ระบบจัดการเครือข่ายและควบคุมการจราจรบนเครือข่าย โดยใช้สถาปัตยกรรมแบบเอสดีเอ็น

NETWORK MANAGEMENT AND TRAFFIC CONTROL SYSTEM USING SDN ARCHITECTURE

โดย

กฤษฎา อิงอาน

KRISSADA INGARN

สถาพร แดงน้อย

SATHAPON DANGNOI

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.สุเมธ ประภาวัต

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2560

ระบบจัดการเครือข่ายและควบคุมการจราจรบนเครือข่าย โดยใช้สถาปัตยกรรมแบบเอสดีเอ็น

โดย

กฤษฎา อิงอาน

สถาพร แดงน้อย

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.สุเมธ ประภาวัต

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2560

NETWORK MANAGEMENT AND TRAFFIC CONTROL SYSTEM

USING SDN ARCHITECTURE

KRISSADA INGARN

SATHAPON DANGNOI

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

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

2/2017

COPYRIGHT 2018

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT’S INSTITUTE OF TECHNOLOGY LADKRABANG

ใบรับรองปริญญานิพนธ์ประจำปีการศึกษา 2560

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจัดการเครือข่ายและควบคุมการจราจรบนเครือข่าย โดยใช้สถาปัตยกรรมแบบเอสดีเอ็น

NETWORK MANAGEMENT AND TRAFFIC CONTROL SYSTEM USING SDN ARCHITECTURE

ผู้จัดทำ

1. **นายกฤษฎา อิงอาน รหัสนักศึกษา 57070002**
2. **นายสถาพร แดงน้อย รหัสนักศึกษา 57070120**

………………………………………… อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.สุเมธ ประภาวัต)

ใบรับรองโครงงาน (PROJECT)

เรื่อง

ระบบจัดการเครือข่ายและควบคุมการจราจรบนเครือข่าย โดยใช้สถาปัตยกรรมแบบเอสดีเอ็น

NETWORK MANAGEMENT AND TRAFFIC CONTROL SYSTEM USING SDN ARCHITECTURE

นายกฤษฎา อิงอาน รหัสนักศึกษา 57070002

นายสถาพร แดงน้อย รหัสนักศึกษา 57070120

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด

รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ

การศึกษาวิชาโครงงาน หลักสูตรวิทยาศาสตรบัณฑิต (เทคโนโลยีสารสนเทศ)

ภาคเรียนที่ 2 ปีการศึกษา 2560

…………………………………………

(นายกฤษฎา อิงอาน)

…………………………………………

(นายสถาพร แดงน้อย)

**หัวข้อโครงการ** ระบบจัดการเครือข่ายและควบคุมการจราจรบนเครือข่าย โดยใช้สถาปัตยกรรมแบบเอสดีเอ็น

**นักศึกษา** นายกฤษฎา อิงอาน รหัสนักศึกษา 57070002

นายสถาพร แดงน้อย รหัสนักศึกษา 57070120

**ปริญญา** วิทยาศาสตรบัณฑิต

**สาขาวิชา** เทคโนโลยีสารสนเทศ

**ปีการศึกษา** 2560

**อาจารย์ที่ปรึกษา** ผู้ช่วยศาสตราจารย์ ดร.สุเมธ ประภาวัต

บทคัดย่อ

ในยุคที่เทคโนโลยีเครือข่ายมีการพัฒนาไปอย่างรวดเร็ว เพื่อรองรับปริมาณการใช้งานเครือข่ายที่เพิ่มสูงขึ้นอย่างต่อเนื่อง ซึ่งได้เกิดแนวคิดในการบริหารจัดการระบบเครือข่ายที่ชื่อว่า Software Define Network (SDN) ที่เป็นการนำซอฟต์แวร์เข้ามาควบคุมการทำงานของระบบเครือข่ายให้เป็นไปตามความต้องการ เพื่อช่วยเพิ่มความยืดหยุ่นและความสามารถในการควบคุมการทำงานของระบบเครือข่ายให้ดียิ่งขึ้น แต่อุปกรณ์เครือข่ายต้องสามารถรองรับการทำงานของ SDN ด้วย ซึ่งต้องมีการลงทุน ทางผู้จัดทำจึงได้พัฒนาระบบบริหารจัดการเครือข่ายที่มีการทำงานอ้างอิงตามแนวคิดของ SDN แต่เป็นการประยุกต์ให้สามารถทำงานบนอุปกรณ์เครือข่ายเดิมที่ไม่รองรับกับการทำงานนั้นให้ทำงานเสมือนกับใช้งานอุปกรณ์เครือข่ายที่รองรับการทำงานของแนวคิดดังกล่าว อีกทั้งยังมีการสร้างช่องทางการเชื่อมต่อสำหรับการเรียกใช้งานความสามารถของระบบเพื่อให้โปรแกรมประยุกต์สำหรับการจัดการเครือข่ายสามารถนำไปใช้ในการทำงานได้ และมีการพัฒนาโปรแกรมประยุกต์ต้นแบบสำหรับการจัดการจราจรภายในเครือข่ายที่นำหลักการของวิศวกรรมจราจรมาปรับใช้ในการควบคุมเส้นทางการจราจรเพื่อลดความคับคั่งบนเส้นทางหนึ่ง ๆ ที่อาจเกิดขึ้น

**Project Title** NETWORK MANAGEMENT AND TRAFFIC CONTROL SYSTEM USING SDN ARCHITECTURE

**Student** Mister Krissada ingarn Student ID 57070002

Mister Sathapon dangnoi Student ID 57070120

**Degree** Bachelor of Science

**Program** Information Technology

**Academic Year** 2017

**Advisor** Assistant Professor Dr. Sumet Prabhavat

ABSTRACT

Nowadays, network systems are scaled up to support a number of users. Most of the network systems discovery a route using dynamic routing protocol which able to find the best route. However, the performance of the network will dramatically decrease when the route is highly utilized and become congested. In this project, we develop a network management software based on Software Defined Network (SDN) concept. The system monitors the route utilization. If it congests, the system will switch to the route which is less congested. As a result, the Network System achieves better performance without requiring manual configuration from network administrator*.*

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สําเร็จลุล่วงไปได้ด้วยดี ด้วยความกรุณาของอาจารย์ที่ปรึกษา โครงงานผู้ช่วยศาสตราจารย์ ดร.สุเมธ ประภาวัต ที่คอยให้ความรู้ ให้คำแนะนำในการทำงานให้ คำปรึกษาเมื่อเกิดปัญหา และให้กำลังใจตลอดระยะเวลาการทำโครงงาน ขอขอบพระคุณอาจารย์เป็นอย่างสูง

ขอขอบพระคุณอาจารย์ในคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่คอยให้ความรู้ที่สามารถนำไปประยุกต์ใช้กับการทำโครงการได้

ขอขอบคุณ นายธนานพ ทองถาวร และนายวรวัชร ณรงคะชวนะ ที่คอยให้คำปรึกษาและคำแนะนำตลอดระยะเวลาในการทำโครงงาน

ขอขอบคุณครอบครัว และเพื่อนพี่น้องชาวไอทีลาดกระบังทุกคนที่คอยช่วยเหลือ ให้การ สนับสนุนและให้กำลังใจเสมอมา

กฤษฎา อิงอาน

สถาพร แดงน้อย

สารบัญ

หน้า

[บทคัดย่อ I](#_Toc515520407)

[ABSTRACT II](#_Toc515520408)

[กิตติกรรมประกาศ III](#_Toc515520409)

[สารบัญ IV](#_Toc515520410)

[สารบัญรูป VII](#_Toc515520411)

[สารบัญตาราง X](#_Toc515520414)

[บทที่ 1 บทนำ 1](#_Toc515520416)

[1.1 ความเป็นมาและความสำคัญของปัญหา 1](#_Toc515520417)

[1.2 ความมุ่งหมายและวัตถุประสงค์ 1](#_Toc515520418)

[1.3 ขอบเขตงานวิจัย 2](#_Toc515520419)

[1.4 ขั้นตอนการพัฒนาโครงการ 2](#_Toc515520420)

[1.5 ประโยชน์ที่คาดว่าจะได้รับ 2](#_Toc515520421)

[บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 3](#_Toc515520422)

[2.1 Software Define Network (SDN) 3](#_Toc515520423)

[2.2 Traffic Engineering 5](#_Toc515520424)

[2.3 การตรวจสอบสถานะของระบบเครือข่าย 6](#_Toc515520425)

[2.3.1 NetFlow 6](#_Toc515520426)

[2.3.2 Simple Network Management Protocol (SNMP) 6](#_Toc515520427)

[2.3.3 Cisco Discovery Protocol (CDP) 8](#_Toc515520428)

[2.4 การควบคุมการทำงานของอุปกรณ์เครือข่าย 9](#_Toc515520429)

[2.4.1 IP-based Routing 9](#_Toc515520430)

[2.4.2 Policy-based Routing 9](#_Toc515520431)

[2.5 Remote access 10](#_Toc515520432)

[บทที่ 3 แนวคิดและการดำเนินงาน 11](#_Toc515520433)

[3.1 สถาปัตยกรรมของระบบ 11](#_Toc515520434)

[3.2 การออกแบบระบบ 12](#_Toc515520435)

[3.2.1 ส่วนติดต่อผู้ใช้งาน 12](#_Toc515520436)

[3.2.2 การตรวจสอบและเก็บสถานะของระบบเครือข่าย 17](#_Toc515520437)

[3.2.5 ฐานข้อมูลภายในระบบ 18](#_Toc515520438)

**สารบัญ (ต่อ)**

หน้า

[3.2.3 การบริหารจัดการจราจรเครือข่าย 24](#_Toc515520439)

[3.2.4 สร้างชุดคำสั่งควบคุมการทำงานของอุปกรณ์เครือข่าย 26](#_Toc515520440)

[3.2.6 กระบวนการทำงานของระบบ 27](#_Toc515520441)

[3.3 แผนภาพ Use-case 34](#_Toc515520442)

[3.3.1 แผนภาพ Use-case Diagram 34](#_Toc515520443)

[3.3.2 แผนภาพ Use-case Description 35](#_Toc515520444)

[บทที่ 4 การทดลอง และการประเมินผล 41](#_Toc515520445)

[4.1 การทดลองการใช้งานผ่าน Command Line Interface 41](#_Toc515520446)

[4.1.1 การเพิ่มอุปกรณ์ 41](#_Toc515520447)

[4.1.2 การแสดงข้อมูลการเชื่อมต่อกันของอุปกรณ์ 42](#_Toc515520448)

[4.1.3 การแสดงข้อมูลอุปกรณ์เครือข่าย 42](#_Toc515520449)

[4.1.4 การแสดงข้อมูลอินเตอร์เฟสของอุปกรณ์เครือข่าย 42](#_Toc515520450)

[4.1.5 การแสดงข้อมูลเส้นทางของอุปกรณ์เครือข่าย 43](#_Toc515520451)

[4.2 การทดลองการใช้งานระบบผ่าน Web UI 43](#_Toc515520452)

[4.2.1 การเพิ่มอุปกรณ์ 43](#_Toc515520453)

[4.2.2 การแสดงข้อมูลอุปกรณ์ 44](#_Toc515520454)

[4.2.3 การสร้างการควบคุมเส้นทางของโฟลว์ 46](#_Toc515520455)

[4.2.4 การแสดงรายการควบคุมเส้นทางของโฟลว์ 46](#_Toc515520456)

[4.2.4 แสดงข้อมูลการใช้งานของโฟลว์ 47](#_Toc515520457)

[4.3 การทดลองระบบเปลี่ยนเส้นทางแบบอัตโนมัติ 47](#_Toc515520458)

[4.3.1 การทดลองรูปแบบที่ 1 47](#_Toc515520459)

[4.3.2 การทดลองรูปแบบที่ 2 50](#_Toc515520460)

[บทที่ 5 วิเคราะห์และสรุปผล 54](#_Toc515520461)

[5.1 สรุปผลการดำเนินงาน 54](#_Toc515520462)

[5.2 ปัญหาและอุปสรรคที่พบในการพัฒนาระบบ 54](#_Toc515520463)

[5.3 ข้อจำกัดของระบบ 54](#_Toc515520464)

[5.4 ข้อเสนอแนะและแนวทางในการพัฒนาระบบในอนาคต 55](#_Toc515520465)

[บรรณานุกรม 56](#_Toc515520466)

[ภาคผนวก 58](#_Toc515520467)

สารบัญ (ต่อ)

หน้า

[ภาคผนวก ก วิธีการใช้งานระบบ 59](#_Toc515520468)

[ภาคผนวก ข วิธีการใช้งาน REST API 64](#_Toc515520470)

[ประวัติผู้เขียน 86](#_Toc515520472)

สารบัญรูป

**รูปที่ หน้า**

[2.1 แสดงลักษณะโครงสร้างการทำงานของอุปกรณ์เครือข่าย 3](#_Toc515228230)

[2.2 แสดงแนวคิดการทำงานของ SDN 4](#_Toc515228231)

[2.3 แสดงสถาปัตยกรรมของ SDN 5](#_Toc515228232)

[2.4 แสดงการกระจายเส้นทางการส่งของข้อมูลในระบบเครือข่าย 5](#_Toc515228233)

[2.5 แสดงการทำงานของ NetFlow 6](#_Toc515228234)

[2.6 แสดงโครงสร้างของ SNMP 7](#_Toc515228235)

[2.7 แสดงตัวอย่างลำดับชั้นการจัดเก็บข้อมูลของฐานข้อมูล MIB 8](#_Toc515228236)

[2.8 แสดงตัวอย่างข้อมูลที่ได้รับจากโพรโทคอล CDP 9](#_Toc515228237)

[2.9 แสดงตัวอย่างคำสั่ง route map 10](#_Toc515228238)

[2.10 แสดงตัวอย่างการใช้งาน SSH 10](#_Toc515228239)

[3.1 แสดงแผนผังภาพรวมของระบบ 11](#_Toc515228222)

[3.2 แสดงตัวอย่าง Command Line Interface 13](#_Toc515228223)

[3.5 แสดงหน้าจอรายละเอียดของอุปกรณ์เครือข่ายผ่านเว็บไซต์ 14](#_Toc515228224)

[3.6 แสดงหน้าจอการแก้ไขข้อมูลอุปกรณ์เครือข่ายผ่านเว็บไซต์ 15](#_Toc515228225)

[3.7 แสดงหน้าจอบังคับเปลี่ยนเส้นทางของอุปกรณ์เครือข่ายแบบกำหนดเองผ่านเว็บไซต์ 15](#_Toc515228226)

[3.8 แสดงหน้าจอข้อมูลของโฟลว์ภายในระบบเครือข่ายผ่านเว็บไซต์ 16](#_Toc515228227)

[3.9 แสดงหน้าจอข้อมูลการบังคับเปลี่ยนเส้นทางของระบบผ่านเว็บไซต์ 16](#_Toc515228228)

[3.18 แสดงแผนภาพ Use-case Diagram 34](#_Toc515228229)

[3. 1 แสดงแผนผังภาพรวมของระบบ 11](#_Toc515233088)

[3. 2 แสดงตัวอย่าง Command Line Interface 13](#_Toc515233089)

[3. 3 แสดงหน้าจอแผนภาพการเชื่อมต่อของระบบเครือข่ายผ่านเว็บไซต์ 13](#_Toc515233090)

[3. 4 แสดงหน้าจอการเพิ่มข้อมูลอุปกรณ์เครือข่ายผ่านเว็บไซต์ 14](#_Toc515233091)

[3. 5 แสดงหน้าจอรายละเอียดของอุปกรณ์เครือข่ายผ่านเว็บไซต์ 14](#_Toc515233092)

[3. 6 แสดงหน้าจอการแก้ไขข้อมูลอุปกรณ์เครือข่ายผ่านเว็บไซต์ 15](#_Toc515233093)

[3. 7 แสดงหน้าจอบังคับเปลี่ยนเส้นทางของอุปกรณ์เครือข่ายแบบกำหนดเองผ่านเว็บไซต์ 15](#_Toc515233094)

[3. 8 แสดงหน้าจอข้อมูลของโฟลว์ภายในระบบเครือข่ายผ่านเว็บไซต์ 16](#_Toc515233095)

[3. 9 แสดงหน้าจอข้อมูลการบังคับเปลี่ยนเส้นทางของระบบผ่านเว็บไซต์ 16](#_Toc515233096)

[3. 10 แสดงตัวอย่างข้อมูลภายในตาราง Device Table 19](#_Toc515233097)

สารบัญรูป (ต่อ)

**รูปที่ หน้า**

[3. 11 แสดงตัวอย่างข้อมูลภายในตาราง Device Relationship Table 20](#_Toc515233098)

[3. 12 แสดงตัวอย่างข้อมูลภายในตาราง Device Routing Table 20](#_Toc515233099)

[3. 13 แสดงตัวอย่างข้อมูลภายในตาราง Flow Stat Table 21](#_Toc515233100)

[3. 14 แสดงตัวอย่างข้อมูลภายในตาราง Flow Routing Table 22](#_Toc515233101)

[3. 15 แสดงตัวอย่างข้อมูลภายในตาราง Link Utilization Table 23](#_Toc515233102)

[3. 16 แสดงตัวอย่างข้อมูลภายในตาราง Used Flow ID List Table 23](#_Toc515233103)

[3. 17 แสดงแนวคิดการทำงานของ Global First Fit 24](#_Toc515233104)

[3. 18 แสดงกระบวนการทำงานโดยรวมของระบบ 27](#_Toc515233105)

[3. 19 แสดงกระบวนการเลือกเส้นทางใหม่ 31](#_Toc515233106)

[3. 20 แสดงกระบวนการลบโฟลว์การบังคับเปลี่ยนเส้นทางกรณีที่ 1 28](#_Toc515233107)

[3. 21 แสดงกระบวนการลบโฟลว์การบังคับเปลี่ยนเส้นทางกรณีที่ 2 29](#_Toc515233108)

[4.1 หน้าจอแสดงการเพิ่มอุปกรณ์ผ่านทาง Command Line Interface 41](#_Toc515228200)

[4.2 หน้าจอแสดงการเชื่อมต่อของอุปกรณ์ 42](#_Toc515228201)

[4.3 หน้าจอแสดงข้อมูลอุปกรณ์ 42](#_Toc515228202)

[4.4 หน้าจอแสดงข้อมูลอินเตอร์เฟสของอุปกรณ์ 42](#_Toc515228203)

[4.5 หน้าจอแสดงข้อมูลเส้นทางของอุปกรณ์ 43](#_Toc515228204)

[4.6 เมนูการเพิ่มอุปกรณ์ 43](#_Toc515228205)

[4.7 หน้าจอแสดงการเพิ่มอุปกรณ์ 44](#_Toc515228206)

[4.8 หน้าจอแสดงเมื่อเพิ่มอุปกรณ์สำเร็จ 44](#_Toc515228207)

[4.9 หน้าจอแสดงข้อมูลอุปกรณ์ 45](#_Toc515228208)

[4.10 หน้าจอแสดงการสร้างการควบคุมเส้นทางของโฟลว์ 46](#_Toc515228209)

[4.11 แสดงรายการควบคุมเส้นทางของโฟลว์ 46](#_Toc515228210)

[4.12 หน้าแสดงข้อมูลการใช้งานของโฟลว์ 47](#_Toc515228211)

[4.13 แสดงแผนภาพการเชื่อมต่อของเครือข่ายการทดลองที่ 1 48](#_Toc515228212)

[4.14 เส้นทางการส่งข้อมูลของการทดลองที่ 1 ก่อนใช้งานระบบ 48](#_Toc515228213)

[4.15 กราฟแสดงการใช้งานของลิงค์ระหว่างเราส์เตอร์ก่อนใช้งานระบบ 49](#_Toc515228214)

[4.16 กราฟแสดงการใช้งานของลิงค์ระหว่างเราส์เตอร์หลังใช้งานระบบ 49](#_Toc515228215)

[4.17 เส้นทางการส่งข้อมูลขณะที่ระบบได้ย้ายเส้นทาง 50](#_Toc515228216)

สารบัญรูป (ต่อ)

**รูปที่ หน้า**

[4.18 แสดงแผนภาพการเชื่อมต่อของเครือข่ายการทดลองที่ 2 50](#_Toc515228217)

[4.19 เส้นทางการส่งข้อมูลของการทดลองที่ 1 51](#_Toc515228218)

[4.20 กราฟแสดงการใช้งานของลิงค์ระหว่างเราส์เตอร์ก่อนใช้งานระบบ 51](#_Toc515228219)

[4.21 กราฟแสดงการใช้งานของลิงค์ระหว่างเราส์เตอร์หลังใช้งานระบบ 52](#_Toc515228220)

[4.22 เส้นทางการส่งข้อมูลขณะที่ระบบได้ย้ายเส้นทาง 52](#_Toc515228221)

[ก.1แสดงตัวอย่างการตั้งค่า Database 60](#_Toc515495918)

[ก.2 แสดงการรันไฟล์ main.py 60](#_Toc515495919)

[ก.3 แสดงการรันไฟล์ main\_web.py 60](#_Toc515495920)

[ก.4 แสดงการตั้งค่า baseURL 61](#_Toc515495921)

[ก.5 แสดงการรันระบบส่วนของ Frontend 61](#_Toc515495922)

[ก.6 แสดงโครงสร้างของไฟล์สำหรับการเพิ่มอุปกรณ์ชนิดใหม่เข้าสู่ระบบ 61](#_Toc515495923)

[ก.7 แสดงการสร้างฟังก์ชันสำหรับการเพิ่มอุปกรณ์ชนิดใหม่เข้าสู่ระบบ 62](#_Toc515495924)

[ก.8 แสดงข้อมูลในฟังก์ชัน generate\_config\_command 63](#_Toc515495925)

[ก.9 แสดงข้อมูลในฟังก์ชัน generate\_remove\_command 63](#_Toc515495926)

[ก.10 แสดงข้อมูลในฟังก์ชัน get\_netmiko\_type 63](#_Toc515495927)

สารบัญตาราง

ตารางที่ หน้า

[ข.1 แสดงรายชื่อของ API ทั้งหมด 65](#_Toc515539835)

[ข.2 ตัวอย่างข้อมูลที่คืนค่าข้อมูลอุปกรณ์ทั้งหมด 66](#_Toc515539836)

[ข.3 ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อเพิ่มอุปกรณ์ 68](#_Toc515539837)

[ข.4 ตัวอย่างข้อมูลที่คืนค่าของการเพิ่มอุปกรณ์ 69](#_Toc515539838)

[ข.5 ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อแก้ไขอุปกรณ์ 70](#_Toc515539839)

[ข.6 ตัวอย่างข้อมูลที่คืนค่าของการแก้ไขอุปกรณ์ 70](#_Toc515539840)

[ข.7 ตัวอย่างการคืนค่าข้อมูลของการขอข้อมูลอุปกรณ์ตามไอดีของอุปกรณ์ 71](#_Toc515539841)

[ข.8 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลอุปกรณ์ที่เชื่อมต่อกัน 73](#_Toc515539842)

[ข.9 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลลิงค์ทั้งหมด 75](#_Toc515539843)

[ข.10 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลลิงค์ตามไอดี 76](#_Toc515539844)

[ข.11 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่ง 77](#_Toc515539845)

[ข.12 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลโฟลว์ทั้งหมด 77](#_Toc515539846)

[ข.13 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นทางของโฟลว์ 79](#_Toc515539847)

[ข.14 ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อสร้างเส้นทางของโฟลว์ 81](#_Toc515539848)

[ข.15 ตัวอย่างข้อมูลที่คืนค่าของการสร้างเส้นทางของโฟลว์ 82](#_Toc515539849)

[ข.16 ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อแก้ไขเส้นทางของโฟลว์ 82](#_Toc515539850)

[ข.17 ตัวอย่างข้อมูลที่คืนค่าของการแก้ไขข้อมูลเส้นทางของโฟลว์ 83](#_Toc515539851)

[ข.18 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นทางของโฟลว์ 83](#_Toc515539852)

[ข.19 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลตารางเส้นทางของอุปกรณ์ 84](#_Toc515539853)

# บทที่ 1 บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

อันเนื่องมาจากการขยายตัวของระบบเครือข่ายที่เพิ่มสูงขึ้น ทำให้ในปัจจุบันระบบเครือข่ายมีขนาดที่ใหญ่และมีความซับซ้อนมากยิ่งขึ้น ซึ่งเมื่อเป็นเช่นนั้นก็เป็นไปได้ยากที่ผู้ดูแลระบบจะสามารถบริหารจัดการระบบเครือข่ายให้สามารถทำงานได้อย่างเต็มประสิทธิภาพ เพราะว่าเมื่อมีปริมาณของข้อมูลที่รับส่งภายในเครือข่ายเป็นจำนวนมากและใช้กระบวนการหาเส้นทางเป็นแบบอัตโนมัติ ซึ่งจะเลือกเส้นทางในการรับส่งข้อมูลจากเส้นทางที่ดีที่สุดแล้วนั้น ก็มีความเป็นไปได้ที่อาจจะเกิดการคับคั่งของข้อมูลในเส้นทางหนึ่งๆที่มีการใช้งานเป็นจำนวนมาก ซึ่งเมื่อเป็นเช่นนั้นก็อาจจะส่งผลตามมาคือ การรับส่งข้อมูลอาจเกิดความล่าช้าหรือข้อมูลอาจจะสูญหายในกรณีที่อุปกรณ์เครือข่ายไม่สามารถรองรับปริมาณของข้อมูลที่มากเกินไปได้ ผู้จัดทำจึงได้มีแนวคิดที่จะสร้างระบบขึ้นมาเพื่อใช้ในการบริหารจัดการในส่วนนี้ โดยระบบที่พัฒนาขึ้นนั้นจะทำงานควบคู่ไปกับการทำงานตามปกติของระบบเครือข่ายที่ใช้การหาเส้นทางแบบอัตโนมัติและจะคอยทำการตรวจสอบหาเส้นทางการรับส่งข้อมูลภายในระบบเครือข่ายที่มีความคับคั่งแล้วทำการย้ายเส้นทางการส่งของข้อมูลที่ใช้เส้นทางนั้นอยู่บางส่วนไปใช้เส้นทางอื่น เพื่อช่วยลดความคับคั่งของข้อมูลในเส้นทางนั้น ๆ และทำให้การรับส่งข้อมูลภายในระบบเครือข่ายมีประสิทธิภาพมากยิ่งขึ้นอีกด้วย โดยระบบที่พัฒนาขึ้นนั้นจะอ้างอิงมาจากแนวคิดของเอสดีเอ็น ซึ่งจะเน้นไปที่การพัฒนาเพื่อให้อุปกรณ์เครือข่ายที่ไม่รองรับการทำงานของเทคโนโลยีเอสดีเอ็นให้มีความสามารถในการทำงานที่เทียบเคียงกับอุปกรณ์ที่รองรับ เพื่อช่วยเพิ่มประสิทธิภาพการทำงานของระบบให้เพิ่มสูงขึ้น โดยที่ไม่จำเป็นต้องลงทุนซื้ออุปกรณ์เครือข่ายใหม่

## 1.2 ความมุ่งหมายและวัตถุประสงค์

1. เพื่อศึกษาแนวทางการพัฒนาระบบจัดการเครือข่ายโดยการประยุกต์หลักการแบบเอสดีเอ็น
2. เพื่อศึกษาแนวทางการควบคุมการจราจรบนเครือข่ายเพื่อการกระจายภาระงานบนเครือข่าย
3. เพื่อพัฒนาระบบแบบจำลองสำหรับการจัดการเครือข่ายและการควบคุมการจราจรบนเครือข่าย

## 1.3 ขอบเขตงานวิจัย

ต้นแบบนี้ถูกพัฒนาสำหรับอุปกรณ์เครือข่ายที่รองรับการตั้งค่าแบบ Command Line Interface (CLI) เท่านั้น โดยเบื้องต้นจะใช้ได้กับอุปกรณ์ Cisco ที่มีอยู่ในห้องปฏิบัติการ 306 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## 1.4 ขั้นตอนการพัฒนาโครงการ

1. ศึกษาเทคโนโลยีการบริหารจัดการเครือข่ายแบบเอสดีเอ็น
2. ศึกษาวิศวกรรมจราจรบนเครือข่ายสำหรับควบคุมการกระจายภาระงาน
3. ศึกษาเทคนิคและกลไกที่จำเป็นสำหรับการตรวจสอบสถานะของเครือข่าย
4. ศึกษาเทคนิคและกลไกที่จำเป็นสำหรับการควบคุมการทำงานของอุปกรณ์เครือข่าย
5. ศึกษาเทคนิคและกลไกที่จำเป็นสำหรับการจัดการแบบจำลองเครือข่ายผ่านเว็บบราวเซอร์
6. พัฒนาระบบจัดการเครือข่ายและควบคุมการกระจายภาระงานบนเครือข่าย
7. ทดสอบ วิเคราะห์ และประเมินผล
8. สรุปผล

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบบริหารจัดการเครือข่ายที่สามารถควบคุมการจราจรและการกระจายภาระงานบนเครือข่าย
2. ระบบบริหารแบบจำลองจัดการเครือข่ายที่ช่วยให้การทำงานของผู้ดูแลระบบง่ายขึ้น
3. ระบบบริหารจัดการเครือข่ายที่ช่วยเพิ่มประสิทธิภาพการทำงานเครือข่ายได้โดยไม่ต้องเปลี่ยนแปลงอุปกรณ์

# บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

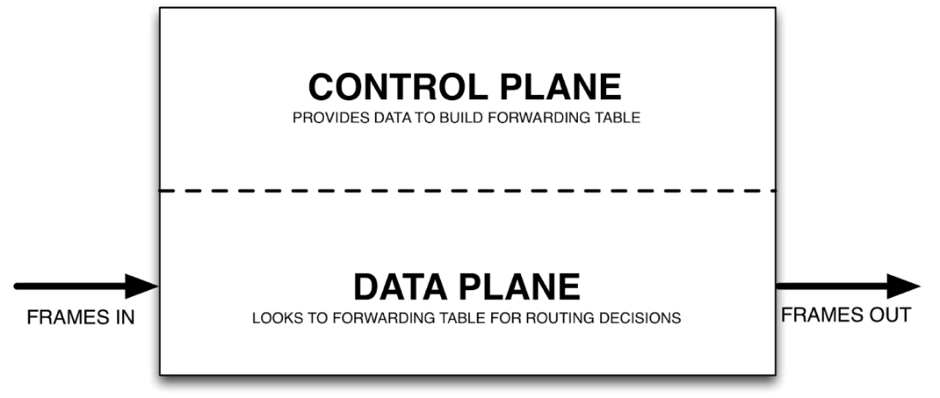
ในบทนี้จะกล่าวถึงทฤษฎีและเทคโนโลยีที่มีความเกี่ยวข้องกับโครงการ ซึ่งผู้พัฒนาได้ศึกษาเพื่อนำมาใช้เป็นแนวทางในการพัฒนาและดำเนินโครงการ โดยมีรายละเอียดดังนี้

