

รายงานการปฏิบัติสหกิจศึกษา

เรื่องการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์

SOFTWARE DEFINED NETWORK

ปฏิบัติงาน ณ บริษัท แอ็ดวานซ์ อินฟอร์เมชัน เทคโนโลยี จำกัด (มหาชน)

โดย

นายจิรสิน ปัญญาวิสุทธิชัย

รหัสประจำตัว 57070020

รายงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา สหกิจศึกษา
สาขาวิชา เทคโนโลยีสารสนเทศ คณะ เทคโนโลยีสารสนเทศ
ภาคการศึกษาที่ 1 ปีการศึกษา 2560
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายงานการปฏิบัติสหกิจศึกษา

เรื่อง การควบคุมระบบเครือข่ายด้วยซอฟต์แวร์
SOFTWARE DEFINED NETWORK

โดย

นายจิรสิน ปัญญาวิสุทธิชัย
รหัสประจำตัว 57070020

อาจารย์ที่ปรึกษา
ดร.ลภัส ประดิษฐ์ทัศนีย์

ปฏิบัติงาน ณ บริษัท แอ็ดวานซ์อินฟอร์เมชัน เทคโนโลยี จำกัด (มหาชน)
เลขที่ 37/2 ถนนสุขุมวิท แขวงสามเสนนอก
เขตห้วยขวาง จังหวัดกรุงเทพมหานคร รหัสไปรษณีย์ 10320
โทรศัพท์ 022759400 โทรสาร 022759100
Website: www.ait.co.th

SOFTWARE DEFINED NETWORK

JIRASIN PUNYAWISUTTICHAJ

**A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR COOPERATING EDUCATION PROGRAM
THE DEGREE OF BACHELOR OF SCIENCE PROGRAM IN
INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

1/ 2017

COPYRIGHT 2017

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

วันที่ 29 กันยายน พ.ศ 2560

เรื่อง ขอส่งรายงานการปฏิบัติงานสหกิจศึกษา

เรียน ดร.ถกัส ประดิษฐ์ทัศนีย์

ที่ปรึกษาสหกิจศึกษาในสาขา เทคโนโลยีสารสนเทศ

ที่ข้าพเจ้านาย จิรสิน ปัญญาวิสุทธิชัย นักศึกษาสาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้ปฏิบัติงานสหกิจศึกษา ระหว่างวันที่ 1 เดือนมิถุนายน พ.ศ. 2560 ถึงวันที่ 30 เดือน พฤศจิกายน พ.ศ. 2560

ในตำแหน่ง Project Engineer – Installation ณ สถานประกอบการชื่อ บริษัท แอ็ดวานซ์ อินฟอร์เมชั่น เทคโนโลยี จำกัด (มหาชน) และได้รับมอบหมายจากพนักงานที่ปรึกษาให้ศึกษาและจัดทำรายงานเรื่อง การควบคุมระบบเครือข่ายด้วยซอฟต์แวร์

บัดนี้ การปฏิบัติงานสหกิจศึกษาได้สิ้นสุดลงแล้ว จึงใคร่ขอส่งรายงานการปฏิบัติงาน สหกิจศึกษาดังกล่าวมาพร้อมนี้ จำนวน 1 เล่ม เพื่อขอรับคำปรึกษาต่อไป

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

(นายจิรสิน ปัญญาวิสุทธิชัย)

กิตติกรรมประกาศ

ตามที่ข้าพเจ้า นาย จิรสิน ปัญญาวิสุทธิชัย ได้มาปฏิบัติงานสหกิจศึกษา ณ บริษัท แอ็ดวานซ์ อินฟอร์เมชั่น เทคโนโลยี จำกัด (มหาชน) ตั้งแต่วันที่ 1 มิถุนายน พ.ศ. 2560 ถึงวันที่ 30 พฤษภาคม พ.ศ. 2560 ทำให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่าง ๆ ที่มีคุณค่ามากมาย สำหรับรายงานสหกิจศึกษานี้สำเร็จลงได้ด้วยดี จากความช่วยเหลือและความร่วมมือสนับสนุนของหลายฝ่าย ดังนี้

- | | |
|--------------------------------------|--|
| 1. คุณ สิริลักษณ์ ปาลกะวงศ์ ณ อยุธยา | ตำแหน่ง Assistant Vice President – Human Resources |
| 2. คุณ จักรพงษ์ ศิริพลตัน | พนักงานที่ปรึกษา |
| 3. คุณ ภาณุวัฒน์ เบญจปฐมรงค์ | ตำแหน่ง Project Engineer |
| 4. คุณ วรุตม์ กุศลชู | ตำแหน่ง Project Engineer |
| 5. คุณ ศุภฤกษ์ บุรณะชีวิน | ตำแหน่ง Project Engineer |

นอกจากนี้ยังมีบุคคลท่านอื่น ๆ อีกที่ไม่ได้กล่าวไว้ ณ ที่นี้ ซึ่งให้ความกรุณาแนะนำในจัดทำรายงานสหกิจศึกษานี้ ข้าพเจ้าจึงใคร่ขอขอบพระคุณทุกท่านที่ได้มีส่วนร่วมในการให้ข้อมูลและให้ความเข้าใจเกี่ยวกับชีวิตของการปฏิบัติงาน รวมถึงเป็นที่ปรึกษาในการจัดทำรายงานฉบับนี้จนเสร็จสมบูรณ์

นายจิรสิน ปัญญาวิสุทธิชัย
ผู้จัดทำรายงาน

วันที่ 29 กันยายน พ.ศ. 2560

ชื่อรายงานการปฏิบัติงานสหกิจศึกษา

การควบคุมระบบเครือข่ายด้วยซอฟต์แวร์

ผู้รายงาน

นายจิรสิน ปัญญาวิสุทธิชัย รหัสนักศึกษา 57070020

คณะ

เทคโนโลยีสารสนเทศ

สาขาวิชา

เทคโนโลยีสารสนเทศ

(ดร.ลภัส ประดิษฐ์ทัศนีย์)

อาจารย์ที่ปรึกษาสหกิจศึกษา

(จักรพงษ์ ศิริพลรัตน์)

พนักงานที่ปรึกษา

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับรายงานการปฏิบัติงานสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ชื่อรายงาน	การควบคุมระบบเครือข่ายด้วยซอฟต์แวร์
ชื่อนักศึกษา	นายจิรสิน ปัญญาวิสุทธิชัย
รหัสนักศึกษา	57070020
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษา	ดร.ลภัส ประดิษฐ์ทัศนีย์
ปีการศึกษา	2560

บทคัดย่อ

รายงานฉบับนี้นำเสนอถึงแนวคิดการทำงานของการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ และการทำงานอื่นๆที่ได้รับมอบหมาย จากการไปปฏิบัติสหกิจศึกษาในตำแหน่ง Project Engineer ณ บริษัท แอ็ดวานซ์ อินฟอร์เมชั่น เทคโนโลยี จำกัด (มหาชน) ซึ่งเป็นบริษัทที่ดำเนินธุรกิจในรูปแบบ ซิสเต็มส์อินทิเกรเตอร์

โดยที่การควบคุมระบบเครือข่ายด้วยซอฟต์แวร์นั้นเป็นการเปลี่ยนแปลงการจัดการระบบเครือข่ายแบบดั้งเดิมให้สามารถควบคุมการส่งข้อมูลจากอุปกรณ์หลักเพียงตัวเดียวและยังส่งผลให้การจัดการในระบบต่างๆสามารถทำได้ง่ายยิ่งขึ้น โดยการใช้แนวคิดอันนี้มาใช้ในองค์กรต่างๆจึงเป็นสิ่งที่เหมาะสมเพราะจะสามารถตรวจสอบและลดขั้นตอนการทำงานไปได้ในหลายส่วน

Project Title	Software Defined Network
Student	Mr. Jirasin Punyawisuttichai
StudentID	57070020
Program	Information Technology
Advisor	Dr.Lapas Pradittasnee
Year	2017

ABSTRACT

This report present my work as a Project Engineer with Software Defined Network or SDN and other task at Advance Information Technology PLC. In correspondence with the Cooperative Education subject

Software Defined Network is a concept that will change management in traditional network to let the software control sending information from all device by manage on core device and it will decrease management work for network engineer and with SDN concept many organization should use this things for decrease process of verifying and working

สารบัญ

หน้า

จดหมายนำส่ง.....	I
กิตติกรรมประกาศ.....	II
หน้าอนุมัติรายงาน.....	III
บทคัดย่อภาษาไทย.....	IV
บทคัดย่อภาษาอังกฤษ.....	V
สารบัญ	VI
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ	1
1.1 ความมุ่งหมายและวัตถุประสงค์.....	2
1.2 ประวัติ และรายละเอียดบริษัท	2
บทที่ 2 รายละเอียดของงานที่ปฏิบัติ	6
2.1 ตำแหน่ง/หน้าที่ของงานที่ได้รับมอบหมาย	6
2.2 รายละเอียดของโครงการที่ได้รับผิดชอบ	6
2.3 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้องกับโครงการ	6
บทที่ 3 การดำเนินงานวิจัย	16
3.1 ทำการจำลองระบบเครือข่ายลงในโปรแกรมจำลอง	16
3.2 Configuration เบื้องต้นให้แก่อุปกรณ์แต่ละตัว	17
3.3 ทำการจำลองแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์.....	27
บทที่ 4 สรุปผลการปฏิบัติงาน	41
4.1 สรุปผลการปฏิบัติงาน	41
4.2 ประโยชน์ที่ได้รับ	42
บทที่ 5 ปัญหาและข้อเสนอแนะ	43
บรรณานุกรม	44
ภาคผนวก	45
ประวัติผู้เขียน	58

สารบัญตาราง

ตารางที่	หน้า
3.1 ตารางไอพีและอินเทอร์เฟซ.....	17
3.2 ตารางข้อมูล Configuration RouterXR1	19
3.3 ตารางข้อมูล Configuration Router2.....	23
3.4 ตารางข้อมูล Configuration Router3.....	26
3.5 ตารางข้อมูล Configuration OpenvSwitch1	27
3.6 ตารางข้อมูล Configuration OpenvSwitch2.....	27
3.7 ตารางคำสั่ง Ingress Policy ของ Eth1 ใน OpenvSwitch1.....	36
3.8 ตารางคำสั่ง Egress Policy ของ Eth1 ใน OpenvSwitch1	37

สารบัญรูป

รูปที่	หน้า
1.1 โครงสร้างการดำเนินงานของบริษัท	3
1.2 ขั้นตอนการดำเนินงานของบริษัท.....	3
1.3 โครงสร้างองค์กรของบริษัท.....	4
2.1 ภาพตัวอย่างโครงสร้างของแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์.....	8
2.2 ภาพตัวอย่างการทำงานของ Northbound และ Southbound API	10
2.3 ภาพตัวอย่างการทำงานของ Openflow Manager	12
2.4 ภาพตัวอย่างการทำงานของ Flow Table.....	13
2.5 ภาพตัวอย่างโครงสร้างของ OpenDaylight.....	13
2.6 ภาพตัวอย่างการทำงานของ OpenFlow Pipeline	14
2.7 ภาพตัวอย่างการทำงานของ Hybrid OpenFlow Pipeline.....	15
3.1 ภาพตัวอย่างระบบโครงข่ายใน GNS3.....	17
3.2 ภาพตัวอย่างการเข้าใช้งาน OpenDaylight	28
3.3 ภาพตัวอย่าง Topology ใน OpenDaylight	28
3.4 ภาพตัวอย่างสถิติข้อมูลที่เก็บอยู่ใน OpenVirtualSwitch1	29
3.5 ภาพตัวอย่าง Topology ใน Openflow Manager	29
3.6 ภาพตัวอย่าง Host ใน Openflow Manager	30
3.7 ภาพตัวอย่าง RouterXR Ping ไปยัง IP 8.8.8.8.....	30
3.8 ภาพตัวอย่างการดักจับข้อมูลโดยใช้ Wireshark.....	30
3.9 ภาพตัวอย่างการสร้าง Flow ใน Openflow Manager	31
3.10 ภาพตัวอย่าง Flow หลังจากสร้าง Flow เสร็จสิ้น	32
3.11 ภาพตัวอย่าง Flow Drop ข้อมูลใน OpenSwitch	32
3.12 ภาพตัวอย่าง RouterXR Ping ไปยัง IP 8.8.8.8 หลังจาก เพิ่ม Flow ลงไป.....	33
3.13 ภาพตัวอย่างการดักจับข้อมูลโดยใช้ Wireshark หลังจากเพิ่ม Flow ลงไป.....	33
3.14 ภาพตัวอย่างการสร้าง Flow ในการเปลี่ยนเส้นทางของข้อมูล	34
3.15 ภาพตัวอย่าง Flow ในการเปลี่ยนเส้นทางของข้อมูล OpenvSwitch1	34
3.16 ภาพตัวอย่างการ Ping จาก Ubuntu2 ไปยัง 203.151.13.166	35
3.17 ภาพตัวอย่างการ Ping จาก Ubuntu2 ไปยัง 8.8.8.8	35
3.18 ภาพตัวอย่างการ Wireshark ของ Interface RouterXR1 Gig0/0/0/0	35
3.19 ภาพตัวอย่างการ Ingress Policy ใน OpenVSwitch 1.....	36

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.20 ภาพตัวอย่างรายละเอียดการส่งข้อมูลของ Ubuntu2	37
3.21 ภาพตัวอย่างรายละเอียด Eth2 ของ OpenvSwitch1	37
3.22 ภาพตัวอย่าง QoS Policy ใน OpenVswitch1	38
3.23 ภาพตัวอย่าง QoS Policy ใน OpenVswitch1	38
3.24 ภาพตัวอย่าง Flow QoS.....	39
3.25 ภาพตัวอย่าง Flow QoS ใน OpenvSwitch1	39
3.26 ภาพตัวอย่างรายละเอียดการส่งข้อมูลของ Ubuntu2 หลังจากกาทำ QoS	40

บทที่ 1

บทนำ

บริษัทแอ็ดวานซ์ อินโฟเมชั่น เทคโนโลยี จำกัด (มหาชน) นั้นก่อตั้งขึ้นในปี 2535 โดยวัตถุประสงค์ประกอบธุรกิจเกี่ยวกับเทคโนโลยีสารสนเทศและการสื่อสาร โดยลักษณะธุรกิจจะเป็นเกี่ยวกับซิสเต็มส์อินทิเกรเตอร์ (System Integrator) หรือ “SI” หรือก็คือเป็นผู้รวบรวมระบบคอมพิวเตอร์ ระบบสื่อสารคอมพิวเตอร์ และอุปกรณ์คอมพิวเตอร์ให้สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพและประสิทธิผลซึ่งต้องตรงต่อความต้องการของลูกค้า

เนื่องจากตัวผมได้ทำเกี่ยวข้องกับทีมของทางฝั่งผู้ให้บริการทางอินเทอร์เน็ต (Internet Service Provider : ISP) ซึ่งจะคอยให้บริการแก่ลูกค้ามากมายทั้งเรื่องของความเร็ว ความสำคัญของข้อมูลโดยทางผู้ให้บริการก็จะใช้โปรโตคอลที่เรียกว่าเอ็มพีแอลเอส (Multiprotocol Label Switching : MPLS) เพื่อเพิ่มความเร็วในการส่งข้อมูลโดยใช้ป้ายบอกเส้นทางจากจุดหนึ่งไปยังอีกจุดหนึ่งเพื่อลดภาระที่อุปกรณ์จะต้องไปหาเส้นทางในตารางข้อมูล รวมทั้งใช้งานกับโปรโตคอลการค้นหาเส้นทาง (Routing Protocol) เช่น OSPF (Open Shortest Path First) หรือ IS-IS (Intermediate System To Intermediate System) เพื่อให้ผู้ใช้งานแต่ละที่สามารถติดต่อสื่อสารกันได้และมีการนำโปรโตคอล EGP (Exterior Gateway Protocol) ซึ่งก็คือ BGP (Boarder Gateway Protocol) เพื่อทำการแลกเปลี่ยนข้อมูลกับเครือข่ายของผู้ให้บริการเจ้าอื่นๆ ซึ่งปัญหาบางครั้งก็ในการที่เราต้องการจะปรับเปลี่ยนการให้บริการต่างๆของอุปกรณ์หรือเปลี่ยนเส้นทางของเครือข่ายอาจจะจำเป็นต้องแก้ไขการทำงานของอุปกรณ์หลายตัวซึ่งอาจจะทำให้เกิดส่งผลกระทบต่อตัวระบบได้และยังมีปัญหาของผู้ผลิตแต่ละเจ้าซึ่งจะทำให้การแก้ไขหรือปรับเปลี่ยนของอุปกรณ์แตกต่างกันทำให้อาจจะเกิดความสับสนของวิศวกรได้ และด้วยการที่อุปกรณ์นั้นอาจจะมีจำนวนที่มากทำให้ต้องใช้เวลาในการหาอุปกรณ์ดังกล่าว รวมทั้งปัญหาเอกสารต่างๆไม่ได้เป็นปัจจุบันก็อาจจะทำให้รูปของระบบเครือข่ายไม่ตรงกับที่เห็นอีกด้วย

