การพัฒนาการใช้งาน WLAN ด้วยการประยุกต์ใช้งาน SDN
WLAN USABILITY IMPROVEMENT BY APPLYING SDN

ชัยพร เขมะภาตะพันธ์ [1]
ศุภชัย ผ่องราษี[2]

บทคัดย่อ

การพัฒนาการใช้งาน WALN ด้วยการ ประยุกต์ใช้งาน SDN เพื่อศึกษาการทำงานเทคโนโลยี Software Defined Networking (SDN) มาประยุกต์ใช้งานในระบบเครือข่ายไร้สาย ในการสร้างกฎในเครือข่าย SDN ในกรณีศึกษา การจัดสรรทรัพยากรเพื่อการใช้งานแบนด์วิธบนเครือข่ายไร้สายและกรณีศึกษา การสร้างกฎของการกำหนดสิทธิ์ไม่ให้เข้าใช้บริการในการโอนย้ายไฟล์ด้วย File transfer Protocol (FTP) การสร้างกฎในการกำหนดไม่ให้ใช้งาน ip address และ mac address ภายในเครือข่ายไร้สายที่กำหนดด้วยซอฟต์แวร์ และเพื่อนำระบบช่วยจัดการเครือข่ายแบบ SDN Controller เป็นทางเลือกในการตัดสินใจในการนำไปประยุกต์ใช้งานในระบบเครือข่าย จากการกำหนดการใช้งานแบนด์วิธและการกำหนดสิทธิ์ไม่ให้ใช้บริการ FTP Protocol ผ่านช่องทางการสื่อสาร Port 21 รวมถึง ip address และ mac address โดยมีการจัดเตรียมระบบเพื่อจำลองเครือข่ายแบบ SDN ตั้งแต่การติดตั้ง Software เพื่อทำหน้าที่เป็น Controller ติดตั้งอุปกรณ์บอร์ด Raspberry Pi เพื่อทำหน้าที่เป็น Switch Open flow เพื่อทำหน้าที่ในการรับคำสั่งการทำงานจาก Controller และมีการกำหนดการตั้งค่าการทำงานของ Controller กับ Flow switch เพื่อให้การทำงานร่วมกันโดยไม่มีข้อผิดพลาดและให้เหมาะสมกับการทดสอบ รวมถึงการกำหนดรูปแบบการเชื่อมต่อในเครือข่ายและการกำหนดการทำงานให้กับ host สามารถปรับปรุงทิศทางการไหลของข้อมูลเพื่อใช้ในการทดสอบให้การใช้งานในเครือข่ายได้อย่างมีประสิทธิภาพและใช้ได้จริง

ผลการทดลองพบว่า การกำหนดกฎในการใช้งานแบนด์วิธบนระบบเครือข่าย SDN ได้ออกแบบไว้ มีการทำงานได้อย่างถูกต้องและปฏิบัติได้จริง สามารถกำหนดการทำงานของข้อมูลได้ด้วย API ที่ใช้เป็นสร้างกฎในการทำงานของเครือข่าย และระบบสามารถนำผลการทดสอบจากการ

---

[1] วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ (chaiyaporn@dpu.ac.th)

[2] วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์ (605162010028@dpu.ac.th)

ทำงานบนเครือข่ายที่กำหนดด้วยซอฟต์แวร์นี้ สามารถที่จะนำมาประยุกต์ใช้งานในการพัฒนาเครือข่ายขององค์กรได้

คำสำคัญ: เครือข่ายกำหนดด้วยซอฟต์แวร์, เอสดีเอ็นคอนโทรลเลอร์, โอดีแอล, โอเพนเดย์ไลท์

บทนำ

ในปัจจุบันในการให้บริการอินเตอร์เน็ตในระบบเครือข่ายแบบไร้สายเข้ามามีบทบาทมากขึ้นและถือได้ว่ามีบทบาทสำคัญต่อการดำรงชีวิตของมนุษย์ เพราะว่าระบบเครือข่ายแบบไร้สายมีความสะดวกในการติดตั้งใช้งาน ลดค่าใช้จ่าย และเข้าถึงการใช้งานได้ง่าย แต่ในกรณีเรื่องความปลอดภัยในระบบหรือมีการใช้งานจำนวนมากในองค์กรโดยที่ไม่มีความจำเป็นก็ถือว่าเป็นเรื่องสำคัญในการบริการจัดการ อย่างเช่น ในเรื่องของการจัดสรร Bandwidth หรือการจำกัดการเข้าถึงที่ไม่มีความจำเป็นในทรัพยากรที่มีจำกัดและการจัดการในการเข้าใช้งานเพื่อความปลอดภัยในระบบเครือข่าย เทคโนโลยีเครือข่ายกำหนดด้วยซอฟต์แวร์ Software Defined Networks (SDN) เป็นแนวคิด ในการจัดการระบบเครือข่ายแบบใหม่ มีการแยก Control plane และ Data plane ออกจากกัน Control Plane เป็นการควบคุมการจัดการ Packet Flow รวบรวมข้อมูลทรัพยากรของระบบทั้งหมดไปไว้ Controller ผู้ดูแลระบบสามารถอาศัย Controller ในการควบคุมการจัดการ Packet Flow ได้หมดในระบบ ดังนั้นในการใช้ซอฟต์แวร์บนระบบเครือข่ายที่ถูกกำหนดหน้าที่ทำงานโดยซอฟต์แวร์ทำให้ระบบในเครือข่ายที่ถูกกำหนดโดยซอฟต์แวร์สามารถกำหนดหน้าที่ได้หลากหลายให้กับอุปกรณ์ในระบบเครือข่าย จึงมีความสามารถจัดการระบบเครือข่ายให้สามารถใช้งานได้อย่างมีประสิทธิภาพ จะส่งผลให้สามารถใช้ทรัพยากรที่มีอยู่อย่างจำกัดให้เกิดประโยชน์และมีประสิทธิภาพสูงสุดโดยที่มีราคาไม่แพง

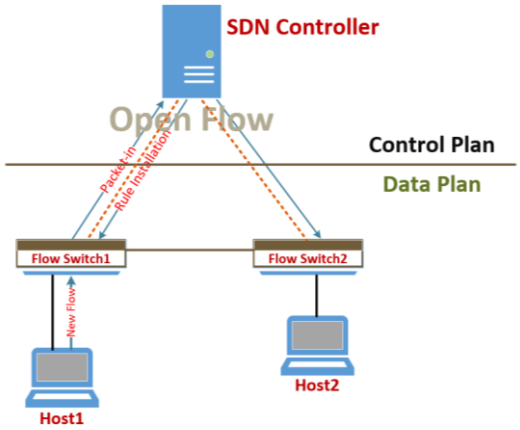
ในงานวิจัยนี้จึงมีแนวคิดในการศึกษาการพัฒนาการใช้งานเครือข่ายไร้สายด้วยการประยุกต์ใช้งานที่กำหนดด้วยซอฟต์แวร์ (WLAN Usability Improvement by Applying SDN) กรณีศึกษาการจัดสรร Bandwidth ในการใช้งานในเครือข่าย การจำกัดสิทธิ์ในการเข้าใช้บริการ File Transfer Protocol (FTP) และการกำหนดการใช้งาน ip address และ mac address ในระบบเครือข่ายที่กำหนดด้วยซอฟต์แวร์ เพื่อความปลอดภัยในระบบและสามารถความคุมการใช้งานระบบเครือข่ายที่มีทรัพยากรอยู่จำกัด จึงนำเทคโนโลยีเครือข่ายกำหนดด้วยซอฟต์แวร์ Software Defined Networks (SDN) เพื่อศึกษาถึงความสามารถในการบริหารจัดการระบบเครือข่ายที่กำหนดด้วยซอฟต์แวร์ ในการทำงานของระบบเครือข่ายมีความสามารถในการบริหารจัดการระบบเครือข่ายให้

มีประสิทธิภาพมากขึ้นและมีความสะดวกในการพัฒนาและสามารถนำแนวคิดและแนวปฏิบัติไปประยุกต์ใช้ในการทำงานต่อไป

แนวคิดและทฤษฎี ที่เกี่ยวข้อง

1. ทฤษฎี Software Defined Network (SDN)

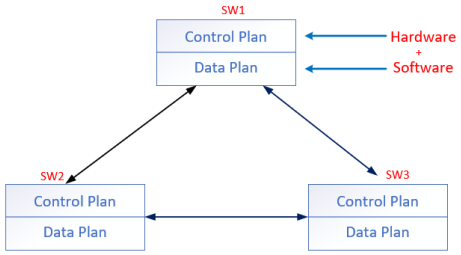
ระบบเครือข่ายที่กำหนดด้วยซอฟต์แวร์ SDN เป็นเทคโนโลยีในแนวคิดแบบใหม่ที่นำมาใช้ในการจัดการการทำงานของข้อมูลบนเครือข่าย โดยมีโครงสร้างในการทำงานบนระบบเครือข่ายที่แยกการทำงานออกจากกันเป็น 2 ส่วน คือ 1. ส่วนควบคุมการทำงาน หรือส่วน Control plane 2. ส่วนรับและส่งข้อมูล หรือ Data Plan



รูปภาพที่ 2.1 หลักการทำงานเครือข่ายกำหนดด้วยซอฟต์แวร์

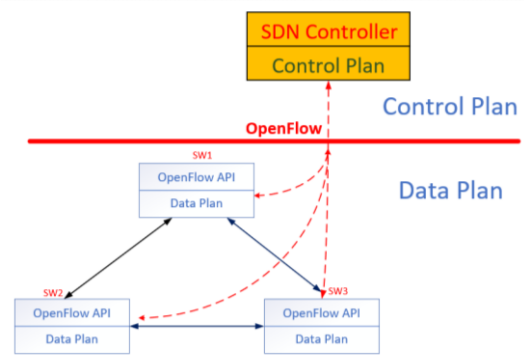
ในรูปแบบในการทำงานแบบ SDN ในการจัดการของระบบทั้งหมดจะอยู่ที่เครื่อง Controller ส่วนกลาง คือส่วน Control Plan สามารถส่วนจัดการกับ Packet Flow หรือ Data plane ที่อยู่บนอุปกรณ์เครือข่าย เช่น Switch แต่เดิมอุปกรณ์เครือข่ายและ Controller จะใช้ Open Flow protocol ในการกำหนดจัดการ Packet Flow ในการทำงาน ดังนั้นภายในเครือข่ายกำหนดด้วยซอฟต์แวร์ ผู้ที่ดูแลเครือข่ายสามารถอาศัยการจัดการเครือข่ายผ่าน Controller ในการบริหารจัดการ Packet Flow ที่เดินทางผ่านเครือข่าย และเพื่อการรวบรวมข้อมูลทรัพยากรบนระบบเครือข่ายทั้งหมด แนวคิดในการควบคุมการทำงานแบบการรวมการจัดการไว้ที่เครื่อง Controller เพียงเครื่องเดียวของเครือข่ายกำหนดด้วยซอฟต์แวร์ จัดเป็นสภาพแวดล้อมที่เหมาะสมต่อการจัดการเพื่อให้มีคุณภาพของการทำงานภายในเครือข่าย

ระบบเครือข่ายข่ายแบบดั้งเดิมก่อนที่จะมีแนวคิดในการนำเครือข่ายที่กำหนดการทำงานด้วยซอฟต์แวร์ SDN Controller มาใช้นั้น Router หรือ Switch ในส่วนที่ทำหน้าที่ควบคุมการเชื่อมต่อ (Control Plan) กับส่วนที่ทำหน้าที่ในการรับส่งข้อมูล (Data plane) จะทำหน้าที่รวมอยู่บนอุปกรณ์ตัวเดียวกัน โดย Control plane มีหน้าที่ควบคุมการเชื่อมต่อของอุปกรณ์บนเครือข่าย ดังภาพ



รูปภาพที่ 2.2 โครงสร้างของโครงข่ายแบบดั้งเดิม

ส่วนในการทำงานของแนวคิดแบบใหม่ของระบบเครือข่ายที่กำหนดด้วยซอฟต์แวร์ Software Define network เป็นการแยกส่วนทางกายภาพของอุปกรณ์ อย่าง Router, Switch ออกจากกันเป็น 2 ส่วนคือ Control plane กับ Data plane การควบคุมการทำงาน switch จะอยู่บน Controller ในส่วนของ Control plane ระบบเครือข่าย SDN สามารถรองรับการเชื่อมต่อของอุปกรณ์ได้หลากหลายยี่ห้อ ทำให้ผู้ที่มีความสนใจในงานวิจัยสามารถที่สร้างนวัตกรรมใหม่ ๆ ให้เกิดขึ้นกับอุปกรณ์ในเครือข่ายได้ เช่นการหาเส้นทางที่ดีที่สุดในการจัดการเครือข่าย แต่ถ้าเป็นเครือข่ายแบบดั้งเดิม นักวิจัยจะไม่สามารถพัฒนาวิธีการหาเส้นทางที่ดีที่สุดได้ดี นอกจากนี้อุปกรณ์บน Data plane จะเป็นอุปกรณ์ที่เป็นมาตรฐาน ผู้ใช้สามารถกำหนดให้ Data plane ทำงานเป็นอะไรก็ได้ตามต้องการ