## 2.1 Software Define Network (SDN)

SDN [1-3] หรือเอสดีเอ็นเป็นแนวคิดในการบริหารจัดการระบบเครือข่าย โดยใช้ซอฟต์แวร์เข้ามาควบคุมการทำงานของระบบเครือข่ายและรวมศูนย์การควบคุมทั้งหมดมาไว้ที่ซอฟต์แวร์เพียงที่เดียว เพื่อทำให้ง่ายต่อการบริหารจัดการระบบเครือข่าย รวมไปถึงลดความซับซ้อนของระบบเครือข่ายที่เกิดจากการประมวลผลการทำงานของอุปกรณ์เครือข่ายแต่ละตัวเพื่อทำงานร่วมกันลง

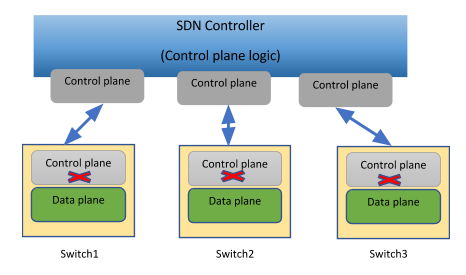
ซึ่งแนวคิดนี้มีความต้องการที่จะเปลี่ยนลักษณะการทำงานของอุปกรณ์เครือข่ายให้แตกต่างจากเดิม โดยที่ปกติในอุปกรณ์เครือข่ายแต่ละตัวจะมีลักษณะการทำงานแบ่งเป็น 2 ส่วน คือ

1. Control Plane ที่เป็นส่วนของการประมวลผลสร้างตารางเส้นทางของระบบเครือข่าย เช่น การประมวลผลเพื่อหาเส้นทางแบบอัตโนมัติ ซึ่งจะต้องมีการแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์เครือข่ายและนำข้อมูลดังกล่าวมาประมวลผลเพื่อสร้างเป็นตารางเส้นทางของระบบเครือข่าย
2. Data Plane หรือ Forwarding Plane ทำหน้าที่ในการรับและส่งต่อข้อมูลออกจากอุปกรณ์เครือข่าย ซึ่งการส่งต่อข้อมูลจะเปรียบเทียบจากตารางเส้นทางของระบบเครือข่ายเพื่อหาทิศทางในการส่งข้อมูลไปยังปลายทางและส่งต่อข้อมูลออกทางอินเตอร์เฟสของอุปกรณ์เครือข่ายตามทิศทางนั้น ๆ



รูปที่ 2.1 แสดงลักษณะโครงสร้างการทำงานของอุปกรณ์เครือข่าย

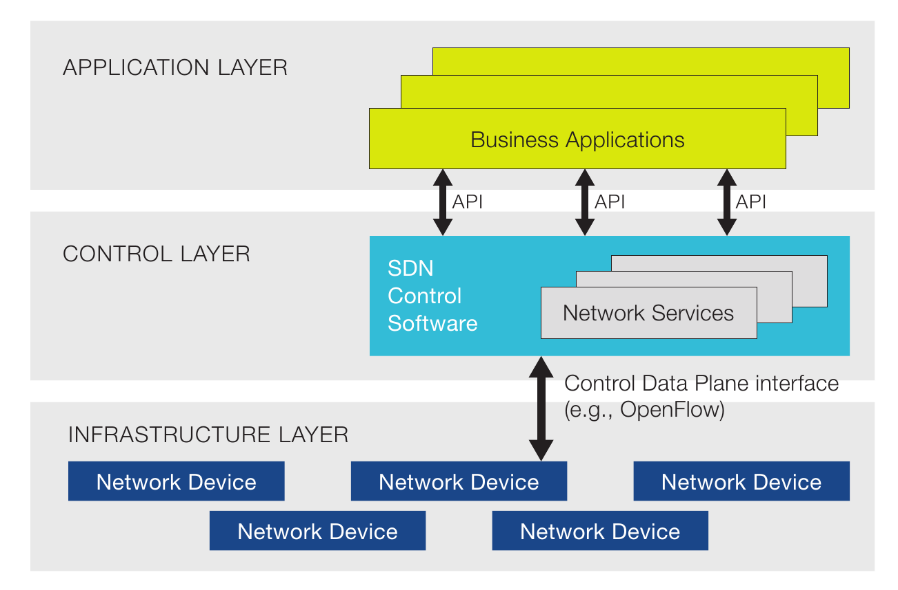
โดยที่แนวคิดของเอสดีเอ็น คือ การย้ายส่วนของ control plane ที่ทำหน้าที่ในการประมวลผลสร้างตารางเส้นทางของอุปกรณ์เครือข่ายมาไว้ที่ซอฟต์แวร์เพื่อให้ซอฟต์แวร์เป็นผู้ประมวลผลการทำงานต่าง ๆ และส่งค่าการทำงานไปให้ยังอุปกรณ์เครือข่าย ส่วนอุปกรณ์เครือข่ายที่มีเพียงแค่ส่วนของ data plane นั้นจะทำหน้าที่เพียงแค่การรับและส่งต่อข้อมูลออกจากอุปกรณ์เครือข่ายตามทิศทางที่ซอฟต์แวร์เป็นผู้กำหนดให้เท่านั้น ซึ่งจากแนวคิดดังกล่าวจะส่งผลให้ซอฟต์แวร์สามารถควบคุมการทำงานของระบบเครือข่ายทั้งหมดได้ รวมไปถึงสามารถเปลี่ยนแปลงการทำงานของระบบเครือข่ายให้เป็นไปตามต้องการได้อย่างรวดเร็ว



รูปที่ 2.2 แสดงแนวคิดการทำงานของ SDN

เอสดีเอ็นมีโครงสร้างของสถาปัตยกรรมแบ่งออกเป็น 3 ลำดับชั้นตามลักษณะการทำงาน โดยที่รายละเอียดแต่ละลำดับชั้น มีดังนี้

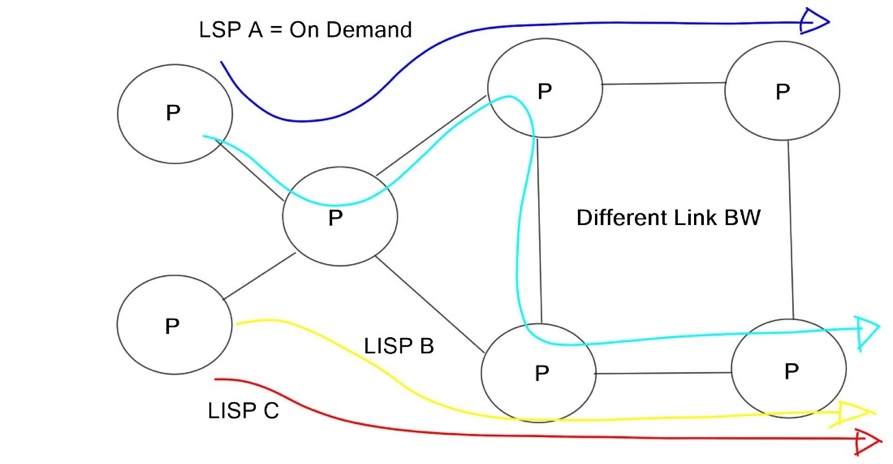
* Application Layer: เป็นส่วนของโปรแกรมประยุกต์ที่ทำงานติดต่อกับผู้ใช้งาน โดยทำหน้าที่ในการรับคำสั่งและแสดงผลการทำงานต่าง ๆ ให้กับผู้ใช้งาน ซึ่งการทำงานนั้นอาจแตกต่างกันออกไปตามโปรแกรมประยุกต์ที่ใช้งานและความต้องการในการใช้งาน
* Control Layer: เป็นส่วนของซอฟต์แวร์ที่ทำหน้าที่เป็นตัวกลางในการทำงานระหว่างโปรแกรมประยุกต์กับระบบเครือข่าย โดยจะทำหน้าที่เป็นศูนย์กลางที่เชื่อมต่อกับอุปกรณ์เครือข่ายทั้งหมดและมีการจัดเก็บค่าสถานะการทำงานต่าง ๆ ของอุปกรณ์เครือข่ายเอาไว้เพื่อให้โปรแกรมประยุกต์นำไปใช้งาน อีกทั้งยังทำหน้าที่ในการส่งคำสั่งเพื่อควบคุมการทำงานของอุปกรณ์เครือข่ายให้ทำงานตามที่ผู้ใช้งานต้องการอีกด้วย
* Infrastructure Layer: คืออุปกรณ์เครือข่ายที่มีหน้าที่ทำงานตามคำสั่งของ Control layer และมีการส่งข้อมูลค่าสถานะทำงานที่เกิดขึ้นไปให้ยัง Control layer อีกด้วย



รูปที่ 2.3 แสดงสถาปัตยกรรมของ SDN

## 2.2 Traffic Engineering

Traffic Engineering [4-6] คือ แนวคิดในการบริหารจัดการการรับส่งข้อมูลภายในระบบเครือข่ายให้มีประสิทธิภาพการทำงานที่สูงขึ้นและสอดคล้องกับความต้องการในการใช้งานระบบเครือข่ายมากยิ่งขึ้น โดยการนำข้อมูลปริมาณการใช้งานของระบบเครือข่ายทั้งในอดีตและปัจจุบันมาวิเคราะห์เพื่อคาดการณ์ปริมาณการใช้งานที่จะเกิดขึ้นในอนาคตและหาแนวทางในการบริหารจัดการเส้นทางในการรับส่งข้อมูลของข้อมูลจำนวนนั้นเพื่อให้การรับส่งข้อมูลภายในระบบเครือข่ายมีประสิทธิภาพมากตรงตามที่ต้องการ ซึ่งในการทำวิศวกรรมจราจรนั้นอาจจะแตกต่างกันไปตามความต้องการใช้งานของระบบเครือข่ายนั้น ๆ เช่น ต้องการการรับส่งข้อมูลที่มีความล่าช้าน้อย ความเร็วในการรับส่งที่สูง เป็นต้น ซึ่งขึ้นอยู่กับความต้องการใช้งานและประเภทของข้อมูลที่มีการรับส่งภายในระบบเครือข่ายด้วย แต่ในส่วนของซอฟต์แวร์ที่พัฒนาขึ้นนั้น ผู้พัฒนามีจุดประสงค์เพื่อลดความคับคั่งของข้อมูลที่อาจจะเกิดขึ้นในเส้นทางหนึ่ง ๆ ซึ่งอาจจะส่งผลทำให้การรับส่งข้อมูลภายในระบบเครือข่ายมีความล่าช้าและอาจทำให้เกิดการสูญหายของข้อมูล โดยการกระจายการรับส่งข้อมูลออกไปยังเส้นทางต่าง ๆ ทำให้ความคับคั่งในเส้นทางหนึ่ง ๆ ลดลง

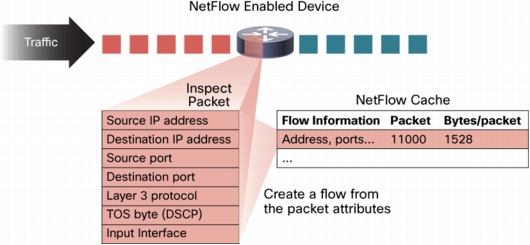


รูปที่ 2.4 แสดงการกระจายเส้นทางการส่งของข้อมูลในระบบเครือข่าย

## 2.3 การตรวจสอบสถานะของระบบเครือข่าย

### 2.3.1 NetFlow

NetFlow [7] เป็นเครื่องมือที่ถูกพัฒนาขึ้นโดยบริษัทซิสโก้ (CISCO) เพื่อใช้ในการเก็บรวบรวมข้อมูลการจราจรบนระบบเครือข่าย โดยข้อมูลที่จัดเก็บจะอยู่ในรูปแบบโฟลว์ข้อมูล (traffic flow) ซึ่งภายในแต่ละโฟลว์จะประกอบไปด้วยข้อมูลที่บอกถึงลักษณะของข้อมูลชุดนั้น ๆ เช่น หมายเลขแอดเดรสต้นทาง หมายเลขแอดเดรสปลายทาง หมายเลขพอร์ตต้นทาง หมายเลขพอร์ตปลายทาง ปริมาณของข้อมูล เป็นต้น ซึ่งข้อมูลที่มีลักษณะตามที่กล่าวมาข้างต้นเหมือนกันทั้งหมดจะถือว่าเป็นโฟลว์ข้อมูลเดียวกันและถูกจัดเก็บอยู่ภายในฐานข้อมูลของ NetFlow ที่มีชื่อว่า NetFlow Cache

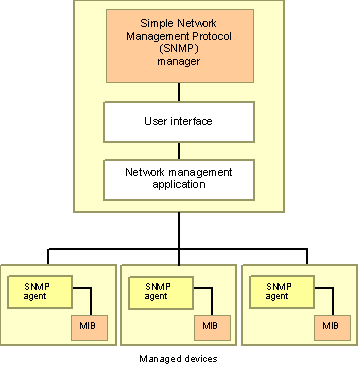


รูปที่ 2.5 แสดงการทำงานของ NetFlow

การเข้าถึงโฟลว์ข้อมูลที่จัดเก็บไว้ทำได้ 2 วิธี คือ เข้าถึงฐานข้อมูลโดยตรงจากอุปกรณ์เครือข่ายผ่านทาง CLI หรือตั้งค่าให้อุปกรณ์เครือข่ายส่งออก (export) โฟลว์ข้อมูลออกไปยังเครื่องปลายทางที่กำหนด ซึ่งการส่งออกข้อมูลจะส่งทุก ๆ ช่วงเวลาที่กำหนด โดยโฟลว์ข้อมูลที่ถูกส่งออกไปนั้นจะเป็นโฟลว์ข้อมูลที่สิ้นสุดการส่งข้อมูลไปแล้วหรือเป็นโฟลว์ข้อมูลที่ใช้เวลาในการรับส่งข้อมูลเกินกว่าที่กำหนด (ค่าพื้นฐานคือ 30 นาที) เท่านั้น ซึ่งรูปแบบของโฟลว์ข้อมูลที่มีการส่งออกนั้นมีหลายรุ่น (version) ซึ่งลักษณะของข้อมูลที่ส่งออกนั้นก็อาจจะแตกต่างกันไปตามแต่ละรุ่น โดยค่ามาตรฐาน คือ รุ่นที่ 5

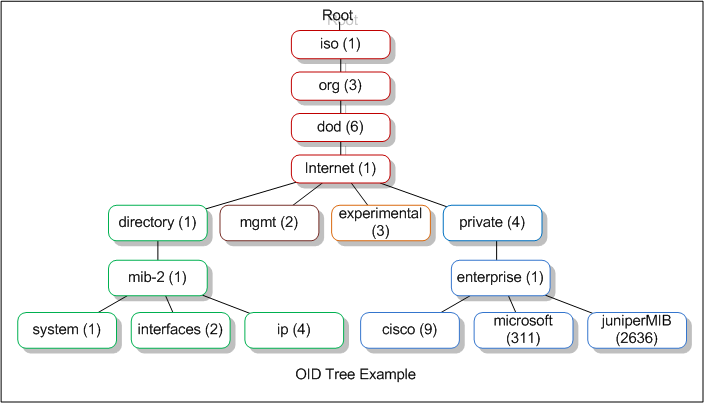
### 2.3.2 Simple Network Management Protocol (SNMP)

Simple Network Management Protocol [8-9] เป็นเครื่องมือที่มีความสามารถในการตรวจสอบสถานะและข้อมูลต่าง ๆ ของระบบเครือข่าย ซึ่งการทำงานของ SNMP นั้นจะแบ่งอุปกรณ์เครือข่ายออกเป็น 2 ประเภท คือ เครื่องที่ใช้บริหารจัดการ (Manager) และเครื่องที่ถูกบริหารจัดการ (Agent) โดยที่ภายในเครื่องที่ถูกบริหารจัดการนั้นจะมีฐานข้อมูลที่ชื่อว่า Management Information Base (MIB) ซึ่งจะจัดเก็บข้อมูลต่าง ๆของอุปกรณ์เครือข่ายเอาไว้ และในส่วนนี้เองเครื่องที่ใช้ในการบริหารจัดการนั้นสามารถส่งคำสั่งเพื่อร้องขอข้อมูลหรือเปลี่ยนแปลงค่าการทำงานบางอย่างของอุปกรณ์เครือข่ายได้ ซึ่งในการตรวจสอบข้อมูลและสถานะการทำงานสามารถทำได้ 2 รูปแบบคือ อุปกรณ์ที่มีหน้าที่บริหารจัดการส่งคำสั่งเพื่อร้องขอข้อมูลจากอุปกรณ์เครือข่ายเครื่องอื่น ๆ เมื่ออุปกรณ์เครือข่ายได้รับคำร้องขอก็จะส่งข้อมูลที่ต้องการกลับมาให้ หรือกำหนดเงื่อนไขให้อุปกรณ์เครือข่ายส่งข้อมูลมาให้เครื่องที่มีหน้าที่บริหารจัดการเมื่อมีการกระทำตรงตามเงื่อนไขที่กำหนด



รูปที่ 2.6 แสดงโครงสร้างของ SNMP

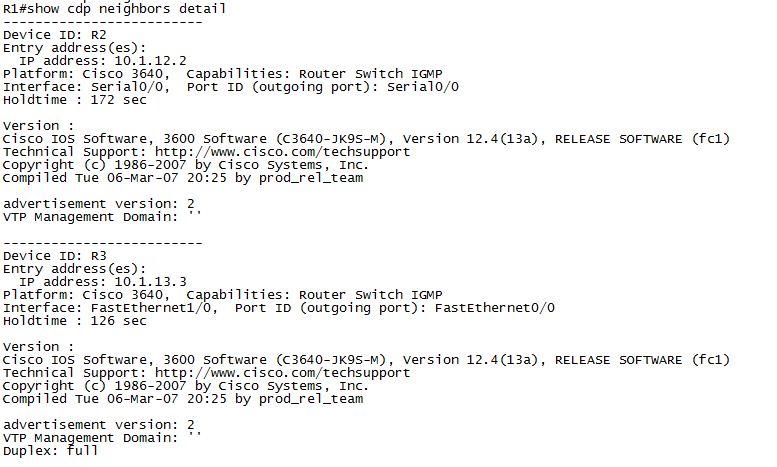
ภายในฐานข้อมูล Management Information Base นั้นจะจัดเก็บข้อมูลในรูปของวัตถุ (Object) ที่มีหมายเลขกำกับแต่ละวัตถุ เรียกว่า Object ID (OID) ซึ่งการจัดเก็บจะมีลักษณะเป็นลำดับชั้นคล้ายต้นไม้ เพื่อให้ง่ายต่อการใช้งานและการจัดการ



รูปที่ 2.7 แสดงตัวอย่างลำดับชั้นการจัดเก็บข้อมูลของฐานข้อมูล MIB

### 2.3.3 Cisco Discovery Protocol (CDP)

Cisco Discovery Protocol [10-12] เป็นโพรโทคอลเฉพาะของบริษัทซิสโก้ ใช้สำหรับการค้นหาข้อมูลพื้นฐานของอุปกรณ์เครือข่ายที่เชื่อมต่อกัน โดยที่อุปกรณ์เครือข่ายแต่ละเครื่องจะประกาศชุดของข้อมูลที่ประกอบไปด้วยข้อมูลพื้นฐานต่าง ๆ ของตัวเองออกไป เช่น ระบบปฏิบัติการ ประเภท รุ่น ชื่อของอุปกรณ์เครือข่าย หมายเลขแอดเดรสของอินเตอร์เฟส เป็นต้น ซึ่งข้อมูลที่ประกาศอาจจะมีความแตกต่างกันตามรุ่นของระบบปฏิบัติการของอุปกรณ์เครือข่าย โดยจะประกาศข้อมูลออกไปทุก ๆ ช่วงเวลา (ค่ามาตรฐานคือ 60 วินาที) ออกไปแบบกลุ่ม (multicast) ปลายทางอยู่ที่หมายเลขแอดเดรส 01:00:0C:CC:CC:CC ซึ่งอุปกรณ์เครือข่ายเครื่องอื่น ๆ นั้นรอรับชุดข้อมูลดังกล่าวอยู่แล้ว เมื่อได้รับชุดข้อมูลเหล่านี้ก็จะทำการจัดเก็บข้อมูลลงในฐานข้อมูล ซึ่งข้อมูลดังกล่าวจะมีการปรับเปลี่ยนทุกครั้งที่ได้รับข้อมูลเข้ามาใหม่ ทำให้อุปกรณ์เครือข่ายแต่ละเครื่องมีข้อมูลพื้นฐานของอุปกรณ์เครือข่ายที่เชื่อมต่อกันทั้งหมด เพื่อใช้ในการสร้างการเชื่อมต่อไปยังอุปกรณ์เครื่องอื่น ๆ และในกรณีที่อุปกรณ์เครือข่ายไม่ได้รับชุดข้อมูลที่ประกาศจากอุปกรณ์เครือข่ายอื่นครบ 3 ครั้งข้อมูลของอุปกรณ์เครือข่ายเครื่องนั้นจะถูกลบออกโดยทันที



รูปที่ 2.8 แสดงตัวอย่างข้อมูลที่ได้รับจากโพรโทคอล CDP

## 2.4 การควบคุมการทำงานของอุปกรณ์เครือข่าย

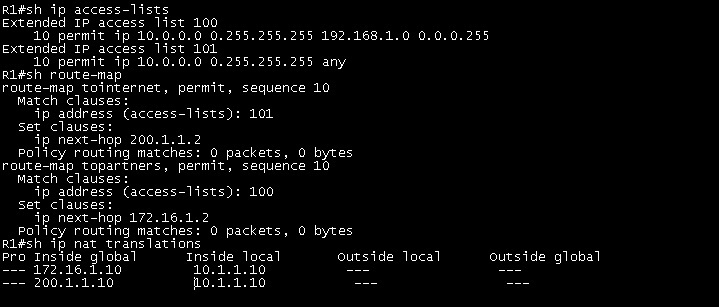
### 2.4.1 IP-based Routing

เป็นรูปแบบในการรับส่งข้อมูลภายในระบบเครือข่ายที่ใช้หมายเลขแอดเดรสของปลายทางในการกำหนดทิศทางการรับส่งข้อมูล ซึ่งมีการใช้งานกันอย่างแพร่หลายในปัจจุบัน รวมไปถึงการนำไปใช้ควบคู่กับกระบวนการหาเส้นทางแบบอัตโนมัติที่มีกระบวนการในการแลกเปลี่ยนข้อมูลซึ่งกันและกันเพื่อสร้างเป็นตารางเส้นทางสำหรับการรับส่งข้อมูลภายในระบบเครือข่าย เช่น โพรโทคอล RIP [13], OSPF [14] เป็นต้น โดยการเลือกเส้นทางที่ใช้ในการรับส่งข้อมูลจะเลือกเส้นทางที่ดีที่สุดจากการเปรียบเทียบค่า Cost ของแต่ละเส้นทางและเลือกเส้นทางที่มีค่า Cost ดีที่สุด ซึ่งค่า Cost อาจจะแตกต่างกันออกไปตามแต่ละโพรโทคอล เช่น จำนวนของอุปกรณ์เครือข่ายที่ข้อมูลต้องผ่านจนกว่าจะถึงปลายทาง, Bandwidth ของเส้นทาง, Delay ของการรับส่งข้อมูล เป็นต้น

### 2.4.2 Policy-based Routing

Policy-based Routing [15-17] เป็นรูปแบบการกำหนดเส้นทางการรับส่งข้อมูลภายในระบบเครือข่าย โดยใช้นโยบาย (Policy) ในการกำหนดทิศทางการรับส่งของข้อมูลแทนการใช้หมายเลขแอดเดรสปลายทางเป็นตัวกำหนด ซึ่งนโยบายจะประกอบไปด้วย 2 ส่วน คือ ลักษณะหรือคุณสมบัติของข้อมูล เช่น หมายเลขแอดเดรสต้นทาง หมายเลขแอดเดรสปลายทาง หมายเลขพอร์ต เป็นต้น และการดำเนินการต่อข้อมูลที่มีลักษณะตรงตามที่ระบุไว้ข้างต้น เช่น ส่งข้อมูลออกอินเตอร์เฟสใดของอุปกรณ์เครือข่าย ซึ่งในการทำงานเมื่ออุปกรณ์เครือข่ายได้รับข้อมูลเข้ามาจะตรวจสอบกับนโยบายที่มีการกำหนดไว้ ถ้าข้อมูลที่เข้ามามีลักษณะตรงตามที่ระบุก็จะดำเนินการกับข้อมูลตามที่ระบุในนโยบาย ซึ่งซอฟต์แวร์ที่พัฒนาใช้ Policy-based Routing เฉพาะของบริษัท Cisco ที่ชื่อว่า route map

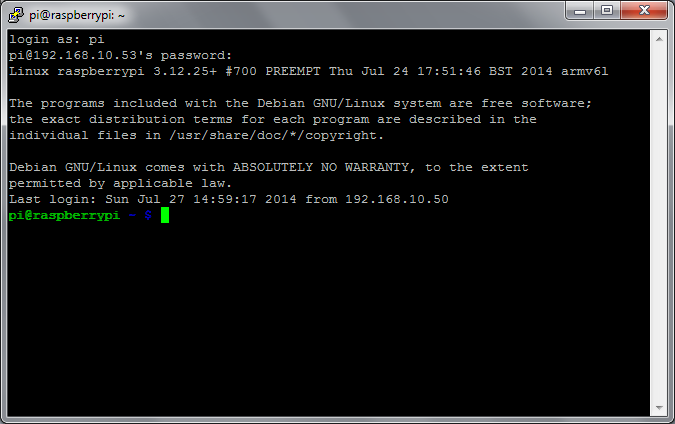
Route map [18] มีลักษณะการทำงานแบ่งเป็น 2 ส่วน คือ ส่วนของการกำหนดเงื่อนไขในการตรวจสอบข้อมูล ซึ่งเงื่อนไขที่สามารถนำมาใช้ในการตรวจสอบ เช่น หมายเลขแอดเดรสต้นทาง หมายเลขแอดเดรสปลายทาง หมายเลขพอร์ต โพรโทคอล เป็นต้น โดยที่การกำหนดเงื่อนไขนั้นสามารถกำหนดให้ตรวจสอบเงื่อนไขจาก Access Control List (ACL) ได้ด้วย อีกส่วนหนึ่ง คือ การกำหนดการกระทำต่อข้อมูลที่มีลักษณะตรงตามเงื่อนไข เช่น ไม่ส่งต่อข้อมูลออกจากอุปกรณ์เครือข่ายหรือส่งออกข้อมูลออกทางอินเตอร์เฟสของอุปกรณ์เครือข่ายที่กำหนด เป็นต้น



รูปที่ 2.9 แสดงตัวอย่างคำสั่ง route map

## 2.5 Remote access

Remote access [19] เป็นเทคโนโลยีที่ใช้สำหรับการเข้าถึงอุปกรณ์จากระยะไกล โดยสามารถควบคุมอุปกรณ์นั้น ๆ ได้เหมือนกับการควบคุมที่อุปกรณ์โดยตรง โดยในการทำงานอุปกรณ์ที่ต้องการเข้าถึงจากระยะไกล (remote) และอุปกรณ์ที่ใช้ในการสร้างการเชื่อมต่อไป (local) จะต้องติดตั้งโปรแกรมที่ใช้สำหรับการเข้าถึงระยะไกล โดยที่โปรแกรมของทั้ง 2 ฝั่งจะติดต่อกันผ่านโพรโทคอลตามที่โปรแกรมนั้น ๆ กำหนด โดยโพรโทคอลที่นำมาใช้ในการทำงานของซอฟต์แวร์นั้นมีชื่อว่า Secure Shell (SSH) เนื่องจากโพรโทคอล SSH นั้นมีการเข้ารหัสข้อมูลที่มีการรับส่งกันผ่านระบบเครือข่าย ทำให้มีความปลอดภัยในการใช้งานมากกว่าโพรโทคอลอื่น

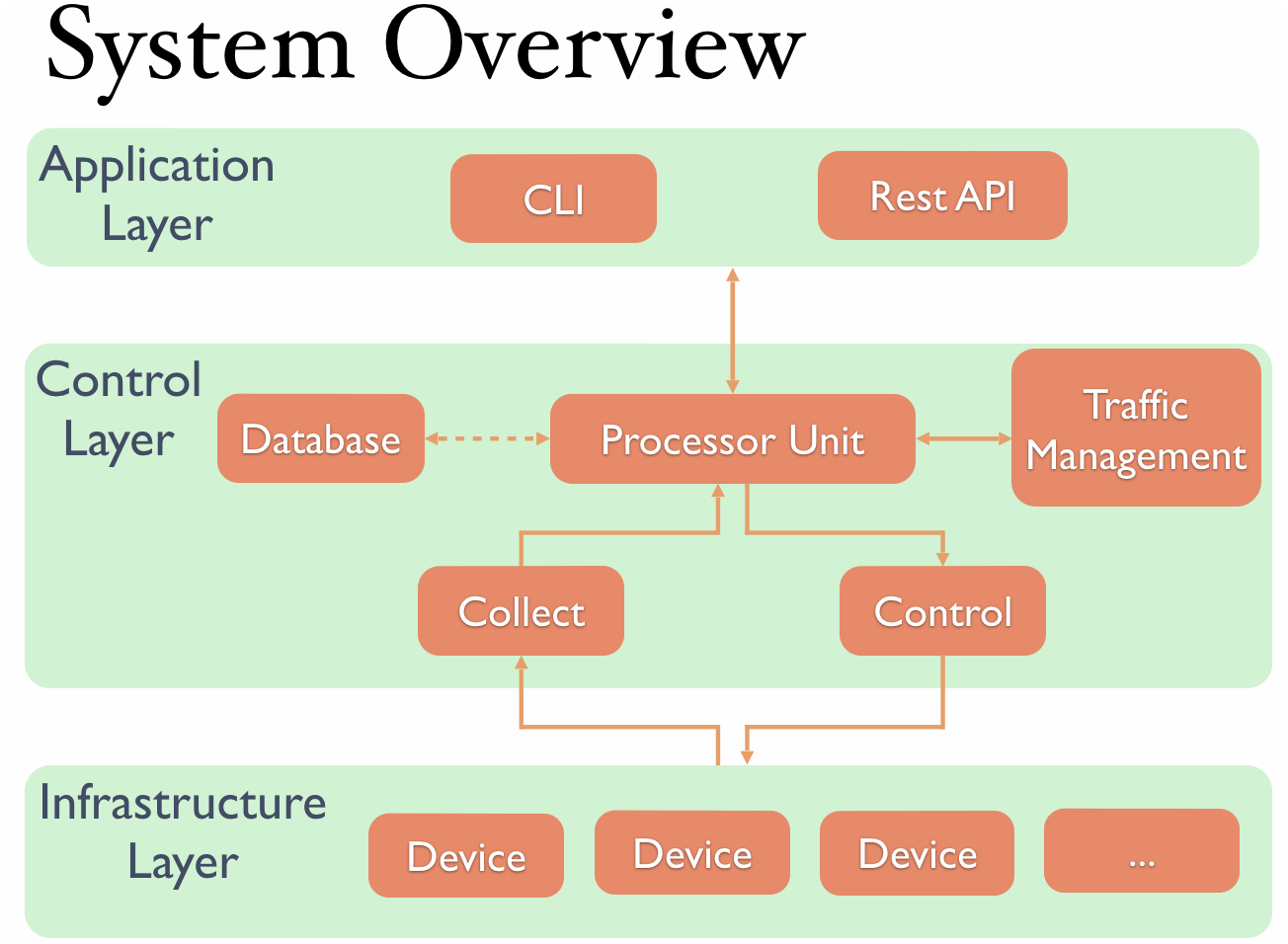


รูปที่ 2.10 แสดงตัวอย่างการใช้งาน SSH

# บทที่ 3 แนวคิดและการดำเนินงาน

ในบทนี้จะกล่าวถึงสถาปัตยกรรมและการออกแบบของระบบที่ผู้พัฒนาได้นำเสนอมาข้างต้น

## 3.1 สถาปัตยกรรมของระบบ



รูปที่ 3.1 แสดงแผนผังภาพรวมของระบบ

สถาปัตยกรรมของระบบแบ่งออกเป็น 3 ระดับชั้นตามลักษณะการทำงานและอ้างอิงตามสถาปัตยกรรมของเอสดีเอ็น ซึ่งระบบที่พัฒนาขึ้นจะเน้นไปที่การพัฒนาในชั้น Control Layer เพื่อให้รองรับกับการพัฒนาโปรแกรมประยุกต์สำหรับการบริหารจัดการระบบเครือข่ายโดยที่ไม่ต้องพัฒนาให้โปรแกรมประยุกต์ไปติดต่อสั่งการอุปกรณ์เครือข่ายเองและมีการพัฒนาโปรแกรมประยุกต์สำหรับการบริหารจัดการการจราจรภายในระบบเครือข่าย เพื่อเป็นโปรแกรมประยุกต์ต้นแบบสำหรับการบริหารจัดการระบบเครือข่าย ซึ่งในแต่ละระดับชั้นของสถาปัตยกรรมระบบมีรายละเอียดดังนี้

