

รูปแบบทางคณิตศาสตร์แบบง่ายสำหรับการแชร์เพิ่มข้อมูลในระบบ Bittorrent

A Simple Mathematical Model for a Bittorrent File Sharing System

คำนำ

เครือข่ายแบบเพียร์ทูเพียร์เป็นรูปแบบเครือข่ายแบบใหม่ที่เกิดขึ้นมาภายใน 2-3 ปีที่ผ่านมา รูปแบบสถาปัตยกรรมของเครือข่ายแบบเพียร์ทูเพียร์จะต่างจากรูปแบบสถาปัตยกรรมแบบ Client/Server ในหลายด้าน ซึ่งหนึ่งในนั้นที่เป็นจุดเด่นคือเพียร์ (Peer) เป็นทั้งผู้ให้บริการและผู้ใช้บริการ ตัวอย่างที่เห็นได้ชัดคือในระบบแจกจ่ายเพิ่มข้อมูลแบบเพียร์ทูเพียร์ ในจำนวนเพียร์หลายตัวที่เชื่อมโยงซึ่งกันและกันอยู่ในเครือข่าย แต่ละเพียร์จะทำการร้องขอเพิ่มข้อมูลจากเพียร์อื่นๆ ในขณะที่เดียวกันก็ทำการเก็บเพิ่มข้อมูลและทำการบริการให้กับเพียร์ตัวอื่นๆ ไปด้วยในเวลาเดียวกัน จะเห็นได้ว่าในขณะที่เพียร์ทำการสร้างงานให้ระบบของเพียร์นั้นๆ เพียร์ยังเป็นทำหน้าที่บริหารงานแก่ระบบไปด้วยกัน

จากข้างต้นจะเห็นได้ว่าการเพิ่มเพียร์ขึ้นมาในระบบไม่เพียงแต่จะเพิ่มงานให้กับระบบแต่ก็เพิ่มการบริหารงานให้กับระบบด้วย เมื่อเทียบกับระบบเครือข่ายแบบ Client/Server งานจะถูกสร้างโดย Client และจะถูกบริหารโดย Server การเพิ่มขึ้นของ Client จะทำให้งานของระบบเพิ่มมากขึ้นแต่การบริหารงานก็ยังจำกัดอยู่กับจำนวนของ server เพียงอย่างเดียว

อย่างไรก็ตามเครือข่ายเพียร์ทูเพียร์จะเกิดขึ้นได้นั้นก็ต้องได้รับการยอมรับจากผู้ใช้งานเป็นจำนวนหนึ่งเพื่อให้เครือข่ายดังกล่าวสามารถดำรงตัวเองให้คงสภาพเครือข่ายไว้ได้ ซึ่งปัจจัยสำคัญที่ทำให้เครือข่ายใดๆ เป็นที่นิยมในการใช้งานนั้นขึ้นอยู่กับว่าเครือข่ายนั้นๆ สามารถรองรับผู้ใช้งานทุกๆ คนได้อย่างเป็นที่น่าสนใจหรือไม่

วัตถุประสงค์

ในงานวิจัยนี้ได้เลือกสังเกตการณ์เครือข่ายพีทูพีที่ชื่อว่าBittorrent (Cohen B., 2003) ซึ่งพัฒนาโดย Bram Cohenเพื่อใช้งานด้านการแจกจ่ายข้อมูล (File distribution system) เหตุผลที่เลือกศึกษาเครือข่ายดังกล่าวเป็นเพราะในอินเทอร์เน็ตในการส่งข้อมูลที่มีขนาดใหญ่ได้มีการใช้เครือข่ายแบบพีทูพีพีทูพีมาใช้งานแทนเครือข่ายแบบClient/Serverมากขึ้นเนื่องจากค่าใช้จ่ายที่ต่ำกว่ามากโดยเครือข่ายBittorrentเป็นที่นิยมมากขึ้นเรื่อยๆในช่วง1-2ปีที่ผ่านมา ทำให้มีตัวอย่างให้สังเกตการณ์ได้หลากหลายรูปแบบ

งานวิจัยนี้ได้จัดทำขึ้นเพื่อศึกษาพฤติกรรมการใช้งานเครือข่ายพีทูพีพีทูพีแบบBittorrent โดยทำการจัดเก็บข้อมูลการใช้งานของเครือข่าย เพื่อค้นหารูปแบบทางคณิตศาสตร์จำลองการแจกจ่ายข้อมูลของเครือข่ายแบบพีทูพีพีทูพี และนำรูปแบบดังกล่าวมาค้นหาความสัมพันธ์ของตัวแปรต่างๆของระบบ ไม่ว่าจะ ขนาดข้อมูล หรือ อัตราการส่งข้อมูลของพีทูพีในระบบ ว่ามีผลต่อระบบหรือไม่อย่างไร

การตรวจเอกสาร

โครงสร้างสถาปัตยกรรมของเครือข่ายเพียร์ทูเพียร์

Zihui et al. (2003) ทำการศึกษาโครงสร้างของการส่งคำสั่งค้นหาและโอกาสในการตอบรับของระบบแชร์ข้อมูลแบบเพียร์ทูเพียร์ โดยอาศัยทฤษฎี Queuing model ซึ่งแบ่งได้เป็น 3 แบบ นั่นคือ

1. กระบวนการสร้างเครือข่าย เป็นกระบวนการที่เพียร์ต่างๆ ทำการสื่อสารกันและกันเพื่อทำการสร้างเครือข่ายเสมือนขนาดใหญ่ขึ้น
2. กระบวนการการสืบค้น เป็นกระบวนการที่เพียร์ใดๆ ใช้ในการสืบค้นเพิ่มข้อมูลต่างๆ ที่กระจายตัวกันในเครือข่าย
3. กระบวนการการดึงข้อมูล เป็นกระบวนการที่เพียร์ทำการดึงเพิ่มข้อมูลที่ต้องการจากเพียร์ใดๆ ที่อยู่ในเครือข่าย

เครือข่ายเพียร์ทูเพียร์ ส่วนใหญ่ในกระบวนการที่ 1 และ 2 จะมีการทำงานร่วมกัน นั่นคือรูปแบบ (Protocol) ที่ใช้ในการสืบค้นข้อมูลจะเป็นส่วนหนึ่งของ Protocol ที่ใช้ในการสร้างเครือข่ายซึ่งมองโดยรวมแล้วเราสามารถแบ่งเครือข่ายเพียร์ทูเพียร์ตามกระบวนการที่กล่าวมาข้างต้นออกได้ 3 ลักษณะใหญ่คือ

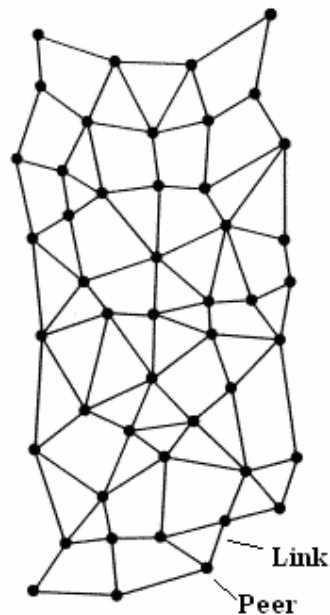
1. แบ่งตามรูปแบบหน้าที่ความสำคัญของเพียร์ที่เชื่อมต่อกัน (Degree of centralization)
2. แบ่งตามรูปแบบการวางตัวของเพียร์ในเครือข่าย (Network structure)
3. แบ่งตามรูปแบบวิธีการดึงข้อมูลของเพียร์ (Download Method)

การแบ่งตามรูปแบบหน้าที่ความสำคัญของเพียร์ที่เชื่อมต่อ

เครือข่ายเพียร์ทูเพียร์สามารถแบ่งตามรูปแบบหน้าที่ความสำคัญของ Peer ที่เชื่อมต่อกัน ซึ่งแบ่งได้เป็นรูปแบบย่อย 3 รูปแบบ นั่นคือ

Purely Decentralized

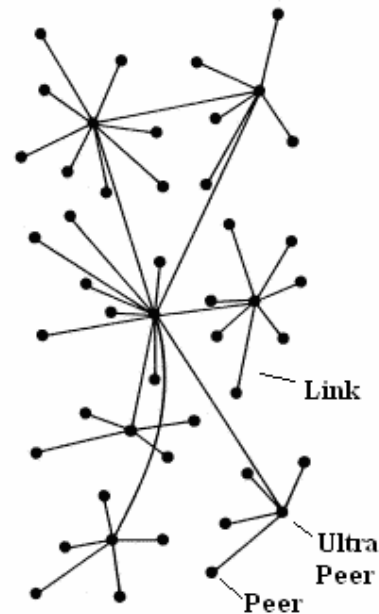
เพียร์ทุกตัวที่ต่อกันเป็นเครือข่ายทำหน้าที่เหมือนกัน นั่นคือทำหน้าที่เป็นทั้ง Server และ Client ในเวลาเดียวกันดังภาพที่1 และไม่มีเพียร์ใดที่ทำหน้าที่เป็นศูนย์กลางในการติดต่อระหว่างเพียร์ต่างๆตัวอย่างของระบบนี้คือ Chord, CAN, Pastry และ Tapestry



ภาพที่ 1 เครือข่ายแบบ Purely Decentralized

Partially Centralized

หลักการงานทั่วไปเหมือนแบบPurely decentralized เพียงแต่เพียร์บางเพียร์ที่มีทรัพยากรของระบบที่ดีกว่า(Network connection)จะหน้าที่เป็นUltraPeer (ภาพที่2) ซึ่งทำหน้าที่เป็นตัวเก็บข้อมูลดัชนีของเพียร์ใกล้เคียง หรือ ทำหน้าที่เป็นเพียร์ที่ทำตัวเป็นศูนย์กลางให้เพียร์ปกติทำการเชื่อมต่อเพื่อลดภาระข้อมูลที่วิ่งภายในเครือข่าย แต่จะไม่ทำให้เครือข่ายล่มในกรณีที่UltraPeerเกิดไม่ทำงาน เนื่องจาก เพียร์ที่เหลือจะทำงานเชื่อมต่อกับUltraPeerอื่นๆทันทีการส่งคำสั่งของระบบนี้สามารถพิจารณาได้ว่าจะทำให้การจราจรในเครือข่ายเพิ่มขึ้นในระดับหนึ่ง (Schollmeier and Schollmeier, 2002) ตัวอย่างของระบบนี้คือ Gnutella



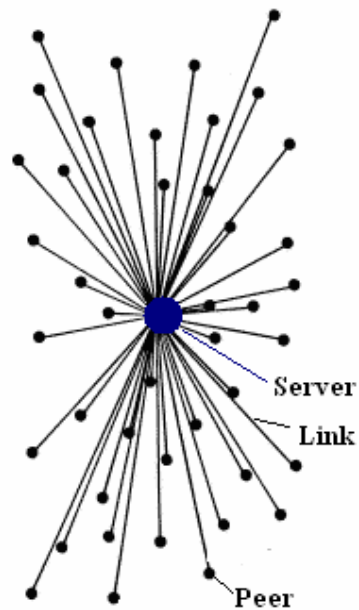
ภาพที่ 2 เครือข่ายแบบ Partially Centralized

Hybrid Decentralized

หลักการของการทำงานของเครือข่ายแบบนี้จะคล้ายกับระบบ Client/Server นั่นคือจะมี Server ทำหน้าที่เป็นตัวเป็นข้อมูลดัชนีของเพียร์ต่างๆที่เชื่อมต่อกันในเครือข่ายซึ่งเพียร์ใดที่ต้องการจะสืบค้นข้อมูลก็จะมาทำการติดต่อกับ Server เพื่อค้นหาว่ามีเพียร์ใดที่มีข้อมูลที่ต้องการก็จะทำการดึงข้อมูลที่เพียร์นั้นต่อไปตามภาพที่ 3 ตัวอย่างของระบบนี้คือ Napster

การแบ่งตามรูปแบบการวางตัวของเพียร์ในเครือข่าย

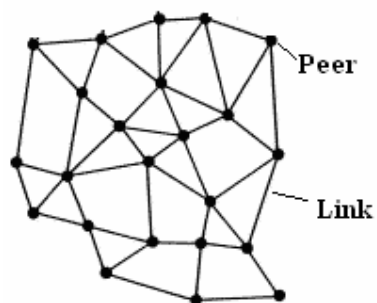
เครือข่ายเพียร์ทูเพียร์ประกอบไปด้วยเพียร์หลายๆเพียร์มารวมกันเป็นเครือข่ายขนาดใหญ่ซึ่งมีรูปแบบการเชื่อมต่อที่ซับซ้อนซึ่งรูปแบบดังกล่าวไม่มีความเกี่ยวข้องกันกับที่เพียร์แต่ละเพียร์ทำการเชื่อมต่อกันในทางกายภาพ เครือข่ายเพียร์ทูเพียร์สามารถแบ่งได้ตามรูปแบบของเครือข่ายว่ามีรูปแบบการเชื่อมต่อของเพียร์ที่แน่นอนหรือว่าเป็นการเชื่อมต่อแบบชั่วคราว ซึ่งแบ่งได้ 2 แบบคือ



ภาพที่ 3 เครือข่ายแบบ Hybrid Decentralized

Unstructured Network

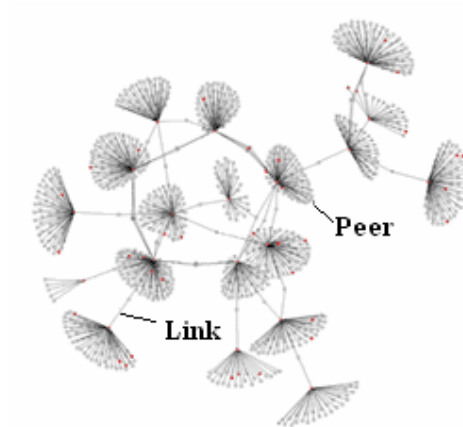
ตามภาพที่ 4 ในเครือข่ายลักษณะนี้ข้อมูลภายในเพียร์ ไม่มีผลต่อตำแหน่งที่เพียร์นั้นๆจะคงอยู่ภายในเครือข่าย การทำการสืบค้นภายในเครือข่ายกระทำโดยการค้นหาแบบสุ่ม ซึ่งจะกระทำโดยการส่งคำสั่งสืบค้นไปยังเพียร์ต่อเพียร์ จุดดีของเครือข่ายประเภทนี้คือสามารถรองรับจำนวนของเพียร์ได้เป็นจำนวนมาก แต่จุดเสียคือในการสืบค้นต้องมีการส่งคำสั่งในการสืบค้นให้มากพอเพื่อที่จะมีโอกาสที่จะเจอข้อมูลที่ต้องการ



ภาพที่ 4 Unstructured Network

Structured Network

ตามภาพที่ 5 เป็นเครือข่ายที่ถูกออกแบบขึ้นเพื่อแก้ไขปัญหาที่เกิดขึ้นกับ Unstructured network ในเรื่องการสืบค้นข้อมูล โดยที่โครงสร้างของเครือข่ายนี้มีการออกแบบเป็นพิเศษข้อมูลที่ถูกรวบรวมไว้ในพีเออร์จะเป็นตัวแปรที่กำหนดว่าพีเออร์นั้นจะอยู่ในตำแหน่งใดในเครือข่าย เมื่อมีการใช้คำสั่งสืบค้นมาก็จะสามารถระบุเจาะจงตำแหน่งของเครือข่ายที่จะส่งคำสั่งสืบค้นไปค้นหาข้อมูลได้ ทำให้มีโอกาสเจอข้อมูลที่ต้องการมากขึ้น ข้อเสียของระบบนี้คือยากที่จะรักษาโครงสร้างของเครือข่ายให้มีความสอดคล้องกับข้อมูลของพีเออร์ทั้งหมดได้ เนื่องจากพีเออร์มีอัตราการเข้าออกเครือข่ายอยู่ในระดับสูง (Zhao et al., 2001)



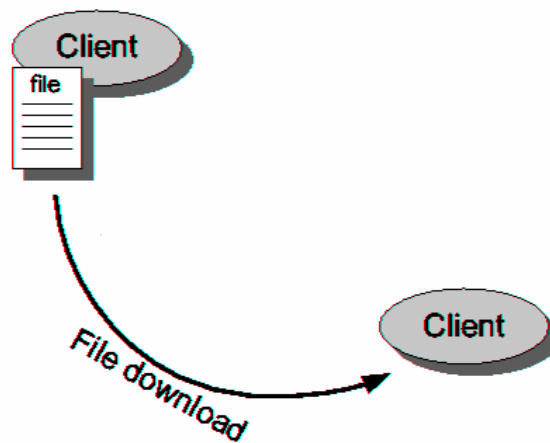
ภาพที่ 5 Structured Network

การแบ่งตามรูปแบบวิธีการดึงข้อมูลของพีเออร์

หลังจากเสร็จสิ้นการสืบค้นข้อมูลจากเครือข่ายพีเออร์พีเออร์แล้ว ก็จะเข้าสู่ขั้นตอนการดึงข้อมูล พีเออร์ที่ต้องการจะดึงข้อมูลก็จะใช้ข้อมูลพีเออร์ที่ได้จากการสืบค้นทำการติดต่อและทำการดึงข้อมูลจากพีเออร์ที่มีข้อมูลนั้นๆอยู่ซึ่งรูปแบบการดึงข้อมูลสามารถแบ่งได้ดังนี้

Direct Download

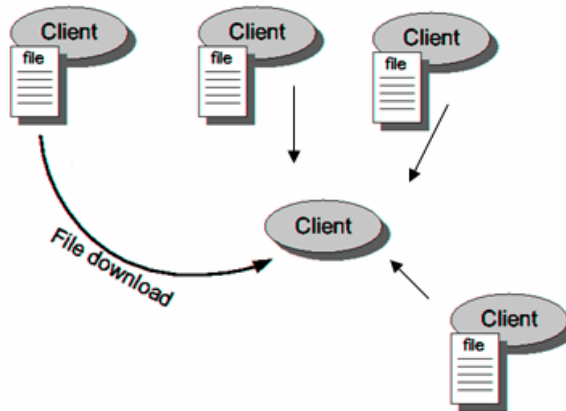
เป็นรูปแบบการดึงข้อมูลที่มีลักษณะเป็นแบบ Client/Server นั่นคือ เพียร์ที่ทำการดึงข้อมูลทำหน้าที่เป็น Clientทำการขอข้อมูลจากServerซึ่งก็คือเพียร์ที่มีข้อมูลอยู่ ซึ่งเป็นการดึงข้อมูลแบบ One-to-One เพียร์ที่ทำการดึงข้อมูลจะต้องทำการดึงข้อมูลให้ได้ครบก่อนจึงจะสามารถให้บริการข้อมูลที่ดึงมาดังกล่าวแก่เพียร์อื่นๆต่อไปได้ ตัวอย่างเช่น FTPและGnutella (1.0)



ภาพที่ 6 Direct Download

Multiple Source Download

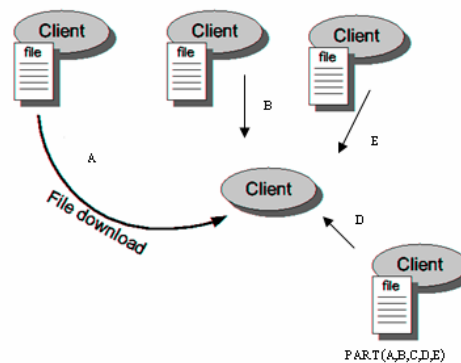
เป็นรูปแบบการที่พัฒนามาจากรูปแบบแรก นั่นคือ เพียร์ที่ทำการดึงข้อมูลสามารถทำการดึงข้อมูลจากเพียร์ต้นฉบับได้หลายที่ ซึ่งเป็นการดึงข้อมูลแบบ Many-to-one ข้อดีคือสามารถทำการดึงข้อมูลได้เร็วมากขึ้น อย่างไรก็ตาม เพียร์ที่ทำการดึงข้อมูลจะต้องทำการดึงข้อมูลให้ได้ครบก่อนจึงจะสามารถให้บริการข้อมูลที่ดึงมาดังกล่าวแก่เพียร์อื่นๆต่อไปได้อีกเหมือนรูปแบบการดึงข้อมูลแบบ Direct downloader ตัวอย่างของระบบนี้คือ eDonkey



ภาพที่ 7 Multiple Source Download

Multiple Source Multiple File Chunks Download

เป็นรูปแบบการที่พัฒนามาจาก Multiple Source Download ข้อมูลที่ใช้ในการดึงข้อมูลจะถูกแบ่งออกเป็นส่วนๆ เพียร์ที่ทำการดึงข้อมูลสามารถตั้งตัวเองเป็นเพียร์ที่สามารถให้ดึงข้อมูลส่วนที่ทำการดึงข้อมูลเสร็จแล้วได้ทันทีโดยไม่ต้องทำการดึงข้อมูลให้เสร็จครบทุกส่วน ข้อดีคือสามารถแก้ปัญหา Free riding ได้ดี (Kant, 2003) เพราะเพียร์ที่ทำการดึงข้อมูลก็สามารถมีส่วนร่วมในการให้บริการข้อมูลได้น้อยจนกว่าเพียร์นั้นจะทำการดึงข้อมูลเสร็จทุกส่วน ตัวอย่างของระบบนี้คือ Bittorrent



ภาพที่ 8 Multiple Source Multiple File Chunks Download

Bittorrent

Bittorrent เป็นเครือข่ายเพียร์ทูเพียร์ที่ใช้ในการกระจายข้อมูลจากเพียร์ต้นฉบับไปยังอีกเพียร์อื่นๆที่อยู่ในเครือข่ายนั้นๆ เครือข่ายBittorrentหนึ่งเครือข่ายประกอบด้วยเพียร์หลายตัวที่ทำหน้าที่ในการให้บริการส่งและเก็บแฟ้มข้อมูลที่ต้องการกระจายเพียง1แฟ้มข้อมูลเท่านั้นเท่านั้น

ชื่อและหน้าที่ของหน่วยต่างๆภายในเครือข่ายBittorrent

Seeder

ใช้เรียกเพียร์ที่มีแฟ้มข้อมูลที่ใช้ในการกระจายภายในเครือข่ายนั้นๆในสภาพครบถ้วน Seederเป็นหัวใจสำคัญของเครือข่ายBittorrent เพราะการที่จะทำให้เครือข่ายสามารถดำเนินการได้นั้น อย่างน้อยที่สุดต้องมีSeederภายในเครือข่ายนั้นๆอย่างน้อยที่สุด1เพียร์ เพื่อที่จะสามารถกระจายแฟ้มข้อมูลที่สมบูรณ์ไปยังDownloaderได้

