

การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุม
การกระจายแพทर्फฟิกแบบที่ผู้ใ้กำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับ
มาตรฐานเอสดีเอ็น

**IMPLEMENTATION OF SDN APPLICATION AND CONTROLLER
FOR USER-DEFINED TRAFFIC DISTRIBUTION IN TRADITIONAL
(NON-SDN) NETWORKS**

โดย

พงศ์พนิช อรัญรัตน์โสภณ

PONGPANIT ARANRATSOPON

ภูริณัฐ จิตมนัส

PURINUT JITMANAS

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2564

การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุม
การกระจายแพทเทรฟฟิกแบบที่ผู้ใช้งานกำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับ
มาตรฐานเอสดีเอ็น

**IMPLEMENTATION OF SDN APPLICATION AND CONTROLLER
FOR USER-DEFINED TRAFFIC DISTRIBUTION IN TRADITIONAL
(NON-SDN) NETWORKS**

โดย
พงศ์พนิช อรัญรัตน์โสภณ
ภูริณัฐ จิตมนัส

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2564

**IMPLEMENTATION OF SDN APPLICATION AND CONTROLLER
FOR USER-DEFINED TRAFFIC DISTRIBUTION IN TRADITIONAL
(NON-SDN) NETWORKS**

**PONGPANIT ARANRATSOPON
PURINUT JITMANAS**

**A PROJECT SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2/2021

COPYRIGHT 2022

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG


ใบรับรองปริญญาโท ประจำปีการศึกษา 2564
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อ
ควบคุมการกระจายทราฟฟิกแบบที่ผู้ใช้กำหนดได้ บนเครือข่ายที่อุปกรณ์
ไม่รองรับมาตรฐานเอสดีเอ็น

IMPLEMENTATION OF SDN APPLICATION AND
CONTROLLER FOR USER-DEFINED TRAFFIC DISTRIBUTION
IN TRADITIONAL (NON-SDN) NETWORKS

ผู้จัดทำ

1. นายพงศ์พนิช อรัญรัตน์โสภณ รหัสนักศึกษา 61070124
2. นายภูริณัฐ จิตมณัส รหัสนักศึกษา 61070171


.....อาจารย์ที่ปรึกษา
(ผศ.ดร. สุเมธ ประภาวัต)

ใบรับรองโครงการ(Project)

เรื่อง

การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุม
การกระจายแพทเทรฟฟิกแบบที่ผู้ใช้กำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับ
มาตรฐานเอสดีเอ็น

**IMPLEMENTATION OF SDN APPLICATION AND CONTROLLER
FOR USER-DEFINED TRAFFIC DISTRIBUTION IN TRADITIONAL
(NON-SDN) NETWORKS**

นายพงศ์พนิช อรัญรัตน์โสภณ รหัสประจำตัว 61070124

นายภูริณัฐ จิตมนัส รหัสประจำตัว 61070171

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด
รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของการศึกษาวิชาโครงการ หลักสูตร
วิทยาศาสตร์บัณฑิต (เทคโนโลยีสารสนเทศ)
ภาคเรียนที่ 2 ปีการศึกษา 2564

พงศ์พนิช อรัญรัตน์โสภณ

(นายพงศ์พนิช อรัญรัตน์โสภณ)

ภูริณัฐ จิตมนัส

(นายภูริณัฐ จิตมนัส)

หัวข้อโครงการ	การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุมการกระจายแพทเทรฟฟิกแบบที่ผู้ใช้กำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับมาตรฐานเอสดีเอ็น	
นักศึกษา	นายพงศ์พนิช อรัญรัตน์โสภณ	รหัสนักศึกษา 61070124
	นายภูริณัฐ จิตมณัส	รหัสนักศึกษา 61070171
ปริญญา	วิทยาศาสตร์บัณฑิต	
สาขาวิชา	เทคโนโลยีสารสนเทศ	
ปีการศึกษา	2564	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต	

บทคัดย่อ

สถาปัตยกรรมที่ใช้ในการบริหารจัดการอุปกรณ์เครือข่ายแบบรวมศูนย์ หรือ สถาปัตยกรรมแบบ Software Defined Network (SDN) ในปัจจุบันไม่สามารถใช้งานร่วมกับอุปกรณ์เครือข่ายแบบดั้งเดิม (Legacy Network Device) ซึ่งไม่รองรับการทำงานตามสถาปัตยกรรมแบบเอสดีเอ็นได้ ในการพัฒนาครั้งนี้คณะผู้จัดทำได้ทำการพัฒนาระบบควบคุมอุปกรณ์เครือข่ายตามสถาปัตยกรรมแบบเอสดีเอ็นที่สามารถจัดการควบคุมอุปกรณ์เครือข่ายแบบดั้งเดิมได้ รวมถึงพัฒนาแอปพลิเคชันที่ใช้งานระบบควบคุมเครือข่ายนี้ในการบริหารจัดการแพทเทรฟฟิกในระบบเครือข่าย

Project Title	Implementation of SDN Application and Controller for User-defined Traffic Distribution in Traditional (Non-SDN) Networks	
Student	Mr. PONGPANIT ARANRATSOPON	Student ID 61070124
	Mr. PURINUT JITMANAS	Student ID 61070171
Degree	Bachelor of Science	
Program	Information Technology	
Academic Year	2021	
Advisor	Asst. Prof. Dr. Sumet Prarabhavat	

ABSTRACT

SDN technology introduces flexible and efficient management system. System and network administrator can easily manage, control, and monitor network systems. However, it supports only SDN-compatible network devices (i.e., not legacy devices). This project develops a management system for legacy network devices, based-on SDN architecture, including a controller and two example applications. Network management and monitoring application provides web-based GUI for provisioning, configuration, and administration of the network devices. Traffic distribution application is an example how to use controller's APIs for collecting status of network and traffic, performing traffic distribution algorithm, and setting up a new route on the network devices. Finally, experiment scenarios are conducted to evaluate our system functionalities. The results show that the applications can work with the controller. When congestion occurs, a large flow in the congested link will be rerouted to a new path having the largest available bandwidth. Therefore, excessive traffic can be distributed from over-utilized link to under-utilized link.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยการให้ความช่วยเหลือแนะนำของ ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้คำแนะนำ ข้อคิดเห็น ตรวจสอบและแก้ไขร่างปริญญานิพนธ์มาโดยตลอด ผู้เขียนจึงขอขอบพระคุณไว้ ณ โอกาสนี้

พงศ์พนิช อรัญรัตน์โสภณ

ภูริณัฐ จิตมนัส

สารบัญ

	หน้า
บทคัดย่อ.....	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง.....	3
2.1 Software Defined Network (SDN)	3
2.1.1 Application Layer	3
2.1.2 Control Layer	4
2.1.3 Infrastructure Layer	4
2.2 Traffic Engineering	4
2.2.1 Traffic Flow	5
2.2.2 Traffic Distribution	5
2.2.3 Routing.....	6

สารบัญ (ต่อ)

	หน้า
2.3 การสำรวจเก็บข้อมูลเครือข่าย	7
2.3.1 Simple Network Management Protocol (SNMP)	7
2.3.2 NetFlow	7
2.3.3 Cisco Discovery Protocol (CDP)	7
2.3.4 Secure Shell (SSH)	7
2.4 เทคโนโลยีที่ใช้ในการพัฒนา	8
2.4.1 Python	8
2.4.2 MongoDB	8
2.4.4 Netmiko	8
2.5 ระบบตัวควบคุมต้นแบบ	9
บทที่ 3 แนวคิดและการดำเนินงาน	10
3.1 ขั้นตอนการดำเนินงาน	10
3.2 ภาพรวมระบบ	11
3.3 ส่วนแสดงผลสำหรับผู้ใช้งาน (Web Application)	12
3.3.1 Home	13
3.3.2 Devices	13
3.3.3 Initialization	15
3.4 ส่วนตัวควบคุม (Controller)	16
3.4.1 Use Case Diagram	16
3.4.2 ขั้นตอนการเพิ่มอุปกรณ์	17
3.4.3 ขั้นตอนการ Initialize อุปกรณ์ในระบบ	18

สารบัญ (ต่อ)

	หน้า
3.4.4 ฐานข้อมูล	19
3.5 แอปพลิเคชันสำหรับการกระจายแพทไฟก (Traffic Distribution Application)	20
3.5.1 Traffic Distribution	20
3.5.2 Policy	21
3.5.3 Aging Policy	21
บทที่ 4 ผลการดำเนินการ	24
4.1 เว็บแอปพลิเคชันสำหรับแสดงผลและควบคุมอุปกรณ์	24
4.1.1 Network Topology	24
4.1.2 Adding Device and Initialization	26
4.1.3 Flow Filter and Policy Routing	28
4.2 แอปพลิเคชันสำหรับกระจายแพทไฟก	30
4.2.1 First Topology	30
4.2.2 Second Topology	36
บทที่ 5 บทสรุป	40
5.1 สรุปผลโครงการ	40
5.2 ปัญหาในการทำโครงการและสรุปผล	40
5.3 แนวทางการพัฒนาต่อ	41
บรรณานุกรม	42
ภาคผนวก	44
ประวัติผู้เขียน	51

สารบัญตาราง

ตารางที่	หน้า
4.1 ข้อมูลชื่ออุปกรณ์ ชื่อ-ขนาดพอร์ต และไอพีของอุปกรณ์เครือข่าย	25

สารบัญรูป

รูปที่	หน้า
2.1 ขอบเขตของวิศวกรรมจราจรที่ใช้ในโครงการนี้	5
2.2 องค์ประกอบภาพรวมระบบต้นแบบ	9
3.1 องค์ประกอบภาพรวมของระบบ	11
3.2 Use Case Diagram ของ Web Application	12
3.3 หน้า Home สำหรับแสดงโทโพโลยี โฟลว์และจัดการนโยบายปรับเปลี่ยนเส้นทาง	13
3.4 หน้า Device สำหรับ เพิ่ม ลบ แก้ไข และดูสถานะเชื่อมต่ออุปกรณ์	14
3.5 ตัวควบคุมติดต่ออุปกรณ์ได้ และอุปกรณ์ถูกเพิ่มเข้าสู่ระบบ	14
3.6 หน้า Initialization สำหรับเปิดการเชื่อมต่อให้อุปกรณ์ส่งข้อมูลมายังตัวควบคุม	15
3.7 Use Case Diagram ของตัวควบคุม	16
3.8 ขั้นตอนกระบวนการเพิ่มอุปกรณ์	17
3.9 ขั้นตอน Initialize เพื่อให้อุปกรณ์ส่งข้อมูลมายังตัวควบคุม	18
3.10 ขั้นตอนการกระจายแพคเกจ	22
3.11 ขั้นตอนการทำ Aging Policy เพื่อลบนโยบายที่โฟลว์ไม่ปรากฏเป็นระยะเวลาหนึ่ง	23
4.1 โทโพโลยีที่ใช้ทดสอบ ตัวควบคุมอยู่ใน Cloud Network.....	24
4.2 ข้อมูลอุปกรณ์ในระบบและสถานะการทำงาน	26
4.3 อุปกรณ์ตั้งค่าให้ส่งข้อมูล SNMP มายังตัวควบคุมไอพี 10.50.34.15	26
4.4 อุปกรณ์ การเชื่อมต่อ การไหลของโฟลว์ และซันเน็ตแสดงออกมาได้ถูกต้อง	27
4.5 ข้อมูลลิงก์ถูกแสดงบนหน้าเว็บ	27
4.6 ใช้ Iperf3 ในการสร้างแพคเกจ	28
4.7 มีข้อมูลถูกส่งในลิงก์ประมาณ 400 Kbit/sec ตรวจสอบโดย Wireshark	29
4.8 แสดงข้อมูลโฟลว์ที่มีไอพีต้นทาง 192.168.110.10	29
4.9 โฟลว์ถูกเปลี่ยนเส้นทาง.....	30
4.10 โทโพโลยีการทดลองที่ 1	30
4.11 นโยบายสำหรับเปลี่ยนเส้นทาง	31

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.12 มีเส้นทางใหม่ที่ถูกใช้งาน.....	32
4.13 เส้นทางก่อนมีการกระจายแรงแฟฟีก.....	32
4.14 เส้นทางหลังมีการกระจายแรงแฟฟีก	33
4.15 ข้อมูลการใช้งานลิงก์ R8-R6.....	33
4.16 ข้อมูลการใช้งานลิงก์ R8-R5.....	34
4.17 ข้อมูลการใช้งานลิงก์ R8-R7.....	34
4.18 ข้อมูลการใช้งานลิงก์ R8-R5 เปรียบเทียบระหว่างมีและไม่มีมีการกระจายแรงแฟฟีก	35
4.19 ข้อมูลการใช้งานลิงก์ R8-R7 เปรียบเทียบระหว่างมีและไม่มีมีการกระจายแรงแฟฟีก	35
4.20 โทโพโลยีการทดลองที่ 2.....	36
4.21 เส้นทางก่อนมีการกระจายแรงแฟฟีก.....	37
4.22 นโยบายสำหรับเปลี่ยนเส้นทาง	37
4.23 เส้นทางหลังมีการกระจายแรงแฟฟีก	38
4.24 ข้อมูลการใช้งานลิงก์ที่ถูกใช้งานก่อนมีการกระจายแรงแฟฟีก.....	38
4.25 ข้อมูลการใช้งานลิงก์ที่ถูกใช้งานหลังมีการกระจายแรงแฟฟีก	39

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันระบบเครือข่ายมีการเติบโต และมีการใช้งานเพิ่มขึ้นเป็นจำนวนมากทำให้ข้อมูลในระบบเครือข่ายมีจำนวนเพิ่มมากขึ้นเป็นทวีคูณ ส่งผลให้เกิดปัญหาความคับคั่งของการจราจรเครือข่าย (เน็ตเวิร์กแทรฟฟิก) ในบางเส้นทางได้ การจัดการแทรฟฟิกจึงมีความสำคัญที่ทำให้ระบบเครือข่ายสามารถใช้งานได้อย่างมีประสิทธิภาพ สถาปัตยกรรมแบบเอสดีเอ็นถูกคิดค้นขึ้นมาเพื่อช่วยบริหารจัดการอุปกรณ์เครือข่ายที่มีความซับซ้อน อย่างไรก็ตามสถาปัตยกรรมดังกล่าวสามารถใช้ได้กับอุปกรณ์เครือข่ายที่รองรับมาตรฐานเอสดีเอ็นเท่านั้น

แอปพลิเคชันและคอนโทรลเลอร์ตามสถาปัตยกรรมแบบเอสดีเอ็นระบบนี้ เป็นระบบที่พัฒนาขึ้นเพื่อที่จะช่วยให้อุปกรณ์เครือข่ายแบบดั้งเดิมที่ไม่รองรับแอปพลิเคชันเอสดีเอ็นสามารถบริหารจัดการเก็บข้อมูลต่าง ๆ ของระบบเครือข่ายแสดงผลให้ผู้ใช้งานเข้าใจง่าย และสามารถตั้งค่าจัดการแทรฟฟิกแบบที่ผู้ใช้ต้องการได้

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาและพัฒนาเอสดีเอ็นคอนโทรลเลอร์เพื่อควบคุมอุปกรณ์เครือข่ายที่ไม่รองรับการทำงานตามสถาปัตยกรรมเอสดีเอ็นได้
2. เพื่อศึกษาแอปพลิเคชันสำหรับจัดการอุปกรณ์เครือข่ายที่ไม่รองรับการทำงานตามสถาปัตยกรรมเอสดีเอ็นผ่านเอสดีเอ็นคอนโทรลเลอร์ที่ถูกพัฒนาขึ้น
3. เพื่อศึกษาแนวทางและพัฒนากลไกการกระจายแทรฟฟิกบนเครือข่ายตามที่ใช้กำหนดผ่านทางส่วนติดต่อกับผู้ใช้งาน
4. เพื่อศึกษาวิธีการทดสอบและประเมินเพื่อนำไปสู่ข้อสรุปเกี่ยวกับประสิทธิภาพการกระจายแทรฟฟิกของระบบจัดการที่พัฒนาขึ้น

1.3 ขอบเขตของโครงการ

พัฒนาระบบจัดการเครือข่ายสำหรับอุปกรณ์ที่ไม่รองรับมาตรฐานเอสดีเอ็นเท่าที่สามารถหาได้จากการที่คณะจัดสรรให้ เพื่อให้สามารถควบคุมอุปกรณ์เหล่านั้นให้ทำงานตามนโยบายที่กำหนดผ่านทางส่วนติดต่อผู้ใช้งานที่เป็นเว็บแอปพลิเคชันได้

1.4 วิธีการดำเนินงาน

1. ประเมินทรัพยากรและเครื่องมือที่มีในการสร้างสถาปัตยกรรมระบบการจัดการเครือข่าย
2. ดำเนินการปรับปรุงระบบเพื่อให้พร้อมต่อการพัฒนาต่อ ยอด
3. ศึกษาเทคโนโลยีและแนวทางการพัฒนาระบบกระจายแพทไฟกในระบบเครือข่าย
4. ออกแบบสถาปัตยกรรมของระบบจัดการเครือข่ายใหม่
5. ดำเนินการพัฒนาระบบกระจายแพทไฟกในระบบเครือข่ายและผสานเข้ากับระบบจัดการเครือข่าย
6. ทดสอบการใช้งานและสรุปผล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ระบบจัดการเครือข่ายตามสถาปัตยกรรมแบบเอสดีเอ็นที่สามารถมอนิเตอร์และกระจายแพทไฟกบนเครือข่ายได้ตามที่กำหนด แม้ว่าอุปกรณ์ในเครือข่านั้นจะไม่รองรับมาตรฐานเอสดีเอ็นก็ตาม

บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

ผู้จัดทำได้มีการศึกษาทฤษฎีที่เกี่ยวข้องกับงานวิจัย และเทคโนโลยีที่ใช้ในการจัดทำ โดยมีรายละเอียดดังนี้

2.1 Software Defined Network (SDN)

Software Defined Network (เอสดีเอ็น) เป็นเทคโนโลยีที่เกี่ยวข้องกับการบริหารจัดการเครือข่ายโดยปรับปรุงการบริหารจัดการเครือข่ายแบบเดิมให้มีการรวมศูนย์ ลดความซับซ้อนของระบบ ทำให้ง่ายแก่การจัดการ ผู้ดูแลระบบเครือข่ายสามารถจัดการระบบเครือข่ายผ่านคอนโทรลเลอร์โดยไม่ต้องเข้าไปจัดการกับอุปกรณ์เครือข่ายอื่น ๆ โดยตรง ระบบเครือข่ายแบบดั้งเดิมการควบคุมแพทไฟกจะขึ้นอยู่กับตารางเส้นทางที่ถูกตั้งค่าไว้ตามอุปกรณ์เครือข่ายในแต่ละเครื่อง ในสถาปัตยกรรมแบบเอสดีเอ็นจะแยกการควบคุม และการส่งข้อมูลออกจากกันทำให้สามารถรู้สถานะของระบบเครือข่ายจากคอนโทรลเลอร์และสามารถควบคุมระบบเครือข่ายทั้งหมดผ่านทางคอนโทรลเลอร์เพียงอย่างเดียว โดยสถาปัตยกรรมแบบเอสดีเอ็นได้แบ่งลำดับชั้นการทำงานเป็น 3 ลำดับชั้นโดยสื่อสารผ่าน Application Programming Interfaces หรือ API ดังนี้ [1]

2.1.1 Application Layer

ชั้นแอปพลิเคชัน เป็นส่วนติดต่อผู้ใช้งานทำหน้าที่รับส่งข้อมูลคำสั่งตามที่ใช้ต้องการ โดยแอปพลิเคชันมีฟังก์ชันการทำงานที่หลากหลายขึ้นอยู่กับการประยุกต์ใช้งาน เช่น การทำโหลดบาลานซ์ ระบบตรวจจับสิ่งแปลกปลอม เป็นต้น โดยจะใช้โปรแกรมติดต่อกับชั้นควบคุมเพื่อที่จะจัดการให้บรรลุวัตถุประสงค์การทำงานตามที่ใช้ต้องการ

2.1.2 Control Layer

ชั้นควบคุม เป็นส่วนควบคุมทำหน้าที่เป็นตัวกลางระหว่าง แอปพลิเคชันและอุปกรณ์เครือข่ายทำหน้าที่ส่งคำสั่งควบคุมการทำงาน จัดเก็บค่าสถานะเครือข่ายเปรียบได้กับสมองของเอสดีเอ็น โดยจะติดต่อกับผู้ใช้งานผ่านชั้นแอปพลิเคชันโดยใช้ Northbound API และติดต่ออุปกรณ์เครือข่ายโดยใช้ Southbound API [2]

Northbound API อินเตอร์เฟซเหนือ ช่วยให้ชั้นควบคุมสามารถติดต่อกับลำดับชั้นบนหรือส่วนของแอปพลิเคชันได้