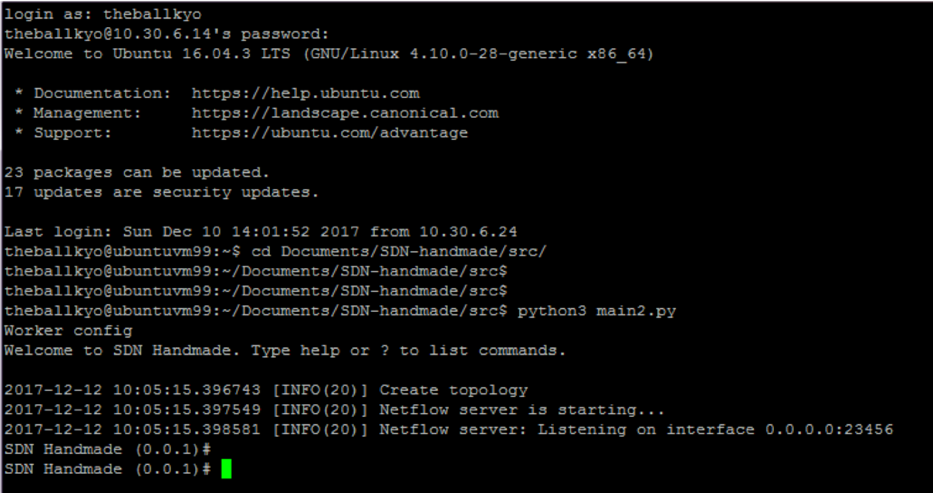
* **Application Layer**: เป็นส่วนของโปรแกรมประยุกต์ที่ทำหน้าที่ในการบริหารจัดการระบบเครือข่ายและเป็นส่วนติดต่อกับผู้ใช้งาน โดยกระบวนการทำงานจะทำการติดต่อกับชั้น Control เพื่อเรียกดูข้อมูลต่าง ๆ ของระบบเครือข่ายและควบคุมการทำงานอุปกรณ์เครือข่ายให้เป็นไปตามที่ต้องการ ซึ่งในการติดต่อกันระหว่างชั้นนั้นจะติดต่อกันผ่านทาง API ที่ชั้น Control เปิดเอาไว้ให้ใช้งาน ทำให้สามารถเรียกดูข้อมูลและควบคุมการทำงานของระบบเครือข่ายได้เลย ไม่จำเป็นต้องไปร้องขอข้อมูลจากอุปกรณ์เครือข่ายเอง โดยที่โปรแกรมประยุกต์สำหรับการบริหารจัดการระบบเครือข่ายที่พัฒนาขึ้นมีส่วนติดต่อผู้ใช้งานอยู่ 2 รูปแบบคือ Command Line Interface (CLI) และ Graphical User Interface (GUI) ซึ่ง GUI จะทำงานผ่าน Rest API อีกทีหนึ่ง
* **Control Layer**: เป็นตัวกลางการทำงานระหว่างโปรแกรมประยุกต์และอุปกรณ์ภายในระบบเครือข่าย โดยจะทำหน้าที่เป็นศูนย์กลางในการเชื่อมต่อกับอุปกรณ์เครือข่ายทั้งหมดและทำการจัดเก็บข้อมูลจากอุปกรณ์ภายในระบบเครือข่ายแล้วนำมาจัดเก็บเอาไว้ในฐานข้อมูล รวมไปถึงทำหน้าที่ในการส่งคำสั่งเพื่อควบคุมการทำงานของอุปกรณ์เครือข่ายให้เป็นไปตามที่โปรแกรมประยุกต์สั่งการ ซึ่งในการติดต่อกับโปรแกรมประยุกต์นั้นจะติดต่อผ่านทาง API ที่ได้มีการเปิดเอาไว้สำหรับให้เรียกใช้งาน และการติดต่อกับอุปกรณ์เครือข่ายนั้นจะใช้ช่องทางการเชื่อมต่อพิเศษที่สร้างเอาไว้สำหรับการติดต่อกับอุปกรณ์เครือข่ายแต่ละเครื่องโดยตรง
* **Infrastructure Layer**: เป็นส่วนของอุปกรณ์เครือข่ายที่ถูกบริหารจัดการ ซึ่งระบบที่พัฒนาขึ้นมีจุดประสงค์เพื่อใช้งานร่วมกับอุปกรณ์เครือข่ายที่ไม่รองรับการทำงานของเอสดีเอ็น ทำให้ในการทำงานอุปกรณ์เครือข่ายจะไม่มีความสามารถในการส่งข้อมูลหรือรับคำสั่งจากชั้น Control โดยตรง จึงต้องมีการใช้งานกระบวนการต่าง ๆ ในการจัดเก็บข้อมูลและควบคุมการทำงานของอุปกรณ์เครือข่าย ซึ่งกระบวนการที่ใช้อาจขึ้นอยู่กับอุปกรณ์เครือข่ายที่แตกต่างกัน

## 3.2 การออกแบบระบบ

### 3.2.1 ส่วนติดต่อผู้ใช้งาน

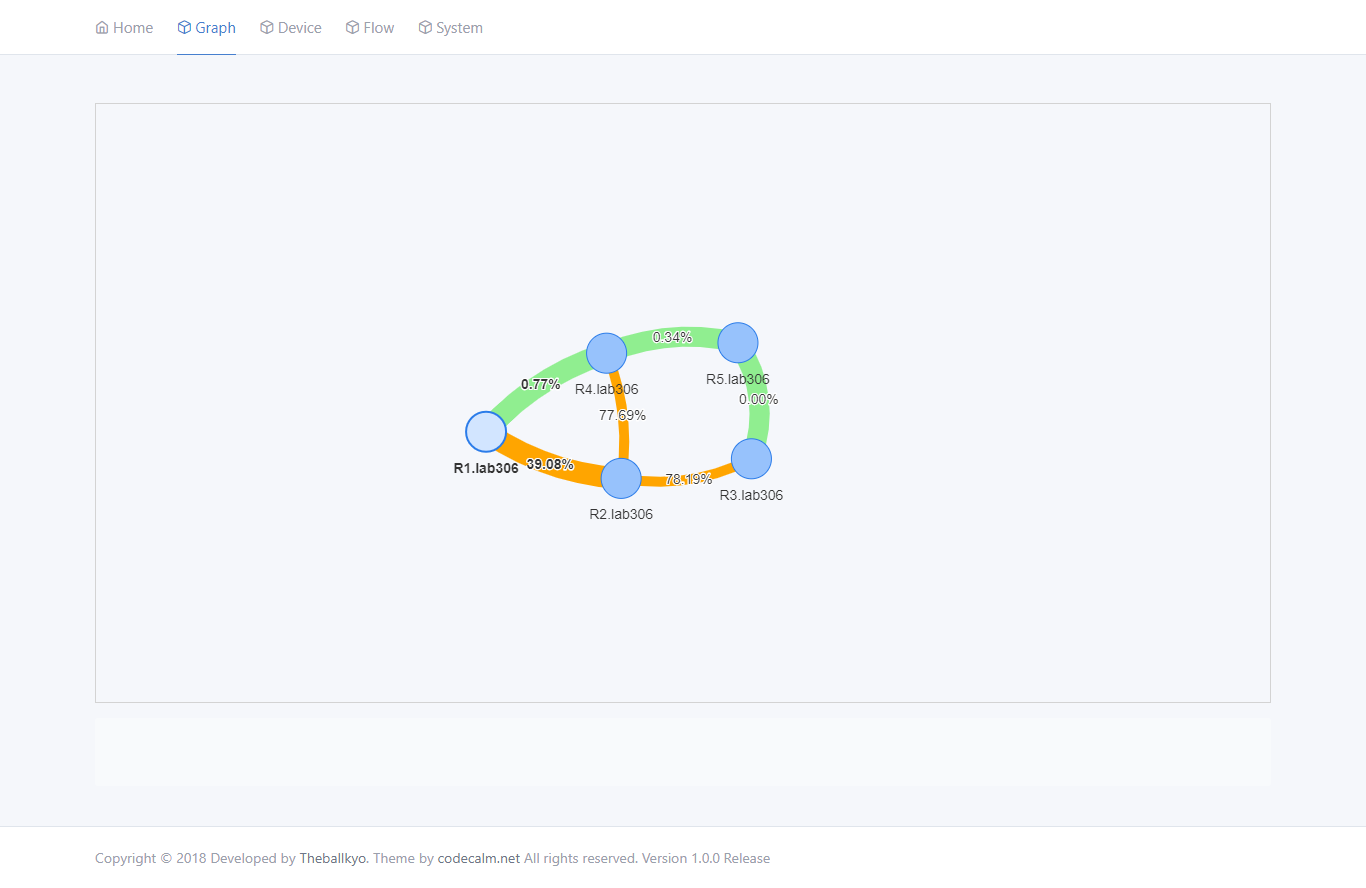
ทำหน้าที่ในการรับคำสั่งและแสดงผลการทำงานต่าง ๆ ให้แก่ผู้ใช้งาน ซึ่งในการทำงานส่วนติดต่อผู้ใช้งานจะเรียกใช้งาน API ต่าง ๆ ของชั้น Control ในการทำงาน ทั้งการเรียกดูข้อมูลของระบบเครือข่ายและการสั่งการควบคุมการทำงานของระบบเครือข่าย การทำงานทั้งหมดจะทำงานผ่านชั้น Control ไม่จำเป็นต้องติดต่อกับอุปกรณ์เครือข่ายเองโดยตรง โดยที่ระบบจัดการจราจรเครือข่ายที่พัฒนาขึ้นมีรูปแบบของส่วนติดต่อผู้ใช้งานอยู่ 2 รูปแบบ คือ

1. Command Line Interface (CLI): เป็นรูปแบบการสั่งการและแสดงผลการทำงานในรูปแบบของตัวอักษร โดยการใช้งานจะสั่งการระบบผ่านการพิมพ์คำสั่งที่มีลักษณะโครงสร้างตามที่กำหนด ซึ่งคำสั่งที่ใช้ภายในระบบจะอ้างอิงมาจากรูปแบบคำสั่งของอุปกรณ์เครือข่ายของ Cisco ทำให้ผู้ที่คุ้นเคยกับการใช้งานอุปกรณ์เครือข่ายของ Cisco สามารถทำความเข้าใจคำสั่งที่ใช้ภายในระบบได้ไม่ยาก อีกทั้งระบบยังมีเครื่องมือสำหรับช่วยเหลือในการค้นหาคำสั่ง คือ พิมพ์คำสั่ง help เพื่อเรียกดูคำสั่งที่สามารถใช้งานได้ ซึ่งเป็นความสามารถในลักษณะเดียวกับอุปกรณ์เครือข่ายของ Cisco

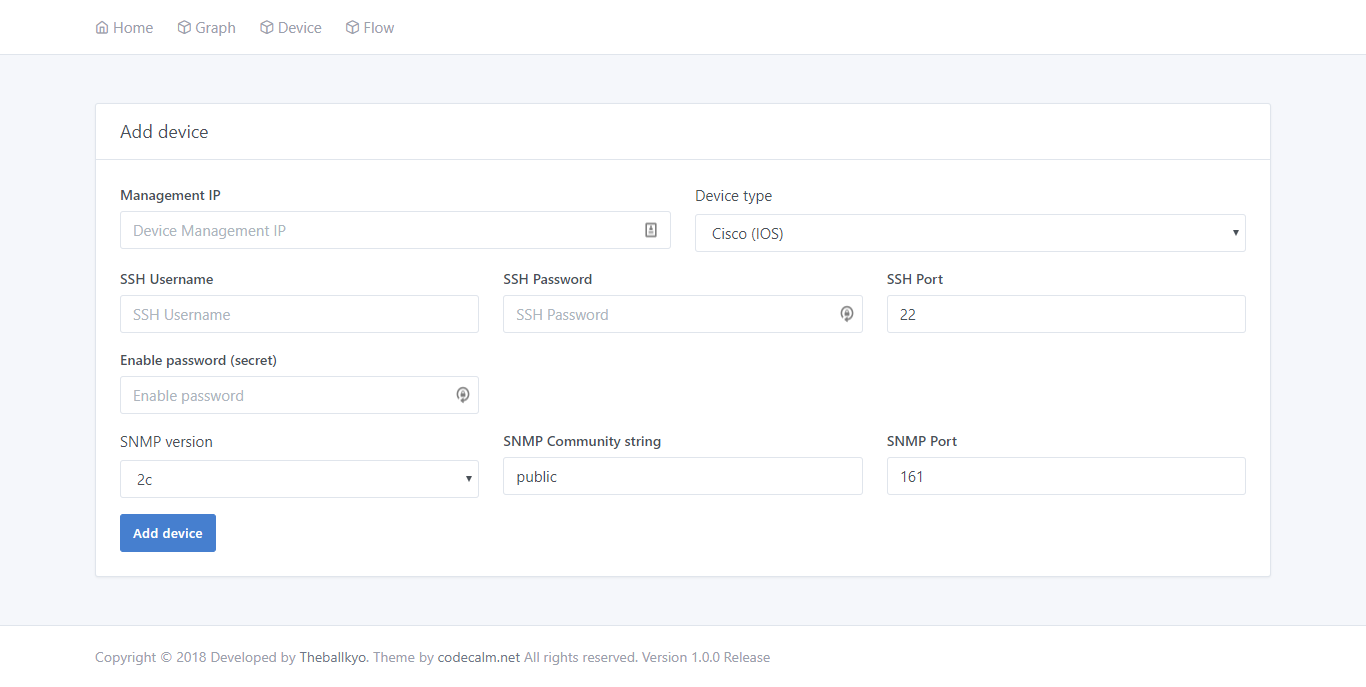


รูปที่ 3.2 แสดงตัวอย่าง Command Line Interface

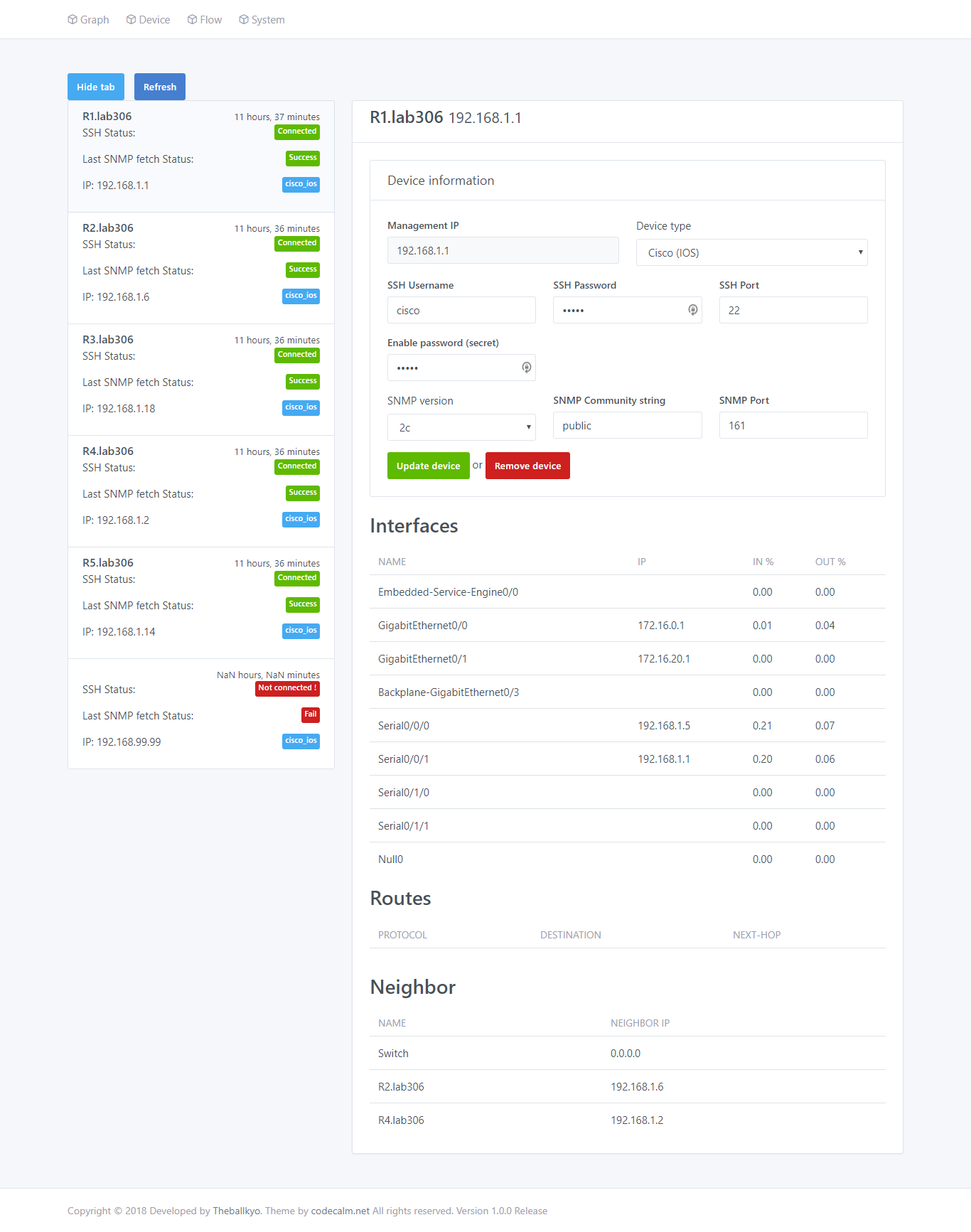
1. Graphical User Interface (GUI) เป็นการใช้งานระบบในรูปแบบของเว็บไซต์ โดยผู้ใช้สามารถใช้งานระบบหรือเรียกดูข้อมูลการทำงานของระบบเครือข่ายผ่านเว็บไซต์ได้เลย อีกทั้งยังสามารถเรียกดูแผนภาพการเชื่อมต่อของระบบเครือข่ายที่แสดงปริมาณการใช้งานของเส้นทางต่าง ๆ ภายในระบบเครือข่ายในรูปแบบของกราฟฟิคแบบเรียลไทม์ได้อีกด้วย



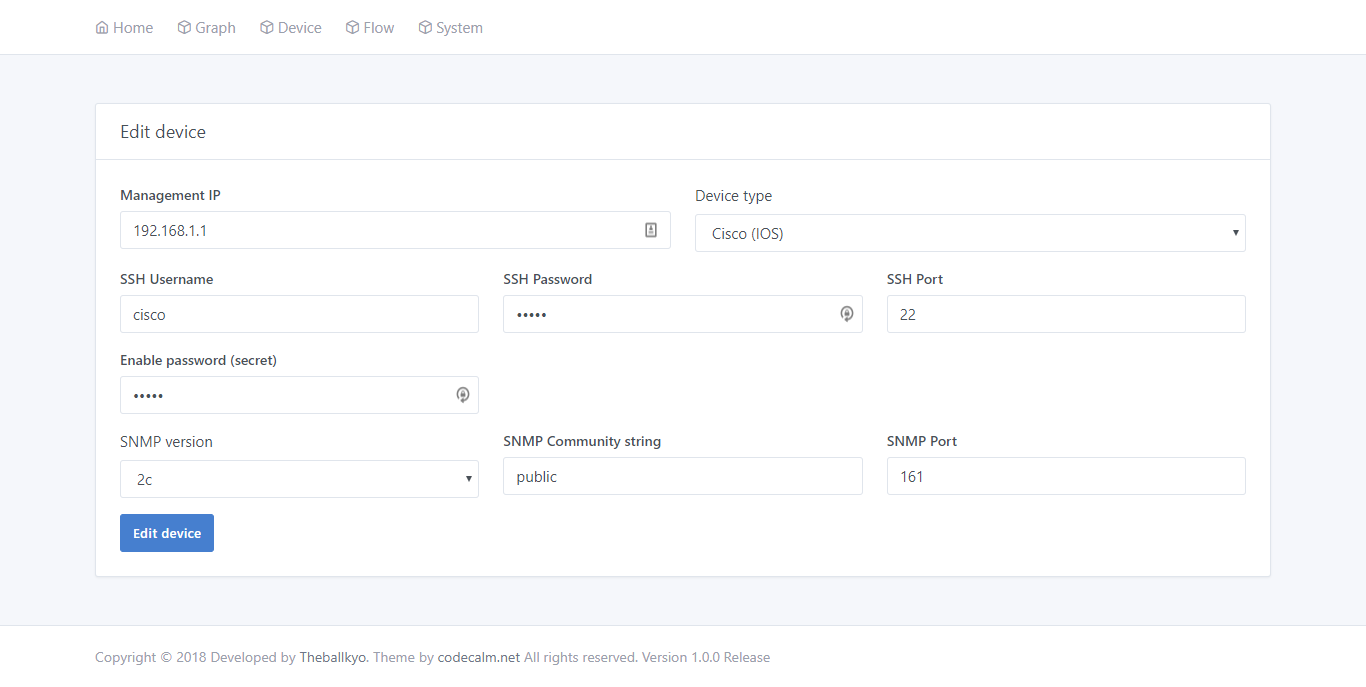
รูปที่ 3.3 แสดงหน้าจอแผนภาพการเชื่อมต่อของระบบเครือข่ายผ่านเว็บไซต์



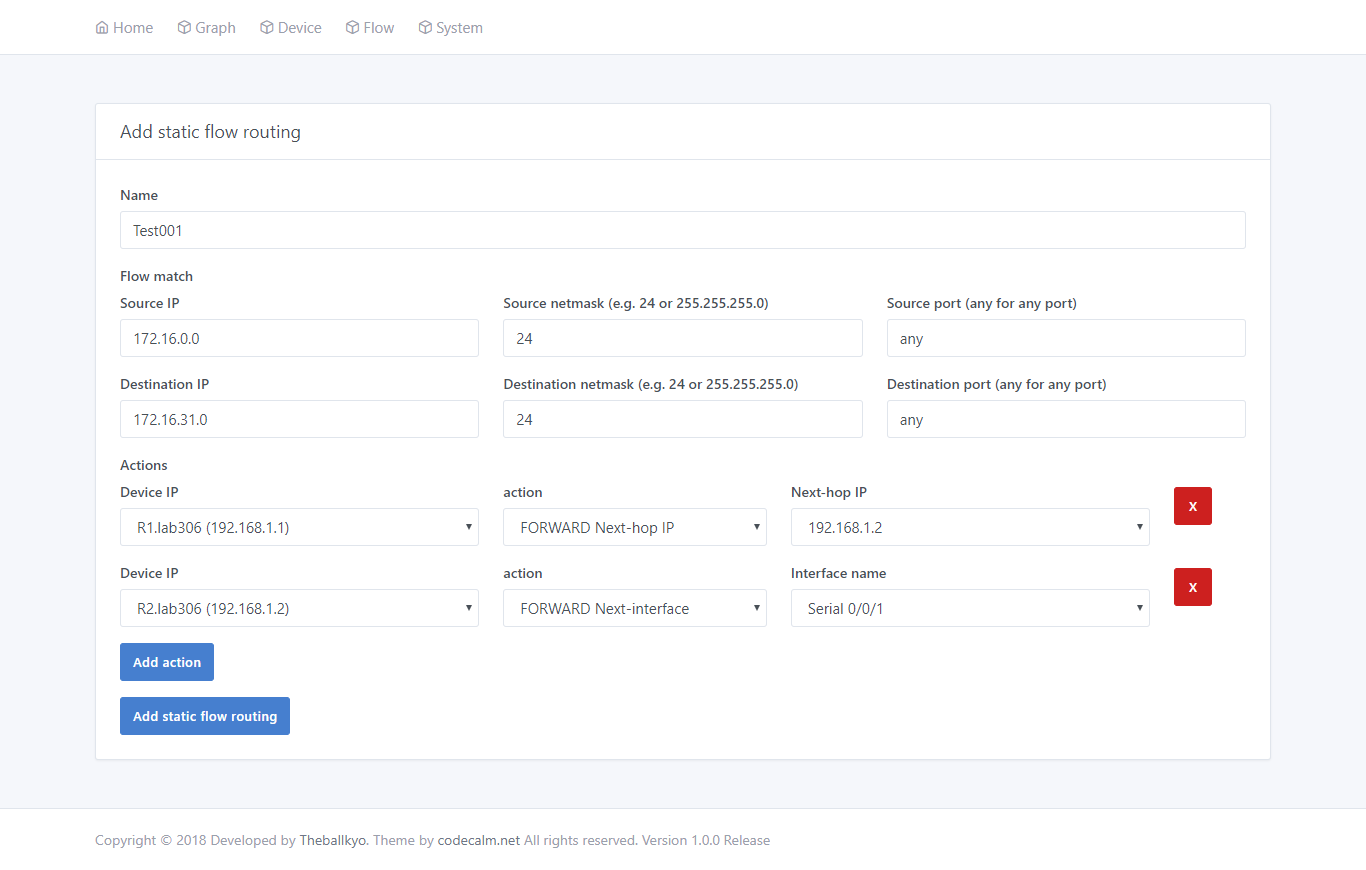
รูปที่ 3.4 แสดงหน้าจอการเพิ่มข้อมูลอุปกรณ์เครือข่ายผ่านเว็บไซต์



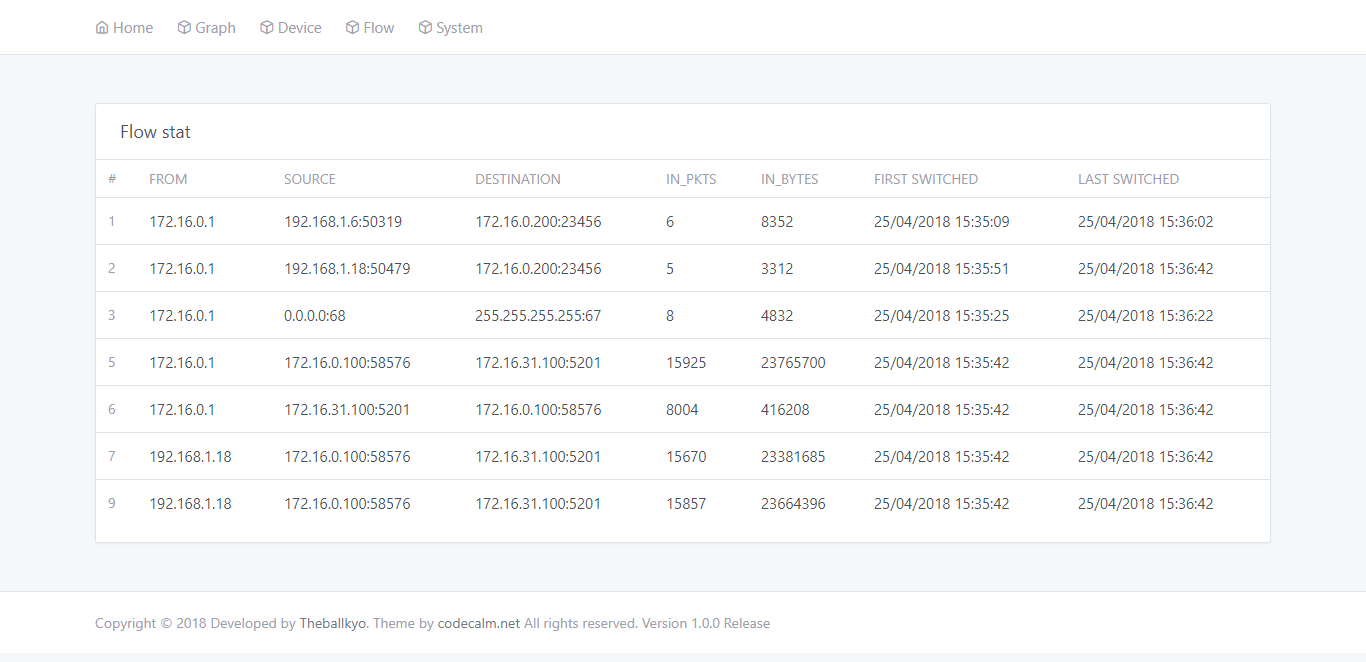
รูปที่ 3.5 แสดงหน้าจอรายละเอียดของอุปกรณ์เครือข่ายผ่านเว็บไซต์