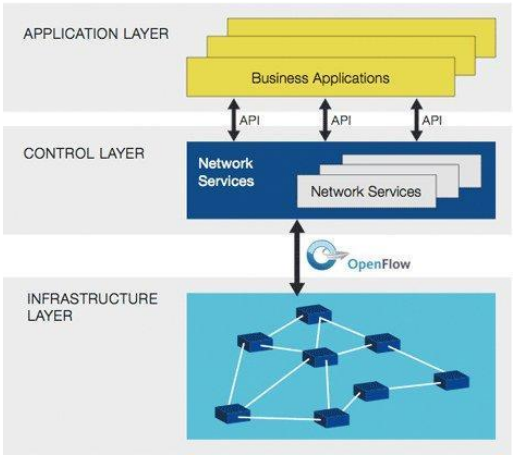
รูปภาพที่ 2.3 โครงสร้างของโครงข่ายแบบที่ใช้แนวคิดของ SDN

แนวคิด Software Defined Network หรือ SDN จะเข้ามาช่วยในการแก้ปัญหาเรื่องการตั้งค่าของอุปกรณ์ที่สามารถลดภาระในการทำงานของผู้ที่ดูแลระบบเครือข่าย ไม่ว่าอุปกรณ์เครือข่ายที่ใช้นั้นจะแตกต่างกันตามผู้ผลิตแต่ละบริษัทก็ตามสามารถจัดการร่วมกันได้ SDN เป็นเทคโนโลยีการจัดการเครือข่ายที่มีการแยกระบบควบคุม หรือระบบบริการเครือข่ายออกจาก Hardware โดยไม่สนใจความแตกต่างของอุปกรณ์เลย โดย Software Defined Networks ออกมาแก้ปัญหาตรงที่ผู้ดูแลเครือข่ายสามารถจัดการตั้งค่าอุปกรณ์ในจุดเดียวเมื่อในระบบเครือข่ายเข้ามาใหม่ผู้ดูแลระบบไม่จำเป็นต้องแก้ไขที่ตัวเครื่อง ระบบเครือข่ายแบบ Software Defined Networks จะทำหน้าที่ในการจัดการทำงานของระบบเครือข่ายให้เอง ผู้ดูแลไม่จำเป็นต้องไปยุ่งกับอุปกรณ์เครือข่ายเลย สิ่งที่ผู้ดูแลระบบต้องทำคือการกำหนด Policy ในการทำงานของอุปกรณ์หรือการจัดการระบบเครือข่ายแล้วระบบเครือข่ายแบบ SDN จะทำการกำหนดค่าให้ตามที่กำหนดกฎการทำงานให้ตามที่ต้องการเอง

รูปแบบของโครงสร้างในการทำงานของเครือข่ายแบบ SDN

Software Defined Networks นั้นสร้างตัวเองอยู่บนฐานของแนวคิดที่ว่า ส่วนที่ควบคุม และส่วนของข้อมูลแยกออกจากกันอย่างชัดเจน โดยมีส่วนของการจัดการเป็นตัวเชื่อมโดยใช้ API ในการเชื่อมต่อทั้งสองส่วนนี้เข้าด้วยกัน

Control plane เป็นส่วนที่จะทำหน้าที่ในการตัดสินใจว่า Packet ที่วิ่งอยู่ภายในระบบหรือเข้าถึงระบบแล้ว จะต้องจัดการส่งต่อหรือทำอย่างไรก็ต่อไป ส่วน Data plane คือส่วนที่จะอนุญาตหรือทำหน้าที่ในการส่งข้อมูลไปตามการตัดสินใจของ Control plane



รูปภาพที่ 2.4 การทำงานระบบเครือข่าย SDN

รูปภาพแสดงถึงเครือข่ายที่ใช้แนวคิด SDN ประกอบด้วย 3 ส่วนได้แก่ Application Control และ Infrastructure โดย Application เป็นส่วนที่เป็นตัวกำหนดการทำงานของอุปกรณ์ในเครือข่าย อย่างเช่น การหาเส้นทางในการเชื่อมต่อ ในส่วนของ SDN controller จะทำการดูสถานะของอุปกรณ์เครือข่าย ด้วยการส่งคำสั่งการทำงานไปยังอุปกรณ์ Switch ของเครือข่าย OpenFlow เพื่อรับคำสั่งในการทำงานจากแอปพลิเคชันผ่านทางมาตรฐานการเชื่อมต่อ Application Programming Interface (API) ได้แก่ Representational State Transfer (REST) API และ JAVA API เป็นต้น

การทำงานของ SDN เป็นการควบคุมการทำงานจากส่วนกลาง Controller เพื่อหาเส้นทางที่ดีที่สุดและสามารถเปลี่ยนแปลงเส้นทางให้เหมาะสมกับการทำงานของเครือข่าย โดยผู้ดูแลระบบสามารถเข้าจัดการเครือข่ายได้และกำหนดเส้นทางโดยใช้ซอฟต์แวร์ API จากการทำงานที่จุดเดียว ซึ่งทำให้สามารถใช้งานเครือข่ายได้อย่างเต็มที่และมีประสิทธิภาพ SDN เป็นระบบเครือข่ายที่มีรูปแบบการทำงานแยกตัวโครงสร้างในการทำงานหน้าที่ส่ง data packetออกจากโครงสร้างการตัดสินใจว่าจะให้ข้อมูลไหลไปในทิศทางใด โดยจะมี API ในการทำงานอย่าง Open Flow เป็นตัวกลางในการประสานระหว่างตัวโครงสร้างของเครือข่ายและส่วนควบคุมของระบบเครือข่าย ซึ่งผลที่ได้จากการใช้งาน SDN คือ

- ผู้ดูแลระบบไม่จำเป็นต้องสนใจอุปกรณ์เครือข่าย คือ อุปกรณ์จะมีระบบการควบคุมและจัดการเครือข่ายที่แยกออกไปจากกัน แม้ว่าอุปกรณ์จะมีทำงานที่แตกต่างกัน เมื่อมีการใช้งานระบบเครือข่ายที่กำหนดด้วยซอฟต์แวร์ SDN ไม่ว่าจะเป็นอุปกรณ์ของผู้ผลิตรายไหนก็สามารถควบคุมจัดการเครือข่ายให้ด้วยโปรโตคอล Open Flow

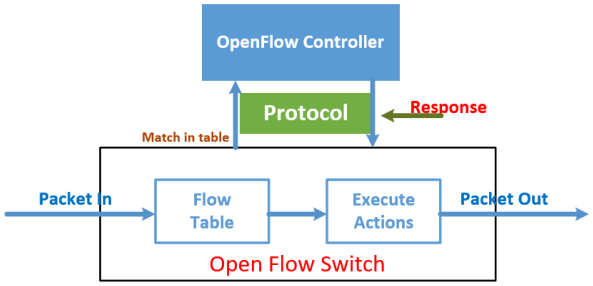
- ความยืดหยุ่นของระบบเครือข่ายแบบ SDN ในการจัดการกับอุปกรณ์ในเครือข่ายด้วยการจัดการระบบเครือข่ายที่กำหนดด้วยซอฟต์แวร์ จะไม่คำนึงอุปกรณ์ต้องมาจากบริษัทไหน ทำให้การวางแผนการปรับเปลี่ยนเครือข่ายทำได้ง่ายขึ้นและรวดเร็ว ลดการผูกขาด ลดค่าใช่จ่ายการใช้อุปกรณ์เครือข่าย

- ผู้ดูแลระบบเครือข่ายสามารถจัดการกับเครือข่ายให้มีประสิทธิภาพได้มากขึ้น เพราะว่า Controller สามารถที่จะกำหนดค่าในการทำงานผ่านซอฟต์แวร์ที่ควบคุมจากส่วนกลางและอุปกรณ์ Switch สามารถควบคุมผ่าน API ได้

2 OpenFlow

ในระบบเครือข่ายแบบ SDN OpenFlow เป็น Protocol ที่ถือว่าได้รับความนิยมเป็นอย่างมาก ในการใช้งานในระบบเครือข่ายแบบ SDN ซึ่ง OpenFlow เป็นการพัฒนาจากโครงการวิจัยของมหาวิทยาลัยสแตมฟอร์ด นำมาใช้งานเมื่อปี 2008 สามารถให้การทำงานของ switch มีความยืดหยุ่น

มากขึ้นในการทำงานบนเครือข่ายและมีประสิทธิภาพ Open flow เป็น protocol ที่ทำงานบนชั้น Data link Layer ใน Layer2 ในการทำงาน OpenFlow ทำหน้าที่ส่งต่อ Frame จาก Switch ไปยัง Switch ในเครือข่าย โดยที่ผู้ดูแลระบบไม่จำเป็นต้องเข้าไป config switch แต่ละตัวโดยตรง เพราะสามารถ config การจัดการเครือข่ายจากส่วนกลางที่เดียว โดย Controller

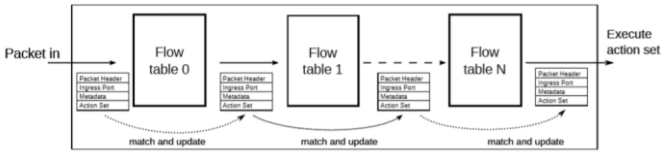


รูปภาพที่ 2.5 การทำงาน OpenFlow Protocol

OpenFlow ออกแบบให้ใช้งานได้กับ Switch ที่ไม่ยึดติดกับผู้ผลิตใด ๆอุปกรณ์ทั้งหมดในเครือข่ายสามารถทำงานร่วมกันผ่าน protocol เดียวกันได้ โดยมีองค์ประกอบหลักๆของ OpenFlow ดังนี้

2.1 Flow Table เป็นตารางบน Switch แต่ละตัว หน้าที่ของ Flow Table คือ เป็นที่เก็บ Rule หรือ Flow เมื่อมี Packet วิ่งเข้ามาที่ Switch ก็จะถูกนำไปเปรียบเทียบว่าตรงกับ Flow ไหน โดยในแต่ละ Flow ก็จะมี คำสั่ง เป็นของตัวเอง เช่น ปล่อยผ่าน Drop ทิ้ง หรือแก้ไขข้อมูลข้างในเป็นต้น ใน 1 Switch แต่ละตัวมีหลาย Table

2.2 Controller เป็นอุปกรณ์ที่ทำหน้าที่แก้ไขกฎการทำงานของ
Flow ใน switch บนเครือข่ายและเมื่อ Packet วิ่งเข้ามา Switch เมื่อกฎในการทำงานไม่ตรงกับ Flow ใดๆ Packet ที่ส่งมาจจะถูก Drop หรือ packet นั้นจะถูก forward มาที่ Controller อีกทีเพื่อให้ Controller ที่จะทำหน้าที่คล้าย hub ส่ง Packet ต่อไป

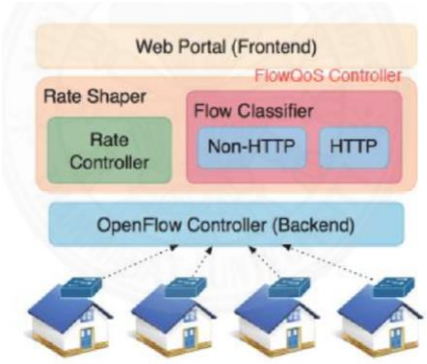


รูปภาพที่ 2.6 การทำงาน flow table

OpenFlow ทำหน้าที่ส่งต่อ Frame จาก Switch ไปยัง Switch ทำหน้าในการ Routing Protocol ที่อยู่บนอุปกรณ์เดียวกัน OpenFlow ทำงานการแยก Control Plan และ Data Plan ออกจากกัน Data plane จะยังคงทำงานอยู่บน Switch ส่วน Control plane ที่ทำหน้าที่เป็นซอฟต์แวร์จะทำการการตัดสินใจเลือกเส้นทางหรือ Routing Protocol จะทำงานอยู่บน Controller โดยที่ OpenFlow switch และ OpenFlow Controller จะทำการติดต่อกันด้วย OpenFlow protocol ด้วย Message ที่กำหนดไว้ เช่น การรับและส่ง packet การเปลี่ยนแปลง Forwarding table หรือการอ่านค่าสถานะของอุปกรณ์

3 การกำหนดแบนด์วิธให้แอพพลิเคชั่นโดยการทำงานของ FlowQoS Flow Quality of Service (FlowQoS)

Flow Quality of Service (FlowQoS) เป็นงานวิจัยของ (Seddiki M S, 2014) เป็นการนำเอาเทคโนโลยีเครือข่ายแบบที่กำหนดด้วยซอฟต์แวร์ SDN สำหรับศึกษาวิจัยการกำหนด แบนด์วิธให้กับแอพพลิเคชั่น (application) ซึ่งผู้ใช้งานสามารถกำหนดกฎให้กับแต่ละแอพลิเคชัน (application) โดยผ่าน Web Portal โดยการทำงานของ FlowQoS ได้แบ่งการทำงานออกเป็นสองส่วนหลักคือ ตัวคัด กรองโฟลว์ (Flow Classifier) และ ตัวจัดการแบนด์วิธ (Rate Controller) ดังรูป

