

อัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูล บนระบบเครือข่ายที่กำหนดโดย
ซอฟต์แวร์



บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

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 University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Queuing-Aware Routing Algorithm in Software Defined Networks

Mr. Pakawat Chumcharoen



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 2017

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์ อัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิว
ของข้อมูล บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์
โดย นายภควัต ชุ่มเจริญ
สาขาวิชา วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก รองศาสตราจารย์ ดร. กุลธิดา โรจนวิบูลย์ชัย

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาโท

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร. กุลธิดา โรจนวิบูลย์ชัย)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภา)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร. อนันต์ ผลเพิ่ม)

ภควัด ชุ่มเจริญ : อัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูล บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Queuing-Aware Routing Algorithm in Software Defined Networks) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. กุลธิดา โรจน์วิบูลย์ชัย, 53 หน้า.

อัลกอริทึมการหาเส้นทางเป็นกลไกที่มีความสำคัญต่อการตัดสินใจในการเลือกเส้นทางในการส่งของข้อมูลในระบบเครือข่ายโดยที่ผ่านมามีอัลกอริทึมการหาเส้นทางต่างๆ มากมายที่พัฒนามาจากทฤษฎีกราฟพื้นฐานด้วยคำนึงถึงการเลือกเส้นทางที่สั้นที่สุดหรือ shortest path และนอกจากนั้นตัวชี้วัดของระบบเครือข่ายจำนวนมากได้ถูกนำไปใช้เพื่อเพิ่มคุณภาพของการควบคุมคุณภาพการให้บริการ (QoS) ที่อุปกรณ์ปลายทาง แต่อย่างไรก็ตามการเลือกเส้นทางจากอัลกอริทึมการหาเส้นทางที่มีอยู่เดิมนั้น เมื่อมีปริมาณการส่งข้อมูลที่เพิ่มขึ้นยังทำให้เกิดเวลาแฝง (Lantency) ของการส่งข้อมูลจากอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่ง และเกิดขึ้นเป็นช่วงเวลายาวนาน ในระบบเครือข่าย โดยปัญหาดังกล่าวจะเรียกว่า ปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ หรือ Bufferbloat Problem ซึ่งเกิดจากการที่มีปริมาณข้อมูลในบัฟเฟอร์เกินขนาดของบัฟเฟอร์ งานวิจัยนี้จึงได้เสนออัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูล บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์โดยใช้ความสามารถของระบบแบบรวมศูนย์กลาง เพื่อรวบรวมข้อมูลขนาดบัฟเฟอร์โดยตรงตามเวลาจริง เพื่อคำนวณเส้นทางที่มีระยะที่สั้นที่สุดที่มีระดับค่าการครอบครองพื้นที่ในบัฟเฟอร์ที่ยอมรับได้ โดยผลประสิทธิภาพของอัลกอริทึม ที่ได้จากทดลองแสดงให้เห็นว่าอัลกอริทึมที่เสนอในงานวิจัยนี้มีประสิทธิภาพเหนือกว่าอัลกอริทึมที่หาเส้นทางที่ระยะทางที่สั้นที่สุดในแง่ของผลรวมของปริมาณงาน (Overall Throughput) ช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) และความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ที่เกิดในระบบเครือข่ายเป็นอย่างมาก

ภาควิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อ อ.ที่ปรึกษาหลัก

ปีการศึกษา 2560

5870303821 : MAJOR COMPUTER ENGINEERING

KEYWORDS: SOFTWARE DEFINED NETWORKING (SDN) / BUFFERBLOAT

PAKAWAT CHUMCHAROEN: Queuing-Aware Routing Algorithm in Software Defined Networks. ADVISOR: ASSOC. PROF. KULTIDA ROJVIBOONCHAI, Ph.D., 53 pp.

Routing Algorithm is an important mechanism for selectivity data transferring path in a network. Several existed routing algorithms are derived from the fundamental of graph theory with shortest path approach. A lot of additional network metrics were applied, to serve the best quality of service (QoS), to end devices. However, a selected path from the existed algorithms may suffer from additional end-to-end latency for long time periods when the number of traffic flows increases. This situation is called the bufferbloat problem which is caused by excessively large queue in buffer. In this thesis, we proposed a novel queuing-aware routing algorithm in software defined networks. The proposed algorithm leverages the capability of centralized system to gather directly the buffering information in real-time to calculate the shortest path with acceptable buffering occupancy level. The simulation result shows that the proposed algorithm outperforms the traditional shortest path approach in term of overall throughput, round trip time, and packet loss significantly.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Department: Computer Engineering Student's Signature

Field of Study: Computer Engineering Advisor's Signature

Academic Year: 2017

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาอย่างสูงจากรองศาสตราจารย์ ดร. กุลธิดา โรจนวิบูลย์ชัย อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่กรุณาสละเวลาช่วยเหลือในการให้ความรู้ คำปรึกษา คำแนะนำ รวมทั้งให้แนวคิดที่เป็นประโยชน์ในการทำวิจัย และช่วยชี้แนะการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นระหว่างทำการวิจัย ขอขอบพระคุณอาจารย์มา ณ โอกาสนี้ และขอขอบพระคุณคณาจารย์ทุกท่านที่แนะนำสั่งสอนและให้ความรู้แก่ข้าพเจ้าตลอดเวลาการศึกษา

ขอขอบพระคุณคณะกรรมการการสอบวิทยานิพนธ์ทุกท่านเป็นอย่างสูง ผู้ช่วยศาสตราจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์ ประธานคณะกรรมการสอบวิทยานิพนธ์, ผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภาก กรรมการสอบวิทยานิพนธ์ และรองศาสตราจารย์ ดร. อนันต์ ผลเพิ่ม กรรมการสอบวิทยานิพนธ์ ที่ให้ความกรุณาตรวจสอบและให้ข้อเสนอแนะอันเป็นประโยชน์ในการปรับปรุงแก้ไข ทำให้วิทยานิพนธ์นี้มีความถูกต้องและสมบูรณ์ยิ่งขึ้น

งานวิจัยนี้ได้รับการสนับสนุนจากหน่วยปฏิบัติการวิจัยโครงข่ายไร้สาย และอินเทอร์เน็ตอนาคต (Wireless Network and Future Internet Research Unit) กองทุนรัชดาภิเษกสมโภช จุฬาลงกรณ์มหาวิทยาลัย

ขอขอบพระคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ให้ทุนอุดหนุนการศึกษา “อัจฉริยะคืนรัง” เพื่อสนับสนุนค่าใช้จ่ายในระหว่างการศึกษาเล่าเรียน รวมถึงสถานที่และทรัพยากรที่จำเป็นต่อการทำวิจัยทำให้การวิจัยสำเร็จและลุล่วงด้วยดี

ขอขอบพระคุณพี่ๆ ในห้องปฏิบัติการ Information System Engineering ภาควิชาวิศวกรรมคอมพิวเตอร์ที่ให้คำปรึกษาและคำแนะนำทั้งเรื่องงานวิจัยและการใช้ชีวิต

ขอขอบพระคุณบิดา มารดา รวมถึงบุคคลในครอบครัวของผู้วิจัย ที่ให้การสนับสนุนและกำลังใจแก่ผู้วิจัยเสมอมา

สุดท้ายนี้ ผู้วิจัยขอขอบพระคุณทุกท่านที่มีส่วนช่วยเหลือจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จเรียบร้อยไปได้ด้วยดีทุกประการ

สารบัญ

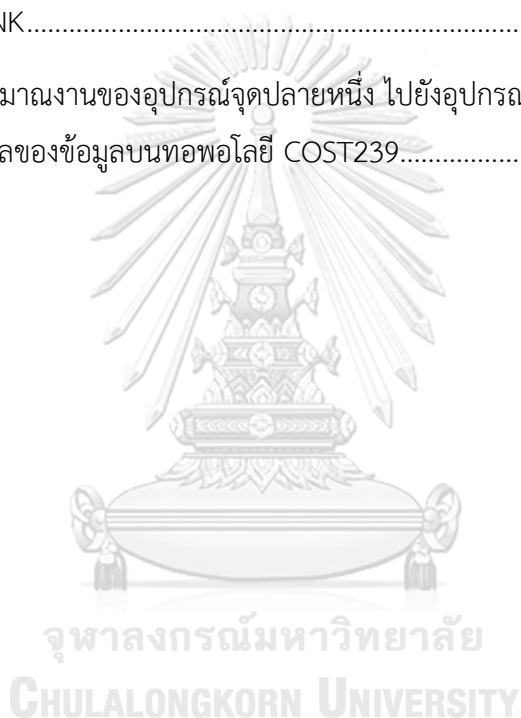
หน้า

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฅ
สารบัญภาพ	ญ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์	3
1.3 ขอบเขตของงานวิจัย.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 ขั้นตอนการดำเนินงานวิจัย.....	4
1.6 งานวิจัยที่ได้การตีพิมพ์.....	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	5
2.1 ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software Defined Networks : SDN).....	5
2.2 ปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ (Bufferbloat Problem).....	6
2.3 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	8
2.3.1 มุมมองในการตัดสินใจเลือกเส้นทางของอัลกอริทึมในการหาเส้นทางบนพื้นฐานของ การใช้ตัวชี้วัดของระบบเครือข่ายที่เฉพาะเจาะจง.....	8
2.3.2 มุมมองในการตัดสินใจเลือกเส้นทางของอัลกอริทึมในการหาเส้นทางบนพื้นฐาน ของการใช้ตัวชี้วัดของระบบเครือข่ายที่เฉพาะเจาะจง	9
บทที่ 3 แนวคิดและวิธีการดำเนินงานวิจัย.....	12
3.1 การปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์.....	12

3.2 อัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์.....	14
บทที่ 4 การทดลองและผลการทดลอง	20
4.1 ประสิทธิภาพของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239	22
4.2 ประสิทธิภาพของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี SPLINTLINK	27
4.3 การปรับหาช่วงเวลาที่เหมาะสมในการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์.....	32
4.4 ประสิทธิภาพการเพิ่มจำนวนการไหลของข้อมูลของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239	35
4.5 ค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane).....	42
บทที่ 5 บทสรุปและแนวทางในการพัฒนาต่อ.....	48
5.1 บทสรุป	48
5.2 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ	49
รายการอ้างอิง	50
ประวัติผู้เขียนวิทยานิพนธ์	53

สารบัญตาราง

ตารางที่ 1 การเปรียบเทียบงานวิจัยที่เกี่ยวข้อง	10
ตารางที่ 2 การตั้งค่าพารามิเตอร์	22
ตารางที่ 3 ค่าเฉลี่ยปริมาณงานของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งบน ทอพอโลยี COST239	26
ตารางที่ 4 ค่าเฉลี่ยปริมาณงานของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งบน ทอพอโลยี SPLINTLINK	31
ตารางที่ 5 ค่าเฉลี่ยปริมาณงานของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งของ การเพิ่มจำนวนการไหลของข้อมูลบนทอพอโลยี COST239	38



สารบัญภาพ

ภาพที่ 1 ชั้นของ Layer ที่ถูกแบ่งเป็น Control Plane กับ Data Plane	5
ภาพที่ 2 การเปรียบเทียบสถาปัตยกรรมของระบบเครือข่ายแบบดั้งเดิม กับระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์.....	6
ภาพที่ 3 การเกิดปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ที่ส่วนต่อประสานของสวิตช์	7
ภาพที่ 4 องค์ประกอบภายในคอนโทรลเลอร์	12
ภาพที่ 5 แผนภาพขั้นตอนการทำงานของอัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ของแต่ละองค์ประกอบภายในคอนโทรลเลอร์	16
ภาพที่ 6 แผนภาพขั้นตอนการตัดสินใจเลือกและติดตั้งเส้นทางสำหรับการไหลของการรับส่งข้อมูลที่เข้ามาใหม่ของอัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์	19
ภาพที่ 7 ทอพอโลยี COST239	21
ภาพที่ 8 ทอพอโลยี SPLINTLINK	21
ภาพที่ 9 กราฟการเปรียบเทียบผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239	23
ภาพที่ 10 กราฟการเปรียบเทียบช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239	24
ภาพที่ 11 กราฟการเปรียบเทียบความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239	25

ภาพที่ 12 กราฟการเปรียบเทียบผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี SPLINTLINK28

ภาพที่ 13 กราฟการเปรียบเทียบช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี SPLINTLINK29

ภาพที่ 14 กราฟการเปรียบเทียบความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี SPLINTLINK30

ภาพที่ 15 กราฟการเปรียบเทียบผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 ในแต่ละช่วงเวลา.....34

ภาพที่ 16 กราฟการเปรียบเทียบผลรวมของปริมาณงาน (Overall Throughput) การเพิ่มจำนวนการไหลของข้อมูลของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST23935

ภาพที่ 17 กราฟการเปรียบเทียบจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้นของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 ในแต่ละช่วงเวลา45

ภาพที่ 18 กราฟการเปรียบเทียบจำนวนข้อความของเกณฑ์วิธี OpenFlow กับจำนวนของสวิตช์ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 ในแต่ละช่วงเวลา47

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในช่วงศตวรรษที่ผ่านมา ได้มีแนวคิดของการใช้ซอฟต์แวร์ เข้ามาช่วยในการแก้ปัญหาของระบบเครือข่ายแบบเดิม เช่น การหาเส้นทางในการส่งข้อมูลนั้นจะต้องขึ้นอยู่กับ อัลกอริทึมที่ใช้ในการหาเส้นทางในอุปกรณ์จัดเส้นทาง (Router) ในแต่ละตัวที่ข้อมูลผ่าน ซึ่งจะใช้อัลกอริทึมที่แตกต่างกัน ขึ้นอยู่กับระบบเครือข่ายนั้นๆ แต่โดยส่วนใหญ่จะใช้อัลกอริทึมในการหาเส้นทางที่สั้นที่สุดหรือ Shortest Path Algorithm ในการหาเส้นทาง ซึ่งในบางครั้งจะได้เส้นทางที่ไม่เหมาะสมกับการส่งข้อมูลบางประเภท ทำให้สูญเสียเวลาในการส่งข้อมูลนั้นๆ เป็นต้น โดยแนวคิดนี้จะทำการแยกส่วนของแพ็คเกจออกเป็นสองส่วนดังนี้ ส่วนของการควบคุม (Control Plane) และส่วนของข้อมูล (Data Plane) ซึ่งส่วนของการควบคุม จะอยู่ในส่วนที่เรียกว่า คอนโทรลเลอร์ (Controller) และ ส่วนของข้อมูล จะอยู่ในส่วนของสวิตช์ (Switch) โดยแนวคิดนี้เรียกว่า ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ หรือ Software Defined Networks (SDN) [1] โดยที่ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ นั้นสามารถแก้ไขปัญหาดังกล่าวได้โดยใช้ ความสามารถของตัวคอนโทรลเลอร์ที่มีภาพรวมทั้งหมดของระบบเครือข่าย เช่น รูปแบบการเชื่อมต่อของระบบเครือข่าย ค่าปริมาณปัจจุบันของความกว้างแถบความถี่ (Bandwidth) ในแต่ละการเชื่อมโยง และ flow entries ของแต่ละสวิตช์ เป็นต้น [2]

ในการหาเส้นทางที่ดีที่สุดที่เหมาะสมกับการส่งข้อมูลนั้นๆ ได้ คอนโทรลเลอร์จะเป็นส่วนที่คำนวณหาเส้นทางและติดตั้ง forwarding rules ในแต่ละสวิตช์ เมื่อมีกลุ่มข้อมูล (Packet) เข้ามาที่สวิตช์ กลุ่มข้อมูลจะวิ่งไปตามเส้นทางที่ถูกติดตั้ง forwarding rules ไว้ แต่ถ้าไม่มี forwarding rules ที่ถูกติดตั้งไว้ สวิตช์จะทำการติดต่อกับคอนโทรลเลอร์ โดยใช้โปรโตคอลมาตรฐานที่ชื่อว่า OpenFlow ซึ่งเป็นคอนโทรลเลอร์ใช้ในการสื่อสารระหว่างส่วนของการควบคุมและส่วนของข้อมูล โดยอัลกอริทึมที่ใช้ในการหาเส้นทางในปัจจุบันจะมีตัวอยู่ด้วยกันหลากหลายอัลกอริทึม ไม่ว่าจะเป็นอัลกอริทึมที่หาเส้นทางที่ระยะทางที่สั้นที่สุด หรือ Shortest Path Algorithm หรืออัลกอริทึมที่พิจารณาข้อมูลภาพรวมของระบบเครือข่าย เช่น พิจารณาค่าความกว้างแถบความถี่ (Bandwidth) และค่าความล่าช้าของการส่งข้อมูลในแต่ละลิงค์ ซึ่งถูกเสนอในงานวิจัยที่ [3-9] เพื่อสนับสนุนโปรแกรมประยุกต์ที่

มีการร้องขอการควบคุมคุณภาพการให้บริการ (QoS) และที่ไม่มีมีการร้องขอการควบคุมคุณภาพการให้บริการ (Non-QoS) แต่อย่างไรก็ตามในการส่งข้อมูลระหว่างโปรแกรมประยุกต์จากอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่ง ยังคงทำให้เกิดความล่าช้าได้ เนื่องจากปริมาณความแออัดของการส่งข้อมูลในเส้นทางนั้นมีปริมาณเพิ่มสูงขึ้น ซึ่งส่งผลให้ปริมาณข้อมูลในบัฟเฟอร์มีขนาดใหญ่ และเกิดขึ้นเป็นช่วงเวลายาวนาน ทำให้มีผลกระทบต่อประสิทธิภาพของการส่งข้อมูลในระบบเครือข่าย โดยปัญหาดังกล่าวจะเรียกว่า Bufferbloat Problem [10]

โดยที่ผ่านมามีงานวิจัยที่พยายามแก้ปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ (Bufferbloat Problem) แต่อย่างไรก็ตามงานวิจัยส่วนใหญ่จะยังคงแก้ไขปัญหานี้บนระบบเครือข่ายแบบดั้งเดิม การแก้ปัญหาของงานวิจัย [11, 12] เสนอการแก้ไขปัญหปริมาณข้อมูลเกินขนาดบัฟเฟอร์ทั้งในระดับฮาร์ดแวร์ และซอฟต์แวร์ โดยต้องมีการปรับตัวแปรต่างๆ อย่างละเอียดให้เหมาะสมกับคุณลักษณะของระบบเครือข่ายที่มีลักษณะที่แตกต่างกัน บนอุปกรณ์ระบบเครือข่าย นอกจากนี้ยังมีงานวิจัย [13, 14] เสนอแบบจำลองการวิเคราะห์ความล่าช้าของส่งการข้อมูลเพื่อปรับปรุงกลไกการส่งข้อมูลระหว่างโปรแกรมประยุกต์จากอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่ง บนโปรแกรมประยุกต์ และระดับชั้นสื่อสารนำส่งข้อมูล (transport layers) ตามมาตรฐาน OSI Model ซึ่งรูปแบบการแก้ปัญหาของงานวิจัยดังกล่าว จะต้องมีการดัดแปลงแก้ไขเคอร์เนลของระบบปฏิบัติการบนอุปกรณ์ปลายทางเพื่อรองรับกลไกการแก้ไขปัญหาดังกล่าว แต่อย่างไรก็ตามคอนโทรลเลอร์นั้นสามารถที่จะเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับขนาดของบัฟเฟอร์จากสวิตช์ได้โดยตรง ซึ่งวิธีนี้เป็นวิธีที่ดีที่สุดที่สามารถจะแก้ไขดังกล่าวได้

งานวิจัยนี้จึงได้เสนออัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ ในส่วนของข้อมูล (Data Plane) โดยทำการเฝ้าสังเกตค่าการครอบครองพื้นที่ในบัฟเฟอร์ของส่วนต่อประสาน (Interface) บนสวิตช์ เมื่อค่าการครอบครองพื้นที่ในบัฟเฟอร์ปัจจุบันมีขนาดที่มากกว่าหรือเท่ากับเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ซึ่งแสดงให้เห็นว่าเกิดความแออัดที่เส้นทางนั้น ส่งผลให้เกิดความล่าช้าและการสูญเสียของกลุ่มข้อมูล ดังนั้นอัลกอริทึมของงานวิจัยนี้จะหลีกเลี่ยงการเกิดความล่าช้าในแต่ละเส้นทางที่เกิดขึ้นในอนาคตได้ โดยการคำนวณหาเส้นทางที่มีระยะทางสั้นที่สุดในแต่ละเส้นทางไว้ก่อน จากนั้นเมื่อเกิดความแออัดที่เส้นทางใด คอนโทรลเลอร์จะเข้าไปทำการติดตั้ง Forwarding Rules ในสวิตช์ใหม่ ตามเส้นทางที่มีระยะทางสั้นที่สุดและความแออัดน้อยที่สุดในลำดับรองลงมา โดยในงานวิจัยนี้จะนำเสนอผลการทดลองในโปรแกรมเลียนแบบพฤติกรรมกรการ

ทำงานของระบบเครือข่ายที่เรียกว่า มินิเน็ต (Mininet) และวัดผลประสิทธิภาพของอัลกอริทึม ในแง่ของผลรวมของปริมาณงาน (Overall Throughput) ช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) และความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ที่เกิดในระบบเครือข่ายเป็นอย่างมาก

