การกระจายไฟล์ข้อมูลสำหรับการเรียนการสอนด้วยสื่ออิเล็กทรอนิกส์

ภายในเครือข่ายคอมพิวเตอร์แบบเพียร์ทูเพียร์

File Distribution Approach for E-Learning Peer to Peer Networks

กิติศักดิ์ วัฒนกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา

วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Computer Engineering

Prince of Songkla University

2553

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์	การกระจายไฟล์ข้อมูลสำหรับการเรียนการสอนด้วยสื่ออิเล็กทรอนิกส์ภายในเครือข่ายคอมพิวเตอร์แบบเพียร์ทูเพียร์
ผู้เขียน	นายกิติศักดิ์ วัฒนกุล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	**คณะกรรมการสอบ**
....	...................................ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุนทร วิทูสุรพจน์)	(ผู้ช่วยศาสตราจารย์ ดร.พิชญา ตัณฑัยย์)
	....กรรมการ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	(ผู้ช่วยศาสตราจารย์ ดร.สุนทร วิทูสุรพจน์)
....	....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต)	(ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต)
	....กรรมการ
	(ผู้ช่วยศาสตราจารย์ ดร.ศุภเชษฐ์ อินทร์เนตร)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

....
(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ การกระจายไฟล์ข้อมูลสำหรับการเรียนการสอนด้วยสื่ออิเล็กทรอนิกส์ ภายในเครือข่ายคอมพิวเตอร์แบบเพียร์ทูเพียร์

ผู้เขียน นายกิติศักดิ์ วัฒนกุล

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ปีการศึกษา 2552

บทคัดย่อ

เทคนิคการกระจายไฟล์แบบเพียร์ทูเพียร์ (Peer-to-Peer) นิยมใช้ในการดาวน์โหลดไฟล์ข้อมูลระหว่างเครื่องคอมพิวเตอร์เพื่อให้เสร็จสิ้นได้อย่างรวดเร็ว เนื่องจากมีการนำเครื่องคอมพิวเตอร์อื่นๆ อีกเป็นจำนวนมากเข้ามาร่วมเพื่อช่วยการส่งต่อข้อมูลด้วย จากการศึกษาประสิทธิภาพของโพรโทคอลบิททอเรนท์ ซึ่งทำงานอยู่บนพื้นฐานของกลไกทำงานแบบเพียร์ทูเพียร์ในวิทยานิพนธ์นี้ พบว่าจะสามารถช่วยลดระยะเวลาในการดาวน์โหลดลงได้อย่างมาก โดยเฉพาะอย่างยิ่งในสถานการณ์ของระบบอีเลิร์นนิ่ง เมื่อร้องขอไฟล์เดียวกันที่มีขนาดใหญ่จากเครื่องผู้ใช้จำนวนมากๆ ในระยะเวลาที่ใกล้เคียงกัน ดังนั้น จึงสามารถช่วยแก้ไขปัญหาข้อด้อยของระบบอีเลิร์นนิ่งจากข้อจำกัดจากสถาปัตยกรรมไคลเอนต์-เซิร์ฟเวอร์ได้เป็นอย่างดี ผลจากการศึกษานี้ได้นำไปใช้การปรับปรุงกลไกทำงานของโพรโทคอลบิททอเรนท์ภายในสภาวะแวดล้อมเครือข่ายท้องถิ่นตามมาตรฐาน UPnP ซึ่งสามารถทำให้เครื่องคอมพิวเตอร์เชื่อมต่อเข้าด้วยกันเป็นเครือข่ายได้โดยอัตโนมัติโดยไม่จำเป็นต้องมีการติดตั้งค่าใดๆ ล่วงหน้า ดังที่พบในห้องเรียนอิเล็กทรอนิกส์สมัยใหม่ทั่วไป ในที่นี้ได้เสนอแนะให้ลดความคับคั่งในการร้องขอข้อมูลจากเครื่องคอมพิวเตอร์อื่นๆ มายังเครื่องแทรกเกอร์เพียงเครื่องเดียว เพื่อรับทราบข้อมูลไอพีแอดเดรสของเครื่องที่เกี่ยวข้องกับกิจกรรมดาวน์โหลดซึ่งจัดเก็บอยู่ในไฟล์ทอเรนท์ โดย 1) ให้สามารถกระจายไปยังเครื่องอื่นๆ ได้ผ่านทางการเรียกใช้ซอฟต์แวร์เชิงบริการที่พัฒนาเพิ่มเติมขึ้น และ 2) ให้สามารถแจ้งเตือนการปรับปรุงข้อมูลไฟล์ทอเรนท์ และให้ข้อมูลโดยอัตโนมัติแทนการร้องขอและเฝ้าตรวจด้วยตนเองผ่านทางการใช้กลไกของโพรโทคอลแจ้งเตือนเหตุการณ์ GENA ของเครือข่าย UPnP ผลการทดสอบของระบบต้นแบบทำงานในสภาพแวดล้อมจริงร่วมกับการใช้ซอฟต์แวร์อีเลิร์นนิ่งมูเดิลได้รับผลการปรับปรุงด้านเวลาในการกระจายไฟล์ที่รวดเร็วมากขึ้น เป็นไปตามที่คาดหวังไว้

คำสำคัญ: อีเลิร์นนิ่ง, มูเดิล, เพียร์ทูเพียร์, โพรโทคอลบิททอเรนท์

Thesis Title File Distribution Approach for E-Learning Peer to Peer Networks

Author Mr. Kitisak Watthanakul

Major Program Computer Engineering

Academic Year 2009

ABSTRACT

A number of Peer-to-Peer (P2P) techniques have been favorably used for fast file transfers, due to the support of other computers involved in the activity, rather than the source and destination computers as found in the normal technique. Based on the study of file transfers using BitTorrent protocol, it is found that the download time can be finished in the shorter time, especially when there are a large number of requests from users attempting to access the same resource file on the server at nearly the same time. So, the P2P-based file transfer has shown a great potential for mitigating the inherent problem found in typical client/server architecture and, hence, should be recommended to any E-learning system. As a consequence, we study further the mechanism of BitTorrent protocol, but in the UPnP network environment, where computers can be automatically formed as a network without any setting in advance and often found in the modern e-classroom. Here, we study the potential congestion problem that may occur at the tracker when a number of peers request for receiving the torrent file, which is kept all IP addresses of peers involved with the file-transfer activity and propose to handle this problem by a) distributing the torrent file to involved peers by issuing a newly proposed service based software, and b) notifying the update of the torrent file to involved peers in the automatic manner (considered as a form of push service) using UPnP's GENA eventing protocol. A first prototype has been developed for working with the Moodle e-learning system and found to achieve the improvement of time duration for file transfers as planned.

Keywords: E-Learning, Moodle, P2P Protocol, BitTorrent

สารบัญ

	หน้า
สารบัญ	(6)
รายการตาราง	(9)
รายการภาพประกอบ	(10)
สัญลักษณ์คำย่อและตัวย่อ	(14)
ศัพทานุกรม	(15)
บทที่	
1. บทนำ	1
1.1 ความสำคัญและที่มาของวิทยานิพนธ์	1
1.2 วัตถุประสงค์ของวิทยานิพนธ์	2
1.3 ขอบเขตของวิทยานิพนธ์	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
2. วิธีการวิจัย	4
2.1 ปัญหาการกระจายไฟล์ของอีเลิร์นนิ่งมูเดิล	4
2.1.1 ที่มาของปัญหา	4
2.1.2 ทบทวนวรรณกรรมที่เกี่ยวข้อง	5
2.1.3 แนวคิดในการแก้ปัญหา	5
2.1.4 กลไกการทำงานของบิททอเรนท์	7
2.2 ปัญหาการกระจายไฟล์อีเลิร์นนิ่งในเครือข่ายท้องถิ่นของชั้นเรียน	9
2.2.1 ที่มาของปัญหา	9
2.2.2 ทบทวนวรรณกรรมที่เกี่ยวข้อง	9
2.2.3 แนวคิดในการแก้ปัญหา	10
2.2.4 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง	10
2.2.4.1 โพรโทคอลภายในเครือข่าย UPnP	10
2.2.4.2 กรอบงานมาตรฐาน OSGi	11

สารบัญ (ต่อ)

หน้า

2.2.4.3 เทคโนโลยี Mashups.. 12

2.3 ปัญหาการใช้งานบิททอเรนท์ในเครือข่าย UPnP.. 13

2.3.1 ที่มาของปัญหา.. 13

2.3.2 ทบทวนวรรณกรรมที่เกี่ยวข้อง.. 14

2.3.3 แนวคิดในการแก้ปัญหา... 14

3. การออกแบบและพัฒนาระบบ.. 16

3.1 กลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่ง มูเดิล.. 16

3.1.1 สถาปัตยกรรมระบบ... 16

3.1.2 การเพิ่มโมดูลกระจายไฟล์ข้อมูล P2P ในมูเดิล.............................. 17

3.2 กลไกบริการ OSGi เพื่อกระจายไฟล์แบบบิททอเรนท์สำหรับอีเลิร์นนิ่งใน เครือข่าย UPnP.. 18

3.2.1 สถาปัตยกรรมระบบ... 18

3.2.2 ระบบบิททอเรนท์ภายในเครือข่าย UPnP.................................... 19

3.3 กลไกบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP................... 21

3.3.1 การแจ้งเตือนบริการและระบบบิททอเรนท์ภายในเครือข่าย UPnP.. 21

4. ผลการทดลอง.. 23

4.1 ผลการทดลอง กลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์ อีเลิร์นนิ่งมูเดิล.. 23

4.1.1 สภาพแวดล้อมในการทดลอง... 23

4.1.2 ผลการทดลอง... 23

4.1.3 สรุปผลการทดลอง... 35

สารบัญ (ต่อ)

หน้า

4.2 ผลการทดลอง กลไกบริการ OSGi เพื่อกระจายไฟล์แบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP........ 36

4.2.1 สภาพแวดล้อมในการทดลอง........ 36

4.2.2 ผลการทดลอง........ 37

4.2.3 สรุปผลการทดลอง........ 38

4.3 ผลการทดลอง กลไกบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP........ 39

4.3.1 สภาพแวดล้อมในการทดลอง........ 39

4.3.2 ผลการทดลอง........ 40

4.3.3 สรุปผลการทดลอง........ 46

5. สรุปผลการวิจัยและข้อเสนอแนะ........ 47

5.1 สิ่งที่ได้นำเสนอในวิทยานิพนธ์........ 47

5.2 ข้อเสนอแนะ........ 48

เอกสารอ้างอิง........ 49

ภาคผนวก........ 52

ภาคผนวก ก: ผลงานที่ได้รับการตีพิมพ์เผยแพร่........ 53

ประวัติผู้เขียน........ 71

รายการตาราง

ตาราง		หน้า
4-1	ตารางเปรียบเทียบประสิทธิภาพด้านเวลา ไฟล์ข้อมูลขนาด 50 MByte…………….….	27
4-2	ตารางเปรียบเทียบประสิทธิภาพด้านเวลา ไฟล์ข้อมูลขนาด 600 MByte………………	31
4-3	ตารางเปรียบเทียบประสิทธิภาพด้านเวลา ไฟล์ข้อมูลขนาด 4330 MByte…………..…	35
4-4	ตารางเปรียบเทียบการสุ่มเวลา 3 ช่วง ของการเปลี่ยนแปลงบริการแบบ Push Service และ Pull Service ……………………………………………………..…….….	40

รายการภาพประกอบ

ภาพประกอบ		หน้า
2-1	การติดต่อระหว่างเพียร์โดยใช้โพรโทคอลบิททอเรนท์..................................	7
2-2	ลำดับขั้นตอนกระจายไฟล์ข้อมูลด้วยกลไกบิททอเรนท์..................................	8
2-3	แนวทางการนำเอเจนท์มาใช้เพื่อลดภาระงานให้กับเซิร์ฟเวอร์อีเลิร์นนิ่ง.............	9
2-4	แผนภาพชุดโพรโทคอลของเครือข่าย UPnP...	11
2-5	กรอบงานการพัฒนาแอพพลิเคชันเชิงบริการตามมาตรฐาน OSGi.....................	12
2-6	วัฏจักรการทำงานของบันเดิลในกรอบงาน OSGi..	12
2-7	สถาปัตยกรรม Mashups..	13
2-8	ลำดับขั้นตอนการทำงานของ Pull Service...	14
2-9	ลำดับขั้นตอนการทำงานของ Push Service...	15
3-1	สถาปัตยกรรมของระบบต้นแบบการกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์...........	17
3-2	รายการที่ปรากฏชื่อโมดูล กระจายไฟล์ข้อมูล P2P..	18
3-3	สถาปัตยกรรมกระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP..	18
3-4	การเพิ่มบริการบิททอเรนท์ผ่านบันเดิลบนมาตรฐาน OSGi..............................	19
3-5	แผนภาพลำดับเวลาการค้นหาบริการเพื่อกระจายไฟล์ข้อมูลด้วยบิททอเรนท์ในเครือข่าย UPnP..	20
3-6	แผนภาพลำดับเวลาการลงทะเบียนบริการ, การแจ้งเตือนเหตุการณ์และกระจายไฟล์ข้อมูลบิททอเรนท์ในเครือข่าย UPnP..	22
4-1	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 1 เครื่อง.........................	24
4-2	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 2 เครื่อง.........................	24
4-3	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 4 เครื่อง.........................	25

รายการภาพประกอบ (ต่อ)

ภาพประกอบ		หน้า
4-4	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 16 เครื่อง......................	25
4-5	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 40 เครื่อง......................	26
4-6	การเปรียบเทียบเวลาในการดาวน์โหลดข้อมูลระหว่างระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte..	26
4-7	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 1 เครื่อง......................	28
4-8	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 2 เครื่อง......................	28
4-9	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 4 เครื่อง......................	29
4-10	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 16 เครื่อง	29
4-11	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 40 เครื่อง	30
4-12	การเปรียบเทียบเวลาในการดาวน์โหลดข้อมูลระหว่างระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte	30
4-13	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 1 เครื่อง	32
4-14	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 2 เครื่อง	32
4-15	การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 4 เครื่อง	33

รายการภาพประกอบ (ต่อ)

ภาพประกอบ | หน้า

4-16 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 16 เครื่อง 33

4-17 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 40 เครื่อง 34

4-18 การเปรียบเทียบเวลาในการดาวน์โหลดข้อมูลระหว่างระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte... 34

4-19 ระบบทดลองของเครือข่าย UPnP เพื่อเปรียบเทียบกลไกกระจายไฟล์ข้อมูลด้วย โพรโทคอล HTTP และบิททอเรนท์.. 36

4-20 กราฟเปรียบเทียบเวลาดาวน์โหลดไฟล์ข้อมูล แบบพื้นฐาน (HTTP) กับแบบบิททอเรนท์ (UPnP-BitTorrent) 37

4-21 ตัวอย่างเว็บแอพพลิชันแบบ Mashups ใช้บริการดาวน์โหลดไฟล์ด้วยบิททอเรนท์ ด้วยเครื่องมือ Open Mashups.. 38

4-22 ตัวอย่างแสดงการใช้เว็บแอพพลิเคชันที่พัฒนาขึ้น.. 38

4-23 ระบบทดลองบิททอเรนท์ภายในเครือข่าย UPnP.. 39

4-24 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการ ดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 30 วินาที, Pull Service: 24 วินาที......... 41

4-25 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการ ดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 55 วินาที, Pull Service: 50 วินาที......... 41

4-26 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการ ดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 87 วินาที, Pull Service: 64 วินาที.......... 42

4-27 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการ ดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 21 วินาที, Pull Service: 30 วินาที.......... 42

4-28 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการ ดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 39 วินาที, Pull Service: 60 วินาที.......... 43

4-29 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการ ดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 95 วินาที, Pull Service: 105 วินาที........ 43

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

4-30 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 11 วินาที, Pull Service: 15 วินาที......... 44

4-31 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 52 วินาที, Pull Service: 31 วินาที.......... 44

4-32 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 97 วินาที, Pull Service: 95 วินาที.......... 45

4-33 เปรียบเทียบเวลาการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการระหว่าง Push Service และ Pull Service.. 46

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของวิทยานิพนธ์

ปัจจุบันแนวโน้มการนำรูปแบบของการเรียนการสอนด้วยสื่ออิเล็กทรอนิกส์ (ซึ่งต่อไปจะขอใช้คำว่า "อีเลิร์นนิ่ง (E-Learning)") เข้ามาใช้งานมากยิ่งขึ้น เนื่องจากความแพร่หลายของเครือข่ายอินเทอร์เน็ต (Internet) ซึ่งเป็นเครือข่ายสาธารณะที่ได้รับความนิยมมากที่สุด สำหรับในประเทศไทย มักนำซอฟต์แวร์ (Software) ประเภทโอเพ่นซอร์ส (Open Source) ทางด้าน Learning Management System (LMS) เช่น โปรแกรมมูเดิล (Moodle) [1] เป็นต้น เข้ามาใช้งานในการสร้าง/จัดเก็บไฟล์ข้อมูลอิเล็กทรอนิกส์สำหรับการเรียนการสอน เนื่องจากซอฟต์แวร์เหล่านี้มีการอินเตอร์เฟส (Interface) กับผู้ใช้ที่ดี จึงง่ายและสะดวกต่อการใช้งาน อย่างไรก็ตาม ข้อจำกัดของระบบอีเลิร์นนิ่งที่สร้างด้วยโปแกรมเหล่านี้ ล้วนเป็นผลมาจากข้อจำกัดในการทำงานแบบสถาปัตยกรรมไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) เอง ตัวอย่างเช่น หากมีไฟล์ข้อมูลขนาดใหญ่และจำนวนผู้เรียกใช้งานในไฟล์เดียวกันนั้นมีจำนวนมาก ก็จะทำให้เครื่องเซิร์ฟเวอร์ (Server) มีประสิทธิภาพที่ลดลงอย่างมีนัยสำคัญ ตามภาระความคับคั่งของการร้องขอไฟล์ที่เพิ่มมากขึ้น

โดยทั่วไป การแก้ไขความคับคั่งแบบรวมศูนย์ ณ เครื่องเซิร์ฟเวอร์ ดังกล่าวข้างต้นนั้น สามารถทำได้โดยการจำกัดจำนวนผู้ใช้ในเวลาหนึ่งๆ ลง เพื่อรักษาให้คุณภาพของการให้บริการอยู่ในระดับที่ยอมรับได้ อย่างไรก็ตาม การศึกษาในงานวิทยานิพนธ์นี้จะได้พิจารณาศึกษาแนวทางการกระจายงานในเครือข่ายแบบเพียร์ทูเพียร์ (Peer-to-Peer) [2] มาใช้ในการแก้ไขปัญหานี้ เนื่องจาก มีข้อดีที่ไม่เพียงแต่จะสามารถลดภาระงานจากเครื่องเซิร์ฟเวอร์ลงได้ แต่จำนวนผู้ใช้ก็มิได้จำกัดลงด้วยเช่นกัน ตามหลักการทำงานของสถาปัตยกรรมแบบเพียร์ทูเพียร์แล้ว ภาระในการกระจายไฟล์ข้อมูลจะไม่กระจุกรวมศูนย์อยู่เฉพาะที่เครื่องเซิร์ฟเวอร์เท่านั้น แต่สามารถจะกระจายไปยังเครื่องคอมพิวเตอร์อื่นๆ ซึ่งได้ติดต่อสื่อสารกันไว้ล่วงหน้า ผ่านทางโพรโทคอล (Protocol) สื่อสารต่างๆ เช่น บิททอเรนท์ (BitTorrent) [3, 4] เป็นต้น ซึ่งปัจจุบันโพรโทคอลนี้ได้รับความนิยมและนำไปใช้งานกันโดยแพร่หลายเพื่อแลกเปลี่ยนไฟล์เอกสารกันเป็นจำนวนมาก

อย่างไรก็ตาม แม้โพรโทคอลบิททอเรนท์จะถูกนำไปใช้อย่างกว้างขวาง แต่พบงานวิจัยเผยแพร่น้อยมากที่เกี่ยวข้องกับการนำไปใช้เพื่อสนับสนุนการกระจายไฟล์ข้อมูลจากระบบอีเลิร์นนิ่ง ดังนั้น งานวิจัยเพื่อวิทยานิพนธ์นี้จะศึกษาประสิทธิภาพของกลไกในการกระจายไฟล์ข้อมูลจากระบบอีเลิร์นนิ่งโดยใช้โพรโทคอลบิททอเรนท์เป็นสำคัญ โดยเฉพาะอย่างยิ่งเมื่อทำงานอยู่ภายในสภาพแวดล้อมของเครือข่ายตามมาตรฐาน Universal Plug and Play (UPnP) [5] ซึ่งอุปกรณ์สื่อสารและคอมพิวเตอร์ต่างๆ สามารถติดต่อสื่อสารเข้าเป็นเครือข่ายได้เองโดยอัตโนมัติ ดังเช่นที่มักจะพบในสถานการณ์ของชั้นเรียนสมัยใหม่แบบอิเล็กทรอนิกส์ทั่วไป

1.2 วัตถุประสงค์ของวิทยานิพนธ์

เพื่อศึกษากลไกทำงานของการลดภาระความคับคั่งในการกระจายไฟล์ข้อมูลจากเว็บไซต์อีเลิร์นนิ่งที่ใช้ลักษณะสถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ ด้วยแนวทางการสื่อสารแบบเพียร์ทูเพียร์

1.3 ขอบเขตของวิทยานิพนธ์

1) ศึกษากลไกการกระจายไฟล์ข้อมูลเครือข่ายคอมพิวเตอร์แบบเพียร์ทูเพียร์โดยใช้โพรโทคอลบิททอเรนท์
2) ทดสอบประสิทธิภาพที่ดีขึ้นทางด้านเวลาในการกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์ด้วยโพรโทคอลบิททอเรนท์ เมื่อเปรียบเทียบกับกลไกกระจายไฟล์ข้อมูลด้วยโพรโทคอล HTTP เช่นปกติทั่วไป
3) ศึกษาข้อจำกัดของกลไกการกระจายไฟล์ข้อมูลด้วยโพรโทคอลบิททอเรนท์ ภายใต้สภาวะแวดล้อมในการทำงานของเครือข่าย UPnP พร้อมเสนอแนวทางในการปรับปรุง

1.4 ขั้นตอนการดำเนินงาน

1) ศึกษาระบบการจัดการไฟล์ข้อมูลภายในฐานข้อมูลของซอฟต์แวร์อีเลิร์นนิ่งที่ใช้สถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ เช่น มูเดิล เป็นต้น

2) ศึกษาเปรียบเทียบประสิทธิภาพของกลไกการกระจายไฟล์ข้อมูลในระบบอีเลิร์นนิ่งระหว่างการทำงานแบบไคลเอนต์-เซิร์ฟเวอร์ กับแบบเพียร์ทูเพียร์ด้วยโพรโทคอลบิททอเรนท์
3) ศึกษาข้อด้อยของกลไกกระจายไฟล์ข้อมูลในระบบอีเลิร์นนิ่งแบบเพียร์ทูเพียร์ด้วยโพรโทคอลบิททอเรนท์ ในเครือข่าย UPnP พร้อมกับการทดสอบในสภาวะแวดล้อมจริง
4) สรุปและวิจารณ์ผล พร้อมเขียนวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) รายงานเชิงประสิทธิภาพด้านเวลาในการกระจายไฟล์ข้อมูลอีเลิร์นนิ่งแบบเพียร์ทูเพียร์ด้วยโพรโทคอลบิททอเรนท์ เมื่อเปรียบเทียบกับกลไกกระจายไฟล์ข้อมูลแบบพื้นฐานด้วยโพรโทคอล HTTP เพื่อลดภาระความคับคั่งแบบคอขวดของเครื่องเซิร์ฟเวอร์อีเลิร์นนิ่งแบบไคลเอนต์-เซิร์ฟเวอร์
2) ซอฟต์แวร์ปลั๊กอิน ที่มีพื้นฐานการทำงานตามกรอบงานเชิงสถาปัตยกรรมบริการสำหรับการกระจายไฟล์ข้อมูลภายในเครือข่ายคอมพิวเตอร์แบบเพียร์ทูเพียร์โดยใช้โพรโทคอลบิททอเรนท์

บทที่ 2

วิธีการวิจัย

ในบทนี้จะเป็นการกล่าวถึงประเด็นปัญหาของงานวิจัยในวิทยานิพนธ์นี้ ซึ่งได้จัดแบ่งเนื้อหาออกเป็น 3 หัวข้อสำคัญ โดยแต่ละหัวข้อจะประกอบไปด้วย 1) ที่มาของปัญหา 2) ทบทวนวรรณกรรมที่เกี่ยวข้อง 3) แนวคิดในการแก้ปัญหา และเทคโนโลยีที่เกี่ยวข้อง ทั้งนี้ได้จัดวางลำดับเริ่มต้นจากปัญหาการกระจายไฟล์ข้อมูลอีเลิร์นนิ่งมูเดิล ตามด้วยปัญหาในการกระจายไฟล์ข้อมูลจากระบบอีเลิร์นนิ่งในเครือข่ายท้องถิ่นของชั้นเรียน และสุดท้ายจึงเป็นปัญหาการใช้งานโพรโทคอลบิททอเรนท์ภายในสภาพแวดล้อมของเครือข่ายมาตรฐาน UPnP

2.1 ปัญหาการกระจายไฟล์ข้อมูลอีเลิร์นนิ่งมูเดิล

2.1.1 ที่มาของปัญหา

การกระจายไฟล์ข้อมูลสำหรับอีเลิร์นนิ่งมูเดิลในปัจจุบันทำงานอยู่บนพื้นฐานของระบบไคลเอนต์-เซิร์ฟเวอร์ คือระบบที่ประกอบด้วยคอมพิวเตอร์เครื่องเซิร์ฟเวอร์ และกลุ่มของคอมพิวเตอร์เครื่องไคลเอนต์ โดยเครื่องไคลเอนต์ต้องการพึ่งพาทรัพยากรของเครื่องเซิร์ฟเวอร์ ตัวอย่างเช่น เว็บเซิร์ฟเวอร์ (Web server) ให้บริการหน้าเว็บเพจ (Web page) กับคอมพิวเตอร์ไคลเอนต์ที่ใช้โปรแกรมเบราเซอร์ (Browser) เพื่อแสดงผลของหน้าเว็บ โดยการดาวน์โหลด (Download) หน้าเว็บเพจจากเว็บเซิร์ฟเวอร์เพื่อนำมาแสดงผลที่เครื่องไคลเอนต์ เป็นต้น ดังนั้น หากเครื่องเซิร์ฟเวอร์ไม่สามารถให้บริการได้ จะทำให้เครื่องไคลเอนต์ทุกๆ เครื่องในระบบเครือข่าย จะไม่สามารถติดต่อขอใช้ทรัพยากรจากเครื่องเซิร์ฟเวอร์ได้ ปัญหานี้เรียกว่า “ความล้มเหลวจากจุดเดียว” (Single point of failure) [6]

อย่างไรก็ตาม สถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ยังมีข้อจำกัดอีกหลายประการ ซึ่งในงานวิจัยเพื่อวิทยานิพนธ์นี้จะมุ่งเน้นการแก้ปัญหาแบบคอขวด (Bottleneck) [7] กล่าวคือหากเครื่องเซิร์ฟเวอร์เพียงที่เดียวต้องรองรับจำนวนเครื่องไคลเอนต์จำนวนมาก เพื่อเข้าใช้งานในระยะเวลาที่ใกล้เคียงกัน ส่งผลให้เครื่องเซิร์ฟเวอร์ให้บริการเครื่องไคลเอนต์ได้ช้าลง หรือ

อาจเป็นไปได้ว่าเครื่องไคลเอนต์บางส่วนไม่ได้รับบริการเลย และอาจทำให้เครื่องเซิร์ฟเวอร์ล่มจนไม่สามารถให้บริการได้อีก ทำให้สถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ขาดคุณสมบัติในการให้บริการเรื่องของการกระจายไฟล์ข้อมูล ดังนั้นงานวิจัยเพื่อวิทยานิพนธ์นี้ จึงได้ทำการศึกษาข้อจำกัดดังกล่าว และนำเสนอสถาปัตยกรรมแบบเพียร์ทูเพียร์ ด้วยโพรโทคอลบิททอเรนท์มาใช้ในระบบอีเลิร์นนิ่งมูเดิลเพื่อเป็นแนวทางการแก้ปัญหาต่อไป

2.1.2 ทบทวนวรรณกรรมที่เกี่ยวข้อง

แม้ว่าสถาปัตยกรรมแบบเพียร์ทูเพียร์จะถูกนำไปใช้งานกันอย่างแพร่หลายเพื่อแลกเปลี่ยนหรือแบ่งปันไฟล์เอกสาร รวมถึงไฟล์มัลติมีเดีย ภายในเครือข่ายคอมพิวเตอร์แบบเพียร์ทูเพียร์ แต่การประยุกต์ใช้งานโดยการนำสถาปัตยกรรมแบบเพียร์ทูเพียร์เพื่อกระจายไฟล์ข้อมูลในระบบเครือข่ายอีเลิร์นนิ่งเผยแพร่ไม่มากนัก ดังตัวอย่างเช่น

- งานวิจัย [8] ได้นำสถาปัตยกรรมแบบเพียร์ทูเพียร์ มาใช้เพื่อทำงานร่วมกับระบบอีเลิร์นนิ่ง แต่มุ่งเน้นไปทางด้านการสืบค้นข้อมูลบทเรียนเท่านั้น
- งานวิจัย [9] เสนอรายงานการวิเคราะห์ความนิยมของข้อมูลกับประสิทธิภาพของเครือข่ายแจกจ่ายข้อมูลโพรโทคอล ซึ่งสนับสนุนการทำงานแบบเพียร์ทูเพียร์ ชื่อว่าบิททอเรนท์ พบว่าความนิยมข้อมูลจะมีค่ามากหรือน้อยเท่าใด โพรโทคอลนี้ก็ยังสามารถขยายการบริการให้กับเพียร์ทุกตัวได้เป็นอย่างดีโดยไม่ได้ทำให้เกิดสูญเสียประสิทธิภาพโดยรวมในเครือข่ายไปมากเท่ากับที่พบในระบบไคลเอนต์-เซิร์ฟเวอร์ทั่วไป
- งานวิจัย [10] มีความคล้ายคลึงกับระบบที่พัฒนาในงานวิจัยเพื่อวิทยานิพนธ์นี้ โดยใช้โพรโทคอลบิททอเรนท์ ทำงานร่วมกับอีเลิร์นนิ่งโอเพนซอร์สมูเดิลเช่นกัน แต่ความแตกต่างจากงานวิจัยเพื่อวิทยานิพนธ์นี้ อยู่ที่การนำหลักการทำงานของเว็บบิท (Web Bit) แบบโอเพ่นซอร์สมาใช้ในการกระจายไฟล์ข้อมูล เพื่อพัฒนาให้โพรโทคอลบิททอเรนท์ทำงานได้อย่างมีประสิทธิภาพมากขึ้น

2.1.3 แนวคิดในการแก้ปัญหา

จากที่มาของปัญหาและทบทวนวรรณกรรมที่เกี่ยวข้องดังกล่าวข้างต้นเป็นปัจจัยสำคัญในการนำมาสู่วิธีการเสนอแนวทางการแก้ปัญหา โดยการนำสถาปัตยกรรมแบบเครือข่ายเพียร์ทูเพียร์มาใช้ ซึ่งเครือข่ายนี้ไม่ต้องพึ่งพาคอมพิวเตอร์ศูนย์กลาง (Server) และไม่ต้องพึ่งพาโปรแกรมกลุ่มใดกลุ่มหนึ่งที่อยู่บนศูนย์กลางเพื่อจัดการบริหารเครือข่าย แต่ยังคงสามารถทำงาน

ต่อไปได้ โดยอาศัยเพียร์ (Peer) ที่มีอยู่เป็นผู้จัดการเครือข่ายกันเอง ลักษณะการทำงานแบบเพียร์ทูเพียร์ เปรียบเหมือนระบบ Self-Organizing [11, 12] ซึ่งคุณสมบัติเด่นหลายประการ เช่น