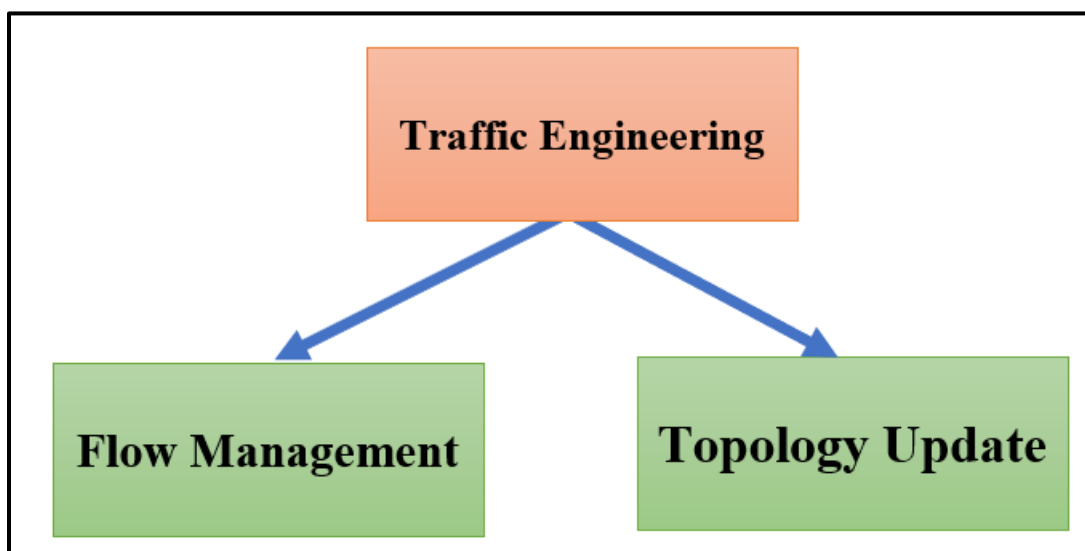
Southbound API อินเตอร์เฟซใต้ ช่วยให้ชั้นควบคุมสามารถติดต่อกับอุปกรณ์เครือข่ายอื่นในระดับล่าง ในเอสดีเอ็น คือ โปรโตคอล Netmiko

2.1.3 Infrastructure Layer

ชั้นโครงสร้างประกอบไปด้วยอุปกรณ์เครือข่ายต่าง ๆ ซึ่งเป็นพื้นฐานของระบบใช้ในการรับ ส่งข้อมูล โดยอุปกรณ์เหล่านี้ในสถาปัตยกรรมเอสดีเอ็นจะมีหน้าที่ส่งข้อมูลไปให้ชั้นควบคุมและทำหน้าที่ตามที่ส่วนควบคุมกำหนด

2.2 Traffic Engineering

วิศวกรรมจราจร คือ วิศวกรรมแขนงหนึ่งซึ่งเกี่ยวข้องกับการวางแผน การออกแบบ การควบคุม บริหารจัดการระบบการจราจรของถนน เพื่อให้ได้มาซึ่งความสะดวก รวดเร็ว และประหยัดเวลาในการขนส่งผู้โดยสาร โดยในที่นี้จะเป็นการออกแบบ ควบคุมการรับส่งข้อมูลหรือแพทर्फิกในระบบเครือข่าย เพื่อให้การรับส่งข้อมูลในระบบสามารถทำได้อย่างมีประสิทธิภาพตามแผนที่ได้วางเอาไว้ ในโครงการนี้จะใช้ขอบเขตของวิศวกรรมจราจรในส่วนการจัดการโพล์เพื่อทำการกระจายแพทर्फิก และการอัปเดตโทโพโลยีโดยการกำหนดนโยบายที่ทำให้เกิดการปรับเปลี่ยนเส้นทางในโพล์ของเครือข่าย [3]



รูปที่ 2.1 ขอบเขตของวิศวกรรมจราจรที่ใช้ในโครงงานนี้

2.2.1 Traffic Flow

แทรฟฟิกโฟลว์ คือ ชุดของแพ็กเก็ต (Sequence of packets) ที่ถูกส่งจากต้นทางหนึ่งไปยังปลายทางหนึ่งในระบบเครือข่าย โฟลว์ถูกแบ่งเป็น 4 ประเภทตามขนาดและระยะเวลาที่โฟลว์อยู่ในระบบ ดังนี้ [4]

1. Short-lived Large Flows เป็นโฟลว์ที่มีขนาดใหญ่และปรากฏอยู่ในระบบเป็นเวลานาน
2. Long-lived Large Flows เป็นโฟลว์ที่มีขนาดใหญ่และปรากฏในระบบเป็นเวลานาน
3. Short-lived Small Flows เป็นโฟลว์ที่มีขนาดเล็กและปรากฏในระบบเป็นเวลานาน
4. Long-lived Small Flows เป็นโฟลว์ที่มีขนาดเล็กและปรากฏในระบบเป็นเวลานาน

ตัวอย่างเช่น โฟลว์จากแอปพลิเคชันแชทในส่วนของการส่งข้อความมักจะเป็นประเภท Short-lived Small Flows และโฟลว์จากการส่งไฟล์ส่วนใหญ่มักจะเป็นโฟลว์ประเภท Long-lived Large Flows เป็นต้น

2.2.2 Traffic Distribution

การกระจายการจราจร ถือเป็นการจัดการโฟลว์รูปแบบหนึ่ง เป้าหมายเพื่อปรับเปลี่ยนเส้นทางของแทรฟฟิกในระบบให้ใช้เส้นทางที่แตกต่างจากเดิม เพื่อลดความคับคั่งของ

เส้นทางที่มีการจราจรหนาแน่น ทำให้การรับส่งข้อมูลสามารถทำได้มีประสิทธิภาพมากยิ่งขึ้น

ในโครงการนี้จะนำแนวคิดการจัดการโฟลว์ของ Hedera มาใช้ [3] โดยแนวคิดการจัดการโฟลว์นี้มีขั้นตอน 2 ขั้นตอน คือ

1. เมื่อพบโฟลว์ขนาดใหญ่ (Large Flows) จะเลือกส่งตามเส้นทาง ตามค่า Hash ของโฟลว์เหล่านั้น ทำให้ไปเรื่อย ๆ จนเกิดเส้นทางที่ถูกใช้งานสูงกว่า Threshold ที่กำหนด
2. นำโฟลว์ขนาดใหญ่นั้น คำนวณหาเส้นทางที่เหมาะสมอื่น เมื่อย้ายโฟลว์ดังกล่าวไปแล้วต้องไม่เกินค่า Threshold ของเส้นทางใหม่เช่นกัน

2.2.3 Routing

2.2.3.1 Destination-Based Routing

เป็นวิธีทั่วไปสำหรับการเลือกเส้นทางสำหรับส่งข้อมูลของอุปกรณ์เครือข่าย ซึ่งจะอิงข้อมูลจุดหมายปลายทางเป็นเกณฑ์ อุปกรณ์จะใช้ข้อมูลจากตารางเส้นทาง (Routing Table) ตารางเส้นทางจะประกอบไปด้วยข้อมูลเกี่ยวกับเครือข่ายปลายทาง และเส้นทางที่จะเลือกไปตามปลายทางเหล่านั้น โดยทั่วไปจะเลือกรายการที่มีค่าตรงกันมากที่สุด โดยทั่วไปก็คือรายการที่มี Prefix ตรงกันยาวที่สุด

2.2.3.2 Flow-Based Routing และ Policy-Based Routing

เป็นวิธีการเลือกเส้นทางที่แตกต่างจาก Destination-Based Routing โดยจะใช้ข้อมูลของโฟลว์เป็นเกณฑ์แทนจุดหมายปลายทาง กล่าวคือ ถึงโฟลว์จะมีจุดหมายปลายทางเดียวกัน แต่อุปกรณ์เครือข่ายไม่จำเป็นต้องเลือกเส้นทางให้เหมือนกัน

Policy-Based Routing (PBR) เป็นวิธีการเลือกเส้นทางโดยใช้นโยบาย ก็คือข้อมูลที่อยู่ต้นทาง ประเภทโปรโตคอล หรือชนิดของแอปพลิเคชัน โดย PBR จะถูกใช้เป็นเกณฑ์การเลือกเส้นทางหลัก แทนที่การเลือกเส้นทางแบบเก่าของอุปกรณ์เครือข่าย

เนื่องจาก Policy-Based Routing เป็นวิธีที่ทำให้สามารถจัดการโฟลว์ได้ยืดหยุ่น และมีประสิทธิภาพ ด้วยคุณสมบัติดังกล่าวจึงสามารถนำมาประยุกต์ใช้ในการกระจายแพคเกจ (Traffic Distribution) ได้ [5][6]

2.3 การสำรวจเก็บข้อมูลเครือข่าย

2.3.1 Simple Network Management Protocol (SNMP)

เอสเอ็นเอ็มพีจัดอยู่ในลำดับชั้นแอปพลิเคชันของ โอเอสไอโมเดล เป็นโปรโตคอลสำหรับตรวจสอบและบริหารจัดการอุปกรณ์เครือข่ายประเภท Local Area Network (LAN) หรือ Wide Area Network (WAN) เอสเอ็นเอ็มพีจะจัดเก็บข้อมูลและจัดการใน Management Information Base หรือ MIB ซึ่งเป็นฐานข้อมูลสำหรับการจัดการอุปกรณ์

2.3.2 NetFlow

NetFlow เป็นเทคโนโลยีที่อยู่ในอุปกรณ์เครือข่ายของ Cisco IOS เป็นเครื่องมือสำหรับใช้ในการตรวจสอบการรับส่งข้อมูลและเก็บสถิติข้อมูลในเครือข่ายเหล่านั้น ผู้ดูแลระบบสามารถใช้ข้อมูลเหล่านั้นในการวิเคราะห์ นำไปสู่การพัฒนาระบบเครือข่ายให้มีประสิทธิภาพมากยิ่งขึ้น [8]

2.3.3 Cisco Discovery Protocol (CDP)

CDP เป็นโปรโตคอลของ Cisco เป็นเครื่องมือที่ช่วยให้อุปกรณ์เครือข่ายสามารถแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์เครือข่าย Cisco ที่อยู่ติดกัน ทำให้สามารถเก็บข้อมูลสถานะของของอุปกรณ์เครือข่ายได้ [9]

2.3.4 Secure Shell (SSH)

Secure Shell เป็นโปรโตคอลที่ออกแบบมาสำหรับการเชื่อมต่อกับอุปกรณ์ต่าง ๆ เช่น คอมพิวเตอร์ เน็ตเวิร์กเราเตอร์ เน็ตเวิร์กสวิตช์ เป็นต้น โดยจะมีการเข้ารหัสข้อมูลในระหว่างการสื่อสารทำให้การเชื่อมต่อมีความปลอดภัยมากยิ่งขึ้น ในโครงการจะใช้ SSH สำหรับเชื่อมต่ออุปกรณ์กับตัวควบคุมเพื่อเปิดช่องทางตั้งค่าอุปกรณ์ [10]

2.4 เทคโนโลยีที่ใช้ในการพัฒนา

2.4.1 Python

ไพทอนเป็นภาษาโปรแกรมมิ่งระดับสูง มีไวยากรณ์คำสั่งพื้นฐานที่เข้าใจและใช้งานง่าย รองรับการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) หรือการเขียนโปรแกรมเชิงฟังก์ชัน (Functional Programming)

ไพทอนสามารถทำงานได้บนหลายระบบปฏิบัติการ เช่น Linux, Unix, Windows เป็นต้น อีกทั้งยังเป็น Open-source ที่เข้าถึงได้ง่าย ทำให้มีไลบรารีให้ใช้อยู่เป็นจำนวนมากเพิ่มความสะดวกสบายในการทำงานที่หลากหลาย [11]

2.4.2 MongoDB

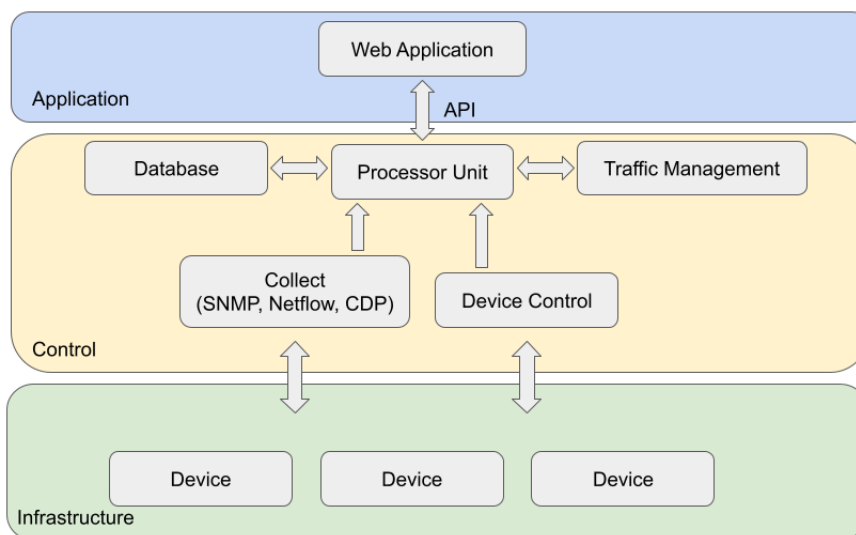
มอนโกดีบี เป็นฐานข้อมูลแบบ Open-source Document ประเภทหนึ่ง โดยเป็นฐานข้อมูลแบบ NoSQL มีจุดเด่นที่ทำงานได้ไว เหมาะกับฐานข้อมูลที่มีขนาดใหญ่ และไม่ซับซ้อน การเก็บข้อมูลจะเก็บคีย์ และข้อมูลเอาไว้โดยต้องมีคีย์หลักที่เป็นเอกลักษณ์เป็นหน่วยพื้นฐานของข้อมูล เนื่องจากการจัดเก็บข้อมูลในรูปแบบ BSON (Binary JSON) ซึ่งมีรูปแบบคล้าย JSON (JavaScript Object Notation) แต่เก็บข้อมูลได้หลากหลายกว่า [12]

2.4.4 Netmiko

เน็ตมิโกะ เป็น SSH Python ไลบรารีที่ช่วยให้กระบวนการเชื่อมต่ออุปกรณ์เครือข่ายผ่าน Secure Shell ทำได้ง่ายขึ้น ในที่นี้จะใช้เน็ตมิโกะสำหรับส่งคำสั่งการตั้งค่าอุปกรณ์เครือข่ายตามที่ตัวควบคุมของเอสดีเอ็นต้องการ [13]

2.5 ระบบควบคุมต้นแบบ

โครงการนี้ได้นำระบบต้นแบบซึ่งออกแบบตามสถาปัตยกรรมแบบเอสดีเอ็น ทำหน้าที่เป็นควบคุม เชื่อมต่อและเก็บข้อมูลอุปกรณ์เครือข่ายลงฐานข้อมูล พร้อมจัดเตรียมระบบ API สำหรับผู้ใช้ให้สามารถดึงข้อมูลที่ระบบบันทึกในฐานข้อมูลนำมาใช้งานต่อได้อย่างสะดวก [14]



รูปที่ 2.2 องค์ประกอบภาพรวมระบบต้นแบบ

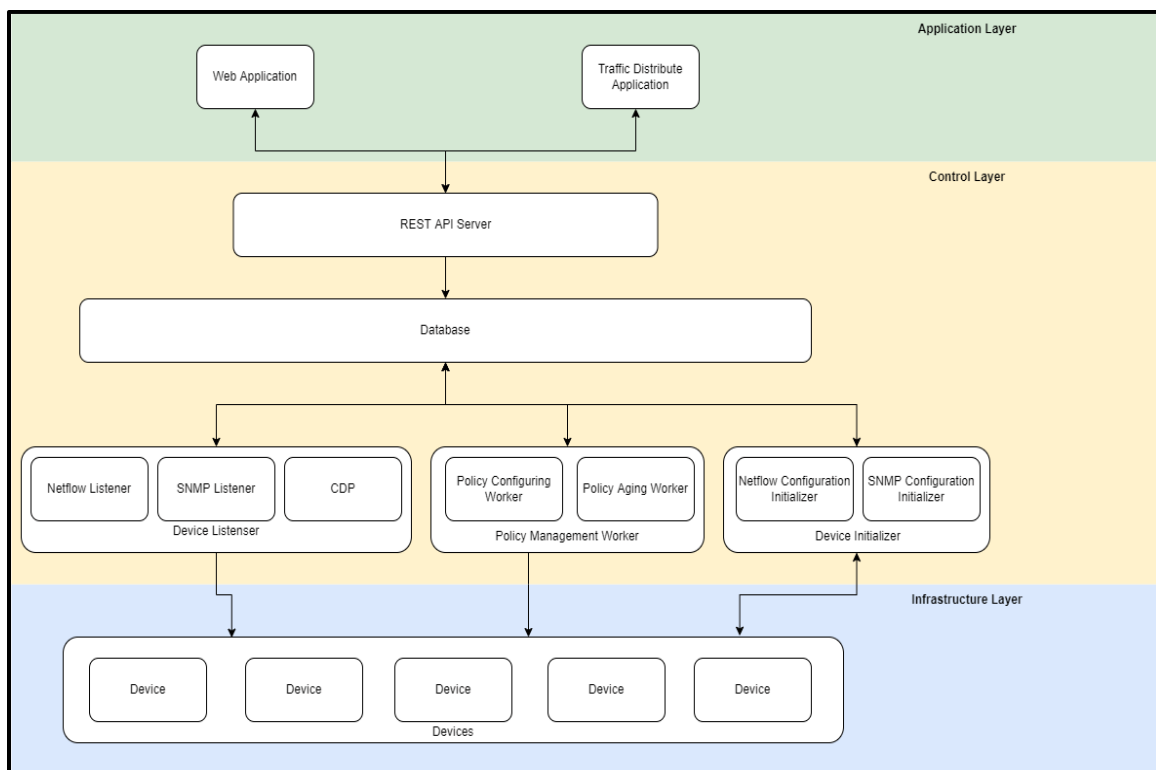
บทที่ 3

แนวคิดและการดำเนินงาน

3.1 ขั้นตอนการดำเนินงาน

1. ศึกษาโครงสร้างและการจัดการระบบเครือข่ายตามสถาปัตยกรรมแบบเอสดีเอ็น
2. ติดตั้งเซิร์ฟเวอร์และจัดหาอุปกรณ์สำหรับการทดลอง
3. ศึกษา ติดตั้งแอปพลิเคชันสำหรับการจัดการอุปกรณ์เครือข่ายที่ไม่รองรับการทำงานตามสถาปัตยกรรมเอสดีเอ็นผ่านเอสดีเอ็นคอนโทรลเลอร์ที่ถูกพัฒนาขึ้น
4. ศึกษาแนวคิดวิธีการสำรวจจัดเก็บข้อมูลเครือข่าย
5. ศึกษาแนวทางกลไกการกระจายแพทช์เฟิร์มแวร์บนเครือข่ายตามที่ใช้กำหนดผ่านทางส่วนติดต่อกับผู้ใช้งาน
6. ปรับปรุงการทำงานของระบบต้นแบบเพื่อเตรียมพร้อมในการนำมาใช้พัฒนาใหม่
7. พัฒนากลไกการกระจายแพทช์เฟิร์มแวร์
8. พัฒนาเว็บแอปพลิเคชันสำหรับแสดงผล
9. ทดสอบการใช้งานและสรุปผล

3.2 ภาพรวมระบบ



รูปที่ 3.1 องค์ประกอบภาพรวมของระบบ

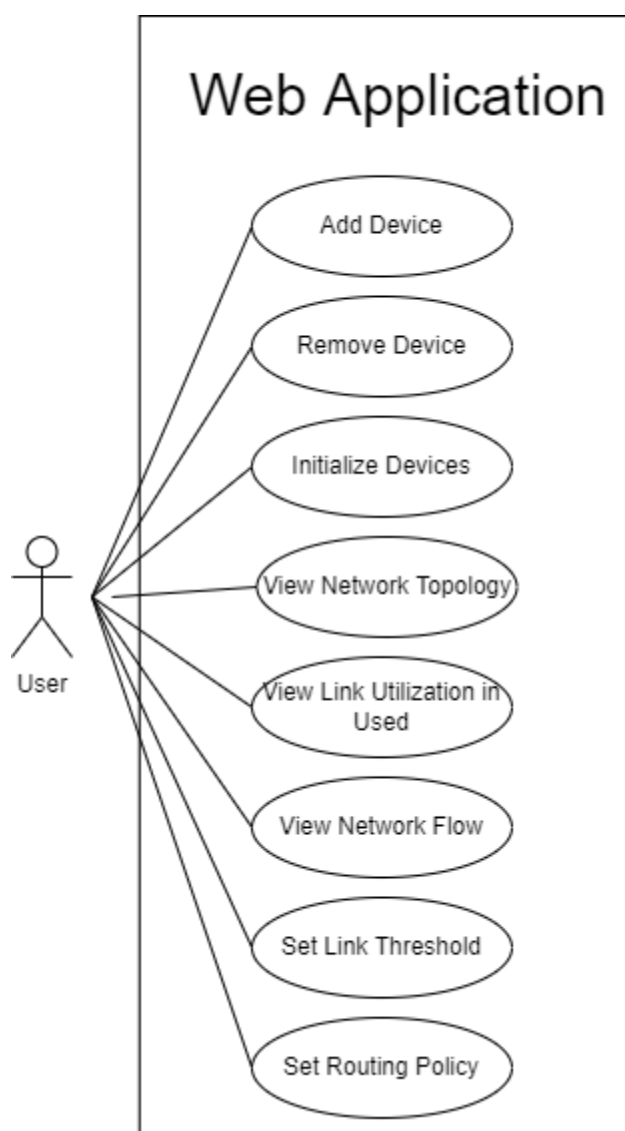
ระบบประกอบไปด้วยอุปกรณ์เครือข่ายที่เชื่อมต่อเข้ากับตัวควบคุม ซึ่งตัวควบคุมจำเป็นต้องสามารถ SSH ไปยังอุปกรณ์สำหรับส่งคำสั่งตั้งค่า เพื่อเปิดใช้งาน SNMP และ NetFlow สำหรับเก็บข้อมูลเครือข่าย และส่งคำสั่งตั้งค่า Policy Based Routing สำหรับการกระจายแพทर्फิก