1.2 วัตถุประสงค์

1. เพื่อเสนออัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์บนพื้นฐานของการหาเส้นทางที่มีระยะทางที่สั้นที่สุด

1.3 ขอบเขตของงานวิจัย

1. งานวิจัยนี้จะทำการทดลองเฉพาะบนโปรแกรมเลียนแบบพฤติกรรมการทำงานของระบบเครือข่าย (Mininet)
2. งานวิจัยนี้จะใช้เกณฑ์วิธีมาตรฐาน OpenFlow เวอร์ชัน 1.0 (OpenFlow 1.0)
3. งานวิจัยนี้จะทำการทดลองโดยใช้ทอพอโลยี COST239 ซึ่งเป็นทอพอโลยีของระบบเครือข่ายที่เชื่อมต่อหัวเมืองใหญ่ในสหภาพยุโรปและ SPLINTLINK ซึ่งเป็นทอพอโลยีของระบบเครือข่ายที่เชื่อมต่อหัวเมืองใหญ่ในสหรัฐอเมริกา
4. งานวิจัยนี้จะพิจารณาเฉพาะการไหลของการรับส่งข้อมูลประเภท TCP เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

งานวิจัยนี้จะทำการสร้างอัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ เพื่อหลีกเลี่ยงเส้นทางที่มีความแออัด ซึ่งส่งผลให้เกิดความล่าช้าและการสูญเสียของกลุ่มข้อมูลในการส่งข้อมูลในระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ ดังนั้นอัลกอริทึมที่เสนองานวิจัยนี้จะช่วยให้การส่งข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์มีประสิทธิภาพและใช้ประโยชน์จากระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ได้อย่างสูงสุด

1.5 ขั้นตอนการดำเนินงานวิจัย

1. ศึกษางานวิจัยที่เกี่ยวข้องกับแก้ปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ (Bufferbloat Problem) ในระบบเครือข่ายแบบดั้งเดิม
2. ศึกษางานวิจัยที่เกี่ยวข้องกับอัลกอริทึมในการหาเส้นทาง บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์
3. ศึกษาฟังก์ชันการทำงานของ POX คอนโทรลเลอร์ ในการหาเส้นทางที่มีระยะทางที่สั้นที่สุด
4. ดำเนินการออกแบบและพัฒนาองค์ประกอบที่คอยตรวจสอบและคำนวณค่าการครอบครองพื้นที่ในบัฟเฟอร์ บนคอนโทรลเลอร์
5. ศึกษาการทำงานของโปรแกรมเลียนแบบพฤติกรรมการทำงานของระบบเครือข่าย หรือ Mininet Emulator
6. พัฒนาและวิเคราะห์การทำงานของอัลกอริทึม บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ บนโปรแกรมเลียนแบบพฤติกรรมการทำงานของระบบเครือข่าย
7. วิเคราะห์และวัดผลประสิทธิภาพของอัลกอริทึม ในแง่ของผลรวมของปริมาณงาน (Overall Throughput), ช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) และความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ที่เกิดในระบบเครือข่าย

1.6 งานวิจัยที่ได้การตีพิมพ์

งานวิจัยที่เป็นส่วนหนึ่งของวิทยานิพนธ์และได้รับการนำเสนอในการประชุมวิชาการมีดังนี้

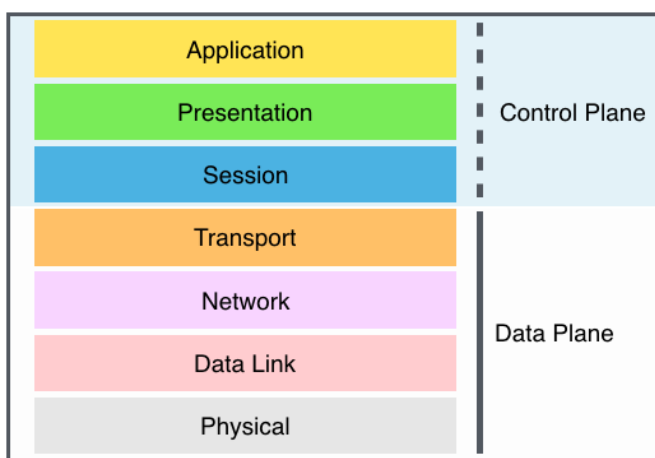
- 1) Pakawat Chumcharoen, Kiattikun Kawila, Kulit Na Nakorn, and Kultida Rojviboonchai, "Queuing-aware Routing Algorithm in Software Defined Networks," in *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2017.

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software Defined Networks : SDN)

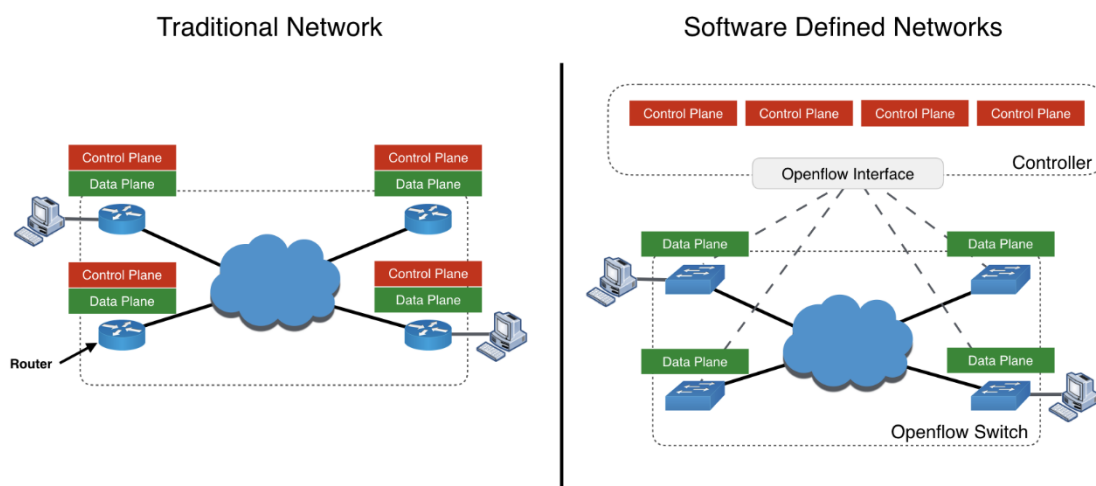
ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์หรือ Software Defined Networks เป็นระบบเครือข่ายที่มีแนวคิดของการใช้ซอฟต์แวร์ เข้ามาช่วยในการแก้ปัญหาของระบบเครือข่ายแบบเดิม เช่น การหาเส้นทางในการส่งข้อมูลนั้นจะต้องขึ้นอยู่กับ อัลกอริทึมที่ใช้ในการหาเส้นทางใน router ในแต่ละตัวที่ข้อมูลผ่าน ซึ่งจะใช้อัลกอริทึมที่แตกต่างกัน ขึ้นอยู่กับระบบเครือข่ายนั้นๆ แต่โดยส่วนใหญ่จะใช้อัลกอริทึมในการหาเส้นทางที่สั้นที่สุดหรือ Shortest Path Algorithm ในการหาเส้นทาง ซึ่งในบางครั้งจะได้เส้นทางที่ไม่เหมาะสมกับการส่งข้อมูลบางประเภท ทำให้สูญเสียเวลาในการส่งข้อมูลนั้นๆ เป็นต้น โดยแนวคิดนี้จะทำการแยกส่วนของแพ็คเกจออกเป็นสองส่วนดังนี้ ส่วนของการควบคุม (Control Plane) และส่วนของข้อมูล (Data Plane) ซึ่งส่วนของการควบคุม จะอยู่ในส่วนที่เรียกว่า คอนโทรลเลอร์ (Controller) และ ส่วนของข้อมูล จะอยู่ในส่วนของสวิตช์ (Switch)



ภาพที่ 1 ชั้นของ Layer ที่ถูกแบ่งเป็น Control Plane กับ Data Plane

โดยที่ SDN นั้นสามารถแก้ไขปัญหาดังกล่าวได้โดยใช้ ความสามารถของตัวคอนโทรลเลอร์ที่มีภาพรวมทั้งหมดของระบบเครือข่าย ในการหาเส้นทางที่ดีที่สุดที่เหมาะสมกับการส่งข้อมูลนั้นๆ ได้ ซึ่งคอนโทรลเลอร์กับสวิตช์ จะมีโปรโตคอลมาตรฐานที่ชื่อว่า OpenFlow ใช้ในการสื่อสารระหว่างกัน นอกจากนี้ SDN จะดึงการจัดการเครือข่ายทั้งหมดมารวมอยู่ที่ซอฟต์แวร์ที่อยู่บนคอนโทรลเลอร์ โดยไม่สนใจอุปกรณ์และความแตกต่างของผู้ผลิต ในส่วนนี้ SDN จะเข้ามาแก้ไขปัญหาตรงที่ผู้ดูแล

เครือข่ายสามารถจัดการตั้งค่าอุปกรณ์ทั้งหมดในระบบเครือข่ายได้ในจุดเดียว แล้วสามารถนำไปใช้ได้เลย เมื่อมีอุปกรณ์ใหม่เพิ่มเข้ามาในระบบ ทำให้ไม่จำเป็นที่จะต้องแก้ไขค่าการตั้งค่าต่างๆ ที่ตัวอุปกรณ์จัดเส้นทาง (Router) สามารถแก้ไขที่ซอฟต์แวร์บนคอนโทรลเลอร์ได้ทันที



ภาพที่ 2 การเปรียบเทียบสถาปัตยกรรมของระบบเครือข่ายแบบดั้งเดิม กับระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

2.2 ปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ (Bufferbloat Problem)

ปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ (Bufferbloat Problem) เป็นปัญหาที่เกิดจากปริมาณข้อมูลในบัฟเฟอร์มีขนาดใหญ่เกินไป และเกิดขึ้นเป็นช่วงเวลาที่ยาวนานในระบบเครือข่าย โดยส่วนใหญ่ระบบการเข้าคิวของข้อมูลในบัฟเฟอร์บนอุปกรณ์เครือข่ายจะเป็นระบบการเข้าคิวแบบเข้าก่อนออกก่อน หรือ First-In-First-Out (FIFO) โดยปัญหาดังกล่าวจะทำให้เกิดปริมาณการเข้าคิวของข้อมูลที่มีความยาวมากขึ้น เมื่อมีกลุ่มข้อมูลที่เข้ามาที่เข้าคิวที่หลังจะทำให้เกิดเวลาแฝง (Lantency) ที่กลุ่มข้อมูลนั้น จนกระทั่งเมื่อมีปริมาณข้อมูลในบัฟเฟอร์เกินขนาดของบัฟเฟอร์แล้ว กลุ่มข้อมูลที่จะมาเข้าคิวอีกก็ไม่สามารถที่จะเข้าคิวได้ ทำให้สูญเสียกลุ่มข้อมูลนั้นไป นอกจากนั้นยังทำให้มีผลกระทบต่อประสิทธิภาพของการส่งข้อมูลในระบบเครือข่าย เช่นทำให้เกิดเวลาแฝง (Lantency) ความล่าช้าของการส่งข้อมูล (Delay) ความแปรปรวนของการส่งข้อมูล (Jitter) และปริมาณความแออัดของการส่งข้อมูลในเส้นทางนั้นมีปริมาณเพิ่มสูงขึ้น รวมไปถึงการส่งผลกระทบต่อปริมาณงาน (Throughput) โดยรวมของระบบเครือข่ายให้มีปริมาณงานที่ลดลง ซึ่งปัญหานี้สามารถเกิดได้ทุกที่ในระบบเครือข่าย ไม่ว่าจะเป็นอุปกรณ์จัดเส้นทาง (Router) หรือ สวิตช์

โดยอาจจะเกิดจากการมีจุดติดขัด (Bottleneck) เกิดขึ้นที่เส้นทางใดเส้นทางหนึ่งในระบบเครือข่าย รวมไปถึงการใช้เส้นทางในการส่งข้อมูลรวมกัน และมีข้อมูลที่ถูกส่งในเส้นทางที่มีการใช้รวมกันมีขนาดใหญ่เกินไปจนเกินขนาดของบัฟเฟอร์ ซึ่งการที่จะมีข้อมูลขนาดใหญ่ที่จะให้เกินขนาดของบัฟเฟอร์ได้นั้นอาจจะเกิดขึ้นจากพฤติกรรมของผู้ใช้หรือโปรแกรมประยุกต์ได้ ดังนี้

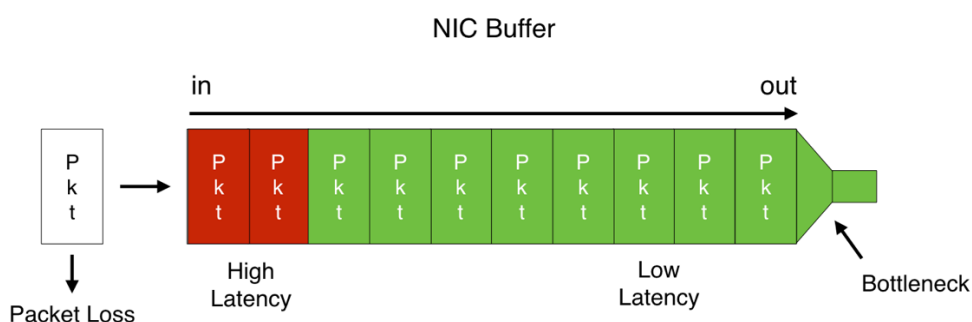
2.2.1 การอัปโหลดวิดีโอที่มีขนาดใหญ่ไปยัง YouTube

2.2.2 การส่งอีเมลข้อความที่มีการแนบรูปภาพขนาดใหญ่

2.2.3 การสำรองข้อมูลไฟล์ขนาดใหญ่หรือไฟล์ระบบปฏิบัติการ

2.2.4 การดาวน์โหลดไฟล์ขนาดใหญ่ เช่น ไฟล์ ISO ของระบบปฏิบัติการ ไฟล์ภาพยนตร์ผ่าน Bittorrent รวมไปถึงการดูวิดีโอผ่านวิดีโอสตรีมมิ่ง ที่มีคุณภาพของวิดีโอขนาดที่คมชัด

2.2.5 การเข้าชมเว็บบางประเภทที่มีการใช้รูปภาพ หรือมีเนื้อหาที่มีขนาดใหญ่



ภาพที่ 3 การเกิดปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ที่ส่วนต่อประสานของสวิตช์

2.3 เอกสารและงานวิจัยที่เกี่ยวข้อง

ในงานวิจัยที่เสนออัลกอริทึมในการหาเส้นทางบนระบบเครือข่ายที่กำหนดโดย ซอฟต์แวร์ สามารถจำแนกได้เป็นสองกลุ่ม ขึ้นอยู่กับลักษณะมุมมองในการตัดสินใจเลือกเส้นทาง ได้ดังนี้

2.3.1 มุมมองในการตัดสินใจเลือกเส้นทางของอัลกอริทึมในการหาเส้นทางบนพื้นฐานของการใช้ตัวชี้วัดของระบบเครือข่ายที่เฉพาะเจาะจง

มุมมองในการตัดสินใจเลือกเส้นทางของอัลกอริทึมในการหาเส้นทางบนพื้นฐานของการใช้ตัวชี้วัดของระบบเครือข่ายที่เฉพาะเจาะจง เช่น ค่าความกว้างแถบความถี่ (Bandwidth) ที่สามารถใช้ได้ ณ เวลาปัจจุบัน และค่าความล่าช้าการแพร่กระจายบนเส้นทางที่ได้รับการคัดเลือก เพื่อสนับสนุนการทำงานของ การควบคุมคุณภาพการให้บริการ (QoS) โดยในของวิจัยของ S. Tomovic et al. [3] ได้เสนออัลกอริทึมในการหาเส้นทางพิจารณาค่าความกว้างแถบความถี่ และค่าความล่าช้าของการส่งข้อมูล รวมไปถึงโมเดลของส่วนควบคุม (Control Plane) ซึ่งจะสามารถจำแนกประเภทการไหลของการรับส่งข้อมูลตามระดับความล่าช้าของส่งข้อมูล นอกจากนี้วิจัยดังกล่าวแล้ว ยังมีงานวิจัยของ A. Al-Jawad et al. [4] ได้เสนออัลกอริทึมในการหาเส้นทาง ที่มีชื่อว่า BaProbSDN โดยเป็นอัลกอริทึมในการหาเส้นทางของการควบคุมคุณภาพการให้บริการ (QoS) ที่อาศัยหลักการของความน่าจะเป็น ซึ่งในงานวิจัยนี้จะใช้ทฤษฎีบทของเบย์ (Bayes' Theorem) ซึ่งเป็นทฤษฎีที่ใช้หลักการของความน่าจะเป็นแบบมีเงื่อนไข และเครือข่ายงานแบบเบย์ (Bayesian Network) ซึ่งเป็นแบบจำลองความน่าจะเป็นเชิงกราฟที่แทน กลุ่มแบบไม่ขึ้นต่อกันอย่างมีเงื่อนไข มาใช้ในการตัดสินใจเส้นทางโดยใช้ค่าความจะเป็นแต่ละเส้นทาง โดยพิจารณาค่าความกว้างแถบความถี่ (Bandwidth) ที่สามารถใช้ได้ แต่อย่างไรก็ตามอัลกอริทึมในการหาเส้นทางบนพื้นฐานของการใช้ตัวชี้วัดของระบบเครือข่ายที่เฉพาะเจาะจงเพียงอย่างเดียวนั้น อาจสามารถทำให้ประสิทธิภาพของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์นั้นลดลงและเกิดความเสียหายได้อย่างรวดเร็ว เนื่องจากปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ (Bufferbloat Problem) เมื่อมีจำนวนการไหลของการรับส่งข้อมูลบนเส้นทางที่เลือกเพิ่มขึ้น

2.3.2. มุมมองในการตัดสินใจเลือกเส้นทางของอัลกอริทึมในการหาเส้นทางบนพื้นฐานของการใช้ตัวชี้วัดของระบบเครือข่ายที่เฉพาะเจาะจง

มุมมองในการตัดสินใจเลือกเส้นทางของอัลกอริทึมในการหาเส้นทางที่พิจารณาเกี่ยวกับทรัพยากรของระบบเครือข่าย เช่น ทรัพยากรของการเชื่อมโยงในแง่ของค่าความกว้างแถบความถี่ (Bandwidth) ที่สามารถใช้ได้ และทรัพยากรของโหนดในแง่ของ Ternary Content Addressable Memory (TCAM) ซึ่งเป็นหน่วยความจำของอุปกรณ์ส่งต่อกลุ่มข้อมูล หรือ สวิตช์บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ ซึ่งในงานวิจัยที่พิจารณาทรัพยากรของโหนด มีงานวิจัยของ P. M. Mohan et al. [5] เสนออัลกอริทึมในการจัดหาเส้นทางใหม่ที่พิจารณา Ternary Content Addressable Memory (TCAM) ภายในสวิตช์เพื่อปรับปรุงอัลกอริทึมในการหาเส้นทางที่ระยะสั้นที่สุดเพื่อสร้างเส้นทางที่ใช้ในการรับส่งข้อมูลโดยใช้ปริมาณ TCAM ของ Forwarding Entries ให้น้อยที่สุด และนอกจากการจัดหาเส้นทางใหม่แล้ว ยังทำให้การจัดหาเส้นทางสำรองอีกด้วย ส่วนงานวิจัยที่พิจารณาทรัพยากรของค่าความกว้างแถบความถี่ (Bandwidth) มีงานวิจัยของ D. Lee et al. [6] เสนออัลกอริทึมในการจัดหาเส้นทางที่มีชื่อว่า RPA-RA หรือ Resource Preference Aware Routing Algorithm ซึ่งเป็นอัลกอริทึมในการจัดหาเส้นทาง บนพื้นฐานของอัลกอริทึมในการหาเส้นทางที่ระยะสั้นที่สุด จำนวน K เส้นทาง หรือ K-Shortest Path Algorithm ซึ่งเป็นอัลกอริทึมที่ไม่ใช่เพียงแค่หาเส้นทางที่ระยะสั้นที่สุด เพียงเส้นทางเดียวเท่านั้น แต่จะทำการให้เส้นทางที่ระยะสั้นที่สุดทั้งหมดระบบเครือข่ายเป็นจำนวน K เส้นทาง และการตัดสินใจด้วยวิธีการวิเคราะห์เชิงลำดับชั้น (Analytic Hierarchy Process : AHP) ซึ่งเป็นรูปแบบการแยกแยะปัญหาที่ซับซ้อนให้ง่ายขึ้น โดยแยกออกเป็นลำดับชั้นและนอกจากนั้นมีการให้น้ำหนักเพื่อเปรียบเทียบในแต่ละปัญหาในแต่ละลำดับชั้น ซึ่งจะประกอบไปด้วยหลายหลากทางเลือกที่แตกต่างจากกันไป จนในที่สุดจะได้ทางเลือกตามที่ต้องการ โดยในงานวิจัยดังกล่าวมีจุดมุ่งหมายเพื่อที่จะรักษาความสมดุลของปริมาณการใช้งานความกว้างแถบความถี่ (Bandwidth) และทรัพยากรของ Flow Table ต่อมางานวิจัยของ M. Huang et al. [7] เสนออัลกอริทึมในการหาเส้นทางที่พิจารณาแบบจำลองต้นทุน หรือ Cost Model โดยศึกษาและเก็บรวบรวมข้อมูลการรับค่าขอของ Unicast และ Multicast แบบไดนามิกบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ เพื่อเพิ่มปริมาณงานของระบบเครือข่าย ต่อมางานวิจัยของ F. Francois et al. [8] เป็นงานวิจัยที่เสนอ Cognitive Routing Algorithm ที่ใช้ Cognitive Packet Network (CPN) โดยใช้หลักการของ Random Neural Network (RNN) กับ Reinforcement Learning (RL) ซึ่งเป็นหนึ่งในรูปแบบของแบบจำลองในการเรียนรู้ของ