1. ความคงทน (Robust) กล่าวคือ หากมีเพียร์ใด เพียร์หนึ่ง หรือกลุ่มของเพียร์ออกจากเครือข่ายอันเนื่องจากสาเหตุใดๆ ก็ตาม เช่น ผู้ใช้(ซึ่งในที่นี้จะเรียกว่า “เพียร์”) หยุดการทำงานของโปรแกรมเพียร์ทูเพียร์ หรือโปรแกรมเพียร์ทูเพียร์ทำงานผิดพลาดและปิดตัวเองลง เป็นต้น เพียร์อื่นๆ ที่อยู่ในเครือข่าย สามารถดำเนินกิจกรรมที่เคยได้ทำในเครือข่ายเพียร์ทูเพียร์ ต่อไปได้โดยไม่ส่งผลกระทบโดยรวม

2. ความคงอยู่ยาวนาน (High Availability) กล่าวคือ เครือข่ายสามารถรักษาทรัพยากร (Resource) ให้คงอยู่ยาวนานเพื่อบริการเพียร์ต่างๆได้ โดยมีวิธีการจัดหาทรัพยากร จากเพียร์อื่นมาทดแทน ด้วยวิธีการทำซ้ำซ้อน (Redundancy) เหมือนการคัดลอก (Copy) ทรัพยากรไปยังที่อื่นๆ ซึ่งหากมีจำนวนผู้ใช้มากขึ้น โอกาสที่ทรัพยากรจะถูกทำซ้ำซ้อนก็มากขึ้นไปด้วย

3. ความสามารถแบ่งเบาภาระ (Load Balancing) กล่าวคือ คุณสมบัติทางอ้อมจากการทำซ้ำซ้อน เพราะเมื่อทรัพยากรกระจายไปยังเพียร์อื่นๆ แล้วนั่น เปรียบได้ว่ามีเพียร์ให้บริการทรัพยากรเดียวกัน มีมากกว่าหนึ่งแห่ง ดังนั้น เพียร์เหล่านี้จึงสามารถแบ่งเบาภาระในการให้บริการทรัพยากรเดียวกันในกรณีที่ผู้ใช้ต้องการจำนวนมาก

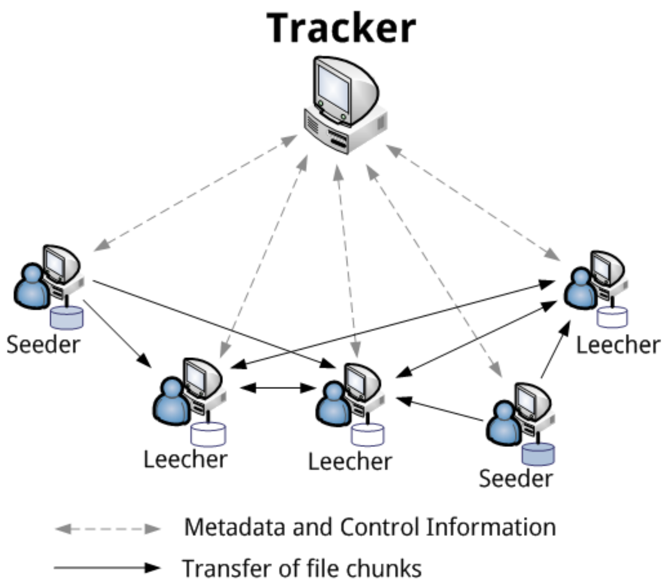
4. ความสามารถยืดขยายได้อย่างมหาศาล (Massively Scalability) กล่าวคือ เครือข่ายที่รองรับจำนวนผู้ใช้จำนวนมากๆ แต่ประสิทธิภาพของระบบไม่ได้ด้อยลง หรือด้อยลงเพียงเล็กน้อย เช่น หากเดิมเครือข่ายรองรับผู้ใช้ หนึ่งพันคนพร้อมๆ กันได้อย่างดี และเมื่อมีผู้ใช้ติดต่อกับเครือข่าย หนึ่งหมื่นคนพร้อมๆ กัน เครือข่ายก็ยังคงสามารถตอบสนองต่อความต้องการของผู้ใช้ได้เหมือนกับว่าจำนวนที่เพิ่มมาสิบเท่า ไม่ได้มีผลกระทบต่อประสิทธิภาพโดยรวม เป็นต้น

5. ความเป็นอิสระภาพสูง (High Freedom) กล่าวคือ เครือข่ายแบบเพียร์ทูเพียร์ไม่มีศูนย์กลางควบคุม ดังนั้นทุกๆเพียร์จึงมีอิสระในการใช้งานเครือข่าย เช่น การแชร์ไฟล์ การเข้าออกจากเครือข่ายตามใจชอบโดยไม่ต้องคำนึงว่าเครือข่ายโดยรวมจะเสียหาย และการเข้าถึงทรัพยากรโดยไม่ต้องขออนุญาต เป็นต้น

จากคุณลักษณะเด่นดังกล่าวข้างต้น สถาปัตยกรรมเครือข่ายแบบเพียร์ทูเพียร์จะสามารถช่วยลดภาระงานภายในเครือข่ายลงได้ เพราะเป็นการกระจายคำสั่งและการควบคุมไปยังเครื่องต่างๆ แทนที่จะให้การควบคุมเหล่านี้เป็นของเครื่องเซิร์ฟเวอร์กลาง ดังนั้น เครือข่ายแบบเพียร์ทูเพียร์ จึงมีประโยชน์มากสำหรับเครือข่ายที่ใช้งานในรูปแบบของการกระจายไฟล์ข้อมูลมากๆ จากที่หนึ่งไปสู่ที่หนึ่ง

2.1.4 กลไกการทำงานของบิททอเรนท์

โพรโทคอลบิททอเรนท์ที่ได้พิจารณาเลือกใช้ในงานวิจัยเพื่อวิทยานิพนธ์นี้ สำหรับการกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์นั้น เนื่องจากเป็นโพรโทคอลที่มีการแลกเปลี่ยนไฟล์ภายในกลุ่มเครือข่าย และมีความสามารถในการแชร์ไฟล์ข้อมูลได้เร็วกว่าการแชร์ไฟล์ข้อมูลในรูปแบบอื่นๆ ดังแสดงในภาพประกอบ 2-1

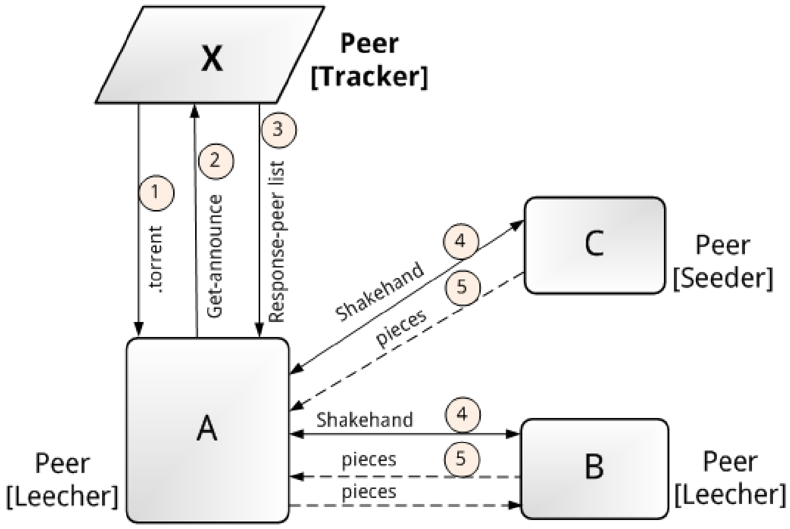


ภาพประกอบ 2-1 การติดต่อระหว่างเพียร์โดยใช้โพรโทคอลบิททอเรนท์

โพรโทคอลบิททอเรนท์มีหน่วยทำงานที่เกี่ยวข้อง ดังนี้

1. Tracker ทำหน้าที่เป็นเครื่องเซิร์ฟเวอร์ เพื่อเก็บรวบรวมไฟล์ทอเรนท์ (ไฟล์ .Torrent) ซึ่งเก็บค่า Uniform Resource Location(URL) และ Internet Protocol(IP) Address ของเครื่องไคลเอนต์ที่ทำหน้าที่เป็นผู้ส่ง (Seeder) รวมทั้งค่าสถิติหรือข้อมูลอื่นๆ ที่จำเป็นต่อการดำเนินงานส่งต่อไฟล์ทั้งระบบ
2. Torrent Client เป็นหน่วยทำงานภายในระบบ สามารถทำหน้าที่แยกออกได้เป็น 2 ลักษณะ คือ
   - Leecher ทำหน้าที่รับข้อมูล (Download) ที่ส่งมาจากหน่วยทำงานอื่น ๆ
   - Seeder ทำหน้าที่ส่งต่อข้อมูล (Upload) ออกไปอีกทอดหนึ่ง เมื่อได้รับการร้องขอจากตัว Leecher อื่นๆ ในภายหลัง

ซึ่งการทำหน้าที่ทั้งสองลักษณะข้างต้นนั้น ต้องสร้างไฟล์ทอเรนท์เพื่อใช้ประกอบการทำงานในภายหลังอีกด้วย



ภาพประกอบ 2-2 ลำดับขั้นตอนกระจายไฟล์ข้อมูลด้วยกลไกบิททอเรนท์

ในภาพประกอบ 2-2 แสดงให้เห็นรายละเอียดลำดับขั้นตอนของการกระจายไฟล์ข้อมูล ดังต่อไปนี้

1. Torrent client A ซึ่งเป็น Leecher ดาวน์โหลดไฟล์ทอเรนท์จาก Tracker ซึ่งได้ติดต่อไปก่อนหน้านี้ และเปิดอ่านข้อมูลภายในซึ่งเป็น URL (หรือ IP Address) ของ Tracker server ที่จะต้องไปติดต่อในลำดับถัดไป (ในกรณีนี้ เป็นหมายเลขของเครื่องเดิมที่ได้อ่านไฟล์ทอเรนท์มาก่อนแล้ว)
2. Torrent client A ติดต่อ Tracker Server X เพื่อส่งข้อมูลเกี่ยวกับไฟล์ที่ต้องการค้นหา
3. Tracker server X ส่งผลลัพธ์ที่ได้จากการค้นหา โดยกรณีที่ค้นพบจะเป็นรายการ IP Address ของไคลเอนต์ทั้งหมดที่ได้โหลดชิ้นส่วนไฟล์นี้ไป พร้อมบันทึก IP Address ของไคลเอนต์ A ที่ถามนี้เพิ่มลงไปในรายการด้วย แต่หากเป็นกรณีที่ค้นหาไม่พบ ก็จะแจ้งข้อความให้ทราบ
4. Tracker A ติดต่อเบื้องต้น (Handshake) ตรงไปยังไคลเอนต์ต่างๆ ตามรายการ IP Address ที่ได้รับแจ้งมา พร้อมแจ้งชิ้นส่วนของไฟล์ที่ต้องการไคลเอนต์ A จะทยอยได้รับข้อมูลชิ้นส่วนของไฟล์ที่ร้องขอจากไคลเอนต์ที่ติดต่ออยู่นั้น (เช่น ไคลเอนต์ B หรือ C) อย่างไรก็ตาม ในภาพประกอบ 2-2 จะสังเกตเห็นว่าไคลเอนต์ C ทำงานเป็น Seeder จึงมีโอกาสรับการร้องขอจากไคลเอนต์ ตัวอื่นด้วย

2.2 ปัญหาการกระจายไฟล์ข้อมูลอีเลิร์นนิ่งในเครือข่ายท้องถิ่นของชั้นเรียน

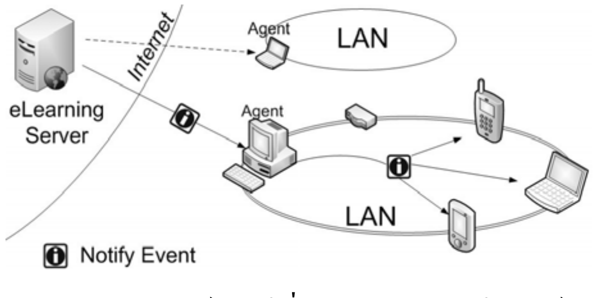
2.2.1 ที่มาของปัญหา

สถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ของระบบอีเลิร์นนิ่งโดยทั่วไป มีข้อด้อยสำคัญในการทำงาน โดยเฉพาะอย่างยิ่งในกรณีที่มีการดาวน์โหลดไฟล์ข้อมูลขนาดใหญ่จากเครื่องคอมพิวเตอร์ของนักศึกษาจำนวนมากๆ ในเวลาที่ใกล้เคียงกัน ส่งผลให้เกิดความคับคั่งของภาระงาน และปริมาณการใช้ (Traffic) ข้อมูลที่เครื่องเซิร์ฟเวอร์ ทำให้ประสิทธิภาพในการให้บริการลดลงไปอย่างมากตามภาระการทำงานที่เพิ่มสูงขึ้น จากปัญหาดังกล่าว เป็นข้อจำกัดหนึ่งที่เกิดขึ้นภายในเครือข่ายอินเทอร์เน็ต ไม่รวมถึงเครือข่ายท้องถิ่นของชั้นเรียนอิเล็กทรอนิกส์สมัยใหม่ ซึ่งไม่มีข้อจำกัดในเรื่องของทรัพยากรและปริมาณการใช้ข้อมูล ดังนั้น จึงมีการนำเสนองานวิจัยเพื่อลดภาระงานที่เกิดขึ้นบนเครื่องเซิร์ฟเวอร์กลางบนเครือข่ายอินเทอร์เน็ต โดยการออกแบบเอเจนท์ เพื่อใช้เป็นตัวกลางในการดาวน์โหลดไฟล์จากเครื่องเซิร์ฟเวอร์กลางมาอยู่กับตัวเอเจนท์ก่อน และจะส่งต่อไฟล์ข้อมูลอีกทอดหนึ่งให้กับเครื่องไคลเอนต์ที่อยู่ในเครือข่ายท้องถิ่นของชั้นเรียนต่อไป (จะกล่าวถึงอีกครั้งในหัวข้อทบทวนวรรณกรรมที่เกี่ยวข้อง)

อย่างไรก็ตาม แม้ว่าแนวคิดนี้จะนำเสนอการจัดการที่ดีที่เครื่องเซิร์ฟเวอร์อีเลิร์นนิ่งกลาง แต่ก็ยังอาจเกิดปัญหาความคับคั่งแบบคอขวดขึ้นได้ที่เอเจนท์ที่เสนอแนะขึ้นใหม่นี้ได้เช่นกัน

2.2.2 ทบทวนวรรณกรรมที่เกี่ยวข้อง

- งานวิจัย [13] ได้เสนอแนวคิดของการนำเอเจนท์เข้ามาใช้เพื่อเป็นตัวบริการแทน (Proxy Agent) สำหรับลดภาระงานและปริมาณการใช้ข้อมูลที่เครื่องบริการเซิร์ฟเวอร์อีเลิร์นนิ่งกลาง โดยรับภาระดาวน์โหลดไฟล์มาเก็บไว้เอง และกระจายต่อไปยังอุปกรณ์ต่างๆ ภายในเครือข่ายย่อยอีกทอดหนึ่ง (ดังแสดงเป็นแผนภาพในภาพประกอบ 2-3)



ภาพประกอบ 2-3 แนวทางการนำเอเจนท์มาใช้เพื่อลดภาระงานให้กับเซิร์ฟเวอร์อีเลิร์นนิ่ง [13]

- งานวิจัย [14] ได้นำเสนอด้านประสิทธิภาพของกลไกการกระจายไฟล์ข้อมูลด้วยโพรโทคอลบิททอเรนท์กับระบบอีเลิร์นนิ่งมูเดิล ซึ่งเมื่อเปรียบเทียบกับวิธีการโอนไฟล์แบบพื้นฐาน จะลดความคับคั่งของปริมาณการใช้ข้อมูลในเครือข่ายย่อยลงได้มาก อย่างไรก็ตาม งานวิจัยนี้เน้นไปที่การใช้งานจากเครือข่ายท้องถิ่น Local Area Networks (LAN) จึงไม่รวมถึงเครือข่าย UPnP ซึ่งช่วยให้คอมพิวเตอร์ต่างๆ สามารถเชื่อมต่อกันเป็นเครือข่ายได้อย่างอัตโนมัติ

2.2.3 แนวคิดในการแก้ปัญหา

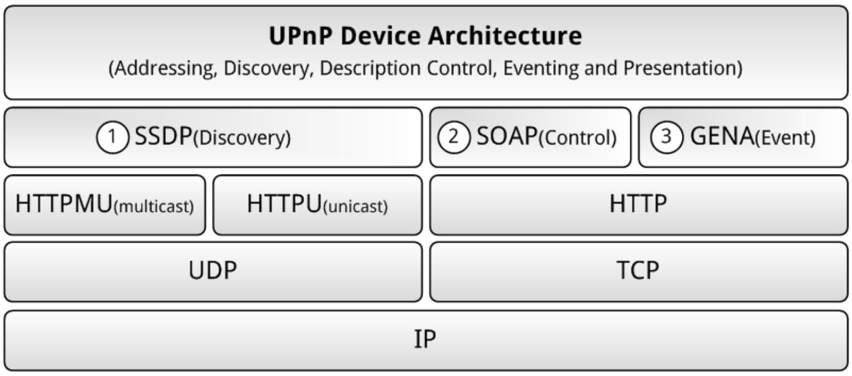
พัฒนาเอเจนท์มาช่วยในการแก้ปัญหาเพื่อลดปัญหาคอขวดในอีเลิร์นนิ่ง ให้มีคุณลักษณะดังนี้ 1) โพรโทคอลบิททอเรนท์เข้ามาสนับสนุนที่เอเจนท์ของเครือข่าย UPnP และ 2) สถาปัตยกรรมเชิงบริการตามมาตรฐาน Open Service Gateway initiative (OSGi) [15] ซึ่งเปิดโอกาสให้เขียนคำสั่งควบคุมการทำงานของซอฟต์แวร์ได้โดยง่าย ซึ่งคาดว่าการศึกษานี้จะไม่เพียงช่วยแบ่งเบาภาระของเซิร์ฟเวอร์อีเลิร์นนิ่ง แต่ยังช่วยเสริมประสิทธิภาพการกระจายไฟล์ข้อมูลโดยใช้ความร่วมมือของคอมพิวเตอร์ภายในระบบผ่านกลไกบิททอเรนท์อีกด้วย

2.2.4 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.2.4.1 โพรโทคอลภายในเครือข่าย UPnP

เครือข่าย UPnP ออกแบบให้ทำงานอยู่บนพื้นฐานของโพรโทคอลที่มีใช้แพร่หลายอยู่เดิมแล้วในเครือข่ายอินเทอร์เน็ต และเทคโนโลยีเว็บ (ดูภาพประกอบที่ 2-4 ประกอบ) เพียงแต่เพิ่มคุณลักษณะให้อุปกรณ์ทุกตัวภายในระบบสามารถติดต่อเพื่อทำงานร่วมเป็นเครือข่ายได้โดยอัตโนมัติ ตามแนวความคิดของบ้านพัก/สำนักงานอัจฉริยะ (Smart Home Network) ดังนั้นโพรโทคอลของ UPnP จึงมักมีพื้นฐานมาจากโพรโทคอลของเครือข่ายอินเทอร์เน็ต เช่น

1. โพรโทคอล Simple Service Discovery Protocol (SSDP) เพื่อค้นพบบริการ (Service Discovery) ใช้บริการของโพรโทคอล Hypertext Transfer Protocol Overview (HTTP) อีกต่อหนึ่ง ทั้งในโหมดทำงานแบบมัลติคาสต์ และยูนิคาสต์
2. โพรโทคอล Simple Object Access Protocol (SOAP) เพื่อส่งข้อความชื่อและรายละเอียดของบริการหรือข้อความที่เป็นผลลัพธ์จากบริการในลักษณะเหมือนกับบริการของเทคโนโลยีเว็บเซอร์วิส และทำงานบนโพรโทคอล HTTP เช่นกัน



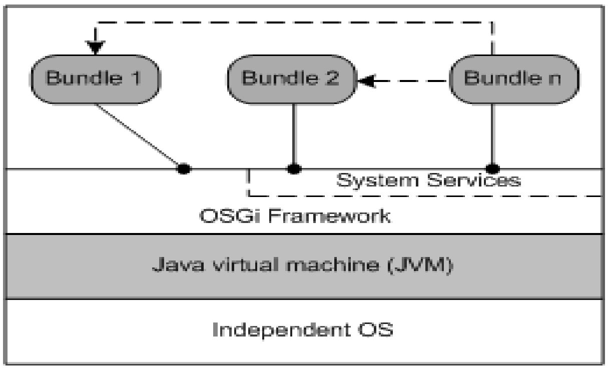
ภาพประกอบ 2-4 แผนภาพชุดโพรโทคอลของเครือข่าย UPnP

3. โพรโทคอล General Event Notification Architecture (GENA) เพื่อแจ้งเตือนเหตุการณ์ที่สนใจ ซึ่งได้สมัคร (Subscribe) ไว้ก่อนหน้าแล้ว เช่นอุปกรณ์เครื่องซักผ้าเมื่อทำงานเสร็จ จะมีข้อความแจ้งเตือนไปยังผู้สั่งงาน เป็นต้น

อย่างไรก็ตาม งานวิจัยเพื่อวิทยานิพนธ์นี้ ได้พัฒนาเอเจนท์เพื่อลดปัญหาคอขวดในอีเลิร์นนิ่ง โดยใช้โพรโทคอลบิททอเรนท์เข้ามาสนับสนุนที่เครือข่าย UPnP นั้น มีการใช้งานโพรโทคอลดังกล่าวข้างต้น ซึ่งมีโพรโทคอล SSDP ใช้สำหรับการค้นพบบริการบิททอเรนท์ โดยแสดงรายละเอียดของบริการต่าง ๆ ผ่านโพรโทคอล SOAP และแจ้งเตือนบริการบิททอเรนท์ที่ค้นพบไปยังผู้ใช้งานในเครือข่าย UPnP ผ่านโพรโทคอล GENA ดังนั้นการออกแบบระบบต้นแบบจะคำนึงถึงคุณลักษณะเด่นของโพรโทคอลดังกล่าวเป็นสำคัญ

2.2.4.2 กรอบงานมาตรฐาน OSGi

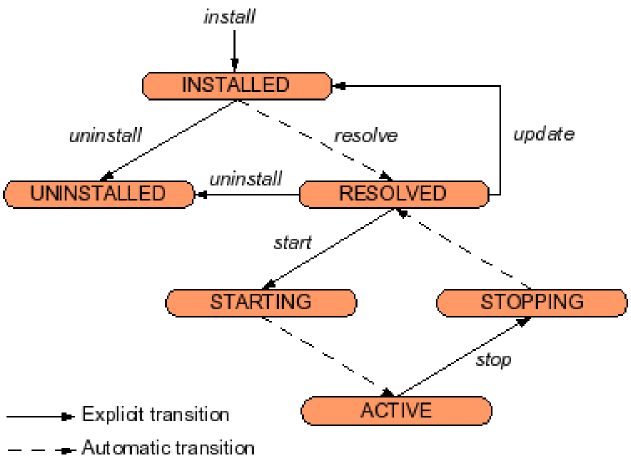
ในบริบทของงานวิจัยเพื่อวิทยานิพนธ์นี้ได้นำกรอบงานมาตรฐาน OSGi ในการพัฒนาโปรแกรม เพื่อให้มีประสิทธิภาพในการกระจายไฟล์ข้อมูลในเครือข่าย UPnP ซึ่งกรอบงาน OSGi เป็นตัวกลางประสานระหว่างเครือข่าย UPnP กับโพรโทคอลแบบเพียร์ทูเพียร์ ให้สามารถใช้งานในรูปแบบของบริการร่วมกัน โดยสร้างทุกอย่างเป็นบริการขึ้นมา ผ่านโมดูลการทำงาน เรียกว่า บันเดิล (Bundle) [16] อย่างอิสระ ทำให้สามารถเพิ่มเติม/ปรับเปลี่ยน หรือเรียกใช้บริการจากบันเดิลอื่นๆ ได้โดยง่าย ดังแสดงในภาพประกอบ 2-5



ภาพประกอบ 2-5 กรอบงานการพัฒนาแอพพลิเคชันเชิงบริการตามมาตรฐาน OSGi

ข้อดีของการพัฒนาแอพพลิเคชันเชิงบริการตามมาตรฐาน OSGi มีดังนี้

- ใช้โมดูลทำงานในลักษณะมาตรฐานของบันเดิล ที่พัฒนาขึ้นด้วยภาษาจาวา ซึ่งสะดวกต่อการจัดการและบำรุงรักษาซอฟต์แวร์
- มีการกำหนดสถานะทำงานของบันเดิลเป็นวัฏจักรของการทำงาน (ดูภาพประกอบ 2-6) ทำให้ง่ายต่อการสั่งงานในลักษณะของบริการเพื่อเปลี่ยนสถานะ ซึ่งมีความเหมาะสมสำหรับการเพิ่มขยาย หรือปรับเปลี่ยนบริการอย่างมีนัยสำคัญ

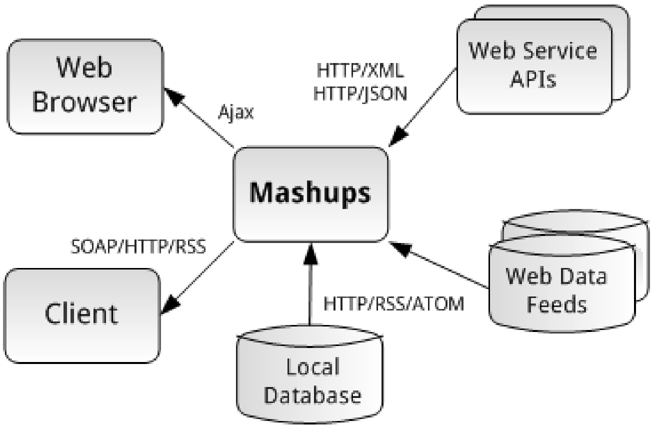


ภาพประกอบ 2-6 วัฏจักรการทำงานของบันเดิลในกรอบงาน OSGi [17]

2.2.4.3 เทคโนโลยี Mashups

เทคโนโลยี Mashups [18] ช่วยสนับสนุนการพัฒนาเว็บแอพพลิเคชัน โดยใช้แนวทางการรวมข้อมูลเว็บหลากหลายรูปแบบจากหลายแหล่งข้อมูลเข้ามาใช้ประกอบกันภายในแอพพลิเคชันเดียว แม้ว่าจะมีแนวทางการพัฒนาหลายแบบ แต่ในวิทยานิพนธ์นี้สนใจแนวทางการ

ใช้เอดิเตอร์แบบกราฟิก ซึ่งการใช้โปรแกรมจะดำเนินการผ่านรูปแบบและการเชื่อมโยงระหว่างกัน ทำให้การพัฒนาโปรแกรมสำเร็จลงได้อย่างง่ายดาย ดูภาพประกอบ 2-7



ภาพประกอบ 2-7 สถาปัตยกรรม Mashups [18]

ในงานวิจัยเพื่อวิทยานิพนธ์นี้ได้นำเทคโนโลยี Mashups มาเพื่อพัฒนาเป็นเอดิเตอร์แบบกราฟิก ด้วยเครื่องมือ Open Mashups [19] ในการออกแบบระบบต้นแบบโดยคำนึงถึงคุณลักษณะเด่นของเทคโนโลยีดังกล่าว

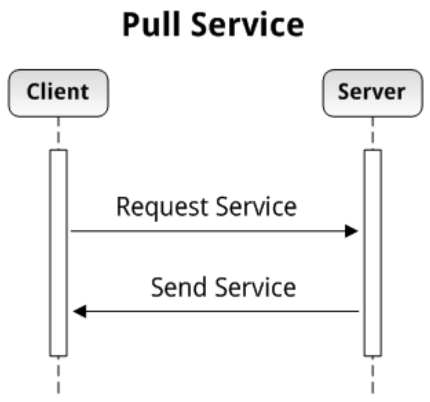
2.3 ปัญหาการใช้งานบิททอเรนท์ในเครือข่าย UPnP

2.3.1 ที่มาของปัญหา

แม้ว่าโพรโทคอลบิททอเรนท์เป็นกลไกการกระจายไฟล์ข้อมูลที่สามารถช่วยลดภาระงานของเครื่องเซิร์ฟเวอร์และลดความคับคั่งของปริมาณการใช้ข้อมูลในเครือข่ายโดยรวมได้เป็นอย่างดี แต่กลไกนี้ยังมีข้อด้อยสำคัญเรื่องของการแจ้งเตือนบริการ กล่าวคือหากผู้ใช้ต้องการกระจายไฟล์ข้อมูลด้วยกลไกบิททอเรนท์ โดยหลักการของกลไกนี้ ผู้ใช้จำเป็นต้องเข้าเว็บไซต์ที่เปิดให้บริการดาวน์โหลดไฟล์ทอเรนท์เท่านั้น จึงสามารถกระจายไฟล์ข้อมูลได้ อย่างไรก็ตาม ผู้ใช้ต้องใช้เสียเวลาเข้าเว็บไซต์รวมถึงค้นหาบริการที่ต้องการอย่างหลีกเลี่ยงไม่ได้ อีกนัยหนึ่ง ผู้ใช้จำนวนมากไม่สามารถรับรู้เหตุการณ์ หากมีการเปลี่ยนแปลงข้อมูลหรือเพิ่มบริการดาวน์โหลดไฟล์ทอเรนท์บนเว็บไซต์ ในช่วงเวลาที่ไม่มีการเข้าเว็บไซต์ดังกล่าวข้างต้น

2.3.2 ทบทวนวรรณกรรมที่เกี่ยวข้อง

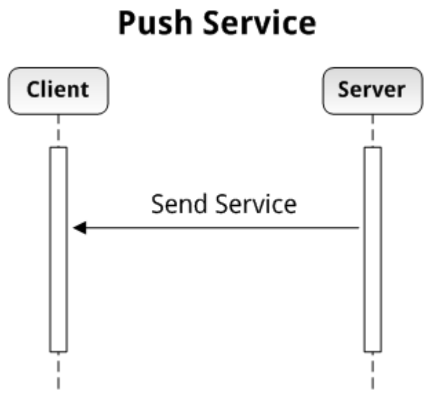
งานวิจัยที่เกี่ยวข้องกับการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการมีการเผยแพร่น้อยมาก ซึ่งงานวิจัย [20] ได้ศึกษาประโยชน์ของการกระจายไฟล์ทอเรนท์ในระบบบิททอเรนท์โดยอาศัยกลไก Really Simple Syndication (RSS) [21] ซึ่งเป็นบริการที่อยู่บนระบบอินเทอร์เน็ตโดยส่งข่าวหรือข้อมูลใหม่ๆ ให้ถึงเครื่องไคลเอนต์ตลอดเวลาที่มีการอัพเดท (Update) เพื่ออำนวยความสะดวกให้กับผู้ใช้ อย่างไรก็ตาม กลไก RSS มีการทำงานแบบ Pull [22] (ดูภาพประกอบ 2-8) ซึ่งเครื่องไคลเอนต์ที่ทำหน้าที่เป็นตัวอ่านข้อมูล (RSS Reader) [23] มีการร้องขอบริการไปยังเครื่องเซิร์ฟเวอร์เพื่อตรวจสอบการเปลี่ยนแปลงบริการ จากนั้นเครื่องเซิร์ฟเวอร์จะแจ้งเตือนกลับมายังเครื่องไคลเอนต์ หากเครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ เครื่องไคลเอนต์สามารถรับรู้เหตุการณ์ในขณะเวลานั้นทันที แต่ในทางกลับกัน เครื่องเซิร์ฟเวอร์ไม่มีการเปลี่ยนแปลงบริการ เครื่องไคลเอนต์ก็จะรับรู้เหตุการณ์เดิม โดยไม่มีการเปลี่ยนแปลงข้อมูลใหม่ ดังนั้นกลไกแบบ RSS ยังมีข้อจำกัดที่ผู้ใช้ต้องมีการร้องขอบริการที่เครื่องเซิร์ฟเวอร์เสมอ ทำให้การแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการไม่สามารถรับรู้เหตุการณ์ได้ทันที