ทางตัวผมจึงได้เห็นความสำคัญของปัญหาจึงได้ทำการศึกษาเกี่ยวกับแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ที่เมื่อนำไปติดตั้งในระบบแล้วเราสามารถรู้โครงสร้างเครือข่ายของระบบได้รวมทั้งเราสามารถควบคุมการส่งข้อมูลของอุปกรณ์เครือข่ายได้อย่างอิสระ โดยทั้งหมดนี้นั้นสามารถทำผ่านอุปกรณ์หลักที่เรียกว่า ส่วนควบคุมกลาง (Controller) เพียงตัวเดียวเท่านั้นและการควบคุมการส่งข้อมูลนั้นจะส่งผลต่ออุปกรณ์ที่เรากำหนดเท่านั้น ทำให้การแก้ไขหรือปรับเปลี่ยนอุปกรณ์ในระบบก็จะทำได้โดยง่ายและสามารถดูข้อมูลสถิติเบื้องต้นของ

อุปกรณ์เครือข่ายได้อีกด้วย ซึ่งอีกอย่างที่สำคัญคือแนวคิดนี้นั้นเป็นมาตรฐานกลางทำให้ใช้ได้กับผู้ผลิตทุกเจ้า ช่วยลดปัญหาความแตกต่างในการแก้ไขข้อมูลในอุปกรณ์

1.1 ความมุ่งหมายและวัตถุประสงค์

1. เพื่อศึกษากระบวนการทำงานและแนวคิดของการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์และสามารถนำไปใช้งานจริงได้
2. เพื่อศึกษากระบวนการทางธุรกิจและกระบวนการทำงานต่างๆในบริษัท
3. เพื่อเข้าใจถึงปัญหาที่เกิดขึ้นในการทำงานของบริษัท
4. เพื่อศึกษาถึงลักษณะการทำงานของบุคลากรในบริษัท
5. เพื่อนำทฤษฎีที่ศึกษามาใช้ประกอบการใช้งานจริง

1.2 ประวัติ และรายละเอียดบริษัท

ชื่อบริษัท(ภาษาไทย) : บริษัทแอ็ดวานซ์ อินโฟเมชั่น เทคโนโลยี จำกัด (มหาชน)

สถานที่ตั้ง : 37/2 ถนนสุทธิสาร แขวงสามเสนนอก เขตห้วยขวาง 10320 กรุงเทพมหานคร

ระยะเวลาที่ปฏิบัติงาน : 1 มิถุนายน 2560 – 18 พฤศจิกายน 2560

ลักษณะธุรกิจ

บริษัท แอ็ดวานซ์อินฟอร์เมชั่นเทคโนโลยี จำกัด (มหาชน) เป็นผู้นำในการให้บริการเทคโนโลยีสารสนเทศและการสื่อสารโทรคมนาคม อย่างครบวงจร ตั้งแต่ การให้คำปรึกษา ออกแบบ ขยายและติดตั้ง พัฒนาระบบงาน และการซ่อมบำรุงรักษาเครื่องอุปกรณ์ (Maintenance Services) การให้เช่าใช้ การเหมารวมระบบ (Turnkey projects) การบริหารจัดการซ่อมบำรุงรักษา (ICT Outsourcing) ICT (Information and Communication Technology) คือการรวมกันของเทคโนโลยีสารสนเทศ (Information Technology - IT) ซึ่งหมายถึงการใช้ประโยชน์ของคอมพิวเตอร์ การจัดเก็บข้อมูล และเทคโนโลยีเครือข่าย เพื่อนำมาประมวลผล ข้อมูล รวมกับเทคโนโลยีการสื่อสาร เช่น เครือข่ายโทรศัพท์ เครือข่ายสายโทรศัพท์พื้นฐาน การสื่อสารทั้งภาพและเสียง เป็นต้น

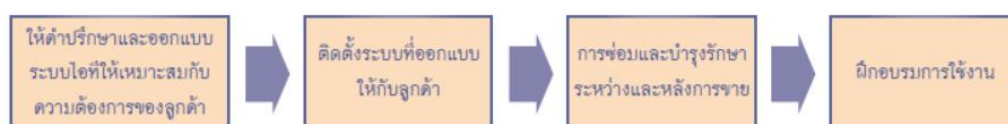
ภาพรวมธุรกิจ

บริษัทเรียกตัวเองว่าเป็น Systems Integrator ซึ่งหมายถึง ผู้รวบรวมระบบและอุปกรณ์ต่างๆ ที่เกี่ยวข้องกับเทคโนโลยีสารสนเทศและการสื่อสารเข้ามาประยุกต์ใช้ให้เหมาะสมกับความต้องการของแต่ละองค์กร ซึ่งการรวบรวมดังกล่าว เรียกว่า Solution หรือระบบเบ็ดเสร็จ ที่จะเพื่อช่วยให้การบริหารจัดการองค์กรมีประสิทธิภาพมากยิ่งขึ้น



รูป 1.1 โครงสร้างการดำเนินงานของบริษัท

ในฐานะที่เป็น System Integrator และ ผู้ให้บริการเทคโนโลยีสารสนเทศและการสื่อสาร (ICT Provider) บริษัทให้บริการอย่างครบวงจร ตั้งแต่ให้คำปรึกษา ออกแบบ ติดตั้ง บริหาร โครงการ ซ่อมและบำรุงรักษา รวมไปถึงฝึกอบรมการใช้งานโดยกระบวนการนี้สามารถแสดงได้ตามแผนภูมิด้านล่าง



รูปที่ 1.2 ขั้นตอนการดำเนินงานของบริษัท

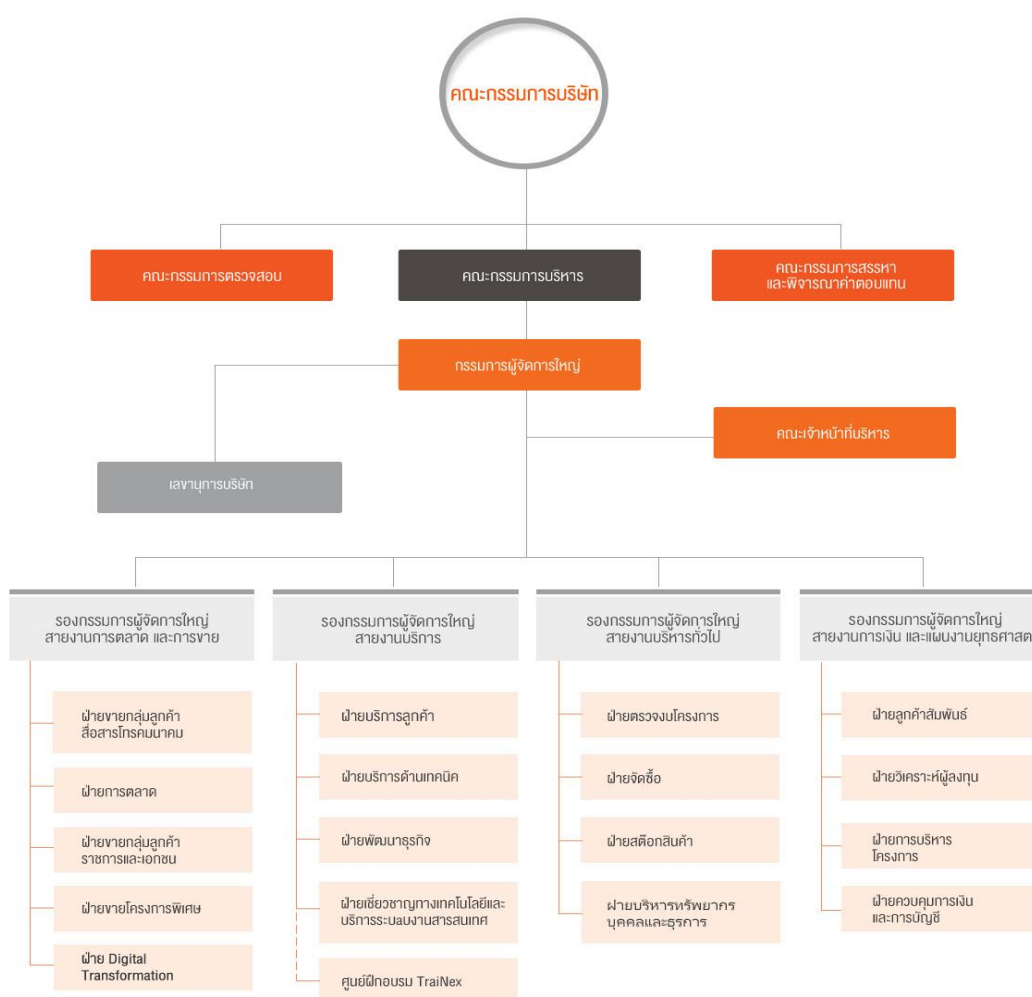
บริษัทประสบความสำเร็จในการสร้างผลงานและรักษาชื่อเสียงในวงการเทคโนโลยีสารสนเทศในประเทศไทยมาช้านานจนเป็นที่ยอมรับทั่วไปในภาคราชการและภาคการสื่อสาร บริษัทได้เข้ามีส่วนร่วมในโครงการต่างๆ ในหลายๆ บทบาทเช่นคู่สัญญาหลัก (Prime Contractor), คู่สัญญาช่วง (Subcontractor), กิจการเข้าร่วม (Consortium) ขึ้นอยู่กับลักษณะของโครงการบริษัทนำเสนอ

1. ผู้ชำนาญการ และการผสมผสานขีดความสามารถในแต่ละด้าน
2. ผลงานที่เป็นที่ยอมรับในระดับนานาชาติ
3. การทำงานแบบกลยุทธ์ในรูปแบบพันธมิตรหรือหุ้นส่วนทางธุรกิจ
4. ความแข็งแกร่งด้านการเงิน

5. ประสบการณ์ในการจัดการงานตั้งแต่ขนาดย่อมจนไปถึงขนาดใหญ่
6. ลดความเสี่ยงด้านเทคนิค ด้านการดำเนินโครงการ และด้านการเงิน ในโครงการที่มีความซับซ้อน

หุ้นส่วนทางธุรกิจของ บริษัท ประกอบไปด้วย ผู้นำทางด้านเทคโนโลยีที่เป็นที่รู้จักในสาขาต่างๆ เช่น Cisco System, Oracle/SUN, IBM, HP, DELL Acer, Alcatel-Lucent, Autodesk, Convergys, SAP, Microsoft, Novell, Synnex และ Symantec เป็นต้น

นอกจากสำนักงานใหญ่ในกรุงเทพมหานครแล้ว บริษัทยังมีศูนย์บริการอยู่ในหัวเมือง 7 จังหวัดเพื่อรองรับการให้บริการที่ทั่วถึงทั้งประเทศ อันได้แก่ ชลบุรี ขอนแก่น เชียงใหม่ สุราษฎร์ธานี พิษณุโลก สงขลา นครราชสีมา และประเทศกัมพูชา



รูปที่ 1.3 โครงสร้างองค์กรของบริษัท

1.2.2 ชื่อและตำแหน่งของพนักงานที่ปรึกษา

ชื่อ: นายจักรพงษ์ ศิริพลตัน

ตำแหน่ง : Assistant Vice President – Installation

1.2.3 ระยะเวลาที่ปฏิบัติงาน

เริ่มการปฏิบัติงานตั้งแต่วันที่ 1 มิถุนายน พ.ศ.2560 ถึงวันที่ 30 พฤศจิกายน พ.ศ. 2560 รวมเป็นระยะเวลา 6 ทั้งสิ้นเดือน

1.2.4 วิธีการดำเนินงาน

- 1) ศึกษากระบวนการและขั้นตอนการทำงานของบริษัทแอ็ดวานซ์ อินโฟเมชั่น เทคโนโลยี จำกัด (มหาชน)
- 2) ศึกษาปัญหาที่เกิดขึ้นในการแก้ไข ดูแล และจัดการระบบของลูกค้าในสถานที่ต่างๆ
- 3) ศึกษาแนวทางการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์เพื่อที่จะสามารถนำมาใช้ในอนาคตและสามารถแก้ไขปัญหาในด้านเครือข่ายของแต่ละสถานที่ได้
- 4) ศึกษาเครื่องมือและอุปกรณ์ต่างๆที่จะนำมาใช้ในแนวคิด การควบคุมระบบเครือข่ายด้วยซอฟต์แวร์
- 5) จำลองการทำงานของแนวคิด การควบคุมระบบเครือข่ายด้วยซอฟต์แวร์
- 6) ควบคุมการส่งข้อมูลในระบบเครือข่ายเน็ตเวิร์ค

1.2.5 ขอบเขตของโครงการ

- 1) ศึกษาแนวทางการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ที่สามารถนำมาใช้กับระบบเน็ตเวิร์คในปัจจุบันได้
- 2) ศึกษาและนำมาจำลองในโปรแกรม GNS3
- 3) ควบคุมการทำงานของระบบเครือข่ายโดยใช้แนวทางการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์

1.2.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้เรียนรู้เกี่ยวกับโครงสร้างและการทำงานในบริษัท บริษัทแอ็ดวานซ์ อินโฟเมชั่น เทคโนโลยี จำกัด (มหาชน)
- 2) ได้เรียนรู้และการปรับตัวในการทำงานจริง
- 3) ได้ความรู้และแนวคิดใหม่จากแนวทางการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์
- 4) ได้เรียนรู้การทำงานและฟังก์ชันของอุปกรณ์ต่างๆหลากหลายผลิตภัณฑ์

บทที่ 2

รายละเอียดของงานที่ปฏิบัติ

2.1 ตำแหน่ง/หน้าที่ของงานที่ได้รับมอบหมาย

ตำแหน่ง : Project Engineer

หน้าที่หลัก : เป็นทีมติดตั้ง โครงการปฏิบัติงานร่วมกับฝ่าย Project Management และลูกค้า

งานที่ทำประจำ

1. ปฏิบัติงานร่วมกับฝ่าย Project Management และลูกค้า ในการติดตั้งโครงการ
2. ออกแบบระบบตามข้อมูล หรือ ตามความต้องการของลูกค้า (Detailed Design)
3. เป็นที่ปรึกษาทางด้านเทคนิคให้แก่ลูกค้า
4. ศึกษา และเข้าอบรมเพื่อหาความรู้เกี่ยวกับผลิตภัณฑ์ หรือเทคโนโลยีต่าง ๆ
5. บันทึกการทำงานในแต่ละวัน
6. จัดทำเอกสารสำหรับการตรวจรับ
7. จัดทำข้อมูลโครงการ
8. ปฏิบัติงานอื่นๆ ตามที่ได้รับมอบหมาย

งานที่มีการมอบหมายเป็นครั้งคราว

1. ปฏิบัติงาน Customer Maintenance ในการแก้ไขปัญหา
2. เป็นวิทยากรฝึกอบรมให้กับลูกค้า
3. ตรวจรับ และส่งมอบงานให้กับลูกค้า

2.2 รายละเอียดของโครงการที่ได้รับผิดชอบ

การศึกษาเกี่ยวกับแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ เพื่อที่จะสามารถตอบโจทย์ลูกค้าบางกลุ่มที่ต้องการระบบเครือข่ายที่เป็นรูปแบบการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ และให้จำลองการทำงานและใช้งานเพื่อที่จะสามารถนำไปใช้งานจริงได้และลดปัญหาความยุ่งยากต่างๆในการจัดการดูแลระบบเครือข่าย

2.3 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้องกับโครงการ

2.3.1 การควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ (Software Defined Network : SDN)

Software Defined Network หรือ SDN นั้นเป็นแนวคิดที่กล่าวไว้ว่าเป็นสถาปัตยกรรมของเครือข่ายเน็ตเวิร์กที่จัดการจัดการเครือข่ายทั้งหมดมารวมอยู่ที่ซอฟต์แวร์ โดยไม่สนใจอุปกรณ์และความแตกต่างของผู้ผลิต ถูกคิดค้นโดย Open Network Foundation (ONF) โดยปกติอุปกรณ์เครือข่ายทั่วไปนั้นจะมี ส่วนควบคุมเส้นทางของข้อมูล (Control Plane) ในการกำหนดว่าข้อมูลในเครือข่ายจะถูกส่งออกไปทางใด และมีส่วนส่งข้อมูล (Data Plane) แต่ทว่าในแนวคิดการควบคุมระบบด้วยซอฟต์แวร์นั้นจะเป็นแนวคิดที่ทำให้ ส่วนควบคุมเส้นทางข้อมูลและส่วนส่งข้อมูลเป็นอิสระจากกัน โดยที่ตัวอุปกรณ์เครือข่ายจะเหลือแค่ส่วนส่งข้อมูลเท่านั้นเท่านั้นและส่วนควบคุมข้อมูลจะกลายเป็นสิ่งที่เรียกว่า ส่วนควบคุมกลาง(Controller) ที่จะคอยเป็นตัวสื่อสารกับอุปกรณ์เครือข่ายตัวอื่นๆ ซึ่งจะมีการใช้ซอฟต์แวร์มาควบคุมและสร้างข้อกำหนดต่างๆ

สิ่งสำคัญของแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์สามารถทำได้คือ

