โครงงานวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเกษตรศาสตร์

เรื่อง แอคเซสพอยต์ที่รองรับซอฟต์แวร์ดีฟายนด์เน็ตเวิร์ค

SDN-Based Access Point

โดย นายสิรภพ สัตตบงกช

5610504007

พ.ศ. 2559

แอคเซสพ้อยต์ที่รองรับซอฟต์แวร์ดีฟายนด์เน็ตเวิร์ค

SDN-Based Access Point

โดย

นายสิรภพ สัตตบงกช เลขประจำตัว 5610504007

โครงงานวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเกษตรศาสตร์

ตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

ได้รับการพิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษา ..... วันที่ ..... เดือน ..... พ.ศ. .....

(รศ.ดร.อนันต์ ผลเพิ่ม)

..... วันที่ ..... เดือน ..... พ.ศ. .....

(ผศ.ดร.ชัยพร ใจแก้ว)

..... วันที่ ..... เดือน ..... พ.ศ. .....

(ผศ.ดร.อภิรักษ์ จันทร์สร้าง)

หัวหน้าภาควิชา ..... วันที่ ..... เดือน ..... พ.ศ. .....

(รศ.ดร.อนันต์ ผลเพิ่ม)

นายสิรภพ สัตตบงกช ปีการศึกษา 2559

แอคเซสพอยต์ที่รองรับซอฟต์แวร์ดีฟายนด์เน็ตเวิร์ค

ปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์) ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

บทคัดย่อ

เทคโนโลยีซอฟต์แวร์ดีฟายนด์เน็ตเวิร์คได้รับการพัฒนาขึ้นเพื่อช่วยเพิ่มความสะดวกในการจัดการอุปกรณ์เครือข่ายที่มาจากผู้ผลิตที่แตกต่างกัน และทำให้การบริหารจัดการอุปกรณ์ทำได้สะดวกมากยิ่งขึ้น อย่างไรก็ตามอุปกรณ์ประเภทแอคเซสพอยต์ซึ่งเป็นส่วนสำคัญในระบบเครือข่ายแลนไร้สายนั้นยังไม่ได้รับการพัฒนาให้รองรับเทคโนโลยีนี้มากเท่าที่ควร อีกทั้งการควบคุมแอคเซสพอยต์จำนวนมากเป็นการสิ้นเปลืองเวลาและทำให้เกิดข้อผิดพลาดได้ง่าย โครงงานนี้จึงได้นำเสนอการพัฒนาแอคเซลพอยต์ให้มีความสามารถรองรับเทคโนโลยีซอฟต์แวร์ดีฟายนด์เน็ตเวิร์ค เพื่อที่จะทำให้การควบคุมและดูแลเครือข่ายเป็นไปอย่างครอบคลุมทุกอุปกรณ์ในเครือข่ายตั้งแต่อุปกรณ์สวิตช์ไปจนถึงอุปกรณ์แอคเซสพอยต์ ระบบที่พัฒนาขึ้นจะทำให้ผู้ดูแลเครือข่ายสามารถควบคุม สั่งการ และออกแบบเครือข่ายได้จากศูนย์กลางโดยผ่านทางโปรแกรมประยุกต์ โดยไม่จำเป็นต้องจัดการตั้งค่าอุปกรณ์ทีละตัว

คำสำคัญ : ซอฟต์แวร์ดีฟายนด์เน็ตเวิร์ค, อุปกรณ์เครือข่าย, อุปกรณ์แอคเซสพอยต์, ระบบเครือข่ายแลนไร้สาย, อุปกรณ์สวิตช์, ผู้ดูแลเครือข่าย, โปรแกรมประยุกต์

Siraphob Sattabongkot Academic Year 2016

SDN-Based Access Point

Bachelor Degree in Computer Engineering. Department of Computer Engineering. Faculty of Engineering, Kasetsart University

Abstract

Software-defined network (SDN) technology has been developed to ease the management of a large volume of equipment, especially from different manufacturers. However, wireless access points, which are the main components of wireless LAN, have not been given much support for SDN. Moreover, controlling a large number of access points potentially waste a lot of time and is very error-prone. This article introduces the development of SDN-enabled access points. This will provide complete management of all network equipment, ranging from switches to wireless access points. The proposed system will allow network operator to control, command, and design networks from the center application instead of accessing each device one by one. Experiments show that, even with a high amount of requests, all devices can respond in a timely manner.

กิตติกรรมประกาศ

โครงงานนี้สามารถสำเร็จลุล่วงได้ด้วยความกรุณาและความช่วยเหลือเป็นอย่างสูงจาก รศ.ดร.อนันต์ ผลเพิ่ม ผศ.ดร.ชัยพร ใจแก้ว และ ผศ.ดร.อภิรักษ์ จันทร์สร้าง ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้คำแนะนำ และข้อเสนอแนะในการพัฒนาโครงงานและพัฒนาตนเอง ตลอดจนช่วยตรวจสอบข้อบกพร่องต่างๆ ที่เกิดขึ้นมาตลอดโครงงาน ทางผู้พัฒนาจึงขอกราบขอบพระคุณเป็นอย่างสูง และขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ที่ได้เอื้อเฟื้อสถานที่และอำนวยความสะดวกในการจัดหาอุปกรณ์สำหรับการพัฒนาโครงงาน รวมถึงช่วยสนับสนุนทุนในการพัฒนาโครงงาน

ขอขอบคุณโครงการการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 19 โดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ที่ได้มอบทุนสนับสนุนในการจัดทำโครงงานนี้

ขอขอบคุณสมาชิกห้องปฏิบัติการวิจัยเครือข่ายไร้สาย (Intelligent Wireless Network Group : IWING) ทุกท่าน รวมไปถึงเพื่อนนิสิต และอาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ที่ได้ให้ความช่วยเหลือมาตลอด ขอขอบคุณครอบครัวของผู้พัฒนาโครงงานที่คอยให้กำลังใจมาโดยตลอด และสุดท้ายขอขอบคุณผู้ที่เกี่ยวของทุกท่านที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ ที่มีส่วนช่วยเหลือให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

นายสิรภพ สัตตบงกช

ผู้จัดทำ

สารบัญ

บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญภาพ	VI
สารบัญตาราง	X
1. บทนำ	1
1.1 วัตถุประสงค์ของโครงงาน	2
1.2 ขอบเขตของโครงงาน	2
1.3 ประโยชน์ที่ได้รับ	2
2. ทฤษฎีและงานที่เกี่ยวข้อง	3
2.1 ทฤษฎีที่เกี่ยวข้อง	3
2.2 งานที่เกี่ยวข้อง	4
3. เครื่องมือที่ใช้ในการทำโครงงาน	6
3.1 ระบบปฏิบัติการ	6
3.2 ภาษาที่ใช้ในการพัฒนา	6
3.3 ซอฟต์แวร์และไลบรารี	6
3.4 ฮาร์ดแวร์	8
4. วิธีการดำเนินโครงงาน	10
4.1 ภาพรวมของระบบ	10
4.2 รายละเอียดของระบบที่พัฒนา	11
4.3 องค์ประกอบโดยรวมของระบบ	11
4.4 ขั้นตอนการพัฒนา	13

5. ผลการดำเนินโครงงานและวิจารณ์ 14
5.1 สภาพแวดล้อมในการทดสอบ 14
5.2 ผลการทดสอบและการวิจารณ์ผล 16

6. สรุปผลการดำเนินงานและข้อเสนอแนะ 17
6.1 ข้อสรุป 17
6.2 ปัญหาและอุปสรรค 17
6.3 แนวทางการพัฒนาต่อ 18
6.4 ข้อเสนอแนะ 18

7. บรรณานุกรม 19

8. ภาคผนวก 20
8.1 คู่มือการติดตั้ง 20
8.2 คู่มือการใช้งาน 21
9. ประวัตินิสิต 23

สารบัญภาพ

รูปที่ 2.1 โครงสร้างพื้นฐานของ SDN 3

รูปที่ 2.2 โครงสร้างของ OpenFlow Switch 4

รูปที่ 3.1 Toshiba Satellite M840 8

รูปที่ 3.2 D-link dir-825c 9

รูปที่ 4.1 ภาพรวมของระบบ 10

รูปที่ 4.2 องค์ประกอบโดยรวมของระบบ 12

รูปที่ 5.1 เวลาการตอบสนองต่อคำสั่งจากคอนโทรลเลอร์ (Rt) 14

รูปที่ 5.2 แผนภูมิแสดงเวลาที่ใช้ตอบสนองต่อคำสั่งชนิด get 15

รูปที่ 5.3 แผนภูมิแสดงเวลาที่ใช้ตอบสนองต่อคำสั่งชนิด set 15

รูปที่ 5.4 แผนภูมิแสดงเวลาที่ใช้ตอบสนองต่อคำสั่งชนิดต่างๆ 16

สารบัญตาราง

ตารางที่ 3.1 คุณสมบัติของ Toshiba Satellite M840 8

ตารางที่ 3.2 คุณสมบัติของ D-link dir-825c 9

1. บทนำ

การให้บริการเครือข่ายคอมพิวเตอร์ภายในองค์กรขนาดกลางและขนาดใหญ่นั้นประกอบด้วยการทำงานร่วมของอุปกรณ์เครือข่ายจำนวนมากและมาจากหลากหลายผู้ผลิต ซึ่งทำให้อุปกรณ์เหล่านี้ต้องการความยืดหยุ่นและความรวดเร็วในการทำงาน นอกเหนือจากการทำงานและตั้งค่าเครือข่ายคอมพิวเตอร์ในรูปแบบเดิมแล้ว ปัจจุบันเทคโนโลยีซอฟต์แวร์ดีฟายนด์เน็ตเวิร์ค (SDN: Software Defined Network)[1] ได้มีส่วนช่วยในด้านของการจัดการข้อมูล และบริหารเส้นทางข้อมูลของเครือข่ายขนาดใหญ่ ซึ่งสามารถรองรับการเชื่อมต่อและสั่งงานแก่อุปกรณ์ที่มีมาจากผู้ผลิตที่แตกต่างกันได้ ทำให้องค์กรสามารถขยายขนาดของเครือข่ายได้โดยไม่ต้องกังวลถึงความแตกต่างของผู้ผลิต

