ขั้นตอนวิธีการคาดเดาแผนที่บัฟเฟอร์โดยวิธีการแบบผลักและการดึงสำหรับการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์

นายประทีป พัตราภรณ์พิศุทธิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

A PUSH-PULL WITH A BUFFER-MAP PREDICTION ALGORTITHM FOR PEER TO PEER LIVE STREAMING

Mr.Prateep Puttrapornpisut

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2011

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	ขั้นตอนวิธีการคาดเดาแผนที่บัฟเฟอร์โดยวิธีการแบบผลักและการดึงสำหรับการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์
โดย	นายประทีป พัตราภรณ์พิศุทธิ์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.กุลธิดา โรจน์วิบูลย์ชัย

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร.กุลธิดา โรจน์วิบูลย์ชัย)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์)

..... กรรมการภายนอกมหาวิทยาลัย

(รองศาสตราจารย์ ดร.อนันต์ ผลเพิ่ม)

ประทีป พัตราภรณ์พิศุทธิ์ : ขั้นตอนวิธีการคาดเดาแผนที่บัฟเฟอร์โดยวิธีการแบบผลักและการดึงสำหรับการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ (A PUSH-PULL WITH A BUFFER-MAP PREDICTION ALGORTITHM FOR PEER TO PEER LIVE STREAMING) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.กุลธิดา โรจน์วิบูลย์ชัย, 60 หน้า.

การส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์โดยวิธีการแบบผลักและดึงกำลังได้รับความนิยมมากในปัจจุบันเพราะสามารถรองรับการขยายตัวของระบบการทำงานและการมีเวลาแฝง งานวิจัยส่วนใหญ่มุ่งเน้นที่อัตราการแพร่กระจายข้อมูลที่ดีและมีความหน่วงของระบบที่ต่ำ โดยใช้การกระจายข้อมูลแบบผลักร่วมกับระบบการดึง แต่กระบวนการทำงานเหล่านี้ต้องมีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ของเพื่อนบ้าน เพื่อที่จะสามารถตัดสินใจผลักชิ้นข้อมูลให้เพื่อนบ้านและร้องขอชิ้นข้อมูลจากเพื่อนบ้าน ดังนั้นค่าใช้จ่ายในการควบคุมการทำงานที่เกิดขึ้นจากกระบวนการเหล่านี้จึงอยู่ในสัดส่วนที่สูงเพื่อที่จะรักษาความทันสมัยของข้อมูลแผนที่บัฟเฟอร์ของเพื่อนบ้าน

ในวิทยานิพนธ์ฉบับนี้ได้เสนอขั้นตอนวิธีการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านในระบบการทำงานแพร่กระจายข้อมูลแบบผลักและดึงบนโครงสร้างแบบตาข่าย เพียร์ที่อยู่ในระบบที่เสนอนั้นไม่จำเป็นต้องมีการแลกเปลี่ยนแผนที่บัฟเฟอร์ ขั้นตอนวิธีที่นำเสนอมีการจำลองการทำงานและวิเคราะห์ผลการทดลองโดยใช้ NS-2 ภายใต้สภาพแวดล้อมที่ใกล้เคียงของจริง จากการทดลองพบว่าสามารถลดค่าใช้จ่ายที่เกิดขึ้นในการควบคุมการทำงานได้มาก อีกทั้งยังสามารถลดอัตราการส่งข้อมูลที่ซ้ำซ้อนกันได้ดี

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ. ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2554

5370455021 : MAJOR COMPUTER ENGINEERING

KEYWORDS : PEER-TO-PEER/ LIVE STREAMMING/ PUSH-PULL METHOD/ MESH OVERLAY/ BUFFER-MAP

PRATEEP PUTTRAPORNPISUT : A PUSH-PULL WITH A BUFFER-MAP PREDICTION ALGORTITHM FOR PEER TO PEER LIVE STREAMING.

ADVISOR : ASST. PROF. KULTIDA ROJVIBOONCHAI, Ph.D., 60 pp.

Currently, push-pull approaches for peer-to-peer live streaming systems are interested by researchers due to high scalability and low latency. Those researchers focus on optimal data rate and delay for push-pull approaches. Nevertheless, those approaches depend on buffer-map information from all neighbors to determine which pieces to push to which neighbors or pull missed pieces from which neighbors. Therefore, control overhead is high because all neighbors need to frequently exchange their buffer-maps for up-to-date information.

This thesis proposes an algorithm to predict the buffer-maps of neighbors. The proposed algorithm uses the push-pull approach for data dissemination on mesh overlays. Peers in this algorithm do not necessarily exchange their buffer-maps. The proposed algorithm is implemented and evaluated on the ns-2 simulator which can simulate almost real situation and environment. In conclusion, this algorithm significantly reduces control overhead and also reduces duplicate data at the same time.

Department: Computer Engineering Student's Signature .....

Field of Study: Computer Engineering Advisor's Signature .....

Academic Year: 2011

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้จะไม่สามารถเกิดขึ้นได้เลยหากไม่ได้ โอกาส รวมถึงการอบรมสั่งสอน และคำแนะนำในการทำงานวิจัยต่างๆ จาก อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.กุลธิดา โรจน์วิบูลย์ชัย และคอยช่วยเหลือชี้แนะแนวทาง รวมถึงผลักดันในการทำวิจัยให้แก่ผู้วิจัยเสมอมา

กราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา ผู้ช่วยศาสตราจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์ และ รองศาสตราจารย์ ดร.อนันต์ ผลเพิ่ม ที่สละเวลามาให้ข้อเสนอแนะ และแนวคิดต่างๆ ทำให้แห็นแนวทางที่เป็นประโยชน์ต่อการพัฒนาวิทยานิพนธ์ฉบับนี้

ขอขอบคุณจุฬาลงกรณ์มหาวิทยาลัยที่ได้ให้ความรู้ การศึกษาสำหรับการศึกษาในระดับปริญญาโท รวมถึงอาจารย์ทุกท่านที่ได้อบรมสั่งสอนและให้ความรู้เสมอมา

ขอขอบคุณพี่ เพื่อนและ น้อง ที่ได้ให้ความช่วยเหลือ ข้อเสนอแนะ ตอบข้อสงสัยต่างๆ ที่เป็นประโยชน์ต่อการทำวิจัย ตลอดช่วงเวลาที่ผ่านมา

และสุดท้ายนี้ต้องขอขอบคุณ พ่อแม่ และ พี่สาว ที่ได้ให้โอกาส สนับสนุน การด้านเงิน และกำลังใจที่ดีตลอดเวลา

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฎ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 3

1.3 ขอบเขตของการวิจัย 3

1.4 ขั้นตอนและวิธีดำเนินการวิจัย 4

1.5 คุณค่าทางวิชาการ 4

1.6 ผลงานตีพิมพ์จากวิทยานิพนธ์ 4

บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง 6

2.1 ทฤษฎีที่เกี่ยวข้อง 6

2.1.1 ชั้นประยุกต์ (Application Layer) 6

2.1.2 สถาปัตยกรรมระบบผู้ให้บริการถึงผู้รับบริการ (Client-Server Architecture) 7

2.1.3 สถาปัตยกรรมระบบเพียร์ทูเพียร์ (Peer to Peer Architecture) 7

2.1.4 ไลฟ์สตรีมเสียงและภาพ (Streaming Live Audio and Video) 8

2.1.5 แผนที่บัฟเฟอร์ (Buffer-Map) 9

2.1.6 โอเวอร์เลย์ (Overlay) 9

2.1.7 การแพร่กระจายของข้อมูล (Data Dissemination) 10

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง 10

หน้า

บทที่ 3 การออกแบบโพรโทคอลการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ 20
3.1 แนวคิดในการออกแบบ 20
3.2 หลักการทำงานของโพรโทคอล 2PMP 20
3.2.1 การสร้างโอเวอร์เลย์ 20
3.2.2 หน้าต่างแผนที่บัฟเฟอร์ 21
3.2.3 การคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้าน 22
3.2.4 การแพร่กระจายชิ้นข้อมูลในหน้าต่างผลักชิ้นข้อมูล 22
3.2.5 การแพร่กระจายชิ้นข้อมูลในหน้าต่างดึงชิ้นข้อมูล 23
3.2.6 การปรับการหน้าต่างรูปแบบ (Predicting pattern) 23
3.3 ตัวอย่างการทำงานของโพรโทคอล 2PMP 27
3.4 การออกแบบโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น 29
บทที่ 4 ผลการทดลอง และวิเคราะห์ผลการทดลอง 33
4.1 ตัววัดสมรรถะของโพรโทคอล (Performance Metrics) 33
4.2 เครื่องมือในการวัดสมรรถนะของโพรโทคอล 33
4.3 สภาพแวดล้อมที่ใช้ในการทดลอง 34
4.4 ผลการทดลองความเร็วของการแพร่กระจายข้อมูลของโพรโทคอล 36
4.5 ผลการทดลองค่าใช้จ่ายของโพรโทคอล 38
4.6 ผลการทดลองการส่งข้อมูลที่ซ้ำกัน 40
4.7 ผลการทดลองความสามารถในการเล่นชิ้นข้อมูลในเวลา 41
4.8 ผลการทดลองค่าใช้จ่ายของโพรโทคอลทั้งหมด 43
4.9 ผลการทดลองโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น 44
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ 50
5.1 สรุปผลการวิจัย 50
5.2 ข้อจำกัด 50

หน้า

5.3 ข้อเสนอแนะ 51
รายการอ้างอิง 52
ภาคผนวก 54
ภาคผนวก ก ตัวอย่างข้อมูลผลการทดลองหน้าต่างรูปแบบเมื่อเวลาผ่านไป 55
ภาคผนวก ข การทดสอบขนาดของหน้าต่างรูปแบบที่เหมาะสม 58
ประวัติผู้เขียนวิทยานิพนธ์ 60

สารบัญตาราง

หน้า

ตารางที่ 1.1 ลักษณะของโปรแกรมประยุกต์ที่มีการใช้งานบนระบบเพียร์ทูเพียร์ 2
ตารางที่ 2.1 เปรียบเทียบโครงสร้างแบบต้นไม้กับแบบตาข่าย 10
ตารางที่ 2.2 การจัดรูปแบบการผลัก 14
ตารางที่ 2.3 ข้อมูลข้อความร้องขอชิ้นข้อมูล 17
ตารางที่ 2.4 ตารางเปรียบเทียบหลักการทำงานของงานวิจัยระบบเพียร์ทูเพียร์ไลฟ์สตรีมมิ่ง 19
ตารางที่ 3.1 ข้อมูลข้อความร้องขอชิ้นข้อมูล 21
ตารางที่ 3.2 ข้อมูลข้อความแจ้งเตือน 24
ตารางที่ 4.1 แสดงตัวแปรที่ใช้ในการทดลอง 35
ตารางที่ 4.2 ข้อมูลจำนวนเฉลี่ยชิ้นข้อมูลที่สามารถเล่นได้ในเวลา 42
ตารางที่ 4.3 แสดงตัวแปรที่ใช้ในการทดลองเพิ่มเติม 44
ตารางที่ 4.4 ข้อมูลจำนวนเฉลี่ยชิ้นข้อมูลที่สามารถเล่นได้ในเวลาใน 2PMP
แบบใช้ความน่าจะเป็น 48

สารบัญภาพ

หน้า

ภาพที่ 1.1 แสดงลักษณโครงข่าย เพียร์ทูเพียร์ 1
ภาพที่ 2.1 แสดงภาพ OSI Model 6
ภาพที่ 2.2 แสดง Client-Server 7
ภาพที่ 2.3 แสดง Peer to Peer Architecture 7
ภาพที่ 2.4 แสดงถึงสถาปัตยกรรมของไลฟ์สตรีม 8
ภาพที่ 2.5 แสดงแผนที่บัฟเฟอร์ 9
ภาพที่ 2.6 แสดงลักษณะโอเวอร์เลย์ 9
ภาพที่ 2.7 แสดง ระบบ Coolstreaming/donet 11
ภาพที่ 2.8 แสดงเพื่อนบ้านในระบบ DONet 11
ภาพที่ 2.9 แสดงการทำงานของ Pattern-Push 12
ภาพที่ 2.10 แสดงความหน่วงที่ผู้ใช้งาน 13
ภาพที่ 2.11 แสดงการแพร่กระจายข้อมูลแบบผลัก 15
ภาพที่ 2.12 แสดงปัญหาการหน่วงในการส่งข้อมูล 15
ภาพที่ 2.13 แสดงปัญหา Hidden Terminal 16
ภาพที่ 2.14 แสดงการแพร่กระจายข้อมูลด้วยการผลักแบบ dp/lu 16
ภาพที่ 2.15 แสดงหน้าต่างผลักและดึง 16
ภาพที่ 2.16 การคำนวณค่าความน่าจะเป็นในการส่งข้อมูลที่ไม่มีการซ้ำซ้อน 16
ภาพที่ 2.17 แสดงค่าที่ปรับ 16
ภาพที่ 2.18 แสดงความน่าจะเป็นในการส่งตามการปรับ 17
ภาพที่ 2.19 แสดง bitmap ที่มีการส่งต่อเนื่องกัน 18
ภาพที่ 3.1 แสดงหน้าต่างผลักชิ้นข้อมูลและดึงชิ้นข้อมูล 21
ภาพที่ 3.2 การสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้าน 22
ภาพที่ 3.3 การเลือกเพียร์และชิ้นที่จะส่งข้อมูลแบบผลักรูปแบบ dp/lu 22

หน้า

ภาพที่ 3.4 การเลือกเพียร์ที่จะส่งข้อมูลการร้องขอ 23
ภาพที่ 3.5 ช่วงข้อมูลในการปรับหน้าต่างรูปแบบ (Predicting pattern) 24
ภาพที่ 3.6 ผังการทำงานการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ 2PMP 26
ภาพที่ 3.7 ตัวอย่างการเริ่มโพรโทคอล 27
ภาพที่ 3.8 ตัวอย่างการแพร่กระจายข้อมูลโดยการผลัก 27
ภาพที่ 3.9 การแพร่กระจายข้อมูลโดยการดึง 28
ภาพที่ 3.10 การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบให้คาดเดาว่ามีข้อมูล 28
ภาพที่ 3.11 การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบให้คาดเดาว่าไม่มีข้อมูล 29
ภาพที่ 3.12 ผังการทำงานการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ 2PMP 32
ภาพที่ 4.1 ลักษณะสภาพแวดล้อมที่ใช้ในการทดลอง 34
ภาพที่ 4.2 กราฟแสดงความเร็วของการแพร่กระจายข้อมูล 36
ภาพที่ 4.3 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการผลัก 37
ภาพที่ 4.4 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการดึง 37
ภาพที่ 4.5 แสดงข้อมูลค่าใช้จ่ายที่เกิดขึ้นในระบบ 38
ภาพที่ 4.6 แสดงจำนวนชิ้นข้อมูลค่าใช้จ่ายที่เกิดขึ้นในระบบ 39
ภาพที่ 4.7 การส่งชิ้นส่วนข้อมูลที่ซ้ำซ้อนกัน 40
ภาพที่ 4.8 ตัวอย่างหน้าต่างรูปแบบที่คาดเดาแผนที่บัฟเฟอร์เพียร์ X เมื่อเวลา 20 วินาที ในขนาด 10 ชิ้นข้อมูล 41
ภาพที่ 4.9 แสดงจำนวนชิ้นข้อมูลที่สามารถเล่นได้ในเวลา 41
ภาพที่ 4.10 แสดงข้อมูลค่าใช้จ่ายทั้งหมดที่เกิดขึ้นในระบบ 43
ภาพที่ 4.11 กราฟแสดงความเร็วของการแพร่กระจายข้อมูลใน 2PMP แบบใช้ความน่าจะเป็น 45
ภาพที่ 4.12 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการผลักใน 2PMP แบบใช้ความน่าจะเป็น 46

หน้า

ภาพที่ 4.13 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการดึงใน 2PMP แบบใช้ความน่าจะเป็น 46
ภาพที่ 4.14 แสดงข้อมูลค่าใช้จ่ายที่เกิดขึ้นในระบบใน 2PMP แบบใช้ความน่าจะเป็น 47
ภาพที่ 4.15 การส่งชิ้นส่วนข้อมูลที่ซ้ำซ้อนกันใน 2PMP แบบใช้ความน่าจะเป็น 47
ภาพที่ 4.16 แสดงจำนวนชิ้นข้อมูลที่สามารถเล่นได้ในเวลาใน 2PMP แบบใช้ความน่าจะเป็น 48
ภาพที่ 4.17 แสดงข้อมูลค่าใช้จ่ายโดยรวมที่เกิดขึ้นในระบบใน 2PMP แบบใช้ความน่าจะเป็น 49
ภาพที่ ก.1 ตัวอย่างรูปแบบหน้าต่าง(Predicting pattern) เมื่อเวลา 0 - 20 วินาที 55
ภาพที่ ก.2 ตัวอย่างรูปแบบหน้าต่าง(Predicting pattern) เมื่อเวลา 30 - 60 วินาที 56
ภาพที่ ก.3 ตัวอย่างรูปแบบหน้าต่าง(Predicting pattern) เมื่อเวลา 70 - 90 วินาที 57
ภาพที่ ข.1 การส่งข้อมูลที่ซ้ำซ้อนกัน 58
ภาพที่ ข.2 ชิ้นข้อมูลที่ไม่สามารถเล่นได้ในเวลา 58
ภาพที่ ข.3 ค่าใช้จ่ายที่เกิดขึ้น 59

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การเพิ่มขึ้นโปรแกรมประยุกต์ใหม่ (Emerging Applications) รวมถึงทีวีทางอินเทอร์เนต และการถ่ายทอดสดทำให้มีความต้องการ การส่งข้อมูลไลฟ์สตรีมเพิ่มขึ้นและในหลายปีนี้มีจุดที่ทำให้เพียร์ทูเพียร์ได้รับความสนใจมากขึ้น คือ 1. เทคโนโลยีเพียร์ทูเพียร์ไม่ต้องการสนับสนุนการค้นหาเส้นทาง หรือโครงสร้างพื้นฐานซึ่งเป็นเหตุผลให้ค่าใช้จ่ายถูกและสามารถติดตั้งได้ง่าย 2. ในเทคโนโลยีนี้ขณะทำงานทุกคนจะไม่เพียงแต่ดาวโหลดข้อมูล แต่จะมีการอัพโหลดข้อมูลไปด้วย ซึ่งจะทำให้เกิดการทำงานแบบคู่ขนานทำให้มีความสามารถในเครือข่ายขนาดใหญ่ มีความสามารถเพิ่มขึ้นตามความต้องการที่มาก ซึ่งมักจะต้องการทรัพยากรที่มากตามไปด้วยโดยลักษณะของโครงข่ายจะมีความเชื่อมโยงกันดังรูปที่ 1.1 จากงานวิจัยที่ผ่านมาเมื่อมีคนดู 100 ล้านคนที่ส่งด้วยคุณภาพ MPEG-4 (1.5 เมกะบิตต่อวินาที (Mbps)) ก็จะมีความต้องการแบนด์วิดท์ถึง 1.5 (เทระบิตต่อวินที (Tbps))

ภาพที่ 1.1 แสดงลักษณโครงข่าย เพียร์ทูเพียร์

รูปแบบของระบบแพร่สัญญาณวีดีโอโดยทั่วไปจะเป็นแบบต้นทางเดียว (Source) โดยมีลักษณะทั่วไปดังนี้คือมีขนาดผู้ใช้จำนวนมาก, มีความต้องการคุณภาพที่ดีโดยมีความต้องการแบนด์วิดท์ประมาณร้อยกิโลบิตต่อวินาที (kbps) , ต้องการความต่อเนื่องในการเล่นร่วมถึงการส่งข้อมูลที่ทันเวลาที่จะเล่น และเพื่อให้มีความราบรื่นสามารถมีการปรับหรือลดบางอย่างได้ ซึ่งรูปแบบที่กล่าวมาเป็นลักษณะของโปรแกรมประยุกต์ที่มีการใช้งานบนระบบเพียร์ทูเพียร์ ดังตารางที่ 1.1

ที่โปรแกรมประยุกต์ส่วนมากใน On-Demand streaming หรือ Audio/Video conferencing จะเข้มงวดเรื่องความล่าช้าและมีความต้องการแบนด์วิดท์ที่สูง ส่วน แฟ้มข้อมูลดาว

โหลดจะมีชิ้นข้อมูลอยู่ในจำนวนมากที่ลำดับการส่งชิ้นข้อมูลจะไม่สำคัญ แต่ในโปรแกรมประยุกต์สตรีมมิ่งจะไม่เหมาะสมเพราะมีเรื่องเวลาที่กำหนดมาเกี่ยวข้อง

ตารางที่ 1.1 ลักษณะของโปรแกรมประยุกต์ที่มีการใช้งานบนระบบเพียร์ทูเพียร์

Category	Bandwidth-sensitive	Delay-sensitive	Scale
File download	✗	✗	Large
On-Demand streaming	✓	✓	Large
Audio/video conferencing	✓ / ✗	✓	Small
Simultaneous broadcast	✓	✓	Large

โดยปัจจุบันนี้มีโปรแกรมประยุกต์ที่มีการใช้งานการส่งข้อมูลไลฟ์สตรีมอย่างแพร่หลายอาทิ เช่น PPLive [1], Sopcast [2], JumpTV [3], UUsee [4]

จากงานวิจัยที่ผ่านมา [5] มีประเด็นปัญหาในการออกแบบที่สำคัญเช่น

- ประสิทธิภาพโอเวอร์เลย์ (Overlay Efficiency) เพื่อให้มีประสิทธิภาพต้องมองในมุมมองของระบบเครือข่ายและโปรแกรมประยุกต์
- ประสิทธิภาพของแบนด์วิดท์ (Bandwidth Efficiency) สามารถรองรับความแตกต่างของลักษณะแบนด์วิดท์ที่มีความแตกต่างกันได้
- ขนาดของระบบ (Scalability) ต้องสามารถรองรับผู้ใช้งานและการใช้งานจำนวนมากได้
- เวลาแฝง (Latency) ระยะเวลาที่ใช้ในการส่งข้อมูลต้องสามารถทันเวลาที่จะเล่น
- ระบบจัดการภายใน (Self-Organizing) ต้องสามารถรองรับและจัดการการเปลี่ยนแปลงเมื่อมีผู้รับข้อมูลในระบบเข้าและออก
- ทนทานต่อการเปลี่ยนแปลง (Robustness to high churn) สามารถรองรับการเปลี่ยนแปลงของเครือข่ายได้

จากปัญหาที่กล่าวมาได้มีงานวิจัยที่ได้แก้ไขในเรื่องของการทนทานต่อความเปลี่ยนแปลง (Robustness to high churn) [6] ที่ได้เสนอแนวทางแก้ไขปัญหาการรับส่งข้อมูลที่ซ้ำซ้อนกัน ที่เกิดจากความหน่วงที่ทำให้แผนที่บัฟเฟอร์ (Buffer-map) ที่แสดงชิ้นข้อมูล(Chunks) ในบัฟเฟอร์ที่มีอยู่ ไม่ทันสมัยเพียงพอโดยใช้รูปแบบการกระจายชิ้นข้อมูลแบบผลักและดึง

รูปแบบการดึงข้อมูลใช้ในส่วนของข้อมูลที่มีที่ใกล้จะมีการเล่นและในส่วนรูปแบบของการผลักข้อมูลใช้รูปแบบของการระบาด (Epidemic) จากงานวิจัย Epidemic live streaming: optimal performance trade-offs [7] ที่มีการศึกษารูปแบบต่างๆ ว่ารูปแบบไหนสามารถมีอัตราการแพร่ของ

