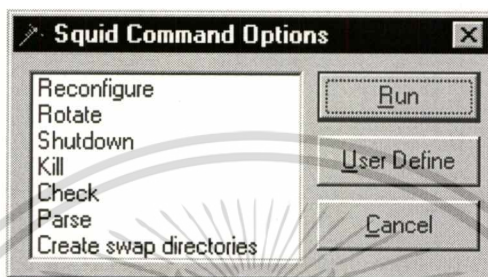
หากผู้ใช้ต้องการเปลี่ยนแปลงข้อมูล Configuration ใน Tag อื่นๆ ด้วยตนเอง สามารถทำได้ด้วยเมนูคำสั่ง "Custom Config" ซึ่งเมนูคำสั่งนี้จะทำการแก้ไขข้อมูล Configuration โดยผ่านโปรแกรม Notepad หรือ Wordpad ของ Windows

เมื่อทำการแก้ไขข้อมูล Configuration ของ Proxy/Cache เสร็จเรียบร้อยแล้ว สามารถนำ Configuration นั้นไปติดตั้งใช้งานด้วยการใช้เมนูคำสั่ง "Upload" และในทางตรงข้ามหากต้องการนำเข้าข้อมูลจาก Proxy/Cache เข้ามาใหม่ก็สามารถใช้คำสั่ง "Download" ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### จ.3 การสั่งงานโปรแกรม

การสั่งงานโปรแกรม ผู้ใช้จะต้องทำการเลือก Proxy/Cache ที่ต้องการก่อน ในโปรแกรม จะมีเมนูคำสั่ง "Start" และ "Stop" ไว้สำหรับการสั่งให้ Proxy/Cache เริ่มและหยุดการให้บริการตามลำดับ หรืออาจใช้เมนูคำสั่ง "Squid Command" ในการสั่งให้โปรแกรม Squid ทำงานตามต้องการ โดยจะมีตัวเลือกให้หรือผู้ใช้สามารถกำหนดเองก็ได้ ดังภาพที่ จ.4



ภาพที่ จ.4 หน้าจอตัวเลือกของคำสั่ง Squid Command

นอกจากสั่งงานโปรแกรม Squid แล้วผู้ใช้งานสามารถกำหนดให้ Proxy/Cache ทำงานคำสั่งใดๆ ก็ได้ตามต้องการด้วยคำสั่ง "System Command" แล้วป้อนคำสั่งเข้าไป ซึ่งผู้ใช้จะต้องมีความเข้าใจในการทำงานคำสั่งนั้นเป็นอย่างดี เพราะผู้ใช้จะทราบผลแต่เพียงว่าการทำงานนั้นเป็นสำเร็จหรือไม่

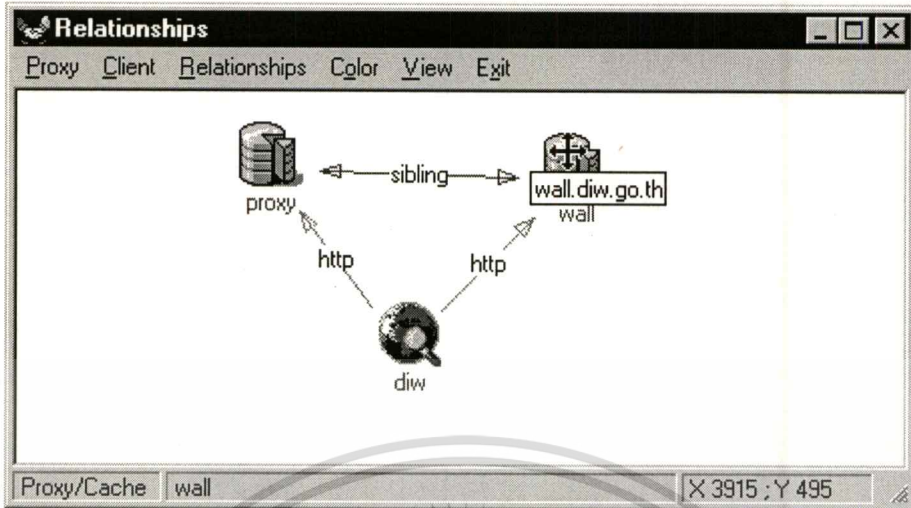
### จ.4 การจัดการภาพแสดงความสัมพันธ์ของ Proxy/Cache