ภาพประกอบ 2-8 ลำดับขั้นตอนการทำงานของ Pull Service

2.3.3 แนวคิดในการแก้ปัญหา

ในหัวข้อนี้ได้นำเสนอการใช้กลไกแจ้งเตือนเหตุการณ์เปลี่ยนแปลง บริการในระบบบิททอเรนท์ด้วยโมเดลแบบ Push Service [24] (ดูภาพประกอบ 2-9) ซึ่งสามารถแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการได้ทันทีเมื่อมีการเปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์ โดยที่เครื่องไคลเอนต์ไม่ต้องคอยตรวจสอบการเปลี่ยนแปลงที่เครื่องเซิร์ฟเวอร์ ส่งผลต่อการปรับปรุงประสิทธิภาพโดยรวมของระบบได้ดียิ่งกว่า



ภาพประกอบ 2-9 ลำดับขั้นตอนการทำงานของ Push Service

บทที่ 3

การออกแบบและพัฒนาระบบ

จากบทที่ผ่านมาได้นำเสนอที่มาของปัญหา รวมถึงแนวคิดในการแก้ปัญหาของงานวิจัยเพื่อวิทยานิพนธ์ ดังนั้นในบทนี้จะได้กล่าวถึงรายละเอียดในการออกแบบและพัฒนาระบบ ซึ่งได้นำเสนอระบบต้นแบบที่มีการนำสถาปัตยกรรมแบบเพียร์ทูเพียร์มาใช้เพื่อลดภาระของเครื่องเซิร์ฟเวอร์ รวมถึงการเพิ่มประสิทธิภาพการกระจายไฟล์ข้อมูลในระบบเครือข่ายให้ดียิ่งขึ้น จึงสามารถแบ่งเนื้อหาออกเป็น 3 หัวข้อที่สำคัญดังนี้ หัวข้อที่ 3.1 กลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล ตามด้วยหัวข้อที่ 3.2 กลไกบริการ OSGi เพื่อกระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP หัวข้อสุดท้ายอธิบายถึงกลไกบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP

3.1 กลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล

3.1.1 สถาปัตยกรรมระบบ

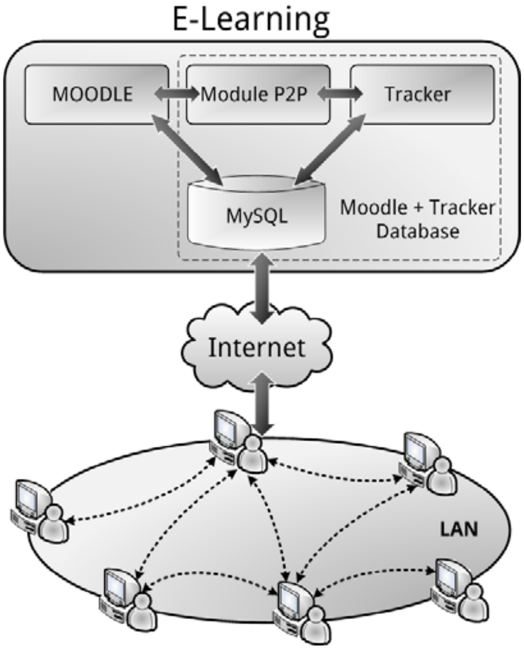
สถาปัตยกรรมระบบการกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล อาศัยกลไกการทำงานแบบบิททอเรนท์ เพื่อการกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์เพื่อแก้ปัญหาข้อจำกัดของสถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ ซึ่งกลไกการทำงานมีส่วนประกอบหลัก 3 ส่วนที่สำคัญ ดังนี้ (ดูภาพประกอบ 3-1)

1.ระบบคอมพิวเตอร์เซิร์ฟเวอร์ สำหรับโปรแกรมมูเดิล

2. ระบบคอมพิวเตอร์ไคลเอนต์โดยติดต่อสื่อสารเครือข่ายแบบเพียร์ทูเพียร์

3. ระบบคอมพิวเตอร์ Tracker เป็นการเริ่มต้นการทำงานของเครือข่ายบิททอเรนท์

ส่วนประกอบทั้ง 3 ส่วน มีการประสานการทำงานกันโดยกำหนดให้มีโมดูลการกระจายไฟล์ข้อมูลแบบ P2P และมี Tracker เป็นตัวกลางในการติดต่อสื่อสาร โดยใช้โพรโทคอลบิททอเรนท์เพื่อให้เครื่องไคลเอนต์ที่อยู่ในเครือข่ายเพียร์ทูเพียร์สามารถดาวน์โหลดข้อมูลได้ และในขณะเดียวกันเครื่องไคลเอนต์ก็สามารถกระจายไฟล์ข้อมูลให้เครื่องคอมพิวเตอร์อื่น ๆ ดาวน์โหลดข้อมูลจากเครื่องตัวเองได้ด้วยเช่นกัน



ภาพประกอบ 3-1 สถาปัตยกรรมของระบบต้นแบบการกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์

3.1.2 การเพิ่มโมดูลกระจายไฟล์ข้อมูล P2P ในมูเดิล

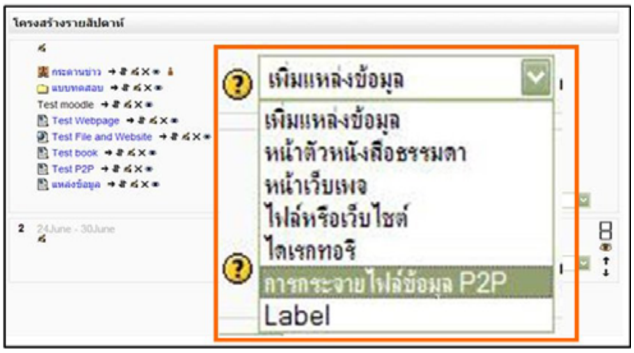
จากการศึกษาแนวทางการเพิ่มโมดูลในมูเดิล โดยค้นคว้าจากเว็บไซต์ของมูเดิลนั้นไม่ปรากฏวิธีการที่จะทำการเพิ่มโมดูลได้ ทำให้ผู้วิจัยได้ศึกษากลไกการทำงานของมูเดิลในส่วนที่เชื่อมโยงไปยังไฟล์ในการทำงานของโมดูลที่มีอยู่ในมูเดิล พบว่าโมดูลการแชร์ไฟล์ของมูเดิล จะเก็บอยู่ที่ … \moodle\mod\resource\type ซึ่งจะมีรายละเอียดของโมดูลที่ใช้งานของอีเลิร์นนิ่งมูเดิลเดิมที่ใช้งานอยู่ในปัจจุบัน ดังนี้

- Directory คือโมดูลไดเรกทรอรี
- File คือโมดูลไฟล์หรือเว็บไซต์
- Html คือโมดูลหน้าเว็บเพจ
- Text คือโมดูลหน้าหนังสือธรรมดา เป็นต้น

ซึ่งภายในโฟลเดอร์ของโมดูลจะประกอบไปด้วยไฟล์ 2 ไฟล์ด้วยกัน ยกตัวอย่างโมดูลหน้าเว็บเพจ จะประกอบไปด้วยไฟล์ชื่อ html.html และ resource.class.php

ดังนั้นแนวทางในการเพิ่มโมดูลกระจายไฟล์ข้อมูล P2P จึงสร้างโฟลเดอร์ชื่อ P2P มาเก็บไว้ยังตำแหน่ง …\moodle\mod\resource\type รวมทั้งสร้างไฟล์ P2P.html และ resource.class .php อยู่ในโฟลเดอร์ P2P

ในส่วนของการแก้ไขโค้ดโปรแกรม ต้องแก้ไขคลาสและฟังก์ชันของไฟล์ resource.class.php การเพิ่มโมดูลการกระจายไฟล์ข้อมูล P2P ต้องแก้ไข ชื่อคลาส และชื่อฟังก์ชัน จากคลาส resource_html เป็นคลาส resource_p2p และฟังก์ชัน resource_html เป็นฟังก์ชัน resource_p2p เมื่อแก้ไขข้อมูลดังกล่าวแล้ว หากเข้าใช้งานมูเดิล แหล่งข้อมูลก็จะปรากฏชื่อรายการที่มีข้อความจากโมดูลกระจายไฟล์ข้อมูลแบบ P2P ดังแสดงในภาพประกอบที่ 3-2

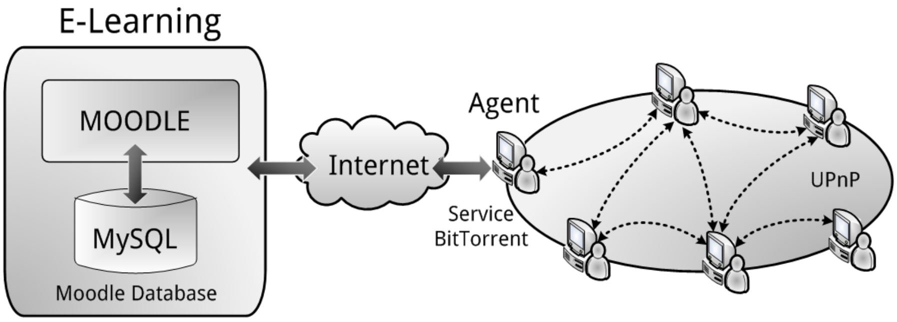


ภาพประกอบ 3-2 รายการที่ปรากฏชื่อโมดูล กระจายไฟล์ข้อมูล P2P

3.2 กลไกบริการ OSGi เพื่อกระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP

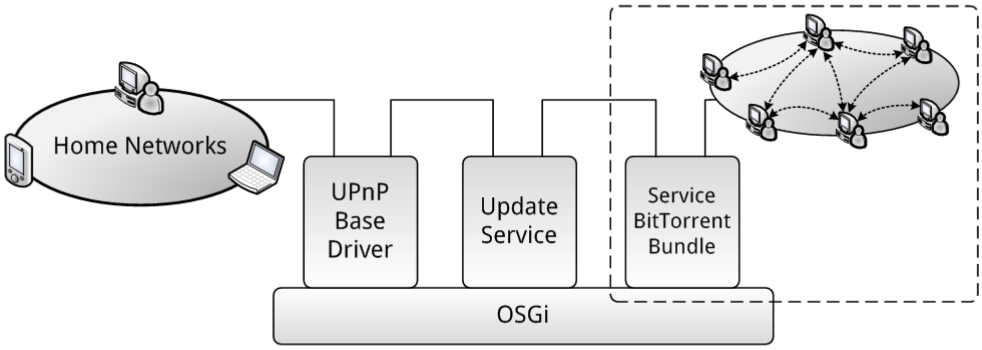
3.2.1 สถาปัตยกรรมระบบ

สถาปัตยกรรมระบบการกระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP โดยมีเอเจนท์เป็นตัวกลางในการดาวน์โหลดไฟล์ข้อมูลจากอีเลิร์นนิ่งมูเดิล ดังแสดงในภาพประกอบ 3-3



ภาพประกอบ 3-3 สถาปัตยกรรมกระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่ง ในเครือข่าย UPnP

กลไกบริการ OSGi กระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP อาศัยกลไกการพัฒนาซอฟต์แวร์ตามมาตรฐาน OSGi เพื่อแก้ปัญหาข้อจำกัดกระจายไฟล์ข้อมูลในเครือข่าย UPnP กลไกดังกล่าว มีการแยกส่วนทำงานย่อยให้เป็นอิสระต่อกันผ่านบันเดิล (ดูภาพประกอบ 3-4 ประกอบ) ในส่วนของกรอบสี่เหลี่ยมเส้นประ เป็นการพัฒนาบริการบิททอเรนท์เพื่อเพิ่มการทำงานกระจายไฟล์ข้อมูลในเครือข่าย UPnP



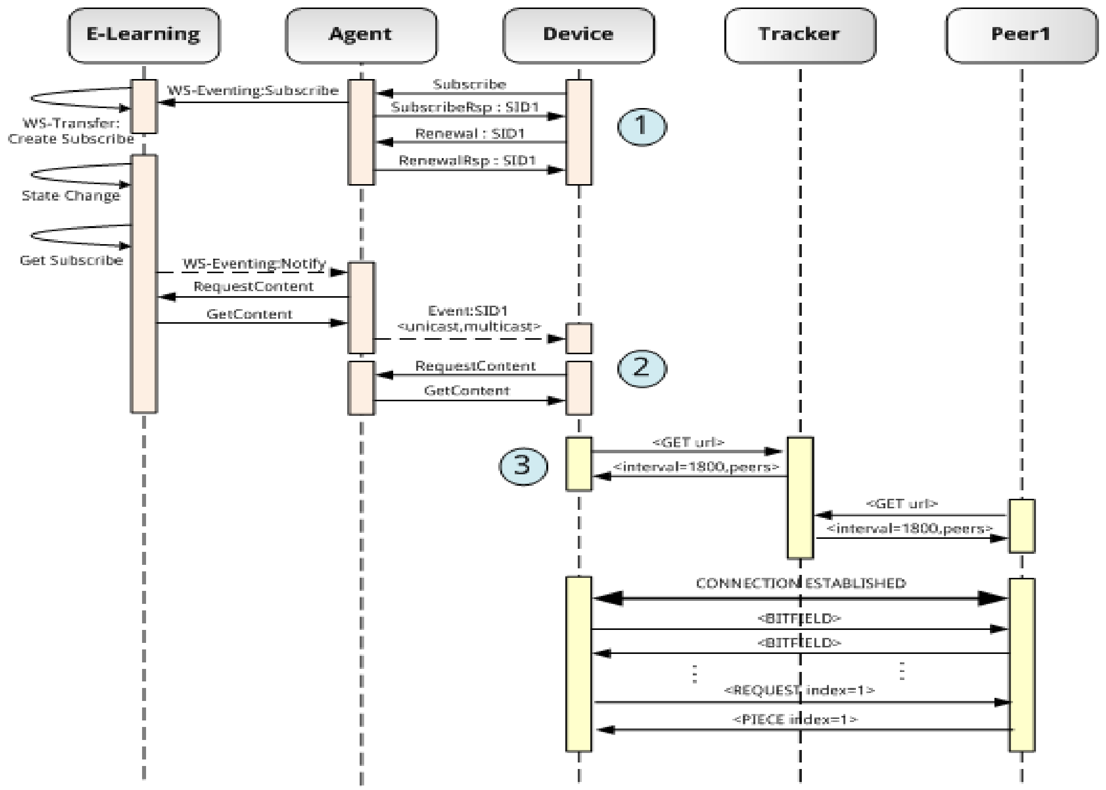
ภาพประกอบ 3-4 การเพิ่มบริการบิททอเรนท์ผ่านบันเดิลบนมาตรฐาน OSGi

3.2.2 ระบบบิททอเรนท์ภายในเครือข่าย UPnP

โดยพื้นฐานแล้ว กลไกบิททอเรนท์ที่เพิ่มเติมขึ้นในวิทยานิพนธ์นี้ จะอยู่ในส่วนของช่วงการแลกเปลี่ยนชิ้นส่วนของไฟล์ข้อมูล ภายหลังจากที่ได้ผ่านกระบวนการค้นหาบริการและร้องขอเพื่อใช้บริการแล้ว ซึ่งจะได้อธิบายเป็นลำดับๆ ไป ตามขั้นตอนที่ระบุไว้ในแผนภาพเวลาในภาพประกอบที่ 3-5 ดังนี้

1) ช่วงการแจ้งเตือนเหตุการณ์

ในช่วงนี้เครื่องผู้เรียน (Device) จะเริ่มส่ง Subscribe Message เพื่อร้องขอลงทะเบียนบริการไปยังเอเจนท์ จากนั้นตอบกลับด้วย SubscribeResponse พร้อมกับแนบหมายเลขของการ Subscribe นั้นๆ ในที่นี้คือ SID1 จากนั้น เอเจนท์จะสร้างช่องทางการเชื่อมต่อไปยังอีเลิร์นนิ่ง เพื่อค้นหาบริการบนเซิร์ฟเวอร์ และลงทะเบียนบริการด้วยการอ่าน Web Services Description Language (WSDL) จากนั้นอีเลิร์นนิ่งจะสร้าง subscription เก็บไว้ในหน่วยความจำ เมื่อทำงานไปซักระยะหนึ่ง หากบริการที่ลงทะเบียนไว้มีการเปลี่ยนแปลงจะเรียก subscription ที่เก็บไว้ และแจ้งเตือนกลับไปยังเอเจนท์ของกลุ่ม เมื่อเอเจนท์ได้รับการแจ้งเตือน ก็สามารถแจ้งเตือนไปยังเครื่องผู้เรียน ในกลุ่มเครือข่าย LAN ได้



ภาพประกอบ 3-5 แผนภาพลำดับเวลาการค้นหาบริการเพื่อกระจายไฟล์ข้อมูลด้วยบิททอเรนท์ในเครือข่าย UPnP

2) ช่วงการดาวน์โหลดไฟล์เอกสาร

ในช่วงนี้ใช้การส่งต่อไฟล์เอกสาร โดยเมื่อเอเจนท์ในเครือข่าย ทราบว่ามีสถานะบริการเปลี่ยนแปลงไปโดยร้องขอ (RequestContent) และรับไฟล์ดังกล่าว (getContent) สำหรับการแจ้งเตือนจากเอเจนท์ไปยังเครื่องผู้เรียน ในเครือข่ายใช้การจัดการโดยผู้ใช้งานบนเครื่องเอเจนท์ โดยส่งคำสั่งการแจ้งเตือนไปยังเครื่องผู้เรียนในเครือข่าย LAN ก็จะทราบว่ามีบริการดังกล่าว อยู่ในเครือข่าย ก็จะร้องขอ (RequestContent) และดาวน์โหลดจากเอเจนท์ (getContent) โดยตรง ไม่ต้องกระทำบนเครื่องเซิร์ฟเวอร์

3) ช่วงการกระจายไฟล์ข้อมูลแบบบิททอเรนท์

ช่วงนี้เริ่มต้นการทำงานโดยหลังจากเครื่องผู้เรียนที่อยู่ในเครือข่าย LAN ทุกๆ เครื่องได้ดาวน์โหลดไฟล์ทอเรนท์จากเอเจนท์ หลังจากนั้น เครื่องผู้เรียนเปิดไฟล์ทอเรนท์ ด้วยโปรแกรม µTorrent [25] ซึ่งเป็นโปรแกรมสำหรับเป็นไคลเอนต์ เพื่อช่วยในการกระจายไฟล์แบบบิททอเรนท์ โดยมีลำดับขั้นตอนเริ่มต้นที่เครื่องผู้เรียนจะติดต่อไปยัง Tracker เพื่อร้องขอ URL

(GET URL) รวมถึง IP Address ของเครื่องผู้เรียนอื่น ๆ ในเครือข่าย จากนั้น Tracker ก็จะตอบกลับด้วยการส่ง URL และ IP Address ของเครื่องผู้เรียนที่อยู่ในเครือข่าย LAN ทั้งหมด และเครื่องผู้เรียนทั้งหมดก็จะเชื่อมต่อกันโดยอัตโนมัติเพื่อแลกเปลี่ยนชิ้นส่วนของไฟล์ในระบบต่อไป

3.3 กลไกบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP

ในหัวข้อนี้จะกล่าวถึง กลไกแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการบิททอเรนท์ด้วยวิธีการแบบพุช (Push) ซึ่งเป็นกลไกที่ช่วยให้ผู้ใช้รับรู้เหตุการณ์ หากมีการเปลี่ยนแปลงข้อมูลที่เครื่องเซิร์ฟเวอร์ โดยเครื่องไคลเอนต์ไม่ต้องร้องขอบริการใด ๆ

3.3.1 การแจ้งเตือนบริการและระบบบิททอเรนท์ภายในเครือข่าย UPnP

หัวข้อนี้ได้ประยุกต์ใช้ลักษณะเด่นของโพรโทคอลภายในเครือข่าย UPnP ซึ่งมีจุดเด่นในเรื่องของการค้นหาบริการโดยอัตโนมัติ โดยไม่ต้องตั้งค่าใด ๆ (Zero Configuration) และมีการแจ้งเตือนบริการภายในเครือข่าย UPnP เป็นพื้นฐาน ดังนั้นจึงออกแบบและพัฒนาระบบต้นแบบเพื่อให้สามารถแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการเพื่อกระจายไฟล์ข้อมูลแบบบิททอเรนท์ในเครือข่าย UPnP ดังแสดงขั้นตอนในภาพประกอบ 3-6

1) ช่วงการติดต่อลงทะเบียนบริการ

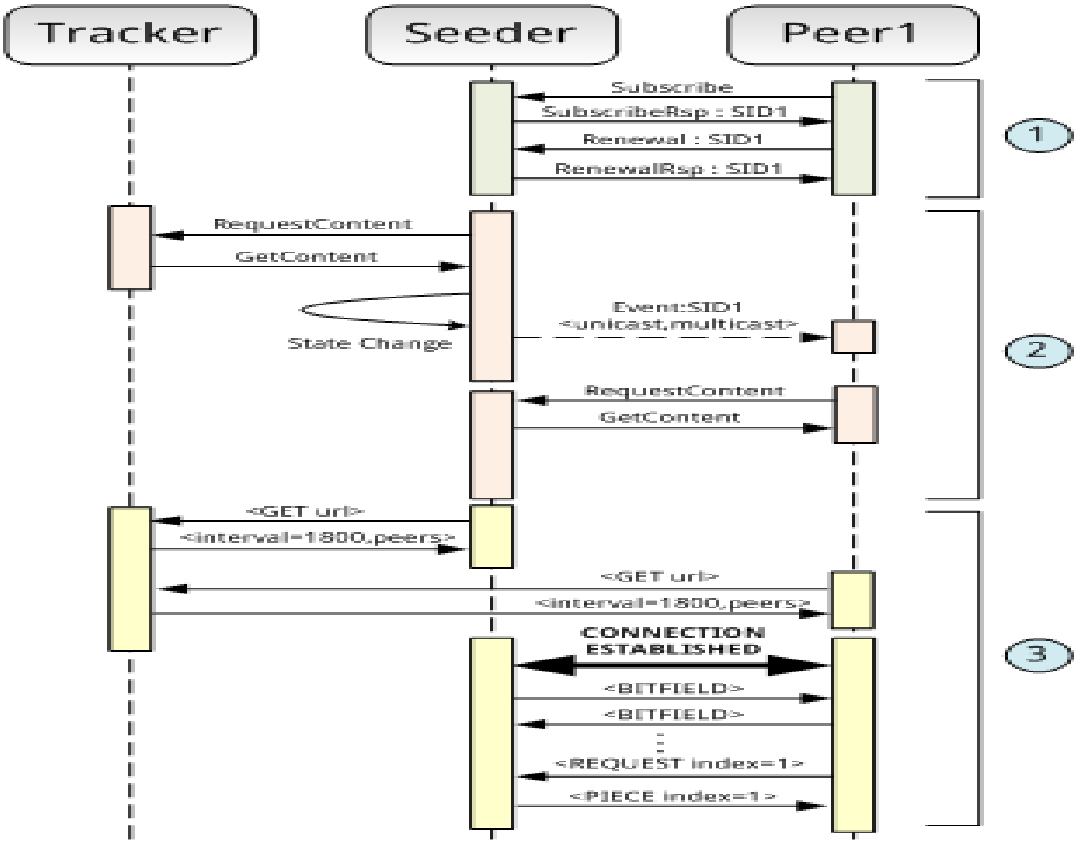
Peer ส่ง Subscribe Message เพื่อขอลงทะเบียนบริการไปยัง Seeder จากนั้นตอบกลับด้วย SubscribeResponse พร้อมกับแนบหมายเลขของการ Subscribe นั้นๆ ในที่นี้คือ SID1 ของ Peer ที่ร้องขอนั้นๆ

2) ช่วงการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการ

ในช่วงนี้ใช้แนวทางแบบ Push Service เข้ามาประยุกต์ใช้กับการส่งต่อไฟล์เอกสารในเครือข่าย เมื่อทราบว่ามีสถานะบริการเปลี่ยนแปลงก็ร้องขอ (RequestContent) และรับไฟล์

3) ช่วงการกระจายไฟล์ข้อมูลแบบบิททอเรนท์

ช่วงนี้เริ่มต้นการทำงานโดยหลังจากเครื่อง ได้รับไฟล์ทอเรนท์จาก Seeder เมื่อเปิดไฟล์ทอเรนท์ด้วยโปรแกรม µTorrent จากนั้น Peer จะทำการติดต่อไปยัง Tracker เพื่อร้องขอ URL เพื่อทำการติดต่อไปยัง Peer อื่น ๆ ต่อไป



ภาพประกอบ 3-6 แผนภาพลำดับเวลาการลงทะเบียนบริการ, การแจ้งเตือนเหตุการณ์และกระจายไฟล์ข้อมูลบิททอเรนท์ในเครือข่าย UPnP

การออกแบบและพัฒนาระบบทั้ง 3 หัวข้อมุ่งเน้นเพื่อพัฒนากลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิลภายในเครือข่ายอินเทอร์เน็ต และเครือข่ายท้องถิ่นแบบชั้นเรียนอิเล็กทรอนิกส์สมัยใหม่ รวมถึงการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการ เพื่อเป็นต้นแบบสำหรับการนำสถาปัตยกรรมแบบเพียร์ทูเพียร์ที่ใช้กลไกการทำงานแบบบิททอเรนท์มาใช้เพื่อลดภาระความคับคั่งแบบคอขวดที่เครื่องเซิร์ฟเวอร์ และเพิ่มประสิทธิภาพด้านเวลาในการกระจายไฟล์ข้อมูลภายในระบบเครือข่ายให้ดียิ่งขึ้น

บทที่ 4

ผลการทดลอง

ในบทนี้จะกล่าวถึงผลการทดลองระบบที่ได้ออกแบบและพัฒนาดังที่กล่าวมาใน บทที่ 3 โดยเรียงลำดับการนำเสนอหัวข้อดังนี้ 4.1 ผลการทดลอง กลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล ตามด้วยหัวข้อ 4.2 ผลการทดลอง กลไกบริการ OSGi เพื่อกระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP และหัวข้อสุดท้าย ผลการทดลอง กลไกบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP

4.1 ผลการทดลอง กลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล

4.1.1 สภาพแวดล้อมในการทดลอง

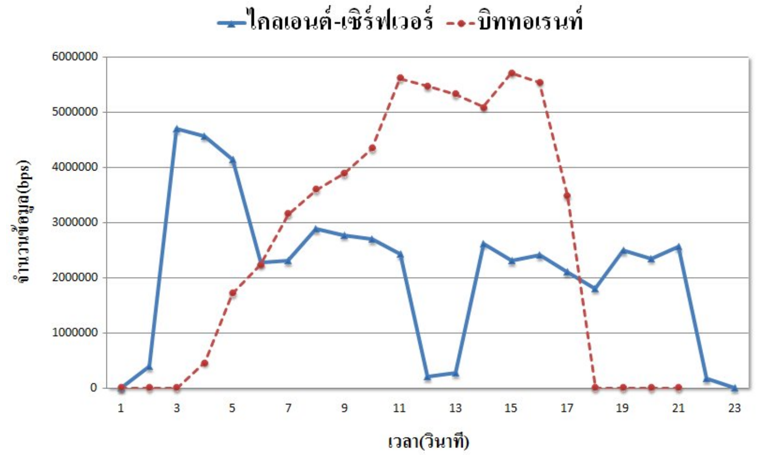
การทดลองดาวน์โหลดไฟล์ข้อมูลนั้นได้กำหนดจำนวนเครื่องคอมพิวเตอร์ในการดาวน์โหลดทั้งหมด 40 เครื่อง มีคุณสมบัติดังนี้ Pentium(R) D CPU 3.40 GHz, RAM 512 MB และใช้ระบบปฏิบัติการ Windows XP เชื่อมต่อเข้าด้วยกันเป็นเครือข่าย LAN ที่อัตราความเร็ว 100 Mbps ส่วนของไฟล์ข้อมูลในการดาวน์โหลดแบ่งออกเป็น 3 ขนาด คือ 50 MByte, 600 MByte และ 4330 MByte ในการทดลองได้ใช้ซอฟต์แวร์ Onlineeye pro [26] เป็นเครื่องมือช่วยเฝ้าตรวจเวลาในการดาวน์โหลดข้อมูลและเก็บสถิติในการดาวน์โหลดข้อมูลเพื่อนำข้อมูลนั้นมาแสดงเป็นกราฟในลำดับต่อไป

4.1.2 ผลการทดลอง

ทำการดาวน์โหลดไฟล์ข้อมูลโดยมีจำนวนเครื่องคอมพิวเตอร์ทำหน้าที่เป็นเครื่องไคลเอนต์ แบ่งเป็น 5 กลุ่ม คือ 1, 2, 4, 16 และ 40 เครื่อง ตามลำดับ เพื่อเปรียบเทียบเวลาการดาวน์โหลดข้อมูลระหว่างระบบไคลเอนต์-เซิร์ฟเวอร์ และบิททอเรนท์ ดังแสดงในภาพประกอบ 4-1 ถึง 4-5

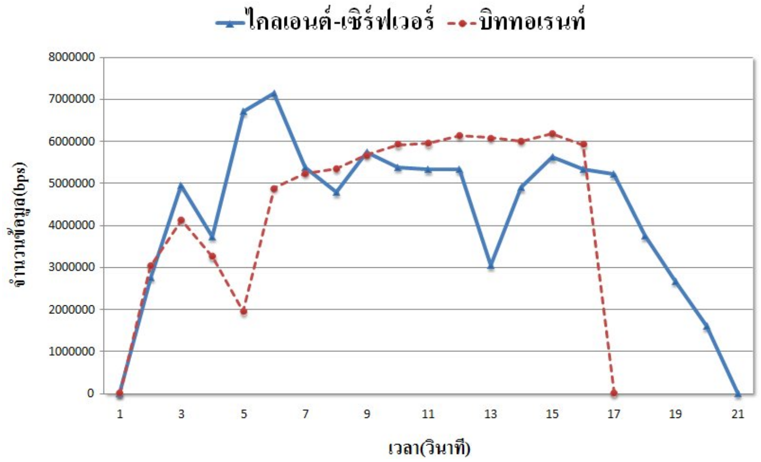
1) ดาวน์โหลดไฟล์ข้อมูลขนาด 50 MByte

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 1 เครื่อง
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 23 วินาที
  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 21 วินาที



ภาพประกอบ 4-1 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 1 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 2 เครื่อง
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 21 วินาที
  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 17 วินาที

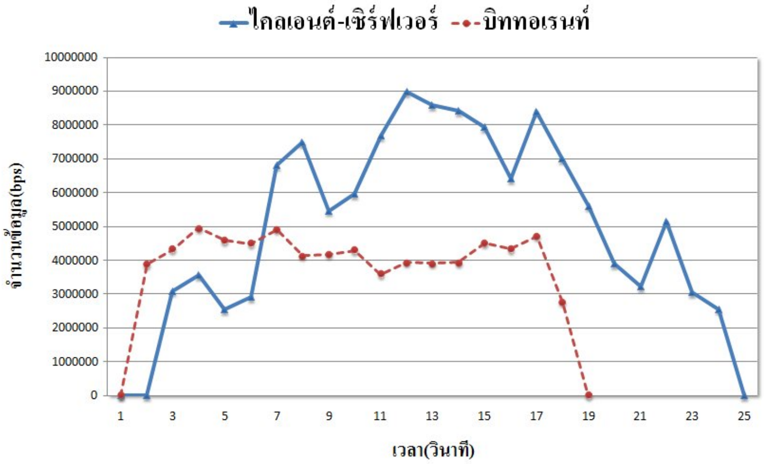


ภาพประกอบ 4-2 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 2 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 4 เครื่อง

  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 25 วินาที

  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 19 วินาที

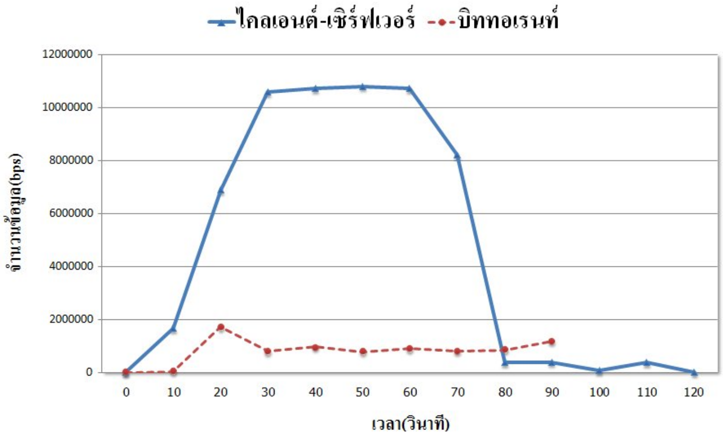


ภาพประกอบ 4-3 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 4 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 16 เครื่อง

