

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง
ระบบแคชสำหรับโปรแกรมบิตทอเรนท
BITTORRENT CACHE SYSTEM



นายภูวรงค์ พรหมโมฆ
นางสาวมนธิชา มะระขงค์

เลขหมู่.....
เลขทะเบียน.....71977
วัน,เดือน,ปี..... 7 ส.ย. 2550

b...117612bx
i.....

ปริญญาโทเป็นส่วนหนึ่งของการศึกษาคามหัตถศูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแคชสำหรับโปรแกรมบิตทอเรนท

BITTORRENT CACHE SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาคามหัทธฤตรปริญญาวิตวกรรมศาสตรบัณฑิต

ธาวาวิตวกรรมคอมพิวเตอร

คณะวิตวกรรมศาสตร

ธวาทันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไวสำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
พ.ศ.2549

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2549

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแคชสำหรับโปรแกรมบิตทอเรนท

BITTORRENT CACHE SYSTEM

ผู้จัดทำ

1. นายภูวรุต พรหมโมบล รหัสนักศึกษา 45010598
2. นางสาวมนสิชา มะระขงค์ รหัสนักศึกษา 46010598



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแคชสำหรับโปรแกรมบิตทอเรนท

นายภูวรุศ พรหมโมบล	45010598
นางสาวมนสิชา มระระยงค์	46010598
อาจารย์ ธนัญชัย ศรีภาค	อาจารย์ที่ปรึกษา
อาจารย์ อัครเดช วัชรระภูพงษ์	อาจารย์ที่ปรึกษา
ปีการศึกษา 2549	

บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้นำเสนอวิธีสร้างระบบการทำงานที่สามารถลดปริมาณการใช้งานเครื่องข่ายสำหรับ โปรแกรมเชื่อมต่อแบบบิตทอเรนท (Cache System for BitTorrent) โดยการใช้การเปลี่ยนแปลงรูปแบบข้อมูลมาตรฐานในการดำเนินงานแลกเปลี่ยนข้อมูล (.torrent file metadata) การกำหนดค่าลงในฐานข้อมูลเพื่อให้สอดคล้องกัน (Database) การแก้ไขวิธีการดำเนินงานของโปรแกรมบิตทอเรนท (BitTorrent Tweak) และการทำงานของพร็อกซีเซิร์ฟเวอร์ (Proxy Server) โดยลักษณะเด่นของวิธีการที่นำเสนอในปริญญาานิพนธ์นี้คือ การลดปริมาณการเชื่อมต่อออกสู่ภายนอกเครื่องข่ายจาก โปรแกรมบิตทอเรนทได้โดยส่งผลกระทบต่อการทำงานของโปรแกรมบิตทอเรนทให้น้อยที่สุด ความสามารถที่จะแลกเปลี่ยนข้อมูลกันภายในได้เมื่อเกิดปัญหาเกี่ยวกับตัวกลางการแลกเปลี่ยนข้อมูล (Tracker) และการแลกเปลี่ยนข้อมูลที่เคยดึงมาแล้วสามารถทำได้ง่ายขึ้น ในปริญญาานิพนธ์ฉบับนี้จะใช้อัตราการดึงข้อมูลเฉลี่ยและอัตราสูงสุดมาเปรียบเทียบกรณีการดึงข้อมูลต่างๆกัน ระหว่างที่ใช้ระบบ กับการทำงานตามปกติ ซึ่งในปริญญาานิพนธ์เล่มนี้ได้แสดงผลของการทดสอบกรณีต่างๆ เพื่อเปรียบเทียบวิธีการที่นำเสนอกับวิธีการแบบเดิมเพื่อดูประสิทธิภาพและข้อได้เปรียบของวิธีการแต่ละแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BITTORRENT CACHE SYSTEM

Mr. Puwarut Prommobol	45010445
Mrs. Monsicha Marayong	45010493
Mr. Thanunchai Threepak	Advisor
Mr. Akkradach Watcharapupong	Advisor

Academic Year 2006

ABSTRACT

This thesis presents the cache system for Bittorrent Protocol that can reduce quantity of connection and bandwidth by integrating torrent metadata modification, Database insertion and modification, BitTorrent Client tweak, and Proxy Server concept. The outstanding features of the thesis is to reduce a connection and bandwidth with a few affect in usual process of Bittorrent Protocol while giving extra abilities to allow file sharing to work within network when tracker has problem and easy to access files which were downloaded before. This thesis employs the average download rate and maximum download rate to compare various situations between certain standard and proposed method. This thesis presents the various testing results of proposed method comparing with current using method to determine efficiency and advantage.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จได้อย่างดีด้วยคำแนะนำต่างๆจากอาจารย์ชั้นมัธยม ตรีภาค ถ้าไม่ได้อาจารย์ท่านคอยแนะนำต่างๆ คงไม่มีบัณฑิตที่เกษมศานต์ที่มีประสิทธิภาพเช่นนี้ รวมไปถึงอาจารย์ อัครเดช วัชรภูงษ์ ที่คอยช่วยให้คำแนะนำและกำลังใจในการทำงานเช่นเดียวกัน

ขอกราบพระคุณท่านคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ได้ให้คำแนะนำและกำลังใจอย่างดี

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยเฉพาะพี่ๆเพื่อนๆ น้องๆ ณ ห้องวิจัยไอแซคแห่งนี้ ทั้งที่จะจบด้วยกัน และยังมีอีกปี ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอพระคุณ บิดา มารดา และครอบครัวที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำปริญญาบัตรฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญาบัตรฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

นาย ภูวรุต พรหมโมบล

นางสาว มนสิชา มะระยงค์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.6 ส่วนประกอบของปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในโครงการ	4
2.1 บิตทอเรนท์	4
2.1.1 บิตทอเรนท์คืออะไร	4
2.1.2 การทำงานของบิตทอเรนท์	4
2.1.2.1 การดาวน์โหลด	6
2.1.2.2 การอัปโหลด	9
2.1.3 รายละเอียดของไฟล์ .torrent	11
2.1.4 การทำงานของแทรคเกอร์	13
2.1.5 การทำงานในรูปแบบ Peer-2-Peer ของเครื่องลูกข่าย	14
2.2 ฟร็อกซี่เซิร์ฟเวอร์	15
2.2.1 ฟร็อกซี่เซิร์ฟเวอร์คืออะไร	15
2.2.2 หลักการทำงานของฟร็อกซี่เซิร์ฟเวอร์	16
2.2.3 การจัดการแคชของฟร็อกซี่เซิร์ฟเวอร์	16
2.2.3.1 Passive Caching	16

เอกสารนี้เป็นเอกสารที่ลงนามแล้วสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.2.3.2 Active Caching	16
2.2.4 ประโยชน์ของการใช้พร็อกซีเซิร์ฟเวอร์	16
2.2.5 Transparent Proxy คืออะไร	17
2.3 โปรแกรม Squid	18
2.3.1 Access Log ของโปรแกรม Squid	18
2.3.2 Cache Dir แดชของโปรแกรม Squid	19
2.3.3 โปรแกรมยูทิลิตี้ของโปรแกรม Squid	19
2.4 โปรแกรม SquidGuard	20
2.4.1 ความสามารถของโปรแกรม SquidGuard	20
2.5 เว็บเซิร์ฟวิสต์	20
2.5.1 เว็บเซิร์ฟวิสต์คืออะไร	20
2.5.2 เว็บแอปพลิเคชันคืออะไร	20
2.6 Packet Filtering	21
2.7 ไฟร์วอลล์	21
2.7.1 ชนิดของไฟร์วอลล์	21
2.7.2 Iptables	22
บทที่ 3 การออกแบบและพัฒนาระบบแคชสำหรับ โปรแกรมมิตทอเรนท์	25
3.1 แบบจำลองของระบบแคชสำหรับโปรแกรมมิตทอเรนท์	25
3.2 องค์ประกอบของระบบแคชสำหรับโปรแกรมมิตทอเรนท์	26
3.2.1 Proxy Server	26
3.2.2 Web Service	26
3.2.3 Internal Tracker	27
3.2.4 Internal Peer	27
3.3 ขั้นตอนการทำงานของระบบแคชสำหรับโปรแกรมมิตทอเรนท์	28
3.3.1 กระบวนการดาวน์โหลด	28
3.3.2 กระบวนการอัปโหลด	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ค้าไฟฟ้าและหลักการทำงานโดยละเอียด	35
4.1 หลักการทำงานหลักของบิตทอเรนท์แคช	35
4.2 หลักการทำงานในส่วนของการดาวน์โหลด	37
4.3 หลักการทำงานในส่วนของการอัปโหลด	38
4.4 หลักการทำงานในส่วนของการแก้ไขไฟล์ทอเรนท์	39
4.5 หลักการทำงานในส่วนของการส่งโปรแกรมทำงาน	40
บทที่ 5 การทดลองและผลการทดลองระบบเครือข่ายสำหรับโปรแกรมบิตทอเรนท์	41
5.1 แบบจำลองและคุณสมบัติของอุปกรณ์ที่ใช้	41
5.2 การตั้งค่าต่างๆในการทดลอง	43
5.2.1 การดาวน์โหลด	44
5.2.1.1 การทำงานทางฝั่ง Client	44
5.2.1.2 การทำงานทางฝั่ง Bittorrent Cache	47
5.2.1 การอัปโหลด	50
5.2.1.1 การทำงานทางฝั่ง Client	50
5.2.1.2 การทำงานทางฝั่ง Bittorrent Cache	53
5.3 การทดลองและผลการทดลอง	55
5.3.1 การทดลองที่ 1	55
5.3.2 การทดลองที่ 2	58
5.4 ปัญหาที่พบระหว่างการทดลองและแนวทางแก้ไข	61
5.5 เปรียบเทียบสภาพทดลองและสภาพการทำงานจริง	62
5.5.1 ปัญหาที่ไม่พบในสภาพการทดลอง	62

สารบัญ (ต่อ)

	หน้า
บทที่ 6 บทวิจารณ์และสรุป	63
6.1 บทสรุป	63
6.2 วิจารณ์สิ่งที่ได้จากโครงการ	63
6.3 ปัญหาอุปสรรคและแนวทางแก้ไข	64
6.4 แนวทางการพัฒนาต่อ	65
บรรณานุกรม	66
ภาคผนวก	67
ภาคผนวก ก ขั้นตอนการติดตั้งระบบแคชสำหรับโปรแกรมบิตทอเรนท์	68
Proxy Server	69
Internal Tracker	77
Internal Peer	83
Packet Filter	86
Web Service	88
ภาคผนวก ข ขั้นตอนการใช้งานระบบแคชสำหรับโปรแกรมบิตทอเรนท์	89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 สถานะการติดต่อกันของ Client กับ Peer	15
5.1 ปริมาณข้อมูลขาเข้าและขาออกจากเครื่องแคช โดยไฟล์ข้อมูลมีขนาด 92 MB	55
5.2 ปริมาณข้อมูลขาเข้าและขาออกจากเครื่องแคช โดยไฟล์ข้อมูลมีขนาด 92 MB	56
5.3 คิวรีโวลต์โดยผ่านระบบแคช	58
5.4 คิวรีโวลต์โดยไม่ผ่านระบบแคช	59



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ระบบภาพรวมของบิตทอเรนทเซอร์ฟเวอร์	5
2.2 ตัวอย่างเว็บไซต์ Bittorrent	6
2.3 การติดต่อกันระหว่าง Client และ Torrent Server เพื่อร้องขอไฟล์ทอเรนท	7
2.4 การติดต่อกันระหว่าง Client กับ Tracker เพื่อร้องขอ Peer list	7
2.5 การติดต่อกันระหว่าง Client กับ Peer เพื่อทำการสร้าง connection การติดต่อก	8
2.6 การเชื่อมต่อกันขณะ Client กำลังดาวน์โหลดข้อมูลใน Bittorrent Network	8
2.7 ตัวอย่างการสร้างไฟล์ torrent จากโปรแกรม Bittorrent (utorrent)	9
2.8 รูปการติดต่อกันระหว่าง Client และ Torrent Server	9
2.9 การติดต่อกันระหว่าง Client กับ Tracker ขณะ Seed ไฟล์	10
2.10 รูปแบบพจนานุกรม info สำหรับไฟล์เดี่ยว	11
2.11 รูปแบบพจนานุกรม info สำหรับไฟล์หลายไฟล์	12
3.1 แบบจำลองของระบบแคชสำหรับโปรแกรมบิตทอเรนท	25
3.2 การทำงานการดาวน์โหลดไฟล์ในขั้นตอนแรก	28
3.3 การทำงานการดาวน์โหลดไฟล์ในขั้นตอนที่สอง	29
3.4 การทำงานการดาวน์โหลดไฟล์ในขั้นตอนที่สาม	29
3.5 การทำงานการดาวน์โหลดไฟล์ในขั้นตอนที่สี่	30
3.6 การทำงานการดาวน์โหลดไฟล์ในขั้นตอนสุดท้าย	31
3.7 การทำงานการอัปโหลดไฟล์ในขั้นตอนแรก	32
3.8 การทำงานการอัปโหลดไฟล์ในขั้นที่สอง	33
3.9 การทำงานการอัปโหลดไฟล์ในขั้นแก้ไขไฟล์ torrent	33
3.10 การทำงานการดาวน์โหลดไฟล์ในขั้นตอนสุดท้าย	34
4.1 แผนผังดาต้าโฟลว์ สำหรับการทำงานหลัก	35
4.2 แผนผังดาต้าโฟลว์ สำหรับการทำงานในส่วนของการดาวน์โหลด	37
4.3 แผนผังดาต้าโฟลว์ สำหรับการทำงานในส่วนของการอัปโหลด	38
4.4 แผนผังดาต้าโฟลว์ สำหรับการทำงานในส่วนของการแก้ไขไฟล์ทอเรนท	39
4.5 แผนผังดาต้าโฟลว์ สำหรับการทำงานในส่วนของการส่งโปรแกรมทำงาน	40
5.1 การตั้งค่าพรีอ็อกซ์ไปยังเครื่อง Bittorrent Cache	44
5.2 ภาพแสดงขณะเครื่อง Client ดาวน์โหลดไฟล์จาก Bittorrent Server	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.3 ไฟล์ที่ได้จากการดาวน์โหลดผ่านทาง proxy server ซึ่งได้ Metadata ด้านในเป็น URL ของ Bittorrent Cache เรียบร้อยแล้ว	45
5.4 ไฟล์ที่เป็นต้นฉบับจากการดาวน์โหลด	46
5.5 ภาพแสดงขณะที่เครื่อง Client ดาวน์โหลดไฟล์ข้อมูลเสร็จเรียบร้อยแล้วสมบูรณ์จากเครื่อง Bittorrent Cache	46
5.6 ภาพขณะที่เครื่อง Bittorrent Cache ดักจับข้อมูลที่เป็น Cookie เพื่อทำการปลอมตัวเป็น Client เพื่อไปดาวน์โหลดไฟล์ที่ต้องการ Bittorrent Server	47
5.7 ไฟล์ torrent ที่ Bittorrent Server (161.246.5.44) ส่งมายัง Bittorrent Cache	48
5.8 ไฟล์ torrent ที่ Bittorrent Cache (161.246.5.43) ทำการเปลี่ยนแปลงข้อมูลภายในไฟล์เรียบร้อยแล้ว	48
5.9 ภาพการทำงานของ TorrentFlux ที่กำลังดาวน์โหลดข้อมูลจากภายนอก และในขณะเดียวกันก็ Seed ไฟล์ดังกล่าวให้กับเครื่องที่อยู่ภายในองค์กรด้วย	49
5.10 ภาพหน้าเว็บไซต์ขณะอัปโหลดไฟล์	50
5.11 ภาพหน้าเว็บไซต์เมื่อทำการอัปโหลดไฟล์สำเร็จและ Client ต้องทำการโหลดไฟล์ torrent อีกครั้ง	51
5.12 ข้อมูลภายในของไฟล์ torrent ที่เครื่อง Client ได้รับมา	51
5.13 หน้าต่างของ Bittorrent Program ขณะทำการ Seed ไฟล์ให้กับ Bittorrent Cache	52
5.14 ข้อมูลของแพ็กเก็ตที่ทำการดักจับได้	53
5.15 ข้อมูลที่ดักจับได้นำไปเพิ่มในฐานข้อมูล	54
5.16 ข้อมูลในไฟล์ torrent ที่เครื่อง Client ได้รับมา	51
5.17 หน้าต่างของ Bittorrent Program ขณะทำการ Seed ไฟล์ให้กับ Bittorrent Cache	52
5.18 แผนภูมิเปรียบเทียบปริมาณข้อมูลขาเข้าจากการดาวน์โหลดของเครื่องภายใน	57
5.19 แผนภูมิเปรียบเทียบปริมาณข้อมูลขาออกจากการดาวน์โหลดของเครื่องภายใน	58
5.20 แผนภูมิเปรียบเทียบเวลาที่ใช้ในการดาวน์โหลดของเครื่องภายใน	60
1ก ภาพตัวอย่างหน้าเว็บไซต์ของ Squid	70
2ก ภาพตัวอย่างหน้าเว็บไซต์ของ Apache-friend :XAMPP	77
3ก ภาพตัวอย่างการสร้างฐานข้อมูลใหม่	79
4ก ตัวอย่างการติดตั้ง phpBB	80
5ก ภาพตัวอย่างหน้าเว็บไซต์ของ TorrentFlux	83

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
1ข ภาพตัวอย่างการใช้งานสั่งให้ Web Server เริ่มทำงาน	90
2ข ภาพตัวอย่างการเริ่มดักจับแพ็คเก็ตจากการดาวน์โหลด	90
3ข ภาพตัวอย่างการเริ่มดักจับแพ็คเก็ตจากการอัปโหลด	91
4ข ภาพตัวอย่างการทำงานการเคลียร์ข้อมูลต่างๆ	91
5ข ภาพตัวอย่างการทำงานการเคลียร์ข้อมูลต่างๆ ขึ้นตอนที่สอง	92



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

ในทุกวันนี้มีการใช้งานอินเทอร์เน็ตกันอย่างแพร่หลายในทุกๆด้านไม่ว่าจะเป็นการใช้งานเว็บไซต์ การรับส่ง e-mail การเล่นเกมออนไลน์ การซื้อขายผ่านทางอินเทอร์เน็ต รวมไปถึงการแลกเปลี่ยนไฟล์ข้อมูล ซึ่งการแลกเปลี่ยนไฟล์ข้อมูลก็มีวิธีการหลายวิธี หนึ่งในนั้นก็คือการแลกเปลี่ยนไฟล์ข้อมูลแบบ P2P (Peer-to-Peer) และโพรโตคอลในการแลกเปลี่ยนไฟล์ข้อมูลแบบ P2P ที่ได้รับความนิยมอยู่ในขณะนี้คือโพรโตคอลบิตทอเรนซ์

โพรโตคอลบิตทอเรนซ์เป็นโพรโตคอลที่คิดขึ้นมาเพื่อจุดประสงค์ทางการแชร์ข้อมูลและส่งข้อมูลอย่างมีประสิทธิภาพโดยตัวโพรโตคอลใช้การติดต่อแบบ peer to peer ซึ่งถือได้ว่ามีประสิทธิภาพการทำงานอย่างยิ่ง และเป็นที่นิยมใช้อย่างมากในปัจจุบัน แต่การทำงานของโพรโตคอลบิตทอเรนซ์มีการใช้ทรัพยากรเครือข่ายสูงมาก หากมีเครื่องที่อยู่ในองค์กรเดียวกันต้องการไฟล์เดียวกันผ่านโพรโตคอลบิตทอเรนซ์ จะทำให้มีการเชื่อมต่อเกิดขึ้นมากเป็นทวีคูณซึ่งในความเป็นจริงแล้ว เมื่ออยู่ในองค์กรเดียวกันก็น่าจะมีการเชื่อมต่อเพียงเท่าเดียวเท่านั้น ถ้าเครื่องที่ต้องการไฟล์เดียวกันนี้มีปริมาณมากขึ้น ก็จะมีปริมาณการเชื่อมต่อมากขึ้นด้วยทั้งที่อาจเป็นร่องขอไฟล์เดียวกัน ซึ่งก่อให้เกิดปัญหาต่างๆกับเครือข่ายอย่างมาก ผลกระทบนี้เป็นปัญหากับองค์กรทั่วไป รวมไปถึง ISP ทุกราย จึงจำเป็นต้องสร้างระบบที่ทำหน้าที่เป็นแคชสำหรับโพรโตคอลบิตทอเรนซ์ที่สามารถลดการเชื่อมต่อสู่ภายนอก โดยเมื่อมีผู้ใช้โปรแกรมบิตทอเรนซ์ดาวน์โหลดไฟล์ข้อมูลที่มีผู้อื่นเคยดาวน์โหลดมาแล้ว ระบบแคชก็จะไปนำไฟล์ข้อมูลในแคชส่งไปให้ผู้ต้องการ และจะไม่ทำการดาวน์โหลดไฟล์ข้อมูลจากอินเทอร์เน็ต ซึ่งจะสามารถลดการเชื่อมต่อออกไปภายนอกเครือข่ายได้ และเวลาที่ใช้ในการดาวน์โหลดไฟล์ข้อมูลที่ซ้ำกันก็จะใช้เวลาน้อยลง นอกจากนี้ยังเป็นการเพิ่มประสิทธิภาพในการใช้ทรัพยากรของเครือข่ายได้อีกมากด้วยเช่นกัน

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อลดภาระการใช้งานไปยังอินเทอร์เน็ตของโปรแกรมบิตทอเรนซ์ให้น้อยลง และทำให้เครื่องของบุคคลอื่นภายในองค์กรสามารถใช้งานอินเทอร์เน็ตและบริการอื่นๆ ได้ดีขึ้น
2. เพื่อให้โปรแกรมบิตทอเรนซ์สามารถทำงานได้อย่างมีประสิทธิภาพขึ้นไม่มีการทำงานที่ซ้ำซ้อนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงการ

สำหรับโครงการระบบแคชสำหรับ โปรแกรมบิตทอเรนท์นี้ ได้กำหนดขอบเขตของการศึกษาและออกแบบระบบไว้ดังนี้

1. เพื่อให้สามารถนำโปรแกรมไปใช้งานได้จริงตามความต้องการของภายในองค์กร
2. เพื่อให้สามารถตอบสนองการทำงานของโปรโตคอลบิตทอเรนท์ และบิตทอเรนท์เซิร์ฟเวอร์ในปัจจุบัน
3. มี logbase เพื่อสามารถดูการทำงานของบิตทอเรนท์แคชได้ตลอดเวลาโดยผ่านทางเว็บเซอร์วิส
4. สามารถควบคุมการทำงานของโปรแกรมบิตทอเรนท์ที่เครื่องพีร็อกซีเซิร์ฟเวอร์ได้ผ่านทางหน้าเว็บไซต์ เพื่อเพิ่มความสะดวกและง่ายต่อการจัดการต่างๆ

1.4 วิธีการดำเนินการ

1. ศึกษาทฤษฎีและการทำงานของโปรโตคอลบิตทอเรนท์ การทำงานของแทรคเกอร์ การทำงานของโปรแกรมบิตทอเรนท์ ซึ่งนำมาใช้ในการออกแบบระบบ
2. ศึกษาทฤษฎีของพีร็อกซีเซิร์ฟเวอร์ และโปรแกรมที่เกี่ยวข้องกับการทำ Redirect Proxy ได้แก่ โปรแกรม Squid และ โปรแกรม SquidGuard
3. ศึกษาหลักการการทำงาน การติดตั้ง และการปรับแต่งค่าของเว็บเซิร์ฟเวอร์
4. ศึกษาการทำงาน คำสั่งต่างๆ การกำหนด Permission และ Owner ให้กับไฟล์ ระบบไฟล์และไคลเอนต์ของระบบปฏิบัติการ Linux หรือ UNIX ที่นำมาใช้งาน
5. ศึกษาหลักการการทำงาน การติดตั้ง และการปรับแต่งค่าของบิตทอเรนท์เซิร์ฟเวอร์ และบิตทอเรนท์แทรคเกอร์
6. ศึกษาการดักจับแพ็กเก็ตต่างๆด้วยโปรแกรม Tcpdump และHTTPDump
7. วิเคราะห์และออกแบบระบบ
8. เตรียมอุปกรณ์ที่จำเป็นต้องใช้ในการติดตั้งระบบแคชสำหรับ โปรแกรมบิตทอเรนท์
9. ทำการติดตั้งและปรับแต่งค่าสำหรับองค์ประกอบต่างๆของระบบ ได้แก่ การติดตั้ง Squid และ SquidGuard สำหรับ Redirect Proxy การติดตั้งแทรคเกอร์และโปรแกรมบิตทอเรนท์ไคลเอนต์
10. ทำการติดตั้งและปรับแต่งโปรแกรมในการดักจับแพ็กเก็ต ได้แก่ โปรแกรม Tcpdump และโปรแกรม Snort-Inline
11. พัฒนาเว็บเซอร์วิสของระบบแคชสำหรับ โปรแกรมบิตทอเรนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ทำการทดสอบและบันทึกผลของระบบแคชสำหรับโปรแกรมบิตทอเรนท์
13. สรุปและวิจารณ์ผลของการทำโครงการ รวมไปถึงเสนอแนะแนวทางการพัฒนาต่อไป

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้และความเข้าใจเกี่ยวกับวิธีการทำงานของโปรโตคอลบิตทอเรนท์ การทำงานของแทรคเกอร์และการทำงานของโปรแกรมบิตทอเรนท์ไคลเอนต์
2. ได้รับความรู้ ความเข้าใจทั้งในด้านทฤษฎีและด้านปฏิบัติเกี่ยวกับการศึกษาและออกแบบในการทำระบบแคชสำหรับโปรแกรมบิตทอเรนท์
3. สามารถออกแบบระบบแคชสำหรับ โปรแกรมบิตทอเรนท์ให้ทำงานได้

1.6 ส่วนประกอบของปฏิญานิพนธ์

ปฏิญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องที่ใช้ในโครงการ ซึ่งประกอบด้วยทฤษฎีเกี่ยวกับบิตทอเรนท์ ทฤษฎีของพรีอ็อกซีเซิร์ฟเวอร์ โปรแกรม Squid โปรแกรม SquidGuard และทฤษฎีของเว็บเซอร์วิส Packet Filtering และไฟร์วอลล์

บทที่ 3 กล่าวถึงการออกแบบและแบบจำลองระบบแคชสำหรับโปรแกรมบิตทอเรนท์ โดยอธิบายหน้าที่และการทำงานขององค์ประกอบต่างๆที่ใช้ในบิตทอเรนท์เน็ตเวิร์คและระบบแคชสำหรับโปรแกรมบิตทอเรนท์

บทที่ 4 กล่าวถึงข้อมูลและหลักการการทำงานของระบบแคชสำหรับโปรแกรมบิตทอเรนท์โดยละเอียด

บทที่ 5 กล่าวถึงเรื่องของการทดลองระบบแคชสำหรับ โปรแกรมบิตทอเรนท์ เภณท์ที่ใช้ในการประเมิน และผลจากการทดลองระบบ

บทที่ 6 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงการ วิจารณ์สิ่งที่ได้รับจากโครงการ รวมไปถึงข้อเสนอนแนะสำหรับเป็นแนวทางในการพัฒนาระบบต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้ในโครงการ

ก่อนที่จะทำการออกแบบหรือพัฒนาระบบใดๆ ขึ้นมานั้น การศึกษาหาข้อมูลของทฤษฎีที่เกี่ยวข้องนั้นเป็นสิ่งจำเป็นที่ต้องกระทำอย่างหลีกเลี่ยงไม่ได้ เพื่อให้ระบบที่ต้องการออกแบบหรือพัฒนานั้นประสบความสำเร็จสูงสุดและเกิดความผิดพลาดน้อยที่สุด

ในบทนี้จะกล่าวถึงทฤษฎีต่างๆที่เกี่ยวข้องกับการทำระบบแลชสำหรับโปรแกรมบิตทอเรนท์ ได้แก่ โพรโตคอลบิตทอเรนท์ องค์ประกอบและการทำงานของบิตทอเรนท์ พร็อกซี เซิร์ฟเวอร์ โปรแกรม Squid โปรแกรม SquidGuard การดักจับแพ็คเกจ ไฟร์วอลล์ และเว็บเซอร์วิส ซึ่งทฤษฎีที่เกี่ยวข้องนี้จะนำไปใช้ในการออกแบบระบบและพัฒนาระบบในบทต่อไป

2.1 บิตทอเรนท์

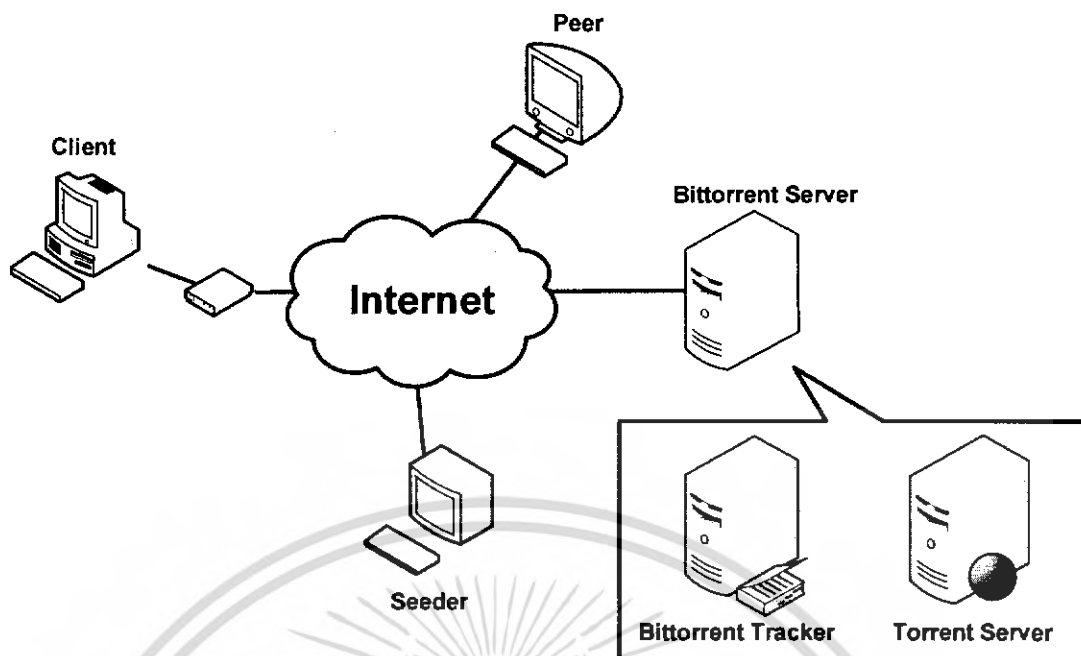
2.1.1 บิตทอเรนท์คืออะไร

บิตทอเรนท์เป็นชื่อของ โพรโตคอลและอุปกรณ์การแจกจ่ายข้อมูลแบบ Peer-to-peer ที่ถูกพัฒนาโดยผู้พัฒนา Bram Cohen โดยเริ่มเป็นที่รู้จักครั้งแรกในงาน CodeCon ปี ค.ศ. 2002 ในครั้งแรกการทำงานเขียนโดยภาษาไพทอนและจดสิทธิบัตรในแบบโอเพ่นซอร์สภายใต้เวอร์ชัน 4.0 คำว่า BitTorrent สามารถใช้อ้างถึงตัวโพรโตคอลในการแจกจ่ายไฟล์ ตัวโปรแกรมประยุกต์ ส่วนลูกข่าย และไฟล์นามสกุล .torrent ได้

โพรโตคอลบิตทอเรนท์จะทำการแบ่งไฟล์ออกเป็นหลายส่วนย่อยๆ โดยประมาณคือหนึ่งในสี่ของเมกะไบต์ แต่ละส่วนจะถูกแจกจ่ายไปยังเครื่อง peer โดยสุ่มลำดับ ซึ่งจะประกอบกลับเป็นไฟล์ไฟล์หนึ่ง ในเครื่องที่ขอไฟล์ไป แต่ละ peer จะใช้ประโยชน์จากการเชื่อมต่อที่ดีที่สุดไปยังส่วนที่ยังขาดไป ในขณะที่ส่งส่วนที่มีอยู่ให้เครื่องอื่นๆ ได้รับ รูปแบบการทำงานอย่างนี้มีประโยชน์อย่างมากในการแลกเปลี่ยนไฟล์ขนาดใหญ่ อย่างเช่น วิดีโอหรือโค้ดต้นแบบของโปรแกรม ในการดาวน์โหลดโดยทั่วไปแล้ว ไฟล์ที่มีความต้องการสูงมักก่อให้เกิดสถานะคอขวดซึ่งผู้ที่ต้องการไฟล์จะใช้แบนวิดท์ของเครื่องให้บริการจนหมด แต่ใน BitTorrent นั้นไฟล์ที่เป็นที่ต้องการอย่างมากจะถูกช่วยในการส่งไฟล์เมื่อแบนวิดท์และจำนวนเครื่องปล่อยไฟล์ (seed) ที่สมบูรณ์มีมากขึ้น

2.1.2 การทำงานของบิตทอเรนท์

ก่อนที่จะทำการออกแบบระบบแลชสำหรับโปรแกรมบิตทอเรนท์นั้น จะต้องศึกษาทฤษฎีหลักการทำงานของโพรโตคอลบิตทอเรนท์เสียก่อน โดยพบว่าระบบการทำงานของโพรโตคอลบิตทอเรนท์นั้นมีองค์ประกอบที่สำคัญต่างๆ ดังรูปที่ 2.1



รูปที่ 2.1 ภาพรวมของระบบบิตทอเรนทเซิร์ฟเวอร์

1. Torrent Server

Torrent Server คือ เครื่องเซิร์ฟเวอร์ที่เก็บไฟล์ .torrent ไว้ โดย Client จะมาโหลดไฟล์ .torrent จากที่นี่ ซึ่ง Torrent Server จะทำหน้าที่เป็น Tracker ด้วย

2. Tracker

Tracker มีหน้าที่ที่สำคัญ คือ คอยแจกจ่ายหมายเลข ไอพีของผู้ที่มีไฟล์ข้อมูล Client ต้องการกลับไปยัง Client และคอยทำหน้าที่เก็บข้อมูลสถิติต่างๆของ seeder และ peer ที่เคยติดต่อกับ Tracker นี้

3. Seeder

Seeder คือ ผู้ที่มีไฟล์ข้อมูลที่สมบูรณ์และเป็นผู้ที่คอยส่งชิ้นของไฟล์ข้อมูลที่มีอยู่ให้กับ Client ที่ต้องการไฟล์ข้อมูลนั้นๆ

4. Peer หรือ Leecher

Peer หรือ Leecher คือ ผู้ที่มีความต้องการที่จะได้ไฟล์ข้อมูล โดยการรับชิ้นส่วนของไฟล์ข้อมูลนั้น เป็นไปได้ทั้งการรับมาจากตัว Seeder และการแลกเปลี่ยนชิ้นส่วนของไฟล์ข้อมูลระหว่าง Peer ด้วยกันเอง

5. Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Client คือ ผู้ที่ใช้งานโปรแกรมบิตทอเรนทเพื่อดาวน์โหลดไฟล์ข้อมูลของตนเอง ต้องการ ซึ่งในความเป็นจริงแล้ว Client กับ Peer นั้นคือลักษณะเดียวกัน แต่ในที่นี้จะขอ กำหนดให้เป็น Client เพื่อง่ายต่อการอธิบายว่า Client คือผู้ที่กำลังร้องขอและต้องการ ไฟล์ที่สนใจ ซึ่งสามารถรับชิ้นส่วนไฟล์ได้จากทั้ง Seeder และ Peer อื่น

ในการทำงานของโปรโตคอลบิตทอเรนท จะแบ่งการทำงานไว้สองแบบคือ การดาวน์โหลด และการอัปโหลด

