**การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุมการกระจายแทรฟฟิกแบบที่ผู้ใช้กำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับมาตรฐานเอสดีเอ็น**

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

**โดย**

**พงศ์พณิช อรัญรัตน์โสภณ**

**ภูริณัฐ จิตมนัส**

**อาจารย์ที่ปรึกษา**

**ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต**

**สาขาวิชาเทคโนโลยีสารสนเทศ**

**คณะเทคโนโลยีสารสนเทศ**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ภาคเรียนที่ 2 ปีการศึกษา 2564**

# สารบัญ

**หน้าที่**

[สารบัญ i](#_Toc106789769)

บทที่ 1 [ภาพรวมระบบ 1](#_Toc106789770)

บทที่ 2 [การติดตั้งระบบ และโปรแกรมที่จำเป็นต่อการใช้งาน 3](#_Toc106789771)

[2.1 การติดตั้งภาษาไพธอน 3](#_Toc106789772)

[2.2 การติดตั้งคอนโทรลเลอร์ 5](#_Toc106789773)

[2.3 การติดตั้งฐานข้อมูล 5](#_Toc106789774)

[2.4 การติดตั้งแอปพลิเคชันส่วน Frontend (Node.js Yarn และ Vue3) 6](#_Toc106789775)

[2.5 การติดตั้งโมดูลอื่นๆที่จำเป็น 10](#_Toc106789776)

[2.6 การเปิดใช้งานคอนโทรลเลอร์ 11](#_Toc106789777)

บทที่ 3 [การเชื่อมต่อคอนโทรลเลอร์กับอุปกรณ์เครือข่าย 14](#_Toc106789778)

[3.1 การตั้งค่าอุปกรณ์เครือข่าย 14](#_Toc106789779)

[3.2 การใช้งานหน้าเว็บเพื่อควบคุมคอนโทรลเลอร์ 15](#_Toc106789780)

[3.3 การตั้งค่าเริ่มต้นให้อุปกรณ์เครือข่ายผ่านการเรียกใช้คำสั่งจากโค้ดไพธอนโดยตรง 20](#_Toc106789781)

บทที่ 4 [ข้อมูลชุดคำสั่งเอพีไอที่สามารถใช้งานได้และตัวอย่างการใช้งาน 21](#_Toc106789782)

[4.1 รายละเอียดคำสั่ง “device” 21](#_Toc106789783)

[4.2 รายละเอียดคำสั่ง “flow” 22](#_Toc106789784)

[4.3 รายละเอียดคำสั่ง “link” 23](#_Toc106789785)

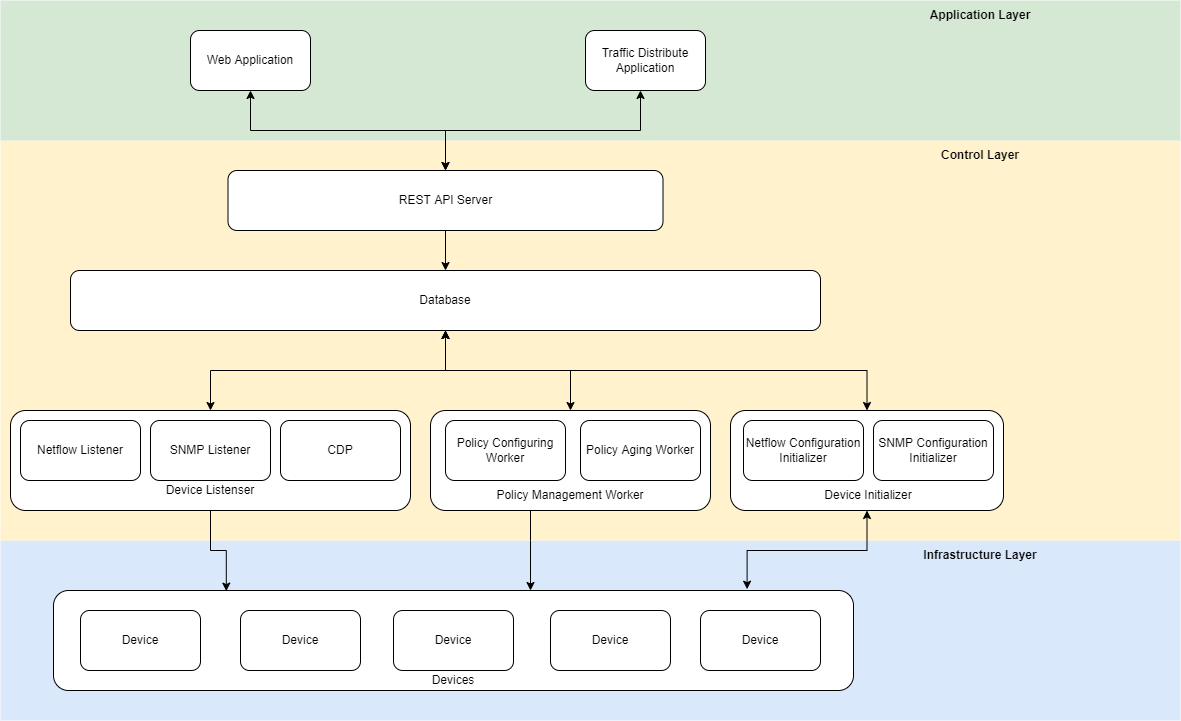
[4.4 รายละเอียดคำสั่ง “path” 24](#_Toc106789786)

[4.5 รายละเอียดคำสั่ง “flow/routing” 25](#_Toc106789787)

