

# การจัดเส้นทางด้วยกลไกการคาดการณ์แบบดิวิดท์ในเครือข่ายเออสดีเอ็น

นาย นราภิชญ์ ประพิบัตนะ 60010509  
นาย กิตติธรรม ศิลาทรัพย์ 60010089  
รศ. ดร. ศักดิ์ชัย พิพัฒน์จักรุตัน อาจารย์ที่ปรึกษา  
ปีการศึกษา 2563

## บทคัดย่อ

เครือข่ายที่กำหนดโดยซอฟต์แวร์เป็นเทคโนโลยีด้านเครือข่าย ที่ได้ทำการเปลี่ยนแปลงโครงสร้างของอุปกรณ์โดยการแยกส่วนการควบคุมไปรวมศูนย์ไว้ที่คอนโทรลเลอร์ ที่ซึ่งสามารถทำงานกับภาษาโปรแกรมมิ่ง ช่วยให้สามารถนำการเรียนรู้ของเครื่องมาประยุกต์ใช้งานในด้านเน็ตเวิร์ก

โครงการนี้ได้นำเสนอโมเดลการเรียนรู้เชิงลึก “กลไกคาดการณ์แบบดิวิดท์ของลิงค์” ซึ่งนำโครงข่ายประสาทเทียมแบบวนกลับชนิด Long Short Term Memory (LSTM) มาใช้สร้างเน็ตเวิร์กแอปพลิเคชันที่จะทำงานอยู่บนริวเออสดีเอ็นคอนโทรลเลอร์ ซึ่งริวจะคอยเก็บรวบรวมข้อมูลสถิติของแพ็กเก็ตข้อมูล ส่งไปยังเน็ตเวิร์กแอปพลิเคชัน เพื่อนำไปใช้สำหรับการเรียนรู้และสร้างโมเดล LSTM แอปพลิเคชันดังกล่าวช่วยให้เราสามารถคาดการณ์ค่าการใช้ประโยชน์แบบดิวิดท์ล่วงหน้า และสามารถทำการกำหนดเส้นทางการส่งแพ็กเก็ตใหม่ได้ก่อนที่จะเกิดความคับคั่งของข้อมูลบนลิงค์ และลดอัตราการเกิดการสูญหายของแพ็กเก็ตข้อมูล

# **Routing Scheme based on Bandwidth Prediction Mechanism in SDN Networks**

Mr. Naravith Prateepratana 60010509

Mr. Kiratitam Silasap 60010089

Assoc. Prof. Dr. Sakchai Thipchaksurat Advisor

Academic Year 2020

## **Abstract**

Software-Defined Network technology is an approach which change the internal structure of network devices in which Control plane is separated from devices and centralized as Controller. SDN Controller can be programmed and control by common programming language. Therefore, intelligence and automatic management have been brought into Networking

In this paper, we propose a Deep Learning model for Software Defined Network called Link Bandwidth Utilization Prediction Application, work based on the Long-Short Term Memory (LSTM) Recurrent Neural Network. This proposed Network Application run within the RYU SDN Controller, Which can acquire the Network Traffic information from network devices and send to the Network Application for learning and modelling of LSTM neural network. The proposed Network Application could predict the Bandwidth Utilization of network links within the SDN Network in advance. This Network intelligence lead by SDN technology could make Re-routing before those specific links become congested and decrease the rate of packet loss.

## กิตติกรรมประกาศ

โครงการจัดเส้นทางด้วยกลไกการคาดการณ์แบบค์วิดที่และพลังงานในเครือข่ายอีสต์โคสเซ่นนี้ ผู้จัดทำขอบพระคุณที่ปรึกษาปริญญาภินพนธ์ รศ. ดร. ศักดิ์ชัย พิพิชัยกุลัตน์ ที่กรุณาให้ความรู้ และคำปรึกษาต่างๆ เกี่ยวกับแนวทางในการดำเนิน โครงการ ตั้งแต่เริ่มออกแบบระบบจนกระทั่งพัฒนา เสร็จสมบูรณ์ พร้อมทั้งกรรมการปริญญาภินพนธ์ ทุกท่านที่ร่วมให้คำแนะนำในการทำปริญญาภินพนธ์ให้ สัมฤทธิ์ผล ได้ด้วยดี ตลอดจนคณาจารย์ที่ให้ความรู้มาโดยตลอด

ขอกราบขอบพระคุณบุคลากรด้วย ที่ให้การอุปการะเลียงดู พร้อมทั้งสนับสนุน สำหรับการศึกษา ให้ได้รับการศึกษา และให้กำลังใจอันดีมาโดยตลอด ผู้จัดทำหวังเป็นอย่างยิ่งว่าปริญญาภินพนธ์นี้จะเป็น ประโยชน์กับท่านผู้สนใจและผู้เกี่ยวข้อง

นาย นราวนิชญ์ ประทีปรัตน์ 60010509

นาย กีรติธรรม ศิลาทรัพย์ 60010089

# สารบัญ

บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญรูป .....	VIII
สารบัญตาราง .....	X
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมา และความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ .....	3
1.3 ขอบเขตการวิจัย .....	3
1.3.1 ขอบเขตด้านเนื้อหา .....	3
1.3.2 ขอบเขตด้านแหล่งข้อมูล .....	4
1.3.3 ขอบเขตด้านระยะเวลา .....	4
1.4 ประโยชน์ที่ได้รับ.....	4
1.5 แผนการดำเนินงาน .....	5
บทที่ 2 ทฤษฎี และงานวิจัยที่เกี่ยวข้อง .....	9
2.1 ทฤษฎีที่เกี่ยวข้อง.....	9
2.1.1 เครื่องข่ายที่กำหนดโดยซอฟต์แวร์ .....	9
2.1.2 การจัดเส้นทาง.....	13
2.1.3 โอลเพนไฟล์วีโพร โทคอล .....	14

# สารบัญ (ต่อ)

2.1.4 อัลกอริทึมเส้นทางสั้นที่สุดของไอดีก์สตรา.....	17
2.1.5 การเรียนรู้เชิงลึก และโครงข่ายประสาทเทียม .....	17
2.1.6 Long-Short Term Memory (LSTM) .....	17
2.1.7 บัฟเฟอร์.....	17
2.1.8 การสูญหายของแพ็กเก็ต.....	18
2.1.9 แบบดิจิต์ และค่าการใช้ประโยชน์แบบดิจิต์ .....	18
2.1.10 ทรัพุต .....	18
2.1.11 ดีเลย์ .....	18
2.2 งานวิจัยที่เกี่ยวข้อง .....	19
บทที่ 3 วิธีการนำเสนอ .....	22
3.1 องค์ประกอบของระบบ .....	24
3.1.1 ชั้นโครงสร้างพื้นฐานเครือข่าย (Infrastructure Layer).....	24
3.1.2 ริเวอสเดี๋ยงคอนโทรลเลอร์ .....	26
3.1.3 แอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบบดิจิต์ LSTM.....	31
3.1.4 ขั้นตอนการสร้างแอปพลิเคชัน LSTM .....	32
3.2 ขั้นตอนการทำงานของระบบ.....	36
3.2.1 ขั้นตอนการทำงานของอัลกอริทึมการเลือกเส้นทางใหม่ .....	38
บทที่ 4 ผลการทดลอง.....	41
4.1 การพิจารณาอัลกอริทึมการเรียนรู้ที่เหมาะสม และการประเมินประสิทธิภาพของ อัลกอริทึมการเรียนรู้ .....	41

# สารบัญ (ต่อ)

4.1.1 Root Mean Squared Error และ Mean Absolute Error .....	41
4.2 การประเมินประสิทธิภาพของอัลกอริทึมการเลือกเส้นทางใหม่ .....	47
4.2.1 ทรูพด .....	47
4.2.2 การคำนวณทรูพดภาพรวม .....	47
4.2.3 การคำนวณปริมาณทราบฟิกที่เข้ามาในเครือข่าย .....	47
4.3 การเลือกค่าเทرزโซลเด็ที่เหมาะสมกับโครงงาน .....	48
4.3.1 ค่าเทرزโซลเด็ที่เหมาะสมกับระดับทราบฟิกต่างๆ .....	48
4.3.2 ค่าเทرزโซลเด็ที่เหมาะสมกับโครงงาน .....	51
4.4 การประเมินประสิทธิภาพของโครงงาน .....	51
4.4.1 พารามิเตอร์การทดลอง .....	52
4.4.2 ภายหลังการทดลอง .....	53
4.4.3 ผลการทดลอง .....	53
4.4.4 การทดลองเพิ่มเติม .....	54
4.4.5 ผลการทดลองเพิ่มเติม .....	56
บทที่ 5 สรุปผล และข้อเสนอแนะ .....	58
5.1 สรุปผลที่ได้จากการทดลอง .....	58
5.2 ปัญหาที่พบ .....	59
5.2.1 การขาดแคลงความรู้ทำให้เกิดความยากในการศึกษา .....	59
5.2.2 การลองผิดลองถูกกับอุปกรณ์หลายอย่างทำให้เริ่มต้นโครงงานได้ช้า .....	59
5.2.3 ข้อจำกัด และปัจจัยจำกัดของทราบฟิกเบนเนเรเตอร์ .....	59

5.3 แนวทางในการพัฒนาต่อ .....	60
บรรณานุกรม .....	61

# สารบัญ

รูปที่ 2.1 โครงสร้างพื้นฐานของเครือข่ายแบบเดิม .....	10
รูปที่ 2.2 โครงสร้างพื้นฐานของเครือข่ายเอกสารดิจิทัล .....	10
รูปที่ 2.3 Software-Defined Network Architecture ที่แบ่งโครงสร้างเป็น 3 Layer .....	11
รูปที่ 3.1 ภาพรวมของระบบงาน .....	22
รูปที่ 3.2 การเปรียบเทียบโครงสร้างระหว่าง SDN มาตรฐาน และโครงสร้าง SDN ของโครงงาน .....	23
รูปที่ 3.3 องค์ประกอบของระบบงานอย่างละเอียด .....	24
รูปที่ 3.4 โครงสร้างเครือข่ายที่สร้างโดยแอปพลิเคชัน SDN Flow Manager .....	25
รูปที่ 3.5 การตรวจสอบโทโพโลยี (Topology) ที่สร้างโดยมินิเน็ตด้วย SDN Flow Manager .....	25
รูปที่ 3.6 RYU SDN Framework มาตรฐาน .....	26
รูปที่ 3.7 RYU SDN Framework ที่ประยุกต์ใช้ในโครงงาน .....	27
รูปที่ 3.8 โมดูลสังเกตการณ์рафฟิกแสดงข้อมูลของลิงค์ในแต่ละวินาที .....	28
รูปที่ 3.9 การทำงานของอัลกอริทึมการเลือกเส้นทางใหม่ .....	30
รูปที่ 3.10 ค่าการใช้ประโยชน์แบบดัชนีที่ตราชันและคาดการณ์ว่าจะเกินเกราะໂไฮลด์ .....	31
รูปที่ 3.11 ภาพรวมของแอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบบดัชนีที่ .....	32
รูปที่ 3.12 การส่งข้อมูลค่าการใช้ประโยชน์แบบดัชนีที่จากวิเคราะห์เชิงมาสู่ปีเตอร์ .....	32
รูปที่ 3.13 สรุปกระบวนการสร้าง Dataset Matrix เริ่มตั้งแต่นำเข้าข้อมูล .....	34
รูปที่ 3.14 โครงสร้างของโครงข่ายประสาทเทียมชนิด LSTM ในโครงงาน .....	35
รูปที่ 3.15 Block Diagram แสดงภาพรวมขั้นตอนการทำงานของระบบ .....	36
รูปที่ 3.16 Flow Chart และขั้นตอนการทำงานโดยรวมของโครงงาน .....	37
รูปที่ 3.17 Flowchart ของ Re-Routing Algorithm .....	38
รูปที่ 3.18 Block Path และ Switch Path .....	40
รูปที่ 3.19 ผลที่เกิดขึ้นหลังจากการดำเนินการเปลี่ยนเส้นทาง .....	40
รูปที่ 4.1 ผลการวัดความแม่นยำของ Linear Regression .....	43
รูปที่ 4.2 ผลการวัดความแม่นยำของ Decision Tree Regressor .....	44
รูปที่ 4.3 ผลการวัดความแม่นยำของ Random Forest Regressor .....	45

## สารบัญรูป (ต่อ)

รูปที่ 4.4 ผลการวัดความแม่นยำของ Multi-Layer Perceptron.....	46
รูปที่ 4.5 ผลการวัดความแม่นยำของ Long-Short Term Memory.....	46
รูปที่ 4.6 ฟิล์ดภายในแพ็คเก็ต FlowStatReplyMessage .....	48
รูปที่ 4.7 ค่าทรูพุตที่ระดับเทอร์โบลดต่างๆ ในрафฟิคระดับต่ำ.....	49
รูปที่ 4.8 ค่าทรูพุตที่ระดับเทอร์โบลดต่างๆ ในrafฟิคระดับกลาง .....	49
รูปที่ 4.9 ค่าทรูพุตที่ระดับเทอร์โบลดต่างๆ ในrafฟิคระดับสูง.....	50
รูปที่ 4.10 เปรียบเทียบค่าทรูพุตเมื่อกำหนดค่าเทอร์โบลดที่ 60% 70% และ 80% ของแบนด์วิดท์ .....	51
รูปที่ 4.11 ผลการเปรียบเทียบระหว่างอัลกอริทึมต่างๆ.....	53
รูปที่ 4.12 トイโพลีแบบเชื่อมต่อ กันทั้งหมด .....	54
รูปที่ 4.13 トイโพลีอ้างอิงจากงานวิจัย Enhance Load Balanced.....	55
รูปที่ 4.14 ผลการทดลองบนトイโพลีแบบเชื่อมต่อ กันหมด .....	56
รูปที่ 4.15 ผลการทดลองบนトイโพลีอ้างอิงจากงานวิจัย Enhance Load Balanced .....	57

# สารบัญตาราง

ตารางที่ 1.1 แผนการดำเนินงานในช่วงเทอม 1 ปีการศึกษา 2563 .....	6
ตารางที่ 1.2 แผนการดำเนินงานในช่วงเทอม 2 ปีการศึกษา 2563 .....	8
ตารางที่ 2.1 OFPPortStatusReply.....	16
ตารางที่ 3.1 ตัวอย่าง ชุดข้อมูลที่ได้ดัดแปลงมาจาก OFPPortStatsReply .....	29
ตารางที่ 4.1 การเปรียบเทียบความแม่นยำของอัลกอริทึมการเรียนรู้.....	42
ตารางที่ 4.2 ความหมายของฟิลด์ต่างๆ ใน FlowStatReplyMessage .....	48
ตารางที่ 4.3 พารามิเตอร์ที่ใช้สร้างเครือข่ายสำหรับการทดลอง.....	52
ตารางที่ 4.4 พารามิเตอร์สำหรับกำหนดรูปแบบการส่งข้อมูลด้วยซอฟต์แวร์ D-ITG.....	52
ตารางที่ 4.5 พารามิเตอร์การส่งข้อมูลด้วยซอฟต์แวร์ D-ITG สำหรับการทดลองเพิ่มเติม .....	55

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมา และความสำคัญ

เครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software Defined Network) หรืออสดีเอ็น (SDN) เป็นแนวคิดการบริหารจัดการเครือข่ายแบบใหม่ ที่จะเปลี่ยนแปลงสถาปัตยกรรมของเครือข่ายให้มีการบริหารจัดการแบบรวมศูนย์ นอกจากจะช่วยแก้ปัญหาที่พบในเครือข่ายแบบเดิมซึ่งเราจะอธิบายต่อไป อสดีเอ็นทำให้การบริหารจัดการมีประสิทธิภาพมากขึ้นในอนาคต

Vendor-specific Command [6] หรือชุดคำสั่งสำหรับการตั้งค่าอุปกรณ์เครือข่ายที่แตกต่างกันไป ขึ้นอยู่กับผู้ผลิต เป็นปัญหาของสถาปัตยกรรมเครือข่ายแบบเดิมที่ผู้ดูแลเครือข่ายต้องพนอยู่เป็นประจำ แต่ในปัจจุบันเครือข่ายมีขนาดขยายใหญ่ขึ้น นอกจากนี้เครือข่ายยังประกอบไปด้วยอุปกรณ์จากผู้ผลิตหลากหลายบริษัทที่ต้องสามารถทำงานร่วมกันให้ได้ [7] ซึ่งประเด็นนี้อาจนำพาไปสู่ความผิดพลาดในการตั้งค่าอุปกรณ์ได้เนื่องจากผู้ดูแลเครือข่ายนั้นต้องเข้าไปจัดการอุปกรณ์เครือข่ายทุกเครื่องโดยตรง

โปรโตคอลการจัดเส้นทาง (Routing Protocol) หมายถึง โปรโตคอลที่ใช้ติดต่อกันระหว่างอุปกรณ์ ในขั้นการสื่อสารที่ 3 ของโมเดล ไอโอเอสไอ สำหรับแลกเปลี่ยนตารางเส้นทาง (Routing Information) เพื่อสร้างตารางเส้นทาง (Routing Table) ซึ่งช่วยให้อุปกรณ์ส่งต้นทางรู้เส้นทาง และสามารถส่งข้อมูล หรือแพ็คเก็ต (Packet) ไปยังอุปกรณ์ปลายทางได้ถูกต้อง [14] โดยตารางเส้นทางจะถูกสร้างด้วยวิธีการค้นหา “เส้นทางที่สั้นที่สุด” [15] เส้นทางที่ถูกเลือกไว้โดยอัลกอริทึมการจัดเส้นทาง (Routing Algorithm) อาจถูกคำนวนใหม่ได้ ความสามารถในการเลือกเส้นทางที่สั้นที่สุดใหม่นี้เรียกว่า Convergence [16] นอกจากนี้แล้วอุปกรณ์ เช่น เรตเตอร์ (Router) และสวิตช์ (Switch) สามารถพบปัญหาการสูญหายของแพ็คเก็ต (Packet Loss) ในสถานการณ์ที่ความสามารถในการส่งข้อมูลมีน้อยกว่าการรับข้อมูล เรตเตอร์ จำเป็นต้องหาพื้นที่ในการเก็บข้อมูลที่เข้ามาเพื่อรอการส่งต่อไป ทำให้เรตเตอร์จำเป็นต้องทำการจัดคิวแพ็คเก็ต (Packet Queuing) ด้วยบัฟเฟอร์ (Buffer) แต่ทั้งนี้ บัฟเฟอร์อาจถูกใช้งานจนเต็มความจุและทำให้เรตเตอร์ต้องเลือกที่จะละทิ้งแพ็คเก็ตบางส่วนไป [17]

เครือข่ายที่กำหนดโดยซอฟต์แวร์ หรืออสดีเอ็น (SDN) เป็นสถาปัตยกรรมเครือข่ายรูปแบบใหม่ ที่จะเข้ามาช่วยแก้ปัญหาของระบบเครือข่ายแบบดั้งเดิม โดยแนวคิดสำคัญของอสดีเอ็นคือ “เครือข่ายที่

สามารถโปรแกรมได้” และ “การแยกส่วนควบคุม (Control Plane) ออกจากส่วนการส่งข้อมูล (Data Plane)” [7] ที่เดิมจะถูกรวบกันไว้ในอุปกรณ์เครือข่ายเครื่องเดียวกัน การแยกเอาส่วนควบคุมออกมาร่วมไว้เป็นส่วนควบคุมศูนย์กลาง (Centralized Control) ที่จุดเดียวบนอสังหาริมทรัพย์ (SDN Controller) จะช่วยให้สามารถควบคุมระบบเครือข่ายทั้งหมดได้ในจุดเดียว [6]

การเรียนรู้เชิงลึก (Deep Learning) เป็นการเรียนรู้แบบอัตโนมัติ มีแนวความคิดจากโครงข่ายประสาทของมนุษย์ จำลองขึ้นเป็นโครงข่ายประสาทหรือที่เรียกว่า โครงข่ายประสาบทีม (Artificial Neural Network) ประกอบด้วยชั้นต่างๆ จำนวนหลายชั้น เพื่อทำการเรียนรู้รูปแบบ หรือจัดหมวดหมู่ของข้อมูลที่รับเข้ามา [12] การเรียนรู้เชิงลึกจะทำการแบ่งแยกข้อมูลและรายละเอียดต่างๆ ที่ได้รับมาทั้งหมด และวิเคราะห์ความสัมพันธ์ต่างๆ ในเชิงลึก คล้ายกับการกรองข้อมูลเป็นชั้นๆ และสรุปผลข้อมูลออกมาเป็นผลลัพธ์ หรือผลการคาดการณ์ ซึ่งอาจจะผิดหรือถูก ใกล้เคียงหรือไม่ใกล้เคียง หากไม่เดลการเรียนรู้เชิงลึกคาดการณ์ผิด จะเกิดการเรียนรู้และปรับเปลี่ยนการประมวลผลเพื่อให้ผลลัพธ์ที่ออกมาก็มีความถูกต้องมากขึ้น และยิ่งเรียนรู้มาก ไม่เดลการเรียนรู้เชิงลึกก็จะคาดการณ์ได้แม่นยำ และมีประสิทธิภาพที่เพิ่มขึ้น [11]

ในเครือข่ายคอมพิวเตอร์ (Computer Networks) สามารถนำเอาหลักการของการเรียนรู้ของเครื่อง (Machine Learning) และการเรียนรู้เชิงลึก มาประยุกต์ใช้เพื่อสร้างความชาญฉลาดให้กับเครือข่าย (Network Intelligence) เช่นการทำ Load Balanced Routing Algorithm ด้วย Reinforcement Learning [10], การทำ QoS Provisioning [3], Routing Optimization โดยการทำ Load Prediction [4], Deep-SDN สำหรับ Traffic Classification [13] เป็นต้น สำหรับโครงงานนี้เราได้นำโครงข่ายประสาททีมมาประยุกต์ใช้ในด้านการจัดการเพิ่มประสิทธิภาพการเลือกเส้นทางของอุปกรณ์ในเครือข่าย โดยการคาดการณ์ค่าการใช้ประโยชน์แบบวิดท์ (Bandwidth Utilization) สำหรับอนาคตบนลิงค์แต่ละลิงค์ หากค่าที่คาดการณ์ออกมาสามารถแปลความหมายได้ว่าจะเกิดความคับคั่งของข้อมูลขึ้น ได้อสังหาริมทรัพย์ โถงโตรลเลอร์จะทำการออกแบบคำสั่งเพื่อ Re-Routing ปรับเปลี่ยนเส้นทางที่ดีที่สุดในการส่งแพ็กเก็จใหม่ ดังนั้นการเรียนรู้เชิงลึกจึง สามารถเข้ามาช่วยในการเลือกเปลี่ยนเส้นทางใหม่สำหรับการสื่อสาร โดยอัตโนมัติได้

## 1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อศึกษาการทำงานของเครือข่ายที่กำหนดโดยซอฟต์แวร์ ซึ่งเป็นเทคโนโลยีใหม่ที่ใช้ในการบริหารจัดการเครือข่าย
- 2) เพื่อศึกษาทฤษฎีของการเรียนรู้เชิงลึกและการนำโครงข่ายประสาทเทียมเข้ามาประยุกต์ใช้กับงานด้านเครือข่ายคอมพิวเตอร์
- 3) เพื่อนำเสนออัลกอริทึมเพื่อปรับปรุงประสิทธิภาพของกระบวนการจัดเส้นทางในอีสต์อินโดฯ การประยุกต์ใช้ทฤษฎีการเรียนรู้เชิงลึก
- 4) เพื่อศึกษาระบวนการการทำวิจัยอย่างเป็นระบบ

## 1.3 ขอบเขตการวิจัย

ขอบเขตการวิจัยในโครงการนี้ถูกแบ่งออกเป็น 3 ด้าน ได้แก่ ด้านเนื้อหา ด้านแหล่งข้อมูล และด้านระยะเวลา โดยมีรายละเอียดดังนี้

### 1.3.1 ขอบเขตด้านเนื้อหา

โครงการนี้มีวัตถุประสงค์เพื่อศึกษาวิธีการทำการจัดเส้นทางให้มีความคลาดมากขึ้น โดยนำโครงข่ายประสาทเทียมมาใช้สำหรับการคาดการณ์ค่าการใช้ประโยชน์แบบดิจิตอลลิก เพื่อหลีกเลี่ยงการสูญเสียของแพ็กเก็ตที่บัฟเฟอร์ของอุปกรณ์เนื่องจากการที่บัฟเฟอร์ถูกใช้งานเต็มความจุ

- 1) มินินีต (Mininet) ซึ่งเป็นโปรแกรมจำลองเครือข่าย (Network Emulator) ใช้สำหรับการจำลองระบบเครือข่ายขึ้นมา โดยรับหน้าที่จำลอง OpenFlow สวิตช์ (switch), โฮสต์ (host) และลิงค์ ขึ้นมา ซอฟต์แวร์มินินีตในโครงการนี้ทำงานบนระบบปฏิบัติการอุบุนตุ (Ubuntu)
- 2) ริว (RYU) เป็นอีสต์อินคอน โตรลเลอร์ที่ทำหน้าที่เป็นส่วนการตัดสินใจหรือระบบปฏิบัติการของเครือข่าย (Network Operating System) หรือ NOS สามารถสั่งงานด้วยภาษาไฟฟอน โดย RYU จะทำหน้าที่รวบรวมข้อมูล Network Traffic จากอุปกรณ์ OpenFlow สวิตช์ที่มินินีตจำลองขึ้นมา ส่งให้กับเน็ตเวิร์กแอปพลิเคชันที่สร้างขึ้นโดยใช้โมเดล Long Short Term Memory (LSTM) ซึ่งเป็นโมเดลการเรียนรู้เชิงลึกประเกทหนึ่ง ทำหน้าที่คาดการณ์ค่าการใช้ประโยชน์แบบดิจิตอลในอนาคต

- 3) การเรียนรู้เชิงลึก และโครงข่ายประสาทเทียมแบบวนกับชั้นนิด LSTM เป็นส่วนสำคัญที่ช่วยให้ระบบเครือข่ายมีความชาญฉลาด สามารถปรับเปลี่ยนการทำงานไปตามสถานการณ์ต่างๆ

### 1.3.2 ขอบเขตด้านแหล่งข้อมูล

การวิจัยนี้เราใช้โปรแกรม Virtual Box สำหรับการใช้งานระบบปฏิบัติการอุปกรณ์ซึ่งเป็นระบบปฏิบัติการที่อุปกรณ์ทั้งริวอสตีอีนคอนโตรลเลอร์ และ มินิเน็ตสามารถทำงานได้

