

การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อ  
ควบคุมการกระจายแพทเทรฟฟิกแบบที่ผู้ใช้กำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่  
รองรับมาตรฐานเอสดีเอ็น

**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  
สาขาเทคโนโลยีสารสนเทศ  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ภาคเรียนที่ 1 ปี การศึกษา 2564

การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุม  
การกระจายแพทเทรฟฟิกแบบที่ผู้ใช้กำหนดได้ 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  
สาขาเทคโนโลยีสารสนเทศ  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ภาคเรียนที่ 1 ปี การศึกษา 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 1/2021**

**COPYRIGHT 2021**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABAN**

ใบรับรองปริญญาโท ประจำปีการศึกษา 2564  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุม  
การกระจายทราฟฟิกแบบที่ผู้ใช้กำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับ  
มาตรฐานเอสดีเอ็น

**Implementation of SDN Application and Controller for User-defined Traffic  
Distribution in Traditional (Non-SDN) Networks.**

ผู้จัดทำ

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

.....อาจารย์ที่ปรึกษา  
(.....)

## ใบรับรองโครงการ(Project)

### เรื่อง

การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุมการกระจาย  
แพทเทิร์นที่ผู้ใช้กำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับมาตรฐานเอสดีเอ็น  
Implementation of SDN Application and Controller for User-defined Traffic Distribution  
in Traditional (Non-SDN) Networks.

นายพงศ์พนิช อรัญรัตน์โสภณ      รหัสนักศึกษา 61070124

นายภูริณัฐ จิตมนัส                      รหัสนักศึกษา 61070171

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด  
รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของการศึกษาวิชา  
โครงการ หลักสูตรวิทยาศาสตรบัณฑิต (เทคโนโลยีสารสนเทศ)  
ภาคเรียนที่ 1 ปีการศึกษา 2564

.....  
(นายพงศ์พนิช อรัญรัตน์โสภณ)

.....  
(นายภูริณัฐ จิตมนัส)

หัวข้อโครงการ	การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุมการกระจายแพทเทรฟฟิกแบบที่ผู้ใช้กำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับมาตรฐานเอสดีเอ็น	
นักศึกษา	นายพงศ์พนิช อรัญรัตน์โสภณ	รหัสนักศึกษา 61070124
	นายภูริณัฐ จิตมนัส	รหัสนักศึกษา 61070171
ปริญญา	วิทยาศาสตร์บัณฑิต	
สาขาวิชา	เทคโนโลยีสารสนเทศ	
ปีการศึกษา	2564	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุมเมธ 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 PRABHAVAT	

## **ABSTRACT**

Legacy network devices are not compatible with modern Software-Defined Network Architecture (SDN) software that is used to manage, control, and monitor network systems. In this project, the team will develop a controller within the SDN architecture that is able to manage and control legacy network devices. In addition, develop a network traffic distribution app that is coordinated with the controller.



## กิตติกรรมประกาศ

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

พงศ์พนิช อรัญรัตน์โสภณ

ภูริณัฐ จิตมนัส

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VI
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีการดำเนินงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
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 การกระจายการรับส่งข้อมูลเครือข่ายและการเลือกเส้นทาง.....	4
2.2.1 Policy-Based Routing (PBR).....	5
2.3 การสำรวจเก็บข้อมูลเครือข่าย.....	5
2.3.1 Simple Network Management Protocol (SNMP).....	5
2.3.2 NetFlow.....	5

## สารบัญ (ต่อ)

2.3.3 Cisco Discovery Protocol (CDP) .....	5
2.4 เทคโนโลยีที่ใช้ในการพัฒนา .....	6
2.4.1 Python .....	6
2.4.2 MongoDB.....	6
แนวคิดและการดำเนินงาน .....	7
3.1 ขั้นตอนการดำเนินงาน.....	7
3.2 วิเคราะห์ระบบต้นแบบ .....	8
3.2.1 โครงสร้างระบบต้นแบบ .....	8
3.2.2 ติดตั้ง Environment และ Topology สำหรับการทดลอง.....	9
3.2.3 ขั้นตอนการติดตั้งใช้งานระบบต้นแบบ .....	11
3.2.4 ขั้นตอนการวางแผนปรับปรุงระบบต้นแบบ.....	11
ผลการดำเนินการ .....	12
4.1 แผนภาพแสดงองค์ประกอบของระบบใหม่ที่ออกแบบ .....	12
4.2 แก้ไขกระบวนการตั้งค่าอุปกรณ์เครือข่ายเบื้องต้น (Initialize).....	13
4.3 แก้ไขกระบวนการเพิ่มอุปกรณ์เข้าสู่ระบบ .....	15
4.4 ปรับปรุงรูปแบบการเก็บข้อมูล เพื่อให้เหมาะสมกับการนำมาใช้พัฒนาต่อ .....	17
4.5 ปรับปรุงกระบวนการตั้งค่า Policy Based Routing ผ่าน API ให้มีความสอดคล้องกับการทำงานจริงมากขึ้น .....	19
4.6 พัฒนา API สำหรับเรียกข้อมูลเพื่อใช้แสดงผลในหน้าแอปพลิเคชันที่จะพัฒนาในอนาคต.....	21
บทสรุป .....	22
5.1 สรุปผลโครงการ.....	22
5.2 ปัญหาในการทำโครงการและสรุปผล.....	22
5.3 แนวทางการพัฒนาต่อ.....	23
บรรณานุกรม.....	24

## สารบัญรูป

หน้า

### รูปที่

รูปที่ 3.1 โครงสร้างระบบต้นแบบ.....	8
รูปที่ 3.2 Topology ที่ใช้ในการทดลอง.....	9
รูปที่ 3.3 อุปกรณ์ที่ทำหน้าที่เป็นตัวควบคุม.....	9
รูปที่ 4.4 VM ที่ทำหน้าที่เป็น GNS3 Server .....	10
รูปที่ 3.5 หน้าเว็บ VMware ESXI ที่ติดตั้งตัวควบคุมและ GNS3 Server.....	10
รูปที่ 4.1 องค์ประกอบของระบบใหม่ .....	12
รูปที่ 4.2 ผังขั้นตอนการตั้งค่าอุปกรณ์เครือข่ายแบบเดิม .....	13
รูปที่ 4.3 ผังขั้นตอนการตั้งค่าอุปกรณ์เครือข่ายแบบใหม่ .....	14
รูปที่ 4.4 ผังขั้นตอนการเพิ่มอุปกรณ์เครือข่ายเข้าสู่ระบบแบบเดิม .....	15
รูปที่ 4.5 ผังขั้นตอนการเพิ่มอุปกรณ์เครือข่ายเข้าสู่ระบบแบบใหม่.....	16
รูปที่ 4.6 เพิ่มการเก็บค่า Serial Number.....	17
รูปที่ 4.7 เพิ่มการเก็บค่า Mbit per second .....	17
รูปที่ 4.8 เพิ่มการเก็บค่า Policy Aging .....	18
รูปที่ 4.9 เพิ่มการเก็บค่า running_flows เพื่อดู flow ที่วิ่งอยู่ในแต่ละ link.....	18
รูปที่ 4.10 สามารถตั้งค่า Policy โดยใช้ Source/Destination IP ตามต้องการ.....	19
รูปที่ 4.11 เมื่อเวลาผ่านไปตามที่กำหนดจะลบ Policy ที่หมดเวลาออก.....	20
รูปที่ 4.12 API สำหรับสร้างกราฟในเว็บแอปพลิเคชัน .....	21

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาของปัญหา

ในปัจจุบันระบบเครือข่ายมีการเติบโต และมีการใช้งานเพิ่มขึ้นเป็นจำนวนมากทำให้ข้อมูลต่างๆ ในระบบเครือข่ายมีจำนวนเพิ่มมากขึ้นเป็นทวีคูณ ส่งผลให้เกิดปัญหาความคับคั่งของการจราจรเครือข่าย (เน็ตเวิร์กแทรฟฟิก) ในบางเส้นทางได้ การจัดการแทรฟฟิกจึงมีความสำคัญที่ทำให้ระบบเครือข่ายสามารถใช้งานได้มีประสิทธิภาพ สถาปัตยกรรมแบบเอสดีเอ็นถูกคิดค้นขึ้นมาเพื่อช่วยบริหารจัดการอุปกรณ์เครือข่ายที่มีความซับซ้อน อย่างไรก็ตามสถาปัตยกรรมดังกล่าวสามารถใช้ได้กับอุปกรณ์เครือข่ายที่รองรับมาตรฐานเอสดีเอ็นเท่านั้น