ข้อมูลที่เก็บมาจากอุปกรณ์เครือข่ายจะถูกบันทึกลงในฐานข้อมูล โดยจะมีหน้าเว็บแอปพลิเคชันที่จะดึงข้อมูลจากฐานข้อมูลผ่านทาง REST API ตัวแอปพลิเคชันจะทำหน้าที่แสดงข้อมูลจากระบบเครือข่ายให้ผู้ใช้สามารถดูและทำความเข้าใจได้ง่าย

ในโครงงานนี้ได้แบ่งระบบออกเป็น 3 ส่วน คือ ส่วนตัวควบคุม (Controller) ส่วนแสดงผลสำหรับผู้ใช้งาน (Web Application) และแอปพลิเคชันสำหรับการกระจายแพทर्फิก (Traffic Distribution Application)

3.3 ส่วนแสดงผลสำหรับผู้ใช้งาน (Web Application)

เว็บไซต์ถูกสร้างขึ้นมาเพื่อจุดประสงค์ให้ผู้ใช้งานทั่วไปสามารถใช้ตัวควบคุม และดูภาพรวมของระบบเครือข่ายได้ง่ายยิ่งขึ้น ซึ่งผู้ใช้งานสามารถเพิ่ม-ลบอุปกรณ์ ส่งคำสั่งให้อุปกรณ์ส่งข้อมูลมายังตัวควบคุม ดูโทโพโลยีภาพรวมของระบบเครือข่ายซึ่งประกอบไปด้วยอุปกรณ์ที่เชื่อมต่อกัน ดูโพล์ว์ที่เดินทางผ่านแต่ละลิงก์ ตั้งค่า Threshold สำหรับทำ Traffic Distribution และสามารถตั้ง Routing Policy ตามที่ผู้ใช้งานต้องการได้ หน้าเว็บถูกแบ่งออกเป็น 3 หน้าตามการใช้งานดังนี้



รูปที่ 3.2 Use Case Diagram ของ Web Application

3.3.1 Home

แสดงหน้าโทโพลยีของเครือข่ายจากอุปกรณ์ในระบบ ผู้ใช้สามารถดูข้อมูลโพล์ว์ที่เดินผ่านแต่ละลิงก์ สามารถกำหนดนโยบายเพื่อปรับเปลี่ยนเส้นทางตามที่ต้องการ และคูนโยบายที่กำลังถูกใช้ในระบบเครือข่ายได้



รูปที่ 3.3 หน้า Home สำหรับแสดงโทโพลยี โพล์ว์และจัดการนโยบายปรับเปลี่ยนเส้นทาง

3.3.2 Devices

ในหน้านี้ผู้ใช้สามารถดูข้อมูล Management IP อุปกรณ์ในระบบ ค่าสถานะเชื่อมต่อ SSH และการทำงานของ SNMP ของอุปกรณ์แต่ละตัวได้ ผู้ใช้สามารถเพิ่ม ลบ อุปกรณ์ได้ในหน้านี้ การเพิ่มอุปกรณ์เป็นการทำให้ตัวควบคุมรู้จักกับอุปกรณ์เครือข่ายตัวดังกล่าวเพื่อที่จะส่งการหรือเก็บข้อมูลเครือข่าย ผู้ใช้สามารถกรอกค่าพารามิเตอร์ที่จำเป็นสำหรับการทำ SSH และ SNMP ลงในฟอร์มทางซ้ายมือซึ่งประกอบไปด้วยข้อมูล Management IP System Type (ยี่ห้ออุปกรณ์เครือข่าย) Username/Password/Port สำหรับการเชื่อมต่อ Enable Secret และ SNMP Community String/Port เมื่อผู้ใช้กดปุ่ม Add Device ตัวควบคุมจะทดลองเชื่อมต่อ SSH ไปยังอุปกรณ์ดังกล่าว ถ้าผู้ใช้กรอกข้อมูลผิดส่งผลให้ตัวควบคุมไม่สามารถ SSH ไปยังอุปกรณ์ได้ และอุปกรณ์ดังกล่าวจะไม่ถูกเพิ่มเข้าสู่ระบบ แต่ถ้าตัวควบคุมสามารถ SSH ไปยังอุปกรณ์ได้จะขึ้นข้อความแจ้งเตือนดังรูปที่ 3.4

The screenshot shows a web interface with a top navigation bar containing 'Home', 'Devices', and 'Initialization'. On the left, a 'Devices List' panel displays five entries: R5.pcn (Management IP: 192.168.59.1), R6.pcn (192.168.69.1), R8.pcn (192.168.78.2), R9.pcn (192.168.69.2), and R7.pcn (192.168.79.1). Each entry has 'SSH' and 'SNMP' status buttons. On the right, the 'R5.pcn' configuration form is visible, with fields for Management IP (192.168.59.1), System Type (cisco_ios), SSH Username (cisco), SSH Password (cisco), SSH Secret (cisco), SSH Port (22), and SNMP Version (2c).

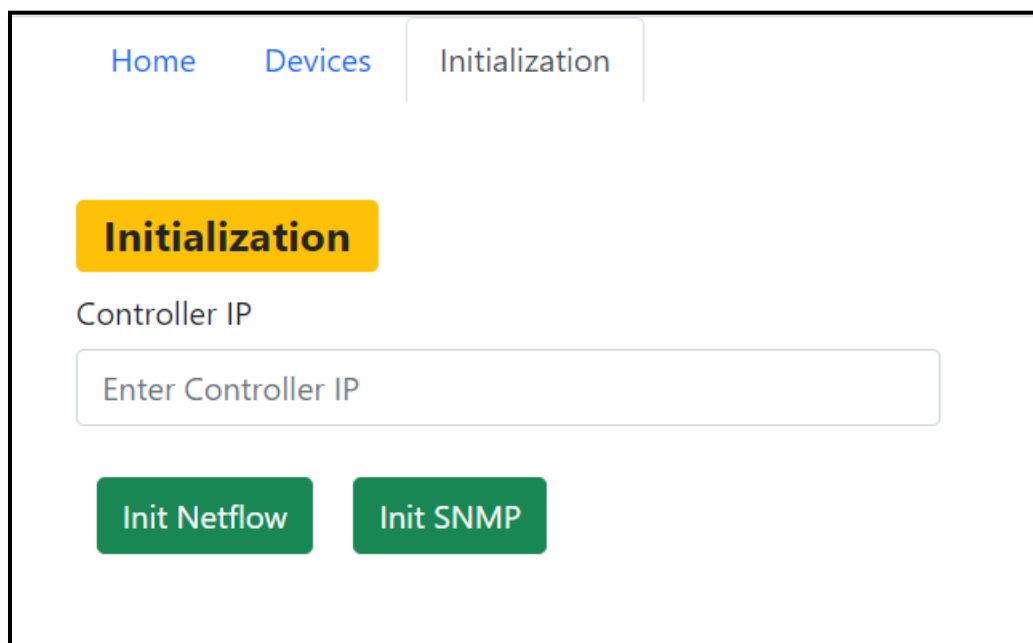
รูปที่ 3.4 หน้า Device สำหรับ เพิ่ม ลบ แก้ไข และดูสถานะเชื่อมต่ออุปกรณ์

This screenshot shows the 'New Device Info' configuration form, which includes fields for device name (cisco), SSH Password, SNMP Community String (public), and SNMP Port (161). An 'Add Device' button is at the bottom right. A white modal dialog box is centered on the screen, displaying a green checkmark icon, the text 'Device was added', and an 'Okay' button.

รูปที่ 3.5 ตัวควบคุมติดต่อกับอุปกรณ์ได้ และอุปกรณ์ถูกเพิ่มเข้าสู่ระบบ

3.3.3 Initialization

หลังจากเพิ่มอุปกรณ์เข้าสู่ระบบแล้ว ผู้ใช้จำเป็นต้องส่งคำสั่งตั้งค่า SNMP และ NetFlow ซึ่งจะเป็นการเปิดช่องทางให้อุปกรณ์ส่งข้อมูลของตัวเองมายังตัวควบคุม ข้อมูลทั้งหมดจะถูกรวบรวม บันทึกลงฐานข้อมูลและแสดงผ่านทางหน้า Home โดยผู้ใช้อาจต้องกรอกไอพีของตัวควบคุมและกดปุ่ม Init SNMP/Netflow



Home Devices Initialization

Initialization

Controller IP

Enter Controller IP

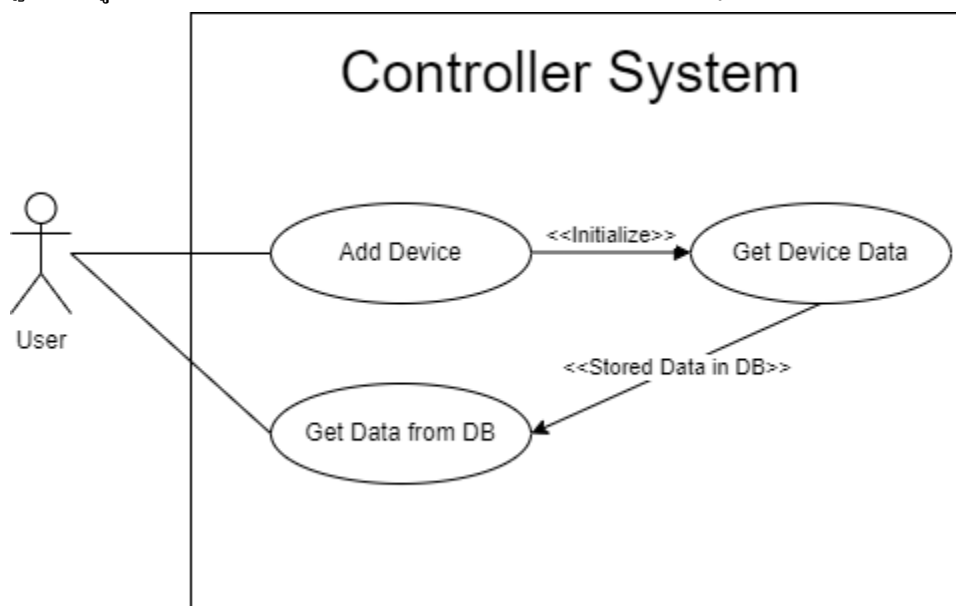
Init Netflow Init SNMP

รูปที่ 3.6 หน้า Initialization สำหรับเปิดการเชื่อมต่อให้อุปกรณ์ส่งข้อมูลมายังตัวควบคุม

3.4 ส่วนตัวควบคุม (Controller)

3.4.1 Use Case Diagram

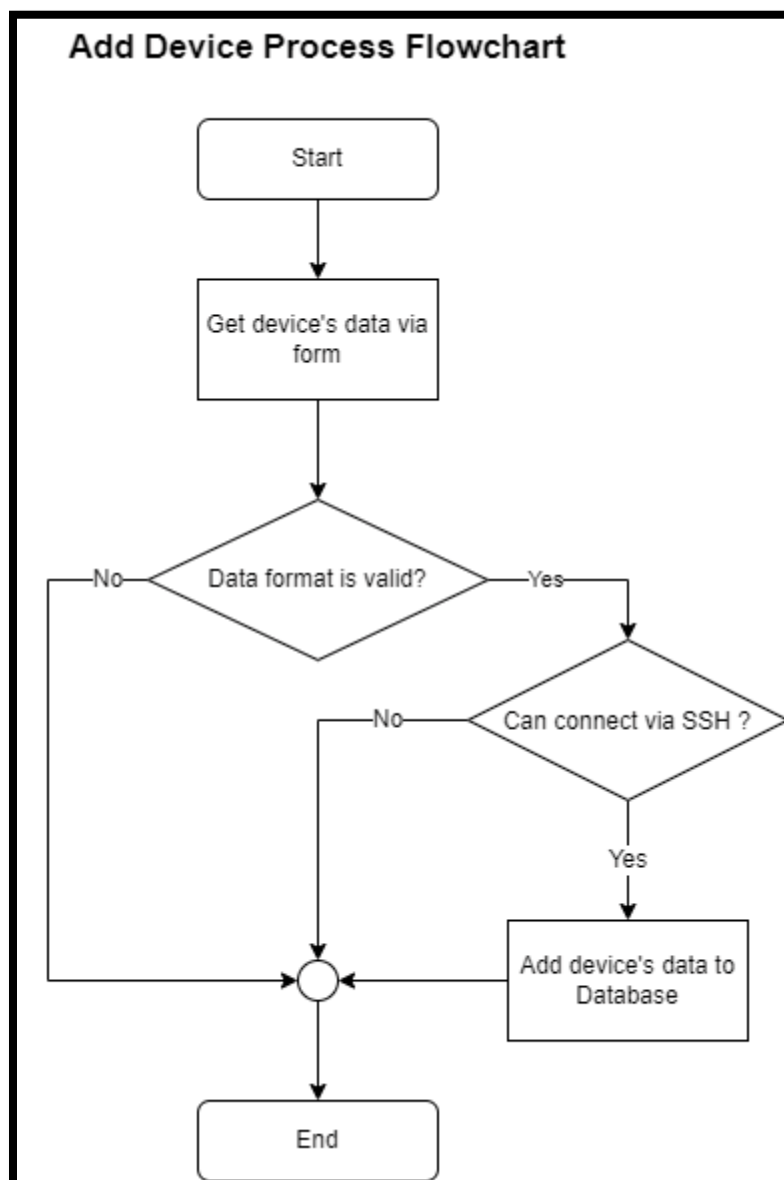
ตัวควบคุมจะทำหน้าที่เป็นเซิร์ฟเวอร์ที่เชื่อมต่อกับอุปกรณ์เครือข่ายทั้งหมด โดยผู้ใช้ต้องส่งคำสั่งเพิ่มอุปกรณ์ผ่านทางเว็บไซต์แอปพลิเคชัน หรือเอพีไอของระบบ เพื่อให้ตัวควบคุมพร้อมต่อการรับค่าข้อมูลต่าง ๆ ที่อุปกรณ์เครือข่ายส่งมา และเก็บข้อมูลที่จำเป็นลงฐานข้อมูลเพื่อให้พร้อมแก่การนำไปใช้ต่อในแอปพลิเคชันอื่น ๆ ผ่านทางเอพีไอ



รูปที่ 3.7 Use Case Diagram ของตัวควบคุม

3.4.2 ขั้นตอนการเพิ่มอุปกรณ์

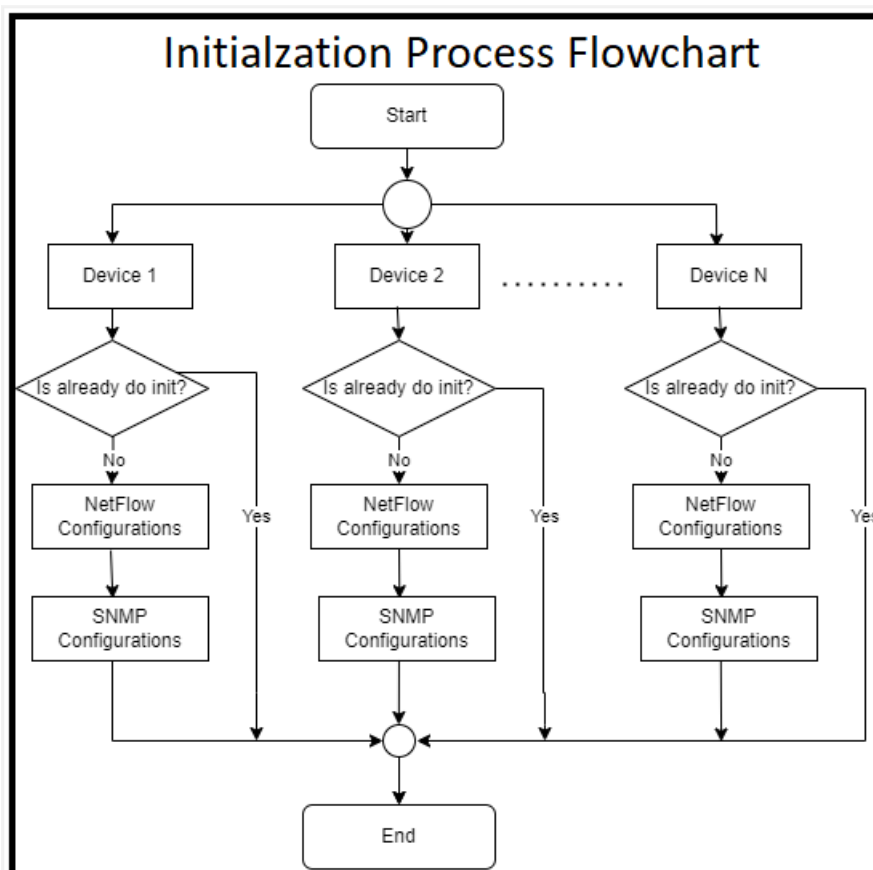
กระบวนการเพิ่มอุปกรณ์จะมีกระบวนการดังภาพที่ 3.8 เมื่อได้ข้อมูลอุปกรณ์จากที่ผู้ใช้อกรอกผ่านฟอร์มของเว็บแอปพลิเคชันตัวควบคุมจะทดลองเชื่อมต่อกับอุปกรณ์ผ่านทาง SSH ก่อนเพื่อทดสอบว่าอุปกรณ์ดังกล่าวสามารถเชื่อมต่อและตั้งค่าผ่าน SSH ได้จริง ก่อนที่จะบันทึกข้อมูลของอุปกรณ์นั้นลงฐานข้อมูล



รูปที่ 3.8 ขั้นตอนกระบวนการเพิ่มอุปกรณ์

3.4.3 ขั้นตอนการ Initialize อุปกรณ์ในระบบ

กระบวนการ Initialize จะมีกระบวนการดังภาพที่ 4 โดยตัวควบคุมจะตรวจสอบว่าอุปกรณ์แต่ละตัวมีการทำ Initialize แล้วหรือไม่ หากยังไม่ทำตัวควบคุมจะส่งคำสั่งเปิดการใช้งาน SNMP และ NetFlow ผ่านทาง Netmiko ให้อุปกรณ์ส่งข้อมูลให้ตัวควบคุมหากอุปกรณ์ทำ Initialize ไปแล้วก็ไม่มีความจำเป็นต้องส่งคำสั่งซ้ำ ลดความล่าช้าของระบบ



รูปที่ 3.9 ขั้นตอน Initialize เพื่อให้อุปกรณ์ส่งข้อมูลมายังตัวควบคุม

3.4.4 ฐานข้อมูล

ข้อมูลอุปกรณ์เครือข่ายที่ส่งมาให้ตัวควบคุมจะถูกบันทึกลงในฐานข้อมูลเพื่อให้ผู้ใช้สามารถดึงข้อมูลเหล่านั้นไปใช้ได้สะดวก ในโครงงานนี้ใช้ MongoDB ซึ่งจัดเก็บข้อมูลแบบ Binary JSON ซึ่งประกอบด้วยตารางที่เก็บข้อมูลดังต่อไปนี้