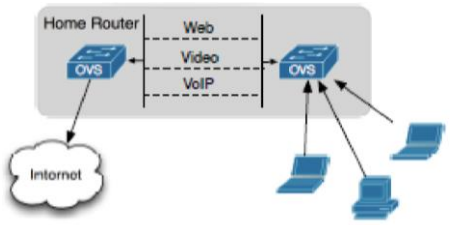


รูปภาพที่ 2.7 แสดงโครงสร้างของ Flow QoS

ตัวคัดกรองโฟลว์ (Flow Classifier) ทำหน้าที่เป็นตัวตรวจสอบ Packet Flow ที่ผ่านเข้ามาในอุปกรณ์เครือข่าย โดยตามหลักการทำงานของเครือข่าย SDN อุปกรณ์เครือข่ายจะคัดลอก Packet Flowไปให้กับ SDN controller ซึ่งขณะนั้นตัวคัดกรองโฟลว์จะทำการตรวจสอบ Packet Flow ที่เข้ามากับฐานข้อมูลที่เก็บข้อมูลของกฎจากการที่ผู้ใช้กำหนดผ่าน Web Portal โดยสิ่งที่ใช้ใน การตรวจสอบประกอบไปด้วย Source IP Address, Destination IP Address , Port Number และ Protocol เพื่อใช้ระบุ ชนิดของ Application เช่น Video, VoIP, Website, Game

ตัวจัดการแบนด์วิธจากที่ได้กล่าวข้างต้นว่าตัวคัดกรองโฟลว์ ทำหน้าที่ในการระบุการทำงานร่วมกันระหว่าง Packet Flow และ Application ดังนั้น ตัวจัดการ Packet จึงทำหน้าที่สำหรับจัดการแบนด์วิธ ให้กับ Packet Flow ตามที่ได้กำหนด ไว้จากผู้ใช้ โดยมีการระบุแต่ละ Flow ที่เข้ามาว่าเป็นApplication ชนิดใดด้วยตัวคัดกรองโฟลว์ (Flow Classifier) ซึ่งวิธีการจัดสรรแบนด์วิธของ FlowQoS นั้นจะเป็น ได้มีการติดตั้ง Open V Switch ขึ้นมาจำนวน 2 ตัว และมีช่องสื่อสารเชื่อมต่อกัน เพื่อเป็นตัวกำกับแบนด์วิธให้กับแต่ละ Application ด้วย

จากงานวิจัยนี้ทำให้เห็นจุดที่แตกต่างจากตัวจัดการทรัพยากร Bandwidth Fine-Grained Bandwidth Allocating Manager (FGBAM) คืองานวิจัยนี้ไม่สามารถทำงานร่วมกับ SDN controller หลากหลาย ชนิดได้ และ ไม่สามารถจัดสรร Bandwidth ตลอดเส้นทางที่แพ็กเก็ตข้อมูลผ่านได้



รูปที่ 2.8 แสดงโครงสร้างภายอุปกรณ์เครือข่ายที่ใช้สำหรับจัดสรร Bandwidth

4. การจัดสรรทรัพยากรแบนด์วิธแบบละเอียดภายในเครือข่าย SDN

นายวิศรุต คุ้มเงิน (2561) การจัดสรรทรัพยากรแบนด์วิธแบบละเอียดภายในเครือข่ายกำหนด ด้วยซอฟต์แวร์ตามความต้องการของผู้ใช้ (Fine-Grained Bandwidth Allocation in Software Defined Networks) bandwidthของช่องสื่อสารจัดเป็นมาตรวัดที่นิยมใช้ในการกำหนดคุณภาพของการให้บริการที่ต้องการ การจัดสรรทรัพยากรแบนด์วิธของแต่ละช่องสื่อสารแบบละเอียด กล่าวคือการจัดสรรให้หลาย Packet Flow (packet flow) สามารถใช้แบนด์วิธของช่องสื่อสารหนึ่งร่วมกันได้ตามความต้องการ ส่งผลให้สามารถใช้ทรัพยากรของเครือข่ายที่มีอยู่อย่างจำกัดได้อย่างมีประสิทธิภาพสูงสุดทั้งในสถานการณ์ปกติและในภาวะฉุกเฉิน สำหรับรับนโยบายความต้องการแบนด์วิธของแต่ละ Packet Flow จากผู้ดูแลเครือข่าย และส่วนด้านหลัง (back-end) ซึ่งใช้โปรโตคอลจัดการฐานข้อมูลของโอเพนวีสวิตช์ (Open vSwitch Database Management protocol) ในการบังคับใช้ นโยบายความต้องการแบนด์วิธ

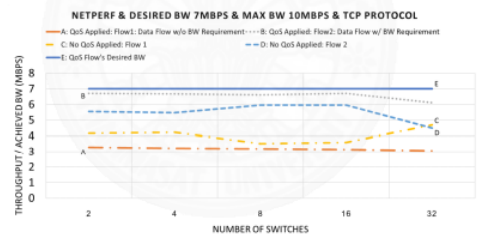
ผลการทดลองแสดงว่า ตัวจัดการทรัพยากร Bandwidth FGBAM ที่พัฒนาขึ้นเพื่อดำเนินการพัฒนาในการวิจัยนั้น มีคุณสมบัติที่พึงประสงค์ได้แก่

1. สามารถจัดสรรแบนด์วิธของแต่ละช่องสื่อสารแบบละเอียด (fine- grained bandwidth allocation) ส่งผลให้หลาย Packet Flow สามารถใช้งานแบนด์วิธของช่องสื่อสารร่วมกันได้ โดยเปอร์เซ็นต์ความคลาดเคลื่อนระหว่างแบนด์วิธที่ Packet Flow ได้รับจริงจากการจัดสรรแบนด์วิธด้วยFGBAM กับแบนด์วิธที่ Packet Flow ต้องการคิดเป็นค่าเฉลี่ยเท่ากับ 37.75 เปอร์เซ็นต์ โดยมีรายละเอียด ดังนี้

Number of Switches	Throughput / Achieved BW (Mbps)				
	When QoS Applied			When No QoS Applied (Used for Referencing)	
	Flow 1: Data Flow without BW Requirement	Flow 2: Data Flow with BW Requirement	Residual BW (%)	Flow 1	Flow 2
2	3.95	5.74	-4.33	4.16	5.54
4	3.97	5.74	-4.35	4.22	5.47
8	4.04	5.67	-5.54	3.53	5.95
16	3.94	5.29	-12.06	3.60	5.96
32	3.94	5.74	-4.33	4.69	4.49

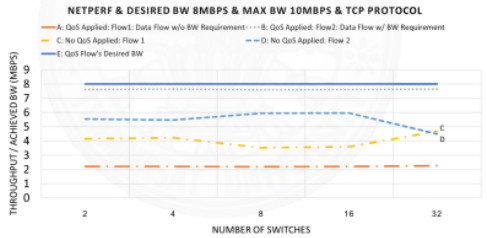
รูปที่ 2.9 แสดงกราฟโดยส่งแพ็กเก็ตยูดีพีด้วยเน็ตเพิร์ฟ ภายในช่องสื่อสารที่มี แบนด์วิธสูงสุด 10 Mbps และมีความต้องการใช้แบนด์วิธเท่ากับ 6 Mbps

Number of Switches	Throughput / Achieved BW (Mbps)				
	When QoS Applied			When No QoS Applied (Used for Referencing)	
	Flow 1: Data Flow without BW Requirement	Flow 2: Data Flow with BW Requirement	Residual BW (%)	Flow 1	Flow 2
2	3.29	6.7	-4.29	4.16	5.54
4	3.22	6.67	-4.75	4.22	5.47
8	3.20	6.61	-5.60	3.53	5.95
16	3.15	6.70	-4.30	3.60	5.96
32	3.06	6.11	-12.71	4.69	4.49



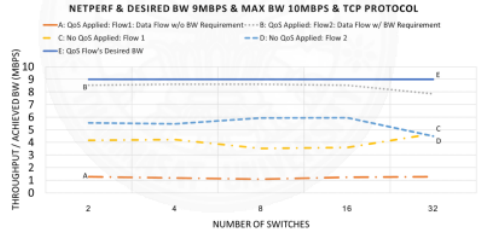
รูปที่ 2.10 แสดงกราฟโดยส่งแพ็กเก็ตยูดีพีด้วยเน็ตเพิร์ฟ ภายในช่องสื่อสารที่มี แบนด์วิธ สูงสุด 10 Mbps และมีความต้องการใช้ แบนด์วิธ เท่ากับ 7 Mbps

Number of Switches	Throughput / Achieved BW (Mbps)				
	When QoS Applied			When No QoS Applied (Used for Referencing)	
	Flow 1: Data Flow without BW Requirement	Flow 2: Data Flow with BW Requirement	Residual BW (%)	Flow 1	Flow 2
2	2.2	7.61	-4.88	4.16	5.54
4	2.21	7.66	-4.38	4.22	6.47
8	2.19	7.60	-4.99	3.53	5.95
16	2.21	7.61	-4.85	3.60	5.96
32	2.25	7.65	-4.40	4.69	4.49



รูปที่ 2.11 แสดงกราฟโดยส่งแพ็กเก็ตยูดีพีด้วยเน็ตเพิร์ฟ ภายในช่องสื่อสารที่มีแบนด์วิธสูงสุด 10 Mbps และมีความต้องการใช้แบนด์วิธเท่ากับ 8 Mbps

Number of Switches	Throughput / Achieved BW (Mbps)				
	When QoS Applied			When No QoS Applied (Used for Referencing)	
	Flow 1: Data Flow without BW Requirement	Flow 2: Data Flow with BW Requirement	Residual BW (%)	Flow 1	Flow 2
2	1.28	8.53	-5.22	4.16	5.54
4	1.18	8.61	-4.36	4.22	5.47
8	1.09	8.61	-4.35	3.53	5.95
16	1.24	8.53	-5.25	3.60	5.96
32	1.29	7.85	-12.74	4.69	4.49



รูปที่ 2.12 แสดงกราฟโดยส่งแพ็กเก็ตยูดีพีด้วยเน็ตเพิร์ฟ ภายในช่องสื่อสารที่มีแบนด์วิธสูงสุด 10 Mbps และมีความต้องการใช้แบนด์วิธเท่ากับ 9 Mbps

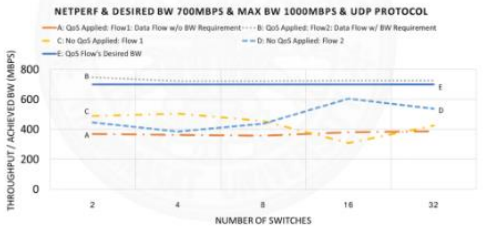
2. การใช้งานเอฟจีแบมเพื่อจัดสรรทรัพยากรแบนด์วิธภายในเครือข่าย SDN มีความสะดวกเพราะไม่ต้องปรับเปลี่ยนการตั้งค่าของอุปกรณ์เครือข่ายและ Controller แสดงว่า เอฟจีแบมสามารถจัดสรร แบนด์วิธ ของแต่ละ Switch Port ให้กับ TCP Packet Flow ที่ใช้งานพอร์ตนั้น ๆ ร่วมกันได้จริง อย่างไรก็ตาม *Throughput* หรือ Bandwidth จริง Packet Flow ที่ FGBAM จัดสรรแบนด์วิธให้

นั้นได้รับยังต่ำกว่าค่าแบนด์วิธที่ Packet Flow ต้องการ ซึ่งมีความคลาดเคลื่อนเฉลี่ย คิดเป็น 3.86 เปอร์เซ็นต์ สภาพแวดล้อมที่ตัวจัดการทรัพยากรแบนด์วิธ FGBAM สามารถจัดสรรแบนด์วิธได้เหมาะสมสำหรับแพ็กเก็ตที่เป็นยูดีพีหรือจัดสรร แบนด์วิธได้ใกล้เคียงกับแบนด์วิธที่ Packet Flow ต้องการมากที่สุด คือ สภาพแวดล้อมที่มีการกำหนดแบนด์วิธเท่ากับ 900 Mbps โดยมีค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อนระหว่างแบนด์วิธที่ Packet Flow ได้รับจริงจากการจัดสรรแบนด์วิธด้วยตัวจัดการทรัพยากรแบนด์วิธ FGBAM กับแบนด์วิธที่ Packet Flow เท่ากับ 1.10 เปอร์เซ็นต์ ดังนี้

Number of Switches	Throughput / Achieved BW (Mbps)				
	When QoS Applied			When No QoS Applied (Used for Referencing)	
	Flow 1: Data Flow without BW Requirement	Flow 2: Data Flow with BW Requirement	Residual BW (%)	Flow 1	Flow 2
2	466.29	650.65	8.44	488.18	446.38
4	465.50	649.38	8.23	504.94	386.03
8	446.28	623.31	3.88	456.14	438.47
16	467.48	645.25	7.54	308.09	604.99
32	480.74	644.41	7.40	425.12	538.22

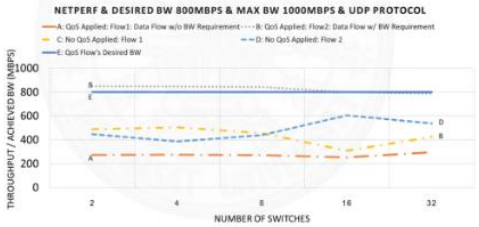
รูปที่ 2.13 แสดงกราฟผลการทดลองโดยส่งแพ็กเก็ตยูดีพีด้วยเน็ตเพิร์ฟ ภายในช่องสื่อสารที่มีแบนด์วิธ สูงสุด 1,000 Mbps และมีความต้องการใช้ แบนด์วิธ เท่ากับ 600 Mbps