แอปพลิเคชันและคอนโทรลเลอร์ตามสถาปัตยกรรมแบบเอสดีเอ็นระบบนี้ เป็นระบบที่พัฒนาขึ้นเพื่อที่จะช่วยให้อุปกรณ์เครือข่ายแบบดั้งเดิมสามารถบริหารจัดการเก็บข้อมูลต่างๆ ของระบบเครือข่ายแสดงผลให้ผู้ใช้งานเข้าใจง่าย และสามารถตั้งค่าจัดการแทรฟฟิกแบบที่ผู้ใช้งานต้องการได้

#### 1.2 วัตถุประสงค์ของโครงการ

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

### 1.3 ขอบเขตของโครงการ

พัฒนาระบบจัดการเครือข่ายสำหรับอุปกรณ์ที่ไม่รองรับมาตรฐานเอสดีเอ็นเท่าที่สามารถหาได้จากการที่คณะจัดสรรให้ เพื่อให้สามารถควบคุมอุปกรณ์เหล่านั้นให้ทำงานตามนโยบายที่กำหนดผ่านทางส่วนติดต่อผู้ใช้งานที่เป็นเว็บแอปพลิเคชันได้

### 1.4 วิธีการดำเนินงาน

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

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ระบบจัดการเครือข่ายตามสถาปัตยกรรมแบบเอสดีเอ็นที่ช่วยให้ผู้ใช้สามารถมอนิเตอร์และการกระจายแพทไฟกบนเครือข่ายได้ตามที่กำหนด แม้ว่าอุปกรณ์ในเครือข่านั้นจะไม่รองรับมาตรฐานเอสดีเอ็นก็ตาม

## บทที่ 2

# ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

ผู้จัดทำได้มีการศึกษาทฤษฎีที่เกี่ยวข้องกับงานวิจัย และเทคโนโลยีที่ใช้ในการจัดทำโดยมีรายละเอียดดังนี้

## 2.1 Software Defined Network (SDN)

Software Defined Network หรือเอสดีเอ็นเป็นเทคโนโลยีที่เกี่ยวข้องกับการบริหารจัดการเครือข่ายโดยปรับปรุงการบริหารจัดการเครือข่ายแบบเก่าให้มีการรวมศูนย์ ลดความซับซ้อนของระบบทำให้ง่ายแก่การจัดการ ผู้ดูแลระบบเครือข่ายสามารถบริหารจัดการระบบเครือข่ายผ่านคอนโทรลเลอร์โดยไม่ต้องเข้าไปจัดการกับอุปกรณ์เครือข่ายอื่นๆ โดยตรง ระบบเครือข่ายแบบดั้งเดิมการควบคุมแพคเกจจะขึ้นอยู่กับตารางเส้นทางที่ถูกตั้งค่าไว้ตามอุปกรณ์เครือข่ายในแต่ละเครื่อง ในสถาปัตยกรรมแบบเอสดีเอ็นจะแยกการควบคุม และการส่งข้อมูลออกจากกันทำให้สามารถรู้สถานะของระบบเครือข่ายจากคอนโทรลเลอร์และสามารถควบคุมระบบเครือข่ายทั้งหมดผ่านทางคอนโทรลเลอร์เพียงอย่างเดียว โดยสถาปัตยกรรมแบบเอสดีเอ็นได้แบ่งลำดับชั้นการทำงานเป็น 3 ลำดับชั้น โดยสื่อสารผ่าน 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 และติดต่อกับอุปกรณ์เครือข่ายโดยใช้ Southbound API [2]

Northbound API อินเทอร์เน็ตเฟสเหนือ ช่วยให้ชั้นควบคุมสามารถติดต่อกับลำดับชั้นบนหรือส่วนของแอปพลิเคชันได้

Southbound API อินเทอร์เน็ตเฟสใต้ ช่วยให้ชั้นควบคุมสามารถติดต่อกับอุปกรณ์เครือข่ายอื่นๆ ในระดับล่าง ในเอสดีเอ็น คือ โปรโตคอล OpenFlow

### 2.1.3 Infrastructure Layer

ชั้นโครงสร้างประกอบไปด้วยอุปกรณ์เครือข่ายต่างๆ ซึ่งเป็นพื้นฐานของระบบใช้ในการรับ ส่งข้อมูล โดยอุปกรณ์เหล่านี้ในสถาปัตยกรรมเอสดีเอ็นจะมีหน้าที่ส่งข้อมูลไปให้ชั้นควบคุมและทำหน้าที่ตามที่ส่วนควบคุมกำหนด

## 2.2 การกระจายการรับส่งข้อมูลเครือข่ายและการเลือกเส้นทาง

ในเครือข่ายประกอบไปด้วยข้อมูลที่ถูกส่งไปมาระหว่างอุปกรณ์ต้นทางและ อุปกรณ์ปลายทางผ่านเส้นทางต่างๆ อย่างไรก็ตามเส้นทางแต่ละเส้นมีทรัพยากรที่ใช้ในการส่งข้อมูลอย่างจำกัด การที่มีข้อมูลผ่านเส้นทางเส้นใดเส้นหนึ่งมากเกินไปส่งผลให้เกิดความคับคั่งของเครือข่ายตามมาด้วยปัญหาการส่งข้อมูลที่ช้า หรือทำข้อมูลสูญหายได้ การกระจายการรับส่งข้อมูลเครือข่ายจึงเป็นแนวคิดที่ช่วยในการเลือกเส้นทางการรับส่งข้อมูลไม่ให้หนักเส้นทางใดเส้นทางหนึ่งมากเกินไป [3]



### 2.2.1 Policy-Based Routing (PBR)

เป็นวิธีการส่งข้อมูลโดยอาศัยข้อมูลจากนโยบายหรือเกณฑ์ที่กำหนดไว้ ซึ่งสามารถกำหนดได้ผ่านตัวแปรต่างๆ เช่น ไอพีต้นทาง ไอพีปลายทาง พอร์ตต้นทาง พอร์ตปลายทาง หรือเกณฑ์อื่นๆ แล้วแต่ผู้ใช้กำหนด การส่งข้อมูลแบบพีบีอาร์ช่วยให้การส่งข้อมูลเป็นไปได้อย่างรวดเร็วและยืดหยุ่น จึงสามารถนำมาประยุกต์ใช้กับการกระจายแพทเทิร์น เพื่อให้ใช้เส้นทางในการรับส่งข้อมูลได้อย่างมีประสิทธิภาพ [4]

## 2.3 การสำรวจเก็บข้อมูลเครือข่าย

### 2.3.1 Simple Network Management Protocol (SNMP)

เอสเอ็นเอ็มพีจัดอยู่ในลำดับชั้นแอปพลิเคชันของ โอเอสไอโมเดล เป็นโปรโตคอลสำหรับตรวจสอบและบริหารจัดการอุปกรณ์เครือข่ายประเภท Local Area Network (LAN) หรือ Wide Area Network (WAN) เอสเอ็นเอ็มพีจะจัดเก็บข้อมูลและจัดการโดย Management Information Base หรือ MIB ซึ่งเป็นฐานข้อมูลสำหรับจัดการอุปกรณ์ โดยการจัดเก็บจะประกอบไปด้วย Object ID (OID) โดยเป็นชื่อเฉพาะที่เป็นเอกลักษณ์ของอุปกรณ์แต่ละตัว และถูกจัดเรียงในรูปแบบของแผนภาพต้นไม้ [5]

### 2.3.2 NetFlow

NetFlow เป็นเทคโนโลยีที่อยู่ในอุปกรณ์เครือข่ายรวมถึง Cisco IOS เป็นเครื่องมือที่ใช้ในการตรวจสอบการรับส่งข้อมูลและเก็บสถิติข้อมูลในเครือข่ายเหล่านั้น ผู้ดูแลระบบสามารถใช้ข้อมูลเหล่านั้นในการวิเคราะห์ นำไปสู่การพัฒนาเครือข่ายให้มีประสิทธิภาพมากยิ่งขึ้น [6]

### 2.3.3 Cisco Discovery Protocol (CDP)

CDP เป็นโปรโตคอลของ Cisco เป็นเครื่องมือที่ช่วยให้อุปกรณ์เครือข่ายสามารถแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์เครือข่าย cisco ที่อยู่ติดกัน ทำให้สามารถเก็บข้อมูลสถานะของของอุปกรณ์เครือข่ายได้ [7]

## 2.4 เทคโนโลยีที่ใช้ในการพัฒนา

### 2.4.1 Python

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

ไพทอนสามารถทำงานได้บนหลายระบบปฏิบัติการ เช่น Linux, Unix, Windows เป็นต้น อีกทั้งยังเป็นโอเพนซอร์สที่เข้าถึงได้ง่าย ทำให้มีไลบรารีให้ใช้อยู่เป็นจำนวนมากเพิ่มความสะดวกสบายในการทำงานที่หลากหลาย [8]