1. Device: เก็บข้อมูลอุปกรณ์ทั่วไปของอุปกรณ์เครือข่ายแต่ละตัวใช้ในการสร้างกราฟเครือข่าย ประกอบไปด้วย ไอพีสำหรับจัดการ (Management IP) เลขประจำอุปกรณ์ (Serial Number) รุ่นอุปกรณ์ สถานการณ์ทำงานของ SNMP/NetFlow/SSH ชื่ออุปกรณ์ ข้อมูลอินเตอร์เฟซ
2. Policy Flow Routing: เก็บข้อมูลนโยบาย (Policy) จากกระบวนการทำ Traffic Distribution เพื่อเปลี่ยนเส้นทางของโฟลว์ในระบบ ประกอบไปด้วยข้อมูล ชื่อนโยบาย ไอพีต้นทาง ไอพีปลายทาง พอร์ตต้นทาง พอร์ตปลายทาง ไวลด์การ์ดต้นทาง ไวลด์การ์ดปลายทาง นโยบายที่ถูกนำไปตั้งค่าให้อุปกรณ์ และเวลาในการลบนโยบายดังกล่าวหลังจากไม่มีโฟลว์ตามที่นโยบายกำหนดออกจากระบบ
3. Flow Stat: เก็บข้อมูลโฟลว์ที่วิ่งอยู่ในระบบ ใช้ในการทำ Traffic Distribution ซึ่งประกอบไปด้วย ไอพีต้นทาง ไอพีปลายทาง พอร์ตต้นทาง พอร์ตปลายทาง ชั้นเน็ตมาสก์ต้นทาง ชั้นเน็ตมาสก์ปลายทาง ขนาดของโฟลว์ และเวลาที่พบโฟลว์ดังกล่าวในระบบ
4. Link Utilization: เก็บข้อมูลลิงก์ของอุปกรณ์ที่เชื่อมต่อกันในระบบใช้ในการสร้างกราฟเครือข่ายและทำ Traffic Distribution ประกอบด้วยข้อมูล ไอพีอินเตอร์เฟซต้นทาง ไอพีอินเตอร์เฟซปลายทาง พอร์ตอินเตอร์เฟซต้นทาง พอร์ตอินเตอร์เฟซปลายทาง ปริมาณแพคเกจในลิงก์ เปอร์เซนต์แบนด์วิดท์ที่ถูกใช้ ชื่อโฟลว์ที่วิ่งอยู่ในลิงก์ และ Threshold สำหรับกระบวนการกระจายแพคเกจของลิงก์นั้น

3.5 แอปพลิเคชันสำหรับการกระจายแพทไฟก (Traffic Distribution Application)

แอปพลิเคชันสำหรับการกระจายแพทไฟก เป็นแอปพลิเคชันที่ทำงานบนตัวควบคุมทำหน้าที่ตรวจจับลิงก์ที่มีเปอร์เซ็นต์แบนด์วิดท์สูงกว่าค่า Threshold ที่กำหนดไว้ในแต่ละลิงก์โดยมุ่งจัดการโฟลว์ที่มีขนาดใหญ่และปรากฏอยู่ในระบบเป็นเวลานาน (Long-lived Large Flows) พร้อมใช้ Policy Based Routing ในการตั้งค่าการย้ายเส้นทางใหม่ และมีฟังก์ชัน Aging Policy เพื่อลบนโยบายจัดการโฟลว์หากไม่มีโฟลว์ที่ตรงกับเงื่อนไขเข้ามาในระบบเข้ามาในระบบเป็นระยะเวลาหนึ่ง เพื่อที่จะคงสภาพการทำงานของเครือข่ายให้เป็นปัจจุบันที่สุด โดยแอปพลิเคชันนี้จะทำงานได้ดีกับแพทไฟกประเภท CBR (Constant bit rate) ในส่วนของแพทไฟกประเภท VBR (Variable bit rate) แอปพลิเคชันยังสามารถรองรับได้ ถ้าระยะเวลาโฟลว์ที่ปรากฏในระบบไม่น้อยกว่าคาบการเก็บข้อมูลของตัวควบคุม มิเช่นนั้นจะมีโอกาสหลุดการตรวจจับได้

3.5.1 Traffic Distribution

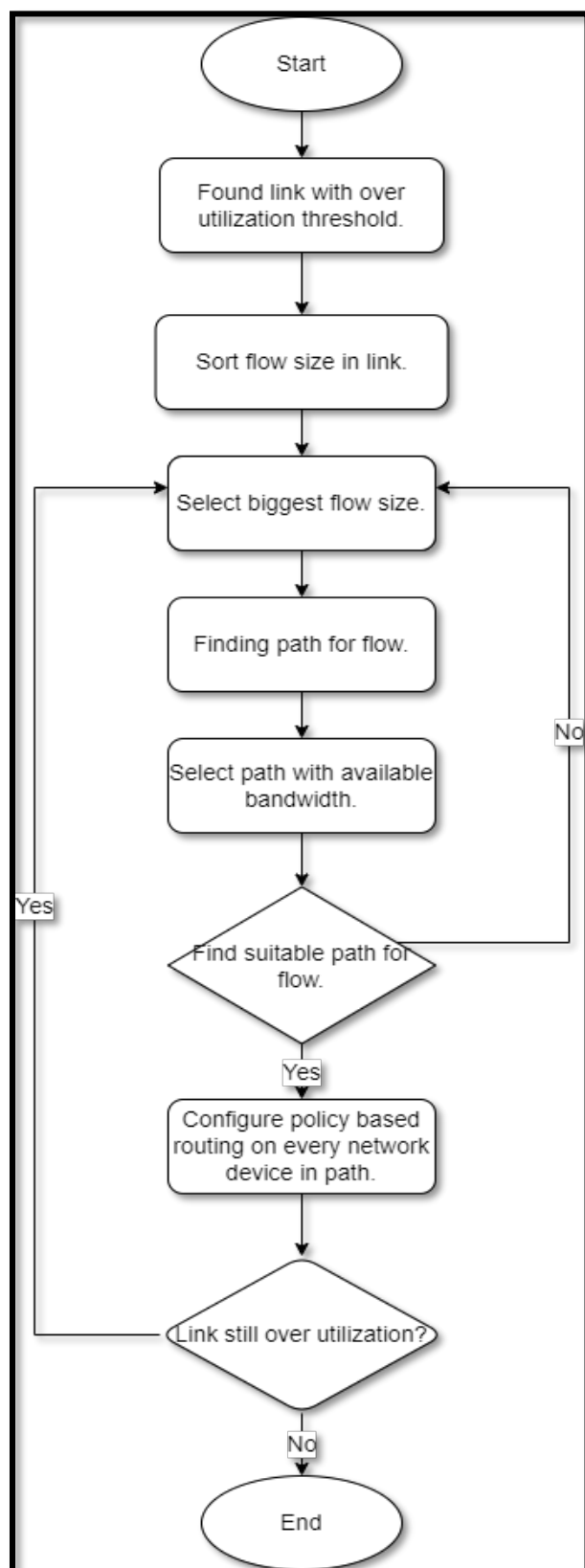
กระบวนการ Initialize จะมีกระบวนการดังภาพที่ 3.10 เมื่อพบลิงก์ที่มีการใช้งานแบนด์วิดท์สูงกว่าค่าที่ลิงก์กำหนดไว้จะทำการตรวจสอบโฟลว์ที่อยู่ในลิงก์และเรียงลำดับขนาดโฟลว์เหล่านั้น เริ่มจากโฟลว์ที่มีขนาดใหญ่ที่สุดและไม่ใช้โฟลว์สำหรับการส่งข้อมูล SNMP และ SSH เนื่องจากเป็นโฟลว์ที่มีความสำคัญต่อการทำงานของตัวควบคุมจึงไม่สมควรในการปรับเปลี่ยนเส้นทางใหม่ หาเส้นทางที่เป็นไปได้ทั้งหมดของโฟลว์ผ่านเอพียอของตัวควบคุมซึ่งเรียกใช้งานโมดูล NetworkX ฟังก์ชัน all_simple_paths ผลลัพธ์จะแสดงเส้นทางที่เป็นไปได้ทั้งหมดจากโหนดต้นทางไปยังโหนดปลายทางโดยจะไม่ใช้โหนดซ้ำในเส้นทางเดียวกัน ทำให้สามารถมั่นใจว่าจะไม่เกิดลูปในเส้นทาง หลังจากนั้นจะประมาณค่าแบนด์วิดท์คงเหลือหลังจากย้ายโฟลว์ดังกล่าวแล้วจะทำให้ลิงก์ในเส้นทางใหม่มีการใช้งานแบนด์วิดท์สูงกว่าที่กำหนดไว้หรือไม่ และเลือกใช้เส้นทางที่มีค่าแบนด์วิดท์คงเหลือในลิงก์สูงที่สุดจากลิงก์ที่เกิดคอขวด ถ้าไม่เจอเส้นทางตามเงื่อนไขจะเลือกโฟลว์ใหม่ที่มีขนาดรองลงมา เมื่อได้โฟลว์และเส้นทางที่จะย้ายแล้วตัวแอปพลิเคชันจะส่งคำสั่งตั้งค่า Policy Based Routing ไปยังอุปกรณ์ในเส้นทางทุกตัวเพื่อปรับเปลี่ยนเส้นทางของโฟลว์ตามที่กำหนด หลังจากนั้นจะทำการตรวจสอบว่าลิงก์ดังกล่าวยังมีการใช้งานแบนด์วิดท์สูงกว่าค่าที่ลิงก์กำหนดหรือไม่ ถ้ามีจะเลือกโฟลว์ที่มีขนาดใหญ่รองลงมาเพื่อปรับเปลี่ยนเส้นทาง

3.5.2 Policy

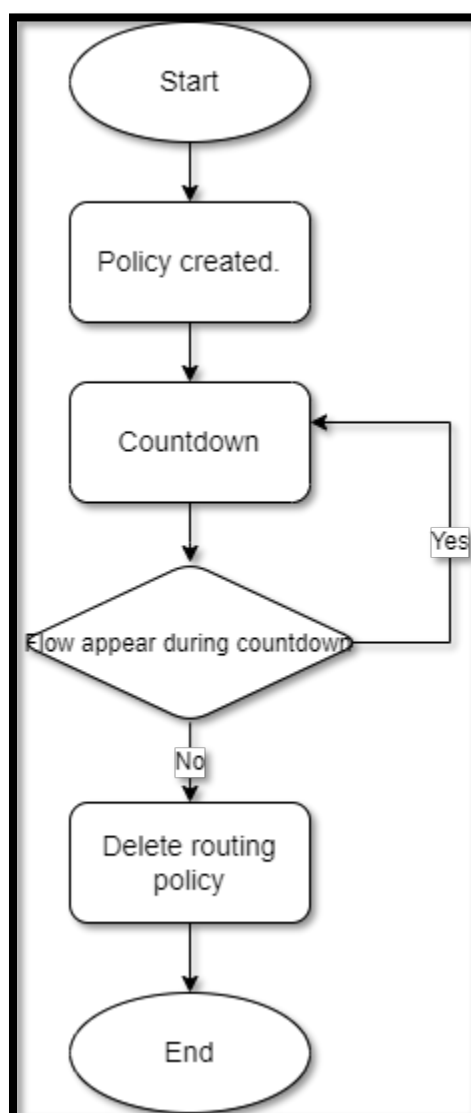
นโยบายสำหรับปรับเปลี่ยนเส้นทางโดยตั้งค่าตามข้อมูลโพล์และเส้นทาง โพล์ที่มี ไอพีต้นทาง ไอพีปลายทาง พอร์ตต้นทาง และพอร์ตปลายทางเหมือนกันจะจัดเป็นโพล์เดียวกันและใช้งานนโยบายที่เหมือนกัน

3.5.3 Aging Policy

กระบวนการ Aging จะมีกระบวนการดังภาพที่ 3.11 โดยมีเป้าหมาย ลบนโยบาย (Policy) ที่ไม่มีการใช้งานเป็นระยะเวลาหนึ่งซึ่งเป็นการคงสภาพการทำงานของเครือข่ายให้เป็นปัจจุบันที่สุด เมื่อมีการสร้างนโยบายสำหรับเปลี่ยนเส้นทางโพล์เกิดขึ้นจะมีการตรวจสอบว่ายังมีโพล์ที่ตรงตามเงื่อนไขในระบบหรือไม่ ถ้าไม่พบจะเริ่มจับเวลา หากโพล์ดังกล่าวไม่ปรากฏจนครบเวลาระบบจะส่งคำสั่งลบนโยบายดังกล่าวออกจากอุปกรณ์



รูปที่ 3.10 ขั้นตอนการกระจายแพคเกจ



รูปที่ 3.11 ขั้นตอนการทำ Aging Policy เพื่อลบนโยบายที่โฟลว์ไม่ปรากฏเป็นระยะเวลาหนึ่ง

บทที่ 4

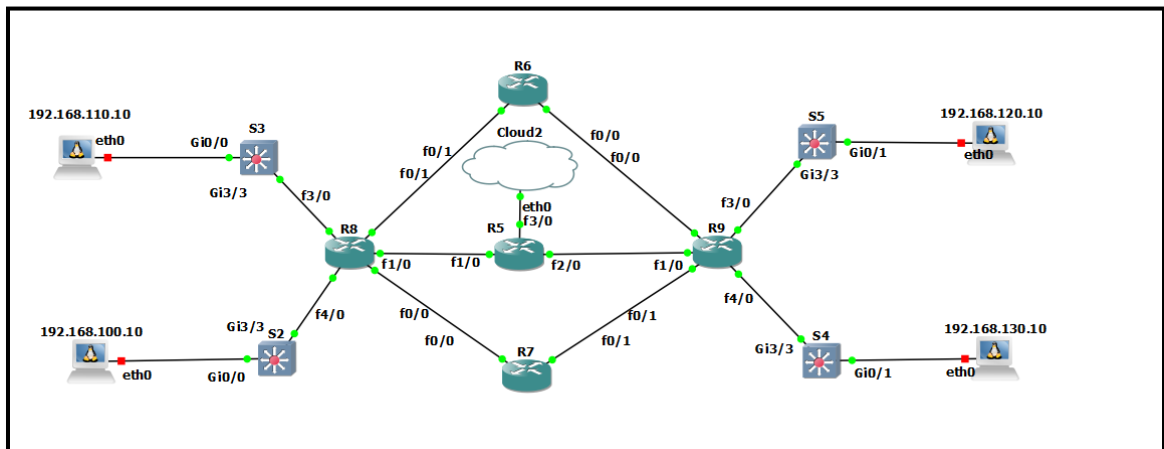
ผลการดำเนินการ

ในการทดลองนี้จะเป็นการจำลองการทำงานแอปพลิเคชันที่พัฒนาขึ้นในบทที่ 3 ซึ่งประกอบไปด้วย เว็บแอปพลิเคชันสำหรับแสดงผลเครือข่าย และแอปพลิเคชันสำหรับการกระจายแพทช์ฟลัก ซึ่งจะทดสอบว่าแอปพลิเคชันสามารถทำงาน และแสดงผลข้อมูลได้ถูกต้องตามความเป็นจริง โดยทดลองสร้างเครือข่ายในเซิร์ฟเวอร์ GNS3 เชื่อมต่ออุปกรณ์เครือข่ายเหล่านั้นกับตัวควบคุม เพิ่มอุปกรณ์เข้าสู่ระบบ เปิดช่องทางการรับข้อมูล SNMP และ NetFlow ทดลองสร้างโพล์เข้าสู่ระบบโดย Iperf3 และสังเกตการแสดงผล

4.1 เว็บแอปพลิเคชันสำหรับแสดงผลและควบคุมอุปกรณ์

4.1.1 Network Topology

อุปกรณ์เครือข่ายถูกเชื่อมต่อกันตามภาพที่ 4.1 ตั้งอยู่ในเครือข่าย 192.168.0.0/16 ตัวควบคุมจะตั้งอยู่ใน Cloud Network ซึ่งมีไอพี คือ 10.50.34.15/24 โดยมีการตั้งค่าอุปกรณ์เป็นไปตามตารางที่ 4.1



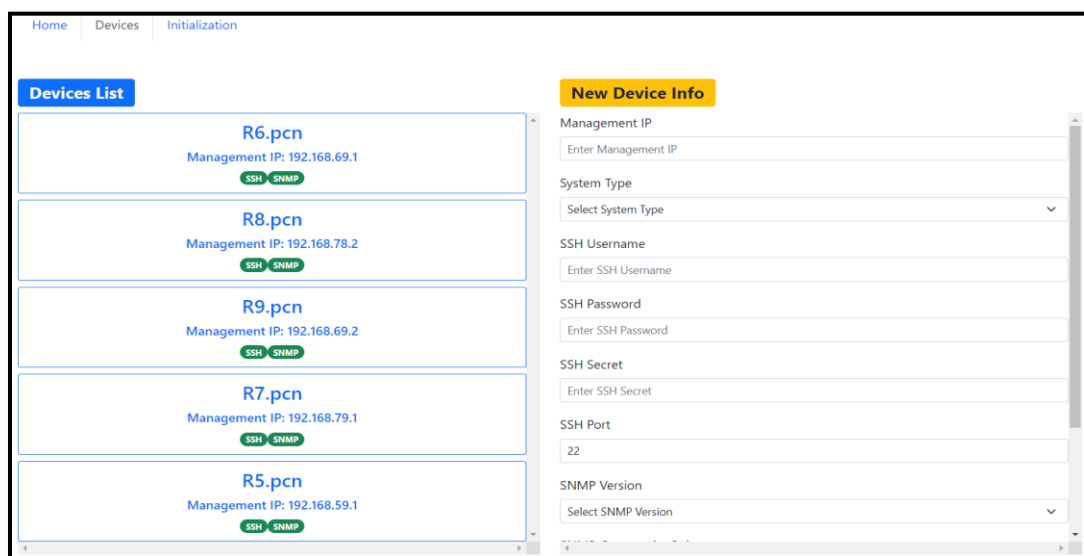
รูปที่ 4.1 โทโพโลยีที่ใช้ทดสอบ ตัวควบคุมอยู่ใน Cloud Network

ตารางที่ 4.1 ข้อมูลชื่ออุปกรณ์ ชื่อ-ขนาดพอร์ต และไอพีของอุปกรณ์เครือข่าย

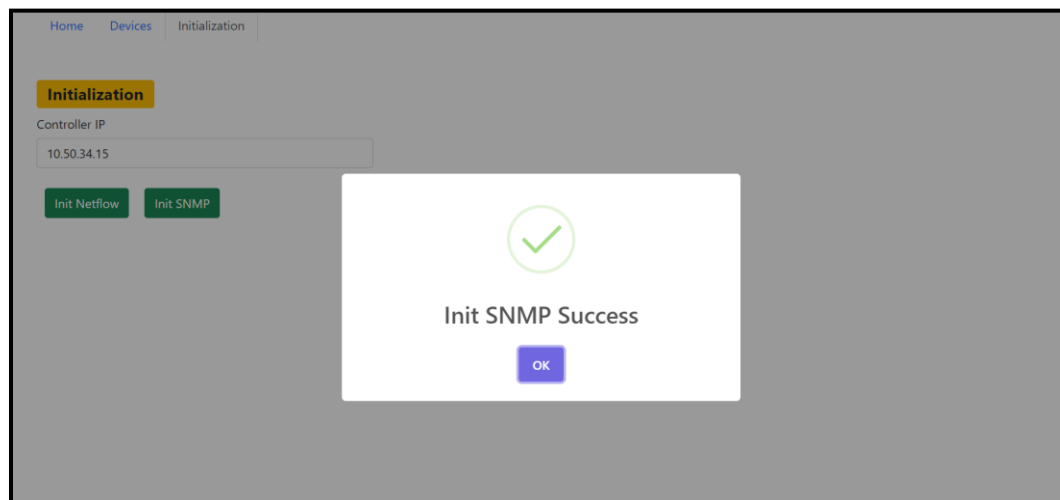
Device Name	Interface	Port Bandwidth	Network IP/Prefix
R5	F1/0	100 Kbit/sec	192.168.58.1/24
	F2/0	100 Kbit/sec	192.168.59.2/24
	F3/0	100 Kbit/sec	DHCP
R6	F0/0	10 Kbit/sec	192.168.69.1/24
	F0/1	10 Kbit/sec	192.168.68.1/24
R7	F0/0	10 Kbit/sec	192.168.78.1/24
	F0/1	10 Kbit/sec	192.168.79.1/24
R8	F0/0	10 Kbit/sec	192.168.78.2/24
	F0/1	10 Kbit/sec	192.168.68.2/24
	F1/0	100 Kbit/sec	192.168.58.2/24
	F3/0	100 Kbit/sec	192.168.110.1/24
	F4/0	100 Kbit/sec	192.168.100.1/24
R9	F0/0	10 Kbit/sec	192.168.69.2/24
	F0/1	10 Kbit/sec	192.168.79.2/24
	F1/0	100 Kbit/sec	192.168.59.2/24
	F3/0	100 Kbit/sec	192.168.120.1/24
	F4/0	100 Kbit/sec	192.168.130.1/24
Iperf-Client1	NIC		192.168.110.10/24
Iperf-Client2	NIC		192.168.100.10/24
Iperf-Server1	NIC		192.168.120.10/24
Iperf-Server2	NIC		192.168.130.10/24
Controller	NIC		10.50.34.15/24