Downloader

ใช้เรียกเพียร์ที่ยังไม่มีแฟ้มข้อมูลหรือมีแต่ไม่ครบแฟ้มที่กำลังทำการกระจายภายในเครือข่ายนั้นๆ Downloader จะทำการดึงข้อมูลจากSeederหรือDownloaderอื่นๆภายในระบบ เมื่อDownloaderใดๆได้รับข้อมูลครบถ้วนแล้วก็จะทำการยกระดับตัวเองกลายเป็นSeederซึ่งทำหน้าที่กระจายแฟ้มข้อมูลนั้นๆไปยังDownloaderอื่นๆต่อไป

Torrent

เป็นแฟ้มข้อมูลอ้างอิงที่Seederตัวแรกสุดทำการแจกจ่ายไปยังWebsiteต่างๆเพื่อเชิญชวนให้ผู้ที่มีความสนใจในข้อมูลที่Seederตัวแรกเสนอ ใช้ข้อมูลที่อยู่ที่ภายในแฟ้มTorrentนั้นๆเป็นข้อมูลนำทางในการติดต่อกับSeederตัวแรกสุด ผ่านTracker ที่อ้างอิงภายในTorrentเพื่อเข้าถึงเครือข่ายBittorrentที่Seederตัวแรกสร้างขึ้น ข้อมูลภายในTorrentจะประกอบไปด้วย ข้อมูลที่อยู่ (IP address) ของTracker เลขที่อ้างอิงของเครือข่ายBittorrentที่ทำการสมัครไว้ในTracker ชุดข้อมูลตรวจสอบความถูกต้องของแฟ้มข้อมูลที่ทำการดึงมาในแต่ละส่วน (Data Checksums)

Tracker

เป็นServerที่ทำหน้าที่เก็บดัชนีที่อยู่ไอพีของพีเออร์ที่อยู่ภายในเครือข่ายBittorrent เมื่อพีเออร์ใดๆเข้ามาเพื่อขอข้อมูลของเครือข่ายBittorrentที่สมัครไว้ที่Trackerนั้นๆ Trackerก็จะทำการส่งชุดข้อมูลIP address ของพีเออร์ที่อยู่ภายในเครือข่ายนั้นๆกลับไปให้กับพีเออร์ที่ร้องขอ และทำการบันทึกIP addressของพีเออร์ที่ทำการร้องขอไว้ในชุดIP address ของเครือข่ายนั้นๆเพื่อเป็นข้อมูลให้กับการร้องขอถัดไป

Bittorrent client

เป็นโปรแกรมที่พีเออร์ใช้ในการเชื่อมต่อเครือข่ายBittorrent ตัวClientนี้ Bram Cohen ได้กำหนดให้เป็น Open Sourceให้นักพัฒนาสามารถออกแบบและสร้างClientของตนเองขึ้นมาได้ Clientที่เป็นที่นิยมใช้งานในปัจจุบันได้แก่ ABC, Azureus และBittornado เป็นต้น

หลักการทำงานของBittorrent

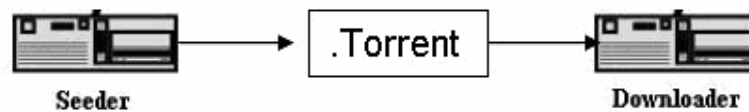
มีขั้นตอนการทำงานดังนี้

1. เมื่อผู้ใช้งานมีความต้องการที่จะสร้างเครือข่ายBittorrentของข้อมูลใดๆจะเริ่มต้นโดยการสมัครตัวเองเข้าไปยังTracker ตามภาพที่ 9



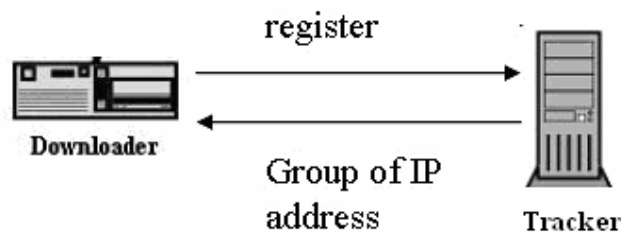
ภาพที่ 9 Seederตัวแรกทำการสมัครเครือข่ายไว้ที่Tracker

2. จากนั้นผู้ใช้งานคนแรกทำหน้าที่เป็นผู้แจกจ่ายข้อมูลหรือSeederจำทำการสร้างTorrent และทำการแจกจ่ายTorrentนี้ไปทางอินเทอร์เน็ต เพื่อให้ผู้ที่มีความสนใจในข้อมูลเดียวกันซึ่งเรียกว่า Downloader ทำการรับTorrent นั้นๆ ตามภาพที่10



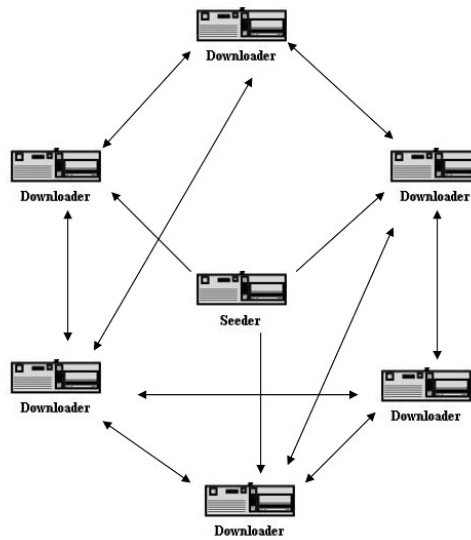
ภาพที่ 10 Seederทำการแจกจ่ายไฟล์Torrent ให้แก่downloader

3. Downloader ก็จะใช้Torrentนั้นๆทำการติดต่อไปยังTracker เพื่อสมัครตัวเองเข้าไปยังเครือข่ายที่ระบุอยู่ในTorrentและทำการรับชุดIP addressของกลุ่มSeederและDownloaderที่ทำการรวมตัวกันสร้างเป็นเครือข่ายBittorrentของข้อมูลนั้นๆ ดังแสดงไว้ในภาพที่11



ภาพที่ 11 รูปแบบการสร้างเครือข่ายของ Bittorrent

4. Downloaderก็จะใช้ข้อมูลIP address นั้นๆทำการติดต่อไปยังเครือข่ายBittorrent ซึ่งจะเชื่อมต่อแบบ Purely Decentralized (Zihui Ge และคณะ, 2003) ดังแสดงไว้ในภาพที่12



ภาพที่ 12 ลักษณะเครือข่าย Bittorrent

ในการกระจายข้อมูลจากSeederไปยังDownloaderนั้น เครือข่ายBittorrentมีวิธีการเฉพาะตัว โดยทำการแตกข้อมูลออกเป็นส่วนๆมีขนาดตั้งแต่256กิโลไบต์ไปจนถึง2เมกะไบต์ตามแต่ขนาดของข้อมูลที่ทำกรแบ่ง โดยข้อมูลแต่ละส่วนจะถูกส่งไปยังDownloaderใดๆที่อยู่ในเครือข่าย เมื่อDownloaderได้รับข้อมูลส่วนที่ต้องการเสร็จสิ้นก็สามารถเป็นตัวแทนในการกระจายข้อมูลส่วนนั้นไปยังDownloaderอื่นๆที่ยังไม่มีข้อมูลส่วนนั้นได้ เมื่อDownloaderใดๆได้รับข้อมูลครบทุกส่วนก็จะเปลี่ยนระดับตัวเองเป็นSeederแล้วทำหน้าที่เป็นตัวแจกจ่ายข้อมูลอย่างเดียว

งานที่เกี่ยวข้อง

Zeinalipour and Folios (2002) ได้การสำรวจพฤติกรรมของผู้ใช้งานGnutellaในปี2002เป็นเวลาชั่วโมงและพบว่าส่วนใหญ่จะเป็นการค้นหาเพิ่มข้อมูลที่เป็นแบบภาพเคลื่อนไหวโดยวัดได้ถึง18%

Kant (2003) ได้วิเคราะห์ถึงนิสัยของผู้ใช้งานระบบแชร์ข้อมูลแบบจุดต่อจุดไว้3แบบ แบบแรกเป็นกลุ่มที่อยู่ในระบบนานมากมีความเร็วในการส่งข้อมูลสูงและทำการแชร์ข้อมูลไว้เป็นจำนวนมาก แบบที่2เป็นกลุ่มที่อยู่ในระบบนานถึงระดับหนึ่งและแชร์ข้อมูลไว้เป็นจำนวนหนึ่ง แบบที่3เป็นกลุ่มที่อยู่ในระบบเป็นช่วงเวลาสั้นๆไม่มีการแชร์ข้อมูลไว้และเมื่อทำการดึงข้อมูล

สำเร็จก็จะออกจากระบบซึ่ง Adar and Huberman (2000) รายงานไว้ว่าในระบบ Gnutella มีแบบที่ 3 เป็นจำนวนถึง 70% ของ Peer ทั้งหมด

Tutschku (2004) ศึกษาระบบ eDonkey ซึ่งเป็นระบบการกระจายข้อมูลแบบเดียวกับ Napster ที่ Peer ที่ทำการดึงข้อมูลสามารถดึงข้อมูลจากกลุ่มของ Peer ที่มีข้อมูลที่ต้องการได้ภายในเวลาเดียวกันและสามารถเป็น Peer ที่ให้บริการดึงข้อมูลของส่วนของผู้เพิ่มที่ทำการดึงข้อมูลเสร็จแล้วได้พร้อมกัน แต่เพราะรูปแบบของการกระจายข้อมูลเป็นแบบคิวทำให้อัตราการกระจายตัวของข้อมูลมีค่าต่ำ

Yang and Veciana (2004) ได้ทำการศึกษาเครือข่าย Bittorrent และได้ใช้ Markov chain อธิบาย Steady State ของระบบ โดยที่ได้นำ Markov chain มาใช้ใน Deterministic fluid model และได้ทำการวิเคราะห์หาค่าประสิทธิภาพในการกระจายข้อมูลของเครือข่ายแบบ Bittorrent และพบว่าค่าดังกล่าวมีค่าใกล้เคียง 1

Poisson Distribution

การกระจายค่าแบบ Poisson (Poisson distribution) เป็นการกระจายค่าแบบตามช่วงเวลาไม่ต่อเนื่อง (Discrete probability distribution) ซึ่งอธิบายความน่าจะเป็นที่เหตุการณ์ที่สังเกตจะเกิดขึ้นในช่วงเวลาที่จำกัด โดยมีอัตราการเกิดของเหตุการณ์เฉลี่ยอยู่ที่ค่าหนึ่งและมีความเป็นอิสระต่อเวลานับจากจุดที่เหตุการณ์เกิดขึ้นก่อนหน้านี้

Poisson Distribution ได้ค้นพบโดย Siméon-Denis Poisson และถูกใช้อย่างแพร่หลายในการอธิบายหลักการทางวิทยาศาสตร์ในเรื่องของอัตราการเข้ามาสู่ระบบ โดยโอกาสที่เหตุการณ์ที่ k จะเกิดขึ้น (k เป็นค่าเลขจำนวน (0, 1, 2, 3 ...)) อธิบายโดยสมการ

$$f(k; \lambda) = \frac{e^{-\lambda} \lambda^k}{k!} \quad (1)$$

โดยที่ λ เป็นค่าเลขจำนวนจริงมีค่าเท่ากับจำนวนของเหตุการณ์ที่เกิดขึ้นในช่วงเวลาที่กำหนด สมมติให้เหตุการณ์หนึ่งเกิดขึ้นในช่วงเวลาเฉลี่ยที่ 4 นาที ถ้าเราสังเกตภายในช่วงเวลา 10 นาที ก็จะกำหนดค่า $\lambda = 2.5(10/4)$

Exponential distribution

ในทฤษฎีความน่าจะเป็นและสถิติ การกระจายตัวแบบเอ็กโปเนนเชียล(Exponential distribution) ถูกใช้บ่อยในการอธิบายช่วงเวลาที่เกิดเหตุการณ์จะเกิดขึ้นในอัตราที่คงที่โดยที่สมการของโอกาสที่เหตุการณ์ใดๆที่จะเกิดในช่วงเวลา X มีค่าเป็น

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & , x \geq 0, \\ 0 & , x < 0. \end{cases} \quad (2)$$

เมื่อ λ เป็นค่าอัตราของระบบ และคำนวณอยู่ในช่วงเวลาดังแต่ $[0, \infty)$

ค่าExponential distribution ถูกใช้อธิบายถึงรูปแบบโอกาสการเกิดเหตุการณ์ของระยะเวลาที่เหตุการณ์ใดๆจะเกิดขึ้น เช่น เวลาที่ถ่านไฟฉายจะหมดอายุ เวลาที่รถคันใหม่จะถูกชน เวลาที่อนุภาคกัมมันตรังสีจะแตกตัวเหลือครึ่งหนึ่ง เป็นต้น

อุปกรณ์และวิธีการ

อุปกรณ์

1. คอมพิวเตอร์ Window XP Ram 1.5GB HDD 150 GB
2. อินเทอร์เน็ตแบบADSL ขนาดความเร็ว 2Mbps/512Kbs
3. โปรแกรม Bittorrent client (Azureus)
4. โปรแกรม Microsoft Office 2003
5. โปรแกรม Data pipe

วิธีการ

การจัดเก็บข้อมูล

ในเครือข่ายBittorrentที่นำมาศึกษาภายในวิทยานิพนธ์นี้เป็นข้อมูลทางสถิติของเครือข่าย Bittorrentหลายๆเครือข่าย ซึ่งได้ทำการรวบรวมข้อมูลดังกล่าวจากเว็บไซต์Tracker (ในงานวิจัยนี้ใช้ a.scarywater.net) ซึ่งเป็นTrackerสาธารณะสำหรับให้กลุ่มแปลงงานอิสระนำผลงานมาลงไว้เพื่อให้บุคคลอื่นที่สนใจสามารถใช้Bittorrentผ่านTrackerดังกล่าวในการดึงข้อมูล การวิจัยนี้ได้ทำการเก็บ ข้อมูลภาพยนตร์ที่มีรูปแบบการเก็บข้อมูลเป็นแบบaviที่มีเนื้อหาเป็นงานละครทีวีอนิเมชั่นที่ฉายใน โทรทัศน์โดยเริ่มเก็บข้อมูลจากวันที่ 11 มิถุนายน 2547 ถึงวันที่ 13 กรกฎาคม 2547 รวมระยะเวลา ประมาณ 1 เดือน โดยทำการเก็บข้อมูลจากหน้าWeb siteแสดงสถานะของ Tracker ดังแสดงไว้ใน ภาพที่ 13 ซึ่งเป็นตัวอย่างสถานะของเครือข่ายBittorrentแต่ละเครือข่าย โดยความหมายแต่ละ สถานะมีอธิบายไว้ดังนี้

time เป็นค่าอายุของเครือข่ายนับตั้งแต่เครือข่ายนั้นๆถือกำเนิดขึ้น มีหน่วยแสดงเป็น วัน(D) ชั่วโมง (h) นาที (m) ในที่นี้จะทำการแปลงค่านี้ให้อยู่ในรูปแบบนาที

ul คือจำนวนSeeder(เพียร์ที่มีสำเนาของข้อมูลครบถ้วน)ที่อยู่ในระบบ ณ เวลานั้นๆ

dl คือจำนวนDownloader(เพียร์ที่มีสำเนาของข้อมูลไม่ครบถ้วน และกำลังดึงข้อมูลที่ขาด
ออกจากSeederและDownloaderอื่นที่อยู่ในระบบเดียวกัน)ที่อยู่ในระบบ ณ เวลานั้นๆ

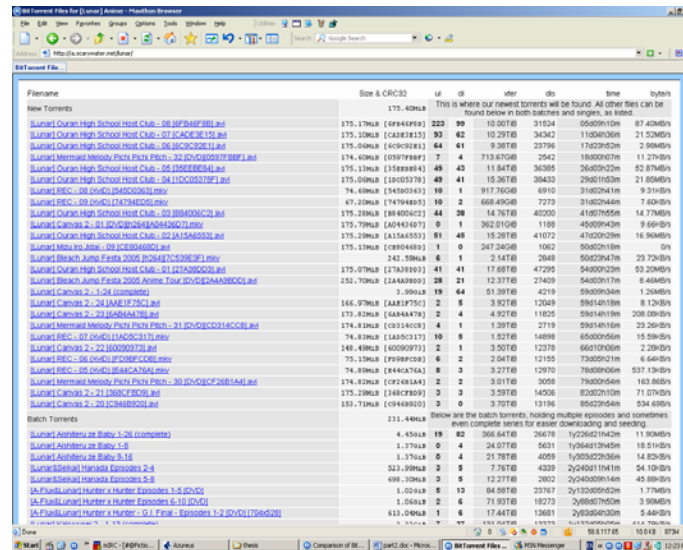
dl/s คือจำนวนเพียร์ที่ทำการดึงข้อมูลเสร็จสิ้นแล้วในช่วงเวลานั้น

byte/s เป็นค่าความเร็วในการจ่ายข้อมูลของเครือข่าย(Bandwidth)ในช่วงเวลานั้น

xfer เป็นค่าขนาดของข้อมูลทั้งหมดที่ระบบปล่อยออกมาจนถึงช่วงเวลานั้น

ในการเก็บข้อมูลจากTrackerจะใช้โปรแกรมOffline Web Browser ทำการเก็บข้อมูลโดย
อัตโนมัติ โดยมีระยะห่างในการเก็บข้อมูลในแต่ละช่วงเท่ากับ 20 นาที เหตุที่ไม่สามารถเก็บข้อมูล
ให้ช่วงเวลาใกล้กว่านี้เนื่องจากจะเป็นการดึงทรัพยากรของTrackerมากกว่าไปทำให้เสี่ยงต่ออาการ
trackerถล่มได้

หลังจากทำการเก็บข้อมูลเป็นเวลา1เดือนข้อมูลทั้งหมดซึ่งอยู่ในรูปแบบHtml จะถูกนำไป
ผ่านกระบวนการคัดกรองข้อมูลสถานะของเครือข่ายโดยโปรแกรมData pipe และนำผลลัพธ์ที่
ได้ นำเข้าบรรจุลงในฐานข้อมูลเพื่อทำการวิเคราะห์ต่อไป



ภาพที่ 13 ตัวอย่างหน้าสถานะของ Tracker

ผลและวิจารณ์

ผล

จากการเก็บข้อมูลเบื้องต้นได้ตัวอย่างข้อมูลเครือข่าย Bittorrent ทั้งหมด 240 เครือข่าย จัดเป็นเครือข่ายที่เกิดมาก่อนเริ่มทำการเก็บข้อมูล 202 เครือข่าย และเป็นเครือข่ายที่เกิดหลังจากเก็บข้อมูล 38 เครือข่าย ซึ่งได้ลงข้อมูลทางสถิติของจำนวน Seeder และ Downloader ในช่วงเวลาเริ่มต้น จนถึงเวลานาทีที่ 8000 ไว้ในภาคผนวก ก. ซึ่งในที่นี้ได้นำตัวอย่างมาแสดงทั้งหมด 6 ตัวอย่าง ดังตารางที่ 1

ตารางที่ 1 ข้อมูลเครือข่าย Bittorrent ตัวอย่าง 6 เครือข่าย

ตัวอย่างที่	ขนาดข้อมูล(MB)	ค่า λ สูงสุด	Finished peer ในวันที่ 5
1	176.89	18	18668
2	19.42	6	4856
3	230.50	11	14962
4	148	2	1325
5	175	2	1870
6	138.74	9	12264

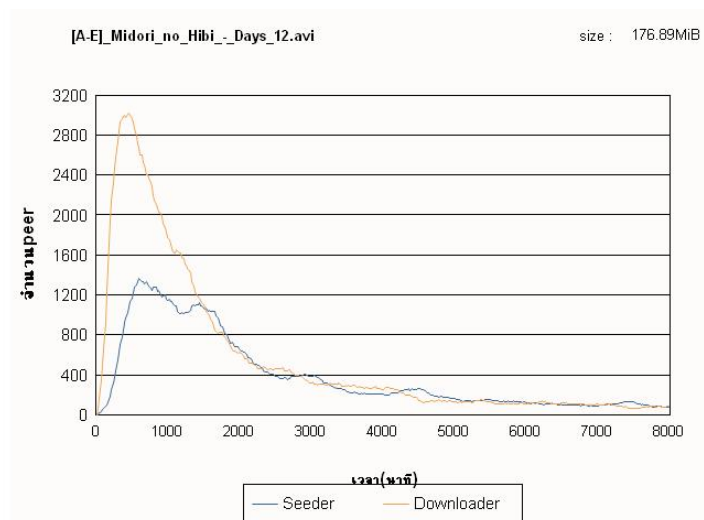
ลักษณะการเติบโตของ Bittorrent

จากภาพที่ 14 ถึงภาพที่ 20 พบว่าในช่วงวงจรชีวิตของเครือข่ายนั้นมีสภาพที่คล้ายคลึงกัน จะเห็นได้ว่าในช่วง 2-3 วันแรกเป็นช่วงที่มีจำนวนผู้ใช้งานอยู่ในระบบมากที่สุด โดย Downloader นั้นจะเพิ่มขึ้นอย่างรวดเร็วภายใน 6 ชั่วโมงแรก หลังจากนั้นจำนวน Seeder ก็จะเริ่มเพิ่มขึ้นมา ซึ่งจำนวน Seeder ที่เพิ่มขึ้นมานี้เป็นผลมาจากที่เมื่อเพียร์ที่เป็น Downloader ทำการดึงข้อมูลจนครบแล้วก็จะเปลี่ยนสถานะตัวเองกลายเป็น Seeder ซึ่งจะสังเกตได้ว่าจำนวน Seeder จะค่อยๆ เพิ่มขึ้นเรื่อยๆ แต่ไม่เพิ่มไปสูงเท่า Downloader นั้นเป็นเพราะเริ่มแรกสุดจะมี Seeder อยู่ในระบบอย่างน้อยแค่ 1 เพียร์ (สามารถมี Seeder เริ่มมากกว่า 1 ได้โดยสร้างสำเนาข้อมูลเพิ่มมาอีกหนึ่งเพียร์) และเพียร์หนึ่งๆจะมี