### 2.4.2 MongoDB

มอนโกดีบี เป็นฐานข้อมูลแบบ open-source document ประเภทหนึ่งโดยเป็นฐานข้อมูลแบบ NoSQL มีจุดเด่นที่ทำงานได้ไว เหมาะกับฐานข้อมูลที่มีขนาดใหญ่ และไม่ซับซ้อน การเก็บข้อมูลจะเก็บคีย์ และข้อมูลเอาไว้โดยต้องมีคีย์หลักที่เป็นเอกลักษณ์เป็นหน่วยพื้นฐานของข้อมูล เนื่องจากการจัดเก็บข้อมูลในรูปแบบ BSON (Binary JSON) ซึ่งมีรูปแบบคล้าย JSON (JavaScript Object Notation) แต่เก็บข้อมูลได้หลากหลายกว่า แต่การเชื่อมตารางไปยังฐานข้อมูลอื่นๆ ทำไปได้ยาก [9]

## บทที่ 3

### แนวคิดและการดำเนินงาน

#### 3.1 ขั้นตอนการดำเนินงาน

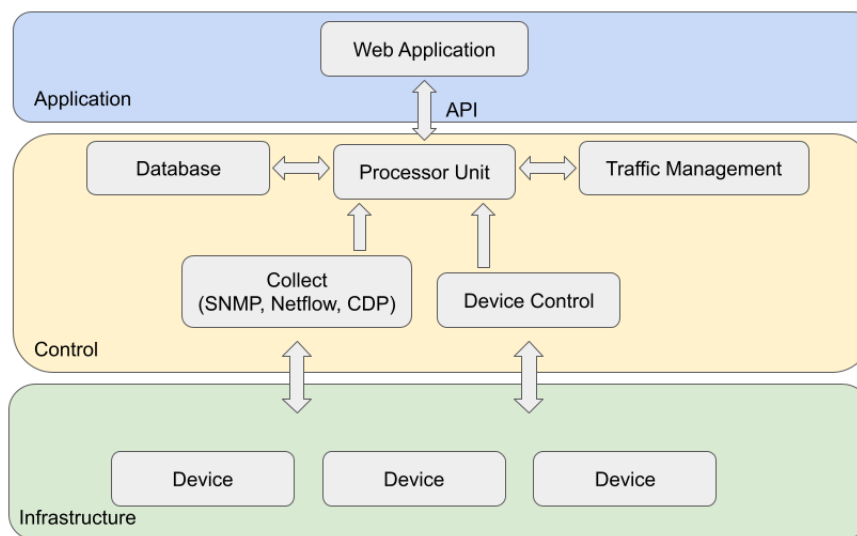
1. ศึกษาโครงสร้างและการจัดการระบบเครือข่ายตามสถาปัตยกรรมแบบเอสดีเอ็น
2. ติดตั้งเซิร์ฟเวอร์และจัดหาอุปกรณ์สำหรับการทดลอง
3. ศึกษา ติดตั้งแอปพลิเคชันสำหรับการจัดการอุปกรณ์เครือข่ายที่ไม่รองรับการทำงานตามสถาปัตยกรรมเอสดีเอ็นผ่านเอสดีเอ็นคอนโทรลเลอร์ที่ถูกพัฒนาขึ้น
4. ปรับปรุงการทำงานของระบบต้นแบบเพื่อเตรียมพร้อมในการนำมาใช้พัฒนาใหม่
5. ศึกษาแนวคิดวิธีการสำรวจจัดเก็บข้อมูลเครือข่าย
6. ศึกษาแนวทางและพัฒนากลไกการกระจายแพทช์เฟิร์มแวร์เครือข่ายตามที่ใช้กำหนดผ่านทางส่วนติดต่อกับผู้ใช้งาน
7. ทดสอบการใช้งานและสรุปผล

### 3.2 วิเคราะห์ระบบต้นแบบ

#### 3.2.1 โครงสร้างระบบต้นแบบ

โครงสร้างระบบต้นแบบ แบ่งเป็น 3 ลำดับชั้น

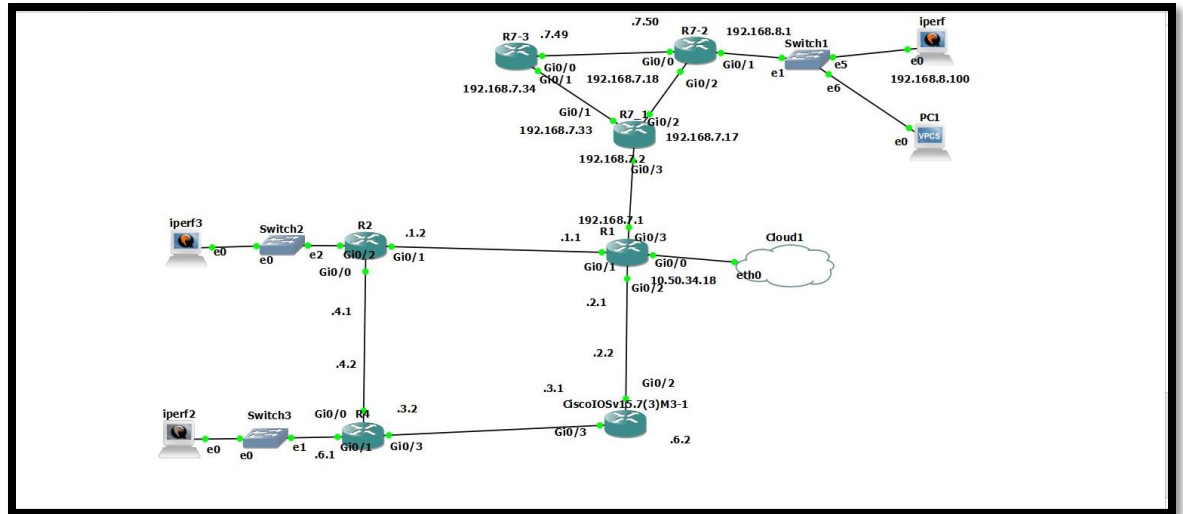
1. Application Layer: ชั้นแอปพลิเคชัน หรือฝั่งหน้าบ้านใช้ในการติดต่อกับฝั่งผู้ใช้ซึ่งถูกพัฒนาด้วย Web Pack Nuxt JS
2. Control Layer: ฝั่งหลังบ้านของระบบ แบ่งเป็น 3 ส่วน คือ
  - 2.1. ส่วนของตัวควบคุมการทำงานระบบเครือข่าย ใช้ Python Netmiko ในการ initialize และการ ทำ policy based routing / py mongo เพื่อติดต่อด้านข้อมูล /การสร้าง socket เพื่อใช้รอรับข้อมูลการเปลี่ยนแปลงของสถานะเครือข่าย]
  - 2.2. ส่วนของการเก็บข้อมูลใช้ mongo DB
  - 2.3. ส่วนของ API ที่ใช้ติดต่อกับ frontend กับ ผู้ใช้งานใช้ Sanic Cors เป็น REST Server
3. Infrastructure Layer: เป็น ส่วนของระบบเครือข่ายที่มีอุปกรณ์ที่ไม่รองรับสถาปัตยกรรมแบบเอสดีเอ็นตามที่คณะจัดทำมาได้ ในส่วนของเซิร์ฟเวอร์ที่ใช้ centos7 เพื่อรันระบบ หน้าบ้าน และหลังบ้าน (ตัวควบคุม)



รูปที่ 3.1 โครงสร้างระบบต้นแบบ

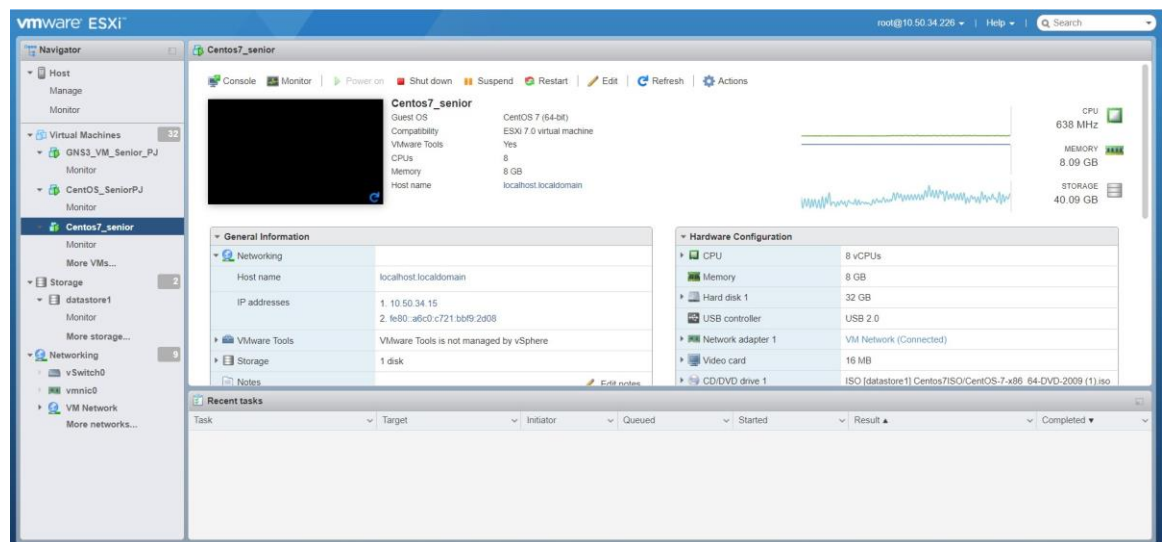
### 3.2.2 ติดตั้ง Environment และ Topology สำหรับการทดลอง