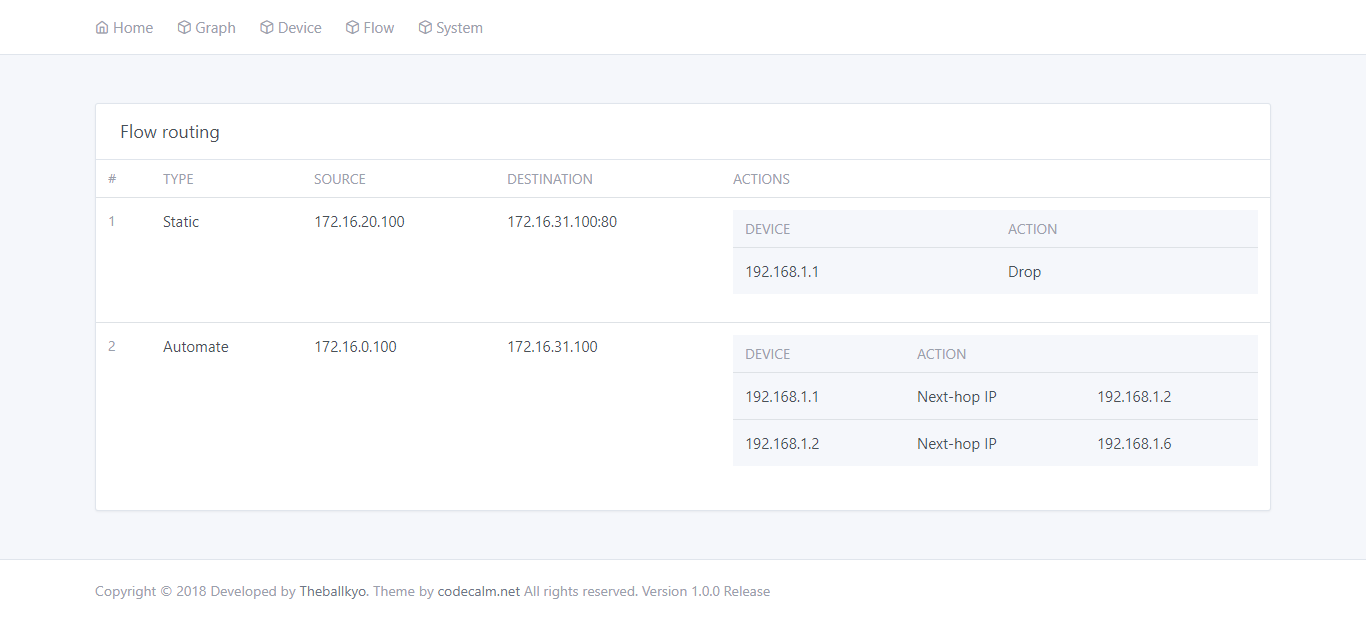
รูปที่ 3.6 แสดงหน้าจอการแก้ไขข้อมูลอุปกรณ์เครือข่ายผ่านเว็บไซต์



รูปที่ 3.7 แสดงหน้าจอบังคับเปลี่ยนเส้นทางของอุปกรณ์เครือข่ายแบบกำหนดเองผ่านเว็บไซต์



รูปที่ 3.8 แสดงหน้าจอข้อมูลของโฟลว์ภายในระบบเครือข่ายผ่านเว็บไซต์



รูปที่ 3.9 แสดงหน้าจอข้อมูลการบังคับเปลี่ยนเส้นทางของระบบผ่านเว็บไซต์

ทั้งนี้ เมื่อผู้ใช้งานเข้าใช้งานระบบจำเป็นจะต้องทำการเพิ่มข้อมูลของอุปกรณ์เครือข่ายเบื้องต้นเข้าสู่ระบบ เพื่อให้ระบบจัดเก็บข้อมูลและทำการสร้างการเชื่อมต่อไปยังอุปกรณ์เครือข่าย โดยข้อมูลเบื้องต้นที่ผู้ใช้งานต้องระบุนั้น ประกอบไปด้วย

1. ประเภทอุปกรณ์เครือข่าย ซึ่งปัจจุบันระบบรองรับเพียง Cisco iOS ซึ่งเป็นอุปกรณ์เครือข่ายของบริษัทซิสโก้ที่รันระบบปฏิบัติการที่มีชื่อว่า iOS
2. หมายเลขแอดเดรสของอุปกรณ์เครือข่ายที่มีการตั้งค่าให้สามารถใช้งาน SNMP และการเข้าถึงอุปกรณ์ระยะไกลผ่านโพรโทคอล Secure shell (SSH) ได้
3. ชื่อผู้ใช้สำหรับการเข้าถึงระยะไกลผ่านโพรโทคอล SSH
4. รหัสผ่านสำหรับการเข้าถึงระยะไกลผ่านโพรโทคอล SSH
5. รหัสการเปิดใช้งานของอุปกรณ์เครือข่าย
6. หมายเลขพอร์ตของโพรโทคอล Secure shell ซึ่งค่าเริ่มต้นคือ 22
7. ค่า SNMP Community string ซึ่งค่าเริ่มต้นคือ public
8. หมายเลขพอร์ตของโพรโทคอล SNMP ซึ่งค่าเริ่มต้นคือ 161

ก่อนการใช้งานระบบผู้ใช้งานจำเป็นจะต้องมีการตั้งค่าอุปกรณ์เครือข่ายเบื้องต้น เพื่อให้สามารถทำงานร่วมกับระบบได้ ดังนี้

* เปิดการใช้งานการเข้าถึงจากระยะไกลผ่านโพรโทคอล Secure Shall (SSH)
* เปิดการใช้งานโพรโทคอล SNMP
* เปิดการใช้งาน NetFlow ตั้งค่าให้ส่งออกข้อมูลไปที่ระบบและกำหนดการส่งออกโฟลว์ Active เป็นเวลา 1 นาที โฟลว์ Inactive เป็นเวลา 20 วินาที และกำหนดค่า refresh-rate เป็น 1 โดยกำหนด version ของการส่งออกข้อมูล คือ 9
* กำหนด policy ที่อินเตอร์เฟสที่มีการเชื่อมต่อและใช้งานให้มีการใช้งาน policy ที่ชื่อ SDN-handmade โดยเข้าไปที่หน้าตั้งค่าอินเตอร์เฟสและพิมพ์คำสั่ง “ip policy route-map SDN-handmade”

### 3.2.2 การตรวจสอบและเก็บสถานะของระบบเครือข่าย

ระบบมีกระบวนการในการจัดเก็บข้อมูลของระบบเครือข่าย ซึ่งข้อมูลที่จัดเก็บเข้าสู่ระบบจะอ้างอิงตามการใช้งานของระบบจัดการจราจรเครือข่ายที่มีการใช้งานกับอุปกรณ์เครือข่ายของ Cisco เท่านั้น แต่ในกรณีที่ต้องการจัดเก็บข้อมูลอื่น ๆ หรือจัดเก็บจากอุปกรณ์เครือข่ายประเภทอื่นก็สามารถที่จะกำหนดเพิ่มเติมเข้าสู่ระบบได้ โดยกระบวนการในการจัดเก็บข้อมูลมีอยู่ 2 ลักษณะ ได้แก่

1. ระบบเป็นผู้ร้องขอข้อมูลจากอุปกรณ์เครือข่าย ซึ่งระบบที่พัฒนาขึ้นในปัจจุบันใช้โพรโทคอล SNMP ในการร้องขอข้อมูลจากอุปกรณ์เครือข่าย โดยข้อมูลที่ระบบร้องขอนั้นจะประกอบไปด้วย

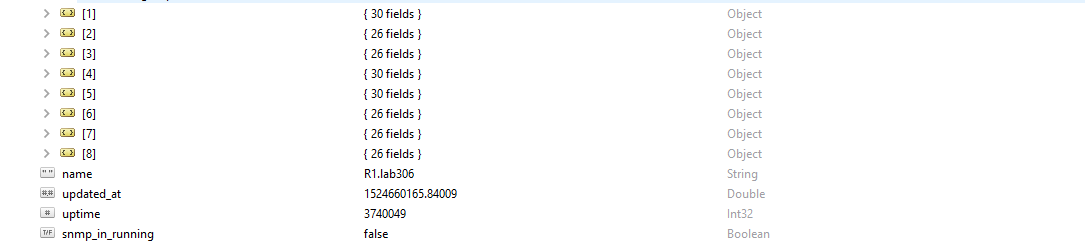
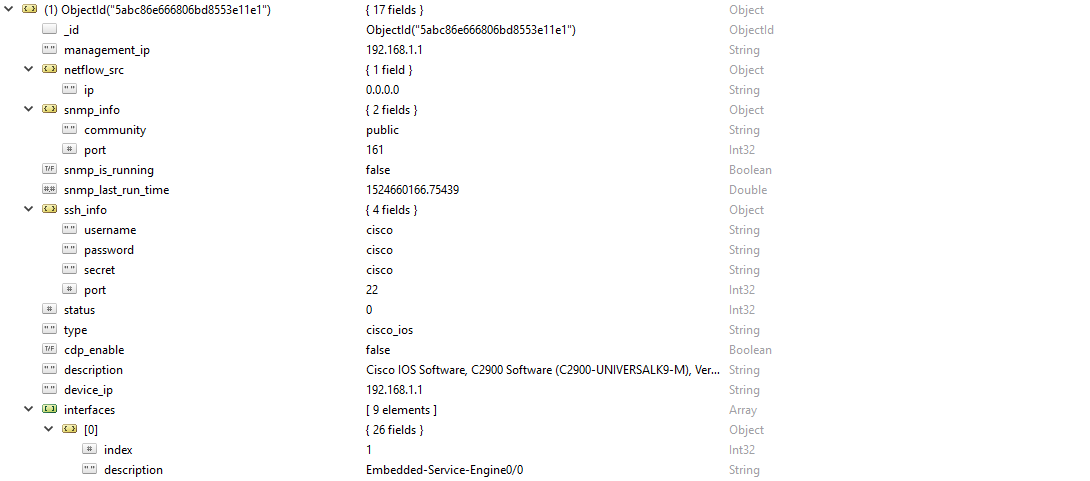
* ข้อมูลพื้นฐานของอุปกรณ์เครือข่าย เช่น ชื่ออุปกรณ์เครือข่าย หมายเลขแอดเดรส เป็นต้น ซึ่งจะจัดเก็บข้อมูลในตาราง Device Table
* ตารางเส้นทางของอุปกรณ์เครือข่าย เพื่อนำมาใช้ในการตรวจสอบเส้นทางการรับส่งข้อมูลของอุปกรณ์เครือข่ายในปัจจุบัน ซึ่งข้อมูลดังกล่าวจะถูกจัดเก็บในตาราง Device Routing Table
* ข้อมูลอินเตอร์เฟสของอุปกรณ์เครือข่าย ซึ่งบอกถึงข้อมูลต่าง ๆ ของอินเตอร์เฟส เช่น ชื่ออินเตอร์เฟส หมายเลขแอดเดรสของอินเตอร์เฟส ปริมาณการใช้งานของอินเตอร์เฟส เป็นต้น โดยที่ข้อมูลในส่วนนี้นั้น จะถูกนำมาใช้ในกระบวนการตรวจสอบความคับคั่งของเส้นทางภายในระบบเครือข่าย ซึ่งจะถูกจัดเก็บในตาราง Device Table
* ข้อมูล CDP เป็นข้อมูลที่บอกถึงรายละเอียดของอุปกรณ์เครือข่ายที่เชื่อมต่อกันของอุปกรณ์เครือข่ายแต่ละเครื่อง โดยข้อมูลดังกล่าวจะถูกนำใช้ในการสร้างแผนภาพการเชื่อมต่อของระบบเครือข่าย ซึ่งจะถูกจัดเก็บในตาราง Device Relationship Table

1. อุปกรณ์เครือข่ายส่งข้อมูลมายังระบบ ซึ่งจะเป็นข้อมูลในส่วนของ NetFlow ที่บอกถึงรายละเอียดของโฟลว์ข้อมูลต่าง ๆ ที่มีการรับส่งผ่านอุปกรณ์เครือข่ายเครื่องดังกล่าว โดยที่การส่งข้อมูลนั้นอุปกรณ์เครือข่ายจะส่งข้อมูล Netflow มายังระบบทุก ๆ ช่วงเวลา เพื่อเป็นการอัพเดตข้อมูลอยู่ตลอดเวลาและระบบจะเปิดช่องทางสำหรับรองรับข้อมูลเอาไว้ โดยข้อมูลดังกล่าวจะถูกเก็บอยู่ในตาราง Flow Stat Table

### 3.2.5 ฐานข้อมูลภายในระบบ

ระบบใช้ฐานข้อมูล MongoDB ในการพัฒนา เนื่องจากมีความยืดหยุ่นในการใช้งานและสามารถรองรับการเปลี่ยนแปลงลักษณะของข้อมูลที่จัดเก็บได้ ทำให้เหมาะกับระบบบริหารจัดการเครือข่ายที่อาจมีการเปลี่ยนแปลงการทำงานได้เป็นอย่างดี ซึ่งภายในฐานข้อมูลจะประกอบไปด้วยตารางข้อมูล ดังต่อไปนี้

* Device Table ซึ่งจะจัดเก็บข้อมูลเกี่ยวกับอุปกรณ์เครือข่าย เช่น หมายเลขแอดเดรสของอุปกรณ์ ชื่อ ข้อมูลการเข้าถึงจากระยะไกล ข้อมูลอินเตอร์เฟส เป็นต้น โดยที่ข้อมูลภายในตารางจะมาจากการที่ผู้ใช้งานเพิ่มข้อมูลอุปกรณ์เครือข่ายเข้าสู่ระบบและระบบได้รับจากการร้องขอข้อมูลจากอุปกรณ์เครือข่ายด้วยโพรโทคอล SNMP



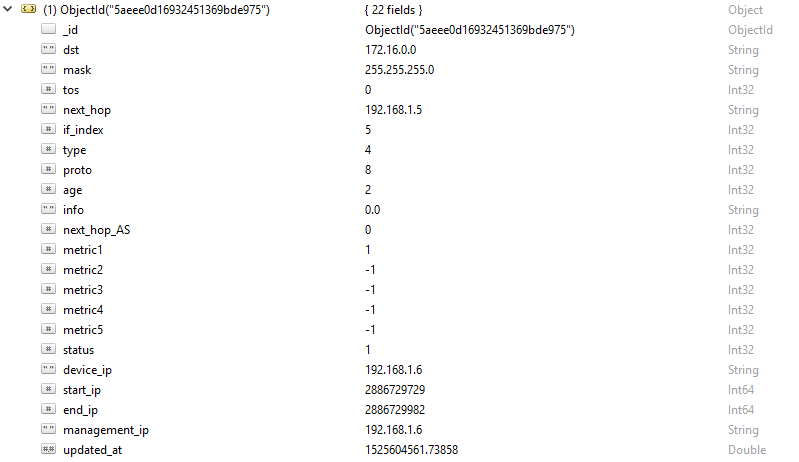
รูปที่ 3.10 แสดงตัวอย่างข้อมูลภายในตาราง Device Table

* Device Relationship Table เป็นตารางที่จัดเก็บข้อมูลที่ได้รับจาก CDP ซึ่งบอกรายละเอียดเกี่ยวกับอุปกรณ์เครือข่ายที่เชื่อมต่อกันของอุปกรณ์เครือข่ายแต่ละเครื่อง โดยที่ระบบจะนำไปใช้ในการสร้างแผนภาพการเชื่อมต่อของระบบเครือข่าย



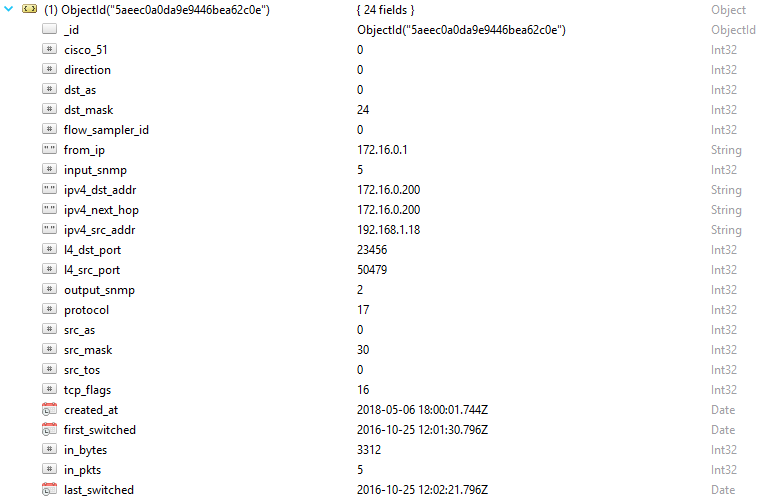
รูปที่ 3.11 แสดงตัวอย่างข้อมูลภายในตาราง Device Relationship Table

* Device Routing Table เป็นตารางที่จัดเก็บข้อมูลตารางเส้นทางของอุปกรณ์เครือข่ายแต่ละตัว เพื่อให้ระบบทราบเส้นทางปัจจุบันที่อุปกรณ์เครือข่ายใช้ในการรับส่งข้อมูล ซึ่งข้อมูลจะถูกอัพเดทจากการที่ระบบได้รับข้อมูลของโพรโทคอล SNMP



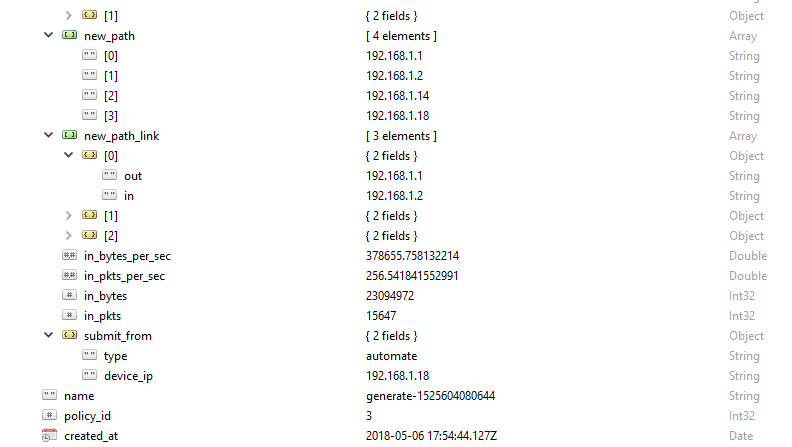
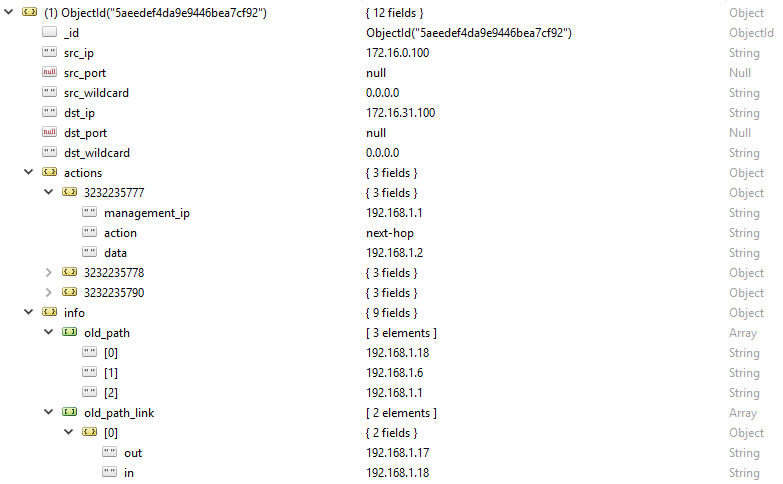
รูปที่ 3.12 แสดงตัวอย่างข้อมูลภายในตาราง Device Routing Table

* Flow Stat Table เป็นตารางที่จัดเก็บข้อมูลที่ได้รับจาก NetFlow ซึ่งเป็นข้อมูลรายละเอียดเกี่ยวกับโฟลว์ข้อมูลที่มีการรับส่งผ่านอุปกรณ์เครือข่ายแต่ละเครื่อง เพื่อนำมาใช้ในกระบวนการเลือกโฟลว์ที่จะบังคับเปลี่ยนเส้นทางการรับส่ง โดยข้อมูลจะถูกอัพเดทจากการที่ระบบได้รับข้อมูลของ NetFlow ที่ส่งมายังระบบ



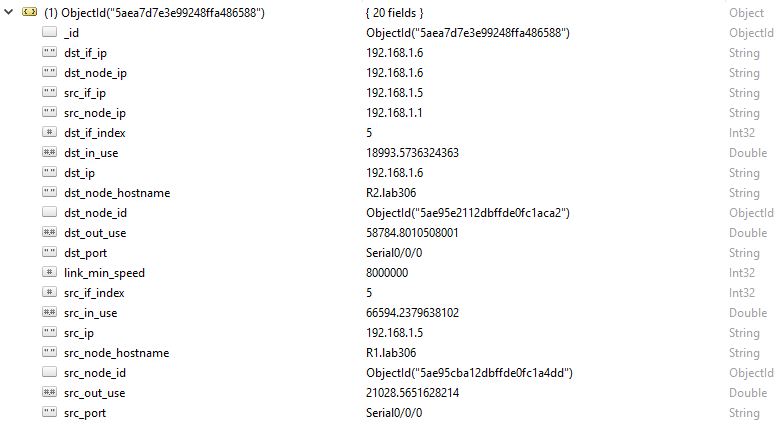
รูปที่ 3.13 แสดงตัวอย่างข้อมูลภายในตาราง Flow Stat Table

* Flow Routing Table เป็นตารางที่จัดเก็บข้อมูลรายละเอียดของโฟลว์ข้อมูลที่ระบบทำการบังคับเปลี่ยนเส้นทางการรับส่ง ซึ่งจะระบุทั้งข้อมูลโฟลว์และเส้นทางที่บังคับเปลี่ยนเส้นทางไป โดยข้อมูลในตารางจะอัพเดทจากกระบวนการบังคับเปลี่ยนเส้นทางการรับส่งข้อมูลของระบบและการบังคับเปลี่ยนเส้นทางการรับส่งข้อมูลของผู้ใช้งาน ทั้งนี้ระบบจะมีการลบข้อมูลของโฟลว์ที่มีการยกเลิกการบังคับเปลี่ยนเส้นทางหรือโฟลว์ที่สิ้นสุดการรับส่งข้อมูลไปแล้ว



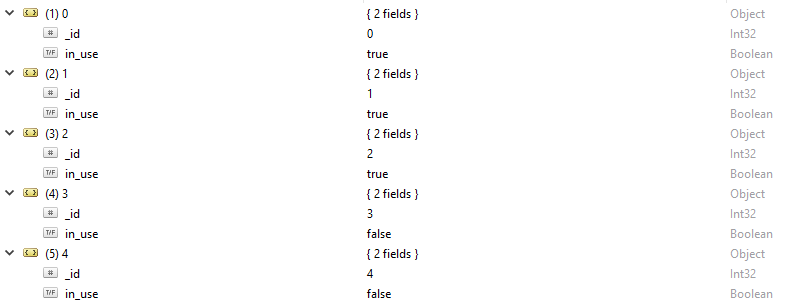
รูปที่ 3.14 แสดงตัวอย่างข้อมูลภายในตาราง Flow Routing Table

* Link Utilization Table: จัดเก็บข้อมูลปริมาณการใช้งานของเส้นทางภายในระบบเครือข่าย เพื่อนำมาใช้การตรวจสอบหาเส้นทางที่มีการใช้งานคับคั่งและเลือกเส้นทางใหม่สำหรับโฟลว์ข้อมูลที่ต้องการบังคับเส้นทางการรับส่ง ซึ่งข้อมูลจะอัพเดทจากการที่ระบบได้รับข้อมูลจากโพรโทคอล SNMP



รูปที่ 3.15 แสดงตัวอย่างข้อมูลภายในตาราง Link Utilization Table

* Used Flow ID List Table: จัดเก็บข้อมูลของหมายเลขโฟลว์ข้อมูลที่มีการใช้งานไปแล้วภายในระบบ ซึ่งจะอัพเดทจากกระบวนการทำงานของระบบ

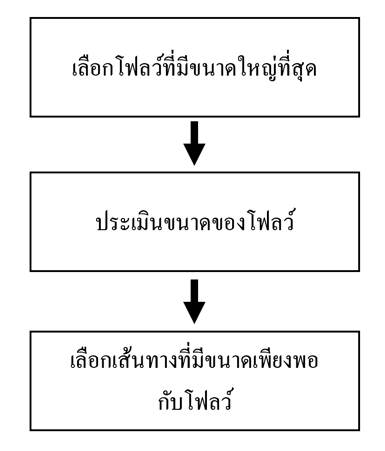


รูปที่ 3.16 แสดงตัวอย่างข้อมูลภายในตาราง Used Flow ID List Table

### 3.2.3 การบริหารจัดการจราจรเครือข่าย

ระบบมีกระบวนการในการตรวจสอบเส้นทางภายในระบบเครือข่ายที่เกิดความคับคับและมีการทำวิศวกรรมจราจรในการบังคับเปลี่ยนเส้นทางการรับส่งข้อมูลบางส่วนของเส้นทางที่เกิดความคับคั่งให้เปลี่ยนไปใช้เส้นทางอื่น เพื่อลดความคับคั่งของเส้นทางที่เกิดขึ้น โดยกระบวนการทำงานแบ่งออกเป็นส่วนต่าง ๆ ดังนี้

* + ตรวจสอบความคับคั่งของเส้นทางภายในระบบเครือข่าย: โดยตรวจสอบจากข้อมูลอินเตอร์เฟสของอุปกรณ์เครือข่าย โดยนำปริมาณแบนด์วิดท์ของอินเตอร์เฟสมาเปรียบเทียบกับปริมาณการรับส่งข้อมูลในปัจจุบัน เพื่อหาระดับการใช้งานของเส้นทางในปัจจุบัน (คิดเป็นเปอร์เซ็นต์การใช้งาน) โดยผู้ใช้งานเป็นผู้กำหนดเกณฑ์เปอร์เซ็นต์การใช้งานของเส้นทางที่ถือว่าเกิดความคับคั่ง หากเส้นทางมีการใช้งานเกินเปอร์เซ็นต์ที่กำหนดจะถือว่าเส้นทางเกิดความคับคั่ง
  + การเลือกโฟลว์และเส้นทางใหม่ของโฟลว์: โดยกระบวนการทำงานนี้จะอ้างอิงตามแนวคิดของ “Global First Fit” [20] ที่มีแนวคิดการทำงานตามรูปที่ 3.17



รูปที่ 3.17 แสดงแนวคิดการทำงานของ Global First Fit

โดยการทำงานจะเริ่มจากการเลือกโฟลว์ข้อมูลที่มีขนาดใหญ่ที่สุดของเส้นทาง นำมาประเมินขนาดของโฟลว์เพื่อหาปริมาณการใช้งานแบนด์วิดท์ที่ต้องการ หลังจากนั้นทำการค้นหาเส้นทางที่มีขนาดแบนด์วิดท์คงเหลือเพียงพอต่อปริมาณการใช้งานของโฟลว์ ถ้าพบเส้นทางมากกว่า 1 เส้นทางจะเลือกเส้นทางแรกที่พบ

ซึ่งระบบมีการประยุกต์ใช้แนวคิดดังกล่าวในกระบวนการทำงานของระบบ โดยการเพิ่มกระบวนการในการเลือกโฟลว์และเส้นทางให้มีประสิทธิภาพมากยิ่งขึ้น ดังนี้

* + การเลือกโฟลว์ ระบบจะเลือกโฟลว์ข้อมูลที่มีขนาดใหญ่ที่สุดของเส้นทางที่มีการใช้งานคับคั่ง
  + ประเมินขนาดของโฟลว์ เพื่อหาปริมาณแบนด์วิดท์ของเส้นทางที่โฟลว์ต้องการใช้ในการรับส่งข้อมูล
  + ค้นหาเส้นทางที่เป็นไปได้ทั้งหมด จากอุปกรณ์เครือข่ายเครื่องที่เกิดความคับคั่งไปยังปลายทาง โดยเส้นทางดังกล่าวจะไม่นับรวมเส้นทางการรับส่งข้อมูลเดิมของโฟลว์ข้อมูลและเส้นทางที่มีการย้อนกลับไปในทิศทางที่โฟลว์ข้อมูลใช้ในการรับส่งข้อมูล เพื่อป้องกันการวนซ้ำ
  + ค้นหาเส้นทางที่มีขนาดแบนด์วิดท์คงเหลือในปัจจุบันเพียงพอต่อการใช้งานของโฟลว์ข้อมูล
  + ถ้าพบเส้นทางที่มีขนาดแบนด์วิดท์คงเหลือเพียงต่อการใช้งานของโฟลว์มากกว่า 1 เส้นทาง ระบบจะเลือกเส้นทางแรกที่ระบบค้นพบ
  + ถ้าไม่สามารถหาเส้นทางการรับส่งข้อมูลที่มีขนาดแบนด์วิดท์คงเหลือเพียงพอต่อการใช้งานของโฟลว์ข้อมูลได้ ระบบจะตรวจสอบเส้นทางการรับส่งข้อมูลของโฟลว์และเริ่มกระบวนการค้นหาเส้นทางใหม่ โดยค้นหาเส้นทางจากอุปกรณ์เครือข่ายก่อนหน้าที่โฟลว์ข้อมูลใช้ในการรับส่งข้อมูลไปยังปลายทาง
  + ถ้าไม่พบเส้นทางที่มีขนาดแบนด์วิดท์คงเหลือเพียงพอต่อการใช้งานของโฟลว์ข้อมูล ระบบจะทำกระบวนการค้นหาเส้นทางโดยย้อนหลังกลับไปหาเส้นทางจากอุปกรณ์เครือข่ายก่อนหน้าที่โฟลว์ข้อมูลใช้ในการรับส่งข้อมูล
  + ระบบจะทำกระบวนการค้นหาเส้นทางโดยย้อนกลับไปหาเส้นทางจากอุปกรณ์เครือข่ายก่อนหน้า จนกว่าระบบจะสามารถค้นหาเส้นทางที่มีขนาดแบนด์วิดท์เพียงพอต่อการรับส่งข้อมูลของโฟลว์ได้
  + ถ้าระบบค้นหาเส้นทางจนไปถึงอุปกรณ์เครือข่ายแรกที่โฟลว์ข้อมูลใช้ในการรับส่งข้อมูลแล้วไม่พบเส้นทาง ระบบจะเริ่มกระบวนการเลือกโฟลว์ข้อมูลที่จะบังคับเปลี่ยนเส้นทางใหม่โดยเลือกโฟลว์ข้อมูลที่มีขนาดใหญ่รองลงมาแทนและทำกระบวนการค้นหาเส้นทางใหม่อีกครั้ง
  + กระบวนการเลือกโฟลว์และค้นหาเส้นทางใหม่จะดำเนินการวนซ้ำไปเรื่อย ๆ จนกว่าจะค้นพบโฟลว์และเส้นทางใหม่สำหรับการบังคับเปลี่ยนเส้นทาง
  + โดยการกระบวนการเลือกโฟลว์และเส้นทางสำหรับการบังคับเปลี่ยนเส้นทางนี้ ผู้ใช้งานสามารถกำหนดเองได้ โดยการกำหนดลักษณะของโฟลว์ข้อมูลที่ต้องการจะบังคับเปลี่ยนเส้นทาง ได้แก่ หมายเลขแอดเดรสต้นทาง หมายเลขแอดเดรสปลายทาง หมายเลขพอร์ตต้นทาง หมายเลขพอร์ตปลายทาง และอีกส่วนคือ การกำหนดทิศทางการส่งออกโฟลว์ข้อมูลของอุปกรณ์เครือข่าย โดยกำหนดอุปกรณ์เครือข่ายและทิศทางการส่งออกโฟลว์ข้อมูล เช่น อุปกรณ์เครือข่าย R1 ส่งออกโฟลว์ข้อมูลทางอินเตอร์เฟส Serial0/0/0, อุปกรณ์เครือข่าย R2 ส่งออกโฟลว์ข้อมูลทางอินเตอร์เฟส Serial0/0/1 เป็นต้น