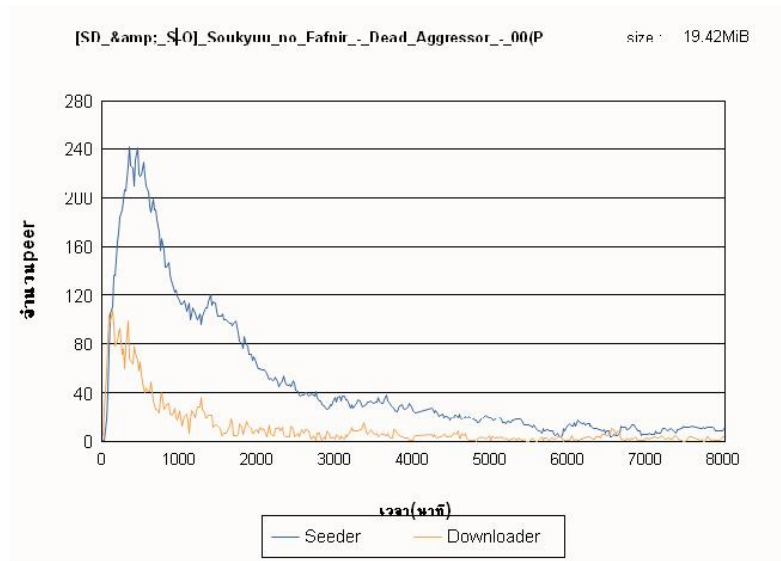
ความเร็วในการส่งข้อมูลจำกัด ดังนั้นเวลาที่Seederเริ่มแรกสุดสามารถส่งข้อมูลให้กับDownloader
ตัวอื่นจนสามารถเปลี่ยนตัวเองกลายเป็นSeederนั้นขึ้นอยู่กับขนาดของข้อมูลที่แจกจ่ายหารด้วย
ความเร็วในการส่งข้อมูลของSeederแรกเริ่ม ดังนั้นถ้าSeederเริ่มต้นมีความเร็วในการส่งข้อมูลต่ำจะ
ทำให้เครือข่ายนั้นมีจำนวนSeederที่น้อยในช่วงเวลาแรกๆ

ข้อมูลที่Downloaderได้รับมาสามารถใช้เป็นฐานข้อมูลที่ใช้ส่งให้กับDownloader อื่นได้
ทันที ทำให้ค่าAvailabilityของข้อมูลแต่ละส่วนนั้นมีค่าเพิ่มขึ้นเรื่อยๆตามเวลาที่ผ่านไป ทำให้
โอกาสที่Downloaderไม่สามารถรับข้อมูลที่ต้องการได้จากPeerอื่นๆมีโอกาสน้อยลง ซึ่งเป็นผลให้
Downloaderแต่ละตัวมีการรับส่งข้อมูลที่ต่อเนื่องซึ่งทำให้Downloaderเปลี่ยนสถานะเป็นSeederทำ
ได้เร็วขึ้น

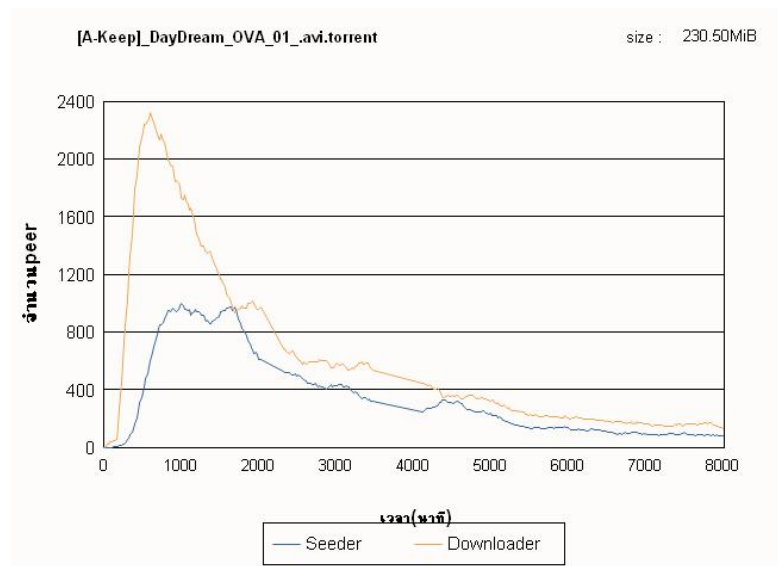
เมื่อผู้ใช้งานได้ข้อมูลที่ต้องการแล้วก็จะเริ่มทำการถอนตัวเองออกจากระบบซึ่งลักษณะ
ดังกล่าวได้ถูกระบุไว้ในการศึกษาของAdar and Huberman (2000) ซึ่งได้จำแนกผู้ใช้งานบางกลุ่ม
เป็นFree riding ซึ่งเมื่อผู้ใช้งานได้ข้อมูลที่ต้องการแล้วก็จะถอนตัวเองออกจากระบบทันที อย่างไรก็ตามการ
ถอนออกไปของผู้ใช้งานดังกล่าวไม่ได้กระทบต่อระบบมากนักเพราะขณะที่ผู้ใช้งานนั้นๆอยู่ใน
ระบบก็กระทำการส่งข้อมูลให้กับDownloaderอื่นๆได้ ซึ่งทำให้ระบบของBittorrentมีผลกระทบ
น้อยกว่าระบบอื่นๆที่จำเป็นต้องเป็นSeederก่อนถึงสามารถส่งข้อมูลให้ Downloader ตัวอย่างเช่น
Gnutella (Ripeanu, 2001)



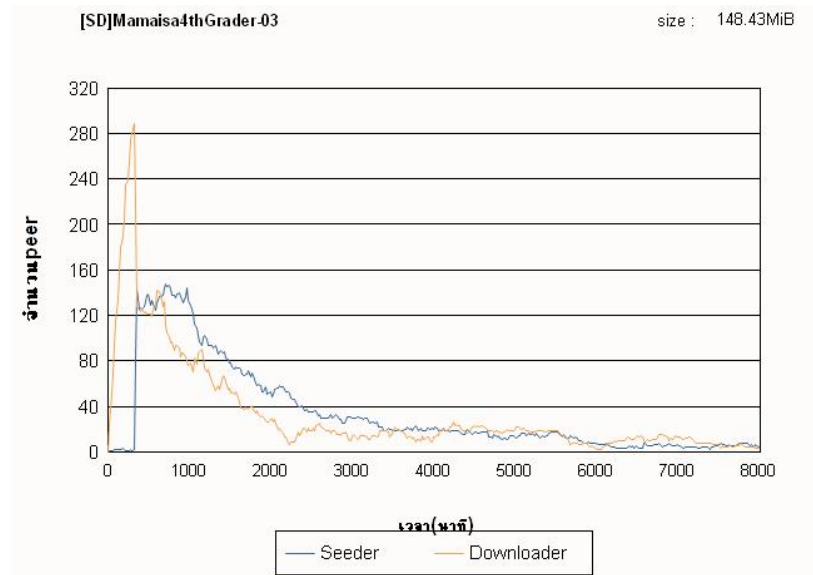
ภาพที่ 14 ข้อมูลเครือข่ายBittorrent ตัวอย่างที่ 1



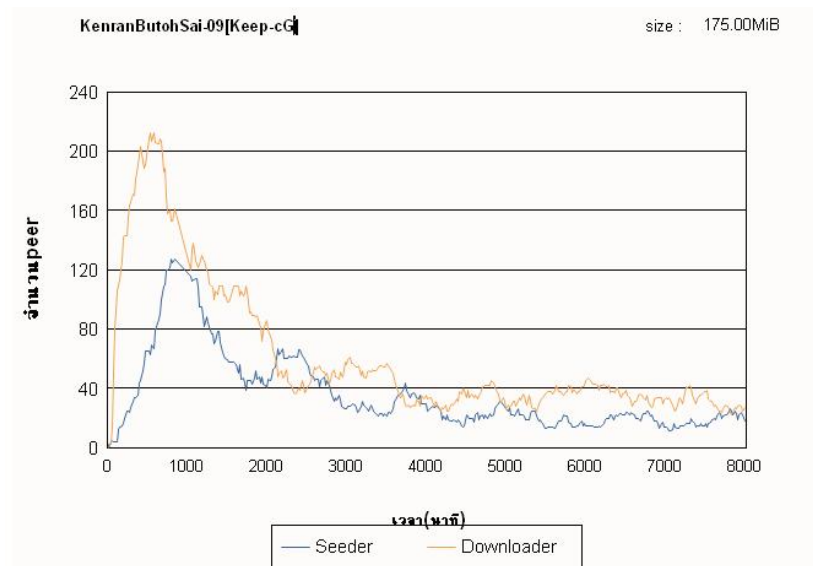
ภาพที่15 ข้อมูลเครือข่ายBittorrent ตัวอย่างที่ 2



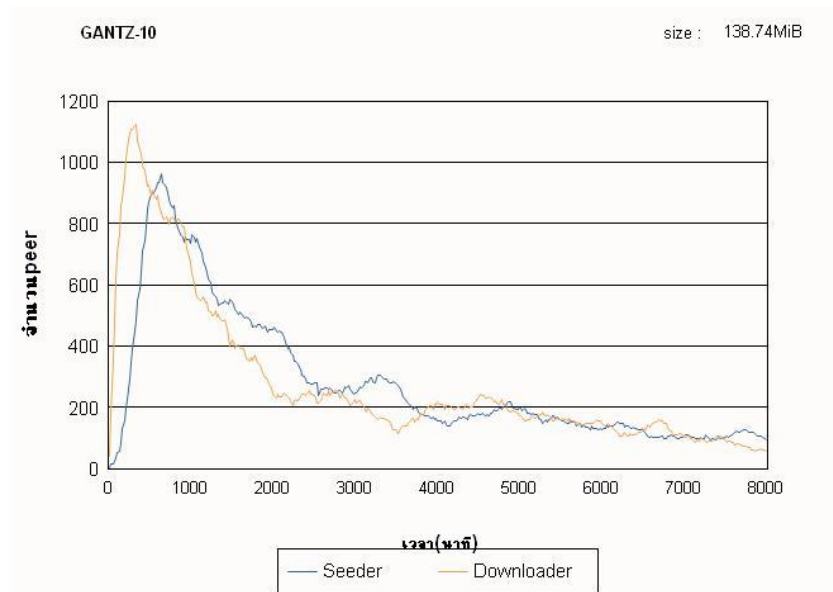
ภาพที่16 ข้อมูลเครือข่ายBittorrent ตัวอย่างที่ 3



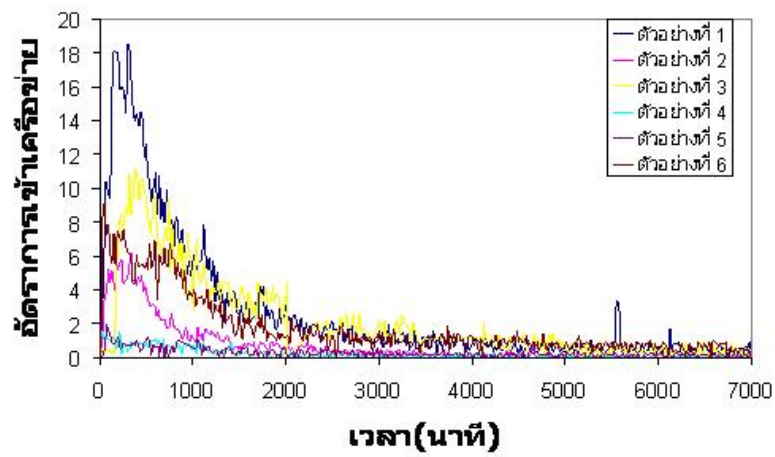
ภาพที่ 17 ข้อมูลเครือข่ายBittorrent ตัวอย่างที่ 4



ภาพที่ 18 ข้อมูลเครือข่ายBittorrent ตัวอย่างที่ 5



ภาพที่ 19 ข้อมูลเครือข่ายBittorrent ตัวอย่างที่ 6



ภาพที่ 20 แสดงค่าไของเพียร์เฉพาะ5วันแรกจาก6เครือข่าย

วิจารณ์

จากข้อมูลที่เก็บไว้ 240 ตัวอย่าง เราได้ยกตัวอย่างทำการวิเคราะห์เครือข่าย 6 เครือข่ายจากเครือข่ายทั้งหมด โดยรายละเอียดของแต่ละเครือข่ายได้แสดงไว้ในตารางที่ 1 นำมาเปรียบเทียบกับสมการทางคณิตศาสตร์ซึ่งให้สมมุติฐานรูปแบบ โครงสร้างของระบบเครือข่าย โดยใช้รูปแบบของ Fluid deterministic model (Veciana and Yang, 2003) โดยกำหนดให้มีตัวแปรดังต่อไปนี้

$D(T)$ เป็นจำนวนของพีเยอร์ที่ทำการดึงข้อมูล (Downloader) ที่อยู่ในระบบในช่วงเวลา T

$S(T)$ เป็นจำนวนของพีเยอร์ที่ทำหน้าที่แจกจ่ายข้อมูลอย่างเดียว (Seeder) ที่อยู่ในระบบในช่วงเวลา t

λ เป็นอัตราการเข้าสู่ระบบของพีเยอร์ (เฉพาะพีเยอร์ที่จะเป็น Downloader) ซึ่งในแต่ละเครือข่ายอัตรานี้จะมากหรือน้อยขึ้นกับความนิยมของข้อมูลของระบบนั้นๆ ว่าเป็นที่สนใจของผู้ใช้งานระบบมากหรือน้อยเท่าใด (ภาพที่ 10) ในที่นี้สมมุติฐานให้อัตราดังกล่าวมีรูปแบบการเข้าสู่ระบบเป็นแบบ Poisson Distribution โดยมีอัตราเข้าสู่ระบบ $= \lambda$ (Peer/นาที)

U เป็นอัตราการแจกจ่ายข้อมูล (Upload bandwidth) ของแต่ละพีเยอร์ (ในที่นี้ให้แต่ละพีเยอร์มีค่านี้เท่ากันทุกๆพีเยอร์) มีหน่วยเป็น MB/m (Mega Byte per Minute)

C เป็นอัตราการดึงข้อมูล (Download bandwidth) ของแต่ละพีเยอร์ (ในที่นี้ให้แต่ละพีเยอร์มีค่านี้เท่ากันทุกๆพีเยอร์) และ C มีค่า $\geq U$ (รูปแบบอินเทอร์เน็ตแบบ ADSL โดยปกติอัตราการส่งข้อมูล Downstream จะมีค่ามากกว่า Upstream) มีหน่วยเป็น MB/m (Mega Byte per minute)

θ เป็นค่าอัตราที่ Downloader จะออกจากระบบ โดยที่ช่วงเวลาที่ Downloader จะทำการถอนตัวออกจากระบบ (ใช้เวลาในการดึงข้อมูลนานเกินไป) ในที่นี้ตั้งสมมุติฐานให้รูปแบบช่วงเวลาที่ Downloader ใดๆ จะออกจากระบบมีรูปแบบเป็น Exponential distribution โดยมีค่า $mean = 1/\theta$ (นาที)

γ เป็นค่าอัตราที่Seederจะออกจากระบบ โดยที่ช่วงเวลาที่Seederใช้เวลาอยู่ในระบบเป็นช่วงระยะเวลาหนึ่งๆก่อนที่จะถอนตัวจากระบบ เราตั้งสมมุติฐานให้รูปแบบช่วงเวลาที่Seederจะออกจากระบบมีรูปแบบเป็น Exponential distribution โดยมีค่า mean = $1/\gamma$ (นาที)

η เป็นค่าแสดงประสิทธิภาพเฉลี่ยของDownloaderทั้งหมด ซึ่งมีค่ามาจากที่Downloaderใดๆสามารถทำการแจกจ่ายข้อมูลที่มีอยู่ในตัวให้แก่downloaderตัวอื่นๆที่ยังไม่มีข้อมูลในส่วนนั้นๆ มีค่าตั้งแต่ $[0, 1]$

Z เป็นขนาดของข้อมูลทำการแจกจ่ายในระบบมีหน่วยเป็น MB (Mega Byte)

ซึ่งเราสามารถอธิบายลักษณะทั่วไปของเครือข่ายBittorrentด้วยตัวแปรดังกล่าวได้ดังนี้

อัตราการส่งข้อมูลโดยรวมของระบบ(System bandwidth)

เนื่องจากอัตราการส่งข้อมูลโดยรวมจะมีค่าเท่ากับอัตราการส่งข้อมูลของแต่ละเพียร์ที่อยู่ในระบบรวมกัน โดยที่เพียร์ที่ทำหน้าที่เป็นSeederสามารถส่งเพิ่มข้อมูลได้ทุกส่วน ในขณะที่Downloaderสามารถส่งข้อมูลเฉพาะที่ตัวเองมีอยู่ในขณะนั้นเท่านั้น ดังนั้น

$$\text{อัตราการส่งข้อมูลโดยรวม} = U(\eta D(t) + S(t)) \quad (3)$$

แต่ถ้าหากว่าในกรณีที่มีDownloadมีจำนวนน้อย หรือรวมกันแล้วมีค่าการรับข้อมูลรวมกันทั้งหมดมีค่าน้อยกว่าอัตราการส่งข้อมูลโดยรวมแล้ว อัตราการส่งข้อมูลโดยรวมจะถูกจำกัดให้อยู่แค่ในระดับที่เท่ากับอัตราการรับข้อมูลโดยรวมทั้งหมดของDownload ($C \cdot D(t)$) ดังนั้น

$$\text{อัตราการส่งข้อมูลโดยรวมของระบบ} = \min(CD(t), U(\eta D(t) + S(t))) \quad (4)$$

อัตราการออกของDownloader

จากเบื้องต้นเรากำหนดให้ θ เป็นอัตราการออกของ Download ที่ออกจากระบบก่อนที่จะการดึงข้อมูลเสร็จ ซึ่งโอกาสที่Downloaderเหล่านี้จะออกจากระบบเกิดจากการที่ผู้ใช้งานยกเลิกการดึง

ข้อมูลเองเนื่องจากรอการดึงข้อมูลจากเพื่อนที่มีขนาดใหญ่ไม่ไหว ในที่นี้เรากำหนดให้ โอกาสที่ผู้ใช้งานจะยกเลิกมีค่าการกระจายแบบExponential โดยมีค่าMeanที่ $1/\theta$

อัตราการออกของDownloaderมีอีก 1 แบบนั้นคือกรณีที่Downloaderทำการดึงข้อมูลเสร็จสิ้นก็จะทำการเปลี่ยนสถานะตัวเองจากDownloaderไปเป็นSeeder ซึ่งถ้าให้ช่วงเวลาใดๆ δ โอกาสที่Downloader จะเปลี่ยนสถานะไปเป็นSeeder จะมีค่าเป็น $(\min(CD, U(\eta D + S))\delta)/Z$

อัตราการออกของSeeder

เป็นโอกาสที่Seederจะออกจากระบบ ซึ่งเกิดขึ้นเมื่อSeederได้ทำการส่งข้อมูลให้กับระบบเครือข่ายเป็นช่วงเวลาหนึ่งเป็นแบบสุ่ม ในที่นี้เรากำหนดให้ ช่วงเวลาดังกล่าวมีค่าการกระจายแบบExponential โดยมีค่ามีนที่ $1/\gamma$

อัตราของDownloader และ Seeder ที่อยู่ในระบบ

จากสมการและตัวแปรข้างต้น สามารถนำมาวิเคราะห์หาอัตราของDownloaderและSeederที่อยู่ในระบบในช่วงเวลาใดๆ โดยการใช้ Markov chain (Yang and Veciana, 2004) โดยพิจารณา Markov chain Matrix ของ DownloaderและSeeder (D, S) ขนาด $N*N^+$ โดยมีสมการการเปลี่ยนค่า (Rate Transition) ดังนี้

$$q((D, S), (D + 1, S)) = \lambda \quad (5)$$

$$q((D, S), (D - 1, S + 1)) = \min\left(\frac{CD}{Z}, \frac{U}{Z}(\eta D + S)\right) \quad (6)$$

$$q((D, S), (D, S - 1)) = \gamma S \quad (7)$$

$$q((D, S), (D - 1, S)) = \theta D \quad (8)$$

จากสมการที่ 5 ถึงสมการที่ 8 จะได้ว่า อัตราการเกิดของDownloaderเท่ากับอัตราการเข้าสู่ระบบของเพียร์ลบด้วยอัตราการออกจากระบบของDownloader และ อัตราการเกิดของSeederเท่ากับอัตราการเปลี่ยนสถานะของDownloaderไปเป็นSeederลบด้วยอัตราการออกของSeeder ซึ่งเขียนได้เป็น

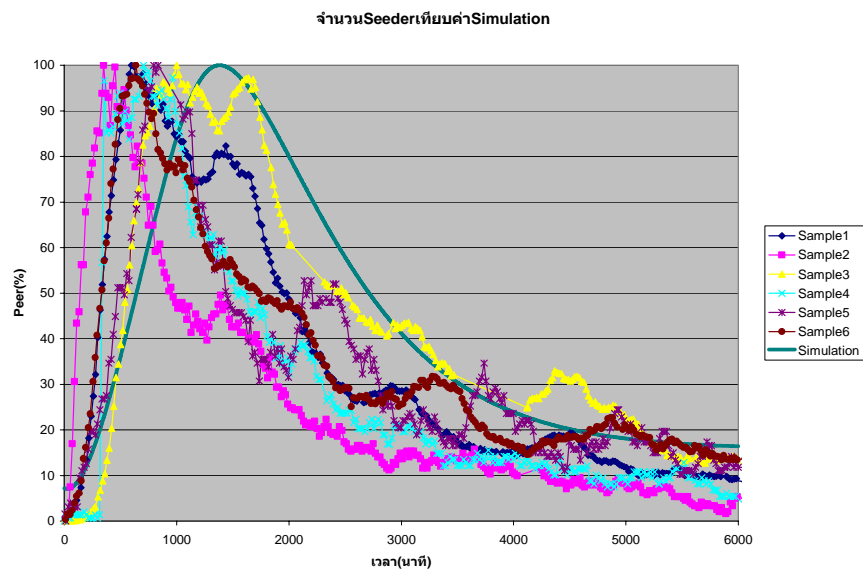
$$\frac{dD}{dt} = \lambda - \theta D(t) - \min\left\{\frac{CD(t)}{Z}, \frac{U}{Z}(\eta D(t) + S(t))\right\} \quad (9)$$

$$\frac{dS}{dt} = \min\left\{\frac{CD(t)}{Z}, \frac{U}{Z}(\eta D(t) + S(t))\right\} - \gamma S(t) \quad (10)$$