ข้อมูลสูงสุด (Optimal Diffusion Rate) และ ความหน่วงในการแพร่ข้อมูลน้อยที่สุด (Optimal Diffusion Delay) ซึ่งรูปแบบที่เลือกใช้คือการเลือกผู้รับข้อมูลโดยมีความแตกต่างของจำนวนชิ้นข้อมูลที่มีในผู้ส่งแต่ไม่มีในผู้รับมากที่สุด (Most Deprives peer) และเลือกชิ้นข้อมูลโดยชิ้นข้อมูลที่มีล่าสุดในผู้ส่งแต่ไม่มีในผู้รับ (Latest useful chunk) ที่สามารถเรียกย่อว่า dp/lu

โดยรูปแบบการแพร่กระจายข้อมูลถ้าต้องมีการแลกเปลี่ยนข้อมูลระหว่างกันเพื่อที่จะสามารถรู้ข้อมูลของแผนที่บัฟเฟอร์ของเพื่อนบ้านถ้าเลือกที่จะส่งทุกครั้งที่มีอยู่จะทำให้เพิ่มค่าใช้จ่าย (Overhead) ในการส่งข้อมูล แต่ถ้าเลือกที่จะส่งตามช่วงเวลาจะสามารถลดค่าใช่จ่าย (Overhead) แต่ต้องเพิ่มเวลาแฝง (Latency) ซึ่งรูปแบบที่เลือกใช้คือการส่งข้อมูลทุกครั้งที่ได้รับชิ้นข้อมูลใหม่มา ทำให้เกิดปัญหาค่าใช้จ่าย(Overhead) ในจุดนี้มากซึ่งเป็นปัญหาในเรื่องของ ขนาดของระบบ (Scalability) ตามมา

ดังนั้นงานวิจัยนี้จึงได้นำเอารูปแบบการกระจายข้อมูลแบบผลักรูปแบบ dp/lu ร่วมกับการดึงข้อมูล บนโอเวอร์เลย์แบบตาข่าย โดยการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านแทนที่การแลกเปลี่ยนแผนที่บัฟเฟอร์ ทำให้สามารถลดค่าใช้จ่าย (Overhead) ในส่วนของการแลกเปลี่ยนแผนที่บัฟเฟอร์ อันเป็นปัญหาที่มีผลต่อเรื่องขนาดของระบบ (Scalability)

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอระบบเพียร์ทูเพียร์ ในการส่งข้อมูลไลฟ์สตรีม โดยใช้การกระจายข้อมูลแบบผลักและดึง บนโอเวอร์เลย์แบบตาข่าย โดยสามารถมีอัตราการแพร่ของข้อมูลที่ดีและความหน่วงที่น้อย มีค่าใช้จ่ายที่เกิดจากการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ที่น้อย

1.3 ขอบเขตของการวิจัย

1.การพัฒนาและทดลองทำบนโปรแกรมจำลอง NS-2.34

2.การจำลองพฤติกรรมของการทำงาน ทำบนโปรแกรมจำลอง Georgia Tech Internetwork Topology Models

3.ในการวิจัยนี้ไม่รวมในประเด็นการฉ้อโกง (Cheating) หรือ เพียร์ที่เห็นแก่ตัว (Selfish peer) ในระบบเครือข่าย

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1.ศึกษาการทำงานของระบบเพียร์ทูเพียร์

2.ออกแบบการทำงานของเพียร์ทูเพียร์ที่สามารช่วยลดค่าใช้จ่าย (Overhead) เนื่องมาจากการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์

3.พัฒนาวิธีการทำงานของเพียร์ทูเพียร์ในการส่งข้อมูลไลฟ์สตรีม

4.สร้างชุดพฤติกรรมเพื่อทำมาทดลอง

5.ทดสอบและเก็บรวบรวมผลการทดลอง

6.วิเคราะห์ผลการทดลอง

7.ปรับปรุงผลการทดลอง

8.สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 คุณค่าทางวิชาการ

1.วิธีการที่เสนอ สามารถทำให้ระบบเพียร์ทูเพียร์ในการส่งข้อมูลไลฟ์สตรีมโดยสามารถมีอัตราการแพร่ของข้อมูลที่ดีและความหน่วงที่น้อย มีค่าใช้จ่ายที่เกิดจากการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ที่น้อย

2.วิธีการที่เสนอสามารถนำไปใช้เป็นส่วนหนึ่งของระบบระบบเพียร์ทูเพียร์ที่ใช้การกระจายข้อมูลแบบผลักและดึง ที่ต้องการลดค่าใช้จ่ายที่เกิดจากการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์

1.6 ผลงานตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของงานวิทยานิพนธ์ได้รับการตีพิพม์เป็นบทความวิชาการในหัวข้อ “2PMP: A Push-Pull with Buffer-Map Prediction Algorithm for Peer to Peer Live Streaming” โดย ประทีป พัตราภรณ์พิศุทธิ์, ศุภเสฏฐ์ ชูชัยศรี, ขวัญจิรา นาคเดช, เฉลิมเอก อิทนากรวิวัฒน์ และ กุลธิดา โรจน์วิบูลย์ชัย ในบันทึกการประชุม “Intelligent Signal Processing and Communication Systems (ISPACS-2011)” ซึ่งจัดขึ้น ณ.โรงแรม Imperial Mae-ping Hotel จังหวัดเชียงใหม่ ประเทศไทย ระหว่างวันที่ 7-9 ธันวาคม 2554

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

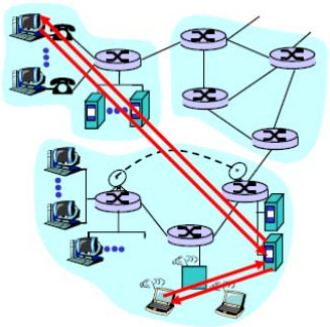
2.1.1 ชั้นประยุกต์ (Application Layer)

ภาพที่ 2.1 แสดงภาพ OSI Model

Open Systems Interconnection Basic Reference Model (OSI Model หรือ OSI Reference Model) เป็นมาตรฐานการอธิบายการติดต่อสื่อสารและโพรโทคอลของระบบคอมพิวเตอร์ที่ถูกพัฒนาขึ้นโดยองค์กรที่ชื่อว่า International Organization for Standardization (ISO) โดยมีการแบ่งแบบ Top-down Approach มี 7 ชั้นด้วยกันคือ Application, Presentation, Session, Transportation, Network, Data Link และ Physical ตามลำดับดังภาพที่ 2.1 แสดงถึง OSI Model

Application Layer เป็นชั้นบนสุด เป็นชั้นที่อยู่ใกล้ผู้ใช้มากที่สุดและเป็นชั้นที่ทำงานส่งและรับข้อมูลโดยตรงกับผู้ใช้ โดยระบบการทำงานในชั้นนี้อาทิเช่น ผู้ให้บริการถึงผู้รับบริการ (Client-server) และเพียร์ทูเพียร์

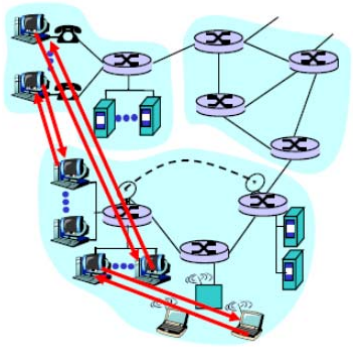
2.1.2 สถาปัตยกรรมระบบผู้ให้บริการถึงผู้รับบริการ (Client-Server Architecture)



ภาพที่ 2.2 แสดง Client-Server

ลักษณะสถาปัตยกรรมระบบผู้ให้บริการถึงผู้รับบริการ[8] แสดงดังภาพที่ 2.2 ผู้ให้บริการ (Server) มีลักษณะรูปแบบดังต่อไปนี้คือ ภาระผู้ที่ผู้บริการทั้งหมด (Always-On Host) , มีหมายเลขไอพีถาวรเพื่อการติดต่อ (Permanent IP Address) และ ผู้ให้บริการทำหน้าที่ขยายการให้บริการเอง (Server Farms for Scaling) ส่วนผู้รับบริการ(Client)มีลักษณะรูปแบบดังต่อไปนี้ มีการติดต่อกับผู้ให้บริการ (Communicate with Server), บางครั้งอาจมีการเชื่อมต่อไม่ต่อเนื่อง (May be Intermittently Connected), บางครั้งมีหมายเลขไอพีที่มีการเปลี่ยนแปลง (May have Dynamic IP Addresses) และ ไม่มีการติดต่อโดยตรงกับผู้อื่น (Do not Communicate Directly with Each Other)

2.1.3 สถาปัตยกรรมระบบเพียร์ทูเพียร์ (Peer to Peer Architecture)



ภาพที่ 2.3 แสดง Peer to Peer Architecture

ลักษณะสถาปัตยกรรมระบบเพียร์ทูเพียร์[8] ดังแสดงภาพที่ 2.3 สามารถสื่อสารได้โดยตรงกับผู้อื่น (Communicate Directly with Each Other) ทำให้สามารถใช้ทรัพยากรร่วมกันได้

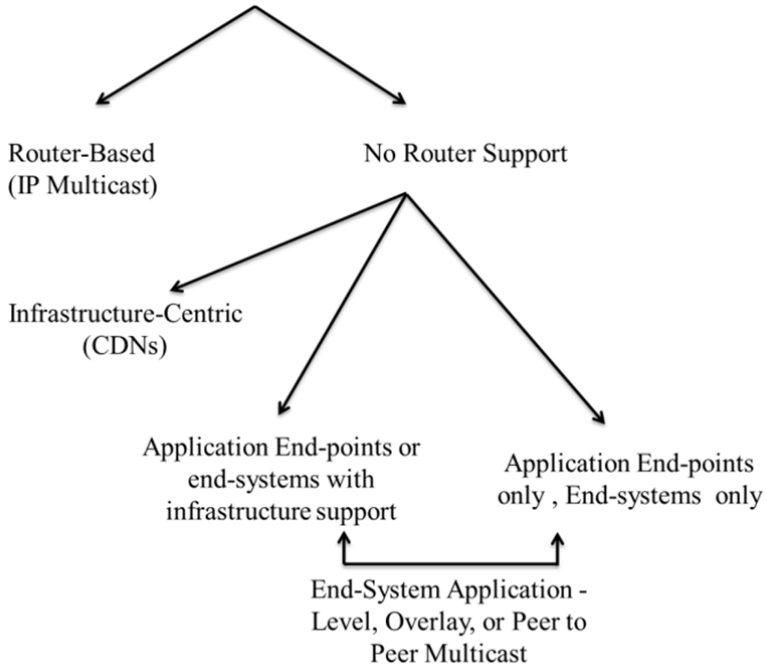
ปัญหาที่ได้มีการกล่าวถึงในระบบโปรแกรมประยุกต์เพียร์ทูเพียร์

1.ความร่วมมือกับไอเอสพี (ISP friendly) เนื่องจากระบบอินเทอร์เนตในปัจจุบันเป็น แบบอสมมาตร (Asymmetrical) คือ ดาวโหลดข้อมูลและอัพโหลดข้อมูล ไม่เท่ากันทำให้เกิดปัญหาในการใช้แบนด์วิดท์

2.การรักษาความปลอดภัย (Security) เนื่องจากเป็นระบบแบบกระจายตัว (Distributed) และมีการเปิดกว้างทำให้เป็นสิ่งที่เสี่ยงต่อระบบการรักษาความปลอดภัย

3.การกระตุ้น (Incentive) การที่ระบบเพียร์ทูเพียร์สามารถประสบความสำเร็จหรือไม่ก็ขึ้นอยู่กับผู้ใช้ที่จะแบ่งปันแบนด์วิดท์หรือทรัพยากรร่วมกันได้

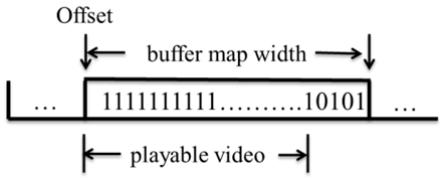
2.1.4 ไลฟ์สตรีมเสียงและภาพ (Streaming Live Audio and Video)



ภาพที่ 2.4 แสดงถึงสถาปัตยกรรมของไลฟ์สตรีม

ในโปรแกรมประยุกต์[5] ดังแสดงโดยภาพที่ 2.4 มีการถ่ายทอดเสียงวิทยุทางอินเทอร์เนตหรือไอพีทีวี (IPTV) โดยการไลฟ์สตรีมไม่จำเป็นต้องมีการเก็บข้อมูล และผู้รับไม่สามารถเล่นได้เร็วกว่าสื่อ แต่ก็ต้องมีการเก็บข้อมูลชั่วคราว และการจัดการบางอย่าง เช่นการหยุดหรือหรือดูย้อนหลัง การแพร่กระจายของข้อมูลสามารถทำโดย IP multicast, CDNs หรือ เพียร์ทูเพียร์

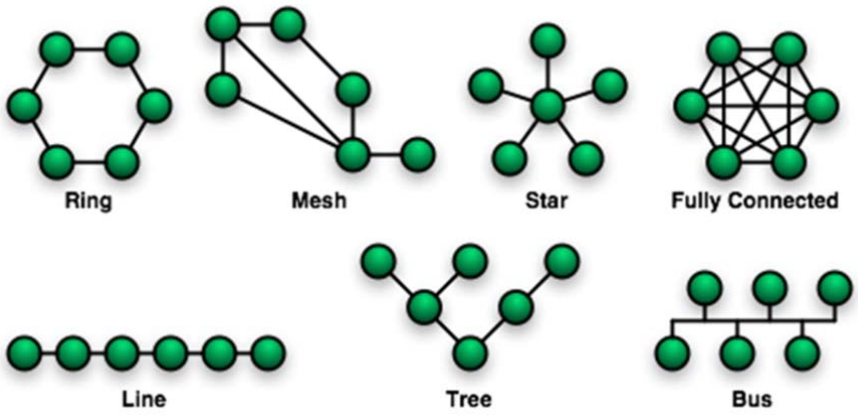
2.1.5 แผนที่บัฟเฟอร์ (Buffer-Map)



ภาพที่ 2.5 แสดงแผนที่บัฟเฟอร์

การส่งข้อมูลไลฟ์สตรีมบนเพียร์ทูเพียร์ [9] ในเพียร์จะมีการเก็บข้อมูลชิ้นส่วนที่มีอยู่ โดยหน้าต่างการเก็บข้อมูล(Window) จะเลื่อนตามการเล่นของข้อมูลที่ทำการเล่น โดยแต่ละเพียร์จะมีการให้ข้อมูลชิ้นที่ตัวเองมีอยู่กับเพียร์เพื่อนบ้าน ที่เรียกว่าการแลกเปลี่ยนแผนที่บัฟเฟอร์ ที่ระบุว่าเรามีชิ้นข้อมูล(Chunk) ทั้งหมดประกอบด้วยอะไรบ้าง และสามารถแบ่งปันชิ้นข้อมูล(Chunk)ส่วนข้อมูลใดให้ได้บ้าง ดังนี้แสดงในภาพที่ 2.5

2.1.6 โอเวอร์เลย์ (Overlay)



ภาพที่ 2.6 แสดงลักษณะโอเวอร์เลย์

โครงสร้างของโอเวอร์เลย์มีลักษณะที่แตกต่างจากลักษณะของกายภาพโดยเป็นการเชื่อมต่อเหมือนกับ สารบัญ หรือ ตัวบ่งชี้ เพื่อทำการติดต่อ โดยมีลักษณะเชื่อมโยงได้หลายรูปแบบ แบบวงแหวน, แบบตาข่าย, แบบดาว, แบบการเชื่อมต่อทุกเพียร์, แบบเส้น, แบบต้นไม้ และแบบบัส ตามภาพที่ 2.6 ซึ่งรูปแบบที่นิยมในการวิจัยทางด้านเพียร์ทูเพียร์คือแบบต้นไม้ [10], [11] และ แบบตาข่าย [12] - [14]

แบบต้นไม้ มีลักษณะโครงสร้างโดยใช้ความสัมพันธ์ ผู้ปกครอง-ลูก (Parent-Child) เมื่อเพียร์ได้รับข้อมูลมาก็จะทำการส่งข้อมูลในนั้นไปให้ลูกแล้วทำการส่งต่อไปเรื่อยๆ ทำให้เกิดไม่เกิดปัญหาข้อมูลวน และไม่จำเป็นต้องมีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ แต่มีข้อเสียคือไม่สามารถทนต่อการเปลี่ยนแปลงได้

แบบตาข่าย มีลักษณะโครงสร้างแบบสุ่มการเชื่อมต่อ ทำให้มีการติดต่อที่หลายทาง ทำให้สามารถสร้างเส้นทางได้ง่าย แต่จำเป็นต้องมีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ เพื่อที่จะสามารถลดเวลาแฝงในการ

ตารางที่ 2.1 เปรียบเทียบโครงสร้างแบบต้นไม้กับแบบตาข่าย

	+	-
Tree-based	• No Latency-Overhead Tradeoff	• Face inherent instability
Mesh-based	• Simplicity	• Latency-Overhead Tradeoff

ซึ่งสามารถสรุป ข้อดี-ข้อเสีย ของทั้งโครงสร้างแบบต้นไม้และแบบตาข่าย ดังตารางที่ 2.1 มีงานวิจัยของ Magharei และคณะ [15] แสดงให้เห็นว่ารูปแบบตาข่ายมีประสิทธิภาพดีกว่ารูปแบบต้นไม้

2.1.7 การแพร่กระจายของข้อมูล (Data Dissemination)

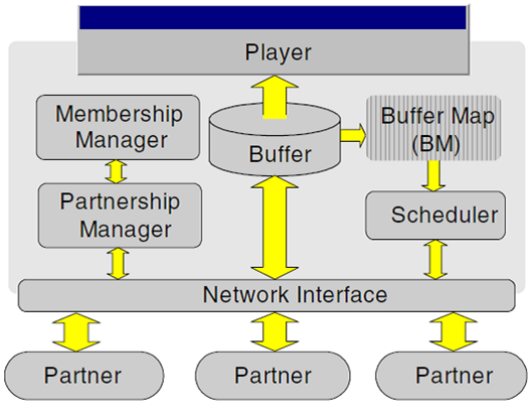
การแพร่กระจายข้อมูลสามารถแบ่งออกได้เป็น การผลัก (Push) และการดึง (Pull) โดยการกระจายข้อมูลแบบผลักจะทำการส่งข้อมูลโดยทันทีที่มีชิ้นส่วนใหม่เข้ามา โดยการตัดสินใจส่งข้อมูลนั้นมีสองรูปแบบคือ ต้องมีการรู้ข้อมูลแผนที่บัฟเฟอร์ของเพื่อนบ้านทำให้ต้องมีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ เช่น การส่งข้อมูลชิ้นสุดท้ายที่เพื่อนบ้านไม่มี กับไม่จำเป็นต้องรู้ข้อมูลแผนที่บัฟเฟอร์ของเพื่อนบ้านถ้าไม่เกี่ยวข้องในการตัดสินใจ เช่นการสุ่มชิ้นข้อมูลที่ส่ง ส่วนการกระจายข้อมูลแบบดึงนั้นเครื่องต้องการข้อมูลต้องทำการร้องขอไปยังเครื่องที่มีข้อมูลถึงสามารถส่งไปให้ดังนั้นจึงจำเป็นต้องมีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์เพื่อที่จะสามารถส่งข้อมูลร้องขอไปได้อย่างถูกต้อง

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

งานวิจัยที่พัฒนาไลฟ์สตรีมมิ่งในระบบเพียร์ทูเพียร์กำลังได้รับความนิยมและมีการพัฒนาอย่างแพร่หลาย ดังนี้

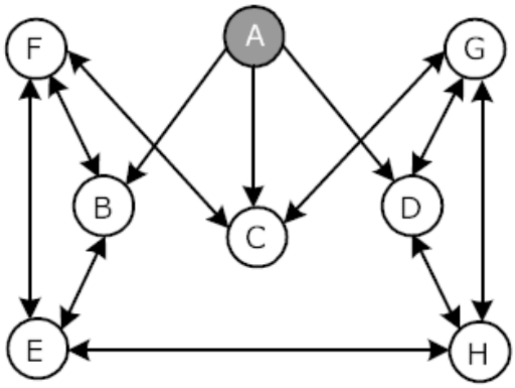
Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming [12]

งานวิจัยนี้เสนอ DONet บนระบบโอเวอร์เลย์แบบตาข่าย ในระบบการแพร่กระจายข้อมูลแบบร้องขอ ที่การทำงานนั้นง่ายแต่ต้องมีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ระหว่างกันซึ่งใช้เพียง 1 บิตเท่านั้นในการแสดงชิ้นส่วนข้อมูล โดยในการออกแบบมี 3 ส่วนด้วยกันคือ 1.ง่ายต่อการไปใช้งาน ไม่ต้องมีการสร้างโดยใช้ข้อมูลและดูแลรักษาโดยใช้ข้อมูลภาพรวมทั้งหมด 2.มีประสิทธิภาพ การส่งต่อข้อมูลสามารถปรับเปลี่ยนได้โดยไม่จำเป็นต้องไปในทิศทางไหน 3.ทนทานและยืดหยุ่น โดยโหนดเพื่อนบ้านสามารถปรับปรุงและเปลี่ยนแปลงต่อผู้ให้ข้อมูลได้หลายโหนด



ภาพที่ 2.7 แสดง ระบบ Coolstreaming/donet

การออกแบบผังการทำงานของโหนด DONet นั้นจะมีการทำงานในสามโมดูล 1.การจัดการสมาชิก 2.การจัดการเพื่อนบ้าน 3.กำหนดการ ดังภาพที่ 2.7



ภาพที่ 2.8 แสดงเพื่อนบ้านในระบบ DONet

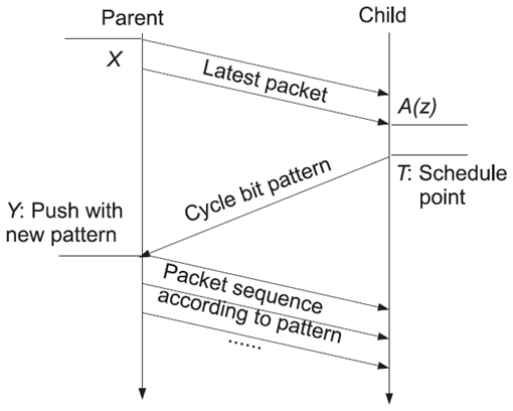
การแสดงแผนที่บัฟเฟอร์และการแลกเปลี่ยนดังภาพที่ 2.8 โดย วีดีโอที่มีการสตรีมมิ่งจะมีการแบ่งเป็นชิ้นส่วนข้อมูลโดยข้อมูลที่มีอยู่ในบัฟเฟอร์สามารถแสดงได้โดยแผนที่บัฟเฟอร์(Buffer map) ซึ่งแต่ละโหนดจะมีการแลกเปลี่ยนแผนที่บัฟเฟอร์กับเพื่อนบ้านอย่างต่อเนื่องและหลังจากนั้น ก็จะมีการดึงข้อมูลชิ้นนั้นมาจากเพื่อนบ้าน

โดยจะมีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ระหว่างกันจากการทดลองพบว่าไม่ควรแตกต่างกันเกิน 1 นาทีโดยใช้แต่ละชิ้นส่วนข้อมูลแทน 1 วินาที ซึ่งหน้าต่างจำนวน 120 ชิ้นส่วนข้อมูล สามารถแทนโดยใช้แผนที่บัฟเฟอร์จำนวน 120 บิต ในการบันทึกแผนที่บัฟเฟอร์ บิตที่มีข้อมูล 1 แทน ชิ้นส่วนข้อมูลที่มีอยู่ บิตที่มีข้อมูล 0 แทนข้อมูลที่ไม่มีอยู่ และชิ้นส่วนข้อมูลเริ่มต้นของหน้าต่างสามารถแสดงโดย 2 ไบต์ ทำให้สามารถสตรีมมิ่งวีดีโอได้มากกว่า 24 ชั่วโมง