4.1.2 Adding Device and Initialization

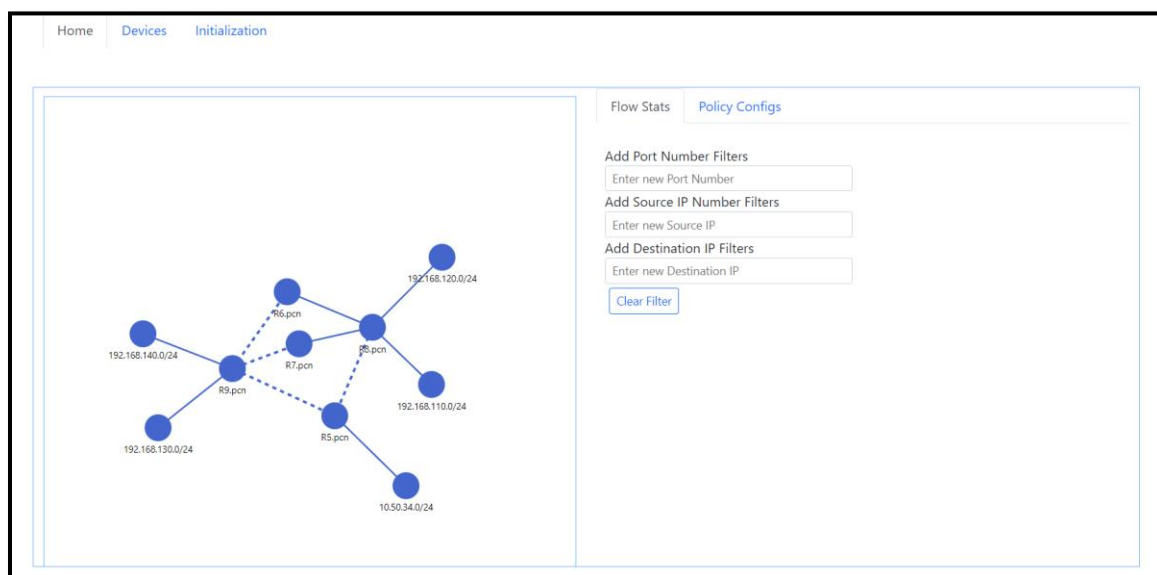
หลังจากตั้งค่าให้อุปกรณ์เครือข่ายสามารถเชื่อมต่ออุปกรณ์กับตัวควบคุมผ่านทาง SSH ได้แล้วจะเป็นการเพิ่มอุปกรณ์สู่ระบบเพื่อให้ตัวควบคุมรู้จักระบบเหล่านั้นและส่งข้อมูล อุปกรณ์เครือข่ายมายังตัวควบคุมเพื่อนำข้อมูลเหล่านั้นมาแสดงผลหรือนำไปใช้ในแอปพลิเคชันกระจายแพคเกจต่อไป เมื่อทำการเพิ่มอุปกรณ์และเปิดช่องทางให้อุปกรณ์ส่งข้อมูลมายังตัวควบคุมแล้วเว็บแอปพลิเคชันจะมีการแสดงข้อมูลโทโปโลยีตามภาพที่ 4.2 และ 4.3 เมื่อผู้ใช้งานกดลิงก์หน้าเว็บจะแสดงข้อมูลโพล์ ขนาดลิงก์ และเปอร์เซ็นต์การใช้งานของลิงก์ตามรูปที่ 4.4



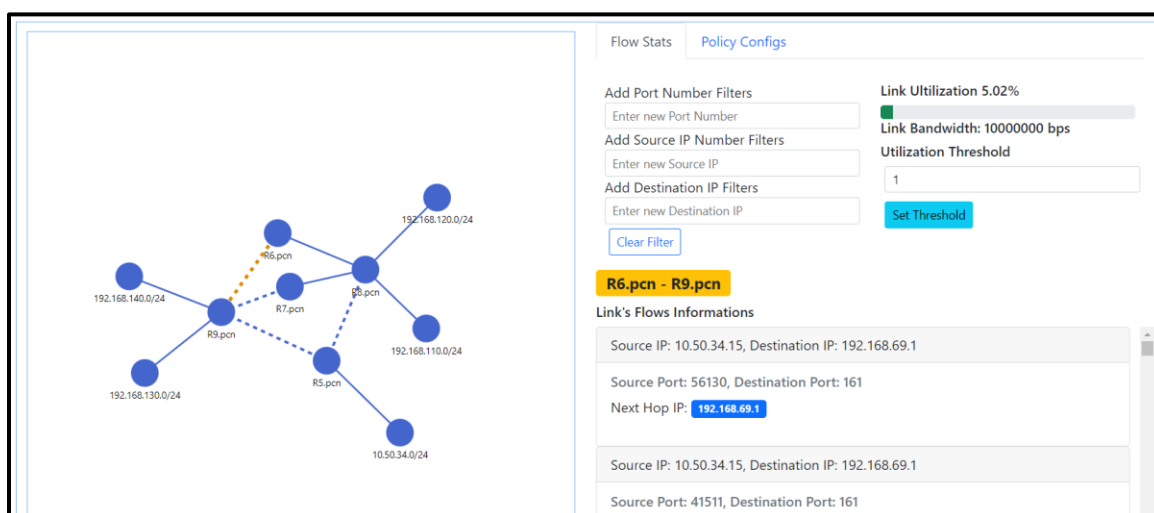
รูปที่ 4.2 ข้อมูลอุปกรณ์ในระบบและสถานะการทำงาน



รูปที่ 4.3 อุปกรณ์ตั้งค่าให้ส่งข้อมูล SNMP มายังตัวควบคุมไอพี 10.50.34.15



รูปที่ 4.4 อุปกรณ์ การเชื่อมต่อ การไหลของโพลี และชั้นเน็ตแสดงออกมาได้ถูกต้อง



รูปที่ 4.5 ข้อมูลลิงก์ถูกแสดงบนหน้าเว็บ

4.1.3 Flow Filter and Policy Routing.

ทดลองสร้างแทรฟฟิกจากเครื่อง Iperf-Client 192.168.110/24 ไปยังเครื่อง Iperf-Server 192.168.130/24 กำหนดขนาดไฟล์ที่ 400 Kbit/sec ตามภาพที่ 4.6 ตรวจสอบโดยใช้ Wireshark ซึ่งเป็นโปรแกรมวิเคราะห์แพ็กเก็ตตามดักจับข้อมูลแทรฟฟิกที่ผ่านลิงก์ R8-R5 พบไฟล์ที่มีขนาด 400 Kbit/sec วิ่งอยู่ในระบบ ด้านการแสดงผลหน้าเว็บ หากมีการเลือกแสดงเฉพาะไฟล์ที่มีไอพีต้นทางคือ 192.168.110.10 จะเป็นไปตามภาพที่ 4.8 โดยไฟล์จะเคลื่อนที่ผ่านลิงก์ R8-R5-R9 มีการใช้แบนด์วิดท์ถึงประมาณ 0.46% จากแบนด์วิดท์ทั้งหมด 100 Mbit ซึ่งตรงกับค่าจาก Wireshark หลังจากนั้นจะทดลองสร้างนโยบายสำหรับเปลี่ยนเส้นทาง ถ้าหากพบไฟล์ที่มีไอพีต้นทางเป็น 192.168.110.10 และไอพีปลายทางเป็น 192.168.130.10 จะส่งไปในเส้นทาง R8-R6-R9 แทน ผลการทดลองเป็นไปตามภาพ 4.9

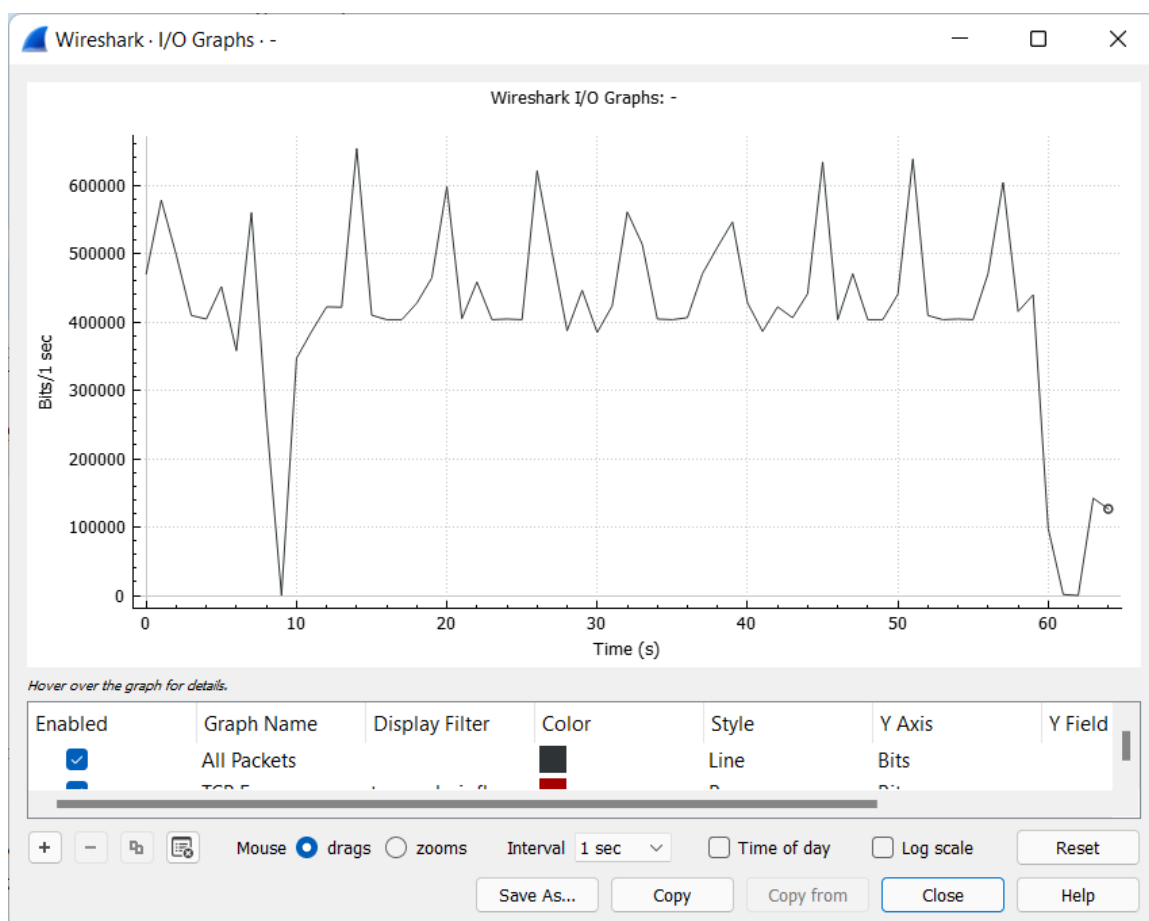
```

iperf Done.
root@Client-192:~#
root@Client-192:~#
root@Client-192:~#
root@Client-192:~#
root@Client-192:~#
root@Client-192:~# iperf3 -c 192.168.130.10 -p 5555 -u -t 30 -b 400K
Connecting to host 192.168.130.10, port 5555
[ 4] local 192.168.110.10 port 36864 connected to 192.168.130.10 port 5555
[ ID] Interval      Transfer    Bandwidth  Total Datagrams
[ 4] 0.00-1.00    sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 1.00-2.00    sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 2.00-3.00    sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 3.00-4.00    sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 4.00-5.00    sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 5.00-6.00    sec 56.0 KBytes 459 Kbits/sec 7
[ 4] 6.00-7.00    sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 7.00-8.00    sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 8.00-9.00    sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 9.00-10.00   sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 10.00-11.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 11.00-12.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 12.00-13.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 13.00-14.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 14.00-15.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 15.00-16.00  sec 56.0 KBytes 459 Kbits/sec 7
[ 4] 16.00-17.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 17.00-18.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 18.00-19.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 19.00-20.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 20.00-21.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 21.00-22.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 22.00-23.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 23.00-24.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 24.00-25.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 25.00-26.00  sec 56.0 KBytes 459 Kbits/sec 7
[ 4] 26.00-27.02  sec 48.0 KBytes 385 Kbits/sec 6
[ 4] 27.02-28.00  sec 48.0 KBytes 402 Kbits/sec 6
[ 4] 28.00-29.00  sec 48.0 KBytes 393 Kbits/sec 6
[ 4] 29.00-30.00  sec 48.0 KBytes 393 Kbits/sec 6
-----
[ ID] Interval      Transfer    Bandwidth  Jitter    Lost/Total Datagrams
[ 4] 0.00-30.00   sec 1.43 MBytes 400 Kbits/sec 19.572 ms 0/183 (0%)
[ 4] Sent 183 datagrams

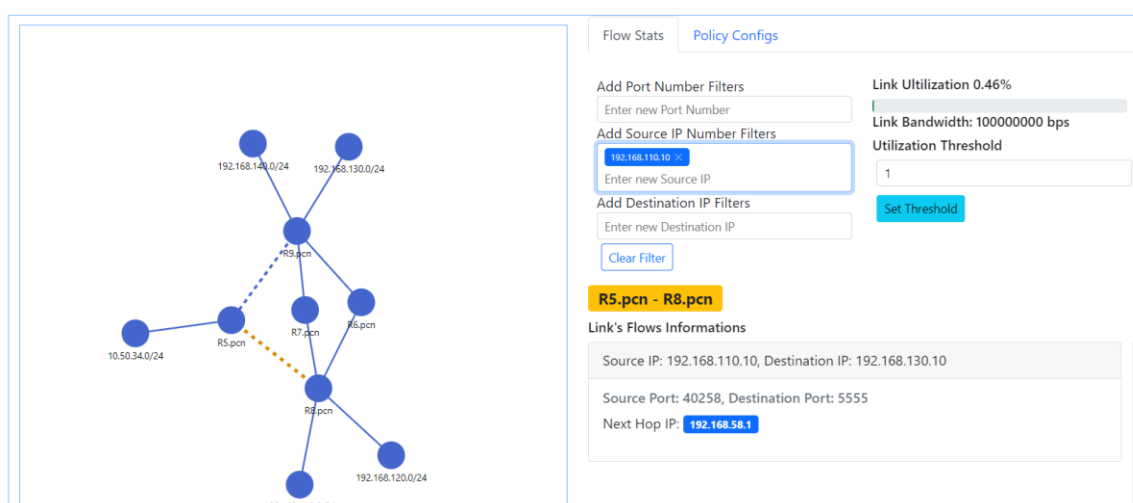
iperf Done.
root@Client-192:~#

```

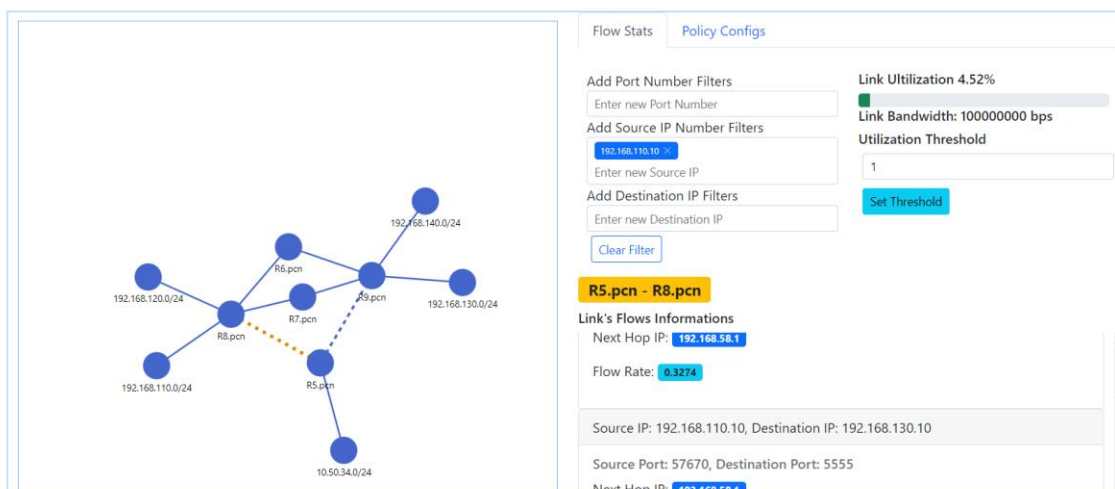
รูปที่ 4.6 ใช้ Iperf3 ในการสร้างแทรฟฟิก



รูปที่ 4.7 มีข้อมูลถูกส่งในลิงก์ประมาณ 400 Kbit/sec ตรวจสอบโดย Wireshark



รูปที่ 4.8 แสดงข้อมูลโฟลว์ที่มีไอพีต้นทาง 192.168.110.10

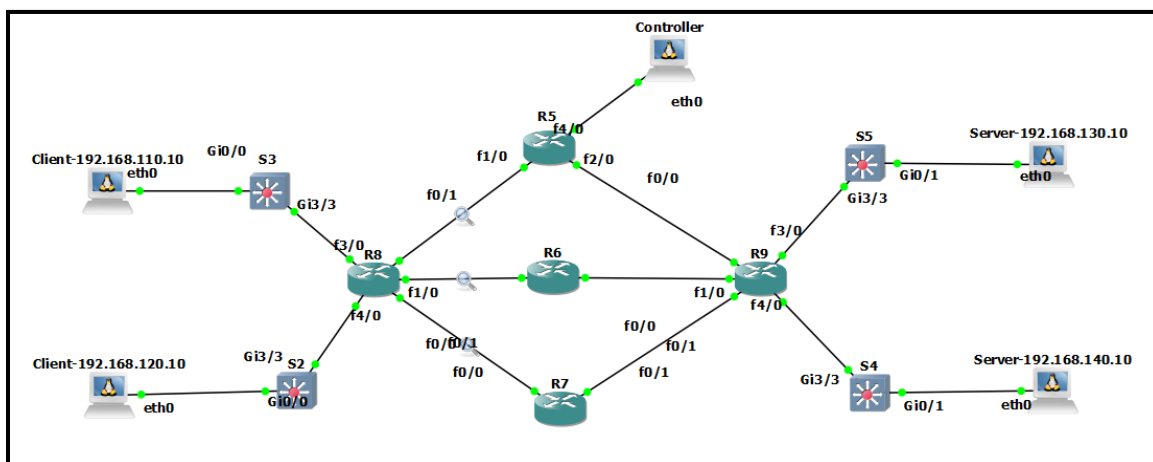


รูปที่ 4.9 โพล์ถูกเปลี่ยนเส้นทาง

4.2 แอปพลิเคชันสำหรับกระจายแพทर्फิก

4.2.1 First Topology

ต่ออุปกรณ์เครือข่ายดังภาพที่ 4.10 ลิงก์เส้นทาง R8-R5-R9 จะมีขนาดใหญ่ที่สุด ส่วน R8-R6-R9 และ R8-R7-R9 จะมีขนาดเท่ากัน เราเตอร์ทั้งหมดเชื่อมต่อโดยใช้ Routing Protocol แบบ OSPF เราเตอร์ทุกตัวสามารถเชื่อมต่อกับตัวควบคุม และติดต่อหากันได้ทุกเครือข่าย



รูปที่ 4.10 โทโปโลยีการทดลองที่ 1

ทดลองขนาดสร้างแตรฟฟิก 1 Mbit/sec จาก Client-192.168.110.10 ไปยัง Server-192.168.110.10 และสร้างแตรฟฟิกขนาด 1 Mbit/sec จาก Client-192.168.142.10 ไปยัง Server-192.168.140.10 ในสถานการณ์ปกติในลิงก์ R8-R5 และ R5-R9 จะมีโฟลว์ขนาดประมาณ 2Mbit/sec อยู่ตามภาพที่ 4.13 เนื่องจาก Routing Protocol OSPF มองว่าเส้นทาง R8-R5-R9 เป็นเส้นทางที่ดีที่สุด ซึ่งเราจะตั้งค่าให้ลิงก์ R8-R5 รับ Flow ได้ไม่เกิน 1Mbit ทำให้ลิงก์ดังกล่าวต้องเกิดการกระจายแตรฟฟิกขึ้น

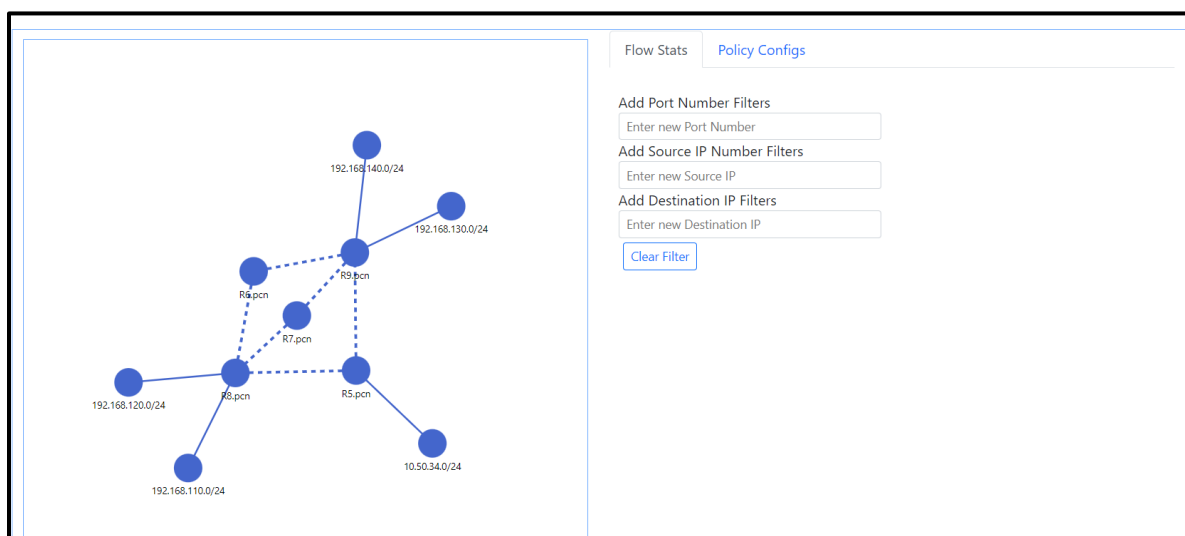
เนื่องจากลิงก์ R8-R5 ประกอบไปด้วยโฟลว์จาก Iperf 2 ตัว ตัวละ 1 Mbit/sec และโฟลว์จากการทำ SNMP อีกเล็กน้อยทำให้โฟลว์จาก Iperf ต้องถูกย้ายไปเส้นทางอื่น จากวิธีการเลือกเส้นทางของแอปพลิเคชันสำหรับกระจายแตรฟฟิกทำให้ระบบสร้าง นโยบายออกมา 2 นโยบายตามรูปที่ 4.11 ส่งผลให้เส้นทาง R8-R6-R9 และ R8-R7-R9 ถูกใช้งานขึ้นมา ตามรูปที่ 4.12