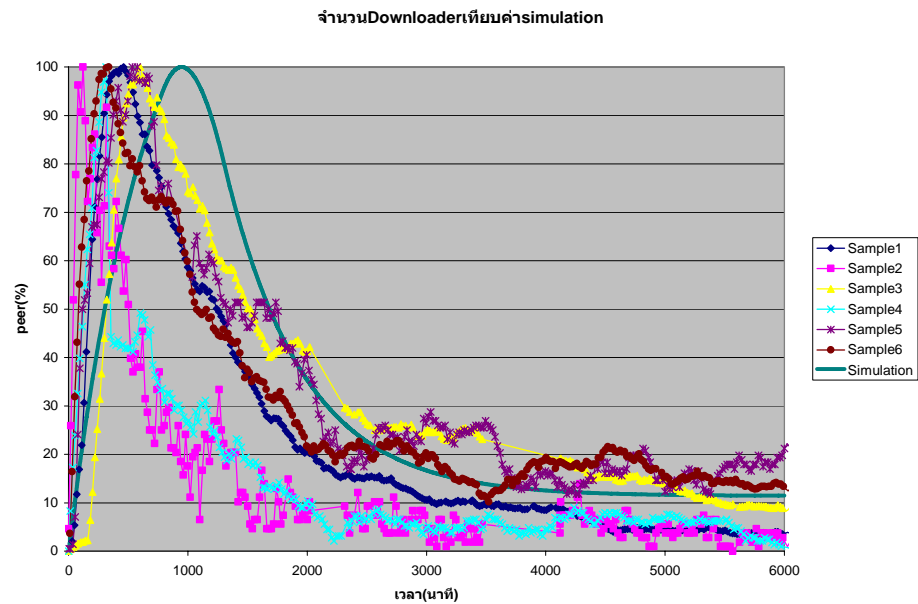
จากสมการ9และ10 เราจะเห็นได้ว่า จำนวนDownloaderที่เพิ่มขึ้นมีค่ามาจาก λ เพียงอย่างเดียวและจากการทดลองเก็บข้อมูลตัวอย่างจะเห็นว่าค่า λ ในแต่ละเครือข่ายในแต่ละช่วงเวลามีค่าที่ไม่เท่ากัน ซึ่งเราจะสังเกตได้ว่าในช่วงเวลาวันแรกตั้งแต่เครือข่ายเริ่มเกิด ค่า λ จะพุ่งสูงขึ้นอย่างรวดเร็ว (ภาพที่5) และจะเริ่มลดลงเมื่อใกล้พ้นวันนับตั้งแต่เวลาที่เครือข่ายเกิดขึ้น ซึ่งลักษณะดังกล่าวจะเกิดขึ้นเกือบทุกเครือข่ายที่ได้ทำการสังเกตตั้งแต่เริ่มต้น

วิเคราะห์สมการเปรียบเทียบค่าสถิติของDownloaderและSeeder

จากสมการ9และ10 นำมาทดสอบเปรียบเทียบกับข้อมูลทางสถิติของตัวอย่างเครือข่ายทั้ง6 ตัวอย่างที่ยกกล่าวมา โดยกำหนดให้ค่าตัวแปรในสมการทดลองมีค่าดังนี้ กำหนดให้ขนาดของแฟ้มข้อมูล 132.31MB เวลาในการจำลองค่าตั้งต้นนาทีที่1ถึง 6000 ให้ λ มีค่า = 6 ที่เวลา 1-800 และมีค่าลดลงจนถึง 0.5 ที่ระยะเวลา 1300 ให้ γ มีค่า 0.0015, $U = 0.0013$, $C = 5U$, $\theta = 0.001$



ภาพที่ 21 ข้อมูลเครือข่ายBittorrent เทียบค่าจำลอง (Seeder)



ภาพที่ 22 ข้อมูลเครือข่ายBittorrent เทียบค่าจำลอง (Downloader)

จากภาพที่21และ22 เป็นการนำค่าSeederและDownloaderของตัวอย่างทั้ง6ตัวอย่างมาทำการNormalized โดยใช้ค่าสูงสุดของแต่ละตัวอย่างมาเป็นตัวNormalized ค่าที่ออกมาจะเป็นค่าเปอร์เซ็นต์ของจำนวนSeeder และDownloader ต่อเวลา แล้วจึงนำมาเปรียบเทียบกับค่าที่จำลองออกมา

จะเห็นได้ว่าผลลัพธ์ที่ได้มาเทียบกับข้อมูลทางสถิติมีความใกล้เคียงกัน โดยช่วงเริ่มแรกของระบบมีค่าความผิดพลาดของช่วงข้อมูลอยู่ที่1-30% และมีความผิดพลาดน้อยลงเมื่อผ่านไประยะเวลาไป1-2วัน โดยมีค่าความผิดพลาด1-10%

สาเหตุที่ผลจากการทดสอบจำลองมีค่าความคลาดเคลื่อนในช่วงแรกนั้น เป็นเพราะในระบบของจริงมีอัตราการเข้ามาในระบบของDownloaderไม่คงที่ ซึ่งในช่วงแรกของระบบมีผู้ที่มีความสนใจในข้อมูลของเครือข่ายนั้นๆได้เข้ามาในระบบเป็นจำนวนมากซึ่งทำให้ในช่วง1วันแรกของระบบนั้นอัตราการเข้ามาที่สูง ซึ่งในการจำลองเราได้กำหนดให้มีค่าเท่ากับ 6 ตั้งแต่หน้าที่ที่ 1 ถึง หน้าที่ที่ 800 ซึ่งในระบบจริงๆนั้น อาจมีอัตราการเข้าที่มีค่าสูงมากเป็นบางระยะ ซึ่งทำให้เกิดความคลาดเคลื่อนขึ้น

ซึ่งเมื่อถึงจุดๆหนึ่งที่จำนวนSeederและDownloaderมีจำนวนมากพอที่ทำระบบอยู่ในสถานะอิ่มตัว (สถานะที่SeederและDownloaderสามารถกระจายข้อมูลที่มีอยู่ไปให้Downloaderทั้งหมดได้อย่างทั่วถึง) ค่าseederและDownloaderจะมีค่าใกล้เคียงกับค่าที่ทำกรจำลองออกมา ซึ่งการที่จำนวนDownloaderจะลดลงได้นั้นขึ้นอยู่กับค่าอัตราการส่งข้อมูลของระบบและขนาดของแฟ้มข้อมูล ถ้าขนาดไฟล์มีขนาดใหญ่มากการที่downloaderใดจะกลายเป็นSeederในตอนแรกนั้นจะมีค่าเท่ากับ $(U(\eta D(t) + S(t)) / Z)$ เนื่องจากระบบBittorrentตัวDownloaderสามารถทำการแจกจ่ายข้อมูลที่มีอยู่ไปให้Downloader ตัวอื่นๆได้ เมื่อเวลาผ่านไประบบมีSeederมากขึ้นและมีมากพอสำหรับDownloaderนั้น อัตราส่งข้อมูลจะจำกัดอยู่ที่ $CD(t)$

วิเคราะห์Steady State

เมื่อทำการวิเคราะห์สมการที่4ในรูปแบบ Steady State เราจะได้ว่าอัตราการเปลี่ยนแปลงในช่วงเวลาใดมีค่าเท่ากับ0 ซึ่งจะได้ว่า

$$\frac{dD(t)}{dt} = \frac{dS(t)}{dt} = 0 \quad (11)$$

จาก(11) แทนค่าใน (9) และ (10) ได้

$$0 = \lambda - \theta \bar{D} - \min\left\{\frac{C\bar{D}}{Z}, \frac{U}{Z}(\eta\bar{D} + \bar{S})\right\} \quad (12)$$

$$0 = \min\left\{\frac{C\bar{D}}{Z}, \frac{U}{Z}(\eta\bar{D} + \bar{S})\right\} - \gamma\bar{S} \quad (13)$$

จากสมการที่12และ13 สามารถพิจารณาได้2กรณี นั่นคือกรณีที่ค่าอัตราการส่งข้อมูลโดยรวมของระบบจะขึ้นอยู่กับค่าใดระหว่าง CD หรือค่า $(U(\eta D + S))$

กรณี $C\bar{D} \leq U(\eta\bar{D} + \bar{S})$ จาก (12) และ (13) จะได้

$$\bar{D} = \frac{\lambda}{\frac{C}{Z}\left(1 + \frac{\theta Z}{C}\right)} \quad (14)$$

$$\bar{S} = \frac{\lambda}{\gamma \left(1 + \frac{\theta Z}{C}\right)} \quad (15)$$

แทนค่าสมการ $C\bar{D} \leq U(\eta\bar{D} + \bar{S})$ ด้วย (14) และ (15) จะได้

$$\frac{Z}{C} \geq \frac{1}{\eta} \left(\frac{Z}{U} - \frac{1}{\gamma} \right) \quad (16)$$

กรณี $C\bar{D} \geq U(\eta\bar{D} + \bar{S})$ จาก (12) และ (13) จะได้

$$\bar{D} = \frac{\lambda}{V \left(1 + \frac{\theta}{V}\right)} \quad (17)$$

$$\bar{S} = \frac{\lambda}{\gamma \left(1 + \frac{\theta}{V}\right)} \quad (18)$$

$$\text{เมื่อ } \frac{1}{V} = \frac{1}{\eta} \left(\frac{Z}{U} - \frac{1}{\gamma} \right)$$

แทนค่าสมการ $C\bar{D} \geq U(\eta\bar{D} + \bar{S})$ ด้วย (11) จะได้

$$\frac{Z}{C} \leq \frac{1}{V} \quad (19)$$

จากกรณีทั้ง 2 ถ้าให้ $\frac{1}{\beta} = \max \left\{ \frac{Z}{C}, \frac{1}{\eta} \left(\frac{Z}{U} - \frac{1}{\gamma} \right) \right\}$ และแทนค่าในสมการที่ 9 และ 10 จะได้

$$\bar{D} = \frac{\lambda}{\beta \left(1 + \frac{\theta}{\beta}\right)} \quad (20)$$

$$\bar{S} = \frac{\lambda}{\gamma(1 + \frac{\theta}{\beta})} \quad (21)$$

เมื่อพิจารณา Little's law โดยให้ $(\lambda - \theta\bar{D})/\lambda$ เป็นอัตราส่วนที่ Downloader จะกลายเป็น Seed และ $\lambda - \theta\bar{D}$ เป็นอัตราเฉลี่ยที่ Downloader จะเป็น Seeder แทนค่าสมการจะได้

$$\frac{\lambda - \theta\bar{D}}{\lambda} \bar{D} = (\lambda - \theta\bar{D})T \quad (22)$$

นำสมการที่ 20 มาแทนค่าใน 22 เราจะได้

$$T = \frac{1}{\theta + \beta} \quad (23)$$

จากสมการที่ 23 จะเห็นว่าค่าเวลาเฉลี่ย (T) ที่ Downloader ทำการดึงข้อมูลตั้งแต่ต้นจนรับข้อมูลครบถ้วนนั้น มีปัจจัยขึ้นกับค่า θ และ β โดยที่ถ้าอัตราการออกของ Downloader (θ) และ Seeder (γ) มีค่าน้อยเท่าใดก็จะทำให้เวลาที่ Downloader ใช้ในการดึงข้อมูลน้อยลงเท่านั้น

จะเห็นว่าอัตราการเข้าสู่ระบบ (λ) ของ Downloader ไม่มีผลกับเวลาที่ใช้ในการดึงข้อมูลไม่ว่าระบบมีจำนวน downloader ในระบบมากเพียงใดแต่จะขึ้นกับอัตราการส่งข้อมูลของ Downloader และ Seeder กับอัตราการออกของ Seeder และ Downloader โดยมีผลโดยตรงกับเวลาที่ใช้ในการดึงข้อมูล ถ้าอัตราการออกของ Seeder มีค่าต่ำแสดงว่า Seeder อยู่ในระบบนานขึ้นทำให้ค่า Upload Bandwidth โดยรวมของระบบก็จะมีค่าสูงขึ้น และทำให้ค่าความความเร็วในการส่งข้อมูลเฉลี่ยมีค่ามากขึ้น แต่ทั้งนี้ก็ไม่สามารถเกินไปกว่าค่า Download Bandwidth โดยรวมของระบบได้

สรุปและข้อเสนอแนะ

สรุป

งานวิจัยนี้ได้นำเสนอการวิเคราะห์ความนิยมของข้อมูลกับประสิทธิภาพของเครือข่าย แจกจ่ายข้อมูลBittorrent โดยทำการเก็บข้อมูลจากตัวอย่างเครือข่าย240เครือข่ายที่มีความนิยมในข้อมูลที่แตกต่างกัน เราพบว่า

รูปแบบการขยายตัวของเครือข่ายไม่ว่าความนิยมของข้อมูลจะมีค่ามากหรือน้อยเท่าใดก็จะมีลักษณะที่เหมือนกัน คือมีจำนวนDownloaderเพิ่มสูงขึ้นมากในช่วงวันแรกของระบบเนื่องจากเป็นช่วงที่มีผู้ที่มีความสนใจข้อมูลเริ่มเข้ามาสู่ระบบเป็นจำนวนมาก และเริ่มมีSeeder เพิ่มสูงขึ้นมาตาม (Downloader ได้รับข้อมูลครบจะเปลี่ยนสถานะตัวเองเป็นSeeder) และเมื่อผ่านไประยะเวลาหนึ่ง (ประมาณ 1 วัน) ทั้งSeeder และ Downloader จะเริ่มลดลง (Seeder ออกจากระบบ และจำนวนผู้ที่สนใจในข้อมูลเริ่มลดลง)

เมื่อวิเคราะห์รูปแบบของเครือข่ายและหาสมการของเวลาเฉลี่ยที่Downloaderใช้ในการดึงข้อมูลก็พบว่าอัตราการเข้าสู่ระบบของDownloader ไม่มีผลกับเวลาที่ใช้ในการดึงข้อมูลไม่ว่าระบบมีจำนวนDownloaderมากเพียงใด แต่จะขึ้นกับอัตราการส่งข้อมูลของDownloader และ Seeder กับอัตราการออกของ Seeder ซึ่งมีผลโดยตรงกับเวลาที่ใช้ในการดึงข้อมูล ซึ่งพบว่าถ้า Seeder ไม่ได้ ออกจากระบบเลยหรือDownloaderตัดสินใจที่จะออกจากระบบน้อยลงจะทำให้อัตราการส่งข้อมูลโดยรวมเพิ่มขึ้น เป็นเพราะเมื่อมีSeederในระบบมากขึ้นจะทำให้มีจำนวนBandwidthในการส่งข้อมูลเพิ่มมากขึ้น ทำให้Downloaderแต่ละตัวได้รับข้อมูลในความเร็วที่เร็วขึ้น แต่ทั้งนี้ก็จะไม่เร็วไปกว่าความเร็วสูงสุดที่Downloaderนั้นๆสามารถรับได้ และถ้าDownloaderตัดสินใจอยู่ในระบบมากขึ้นก็จะทำให้มีจำนวนBandwidthในการส่งข้อมูลเพิ่มขึ้นมากเช่นกัน แต่Downloaderก็จะทำการรับข้อมูลในเวลาเดียวกันด้วย

เมื่อนำสมการที่9และ10ซึ่งเป็นสมการการเปลี่ยนแปลงของจำนวนSeederและDownloader เทียบกับเวลา โดยนำมาเปรียบเทียบกับตัวอย่างเครือข่ายทั้ง6ตัวอย่าง พบว่าลักษณะของกราฟมีค่าใกล้เคียงกัน โดยในช่วงแรกของระบบ มีค่าความผิดพลาดของจำนวนอยู่ที่ 1-30% ซึ่งความผิดพลาดเป็นเพราะในแต่ละตัวอย่างมีช่วงเวลาที่ผู้ที่สนใจในข้อมูลนั้นเข้าสู่ระบบ(λ)เป็นจำนวน

มากอยู่ช่วงเวลานึง และช่วงเวลาดังกล่าวแต่ละเครือข่ายไม่เหมือนกัน บางเครือข่ายใช้เวลานาน และบางเครือข่ายใช้เวลาสั้นแต่ทั้งหมดจะกินเวลาไม่เกิน 1 วัน และจะเริ่มลดลงถึงค่าหนึ่ง ซึ่งค่าที่จำลองกำหนดให้หน้าที่ที่ 1 ถึง 600 มีค่า λ เท่ากับ 6 และจะเริ่มลดลงเหลือ 0.5 ในหน้าที่ที่ 1300 ซึ่งเป็นช่วงอ้อมตัวของระบบที่ค่า λ ของระบบจะลดลงถึงระดับต่ำ (ผู้ที่สนใจในข้อมูลได้รับข้อมูลไปแล้ว และมีผู้ที่สนใจข้อมูลคนใหม่เข้ามาในระบบอย่างประปราย) ซึ่งช่วงนี้ค่าจำลองเมื่อเทียบกับตัวอย่างเครือข่ายทั้ง 6 เครือข่าย พบว่ามีช่วงค่าความผิดพลาดอยู่ที่ 1-10%

ข้อเสนอแนะ

เมื่อเราสามารถวิเคราะห์และสรุปรูปแบบโครงสร้างของการดึงข้อมูลในระบบ Peer-to-Peer ได้แล้ว เราสามารถนำเอารูปแบบโครงสร้างไปใช้ประโยชน์ได้หลายทาง ตัวอย่างเช่น เราสามารถวิเคราะห์ได้ว่ามีโอกาสเท่าไรที่เพียร์ใดๆสามารถเข้าไปให้บริการการดึงข้อมูลได้ทันทีโดยไม่ต้องรอคิว หรือเราสามารถคำนวณได้ว่าเวลารอเฉลี่ยของคิวเป็นเท่าใด ถ้าในระบบมีสำเนาของข้อมูลและความเร็วในการขนส่งอยู่ในระดับที่กำหนด โดยสามารถหาคำตอบได้จากรูปแบบโครงสร้างที่ได้จากงานวิจัยนี้

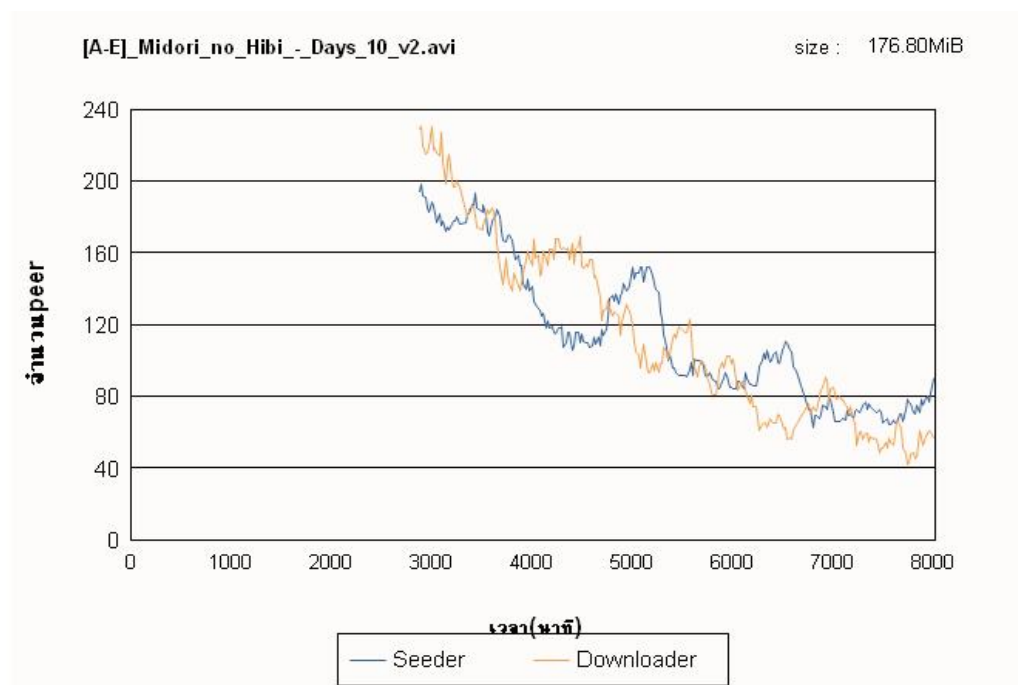
เอกสารและสิ่งอ้างอิง

- Adar, B. and B.A. Huberman 2000. **Free riding on gnutella**. Technical report, Xerox PARC, 10 Aug. 2000
- Cohen, B. 2003. **Incentives Build Robustness in BitTorrent**. <http://bitconjurer.org/BitTorrent>.
- Izal, M., G. Urvoy-Keller., E.W. Biersack., P.A. Felber., A. Hamra., and L Garc_es-Erice. 2004. Dissecting BitTorrent: Five Months in a Torrent's Lifetime, *In PAM2004 5. ed.* The 5th annual Passive & Active Measurement Workshop
- Kant, K. 2003. An Analytic Model for peer to peer File sharing networks, *In IEEE* .
- Ripeanu, M. November 2001. Peer-to-peer Architecture Case Study “Gnutella Network”, **Technical report TR-2001-26**. University of Chicago.
- Schollmeier, R. and G. Schollmeier 2002. Why Peer-to-Peer (P2P) Does Scale: An Analysis of P2P Traffic Patterns, p. 112. *In Second International Conference on Peer-to-Peer Computing (P2P'02) 2. ed.*
- Tutschku, K. 2004. A Measurement-based Traffic Profile of the eDonkey Filesharing Service, *In PAM2004 5. ed.* The 5th annual Passive & Active Measurement Workshop
- Xiangying, Y. and G.D. Veciana 2004. Service Capacity of Peer to Peer Networks, *In INFOCOM* .
- Yazti, N., D. Zeinalipour and T. Folias 2002. Quantitative Analysis of the Gnutella Network Traffic, **TR-CS-89**. Dept. of Computer Science, University of California, Riverside.

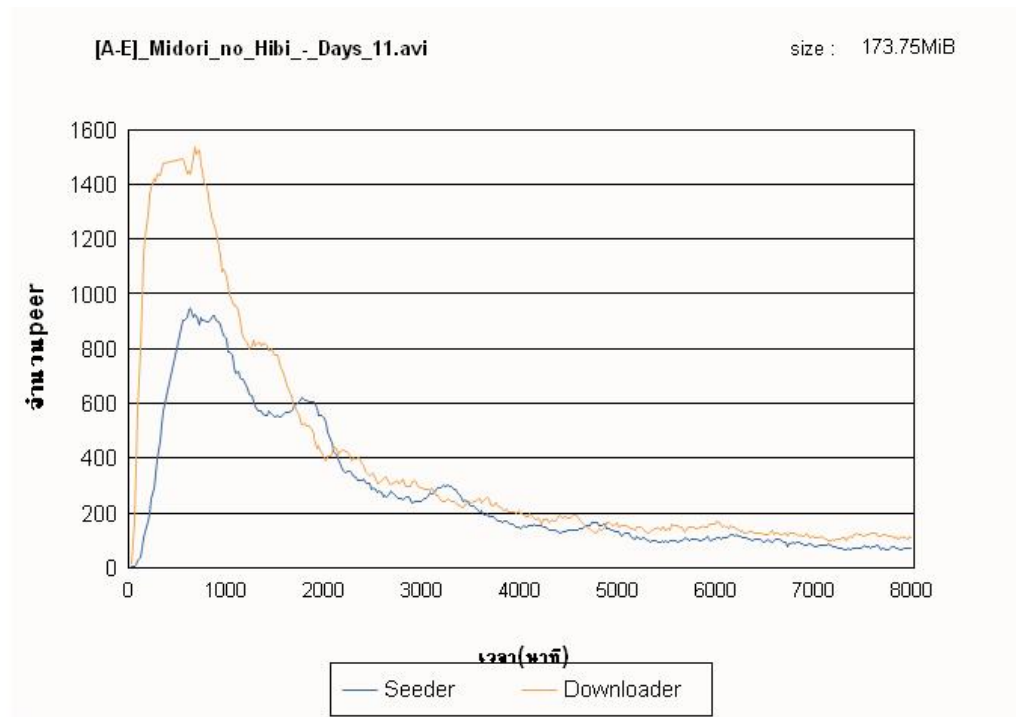
Zhao, B.Y., J.D. Kubiatowicz and A.D. Joseph 2001. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. UC Berkeley (apr): Abstract No. UCB/CSD-01-1141.