เครื่อง หรือ Machine Learning เพื่อเพิ่มประสิทธิภาพให้เหมาะสมกับการค้นหาเส้นทางที่ต้องการการควบคุมคุณภาพการให้บริการ (QoS) ที่สูง นอกจากนี้ มีการทำการจำลองเสมือน ในงานวิจัยของ A. Xifra Porxas et al. [9] โดยเป็นเทคนิคอีกรูปแบบหนึ่งในการคำนวณหาเส้นทางได้ ซึ่งที่จะสามารถนำมาประยุกต์ใช้ในการจัดการทรัพยากรระบบเครือข่ายได้ โดยทำการแบ่งชั้นกายภาพ หรือ Physical Layer เป็นโครงสร้างเครือข่ายย่อยที่แยกออกจากกัน ซึ่งในงานวิจัยในมุมมองในการตัดสินใจเลือกเส้นทางของอัลกอริทึมในการหาเส้นทางที่พิจารณาเกี่ยวกับทรัพยากรของระบบเครือข่ายถือได้ว่าเป็นมุมมองที่เหมาะสมกับใช้ในการตัดสินใจเลือกเส้นทาง เพราะว่ามีมุมมองดังกล่าวสามารถกระจายการไหลของการรับส่งข้อมูลโดยรวมได้ โดยใช้ข้อมูลเกี่ยวกับทรัพยากรของระบบเครือข่าย

จากงานวิจัยที่กล่าวมาข้างต้นนั้น จะเห็นได้ว่ายังไม่มีงานวิจัยบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ที่เสนออัลกอริทึมในการหาเส้นทางที่พิจารณาเกี่ยวกับ ปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ (Bufferbloat Problem) ซึ่งงานวิจัยของเราเป็นงานวิจัยแรกบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ที่พิจารณาสถานะของบัฟเฟอร์บนอุปกรณ์ส่งต่อกลุ่มข้อมูล เพื่อตัดสินใจในการเลือกเส้นทาง ดังที่แสดงในตารางที่ 1

ตารางที่ 1 การเปรียบเทียบงานวิจัยที่เกี่ยวข้อง

Topic	Based on	Overhead of Control Plane	Overhead of Data Plane	Bandwidth	Delay	Buffer
Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks [3]	K Shortest Paths	○	○	○	○	X
TCAM-aware Local Rerouting for Fast and Efficient Failure Recovery in Software Defined Networks [5]	Shortest Path	○ TCAM	X	X	X	X
Towards a Cognitive Routing Engine for Software Defined Networks [8]	Shortest Path with Reinforcement Learning	X	○	○	○	X

QoS-Aware Virtualization-Enabled Routing in Software-Defined Networks [9]	Shortest Path	X	O	O	O	X
Dynamic Routing for Network Throughput Maximization in Software-Defined Networks [7]	Shortest Path	X	O	O	X	X
BaProbSDN: A Probabilistic-based QoS Routing Mechanism for Software Defined Networks [4]	Shortest Path	X	O	O	X	X
Queuing-Aware Routing Algorithm in Software Defined Networks	Shortest Path	O	X	X	X	O

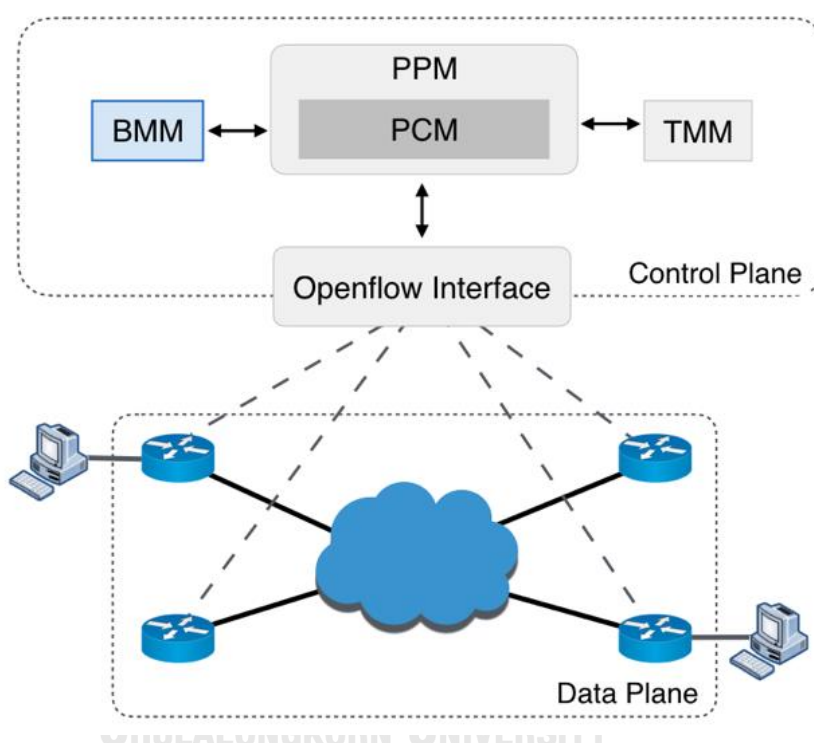
หมายเหตุ O คือ พิจารณา และ X คือ ไม่พิจารณา

บทที่ 3

แนวคิดและวิธีการดำเนินงานวิจัย

สำหรับงานวิจัยนี้จะแบ่งออกเป็น 2 ส่วน การปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์ และอัลกอริทึมการเส้นทางที่หลีกเลี่ยงความแออัดในระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

3.1 การปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์



ภาพที่ 4 องค์ประกอบภายในคอนโทรลเลอร์

การปรับปรุงองค์ประกอบภายในของคอนโทรลเลอร์นั้น จะนำองค์ประกอบมาตรฐานภายในคอนโทรลเลอร์ของ POX Controller มาปรับปรุง โดยที่องค์ประกอบองค์ประกอบมาตรฐานภายในคอนโทรลเลอร์จะประกอบไปด้วยส่วนต่างๆ ดังนี้

3.1.1 Path Provision Module (PPM)

ทำหน้าที่ Handle OF Messages ทั้งหมดจากองค์ประกอบภายในคอนโทรลเลอร์ โดยภายในจะมีองค์ประกอบย่อยคือ Path Computation เมื่อ Path Provision Module ได้รับ Packet_In message จากสวิตช์ Path Provision Module จะทำการส่งต่อข้อความไปยัง Path Computation Module เพื่อคำนวณหาเส้นทางต่อไป

3.1.2 Path Computation Module (PCM)

ทำหน้าที่ในการหาเส้นทางที่ระยะทางสั้นที่สุด ในการส่งข้อมูล โดยใช้ข้อมูล Topology ที่จากการ Topology Management Module จากนั้นจะส่งผลการคำนวณเส้นทางกลับไปยัง Path Provision Module และทำการสร้าง FlowMod Messages ซึ่งเป็นชนิดข้อความของเกณฑ์วิธี OpenFlow เพื่อใช้ในการติดตั้ง Forwarding Rules ในสวิตช์ จากนั้นจะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้

3.1.3 Topology Management Module (TMM)

ทำหน้าที่คอยตรวจสอบและปรับปรุง Topology ของระบบเครือข่ายที่เปลี่ยนแปลงไปตามเวลาจริงเมื่อมีการเปลี่ยนแปลงของ Topology ในระบบเครือข่าย โดยในการตรวจสอบและปรับปรุง Topology จะใช้ Link Layer Discovery Protocol (LLDP) เกณฑ์วิธีเพื่อใช้ในการค้นหา Topology ในระบบเครือข่ายด้วย OpenFlow Messages

โดยการปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์จะเพิ่มองค์ประกอบในการคอยตรวจสอบและเรียกค่าการครอบครองพื้นที่ในบัฟเฟอร์ตามเวลาจริง โดยจะเรียกว่า Buffer Monitoring Module (BMM) ซึ่งค่าการครอบครองพื้นที่ในบัฟเฟอร์ จะเป็นค่าที่ใช้ในการลบสวิตช์ที่มีโอกาสเสี่ยงสูงที่จะเกิดความแออัดบนเส้นทางที่พร้อมใช้งาน จากนั้นเมื่อรู้สวิตช์ที่มีโอกาสเสี่ยงสูงที่จะเกิดความแออัดแล้ว Buffer Monitoring Module จะทำการส่งต่อข้อมูลของสวิตช์นั้นไปยัง Path Provision Module เพื่อทำการคำนวณหาเส้นทางที่สั้นที่สุดใหม่อีกครั้ง และ Topology

Management Module จะทำการปรับปรุง Topology ของระบบเครือข่ายให้กับสวิตช์ที่เกี่ยวข้องทั้งหมด

3.2 อัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

สำหรับอัลกอริทึมในงานวิจัยนี้ จะเสนออัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิว ของข้อมูลที่หลีกเลี่ยงเส้นทางที่มีความแออัด โดยปกติแล้วสวิตช์จะมีกลไกในการบริหารจัดการปริมาณกลุ่มข้อมูลในอยู่ในบัฟเฟอร์ในแต่ละช่องทางของสวิตช์ เมื่อมีจำนวนปริมาณของกลุ่มข้อมูลในบัฟเฟอร์ จะแสดงเกิดความแออัดเกิดบนเส้นทางนั้น ดังนั้นอัลกอริทึมที่เสนอในงานวิจัยนี้ จะเลือกเส้นทาง ที่มีค่าการครอบครองพื้นที่ในบัฟเฟอร์น้อยกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมดเพื่อที่จะคำนวณหาเส้นทางที่มีระยะทางสั้นที่สุดได้อย่างมีประสิทธิภาพ คอนโทรลเลอร์ต้องการที่จะคำนวณหาค่าการครอบครองพื้นที่ในบัฟเฟอร์ในแต่ละช่องทางของแต่ละสวิตช์ โดยที่คอนโทรลเลอร์จะสอบถามค่าการครอบครองพื้นที่ในบัฟเฟอร์ ณ ปัจจุบันในทุกๆ 0.01 วินาทีเพื่อทำการปรับปรุงเส้นทางที่มีระยะทางสั้นที่สุดในแต่ละการไหลของการรับส่งข้อมูล โดยการเลือกเส้นทาง จะมีเปลี่ยนแปลงไปอย่างรวดเร็ว สำหรับในการคำนวณค่าการครอบครองพื้นที่ในบัฟเฟอร์ในแต่ละครั้ง โดยค่าการครอบครองพื้นที่ในบัฟเฟอร์ จะเป็นค่าเฉลี่ยที่มาจากคำนวณโดยใช้สมการที่มีการปรับให้มีค่าความราบเรียบ หรือสมการค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบเอกซ์โพเนนเชียล (Exponential Weighted Moving Average : EWMA)

$$\bar{B}_{p,s}^t = (1 - \alpha)\bar{B}_{p,s}^{t-1} + \alpha B_{p,s}^t ; 0 < \alpha < 1$$

โดย $\bar{B}_{p,s}^t$ คือ ค่าปริมาณบัฟเฟอร์เฉลี่ย ณ เวลาที่ t ของช่องทาง p บนสวิตช์ s

$B_{p,s}^t$ คือ ค่าปริมาณบัฟเฟอร์ ณ เวลาที่ t ของช่องทาง p บนสวิตช์ s

$\bar{B}_{p,s}^{t-1}$ คือ ค่าปริมาณบัฟเฟอร์เฉลี่ยก่อนหน้า ณ เวลาที่ $t-1$ ของช่องทาง p บนสวิตช์ s

α คือ ค่า Smoothing Factor โดยค่าอยู่ระหว่าง 0 กับ 1

หลังการที่คำนวณค่าปริมาณบัพเฟอร์เฉลี่ยของแต่ละช่องทางของแต่ละสวิตช์แล้วจากนั้นจะ
มาทำการหาค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบ
เอกซ์โพเนนเชียล (EWMA)

$$\sigma_{p,s}^t = (1 - \alpha)\sigma_{p,s}^{t-1} + \alpha|B_{p,s}^t - \bar{B}_{p,s}^t|; 0 < \alpha < 1$$

โดย $\sigma_{p,s}^t$ คือ ค่าเบี่ยงเบนมาตรฐาน ณ เวลาที่ t ของช่องทาง p บนสวิตช์ s

$B_{p,s}^t$ คือ ค่าปริมาณบัพเฟอร์ ณ เวลาที่ t ของช่องทาง p บนสวิตช์ s

$\sigma_{p,s}^{t-1}$ คือ ค่าเบี่ยงเบนมาตรฐานก่อนหน้า ณ เวลาที่ $t-1$ ของช่องทาง p
บนสวิตช์ s

$\bar{B}_{p,s}^t$ คือ ค่าปริมาณบัพเฟอร์เฉลี่ย ณ เวลาที่ t ของช่องทาง p บนสวิตช์ s

α คือ ค่า Smoothing Factor โดยค่าอยู่ระหว่าง 0 กับ 1

เมื่อได้ค่าปริมาณบัพเฟอร์เฉลี่ยและค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนัก
แบบเอกซ์โพเนนเชียลแล้ว จะมาสามารถคำนวณหาค่าการครอบครองพื้นที่ในบัพเฟอร์ ได้จากสมการ
ดังนี้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

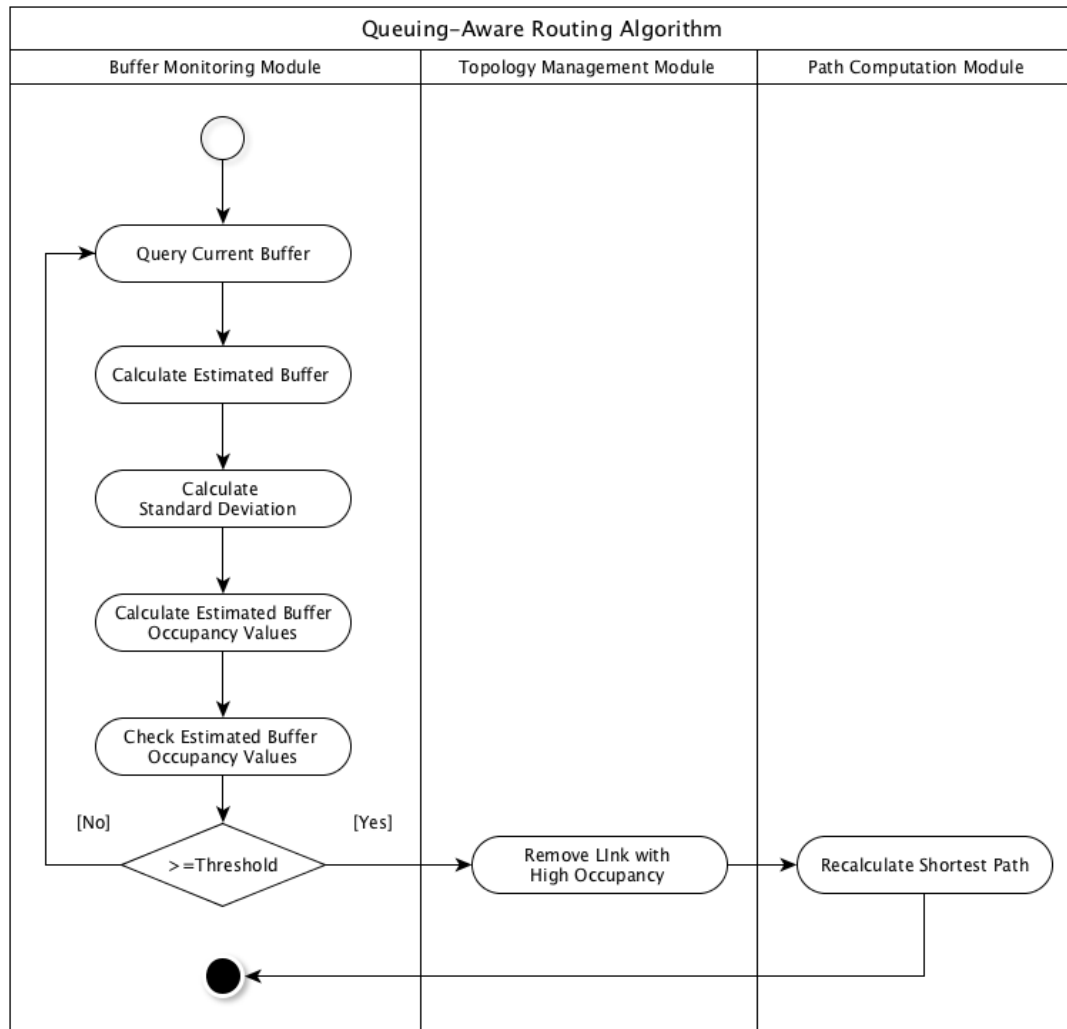
$$\beta_{p,s} = \bar{B}_{p,s}^t + |\sigma_{p,s}^t|$$

โดย $\beta_{p,s}$ คือ ค่าการครอบครองพื้นที่ในบัพเฟอร์ของช่องทาง p บนสวิตช์ s

$\sigma_{p,s}^t$ คือ ค่าเบี่ยงเบนมาตรฐาน ณ เวลาที่ t ของช่องทาง p บนสวิตช์ s

$\bar{B}_{p,s}^t$ คือ ค่าปริมาณบัพเฟอร์เฉลี่ย ณ เวลาที่ t ของช่องทาง p บนสวิตช์ s

เพื่อที่จะเลือกเส้นทางที่มีระยะทางที่สั้นที่สุดโดยพิจารณาจากค่าการครอบครองพื้นที่ในบัฟเฟอร์นั้นสามารถอธิบายการทำงานตามแผนภาพได้ดังนี้



ภาพที่ 5 แผนภาพขั้นตอนการทำงานของอัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ของแต่ละองค์ประกอบภายในคอนโทรลเลอร์

3.2.1 Query Current Buffer

ทำหน้าที่ในการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์

3.2.2 Calculate Estimated Buffer

ทำหน้าที่ในการคำนวณหาค่าปัจจุบันของปริมาณข้อมูลที่อยู่บัฟเฟอร์เฉลี่ยของทีมาจากคำนวณโดยใช้ สมการค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบเอกซ์โพเนนเชียล (EWMA)

3.2.3 Calculate Standard Deviation

เมื่อได้ค่าปริมาณบัฟเฟอร์เฉลี่ยของแต่ละช่องทางของแต่ละสวิตช์แล้ว จากนั้นจะมาทำการหาค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบเอกซ์โพเนนเชียล (EWMA)

3.2.4 Calculate Estimated Buffer Occupancy Values

นำค่าปริมาณบัฟเฟอร์เฉลี่ยและค่าเบี่ยงเบนมาตรฐานของค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบเอกซ์โพเนนเชียล มาคำนวณหาค่าการครอบครองพื้นที่ในบัฟเฟอร์

3.2.5 Check Estimated Buffer Occupancy Values

เมื่อได้ค่าการครอบครองพื้นที่ในบัฟเฟอร์แล้ว ในส่วนนี้จะทำการตรวจสอบว่าค่าปัจจุบันของการครอบครองพื้นที่ในบัฟเฟอร์ มีค่ามากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมดหรือไม่ ถ้าไม่เกินเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด จะทำการวนซ้ำไปที่กระบวนการ Query Current Buffer ใหม่อีกครั้ง โดยที่กระบวนการการวนซ้ำจะทำงานทุกๆ 0.01 วินาที เนื่องจากค่าปริมาณบัฟเฟอร์นั้น จะมีเปลี่ยนแปลงไปอย่างรวดเร็ว