รูปที่ 4.13 และ 4.14 แสดงถึงโทโพโลยีที่ใช้ในการทดลอง เส้นสีเขียว และสีแดงแสดงถึงเส้นทางการเคลื่อนที่ของโฟลว์ โดยข้อมูลจาก Wireshark ระหว่างลิงก์ R8-R5 R8-R6 และ R8-R7 เป็นไปตามภาพที่ 4.15 4.16 และ 4.17 เมื่อทำการทดลองอีกครั้งโดยเอาแอปพลิเคชันสำหรับการกระจายแตรฟฟิกออกไปการใช้งานแบบวิดจ์จะเป็นไปตามรูปที่ 4.18 และ 4.19

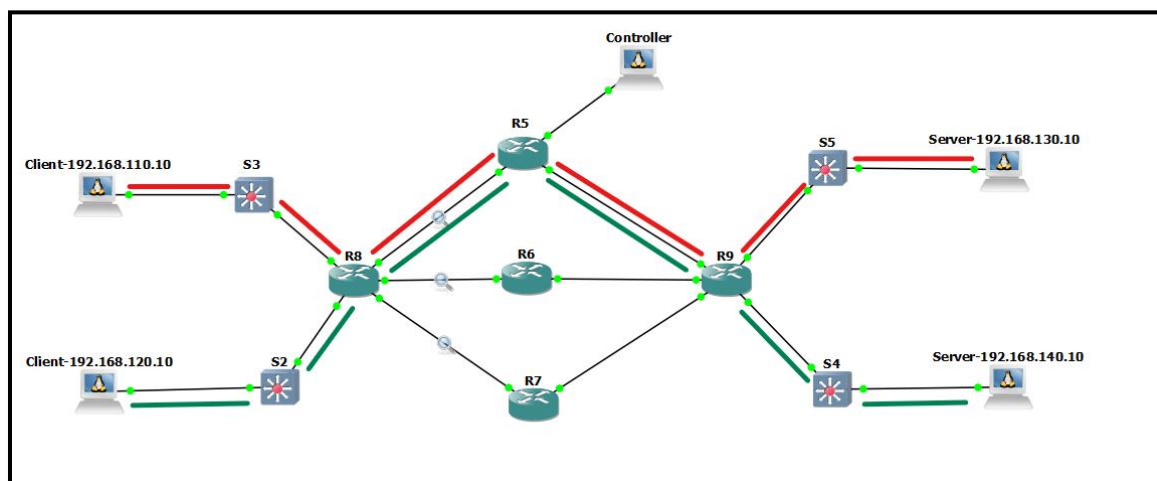
Flow routing

#	TYPE	STATUS	SOURCE	DESTINATION	ACTIONS									
0	static	active	192.168.110.10/24 [40274]	192.168.130.10/24 [5555]	<table><tr><th>DEVICE</th><th colspan="2">ACTION</th></tr><tr><td>R8.pcn</td><td>Next-hop IP</td><td>192.168.68.1</td></tr><tr><td>R6.pcn</td><td>Next-hop IP</td><td>192.168.69.2</td></tr></table>	DEVICE	ACTION		R8.pcn	Next-hop IP	192.168.68.1	R6.pcn	Next-hop IP	192.168.69.2
DEVICE	ACTION													
R8.pcn	Next-hop IP	192.168.68.1												
R6.pcn	Next-hop IP	192.168.69.2												
1	static	active	192.168.120.10/24 [51169]	192.168.140.10/24 [4444]	<table><tr><th>DEVICE</th><th colspan="2">ACTION</th></tr><tr><td>R8.pcn</td><td>Next-hop IP</td><td>192.168.78.1</td></tr><tr><td>R7.pcn</td><td>Next-hop IP</td><td>192.168.79.2</td></tr></table>	DEVICE	ACTION		R8.pcn	Next-hop IP	192.168.78.1	R7.pcn	Next-hop IP	192.168.79.2
DEVICE	ACTION													
R8.pcn	Next-hop IP	192.168.78.1												
R7.pcn	Next-hop IP	192.168.79.2												

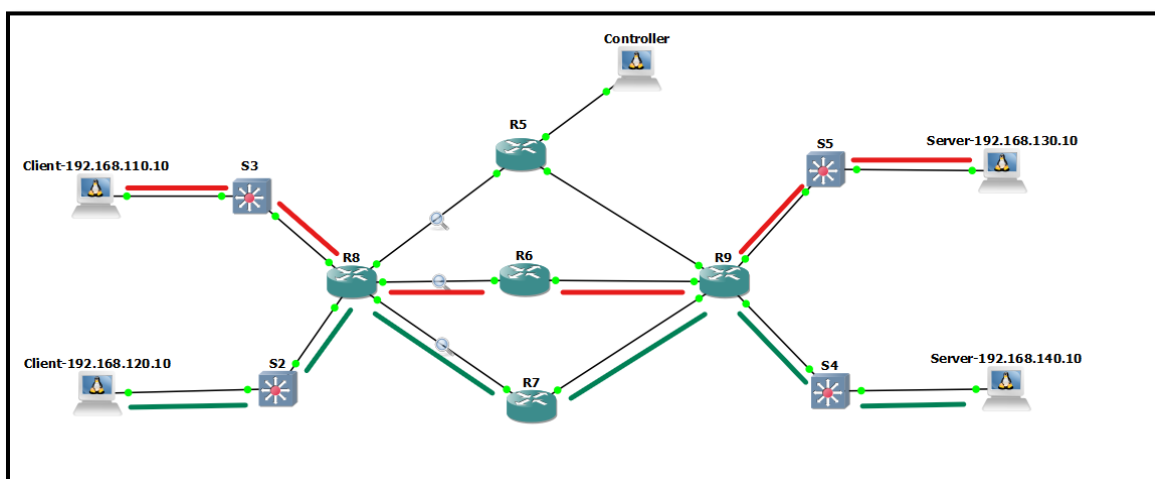
รูปที่ 4.11 นโยบายสำหรับเปลี่ยนเส้นทาง



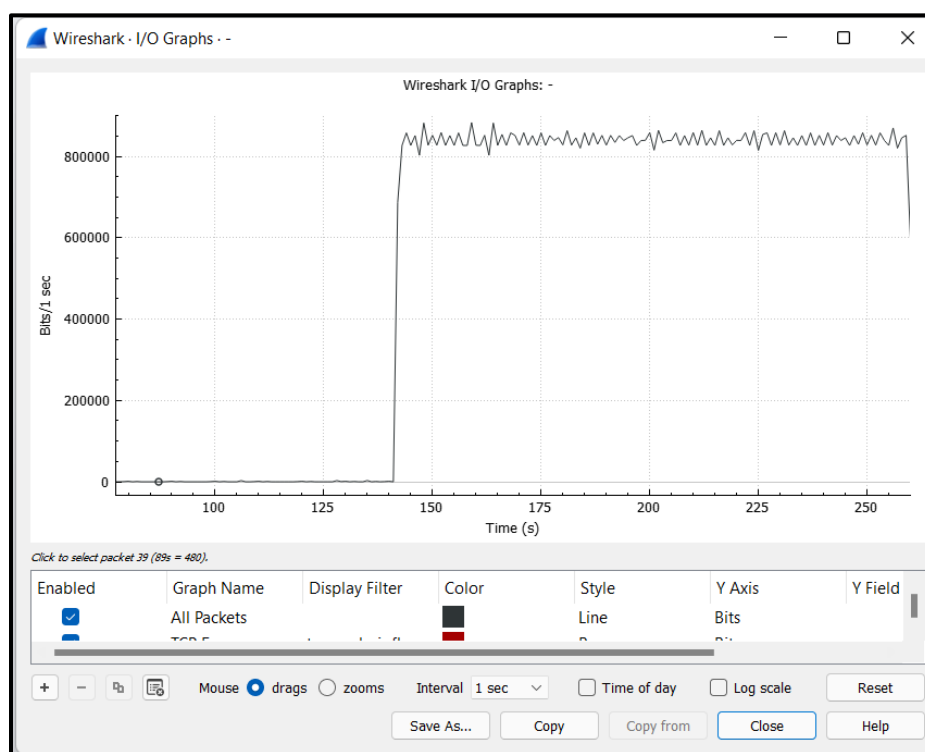
รูปที่ 4.12 มีเส้นทางใหม่ที่ถูกใช้งาน



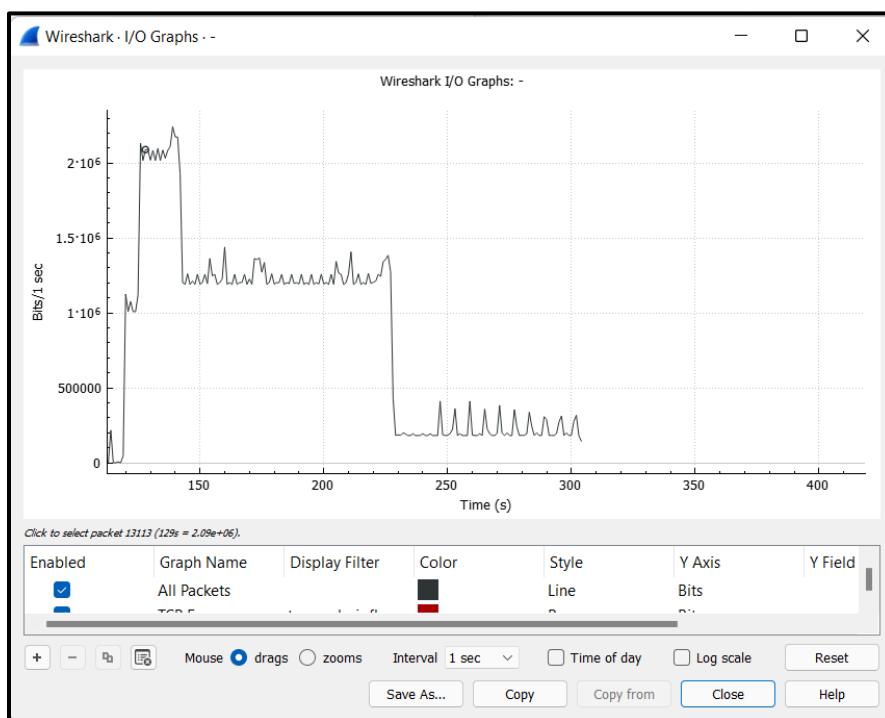
รูปที่ 4.13 เส้นทางก่อนมีการกระจายแพทไฟฟีก



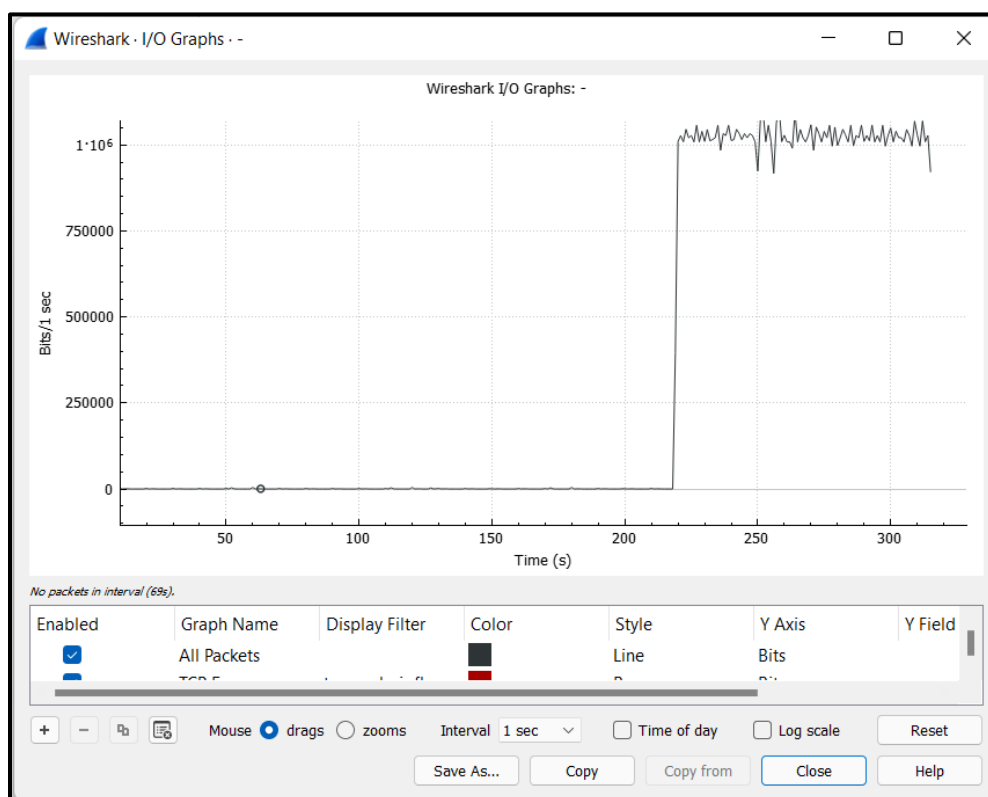
รูปที่ 4.14 เส้นทางหลังมีการกระจายแพคเกจ



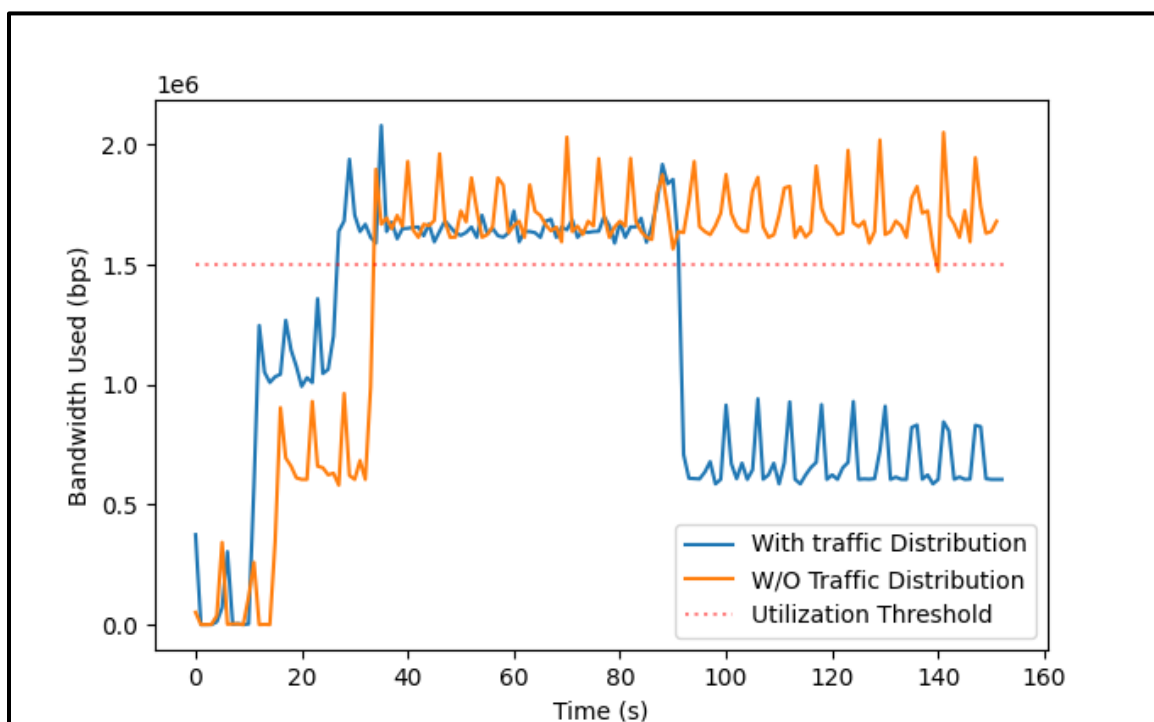
รูปที่ 4.15 ข้อมูลการใช้งานลิงก์ R8-R6-R9



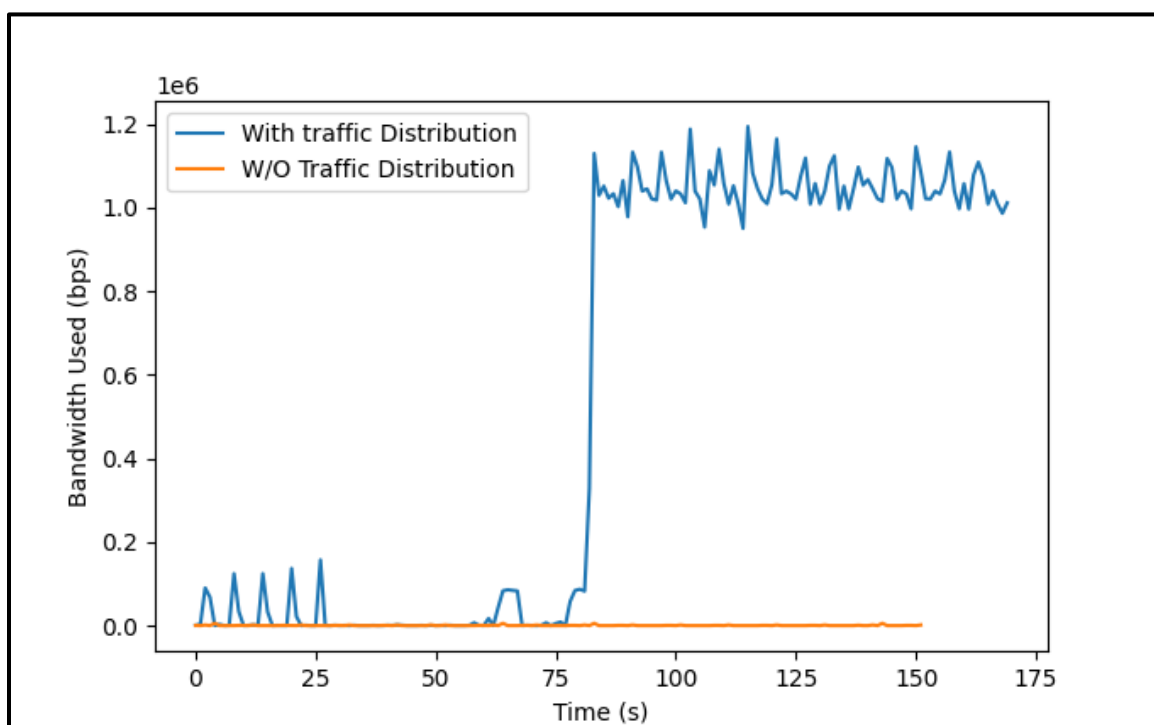
รูปที่ 4.16 ข้อมูลการใช้งานลิงก์ R8-R5-R9



รูปที่ 4.17 ข้อมูลการใช้งานลิงก์ R8-R7-R9



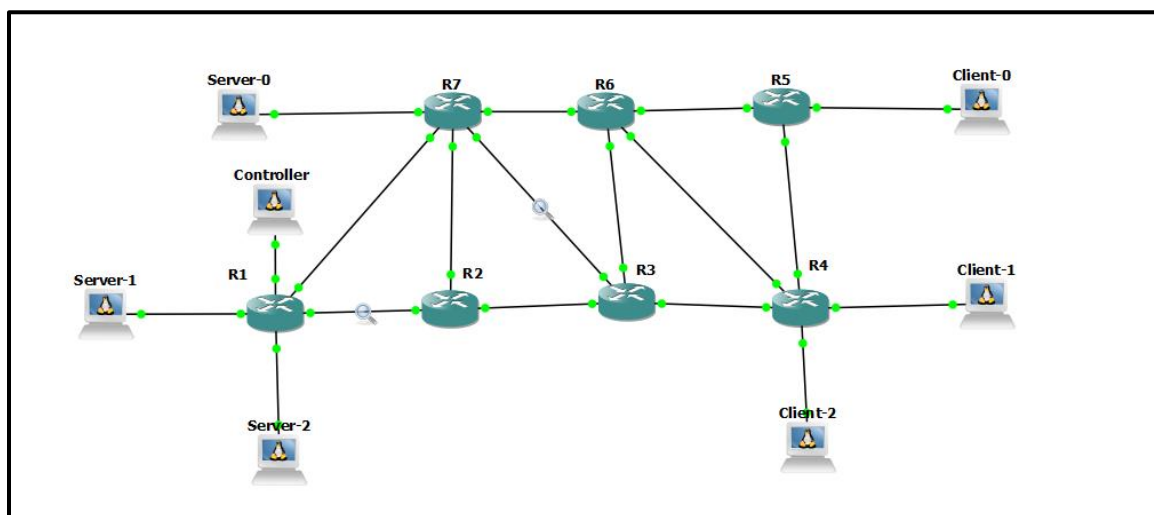
รูปที่ 4.18 ข้อมูลการใช้งานลิงก์ R8-R5 เปรียบเทียบระหว่างมีและไม่มีกระจายแพทर्फิก