2.1.2.1 การดาวน์โหลด

โดยปกติ Client จะเข้าไปยังหน้าเว็บไซต์ของบิตทอเรนทเซิร์ฟเวอร์ โดยในเว็บไซต์จะมี ไฟล์ทอเรนทซึ่ง Seeder อัปโหลดไว้ Client จะทำการเลือกไฟล์ที่ต้องการ ดังรูป 2.2

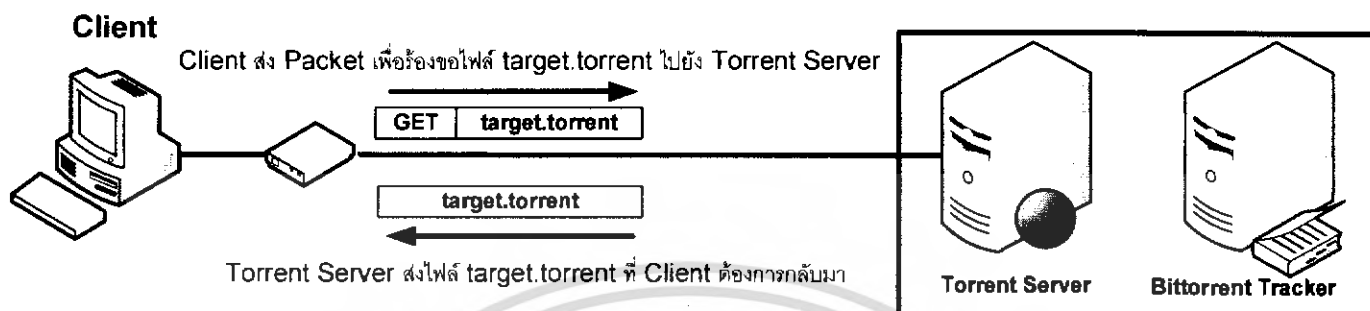
The screenshot shows the Thailand BitTorrent Tracker interface. At the top, there's a navigation bar with tabs for General, Torrents, Personal, Entertainment, and Misc. Below this is a search bar and a list of torrents. The torrents list includes:

Type	Name	Files	Comps.	Added	Time A/V	Size	Snatched	Seeders	Leechers	Uploaded by
DVD	Jurassic Park I The Collection 1.D9>D5 Master Zone 3 coloz1 (NEW)	6	0	2006-12-25 16:11:54	0 hours	13.06 GB	0 times	1	0	wacmaster
Software	Portable Internet Download Manager 5.03 Build 2 - Full (NEW)	1	0	2006-12-25 16:08:27	0 hours	1.73 MB	0 times	1	1	Moray
Software	เพลงกลอนกลอน (NEW)	233	0	2006-12-25 16:02:57	0 hours	2.55 GB	0 times	1	0	chloe
Software	P.O.P.-51 Period of party (NEW)	14	0	2006-12-25 16:00:19	0 hours	54.32 MB	0 times	1	0	Anonymous

รูปที่ 2.2 ตัวอย่างเว็บไซต์ Bittorrent

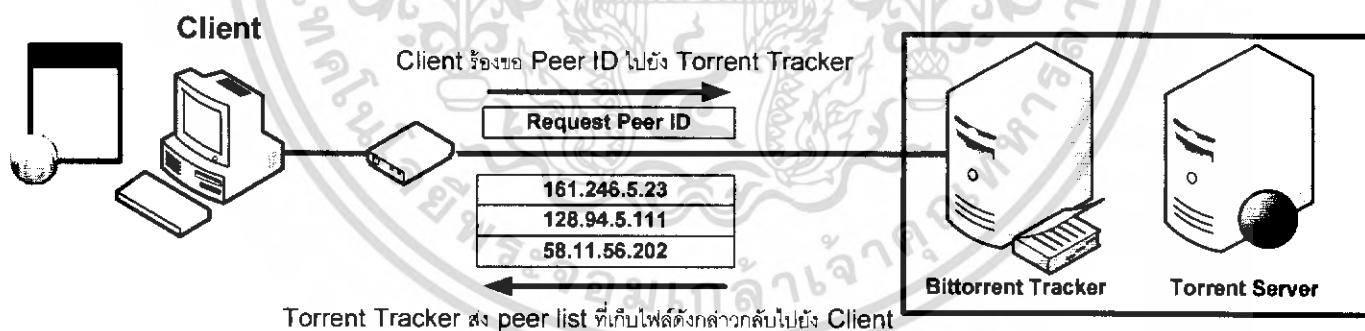
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อเริ่มต้น Client จะต้องมีไฟล์ .torrent ของไฟล์ที่ต้องการ (โดยในที่นี้จะขอกำหนดไฟล์ torrent ที่ Client ต้องการเป็น target.torrent) โดย Client จะต้องไปดาวน์โหลดไฟล์ target.torrent ที่ต้องการจาก Torrent Server ซึ่งเป็นเว็บเซิร์ฟเวอร์ดังรูปที่ 2.3



รูปที่ 2.3 การติดต่อกันระหว่าง Client และ Torrent Server เพื่อร้องขอไฟล์ทอเรนท

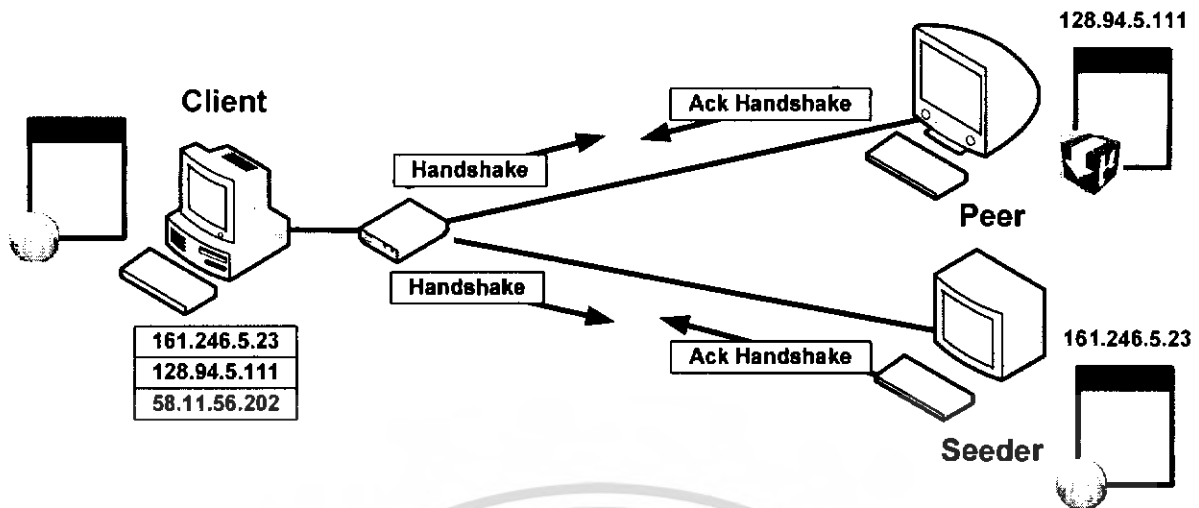
2. เมื่อ Client ได้ไฟล์ .torrent จาก Torrent Server แล้วก็จะทำการรันโปรแกรมบิตทอเรนทขึ้นมา จากนั้น Client จะติดต่อกับ Tracker ตามที่ประกาศเอาไว้ในไฟล์ .torrent (ซึ่งโดยทั่วไป Torrent Server และ Tracker มักจะอยู่ที่เดียวกัน) ที่ได้โหลดมา จากนั้นตัว Tracker จะส่งหมายเลขไอพีและ Peer ID ของผู้ที่มีไฟล์ที่เครื่อง Client ต้องการกลับมาให้ Client ดังรูปที่ 2.4



รูปที่ 2.4 การติดต่อกันระหว่าง Client กับ Tracker เพื่อร้องขอ Peer list

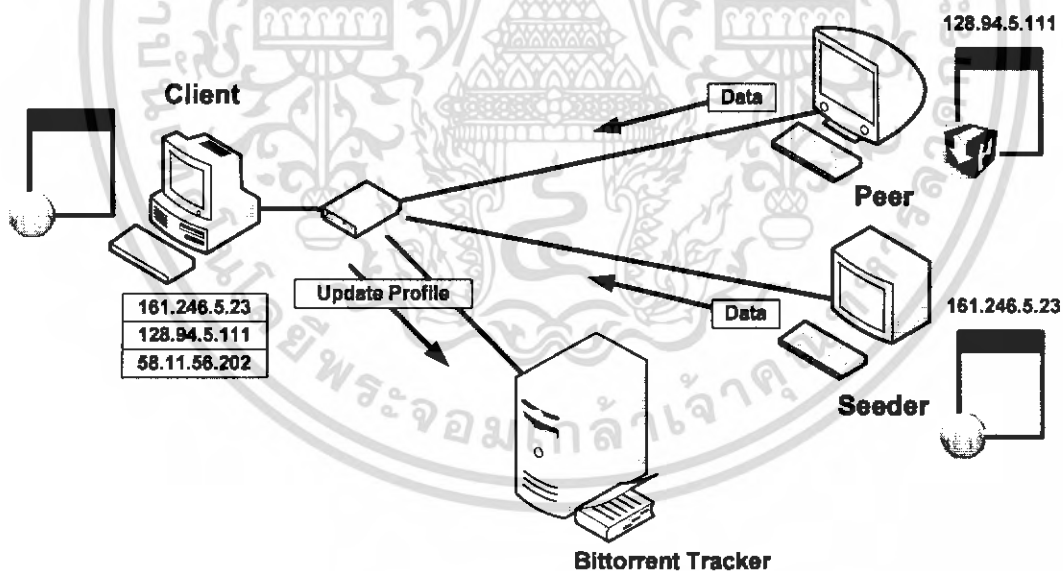
3. เมื่อ Client ได้หมายเลขไอพีและ Peer ID มาแล้ว ก็จะทำการติดต่อไปยังแต่ละ Peer ตามหมายเลขไอพีที่ได้มา โดยจะส่งข้อความไปทำ Handshake กับ Peer นั้นๆ ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 การติดต่อกันระหว่าง Client กับ Peer เพื่อทำการสร้าง connection การติดต่อ

4. จากนั้น Client ก็จะมีการดาวน์โหลดและแลกเปลี่ยนชิ้นส่วนของไฟล์กับ Peer หรือ Seed ที่ติดต่อได้ ระหว่างนั้น Client ก็จะมีการติดต่อกับ Tracker เป็นระยะๆ เพื่ออัปเดตข้อมูลของตัวเองกับ Tracker และรับข้อมูลใหม่ๆจาก Tracker ดังรูป 2.6

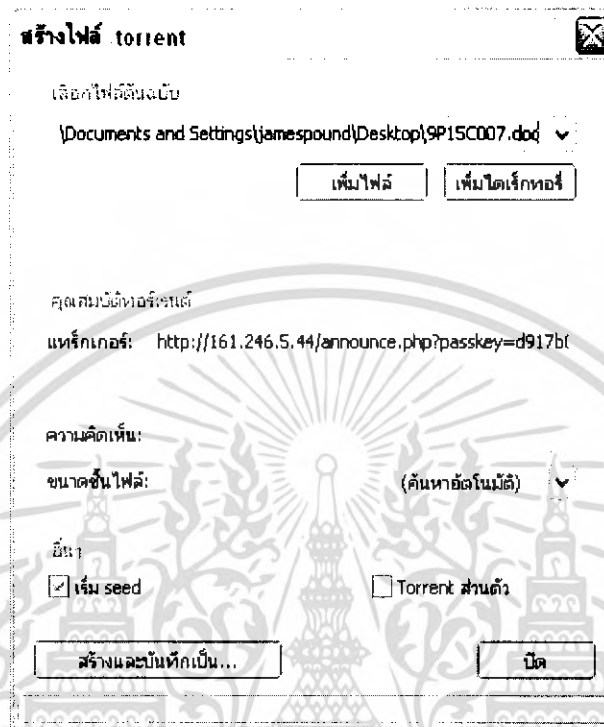


รูปที่ 2.6 การเชื่อมต่อกันขณะ Client กำลังดาวน์โหลดข้อมูลใน Bittorrent Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

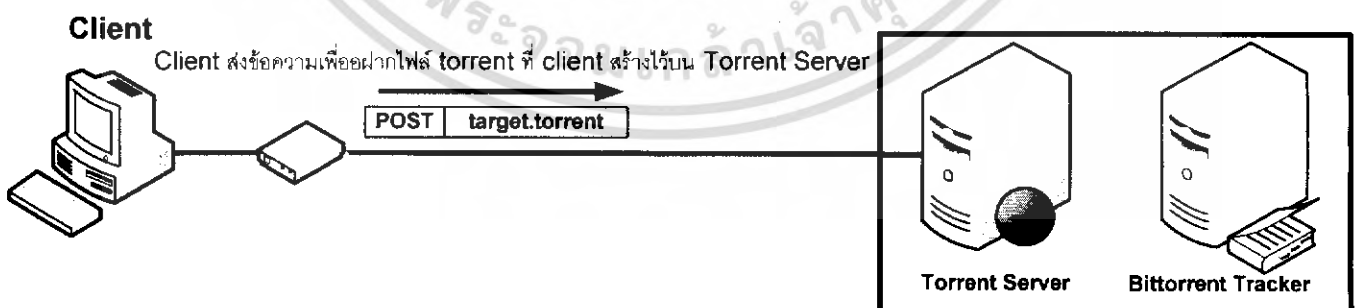
2.1.2.2 การอัปโหลด

โดยปกติแล้ว Client จะต้องทำหน้าที่เป็น Seeder ดังนั้น Client จึงต้องทำการสร้างไฟล์ torrent ขึ้นมา ซึ่งจะสร้างมาจากโปรแกรม Bittorrent ดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างการสร้างไฟล์ torrent จากโปรแกรม Bittorrent (utorrent)

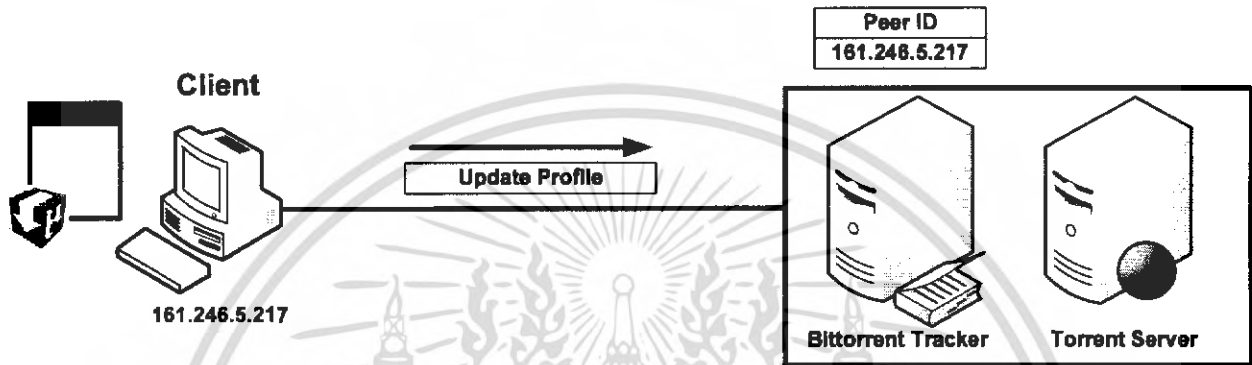
1. เมื่อเริ่มต้น Client จะต้องทำการฝากไฟล์ไว้ที่ Torrent Server ซึ่งเมื่อมีผู้ใช้อื่นต้องการไฟล์ที่ Client กำลัง Seed ไว้ ผู้นั้นก็จะมีดาวน์โหลดไฟล์ torrent จากที่ Torrent Server แล้วนำไฟล์นั้น ไปรันเพื่อขอดาวน์โหลดด้วยวิธีการที่กล่าวไว้ข้างต้น ดังรูปที่ 2.8



รูปที่ 2.8 รูปการติดต่อระหว่าง Client และ Torrent Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ซึ่ง torrent file ในขณะนี้ได้ถูกฝากไว้ใน Torrent Server ขั้นตอนต่อไปคือ Client จะต้องดาวน์โหลดไฟล์ torrent ที่ตัวเองฝากไว้มาและนำไฟล์ดังกล่าวไปใช้งานกับ โปรแกรม Bittorrent
3. ในขณะที่โปรแกรม Bittorrent ทำงานอยู่นั้นก็จะมีการรายงานผลไปยัง tracker เป็นระยะ เพื่อบอกว่ามีการปล่อยไฟล์อยู่ ซึ่ง Torrent Tracker ก็จะเก็บไอพีแอดเดรสของเครื่องดังกล่าวไว้เป็น Peer ID ดังรูปที่ 2.9



รูปที่ 2.9 การติดต่อกันระหว่าง Client กับ Tracker ขณะ Seed ไฟล์

4. จากนั้น ถ้าหากมี Peer อื่นต้องการไฟล์ดังกล่าวก็จะทำไปยังกระบวนการดาวน์โหลดต่อไป

2.1.3 รายละเอียดของไฟล์ .torrent

ไฟล์นามสกุล .torrent จะมีข้อมูลที่เกี่ยวข้องกับไฟล์จริงๆที่จะทำการดาวน์โหลด เรียกข้อมูลนี้ว่า Metainfo ซึ่งถูกจัดรูปแบบในลักษณะ Bencode (อ่านว่า บี-เอนโค้ด) โดยมีรายละเอียดของรูปแบบดังนี้

- ข้อมูลที่เป็น String จะอยู่ในรูป <ความยาวของstringในเลขฐานสิบ>:<ข้อมูลstring> เช่น 4:spam
- ข้อมูลที่เป็นจำนวนเต็มจะอยู่ในรูป i<จำนวนเต็มในAsciiฐานสิบ>e โดยสามารถใส่เครื่องหมายบอกจำนวนลบไว้ก่อนตัวเลขได้ จำนวนเต็มศูนย์ให้ใส่เพียงเลขศูนย์ เช่น i7e i-3e i0e
- ชุดข้อมูล (list) จะอยู่ในรูป l<ข้อมูลในรูปแบบbencode>e ซึ่งในชุดข้อมูลสามารถประกอบไปด้วยข้อมูลในรูปแบบ bencode ที่เป็น string จำนวนเต็ม พจนานุกรม และ/หรือชุดข้อมูลอื่นๆก็ได้
- dictionary จะอยู่ในรูป d<bencoded string><bencoded element>e โดยตัวตั้งต้นจะต้องเป็น string ส่วนค่านั้นจะเป็นอะไรก็ได้ที่อยู่ในรูปแบบ bencode และหากมีมากกว่าหนึ่งตัวตั้งต้นจะต้องทำการจัดลำดับตัวตั้งต้นตามลำดับ alphabet ของ string เช่น d4:bone7:calcium3:cow3:mooe ซึ่งก็คือ bone=calcium และ cow=moo

โครงสร้างของ Metainfo ทุกอย่างจะอยู่ในรูปของ bencode โดย metainfo ในไฟล์นามสกุล .torrent เป็น bencoded dictionary ที่มีข้อมูลกล่าวไว้ด้านล่าง ทุกตัวอักษร string อยู่ในรูป UTF-8 โดยส่วนใหญ่ไม่ได้ระบุว่าเป็นตัวเลือกคือข้อมูลที่จำเป็น

- info: เป็น dictionary ที่บรรยายลักษณะของไฟล์หนึ่งๆหรือหลายๆไฟล์ใน torrent ซึ่งสามารถมีได้สองลักษณะ คือแบบไฟล์เดี่ยวที่ไม่ต้องมีโครงสร้าง Directory กับแบบไฟล์ torrent ที่บอกถึงหลายไฟล์

สำหรับไฟล์เดี่ยว รูปแบบพจนานุกรม info จะมีโครงสร้างดังรูปที่ 2.10 นี้

Info	Length	MD5 Checksum	Name	Piece	Pieces
Announce	Announce List	Create Date	Comment	Created by	

รูปที่ 2.10 รูปแบบพจนานุกรม info สำหรับไฟล์เดี่ยว

- length: ความยาวของไฟล์ (จำนวนเต็ม)
- md5sum: (ตัวเล็ก) string ความยาว 32 ตัวอักษรในรูปเลขฐาน 16 ที่ตรงกับ MD5 Sum ของไฟล์ ข้อมูลนี้แทบไม่ถูกใช้โดยโปรแกรม BitTorrent เลย แต่ถูกรวมไว้ในบางโปรแกรมเพื่อเพิ่มความยืดหยุ่นในการใช้งาน
- name: ชื่อของไฟล์ (string)
- piece length: ความยาวของแต่ละชิ้นส่วนของไฟล์ในหน่วยไบต์ ตัวเลขจะเป็นกำลังของ 2 ขึ้นไป โดยปกติจะตั้งไว้ให้น้อยกว่าหรือเท่ากับ 512 KB (จำนวนเต็ม)
- pieces: string ที่มีค่า SHA-1 hash ของแต่ละชิ้น โดยค่าเป็น 20 bytes ต่อหนึ่งชิ้น ทั้งหมดต่อกันเป็น string เดียวที่มีความยาวเป็นผลคูณของ 20

สำหรับไฟล์หลายไฟล์ รูปแบบพจนานุกรม info จะมีโครงสร้างดังรูปที่ 2.11 นี้

Info	Files	Length	MD5	Checksum
Path	Name	Piece Length	Pieces	
Announce	Announce List	Create Date	Comment	Created by

รูปที่ 2.11 รูปแบบพจนานุกรม info สำหรับไฟล์หลายไฟล์

- files: ชุดข้อมูลของพจนานุกรม แต่ละอันสำหรับแต่ละไฟล์ แต่ละพจนานุกรมจะมีชุดข้อมูลดังนี้
 - length: ความยาวของไฟล์ (จำนวนเต็ม)
 - md5sum: (ตัวเล็ก) string ความยาว 32 ตัวอักษรในรูปเลขฐาน 16 ที่ตรงกับ MD5 Sum ของไฟล์ ข้อมูลนี้แทบไม่ถูกใช้โดยโปรแกรม BitTorrent เลย แต่ถูกรวมไว้ในบางโปรแกรมเพื่อเพิ่มความยืดหยุ่นในการใช้งาน
 - path: ชุดข้อมูลที่บันทึกหนึ่ง string หรือมากกว่าเพื่อบอก path และชื่อไฟล์ แต่ละค่าจะบอกถึงชื่อของ directory หรือ (ค่าสุดท้ายในชุดข้อมูล) ชื่อไฟล์
 - name: ชื่อของ directory บนสุดของโครงสร้าง—Directory ที่มีไฟล์ทุกไฟล์ตามที่กล่าวไว้ในชุดข้อมูล files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- piece length: ความยาวของแต่ละชิ้นส่วนของไฟล์ในหน่วยไบต์ ตัวเลขจะเป็นกำลังของ 2 ขึ้นไป โดยปกติจะตั้งไว้ให้น้อยกว่าหรือเท่ากับ 512 KB (จำนวนเต็ม)
- pieces: string ที่มีค่า SHA-1 hash ของแต่ละชิ้น โดยค่าเป็น 20 bytes ต่อหนึ่งชิ้น ทั้งหมดต่อกันเป็น string เดียวที่มีความยาวเป็นผลคูณของ 20

หลังจาก info จะเป็นรูปแบบที่ทั้งไฟล์เดี่ยวและหลายไฟล์ใช้เหมือนกัน

- announce: URL ของ tracker (string)
- announce-list: (ตัวเลือก) ส่วนขยายของข้อมูลปกติเพื่อบอก tracker สำรองในกรณีที่ tracker หลักล้มหรือติดต่อไม่ได้
- create date: (ตัวเลือก) เวลาที่สร้างไฟล์นามสกุล .torrent นี้ขึ้น ใช้รูปแบบ Unix epoch โดยนับเวลาเป็นวินาทีจาก 1 Jan 1970 ณ เวลา 00:00:00 UTC (จำนวนเต็ม)
- comment: (ตัวเลือก) ข้อความของผู้สร้าง (string)
- created by: (ตัวเลือก) ชื่อและรุ่นของโปรแกรมที่ใช้สร้างไฟล์ .torrent นี้ (string)

2.1.4 การทำงานของแทรคเกอร์

ในการเชื่อมต่อระหว่างแต่ละเครื่อง Peer นั้น จะมี BitTorrent Tracker เป็นตัวช่วยในการแลกเปลี่ยนข้อมูลสถานะของแต่ละ peer เพื่อให้ทราบว่า มีเครื่องใดบ้างที่ต้องการหรือมีไฟล์ดังกล่าวที่ตรงกับไฟล์นามสกุล .torrent บอกไว้ แทรคเกอร์มีหลายแบบ ที่พัฒนาแรกสุดคือ HTTP/HTTPS protocol Tracker พัฒนาในภาษาไพทอน ซึ่งตอบสนองต่อคำสั่ง HTTP GET คำสั่งที่ส่งมาจะมี ค่า metrics จากเครื่องลูกข่ายที่ช่วยแทรคเกอร์ในด้านสถิติของ torrent นั้นๆ การตอบสนองจะมี peer list ที่ช่วยให้เครื่องลูกข่ายทราบถึงเครื่องอื่นๆที่เข้าร่วมงาน torrent นี้ ปกติแล้วคำสั่งจะใช้ URL จาก announce ใน metainfo ตามด้วย Parameter (ใช้ ? หลัง announce URL ตามด้วยชุด parameter=value ซึ่งขึ้นจากกันด้วย &) ซึ่งค่าทั้งหมดนั้น ถ้าไม่ได้อยู่ในรูป 0-9 a-z A-Z และ \$-_.+!*() จะต้องอยู่ในรูป %nn โดยที่ nn เป็นค่าเลขฐานสิบหกที่ตรงกับไบต์ตัวอักษรนั้นๆ

2.1.5 การทำงานในรูปแบบ Peer-2-Peer ของเครื่องลูกข่าย

หลังจากมี Peers Table แล้ว โปรแกรมลูกข่าย(Client) จะติดต่อระหว่างกันโดยใช้ Peer Wire Protocol (TCP) เพื่อแลกเปลี่ยนชิ้นส่วนของข้อมูลที่ Torrent ที่ทำงานอยู่นั้นกล่าวถึง ซึ่งในจำนวน peer ที่มีทั้งหมดใน peer table นั้น โปรแกรมลูกข่ายจะไม่ได้ติดต่อไปทั้งหมดเพื่อรักษา Load Balance โดยมีการระบุสถานะกับแต่ละ peer ไว้ ซึ่งมี

- choked: บ่งบอกว่าเครื่องลูกข่ายนั้นถูกปิดกั้น โดย peer หรือไม่ ซึ่งในขณะที่ถูกปิดกั้น คำร้องขอต่างๆจะไม่ได้รับการตอบรับจนกว่าจะยุติการปิดกั้น เครื่องลูกข่ายควรระวังการร้องขอชิ้นส่วน และรู้ว่าคำร้องที่ไม่ได้รับการตอบจะถูกละเลยจาก peer
- interested: บ่งบอกว่า Peer สนใจที่จะขอข้อมูลจากเครื่องลูกข่าย ซึ่งทันทีที่ได้ยุติการปิดกั้น peer จะเริ่มส่งคำขอไปยังเครื่องลูกข่ายทันที

ซึ่งในความเป็นจริง ทั้ง Peer และเครื่องลูกข่ายนั้นนับเป็น peer เช่นกัน ดังนั้นข้อมูลสถานะจริงๆจะเป็นดังนี้

- am_choking: เครื่องลูกข่ายนี้กำลังปิดกั้น peer
- am_interested: เครื่องลูกข่ายนี้สนใจ peer
- peer_choking: เครื่อง peer กำลังปิดกั้นเครื่องลูกข่ายนี้
- peer_interested: เครื่อง peer สนใจเครื่องลูกข่าย

สถานะในการเริ่มการเชื่อมต่อจะเป็นการปิดกั้นและไม่สนใจจากทั้งสองฝ่าย ชิ้นส่วนจะถูกส่งมาสู่เครื่องลูกข่ายก็ต่อเมื่อ peer ไม่ได้ปิดกั้นเครื่องลูกข่าย และเครื่องลูกข่ายสนใจในตัว peer สิ่งสำคัญที่เครื่องลูกข่ายควรทำก็คือการบอก peer ทุก peer ถึงความสนใจที่มีต่อแต่ละ peer ซึ่งควรบอกไว้ทุกครั้งที่มีการเปลี่ยนแปลงแม้ว่า peer จะปิดกั้นอยู่เพื่อให้ peer สามารถรับรู้ได้ว่าเครื่องลูกข่ายจะโหลดข้อมูลจาก peer ทันทีที่ยุติการปิดกั้น

ตารางที่ 2.1 สภาวะการติดต่อกันของ Client กับ Peer

Peer to Client	Client to Peer			
	Interested	Not Interested	Choking	Unchoking
Interested	ต่างก็มีชิ้นที่อีกฝั่งยังขาด	มีชิ้นที่ peer ยังขาด	พิจารณาการ unchoke Peer	ส่งชิ้นส่วนให้ Peer
Not Interested	Peer มีชิ้นที่ยังขาด	หลังจากหนึ่ง จะเลิกติดต่อ	ไม่จำเป็นต้อง unchoke	x
Choking	รอส่ง request	ไม่มีชิ้นส่วน	สถานะเริ่มต้น	x
Unchoking	รับชิ้นส่วนจาก Peer	x	x	x

2.2 พร็อกซีเซิร์ฟเวอร์

2.2.1 พร็อกซีเซิร์ฟเวอร์คืออะไร

พร็อกซีเซิร์ฟเวอร์หรืออาจเรียกกันว่าแคชเป็นอุปกรณ์เครือข่ายชนิดหนึ่ง ซึ่งจะทำการเก็บข้อมูลของหน้าเว็บเพจต่างๆ ที่เราเรียกมาดูนั่นเอง โดยจะทำหน้าที่เป็นเสมือนที่เก็บข้อมูลหน้าเว็บเพจที่มีผู้เคยเรียกดูหน้าเว็บนั้นไว้ในระบบของเซิร์ฟเวอร์และเมื่อมีผู้อื่นเรียกดูข้อมูลที่เป็นหน้าเว็บเพจตัวเดียวกันกับที่เคยมีเก็บไว้แล้วก็ไม่จำเป็นต้องไปเรียกอ่านข้อมูลมาจากเซิร์ฟเวอร์ตัวต้นฉบับจริง เพียงแค่ค้นหาข้อมูลเว็บเพจนั้นจากพร็อกซีก่อน ซึ่งถ้าหากหาพบหรือมีผู้อื่นเคยเรียกอ่านแล้วระบบพร็อกซีเซิร์ฟเวอร์ก็สามารถส่งข้อมูลนั้นมาให้ได้ทันที ซึ่งวิธีนี้จะทำให้สามารถเรียกดูข้อมูลได้เร็วกว่าการเรียกมาจากเว็บไซด์ต้นฉบับจริงๆ แต่ข้อเสียของการตั้งใช้ พร็อกซีเซิร์ฟเวอร์ คือ ข้อมูลที่มีการอัปเดตเร็วๆ เช่นเว็บบอร์ดต่างๆ อาจจะไม่มีการอัปเดตตามได้ทันที เพราะจะได้ข้อมูลทีมาจากพร็อกซีไม่ใช่มาจากต้นฉบับหรือแหล่งกำเนิดจริงๆ ในกรณีเช่นนี้ อาจจะทำให้การแก้ไขเบื้องต้น ได้โดย การกดปุ่ม Ctrl พร้อมๆ กับการใช้เมาส์คลิกที่ปุ่ม Refresh จะเป็นการร้องขอ ข้อมูล หน้าเว็บเพจนั้นๆ ใหม่อีกจากเซิร์ฟเวอร์ต้นฉบับ

2.2.2 หลักการทำงานของพร็อกซีเซิร์ฟเวอร์

เมื่อมีผู้ใช้บริการทำการเรียกข้อมูลของเว็บไซต์โดยผ่านพร็อกซีเซิร์ฟเวอร์ ในครั้งแรก พร็อกซีเซิร์ฟเวอร์ จะทำการตรวจสอบว่ามีข้อมูลของเว็บไซต์นั้นมืออยู่หรือไม่ หากพบว่าไม่มีข้อมูล พร็อกซีเซิร์ฟเวอร์ จะทำการเรียกข้อมูลนั้นจากเว็บไซต์แล้วเก็บไว้ในเครื่อง และเมื่อมีผู้ใช้บริการทำการเรียกเว็บไซต์นี้อีกครั้ง พร็อกซีเซิร์ฟเวอร์จะทำการส่งข้อมูลไปยังเครื่องของผู้ใช้บริการทันที ในกรณีที่เว็บไซต์มีการอัปเดตข้อมูล พร็อกซีเซิร์ฟเวอร์จะทำการตรวจสอบข้อมูลที่มีอยู่ว่าอัปเดตหรือไม่ และจะทำการอัปเดตข้อมูลใหม่ทันที ในกรณีที่มิผู้เรียกใช้บริการก็จะได้ข้อมูลที่อัปเดตอยู่เสมอ

2.2.3 การจัดการแคชของพร็อกซีเซิร์ฟเวอร์

ในการจัดการแคชของ พร็อกซีเซิร์ฟเวอร์ จะมีอยู่ 2 แบบ คือ

2.2.3.1 Passive Caching

จะเป็นการจัดการแคชแบบทั่วไป คือจะบันทึกข้อมูลลงในแคชของพร็อกซี เฉพาะเมื่อไคลเอนต์มีการขอใช้ข้อมูลผ่านทางพร็อกซีเซิร์ฟเวอร์เท่านั้น

2.2.3.2 Active Caching

จะใช้เทคนิคที่สูงกว่า โดย พร็อกซีเซิร์ฟเวอร์ จะพิจารณาเองว่าควรจะเก็บข้อมูลจากเว็บเซิร์ฟเวอร์ใด และเก็บไว้ในแคชเมื่อใด โดยไม่จำเป็นต้องรอให้ไคลเอนต์ขอใช้ข้อมูลเข้ามาก่อน

2.2.4 ประโยชน์ของการใช้พร็อกซีเซิร์ฟเวอร์

ผู้ใช้บริการสามารถเรียกดูข้อมูลจากเว็บไซต์ต่าง ๆ ได้รวดเร็วยิ่งขึ้น และช่วยประหยัดเวลาในการใช้งานอินเทอร์เน็ต ทั้งนี้ก็เพราะพร็อกซีเซิร์ฟเวอร์ ก็จะสามารถใช้ข้อมูลที่เก็บไว้จากการร้องขอของผู้ใช้รายแรกมาส่งให้แก่ผู้ใช้รายอื่นๆ ได้เลยโดยไม่จำเป็นต้องทำการร้องขอไปยังเว็บเซิร์ฟเวอร์อีกครั้ง ทำให้สามารถประหยัดได้ทั้งเวลาและแบนด์วิธของเครือข่าย นอกจากนี้ ยังได้ข้อมูลที่มีความถูกต้องและทันสมัยอยู่ตลอดเวลา

นอกจากนี้พร็อกซียังเกี่ยวข้องกับเรื่องของความปลอดภัยของผู้ใช้ด้วย ตามปกติแล้วถ้าผู้ใช้งานที่มีความรู้พื้นฐานเกี่ยวกับอินเทอร์เน็ตบ้าง ก็คงจะเคยได้ยินหรือรู้จักคำว่าไอพีแอดเดรส มาบ้าง คอมพิวเตอร์ที่อยู่บนเน็ตเวิร์ค (ที่ใช้โปรโตคอล TCP/IP) หรือบนอินเทอร์เน็ตจะมีไอพีแอดเดรสเอาไว้เพื่อเป็นการบอกที่อยู่ของเครื่องคอมพิวเตอร์แต่ละเครื่องบนเน็ตเวิร์คนั้นๆ หรือบนอินเทอร์เน็ต ในบางครั้งเวลาที่เราเปิดเว็บเบราว์เซอร์แล้วไปยังเว็บไซต์ต่าง ๆ นั้น พวกเว็บไซต์เหล่านี้จะเก็บข้อมูลหลายๆอย่างของผู้ที่เข้ามาเยี่ยมชมเว็บไซต์ โดยส่วนมากเก็บ log เป็น ไฟล์ไว้