Number of Switches	Throughput / Achieved BW (Mbps)				
	When QoS Applied			When No QoS Applied (Used for Referencing)	
	Flow 1: Data Flow without BW Requirement	Flow 2: Data Flow with BW Requirement	Residual BW (%)	Flow 1	Flow 2
2	369.57	747.50	6.79	488.18	446.38
4	363.32	722.77	3.25	504.94	386.03
8	357.72	723.32	3.33	456.14	438.47
16	381.39	725.32	3.62	308.09	604.99
32	387.21	726.76	3.82	425.12	538.22



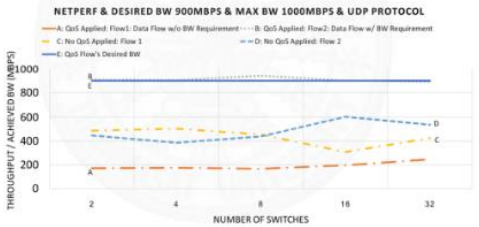
รูปที่ 2.14 แสดงกราฟผลการทดลองโดยส่งแพ็กเก็ตยูดีพีด้วยเน็ตเพิร์ฟ ภายในช่องสื่อสารที่มีแบนด์วิธสูงสุด 1,000 Mbps และมีความต้องการใช้แบนด์วิธเท่ากับ 700 Mbps

Number of Switches	Throughput / Achieved BW (Mbps)				
	When QoS Applied			When No QoS Applied (Used for Referencing)	
	Flow 1: Data Flow without BW Requirement	Flow 2: Data Flow with BW Requirement	Residual BW (%)	Flow 1	Flow 2
2	273.54	849.21	6.15	488.18	446.38
4	274.79	845.71	5.71	504.94	386.03
8	270.33	841.50	5.19	456.14	438.47
16	253.11	799.60	-0.05	308.09	604.99
32	295.12	787.79	-1.53	425.12	538.22



รูปที่ 2.15 แสดงกราฟผลการทดลองโดยส่งแพ็กเก็ตยูดีพีด้วยเน็ตเพิร์ฟ ภายในช่องสื่อสารที่มีแบนด์วิธสูงสุด 1,000 Mbps และมีความต้องการใช้แบนด์วิธเท่ากับ 800 Mbps

Number of Switches	Throughput / Achieved BW (Mbps)				
	When QoS Applied			When No QoS Applied (Used for Referencing)	
	Flow 1: Data Flow without BW Requirement	Flow 2: Data Flow with BW Requirement	Residual BW (%)	Flow 1	Flow 2
2	170.78	910.60	1.18	488.18	446.38
4	175.53	904.98	0.55	504.94	386.03
8	167.67	941.84	4.65	456.14	438.47
16	195.83	901.66	0.18	308.09	604.99
32	247.73	890.41	-1.07	425.12	538.22



รูปที่ 2.16 แสดงกราฟผลการทดลองโดยส่งแพ็กเก็ตยูดีพีด้วยเน็ตเพิร์ฟ ภายในช่องสื่อสารที่มีแบนด์วิธ สูงสุด 1,000 Mbps และมีความต้องการใช้แบนด์วิธเท่ากับ 900 Mbps

3. เอฟจีแบมสามารถทำงานร่วมกับคอนโทรลเลอร์ได้หลากหลายชนิด FGBAM สามารถรองรับเครือข่ายที่กำหนดด้วยซอฟต์แวร์ ประกอบด้วยอุปกรณ์เครือข่ายที่ใช้งาน Open Flow Protocol หลากหลายเวอร์ชั่นได้

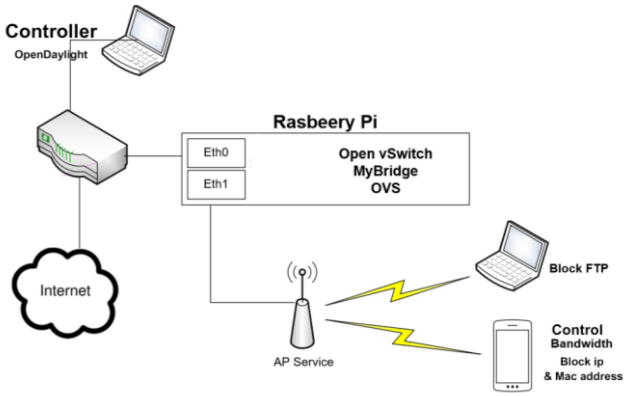
ตารางที่ 2.1 ผลการทดลองจัดการทรัพยากรแบนด์วิธ FGBAM

สถานการณ์	วินาทีที่โฟลว์มอดเมสเสจออกจากเอสดีเอ็นคอนโทรลเลอร์ (วินาที)	วินาทีที่อุปกรณ์เครือข่ายตอบกลับเอสดีเอ็นคอนโทรลเลอร์เมื่อได้รับโฟลว์มอดเมสเสจ (วินาที)	ความหน่วงที่เกิดขึ้นกับโฟลว์มอดเมสเสจ (มิลลิวินาที)
เอฟจีแบมต้องจัดสรร แบนด์วิธที่พอร์ตของอุปกรณ์เครือข่าย	101.9539	101.9956	41.7261
เอฟจีแบมไม่ต้องจัดสรร แบนด์วิธที่พอร์ตของอุปกรณ์เครือข่าย	28.4968	28.5357	38.8527
ไม่มีเอฟจีแบมทำงาน ร่วมด้วย	47.4914	47.4934	1.9984

ตารางที่ 2.1 ผลการทดลองประกอบไปด้วยสามสถานการณ์ คือ สถานการณ์ภายในเครือข่ายที่กำหนดด้วยซอฟต์แวร์ โดยที่มีตัวจัดการทรัพยากรแบนด์วิธ FGBAM มีการทำงานร่วมด้วย และFGBAM ต้องจัดสรร แบนด์วิธ ที่ Port ของอุปกรณ์เครือข่าย สถานการณ์ที่ภายในเครือข่ายที่กำหนดด้วยซอฟต์แวร์ที่มีตัวจัดการทรัพยากร แบนด์วิธ FGBAM มีการทำงานร่วมอยู่ด้วย และ FGBAM ไม่ต้องจัดสรร แบนด์วิธที่ Port ของอุปกรณ์เครือข่าย สถานการณ์ที่ภายในเครือข่ายที่กำหนดด้วยซอฟต์แวร์ไม่มี FGBAM ทำงานร่วมด้วย

แนวทางการดำเนินงาน

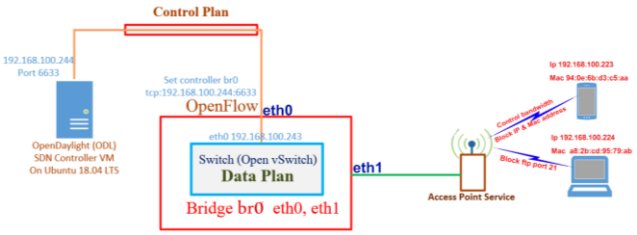
ในงานวิจัยนี้ประกอบด้วย SDN Controller ใช้ Open Daylight (ODL) ทำหน้าที่เป็น SDN Controller ติดตั้งบน Linux Ubuntu 18.04 LTS และระบบเครือข่ายแบบไร้สายใช้อุปกรณ์ Router เชื่อมต่อเข้า Interface บน Virtual Switch ที่ติดตั้งบน Raspberry Pi3 Model B ทำหน้าที่เป็น OpenFlow เพื่อรับคำสั่งจาก Controller



รูปที่ 3.1 โทโปโลยีระบบเครือข่ายที่ใช้ในการทดลอง

โดยในงานวิจัยเป็นการศึกษา Open Flow ในการสร้างกฎควบคุมการทำงานในเครือข่ายไร้สายที่กำหนดด้วยซอฟต์แวร์ กรณีศึกษาการจัดสรรแบนด์วิธ, การกำหนดสิทธิ์ในการใช้งาน File Transfer Protocol และการกำหนดไม่ให้ใช้งาน ip address และ mac address ในระบบเครือข่ายที่กำหนดซอฟต์แวร์

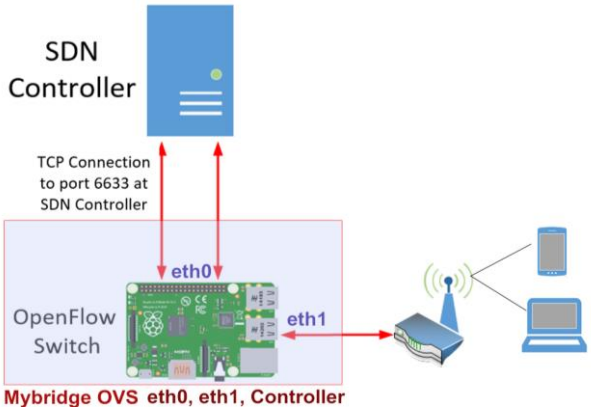
โดยใช้ OpenDaylight (ODL) ที่ทำหน้าที่เป็น SDN controller ในการติดตั้งต้องทำการสร้างระบบปฏิบัติการ Linux Ubuntu version 18.04.6 LTS 64 bit เพื่อรองรับการติดตั้ง OpenDaylight หลังจากนั้นติดตั้ง OpenDaylight (ODL) (karaf-0.8.4) บน Linux Ubuntu ดังกล่าวข้างต้น ในการสร้างระบบปฏิบัติการ Linux Ubuntu 18.04 LTS 64 bit โดยจะทำการติดตั้งบน Virtual Machine ที่ชื่อ VMware Workstation 16 Player ด้วยการสร้าง VM ใหม่ขึ้นมา จากไฟล์ Ubuntu_18.04_LinuxVMImages.ovf กำหนดค่า Edit virtual machine setting ของ Network adapter โดยเลือกที่ bridged networking เพื่อใช้การเชื่อมต่อ Internet ให้ OpenDaylight และ VMware Machine สามารถติดต่อกับเครื่อง Host ที่อยู่ภายนอก และ OpenFlow Manager ได้บน Control Plan หลังจากที่ได้สร้าง Linux Ubuntu 18.04 LTS เสร็จเรียบร้อย ทำการติดตั้ง OpenDayLight (ODL) โดยทำการการติดตั้ง Java run-time environment ด้วย



รูปภาพที่ 3.2 ภาพรวมการการเชื่อมต่อการทำงานระบบเครือข่ายในการทดสอบ

1 การติดตั้ง Software ในส่วน Data Plan

Data Plan โดยใช้ Hardware เป็นบอร์ด Raspberry Pi3 Model B ในตัวบอร์ดจะรองรับการติดตั้งระบบปฏิบัติการ Linux Raspbian โดยจะมีติดตั้งระบบปฏิบัติการผ่านอุปกรณ์หน่วยความจำ Micro SD Card ในระบบงานจะติดตั้ง Open Virtual Switch (OVS) เป็น Switch OpenFlow ทำหน้าที่รับคำสั่งจาก Controller



รูปภาพที่ 3.2 การติดตั้ง Raspberry Pi3 Model B

ทำการติดตั้ง Open Virtual Switch (OVS)

1. ทำการติดตั้ง Open vSwitch ต้องทำการสร้างระบบปฏิบัติการ Raspbian บนตัวอุปกรณ์บอร์ด Raspberry Pi จากนั้นทำการ Update และ Upgrade

#sudo apt-get update

#sudo apt-get upgrade

การ Update และ upgrade เป็นการตรวจหาโปรแกรมใหม่จาก Repository หรือ จาก Server และดาวโหลดติดตั้ง ไหม่ลงไปใน Raspberry pi

2. เมื่อติดตั้งระบบปฏิบัติการเสร็จ ทำการติดตั้ง Open vSwitch โดยการดาวโหลดไฟล์ openvswitch-2.7.0.tar.gz แล้วทำการแตกไฟล์การติดตั้ง มีขั้นตอนการติดตั้ง ดังนี้

#apt-get install python-simplejson python-qt4 libssl-dev python-twisted-conch automake autoconf gcc uml-utilities libtool build-essential pkg-config

#apt-get install -y linux-headers-4.9.0-6-rpi

#cd openvswicth-2.7.0

#./configure --with-linux=/lib/modules/4.9.0-6-rpi/build

#make & make install

#touch /usr/local/etc/ovs-vswitchd.conf

#mkdir -p /usr/local/etc/openvswitch

#ovsdb-tool create /usr/local/etc/openvswitch/conf.db vswitchd/vswitch.ovsschema

#nano script

```
ovsdb-server --remote=punix:/usr/local/var/run/openvswitch/db.sock \
    --remote=db:Open_vSwitch,Open_vSwitch,manager_options \
    --private-key=db:Open_vSwitch,SSL,private_key \
    --certificate=db:Open_vSwitch,SSL,certificate \
    --bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert \
    --pidfile --detach
    ovs-vswitchd --pidfile --detach
    ovs-vsctl --no-wait init
    ovs-vsctl show

    #chmod +x script
    #./script
```

การทำ Bridge Port เชื่อมเครือข่ายเข้าด้วยกัน ดังนี้

- ทำการสร้าง Bridge Port ชื่อ bro #ovs-vsctl add-br br0
- ทำการเพิ่ม eth0 เข้ายัง Bridge Port ชื่อ bro #ovs-vsctl add-port br0 eth0
- ทำการเพิ่ม eth1 เข้ายัง Bridge Port ชื่อ bro #ovs-vsctl add-port br0 eth1

ทำการ Set-controller เพื่อให้ Controller สามารถเชื่อมต่อในการทำงานกับ Openflow โดย Bridge เข้ากับหมายเลข IP 192.168.100.244 ที่ Controller โดยผ่าน Port 6633