Zihui, Ge., D.R. Figueriredo., S. Jaiswal., J. Kurose and D. Towsley 2003. Modeling Peer-Peer File Sharing System, *In IEEE* .

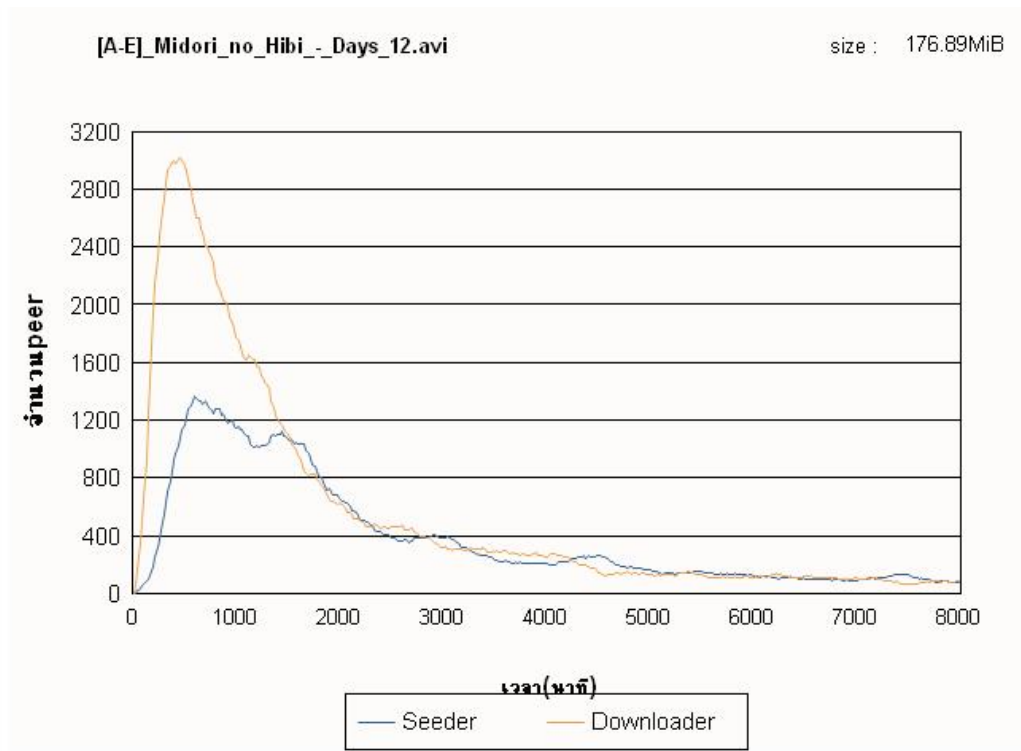
ภาคผนวก



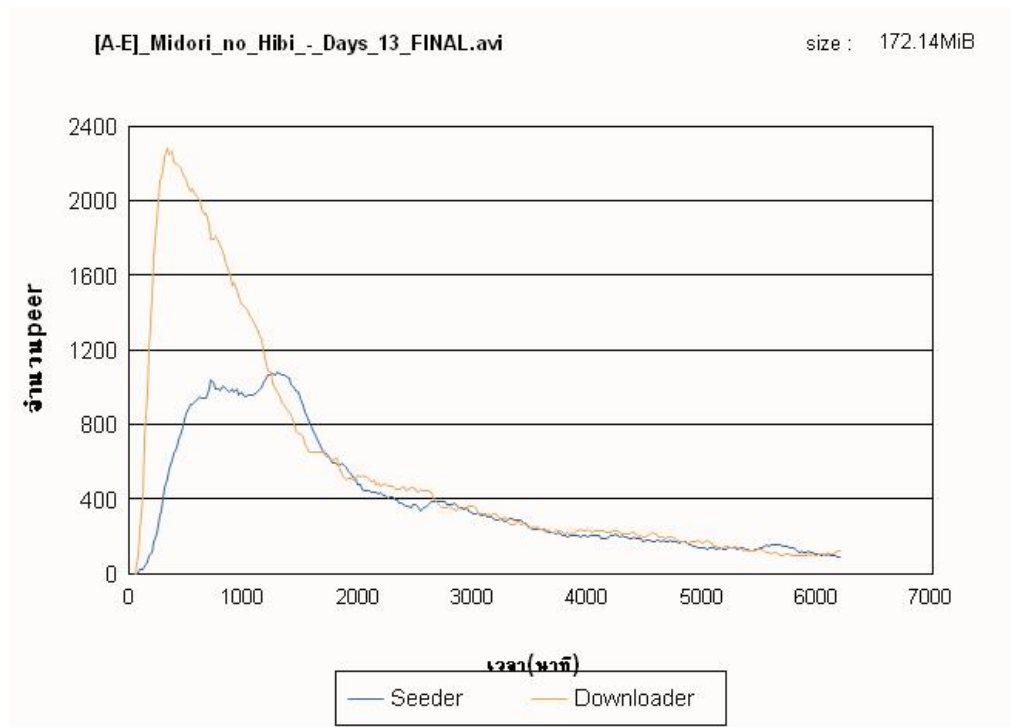
ภาพผนวกที่ ก.1 ตัวอย่างเครือข่ายBittorrentที่1



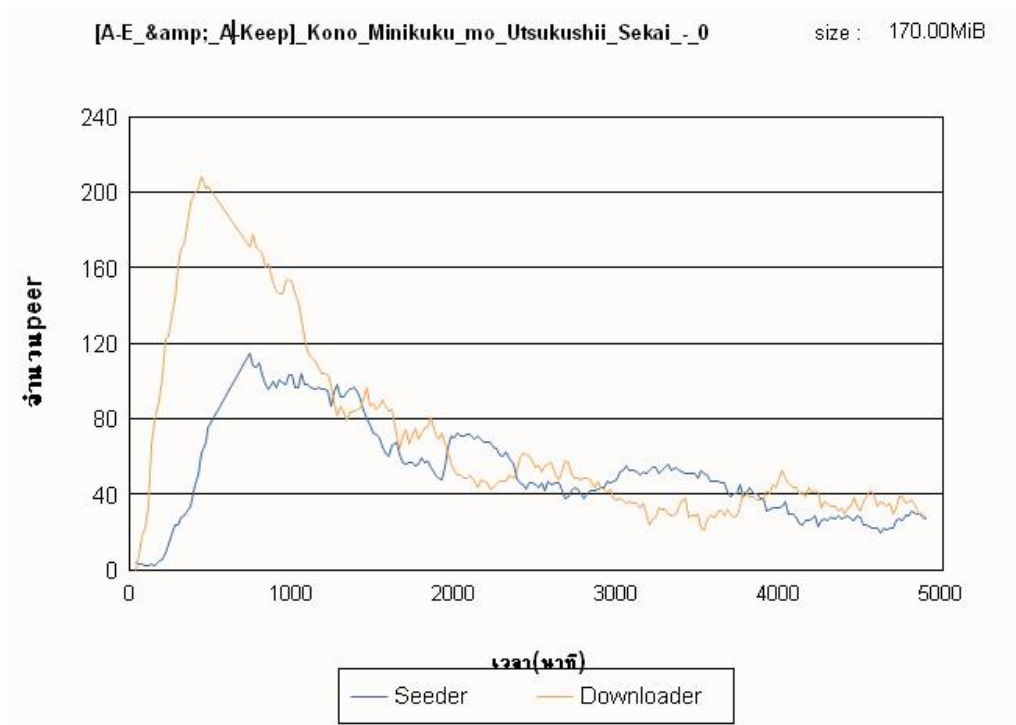
ภาพผนวกที่ ก.2 ตัวอย่างเครือข่ายBittorrentที่2



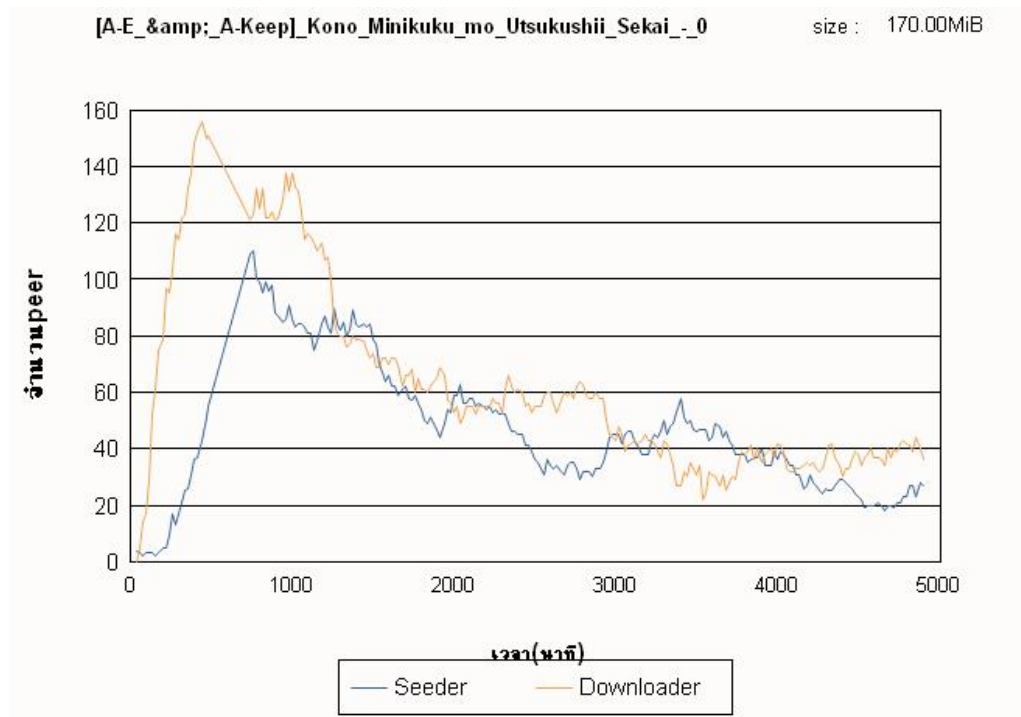
ภาพผนวกที่ ก.3 ตัวอย่างเครือข่ายBittorrentที่3



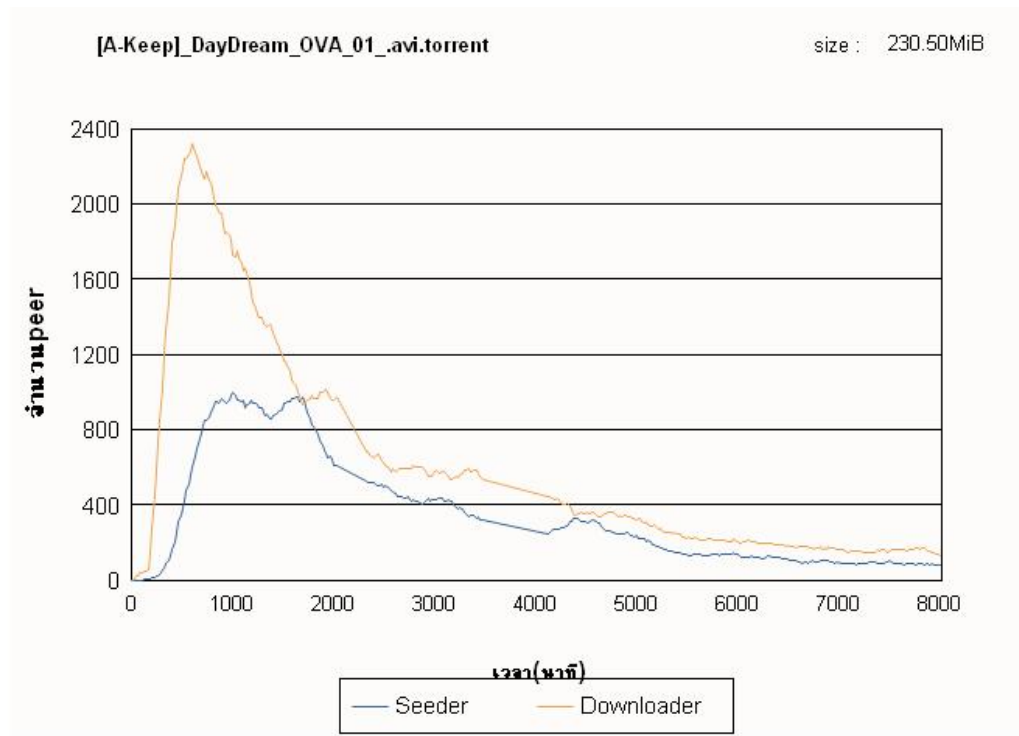
ภาพผนวกที่ ก.4 ตัวอย่างเครือข่ายBittorrentที่4



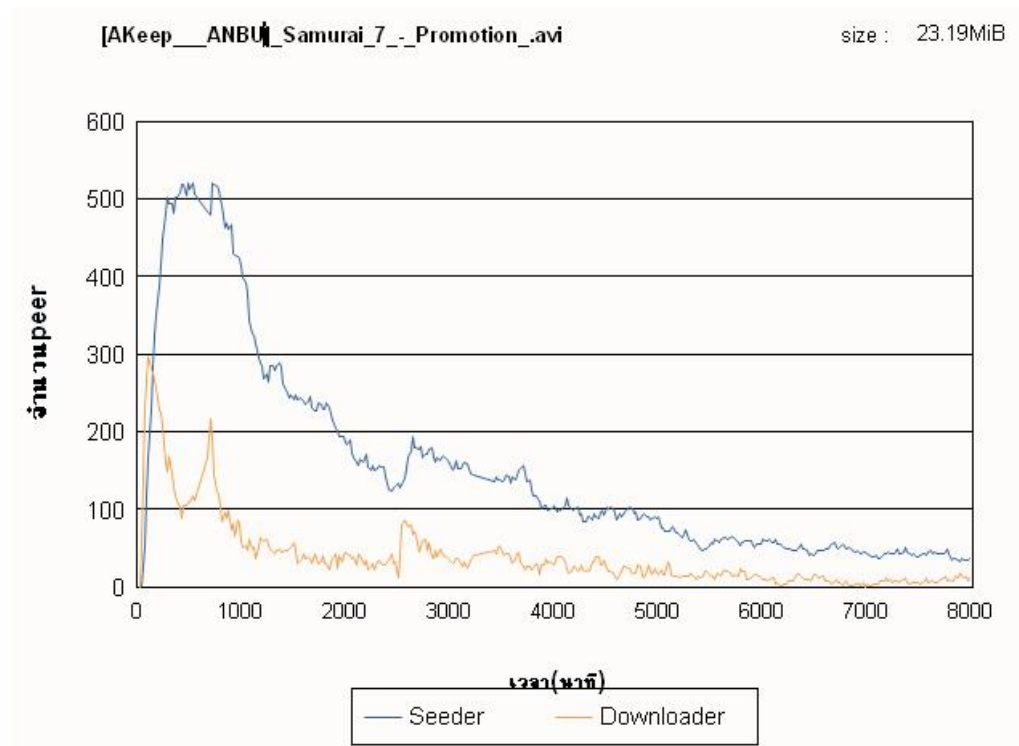
ภาพผนวกที่ ก.5 ตัวอย่างเครือข่ายBittorrentที่5



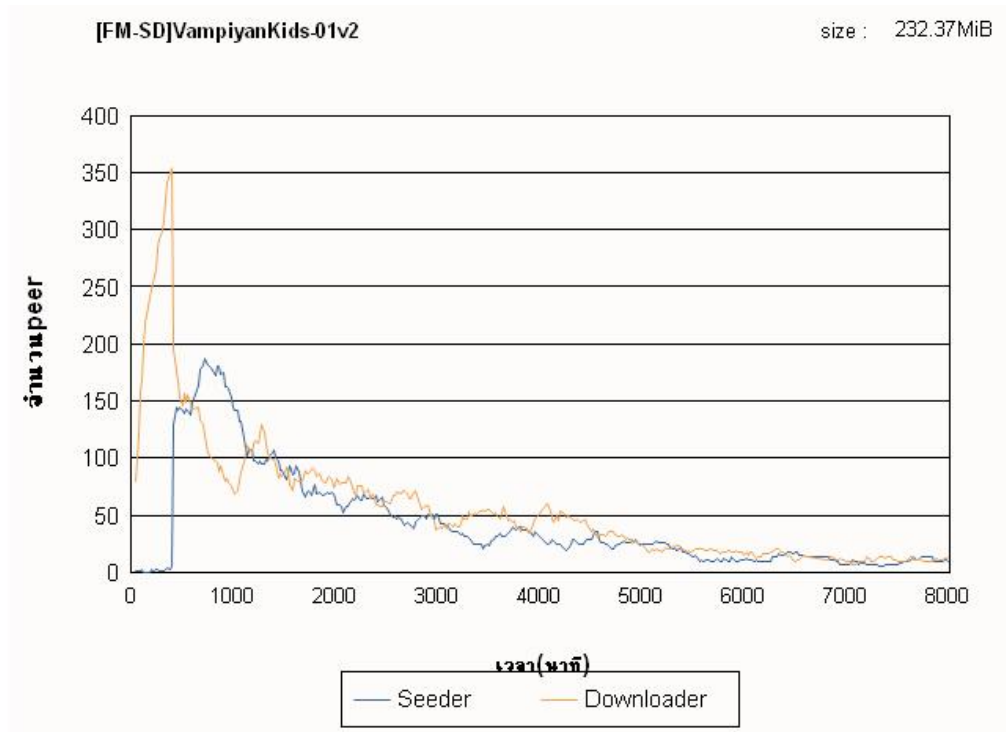
ภาพผนวกที่ ก.6 ตัวอย่างเครือข่ายBittorrentที่6



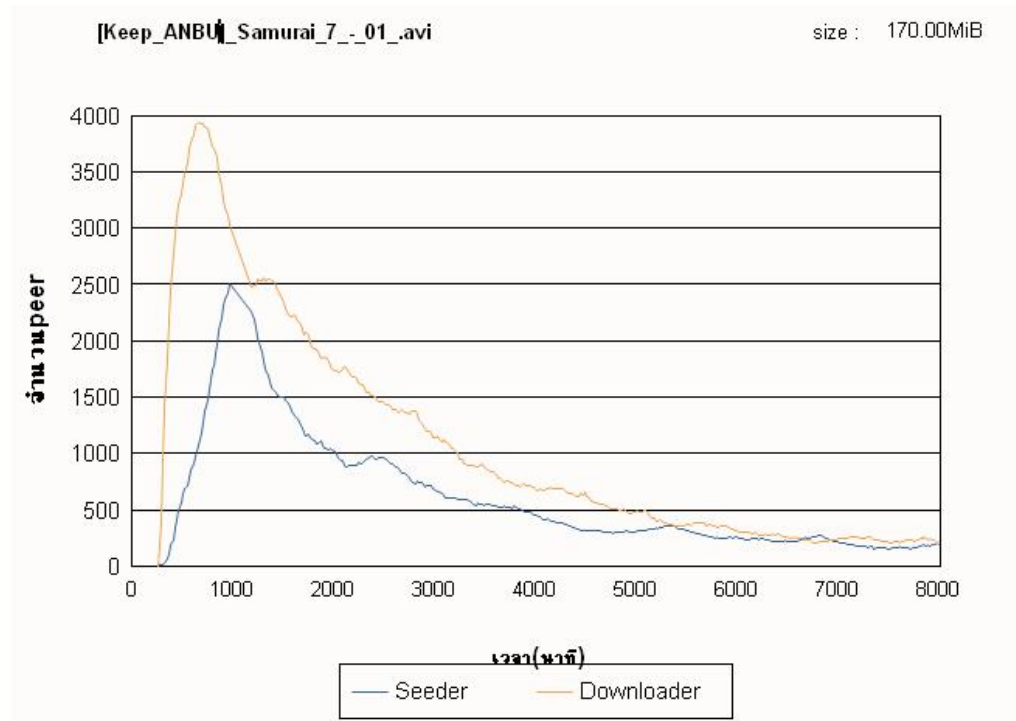
ภาพผนวกที่ ก.7 ตัวอย่างเครือข่ายBittorrentที่7



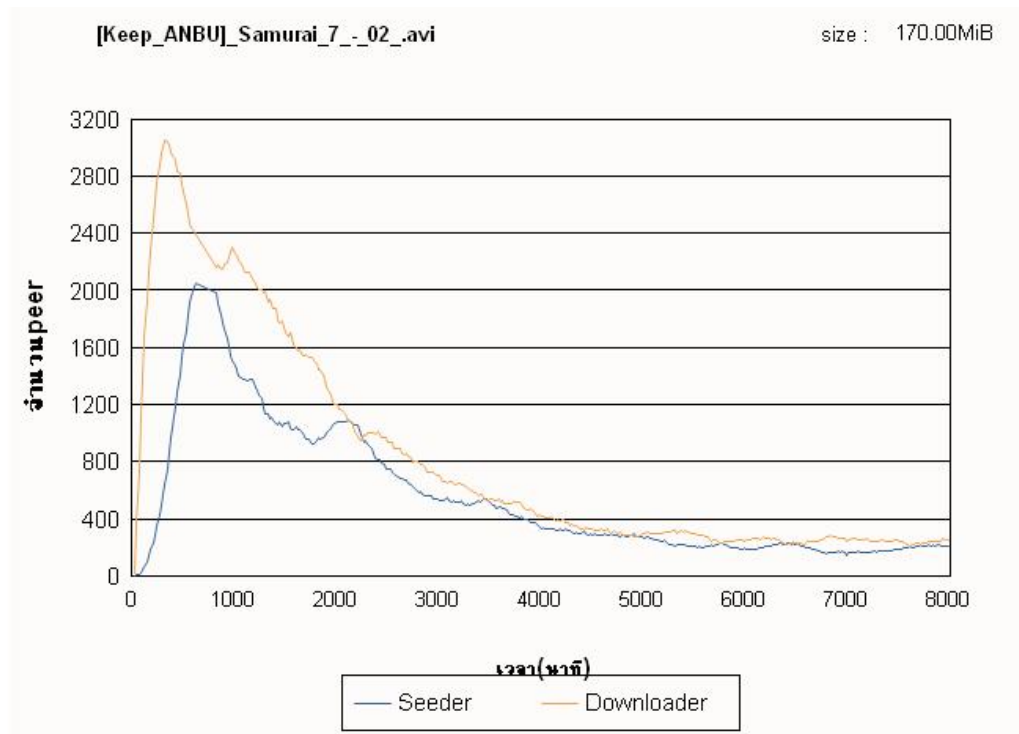
ภาพผนวกที่ ก.8 ตัวอย่างเครือข่ายBittorrentที่8