งานวิจัยนี้เป็นงานกลุ่มแรก ที่มีการทำงานบนระบบโอเวอร์เลย์แบบตาข่าย โดยมีการแลกเปลี่ยนแผนที่บัฟเฟอร์เพื่อแสดงชิ้นส่วนข้อมูลของเพื่อนบ้าน ทำให้มีความทนทานและยืดหยุ่นต่อความเปลี่ยนแปลงเนื่องจาก สามารถมีช่องทางการส่งข้อมูลและรับได้หลายโหนด

Pattern-Push: A Low-Delay Mesh-Push Scheduling For Live Peer-to-Peer Streaming [13]

งานวิจัยนี้เสนอ Pattern-Push บนระบบโอเวอร์เลย์แบบตาข่าย ในการแพร่กระจายข้อมูลแบบผลัก แต่ละเพียร์ลูกจะมีจำนวนเพียร์ผู้ปกครองที่ส่งข้อมูลตามตารางและส่งต่อเนื่องไปที่ลูก ซึ่งเมื่อเปรียบเทียบระบบโอเวอร์เลย์แบบตาข่ายแบบเดิมจะไม่จำเป็นต้องมีการแลกเปลี่ยนแผนที่บัฟเฟอร์จากเพียร์ผู้ปกครอง โดยมีการจัดการในระดับกลุ่มข้อมูลแทนระดับชิ้นส่วนข้อมูล ในแต่ละเพียร์ปกครองจะมีการผลักกลุ่มข้อมูล ด้วยรูปแบบโดยเริ่มจากไอดีกลุ่มข้อมูลและแผนที่บิตที่เป็นระบบวนรอบ



ภาพที่ 2.9 แสดงการทำงานของ Pattern-Push

กระบวนหลักการทำงานของระบบสามารถแสดงโดยภาพที่ 2.9 โดยมีลำดับขั้นตอนต่อไปนี้

1. เริ่มต้น: แต่ละเพียร์ผู้ปกครองจะทำการส่งกลุ่มข้อมูลสุดท้ายที่ได้รับ

2.จุดจัดตาราง: เมื่อได้รับข้อมูลมาจะมีการสร้างข้อมูลรูปแบบขึ้นมา จนเมื่อมีการเปลี่ยนแปลงในเงื่อนไขของระบบเครือข่ายที่สำคัญจะมีการคำนวณใหม่

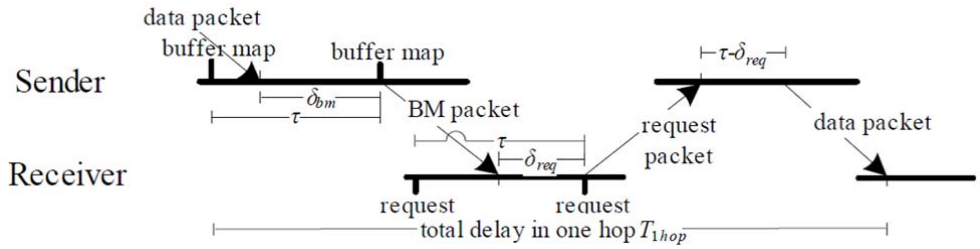
3. การส่งกลุ่มข้อมูล หลังจากที่ได้รับข้อมูลรูปแบบใหม่ (ในรูปของแผนที่บิตที่เป็นระบบวนรอบและเริ่มต้นด้วยลำดับกลุ่มข้อมูล) เพียร์ลูกก็จะมีการส่งข้อมูลกลับไปให้ผู้เพียร์ปกครอง เมื่อเพียร์ผู้ปกครองได้รับก็จะมีการส่งกลุ่มข้อมูลตามรูปแบบที่ได้รับ

4.การซ่อมแซมและฟื้นฟู: เพียร์ลูกจะมีการเก็บสำรองเพียร์ผู้ปกครองที่มีแบนด์วิดท์สูงและมีข้อมูลใหม่ ดังนั้นเมื่อตรวจพบช่องว่างก็จะมีการส่งข้อมูลร้องขอไปยังเพียร์ผู้ปกครองที่สำรองไว้ได้

งานวิจัยนี้มีลักษณะคล้ายกับโครงสร้างโอเวอร์เลย์แบบต้นไม้ ที่ไม่มีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ เนื่องจากในการส่งชิ้นข้อมูลไหนอยู่ที่การรูปแบบของเพียร์ผู้ปกครองที่มีการส่งข้อมูลให้ไม่มีการซ้ำกันให้แก่เพียร์ลูก

A peer-to-peer network for live media streaming using a push-pull approach [14]

งานวิจัยนี้เสนอ GridMedia ที่ทำงานบนโอเวอร์เลย์แบบตาข่าย และมีการแพร่กระจายข้อมูลแบบการผลักข้อมูลกับการดึงข้อมูล โดยใช้การดึงข้อมูลเพื่อให้สามารถทำงานได้ดีเมื่อมีการเปลี่ยนแปลงในระบบ ขณะที่ใช้การผลักข้อมูลเพื่อให้สามารถลดความหน่วงที่ผู้ใช้งาน



ภาพที่ 2.10 แสดงความหน่วงที่ผู้ใช้งาน

โดยในงานวิจัยนี้ได้วิเคราะห์การดึงข้อมูลในรูปแบบเดิมที่มีความหน่วงที่ผู้ใช้งานโดยมี 3 ขั้นตอนคือ เมื่อมีการได้รับกลุ่มข้อมูลมา ขั้นตอนแรก ผู้ส่งก็จะมีการบอกไปยังผู้รับว่ามีกลุ่มข้อมูลอยู่ในบัฟเฟอร์ ขั้นตอนที่สอง ถ้าผู้รับต้องการกลุ่มข้อมูลก็จะส่งข้อมูลร้องขอไป ขั้นตอนที่สาม ผู้ส่งก็จะมีการ

ส่งกลุ่มข้อมูลตามที่ได้รับการร้องขอ ดังภาพที่ 2.10 โดยเสนอรูปแบบการแพร่กระจายข้อมูลแบบการผลักข้อมูลกับการดึงข้อมูลเพื่อแก้ไขปัญหานี้

Epidemic live streaming: optimal performance trade-offs [7]

งานวิจัยนี้เสนอการทดลองรูปแบบ Epidemic ที่ทำงานบนโอเวอร์เลย์แบบตาข่าย และมีการแพร่กระจายข้อมูลแบบการผลักข้อมูล ที่กำลังได้รับความนิยมมากในปัจจุบัน โดยมีการทดลองเลือกข้อดีข้อเสียของแต่ละรูปแบบ โดยมีการเสนอรูปแบบการเลือกเพียร์หรือเลือกชิ้นส่วนก่อนขึ้นอยู่กับรูปแบบที่เลือกใช้

โดยวิธีการเลือกเพียร์และชิ้นส่วนมีรูปแบบดังต่อไปนี้

วิธีการเลือกเพียร์

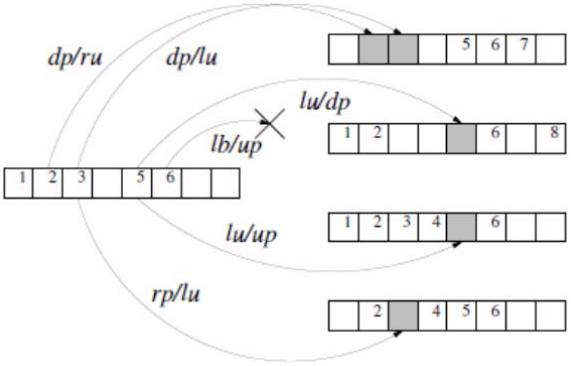
- สุ่มเพียร์ (Random peer, rp) เลือกเพียร์ที่ส่งไปให้โดยการสุ่ม
- สุ่มเพียร์ที่ต้องการข้อมูล (Random useful peer, up) เลือกเพียร์ที่ส่งไปให้โดยการสุ่มของเพียร์ที่รับข้อมูลที่ยังไม่มีชิ้นข้อมูล
- เพียร์ที่ขาดมากที่สุด (Most Deprives peer, dp) เลือกเพียร์ที่มีความแตกต่างชิ้นส่วนระหว่างผู้ส่งและรับกันมากที่สุด

วิธีการเลือกชิ้นข้อมูล

- เลือกชิ้นสุดท้าย (Latest blind chunk, lb) เลือกชิ้นที่ได้รับล่าสุดส่งไปให้
- เลือกชิ้นที่มีประโยชน์มากสุด (Latest useful chunk, lu) เลือกชิ้นที่ผู้ส่งมีล่าสุดแต่ผู้รับยังไม่มีส่งไปให้
- เลือกชิ้นที่มีประโยชน์แบบสุ่ม (Random useful chunk, ru) เลือกชิ้นที่ผู้ส่งมีแบบสุ่มแต่ผู้รับยังไม่มีส่งไปให้

ตารางที่ 2.2 การจัดรูปแบบการผลัก

Notation	Scheme
rp/lb	random peer, latest blind chunk
rp/lu	random peer, latest useful chunk
dp/lu	most deprived peer, latest useful chunk
dp/ru	most deprived peer, random useful chunk
lb/rp (= *rp/lb*)	latest blind chunk, random peer
lb/up	latest blind chunk, useful peer
lu/up	latest useful chunk, useful peer
lu/dp	latest useful chunk, most deprived peer



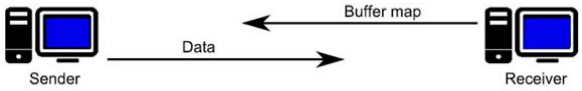
ภาพที่ 2.11 แสดงการแพร่กระจายข้อมูลแบบผลัก

โดยส่วนมากที่มีการจัดรูปแบบการผลักจะใช้วิธีการ เลือกเพียร์แล้วค่อยเลือกชิ้นส่วนข้อมูล สามารถสรุปได้ดังตารางที่ 2.1 และภาพที่ 2.11

ในส่วนของค่าใช้จ่าย (Overhead) ที่เกิดขึ้นนั้นรูปแบบ rp/lb นั้นไม่ต้องรู้ข้อมูลชิ้นส่วนของเพื่อนบ้าน ทำให้ไม่มีค่าใช้จ่ายในส่วนนี้เกิดขึ้น แต่ในรูปแบบของ dp/lu นั้นจำเป็นต้องมีการรู้ชิ้นส่วนข้อมูลของเพื่อนบ้าน จึงต้องมีการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ ทำให้เกิดค่าใช้จ่ายในส่วนนี้สูงมาก

จากผลการทดลองพบว่ารูปแบบ dp/lu, lu/dp, lu/up และ lb/up มีอัตราการแพร่ของข้อมูล (Optimal Diffusion Rate) สูงที่สุดและความหน่วงในการแพร่ข้อมูล (Optimal Diffusion Delay) น้อยที่สุด

A Reinforcement-based Push-Pull Approach for Peer-to-Peer Live Streaming [6]

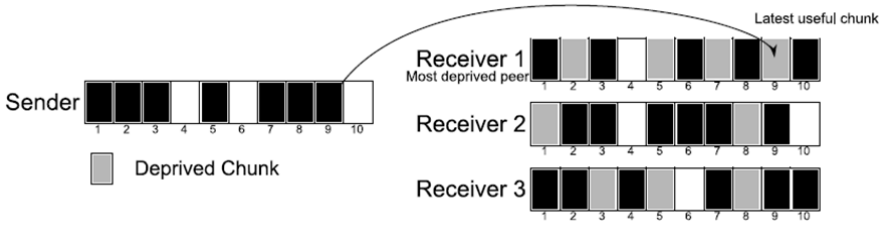


ภาพที่ 2.12 แสดงปัญหาการหน่วงในการส่งข้อมูล

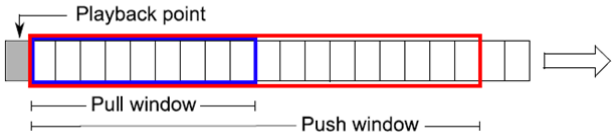
งานวิจัยนี้เสนอ Reinforcement ที่ทำงานบนโอเวอร์เลย์แบบตาข่าย และมีการแพร่กระจายข้อมูลแบบการผลักข้อมูลกับการดึงข้อมูล เพื่อแก้ไขปัญหาการเปลี่ยนแปลงของข้อมูลเมื่อมีความหน่วงในการส่งข้อมูลบนเครือข่ายดังที่แสดงในภาพที่ 2.12 ที่เมื่อข้อมูลแผนที่บัฟเฟอร์มีความหน่วงทำให้เกิดการตัดสินใจที่ผิดพลาดมีการส่งข้อมูลที่ซ้ำซ้อนกัน

ภาพที่ 2.13 แสดงปัญหา Hidden Terminal

ปัญหา Hidden Problem ดังที่แสดงในภาพที่ 2.13 โดยเมื่อผู้ส่งข้อมูลได้รับข้อมูลแผนที่บัฟเฟอร์ที่มีความหน่วง ทำให้เกิดการตัดสินใจส่งข้อมูลชุดเดียวกันไปให้ ทำให้เกิดความซ้ำซ้อนของข้อมูล



ภาพที่ 2.14 แสดงการแพร่กระจายข้อมูลด้วยการผลักแบบ dp/lu



ภาพที่ 2.15 แสดงหน้าต่างผลักและดึง

งานวิจัยนี้ใช้ระบบการแพร่กระจายข้อมูลแบบการผลักข้อมูลด้วยรูปแบบ dp/lu จากงานวิจัย Epidemic[7] ดังภาพที่ 2.14 แสดงการเลือกเพียร์และเลือกชิ้นส่วนข้อมูล ร่วมกับการดึงข้อมูล ที่แบ่งหน้าต่างในการผลักและการดึงชิ้นส่วนข้อมูลเคลื่อนตามจุดที่เล่นดังภาพที่ 2.15

$$P'_{ji} = \frac{\text{Number of non-duplicate chunks from peer } j}{\text{Number of total received chunks from peer } j}.$$

ภาพที่ 2.16 การคำนวณค่าความน่าจะเป็นในการส่งข้อมูลที่ไม่มีการซ้ำซ้อน

$$\delta_{ji} = P'_{ji} - P_{ji}$$

ภาพที่ 2.17 แสดงค่าที่ปรับ

$$P_{ji} = P_{ji} + \alpha\delta_{ji}.$$

ภาพที่ 2.18 แสดงความน่าจะเป็นในการส่งตามการปรับ

โดยมีการแก้ไขปัญหาโดยลดอัตราการส่งของข้อมูลจากเพียร์ที่ส่งซ้ำซ้อน โดยจะมีการคำนวณค่าความน่าจะเป็นในการส่งข้อมูลที่ไม่มีการซ้ำซ้อน จาก จำนวนที่ได้รับไม่ซ้ำหารด้วยจำนวนที่ได้รับทั้งหมด ดังภาพที่ 2.16 และจะส่งข้อมูล ข้อความน่าจะเป็นของการผลักข้อมูล (Push reinforcement message) ดังตารางที่ 2.3 ไปยังผู้ส่ง จากนั้นผู้ส่งจะคำนวณความน่าจะเป็นที่ได้รับไปคูณกับขั้นในการปรับ(Step size) ดังภาพที่ 2.17 และ 2.18 เพื่อลดความน่าจะเป็นในการผลักชิ้นข้อมูลเมื่อมีการตรวจพบอัตราการส่งข้อมูลซ้ำซ้อนกัน

ตารางที่ 2.3 ข้อมูลข้อความร้องขอชิ้นข้อมูล

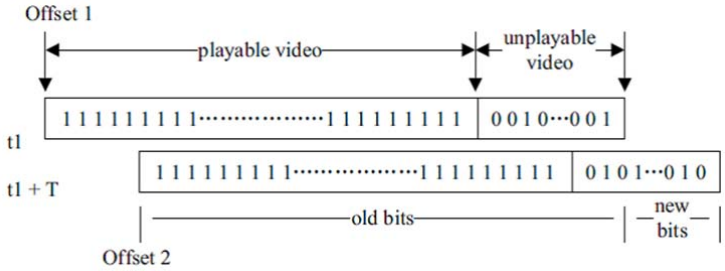
ชื่อ	รายละเอียด	ขนาดข้อมูล
Node ID	หมายเลขโหนด	4 bytes
Probability	ความน่าจะเป็นในการผลักชิ้นข้อมูล	8 bytes

A Bitmap Coding Method for P2P Streaming Protocols [9]

งานวิจัยนี้เสนอการลดค่าใช้จ่าย (Overhead) ในการส่งข้อมูลด้วยวิธีการเข้ารหัส โดยใช้ส่วนที่มีการเปลี่ยนแปลงระหว่างบิตที่ส่งไปยังเพียร์เดียวกันที่ส่งต่อเนื่องกัน ซึ่งรูปแบบที่งานวิจัยนี้ใช้ทำให้มีส่งข้อมูลที่น้อยลงเนื่องจากการเข้ารหัสนั้นทำให้ขนาดข้อมูลที่ส่งนั้นสั้นลง

รูปแบบของการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ (Buffer-map) หรือสามารถเรียกย่อว่าบิตแมพ (Bitmap) ที่มีการใช้ในงานวิจัยนี้จะประกอบไปด้วย 1.ออฟเซต (Offset) แสดงถึงชิ้นข้อมูลที่เริ่มต้น 2.ความยาวของแผนที่บัฟเฟอร์ 3.แผนที่ชิ้นส่วนข้อมูลที่มีรูปแบบฐานสองที่แสดงถึงชิ้นข้อมูลที่ตัวเองมีอยู่

ดังนั้นจึงใช้ความเหมือนกันของลักษณะแผนที่บัฟเฟอร์ที่ส่งต่อเนื่องกันโดยส่งเฉพาะส่วนที่มีความแตกต่างกัน ดังภาพที่ 2.19 จะเห็นได้ว่าข้อมูลของแผนที่บัฟเฟอร์ที่ส่งไปให้ทั้งสองครั้งที่ส่งต่อเนื่องกัน มีข้อมูลส่วนที่เหมือนเดิม และส่วนที่แตกต่างกัน



ภาพที่ 2.19 แสดง bitmap ที่มีการส่งต่อเนื่องกัน

แต่งานวิจัยนี้มีข้อจำกัดคือต้องการแผนที่บัฟเฟอร์ที่ถูกต้องอย่างน้อยหนึ่งตัวในการถอดรหัส และข้อมูลของการแลกเปลี่ยนแผนที่บัฟเฟอร์ที่มีการเข้ารหัสไม่สามารถสูญหายได้ มิฉะนั้นต้องมีการเริ่มส่งการเข้ารหัสใหม่ทั้งหมด โดยการทดลองที่ใช้ได้เพิ่ม parity check เข้าไปในกระบวนการ

โพรโทคอล	แบบโอเวอร์เลย์	แบบการกระจายข้อมูล	แผนที่บัฟเฟอร์	อัตราการแพร่ของข้อมูลสูงที่สุดและความหน่วงในการแพร่ข้อมูลน้อยที่สุด	ใช้การคาดการณ์แผนที่บัฟเฟอร์
Coolstream/DONet	ตาข่าย	ดึง	ใช้	ไม่	ไม่ใช้
Pattern-Push	ตาข่าย	ผลัก	ไม่ใช้	ไม่	ไม่ใช้
GridMedia	ตาข่าย	ผลักและดึง	ใช้	ไม่	ไม่ใช้
Epidemic	ตาข่าย	ผลัก	ใช้	ใช่	ไม่ใช้
Reinforcement	ตาข่าย	ผลักและดึง	ใช้	ใช่	ไม่ใช้
Proposed	ตาข่าย	ผลักและดึง	ไม่	ใช่	ใช้

ตารางที่ 2.4 ตารางเปรียบเทียบหลักการทำงานของงานวิจัยระบบเพียร์ทูเพียร์ไลฟ์สตรีมมิ่ง

บทที่ 3

การออกแบบโพรโทคอลการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์

การส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ ถูกออกแบบโดยคำนึงถึงปัจจัยที่สำคัญ 2 ประการ ดังต่อไปนี้คือ 1) ความสามารถในการแพร่กระจายข้อมูล ให้สามารถมีอัตราการแพร่ของข้อมูลที่ดีและความหน่วงในการแพร่ข้อมูลน้อยที่สุด 2) ค่าใช้จ่าย ที่เกิดขึ้นจากการส่งข้อมูล ซึ่งถือเป็นจุดประสงค์หลักของโพรโทคอล

3.1 แนวคิดในการออกแบบ

การออกแบบโพรโทคอลสำหรับการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์มีแนวคิดในการออกแบบดังนี้

1.การส่งข้อมูลแผนที่บัฟเฟอร์มีปัญหาในเรื่องความหน่วงทำให้เกิดความช้าล่าในการรับรู้ข้อมูล ดังนั้นการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ไม่ได้ทำให้เกิดประโยชน์ทุกครั้ง

2.การแพร่กระจายข้อมูลแบบการผลักในรูปแบบของการเลือกเพียร์โดยเพียร์ที่ขาดมากที่สุด (Most Deprives peer) และเลือกชิ้นข้อมูลโดยเลือกชิ้นที่มีประโยชน์มากสุด (Latest Useful Chunk) จะทำให้มีการส่งข้อมูลใหม่ๆไปให้ผู้รับอยู่เสมอ ดังนั้นข้อมูลแผนที่บัฟเฟอร์ที่สามารถทำให้การแพร่กระจายข้อมูลได้เร็วแต่ไม่จำเป็นต้องถูกต้อง ก็จะส่งผลให้การทำงานของโพรโทคอลรวดเร็วยิ่งขึ้น

3.โพรโทคอลสามารถปรับเปลี่ยนรูปแบบการแพร่กระจายข้อมูลตามรูปแบบแผนที่บัฟเฟอร์ของเพื่อนบ้านเพื่อใช้ในการตัดสินใจการแพร่กระจายข้อมูลทำให้มีการตัดสินใจที่ดีมากขึ้น

3.2 หลักการทำงานของโพรโทคอล 2PMP

โพรโทคอลการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ ได้ถูกเผยแพร่ในงานทางวิชาการ โดยใช้ชื่อว่า “2PMP: A Push-Pull with Buffer-Map Prediction Algorithm for Peer to Peer Live Streaming” โดยมีหลักการทำงานที่สำคัญของ มีส่วนประกอบที่สำคัญดังต่อไปนี้

3.2.1 การสร้างโอเวอร์เลย์

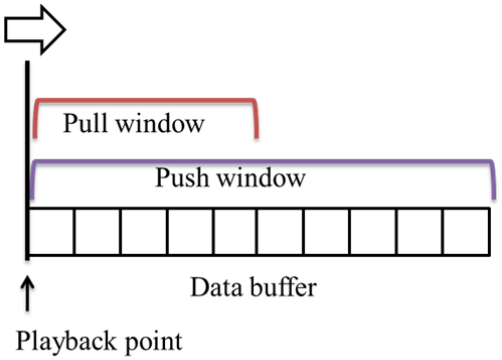
การสร้างโอเวอร์เลย์ เพียร์ที่ต้องการเข้าร่วมจะมีการติดต่อกับตัวกลางในการติดต่อระหว่างผู้ใช้งานในระบบ (Tracker) ซึ่งทำหน้าที่เก็บข้อมูลรายชื่อของเครื่องที่กำลังเชื่อมต่อในระบบเพียร์ทู