องค์ประกอบของซอฟต์แวร์ดีฟายนด์เน็ตเวิร์คประกอบด้วย 2 ส่วน คือ 1) อุปกรณ์คอนโทรลเลอร์ (Controller) ซึ่งจะเป็นส่วนสั่งการและประมวลผลข้อมูลของอุปกรณ์ต่าง ๆ ที่อยู่ภายในเครือข่าย 2) อุปกรณ์เครือข่ายที่รองรับคำสั่งสั่งจากคอนโทรลเลอร์เพื่อใช้ในการกำหนดพอร์ทและเส้นทางการเดินทางการส่งแพ็กเก็ตที่ได้รับข้อมูลเข้ามา เช่น อุปกรณ์สวิตซ์

ข้อดีของการนำเทคโนโลยี SDN มาใช้งาน คือ ช่วยให้ผู้ดูแลเครือข่ายสามารถควบคุมดูแลเครือข่ายได้จากศูนย์กลางผ่านทางคอนโทรลเลอร์ โดยผู้ดูแลเครือข่ายไม่จำเป็นต้องปรับแต่งค่าของอุปกรณ์ทีละตัว ซึ่งผู้ดูแลเพียงแค่ออกแบบเครือข่ายตามความต้องการ แล้วคอนโทรลเลอร์จะทำหน้าที่ส่งคำสั่งไปยังอุปกรณ์ที่เกี่ยวข้องโดยอัตโนมัติ

อย่างไรก็ตามอุปกรณ์ประเภทแอคเซสพอยต์ ซึ่งเป็นส่วนสำคัญในระบบเครือข่ายแลนไร้สายนั้นยังไม่ได้รับการพัฒนาให้รองรับเทคโนโลยีนี้มากเท่าที่ควร อีกทั้งการควบคุมแอคเซสพอยต์จำนวนมากเป็นการสิ้นเปลืองเวลาและเกิดข้อผิดพลาดได้ง่าย และด้วยปริมาณการใช้งานเครือข่ายไร้สายที่เพิ่มขึ้นอย่างมาก ทำให้เครือข่ายไร้สายได้ถือเป็นส่วนสำคัญในองค์กรหรือถือเป็นส่วนสำคัญในการใช้ชีวิตประจำวันได้เลยทีเดียว

ถึงแม้ว่าเทคโนโลยีที่ใช้จัดการควบคุมแอคเซสพอยต์จำนวนมากนั้นจะถูกพัฒนาขึ้นมาบ้างแล้ว เช่น อุปกรณ์ Wireless LAN Controller[2] แต่ทว่าอุปกรณ์นี้มีต้นทุนที่สูง เนื่องจากเป็นอุปกรณ์ฮาร์ดแวร์ขนาดใหญ่ และอุปกรณ์นี้ยังต้องทำงานภายใต้เงื่อนไขที่ว่าอุปกรณ์ทั้งหมดจะต้องมาจากผู้ผลิตเดียวกันอีกด้วย

ดังนั้น ในบทความนี้ผู้วิจัยจึงนำเสนอการพัฒนาแอคเซสพอยต์ที่รองรับ SDN เพื่อให้ผู้ดูแลระบบสามารถควบคุมสั่งการและตั้งค่าอุปกรณ์แอคเซสพอยต์ผ่านคอนโทรลเลอร์ได้ ซึ่งทำให้มีความสะดวกมากขึ้นและยังช่วยลดความผิดพลาดที่อาจเกิดขึ้นจากการตั้งค่าของผู้ดูแลเครือข่ายเองอีกด้วย

1.1 วัตถุประสงค์ของโครงงาน

- เพื่อให้ผู้ดูแลเครือข่ายสามารถสั่งการแอคเซสพอยต์ทั้งหมดผ่านทางซอฟต์แวร์ได้จากศูนย์กลาง
- เพื่อลดค่าใช้จ่ายสำหรับฮาร์ดแวร์สำหรับการจัดการแอคเซสพอยต์ปริมาณมาก
- เพื่อให้องค์กรสามารถขยายขนาดของเครือข่ายได้โดยไม่ต้องคำนึงถึงความแตกต่างของผู้ผลิตของอุปกรณ์ที่นำมาใช้งานในองค์กร
- เพื่อลดเวลาในการแก้ไขปัญหาที่เกิดขึ้นจากความผิดพลาดของทั้งการตั้งค่าอุปกรณ์และความผิดพลาดที่เกิดจากอุปกรณ์

1.2 ขอบเขตของโครงงาน

การพัฒนาระบบแอคเซสพอยต์ที่รองรับ SDN มีขอบเขตและข้อจำกัดดังนี้

ขอบเขตของโครงงาน

- ผู้ดูแลเครือข่ายสามารถสั่งการแอคเซสพอยต์จำนวนมากได้จากศูนย์กลาง
- ระบบสามารถรองรับแอคเซสพอยต์ที่มาผู้ผลิตที่แตกต่างกันได้
- ผู้ดูแลเครือข่ายสามารถปรับปรุงเครือข่ายไร้สายของตนเองได้ โดยจะสามารถปรับแต่งช่องสัญญาณ โหมดการทำงาน ชื่อของเครือข่าย และการจัดการแบนด์วิดท์ ได้ตามความต้องการขององค์กร

ข้อจำกัดของโครงงาน

- แอคเซสพอยต์จะต้องสามารถรองรับเฟิร์มแวร์ของ OpenWrt[3] ได้
- เฟิร์มแวร์ของแอคเซสพอยต์จะต้องสามารถรองรับภาษาไพธอนได้
- เครื่องคอมพิวเตอร์หรือเซิร์ฟเวอร์ที่ทำงานเป็นคอนโทรลเลอร์จะต้องสามารถรองรับภาษาไพธอนได้

1.3 ประโยชน์ที่ได้รับ

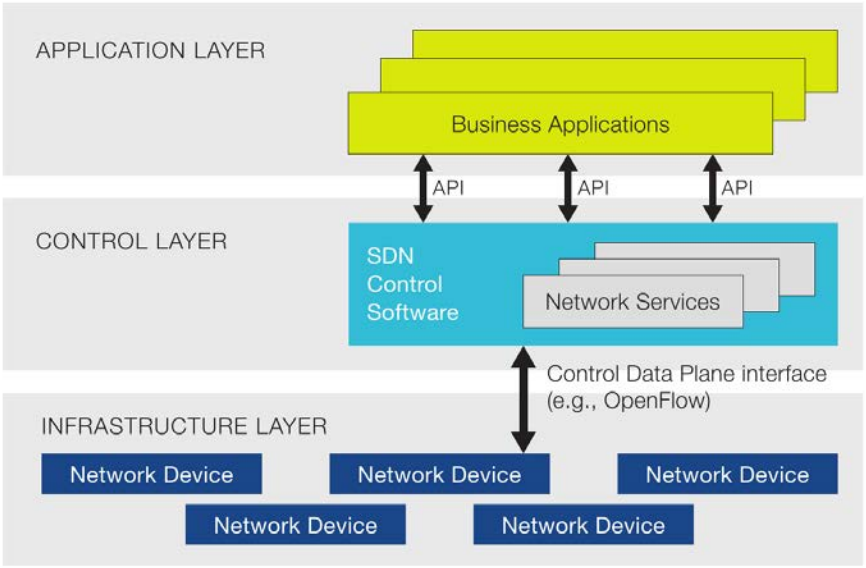
- ผู้ดูแลเครือข่ายสามารถทำงานได้สะดวกและครอบคลุมอุปกรณ์ในเครือข่ายมากขึ้น
- องค์กรสามารถขยายขนาดเครือข่ายได้โดยไม่ต้องพะวงถึงความแตกต่างของผู้ผลิต
- ช่วยลดค่าใช้จ่ายในการจัดซื้ออุปกรณ์สำหรับความคุมแอคเซสพอยต์จำนวนมาก