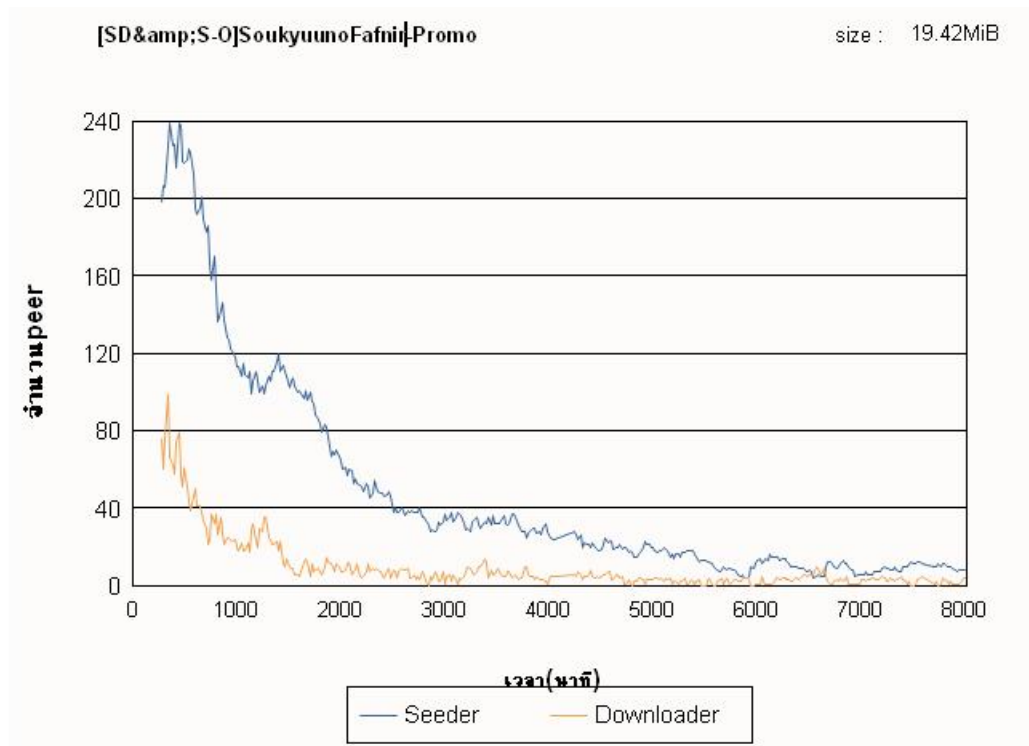
ภาพผนวกที่ ก.9 ตัวอย่างเครือข่ายBittorrentที่9



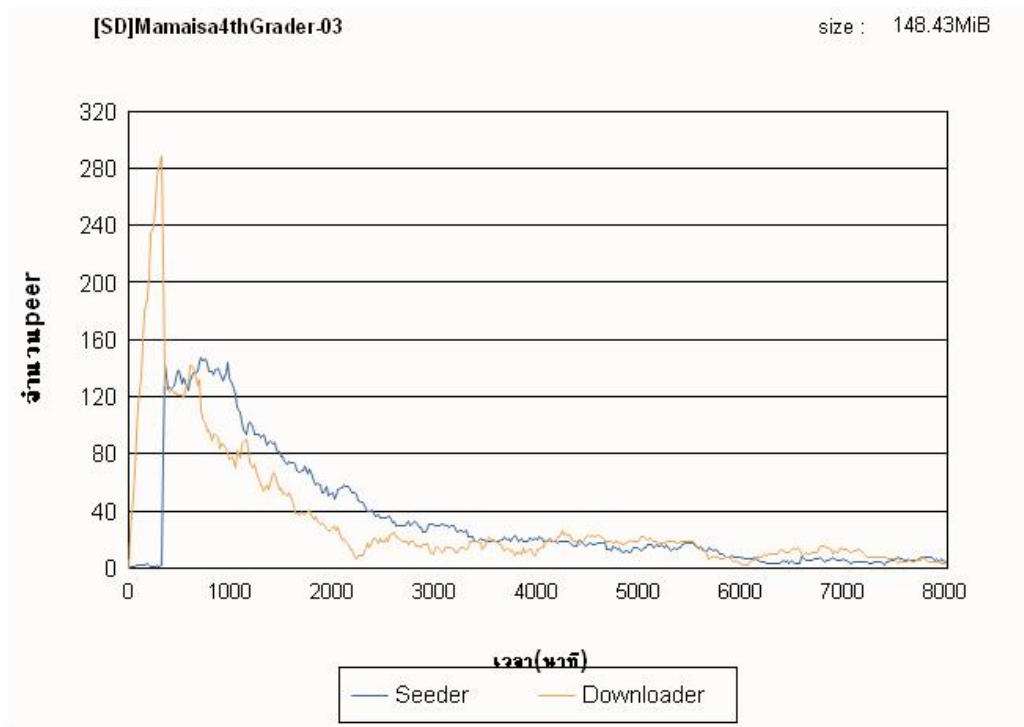
ภาพผนวกที่ ก.10 ตัวอย่างเครือข่ายBittorrentที่10



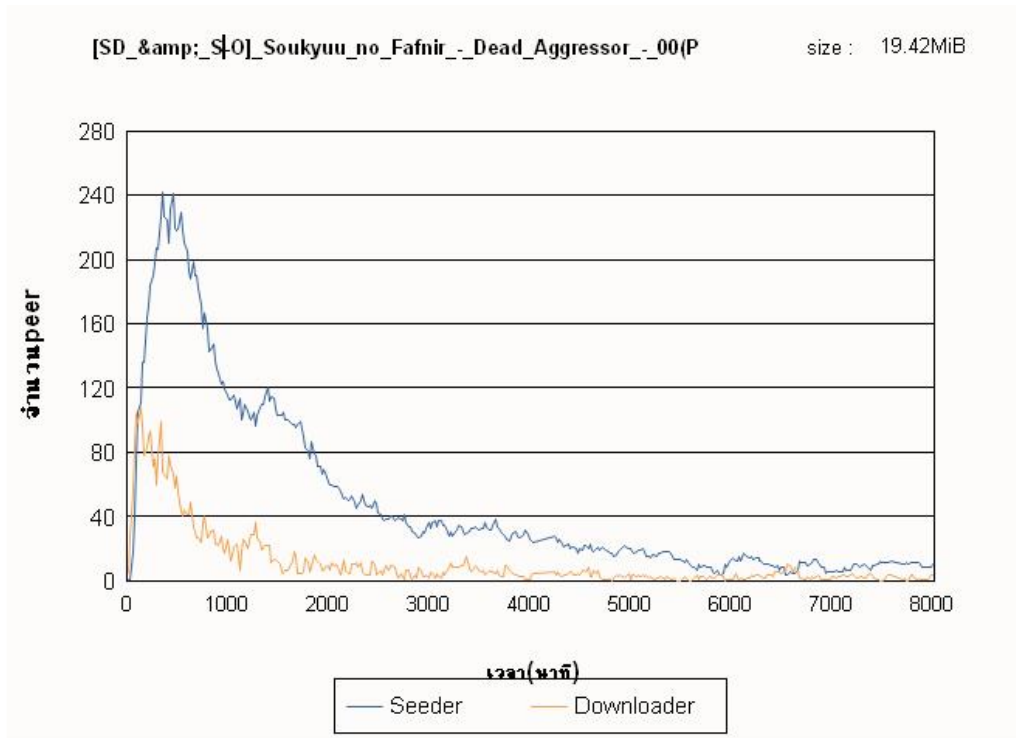
ภาพผนวกที่ ก.11 ตัวอย่างเครือข่ายBittorrentที่11



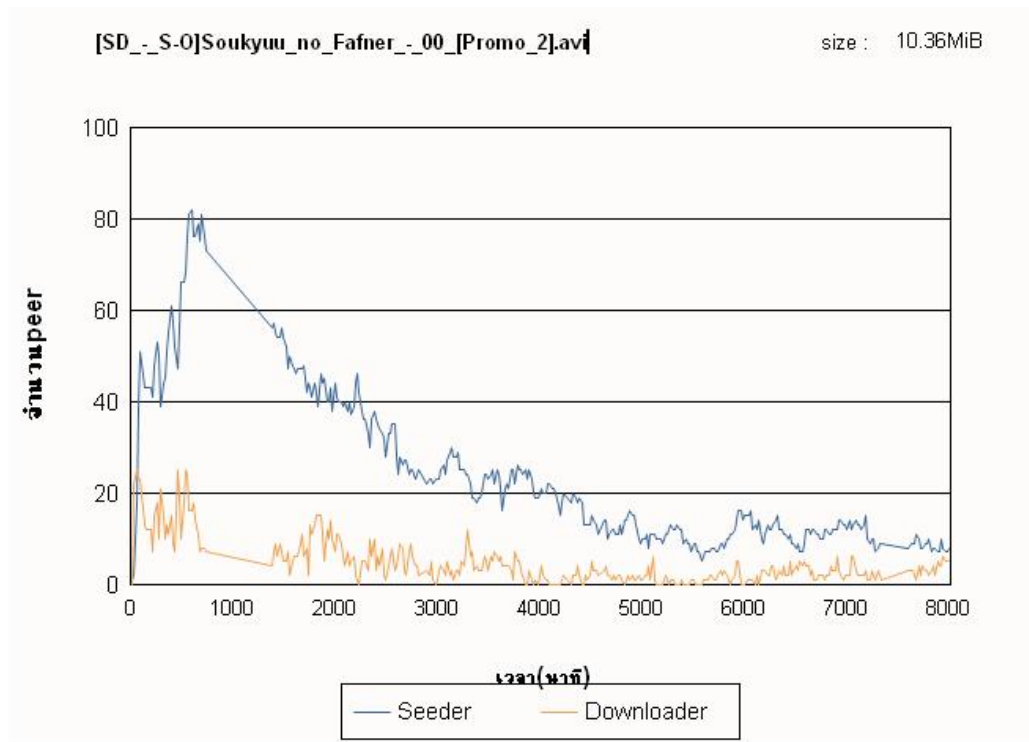
ภาพผนวกที่ ก.12 ตัวอย่างเครือข่ายBittorrentที่12



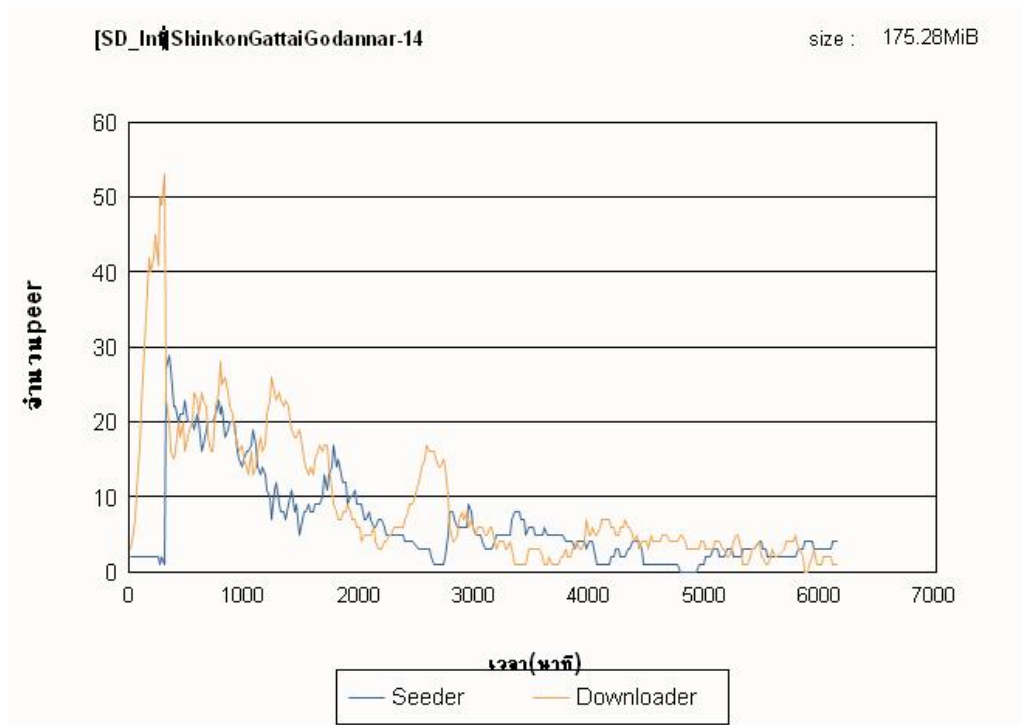
ภาพผนวกที่ ก.13 ตัวอย่างเครือข่ายBittorrentที่13



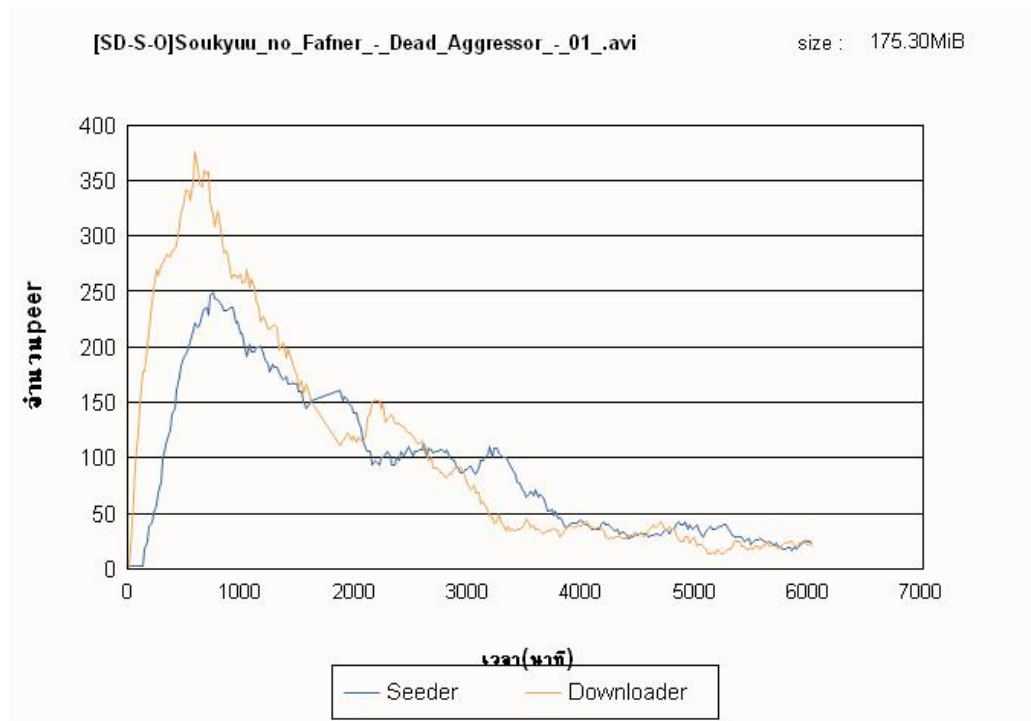
ภาพผนวกที่ ก.14 ตัวอย่างเครือข่ายBittorrentที่14



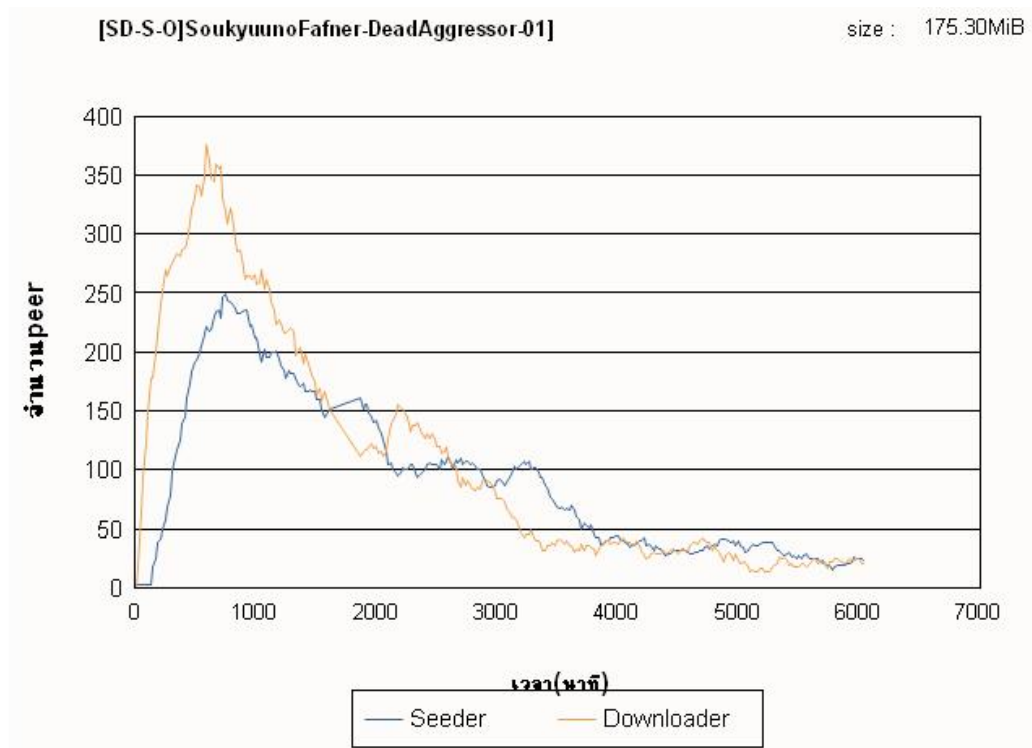
ภาพผนวกที่ ก.15 ตัวอย่างเครือข่ายBittorrentที่15



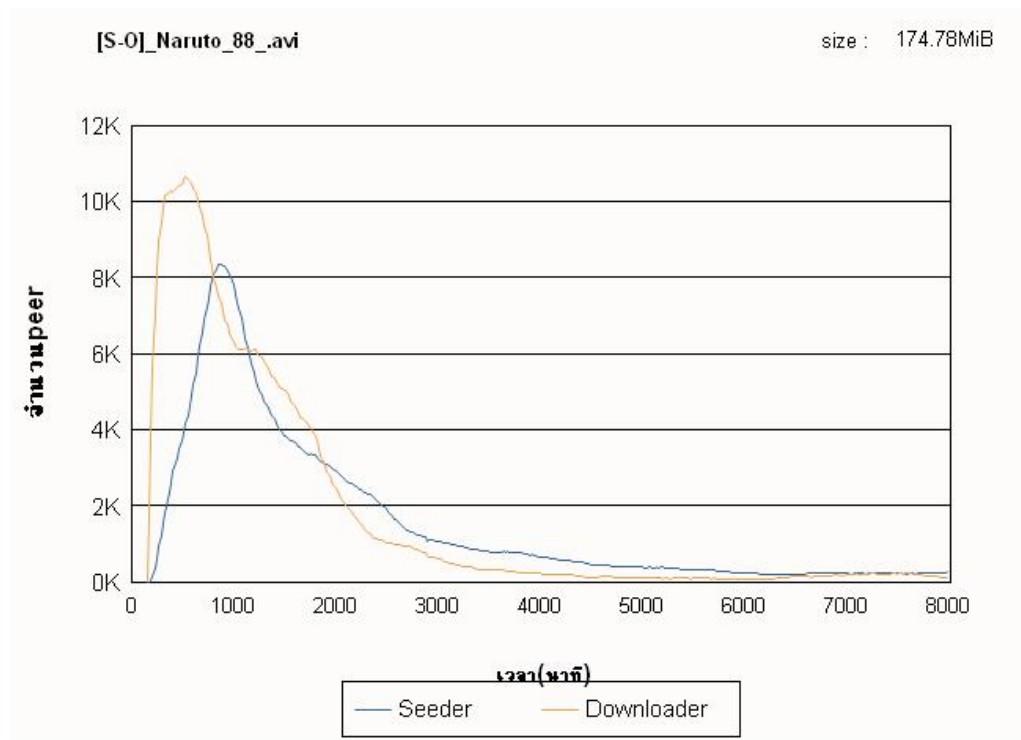
ภาพผนวกที่ ก.16 ตัวอย่างเครือข่ายBittorrentที่16



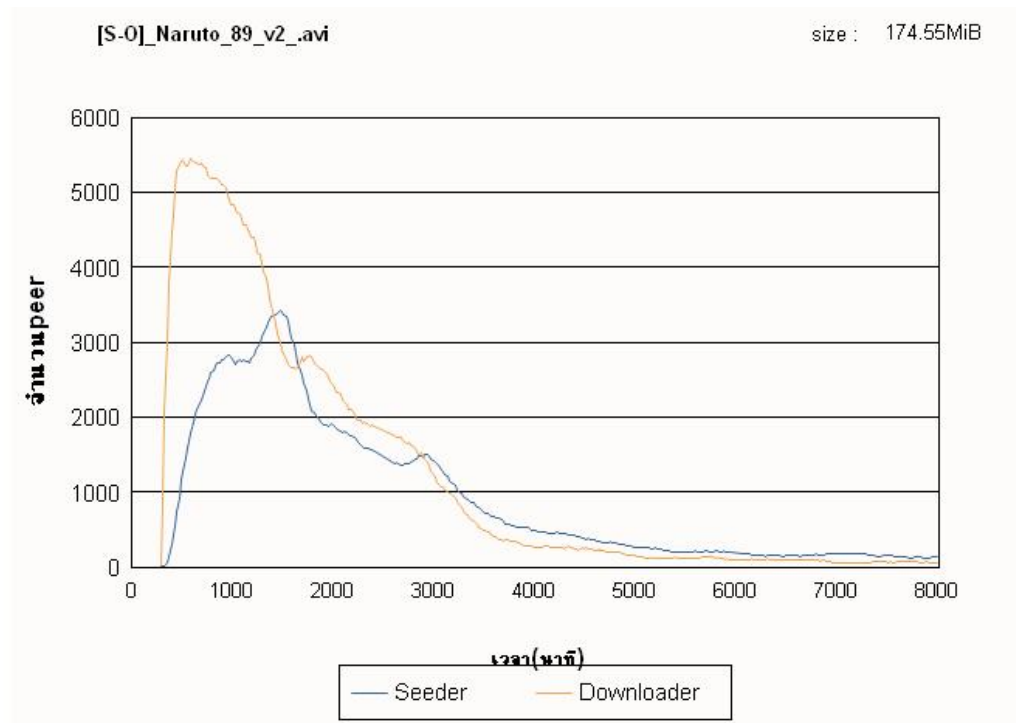
ภาพผนวกที่ ก.17 ตัวอย่างเครือข่ายBittorrentที่17