#ovs-vsctl set-controller br0 tcp:192.168.100.244:6633

2 การติดตั้ง Software ในส่วน Control Plan

Control Plan ส่วนนี้ถูกติดตั้งอยู่ภายใต้ Virtual Machine ที่ถูกจำลองขึ้นมาด้วยโปรแกรม VMware Workstation 16 Player ซอฟต์แวร์ที่นำมาใช้ติดตั้ง คือ Open Daylight (karaf-0.8.4) ทำหน้าที่เป็น Controller ซึ่ง OpenDaylight (ODL) เป็นแพลตฟอร์ม SDN ทั่วไป**และเป็น** Open Source เป็นส่วนควบคุมเส้นทางของข้อมูลทำหน้าที่**เป็น**ตัวควบคุม OpenFlow **ซึ่งจะเป็น**โปรโตคอลที่อนุญาตให้ส่งผ่านข้อมูลไปที่สวิตช์เครือข่ายว่าจะให้ส่ง Packet ไปที่ใด

2.1 ติดตั้ง OpenDaylight (ODL):Controller

เมื่อทำการติดตั้ง Virtual Machine แล้ว สร้างระบบปฏิบัติการ Linux Ubuntu 18.0.4 LTS เพื่อทำการติดตั้ง Controller โดยจะมีขั้นตอนการติดตั้งดังนี้ จะต้องทำการติดตั้ง Java run-time environment ก่อน

ติดตั้ง Java script 8

sudo apt-get -y install openjdk-8-jre

ทำการติดตั้ง OpenDaylight (karaf-0.8.4.zip)

ดาวโหลด OpenDaylight (karaf-0.8.4.zip) เพื่อเตรียมติดตั้ง

#wget https://nexus.Open Daylight.org/content/repositories/OpenDaylight.release/org /Open Daylight/integration/karaf/0.8.4/karaf-0.8.4.zip

สร้าง Directory ชื่อ karaf ที่ /usr/local/karaf สำหรับเก็บไฟล์ แล้วทำการแตกไฟล์ไปเก็บไว้แล้วทำการติดตั้ง

sudo unzip /usr/local/karaf/karaf-0.8.4.zip -d /usr/local/karaf/

#sudo update-alternatives --install /usr/bin/karaf karaf /usr/local/karaf/karaf-0.8.4/bin/karaf 1

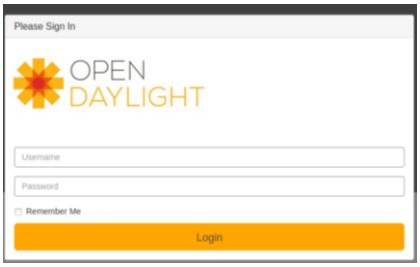
2.2 ขั้นตอนพัฒนาระบบเครือข่ายไร้สายกำหนดด้วยซอฟต์แวร์

การเปิดใช้งาน Open Daylight

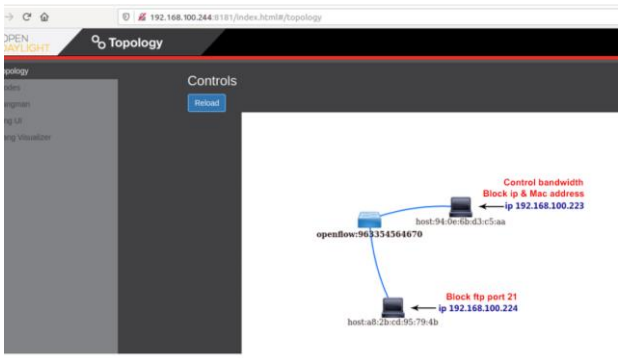
เมื่อติดตั้ง Controller เสร็จ ทำการ Run เพื่อการทำงานด้วยคำสั่ง #sudo –E karaf

รูปภาพที่ 3.3 แสดงผลการเปิดใช้งาน OpenDaylight

เข้าใช้งานด้วย Browser Chrome เข้าไปที่ URL ที่ชื่อ http://192.168.100.244:8181/index.html จะเห็นเครือข่ายที่แสดง ดังรูป



รูปภาพที่ 3.4 เข้าสู่หน้า Login



รูปภาพที่ 3.5 แสดงการเชื่อมต่อโทโปโลยีเครือข่ายแบบ SDN ในการทดลอง

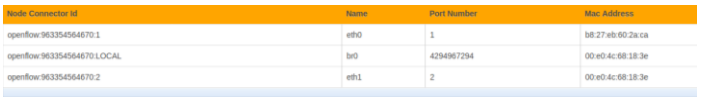
จากรูปภาพแสดงให้เห็นถึงการเชื่อมต่อของโทโปโลยีตามที่กำหนดไว้ เพื่อการทดสอบในการทำงานของระบบ การกำหนดรูปแบบโทโปโลยีในการทดสอบ จะติดตั้ง switch openflow จำนวน 1 ตัว และอุปกรณ์ในการทดสอบจำนวน 2 เครื่อง คือ

Node Connector Id	Name	Port Number	Mac Address
openflow:963354564670:1	eth0	1	b8:27:eb:60:2a:ca
openflow:963354564670:LOCAL	br0	4294967294	00:e0:4c:68:18:3e
openflow:963354564670:2	eth1	2	00:e0:4c:68:18:3e

รูปภาพที่ 3.6 แสดง node การเชื่อมต่อในระบบ

จากดังรูปจะเห็นว่ามี Switch จำนวน 1 ตัว ชื่อเป็น openflow:963354564670 เป็น nod connector id ในการเชื่อมต่อที่ interface eth0 มีการเชื่อมต่อจำนวน 1 เครื่อง คือเครื่องที่ทำหน้าที่เป็น SDN Controller และ interface eth1 มีการเชื่อมต่อจำนวน port ออกไปจำนวน 2 เครื่อง คือเครื่อง host ที่ใช้เพื่อทำการทดสอบ โดยทั้ง eth0 และ eth1 จะมีการ Bridge เข้าหากันโดยผ่าน interface port br0 เพื่อให้ทั้ง Controller และ เครื่อง host สามารถเชื่อมต่อการทำงานในระบบได้

จากรูปภาพแสดงให้เห็นถึงการเชื่อมต่อของโทโปโลยีตามที่กำหนดไว้ เพื่อการทดสอบในการทำงานของระบบ การกำหนดรูปแบบโทโปโลยีในการทดสอบ จะติดตั้ง switch openflow จำนวน 1 ตัว และอุปกรณ์ในการทดสอบจำนวน 2 เครื่อง คือ



Node Connector Id	Name	Port Number	Mac Address
openflow:963354564670:1	eth0	1	b8:27:eb:60:2a:ca
openflow:963354564670:LOCAL	br0	4294967294	00:e0:4c:68:18:3e
openflow:963354564670:2	eth1	2	00:e0:4c:68:18:3e

รูปภาพที่ 3.7 แสดง node การเชื่อมต่อในระบบ

จากดังรูปจะเห็นว่ามี Switch จำนวน 1 ตัว ชื่อเป็น openflow:963354564670 เป็น nod connector id ในการเชื่อมต่อที่ interface eth0 มีการเชื่อมต่อจำนวน 1 เครื่อง คือเครื่องที่ทำหน้าที่เป็น SDN Controller และ interface eth1 มีการเชื่อมต่อจำนวน port ออกไปจำนวน 2 เครื่อง คือเครื่อง host ที่ใช้เพื่อทำการทดสอบ โดยทั้ง eth0 และ eth1 จะมีการ Bridge เข้าหากันโดยผ่าน interface port br0 เพื่อให้ทั้ง Controller และ เครื่อง host สามารถเชื่อมต่อการทำงานในระบบได้

ผลการดำเนินงาน

การศึกษาการพัฒนาการใช้งาน WLAN ด้วยการประยุกต์ใช้งาน SDN (WLAN Usability Improvement by Applying SDN) ในการทดลองผู้วิจัยได้ทำการออกแบบการทดลองเพื่อศึกษาความสามารถในการทำงานของ SDN Controller และความสามารถในการทำงานร่วมกับ

OpenFlow บน Open vSwitch ในระบบเครือข่ายแบบไร้สาย นั้น ในกรณีการศึกษาการทดลองระบบจะมีการพัฒนาเพื่อศึกษาการทำงานอยู่ 4 รูปแบบดังนี้

การทดลองแบบที่ 1 การทำงานของระบบเครือข่ายแบบไร้สายที่กำหนดด้วยซอฟต์แวร์ กรณีศึกษาการตั้งกฎเพื่อจัดสรรแบนด์วิธด้วยเครือข่ายที่กำหนดด้วยซอฟต์แวร์

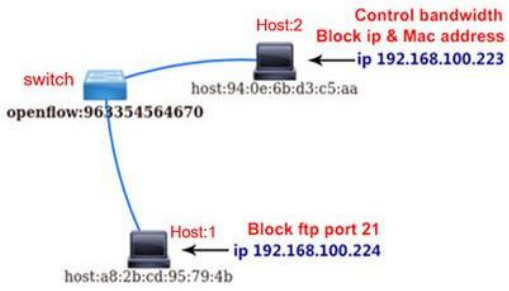
การทดลองแบบที่ 2 การทำงานของระบบเครือข่ายแบบไร้สายที่กำหนดด้วยซอฟต์แวร์ กรณีศึกษาการกำหนดกฎการทำงาน ไม่อนุญาตให้ใช้งานหมายเลข IP ตามที่กำหนดด้วยเครือข่ายที่กำหนดด้วยซอฟต์แวร์

การทดลองแบบที่ 3 การทำงานของระบบเครือข่ายแบบไร้สายที่กำหนดด้วยซอฟต์แวร์ กรณีศึกษาการกำหนดกฎการทำงาน ไม่อนุญาตให้ใช้งานหมายเลข Mac address ตามที่กำหนดด้วยเครือข่ายที่กำหนดด้วยซอฟต์แวร์

การทดลองแบบที่ 4 การทำงานของระบบเครือข่ายแบบไร้สายที่กำหนดด้วยซอฟต์แวร์ โดยกรณีศึกษาการตั้งกฎในการทำงานเพื่อกำหนดสิทธิ์ในการอนุญาตหรือไม่อนุญาตในเข้าถึงข้อมูลด้วย File Transfer Protocol (FTP) ด้วยการปิดหรือเปิดการให้บริการใช้งาน port 21 ด้วยเครือข่ายที่กำหนดด้วยซอฟต์แวร์

1. รูปแบบการกำหนดโทโปโลยีในการทดสอบ

แสดงการเชื่อมต่ออุปกรณ์เครือข่ายที่กำหนด บนเครือข่าย SDN ประกอบด้วยอุปกรณ์กระจายสัญญาณสวิตซ์ Open V switch ติดตั้งบนบอร์ด Raspberry Pi จำนวน 1 เครื่องและ host จำนวน 2 เครื่อง



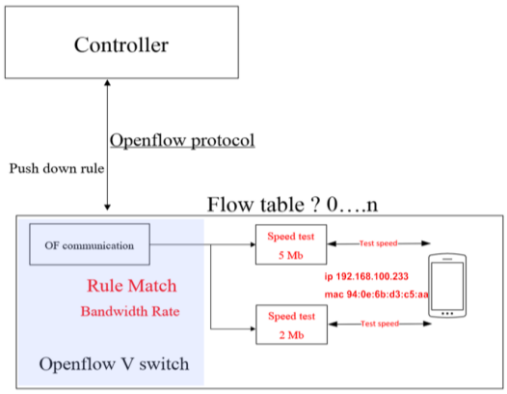
รูปภาพที่ 4.1 แสดงการเชื่อมต่อโทโปโลยีเครือข่ายแบบ SDN

โดยแต่ละขั้นตอนจะมีแนวทางในการทดสอบการทำงานต่างๆ ผ่านเครือข่าย SDN Controller ตามรูปแบบการทำงานที่ได้กำหนดไว้ ในการกำหนดคำสั่งการทดสอบ ให้ host ในการทำงานตามที่ต้องการนั้น อย่างแรกคือการกำหนดกฎการทำงานโดย Controller เพื่อ add Flow ไปยัง

Flow Table ของ switch เพื่อทำงานตามที่ได้ออกแบบไว้ ในการ add Flow จะใช้แอปพลิเคชั่น Postman เป็นแอปพลิเคชั่นที่ใช้สำหรับการพัฒนา API ที่ใช้สำหรับการทดสอบการทำงาน คือ สำหรับการทดสอบ API และ การ Test แบบ Automated ในการใช้งาน Postman

2. การทดสอบการปรับปรุงประสิทธิภาพในการจัดสรรแบนด์วิดธ์

การจัดสรรแบนด์วิดธ์ เป็นการลดทราฟฟิกบนเครือข่ายให้มีประสิทธิภาพในการทำงานมากขึ้น ผู้วิจัยได้ดำเนินการทดสอบ ด้วยการกำหนดค่าแบนด์วิดธ์สูงสุดของช่องทางการสื่อสารในการใช้งาน ในการทดสอบผู้วิจัยได้ทำการเพิ่มกฎการทำงานบน Controller เพื่อกำหนดค่าการใช้งาน Bandwidth ให้กับ host2 บนเครือข่าย ซึ่งอุปกรณ์ที่นำมาเพื่อการทดสอบครั้งนี้ จะใช้เครื่องโทรศัพท์มือถือเคลื่อนที่รองรับการเชื่อมต่อแบบไร้สายนำมาทดสอบการทำงานในกรณีศึกษาการทำงานของเครือข่ายที่กำหนดด้วยซอฟต์แวร์ จำนวน 1 เครื่อง โดยมีการกำหนดกฎในจัดสรรแบนด์วิดธ์ที่แตกต่างกัน ดังนี้