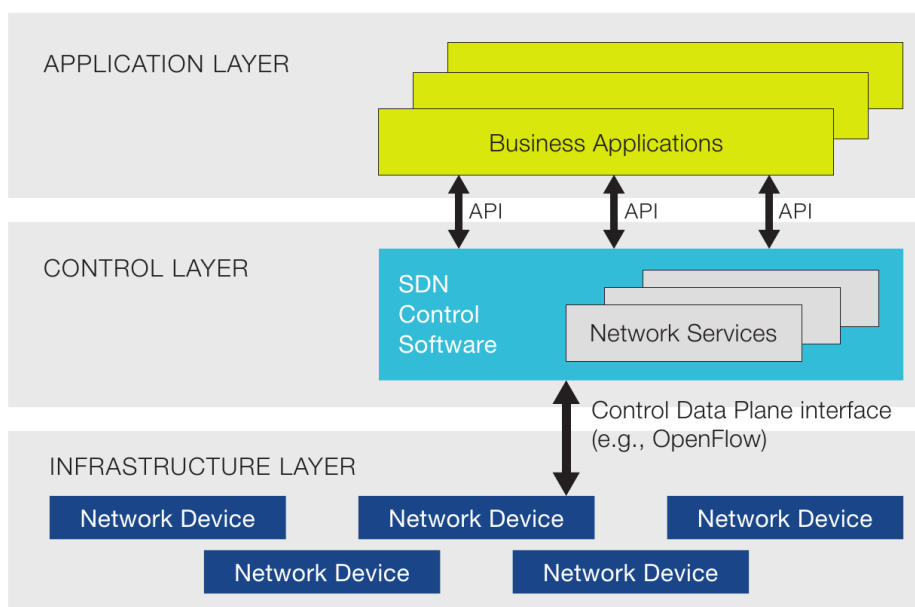
1. แยกส่วนควบคุมเส้นทางของข้อมูลออกจากส่วนส่งข้อมูล
2. การควบคุมจากส่วนกลาง
3. สามารถแยกระบบเครือข่ายออกจากกันได้
4. การเพิ่มเติมความสามารถต่างๆจะไม่ส่งผลกระทบต่อระบบเครือข่าย
5. ลดการแก้ไขข้อมูลในอุปกรณ์เครือข่าย
6. ต้องตอบรับการทำงานแบบอัตโนมัติ

โดยผลลัพธ์จะทำให้เราสามารถควบคุมการจราจรบนเครือข่ายได้ตามที่เราต้องการ จุดที่สำคัญที่สุดของการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์คือจะทำให้การควบคุมระบบเครือข่ายให้มาอยู่ในจุดเดียวกันทั้งหมดเพื่อลดความยุ่งยากเรื่องจำนวนอุปกรณ์และการแก้ไขที่แตกต่างกันออกไปของแต่ละผู้ผลิต

โครงสร้างพื้นฐานของแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์จะมีอยู่ 3 ส่วนหลักด้วยกัน

1. SDN Application : ในส่วนของแอปพลิเคชันนั้นจะเป็นโปรแกรมหรือซอฟต์แวร์ที่ทำการติดต่อสื่อสารกับตัว Controller ผ่านทางส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface : API) โดยที่ในแอปพลิเคชันนั้นสามารถรวบรวมข้อมูลจาก ส่วนควบคุมกลางเพื่อมาใช้ในการตัดสินใจหรือการแก้ไขระบบเครือข่าย วิเคราะห์การส่งของข้อมูล
2. SDN Controller : ในส่วนของตัว SDN Controller นั้นจะเป็นส่วนที่รับความต้องการหรือคำสั่งจากแอปพลิเคชันและส่งผ่านไปยังอุปกรณ์เครือข่ายตัวอื่นๆ และทั้งยังรวบรวมข้อมูลจากอุปกรณ์เครือข่ายและส่งไปยัง SDN Application เพื่อสร้างโครงข่ายจำลองรวมทั้งสถิติและเหตุการณ์ต่างๆที่เกิดขึ้น

3. SDN Networking Device : เป็นอุปกรณ์เครือข่ายที่ควบคุมการส่งข้อมูล เส้นทางต่างๆ ซึ่งรวมทั้งการส่งข้อมูลและเส้นทางด้วย



รูปที่ 2.1 ภาพตัวอย่างโครงสร้างของแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์

โดยที่การติดตั้งแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์นั้นจะมีการออกรูปแบบเป็น 3 รูปแบบด้วยกัน

1. **Overlayed-Based SDN :** รูปแบบนี้เป็นรูปแบบที่มักเห็นกันมากที่สุดในศูนย์ข้อมูล (Data Center) เนื่องจากจะเป็นรูปแบบที่ส่วนควบคุมกลางนั้นจะควบคุมอุปกรณ์เครือข่ายต้นทางและอุปกรณ์เครือข่ายปลายทางของระบบเท่านั้น โดยที่ จะไม่สามารถควบคุมอุปกรณ์ระหว่างทางได้เลย มักใช้กับ Virtual Machine เพื่อให้ต้นทางและปลายทางนั้นอยู่ใน Logical Network เดียวกันทำให้ Software ก็สามารถจัดการรูปแบบต่างๆ เสมือนว่าอยู่ใน Virtual Switch ตัวเดียวกัน
2. **Hybrid-Based SDN :** ตัว Hybrid SDN นั้นมีความคล้ายคลึงกับ Overlayed-Based SDN แต่ที่ว่า Hybrid-Based SDN นั้นสามารถใช้งานร่วมกับระบบเดิมที่ไม่รองรับการทำงานแบบแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ ได้เพียงแต่อุปกรณ์เกทเวย์นั้นต้องรองรับการทำงานในรูปแบบแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์เพื่อที่จะสามารถรองรับการทำงานของส่วนควบคุมกลางได้

3. Device-Based SDN : เป็นรูปแบบที่ส่วนควบคุมกลาง สามารถเข้าไปควบคุมระบบทั้งได้ซึ่งจะทำให้ตอบสนองการทำงานได้หลากหลายรูปแบบแต่ก็มีการแลกเปลี่ยนด้วยค่าใช้จ่ายที่สูงยิ่งขึ้นในการปรับเปลี่ยนอุปกรณ์ทั้งหมดให้เป็นอุปกรณ์ที่รองรับการทำงานอยู่ในรูปแบบการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์

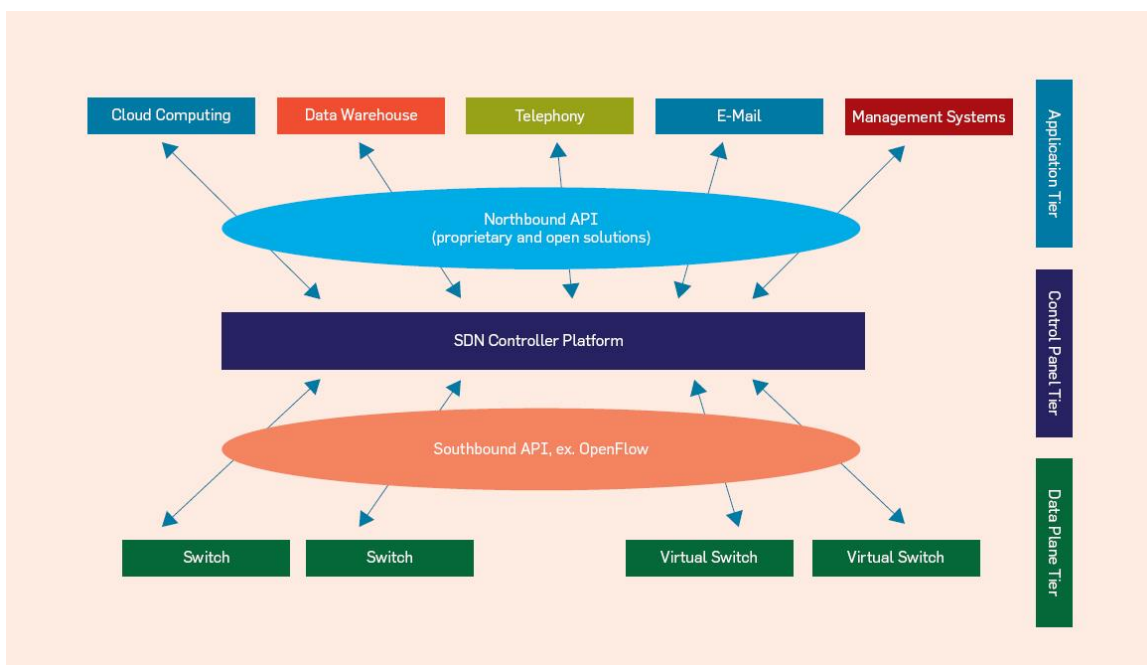
2.3.2 ส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API)

API นั้นเป็นระบบบริการข้อมูลกลางระหว่างผู้ใช้งาน (Client) กับการทำงานของฝั่งของผู้ให้บริการ (Server) หน้าที่หลักคือคอยรับคำสั่ง เมื่อเกิดคำสั่งใดๆหรือการร้องขอใดๆ ส่วนต่อประสานโปรแกรมประยุกต์ก็จะไปรับคำสั่งนั้นๆและนำไปประมวลผลและสรุปเป็นก้อนข้อมูลที่ตรงกับการร้องขอและส่งข้อมูลนั้นกลับไปให้ผู้ใช้งานหรือผู้ให้บริการว่าจะให้ทำอะไร

2.3.3 Northbound and Southbound API

ในการแบ่งการทำงานของแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์นั้นเราจะแบ่งออกเป็น 2 ส่วนหลักๆ

1. Northbound: เป็นส่วนการทำงานประสานระหว่างส่วนควบคุมกลางกับแอปพลิเคชัน ซึ่งเป็นส่วนที่กำหนดข้อกำหนดต่างๆ รวมทั้งปรับเปลี่ยนเกี่ยวกับการเดินทางของข้อมูลในระบบเครือข่ายและทำการแสดงข้อมูลออกมาให้ผู้ใช้งาน แสดงสถิติและจำนวนของข้อมูล โดยจะมีส่วนต่อประสานโปรแกรมประยุกต์ที่มีชื่อว่า RESTCONF เป็นตัวประสาน
2. Southbound: เป็นส่วนการทำงานระหว่างส่วนควบคุมกลางกับอุปกรณ์เครือข่ายต่างๆ โดยฟังก์ชันหลักคือทำการติดต่อสื่อสารและส่งข้อมูลต่างๆไปยังส่วนควบคุมกลางและคอยรับข้อกำหนด (Flow) ที่ถูกสร้างขึ้นจากผู้ใช้งานเพื่อส่งไปยังอุปกรณ์เครือข่ายตัวต่างๆโดยในที่นี้จะใช้ส่วนต่อประสานโปรแกรมประยุกต์ที่มีชื่อว่า Openflow เป็นตัวประสาน



รูปที่ 2.2 ภาพตัวอย่างการทำงานของ Northbound และ Southbound API

2.3.4 RESTCONF API

RESTCONF นั้นเป็นโปรโตคอล REST แบบหนึ่งทำงานอยู่บน Hyper Text Transport Protocol (HTTP) โดยที่จะเป็นการอนุญาตให้ผู้ใช้งานนั้นสามารถเข้าไปยังที่เก็บข้อมูลของตัวควบคุมกลางได้ซึ่งในที่นี้เป็นโปรโตคอลในส่วนของ Northbound API โดยที่จะมีที่เก็บข้อมูล 2 ประเภทคือ

1. Config - จะเป็นข้อมูลที่อยู่ในตัวควบคุมกลาง
2. Operational - จะเป็นข้อมูลที่อยู่ในตัวอุปกรณ์เครือข่าย

โดยที่ RESTCONF นั้นรองรับการทำฟังก์ชันต่างๆ เช่น GET, POST, PUT, DELETE, OPTION

โดยที่รูปแบบการส่งข้อมูลนั้นจะสามารถอยู่ได้ทั้งในรูปแบบ eXtensible Markup Language (XML) และ JavaScript Object Notation (JSON)

2.3.5 OpenFlow Protocol

OpenFlow นั้นเป็นโปรโตคอลที่ใช้ทำงานใน Southbound API เพื่อจะสามารถทำให้ ส่วนควบคุมกลางสามารถติดต่อกับอุปกรณ์เครือข่ายตัวอื่นๆได้ ซึ่ง OpenFlow นั้นจะสามารถทำให้ผู้ใช้งานนั้นสามารถกำหนดข้อกำหนดต่างๆไปยังตัวอุปกรณ์และยังสามารถทำให้รู้ถึงสถิติของระบบเครือข่าย ข้อมูลอุปกรณ์ พอร์ต และการเชื่อมต่อของอุปกรณ์ต่างๆรวมถึงสามารถสร้างเป็นโครงข่ายจำลองของอุปกรณ์เครือข่ายได้ โดยที่การทำงานของ

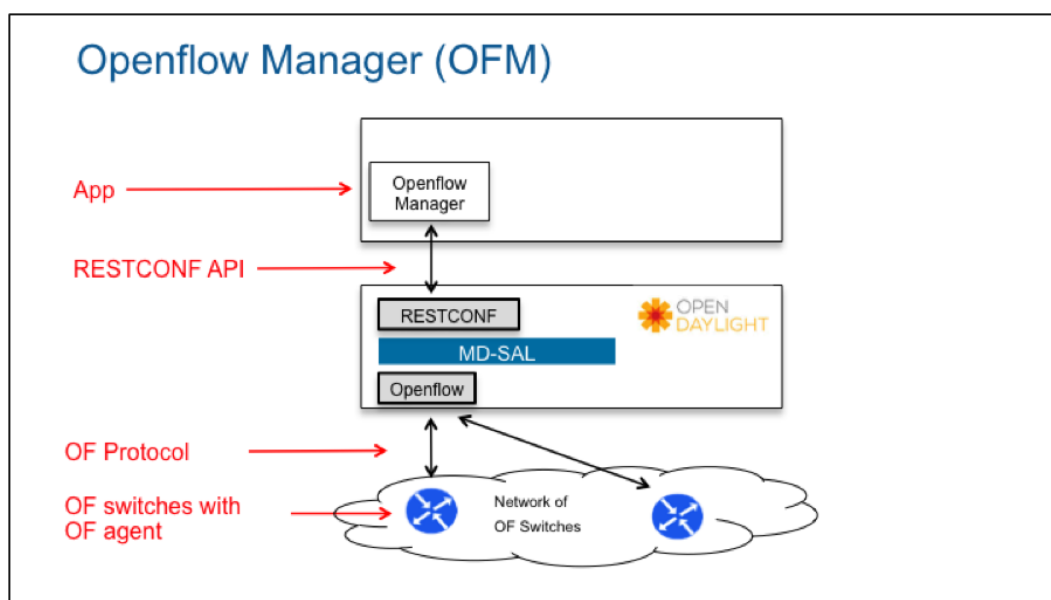
โดยการทำงานในการเลือก Ternary Content Addressable Memory (TCAM) ของ Openflow ที่ใช้สวิตช์นั้นจะมีด้วยกัน 2 รูปแบบก็คือ

1. Reactive Flow - การทำงานของรูปแบบนี้คือ เมื่อมีข้อมูลใดๆก็ตามเข้ามาในตัวอุปกรณ์ และไม่ตรงกับข้อกำหนดใดๆเลย อุปกรณ์จะส่งข้อมูลไปถามหาที่ส่วนควบคุมกลางว่าควรทำเช่นไรหลังจากนั้นส่วนควบคุมกลางก็จะส่งข้อกำหนดไปยังอุปกรณ์นั้นๆ
2. Proactive Flow – จะเป็นการสร้างข้อกำหนดพื้นฐานที่เป็นข้อกำหนดท้ายสุดหากไม่มีข้อกำหนดใดๆตรงกับข้อมูลที่รับมาหากเปรียบเทียบกับอุปกรณ์สมัยก่อนก็เหมือนสวิตช์เมื่อได้รับข้อมูลใดๆมาแล้วไม่มีในอยู่ตารางของตัวเองกับก็จะทำการ Flood and Spray หรือก็คือส่งออกทุกพอร์ตโดยในที่นี้เราสามารถกำหนดการกระทำต่างๆได้เช่น ตั้งให้ทิ้งข้อมูลนั้น หรือ ส่งออกทางพอร์ตใดพอร์ตหนึ่ง

2.3.6 Openflow Manager

เป็นแอปพลิเคชันที่ทำงานบนตัวของส่วนควบคุมกลางที่มีชื่อว่า OpenDaylight เพื่อจำลองโครงสร้างเครือข่าย เส้นทางการทำงานและสถานะต่างๆของ โดยการติดต่อกับ OpenDaylight กับแอปพลิเคชันนั้นจะทำงานผ่านทาง RESTCONF API ซึ่งจะเป็นตัวดึงข้อมูลมาจากอุปกรณ์เครือข่ายทั้งข้อกำหนดและสถิติต่างๆรวมทั้งสร้างข้อกำหนดต่างๆไปยังอุปกรณ์เครือข่าย โดยที่ฟังก์ชันหลักๆที่มีนั้นได้แก่

1. Basic View: เป็นหน้าหลักที่จะแสดงโครงสร้างของระบบเครือข่ายที่เปิดการใช้งาน Openflow อยู่และอุปกรณ์อื่นๆที่เชื่อมต่ออยู่
2. Flow Management: เป็นส่วนที่ไว้ดูข้อกำหนดต่างๆของอุปกรณ์แต่ละตัวและสามารถกำหนด แก้ไข ลบข้อกำหนด ออกจากอุปกรณ์เครือข่าย
3. Statistics: เป็นส่วนที่แสดงค่าสถิติของข้อมูลต่างๆที่เข้ามายังพอร์ตและข้อมูลของข้อกำหนดต่างๆว่ามีข้อมูลไหลเข้าออกมากเพียงใด
4. Hosts: เป็นข้อมูลย่อยๆสำหรับตัวอุปกรณ์ที่เปิดใช้งาน Openflow ว่าผู้ใช้งานที่มาเชื่อมต่อ นั้นมีรายละเอียดอะไรบ้าง