2. ทฤษฎีและงานที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

- **Software Defined Network (SDN)**

เป็นการออกแบบสถาปัตยกรรมของระบบเครือข่ายโดยให้ซอฟต์แวร์เป็นส่วนที่ควบคุมอุปกรณ์ในเครือข่าย เพื่อแก้ปัญหาที่เกิดขึ้น กล่าวคือ ปัญหาที่เกิดจากผู้ผลิตไม่ตรงกันในแต่ละอุปกรณ์ ปัญหาที่เกิดจากการต้องเข้าไปแก้ปัญหาที่อุปกรณ์ทีละอุปกรณ์ ซึ่งระบบนี้จะทำให้เครือข่ายเกิดความยืดหยุ่นและเอื้อต่อการขยายเครือข่ายมากขึ้น โดย SDN อยู่บนฐานของแนวคิดที่ว่า ส่วนที่ควบคุม (Control Plane) และส่วนของข้อมูล (Data Plane) จะถูกแยกออกจากกันโดยชัดเจน โดยอาศัยโปรโตคอลโอเพ่นโฟลวที่เชื่อมต่อระหว่างทั้งสองส่วน ซึ่งส่วนที่ควบคุมคือส่วนที่จะทำการตัดสินใจและอนุญาต รวมทั้งกำหนดทิศทางการส่งข้อมูลของส่วนของข้อมูล

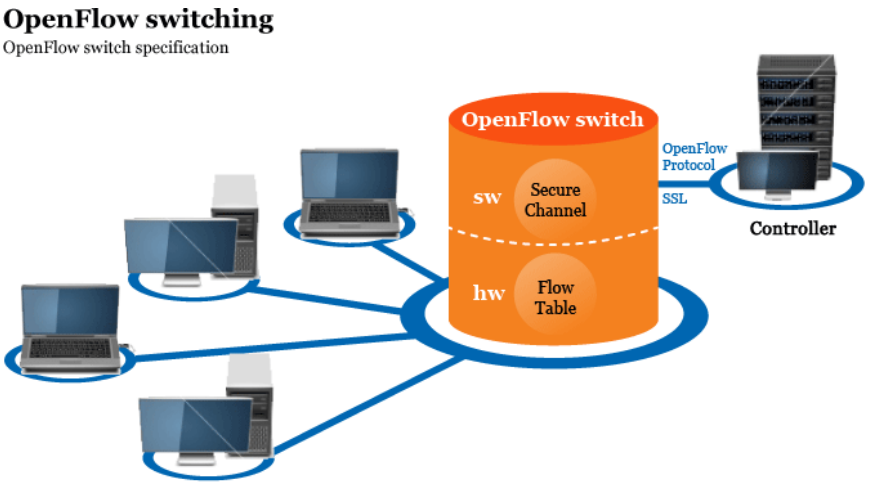


รูปที่ 2.1 โครงสร้างพื้นฐานของ SDN

[ที่มา: https://www.sdxcentral.com/sdn/definitions/inside-sdn-architecture/]

- **OpenFlow**

คือมาตรฐานเปิดที่อนุญาตให้ผู้ใช้งานสามารถนำโปรโตคอลที่สร้างขึ้นเองมาใช้ในระบบเครือข่ายได้[4] โดยการทำงานจะเปรียบเสมือนกับการเพิ่มฟีเจอร์เข้าไปยังอุปกรณ์เครือข่าย โดย OpenFlow จะเป็นตัวกลางในการติดต่อระหว่างส่วนควบคุมและส่วนของข้อมูลโดยใช้โปรโตคอลที่กำหนดไว้ เช่น การรับส่งแพ็คเกจ การเปลี่ยนแปลงตารางฟอร์เวิร์ดดิ้ง หรือการอ่านค่าสถานะของอุปกรณ์



รูปที่ 2.2 โครงสร้างของ OpenFlow Switch

[ที่มา: http://www.xinguard.com/en/content.aspx?id=70]

- **มาตรฐาน IEEE 802.11 (Wi-Fi)**

  เป็นมาตรฐานที่ถูกกำหนดขึ้นมาเพื่อใช้ในเครือข่ายแลนไร้สาย ซึ่งมีการรับส่งข้อมูลอยู่บนคลื่นวิทยุที่ย่านความถี่ 2.4 และ 5 GHz โดยสามารถแบ่งหมวดหมู่ได้ตามคุณสมบัติของคลื่นเป็น AC, N, G และ B

- **Socket**

  คือการสื่อสารระหว่างโปรแกรมบนเครือข่าย โดยการติดต่อกันผ่าน Socket API ซึ่งจะประกอบไปด้วย IP address และ port number ทำให้โปรแกรมที่อยู่คนละตำแหน่งในเครือข่ายสามารถทำการส่งข้อมูลให้กันได้

2.2 งานที่เกี่ยวข้อง

การประยุกต์ใช้ SDN

การนำ SDN มาประยุกต์ใช้ในงานเครือข่ายต่างๆเริ่มเป็นแนวทางที่ได้รับความนิยมมากขึ้นในปัจจุบัน โดยตัวอย่างการนำ SDN มาปรับใช้ได้แก่ An SDN-Based Architecture for Next-Generation Wireless Networks[5] ที่เห็นความสำคัญของการเจริญเติบโตอย่างรวดเร็วของเครือข่ายไร้สาย และนำเสนอการพัฒนาสถาปัตยกรรมของเครือข่ายไร้สายรุ่นใหม่ ให้สามารถรองรับกับ SDN ได้

การใช้งาน OpenFlow

การนำ OpenFlow มาใช้ใน Ethernet Switch นั้นได้มีการแพร่หลายออกไปอย่างกว้างขวาง ซึ่งการวิจัย OpenFlow: Enabling innovation in campus networks[6] ได้ทำการศึกษาการนำ OpenFlow เข้ามาใช้กับเครือข่ายของมหาวิทยาลัย โดยชี้จุดเด่นด้านสมบัติในการขยายขนาดของเครือข่าย เพื่อที่จะส่งเสริมให้โรงเรียนและมหาวิทยาลัยให้ความสนใจกับ OpenFlow กันมากขึ้น

3. เครื่องมือที่ใช้ในการทำโครงงาน

3.1 ระบบปฏิบัติการ

- **Ubuntu 16.04**

  Ubuntu 16.04 เป็นระบบปฏิบัติการคอมพิวเตอร์ชนิดหนึ่งซึ่งเป็นซอฟต์แวร์เสรีที่สามารถนำไปใช้งานได้โดยไม่เสียค่าใช้จ่ายใดๆ ซึ่งโครงงานนี้ได้นำมาใช้ในการพัฒนาโปรแกรมในฝั่งคอนโทรลเลอร์และฝั่งแอคเซสพอยต์ และยังใช้ในการทำงานของคอนโทรลเลอร์ด้วย

- **OpenWrt Chaos Calmer 15.05**

  Chaos Calmer เป็น linux kernel ที่สามารถทำงานในแอคเซสพอยต์ โดยทำหน้าที่รองรับการทำงานของโปรแกรมในฝั่งแอคเซสพอยต์ โดยผู้ใช้งานสามารถปรับแต่งความสามารถของ kernel ได้ตามต้องการ และยังเป็นซอฟต์แวร์เสรีที่นำไปใช้งานได้โดยไม่ต้องเสียค่าใช้จ่ายอีกด้วย

3.2 ภาษาที่ใช้ในการพัฒนา

- **ภาษาไพธอน (Python)**

  ใช้ในการพัฒนาโปรแกรมในส่วนของการทำงานทั้งส่วนของคอนโทรลเลอร์และแอคเซสพอยต์

3.3 ซอฟต์แวร์และไลบรารี

ซอฟต์แวร์

- **Gedit**

  เป็น text editor ที่ใช้แก้ไขไฟล์ text โดยสามารถรองรับภาษาที่ใช้ในการ programming ได้หลากหลายภาษา สามารถไฮไลต์ส่วนสำคัญของ source code ได้ และยังเป็นซอฟต์แวร์ที่ใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย เพราะติดมากับระบบปฏิบัติการ Ubuntu เลย

ไลบรารี

- **Socket**

  เป็นไลบรารีของไพธอนที่สามารถทำการเชื่อมต่อระหว่างเครื่อง client กับเครื่อง server ได้ โดยใช้การเชื่อมต่อผ่าน python socket ที่จะทำการเชื่อมโยง ip ของเครื่อง client เข้ากับ port ของ server แล้วจึงสามารถส่งข้อความให้กันได้

- **Sys**

  หรือ System specific parameters and functions เป็นไลบรารีของไพธอนที่ใช้สำหรับเข้าถึงค่าตัวแปรที่ถูกใช้หรือดูแลโดย interpreter และฟังก์ชั่นที่ส่งผลกับ interpreter[7]