### 1.3.3 ขอบเขตด้านระยะเวลา

ระยะเวลาที่ใช้ในการวิจัยคือช่วงเดือนสิงหาคม 2563 จนถึงเดือนพฤษภาคม 2563

## 1.4 ประโยชน์ที่ได้รับ

ในปัจจุบันโภพโลจิ (Topology) ส่วนมากมีกระบวนการทางเส้นทางที่ไม่มีประสิทธิภาพเพียงพอ รองรับปริมาณข้อมูลจำนวนมากที่ไหลอยู่ในเครือข่ายได้ เนื่องจากเส้นทางการไหลของข้อมูลจะถูกตัดสินใจด้วยอัลกอริทึม Shortest Path แบบหนึ่งอยู่ตลอดเวลา โดยไม่มีการคาดการณ์ล่วงหน้าว่าในจะมีกรัฟเฟอร์รูปใช้งานเต็มความจุหรือไม่ ผลที่ตามมาจากการเต็มของบัฟเฟอร์คือการที่อุปกรณ์เครือข่ายจะทำการละทิ้งแพ็กเก็ต (Drop) ที่เข้ามาหลังจากนั้นทิ้งหมดจนกว่าจะสามารถสร้างพื้นที่ว่างในบัฟเฟอร์ได้ เหตุดังกล่าวเนื่องทำให้เกิดการสูญหายของแพ็กเก็ต

โครงการนี้มุ่งเน้นที่จะแก้ปัญหาเครือข่ายดังกล่าวโดยการประยุกต์ใช้โครงข่ายประสาทเทียมแบบวนกับชั้นนิด LSTM มาสร้างเน็ตเวิร์กแอปพลิเคชันสำหรับการคาดการณ์ค่าการใช้ประโยชน์แบบดัชนีที่จะเกิดขึ้นบนลิงค์ในอนาคตอันใกล้ โดยแอปพลิเคชัน ดังกล่าวจะทำงานอยู่บน ริวอสตีอีนคอน โตรลเลอร์ ที่จะเป็นตัวกลางการทำงานระหว่าง โครงสร้างพื้นฐานเครือข่าย (Network Infrastructure) และเน็ตเวิร์กแอปพลิเคชัน ค่าการใช้ประโยชน์แบบดัชนีที่ถูกคาดการณ์สำหรับใช้งานในอนาคตโดยโมเดล LSTM จะถูกนำมาใช้ในการตัดสินใจว่าลิงค์ใดที่จะเกิดความคับคั่งของข้อมูล (Congestion) แล้วจึงตัดสินใจเปลี่ยนเส้นทางการส่งแพ็กเก็ตด้วยการออกคำสั่งผ่านตัว ริวอสตีอีนคอน โตรลเลอร์ ไปยังโครงสร้างพื้นฐานเครือข่าย ต่อไป การเปลี่ยนเส้นทางส่งแพ็กเก็ตจะช่วยลดอัตราการสูญหายของแพ็กเก็ตเนื่องจากเราได้ทำการเปลี่ยนเส้นทางการส่งข้อมูล ก่อนที่จะเกิดความคับคั่งขึ้นจริง

## 1.5 แผนการดำเนินงาน

ลำดับ	ขั้นตอนการ ดำเนินงาน	สิงหาคม					กันยายน				ตุลาคม					พฤษจิกายน				
		3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	
6	ออกแบบและ สร้างโครงข่าย ประสานเที่ยม																			
7	จัดเตรียม รายงานและ ไฟล์การ นำเสนอ โครงการ																			

ตารางที่ 1.1 แผนการดำเนินงานในช่วงท่อน 1 ปีการศึกษา 2563



ลำดับ	ขั้นตอนการ ดำเนินงาน	มกราคม					กุมภาพันธ์					มีนาคม				เมษายน			
		4	11	18	25	31	1	8	15	22	1	8	15	22	5	12	19	26	30
6	จัดทำ ปริญญาอิพนธ์																		

**ตารางที่ 1.2 แผนการดำเนินงานในช่วงทอม 2 ปีการศึกษา 2563**

ตารางที่ 1.1 และตารางที่ 1.2 แสดงถึงแผนการดำเนินงานตลอดช่วงการทำโครงการ เริ่มตั้งแต่ เดือนสิงหาคม 2563 ไปจนถึงเดือนเมษายน 2564

## บทที่ 2

# ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในบทนี้ เราได้ทำการนำเสนอทฤษฎีของเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software Defined Network) หรืออีสต์ดีเอ็น (SDN) กระบวนการจัดเส้นทาง (Routing Process) การเรียนรู้เชิงลึก (Deep Learning) และตัวชี้วัดต่างๆ สำหรับวัดประสิทธิภาพของเครือข่าย (Performance Metrics) ตลอดจนถึงงานวิจัยต่างๆ ที่ได้นำการเรียนรู้ของเครื่อง (Machine Learning) และการเรียนรู้เชิงลึกเข้ามาประยุกต์ใช้เพิ่มประสิทธิภาพการทำงานของเครือข่ายอีสต์ดีเอ็น เช่น การคาดการณ์ Queue Utilization โดยใช้การเรียนรู้ของเครื่อง การใช้อัลกอริทึม SPFLR สำหรับการคาดการณ์ปริมาณโหลด (Load) ของเส้นทาง (Path) บนเครือข่ายขนาดใหญ่ เป็นต้น

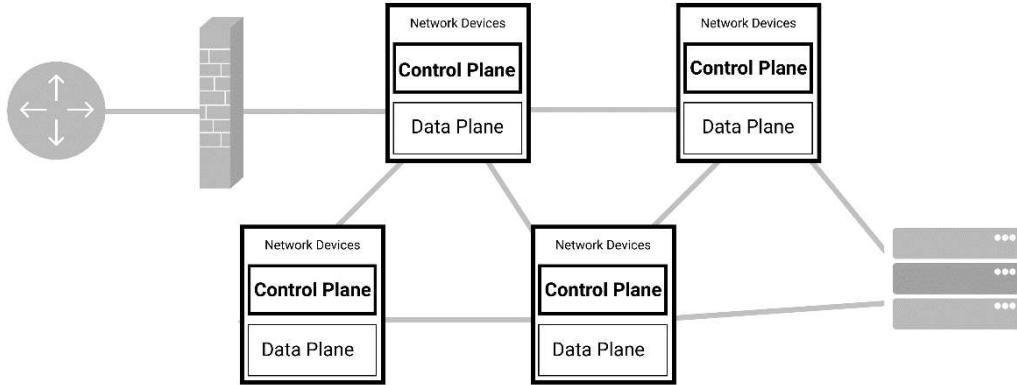
## 2.1 ทฤษฎีที่เกี่ยวข้อง

### 2.1.1 เครือข่ายที่กำหนดโดยซอฟต์แวร์

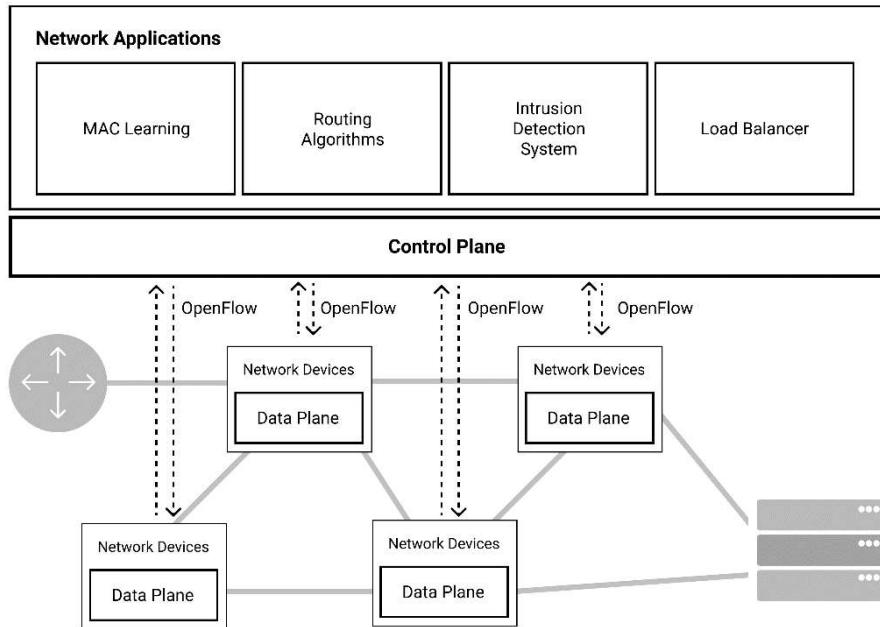
เครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software-Defined Network) หรืออีสต์ดีเอ็น (SDN) เป็นแนวคิดเกี่ยวกับสถาปัตยกรรมเครือข่ายที่จะเปลี่ยนแปลงเครือข่ายแบบดั้งเดิมให้กลายเป็น “Programmable Network” [19] หรือเครือข่ายที่สามารถควบคุมและออกแบบสั่งอุปกรณ์ผ่านภาษาโปรแกรมต่างๆ ได้ เช่น ไพทอน (Python) และจาวา (Java) เป็นต้น ซึ่งแตกต่างจากการตั้งค่าอุปกรณ์แบบเดิมที่รูปแบบคำสั่งที่ใช้ในการจัดการอุปกรณ์ก็จะแตกต่างไปตามผู้ผลิต

ในอุปกรณ์เครือข่ายแบบเดิมจะประกอบขึ้นด้วย ส่วนควบคุม (Control plane) เป็นส่วนประมวลผลของอุปกรณ์ และส่วนการส่งข้อมูล (Data plane) ที่รับหน้าที่ส่งข้อมูลออกไปยังอุปกรณ์ทั้งสองจะถูกรวบกันไว้ในอุปกรณ์เครื่องหนึ่งคั่งแสดงในรูปที่ 2.1

แนวคิดที่สำคัญของเครือข่ายที่กำหนดโดยซอฟต์แวร์อิกข้อนี้คือ การแยกส่วนควบคุม (Control Plane) ออกจากส่วนการส่งข้อมูล (Data Plane) ดังแสดงในรูปที่ 2.2 ให้อุปกรณ์เครือข่ายเครื่องหนึ่งมีเพียงส่วนการส่งข้อมูลเท่านั้น การแยกส่วนควบคุมของอุปกรณ์แล้วนำรวมเป็น “Centralized Control Plane” ที่อุปกรณ์เครื่องหนึ่งที่เรียกว่าคอนโทรลเลอร์ (Controller) จะช่วยให้การตั้งค่าอุปกรณ์ในเครือข่ายทำได้ในที่เดียว คำสั่งต่างๆ จากคอนโทรลเลอร์จะถูกส่งมายังอุปกรณ์เครือข่ายผ่านโปรโตคอล (Protocol) กลางที่เรียกว่า โพรโทคอลโอเพนไฟล์ (OpenFlow Protocol) [20]



รูปที่ 2.1 โครงสร้างพื้นฐานของเครือข่ายแบบเดิม



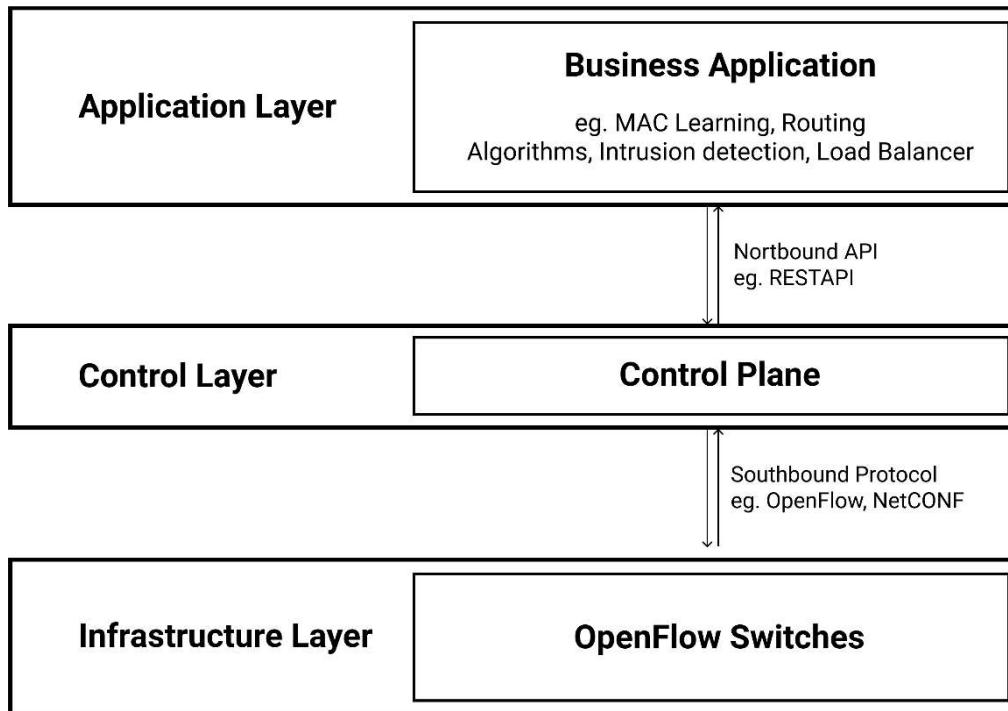
รูปที่ 2.2 โครงสร้างพื้นฐานของเครือข่ายอสดี.อี็น

ด้วยเหตุผลที่อธิบายไปข้างต้นจึงทำให้เครือข่ายที่กำหนดโดยซอฟต์แวร์ มีความยืดหยุ่นในการบริหารจัดการระบบเครือข่าย เนื่องจากไม่ต้องคำนึงถึงผู้ผลิตอุปกรณ์ว่าจะเป็นบริษัทเดียวกับที่ใช้อยู่เดิมหรือไม่ การขยายขนาดของเครือข่ายหรือการซื้ออุปกรณ์ใหม่เข้ามาติดตั้งเพิ่มทำได้

ง่ายขึ้น วางแผนง่ายขึ้น และยังประหยัดค่าใช้จ่าย [20] เนื่องจากไม่มีข้อผูกขาดเรื่องการเลือกใช้อุปกรณ์จากผู้ผลิตรายใดรายหนึ่ง เลือกอุปกรณ์ที่ถูกกลบแต่มาจากการอิสระ ไม่ต้องกังวล การตั้งค่าอุปกรณ์สามารถทำได้โดยใช้ภาษาโปรแกรมมิ่งภาษาเดียวกัน

#### 2.1.1.1 สถาปัตยกรรมของอีสเดีอีน (SDN Architecture)

สถาปัตยกรรมพื้นฐานของเครือข่ายกำหนดโดยซอฟต์แวร์ สามารถแบ่งออกเป็น 3 ส่วน ได้แก่ ชั้นเน็ตเวิร์กและพลิกชั้น ชั้นการควบคุม และชั้นโครงสร้างพื้นฐานเครือข่ายดังแสดงในรูปที่ 2.3



รูปที่ 2.3 Software-Defined Network Architecture ที่แบ่งโครงสร้างเป็น 3 Layer

- 1) ชั้นเน็ตเวิร์กและพลิกชั้น (Network Application Layer) เป็นชั้นของเครือข่ายที่กำหนดโดยซอฟต์แวร์ ที่ประกอบไปด้วยแอปพลิเคชันที่สามารถทำหน้าที่แทนอุปกรณ์มิดเดิลแวร์ (Middleware) ของเครือข่าย แบบเดิมเช่น ไฟร์วอลล์ (Firewall) หรือ โหลดබานเชอร์ (Load

- Balancer) แต่ภายในได้เครื่อข่ายที่กำหนดโดยซอฟต์แวร์ อุปกรณ์ฮาร์ดแวร์ เหล่านี้จะถูกเปลี่ยนให้อยู่ในรูปแบบของซอฟต์แวร์แทน [13]
- 2) ชั้นการควบคุม (Control Layer) เป็นส่วนสมองศูนย์กลางของเครือข่ายที่กำหนดโดยซอฟต์แวร์ รับหน้าที่จัดการและควบคุมนโยบายเครือข่าย (Network Policy) และการไหลของข้อมูล (Traffic Flow) ภายในเครือข่าย เอสดีเอ็นคอนโทรลเลอร์ (SDN Controller) มักทำงานอยู่บนเครื่องเซิร์ฟเวอร์ (Server) [25] โดยคอนโทรลเลอร์หนึ่งตัวสามารถควบคุม สวิตช์ได้พร้อมกันมากกว่าหนึ่งเครื่อง [21]
  - 3) ชั้นโครงสร้างพื้นฐานเครือข่าย (Network Infrastructure Layer) คือส่วนที่ประกอบด้วยสวิตช์ในเครือข่าย ชั้นนี้มีหน้าที่หลัก 2 อย่าง ได้แก่
    - ทำการส่งข้อมูลตามที่โฟล์วテーブิล (Flow Table) ระบุ [21]
    - ติดต่อกับคอนโทรลเลอร์ผ่านโอลิเอนโฟล์วโพรโทคอล (OpenFlow Protocol) เพื่อรายงานข้อมูลการเปลี่ยนแปลงของพอร์ต (Port changes) สถิติการไหลของข้อมูล (Flow Statistic) และสามารถส่งแพ็คเก็ต (Packet) ที่สวิตช์ยังไม่ทราบว่าต้อง จัดการส่งต่อไปอย่างไร [6]

#### 2.1.1.2 คำศัพท์ที่สำคัญเกี่ยวกับเอสดีเอ็น

ในเครือข่ายเอสดีเอ็นมีคำศัพท์ที่สำคัญซึ่งสามารถอธิบายได้ดังนี้

- 1) อุปกรณ์รับส่งข้อมูล (Forwarding Device) หมายถึงอุปกรณ์เครือข่ายทั้ง ในรูปแบบของฮาร์ดแวร์ (hardware-based) และซอฟต์แวร์ (software-based) ที่ทำหน้าที่ส่งแพ็คเก็ตออกไปตามพอร์ต ครอบแพ็คเก็ต (Drop) หรือ ส่งต่อแพ็คเก็ตไปยังเอสดีเอ็นคอนโทรลเลอร์ [6] เมื่อแพ็คเก็ตที่วิ่ง เข้ามาในสวิตช์ไม่แมตช์ (match) กับโฟล์วเอนทรี (Flow Entry) ได้ฯ ภายในโฟล์วテーブิล [21]
- 2) ส่วนการส่งข้อมูล (Data Plane) เป็นชื่อเรียกแทนกลุ่มของอุปกรณ์รับส่ง ข้อมูลที่เชื่อมต่อกันไม่ว่าจะด้วยวิธีเชื่อมต่อแบบมีสาย (Wire) หรือ เชื่อมต่อแบบไร้สาย (Wireless) กล่าวได้ว่าส่วนการส่งข้อมูลคือชั้น โครงสร้างพื้นฐานเครือข่ายนั้นเอง [6]

- 3) โพรโทคอลด้านใต้ (Southbound Protocol) เป็นโพรโทคอลกลางสำหรับการติดต่อระหว่างคอนโทรลเลอร์และอุปกรณ์รับส่งข้อมูล OpenFlow และ NETCONF ถือเป็นโพรโทคอลทิศใต้ที่ได้รับความนิยม [23]
- 4) โพรโทคอลด้านเหนือ (Northbound Protocol) เป็นโพรโทคอลที่เชื่อมต่อระหว่างเน็ตเวิร์กและแพลตฟอร์ม และความต้องการของผู้ใช้งาน โพรโทคอลนี้เป็นช่องทางสำหรับให้ค่อนโทรลเลอร์ส่ง บัญชีของเครือข่ายแบบ抽象แล็บแสตริก (Abstract Network View) ขึ้นไปยังชั้นแอปพลิเคชัน [23]
- 5) โอเพนไฟล์วีโพรโทคอล (OpenFlow Protocol) เป็นโพรโทคอลที่ตอบสนองกับแนวคิดของ เครือข่ายที่กำหนดโดยซอฟต์แวร์ โดยค่อนโทรลเลอร์ และอุปกรณ์รับส่งข้อมูลจะใช้โพรโทคอลนี้ในการติดต่อสื่อสาร, รับคำสั่ง, เพิ่มกฎ (Rule) ไว้สำหรับรองรับแพ็กเก็ตที่กำลังจะเข้ามาที่อุปกรณ์รับส่งข้อมูล [24]
- 6) ไฟล์ เทเบิล (Flow Table) เป็นตารางที่จะถูกเก็บไว้บนอุปกรณ์รับส่งข้อมูลแต่ละเครื่องซึ่งจะเก็บไฟล์เอนทรี หรือลิสต์ของกฎ ที่เป็นเหมือนชุดคำสั่งที่อุปกรณ์รับส่งข้อมูลต้องดำเนินการกับแพ็กเก็ตหนึ่งที่รับเข้ามา [21] โดยแต่ละไฟล์เอนทรีภายในไฟล์เทเบิลจะระบุข้อมูลเกี่ยวกับการส่งแพ็กเก็ตต่อไป เช่น Match field, Flow priority, Counters, Packet processing instruction, flow timeouts [22] ทั้งนี้การแก้ไขข้อมูลในไฟล์เอนทรีเป็นหน้าที่ของค่อนโทรลเลอร์ [21]

### 2.1.2 การจัดเส้นทาง

การจัดเส้นทาง (Routing) หมายถึงโพรโทคอลที่ใช้ในการแลกเปลี่ยนข่าวสารเส้นทาง (Routing Information) ระหว่างอุปกรณ์เครือข่ายที่ทำงานในระดับการสื่อสารระดับชั้นที่ 3 (Layer 3) หรือชั้นการสื่อสารเครือข่าย (Network Layer) และสร้างตารางเส้นทาง (Routing Table) ขึ้นมา เพื่อให้อุปกรณ์สามารถส่งแพ็กเก็ตข้อมูล ไปยังเครื่องปลายทาง ได้อย่างถูกต้อง โพรโทคอลกำหนดเส้นทางถูกแบ่งออกเป็น 2 ประเภท ได้แก่ การจัดเส้นทางแบบคงที่ และการจัดเส้นทางแบบพลา沃ต

### 2.1.2.1 การจัดเส้นทางแบบคงที่

การจัดเส้นทางแบบคงที่ (Static Routing Protocol) หมายถึง โพรโทคอลจัดเส้นทางชนิดที่ต้องกำหนดค่าเข้าไปในตัวเราเตอร์เองเพื่อที่จะบอกเราเตอร์ ว่าหากต้องการส่งแพ็คเก็ตไปยังปลายทางต่างๆแล้ว การออกไปยังชอปตัวถัดไป (Next hop) หรือเราเตอร์ตัวถัดไปจำเป็นต้องส่งออกไปที่อินเตอร์เฟซ (Interface) ได้ [27]

### 2.1.2.2 การจัดเส้นทางแบบพลวัต

การจัดเส้นทางแบบพลวัต (Dynamic Routing Protocol) หมายถึง โพรโทคอลจัดเส้นทางชนิดนี้จะทำการสร้างตารางเส้นทาง และติดต่อกับเราเตอร์เพื่อบ้านเพื่อสร้างเส้นทางการส่งข้อมูลที่ดีที่สุดด้วยตัวเอง อธิบายเพิ่มเติม ได้ว่าสามารถเรียนรู้เครื่องปลายทางได้เองโดยที่ไม่จำเป็นต้องให้มนุษย์ออกคำสั่งให้โดยตรง มีการปรับปรุงข้อมูลเส้นทางอยู่เสมอ สามารถหาเส้นทางใหม่ได้เมื่อเส้นทางเดิมเกิดปัญหาขึ้น [28] โพรโทคอลกำหนดเส้นทางแบบพลวัต ยังสามารถแบ่งออกเป็นอีก 2 ประเภท ได้แก่ IGP และ EGP [14] โดย IGP (Interior Gateway Protocol) เป็นโพรโทคอลการจัดเส้นทางที่มีการแลกเปลี่ยนตารางเส้นทางอยู่ภายใน AS (Autonomous System) เดียวกัน ซึ่งตีความได้ว่าเป็นระบบเครือข่ายที่อยู่ภายใต้การบริหาร โดยผู้ดูแลเดียวกัน หรือข้อกำหนด (Policy) เดียวกัน และสำหรับ EGP (Exterior Gateway Protocol) เป็นโพรโทคอลการจัดเส้นทางที่มีการแลกเปลี่ยนตารางเส้นทางระหว่างหลักหลาย AS

## 2.1.3 โอเพนโฟล์ฟอร์โพรโทคอล

โอเพนโฟล์ฟอร์โพรโทคอล (OpenFlow Protocol) หมายถึง โพรโทคอลด้านได้ที่ใช้ติดต่อสื่อสารกันระหว่าง โอเพนโฟล์สวิตช์ และ โอเพนโฟล์คอนโทรลเลอร์ ซึ่งหมายถึงอสเดีเอ็นคอนโทรลเลอร์ใดๆที่รองรับการทำงานร่วมกับโพรโทคอลโอเพนโฟล์

### 2.1.3.1 โอเพนโฟล์สวิตช์

โอเพนโฟล์สวิตช์ (OpenFlow Switch) หมายถึง อุปกรณ์สวิตช์ที่รองรับการทำงานร่วมกับ โอเพนโฟล์ฟอร์โพรโทคอล สามารถอยู่ในรูปแบบของชาร์ดแวร์ ซอฟต์แวร์ เช่น โอเพนวีสวิตช์ (Open V Switch) และอุปกรณ์จำลอง (Virtual Machine) เช่น มินิเน็ต ที่สามารถจำลอง โอเพนวีสวิตช์ได้

### 2.1.3.2 โอลีฟล็อว์พอร์ต

โอลีฟล็อว์พอร์ต (OpenFlow Port) หมายถึงพอร์ตที่ใช้ติดต่อสื่อสารระหว่างสวิตช์ที่สื่อสารกันด้วยโอลีฟล็อว์ไฟเบอร์ทูไฟเบอร์ ลูกแม่งออกเป็น 3 ประเภทได้แก่

- 1) พอร์ตทางกายภาพ (Physical Port) คือพอร์ตที่เชื่อมโยงโดยตรงกับพอร์ตจริงของเครื่องสวิตช์ ผู้ใช้งานสามารถตั้งค่าเพื่อเลือกให้พอร์ตใดๆ ของสวิตช์รับหน้าที่เป็นโอลีฟล็อว์พอร์ตชนิดนี้
- 2) พอร์ตทางโลจิคัล (Logical Port) คือพอร์ตที่ไม่มีอยู่จริงในทางกายภาพ เป็นพอร์ตที่สวิตช์ทำการจำลองขึ้นมา เช่น Link Aggregation groups, Tunnels และ Loop back
- 3) พอร์ตสำรอง (Reserved Port) คือพอร์ตที่ลูกจ่องไว้สำหรับการทำงานเฉพาะด้านที่สำคัญ เช่น การติดต่อกับคอนโทรลเลอร์ การทำ Flooding หรือการทำ Normal processing ซึ่งหมายถึงการส่งแพ็กเก็ตด้วยวิธีการทางเครือข่ายดั้งเดิม ที่สามารถทำได้ เช่น กัน