**บทที่ 1**

# ภาพรวมระบบ

ชุดทดลองระบบจัดการเครือข่ายตามสถาปัตยกรรมแบบเอสดีเอ็น เป็นระบบที่ถูกออกแบบมาเพื่อทำให้อุปกรณ์เครือข่ายที่ไม่รองรับการทำงานภายใต้สถาปัตยกรรมแบบเอสดีเอ็น (Software Defined Network: SDN) สามารถถูกควบคุมจากระบบดังกล่าวเพื่อที่จะช่วยอำนวยความสะดวก ให้กับผู้ควบคุมระบบเครือข่าย ให้สามารถมองเห็นภาพรวม และควบคุมระบบเครือข่ายน้ันผ่านระบบคอนโทรลเลอร์ซึ่งทำหน้าที่เสมือนระบบที่ถูกออกแบบภายใต้สถาปัตยกรรมแบบเอสดีเอ็นได้ โดยภาพรวมของระบบที่ถูกพัฒนาขึ้นจะถูกแสดงเอาไว้ตามรูปที่ 1.1



**รูปที่ 1.1** ภาพรวมของระบบ

ระบบประกอบไปด้วยอุปกรณ์เครือข่ายที่เชื่อมต่อเข้ากับตัวควบคุม ซึ่งตัวควบคุมจำเป็นต้องสามารถเชื่อมต่อโดย Secure Shell (SSH) ไปยังอุปกรณ์เครือข่ายได้ เพื่อส่งคำสั่งตั้งค่า สำหรับเปิดใช้งาน SNMP และ NetFlow สำหรับเก็บข้อมูลเครือข่าย และส่งคำสั่งตั้งค่า Policy Based Routing

ข้อมูลที่เก็บมาจากอุปกรณ์เครือข่ายจะถูกบันทึกลงในฐานข้อมูล โดยจะมีหน้าเว็บแอปพลิเคชันที่จะดึงข้อมูลจากฐานข้อมูลผ่านทาง REST API ตัวแอปพลิเคชันจะทำหน้าที่แสดงข้อมูลจากระบบเครือข่ายให้ผู้ใช้สามารถดูและทำความเข้าใจได้ง่าย

จากที่กล่าวมาข้างต้นทำให้แอปพลิเคชันตัวควบคุมจะประกอบไปด้วยระบบหลักๆ 4 ส่วนดังต่อไปนี้

1. Controller Application เป็นแอปพลิเคชันที่ทำหน้าที่ประมวลผล ส่งคำสั่งตั้งค่าไปยังอุปกรณ์เครือข่าย รวมถึงรับข้อมูลจากอุปกรณ์เครือข่ายเพื่อบันทึกลงฐานข้อมูล
2. Database เป็นส่วนที่เก็บข้อมูลต่าง ๆ ของระบบเครือข่าย
3. Web Application เป็นส่วนที่ช่วยให้ผู้ใช้สามารถส่งคำสั่งเพิ่ม-ลบอุปกรณ์ เปิดการเก็บข้อมูลจาก SNMP และ NetFlow และใช้แสดงผลเครือข่าย
4. API Module เป็นส่วนที่ถูกใช้ในการรับส่งคำสั่งระหว่างผู้ใช้ ตัวควบคุม และอุปกรณ์เครือข่าย ช่วยสามารถใช้เขียนโปรแกรมเพื่อควบคุมโดยตรง หรือสั่งการผ่านหน้าเว็บแอปพลิเคชันได้

**บทที่ 2**

# การติดตั้งระบบ และโปรแกรมที่จำเป็นต่อการใช้งาน

ระบบคอนโทรลเลอร์ ถูกพัฒนาขึ้นภายใต้ระบบปฏิบัติการ Linux ซึ่งในที่นี้ผู้พัฒนาได้ทำการพัฒนาระบบภายใต้ระบบปฏิบัติการ CentOS 7 แต่ในขณะเดียวกันระบบคอนโทรลเลอร์นี้ก็ยังนำไปใช้กับระบบปฏิบัติการอื่นได้เช่นกัน แต่รูปแบบการติดตั้งอาจจะแตกต่างกันไปตามแต่คำสั่งของระบบปฏิบัติการ โดยในคู่มือฉบับนี้ผู้พัฒนาระบบจะแสดงวิธีการติดตั้งผ่านระบบคอนโทรลเลอร์ภายใต้ระบบปฏิบัติการ CentOS 7 เพื่อเป็นตัวอย่าง

เพื่อให้ระบบคอนโทรลเลอร์สามารถเข้าถึงอุปกรณ์เครือข่ายได้นั้น ผู้ใช้จะต้องติดตั้งระบบ คอนโทรลเลอร์นั้นลงไปบนเซิร์ฟเวอร์หรือคอมพิวเตอร์ที่สามารถติดต่อกับอุปกรณ์เครือข่ายที่ต้องการควบคุมได้โดยในขั้นแรกผู้ใช้ต้องทำการติดตั้งโปรแกรมที่จำเป็นต่อการใช้งานระบบเสียก่อน โดยมีขั้นตอนดังต่อไปนี้

## 2.1 การติดตั้งภาษาไพธอน

ระบบคอนโทรลเลอร์นั้นได้ถูกพัฒนาผ่านภาษาไพธอนเวอร์ชัน 3.6 (Python 3.6) ดังน้ัน ผู้ใช้งานจำเป็นต้องติดตั้งภาษาดังกล่าว รวมทั้งเครื่องมือที่เกี่ยวข้องก่อนเริ่มใช้งาน โดยมีขั้นตอน ดังต่อไปนี้ (การติดตั้งจำเป็นต้องดาวน์โหลดข้อมูลผ่านอินเตอร์เน็ต)