3.2.6 Remove Link with High Occupancy

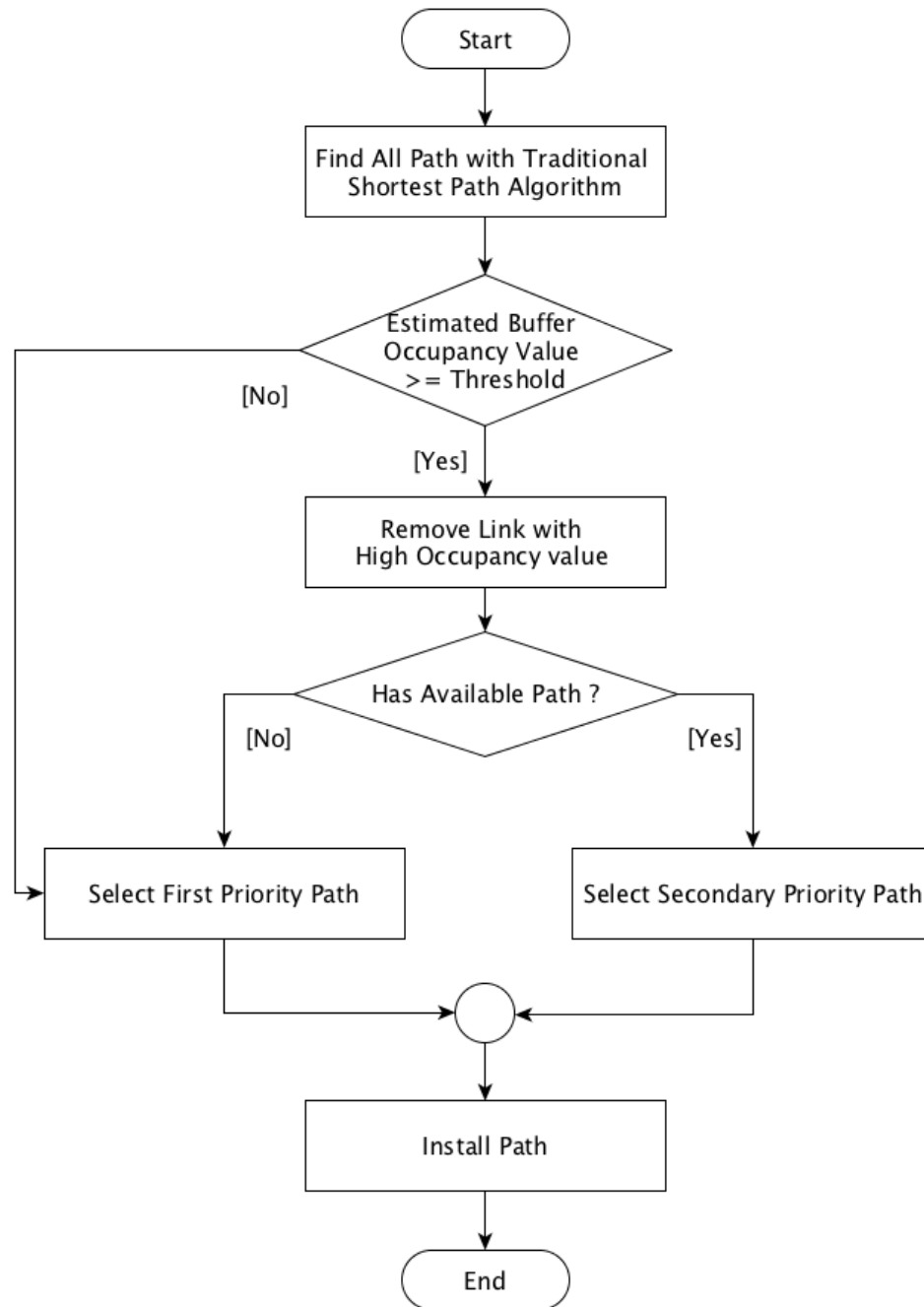
ทำหน้าที่ในการลบการเชื่อมโยงของสวิตช์นั้น ออกจากการคำนวณหาเส้นทางที่มีระยะทางที่สั้นที่สุดในครั้งต่อไป ที่มีการไหลของการรับส่งข้อมูลใหม่เข้ามาในระบบเครือข่าย โดยกระบวนการนี้จะอยู่ในองค์ประกอบของ Topology Management Module

3.2.5 Recalculate Shortest Path

ทำหน้าที่ในการคำนวณหาเส้นทางที่สั้นที่สุดใหม่อีกครั้ง โดยคำนวณจาก Topology ที่มาจากปรับปรุง Topology ใหม่ของระบบเครือข่าย โดยกระบวนการนี้จะอยู่ในองค์ประกอบของ Path Provision Module

โดยการตัดสินใจเลือกและติดตั้งเส้นทางสำหรับของอัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์มีลักษณะการทำงานตามขั้นตอนดังนี้

เมื่อมีการไหลของการรับส่งข้อมูลเข้ามาใหม่ในระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์คอนโทรลเลอร์จะทำการคำนวณหาเส้นทางที่ระยะทางสั้นที่สุดทั้งหมดที่เกิดขึ้นในระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ ซึ่งจะมีเส้นทางที่แตกต่างกันไปตามทอพอโลยีที่นำไปใช้ โดยใช้อัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm จากนั้นเมื่อได้เส้นทางที่มีระยะทางที่สั้นที่สุดมาแล้ว ก็จะมีการตรวจสอบว่ามีค่าปัจจุบันของการครอบครองพื้นที่ในบัฟเฟอร์ในเฉลี่ยแต่ละช่องทางของสวิตช์ตามเส้นทางที่คำนวณมาได้นั้น มีค่ามากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมดหรือไม่ ถ้าไม่เกินกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ก็จะมีการเลือกเส้นทางที่ระยะทางสั้นที่สุดอันดับแรก และทำการติดตั้งเส้นทางดังกล่าวไปยังสวิตช์ที่เกี่ยวข้อง แต่ถ้าเกินกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด จะทำการลบการเชื่อมโยงของสวิตช์นั้นออกจากการคำนวณหาเส้นทางที่มีระยะทางที่สั้นที่สุดในครั้งต่อไป ต่อมาก่อนที่จะทำการคำนวณหาเส้นทางที่ระยะทางสั้นที่สุดทั้งหมดใหม่ ก็จะมีการตรวจสอบว่าเส้นทางที่เหลืออยู่นั้นมีเส้นทางที่มีระยะทางสั้นที่สุดอันดับรองลงมาหรือไม่ ถ้ามีจากนั้นคอนโทรลเลอร์ก็จะทำการเลือกเส้นทางที่มีระยะทางที่สั้นที่สุดอันดับรองลงมา และทำการติดตั้งเส้นทางดังกล่าวไปยังสวิตช์ที่เกี่ยวข้อง แต่ถ้าไม่มีเส้นทางที่มีระยะทางที่สั้นที่สุดอันดับรองลงมา ก็จะมีการกลับไปเลือกเส้นทางระยะทางที่สั้นที่สุดในอันดับแรกก่อนที่จะมีการลบการเชื่อมโยงของสวิตช์ในเส้นทางที่เกิดความแออัด จากนั้นทำการติดตั้งเส้นทางดังกล่าวไปยังสวิตช์ที่เกี่ยวข้อง ซึ่งทุกครั้งที่มีการไหลของการรับส่งข้อมูลเข้ามาใหม่ ในระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ จะทำตามขั้นตอนตามแผนภาพที่ 6



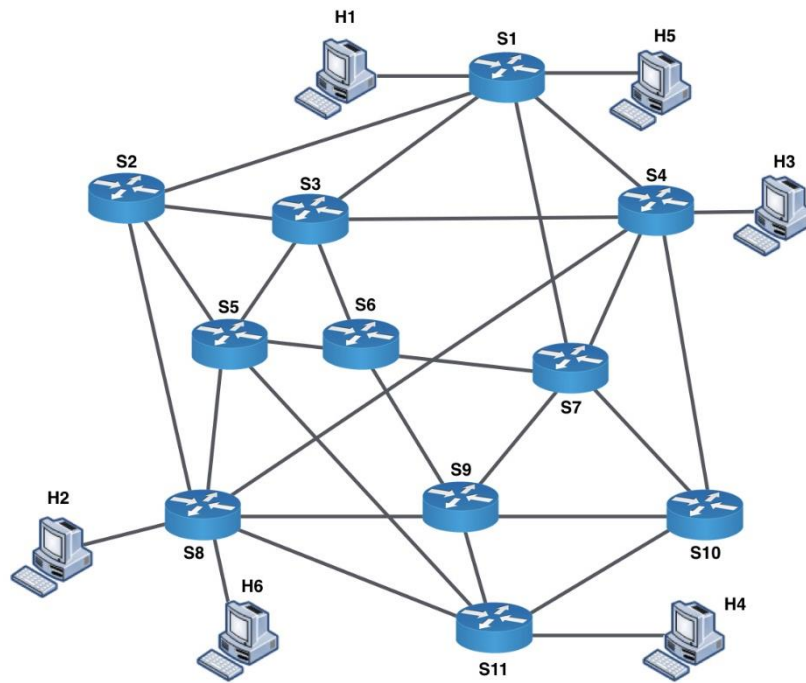
ภาพที่ 6 แผนภาพขั้นตอนการตัดสินใจเลือกและติดตั้งเส้นทางสำหรับการไหลของการรับส่งข้อมูลที่เข้ามาใหม่ของอัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูลบนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

บทที่ 4

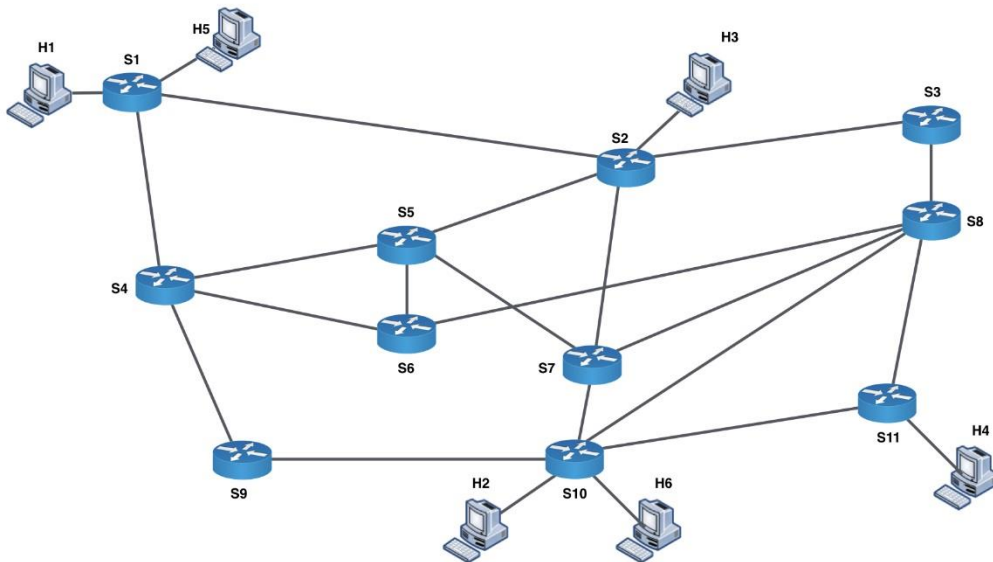
การทดลองและผลการทดลอง

ในบทนี้จะอธิบายถึงการนำอัลกอริทึมการเส้นทางที่ได้ออกแบบไว้ในบทที่ 3 นำมาทำการทดลองและวัดประสิทธิภาพโดยทำการสร้างระบบเครือข่ายเสมือนจริงจากโปรแกรมเลียนแบบพฤติกรรมการทำงานของระบบเครือข่ายหรือ Mininet ซึ่ง Mininet เป็นโปรแกรมที่จะช่วยให้สร้างระบบเครือข่ายที่เลียนแบบการทำงานของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ บนคอมพิวเตอร์เครื่องเดียวได้ ไม่ว่าจะเป็น OpenFlow สวิตช์ และเครื่องเสมือนจากนั้นแล้ว OpenFlow คอลโทรเลอร์ ที่จะใช้ซอฟต์แวร์ชื่อว่า POX คอนโทรลเลอร์ โดยที่ POX คอนโทรลเลอร์เป็นคอลโทรเลอร์ที่ค่อนข้างเร็วและมีประสิทธิภาพสามารถรองรับปริมาณของ flow entries ได้มากกว่า 3,000 flow ต่อวินาที โดยในงานวิจัยนี้ทำการทดลองบนทอพอโลยี COST239 [15] ซึ่งเป็นทอพอโลยีใยแก้วนำแสงของข่ายงานบริเวณกว้าง หรือ WAN ที่เชื่อมต่อระหว่างหัวเมืองใหญ่ของทวีปยุโรปและทอพอโลยี SPLINTLINK [16] ซึ่งเป็นทอพอโลยีใยแก้วนำแสงของข่ายงานบริเวณกว้าง หรือ WAN ที่เชื่อมต่อระหว่าง รัฐต่างๆ ของประเทศสหรัฐอเมริกา โดยภายในทอพอโลยี COST239 และทอพอโลยี SPLINTLINK ประกอบไปด้วยอุปกรณ์จุดปลาย 6 เครื่องซึ่งประกอบไปด้วยผู้ส่ง 3 ผู้ส่ง (H1 H3 และ H5) และผู้รับ 3 ผู้รับ (H2 H4 และ H6) สำหรับก่อกำเนิดการไหลของการรับส่งข้อมูล ซึ่งจะแสดงองค์ประกอบต่างๆ ภายในทอพอโลยีที่ใช้ในการทดลองของทั้งสองทอพอโลยีดังแสดงในภาพที่ 6 และภาพที่ 7

โดยผู้ส่งทั้งหมดจะใช้โปรแกรม iPerf [17] ซึ่งเป็นเครื่องมือที่ใช้ในการวัดความกว้างแถบความถี่ด้วยเกณฑ์วิธี TCP เพื่อก่อกำเนิดปริมาณการใช้การไหลของการรับส่งข้อมูลประเภท TCP โดยในระหว่างทำการทดลอง แต่ละการไหลของการรับส่งข้อมูลประเภท TCP จะถูกป้อนเข้าไปในระบบเครือข่าย ด้วยระยะเวลาที่ต่างกัน 5 วินาที ซึ่งเป็นช่วงเวลาที่การไหลของการรับส่งข้อมูลประเภท TCP ในแต่ละคู่ส่งจะเริ่มทำการส่งข้อมูล โดยลำดับการส่งข้อมูลที่จะถูกป้อนเข้าไปในระบบเครือข่าย จะเริ่มจากคู่ส่ง TCP ระหว่าง H1 และ H2 ต่อมาเป็นคู่ส่ง TCP ระหว่าง H5 และ H6 และคู่ส่ง TCP ระหว่าง H3 และ H4 ตามลำดับ โดยการตั้งค่าพารามิเตอร์ในการทดลองดังกล่าว จะแสดงในตารางที่ 2



ภาพที่ 7 ทอพอโลยี COST239



ภาพที่ 8 ทอพอโลยี SPLINTLINK

ตารางที่ 2 การตั้งค่าพารามิเตอร์

พารามิเตอร์	COST239	SPLINTLINK
ความกว้างแถบความถี่ของแต่ละการเชื่อมโยงเครื่องจุดปลาย (Mbits/s)	1000	1000
ความกว้างแถบความถี่ของแต่ละการเชื่อมโยงระหว่างแต่ละสวิตช์ (Mbits/s)	1	1
ความล่าช้าของแต่ละการเชื่อมโยง (ms)	10	10
เวลาที่ใช้ในการส่งของแต่ละการไหลของการรับส่งข้อมูล (s)	180	180
เวลาเริ่มต้นของการไหลของการรับส่งข้อมูลคู่ที่ 1 (H1-H2) (s)	5	5
เวลาเริ่มต้นของการไหลของการรับส่งข้อมูลคู่ที่ 2 (H5-H6) (s)	10	10
เวลาเริ่มต้นของการไหลของการรับส่งข้อมูลคู่ที่ 3 (H3-H4) (s)	15	15

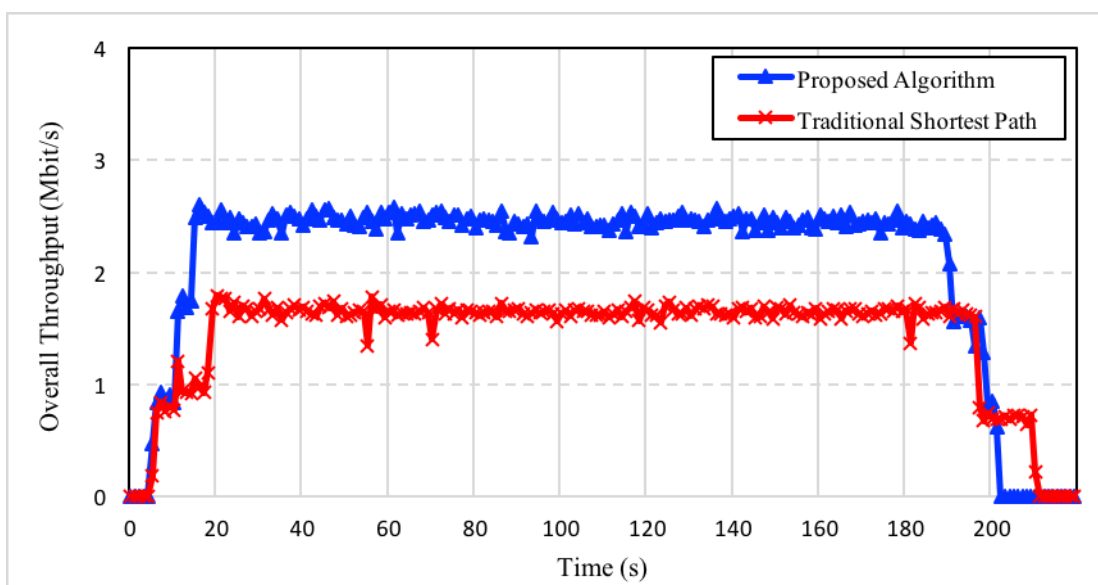
4.1 ประสิทธิภาพของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239

ในการวัดประสิทธิภาพการทำงานของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 จะใช้ตัววัดประสิทธิภาพการทำงานของอัลกอริทึมอยู่ด้วยกัน 3 ตัววัด ดังนี้

- ผลรวมของปริมาณงาน (Overall Throughput)
- ช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time)
- ความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss)

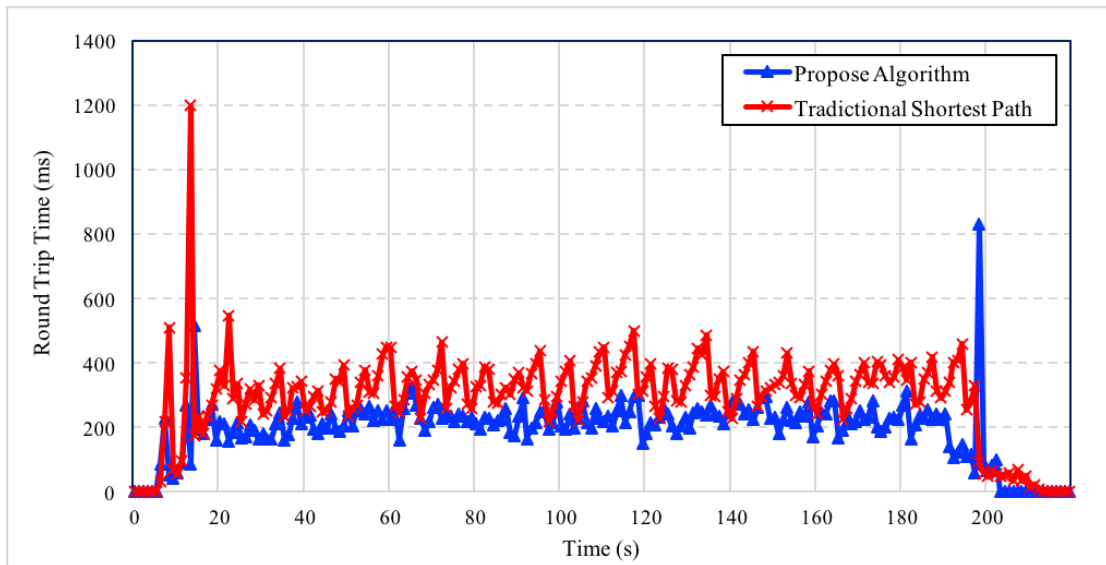
ผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ที่แสดงดังภาพที่ 9 จะแสดงที่แกน Y ของกราฟในหน่วย เมกะบิตต่อวินาที (Mbit/s) และแกน X จะแสดงเวลาที่ใช้ในการทำการทดลองในหน่วยวินาที โดยเส้นแดงจะเป็นผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ส่วนเส้นสีฟ้าผลรวมของปริมาณงาน

(Overall Throughput) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้



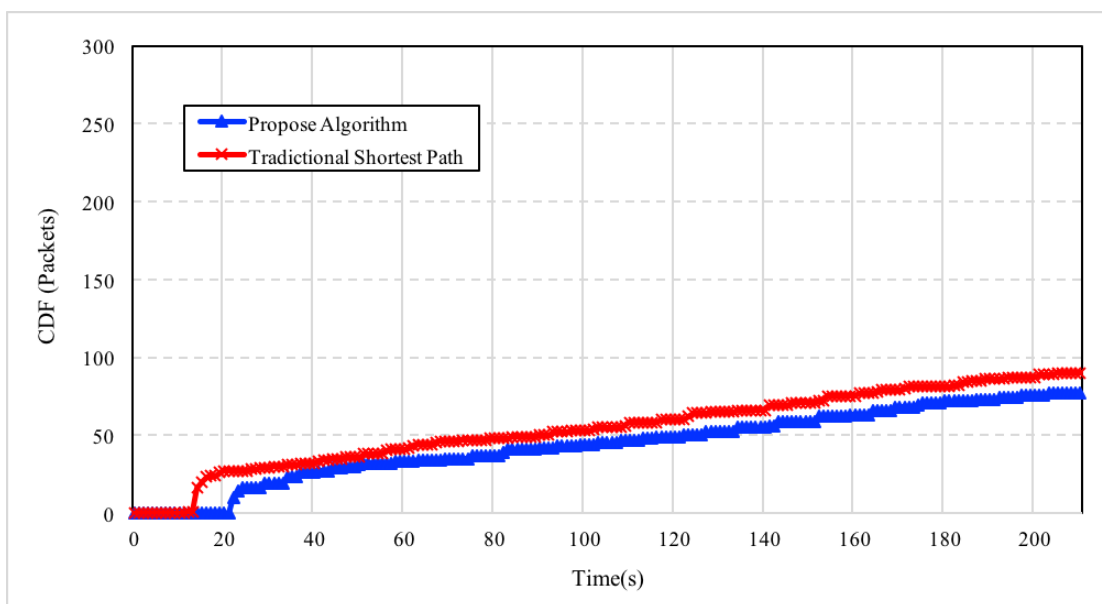
ภาพที่ 9 กราฟการเปรียบเทียบผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239