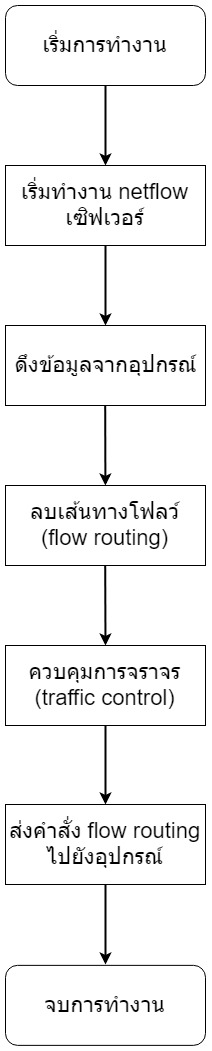
### 3.2.4 สร้างชุดคำสั่งควบคุมการทำงานของอุปกรณ์เครือข่าย

ซึ่งเป็นลักษณะของการควบคุมเส้นทางการส่งออกข้อมูลของอุปกรณ์เครือข่ายเมื่อลักษณะของโฟลว์ข้อมูลเป็นไปตามที่กำหนด โดยชุดคำสั่งประกอบไปด้วยองค์ประกอบ 2 ส่วน คือ

1. ส่วนการกำหนดโฟลว์ จะสร้างจาก Access Control List (ACL) ซึ่งระบุลักษณะของโฟลว์ข้อมูล ได้แก่ หมายเลขแอดเดรสต้นทาง หมายเลขแอดเดรสปลายทาง หมายเลขพอร์ตต้นทาง หมายเลขพอร์ตปลายทาง โพรโทคอล
2. การกำหนดทิศทาง สร้างโดยใช้ Route map ซึ่งเป็นลักษณะของการสร้างนโยบาย โดยกำหนดให้เงื่อนไขในการกระทำตามนโยบาย คือ ข้อมูลมีลักษณะตามที่กำหนดไว้ใน ACL และมีการกำหนดการกระทำต่อข้อมูลเป็นการส่งออกข้อมูลทางอินเตอร์เฟสของอุปกรณ์เครือข่ายตามทิศทางที่กำหนด

### 3.2.6 กระบวนการทำงานของระบบ

1. กระบวนการทำงานโดยรวมของระบบ



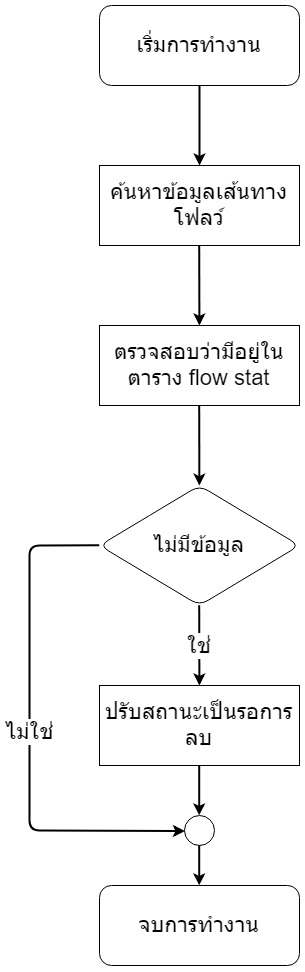
รูปที่ 3.18 แสดงกระบวนการทำงานโดยรวมของระบบ

การทำงานของระบบจะเริ่มจากการเปิดการทำงานในส่วนของ NetFlow Server ภายในระบบเพื่อรอรับข้อมูล NetFlow ที่อุปกรณ์จะส่งมาให้กับระบบ หลังจากนั้นจะทำการดึงข้อมูลต่าง ๆ จากอุปกรณ์เครือข่ายโดยสร้างคำสั่งในการร้องขอข้อมูลจากโพรโทคอล SNMPแล้วส่งไปยังอุปกรณ์ หลังจากนั้นระบบจะทำการตรวจสอบโฟลว์การบังคับเปลี่ยนเส้นทางที่ได้สร้างไว้ก่อนหน้านี้ว่าโฟลว์นั้นสิ้นสุดการทำงานไปแล้วหรือว่าสามารถย้ายกลับไปใช้เส้นทางเดิมได้หรือยัง ถ้าตรงตามเงื่อนไขระบบก็จะทำการลบโฟลว์การบังคับเปลี่ยนเส้นทางออกจากฐานข้อมูลในระบบและส่งคำสั่งไปยังอุปกรณ์เพื่อยกเลิกการบังคับเปลี่ยนเส้นทาง หลังจากนั้นระบบจะเริ่มดำเนินการในการควบคุมการจราจรโดยเริ่มจากการที่ระบบจะตรวจสอบเพื่อค้นหาเส้นทางที่เกิดความคับคั่งภายในเครือข่าย ถ้าพบเส้นทางที่เกิดความคับคั่งระบบจะเข้าสู่กระบวนการในการเลือกโฟลว์และเส้นทางในการบังคับเปลี่ยนเส้นทาง ซึ่งเมื่อระบบเลือกโฟลว์และเส้นทางในการย้ายได้แล้ว ระบบจะเริ่มกระบวนการในการสร้างคำสั่งเพื่อส่งไปยังอุปกรณ์ให้บังคับเปลี่ยนเส้นทางของโฟลว์

2. การลบโฟลว์การบังคับเปลี่ยนเส้นทาง

การลบโฟลว์การบังคับเปลี่ยนเส้นทางที่ระบบได้สร้างขึ้นออกจากฐานข้อมูลของระบบและจากอุปกรณ์เครือข่าย โดยจะมีลักษณะของการลบโฟลว์การบังคับเปลี่ยนเส้นทางอยู่ 2 กรณีดังนี้

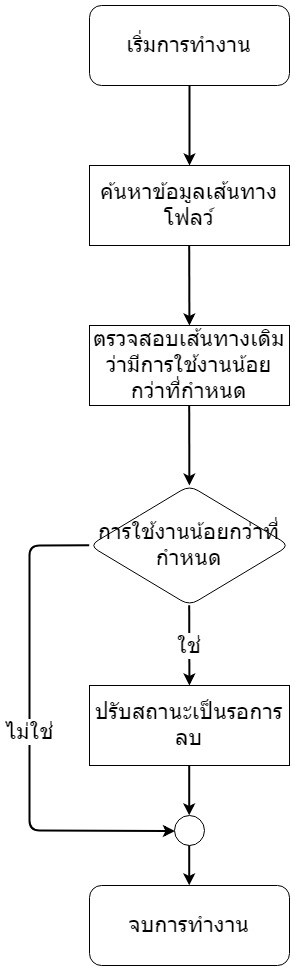
กรณีที่ 1 โฟลว์ที่ทำการบังคับเปลี่ยนเส้นทางสิ้นสุดการทำงานแล้ว



รูปที่ 3.19 แสดงกระบวนการลบโฟลว์การบังคับเปลี่ยนเส้นทางที่สิ้นสุดการทำงานแล้ว

กรณีดังกล่าวระบบจะทำการตรวจสอบโฟลว์ของการบังคับเปลี่ยนเส้นทางทั้งหมดที่ระบบสร้างไว้จากตาราง Flow Routing Table และตรวจสอบการคงอยู่ของโฟลว์ข้อมูลดังกล่าวจากตาราง Flow Stat Table ซึ่งจัดเก็บข้อมูลของโฟลว์ที่มีการใช้งานอยู่ทั้งหมดภายในเครือข่ายในปัจจุบัน โดยที่ถ้าไม่พบโฟลว์ข้อมูลดังกล่าวจากตารางแสดงว่า โฟลว์ข้อมูลดังกล่าวสิ้นสุดการใช้งานไปแล้ว ซึ่งระบบจะทำการเปลี่ยนสถานะของโฟลว์การบังคับเปลี่ยนเส้นทางดังกล่าวเป็น รอการลบ

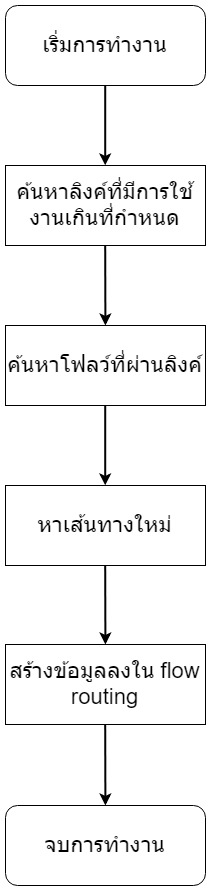
กรณีที่ 2 ย้ายกลับไปใช้งานเส้นทางเดิม



รูปที่ 3.20 แสดงกระบวนการลบโฟลว์การบังคับเปลี่ยนเส้นทางกรณีที่ 2

ระบบจะทำการตรวจสอบโฟลว์การบังคับเปลี่ยนเส้นทางทั้งหมดจากตาราง Flow Routing Table และทำการตรวจสอบปริมาณการใช้งานของเส้นทางเดิมที่โฟลว์ใช้งานก่อนการบังคับเปลี่ยนเส้นทาง ถ้าปริมาณการใช้งานของเส้นทางรวมกับปริมาณการใช้งานของโฟลว์แล้วต่ำกว่าเกณฑ์ความคับคั่งที่กำหนดไว้ ระบบจะทำการเปลี่ยนสถานะของโฟลว์การบังคับเปลี่ยนเส้นทางเป็น รอการลบ

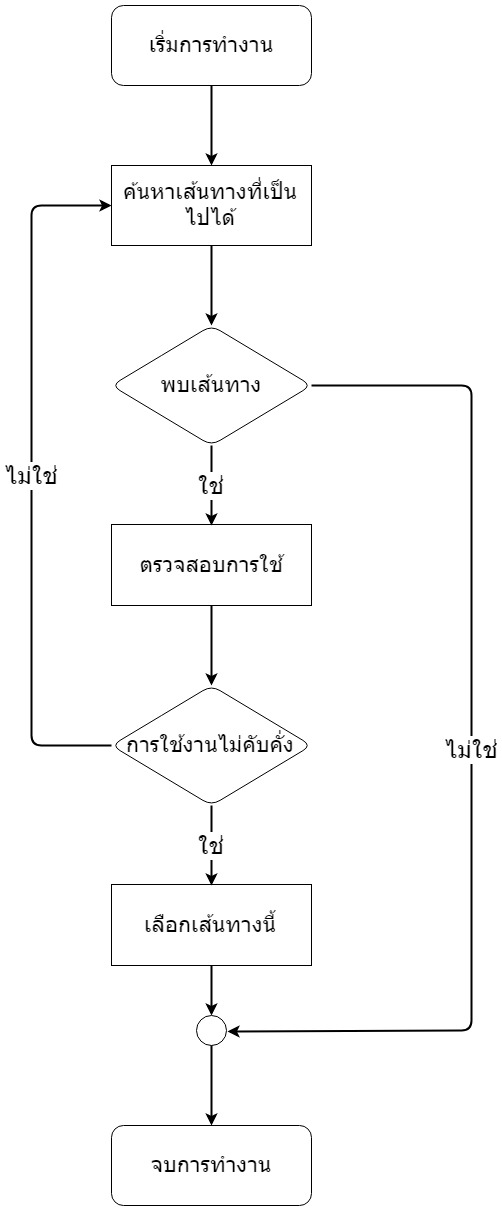
3. กระบวนการควบคุมการจราจร



รูปที่ 3.21 แสดงกระบวนการทำงานของการควบคุมจราจร

การควบคุมการจราจรจะเริ่มจากการค้นหาเส้นทางภายในระบบเครือข่ายที่เกิดความคับคั่ง โดยเปรียบเทียบปริมาณการใช้งานของเส้นทางว่าเกิดกว่าที่กำหนดไว้หรือไม่ (โดยคิดเป็นเปอร์เซ็นต์) ถ้ามีปริมาณการใช้งานเกินกว่าที่กำหนดจะถือว่า เส้นทางเกิดความคับคั่ง และระบบจะตรวจสอบว่ามีโฟลว์ข้อมูลที่มีการใช้งานผ่านเส้นทางดังกล่าว โดยการเลือกโฟลว์ระบบจะเลือกจากโฟลว์ที่มีปริมาณการใช้งานเส้นทางมากที่สุดที่สามารถย้ายไปใช้งานเส้นทางอื่นได้ และเลือกเส้นทางที่สามารถรองรับการใช้งานของโฟลว์ได้ เมื่อเลือกโฟลว์และเส้นทางได้แล้วระบบจะทำการบันทึกค่าดังกล่าวลงใน Flow Routing Table

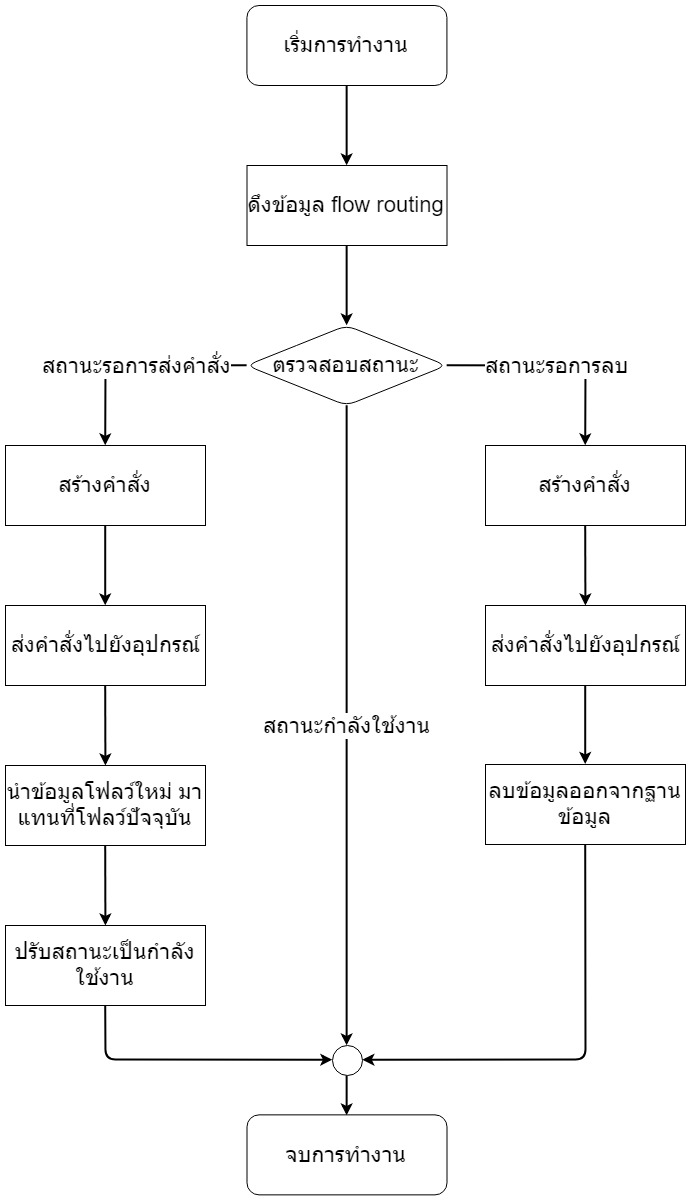
4. กระบวนการเลือกเส้นทางใหม่



รูปที่ 3.22 แสดงกระบวนการเลือกเส้นทางใหม่

การเลือกเส้นทางใหม่ จะเริ่มจากการค้นหาเส้นทางที่เป็นไปได้ โดยค้นหาจากการสร้างแผนภาพการเชื่อมต่อของทั้งระบบเครือข่ายและหาเส้นทางที่เป็นไปได้จากจุดเริ่มต้นไปยังปลายทาง ซึ่งเส้นทางที่ค้นหาได้จะไม่มีการเรียงลำดับของเส้นทาง หลังจากนั้นระบบจะทำการตรวจสอบว่าสามารถย้ายโฟลว์ไปใช้เส้นทางนั้นได้หรือไม่ จากการตรวจสอบปริมาณการใช้งานของลิงค์ที่มีขนาดเล็กที่สุดของเส้นทางเมื่อรวมกับปริมาณการใช้งานของโฟลว์ข้อมูลที่จะย้ายไปแล้วเกินกว่าเกณฑ์ความคับคั่งที่กำหนดไว้หรือไม่ ถ้าไม่เกินระบบก็จะเลือกเส้นทางดังกล่าว แต่ถ้าเกินระบบก็จะทำการค้นหาเส้นทางใหม่

5. กระบวนการส่งคำสั่ง flow routing ไปยังอุปกรณ์

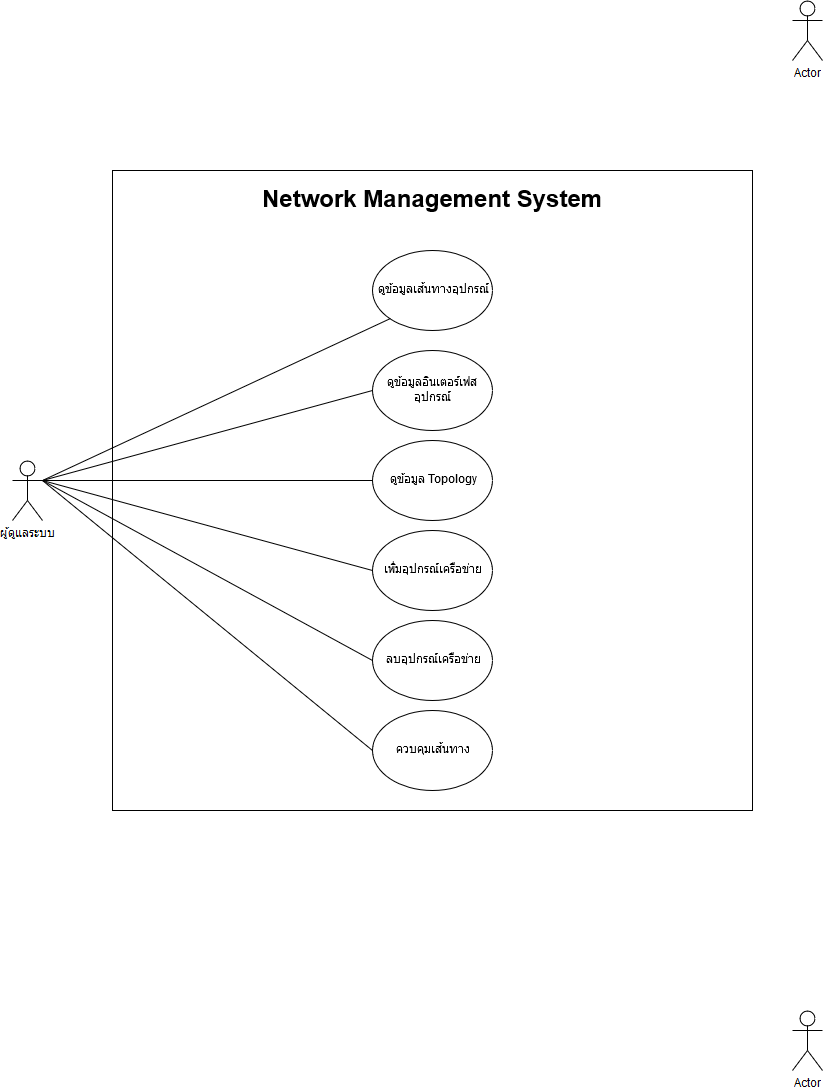


รูปที่ 3.23 แสดงกระบวนการทำงานส่งคำสั่งไปยังอุปกรณ์

การสร้างคำสั่งเพื่อส่งข้อมูลไปยังอุปกรณ์ จะเริ่มจากการที่ระบบตรวจสอบข้อมูลใน Flow Routing Table ซึ่งภายในจะจัดเก็บข้อมูลโฟลว์การบังคับเปลี่ยนเส้นทางที่ระบบได้สร้างขึ้น ซึ่งข้อมูลดังกล่าวจะมีสถานะอยู่ 3 สถานะ คือ รอการส่งคำสั่ง กำลังใช้งานและรอการลบ โดยที่ถ้าค้นพบโฟลว์การบังคับเปลี่ยนเส้นทางที่มีสถานะรอการส่งคำสั่งระบบจะดำเนินการสร้างคำสั่งสำหรับการบังคับเปลี่ยนเส้นทางและทำการส่งคำสั่งไปยังอุปกรณ์ หลังจากนั้นจะนำข้อมูลโฟลว์การบังคับเปลี่ยนเส้นทางใหม่มาแทนที่โฟลว์ปัจจุบันและเปลี่ยนสถานะของโฟลว์การบังคับเปลี่ยนเส้นทางเป็น กำลังใช้งาน และในกรณีที่ระบบพบโฟลว์ข้อมูลเป็นสถานะกำลังใช้งานระบบจะไม่ดำเนินการใด ๆ แต่ถ้าระบบพบว่าโฟลว์ข้อมูลอยู่ในสถานะรอการรับ ระบบจะดำเนินการในการสร้างคำสั่งเพื่อยกเลิกการบังคับเปลี่ยนเส้นทางของโฟลว์และส่งไปยังอุปกรณ์ หลังจากนั้นจะทำการลบข้อมูลของโฟลว์การบังคับเปลี่ยนเส้นทางดังกล่าวออกจาก Flow Routing Table

## 3.3 แผนภาพ Use-case

### 3.3.1 แผนภาพ Use-case Diagram



รูปที่ 3.22 แสดงแผนภาพ Use-case Diagram

### 3.3.2 แผนภาพ Use-case Description

ตารางที่ 3.1 แสดงคำอธิบายของยูสเคส เพิ่มอุปกรณ์เครือข่าย

|  |  |  |
| --- | --- | --- |
| Use Case Name: | เพิ่มอุปกรณ์เครือข่าย | |
| Actor: | ผู้ดูแลระบบ | |
| Brief Description: | ผู้ใช้ทำการป้อนคำสั่งเพื่อเพิ่มอุปกรณ์เครือข่ายเข้าสู่ระบบ แล้วระบบจะทำการติดต่อขอข้อมูลจากอุปกรณ์เครือข่ายเอง | |
| Flow of Event: | Actor | System |
| 1. ผู้ใช้ป้อน ไอพีแอดเดรสของอุปกรณ์เครือข่าย ชื่อผู้ใช้ รหัสผ่าน พอร์ต Secure shell, คอมมิวนิตี้สตริง และ พอร์ต SNMP  2. ผู้ใช้กดปุ่มเอ็นเทอร์ | 3.ระบบรับค่าที่ผู้ใช้ป้อนแล้วติดต่อขอข้อมูลกับอุปกรณ์เครือข่าย |
| Pre-Conditions: | เข้าสู่โหมดการตั้งค่าโดยการพิมพ์ config ในหน้าจอ | |
| Post-Conditions: | ข้อมูลของอุปกรณ์เครือข่ายถูกเพิ่มเข้าสู่ระบบ | |

ตารางที่ 3.2 แสดงคำอธิบายของยูสเคส ลบอุปกรณ์เครือข่าย

|  |  |  |
| --- | --- | --- |
| Use Case Name: | ลบอุปกรณ์เครือข่าย | |
| Actor: | ผู้ดูแลระบบ | |
| Brief Description: | ผู้ใช้ทำการป้อนคำสั่งเพื่อลบอุปกรณ์เครือข่ายออกจากระบบ แล้วระบบจะทำการยกเลิกการติดต่อขอข้อมูลจากอุปกรณ์เครือข่ายและลบข้อมูลออกจากระบบ | |
| Flow of Event: | Actor | System |
| 1. ผู้ใช้ป้อนไอพีแอดเดรสของอุปกรณ์  2. ผู้ใช้กดปุ่ม เพื่อยืนยันคำสั่ง | 3.ระบบรับค่าที่ผู้ใช้ป้อนและทำการลบข้อมูลอุปกรณ์ออกจากระบบ |
| Pre-Conditions: | เข้าสู่โหมดการตั้งค่าโดยการพิมพ์ config ในหน้าจอ | |
| Post-Conditions: | ข้อมูลของอุปกรณ์เครือข่ายถูกลบออกจากระบบ | |

ตารางที่ 3.3 แสดงคำอธิบายของยูสเคส ควบคุมเส้นทาง

|  |  |  |
| --- | --- | --- |
| Use Case Name: | ควบคุมเส้นทาง | |
| Actor: | ผู้ดูแลระบบ | |
| Brief Description: | ผู้ใช้ทำการป้อนคำสั่งเพื่อควบคุมเส้นทางของแต่ละโฟลว์ในเครือข่าย และระบบจะส่งคำสั่งไปแก้ไขเส้นทางยังอุปกรณ์เครือข่ายแต่ละตัวตามที่ผู้ใช้งานตั้งค่าไว้ | |
| Flow of Event: | Actor | System |
| 1. ผู้ใช้ป้อนข้อมูลเส้นทางที่ต้องการจะควบคุม และใส่ข้อมูลการเปลี่ยนเส้นทางของแต่ละอุปกรณ์เครือข่าย  2.ผู้ใช้ส่งคำสั่ง “apply” เพื่อยืนยันการแก้ไขเส้นทาง | 3.ระบบรับค่าที่ผู้ใช้ป้อน และสั่งการแก้ไขเส้นทางไปยังอุปกรณ์เครือข่ายต่างๆ |
| Pre-Conditions: | เข้าสู่โหมดการตั้งค่าโดยการพิมพ์ config ในหน้าจอ | |
| Post-Conditions: | เส้นทางถูกเปลี่ยนไปตามที่ผู้ใช้กำหนด | |

ตารางที่ 3.4 แสดงคำอธิบายของยูสเคส ดูข้อมูล Topology

|  |  |  |
| --- | --- | --- |
| Use Case Name: | ดูข้อมูล Topology | |
| Actor: | ผู้ดูแลระบบ | |
| Brief Description: | ผู้ใช้ทำการป้อนคำสั่งเพื่อดูข้อมูลการเชื่อมต่อของแต่ละอุปกรณ์เครือข่ายในระบบว่ามีการเชื่อมต่อกันอย่างไร | |
| Flow of Event: | Actor | System |
| 1. ผู้ใช้ส่งคำสั่งว่า “show topology”  2.ผู้ใช้กดปุ่ม เพื่อยืนยันคำสั่ง | 3.ระบบแสดงข้อมูลการเชื่อมต่ออุปกรณ์อยู่ในรูปแบบ matrix |
| Pre-Conditions: | - | |
| Post-Conditions: | ระบบแสดงข้อมูลการเชื่อมต่อของอุปกรณ์เครือข่าย | |

ตารางที่ 3.5 แสดงคำอธิบายของยูสเคส ดูข้อมูลอินเตอร์เฟสของอุปกรณ์เครือข่าย

|  |  |  |
| --- | --- | --- |
| Use Case Name: | ดูข้อมูลอินเตอร์เฟสของอุปกรณ์เครือข่าย | |
| Actor: | ผู้ดูแลระบบ | |
| Brief Description: | ผู้ใช้ทำการป้อนคำสั่งเพื่อดูข้อมูลอินเตอร์เฟสของอุปกรณ์เครือข่ายในระบบ | |
| Flow of Event: | Actor | System |
| 1. ผู้ใช้ป้อน ไอพีแอดเดรสของอุปกรณ์เครือข่าย  2.ผู้ใช้กดปุ่ม เพื่อยืนยันคำสั่ง | 3.ระบบรับค่าที่ผู้ใช้ป้อน และแสดงข้อมูลอินเตอร์ของอุปกรณ์เครือข่ายตามที่ผู้ใช้ระบุ |
| Pre-Conditions: | เพิ่มข้อมูลอุปกรณ์เครือข่ายเข้าสู่ระบบ | |
| Post-Conditions: | ระบบแสดงข้อมูลอินเตอร์เฟสของอุปกรณ์เครือข่าย | |

ตารางที่ 3.6 แสดงคำอธิบายของยูสเคส ดูข้อมูลเส้นทางของอุปกรณ์เครือข่าย

|  |  |  |
| --- | --- | --- |
| Use Case Name: | ดูข้อมูลเส้นทางของอุปกรณ์เครือข่าย | |
| Actor: | ผู้ดูแลระบบ | |
| Brief Description: | ผู้ใช้ทำการป้อนคำสั่งเพื่อดูข้อมูลเส้นทางของอุปกรณ์เครือข่ายในระบบ | |
| Flow of Event: | Actor | System |
| 1. ผู้ใช้ป้อน ไอพีแอดเดรสของอุปกรณ์เครือข่าย  2.ผู้ใช้กดปุ่ม เพื่อยืนยันคำสั่ง | 3.ระบบรับค่าที่ผู้ใช้ป้อน และแสดงข้อมูลเส้นทางของอุปกรณ์เครือข่ายตามที่ผู้ใช้ระบุ |
| Pre-Conditions: | เพิ่มข้อมูลอุปกรณ์เครือข่ายเข้าสู่ระบบ | |
| Post-Conditions: | ระบบแสดงข้อมูลเส้นทางของอุปกรณ์เครือข่าย | |

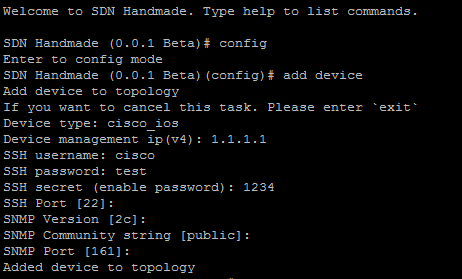
# บทที่ 4 การทดลอง และการประเมินผล

## 4.1 การทดลองการใช้งานผ่าน Command Line Interface

### 4.1.1 การเพิ่มอุปกรณ์

การทดลองในส่วนนี้จะเป็นการเพิ่มอุปกรณ์เครือข่ายเข้าสู่ระบบ เพื่อให้ระบบรู้จักอุปกรณ์เครือข่ายที่มีอยู่ในระบบเครือข่ายนั้น ๆ โดยมีขั้นตอนในการเพิ่มอุปกรณ์เครือข่ายเข้าสู่ระบบ ดังนี้