เพียร์ เมื่อได้รับข้อมูลรายชื่อมาก็จะทำการสุ่มเลือกเพียร์เพื่อทำการติดต่อในรูปแบบของโอเวอร์เลย์แบบตาข่าย

3.2.2 หน้าต่างแผนที่บัฟเฟอร์

ตารางที่ 3.1 ข้อมูลข้อความร้องขอชิ้นข้อมูล

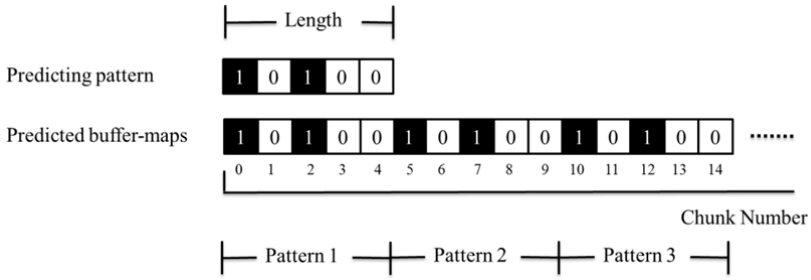
ชื่อ	รายละเอียด	ขนาดข้อมูล
Peer ID	หมายเลขเพียร์	4 bytes
Chunk ID	หมายเลขชิ้นข้อมูลที่ต้องการ	4 bytes



ภาพที่ 3.1 แสดงหน้าต่างผลักชิ้นข้อมูลและดึงชิ้นข้อมูล

หน้าต่างบัฟเฟอร์ ที่เป็นส่วนของการเก็บชิ้นข้อมูลก่อนที่จะทำการเล่นโดยมีการแบ่งเป็นสองส่วน คือ หน้าต่างที่ใช้สำหรับการผลักชิ้นข้อมูล และหน้าต่างที่ใช้สำหรับการดึงชิ้นข้อมูล ในส่วนของหน้าต่างผลักชิ้นข้อมูลเมื่อชิ้นส่วนข้อมูลใดอยู่ในส่วนนี้ก็จะมีการตัดสินใจผลักชิ้นส่วนข้อมูลนี้ไปให้เพื่อนบ้านโดยใช้การตัดสินใจจาก การคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้าน ในส่วนของหน้าต่างดึงชิ้นข้อมูลเมื่อชิ้นส่วนข้อมูลใดไม่มีอยู่ในส่วนนี้ก็จะมีการตัดสินส่งข้อความร้องขอชิ้นข้อมูล (Request Message) ดังตารางที่ 3.1 ออกไปยังเพื่อนบ้านที่มี โดยใช้การตัดสินใจจากการ การคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้าน โดยหน้าต่างบัฟเฟอร์จะมีการเคลื่อนที่ด้วยความเร็วเท่ากับอัตราการเล่น ดังภาพที่ 3.1

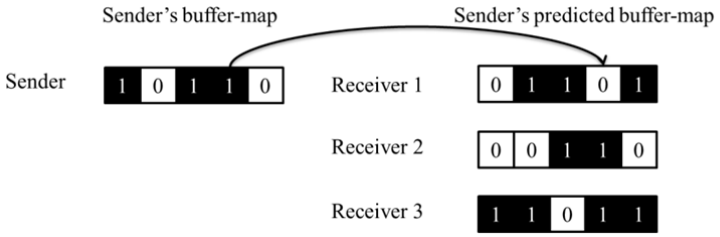
3.2.3 การคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้าน



ภาพที่ 3.2 การสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้าน

แผนที่บัฟเฟอร์ของเพื่อนบ้านจะมีการคาดเดาโดยใช้หน้าต่างรูปแบบ (Predicting pattern) ของแต่ละเพียร์ เพื่อทำการสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านแต่ละเพียร์ที่มีรูปแบบที่ซ้ำกันต่อไปเรื่อยๆโดยในช่วงเริ่มต้นการทำงานจะใช้การสุ่มเพื่อสร้างการคาดเดาแผนที่บัฟเฟอร์ (Predicted buffer-maps) ของเพื่อนบ้านแต่ละตัว ดังภาพที่ 3.2 เมื่อหน้าต่างรูปแบบได้มีการสุ่มว่าเพื่อนบ้าน มีข้อมูลลำดับที่ 0, 2 ดังนั้นก็จะมีการสร้างแผนที่การคาดเดาบัฟเฟอร์ของเพื่อนบ้านว่ามี 0, 2, 5, 7, 10, 12 ไปเรื่อยๆ เพื่อใช้ในการตัดสินใจแพร่กระจายข้อมูล

3.2.4 การแพร่กระจายชิ้นข้อมูลในหน้าต่างผลักชิ้นข้อมูล

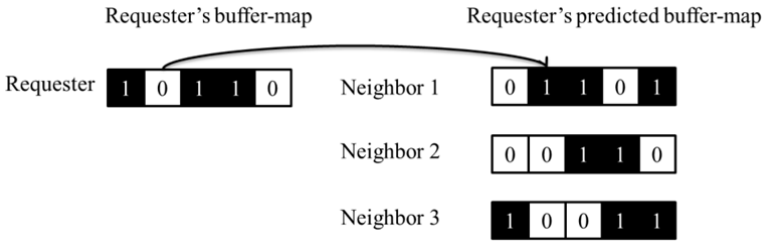


ภาพที่ 3.3 การเลือกเพียร์และชิ้นที่จะส่งข้อมูลแบบผลักรูปแบบ dp/lu

การแพร่กระจายชิ้นข้อมูลแบบในหน้าต่างผลักชิ้นข้อมูลจะใช้การเลือกเพียร์ก่อนแล้วถึงจะเลือกชิ้นข้อมูลที่จะส่ง โดยการเลือกเพียร์จะใช้รูปแบบการเลือกเพียร์ที่ขาดมากที่สุด (Most Deprives peer, dp) และมีการเลือกชิ้นที่มีประโยชน์มากสุด (Latest useful chunk, lu) เลือกชิ้นที่ผู้ส่งมีล่าสุดแต่ผู้รับยังไม่มีส่งไปให้ โดยใช้การตัดสินใจจากการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านแต่ละเพียร์ จากตัวอย่างดังภาพที่ 3.3 ที่ผู้ส่งทางด้านซ้ายมือ จะทำการเลือกเพียร์ที่จะส่งข้อมูลโดยใช้ข้อมูลที่คาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านแต่ละตัว โดยผู้รับที่ 1 มีชิ้นข้อมูลที่ 1, 2, 4 ในขณะที่ผู้ส่งข้อมูลมีชิ้นข้อมูลที่ 0, 2, 3 ดังนั้นจะมีจำนวนชิ้นที่ผู้รับขาดคือ 2 ชิ้น ต่อมาผู้รับที่ 2 มีชิ้นข้อมูลที่ 2, 3

ในขณะที่ผู้ส่งข้อมูลมีชิ้นข้อมูลที่ 0, 2, 3 ดังนั้นจะมีจำนวนชิ้นที่ผู้รับขาดคือ 1 ชิ้น ต่อมาผู้รับที่ 3 มีชิ้นข้อมูลที่ 0, 1, 3, 4 ในขณะที่ผู้ส่งข้อมูลมีชิ้นข้อมูลที่ 0, 2, 3 ดังนั้นจะมีจำนวนชิ้นที่ผู้รับขาดคือ 1 ชิ้น จากการคำนวณนี้จะทำการเลือกผู้รับที่ 1 ที่มีจำนวนที่ขาดมากสุด (Most Deprives peer, dp) จำนวน 2 ชิ้นและทำการเลือกชิ้นที่มีประโยชน์มากสุด (Latest useful chunk, lu) โดยเลือกชิ้นที่ 4 ที่ยังไม่มีในผู้รับที่ 1 ส่งไปให้ ซึ่งการส่งแบบนี้จากงานวิจัย Epidemic [7] แสดงให้เห็นว่าสามารถมีอัตราการแพร่ของข้อมูล (Optimal Diffusion Rate) สูงที่สุดและความหน่วงในการแพร่ข้อมูล (Optimal Diffusion Delay) น้อยที่สุด

3.2.5 การแพร่กระจายชิ้นข้อมูลในหน้าต่างดึงชิ้นข้อมูล



ภาพที่ 3.4 การเลือกเพียร์ที่จะส่งข้อมูลการร้องขอ

การแพร่กระจายชิ้นข้อมูลในหน้าต่างดึงชิ้นข้อมูลจะทำการเลือกเพียร์ที่จะส่งข้อความร้องขอชิ้นข้อมูล (Request Message) โดยใช้การตัดสินใจจากการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านที่คิดว่าเพื่อนบ้านนั้นมีชิ้นนั้นอยู่ ดังภาพที่ 3.4 เมื่อผู้ร้องขอไม่มีชิ้นข้อมูลที่ 1 ก็จะตัดสินใจโดยดูจากการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้าน โดยเพื่อนบ้านที่ 1 มีชิ้นข้อมูลที่ 1, 2, 4 เพื่อนบ้านที่ 2 มีชิ้นข้อมูลที่ 2, 3 และเพื่อนบ้าน 3 มีชิ้นข้อมูลที่ 0, 3, 4 ดังนั้น ผู้ร้องขอจะส่งข้อความร้องขอชิ้นข้อมูล (Request Message) ที่ 1 ไปยังเพื่อนบ้านที่ 1

3.2.6 การปรับการหน้าต่างรูปแบบ (Predicting pattern)

เนื่องจากในตอนเริ่มต้นการทำงานของระบบนั้น หน้าต่างรูปแบบ (Predicting pattern) ที่ใช้ในการสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านแต่ละเพียร์นั้น จะใช้การสุ่มดังนั้นเพื่อให้เกิดการตัดสินใจให้ดีมากขึ้นในการแพร่กระจายชิ้นข้อมูล จึงจำเป็นต้องมีการปรับหน้าต่างรูปแบบ (Predicting pattern) โดยการปรับจะมีส่วนประกอบที่สำคัญอยู่ 3 ส่วนด้วยกันคือ 1.การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบ (Predicting pattern)ให้คาดเดาว่ามีข้อมูล(แทนด้วย 1) 2.การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบ (Predicting pattern)ให้คาดเดาว่าไม่มีข้อมูล(แทนด้วย 0) 3.ช่วงข้อมูลในการปรับหน้าต่างรูปแบบ (Predicting pattern) เนื่องจากปรับหน้าต่างรูปแบบ (Predicting

pattern) จำเป็นต้องมีการส่งข้อความเพื่อควบคุม เพื่อให้เกิดประโยชน์สูงสุดจึงมีการกำหนดช่วงข้อมูลในการปรับหน้าต่างเพื่อให้เกิดประโยชน์สูงสุด

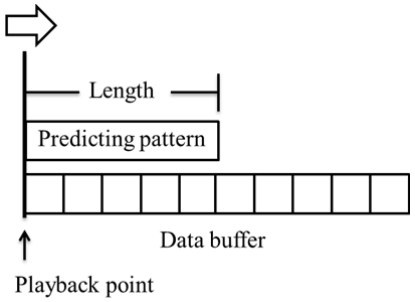
ตารางที่ 3.2 ข้อมูลข้อความแจ้งเตือน

ชื่อ	รายละเอียด	ขนาดข้อมูล
Peer ID	หมายเลขเพียร์	4 bytes
Array of Chunk ID	หมายเลขชิ้นข้อมูลที่ซ้ำ	4 bytes *Array size

1. การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบ (Predicting pattern) ให้คาดเดาว่ามีข้อมูล (แทนด้วย 1) เมื่อมีการได้รับชิ้นข้อมูล (Chunk) ที่มีข้อมูลซ้ำซ้อนกับชิ้นส่วนที่เรามีอยู่ และอยู่ในช่วงข้อมูลในการปรับหน้าต่างรูปแบบ (Predicting pattern) จะทำการส่ง ข้อความแจ้งเตือน (Notification message) ดังตารางที่ 3.2 ไปยังเพียร์ที่มีการส่งชิ้นข้อมูลซ้ำมา

เมื่อเพียร์ที่ส่งได้รับข้อความแจ้งเตือน (Notification message) นั้นมาก็จะรู้ว่าชิ้นส่วนข้อมูลที่ส่งไปนั้นมีการซ้ำซ้อนกันก็จะทำการปรับหน้าต่างรูปแบบ (Predicting pattern) โดยปรับตำแหน่งบิตที่ส่งชิ้นข้อมูลซ้ำนั้นจากคาดเดาว่าไม่มีข้อมูล(แทนด้วย 0) ไปเป็นคาดเดาว่ามีข้อมูล(แทนด้วย 1) และทำการสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านนั้นใหม่

2. การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบ (Predicting pattern) ให้คาดเดาว่าไม่มีข้อมูล (แทนด้วย 0) เมื่อมีการได้รับข้อความร้องขอชิ้นข้อมูล (Request message) ก็จะรู้ว่าเพียร์ที่ส่งมานั้นมีความต้องการหมายเลขชิ้นข้อมูลนั้น และอยู่ในช่วงข้อมูลในการปรับหน้าต่างรูปแบบ (Predicting pattern) ก็จะทำการปรับหน้าต่างรูปแบบ (Predicting pattern) ในตำแหน่งที่ได้รับข้อความร้องขอชิ้นข้อมูลไปเป็นไม่มีข้อมูล (แทนด้วย 0) และทำการสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านนั้นใหม่



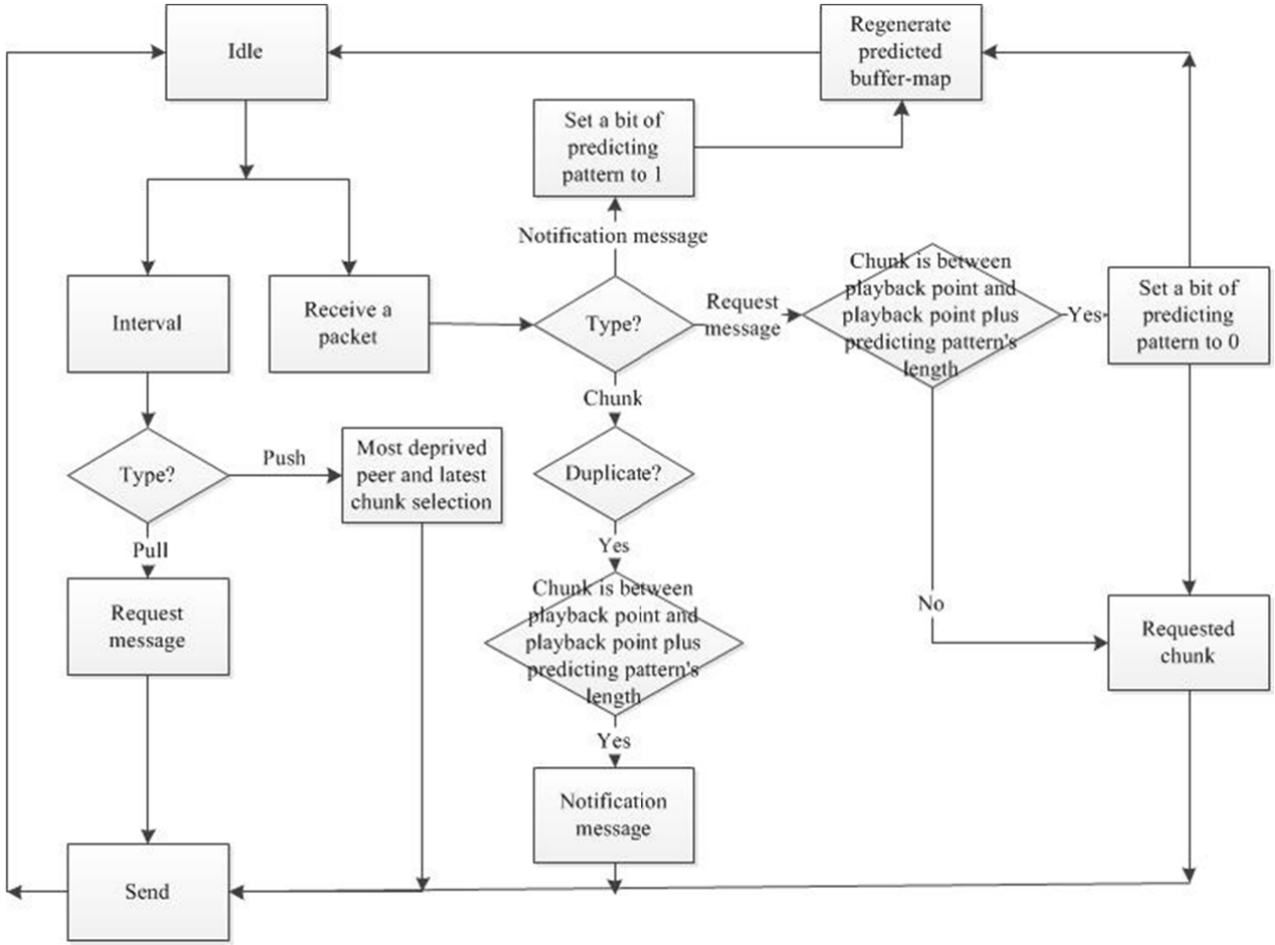
ภาพที่ 3.5 ช่วงข้อมูลในการปรับหน้าต่างรูปแบบ (Predicting pattern)

3.ช่วงข้อมูลในการปรับหน้าต่างรูปแบบ (Predicting pattern) ให้เกิดประโยชน์มากที่สุดจะมีการกำหนดช่วงข้อมูลในการปรับ

ในส่วนของการส่งข้อความแจ้งเตือน (Notification message) ถ้าชิ้นข้อมูลที่ได้ส่งมานั้นมีการซ้ำซ้อนกันอยู่ในช่วงตั้งแต่ข้อมูลที่ทำการเล่น (Playback point)ถึงข้อมูลที่ทำการเล่น (Playback point)บวกขนาดความยาวของหน้าต่างรูปแบบ (Predicting pattern) ดังภาพที่ 3.5 จะถึงมีการส่งข้อความแจ้งเตือน (Notification message) ไป แต่ถ้าไม่อยู่ในช่วงข้อมูลนี้จะไม่มีการส่งข้อความแจ้งเตือน (Notification message) กลับไปยังผู้ส่ง เพื่อปรับหน้าต่างรูปแบบ (Predicting pattern)

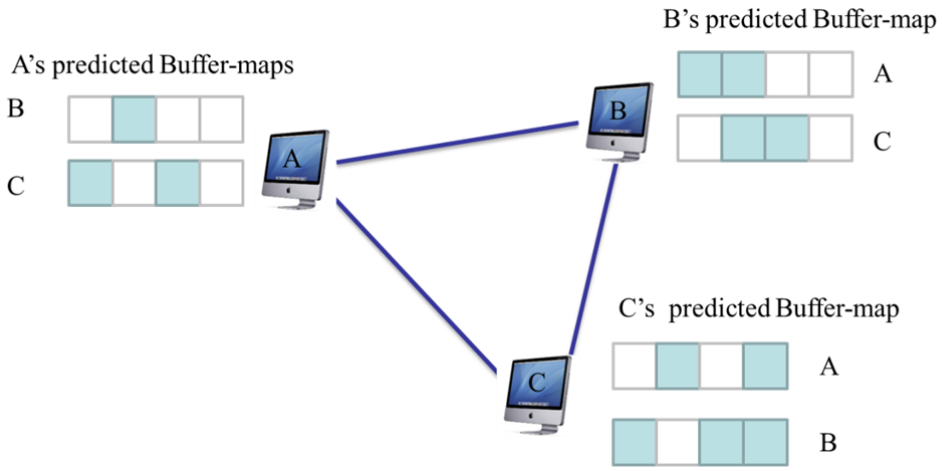
ต่อมาในส่วนของการได้รับข้อความร้องขอชิ้นข้อมูล (Request message) ถ้าอยู่ในช่วงตั้งแต่ข้อมูลที่ทำการเล่น (Playback point)ถึงข้อมูลที่ทำการเล่น (Playback point)บวกขนาดความยาวของหน้าต่างรูปแบบ (Predicting pattern)ถึงจะมีการปรับหน้าต่างรูปแบบ (Predicting pattern)

ขั้นตอนการทำงานของระบบสามารถเขียนเป็นผังการทำงานได้ดังภาพที่ 3.6



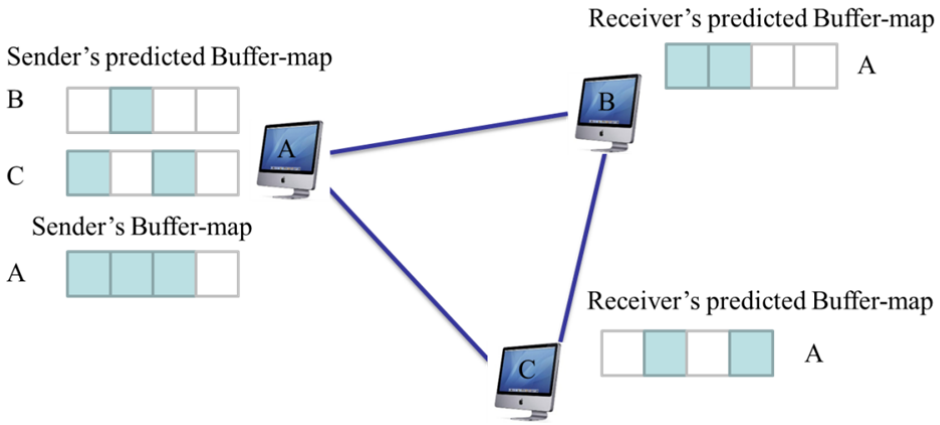
ภาพที่ 3.6 ผังการทำงานการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ 2PMP

3.3 ตัวอย่างการทำงานของโพรโทคอล 2PMP



ภาพที่ 3.7 ตัวอย่างการเริ่มโพรโทคอล

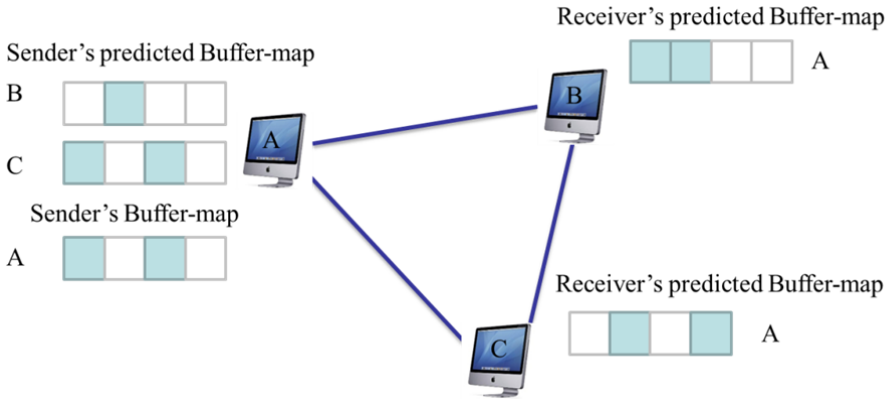
ในภาพที่ 3.7 เป็นการทำงานในการเริ่มต้นของโพรโทคอลเมื่อมีการสร้างโอเวอร์เลย์แล้ว จากนั้น เพียร์ A มีเพียร์ B และ C เป็นเพื่อนบ้าน ในกรณีนี้ เพียร์ A ก็จะทำการสร้าง Predicting pattern ของ B และ C เพื่อสร้างการคาดเดาแผนที่บัฟเฟอร์ของ B และ C