1. Topology ที่ใช้ในการทดลองประกอบไปด้วย Routers และ Hosts เพื่อใช้สำหรับการใช้งานระบบตลอดการพัฒนา โดยเครือข่ายดังกล่าวต่อกับตัวควบคุมที่ตั้งอยู่ใน Cloud เครือข่ายของคณะ



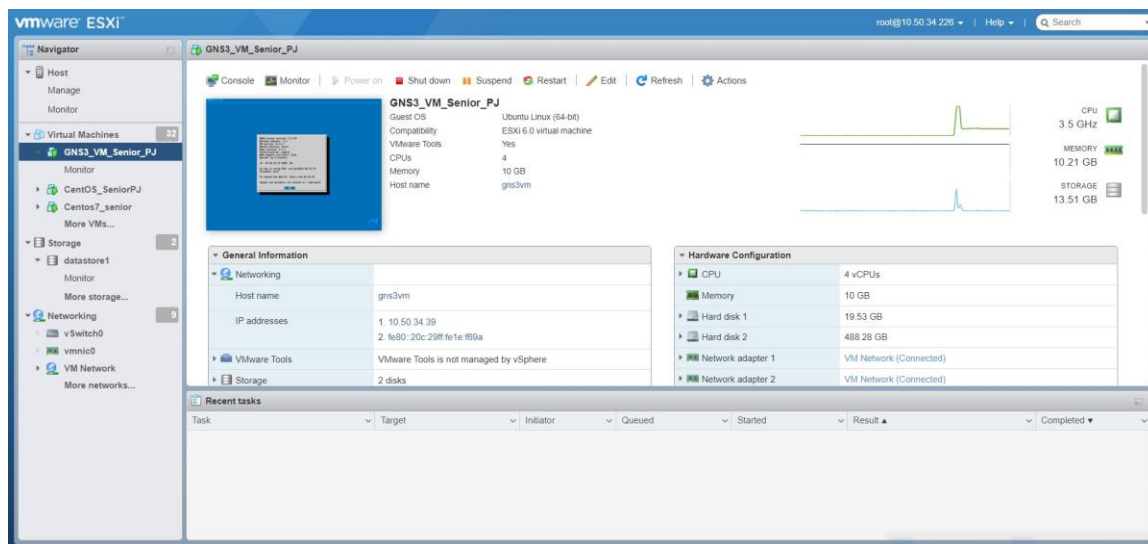
รูปที่ 3.2 Topology ที่ใช้ในการทดลอง

2. เซิร์ฟเวอร์สำหรับทำหน้าที่เป็นตัวคอนโทรลเลอร์เป็น Virtual Machine ระบบปฏิบัติการ CentOS7



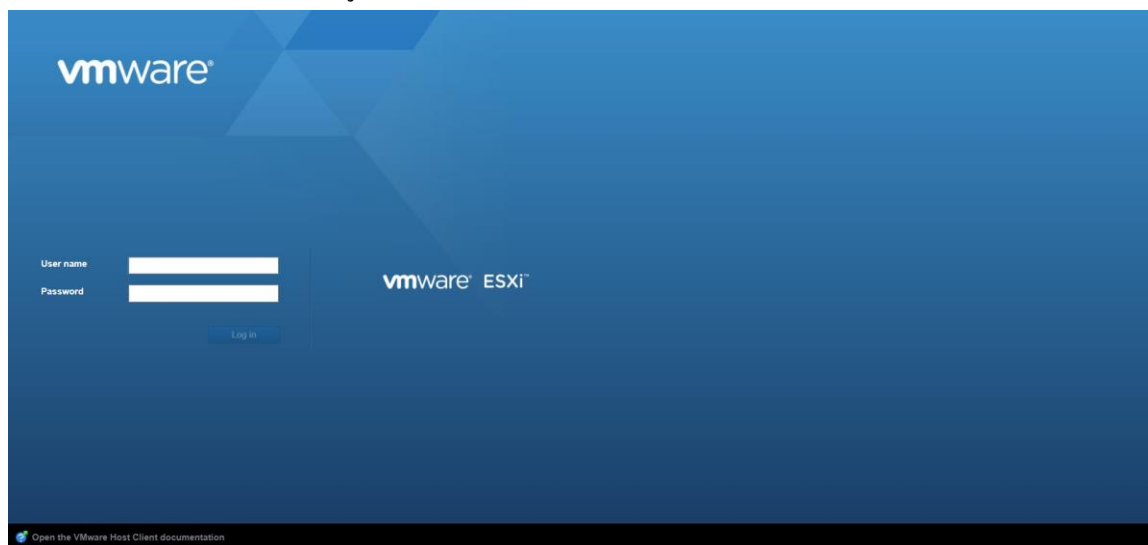
รูปที่ 3.3 VM ที่ทำหน้าที่เป็นตัวควบคุม

### 3. เซิร์ฟเวอร์สำหรับจำลองตัวอุปกรณ์เครือข่ายเป็น Virtual Machine GNS3 VM ระบบปฏิบัติการ Ubuntu Linux



รูปที่ 3.4 VM ที่ทำหน้าที่เป็น GNS3 Server

### 4. Environment ที่ใช้ติดตั้งอยู่บน VMware ESXI เซิร์ฟเวอร์ของคุณะ



รูปที่ 3.5 หน้าเว็บ VMware ESXI ที่ติดตั้งตัวควบคุมและ GNS3 Server

### 3.2.3 ขั้นตอนการติดตั้งใช้งานระบบต้นแบบ

1. ติดตั้งระบบปฏิบัติการ Centos หรือ Linux (ในที่นี้จะใช้ CentOS7) เพื่อใช้เป็นตัวเซิร์ฟเวอร์สำหรับติดตั้งคอนโทรลเลอร์
2. ติดตั้ง MongoDB Python3.6 Node.js ลงบนตัวเซิร์ฟเวอร์ CentOS
3. ติดตั้งโปรแกรมสำหรับทำหน้าที่เป็นคอนโทรลเลอร์
4. ติดตั้งไลบรารีที่จำเป็น
5. เปิดการทำงานโปรแกรม

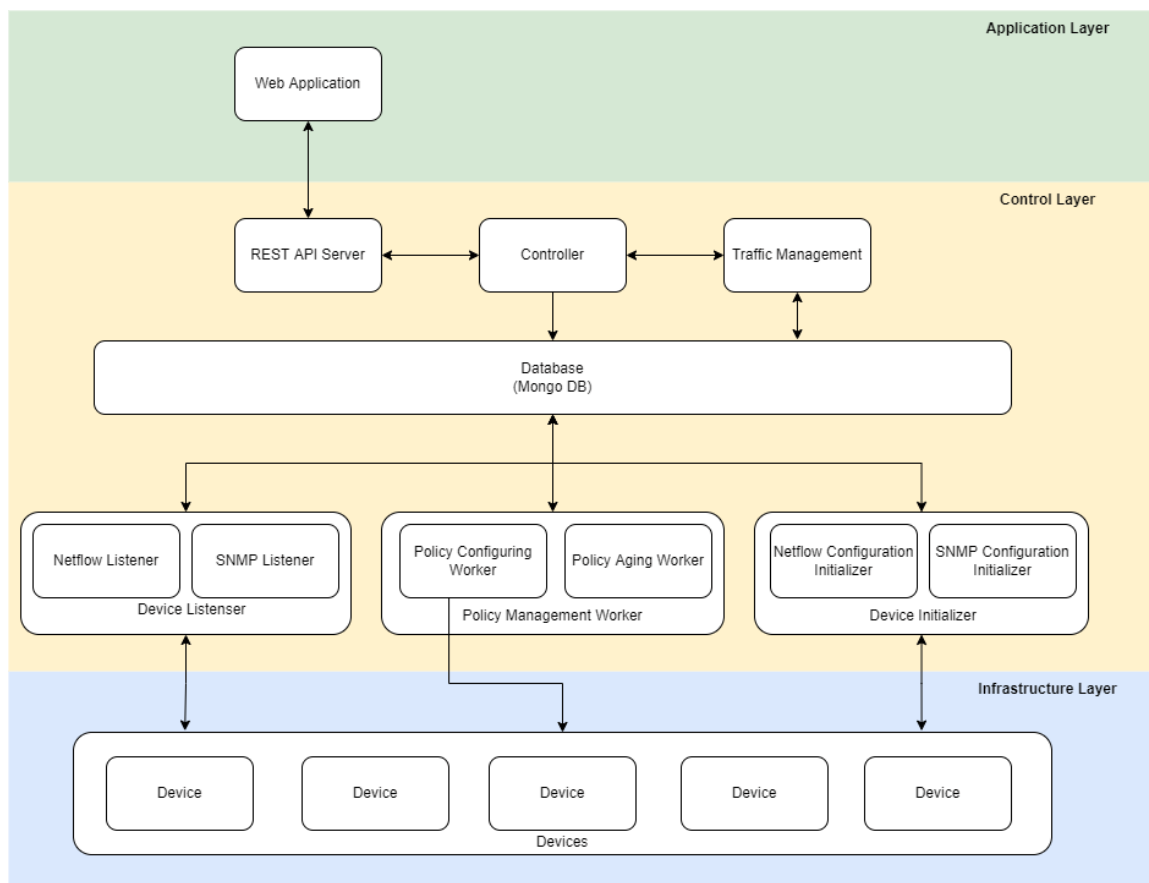
### 3.2.4 ขั้นตอนการวางแผนปรับปรุงระบบต้นแบบ