การสร้างหรือดูภาพแสดงความสัมพันธ์ของ Proxy/Cache สามารถทำได้โดยใช้เมนูคำสั่ง "Show Relationships" ซึ่งจะมีฟอร์มสำหรับการจัดการทั้งหมด เราสามารถใช้วิธี Drag & Drop ในการเคลื่อนย้ายวัตถุและการเพิ่มความสัมพันธ์ได้ วัตถุจะมีชื่อกำกับอยู่ด้านล่างและความสัมพันธ์จะแสดงด้วยลูกศรสีต่างๆ พร้อมฉลาก(Label) แสดงชนิดของความสัมพันธ์ด้วย ในภาพนี้และสามารถใช้ Mouse ชี้ที่วัตถุเพื่อดูรายละเอียดที่แถบแสดงสถานะด้านล่าง แสดงดังภาพที่ จ.5

การใช้งานเริ่มต้นจะต้องมีการเพิ่มวัตถุเข้ามาในภาพ ซึ่งจะมีวัตถุอยู่ 2 ชนิดด้วยกันก็คือ "Proxy" และ "Client" โดยที่ "Proxy" จะหมายถึงเครื่อง Proxy/Cache ที่อยู่ในระบบซึ่งให้บริการแก่ผู้ใช้งานในขอบเขตของ IP Address ที่กำหนด ในที่นี้จะแทนด้วย "Client" หลังจากนั้นเราสามารถสร้างความสัมพันธ์ระหว่าง "Proxy to Proxy" หรือระหว่าง "Client to Proxy/Cache" ได้

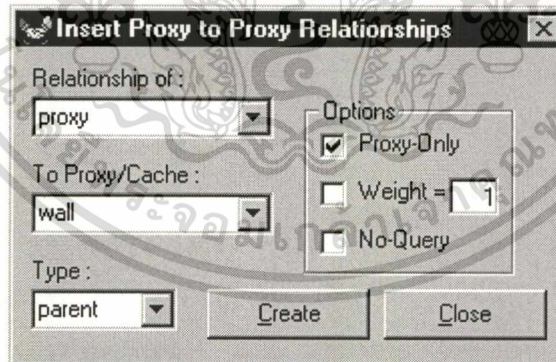
การเพิ่ม Proxy เข้ามาในภาพ ทำได้โดยใช้คำสั่ง Proxy → Insert ซึ่งจะมี Dialog ที่แสดงรายการของ Proxy/Cache ที่อยู่ในระบบทั้งหมดให้เลือก เมื่อเลือกแล้วให้ทำการลาก Proxy ไปวางในจุดที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ จ.5 หน้าจอแสดงภาพความสัมพันธ์ของ Proxy/Cache

ความสัมพันธ์จะมีด้วยกัน 2 ชนิด ชนิดแรกคือความสัมพันธ์ระหว่าง Proxy to Proxy การสร้างความสัมพันธ์ชนิดนี้ สามารถใช้วิธี Drag & Drop จาก Proxy หนึ่งไปยังอีก Proxy หนึ่ง ซึ่งจะมี Dialog สำหรับป้อนข้อมูลดังภาพที่ จ.6 ความสัมพันธ์มีด้วยกัน 2 แบบคือ "parent" และ "sibling" สามารถเลือกได้ในซึ่งช่อง Type และมี Options ให้เลือกโดยการกาเครื่องหมาย "✓" ในช่องที่ต้องการ สำหรับรายละเอียด ได้อธิบายไว้แล้วในทฤษฎีที่เกี่ยวข้อง