ภาพที่ 3.8 ตัวอย่างการแพร่กระจายข้อมูลโดยการผลัก

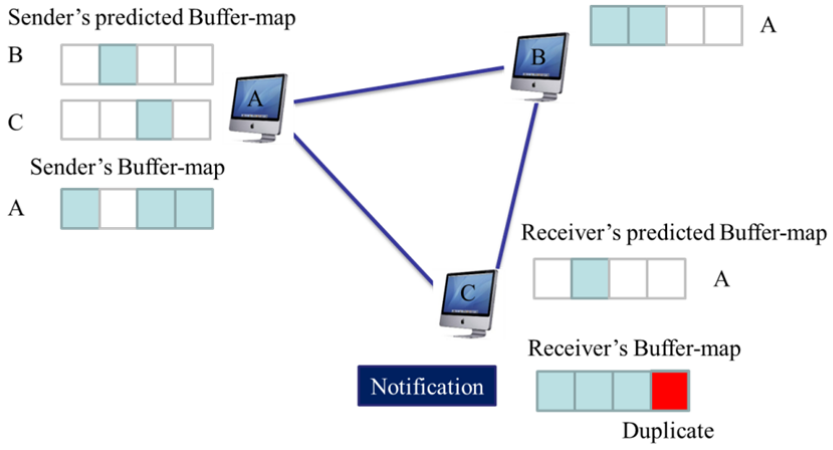
ในภาพที่ 3.8 แสดงการทำงานในกรณีที่หน้าต่างบัฟเฟอร์ที่เก็บชิ้นข้อมูลที่มีอยู่ของเพียร์ A มีข้อมูลชิ้นลำดับที่ 0, 1, 2 ตามลำดับ อยู่ในหน้าต่างผลักชิ้นข้อมูล (Push window) ก็จะทำการตัดสินใจในการเลือกเพียร์จะใช้รูปแบบการเลือกเพียร์ที่ขาดมากที่สุด (Most Deprives peer, dp) และมีการเลือกชิ้นที่มีประโยชน์มากสุด (Latest useful chunk, lu) โดยจากการคาดเดาแผนที่บัฟเฟอร์เมื่อเทียบกับผู้ส่ง ผู้รับ B มีชิ้นข้อมูลลำดับที่ 1 ทำให้มีการขาดชิ้นข้อมูล 2 ชิ้น ผู้รับ C มีชิ้น

ข้อมูลลำดับที่ 0, 2 ทำให้ขาดชิ้นข้อมูล 1 ชิ้น ทำให้มีการเลือกเพียร์ผู้รับเป็นเพียร์ B เนื่องจากเป็นเพียร์ที่ขาดมากที่สุด (Most Deprives peer, dp) จากนั้นจึงทำการเลือก ชิ้นที่มีประโยชน์มากสุด (Latest useful chunk, lu) ซึ่งก็คือชิ้นที่ 2 ในการส่งให้เพียร์ B



ภาพที่ 3.9 การแพร่กระจายข้อมูลโดยการดึง

ในภาพที่ 3.9 แสดงการทำงานในกรณีที่หน้าต่างบัฟเฟอร์ที่เก็บชิ้นข้อมูลที่มีอยู่ของเพียร์ A มีชิ้นข้อมูลลำดับที่ 0, 2 ตามลำดับ ไม่มีชิ้นข้อมูลลำดับที่ 1 อยู่ในหน้าต่างดึงชิ้นข้อมูล (Pull window) ก็จะทำการส่งข้อความร้องขอชิ้นข้อมูล (Request message) โดยใช้การตัดสินใจจาก การคาดเดาแผนที่บัฟเฟอร์โดย เพียร์ B มีชิ้นข้อมูลลำดับที่ 1 และ เพียร์ C มีชิ้นข้อมูลลำดับที่ 0, 1 ดังนั้นเพียร์ A จึงตัดสินใจส่งข้อความร้องขอชิ้นข้อมูล (Request message) ลำดับที่ 1 ไปยังเพียร์ B

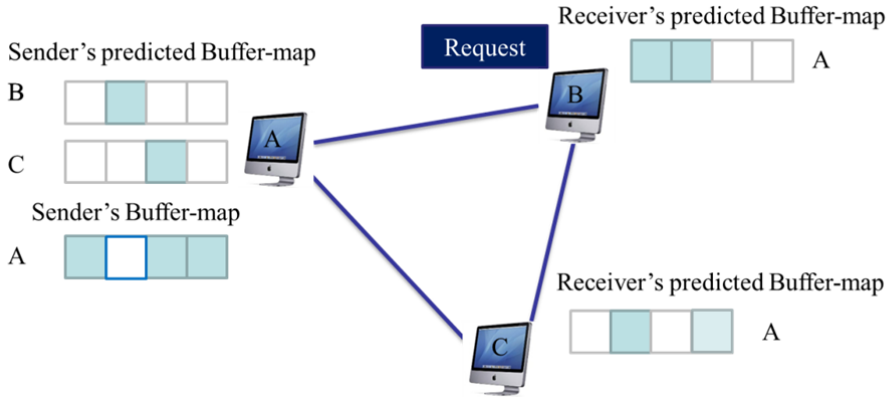


ภาพที่ 3.10 การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบให้คาดเดาว่ามีข้อมูล

ในภาพที่ 3.10 แสดงถึงการทำงานในกรณีที่ เพียร์ C ได้รับชิ้นข้อมูลลำดับที่ 3 จากเพียร์ A เมื่อเพียร์ C เคยได้รับชิ้นข้อมูลนี้แล้วทำให้เกิดข้อมูลซ้ำกัน จากนั้นจะทำการตรวจว่าอยู่ในช่วงข้อมูลที่ทำการเล่นถึงข้อมูลที่ทำการเล่นบวกขนาดความยาวของหน้าต่างรูปแบบ (Predicting pattern) จะทำ

การส่งข้อความแจ้งเตือน (Notification message) ไปยังเพียร์ A เพื่อแจ้งให้รู้ว่าชิ้นข้อมูลที่ส่งมาลำดับที่ 3 นั้นมีการส่งข้อมูลซ้ำซ้อน

เมื่อเพียร์ A ได้รับข้อความแจ้งเตือน (Notification message) ก็จะทำการปรับบิตในตำแหน่งที่ 3 ในหน้าต่างรูปแบบ (Predicting pattern) ของเพียร์ C ให้เป็นการคาดเดาว่ามีข้อมูลแล้วทำการสร้างการคาดเดาแผนที่บัฟเฟอร์ของ C ใหม่ตามรูปแบบของหน้าต่างรูปแบบ (Predicting pattern)



ภาพที่ 3.11 การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบให้คาดเดาว่าไม่มีข้อมูล

ในภาพที่ 3.11 แสดงถึงการทำงานในกรณีที่ เพียร์ B ได้รับข้อความร้องขอชิ้นข้อมูล (Request message) ลำดับที่ 1 จากเพียร์ A

เพียร์ B จะทำการปรับบิตในตำแหน่งที่ 1 ในหน้าต่างรูปแบบ (Predicting pattern) ของเพียร์ A ให้เป็นการคาดเดาว่าไม่มีข้อมูลแล้วทำการสร้างการคาดเดาแผนที่บัฟเฟอร์ของ A ใหม่ตามรูปแบบของหน้าต่างรูปแบบ (Predicting pattern)

3.4 การออกแบบโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น

จากการทดลองโพรโทคอล 2PMP พบว่าสามารถลดค่าใช้จ่ายที่เกิดจากการส่งข้อมูลควบคุมการทำงานได้อย่างน้อยร้อยละ 96 รายละเอียดการทดลองจะแสดงในบทที่ 4 แต่ จะยังมีการส่งข้อมูลซ้ำซ้อนกันที่มากอยู่เมื่อเปรียบเทียบกับโพรโทคอล Reinforcement ที่ออกแบบมาสำหรับการแก้ไขปัญหาการส่งข้อมูลที่ซ้ำซ้อนกัน ดังนั้น ดังนั้นงานวิจัยโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น มีการพัฒนาเพื่อเพิ่มประสิทธิภาพในการทำงาน ประกอบด้วย 4 ส่วนหลักดังต่อไปนี้

1.การปรับหน้าต่างรูปแบบจาก ข้อมูลที่ได้รับจากการผลัก โดยถือหลักการเดิมคือเชื่อถือตามรูปแบบที่เพื่อนบ้านกำหนดให้

2.การเพิ่มจำนวนการส่งข้อความร้องขอ

3.การใช้ขั้นตอนวิธีของ Reinforcement

4.เพิ่มขนาดของหน้าต่างรูปแบบ

การปรับการหน้าต่างรูปแบบ (Predicting pattern)

เนื่องจากในตอนเริ่มต้นการทำงานของระบบนั้น หน้าต่างรูปแบบ (Predicting pattern) ที่ใช้ในการสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านแต่ละเพียร์นั้น จะใช้การสุ่มดังนั้นเพื่อให้เกิดการตัดสินใจให้ดีมากขึ้นในการแพร่กระจายชิ้นข้อมูล จึงจำเป็นต้องมีการปรับหน้าต่างรูปแบบ (Predicting pattern) โดยการปรับจะมีส่วนประกอบที่สำคัญอยู่ 2 ส่วนด้วยกันคือ 1.การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบ (Predicting pattern)ให้คาดเดาว่ามีข้อมูล(แทนด้วย 1) 2.การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบ (Predicting pattern)ให้คาดเดาว่าไม่มีข้อมูล(แทนด้วย 0)

1. การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบ (Predicting pattern) ให้คาดเดาว่ามีข้อมูล (แทนด้วย 1) เมื่อมีการได้รับชิ้นข้อมูล (Chunk) เมื่อได้รับชิ้นข้อมูลที่ได้รับจากการผลักจากเพื่อนบ้านจะทำการปรับหน้าต่างรูปแบบ (Predicting pattern) ในตำแหน่งที่ได้รับชิ้นข้อมูลที่ได้รับจากการผลักไปเป็นมีข้อมูล (แทนด้วย 1) และทำการสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านนั้นใหม่

2. การปรับบิตที่ตำแหน่งในหน้าต่างรูปแบบ (Predicting pattern) ให้คาดเดาว่าไม่มีข้อมูล (แทนด้วย 0) เมื่อมีการได้รับข้อความร้องขอชิ้นข้อมูล (Request message) ก็จะรู้ว่าเพียร์ที่ส่งมานั้นมีความต้องการหมายเลขชิ้นข้อมูลนั้น ก็จะทำการปรับหน้าต่างรูปแบบ (Predicting pattern) ในตำแหน่งที่ได้รับข้อความร้องขอชิ้นข้อมูลไปเป็นไม่มีข้อมูล (แทนด้วย 0) และทำการสร้างการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านนั้นใหม่

การใช้ขั้นตอนวิธีการของ Reinforcement

เนื่องจากการทดลองพบว่าเพียร์ที่ส่งข้อมูลซ้ำส่วนใหญ่ก็จะมีการส่งข้อมูลซ้ำส่วนใหญ่ต่อไปเรื่อยๆดังนั้นจึงใช้ขั้นตอนวิธีการของ Reinforcement ในการลดการส่งข้อมูลซ้ำที่มาจากเพียร์เดิมโดยเมื่อมีการได้รับชิ้นข้อมูลที่ได้รับจากการผลัก จะมีการคำนวณค่าความน่าจะเป็นในการส่งข้อมูลที่ไม่มีการซ้ำซ้อน จาก จำนวนที่ได้รับไม่ซ้ำหารด้วยจำนวนที่ได้รับทั้งหมด แล้วทำการส่ง ข้อความน่าจะเป็น

ของการผลักข้อมูล (Push reinforcement message) ให้ผู้ส่ง จากนั้นผู้ส่งจะคำนวณความน่าจะเป็นที่ได้รับไปคูณกับขั้นในการปรับ(Step size) เพื่อลดความน่าจะเป็นในการผลักชิ้นข้อมูล

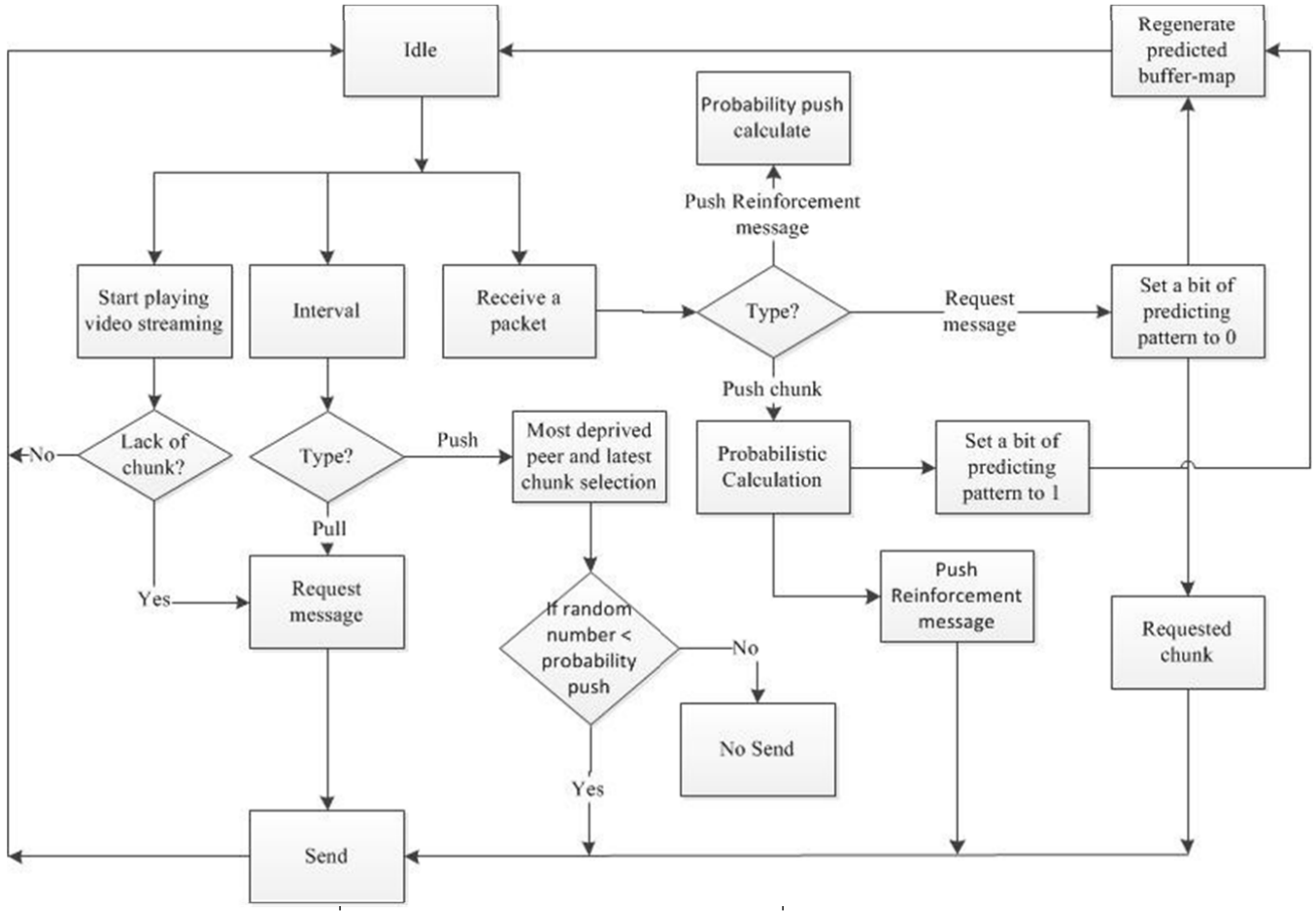
การเพิ่มจำนวนการส่งข้อความร้องขอ

เนื่องจากการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านนั้นเป็นการคาดเดาดังนั้นเพื่อเพิ่มชิ้นข้อมูลที่ได้รับจึงทำการเพิ่มการร้องขอข้อมูลเพื่อนบ้านอีกครั้งเพื่อให้มีโอกาศที่ได้รับมากขึ้น จากเดิมที่มีการร้องขอเพียงแค่เพียร์เดียวจากแผนที่บัฟเฟอร์ที่คาดเดาได้ และทำการเพิ่มร้องขออีกครั้งจากการสุ่มในชิ้นข้อมูลที่ยังไม่ได้รับจากเพื่อนบ้านเมื่อมีการเริ่มเล่นเกิดขึ้น

เพิ่มขนาดหน้าต่างของรูปแบบ

จากงานวิจัยเดิมไม่ค่อยมีการปรับหน้าต่างรูปแบบแค่กำหนดรูปแบบการส่งข้อมูล แต่งานวิจัยที่พัฒนาเพิ่มเติมนี้ต้องการให้มีการปรับหน้าต่างรูปแบบที่มากขึ้นทำให้เพื่อให้ข้อมูลที่ส่งมีความหลากหลายในการส่ง ซึ่งทำให้การปรับตำแหน่งที่ผิดไปเพียง 1 ตำแหน่ง ทำให้เกิด ความผิดพลาดได้มากขึ้น ตัวอย่างเมื่อมีข้อมูลทั้งหมด 2000 ชิ้น หน้าต่างแผนที่ 10 ชิ้น ถ้ามีการปรับพลาดไป 1 ตำแหน่งทำให้เกิดความผิดพลาดไปถึง 200 ตำแหน่ง โดยกำหนดให้เท่ากับระยะเวลาก่อนเริ่มเล่น (Playback Delay) หรือ 1000 ชิ้น

ขั้นตอนการทำงานของระบบสามารถเขียนเป็นผังการทำงานได้ดังภาพที่ 3.12



ภาพที่ 3.12 ผังการทำงานการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ 2PMP

บทที่ 4

ผลการทดลอง และวิเคราะห์ผลการทดลอง

ในบทนี้จะทำการวิเคราะห์สมรรถนะของโพรโทคอลการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ โดยเริ่มจากการกำหนดตัววัดสมรรถนะของระบบ (Performance Metrics) ในแต่ละด้านคือ ความเร็วของการแพร่กระจายข้อมูล ค่าใช้จ่าย การส่งข้อมูลที่ซ้ำกัน และความสามารถในการเล่นชิ้นข้อมูลในเวลา กำหนดเครื่องมือ โปรแกรมจำลองที่ใช้ และสภาพแวดล้อมในการทดลอง ผลการทดลอง เปรียบเทียบสมรรถนะระหว่างโพรโทคอลการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์

4.1 ตัววัดสมรรถะของโพรโทคอล (Performance Metrics)

ในงานวิจัยนี้ได้ทำการวัดสมรรถนะของโพรโทคอลใน 4 ด้าน คือ

1.ความเร็วของการแพร่กระจายข้อมูล (Cumulative distribution function) เป็นเกณฑ์ในการวัดความหน่วงของเวลาสำหรับการแพร่กระจายข้อมูลของระบบโดยให้สามารถมีอัตราการแพร่ของข้อมูลที่ดีและ ความหน่วงในการแพร่ข้อมูลน้อยที่สุด

2.ค่าใช้จ่าย (Overhead) โดยเกณฑ์ที่ใช้วัดประกอบไปด้วย แผนที่บัพเฟอร์ (Buffer-maps), ข้อความการร้องขอ (Request message), ข้อความน่าจะเป็นของการผลักข้อมูล (Push reinforcement message)[6] และข้อความบอก (Notification message) โดยวัตถุประสงค์หลักในการออกแบบโพรโทคอลนี้คือมีการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ที่มีค่าใช้จ่ายในการส่งข้อมูลในระดับต่ำ

3.การส่งข้อมูลที่ซ้ำกัน (Duplicate) เป็นเกณฑ์ในการวัดการส่งข้อมูลที่ได้รับซ้ำกัน

4.ความสามารถในการเล่นข้อมูลในเวลา (Play in time chunk) เป็นเกณฑ์ที่ใช้วัดประสิทธิภาพในส่งข้อมูลไลฟ์สตรีมมิ่ง ข้อมูลที่ส่งถึงแม้ว่าจะส่งสำเร็จแต่ถ้าไม่สามารถส่งได้ทันเวลาที่เริ่มเล่นก็ไม่สามารถมีประโยชน์ได้

4.2 เครื่องมือในการวัดสมรรถนะของโพรโทคอล

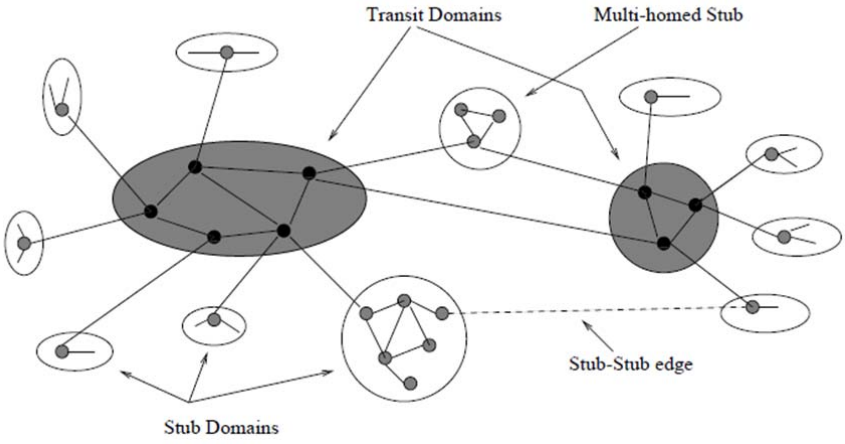
เนื่องจากโพรโทคอลถูกออกแบบเพื่อให้ทำงานในสภาพแวดล้อมที่มีจำนวนเพียร์มากมายในระบบ ดังนั้นเพื่อให้เป็นการจำลองสภาพการทดลองให้ได้ใกล้เคียงของจริงมากที่สุด ดังนั้นในการวัดสมรรถนะของโพรโทคอลจึงใช้โปรแกรมจำลอง ซึ่งประกอบด้วยโปรแกรมจำลอง 3 ส่วนคือ

1. โปรแกรมจำลอง NS-2.34(Network Simulation) [16] เป็นโปรแกรมจำลองลักษณะการทำงานของเครือข่ายที่ใกล้เคียงของจริง

2. โปรแกรมจำลอง GT-ITM (Georgia Tech Internetwork Topology Models)[17] เป็นโปรแกรมสร้างโทโพโลยี ที่จะสร้างจำลองสภาพแวดล้อมที่เสมือนจริงของเครือข่าย สามารถกำหนดรูปแบบของเครือข่าย ลักษณะของเครือข่าย ขนาดของเครือข่าย ขนาดช่องสัญญาณ และความหน่วงของเครือข่ายของการเชื่อมต่อ ทำให้มีลักษณะที่เสมือนจริงของเครือข่ายได้ดี

3. เฟรมเวิร์ก NS2-app [18] เป็นเฟรมเวิร์กชั้นแอพพลิเคชั่นที่ทำงานบนโปรแกรมจำลอง NS-2 สามารถพัฒนาชั้นแอพพลิเคชั่นได้สะดวกและถูกต้องได้ดีขึ้น

4.3 สภาพแวดล้อมที่ใช้ในการทดลอง



ภาพที่ 4.1 ลักษณะสภาพแวดล้อมที่ใช้ในการทดลอง

การทดลองนี้เป็นการทดลองในลักษณะของเครือข่ายที่มีรูปแบบเป็นตาข่ายเพื่อทดสอบการทำงานของโพรโทคอล บนสภาพแวดล้อมที่เสมือนจริงของเครือข่ายจะใช้ โปรแกรมจำลอง GT-ITM (Georgia Tech Internetwork Topology Models) [17] ในรูปแบบของ Transit-stub Model ดังภาพที่ 4.1 แล้วแปลงรูปแบบที่ได้ให้สามารถทำงานได้บนโปรแกรมจำลอง NS-2 โดยใช้โปรแกรม sgb2ns (sgb2ns Conversion Program) ทดลองระบบเครือข่ายแบบมีสาย โพรโทคอลที่ใช้ในการเปรียบเทียบมี ดังนี้