1. Application Layer ในส่วนแอปพลิเคชัน หรือฝั่งหน้าบ้านทำงานบน Vue version 2 ซึ่งเป็นเทคโนโลยีเก่า สร้างความลำบากในการดูแลและพัฒนาต่อ หากต้องการจะเพิ่ม feature ใหม่เข้าไปซึ่ง Vue 2 ไม่สนับสนุน ทั้งนี้ตัวของ Nuxt JS ยังไม่รองรับการ migrate ไปยัง Vue 3 จึงตัดสินใจที่จะพัฒนาหน้าเว็บ แอปพลิเคชันขึ้นใหม่ เพื่อให้เหมาะสมกับตัว feature ของระบบใหม่ และง่ายต่อการพัฒนาต่อในอนาคต
2. Control Layer
  1. แก้ไขการทำงานในขั้นตอนของการ initialize ให้เป็นแบบ parallel เพื่อลดเวลาทำงานให้เร็วขึ้น และแก้ไขปัญหาที่กระบวนการ initialization จะหยุดทำงานเมื่อพบกับ error ในการทำ device configuration
  2. แก้ไขกระบวนการเพิ่ม device เข้าสู่ระบบให้มีการทดสอบทำ secure shell connection ให้สำเร็จก่อนจึงจะเพิ่มอุปกรณ์เข้าสู่ระบบได้ เพื่อลดโอกาสที่จะเกิด error ในกระบวนการ initialization
  3. ปรับปรุงรูปแบบการเก็บข้อมูลในฐานข้อมูล เพื่อให้เหมาะสมกับการใช้งานของระบบใหม่
  4. ปรับปรุงการทำงานของ การตั้งค่าอุปกรณ์เพื่อทำ Policy Based Routing ผ่าน API โดยปรับปรุงใหม่ให้การตั้งไอดีต้นทางและปลายทางสามารถทำได้อย่างยืดหยุ่นไม่จำเป็นต้องใช้เฉพาะ Management IP เพียงอย่างเดียว
  5. พัฒนากลไกในการทำ policy aging เพื่อรองรับการนำไปพัฒนาเป็นแอปพลิเคชันสำหรับควบคุมแทรกฟิสิกเครือข่าย และเพื่อประหยัดพื้นที่การเก็บข้อมูลในฐานข้อมูล
  6. พัฒนา API สำหรับเรียกข้อมูลเพื่อไปแสดงผลในหน้าแอปพลิเคชันที่จะพัฒนาในอนาคต