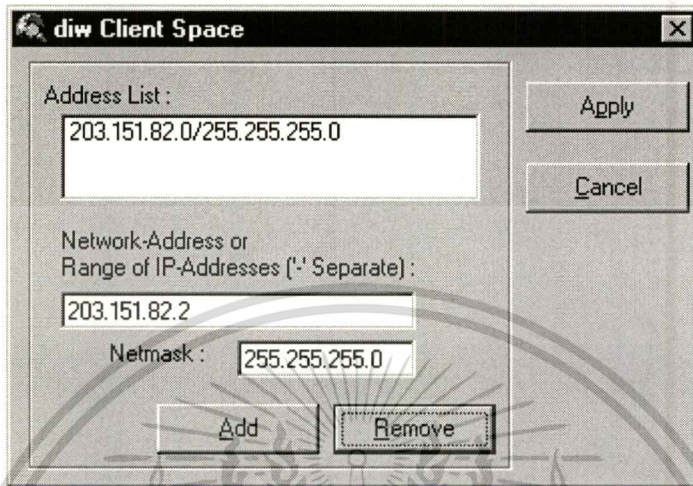


ภาพที่ จ.6 หน้าจอป้อนข้อมูลความสัมพันธ์ระหว่าง Proxy to Proxy

ความสัมพันธ์ชนิดที่สองเป็นความสัมพันธ์ระหว่าง Client to Proxy เป็นการยินยอมให้บริการของ Proxy/Cache ในขอบเขตของ IP Address กำหนด(ในที่นี้จะเรียกว่า Client Space) โดยก่อนอื่นจะต้องสร้างขอบเขตขึ้นจากเมนู Client → Insert ในการกำหนดขอบเขตจะมี Dialog สำหรับป้อนข้อมูล Network Address ซึ่งสามารถใส่ได้หลายๆ ค่า ดังภาพที่ จ.7 หลังจากนั้นจึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถสร้างความสัมพันธ์ด้วยวิธี Drag & Drop จาก Client ไปยัง Proxy ซึ่งจะมี 2 แบบด้วยกันคือ "http" และ "icp" ตามการใช้งานของผู้ใช้ ซึ่งโดยทั่วไปจะเป็นแบบ "http"



ภาพที่ จ.7 หน้าจอป้อนข้อมูลขอบเขตของ IP Address

ความสัมพันธ์ที่สร้างขึ้นทั้งหมดนี้ สามารถแสดงได้จากเมนูคำสั่ง Show All และหากผู้ใช้มีความต้องการยกเลิกความสัมพันธ์ที่สร้างขึ้น ก็สามารถเลือกความสัมพันธ์แล้วกดปุ่มคำสั่ง "Delete" ได้ดังภาพที่ จ.8 หรืออาจทำได้โดยการใช้เมนู Pop-up จากการคลิกขวาที่ผลากแสดงชนิดของความสัมพันธ์ได้เช่นกัน



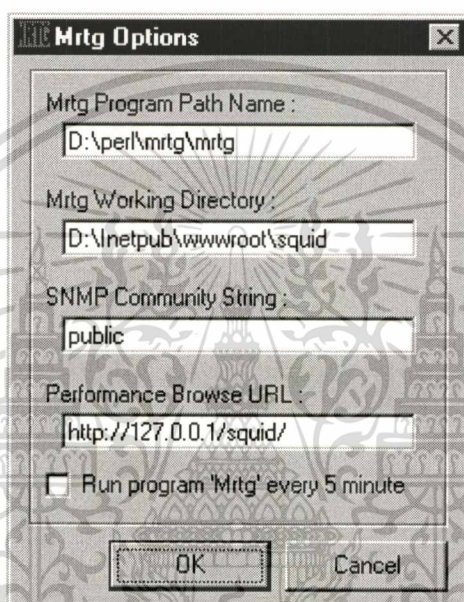
ภาพที่ จ.8 หน้าจอแสดงความสัมพันธ์ที่สร้างขึ้นทั้งหมด

นอกจากนั้น ในภาพแสดงความสัมพันธ์ยังเปิดโอกาสให้ผู้ใช้สามารถเปลี่ยนแปลงสีต่างๆ ได้ตามความต้องการ เช่น สีพื้นหรือสีของผลากต่างๆ เพื่อความเหมาะสมตามที่ผู้ใช้ต้องการ โดยใช้เมนูคำสั่ง "Color"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

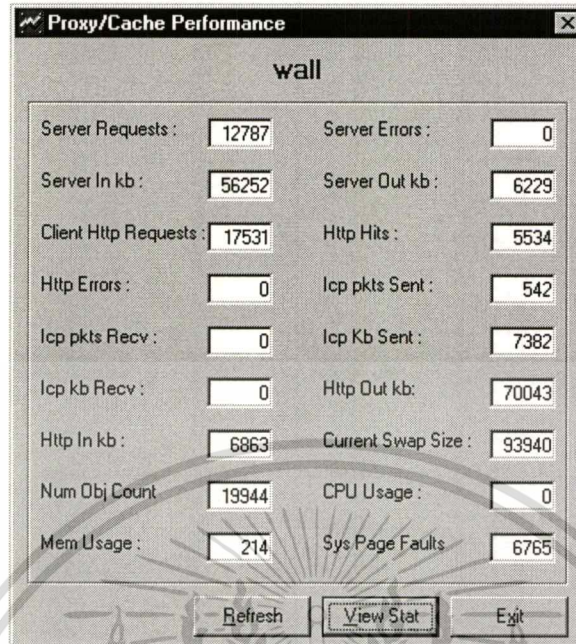
#### จ.4 การตรวจสอบประสิทธิภาพการทำงาน

การตรวจสอบประสิทธิภาพการทำงานของ Proxy/Cache สามารถทำได้โดยใช้โปรแกรมนี้ควบคุมการทำงานของโปรแกรม Mrtg โดยเริ่มต้นจะต้องมีการกำหนดค่าต่างๆ จากเมนู "Mrtg Option" ดังภาพที่ จ.9 เสียก่อน ส่วนขั้นถัดไปก็คือการสร้าง Config ของ Mrtg ของแต่ละ Proxy/Cache ด้วยเมนูคำสั่ง "Make Mrtg Config" ซึ่งจะทำการสร้างเพิ่มข้อมูลที่ต้องการและนำไปเก็บไว้ในที่ที่เหมาะสม (ใน Mrtg Working Directory)