- **Re**

  หรือ Regular expression operation เป็นไลบรารีของไพธอนที่ใช้สำหรับตรวจสอบหารูปแบบหรือกลุ่มคำ โดยตรวจสอบให้ตรงกับเงื่อนไขที่เรากำหนดไว้ โดยสามารถใช้ค้นหา เปลี่ยนแปลงค่า ตรวจสอบรูปแบบต่างๆได้[8]

- **Threading**

  เป็นไลบรารีของไพธอนที่ใช้ในการเพิ่มจำนวน thread ของ processor เพื่อแบ่งเบาการทำงานของ processor แต่ละตัว ซึ่งในโครงงานนี้ใช้ thread เพื่อแยกส่วนระหว่างคำสั่งของผู้ดูแลระบบและส่วนของการเชื่อมต่อกับแอคเซสพอยต์

- **Time**

  เป็นไลบรารีของไพธอนที่เกี่ยวกับฟังก์ชั่นที่เกี่ยวข้องกับเวลาทั้งหลาย ในโครงงานนี้ใช้เพื่อกำหนดช่วงเวลาในการทำงาน โดยนับเป็นวินาที เช่นการหยุดการทำงานของ process นั้นๆเพื่อรอการทำงานของ process อื่นๆ [9]

- **Subprocess**

  เป็นไลบรารีของไพธอนที่ทำให้สามารถสร้าง process ขึ้นมาใหม่ หรือเชื่อมต่อ input/output/error pipe และรับค่าที่ return กลับมา จากการส่งคำสั่งเหล่านั้น[10]

3.4 ฮาร์ดแวร์

- **เครื่องคอมพิวเตอร์พกพา**

ในโครงงานนี้จะใช้คอมพิวเตอร์ Toshiba Satellite M840 ดังในรูป โดยทำหน้าที่เป็นคอนโทรลเลอร์ ดังรูปที่ 3.1 และมีคุณสมบัติดังตาราง 3.1

รูปที่ 3.1 Toshiba Satellite M840

ที่มา: http://www.digit.in/laptops/toshiba-satellite-m840-i4010-review-1305.html]

ตารางที่ 3.1 คุณสมบัติของ Toshiba Satellite M840

Model	Toshiba Satellite M840
CPU	Intel Core i7-3612QM (2.10-3.10 GHz)
VGA	AMD Radeon HD 7670M (1GB GDDR3)
RAM	4GB DDR3x2
Harddisk	750 GB 5400 RPM
OS	Windows 7 Home Premium (64 bit)
Wireless Lan card	Intel Jackson Peak N2230

- **อุปกรณ์ที่ใช้ในการทดสอบ**

อุปกรณ์แอคเซสพอยต์ D-Link โมเดล 825 c ทำหน้าที่เป็นแอคเซสพอยต์ที่อยู่ในเครือข่ายรอรับคำสั่งจากคอนโทรลเลอร์และตอบสนองต่อคำสั่งต่างๆ ดังรูปที่ 3.2 และตารางที่ 3.2

รูปที่ 3.2 D-link dir-825 c

[ที่มา: http://www.dlink.com/uk/en/support/product/

dir-825-xtreme-n-dual-band-gigabit-router]

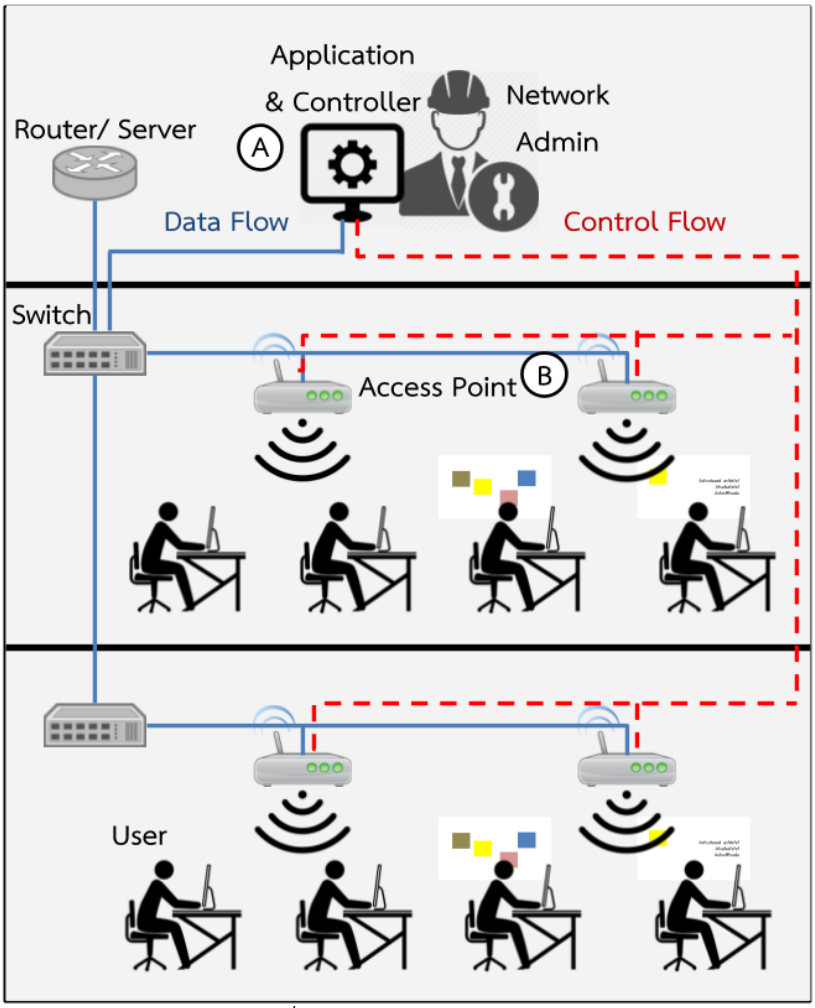
ตารางที่ 3.2 คุณสมบัติของ D-link dir-825c

Model	D-link dir-825c
System-On-Chip	AR7161 rev 2 (MIPS 24Kc V7.4)
CPU	24Kc V7.4 680 MHz
Flash-Chip	Spansion S25FL064A
Flash size	8192 KiB
RAM	64 MiB
Wireless	2 x Atheros AR922X 2.4GHz/5.0GHz 802.11abgn
Ethernet	RealTek RTL8366S Gigabit w/ port based vlan support

4. วิธีการดำเนินโครงงาน

4.1 ภาพรวมของระบบ

ภาพรวมของระบบสามารถแสดงได้ดังรูปที่ 4.1 โดยเริ่มต้นผู้ดูแลเครือข่ายจะทำการเปิดใช้งานคอนโทรลเลอร์ที่จุดศูนย์กลาง (A) จากนั้นเมื่อแอคเซสพอยต์ (B) เชื่อมต่อกับระบบเครือข่ายได้แล้ว แอคเซสพอยต์จะทำการเชื่อมต่อไปยังคอนโทรลเลอร์เพื่อให้คอนโทรลเลอร์ทราบสถานะการเชื่อมต่อของตนเอง จากนั้นเมื่อผู้ดูแลทำการตั้งค่าเครือข่ายหรือเปลี่ยนแปลงการตั้งค่าของเครือข่าย คอนโทรลเลอร์จะส่งคำสั่งไปยังแอคเซสพอยต์ที่เชื่อมต่ออยู่ในเครือข่าย และแอคเซสพอยต์จะทำการแก้ไขการตั้งค่าพื้นฐานของตัวเอง ให้สอดคล้องกับการทำงานของเครือข่ายตามที่ผู้ดูแลเครือข่ายออกแบบไว้ จากในรูป จะเห็นได้ว่าเส้นทางการเชื่อมต่อระหว่างแอคเซสพอยต์และคอนโทรลเลอร์ (Control Flow) และเส้นทางการส่งข้อมูลทั่วไป (Data Flow) นั้นไม่ได้เป็นเส้นทางเดียวกันในเชิงทฤษฎี แต่ในเชิงกายภาพนั้นยังคงใช้งานสายแลนและอุปกรณ์เครือข่ายเดียวกันอยู่



รูปที่ 4.1 ภาพรวมของระบบ

4.2 รายละเอียดของระบบที่พัฒนา

Input Specification

- ผู้ดูแลระบบทำการตั้งค่าเครือข่ายในแบบที่ต้องการผ่านทางแอพพลิเคชั่น

Output Specification

- คอนโทรลเลอร์ทำการคำนวณและส่งผลการคำนวณการตั้งค่าไปให้กับแอคเซสพอยต์
- แอคเซสพอยต์ที่เกี่ยวข้องกับเครือข่ายดังกล่าวจะทำการตั้งค่าเครือข่ายตามที่ผู้ดูแลต้องการ

Functional Specification

- สามารถแก้ไขคุณสมบัติของเครือข่ายไร้สายผ่านคอมมานไลน์ได้
- สามารถควบคุมการใช้งานเครือข่ายไร้สายในแต่ละเครือข่ายได้