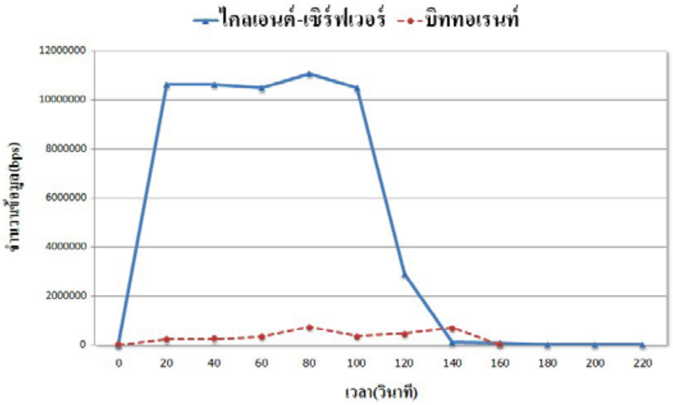
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 120 วินาที

  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 90 วินาที



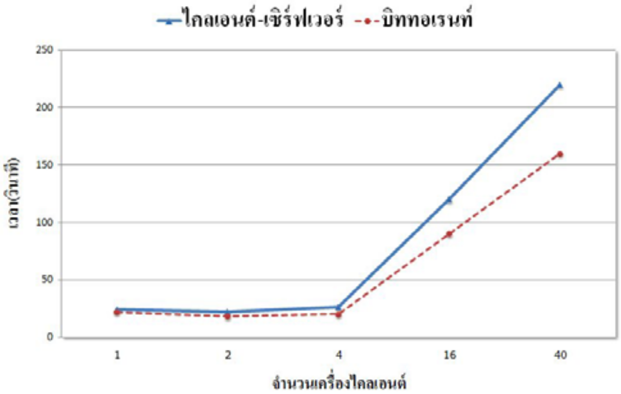
ภาพประกอบ 4-4 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 16 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 40 เครื่อง
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 220 วินาที
  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 160 วินาที



ภาพประกอบ 4-5 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte จำนวนเครื่องดาวน์โหลด 40 เครื่อง

อย่างไรก็ตามเมื่อเปรียบเทียบเวลาการดาวน์โหลดทั้งสองระบบดังแสดงในภาพประกอบ 4-1 ถึง 4-5 จำนวนเครื่องไคลเอนต์ในการดาวน์โหลด 1, 2, และ 4 เครื่อง ใช้เวลาดาวน์โหลดใกล้เคียงกันทั้งสองระบบ รวมถึงภาระงานที่เครื่องเซิร์ฟเวอร์มีจำนวนการดาวน์โหลดปริมาณที่ใกล้เคียงกัน แต่เมื่อจำนวนเครื่องไคลเอนต์ในการดาวน์โหลดที่ 16 และ 40 เครื่อง จะพบว่าบิททอเรนท์ มีความเร็วด้านเวลาในการดาวน์โหลดข้อมูลเสร็จสมบูรณ์และภาระงานที่เครื่องเซิร์ฟเวอร์ใช้เวลาน้อยกว่าไคลเอนต์-เซิร์ฟเวอร์ อย่างมีนัยสำคัญ ดังแสดงในภาพประกอบ 4-6



ภาพประกอบ 4-6 การเปรียบเทียบเวลาในการดาวน์โหลดข้อมูลระหว่างระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 50 MByte

ตาราง 4-1 ตารางเปรียบเทียบประสิทธิภาพด้านเวลา ไฟล์ข้อมูลขนาด 50 MByte

จำนวนเครื่องไคลเอนต์	ไคลเอนต์-เซิร์ฟเวอร์ (เวลาดาวน์โหลดข้อมูล) (วินาที)	บิททอเรนท์ (เวลาดาวน์โหลดข้อมูล) (วินาที)	เปรียบเทียบประสิทธิภาพด้านเวลา (เปอร์เซ็นต์)
1	23	21	9
2	21	17	19
4	25	19	24
16	120	90	25
40	220	160	27

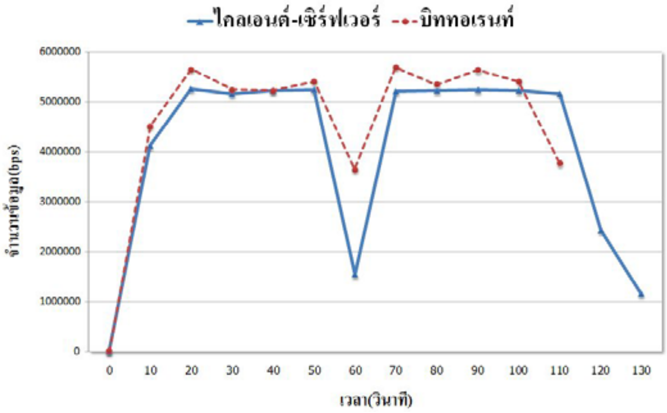
จากตารางที่ 4-1 แสดงถึงประสิทธิภาพด้านเวลาในการดาวน์โหลดไฟล์ข้อมูลบิททอเรนท์เมื่อเปรียบเทียบกับไคลเอนต์-เซิร์ฟเวอร์ ซึ่งจำนวนค่าเปอร์เซ็นต์ของประสิทธิภาพด้านเวลาที่เพิ่มขึ้นตามจำนวนเครื่องไคลเอนต์ดังตาราง เมื่อพิจารณาผลการทดลองดังกล่าว สรุปได้ว่าหากเครื่องไคลเอนต์จำนวนมากขึ้นทำให้ประสิทธิภาพด้านเวลาในการดาวน์โหลดข้อมูลแบบบิททอเรนท์มีเปอร์เซ็นต์ที่มากขึ้นด้วยเช่นกัน

2) ดาวน์โหลดไฟล์ข้อมูลขนาด 600 MByte

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 1 เครื่อง

  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 130 วินาที

  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 110 วินาที

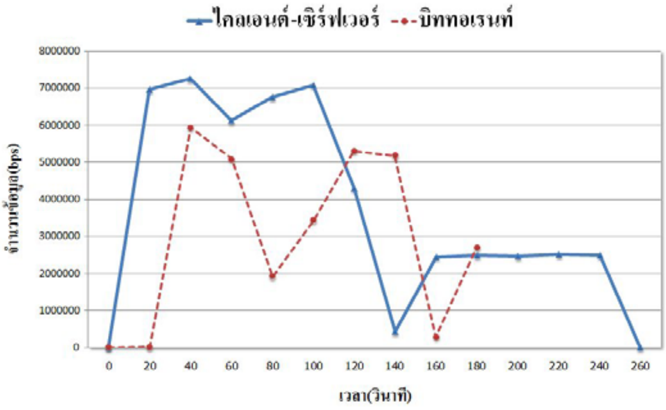


ภาพประกอบ 4-7 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 1 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 2 เครื่อง

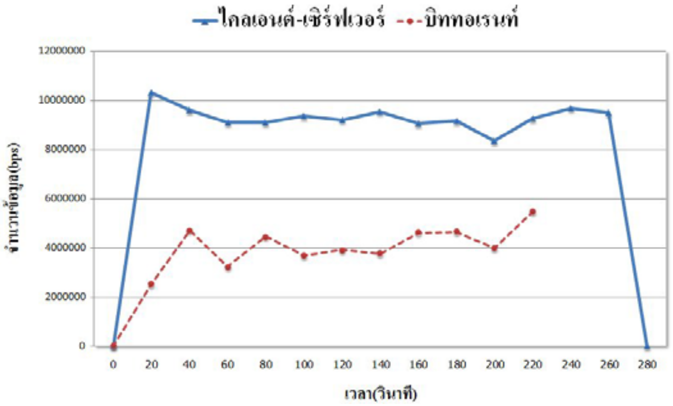
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 260 วินาที

  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 180 วินาที



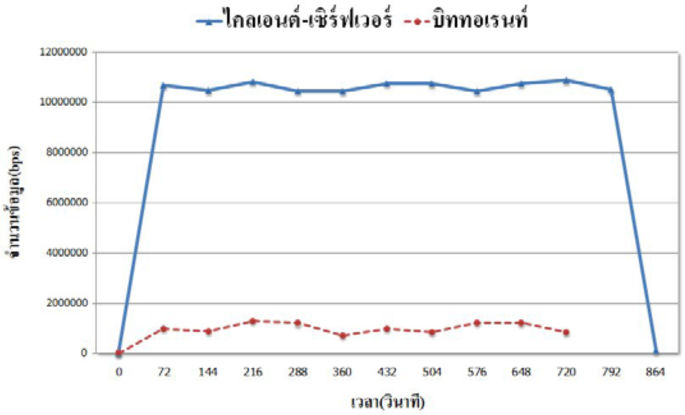
ภาพประกอบ 4-8 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 2 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 4 เครื่อง
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 280 วินาที
  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 220 วินาที



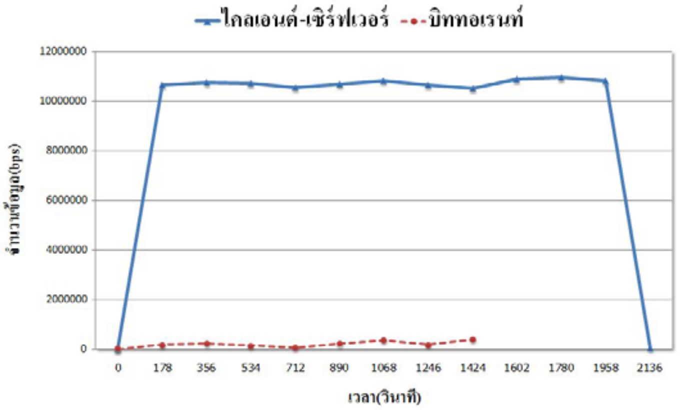
ภาพประกอบ 4-9 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 4 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 16 เครื่อง
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 864 วินาที
  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 720 วินาที



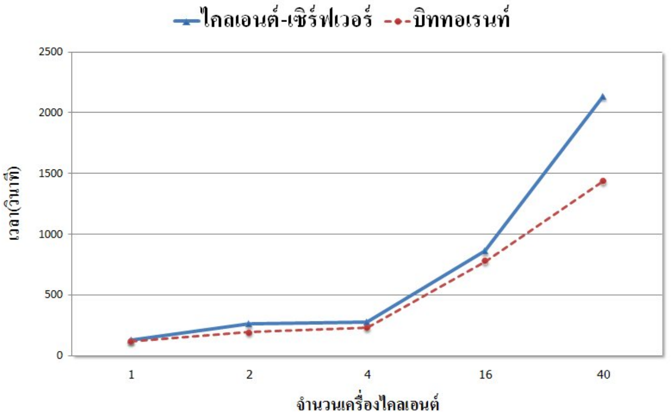
ภาพประกอบ 4-10 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 16 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 40 เครื่อง
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 2136 วินาที
  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 1424 วินาที



ภาพประกอบ 4-11 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte จำนวนเครื่องดาวน์โหลด 40 เครื่อง

อย่างไรก็ตามเมื่อเปรียบเทียบเวลาการดาวน์โหลดทั้งสองระบบดังแสดงในภาพประกอบ 4-7 ถึง 4-11 จำนวนเครื่องไคลเอนต์ในการดาวน์โหลด 1 และ 2 เครื่อง ใช้เวลาดาวน์โหลดใกล้เคียงกันทั้งสองระบบ รวมถึงภาระงานที่เครื่องเซิร์ฟเวอร์มีจำนวนการดาวน์โหลดปริมาณที่ใกล้เคียงกัน แต่เมื่อจำนวนเครื่องไคลเอนต์ในการดาวน์โหลดที่ 4, 16 และ 40 เครื่อง จะพบว่าบิททอเรนท์ มีความเร็วด้านเวลาในการดาวน์โหลดข้อมูลเสร็จสมบูรณ์และภาระงานที่เครื่องเซิร์ฟเวอร์ใช้เวลาน้อยกว่าไคลเอนต์-เซิร์ฟเวอร์ อย่างมีนัยสำคัญ ดังแสดงในภาพประกอบ 4-12



ภาพประกอบ 4-12 การเปรียบเทียบเวลาในการดาวน์โหลดข้อมูลระหว่างระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 600 MByte

ตาราง 4-2 ตารางเปรียบเทียบประสิทธิภาพด้านเวลา ไฟล์ข้อมูลขนาด 600 MByte

จำนวนเครื่องไคลเอนต์	ไคลเอนต์-เซิร์ฟเวอร์ (เวลาดาวน์โหลดข้อมูล) (วินาที)	บิททอเรนท์ (เวลาดาวน์โหลดข้อมูล) (วินาที)	เปรียบเทียบประสิทธิภาพด้านเวลา (คิดเป็นเปอร์เซ็นต์)
1	130	110	15
2	260	180	31
4	280	220	21
16	864	720	17
40	2136	1424	33

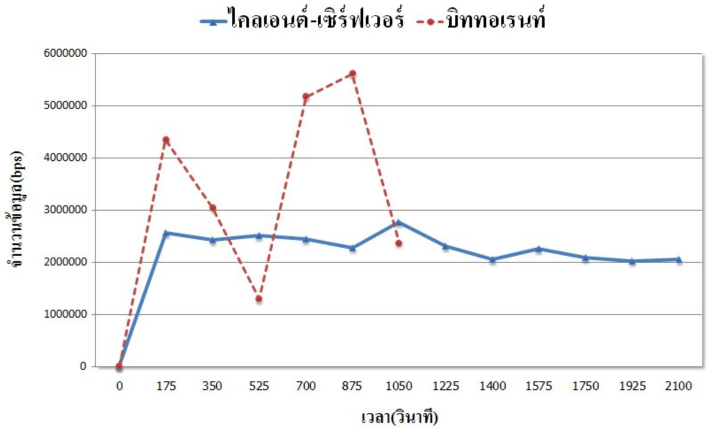
จากตารางที่ 4-2 แสดงถึงประสิทธิภาพด้านเวลาในการดาวน์โหลดไฟล์ข้อมูลบิททอเรนท์เมื่อเปรียบเทียบกับไคลเอนต์-เซิร์ฟเวอร์ ซึ่งจำนวนค่าเปอร์เซ็นต์ของประสิทธิภาพด้านเวลาที่เพิ่มขึ้นตามจำนวนเครื่องไคลเอนต์ดังตาราง เมื่อพิจารณาผลการทดลองดังกล่าว สรุปได้ว่าหากเครื่องไคลเอนต์จำนวนมากขึ้นทำให้ประสิทธิภาพด้านเวลาในการดาวน์โหลดข้อมูลแบบบิททอเรนท์มีเปอร์เซ็นต์ที่มากขึ้นด้วยเช่นกัน

3) ดาวน์โหลดไฟล์ข้อมูลขนาด 4330 MByte

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 1 เครื่อง

  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 2100 วินาที

  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 1050 วินาที

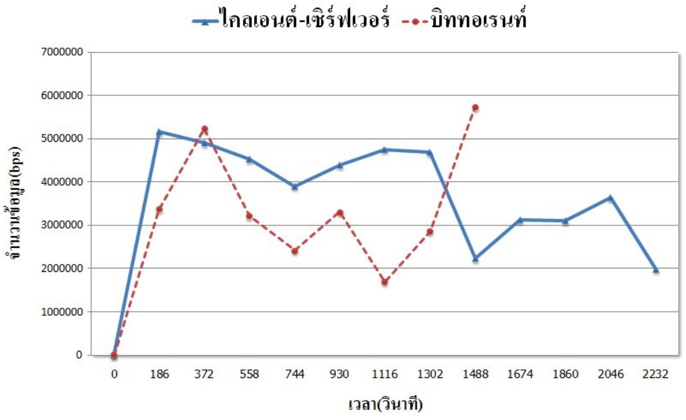


ภาพประกอบ 4-13 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 1 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 2 เครื่อง

  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 2232 วินาที

  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 1488 วินาที

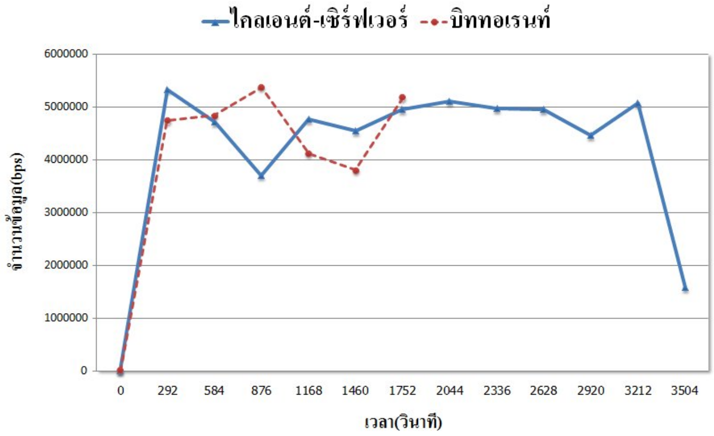


ภาพประกอบ 4-14 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 2 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 4 เครื่อง

  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 3504 วินาที

  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 1752 วินาที

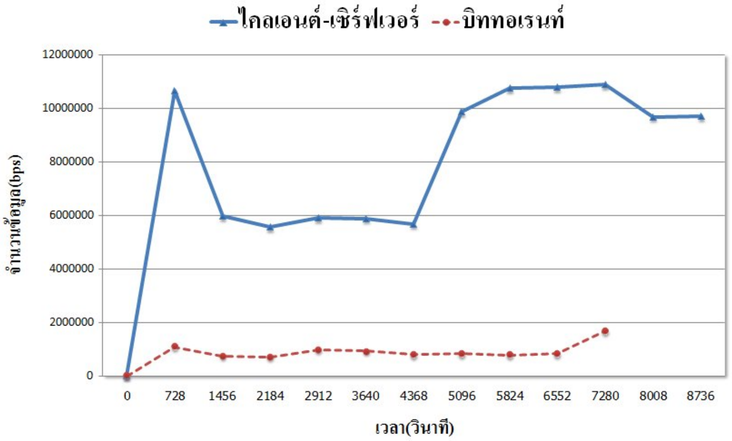


ภาพประกอบ 4-15 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 4 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 16 เครื่อง

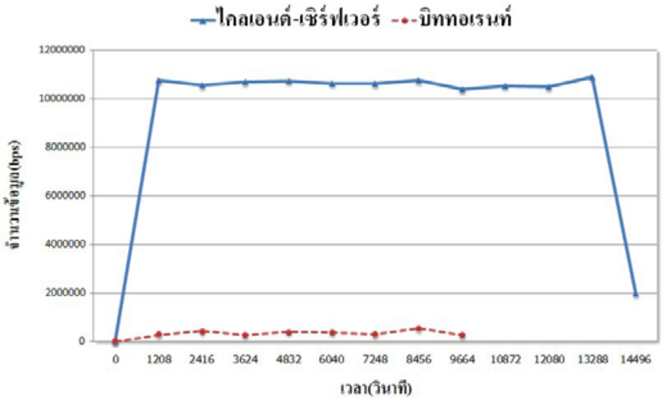
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 8736 วินาที

  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 7280 วินาที



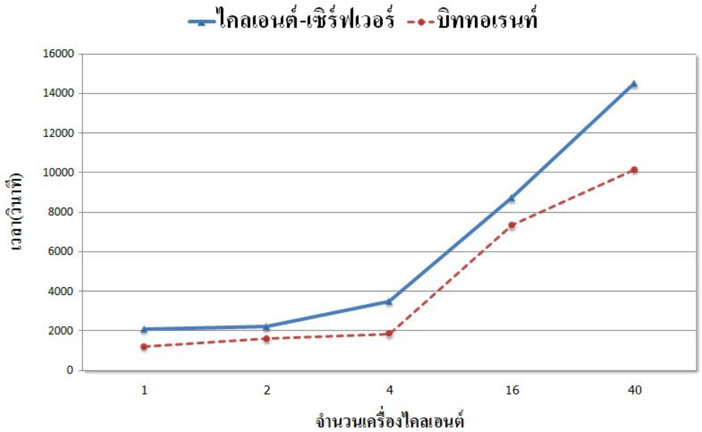
ภาพประกอบ 4-16 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 16 เครื่อง

- จำนวนเครื่องไคลเอนต์ดาวน์โหลด 40 เครื่อง
  -ไคลเอนต์-เซิร์ฟเวอร์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 14496 วินาที
  -บิททอเรนท์ ดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ใช้เวลา 9664 วินาที



ภาพประกอบ 4-17 การเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte จำนวนเครื่องดาวน์โหลด 40 เครื่อง

อย่างไรก็ตามเมื่อเปรียบเทียบเวลาการดาวน์โหลดทั้งสองระบบดังแสดงในภาพประกอบ 4-13 ถึง 4-17 จำนวนเครื่องไคลเอนต์ในการดาวน์โหลด 1, 2 และ 4 เครื่อง ใช้เวลาดาวน์โหลดใกล้เคียงกันทั้งสองระบบ รวมถึงภาระงานที่เครื่องเซิร์ฟเวอร์มีจำนวนการดาวน์โหลดปริมาณที่ใกล้เคียงกัน แต่เมื่อจำนวนเครื่องไคลเอนต์ในการดาวน์โหลดที่ 16 และ 40 เครื่อง จะพบว่าบิททอเรนท์ มีความเร็วด้านเวลาในการดาวน์โหลดข้อมูลเสร็จสมบูรณ์และภาระงานที่เครื่องเซิร์ฟเวอร์ใช้เวลาน้อยกว่าไคลเอนต์-เซิร์ฟเวอร์ อย่างมีนัยสำคัญ ดังแสดงในภาพประกอบ 4-18



ภาพประกอบ 4-18 การเปรียบเทียบเวลาในการดาวน์โหลดข้อมูลระหว่างระบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ ไฟล์ข้อมูลขนาด 4330 MByte

ตาราง 4-3 ตารางเปรียบเทียบประสิทธิภาพด้านเวลา ไฟล์ข้อมูลขนาด 4330 MByte

จำนวนเครื่องไคลเอนต์	ไคลเอนต์-เซิร์ฟเวอร์ (เวลาดาวน์โหลดข้อมูล) (วินาที)	บิททอเรนท์ (เวลาดาวน์โหลดข้อมูล) (วินาที)	เปรียบเทียบประสิทธิภาพด้านเวลา (คิดเป็นเปอร์เซ็นต์)
1	2100	1220	50
2	2232	1606	35
4	3504	1849	50
16	8736	7280	17
40	14496	10171	33

4.1.3 สรุปผลการทดลอง

หัวข้อนี้ได้นำเสนอกลไกทำงานเพื่อลดภาระความคับคั่งในการกระจายไฟล์ข้อมูลจากเว็บไซต์อีเลิร์นนิ่งมูเดิล ด้วยแนวทางการสื่อสารแบบเพียร์ทูเพียร์โดยการเพิ่มโมดูลการกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์ ด้วยโพรโทคอลบิททอเรนท์

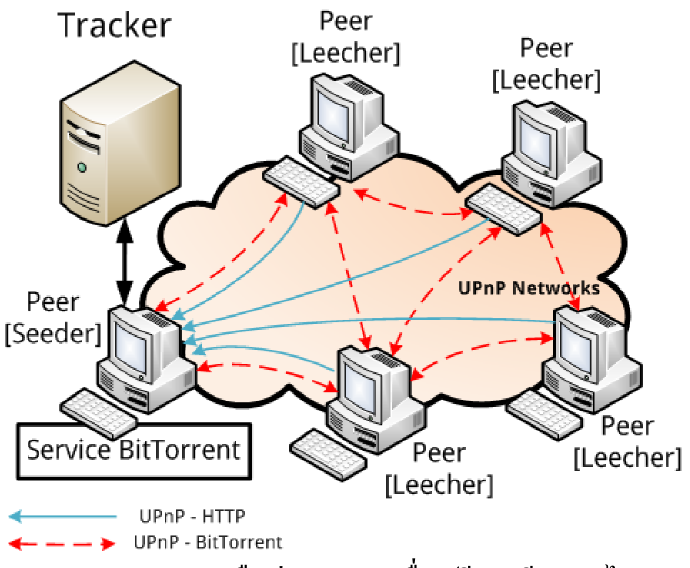
จากการทดลองดาวน์โหลดไฟล์ข้อมูลเพื่อแสดงให้เห็นประสิทธิภาพด้านเวลาการดาวน์โหลดข้อมูลแบบไคลเอนต์-เซิร์ฟเวอร์ กับบิททอเรนท์ โดยแบ่งไฟล์ข้อมูลออกเป็น 3 ขนาด คือ 50 MByte, 600 MByte และ 4330 MByte ทำการดาวน์โหลดไฟล์ข้อมูลโดยมีจำนวนเครื่องไคลเอนต์ แบ่งเป็น 5 กลุ่ม คือ 1, 2, 4, 16 และ 40 เครื่อง ตามลำดับ เมื่อเปรียบเทียบการดาวน์โหลดทั้งสองรูปแบบพบว่า ขนาดของไฟล์ข้อมูลและจำนวนเครื่องไคลเอนต์มีผลต่อการดาวน์โหลดไฟล์ข้อมูลเสร็จสมบูรณ์ ดังที่กล่าวมาแล้วในหัวข้อผลการทดลองข้างต้น หากจำนวนเครื่องไคลเอนต์ในการดาวน์โหลดมีจำนวน 1 และ 2 เครื่อง จำนวนเวลาดาวน์โหลดไฟล์ข้อมูลใกล้เคียงกัน แต่เมื่อจำนวนเครื่องไคลเอนต์เพิ่มขึ้นเป็น 4, 16 และ 40 เครื่อง ความเร็วในการดาวน์โหลดแบบบิททอเรนท์ มีความเร็วด้านเวลาในการดาวน์โหลดไฟล์ข้อมูลสูงถึง 33 % ในกรณีไฟล์ข้อมูลขนาด 4330 MByte และจำนวนเครื่องไคลเอนต์ 40 เครื่อง เป็นต้น ดังนั้นสรุปได้ว่าการนำเทคนิคการกระจายไฟล์ข้อมูลแบบบิททอเรนท์มาใช้งานในอีเลิร์นนิ่งมูเดิลสามารถลดภาระความคับคั่งแบบคอขวดและมีประสิทธิภาพด้านเวลาในการกระจายไฟล์ข้อมูลได้เป็นอย่างดี

4.2 ผลการทดลอง กลไกบริการ OSGi เพื่อกระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP

4.2.1 สภาพแวดล้อมในการทดลอง

กำหนดจำนวนเครื่องคอมพิวเตอร์ทั้งหมด 40 เครื่อง (ทำหน้าเป็นเครื่องไคลเอนต์ในที่นี้เรียกว่า เพียร์) โดยแบ่งกลุ่มการทดลอง ออกเป็น 5 กลุ่ม คือ 1, 2, 4, 16 และ 40 เครื่อง เพื่อดาวน์โหลดไฟล์ข้อมูลขนาด 600 MByte ซึ่งการกระจายไฟล์ข้อมูลนี้ ใช้กลไกกระจายไฟล์ข้อมูลแบบบิททอเรนท์ โดยทำการแบ่งไฟล์ออกเป็นส่วนๆไว้ที่ผู้ส่ง (Seeder) จากนั้นผู้รับ (Leecher) ในเครือข่ายจะช่วยกันทำการดาวน์โหลดชิ้นส่วนของไฟล์จาก Seeder และขณะเดียวกัน Leecher ก็จะส่งต่อชิ้นส่วนที่รับมาจาก Seeder ไปยังเพียร์ตัวอื่นๆอีกทอดหนึ่ง จนทุกๆเพียร์ในเครือข่ายมีชิ้นส่วนของไฟล์ครบสมบูรณ์

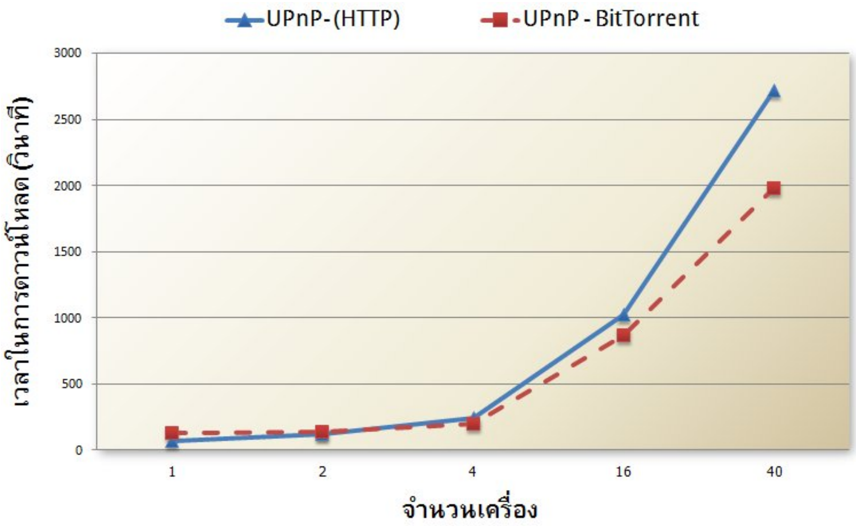
คอมพิวเตอร์ที่ใช้ร่วมในการทดลองมีคุณสมบัติดังนี้ Pentium(R) D CPU 3.40 GHz, RAM 512 MB และใช้ระบบปฏิบัติการ Windows XP เชื่อมต่อเข้าด้วยกันเป็นเครือข่าย LAN ที่อัตราความเร็ว 100 Mbps (ดูภาพประกอบที่ 4-19)



ภาพประกอบ 4-19 ระบบทดลองของเครือข่าย UPnP เพื่อเปรียบเทียบกลไกกระจายไฟล์ข้อมูลด้วยโพรโทคอล HTTP และบิททอเรนท์

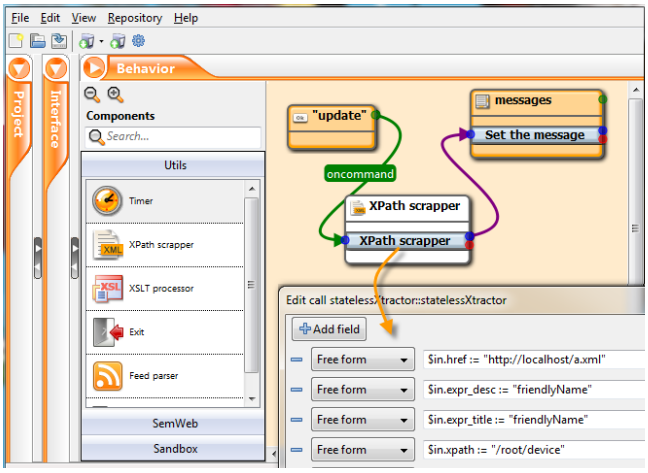
4.2.2 ผลการทดลอง

ในภาพประกอบ 4-20 เป็นผลลัพธ์ที่ได้จากการทดลอง โดยจะเห็นว่า ถ้าเครื่องที่ใช้มีจำนวนไม่มากเพียง 1 - 2 เครื่อง ระยะเวลาที่ใช้ในการดาวน์โหลดโดยใช้โพรโทคอลทั้งสองแบบนั้นจะใกล้เคียงกัน แต่เมื่อจำนวนเครื่องในการดาวน์โหลดมากขึ้นเป็น 4, 16 และ 40 เครื่อง ความเร็วในการดาวน์โหลดโดยใช้โพรโทคอลบิททอเรนท์จะมีความเร็วด้านเวลาในการกระจายไฟล์ข้อมูลเพิ่มขึ้นเป็นลำดับ ในกรณีเครื่องไคลเอนต์จำนวน 40 เครื่อง สามารถลดเวลาในการดาวน์โหลดไฟล์ข้อมูลได้ถึง 27 เปอร์เซ็นต์



ภาพประกอบ 4-20 กราฟเปรียบเทียบเวลาดาวน์โหลดไฟล์ข้อมูล แบบพื้นฐาน (HTTP) กับแบบบิททอเรนท์ (UPnP-BitTorrent)