ด้วยความสามารถของตัวเว็บเซิร์ฟเวอร์เองที่มีอยู่แล้วหรือ Script ต่างๆ ข้อมูลหลายๆอย่างที่บุคคล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มานั้น หนึ่งในนั้นก็จะมีไอพีแอดเดรสรวมอยู่ด้วย ในบางครั้งตัวพรีอกซีเซิร์ฟเวอร์สามารถช่วยปิดบัง ข้อมูลต่างๆรวมทั้งไอพีแอดเดรสของเรา เช่น การเรียกดูเว็บเพจจะเป็นพรีอกซีเซิร์ฟเวอร์เองที่ไปเรียกดูจากเว็บไซต์นั้นๆ แทนที่จะเป็นเครื่องคอมพิวเตอร์ของเราที่ไปเรียกดูโดยตรง เว็บไซต์เหล่านั้นจึงได้แต่ข้อมูลของพรีอกซีเซิร์ฟเวอร์ไปแทนที่จะได้จากเครื่องคอมพิวเตอร์ของผู้ใช้งานที่เรียกดู

อีกประการหนึ่งพรีอกซียังมีคุณสมบัติในด้านการจำกัดสิทธิ์ที่จะเข้าถึงเว็บไซต์บางแห่งที่มีเนื้อหาไม่สมควรเข้าชม การจำกัดผู้ใช้ใช้งานในเวลาที่นอกเหนือจากเวลางาน หรือข้อจำกัดอื่น ๆ ที่ทำให้สิ้นเปลืองค่าใช้จ่ายไปโดยไม่ก่อให้เกิดประโยชน์ คุณสมบัติเช่นนี้ถ้าพึงอินเตอร์เน็ตเดสทอปเว็บบอกยังไม่เพียงพอ จึงจะต้องอาศัยพรีอกซีเซิร์ฟเวอร์เข้ามาช่วยเสริมอีกแรงหนึ่ง ซึ่งภายในลินุกซ์ทุกดิสทริบิวชันจะมีโปรแกรมพรีอกซีเซิร์ฟเวอร์ที่มีประสิทธิภาพสูงให้มาพร้อมแล้ว คือ โปรแกรม Squid

โปรแกรม Squid เป็น พรีอกซีเซิร์ฟเวอร์ ที่มีคุณสมบัติในการจำกัด ควบคุมการแอกเซสเข้าสู่เว็บไซต์ภายนอกองค์กรได้เป็นอย่างดีและมีประสิทธิภาพ ที่เรียกว่า Access Control List (ACL) ซึ่งเป็นการนิยามชื่อลิสต์ขึ้นแทนคุณสมบัติของสิ่งที่ต้องการอ้างอิง จากนั้นจึงตั้งข้อกำหนดลงไปว่าต้องการให้ลิสต์นั้นสามารถแอกเซสผ่านพรีอกซีได้หรือไม่ ดังนั้นการที่เสริมการทำงานของอินเตอร์เน็ตเซิร์ฟเวอร์ด้วย Squid พรีอกซีเซิร์ฟเวอร์ จึงเป็นการควบคุมการเข้าสู่อินเตอร์เน็ตของผู้ใช้งานในองค์กรได้ตามต้องการ และยังช่วยเพิ่มประสิทธิภาพให้แก่ระบบอีกด้วยเพราะ Squid จะมีคุณสมบัติเป็น HTTP Object cache ที่ช่วยเก็บข้อมูลจากเว็บไซต์ภายนอกไว้ในหน่วยความจำ (RAM และฮาร์ดดิสก์) ของตัวเซิร์ฟเวอร์เองอีกด้วย ช่วยให้การเรียกเว็บไซต์ที่เคยเข้าถึงมาก่อนทำได้รวดเร็วยิ่งขึ้น เนื่องจากมีข้อมูลบางส่วนของเว็บเพจที่ ยังคงอยู่ในแคชนั่นเอง

2.2.5 Transparent Proxy คืออะไร

พรีอกซีเซิร์ฟเวอร์นั้นผู้ใช้งานทั่วไปจำเป็นต้องเข้าไปเซตค่าพรีอกซีเซิร์ฟเวอร์ ที่ตัวบราวเซอร์ก่อน ซึ่งอาจจะมิใช่ผู้ใช้งานที่ไม่ได้ทำตามนั้น ส่วน Transparent Proxy นั้นผู้ใช้งานไม่จำเป็นต้องเซตค่าอะไรในเครื่องคอมพิวเตอร์ของตัวเอง ตัวของ Transparent Proxy นั้นจะใช้หลักของ iptables ในการ redirects ผู้ที่ติดต่อผ่านมาทางพอร์ต http ให้ไปยัง พรีอกซีเซิร์ฟเวอร์ได้เองโดยอัตโนมัติ

2.3 โปรแกรม Squid

บนระบบปฏิบัติการลินุกซ์ มีโปรแกรมพร็อกซีเซิร์ฟเวอร์ที่มีประสิทธิภาพสูงมาก คือ โปรแกรม Squid ซึ่งเป็นโปรแกรมประเภท Proxy Caching Server สำหรับการให้บริการ Web Caching Service คือ จะคอยรับคำร้องขอบริการจากเครื่องลูกข่าย และส่งผ่านไปยังเซิร์ฟเวอร์ปลายทางที่เหมาะสม ข้อมูลต่าง ๆ ที่ผ่านเข้ามาจะถูกสำเนาเก็บไว้ในหน่วยความจำแคช และดิสก์ ดังนั้นเมื่อมีการร้องขอข้อมูลซ้ำอีกในครั้งต่อมาจะสามารถนำข้อมูลในแคชมาให้บริการได้รวดเร็วกว่าการติดต่อไปยังเซิร์ฟเวอร์โดยตรง ช่วยให้การใช้ช่องทางสื่อสารข้อมูลลงได้

2.3.1 Access Log ของโปรแกรม squid

เมื่อเครื่องลูกข่ายเรียกใช้งานเว็บไซต์ภายนอก จะทำการติดต่อผ่านพอร์ต 8080 มายังเครื่องเซิร์ฟเวอร์ลินุกซ์ที่รันโปรแกรม Squid ไว้ เราสามารถมอนิเตอร์ดูความเคลื่อนไหวของการเรียกใช้งานดังกล่าวได้จาก LOG ไฟล์ของ squid ด้วยคำสั่ง

```
tail -f /var/log/squid/access.log (แล้วแต่จะกำหนดตำแหน่งของ LOG ไฟล์)
```

จะเห็นข้อความแสดงเหตุการณ์ต่าง ๆ ที่เกิดขึ้นเกี่ยวกับโปรแกรม Squid ได้ดังรูป ซึ่งเราสามารถตีความหมายได้ เช่น

TCP_MISS หมายถึง การร้องขอนั้นไม่มีการแคชข้อมูลไว้ กรณีนี้จะต้องไป GET มาจากเว็บไซต์ปลายทางมาจริง

TCP_HIT หรือ TCP_MEM_HIT หมายถึง มีข้อมูลที่เครื่องลูกข่ายร้องขอมาอยู่แล้วในแคช กรณีนี้จะเอาข้อมูลที่อยู่ในแคชส่งกลับไปให้ผู้ร้องขอ

```
1025162259.349 6 192.168.0.254 TCP_MEM_HIT/200 440 GET
http://www.thairath.co.th/picuse/rmore_it.gif - NONE/- image/gif
1025162259.454 11 192.168.0.254 TCP_MEM_HIT/200 653 GET
http://www.thairath.co.th/picuse/foot_it.gif - NONE/- image/gif
1025162259.758 3 192.168.0.254 TCP_HIT/200 9413 GET
http://www.thairath.co.th/link/schedule.jpg - NONE/- image/jpeg
1025162259.970 17 192.168.0.254 TCP_HIT/200 8550 GET
http://www.thairath.co.th/link/IT.banner.gif - NONE/- image/gif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ access.log ของโปรแกรม Squid นี้จะมีขนาดเพิ่มขึ้นเรื่อย ๆ จนอาจทำให้เนื้อที่ดิสก์เต็ม และสร้างปัญหาให้กับระบบได้ ดังนั้นจึงควรใช้กำหนดให้ Squid ทำการเปลี่ยนไฟล์ Log ด้วยคำสั่ง

```
squid -k rotate
```

โดยอาจจะเขียนเป็นเชลล์สคริปต์เพื่อให้ Log ไฟล์เกิดการหมุนเวียนกันไป ในแต่ละสัปดาห์ก็ได้ แต่ถ้าไม่จำเป็นต้องใช้ข้อมูลจาก Log ก็สามารถยกเลิกการเก็บ Log ไว้ได้โดยใช้สัว์เวิร์ด ดังนี้

```
cache_access_log /dev/null
```

นอกจาก access.log แล้ว ไฟล์ที่สำคัญอีกไฟล์หนึ่งก็คือ cache.log มีหน้าที่เก็บสถานะและข้อมูลเกี่ยวกับการ debug โปรแกรม Squid ซึ่งมีประโยชน์อย่างมากในการแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้น

2.3.2 Cache Dir แครชของโปรแกรม Squid

พื้นที่เก็บข้อมูลแคชบนดิสก์ของ Squid จะอยู่ที่ /var/spool/squid (อาจไม่อยู่ในตำแหน่งนี้เสมอไป ขึ้นอยู่กับการกำหนดของผู้ติดตั้ง โปรแกรม) พื้นที่นี้ควรแยกออกจากพาร์ทิชันของระบบอย่างเด็ดขาด เนื่องจากพื้นที่นี้จะมีข้อมูลอ่าน เขียนอยู่ตลอดเวลาทำให้มีโอกาสเสื่อมสภาพมากกว่าส่วนอื่นๆ หากเกิดข้อผิดพลาดขึ้นแล้วจะได้ไม่สร้างความเสียหายไปยังส่วนอื่นๆ ที่สำคัญต่อระบบด้วย นอกจากนี้หากพิจารณาเรื่องความเร็ว และประสิทธิภาพแล้ว ควรสร้างพื้นที่นี้ในฮาร์ดดิสก์ที่มีความเร็วสูง และหากเป็นไปได้ควรใช้ฮาร์ดดิสก์ชนิด SCSI จะเหมาะสมกว่าชนิด IDE

หากเราต้องการตรวจสอบว่า พื้นที่ที่เราใช้งานเป็นดิสก์แคชมีการใช้เนื้อที่ไปแล้วมากน้อยแค่ไหน สามารถตรวจสอบได้จากคำสั่ง

```
du -sh /var/spool/squid
```

หรือกรณีที่เป็นแยกพาร์ทิชันนี้ออกมา ดังที่กล่าวไว้ข้างต้น สามารถตรวจสอบได้ด้วยคำสั่ง df -h อีกคำสั่งหนึ่ง

2.3.3 โปรแกรมยูทิลิตี้ของโปรแกรม Squid

โปรแกรม cachemgr.cgi เป็นยูทิลิตี้ของ Squid ที่ช่วยในด้านการรายงานสถานะ ค่าสถิติต่าง ๆ เกี่ยวกับคอนฟิกและประสิทธิภาพของโปรแกรม Squid โดยจะมีติดตั้งมาพร้อมกันกับ Squid แล้ว โปรแกรมนี้ถูกออกแบบให้ใช้งานผ่านเว็บ (Web Interface)

2.4 โปรแกรม SquidGuard

โปรแกรม SquidGuard คือ โปรแกรมที่เป็น plugin สำหรับโปรแกรม Squid ซึ่งมี ความสามารถของการทำ filter, redirect และการทำ access controller โดยโปรแกรม SquidGuard นั้น เป็นโปรแกรมฟรีแวร์ที่ทำงานบนระบบปฏิบัติการ UNIX

2.4.1 ความสามารถโปรแกรม SquidGuard

จำกัดการเข้าเว็บไซต์สำหรับผู้ใช้งานบางคน โดยพิจารณาจากไอพีแอดเดรส และ/หรือ URLs

- บล็อกการเข้าเว็บไซต์ที่อยู่ใน list หรือ blacklist
- บล็อกการเข้า URLs ที่ตรงกับ list ของ regular expression หรือ words
- Redirect จากเว็บไซต์ที่บล็อกไปยังหน้าเว็บไซต์ที่ต้องการได้

2.5 เว็บเซอร์วิส

2.5.1 เว็บเซอร์วิสคืออะไร

เว็บเซอร์วิส คือ แอปพลิเคชันหรือโปรแกรมที่ทำงานอย่างใดอย่างหนึ่ง ในลักษณะ ให้บริการ โดยจะถูกเรียกใช้งานจากแอปพลิเคชันอื่นๆ ในรูปแบบ RPC(Remote Procedure Call) ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เป็นตัวในการแลกเปลี่ยนคือ XML ทำให้เราสามารถเรียกใช้ส่วนประกอบใดๆ ก็ได้ ในระบบหรือ platform ใดๆ ก็ได้ บน โพรโตคอล HTTP ซึ่งเป็น โพรโตคอลสำหรับอินเทอร์เน็ต อันเป็นช่องทางที่ได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกันระหว่างแอปพลิเคชันกับแอปพลิเคชันในปัจจุบัน

2.5.2 เว็บแอปพลิเคชันคืออะไร

เว็บแอปพลิเคชัน (Web application) คือ โปรแกรมที่อยู่ในเว็บเซิร์ฟเวอร์ที่คอยให้บริการ สิ่งที่ต้องการ (request) จากทางไคลเอนต์ผ่าน โพรโตคอล HTTP ซึ่งจะแสดงผลที่ต้องการในรูปแบบของ HTML page ผ่านทางเบราว์เซอร์ ซึ่งก็คือเว็บไซต์ต่างๆ ที่เราใช้บริการอยู่นั่นเอง

เว็บแอปพลิเคชันสามารถตอบสนองความคิด Distributed Processing ได้ในระดับหนึ่ง ซึ่งก็คือ การแบ่งการประมวลผลไว้ที่ฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ และมักจะมีการใช้ฐานข้อมูล (database) ควบคู่กับการทำเว็บแอปพลิเคชันไปด้วยตามความต้องการในการทำ E-Business และ E-Commerce ที่กำลังเป็นที่นิยมในปัจจุบัน แต่ปัญหาที่ตามมาคือ เรื่องของการจ่ายเงินหรือที่เรียกว่า E-payment หรือ Payment-Gateway ซึ่งเว็บแอปพลิเคชันที่ทำ E-Commerce ต้องใช้ บริการจากธนาคารออนไลน์ ในการจัดเก็บเงินกับลูกค้าด้วยเทคโนโลยีนี้ การใช้บริการเก็บเงิน จากธนาคารออนไลน์ จึงเป็นสิ่งจำเป็นที่ผู้ให้บริการจะต้องดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Packet Filtering

การทำ Packet Filtering เป็นการดักจับแพ็คเกจต่างๆ ภายในเครือข่าย โดยมีเครื่องมือต่างๆ แต่ที่เป็นที่นิยมใช้กันได้แก่ โปรแกรม Tcpdump ซึ่งเป็นเครื่องมือด้านเครือข่ายที่ทำหน้าที่เป็นตัวดักจับแพ็คเกจ โดยอาศัยการทำงานร่วมกับไลบรารี PCAP โดยจะทำงานร่วมกับ Network Interface ในโหมด Promiscuous คล้ายกับเป็นผู้ดักจับหรือแอบฟังความเคลื่อนไหวในระบบเครือข่าย tcpdump แต่ในโครงการนี้การดักจับแพ็คเกจส่วนใหญ่เป็นการดักจับโดยการกรองส่วน Header ของ HTTP Packet จึงใช้สคริปต์ภาษา Ruby เป็นตัวดักจับแพ็คเกจแทน ซึ่งการทำงานของสคริปต์นี้อาศัยการทำงานร่วมกับไลบรารี PCAP ของ Ruby เช่นกัน

2.7 ไฟร์วอลล์

ปัจจุบันอินเทอร์เน็ตมีบทบาทสำคัญต่อการดำเนินกิจกรรมต่างๆ เป็นอย่างมาก ไม่ว่าจะเป็นด้านการติดต่อสื่อสาร ธุรกิจ การศึกษา หรือว่าเพื่อความบันเทิง องค์กรต่างๆ ทั้งภาครัฐและเอกชน ต่างก็นำเอาระบบเครือข่ายของตนเชื่อมต่อเข้ากับอินเทอร์เน็ตเพื่อที่จะได้รับประโยชน์เหล่านี้ แต่การนำเอาระบบเครือข่ายไปเชื่อมต่อกับอินเทอร์เน็ตนั้น ทำให้เกิดปัญหาตามมาในเรื่องความปลอดภัยของระบบเครือข่าย เช่น ทำให้เกิดความเสี่ยงต่อการถูกเจาะระบบ และ ขโมยข้อมูล เป็นต้น จากปัญหาดังกล่าวทำให้ต้องมีวิธีการในการรักษาความปลอดภัย สิ่งที่สามารถช่วยลดความเสี่ยงนี้ได้ก็คือ ไฟร์วอลล์ โดยไฟร์วอลล์นั้นจะทำหน้าที่ป้องกันอันตรายต่างๆ จากภายนอกที่จะเข้ามาในระบบเครือข่ายของเรา

2.7.1 ชนิดของไฟร์วอลล์

ชนิดของไฟร์วอลล์แบ่งตามเทคโนโลยีที่ใช้ในการตรวจสอบและควบคุม แบ่งได้เป็น

- Packet Filtering คือเราเตอร์ที่ทำกรหาเส้นทางและส่งต่ออย่างมีเงื่อนไข โดยจะพิจารณาจากข้อมูลส่วนที่อยู่ในส่วนหัวหรือ Header ของแพ็คเกจที่ผ่านเข้ามาเทียบกับกฎที่กำหนดไว้และตัดสินใจว่าจะทิ้งแพ็คเกจนั้นไปหรือว่าจะยอมให้แพ็คเกจนั้นผ่านไป
- Proxy Service หรือ Application Gateway เป็นแอปพลิเคชันโปรแกรมที่ทำงานอยู่บนไฟร์วอลล์ที่ตั้งอยู่ระหว่างเครือข่าย 2 เครือข่าย ทำหน้าที่เพิ่มความปลอดภัยของระบบเครือข่ายโดยการควบคุมการเชื่อมต่อระหว่างเครือข่ายภายในและภายนอก โดย Proxy จะช่วยเพิ่มความปลอดภัยได้มากเนื่องจากการตรวจสอบข้อมูลถึงในระดับของแอปพลิเคชันเลเยอร์ (Application Layer)
- Stateful Inspection เป็นเทคโนโลยีที่เพิ่มเข้าไปใน Packet Filtering โดยในการพิจารณาว่าจะยอมให้แพ็คเกจผ่านไปนั้น แทนที่จะดูข้อมูลจากส่วนหัวเพียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างเคียว Stateful Inspection จะนำเอาส่วนข้อมูลของแพ็กเก็ต และข้อมูลที่ได้ จากแพ็กเก็ตก่อนหน้านี้ที่ได้ทำการบันทึกเอาไว้ นำมาพิจารณาด้วย จึงทำให้ สามารถระบุได้ว่าแพ็กเก็ตใดเป็นแพ็กเก็ตที่ติดต่อเข้ามาใหม่ หรือว่าเป็นส่วน หนึ่งของการเชื่อมต่อที่มีอยู่แล้ว โดยในโครงการนี้มีการใช้ไฟร์วอลล์ประเภท Stateful Inspection คือ Iptables นั่นเอง

2.7.2 Iptables

ในโครงการนี้มี การนำ Iptables มาใช้ในขั้นตอนของการบล็อกผู้ที่ต้องการทำการขอ ความ์โหลดไฟล์ข้อมูลจากเครื่องภายในระบบ โดยรายละเอียดทั้งหมดเกี่ยวกับ Iptables นั้นมาอยู่ หลายส่วน แต่จะกล่าวถึงเฉพาะส่วนที่นำมาใช้ในโครงการนี้คือในส่วนของการบล็อกไอพี แอดเดรสและพอร์ตที่เราต้องการ

iptables จะมีรูปแบบการใช้งานดังนี้คือ