กลุ่มผู้ใช้งาน

กลุ่มผู้ใช้งานจะเหมาะกับผู้ดูแลเครือข่ายที่ทำงานในเครือข่ายไร้สายที่มีขนาดกลางขึ้นไป ที่จะต้องจัดการแอคเซสพอยต์จำนวนมาก เช่น โรงเรียน โรงแรม องค์กรขนาดกลาง และองค์กรขนาดใหญ่ ไม่เหมาะกับการใช้ในครัวเรือนที่มีจำนวนแอคเซสพอยต์จำนวนน้อย สามารถดูแลให้ทั่วถึงได้ง่าย

4.3 องค์ประกอบโดยรวมของระบบ

องค์ประกอบโดยรวมของระบบสามารถแสดงได้ดังรูปที่ 4.2 ซึ่งประกอบด้วย 2 ส่วนหลัก คือ

- **คอนโทรลเลอร์**

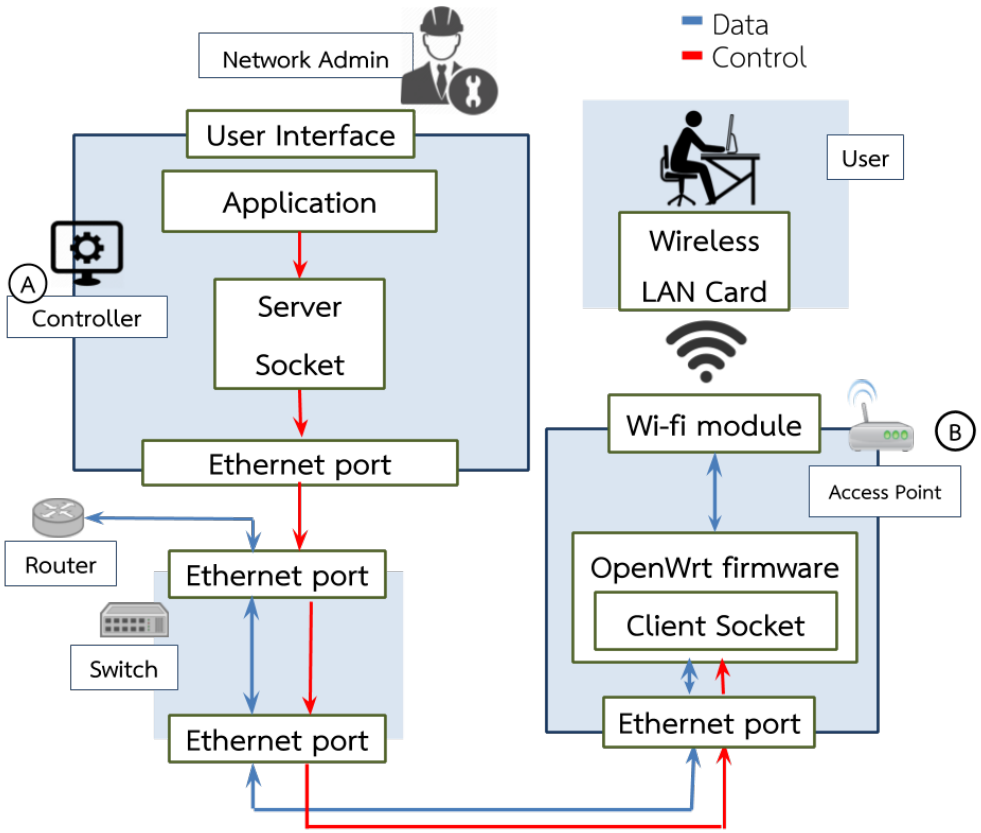
  ในส่วนของคอนโทรลเลอร์ (A) จะประกอบด้วยส่วนของแอพพลิเคชั่นที่ใช้ในการจัดการและคอนโทรลเลอร์ เมื่อเริ่มเปิดโปรแกรมคอนโทรลเลอร์จะทำการเปิด socket server เอาไว้เพื่อให้แอคเซสพอยต์ทำการเชื่อมต่อ เมื่อแอคเซสพอยต์ทำการเชื่อมต่อเข้ามาในระบบ คอนโทรลเลอร์จะทำการสร้าง thread เพื่อทำการเชื่อมต่อกับแอคเซสพอยต์นั้นๆ

  จากนั้นผู้ดูแลระบบจะทำหน้าที่ออกแบบเครือข่ายให้มีการทำงานตามที่ต้องการ จากนั้นอุปกรณ์คอนโทรลเลอร์จะทำหน้าที่แปลงคำสั่งเหล่านั้นให้อยู่ในรูปแบบของโปรโตคอลที่ใช้ในการสื่อสารระหว่างคอนโทรลเลอร์และแอคเซสพอยต์ แล้วส่งไปยังแอคเซสพอยต์ที่ตนเองควบคุมอยู่ผ่านทางอุปกรณ์เครือข่ายอื่นๆ เช่น สวิตช์ เพื่อทำการส่งต่อข้อมูลไปยังอุปกรณ์แอคเซสพอยต์ที่อยู่ภายในความควบคุมของผู้ดูแล

- **แอคเซสพอยต์**

ในส่วนของแอคเซสพอยต์ (B) ผู้วิจัยเลือกใช้งานเฟิร์มแวร์ OpenWRT เพื่อทำให้สามารถพัฒนาโปรแกรมบนแอคเซสพอยต์ได้อย่างอิสระ ซึ่งหลังจากที่แอคเซสพอยต์ได้ทำการเชื่อมต่อกับคอนโทรลเลอร์ผ่าน socket แล้ว แอคเซสพอยต์จะทำการรอรับคำสั่งจากคอนโทรลเลอร์

เมื่อได้รับคำสั่งจากคอนโทรลเลอร์แล้ว ซอฟต์แวร์ที่อยู่บนเฟิร์มแวร์จะทำหน้าที่ปรับตั้งค่าของระบบเพื่อตอบสนองเปลี่ยนแปลงการตั้งค่าของตนเองตามที่ผู้ดูแลระบบได้ออกแบบไว้ และทำการแจ้งกลับไปยังคอนโทรลเลอร์ว่าตนเองได้ทำการเปลี่ยนแปลงการตั้งค่าเรียบร้อยแล้ว แล้วจึงทำการรอคำสั่งถัดไป



รูปที่ 4.2 องค์ประกอบโดยรวมของระบบ

4.4 ขั้นตอนการพัฒนา

ศึกษาทฤษฎีและเทคโนโลยีที่ใช้ในการพัฒนาระบบ

- ศึกษาเทคโนโลยี SDN
- ศึกษาหลักการทำงานของ Openflow
- ศึกษาการใช้งาน OpenWRT
- ศึกษาคุณสมบัติพื้นฐานของแอคเซสพอยต์
- ศึกษาหลักการทำงานของ Wireless LAN Controller

ออกแบบระบบที่จะทำการพัฒนา

- ออกแบบการเชื่อมต่อระหว่างคอนโทรลเลอร์และแอคเซสพอยต์
- ออกแบบการทำงานของแอปพลิเคชั่นและคอนโทรลเลอร์
- ออกแบบโปรโตคอลที่ใช้ในการสื่อสารระหว่างคอนโทรลเลอร์และแอคเซสพอยต์
- ออกแบบคุณสมบัติต่างๆที่คอนโทรลเลอร์สามารถควบคุมได้

พัฒนาระบบ

- ติดตั้งโปรแกรมและไลบรารี่ที่จะใช้พัฒนา
- ทำการสร้าง kernel ที่รองรับ OpenWRT ให้กับแอคเซสพอยต์
- ทำการพัฒนาโปรแกรมส่วนคอนโทรลเลอร์
- ทำการพัฒนาโปรแกรมในส่วนแอคเซสพอยต์