ในภาพประกอบ 4-21 แสดงให้เห็นกลไกทำงานในเว็บแอพพลิเคชันแสดงบริการบิททอเรนท์ที่ออกแบบขึ้น ด้วยเครื่องมือ Open Mashups ในที่นี้ได้ออกแบบให้มีปุ่มกดเพื่ออัพเดทบริการ (แสดงด้วยกล่องชื่อ Update) ผ่าน XPath scapper เป็นตัวอ่านไฟล์ Extensible Markup Language (XML) (ซึ่งเป็นไฟล์ที่เก็บค่าบริการต่างๆ) ซึ่งทำให้ผู้ใช้เองจะสามารถระบุกำหนดเงื่อนไข หรือบริการที่ต้องการได้ ในที่นี้กำหนดให้แสดงเฉพาะบริการบิททอเรนท์ ผลลัพธ์จากกล่องนี้ จะถูกส่งผ่านไปยังกล่องตัวแสดงผล (Message) ผลของรันเว็บแอพพลิเคชันนี้แสดงให้เห็นตัวอย่างดังภาพประกอบ 4-22 ซึ่งหากต้องการใช้บริการ เพียงนำเมาส์เลือกที่บริการนั้นๆ ก็จะดาวน์โหลดงานได้ทันที



ภาพประกอบ 4-21 ตัวอย่างเว็บแอพพลิชันแบบ Mashups ใช้บริการดาวน์โหลดไฟล์ด้วยบิททอเรนท์ ด้วยเครื่องมือ Open Mashups

ภาพประกอบ 4-22 ตัวอย่างแสดงการใช้เว็บแอพพลิเคชันที่พัฒนาขึ้น

4.2.3 สรุปผลการทดลอง

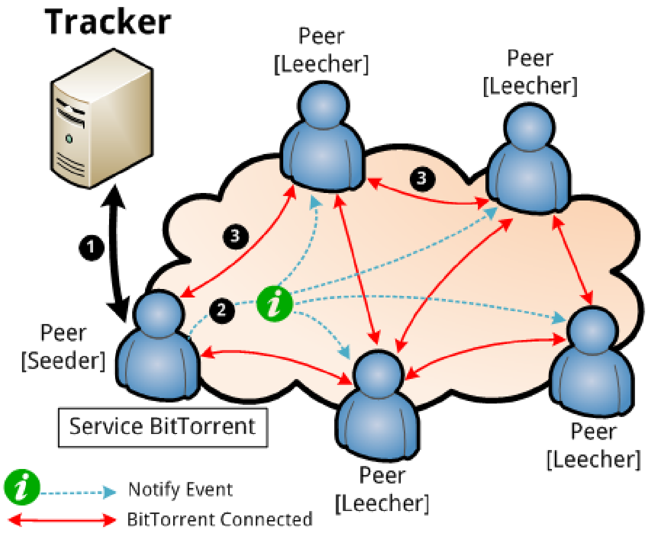
ผลการทดลองระบบต้นแบบในหัวข้อนี้ ได้เปรียบเทียบเวลาดาวน์โหลดไฟล์ข้อมูลแบบพื้นฐาน กับแบบบิททอเรนท์ ภายในเครือข่าย UPnP จากการทดลองพบว่าความเร็วด้านเวลาในการกระจายไฟล์ข้อมูล ในกรณีเครื่องไคลเอนต์จำนวน 40 เครื่อง สามารถลดเวลาในการดาวน์โหลดไฟล์ข้อมูลได้ถึง 27 เปอร์เซ็นต์ หากเพิ่มจำนวนเครื่องไคลเอนต์มากขึ้น มีแนวโน้มที่สามารถลดเวลาการกระจายไฟล์ข้อมูลได้อีกเป็นจำนวนมาก ดังนั้นแสดงให้เห็นและเป็นสิ่งที่ยืนยันในเบื้องต้นได้ว่า การนำโพรโทคอลบิททอเรนท์ มาช่วยการกระจายไฟล์ข้อมูลอีเลิร์นนิ่ง

ภายในเครือข่าย UPnP นั้น มีประโยชน์ในแง่ของการลดภาระงานของเครื่องเซิร์ฟเวอร์ได้เป็นอย่างดี และมีประสิทธิภาพด้านเวลา เมื่อเปรียบเทียบกับการกระจายไฟล์ข้อมูลแบบเดิม ในการทดลองมีการนำข้อดีของเทคโนโลยี Mashups มาใช้ทำให้ผู้ใช้สะดวกต่อการออกแบบสร้างอินเตอร์เฟสในการใช้ประโยชน์จากงานที่นำเสนอได้มากยิ่งขึ้น

4.3 ผลการทดลอง กลไกบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP

4.3.1 สภาพแวดล้อมในการทดลอง

กำหนดจำนวนเครื่องคอมพิวเตอร์ทั้งหมด 40 เครื่อง (ทำหน้าเป็นเครื่องไคลเอนต์ในที่เรียกว่า เพียร์) โดยแบ่งกลุ่มการทดลอง ออกเป็น 5 กลุ่ม คือ 1, 2, 4, 16 และ 40 เครื่อง เชื่อมต่อเข้าด้วยกันเป็นเครือข่าย LAN ที่อัตราความเร็ว 100 Mbps คอมพิวเตอร์ที่ใช้ในการทดลองมีคุณสมบัติดังนี้ Pentium(R) D CPU 3.40 GHz, RAM 512 MB และใช้ระบบปฏิบัติการ Windows XP (ดูภาพประกอบ 4-23)



ภาพประกอบ 4-23 ระบบทดลองบิททอเรนท์ภายในเครือข่าย UPnP

จากภาพประกอบ 4-23 สามารถอธิบายลำดับการทดลองของระบบต้นแบบได้ดังนี้

1) Seeder สร้างไฟล์ทอเรนท์ และอัพเดทไฟล์ทอเรนท์ไปยัง Tracker

2) Seeder แจ้งเตือนบริการไปยังเครื่องไคลเอนต์

3) กระจายไฟล์ข้อมูลด้วยกลไกบิททอเรนท์

การทดลองนี้ เป็นการเปรียบเทียบโมเดลการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการ 2 รูปแบบคือ

1) Push Service กำหนดให้มีการเปลี่ยนแปลงบริการจากเครื่องเซิร์ฟเวอร์โดยใช้วิธีการสุ่มเวลา (Random Time) เป็น 3 ช่วงเวลา คือช่วงที่ 1 (1-30 วินาที), ช่วงที่ 2 (31-60 วินาที) และช่วงที่ 3 (61-120 วินาที)
2) Pull Service กำหนดให้มีการร้องขอบริการจากเครื่องเซิร์ฟเวอร์ โดยใช้วิธีการสุ่มเวลา เป็น 3 ช่วงเวลา คือช่วงที่ 1 (1-30 วินาที), ช่วงที่ 2 (31-60 วินาที) และช่วงที่ 3 (61-120 วินาที)

4.3.2 ผลการทดลอง

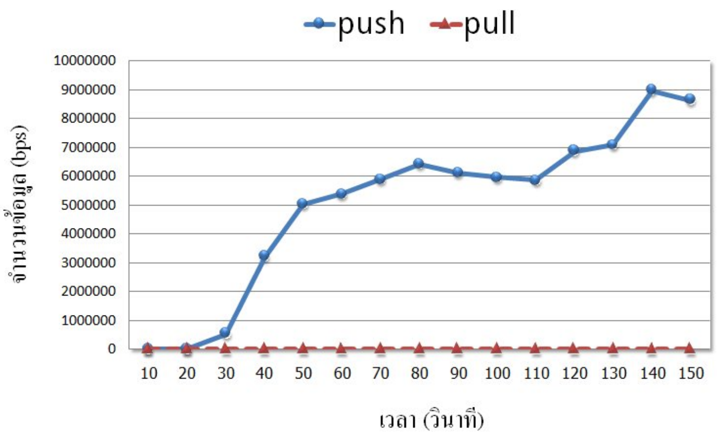
ทำการทดลองจำนวน 3 ครั้ง โดยสุ่มเวลา 3 ช่วงเวลาของการเปลี่ยนแปลงบริการทั้ง 2 แบบ ดังตาราง 4-4

ตาราง 4-4 ตารางเปรียบเทียบการสุ่มเวลา 3 ช่วง ของการเปลี่ยนแปลงบริการแบบ Push Service และ Pull Service

จำนวนครั้ง	Push Service			Pull Service		
	สุ่มเวลา ช่วงที่ 1 (วินาที)	สุ่มเวลา ช่วงที่ 2 (วินาที)	สุ่มเวลา ช่วงที่ 3 (วินาที)	สุ่มเวลา ช่วงที่ 1 (วินาที)	สุ่มเวลา ช่วงที่ 2 (วินาที)	สุ่มเวลา ช่วงที่ 3 (วินาที)
1	30	55	87	24	50	64
2	21	39	95	30	60	105
3	11	52	97	15	31	95

- จำนวนเครื่องไคลเอนต์ ในการแจ้งเตือนเหตุการณ์และดาวน์โหลดข้อมูล 1 เครื่อง
    - ครั้งที่ 1 สุ่มเวลาช่วงที่ 1 (Push Service: 30 วินาที, Pull Service: 24 วินาที)

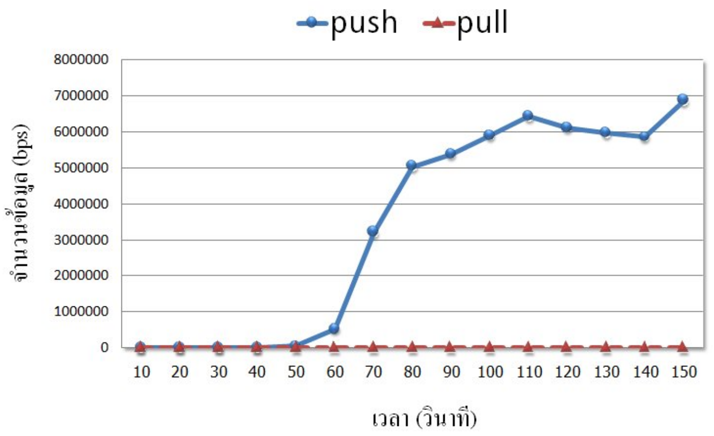
      -โมเดลแบบ Pull Service ไม่ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจากร้องขอข้อมูลก่อนเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



ภาพประกอบ 4-24 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 30 วินาที, Pull Service: 24 วินาที

- ครั้งที่ 1 สุ่มเวลาช่วงที่ 2 (Push Service: 55 วินาที, Pull Service: 50 วินาที)

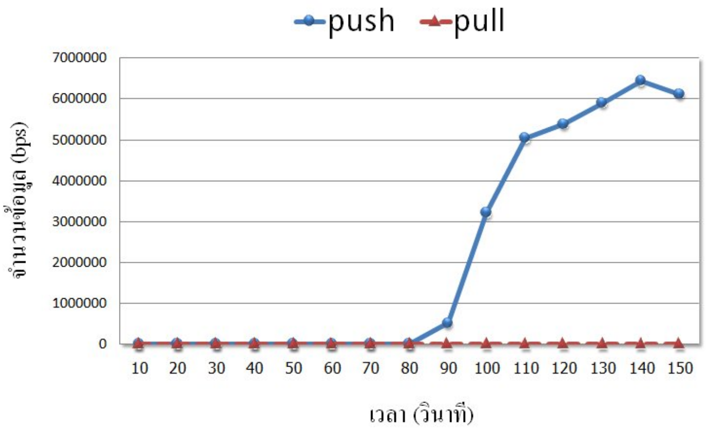
  -โมเดลแบบ Pull Service ไม่ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจากร้องขอข้อมูลก่อนเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



ภาพประกอบ 4-25 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 55 วินาที, Pull Service: 50 วินาที

- ครั้งที่ 1 สุ่มเวลาช่วงที่ 3 (Push Service: 87 วินาที, Pull Service: 64 วินาที)

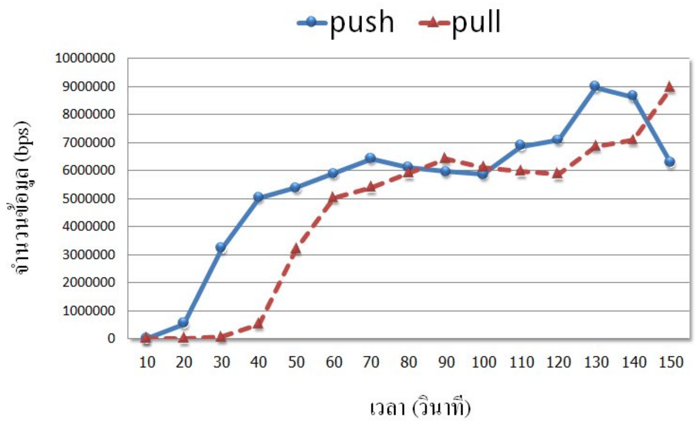
  -โมเดลแบบ Pull Service ไม่ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจากร้องขอข้อมูลก่อนเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



ภาพประกอบ 4-26 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 87 วินาที, Pull Service: 64 วินาที

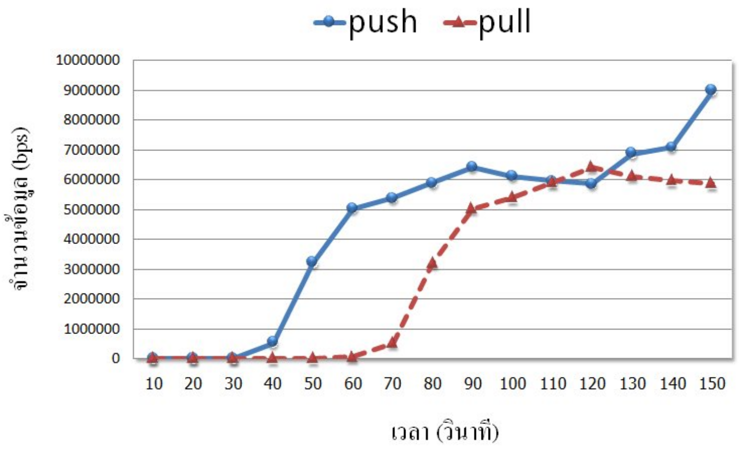
- ครั้งที่ 2 สุ่มเวลาช่วงที่ 1 (Push Service: 21 วินาที, Pull Service: 30 วินาที)

  -โมเดลแบบ Pull Service ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจากร้องขอข้อมูลหลังเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



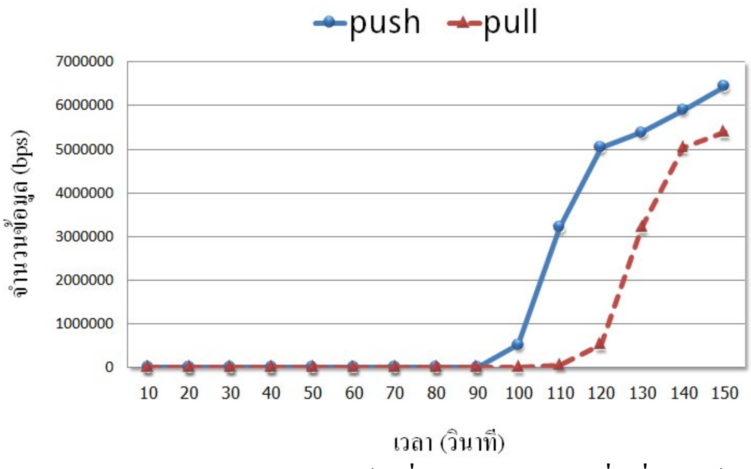
ภาพประกอบ 4-27 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 21 วินาที, Pull Service: 30 วินาที

- ครั้งที่ 2 สุ่มเวลาช่วงที่ 2 (Push Service: 39 วินาที, Pull Service: 60 วินาที)
  -โมเดลแบบ Pull Service ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจาก
  ร้องขอข้อมูลหลังเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



ภาพประกอบ 4-28 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 39 วินาที, Pull Service: 60 วินาที

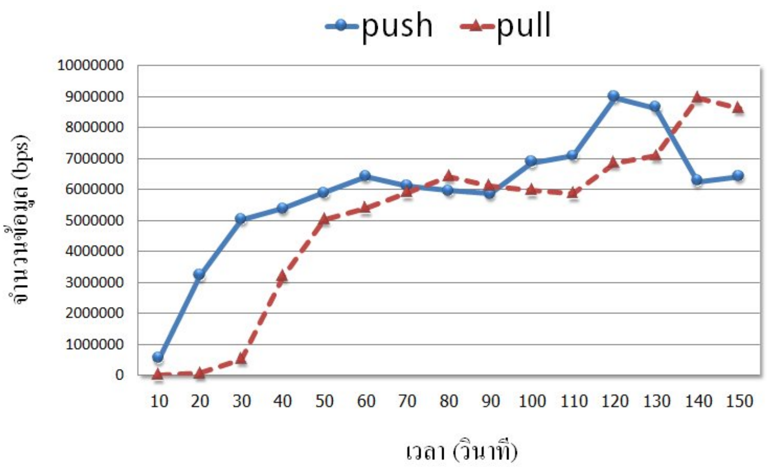
- ครั้งที่ 2 สุ่มเวลาช่วงที่ 3 (Push Service: 95 วินาที, Pull Service: 105 วินาที)
  -โมเดลแบบ Pull Service ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจาก
  ร้องขอข้อมูลหลังเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



ภาพประกอบ 4-29 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 95 วินาที, Pull Service: 105 วินาที

- ครั้งที่ 3 สุ่มเวลาช่วงที่ 1 (Push Service: 11 วินาที, Pull Service: 15 วินาที)

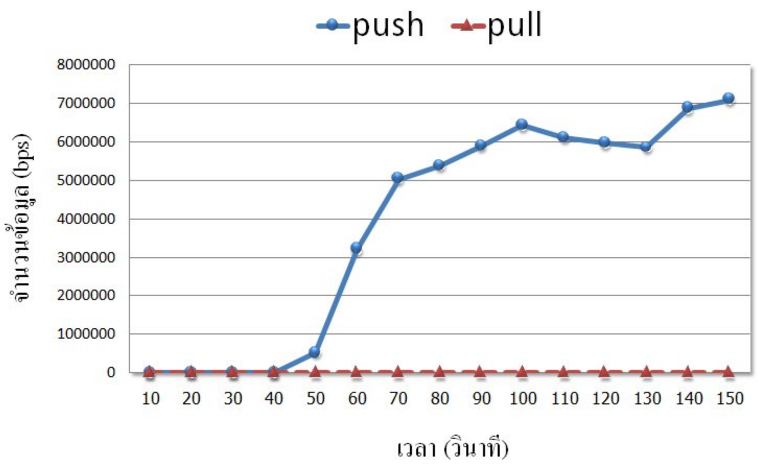
  -โมเดลแบบ Pull Service ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจากร้องขอข้อมูลหลังเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



ภาพประกอบ 4-30 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 11 วินาที, Pull Service: 15 วินาที

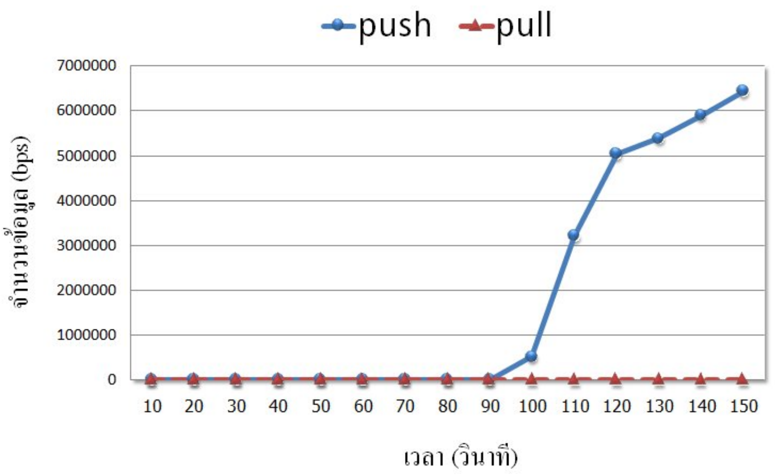
- ครั้งที่ 3 สุ่มเวลาช่วงที่ 2 (Push Service: 52 วินาที, Pull Service: 31 วินาที)

  -โมเดลแบบ Pull Service ไม่ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจากร้องขอข้อมูลก่อนเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



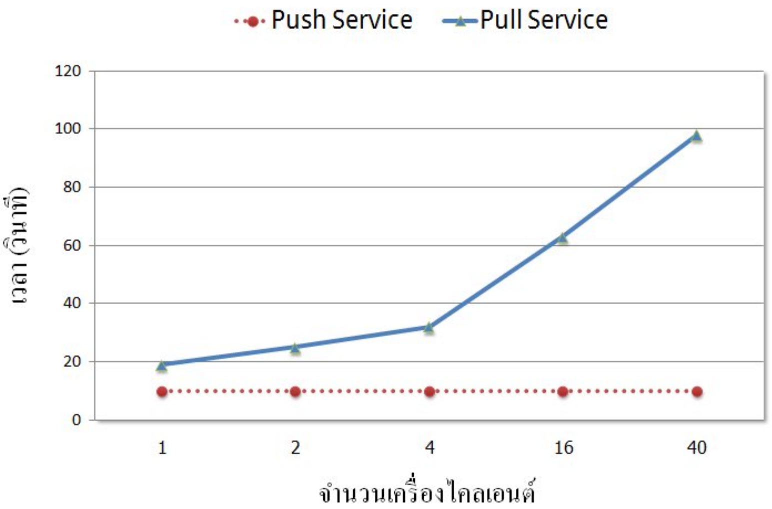
ภาพประกอบ 4-31 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 52 วินาที, Pull Service: 31 วินาที

- ครั้งที่ 3 สุ่มเวลาช่วงที่ 3 (Push Service: 97 วินาที, Pull Service: 95 วินาที
  -โมเดลแบบ Pull Service ไม่ได้รับรู้บริการดาวน์โหลดไฟล์ข้อมูล เนื่องจากร้องขอข้อมูลก่อนเวลาที่เครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ



ภาพประกอบ 4-32 เปรียบเทียบการรับรู้เหตุการณ์เปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อ Push Service: 97 วินาที, Pull Service: 95 วินาที

อย่างไรก็ตามการเปรียบเทียบกลไกการแจ้งเตือนบริการโมเดลแบบ Push service และ Pull service ดังแสดงในภาพประกอบ 4-24 ถึง 4-32 นั้น แสดงให้เห็นถึงช่วงเวลาของรับรู้เหตุการณ์ของแต่ละโมเดล โดยที่ Push service จะรับรู้เหตุการณ์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อเครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการโดยทันที แต่ Pull service ต้องมีการร้องขอบริการไปยังเครื่องเซิร์ฟเวอร์ก่อนเสมอ หากมีการร้องขอไปยังเครื่องเซิร์ฟเวอร์ก่อนเวลาที่จะมีการเปลี่ยนแปลงบริการ Pull service ก็ไม่สามารถรับรู้เหตุการณ์และดาวน์โหลดข้อมูลได้ทันท่วงที ดังนั้นจึงมีความแตกต่างในเรื่องของเวลาในการแจ้งเตือนบริการอย่างเด่นชัด หากจำนวนเครื่องไคลเอนต์ในการแจ้งเตือนเหตุการณ์และดาวน์โหลดข้อมูลที่ 4, 16 และ 40 ดังแสดงในภาพประกอบ 4-33



ภาพประกอบ 4-33 เปรียบเทียบเวลาการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการระหว่าง Push Service และ Pull Service

4.3.3 สรุปผลการทดลอง

ผลการทดลองตามแนวคิดที่นำเสนอในหัวข้อนี้ ได้ทำการทดลองเปรียบเทียบกลไกการแจ้งเตือนบริการโมเดลแบบ Push service และ Pull service พบว่า Push service จะรับรู้เหตุการณ์และทำการดาวน์โหลดไฟล์ข้อมูลเมื่อเครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการโดยทันที ทำให้สามารถเพิ่มประสิทธิภาพโดยรวมของระบบการทำงาน โดยมีกลไกแจ้งเตือนเหตุการณ์ประยุกต์ใช้งาน ซึ่งมีประโยชน์ในแง่ของการลดระยะเวลาในการค้นหาบริการและตรวจสอบบริการใดๆ บนเครื่องเซิร์ฟเวอร์ได้เป็นอย่างดี

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

บทนี้จะกล่าวถึงบทสรุปของงานวิจัยเพื่อวิทยานิพนธ์นี้ ซึ่งทำการศึกษากลไกทำงานเพื่อลดภาระความคับคั่งแบบคอขวดในการกระจายไฟล์ข้อมูลจากเว็บไซต์อีเลิร์นนิ่งที่ใช้ลักษณะสถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ ด้วยแนวทางการสื่อสารแบบเพียร์ทูเพียร์ ซึ่งงานวิจัยนี้ได้ศึกษาและพัฒนาระบบต้นแบบเพื่อนำกลไกการทำงานของโพรโทคอลบิททอเรนท์มาช่วยในการกระจายไฟล์ข้อมูล จากการทดลองแนวคิดที่ผ่านมา สามารถสรุปเป็นรายงานเชิงประสิทธิภาพด้านเวลาในการกระจายไฟล์ข้อมูลอีเลิร์นนิ่งแบบเพียร์ทูเพียร์ด้วยโพรโทคอลบิททอเรนท์ เมื่อเปรียบเทียบกับกลไกกระจายไฟล์ข้อมูลแบบพื้นฐานด้วยโพรโทคอล HTTP นั้นมีแนวโน้มด้านความเร็วในการกระจายไฟล์ข้อมูลที่ดีขึ้นอย่างมีนัยสำคัญ ดังนั้นปัญหาภาระความคับคั่งแบบคอขวดที่เครื่องเซิร์ฟเวอร์อีเลิร์นนิ่งมูเดิล สามารถพัฒนาแก้ไขได้ด้วยวิธีที่นำเสนอในงานวิจัยเพื่อวิทยานิพนธ์นี้

5.1 สิ่งที่ได้นำเสนอในวิทยานิพนธ์

5.1.1 กลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล

เป็นกลไกการทำงานเพื่อลดภาระความคับคั่งแบบคอขวดที่เครื่องเซิร์ฟเวอร์อีเลิร์นนิ่งมูเดิล ซึ่งนำโพรโทคอลบิททอเรนท์มาช่วยในการกระจายไฟล์ข้อมูล โดยพัฒนาโมดูลการกระจายไฟล์ที่อีเลิร์นนิ่งมูเดิลให้สามารถสื่อสารแบบเพียร์ทูเพียร์ได้ โดยแนวทางนี้เป็นการแก้ปัญหาข้อจำกัดของสถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ ที่เป็นกลไกการทำงานของอีเลิร์นนิ่งมูเดิลดังกล่าว โดยเนื้อหาส่วนนี้ได้ถูกนำเสนอเป็นบทความชื่อ “การกระจายไฟล์ข้อมูลแบบ Peer to Peer สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล” การประชุมวิชาการทางวิศวกรรมศาสตร์มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 6, อ.หาดใหญ่ จ.สงขลา, ประเทศไทย, 8-9 พฤษภาคม 2551.

5.1.2 กลไกบริการ OSGi เพื่อกระจายไฟล์ข้อมูลแบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP

เป็นงานวิจัยต่อเนื่องจากกลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล ซึ่งงานวิจัยนี้เป็นการประยุกต์ใช้งานด้านอีเลิร์นนิ่งมูเดิล ภายในเครือข่าย UPnP โดยพัฒนาอุปกรณ์ที่ทำหน้าที่เป็นเอเจนท์ให้สามารถกระจายไฟล์ข้อมูลไปยังอุปกรณ์ตัวอื่นๆ ภายในเครือข่าย ซึ่งออกแบบแอพพลิเคชันการบริการให้สามารถดาวน์โหลดไฟล์ข้อมูลด้วยโพรโทคอลบิททอเรนท์ ที่ทำงานในสิ่งแวดล้อมของซอฟต์แวร์เชิงบริการตามมาตรฐาน OSGi โดยเนื้อหาส่วนนี้ได้ถูกนำเสนอเป็นบทความชื่อ "การออกแบบกลไกบริการ OSGi เพื่อกระจายไฟล์แบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP" การประชุมวิชาการวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต ครั้งที่ 8, กรุงเทพมหานคร, ประเทศไทย, 19 มีนาคม 2553.

5.1.3 กลไกบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP

งานวิจัยนี้เป็นการประยุกต์เพื่อพัฒนากลไกแจ้งเตือนบริการแบบพุชด้วยโพรโทคอลบิททอเรนท์ในเครือข่าย UPnP โดยใช้คุณลักษณะเด่นของโพรโทคอลภายในเครือข่าย UPnP เช่น GENA เป็นต้น เพื่อช่วยให้ผู้ใช้สามารถรับรู้เหตุการณ์ หากมีการเปลี่ยนแปลงข้อมูลเพื่อช่วยลดปัญหาเรื่องเวลาในการเข้าใช้บริการโดยรวม และลดภาระงานที่เครื่องเซิร์ฟเวอร์ โดยเนื้อหาส่วนนี้ได้ถูกนำเสนอเป็นบทความชื่อ "การออกแบบบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP" การประชุมวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีครั้งที่ 6 (NCCIT 2010), กรุงเทพมหานคร, ประเทศไทย, 3-5 มิถุนายน 2553.

5.2 ข้อเสนอแนะ

- ปรับเปลี่ยนการกระจายไฟล์ข้อมูลแบบสภาพแวดล้อมจริง เป็นสภาพแวดล้อมแบบจำลอง เพื่อสามารถเพิ่ม/ลดจำนวนเครื่องคอมพิวเตอร์ในการทดลอง และสามารถกำหนดจำนวนเครื่องคอมพิวเตอร์ที่เหมาะสมสำหรับการกระจายไฟล์ข้อมูลได้
- ควรมีการพัฒนารูปแบบของการกระจายไฟล์เอกสาร หรือไฟล์มัลติมีเดีย ที่เกี่ยวกับอีเลิร์นนิ่งให้สามารถกระจายไฟล์แบบสตรีมมิ่งได้

เอกสารอ้างอิง

[1] Moodle, [online]. Available: http://www.moodle.org

[2] S. Androutsellis-Theotokis and D.Spinellis, “A Survey of Peer-to-Peer Content Distribution Technologies,” *ACM Computing Surveys*, Vol.36, No.4, December 2004, pp.335–371.

[3] BitTorrent, [online]. Available: http://www.bittorrent.org

[4] L. Scully, “Network File Distribution with the BitTorrent Protocol,” *Proceeding of the 7th Winona Computer Science Undergraduate Research Seminar*, April 19, 2007, Winona, MN, US.

[5] UPnP Forum, 2009, UPnP Standards, Retrieved December 15, 2009, [online]. Available: http://www.upnp.org/

[6] Single point of failure, [online]. Available: http://en.wikipedia.org/wiki/Single_point_of_failure

[7] Bottleneck, [online]. Available: http://en.wikipedia.org/wiki/Bottleneck

[8] Qu, C., and W. Nejdl, “ Searching SCORM Metadata in a RDF-based E-Learning P2P Network Using XQuery and Query by Example,” *in Proc. of the 3rd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2003)*, IEEE Computer Society Press, July 2003, Athens, Greece, pp.81-85.

[9] ศานติ ชิณการณ์ และประดนเดช นีละคุปต์, “การวิเคราะห์ความนิยมของข้อมูลกับประสิทธิ์ภาพของเครือข่ายแจกจ่ายข้อมูลแบบ Peer-to-Peer “กรณีศึกษา BitTorrent”,” *NECSEC'2005: The first Northeastern Computer Science and Engineering Conference*, 3 2005.

[10] M. Meulpolder and V.A. Pijano and D.H.J. Epema and H.J. Sips (2007). TriblerCampus: Integrated peer-to-peer file distribution in course management systems. In C. Montgomerie and J. Seale (Eds.). *ED-MEDIA 2007*, pp. 528-536, Vancouver, Canada. AACE.