### 2.1.3.3 โอลีฟล็อว์ชานแนล

โอลีฟล็อว์ชานแนล (OpenFlow Channel) หมายถึง อินเตอร์เฟสที่เชื่อมต่อระหว่างโอลีฟล็อว์สวิตช์แต่ละเครื่องกับคอนโทรลเลอร์ ด้วยช่องทางดังกล่าว คอนโทรลเลอร์สามารถทำงานร่วมกับสวิตช์เพียงหนึ่งเครื่อง หรือมากกว่าหนึ่งเครื่องก็ได้ โดยจะติดต่อสื่อสารกันโดยใช้ข้อความโอลีฟล็อว์ (OpenFlow Message)

### 2.1.3.4 ข้อความโอลีฟล็อว์

ข้อความโอลีฟล็อว์ (OpenFlow Message) หมายถึง ลักษณะของข้อความที่คอนโทรลเลอร์และโอลีฟล็อว์สวิตช์ใช้ติดต่อสื่อสารกัน แบ่งออกเป็น 3 ประเภทได้แก่

- 1) Controller-to-switch เป็นข้อความที่สร้างโดยคอนโทรลเลอร์เพื่อส่งไปยังสวิตช์ สวิตช์อาจตอบกลับหรือไม่ขึ้นอยู่กับประเภทของข้อความ ตัวอย่างเช่น Modify-state ที่ใช้เพื่อออกรายการตั้งให้สวิตช์ทำการเปลี่ยนแปลงโอลีฟล็อว์เอนทรี , Read-state ที่ใช้สำหรับรวมรวมข้อมูลสถิติต่างๆจากสวิตช์
- 2) Asynchronous เป็นข้อความที่ลูกส่งจากสวิตช์ไปยังคอนโทรลเลอร์ โดยที่สวิตช์ไม่ต้องรอให้คอนโทรลเลอร์ออกคำสั่งมาก่อน ตัวอย่างเช่น

Packet-in ที่ใช้เมื่อสวิตช์ต้องการถามค่อน โตรเลอร์ถึงการรับมือกับแพ็คเก็ตที่ไม่รู้จัก ,Flow-Removed ที่สวิตช์ใช้แจ้งไปยังค่อน โตรเลอร์ว่าได้ทำการลบไฟล์เอนทรีแล้ว

- 3) Symmetric เป็นข้อความที่ทั้งสวิตช์ และค่อน โตรเลอร์สามารถส่งหากัน ได้อย่างไม่มีข้อบังคับใดๆ เช่น Hello ที่ใช้สำหรับเริ่มต้นการเชื่อมต่อ, Echo ที่ใช้เช็ค Liveliness ของผู้ทรงข้าม

#### 2.1.3.5 OpenFlow Port Status Reply

OpenFlow Port Status Reply เป็นข้อความโอเพนไฟล์ที่สวิตช์ใช้ตอบกลับไปยังค่อน โตรเลอร์ เพื่อบอกถึงข้อมูลทางสถิติที่เกี่ยวข้องกับการสื่อสาร เป็นข้อความที่โครงงานนี้ได้นำมาดัดแปลงเป็นอินพุตสำหรับอัลกอริทึมการเรียนรู้ ประกอบไปด้วยฟิลด์ดังต่อไปนี้

Index	ฟิลด์	คำอธิบาย
1	port_no	หมายเลขพอร์ต
2	rx_packets	จำนวนแพ็คเก็ตที่ได้รับ
3	Tx_packets	จำนวนแพ็คเก็ตที่ส่ง
4	Rx_bytes	จำนวนไบต์ที่ได้รับ
5	Tx_bytes	จำนวนไบต์ที่ส่ง
6	Rx_dropped	จำนวนแพ็คเก็ตที่ถูกครอบโดย RX
7	Tx_dropped	จำนวนแพ็คเก็ตที่ถูกครอบโดย TX
8	Rx_errors	จำนวน error ที่ได้รับ
9	Tx_errors	จำนวน error ที่ส่ง
10	Rx_frame_err	จำนวน frame alignment error
11	Rx_over_err	จำนวนแพ็คเก็ตที่มี RX overrun
12	Rx_crc_err	จำนวน CRC error
13	collisions	จำนวนครั้งที่เกิด collision

ตารางที่ 2.1 OFPPortStatusReply

### 2.1.4 อัลกอริทึมเส้นทางสั้นที่สุดของไดก์สตรา

อัลกอริทึมเส้นทางสั้นที่สุดของไดก์สตรา (Dijkstra's Shortest Path Algorithm) ถูกคิดค้นขึ้นโดยนักวิทยาการคอมพิวเตอร์ชาวดัตช์ แอ็ดส์เดอร์ ไดก์สตรา (Edsger Dijkstra) ในปี 1959 เป็นอัลกอริทึมสำหรับการหาเส้นทางที่สั้นที่สุดระหว่างโหนดต้นทาง และโหนดปลายทางโดยใช้หลักการของกราฟ

### 2.1.5 การเรียนรู้เชิงลึก และโครงข่ายประสาทเทียม

การเรียนรู้เชิงลึก (Deep Learning) หมายถึงซอฟต์แวร์ที่เลียนแบบการทำงานของโครงข่ายระบบประสาท (Nervous System) ของมนุษย์ โดยการเรียนรู้เชิงลึกถูกจัดว่าเป็นแบบหนึ่งของการเรียนรู้ของเครื่อง (Machine Learning) กล่าวโดยทั่วไปแล้วอัลกอริทึมของการเรียนรู้เชิงลึกถูกสร้างอยู่ในรูปแบบของโครงข่ายประสาทเทียม (Artificial Neural Network) หลายชั้นซึ่งมีความสามารถแบ่งออกเป็น 3 ส่วน ได้แก่ ชั้นอินพุต (Input Layer), ชั้นซ่อนตัว (Hidden Layer) และชั้นเอาท์พุต (Output Layer) เมื่อประกอบไปด้วยชั้นซ่อนตัวอย่างน้อย 2 ชั้น ก็จะถือว่าโครงข่ายนี้เป็นโครงข่ายประสาทเทียมเชิงลึก (Deep-Neural Network) [29]

### 2.1.6 Long-Short Term Memory (LSTM)

LSTM เป็นโครงข่ายประสาทเทียมแบบวนกับ (Recurrent Neural Network หรือ RNN) ซึ่ง RNN มีลักษณะการทำงานที่นำเอาผลลัพธ์จากการคำนวณ (Output) ที่ผ่านมาแล้วนำมาใช้เป็นอินพุต (Input) อีกรอบหนึ่ง ทำให้ RNN เหมาะที่จะใช้กับข้อมูลประเภทอนุกรมเวลา (Time Series) แต่ RNN ดังเดิมมีปัญหา Gradient Vanishing ซึ่งส่งผลให้ขนาดของ gradient ที่ถูกส่งค่าบันกลับ (Back Propagate) มีขนาดที่เล็กมาก จนทำให้น้ำหนัก (Weight) มีการเปลี่ยนแปลงที่ต่ำเกินไป ด้วยปัญหาดังกล่าวจึงมีการพัฒนา RNN ชนิด LSTM ที่สามารถแก้ปัญหาดังกล่าวได้ โดยภายในโหนด LSTM (LSTM Node) จะประกอบไปด้วยฟังก์ชันพิเศษเช่น อ่าน-เขียน-อัปเดตข้อมูล และลืมข้อมูล [30]

### 2.1.7 บัฟเฟอร์

นอกเหนือจากการตัดสินใจการส่งต่อแพ็คเก็ตข้อมูลไปยังปลายทางที่ถูกต้องแล้ว เราต้องยังต้องรับผิดชอบการบริหารจัดการความแออัดอีกด้วย เมื่อแพ็คเก็ตข้อมูลเข้ามาในอัตราที่สูงเกินกว่าที่เราต้องสามารถประมวลผลและส่งออกได้ เราต้องจำเป็นต้องมีพื้นที่สำหรับพักแพ็คเก็ตข้าม เอาไว้ พื้นที่ดังกล่าวคือบัฟเฟอร์ (Buffer) นั่นเอง [31] บัฟเฟอร์ทำการจัดเก็บแพ็คเก็ตอยู่ในลำดับแบบคิว (FIFO) ทั้งนี้บัฟเฟอร์เองก็สามารถถูกใช้งานจนเต็มความจุได้ ซึ่งส่งผลให้เกิดการสูญหายของแพ็คเก็ตตามมา เนื่องจากเราต้องทำการลงทะเบียนแพ็คเก็ตที่เข้ามาใหม่นั่นเอง [17]

### 2.1.8 การสูญหายของแพ็คเก็ต

การสูญหายของแพ็คเก็ต (Packet Loss) เกิดขึ้นเมื่อแพ็คเก็ตไม่สามารถถูกส่งไปยังปลายทางได้สำเร็จอันเนื่องมาจากสาเหตุหลายประการ เช่น อุปกรณ์ฮาร์ดแวร์ที่รับหน้าที่ส่งข้อมูลได้รับความเสียหาย ข้อมูลปริมาณสูงถูกส่งอยู่ในเครือข่ายในเวลาเดียวกัน ปัญหาซอฟต์แวร์ การเชื่อมต่ออินเทอร์เน็ตที่มีปัญหา เป็นต้น [33] ตัวอย่างของผลกระทบที่เกิดขึ้นจากการสูญหายของแพ็คเก็ตในชีวิตประจำวัน เช่น การประสบกับ Ping Spike หมายถึงอาการที่เราส่ง Ping ไปยังอุปกรณ์ปลายทางแล้วเกิดอาการล่าช้า (Lag) ขึ้น จนทำให้ช่วงเวลาหนึ่งมีค่ากราฟโดดมากกว่าปกติ ซึ่งส่งผลให้ดีโอสตรีมมิ่งที่กำลังรับชมอยู่เกิดการหยุดชั่วคราว [32]

### 2.1.9 แบนด์วิดท์ และค่าการใช้ประโยชน์แบนด์วิดท์

แบนด์วิดท์ (Bandwidth) หมายถึง ความกว้างของแถบคลื่นความถี่ ยิ่งแบนด์วิดท์สูง การรับส่งข้อมูลเข้า-ออก ก็ยิ่งมีประสิทธิภาพสูง เช่นเดียวกัน ในความหมายทั่วไปแบนด์วิดท์เป็นคำที่ใช้วัดความเร็วในการส่งข้อมูลของอินเทอร์เน็ต หรือกล่าวได้ว่าเป็นสัดส่วนโดยตรงของจำนวนข้อมูลทั้งหมดที่สามารถส่งผ่านหรือรับต่อหน่วยเวลาในความหมายเชิงคุณภาพ [34] สำหรับค่าการใช้ประโยชน์แบนด์วิดท์ (Bandwidth Utilization) เป็นค่าปรอร์เซนต์ที่บ่งบอกถึงปริมาณแบนด์วิดท์ที่ถูกใช้งานอยู่เมื่อเทียบกับปริมาณแบนด์วิดท์สูงสุดที่มี

### 2.1.10 ทรัพุต

ทรัพุต (Throughput) หมายถึง ปริมาณข้อมูลที่สามารถรับ และส่งได้ในระยะเวลาหนึ่ง นิยมวัดโดยมีหน่วยเป็นเมกะบิตต่อวินาที (Megabit per Second หรือ Mbps) ตัวอย่างของค่าทรัพุตที่เรา Mackup ในชีวิตประจำวัน เช่น ค่าความเร็วดาวน์โหลด (Download Speed) และความเร็วอัปโหลด (Upload Speed) ที่เห็นเมื่อทำการวัดค่าความเร็วของอินเทอร์เน็ตนั้นเอง

### 2.1.11 ดีเลย์

ดีเลย์ (Delay) หมายถึง ระยะเวลาทั้งหมดที่ใช้สำหรับการส่งแพ็คเก็ตจากโหนดต้นทางไปยังโหนดปลายทาง การเกิดดีเลย์นั้นมาจากการขยายปัจจัยด้วยกัน ได้แก่ กระบวนการทำงานภายในโหนด การถูกพักข้อมูล ไว้ที่บัฟเฟอร์, ดีเลย์ที่เป็นผลมาจากการแบนด์วิดท์ที่มีจำกัด และ Propagation Delay ที่หมายถึงระยะเวลาที่ข้อมูลสามารถถึงไปได้บนตัวกลางภายในระยะเวลา 1 วินาที

## 2.2 งานวิจัยที่เกี่ยวข้อง

จากการศึกษาที่เกี่ยวกับเครือข่ายที่กำหนดโดยซอฟต์แวร์ การจัดสื่อสาร การเรียนรู้เชิงลึก และโมเดลการเรียนรู้เชิงลึกแบบวนกลับชนิด LSTM เราได้ค้นคว้าเอกสารงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

S. WilsonPrakash and P. Deepalakshmi (2019) [1] ได้ศึกษาเกี่ยวกับการเพิ่มประสิทธิภาพของการให้ทรัพยากร (Resource) ของการคำนวณแบบคลาวด์แก่ คอมพิวเตอร์เสมือน (Virtual Machine หรือ VM) เนื่องจากถ้า VM เกิดการโอเวอร์โหลด (overload) จะทำให้ประสิทธิภาพการทำงานลดลง แต่ถ้าอันเดอร์โหลด (underload) จะเป็นการใช้ทรัพยากรอย่างไม่มีประสิทธิภาพ ซึ่งงานวิจัยนี้ก็จะนำ Back Propagation Artificial balancing Neural Network (BPANN) มาคาดการณ์โหลดของ VM ที่จะเกิดขึ้น และทำการปรับสมดุลของการใช้งานทรัพยากร

Haipeng Yao , Xin Yuan , Peiying Zhang , Jingjing Wang , Chunxiao Jiang , and Mohsen Guizani (2019) [2] ได้ศึกษาเกี่ยวกับการเพิ่มประสิทธิภาพของการกำหนดเส้นทาง (Routing) ด้วยกระบวนการ 3 ขั้นตอน คือ Dimension Reduction , The QU Prediction และ Routing Based On Queue Utilization ตามลำดับ โดยขั้นตอนแรกจะทำเพื่อลดความซับซ้อนของข้อมูลลงให้เป็นที่เข้าใจได้ง่ายขึ้น เมื่อมีการมองชุดข้อมูลด้วยมนุษย์ใหม่ และไม่ได้ทำการเปลี่ยนแปลงข้อมูลเดิมเลยแม้แต่ตัวเดียว ซึ่งจะทำให้หา Next hop ได้ดียิ่งขึ้น ต่อมาขั้นตอนที่ 2 จะเป็นการรวมรวมข้อมูลของทุกเส้นทางเพื่อพิจารณาและหลีกเลี่ยงความคับคั่ง (congestion) ที่จะเกิดขึ้นในอนาคต ซึ่งจะเป็นปัญหาแบบ Non-linear ซึ่งทางคณะวิจัยได้เลือกใช้โครงข่ายประสาทเทียมเข้ามาช่วยในการแก้ปัญหา ในส่วนของขั้นตอนสุดท้ายจะเป็นการนำการเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึกมาช่วยในการพิจารณา Queue Utilization เพื่อที่จะหลีกเลี่ยงการสูญเสียของแพ็กเก็ตที่เกิดจากบัฟเฟอร์ที่มีไม่เพียงพอต่อการรองรับ แพ็กเก็ตที่เข้ามาจำนวนมาก ซึ่งทำให้แต่ละการกำหนดเส้นทางสามารถเลือก Next hop ที่อาจจะไม่ใช่เส้นทางที่ไปยังปลายทางได้เร็วที่สุด แต่เป็นเส้นทางที่มี Queue- Utilization ต่ำ ทำให้การใช้ทรัพยากรนั้นมีประสิทธิภาพ

R. Dahlia Devapriya and S. Indira Gandhi (2020) [3] ได้ศึกษาเกี่ยวกับการเพิ่มประสิทธิภาพของการทำโหลดбалานซ์ (Load Balancing) และให้ความสำคัญกับคุณภาพของการให้บริการ (Quality of Service หรือ QoS) มากขึ้น โดยพิจารณาว่าแพ็กเก็ตที่เข้ามายังเป็น QoS หรือไม่ด้วยวิธีการ Differentiated Service Framework ซึ่งจะเป็นการแบ่งคลาสของข้อมูล ถ้าหากเป็นคลาสเดียวกันก็จะใช้วิธีส่งข้อมูลแบบเดียวกัน โดยงานวิจัยชุดนี้จะแบ่งออกเป็น 2 คลาส ได้แก่ เป็น QoS และไม่เป็น QoS ซึ่งถ้าหากเป็น QoS ก็จะทำการเลือกเส้นทางที่มีการใช้งานน้อยที่สุด จากนั้นจะทำการลือกเส้นทางนั้นสำหรับ QoS Flow แต่ถ้าหากไม่ใช่ก็จะทำการเลือกเส้นทางที่สั้นที่สุด (Shortest Path) โดยการดูจำนวนรอบแพนซึ่ง

ในโทโพโลยี (Topology) ก็จะมีการเชื่อมความคับคั่งอยู่ตลอดเวลา ซึ่งหากเกิดความคับคั่งขึ้น ก็จะทำการเปิดใช้งานโหลดบล๊อคเพื่อลดอัตราการสูญหายของแพ็กเก็ต

Manel Majdoub , Ali El Kamel , and Habib Youssef (2020) [4] ได้ศึกษาเกี่ยวกับอัลกอริทึมเดิมที่มีชื่อว่า Predicting of Future Load-based Routing (PFLR) ซึ่งเป็นอัลกอริทึมการกำหนดเส้นทางที่ไม่เหมาะกับเน็ตเวิร์กขนาดใหญ่ เนื่องจากมีลิงค์จำนวนมากที่ต้องควบคุม ทำให้ผู้วิจัยของงานวิจัยชุดนี้คิดอัลกอริทึมใหม่ที่สามารถรองรับการทำงานกับเน็ตเวิร์กขนาดใหญ่ได้ มีชื่อว่า Scalable Predicting of Future Load-based Routing (SPFLR) เป็นอัลกอริทึมที่ได้นำเอาโครงสร้างพื้นฐานแบบเพอร์เซปตรอนหลายชั้น (Multi-Layer Perceptron หรือ MLP) เข้ามาใช้คาดการณ์ค่าการใช้งานแบบดิวิดท์ (Bandwidth Utilization) ของเส้นทาง (path) ในอนาคต และนำไปใช้คำนวณเส้นทางที่ดีที่สุดใหม่วร่วมกับขั้นตอนวิธีของไดก์สตรา (Dijkstra's Algorithm) ผู้วิจัยได้เลือกใช้ MLP จำนวน 1 โมเดลต่อ 1 เส้นทาง ซึ่งต่างกับอัลกอริทึม PFLR ที่ได้เลือกใช้ MLP 1 โมเดลต่อ 1 link อธิบายอย่างกระชับได้ว่า SPFLR เลือกที่จะคาดการณ์ค่าการใช้งานแบบดิวิดท์ของ Path ต่างกับ PFLR ที่ได้เลือกคาดการณ์เพียงค่าการใช้งานแบบดิวิดท์ของลิงค์ซึ่งทำให้ SPFLR สามารถจัดการกับปัญหา Scalability ที่ PFLR ประสบอยู่

Kuldeep Kurroliya , Sagarika Mohanty , Khushboo Kanodia , and Bibhudatta Sahoo (2020) [5] งานวิจัยนี้ได้นำเสนอ Edge Weight Optimization Algorithm รูปแบบใหม่ในชื่อ Grey Wolf Aware Load balancing and Energy saving (GLE) ซึ่งสามารถปรับการใช้พลังงานและโหลดบล๊อคในเครือข่ายที่กำหนดโดยซอฟต์แวร์ โดย GLE นี้สามารถนำมาระบุกตัวกับกลไกในปัจจุบันที่ใช้แนวคิด Active / Inactive ได้ เพื่อเพิ่มประสิทธิภาพในการใช้พลังงานซึ่งจะไม่ส่งผลกระทบต่อการรับรู้ ปริมาณข้อมูล (Traffic) ที่กำลังวิ่งอยู่ในเครือข่าย

Truong Thu Huong ; Ngo Do Dang Khoa ; Nguyen Xuan Dung ; Nguyen Huu Thanh (2019) [10] งานวิจัยนี้ได้นำเสนอวิธีการทำ Multipath Load Balanced Routing Algorithm ที่ทำงานบนพื้นฐานของการเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) ความสามารถในการมองเห็น “Global View” ของแอสเดีอีนคอนโทรลเลอร์ ถูกนำมาใช้ประโยชน์เพื่อแก้ปัญหาดีเลย์และ การใช้งานทรัพยากรเน็ตเวิร์ก (Network Utilization) คอนโทรลเลอร์สามารถรับรู้ถึงค่าสถิติ เช่น rx\_packets, tx\_bytes, rx\_dropped ของสวิตช์เบื้องล่าง ได้โดยใช้ OpenFlow แพ็กเก็ตชนิด OFPPortStatsRequest ส่งไปยังสวิตช์ และสวิตช์ที่ได้รับก็จะตอบกลับด้วย OFPPortStatsReply ค่าสถิติดังกล่าวจะถูกนำไปคำนวณและสร้างเป็นทรัพฟิกเมทริกซ์ (Traffic Matrix) สำหรับเป็นอินพุตของโครงสร้างพื้นฐานของโครงสร้างพื้นฐาน “Policy Maker” ที่จะเรียนรู้และให้คำแนะนำหนักของแต่ละลิงค์อันเป็นการทำโหลดบล๊อคในเครือข่าย

แล้วยังมีโครงข่ายประสาทเทียมชื่อ “Estimator” ที่รับหน้าที่ประเมินผลและปรับเปลี่ยนหน้าหนักที่ Policy Maker ให้ค่าไว อันเป็นกระบวนการส่งค่าข้อมูลกลับ (Back Propagation) นั่นเอง

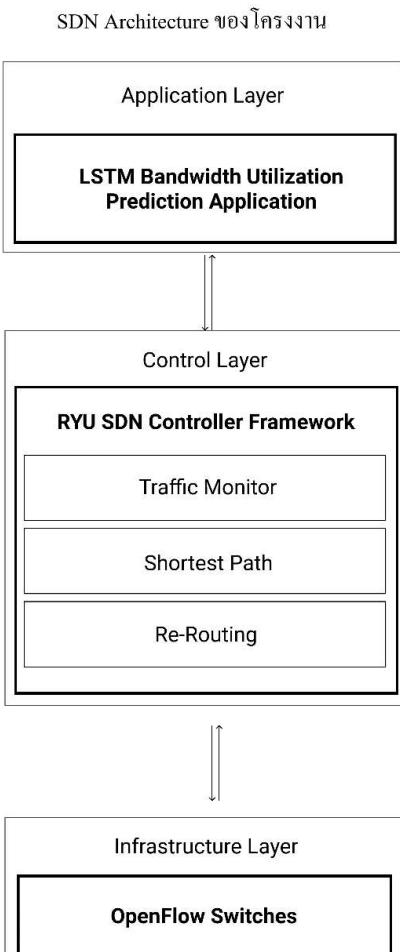
Yi-Ren Chen ; Amir Rezapour ; Wen-Guey Tzeng ; Shi-Chun Tsai (2020) [18] งานวิจัยชิ้นนี้ได้นำเสนออัลกอริทึมการกำหนดเส้นทาง “RL-Routing” ที่ได้นำการเรียนรู้แบบเสริมกำลังมาประยุกต์ใช้โดยให้มีเอเย่นต์ (Agent) จำนวนหนึ่งตัวต่อสวิตซ์หนึ่งเครื่อง สำหรับโครงสร้างเครือข่ายกำหนดโดยซอฟต์แวร์ ในส่วนของชั้นเน็ตเวิร์กแอปพลิเคชันนั้นประกอบด้วย 2 แอปพลิเคชัน ได้แก่ OpenFlow Network Discovery สำหรับการมองเห็นไฟฟ้าโลหะของส่วนการส่งข้อมูลเบื้องล่าง และ RL-Routing Application ซึ่งเป็นโมดูลการทำงานหลัก ภายใน RL-Routing Application ยังถูกแบ่งออกเป็น “Network Monitoring Module” สำหรับรวบรวมข้อมูลเน็ตเวิร์ก (Network Information) และ “Action Module” สำหรับการส่ง Openflow Message เพื่อเปลี่ยนแปลงไฟล์เทเบิลของสวิตซ์ ข้อมูลเน็ตเวิร์กจะถูกนำไปคำนวณต่อสำหรับใช้เป็น “State” และใช้สร้างโมเดลการเรียนรู้ต่อไป งานวิจัยชิ้นนี้ได้ทำการทดลองกับไฟฟ้าโลหะ 3 แบบ ได้แก่ Fat-tree, NSFNet, ARPANET

ในโครงงานของเราเน้นในการใช้การเรียนรู้เชิงลึกประเกท โครงข่ายประสาทเทียมแบบวนกลับ ชนิด Long Short Term Memory (LSTM) มาประยุกต์ใช้กับเครือข่ายที่กำหนดโดยซอฟต์แวร์เพื่อคาดการณ์ค่าการใช้งานแบบดิจิทัลที่อนาคตของแต่ละลิงค์ภายในเน็ตเวิร์ก และทำการเปลี่ยนเส้นทางเพื่อหลีกเลี่ยงเหตุการณ์ที่อาจจะส่งผลกระทบให้อัตราการสูญหายของแพ็กเก็ต และดีเลย์ของลิงค์เพิ่มสูงขึ้น