1. เปิดใช้งานเทอร์มินัล
2. พิมคำสั่ง yum install update -y เพื่อปรับปรุงเวอร์ชันแพ็คเกจให้เป็นปัจจุบันพร้อมใช้งาน
3. พิมคำสั่งดังต่อไปนี้เพื่อติดตั้งไพธอนและไลบรารีที่จำเป็น
   1. yum install -y python3
   2. pip3.6 install --upgrade pip

Text

Description automatically generated

**รูปที่ 2.1** อัพเดทแพ็คเกจสำเร็จ

Text

Description automatically generated

**รูปที่ 2.2** ติดตั้ง Python3 และ Pip3 สำเร็จ

## 2.2 การติดตั้งคอนโทรลเลอร์

ระบบถูกแบ่งออกเป็นสามส่วน ได้แก่ Control module, API module และ Webpage module โดยวิธีการติดตั้งนั้นผู้ใช้สามารถดาวน์โหลดไฟล์ดังกล่าวจากเว็บไซต์ Github และติดตั้งผ่านเทอร์มินัลได้ โดยวิธีการติดตั้งผ่านเทอร์มินัลนั้นมีรายละเอียดดังต่อไปนี้

1. พิมพ์คำสั่ง yum install git -y เพื่อติดตั้ง github
2. พิมพ์คำสั่ง git clone https://github.com/FameIllusionMaya/SDN-handmade\_v3.git

## 2.3 การติดตั้งฐานข้อมูล

ผู้พัฒนาระบบได้เลือกใช้ฐานข้อมูล mongo เพื่อเก็บข้อมูลของระบบเครือข่ายโดยมีขั้นตอนดังต่อไปนี้

1. เรียกใช้คำสั่ง vi /etc/yum.repos.d/mongodb-org.repo เพื่อสร้าง และเปิดไฟล์ใหม่เพื่อใช้ในการติดตั้งฐานข้อมูล
2. จากนั้นให้เพิ่มรายละเอียดของไฟล์ตามที่แสดงไว้ตามรูปที่ 2.3 หลังจากนั้นบันทึกและปิดไฟล์ด้วยคำสั่ง :wq
3. พิมพ์คำสั่ง yum install mongodb-org -y เพื่อติดตั้ง mongoDB
4. พิมพ์คำสั่ง systemctl start mongod เพื่อเปิดใช้งานฐานข้อมูล
5. เพื่อทดสอบว่าฐานข้อมูลถูกติดตั้งสำเร็จ และสามารถใช้งานได้ ให้ผู้ใช้พิมพ์คำสั่ง **mongo** ลงไปในคอนโซล ถ้าระบบแสดงหน้าต่างดังรูปที่ 2.4 แสดงว่าผู้ใช้สามารถติดตั้งฐานข้อมูลสำเร็จ

Graphical user interface, text, application

Description automatically generated

**รูปที่ 2.3** รายละเอียดข้อมูลที่ต้องเพิ่มลงไปในไฟล์ที่ถูกสร้างขึ้น

Text

Description automatically generated

**รูปที่ 2.4** ติดตั้งฐานข้อมูลสำเร็จ

## 2.4 การติดตั้งแอปพลิเคชันส่วน Frontend (Node.js Yarn และ Vue3)

เนื่องจากผู้พัฒนาได้ใช้ Node.js Yarn และ Vue3 ในการพัฒนาหน้าเว็บที่ใช้ในการติดต่อสื่อสารกับระบบดังนั้นผู้ใช้จำเป็นต้องติดตั้ง Node.js Yarn และ Vue3 เพื่อใช้งานหน้าเว็บดังกล่าว โดยให้ผู้ใช้ไปยัง Directory SDN-handmade\_v3/sdn\_frontendv3 และเรียกใช้คำสั่งดังต่อไปนี้

1. curl -sL https://rpm.nodesource.com/setup\_16.x | sudo bash -
2. yum install nodejs -y
3. npm install -g @vue/cli
4. curl --silent --location https://dl.yarnpkg.com/rpm/yarn.repo | sudo tee /etc/yum.repos.d/yarn.repo
5. rpm --import <https://dl.yarnpkg.com/rpm/pubkey.gpg>
6. yum install yarn -y

A screenshot of a computer screen

Description automatically generated with medium confidence

**รูปที่ 2.5** ติดตั้ง Nodejs สำเร็จ

Text

Description automatically generated

**รูปที่ 2.6** ติดตั้ง Vue3 สำเร็จ

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 2.7** ติดตั้ง Yarn สำเร็จ

และผู้ใช้ปิด Firewall โดยคำสั่ง

1. systemctl stop firewalld
2. systemctl disable firewalld
3. systemctl mask --now firewalld

หลังจากนั้นให้ผู้ใช้แก้ไขข้อมูลในไฟล์ที่ชื่อ httpclient.ts (อยู่ใน directory SDN/sdn\_frontendv3/src/) โดยเปลี่ยน baseURL ให้ตรงกับไอพีของเครื่องคอนโทรลเอลร์

Text

Description automatically generated

**รูปที่ 2.8** ปิด Firewall สำเร็จ และไอพีของคอนโทรลเลอร์คือ 10.50.34.25

Graphical user interface, text, application

Description automatically generated with medium confidence

**รูปที่ 2.9** แก้ไขไอพีให้ตรงกับคอนโทรลเลอร์

กลับไปที่ Directory SDN/sdn\_frontendv3 ลบตัว node modules เก่าและลงใหม่โดยใช้คำสั่ง