[11] Self organizing system, [online]. Available: http://javaboom.wordpress.com/2008/03/26/piratep2p/

[12] A. Oram et al., *Peer-to-Peer:Hardnessing the Power of Disruptive Technologies*, Oreilly, 2001

[13] พิทักษ์ เศวตสุนทร และ สุนทร วิทูสุรพจน์, “การออกแบบและพัฒนาเอเจนต์บริการข้อมูลเว็บอีลิร์นนิ่งให้กับอุปกรณ์สื่อสารไร้สายในเครือข่าย UPnP”, *การประชุมวิชาการเสนอผลงานวิจัยระดับบัณฑิตศึกษา ครั้งที่ 11 มหาวิทยาลัยขอนแก่น*, 2553

[14] กิติศักดิ์ วัฒนกุล, สุนทร วิทูสุรพจน์ และ วรรณรัช สันติอมรทัต. 2551, “การกระจายไฟล์ข้อมูลแบบ Peer to Peer สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล,” *การประชุมวิชาการทางวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 6*, อ.หาดใหญ่ จ.สงขลา, ประเทศไทย, 8-9 พฤษภาคม 2551.

[15] OSGi -The Dynamic Module System for Java, [online]. Available: http://www.osgi.org

[16] Bundle, [online]. Available: http://www.osgi.org/javadoc/r4v42/org/osgi/framework/Bundle.html

[17] Physical bundle life-cycle, [online]. Available: http://gravity.sourceforge.net/servicebinder/osginutshell.html

[18] M. Ujjal and J. Shibsankar, “Mashups: An Emerging Content Aggregation Web 2.0 Paradigm,” *7^{th} internation CALIBER-2009*, Pondicherry University, Puducherry, February 25-27, 2009, pp.296-303.

[19] Open Mashups, [online]. Available: http://www.open-mashups.org

[20] Z. Zhang, Y. Lin, Y. Chen, Y. Xiong, J. Shen, H. Liu, B. Deng and X. Li, "Experimental Study of Broadcatching in BitTorrent," *Proc. of IEEE Consumer Communications and Networking Conference (CCNC'09)*, Las Vegas, Nevada, Jan. 2009.

[21] The application/rss + xml, Media Type: http://www.rssboard.org/rss- mime-type-application.txt Network Working Group, May 22, 2006.

[22] Pull Technology, [online]. Available: http://en.wikipedia.org/wiki/Pull_technology

[23] RSS Reader, [online]. Available: http://www.rssreader.com/

[24] Push Technology, [online]. Available: http://en.wikipedia.org/wiki/Push_technology

[25] µTorrent, [online]. Available: http://www.utorrent.com

[26] Onlineeye pro, [online]. Available: http://www.pmasoft.net

ภาคผนวก

ภาคผนวก ก : ผลงานที่ได้รับการตีพิมพ์เผยแพร่

1. กิติศักดิ์ วัฒนกุล, สุนทร วิทูสุรพจน์ และ วรรณรัช สันติอมรทัต. 2551, "การกระจายไฟล์ข้อมูลแบบ Peer to Peer สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล," *การประชุมวิชาการทางวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 6*, อ.หาดใหญ่ จ.สงขลา, ประเทศไทย, 8-9 พฤษภาคม 2551.
2. กิติศักดิ์ วัฒนกุล และสุนทร วิทูสุรพจน์. 2553, "การออกแบบกลไกบริการ OSGi เพื่อกระจายไฟล์แบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP," *การประชุมวิชาการวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต ครั้งที่ 8*, กรุงเทพมหานคร, ประเทศไทย, 19 มีนาคม 2553.
3. กิติศักดิ์ วัฒนกุล และสุนทร วิทูสุรพจน์. 2553, "การออกแบบบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP," *การประชุมวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยี ครั้งที่ 6 (NCCIT 2010)*, กรุงเทพมหานคร, ประเทศไทย, 3-5 มิถุนายน 2553.

การประชุมวิชาการทางวิศวกรรมศาสตร์มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 6
8-9 พฤษภาคม 2551

การกระจายไฟล์ข้อมูลแบบ Peer to Peer สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิ้ล

Peer-to-Peer File Distribution for Moodle E-Learning

กิติศักดิ์ วัฒนกุล สุนทร วิทูสุรพจน์ วรรณรัช สันติอมรทัต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ อ.หาดใหญ่ จ.สงขลา 90112
E-mail: vip.pung@gmail.com wsuntorn@coe.psu.ac.th wannarat@coe.psu.ac.th

Kitisak Wattanakul Suntorn Witosurapot Wannarat Suntiamorntut
Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla 90112
E-mail: vip.pung@gmail.com wsuntorn@coe.psu.ac.th wannarat@coe.psu.ac.th

บทคัดย่อ

บทความนี้นำเสนอแนวทางการประยุกต์ใช้เทคนิคการกระจายไฟล์แบบ Peer-to-Peer (P2P) เพื่อนำมาใช้งานในลักษณะโมดูลเสริมสำหรับซอฟต์แวร์โอเพ่นซอร์สอีเลิร์นนิ่งมูเดิ้ล (Moodle) ซึ่งนิยมใช้งานกันในประเทศไทย ทั้งนี้เพื่อเป็นการแก้ไขปัญหาที่เป็นผลจากข้อด้อยเชิงสถาปัตยกรรมไคลเอนต์-เซิร์ฟเวอร์ในการตอบสนองต่อความคับคั่ง ขณะเมื่อมีการเรียกใช้ไฟล์ข้อมูลขนาดใหญ่มาจากผู้ใช้จำนวนมากในเวลาที่ใกล้เคียงกัน เนื่องจากโปรโตคอลที่มีพื้นฐานการทำงานแบบ P2P มีการนำคอมพิวเตอร์หลายเครื่องมาใช้ร่วมกิจกรรมกระจายไฟล์แทนที่จะเป็นเพียงเครื่องเดียวเช่นปกติ ดังนั้นจึงสามารถปรับปรุงประสิทธิภาพด้านเวลา ในการกระจายไฟล์ได้มาก ซึ่งสอดคล้องกับผลการทดลองจากระบบต้นแบบที่พัฒนาขึ้นด้วยโปรโตคอลบิตทอเรนต์ โดยเมื่อเปรียบเทียบกับวิธีการที่ใช้อยู่เดิมแล้ว จะเห็นความแตกต่างด้านเวลาในการส่งไฟล์ข้อมูลได้อย่างมีนัยสำคัญ โดยเฉพาะอย่างยิ่งในกรณีที่มีผู้ใช้งานไฟล์เพิ่มมากขึ้น
คำหลัก อีเลิร์นนิ่ง, มูเดิ้ล, โปรโตคอล P2P, บิตทอเรนต์

Abstract

This paper considers an application of peer-to-peer (P2P) protocol for improving typical approach of file distribution in Moodle – a popular, open source software for e-learning in Thailand. By working on the client-server architecture, it is certain that Moodle would not scale well in response to a large number of requests from users attempting to access the same resource file on the server at nearly the same time. By including the support of P2P protocol, this deficiency can be alleviated since many computers are demanded to work in concert for file distribution, rather than handling by a sole computer as in the normal manner. So, the duration time for file distribution expects to be improved significantly even when the number of users gradually increased. This can be noticed from the experiment results from our first prototype employing the Bittorent protocol to work for Moodle.
Keywords: E-Learning, Moodle, P2P Protocol, Bittorrent

1. บทนำ

ปัจจุบันแนวโน้มการนำรูปแบบของการเรียนการสอนด้วยสื่ออิเล็กทรอนิกส์ (ซึ่งต่อไปจะใช้คำว่า E-Learning) เข้ามาใช้งานมากยิ่งขึ้น เนื่องจากความแพร่หลายของเครือข่ายอินเตอร์เน็ตซึ่งเป็นเครือข่ายสาธารณะที่ได้รับความนิยมมากที่สุด สำหรับประเทศไทยนิยมนำซอฟต์แวร์ประเภทโอเพ่นซอร์สทางด้าน LMS (Learning Management System) เช่น โปรแกรมมูเดิ้ล (Moodle) [1] เข้ามาใช้งานในการสร้างและจัดเก็บไฟล์ข้อมูลอิเล็กทรอนิกส์เพื่อจัดการเรียนการสอน อย่างไรก็ตาม ข้อจำกัดของระบบ E-learning ที่สร้างด้วยโปแกรมเหล่านี้ ล้วนเป็นผลมาจากข้อจำกัดในการทำงานสถาปัตยกรรมไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) เอง ตัวอย่างเช่น เมื่อไฟล์ข้อมูลมีขนาดใหญ่ และจำนวนผู้เข้ามาใช้งานในไฟล์เดียวกันนั้นมีจำนวนมาก ก็จะทำให้เครื่องแม่ข่ายมีประสิทธิภาพที่ลดลงอย่างมีนัยสำคัญ ตามภาระการทำงานที่เพิ่มสูงขึ้น และความคับคั่งของทราฟฟิกข้อมูลในเครือข่าย ในบทความนี้ นำเสนอแนวทางการแก้ปัญหาข้างต้นนี้ด้วยการนำโปรโตคอลแบบ Peer-to-Peer (P2P) เช่น บิตทอเรนต์ (Bittorent) [2] เข้ามาช่วยงาน โดยเล็งเห็นถึงประโยชน์จากความมีประสิทธิภาพในการกระจายไฟล์แบบที่ไม่รวมศูนย์อยู่ที่เครื่องใดเครื่องหนึ่ง จึงคาดว่าน่าจะลดปัญหาจากความคับคั่งแบบคอขวดในการกระจายไฟล์ที่เครื่องแม่ข่ายลงได้

บทความนี้ได้วางแนวทางการนำเสนอดังนี้ ในหัวข้อย่อยที่ 2 กล่าวถึงงานวิจัยที่เกี่ยวข้อง ตามด้วยกลไกทำงานของโปรโตคอลบิตทอเรนต์ในหัวข้อย่อยที่ 3 จากนั้นจึงเป็นแนวทางการเพิ่มเติมส่วนขยายการทำงานที่ใช้โปรโตคอลบิตทอเรนต์เข้าไปในระบบ

มูลเดิ้ลในหัวข้อย่อยที่ 4 และแสดงผลลัพธ์จากการทดลองในหัวข้อย่อยที่ 5 และสรุปบทความในหัวข้อย่อยที่ 6

2. งานวิจัยที่เกี่ยวข้อง

แม้ว่าโปรโตคอล แบบ P2P จะใช้งานกันอย่างแพร่หลายเพื่อแลกเปลี่ยนหรือแบ่งปันไฟล์เอกสารหรือมัลติมีเดียระหว่างเครื่องคอมพิวเตอร์ผู้ใช้ด้วยกัน แต่มีข้อมูลงานวิจัยที่เกี่ยวข้องกับระบบอีเลิร์นนิ่งเผยแพร่ไม่มากนัก ตัวอย่างเช่น

- งานวิจัย [3] ได้นำเทคโนโลยีเครือข่ายแบบ P2P มาใช้เพื่อทำงานรวมกับระบบ E-Learning แต่มุ่งเน้นไปทางด้านการสืบค้นข้อมูลบทเรียน
- งานวิจัย [4] เสนอรายงานการวิเคราะห์ความนิยมของข้อมูลกับประสิทธิ์ภาพของเครือข่ายแจกจ่ายข้อมูลโปรโตคอล ซึ่งสนับสนุนการทำงานแบบ P2P ชื่อว่า Bittorrent พบว่าไม่ว่าความนิยมข้อมูลจะมีค่ามากหรือน้อยเท่าใด โปรโตคอลนี้ก็ยังสามารถขยายการบริการให้กับ Peer ทุกตัวได้เป็นอย่างดีโดยไม่ได้ทำให้เกิดสูญเสียประสิทธิภาพโดยรวมในเครือข่ายไปมากเท่ากับที่พบในระบบไคลเอนต์-เซิร์ฟเวอร์ทั่วไป
- งานวิจัย [5] มีความคล้ายคลึงกับระบบที่พัฒนาในงานวิจัยนี้ โดยใช้โปรโตคอลบิตทอเรนต์ เพื่อทำงานร่วมกับซอฟต์แวร์มูเดิ้ลเช่นกัน แต่ความแตกต่างจากงานในบทความนี้ อยู่ที่การนำหลักการทำงานของเว็บบิตแบบโอเพ่นซอร์สมาใช้ในการกระจายไฟล์ข้อมูล และในอนาคตมุ่งเน้นการพัฒนาให้โปรโตคอลบิตทอเรนต์ทำงานได้อย่างมีประสิทธิภาพมากขึ้น

3. กลไกทำงานของโปรโตคอลบิตทอเรนต์

โปรโตคอลบิตทอเรนต์ที่ได้พิจารณาเลือกใช้ในบทความนี้สำหรับการกระจายไฟล์แบบ P2P นั้น เนื่องจากเป็นโปรโตคอลที่มีการแลกเปลี่ยนไฟล์ภายในกลุ่มเครือข่าย และมีความสามารถในการแชร์ไฟล์ข้อมูลได้เร็วกว่าการแชร์ไฟล์ข้อมูลในรูปแบบอื่นๆ

ส่วนประกอบที่สำคัญของโปรโตคอลบิตทอเรนต์ คือ

1. Tracker Client คือโปรแกรมที่ใช้สำหรับดาวน์โหลดและอัปโหลดข้อมูล รวมทั้งเป็นตัวสร้างไฟล์ .torrent

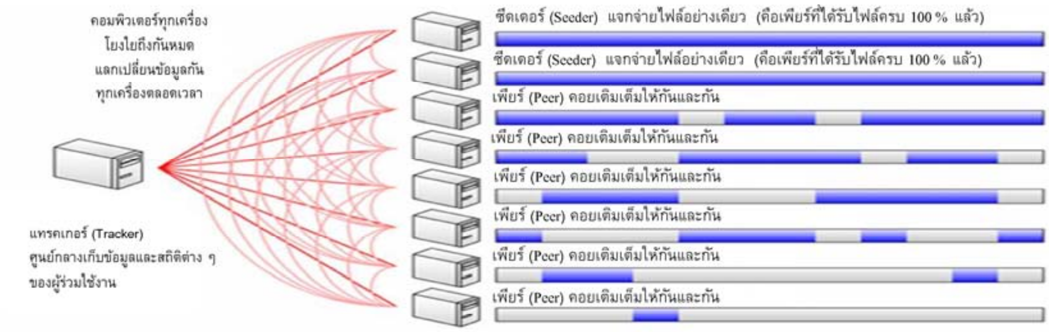
2. Tracker Server คือเซิร์ฟเวอร์ทำหน้าที่เก็บรวบรวมทอเรนต์ไฟล์ (ไฟล์ .Torrent)

3. Torrent File (ไฟล์ .Torrent) คือไฟล์ที่สำคัญซึ่งไฟล์นี้จะถูกกำหนดค่าต่าง ๆ เช่น URL ของ Tracker Server รวมถึง IP Address ของผู้ส่ง และรูปแบบของประเภทของไฟล์นั้น ๆ

หลักการทำงานของโปรโตคอลบิตทอเรนต์ มีขั้นตอนดังนี้

1. เมื่อดาวน์โหลดไฟล์ .torrent มาจาก Tracker Server แล้วใช้ Torrent client เปิดไฟล์ .torrent
2. Torrent client จะอ่านค่าที่อยู่ของ Tracker server ในไฟล์ที่เปิด แล้วติดต่อไปยัง Tracker Server เพื่อทำการส่งข้อมูลเกี่ยวกับไฟล์ที่ต้องการไปหา
3. Tracker server จะตรวจสอบว่ามีไฟล์ที่ขอมามีการลงทะเบียนไว้ในระบบหรือไม่ ถ้ามีจะตรวจดูว่ามีคนเข้ามาโหลดไฟล์นี้อยู่เท่าไหร่แล้วก็จะทำการส่งรายการ IP Address ของคนที่โหลดอยู่กลับไปรวมทั้งเก็บ IP Address ของผู้ส่งไว้ด้วย
4. เมื่อ Torrent client ได้ IP Address แล้ว ก็ทำการติดต่อไปยัง IP Address ที่ได้มา Torrent client จะตรวจว่ายังขาดส่วนไหน แล้วส่งคำร้องขอส่วนที่ต้องการไปให้ ปลายทางก็จะส่งส่วนที่ผู้ร้องขอ กลับมาให้ขณะเดียวกันนั้นเองตัว Torrent client ก็จะเปิดรับคำร้องจากคนอื่นๆ ในแบบเดียวกัน

ภายในระบบเครือข่ายที่ใช้โปรโตคอลบิตทอเรนต์ การกระจายข้อมูลจาก Seeder (ผู้ที่เป็นเจ้าของไฟล์ที่เป็นผู้เริ่มต้นแจกจ่ายไฟล์ไปให้คนอื่น ๆ)ไปยัง Downloader (ผู้ที่เข้ามาดาวน์โหลดไฟล์ หรือกำลังดาวน์โหลดอยู่แต่ยังไม่เสร็จ)นั้น เครือข่ายบิตทอเรนต์ มีวิธีการเฉพาะตัวโดยทำการแตกข้อมูลออกเป็นส่วน ๆ มีขนาดตั้งแต่ 256 กิโลไบต์ ไปจนถึง 4 เมกกะไบต์ ตามแต่ขนาดของข้อมูลที่ทำการแบ่งโดยข้อมูลแต่ละส่วนจะถูกส่งไปยัง Downloader ใด ๆ ที่อยู่ในเครือข่าย เมื่อ Downloader ได้รับข้อมูลส่วนที่ต้องการเสร็จสิ้นก็สามารถเป็นตัวแทนในการกระจายข้อมูลส่วนนั้นไปยัง Downloader อื่น ๆ ที่ยังไม่มีข้อมูลส่วนนั้นได้ เมื่อ Downloader ใด ๆ ได้รับข้อมูลครบทุกส่วนก็จะเปลี่ยนระดับตัวเองเป็น Seeder แล้วทำหน้าที่เป็นตัวแจกจ่ายข้อมูลอย่างเดียว



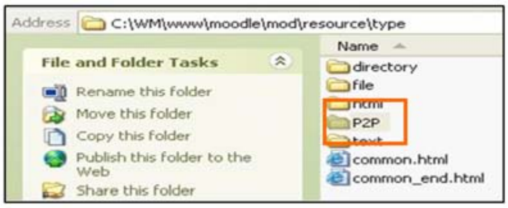
รูปที่ 1 แสดงการกระจายไฟล์ข้อมูลโปรโตคอลบิตทอเรนต์ [6]

จากรูปที่ 1 แสดงการกระจายไฟล์ข้อมูล โดยคอมพิวเตอร์ทุกเครื่องเชื่อมโยงถึงกัน และแลกเปลี่ยนข้อมูลกันทุกเครื่องตลอดเวลา โดยมี Seeder จำนวน 2 เครื่อง ทำหน้าที่แจกจ่ายไฟล์ให้กับเพียร์อื่น ๆ ซึ่งเพียร์ที่ได้รับไฟล์จาก Seeder ก็จะคอยเติมเต็มชิ้นส่วนไฟล์ที่ตัวเองได้รับนั้นให้กับเพียร์อื่น ๆ ซึ่งกันและกันจนทั้งหมดได้รับไฟล์ครบทุกส่วนเท่ากัน

4. การเพิ่มโมดูลกระจายไฟล์ข้อมูล P2P ในมูเดิ้ล

จากการศึกษาแนวทางการเพิ่มโมดูลในมูเดิ้ล โดยค้นคว้าจากเว็บไซต์ของมูเดิ้ล นั้นไม่ปรากฏวิธีการที่จะทำการเพิ่มโมดูลได้ ทำให้ผู้วิจัยได้ศึกษากลไกการทำงานของมูเดิ้ลในส่วนที่เชื่อมโยงไปยังไฟล์ในการทำงานของโมดูลที่มีอยู่ในมูเดิ้ล พบว่าโมดูลการแชร์ไฟล์ของมูเดิ้ล จะเก็บอยู่ที่ … moodle\mod\resource\type ซึ่งจะมีโฟลเดอร์ Directory คือโมดูลไดเรกทอรี, File คือโมดูลไฟล์หรือเว็บไซต์, Html คือโมดูลหน้าเว็บเพจ, Text คือโมดูลหน้าหนังสือธรรมดา เป็นต้น ซึ่งภายในโฟลเดอร์ของโมดูลจะประกอบไปด้วยไฟล์ 2 ไฟล์ด้วยกัน ยกตัวอย่าง โมดูลหน้าเว็บเพจ จะประกอบไปด้วยไฟล์ชื่อ html.html และ resource.class.php

ดังนั้นแนวทางในการเพิ่มโมดูลกระจายไฟล์ข้อมูล P2P จึงสร้างโฟลเดอร์ชื่อ P2P มาเก็บไว้ยังตำแหน่ง moodle\mod\resource\type รวมทั้งสร้างไฟล์ P2P.html และ resource.class.php อยู่ในโฟลเดอร์ P2P (หรือก็อปปี้โมดูลอื่นๆ ที่ใช้งานอยู่ แล้วทำการแก้ไขชื่อให้เหมาะสม) ดังรูปที่ 2 แสดงการสร้างโมดูลการกระจายไฟล์ข้อมูล P2P ตามที่กล่าวมาข้างต้น



รูปที่ 2 แสดงโฟลเดอร์โมดูลการกระจายไฟล์ข้อมูล

ในส่วนของการแก้ไขโค้ดโปรแกรม ต้องแก้ไข Class function ของไฟล์ resource.class.php จากรูปที่ 3 เป็นโค้ดโปรแกรมโมดูลหน้าเว็บเพจ ซึ่งเป็นโค้ดโปรแกรมที่ มูเดิ้ลได้กำหนดและใช้งานอยู่ เมื่อใช้โค้ดโปรแกรมที่มีอยู่นั้นเป็นแนวทางในการแก้ไข การเพิ่มโมดูลการกระจายไฟล์ข้อมูล P2P ต้องแก้ไขชื่อคลาส และชื่อฟังก์ชัน จาก class resource_html เป็น class

```
class resource_html extends resource_base {


function resource_html($cmid=0) {
    parent::resource_base($cmid);
}

```

รูปที่ 3 โค้ดโปรแกรมของโมดูลหน้าเว็บเพจ

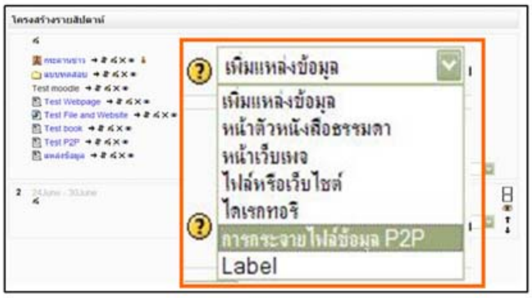
```
class resource_p2p extends resource_base {

function resource_P2P($cmid=0) {
    parent::resource_base($cmid);
}

var $parameters;
var $maxparameters = 5;
```

รูปที่ 4 โค้ดโปรแกรมของโมดูลการกระจายไฟล์ข้อมูล P2P

resource_p2p และ function resource_html เป็น function resource_p2p เมื่อแก้ไขข้อมูลดังกล่าวแล้ว แสดงได้ดังรูปที่ 4 ซึ่งเมื่อเข้าไปใช้งานมูเดิ้ล แหล่งข้อมูลก็จะปรากฏชื่อรายการที่มีข้อความจากโมดูลกระจายไฟล์ข้อมูลแบบ P2P ดังแสดงในรูปที่ 5



รูปที่ 5 แสดงรายการที่ปรากฏชื่อโมดูล กระจายไฟล์ข้อมูล P2P

5. ผลการทดลองระบบต้นแบบ

การทดลองดาวน์โหลดไฟล์ข้อมูลนั้นได้กำหนดจำนวนเครื่องคอมพิวเตอร์ในการดาวน์โหลดไว้ที่ 1, 2, 4, 16 และ 40 เครื่อง เพื่อดาวน์โหลดไฟล์ข้อมูลขนาด 600 MB คอมพิวเตอร์ที่ใช้ในการทดลองมีคุณสมบัติดังนี้ Pentium(R) D CPU 3.40 GHz, RAM 512 MB และใช้ระบบปฏิบัติการ Windows XP เชื่อมต่อเข้าด้วยกันเป็นเครือข่าย LAN ที่อัตราความเร็ว 100 Mbps

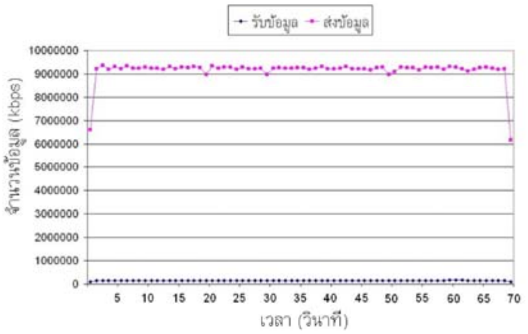
ในการทดลองได้ใช้ซอฟต์แวร์ onlineeye pro [7] เข้ามาเป็นตัวช่วยเฝ้าตรวจเวลาในการดาวน์โหลดข้อมูลและเก็บสถิติในการดาวน์โหลดข้อมูลเพื่อนำข้อมูลนั้นมาแสดงเป็นกราฟในลำดับต่อไป

จากรูปที่ 6 และรูปที่ 7 แสดงการเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์ - เซิร์ฟเวอร์ และ ระบบ P2P โดยใช้จำนวนเครื่องดาวน์โหลด 1 เครื่อง จะเห็นว่าระบบไคลเอนต์ - เซิร์ฟเวอร์โหลดข้อมูลได้สม่ำเสมอตั้งแต่ต้นจนเสร็จสิ้นขบวนการโหลด ซึ่งใช้เวลาในการดาวน์โหลดประมาณ 69 วินาที และระบบ P2P การดาวน์โหลดข้อมูลในช่วงแรกจะโหลดข้อมูลได้ช้า หลังจากนั้นมีการโหลดข้อมูลสวิงขึ้นลง จนเสร็จสิ้นขบวนการโหลด ใช้เวลาในการดาวน์โหลดประมาณ 75 วินาที

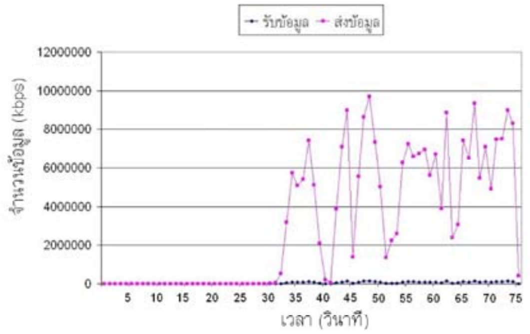
จากรูปที่ 8 และรูปที่ 9 แสดงการเปรียบเทียบการดาวน์โหลดข้อมูลระบบไคลเอนต์ - เซิร์ฟเวอร์ และ ระบบ P2P โดยใช้จำนวนเครื่องดาวน์โหลด 2 เครื่อง จะเห็นว่าระบบไคลเอนต์ - เซิร์ฟเวอร์โหลดข้อมูลจะสม่ำเสมอตั้งแต่ต้นจนเสร็จสิ้นขบวนการโหลด ซึ่งใช้เวลาในการดาวน์โหลดประมาณ 85 วินาที และระบบ

P2P มีการโหลดข้อมูลสวิงขึ้นลง จนเสร็จสิ้นขบวนการโหลด ใช้เวลาในการดาวน์โหลดประมาณ 91 วินาที

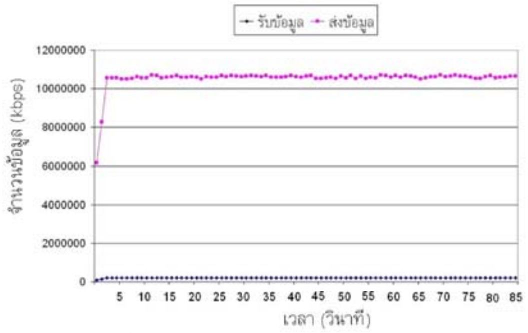
อย่างไรก็ตามเมื่อเปรียบเทียบการดาวน์โหลดทั้งสองระบบ เมื่อมีเครื่องในการดาวน์โหลดน้อยเพียง 1 - 2 เครื่อง จะใช้เวลาในการดาวน์โหลดใกล้เคียงกันทั้ง 2 ระบบ แต่เมื่อจำนวนเครื่องในการดาวน์โหลดมากขึ้น จะพบว่าระบบ P2P มีความเร็วในการดาวน์โหลดข้อมูลมากกว่า ดังแสดงในรูปที่ 10 ซึ่งแสดงให้เห็นเวลาในการดาวน์โหลดจะน้อยกว่า เมื่อจำนวนเครื่องในการดาวน์โหลดที่ 4 , 16 และ 40 เครื่อง



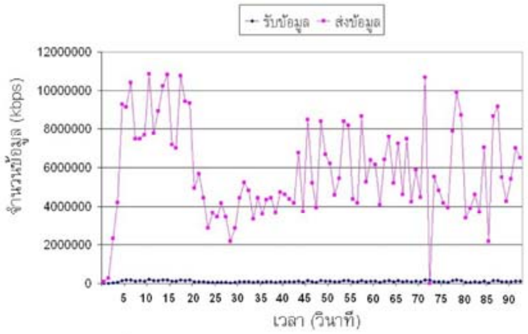
รูปที่ 6 แสดงจำนวนข้อมูลในการดาวน์โหลด
ระบบไคลเอนต์-เซิร์ฟเวอร์
จำนวนเครื่องดาวน์โหลด 1 เครื่อง



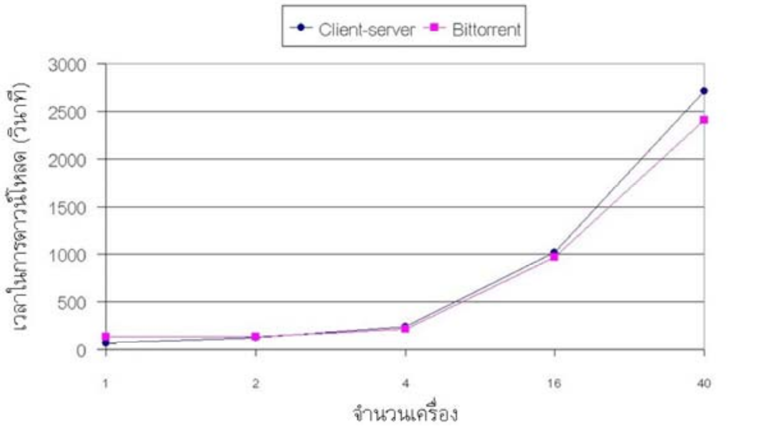
รูปที่ 7 แสดงจำนวนข้อมูลในการดาวน์โหลด
ระบบ P2P
จำนวนเครื่องดาวน์โหลด 1 เครื่อง



รูปที่ 8 แสดงจำนวนข้อมูลในการดาวน์โหลด
ระบบไคลเอนต์-เซิร์ฟเวอร์
จำนวนเครื่องดาวน์โหลด 2 เครื่อง



รูปที่ 9 แสดงจำนวนข้อมูลในการดาวน์โหลด
ระบบ P2P
จำนวนเครื่องดาวน์โหลด 2 เครื่อง



รูปที่ 10 แสดงการเปรียบเทียบเวลาในการโหลดข้อมูลระหว่างระบบไคลเอนต์ - เซิร์ฟเวอร์ และ เครือข่าย P2P

6. สรุปและงานที่จะดำเนินการต่อไป

บทความนี้ได้นำเสนอกลไกทำงานของการลดภาระความคับคั่งในการกระจายไฟล์ข้อมูลจากเว็บไซต์อีเลิร์นนิ่ง ที่พัฒนาขึ้นด้วยซอฟต์แวร์ประเภท LMS เช่น Moodle ด้วยแนวทางการสื่อสารแบบ P2P โดยการเพิ่มโมดูลการกระจายไฟล์ข้อมูลแบบ P2P ด้วยโปรโตคอลบิตทอเรนต์ และได้ทำการทดลองในการดาวน์โหลดไฟล์ข้อมูล

จากการทดลองการดาวน์โหลดไฟล์ข้อมูลเพื่อแสดงให้เห็นประสิทธิภาพ การดาวน์โหลดข้อมูลแบบเครือข่ายไคลเอนต์-เซิร์ฟเวอร์และ การดาวน์โหลดข้อมูลแบบเครือข่าย P2P เมื่อเปรียบเทียบทั้งสองเครือข่ายแล้วจะเห็นว่า ถ้าจำนวนเครื่องในการดาวน์โหลดมีจำนวน 1 และ 2 เครื่อง จำนวนเวลาในการในการดาวน์โหลดใกล้เคียงกัน โดยระบบเครือข่ายไคลเอนต์-เซิร์ฟเวอร์ มีความเร็วมากกว่า แต่เมื่อจำนวนเครื่องในการดาวน์โหลดมากขึ้นเป็น 4, 16, และ 40 เครื่อง ความเร็วในการดาวน์โหลดแบบ P2P มีความเร็วมากกว่า ดังนั้นสรุปได้ว่าเมื่อนำเทคนิคการกระจายไฟล์แบบ P2P มาใช้งานในมูเดิ้ลสามารถลดภาระความคับคั่งในการกระจายไฟล์ได้เป็นอย่างดี

อย่างไรก็ตามในการดาวน์โหลดข้อมูลแบบเครือข่าย P2P โดยใช้หลักการของบิตทอเรนต์นั้นยังทำงานได้ไม่สมบูรณ์มากนัก เนื่องจากกลไกระบบการทำงานของบิตทอเรนต์มีผลกระทบต่อประสิทธิภาพการกระจายไฟล์ข้อมูล ซึ่งมีงานวิจัยที่ตีพิมพ์ [8], [9] ได้กล่าวถึงปัญหาของโปรโตคอลบิตทอเรนต์ ในเรื่องของประสิทธิภาพการกระจายไฟล์ข้อมูล เช่นเรื่องของกลไกการทำงานแบบ Tit for Tat (ให้มากได้มาก ให้น้อยได้น้อย) , การติดต่อสื่อสารของโหนดใกล้เคียงหยุดการติดต่อเนื่องจากมีข้อผิดพลาดจากการติดต่อสื่อสารจะส่งผลให้เวลาการดาวน์โหลดของโหนดอื่น ๆ ในเพิ่มขึ้น ซึ่งปัญหาดังกล่าวนั้นจะเป็นที่มาในการทำงานวิจัยต่อไปในอนาคต

เอกสารอ้างอิง

[1] Moodle. http://www.moodle.org

[2] Bittorrent. http://www.bittorrent.org

[3] Changtao Qu and Wolfgang Nejdl, Searching SCORM Metadata in a RDF-based E-Learning P2P Network Using XQuery and Query by Example, Learning Lab Lower Saxony University of Hannover, Germany, 2003

[4] ศานติ ชิณการณ์ และประดนเดช นีละคุปต์, การวิเคราะห์ความนิยมของข้อมูลกับประสิทธิ์ภาพของเครือข่ายแจกจ่ายข้อมูลแบบ Peer-to-Peer “กรณีศึกษาBittorrent”, ภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, 2548.