## บทที่ 3

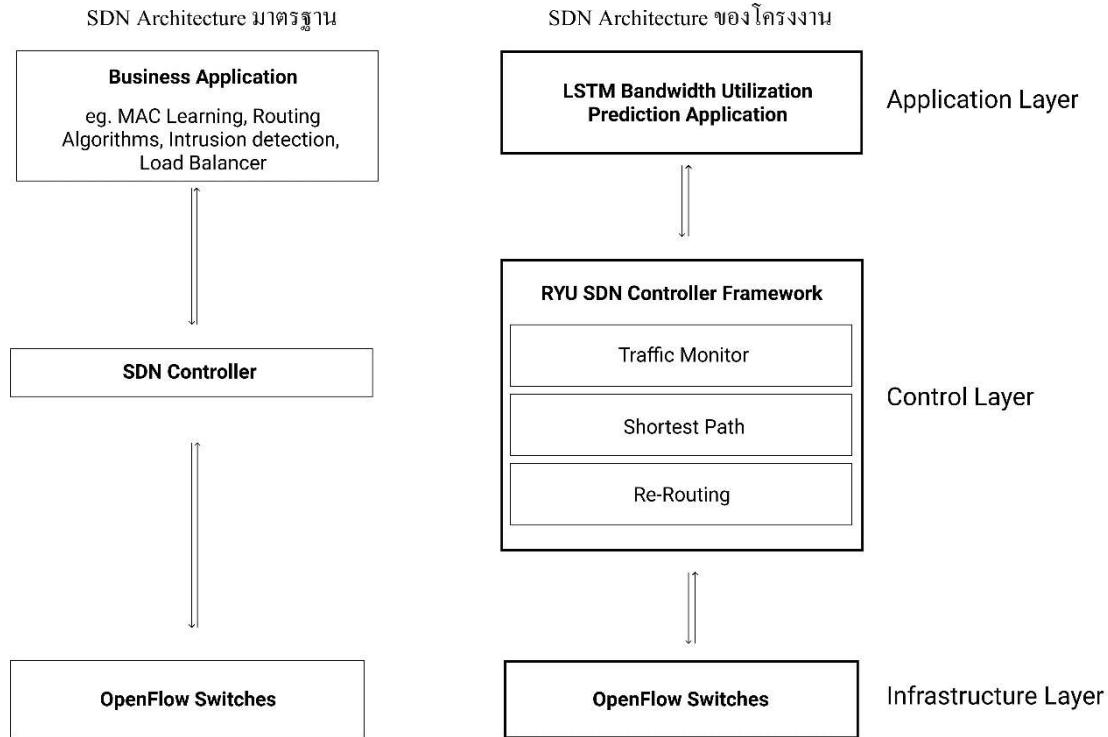
### วิธีการนำเสนอด้วย LSTM

ในบทนี้ เราได้ทำการนำเสนอด้วย LSTM ของโครงสร้าง องค์ประกอบ และลำดับการทำงานทั้งหมดของ SDN ที่อันประกอบขึ้นจากองค์ประกอบหลัก 3 ส่วน ได้แก่ โอลฟ์ฟล็อว์สวิตช์ (OpenFlow Switches) ที่ถูกจำลองขึ้น ริวเอสดีเอ็นคอนโทรลเลอร์ (RYU SDN Controller) และ แอปพลิเคชันสำหรับคาดการณ์ค่าการใช้ประโยชน์แบบดิจิตท์โดยใช้โน้ตเครื่องเรียนรู้เชิงลึก LSTM ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ภาพรวมของระบบงาน

รูปที่ 3.1 แสดงให้เห็นถึงภาพรวมและการทำงานร่วมกันขององค์ประกอบหลัก 3 ส่วนของระบบ และในรูปที่ 3.2 เราได้แสดงการเปรียบเทียบระหว่างโครงสร้างเอกสารีเอนมาตรฐาน กับโครงสร้างของโครงงานที่ได้ประยุกต์มาจากการสร้างมาตรฐาน

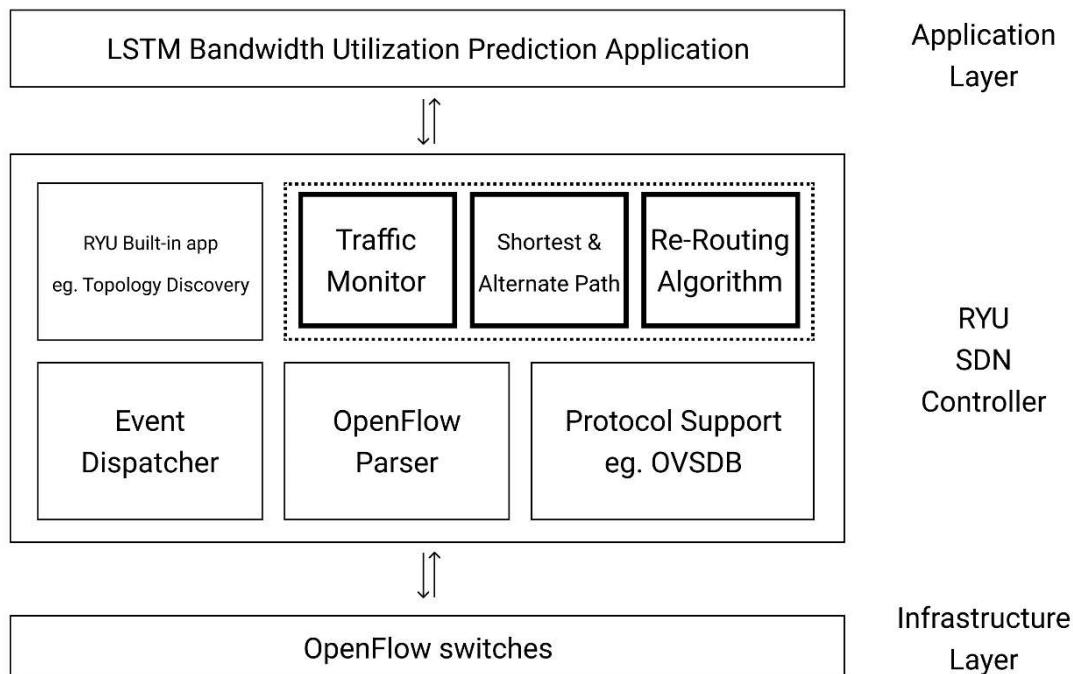


รูปที่ 3.2 การเปรียบเทียบโครงสร้างระหว่าง SDN มาตรฐาน และโครงสร้าง SDN ของโครงงาน

เราได้สร้างแอปพลิเคชันสำหรับการคาดการณ์ค่าการใช้ประโยชน์แบบควิกท์ในชั้นแอปพลิเคชัน (Application Layer) เลือกใช้ริวอสตีเอนคอนโทรลเลอร์ในชั้นการควบคุม (Control Layer) และจำลองโอลูนไฟล์สวิตช์ชิ้นมาด้วยซอฟต์แวร์มินิเน็ต (Mininet) ในชั้นโครงสร้างพื้นฐานเครือข่าย

### 3.1 องค์ประกอบของระบบ

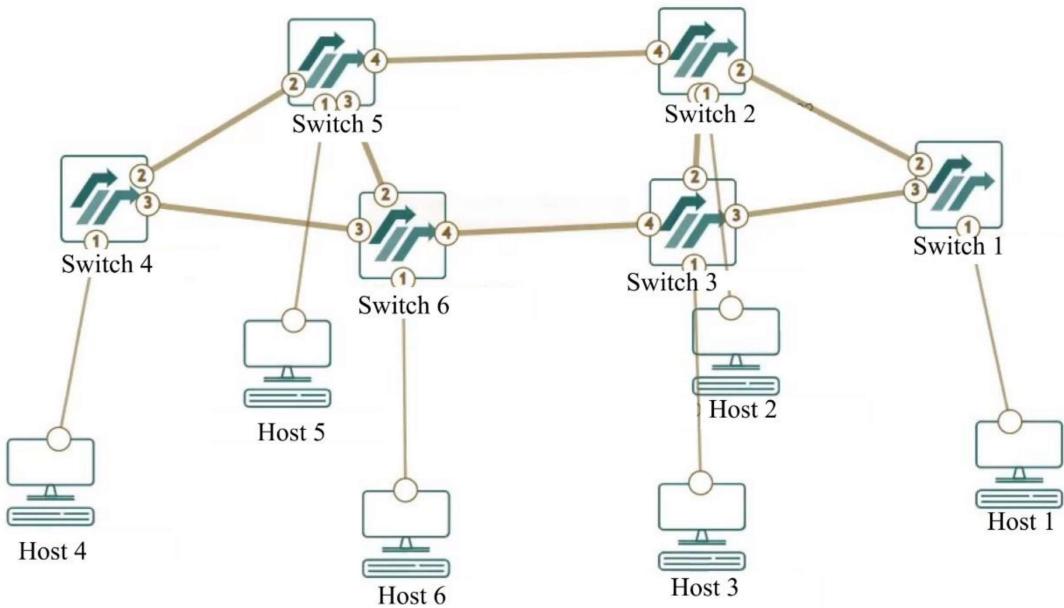
ระบบของโครงงานนี้มีองค์ประกอบหลัก 3 ส่วน ได้แก่ กลุ่มของโอเพนไฟล์สวิตช์และไฮสท์ในชั้นโครงสร้างพื้นฐานเครือข่าย ริเวอสตีเอ็นคอนโทรลเลอร์ และแอปพลิเคชันภาคการณ์ค่าการใช้ประโยชน์แบบดิวิดท์ที่ใช้งานโดยเดลการเรียนรู้เชิงลึกประเภท Long-Short Term Memory (LSTM) ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 องค์ประกอบของระบบงานอย่างละเอียด

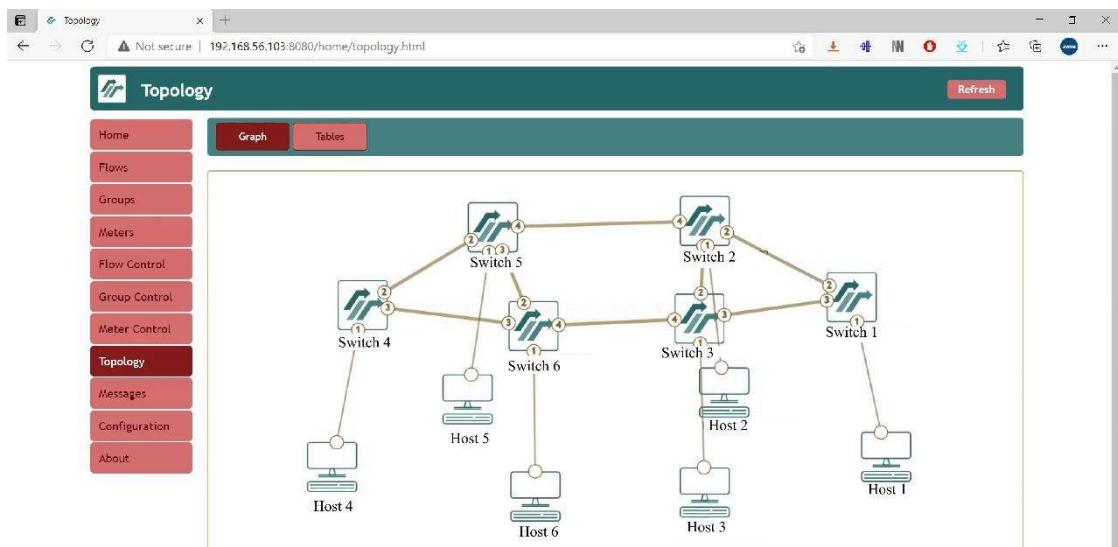
#### 3.1.1 ชั้นโครงสร้างพื้นฐานเครือข่าย (Infrastructure Layer)

เครือข่ายของอุปกรณ์โอเพนไฟล์สวิตช์ 6 เครื่องและไฮสท์ (host) 6 เครื่องเชื่อมต่อกันดังแสดงในรูปที่ 3.4 โอเพนไฟล์สวิตช์และไฮสท์ทั้งหมดถูกจำลองขึ้นด้วยซอฟต์แวร์มินิเน็ต (Mininet) ซึ่งทำงานอยู่บนระบบปฏิบัติการอูบุนตู (Ubuntu)



รูปที่ 3.4 โครงสร้างเครือข่ายที่สร้างโดยแอปพลิเคชัน SDN Flow Manager

ขั้นตอนการสร้างพื้นฐานเครือข่ายจะเชื่อมต่อกับรีวิวคอนโทรลเลอร์ด้วยโอลีฟอนฟล็อกว์โปรโทคอล (OpenFlow Protocol) ในโครงงานของเราได้กำหนดให้มีการสื่อสารระหว่างโอลีฟท์ต้นทาง 3 เครื่อง ได้แก่ Host 1, Host 2 และ Host 3 ส่งข้อมูลไปยังโอลีฟท์ปลายทาง 3 เครื่อง ได้แก่ Host 4, Host 5 และ Host 6

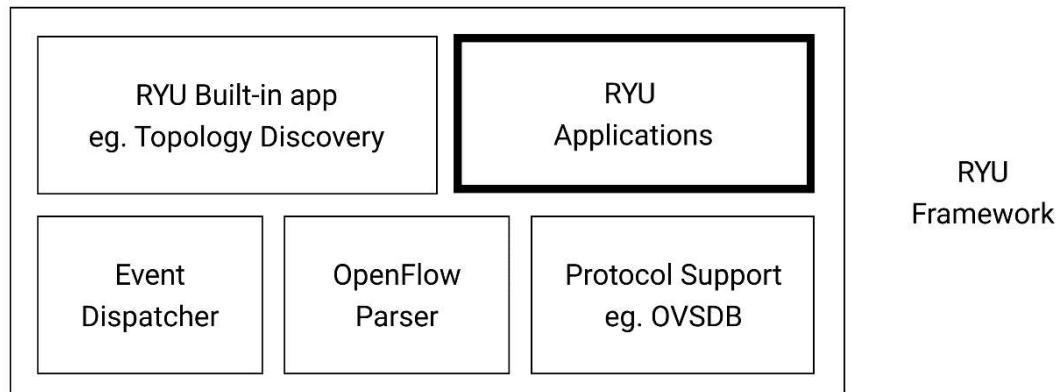


รูปที่ 3.5 การตรวจสอบ拓扑ology (Topology) ที่สร้างโดยมินิเน็ตด้วย SDN Flow Manager

เราสามารถตรวจสอบโครงสร้างเครือข่าย ไฟล์เทเบิลของสวิตช์แต่ละเครื่อง และข้อมูลอื่นๆ ของเครือข่ายเอกสารนี้ได้โดยใช้อปเพลิกเช่นไฟล์เมเนจอร์ (Flow Manager) ดังแสดงในรูปที่ 3.5 หลังจากที่เข้มต่อ มินิเน็ตกับวิวอสตีอีนคอนโทรลเลอร์เรียบร้อยแล้ว

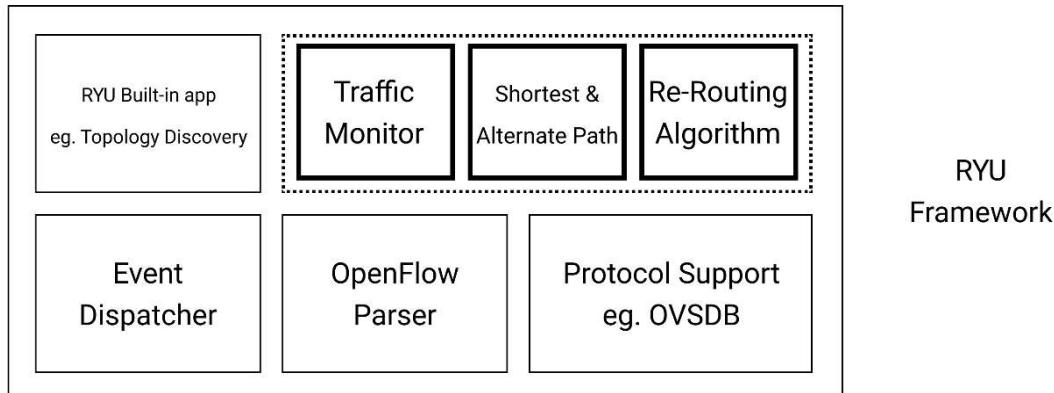
### 3.1.2 วิวอสตีอีนคอนโทรลเลอร์

วิวอสตีอีนคอนโทรลเลอร์ (RYU SDN Controller) เป็นส่วนสมองของเครือข่ายที่กำหนดโดยซอฟต์แวร์ เป็นตัวกลางการสื่อสารระหว่างชั้นแอปพลิเคชันและชั้นโครงสร้างพื้นฐาน เครือข่าย ภายใต้วิวอสตีอีนคอนโทรลเลอร์จะประกอบไปด้วยโมดูลพื้นฐานสำหรับการทำงานร่วมกับโอเพนไฟล์โปรโตคอล (OpenFlow Protocol) เช่น RYU Built-in app, Event Dispatcher, OpenFlow Parser, Protocol Support ดังแสดงในรูปที่ 3.5



รูปที่ 3.6 RYU SDN Framework มาตรฐาน

จากรูปที่ 3.6 ในส่วนของวิวแอปพลิเคชัน (RYU Applications) จะเป็นโมดูลที่นักพัฒนาต้องเขียนแอปพลิเคชันด้านในเพื่อตอบสนองโจทย์ ความต้องการ หรือการทำงานรูปแบบต่างๆ ด้วยตนเอง โดยงานของเราได้สร้างโมดูลไว้ภายในวิวแอปพลิเคชันทั้งหมด 3 โมดูล ซึ่งจะอธิบายเพิ่มเติมต่อจากนี้

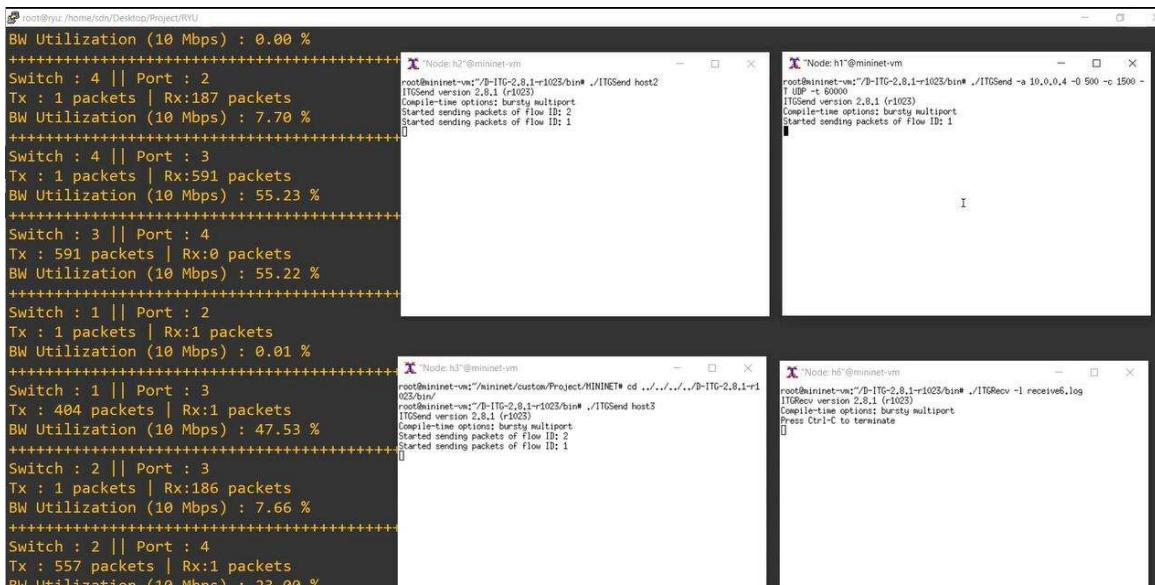


รูปที่ 3.7 RYU SDN Framework ที่ประยุกต์ใช้ในโครงงาน

โครงงานของเราราได้ติดตั้งริวคอนโทรลเลอร์ให้ทำงานบนระบบปฏิบัติการอุบัณฑูช์นเดียวกับมินิเน็ต เราได้สร้างริวแอปพลิเคชันสำหรับการจัดการเครือข่ายรวม 3 โมดูลได้แก่ โมดูลสังเกตุการณ์ข้อมูลในเครือข่าย (Traffic Monitor) โมดูลการหาเส้นทางที่สั้นที่สุด และเส้นทางทางเลือก (Shortest Path & Alternate Path Finder) และโมดูลการเลือกเส้นทางใหม่ (Re-Routing Module) ดังแสดงในรูปที่ 3.7

### 3.1.2.1 โมดูลสังเกตการณ์ข้อมูลในเครือข่าย

โมดูลสังเกตการณ์ข้อมูลในเครือข่าย (Traffic Monitor Module) เป็นโมดูลสำหรับการเก็บรวบรวมข้อมูลในเครือข่ายจากอุปกรณ์โอเพนไฟล์สวิตช์ คำนวนค่าการใช้ประโยชน์แบบดิจิตที่ และนำมาแสดงผลเพื่อใช้ในเคราะห์สถานการณ์ของลิงค์แบบเรียลไทม์ ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 โมดูลสังเกตการณ์рафฟิกแสดงข้อมูลของลิงค์ในแต่ละวินาที

ข้อมูลที่โมดูลสังเกตการณ์ข้อมูลในเครือข่ายจะประกอบไปด้วยหมายเลขสวิตช์ (Switch) หมายเลขพอร์ตของสวิตช์ (Port) เวลาที่เก็บข้อมูล (Time) จำนวนแพ็คเก็ตที่ถูกส่งออกจากพอร์ต (Tx) จำนวนแพ็คเก็ตที่รับเข้ามาทางพอร์ต (Rx) และค่าการใช้ประโยชน์แบบดิจิตของลิงค์นั้น (BW\_Utillization)

### 3.1.2.2 การคำนวนค่าการใช้ประโยชน์แบบดิจิตที่

เราได้ใช้ไฟล์ด้วยข้อมูลในแพ็คเก็ตชนิด OFPPortStatsReply ของโอเพนไฟล์ไฟร์วอลล์เพื่อเป็นแหล่งข้อมูลที่สำคัญ ที่นำไปสู่การคำนวนค่าการใช้ประโยชน์แบบดิจิตที่

OFPPortStatsReply เป็นแพ็คเก็ตที่โอเพนไฟล์สวิตช์ใช้ตอบกลับมาเมื่อรับคำขอของไฟร์วอลล์ที่ต้องการข้อมูลสถิติของอุปกรณ์ จำนวนแพ็คเก็ตที่รับ-ส่ง (rx\_packets, tx\_packets) จำนวนไบต์ที่รับ-ส่ง (rx\_bytes, tx\_bytes) จำนวนแพ็คเก็ตที่ถูก丢弃ที่ไป

(rx\_dropped, tx\_dropped) โดยเราได้นำไฟล์ Tx\_Bytes และ Rx\_Bytes ในแพ็คเก็ต OFPPortStatsReply มาใช้คำนวณค่าการใช้ประโยชน์แบบดิจิตที่ด้วยสูตรคำนวณดังนี้

$$Bw\_Utilization = \frac{(Tx\ Bytes + Rx\ Bytes)}{1,310,720\ Bytes}$$

ค่า 1,310,720 Bytes มีค่าเท่ากับ 10 Megabits ซึ่งตรงกับค่าแบบดิจิตที่ของลิงค์ที่เรากำหนดผ่านซอฟต์แวร์มินิเน็ต ค่าการใช้ประโยชน์แบบดิจิตที่ หรือ BW\_Utilization ที่คำนวณออกมากจะมีค่าอยู่ในช่วง [0,1] โดยที่ 1 หมายถึงค่าการใช้ประโยชน์แบบดิจิตที่ 100% หลังจากที่คำนวณค่าการใช้ประโยชน์แบบดิจิตที่ของลิงค์ ณ เวลาต่างๆ ก็จะนำมาประกอบเป็นตารางสำหรับนำไปใช้เป็นชุดข้อมูล ดังแสดงในตารางที่ 3.1

Timestamp	Tx_Packets	Rx_Packets	BW_Utilization
2020-10-27 11:47:18	1381	579	0.39374526977539065
2020-10-27 11:47:19	1241	549	0.3672716522216797
2020-10-27 11:47:20	1865	949	0.5010334777832032
2020-10-27 11:47:21	2880	1596	0.6822846984863281
2020-10-27 11:47:22	3132	1728	0.7208555603027343
2020-10-27 11:47:23	2872	1849	0.7225883483886719
2020-10-27 11:47:24	3087	1662	0.7452517700195312
2020-10-27 11:47:25	3076	1869	0.7375452423095703
2020-10-27 11:47:26	3070	1677	0.7493332672119141
2020-10-27 11:47:27	2887	1874	0.6503284454345704

ตารางที่ 3.1 ตัวอย่าง ชุดข้อมูลที่ได้ดัดแปลงมาจากการ OFPPortStatsReply

ชุดข้อมูลที่ได้ดังตารางที่ 3.1 นี้จะนำไปใช้สำหรับให้โมเดลการเรียนรู้เชิงลึก LSTM เพื่อทำการเรียนรู้ และใช้เป็นอินพุตสำหรับแอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบบดิจิตที่ LSTM เพื่อคาดการณ์ค่าการใช้ประโยชน์แบบดิจิตที่แบบเรียลไทม์

### 3.1.2.3 โมดูลการหาเส้นทางที่สั้นที่สุด และเส้นทางทางเลือก

โมดูลการหาเส้นทางที่สั้นที่สุดและเส้นทางทางเลือก (Shortest Path Algorithm and Alternate Path Finder) เป็นโมดูลสำหรับการค้นหาเส้นทางที่สั้นที่สุดระหว่างโหนดต้นทางไปยังโหนดปลายทาง และเส้นทางที่เป็นไปได้ทั้งหมดระหว่างสองโหนดเพื่อเลือกใช้เป็นเส้นทางสำรองในการเลือกเส้นทางใหม่

### 3.1.2.4 โมดูลการเลือกเส้นทางใหม่

โมดูลการเลือกเส้นทางใหม่ (Re-Routing Module) จะรับค่าการใช้ประโยชน์แบบดิจิตที่ถูกคาดการณ์จากแอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบบดิจิตที่ LSTM มาพิจารณาว่า หากได้ค่าที่มากเกินกว่าค่าเทอร์ชิลอดด์ (Threshold) ที่กำหนดไว้ เอสเดี้ยนคอนโทรลเลอร์ก็จะสั่งการให้อุปกรณ์โอเพนไฟล์สวิตซ์เปลี่ยนเส้นทางการส่งข้อมูล เพื่อบังกันการเกินความคับคั่งของข้อมูล

เมื่ออัลกอริทึมการเลือกเส้นทางใหม่ทำงาน เราจะพบข้อความบนหน้าจอเทอร์มินัล (Terminal) ของอสเดี้ยนคอนโทรลเลอร์เกี่ยวกับการเลือกเส้นทางใหม่สำหรับการส่งข้อมูลระหว่างสวิตซ์ดังแสดงในรูปที่ 3.9 รวมไปถึงค่าแบบดิจิตที่ตรวจจับ และค่าที่คาดการณ์ว่าจะสูงกว่าเทอร์ชิลอดด์ที่กำหนดดังแสดงในรูปที่ 3.10

```
Re-Routing Process :
Banned Link Between Switch : 3 and Switch : 6
rerouting_effect :
{'1': {'1->6': [[1, 3, 6]]}, '2': {'2->6': [[2, 3, 6]]}, '3': {'3->4': [[3, 6, 4]], '3->6': [[3, 6]]}, '4': {}, '6': {}}
1->2 [1, 2]
1->3 [1, 3]
1->4 [1, 2, 5, 4]
1->5 [1, 2, 5]
1->6 [1, 3, 6]
2->1 [2, 1]
2->3 [2, 3]
2->4 [2, 5, 4]
2->5 [2, 5]
2->6 [2, 3, 6]
3->1 [3, 1]
3->2 [3, 2]
3->4 [3, 6, 4]
3->5 [3, 2, 5]
3->6 [3, 2, 5, 6]
4->1 [4, 5, 3, 1]
```