ช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ที่แสดงดังภาพที่ 10 จะแสดงที่แกน Y ของกราฟในหน่วย มิลลิวินาที (ms) และแกน X จะแสดงเวลาที่ใช้ในการทำการทดลองในหน่วยวินาที โดยเส้นแดงจะเป็นช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ส่วนเส้นสีฟ้าจะเป็นช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้



ภาพที่ 10 กราฟการเปรียบเทียบช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239

ความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ที่แสดงดังภาพที่ 11 จะแสดงที่แกน Y ของกราฟในหน่วยจำนวนของกลุ่มข้อมูล (Packets) และแกน X จะแสดงเวลาที่ใช้ในการทำการทดลองในหน่วยวินาที โดยเส้นแดงจะเป็นความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ส่วนเส้นสีฟ้าจะเป็นความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้



ภาพที่ 11 กราฟการเปรียบเทียบความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239

ในกรณีแนวทางของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ที่เวลา 5 วินาที เมื่อเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งคู่แรกระหว่าง H1 และ H2 จากนั้น โพรโทคอลเลออร์จะทำการคำนวณหาเส้นทางที่สั้นที่สุดหรือ Shortest Path และทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S2 \rightarrow S8$ ซึ่งเป็นการไหลของการรับส่งข้อมูลประเภท TCP ที่เริ่มจะทำการส่งเป็นการไหลของการรับส่งข้อมูลแรก ต่อมาที่วินาทีที่ 10 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S2 \rightarrow S8$ ซึ่งเห็นได้ชัดเจนว่า คู่ส่งที่ 2 จะใช้เส้นทางที่ใช้ในการส่งข้อมูลเส้นทางเดียวกันกับคู่ส่งแรกเพราะว่าวิธีการแบบดั้งเดิมไม่ได้พิจารณาเกี่ยวกับปริมาณบัฟเฟอร์ที่เกิดขึ้นบนอุปกรณ์ระบบเครือข่าย และนอกจากนั้นวิธีการแบบดั้งเดิมจะเลือกเส้นทางที่ดีที่สุดจากระยะทางที่สั้นที่สุดเสมอ ซึ่งส่งผลให้ประสิทธิภาพของผลรวมของปริมาณงาน (Overall Throughput) ลดลงเนื่องจากคู่ส่งที่ 1 และคู่ส่งที่ 2 ได้ใช้เส้นทางที่ใช้ในการส่งร่วมกัน ต่อมาที่วินาทีที่ 15 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 3 ระหว่าง H3 และ H4 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้

คำนวณไว้สำหรับ Forwarding Path คือ $S4 \rightarrow S10 \rightarrow S11$ โดยค่าเฉลี่ยปริมาณงาน (Throughput) ของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งของแต่ละ การไหลของการรับส่งข้อมูล ประเภท TCP จะแสดงในตารางที่ 3

ตารางที่ 3 ค่าเฉลี่ยปริมาณงานของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งบน ทอพอโลยี COST239

Flow	อัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิม (Mbit/s)	อัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ (Mbit/s)
H1-H2	0.40	0.75
H5-H6	0.47	0.74
H3-H4	0.64	0.65

ในกรณีแนวทางของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ ที่เวลา 5 วินาที เมื่อเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งคู่แรกระหว่าง H1 และ H2 จากนั้น POX คอนโทรลเลอร์จะทำการคำนวณหาเส้นทางที่สั้นที่สุด หรือ Shortest Path และทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S2 \rightarrow S8$ ซึ่งเป็นการไหลของการรับส่งข้อมูลประเภท TCP ที่เริ่ม จะทำการส่งเป็นการไหลของการรับส่งข้อมูลแรก โดยในการติดตั้งเส้นทางมีลักษณะการทำงาน คล้ายคลึงกับวิธีการเลือกเส้นทางแบบดั้งเดิม เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์ น้อยกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมดบนเส้นทางที่ถูกเลือก หรือน้อยกว่าร้อยละ 70 ในกรณีนี้ ต่อมาที่วินาทีที่ 10 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S2 \rightarrow S8$ ซึ่งเห็นได้ชัดเจนว่า มีเส้นทางที่พร้อมใช้งานอยู่ 2 เส้นทางที่สอดคล้องกับระยะทางที่สั้นที่สุด คือ $S1 \rightarrow S2 \rightarrow S8$ และ $S1 \rightarrow S4 \rightarrow S8$ แต่อย่างไรก็ตามในเส้นทางแรก คือ $S1 \rightarrow S2 \rightarrow S8$ จะไม่ถูกเลือกใช้ในการส่งข้อมูล เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ดังนั้นเส้นทางที่สองที่จะใช้ในการส่งการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 จะเลือกเส้นทาง $S1 \rightarrow S4 \rightarrow S8$ ซึ่งจะถูกติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้ จากนั้นวินาทีที่ 15 จะเริ่มทำการเชื่อมต่อด้วย การไหลของการรับส่งข้อมูล ประเภท TCP ของคู่ส่งที่ 3 ระหว่าง H3 และ H4 ซึ่งจะทำให้การเลือกเส้นทาง $S4 \rightarrow S10 \rightarrow S11$ แทน

S4→S8→S11 เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัพเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัพเฟอร์ทั้งหมดบนเส้นทางที่ถูกเลือก

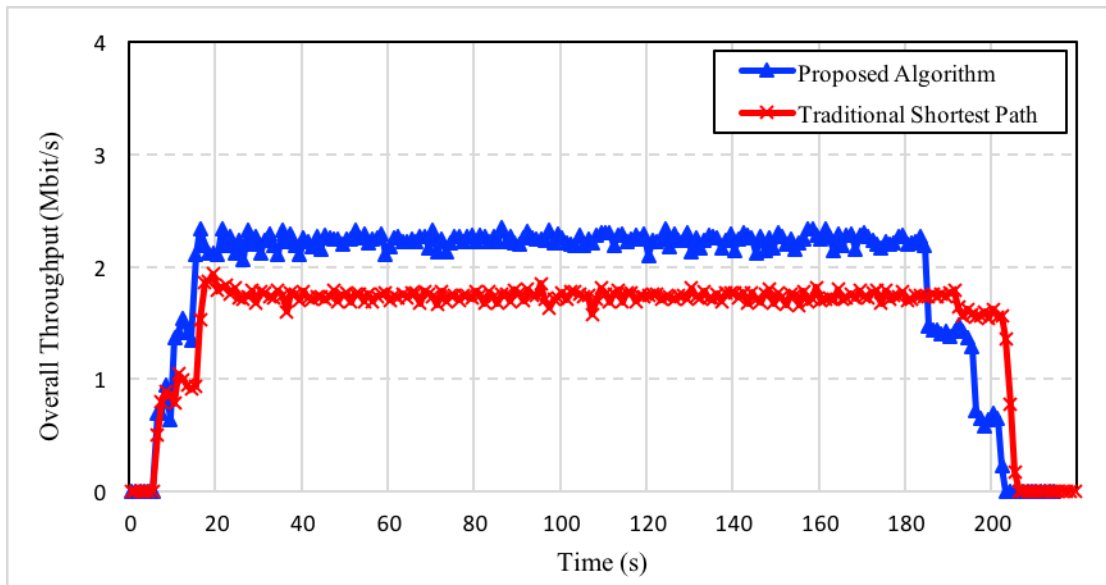
จากผลทดลองอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้ มีประสิทธิภาพกว่าอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ในแง่ของผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่ายถึงร้อยละ 46 แ่งของช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) เป็นจำนวนร้อยละ 48 และแง่ของความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) เป็นจำนวนร้อยละ 23 ซึ่งผลของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm จะยังคงแย่กว่าอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้ เนื่องจากในกรณีนี้จะเกิดปริมาณการใช้ที่มีความแออัดบนเส้นทาง S1→S2→S8 ตลอดจนการทำการทดลองหลังจากนั้นเมื่อเวลาในการส่งเพิ่มขึ้นเรื่อยๆ จนกระทั่งถึงเวลาที่ 180 วินาทีการเชื่อมต่อระหว่างของสองเครื่องจุดปลายจะสิ้นสุดการทำงานที่ละการไหลของการรับส่งข้อมูลตามลำดับ

4.2 ประสิทธิภาพของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี SPLINTLINK

ในการวัดประสิทธิภาพการทำงานของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี SPLINTLINK จะใช้ตัววัดประสิทธิภาพการทำงานของอัลกอริทึมอยู่ด้วยกัน 3 ตัววัด ดังนี้

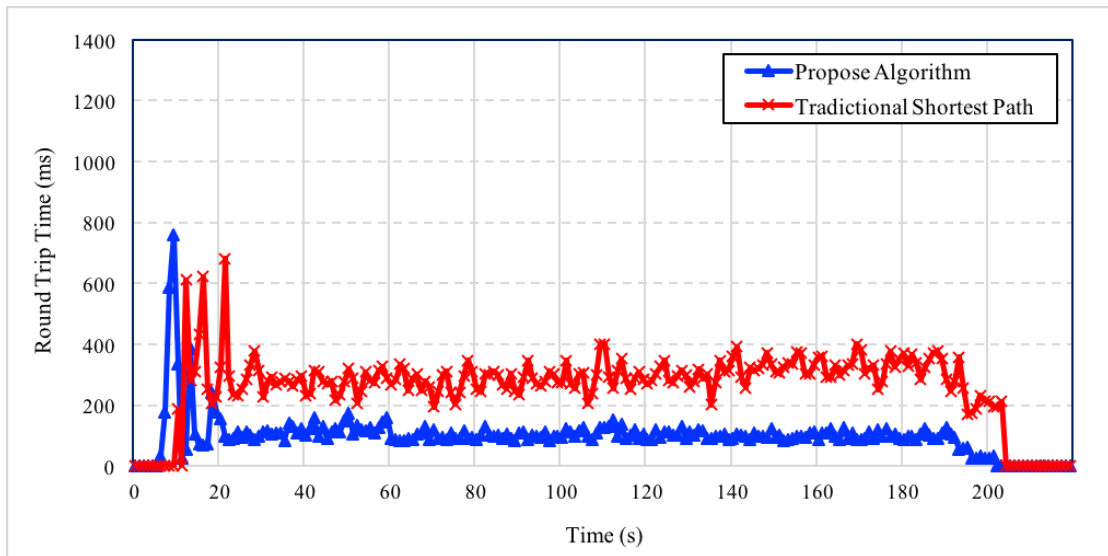
- ผลรวมของปริมาณงาน (Overall Throughput)
- ช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time)
- ความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss)

ในภาพที่ 12 ผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้ จะแสดงที่แกน Y ของกราฟในหน่วย เมกะบิตต่อวินาที (Mbit/s) และแกน X จะแสดงเวลาที่ใช้ในการทำการทดลองในหน่วยวินาที โดยเส้นแดงจะเป็นผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ส่วนเส้นสีฟ้าผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้



ภาพที่ 12 กราฟการเปรียบเทียบผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี SPLINTLINK

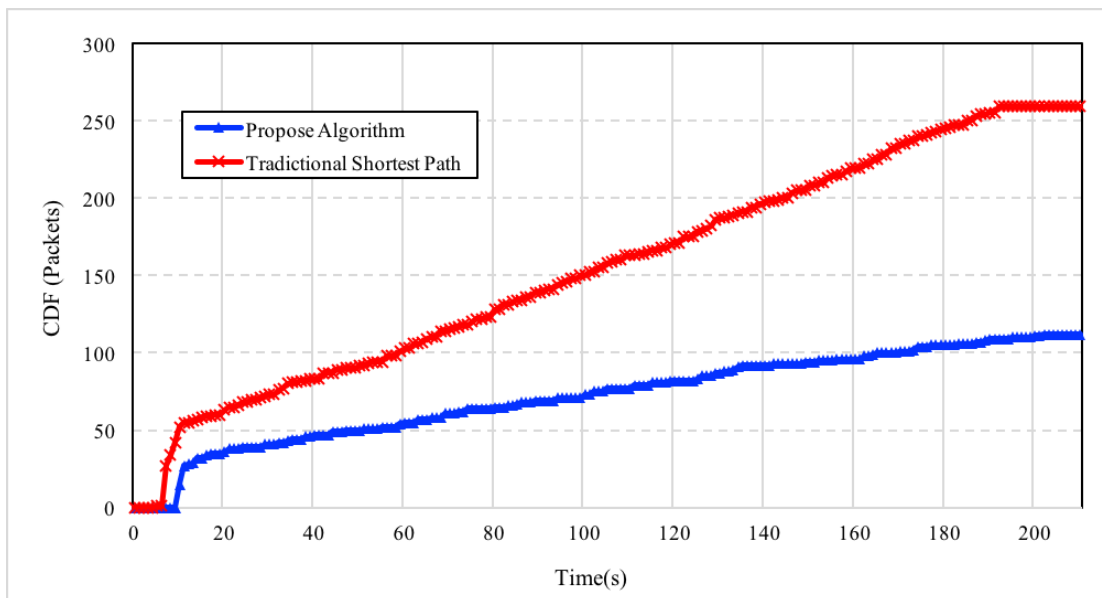
ช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ที่แสดงดังภาพที่ 13 จะแสดงที่แกน Y ของกราฟในหน่วย มิลลิวินาที (ms) และแกน X จะแสดงเวลาที่ใช้ในการทำการทดลองในหน่วยวินาที โดยเส้นแดงจะเป็นช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ส่วนเส้นสีฟ้าจะเป็นช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้



ภาพที่ 13 กราฟการเปรียบเทียบช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี

SPLINTLINK

ความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ที่แสดงดังภาพที่ 14 จะแสดงที่แกน Y ของกราฟในหน่วยจำนวนของกลุ่มข้อมูล (Packets) และแกน X จะแสดงเวลาที่ใช้ในการทำการทดลองในหน่วยวินาที โดยเส้นแดงจะเป็นความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ส่วนเส้นสีฟ้าจะเป็นความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้



ภาพที่ 14 กราฟการเปรียบเทียบความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี SPLINTLINK

ในกรณีแนวทางของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ที่เวลา 5 วินาที เมื่อเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งคู่แรกระหว่าง H1 และ H2 จากนั้น POX คอนโทรลเลอร์จะทำการคำนวณหาเส้นทางที่สั้นที่สุดหรือ Shortest Path และทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S4 \rightarrow S9 \rightarrow S10$ ซึ่งเป็น การไหลของการรับส่งข้อมูลประเภท TCP ที่เริ่มจะทำการส่งเป็น การไหลของการรับส่งข้อมูลแรก ต่อมาที่วินาทีที่ 10 จะเริ่มทำการเชื่อมต่อด้วย การไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S4 \rightarrow S9 \rightarrow S10$ ซึ่งเห็นได้ชัดเจนว่า คู่ส่งที่ 2 จะใช้เส้นทางที่ใช้ในการส่งข้อมูลเส้นทางเดียวกันกับคู่ส่งแรกเพราะว่าวิธีการแบบดั้งเดิมไม่ได้พิจารณาเกี่ยวกับปริมาณบัฟเฟอร์ที่เกิดขึ้นบนอุปกรณ์ระบบเครือข่าย และนอกจากนั้นวิธีการแบบดั้งเดิมจะเลือกเส้นทางที่ดีที่สุดจากระยะทางที่สั้นที่สุดเสมอ ซึ่งส่งผลให้ประสิทธิภาพของผลรวมของปริมาณงาน (Overall Throughput) ลดลงเนื่องจากคู่ส่งที่ 1 และคู่ส่งที่ 2 ได้ใช้เส้นทางที่ใช้ในการส่งร่วมกัน ต่อมาที่วินาทีที่ 15 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 3 ระหว่าง H3 และ H4 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่

อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S2 \rightarrow S7 \rightarrow S10 \rightarrow S11$ โดยค่าเฉลี่ยปริมาณงาน (Throughput) ของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งของแต่ละการไหลของการรับส่งข้อมูลประเภท TCP จะแสดงในตารางที่ 4

ตารางที่ 4 ค่าเฉลี่ยปริมาณงานของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งบนทอพอโลยี SPLINTLINK

Flow	อัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิม (Mbit/s)	อัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ (Mbit/s)
H1-H2	0.45	0.77
H5-H6	0.37	0.66
H3-H4	0.68	0.62

ในกรณีแนวทางของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ ที่เวลา 5 วินาที เมื่อเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งคู่แรกระหว่าง H1 และ H2 จากนั้น POX คอนโทรลเลอร์จะทำการคำนวณหาเส้นทางที่สั้นที่สุดหรือ Shortest Path และทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S4 \rightarrow S9 \rightarrow S10$ ซึ่งเป็นการไหลของการรับส่งข้อมูลประเภท TCP ที่เริ่มจะทำการส่งเป็นการไหลของการรับส่งข้อมูลแรก โดยในการติดตั้งเส้นทางมีลักษณะการทำงานคล้ายคลึงกับวิธีการเลือกเส้นทางแบบดั้งเดิม เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์น้อยกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมดบนเส้นทางที่ถูกเลือก หรือน้อยกว่าร้อยละ 70 ในกรณีนี้ ต่อมาที่วินาทีที่ 10 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S4 \rightarrow S9 \rightarrow S10$ ซึ่งเห็นได้ชัดเจนว่า มีเส้นทางที่พร้อมใช้งานอยู่ 2 เส้นทางที่สอดคล้องกับระยะทางที่สั้นที่สุด คือ $S1 \rightarrow S4 \rightarrow S9 \rightarrow S10$ และ $S1 \rightarrow S2 \rightarrow S7 \rightarrow S10$ แต่อย่างไรก็ตามในเส้นทางแรก คือ $S1 \rightarrow S4 \rightarrow S9 \rightarrow S10$ จะไม่ถูกเลือกใช้ในการส่งข้อมูล เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ดังนั้นเส้นทางที่สองที่จะใช้ในการส่งการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 จะเลือกเส้นทาง คือ $S1 \rightarrow S2 \rightarrow S7 \rightarrow S10$ ซึ่งจะถูกติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้ จากนั้นวินาทีที่ 15 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 3 ระหว่าง H3 และ H4 ซึ่งจะทำ

การเลือกเส้นทาง $S2 \rightarrow S3 \rightarrow S8 \rightarrow S11$ แทน $S2 \rightarrow S7 \rightarrow S10 \rightarrow S11$ เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัพเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัพเฟอร์ทั้งหมดบนเส้นทางที่ถูกเลือก

จากผลทดลองอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้ มีประสิทธิภาพกว่าอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ในแง่ของผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่ายถึงร้อยละ 37 แ่งของช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) เป็นจำนวนร้อยละ 165 และแง่ของความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) เป็นจำนวนร้อยละ 114 ซึ่งผลของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm จะยังคงแย่กว่าอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้ เนื่องจากในกรณีนี้จะเกิดปริมาณการใช้ที่มีความแออัดบนเส้นทาง $S1 \rightarrow S2 \rightarrow S8$ ตลอดการทำการทดลอง หลังจากนั้นเมื่อเวลาในการส่งเพิ่มขึ้นเรื่อยๆ จนกระทั่งถึงเวลาที่ 180 วินาทีการเชื่อมต่อระหว่างของสองเครื่องจุดปลายจะสิ้นสุดการทำงานที่ละการไหลของการรับส่งข้อมูลตามลำดับ

จากผลวัดประสิทธิภาพของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้ จากทั้งสองทอพอโลยี ทั้งทอพอโลยี COST239 และทอพอโลยี SPLINTLINK จะแสดงให้เห็นว่าอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้ มีประสิทธิภาพกว่าอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ในแง่ของผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่าย แ่งของช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) และแง่ของความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss)

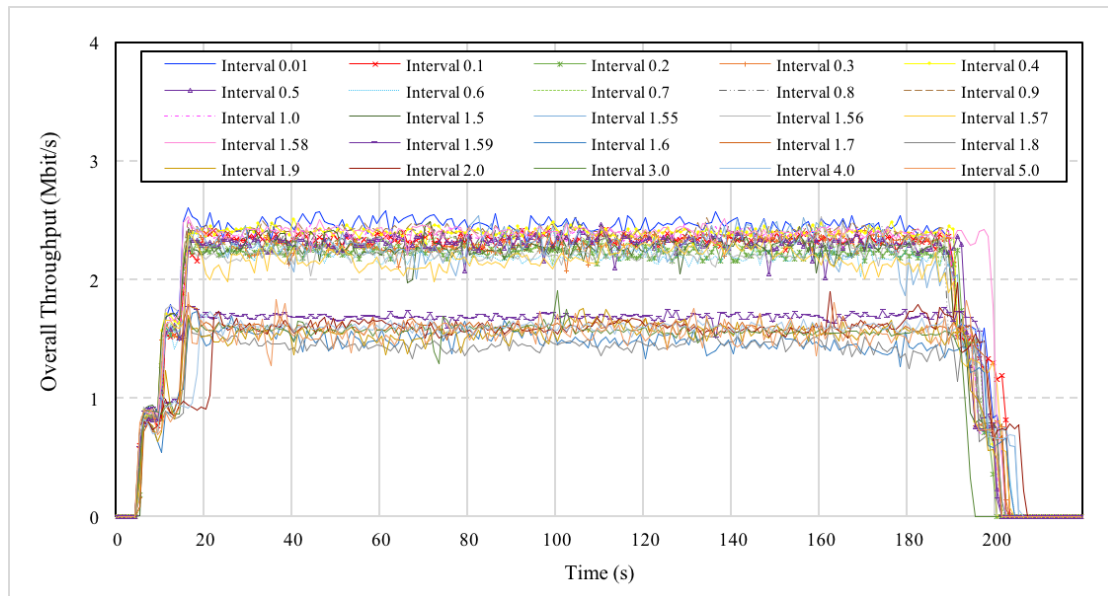
จุฬาลงกรณ์มหาวิทยาลัย

4.3 การปรับหาช่วงเวลาที่เหมาะสมในการเรียกค่าปริมาณบัพเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์