1. rm -rf node\_modules/
2. npm install

Text

Description automatically generated

**รูปที่ 2.10** ติดตั้ง npm สำเร็จ

## 2.5 การติดตั้งโมดูลอื่น ๆ ที่จำเป็น

ผู้ใช้จำเป็นต้องติดตั้งโมดูลที่เหลือบางส่วนเพื่อให้คอนโทรลเลอร์สามารถใช้งานได้ โดยให้ผู้ใช้ไปยัง Directory SDN-handmade\_v3/backend/src และเรียกใช้คำสั่งดังต่อไปนี้

1. pip3.6 install --upgrade pip
2. pip3 install -r requirements.txt
3. pip3 install -U sanic-cors
4. python3 -m pip install --upgrade textfsm==0.4.1
5. pip3 install requests
6. yum install python3-tkinter

Text

Description automatically generated

**รูปที่ 2.11** ติดตั้ง requirement สำเร็จ

Text

Description automatically generated

**รูปที่ 2.12** ติดตั้ง sanic-cors และ textfsm สำเร็จ

## 2.6 การเปิดใช้งานคอนโทรลเลอร์

1. เปิดใช้งานฐานข้อมูลโดยคำสั่ง **systemctl start mongod**
2. เปิดใช้งานคอนโทรลเลอร์โดย ไปที่ Directory /SDN-handmade\_v3/backend/src พิมพ์คำสั่ง **python3 main.py**
3. เปิดใช้งานโมดูลเอพีไอโดย ไปที่ Directory /SDN-handmade\_v3/backend/src พิมพ์คำสั่ง **python3 main\_web.py**
4. เปิดใช้งานหน้าเว็บโดย ไปที่ Directory /SDN-handmade\_v3/sdn\_frontend พิมพ์คำสั่ง **yarn serve** (กรณีเปิดใช้งานหน้าเว็บมีปัญหาให้ลบ node\_modules และลง npm ใหม่)

A picture containing text

Description automatically generated

**รูปที่ 2.13** เปิดใช้งานคอนโทรลเลอร์สำเร็จ

Text

Description automatically generated

**รูปที่ 2.14** เปิดใช้งานโมดูลเอพีไอสำเร็จ

Text

Description automatically generated

**รูปที่ 2.15** เปิดใช้งานหน้าเว็บสำเร็จ

**บทที่ 3**

# การเชื่อมต่อคอนโทรลเลอร์กับอุปกรณ์เครือข่าย

การที่ระบบคอนโทรลเลอร์จะสามารถติดต่อกับอุปกรณ์เครือข่ายได้นั้น จะเกิดขึ้นจากการที่ ผู้ใช้ตั้งค่าอุปกรณ์เครือข่ายเพื่อให้ระบบคอนโทรลเลอร์สามารถส่งคำสั่งเพื่อตั้งค่า SNMP ให้แก่ อุปกรณ์เครือข่ายก่อนที่ Collector module (SNMP และ NetFlow) จะสามารถดึงข้อมูลของเราท์เตอร์เข้ามาเก็บไว้เพื่อการประมวลผลในฐานข้อมูลได้ และในขณะเดียวกันคอนโทรลเลอร์ก็ต้องทราบว่าจะต้องติดต่อไปยัง อุปกรณ์ใน subnet ใด และผ่านเส้นทางใด รวมถึงมีอุปกรณ์เครือข่ายใดอยู่ในระบบบ้าง

## 3.1 การตั้งค่าอุปกรณ์เครือข่าย

ระบบคอนโทรลเลอร์ถูกออกแบบมาให้สามารถตั้งค่า SNMP ให้กับอุปกรณ์เครือข่ายผ่าน ทางการเชื่อมต่อระยะไกลผ่านโพรโทคอล Secure shell (SSH) ดังนั้นหลังจากที่ผู้ใช้ได้ทำการ กำหนดที่อยู่ไอพี และกำหนดวิธีการค้นหาเส้นทางเพื่อเชื่อมต่ออุปกรณ์เครือข่ายเข้าด้วยกันแล้ว ผู้ใช้จำเป็นจะต้องตั้งค่าอุปกรณ์เครือข่ายทั้งหมด ผ่านชุดคำสั่งเหล่านี้เพื่อให้ระบบคอนโทรลเลอร์ สามารถส่งคำ สั่งผ่านโพรโทคอล SSH ได้อย่างถูกต้อง

Router(config)#ip domain-name pcn //ผู้ใช้ตั้งชื่อโดเมนเนมได้อย่างอิสระ (ในตัวอย่างนี้ถูกตั้งเป็น pcn)

Router (config)#crypto key generate rsa

How many bits : 1024

Router (config)#username cisco password cisco

Router (config)#ip ssh version 2 //ผู้ใช้สามารถตั้ง username และ password ได้อย่างอิสระ แต่ในตัวอย่างจะใช้ทั้ง username และ password เป็น cisco

Router (config)#enable password cisco

Router (config)#line vty 0 xxx //กำหนดจำนวน channel ในการเชื่อมต่อโดยค่า xxx ผู้พัฒนาแนะนำให้ตั้งค่าสูงสุดที่เป็นไปได้เพื่อให้ระบบคอนโทรลเลอร์สามารถติดต่อกับเราท์เตอร์ได้อย่างมีประสิทธิภาพ