รูปที่ 3.9 การทำงานของอัลกอริทึมการเลือกเส้นทางใหม่

```

3->5 [3, 2, 5]
3->6 [3, 2, 5, 6]
4->1 [4, 5, 2, 1]
4->2 [4, 5, 2]
4->3 [4, 6, 3]
4->5 [4, 5]
4->6 [4, 6]
5->1 [5, 2, 1]
5->2 [5, 2]
5->3 [5, 2, 3]
5->4 [5, 4]
5->6 [5, 6]
6->1 [6, 3, 1]
6->2 [6, 3, 2]
6->3 [6, 3]
6->4 [6, 4]
6->5 [6, 5]
Re-Routing Success ! !
#####
##### Link between switch 3 - switch 6
      Detected Bandwidth Utilization : 0.86 %
      Predicted Bandwidth Utilization : 0.90 %

#####
Switch : 1 || Port : 2
Tx : 275 packets | Rx:219 packets
BW Utilization (10 Mbps) : 29.15 %

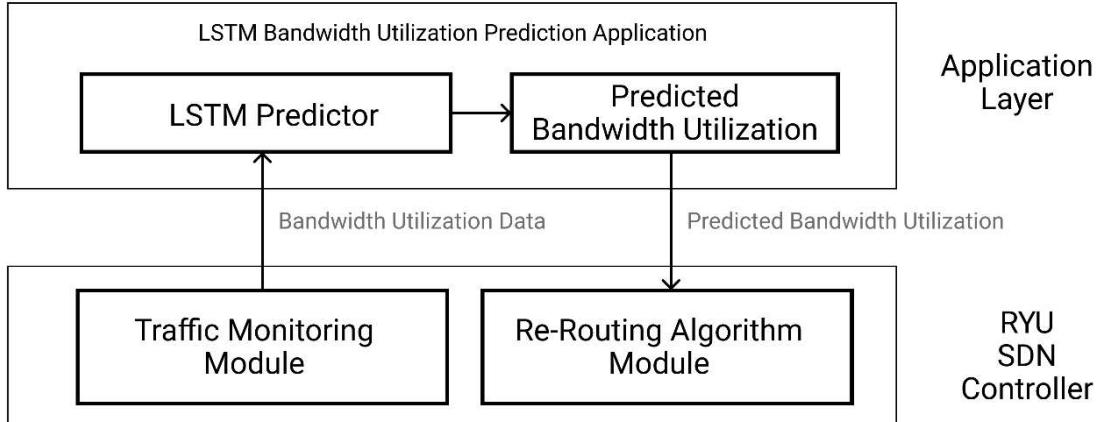
```

รูปที่ 3.10 ค่าการใช้ประโยชน์แบบดั่วิดท์ที่ตรวจสอบและคาดการณ์ว่าจะเกินทรัพยากรสื่อสาร

เราได้กำหนดให้อัลกอริทึมการจัดเส้นทางใหม่เมื่อเดือนทำงานอยู่ที่ 30 วินาที กล่าวคือหลังจากที่ริบคอนโทรลเลอร์สั่งการให้เปลี่ยนเส้นทางแล้ว จะไม่มีการคาดการณ์ค่าการใช้ประโยชน์แบบดั่วิดท์และการทำงานของอัลกอริทึมการจัดเส้นทางใหม่เป็นเวลา 30 วินาที ทั้งนี้เพื่อป้องกันไม่ให้เกิดการเปลี่ยนแปลงเส้นทางการส่งข้อมูลอย่างต่อเนื่อง

### 3.1.3 แอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบบดั่วิดท์ LSTM

แอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบบดั่วิดท์ LSTM (LSTM Bandwidth Utilization Prediction Application) เป็นส่วนประกอบของชั้นแอปพลิเคชัน จะทำการคาดการณ์ค่าการใช้ประโยชน์แบบดั่วิดท์หลังจากที่รับอินพุตค่าการใช้ประโยชน์แบบดั่วิดท์แบบเรียลไทม์มาจากการริบอฟฟ์เวิร์ดและจัดส่งค่าที่คาดการณ์กลับไปยังโ้มคุลการเลือกเส้นทางใหม่ที่อยู่ภายใต้การริบอฟฟ์เวิร์ดในรูปที่ 3.11

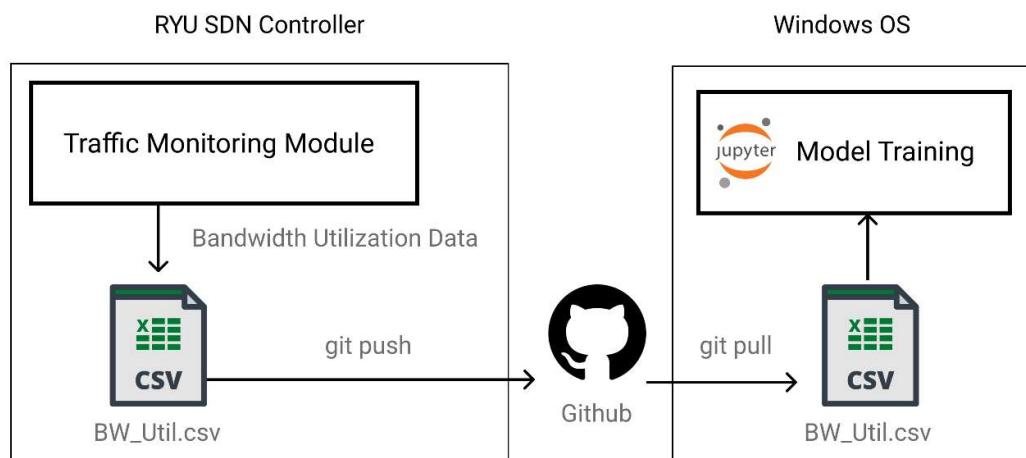


รูปที่ 3.11 ภาพรวมของแอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบบด้วยที่ LSTM

**3.1.4 ขั้นตอนการสร้างแอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบบด้วยที่ LSTM**  
 โดยเดลารีเยนรุ่งเชิงลีกประเภท Long-short Term Memory (LSTM) ลูกเขียงขึ้นโดยใช้แพลตฟอร์มจูปิเตอร์ (Jupyter Notebook) บนระบบปฏิบัติการวินโดว์ (Windows)

#### 3.1.4.1 การรวบรวมข้อมูล

เราเริ่มจากการรวบรวมข้อมูล (Data Gathering) สำหรับนำไปให้โมเดล LSTM เรียนรู้ โดยนำเอาข้อมูลที่ไม่ถูกสังเกตการณ์ทรายฟิกแสดงผลออกมานำไปสร้างเป็นไฟล์ชนิด .csv เก็บเอาไว้ในรีโพสติชัน GitHub และนำออกไฟล์มาสู่ระบบปฏิบัติการวินโดว์โดยการใช้ Git ดังแสดงในรูปที่ 3.12



รูปที่ 3.12 การส่งข้อมูลค่าการใช้ประโยชน์แบบด้วยที่วิเคราะห์อัตโนมัติจาก GitHub

### 3.1.4.2 กระบวนการเตรียมข้อมูล

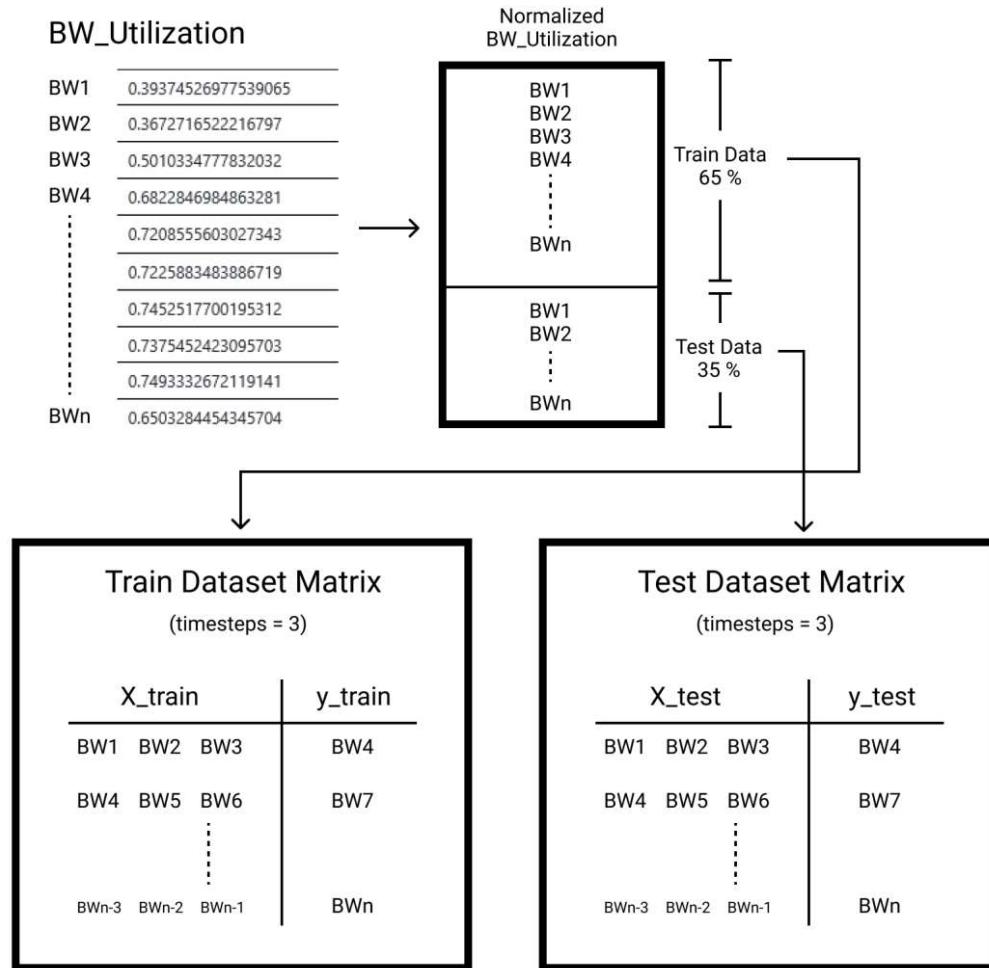
หลังจากที่ได้ข้อมูลค่าการใช้ประโยชน์แบบดิจิทัลมาอยู่ในแพลตฟอร์มจูปิเตอร์ สิ่งถัดมาที่ต้องทำคือการเตรียมข้อมูล (Data Pre-Processing) เพื่อให้ข้อมูลอยู่ในรูปแบบที่เหมาะสมกับการเรียนรู้ของการเรียนรู้เชิงลึก กระบวนการที่จำเป็นต้องใช้คือ Normalization ซึ่งเป็นกระบวนการแปลงข้อมูลตัวเลขให้มีค่าอยู่ในช่วงระหว่าง 0-1 ซึ่งจะช่วยให้โมเดลการเรียนรู้เชิงลึก LSTM เรียนรู้ได้ดี

### 3.1.4.3 Train-Test Split

ชุดข้อมูลจะถูกแบ่งออกเป็น 2 ส่วน ได้แก่ ข้อมูลสอน(Training Data) และชุดทดสอบ (Testing Data) โดยแบ่งออกเป็นสัดส่วน ข้อมูลสอน 65% และ ชุดทดสอบ 35% หลังจากสร้างโมเดลการเรียนรู้ด้วยข้อมูลสอนเป็นที่เรียบร้อย จะทำการทดสอบด้วยข้อมูลชุดทดสอบว่าโมเดลของเรา มีความแม่นยำสูงเพียงใด โดยทดสอบโดยยกเว้นชุดทดสอบ ไม่รวมกับชุดทดสอบ

### 3.1.4.4 Dataset Matrix

เมตริกซ์ (Matrix) ในที่นี้หมายถึงรูปแบบอินพุตสุดท้ายสำหรับนำไปใช้ฝึกสอนโมเดลการเรียนรู้เชิงลึก LSTM ชุดข้อมูลที่ถูกแบ่งออกเป็น Train Data และ Test Data จะถูกแปลงเป็น Train Dataset Matrix และ Test Dataset Matrix ดังแสดงในรูปที่ 3.13



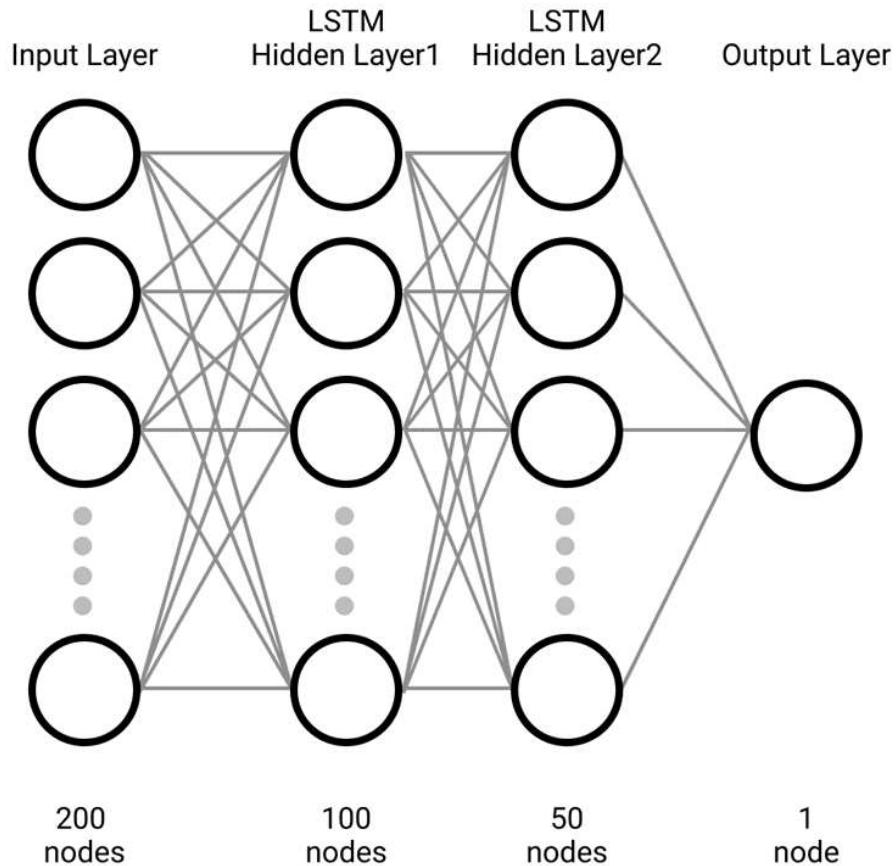
รูปที่ 3.13 สรุปกระบวนการสร้าง Dataset Matrix เริ่มตั้งแต่นำเข้าข้อมูล

Timesteps เป็นตัวแปรที่กำหนดถึงความกว้างของ x\_train และจำนวนโหนด อินพุตของโครงข่าย LSTM เช่นหากกำหนดให้ timesteps = 3, X\_train ของ Dataset Matrix ที่ได้จะมี จำนวนคอลัมน์เป็น 3 คอลัมน์ ส่วนค่าของ y\_train จะมีขนาดเป็น 1 คอลัมน์เสมอ และอินพุตของ LSTM จะมีจำนวนเท่ากับ 3 โหนด สำหรับโครงงานนี้ได้ทำการกำหนดค่า timesteps = 200 ซึ่งส่งผลให้ ชั้นอินพุตของ LSTM มีจำนวน 200 โหนด

#### 3.1.4.5 โครงสร้างโมเดล LSTM

หลังจากที่จัดเตรียมข้อมูลสำหรับการฝึกสอนแล้ว ขั้นตอนถัดมาคือการสร้าง โมเดลการเรียนรู้เชิงลึกประภาค Long-Short Term Memory และปรับจุนพารามิเตอร์ต่างๆ ก่อนที่จะเริ่ม การเรียนรู้ โมเดล LSTM ที่ใช้ในโครงงานประกอบด้วยชั้นอินพุตจำนวน 200 โหนด เชื่อมต่อด้วยชั้น

ชั้อนตัว (Hidden Layer) ชั้น 2 ชั้น ชั้นละ 100 โหนด และ 50 โหนดตามลำดับ และต่อไปยังชั้නเอาท์พุต 1 โหนดดังแสดงในรูปที่ 3.14



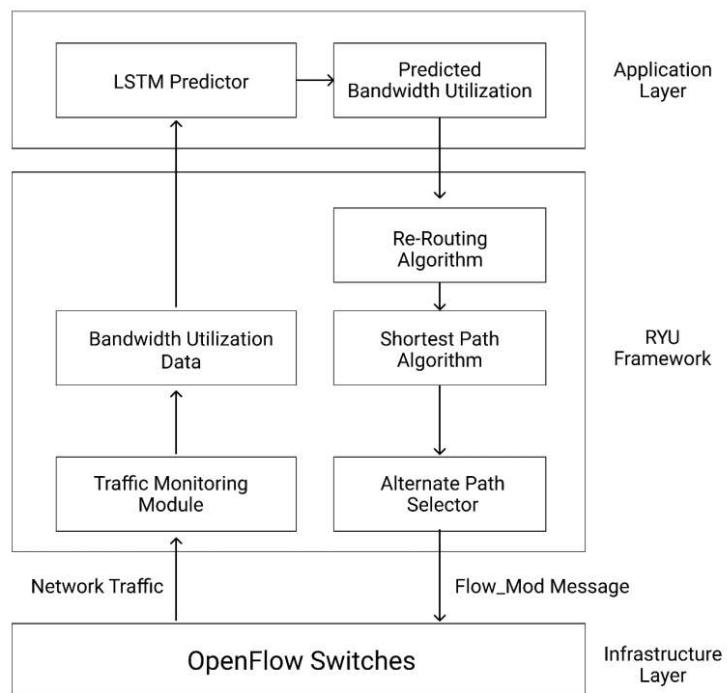
รูปที่ 3.14 โครงสร้างของโครงข่ายประสาทเทียมชนิด LSTM ในโครงงาน

ค่าการใช้ประโยชน์แบบด์วิดท์จะเป็นเอาท์พุต (output) ที่โมเดลทำการคาดการณ์ออกมา และจะถูกนำไปเปรียบเทียบกับ Label หรือค่าการใช้ประโยชน์แบบด์วิดท์ในชุดข้อมูลจริง เพื่อวัดประสิทธิภาพโมเดล ว่าสามารถคาดการณ์ค่าในอนาคตออกมายได้แม่นยำเพียงใด

หลังจากที่ได้โมเดล LSTM ที่ผ่านการเรียนรู้ และปรับจูนให้ทำงานได้อย่างมีประสิทธิภาพ มีความแม่นยำแล้ว จึงจะนำไปทำงานบนริเวอสตีเอ็นคอน โทรลเลอร์เพื่อคาดการณ์ค่าการใช้ประโยชน์แบบด์วิดท์แบบเรียลไทม์

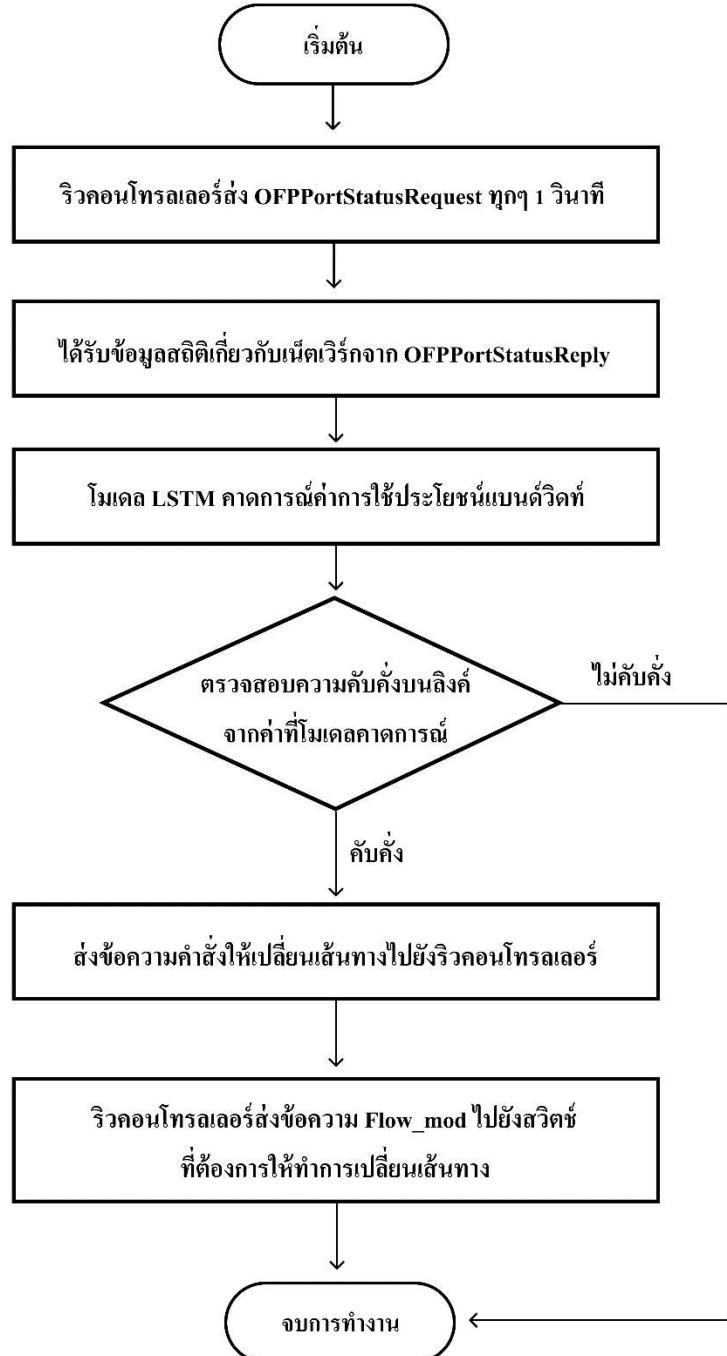
### 3.2 ขั้นตอนการทำงานของระบบ

กระบวนการทำงานเริ่มต้นจากวิวอีอีนคอน โทรลเลอร์ ทำการเก็บรวบรวมข้อมูลในเครือข่าย (Network Traffic) จากโครงสร้างพื้นฐานเครือข่าย (Network Infrastructure) อย่างต่อเนื่อง และส่งไปยังชั้นแอปพลิเคชันที่ประกอบด้วย โมเดลการเรียนรู้เชิงลึก LSTM ซึ่งทำหน้าที่คาดการณ์ค่าการใช้ประโยชน์แบบดิวท์ (Bandwidth Utilization) จากนั้น โมเดลจะส่งค่าที่คาดการณ์กลับมายังวิวอีอีนคอน โทรลเลอร์ เพื่อวิเคราะห์ถึงความเป็นไปได้ที่จะเกิดความคับคั่งบนลิงค์ (Link) และส่งข้อความคำสั่ง (Flow\_mod Message) ให้สิวิตช์ทำการเปลี่ยนเส้นทางการส่งข้อมูล หากคาดว่ามีลิงค์ที่จะเกิดความคับคั่งของข้อมูล ภาพรวมขั้นตอนการทำงานของระบบเป็นดังแสดงในรูปที่ 3.2



รูปที่ 3.15 Block Diagram แสดงภาพรวมขั้นตอนการทำงานของระบบ

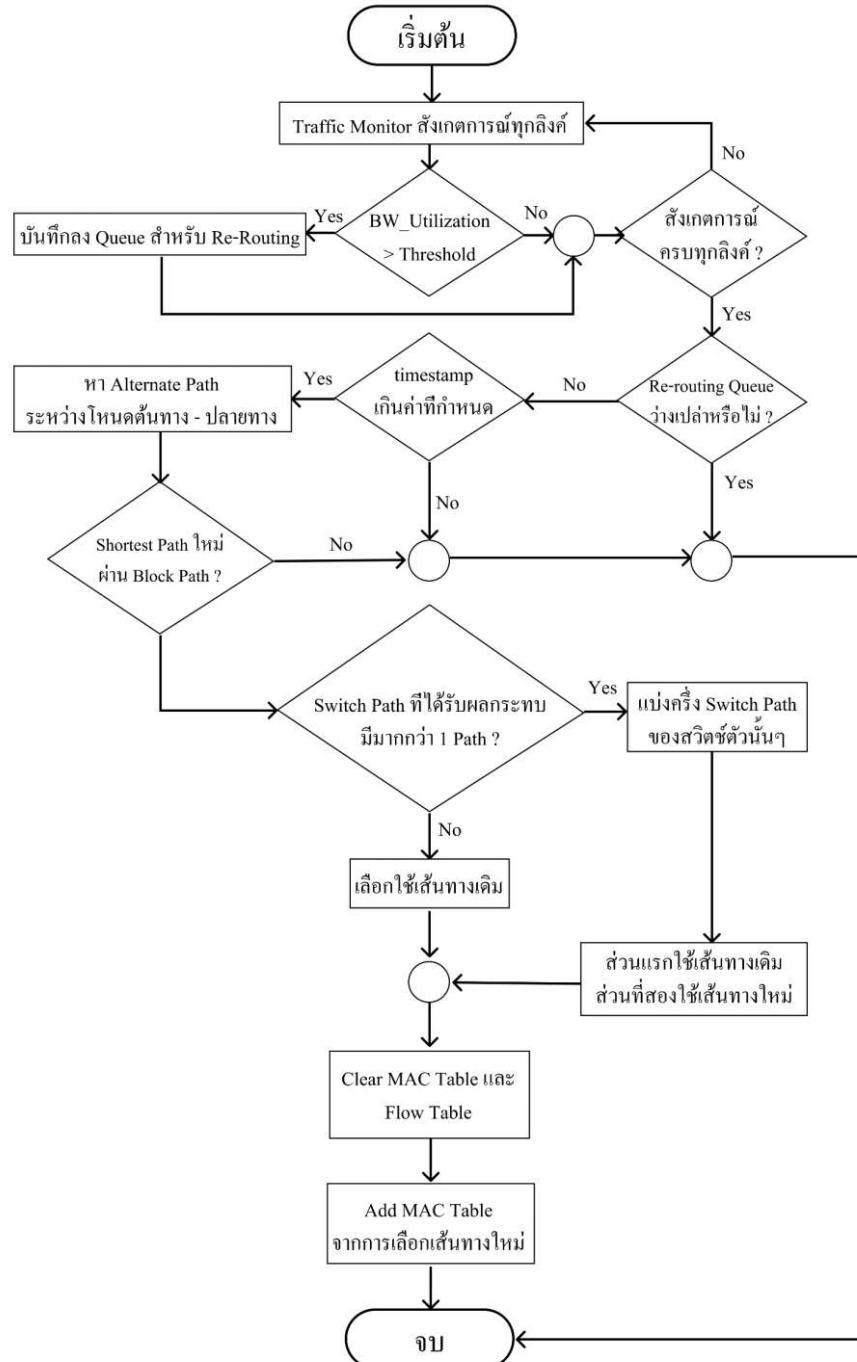
เราสามารถ ไฟล์วิวการทำงานของระบบทั้งหมดผ่านไฟล์ชาร์ท ดังแสดงในรูปที่ 3.16