```
iptables [table] <command> <match> <target/jump>
```

- **[table]** หมายถึง ตาราง ที่ต้องการระบุ เช่น iptables -t nat หมายถึงให้ทำงานกับ nat table ในกรณีที่ไม่ได้ระบุตาราง iptables จะถือว่าคำสั่งดังกล่าวระบุถึง filter table โดยอัตโนมัติ
- **<command>** จะเป็นตัวสั่งให้ iptables ทำในสิ่งที่ต้องการ เช่น iptables -A INPUT ซึ่งหมายถึงให้สร้าง rule ต่อท้าย INPUT chain ใน filter table
- **<match>** เป็นส่วนที่ใช้ตรวจสอบว่า packet มีข้อมูลตรงกับที่ระบุไว้หรือไม่ เช่น มีไอพีแอดเดรสเป็น 1.2.3.4
- **<target/jump>** เป็นตัวระบุว่าจะเมื่อเจอ packet ที่ตรงกับที่ระบุไว้ก็จะกระทำ ตามที่ระบุไว้ เช่น ถ้าแพ็กเก็ตใดมีไอพีแอดเดรสต้นทางเป็น 1.2.3.4 ให้ทิ้งแพ็กเก็ต นั้นไป

Table

iptables สามารถทำงานได้กับตาราง 3 ตารางหลัก สามารถระบุตารางได้โดยใช้ชื่อปชัณ - t ตามด้วยชื่อตาราง คือ

1. **Filter table** ใช้สำหรับกรอง packet มี 3 built-in chain คือ INPUT, OUTPUT, FORWARD ซึ่งจะได้อธิบายรายละเอียดต่อไป
2. **Nat table** ใช้สำหรับการแปลงแอดเดรส (Network Address Translation) มี 3 built-in chain คือ PREROUTING, POSTROUTING, OUTPUT ซึ่งรายละเอียดจะได้ธิบาย ต่อไป
3. **Mangle table** เป็นตารางที่ใช้เปลี่ยนแปลงหรือแก้ไข packet เช่น เปลี่ยนค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TTL, MARK ซึ่งปกติจะใช้ในการทำ routing ที่มีความซับซ้อนสูง มี 2 built-in chain คือ PREROUTING chain (ใช้แก้ไข packet ก่อนที่จะเข้าสู่ไฟร์วอลล์และก่อนเข้าสู่ routing decision) และ OUTPUT chain (ใช้แก้ไข packet ที่ถูกสร้างโดยไฟร์วอลล์ก่อนที่มันจะถูกส่งไปยัง routing decision) ทั้งนี้ไม่สามารถทำ network address translation หรือ masquerading ที่ table นี้ได้ และในเอกสารฉบับนี้จะไม่กล่าวถึง mangler อีก เนื่องจากเป็นส่วนที่ไม่นิยมนำไปใช้งาน

Command

จะเป็นตัวสั่งให้ iptables ทำในสิ่งที่ต้องการซึ่งมีคำสั่งและออพชั่นต่างๆมากมาย โดยจะกล่าวถึงเฉพาะในส่วนที่ใช้ในโครงงานนี้ดังนี้

- -A เพิ่ม rule ใหม่ต่อท้าย chain (Append rule)
- -D ลบ rule (Delete rule)
- -I เพิ่ม rule ใหม่ ใน chain (Insert rule)
- -R แทนที่ rule เดิม ด้วย rule ใหม่ (Replace rule)
- -L แสดง rule ทั้งหมดใน chain (ถ้าไม่ระบุ chain จะแสดง rule ทั้งหมดใน filter table ทั้งสาม built-in chain)
- -F ลบ rule ทั้งหมดใน chain ทิ้ง
- -Z ใช้ reset byte counter สำหรับทุก rule ใน chain ที่กำหนด
- -N ใช้สร้าง chain ใหม่
- -E ใช้เปลี่ยนชื่อ chain ใหม่

นอกจากนี้การใช้คำสั่งต่างๆยังสามารถใช้ร่วมกับออพชั่นต่างๆได้อีกด้วย ดังนี้

- -V, --verbose ใช้ร่วมกับ -L, -A, -I, -D, -R เพื่อให้แสดงจำนวน byte ที่ match กับ rule ออกมาด้วย
- -x, --exact ใช้ร่วมกับ -L และ -v เพื่อให้แสดงจำนวน packet และจำนวนของ byte ข้อมูลที่ match โดยไม่แสดงผลในหน่วยของ K,M,G
- -n, --numeric ใช้ร่วมกับ -L เพื่อสั่งให้ iptables แสดงข้อมูลไอพีแอดเดรสและ port เป็นตัวเลขเท่านั้น
- --line-numbers ใช้ร่วมกับ -L เพื่อแสดงเลขบรรทัดของ rule ซึ่งตัวเลขที่แสดงนี้ จะสามารถใช้ได้กับคำสั่ง insert rule ที่ระบุเป็นลำดับที่ของ rule
- --modprobe=command เพื่อโหลด module ที่เกี่ยวข้อง

Match

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การระบุ source, destination IP address
สามารถระบุไอพีแอดเดรสต้นทางของแพ็คเก็ตโดยใช้ -s หรือ --source หรือ --src และสำหรับไอพีแอดเดรสปลายทางก็ใช้ -d หรือ --destination หรือ --dst การระบุไอพีแอดเดรสนั้นสามารถทำได้ 4 แบบด้วยกันคือ
 1. ใช้ชื่อเต็มแทน เช่น localhost หรือ www.nectec.or.th
 2. ระบุไอพีแอดเดรสโดยตรง เช่น 127.0.0.1 หรือ 202.44.204.33
 3. ระบุเป็นกลุ่มของไอพีแอดเดรส เช่น 202.44.204.0/24 ซึ่งหมายถึงไอพีแอดเดรสตั้งแต่ 202.44.204.0 - 202.44.204.255
 4. หรืออาจจะใช้ 202.44.204.0/255.255.255.0 แทน 202.44.204.0/24 ได้
- การระบุโปรโตคอล
สามารถระบุโปรโตคอลที่ต้องการได้ดังนี้คือ TCP, UDP, ICMP หรือสามารถใช้ตัวเลขแทนได้ และยังสามารถใช้ได้ทั้งตัวอักษรเล็กหรือใหญ่ เช่น -p TCP หรือ -p ! tcp

การระบุ target

เมื่อมีแพ็คเก็ตที่ตรงกับกฎแล้ว ต้องกำหนดเป้าหมายสำหรับแพ็คเก็ตนั้นไว้ด้วย โดยปกติจะใช้กัน 2 ลักษณะคือ DROP และ ACCEPT

บทที่ 3

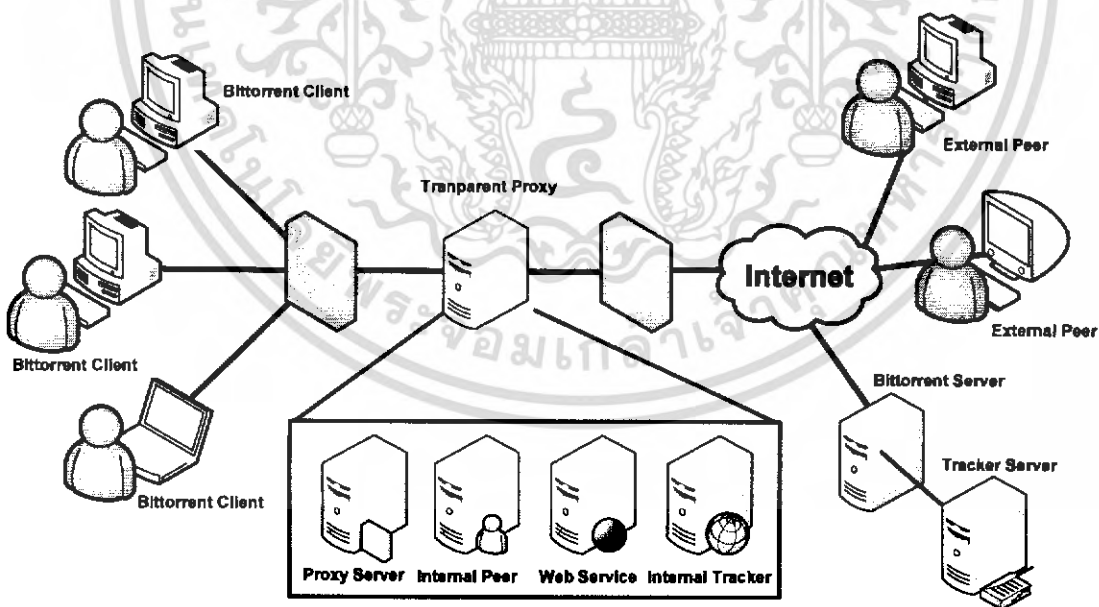
การออกแบบและพัฒนาระบบแคชสำหรับโปรแกรมบิตทอเรนท์

ในการออกแบบระบบแคชสำหรับโปรแกรมบิตทอเรนท์นั้น จะต้องคำนึงถึงหลักของ โพรโตคอลบิตทอเรนท์ หลักของพรีอ็อกซีเซิร์ฟเวอร์ หลักการของเว็บเซอร์วิส รวมไปถึงกระบวนการทำงานของตัวโปรแกรมบิตทอเรนท์ที่ใช้ในการดาวน์โหลดและอัปโหลดไฟล์

บทนี้จะกล่าวถึงการออกแบบและพัฒนาระบบแคชสำหรับโปรแกรมบิตทอเรนท์ โดยจะบอกถึงแบบจำลองของระบบ องค์ประกอบต่างๆที่สำคัญของระบบ รวมไปถึงขั้นตอนการทำงานของระบบแคชสำหรับโปรแกรมบิตทอเรนท์

3.1 แบบจำลองของระบบแคชสำหรับโปรแกรมบิตทอเรนท์

จากความต้องการของโครงการระบบแคชสำหรับโปรแกรมบิตทอเรนท์นี้ คือ การลดการเชื่อมต่อจากเครือข่ายภายในที่มีการใช้โปรแกรมบิตทอเรนท์ออกไปยังภายนอกเครือข่าย ดังนั้น จำเป็นที่จะต้องจำลองระบบการทำงานของบิตทอเรนท์ขึ้นมาภายในเครือข่าย และจะต้องทำให้เมื่อเครื่องภายในเครือข่ายต้องการใช้งาน โปรแกรมบิตทอเรนท์จะต้องผ่านระบบแคชเสมอ ในส่วนนี้ จะต้องนำหลักการของพรีอ็อกซีเซิร์ฟเวอร์มาใช้งาน ซึ่งจากการวิเคราะห์แล้วจึงสร้างแบบจำลองของระบบแคชสำหรับโปรแกรมบิตทอเรนท์เป็นดังรูป 3.1



รูปที่ 3.1 แบบจำลองของระบบแคชสำหรับโปรแกรมบิตทอเรนท์

3.2 องค์ประกอบของระบบแคชสำหรับโปรแกรมบิตทอเรนท

จากแบบจำลองของระบบแคชสำหรับโปรแกรมบิตทอเรนทที่ได้ออกแบบมานั้น สามารถแบ่งองค์ประกอบของระบบออกเป็น 4 ส่วนได้ดังนี้

3.2.1 Proxy Server

Proxy Server นี้ทำหน้าที่คัดกรอง URL ที่เป็นการดาวน์โหลดไฟล์ .torrent เมื่อพบแล้วจะทำการ Redirect หรือส่งต่อไปยัง Web Service ซึ่งการทำงานของ Proxy Server นี้สามารถสั่งให้ทำงานแบบ Transparent Proxy ได้ ขึ้นอยู่กับความต้องการและการติดตั้ง

Proxy Server จะทำงานโดยใช้โปรแกรม Squid ร่วมกับโปรแกรม SquidGuard บนระบบปฏิบัติการลินุกซ์ ซึ่งขั้นตอนการติดตั้งสำหรับการทำงานเป็นระบบแคชสำหรับโปรแกรมบิตทอเรนทนั้นจะอยู่ในคู่มือการติดตั้ง

3.2.2 Web Service

Web Service สามารถแบ่งการทำงานได้เป็น 2 ส่วนใหญ่ๆคือกระบวนการดาวน์โหลดและกระบวนการอัปโหลดดังนี้

- กระบวนการดาวน์โหลดมีฟังก์ชันการทำงานดังนี้
 1. ดาวน์โหลดไฟล์ .torrent ที่ Client ต้องการจะดาวน์โหลด โดยทำการส่งแพ็คเกจไปยัง Request URL ที่โปรแกรม SquidGuard ส่งมาให้โดยในแพ็คเกจดังกล่าวต้องมี Cookie ของ Client ซึ่งได้มาจากการดักจับแพ็คเกจด้วยโปรแกรม tcpdump
 2. แก้ไขไฟล์ .torrent ที่ดาวน์โหลดมา โดยทำการเปลี่ยนค่า announce ของไฟล์ torrent ให้เป็น URL ของ Internal Tracker เพื่อให้เวลา Client ต้องการดาวน์โหลดไฟล์จะต้องมาทำการดาวน์โหลดจาก Internal Tracker
 3. ส่งไฟล์ที่แก้ไขแล้วกลับไปยัง Client ผู้ที่ต้องการดาวน์โหลดไฟล์
 4. ทำการลงทะเบียนข้อมูลของไฟล์ .torrent ที่แก้ไขแล้วลงใน Internal Tracker
 5. สั่งให้ Internal Peer เริ่มทำการดาวน์โหลดข้อมูลของไฟล์ .torrent นั้นๆ
- กระบวนการอัปโหลดมีฟังก์ชันการทำงานดังนี้
 1. สั่งให้ Internal Peer เริ่มทำการดาวน์โหลดไฟล์ .torrent ที่ดักจับได้จากแพ็คเกจ HTTP POST

- จัดการกับรายชื่อไอพีแอดเดรสและพอร์ตที่ได้รับการตอบสนองกลับมาจาก External Tracker โดยนำค่าต่าง ๆ นั้นส่งต่อไปให้กับ Iptables เพื่อทำการบล็อกการเข้าถึงไอพีแอดเดรสและพอร์ตนั้นๆ

ฟังก์ชันทั้งหมดใน Web Service นี้เขียนขึ้นเองโดยใช้ภาษา PHP โดยในระบบแลชสำหรับโปรแกรมบิตทอเรนทนี่ จะให้ Web Service ทำงานบนระบบปฏิบัติการลินุกซ์

3.2.3 Internal Tracker

เป็นแทรคเกอร์ของระบบแลชสำหรับโปรแกรมบิตทอเรนท ทำหน้าที่ในการเก็บข้อมูลของไฟล์ .torrent และเก็บหมายเลขไอพีของ Internal Peer และหมายเลข IP ของ Client ที่ทำการดาวน์โหลดไฟล์นั้นๆไปแล้ว เมื่อ Client ใช้งานโปรแกรมบิตทอเรนทติดต่อมายัง Internal Tracker ตัว Internal Tracker ก็จะส่งหมายเลขไอพีของ Internal Peer และ Peer อื่นๆที่มีไฟล์นั้นๆกลับไปยัง Client ที่ติดต่อมา

Internal Tracker ที่ใช้ในระบบแลชสำหรับโปรแกรมบิตทอเรนทนี่ ใช้ Torrentpier ซึ่งเป็นแทรคเกอร์แบบ Forum โดยใช้ร่วมกับ phpBB2 สาเหตุที่เลือกใช้แทรคเกอร์ตัวนี้ก็เพราะมีการกำหนดไม่ให้ใช้งาน passkey ได้

3.2.4 Internal Peer

Internal Peer คือโปรแกรม Bittorrent Client ที่ทำหน้าที่ดาวน์โหลดไฟล์ข้อมูลของไฟล์ .torrent ต่างๆ สามารถแบ่งกระบวนการทำงานออกเป็น 2 กระบวนการใหญ่ๆ คือ

- กระบวนการดาวน์โหลด

Internal Peer จะทำหน้าที่ดาวน์โหลดไฟล์ข้อมูลจากเครือข่ายภายนอก และทำหน้าที่เป็น Seeder ให้กับเครื่อง Client ภายในเครือข่ายที่ใช้งานโปรแกรมบิตทอเรนท

- กระบวนการอัปโหลด

Internal Peer จะทำหน้าที่ดาวน์โหลดไฟล์ข้อมูลจากเครื่องภายในระบบที่ต้องการทำการอัปโหลด เพื่อทำหน้าที่เป็น Seeder ให้เครื่องภายนอกระบบ

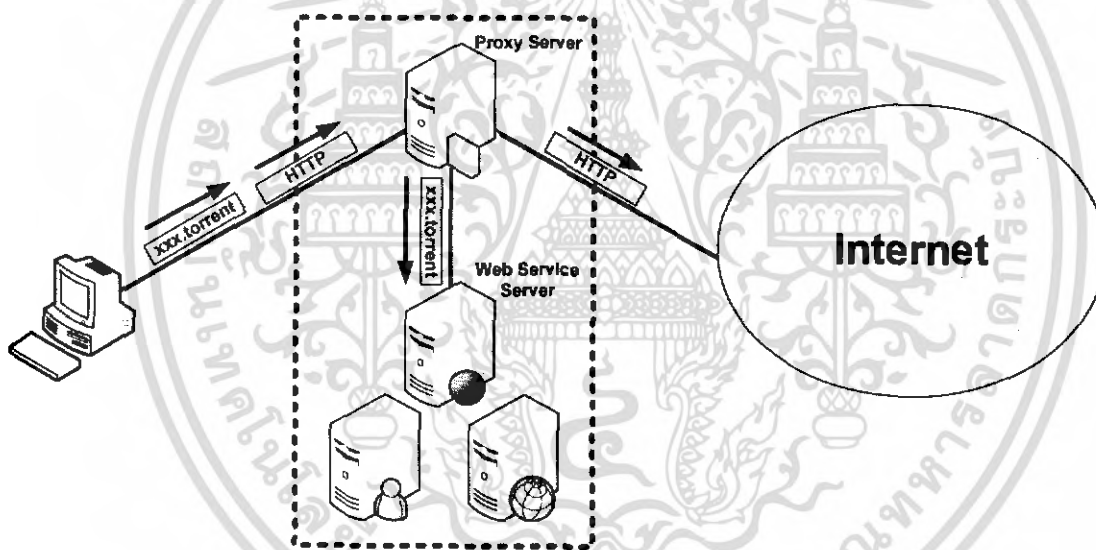
Internal Peer ที่ใช้ในระบบแลชสำหรับโปรแกรมบิตทอเรนทนี่ ใช้โปรแกรมบิตทอเรนท

3.3 ขั้นตอนการทำงานของระบบแคชสำหรับโปรแกรมบิตทอเรนท์

หลังจากที่ได้ทราบถึงแบบจำลองและองค์ประกอบต่างๆของระบบแคชสำหรับโปรแกรมบิตทอเรนท์แล้ว ในหัวข้อนี้จะกล่าวถึงขั้นตอนการทำงานของระบบแคชสำหรับโปรแกรมบิตทอเรนท์ ซึ่งมีขั้นตอนการทำงานดังนี้

3.3.1 กระบวนการดาวน์โหลด

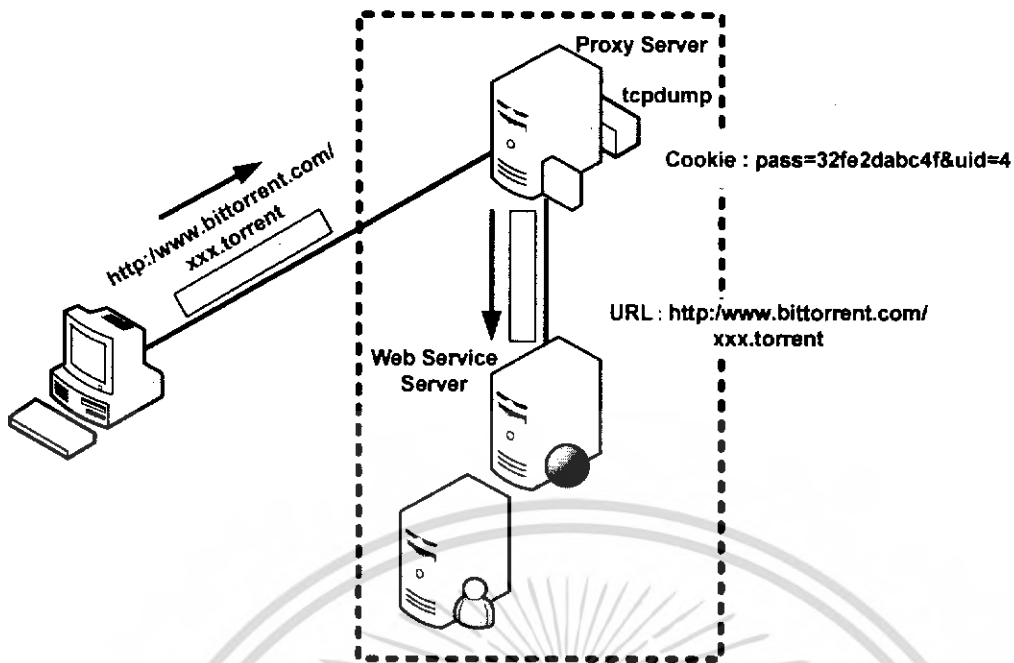
1. ในระบบนี้เครื่องที่อยู่ภายในระบบเมื่อต้องการจะใช้งานอินเทอร์เน็ตจะต้องผ่าน Proxy Server ก่อน ดังนั้นเมื่อ Client ต้องการที่จะดาวน์โหลดไฟล์นามสกุล .torrent จากอินเทอร์เน็ต จะถูก Proxy Server ทำการตรวจสอบ URL ว่าเป็นการเรียกไฟล์ .torrent หรือไม่ ถ้าเป็นการเรียกไฟล์ .torrent จะ redirect URL นั้นไปยัง Web Service แต่ถ้าไม่ใช่การเรียกไฟล์ .torrent ก็จะปล่อยไป ซึ่งแสดงดังรูปที่ 3.2



รูปที่ 3.2 แสดงการทำงานการดาวน์โหลดไฟล์ในขั้นตอนแรก

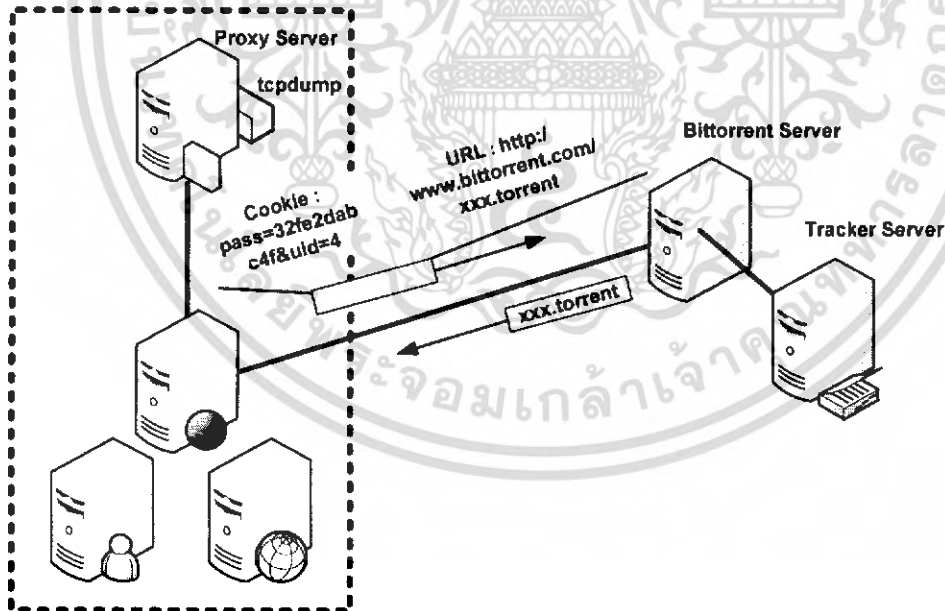
2. เมื่อ Proxy Server ตรวจสอบแล้วว่าเป็นการเรียกไฟล์ .torrent ก็จะส่ง URL ไปยัง Web Service และในขณะที่เดียวกัน tcpdump ก็จะทำการจับแพ็คเก็ตที่ถูกส่งออกมาจาก Client ในขณะนั้นเพื่อดักจับ Cookie ของ Client ดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงการทำงานการดาวน์โหลดไฟล์ในขั้นตอนที่สอง

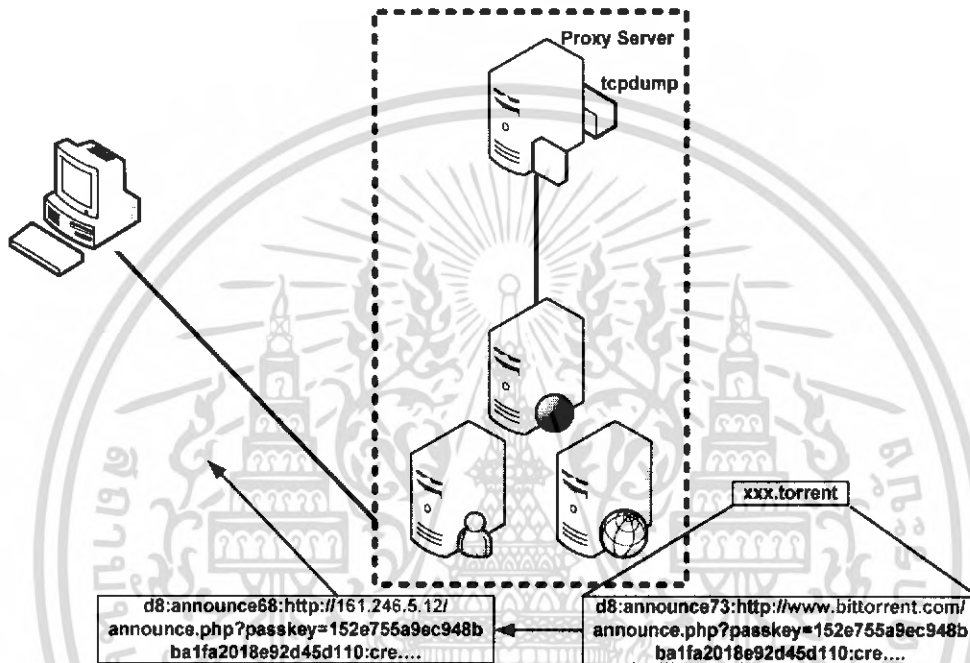
3. Web Service เมื่อได้รับ URL และ Cookie มาแล้ว จะทำการส่งแพ็คเกจที่สร้างขึ้นใหม่ออกไปเพื่อดาวน์โหลดไฟล์ .torrent นั้นๆมา ดังรูปที่ 3.4



รูปที่ 3.4 แสดงการทำงานการดาวน์โหลดไฟล์ในขั้นตอนที่สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

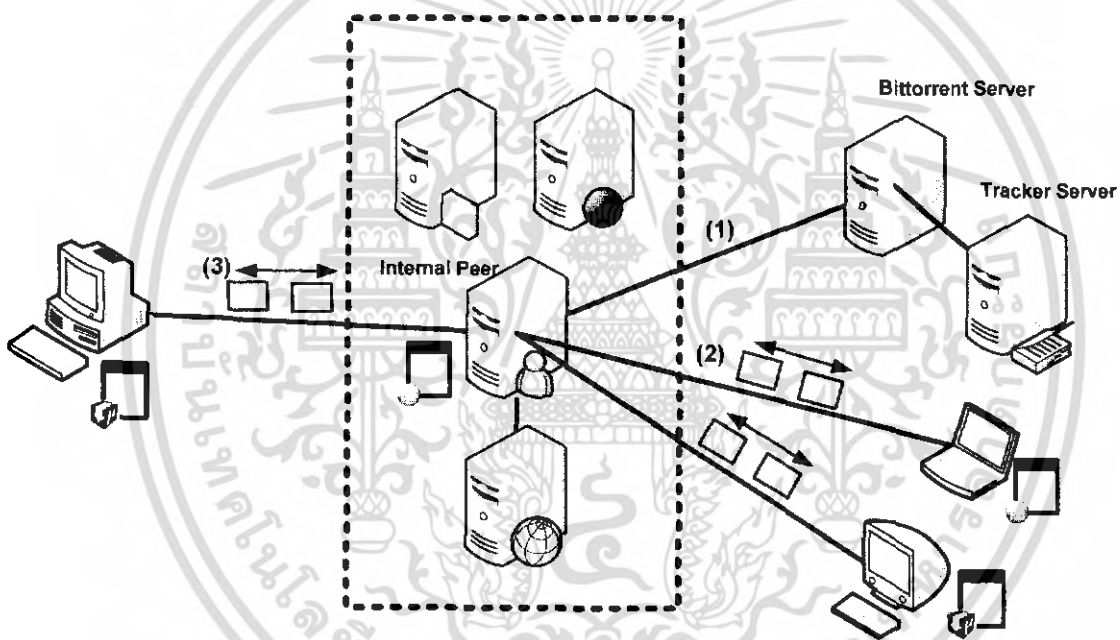
4. เมื่อดาวน์โหลดมาแล้วจะทำการตรวจสอบค่า infohash ของไฟล์ .torrent ที่ดาวน์โหลดมากับฐานข้อมูลของ Internal Tracker ว่ามีไฟล์นี้อยู่ใน Internal Tracker แล้วหรือยัง ถ้าพบว่ามีอยู่แล้วจะส่งไฟล์ที่เคยแก้ไขแล้วกลับไปยัง Client ผู้ร้องขอ ถ้าไม่พบจะทำการทำสำเนา (copy) ไฟล์ต้นฉบับไว้ แล้วทำการแก้ไขไฟล์ .torrent ที่ทำการสำเนาไว้ โดยเปลี่ยนการประกาศ URL Tracker เป็น URL ของ Internal Tracker แทน เมื่อ Web Service ทำการแก้ไขไฟล์ .torrent ที่ดาวน์โหลดมาเรียบร้อยแล้ว ก็จะส่งไฟล์ .torrent ที่แก้ไขแล้ว กลับ ไปยัง Client ที่ร้องขอมา ดังรูปที่ 3.5



รูปที่ 3.5 แสดงการทำงานการดาวน์โหลดไฟล์ในขั้นตอนที่สี่

5. จากนั้น Web Service จะติดต่อกับ Internal Tracker เพื่อลงทะเบียนไฟล์ .torrent ที่แก้ไขแล้วลงฐานข้อมูล Internal Tracker เพื่อให้ Internal Tracker มีข้อมูลสำหรับการติดต่อกับ Client เมื่อลงทะเบียนไฟล์ .torrent กับ Internal Tracker แล้ว Web Service จะเพิ่มข้อมูล seeder ให้กับฐานข้อมูลของ Internal Tracker โดยข้อมูลของ seeder ที่เพิ่มเข้าป็นั้นคือหลายเลขไอพี และพอร์ตที่ใช้ในการเชื่อมต่อและ Peer ID ของ Internal Peer นั้นเอง
6. จากนั้น Web Service จะทำการสั่งให้โปรแกรมบิตทอเรนที่ในเครื่อง Internal Peer ทำงาน โดยให้ Internal Peer ทำงานจากไฟล์ .torrent ที่เป็นไฟล์ต้นฉบับ เพื่อไปดาวน์โหลดชิ้นส่วนของไฟล์ข้อมูลจากอินเทอร์เน็ต

7. เมื่อ Internal Peer ถูกสั่งให้ทำงาน ก็จะไปติดต่อกับ External Tracker เพื่อรับข้อมูลของผู้ที่ไฟล์ที่ต้องการมา แล้วทำการเชื่อมต่อกับ External Peer เพื่อแลกเปลี่ยนชิ้นส่วนไฟล์ข้อมูลระหว่างกัน ดังรูป 3.6 ขั้นตอนที่ 1 และ 2
8. ส่วน Client นั้นเมื่อสั่งให้โปรแกรมบิตทอเรนททำงานจากไฟล์ .torrent ที่ถูกแก้ไขแล้ว ก็จะไปติดต่อกับ Internal Tracker ซึ่งเมื่อติดต่อกับ Internal Tracker ได้แล้วก็จะได้ข้อมูลของ seeder มา ซึ่งข้อมูลนี้ก็เป็นข้อมูลของตัว Internal Peer ที่ Web Service ได้ทำการเพิ่มลงฐานข้อมูลของ Internal Tracker
9. เมื่อได้ข้อมูลของ seeder มาแล้ว Client จะทำการติดต่อไปยัง Internal Peer เพื่อทำการดาวน์โหลดชิ้นส่วนของไฟล์ข้อมูลที่ต้องการมายังเครื่องของตัวเอง ดังรูปที่ 3.6 ขั้นตอนที่ 3

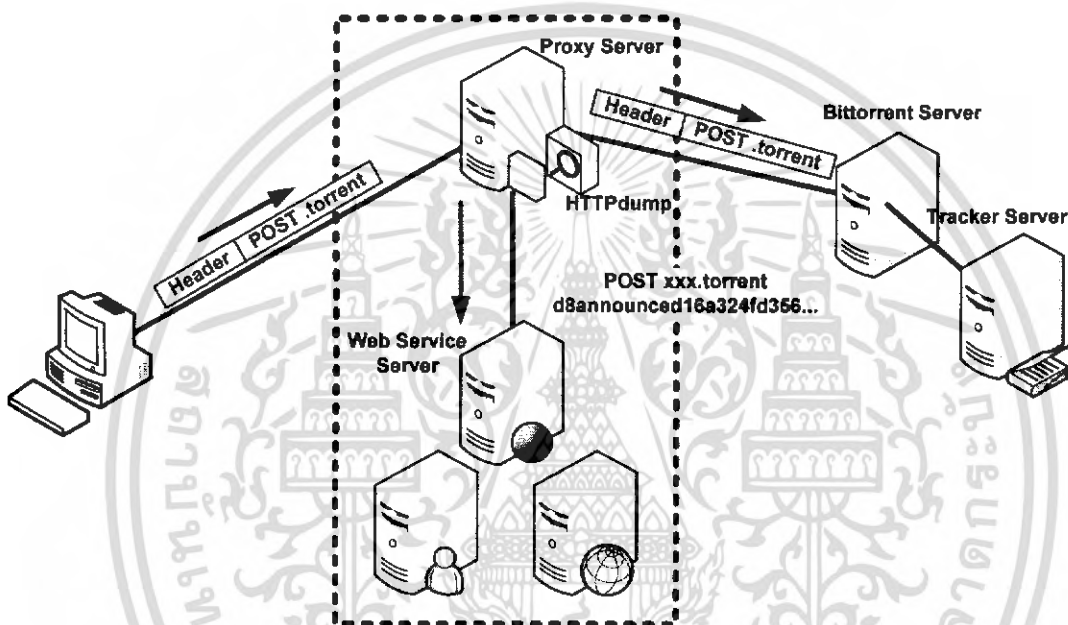


รูปที่ 3.6 แสดงการทำงานการดาวน์โหลดไฟล์ในขั้นตอนสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

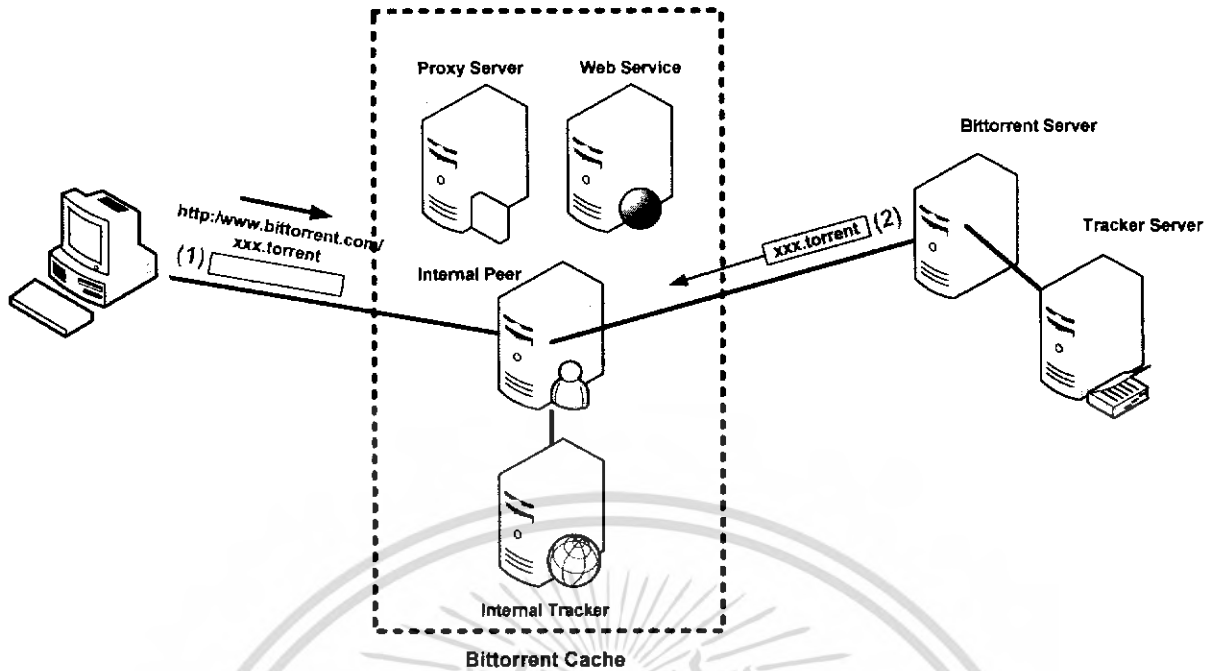
3.3.2 กระบวนการอัฟโหลด

1. เมื่อเครื่องที่อยู่ภายในระบบต้องการทำการอัฟโหลดไฟล์ .torrent ไปยัง External Tracker ภายนอก โดยจะมีการส่งแพ็คเกจที่ข้างในมีส่วนข้อมูลเป็น HTTP POST ออกไป ซึ่ง Proxy Server จะมีการสั่งให้โปรแกรม HTTPdump ซึ่งเขียนโดย Ruby ตรวจสอบว่าเป็นแพ็คเกจ HTTP POST ของไฟล์ .torrent หรือไม่ ซึ่งจะทำโดยการคัดลอกชื่อไฟล์และชื่อ Host มาเก็บไว้ในฐานข้อมูล ซึ่งถ้าเป็นการอัฟโหลดจะมี packet ตอบรับกลับมาตามชื่อที่เก็บไว้ ดังรูปที่ 3.7



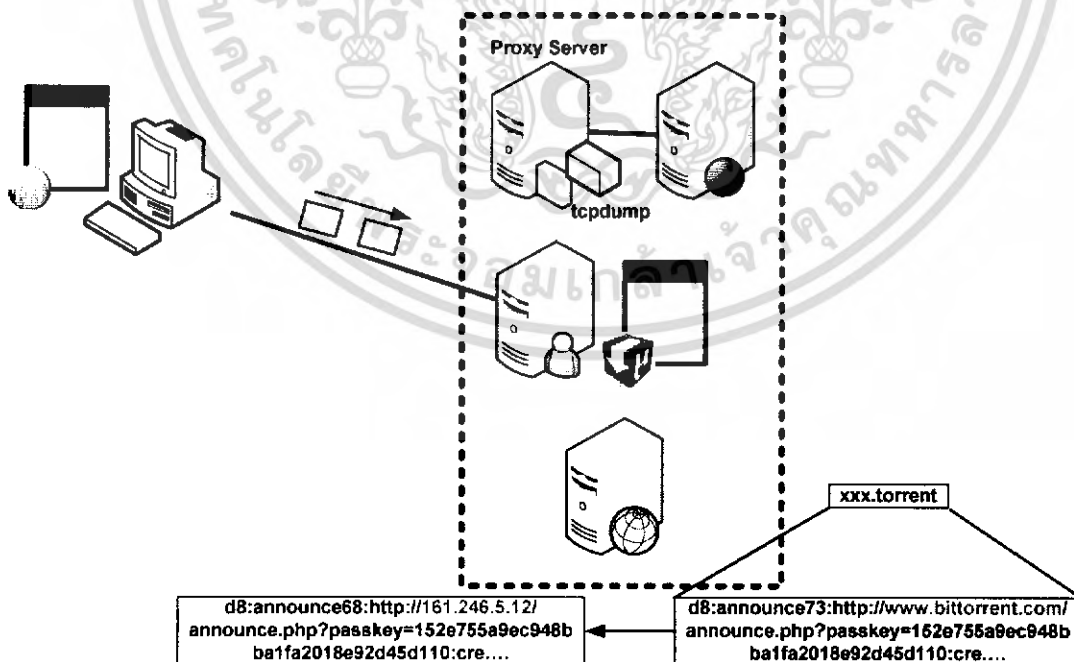
รูปที่ 3.7 แสดงการทำงานการอัฟโหลดไฟล์ในขั้นตอนแรก

2. จากนั้นโดยปกติแล้ว Client จะทำการดาวน์โหลดข้อมูลที่เป็นไฟล์ torrent อีกครั้ง เพราะไฟล์ torrent ที่สร้างขึ้นมานั้นจะต้องไปทำการ register ที่ Bittorrent Server เสียก่อนจึงจะใช้งานได้ เมื่อ Client ดาวน์โหลดไฟล์ torrent มาแล้ว Web Service จะทำการดึงไฟล์ดังกล่าวมา ดังรูปที่ 3.8



รูปที่ 3.8 แสดงการทำงานการอัปโหลดไฟล์ในขั้นที่สอง

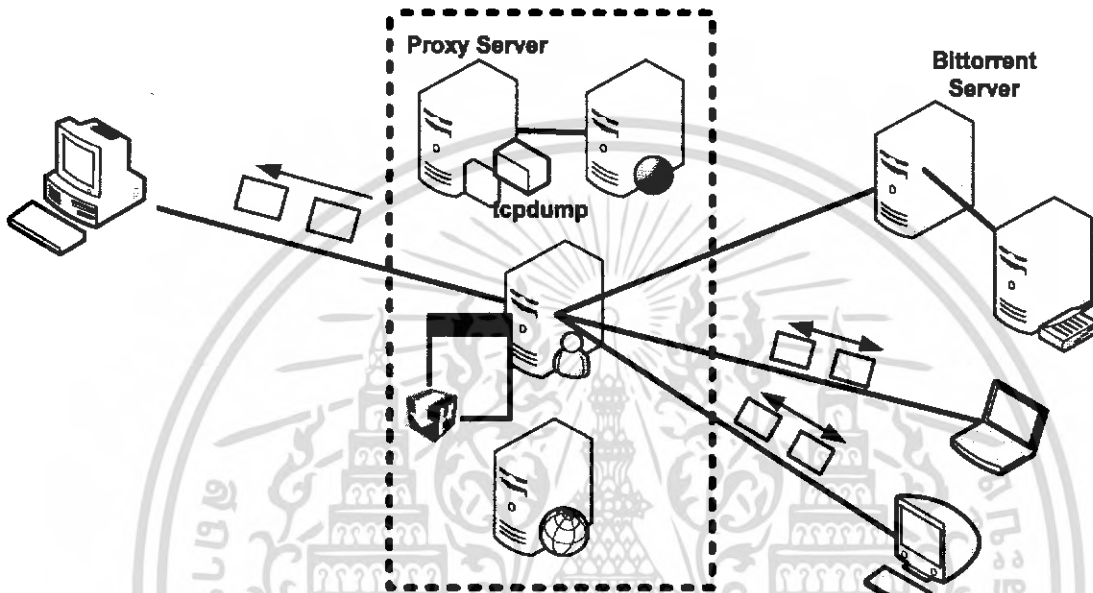
2. Web Service ทำการแก้ไขไฟล์ torrent จาก URL ของ Bittorrent Server ไปเป็นของ Virtual Tracker แทน และใช้ไฟล์ที่ทำการเปลี่ยนข้อมูลแล้วมารันบน Bittorrent Client ดังนั้นเครื่อง Bittorrent Cache จะทำการดาวน์โหลดข้อมูลจากเครื่อง Client ที่อยู่ในแทนดังรูปที่ 3.9



รูปที่ 3.9 แสดงการทำงานการอัปโหลดไฟล์ในขั้นแก้ไขไฟล์ torrent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อเครื่อง Bittorrent Cache ทำการดาวน์โหลดข้อมูลจากเครื่อง Client มาเรื่อยๆ ในขณะเดียวกันก็จะ Seed ไฟล์ดังกล่าวให้กับเครื่องที่เป็น External Peer ซึ่งทำงานเช่นเดียวกับการดาวน์โหลดแต่จะกลับกันตรงที่ การอัปโหลดจะเข้าไปดาวน์โหลดไฟล์จากเครื่อง Client ที่อยู่ในองค์กรก่อน แล้วจึง Seed ให้กับภายนอก โดยภายนอกจะไม่ได้เข้าไปยุ่งเกี่ยวกับเครื่องที่อยู่ภายในแต่อย่างใด ดังรูปที่ 3.10



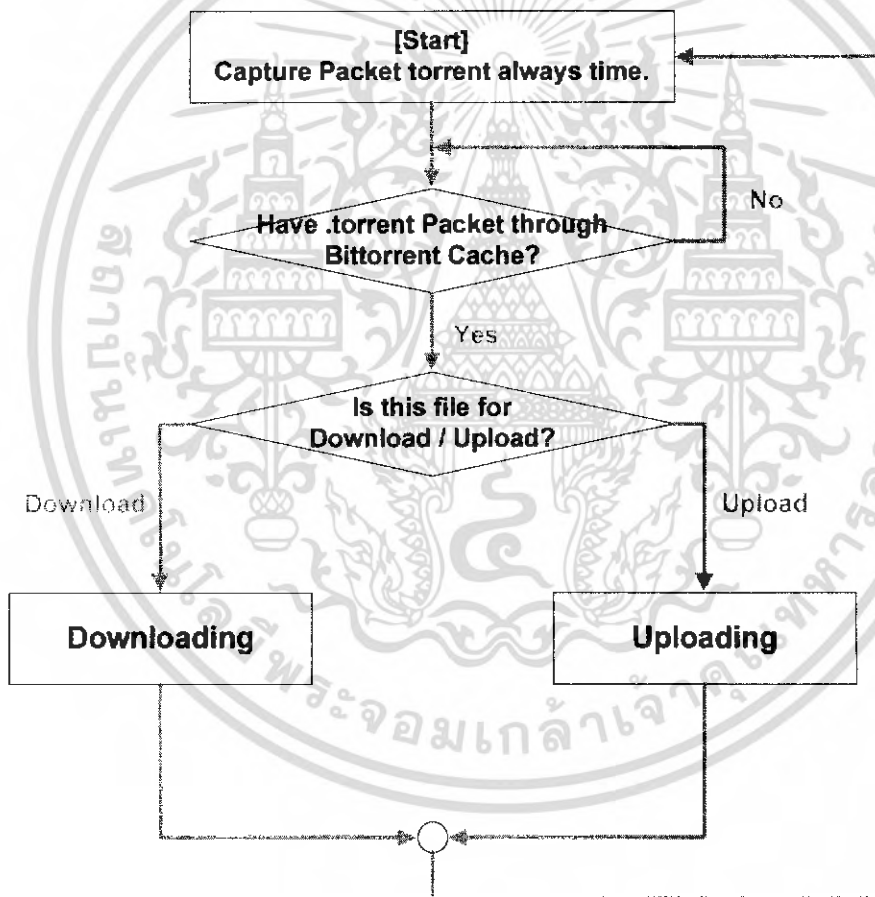
รูปที่ 3.10 แสดงการทำงานการดาวน์โหลดไฟล์ในชั้นตอนสุดท้าย

บทที่ 4

ด้าไฟล์ และหลักการงานโดยละเอียด

ในบทนี้จะกล่าวถึงหลักการงานโดยละเอียดรวมทั้งแสดงออกมาในรูปแบบของแผนผังเพื่อช่วยต่อการเข้าใจในการทำงานของระบบและเพื่อการพัฒนาต่อให้ดียิ่งขึ้น ซึ่งในบทนี้จะไม่อธิบายลงไปถึงในส่วนของโค้ด โปรแกรม ถ้าหากผู้ที่สนใจในรายละเอียดของโค้ดโปรแกรมสามารถเข้าไปดูได้ที่คู่มือโปรแกรมเมอร์ ซึ่งในบทนี้จะอธิบายเฉพาะการทำงาน ตั้งแต่เมื่อปิดทอเรนท์แคชเริ่มทำงาน จะมีการทำงานอย่างไร และเกิดเหตุการณ์ใดขึ้น

4.1 หลักการงานหลักของปิดทอเรนท์แคช



รูปที่ 4.1 แผนผังด้าไฟล์ สำหรับการงานหลัก

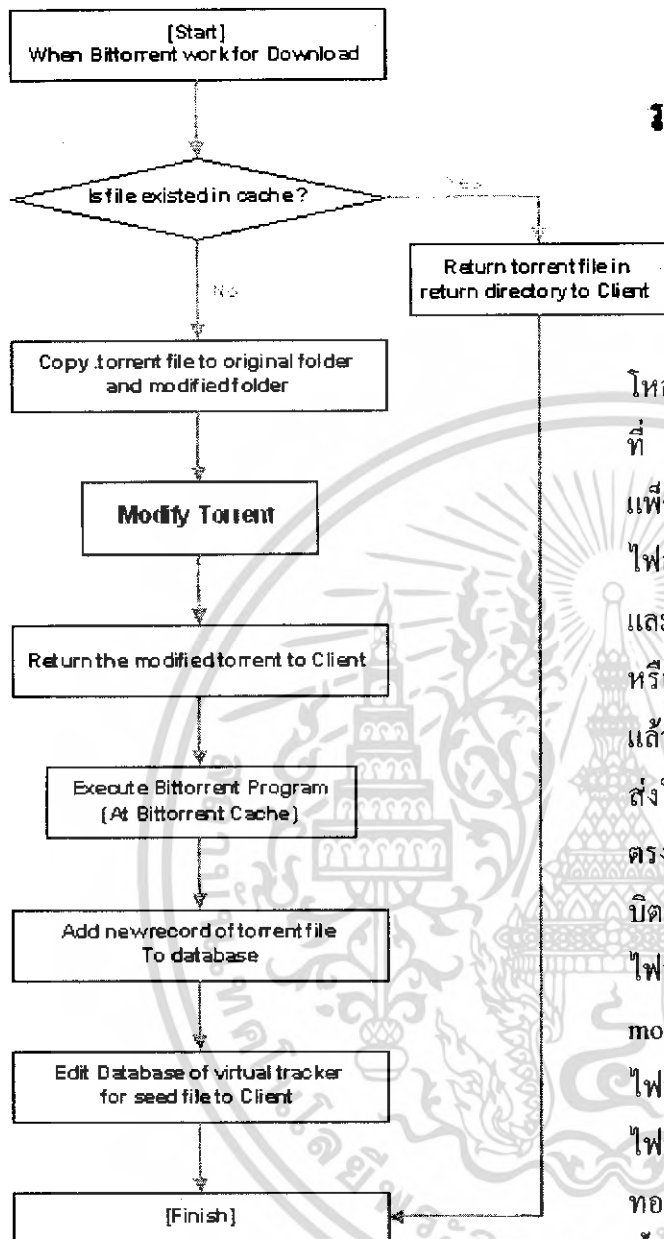
ในส่วนของการงานหลักจะเป็นไปตามแผนผังด้าไฟล์รูปที่ 4.1 คือเมื่อเราตั้งทำงานในบิตทอเรนท์แคชจะเปิดบริการ Tcpdump และ Httpdump บนพรีอิกซ์เซิร์ฟเวอร์ไว้ตลอดเวลาเพื่อดักจับแพ็คเกจทุกแพ็คเกจที่ออกไปยังอินเทอร์เน็ต ซึ่งถ้าเป็นแพ็คเกจปกติก็จะ forward ไปยังเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์เน็ตภายนอกทันที แต่ถ้าหากมีไฟล์เกจที่มีนามสกุล .torrent ซึ่งหมายความว่านี่เป็นไฟล์ที่ เป็นไฟล์ torrent ผ่านมาโปรแกรมดังกล่าวจะดึงไฟล์เกจนั้นไว้ซึ่งก็จะมี การเชื่อมที่ไฟล์เกจนั้นมา จากการ upload หรือการ download ซึ่งวิธีการตรวจสอบจะอยู่ในหัวข้อการอัปโหลด ซึ่งจะกล่าวใน หัวข้อถัดไป จากนั้นถ้าเป็นการดาวน์โหลดหรืออัปโหลดก็จะเข้าไปทำงานไปส่วนดังกล่าว และเมื่อ เสร็จเรียบร้อยแล้ว ก็จะทำการดึงไฟล์เกจต่อไปเรื่อยๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 หลักการทำงานในส่วนของการดาวน์โหลด