ทดสอบการใช้งาน

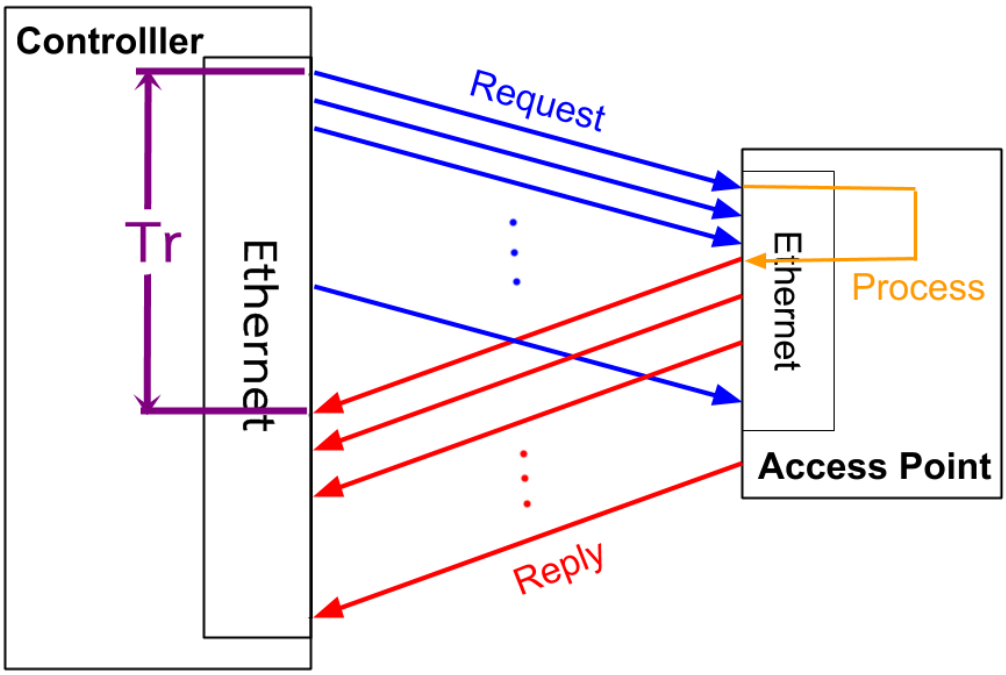
- ทดสอบการทำงานของแอคเซสพอยต์
- ทดสอบการเปลี่ยนแปลงการทำงานของแอคเซสพอยต์เมื่อได้รับคำสั่ง
- ทดสอบการตอบสนองผลการทำงานกลับมายังแอคเซสพอยต์

จัดทำเอกสารโครงงาน

- จัดทำรูปเล่มรายงาน
- จัดทำเอกสารคู่มือการติดตั้งและใช้งาน

5. ผลการดำเนินโครงงานและวิจารณ์

การทดสอบระบบแอคเซสพอยต์ที่รองรับ SDN จะทำการทดสอบโดยการทำการเชื่อมต่อแอคเซสพอยต์เข้ากับคอนโทรลเลอร์ และทำการจับเวลาที่แอคเซสพอยต์ใช้ในการตอบสนองต่อคำสั่งเมื่อผู้ดูแลเครือข่ายทำการส่งคำสั่งออกไป โดยจะนำเวลาที่แอคเซสพอยต์ตอบกลับมาถึงคอนโทรลเลอร์หักลบกับเวลาที่ส่งคำสั่งออกจากคอนโทรลเลอร์ จะได้เวลาทั้งหมดที่แอคเซสพอยต์ใช้เพื่อตอบสนองต่อคำสั่งจากคอนโทรลเลอร์ (Round trip time:Tr) ดังแสดงได้ในรูปที่ 5.1

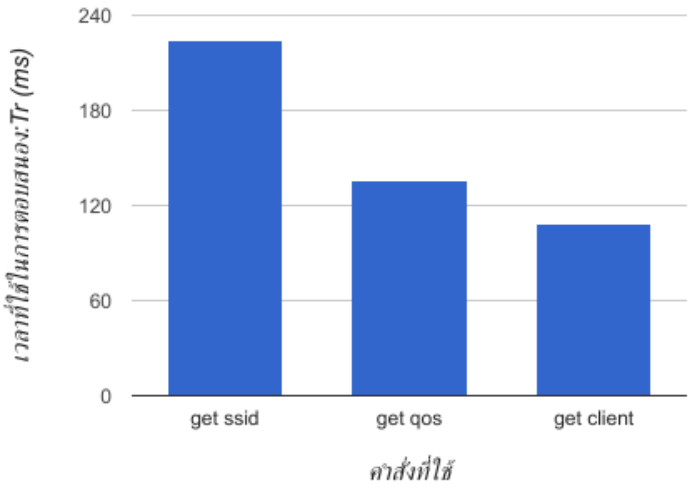


รูปที่ 5.1 เวลาการตอบสนองต่อคำสั่งจากคอนโทรลเลอร์ (Rt)

5.1 สภาพแวดล้อมในการทดสอบ

การทดสอบเวลาตอบสนองต่อคำสั่งชนิด get

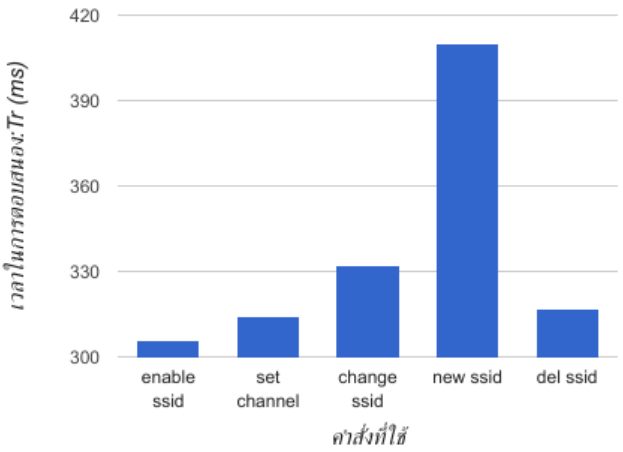
การทดสอบเวลาที่ใช้ในการตอบสนองต่อคำสั่งในรูปแบบของการร้องขอเพื่อทราบค่าสถานะของแอคเซสพอยต์และอินเตอร์เฟสต่างๆที่ใช้งานอยู่ โดยผู้วิจัยทดสอบโดยส่งคำสั่งต่างๆจากคอนโทรลเลอร์ เพื่อให้แสดงค่าสถานะพร้อมทั้งแสดงค่าบนคอนโทรลเลอร์ โดยใช้คำสั่ง 3 แบบ คือ 1) get ssid, 2) get qos และ 3) get client จากนั้นเริ่มจับเวลาที่คอนโทรลเลอร์ในขณะที่ทำการส่งคำสั่งออกไปยังแอคเซสพอยต์ จากนั้นเมื่อแอคเซสพอยต์ทำการตอบกลับค่าสถานะที่ทำการร้องขอไปมายังคอนโทรลเลอร์ ก็จะนำเวลา ณ ขณะนั้นมาหักลบกับเวลาตอนส่งคำร้องขอ เพื่อคำนวณเวลาที่ใช้ทั้งหมดในการตอบสนองของแอคเซสพอยต์ต่อคำสั่งชนิด get ดังแสดงได้ดังรูปที่ 5.2



รูปที่ 5.2 แผนภูมิแสดงเวลาที่ใช้ตอบสนองต่อคำสั่งชนิด get

การทดสอบเวลาตอบสนองต่อคำสั่งชนิด set

การทดสอบเวลาที่ใช้ในการตอบสนองต่อคำสั่งในรูปแบบของคำสั่งเพื่อเปลี่ยนแปลงการตั้งค่าของแอคเซสพอยต์และอินเตอร์เฟสต่างๆ โดยผู้วิจัยทดสอบโดยส่งคำสั่งต่างๆจากคอนโทรลเลอร์ เพื่อให้แอคเซสพอยต์เปลี่ยนแปลงการตั้งค่าของตนเอง โดยใช้คำสั่ง 5 แบบ คือ 1) enable ssid, 2) set channel, 3) change ssid, 4) new ssid และ 5) delete ssid จากนั้นเริ่มจับเวลาที่คอนโทรลเลอร์ในขณะที่ทำการส่งคำสั่งออกไปยังแอคเซสพอยต์ จากนั้นเมื่อแอคเซสพอยต์ทำการตอบกลับมายังคอนโทรลเลอร์ว่าตัวเองทำการเปลี่ยนแปลงค่าเรียบร้อยแล้ว ก็จะนำเวลา ณ ขณะนั้นมาหักลบกับเวลาตอนส่งคำสั่ง เพื่อคำนวณเวลาที่ใช้ทั้งหมดในการตอบสนองของแอคเซสพอยต์ต่อคำสั่งชนิด set ดังแสดงได้ดังรูปที่ 5.3

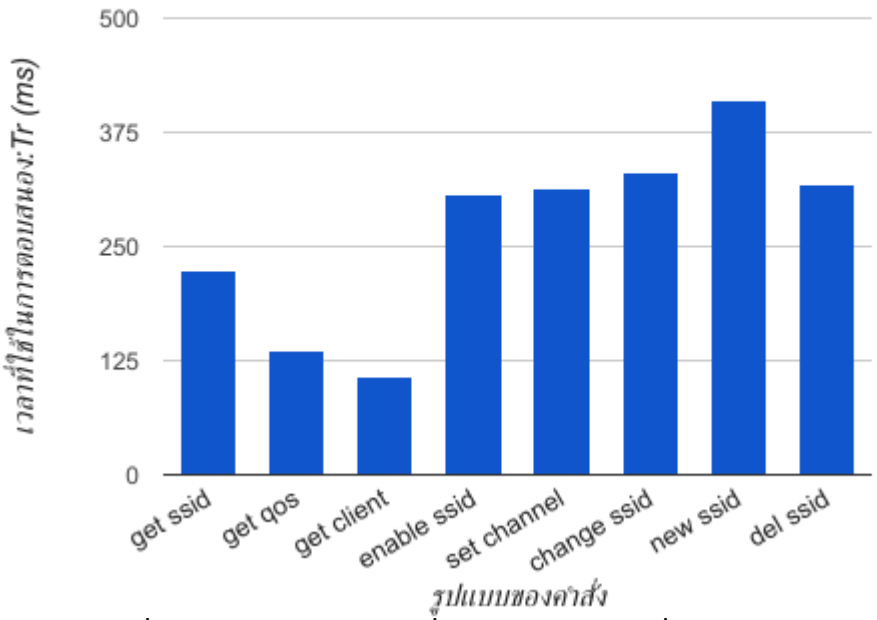


รูปที่ 5.3 แผนภูมิแสดงเวลาที่ใช้ตอบสนองต่อคำสั่งชนิด set

5.2 ผลการทดสอบและการวิจารณ์ผล

จากการทดสอบส่งคำสั่งชนิด get จะเห็นได้ว่าคำสั่ง get ssid นั้นใช้เวลาเฉลี่ยมากกว่าคำสั่ง get อื่นๆ เนื่องจากค่าตัวแปรของ ssid นั้นมีจำนวนที่มากกว่าตัวแปรของ qos และ client ทำให้ต้องใช้เวลาในการประมวลผลและส่งค่ากลับมานานกว่า เช่นเดียวกับการทดลองส่งคำสั่งชนิด set ที่คำสั่ง new ssid นั้นใช้เวลาเฉลี่ยมากกว่าคำสั่ง set อื่นๆ