- Most Deprives peer and Latest useful chunk (dp/lu) + pull งานวิจัย Epidemic [7] ที่ใช้วิธีการแพร่กระจายข้อมูลด้วยวิธีการผลักที่สามารถมีอัตราการแพร่ของข้อมูลที่ดีและความหน่วงใน

การแพร่ข้อมูลน้อยที่สุด และเพื่อความเท่าเทียบกันในการเปรียบเทียบผลการทดลองจึงมีการเพิ่มการแพร่กระจายข้อมูลด้วยการดึงเข้าไปเพื่อความเท่าเทียมกันในการเปรียบเทียบผลการทดลอง

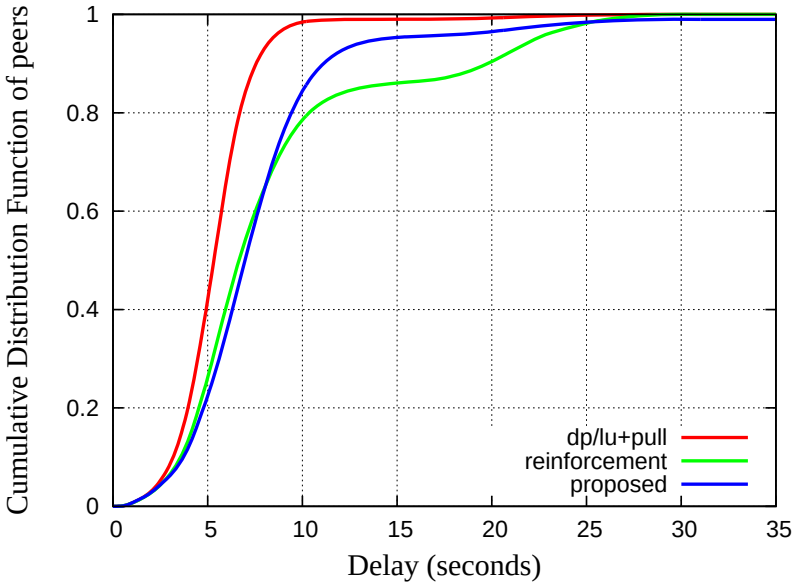
- A Reinforcement-Based Push-pull [6] งานวิจัยนี้ใช้วิธีการแพร่กระจายข้อมูลด้วยวิธีการผลักแบบ dp/lu ร่วมเข้ากับวิธีการดึง และแก้ไขปัญหาการส่งข้อมูลซ้ำซ้อนด้วยการลดอัตราการส่งข้อมูลที่ซ้ำซ้อน
- 2PMP เป็นงานวิจัยที่เสนอ ใช้วิธีการแพร่กระจายข้อมูลด้วยวิธีการผลักแบบ dp/lu ร่วมเข้ากับวิธีการดึง ที่ใช้การคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านในการตัดสินแทนข้อมูลบัฟเฟอร์เพื่อนบ้านจริง

ในการสร้างโอเวอร์เลย์แบบตาข่าย จะมีการสุ่มเลือกเพียร์เพื่อนบ้านประมาณ 10 เพียร์ ไม่มีข้อมูลการส่งที่สูญหายในระบบและการปรับตั้งค่าตัวแปรที่ใช้ในการทดลองอื่นๆเป็นในตารางที่ 4.1

ตารางที่ 4.1 แสดงตัวแปรที่ใช้ในการทดลอง

ตัวแปรที่ใช้ในการทดลอง	ค่าตัวแปร
จำนวนของโหนดในเครือข่าย	650 โหนด
จำนวนของเพียร์ในโอเวอร์เลย์	50, 100, 150, 200, 250 เพียร์
จำนวนชิ้นข้อมูล	2000 ชิ้น
ขนาดของชิ้นข้อมูล	1024 ไบต์
ขนาดหน้าต่างผลักชิ้นข้อมูล	1000 ชิ้น
ขนาดหน้าต่างดึงชิ้นข้อมูล	500 ชิ้น
ขนาดรูปแบบคาดเดา	5, 10, 15, 20, 25 ชิ้น
อัตราการสตริมมิ่ง	300 กิโลบิตต่อวินาที
อัตราการอัพโหลดข้อมูล	1.8 เมกะบิตต่อวินาที
อัตราการดาวโหลดข้อมูล	100 เมกะบิตต่อวินาที

4.4 ผลการทดลองความเร็วของการแพร่กระจายข้อมูลของโพรโทคอล



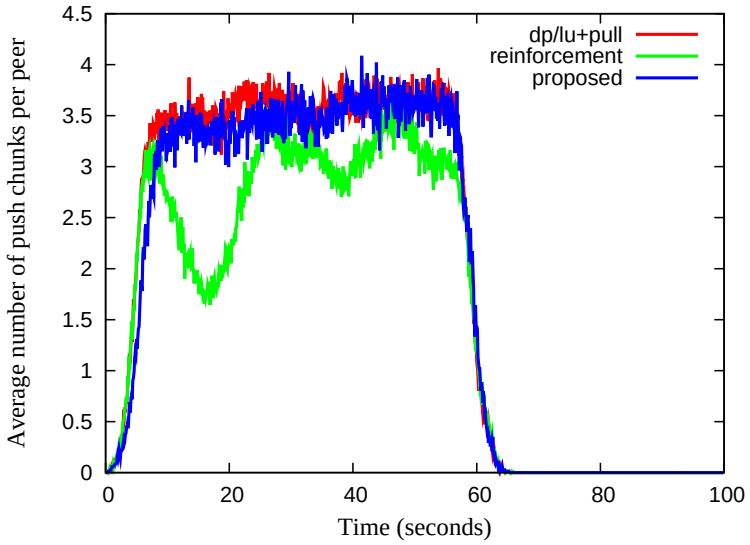
ภาพที่ 4.2 กราฟแสดงความเร็วของการแพร่กระจายข้อมูล

ผลการทดลองความเร็วในการกระจายข้อมูลโดยแสดงจำนวนเพียร์ที่ได้รับชิ้นข้อมูลต่อความหน่วงในการกระจายข้อมูลโดยแสดงผลการทดลองที่ 50 เพียร์และใช้ขนาดหน้าต่างรูปแบบที่ 10 ชิ้น

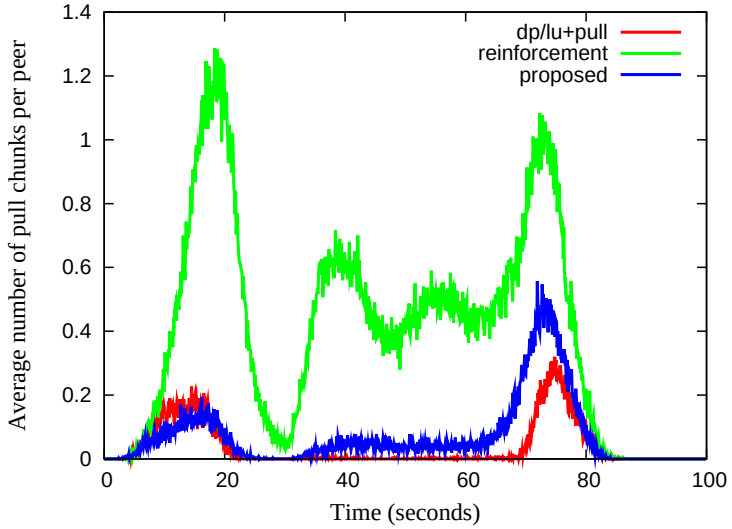
ในการทดลองพบว่า dp/lu+pull จะเป็นโพรโทคอลที่มีความเร็วการแพร่กระจายข้อมูลสูงที่สุดเสมอ เนื่องจากจะมีการแพร่กระจายข้อมูลทันทีที่ได้รับชิ้นส่วนข้อมูลใหม่ จากภาพที่ 4.3 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการผลักต่อเวลาจะเห็นได้ว่ามีจำนวนชิ้นข้อมูลที่ได้รับจากการผลักสูงที่สุด ซึ่งไม่มีความหน่วงในการแพร่กระจายข้อมูล และจากภาพที่ 4.4 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการดึงต่อเวลา จะเห็นได้ว่ามีจำนวนชิ้นข้อมลที่ได้รับจากการดึงต่ำที่สุด จึงมีความหน่วงในการแพร่กระจายข้อมูลที่ผู้ใช้งานต่ำ เป็นเหตุผลที่ทำให้มีการแพร่กระจายข้อมูลที่ดีที่สุดเมื่อเปรียบเทียบกับโพรโทคอลที่เหลือ

Reinforcement เป็นโพรโทคอลที่สามารถแพร่กระจายข้อมูลได้ช้าที่สุดเนื่องจากมีการลดอัตราการส่งข้อมูลเมื่อมีการส่งข้อมูลที่ซ้ำกันทำให้ จำนวนชิ้นข้อมูลเฉลี่ยที่ได้รับจากการแพร่กระจายโดยการผลักต่อเวลาจะเห็นได้ว่ามีจำนวนชิ้นข้อมูลที่ได้รับจากการผลักน้อยที่สุด ดังภาพที่ 4.3 ที่การแพร่กระจายรูปแบบนี้มีความหน่วงข้อมูลที่น้อยแต่มีการได้รับจากรูปแบบนี้ที่น้อยที่สุดเมื่อเทียบกับโพรโทคอลอื่น ในทางกลับกัน เมื่อดู ภาพที่ 4.4 ให้ จำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการดึงต่อเวลาจะเห็นได้ว่าสูงที่สุด จึงมีปัญหาเรื่องความหน่วงในการแพร่กระจายข้อมูลบางช่วงเวลา

โพรโทคอล 2PMP สามารถแพร่กระจายข้อมูลได้ใกล้เคียงและมีลักษณะรูปแบบเดียวกับ dp/lu+pull เนื่องจากการแพร่กระจายข้อมูลสามารถตัดสินใจส่งชิ้นส่วนข้อมูลได้ทันทีเมื่อมีข้อมูลชิ้นใหม่มา ดังภาพที่ 4.3 ที่มีการแพร่กระจายข้อมูลด้วยวิธีการผลักที่เกือบเท่ากับ dp/lu+pull ทำให้มีความหน่วงในการแพร่กระจายข้อมูลน้อย และ ภาพที่ 4.4 ที่มีการแพร่กระจายข้อมูลด้วยวิธีการดึงที่น้อยว่า Reinforcement ทำให้มีการแพร่กระจายข้อมูลที่ดีกว่า

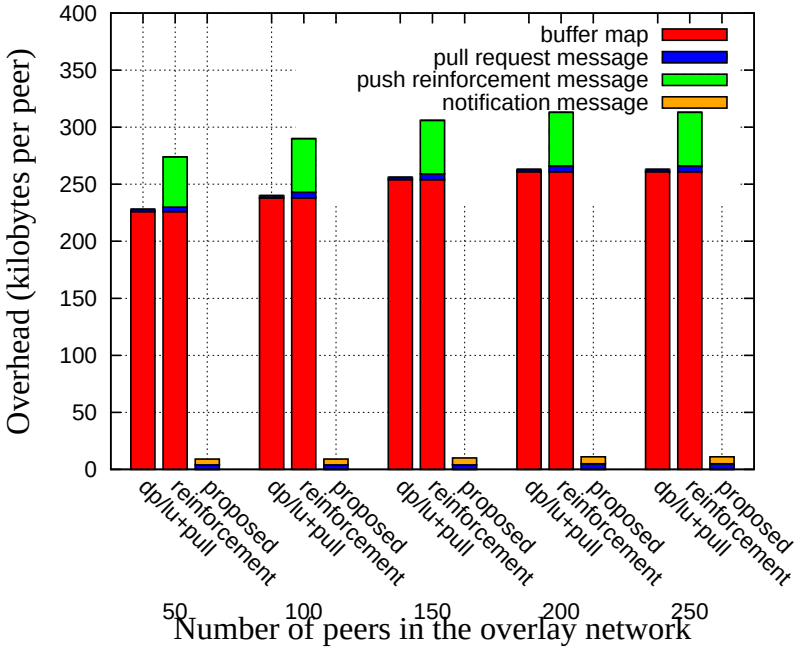


ภาพที่ 4.3 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการผลัก



ภาพที่ 4.4 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการดึง

4.5 ผลการทดลองค่าใช้จ่ายของโพรโทคอล



ภาพที่ 4.5 แสดงข้อมูลค่าใช้จ่ายที่เกิดขึ้นในระบบ

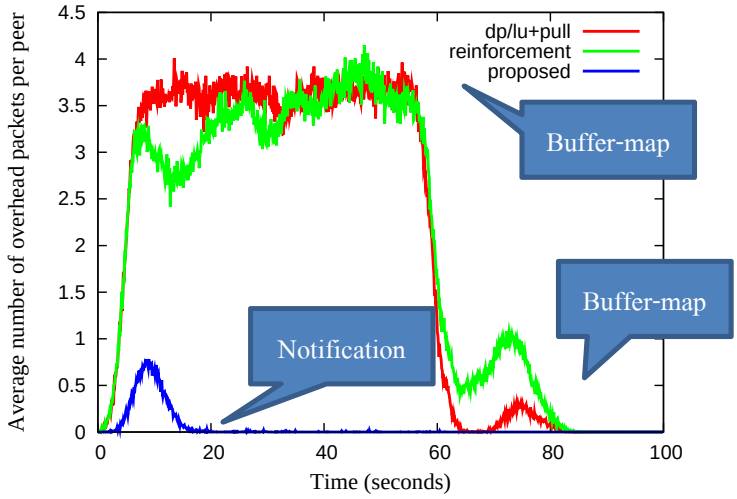
ผลการทดลองในเรื่องค่าใช้จ่ายที่ใช้ในการควบคุมการทำงานของโพรโทคอล ดังภาพที่ 4.5 คิดมากจาก แผนที่บัพเฟอร์ (Buffer-maps), ข้อความการร้องขอ (Request message), ข้อความน่าจะเป็นของการผลักข้อมูล (Push reinforcement message)[6] และข้อความแจ้งเตือน (Notification message) โดยแสดงผลการทดลองที่ 50, 100, 150, 200, 250 เพียร์และใช้ขนาดหน้าต่างรูปแบบที่ 10 ชิ้น

ในผลการทดลองพบว่าโพรโทคอล 2PMP นั้นสามารถมีค่าใช้จ่ายที่เกิดขึ้นนั้นต่ำที่สุด เนื่องจากประกอบไปด้วย ข้อความแจ้งเตือน(Notification message) และ ข้อความร้องขอชิ้นข้อมูล (Request message) ซึ่งตามโพรโทคอล ข้อความแจ้งเตือนจะมีการส่งข้อความไปเมื่อชิ้นข้อมูลที่ได้รับมานั้นอยู่ในช่วงระหว่างข้อมูลที่ทำการเล่น (Playback point)ถึงข้อมูลที่ทำการเล่น (Playback point)บวกขนาดความยาวของหน้าต่างรูปแบบ (Predicting pattern) ดังภาพที่ 4.6 จะมีการส่งข้อความแจ้งเตือน (Notification message) เฉพาะช่วงเวลาหนึ่งที่อยู่ในช่วงการปรับ ดังนั้นเมื่อเปรียบเทียบกับโพรโทคอลอื่นจะมีการส่งข้อมูลแผนที่บัฟเฟอร์เพื่อนบ้านตลอดเวลาที่มีการส่งชิ้นส่วนข้อมูลทำให้มีค่าใช้จ่ายที่เกิดขึ้นน้อยกว่ามาก

dp/lu+pull เป็นงานวิจัยของ Epidemic ที่ต้องการข้อมูลแผนที่บัพเฟอร์ของเพื่อนบ้านเพื่อใช้ในการตัดสินใจแพร่กระจายข้อมูลด้วยวิธีการผลักและการดึง มีค่าใช้จ่ายที่เกิดขึ้นสูงในลำดับที่สอง โดยมี

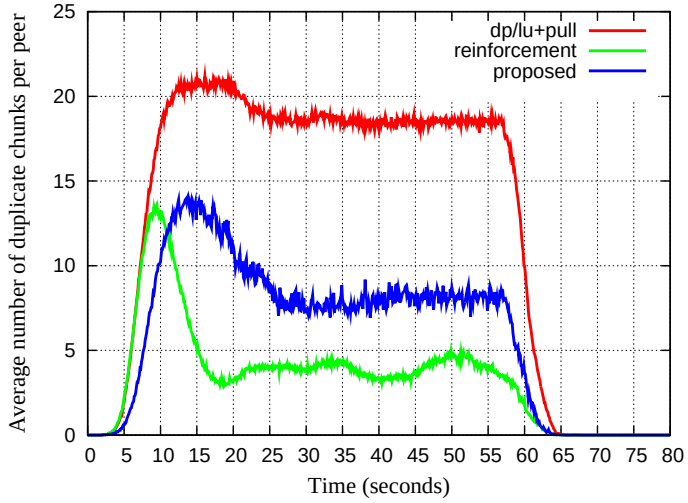
ประกอบไปด้วยข้อมูลแผนที่บัฟเฟอร์(Buffer-maps) และ ข้อความร้องขอชิ้นข้อมูล (Request message) จะมีการส่งข้อมูลแผนที่บัฟเฟอร์ทุกครั้งที่ได้รับชิ้นส่วนข้อมูลใหม่มา ดังภาพที่ 4.6

Reinforcement เป็นโพรโทคอลที่มีการใช้งาน ข้อมูลแผนที่บัฟเฟอร์ (Buffer-maps) ข้อความร้องขอชิ้นข้อมูล (Request message) และ ข้อความน่าจะเป็นของการผลักข้อมูล (Push reinforcement message) โดย ข้อความน่าจะเป็นของการผลักข้อมูล (Push reinforcement message) เป็นค่าใช้จ่ายที่เพิ่มขึ้นมาเพื่อใช้ในการส่งข้อมูลเพื่อลดอัตราการส่งข้อมูลที่ซ้ำกันและมีการส่งข้อมูลแผนที่บัฟเฟอร์ทุกครั้งที่ได้รับชิ้นส่วนข้อมูลใหม่มา ดังนั้นจึงมีค่าใช้จ่ายที่สูงกว่าโพรโทคอลอื่น ดังภาพที่ 4.6



ภาพที่ 4.6 แสดงจำนวนชิ้นข้อมูลค่าใช้จ่ายที่เกิดขึ้นในระบบ

4.6 ผลการทดลองการส่งข้อมูลที่ซ้ำกัน



ภาพที่ 4.7 การส่งชิ้นส่วนข้อมูลที่ซ้ำซ้อนกัน

ผลการทดลองในเรื่องการส่งข้อมูลที่ซ้ำกันในสภาพการทดลองนี้ที่เป็นระบบตาข่ายและใกล้เคียงกับสภาพเครือข่ายจริงจะมีปัญหาเรื่องความหน่วงในการได้รับข้อมูลแผนที่บัฟเฟอร์ที่ทันสมัยเพียงพอทำให้เกิดปัญหาในการตัดสินใจส่งข้อมูลที่ซ้ำซ้อนกันโดยแสดงผลการทดลองที่ 50 เพียร์และใช้ขนาดหน้าต่างรูปแบบที่ 10 ชิ้น

ในผลการทดลองพบว่า Reinforcement สามารถลดอัตราชิ้นส่วนข้อมูลซ้ำซ้อนกันได้มากที่สุด เนื่องจากเมื่อมีการตรวจพบการส่งข้อมูลที่ซ้ำซ้อนกันจะลดความน่าจะเป็นในการส่งให้เพียร์ที่ส่งข้อมูลซ้ำซ้อนกัน เป็นหลักกระบวนการของโพรโทคอลนี้ช่วยในการแก้ไขปัญหาในเรื่องนี้นี้จึงได้ประสิทธิภาพการลดชิ้นข้อมูลซ้ำที่ดีที่สุด

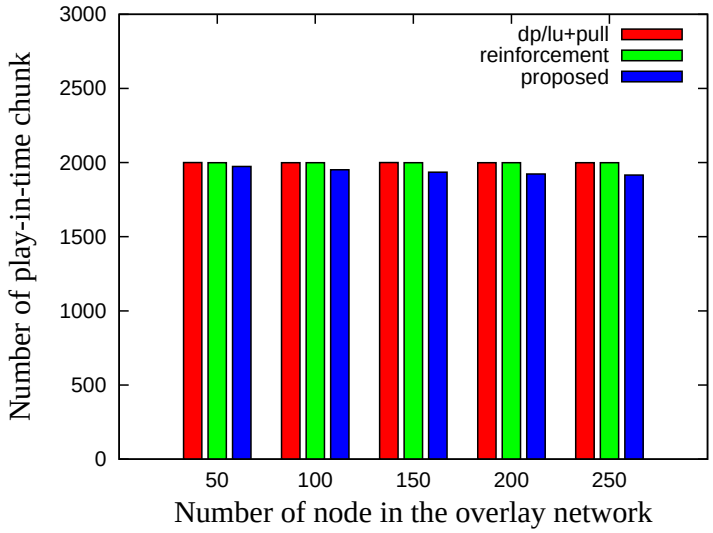
โพรโทคอล 2PMP สามารถลดได้ใกล้เคียงกับ โพรโทคอล Reinforcement เนื่องจากเมื่อดูข้อมูลผลการทดลอง ตัวอย่างหน้าต่างรูปแบบ(Predicting pattern) ที่เกิดขึ้นเมื่อเวลาผ่านไป 20 วินาที ดังภาพที่ 4.8 พบว่าแต่ละเพียร์ที่คาดเดาแผนที่บัพเฟอร์ของเพียร์ X จะมีการคาดเดาว่าเพียร์ว่าไม่ชิ้นข้อมูลลำดับที่แตกต่างกัน ทำให้ถ้ามีการตัดสินใจส่งข้อมูลไปให้ที่เพียร์ X จะไม่ส่งข้อมูลที่ซ้ำซ้อนกัน ดังภาพที่ 4.8 เป็นผลทำให้ส่งชิ้นข้อมูลไปให้ลดความซ้ำซ้อนกัน

dp/lu+pull เป็นงานวิจัยที่เน้นการแพร่กระจายข้อมูลได้รวดเร็วแต่ในการทดลองสภาพแวดล้อมที่มีความหน่วงในระบบ ทำให้เกิดปัญหาเรื่องข้อมูลแผนที่บัฟเฟอร์ของเพื่อนบ้านที่ทันสมัยเพียงพอ ทำให้การตัดสินใจส่งข้อมูลมีการซ้ำซ้อนกันมากที่สุด

Time	Peer	Predict	Predicting Pattern
20	A	X	1 1 1 1 1 1 1 1 1 1
20	B	X	1 1 1 1 1 1 0 1 1 1
20	C	X	1 1 1 1 1 1 1 1 1 1
20	D	X	1 1 1 1 1 0 1 1 1 1
20	E	X	1 1 1 1 1 1 1 0 1 1
20	F	X	1 1 1 1 1 1 1 1 0 1
20	H	X	1 1 1 0 1 1 1 1 1 1
20	I	X	1 1 1 1 0 1 1 1 1 1
20	J	X	0 1 0 1 1 1 1 1 1 1
20	K	X	1 0 1 1 1 1 1 1 1 0

ภาพที่ 4.8 ตัวอย่างหน้าต่างรูปแบบที่คาดเดาแผนที่บัฟเฟอร์เพียร์ X เมื่อเวลา 20 วินาทีในขนาด 10 ชิ้นข้อมูล

4.7 ผลการทดลองความสามารถในการเล่นชิ้นข้อมูลในเวลา



ภาพที่ 4.9 แสดงจำนวนชิ้นข้อมูลที่สามารถเล่นได้ในเวลา

ผลการทดลองในเรื่องการส่งชิ้นข้อมูลที่สามารถเล่นได้ในเวลานั้น เมื่อทดลองวิธีการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ที่มีการส่งชิ้นข้อมูลจำนวน 2000 ชิ้น โดยแสดงผลการทดลองที่ 50 ,100 ,150, 200, 250 เพียร์และใช้ขนาดหน้าต่างรูปแบบที่ 10 ชิ้น