[5] Michel Meulpolder, “TRIBLERCAMPUS: An Integrated Peer-to-peer Platform for File Distribution in Course Management Systems ”, Parallel and Distributed Systems Group Faculty of Electrical Engineering, Mathematics, and Computer Science Delft University of Technology, 2006.

[6] http://www.vcharkarn.com/include/article/showarticle.php?Aid=18043&page=3

[7] onlineeye pro : http://www.pmasoft.net

[8] Tan Jia Miew, John , Performance Issues in Peer-Peer (P2P) Networks Mini-Conference Report (S1 2007) – Group 2

[9] Ruchir Bindal and Pei Cao, Can Self-Organizing P2P File Distribution Provide QoS Guarantee, Department of Computer Science ,Stanford University, 2007

การออกแบบกลไกบริการ OSGi เพื่อกระจายไฟล์แบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP

A Design of OSGi-based BitTorrent File Distribution for E-Learning in UPnP Networks

<u>กิติศักดิ์ วัฒนกุล</u> และสุนทร วิฑูสุรพจน์

<u>Kitisak Watthanakul</u> and Suntorn Witosurapot

บทคัดย่อ

มาตรฐาน UPnP เป็นกลุ่มโปรโตคอลซึ่งทำหน้าที่ในการจัดการเชื่อมต่ออุปกรณ์เข้าเป็นเครือข่ายโดยอัตโนมัติ โดยไม่ต้องมีการติดตั้งค่าล่วงหน้า แต่มีข้อด้อยสำคัญด้านความคับคั่งแบบคอขวดในกรณีที่ใช้กลไกการกระจายไฟล์ข้อมูลแบบพื้นฐาน ดังเช่นที่พบในงานประยุกต์ด้านอีเลิร์นนิ่ง ซึ่งใช้อุปกรณ์ที่ทำหน้าที่เป็นเอเจนต์กระจายไฟล์ไปยังอุปกรณ์ตัวอื่นๆ ที่เหลือภายในเครือข่าย ดังนั้น เพื่อศึกษาประสิทธิภาพของการนำกลไกกระจายไฟล์แบบ peer-to-peer ด้วยโปรโตคอลบิททอเรนท์ มาใช้แก้ไขปัญหาที่พบในข้อด้อยข้างต้น ในบทความนี้เป็นการศึกษาเพื่อออกแบบแอพพลิเคชั่นให้บริการไฟล์ด้วยโปรโตคอลบิททอเรนท์ ซึ่งทำงานในสิ่งแวดล้อมของซอฟต์แวร์เชิงบริการตามมาตรฐาน OSGi และเครือข่าย UPnP ผลการทดสอบจากต้นแบบระบบที่พัฒนาขึ้นพบว่าสามารถปรับปรุงประสิทธิภาพด้านเวลาในการกระจายไฟล์ได้มาก เมื่อเปรียบเทียบกับวิธีแบบพื้นฐานแล้วจะเห็นถึงความแตกต่างทางด้านเวลาในการส่งไฟล์ข้อมูลอย่างมีนัยสำคัญ โดยเฉพาะอย่างยิ่งกรณีที่มีจำนวนอุปกรณ์ที่ใช้งานไฟล์เพิ่มมากขึ้น นอกจากนั้น ยังพบว่ากลไกแจ้งเตือนบริการในบันเดิ้ลที่ออกแบบ/พัฒนาขึ้นสามารถสนับสนุนเทคโนโลยี Mashups ได้เป็นอย่างดี จึงช่วยในการนำไปใช้ประโยชน์จากเว็บแอพพลิเคชั่นที่พัฒนาขึ้นด้วยผู้ใช้เองได้โดยง่ายและอย่างสะดวกสบาย

ABSTRACT

The Universal Plug and Play (UPnP) standard relies on a group of existing Internet standards for enabling seamless proximity networking among networked devices in automatic fashion, without requiring any configuration in priori. However, a crucial weakness of UPnP network environment may be occurred in some situation, such as in E-learning applications, where a device acting as an agent is solely responsible for distributing file contents to a number of requested devices. To find a solution for handling this weakness by means of peer-to-peer principle of file distribution, this paper aims at studying the efficiency of BitTorrent mechanism. In more specific, it focuses on the design of BitTorrent style of file distributed application working in OSGi-based environment (known as bundle) in the UPnP networks. According to the experiments taken from our first prototype, it can be seen the significant improvement of BitTorrent-based file distribution over the centralized file distribution, in terms of reducing the download time, particularly when the number of participant devices are increased. In addition, we found that the event notification can provide benefit for developing user-owned web application via the Mashups technology at ease and with convenience.

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

15 ถ.กาญจนวณิชย์ อ.หาดใหญ่ จ.สงขลา 90110

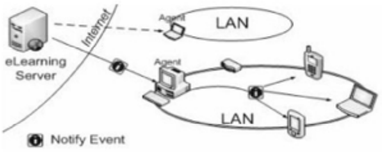
Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Songkhla 90110

บทนำ

การเรียนการสอนแบบอีเลิร์นนิ่ง (E-Learning) ซึ่งใช้การเรียนการสอนด้วยสื่ออิเล็กทรอนิกส์ผ่านอินเตอร์เน็ตเป็นสำคัญ อย่างไรก็ตาม ด้วยข้อจำกัดเชิงสถาปัตยกรรมทำงานแบบไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) ของระบบอีเลิร์นนิ่งทั่วไป ได้ก่อให้เกิดข้อด้อยสำคัญในการใช้งาน โดยเฉพาะอย่างยิ่งในกรณีที่มีการดาวน์โหลดไฟล์ข้อมูลขนาดใหญ่จากเครื่องคอมพิวเตอร์ของนักศึกษาจำนวนมากๆ ในเวลาที่ใกล้เคียงกัน ส่งผลให้เกิดความคับคั่งของภาระงานและทราฟฟิกข้อมูลที่เครื่องแม่ข่าย ทำให้ประสิทธิภาพในการให้บริการลดลงไปอย่างมากตามภาระการทำงานที่เพิ่มสูงขึ้น

แม้จะมีงานวิจัยจำนวนมากที่ได้นำเสนอวิธีการแก้ไขปัญหาข้อด้อยข้างต้นนั้น เช่น

- งานวิจัย (พิทักษ์ และสุนทร, 2553) ได้เสนอแนวคิดของการนำเอเจนต์เข้ามาใช้เพื่อเป็นตัวกลางแบบพร็อกซี่ (Proxy Agent) สำหรับลดภาระงานและทราฟฟิกที่เครื่องบริการเซิร์ฟเวอร์อีเลิร์นนิ่งกลาง โดยรับภาระดาวน์โหลดไฟล์มาเก็บไว้เอง และกระจายต่อไปยังอุปกรณ์ต่างๆ ภายในเครือข่ายย่อยอีกทอดหนึ่ง (ดังแสดงเป็นแผนภาพในรูปที่ 1) อย่างไรก็ตาม แม้ว่าแนวความคิดนี้จะนำเสนอการจัดการที่ดีที่เซิร์ฟเวอร์อีเลิร์นนิ่งกลาง แต่ก็ยังอาจเกิดปัญหาความคับคั่งแบบคอขวดขึ้นได้ที่เอเจนต์ที่เสนอแนะขึ้นใหม่นี้ได้เช่นกัน
- งานวิจัย (กิติศักดิ์ และคณะ, 2551) ได้นำเสนอด้านประสิทธิภาพของกลไกการกระจายไฟล์ข้อมูลด้วยโปรโตคอลบิททอเรนท์กับระบบอีเลิร์นนิ่งมูเดิ้ล ซึ่งเมื่อเปรียบเทียบกับวิธีการโอนไฟล์แบบพื้นฐาน จะลดความคับคั่งของทราฟฟิกข้อมูลในเครือข่ายย่อยลงได้มาก



รูปที่ 1 การใช้ประโยชน์เอเจนต์ของเครือข่ายย่อยเพื่อลดภาระงานให้เซิร์ฟเวอร์อีเลิร์นนิ่ง (พิทักษ์ และสุนทร, 2553)

อย่างไรก็ตาม งานวิจัยนี้เน้นไปที่การใช้งานจากเครือข่ายท้องถิ่น (LAN) จึงไม่รวมถึงเครือข่าย UPnP (UPnP, 2009) ซึ่งช่วยให้คอมพิวเตอร์ต่างๆ สามารถติดต่อ เพื่อเชื่อมต่อกันเป็นเครือข่ายได้อย่างอัตโนมัติ

ดังนั้น ในบทความนี้ จึงเน้นไปเพื่อแก้ไขข้อด้อยที่พบในงานวิจัยข้างต้นทั้งคู่ โดยใช้คุณลักษณะของ 1) โปรโตคอลบิททอเรนท์เข้ามาสนับสนุนที่เอเจนต์ของเครือข่าย UPnP และ 2) สถาปัตยกรรมเชิงบริการตามมาตรฐาน OSGi ซึ่งเปิดโอกาสให้เขียนคำสั่งควบคุมการทำงานของซอฟต์แวร์ได้โดยง่าย ซึ่งคาดว่าการศึกษานี้จะไม่เพียงช่วยแบ่งเบาภาระของเซิร์ฟเวอร์อีเลิร์นนิ่ง แต่ยังช่วยเสริมประสิทธิภาพการกระจายไฟล์โดยใช้ความร่วมมือของคอมพิวเตอร์ภายในระบบผ่านกลไกบิททอเรนท์อีกด้วย

วัตถุประสงค์

1. เพื่อศึกษาการประยุกต์โปรโตคอลบิททอเรนท์ มาใช้ในการพัฒนาแอพพลิเคชั่นที่สอดคล้องตามมาตรฐาน OSGi สำหรับให้บริการกระจายไฟล์ข้อมูลอีเลิร์นนิ่ง ในเครือข่าย UPnP
2. เพื่อศึกษาการประยุกต์ใช้งานเทคโนโลยี Mashups เพื่อแจ้งเตือนเหตุการณ์หรือแสดงบริการโดยอัตโนมัติ ในรูปแบบของอินเตอร์เฟส

วัสดุและระเบียบวิธีวิจัย

ในการศึกษาวิจัยนี้ ใช้เครื่องมือด้านซอฟต์แวร์ ดังนี้

1. ซอฟต์แวร์โอเพ่นซอร์ส Eclipse (Eclipse, 2010) ช่วยการพัฒนาโปรแกรมแบบเบ็ดเสร็จด้วยภาษา Java
2. ซอฟต์แวร์ไลบรารี่ Cyberlink for Java (Cyberlink, 2010) ช่วยงานด้านโปรโตคอลสื่อสารเครือข่าย UPnP
3. ซอฟต์แวร์ไลบรารี่ Apache Felix (Apache Felix, 2009) ช่วยงานควบคุมตามมาตรฐาน OSGi
4. ปลั๊กอินแบบโอเพ่นซอร์ส Open Mashups (Open Mashups, 2010) ช่วยพัฒนาเว็บแอพลิเคชั่นแบบ Mashups ผ่านทางเว็บเบราเซอร์ Firefox

5. ซอฟต์แวร์ฟรีแวร์ µTorrent (µTorrent, 2010) สำหรับควบคุมการดาวน์โหลดข้อมูลด้วยโปรโตคอลบิททอเรนท์ ซึ่งจะติดตั้งไว้ที่คอมพิวเตอร์ทุกเครื่องภายในเครือข่าย
6. ซอฟต์แวร์แชร์แวร์ onlineeye pro (Onlineeye pro, 2008) ช่วยการเฝ้าตรวจเกี่ยวกับโปรโตคอลบิททอเรนท์ เช่น เวลาในการดาวน์โหลดข้อมูล และสถิติต่างๆ เพื่อนำมาแสดงเป็นรูปกราฟต่อไป

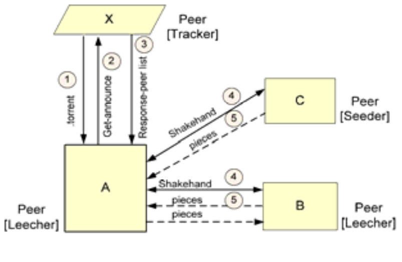
งานและทฤษฎีที่เกี่ยวข้อง

1. โปรโตคอลบิททอเรนท์

โปรโตคอลบิททอเรนท์มีหน่วยทำงานที่เกี่ยวข้อง ดังนี้

1. Tracker ทำหน้าที่เป็นแม่ข่ายบริการ (Server) เพื่อเก็บรวบรวมทอเรนท์ไฟล์ (ไฟล์ .Torrent) ซึ่งเก็บค่า URL และ IP Address ของเครื่องลูกข่ายที่ทำหน้าที่เป็นผู้ส่ง (Seeder) รวมทั้งค่าสถิติหรือข้อมูลอื่นๆ ที่จำเป็นต่อการดำเนินงานส่งต่อไฟล์ทั้งระบบ
2. Torrent Client เป็นหน่วยทำงานภายในระบบ สามารถทำหน้าที่ โดยแยกออกได้เป็น 2 ลักษณะ คือ
   - Leecher ทำหน้าที่รับข้อมูล (Download) ที่ส่งมาจากหน่วยทำงานอื่นๆ และ
   - Seeder ทำหน้าที่ส่งต่อข้อมูล (Upload) ออกไปอีกทอดหนึ่ง เมื่อได้รับการร้องขอจากตัว Leecher อื่นๆ ในภายหลัง

   ซึ่งการทำหน้าที่ทั้งสองข้างต้นนั้น ยังต้องสร้างไฟล์ .torrent เพื่อใช้ประกอบการทำงานภายหลังอีกด้วย



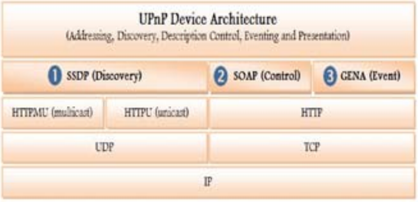
รูปที่ 2 ลำดับขั้นตอนกระจายไฟล์ด้วยกลไกบิททอเรนท์

ในรูปที่ 2 แสดงให้เห็นรายละเอียดลำดับขั้นตอนของการกระจายไฟล์ ดังต่อไปนี้

1. Torrent client A ซึ่งเป็น Leecher ดาวน์โหลดไฟล์ .torrent จาก Tracker ซึ่งได้ติดต่อไปก่อนหน้านี้ และเปิดอ่านข้อมูลภายในซึ่งเป็น URL (หรือ IP Address) ของ Tracker server ที่จะต้องไปติดต่อในลำดับถัดไป (ในกรณีนี้ เป็นหมายเลขของเครื่องเดิมที่ได้ไปอ่านไฟล์ .torrent มาก่อนหน้านั้น)
2. Torrent client A ทำการติดต่อไปยัง Tracker Server X เพื่อทำการส่งข้อมูลเกี่ยวกับไฟล์ที่ต้องการค้นหา
3. Tracker server X ส่งผลลัพธ์ที่ได้จากการค้นหา โดยกรณีที่ค้นพบจะเป็นรายการ IP Address ของ client ทั้งหมดที่ได้โหลดชิ้นส่วนไฟล์นี้ไป พร้อมบันทึก IP Address ของ client A ที่ถามนี้เพิ่มลงไปในรายการด้วย แต่หากเป็นกรณีค้นหาไม่พบ ก็จะแจ้งข้อความให้ทราบ
4. Tracker A ทำการติดต่อกันเบื้องต้น (Handshake) ตรงไปยังรายการของ client ต่างๆ ตาม IP Address ที่ได้รับแจ้งมา พร้อมแจ้งชิ้นส่วนของไฟล์ที่ต้องการ
5. Client A จะทยอยได้รับข้อมูลชิ้นส่วนของไฟล์ที่ร้องขอจาก client ที่ติดต่ออยู่นั้น (เช่น client B หรือ C) อย่างไรก็ตาม ในรูปที่ 2 จะสังเกตเห็นว่า client C ทำงานเป็น Seeder จึงมีโอกาสรับการร้องขอจาก client ตัวอื่นด้วย

2. โปรโตคอลภายในเครือข่าย UPnP

เครือข่าย UPnP ออกแบบให้ทำงานอยู่บนพื้นฐานของโปรโตคอลที่มีใช้แพร่หลายอยู่เดิมแล้วในเครือข่ายอินเตอร์เน็ต และเทคโนโลยีเว็บ (ดูรูปที่ 3 ประกอบ) เพียงแต่เพิ่มคุณลักษณะให้อุปกรณ์ทุกตัวภายในระบบสามารถติดต่อเพื่อทำงานร่วมเป็นเครือข่ายได้โดยอัตโนมัติ ตามแนวความคิดของบ้านพัก/สำนักงานอัจฉริยะ (Smart Home Network) ดังนั้น โปรโตคอลของ UPnP จึงมักมีพื้นฐานมาจากโปรโตคอลของเครือข่ายอินเตอร์เน็ต เช่น



รูปที่ 3 แผนภาพชุดโปรโตคอลของเครือข่าย UPnP

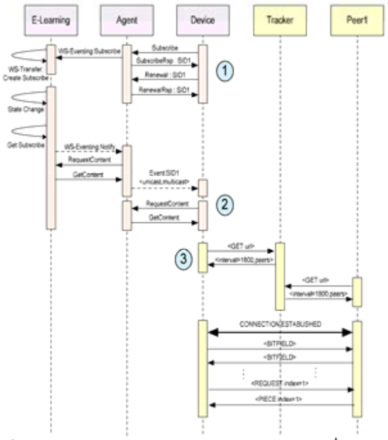
1. โปรโตคอล SSDP (Simple Service Discovery Protocol) เพื่อค้นพบบริการ (Service Discovery) ใช้บริการของโปรโตคอล HTTP (Hypertext Transfer Protocol Overview) อีกต่อหนึ่ง ทั้งในโหมดทำงานแบบมัลติคาสต์ และยูนิคาสต์
2. โปรโตคอล SOAP (Simple Object Access Protocol) เพื่อส่งข้อความชื่อและรายละเอียดของบริการหรือข้อความที่เป็นผลลัพธ์จากบริการ ในลักษณะเหมือนกับบริการของเทคโนโลยีเว็บเซอร์วิส และทำงานอยู่บนโปรโตคอล HTTP เช่นกัน
3. โปรโตคอล GENA (General Event Notification Architecture) เพื่อแจ้งเตือนเหตุการณ์ที่สนใจ ซึ่งได้สมัคร (Subscribe) ไว้ก่อนหน้าแล้ว เช่นอุปกรณ์เครื่องซักผ้าเมื่อทำงานเสร็จ จะมีข้อความแจ้งเตือนไปยังผู้สั่งงาน เป็นต้น

3. เทคโนโลยี Web Mashups

เทคโนโลยีนี้ช่วยสนับสนุนการพัฒนาเว็บแอพลิเคชั่นที่ใช้แนวทางจากการรวมข้อมูลเว็บหลากหลายรูปแบบจากหลายแหล่งข้อมูลเข้ามาใช้ประกอบกันภายในแอพลิเคชั่นเดียว แม้ว่าจะมีแนวทางการพัฒนาหลายแบบ แต่ในที่นี้สนใจแนวทางการใช้เอดิเตอร์แบบกราฟิก ซึ่งการใช้โปรแกรมจะดำเนินการผ่านรูปแบบและการเชื่อมโยงระหว่างกัน ทำให้การพัฒนาโปรแกรมสำเร็จลงได้อย่างง่ายดาย

แนวคิดของระบบบิททอเรนท์ภายในเครือข่าย UPnP

โดยพื้นฐานแล้ว กลไกบิททอเรนท์ที่เพิ่มเติมขึ้น จะอยู่ในส่วนของช่วงการแลกเปลี่ยนชิ้นส่วนของไฟล์ข้อมูล ภายหลังจากที่ได้ผ่านกระบวนการค้นหาบริการ และร้องขอเพื่อใช้บริการแล้ว ซึ่งจะได้อธิบายเป็นลำดับๆ ไป ตามขั้นตอนที่ระบุไว้ในแผนภาพเวลาในรูปที่ 4 ดังนี้



รูปที่ 4 แผนภาพลำดับเวลาการค้นหาบริการเพื่อกระจายไฟล์ด้วยบิททอเรนท์ในเครือข่ายUPnP และกระจายไฟล์

1) ช่วงการแจ้งเตือนเหตุการณ์

ในช่วงนี้เครื่องผู้เรียน (Device) จะเริ่มส่ง Subscribe Message เพื่อร้องขอลงทะเบียนบริการไปยังเอเจนต์ จากนั้นตอบกลับด้วย SubscribeResponse พร้อมกับแนบหมายเลขของการ Subscribe นั้นๆ ในที่นี้คือ SID1 จากนั้น เอเจนต์จะสร้างช่องทางการเชื่อมต่อไปยังอีเลิร์นนิ่ง เพื่อค้นหาบริการบนเซิร์ฟเวอร์ และทำการลงทะเบียนบริการด้วยการอ่าน WSDL (Web Services Description Language) จากนั้นอีเลิร์นนิ่งจะทำการสร้าง subscription เก็บไว้ในหน่วยความจำ เมื่อทำงานไปซักระยะหนึ่ง หากบริการที่ลงทะเบียนไว้มีการเปลี่ยนแปลงจะเรียก subscription ที่ทำการเก็บไว้ และทำการแจ้งเตือนกลับไปยังเอเจนต์ของกลุ่ม เมื่อเอเจนต์ได้รับการแจ้งเตือน

ก็สามารถแจ้งเตือนไปยังเครื่องผู้เรียน (Device) ในกลุ่มเครือข่าย LAN ได้

2) ช่วงการดาวน์โหลดไฟล์เอกสาร

ในช่วงนี้ใช้แนวทางแบบ Push Service เข้ามาประยุกต์ใช้กับการส่งต่อไฟล์เอกสาร โดยเมื่อเอเจนต์ในเครือข่าย ทราบว่ามีสถานะบริการเปลี่ยนแปลงไปโดยร้องขอ (RequestContent) และรับไฟล์ดังกล่าว (getContent) สำหรับการแจ้งเตือนจากเอเจนต์ไปยังเครื่องผู้เรียน (Device) ในเครือข่ายใช้การจัดการโดยผู้ใช้งานบนเครื่องเอเจนต์ โดยส่งคำสั่งการแจ้งเตือนไปยังเครื่องผู้เรียน ในเครือข่าย LAN ก็จะทราบว่ามีบริการดังกล่าว อยู่ในเครือข่าย ก็จะร้องขอ (RequestContent) และดาวน์โหลดจากเอเจนต์ (getContent) โดยตรง ไม่ต้องกระทำบนเครื่องแม่ข่าย

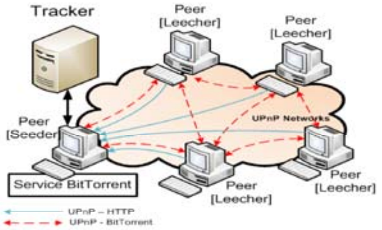
3) ช่วงการกระจายไฟล์ข้อมูลแบบบิททอเรนท์

ช่วงนี้เริ่มต้นการทำงานโดยหลังจากเครื่องผู้เรียนที่อยู่ในเครือข่าย LAN ทุก ๆ เครื่องได้ดาวน์โหลดไฟล์ .torrent จากเอเจนต์ หลังจากนั้น เครื่องผู้เรียนทำการเปิดไฟล์ .torrent ด้วยโปรแกรม µTorrent ซึ่งเป็นโปรแกรมสำหรับเป็น Client เพื่อช่วยในการกระจายไฟล์แบบบิททอเรนท์ โดยมีลำดับขั้นตอนเริ่มต้นที่ เครื่องผู้เรียนจะทำการติดต่อไปยัง Tracker เพื่อร้องขอ URL (GET URL) รวมถึง IP Address ของเครื่องผู้เรียนอื่น ๆ ในเครือข่าย จากนั้น Tracker ก็จะตอบกลับด้วยการส่ง URL และ IP Address ของเครื่องผู้เรียนที่อยู่ในเครือข่าย LAN ทั้งหมด และเครื่องผู้เรียนทั้งหมดก็จะทำการเชื่อมต่อกันโดยอัตโนมัติเพื่อทำการแลกเปลี่ยนชิ้นส่วนของไฟล์ในระบบต่อไป

ผลการทดลอง

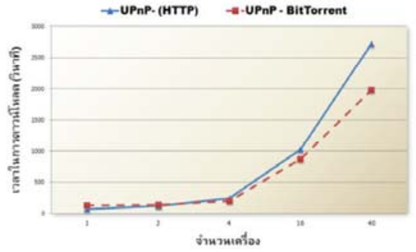
การทดลองดาวน์โหลดไฟล์ข้อมูลนั้น ได้กำหนดจำนวนเครื่องคอมพิวเตอร์ในการดาวน์โหลดไว้ที่ 1, 2, 4, 16 และ 40 เครื่อง เพื่อดาวน์โหลดไฟล์ข้อมูลขนาด 600 MB ซึ่งการกระจายไฟล์นี้ ใช้กลไกกระจายไฟล์แบบบิททอเรนท์ โดยทำการแบ่งไฟล์ออกเป็นส่วน ๆ ไว้ที่ผู้ส่ง (Seeder) จากนั้นผู้รับ (Leecher) ในเครือข่ายจะช่วยกันทำการดาวน์โหลด ชิ้นส่วนของไฟล์จาก Seeder และขณะเดียวกัน Leecher ก็จะส่งต่อชิ้นส่วนที่รับมาจาก Seeder ไปยัง Peer ตัวอื่น ๆ อีกทอดหนึ่ง จนทุก ๆ Peer ในเครือข่าย มีชิ้นส่วนของไฟล์ครบสมบูรณ์

คอมพิวเตอร์ที่ใช้ร่วมในการทดลองมีคุณสมบัติดังนี้ Pentium(R) D CPU 3.40 GHz, RAM 512 MB และใช้ระบบปฏิบัติการ Windows XP เชื่อมต่อเข้าด้วยกันเป็นเครือข่าย LAN ที่อัตราความเร็ว 100 Mbps (ดูรูปที่ 5)



รูปที่ 5 ระบบทดสอบของเครือข่าย UPnP เพื่อเปรียบเทียบกลไกกระจายไฟล์ด้วยโปรโตคอล HTTP และบิททอเรนท์

ในรูปที่ 6 เป็นผลลัพธ์ที่ได้จากการทดลอง โดยจะเห็นว่า ถ้าเครื่องที่ใช้มีจำนวนไม่มากเพียง 1 - 2 เครื่อง ระยะเวลาที่ใช้ในการดาวน์โหลดโดยใช้โปรโตคอลทั้งสองแบบนั้นจะใกล้เคียงกัน แต่เมื่อจำนวนเครื่องในการดาวน์โหลดมากขึ้นเป็น 4, 16 และ 40 เครื่อง ความเร็วในการดาวน์โหลดโดยใช้โปรโตคอลบิททอเรนท์จะมีความเร็วในการกระจายไฟล์มากกว่าเป็นลำดับ



รูปที่ 6 กราฟเปรียบเทียบเวลาดาวน์โหลดไฟล์ข้อมูล แบบพื้นฐาน (HTTP) กับแบบบิททอเรนท์ (UPnP-BitTorrent)

รูปที่ 7 ตัวอย่างเว็บแอพลิชั่นแบบ Mashups ใช้บริการดาวน์โหลดไฟล์ด้วยบิททอเรนท์ ด้วยเครื่องมือ Open Mashups

ในรูปที่ 7 แสดงให้เห็นกลไกทำงานในเว็บแอพลิเคชั่นแสดงบริการบิททอเรนท์ที่ออกแบบขึ้น ด้วยเครื่องมือ Open Mashups ในที่นี้ได้ออกแบบให้มีปุ่มกดเพื่ออัพเดทบริการ(แสดงด้วยกล่องชื่อ Update) ผ่าน XPath scapper เป็นตัวอ่านไฟล์ XML (ซึ่งเป็นไฟล์ที่เก็บค่าบริการต่างๆ) ซึ่งทำให้ผู้ใช้เองจะสามารถระบุกำหนดเงื่อนไข หรือบริการที่ต้องการได้ ในที่นี้กำหนดให้แสดงเฉพาะบริการบิททอเรนท์ ผลลัพธ์จากกล่องนี้ จะถูกส่งผ่านไปยังกล่องตัวแสดงผล (Message) ผลของรันเว็บแอพลิเคชั่นนี้แสดงให้เห็นตัวอย่างดังรูปที่ 8 ซึ่งหากต้องการใช้บริการ เพียงนำเมาส์เลือกที่บริการนั้นๆ ก็จะดาวน์โหลดงานได้ทันที

รูปที่ 8 ตัวอย่างแสดงการใช้เว็บแอพลิเคชั่นที่พัฒนาขึ้น

วิจารณ์และสรุปผล

ผลการทดลองระบบต้นแบบในบทความนี้ แสดงให้เห็นและเป็นสิ่งที่ยืนยันในเบื้องต้นได้ว่า การนำโปรโตคอลบิททอเรนท์ มาช่วยการกระจายไฟล์ข้อมูลอีเลิร์นนิ่งภายในเครือข่าย UPnP นั้น มีประโยชน์ในแง่ของการลดภาระงานของเครื่องแม่ข่ายได้เป็นอย่างดี เมื่อเปรียบเทียบกับการกระจายไฟล์แบบเดิม

บรรณานุกรม

BitTorrent, http://www.bittorrent.org [February 2010]

onlineeye pro, http://www.pmasoft.net [February 2010]

µTorrent, http://www.utorrent.com [February 2010]

Open Mashups, http://www.open-mashups.org [February 2010]

Eclipse, 2010, http://www.eclipse.org [February 2010]

UPnP Forum, 2009, UPnP Standards, Retrieved December 15, 2009, from http://www.upnp.org/

Apache Felix UPnP Documentation, 2009, http://felix.apache.org/site/apache-felix-upnp.html

กิติศักดิ์ วัฒนกุล, สุนทร วิทูสุรพจน์ และ วรรณรัช สันติอมรทัต, "การกระจายไฟล์ข้อมูลแบบ Peer to Peer สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิ้ล", การประชุมวิชาการทางวิศวกรรมศาสตร์ ครั้งที่ 6 มหาวิทยาลัยสงขลานครินทร์, อ.หาดใหญ่ จ.สงขลา, 2551

พิทักษ์ เศวตสุนทร และ สุนทร วิทูสุรพจน์, "การออกแบบและพัฒนาเอเจนต์บริการข้อมูลเว็บอีลิร์นนิ่งให้กับอุปกรณ์สื่อสารไร้สายในเครือข่าย UPnP", การประชุมวิชาการเสนอผลงานวิจัยระดับบัณฑิตศึกษา ครั้งที่ 11 มหาวิทยาลัยขอนแก่น, 2553

การออกแบบบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP
A Design of Push-based BitTorrent File Transfer Service in UPnP Networks

กิติศักดิ์ วัฒนกุล (Kitisak Watthanakul)[1] *และสุนทร วิทูสุรพจน์ (Suntorn Witosurapot)*[2]

[1, 2]*ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์*

vip.pung@gmail.com, wsuntorn@coe.psu.ac.th

บทคัดย่อ

บิททอเรนท์(BitTorrent) เป็นกลไกกระจายไฟล์ข้อมูลแบบเพียร์ทูเพียร์(Peer-to-Peer) ที่สามารถลดความคับคั่งของภาระงานที่เครื่องเซิร์ฟเวอร์(Server) ในกรณีที่มีการร้องขอจากเครื่องไคลเอนต์(Client) จำนวนมากในเวลาใกล้เคียงกันได้เป็นอย่างดี แม้จะมีงานวิจัยที่นำเสนอการปรับปรุงกลไกทำงานของบิททอเรนท์ให้เหมาะสมกับข้อดีของสถาปัตยกรรมมาตรฐานของเครือข่าย UPnP แต่ในบทความนี้ เสนอแนะเพิ่มเติมเพื่อปรับปรุงประสิทธิภาพด้านความเร็วในการดาวน์โหลดไฟล์ผ่านการใช้โมเดลการให้บริการแบบพุช(Push Service) ซึ่งผลทดสอบเบื้องต้น พบว่าได้ผลลัพธ์ที่น่าพึงพอใจ เป็นไปตามที่คาดไว้ทุกประการ

คำสำคัญ: โปรโตคอลบิททอเรนท์ ยูพีเอ็นพี การถ่ายโอนไฟล์

Abstract

BitTorrent Protocol, a peer-to-peer (P2P) based file-transfer protocol, can be alternatively used at computer server in a case that the service load becomes excessive due to huge demands of clients occurring nearly at the same time. Although it is suggested in some other works that the mechanism of BitTorrent can be improved by taking advantages of underlying UPnP architecture. In this paper, we propose a further suggestion for accelerating the download time through the Push-service model. As expects, the results take from our initial prototype can be significantly improved.