รูปที่ 4.19 ข้อมูลการใช้งานลิงก์ R8-R7 เปรียบเทียบระหว่างมีและไม่มีกระจายแพทर्फิก

4.2.2 Second Topology

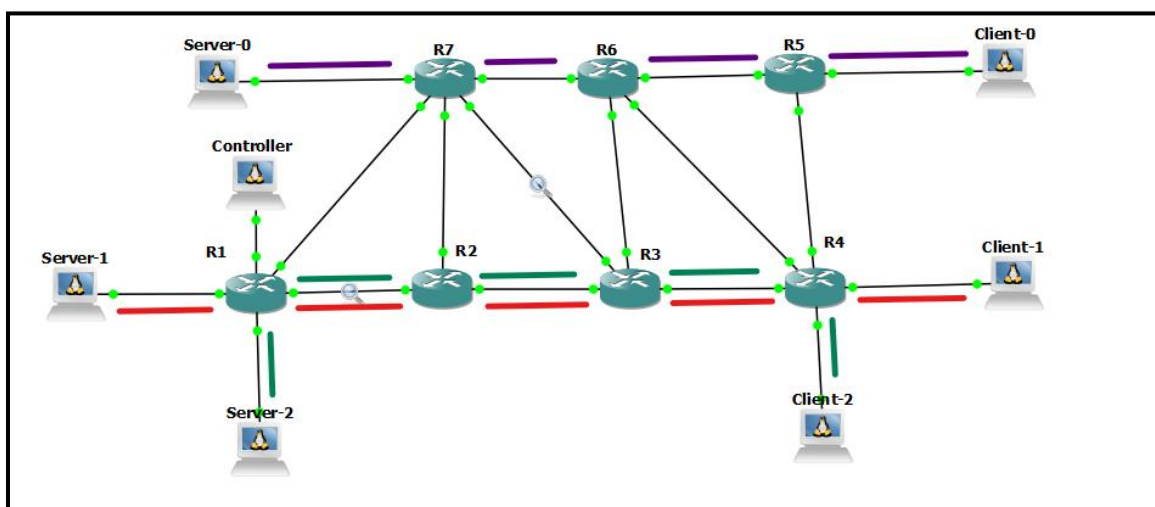
ต่ออุปกรณ์เครือข่ายดังภาพที่ 4.20 ประกอบไปด้วยเราเตอร์ทั้งหมด 7 ตัว อุปกรณ์สำหรับสร้างแทรฟฟิก 3 คู่ และตัวควบคุม เราเตอร์ทั้งหมดเชื่อมต่อโดยใช้ Routing Protocol แบบ OSPF เราเตอร์ทุกตัวสามารถเชื่อมต่อกับตัวควบคุม และติดต่อหากันได้ทุกเครือข่าย



รูปที่ 4.20 โทโพโลยีการทดลองที่ 2

ทดลองสร้างแทรฟฟิกขนาด 1 Mbit/sec จาก Client-0 Client-1 Client-2 ไปยัง Server-0 Server-1 Server-2 ตามลำดับ ในสถานการณ์ปกติการเคลื่อนที่ของโฟลว์ควรจะเป็นไปตามรูปที่ 4.21 แต่เมื่อมีการใช้งานถึงกับบริเวณ R1-R2-R3-R4 สูงกว่าค่า Threshold ที่กำหนดจะมีการสร้างนโยบายตามรูปที่ 4.22 สำหรับโฟลว์ที่สร้างขึ้นโดย Client-1 ที่เดินทางไปยัง Server-1 ให้เปลี่ยนเส้นทาง ผลลัพธ์การเคลื่อนที่ของโฟลว์ในระบบจึงเป็นไปตามภาพที่ 4.23

รูปที่ 4.21 และ 4.23 แสดงถึงโทโพโลยีที่ใช้ในการทดลอง เส้นสีเขียว สีแดง และสีม่วงแสดงถึงเส้นทางการเคลื่อนที่ของโฟลว์ หากใช้ Wireshark ตรวจสอบลิงก์ที่เคยถูกใช้งาน และลิงก์ที่ถูกย้ายโฟลว์เข้ามากราฟจะเป็นไปตามรูปที่ 4.24 และ 4.25

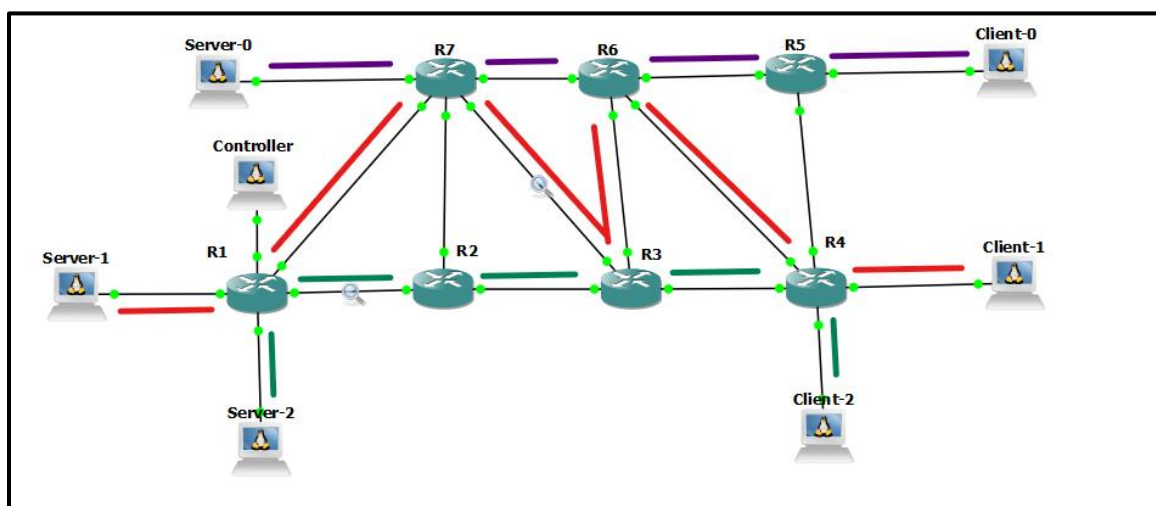


รูปที่ 4.21 เส้นทางก่อนมีการกระจายแพคเกจ

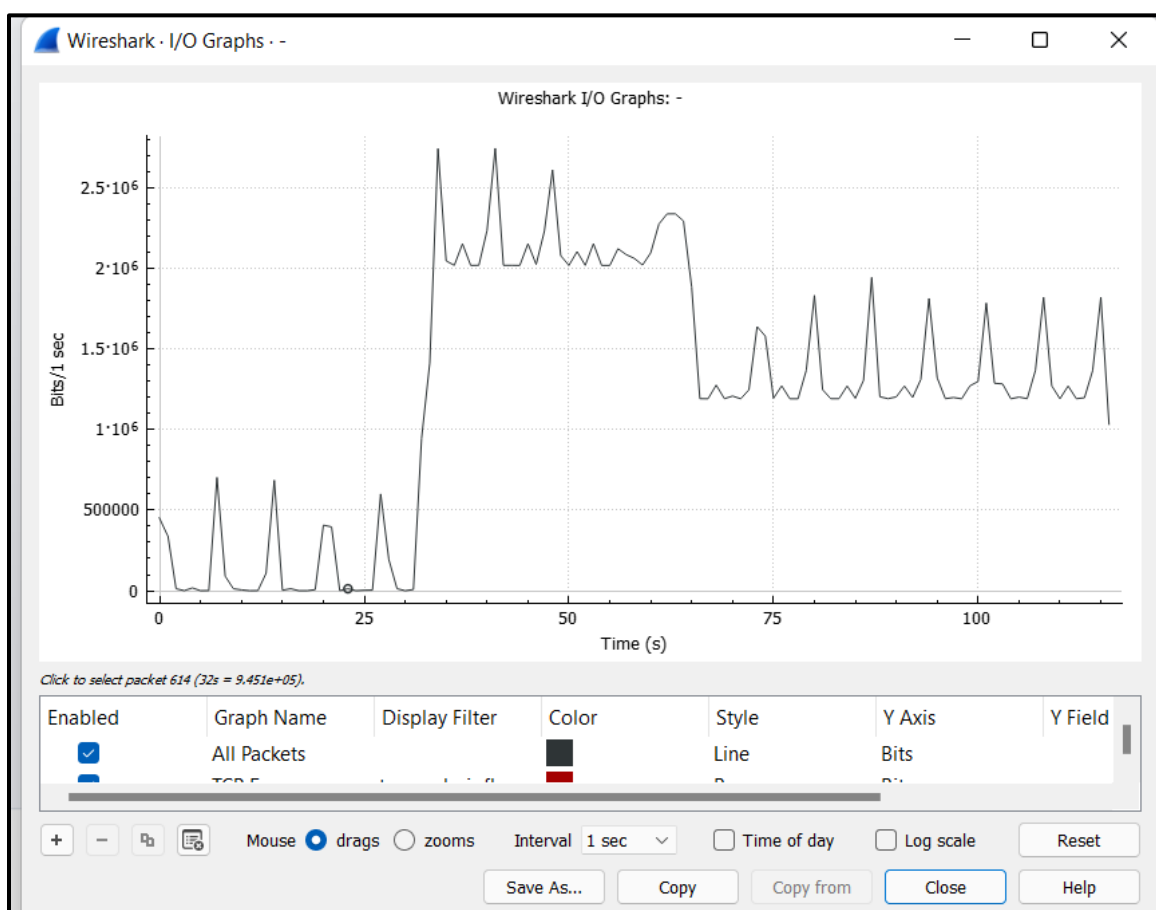
Flow routing

#	TYPE	STATUS	SOURCE	DESTINATION	ACTIONS		
0	static	active	192.168.210.10/24 [42808]	192.168.110.10/24 [5551]	DEVICE	ACTION	<div>Edit</div>
					R4.pcn	Next-hop IP 192.168.46.2	
					R6.pcn	Next-hop IP 192.168.36.1	
					R3.pcn	Next-hop IP 192.168.37.2	
					R7.pcn	Next-hop IP 192.168.17.1	

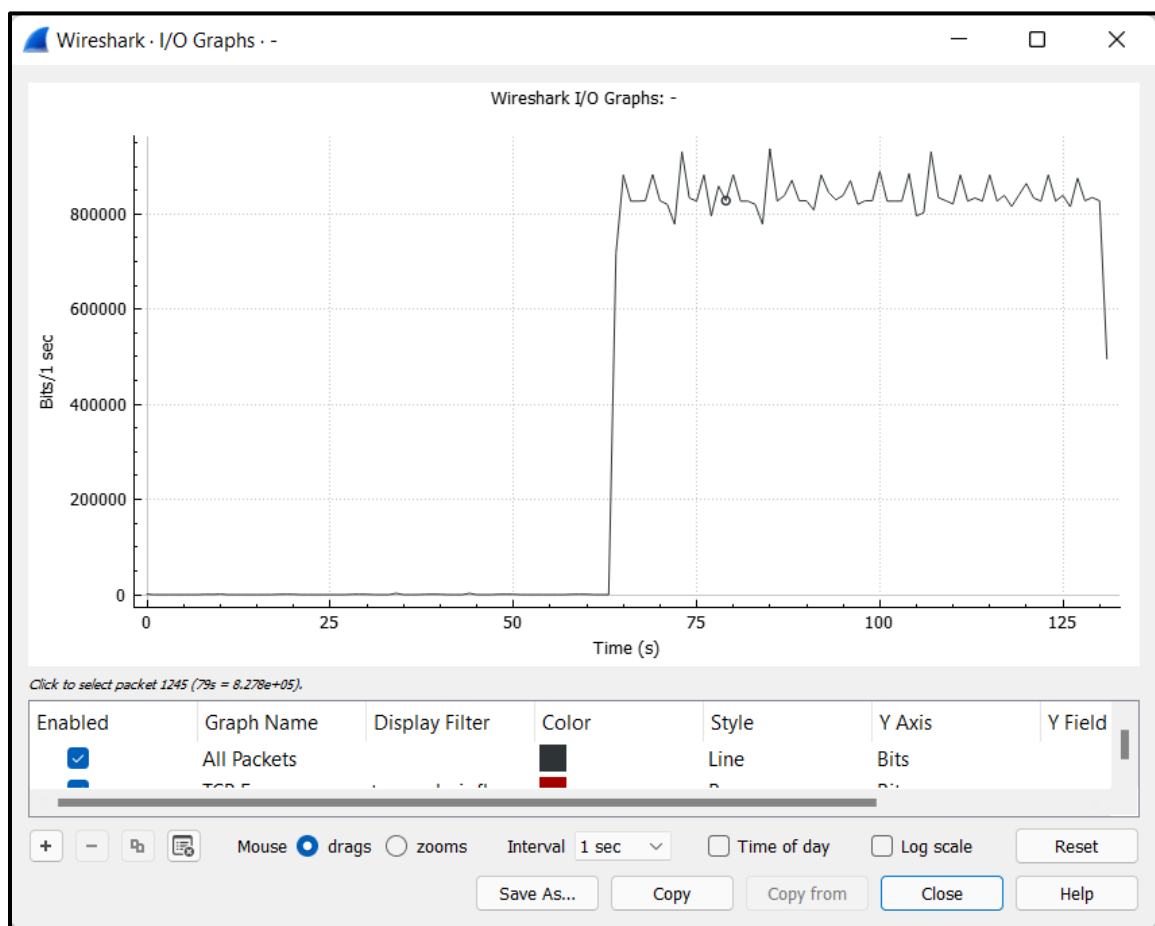
รูปที่ 4.22 นโยบายสำหรับเปลี่ยนเส้นทาง



รูปที่ 4. 23 เส้นทางหลังมีการกระจายแพคเกจ



รูปที่ 4.24 ข้อมูลการใช้งานลิงก์ที่ถูกใช้งานก่อนมีการกระจายแพคเกจ



รูปที่ 4.25 ข้อมูลการใช้งานลิงก์ที่ถูกใช้งานหลังมีการกระจายแพคเกจ

บทที่ 5

บทสรุป

5.1 สรุปผลโครงการ

แอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุมการกระจายแพทไฟกแบบที่ผู้ใช้งานกำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับมาตรฐานเอสดีเอ็น เป็นแอปพลิเคชันที่พัฒนาต่อออกมาจากระบบต้นแบบ โดยระบบที่พัฒนาขึ้นมาใหม่ประกอบไปด้วยหน้าเว็บแอปพลิเคชันสำหรับใช้งานตัวควบคุมพร้อมแสดงผลข้อมูลเครือข่าย และแอปพลิเคชันสำหรับการทำการกระจายแพทไฟกที่ทำงานตามเงื่อนไขเปอร์เซ็นต์การใช้งานแบนด์วิดท์ที่ผู้ใช้งานสามารถกำหนดได้

จากการทดลองในส่วนเว็บแอปพลิเคชันพบว่าระบบสามารถจัดการอุปกรณ์เครือข่าย และแสดงข้อมูลโทโปโลยีออกมาได้อย่างถูกต้อง และข้อมูลโฟลว์ในระบบก็สอดคล้องกับข้อมูลที่ได้จากโปรแกรม Wireshark ในส่วนของแอปพลิเคชันสำหรับการกระจายแพทไฟกสามารถสร้างนโยบายสำหรับปรับเปลี่ยนเส้นทางได้ตามเงื่อนไขที่วางแผนไว้เช่นเดียวกัน

5.2 ปัญหาในการทำโครงการและสรุปผล

1. ปัญหาโรคระบาดโควิด 19 ทำให้ไม่สามารถเข้าไปใช้งานทรัพยากรของทางคณะได้อย่างอิสระ
2. เทคโนโลยีที่ใช้พัฒนาในส่วนเว็บแอปพลิเคชัน ทางผู้จัดทำต้องใช้เวลาศึกษาเพิ่มเติมอย่างละเอียดทำให้ดำเนินการได้ล่าช้า
3. ระบบต้นแบบมีปัญหาที่ยังไม่ได้รับการแก้ไขในบางส่วน ทำให้ต้องแก้ไขปัญหาและปรับปรุงตัวควบคุมเพิ่มเติมก่อนนำมาพัฒนาต่อได้
4. ระบบต้นแบบมีการใช้ไลบรารีที่เก่า การพัฒนาระบบในบางส่วนจึงต้องสร้างใหม่ตั้งแต่ต้นทั้งหมดส่งผลให้ใช้เวลานานกว่าที่ควรจะเป็น

5.3 แนวทางการพัฒนาต่อ

1. ปรับปรุงประสิทธิภาพการทำงานจากการใช้ Netmiko ในการตั้งค่าอุปกรณ์
2. เพิ่มประสิทธิภาพให้เว็บแอปพลิเคชัน และตัวควบคุมรองรับการใช้งานจากผู้ใช้หลายคนในคราวเดียว

บรรณานุกรม

- [1] Ciena. **“Networking Insights What is SDN.”** [Online].Available:
www.ciena.com/insights/what-is/What-Is-SDN.html

- [2] Kamal Benzekki. **“Software-defined networking (SDN): A survey”** Security and
 Communication Networks, vol.1, no. 1, Febuary2017.pp5805-5805

- [3] Ian F.Akyildiz. **“A roadmap for traffic engineering in SDN-OpenFlow networks”**, vol.1, no. 1,
 June2014.pp1-30

- [4] Konstantin Avrachenkov. **“Differentiation Between Short and Long TCP
 Flows”**[Online].Available:<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.6517&rep=rep1>

- [5] Cisco. **“Manipulating Routing Updates Supplement”**[Online].Available:
https://ptgmedia.pearsoncmg.com/imprint_downloads/cisco/bookreg/2237xxd.pdf?fbclid=IwAR22pchWECvs2dGmci8D4nmXYm_EF5KqqyUeDCAAuf-KISHseaEBoocDzfU

- [6] Juniper. **“what-is-policy-based-routing”**[Online].Available: www.juniper.net/us/en/research-topics/what-is-policy-based-routing.html

- [7] Saixiii. **“SNMP คืออะไร โปรโตคอลสำหรับมอนิเตอร์อุปกรณ์ในระบบ”**[Online].Available:
www.saixiii.com/what-is-snmp/

- [8] Solarwinds, **“What is NetFlow?”** [Online].Available: www.solarwinds.com/netflow-traffic-analyzer/use-cases/what-is-netflow

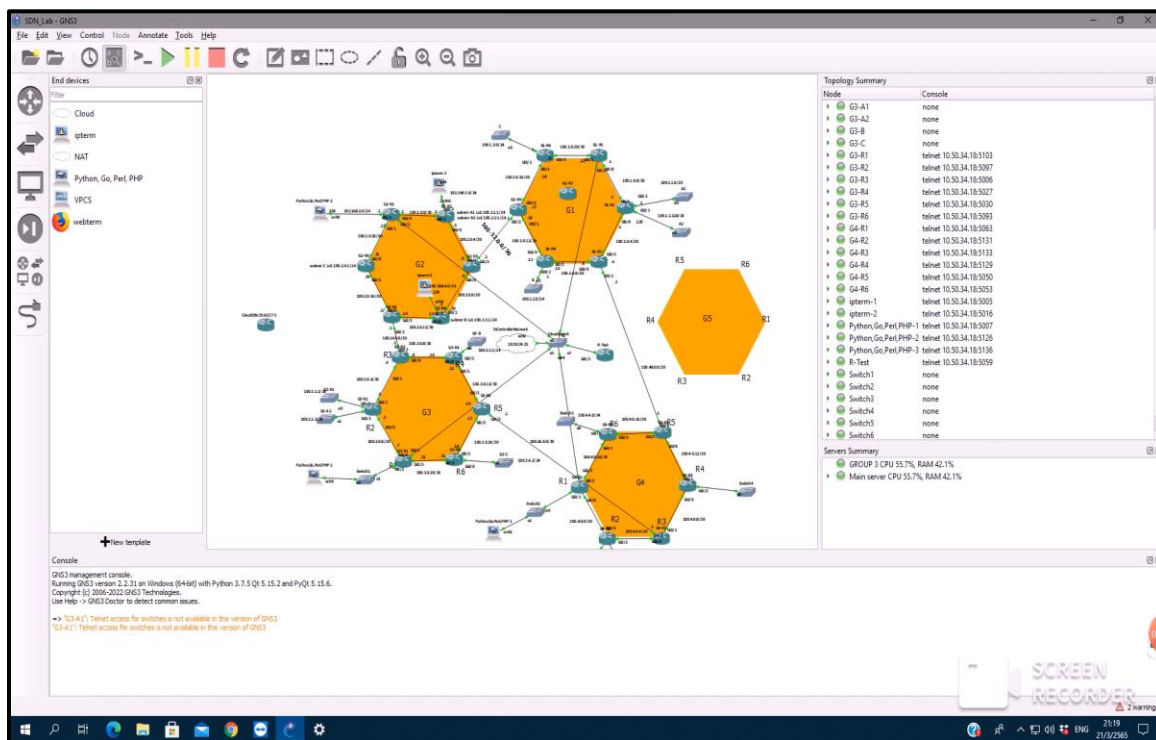
- [9] Cisco. **“Cisco Discovery Protocol (CDP)”** [Online]. Available:
www.learningnetwork.cisco.com/s/article/cisco-discovery-protocol-cdp-x