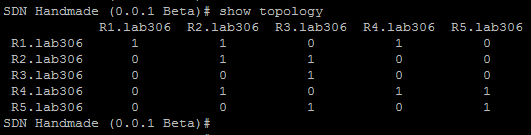
1. ทำการเข้าสู่โหมด config โดยพิมพ์ว่า “config”
2. พิมพ์คำสั่ง “add device”
3. ใส่ประเภทอุปกรณ์เครือข่ายที่ต้องการเพิ่มเข้าสู่ระบบ ซึ่งปัจจุบันระบบรองรับเพียง cisco\_ios ซึ่งเป็นอุปกรณ์เครือข่ายของบริษัทซิสโก้ที่รันระบบปฏิบัติการที่มีชื่อว่า IOS
4. ใส่ชื่อผู้ใช้สำหรับการเข้าถึงอุปกรณ์ระยะไกลผ่านโพรโทคอล Secure shell
5. ใส่รหัสผ่านสำหรับการเข้าถึงอุปกรณ์ระยะไกลผ่านโพรโทคอล Secure shell
6. ใส่รหัสการเปิดใช้งานของอุปกรณ์เครือข่าย
7. ใส่หมายเลขพอร์ตของโพรโทคอล Secure shell ซึ่งค่าเริ่มต้นคือ 22
8. ใส่เวอร์ชั่นของโพรโทคอล SNMP ซึ่งปัจจุบันรองรับเพียงเวอร์ชั่น 2c
9. ใส่ค่า SNMP Community string ซึ่งค่าเริ่มต้นคือ public
10. ใส่หมายเลขพอร์ตของโพรโทคอล SNMP ซึ่งค่าเริ่มต้นคือ 161



รูปที่ 4.1 หน้าจอแสดงการเพิ่มอุปกรณ์ผ่านทาง Command Line Interface

### 4.1.2 การแสดงข้อมูลการเชื่อมต่อกันของอุปกรณ์

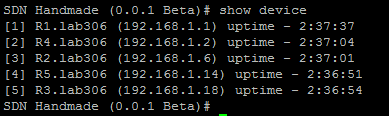
ทดลองโดยการพิมพ์คำสั่ง “show topology”



รูปที่ 4.2 หน้าจอแสดงการเชื่อมต่อของอุปกรณ์

### 4.1.3 การแสดงข้อมูลอุปกรณ์เครือข่าย

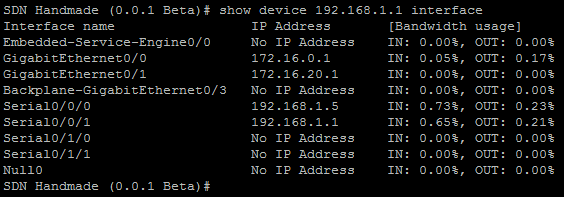
ทดลองโดยการพิมพ์คำสั่ง “show device”



รูปที่ 4.3 หน้าจอแสดงข้อมูลอุปกรณ์

### 4.1.4 การแสดงข้อมูลอินเตอร์เฟสของอุปกรณ์เครือข่าย

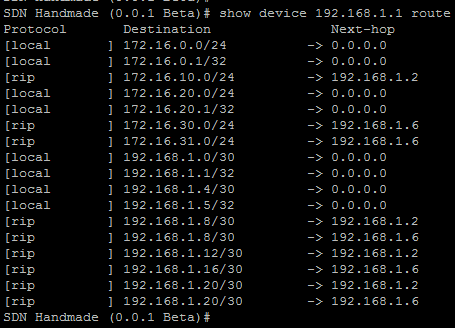
ทดลองโดยการพิมพ์คำสั่ง “show device <หมายเลขแอดเดรสของอุปกรณ์> interface”



รูปที่ 4.4 หน้าจอแสดงข้อมูลอินเตอร์เฟสของอุปกรณ์

### 4.1.5 การแสดงข้อมูลเส้นทางของอุปกรณ์เครือข่าย

ทดลองโดยการพิมพ์คำสั่ง “show device <หมายเลขแอดเดรสของอุปกรณ์> route”

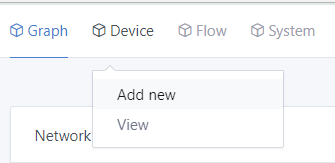


รูปที่ 4.5 หน้าจอแสดงข้อมูลเส้นทางของอุปกรณ์

## 4.2 การทดลองการใช้งานระบบผ่าน Web UI

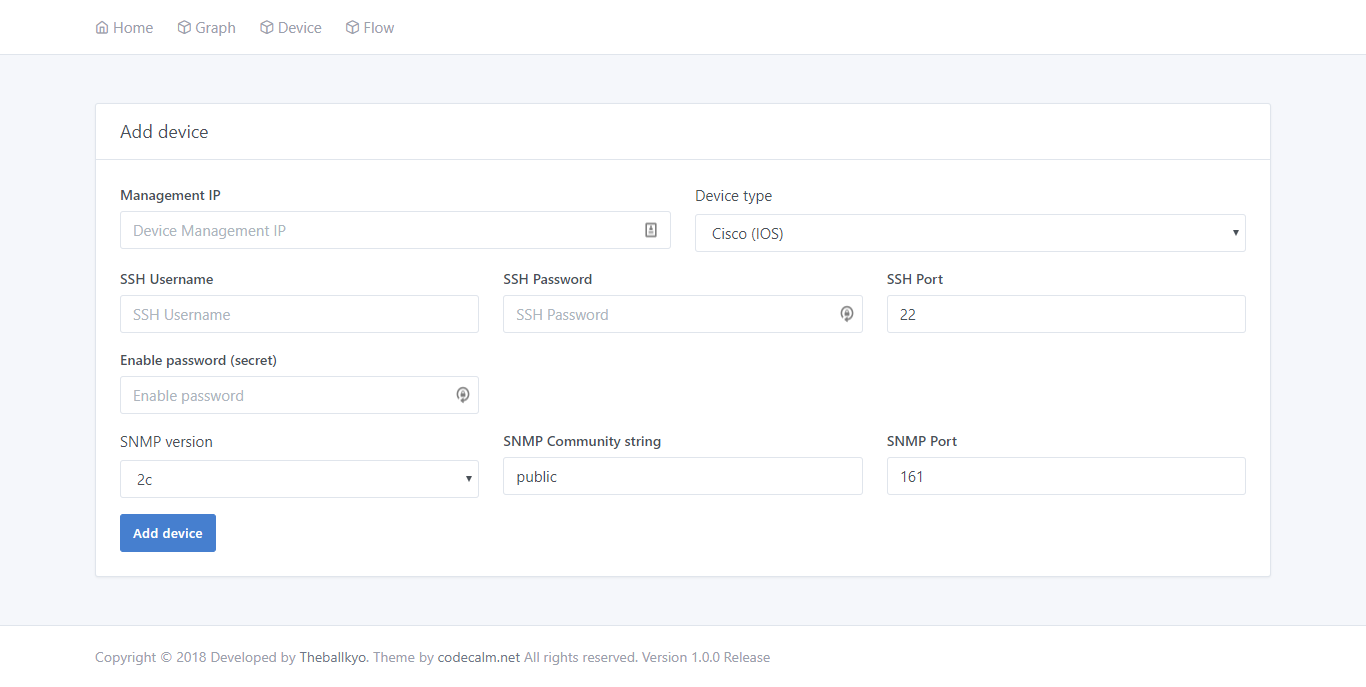
### 4.2.1 การเพิ่มอุปกรณ์

เลือก Device > Add new

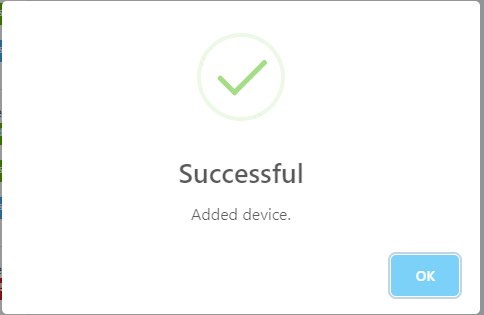


รูปที่ 4.6 เมนูการเพิ่มอุปกรณ์

หลังจากนั้นให้ใส่ข้อมูลของอุปกรณ์ แล้วกด Add device



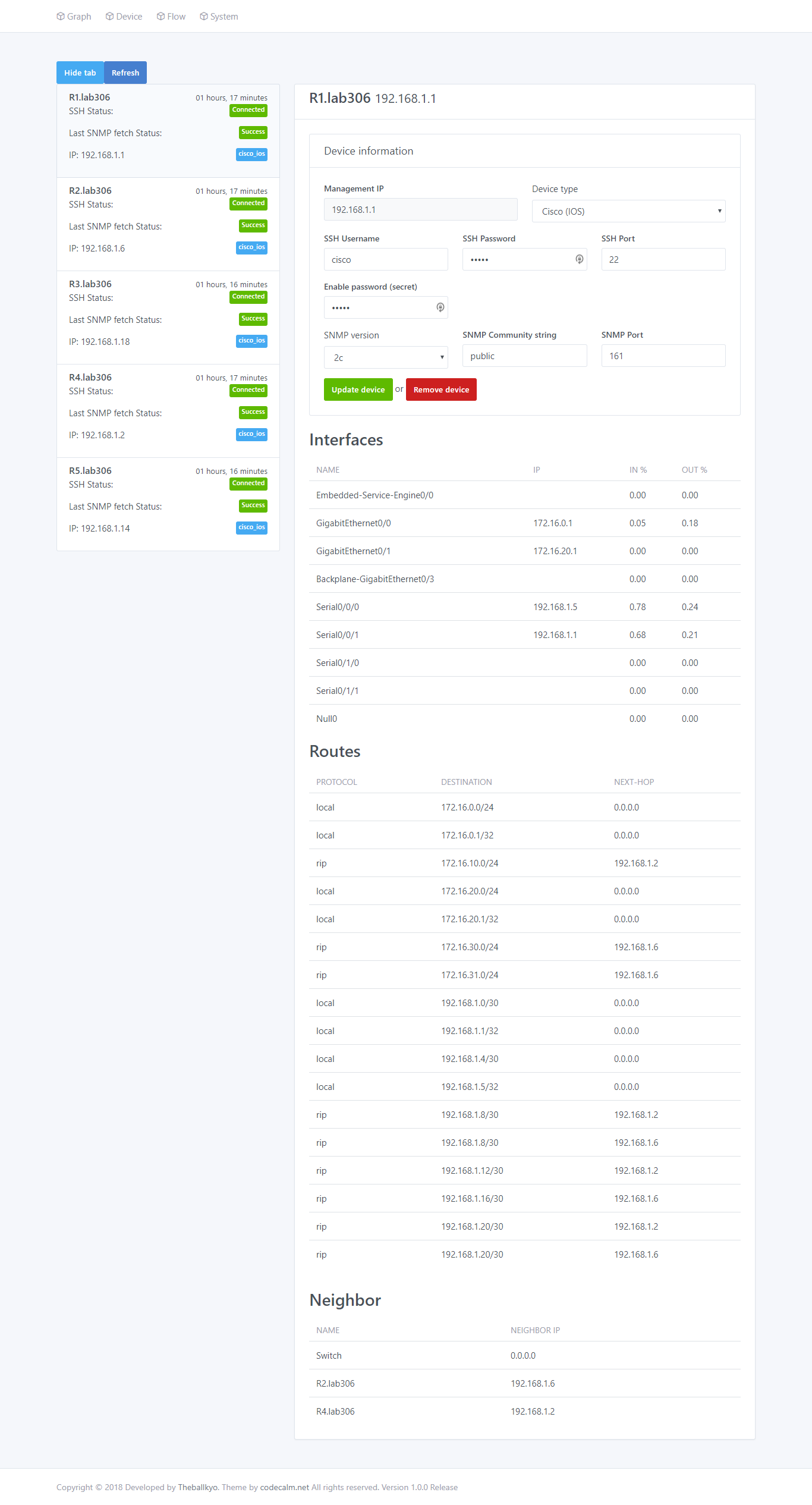
รูปที่ 4.7 หน้าจอแสดงการเพิ่มอุปกรณ์



รูปที่ 4.8 หน้าจอแสดงเมื่อเพิ่มอุปกรณ์สำเร็จ

### 4.2.2 การแสดงข้อมูลอุปกรณ์

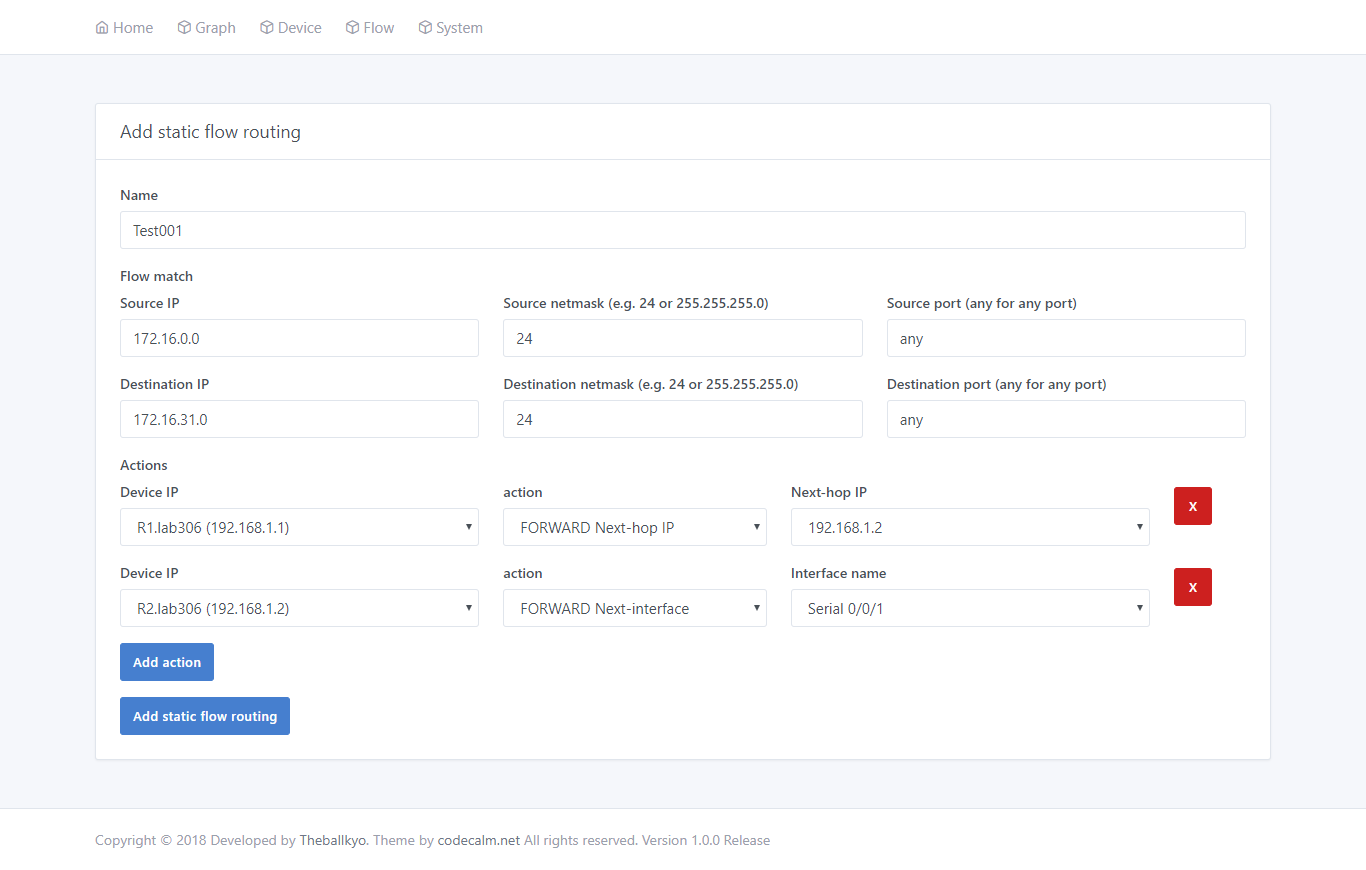
ในการทดลองนี้จะเป็นการแสดงข้อมูลอุปกรณ์ โดยจะมีข้อมูล อินเตอร์เฟส, ข้อมูลเส้นทางและข้อมูลอุปกรณ์ที่เชื่อมต่ออยู่ โดยเข้าที่เมนู Device > View จากนั้นคลิกเลือกอุปกรณ์ที่ต้องการดูข้อมูลจากหน้าต่างทางด้านซ้ายมือ



รูปที่ 4.9 หน้าจอแสดงข้อมูลอุปกรณ์

### 4.2.3 การสร้างการควบคุมเส้นทางของโฟลว์

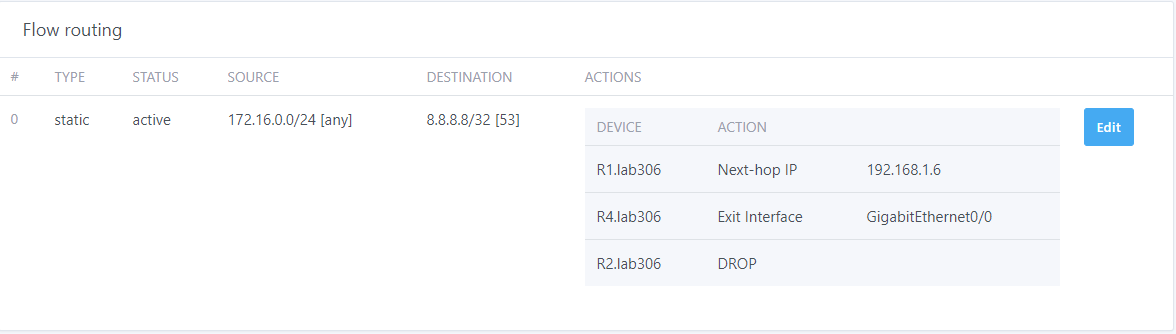
คลิกที่เมนู Flow > Add static flow routing จากนั้นกรอกข้อมูล แล้วกด Add static flow routing



รูปที่ 4.10 หน้าจอแสดงการสร้างการควบคุมเส้นทางของโฟลว์

### 4.2.4 การแสดงรายการควบคุมเส้นทางของโฟลว์

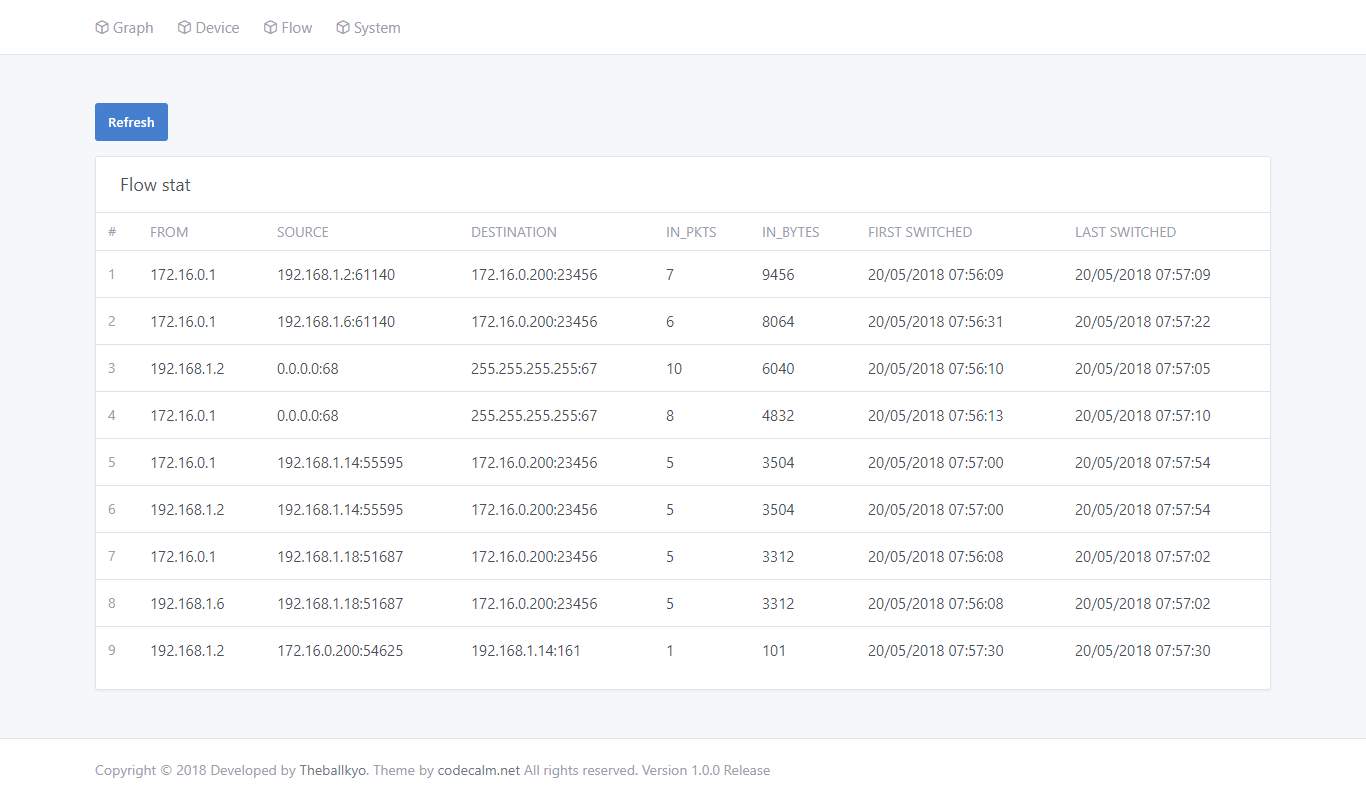
คลิกที่เมนู Flow > Flow routing โดยจะมีรายละเอียดข้อมูล Flow routing ที่ถูกสร้างขึ้นมา



รูปที่ 4.11 แสดงรายการควบคุมเส้นทางของโฟลว์

### 4.2.4 แสดงข้อมูลการใช้งานของโฟลว์

คลิกที่เมนู Flow > Flow stat



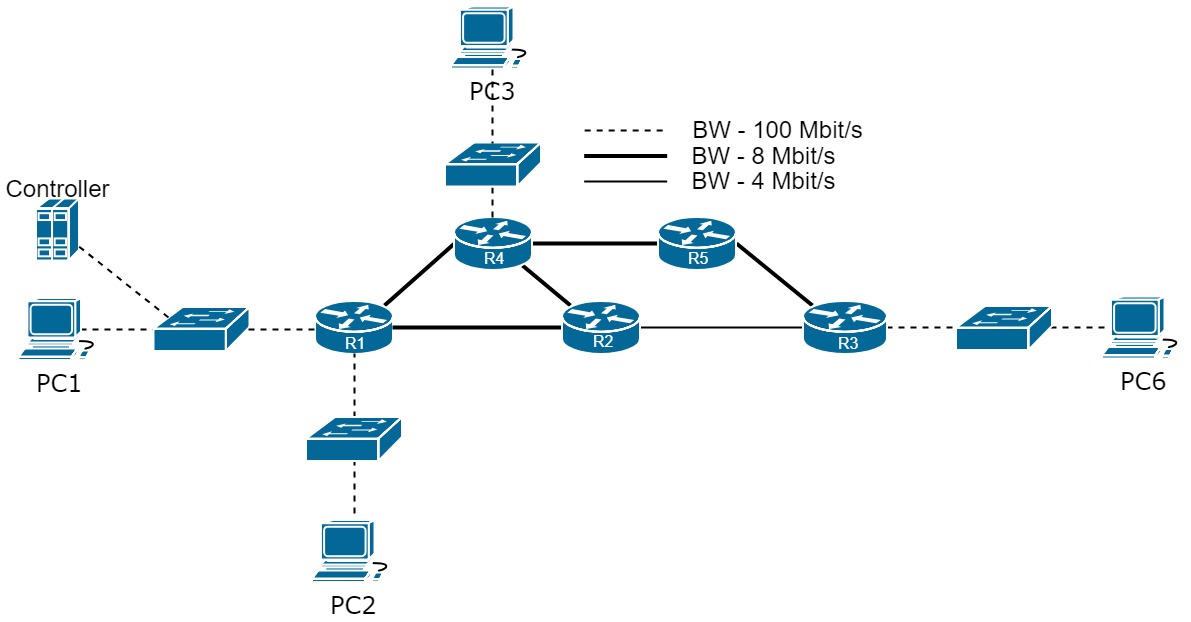
รูปที่ 4.12 หน้าแสดงข้อมูลการใช้งานของโฟลว์

## 4.3 การทดลองระบบเปลี่ยนเส้นทางแบบอัตโนมัติ

โดยการจำลองระบบเครือข่ายและสร้างสถานการณ์การส่งข้อมูลภายในระบบเครือข่ายในปริมาณมาก ๆ จนทำให้เกิดความคับคั่งขึ้นภายในเส้นทางหนึ่ง ๆ จากนั้นเปิดใช้งานระบบเพื่อทดสอบกระบวนการทำงานและผลจากการทำงานของระบบ โดยจะมีการตรวจสอบค่าปริมาณการใช้งานของลิงค์ต่าง ๆ เพื่อเปรียบเทียบความแตกต่างระหว่างก่อนและหลังการใช้งานระบบ โดยการทดสอบจะแบ่งเป็น 2 รอบดังนี้

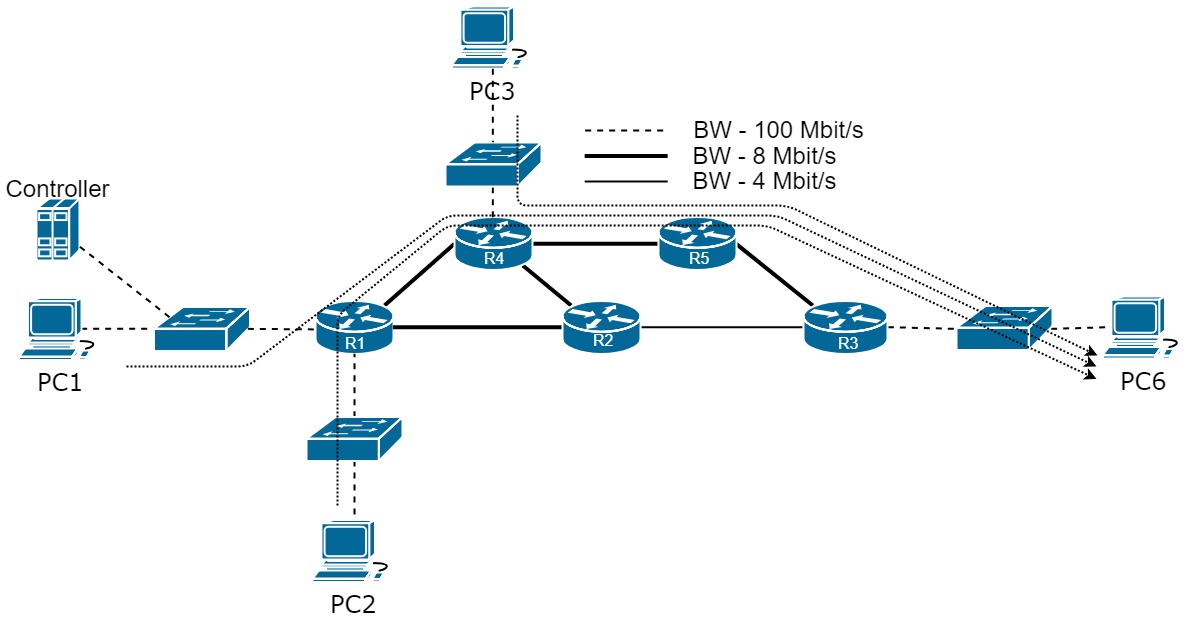
### 4.3.1 การทดลองรูปแบบที่ 1

ในการทดลองนี้ จะใช้ PC ทั้งหมด 4 PC โดยมี PC1, PC2, PC3 และ PC6 และอุปกรณ์เครือข่ายจำนวน 5 ตัวโดยมี R1, R2, R3, R4 และ R5 ซึ่งระบบถูกติดตั้งอยู่ที่ Controller ที่เชื่อมต่ออยู่กับ R1 โดยลักษณะการเชื่อมต่อจะเป็นดังรูปที่ 4.13



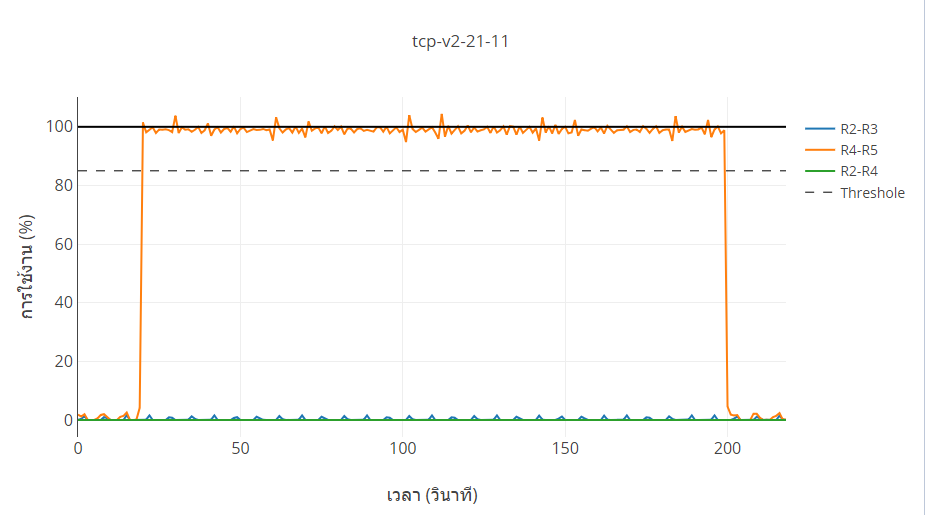
รูปที่ 4.13 แสดงแผนภาพการเชื่อมต่อของเครือข่ายการทดลองที่ 1

ในการทดลองจะกำหนดให้ PC1, PC2 และ PC3 ทำการส่งข้อมูลไปยัง PC6 พร้อมกันเป็นเวลา 180 วินาทีและมีการกำหนดปริมาณการส่งข้อมูลของ PC ไว้ที่เครื่องละ 3 Mbit/s ซึ่งเส้นทางการส่งข้อมูลก่อนใช้งานระบบจะเป็นไปดังรูปที่ 4.14