สำหรับในการทดลองนี้จะทำการทดลองเพื่อปรับหาช่วงเวลาที่เหมาะสมในการเรียกค่าปริมาณบัพเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์ จะแสดงดังภาพที่ 15 ผลรวมของปริมาณงาน (Overall Throughput) อัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้แต่ละช่วงเวลาจะแสดงที่แกน Y ของกราฟในหน่วยเมกะบิตต่อวินาที (Mbit/s) และแกน X จะแสดงเวลาที่ใช้ในการทำการทดลองในหน่วยวินาที โดยจะทำการทดลองเช่นเดียวกันการทดลองใน ข้อที่ 4.1 ประสิทธิภาพของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 รวมทั้งการตั้งค่าพารามิเตอร์ต่างๆ แต่จะแตกต่างกันที่การทดลองนี้จะทำการเริ่มปรับหาช่วงเวลาตั้งแต่ 0.01 วินาทีเป็นต้นไป เนื่องจาก 0.01 วินาทีเป็นข้อจำกัดของฟังก์ชัน Timer ในระบบเครือข่าย

เสมือนจริงจากโปรแกรมเลียนแบบพฤติกรรมการทำงานของระบบเครือข่ายหรือ Mininet ดังนั้นจึงเป็นช่วงเวลาที่เร็วที่สุดที่จะเป็นไปได้ จนถึงเวลาที่ 5 วินาที เนื่องจากเวลาที่ 5 วินาที เป็นเวลาที่การไหลของการรับส่งข้อมูลที่ 2 และ 3 จะเริ่มทำการส่งข้อมูลตามลำดับ หรือถ้ามากกว่า 5 วินาที ซึ่งเป็นช่วงเวลาที่การไหลของการรับส่งข้อมูลที่ 2 ได้เริ่มทำการส่งข้อมูลไปแล้ว จึงทำให้ไม่สามารถตรวจจับค่าปริมาณบัพเฟอร์ที่เกิดขึ้นก่อนหน้าการไหลของการรับส่งข้อมูลที่ 2 จะเริ่มทำการส่งข้อมูลได้

โดยที่ช่วงเวลาที่ 0.01 วินาทีจนถึงช่วงเวลา 1.58 วินาที ในเวลา 5 วินาที จนกระทั่งถึงเวลาที่ 180 วินาทีการเชื่อมต่อระหว่างของสองเครื่องจุดปลายจะสิ้นสุดการทำงานที่ละการไหลของการรับส่งข้อมูลตามลำดับ ซึ่งจะเห็นได้ว่าแนวโน้มของกราฟในช่วงเวลาดังกล่าวจะมีความคล้ายคลึงกับอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัพเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 ซึ่งใช้ช่วงเวลาที่ 0.01 วินาทีเช่นเดียวกัน จึงทำให้เส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path ของการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งคู่แรก (H1-H2) คู่สอง (H5-H6) และคู่สาม (H3-H4) เป็นเส้นทางเดียวกันได้คือ S1→S2→S8 ของคู่ส่งคู่ที่ 1, S1→S4→S8 ของคู่ส่งคู่ที่ 2 และ S4→S10→S11 ของคู่ส่งคู่ที่ 3 โดยผลรวมของปริมาณงาน (Overall Throughput) เฉลี่ยอยู่ที่ 2.3 Mbit/s ซึ่งช่วงเวลาที่ 0.01 วินาทีนั้นสามารถได้ผลรวมของปริมาณงาน (Overall Throughput) ได้มากกว่าที่สุด แต่เมื่อใช้ช่วงเวลาตั้งแต่ 1.59 วินาที จะเห็นผลรวมของปริมาณงาน (Overall Throughput) จะลดลงอย่างเห็นได้ชัด เนื่องจาก ตั้งแต่ช่วงเวลา 1.59 วินาทีเป็นต้นไป เส้นทางที่คำนวณไว้สำหรับ Forwarding Path ของการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งคู่แรก (H1-H2) และ คู่สอง (H5-H6) จะให้เลือกใช้เส้นทางเดียวกัน นั่นคือ S1→S2→S8 ซึ่งเป็นการไหลของการรับส่งข้อมูลประเภท TCP ที่เริ่มจะทำการส่งเป็นการไหลของการรับส่งข้อมูลแรก โดยในการติดตั้งเส้นทางมีลักษณะการทำงานคล้ายคลึงกับวิธีการเลือกเส้นทางแบบดั้งเดิม โดยเป็นผลลัพธ์ที่เกิดขึ้นมาจากที่ช่วงเวลาดังกล่าวตรวจพบว่า มีปริมาณค่าการครอบครองพื้นที่ในบัพเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัพเฟอร์ทั้งหมดบนเส้นทางที่ถูกเลือก หรือมากกว่าร้อยละ 70 ในกรณีนี้ แต่ไม่สามารถทำการลบการเชื่อมโยงของสวิตช์นั้น ออกจากการคำนวณในการหาเส้นทางที่มีระยะทางที่สั้นที่สุดในครั้งต่อไปได้ทันเวลาก่อนที่จะมีการไหลของการรับส่งข้อมูลใหม่เข้ามาในระบบเครือข่ายที่วินาทีที่ 10 เพราะหลังจากที่ตรวจพบว่ามีค่าความแออัดเกิดขึ้นในเส้นทางนั้น จะต้องมีช่วงเวลาของกระบวนการที่จะทำการส่งข้อมูลการเชื่อมโยงของช่องทางของแต่ละสวิตช์ที่มีปริมาณค่าการครอบครองพื้นที่ในบัพเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัพเฟอร์ทั้งหมด ไปยังองค์ประกอบ Topology Management Module (TMM) เพื่อที่จะทำการลบการเชื่อมโยงของสวิตช์นั้น ออกจากการคำนวณในการหาเส้นทางที่มีระยะทางที่สั้นที่สุดในครั้งต่อไป หรือทำการปรับปรุงเส้นทางที่สามารถทำการส่งข้อมูลได้ โดยที่ไม่เกิดความแออัดเกิดขึ้นบนทอพอโลยีที่นำไปใช้งาน



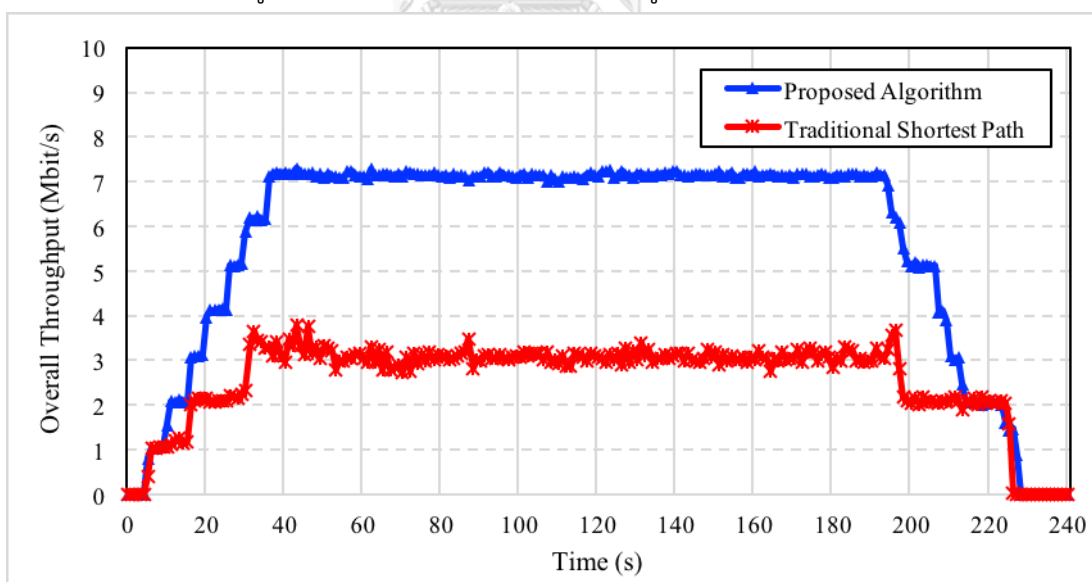
ภาพที่ 15 กราฟการเปรียบเทียบผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 ในแต่ละช่วงเวลา

จากผลทดลองการปรับหาช่วงเวลาที่เหมาะสมในการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ ในแง่ของผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่าย ช่วงเวลาที่เหมาะสมกับการนำไปใช้ในกระบวนการการวนซ้ำจะทำงานของคำนวณค่าปัจจุบันของการครอบครองพื้นที่ในบัฟเฟอร์เฉลี่ยจะอยู่ที่ช่วงเวลาตั้งแต่ 0.01 วินาทีถึง 1.58 วินาที เนื่องจากค่าปริมาณบัฟเฟอร์นั้นจะมีเปลี่ยนแปลงไปอย่างรวดเร็ว และจะต้องคำนึงถึงช่วงเวลาของกระบวนการของการลบการเชื่อมโยงของสวิตช์นั้น ออกจากการคำนวณในการหาเส้นทางที่มีระยะทางที่สั้นที่สุดในครั้งต่อไปด้วย ซึ่งกระบวนการในส่วนนี้จะต้องใช้เวลาอย่างน้อย 0.24 วินาที ดังนั้นเวลาที่จะนำช่วงเวลาดังกล่าวไปใช้จะต้องคำนึงถึงเวลาของกระบวนการนี้ด้วย ซึ่งในงานวิจัยนี้ได้เลือกช่วงเวลาที่จะนำมาใช้ที่ 0.01 วินาที เพราะว่าเป็นช่วงเวลาต่ำที่สุดที่เป็นไปได้ที่จะคำนวณและตรวจสอบค่าปัจจุบันของการครอบครองพื้นที่ในบัฟเฟอร์เฉลี่ยได้ รวมทั้งเวลาที่วินาทีนี้ได้ ผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่ายมากที่สุด แต่ในการนำไปใช้งานจริงไม่จำเป็นต้องใช้ 0.01 วินาทีเสมอไป เพราะจากผลการทดลองในภาพที่ 15 สามารถที่จะเลือกและเพิ่มลดช่วงเวลาดังกล่าวได้ตั้งแต่ 0.01 วินาทีถึง 1.58 วินาที ขึ้นอยู่กับประสิทธิภาพการทำงานของคอนโทรลเลอร์ที่จะนำไปใช้งานจริง ลักษณะของทอพอโลยีที่นำไปใช้งาน รวมไปถึงการนำไปใช้ในการ

ทดลองอื่นๆ อีกด้วย ซึ่งโดยทั่วไปช่วงเวลา 1 วินาทีนั้นเป็นช่วงเวลาทีหน่วยประมวลผลของคอนโทรลเลอร์ทั่วไปสามารถรองรับได้

4.4 ประสิทธิภาพการเพิ่มจำนวนการไหลของข้อมูลของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239

ในการวัดประสิทธิภาพการทำงานของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 จะใช้ผลรวมของปริมาณงาน (Overall Throughput) เป็นตัววัดประสิทธิภาพการทำงานของอัลกอริทึม โดยจะทำการทดลองเช่นเดียวกับการทดลองใน ข้อที่ 4.1 ประสิทธิภาพของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 รวมทั้งการตั้งค่าพารามิเตอร์ต่างๆ แต่จะแตกต่างกันที่ตำแหน่งของอุปกรณ์จุดปลายของแต่ละคู่ส่งและจำนวนการไหลของข้อมูลที่เพิ่มขึ้น ซึ่งในการทดลองนี้จะทำการสุ่มอุปกรณ์จุดปลายของแต่ละคู่ส่งบนทอพอโลยี COST239 และทำการเพิ่มจำนวนการไหลของข้อมูลเข้าไปในระบบเครือข่ายเรื่อยๆ จนถึงจำนวนการไหลของข้อมูลสูงสุดที่อัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้สามารถทำงานได้บนทอพอโลยี COST239 โดยทีแต่ละคู่ส่งที่เพิ่มเข้าไปในระบบเครือข่ายจะเริ่มการส่งคู่ส่งแรกที่ 5 วินาที และต่อมาแต่ละคู่ส่งจะเริ่มห่างกันที่ 5 วินาที



ภาพที่ 16 กราฟการเปรียบเทียบผลรวมของปริมาณงาน (Overall Throughput) การเพิ่มจำนวนการไหลของข้อมูลของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239

ผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ที่แสดงดังภาพที่ 16 จะแสดงที่แกน Y ของกราฟในหน่วยเมกะบิตต่อวินาที (Mbit/s) และแกน X จะแสดงเวลาที่ใช้ในการทำการทดลองในหน่วยวินาที โดยเส้นแดงจะเป็นผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ส่วนเส้นสีฟ้าผลรวมของปริมาณงาน (Overall Throughput) ของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้

ในกรณีแนวทางของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm

ที่เวลา 5 วินาที เมื่อเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งคู่แรกระหว่าง H1 และ H2 ซึ่งสู่มตำแหน่งของ H1 ได้ที่ S1 และตำแหน่งของ H2 ที่ S8 จากนั้นคอนโทรลเลอร์จะทำการคำนวณหาเส้นทางที่สั้นที่สุดหรือ Shortest Path และทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S2 \rightarrow S8$ ซึ่งเป็นการไหลของการรับส่งข้อมูลประเภท TCP ที่เริ่มจะทำการส่งเป็นการไหลของการรับส่งข้อมูลแรก

ต่อมาที่วินาทีที่ 10 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 ซึ่งสู่มตำแหน่งของ H5 ได้ที่ S1 และตำแหน่งของ H6 ที่ S8 โดยได้ตำแหน่งเดียวกันกับคู่ส่งที่ 1 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S2 \rightarrow S8$ ซึ่งเห็นได้ชัดเจนว่า คู่ส่งที่ 2 จะใช้เส้นทางที่ใช้ในการส่งข้อมูลเส้นทางเดียวกันกับคู่ส่งแรกเพราะว่าวิธีการแบบดั้งเดิมไม่ได้พิจารณาเกี่ยวกับปริมาณบัฟเฟอร์ที่เกิดขึ้นบนอุปกรณ์ระบบเครือข่าย และนอกจากนั้นวิธีการแบบดั้งเดิมจะเลือกเส้นทางที่ดีที่สุดจากระยะทางที่สั้นที่สุดเสมอ ซึ่งส่งผลให้ประสิทธิภาพของผลรวมของปริมาณงาน (Overall Throughput) ไม่เพิ่มขึ้นเนื่องจากคู่ส่งที่ 1 และคู่ส่งที่ 2 ได้ใช้เส้นทางที่ใช้ในการส่งร่วมกัน

ต่อมาที่วินาทีที่ 15 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 3 ระหว่าง H3 และ H4 ซึ่งสู่มตำแหน่งของ H3 ได้ที่ S4 และตำแหน่งของ H4 ที่ S11 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S4 \rightarrow S10 \rightarrow S11$

ต่อมาที่วินาทีที่ 20 เริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 4 ระหว่าง H7 และ H8 ซึ่งสุมตำแหน่งของ H7 ได้ที่ S4 และตำแหน่งของ H8 ที่ S11 โดยได้ตำแหน่งเดียวกันกับคู่ส่งที่ 3 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S4 \rightarrow S10 \rightarrow S11$ ซึ่งคู่ส่งที่ 4 จะใช้เส้นทางที่ใช้ในการส่งข้อมูลเส้นทางเดียวกันกับคู่ส่งที่ 3 ดังนั้นจะส่งผลให้ประสิทธิภาพของผลรวมของปริมาณงาน (Overall Throughput) ไม่เพิ่มขึ้นเนื่องจากคู่ส่งที่ 3 และคู่ส่งที่ 4 ได้ใช้เส้นทางที่ใช้ในการส่งร่วมกัน

ต่อมาที่วินาทีที่ 25 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 5 ระหว่าง H9 และ H10 ซึ่งสุมตำแหน่งของ H9 ได้ที่ S2 และตำแหน่งของ H10 ที่ S10 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S2 \rightarrow S1 \rightarrow S4 \rightarrow S10$ สำหรับเส้นทางนี้เป็นเส้นทางที่คู่ส่งที่ 5 จะใช้ในการส่งข้อมูล ที่ซึ่งไปซับซ้อนกับคู่ส่งที่ 1, คู่ส่งที่ 2, คู่ส่งที่ 3 และคู่ส่งที่ 4 เพราะว่าเป็นผลของการที่วิธีการแบบดั้งเดิมนั้นไม่ได้พิจารณาเกี่ยวกับปริมาณบัฟเฟอร์ที่เกิดขึ้นบนอุปกรณ์ระบบเครือข่าย และจะทำการเลือกเส้นทางที่ดีที่สุดจากระยะทางที่สั้นที่สุดเสมอ ทำให้ส่งผลต่อประสิทธิภาพของผลรวมของปริมาณงาน (Overall Throughput) ไม่เพิ่มขึ้นเนื่องจากเกิดความแออัดของการส่งข้อมูลที่บริเวณ $S2 \rightarrow S1 \rightarrow S4 \rightarrow S10$

ต่อมาที่วินาทีที่ 30 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 6 ระหว่าง H11 และ H12 ซึ่งสุมตำแหน่งของ H11 ได้ที่ S3 และตำแหน่งของ H12 ที่ S9 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S3 \rightarrow S16 \rightarrow S9$

ต่อมาที่วินาทีที่ 35 เริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 7 ระหว่าง H13 และ H14 ซึ่งสุมตำแหน่งของ H13 ได้ที่ S2 และตำแหน่งของ H14 ที่ S10 โดยได้ตำแหน่งเดียวกันกับคู่ส่งที่ 5 จากนั้นเส้นทางที่ได้คำนวณได้ก็จะใช้เส้นทางเดียวกันกับส่งที่ 5 คือ $S2 \rightarrow S1 \rightarrow S4 \rightarrow S10$ ซึ่งเป็นเส้นทางที่ไปซับซ้อนกับคู่ส่งส่วนมาก ไม่ว่าจะเป็นคู่ส่งที่ 1, คู่ส่งที่ 2, คู่ส่งที่ 3, คู่ส่งที่ 4 และคู่ส่งที่ 5 จึงทำให้เส้นทาง $S2 \rightarrow S1$ ที่วินาทีที่ 35 มี 4 คู่ส่งที่ใช้เส้นทางซับซ้อนกัน คือ คู่ส่งที่ 1, คู่ส่งที่ 2, คู่ส่งที่ 5 และคู่ส่งที่ 7 และเส้นทาง $S1 \rightarrow S4 \rightarrow S10$ ที่วินาทีที่ 35 มี 4 คู่ส่งที่ใช้เส้นทางซับซ้อนกันคือคู่ส่งที่ 3, คู่ส่งที่ 4, คู่ส่งที่ 5 และคู่ส่งที่ 7

ต่อมาที่วินาทีที่ 40 เริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 8 ระหว่าง H15 และ H16 ซึ่งสุมตำแหน่งของ H15 ได้ที่ S3 และตำแหน่งของ H15 ที่ S19 โดยทำให้ได้เส้นทางเดียวกันกับคู่ส่งที่ 6 ซึ่งทำให้เส้นทางนี้เกิดความแออัดเช่นเดียวกัน โดยที่คู่ส่งที่ 7 และคู่ส่งที่ 8 นั้นไปใช้เส้นทางที่เกิดความแออัด หรือไปใช้เส้นทางที่ซ้อนกับคู่ส่งคู่อื่นๆ ซึ่งเป็นผลมาจากการเลือกเส้นทางที่ดีที่สุดจากระยะทางที่สั้นที่สุดเสมอของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm จึงทำให้ส่งผลต่อประสิทธิภาพของผลรวมของปริมาณงาน (Overall Throughput) ตั้งแต่คู่ส่งที่ 7 เป็นต้นไปนั้นไม่เพิ่มขึ้นและคงที่ตลอดจนการทำงานเสร็จสิ้น โดยค่าเฉลี่ยปริมาณงาน (Throughput) ของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งของแต่ละการไหลของการรับส่งข้อมูลประเภท TCP จะแสดงในตารางที่ 5

ตารางที่ 5 ค่าเฉลี่ยปริมาณงานของอุปกรณ์จุดปลายหนึ่ง ไปยังอุปกรณ์จุดปลายอีกจุดหนึ่งของการเพิ่มจำนวนการไหลของข้อมูลบนทอพอโลยี COST239