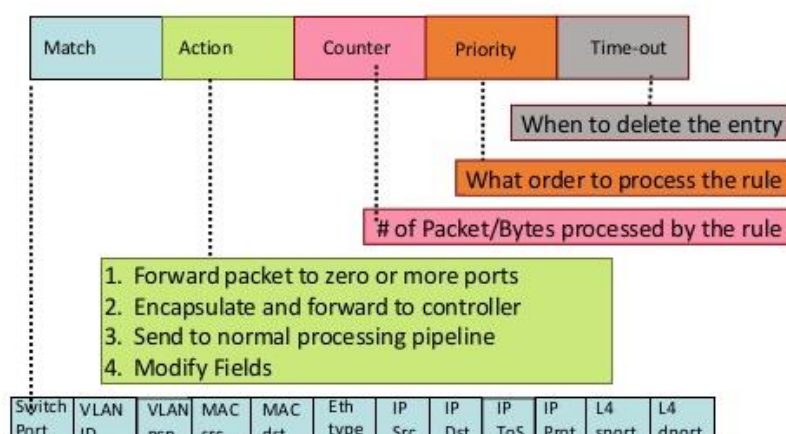
รูปที่ 2.3 ภาพตัวอย่างการทำงานของ Openflow Manager

2.3.7 ตารางข้อกำหนด (Flow Table)

เป็นตารางที่เก็บข้อกำหนดต่างๆเมื่อมีข้อมูลเข้ามาในอุปกรณ์เครือข่ายก็จะนำข้อมูลนั้นไปเปรียบเทียบกับแต่ละส่วนว่ามีในตารางข้อกำหนดหรือไม่ หากมีก็จะทำตามคำสั่งของส่วนที่กำหนดไว้หากไม่มีก็จะไม่ทำการส่งข้อมูลหรือรับข้อมูล โดยการทำงานของตารางข้อกำหนดนั้นจะมีส่วนหลักๆ คือ

1. Match – ทำการตรวจสอบว่ามีค่าใดบ้างของข้อมูลที่ตรงตามค่าที่กำหนดไว้ในตารางข้อกำหนดหากมีก็จะทำปฏิบัติตามข้อกำหนด หากไม่มีก็จะทำการดูที่ตารางข้อกำหนดอื่นๆหากไม่มีก็จะไม่ส่งข้อมูลหรือทำการครอบข้อมูลนั้นทิ้ง
2. Action – หลังจากข้อมูลนั้นมีการตรงกับข้อกำหนดก็ Action ก็จะเป็นสิ่งที่กำหนดว่าจะให้ทำเช่นไร เช่น ส่งต่อออกพอร์ตใด ส่งไปให้ส่วนควบคุมกลางหรือ ทำการส่งไปหาบริการต่างๆ หรือจะเป็นการแก้ไขข้อมูลก็สามารถทำได้
3. Counter – จะเป็นการนับว่าข้อมูลที่เข้ามานั้นมีขนาดเท่าใดหลังจากผ่านการทำงานของข้อกำหนดนั้นๆ
4. Priority – หากมีข้อมูลที่ตรงกับข้อกำหนดมากกว่า 1 ตัวขึ้นไปข้อกำหนดไหนมีค่าลำดับความสำคัญมากกว่าก็จะปฏิบัติตามข้อกำหนดอันนั้นก่อนเสมอ
5. Time-out – ข้อกำหนดต่างๆนั้นสามารถถูกกำหนดให้มีระยะเวลาได้ หากถึงระยะเวลาที่กำหนดก็จะถูกลบออกจากตารางข้อกำหนดไป

OpenFlow: Anatomy of a Flow Table Entry

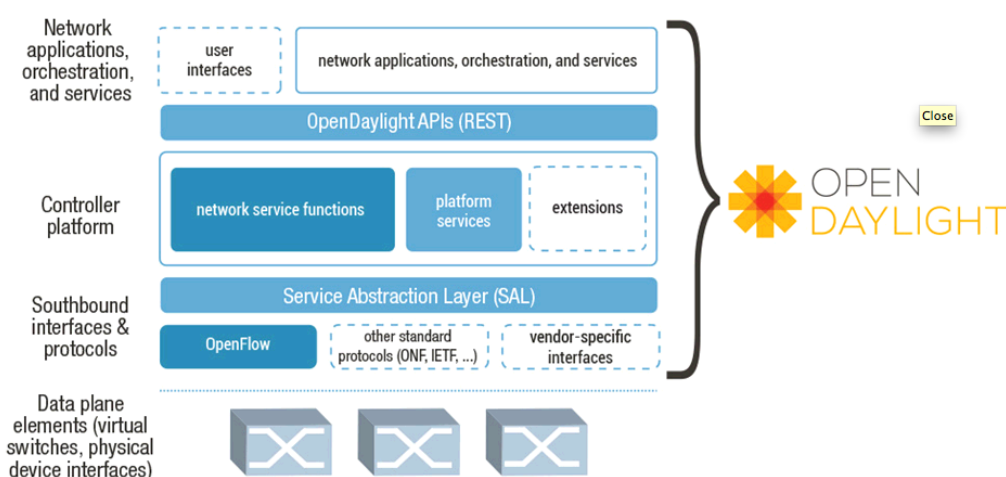


รูปที่ 2.4 ภาพตัวอย่างการทำงานของ Flow Table

2.3.8 OpenDaylight (ODL)

OpenDaylight นั้นเป็นส่วนควบคุมกลางตัวหนึ่งที่สามารถเปลี่ยนแปลงระบบเครือข่ายได้แบบอัตโนมัติ (Networks Automation) ผ่านการใช้ภาษาโปรแกรมมิ่ง (Programming Language) โดยที่ OpenDaylight ถูกสร้างขึ้นมาเพื่อให้รองรับแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ โดยที่จะเฟื่องฟูไปส่วนการควบคุมระบบเครือข่ายด้วยภาษาโปรแกรมมิ่งเพื่อจะลดและแก้ไขปัญหาดังกล่าวในระบบเครือข่ายปัจจุบัน

OpenDaylight



รูปที่ 2.5 ภาพตัวอย่างโครงสร้างของ OpenDaylight

2.3.9 OpenFlow Switch

OpenFlow Switch นั้นเป็นเป็นอุปกรณ์ที่จะทำหน้าที่ส่งข้อมูลต่างๆในสภาพแวดล้อมของแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์โดยจะมีโปรโตคอล Openflow หรือโปรโตคอลอื่นๆในการสื่อสารกับตัวควบคุมกลาง โดยที่จะทำตามตารางข้อกำหนดที่ส่งมาจากส่วนควบคุมกลาง

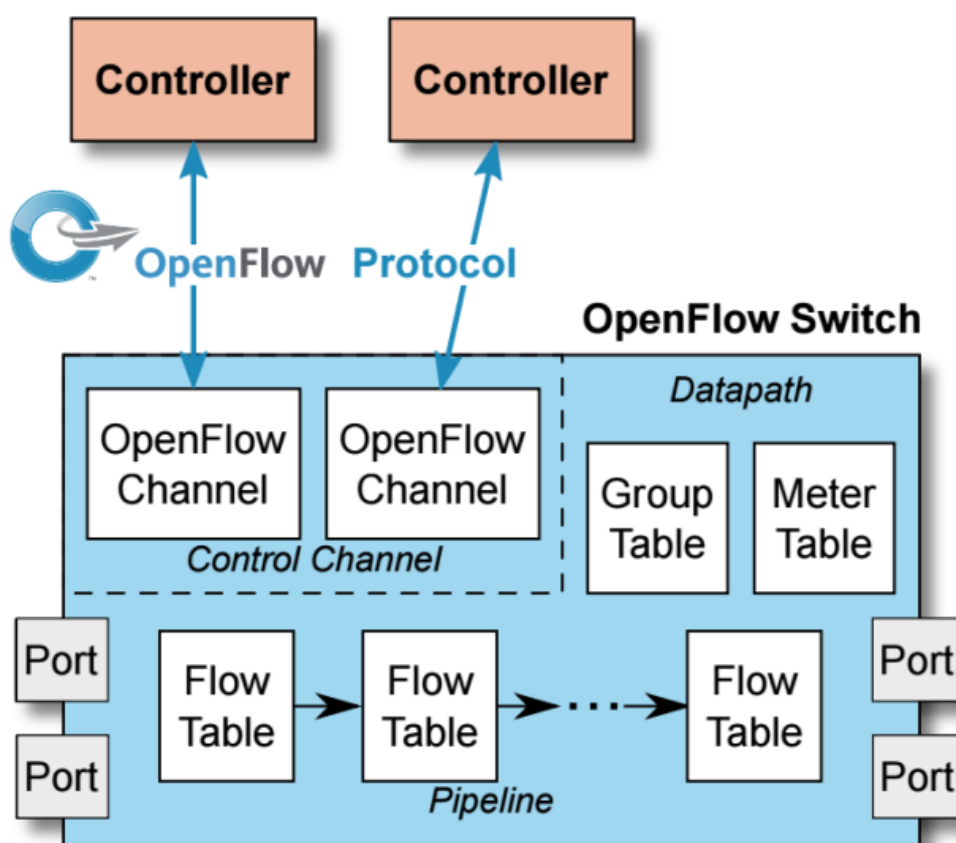


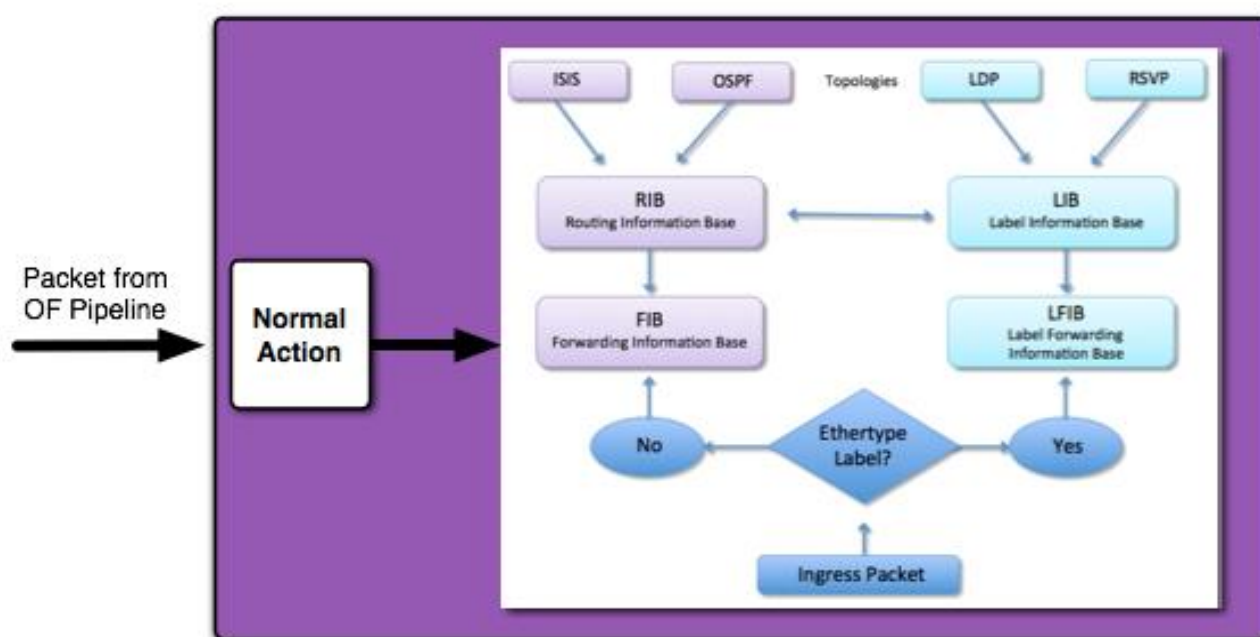
Figure 1: Main components of an OpenFlow switch.

รูปที่ 2.6 ภาพตัวอย่างการทำงานของ OpenFlow Pipeline

ในการขั้นตอนการทำงาน Openflow Switch นั้นข้อมูลทุกข้อมูลจะเข้าตาราง (Table) ที่ 0 ก่อนเสมอซึ่งการที่เราจะสามารถส่งข้อมูลไปยังตารางอื่นๆต้องทำการกำหนดข้อปฏิบัติ (Action)

ให้แก่มัน โดยส่วนใหญ่ นั้น ตารางที่ 0 จะเป็นตารางที่คอยกรองและตรวจสอบข้อมูลต่างๆเพื่อแยกประเภทของข้อมูลและส่งไปยังตารางอื่นๆเพื่อความสะดวกในการดูแลและการจัดการ

แต่ถ้าจะมีการทำงานอีกหนึ่งประเภทซึ่งเป็นการทำงานแบบผสมผสาน (Hybrid Openflow Pipeline) ระหว่างการทำงานแบบแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์กับการค้นหาเส้นทางแบบดั้งเดิม (Routing) โดยที่การทำงานนั้นจะทำการใส่สิ่งที่เรียกว่า NORMAL Action ลงไปในตารางข้อกำหนดซึ่งเป็นข้อกำหนดที่บอกว่าให้ทำงานตามแบบปกติ ซึ่งการทำแบบนี้จะสามารถให้ระบบเครือข่ายแบบดั้งเดิมยังสามารถทำงานไปต่อและค่อยๆปรับเปลี่ยนระบบเครือข่ายทีละส่วนให้เป็นแบบแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ และยังสามารถช่วยเรื่องจำนวนข้อกำหนดต่างๆที่เราต้องใส่ลงไปในการอุปกรณ์



รูปที่ 2.7 ภาพตัวอย่างการทำงานของ Hybrid OpenFlow Pipeline

2.3.10 Graphic Network Simulator 3 (GNS3)

GNS3 เป็นโปรแกรมที่จำลองการทำงานของซอฟต์แวร์ระบบปฏิบัติการในอุปกรณ์เครือข่ายซึ่งมีความสามารถใกล้เคียงกับอุปกรณ์จริงและสามารถนำมาจำลองในระบบเครือข่ายได้ (Network Simulator) โดยที่เป็นโปรแกรมที่มีจุดประสงค์เพื่อให้ผู้ใช้ได้เรียนรู้และศึกษาเกี่ยวกับอุปกรณ์ต่างๆที่ตนสนใจและนำมาจำลองในระบบเพื่อดูว่าสามารถทำงานได้ตามที่เราต้องการหรือไม่

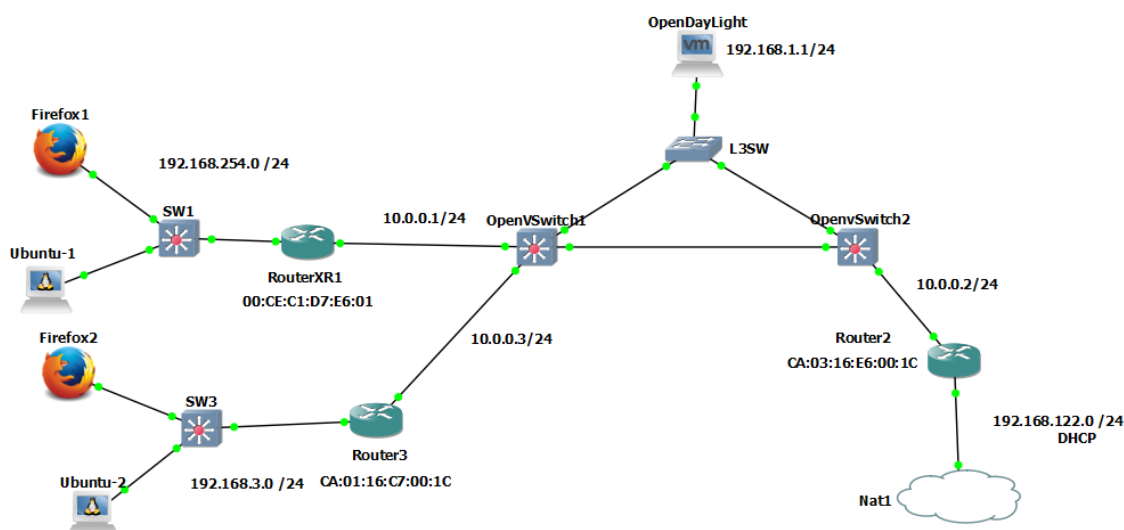
บทที่ 3

การดำเนินงานวิจัย

3.1 ทำการจำลองระบบเครือข่ายลงในโปรแกรมจำลอง

ทำการจำลองระบบเครือข่ายลงในโปรแกรม GNS3 เพื่อที่จะศึกษาการทำงานของแนวคิดการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์ว่าสามารถควบคุมการส่งข้อมูลของอุปกรณ์เครือข่ายโดยส่วนควบคุมกลางจะเป็นตัวกำหนดข้อกำหนดต่างๆผ่านการใช้งานซอฟต์แวร์ของผู้ใช้งานจากนั้นก็ทำการใช้โปรแกรม Wireshark เพื่อตรวจสอบและมุ่งไปการทำ Traffic Engineering และ Non-Best Path Forwarding และการทำ Quality of Service เพื่อจำลองการทำงานที่เป็นปัจจัยหลักที่หลายๆ User ต้องการในปัจจุบัน