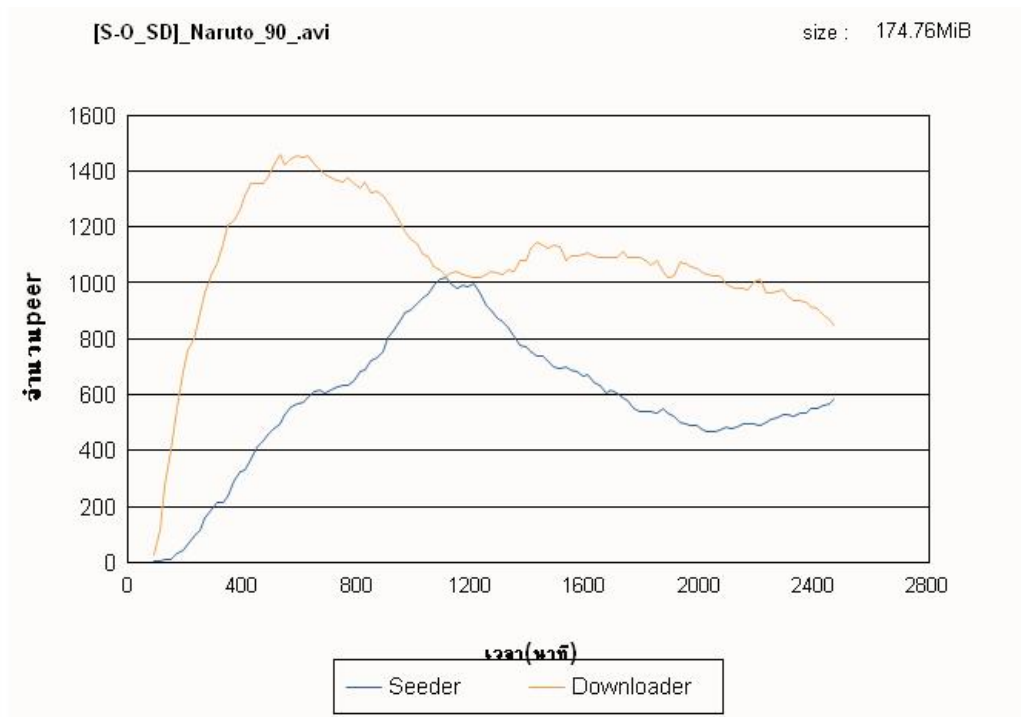
ภาพผนวกที่ ก.18 ตัวอย่างเครือข่ายBittorrentที่18



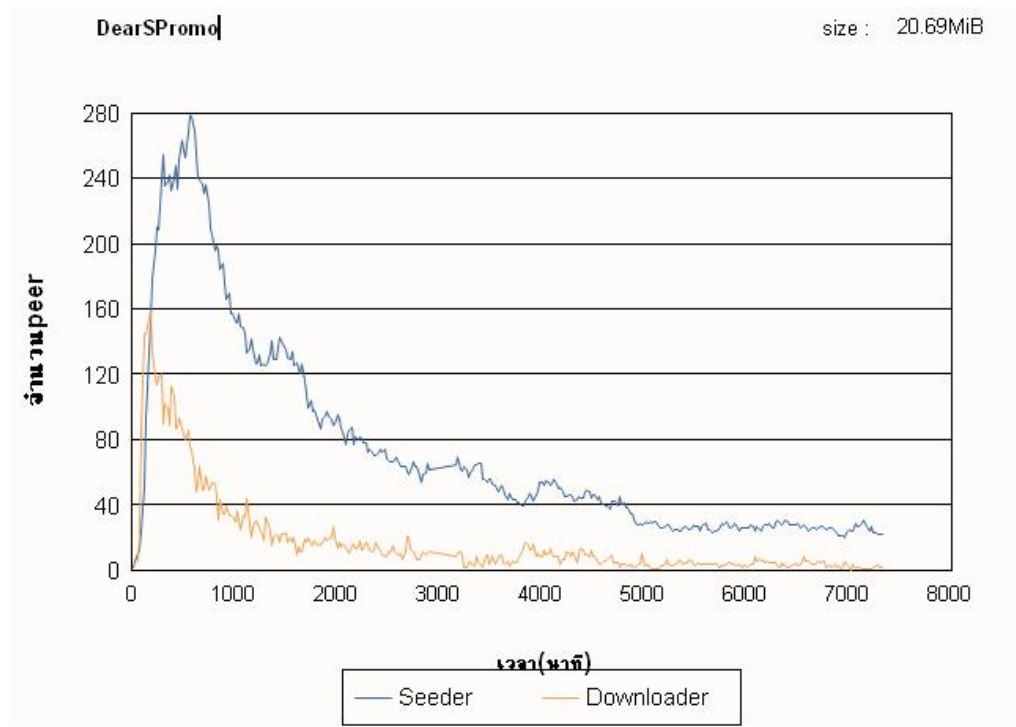
ภาพผนวกที่ ก.19 ตัวอย่างเครือข่ายBittorrentที่19



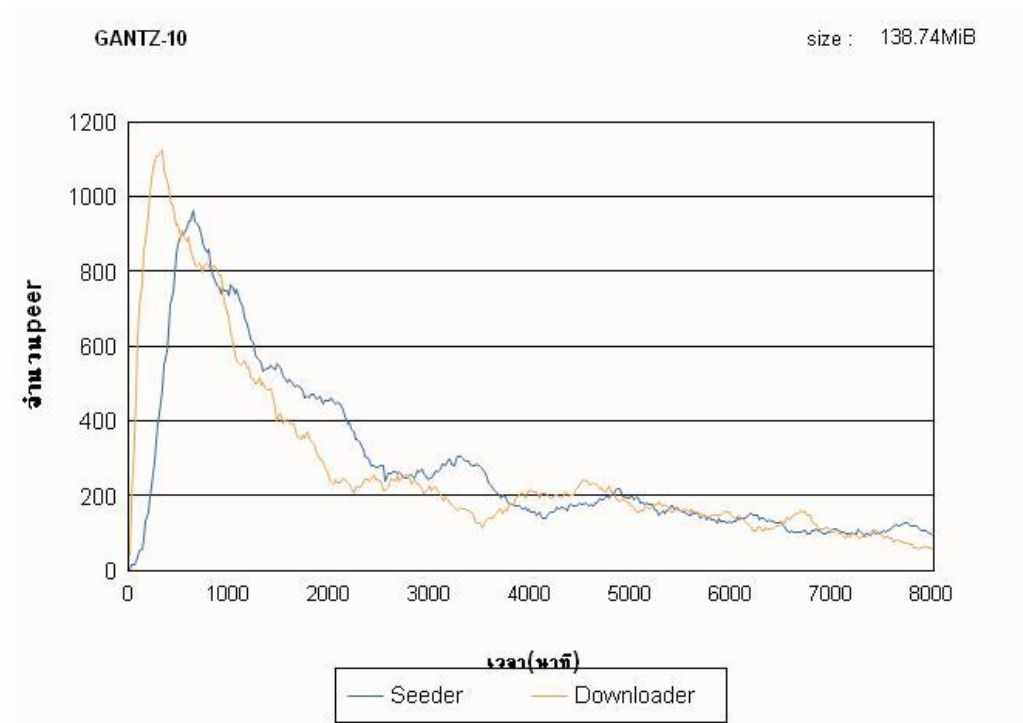
ภาพผนวกที่ ก.20 ตัวอย่างเครือข่ายBittorrentที่20



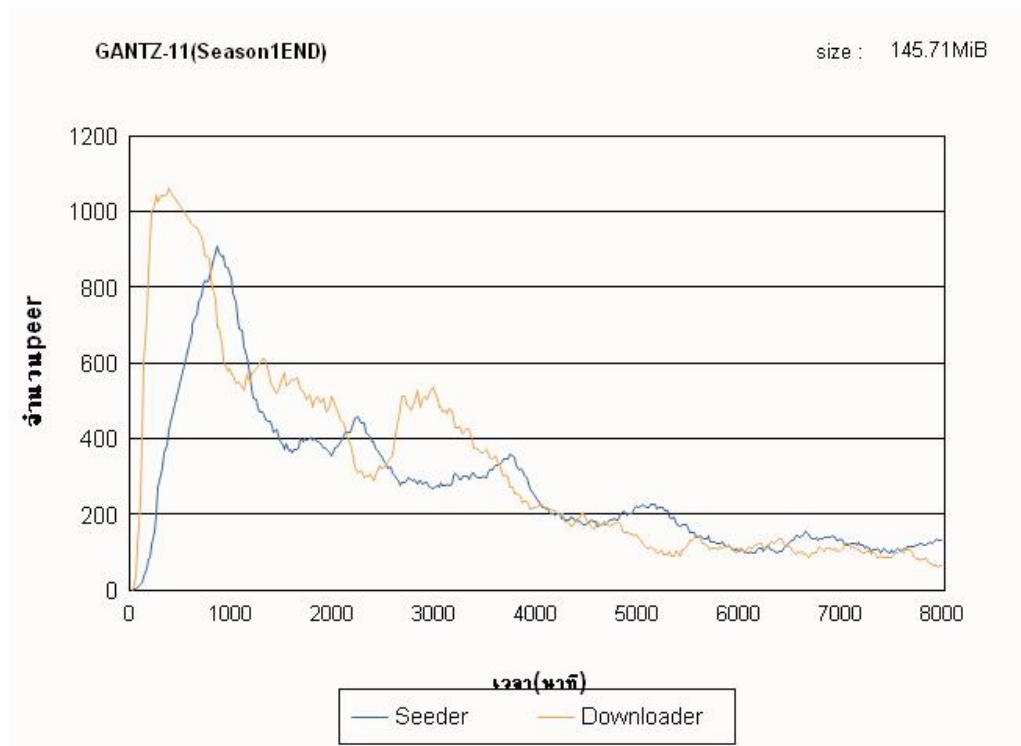
ภาพผนวกที่ ก.21 ตัวอย่างเครือข่ายBittorrentที่21



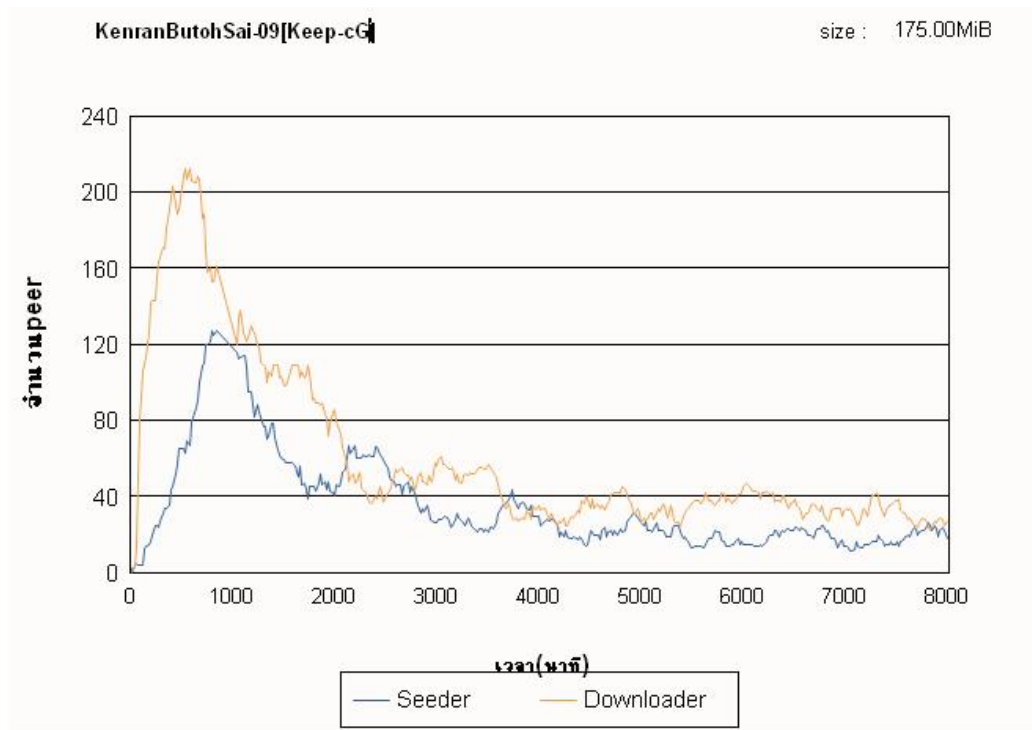
ภาพผนวกที่ ก.22 ตัวอย่างเครือข่ายBittorrentที่22



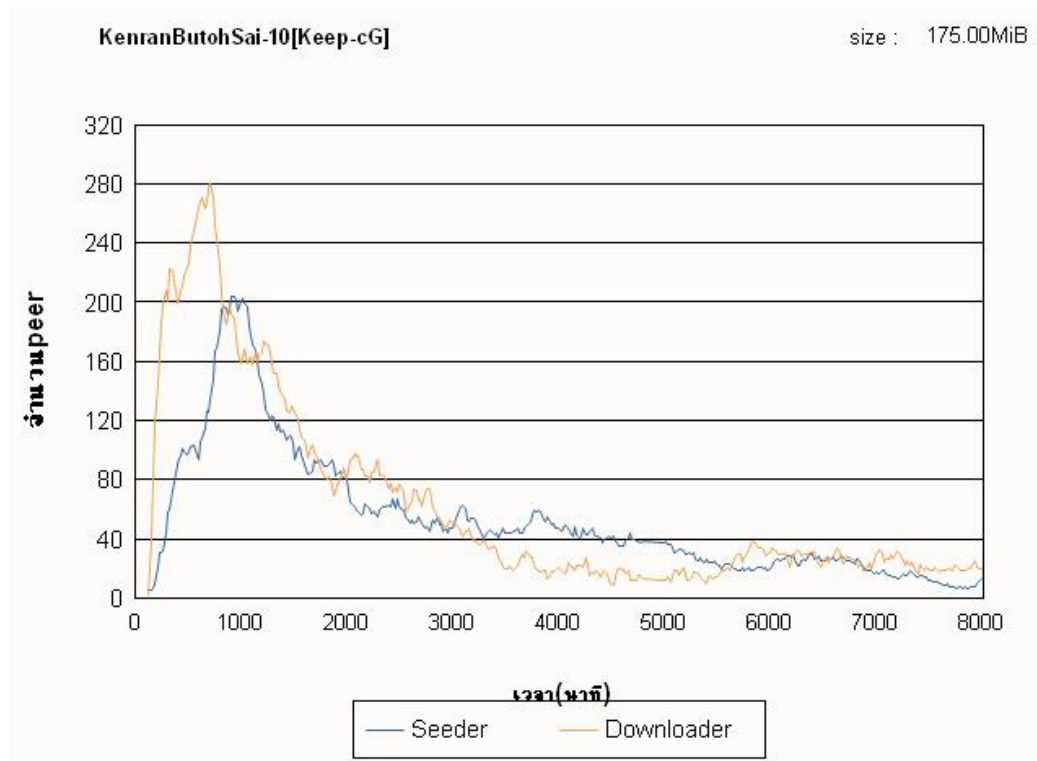
ภาพผนวกที่ ก.23 ตัวอย่างเครือข่ายBittorrentที่23



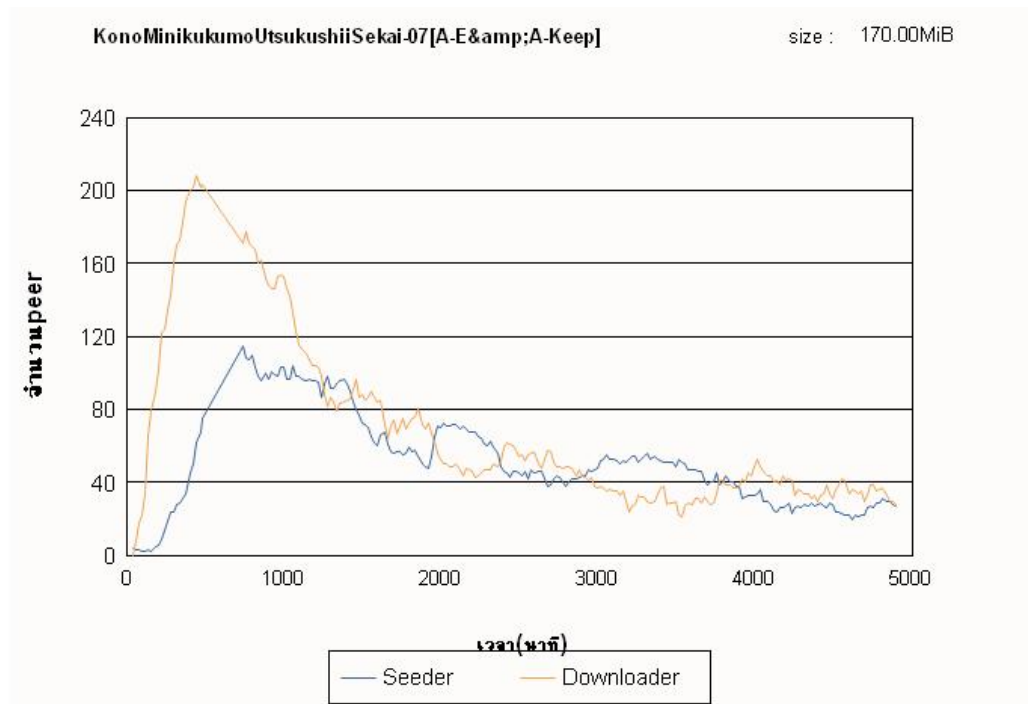
ภาพผนวกที่ ก.24 ตัวอย่างเครือข่ายBittorrentที่24



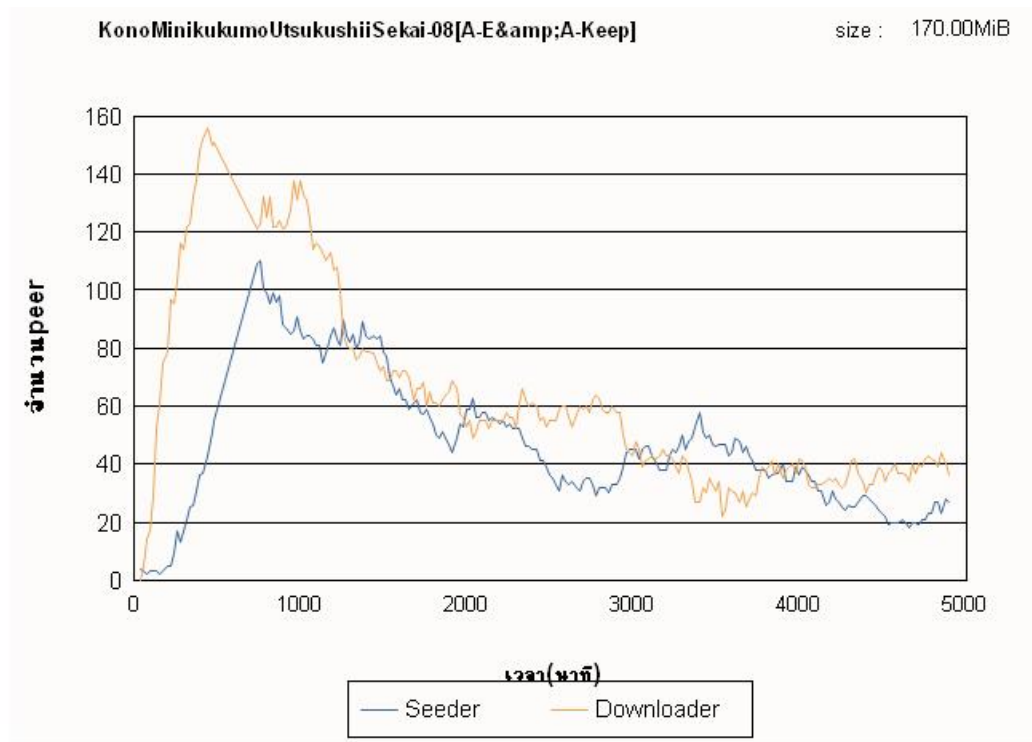
ภาพผนวกที่ ก.25 ตัวอย่างเครือข่ายBittorrentที่25



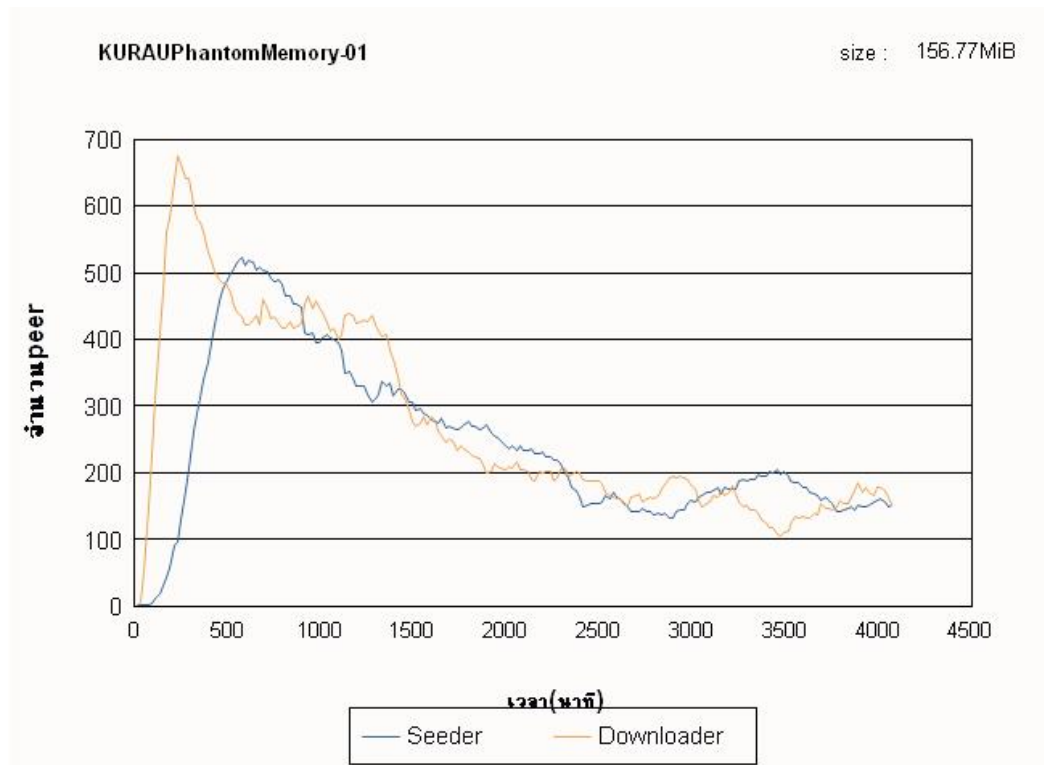
ภาพผนวกที่ ก.26 ตัวอย่างเครือข่ายBittorrentที่26