Flow	อัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิม (Mbit/s)	อัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ (Mbit/s)
H1-H2	0.44	0.81
H5-H6	0.39	0.80
H3-H4	0.39	0.81
H7-H8	0.24	0.81
H9-H10	0.12	0.80
H11-H12	0.46	0.43
H13-H14	0.14	0.78
H15-H16	0.37	0.39

ในกรณีแนวทางของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้

ที่เวลา 5 วินาที เมื่อเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งคู่แรกระหว่าง H1 และ H2 ซึ่งสุมตำแหน่งของ H1 ได้ที่ S1 และตำแหน่งของ H2 ที่ S8 จากนั้น POX คอนโทรลเลอร์จะทำการคำนวณหาเส้นทางที่สั้นที่สุดหรือ Shortest Path และทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S2 \rightarrow S8$ ซึ่งเป็นการไหลของการรับส่งข้อมูลประเภท TCP ที่เริ่มจะทำการส่งเป็นการไหลของการรับส่งข้อมูลแรก โดยในการติดตั้งเส้นทางมีลักษณะการทำงานคล้ายคลึงกับวิธีการเลือกเส้นทางแบบดั้งเดิม เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์น้อยกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมดบนเส้นทางที่ถูกเลือก หรือน้อยกว่าร้อยละ 70 ในกรณีนี้

ต่อมาที่วินาทีที่ 10 จะเริ่มทำการเชื่อมต่อด้วยการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 ซึ่งสุมตำแหน่งของ H5 ได้ที่ S1 และตำแหน่งของ H6 ที่ S8 โดยได้ตำแหน่งเดียวกันกับคู่ส่งที่ 1 จากนั้นคอนโทรลเลอร์จะทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้สำหรับ Forwarding Path คือ $S1 \rightarrow S2 \rightarrow S8$ ซึ่งเห็นได้ชัดเจนว่ามีเส้นทางที่พร้อมใช้งานอยู่ 2 เส้นทางที่สอดคล้องกับระยะทางที่สั้นที่สุด คือ $S1 \rightarrow S2 \rightarrow S8$ และ $S1 \rightarrow S4 \rightarrow S8$ แต่อย่างไรก็ตามในเส้นทางแรก คือ $S1 \rightarrow S2 \rightarrow S8$ จะไม่ถูกเลือกใช้ในการส่งข้อมูล เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ดังนั้นเส้นทางที่สองที่จะใช้ในการส่งการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 2 ระหว่าง H5 และ H6 จะเลือกเส้นทาง $S1 \rightarrow S4 \rightarrow S8$ ซึ่งจะถูกติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้

จากนั้นวินาทีที่ 15 จะเริ่มทำการเชื่อมต่อด้วย การไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 3 ระหว่าง H3 และ H4 ซึ่งสุมตำแหน่งของ H3 ได้ที่ S4 และตำแหน่งของ H4 ที่ S11 โดยจะทำการเลือกเส้นทาง $S4 \rightarrow S10 \rightarrow S11$ แทน $S4 \rightarrow S8 \rightarrow S11$ เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมดบนเส้นทางที่ถูกเลือก

ต่อมาวินาทีที่ 20 จะเริ่มทำการเชื่อมต่อด้วย การไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 4 ระหว่าง H7 และ H8 ซึ่งสุมตำแหน่งของ H7 ได้ที่ S4 และตำแหน่งของ H8 ที่ S11 โดยได้ตำแหน่งเดียวกันกับคู่ส่งที่ 3 ซึ่งมีเส้นทางที่พร้อมใช้งานอยู่ 4 เส้นทางที่สอดคล้องกับระยะทางที่สั้นที่สุด คือ $S4 \rightarrow S8 \rightarrow S11$, $S4 \rightarrow S10 \rightarrow S11$, $S4 \rightarrow S3 \rightarrow S5 \rightarrow S11$ และ $S4 \rightarrow S7 \rightarrow S9 \rightarrow S11$ แต่อย่างไรก็ตามในเส้นทาง $S4 \rightarrow S8 \rightarrow S11$ และ $S4 \rightarrow S10 \rightarrow S11$ จะไม่ถูกเลือกใช้ในการส่งข้อมูล เนื่องจาก

ปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ดังนั้นเส้นทางที่จะใช้ในการส่งการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 4 ระหว่าง H7 และ H8 จะเลือกเส้นทาง $S4 \rightarrow S3 \rightarrow S5 \rightarrow S11$ เพราะเป็นเส้นทางที่ระยะทางสั้นสุดลำดับรองถัดมา ซึ่งจะถูกติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้

ต่อมาวินาทีที่ 25 จะเริ่มทำการเชื่อมต่อด้วย การไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 5 ระหว่าง H9 และ H10 ซึ่งสุมตำแหน่งของ H9 ได้ที่ S2 และตำแหน่งของ H10 ที่ S14 ซึ่งมีเส้นทางที่พร้อมใช้งานที่สอดคล้องกับระยะทางที่สั้นที่สุด คือ $S2 \rightarrow S5 \rightarrow S4 \rightarrow S10$, $S2 \rightarrow S8 \rightarrow S11 \rightarrow S10$ และ $S2 \rightarrow S3 \rightarrow S6 \rightarrow S7 \rightarrow S10$ แต่อย่างไรก็ตามในเส้นทาง $S2 \rightarrow S5 \rightarrow S4 \rightarrow S10$ และ $S2 \rightarrow S8 \rightarrow S11 \rightarrow S10$ จะไม่ถูกเลือกใช้ในการส่งข้อมูล เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ดังนั้นเส้นทางที่จะใช้ในการส่งการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 5 ระหว่าง H9 และ H10 จะเลือกเส้นทาง $S2 \rightarrow S3 \rightarrow S6 \rightarrow S7 \rightarrow S10$ เพราะเป็นเส้นทางที่ระยะทางสั้นสุดลำดับรองถัดมา ซึ่งจะถูกติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้

ต่อมาวินาทีที่ 30 จะเริ่มทำการเชื่อมต่อด้วย การไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 6 ระหว่าง H11 และ H12 ซึ่งสุมตำแหน่งของ H11 ได้ที่ S3 และตำแหน่งของ H12 ที่ S9 ซึ่งมีเส้นทางที่พร้อมใช้งานอยู่ 4 เส้นทางที่สอดคล้องกับระยะทางที่สั้นที่สุด คือ $S3 \rightarrow S6 \rightarrow S9$, $S3 \rightarrow S1 \rightarrow S7 \rightarrow S9$, $S3 \rightarrow S5 \rightarrow S11 \rightarrow S9$ และ $S3 \rightarrow S4 \rightarrow S7 \rightarrow S9$ แต่อย่างไรก็ตามในเส้นทาง $S3 \rightarrow S6 \rightarrow S9$, $S3 \rightarrow S5 \rightarrow S11 \rightarrow S9$ และ $S3 \rightarrow S4 \rightarrow S7 \rightarrow S9$ จะไม่ถูกเลือกใช้ในการส่งข้อมูล เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ดังนั้นเส้นทางที่จะใช้ในการส่งการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 6 ระหว่าง H11 และ H12 จะเลือกเส้นทาง $S3 \rightarrow S1 \rightarrow S7 \rightarrow S9$ เพราะเป็นเส้นทางที่ระยะทางสั้นสุดลำดับรองถัดมา ซึ่งจะถูกติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้

ต่อมาวินาทีที่ 35 จะเริ่มทำการเชื่อมต่อด้วย การไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 7 ระหว่าง H13 และ H14 ซึ่งสุมตำแหน่งของ H13 ได้ที่ S2 และตำแหน่งของ H14 ที่ S14 โดยได้ตำแหน่งเดียวกันกับคู่ส่งที่ 5 ซึ่งมีเส้นทางที่พร้อมใช้งานที่สอดคล้องกับระยะทางที่สั้นที่สุด คือ $S2 \rightarrow S3 \rightarrow S1 \rightarrow S7 \rightarrow S9 \rightarrow S10$, $S2 \rightarrow S3 \rightarrow S1 \rightarrow S7 \rightarrow S4 \rightarrow S10$, $S2 \rightarrow S5 \rightarrow S8 \rightarrow S9 \rightarrow S10$, $S2 \rightarrow S5 \rightarrow S6 \rightarrow S9 \rightarrow S10$ และ $S2 \rightarrow S5 \rightarrow S8 \rightarrow S11 \rightarrow S10$ แต่อย่างไรก็ตามในเส้นทาง $S2 \rightarrow S3 \rightarrow S1 \rightarrow S7 \rightarrow S9 \rightarrow S10$, $S2 \rightarrow S3 \rightarrow S1 \rightarrow S7 \rightarrow S4 \rightarrow S10$, และ $S2 \rightarrow S5 \rightarrow S8 \rightarrow S11 \rightarrow S10$ จะไม่ถูกเลือกใช้ในการส่งข้อมูล เนื่องจากปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ดังนั้นเส้นทางที่จะใช้ในการส่งการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 7

ระหว่าง H13 และ H14 จะเลือกเส้นทาง $S2 \rightarrow S5 \rightarrow S6 \rightarrow S9 \rightarrow S10$ เพราะเป็นเส้นทางที่ระยะทางสั้นสุดลำดับรองถัดมา ซึ่งจะถูกติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้

ต่อมาวินาทีที่ 40 จะเริ่มทำการเชื่อมต่อด้วย การไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 8 ระหว่าง H15 และ H16 ซึ่งสุ่มตำแหน่งของ H15 ได้ที่ S3 และตำแหน่งของ H16 ที่ S9 โดยได้ตำแหน่งเดียวกันกับคู่ส่งที่ 6 ซึ่งที่วินาทีนี้จะไม่มีความพร้อมใช้งาน เพราะเส้นทางทั้งหมดนั้นได้มีปริมาณค่าการครอบครองพื้นที่ในบัฟเฟอร์มากกว่าเกณฑ์ที่ตั้งไว้ของขนาดบัฟเฟอร์ทั้งหมด ดังนั้นเส้นทางที่จะใช้ในการส่งการไหลของการรับส่งข้อมูลประเภท TCP ของคู่ส่งที่ 8 ระหว่าง H15 และ H16 จะกลับไปเลือกเส้นทางที่สอดคล้องกับระยะทางที่สั้นที่สุดอันดับแรก ตามการเลือกเส้นทางของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ก็คือ $S3 \rightarrow S6 \rightarrow S9$ และทำการติดตั้ง Forwarding Rules ทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้ โดยที่เส้นทางดังกล่าวนี้ได้ทับซ้อนกับคู่ส่งที่ 6 ทำให้ปริมาณงาน (Throughput) ของคู่ส่งที่ 6 ถูกปรับลดลงตั้งแต่วินาทีที่ 40 เป็นต้นไป และส่งผลให้ประสิทธิภาพของผลรวมของปริมาณงาน (Overall Throughput) ไม่เพิ่มขึ้นและคงที่ที่คู่ส่งนี้

จากผลทดลองเมื่อเพิ่มจำนวนการไหลของข้อมูลเข้าระบบเรื่อยๆ จำนวนการไหลของข้อมูลที่อัลกอริทึมในการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ จะที่สามารถทำงานได้อย่างเต็มประสิทธิภาพบนทอพอโลยี COST239 นั้นจะอยู่ที่ 7 คู่ส่ง เพราะเมื่อมีการเพิ่มคู่ส่งที่ 8 เข้าไปในระบบ อัลกอริทึมที่เสนอในงานวิจัยนี้จะไม่สามารถค้นหาเส้นทางใหม่ให้คู่ส่งคู่นี้ได้ เนื่องจากเซตของการเชื่อมโยงที่ใช้การคำนวณหาเส้นทางของอัลกอริทึมที่เสนอในงานวิจัยนั้นบนคอนโทรลเลอร์ ได้ถูกลบออกหมดแล้ว จึงได้ให้อัลกอริทึมที่เสนอในงานวิจัยนี้ จะกลับไปเลือกเส้นทางที่สั้นที่สุดแบบดั้งเดิมแทน จึงทำให้คู่ส่งที่ 8 กลับไปใช้เส้นทางเดิมที่อัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm คำนวณได้ ส่งผลให้ผลรวมของปริมาณงาน (Overall Throughput) ในคู่ส่งคู่นี้ไม่เพิ่มขึ้น แต่อย่างไรก็ตามอัลกอริทึมในการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ ยังคงมีประสิทธิภาพกว่าอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ในแง่ของผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่ายถึงร้อยละ 120 ซึ่งผลของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm จะยังคงแย่กว่าอัลกอริทึมในการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ เนื่องจากในกรณีนี้จะเกิดปริมาณการใช้ที่มีความแออัดบนเส้นทาง $S1 \rightarrow S2 \rightarrow S8$, $S8 \rightarrow S10 \rightarrow S11$ และ $S3 \rightarrow S6 \rightarrow S9$ ตลอดการทำการทดลอง หลังจากนั้นเมื่อเวลาในการส่งเพิ่มขึ้นเรื่อยๆ จนกระทั่งถึงเวลาที่ 180 วินาทีการเชื่อมต่อระหว่างของสองเครื่องจุดปลายจะสิ้นสุดการทำงานที่ละการไหลของการรับส่งข้อมูลตามลำดับ

4.5 ค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane)

สำหรับในการทดลองนี้จะทำการทดลองเพื่อวัดหาค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane) ซึ่งเป็นส่วนที่ใช้ติดต่อระหว่างส่วนของคอนโทรลเลอร์ (Controller) กับส่วนของสวิตช์ (Switch) โดยในการทดลองนี้จะทำการทดลองบนทอพอโลยี COST239 และ ค่าใช้จ่ายที่เพิ่มขึ้น (Overhead) บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ หรือ Software Defined Networks (SDN) นั่นก็คือจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้น ซึ่งข้อความของเกณฑ์วิธี OpenFlow ทั้งหมดที่เกิดขึ้นนั้น โดยปกติแล้วจะประกอบไปด้วยข้อความชนิดต่างๆ ดังนี้

1. HELLO

เป็นข้อความแรกที่คอนโทรลเลอร์ใช้การทักทายกับสวิตช์ เมื่อการเชื่อมต่อของคอนโทรลเลอร์กับสวิตช์สำเร็จ

2. PORT_STATUS

เป็นข้อความประเภทเหตุการณ์แบบไม่ประสานเวลา (Asynchronous) ที่ส่งจากสวิตช์ไปยังคอนโทรลเลอร์เพื่อระบุสถานะของช่องทางที่ระบุ

3. FEATURES_REQUEST และ FEATURES_REPLY

เป็นข้อความที่จะเกิดขึ้นเมื่อมีการสร้างช่องทางของการขนส่ง (Transport Channel) ระหว่างสวิตช์และคอนโทรลเลอร์ เช่น TCP, SCTP และ TLS โดยคอนโทรลเลอร์จะส่งข้อความ FEATURES_REQUEST เพื่อร้องขอข้อมูลความสามารถและการดำเนินการต่างๆ ของสวิตช์ จากนั้นสวิตช์จะส่งข้อมูลที่คอนโทรลเลอร์ร้องขอกลับไปยังคอนโทรลเลอร์ โดยจะส่งข้อความ FEATURES_REPLY

4. SET_CONFIG

เป็นข้อความที่คอนโทรลเลอร์ส่งไปหาสวิตช์เพื่อทำการเปลี่ยนแปลงคุณสมบัติของกลุ่มข้อมูลที่กระจายตัวอยู่ในการประมวลผลแบบ Pipeline เช่น การ Drop หรือ การ Reassemble เป็นต้น

5. PACKET_IN

เป็นข้อความที่สวิตช์ส่งไปหาคอนโทรลเลอร์เพื่อแจ้งให้คอนโทรลเลอร์ทราบว่ามีการกลุ่มข้อมูลใหม่เข้ามาในสวิตช์

6. FLOW_MOD

เป็นข้อความที่คอนโทรลเลอร์ส่งไปหาสวิตช์เพื่อการติดตั้ง Forwarding Rules กับทุกสวิตช์ที่อยู่ในเส้นทางที่คอนโทรลเลอร์ได้คำนวณไว้

7. PACKET_OUT

เป็นข้อความที่คอนโทรลเลอร์ส่งไปหาสวิตช์เพื่อทำการเปิดทางให้ Forwarding Rules ที่ได้ถูกติดตั้งไว้บนสวิตช์ทำงาน

8. STATS_REQUEST และ STATS_REPLY

เป็นข้อความที่คอนโทรลเลอร์ส่งไปหาสวิตช์เพื่อร้องขอข้อมูลต่างๆ เชิงสถิติในแต่ละการไหล โดยคอนโทรลเลอร์จะส่งข้อความ STATS_REQUEST ไปยังสวิตช์และจากนั้นสวิตช์จะส่งข้อมูลที่คอนโทรลเลอร์ร้องขอกลับไป ด้วยการส่งข้อความ STATS_REPLY

9. BARRIER_REQUEST และ BARRIER_REPLY

เป็นข้อความที่คอนโทรลเลอร์ส่งไปหาสวิตช์เพื่อทำการตรวจสอบเพื่อให้มั่นใจว่าสถานะของข้อความก่อนหน้าได้เสร็จสมบูรณ์แล้ว ก่อนที่ส่งข้อความ BARRIER_REPLY กลับไปยังคอนโทรลเลอร์

10. ECHO_REQUEST และ ECHO_REPLY

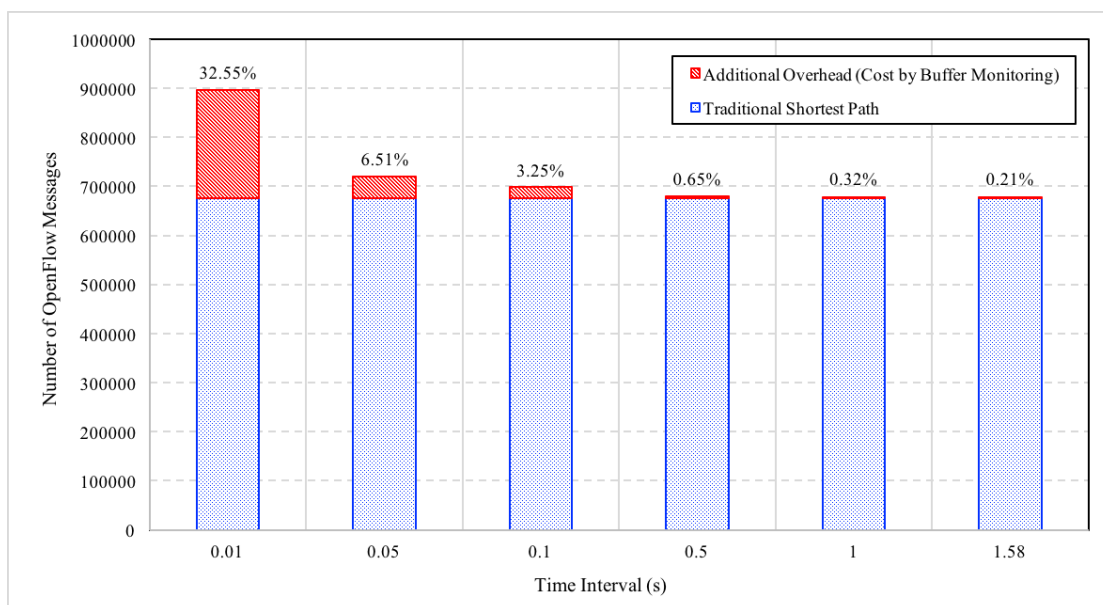
เป็นข้อความที่ใช้แลกเปลี่ยนข้อมูลต่างๆ เช่น เวลาแฝง ค่าความกว้างแถบความถี่ สถานะการเชื่อมต่อกับสวิตช์ เป็นต้น ระหว่างคอนโทรลเลอร์กับสวิตช์ โดยเมื่อมีการร้องขอข้อมูลจากคอนโทรลเลอร์ คอนโทรลเลอร์จะส่งข้อความ ECHO_REQUEST ไปยังสวิตช์ จากนั้นถ้าสถานะการเชื่อมต่อกับสวิตช์ปกติ สวิตช์จะส่งข้อมูลที่คอนโทรลเลอร์ร้องขอกลับไป ด้วยการส่งข้อความ ECHO_REPLY ถ้าคอนโทรลเลอร์ไม่ได้รับ ECHO_REPLY หรือการร้องขอหมดเวลารอ (Timeout) จะแสดงว่าคอนโทรลเลอร์ได้ขาดการเชื่อมต่อกับสวิตช์นั้น

ในภาพที่ 17 จำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้นของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 จะแสดงที่แกน Y ของกราฟ และแกน X จะแสดงช่วงเวลาในการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละ

สวิตช์โดยคอนโทรลเลอร์ และตัวเลขที่ปรากฏบนกราฟแต่ละแท่งจะแสดงปริมาณข้อความของเกณฑ์วิธี OpenFlow ที่เพิ่มขึ้นจากอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ในรูปแบบของเปอร์เซ็นต์ โดยจะทำการทดลองเช่นเดียวกับการทดลองในข้อที่ 4.1 ประสิทธิภาพของอัลกอริทึมที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 ไม่ว่าจะเป็นเวลาที่ใช้ในการทดลอง จำนวนของคูปองที่ใช้ รวมทั้งการตั้งค่าพารามิเตอร์ต่างๆ แต่จะแตกต่างกันที่ การทดลองนี้จะทำการวัดเฉพาะปริมาณข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้นทั้งหมด ในแต่ละช่วงเวลาตั้งแต่ 0.01 วินาที จนถึง 1.58 วินาที ตามช่วงเวลาที่ได้จากการทดลองข้อที่ 4.3 การปรับหาช่วงเวลาที่เหมาะสมในการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์