รูปภาพที่ 4.2 รูปแบบการทำงานเมื่อเพิ่มกฎในการจัดสรรแบนด์วิด

โดยกำหนดกฎในการทำการทำสอบ ดังนี้

การทดสอบแบบที่ 1 กำหนดให้ Host2 หมายเลข IP 192.168.100.223, Mac address host:94:0e:6b:d3:c5:aa กำหนดกฎในการทำงาน ให้ใช้งานแบนด์วิดธ์ในการใช้งานได้ไม่เกิน 5 Mbps

การทดสอบแบบที่ 2 กำหนดให้ Host2 กำหนดกฎในการทำงาน ให้ใช้งานแบนด์วิดธ์ในการใช้งานได้ไม่เกิน 2 Mbps

จากนั้นสร้างกฎการทำงานด้วยการ PUT ข้อมูลลงไปที่ flow table โดย Controller เพื่อกำหนดการใช้งานโดยใช้ Postman

การสร้างกฎคำสั่งให้ทำงาน ด้วยการ PUT ข้อมูลลงไปที่ flow table

ต้องทำการ add meter id ไปที่ openflow:963354564670

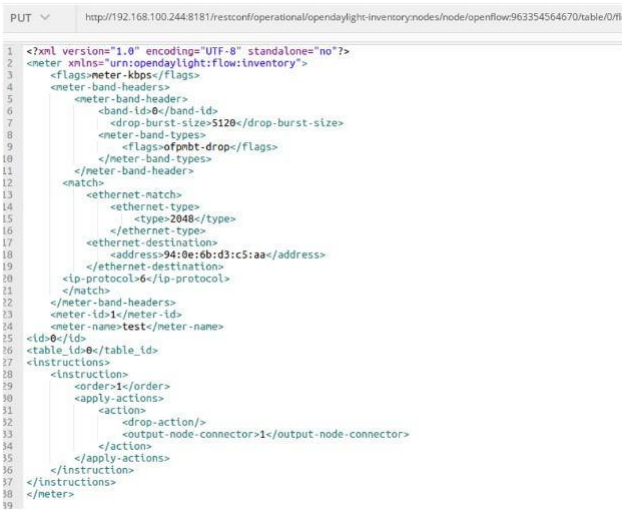
PUT : http://192.168.100.244:8181/restconf/operations/sal-meter:add-meter

รูปภาพที่ 4.3 การกำหนดค่า meter

ทำการสร้างกฎกำหนดการทำงานในการให้ใช้งานแบนด์วิธ

PUT : http://192.168.100.244:8181/restconf/operational/ opendaylight-inventory:nodes/node/openflow:963354564670:2/flow-node-inventory:meter/1

การกำหนดค่าทำงานจะเป็นรูปแบบ XML ดังรูป



รูปภาพที่ 4.4 ข้อมูล Flow Table การเพิ่มกฎการกำหนดแบนด์วิดธ์

คำสั่งรูปแบบ XML นี้ เป็นการเพิ่มกฎบน flow โดย Controller ค่าแบนด์วิดธ์ในการใช้งานได้ไม่เกิน 5 Mb ทำการกำหนดแบนด์วิดธ์ ใช้งานแบนด์วิดธ์ได้ไม่เกิน 5120 kbps หรือไม่เกิน 5 Mb ค่า meter =1 และทำการทอสอบเช่นเดิม แต่เปลี่ยนการกำหนดค่าการใช้งานแบนด์วิธในเครือข่ายกำหนดให้ใช้งานไม่เกิน 2048 Kbps หรือ ไม่เกิน 2 Mb

Host2 ที่ทำการทดสอบ Destination MAC Address 94:0e:6b:d3:c5:aa

Id = 0 คือลำดับ Flow

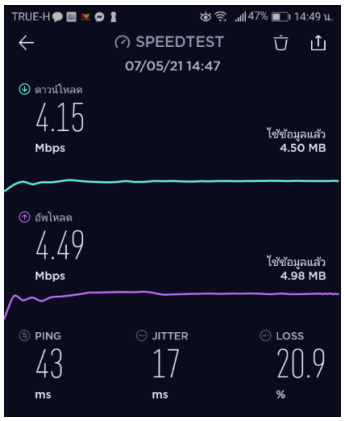
Table_id = 0 คือ flow table 0

Output-node-connector = 1 หมายถึง output port ที่ 1

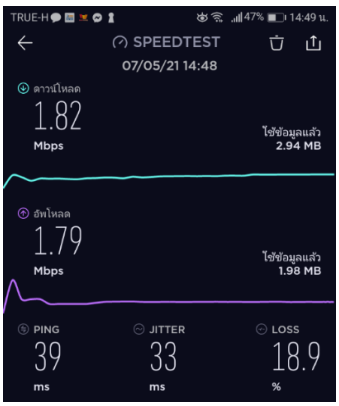
3. ผลการทดสอบการจัดสรรแบนด์วิธ

จากการทดสอบการทำงานของระบบเครือข่าย เมื่อมีการเพิ่มกฎในการทำงานพบว่า ในการกำหนดในการใช้งานแบนด์วิธของ เครื่อง host2 หมายเลข ip 192.168.100.223,
Mac address host: 94:0e:6b:d3:c5:aa ทำการทดสอบความเร็ว 2 แบบ การทดสอบแบบที่ 1 กำหนดความเร็วไม่เกิน 5120 Kbps หรือ 5 Mb และการทดสอบแบบที่ 2 กำหนดความเร็วไม่เกิน 2048 หรือ 2 Mb ในการใช้งานในเครือข่าย SDN ผลการทดสอบพบว่า กฎในการทำงานลนเครือข่ายที่กำหนดด้วยซอฟต์แวร์สามารถทำงานได้

รูปภาพที่ 4.5 ผลการทดสอบการกำหนดแบนด์วิธ



รูปภาพที่ 4.6 ผลการทดสอบการใช้งานแบนด์วิธไม่เกิน 5 Mb

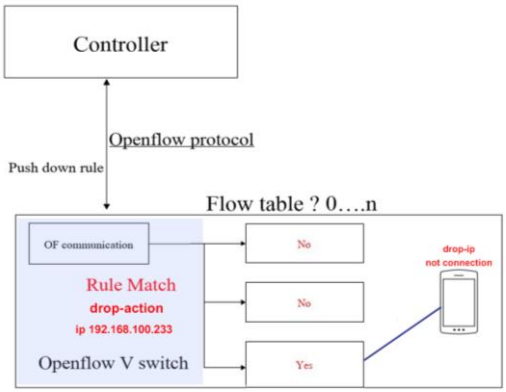


รูปภาพที่ 4.7 ผลการทดสอบการใช้งานแบนด์วิธไม่เกิน 2 Mb

จากการทดสอบ พบว่า การกำหนดกฎในการทำงานของแบนด์วิธ บนระบบเครือข่ายแบบ SDN สามารถกำหนดการใช้งานแบนด์วิธได้ตามกำหนดไว้ ที่กำหนดความเร็วไม่เกิน 5120 Kbps หรือ 5 Mb และการทดสอบแบบที่ 2 กำหนดความเร็วไม่เกิน 2048 หรือ 2 Mb

4 การทดสอบกฎการทำงานไม่อนุญาตให้หมายเลข ip address ใช้งาน

IP Address คือ หมายเลขประจำเครื่องคอมพิวเตอร์แต่ละเครื่องในระบบเครือข่าย โดยใช้โปรโตคอลแบบ TCP/IP สามารถบอกได้ว่าเครื่องคอมพิวเตอร์ตั้งอยู่ที่ใดและสามารถเชื่อมต่อใช้งานอินเตอร์เน็ตในระบบเครือข่ายได้ ในการทดสอบนี้จะศึกษากรณีการทำงานของเครือข่ายที่กำหนดด้วยซอฟต์แวร์ โดยการกำหนดกฎการทำงานบนเครือข่ายไม่อนุญาตให้หมายเลข ip ใช้งานอินเตอร์เน็ตได้ ซึ่งอุปกรณ์ที่นำมาเพื่อการทดสอบครั้งนี้ จะใช้เครื่องโทรศัพท์มือถือเคลื่อนที่รองรับการเชื่อมต่อแบบไร้สายและกำหนดหมายเลข คือ host2 ip 192.168.100.223 ในการทดสอบผู้วิจัยได้ทำการเพิ่มกฎการทำงานบน Controller เพื่อกำหนดกฎในการทำงานมีรูปแบบการทำงาน ดังนี้



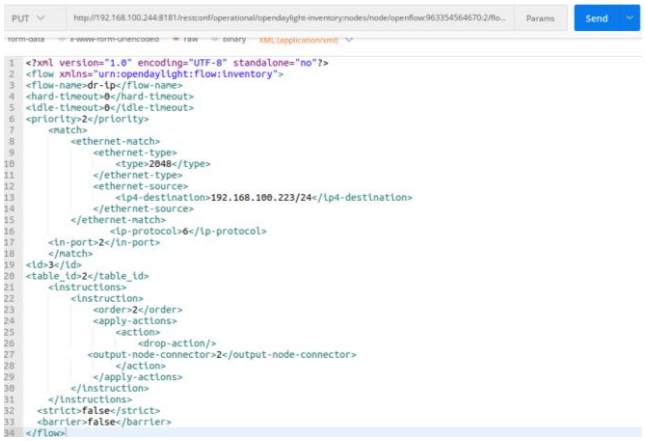
รูปภาพที่ 4.8 แสดงการทำงานไม่อนุญาตให้หมายเลข ip address ใช้งาน

ในการทดสอบการทำงานบนเครือข่ายกำหนดด้วยซอฟต์แวร์ ใช้เครื่องโทรศัพท์มือถือเคลื่อนที่รองรับการเชื่อมต่อแบบไร้สายและกำหนดหมายเลข คือ host2 ip 192.168.100.223 ในการทดสอบจะทำการ Drop-action ในการทำงานของหมายเลข ip 192.168.100.223 เพื่อไม่ให้ใช้งานอินเตอร์เน็ต โดยสร้างกฎการทำงานด้วยการ PUT ข้อมูลลงไปที่ flow table โดย Controller เพื่อกำหนดการใช้งานโดยใช้ Postman

การสร้างกฎคำสั่งให้ทำงาน ด้วยการ PUT ข้อมูลลงไปที่ flow table

PUT : http://192.168.1.62:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:963354564670:2/flow-node-inventory:table/2/flow/flow:3/tcp-flags-match

การกำหนดค่าทำงานจะเป็นรูปแบบ XML ดังรูป



รูปภาพที่ 4.9 สร้างกฎการทำงาน XML ไม่อนุญาตให้หมายเลข ip address ใช้งาน

คำสั่งรูปแบบเป็น XML เป็นการเพิ่มกฎลงบน flow table โดย Controller ในการทำงานตามที่กำหนด สิ่งที่ต้องระบุในการทำงานคือหมายเลข ip address ของเครื่องเป้าหมาย host2 คือ Destination 192.168.100.223 คำสั่งในการทำงานให้ drop-action ของหมายเลข ip 192.168.100.223 ที่ output-node-connector =2

Host2 การทดสอบ Destination IP Address 192.168.100.223

Id = 3 คือลำดับ Flow

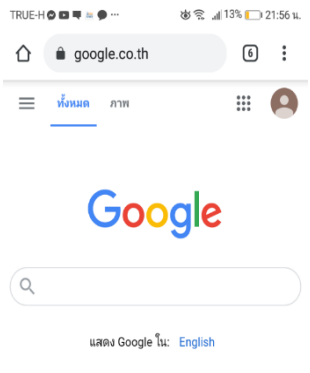
table id = 2 คือ flow table 2

เมื่อมีการเข้ามาใช้งานในระบบเครือข่ายไร้สายผ่านหมายเลข ip 192.168.10.223 ระบบทำการตรวจสอบว่าตรงตามกฎที่ได้สร้างไว้บน flow หรือไม่ ถ้า match กันตรงกฎที่ได้กำหนดไว้หมายเลข IP Address 192.168.100.223 ให้ทำการ drop-action การใช้งานของเครื่องนั้นไป

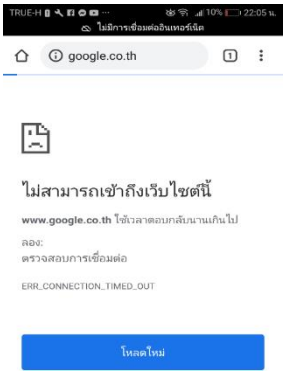
ในการทดสอบการทำงานจะทำการทดสอบ 2 กรณี

กรณีที่ 1 การทดสอบการใช้งานอินเตอร์เน็ต ก่อนการสร้างกฎการทำงาน

ทำการทดสอบด้วยการเข้าใช้งานอินเตอร์เน็ตก่อนการสร้างกฎคำสั่ง ด้วยการเชื่อมต่อกับเครือข่ายอินเตอร์เน็ตแบบไร้สายภายในเครือข่าย และทำการเข้าใช้งานอินเตอร์เน็ตผ่านบราวเซอร์ Google chrome เพื่อทำการทดสอบใช้งานอินเตอร์เน็ต ผลการทดสอบดังนี้