รูปที่ 3.16 Flow Chart แสดงขั้นตอนการทำงานโดยรวมของโครงการ

### 3.2.1 ขั้นตอนการทำงานของอัลกอริทึมการเลือกเส้นทางใหม่

โมดูลการเลือกเส้นทางใหม่ (Re-Routing Module) มีลำดับการทำงานดังแสดงในรูปแบบของโฟลว์ชาร์ทในรูปที่ 3.17



รูปที่ 3.17 Flowchart ของ Re-Routing Algorithm

### 3.2.1.1 ตรวจสอบ Re-routing Queue

การทำงานของโมดูลการเลือกเส้นทางใหม่ (Re-Routing Module) จะเริ่มขึ้น เมื่อ Re-routing Queue ไม่ได้มีค่าที่ว่างเปล่า เราจะทำการเก็บบันทึกลิงค์ที่ถูกคาดการณ์ว่าจะมีค่าการใช้ประโยชน์แบบดัชนีเพื่อกินกว่าค่าเทอร์โบลด์ เพื่อใช้ดำเนินการต่อไป

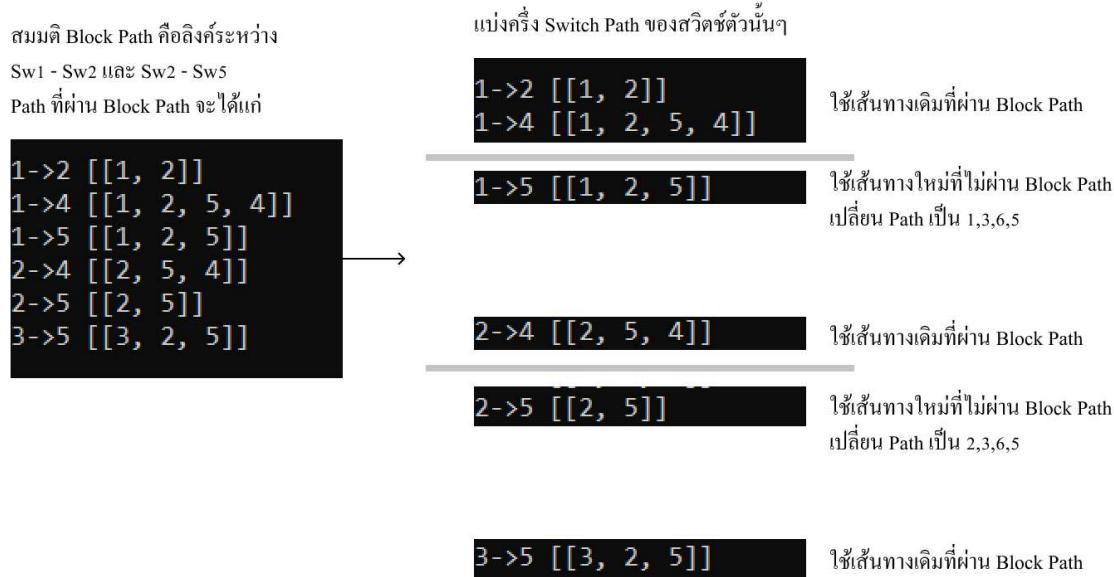
แต่ก่อนที่จะดำเนินการเปลี่ยนเส้นทางส่งข้อมูลต่อไป จะมีการตรวจสอบก่อนว่า timestamp เกินค่าที่กำหนดหรือยังเพื่อไม่ให้มีการเปลี่ยนเส้นทางแบบเกิดขึ้นฉันด้วยในงานของเรา กำหนด timestamp ไว้ที่ 30 วินาที ซึ่งหมายความว่าระบบจะไม่มีการทำการเปลี่ยนเส้นทางอีกรัง จนกว่าจะเลยเวลา 30 วินาที

### 3.2.1.2 ค้นหาและเลือกเส้นทางใหม่

โมดูลการหาเส้นทางที่สั้นที่สุด และเส้นทางทางเลือกจะค้นหาและเก็บข้อมูลเส้นทางทั้งหมดระหว่างโสมท์แต่ละตัวเอาไว้ตั้งแต่เริ่มต้นจำลองเครือข่าย ระบบจะทำการค้นหาดูว่า หากจะยกเลิกการใช้งานลิงค์ๆหนึ่ง จำเป็นต้องให้สวิตช์ตัวใดทำการเปลี่ยนเส้นทางการส่งข้อมูลบ้าง ซึ่งเส้นทางที่จะยกเลิกไม่ให้ใช้งานจะถูกเรียกว่าเป็นเส้นทางที่ถูกสกัดกั้น (Block Path)

### 3.2.1.3 Block Path และ Switch Path

Block Path หมายถึงลิงค์ที่ห้ามไม่ให้ข้อมูลในเครือข่ายวิ่งผ่าน เนื่องจากถูกคาดการณ์ว่าค่าการใช้ประโยชน์แบบดัชนีของลิงค์ดังกล่าวจะต้องเปลี่ยนเส้นทาง โดยการ Re-Routing ของสวิตช์นั่นเอง ส่วน Switch Path หมายถึงเส้นทางการส่งข้อมูลของสวิตช์ที่ได้รับผลกระทบจาก Block Path ดังอธิบายในรูปที่ 3.18



รูปที่ 3.18 Block Path และ Switch Path

หลังจากทำการกำหนด Block Path เป็นที่เรียบร้อย จึงจะทำการออกแบบเพื่อทำการเปลี่ยนเส้นทาง ดังแสดงในรูปที่ 3.19

Before Re-Routing	After Re-Routing
1->2 [1, 2]	1->2 [1, 2]
1->3 [1, 3]	1->3 [1, 3]
1->4 [1, 2, 5, 4]	1->4 [1, 2, 5, 4]
1->5 [1, 2, 5]	1->5 [1, 3, 6, 5]
1->6 [1, 3, 6]	1->6 [1, 3, 6]
5->1 [5, 2, 1]	5->1 [5, 6, 3, 1]
5->2 [5, 2]	5->2 [5, 6, 3, 2]
5->3 [5, 2, 3]	5->3 [5, 2, 3]
5->4 [5, 4]	5->4 [5, 4]
5->6 [5, 6]	5->6 [5, 6]

รูปที่ 3.19 ผลที่เกิดขึ้นหลังจากออกแบบการเปลี่ยนเส้นทาง

สำหรับกรณี Switch Path ของสวิตช์ตัวหนึ่งมีแค่เพียง 1 เส้นทาง อัลกอริทึมจะทำการรวมกลุ่ม Switch Path ของสวิตช์ทุกตัวที่มีเพียง 1 เส้นทางเข่นเดียวกัน และแบ่งครึ่งกลุ่มของ Switch Path นี้อีกครึ่งดังที่แสดงในรูป 3.18 เป็นอันเสร็จสิ้นกระบวนการเลือกเส้นทางใหม่

## บทที่ 4

### ผลการทดลอง

ในบทนี้ เราได้ทำการนำเสนอกระบวนการทดลอง และผลการทดลอง โดยเริ่มจากการเปรียบเทียบอัลกอริทึมการเรียนรู้หลายชนิด นำมาคาดการณ์ค่าการใช้ประโยชน์แบบดิจิต์ และเปรียบเทียบความแม่นยำในการคาดการณ์ เพื่อเลือกใช้เป็นอัลกอริทึมที่จะนำไปใช้ในโครงงาน จากนั้น เราได้ทำการทดลองเพื่อเลือกค่าเทرزโซลด์ที่เหมาะสมกับการใช้งาน และตามด้วยนำเสนอภาพของ การวัดค่าทรูพุต เพื่อกับปัจมันทรافฟิกโดยเฉลี่ยที่เข้ามาในเครือข่าย

#### 4.1 การพิจารณาอัลกอริทึมการเรียนรู้ที่เหมาะสม และการประเมินประสิทธิภาพของ อัลกอริทึมการเรียนรู้

โครงงานนี้ ได้ทดลองเลือกใช้อัลกอริทึมการเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึกหลายชนิดเข้ามาทำการคาดการณ์ค่าการใช้ประโยชน์แบบดิจิต์ เพื่อวิเคราะห์และพิจารณาว่าจะเลือกใช้อัลกอริทึมที่เหมาะสมสำหรับการทดลอง

เราได้เลือกใช้อัลกอริทึมทั้งหมด 5 ชนิดมาเปรียบเทียบความแม่นยำ ได้แก่ Regression, Decision Tree Regressor, Random Forest, Multi-Layer Perceptron และ Long-Short Term Memory โดยได้เลือกค่า Root Mean Squared Error (RMSE) และ Mean Absolute Error (MAE) เป็นค่าเมตริกซ์เพื่อเปรียบเทียบความแม่นยำของอัลกอริทึม

##### 4.1.1 Root Mean Squared Error และ Mean Absolute Error

Root Mean Squared Error และ Mean Absolute Error เป็นเมตริกซ์ที่นิยมใช้สำหรับวัดความถูกต้องของโมเดลที่ทำนายค่าอุปกรณ์เป็นค่าตัวเลข (Numerical Value) คำนวณโดยสูตร

$$RMSE = \sqrt{\frac{1}{n} * \sum_{i=1}^n (\hat{Y}_i - Y_i)^2} \quad (1)$$

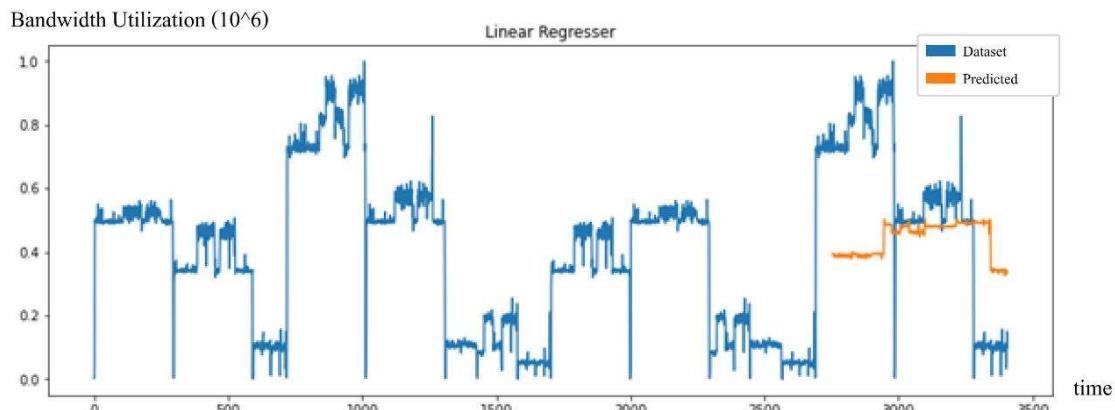
$$MAE = \frac{\sum_{i=1}^n |\hat{Y}_i - Y_i|}{n} \quad (2)$$

$\hat{Y}_i - Y_i$  แสดงถึงการคำนวณความผิดพลาด (Error) หรือ Loss ของการทำนายค่าโดยที่  $\hat{Y}_i$  หมายถึงค่าที่โมเดลคาดการณ์ (Predict) และ  $Y_i$  หมายถึงค่าจริง (Actual)

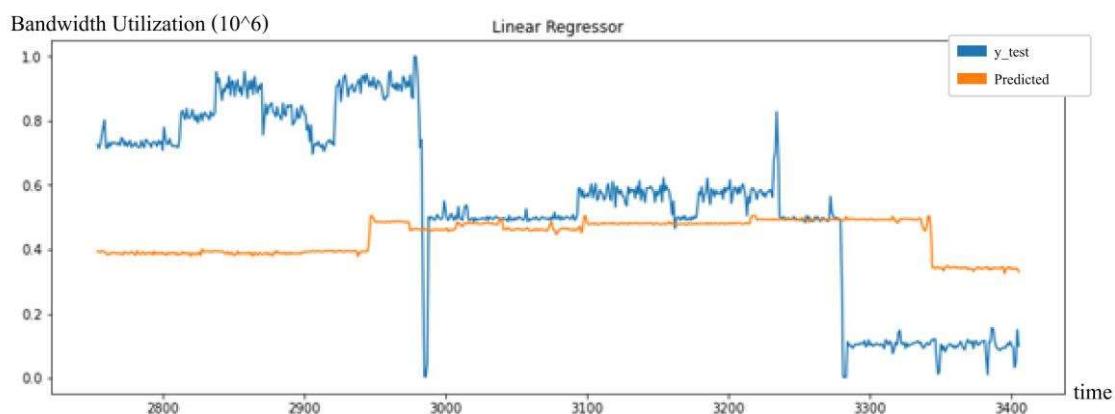
ค่า Root Mean Squared Error (RMSE) และ Mean Absolute Error (MAE) วัดได้ดังแสดงในตารางที่ 4.1 แนวโน้มของการคาดการณ์ค่าของอัลกอริทึม Linear Regression, Decision Tree Regressor, Random Forest, Multi-Layer Perceptron และ Long-Short Term Memory เป็นไปตามกราฟดังแสดงในรูปที่ 4.1 4.2 4.3 4.4 และ 4.5 ตามลำดับ

อัลกอริทึมการเรียนรู้	Root Mean Squared Error	Mean Absolute Error
Linear Regression	0.2778	0.2334
Decision Tree Regressor	0.3486	0.2823
Random Forest	0.3389	0.2723
Multi-Layer Perceptron	0.2741	0.2293
Long-Short Term Memory	0.0654	0.0318

ตารางที่ 4.1 การเปรียบเทียบความแม่นยำของอัลกอริทึมการเรียนรู้

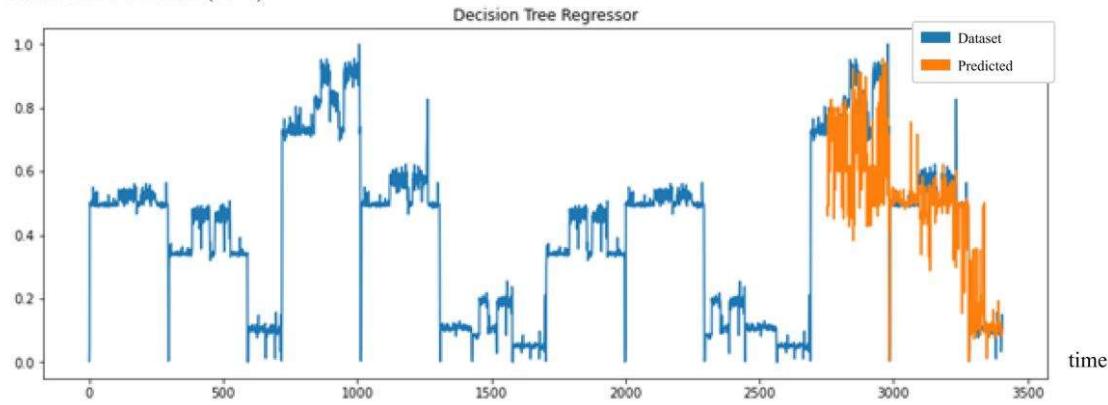


(a) กราฟเปรียบเทียบค่าชุดข้อมูลทั้งหมด กับค่าที่ Linear Regressor คาดการณ์

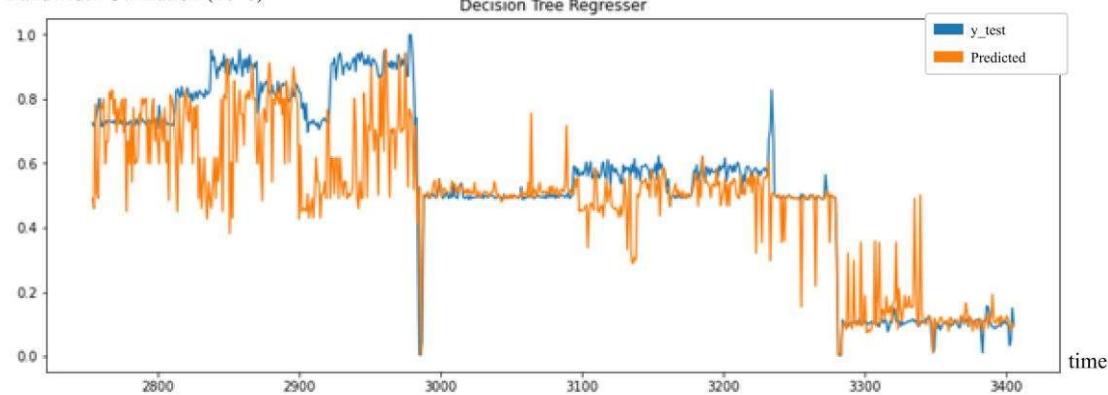


(b) กราฟเปรียบเทียบค่าชุดข้อมูลเฉพาะส่วน  $y_{\text{test}}$  กับค่าที่ Linear Regressor คาดการณ์

รูปที่ 4.1 ผลการวัดความแม่นยำของ Linear Regression

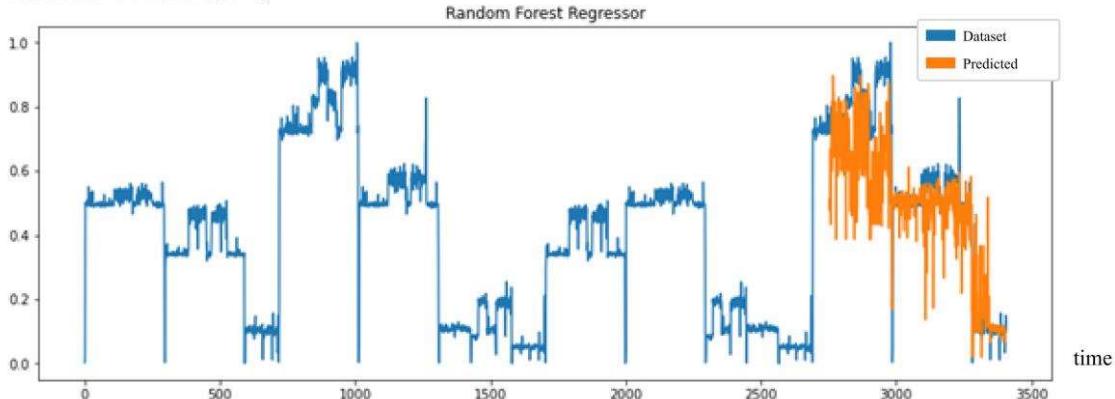
Bandwidth Utilization ( $10^6$ )

(a) กราฟเปรียบเทียบค่าชุดข้อมูลทั้งหมด กับค่าที่ Decision Tree Regressor คาดการณ์

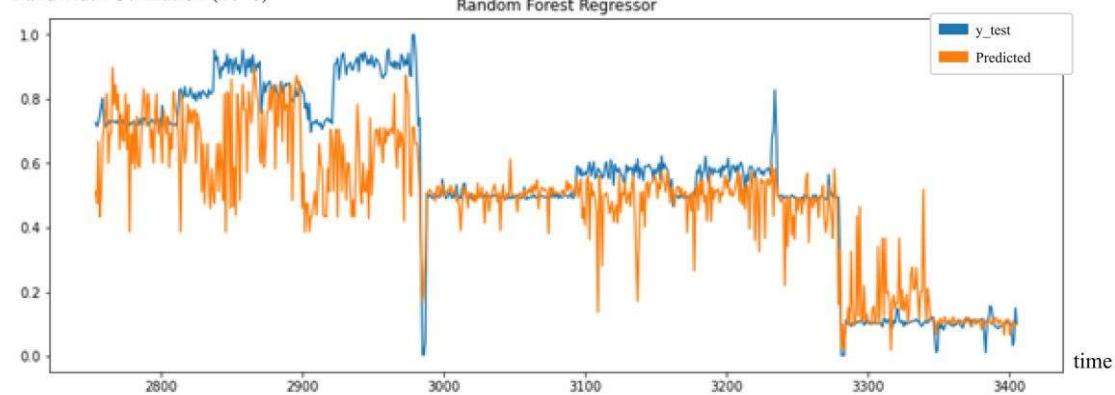
Bandwidth Utilization ( $10^6$ )

(b) กราฟเปรียบเทียบค่าชุดข้อมูลเฉพาะส่วน y\_test กับค่าที่ Decision Tree Regressor คาดการณ์

#### รูปที่ 4.2 ผลการวัดความแม่นยำของ Decision Tree Regressor

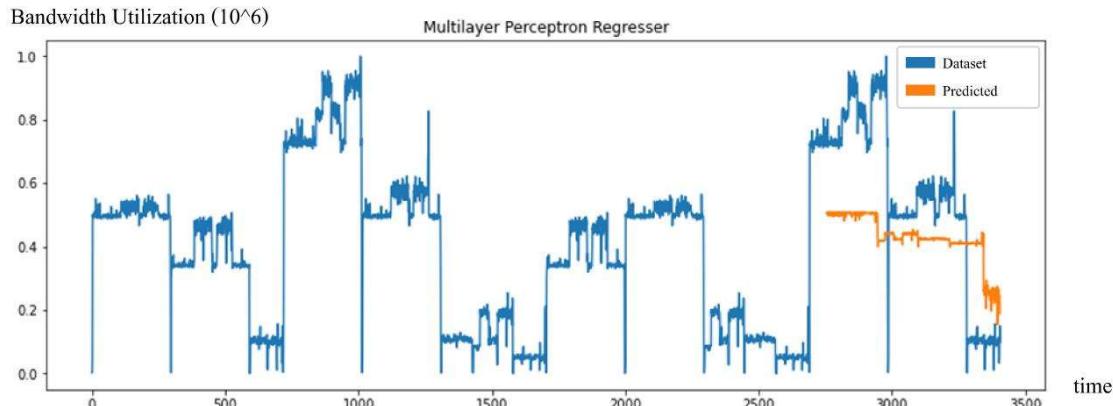
Bandwidth Utilization ( $10^6$ )

(a) กราฟเปรียบเทียบค่าชุดข้อมูลทั้งหมด กับค่าที่ Random Forest Regressor คาดการณ์

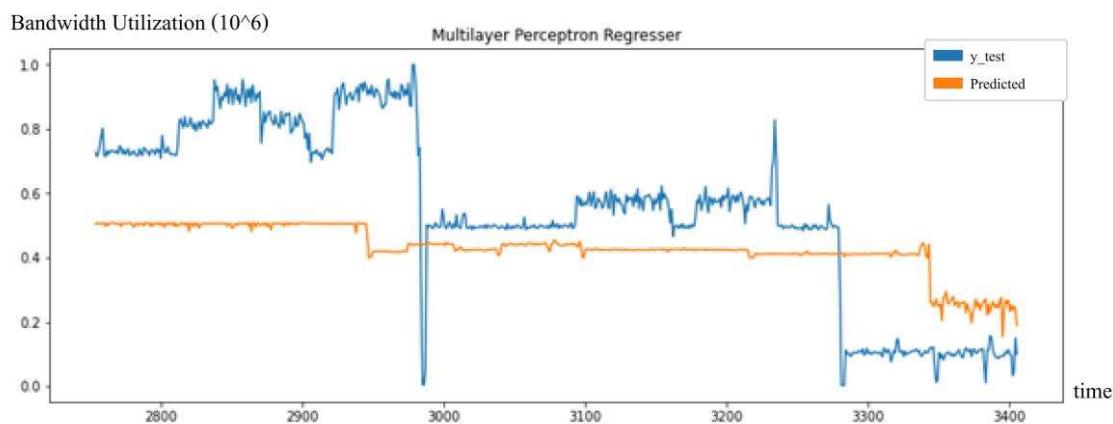
Bandwidth Utilization ( $10^6$ )

(b) กราฟเปรียบเทียบค่าชุดข้อมูลเฉพาะส่วน y\_test กับค่าที่ Linear Regressor คาดการณ์

รูปที่ 4.3 ผลการวัดความแม่นยำของ Random Forest Regressor

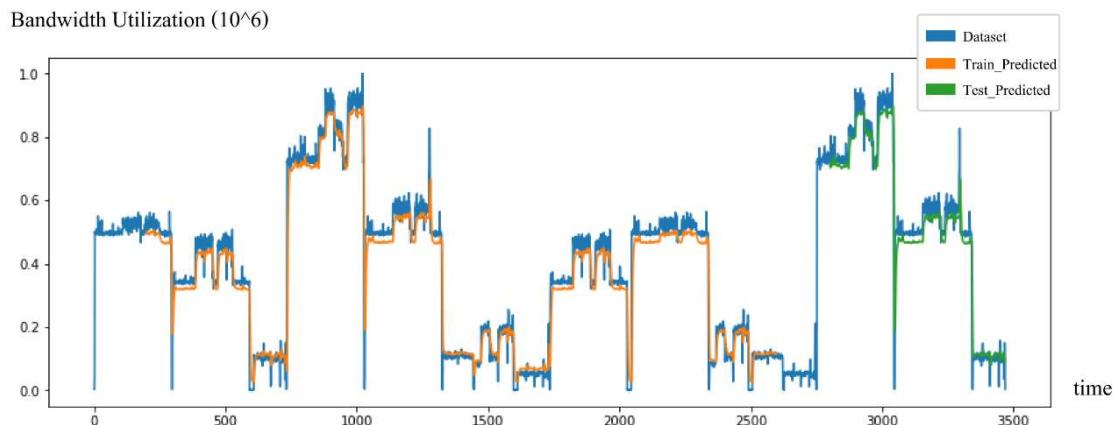


(a) กราฟเปรียบเทียบค่าชุดข้อมูลทั้งหมด กับค่าที่ MLP Regressor คาดการณ์



(a) กราฟเปรียบเทียบค่าชุดข้อมูลเฉพาะส่วน  $y_{\text{test}}$  กับค่าที่ MLP Regressor คาดการณ์