## บทที่ 4

### ผลการดำเนินการ

#### 4.1 แผนภาพแสดงองค์ประกอบของระบบใหม่ที่ออกแบบ

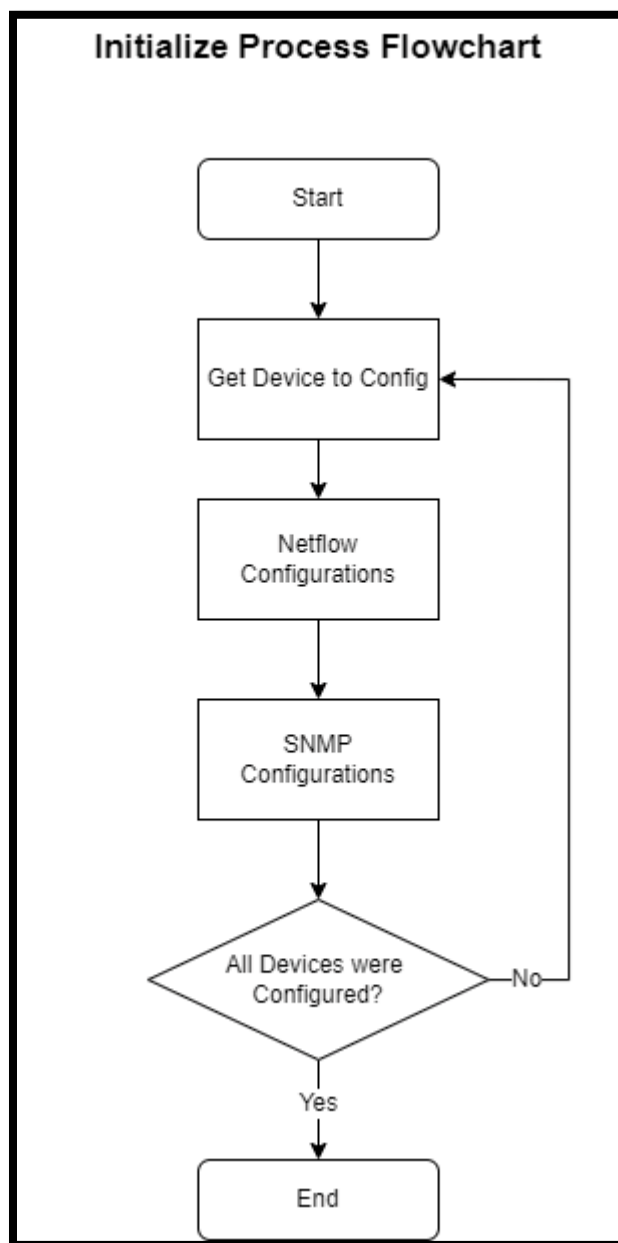


รูปที่ 4.1 องค์ประกอบของระบบใหม่

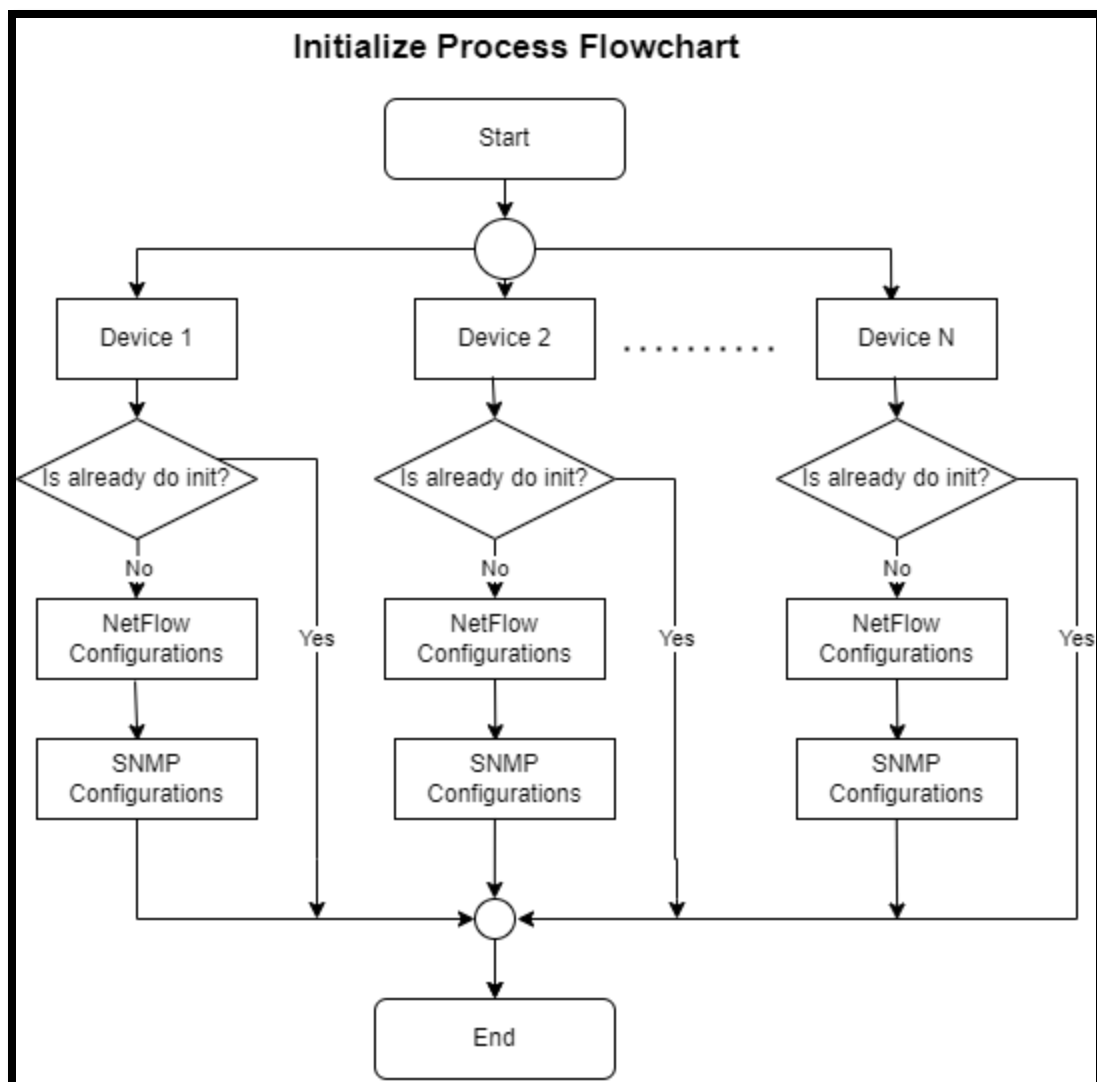


#### 4.2 แก้ไขกระบวนการตั้งค่าอุปกรณ์เครือข่ายเบื้องต้น (Initialize)

ปรับปรุงให้สามารถทำงานแบบคู่ขนาน ป้องกันกรณีที่อุปกรณ์บางตัวมีปัญหาจนระบบทำงานต่อไม่ได้จนกว่าจะสามารถตั้งค่าอุปกรณ์ดังกล่าวได้สำเร็จ



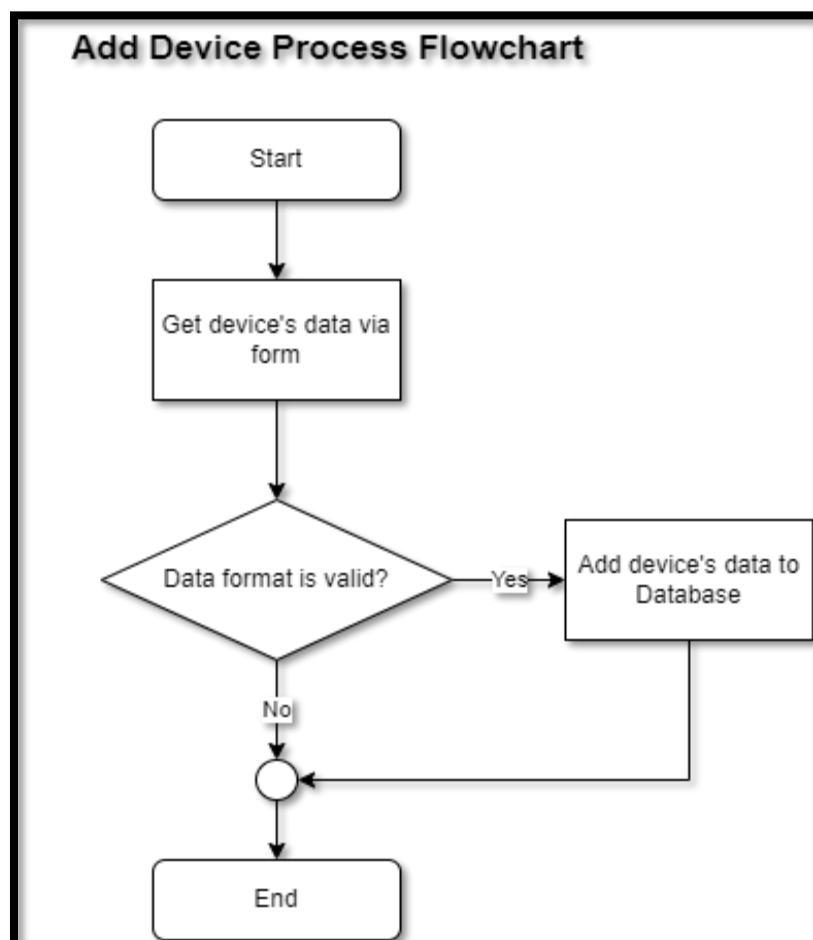
รูปที่ 4.2 ผังขั้นตอนการตั้งค่าอุปกรณ์เครือข่ายแบบเดิม



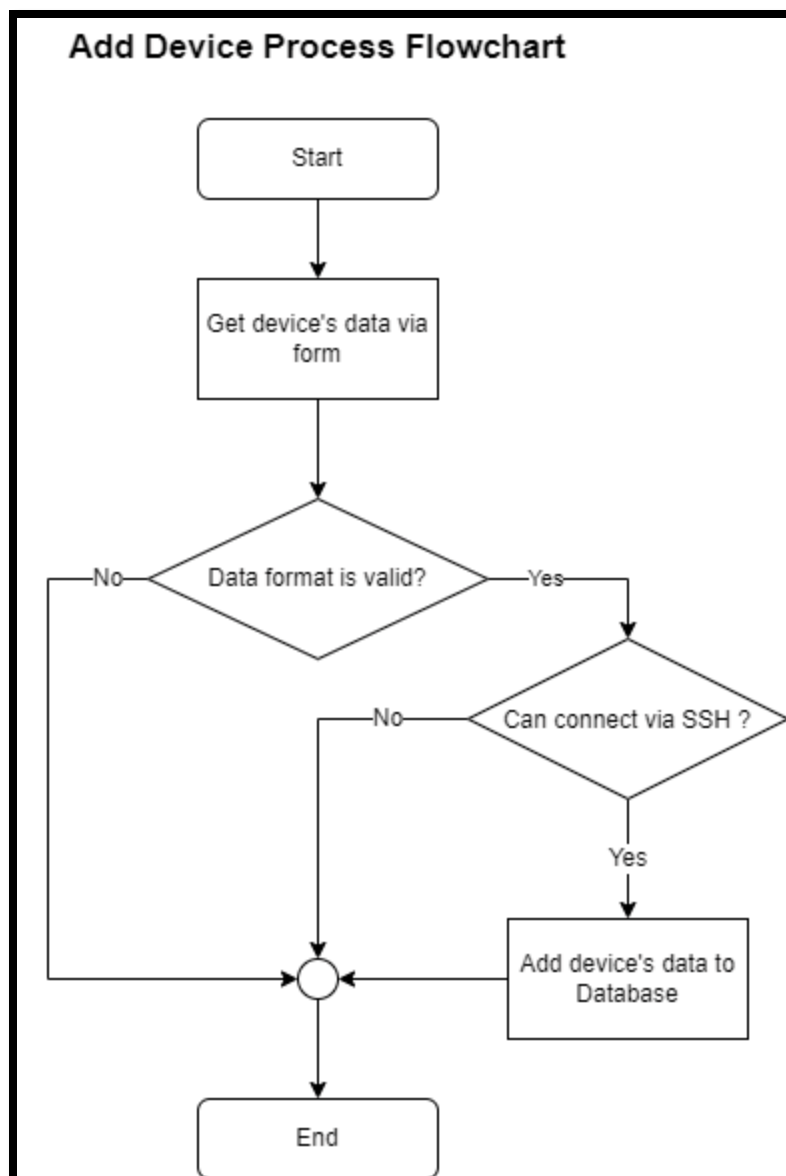
รูปที่ 4.3 ฟังก์ชันตอนการตั้งค่าอุปกรณ์เครือข่ายแบบใหม่

### 4.3 แก้ไขกระบวนการเพิ่มอุปกรณ์เข้าสู่ระบบ

ในขั้นตอนการเพิ่มอุปกรณ์จำเป็นต้องยืนยันก่อนว่าคอนโทรลเลอร์สามารถ secure shell ไปยังอุปกรณ์ดังกล่าวได้ ก่อนที่จะเพิ่มอุปกรณ์ดังกล่าวลงในฐานข้อมูลเพื่อลดปัญหาการเพิ่มอุปกรณ์ที่อาจมีปัญหาสู่ระบบ