Router (config-line)#login local

Router (config-line)#transport input ssh

Router (config-line)#exit

## 3.2 การใช้งานหน้าเว็บเพื่อควบคุมคอนโทรลเลอร์

ในการเข้าถึงหน้าเว็บเพื่อใช้ควบคุมระบบคอนโทรลเลอร์นั้น ผู้ใช้สามารถเปิดใช้งานเบราเซอร์ใด ๆ และพิมพ์ไอพีของเครื่องเซิร์ฟเวอร์ที่ทำ การเปิดโมดูลทั้งหมดเอาไว้ตามด้วยหมายเลขพอร์ตที่ถูกตั้งเอาไว้ตามหัวข้อ 2.4 ในตัวอย่างตามรูปที่ 3.1 หมายเลขพอร์ตที่กำหนดไว้ค่าเท่ากับ 8080 และไอพีของเครื่องเซิร์ฟเวอร์คือ 10.50.34.25

Graphical user interface, application

Description automatically generated

**รูปที่ 3.1** ตัวอย่างหน้าเว็บไซต์สำหรับแสดงผล

เพื่อให้หน้าเว็บสามารถเข้าถึงอุปกรณ์เครือข่ายได้ ให้ผู้ใช้เลือกไปที่แท็บ “Devices” ซึ่งก็จะปรากฏหน้าเว็บตามรูปที่ 3.2 โดยผู้ใช้จะต้องทำการเพิ่มข้อมูลอุปกรณ์เครือข่ายทั้งหมดลงไปในฟอร์มทางขวามือ โดยกำหนด SSH username และ SSH password ให้ตรง กับที่ตั้งไว้ในอุปกรณ์ตามหัวข้อที่ 3.1 โดยรายละเอียดอื่น ๆ มีดังต่อไปนี้

1. ใส่หมายเลข Management IP ของตัว Router

2. ใส่ประเภทอุปกรณ์เครือข่าย ซึ่งปัจจุบันรองรับเพียง Cisco IOS

3. ใส่ชื่อผู้ใช้สำหรับการเข้าถึงอุปกรณ์เครือข่ายผ่าน SSH (SSH Username)

4. ใส่รหัสผ่านสำหรับการเข้าถึงอุปกรณ์เครือข่ายผ่าน SSH (SSH Password)

5. ใส่รหัสการเปิดใช้งานอุปกรณ์เครือข่าย (Enable Password)

6. ใส่หมายเลขพอร์ต SSH ซึ่งค่าเริ่มต้นคือ 22

7. ใส่เวอร์ชันของ SNMP ซึ่งปัจจุบันรองรับเพียงเวอร์ชัน 2c

8. ใส่ค่า SNMP Community string ซึ่งค่าเริ่มต้นคือ public

9. ใส่หมายเลขพอร์ตของ SNMP ซึ่งค่าเริ่มต้นคือ 161

Graphical user interface, text, application

Description automatically generated

**รูปที่ 3.2** ตัวอย่างหน้าเว็บไซต์สำหรับแสดงผลหน้า Devices

จากนั้น เพื่อให้ระบบสามารถส่งคำสั่งเข้าไปตั้งค่าอุปกรณ์เครือข่ายได้ผู้ใช้ต้องทำการตั้งค่า เราท์ติ้งเพื่อไปยัง Subnet ของอุปกรณ์เครือข่าย ผ่านคำสั่ง vi /etc/sysconfig/network-scripts/route-ens192 โดย ens192 คือชื่ออินเตอร์เฟส จากตัวอย่างตามรูปที่ 3.3 ผู้ใช้ได้ทำการตั้งค่าไอพี ของอุปกรณ์เครือข่ายเอาไว้ในกลุ่ม Subnet 192.168.0.0/16 โดยคอนโทรลเลอร์จะสามารถติดต่อกับ อุปกรณ์ผ่านอินเตอร์เฟสที่มีหมายเลขไอพีคือ 10.50.34.23 (Gateway)

Text

Description automatically generated with medium confidence

**รูปที่ 3.3** ตัวอย่างการตั้งค่าเราท์ติ้ง

ในขั้นต่อไปให้ผู้ใช้เลือกไปที่แท็บ “Initialization” จะปรากฎหน้าตามรูปที่ 3.4 จากนั้นใส่ข้อมูลหมายเลขไอพีของเครื่องที่ทำหน้าที่เป็นเซิร์ฟเวอร์ให้ระบบคอนโทรลเลอร์แล้วกด คลิกที่ปุ่ม “Init SNMP” เพื่อส่งคำสั่งในการตั้งค่า SNMP ไปยังอุปกรณ์ทั้งหมดที่ได้ทำการเพิ่ม เอาไว้ในหน้าเว็บ เมื่อระบบเชื่อมต่อกับอุปกรณ์สำเร็จ และตั้งค่าอุปกรณ์เครือข่ายสำเร็จแล้วหน้าเว็บจะสามารถแสดงผลในแท็บ “Devices” ดังรูปที่ 3.5 โดยที่สถานะ SSH และ SNMP ต้องขึ้นเป็นไอคอนสีเขียว โดยผู้ใช้สามารถกดดูข้อมูลของอุปกรณ์เครือข่ายใด ๆ ได้โดยกดไปที่แถบข้อมูลของ อุปกรณ์เครือข่ายนั้น ซึ่งจะแสดงผลตามรูปที่ 3.6