รูปที่ 4.14 เส้นทางการส่งข้อมูลของการทดลองที่ 1 ก่อนใช้งานระบบ

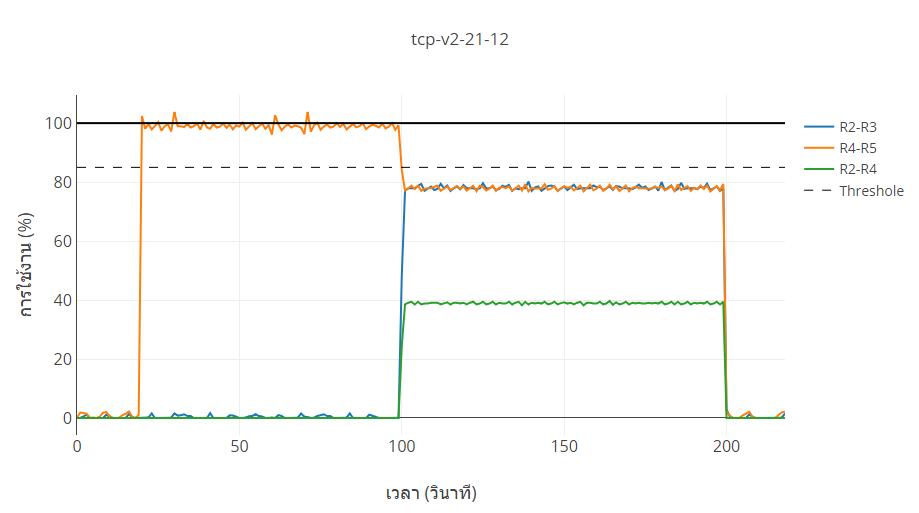
โดยมีการวัดผลการใช้งานลิงค์ที่เชื่อมต่อกันระหว่างอุปกรณ์เครือข่าย ที่เป็น bottleneck link ของเส้นทางที่ใช้ส่งข้อมูลไปยัง PC6 ด้วยได้ดังนี้



รูปที่ 4.15 กราฟแสดงการใช้งานของลิงค์ระหว่างเราส์เตอร์ก่อนใช้งานระบบ

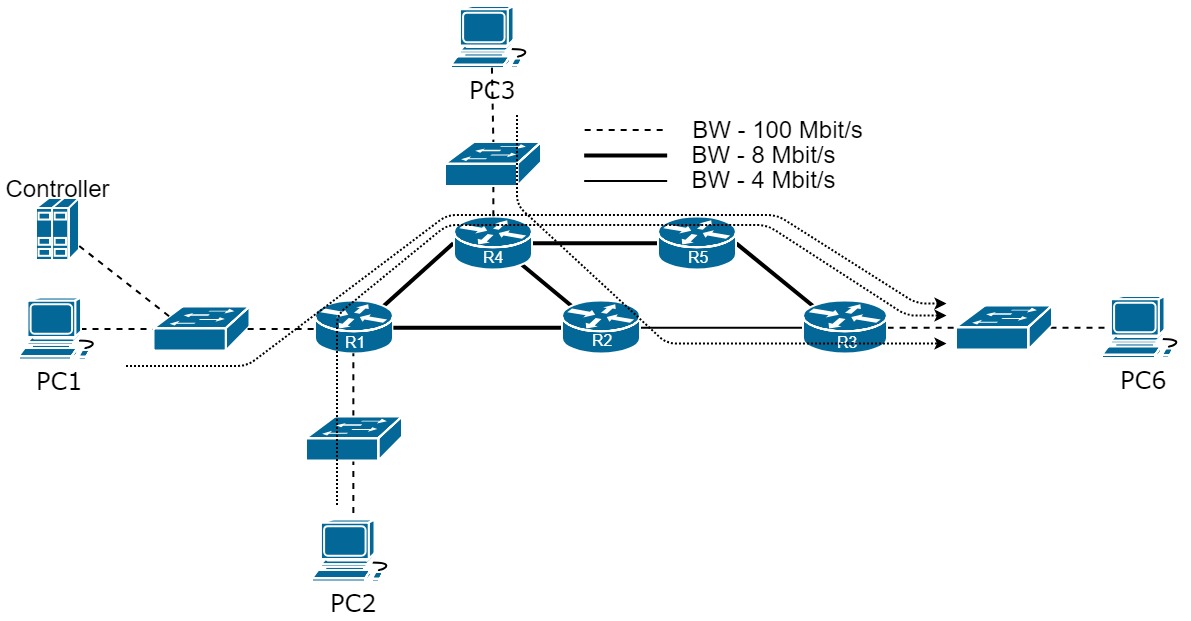
จะเห็นว่ามีการใช้เส้นทางที่ไปยัง R6 ผ่านเส้นทาง R1, R4, R5, R3 สำหรับ PC1 และ PC2 ส่วน PC 3 จะใช้เส้นทาง R4, R5 และ R5 จึงทำให้มีการใช้งานที่สูงในลิงค์ระหว่าง R4 และ R5

หลังจากนั้นทำการเปิดใช้งานโปรแกรมและทำการทดลองแบบเดิมอีกรอบ โดยผลที่ได้หลังจากการเปิดใช้งานโปรแกรมได้ดังนี้



รูปที่ 4.16 กราฟแสดงการใช้งานของลิงค์ระหว่างเราส์เตอร์หลังใช้งานระบบ

จากกราฟจะเห็นว่าช่วงเวลาวินาทีที่ 100 ระบบมีการย้ายเส้นทางจากบางส่วนให้ไปใช้งานเส้นทาง R4, R2 และ R3 เพื่อลดความคับคั่งของเส้นทาง R4, R5 และ R3 ส่งผลใช้เปอร์เซ็นต์การใช้งานมีการลดลง โดยเส้นทางจากการที่ระบบได้ทำการย้ายระหว่างเกิดความคับคั่งเป็นดังนี้

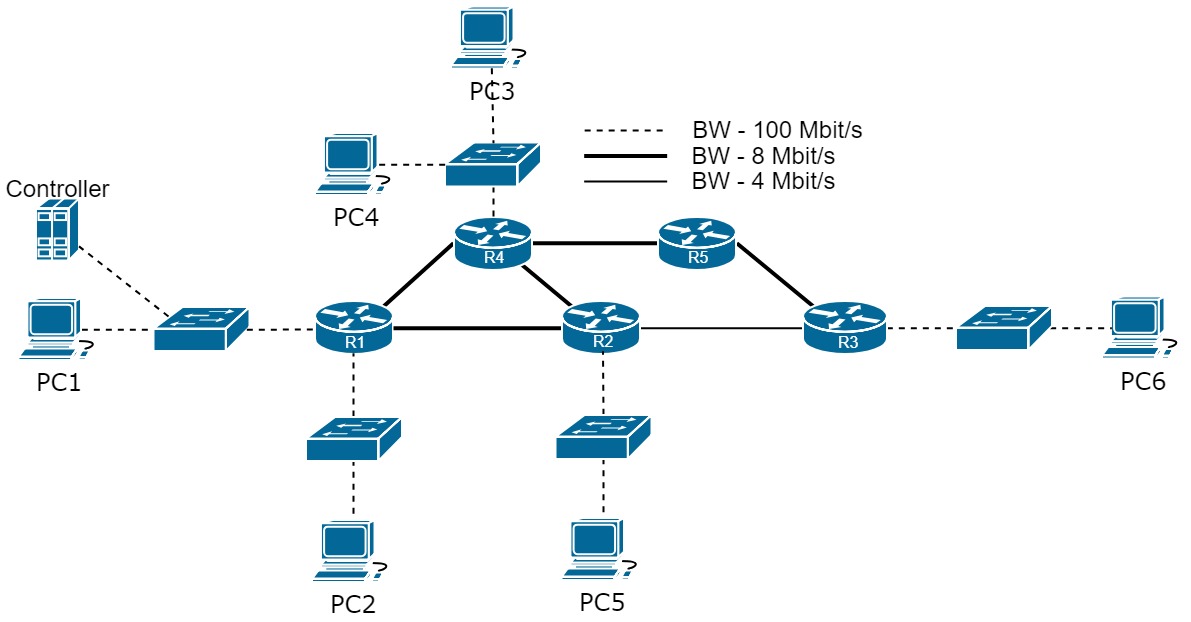


รูปที่ 4.17 เส้นทางการส่งข้อมูลขณะที่ระบบได้ย้ายเส้นทาง

จากรูปเส้นทางจะเห็นว่าระบบได้มีการย้ายเส้นทางจาก PC3 ไปยัง PC6 จากเดิมที่ใช้งานเส้นทาง R4, R5 และ R3 ไปใช้เส้นทาง R4, R2 และ R3

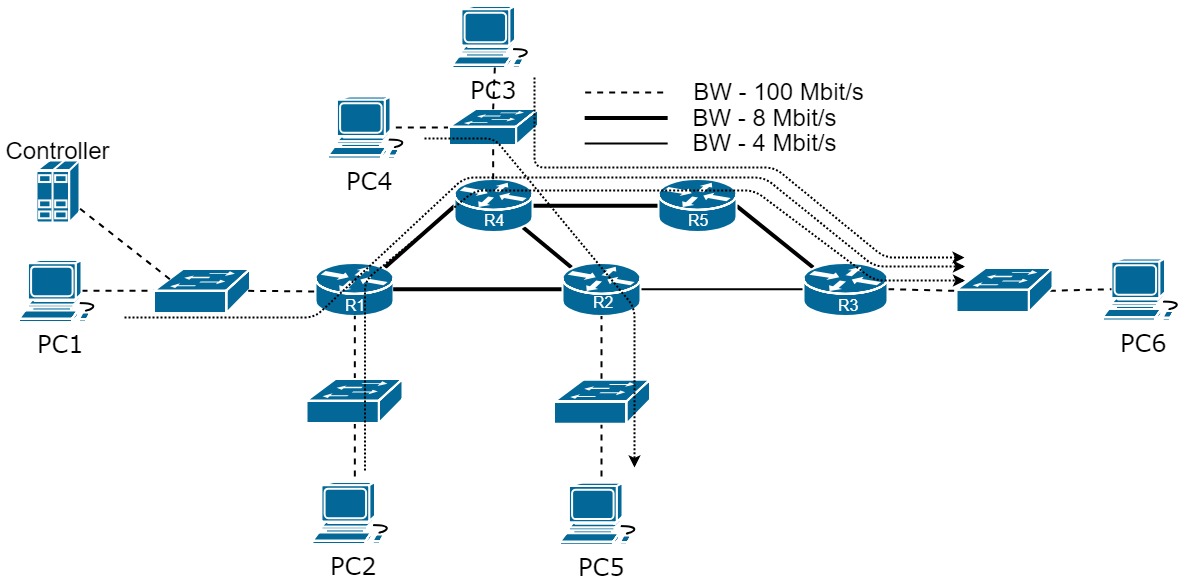
### 4.3.2 การทดลองรูปแบบที่ 2

ในการทดลองนี้ จะใช้ PC ทั้งหมด 6 PC โดยมี PC1, PC2, PC3, PC4, PC5 และ PC6 และอุปกรณ์เราส์เตอร์จำนวน 5 ตัวโดยมี R1, R2, R3, R4 และ R5 มีการใช้งาน routing protocol OSPF และมีตัวระบบรันอยู่ที่ Controller ที่เชื่อมต่ออยู่กับ R1 โดยการเชื่อมต่อกันจะเป็นดังรูปที่ 4.18



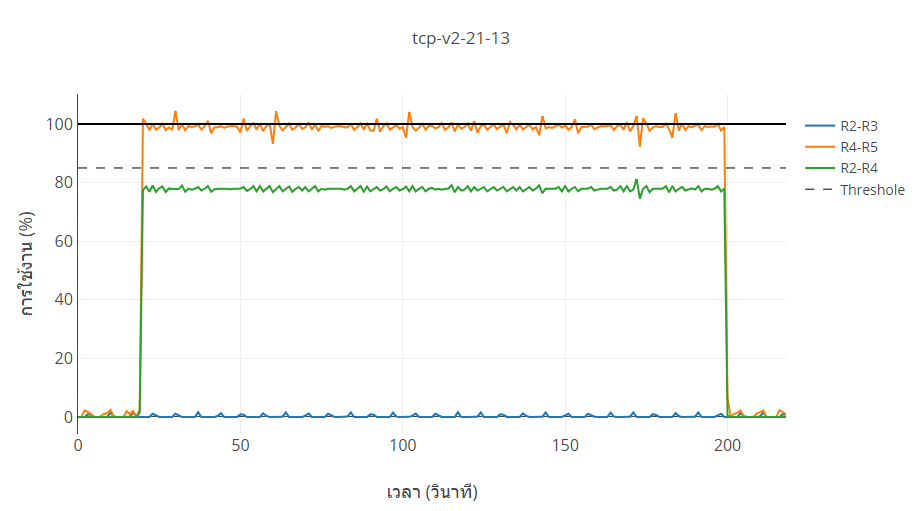
รูปที่ 4.18 แสดงแผนภาพการเชื่อมต่อของเครือข่ายการทดลองที่ 2

ในการทดลองจะมีการให้ PC1, PC2 และ PC3 ทำการส่งข้อมูลไปยัง PC6 พร้อมกันเป็นเวลา 180 วินาที โดยมีการกำหนดปริมาณการส่งข้อมูลไว้ที่เครื่องละ 3 Mbit/s และให้ PC4 ส่งข้อมูลไปยัง PC5 โดยมีการกำหนดปริมาณการส่งข้อมูลไว้ที่ 6 Mbit/s โดยเส้นทางการส่งข้อมูลก่อนใช้งานโปรแกรมจะเป็นได้ดังรูปที่ 4.19



รูปที่ 4.19 เส้นทางการส่งข้อมูลของการทดลองที่ 1

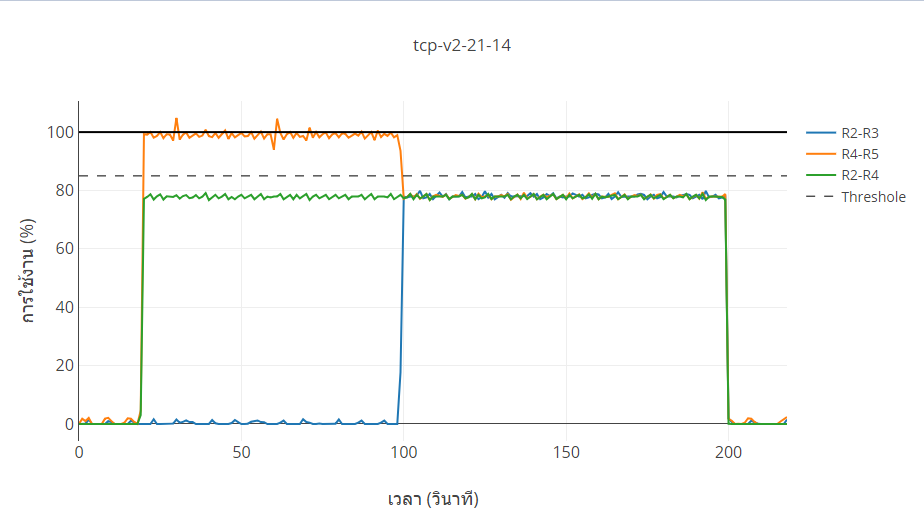
โดยมีการวัดผลการใช้งานลิงค์ที่เชื่อมต่อกันระหว่างเราสเตอร์ ที่เป็น bottleneck link ของเส้นทางที่ใช้ส่งข้อมูลไปยัง PC6 ด้วยได้ดังนี้



รูปที่ 4.20 กราฟแสดงการใช้งานของลิงค์ระหว่างเราส์เตอร์ก่อนใช้งานระบบ

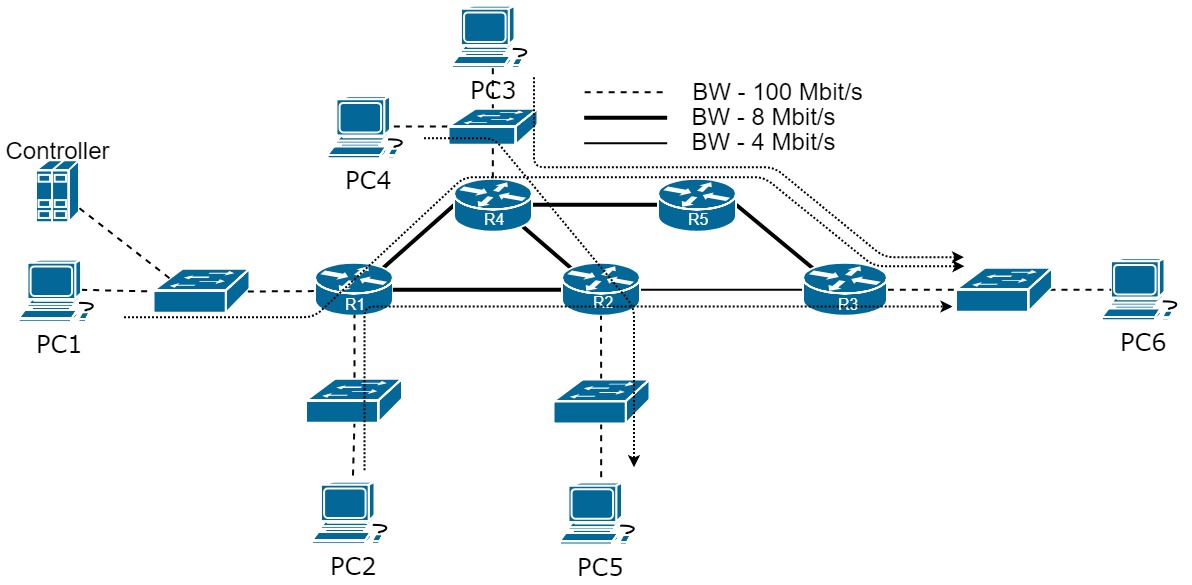
จะเห็นว่ามีการใช้เส้นทางที่ไปยัง R6 ผ่านเส้นทาง R1, R4, R5, R3 สำหรับ PC1 และ PC2 ส่วน PC 3 จะใช้เส้นทาง R4, R5 และ R5 จึงทำให้มีการใช้งานที่สูงในลิงค์ระหว่าง R4 และ R5 และมีการใช้งานประมาณ 78 – 80 เปอร์เซ็นต์สำหรับลิงค์ระหว่าง R2 และ R4 ซึ่งเป็นเส้นทางที่ใช้ในการส่งข้อมูลจาก PC4 ไปยัง PC5

หลังจากนั้นทำการเปิดใช้งานโปรแกรมและทำการทดลองแบบเดิมอีกรอบ โดยผลที่ได้หลังจากการเปิดใช้งานโปรแกรมได้ดังนี้



รูปที่ 4.21 กราฟแสดงการใช้งานของลิงค์ระหว่างเราส์เตอร์หลังใช้งานระบบ

จากกราฟจะเห็นว่าช่วงเวลาวินาทีที่ 100 ระบบมีการย้ายเส้นทางจากบางส่วนให้ไปใช้งานเส้นทางที่ผ่านลิงค์ระหว่าง R2 และ R3 เพื่อลดความคับคั่งของลิงค์ระหว่าง R4 และ R5 ส่งผลใช้เปอร์เซ็นต์การใช้งานมีการลดลง โดยเส้นทางจากการที่ระบบได้ทำการย้ายระหว่างเกิดความคับคั่งเป็นดังนี้



รูปที่ 4.22 เส้นทางการส่งข้อมูลขณะที่ระบบได้ย้ายเส้นทาง

จากรูปเส้นทางจะเห็นว่าระบบได้มีการย้ายเส้นทางจาก PC1 ไปยัง PC6 จากเดิมที่ใช้งานเส้นทาง R1, R4, R5 และ R3 ไปใช้เส้นทาง R1, R2 และ R3 โดยที่การคับคั่งของเส้นทางได้เกิดขึ้นที่ลิงค์ระหว่าง R4 และ R5 โดยที่เลือกเส้นทางนี้เนื่องจากเส้นทาง R4, R2 และ R3 มีการใช้งานอยู่ที่ค่อนข้างสูง ที่ลิงค์ระหว่าง R2 และ R4 ไม่เพียงพอต่อการที่จะย้ายไปได้

# บทที่ 5 วิเคราะห์และสรุปผล

## 5.1 สรุปผลการดำเนินงาน

ระบบจัดการเครือข่าย จะตรวจสอบข้อมูลจากอุปกรณ์เครือข่ายที่ผู้ใช้งานได้เพิ่มเข้าไปในระบบได้แบบอัตโนมัติ ผู้ใช้งานสามารถแก้ไขเส้นทางการจราจรในเครือข่ายได้ผ่านระบบโดยตรง สามารถตรวจสอบข้อมูลการใช้งานของอินเตอร์เฟสและข้อมูลการกำหนดเส้นทางของอุปกรณ์ได้

และในการทดลองเปลี่ยนเส้นทางโดยอัตโนมัติ ผลลัพธ์ที่ได้เป็นไปตามที่ได้คิดไว้คือ มีการเปลี่ยนเส้นทางได้เองในกรณีที่มีการคับคั่งเกิดขึ้น

## 5.2 ปัญหาและอุปสรรคที่พบในการพัฒนาระบบ

การพัฒนาระบบนี้โดยส่วนใหญ่การทำงานจะเป็นการทำงานแบบการประมวลหลาย ๆอย่างพร้อมกัน (Multiprocessing) ซึ่งมีปัญหาในการควบคุมและส่งข้อมูลหากัน ซึ่งทางผู้พัฒนาไม่ค่อยมีประสบการณ์การพัฒนาระบบในลักษณะนี้เท่าที่ควร จึงทำให้ในช่วงแรกของการพัฒนาระบบมีปัญหาระบบไม่สามารถทำงานได้ดังที่คาดหวัง การสั่งงานอุปกรณ์เครือข่ายมีการทดลองและหาคำสั่งที่จำเป็นที่จะให้อุปกรณ์เครือข่ายทำงานได้ตามที่ได้วางแผนไว้ค่อนข้างใช้เวลานานกว่าที่คาดการไว้จึงทำให้การทำงานความล่าช้า และการออกแบบตัวระบบที่อิงกับสถาปัตยกรรม SDN นั้นใช้เวลานานกว่าที่คาดการณ์ไว้

## 5.3 ข้อจำกัดของระบบ

1. ในการสั่งควบคุมอุปกรณ์เพื่อทำการเปลี่ยนเส้นทางนั้นระบบจะทำการแก้ไขอุปกรณ์ทีละตัวซึ่งทำให้เกิดความล่าช้า
2. การสร้างเส้นทางนั้นระบบสามารถสร้างได้สูงสุดเพียง 65536 เส้นทางเนื่องจากเป็นข้อจำกัดของอุปกรณ์ที่ทางผู้จัดทำนำมาทดสอบระบบ
3. ในขั้นตอนการเก็บข้อมูลผ่านทาง SNMP ตัวระบบมีการใช้ทรัพยากรมาก
4. ระบบยังไม่สามารถตรวจสอบโฟลว์ที่มีระยะเวลาการเกิดสั้นๆได้ เนื่องจากมีข้อจำกัดจาก NetFlow ซึ่งไม่สามารถส่งข้อมูลมาให้ระบบแบบทันทีได้

## 5.4 ข้อเสนอแนะและแนวทางในการพัฒนาระบบในอนาคต

ในอนาคตจะมีการปรับปรุงระบบให้สามารถสั่งงานผ่านทางเว็บเบราวเซอร์แบบวิชวลได้สมบูรณ์แบบมากกว่านี้ ซึ่งจะทำให้การใช้งานระบบมีความง่ายมากขึ้น สามารถวิเคราะห์ปริมาณการใช้งานของโฟลว์ที่เป็นโฟลว์เล็กๆได้ เพื่อสามารถเลือกโฟลว์ที่ดีกว่าเดิมในการย้ายได้

บรรณานุกรม

[1] Open Networking Foundation. (2012). **Software-Defined Networking: The New Norm for Networks**. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.

[2] Raj Jain. (2013). **Introduction to Software Defined Networking(SDN)**. Available: https://www.cse.wustl.edu/~jain/cse570-13/ftp/m\_16sdn.pdf.

[3] Xenofon Foukas, Mahesh K. Marina, Kimon Kontovasilis. **Software Defined Networking Concepts**. Available: http://homepages.inf.ed.ac.uk/mmarina/papers/sdn-chapter.pdf.

[4] Awduche, et. al. **Overview and Principles of Internet Traffic Engineering**. Available: https://tools.ietf.org/html/rfc3272 .

[5] Tom Scholl. (2014). **Internet Routing and Traffic Engineering**. Available: https://aws.amazon.com/blogs/architecture/internet-routing-and-traffic-engineering/.

[6] Zhaogang shu, Jiafu wan, Jiaxiang lin, Shiyong wang, Di li, Seungmin rho and Changcai yang. **Traffic Engineering in Software-Defined Networking: Measurement and**

**Management**. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7496952.

[7] Cisco. (2012). **Introduction to Cisco IOS NetFlow – A Technical Overview**. Available: https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod\_white\_paper0900aecd80406232.html.

[8] Cisco. **Configuring Simple Network Management Protocol (SNMP)**. Available: https://www.cisco.com/c/en/us/td/docs/app\_ntwk\_services/data\_center\_app\_services/css11500series/v8-20/configuration/administration/guide/admgd/SNMP.html.

[9] Justin Ellingwood. 2014. **An Introduction to SNMP (Simple Network Management** **Protocol)**. Available: https://www.digitalocean.com/community/tutorials/an-introduction-to-snmp-simple-network-management-protocol.

[10] Rowell Dionicio. **Cisco NDP – Neighbor Discovery Protocol – What Is It?**.

https://www.packet6.com/cisco-ndp-neighbor-discovery-protocol/.

**บรรณานุกรม (ต่อ)**

[11] Cisco. (2016). **Cisco Discovery Protocol Version 2**.

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cdp/configuration/15-mt/cdp-15-mt-book/nm-cdp-discover.html.

[12] Cisco. (2008). **Cisco Discovery Protocol**.

https://www.cisco.com/c/en/us/td/docs/routers/access/3200/software/wireless/3200WirelessConfigGuide/CiscoDiscovProto.pdf.

[13] Cisco. **Routing Information Protocol**. https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960x/software/15-2\_5\_e/configuration\_guide/b\_1525e\_consolidated\_2960x\_cg/b\_1525e\_consolidated\_2960x\_cg\_chapter\_01011110.pdf.

[14] Cisco. **OSPF Design Guide**. https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html#t9.

[15] Cisco. (2014). **Configuring Policy-Based Routing**. https://www.cisco.com/c/en/us/td/docs/ios/12\_2/qos/configuration/guide/fqos\_c/qcfpbr.pdf.

[16] Cisco. (2005). **Understanding Policy Routing**. https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/10116-36.html.

[17] Fabio Semperboni. (2013). **PBR: Route a packet based on source IP address**. http://www.ciscozine.com/pbr-route-a-packet-based-on-source-ip-address/.

[18] Cisco. **Route Maps**. https://www.cisco.com/c/en/us/td/docs/security/asa/asa94/config-guides/cli/general/asa-94-general-config/routes-maps.pdf.

[19] www.ssh.com. **SSH (SECURE SHELL)**. https://www.ssh.com/ssh/

[20] Mohammad Al-Fares, Barath Raghavan, Sivasankar Radhakrishnan, Nelson Huang and Amin Vahdat. **Hedera: Dynamic Flow Scheduling for Data Center Network**. https://www1.icsi.berkeley.edu/~barath/papers/hedera-nsdi10.pdf.