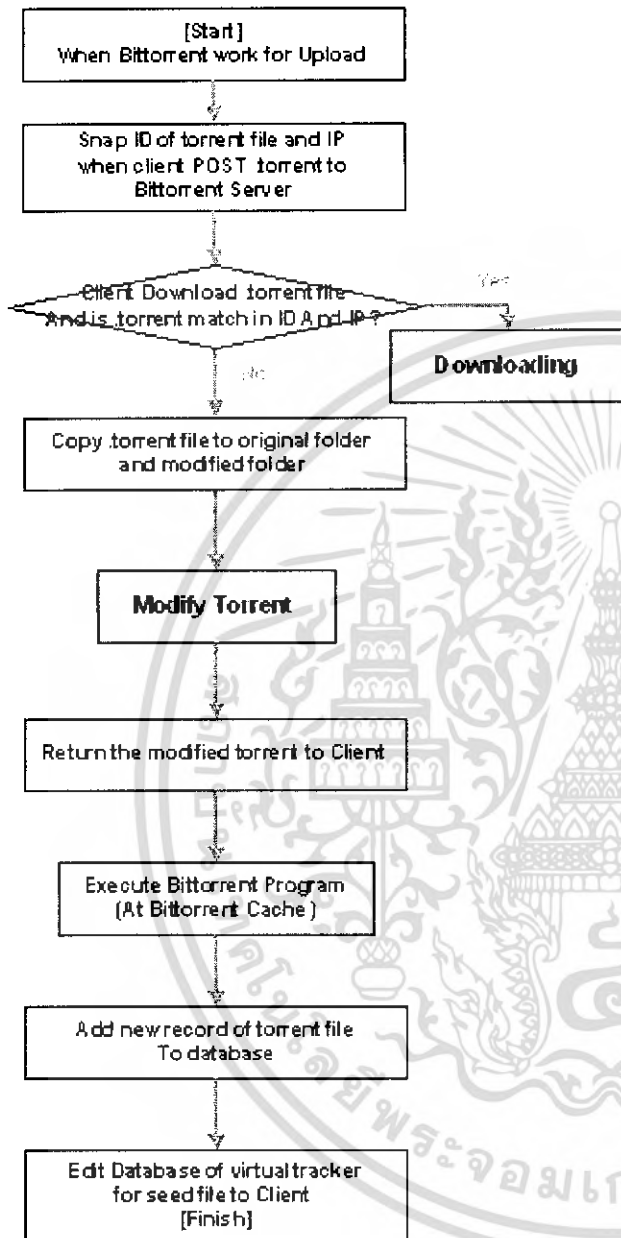
รูปที่ 4.2 แผนผังลำดับไฟล์ สำหรับ

การทำงานในส่วนของการดาวน์โหลด

ในส่วนการทำงานของ การดาวน์โหลด นั้นจะเป็นไปตามแผนผังลำดับไฟล์รูปที่ 4.2 คือเมื่อโปรแกรมได้ทำการดักจับแพ็คเกจที่เป็น .torrent ไฟล์ไว้แล้ว ก็จะนำไฟล์ดังกล่าวมาหาค่า infohash ของไฟล์นั้น และมาเทียบดูว่ามีค่าตรงกับในลำดับเบสหรือไม่ ถ้าตรงแสดงว่าไฟล์นี้เคยดาวน์โหลดแล้ว โปรแกรมก็จะนำไฟล์ทอเรนทที่มีอยู่แล้วส่งให้กับผู้ที่ทำการร้องขอไฟล์ แต่ถ้าหากไม่ตรงแสดงว่าไฟล์นี้เป็นไฟล์ใหม่ที่ไม่เคยมีในบิตทอเรนทแคช ดังนั้นโปรแกรมจะคัดลอกไฟล์ทอเรนทไปยังโฟลเดอร์ original และ modify จากนั้นโปรแกรมก็จะทำการแก้ไขไฟล์ ซึ่งสามารถดูหลักการทำงานการแก้ไขไฟล์ได้ในหัวข้อถัดไป เมื่อทำการแก้ไขไฟล์ทอเรนทเสร็จ โปรแกรมก็จะนำไฟล์ที่แก้ไขนั้นส่งให้กับผู้ร้องขอไฟล์ จากนั้นโปรแกรมก็จะรันโปรแกรมบิตทอเรนท ซึ่งสามารถดูรายละเอียดในส่วนนี้ได้

ในหัวข้อถัดไป ซึ่งในขณะนี้โปรแกรมจะเริ่มทำการดาวน์โหลดไฟล์จากอินเทอร์เน็ตภายนอกและกำลังปล่อยไฟล์ให้กับผู้ใช้งานภายในด้วย และทำการเพิ่มค่าการใช้งานต่างๆลงในลำดับเบสของ torrent และยังเพิ่มไปในส่วนของที่เป็น Virtual Tracker ด้วยเพื่อให้ Virtual Tracker สามารถทำงานได้กับไฟล์ torrent ที่ทำการแก้ไขแล้ว เพื่อให้ผู้ใช้งานสามารถดาวน์โหลดไฟล์ข้อมูลจากบิตทอเรนทแคชได้

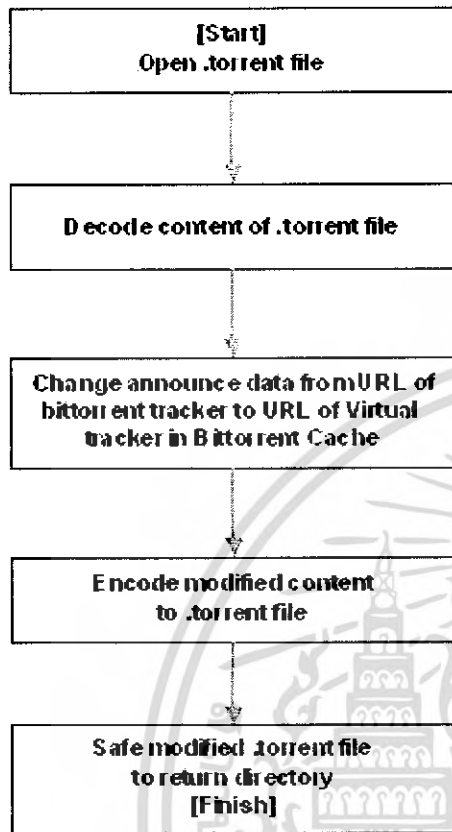
4.3 หลักการทำงานในส่วนของการอัปโหลด



ในส่วนการทำงานของการทำงานอัปโหลดนั้นจะเป็นไปตามแผนผังดาต้าโฟลว์รูปที่ 4.3 คือ การแยกแยะว่าจะเป็นการอัปโหลดหรือดาวน์โหลดนั้น จะแยกตรงที่ถ้าเป็นการอัปโหลด ผู้ใช้จะต้องมีการฝากไฟล์ทอเรนทที่สร้างไว้ไปยังบิตทอเรนทเซิร์ฟเวอร์ ซึ่งขณะที่ทำการฝากนั้นจะมีการใช้ HTTP Packet method POST ซึ่งโปรแกรม HTTPdump จะทำการดักจับแพ็คเกจดังกล่าวเพื่อดึงข้อมูลที่เป็นหมายเลขไอดีของไฟล์และไอพีแอดเดรสหรือชื่อของบิตทอเรนทเซิร์ฟเวอร์ไว้ ซึ่งเมื่อผู้ใช้จะทำการใช้งานไฟล์จะต้องไปดาวน์โหลดไฟล์ที่ฝากไว้อีกที ซึ่งโปรแกรมก็จะมีการตรวจสอบว่าไฟล์ที่ร้องขอนั้นเป็นไฟล์ที่ทำงานเพื่อการอัปโหลดหรือไม่โดยตรวจสอบการไอดีและไอพีที่เก็บไว้ ถ้าไม่ใช่การอัปโหลดก็จะส่งไปทำงานในส่วนของการดาวน์โหลด แต่ถ้าใช่โปรแกรมก็จะทำงานเช่นเดียวกับการดาวน์โหลดทั้งสิ้น แต่จะมีส่วนที่แตกต่างกันในส่วนของการส่งโปรแกรมบิตทอเรนทขึ้นทำงาน ซึ่งได้อธิบายไว้ในหัวข้อหลักการทำงานในส่วนของการส่งโปรแกรมทำงาน เรียบร้อยแล้ว

รูปที่ 4.3 แผนผังดาต้าโฟลว์ สำหรับการทำงานในส่วนของการอัปโหลด

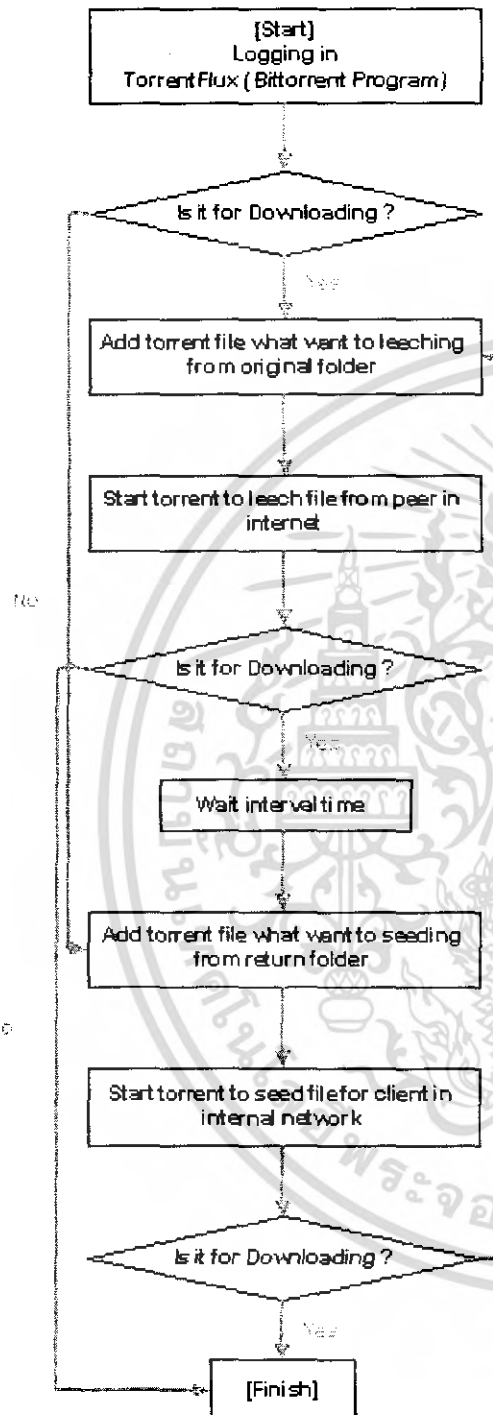
4.4 หลักการทำงานในส่วนของการแก้ไขไฟล์ทอเรนท



ในส่วนการทำงานของ การแก้ไขไฟล์ทอเรนท นั้นจะเป็นไปตามแผนผังคาต้าไฟล์รูปที่ 4.4 คือโปรแกรมจะทำการเปิดไฟล์ทอเรนทดังกล่าวขึ้นมา จากนั้นก็จะต้องทำการถอดรหัสข้อมูลต่างที่อยู่ในไฟล์ทอเรนทเป็น ส่วนๆ จากนั้นในส่วนที่เป็น announce ซึ่งเป็นส่วนที่บอกถึงที่อยู่ของ tracker เพื่อให้โปรแกรมบิตทอเรนทที่ใช้งานเข้าไปร้องขอรายชื่อ peer ต่างๆ รวมถึงส่งข้อมูลต่างๆ ให้กับ tracker นั้นๆ โดยการแก้ไขจาก tracker เดิมไปเป็นของ Virtual Tracker ที่อยู่ที่บิตทอเรนทเคซ ซึ่งก็คือไอพีของบิตทอเรนทเคซ เมื่อแก้ไขเสร็จก็จะต้องทำการเข้ารหัสข้อมูลกลับไปเป็นไฟล์ทอเรนทเหมือนเดิม และ save ไฟล์ดังกล่าวไปยัง return directory

รูปที่ 4.4 แผนผังคาต้าไฟล์ สำหรับการทำงานในส่วนของการแก้ไขไฟล์ทอเรนท

4.5 หลักการทำงานในส่วนของการสั่งโปรแกรมทำงาน



ในการสั่งให้โปรแกรมทำงานนั้นจะมีกระบวนการตามแผนผังด้าไฟล์รูปที่ 4.5 คือถ้าเป็นการดาวน์โหลด จะนำไฟล์ที่เป็นไฟล์ที่ดักจับมานั้นทำการรันก่อน เพื่อให้บิตทอเรนท์แคชโหลดไฟล์ข้อมูลจากภายนอกก่อน จากนั้นจึงเอาไฟล์ที่ทำการแก้ไขแล้วนั้นมารันต่อเพื่อทำการปล่อยไฟล์ที่ดาวน์โหลดมานั้นให้กับเครื่องภายในองค์กร แต่ในทางกลับกันถ้าเป็นการอัปโหลดนั้น โปรแกรมจะทำการรันไฟล์ที่แก้ไขก่อน เพื่อดาวน์โหลดไฟล์จากเครื่องที่อยู่ภายในองค์กร จากนั้นจึงรันไฟล์ที่ดักจับมาได้ เพื่อปล่อยไฟล์ดังกล่าวออกไปยังอินเทอร์เน็ตแทน

รูปที่ 4.5 แผนผังด้าไฟล์ สำหรับการสั่งโปรแกรมทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

ระบบแคชสำหรับโปรแกรมบิตทอเรนท์

ในบทนี้จะกล่าวถึงคุณสมบัติของอุปกรณ์ที่ใช้ในการทดลองและผลที่ได้จากการทดลองระบบซึ่งแสดงความสามารถในการจำกัดการใช้งานเครือข่าย โดยในการทดลองจะแบ่งออกเป็น 2 การทดลองคือในส่วนแรกเป็นการเปรียบเทียบระหว่างการดาวน์โหลดแบบผ่านระบบแคชกับการดาวน์โหลดแบบปกติ และในส่วนที่สองเป็นการเปรียบเทียบเวลาที่ใช้ในการดาวน์โหลดระหว่างการดาวน์โหลดโดยผ่านระบบแคชกับการดาวน์โหลดแบบปกติ นอกจากนี้ยังทดสอบการทำงานว่าเป็นไปตามที่ได้ออกแบบไว้หรือไม่ ปัญหาที่พบและการแก้ไขได้บันทึกไว้ที่ช่วงท้ายของบทนี้

5.1 แบบจำลองและคุณสมบัติของอุปกรณ์ที่ใช้ทดลอง

เนื่องจากข้อจำกัดด้านทรัพยากร ความแตกต่างของประสิทธิภาพเครื่อง ในการทดสอบนี้แบบจำลองใช้เครื่องคอมพิวเตอร์ทั้งหมดห้าเครื่องตั้งอยู่บนเครือข่ายเดียวกัน และอีกหนึ่งเครื่องอยู่คนละเครือข่าย โดยมีหน้าที่แตกต่างกันไปดังนี้ เครื่องในเครือข่ายเครื่องแรกทำหน้าที่เป็นแคช ส่วนอีกสี่เครื่องทำหน้าที่เป็นเครื่องภายในเครือข่ายทำหน้าที่ดาวน์โหลดไฟล์โดยผ่านเครื่องแคช และเครื่องภายนอกเครือข่ายทำหน้าที่ปล่อยไฟล์ให้กับเครื่องภายในเครือข่าย

- คุณสมบัติของเครื่องที่ทำงานเป็นแคช
 - Central Processing Unit: Intel® Pentium® 4 CPU 3.2 GHz
 - Main board: ASUS® P4P800 Special Edition
 - Memory: 1024 MB Dual Channel at 800 MHz
 - Operating System: Fedora Core 5 Linux, Kernel 2.6
 - Network Adapter: Realtek RTL8139 Family PCI Fast Ethernet NIC, connected at 100.0 Mbps
 - IP Address: 161.246.5.43
 - BitTorrent Client: TorrentFlux 2.3
 - Web Server: Lampp Web Service pack for Linux. Included: Apache, MySql and PhPMyAdmin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คุณสมบัติของเครื่องที่ทำงานเป็นเครื่องบิตทอเรนทเซอร์ฟเวอร์
 - Central Processing Unit: AMD Athlon™ XP 2200+ 2.3 GHz
 - Main board: ASUS® A7V8X-X
 - Memory: 512 MB
 - Operating System: Fedora Core 5 Linux, Kernel 2.6
 - Network Adapter: VIA Rhine 2 Fast Ethernet Adapter, connected at 100.0 Mbps
 - IP Address: 161.246.5.44
 - Web Server: XAMPP Web Service pack for Windows. Included: Apache, MySql , PhPMyAdmin and Perl

- คุณสมบัติของเครื่องที่ทำงานเป็นเครื่องเป็น Client ภายในองค์กรและ Peer ที่อยู่นอกองค์กร
 - Central Processing Unit: AMD Athlon™ XP 2200+ 2.3 GHz
 - Main board: ASUS® A7V8X-X
 - Memory: 512 MB
 - Operating System: Microsoft® Windows® XP Professional, Service Pack 2
 - Network Adapter: VIA Rhine 2 Fast Ethernet Adapter, connected at 100.0 Mbps
 - IP Address: 161.246.5.43 และ 161.246.5.22
 - BitTorrent Client: uTorrent 2.3 และ BitComet 0.7

- ไฟล์ที่ใช้ทดสอบ
 - ไฟล์ขนาด 92 MB โดยมี Metadata เป็น TOEIC.torrent ขนาด 14.9 KB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การตั้งค่าต่างๆในการทดลอง และการทดลองการใช้งาน

- เครื่อง Bittorrent Server

เปิดการทำงานของเว็บเซิร์ฟเวอร์ ซึ่งได้ติดตั้ง Bittorrent Server ไว้เรียบร้อยแล้วโดย Server ที่ทำการใช้คือ TBDEV.NET sa27s edition2 ซึ่งในปัจจุบันนั้น ไฟล์ของบิตทอเรนทเซิร์ฟเวอร์ไม่ใช่แล้ว เพราะเนื่องจากช่วงที่โครงการนี้นำมาใช้งานนั้น ยังอยู่ในการพัฒนาและเป็น open source โครงการนี้จึงนำไปใช้ เพราะบิตทอเรนทเซิร์ฟเวอร์นี้เป็นรูปแบบที่ใช้ในอินเทอร์เน็ตทั่วไป ซึ่งมีการใช้ระบบของ Cookie tracker และ ratio ที่เหมาะสม ไม่เหมือนกับบิตทอเรนทเซิร์ฟเวอร์สมัยก่อนซึ่งเปิดให้กับ peer ที่ไม่ได้เป็นสมาชิกสามารถเข้าไปดาวน์โหลดไฟล์ได้

- เครื่องภายนอก

สร้าง Metadata ของแต่ละไฟล์ขึ้นมาตามที่กำหนด โดยใช้โปรแกรม uTorrent สร้างไฟล์ torrent และนำไฟล์ดังกล่าวฝากไว้ที่ Bittorrent Server จากนั้นเครื่องภายนอกก็จะทำการดาวน์โหลดไฟล์ torrent ที่ทำการ register กับทาง Bittorrent Server อีกครั้งและนำไฟล์ดังกล่าวทำงานบนโปรแกรม uTorrent เพื่อทำการ Seed file

- เครื่อง Bittorrent Cache

เปิดการทำงานของพร็อกซี่ให้มีการใช้งานและเปิดให้เว็บเซิร์ฟเวอร์ทำงานซึ่งจะมี Torrentpier และ phpBB2 เป็น Virtual Tracker, TorrentFlux เป็น Bittorrent Program, และมี NTOP เพื่อตรวจสอบประสิทธิภาพการใช้งานของเครือข่าย จากนั้นเครื่อง Bittorrent Cache จะต้องปิด firewall

- เครื่องภายใน

ทำการโหลดไฟล์ Torrent ที่ต้องการจาก Bittorrent Server และนำมาทำงานบนโปรแกรม Bitcomet จากนั้นสังเกตการณ์และบันทึกค่าต่างๆที่เกิดขึ้น

โดยจะแสดงการทำงาน โดยละเอียดดังนี้

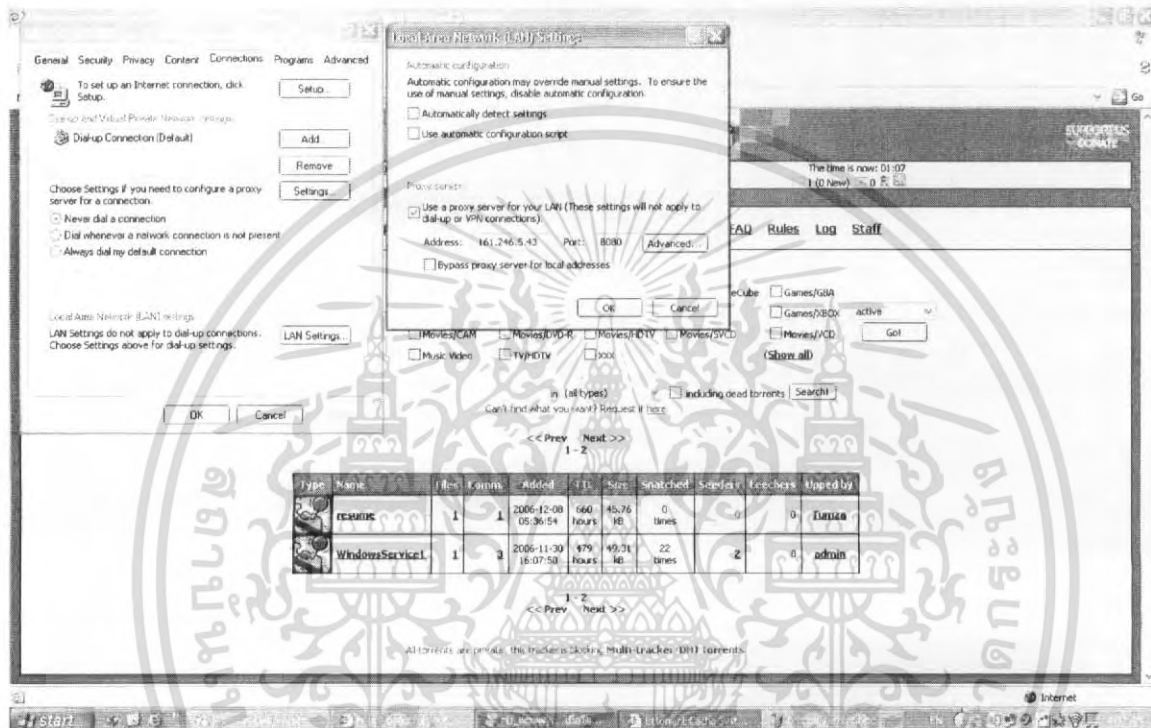
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.1 การดาวน์โหลด

การดาวน์โหลดนั้นจะแสดงผลให้ดูที่ละฝั่งเพื่อง่ายต่อการเข้าใจซึ่งแบ่งเป็นสองฝั่งคือ ฝั่ง Client และฝั่ง Bittorrent Cache

5.2.1.1 การทำงานทางฝั่ง Client

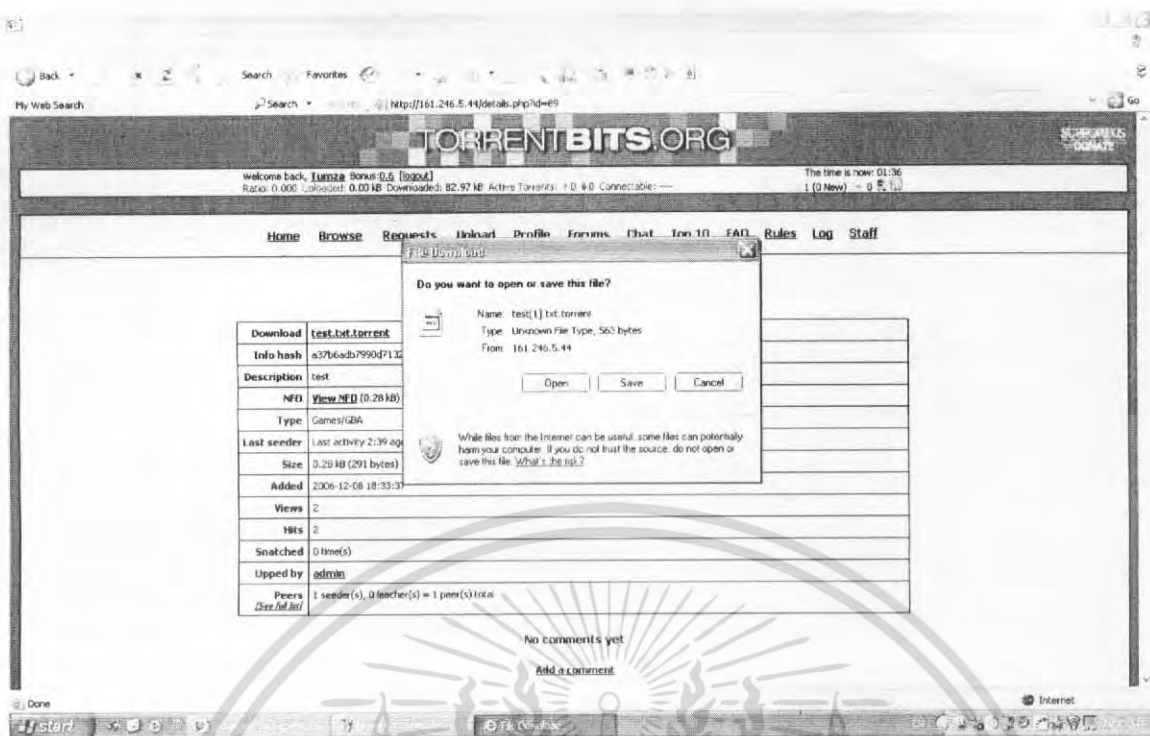
1. ผู้ใช้ซึ่งเป็น Client ภายในองค์กรจะต้องทำการตั้งค่า proxy ไปยังเครื่อง proxy server ซึ่งก็คือเครื่องที่เป็น Bittorrent Cache (161.246.5.43) ดังรูปที่ 5.1



รูปที่ 5.1 การตั้งค่าพร็อกซีไปยังเครื่อง Bittorrent Cache

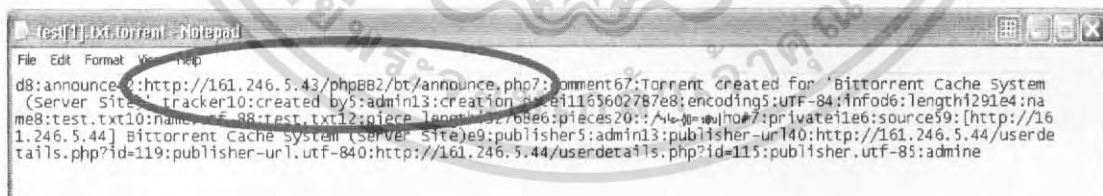
2. ผู้ใช้จะทำการดาวน์โหลดไฟล์ .torrent ที่ต้องการจาก เครื่องที่เป็น Bittorrent Server ซึ่งในที่นี้ใช้ IP 161.246.5.44 ดังรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 ภาพแสดงขณะเครื่อง Client ดาวน์โหลดไฟล์จาก Bittorrent Server

- ไฟล์ torrent ดังกล่าวได้ถูกดึงโดย Bittorrent Cache และทำการเปลี่ยน Metadata ด้านในของไฟล์ torrent จาก URL ของ Bittorrent Server เป็นของ Bittorrent Cache แทน รวมถึงในขณะนี้ Bittorrent Cache ก็จะนำไฟล์ torrent ที่เป็นต้นฉบับนั้นทำงานบน TorrentFlux เพื่อทำการดาวน์โหลดไฟล์จาก Peer ที่อยู่ภายนอกองค์กรแทน จากนั้น Bittorrent Cache ก็จะส่งไฟล์ torrent ที่ทำการแก้ไขแล้วกลับไปยังเครื่อง Client โดยไฟล์ที่ทำการแก้ไขข้อมูลแล้วมีลักษณะข้อมูลภายในดังรูปที่ 5.3 ส่วนไฟล์ต้นฉบับจะมีลักษณะดังรูปที่ 5.4



รูปที่ 5.3 ไฟล์ที่ได้จากการดาวน์โหลดผ่านทาง proxy server ซึ่งได้ Metadata ด้านในเป็น URL ของ Bittorrent Cache เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

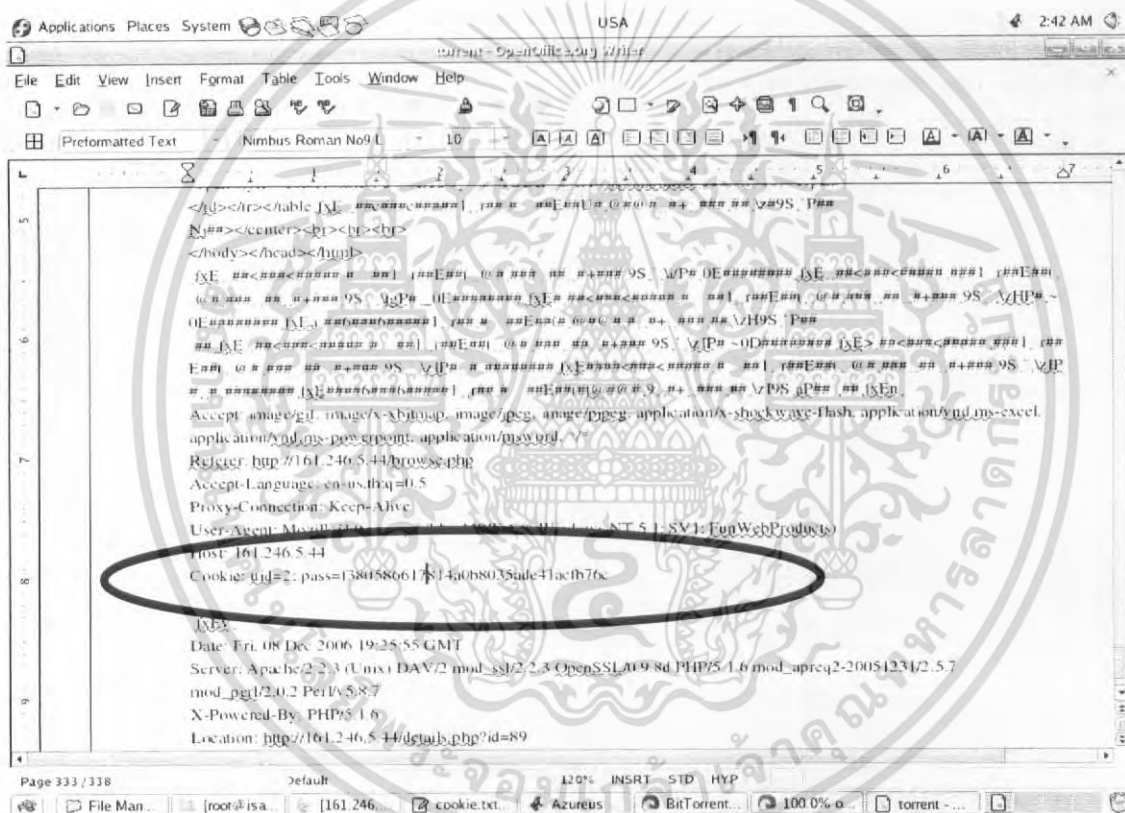
1. โดยปกติเมื่อเริ่มการทำงานของ Bittorrent Cache จะต้องทำการรัน

/opt/lampp/lampp start เป็นการสั่งให้ Web Server เริ่มทำงาน

/opt/lampp/htdocs/httpdump_download/httpdump.sh เป็นการเริ่มดักจับแพ็คเก็ตจากการดาวน์โหลด

#/opt/lampp/htdocs/httpdump_upload/httpdump.sh เป็นการเริ่มดักจับแพ็คเก็ตจากการอัปโหลด

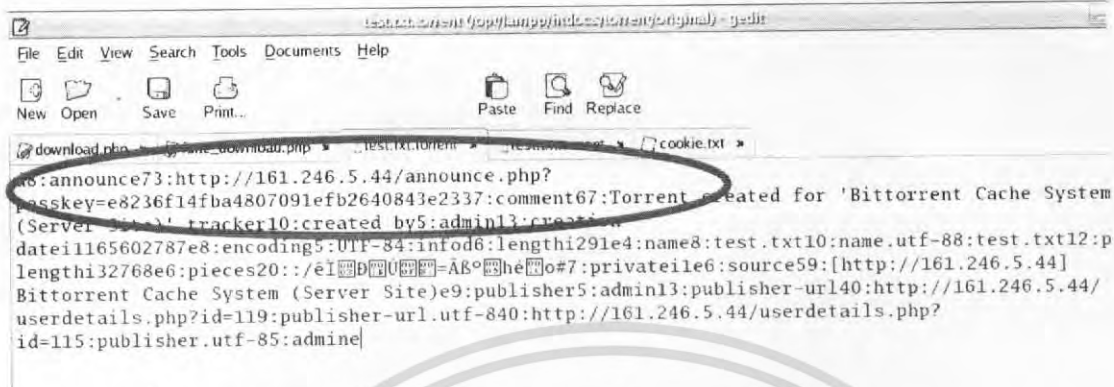
ซึ่งจากรูปที่ 5.6 จะสามารถเห็นได้ว่าขณะที่เครื่อง Client กำลังดาวน์โหลดไฟล์จาก Bittorrent Server และ Bittorrent กำลังดักจับ Cookie และ URL เพื่อทำการปลอมตัวเป็น Client ผู้นั้นและไปขอดาวน์โหลดไฟล์แทน



รูปที่ 5.6 ภาพแสดงขณะที่เครื่อง Bittorrent Cache ดักจับข้อมูลที่เป็น Cookie เพื่อทำการปลอมตัวเป็น Client เพื่อไปดาวน์โหลดไฟล์ที่ต้องการ Bittorrent Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ซึ่งจากรูปที่ 5.7 จะเห็นว่าเมื่อ Bittorrent Server เห็นการร้องขอจากไฟล์ดังกล่าวก็จะส่งไฟล์ torrent ดังกล่าวไปยังเครื่อง Bittorrent Cache



รูปที่ 5.7 ไฟล์ torrent ที่ Bittorrent Server (161.246.5.44) ส่งมายัง Bittorrent Cache

3. จากนั้น Bittorrent Cache ทำการเปลี่ยนแปลงแก้ไขไฟล์ .torrent นั้นๆ โดยแก้ไข Metadata จาก URL ที่เป็น tracker ของ Bittorrent Server เป็นของ Virtual Tracker ที่อยู่ที่ Bittorrent Cache เอง และส่งไฟล์ที่แก้ไขกลับไปให้ผู้ใช้ โดยไฟล์ที่ถูกแก้ไขแล้วจะมีข้อมูลภายในเป็นดังรูปที่ 5.8



รูปที่ 5.8 ไฟล์ torrent ที่ Bittorrent Cache (161.246.5.43) ทำการเปลี่ยนแปลงข้อมูลภายในไฟล์เรียบร้อยแล้ว

4. จากนั้น Bittorrent Cache ก็จะสั่งให้ TorrentFlux ซึ่งเป็น Bittorrent Program ทำงาน โดยใช้ไฟล์ที่เป็นต้นฉบับที่ได้มาจาก Bittorrent Server ทำงานดังรูปที่ 5.9 ซึ่งแสดงว่า Bittorrent Cache ทำการดาวน์โหลดไฟล์จาก Peer ที่อยู่บน Internet ภายนอกและใช้ไฟล์ที่ทำการแก้ไขทำงานด้วยซึ่งนั่นหมายความว่าไฟล์ที่ดาวน์โหลดได้มานั้นจะถูก Seed ให้กับเครื่องที่อยู่ภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The screenshot shows the BitTorrent Flux web interface. At the top, the browser title is "BitTorrent Client - Mozilla Firefox" and the address bar shows "http://localhost/torrentflux/html/index.php". The interface includes navigation links like "Latest Release Notes", "Fedora Project", and "Fedora Weekly News". A storage indicator shows "43% (7.97 GB Free)". Below this is a table of torrents:

Name	Size	Location	Speed	Status
LogMeIn.exe.torrent.cache	3.6 MB	root	seedina 0.01 kB/s @ 100.0%	Download Succeeded
LogMeIn.exe.torrent	3.6 MB	root	seedina 0.11 kB/s @ 0.0%	Download Succeeded
3convar.torrent.cache	11.8 MB	root	97.2% @ 0.0 kB/s	Torrent Stopped
3convar.torrent	11.8 MB	root	97.2% @ 0.0 kB/s	Connecting to Peers
Umplyerc.exe.torrent.cache	5.4 MB	root	seedina 0.01 kB/s @ 100.0%	Download Succeeded
Umplyerc.exe.torrent	5.4 MB	root	seedina 0.01 kB/s @ 100.0%	Download Succeeded

At the bottom of the interface, there are controls for "Torrent Details", "Run Torrent", "Stop Torrent", "Seed Torrent", "Delete", and "Download Torrent". A status bar shows "Current Download: 0.00 kB/s", "Current Upload: 0.00 kB/s", "Free Space: 7.97 GB", and "Server Load: 1.03". A refresh message says "Page Will Refresh Every 60 Seconds".

รูปที่ 5.9 ภาพแสดงการทำงานของ TorrentFlux ที่กำลังดาวน์โหลดข้อมูลจากภายนอก และในขณะเดียวกันก็ Seed ไฟล์ดังกล่าวให้กับเครื่องที่อยู่ภายในองค์กรด้วย

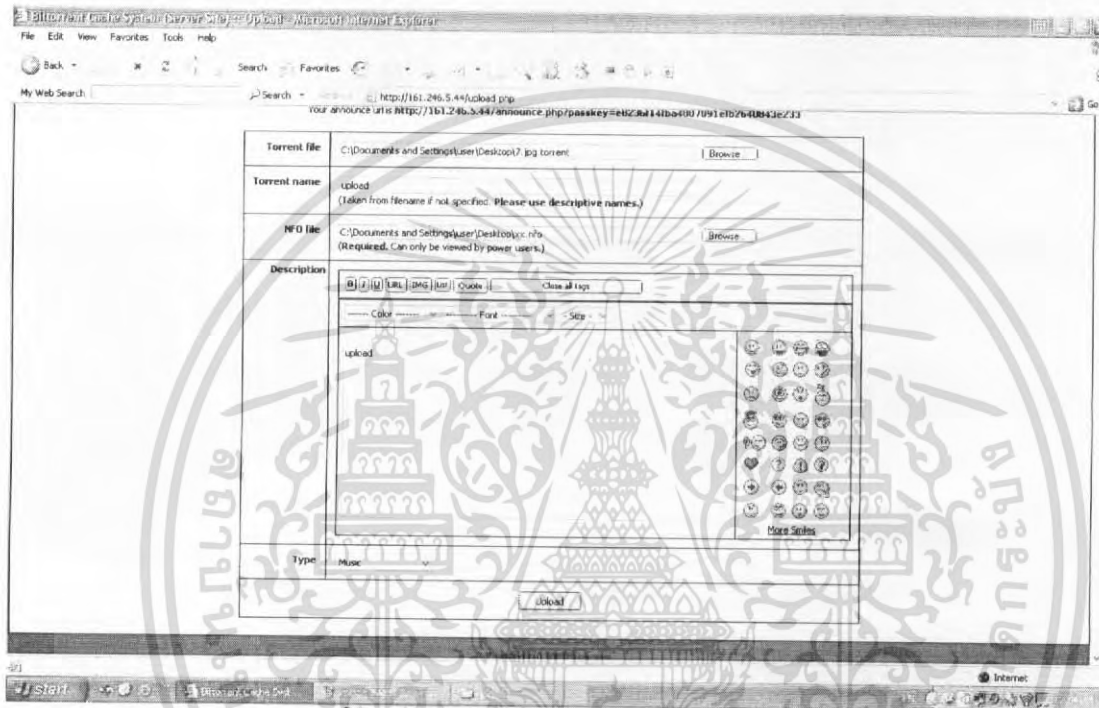
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 การอัปโหลด

การอัปโหลดนั้นเราก็จะแสดงผลให้ดูที่ละฝั่งเช่นเดียวกัน เพื่อต่อการเข้าใจซึ่งแบ่งเป็นสองฝั่งคือ ฝั่ง Client และฝั่ง Bittorrent Cache

5.2.2.1 การทำงานทางฝั่ง Client

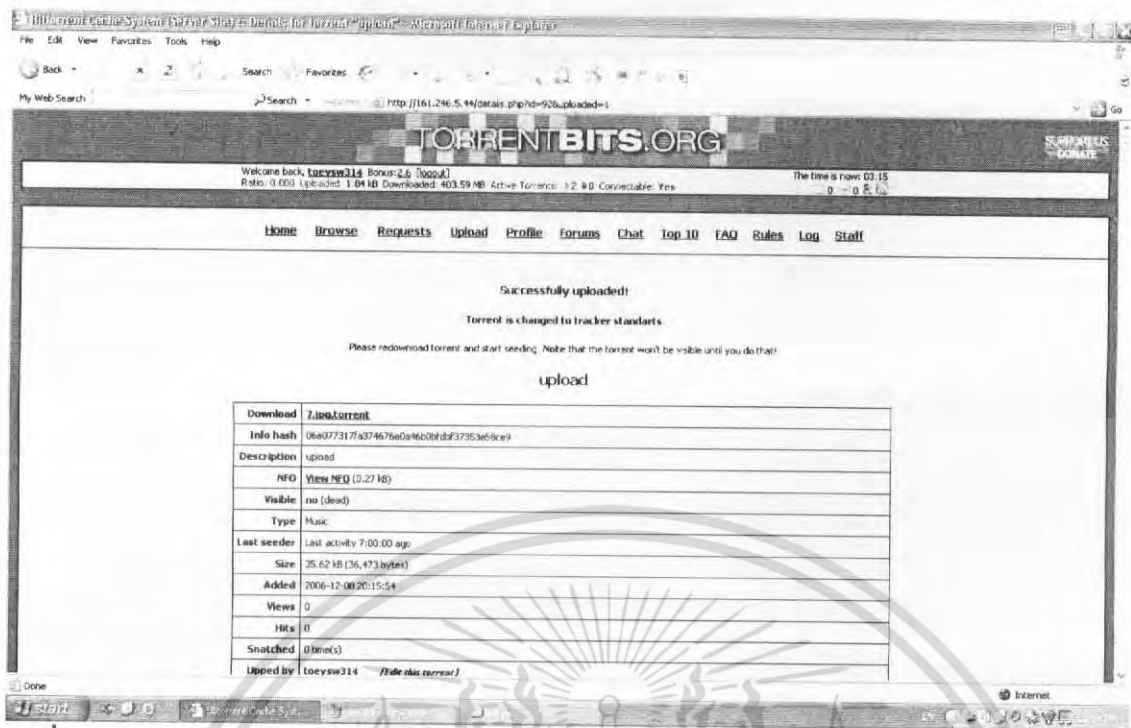
1. Client จะต้องทำการฝากไฟล์ที่สร้างโดย Bittorrent Program และนำไฟล์ดังกล่าวไปอัปโหลดไว้ที่ Bittorrent Server ดังรูปที่ 5.10



รูปที่ 5.10 ภาพหน้าเว็บไซต์ขณะอัปโหลดไฟล์

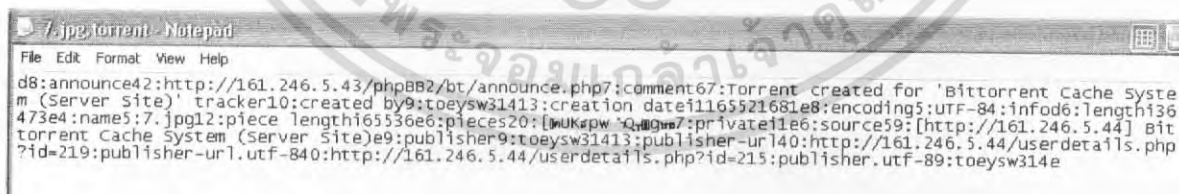
2. จากนั้น Bittorrent Server จะรายงานผลงานสามารถอัปโหลดได้สำเร็จหรือไม่ ดังรูปที่ 5.11 จากนั้นเครื่อง Client ก็จะทำการดาวน์โหลดไฟล์ที่ฝากไว้ที่ Bittorrent Server อีกครั้งหนึ่งเพื่อนำไฟล์ดังกล่าวมา Seed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 ภาพหน้าเว็บไซต์เมื่อทำการอัปโหลดไฟล์สำเร็จและ Client ต้องทำการโหลดไฟล์ torrent อีกครั้ง

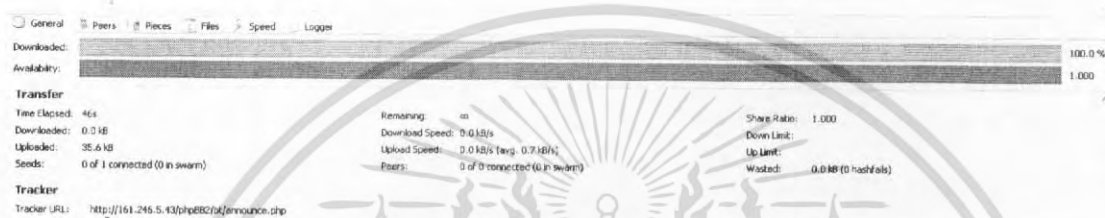
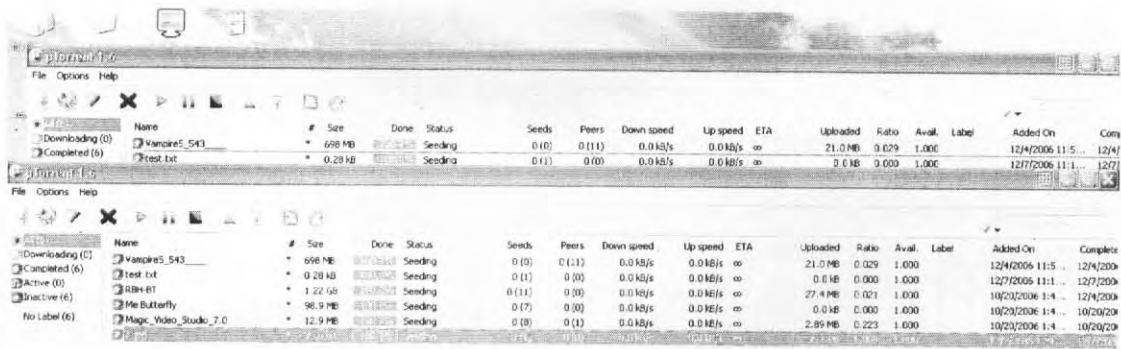
3. ในขณะที่เครื่อง Client กำลังดาวน์โหลดไฟล์อยู่นั้น เครื่อง Bittorrent Cache นั้นก็จะทำการดึงข้อมูลที่เป็น Cookie และ URL เพื่อนำไปโหลดไฟล์ torrent ดังกล่าวแทน และทำการแก้ไขไฟล์ Metadata จาก URL ที่เป็นของ Bittorrent Tracker ซึ่งโดยปกติจะเป็น IP เดียวกับ Bittorrent Server ไปเป็น URL ของเครื่องที่เป็น Virtual Tracker ที่อยู่บน Bittorrent Cache ดังรูปที่ 5.12 และทำการส่งไฟล์ที่แก้ไขกลับไปให้กับเครื่อง Client



รูปที่ 5.12 ข้อมูลข้างในของไฟล์ torrent ที่เครื่อง Client ได้รับมา

4. จากนั้นเครื่อง Client จะนำไฟล์ torrent ดังกล่าวมาทำงาน uTorrent (Bittorrent Program) ดังรูปที่ 5.13 ซึ่งจะสามารถ seed ได้สำเร็จแต่เป็นการ seed ให้กับเครื่อง Bittorrent Cache แทน ซึ่งในขณะที่เครื่อง Bittorrent Cache จะทำการดาวน์โหลดไฟล์จากเครื่อง Client เพื่อจะใช้ไฟล์ดังกล่าวไป Seed ให้กับภายนอกแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.13 หน้าต่างของ Bittorrent Program ขณะทำการ Seed ไฟล์ให้กับ Bittorrent Cache

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2.2 การทำงานทางฝั่ง Bittorrent Cache

1. เมื่อเครื่อง Client ทำการอัปโหลดไฟล์ไปยัง Bittorrent Server ซึ่ง Bittorrent Cache จะทำการดักจับแพ็กเก็ตได้ 3 แพ็กเก็ตเพื่อจับข้อมูลต่างๆ ดังภาพที่ 5.14