Keyword: BitTorrent Protocol, UPnP, File Transfer.

1. บทนำ

บิททอเรนท์ [1] เป็นกลไกการกระจายไฟล์ข้อมูลอยู่บนพื้นฐานเครือข่ายแบบเพียร์ทูเพียร์ ซึ่งกลไกนี้สามารถลดภาระงานที่เครื่องเซิร์ฟเวอร์ และลดความคับคั่งของทราฟฟิกข้อมูลในเครือข่ายโดยรวมได้เป็นอย่างดี [2], [3], [4] ข้อจำกัดสำคัญของกลไกกระจายไฟล์ข้อมูลแบบบิททอเรนท์ คือผู้ใช้จำเป็นต้องเข้าเว็บไซต์(Web Site) ที่เปิดให้บริการดาวน์โหลดทอเรนท์ไฟล์(ไฟล์ .torrent) เท่านั้น จึงส่งผลให้ผู้ใช้ต้องใช้เวลาในการเข้าเว็บไซต์ รวมถึงค้นหาบริการที่ต้องการอย่างหลีกเลี่ยงไม่ได้ อีกนัยหนึ่ง ผู้ใช้จำนวนมากไม่สามารถรับรู้เหตุการณ์หากมีการเปลี่ยนแปลงข้อมูลหรือเพิ่มบริการในการดาวน์โหลดในช่วงขณะที่ไม่มีการเข้าใช้งานเว็บไซต์

บทความนี้จึงมุ่งนำเสนอกลไกแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการบิททอเรนท์ ด้วยวิธีการแบบ Push ซึ่งเป็นกลไกที่ช่วยให้ผู้ใช้รับรู้เหตุการณ์ หากมีการเปลี่ยนแปลงข้อมูล ดังนั้นจึงคาดว่าจะช่วยลดปัญหาเรื่องเวลาในการเข้าใช้บริการโดยรวมและช่วยลดภาระงานของเครื่องเซิร์ฟเวอร์ได้

บทความนี้ได้วางแนวทางการนำเสนอดังนี้ ในหัวข้อย่อยที่ 2 กล่าวถึงงานวิจัยที่เกี่ยวข้อง หัวข้อย่อยที่ 3 กล่าวถึงทฤษฎีที่เกี่ยวข้อง ตามด้วยสถาปัตยกรรมของระบบที่ออกแบบขึ้นใหม่ในหัวข้อย่อยที่ 4 จากนั้นจึงเป็นการทดสอบระบบต้นแบบในหัวข้อย่อยที่ 5 และสรุปบทความในหัวข้อย่อยที่ 6

2. งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการสำหรับการกระจายไฟล์ข้อมูลแบบบิททอเรนท์มีปรากฏน้อยมาก อย่างไรก็ตาม งานวิจัย [5] ได้ศึกษาประโยชน์ของการกระจายทอเรนท์ไฟล์ โดยอาศัยกลไก RSS(Really Simple Syndication) [6] ซึ่งเป็นกลไกแบบ Pull Service โดยเครื่องไคลเอนต์ ต้องมีการร้องขอบริการไปยังเครื่องเซิร์ฟเวอร์ก่อน หากต้องการทราบการเปลี่ยนแปลงบริการที่เครื่องเซิร์ฟเวอร์

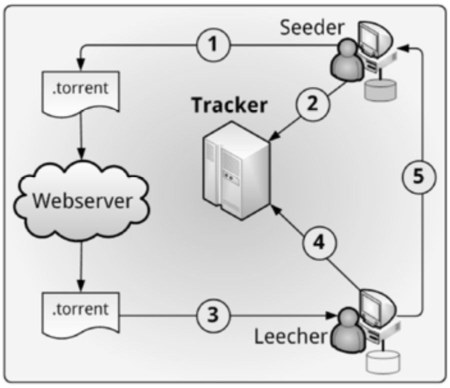
ดังนั้นแสดงให้เห็นว่า งานวิจัยดังกล่าว สามารถปรับปรุงประสิทธิภาพการกระจายทอเรนท์ไฟล์ของกลไกบิททอเรนท์ ที่ทำให้ทุกคนในระบบมีแนวโน้มได้รับไฟล์ที่เร็วขึ้นกว่าเดิม ซึ่งบทความนี้ ได้นำเสนอการใช้กลไกแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการในระบบบิททอเรนท์ด้วยโมเดลแบบ Push Service ซึ่งสามารถแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการได้ทันทีโดยที่เครื่องไคลเอนต์ไม่ต้องคอยตรวจสอบการเปลี่ยนแปลงที่เครื่องเซิร์ฟเวอร์

3. ทฤษฎีที่เกี่ยวข้อง

3.1 โปรโตคอลบิททอเรนท์

โปรโตคอลบิททอเรนท์มีหน่วยทำงานที่เกี่ยวข้อง ดังนี้

1) Tracker ทำหน้าที่เป็นเซิร์ฟเวอร์บริการ(Server) เพื่อเก็บรวบรวมทอเรนท์ไฟล์ ซึ่งเก็บค่า URL(Uniform Resource Location) และ IP(Internet Protocol) Address ของเครื่องไคลเอนต์ที่ทำหน้าที่เป็นผู้ส่ง (Seeder) รวมทั้งค่าสถิติหรือข้อมูลอื่นๆ ที่จำเป็นต่อการดำเนินงานส่งต่อไฟล์ทั้งระบบ
2) Torrent Client เป็นหน่วยทำงานภายในระบบ สามารถทำหน้าที่ ได้ 2 ลักษณะ คือ
    1) Leecher ทำหน้าที่รับข้อมูล(Download) ที่ส่งมาจากหน่วยทำงานอื่นๆ
    2) Seeder ทำหน้าที่ส่งต่อข้อมูล(Upload) ออกไปอีกทอดหนึ่งเมื่อได้รับการร้องขอจากตัวLeecher อื่นๆ ในภายหลัง(Seeder มีไฟล์ที่สมบูรณ์ เป็นผู้ส่งเพียงอย่างเดียว)



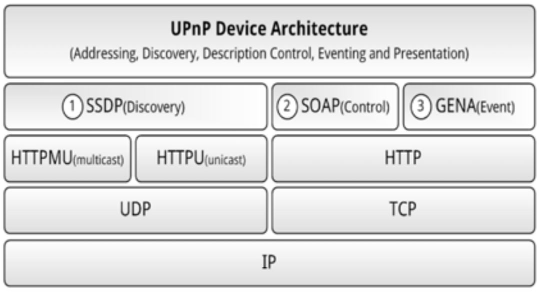
ภาพที่ 1: กลไกการกระจายไฟล์แบบบิททอเรนท์

จากภาพที่ 1 มีลำดับการทำงานของบิททอเรนท์ดังนี้

1) Seeder สร้างทอเรนท์ไฟล์ และอัพโหลดไปยังเว็บเซิร์ฟเวอร์(Webserver)
2) Seeder ติดต่อไปยัง Tracker เพื่อแบ่งปันไฟล์ที่แสดงรายละเอียดทั้งหมดในทอเรนท์ไฟล์
3) Leecher ทำการดาวน์โหลดทอเรนท์ไฟล์จากเว็บเซิร์ฟเวอร์
4) Leecher ติดต่อไปยัง Tracker ตามรายละเอียดของทอเรนท์ไฟล์ เมื่อเปิดไฟล์ขึ้นมา จากนั้น Tracker จะส่งรายละเอียดกลับมายัง Leecher ว่ามีใครแบ่งปันไฟล์อยู่ในเครือข่ายบ้าง
5) Leecher ติดต่อไปยังเครื่องที่แบ่งปันไฟล์ตามรายชื่อที่ได้รับจาก Tracker และรอรับไฟล์ต่อไป

3.2 โปรโตคอลภายในเครือข่าย UPnP

เครือข่าย UPnP(Universal Plug and Play) [7] ออกแบบให้ทำงานอยู่บนพื้นฐานของโปรโตคอลที่มีใช้แพร่หลายอยู่เดิมแล้วในเครือข่ายอินเตอร์เน็ต และเทคโนโลยีเว็บ(ดูภาพที่ 2 ประกอบ) เพียงแต่เพิ่มคุณลักษณะให้อุปกรณ์ทุกตัวภายในระบบสามารถติดต่อเพื่อทำงานร่วมเป็นเครือข่ายได้โดยอัตโนมัติตามแนวความคิดของบ้านพัก/สำนักงานอัจฉริยะ (Smart Home Network) ดังนั้นโปรโตคอลของ UPnP จึงมีพื้นฐานมาจากโปรโตคอลของเครือข่ายอินเตอร์เน็ต เช่น

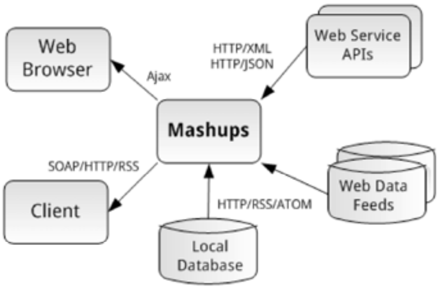


ภาพที่ 2: แผนภาพชุดโปรโตคอลเครือข่าย UPnP

1) โปรโตคอล SSDP(Simple Service Discovery Protocol) เพื่อค้นพบบริการ(Service Discovery) ใช้บริการของโปรโตคอล HTTP(Hyper Text Transfer Protocol) อีกต่อหนึ่ง ทั้งในโหมดทำงานแบบมัลติ-คาสต์ และยูนิคาสต์
2) โปรโตคอล SOAP(Simple Object Access Protocol) เพื่อส่งข้อความชื่อและรายละเอียดของบริการหรือข้อ-ความที่เป็นผลลัพธ์จากบริการ ในลักษณะเหมือนกับบริการของเทคโนโลยีเว็บเซอร์วิส และทำงานอยู่บนโปรโตคอล HTTP เช่นกัน
3) โปรโตคอล GENA(General Event Notification Architecture) เพื่อแจ้งเตือนเหตุการณ์ที่สนใจ ซึ่งได้สมัคร(Subscribe) ไว้ก่อนหน้าแล้ว เช่นอุปกรณ์เครื่องซักผ้าเมื่อทำงานเสร็จ จะมีข้อความแจ้งเตือนไปยังผู้สั่งงาน เป็นต้น

3.3 เทคโนโลยี Web Mashups

เทคโนโลยี Mashups [8] ช่วยสนับสนุนการพัฒนาเว็บแอพลิเคชั่น ที่ใช้แนวทางจากการรวมข้อมูลเว็บหลากหลายรูปแบบจากหลายแหล่งข้อมูล เข้ามาใช้ประกอบกันภายในแอพลิเคชั่นเดียว(ดูภาพที่ 3 ประกอบ) แม้ว่าจะมีแนวทางการพัฒนาหลายแบบ แต่ในที่นี้สนใจแนวทางการใช้เอดิเตอร์แบบกราฟิก ซึ่งการโปรแกรมจะดำเนินการผ่านรูปแบบและการเชื่อมโยงระหว่างกันทำให้การพัฒนาโปรแกรมสำเร็จลงได้อย่างง่ายดาย

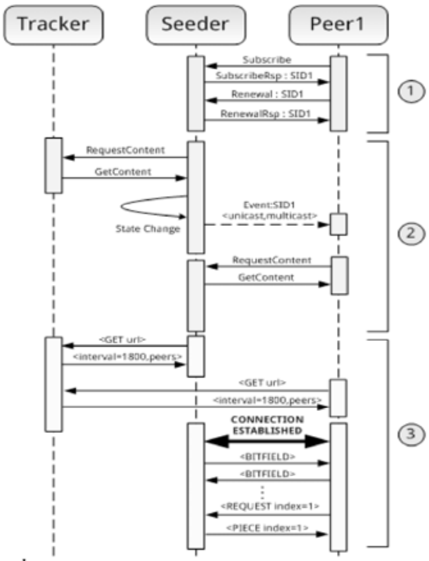


ภาพที่ 3: สถาปัตยกรรม Mashups

4. สถาปัตยกรรมของระบบที่ออกแบบขึ้นใหม่

4.1 แนวความคิด

บทความนี้ ประยุกต์ใช้ลักษณะเด่นของโปรโตคอลภายในเครือข่าย UPnP ซึ่งมีจุดเด่นในเรื่องของการค้นหาบริการโดยอัตโนมัติ โดยไม่ต้องตั้งค่าใดๆ (Zero Configuration) และมีการแจ้งเตือนบริการภายในเครือข่าย UPnP เป็นพื้นฐาน ดังนั้นจึงออกแบบและพัฒนาระบบต้นแบบเพื่อให้สามารถแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการ เพื่อกระจายไฟล์ข้อมูลแบบบิททอเรนท์ในเครือข่าย UPnP ดังแสดงขั้นตอนในภาพที่ 4



ภาพที่ 4: แผนภาพลำดับเวลาการลงทะเบียนบริการ, การแจ้งเตือนเหตุการณ์และกระจายไฟล์ข้อมูลบิททอเรนท์ในเครือข่าย UPnP

1) ช่วงการติดต่อลงทะเบียนบริการ

Peer1 ส่ง Subscribe Message เพื่อขอลงทะเบียนบริการไปยัง Seeder จากนั้นตอบกลับด้วย SubscribeResponse พร้อมกับแนบหมายเลขของการ Subscribe นั้นๆ ในที่นี้คือ SID1 ของ Peer ที่ทำการร้องขอนั้นๆ

2) ช่วงการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการ

ในช่วงนี้ใช้แนวทางแบบ Push Service เข้ามาประยุกต์ใช้กับการส่งต่อไฟล์เอกสารในเครือข่าย เมื่อทราบว่ามีสถานะบริการเปลี่ยนแปลงก็ทำการร้องขอ(RequestContent) และรับไฟล์

3) ช่วงการกระจายไฟล์ข้อมูลแบบบิททอเรนท์

ช่วงนี้เริ่มต้นการทำงาน หลังจาก Peer1 ได้รับทอเรนท์ไฟล์จาก Seeder และทำการเปิดทอเรนท์ไฟล์ ด้วยโปรแกรม µTorrent [9] จากนั้น Peer1 จะทำการติดต่อไปยัง Tracker เพื่อร้องขอ URL เพื่อทำการติดต่อและเริ่มต้นขบวนการกระจายไฟล์ข้อมูลด้วยกลไกบิททอเรนท์ไปยัง Peer อื่นๆ ในระบบต่อไป

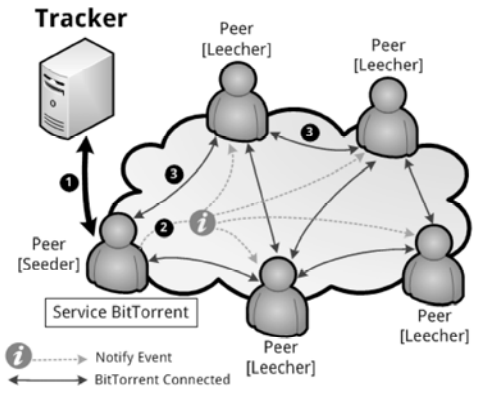
5. การทดลองระบบต้นแบบ

5.1 สภาพแวดล้อมในการทดลอง

กำหนดจำนวนเครื่องคอมพิวเตอร์ทั้งหมด 40 เครื่อง โดยแบ่งกลุ่มการทดลอง ออกเป็น 5 กลุ่ม คือ 1, 2, 4, 16 และ 40 เครื่อง เชื่อมต่อเข้าด้วยกันเป็นเครือข่าย LAN(Local Area Network) ที่อัตราความเร็ว 100 Mbps คอมพิวเตอร์ที่ใช้ในการทดลองมีคุณสมบัติดังนี้ Pentium(R) D CPU 3.40 GHz, RAM 512 MB และใช้ระบบปฏิบัติการ Windows XP (ดูภาพที่ 5)

จากภาพที่ 5 สามารถอธิบายลำดับการทดลองของระบบต้นแบบได้ดังนี้

1) Seederสร้างทอเรนท์ไฟล์ และอัพโหลดไปยัง Tracker
2) Seederแจ้งเตือนบริการไปยังเครื่องไคลเอนต์
3) กระจายไฟล์ข้อมูลด้วยกลไกบิททอเรนท์

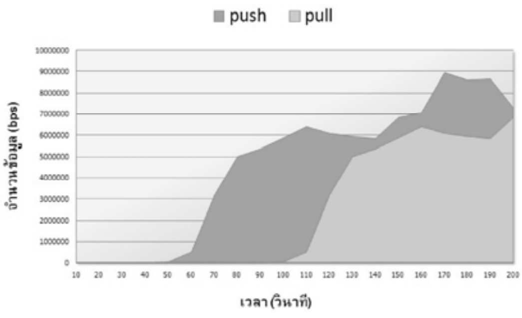


ภาพที่ 5: ระบบทดลองบิททอเรนท์ภายในเครือข่าย UPnP

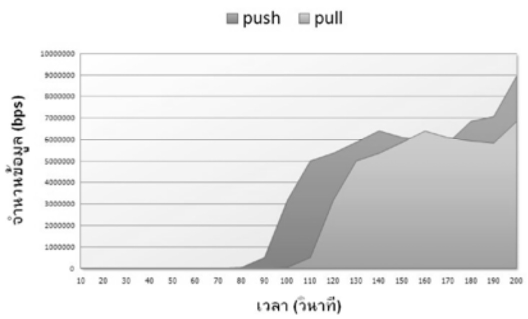
5.2 ข้อกำหนดในการทดลอง

ในการทดลองครั้งนี้ กำหนดให้มีการเปลี่ยนแปลงบริการจากเครื่องเซิร์ฟเวอร์ ที่เวลา 30, 60 และ 120 วินาที และเปรียบเทียบโมเดลการทดลองแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการ 2 รูปแบบคือ 1) Pull Service (เครื่องไคลเอนต์ต้องมีการร้องขอบริการจากเครื่องเซิร์ฟเวอร์ หรือมีการกำหนดค่าสำหรับอัพเดทข้อมูลไว้ที่ 80 และ 160 วินาที) และ 2) Push Service (เครื่องเซิร์ฟเวอร์แจ้งเตือนเหตุการณ์ไปยังเครื่องไคลเอนต์อัตโนมัติ) แสดงผลการทดลองดังต่อไปนี้

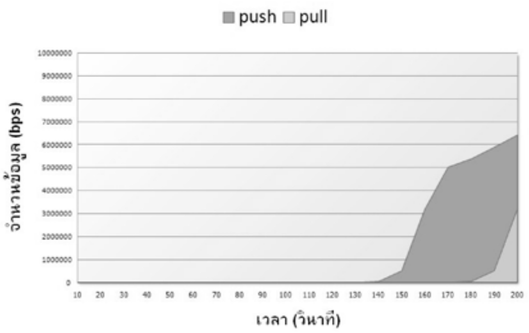
5.3 ผลการทดลอง



ภาพที่ 6: เปรียบเทียบการดาวน์โหลดไฟล์ ของกลุ่มการทดลองจำนวน 1 เครื่องเมื่อเครื่องเซิร์ฟเวอร์เปลี่ยนแปลงบริการที่ 30 วินาที



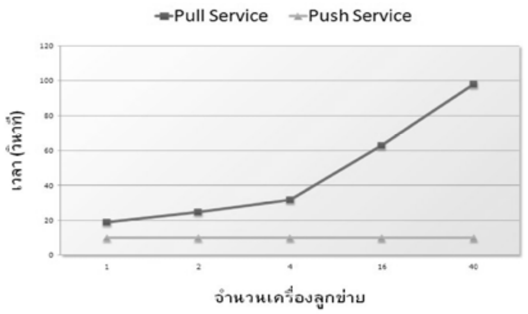
ภาพที่ 7: เปรียบเทียบการดาวน์โหลดไฟล์ ของกลุ่มการทดลองจำนวน 1 เครื่อง เมื่อเครื่องเซิร์ฟเวอร์เปลี่ยนแปลงบริการที่ 60 วินาที



ภาพที่ 8: เปรียบเทียบการดาวน์โหลดไฟล์ ของกลุ่มการทดลองจำนวน 1 เครื่อง เมื่อเครื่องเซิร์ฟเวอร์เปลี่ยนแปลงบริการที่ 120 วินาที

จากภาพที่ 6, 7 และ 8 แสดงให้เห็นถึงการเปรียบเทียบโมเดลการทดลองแจ้งเตือนเหตุการณ์เปลี่ยนแปลง 2 รูปแบบ คือ Pull Service และ Push Service จากกราฟแสดงให้เห็นว่าการแจ้งเตือนเหตุการณ์แบบ Push สามารถดาวน์โหลดไฟล์ข้อมูล ณ ขณะเวลาที่เครื่องเซิร์ฟเวอร์เปลี่ยนแปลงบริการได้โดยทันที ต่างจาก Pull ซึ่งต้องรอเวลาจนถึงช่วงเวลาที่ตั้งค่าเตรียมไว้ตั้งแต่แรก ซึ่งการทดลองครั้งนี้มีจำนวนเพียง 1 เครื่องเท่านั้น ในการดาวน์โหลดไฟล์

อย่างไรก็ตามหากมีการเข้าใช้งานระบบ จากเครื่องไคลเอนต์ในเวลาใกล้เคียงกันจำนวนมาก โมเดลการแจ้งเตือนเหตุการณ์แบบ Push Service มี ประสิทธิภาพโดยรวมทั้งระบบดีกว่าแบบ Pull Service อย่างมีนัยสำคัญ ดังแสดงในภาพที่ 9



ภาพที่ 9: เปรียบเทียบเวลาการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการระหว่าง Pull Service และ Push Service

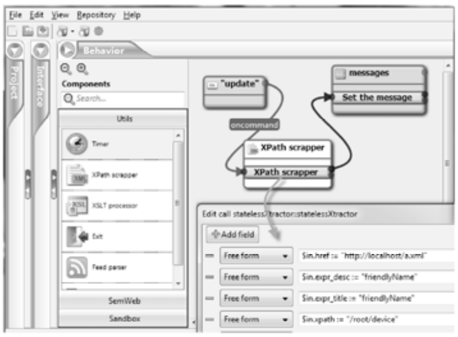
ในการทดลองดังกล่าวที่ผ่านมาเครื่องเซิร์ฟเวอร์มีการเปลี่ยนแปลงบริการ ที่เวลา 30, 60 และ 120 วินาที หากมีการ ตั้งค่าโมเดลการแจ้งเตือนวิธีการแบบ Pull Service เปลี่ยน แปลงค่าอัตโนมัติ ทุกๆ 3 วินาที เวลาที่เสียไปกับการตรวจ สอบการเปลี่ยนแปลงบริการบนเครื่องเซิร์ฟเวอร์ แสดงได้จากตารางที่ 1

ตาราง 1: เปรียบเทียบเวลาการแจ้งเตือนเหตุการณ์ระหว่าง Pull Service และ Push Service

เวลาแจ้งเตือนบริการใหม่ (วินาที)	การแจ้งเตือน (จำนวนครั้ง ก่อนรับรู้บริการใหม่)	
	Pull (อัพเดท 3 วินาที)	Push
30	10	0 (ไม่มี)
60	20	0 (ไม่มี)
120	40	0 (ไม่มี)

ในส่วนของแอพลิเคชั่นในการแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการนั้น สามารถดูได้จากภาพที่ 10 ซึ่งแสดงให้เห็นกลไกทำงานเว็บแอพลิเคชั่นแสดงบริการบิททอเรนท์ ที่ออกแบบขึ้นด้วยเครื่องมือ Open Mashups [10] ในที่นี้ได้ออกแบบให้มีปุ่มกดเพื่ออัพเดทบริการ (แสดงด้วยกล่องชื่อ Update Services) ผ่าน XPath scapper เป็นตัวอ่านไฟล์ XML (Extensible Markup Language) (เป็นไฟล์ที่เก็บค่าบริการต่างๆ) ซึ่งทำให้ผู้ใช้เองจะสามารถระบุกำหนดเงื่อนไข หรือบริการที่ต้องการได้ ในที่นี้กำหนดให้แสดงเฉพาะบริการบิททอเรนท์ผลลัพธ์จากกล่องนี้ จะถูกส่งผ่านไปยังกล่องตัว

แสดงผล (Message) ผลของรันเว็บแอพลิเคชั่นนี้แสดงให้เห็นตัวอย่างดังภาพที่ 11 ซึ่งหากต้องการใช้บริการ เพียงนำเมาส์เลือกที่บริการนั้นๆ ก็จะดาวน์โหลดงานได้ทันที



ภาพที่ 10: เครื่องมือ Open Mashups

ภาพที่ 11: แสดงการแจ้งเตือนเหตุการณ์ที่พัฒนาขึ้นด้วย Mashups

6. บทสรุป

ผลการทดลองตามแนวคิดที่นำเสนอในบทความนี้ แสดงให้เห็นถึงการเพิ่มประสิทธิภาพโดยรวมของระบบการทำงาน โดยมีกลไกแจ้งเตือนเหตุการณ์เปลี่ยนแปลงบริการประยุกต์ใช้งาน ซึ่งมีประโยชน์ในแง่ของการลดระยะเวลา ในการค้นหาบริการและตรวจสอบบริการใด ๆ บนเครื่องเซิร์ฟเวอร์ได้เป็นอย่างดี

7. เอกสารอ้างอิง

[1] BitTorrent. Available: http://www.bittorrent.org [February 2010]

[2] กิติศักดิ์ วัฒนกุล, สุนทร วิฑูสุรพจน์ และ วรรณรัช สันติอมรทัต, "การกระจายไฟล์ข้อมูลแบบ Peer to Peer สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิ้ล," *การประชุมวิชาการทางวิศวกรรมศาสตร์ ครั้งที่ 6 มหาวิทยาลัยสงขลานครินทร์*, อ.หาดใหญ่ จ.สงขลา, 2551

[3] พิทักษ์ เสวตสุนทร และ สุนทร วิฑูสุรพจน์, "การออกแบบและพัฒนาเอเจนต์บริการข้อมูลเว็บอีลิร์นนิ่งให้กับอุปกรณ์สื่อสารไร้สายในเครือข่าย UPnP," *การประชุมวิชาการเสนอผลงานวิจัยระดับบัณฑิตศึกษา ครั้งที่ 11 มหาวิทยาลัยขอนแก่น*, อ.เมือง จ.ขอนแก่น, 2553

[4] ศานติ ชิณการณ์ และประดนเดช นีละคุปต์, "การวิเคราะห์ความนิยมของข้อมูลกับประสิทธิภาพของเครือข่ายแจกจ่ายข้อมูลแบบ Peer-to-Peer "กรณีศึกษาBitTorrent"," *NECSEC'2005: The first Northeastern Computer Science and Engineering Conference*, 3 2005.

[5] Z. Zhang, Y. Lin, Y. Chen, Y. Xiong, J. Shen, H. Liu, B. Deng and X. Li, "Experimental Study of Broadcatching in BitTorrent," *Proc. of IEEE Consumer Communications and Networking Conference (CCNC'09)*, Las Vegas, Nevada, Jan. 2009.

[6] The application/rss + xml, Media Type: http://www.rssboard.org/rss- mime-type-application.txt Network Working Group, May 22, 2006.

[7] UPnP Forum, 2009, UPnP Standards, Retrieved December 15, 2009, Available: http://www.upnp.org/

[8] M. Ujjal and J. Shibsankar, "Mashups: An Emerging Content Aggregation Web 2.0 Paradigm," *7th international CALIBER-2009*, Pondicherry University, Puducherry, February 25-27, 2009.

[9] µTorrent. Available: http://www.utorrent.com [February 2010]

[10] Open Mashups. Available: http://www.open-mashups.org [February 2010]

ประวัติผู้เขียน

ชื่อ สกุล นายกิติศักดิ์ วัฒนกุล

รหัสประจำตัวนักศึกษา 4812132

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
ครุศาสตร์อุตสาหกรรมบัณฑิต (วิศวกรรมคอมพิวเตอร์)	สถาบันเทคโนโลยีราชมงคล (ศูนย์กลาง)	2545

การตีพิมพ์เผยแพร่ผลงาน

1. กิติศักดิ์ วัฒนกุล, สุนทร วิทูสุรพจน์ และ วรรณรัช สันติอมรทัต. 2551, "การกระจายไฟล์ข้อมูลแบบ Peer to Peer สำหรับซอฟต์แวร์อีเลิร์นนิ่งมูเดิล," *การประชุมวิชาการทางวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 6,* อ.หาดใหญ่ จ.สงขลา, ประเทศไทย, 8-9 พฤษภาคม 2551.
2. กิติศักดิ์ วัฒนกุล และสุนทร วิทูสุรพจน์. 2553, "การออกแบบกลไกบริการ OSGi เพื่อกระจายไฟล์แบบบิททอเรนท์สำหรับอีเลิร์นนิ่งในเครือข่าย UPnP," *การประชุมวิชาการวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต ครั้งที่ 8,* กรุงเทพมหานคร, ประเทศไทย, 19 มีนาคม 2553.
3. กิติศักดิ์ วัฒนกุล และสุนทร วิทูสุรพจน์. 2553, "การออกแบบบริการโอนไฟล์แบบพุชด้วยบิททอเรนท์ในเครือข่าย UPnP," *การประชุมวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยี ครั้งที่ 6 (NCCIT 2010),* กรุงเทพมหานคร, ประเทศไทย, 3-5 มิถุนายน 2553.