และจากการผลการทดสอบส่งคำสั่งทั้งสองรูปแบบ ดังแสดงได้ตามรูปที่ 5.4 จะสังเกตได้ว่าเวลาที่ใช้ในการตอบสนองต่อคำสั่งชนิด get โดยเฉลี่ยรวมแล้วใช้เวลาน้อยกว่าคำสั่งชนิด set ซึ่งแอคเซสพอยต์จะต้องใช้เวลาในการแก้ไขค่าตัวแปรต่างๆภายใน และต้องทำการรีเซ็ตการเชื่อมต่อแบบไร้สายเพื่อให้การตั้งค่าที่ตั้งค่าไปสามารถแสดงผลได้อย่างถูกต้องทำให้เวลาในการตอบสนองสูงกว่าคำสั่งชนิด get ที่แอคเซสพอยต์จะทำการอ่านค่าตัวแปรต่างๆที่ใช้ในการตั้งค่ามาเท่านั้น ไม่จำเป็นต้องแก้ไขค่าและรีเซ็ตการเชื่อมต่อ



รูปที่ 5.4 แผนภูมิแสดงเวลาที่ใช้ตอบสนองต่อคำสั่งชนิดต่างๆ

6. สรุปผลการดำเนินงานและข้อเสนอแนะ

6.1 ข้อสรุป

การใช้งานแอคเซสพอยต์ที่รองรับ SDN นั้นสามารถทำงานตามที่คอนโทรลเลอร์สั่งการได้ ซึ่งทำให้ผู้ดูแลเครือข่ายสามารถออกแบบและปรับแต่งเครือข่ายไร้สายของตนเองได้อย่างสะดวกและรวดเร็วมากยิ่งขึ้น และยังช่วยลดความผิดพลาดที่เกิดจากความผิดพลาดของมนุษย์ได้ด้วย

ซึ่งระบบนี้สามารถรองรับแอคเซสพอยต์ที่แตกต่างผู้ผลิตกันได้เพียงแค่แอคเซสพอยต์นั้นสามารถลงซอฟต์แวร์ OpenWRT ได้ ก็จะสามารถนำมาใช้เพื่อขยายขนาดของเครือข่ายไร้สายได้ทันที โดยไม่ต้องเสียค่าใช้จ่ายในการติดตั้งคอนโทรลเลอร์เพิ่มแต่อย่างใด

ผู้พัฒนาได้พัฒนาในส่วนของซอฟต์แวร์ทั้งส่วนของคอนโทรลเลอร์และแอคเซสพอยต์ให้ตรงตามวัตถุประสงค์ที่ตั้งไว้ และได้พัฒนาให้การเชื่อมต่อระหว่างคอนโทรลเลอร์และแอคเซสพอยต์ให้สามารถใช้งานได้โดยไม่ข้องเกี่ยวกับเส้นทางการเดินทางของข้อมูลภายในเครือข่ายด้วย

6.2 ปัญหาและอุปสรรค

ปัญหาที่พบในระหว่างการพัฒนามีดังนี้

- หากเกิดความผิดพลาดในการแก้ไขการตั้งค่าที่ส่งผลทำให้การเชื่อมต่อกับแอคเซสพอยต์ถูกตัดขาด จะทำให้ไม่สามารถเข้าไปแก้ไขการตั้งได้อีกเลย
- ในกรณีที่ต้องมีการใช้ DHCP server แยกออกไป จะต้องทำการตั้งค่าใหม่ด้วยตัวเอง ยังไม่สามารถแก้ไขผ่านทางซอฟต์แวร์ได้
- ขั้นตอนการลง kernel ในแอคเซสพอยต์มีขั้นตอนที่มากมาย และเกิดข้อผิดพลาดได้ง่าย ซึ่งหากเกิดข้อผิดพลาด ก็มีโอกาสที่จะเปิดเครื่องแอคเซสพอยต์ไม่ติดไปด้วย
- ในการใช้งานจริง จำเป็นต้องแก้ไขค่าพื้นฐานบางส่วนของแอคเซสพอยต์ก่อน จึงจะสามารถใช้งานซอฟต์แวร์ได้อย่างเต็มประสิทธิภาพ
- หน่วยประมวลผลของเครื่องที่ใช้งานเป็นคอนโทรลเลอร์ส่งผลต่อความสามารถในการทำงานด้วย เพราะจำเป็นต้องใช้ thread 1 thread ต่อการเชื่อมต่อ 1 การเชื่อมต่อ

6.3 แนวทางการพัฒนาต่อ

แนวทางในการพัฒนาแอคเซสพอยต์ให้รองรับ SDN มีดังนี้

- พัฒนาให้แอคเซสพอยต์สามารถรับค่าการตั้งค่าได้เลย โดยไม่ต้องไปแก้ไขค่าพื้นฐานก่อนในขั้นตอนการติดตั้ง
- พัฒนา protocol ให้สามารถใช้งานกับ protocol มาตรฐานได้
- พัฒนาให้สามารถใช้ในกรณีที่มี DHCP server เชื่อมต่ออยู่ด้วยได้
- พัฒนาให้คอนโทรลเลอร์สามารถควบคุมแอคเซสพอยต์ให้ครอบคลุมถึงคุณสมบัติต่างๆของเครือข่ายไร้สายได้มากขึ้น
- พัฒนาหน้าจอ User interface ให้สามารถใช้งานได้ง่ายขึ้น
- หรือพัฒนาหน้าเว็บขึ้นมาเพื่อเป็นส่วนแอปพลิเคชั่นสำหรับให้ผู้ดูแลเครือข่ายสามารถใช้งานได้ง่ายขึ้น

6.4 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับผู้ที่จะทำการพัฒนาโครงงานมีดังนี้

- ศึกษาและฝึกพัฒนาโปรแกรมภาษาไพธอนเพื่อให้สามารถพัฒนาโปรแกรมได้อย่างรวดเร็ว
- ศึกษาทฤษฎีการทำงานของ SDN ให้เข้าใจก่อนลงมือพัฒนา
- ศึกษาขั้นตอนการประยุกต์ใช้โปรโตคอล OpenFlow มาตรฐาน
- ศึกษาการพัฒนาคอนโทรลเลอร์บนโปรโตคอลมาตรฐาน เช่น Ryu, POX, Floodlight โดยแนะนำให้ทดลองศึกษา Ryu เพราะเป็นเฟรมเวิร์คที่ใช้ภาษา python ทำให้สามารถประยุกต์ใช้กับโครงงานนี้ได้อย่างสะดวก
- เพิ่มการทดลองให้หลากหลายและครอบคลุมการทำงาน เพื่อให้ได้ผลลัพท์ที่มีข้อผิดพลาดน้อยที่สุด

7. บรรณานุกรม

[1] “SDN definition” [ออนไลน์] เข้าถึงได้ ณ วันที่ 29 พฤษภาคม 2559: https://www.opennetworking.org/sdn-resources/sdn-definition

[2] “Wireleass LAN Controller (WLC) FAQ” [ออนไลน์] เข้าถึงได้ ณ วันที่ 29 พฤษภาคม 2559: http://www.cisco.com/c/en/us/support/docs/wireless/4400-series-wireless-lan-controllers/69561-wlc-faq.html

[3] “About OpenWRT” [ออนไลน์] เข้าถึงได้ ณ วันที่ 29 พฤษภาคม 2559: https://wiki.openwrt.org/

[4] “What is OpenFlow” [ออนไลน์] เข้าถึงได้ ณ วันที่ 29 พฤษภาคม 2559: https://www.sdxcentral.com/sdn/definitions/what-is-openflow/

[5] K. Wang, Y. Wang, D. Zeng and S. Guo, "An SDN-Based Architecture for Next-Generation Wireless Networks," in IEEE Wireless Communications, vol. 24, no. 1, pp. 25-31, February 2017.