รูปที่ 4.10 ก่อนการสร้างกฎ



รูปที่ 4.11 การสร้างกฎการ block ip

จากการทดสอบกฎการใช้งานทั้ง 2 แบบ คือ

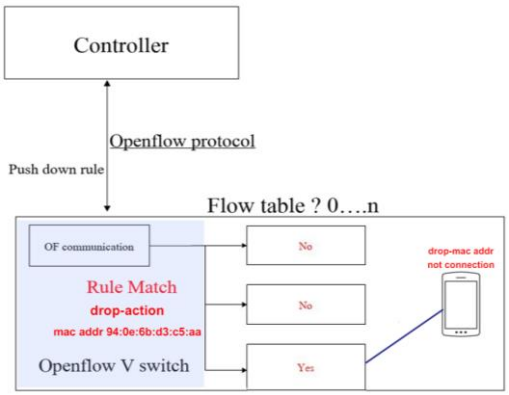
แบบที่ 1 การไม่กำหนดกฎการทำงาน การทดสอบการใช้งานก่อนการสร้างกฎคำสั่ง ผลปรากฏว่าสามารถเชื่อมต่อเครือข่ายไร้สายได้และสามารถใช้งานอินเตอร์เน็ตผ่านบราวเซอร์ google chrome สามารถเข้าใช้งานอินเตอร์เน็ตได้ปกติ

แบบที่ 2 การกำหนดกฎการทำงานด้วยการบล๊อคหมายเลข ip 192.168.100.223 ผลปรากฏว่าสามารถเชื่อมต่อเครือข่ายไร้สายได้และไม่สามารถเข้าใช้งานอินเตอร์เน็ตบนเครือข่ายไร้สายได้

4. การกำหนดกฎการทำงาน โดยหมายเลข Mac address ไม่อนุญาตให้ใช้งาน

Mac Address เป็น Physical Address ตัวเลขเฉพาะของอุปกรณ์ที่สามารถเชื่อมต่อ Network ได้เพื่อไม่ให้เกิดการซ้ำกัน ในการระบุหมายเลขของเครื่องในระบบเครือข่าย ในการทดสอบนี้จะศึกษากรณีการทำงานของเครือข่ายที่กำหนดด้วยซอฟต์แวร์ โดยการกำหนดกฎการทำงานบนเครือข่ายไม่อนุญาตให้หมายเลข mac address ใช้งานอินเตอร์เน็ตได้ ซึ่งอุปกรณ์ที่นำมาเพื่อการ

ทดสอบครั้งนี้ จะใช้เครื่องโทรศัพท์มือถือเคลื่อนที่รองรับการเชื่อมต่อแบบไร้สายเชื่อมต่อที่ host2 หมายเลข mac address คือ host:94:0e:6b:d3:c5:aa ในการทดสอบผู้วิจัยได้ทำการเพิ่มกฎการทำงานบน Controller เพื่อกำหนดกฎในการทำงานมีรูปแบบการทำงาน ดังนี้



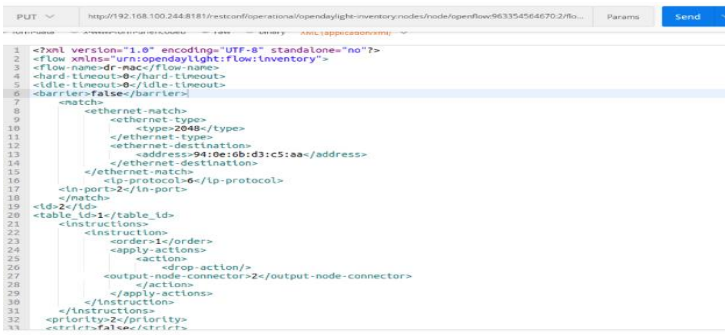
รูปภาพที่ 4.12 แสดงการทำงานไม่อนุญาตให้หมายเลข mac address ใช้งาน

ในการทดสอบการทำงานบนเครือข่ายกำหนดด้วยซอฟต์แวร์ ใช้เครื่องโทรศัพท์มือถือเคลื่อนที่รองรับการเชื่อมต่อแบบไร้สาย การเชื่อมต่อเครือข่ายที่ host2 หมายเลข mac address คือ host:94:0e:6b:d3:c5:aa ในการทดสอบจะทำการ Drop-action ในการทำงานเพื่อไม่ให้หมายเลข mac address 94:0e:6b:d3:c5:aa ใช้งาน โดยสร้างกฎการทำงานด้วยการ PUT ข้อมูลลงไปที่ flow table โดย Controller เพื่อกำหนดการใช้งานโดยใช้ Postman

การสร้างกฎคำสั่งให้ทำงาน ด้วยการ PUT ข้อมูลลงไปที่ flow table

PUT : http://192.168.1.62:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:963354564670:2/flow-node-inventory:table/1/flow/flow:2/match/ip-match

การกำหนดค่าทำงานจะเป็นรูปแบบ XML ดังรูป



รูปภาพที่ 4.13 สร้างกฎการทำงาน XML ไม่อนุญาตให้หมายเลข mac address ใช้งาน

คำสั่งรูปแบบเป็น XML เป็นการเพิ่มกฎลงบน flow table โดย Controller ในการทำงานตามที่กำหนด สิ่งที่ต้องระบุในการทำงานคือหมายเลข mac address ของเครื่องเป้าหมาย host2 คือ Destination ที่มี mac address 94:0e:6b:d3:c5:aa คำสั่งในการทำงานให้ drop-action ของ mac address 94:0e:6b:d3:c5:aa ที่ output-node-connector =2

Host2 ทำการทดสอบ Destination MAC Address 94:0e:6b:d3:c5:aa

Id = 2 คือลำดับ Flow

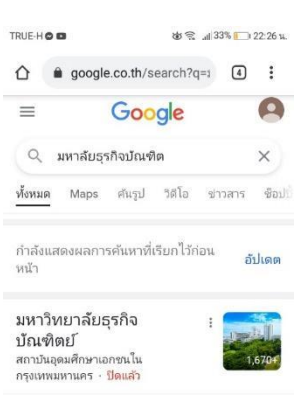
Table_id = 1 คือ flow table 1

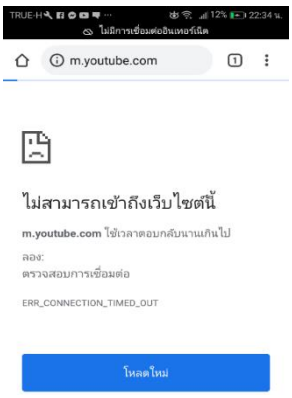
เมื่อมีการเข้ามาใช้งานในระบบเครือข่ายไร้สาย ที่มีหมายเลข mac address 94:0e:6b:d3:c5:aa ตามที่สร้างกฎไว้ ระบบทำการตรวจสอบว่าตรงตามกฎที่ได้สร้างไว้บน flow หรือไม่ ถ้า match กันตรงกฎที่ได้กำหนดการทำงานของ mac address 94:0e:6b:d3:c5:aa นี้ให้ทำการ drop-action การใช้งานของเครื่องนั้นไป

ในการทดสอบการทำงานจะทำการทดสอบ 2 กรณี

กรณีที่ 1 การทดสอบการใช้งานอินเตอร์เน็ต ก่อนการสร้างกฎการทำงาน

ทำการทดสอบด้วยการเข้าใช้งานอินเตอร์เน็ตก่อนการสร้างกฎคำสั่ง ด้วยการเชื่อมต่อกับเครือข่ายอินเตอร์เน็ตแบบไร้สายภายในเครือข่าย และทำการเข้าใช้งานอินเตอร์เน็ตผ่านบราวเซอร์ Google chrome เพื่อทำการทดสอบใช้งานอินเตอร์เน็ต ผลการทดสอบดังนี้





รูปที่ 4.14 ก่อนการสร้างกฎ รูปที่ 4.15 การสร้างกฎการ block mac address

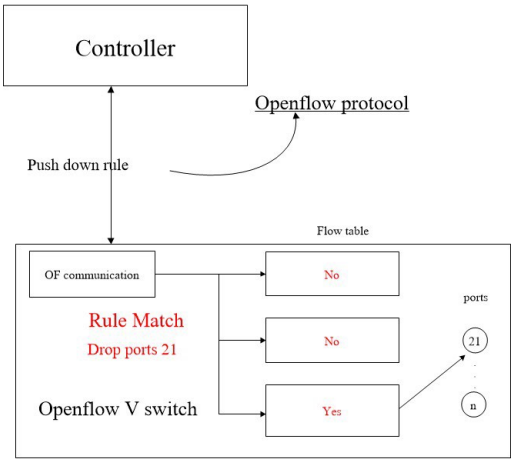
จากการทดสอบกฎการใช้งานทั้ง 2 แบบ คือ

แบบที่ 1 การไม่กำหนดกฎการทำงาน การทดสอบการใช้งานก่อนการสร้างกฎคำสั่ง ผลปรากฏว่าสามารถเชื่อมต่อเครือข่ายไร้สายได้และสามารถใช้งานอินเตอร์เน็ตผ่านบราวเซอร์ google chrome สามารถเข้าใช้งานอินเตอร์เน็ตได้ปกติ

แบบที่ 2 การกำหนดกฎการทำงานด้วยการบล๊อคหมายเลข mac address 94:0e:6b:d3:c5:aa ผลปรากฏว่า ไม่สามารถที่จะเชื่อมต่อเครือข่ายไร้สายได้ และไม่สามารถเข้าใช้งานอินเตอร์เน็ตบนเครือข่ายไร้สายได้

5 การทดสอบการกำหนดไม่ให้สิทธิ์การใช้งาน FTP

File transfer protocol (FTP) เป็นโปรโตคอลที่ใช้ในการโอนย้ายไฟล์หรือเป็นการถ่ายโอนข้อมูลข้ามเครือข่าย ระหว่างเครื่องลูก คือ Client และ เครื่องแม่ คือ เครื่อง Server โดยในการทำงานโดยทั่วไปแล้วจะใช้พอร์ตหมายเลข 21 ในการสื่อสาร ผู้วิจัยทำการทดสอบในการกำหนดกฎในการทำงานบนเครือข่าย SDN โดยไม่ให้สิทธิ์ในการใช้งาน FTP ผู้วิจัยได้ทำการเช่าพื้นที่โฮสต์สำหรับให้บริการโอนย้ายข้อมูล เพื่อทำการทดสอบการทำงาน ระหว่างเครื่อง Client ที่ติดตั้งอยู่บนเครือข่าย SDN ที่จะโอนย้ายข้อมูลไปยังเครื่อง host server



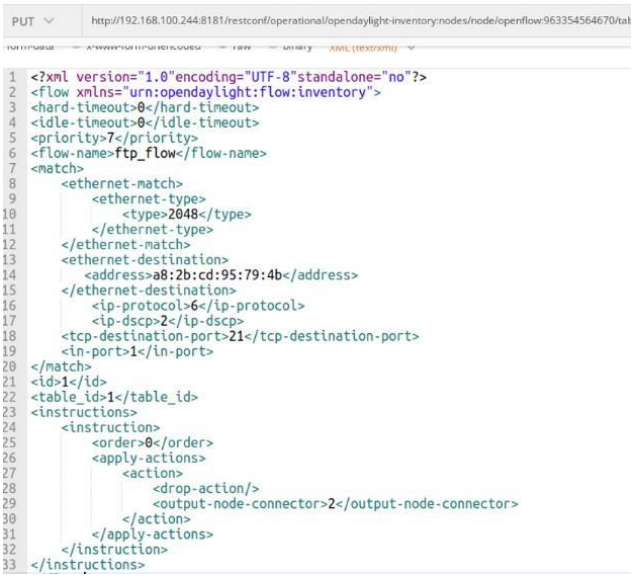
รูปภาพที่ 4.16 แสดงการทำงานไม่ให้ใช้งาน FTP

ในการทดสอบการทำงานบนเครือข่าย SDN ใช้เครื่องคอมพิวเตอร์โน๊ตบุ๊คจำนวน 1 เครื่อง เพื่อใช้ในการทดสอบการใช้งาน file transfer protocol (FTP) ในการทดสอบจะทำการ Drop Port เพื่อไม่ให้สิทธิ์ในการเข้าใช้งาน โดยสร้างกฎการทำงานด้วยการ PUT ข้อมูลลงไปที่ flow table โดย Controller เพื่อกำหนดการใช้งานโดยใช้ Postman

การสร้างกฎคำสั่งให้ทำงาน ด้วยการ PUT ข้อมูลลงไปที่ flow table

PUT : http://192.168.100.244:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:963354564670/table/1/stale-flow/ftp_flow/match

การกำหนดค่าทำงานจะเป็นรูปแบบ XML ดังรูป



รูปภาพที่ 4.17 สร้างกฎการทำงาน XML

คำสั่งรูปแบบเป็น XML เป็นการเพิ่มกฎลงบน flow table โดย Controller ในการทำงานตามที่กำหนด สิ่งท่ต้องระบุในการทำงานคือ MAC Address ของเครื่องเป้าหมาย โดยค่า Destination ที่ a8:2b:cd:95:79:4b ทำการ Drop port 21 เมื่อมีการ Action ของ node-connector =2