1. OpenDaylight เป็นส่วนควบคุมกลาง
2. OpenVirtualSwitch เป็น OpenFlow Switch
3. Ubuntu เป็นผู้ใช้งานในระบบ
4. Firefox เพื่อทดสอบการเข้าใช้บริการอินเทอร์เน็ต
5. NAT เป็นทางออกสู่อินเทอร์เน็ต
6. RouterXR – เป็นอุปกรณ์เครือข่ายในรูปแบบอุปกรณ์จัดหาเส้นทาง (Router) รุ่นใหม่ที่สามารถทำงานได้หลากหลายอย่างมักใช้ในโครงข่ายของผู้ให้บริการทางอินเทอร์เน็ตเนื่องจากมีความสามารถที่ค่อนข้างสูงมากกว่าอุปกรณ์ตัวอื่นๆ โดยจะเป็นตระกูล ASR 9000
7. Router – เป็นอุปกรณ์เครือข่ายในรูปแบบอุปกรณ์จัดหาเส้นทางที่สมัยก่อนมักใช้งานค่อนข้าง



รูปที่ 3.1 ภาพตัวอย่างระบบโครงข่ายใน GNS3
ตารางไอพีแอดเดรสและอินเทอร์เฟซที่เชื่อมต่อ

Device	Interface	Connect To	IP Address	Subnet Mask
RouterXR1	GigabitEthernet0/0/0/0	OpenvSwitch1 – Ethernet 2	10.0.0.1	255.255.255.0
	GigabitEthernet0/0/0/7	SW1 - GigabitEthernet0/0	192.168.254.254	255.255.255.0
Router2	GigabitEthernet1/0	OpenvSwitch2 – Ethernet 2	10.0.0.2	255.255.255.0
	GigabitEthernet2/0	NAT	DHCP	DHCP
Router3	GigabitEthernet1/0	OpenvSwitch1 – Ethernet 3	10.0.0.3	255.255.255.0
	GigabitEthernet2/0	SW3 - GigabitEthernet0/0	192.168.3.254	255.255.255.0
OpenDay light	Ethernet 0/0	L3Switch – GigabitEthernet0/0	192.168.1.1	255.255.255.0
Open vSwitch 1	Ethernet 0/0	L3Switch – GigabitEthernet0/1	192.168.1.2	255.255.255.0
Open vSwitch 2	Ethernet 0/0	L3Switch – GigabitEthernet0/2	192.168.1.3	255.255.255.0
Ubuntu-1	Ethernet 0/0	SW1 - GigabitEthernet0/1	192.168.254.1	255.255.255.0
Ubuntu-2	Ethernet 0/0	SW3 - GigabitEthernet0/1	DHCP	DHCP
Firefox1	Ethernet 0/0	SW1 - GigabitEthernet0/2	192.168.254.2	255.255.255.0
Firefox2	Ethernet 0/0	SW3 - GigabitEthernet0/2	DHCP	DHCP

ตารางที่ 3.1 ตารางไอพีแอดเดรสและอินเทอร์เฟซ

3.2 การตั้งค่าเบื้องต้นให้แก่อุปกรณ์แต่ละตัว

ในการที่จะทำให้อุปกรณ์เครือข่ายนั้นสามารถทำงานได้ตามที่เราต้องการนั้นการใส่ข้อมูลต่างๆก็ยังเป็นเรื่องสำคัญหากขาดตรงจุดนี้ไประบบเครือข่ายก็ไม่สามารถทำงานได้ตามที่เราต้องการ

RouterXR1

```
!! IOS XR Configuration 6.0.1
!! Last configuration change at Thu Nov 9 16:07:16 2017 by admin
!
domain name-server 8.8.8.8
interface MgmtEth0/0/CPU0/0
shutdown
!
interface GigabitEthernet0/0/0/0
ipv4 address 10.0.0.1 255.255.255.0
!
interface GigabitEthernet0/0/0/1
shutdown
!
interface GigabitEthernet0/0/0/2
shutdown
!
interface GigabitEthernet0/0/0/3
shutdown
!
interface GigabitEthernet0/0/0/4
shutdown
!
interface GigabitEthernet0/0/0/5
shutdown
!
```

```

interface GigabitEthernet0/0/0/6
shutdown
!
interface GigabitEthernet0/0/0/7
ipv4 address 192.168.254.254 255.255.255.0
!
router static
address-family ipv4 unicast
0.0.0.0/0 GigabitEthernet0/0/0/0
!
!
router ospf 10
area 0
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/7
!
!
!
end

```

ตารางที่ 3.2 ตารางข้อมูล Configuration RouterXR1

Router2

```

Current configuration : 1314 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname Router2
!

```



```
boot-start-marker
boot-end-marker

!
!
!

no aaa new-model
no ip icmp rate-limit unreachable

!
!
!
!
!
!

ip name-server 8.8.8.8
ip cef
no ipv6 cef

!
!

multilink bundle-name authenticated

!
!
!
!
!
!
!
!
!
!
!
```

```
ip tcp synwait-time 5

!
!
!
!
!
!
!
!
!

interface FastEthernet0/0

no ip address

shutdown

duplex full

!

interface GigabitEthernet1/0

ip address 10.0.0.2 255.255.255.0

ip nat inside

negotiation auto

!

interface GigabitEthernet2/0

ip address dhcp

ip nat outside

negotiation auto

!

interface GigabitEthernet3/0

no ip address

shutdown

negotiation auto

!

interface GigabitEthernet4/0
```

```
no ip address
shutdown
negotiation auto
!
router ospf 10
network 10.0.0.0 0.0.0.255 area 0
network 192.168.122.0 0.0.0.255 area 0
default-information originate
!
ip nat inside source list 1 interface GigabitEthernet2/0 overload
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 192.168.122.1
!
access-list 1 permit any
!
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
stopbits 1
line aux 0
exec-timeout 0 0
```

```

privilege level 15
logging synchronous
stopbits 1
line vty 0 4
login
!
!
end

```

ตารางที่ 3.3 ตารางข้อมูล Configuration Router2

Router3

```

Current configuration : 1291 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname Router3
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
!
ip dhcp excluded-address 192.168.3.254
!
ip dhcp pool pool3

```

```
network 192.168.3.0 255.255.255.0
```

```
dns-server 8.8.8.8
```

```
default-router 192.168.3.254
```

```
domain-name mydomain.com
```

```
!
```

```
!
```

```
!
```

```
ip name-server 8.8.8.8
```

```
ip cef
```

```
no ipv6 cef
```

```
!
```

```
!
```

```
multilink bundle-name authenticated
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
ip tcp synwait-time 5
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!  
!  
!  
!  
interface FastEthernet0/0  
  no ip address  
  shutdown  
  duplex full  
!  
interface GigabitEthernet1/0  
  ip address 10.0.0.3 255.255.255.0  
  negotiation auto  
!  
interface GigabitEthernet2/0  
  ip address 192.168.3.254 255.255.255.0  
  negotiation auto  
!  
interface GigabitEthernet3/0  
  no ip address  
  shutdown  
  negotiation auto  
!  
interface GigabitEthernet4/0  
  no ip address  
  negotiation auto  
!  
router ospf 10  
  network 10.0.0.0 0.0.0.255 area 0  
  network 192.168.3.0 0.0.0.255 area 0  
!  
ip forward-protocol nd
```

```

!
!
no ip http server
no ip http secure-server
!
!
!
!
control-plane
!
!
line con 0
  exec-timeout 0 0
  privilege level 15
  logging synchronous
  stopbits 1
line aux 0
  exec-timeout 0 0
  privilege level 15
  logging synchronous
  stopbits 1
line vty 0 4
  login
!
!
end

```

ตารางที่ 3.4 ตารางข้อมูล Configuration Router3

OpenvSwitch1

```

ovs-vsctl set-controller br0 tcp:192.168.1.1:6633
ovs-vsctl set bridge br0 protocols=OpenFlow13

```

```

ovs-vsctl set Bridge br0 stp_enable=true
ovs-vsctl set-fail-mode br0 secure
ifconfig eth0 192.168.1.2 netmask 255.255.255.0

```

ตารางที่ 3.5 ตารางข้อมูล Configuration OpenvSwitch1

OpenvSwitch2

```

ovs-vsctl set-controller br0 tcp:192.168.1.1:6633
ovs-vsctl set bridge br0 protocols=OpenFlow13
ovs-vsctl set Bridge br0 stp_enable=true
ovs-vsctl set-fail-mode br0 secure
ifconfig eth0 192.168.1.3 netmask 255.255.255.0

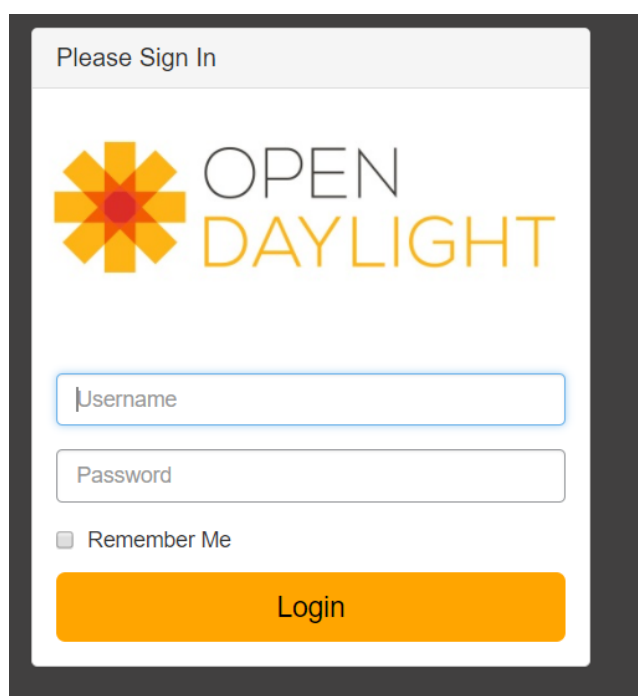
```

ตารางที่ 3.6 ตารางข้อมูล Configuration OpenvSwitch2

3.3 ทำการสร้าง Flow เพื่อควบคุมการส่งข้อมูลในตัวอุปกรณ์

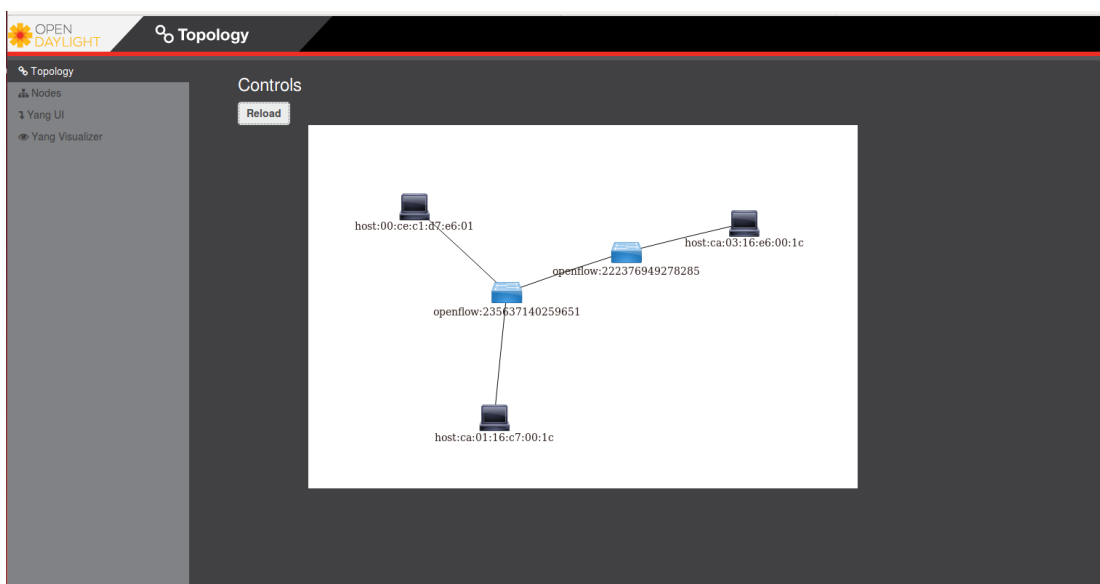
3.3.1 ทำการเปิดตัวควบคุมกลาง Opendaylight

ในการที่เราจะจำลองการทำงานแบบ Software Defined Network นั้นสิ่งที่สำคัญคือการที่ต้องมีตัวอุปกรณ์ส่วนควบคุมกลาง (Controller) โดยหลังจากการเปิดการทำงานแล้ว Opendaylight ก็จะมี User Interface ส่วนหนึ่งที่ยกถึงข้อมูลของอุปกรณ์ต่างๆในระบบที่ Controller สามารถติดต่อด้วยกันได้



รูปที่ 3.2 ภาพตัวอย่างการเข้าใช้งาน OpenDaylight

โดยจะสามารถบอกรายละเอียดของ Topology ในระบบเครือข่ายตอนนี้อยู่ได้และพร้อม MAC Address และ IP ที่มาจากอุปกรณ์นั้นๆ และในส่วนทางด้านสถิติจะสามารถแสดงถึงจำนวนข้อมูลที่เข้ามาในพอร์ตต่างๆ ของใน Open Switch ทำให้เราสามารถดูรายละเอียดคร่าวๆ ได้ว่ามีช่องทางใดใช้ข้อมูลเยอะซึ่งเราอาจจะสลับปรับเปลี่ยนให้ไปใช้อีกเส้นทางเพื่อลดความหนาแน่นของข้อมูล



รูปที่ 3.3 ภาพตัวอย่าง Topology ใน OpenDaylight

Restore Session x OpenDaylight Dlux x +

192.168.1.1:8181/index.html#/node/openflow:222376949278285/port-stat

OPEN DAYLIGHT Nodes

Topology Nodes Yang UI Yang Visualizer

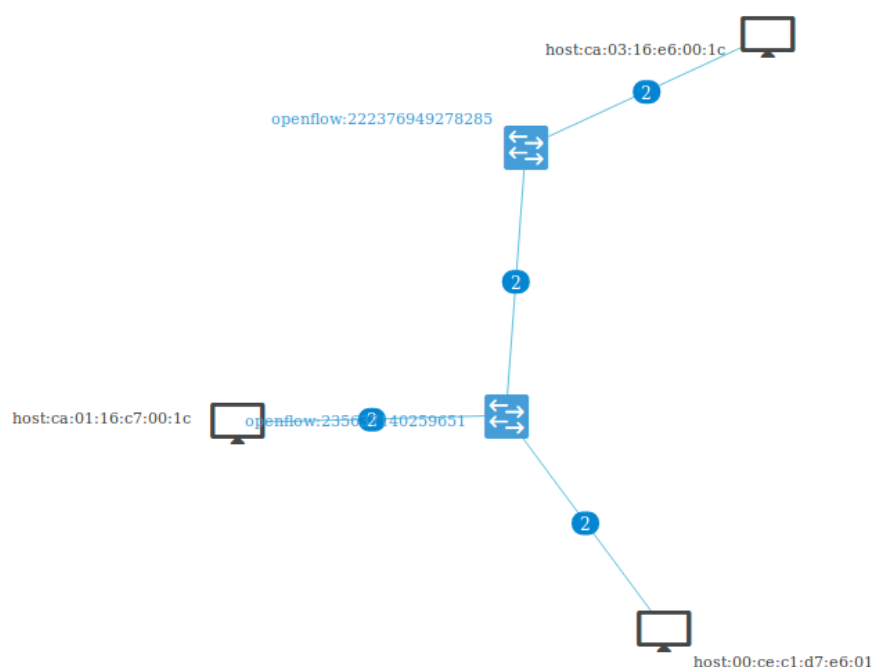
Node Connector Statistics for Node Id - openflow:222376949278285

Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:222376949278285:15	1152	3109	116740	189350	0	11316	0	0	0	0	0	0
openflow:222376949278285:14	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:13	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:12	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:1	4718	11509	450792	3671549	0	0	0	0	0	0	0	0
openflow:222376949278285:11	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:10	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:5	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:4	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:3	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:2	5078	8270	3282644	611892	0	2339	0	0	0	0	0	0
openflow:222376949278285:9	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:LOCAL	89544	5949	4650520	326651	0	121	0	0	0	0	0	0
openflow:222376949278285:8	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:7	0	0	0	0	0	14651	0	0	0	0	0	0
openflow:222376949278285:6	0	0	0	0	0	14651	0	0	0	0	0	0

รูปที่ 3.4 ภาพตัวอย่างสถิติข้อมูลที่เก็บอยู่ใน OpenVirtualSwitch1

3.3.2 เปิดการใช้งาน Application Openflow Manager

ในส่วนของ Application นั้นจะมีไว้เพื่อกำหนด Flow ต่างๆและส่งไปยัง OpenSwitch เพื่อที่เราจะสามารถจำลองการทำงานแบบ SDN ได้ โดยการทำงานหลักๆของ Openflow Manager นั้นสามารถทำงานหลักๆได้ตั้งแต่ Layer 4 Layer 3 และ Layer 2 ซึ่งก็คือ Transport Layer Network Layer และ Data Link Layer โดยที่เราสามารถปรับเปลี่ยน IP Destination หรือ Source MAC address เส้นทางของการส่งข้อมูลและยังสามารถปรับเปลี่ยนการใช้ Port ของทั้ง TCP และ UDP แต่ต้องพึงระวังการแก้ไข Header ต่างๆของข้อมูล หากตั้งค่าไม่ถูกต้องอาจจะทำให้เกิดความเสียหายของข้อมูลและระบบต่างๆได้เช่นข้อมูลอาจจะสูญหายหรือไม่ก็ถูกส่งไปหา Client ตัวอื่นๆ หรือกระทั่งทำให้เกิด Flood ของข้อมูลในรูปแบบต่างๆ จนอาจจะทำให้ระบบมีปัญหาได้