ในผลการทดลองพบว่า dp/lu+pull สามารถเล่นข้อมูลตามจำนวนชิ้นที่ส่งได้ทั้งหมด แม้จะเพิ่มจำนวนเพียร์ในระบบมากขึ้น ก็ยังสามารถเล่นได้ไม่สูญหาย เนื่องจากมีข้อมูลแผนที่บัฟเฟอร์ของเพื่อน

บ้านที่ถูกต้อง ถ้ามีข้อมูลชิ้นส่วนที่ยังไม่มี ก็สามารถส่งข้อความร้องขอ(Request message) ไปยังเพื่อนบ้านเพื่อขอชิ้นส่วนนั้นมาได้ ทำให้สามารถมีชิ้นส่วนข้อมูลได้ครบ

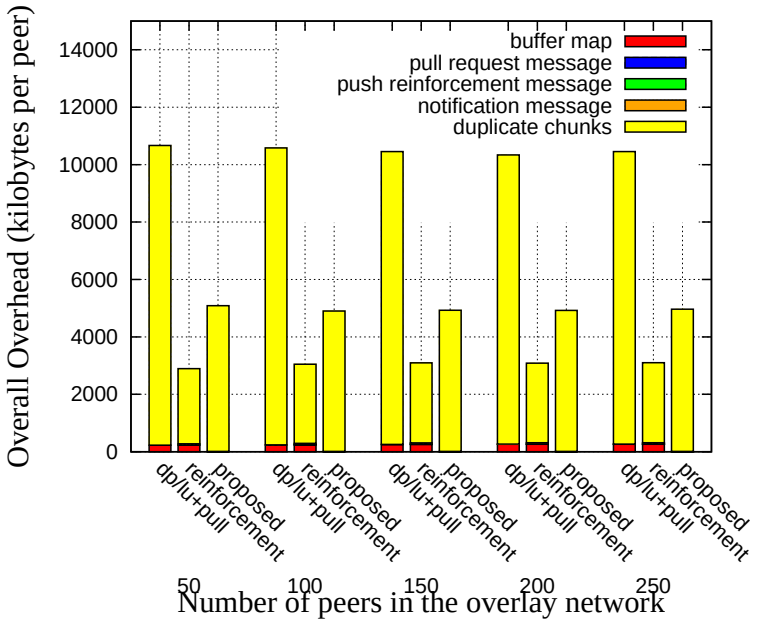
โพรโทคอล Reinforcement พบว่าสามารถเล่นข้อมูลตามจำนวนชิ้นที่ส่งได้ทั้งหมด ถึงแม้ว่าโพรโทคอลจะมีการลดการส่งชิ้นส่วนข้อมูลซ้ำ ทำให้บางช่วงมีการขาดชิ้นข้อมูลแต่ มีข้อมูลแผนที่บัฟเฟอร์เพื่อนบ้านที่ถูกต้อง จึงสามารถส่งข้อความร้องขอ(Request message) ไปยังเพื่อนบ้านเพื่อขอชิ้นส่วนที่ขาดไปมาได้ ทำให้สามารถมีชิ้นส่วนข้อมูลได้ครบ

โพรโทคอล 2PMP นั้นพบว่าสามารถเล่นข้อมูลตามจำนวนชิ้นที่ส่งได้ดีอยู่ เนื่องจากถึงแม้ว่าข้อมูลในการตัดสินใจผลักชิ้นข้อมูลหรือดึงชิ้นข้อมูลจะใช้การคาดเดาแผนที่บัฟเฟอร์แต่จาก ภาพที่ 4.8 ก็มีสามารถส่งชิ้นข้อมูลด้วยการแพร่กระจายแบบผลักไปเกือบทุกลำดับชิ้นข้อมูล หรือสามารถส่งข้อความร้องขอ(Request message) ไปยังเพื่อนบ้านเพื่อเพิ่มการได้รับชิ้นข้อมูลนั้น มาทำให้มีชิ้นข้อมูลที่เพิ่มขึ้นมา ซึ่งจากการทดลองเมื่อมีจำนวนเพียร์ในโอเวอร์เลย์ 50 เพียร์มีการสูญเสียเพียง 26 ชิ้น หรือ ร้อยละ 1.25 เท่านั้น

ตารางที่ 4.2 ข้อมูลจำนวนเฉลี่ยชิ้นข้อมูลที่สามารถเล่นได้ในเวลา

เพียร์ในโอเวอรเลย์	เฉลี่ยจำนวนที่สามารถเล่นได้ในเวลา	ร้อยละที่ไม่สามารถเล่นได้ในเวลา
50	1974	1.25
100	1952	2.35
150	1935	3.20
200	1923	3.80
250	1916	4.15

4.8 ผลการทดลองค่าใช้จ่ายของโพรโทคอลทั้งหมด



ภาพที่ 4.10 แสดงข้อมูลค่าใช้จ่ายทั้งหมดที่เกิดขึ้นในระบบ

ผลการทดลองในเรื่องค่าใช้จ่ายทั้งหมดที่เกิดขึ้นในระบบเมื่อนำค่าใช้จ่ายที่เกิดขึ้นจากการส่งข้อมูลควบคุมการทำงาน กับ ค่าใช้จ่ายที่เกิดจากการส่งข้อมูลซ้ำซ้อนกัน โดยแสดงผลการทดลองที่ 50, 100, 150 , 200, 250 เพียร์และใช้ขนาดหน้าต่างรูปแบบที่ 10 ชิ้น

ในการทดลองพบว่า reinforcement สามารถมีค่าใช้จ่ายทั้งหมดที่เกิดขึ้นน้อยที่สุด เนื่องมาจากปัญหาความหน่วงในระบบทำให้เกิดการส่งข้อมูลที่ซ้ำซ้อนกันโดยแต่ละชิ้นข้อมูลที่ส่งมีขนาดถึง 1024 ไบต์ ทำให้เกิดค่าใช้ จ่ายในส่วนนี้สูงมาก เมื่อสามารถลดการส่งข้อมูลที่ซ้ำซ้อนกันส่งผลให้ค่าใช้จ่ายทั้งหมดที่เกิดขึ้นน้อยที่สุด

โพรโทคอล 2PMP นั้นพบว่าสามารถมีค่าใช้จ่ายที่เกิดขึ้นในระบบที่รองลงมาเนื่องมาจากสามารถลดค่าใช้จ่ายท่เกิดขึ้นจากการส่งข้อมูลควบคุม และ การส่งข้อมูลที่ซ้ำซ้อนกันในระบบ

โพรโทคอล dp/lu+pull มีค่าใช้จ่ายที่เกิดขึ้นในระบบสูงที่สุด เนื่องมากจากในระบบเครือข่ายที่มีความหน่วงจะมีการส่งข้อมูลที่มีความซ้ำซ้อนกันอยู่ในระดับสูง อีกทั้งต้องมีการส่งข้อมูลควบคุมการทำงานโดยแลกเปลี่ยนแผนที่บัฟเฟอร์ทุกครั้งที่มีการส่งข้อมูลใหม่มา ทำให้เกิดค่าใช้จ่ายยู่ในระดับสูง

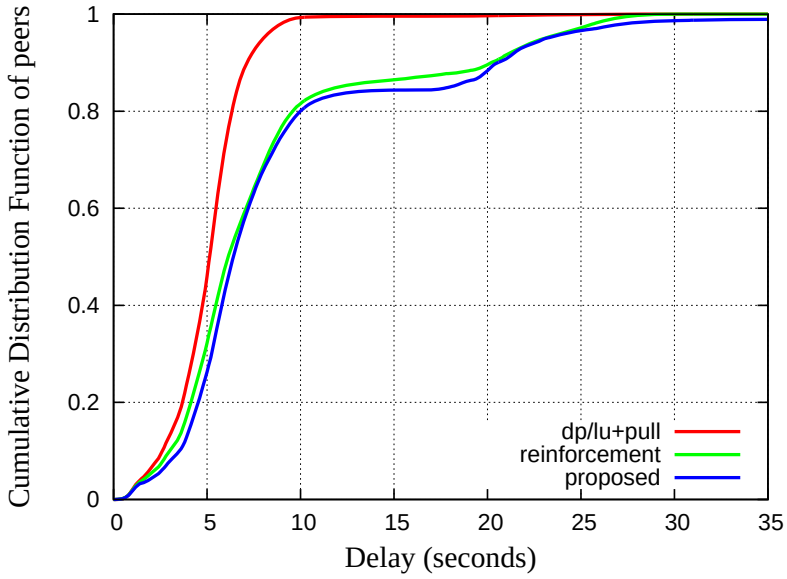
4.9 ผลการทดลองโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น

จากปัญหาในเรื่องของค่าใช้จ่ายของโพรโทคอลทั้งหมดที่เกิดจากการการส่งข้อมูลที่ซ้ำซ้อนกันทำให้มีการพัฒนางานวิจัยเพิ่มเติมเพื่อเพิ่มประสิทธิภาพในการทำงานในรายละเอียดที่กล่าวไว้ในบทที่ 3โดยมีการปรับตั้งค่าตัวแปรที่ใช้ในการทดลองโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น เป็นในตารางที่ 4.3 โดยมีการใช้ขนาดรูปแบบคาดเดา (Predicting pattern) ที่ 1000 ชิ้นเนื่องจากเมื่อมีการปรับหน้าต่างรูปแบบตามเวลาเมื่อเวลาเปลี่ยนไปมากขึ้น ดังนั้นถ้ากำหนดขนาดเล็กจะทำให้เกิดความผิดพลาดต่อเนื่องได้จำนวนมาก

ตารางที่ 4.3 แสดงตัวแปรที่ใช้ในการทดลองเพิ่มเติม

ตัวแปรที่ใช้ในการทดลอง	ค่าตัวแปร
จำนวนของโหนดในเครือข่าย	650 โหนด
จำนวนของเพียร์ในโอเวอร์เลย์	50, 100, 150, 200, 250 เพียร์
จำนวนชิ้นข้อมูล	2000 ชิ้น
ขนาดของชิ้นข้อมูล	1024 ไบต์
ขนาดหน้าต่างผลักชิ้นข้อมูล	1000 ชิ้น
ขนาดหน้าต่างดึงชิ้นข้อมูล	500 ชิ้น
ขนาดรูปแบบคาดเดา	1000 ชิ้น
อัตราการสตริมมิ่ง	300 กิโลบิตต่อวินาที
อัตราการอัพโหลดข้อมูล	1.8 เมกะบิตต่อวินาที
อัตราการดาวโหลดข้อมูล	100 เมกะบิตต่อวินาที

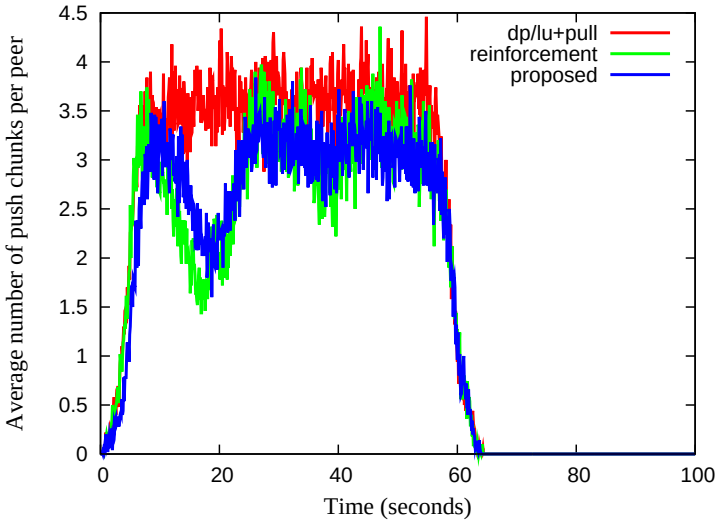
ผลการทดลองความเร็วของการแพร่กระจายข้อมูลของโพรโทคอล



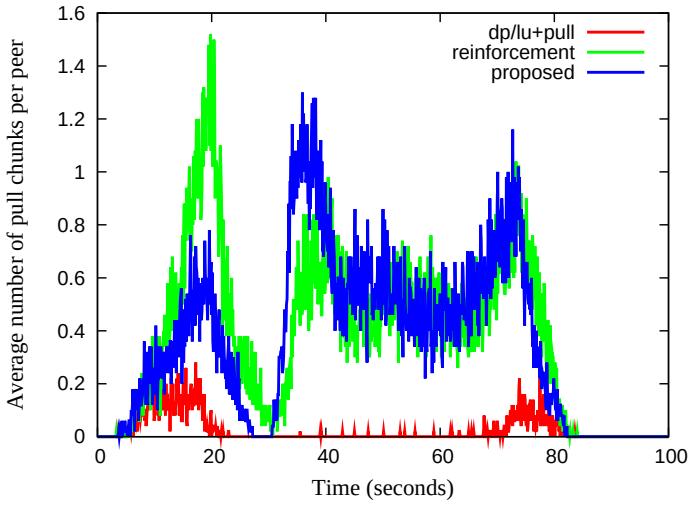
ภาพที่ 4.11 กราฟแสดงความเร็วของการแพร่กระจายข้อมูลใน 2PMP แบบใช้ความน่าจะเป็น

จากการทดลองความเร็วของการแพร่กระจายข้อมูลในโพรโทคอลเพิ่มเติมที่แสดงจำนวนที่เพียร์ได้รับชิ้นข้อมูลต่อความหน่วงในการแพร่กระจายข้อมูล

โพรโทคอล 2PMP แบบใช้ความน่าจะเป็น ดังภาพที่ 4.11 พบว่าสามารถแพร่กระจายข้อมูลได้เหมือนกับ Reinforcement เนื่องมาจากการต้องมีการลดอัตราการส่งข้อมูล ทำให้ชิ้นข้อมูลที่ได้มีอัตราการแพร่กระจายที่ลดลง ดังภาพที่ 4.12 ที่มีการแพร่กระจายโดยการผลักชิ้นข้อมูลที่ลดลงบางช่วง และ จากภาพที่ 4.13 แสดงให้เห็นถึงการแพร่กระจายข้อมูลด้วยวิธีการดึงที่มีการใช้มากขึ้นเมื่อเปรียบเทียบกับโพรโทคอล 2PMP ทำให้มีความหน่วงในการแพร่กระจายมากขึ้นตามผลการทดลองที่แสดงออกมา

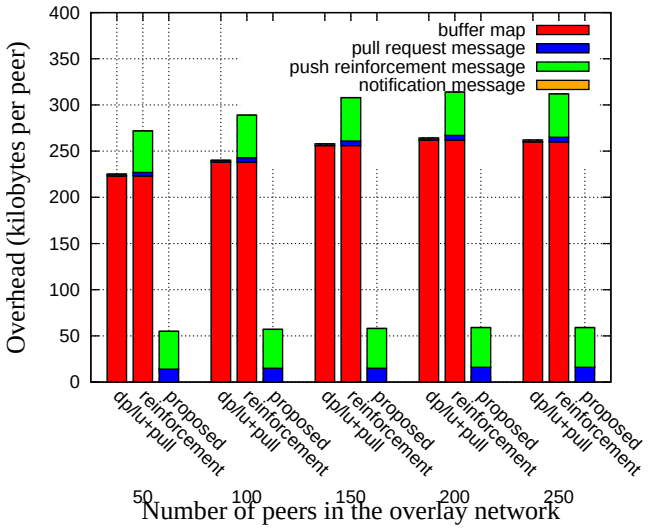


ภาพที่ 4.12 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการผลักใน 2PMP แบบใช้ความน่าจะเป็น



ภาพที่ 4.13 แสดงข้อมูลจำนวนชิ้นข้อมูลเฉลี่ยที่การแพร่กระจายโดยการดึงใน 2PMP แบบใช้ความน่าจะเป็น

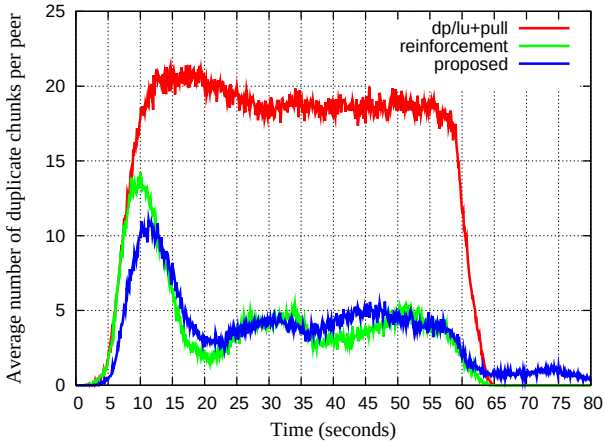
ผลการทดลองค่าใช้จ่ายของโพรโทคอล



ภาพที่ 4.14 แสดงข้อมูลค่าใช้จ่ายที่เกิดขึ้นในระบบใน 2PMP แบบใช้ความน่าจะเป็น

ผลการทดลองในเรื่องค่าใช้จ่ายที่ใช้ในการควบคุมการทำงานของโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น จากการทดลองดังภาพที่ 4.14 พบว่า มีค่าใช่จ่ายที่สูงขึ้นกว่าโพรโทคอลแบบเดิม ที่ลดได้อย่างน้อยร้อยละ 96 เป็นลดได้อย่างน้อยร้อยละ 76 เนื่องมาจากมีการเพิ่มอัตราการส่งข้อความการร้องขอ(Request message) มากขึ้น รวมถึง มีการใช้ขั้นตอนวิธีการของ Reinforcement โดยมีการใช้ข้อความน่าจะเป็นของการผลักข้อมูล (Push reinforcement message) เพื่อใช้ในการลดจำนวนส่งข้อมูลที่ซ้ำกัน

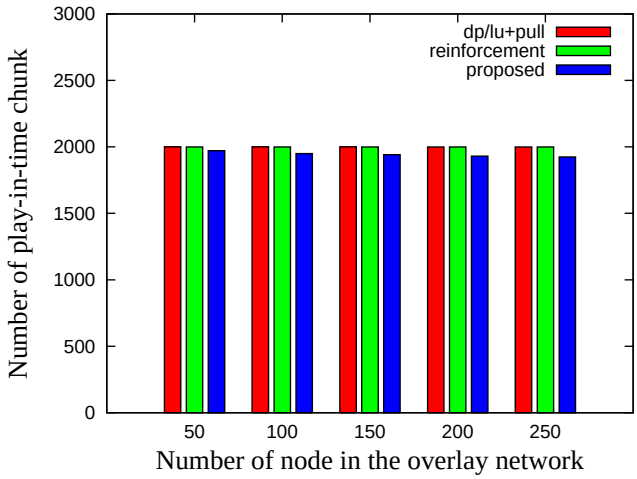
ผลการทดลองการส่งข้อมูลที่ซ้ำกัน



ภาพที่ 4.15 การส่งชิ้นส่วนข้อมูลที่ซ้ำซ้อนกันใน 2PMP แบบใช้ความน่าจะเป็น

ผลการทดลองในเรื่องการส่งข้อมูลที่ซ้ำกันนั้น ดังภาพที่ 4.15 พบว่า โพรโทคอล 2PMP แบบใช้ความน่าจะเป็น นั้นสามารถลดอัตราการส่งข้อมูลซ้ำได้ใกล้เคียงกับ Reinforcement เนื่องมาจากมีการใช้ขั้นตอนวิธีการของโพรโทคอล Reinforcement เพราะว่าจะมีการส่งข้อความน่าจะเป็นของการผลักข้อมูล (Push reinforcement message) เพื่อปรับอัตราการส่งข้อมูลของเพียร์ที่ซ้ำกันทำให้สามารถลดการส่งชิ้นข้อมูลที่ซ้ำกัน

ผลการทดลองความสามารถในการเล่นชิ้นข้อมูลในเวลา



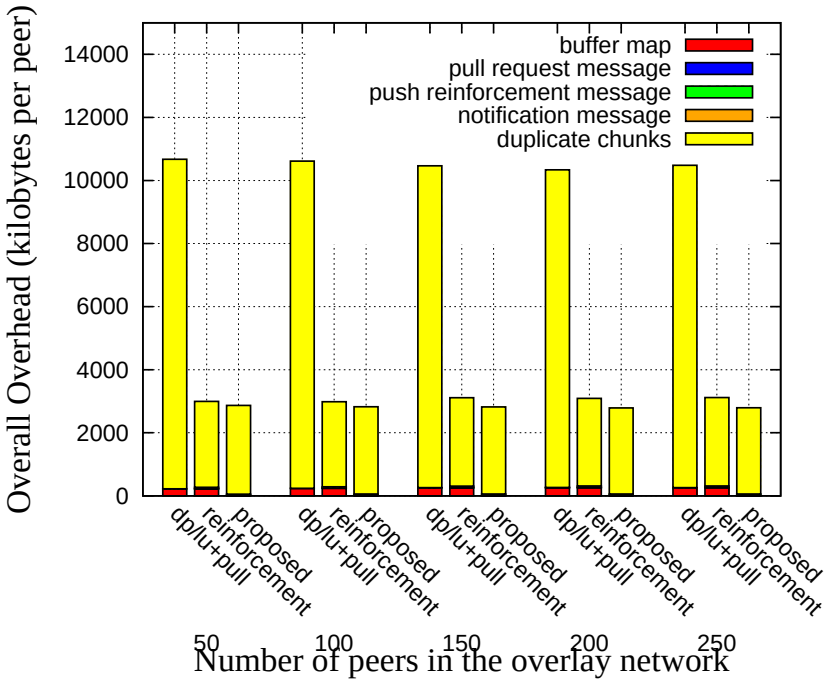
ภาพที่ 4.16 แสดงจำนวนชิ้นข้อมูลที่สามารถเล่นได้ในเวลาใน 2PMP แบบใช้ความน่าจะเป็น

ผลการทดลองในเรื่องการส่งชิ้นข้อมูลที่สามารถเล่นได้ในเวลา จากภาพที่ 4.16 พบว่าโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น นั้นสามารถเพิ่มอัตราการได้รับชิ้นข้อมูลเมื่อเปรียบเทียบกับโพรโทคอล 2PMP เนื่องมาจากมีการส่งข้อความร้องขอ (Request message) มากขึ้นทำให้ได้รับชิ้นข้อมูลที่มากขึ้น ดังตารางที่ 4.4

ตารางที่ 4.4 ข้อมูลจำนวนเฉลี่ยชิ้นข้อมูลที่สามารถเล่นได้ในเวลาใน 2PMP แบบใช้ความน่าจะเป็น

เพียร์ในโอเวอรเลย์	เฉลี่ยจำนวนที่สามารถเล่นได้ในเวลา	ร้อยละที่ไม่สามารถเล่นได้ในเวลา
50	1978	1.05
100	1960	1.95
150	1950	2.45
200	1940	2.95
250	1934	3.25

ผลการทดลองค่าใช้จ่ายทั้งหมดของโพรโทคอล



ภาพที่ 4.17 แสดงข้อมูลค่าใช้จ่ายโดยรวมที่เกิดขึ้นในระบบใน 2PMP แบบใช้ความน่าจะเป็น