รูปที่ 4.4 ฟังก์ชันขั้นตอนการเพิ่มอุปกรณ์เครือข่ายเข้าสู่ระบบแบบเดิม



รูปที่ 4.5 ฟังก์ชันตอนการเพิ่มอุปกรณ์เครือข่ายเข้าสู่ระบบแบบใหม่

#### 4.4 ปรับปรุงรูปแบบการเก็บข้อมูล เพื่อให้เหมาะสมกับการนำมาใช้พัฒนาต่อ

```

_id: ObjectId("61a2562c5bf3f66c46380407")
management_ip: "192.168.1.1"
serial: "96E61WAF4W671QHQBQ5T"
device_ip: "192.168.1.1"
type: "cisco_ios"
> ssh_info: Object
> snmp_info: Object
  status: 1
  snmp_is_running: false
  snmp_last_run_time: 1638177280.63685
  is_ssh_connect: true
  cdp_enable: true
  is_snmp_connect: true
  description: "Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15..."
  uptime: 517215
  name: "R1.pcn"
> interfaces: Array
  interfaces_update_time: 1638177279.950443
  updated_at: 1638177280.2342103

```

รูปที่ 4.6 เพิ่มการเก็บค่า Serial Number เพื่อใช้เป็นเอกลักษณ์เฉพาะอุปกรณ์

```

>
_id: ObjectId("61a492cb5bf3f66c4655f33e")
cisco_51: 0
direction: 0
dst_as: 0
dst_mask: 0
flow_sampler_id: 0
from_ip: "10.50.34.17"
input_snmp: 3
ipv4_dst_addr: "224.0.0.5"
ipv4_next_hop: "0.0.0.0"
ipv4_src_addr: "192.168.2.2"
l4_dst_port: 0
l4_src_port: 0
output_snmp: 0
protocol: 89
src_as: 0
src_mask: 24
src_tos: 192
tcp_flags: 16
last_switched: 2021-11-29T17:14:42.834+00:00
first_switched: 2021-11-29T17:13:46.280+00:00
in_bytes: 560
in_pkts: 7
created_at: 2021-11-29T09:17:11.537+00:00
Mbits_per_sec: 0.0000688456435798333

```

รูปที่ 4.7 เพิ่มการเก็บค่า Mbit per second เพื่อใช้ดูขนาดของ Flow

```
> _id: ObjectId("61a49d93e1382306f9a2b4eb")
> new_flow: Object
> info: Object
  created_at: 2021-11-29T09:29:55.084+00:00
  updated_at: 2021-11-29T09:30:13.218+00:00
  name: "test"
  src_ip: "192.168.7.18"
  src_port: "any"
  src_wildcard: "0.0.0.7"
  dst_ip: "192.168.4.2"
  dst_port: "any"
  dst_wildcard: "0.0.0.255"
> actions: Array
  aging_time: 0
  flow_id: 1
```

รูปที่ 4.8 เพิ่มการเก็บค่า aging\_time สำหรับการทำ Policy Aging

```
_id: ObjectId("6236ebbf6b3e4f9c9cc4c9d1")
dst_if_ip: "100.1.1.2"
dst_node_ip: "100.1.2.1"
src_if_ip: "100.1.1.1"
src_node_ip: "100.1.1.1"
src_node_id: ObjectId("6236e9896b3e4f9c9cc4bf00")
src_ip: "100.1.1.1"
src_node_hostname: "G1-1.pcn"
src_port: "GigabitEthernet0/0"
src_if_index: 1
src_in_use: 15311.666744219036
src_out_use: 5540.63714003791
dst_node_id: ObjectId("6236e9916b3e4f9c9cc4bf1a")
dst_ip: "100.1.1.2"
dst_node_hostname: "G1-2.pcn"
dst_port: "GigabitEthernet0/1"
dst_if_index: 2
dst_in_use: 5723.895990901173
dst_out_use: 19161.12584202706
link_min_speed: 1000000000
utilization_threshold: 1
> running_flows: Array
```

รูปที่ 4.9 เพิ่มการเก็บค่า running\_flows และ utilization\_threshold เพื่อดู flow และปริมาณ utilization ที่ถูกใช้งาน

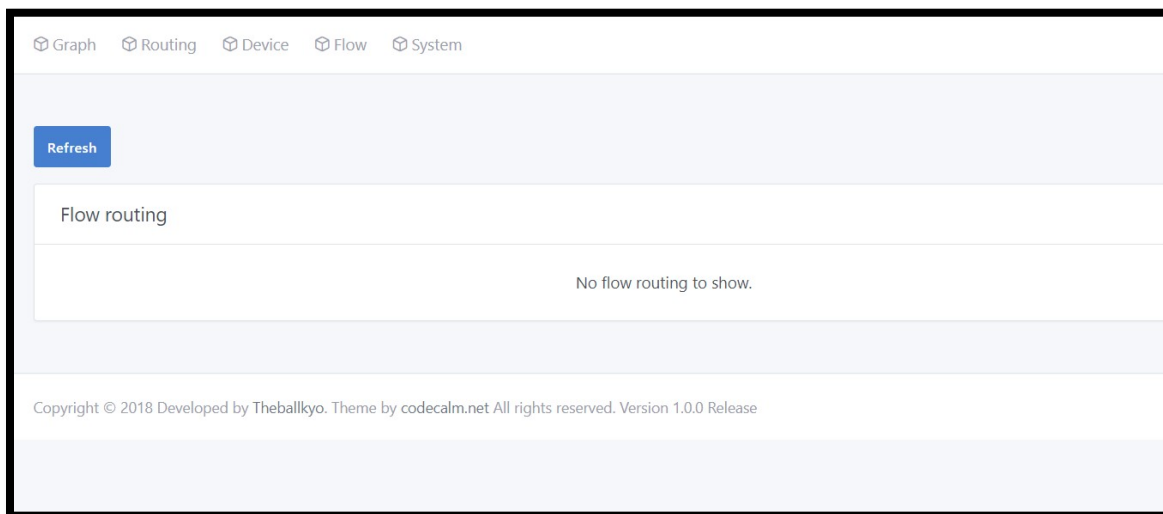
## 4.5 ปรับปรุงกระบวนการตั้งค่า Policy Based Routing ผ่าน API ให้มีความสอดคล้องกับ การทำงานจริงมากขึ้น

#	TYPE	STATUS	SOURCE	DESTINATION	ACTIONS
0	static	active	192.168.8.100/24 [any]	192.168.10.0/24 [any]	<div>DEVICE</div> <div>R7_2.pcn</div> <div>ACTION</div> <div>Next-hop IP</div> <div>192.168.7.49</div>
1	static	active	192.168.7.18/29 [any]	192.168.4.2/24 [any]	<div>DEVICE</div> <div>R7_1.pcn</div> <div>ACTION</div> <div>Next-hop IP</div> <div>192.168.7.34</div>

รูปที่ 4.10 สามารถตั้งค่า Policy โดยใช้ Source/Destination IP ตามต้องการ

ปรับปรุงให้สามารถตั้งค่า Source IP Address และ Destination IP Address สำหรับการทำให้ Policy Based Routing โดยไม่จำกัดอยู่เพียง Management IP Address ของอุปกรณ์ในเครือข่าย

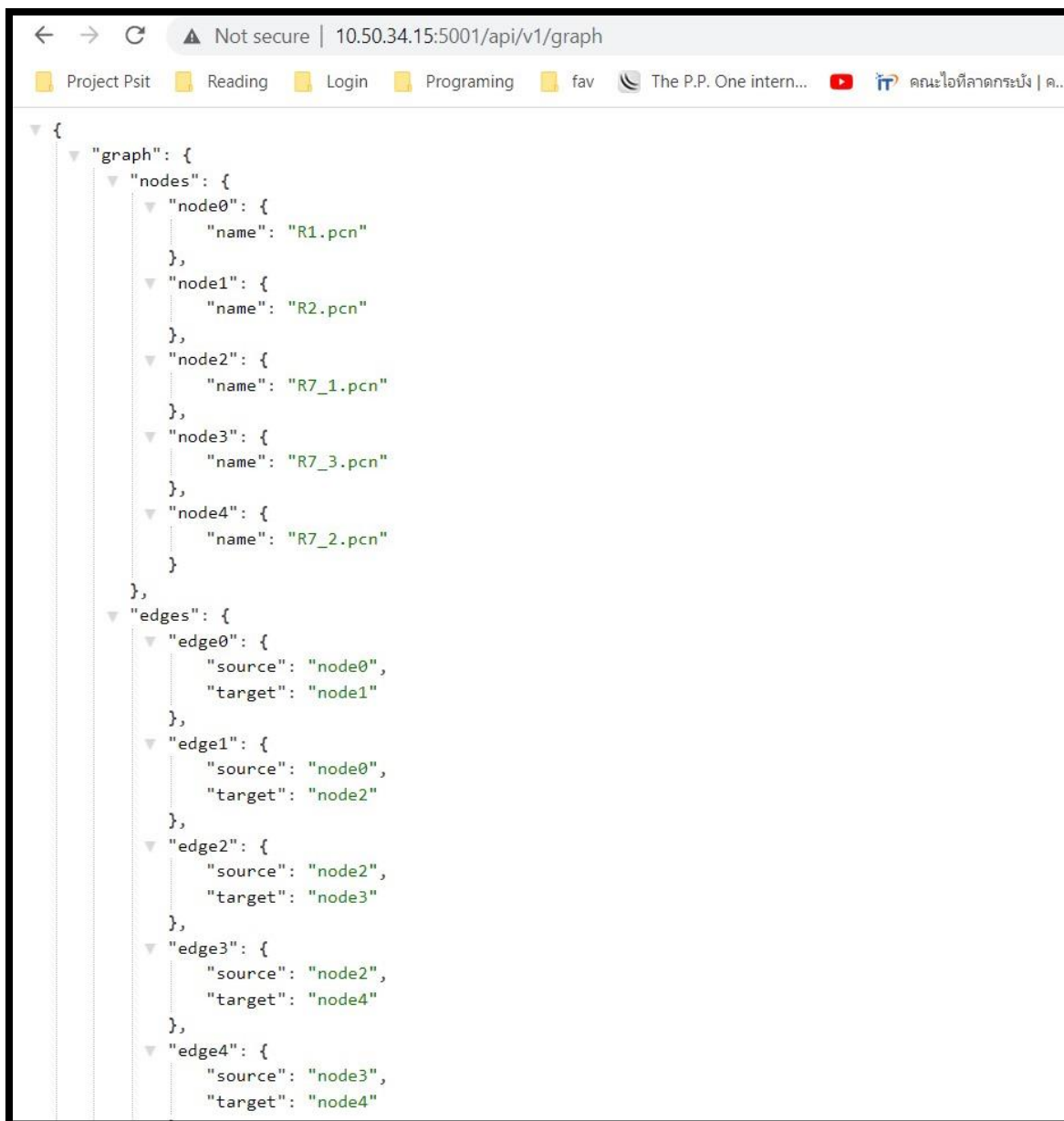
#### 4.5 พัฒนากลไก Policy Aging ถ้าหากเข้าเงื่อนไขก็สามารถลบ Policy ดังกล่าวออกไปได้