รูปที่ 3.5 ภาพตัวอย่าง Topology ใน Openflow Manager

Host ID	Attachment point ID	Attachment point status	HTS address IP	HTS address MAC	HTS address last seen
host:ca:01: 16:c7:00:1c	openflow:235637140259 651:3	true	10.0.0.3	ca:01:16:c7:00:1c	9 Nov 2017 7:54:32
host:ca:03: 16:e6:00:1c	openflow:222376949278 285:2	true	10.0.0.2	ca:03:16:e6:00:1c	9 Nov 2017 7:54:33
host:00:ce: c1:d7:e6:0 1	openflow:235637140259 651:2	true	10.0.0.1	00:ce:c1:d7:e6:01	9 Nov 2017 7:54:32

10
15
20
25
30

รูปที่ 3.6 ภาพตัวอย่าง Host ใน Openflow Manager

3.3.3 ทำการ Drop ข้อมูลที่มี Source มาจาก 10.0.0.1

เริ่มต้นการทดลองแนวคิดแบบการควบคุมระบบเครือข่ายด้วยซอฟต์แวร์เพื่อตรวจสอบว่า Flow ที่เราสร้างขึ้นมานั้นสามารถทำงานได้จริงหรือไม่ โดยทำการสร้าง Flow ให้ข้อมูลใดก็ตาม ที่มาจาก IP 10.0.0.1 ให้ทำการ Drop ข้อมูลนั้นทิ้งโดยในตอนแรกนั้น RouterXR นั้นที่มี IP 10.0.0.1 ที่ Interface GigabitEthernet0/0/0/0 จะยังสามารถ Ping ไปยัง Internet ได้

```
RP/0/0/CPU0:ios#ping 8.8.8.8
Thu Nov  9 16:11:24.594 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 29/39/49 ms
RP/0/0/CPU0:ios#
```

รูปที่ 3.7 ภาพตัวอย่าง RouterXR Ping ไปยัง IP 8.8.8.8

No.	Time	Source	Destination	Protocol	Length	Info
9	5.155846	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0x10ce, seq=0/0, ttl=255 (reply in 10)
10	5.109439	8.8.8.8	10.0.0.1	ICMP	114	Echo (ping) reply id=0x10ce, seq=0/0, ttl=126 (request in 9)
11	5.202834	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0x10ce, seq=1/256, ttl=255 (reply in 12)
12	5.353052	8.8.8.8	10.0.0.1	ICMP	114	Echo (ping) reply id=0x10ce, seq=1/256, ttl=126 (request in 11)
13	5.356313	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0x10ce, seq=2/512, ttl=255 (reply in 14)
14	5.394879	8.8.8.8	10.0.0.1	ICMP	114	Echo (ping) reply id=0x10ce, seq=2/512, ttl=126 (request in 13)
15	5.402163	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0x10ce, seq=3/768, ttl=255 (reply in 16)
16	5.436263	8.8.8.8	10.0.0.1	ICMP	114	Echo (ping) reply id=0x10ce, seq=3/768, ttl=126 (request in 15)
17	5.439632	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0x10ce, seq=4/1024, ttl=255 (reply in 18)
18	5.478450	8.8.8.8	10.0.0.1	ICMP	114	Echo (ping) reply id=0x10ce, seq=4/1024, ttl=126 (request in 17)

รูปที่ 3.8 ภาพตัวอย่างการดักจับข้อมูลโดยใช้ Wireshark

จากนั้นเราจะทำการสร้าง Flow ขึ้นมาโดยไปสร้างในตัว Application นั่นก็คือ Openflow Manager โดยจะทำการสร้างข้อกำหนดดังต่อไปนี้

The screenshot displays the Openflow Manager configuration interface. On the left, a sidebar lists 'General properties' and 'Match' criteria. Under 'General properties', 'Flow name', 'Table', and 'ID' are marked as 'ADDED'. Under 'Match', 'Priority' is also marked as 'ADDED'. The main panel shows the configuration details for the selected flow. It includes a 'Device' dropdown set to 'openflow:235637140259651 [None]'. The 'General properties' section contains input fields for 'Table' (0-255), 'ID' (Flow ID), 'Priority' (2), and 'Flow name' (Flow name). At the bottom, the 'Actions' section features four buttons: 'Show preview', 'Send request', 'Send all', and 'Back'.

รูปที่ 3.9 ภาพตัวอย่างการสร้าง Flow ใน Openflow Manager

โดยเราจะกำหนดให้ชื่อของข้อกำหนดนี้ว่า Drop_IP_XR โดยที่เราจะตั้งค่าให้ Idle Timeout และ Hard Timeout มีค่าเท่ากับ 0 เพื่อให้ข้อกำหนดดังกล่าวอยู่ในตารางข้อกำหนดตลอดเวลาและตั้งค่า Ethernet Type ให้เป็น 2048 โดยที่ค่านี้จะเป็นการเจาะจงประเภทของ Ethernet ว่าเป็น Ipv4 และตั้ง Ipv4 Source เป็น 10.0.0.1/32 เพื่อเจาะจงแค่ IP นี้เท่านั้น และในส่วนของ Action ก็กำหนดให้เป็น Drop เพื่อที่เมื่อ OpenSwitch ตัวนี้ได้รับ IP ที่มาจาก Source 10.0.0.1 จะทำการ Drop ข้อมูลนั้นทิ้งทันที

รูปที่ 3.10 ภาพตัวอย่าง Flow หลังจากสร้าง Flow เสร็จสิ้น

จากนั้นให้ไปทำการตรวจสอบที่ตัว OpenvSwitch1 ว่า Flow นั้นได้ถูกติดตั้งมายังในตัวอุปกรณ์แล้วหรือไม่

```
/ # ovs-ofctl -O Openflow13 dump-flows br0 | grep 0x0
cookie=0x0, duration=281.461s, table=0, n_packets=31, n_bytes=3018, priority=1001,ip,nw_src=10.0.0.1 actions=drop
cookie=0x0, duration=628.278s, table=0, n_packets=730, n_bytes=62469, priority=1000 actions=CONTROLLER:65535,
NORMAL
/ #
```

รูปที่ 3.11 ภาพตัวอย่าง Flow Drop ข้อมูลใน OpenSwitch

หลังจากนั้นให้ RouterXR ทำการ Ping อีกครั้งหนึ่งจะเห็นได้ว่าการ Ping นั้นจะไม่สำเร็จเนื่องจากหลังมีข้อกำหนดที่บอกตัว OpenSwitch ไว้ว่าหากมีข้อมูลใดๆที่มาจาก Source IP 10.0.0.1 ให้ทำการ Drop ข้อมูลนั้นทิ้งทันที

```
RP/0/0/CPU0:ios#ping 8.8.8.8
Thu Nov  9 16:13:41.325 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
RP/0/0/CPU0:ios#
```

รูปที่ 3.12 ภาพตัวอย่าง RouterXR Ping ไปยัง IP 8.8.8.8 หลังจาก เพิ่ม Flow ลงไป



No.	Time	Source	Destination	Protocol	Length	Info
209	12302.335391	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0xf0ce, seq=0/0, ttl=255 (no response found!)
211	12304.006894	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0xf0ce, seq=1/256, ttl=255 (no response found!)
213	12306.027645	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0xf0ce, seq=2/512, ttl=255 (no response found!)
217	12308.047010	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0xf0ce, seq=3/768, ttl=255 (no response found!)
221	12310.067932	10.0.0.1	8.8.8.8	ICMP	114	Echo (ping) request id=0xf0ce, seq=4/1024, ttl=255 (no response found!)

รูปที่ 3.13 ภาพตัวอย่างการดักจับข้อมูลโดยใช้ Wireshark หลังจากเพิ่ม Flow ลงไป

จะเห็นได้ว่า RouterXR นั้นจะไม่สามารถติดต่อกับอุปกรณ์ตัวไหนได้ทั้งสิ้นนอกจากใน Local ของตัวเองเพราะ Flow ที่เรากำหนดลงไปใน OpenSwitch นั้นได้ทำการ Drop ข้อมูลอื่นๆแต่หาอุปกรณ์ตัวอื่นๆก็สามารถติดต่อกันได้ตามปกติ

3.3.4 ทำการเปลี่ยนเส้นทางของข้อมูล

ในบางครั้ง User ก็มีความต้องการให้ข้อมูลนั้นไปในเส้นทางอื่นๆที่เขาต้องการซึ่งเส้นทางนั้นอาจจะไม่ใช่ Best Path เราก็สามารถใช้การสร้าง Flow ในการควบคุมเส้นทางของข้อมูลเพื่อให้ตรงกับความต้องการของลูกค้า

ในการเปลี่ยนเส้นทางของข้อมูลนั้นเราตั้งข้อกำหนดว่าหากมี IP ใดๆก็ตามที่อยู่ใน Subnet 192.168.3.0 /24 และมี IP Destination = 8.8.8.8 จะให้การส่งข้อมูลนั้นออกไปยังอีกเส้นทางหนึ่งก่อนอื่นก็สร้าง Flow ขึ้นมาใน Openflow Manager

รูปที่ 3.14 ภาพตัวอย่างการสร้าง Flow ในการเปลี่ยนเส้นทางของข้อมูล

จากนั้นทำการตรวจสอบว่า Flow นั้นได้ถูกติดตั้งลงใน OpenvSwitch1 แล้วหรือไม่

```
/ # ovs-ofctl -O Openflow13 dump-flows br0 | grep 192.168
cookie=0x0, duration=420.267s, table=0, n_packets=0, n_bytes=0, priority=1001,ip,nw_src=192.168.3.0/24,nw_dst=8.8.8.8 actions=output:2
/ #
```

รูปที่ 3.15 ภาพตัวอย่าง Flow ในการเปลี่ยนเส้นทางของข้อมูล OpenvSwitch1

จากนั้นให้ Ubuntu-2 ทำการ Ping ออกสู่ Internet โดยตอนแรกจะให้ทำการ Ping ไปยัง 203.151.13.166 โดย IP ดังกล่าวจะเป็น IP ของ www.pantip.com เพื่อตรวจสอบว่าสามารถ Ping ออกสู่ Internet ได้หรือไม่

```

root@Ubuntu-2:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr f2:96:f9:e1:42:60
          inet addr:192.168.3.1  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::f096:f9ff:fe01:4260/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:155 errors:0 dropped:2 overruns:0 frame:0
          TX packets:324 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15822 (15.8 KB)  TX bytes:97098 (97.0 KB)

root@Ubuntu-2:~# ping 203.151.13.166
PING 203.151.13.166 (203.151.13.166) 56(84) bytes of data.
64 bytes from 203.151.13.166: icmp_seq=1 ttl=125 time=139 ms
64 bytes from 203.151.13.166: icmp_seq=2 ttl=125 time=33.7 ms
64 bytes from 203.151.13.166: icmp_seq=3 ttl=125 time=25.2 ms
64 bytes from 203.151.13.166: icmp_seq=4 ttl=125 time=86.2 ms
64 bytes from 203.151.13.166: icmp_seq=5 ttl=125 time=175 ms
^C
--- 203.151.13.166 ping statistics ---
6 packets transmitted, 5 received, 16% packet loss, time 5007ms
rtt min/avg/max/mdev = 25.247/92.107/175.763/58.571 ms
root@Ubuntu-2:~#

```

รูปที่ 3.16 ภาพตัวอย่างการ Ping จาก Ubuntu2 ไปยัง 203.151.13.166

จากนั้นให้ทำการลอง Ping ไปยัง IP 8.8.8.8 โดยจะสังเกตว่าการ Ping นั้นจะไม่สำเร็จเนื่องจากถูกเปลี่ยนเส้นทางไปยังอีกเส้นทางหนึ่งโดย หลังจากใช้ Wireshark ในการตรวจจับข้อมูลแล้วจะเห็นได้ว่าข้อมูลนั้นถูกส่งออกไปยังพอร์ต 2 ของ OpenSwitch ซึ่งไม่ใช่เส้นทางที่ใช้ในการออกสู่ Internet

```

root@Ubuntu-2:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr f2:96:f9:e1:42:60
          inet addr:192.168.3.1  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::f096:f9ff:fe01:4260/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:89 errors:0 dropped:2 overruns:0 frame:0
          TX packets:309 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10740 (10.7 KB)  TX bytes:95846 (95.8 KB)

root@Ubuntu-2:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6120ms

```

รูปที่ 3.17 ภาพตัวอย่างการ Ping จาก Ubuntu-2 ไปยัง 8.8.8.8

*Standard input [RouterXR1 Gi0/0/0 to OpenVSwitch1 eth2]

No.	Time	Source	Destination	Protocol	Length	Info
10	11.140999	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=1/256, ttl=63 (no response found!)
12	12.137013	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=2/512, ttl=63 (no response found!)
14	13.142718	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=3/768, ttl=63 (no response found!)
17	14.142043	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=4/1024, ttl=63 (no response found!)
19	15.152086	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=5/1280, ttl=63 (no response found!)
21	16.162000	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=6/1536, ttl=63 (no response found!)
22	17.212560	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=7/1792, ttl=63 (no response found!)
24	18.260071	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=8/2048, ttl=63 (no response found!)
25	19.252756	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=9/2304, ttl=63 (no response found!)
27	20.271407	192.168.3.1	8.8.8.8	ICMP	98	Echo (ping) request id=0x00b4, seq=10/2560, ttl=63 (no response found!)

รูปที่ 3.18 ภาพตัวอย่างการ Wireshark ของ Interface RouterXR1 Gig0/0/0/0 กับ OpenvSwitch1 Eth2

3.3.5 ทำการติดตั้ง Quality of Service

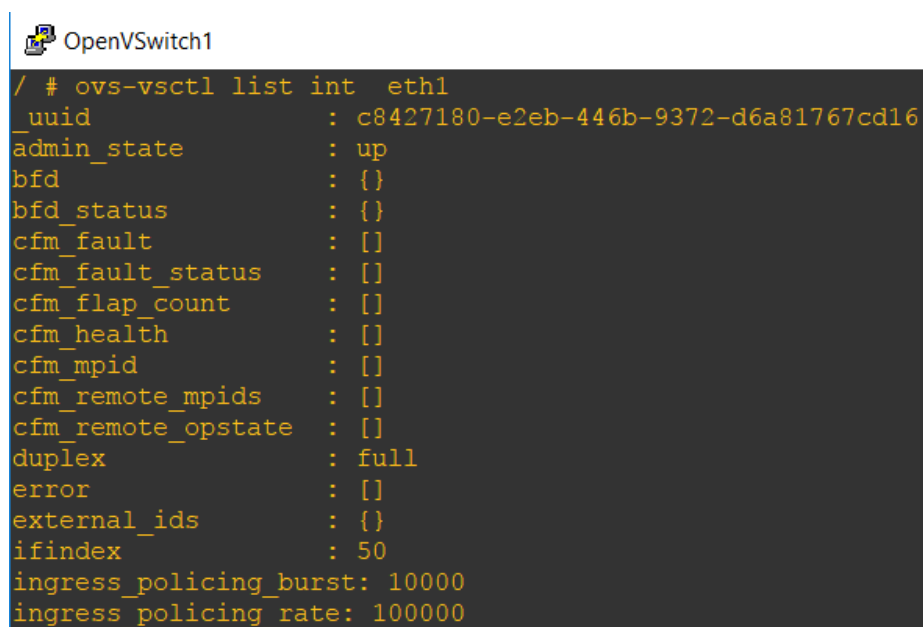
ในการทำ QoS นั้นเราสามารถทำได้ 2 รูปแบบ คือ ต่อ 1 Port กับ ต่อ 1 Flow โดยในลักษณะ ต่อ 1 Port มักใช้กับแบบ Ingress เพื่อจำกัดปริมาณข้อมูลที่จะเข้ามาในตัวอุปกรณ์โดยใช้คำสั่ง

```
ovs-vsctl set interface eth1 ingress_policing_rate=100000
ovs-vsctl set interface eth1 ingress_policing_burst=10000
```

ตารางที่ 3.7 ตารางคำสั่ง Ingress Policy ของ Eth1 ใน OpenvSwitch1

โดย Ingress_Policing_Rate จะเป็นตัวกำหนดความเร็วที่ Interface นี้สามารถส่งออกมาได้ และ Ingress_policing_burst จะเป็นตัวกำหนดความเร็วที่สามารถเกินมาจาก Policing_rate ได้

โดยเมื่อทำการแสดงใน OpenSwitch ก็จะเห็นค่าที่เรากำหนดเอาไว้ซึ่งในลักษณะแบบนี้จะไม่สามารถใช้ Application ควบคุมได้เนื่องจากการตั้งค่าแค่พอร์ตนั้นพอร์ตเดียว



```
OpenVSwitch1
/ # ovs-vsctl list int eth1
_uuid          : c8427180-e2eb-446b-9372-d6a81767cd16
admin_state    : up
bfd            : {}
bfd_status     : {}
cfm_fault      : []
cfm_fault_status : []
cfm_flap_count : []
cfm_health     : []
cfm_mpid       : []
cfm_remote_mpid : []
cfm_remote_opstate : []
duplex         : full
error          : []
external_ids   : {}
ifindex        : 50
ingress_policing_burst: 10000
ingress_policing_rate: 100000
```