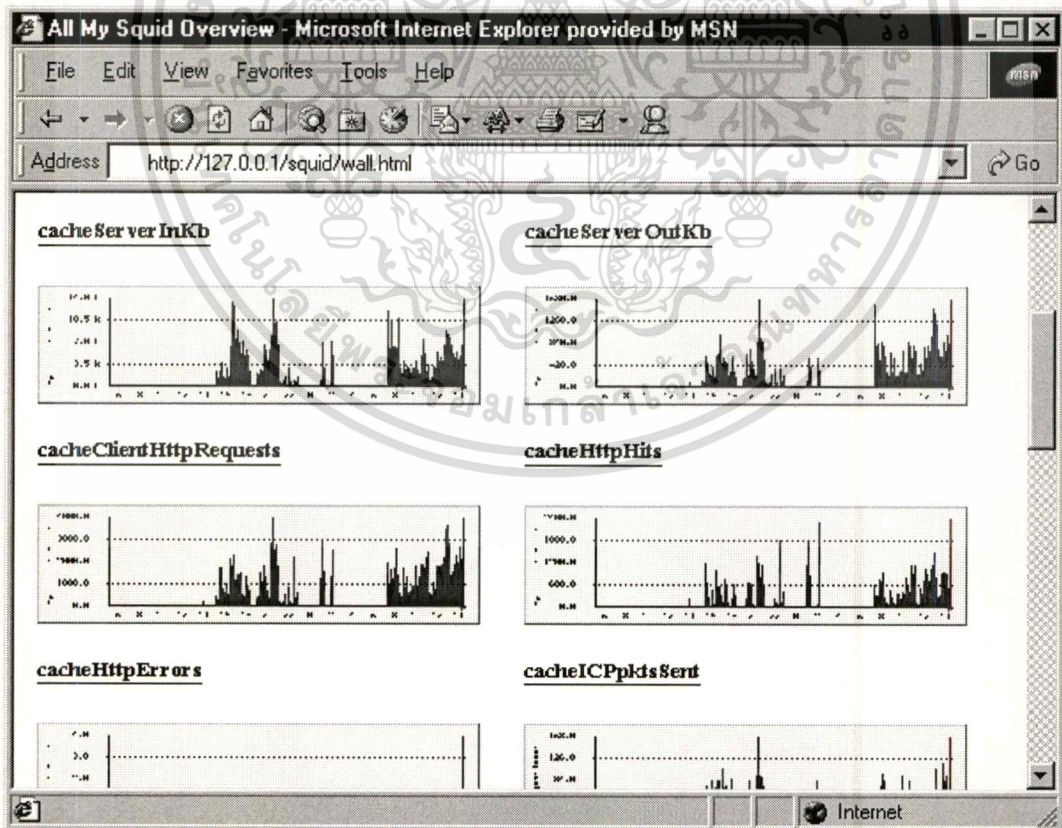


ภาพที่ จ.9 หน้าจอการกำหนดค่าต่างๆ ของโปรแกรม Mrtg

ในการตรวจสอบประสิทธิภาพการทำงานปัจจุบันของ Proxy/Cache ทำได้โดยใช้เมนูคำสั่ง "Show Performance" ซึ่งแสดงเป็น Dialog ในภาพที่ จ.10 และหากต้องการดูสถิติในรูปแบบกราฟที่สามารถเข้าใจได้ง่าย ก็สามารถใช้คำสั่ง "View Stat" ในการเรียก Browser ดูข้อมูลสถิติจากโปรแกรม Mrtg ซึ่งหน้าแรกจะเป็นสถิติโดยรวมซึ่งสามารถคลิกเลือกดูรายละเอียดของประสิทธิภาพในแต่ละค่าได้ในหน้าถัดไป (แสดงในภาพที่ จ.11)



ภาพที่ จ.10 หน้าจอแสดงประสิทธิภาพการทำงานปัจจุบันของ Proxy/Cache



ภาพที่ จ.11 หน้าจอแรกของการใช้ Browser เรียกดูข้อมูลจากโปรแกรม Mrtg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อผู้เขียน	นายสันติ สิทธิเลิศพิศาล
วันเกิด	5 กุมภาพันธ์ 2514
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วศ.บ.(เครื่องกล) คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ตำแหน่งหน้าที่	วิศวกรตรวจโรงงาน
สถานที่ทำงาน	ศูนย์สารสนเทศ กรมโรงงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้