#### รูปที่ 4.4 ผลการวัดความแม่นยำของ Multi-Layer Perceptron



#### รูปที่ 4.5 ผลการวัดความแม่นยำของ Long-Short Term Memory

ผลจากการทดลองเบรียบเทียบความแม่นยำในการคาดการณ์ค่าการใช้ประโยชน์แบบดิจิตที่ โดยเลือกใช้อัลกอริทึมการเรียนรู้ชนิดต่างๆ ดังที่ได้แสดงไปในตารางที่ 4.1 และกราฟดังแสดงในรูปที่ 4.1 จนถึง 4.5 ช่วยให้เราตัดสินใจเลือกใช้อัลกอริทึมการเรียนรู้เชิงลึกประเภท Long-Short Term Memory มาใช้ในโครงการ เนื่องจากมีค่า Root Mean Square Error, Mean Absolute Error ที่ต่ำกว่าอัลกอริทึมชนิดอื่นๆ ซึ่งมีความแม่นยำในการคาดการณ์ที่สูงกว่าอัลกอริทึมชนิดอื่นๆ

## 4.2 การประเมินประสิทธิภาพของอัลกอริทึมการเลือกเส้นทางใหม่

โครงการนี้เราใช้ทรัพุต (Throughput) เป็นตัวชี้วัดสำหรับวัดประสิทธิภาพของอัลกอริทึมการเลือกเส้นทางใหม่ โดยทำการเขียนกราฟความสัมพันธ์ระหว่างทรัพุตเทียบกับปริมาณрафฟิกที่เข้ามาในเครือข่ายในหน่วยไบตต่อวินาที (pkt/s)

### 4.2.1 ทรัพุต

ทรัพุต (Throughput) หมายถึง ปริมาณข้อมูลที่สามารถส่งໄ�回ในระยะเวลาหนึ่ง วัดในหน่วยไบตต่อวินาที (bps) โดยเราจะวัดค่าทรัพุตระหว่างโอดส์ต้นทาง  $Host_i$  และโอดส์ปลายทาง  $Host_j$  หนึ่งคู่ โครงการนี้ได้คำนวณทรัพุตให้ออกมาในรูปแบบของ ทรัพุตavarum ซึ่งขยายความได้ว่า เป็นการคำนวณทรัพุตเฉลี่ยของคู่การส่งข้อมูลทั้ง 3 คู่ในเครือข่าย

### 4.2.2 การคำนวณทรัพุตavarum

เราได้นำแพ็กเก็ต FlowStatReplyMessage ของโอเพนไฟล์วีโพรโทคอลมาใช้เป็นส่วนหนึ่งสำหรับการคำนวณทรัพุต โดยเราสามารถทราบค่าปริมาณข้อมูลที่เข้ามายังแต่ละพอร์ตของสวิตช์ ในหน่วยไบต์ได้จากฟิลด์ Bytes ที่อยู่ภายใต้ FlowStatReplyMessage ดังแสดงในรูปที่ 4.6

จากนั้นคำนวณปริมาณข้อมูลเฉลี่ย แบ่งหน่วยปริมาณข้อมูลเฉลี่ยที่เข้ามายังหน่วยไบต์ให้เป็นหน่วยบิตและนำไปหารเวลาทั้งหมดที่มีการส่งข้อมูล

### 4.2.3 การคำนวณปริมาณрафฟิกที่เข้ามาในเครือข่าย

เราได้ใช้แพ็กเก็ต FlowStatReplyMessage มาใช้คำนวณปริมาณราฟฟิกที่เข้ามายังเครือข่าย เช่นกัน โดยเราสามารถดึงค่าจากฟิลด์ Pkts ที่บ่งบอกถึงจำนวนแพ็กเก็ตที่เข้ามายังแต่ละพอร์ตของสวิตช์มาใช้ได้ ดังแสดงในรูปที่ 4.6 จากนั้นนำไปหารเวลาทั้งหมดที่มีการส่งข้อมูล

datapath	in-port	destination	out-port	pkts	Bytes
----------	---------	-------------	----------	------	-------

รูปที่ 4.6 ฟิลด์ภายในแพ็กเก็ต FlowStatReplyMessage

ความหมายของฟิลด์ต่างๆภายในแพ็กเก็ต FlowStatReplyMessage เป็นไปตามที่ระบุในตารางที่ 4.2

Datapath	หมายเลขสวิตช์
In-port	หมายเลขพอร์ตขาเข้าของสวิตช์ที่รับแพ็กเก็ตเข้ามา
Destination	MAC Address ของเครื่องปลายทาง
Out-port	หมายเลขพอร์ตขาออกของสวิตช์ที่แพ็กเก็ตถูกส่งออกไป
pkts	จำนวนแพ็กเก็ตที่รับเข้ามา
Bytes	ปริมาณข้อมูลรวมทุกแพ็กเก็ตที่เข้ามา

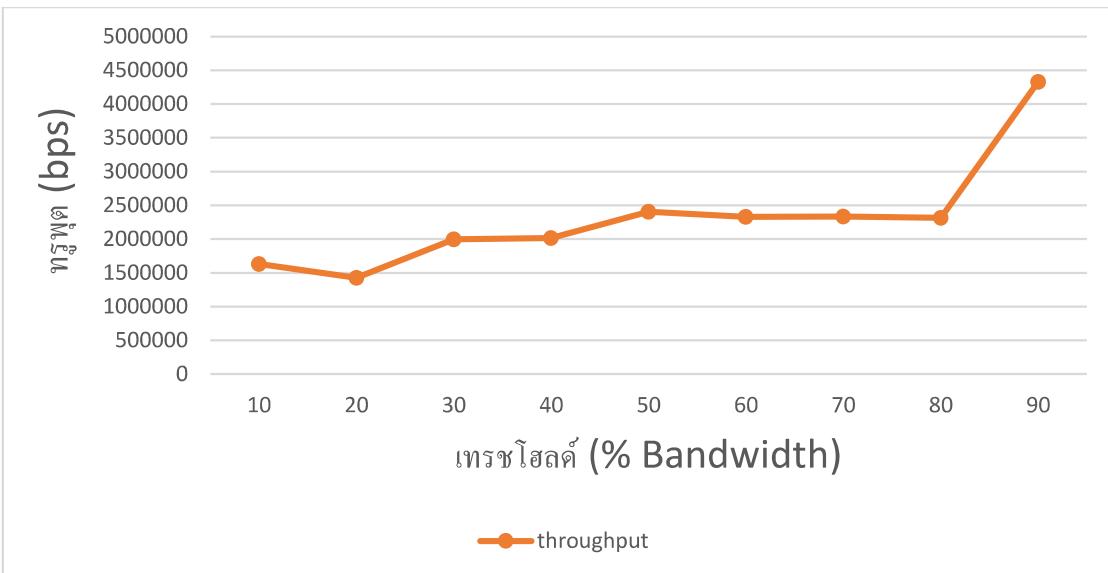
ตารางที่ 4.2 ความหมายของฟิลด์ต่างๆใน FlowStatReplyMessage

### 4.3 การเลือกค่าเทرزโอลด์ที่เหมาะสมกับโครงงาน

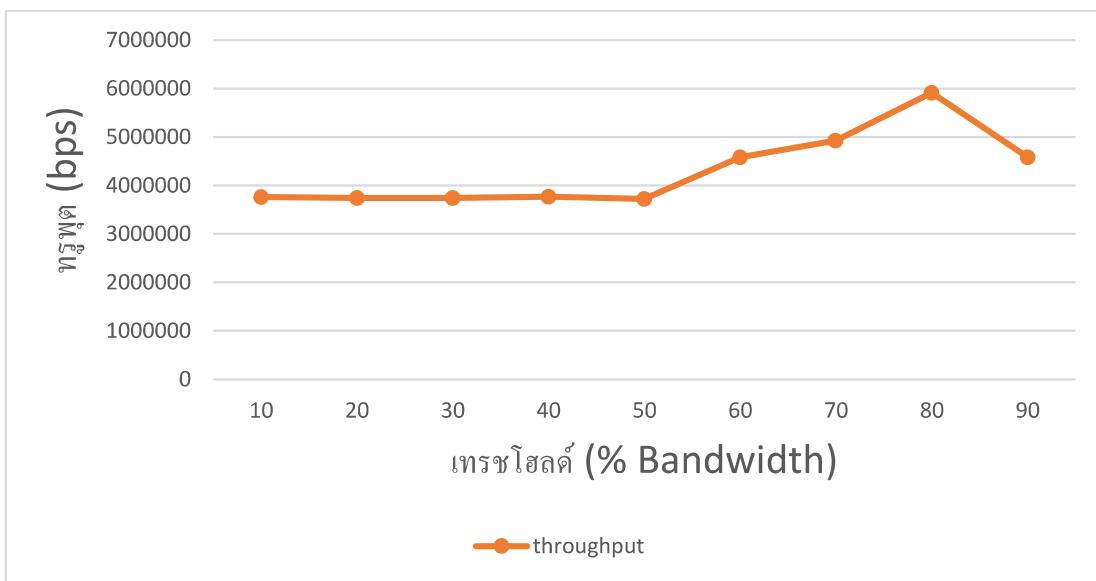
ค่าเทرزโอลด์เป็นพารามิเตอร์ที่มีส่วนสำคัญอย่างยิ่งต่อการเริ่มทำงานของขั้ลกอวิทึมการเลือกเส้นทางใหม่ เพื่อให้ได้ค่าเทرزโอลด์ที่เหมาะสมกับการทำงานมากที่สุด เราได้ทำการทดลองดังต่อไปนี้

#### 4.3.1 ค่าเทرزโอลด์ที่เหมาะสมกับระดับทราฟฟิกต่างๆ

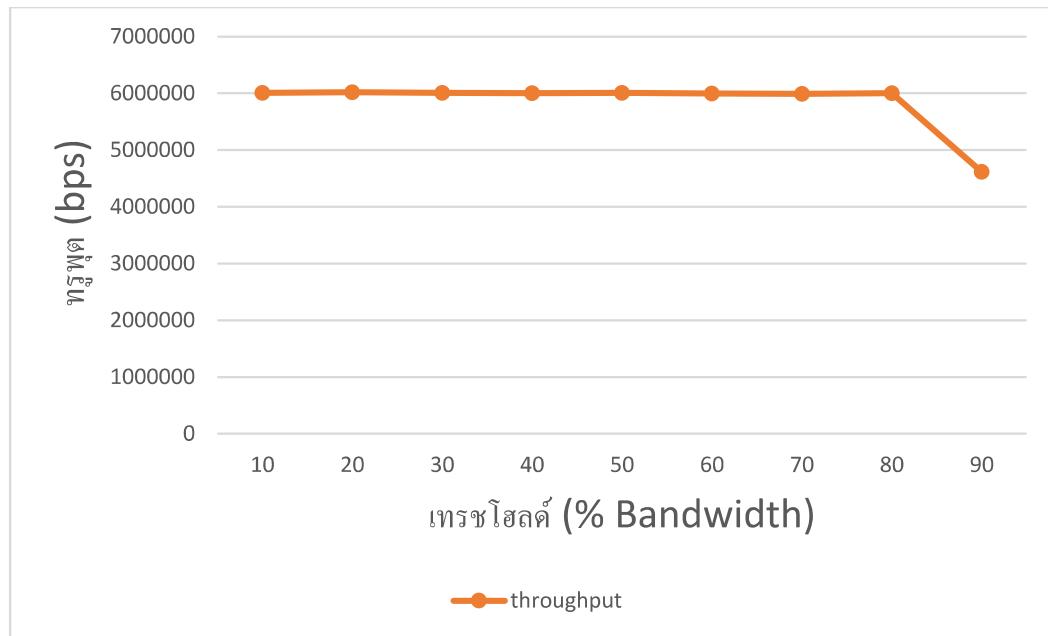
เราได้ทำการทดลองปรับจูนค่าเทرزโอลด์ที่ระดับทราฟฟิกต่ำ กลาง และสูง โดยทำการทดลองกำหนดค่าเทرزโอลด์เริ่มต้นจาก 10% ของแบบค์วิดท์ หรือ 1 Mbps เพิ่มขึ้นไปจนถึง 90% ของแบบค์วิดท์ หรือ 9 Mbps ผลการทดลองสำหรับระดับทราฟฟิกต่ำเป็นไปตามรูปที่ 4.7 ระดับกลาง เป็นไปตามรูปที่ 4.8 และระดับสูงเป็นไปตามรูปที่ 4.9



รูปที่ 4.7 ค่า吞吐率ที่ระดับเทอร์บิਊลด์ต่างๆ ในрафฟิคระดับคำ



รูปที่ 4.8 ค่า吞吐率ที่ระดับเทอร์บิਊลด์ต่างๆ ในrafฟิคระดับกล่อง

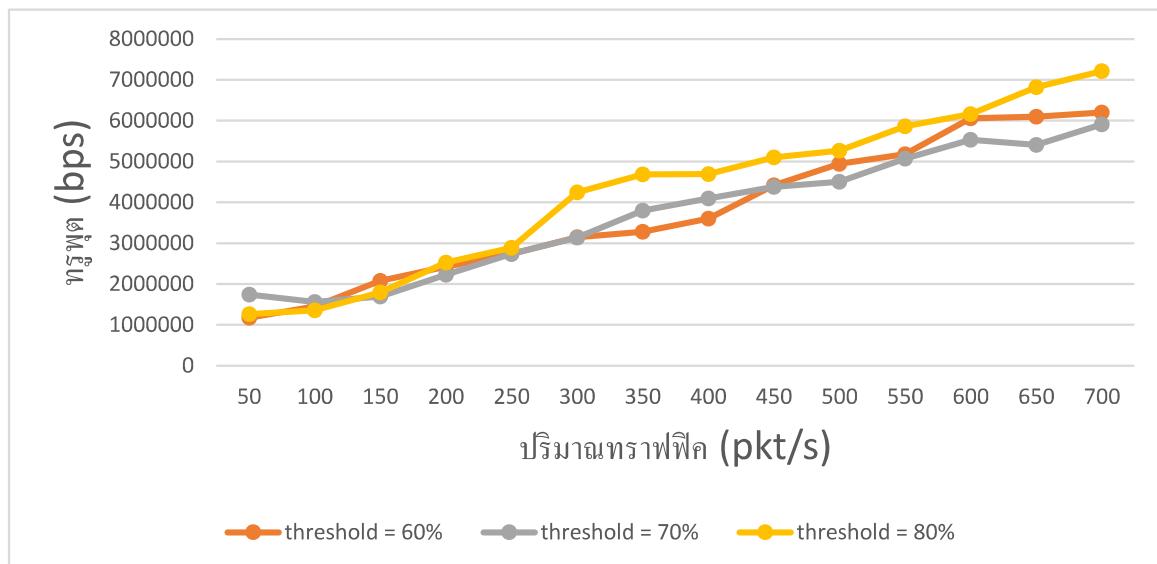


รูปที่ 4.9 ค่า throughput ที่ระดับเทอร์ชไอลด์ต่างๆ ใน trafic ไฟล์ ดับสูง

จากกราฟดังแสดงในรูปที่ 4.7 4.8 และ 4.9 เราได้วิเคราะห์ว่าค่าเทอร์ชไอลด์ที่เหมาะสมกับการนำมาใช้ในโครงงาน 3 อันดับแรก ได้แก่ ค่าเทอร์ชไอลด์ที่ 60% 70% และ 80% ของแบบค์วิดท์ โดยเราได้พิจารณาว่าเมื่อกำหนดค่าเทอร์ชไอลด์ไว้ที่ค่าดังกล่าว ค่า throughput ที่วัดได้ในเครือข่ายมีค่าที่สูงในทุกระดับ trafic ไฟล์

### 4.3.2 ค่าเทรชโซลด์ที่เหมาะสมกับโครงงาน

จากค่าเทรชโซลด์ทั้ง 3 ค่าที่ได้มาจากการหัวข้อที่แล้ว เรานำค่าดังกล่าวมาใช้กำหนดค่าเทรชโซลด์สำหรับการทดลองเพื่อศูนย์ประสิทธิภาพของอัลกอริทึมที่เรานำเสนอ และได้ผลลัพธ์ดังแสดงในรูปที่ 4.10



รูปที่ 4.10 เปรียบเทียบค่าทรูพุตเมื่อกำหนดค่าเทรชโซลด์ที่ 60% 70% และ 80% ของแบบดั้วิดที่

จากผลการทดลองดังแสดงในรูปที่ 4.10 เราได้เลือกใช้ค่าเทรชโซลด์ที่ 80% สำหรับการทดลองเนื่องจากสามารถทำให้ค่าทรูพุตสูงขึ้นไปได้สูงกว่าการเลือกเทรชโซลด์ที่ 60% และ 70% เกือบตลอดช่วงการทดลอง จะเห็นได้ว่ามีจุดที่ด้อยกว่าในช่วงต้นของการทดลองที่ยังมีปริมาณทราฟฟิกในเครือข่ายต่ำเท่านั้น

## 4.4 การประเมินประสิทธิภาพของโครงงาน

โครงงานนี้ได้เลือกใช้ทรูพุตสำหรับประเมินประสิทธิภาพของอัลกอริทึมการเลือกเส้นทางใหม่ โดยเป็นการวัดทรูพุต平均ของเครือข่าย (bps) เพียงกับปริมาณทราฟฟิกที่เข้ามาในเครือข่าย (pkt/s) หลังจากที่เราได้ทำการวัดประสิทธิภาพของเครือข่ายที่ใช้งานอัลกอริทึมการเลือกเส้นทางใหม่ เราได้นำไปเปรียบเทียบผลกับเครือข่ายแบบดั้งเดิมที่ไม่มีการใช้งานอัลกอริทึมการเลือกเส้นทางใหม่ และ

อัลกอริทึมจากการวิจัย “Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network”

#### 4.4.1 พารามิเตอร์การทดลอง

โครงงานนี้ได้ทำการจำลองเครือข่ายสำหรับการส่งข้อมูลตามพารามิเตอร์การทดลองดังที่แสดงในตารางที่ 4.2

อสเดีนคอนไทรอลเลอร์	ริวอสเดีนคอนไทรอลเลอร์ (RYU SDN Controller)
เน็ตเวิร์ค อิมูเลเตอร์	มินินेट (Mininet)
จำนวนสวิตช์	6 เครื่อง
จำนวน ไอดี	6 เครื่อง
แบบดิจิทอลิงค์	10 Mbps
เทอร์โบลด์	8 Mbps ( เที่ยบเท่า 80% Bandwidth )
โอเพนไฟล์	V1.3
โปรโทคอล	TCP

ตารางที่ 4.3 พารามิเตอร์ที่ใช้สร้างเครือข่ายสำหรับการทดลอง

สำหรับการส่งข้อมูลของไอดีที่ 3 เครื่องอันได้แก่ Host1, Host2 และ Host3 เราได้ทำการกำหนดพารามิเตอร์การส่งข้อมูลในซอฟต์แวร์ D-ITG ตามที่แสดงในตารางที่ 4.4

เครื่องต้นทาง	เครื่องปลายทาง	Inter-Departure time	ขนาดแพ็คเก็ต	โปรโทคอลการสื่อสาร	ระยะเวลา
Host 1	Host 4,5,6	-O X	-c 1500	-T TCP	1 นาที
Host 2	Host 4,5,6	-O 200	-c 1500	-T TCP	15 นาที
Host 3	Host 4,5,6	-O 200	-c 1500	-T TCP	15 นาที

ตารางที่ 4.4 พารามิเตอร์สำหรับกำหนดรูปแบบการส่งข้อมูลด้วยซอฟต์แวร์ D-ITG

Host1, Host2 และ Host3 จะสลับเปลี่ยนปลายทางของการส่งข้อมูลไประหว่าง Host4, Host5 และ Host6 ตลอดการทดลอง

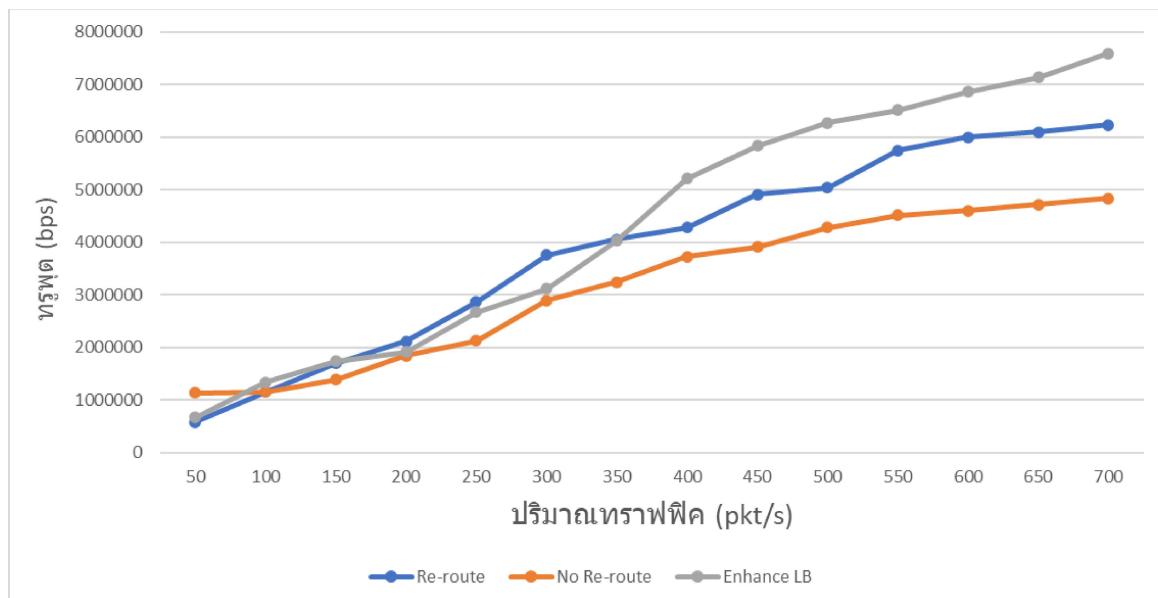
พารามิเตอร์ Inter-Departure Time ของ Host1 มีการกำหนดให้ปรับเปลี่ยนได้ขณะทำการทดลอง โดยค่าของ  $X$  จะเริ่มที่ 50 เพิ่มขึ้นทีละ 50 ไปสิ้นสุดที่ 700 เหตุผลที่เราต้องทำการปรับเปลี่ยนค่า Inter-Departure time ของ Host1 ระหว่างการทดลองก็เนื่องมาจากการที่ค่า Inter-Departure time นั้นมีลักษณะที่แปรผันตรงกับปริมาณทราฟฟิกที่เข้ามาในเครือข่าย ซึ่งเป็นค่าที่เรานำไปใช้สำหรับการวางแผนกับค่าทรูพุตเพื่อประเมินประสิทธิภาพของอัลกอริทึมการเลือกเส้นทางใหม่

#### 4.4.2 วิธีการทดลอง

ค่าทรูพุตและปริมาณทราฟฟิกที่เข้ามาในเครือข่ายจะถูกจดบันทึกไว้ระหว่างการทดลอง โดยเราได้ใช้วิธีอัปเดตอัตโนมัติในการบันทึกค่าดังกล่าวไว้ในไฟล์นามสกุล .csv เก็บเอาไว้ และใช้ git เป็นตัวกลางเพื่อนำออกไฟล์บันทึกผลการทดลองไปสู่ระบบปฏิบัติการwin โคลเวิร์สำหรับการวางแผนการณ์นำเสนอผลการทดลองต่อไป

#### 4.4.3 ผลการทดลอง

ผลการเปรียบเทียบประสิทธิภาพของเครือข่ายที่เปิดใช้งานอัลกอริทึมการเลือกเส้นทางใหม่ กับเครือข่ายแบบเดิมกันที่ไม่มีการเปิดใช้งานอัลกอริทึมการเลือกเส้นทางใหม่ และอัลกอริทึมจากงานวิจัย “Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network” เป็นไปตามกราฟดังแสดงในรูปที่ 4.11

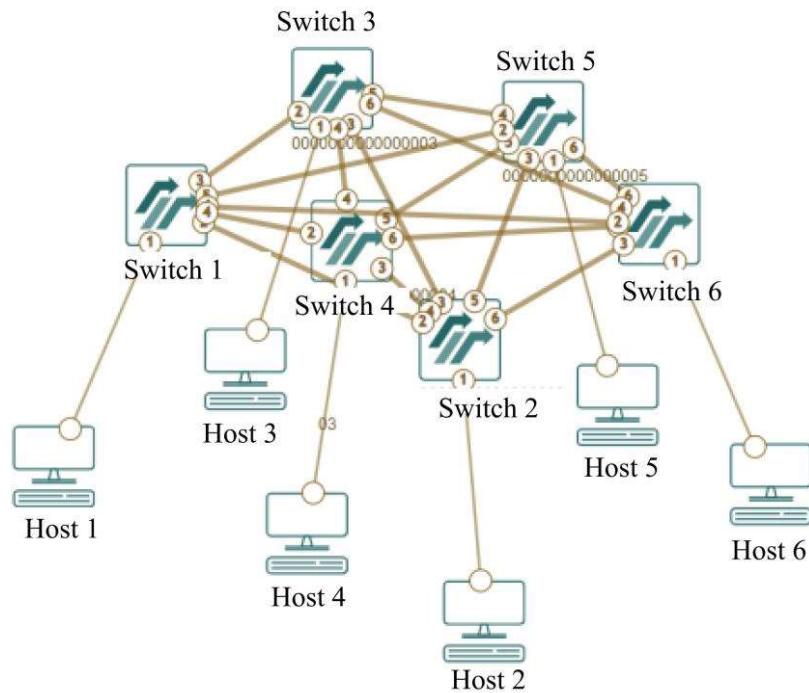


รูปที่ 4.11 ผลการเปรียบเทียบระหว่างอัลกอริทึมต่างๆ

จากกราฟ องค์นิยมได้รับการเปิดใช้งานอัลกอริทึมการเลือกเส้นทางใหม่สามารถช่วยให้ค่าทรูพุตภายนอกในเครือข่ายมีค่าสูงกว่าในกรณีเครือข่ายธรรมชาติไม่มีการใช้งานอัลกอริทึมตลอดช่วงการทดลอง แต่เมื่อนำไปเบริญเทียบกับอัลกอริทึมจากการวิจัย “Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network” แล้วโดยรวมจะเห็นได้ว่าอัลกอริทึมของเรามีประสิทธิภาพด้านทรูพุตที่ดีอยกว่า ถึงแม้ว่าจะเหนื่อยกว่าได้เป็นบางจุดในช่วงที่ปริมาณрафฟิกในเครือข่ายยังอยู่ในระดับต่ำ