บรรณานุกรม (ต่อ)

- [10] TechTarget. “**What is SSH (Secure Shell) and How Does it Work?**” [Online].Available:
<https://www.techtarget.com/searchsecurity/definition/Secure-Shell>
- [11] Python. “**What is Python? Executive Summary**”[Online].Available:
www.python.org/doc/essays/blurb/
- [12] Chai Phonbopit “**MongoDB คืออะไร? + สอนวิธีใช้งานเบื้องต้น**”[Online].Available:
<https://devahoy.com/blog/2015/08/getting-started-with-mongodb/>
- [13] Packet Coders “**What is Netmiko?**”[Online].Available: <https://www.packetcoders.io/netmiko-the-what-and-the-why/>
- [14] ชยุตม์ สว่าง และอนุชิต มัชฌิมา. (2019). ระบบจัดการเครือข่ายเพื่อกระจายการจราจรบนเครือข่ายโดยใช้โครงสร้างตามสถาปัตยกรรมเอสดีเอ็น (ปริญญาโท) กรุงเทพฯ: สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

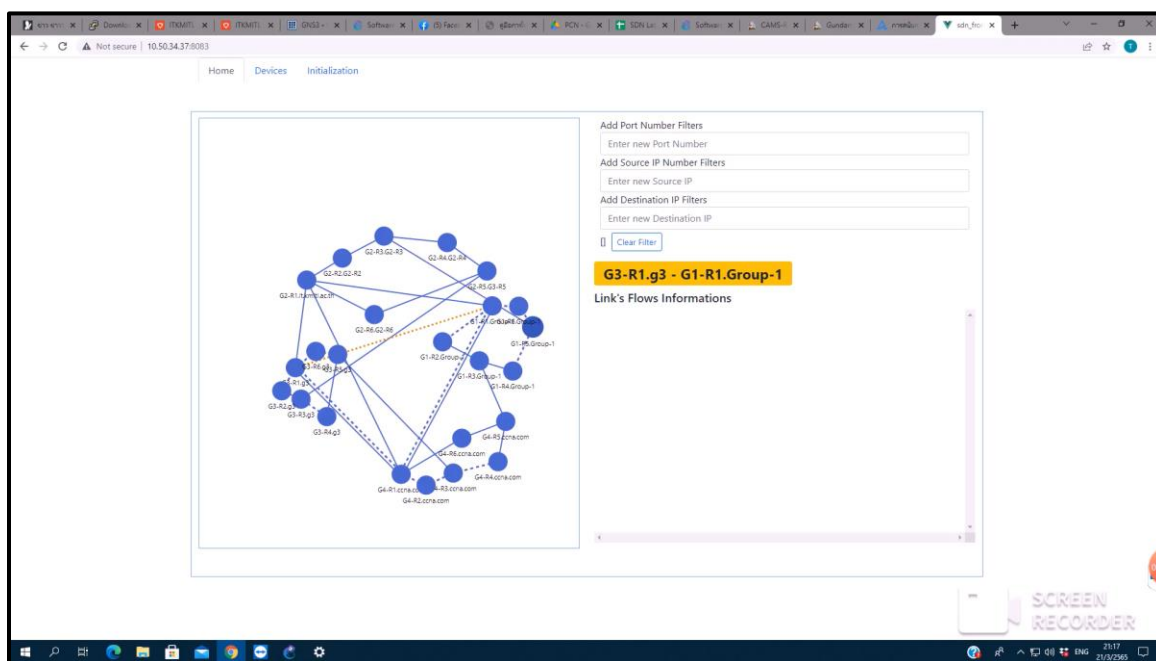
ภาคผนวก

การทดลองใช้งานระบบที่พัฒนาขึ้นใน วิชา PCN 06016331 ภาคเรียนที่ 2 ปีการศึกษา 2564 และผลจากแบบสอบถาม



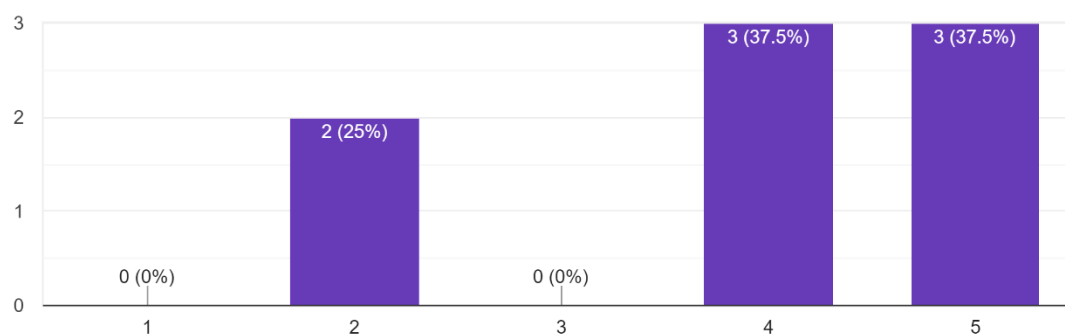
The screenshot shows the GNS3 web interface. The 'New Device Info' form is displayed on the right, and the 'Devices List' is on the left. The form includes fields for Management IP, System Type, SSH Username, SSH Password, SSH Secret, SSH Port, SNMP Version, SNMP Community String, and SNMP Port. The 'Devices List' shows a table of existing devices with their names and management IPs.

Device Name	Management IP
G2-R1.it.kmitl.ac.th	100.2.0.1
G2-R2.G2-R2	100.2.0.5
G2-R3.G2-R3	100.2.0.9
G2-R4.G2-R4	100.2.0.13
G2-R5.G3-R5	100.2.0.17
G2-R6.G2-R6	100.2.0.21
G3-R1.g3	100.3.0.1
G3-R2.g3	



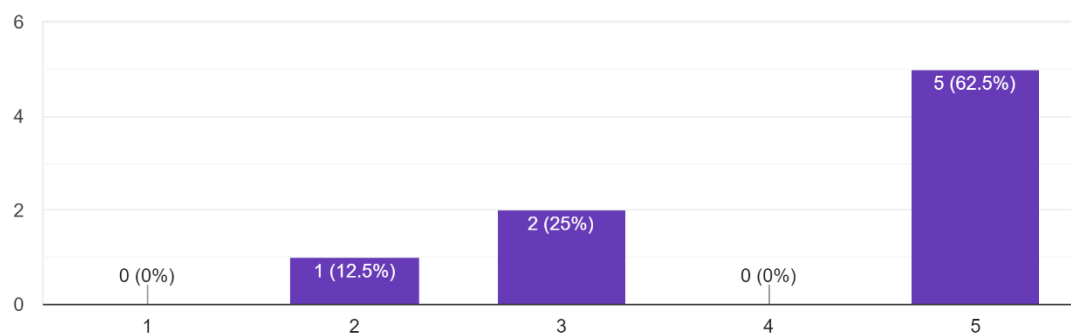
ตัวระบบทำให้ผู้ใช้ สามารถเข้าใจการทำงานของเครือข่ายตามสถาปัตยกรรมแบบเอสดีเอ็นได้

8 responses



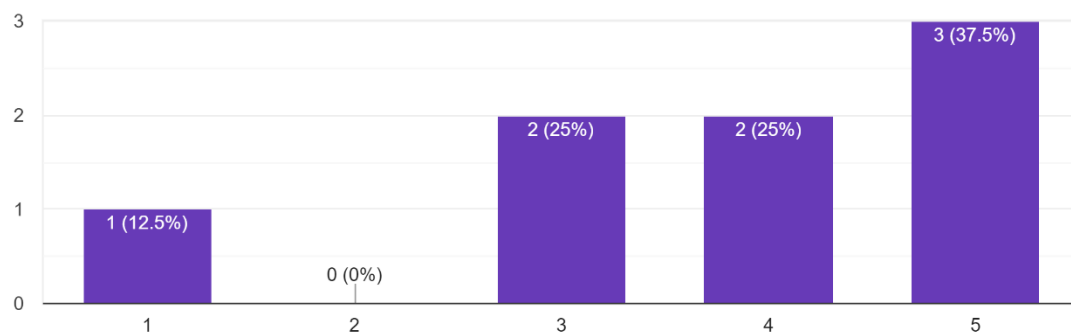
ระบบสามารถแสดงข้อมูลอุปกรณ์ และการเชื่อมต่อ Topology ได้ถูกต้องสมบูรณ์

8 responses



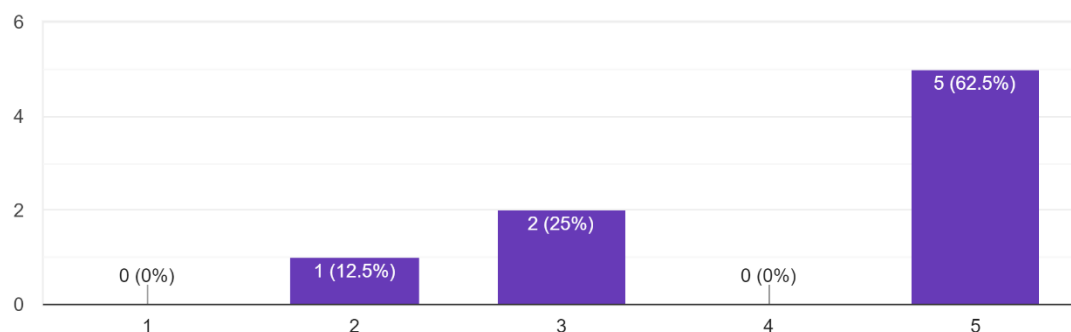
การใช้งานของหน้าจอ Web Interface ของระบบ

8 responses



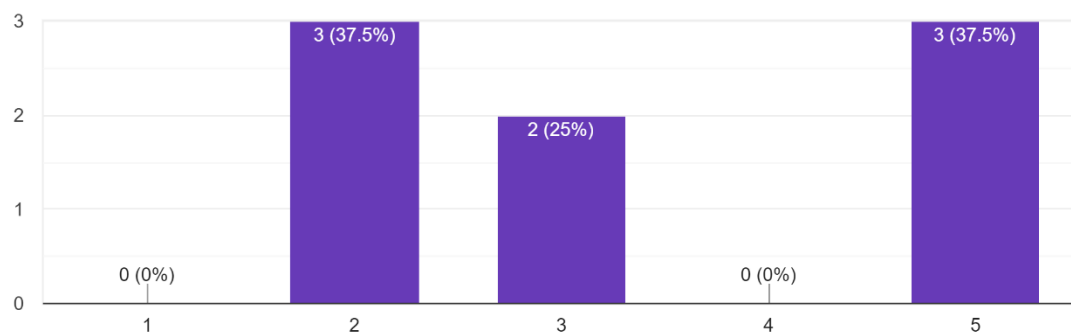
ระบบสามารถแสดงข้อมูลอุปกรณ์ และการเชื่อมต่อ Topology ได้ถูกต้องสมบูรณ์

8 responses



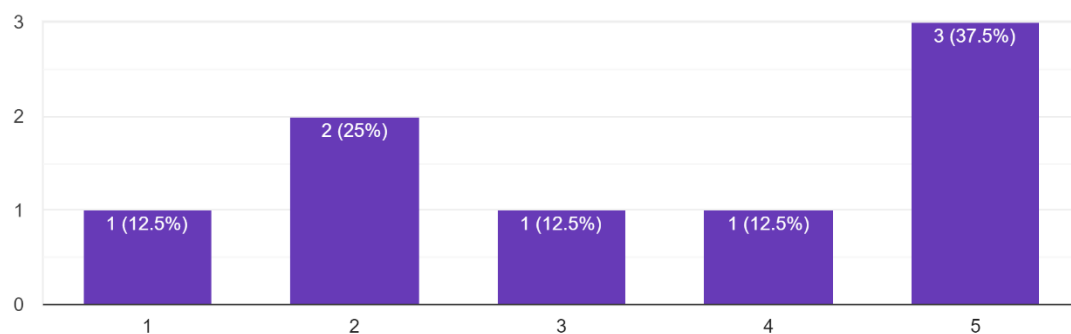
การเพิ่ม-ลบ-แก้ไข อุปกรณ์สามารถทำงานตอบสนองความต้องการของผู้ใช้ได้ถูกต้อง ครบถ้วน

8 responses



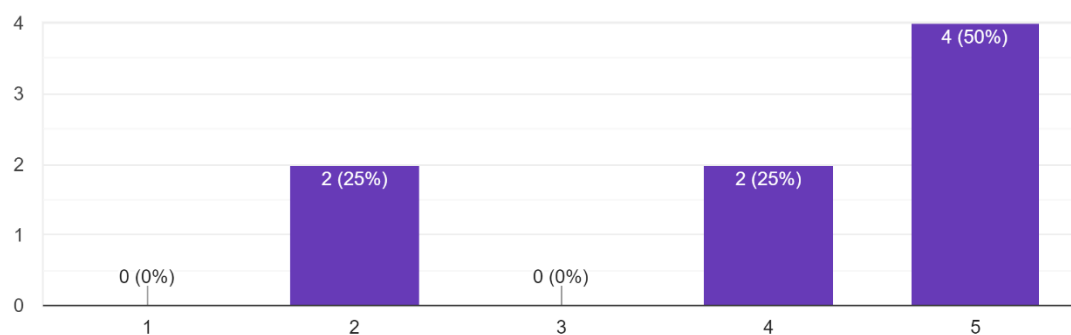
การเชื่อมต่ออุปกรณ์เครือข่ายกับตัวควบคุมสามารถทำได้ง่าย และรวดเร็ว

8 responses



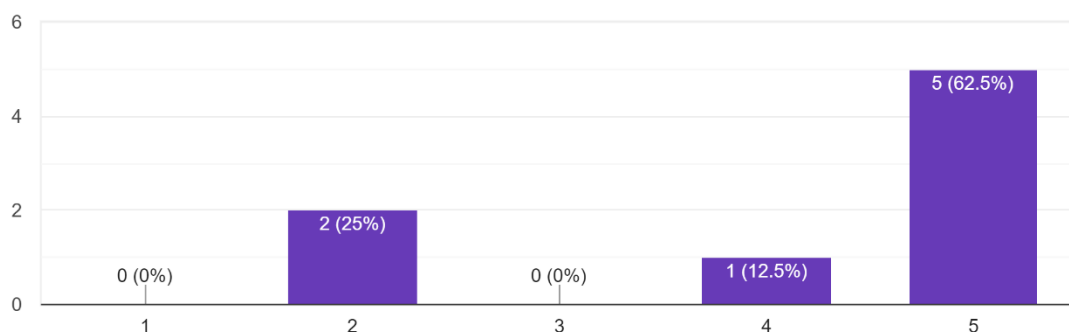
ระบบสามารถแสดงการเคลื่อนที่ และข้อมูลโฟลว์ในระบบเครือข่ายได้ถูกต้อง และรวดเร็ว

8 responses



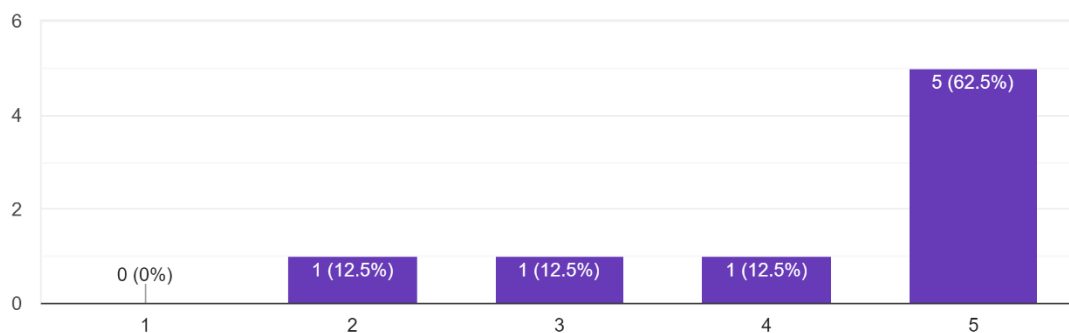
ระบบสามารถเปลี่ยนเส้นทางในการส่งข้อมูลได้อย่างรวดเร็ว และถูกต้องตามที่ผู้ใช้งานต้องการ

8 responses



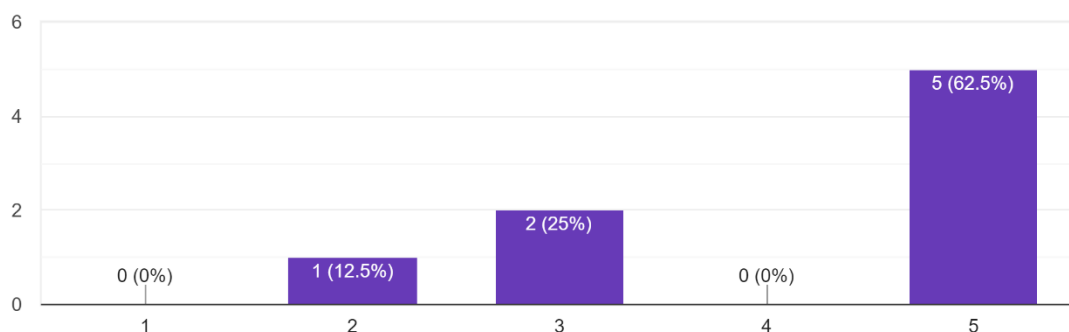
ความครอบคลุมของ API ที่ระบบมีให้เพียงพอต่อการใช้งาน

8 responses



API มีการกำหนดชื่อ และมีการใช้งานที่เข้าใจได้ง่ายไม่ซับซ้อน

8 responses



ข้อเสนอแนะสำหรับระบบ ปัญหาที่พบ ความล่าช้า (สิ่งที่ควรปรับปรุง/เพิ่มเติม) ในส่วนของหน้าจอของระบบ และ API เพื่อให้สามารถใช้ในการเรียนการสอน และการนำไปประยุกต์ใช้งานต่อไปในอนาคต

4 responses

backend อยากให้เริ่มรองรับการทำงานแบบ concurrency และเวลาที่มี process ไหนที่ต้องดำเนินการนานๆ อาจจะมี queue status หรือการ Tracking แทนการ request ค้างรอเพราะอาจเกิด Timeout และไม่สามารถทราบสถานะการทำงานได้

frontend หลักๆจะเป็นเรื่อง state การทำงานต่างๆที่ยังไม่ค่อยไหลลื่นเท่าไร

โดยรวม

โดยรวมโอเคครับมีระบบให้ใช้ ให้ทดสอบเป็น auto ซึ่งถ้าสมมุติว่านี่จะทำงานได้ดีมากๆในอนาคตครับ

เป็นกำลังใจให้ทีม Dev นะครับ ได้แบบนี้ถือว่าสุดยอดมากครับ แต่อาจจะต้องใช้เวลาลึกหน่อย

...

การ enable secret กับ username password ที่ไม่เหมือนกันทำให้เพิ่มอุปกรณ์ลงเว็บไม่ได้ครับ

ตอน add device มันมีบั๊กเล็กน้อยตอนเลือกแต่ละรายการ

ตอนลบ device บางครั้งลบแล้วมันไม่หายไป

อยากให้คำอธิบายแต่ละส่วนจะได้ไม่ต้องถามทีไร

ประวัติผู้เขียน

ชื่อ - นามสกุล นายพงพนิช อรัญรัตน์โสภณ

รหัสนักศึกษา 61070124

วัน เดือน ปี เกิด 10 มีนาคม 2543

ประวัติการศึกษา

วุฒิม.6 ชื่อที่อยู่สถาบัน โรงเรียนเบญจมราชูทิศ

ภูมิลำเนา 148/5 หมู่ 8 ตำบล ห้วยไทร อำเภอ ห้วยไทร จังหวัด นครศรีธรรมราช

เบอร์โทร 093-6588282 Email 61070124@kmitl.ac.th

สาขาที่จบ เทคโนโลยีสารสนเทศ รุ่นที่ 16 ปีการศึกษาที่จบ 2564



ชื่อ - นามสกุล นายภูริณัฐ จิตมนัส

รหัสนักศึกษา 61070171

วัน เดือน ปี เกิด 29 กรกฎาคม 2542

ประวัติการศึกษา

วุฒิม.6 ชื่อที่อยู่สถาบัน โรงเรียนเบญจมราชูทิศ

ภูมิลำเนา 44/12 หมู่ 4 ตำบล ท่าจีว อำเภ เมือง จังหวัด นครศรีธรรมราช

เบอร์โทร 089-4728789 Email 61070171@kmitl.ac.th

สาขาที่จบ เทคโนโลยีสารสนเทศ รุ่นที่ 16 ปีการศึกษาที่จบ 2564