Id = 1 คือลำดับ Flow

Table_id = 1 คือ flow table 1

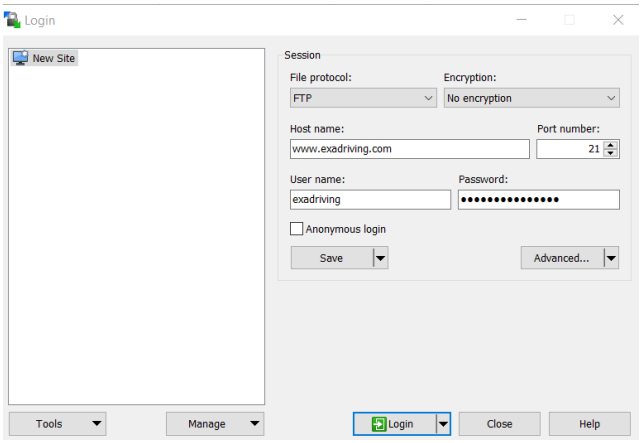
Output-node-connector = 2 หมายถึง output port ที่ 2 ของ switch

คือเมื่อมีการเข้ามาใช้งาน file transfer protocol โดยใช้ Port 21 ในการสื่อสาร ระบบทำการตรวจสอบว่าตรงตามกฎที่ได้สร้างไว้บน flow หรือไม่ ถ้า match กันตรงกับหมายเลข MAC Address a8:2b:cd:95:79:4b ที่กำหนดไว้ ให้ทำการ Drop การใช้งานของเครื่องนั้นไป

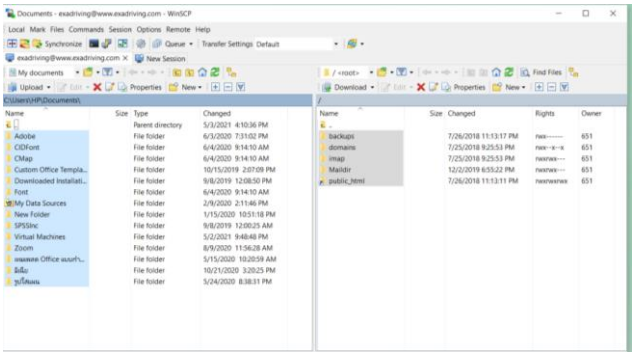
ในการทดสอบการทำงานจะทำการทดสอบ 2 กรณี

กรณีที่ 1 การทดสอบการใช้งาน FTP แบบไม่สร้างกฎ

ทำการทดสอบด้วยโปรแกรม WinSCP v.5.17 ในการใช้งานการโอนย้ายไฟล์ระหว่าง host หมายเลข MAC Address a8:2b:cd:95:79:4b ตามที่ได้กำหนดไว้นั้น สามารถเข้าใช้งานได้ปกติ เข้าถึงข้อมูลของไฟล์ทั้งฝั่ง host และฝั่ง server ดังรูป



รูปภาพที่ 4.18 การเข้าใช้งาน WinSCP (FTP)

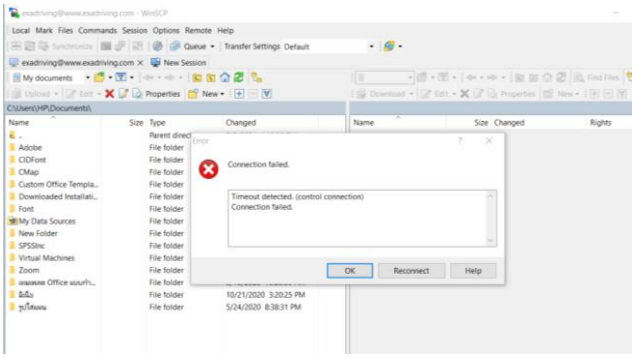


รูปภาพที่ 4.19 การทดสอบ FTP แบบไม่กำหนดกฎการทำงาน

รูปภาพที่ 4.20 การทดสอบ FTP แบบไม่กำหนดกฎการทำงานผ่าน Command Line

กรณีที่ 2 การทดสอบการใช้งาน FTP แบบสร้างกฎ

ทำการทดสอบด้วยโปรแกรม WinSCP v.5.17 เมื่อทำการเพิ่มกฎในการทำงานลง flow table โดย Controller ในรูปแบบของ XML ข้างต้นนั้น เมื่อทำการทดสอบการทำงานการเข้าใช้งานโดยผ่านโปรโตคอล FTP ระหว่าง host หมายเลข MAC Address a8:2b:cd:95:79:4b ที่เป็นเครื่อง host ไปยังเครื่อง server



รูปภาพที่ 4.18 การทดสอบ FTP แบบสร้างกฎการทำงาน

รูปภาพที่ 4.19 การทดสอบ FTP แบบสร้างกฎการทำงานผ่าน Command Line

ผลการทดสอบการใช้งาน พบว่า host หมายเลข MAC Address a8:2b:cd:95:79:4b ไม่สามารถเชื่อมต่อโดยผ่านโปรโตคอล FTP ไปยังเครื่อง server ได้ เกิดการ Connection time out

สรุปผลการทดลองและข้อเสนอแนะ

เนื้อหาในบทนี้จะเป็นการอภิปรายเพื่อสรุปผลที่ได้จากการทดลองงานวิจัยเรื่องการพัฒนาระบบเครือข่ายไร้สายที่กำหนดด้วยซอฟต์แวร์ รวมทั้งข้อจำกัดของระบบที่พบจากการทดลองระบบ

และข้อเสนอแนะสำหรับแนวทางในการพัฒนางานวิจัยนี้ต่อไป เพื่อแก้ไขข้อบกพร่องของระบบให้มีประสิทธิภาพมากยิ่งขึ้น

1 สรุปผลการวิจัย

จากงานวิจัยเรื่องการพัฒนาระบบเครือข่ายไร้สายที่กำหนดด้วยซอฟต์แวร์ ผลการทดลองในกรณีศึกษาการกำหนดการใช้งานแบนด์วิธและการกำหนดสิทธิ์ไม่ให้ใช้บริการ FTP Protocol ผ่านช่องทางการสื่อสาร Port 21 พบว่า การกำหนดกฎในการใช้งานแบนด์วิธบนระบบเครือข่าย SDN ได้ออกแบบไว้ มีการทำงานได้อย่างถูกต้องและปฏิบัติได้จริง และสามารถกำหนดการทำงานของข้อมูลได้ด้วย REST API ที่ใช้เป็นสร้างกฎในการทำงานของเครือข่าย ซึ่งจากการทดลองได้มีการจัดเตรียมระบบเพื่อจำลองเครือข่ายแบบ SDN ตั้งแต่การติดตั้ง Software เพื่อทำหน้าที่เป็น Controller ติดตั้งอุปกรณ์บอร์ด Raspberry Pi เพื่อทำหน้าที่เป็น Switch Open flow เพื่อทำหน้าที่ในการรับคำสั่งการทำงานจาก Controller และมีการกำหนดการตั้งค่าการทำงานของ Controller กับ Flow switch เพื่อให้การทำงานร่วมกันโดยไม่มีข้อผิดพลาดและให้เหมาะสมกับการทดสอบ รวมถึงการกำหนดรูปแบบการเชื่อมต่อในเครือข่ายและการกำหนดการทำงานให้กับ host สามารถปรับปรุงทิศทางการไหลของข้อมูลเพื่อใช้ในการทดสอบการให้การใช้งานในเครือข่ายได้อย่างมีประสิทธิภาพและใช้ได้จริง

ผู้วิจัยเห็นว่าจากผลการทดสอบ สำหรับผู้ที่ดูแลระบบสามารถนำผลการทดสอบจากการทำงานบนเครือข่ายที่กำหนดด้วยซอฟต์แวร์นี้ สามารถที่จะนำมาประยุกต์ใช้งานในการพัฒนาเครือข่ายขององค์กรได้อย่างเหมาะสม เพราะระบบเครือข่าย SDN ตอบโจทน์ในเรื่องของการลดค่าใช้จ่ายในการที่ไม่จำเป็นภายในองค์กรในเรื่องของเครื่องมือเครือข่าย บุคลากร และสามารถลดข้อผิดพลาดในการทำงานของระบบได้อย่างดี

2 ปัญหาและข้อเสนอแนะ

2.1 ปัญหาที่พบในการวิจัย

ในงานวิจัยนี้ Software ที่ใช้ในการทดสอบครั้งนี้ เป็นรูปแบบ Open Source นำมาใช้งานเพื่อทำหน้าที่เป็น Controller ถึงแม้จะเป็น Software ที่หลาย ๆ ท่านมีการศึกษาวิจัยมาแล้วนำมาใช้ในการทดสอบมาแล้ว แต่มีการนำไปใช้ต่างกรณี ทำให้เกิดข้อผิดพลาดของโปรแกรมเกิดขึ้นได้ ทำให้มีการแก้ไขในการทำงานในเรื่องของการหาเวอร์ชั่นของซอฟต์แวร์ที่สามารถทำงานให้สอดคล้องกับระบบการทำงานของ Software แต่ละเวอร์ชั่นนั้น รวมถึงการใช้งาน REST API ในการส่งข้อมูลต่าง ๆ ไปยัง Flow Switch มีฟอร์แมตของ JSON และ XML ที่ไม่รองรับในการทำงานของ

ระบบเครือข่ายแบบ SDN รวมถึงการทำงานของ หน้า GUI ต้องหาเวอร์ชั่นของจาวาที่สามารถรองรับการทำงาน GUI ด้วย

และฮาร์ดแวร์คืออุปกรณ์ บอร์ด Raspberry Pi ที่นำมาติดตั้งเป็น Open Virtual Switch (OVS) บอร์ด Raspberry Pi บางรุ่นไม่สนับสนุนในการทำงานในการติดตั้ง Open V Switch ผู้วิจัยต้องทำการศึกษาและทดลองเพื่อติดตั้งใช้งานให้สอดคล้องกับการทำงานร่วมกับ Controller ให้สามารถทำงานร่วมกันได้

ในระบบเครือข่ายแบบ SDN ในการกำหนดกฎการทำงานของข้อมูล ไปที่ Flow Table ไม่มีเครื่องมือในการมอนิเตอร์การทำงานของระบบได้เลยทำให้ไม่รู้ว่าจะเกิดการผิดพลาดขึ้นตอนไหน เพราะในการส่ง Packet ออกไปบางครั้งข้อมูลที่ส่งเกิดการผิดพลาดของคำสั่งอาจทำให้เกิดการ loop ในเครือข่ายเกิดขึ้นได้ แล้วทำให้เครือข่ายเกิดใช้ไม่ได้

2.2 ข้อเสนอแนะ

ในงานวิจัยนี้ ในการจัดการระบบเครือข่ายที่กำหนดด้วยซอฟต์แวร์ ในการติดตั้ง Controller โดยติดตั้งซอฟต์แวร์ที่เป็น Open Source ในการทดสอบการทำงาน Feature ที่ทำการติดตั้งอาจเกิดการ Error ในการทดสอบการทำงาน และในการตั้งกฎในการกำหนดต้นทางและปลายทางในระบบยังไม่สามารถกำหนดการทำงานได้ทั้ง Subnet ได้ รวมถึงในการทดสอบการทำงานนั้น ยังไม่มีมอนิเตอร์ในการตรวจสอบการทำงานของกฎที่ได้กำหนดในระบบได้ โดยในการทำงานจริงในระบบ ผู้ดูแลระบบเองต้องมากำหนดดูแลการทำงานบนระบบเครือข่ายเอง หากเป็นเพราะ Software ที่นำมาทดสอบเป็นแบบ Open Source ถ้าเป็น Software Define Network แบบสมบูรณ์ สามารถเรียกดูสถานะเครือข่ายและสามารถตัดสินใจในการรับประกันการทำงานให้กับการให้บริการต่างๆ บนเครือข่ายได้โดยอัตโนมัติและสามารถยกเลิกการรับประกันในการทำงานได้เมื่อไม่มีการใช้งานในเครือข่าย

ในการทดสอบการทำงาน ซอฟต์แวร์ที่เป็น Controller จะเกิดการ error บ่อยครั้งเมื่อมีการทำการ Run controller เพื่อให้ controller ทำงานนั้น อาจจะเป็นเพราะว่าซอฟต์แวร์ที่นำมาทำการทดสอบนั้นนั้นเป็น แบบ Open Source

บรรณานุกรม

วิศรุต คุ้มเงิน. (2561). *การจัดสรรแบนด์วิธแบบละเอียดในเครือข่ายกำหนดด้วยซอฟต์แวร์*.วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยีมหาวิทยาลัยธรรมศาสตร์

ปิยพงษ์ เคนเหลื่อม. (2561). *คุณภาพการให้บริการสตรีมมิ่งที่มีการจัดความสำคัญบนเครือข่าย เอสดีเอ็น*. สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม มหาวิทยาลัยธุรกิจบัณฑิต

Wang W. (2014). *Autonomic QoS Management Mechanism in Software Defined Networks* (AQSDN). China Communications.

Seddiki M S. (2014). *Flow Quality of Service (FlowQoS) Flow Quality of Service* (FlowQoS)

Software Defined Network (SDN). แหล่งที่มา: https://www.cyberthai.com/th/single-item/without-sidebar

OpenDaylight Foundation. แหล่งที่มา: https://www.opendaylight.org/.

OpenDaylight Flow Examples. แหล่งที่มา: https://docs.opendaylight.org/projects/openflowplugin/en/latest/users/flow-examples.html

Open V witch on Raspberry Pi. แหล่งที่มา: https://techflow360.com/install-ovs-openvswitch-in-raspberry-pi/

Open Networking Foundation (ONF). แหล่งที่มา: https://www.opennetworking.org/sdn-resources/openflow