```
HTTP/1.1 302 Found
Date: Fri, 08 Dec 2006 20:15:54 GMT
Server: Apache/2.2.3 (Unix) DAV/2 mod_ssl/2.2.3 OpenSSL/0.9.8d PHP/5.1.6 mod_apreq2-20051231/2.
mod_perl/2.0.2 Perl/v5.8.7
X-Powered-By: PHP/5.1.6
Location: http://161.246.5.44/detail.php?id=92&uploaded=1
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

```
GET /details.php?id=92&uploaded=1 HTTP/1.0
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Referer: http://161.246.5.44/upload.php
Accept-Language: en-us,th;q=0.5
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FunWebProducts)
Host: 161.246.5.44
Pragma: no-cache
Cookie: uid=2; pass=f380586617814a0b8035ade41acfb76c
Via: 1.0 isag43.ce.kmitl.ac.th:8080 (squid/2.5.STABLE12)
X-Forwarded-For: 161.246.5.11
Cache-Control: max-age=259200
Connection: keep-alive
```

รูปที่ 5.14 ข้อมูลของแพ็กเก็ตที่ทำการดักจับได้

2. ซึ่ง Bittorrent Cache จะใช้ script ในกรองข้อมูลที่เป็น id และ ip ออกมาและนำข้อมูลดังกล่าวเก็บลงไว้ในฐานข้อมูลดังรูปที่ 5.15 เมื่อ Client ทำการดาวน์โหลดไฟล์ torrent อีกครั้งหนึ่ง Bittorrent Cache จะรู้ว่าการโหลดไฟล์ดังกล่าวเป็นการโหลดไฟล์เพื่ออัปโหลด

รูปที่ 5.15 ข้อมูลที่คัดจับได้นำไปเพิ่มในฐานข้อมูล

- จากนั้น เมื่อเครื่อง Client ทำการดาวน์โหลดไฟล์กลับไปเพื่อ seed ให้กับ tracker จะเข้าสู่กระบวนการดาวน์โหลดโดยจะมีการตรวจสอบก่อนว่าเป็นการดาวน์โหลดเพื่ออัปเดตหรือไม่ ถ้าเป็น Bittorrent Cache จะทำการส่งไฟล์ .torrent ที่ทำการแก้ไขก่อนโดย TorrentFlux เพื่อทำการดาวน์โหลดไฟล์จากเครื่อง Client จากนั้นจึงส่งไฟล์ torrent ที่เป็นต้นฉบับที่ดาวน์โหลดมาเพื่อทำการ Seed ไฟล์จากเครื่อง Client ให้กับ Peer ที่อยู่บนอินเทอร์เน็ต ดังนั้นเครื่อง Bittorrent Cache ก็จะเป็นผู้ Seed ไฟล์ให้กับ Peer ที่อยู่บน Internet ข้างนอกแทนเครื่อง Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การทดลองและผลการทดลอง

5.3.1 การทดลองที่ 1 เปรียบเทียบข้อมูลขาเข้าและขาออกในการดาวน์โหลดระหว่างการดาวน์โหลดผ่านระบบแคชและการดาวน์โหลดแบบปกติ

การทดสอบนี้ ใช้โปรแกรม Ntop ซึ่งมีความสามารถในการดูจำนวนข้อมูลและการรับส่งข้อมูลต่างๆ ของเครื่องภายในเครือข่ายที่ติดต่อกับอุปกรณ์ของเครื่องที่มีโปรแกรมนี้ โดยติดตั้งโปรแกรมไว้ที่เครื่องแคช จึงสามารถตรวจสอบการรับส่งข้อมูลของเครื่องทั้งสี่ได้ โปรแกรม Ntop นี้ใช้ WinPCap เป็นตัวจับข้อมูล รายละเอียดของโปรแกรมนี้ สามารถดูได้ที่ <http://www.ntop.org/> โดยโปรแกรม Ntop นี้สามารถทำงานได้ทั้งในลักษณะ Interactive Mode และ Web Mode

ทุกครั้งก่อนการทดสอบทุกครั้ง จะมีการลบข้อมูลเก่าและเตรียมโปรแกรม Ntop ใหม่ โดยทำการลบงานที่โหลดออกจากโปรแกรม BitComet และหยุดการทำงานของ Ntop เพื่อลบสถิติที่มีอยู่แล้ว ก่อนการทดสอบกรณีที่โหลดพร้อมกับแคช จะต้องทำการลบข้อมูลออกจากแทรคเกอร์และ Bittornado และไฟล์ Metadata ที่อยู่ในเครื่อง

ตอนที่ 1 ทดลองเปรียบเทียบปริมาณข้อมูลขาเข้าและขาออกเมื่อปริมาณเครื่องที่ทำการดาวน์โหลดมีมากขึ้น โดยเป็นการดาวน์โหลดผ่านระบบแคช

ทำการเรียก Ntop ให้ทำงานก่อนเริ่มการโหลดข้อมูล จากนั้นให้เครื่องภายในเครือข่ายเครื่องแรกทำการดาวน์โหลดข้อมูล แล้วตามด้วยเครื่องที่สอง , สาม , และเครื่องสุดท้ายตามลำดับ โดยในระหว่างทำการทดลองให้ยุติการใช้งานอื่นๆ ในแต่ละเครื่องเพื่อให้ได้ผลที่ใกล้เคียงกับการใช้งานที่เกิดขึ้นจริง เก็บผลต่างๆ เมื่อทุกเครื่องดาวน์โหลดเสร็จสมบูรณ์

ผลการทดสอบ

เครื่อง	ปริมาณข้อมูลขาออก	ปริมาณข้อมูลขาเข้า
1	93.8 MB	96.4 MB
2	123.1 MB	96.5 MB
3	138.4 MB	96.5 MB
4	143 MB	96.6 MB

ตารางที่ 5.1 ปริมาณข้อมูลขาเข้าและขาออกจากเครื่องแคช โดยไฟล์ข้อมูลมีขนาด 92 MB

จากผลการทดลองสังเกตได้ว่าเมื่อเครื่องแรกทำการดาวน์โหลดนั้นปริมาณข้อมูลขาเข้าซึ่งเป็นการดาวน์โหลดไฟล์ข้อมูลจากภายนอกมาเก็บไว้ที่เครื่องแคชและปริมาณข้อมูลขาออกมีขนาดมากกว่าขนาดข้อมูลจริงๆ เล็กน้อยซึ่งเกิดจากการดาวน์โหลด Metadata และการเชื่อมต่อระหว่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เมื่อเครื่องต่อไป ทำการดาวน์โหลดจะเห็นได้ว่าปริมาณข้อมูลทั้งขาเข้าและขาออกมีขนาดลดลงอย่างมาก

ข้อมูลขาเข้าเป็นการดาวน์โหลดข้อมูลจากเครื่องภายนอกมาเก็บไว้ที่เครื่องแคช แต่เครื่องแคชมีข้อมูลอยู่แล้วทำให้การดาวน์โหลดในครั้งต่อมา ไม่จำเป็นต้องดาวน์โหลดอะไรอีกแล้วทำให้ปริมาณข้อมูลขาเข้าลดลงอย่างมาก

ข้อมูลขาออกเป็นการปล่อยไฟล์ให้กับเครื่องภายในที่ทำการร้องขอไฟล์ แต่ข้อมูลที่เครื่องที่ 2 ต้องการนั้นมีอยู่แล้วในเครื่องที่ 1 และเครื่องแคช ทำให้เครื่องที่ 2 ทำการคัดไฟล์จากทั้ง 2 เครื่อง แต่เนื่องจากเครื่องแคชมีการทำงานหลายอย่างพร้อมๆกัน จึงทำให้การคัดไฟล์จากเครื่องที่ 1 ทำได้มากกว่า ทำให้ข้อมูลขาออกจากเครื่องแคชในการดาวน์โหลดครั้งต่อมา ลดลงจากครั้งแรกอย่างมาก

ตอนที่ 2 ทดลองเปรียบเทียบปริมาณข้อมูลขาเข้าและขาออกเมื่อปริมาณเครื่องที่ทำการดาวน์โหลดมีมากขึ้น โดยเป็นการดาวน์โหลดแบบปกติ

ลบข้อมูลเก่าในเครื่องทั้งสามเครื่อง เริ่มโปรแกรม Ntop เมื่อพร้อมที่จะทดสอบ ดาวน์โหลดไฟล์จากเครื่องภายนอกโดยตรงเพื่อทำงานแบบไม่ผ่านแคช แล้วเก็บข้อมูลการใช้งานเครือข่ายเมื่อการดาวน์โหลดเสร็จสมบูรณ์

ผลการทดสอบ

เครื่อง	ปริมาณข้อมูลขาออก	ปริมาณข้อมูลขาเข้า
1	95.1 MB	95.2 MB
2	170.6 MB	170.6 MB
3	263.7 MB	236.9 MB
4	318 MB	318.2 MB

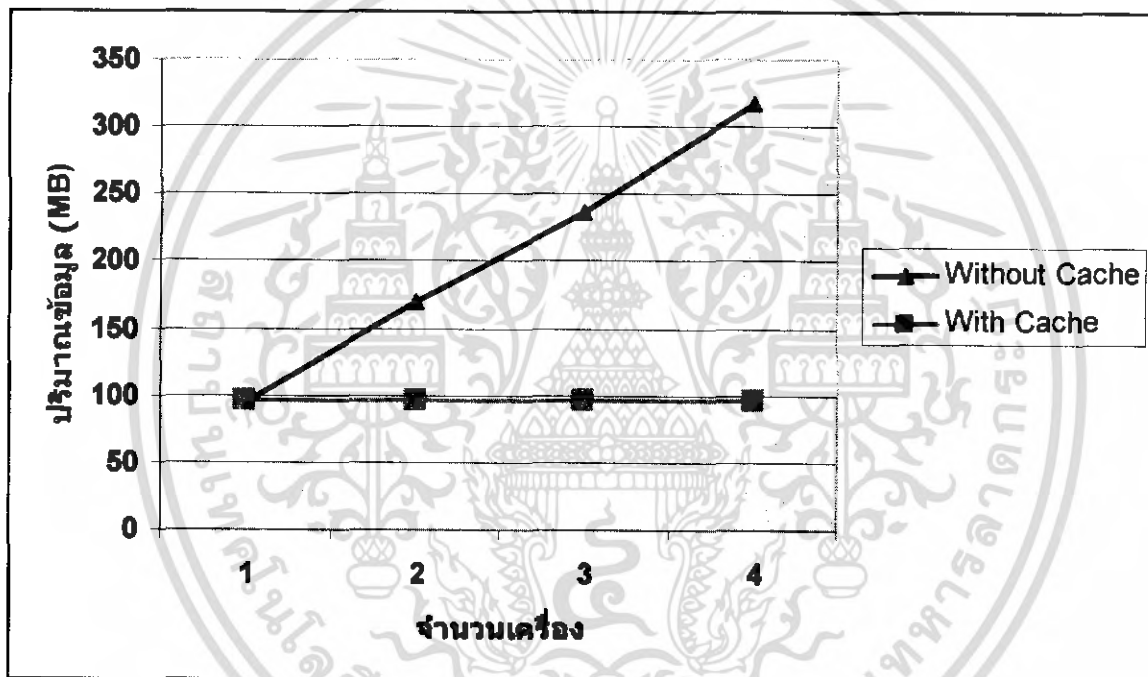
ตารางที่ 5.2 ปริมาณข้อมูลขาเข้าและขาออกจากเครื่องแคช โดยไฟล์ข้อมูลมีขนาด 92 MB

จากตารางที่ 5.1 และ 5.2 ความแตกต่างที่เกิดจากการโหลดในแบบไม่ใช้แคชกับใช้ระบบแคช คือจำนวนการใช้งานเครือข่ายทั้งขาเข้าและขาออก มีขนาดประมาณสองเท่าของขนาดไฟล์ที่โหลดจริง ซึ่งมาจากการโหลดไฟล์ของเครื่องทั้งสองเครื่องแยกกัน ไม่เหมือนในตอนหนึ่งที่เครื่องแคชเครื่องเดียวคัดออกภายนอก ซึ่งขนาดการรับส่งไฟล์ มากกว่าขนาดของไฟล์จริงเพียงเล็กน้อย

จากตารางที่ 5.1 จะเห็นได้ว่าจำนวนเครื่องที่มากขึ้นนั้น แทบจะไม่ส่งผลกระทบต่อการใช้งานเครือข่ายเลยเมื่อทำงานภายใต้ระบบแคช ที่เป็นเช่นนี้ก็เพราะเครื่อง Internal Peer ของระบบจะเป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

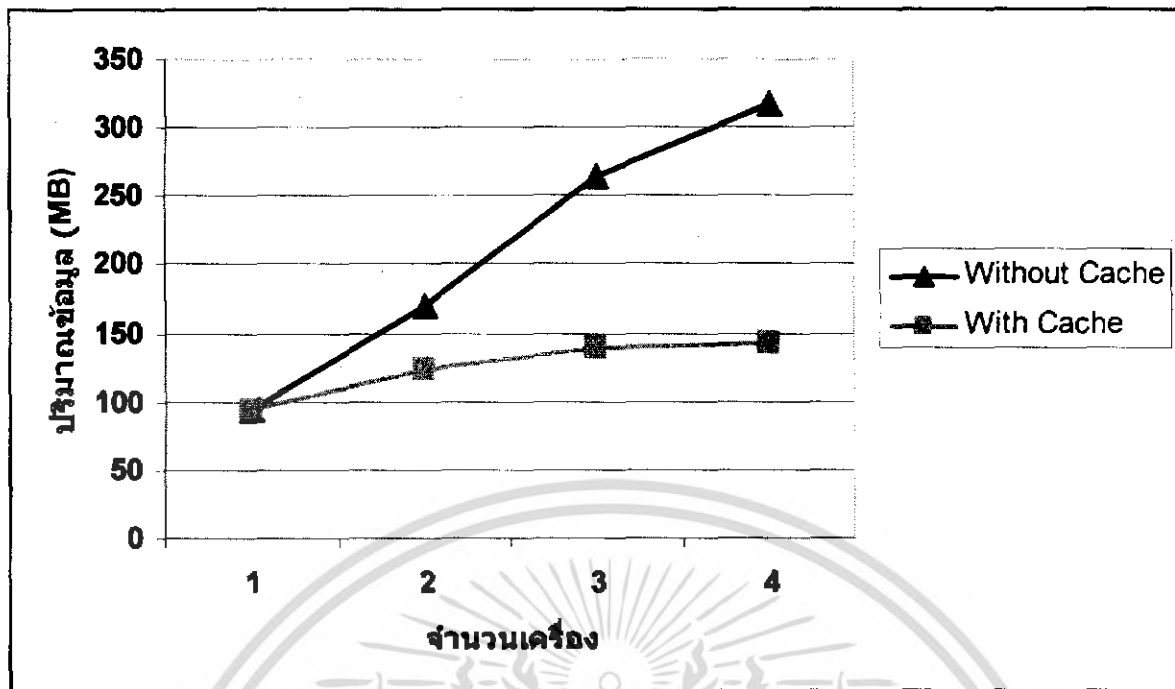
เครื่องเดียวที่ติดต่อออกนอกเครือข่าย ทำให้ขนาดการส่งและรับข้อมูล มากกว่าขนาดของไฟล์เพียงเล็กน้อย (จากความคลาดเคลื่อนที่มาจาก Protocol ต่างๆของเครือข่าย) ซึ่งเมื่อเปรียบเทียบกับการทำงานแบบปกติของ BitTorrent จะเห็นว่าการใช้งานเครือข่าย โดยเฉพาะขนาดของข้อมูลขาเข้า มีขนาดใกล้เคียงกับผลคูณของขนาดของไฟล์ที่โหลดกับจำนวนเครื่อง ซึ่งแสดงให้เห็นอย่างชัดเจนว่าระบบแคชสามารถลดการใช้งานเครือข่ายได้มากขนาดไหนเมื่อนำมาเปรียบเทียบกัน

ปริมาณข้อมูลขาเข้าและขาออกจากการดาวน์โหลดของเครื่องภายใน โดยเปรียบเทียบระหว่างการดาวน์โหลดผ่านระบบแคชและการดาวน์โหลดแบบปกติสามารถเห็นความแตกต่างได้อย่างชัดเจนจากกราฟดังรูป 5.18 และ 5.19 ตามลำดับ



รูปที่ 5.18 แผนภูมิเปรียบเทียบปริมาณข้อมูลขาเข้าจากการดาวน์โหลดของเครื่องภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.19 แผนภูมิเปรียบเทียบปริมาณข้อมูลออกจากการดาวน์โหลดของเครื่องภายใน

5.3.2 การทดลองที่ 2 เปรียบเทียบเวลาที่ใช้ในการดาวน์โหลดข้อมูลระหว่างการดาวน์โหลดผ่านระบบแคชและการดาวน์โหลดแบบปกติ

ตอนที่ 1 การโหลดโดยผ่านระบบแคช

เครื่องภายในเครือข่าย 4 เครื่อง ทำการดาวน์โหลดไฟล์ .torrent โดยทำการดาวน์โหลดข้อมูลต่อกัน

ผลการทดสอบ

เครื่อง	เวลา
1	3 min 18 sec
2	24 sec
3	20 sec
4	16 sec

ตารางที่ 5.3 ดาวน์โหลดโดยผ่านระบบแคช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองสังเกตเครื่องแรกที่ทำการดาวน์โหลดจะใช้เวลามากกว่าเครื่องอื่นอย่างมาก ส่วนเครื่องที่ 2 , 3 และ 4 ทำการดาวน์โหลด เวลาที่ใช้ในการดาวน์โหลดจะลดลงอย่างมาก เนื่องจากการดาวน์โหลดของเครื่องแรกนั้นทำการดูไฟล์จากเครื่องแคชเพียงเครื่องเดียวเท่านั้น แต่เครื่องที่ 2 จะทำการดาวน์โหลดจากเครื่องแคชและเครื่องที่ 1 และเครื่องที่ 3 จะทำการดาวน์โหลดจากเครื่องแคช , เครื่องที่ 1 , เครื่องที่ 2 ไปเรื่อยๆ ทำให้ยังมีการดาวน์โหลดมากขึ้นเวลาที่ใช้ก็จะยิ่งน้อยลง

ตอนที่ 2 การดาวน์โหลดแบบปกติ

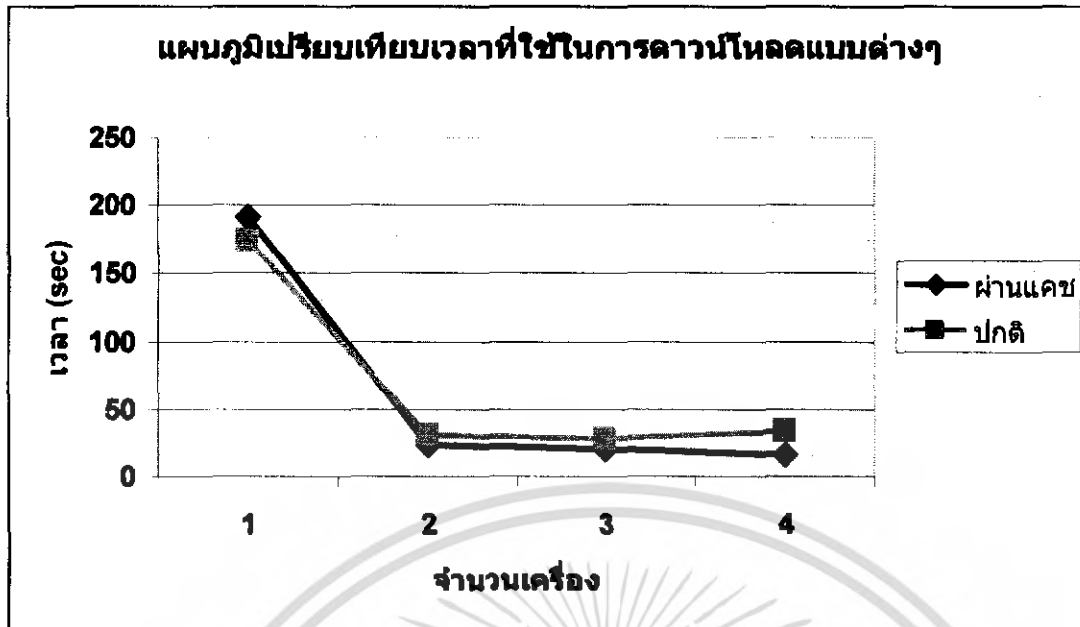
ผลการทดสอบ

เครื่อง	เวลา
1	2 min 54 sec
2	30 sec
3	28 sec
4	24 sec

ตารางที่ 5.4 คำนวณโดยไม่ผ่านระบบแคช

จากตารางที่ 5.3 และ 5.4 จะเห็นว่าเวลาที่ใช้ในการดาวน์โหลดเครื่องแรกแบบผ่านระบบแคชจะมากกว่าแบบปกติเพราะเครื่องแคชต้องทำงานหลายๆอย่างพร้อมๆกันทำให้การดูไฟล์จากเครื่องแคชใช้เวลามากกว่าการทำงานปกติ แต่ในเครื่องต่อมาเวลาที่ใช้จะพอกัน ดังกราฟในรูปที่ 5.20 แต่จากการทดลองที่ 1 จะเห็นว่าถึงแม้เวลาที่ใช้ในการดาวน์โหลดจะพอกัน แต่ปริมาณข้อมูลที่วิ่งเข้าออกจากเครือข่ายมีปริมาณน้อยกว่ามากเมื่อผ่านระบบแคช

เมื่อเปรียบเทียบกรณีต่างๆ เฉพาะการทำงานของเครื่องภายในจะพบว่า ผลกระทบที่เกิดจากการนำระบบแคชมาใช้ มีผลเล็กน้อยมาก ซึ่งมาจากการติดต่อทั้งโหลดจากเครื่องภายนอก และการส่งให้เครื่องภายใน ทำให้การโหลดซ้ำในการ โหลดพร้อมกับเครื่องภายใน แต่การโหลดซ้ำนั้นแทบจะไม่ส่งผลใดๆเลย สำหรับวิธีลดปัญหาการโหลดซ้ำในการโหลดพร้อมกันกับเครื่องภายในสามารถทำได้โดยเพิ่มจำนวนเครื่องที่เป็นแคช เพื่อลดภาระในการโหลดงานของเครื่องแคช



รูปที่ 5.20 แผนภูมิเปรียบเทียบเวลาที่ใช้ในการดาวน์โหลดของเครื่องภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ปัญหาที่พบระหว่างการทดลองและแนวทางแก้ไข