รูปที่ 4.11 เมื่อเวลาผ่านไปตามที่กำหนดจะลบ Policy ที่หมดเวลาออก



## 4.6 พัฒนา API สำหรับเรียกข้อมูลเพื่อใช้แสดงผลในหน้าแอปพลิเคชันที่จะพัฒนาในอนาคต



The screenshot shows a web browser window with the address bar displaying "10.50.34.15:5001/api/v1/graph". The browser's address bar also shows "Not secure". The page content displays a JSON response from an API, which is a graph structure. The JSON is as follows:

```
{
  "graph": {
    "nodes": {
      "node0": {
        "name": "R1.pcn"
      },
      "node1": {
        "name": "R2.pcn"
      },
      "node2": {
        "name": "R7_1.pcn"
      },
      "node3": {
        "name": "R7_3.pcn"
      },
      "node4": {
        "name": "R7_2.pcn"
      }
    },
    "edges": {
      "edge0": {
        "source": "node0",
        "target": "node1"
      },
      "edge1": {
        "source": "node0",
        "target": "node2"
      },
      "edge2": {
        "source": "node2",
        "target": "node3"
      },
      "edge3": {
        "source": "node2",
        "target": "node4"
      },
      "edge4": {
        "source": "node3",
        "target": "node4"
      }
    }
  }
}
```

รูปที่ 4.12 API สำหรับสร้างกราฟในเว็บแอปพลิเคชัน

## บทที่ 5

### บทสรุป

#### 5.1 สรุปผลโครงการ

แอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุมการกระจายแพทเทรฟฟิกแบบที่ผู้ใช้กำหนดเองได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับมาตรฐานเอสดีเอ็น เป็นแอปพลิเคชันที่พัฒนาต่อออกมาจากระบบต้นแบบ โดยระบบที่พัฒนาขึ้นมาใหม่จะทำให้ผู้ใช้งานสามารถควบคุมการกระจายแพทเทรฟฟิกตามที่ต้องการได้

โครงการนี้เสนอถึงแอปพลิเคชันที่แก้ปัญหาจากระบบต้นแบบ พร้อมปรับปรุงรูปแบบการเก็บข้อมูลให้มีประสิทธิภาพมากขึ้นและพร้อมแก่การนำไปพัฒนาต่อ และเพิ่มฟังก์ชันการทำงานที่เกี่ยวกับ Policy Aging เพื่อช่วยในการจัดการ Policy ที่ไม่ถูกใช้งานตามระยะเวลาที่ผู้ใช้กำหนด

จากการทดลองพบว่าระบบสามารถป้องกันปัญหา การหยุดทำงานที่เกิดขึ้นระหว่างกระบวนการทำInitialize โดยการแยกกระบวนการในการตั้งค่าอุปกรณ์ให้ทำงานแบบ parallel และยังสามารถเรียกใช้ข้อมูลโครงสร้างระบบเครือข่ายผ่านทาง API ได้ ระบบสามารถเรียกใช้ข้อมูลโครงสร้างเครือข่ายข้อมูลอุปกรณ์ ข้อมูลการเชื่อมต่ออุปกรณ์ ข้อมูลแพทเทรฟฟิกในเครือข่ายผ่านทาง API ได้ทั้งยังสามารถกำหนดการเลือกเส้นทาง แบบชั่วคราวและถาวรให้กับแพทเทรฟฟิกในเครือข่ายได้ โดยส่งการผ่านทาง API ของระบบ

#### 5.2 ปัญหาในการทำโครงการและสรุปผล

1. ปัญหาโรคระบาดโควิด 19 ทำให้ไม่สามารถเข้าไปใช้งานทรัพยากรของทางคณะได้อย่างอิสระ
2. เทคโนโลยีที่ใช้พัฒนาในส่วนเว็บแอปพลิเคชัน ทางผู้จัดทำต้องใช้เวลาศึกษาเพิ่มเติมอย่างละเอียดทำให้ดำเนินการได้ล่าช้า

### 5.3 แนวทางการพัฒนาต่อ

1. ทำส่วนเว็บแอปพลิเคชันใหม่ด้วย Vue version 3 ปรับปรุงการแสดงผลข้อมูลเครือข่ายใหม่ให้สะดวกต่อการใช้งานมากขึ้น
2. พัฒนาส่วนติดต่อผู้ใช้งานในการเลือกเส้นทาง แทรฟฟิกเครือข่ายให้เป็นมิตรกับผู้ใช้งาน
3. พัฒนาส่วนติดต่อผู้ใช้งานในการเพิ่มอุปกรณ์เครือข่ายเข้าสู่ระบบ
4. พัฒนาแอปพลิเคชันในการกระจายแทรฟฟิกในเครือข่ายผ่านทางตัวควบคุม

## บรรณานุกรม

- [1] Ciena. “Networking Insights What is SDN.”[Online].Available: [www.ciena.com/insights/what-is/What-Is-SDN.html](http://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] Fortinet, Inc, “Network Traffic.”[Online].Available: [www.fortinet.com/resources/cyberglossary/network-traffic](http://www.fortinet.com/resources/cyberglossary/network-traffic)
- [4] Juniper. “what-is-policy-based-routing”[Online].Available: [www.juniper.net/us/en/research-topics/what-is-policy-based-routing.html](http://www.juniper.net/us/en/research-topics/what-is-policy-based-routing.html)
- [5] Saixiii. “SNMP คืออะไร โปรโตคอลสำหรับมอนิเตอร์อุปกรณ์ในระบบ”[Online].Available: [www.saixiii.com/what-is-snmp/](http://www.saixiii.com/what-is-snmp/)
- [6] Solarwinds, “What is NetFlow?”[Online].Available: [www.solarwinds.com/netflow-traffic-analyzer/use-cases/what-is-netflow](http://www.solarwinds.com/netflow-traffic-analyzer/use-cases/what-is-netflow)
- [7] Cisco. “Cisco Discovery Protocol (CDP)”[Online].Available: [www.learningnetwork.cisco.com/s/article/cisco-discovery-protocol-cdp-x](http://www.learningnetwork.cisco.com/s/article/cisco-discovery-protocol-cdp-x)
- [8] Python. “What is Python? Executive Summary”[Online].Available: [www.python.org/doc/essays/blurb/](http://www.python.org/doc/essays/blurb/)
- [9] Chai Phonbopit “MongoDB คืออะไร? + สอนวิธีใช้งานเบื้องต้น”[Online].Available: <https://devahoy.com/blog/2015/08/getting-started-with-mongodb/>

## 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](mailto: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 ท่าจั่ว อำเภอ เมือง จังหวัด นครศรีธรรมราช

เบอร์โทร 089-4728789 Email [610702171@kmitl.ac.th](mailto:610702171@kmitl.ac.th)

สาขาที่จบ เทคโนโลยีสารสนเทศ รุ่นที่ 16 ปีการศึกษาที่จบ 2564