รูปที่ 3.19 ภาพตัวอย่างการ Ingress Policy ใน OpenVSwitch 1

ในลักษณะต่อมาจะเป็นการใช้ในลักษณะ Egress โดยจะเป็นการเลือกพอร์ตที่เป็นทางออก จากนั้นจะมีการกำหนดค่าความเร็ว (Bandwidth) ไว้ ก่อนอื่นเราจะทำการทดสอบความเร็วระหว่าง Ubuntu1 และ Ubuntu2 โดยให้ทาง Ubuntu1 เป็น Server เพื่อตรวจสอบความเร็วของ Client ที่ส่งข้อมูลเข้ามาและให้ Ubuntu2 เป็น Client ที่ส่งข้อมูลเข้าหา Server

```
[ 5] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35070
[ 5] 0.0-11.6 sec 2.00 MBytes 1.44 Mbits/sec
[ 4] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35080
[ 4] 0.0-12.7 sec 2.12 MBytes 1.40 Mbits/sec
[ 5] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35090
[ 5] 0.0-11.8 sec 2.12 MBytes 1.52 Mbits/sec
[ 4] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35106
[ 4] 0.0-11.9 sec 2.00 MBytes 1.41 Mbits/sec
[ 5] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35112
[ 5] 0.0-12.3 sec 2.12 MBytes 1.45 Mbits/sec
[ 4] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35118
[ 4] 0.0-12.2 sec 2.12 MBytes 1.46 Mbits/sec
[ 5] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35234
[ 5] 0.0-14.0 sec 1.88 MBytes 1.12 Mbits/sec
[ 4] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35240
[ 4] 0.0-13.6 sec 1.88 MBytes 1.15 Mbits/sec
```

รูปที่ 3.20 ภาพตัวอย่างรายละเอียดการส่งข้อมูลของ Ubuntu2

โดยจะสังเกตได้ว่าความเร็วในการส่งนั้นอยู่ที่ประมาณ 1.4 Mb/S จากนั้นเราจะทำการสร้าง QoS ไว้ในตัวอุปกรณ์โดยใช้คำสั่ง

```
ovs-vsctl set port eth2 QoS=@newQoS -- --id=@newQoS create QoS type=linux-htb
queues=0=@q0 -- --id=@q0 create queue other-config:min-rate=500000 other-config:max-
rate=500000
```

ตารางที่ 3.8 ตารางคำสั่ง Egress Policy ของ Eth1 ใน OpenvSwitch1

ซึ่งใน QoS ที่เราได้สร้างไว้นั้นจะกำหนดให้พอร์ตที่ 2 ของ OpenvSwitch1 ตั้งค่าไว้ว่าหากมีข้อมูลใดๆก็ตั้งค่า Queue ให้เป็น 0 และออกมาทางพอร์ตนี้จะถูกจำกัดความเร็วให้เหลือ 500 Kb/s ซึ่งนั้นเมื่อแสดงข้อมูลใน OpenvSwitch2 ก็จะเห็นค่าต่างๆดังรูปโดยจะมีชื่อที่เป็นตัวบอกว่าพอร์ตนี้กำลังใช้ QoS ตัวใดอยู่และใน QoS ก็จะมีรายละเอียด Queue ที่อยู่ด้านในอยู่

```
OpenVSwitch1
/ #
/ # ovs-vsctl list port eth2
{
  uuid              : f2d6d629-67b7-43e1-a3aa-384625a3a26c
  bond_active_slave : []
  bond_downdelay    : 0
  bond_fake_iface   : false
  bond_mode         : []
  bond_updelay      : 0
  external_ids      : {}
  fake_bridge       : false
  interfaces        : [e3649c16-111a-49b7-a529-c87114afd3b4]
  lacp              : []
  mac               : []
  name              : "eth2"
  other_config      : {}
  qos               : 238998fd-d84e-496b-b391-325c7dbc4d0d
  rstp_statistics   : {}
  rstp_status       : {}
  statistics        : {stp_error_count=0, stp_rx_count=0, stp_tx_count=1591}
  status            : {stp_port_id="800f", stp_role=designated, stp_sec_in_state="3162", stp_state=forwarding}
  tag               : []
  trunks            : []
  vlan_mode         : []
}
```

รูปที่ 3.21 ภาพตัวอย่างรายละเอียด Eth2 ของ OpenvSwitch1

```

/ # ovs-vsctl list qos
_uuid          : ea32aa92-f0a1-4fff9-b034-eab8ba46fa92
external_ids   : {}
other_config   : {}
queues         : {0=ffee393a-0e24-4967-a9fe-224009c2497e}
type           : linux-htb
/ #

```

รูปที่ 3.22 ภาพตัวอย่าง QoS Policy ใน OpenVswitch1

โดยในรูปจะมี Queue ที่เป็นตัวเลข Bandwidth ที่อยู่ในอุปกรณ์ซึ่งในรูปจะถูกตั้งค่าให้มีความเร็วเท่ากับ 500 Kb/s

```

/ # ovs-vsctl list queue
_uuid          : ffee393a-0e24-4967-a9fe-224009c2497e
dscp           : []
external_ids   : {}
other_config   : {max-rate="500000", min-rate="500000"}
/ #

```

รูปที่ 3.23 ภาพตัวอย่าง QoS Policy ใน OpenVswitch1

ซึ่งในการสร้าง Queue นั้นเราสามารถสร้างได้สูงสุด 8 Queue ใน 1 QoS Policy ซึ่งสิ่งนี้จะทำให้เราสามารถแยกประเภทของลูกค้าและ Traffic ต่างๆ จากนั้นนำ QoS มาใช้เพื่อตอบสนองความต้องการของลูกค้า จากนั้นทำการสร้าง Flow ขึ้นมาและทำการ Set_queue Flow นั้นเพื่อให้สามารถใช้งาน QoS นั้นและ Bandwidth ที่เราได้ตั้งค่าเอาไว้

The screenshot shows the OpenvSwitch Flow configuration interface. It is divided into two main sections: 'General properties' and 'Actions'.

General properties:

- Table: 0
- ID: 50
- Priority: 5000
- Flow name: QosOVS1
- Hard timeout: 0
- Idle timeout: 0
- Cookie: 0
- In port: openflow:235637140259651::

Actions:

- Enqueue:** Queue: Qos, Queue Id: 0
- Normal:** Maximum length: 65535

At the bottom, there are four buttons: 'Show preview', 'Send request', 'Send all', and 'Back'.

รูปที่ 3.24 ภาพตัวอย่าง Flow QoS

```
/ # ovs-ofctl -O Openflow13 dump-flows br0 | grep set_queue
cookie=0x0, duration=622.966s, table=0, n_packets=6113, n_bytes=9017841, priority=5000,in_port=3 actions=set_queue:0,NORMAL
```

รูปที่ 3.25 ภาพตัวอย่าง Flow QoS ใน OpenvSwitch1

จากนั้นในตามข้อกำหนดที่เราได้ตั้งค่าไว้ข้อมูลใดๆก็ตามที่เข้ามาทางพอร์ต Eth2 และออกไปยัง Eth1 จะถูกจำกัด Bandwidth ดังในภาพ Queue ที่ 0 จะถูกตั้งค่า Bandwidth ให้มีค่า 800Mb เป็นค่าสูงสุดและ 200Mb เป็นค่าต่ำสุดซึ่งจะเห็นได้ว่า Flow ที่กำหนดนั้นมีข้อมูลวิ่งเข้ามาอย่างถูกต้องและทำให้ Ubuntu2 ส่งข้อมูลคู่อีกครั้ง

```
[ 5] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35304
[ 5] 0.0-31.4 sec 1.50 MBytes 400 Kbits/sec
[ 4] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35330
[ 4] 0.0-31.8 sec 1.75 MBytes 461 Kbits/sec
[ 5] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35344
[ 5] 0.0-26.4 sec 1.50 MBytes 477 Kbits/sec
[ 4] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35352
[ 4] 0.0-26.6 sec 1.50 MBytes 473 Kbits/sec
[ 5] local 192.168.254.1 port 5001 connected with 192.168.3.2 port 35362
[ 5] 0.0-27.4 sec 1.50 MBytes 460 Kbits/sec
```

รูปที่ 3.26 ภาพตัวอย่างรายละเอียดการส่งข้อมูลของ Ubuntu2 หลังจากการทำ QoS

ซึ่งจะเห็นได้ความเร็วของทางผู้ส่งถูกลดลงมาเหลือประมาณ 460 Kb/s เนื่องจากการที่เราตั้งค่าความเร็วให้ถูกลดลงเหลือแค่เพียง 500 Kb/s และอาจจะเกิดการสูญหายความเร็วบางส่วนเนื่องจาก Delay จากสภาพแวดล้อมต่างๆ

บทที่ 4

สรุปผลการปฏิบัติงาน

จากการที่ได้ปฏิบัติงานสหกิจศึกษา ณ บริษัท แอ็ดวานซ์ อินฟอร์เมชั่น เทคโนโลยี จำกัด (มหาชน) สามารถสรุปผลการไปปฏิบัติงานที่ได้รับมอบหมายและได้ประโยชน์ดังนี้

4.1 สรุปผลการปฏิบัติงาน

จากการที่ข้าพเจ้าได้ปฏิบัติงานสหกิจศึกษาในตำแหน่ง Project Engineer ณ บริษัทแอ็ดวานซ์ อินฟอร์เมชั่น เทคโนโลยี จำกัด (มหาชน)

หัวข้อ	ผลงาน	จำนวน
1.จัดทำเอกสารและแบบฟอร์มต่างๆ	คู่มือการอัปเดตอุปกรณ์	3 เล่ม
	คู่มือการลง License อุปกรณ์	1 เล่ม
2. อัปเดตอุปกรณ์เครือข่ายต่างๆและแก้ไข Configuration ต่างๆ	ASR 9000 Series	13 ตัว
	Router 2900 Series	125 ตัว
	ASR 900 Series	30 ตัว
	ISR 4000 Series	15 ตัว
3.ปฏิบัติที่หน้าไซต์งาน	Data Center Site	9 ครั้ง
	Service Provider Site	35 ครั้ง
4.ช่วยการจัดการทำ Lab และบรรยายให้แก่ลูกค้า	KTCB	3 ครั้ง
5.Troubleshooting อุปกรณ์ต่างๆ	Network Device	7 ครั้ง
6.เข้าร่วมการบรรยายและการประชุม	เข้าร่วมการประชุม	2 ครั้ง
	บรรยายเพื่อเพิ่มความเข้าใจแก่ตัวพนักงาน	6 ครั้ง
7.Training Course	Pre-CCNA	1 ครั้ง
	Manner & Presentation	1 ครั้ง

ตารางที่ 4.1 ตารางสรุปผลการปฏิบัติงาน

4.2 ประโยชน์ที่ได้รับ

4.2.1 ประโยชน์ต่อตนเอง

1. เข้าใจการทำงานของแต่ละแผนกในองค์กร และขั้นตอนการทำงานของแต่ละฝ่ายตั้งแต่เริ่มจนจบการทำงาน
2. เรียนรู้และฝึกทักษะต่างๆในการทำงานจริง
3. ทราบถึงปัญหาต่างๆที่เกิดขึ้นในระบบเครือข่ายและปัญหาของซอฟต์แวร์ต่างๆ
4. ได้นำความรู้ที่มีไปใช้ในงานดำเนินงานจริงพร้อมทั้งแก้ไขปัญหาต่างๆ
5. ได้ฝึกฝนการเขียน Config ในอุปกรณ์รูปแบบอื่นๆ
6. ได้รู้จักการปรับตัวเพื่อแก้ไขสถานการณ์ที่เกิดขึ้น

4.2.2 ประโยชน์ต่อสถานประกอบการ

1. ช่วยลดการจ้างพนักงานประจำลงได้ เนื่องจากมีนักศึกษาช่วยปฏิบัติงาน
2. สามารถทำงาน ตอบสนองกับลูกค้าได้รวดเร็วมากยิ่งขึ้น
3. ช่วยแบ่งเบาภาระของพนักงานประจำ
4. สามารถช่วยคัดกรองนักศึกษาที่จะเข้ามาเป็นพนักงานประจำในอนาคต

4.2.3 ประโยชน์ต่อมหาวิทยาลัย

1. เกิดความร่วมมือทางวิชาการระหว่างสถานประกอบการและมหาวิทยาลัย
2. มหาวิทยาลัยได้รับข้อมูลย้อนกลับเพื่อนำไปปรับปรุงและพัฒนาหลักสูตร ในการเรียนการสอน
3. เป็นการแสดงถึงศักยภาพของมหาวิทยาลัยช่วยให้มหาวิทยาลัยได้รับการยอมรับจากสถานประกอบการและตลาดแรงงานมากยิ่งขึ้น
4. มหาวิทยาลัยสามารถผลิตนักศึกษาที่มีความคิดสร้างสรรค์ สามารถบูรณาการวิทยาการต่าง ๆ ที่เกี่ยวข้องกับเทคโนโลยีสารสนเทศได้อย่างมีประสิทธิภาพ

บทที่ 5

ปัญหาและข้อเสนอแนะ

จากการที่ข้าพเจ้าปฏิบัติงานสหกิจในตำแหน่ง Project Engineer ณ บริษัทแอ็ดวานซ์ อินฟอร์เมชั่น เทคโนโลยี จำกัด (มหาชน) ตั้งแต่วันที่ 1 มิถุนายน พ.ศ. 2560 ถึงวันที่ 30 พฤศจิกายน พ.ศ. 2560 เป็นระยะเวลา 6 เดือน นั้นได้รับความรู้และประสบการณ์ต่างๆ ที่จะเป็นประโยชน์ต่อในอนาคตและในระหว่างปฏิบัติงาน เป็นประโยชน์ต่อในอนาคตและในระหว่างปฏิบัติงานนั้นได้พบกับปัญหา อุปสรรคต่างๆ ซึ่งสามารถนำมาสรุปและเสนอแนะได้ดังนี้

1. โครงการไม่เกี่ยวข้องโดยตรงกับงานที่ทำอยู่ประจำแค่เพียงสามารถนำมาใช้หรือปรับปรุงระบบต่างๆ ได้ในอนาคต
2. ไม่มีแผนการดำเนินงานสำหรับนักศึกษาสหกิจที่แน่นอน

โดยปัญหาดังกล่าว ทางบริษัทที่เข้าร่วม โครงการสหกิจศึกษา กับสถาบันจะต้องร่วมมือพร้อมทั้งประสานงาน และติดตามผลงานจนเสร็จสิ้นโครงการ

บรรณานุกรม

- [1] บริษัท แอ็ดวานซ์ อินฟอร์เมชั่น เทคโนโลยี จำกัด (มหาชน) [Online] Available:
<http://www.ait.co.th>
- [2] SDxCentral “Understanding SDN Architecture” [Online] Available:
<https://www.sdxcentral.com/sdn/definitions/inside-sdn-architecture/>
- [3] Wiki Opendaylight “Opendaylight Controller:MDSAL:RESTCONF “ [Online] Available:
https://wiki.opendaylight.org/view/OpenDaylight_Controller:MDSAL:Restconf#Restconf_URI
- [4] Devnet Cisco “Openflow Manager” [Online] Available:
<https://developer.cisco.com/site/devnetcreations/openflow-mgr/>
- [5] Margaret Rouse “OpenFlow switch” [Online] Available :
<http://searchsdn.techtarget.com/definition/OpenFlow-switch>
- [6] Opendaylight [Online] Available:
<https://www.opendaylight.org>
- [7] Jasmie Russello “Southbound vs Northbound SDN: What are the differences?” [Online]
Available: <http://blog.webwerks.in/data-centers-blog/southbound-vs-northbound-sdn-what-are-the-differences>
- [8] สารานุกรม วิกีพีเดีย “Software-defined networking” [Online] Available:
https://en.wikipedia.org/wiki/Software-defined_networking
- [9] Matt Osawalt “Openflow Deep-Dive” [Online] Available:
<https://keepingitclassless.net/2014/07/sdn-protocols-2-openflow-deep-dive/>
- [10] Brent Salisbury “Hybrid Openflow Using The Normal Action” [Online] Available
<http://networkstatic.net/hybrid-openflow-using-the-normal-action/#!prettyPhoto>
- [11] Wiki Opendaylight “ODL_Hybrid_Mode” [Online] Available :
https://wiki.opendaylight.org/images/1/1d/ODL_Hybrid_Mode.pdf
- [12] Daehee Kim, “Understand QoS at OpenSwitch” [Online] Available:
<http://dannykim.me/danny/57771>