ภาคผนวก

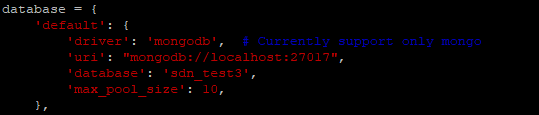
ภาคผนวก ก

วิธีการใช้งานระบบ

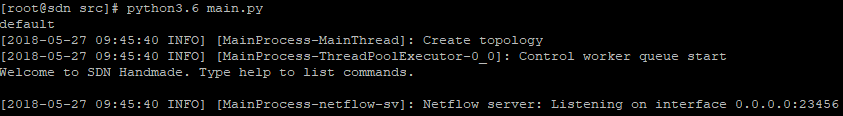
**ขั้นตอนการใช้งานระบบ**

ในการใช้งานระบบจำเป็นจะต้องติดตั้งระบบบนเครื่องเซิฟเวอร์ที่ทำงานอยู่บนระบบปฏิบัติการ linux เท่านั้น และจำเป็นต้องติดตั้งระบบฐานข้อมูล mongodb ซึ่งจะอยู่ที่ภายในเครื่องเซิฟเวอร์เดียวกันกับระบบหรืออยู่บนเซิฟเวอร์เครื่องอื่นก็ได้ โดยขั้นตอนในการติดตั้งระบบมีดังนี้

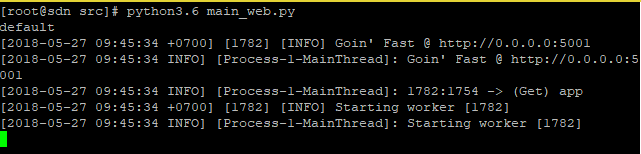
1. เข้าไปที่โฟลเดอร์ที่จัดเก็บโค้ดต้นฉบับ (source code) และเข้าไปที่โฟลเดอร์ SDN-handmade/backend/src หลังจากนั้นทำการแก้ไขไฟล์ settings.py เพื่อตั้งค่าในส่วน database ให้ตรงตามที่ติดตั้งไว้



รูปที่ ก.1 แสดงตัวอย่างการตั้งค่า Database

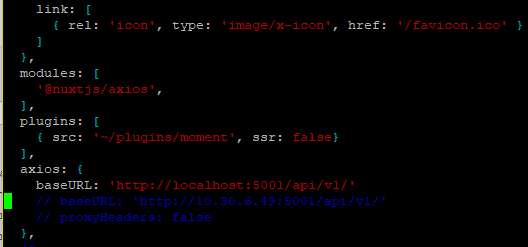
1. ติดตั้งไลบารี่ที่จำเป็นสำหรับการทำงานของระบบทั้งหมด โดยพิมพ์ “pip install -r requirement.txt” ที่หน้า Command Line Interface
2. รันโปรแกรมที่โฟลเดอร์ SDN-handmade/backend/src จำนวน 2 ไฟล์คือไฟล์ main.py และ main\_web.py

รูปที่ ก.2 แสดงการรันไฟล์ main.py



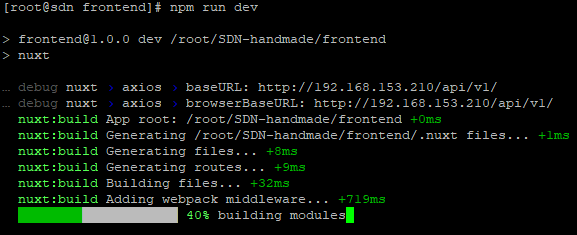
รูปที่ ก.3 แสดงการรันไฟล์ main\_web.py

1. เข้าไปที่โฟลเดอร์ SDN-handmade/frontend และทำการแก้ไขไฟล์ nuxt.config.js โดยแก้ไขในส่วน baseURL ให้เป็น url ของเครื่องเซิฟเวอร์ที่รันไฟล์ main\_web.py ในข้อที่ 3



รูปที่ ก.4 แสดงการตั้งค่า baseURL

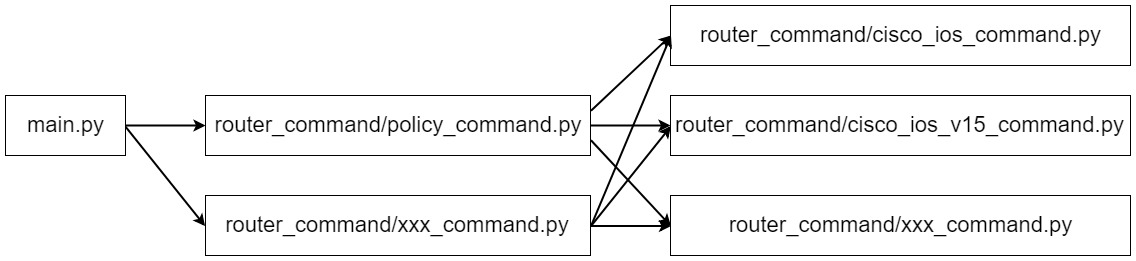
1. ติดตั้งไลบารี่ที่จำเป็นต่อการทำงานของระบบ โดยพิมพ์คำสั่ง “npm install”
2. รันระบบส่วนของ Frontend โดยพิมพ์คำสั่ง “npm run dev”



รูปที่ ก.5 แสดงการรันระบบส่วนของ Frontend

2. ขั้นตอนการเพิ่มอุปกรณ์ชนิดใหม่เข้าสู่ระบบ

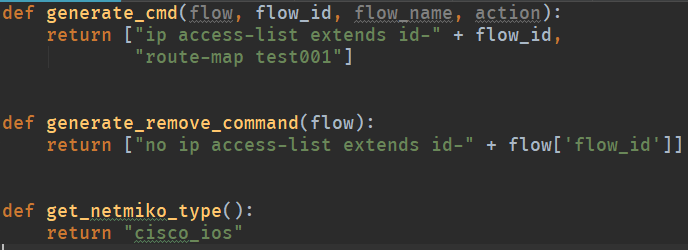
โครงสร้างของไฟล์ที่เกี่ยวข้องกับการเพิ่มอุปกรณ์ชนิดใหม่เข้าสู่ระบบ มีลักษณะดังนี้



รูปที่ ก.6 แสดงโครงสร้างของไฟล์สำหรับการเพิ่มอุปกรณ์ชนิดใหม่เข้าสู่ระบบ

โดยการเพิ่มอุปกรณ์ชนิดใหม่เข้าสู่ระบบมีขั้นตอนดังนี้

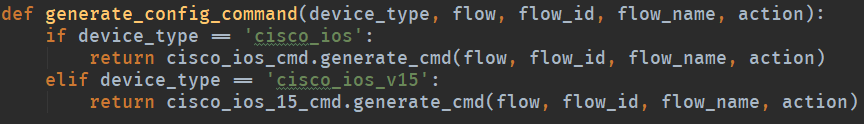
1. สร้างไฟล์ python (.py) ที่จะใช้ในการสร้างคำสั่ง (command) เพื่อควบคุมอุปกรณ์เครือข่ายประเภทนั้น ๆ โดยสร้างไฟล์เอาไว้ที่โฟลเดอร์ router\_command จากนั้นให้สร้างฟังก์ชันดังต่อไปนี้
   * generate\_cmd ซึ่งเป็นฟังก์ชันที่ใช้ในการสร้างคำสั่งสำหรับควบคุมการทำงานของอุปกรณ์เครือข่าย โดยกำหนดให้ฟังก์ชันรับค่า argument 4 ค่าคือ flow, flow\_id, flow\_name และ action พร้อมให้ฟังก์ชั่นทำการคืนค่า คำสั่งสำหรับการควบคุมการทำงานของอุปกรณ์ออกมามาเป็นข้อมูลประเภท list ของ string
   * generate\_remove\_command ซึ่งเป็นฟังก์ชันสำหรับการสร้างคำสั่งเพื่อยกเลิกคำสั่งสำหรับการควบคุมการทำงานของอุปกรณ์เครือข่าย โดยกำหนดให้ฟังก์ชันรับค่า argument 1 ค่าคือ flow พร้อมให้ฟังก์ชั่นทำการคืนค่าคำสั่งออกมาเป็นข้อมูลประเภท list ของ string
   * get\_netmiko\_type ซึ่งเป็นฟังก์ชันสำหรับบอกประเภทของอุปกรณ์เครือข่ายที่จะทำการควบคุม โดยกำหนดให้ฟังก์ชันคืนค่าประเภทของอุปกรณ์ออกมาเป็น string (netmiko เป็นไลบารี่สำหรับช่วยในการสร้างการเชื่อมต่อ Secure Shell ไปยังอุปกรณ์เครือข่าย ซึ่งสามารถดูข้อมูลอุปกรณ์เครือข่ายที่รองรับการใช้งานได้ที่ https://github.com/ktbyers/netmiko)



รูปที่ ก.7 แสดงการสร้างฟังก์ชันสำหรับการเพิ่มอุปกรณ์ชนิดใหม่เข้าสู่ระบบ

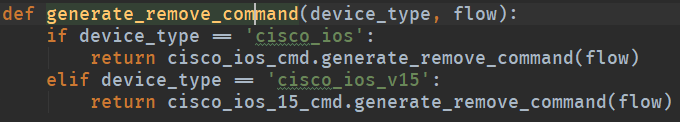
1. เปิดไฟล์ policy\_command.py ที่โฟลเดอร์ router\_command จากนั้นแก้ไขฟังก์ชันดังต่อไปนี้

* generate\_config\_command ซึ่งเป็นฟังก์ชันที่จะเรียกใช้งานฟังก์ชันสำหรับการสร้างคำสั่งควบคุมอุปกรณ์เครือข่ายตามประเภทของอุปกรณ์เครือข่าย โดยใช้การตรวจสอบเงื่อนไข if elif ในการตรวจสอบประเภทของอุปกรณ์ เมื่อประเภทของอุปกรณ์ตรงตามเงื่อนไขก็จะเรียกใช้งานฟังก์ชันสำหรับการสร้างคำสั่งที่ตรงกับอุปกรณ์นั้น ๆ เมื่อเพิ่มอุปกรณ์ชนิดใหม่เข้าสู่ระบบก็สร้างเงื่อนไขในการตรวจสอบเพิ่มเติม



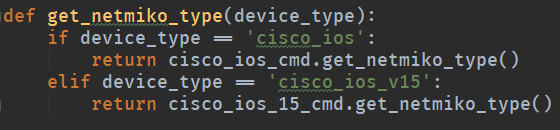
รูปที่ ก.8 แสดงข้อมูลในฟังก์ชัน generate\_config\_command

* generate\_remove\_command เป็นฟังก์ชันที่มีลักษณะคล้ายกับฟังก์ชันก่อนหน้า แต่แตกต่างตรงที่ฟังก์นี้ใช้ตรวจสอบประเภทของอุปกรณ์เครือข่าย เพื่อสร้างคำสั่งสำหรับการยกเลิกการควบคุมการทำงานของอุปกรณ์



รูปที่ ก.9 แสดงข้อมูลในฟังก์ชัน generate\_remove\_command

* get\_netmiko\_type เป็นฟังก์ชันสำหรับตรวจสอบประเภทอุปกรณ์เพื่อให้เรียกฟังก์ชันที่สร้างขึ้นมา เมื่อเพิ่มอุปกรณ์เครือข่ายก็เพิ่มเงื่อนไขในการตรวจสอบประเภทของอุปกรณ์



รูปที่ ก.10 แสดงข้อมูลในฟังก์ชัน get\_netmiko\_type

ภาคผนวก ข

วิธีการใช้งาน REST API

**วิธีการใช้งาน REST API**

การเรียกใช้ api นั้นสามารถทำได้โดยการสร้าง http request และใส่ url เป็นของเซิฟเวอร์ api ซึ่งจะมีลักษณะดังนี้ http://<server\_ip>:5001/api/v1<api> โดยรูปแบบการ request ที่จะอธิบายด้านล่างจะอยู่ในรูป <http\_method> <url> เช่น GET /device คือให้ใช้ http method GET ในการเรียกข้อมูลที่ url http:<server\_ip>:5001/api/v1/device และใน url ข้อความที่มีตัวอักษร : นำหน้าเช่น :device\_id ให้แทนเป็นข้อมูลของค่านั้น ๆ แทน และในส่วนของการ response จะเป็น http\_status พร้อมทั้งรูปแบบข้อมูลตัวอย่าง

ตารางที่ ข.1 แสดงรายชื่อของ API ทั้งหมด

|  |  |  |
| --- | --- | --- |
|  | URL | คำอธิบาย |
| 1 | GET /device | ขอข้อมูลอุปกรณ์ทั้งหมด |
| 2 | POST /device | เพิ่มอุปกรณ์เข้าสู่ระบบ |
| 3 | PATCH /device/:device\_id | แก้ไขข้อมูลอุปกรณ์ |
| 4 | DELETE /device?device\_id=:device\_id | ลบข้อมูลอุปกรณ์ |
| 5 | GET /device/:device\_id | ขอข้อมูลอุปกรณ์ตามไอดีของอุปกรณ์ |
| 6 | GET /device/:device\_id/neighbor | ขอข้อมูลอุปกรณ์ที่เชื่อมต่อกัน |
| 7 | GET /link | ขอข้อมูลลิงค์ทั้งหมด |
| 8 | GET /link/:link\_id | ขอข้อมูลลิงค์ตามลิงค์ไอดี |
| 9 | GET /path/:from\_device\_ip,:to\_device\_ip | ข้อมูลเส้นทางจากอุปกรณ์เครื่องหนึ่งไปยังอุปกรณ์อีกเครื่องหนึ่ง |
| 10 | GET /flow | ขอข้อมูลโฟลว์ทั้งหมด |
| 11 | GET /flow/routing | ขอข้อมูลเส้นทางของโฟลว์ |
| 12 | POST /flow/routing | สร้างเส้นทางของโฟลว์ |
| 13 | PATCH /flow/routing | แก้ไขข้อมูลเส้นทางของโฟลว์ |
| 14 | DELETE /flow/routing?flow\_id=:flow\_id | ลบเส้นทางของโฟลว์ |
| 15 | GET /flow/routing/:flow\_id | ขอเส้นทางของโฟลว์ |
| 16 | GET /routes/device\_id | ขอข้อมูลตารางเส้นทางของอุปกรณ์ |

**1 การขอข้อมูลอุปกรณ์ทั้งหมด**

GET /device

**Response:** Status 200 OK

ตารางที่ ข. ตัวอย่างข้อมูลที่คืนค่าข้อมูลอุปกรณ์ทั้งหมด

|  |
| --- |
| {  "devices":[  {  "\_id":{  "$oid":"5af97cbf02b9c80969284ca6"  },  "management\_ip":"192.168.1.1",  "device\_ip":"192.168.1.1",  "snmp\_info":{  "version":"2c",  "community":"public",  "port":161  },  "snmp\_is\_running":false,  "snmp\_last\_run\_time":1526992492.2371502,  "ssh\_info":{  "username":"cisco",  "password":"cisco",  "port":22,  "secret":"cisco"  },  "status":1,  "type":"cisco\_ios",  "is\_ssh\_connect":true,  "cdp\_enable":true,  "description":"Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.5(3)M6, RELEASE SOFTWARE (fc2)\r\nTechnical Support: http://www.cisco.com/techsupport\r\nCopyright (c) 1986-2017 by Cisco Systems, Inc.\r\nCompiled Tue 25-Jul-17 13:17 by prod\_rel\_team",  "interfaces":[ |

ตารางที่ ข.2 (ต่อ) ตัวอย่างข้อมูลที่คืนค่าข้อมูลอุปกรณ์ทั้งหมด

|  |
| --- |
| {  "index":1,  "description":"Embedded-Service-Engine0/0",  "type":6,  "mtu":1500,  "speed":10000000,  "physical\_address":"0x000000000000",  "admin\_status":2,  "operational\_status":2,  "last\_change":4275,  "in\_octets":0,  "in\_ucast\_packets":0,  "in\_discards":0,  "in\_errors":0,  "in\_unknown\_protos":0,  "out\_octets":0,  "out\_ucast\_packets":0,  "out\_discards":0,  "out\_errors":0,  "device\_ip":"192.168.1.1",  "bw\_in\_usage\_octets":0,  "bw\_in\_usage\_persec":0.0,  "bw\_in\_usage\_percent":0.0,  "bw\_out\_usage\_octets":0,  "bw\_out\_usage\_persec":0.0,  "bw\_out\_usage\_percent":0.0,  "bw\_usage\_update":1526992492.0793455  },  ],  "interfaces\_update\_time":1526992491.794834, |

ตารางที่ ข.2 (ต่อ) ตัวอย่างข้อมูลที่คืนค่าข้อมูลอุปกรณ์ทั้งหมด

|  |
| --- |
| "name":"R1.lab306",  "updated\_at":1526992492.0795121,  "uptime":4448648,  "is\_snmp\_connect":true  },  ],  "success":true  } |

2 การเพิ่มอุปกรณ์เข้าสู่ระบบ

POST /device

ข้อมูลที่ต้องส่งไป

ตารางที่ ข.3 ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อเพิ่มอุปกรณ์

|  |
| --- |
| {  "management\_ip":"192.168.1.40",  "type":"cisco\_ios",  "ssh\_info":{  "username":"cisco",  "password":"cisco\_pass",  "port":22,  "secret":"enable\_pass"  },  "snmp\_info":{  "version":"2c",  "community":"public",  "port":161  }  } |

**Response:** Status 201 Created

ตารางที่ ข.4 ตัวอย่างข้อมูลที่คืนค่าของการเพิ่มอุปกรณ์

|  |
| --- |
| {  "success":true,  "message":{  "management\_ip":"192.168.1.40",  "type":"cisco\_ios",  "ssh\_info":{  "username":"cisco",  "password":"cisco\_pass",  "port":22,  "secret":"enable\_pass"  },  "snmp\_info":{  "version":"2c",  "community":"public",  "port":161  }  }  } |

**3 การแก้ไขข้อมูลอุปกรณ์**

PATCH /device/:device\_id

ข้อมูลที่ต้องส่งไป

ตารางที่ ข.5 ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อแก้ไขอุปกรณ์

|  |
| --- |
| {  "management\_ip":"192.168.1.1",  "type":"cisco\_ios",  "ssh\_info":{  "username":"cisco",  "password":"cisco",  "port":22,  "secret":"cisco"  },  "snmp\_info":{  "version":"2c",  "community":"public",  "port":161  },  "\_id":"5af97cbf02b9c80969284ca6"  } |

**Response:** Status 200 OK

ตารางที่ ข.6 ตัวอย่างข้อมูลที่คืนค่าของการแก้ไขอุปกรณ์

|  |
| --- |
| {  "status":true,  "message":"Update device!"  } |

**4 การลบข้อมูลอุปกรณ์**

DELETE /device?device\_id=:device\_id

**Response:** Status 204 No Content

**5 การขอข้อมูลอุปกรณ์ตามไอดีของอุปกรณ์**

GET /device/:device\_id

**Response:** Status 200 OK

ตารางที่ ข.7 ตัวอย่างการคืนค่าข้อมูลของการขอข้อมูลอุปกรณ์ตามไอดีของอุปกรณ์

|  |
| --- |
| {  "device":{  "\_id":{  "$oid":"5af97cbf02b9c80969284ca6"  },  "management\_ip":"192.168.1.1",  "device\_ip":"192.168.1.1",  "snmp\_info":{  "version":"2c",  "community":"public",  "port":161  },  "snmp\_is\_running":false,  "snmp\_last\_run\_time":1526992492.2371502,  "ssh\_info":{  "username":"cisco",  "password":"cisco",  "port":22,  "secret":"cisco"  },  "status":1,  "type":"cisco\_ios",  "is\_ssh\_connect":true, |

**ตารางที่ ข.7 (ต่อ)** ตัวอย่างการคืนค่าข้อมูลของการขอข้อมูลอุปกรณ์ตามไอดีของอุปกรณ์

|  |
| --- |
| "cdp\_enable":true,  "description":"Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.5(3)M6, RELEASE SOFTWARE (fc2)\r\nTechnical Support: http://www.cisco.com/techsupport\r\nCopyright (c) 1986-2017 by Cisco Systems, Inc.\r\nCompiled Tue 25-Jul-17 13:17 by prod\_rel\_team",  "interfaces":[  {  "index":1,  "description":"Embedded-Service-Engine0/0",  "type":6,  "mtu":1500,  "speed":10000000,  "physical\_address":"0x000000000000",  "admin\_status":2,  "operational\_status":2,  "last\_change":4275,  "in\_octets":0,  "in\_ucast\_packets":0,  "in\_discards":0,  "in\_errors":0,  "in\_unknown\_protos":0,  "out\_octets":0,  "out\_ucast\_packets":0,  "out\_discards":0,  "out\_errors":0,  "device\_ip":"192.168.1.1",  "bw\_in\_usage\_octets":0,  "bw\_in\_usage\_persec":0.0,  "bw\_in\_usage\_percent":0.0,  "bw\_out\_usage\_octets":0, |

**ตารางที่ ข.7 (ต่อ)** ตัวอย่างการคืนค่าข้อมูลของการขอข้อมูลอุปกรณ์ตามไอดีของอุปกรณ์

|  |
| --- |
| "bw\_out\_usage\_persec":0.0,  "bw\_out\_usage\_percent":0.0,  "bw\_usage\_update":1526992492.0793455  },  ],  "interfaces\_update\_time":1526992491.794834,  "name":"R1.lab306",  "updated\_at":1526992492.0795121,  "uptime":4448648,  "is\_snmp\_connect":true  },  "success":true  } |

**6 การขอข้อมูลอุปกรณ์ที่เชื่อมต่อกัน**

GET /device/:device\_id/neighbor

**Response:** Status 200 OK

ตารางที่ ข.8 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลอุปกรณ์ที่เชื่อมต่อกัน

|  |
| --- |
| {  "neighbor":[  {  "ip\_addr":"0.0.0.0",  "version":"Cisco Internetwork Operating System Software \nIOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(22)EA8a, RELEASE SOFTWARE (fc1)\nCopyright (c) 1986-2006 by cisco Systems, Inc.\nCompiled Fri 28-Jul-06 15:16 by weiliu",  "name":"Switch",  "port":"FastEthernet0/24",  "local\_ifindex":2,  "device\_ip":"192.168.1.1"  }, |

**ตารางที่ ข.8 (ต่อ)** ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลอุปกรณ์ที่เชื่อมต่อกัน

|  |
| --- |
| {  "ip\_addr":"192.168.1.6",  "version":"Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.5(3)M6, RELEASE SOFTWARE (fc2)\nTechnical Support: http://www.cisco.com/techsupport\nCopyright (c) 1986-2017 by Cisco Systems, Inc.\nCompiled Tue 25-Jul-17 13:17 by prod\_rel\_team",  "name":"R2.lab306",  "port":"Serial0/0/0",  "local\_ifindex":5,  "device\_ip":"192.168.1.1"  },  {  "ip\_addr":"192.168.1.2",  "version":"Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.5(3)M6, RELEASE SOFTWARE (fc2)\nTechnical Support: http://www.cisco.com/techsupport\nCopyright (c) 1986-2017 by Cisco Systems, Inc.\nCompiled Tue 25-Jul-17 13:17 by prod\_rel\_team",  "name":"R4.lab306",  "port":"Serial0/0/1",  "local\_ifindex":6,  "device\_ip":"192.168.1.1"  }  ],  "success":true  } |

**7 การขอข้อมูลลิงค์ทั้งหมด**

GET /link

**Response:** Status 200 OK

ตารางที่ ข.9 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลลิงค์ทั้งหมด

|  |
| --- |
| {  "links":[  {  "\_id":{  "$oid":"5af97e6c02b9c80969285511"  },  "dst\_if\_ip":"192.168.1.2",  "dst\_node\_ip":"192.168.1.2",  "src\_if\_ip":"192.168.1.1",  "src\_node\_ip":"192.168.1.1",  "dst\_if\_index":6,  "dst\_in\_use":18751.97607585901,  "dst\_ip":"192.168.1.2",  "dst\_node\_hostname":"R4.lab306",  "dst\_node\_id":{  "$oid":"5af97e4302b9c809692853c0"  },  ],  "status":"ok"  } |

**8 การขอข้อมูลลิงค์ตามลิงค์ไอดี**

GET /link/:link\_id

**Response:** Status 200 OK

ตารางที่ ข.10 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลลิงค์ตามไอดี

|  |
| --- |
| {  "link":{  "\_id":{  "$oid":"5af97e6c02b9c80969285511"  },  "dst\_if\_ip":"192.168.1.2",  "dst\_node\_ip":"192.168.1.2",  "src\_if\_ip":"192.168.1.1",  "src\_node\_ip":"192.168.1.1",  "dst\_if\_index":6,  "dst\_in\_use":18751.97607585901,  "dst\_ip":"192.168.1.2",  "dst\_node\_hostname":"R4.lab306",  "dst\_node\_id":{  "$oid":"5af97e4302b9c809692853c0"  },  "dst\_out\_use":87632.65940608659,  "dst\_port":"Serial0/0/1",  "link\_min\_speed":8000000,  "src\_if\_index":6,  "src\_in\_use":104929.19999419784,  "src\_ip":"192.168.1.1",  "src\_node\_hostname":"R1.lab306",  "src\_node\_id":{  "$oid":"5af97cbf02b9c80969284ca6"  },  "src\_out\_use":26050.905835436173, |

**ตารางที่ ข.10 (ต่อ)** ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลลิงค์ตามไอดี

|  |
| --- |
| "src\_port":"Serial0/0/1"  },  "status":"ok"  } |

**9 การข้อมูลเส้นทางจากอุปกรณ์เครื่องหนึ่งไปยังอุปกรณ์อีกเครื่องหนึ่ง**

GET /path/:from\_device\_ip,:to\_device\_ip

**Response:** Status 200 OK

ตารางที่ ข.11 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่ง

|  |
| --- |
| {  "paths":[  [  "192.168.1.1",  "192.168.1.6"  ],  ],  "status":"ok"  } |

**10 การขอข้อมูลโฟลว์ทั้งหมด**

GET /flow

**Response:** Status 200 OK

ตารางที่ ข.12 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลโฟลว์ทั้งหมด

|  |
| --- |
| {  "flows":[  {  "\_id":{  "$oid":"5b037c08eb6bb6611acaf91f"  },  "cisco\_51":0, |

**ตารางที่ ข.12 (ต่อ)** ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลโฟลว์ทั้งหมด

|  |
| --- |
| "direction":0,  "dst\_as":0,  "dst\_mask":24,  "flow\_sampler\_id":0,  "from\_ip":"172.16.0.1",  "input\_snmp":6,  "ipv4\_dst\_addr":"172.16.0.200",  "ipv4\_next\_hop":"172.16.0.200",  "ipv4\_src\_addr":"192.168.1.2",  "l4\_dst\_port":23456,  "l4\_src\_port":64735,  "output\_snmp":2,  "protocol":17,  "src\_as":0,  "src\_mask":30,  "src\_tos":0,  "tcp\_flags":16,  "created\_at":{  "$date":1526992436856  },  "first\_switched":{  "$date":1526992371388  },  "in\_bytes":11328,  "in\_pkts":8,  "last\_switched":{  "$date":1526992429388  }  },  ], |

**ตารางที่ ข.12 (ต่อ)** ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลโฟลว์ทั้งหมด

|  |
| --- |
| "status":"ok"  } |

**11 การขอข้อมูลเส้นทางของโฟลว์**

GET /flow/routing

**Response:** Status 200 OK

ตารางที่ ข.13 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นทางของโฟลว์

|  |
| --- |
| {  "flows":[  {  "\_id":{  "$oid":"5b0635c669324570da9f63ab"  },  "new\_flow":{  "name":"test-flow-routing",  "src\_ip":"10.0.0.0",  "src\_port":"any",  "src\_wildcard":"0.0.0.255",  "dst\_ip":"172.16.99.1",  "dst\_port":"5201",  "dst\_wildcard":"0.0.0.0",  "actions":[  {  "device\_id":{  "$oid":"5af97cbf02b9c80969284ca6"  },  "action":2,  "data":"192.168.1.6"  },  { |

**ตารางที่ ข.13 (ต่อ)** ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นทางของโฟลว์

|  |
| --- |
| {  "device\_id":{  "$oid":"5af97e4302b9c809692853c0"  },  "action":3,  "data":"GigabitEthernet0/1"  },  {  "device\_id":{  "$oid":"5af97e7202b9c8096928555e"  },  "action":1,  "data":""  }  ]  },  "info":{  "submit\_from":{  "type":1,  "user":"Unknown - Todo Implement"  },  "status":0  },  "created\_at":{  "$date":1527133638407  },  "updated\_at":{  "$date":1527133638407  }  } |

**ตารางที่ ข.13 (ต่อ)** ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นทางของโฟลว์

|  |
| --- |
| ],  "status":"ok"  } |

**12 การสร้างเส้นทางของโฟลว์**

POST /flow/routing

ข้อมูลที่ต้องส่งไป

ตารางที่ ข.14 ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อสร้างเส้นทางของโฟลว์

|  |
| --- |
| {  "name":"test-flow-routing",  "src\_ip":"10.0.0.0",  "src\_subnet":"0.0.0.255",  "src\_port":"any",  "dst\_ip":"172.16.99.1",  "dst\_subnet":"0.0.0.0",  "dst\_port":"5201",  "actions":[  {  "device\_id":"5af97cbf02b9c80969284ca6",  "action":2,  "data":"192.168.1.6"  },  {  "device\_id":"5af97e4302b9c809692853c0",  "action":3,  "data":"GigabitEthernet0/1"  }, {  "device\_id":"5af97e7202b9c8096928555e",  "action":"1",  "data":"" |

**ตารางที่ ข.14 (ต่อ)** ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อสร้างเส้นทางของโฟลว์

|  |
| --- |
| }  ]  } |

**Reponse:** Status 201 Created

ตารางที่ ข.15 ตัวอย่างข้อมูลที่คืนค่าของการสร้างเส้นทางของโฟลว์

|  |
| --- |
| {"status": "ok"} |

**13 การแก้ไขข้อมูลเส้นทางของโฟลว์**

PATCH /flow/routing

ข้อมูลที่ต้องส่งไป

ตารางที่ ข.16 ตัวอย่างข้อมูลที่ใช้ในการส่งเพื่อแก้ไขเส้นทางของโฟลว์

|  |
| --- |
| {  "name":"test-flow-routing",  "src\_ip":"10.0.0.0",  "src\_port":"any",  "src\_subnet":"0.0.0.255",  "dst\_ip":"172.16.99.1",  "dst\_port":"5201",  "dst\_subnet":"0.0.0.0",  "actions":[  {  "device\_id":"5af97e7202b9c8096928555e",  "action":1,  "data":""  }  ],  "flow\_id":1  } |

**Response**: Status 200 OK

ตารางที่ ข.17 ตัวอย่างข้อมูลที่คืนค่าของการแก้ไขข้อมูลเส้นทางของโฟลว์

|  |
| --- |
| {  "status":true,  "message":"Update flow routing!"  } |

**14 การลบเส้นทางของโฟลว์**

DELETE /flow/routing?flow\_id=:flow\_id

**Response:** Status 204 No Content

**15 การขอเส้นทางของโฟลว์**

GET /flow/routing/:flow\_id

ตารางที่ ข.18 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นทางของโฟลว์

|  |
| --- |
| {  "flow":{  "\_id":{  "$oid":"5b0635c669324570da9f63ab"  },  "new\_flow":{  "name":"test-flow-routing",  "src\_ip":"10.0.0.0",  "src\_port":"any",  "src\_wildcard":"0.0.0.255",  "dst\_ip":"172.16.99.1",  "dst\_port":"5201",  "dst\_wildcard":"0.0.0.0",  "actions":[  {  "device\_id":{  "$oid":"5af97cbf02b9c80969284ca6" |

**ตารางที่ ข.18 (ต่อ)** ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลเส้นทางของโฟลว์

|  |
| --- |
| },  "action":2,  "data":"192.168.1.6"  ]  },  "info":{  "submit\_from":{  "type":1,  "user":"Unknown - Todo Implement"  },  "status":0  },  "created\_at":{  "$date":1527133638407  },  "updated\_at":{  "$date":1527133638407  }  },  "status":"ok"  } |

**16 การขอข้อมูลตารางเส้นทางของอุปกรณ์**

GET /routes/device\_id

**Response:** Status 200 OK

ตารางที่ ข.19 ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลตารางเส้นทางของอุปกรณ์

|  |
| --- |
| {  "routes":[  {  "\_id":{ |

**ตารางที่ ข.19 (ต่อ)** ตัวอย่างข้อมูลที่คืนค่าของการขอข้อมูลตารางเส้นทางของอุปกรณ์

|  |
| --- |
| "$oid":"5b040e6c69324522cf090cfe"  },  "dst":"172.16.0.0",  "mask":"255.255.255.0",  "tos":0,  "next\_hop":"0.0.0.0",  "if\_index":2,  "type":3,  "proto":2,  "age":44447,  "info":"0.0",  "next\_hop\_AS":0,  "metric1":0,  "metric2":-1,  "metric3":-1,  "metric4":-1,  "metric5":-1,  "status":1,  "start\_ip":2886729729,  "end\_ip":2886729982,  "device\_id":{  "$oid":"5af97cbf02b9c80969284ca6"  },  "updated\_at":1526992492.2342358  },  ],  "status":"ok"  } |

ประวัติผู้เขียน

ชื่อ – นามสกุล นายกฤษฎา อิงอาน

วัน เดือน ปีเกิด 12 เมษายน 2539

สถานที่เกิด จังหวัดกรุงเทพมหานคร

ที่อยู่ เลขที่ 471 หมู่บ้านฟ้าคราม หมู่5 ต.คูคต อ.ลำลูกกา จ.ปทุมธานี

ประวัติการศึกษา 2560 วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ ทหารลาดกระบัง

อีเมล krissada.ing@gmail.com

ชื่อ – นามสกุล นายสถาพร แดงน้อย

วัน เดือน ปีเกิด 12 กันยายน 2538

สถานที่เกิด จังหวัดสมุทรปราการ

ที่อยู่ เลขที่ 1 หมู่ 7 ตำบล คลองสวน อำเภอ บางบ่อ จังหวัด สมุทรปราการ 10560

ประวัติการศึกษา 2560 วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ ทหารลาดกระบัง

อีเมล Sathapon\_Dangnoi@outlook.com