ปัญหา DHT Function ที่ทำให้ Peer สามารถค้นพบ Peer ตัวอื่นได้โดยไม่ต้องพึ่งแทรคเกอร์ เนื่องจากสภาพการทดสอบแตกต่างจากสภาพจริงคือตำแหน่งการวางเครื่องในเครือข่ายในสภาพการทำงานจริง จะมี Firewall และ Gateway อยู่ ซึ่งสามารถหยุดการค้นหา DHT ได้โดยการปิด UDP Packet ที่ส่งออกนอกเครือข่าย ดังนั้นปัญหานี้ในสภาพการทดสอบ แก้ไขโดยการปิด DHT Function ของตัวเครื่องภายนอกออก รวมถึงสร้างไฟล์ Metadata ที่ไม่รองรับ DHT ข้อแตกต่างระหว่าง Metadata ที่รองรับ DHT และไม่รองรับ DHT คือ แบบที่รองรับ DHT จะมีไอพีแอดเดรสของ DHT Nodes พ่วงที่ท้ายไฟล์หลัง Info field ซึ่งสามารถให้ระบบแคช จัดการทิ้งไปได้

ปัญหา Endgame Download Speed ที่ทำให้การไหลลื่นช้าลงที่ 99% พบในโปรแกรมตัวที่ใช้เป็นแคช ทำให้ผลการทดสอบช่วงแรกๆ มีความคลาดเคลื่อน สภาพการทดสอบนั้น มีข้อจำกัดก็คือมี Seeder แค่ตัวเดียว ทำให้ปัญหานี้เกิดขึ้นเป็นประจำ เพราะการทำงาน Endgame คือ การลดอัตราการดาวน์โหลด แล้วส่ง Request ของชิ้นส่วนสุดท้ายไปยังทุกๆ Peer ที่มี แล้วเมื่อได้รับชิ้นส่วนมาจาก Peer ใดๆ ก็ส่ง Request Cancel ไปยัง Peer ตัวอื่นๆ การที่มี Seeder ตัวเดียว ทำให้มีโอกาสที่ Seeder จะรอให้ติดต่อขอจาก Peer ตัวอื่นแทน แต่ในสภาพจริง โอกาสที่งานไหลลื่นบางงานจะมี Seeder แค่ตัวเดียวก็เกิดขึ้นได้ จึงต้องแก้ไขที่โค้ดของโปรแกรมให้ทำงาน Endgame เมื่อเลขจุดที่ควรจะเริ่มไปสั๊กพัก แล้วนำมาทดสอบใหม่

5.5 เปรียบเทียบสภาพการทดลองและสภาพการทำงานจริง

เนื่องจากมีความแตกต่างของสภาพที่ใช้ทดลองกับสภาพการทำงานจริง จำเป็นต้องนำผลการทดลองและข้อที่ต้องคำนึงถึงในสภาพที่ต่างกันมาใช้เพื่อนำไปประยุกต์ เพื่อให้รู้ว่า ในการใช้งานจริง ระบบนี้จะมีประสิทธิภาพเช่นใด จะพบปัญหาใดได้บ้าง และระบบเก่ากับระบบแคะจะจัดการกับปัญหาได้อย่างไร

5.5.1 ปัญหาที่ไม่พบในสภาพการทดลอง

การล่มของแทรคเกอร์

ปัญหาการล่มของแทรคเกอร์นั้น พบได้บ่อยมากในสภาพการทำงานจริง ซึ่งในสภาพจริงนั้น ในระบบปัจจุบันมีการใช้งาน DHT Trackerless Torrent ซึ่งใช้ UDP Port ค้นหา Node และ Peer ตัวอื่นๆ แต่บางครั้ง DHT ก็ไม่ได้รับการสนับสนุนในโปรแกรมที่ต่างกันไป เช่น Azureus ใช้ DHT ที่ไม่เหมือนกับ BitComet และทั้งสองโปรแกรมก็เป็นที่นิยมใช้เช่นกัน ส่วนระบบแคะนั้น ปัญหาการล่มของแทรคเกอร์ภายใน ขึ้นอยู่กับเครื่องที่ทำงานเป็นแทรคเกอร์ ซึ่งหากระบบเครือข่ายภายในไม่ล่ม และไม่มีปัญหาด้าน ไฟฟ้าหรืออุปกรณ์แทรคเกอร์ภายในก็มีโอกาสล่มน้อยมาก ส่วนการติดต่อกับแทรคเกอร์ภายนอกของ Internal Peer จะเป็นปัญหาที่แก้ไขได้ยาก เนื่องจาก Internal Peer จะถูกปิดการติดต่อ DHT ไว้เพื่อไม่ให้ส่ง Peers Table ไปให้เครื่องในเครือข่าย แต่ทั้งนี้ ระบบแคะยังสามารถปล่อยไฟล์ที่เครื่อง Internal Peer โหลดมาไว้ได้แม้จะไม่สามารถโหลดต่อได้ขณะที่แทรคเกอร์ภายนอกล่ม ไม่มี Seeder ที่สมบูรณ์

สภาพไม่มี Seeder ที่สมบูรณ์

ปัญหาสภาพไม่มี Seeder ที่สมบูรณ์ในสภาพจริงนั้นเกิดขึ้นบ่อยมาก โดยเฉพาะในไฟล์ขนาดใหญ่ หรือไฟล์ที่ไม่เป็นที่นิยม ซึ่งเครื่องที่ปล่อยไฟล์เป็นเครื่องแรกอาจปล่อยไม่สมบูรณ์ หรือไม่มีเครื่องอื่นที่โหลด โดยสมบูรณ์ก่อนที่เครื่องเครื่องนั้นจะหยุดแจกจ่ายไฟล์ นอกจากนี้กรณีที่เครื่องที่โหลดจนเกือบสมบูรณ์เกิดปัญหา Endgame ก็จะมีเพียง 99% จึงไม่สามารถทำให้การโหลดเสร็จสิ้นได้ รวมถึงทำให้เครื่องอื่นๆเจอปัญหาเดียวกัน ทางแก้ของการทำงานปกติ คือพยายามเปิดเครื่องไว้นานกว่าไฟล์จะโหลดเสร็จ ซึ่งสิ้นเปลืองพลังงาน ระบบแคะที่ออกแบบมาไม่สามารถแก้ไข ปัญหาที่เครื่องทั่วไปอาจพบเจอได้ แต่ในทางกลับกัน ระบบสามารถทำงานได้ตลอดเวลา (เนื่องจากถือเป็นส่วนหนึ่งของระบบเครือข่าย) ทำให้สามารถโหลดได้ทันทีที่มี Seeder ใหม่เข้ามาแจกจ่ายไฟล์ นอกจากนี้เครื่องภายในเครือข่ายไม่จำเป็นต้องเปิดไว้ตลอดเวลาเพื่อหา Seeder เนื่องจากระบบจะหาและโหลดมารอไว้ให้ ทำให้สามารถประหยัดการใช้งานได้ด้วย และเมื่อไฟล์ในระบบแคะสมบูรณ์ เครื่องภายในเครือข่ายจะไม่พบสภาพไม่มี Seeder สมบูรณ์สำหรับไฟล์ไฟล์ดังกล่าวเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทวิจารณ์และสรุป

6.1 บทสรุป

ในทุกวันนี้การใช้งานโปรแกรมบิตทอเรนทเพื่อดาวน์โหลดข้อมูลระหว่างกันแบบ Peer-to-Peer เป็นที่นิยมใช้กันอย่างแพร่หลาย โดยสังเกตได้จากจำนวนเว็บไซต์ที่เป็นแทรคเกอร์ที่มีอยู่เป็นจำนวนมากมาย และมีการเปิดขึ้นอย่างต่อเนื่อง รวมไปถึงตัวโปรแกรม Bittorrent Client ที่เป็นโปรแกรมสำหรับการใช้งานบิตทอเรนทนั้นก็มีผู้ผลิตออกมาหลายค่าย หลากหลายรูปแบบให้ได้ใช้งานกัน ซึ่งกลุ่มผู้ใช้งานโปรแกรมบิตทอเรนทนั้นก็มีความหลากหลาย ตั้งแต่ นักเรียน นักศึกษา รวมไปถึงคนวัยทำงาน ซึ่งเหตุผลในการใช้งานก็แตกต่างกันไป จึงสามารถกล่าวได้ว่าแทบทุกหน่วยงานจะต้องมีคนใช้งานโปรแกรมบิตทอเรนท

ดังที่ได้กล่าวมาแล้วว่าการเชื่อมต่อ Peer-to-Peer ด้วยโปรโตคอลบิตทอเรนทนั้น จะเป็นการเชื่อมต่อแบบหนึ่งต่อหนึ่ง คือ หนึ่งงานที่รันอยู่ต่อหนึ่ง Peer ที่เชื่อมต่อด้วย ยิ่งไฟล์ๆหนึ่งมี Peer มากๆ ก็จะเกิดการเชื่อมต่อมากมาย แล้วยังมีผู้ใช้งานโปรแกรมบิตทอเรนทพร้อมๆกันที่หลายๆคน ก็จะทำให้เกิดการเชื่อมต่อของบิตทอเรนทอันมากมายมหาศาล

ถ้ามีการนำระบบแคชสำหรับโปรแกรมบิตทอเรนทมาใช้งานนั้น ก็จะทำให้การเชื่อมต่อของการใช้โปรแกรมบิตทอเรนทจากภายในเครือข่ายที่ออกไปลดน้อยลง กล่าวคือจะเป็นการเชื่อมต่อของตัว Internal Peer ของเราเท่านั้น นอกจากนี้เมื่อมีการดาวน์โหลดไฟล์ที่ซ้ำกับไฟล์ที่มีอยู่ในระบบแคชก็จะทำให้ผู้ใช้งานดาวน์โหลดไฟล์โดยตรงเลย ช่วยลดเวลาในการดาวน์โหลดได้

6.2 วิจารณ์สิ่งที่ได้จากโครงการ

ในโครงการนี้นั้นสามารถทำในส่วนของการใช้งานได้อย่างสมบูรณ์ เมื่อเทียบกับโครงการเดียวกันที่พัฒนาจากปีที่แล้ว ซึ่งสามารถทำได้ทั้งในส่วนที่ดาวน์โหลด และอัปโหลด โดยที่สามารถพิสูจน์ได้ทั้งทางทฤษฎีและทางปฏิบัติ รวมถึงได้รู้เกี่ยวกับทฤษฎีการทำงานพื้นฐานของโปรโตคอลบิตทอเรนท รวมถึงการทำงานทางด้านอื่นๆ ทางติดตั้งและควบคุมโปรแกรมบิตทอเรนทให้ได้ประสิทธิภาพสูงสุด

ได้เรียนรู้ถึงการติดตั้งโปรแกรมต่างๆ อาทิเช่นพร็อกซีเซิร์ฟเวอร์ หรือไฟร์วอลล์ โปรแกรมบิตทอเรนทบนระบบปฏิบัติการลินุกซ์ และสามารถปรับแต่งระบบดังกล่าวให้สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ รวมถึงการเขียนโปรแกรมและสคริปเพื่อควบคุมให้บิตทอเรนทแคช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานได้อย่างเป็นระบบ รวมถึงการติดตั้งบิตทอเรนทเซิร์ฟเวอร์ให้เหมาะกับการใช้บิตทอเรนทเซิร์ฟเวอร์ในปัจจุบัน เช่นมีการรายงานค่า ratio หรือระบบสมาชิกต่างๆ

นอกจากนั้นยังได้ทดลองต่างๆเพื่อตรวจสอบประสิทธิภาพการใช้งานของระบบ เพื่อสามารถพิสูจน์ได้ว่า ระบบบิตทอเรนทแสดที่สามารถขึ้นมานั้นสามารถบรรลุถึงวัตถุประสงค์ได้อย่างถูกต้อง

สุดท้ายได้เรียนรู้ถึงปัญหาต่างๆ เช่น โปรแกรมบิตทอเรนทบางตัว มีปัญหา Endgame Download Speed ซึ่งทำให้การโหลดไฟล์ ชะงักที่ 99% ไปเรื่อยๆ มักจะเกิดขึ้นกรณีที่ไฟล์ที่แลกเปลี่ยนนั้น มี Seeder ไม่กี่เครื่อง ซึ่งไม่พอเพียงพอที่จะทำงานแบบ Endgame ได้จนกว่าจะมีการค้นพบ Peer ตัวอื่นที่สมบูรณ์แล้วเช่นกัน แล้วติดต่อขอชิ้นส่วนสุดท้ายมาได้ นอกจากนี้ยังสามารถแก้ไขได้โดยการติดต่องานนี้ใหม่อีกครั้ง Endgame Download โดยปกติ คือการชะลอการโหลด แล้วเรียก Request ชิ้นสุดท้ายไปยัง Peer ทุก Peer ที่มีชิ้นนี้ แล้วส่งยกเลิกตามไปที่ peer อื่นๆเมื่อได้รับชิ้นส่วนนี้แล้วจาก Peer ตัวใดๆ

6.3 ปัญหาอุปสรรคและแนวทางการแก้ไข

ในช่วงระหว่างการค้าเนินโครงการระบบแสดสำหรับ โปรแกรมบิตทอเรนทนั้น ได้ประสบปัญหาต่างๆ ซึ่งได้รวบรวมมาเป็นข้อๆ ดังนี้

1. การใช้ระบบปฏิบัติการลินุกซ์ในการทำโครงการนั้น ประสบปัญหาที่สำคัญคือผู้ใช้ไม่มีความรู้ ความชำนาญในการติดตั้ง โปรแกรมบนลินุกซ์ รวมไปถึงการเลือกว่าจะใช้ลินุกซ์ตระกูลไหน รุ่นอะไร Kernel อะไรมาใช้สำหรับการทำโครงการนี้ เพราะการใช้ลินุกซ์เวอร์ชันเก่าทำให้บางครั้งการลง โปรแกรมใหม่จะเกิดปัญหาเรื่อง library ซึ่งแนวทางการแก้ไขปัญหานี้ก็คือ หาคู่มือการลงโปรแกรมแต่ละ โปรแกรม และลงลินุกซ์เวอร์ชันใหม่ๆหรืออัปเดต Kernel เสมอๆ
2. ในการเลือกใช้โปรแกรมบิตทอเรนท พบว่าโปรแกรมบิตทอเรนทส่วนใหญ่จะเหมาะกับการทำงานบนระบบปฏิบัติการ Windows ส่วนโปรแกรมบิตทอเรนทที่สามารถทำงานบนระบบปฏิบัติการลินุกซ์ได้นั้น บาง โปรแกรมก็เป็นเวอร์ชันที่อยู่ระหว่างพัฒนา หรือไม่ก็ยังติดปัญหาอยู่ ซึ่งแนวทางการแก้ไขปัญหานี้คือ อาจจะต้องพัฒนาโปรแกรมบิตทอเรนทขึ้นมาให้งานเอง เพื่อที่จะได้ตรงกับความต้องการของระบบที่สุด
3. ในการเขียนเว็บเซอร์วิสนั้น เนื่องจากผู้เขียนขาดความรู้ และประสบการณ์ในการเขียนด้วย PHP ทำให้บางครั้งมีปัญหาในการเขียน เช่น การส่งให้รัน โปรแกรมบิตทอเรนทผ่านทางเว็บ หรือการเขียนเว็บเซอร์วิสจัดการข้อมูลที่อยู่ในแตรคเกอร์นั้น จำเป็นที่จะต้องเข้าใจโครงสร้างของฐานข้อมูลของแตรคเกอร์ที่ใช้ด้วย ซึ่งแนวทางการแก้ปัญหานี้คือ การศึกษาหาความรู้จากหนังสือ หรือเว็บไซต์ หรืออาจจะสอบถามจากผู้มีประสบการณ์ในด้านนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ในการติดตั้งโปรแกรมเพื่อมาใช้งานในระบบนั้น มีหลายครั้งที่วิธีการติดตั้งที่อยู่ในเว็บไซต์นั้นไม่ละเอียด หรือติดตั้งไปแล้วเกิดความผิดพลาด โดยที่ไม่มีวิธีแก้ไขบอกเอาไว้ ซึ่งแนวทางการแก้ปัญหาก็คือ การหาข้อมูลจากหลายๆแห่งมาเปรียบเทียบกัน หรือลองดูในกระทู้ตอบปัญหาของทางเว็บไซต์ของตัวโปรแกรม
5. ในการดักจับแพ็คเก็ตนั้นต้องทำการกั้นกรองในส่วนของ Header ของ HTTP Packet ซึ่ง Topdump ทำได้ค่อนข้างลำบาก ทำให้ต้องเขียนโปรแกรมในการดักจับแพ็คเก็ตขึ้นมาเองด้วย ภาษา Ruby ซึ่งต้องทำการศึกษากับภาษานี้ใหม่ทั้งหมด ซึ่งแนวทางแก้ไขปัญหาคือ การหาข้อมูลต่างๆ จากอินเทอร์เน็ตและสอบถามผู้มีความรู้

6.4 แนวทางการพัฒนาต่อ

สำหรับแนวทางในการพัฒนาต่อของโครงการระบบแคชสำหรับโปรแกรมบิตทอเรนที่นั้น ขอเสนอแนะแนวทางเป็นข้อๆดังนี้

1. พัฒนาในส่วนของโปรแกรมบิตทอเรนที่ให้มีประสิทธิภาพมากขึ้น ซึ่งอาจจะพัฒนาโดยการเขียนขึ้นเองทั้งหมด หรือการนำโค้ดที่มีอยู่ตามเว็บไซต์มาพัฒนาต่อให้มีความเหมาะสมกับตัวของระบบ
2. พัฒนาให้ระบบสามารถทำการ filter ในชั้นของ attachment ได้ ซึ่งจะทำให้สามารถใช้งานกับแทรคเกอร์แบบที่เป็น Forum ได้
3. พัฒนาให้ระบบมีกระบวนการเคลียร์ค่าข้อมูลที่ไม่ได้ใช้งานบ่อยๆ หรือหมดอายุการใช้งานที่อยู่ในแคชของระบบ เพื่อให้ระบบไม่เต็มไปด้วยข้อมูลที่ไม่ได้ใช้
4. พัฒนาในด้านความเร็วในการทำงานของระบบ โดยอาจจะเพิ่มจำนวน Internal Peer ให้มีจำนวนมากขึ้น เพื่อแบ่งเบาภาระในการทำงานของแต่ละตัว
5. พัฒนาในด้านการรักษาความปลอดภัยให้กับระบบ เพราะระบบที่ออกแบบมานี้ไม่ได้คำนึงถึงในด้านนี้ ซึ่งถ้าจะนำไปใช้งานจริงก็จำเป็นต้องคำนึงถึงด้านความปลอดภัยของทั้งตัวระบบ รวมไปถึงเครือข่ายภายใน

บรรณานุกรม

- [1] BitTorrent – Protocol,[Online] URL : <http://www.bittorrent.com/protocol.html>
- [2] Yahoo! Groups BitTorrent,[Online] URL :<http://groups.yahoo.com/group/BitTorrent/>
- [3] P2P Forums Index,[Online] URL : <http://www.p2pforums.com/index.php>
- [4] เลือกใช้ Proxy อย่างไรให้มีประสิทธิภาพ,[Online] URL :
http://www.sut.ac.th/ccs/news/tip_tech/tip002.asp
- [5] TransparentProxy,[Online] URL :
<http://phst.ph.mahidol.ac.th/Linux/html/transparentproxy.html>
- [6] Squid Proxy Caching Server,[Online] URL :
http://micro.se-ed.com/content/mc205/MC205_181.asp
- [7] Squid Web Proxy Cache,[Online] URL : <http://www.squid-cache.org>
- [8] Squidguard – An ultrafast and free filter/redirector/access controller for Squid,[Online] URL
: <http://www.squidguard.org/>
- [9] Wiki.Theory.Org - Bit Torrent Specification,[Online] URL :
<http://wiki.theory.org/BitTorrentSpecification>
- [10] Python Programming Language,[Online] URL : <http://www.python.org>
- [11] Fedora Project,sponsored by Red Hat,[Online] URL : <http://fedora.redhat.com>
- [12] เว็บเซอร์วิส คือ อะไร?,[Online] URL : <http://chatpong.exteen.com/20050814/entry-1>
- [13] Web Services เครื่องมือธุรกิจยุคใหม่,[Online] URL :
http://www.wsiam.com/document/WebServices_BusinessTool.pdf
- [14] สุวัฒน์ ปุณณชัยยะ, ต้น ต้นท์สุทธิวงษ์ และ สุพจน์ ปุณณชัยยะ 2545 เปิดโลก TCP/IP และ
โปรโตคอลของอินเทอร์เน็ต Second Edition สำนักพิมพ์ Provision
- [15] ก่อกิจ วีระอาชากุล 2545 คิดตั้งและปรับแต่งเซิร์ฟเวอร์ Linux สำหรับ Admin Linux โดยเฉพาะ
Third Edition สำนักพิมพ์ Infopress Developer Book
- [16] กิตติภูมิ วรรณิตร 2543 php เปลี่ยนวิถีสู่การสร้างโฮมเพจอย่างมือโปร พิมพ์ครั้งที่ 1 สำนักพิมพ์
บริษัท วิตตี้ กรุ๊ป จำกัด
- [17] นราวุธ พลับประสิทธิ์ 2546 php เปลี่ยนวิถีสู่การสร้างโฮมเพจอย่างมือโปร ชั้นที่2 พิมพ์ครั้งที่ 1
สำนักพิมพ์ บริษัท วิตตี้ กรุ๊ป จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.

ขั้นตอนการติดตั้งระบบแคชสำหรับ โปรแกรมบิตทอเรนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ขั้นตอนการติดตั้งระบบแคชสำหรับโปรแกรมบิตทอเรนท

ระบบแคชสำหรับโปรแกรมบิตทอเรนทนี้ประกอบด้วยองค์ประกอบที่สำคัญของระบบ ได้แก่ Proxy Server, Internal Tracker, Internal Peer, Packet Filter และ Web Service ซึ่งมีวิธีการติดตั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ของแต่ละองค์ประกอบ ดังนี้

1. Proxy Server

ความต้องการทางด้านฮาร์ดแวร์

- CPU: ความเร็วอย่างต่ำ 1 GHz
- Hard disk: อย่างต่ำ 20 GB
- RAM: อย่างต่ำ 256 MB
- Network Interface Card: รองรับความเร็วอย่างต่ำ 100 Mbps

ความต้องการทางด้านซอฟต์แวร์

- ระบบปฏิบัติการ: UNIX, Linux
- Proxy Server: Squid เวอร์ชัน 2.4 ขึ้นไป
- Berkeley DB: Berkeley DB เวอร์ชัน 2.x.x เท่านั้น
- SquidGuard: SquidGuard เวอร์ชัน 1.2

Squid Web Proxy Cache - Windows Internet Explorer

http://www.squid-cache.org

Squid Web Proxy Cache

Documentation
[FAQ](#) · [Wiki](#)
[Configuration Guide](#)
[Users Guide](#)
[Home/English](#)
[Addresses](#) [More...](#)

Versions
[FFAME 1.1](#)
[DEV 1.1](#)
[Older Versions](#)


Mirror Sites
[FTP files](#)
[Web site](#)

Support
[Bugzilla Database](#)
[Mailing lists](#)
[Contacting us](#)
[Commercial services](#)
[Project Sponsors](#)
[Squid-based products](#)
[Donations](#)

Related
[iSCache project](#)
[Logfile Analyser](#)
[Software](#)
[Weblogs](#)
[Binary distributions](#)

Misc
[Team Squid](#)

www.squid-cache.org
Squid Web Proxy Cache



Squid is...

- a full-featured web proxy cache
- free, open-source software
- the result of many contributions by unpaid (and paid) volunteers

Squid supports...

- proxying and caching of HTTP, FTP, and other URLs
- proxying for SSL
- cache hierarchies
- ICP, HTCP, CARF, Cache Digests
- transparent caching
- WCCP (Squid v2.3 and above)

Done

รูปที่ 1ก ภาพตัวอย่างหน้าเว็บไซต์ของ Squid

การติดตั้งโปรแกรม Squid

Squid เป็นตัวที่ใช้ในการสร้างแคชหรือว่าพร็อกซี่ เพื่อเก็บเว็บต่างๆ โดยปกติถ้าใช้ระบบปฏิบัติการ Fedora ตั้งแต่ Core 4 ขึ้นมาจะมีการติดตั้ง Squid ไว้ให้อยู่แล้ว แต่ต้องทำการตั้งค่าต่างๆ ตามขั้นตอนดังนี้

- ทำการแก้ไขไฟล์ squid.conf ที่ /etc/squid/squid.conf ด้วยคำสั่ง
`# vi /etc/squid/squid.conf`

แล้วทำการแก้ไขส่วนต่างๆของไฟล์ดังนี้

```
http_port 8080
icp_port 3130
hierarchy_stoplister cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 96 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cache_dir diskd /cache 100 16 256
cache_store_log none
acl localhost src 127.0.0.1/255.255.255.255
acl localnet src 161.246.5.0/255.255.255.0
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localhost
http_access allow localnet
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
cache_mgr root
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on

```

- สั่งให้ Squid ทำงาน ด้วยคำสั่ง

```
[root@one root]# service squid start
```

```
Starting squid: [ OK ]
```

แต่ถ้ายังไม่ได้ติดตั้งให้ดาวน์โหลดตัวโปรแกรม Squid ที่เป็น RPM มาจากเว็บไซต์ของ Squid (website: <http://www.squid-cache.org>) จากนั้นให้ทำการติดตั้งตามขั้นตอนดังนี้

- ใช้คำสั่ง rpm ติดตั้ง package แบบ rpm

```
[root@one root]# rpm -Uvh squid-2.4.STABLE6-1.7.2.i386.rpm
```

```
Preparing... #####
```

```
[100%]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
l:squid #####
[100%]
```

- จากนั้นทำการแก้ไขไฟล์ squid.conf ที่ /etc/squid/squid.conf ดังที่กล่าวไว้ข้างต้น
- จากนั้นทำการเปลี่ยน permission และ owner ของไฟล์เคอร์ /var/spool/squid

```
[root@one root]# cd
```

```
[root@one root]# chmod 750 /var/spool/squid
```

```
[root@one root]# chown squid.squid /var/spool/squid
```

- จากนั้นทำการสร้าง swap directory โดยใช้คำสั่ง

```
[root@one root]# squid -z
```

```
2002/08/12 18:50:12| Creating Swap Directories
```

- สุดท้ายสั่งให้ squid ทำงาน

```
[root@one root]# service squid start
```

```
Starting squid:
```

การทำ Transparent Proxy

เป็นการบังคับให้เครื่องลูกต้องผ่าน Proxy Server โดยที่ไม่ต้องไป set proxy ที่ตัว Browser ทำโดยเพิ่มเข้าไปใน squid.conf ดังนี้

```
httpd_accel_host virtual
```

```
httpd_accel_port 80
```

```
httpd_accel_with_proxy on
```

```
httpd_accel_uses_host_header on
```

```
visible_hostname 161.246.5.225
```

จากนั้นเพิ่ม iptables อีกบรรทัด ดังนี้

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080
```

```
[root@one root]# /etc/init.d/iptables save
```

```
[root@one root]# /etc/init.d/iptables restart
```

Flushing all current rules and user defined chains: [OK]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Clearing all current rules and user defined chains: [OK]
 Applying iptables firewall rules: [OK]
 [OK]

[root@onc root]#

สั่งให้ squid ทำงาน โดย /etc/inet.d/squid restart

[root@onc root]# /etc/init.d/squid restart

Stopping squid: [OK]

Starting squid: [OK]

[root@onc root]

เป็นการเสร็จสิ้นการทำ Transparent Proxy

หมายเหตุ

เครื่องที่เป็น Proxies Server จำเป็นจะต้องเซตให้ IPv4 forwarding เป็น 1 ด้วยเสมอ สามารถทำได้โดย echo "1" > /proc/sys/net/ipv4_forward วิธีนี้เป็นการแก้ไขเพียงชั่วคราวเท่านั้น เมื่อเปิดเครื่องใหม่ก็ต้องทำใหม่ทุกครั้ง ทำการแก้ไขโดยถาวรทำได้โดยแก้ไขไฟล์ configuration ที่ /etc/sysctl.conf แล้ว set ให้ net.ipv4.ip_forward = 1

การติดตั้งโปรแกรม squidGuard

squidGuard ทำงานร่วมกับ Squid Proxy เพื่อบล็อกการเข้าถึงเว็บไซต์ต่าง ๆ โดยการบล็อกสามารถบล็อกด้วยค่าของโดเมน IP Address หรือแม้แต่ keyword นอกจากนี้สามารถกำหนดได้ว่าจะบล็อกในช่วงเวลาไหนของแต่ละวันและสามารถกำหนดกลุ่มในการบล็อกได้

โปรแกรมที่ต้องติดตั้ง

1. Squid
2. BerkeleyDB
3. SquidGuard

การติดตั้งและการปรับแต่งค่าของ Squid

ดูได้จาก การติดตั้งและปรับแต่งค่าของ Squid ก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้ง BerkeleyDB

สามารถดาวน์โหลด BerkeleyDB และ Patch ได้จาก

- <http://www.sleepycat.com/update/snapshot/db-3.2.9.tar.gz>

- <http://www.sleepycat.com/update/3.2.9/patch.3.2.9.html>

โดยให้ดาวน์โหลดมาเก็บไว้ที่ /usr/local/src จากนั้นให้เข้าไปยังตำแหน่งที่ดาวน์โหลดไฟล์มาเก็บไว้แล้วใช้คำสั่งดังนี้

```
# cd /usr/local/src
# tar zxvf db-3.2.9.tar.gz
# cd /usr/local/src/db-3.2.9
# patch -p0 < patchdb.3.2.9-1
# patch -p0 < patchdb.3.2.9-2
# cd build_unix
# ./dist/configure
# make
# make install
# cd../..
# echo /usr/local/BerkeleyDB.3.2/lib >> /etc/ld.so.conf
# ldconfig บรรทัดนี้ถ้าไม่สั่งจะไม่สามารถ compile squidGuard ได้
```

การติดตั้ง squidGuard

สามารถดาวน์โหลดโปรแกรม squidGuard ได้จาก <http://www.squidguard.org> มาไว้ที่ /usr/local/src จากนั้นก็เข้าไปยังตำแหน่งที่โหลดไฟล์มาเก็บไว้แล้วใช้คำสั่งตามนี้

```
# cd /usr/local/src
# tar xvfz squidGuard-1.2.0.tar.gz
# cd /usr/local/src/squidGuard-1.2.0
# export LIBS="-lpthread"
# ./configure --with-db=/usr/local/BerkeleyDB.3.2
# make
# make install
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับแต่งค่าของ SquidGuard

หลังจากการคอมไพล์ SquidGuard แล้วก็ต้องทำการสร้างไฟล์ SquidGuard.conf ขึ้นมาเองดังนี้

```
# cd /usr/local/

# mkdir /usr/local/squidGuard

# vi squidGuard.conf เป็นการสร้างไฟล์ config ของ SquidGuard โดยเนื้อหาไฟล์สามารถจะเป็นได้หลาย ๆ แบบ ซึ่งในที่นี้จะทำการคอนฟิกเพื่อ redirect การเปิดไฟล์ .torrent ไปยังตัว web service ของเรา ซึ่งจะมีเนื้อหาของไฟล์เป็นดังนี้
```

```
logdir /usr/local/squidGuard/log
dbhome /usr/local/squidGuard/db
dest file {
    expressionlist block/files
}
acl {
    default {
        pass !file all
        redirect http://161.246.5.224/download.php?url=%u
    }
}
```

คำอธิบายเพิ่มเติม

เมื่อมีผู้ที่กำลังจะเปิดหรือดาวน์โหลดไฟล์ .torrent ตัว squidGuard ก็จะมีการ redirect ไปยัง url ที่เราตั้งไว้ โดยส่งค่าตัวแปร url ไปด้วย โดยที่ url=%u นั้น ตัว %u คือ url ที่เครื่อง client ต้องการจะใช้งาน

สร้างตำแหน่งเก็บ log file และสร้างฐานข้อมูล file ที่ไม่ต้องการให้เปิดหรือดาวน์โหลด

```
# cd /usr/local/squidGuard/
# mkdir log สร้างตำแหน่งเก็บ log
# mkdir db
# mkdir db/block สร้างตำแหน่งที่เก็บฐานข้อมูล file ที่ไม่ต้องการให้เปิดหรือดาวน์โหลด
# cd db/block
```

vi files สร้างไฟล์ที่ใช้เก็บค่าของนามสกุลไฟล์ที่ไม่ต้องการให้เข้าถึง ซึ่งมีเนื้อหาเป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(\.torrent*)

จากนั้นสร้างฐานข้อมูลจากไฟล์ files ที่ได้เตรียมไว้โดยใช้คำสั่ง

```
# /usr/local/bin/squidGuard -C all
```

```
# chown -R nobody:nobody /usr/local/squidGuard
```

จากนั้นแก้ไขไฟล์คอนฟิกของ squid

```
# vi /etc/squid/squid.conf
```

โดยแก้ไขไฟล์ในส่วนของ redirect_program และ redirect_children ให้เป็นดังนี้

```
redirect_program /usr/local/bin/squidGuard
```

```
redirect_children 10
```

จากนั้นให้ restart squid โดยใช้คำสั่งดังนี้

```
# service squid restart
```

หมายเหตุ ทุกครั้งที่แก้ไขไฟล์ files ต้องทำการเพิ่มเข้าไปในฐานข้อมูลด้วยทุกครั้ง โดยใช้คำสั่ง

```
# /usr/local/bin/squidGuard -C all
```

```
# chown -R nobody:nobody /usr/local/squidGuard
```

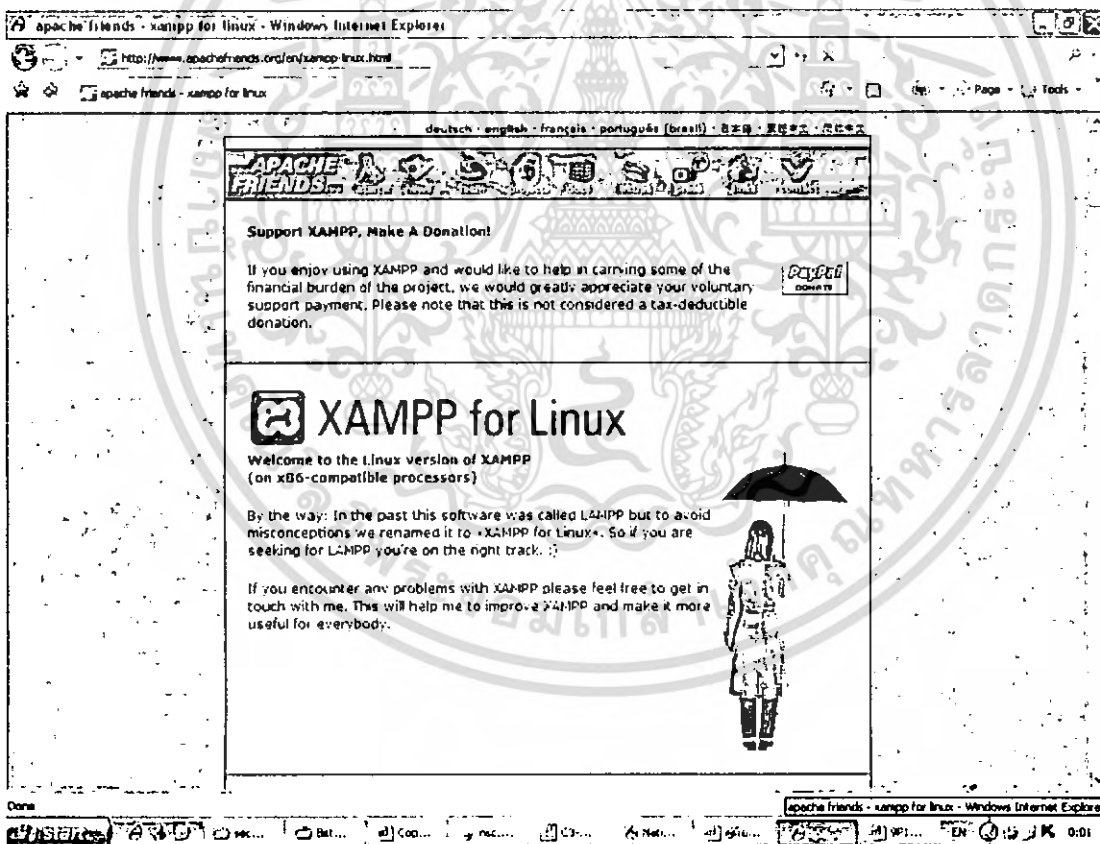
2. Internal Tracker

ความต้องการทางด้านฮาร์ดแวร์

- CPU: ความเร็วอย่างต่ำ 1 GHz
- Hard disk: อย่างต่ำ 20 GB
- RAM: อย่างต่ำ 256 MB
- Network Interface Card: รองรับความเร็วอย่างต่ำ 100 Mbps