Graphical user interface, application, Word

Description automatically generated

**รูปที่ 3.4** ตัวอย่างหน้าเว็บไซต์สำหรับแสดงผลหน้า Initialization

Graphical user interface, application

Description automatically generated

**รูปที่ 3.5** หน้า Devices แสดงผลอุปกรณ์ที่เพิ่มเข้ามา

Graphical user interface

Description automatically generated

**รูปที่ 3.6** หน้า Devices สามารถกดที่อุปกรณ์เพื่อดูรายละเอียด

เมื่อระบบสามารถเชื่อมต่อกับอุปกรณ์ทั้งหมดได้สำเร็จ ระบบจะสามารถแสดงโทโพโลยีของเครือข่ายออกมาได้โดยให้ผู้ใช้เลือกไปที่แท็บ “Home” ซึ่งจะแสดงผลออกมาตามรูปที่ 3.7

A picture containing graphical user interface

Description automatically generated

**รูปที่ 3.7** ตัวอย่างการแสดงผลโทโพโลยีเครือข่าย

ข้อมูลที่วิ่งอยู่ในอุปกรณ์เครือข่ายได้ผู้ใช้ต้องเลือก ไปที่แท็บ “Initialization” และกดเลือกไปที่หัวข้อ “Initialization” ตามรูปที่ 3.4 จากนั้นใส่ข้อมูล หมายเลขไอพีของเครื่องที่ทำหน้าที่เป็นเซิร์ฟเวอร์ให้ระบบคอลโทรลเลอร์แล้วกดที่ปุ่ม “Init Netflow” เพื่อให้ระบบคอนโทรลเลอร์สามารถดึงข้อมูลของโฟลว์มาแสดงได้ (ขั้นตอนนี้จำเป็นต้องทำหลังจากการตั้งค่า SNMP เรียบร้อยแล้วเท่าน้ัน) โดยผู้ใช้สามารถกดไปที่แท็บ “Home” จะแสดงเอฟเฟคการเคลื่อนไหวซึ่งแสดงให้เห็นว่าลิงก์ใดมีการไหลของโฟลว์เกิดขึ้น โดยผู้ใช้สามารถกดเข้าไปในลิงก์เพื่อดูข้อมูลโฟลว์ตามรูปที่ 3.8 และสามารถใช้ filter เพื่อกรองเฉพาะข้อมูลโฟลว์ที่ต้องการตามรูปที่ 3.9

Chart

Description automatically generated with low confidence

**รูปที่ 3.8** หน้าเว็บแสดงผลลิงก์ที่มีโฟลว์ไหลอยู่

A picture containing chart

Description automatically generated

**รูปที่ 3.9** ผู้ใช้สามารถกดที่ลิงก์เพื่อแสดงข้อมูลโฟลว์ และกรองโฟลว์เพื่อแสดงผลตามที่ผู้ใช้ต้องการได้

## 3.3 การตั้งค่าเริ่มต้นให้อุปกรณ์เครือข่ายผ่านการเรียกใช้คำสั่งจากโค้ดไพธอนโดยตรง

นอกจากการใช้งานผ่านหน้าเว็บแล้ว ผู้ใช้สามารถเรียกใช้ฟังก์ชันที่มีในหน้าเว็บผ่านทางเอพีไอโดยตรงได้เช่นกัน ซึ่งสามารถศึกษาวิธีเรียกใช้งานผ่านไฟล์ add\_device.py ที่อยู่ใน directory SDN-handmade\_v3/Fame\_test ไฟล์ประกอบไปด้วยฟังก์ชันต่างๆ และตัวอย่างการใช้งาน ผู้ใช้สามารถแก้ไขตัวแปร device\_list ตามชุดไอพีอุปกรณ์ที่มี และทดลองรันโปรแกรมเพื่อทดสอบใช้งานฟังก์ชันต่างๆ ที่มีในหน้าเว็บปกติได้

**บทที่ 4**

# ข้อมูลชุดคำสั่งเอพีไอที่สามารถใช้งานได้และตัวอย่างการใช้งาน

การใช้งาน API module ผู้ใช้จะต้องเข้าผ่านเว็บเบราเซอร์โดยมีรูปแบบ URL ดังนี้ **http://ControllerIp:5001/api/v1/Command/**

ตัวอย่าง: <http://10.50.34.15:5001/api/v1/device>

1. Controller Ip เป็นหมายเลขไอพีของเครื่องเซิร์ฟเวอร์ที่รันระบบคอนโทรลเลอร์อยู่ (ในแลปนี้ ใช้ไอพี 10.50.34.15)
2. Command เป็นกลุ่มชุดคำสั่งที่สามารถเรียกใช้ได้โดยตัวเลือกชุดคำสั่งมีดังต่อไปนี้
   1. device
   2. flow
   3. link
   4. path
   5. flow/routing

## 4.1 รายละเอียดคำสั่ง “device”

ผู้ใช้งานสามารถเรียกดูข้อมูลผ่านทาง http://10.50.34.15:5001/api/v1/device โดยหน้านี้

จะแสดงข้อมูลพื้นฐานอุปกรณ์ทั้งหมดในเครือข่าย เช่น ชื่ออุปกรณ์ ชื่อ รายละเอียดต่างๆที่เกี่ยวข้อง