จากผลการทดลองเมื่อคิดค่าใช้จ่ายโดยรวมที่เกิดขึ้นในระบบโดยมีการรวมค่าใช่จ่ายที่เกิดขึ้นจากการส่งข้อมูลควบคุมการทำงานเข้ากับการส่งข้อมูลที่ซ้ำซ้อนกัน ดังภาพที่ 4.17 พบว่าโพรโทคอล 2PMP แบบใช้ความน่าจะเป็น สามารถลดค่าใช้จ่ายได้มากที่สุดเมื่อเปรียบเทียบกับโพรโทคอลอื่น โดยลดได้อย่างน้อยร้อยละ 4.14 เนื่องมากจาก การที่สามารถลดอัตราการส่งข้อมูลซ้ำ รวมถึงการลด การแลกเปลี่ยนแผนที่บัฟเฟอร์

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอวิธีการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์โดยงานวิจัย 2PMP ได้ใช้ขั้นตอนวิธีการคาดเดาแผนที่บัฟเฟอร์ โดยวิธีการแบบผลักและการดึงสำหรับการส่งข้อมูล

ในงานวิจัยนี้ได้วิเคราะห์งานวิจัยวิธีการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ก่อนหน้าที่สามารถแพร่กระจายข้อมูลได้ดีและมีความหน่วงน้อย แต่มีปัญหาค่าใช้จ่ายในการส่งข้อมูลควบคุมการทำงานอันเนื่องมาจากการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์ในสัดส่วนที่สูง ดังนั้นในงานวิจัยนี้จึงเสนอการ ขั้นตอนที่ช่วยลดการแลกเปลี่ยนข้อมูลแผนที่บัฟเฟอร์โดยใช้การคาดเดาแทนเพื่อลดการเปลี่ยนข้อมูลแผนที่บัฟเฟอร์

วิธีการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้านนั้น สามารถเริ่มการทำงานได้ตั้งแต่เริ่มต้นการทำงานได้ทันที โดยเริ่มต้นด้วยการสุ่มข้อมูล และใช้การเรียนรู้รูปแบบแผนที่บัฟเฟอร์ของเพื่อนบ้านโดยมีการส่งข้อมูลแลกเปลี่ยนเล็กน้อยและปรับการคาดเดาแผนที่ตามที่จำเป็นเท่านั้น ดังนั้นจึงสามารถลดค่าใช้จ่ายที่เกิดขึ้นจากการแลกเปลี่ยนแผนที่บัฟเฟอร์ได้ดีกว่าโพโตคอบอื่นที่นำมาเปรียบเทียบ

ซึ่งจากผลการทดลองพบว่าสามารถลดค่าใช้จ่ายในการส่งข้อมูลควบคุมการทำงานของระบบได้อย่างน้อยร้อยละ 96 เมื่อเทียบกับรูปแบบโพรโทคอลอื่น

ต่อมาในงานวิจัย 2PMP แบบใช้ความน่าจะเป็น มีการปรับปรุงพัฒนาเพิ่มให้สามารถลดค่าใช้จ่ายที่เกิดขึ้นจากการส่งข้อมูลซ้ำซ้อนทำให้ลดค่าใช้จ่ายรวมในระบบได้อย่างน้อยร้อยละ 4.14 เมื่อเปรียบเทียบกับโพรโทคอลอื่น

5.2 ข้อจำกัด

ถึงแม้ว่างานวิจัยนี้จะแสดงให้เห็นถึงประสิทธิภาพในการลดค่าใช้จ่ายที่เกิดขึ้นจากการแลกเปลี่ยนแผนที่บัฟเฟอร์ในสัดส่วนที่สูงมากเมื่อเทียบกับโพรโทคอลที่มาเปรียบเทียบ แต่การคาดเดาแผนที่บัฟเฟอร์ ไม่ได้มีข้อมูลแผนที่บัฟเฟอร์ที่ถูกต้อง แม่นยำ เป็นการกำหนดรูปแบบการส่งข้อมูล และร้องขอข้อมูล ทำให้เกิดปัญหาในการไม่ได้รับชิ้นข้อมูลบางส่วน และประสิทธิภาพการทำงานลดลง อีกทั้งผลการทดลองยังทำในโปรแกรมจำลองจะยังไม่สามารถทราบถึงปัญหาเมื่อนำมาทดลองใช้งานจริง

5.3 ข้อเสนอแนะ

โพรโทคอลการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ที่นำเสนอนั้น สามารถทำงานได้ดี โดยที่มีค่าใช้จ่ายที่เกิดขึ้นในการควบคุมการทำงานต่ำ อีกทั้งเมื่อมีการพัฒนารวมเข้ากับขั้นตอนวิธีการ Reinforcement ทำให้สามารถลดค่าใช้จ่ายที่เกิดขึ้นในระบบได้ดี แต่เป็นการคาดเดาแผนที่บัฟเฟอร์ของเพื่อนบ้าน เพื่อกำหนดรูปแบบในการส่งและร้องขอข้อมูล ดังนั้นจึงทำให้เกิดชิ้นข้อมูลที่ไม่ได้รับ ดังนั้นถ้าสามารถคาดเดาแผนที่เพื่อนบ้าน เพื่อกำหนดรูปแบบการส่งข้อมูลได้ถูกต้องและมีประสิทธิภาพแล้วค่อยมีการส่งข้อมูลให้เพื่อนบ้านได้ จะทำให้สามารถลดการขาดชิ้นข้อมูล และการทำงานมีประสิทธิภาพมากขึ้น

ประเด็นเมื่อเพียร์ในระบบหายไป

โพรโทคอลการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์ที่นำเสนอนั้น ไม่ได้มีการรองรับการทำงานเมื่อเพียร์ในระบบมีการหายไป ดังนั้นในการประยุกต์การใช้งานเมื่อเพียร์ในระบบมีการหายไปสามารถทำได้หลายวิธี โดยใช้การส่งข้อความที่มีอยู่เดิม หรือเพิ่มเติมระบบการทำงานเข้าไป ตัวอย่างเช่น 1. สามารถใช้ Reinforcement push message ที่ส่งมาจากเพื่อนบ้านที่ได้รับการผลักชิ้นข้อมูลไปให้ จะทำให้สามารถตรวจว่าเพื่อนบ้านที่ผลักชิ้นข้อมูลไปให้ยังอยู่ในระบบหรือไม่ โดยถ้ามีการได้รับ Reinforcement push message แสดงว่าเพื่อนบ้านที่ผลักชิ้นข้อมูลไปให้ยังอยู่ในระบบ แต่ถ้าไม่มีการได้รับ Reinforcement push message จากเพื่อนบ้านที่ผลักชิ้นข้อมูลไปให้ แสดงว่าไม่อยู่ในระบบ 2. สามารถใช้ Time out ในการตั้งเวลานับถอยหลัง หลังจากผลักชิ้นข้อมูลไปให้หรือการส่งข้อความร้องขอไป โดยหลังจากเวลาถอยหลังที่ตั้งไว้เมื่อไม่มีการผลักชิ้นข้อมูลมาให้ หรือไม่มีการส่งชิ้นข้อมูลที่ร้องขอไปจะทำการตรวจพบว่า เพื่อนบ้านที่ส่งข้อมูลไปนั้นไม่อยู่ในระบบ

รายการอ้างอิง

[1] PPLive. [Online]. Available from : http://www.pplive.com/ [2011, Dec]

[2] Sopcast. [Online]. Available from : http://www.sopcast.com/ [2011, Dec]

[3] JumpTV. [Online]. Available from : http://www.jumptv.com/ [2011, Dec]

[4] UUSee. [Online]. Available from : http://www.uusee.com/ [2011, Dec]

[5] J. Liu, S. G. Rao, B. Li, and H. Zhang. Opportunities and challenges of peer-to-peer internet video broadcast. Proceedings of the IEEE. 96, 1(2008) : 11-24.

[6] K. Narkdej, S. Choochaisri, and C. Intanagonwiwat. A reinforcementbased push-pull approach for pee r-to-peer live streaming, Proceedings 7th IEEE Int.Conf. on Wireless Communications,Networking, and Mobile Computing, 2011.

[7] T. Bonald, L. Massouli´e, F. Mathieu, D. Perino, and A. Twigg. Epidemic live streaming: optimal performance trade-offs, Proceedings 2008 ACM SIGMETRICS int. Conf. on Measurement and modeling of computer systems, 2008, pp. 325–336.

[8] J. F. Kurose and K. W. Ross. Computer Networking: A Top-Down Approach Featuring the Internet 5th Ed., Pearson Education, 2010.

[9] G. Deng, C. Li, C. Chen, and Y. Zhang. A bitmap coding method for p2p streaming protocols, Proceedings 2nd Int Informatics in Control, Automation and Robotics (CAR) Asia Conf, vol. 3, 2010, pp. 368–372.

[10] V. Venkataraman, K. Yoshida, and P. Francis. Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast, Proceedings 14th IEEE Int. Conf. Network Protocols ICNP ’06, 2006, pp. 2–11.

[11] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: high-bandwidth multicast in cooperative environments, Proceedings 19th ACM symposium on Operating systems principles, 2003, pp. 298–313.

[12] X. Zhang, J. Liu, B. Li, and Y.-S. P. Yum. Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming, Proceedings IEEE 24th Annual Joint Conf. of the IEEE Computer and Communications Societies INFOCOM 2005, vol. 3, 2005, pp. 2102–2111.

[13] G. Zheng, S.-H. G. Chan, X. Luo, and A. C. Begen. Pattern-push: A low-delay mesh-push scheduling for live peer-to-peer streaming, Proceedings IEEE Int. Conf. Multimedia and Expo ICME 2009, 2009.

[14] M. Zhang, J.-G. Luo, L. Zhao, and S.-Q. Yang. A peer-to-peer network for live media streaming using a push-pull approach, Proceedings 13th annual ACM int. Conf. on Multimedia, 2005, pp. 287–290.

[15] N. Magharei, R. Rejaie, and Y. Guo. Mesh or multiple-tree: A comparative study of live p2p streaming approaches, Proceedings INFOCOM 2007 26th IEEE Int. Conf. Computer Communications IEEE, 2007, pp. 1424–1432.

[16] The Network Simulator(NS-2). [Online]. Available from : http://www.isi.edu/nsnam/ns/ [2011, Dec]

[17] Georgia Tech Internetwork Topology Models. [Online]. Available from : http://www.cc.gatech.edu/projects/gtitm/ [2011, Dec]

[18] ns2-app A framework to simplify application layer simulations on NS2. [Online]. Available from : http://code.google.com/p/ns2-app/ [2011, Dec]

ภาคผนวก

ภาคผนวก ก

ตัวอย่างข้อมูลผลการทดลองหน้าต่างรูปแบบเมื่อเวลาผ่านไป

เมื่อพิจารณาหน้าต่างรูปแบบ (Predicting pattern) เมื่อเวลาผ่านไปได้ผลการทดลองโดยจะสังเกตุได้ว่าจะมีการปรับหน้าต่างรูปแบบ (Predicting pattern) ตั้งแต่เริ่มต้นถึงเวลาผ่านไป 20 วินาที แล้วหน้าต่างรูปแบบ (Predicting pattern) จะคงที่เพื่อการส่งรูปแบบนี้ต่อไป ดังนั้นถ้าใน 20 วินาทีแรกมีการปรับหน้าต่างได้ไม่ดี เช่นมีการคาดเดาว่ามีข้อมูล ครบแล้วก็จะไม่มีการส่งข้อมูลนั้นให้ตัวอย่าง เพียร์ A ที่คาดเดาแผนที่บัฟเฟอร์ เพียร์ X ที่เวลา 0 - 90 วินาที ตลอดการส่งข้อมูลไลฟ์สตรีมมิ่งแบบเพียร์ทูเพียร์

Time	Peer	Predict	Predicting Pattern									
0	A	X	0	0	0	1	0	0	0	0	1	1
0	B	X	0	0	0	1	1	0	0	1	0	1
0	C	X	1	0	0	1	1	1	1	1	0	0
0	D	X	0	1	1	1	1	0	0	0	0	0
0	E	X	1	1	0	1	1	1	1	0	1	1
0	F	X	0	0	1	0	1	1	1	1	0	1
0	H	X	0	0	1	0	0	1	1	1	0	1
0	I	X	1	0	0	1	0	1	1	0	1	1
0	J	X	0	1	0	0	0	1	1	0	1	0
0	K	X	0	0	1	0	0	1	1	1	1	0
10	A	X	1	1	1	1	1	1	1	1	1	1
10	B	X	1	1	1	1	1	1	0	1	1	1
10	C	X	1	1	1	1	1	1	1	1	1	1
10	D	X	1	1	1	1	1	0	1	1	1	1
10	E	X	1	1	1	1	1	1	1	0	1	1
10	F	X	1	1	1	1	1	1	1	1	0	1
10	H	X	0	0	1	0	1	0	1	1	0	1
10	I	X	0	0	0	0	0	1	0	0	1	1
10	J	X	0	1	0	1	1	1	1	1	1	1
10	K	X	1	0	1	1	1	1	1	1	1	0
20	A	X	1	1	1	1	1	1	1	1	1	1
20	B	X	1	1	1	1	1	1	0	1	1	1
20	C	X	1	1	1	1	1	1	1	1	1	1
20	D	X	1	1	1	1	1	0	1	1	1	1
20	E	X	1	1	1	1	1	1	1	0	1	1
20	F	X	1	1	1	1	1	1	1	1	0	1
20	H	X	1	1	1	0	1	1	1	1	1	1
20	I	X	1	1	1	1	0	1	1	1	1	1
20	J	X	0	1	0	1	1	1	1	1	1	1
20	K	X	1	0	1	1	1	1	1	1	1	0

ภาพที่ ก.1 ตัวอย่างรูปแบบหน้าต่าง(Predicting pattern) เมื่อเวลา 0 - 20 วินาที

Time	Peer	Predict	Predicting Pattern									
30	A	X	1	1	1	1	1	1	1	1	1	1
30	B	X	1	1	1	1	1	1	0	1	1	1
30	C	X	1	1	1	1	1	1	1	1	1	1
30	D	X	1	1	1	1	1	0	1	1	1	1
30	E	X	1	1	1	1	1	1	1	0	1	1
30	F	X	1	1	1	1	1	1	1	1	0	1
30	H	X	1	1	1	0	1	1	1	1	1	1
30	I	X	1	1	1	1	0	1	1	1	1	1
30	J	X	0	1	0	1	1	1	1	1	1	1
30	K	X	1	0	1	1	1	1	1	1	1	0
40	A	X	1	1	1	1	1	1	1	1	1	1
40	B	X	1	1	1	1	1	1	0	1	1	1
40	C	X	1	1	1	1	1	1	1	1	1	1
40	D	X	1	1	1	1	1	0	1	1	1	1
40	E	X	1	1	1	1	1	1	1	0	1	1
40	F	X	1	1	1	1	1	1	1	1	0	1
40	H	X	1	1	1	0	1	1	1	1	1	1
40	I	X	1	1	1	1	0	1	1	1	1	1
40	J	X	0	1	0	1	1	1	1	1	1	1
40	K	X	1	0	1	1	1	1	1	1	1	0
50	A	X	1	1	1	1	1	1	1	1	1	1
50	B	X	1	1	1	1	1	1	0	1	1	1
50	C	X	1	1	1	1	1	1	1	1	1	1
50	D	X	1	1	1	1	1	0	1	1	1	1
50	E	X	1	1	1	1	1	1	1	0	1	1
50	F	X	1	1	1	1	1	1	1	1	0	1
50	H	X	1	1	1	0	1	1	1	1	1	1
50	I	X	1	1	1	1	0	1	1	1	1	1
50	J	X	0	1	0	1	1	1	1	1	1	1
50	K	X	1	0	1	1	1	1	1	1	1	0
60	A	X	1	1	1	1	1	1	1	1	1	1
60	B	X	1	1	1	1	1	1	0	1	1	1
60	C	X	1	1	1	1	1	1	1	1	1	1
60	D	X	1	1	1	1	1	0	1	1	1	1
60	E	X	1	1	1	1	1	1	1	0	1	1
60	F	X	1	1	1	1	1	1	1	1	0	1
60	H	X	1	1	1	0	1	1	1	1	1	1
60	I	X	1	1	1	1	0	1	1	1	1	1
60	J	X	0	1	0	1	1	1	1	1	1	1
60	K	X	1	0	1	1	1	1	1	1	1	0

ภาพที่ ก.2 ตัวอย่างรูปแบบหน้าต่าง(Predicting pattern) เมื่อเวลา 30 - 60 วินาที

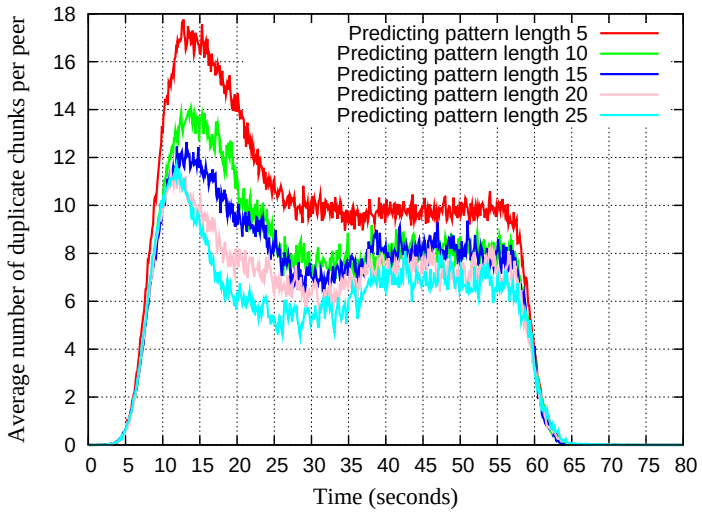
Time	Peer	Predict	Predicting Pattern									
70	A	X	1	1	1	1	1	1	1	1	1	1
70	B	X	1	1	1	1	1	1	0	1	1	1
70	C	X	1	1	1	1	1	1	1	1	1	1
70	D	X	1	1	1	1	1	0	1	1	1	1
70	E	X	1	1	1	1	1	1	1	0	1	1
70	F	X	1	1	1	1	1	1	1	1	0	1
70	H	X	1	1	1	0	1	1	1	1	1	1
70	I	X	1	1	1	1	0	1	1	1	1	1
70	J	X	0	1	0	1	1	1	1	1	1	1
70	K	X	1	0	1	1	1	1	1	1	1	0
80	A	X	1	1	1	1	1	1	1	1	1	1
80	B	X	1	1	1	1	1	1	0	1	1	1
80	C	X	1	1	1	1	1	1	1	1	1	1
80	D	X	1	1	1	1	1	0	1	1	1	1
80	E	X	1	1	1	1	1	1	1	0	1	1
80	F	X	1	1	1	1	1	1	1	1	0	1
80	H	X	1	1	1	0	1	1	1	1	1	1
80	I	X	1	1	1	1	0	1	1	1	1	1
80	J	X	0	1	0	1	1	1	1	1	1	1
80	K	X	1	0	1	1	1	1	1	1	1	0
90	A	X	1	1	1	1	1	1	1	1	1	1
90	B	X	1	1	1	1	1	1	0	1	1	1
90	C	X	1	1	1	1	1	1	1	1	1	1
90	D	X	1	1	1	1	1	0	1	1	1	1
90	E	X	1	1	1	1	1	1	1	0	1	1
90	F	X	1	1	1	1	1	1	1	1	0	1
90	H	X	1	1	1	0	1	1	1	1	1	1
90	I	X	1	1	1	1	0	1	1	1	1	1
90	J	X	0	1	0	1	1	1	1	1	1	1
90	K	X	1	0	1	1	1	1	1	1	1	0

ภาพที่ ก.3 ตัวอย่างรูปแบบหน้าต่าง(Predicting pattern) เมื่อเวลา 70 - 90 วินาที

ภาคผนวก ข

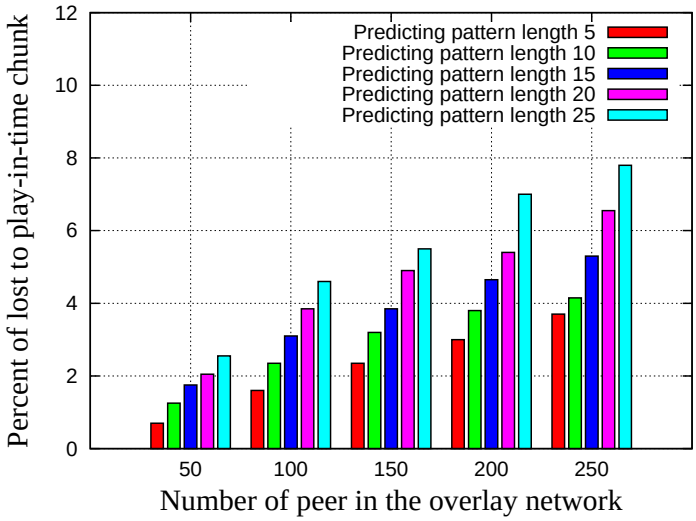
การทดสอบขนาดของหน้าต่างรูปแบบที่เหมาะสม

มุมมองทางด้านการส่งข้อมูลที่ซ้ำซ้อนกัน



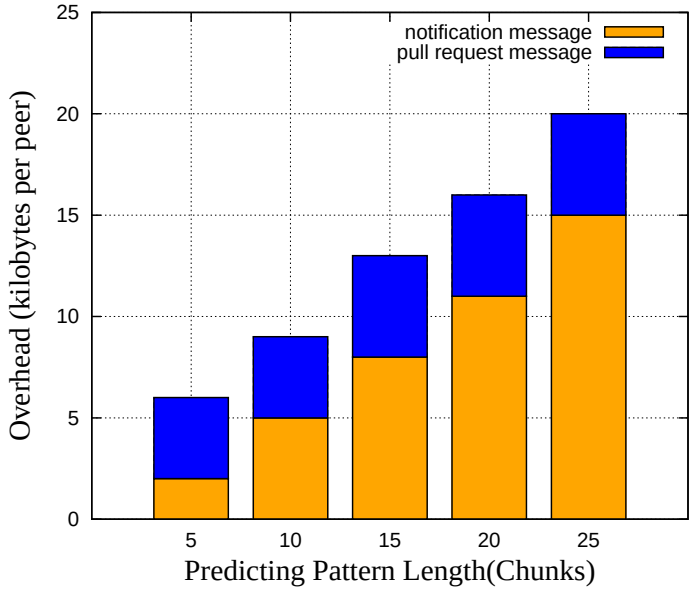
ภาพที่ ข.1 การส่งข้อมูลที่ซ้ำซ้อนกัน

มุมมองทางด้านความสามารถในการเล่นข้อมูลได้ในเวลา



ภาพที่ ข.2 ชิ้นข้อมูลที่ไม่สามารถเล่นได้ในเวลา

มุมมองทางด้านค่าใช้จ่าย



ภาพที่ ข.3 ค่าใช้จ่ายที่เกิดขึ้น

พิจารณาขนาดของหน้าต่างรูปแบบ (Predicting pattern) ของโพรโทคอลที่นำเสนอ โดยทดสอบที่ค่า 5, 10, 15, 20,25 ชิ้น ตามลำดับจากกราฟที่ได้ แสดงให้เห็นว่าเมื่อ ใช้ขนาดหน้าต่างรูปแบบ(Predicting pattern) ที่มากขึ้นจะทำให้มีค่าใช้จ่ายที่มากขึ้นเพราะมีช่วงในการปรับที่มากขึ้น รวมถึงมีการสูญเสียชิ้นข้อมูลที่ไม่สามารถเล่นได้ในเวลาที่มากขึ้น แต่ในทางกลับกันจะทำให้มีชิ้นข้อมูลที่ซ้ำซ้อนกันน้อยลง

ประวัติผู้เขียนวิทยานิพนธ์

นาย ประทีป พัตราภรณ์พิศุทธิ์ เกิดเมื่อวันที่ 29 ตุลาคม พ.ศ. 2528 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียน ศรีพฤฒา จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมสารสนเทศ ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2550 และเข้าศึกษาในหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2553