ความต้องการทางด้านซอฟต์แวร์

- ระบบปฏิบัติการ: UNIX, Linux
- Web Server: XAMPP (Apache, MySQL, PHP, Perl)
- Virtual Torrent Server: phpBB2
- Virtual Tracker: Torrentpicr เวอร์ชัน 0.3.5



รูปที่ 2ก ภาพตัวอย่างหน้าเว็บไซต์ของ Apache-friend :XAMPP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งโปรแกรม XAMPP for Linux

XAMPP เป็นชุดโปรแกรมสำหรับการทำงานเป็น Web Server ที่สมบูรณ์ในตัว มีทั้ง Apache Web Service, MySQL Database, PHP และ Mail Sever หากยังไม่มี XAMPP for Linux (LAMP) ติดตั้ง สามารถโหลดได้จากเว็บไซต์ของ Apache Friends (<http://www.apachefriends.org/en/xampp-linux.html>) จากนั้นทำการติดตั้งตามขั้นตอน

- ทำการเปลี่ยน User เป็น system administrator ของระบบ โดยใช้คำสั่ง su
- แยกไฟล์ไปที่โฟลเดอร์ /opt โดยใช้คำสั่ง

```
# tar xvfz xampp-linux-1.5.1.tar.gz -C /opt
```

การเริ่มใช้งานโปรแกรม XAMPP for Linux ทำได้โดยใช้คำสั่งต่อไปนี้

```
# /opt/lampp/lampp start
```

เมื่อโปรแกรมทำงาน บรรทัดเหล่านี้จะปรากฏบนหน้าจอ

```
Starting XAMPP 1.5.1...
LAMP: Starting Apache...
LAMP: Starting MySQL...
LAMP started.
```

ทำการทดสอบ โดยเรียก Web Browser ขึ้นมา แล้วพิมพ์ <http://localhost/> ที่ช่อง Address หากทำงานได้สมบูรณ์ จะเข้าสู่หน้าจอต้อนรับของ XAMPP ทำการแก้ไขระบบความปลอดภัยต่างๆ โดยใช้คำสั่งต่อไปนี้

```
# /opt/lampp/lampp security แล้วทำการตั้งรหัสผ่านตามต้องการให้ปลอดภัย
```

การติดตั้งโปรแกรม phpBB

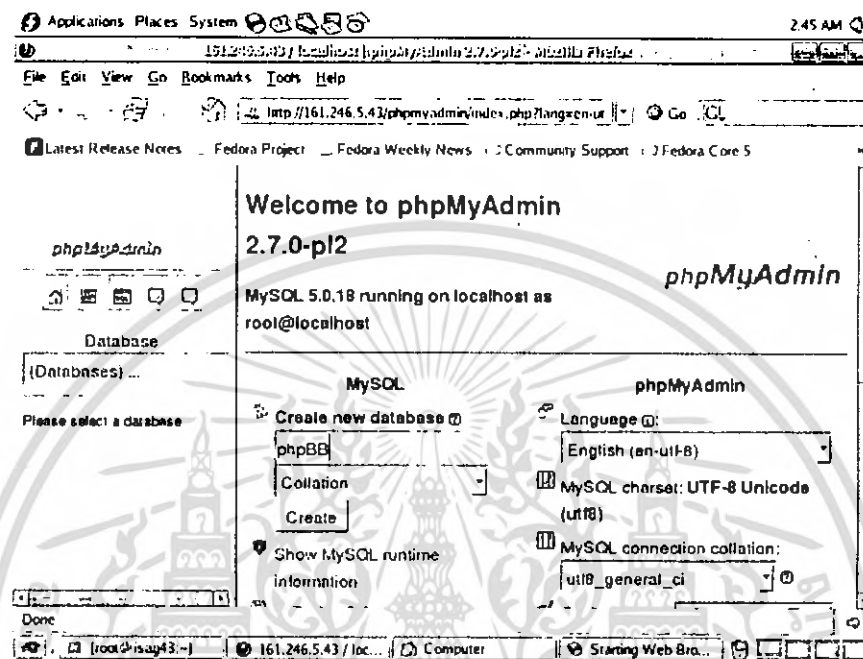
phpBB เป็นชุดการทำงาน Web Page สำเร็จรูปที่ใช้การทำงานของ PHP และ MySQL เป็นโปรแกรมที่จำเป็นต้องใช้ในการทำงานของ Internal Tracker ตัวติดตั้ง phpBB สามารถโหลดได้จากเว็บไซต์ของ phpBB (<http://www.phpbb.com/downloads.php>) การติดตั้ง ให้ทำตามขั้นตอนดังนี้

1. ทำการเปลี่ยน User เป็น system administrator ของระบบ โดยใช้คำสั่ง su
2. แยกไฟล์ไปที่โฟลเดอร์ /opt/lampp/htdocs โดยใช้คำสั่ง
tar xvfz xampp-linux-1.5.1.tar.gz -C /opt/lampp/htdocs
3. กรณีที่โปรแกรม LAMP ยังไม่ทำงาน ให้เรียกขึ้นมาทำงานโดยใช้คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/opt/lampp/lampp start

- เปิด Web Browser ขึ้นมา พิมพ์ URL ดังต่อไปนี้เข้าไปที่ช่อง Address
<http://localhost/phpmyadmin/>
- ที่ช่อง สร้างฐานข้อมูลใหม่ พิมพ์ phpBB แล้วคลิก สร้าง ดังรูป



รูปที่ 3ก ภาพตัวอย่างการสร้างฐานข้อมูลใหม่

- พิมพ์ URL ดังต่อไปนี้เข้าไปที่ช่อง Address
<http://localhost/phpBB2/>
- กรณีที่เป็นการใช้งานครั้งแรก จะเข้าสู่หน้าจอติดตั้ง กรอกข้อมูลลงในช่องดังต่อไปนี้

Database Type: MySQL 4.x/5.x

Choose your installation method: Install

Database Server Hostname / DSN: localhost (กรณีที่ใช้เครื่อง Database คนละเครื่อง ให้ใส่ Domain Name ของเครื่อง Database เช่น bitcachedb.kmitl.ac.th)

Your Database Name: phpBB (ให้ตรงกับที่สร้างไว้ในข้อ 5.)

Database Username: (ใส่ชื่อตามที่ตั้งไว้ตอนตั้งค่า Security ของ XAMPP)

Database Password: (ใส่รหัสผ่านตามที่ตั้งไว้ตอนตั้งค่า Security ของ XAMPP)

Administrator Username: (ตั้งชื่อผู้ดูแลสำหรับ phpBB)

Administrator Password & [Confirm]: (ตั้งรหัสผ่านของผู้ดูแล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Basic Configuration	
Default board language:	English ▾
Database Type:	MySQL 4.x/5.x ▾
Choose your installation method:	Install ▾
Database Configuration	
Database Server Hostname / DSN:	<input type="text" value="localhost"/>
Your Database Name:	<input type="text" value="phpBB"/>
Database Username:	<input type="text" value="root"/>
Database Password:	<input type="password" value="••••"/>
Prefix for tables in database:	<input type="text" value="phpbb_"/>
Admin Configuration	
Admin Email Address:	<input type="text"/>
Domain Name:	<input type="text" value="localhost"/>
Server Port:	<input type="text" value="80"/>
Script path:	<input type="text" value="/phpbb2/"/>
Administrator Username:	<input type="text" value="root"/>
Administrator Password:	<input type="password" value="••••"/>
Administrator Password [Confirm]:	<input type="password" value="••••"/>
<input type="button" value="Start Install"/>	

รูปที่ 4ก รูปตัวอย่างการติดตั้ง phpBB

8. เมื่อสำเร็จ จะเข้าสู่หน้าจอยืนยัน ให้โหลดไฟล์ config.php มาแล้วบันทึกแทนที่ลงในโฟลเดอร์ /opt/lampp/htdocs/phpBB2/ คลิก Finish Installation จะเตือนให้ลบโฟลเดอร์ install และ contrib ออก ทำการลบออกโดยใช้คำสั่งดังต่อไปนี้

```
rm -f -r /opt/lampp/htdocs/phpBB2/install
rm -f -r /opt/lampp/htdocs/phpBB2/contrib
```

9. ทำการเรียก <http://localhost/phpBB2/> จะเข้าสู่หน้าจอ log in ถือว่าติดตั้งได้สมบูรณ์

การติดตั้งโปรแกรม Torrent Pier

Torrent Pier เป็นส่วนของ Tracker ที่ใช้งานจริง โดยตัว Torrent Pier เป็นส่วนเสริมของ phpBB จึงต้องติดตั้งใน phpBB ไฟล์ Torrent Pier สามารถโหลดได้ที่เว็บไซต์ของ Torrent Pier (<http://www.torrentpier.com/>) จากนั้นติดตั้งดังนี้ต่อไปนี้

1. ทำการแยกไฟล์ไปที่ /opt/lampp/htdocs/phpBB2/ โดยใช้โปรแกรม Unzip
2. กรณีที่โปรแกรม LAMPP ยังไม่ทำงาน ให้เรียกขึ้นมาทำงาน โดยใช้คำสั่ง
/opt/lampp/lampp start
3. เปิด Web Browser ขึ้นมา พิมพ์ URL ดังต่อไปนี้เข้าไปที่ช่อง Address
<http://localhost/phpmyadmin/>
4. ที่ช่องสร้างฐานข้อมูลใหม่ พิมพ์ tracker แล้วคลิก สร้าง
5. ที่ Web Browser พิมพ์ <http://localhost/phpBB2/forum/> ที่ Address
6. จะเข้าสู่หน้าจอติดตั้งเหมือน phpBB กรอกข้อมูลเหมือนใน phpBB แต่ให้เปลี่ยนช่อง Your Database Name ให้เป็น tracker (หรือให้ตรงกับข้อ 4.)
7. เมื่อสำเร็จ จะเข้าสู่หน้าจอยืนยัน ให้โหลดไฟล์ config.php มาแล้วบันทึกแทนที่ลงใน โฟลเดอร์ /opt/lampp/htdocs/phpBB2/forum คลิก Finish Installation จะเตือนให้ลา โฟลเดอร์ install และ contrib ออก ทำการลบออกโดยใช้คำสั่งดังต่อไปนี้

```
rm -f -r /opt/lampp/htdocs/phpBB2/forum/install
```

```
rm -f -r /opt/lampp/htdocs/phpBB2/forum/contrib
```
8. ทำการเรียก <http://localhost/phpBB2/forum/> อีกครั้งจะเข้าสู่หน้าจอ log in ถือว่าติดตั้งโดยสมบูรณ์

นอกเหนือจากการติดตั้งแล้ว ยังต้องทำการกำหนดค่าต่างๆให้สามารถทำงานได้ตามที่ต้องการ การกำหนดค่าต่างๆ ทำตามขั้นตอน ดังนี้

1. เมื่อ log in แล้ว ที่แถบ links ด้านซ้ายมือในแถบ TorrentPier เลือก Forum Config
2. แก้ไขช่องข้อมูลต่างๆ ดังนี้
Announce url: ใส่ url หรือ ip ของเครื่อง Internal Tracker ตามด้วย path ที่อยู่หลัง htdocs ของเครื่อง ไปที่โฟลเดอร์ /phpBB2/bt/announce.php (เช่น <http://btcache.kmitl.ac.th/phpBB2/bt/announce.php>)
Passkey: ตั้งเป็น No ทั้งสามข้อ
3. กาช่อง confirm แล้วคลิก submit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ที่แถบ links ด้านซ้ายมือในแถบ TorrentPier เลือก Tracker Config
5. แก้ไขช่องข้อมูลต่างๆ ดังนี้
 Browser Redirect URL: กรณีที่มีผู้ต้องการเข้าไปที่ announce.php สามารถกำหนด URL ที่จะให้ redirect ไปได้
 Passkey: เลือก No
 Allow guess access to tracker: เลือก Yes
6. ภาช่อง confirm แล้วคลิก submit

เมื่อตั้งค่าเสร็จเรียบร้อยแล้วให้ทำการสร้างตาราง filetorrents เพิ่มเป็นฐานข้อมูลของ Internal Tracker ด้วย เพื่อเก็บข้อมูล info hash เอาไว้ตรวจสอบว่าเคยมีดาวน์โหลดไฟล์นี้มาแล้วหรือยัง และตาราง fileuploads เพื่อเก็บข้อมูลเอาไว้ตรวจสอบว่าเป็นการดาวน์โหลดข้อมูลจริงๆหรือเป็นการดาวน์โหลดเพื่ออัปโหลด โดยการสร้างตารางทำได้ตามขั้นตอนดังนี้

1. เข้า phpMyAdmin ในเครื่อง internal tracker กรอก user name และ password จากนั้นเลือกฐานข้อมูลที่ใช้งานแทรคเกอร์ (tracker)
2. ที่ Create new table on database แล้วตั้งชื่อตารางว่า filetorrents ใส่จำนวนฟิลด์เท่ากับ 3 ฟิลด์ แล้วกด GO
3. ในช่องฟิลด์ให้ใส่ torrent_id, info_hash และ file_path ตามลำดับ
4. torrent_id ให้ใส่ type เป็น mediumint(8), Null เป็น No
5. info_hash ให้ใส่ type เป็น binary(20), Null เป็น No
6. file_path ให้ใส่ type เป็น varchar(255), Null เป็น No
7. สร้างตารางชื่อ fileuploads ใส่จำนวนฟิลด์เท่ากับ 2 ฟิลด์แล้วกด GO
8. ในช่องฟิลด์ให้ใส่ ip, id ตามลำดับ
9. ip ให้ใส่ type เป็น varchar(255), Null เป็น No
10. id ให้ใส่ type เป็น mediumint(8), Null เป็น No
11. จากนั้นกด save

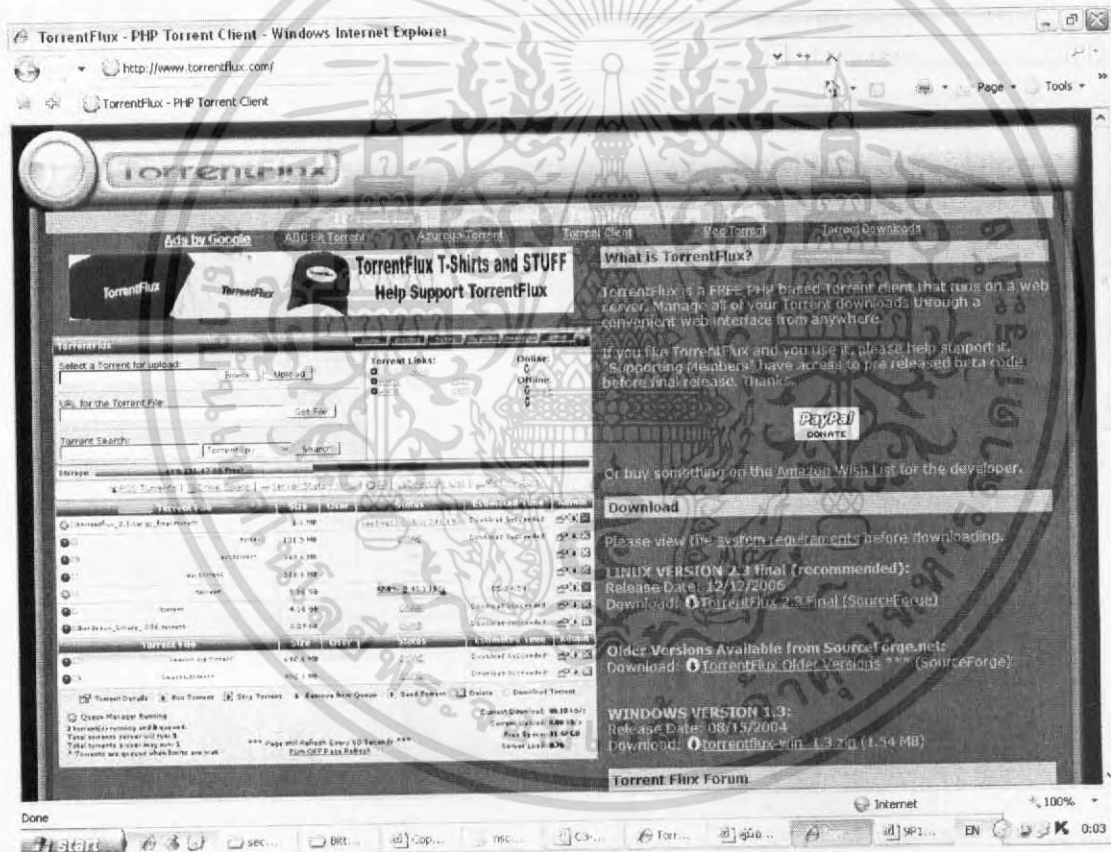
3. Internal Peer

ความต้องการทางด้านฮาร์ดแวร์

- CPU: ความเร็วอย่างต่ำ 1 GHz
- Hard disk: อย่างต่ำ 30 GB (เพราะเป็นเครื่องที่เก็บไฟล์ที่ดาวน์โหลดมา)
- RAM: อย่างต่ำ 256 MB
- Network Interface Card: รองรับความเร็วอย่างต่ำ 100 Mbps

ความต้องการทางด้านซอฟต์แวร์

- ระบบปฏิบัติการ: UNIX, Linux
- Bittorrent Client: TorrentFlux เวอร์ชัน 2.3



รูปที่ 5ก ภาพตัวอย่างหน้าเว็บไซต์ของ TorrentFlux

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งโปรแกรม Torrentflux

โปรแกรม Torrentflux เป็น Bittorrent Client ที่ทำงานด้วยภาษา php หรือการทำงานจะเป็นลักษณะของ Web-based คือควบคุมการทำงานผ่านทางหน้าเว็บไซต์ได้ทำให้ง่ายและสะดวกต่อการใช้งาน สามารถดาวน์โหลดได้จาก www.torrentflux.com โดยดาวน์โหลดมาเก็บไว้ที่ /opt/lampp/htdocs จากนั้นเข้าไปยังตำแหน่งที่ดาวน์โหลดมาเก็บไว้แล้วใช้คำสั่งดังนี้

- cd /opt/lampp/htdocs
- tar xvfz torrentflux_2.3.tar.gz
- เปิด Web browser ขึ้นมาแล้วพิมพ์ <http://localhost/phpmyadmin> กรอกข้อมูล user และ password ที่ตั้งไว้ถ้าล็อกอินสำเร็จจะเข้าสู่หน้า phpmyadmin ให้ทำการสร้างฐานข้อมูลชื่อ torrentflux
- import ไฟล์ /opt/lampp/htdocs/torrentflux_2.3/sql/mysql_torrentflux.sql เพื่อสร้างตารางต่างๆที่จำเป็นสำหรับโปรแกรม
- แก้ไขไฟล์ /opt/lampp/htdocs/torrentflux_2.3/html/config.php ดังนี้
db_type = "mysql" , db_host = "localhost" , db_name = "torrentflux" ,
db_usr = "username ที่ตั้งไว้" , และ db_pass = "password ที่ตั้งไว้"
- เปิด Web browser ขึ้นมาแล้วพิมพ์ http://localhost/torrentflux_2.3/html/index.php จะปรากฏหน้าต่างให้ทำการล็อกอินซึ่ง username และ password ที่ใส่ในครั้งแรกที่เข้าสู่เว็บไซต์นี้จะถูกตั้งให้เป็น Super Administrator โดยอัตโนมัติ
- เมื่อล็อกอินสำเร็จจะเข้าสู่หน้าต่างในการตั้งค่าต่างๆ โดยส่วนบนสุดจะเป็น Path ที่จะเก็บข้อมูลที่ได้จากการดาวน์โหลดให้ใส่เป็น
/opt/lampp/htdocs/torrentflux_2.3/html/downloads
- จากนั้นให้ทำการเปลี่ยน Permissions ดังนี้
chmod 777 /opt/lampp/htdocs/torrentflux_2.3/html/downloads เพื่อให้สามารถเขียนข้อมูลลงในโฟลเดอร์นี้ได้
- ถ้า Permissions ของ Path ถูกต้องแล้ว ปุ่มที่อยู่ด้านหลัง Path จะกลายเป็นสีเขียว แต่ถ้าเป็นสีแดงแสดงว่ายังไม่ถูกต้อง
- เปิด Browser ขึ้นมาแล้วพิมพ์ /opt/lampp/htdocs/torrentflux_2.3/html/index.php อีกครั้งถ้าเข้าสู่หน้าจอของ Torrentflux ได้แสดงว่าการติดตั้งเสร็จสมบูรณ์
- ทำการคัดลอกไฟล์ login.php และ index.php , filename.txt เข้าไปแทนที่ใน
/opt/lampp/htdocs/torrentflux_2.3/html
- คัดลอกไฟล์ start_torrent.php ไปไว้ที่ /opt/lampp/htdocs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คัดลอกไฟล์ `run_torrent.pl` , `filesize.pl` , `start_torrent.pl` ไปไว้ที่
`/opt/lampp/htdocs/torrent/return`



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Packet Filter

เป็นส่วนของการดักจับแพ็คเก็ตข้อมูลเพื่อดักจับค่าต่างๆ เพื่อนำไปใช้ในระบบ โดยสามารถแบ่งออกได้เป็น 2 ส่วนได้แก่การดักจับข้อมูลในการดาวน์โหลด และการดักจับข้อมูลในอีพีโพลด

ความต้องการทางด้านซอฟต์แวร์

- ระบบปฏิบัติการ: UNIX, Linux
- Base Program: Perl , Ruby , Libpcap
- TCP and HTTP Capturing: TCPdump

ในการดักจับแพ็คเก็ตนั้นจะใช้สคริปต์ภาษา Ruby ในการดักจับโดยจำเป็นต้องทำการติดตั้งไลบรารี libpcap สำหรับภาษา Ruby ก่อนตามขั้นตอนดังนี้

- ดาวน์โหลดไลบรารี libpcap-ruby_0.6.orig.tar.gz
- ทำการติดตั้งตามขั้นตอนดังนี้


```
# tar xvfz libpcap-ruby_0.6.orig.tar.gz
# cd pcap
# ruby extconf.rb
# make
# make install
```
- เมื่อติดตั้งไลบรารีเรียบร้อยแล้วให้ทำการสร้างโฟลเดอร์ดังนี้


```
# mkdir /opt/lampp/htdocs/httpdump
```

 เป็นโฟลเดอร์สำหรับเก็บโค้ดและข้อมูลที่ดักจับได้จากการดาวน์โหลด


```
# mkdir /opt/lampp/htdocs/httpdump_upload
```

 เป็นโฟลเดอร์สำหรับเก็บโค้ดและข้อมูลที่ดักจับได้จากการอัปโหลด
- สร้างไฟล์ httpdump.txt ไว้ที่ /opt/lampp/htdocs/httpdump เพื่อรองรับการทำงานของ httpdump.sh
- ทำการคัดลอกไฟล์ httpdump.sh ไปไว้ที่ /opt/lampp/htdocs/httpdump และไฟล์ httpdump_upload.sh ไปไว้ที่ /opt/lampp/htdocs/httpdump_upload
- สร้างไฟล์ httpdump.txt ไว้ที่ /opt/lampp/htdocs/httpdump_upload เพื่อรองรับการทำงานของ httpdump_upload.sh

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อติดตั้งในส่วนของการดักจับแพ็คเกจเรียบร้อยแล้ว ให้ทำการติดตั้งในส่วนของการจับข้อมูลที่ต้องการจากแพ็คเกจที่ดักจับได้ โดยในส่วนนี้จะใช้สคริปต์ภาษา Perl ในการดักจับตามขั้นตอนดังนี้

- ทำการคัดลอกไฟล์ filter1.pl และไฟล์ cookie_filter.pl ไปไว้ที่ /opt/lampp/htdocs/httpdump โดยไฟล์ filter1.pl จะทำหน้าที่ในการดักจับค่าไอพีแอดเดรส , ไอดีของไฟล์ .torrent , และชื่อไฟล์ .torrent นั้นๆ แล้วส่งต่อให้กับไฟล์ start_torrent.php เพื่อสั่งให้ไฟล์ .torrent เริ่มทำงาน ส่วนไฟล์ cookie_filter.pl ทำหน้าที่ในการดักจับ cookie ของผู้ใช้เพื่อใช้ในการดาวน์โหลดไฟล์ .torrent
- สร้างไฟล์ cookie.txt , temp.txt ไว้ที่ /opt/lampp/htdocs/httpdump เพื่อรองรับการทำงานของ filter1.pl และ cookie_filter.pl
- เปลี่ยนโหมดให้สามารถเขียน Text File ที่สร้างขึ้นด้วยคำสั่ง
chmod -R 750 /opt/lampp/htdocs/httpdump
- ทำการคัดลอกไฟล์ filter2.pl ไปไว้ที่ /opt/lampp/htdocs/httpdump_upload โดยไฟล์ filter2.pl ทำหน้าที่ในการดักจับไอพีแอดเดรส , ไอดีของไฟล์ .torrent , และชื่อไฟล์เมื่อมีการอัปโหลด แล้วส่งต่อให้ไฟล์ add_ip.php เพื่อทำการเพิ่มข้อมูลเหล่านี้ลงในฐานข้อมูลว่ามีการอัปโหลดข้อมูลเกิดขึ้น

5. Web Service

ความต้องการทางด้านฮาร์ดแวร์

- CPU: ความเร็วอย่างต่ำ 1 GHz
- Hard disk: อย่างต่ำ 20 GB
- RAM: อย่างต่ำ 256 MB
- Network Interface Card: รองรับความเร็วอย่างต่ำ 100 Mbps

ความต้องการทางด้านซอฟต์แวร์

- ระบบปฏิบัติการ: UNIX, Linux
- Web Server: XAMPP (apache, MySQL, PHP)

เมื่อติดตั้งซอฟต์แวร์ต่างๆเรียบร้อยแล้ว ให้ทำการคัดลอกไฟล์ download.php, func_download.php, add_ip.php, start_torrent.php และ bencode.inc.php ไปไว้ยังโฟลเดอร์ /opt/lampp/htdocs/ จากนั้นให้ทำการเปลี่ยน permission ให้สามารถ execute ได้ โดยใช้คำสั่ง chmod 755 ตามด้วยชื่อไฟล์ทั้ง 3 ดังที่ได้กล่าวไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

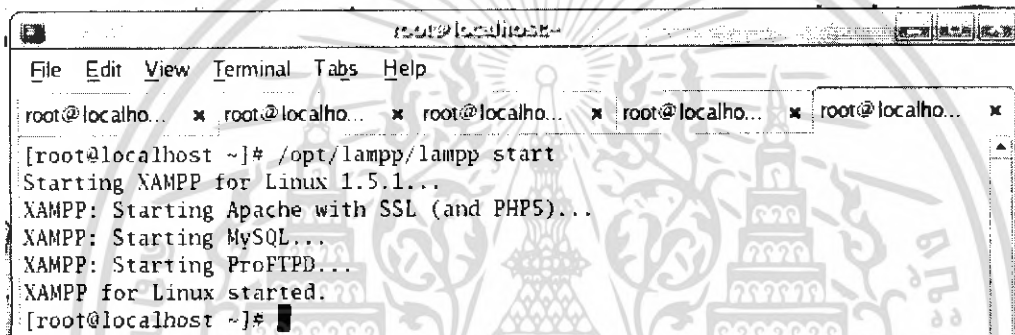
ภาคผนวก ข.

ขั้นตอนการใช้งานระบบแคชสำหรับโปรแกรมบิตทอเรนท์

หลังจากที่ได้ทำการติดตั้งองค์ประกอบต่างๆของระบบแคชสำหรับโปรแกรมบิตทอเรนท์เป็นที่เรียบร้อยแล้ว ก็มาถึงการสั่งให้ระบบทำงาน ซึ่งมีขั้นตอนดังนี้

1. เปิดเครื่องที่ทำหน้าที่เป็น Proxy Server (โดยปกติแล้ว Squid จะทำงานทันทีที่เปิดเครื่อง)
2. เปิดเครื่องที่ทำหน้าที่เป็น Web Service, Internal Tracker , Packet Filter และ Internal Peer จากนั้นทำการสั่งให้ระบบทำงาน โดยรันคำสั่งดังต่อไปนี้

/opt/lampp/lampp start เป็นการสั่งให้ Web Server เริ่มทำงาน

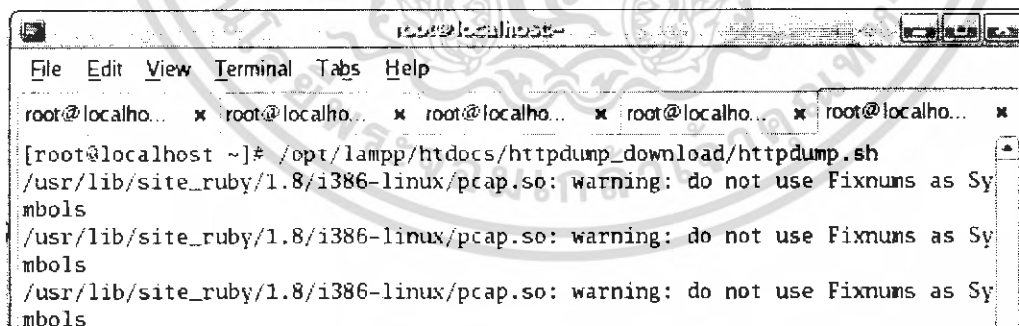


```

root@localhost~# /opt/lampp/lampp start
Starting XAMPP for Linux 1.5.1...
XAMPP: Starting Apache with SSL (and PHP5)...
XAMPP: Starting MySQL...
XAMPP: Starting ProFTPD...
XAMPP for Linux started.
root@localhost~#
  
```

รูปที่ 1x ภาพตัวอย่างการใช้งานสั่งให้ Web Server เริ่มทำงาน

/opt/lampp/htdocs/httpdump_download/httpdump.sh เป็นการเริ่มดักจับแพ็คเก็ตเนื่องจากการดาวน์โหลด



```

root@localhost~# /opt/lampp/htdocs/httpdump_download/httpdump.sh
/usr/lib/site_ruby/1.8/i386-linux/pcap.so: warning: do not use Fixnums as Symbols
/usr/lib/site_ruby/1.8/i386-linux/pcap.so: warning: do not use Fixnums as Symbols
/usr/lib/site_ruby/1.8/i386-linux/pcap.so: warning: do not use Fixnums as Symbols
  
```

รูปที่ 2x ภาพตัวอย่างการเริ่มดักจับแพ็คเก็ตเนื่องจากการดาวน์โหลด

/opt/lampp/htdocs/httpdump_upload/httpdump.sh เป็นการเริ่มดักจับแพ็คเก็ตเนื่องจากการอัปโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

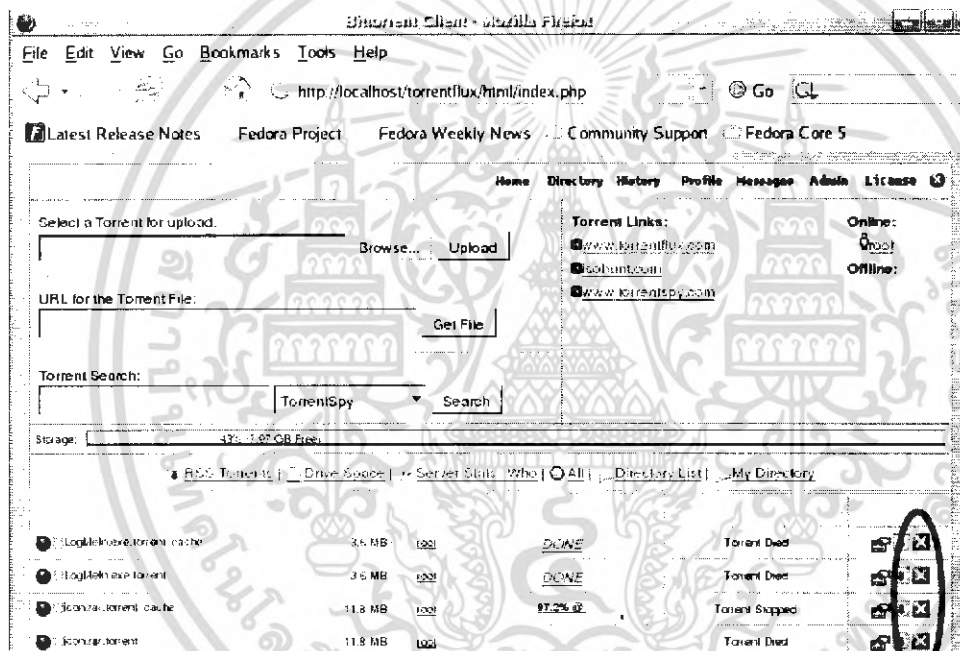
```

root@localho... x root@localho... x root@localho... x root@localho... x root@localho... x
[root@localhost ~]# /opt/lampp/htdocs/httpdump_upload/httpdump.sh
/usr/lib/site_ruby/1.8/i386-linux/pcap.so: warning: do not use Fixnums as Symbols
/usr/lib/site_ruby/1.8/i386-linux/pcap.so: warning: do not use Fixnums as Symbols
/usr/lib/site_ruby/1.8/i386-linux/pcap.so: warning: do not use Fixnums as Symbols

```

รูปที่ 3 ภาพตัวอย่างการเริ่มดักจับแพ็คเก็ตจากการอัปเดต

3. หากต้องการเริ่มต้นระบบทั้งหมดใหม่ต้องทำการเคลียร์ข้อมูลต่างๆดังต่อไปนี้
 - a. เปิด Web Browser ขึ้นมาแล้วพิมพ์ localhost/torrentflux/html/index.php
 - b. ทำการลบข้อมูลทั้งหมดที่ทำงานอยู่



รูปที่ 4 ภาพตัวอย่างการทำงานการเคลียร์ข้อมูลต่างๆ

- c. เปิด Web Browser ขึ้นมาแล้วพิมพ์ localhost/PhpMyAdmin
- d. ล็อกอินเข้าสู่ระบบ แล้วเลือก Database tracker ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The screenshot shows the phpMyAdmin interface for a database named 'tracker'. The left sidebar shows a tree view of the database structure, with 'tracker (54)' selected. The main area displays the table structure for 'tracker', listing tables and their record counts.

Table	Action	Records
filetorrents	[Icons]	3
fileuploads	[Icons]	1
phpbb_attachments	[Icons]	1
phpbb_attachments_config	[Icons]	34
phpbb_attachments_desc	[Icons]	1
phpbb_attach_quota	[Icons]	0
phobb_auth_access	[Icons]	0

รูปที่ 3 ภาพตัวอย่างการทำงานการเคลียร์ข้อมูลต่างๆ ขั้นตอนที่สอง

- e. ทำการ Empty ตารางต่างๆ ดังนี้
 - filetorrents
 - fileuploads
 - php_bt_torrent
 - php_bt_tracker
 - f. ลบข้อมูลที่ได้ทำการดาวน์โหลดมาแล้วทั้งหมดในโฟลเดอร์
/opt/lampp/htdocs/torrentflux/html/downloads/root
4. เชื่อมพร็อกซีที่เครื่องไคลเอนต์ให้เป็นไอพีแอดเดรสของ Proxy Server (ในกรณีที่ไม่ได้ใช้ transparent proxy)
จากนั้นเมื่อเครื่อง client มีการร้องขอไฟล์ .torrent ก็จะเข้าสู่การทำงานของระบบแคช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้