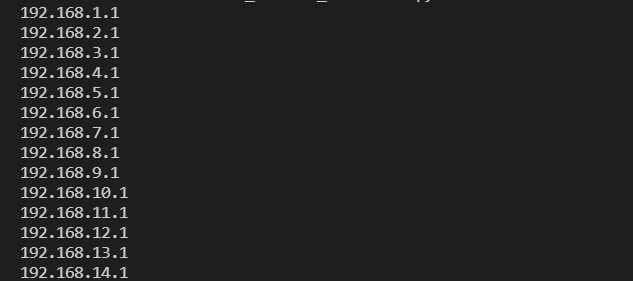
กับอินเตอร์เฟส ข้อมูล SSH และสถานการณ์ทำงานของ SNMP CDP และ NetFlow โดยผู้ใช้

สามารถเลือกดูข้อมูลอุปกรณ์ที่ต้องการแบบเฉพาะเจาะจงโดยการเพิ่ม /hostname ของอุปกรณ์ที่

ต้องการลงไปต่อท้ายชุดคำสั่งข้างต้นได้ ตัวอย่าง <http://10.50.34.15:5001/api/v1/device/R1>



**รูปที่ 4.1** ตัวอย่างการเรียกใช้ device เอพีไอ

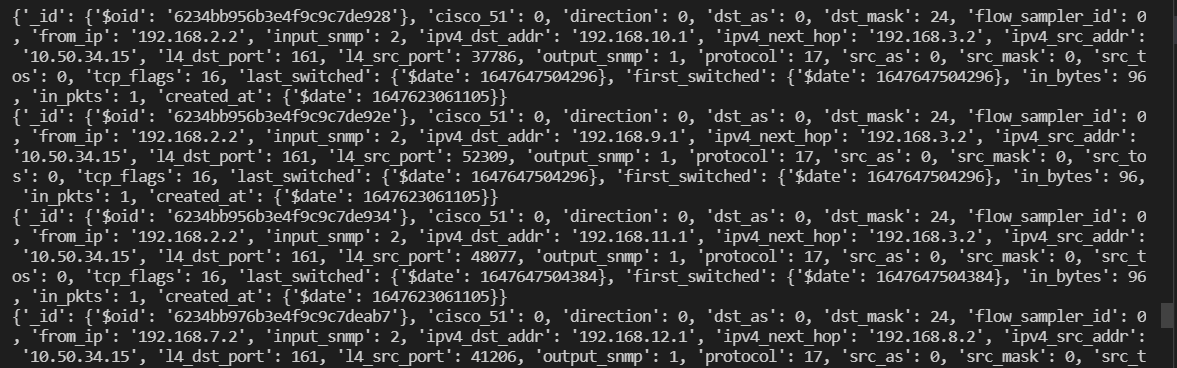
**รูปที่ 4.2** ผลลัพธ์จากการเรียกใช้โปรแกรมในรูปที่ 4.1

## 4.2 รายละเอียดคำสั่ง “flow”

ผู้ใช้งานสามารถเรียกดูข้อมูลผ่านทาง http://10.50.34.15:5001/api/v1/flow โดยหน้านี้จะ แสดงข้อมูลโฟลว์ทได้รับจากอุปกรณ์เครือข่าย เป็นรายละเอียดเกี่ยวกับข้อมูลที่มีการรับส่งผ่าน อุปกรณ์เครือข่าย โดยข้อมูลเหล่านี้สามารถนำมาประยุกต์ใช้ในการบังคับเปลี่ยนเส้นทางเพื่อทำ Network Traffic Engineering ได้

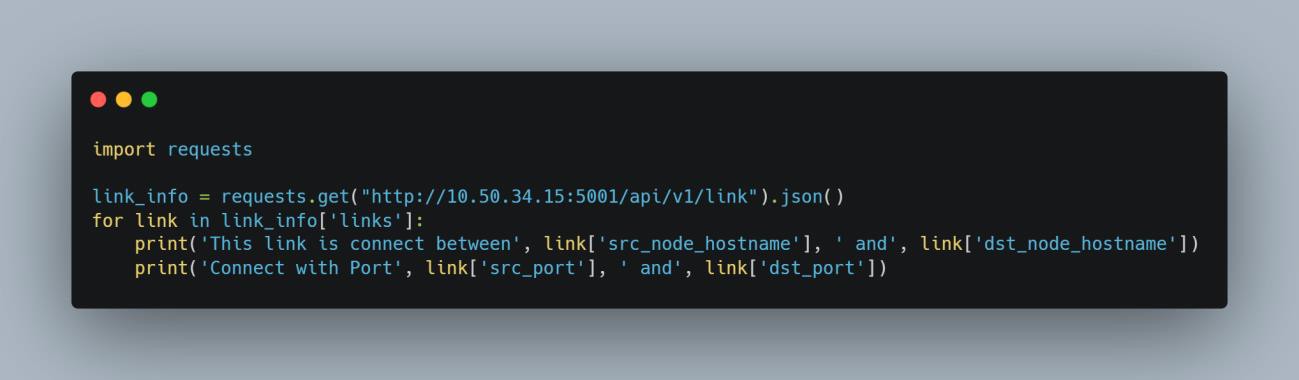


**รูปที่ 4.3** ตัวอย่างการเรียกใช้ flow เอพีไอ

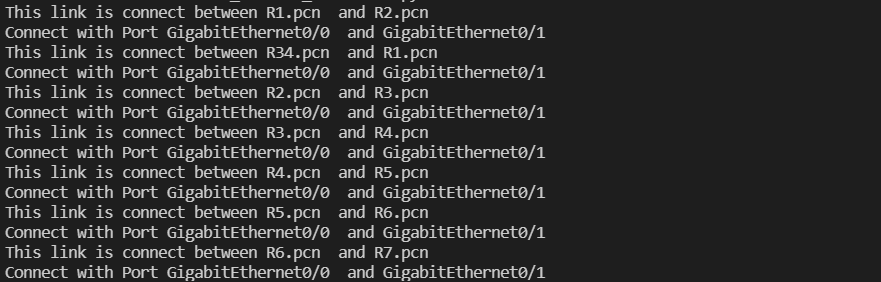
**รูปที่ 4.4** ผลลัพธ์จากการเรียกใช้โปรแกรมในรูปที่ 4.3