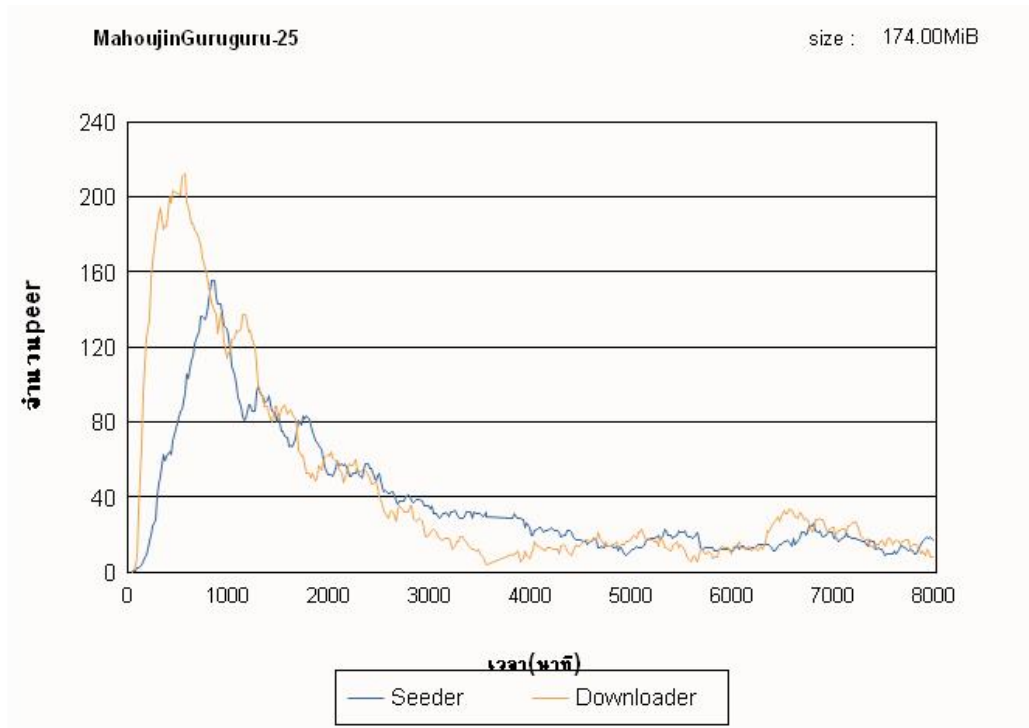
ภาพผนวกที่ ก.27 ตัวอย่างเครือข่ายBittorrentที่27



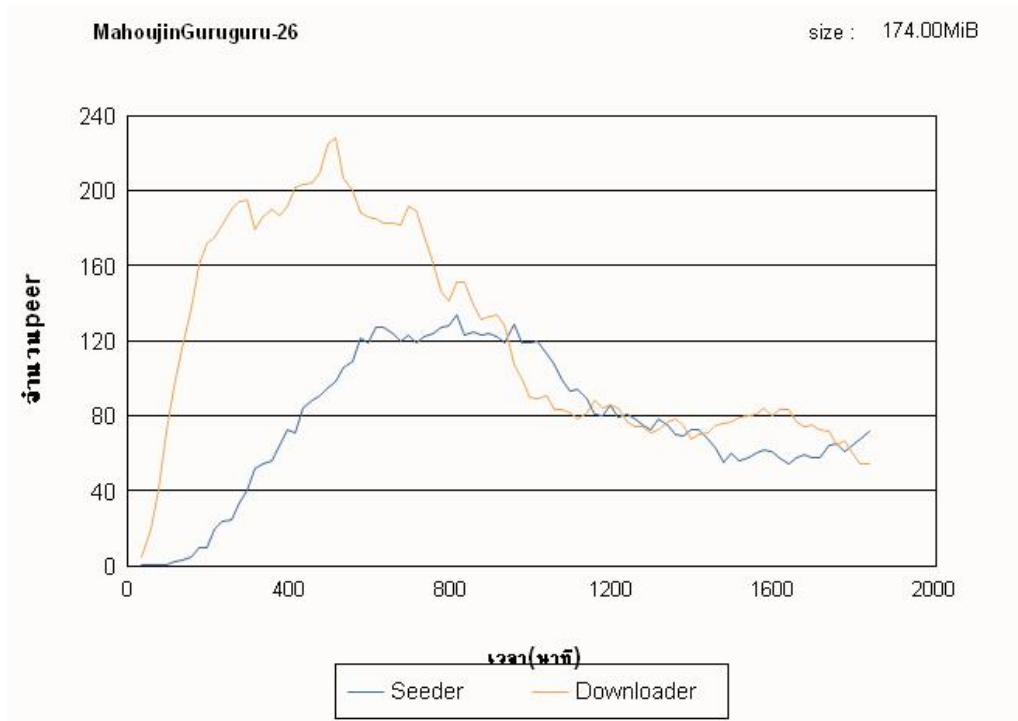
ภาพผนวกที่ ก.28 ตัวอย่างเครือข่ายBittorrentที่28



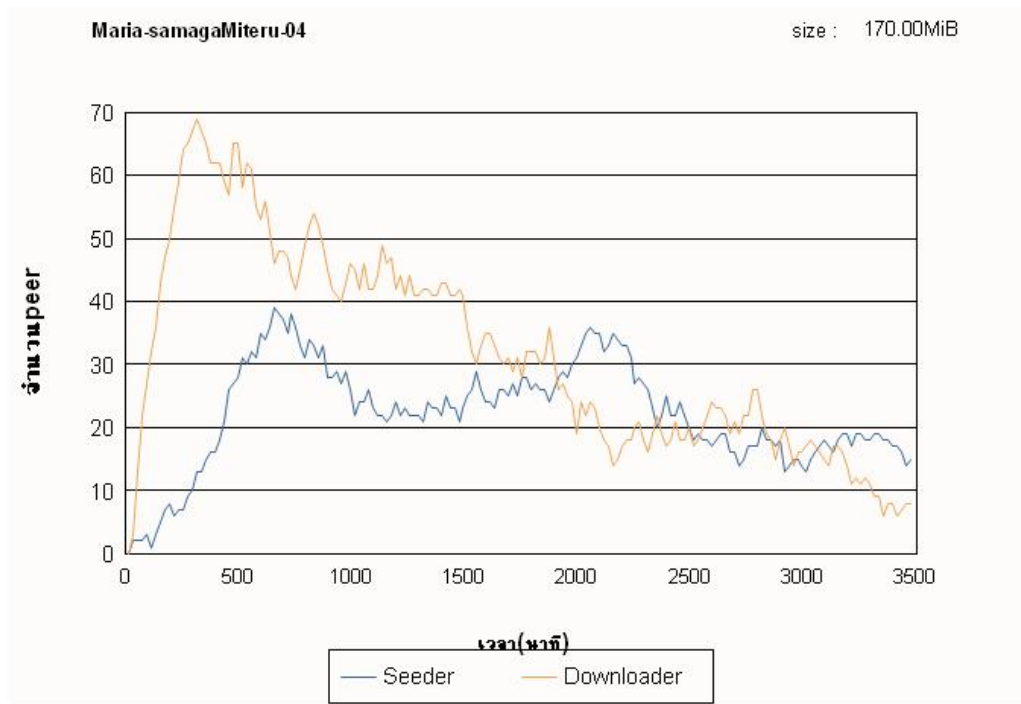
ภาพผนวกที่ ก.29 ตัวอย่างเครือข่ายBittorrentที่29



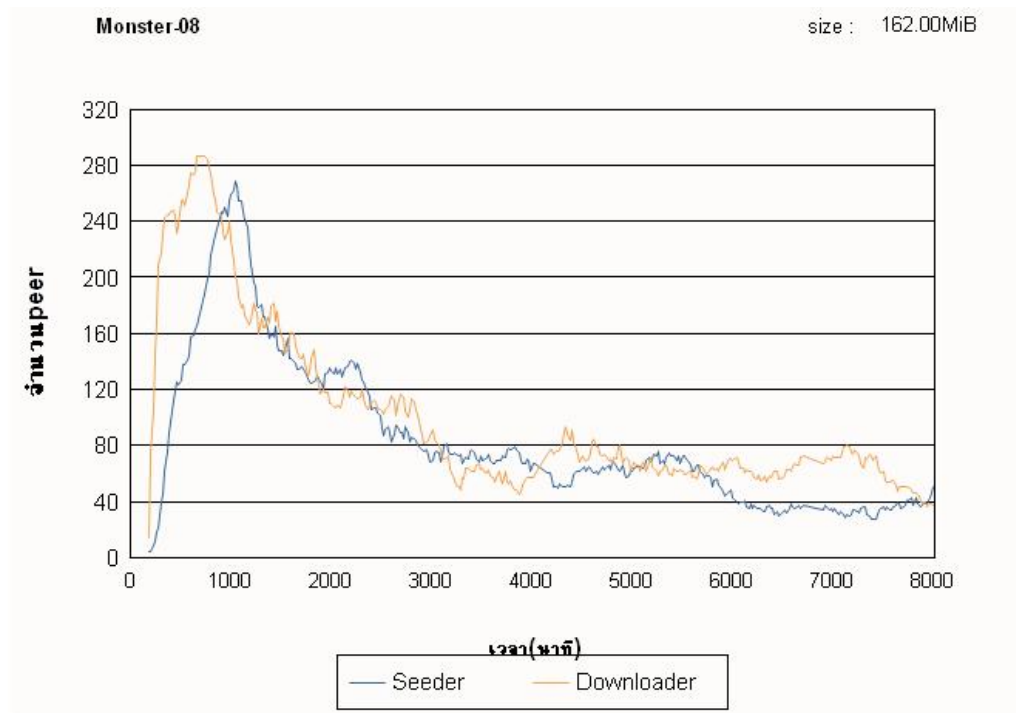
ภาพผนวกที่ ก.30 ตัวอย่างเครือข่ายBittorrentที่30



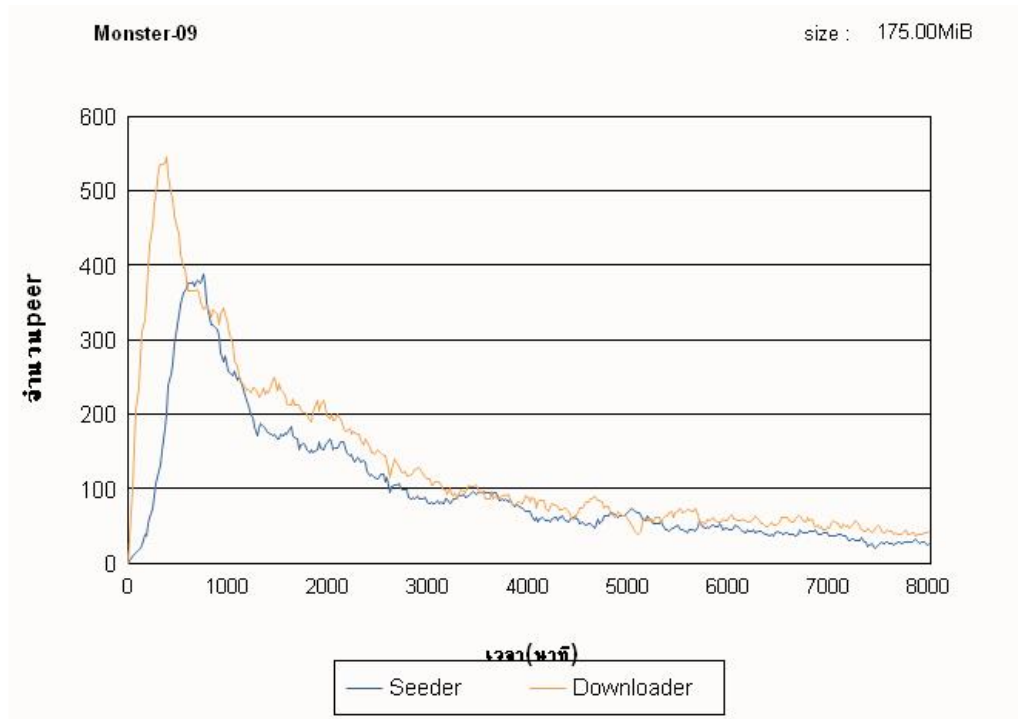
ภาพผนวกที่ ก.31 ตัวอย่างเครือข่ายBittorrentที่31



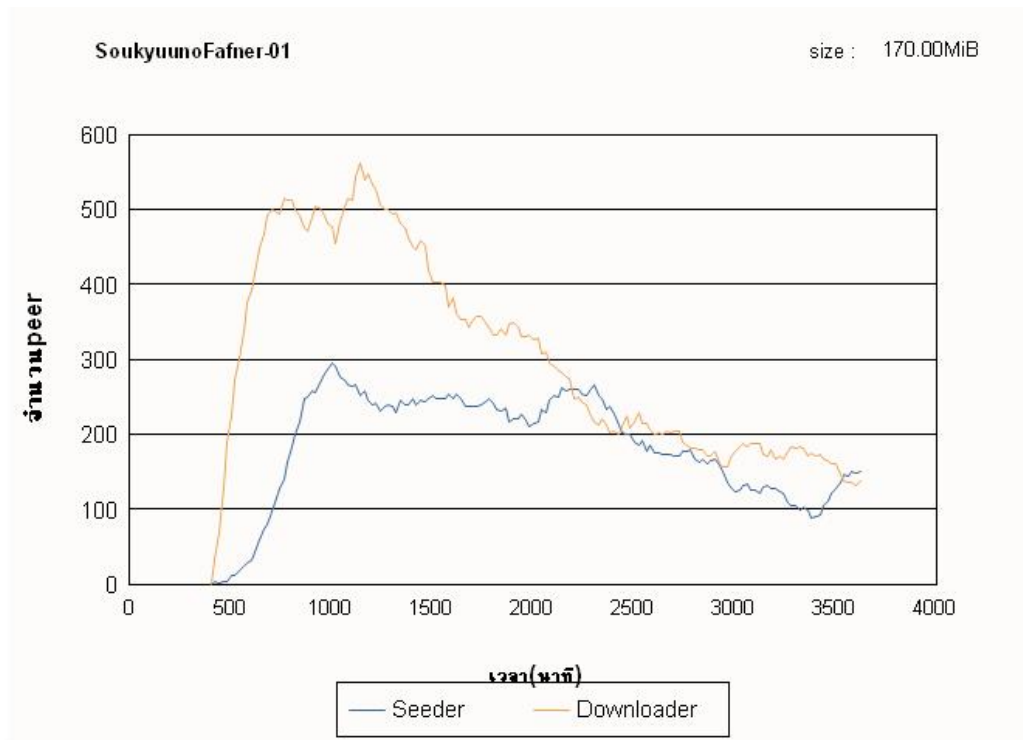
ภาพผนวกที่ ก.32 ตัวอย่างเครือข่ายBittorrentที่32



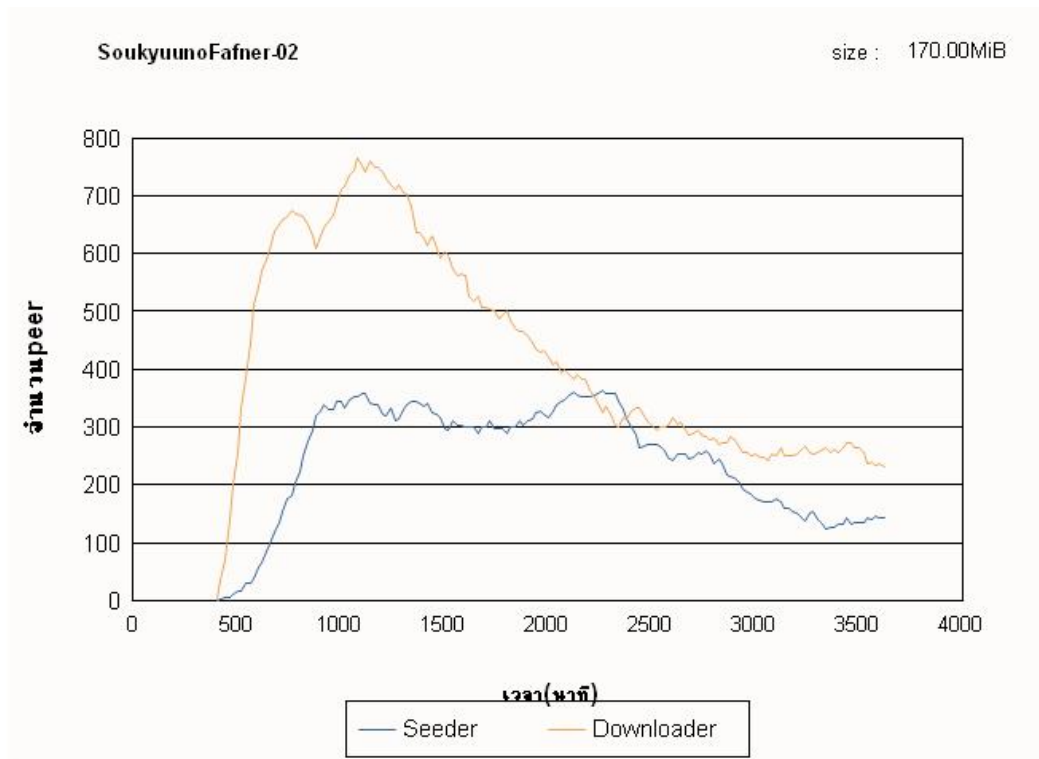
ภาพผนวกที่ ก.33 ตัวอย่างเครือข่ายBittorrentที่33



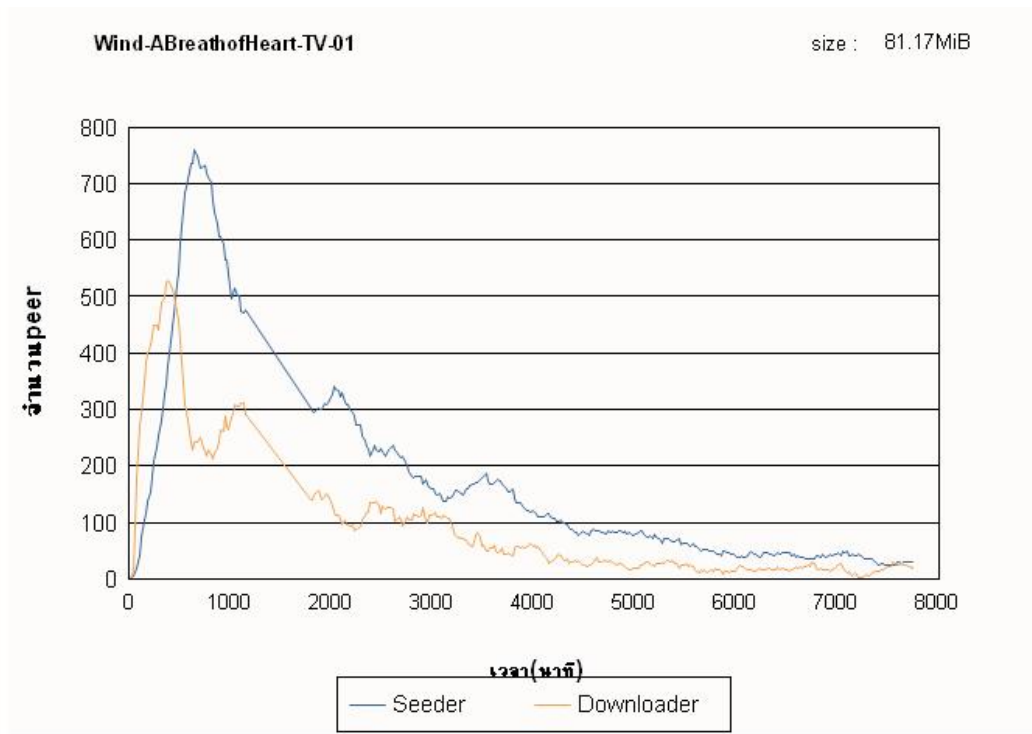
ภาพผนวกที่ ก.34 ตัวอย่างเครือข่ายBittorrentที่34



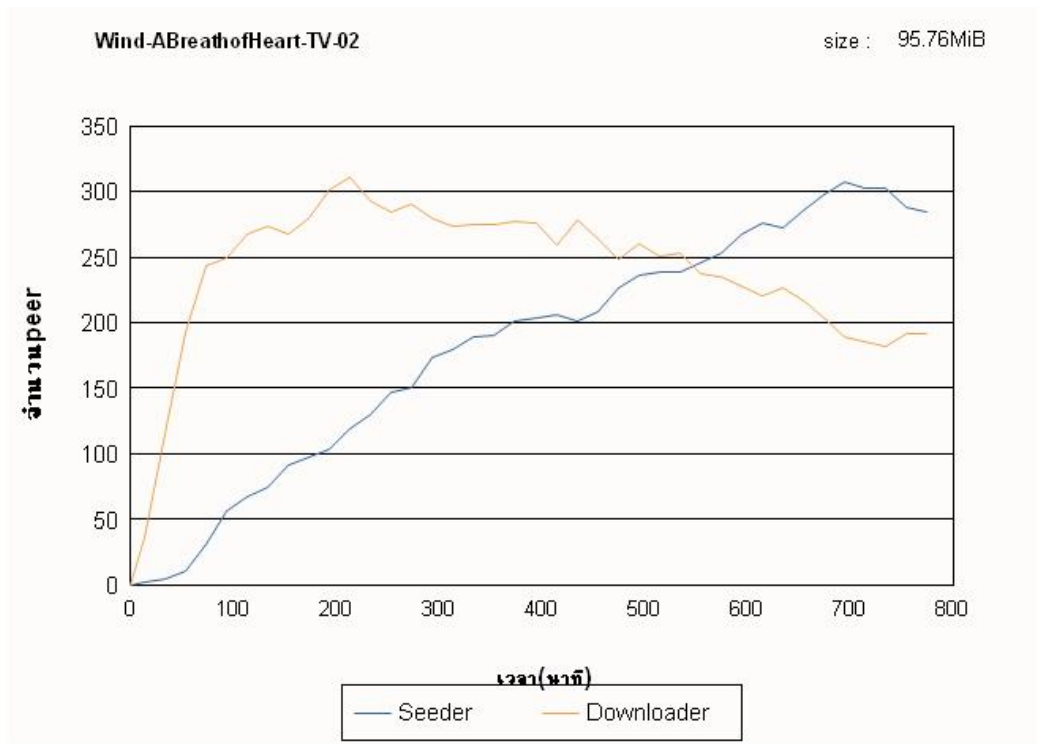
ภาพผนวกที่ ก.35 ตัวอย่างเครือข่ายBittorrentที่35



ภาพผนวกที่ ก.36 ตัวอย่างเครือข่ายBittorrentที่36



ภาพผนวกที่ ก.37 ตัวอย่างเครือข่ายBittorrentที่37



ภาพผนวกที่ ก.38 ตัวอย่างเครือข่ายBittorrentที่38

ประวัติการศึกษา และการทำงาน

ชื่อ-นามสกุล	นายศานติ ชินการณ์
วัน เดือน ปี ที่เกิด	วันที่ 19 พฤศจิกายน 2522
สถานที่เกิด	ภูเก็ต
ประวัติการศึกษา	ระดับปริญญาตรี คณะวิศวกรรมศาสตร์ วิชาเอกวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์
ตำแหน่งหน้าที่การงานปัจจุบัน	นักพัฒนาโปรแกรม
สถานที่ทำงานปัจจุบัน	Thai Smartcard ชั้น 27 สีลมคอมเพล็กซ์