[6] Nick McKeown; et al. (April 2008). "OpenFlow: Enabling innovation in campus networks". ACM Communications Review. Retrieved 2009-11-02.

[7] “System-specific parameters and functions” [ออนไลน์] เข้าถึงได้ ณ วันที่ 29 พฤษภาคม 2559: https://docs.python.org/2/library/sys.html

[8] “Regular Expression Operation” [ออนไลน์] เข้าถึงได้ ณ วันที่ 29 พฤษภาคม 2559: https://docs.python.org/2/library/re.html

[9] “Time access and conversion” [ออนไลน์] เข้าถึงได้ ณ วันที่ 29 พฤษภาคม 2559: https://docs.python.org/2/library/time.html

[10] “Subprocess management” [ออนไลน์] เข้าถึงได้ ณ วันที่ 29 พฤษภาคม 2559: https://docs.python.org/2/library/subprocess.html

8. ภาคผนวก

8.1 คู่มือการติดตั้ง

ในการติดตั้งระบบนี้แบ่งได้เป็นสองส่วนคือส่วนของคอนโทรลเลอร์และส่วนของแอคเซสพอยต์

ส่วนคอนโทรลเลอร์

- ทำการติดตั้ง python ในเครื่องที่ใช้ระบบปฏิบัติการ Ubuntu ด้วยคำสั่ง

  sudo apt-get install python
- ทำการดาวน์โหลดไฟล์ mycontroller.py และทำการ save ไว้ในพื้นที่ที่ต้องการในเครื่องที่จะใช้เป็นคอนโทรลเลอร์
- แก้ไข ip address ของคอนโทรลเลอร์ให้กลายเป็น ip ที่ไม่ได้ใช้งานในเครือข่ายหรือเป็น ip ที่เตรียมไว้สำหรับคอนโทรลเลอร์โดยเฉพาะ
- เริ่มการทำงานของคอนโทรลเลอร์โดยใช้คำสั่ง

  python mycontroller.py

ส่วนของแอคเซสพอยต์

- ทำการติดตั้ง git ในเครื่องที่ใช้ระบบปฏิบัติการ Ubuntu ด้วยคำสั่ง

  sudo apt-get install git
- ทำการดาวน์โหลดไดเรคทอรี่ OpenWrt
- จากนั้นให้เข้าไปยังไดเรคทอรี่ของ Openwrt ด้วยคำสั่ง

  cd OpenWrt
- ทำการติดตั้งส่วนเสริมให้ครบ gcc, binutils, bzip2, flex, python, perl, make, find, grep, diff, unzip, gawk, getopt, subversion, libz-dev and libc headers โดยใช้คำสั่ง

  apt-get install ...
- ทำการอัพเดต OpenWrt ด้วยคำสั่ง

  ./scripts/feeds update -a

  ./scripts/feeds install -a
- จากนั้นให้ทำการแก้ไขรุ่นของแอคเซสพอยต์ที่จะใช้ด้วยคำสั่ง

  make menuconfig

  แล้วเลือก target system เป็นรุ่นของแอคเซสพอยต์ที่ต้องการ

และสามารถเลือก packet ที่จะติดตั้งเพิ่มเติมได้จากหน้าต่างนี้

จากนั้นกด Save และ Exit ออกมาที่ terminal

- ทำการสร้าง kernel ให้กับแอคเซสพอยต์ ด้วยคำสั่ง

  make

  จะได้ไฟล์ .bin ซึ่งเป็น binary file สำหรับใช้ติดตั้งในแอคเซสพอยต์
- จากนั้นทำการติดตั้งไฟล์ .bin ลงไปในแอคเซสพอยต์ที่ต้องการ โดยวิธีการลง binay file นั้นจะแตกต่างกันไปแล้วแต่รุ่นและผู้ผลิตของแอคเซสพอยต์นั้นๆ
- ทำการเชื่อมต่อเข้ากับแอคเซสพอยต์และทำการ telnet เข้าไปยัง ip 192.168.1.1 เพื่อทำการตั้งค่า password สำหรับทำ ssh

  ตั้ง password โดยใช้คำสั่ง

  passwd
- ทำการ secure shell ไปยังแอคเซสพอยต์และทำการแก้ไขไฟล์พื้นฐานใน /etc/config ให้มี format ตรงกับไฟล์ wireless, qos, dhcp. network, firewall
- ทำการดาวน์โหลดไฟล์ myap.py และทำการแก้ไข ip address ของ host ในไฟล์ให้เป็น ip address ของคอนโทรลเลอร์
- วิธีนำไฟล์ myapp.py มาติดตั้งในแอคเซสพอยต์สามารถทำได้ด้วยคำสั่ง

  scp myap.py user@accesspoint_ipaddr
- เริ่มการทำงานของแอคเซสพอยต์ด้วยคำสั่ง

  python myap.py

8.2 คู่มือการใช้งาน

ผู้ใช้งานสามารถควบคุมแอคเซสพอยต์ที่เชื่อมต่ออยู่ได้ ผ่านคอนโทรลเลอร์ โดยหน้าคอนโทรลเลอร์จะแสดงถึงแอคเซสพอยต์ที่เชื่อมต่อเข้ามา และแสดงถึงฟังก์ชั่นต่างๆที่คอนโทรลเลอร์สามารถทำได้

และผู้ใช้สามารถเลือกได้ด้วยการพิมพ์ตัวเลขของหัวข้อ แล้วกดปุ่ม enter ก็จะสามารถเข้าใช้ฟังก์ชันต่างๆได้ โดยแบ่งออกเป็น

[1] Enable/Disable Wifi เป็นการเลือกเปิดหรือปิดการใช้งาน wifi interface ต่างๆ โดยจะมีข้อมูลแสดงให้ผู้ใช้เลือกว่าจะเปิดหรือปิด interface ไหน โดยบอกรายละเอียดของ interface นั้นๆ และเลือกได้ว่าจะทำการเปิดใช้งานหรือปิดการใช้งาน

[2] Change Channel เป็นการเลือกเปลี่ยน channel ของคลื่นความถี่ 2.4 GHz ที่สามารถเปลี่ยนได้ตั้งแต่ 1-11

[3] Change SSID Name เป็นการเปลี่ยนชื่อของ SSID ที่มีอยู่ ให้เป็นชื่อ SSID ใหม่ โดยจะมีข้อมูลแสดงให้ผู้ใช้เลือกว่าจะแก้ไข SSID ของ interface ไหน
[4] Create new SSID เป็นการสร้าง SSID ใหม่ขึ้นมาใช้งาน โดยสามารถเลือกได้ว่าจะสร้างบนความถี่อะไร มีไว้สำหรับ guess รึเปล่า และใช้ SSID อะไร

[5] Delete current SSID เป็นการเลือกลบ interface ที่ไม่ใช้งานแล้ว โดยจะมีข้อมูลแสดงให้ผู้ใช้เลือกว่าลบ interface ไหน โดยบอกรายละเอียดของ interface นั้นๆ

[6] Show Wireless Status เป็นการแสดงสถานะของ interface ต่างๆ โดยจะแสดง SSID ความถี่เครือข่ายที่ใช้งาน การเข้ารหัส และสถานะการใช้งาน

[7] Bandwidth Customise เป็นการเลือกปรับอัตราการส่งถ่ายข้อมูลได้ โดยสามารถเลือกเครือข่ายที่จะตั้งค่า และตั้งค่าอัตราการ upload และ download ให้กับเครือข่ายนั้นๆได้

[8] Show Bandwidth Cumtomise เป็นการแสดงการตั้งค่าอัตราการส่งถ่ายข้อมูลของแต่ละเครือข่าย รวมทั้งสถานะการใช้งานด้วย

[9] Show Connecting Client เป็นการแสดงรายละเอียดเกี่ยวกับเครื่อง Client ที่เชื่อมต่ออยู่กับแอคเซสพอยต์ตัวนั้นๆ ทั้ง Mac address ความเร็วในการเชื่อมต่อ และความแรงของสัญญาน

9. ประวัตินิสิต

นายสิรภพ สัตตบงกช เลขประจำตัวนิสิต 5610504007

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ที่อยู่ปัจจุบัน 7/96 หมู่ 2 ต.บางพูด อ.ปากเกร็ด จ.นนทบุรี 11120

โทรศัพท์ 02-964-5064

โทรศัพท์มือถือ 087-510-1505

E-mail siraphob.sattabongkot@gmail.com

ระดับการศึกษา ปริญญาตรี

คุณวุฒิการศึกษา	**สถาบัน**	**ปีการศึกษาที่จบ**
มัธยมศึกษาตอนปลาย	โรงเรียนสวนกุหลาบวิทยาลัยนนทบุรี	2555
มัธยมศึกษาตอนต้น	โรงเรียนสวนกุหลาบวิทยาลัยนนทบุรี	2552