โดยที่ช่วงเวลา 0.01 วินาทีจนถึงช่วงเวลา 1.58 วินาที จำนวนข้อความของเกณฑ์วิธี OpenFlow ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ที่เกิดขึ้นทั้งหมดจะอยู่ที่ 675,923 ข้อความ ซึ่งจะรวมข้อความตั้งแต่ข้อความแรกคือข้อความ HELLO จนถึงกระทั่งข้อความที่ติดตั้ง Forwarding Rule ของการไหลของการรับส่งข้อมูลประเภท TCP ของคูปองคู่ที่สาม (H3-H4) เสร็จเรียบร้อย โดยจะเห็นว่าจำนวนข้อความของเกณฑ์วิธี OpenFlow ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm จะคงที่ตลอดทุกช่วงเวลา เนื่องจากอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ไม่มีกลไกการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์ แต่ส่วนของอัลกอริทึมในการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ จะมีกลไกการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์ เข้ามาใช้ในการพิจารณาหาเส้นทาง โดยมีช่วงเวลาที่ใช้ที่แตกต่างกันไป ขึ้นอยู่กับประสิทธิภาพการทำงานของคอนโทรลเลอร์ที่จะนำไปใช้งานจริง ในช่วงเวลา 0.01 วินาที จำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เพิ่มมากขึ้นจะอยู่ที่ 220,000 ข้อความหรือเป็นจำนวนร้อยละ 32.55 ของจำนวนข้อความของเกณฑ์วิธี OpenFlow ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ต่อมาในช่วงเวลาที่ 0.05 วินาทีจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เพิ่มขึ้นมาจะอยู่ที่ 44,000 ข้อความหรือเป็นจำนวนร้อยละ 6.51 ต่อมาในช่วงเวลาที่ 0.1 วินาทีจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เพิ่มขึ้นมาจะอยู่ที่ 22,000 ข้อความหรือเป็นจำนวนร้อยละ 3.25 ต่อมาในช่วงเวลาที่ 0.1 วินาทีจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เพิ่มขึ้นมาจะอยู่ที่ 4,400 ข้อความ หรือเป็นจำนวนร้อยละ 0.65 ต่อมาในช่วงเวลาที่ 0.5 วินาทีจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เพิ่มขึ้นมาจะอยู่ที่ 4,400 ข้อความ หรือเป็นจำนวนร้อยละ 0.65 ต่อมาในช่วงเวลาที่ 1 วินาทีจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เพิ่มขึ้นมาจะอยู่ที่ 2,200 ข้อความ หรือเป็นจำนวนร้อยละ 0.32 และในช่วงเวลาที่ 1.58 วินาที ซึ่งเป็น

ช่วงเวลาที่เป็นไปได้ ที่จะส่งผลให้ผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่าย มีปริมาณสูงกว่าอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ได้ ซึ่งจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เพิ่มขึ้นมาจะอยู่ที่ 2,200 ข้อความ หรือเป็นจำนวนร้อยละ 0.32



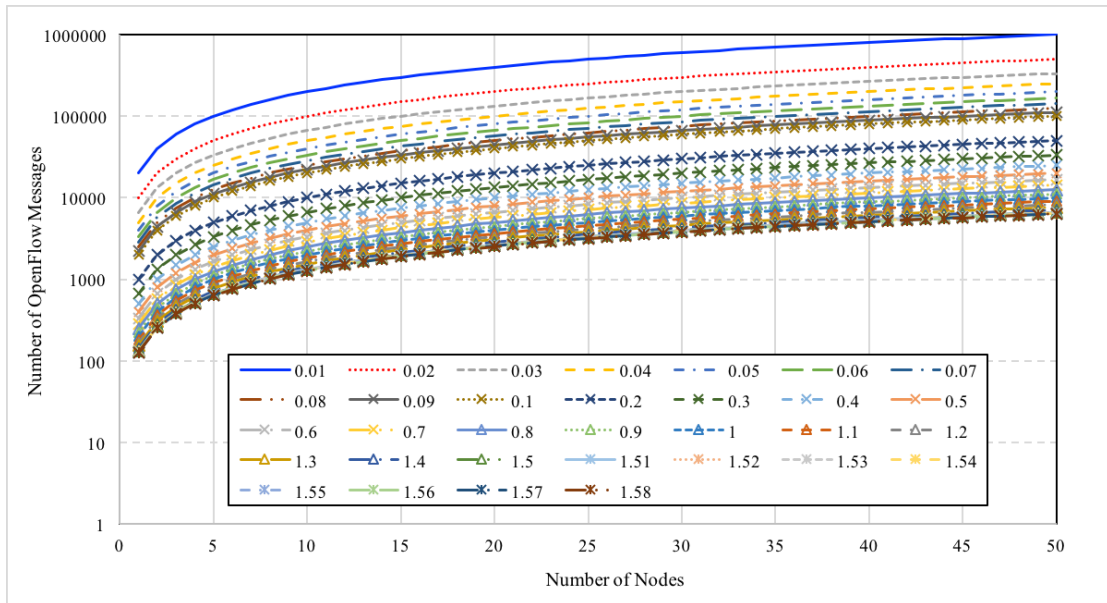
ภาพที่ 17 กราฟการเปรียบเทียบจำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้นของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm และอัลกอริทึมในการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 ในแต่ละช่วงเวลา

จากผลทดลองหาค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane) เมื่อมีการเพิ่มกลไกการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์ในแต่ละช่วงเวลา ลงไปในอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ ซึ่งจะทำให้มีค่าใช้จ่ายที่เพิ่มขึ้น (Overhead) จากอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm โดยปริมาณค่าใช้จ่ายที่เพิ่มขึ้น (Overhead) นั้นจะขึ้นอยู่กับช่วงเวลาในการเรียกค่าปริมาณบัฟเฟอร์ที่นำไปใช้งาน ซึ่งช่วงเวลาที่เหมาะสมกับการนำไปใช้ในกระบวนการการวนซ้ำทำงานของการคำนวณค่าปัจจุบันของการครอบครองพื้นที่ในบัฟเฟอร์เฉลี่ยจะอยู่ที่ช่วงเวลาตั้งแต่ 0.01 วินาทีถึง 1.58 วินาที จากการทดลองที่ 4.3 โดยที่ช่วงเวลาที่ 0.01 วินาทีจะมีปริมาณค่าใช้จ่ายที่เพิ่มขึ้น (Overhead) มากกว่าอัลกอริทึมใน

การหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm ถึงร้อยละ 32.55 จากนั้นเมื่อช่วงเวลาในการเรียกค่าปริมาณบัฟเฟอร์ เพิ่มขึ้นเรื่อยๆ จนกระทั่งถึงวินาทีที่ 1.58 วินาที จากกราฟจะเห็นได้ว่าจากช่วงเวลา 0.01 วินาทีถึง 1.58 วินาทีนั้นมีปริมาณค่าใช้จ่ายที่เพิ่มขึ้น (Overhead) ที่ลดลงตามลำดับ ซึ่งมีอัตราการเปลี่ยนแปลงตามลักษณะของฟังก์ชันเลขชี้กำลัง หรือ Exponential Function ที่เข้าสู่จำนวนข้อความของเกณฑ์วิธี OpenFlow ของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm

โดยในการทดลองนี้ นอกจากที่หาค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane) ในแต่ละช่วงเวลาของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ เปรียบเทียบกับอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm แล้วยังได้ทำการทดลองเพื่อหาค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane) ในแต่ละจำนวนสวิตช์อีกด้วย โดยจะแสดงดังรูปภาพที่ 18 จำนวนข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้นของอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239 จะแสดงที่แกน Y ของกราฟ และแกน X จะแสดงจำนวนของจำนวนสวิตช์ที่เพิ่มขึ้น โดยในการทดลองนี้จะทำการวัดเฉพาะปริมาณข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้นในกลไกการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์ในแต่ละช่วงเวลา โดยจะเริ่มตั้งแต่ช่วงเวลาที่ 0.01 วินาที จนถึง 1.58 วินาที ตามช่วงเวลาที่ได้จากการทดลองข้อที่ 4.3 การปรับหาช่วงเวลาที่เหมาะสมในการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์โดยคอนโทรลเลอร์ และจำนวนสวิตช์ที่ใช้วัดจะเริ่มตั้งแต่สวิตช์จำนวน 1 โหนดจนถึงจำนวนสวิตช์ 50 โหนด

จากผลการทดลองหาค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane) ในแต่ละจำนวนสวิตช์ และช่วงเวลา เมื่อเริ่มที่จำนวนสวิตช์ 1 โหนด ซึ่งจะเห็นว่าช่วงเวลาที่ 0.01 วินาทีเป็นช่วงเวลาที่ จะทำการให้เกิดปริมาณข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้นในกลไกการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์ มากที่สุดเมื่อเปรียบเทียบกับช่วงเวลาอื่นๆ ในจำนวนสวิตช์ที่เท่ากัน และเมื่อจำนวนของสวิตช์มีปริมาณที่เพิ่มขึ้นเรื่อยๆ จนกระทั่งถึงจำนวนสวิตช์ที่ 50 โหนด ซึ่งปริมาณข้อความของเกณฑ์วิธี OpenFlow ที่เกิดขึ้นในกลไกนี้จะเพิ่มขึ้นตามจำนวนของสวิตช์ที่เพิ่มขึ้น และช่วงเวลาที่ใช้ในกระบวนการการวนซ้ำทำงานของการคำนวณค่าปัจจุบันของการครอบครองพื้นที่ในบัฟเฟอร์เฉลี่ย โดยถ้าช่วงเวลาที่ใช้มีความถี่ในเรียกค่าดังกล่าวมากเท่าไร จะส่งผลการเพิ่มขึ้นของปริมาณข้อความของเกณฑ์วิธี OpenFlow ที่เกิดมากขึ้น หรือ การใช้ช่วงเวลาในการเรียกค่าการครอบครองพื้นที่ในบัฟเฟอร์ที่มีความถี่มากขึ้นจะต้องนำไปสู่การแลกมาด้วยค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane)



ภาพที่ 18 กราฟการเปรียบเทียบจำนวนข้อความของเกณฑ์วิธี OpenFlow กับจำนวนของสวิตช์ของ
 อัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้บนทอพอโลยี COST239
 ในแต่ละช่วงเวลา

บทที่ 5

บทสรุปและแนวทางในการพัฒนาต่อ

5.1 บทสรุป

ในปัจจุบันปริมาณการใช้บริการอินเทอร์เน็ตมีปริมาณเพิ่มขึ้น และอัลกอริทึมในการหาเส้นทางแบบดั้งเดิมที่จะทำการเลือกเส้นทางที่ดีที่สุดโดยเลือกเส้นทางที่สั้นที่สุดนั้น ยังไม่เพียงพอต่อการรองรับการให้บริการการรับส่งข้อมูลในปริมาณการใช้งานจำนวนมากได้ และเกิดความล่าช้าอย่างต่อเนื่องเป็นระยะเวลาที่ยาวนาน ดังนั้นในงานวิจัยนี้จึงเสนออัลกอริทึมในการหาเส้นทางที่พิจารณาปริมาณการเข้าคิวของข้อมูล บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ โดยอัลกอริทึมที่เสนอในงานวิจัยนี้ จะทำการตัดสินใจในการเลือกเส้นทางที่ขึ้นอยู่กับข้อมูลของปริมาณบัฟเฟอร์ที่คอนโทรลเลอร์ของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ได้รับ ซึ่งการเลือกเส้นทางที่สามารถใช้งานได้นั้น จะต้องสอดคล้องกับเกณฑ์ที่กำหนดไว้ โดยผลลัพธ์ที่ได้การทดลองจะแสดงให้เห็นว่าอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ มีประสิทธิภาพกว่าอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm บนทอพอโลยี COST239 ซึ่งเป็นทอพอโลยีใยแก้วนำแสงของข่ายงานบริเวณกว้าง หรือ WAN ที่เชื่อมต่อระหว่างหัวเมืองใหญ่ของทวีปยุโรป ในแง่ของผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่ายถึงร้อยละ 46 แห่งของช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) เป็นจำนวนร้อยละ 48 และแง่ของความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) เป็นจำนวนร้อยละ 23 นอกจากนี้แล้วยังได้ทำการทดลองบนทอพอโลยี SPLINTLINK ซึ่งเป็นทอพอโลยีใยแก้วนำแสงของข่ายงานบริเวณกว้าง หรือ WAN ที่เชื่อมต่อระหว่างหัวเมืองใหญ่ เช่นเดียวกับกับ COST239 แต่ในทอพอโลยีนี้จะเป็นหัวเมืองใหญ่ของประเทศสหรัฐอเมริกา โดยการทดลองดังกล่าวจะแตกต่างกัน ตรงที่ตำแหน่งของผู้ส่งและผู้รับในแต่ละคู่รับส่งจะมีจำนวนระยะทางที่ไกลกว่าของ COST239 ซึ่งผลลัพธ์ที่ได้การทดลองจะแสดงให้เห็นว่าอัลกอริทึมการหาเส้นทางที่พิจารณาขนาดของบัฟเฟอร์ที่เสนอในงานวิจัยนี้ มีประสิทธิภาพกว่าอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดแบบดั้งเดิมหรือ Traditional Shortest Path Algorithm เช่นเดียวกันทั้งแง่ของผลรวมของปริมาณงาน (Overall Throughput) ที่เกิดในระบบเครือข่ายถึงร้อยละ 37 แห่งของช่วงเวลาในการเดินทางของข้อมูล (Round Trip Time) เป็นจำนวนร้อยละ 165 และแง่ของความสูญเสียสะสมของกลุ่มข้อมูล (Packet Loss) เป็นจำนวนร้อยละ 114

นอกจากนั้นแล้วในงานวิจัยนี้ยังได้ทำการทดลองเพื่อปรับหาช่วงเวลาที่เหมาะสมในการเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์ โดย POX คอนโทรลเลอร์ เนื่องจากปัญหาปริมาณข้อมูลเกินขนาดบัฟเฟอร์ (Bufferbloat Problem) เป็นปัญหาที่เกิดขึ้นอย่าง

รวดเร็ว ดังนั้นในงานวิจัยนี้จึงได้หาช่วงเวลาที่จะไปเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์ โดย POX คอนโทรลเลอร์ที่เหมาะสม โดยผลลัพธ์ที่ได้การทดลองจะแสดงให้เห็นว่า ช่วงเวลาที่เหมาะสมกับการนำไปใช้งาน ในแง่ของผลรวมของปริมาณงาน (Overall Throughput) จะอยู่ที่ช่วงเวลาตั้งแต่ 0.01 วินาทีถึง 1.58 วินาที โดยในแต่ละช่วงเวลาก็จะแลกมาด้วยค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane) ที่ไม่เท่ากัน ซึ่งถ้าใช้ช่วงเวลาที่ยาวเกินไป ก็จะทำให้เกิดค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane) ค่อยข้างสูงตามไปด้วย และจะลดลงก็ต่อเมื่อใช้ช่วงเวลาที่สั้นลงเรื่อยๆ ตามลำดับ ดังนั้นในการนำไปใช้งานจริงสามารถปรับเปลี่ยนเพิ่มลดได้ แต่จะต้องคำนึงถึงค่าใช้จ่ายที่เพิ่มขึ้นบนส่วนของการควบคุม (Overhead of Control Plane) และประสิทธิภาพการทำงานของคอนโทรลเลอร์ที่จะนำไปใช้งานจริง รวมไปถึงลักษณะของสถานการณ์หรือทอพอโลยีที่นำไปใช้งาน ซึ่งโดยทั่วไปช่วงเวลาที่ 1 วินาทีนั้น เป็นช่วงเวลาที่หน่วยประมวลผลของคอนโทรลเลอร์โดยทั่วไปสามารถรองรับได้

5.2 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ

จากการทดลองพบว่า การนำไปใช้งานจริงในระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ จะต้องคำนึงถึงการทำงานของโปรโตคอลมาตรฐานที่ชื่อว่า OpenFlow ซึ่งคอนโทรลเลอร์ใช้ในการสื่อสารระหว่างส่วนของการควบคุมและส่วนของข้อมูล เพราะเนื่องจากในการทดลองในงานวิจัยนี้ได้ทำการทดลองเฉพาะระบบเครือข่ายเสมือนจริงจากโปรแกรมเลียนแบบพฤติกรรมการทำงานของระบบเครือข่ายหรือ Mininet ซึ่งสามารถจำลอง คอนโทรลเลอร์ สวิตช์ และเครื่องจุดปลายหลายเครื่องได้ ทำให้สามารถที่จะเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์ทั้งหมดในระบบโดยคอนโทรลเลอร์ได้ แต่ในโลกของความเป็นจริงนั้น คอนโทรลเลอร์จะไม่สามารถเรียกค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์ทั้งหมดในระบบได้ จากฟังก์ชันการเรียกค่าปริมาณบัฟเฟอร์ของอัลกอริทึมที่เสนอในงานวิจัยนี้ เพราะเนื่องจากฟังก์ชันการเรียกค่าปริมาณบัฟเฟอร์ดังกล่าว จำเป็นต้องมีช่องทางของแต่ละสวิตช์ทั้งหมดในระบบ อยู่ภายในเครื่องที่จะต้องใช้ทำทดลองเดียวกัน จึงจะสามารถค่าปริมาณบัฟเฟอร์ของแต่ละช่องทางของแต่ละสวิตช์ทั้งหมดในระบบได้ ดังนั้นจึงทำให้ต้องอาศัยความสามารถของโปรโตคอลมาตรฐาน OpenFlow ในการเรียกค่าบัฟเฟอร์ดังกล่าว ซึ่งจะมีรูปแบบการเรียกค่าบัฟเฟอร์ที่แตกต่างออกไป โดยเมื่อนำโปรโตคอลมาตรฐาน OpenFlow มาใช้งานแล้วจะทำให้อัลกอริทึมที่เสนอในงานวิจัยนี้ สามารถนำไปใช้แก้ไขปัญหาในโลกแห่งความเป็นจริงได้ รวมถึงยังทำให้ในงานวิจัยนี้มีความน่าสนใจอีกด้วย

รายการอ้างอิง

1. McKeown, N., et al., *OpenFlow: Enabling Innovation in Campus Networks*. SIGCOMM Comput. Commun. Rev., 2008. **38**(2): p. 69-74.
2. *OpenFlow Switch Specification: Version 1.0.0* Available from: <http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf>.
3. Tomovic, S. and I. Radusinovic. *Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks*. in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*. 2016.
4. Al-Jawad, A., et al. *BaProbSDN: A probabilistic-based QoS routing mechanism for Software Defined Networks*. in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*. 2015.
5. Mohan, P.M., T. Truong-Huu, and M. Gurusamy. *TCAM-Aware Local Rerouting for Fast and Efficient Failure Recovery in Software Defined Networks*. in *2015 IEEE Global Communications Conference (GLOBECOM)*. 2015.
6. Lee, D., P. Hong, and J. Li. *RPA-RA: A Resource Preference Aware Routing Algorithm in Software Defined Network*. in *2015 IEEE Global Communications Conference (GLOBECOM)*. 2015.
7. Huang, M., et al. *Dynamic routing for network throughput maximization in software-defined networks*. in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 2016.
8. Francois, F. and E. Gelenbe. *Towards a cognitive routing engine for software defined networks*. in *2016 IEEE International Conference on Communications (ICC)*. 2016.
9. Porxas, A.X., S.C. Lin, and M. Luo. *QoS-aware virtualization-enabled routing in Software-Defined Networks*. in *2015 IEEE International Conference on Communications (ICC)*. 2015.
10. Gettys, J., *Bufferbloat: Dark Buffers in the Internet*. IEEE Internet Computing, 2011. **15**(3): p. 96-96.

11. Gong, Y., et al. *Fighting the bufferbloat: On the coexistence of AQM and low priority congestion control*. in *2013 Proceedings IEEE INFOCOM*. 2013.
12. Pan, R., et al. *PIE: A lightweight control scheme to address the bufferbloat problem*. in *2013 IEEE 14th International Conference on High Performance Switching and Routing (HPSR)*. 2013.
13. Ferlin-Oliveira, S., T. Dreibholz, and Ö. Alay. *Tackling the challenge of bufferbloat in Multi-Path Transport over heterogeneous wireless networks*. in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*. 2014.
14. Park, S.Y., et al. *Impact of traffic splitting on the delay performance of MPTCP*. in *2014 IEEE International Conference on Communications (ICC)*. 2014.
15. Mahony, M., M.C. Sinclair, and B. Mikac., *Ultra-High Capacity Optical Transmission Network: European Research Project Cost239*. IEEE Internet Computing, 1993. **12**: p. 33-45.
16. Spring, N., R. Mahajan, and D. Wetherall. *Measuring ISP topologies with rocketfuel*. in *ACM SIGCOMM*. 2002.
17. *iPerf*. Available from: <https://iperf.fr>.



ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียนวิทยานิพนธ์

นายภควัต ชุ่มเจริญ เกิดเมื่อวันที่ 15 มกราคม พุทธศักราช 2535 ที่จังหวัดระยอง สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2557 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2558