## 4.3 รายละเอียดคำสั่ง “link”

ผู้ใช้งานสามารถเรียกดูข้อมูลผ่านทาง http://10.50.34.15:5001/api/v1/link โดยหน้านี้จะ แสดงข้อมูลการเชื่อมต่อของอุปกรณ์ สามารถรู้ได้ว่าอุปกรณ์กำลังต่ออยู่กับอุปกรณ์อื่นใดบ้าง และสามารถแสดงข้อมูลปริมาณการใช้งานของแต่ละลิงก์ได้เช่นกัน



**รูปที่ 4.5** ตัวอย่างการเรียกใช้ link เอพีไอ

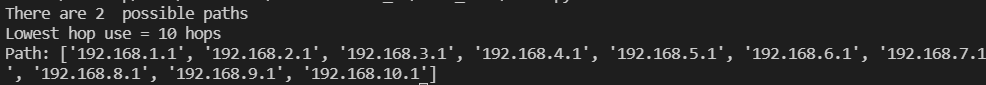


**รูปที่ 4.6** ผลลัพธ์จากการเรียกใช้โปรแกรมในรูปที่ 4.5

## 4.4 รายละเอียดคำสั่ง “path”

ผู้ใช้งานสามารถเรียกดูข้อมูลผ่านทาง http://10.50.34.15:5001/api/v1/path/srcip,dstip ตัวอย่าง http://10.50.34.15:5001/api/v1/path/192.168.1.1,192.168.10.1 โดยหน้านี้จะ สามารถแสดงข้อมูลเส้นทางที่เป็นไปได้ทั้งหมด จากไอพีต้นทาง และไอพีปลายทาง



**รูปที่ 4.7** ตัวอย่างการเรียกใช้ path เอพีไอ

**รูปที่ 4.8** ผลลัพธ์จากการเรียกใช้โปรแกรมในรูปที่ 4.7

## 4.5 รายละเอียดคำสั่ง “flow/routing”

ผู้ใช้สามารถจัดการ policy based routing สำหรับการตั้งค่าเส้นทางโดยเฉพาะตามที่ ต้องการได้ โดยผู้ใช้สามารถเรียกดู สร้าง และลบ policy ดังกล่าวผ่าน API นี้ได้โดย method GET POST DELETE ตามลำดับ ผ่าน URL http://10.50.34.15:5001/api/v1/flow/routing โดย พารามิเตอร์ที่ใช้สำหรับ DELETE และ POST method มีดังต่อไปนี้

**Delete method parameter สำหรับลบ policy routing.**

ผู้ใช้สามารถเรียกใช้ API โดยมี payload ดังต่อไปนี้

1. flow\_id: ไอดี policy ที่ต้องการลบ โดยผู้ใช้สามารถดูไอดีผ่าน GET method ของ flow/routing API



**รูปที่ 4.9** ตัวอย่าง Payload ของ DELETE Method

**POST method parameter สำหรับสร้าง policy routing.**

ผู้ใช้สามารถเรียกใช้ API โดยมี payload ดังต่อไปนี้

1. name: ชื่อ policy สามารถตั้งอะไรก็ได้
2. src\_ip/dst\_ip: ไอพีเครือข่ายต้นทาง และปลายทาง
3. src\_port/dst\_port: พอร์ตต้นทาง และปลายทาง หากเป็นพอร์ตอะไรก็ได้ให้ใส่ any
4. src\_subnet/dst\_subnet: ไวลด์การ์ดต้นทาง และปลายทาง
5. actions: กำหนดอุปกรณ์ที่จะตดตั้ง policy และวิธการรับมือของอุปกรณ์เหล่านั้นเมื่อมี Flow ตาม Policy ผ่านเข้ามา สามารถติดตั้งบนอุปกรณ์หลายตัวพร้อมกันได้โดยคำสั่งเดียว
   1. device\_id: ไอดีของอุปกรณ์ สามารถดูได้จาก API device
   2. action: วิธีการส่งข้อมูลตัวอย่างใช้ 2 หมายถึงส่งให้ Next Hop Ip (1 หมายถึง Next Hop Interface)
   3. data: ข้อมูล Next hop ต้องสอดคล้องกับ action ถ้า action เป็น 1 ให้ใส่ชื่อ Next Hop Interface เป็นต้น



**รูปที่ 4.10** ตัวอย่าง Payload ของ POST Method



**รูปที่ 4.11** ตัวอย่างการใช้ POST method สำหรับสร้าง policy routing ตัวอย่างหาก Router R1 ได้รับโฟลว์ท มีไอพีเครือข่ายต้นทางเป็น 192.168.200.0 ไอพีเครือข่ายปลายทางเป็น 192.168.201.0 พอร์ตต้น ทางและปลายทางเป็นอะไรก็ได้ จะส่ง Flow ดังกล่าวไปยัง Next Hop IP ที่เป็น 192.168.2.1

**รูปที่ 4.12** ตัวอย่างการใช้ DELETE method โดยการลบ policy ที่มีชื่อว่า Test\_policy โดยดึง flow\_id จาก GET method