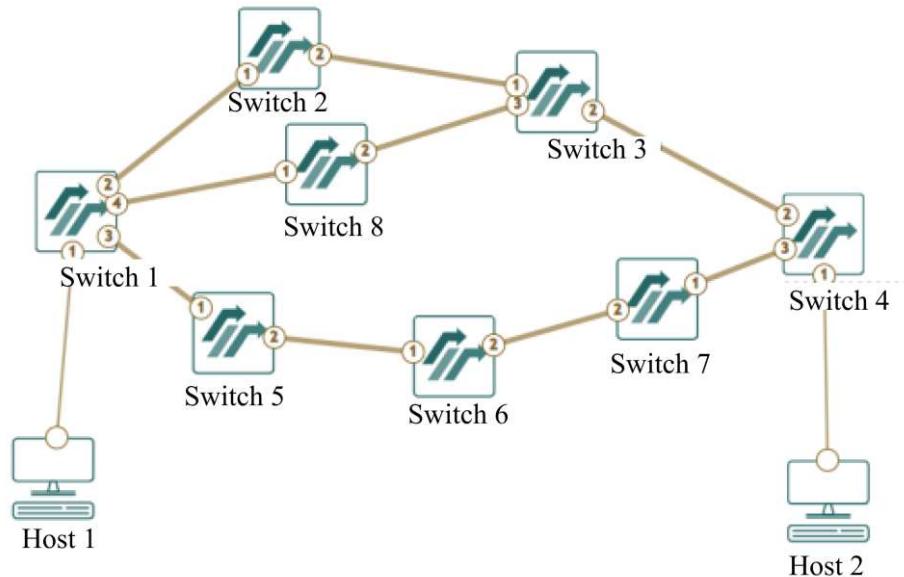
#### 4.4.4 การทดลองเพิ่มเติม

เราได้ทำการทดลองเพิ่มเติม โดยนำอัลกอริทึมการเลือกเส้นทางใหม่ของเรามาไปทำงาน และวัดประสิทธิภาพด้านทรูพุตบน拓扑โลจิกที่มีลักษณะเชื่อมต่อ กันทั้งหมด (Fully-Connected) ดังแสดงในรูปที่ 4.12 และ拓扑โลจิกที่งานวิจัย “Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network” ใช้ดังแสดงในรูปที่ 4.13



รูปที่ 4.12 拓扑โลจิกแบบเชื่อมต่อ กันทั้งหมด

สำหรับการทดลองบนトイโพโลยีแบบเชื่อมต่อกันทั้งหมด เราจึงคงใช้กระบวนการทดลองแบบเดิม กล่าวคือ Host 1 Host 2 และ Host 3 ทำหน้าที่เป็นเครื่องส่งข้อมูลเข้าสู่เดิม และส่งค่วยพารามิเตอร์การส่งข้อมูลด้วย D-ITG เช่นเดิม



รูปที่ 4.13 トイโพโลยีอ้างอิงจากงานวิจัย Enhance Load Balanced

การทดลองบนトイโพโลยีที่อ้างอิงจากงานวิจัยตามที่แสดงในรูปที่ 4.13 นี้ เราได้กำหนดให้ Host 2 เป็นโฉลท์สำหรับส่งข้อมูล และ Host 1 เป็นโฉลท์สำหรับรับข้อมูล ซึ่งเราได้อ้างอิงรูปแบบการสื่อสารดังกล่าวตามงานวิจัยต้นฉบับ

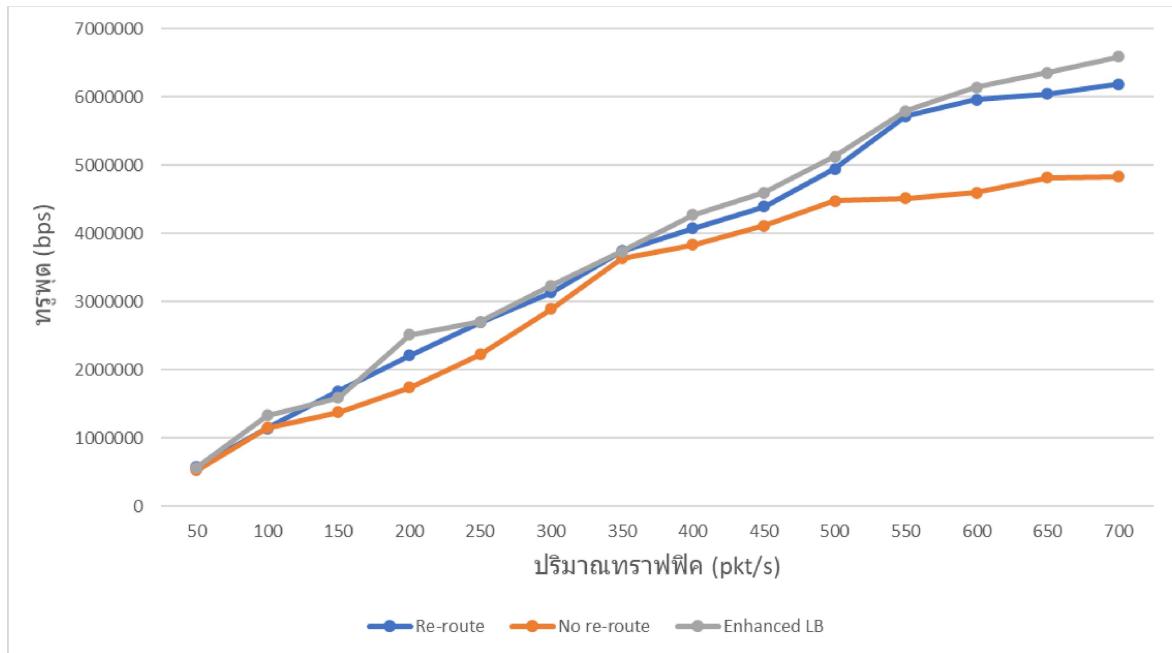
เครื่องต้นทาง	เครื่องปลายทาง	Inter-Departure time	ขนาดแพ็คเก็ต	โปรโตคอลการสื่อสาร	ระยะเวลา
Host 2	Host 1	-O X	-c 1500	-T TCP	1 นาที

ตารางที่ 4.5 พารามิเตอร์การส่งข้อมูลด้วยซอฟต์แวร์ D-ITG สำหรับการทดลองเพิ่มเติม

ในการทดลอง เราได้กำหนดค่าของ  $x$  ไว้ เช่นเดียวกันกับการทดลองก่อนหน้านี้ กล่าวคือ เริ่มต้นที่ 50 เพิ่มขึ้นทีละ 50 ไปสิ้นสุดที่ 700 เช่นเดิม

#### 4.4.5 ผลการทดลองเพิ่มเติม

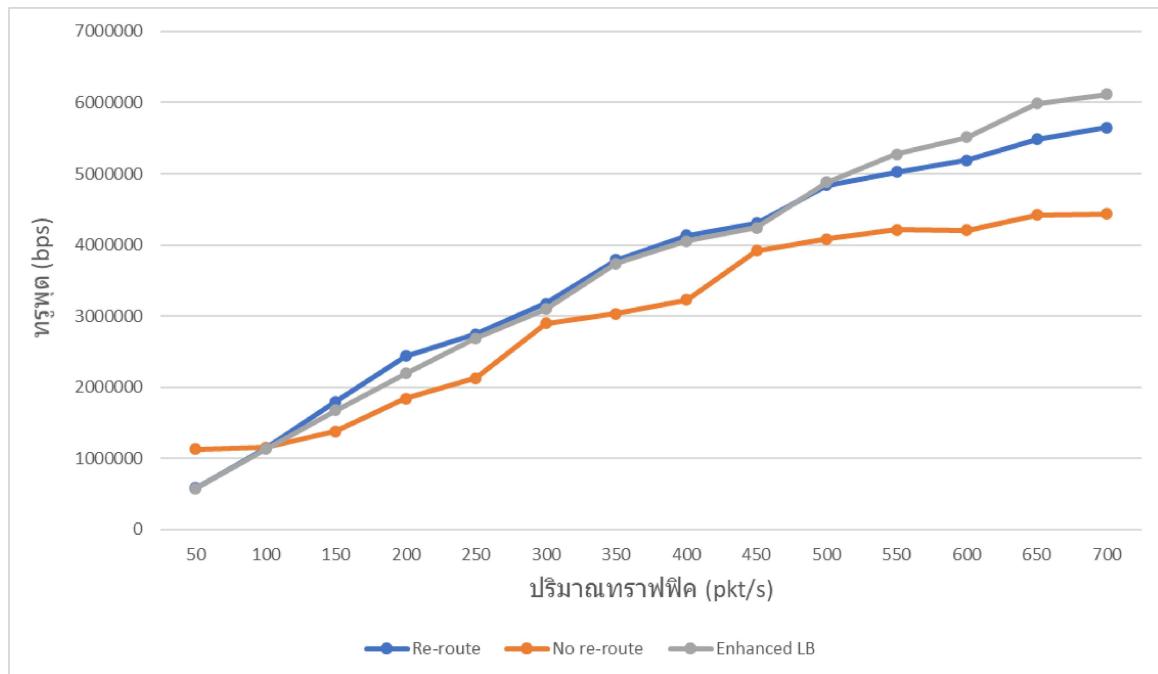
ผลจากการทดลองวัดประสิทธิภาพอัลกอริทึมการเลือกเส้นทางใหม่ บนโ拓โพโลยีที่มีลักษณะเชื่อมต่อ กันหมด ได้ผลดังกราฟที่แสดงในรูปที่ 4.14



รูปที่ 4.14 ผลการทดลองบนโ拓โพโลยีแบบเชื่อมต่อ กันหมด

จากราฟ อธิบายได้ว่าเมื่อปรับโ拓โพโลยีการทดลองให้มีลักษณะเป็นแบบเชื่อมต่อ กันหมด อัลกอริทึมการเลือกเส้นทางใหม่ของเรายังคงทำงานได้ดีกว่า เครือข่ายแบบธรรมชาติที่ไม่มีการใช้งานอัลกอริทึมการเลือกเส้นทางใหม่ เช่นเดิม แต่สิ่งที่สังเกตุได้เพิ่มเติมคือ อัลกอริทึมของเรามีประสิทธิภาพการทำงานที่ใกล้เคียงกับอัลกอริทึมจากงานวิจัย “Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network” มากขึ้นกว่าเดิม เราระยะห์ผลได้ว่าเหตุที่ อัลกอริทึมของเรามีประสิทธิภาพสูงขึ้น เนื่องมาจากการมีเส้นทางสำรองสำหรับเป็นตัวเลือกในการเปลี่ยนเส้นทางเพิ่มมากขึ้น เมื่อโ拓โพโลยีมีลักษณะเป็นแบบเชื่อมต่อ กันหมด

ผลจากการทดลองวัดประสิทธิภาพอัลกอริทึมการเลือกเส้นทางใหม่ บนโทโพโลยีที่อ้างอิงจากงานวิจัย “Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network” ได้ผลดังกราฟที่แสดงในรูปที่ 4.15



รูปที่ 4.15 ผลการทดลองบนโทโพโลยีอ้างอิงจากงานวิจัย Enhance Load Balanced

จากราฟ สามารถอธิบาย ได้ว่าเมื่อปรับโทโพโลยีการทดลองให้มีลักษณะ อ้างอิงตามที่งานวิจัย “Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network” ได้ใช้ อัลกอริทึมการเลือกเส้นทางใหม่ของเรายังคงทำงาน ได้ดีกว่า เครือข่ายแบบธรรมดาก็ไม่มีการใช้งานอัลกอริทึมการเลือกเส้นทางใหม่ เช่นเดียวกันกับการทดลองทั้งหมดก่อนหน้านี้ แต่สิ่งที่สังเกตุได้เพิ่มเติมคือ อัลกอริทึมของเรามีประสิทธิภาพการทำงานที่ทัดเทียมและเหนือกว่าเล็กน้อย ในช่วงที่ในเครือข่ายมีปริมาณทราฟฟิกระดับต่ำและระดับกลาง แต่อย่างไรก็ตาม เมื่อในเครือข่ายมีปริมาณทราฟฟิกในระดับสูง อัลกอริทึมของเราจะมีประสิทธิภาพด้านทรัพุที่ดีอยกว่าเล็กน้อย

## บทที่ 5

### สรุปผล และข้อเสนอแนะ

ในบทนี้ เราได้นำเสนอการสรุปผลของการทดลอง การจัดเส้นทางด้วยกลไกคาดการณ์แบบด้วยวิธีที่ไม่ใช้เครือข่ายอัลกอริทึม ซึ่งแสดงถึงปัญหาหากหลายค้านที่ได้พบตลอดการทำโครงการ และเสนอแนวทางในการพัฒนาโครงการนี้ให้ดียิ่งขึ้นต่อไปในอนาคต

#### 5.1 สรุปผลที่ได้จากการทดลอง

โครงการนี้ได้นำเสนออัลกอริทึมการเลือกเส้นทางใหม่ (Re-Routing Algorithm) ด้วยกลไกคาดการณ์ค่าการใช้ประโยชน์แบบด้วยวิธีที่ไม่ใช้เครือข่ายอัลกอริทึม (SDN) เราได้นำอัลกอริทึมการเรียนรู้เชิงลึกประเพณี Long-short Term Memory (LSTM) มาใช้ในการคาดการณ์ค่าการใช้ประโยชน์แบบด้วยวิธีของลิงค์ หากค่าที่คาดการณ์นั้นสูงกว่าค่าเทreshold (Threshold) ที่กำหนดไว้ก่อนโตรลเลอร์ (RYU Controller) จะทำการออกคำสั่งไปยังสวิตช์ให้เปลี่ยนเส้นทางการส่งข้อมูล

จากการทดลองแสดงให้เห็นถึงความสามารถของเทคโนโลยีอัลกอริทึมการเรียนรู้เชิงลึก (Deep Learning) ที่สามารถช่วยพัฒนาระบบเครือข่ายแบบดั้งเดิม (Traditional Network) ให้มีความสามารถในการส่งข้อมูลที่สูงขึ้น โดยเราได้แสดงให้เห็นผ่านการทดลองส่งข้อมูล และวัดค่าทรัพุตภาระ (Throughput) ของเครือข่าย ซึ่งได้ผลลัพธ์ว่า เมื่อเพิ่มปริมาณทราฟฟิกที่เข้ามาในเครือข่ายอย่างต่อเนื่อง เครือข่ายที่มีการเปิดใช้งานอัลกอริทึมการเลือกเส้นทางใหม่ จะมีค่าทรัพุตที่สูงกว่าเครือข่ายที่ไม่มีการเปิดใช้งานอัลกอริทึมการเลือกเส้นทางใหม่ ดังแสดงในรูปที่ 4.11 แต่อย่างไรก็ตาม โดยรวมแล้วอัลกอริทึมการจัดเส้นทางด้วยกลไกคาดการณ์แบบด้วยวิธีที่ไม่ใช้เครือข่าย สามารถแสดงให้เห็นว่า อัลกอริทึมการเลือกเส้นทางของเรามีประสิทธิภาพด้านทรัพุตที่ใกล้เคียงกับอัลกอริทึมจากงานวิจัยที่เราอ้างอิงมากขึ้น

## 5.2 ปัญหาที่พบ

### 5.2.1 การขาดแคล่งความรู้ทำให้เกิดความยากในการศึกษา

ด้วยเทคโนโลยีอิเล็กทรอนิกส์เป็นเทคโนโลยีที่ค่อนข้างใหม่ และยังอยู่ในช่วงของการศึกษา และพัฒนา ยังไม่มีการนำไปใช้งานจริงเป็นวงกว้าง มีการใช้งานจริงอยู่แค่ภายในวงแคบๆ ของบริษัท ระดับโลกขนาดใหญ่เท่านั้น ทำให้การค้นคว้าหาความรู้ของอิเล็กทรอนิกส์ในเชิงปฏิบัติและการทดลอง เป็นไปด้วยความยากลำบาก การหาความรู้บนโลกอินเทอร์เน็ตมักพบแต่เพียงทฤษฎีต่างๆ ของอิเล็กทรอนิกส์ เกี่ยวกับความรู้ด้านการทดลองการใช้งานอิเล็กทรอนิกส์มักหาได้เพียงเฉพาะจากการศึกษาเอกสารงานวิจัย ที่เก่า舊 แล้วคุณมือการใช้อุปกรณ์เบื้องต้น (Document) ที่จัดทำขึ้นโดยผู้พัฒนา

### 5.2.2 การลองผิดลองถูกกับอุปกรณ์หลายอย่างทำให้เริ่มต้นโครงการได้ช้า

สืบเนื่องจากปัญหาในข้อที่ 5.2.1 ในช่วงต้นของการทำโครงการนี้ เราได้ทดลองใช้ อุปกรณ์ไปหลากหลายตัว เนื่องจากเราไม่รู้ว่างานในลักษณะต่างๆ ควรที่จะใช้อุปกรณ์ หรือซอฟต์แวร์ ตัวไหนจะดีกว่ากัน และด้วยปัญหาที่ว่าขาดแคล่งข้อมูลที่แพร่หลาย ทำให้เราจำเป็นต้องลองผิดลอง ถูกกับซอฟต์แวร์ไปหลายตัว เช่น ในช่วงต้นของการศึกษาโครงการ เราได้ใช้ซอฟต์แวร์ GNS3 สำหรับ การสร้างเครือข่ายจำลองขึ้นมา ด้วยเหตุที่ว่าเป็นซอฟต์แวร์ที่ได้รับความนิยม และสามารถใช้งานร่วมกับซอฟต์แวร์อื่นอย่างอิเล็กทรอนิกส์คอนโทรลเลอร์ (SDN Controller) มินิเน็ต (Mininet) อย่างที่เราใช้ใน โครงการนี้ แต่ก็พบปัญหาที่ GNS3 มีการใช้ทรัพยากรเครื่องที่สูงมาก ทำให้ต้องเปลี่ยนวิธีการทำ โครงการไป

### 5.2.3 ข้อจำกัด และปัจจัยด้านทรัพยากรีเตอร์

ซอฟต์แวร์ D-ITG ซึ่งเป็นทรัพยากรีเตอร์ (Traffic Generator) ที่เราใช้มีจุด ความสามารถในการส่งข้อมูลที่น้อยเกินไป ในตอนแรกโครงการของเราได้ตั้งค่าแบบนิวตันวิดท์ของลิงค์ให้อยู่ที่ 100 Mbps ซึ่งสอดคล้องกับเทคโนโลยีไฟเบอร์ออฟติก (Fast Ethernet) แต่ D-ITG ไม่สามารถส่ง การให้ไฟเบอร์ส่งข้อมูลมากในระดับที่จะทำให้ค่าการใช้ประโยชน์แบบนิวตันวิดท์ (Bandwidth Utilization) สูงขึ้นไปถึงระดับเทอร์โบไซด์ (Threshold) ที่เรากำหนด ส่งผลให้เราต้องปรับลดแบบนิวตันวิดท์ของลิงค์ลง มาให้เหลือเพียง 10 Mbps แต่ทั้งนี้เราเข้าใจได้ว่าหากซอฟต์แวร์ D-ITG ไม่จำกัดปัจจัยความสามารถเอาไว้ ก็อาจทำให้คอมพิวเตอร์ของเรามีการทำงานที่หนักเกินไป

### 5.3 แนวทางในการพัฒนาต่อ

จากการทำโครงการ การจัดเส้นทางด้วยกลไกการคาดการณ์แบบด้วยที่ในเครือข่ายอสังหาริมทรัพย์ เราได้เรียนรู้จากข้อผิดพลาด และวางแผนถึงการพัฒนาต่อไปดังนี้

- 1) นำอัลกอริทึมการจัดเส้นทางด้วยกลไกการคาดการณ์แบบด้วยที่ไปใช้กับอุปกรณ์เครือข่ายจริง ที่ไม่ได้เป็นเพียงอุปกรณ์จำลอง
- 2) ในการเลือกเส้นทางใหม่ อาจมีการพิจารณาถึงองค์ประกอบทางเครือข่าย เช่น บัฟเฟอร์ ค่าการใช้ประโยชน์คิว (Queue Utilization) เป็นปัจจัยการเลือกเส้นทาง สำรองเพิ่มเติมจากการใช้เพียงแค่เส้นทางที่ ไดกราฟตราอัลกอริทึม (Dijkstra's Algorithm) เลือกไว้
- 3) ปรับปรุงอัลกอริทึมให้สามารถทำงานได้ดีในโทโพโลยีห้ามหลายรูปแบบ
- 4) ในอนาคตอาจมีการนำโมเดลการเรียนรู้เชิงลึกชนิดอื่นๆ ที่มีความคลาด มีความแม่นยำในการคาดการณ์ข้อมูลประเภทอนุกรมเวลา (Time-Series) ที่มากกว่า Long-Short Term Memory มาใช้งาน

## បររបាណអ្នករោម

- [1] Artificial Neural Network Based Load Balancing On Software Defined Networking  
<https://ieeexplore.ieee.org/document/8951365>
- [2] Haipeng Yao ; Xin Yuan ; Peiying Zhang ; Jingjing Wang ; Chunxiao Jiang ; Mohsen Guizani “Machine Learning Aided Load Balance Routing Scheme Considering Queue Utilization”, IEEE, 2019 <https://ieeexplore.ieee.org/document/8733872>
- [3] R. Dahlia Devapriya ; S. Indira Gandhi “Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network”, IEEE, 2020  
<https://ieeexplore.ieee.org/document/9077801>
- [4] Manel Majdoub ; Ali El Kamel ; Habib Youssef “Routing Optimization in SDN using Scalable Load Prediction”, IEEE, 2020 <https://ieeexplore.ieee.org/document/9044960>
- [5] Kuldeep Kurroliya ; Sagarika Mohanty ; Khushboo Kanodia ; Bibhudatta Sahoo “Grey Wolf Aware Energy-saving and Load-balancing in Software Defined Networks Considering Real Time Traffic”, IEEE, 2020 <https://ieeexplore.ieee.org/document/9112569>
- [6] Diego Kreutz, Member, IEEE, Fernando M. V. Ramos, Member, IEEE, Paulo Verissimo, Fellow, IEEE, Christian Esteve Rothenberg, Member, IEEE, Siamak Azodolmolky, Senior Member, IEEE, and Steve Uhlig, Member, IEEE, “Software-Defined Networking: A Comprehensive Survey” ,IEEE, 2014
- [7] វិច្ឆិកអំពី Software-defined Network(SDN) <https://www.blognone.com/node/56144>
- [8] Tony Bourke, “Server Load Balancing” , 1st Edition, United States of America: O'Reilly, 2001
- [9] Cloud Computing – Load Balancing <https://www.youtube.com/watch?v=zaLJ6SIHztk>

- [10] Truong Thu Huong ; Ngo Do Dang Khoa ; Nguyen Xuan Dung ; Nguyen Huu Thanh, “A global multipath load-balanced routing algorithm based on Reinforcement Learning in SDN” , IEEE, 2019
- [11] สร้าง AI ให้รู้ลึก รู้จริง ! ทำความรู้จักกับ Deep Learning คืออะไร <https://blog.pttexpresso.com/get-to-know-deep-learning/>
- [12] Deep Learning คืออะไร? <https://new.abb.com/news/detail/58004/deep-learning>
- [13] Ali Malik ; Ruairí de Fréin ; Mohammed Al-Zeyadi ; Javier Andreu-Perez “Intelligent SDN Traffic Classification Using Deep Learning: Deep-SDN”, 2020 2nd International Conference on Computer Communication and the Internet (ICCCI)
- [14] Routing Protocol by Mr.Jodoi <http://www.jodoi.org>
- [15] Routing คืออะไร <http://prapapon2131.blogspot.com/2009/08/routing.html>
- [16] แนวทางการเลือกใช้ routing protocol ภายในองค์กร (IGP) ในมุมมองของ โภค-ชัยวัฒน์ <https://www.bloggang.com/viewdiary.php?id=likecisco&month=08-2015&date=31&group=3&gblog=52>
- [17] Packet Queuing and Dropping in Routers <https://www.geeksforgeeks.org/packet-queuing-and-dropping-in-routers/>
- [18] Yi-Ren Chen ; Amir Rezapour ; Wen-Guey Tzeng ; Shi-Chun Tsai “RL-Routing: An SDN Routing Algorithm Based on Deep Reinforcement Learning”, IEEE, 2020
- [19] Siamak Azodolmolky, “Software Defined Networking with OpenFlow” 1st Edition, Birmingham : PACKT Publishing, 2013
- [20] Leonardo de Carvalho Freitas Padilha Aguilar, “Analysis of the use of SDN for load balancing” , Sao Paolo, 2018
- [21] OpenFlow/SDN คืออะไร <https://krisachai.wordpress.com/2013/11/25/openflowsdn-คืออะไร/>

- [22] The basics of SDN and the OpenFlow Network Architecture <https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture/>
- [23] Iman Akbari, “SDN part 1: What is Software-defined Networking (SDN) and why should I know about it?” <https://medium.com/@blackvvine/sdn-part-1-what-is-software-defined-networking-sdn-and-why-should-i-know-about-it-e73a250cecc>
- [24] “Software-defined network” ແນວດຂອງ network ແນບໃໝ່ ໄກສະລັບເປົ້າ <https://medium.com/oeufhp/software-defined-network-ແນວດີຂອງ-network-ແນບໃໝ່-ໄກສະລັບເປົ້າ-805c7046ae9>
- [25] Mindaugas Rimavičius, “Software Defined Network” <https://medium.com/noia/software-defined-network-572b1deeabb>
- [26] Routing Protocol [https://th.wikipedia.org/wiki/Routing\\_Protocol](https://th.wikipedia.org/wiki/Routing_Protocol)
- [27] ນທ 16 Lab Static Route [https://www.jodoi.com/book/Static\\_Route\\_lab\\_jodoi.pdf](https://www.jodoi.com/book/Static_Route_lab_jodoi.pdf)
- [28] ກາຣຈັດເສັ້ນທາງແນບພລວຕ ດິນາມີຣູ ຮູ ຕັ້ງ <http://blog.bru.ac.th/wp-content/uploads/bp-attachments/12690/CH11-Dynamic-Routing.pdf>
- [29] Deep Learning ຕື່ອອະໄຮ ? [https://www.thaiprogrammer.org/2\\_0\\_1\\_8 / 1\\_2 / deep-learning-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/](https://www.thaiprogrammer.org/2_0_1_8 / 1_2 / deep-learning-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/)
- [30] [ML] LSTMs ແນບ (ເກືອບ) ລະເອີຍດີນັງ <https://medium.com/@tongkornkitt/ml-lstms-ແນບ-ເກືອບ-ລະເອີຍດີນັງ-a3a55cd37883>
- [31] ເຮາເຕອຮ້ <https://th.wikipedia.org/wiki/ເຮາເຕອຮ້>
- [32] ກາຣສູນເລີຍແພັກເກີ້ຕ: ກາຣຕຽບສອນແລະແກ້ໄຂ <https://th.geekmarkt.com/packet-loss-checking-it>
- [33] How to fix Packet Loss for good <https://www.comparitech.com/net-admin/how-to-fix-packet-loss/>
- [34] Bandwidth <https://www.nstda.or.th/th/nstda-knowledge/3020-bandwidth>