ภาคผนวก

บันทึกรายงานการปฏิบัติงาน

วัน/เดือน/ปี	รายการ	หมายเหตุ
1 มิ.ย 2560	ปฐมนิเทศให้นักศึกษาและรายละเอียดเกี่ยวกับทางบริษัท AIT	
2 มิ.ย 2560	ทำความรู้จักกับบุคลากรภายในบริษัทและทำความเข้าใจ โครงสร้างและวัฒนธรรมขององค์กร	
5 มิ.ย 2560	ทำการติดตั้ง VM Wareworkstation เพื่อนำไปจำลอง อุปกรณ์เครือข่ายต่างๆเพื่อใช้ในการศึกษา	
6 มิ.ย 2560	ทำการเชื่อมต่ออุปกรณ์เครือข่ายที่ Warehouse ของทางบริษัท AIT พร้อมอัปเดต IOS ให้เป็นเวอร์ชันที่ตรงตาม Requirement ของลูกค้าพร้อมเรียนรู้การตรวจสอบความ ถูกต้องต่างๆ	
7 มิ.ย 2560	เรียนรู้ระบบ Network ในเครื่องของลูกค้าและเรื่อง MPLS BGP และ ISIS พร้อมเข้าใจการทำงานของแต่ละส่วนใน เครื่องของลูกค้า	
8 มิ.ย 2560	เข้ารับฟังการบรรยายโครงการของลูกค้าในเครือ CAT โดย มีรุ่นพี่ในทีมเป็นผู้บรรยาย	
9 มิ.ย 2560	ทำการเชื่อมต่ออุปกรณ์เครือข่ายที่ Warehouse ของทางบริษัท AIT พร้อมอัปเดต IOS ให้เป็นเวอร์ชันที่ตรงตาม	
12 มิ.ย 2560	เข้ารับ Training โครงการ Pre-CCNA ที่ทางบริษัทเป็นคน จัดขึ้นให้นักศึกษา	
13 มิ.ย 2560	ศึกษาเกี่ยวกับแนวคิด Software Defined Network	
14 มิ.ย 2560	ทำการเชื่อมต่ออุปกรณ์เครือข่ายที่ Warehouse ของทางบริษัท AIT พร้อมทั้งตรวจสอบจำนวนของที่ได้รับว่าครบถูกต้อง สามารถใช้งานได้ครบทุกชิ้นงานหรือไม่	
15 มิ.ย 2560	ทำการประชุมเรื่องหัวข้องานของโครงการ AIT Internship ที่ทางบริษัทเป็นคนจัดตั้งขึ้น	
16 มิ.ย 2560	เข้าฝึกที่ Trainex จากหัวหน้าทีมของทีมติดตั้งแต่ละทีมเพื่อ เพิ่มความเข้าใจในการทำงานให้มากขึ้น	

19 มิ.ย 2560	ทำการศึกษาเกี่ยวกับ Site บาง Site เพื่อเพิ่มความเข้าใจเกี่ยวกับ MPLS และ BGP และเข้าใจถึง Configuration Template ว่าทำงานเช่นไร	
20 มิ.ย 2560	ศึกษาเกี่ยวกับแนวคิด Software Defined Network	
21 มิ.ย 2560	ตรวจสอบอุปกรณ์ในบริษัทเพื่อเช็คสถานะและการทำงานว่ายังสามารถทำได้ครบถ้วน	
22 มิ.ย 2560	ศึกษาเกี่ยวกับ ACS และการทำงานของ TACACS+	
23 มิ.ย 2560	ศึกษาข้อมูลเกี่ยวกับ SDN อุปกรณ์ที่รองรับและการทำงานของตัวมันเอง	
26 มิ.ย 2560	ศึกษาการทำงานของ ASR 9000 Series และทำความเข้าใจการทำงานและการ Configuration	
27 มิ.ย 2560	ติดตั้ง ASR 9001 เพื่อทดสอบการทำงานและ Configuration ค่าต่างๆ	
28 มิ.ย 2560	ทำการศึกษาเกี่ยวกับ Opendaylight	
29 มิ.ย 2560	ทำการศึกษาเกี่ยวกับ Opendaylight	
30 มิ.ย 2560	ไปติดตั้งอุปกรณ์ต่างๆที่ ศูนย์อบรม Trainex เพื่อเตรียมการอบรม CCNP R&S ในวันถัดไป	
3 ก.ค 2560	ทำการศึกษาและติดตั้ง Opendaylight บน VM	
4 ก.ค 2560	ทำการศึกษาและติดตั้ง Opendaylight บน VM	
5 ก.ค 2560	ศึกษาเกี่ยวกับการใช้ UI บน Opendaylight	
6 ก.ค 2560	ทำความเข้าใจเรื่อง API และ ความสำคัญ	
7 ก.ค 2560	ทำการเช็คอุปกรณ์เครือข่ายที่ Warehouse ของทางบริษัท AIT พร้อมอัปเดต IOS ให้เป็นเวอร์ชันที่ตรงตาม Requirement ของลูกค้าพร้อมเรียนรู้การตรวจสอบความถูกต้องต่างๆ	
10 ก.ค 2560	หยุดชดเชยวันอาสาฬหบูชา(วันเสาร์ที่ 8 ก.ค 2560)	
11 ก.ค 2560	หยุดชดเชยวันอาสาฬหบูชา(วันอาทิตย์ที่ 9 ก.ค 2560)	
12 ก.ค 2560	ทำการ Simulate และลองส่งค่า API เพื่อดึงข้อมูล	
13 ก.ค 2560	ทำ Manual Guide การติดตั้ง IOS ของอุปกรณ์ Router	

14 ก.ค 2560	คุยเรื่องหัวข้อ โครงการ AIT Internship พร้อมแบ่งหน้าที่การทำงาน	
17 ก.ค 2560	เดินทางไป CAT นนทบุรีและทำการตรวจสอบ CSR 16 Slot ทำการเปลี่ยน Card ในตัวอุปกรณ์ให้เป็นอุปกรณ์ใหม่	
18 ก.ค 2560	เข้ารับฟังบรรยายที่ Cisco Thailand และรับชมสถานที่และอุปกรณ์ต่างๆใน Cisco	
19 ก.ค 2560	ทบทวนเนื้อหาเพื่อแข่งขัน Netrider	
20 ก.ค 2560	สอบแข่งขัน Net Rider Round 1	
21 ก.ค 2560	ทำการเช็คอุปกรณ์เครือข่ายที่ Warehouse ของทางบริษัท AIT และทำเอกสารเก็บข้อมูลและจำนวนของที่ได้รับ	
24 ก.ค 2560	เข้าร่วมโครงการ AIT Internship พร้อมเสนอหัวข้อ Start up ให้คณะผู้บริหารของทางบริษัท AIT	
25 ก.ค 2560	ลางาน	ไปช่วยกิจกรรม ที่คณะ
26 ก.ค 2560	ทำการศึกษาเกี่ยวกับ OpenVirtualSwitch	
27 ก.ค 2560	ทำการศึกษาเกี่ยวกับ OpenVirtualSwitch	
28 ก.ค 2560	ทำการศึกษาเกี่ยวกับ OpenFlow	
31 ก.ค 2560	ทำการศึกษาเกี่ยวกับ OpenFlow	
1 ส.ค 2560	ทำการตรวจสอบและติดตั้ง Card และ Module ต่างๆของอุปกรณ์ ASR 9006 ของแต่ละภูมิภาค	
2 ส.ค 2560	ทำการตรวจสอบอุปกรณ์ ASR 9001 ว่าทำงานได้ถูกต้องโดยการ Show ข้อมูลจากข้างในออกมาดูและทำการตรวจสอบ	
3 ส.ค 2560	อธิบาย Topology ของ Site ลูกค้านำเบื้องต้นให้กับพี่ที่พี่ย้ายมาจากอีกทีมนี้	
4 ส.ค 2560	เข้าร่วมการประชุมของทีมเพื่อแบ่งงานและหาผู้ช่วยเหลือแต่ละงานพร้อมชี้แจงงานให้ชัดเจน	
7 ส.ค 2560	ทำการเช็คสายส่งข้อมูลและหาต้นทาง-ปลายทางพร้อมทำ Label บอกรับส่งและปลายทางเพื่อความสะดวกในการทำงานในครั้งต่อไป	

8 ส.ค 2560	ทำการทดสอบการใช้งานระหว่าง OpenvSwitch กับ Opendaylight	
9 ส.ค 2560	เข้าร่วมโครงการ AIT Openhouse และได้รับการทดสอบ เพื่อที่จะเข้าเป็นพนักงานในเครือบริษัท	
10 ส.ค 2560	ศึกษาการทำงานของ Site ToT และ ซ่อมแซม VM มี ปัญหาต้องการลงใหม่	
11 ส.ค 2560	ทำความเข้าใจเกี่ยวกับระบบ Network ที่ Site ToT	
14 ส.ค 2560	หยุดชดเชยวันเฉลิมพระชนมพรรษาสมเด็จพระนางเจ้า สิริกิติ์พระบรมราชินีนาถ(วันเสาร์ที่ 12 ส.ค 2560)	
15 ส.ค 2560	ทำการศึกษาเกี่ยวกับ Docker เพื่อช่วยลดภาระการทำงานของ ของ PC	
16 ส.ค 2560	ทำการศึกษาเกี่ยวกับ Docker เพื่อช่วยลดภาระการทำงานของ ของ PC	
17 ส.ค 2560	ทำการศึกษาเกี่ยวกับ Flow Table	
18 ส.ค 2560	ช่วยทำเอกสารให้กับฝ่าย HR และทำแบบทดสอบ ออนไลน์สำหรับผู้ที่ต้องการเข้าทำงานที่บริษัท	
21 ส.ค 2560	หาข้อมูลเกี่ยวกับผู้ให้บริการทางอินเทอร์เน็ต	
22 ส.ค 2560	ประชุมการทำงาน	
23 ส.ค 2560	ตรวจสอบบัคในระบบ	
24 ส.ค 2560	ทำการตรวจสอบอุปกรณ์ ASR 920 ว่าทำงานได้ถูกต้อง โดยการ Show ข้อมูลจากข้างในออกมาดูและทำการ ตรวจสอบ	
25 ส.ค 2560	ทำ Manual Guide การติดตั้ง IOS ของอุปกรณ์ ASR 920	
28 ส.ค 2560	ทำการอัปเดต XFPD และ อัปเดต IOS และติดตั้ง Software Maintenance Update (SMU) – วันที่ 1	
29 ส.ค 2560	ทำการอัปเดต XFPD และ อัปเดต IOS และติดตั้ง Software Maintenance Update (SMU) – วันที่ 2	
30 ส.ค 2560	ศึกษาเกี่ยวกับปัญหาที่จะเกิดขึ้นเมื่อ Controller เสียหาย หรือมีปัญหา	
31 ส.ค 2560	จำลองระบบลง GNS3 อย่างครบถ้วนและแก้ไขปัญหาที่ เจอเบื้องต้น	

1 ก.ย 2560	ตรวจสอบอุปกรณ์ที่มีปัญหาที่ Site งาน CAT และทำการแก้ไขและเปลี่ยนอุปกรณ์พร้อมเฝ้าสังเกตการณ์	
4 ก.ย 2560	ลางาน	อ่านหนังสือเตรียมความพร้อมแข่งขัน Netrider
5 ก.ย 2560	ลางาน	อ่านหนังสือเตรียมความพร้อมแข่งขัน Netrider
6 ก.ย 2560	แข่งขัน Netrider Round 2	
7 ก.ย 2560	เตรียมอุปกรณ์ Fiber และ แคปสาย Lan แบบ Cross เพื่อเตรียมไป Connect กะ อุปกรณ์ตัวใหม่ที่ CAT นนทบุรี	
8 ก.ย 2560	ทำการติดตั้งและเปลี่ยนอุปกรณ์ที่ Site CAT และทำ Label ให้สามารถเข้าใจได้ง่ายมากขึ้น	
11 ก.ย 2560	เข้าร่วมการอบรมที่ KTBCS โดยเป็นผู้สอนด้าน Lab Network เบื้องต้น – วันที่ 1	
12 ก.ย 2560	ทำการศึกษาการทำงานของ API แต่ละอัน	
13 ก.ย 2560	เข้าร่วมการอบรมที่ KTBCS โดยเป็นผู้สอนด้าน Lab Network เบื้องต้น - วันที่ 2	
14 ก.ย 2560	ทำการ โกสเครื่อง PC และอัปเดต Driver ให้กับทางบริษัทชน(บขส.)	
15 ก.ย 2560	เข้าร่วมการอบรมที่ KTBCS โดยเป็นผู้สอนด้าน Lab Network เบื้องต้น – วันที่ 3	
18 ก.ย 2560	ศึกษา Solution ที่ลูกค้าต้องการจะให้แก้พร้อมทำความเข้าใจ Configuration ต่างๆ	
19 ก.ย 2560	เตรียมพร้อมสำหรับการ Migrate CRS 16 slot ในวันที่ 20 ก.ย 2560	
20 ก.ย 2560	Migrate CRS 16 slot ที่ CAT เพิ่ม Link ช่วยในการทำโหลด Balance และ โยก Traffic	

21 ก.ย 2560	ตรวจสอบข้อผิดพลาดจากการโยก Traffic ของวันที่ 20 ก.ย 2560 และทำการแก้ไข Configuration เพิ่มลงไป	
22 ก.ย 2560	คุยกับลูกค้าเรื่องการส่งขอ และประสานงานกับคนส่งของฝั่ง AIT พร้อมติดตามการขนส่งของ	
25 ก.ย 2560	ตรวจสอบโปรเจกต์ว่าสามารถทำงานตามที่คาดหวังไว้ได้	
26 ก.ย 2560	ทำรายงานความคืบหน้า	
27 ก.ย 2560	ทำรายงานความคืบหน้า	
28 ก.ย 2560	ทำรายงานความคืบหน้า	
29 ก.ย 2560	ทำรายงานความคืบหน้า	
2 ต.ค 2560	เข้าไปตรวจสอบสภาพอุปกรณ์ ASR 9001 พร้อมเช็คความเรียบร้อย จำนวนของและการทำงานต่างๆ ที่ Warehouse	
3 ต.ค 2560	ทำรายงานสรุปเล่มสหกิจศึกษา	
4 ต.ค 2560	เข้า Warehouse ลง Package ต่างๆบนตัว ASR 9001 พร้อมทั้งอัปเดตเวอร์ชันต่างๆให้ตรงตามความต้องการของลูกค้า	
5 ต.ค 2560	ทำรายงานสรุปเล่มสหกิจ	
6 ต.ค 2560	ลางาน	ทีม Installtion ทุกทีมได้ไป Outing
9 ต.ค 2560	ทำการทดลอง Lab GNS3	
10 ต.ค 2560	ทำรายงานสรุปเล่ม	
11 ต.ค 2560	ทำรายงานสรุปเล่ม	
12 ต.ค 2560	ทำรายงานสรุปเล่ม	
13 ต.ค 2560	หยุดเนื่องด้วยวันคล้ายวันสวรรคต	
16 ต.ค 2560	ทำการช่วยแก้ปัญหาด้านการ Design Solution ให้แก่ลูกค้า	
17 ต.ค 2560	ทำการ Active License ให้กับ ASR920 ที่ Warehouse พร้อม Troubleshooting ปัญหาจากการ Downgrade รุ่นให้ต่ำกว่าที่ได้รับมา	
18 ต.ค 2560	บรรยายลักษณะงานกับสิ่งที่ฝึกงานที่บริษัทให้แก่อาจารย์นิเทศ	
19 ต.ค 2560	ทำ Manual Guide Active License และ Downgrade Software ASR920 ให้แก่ลูกค้า	

20 ต.ค 2560	ทำรายงานรูปเล่ม	
23 ต.ค 2560	หยุดเนื่องด้วยเป็นวันปิยมหาราช	
24 ต.ค 2560	ทำรายงานรูปเล่ม	
25 ต.ค 2560	เข้า CAT บางรักเพื่อทำการ โยก License	
26 ต.ค 2560	ผลงาน	วาง ดอกไม้จันทร์ที่ สนามหลวง
27 ต.ค 2560	ทำความเข้าใจเรื่อง Overlay Tansport Virtualization	
30 ต.ค 2560	ทำ Manual Guide ISR 4351 การเปลี่ยน License จาก อุปกรณ์หนึ่งไปยังอุปกรณ์หนึ่งและวิธีการติดตั้ง License และ Active	
31 ต.ค 2560	ทำรายงานรูปเล่ม	
1 พ.ย 2560	ทำรายงานรูปเล่ม	
2 พ.ย 2560	ทำรายงานรูปเล่ม	
3 พ.ย 2560	ทำรายงานรูปเล่ม	
6 พ.ย 2560	เข้าร่วมประชุม โครงการ BroadBandGateway	
7 พ.ย 2560	ทำการเช็คอุปกรณ์ Switch และ Router ที่เป็น Spare เพื่อ นำไปไว้ที่ไซส์ลูกค้า	
8 พ.ย 2560	เข้าคอร์สอบรม Personality ของทางบริษัท	
9 พ.ย 2560	เข้าคอร์สอบรม Presentation ของทางบริษัท	
10 พ.ย 2560	แก้ไข Config Router 2900 ที่มีปัญหา	
13 พ.ย 2560	ช่วยเหลือ Site งานฝั่ง System Data Cenetr ในการ Upgrade ทั้ง Office เป็น Window รุ่นล่าสุด	
14 พ.ย 2560	ช่วยเหลือ Site งานฝั่ง System Data Cenetr ในการ Upgrade ทั้ง Office เป็น Window รุ่นล่าสุด	
15 พ.ย 2560	ทำการ Config TACCAS และทำการ Standby แก้ไขให้กับ User ที่เข้าใช้งานไม่ได้	
16 พ.ย 2560	ทำการหา Configuration ของ ISR 4331	
17 พ.ย 2560	ทำการตรวจสอบอุปกรณ์ UCS Switch 3506 ACS ICE ก่อนนำไปติดตั้ง	
20 พ.ย 2560	บรรยายโปรเจกต์สภกิจที่ตนทำให้กับพี่ๆในทีม	

21 พ.ย 2560	ทำการเปลี่ยน Config อุปกรณ์ 7600 ให้เป็นของ ASR9000 เพื่อเตรียมนำไป Migrate	
22 พ.ย 2560	Present ผลงานให้กับทางบริษัท	
23 พ.ย 2560	ทำการเปลี่ยน Config อุปกรณ์ 7600 ให้เป็นของ ASR9000 เพื่อเตรียมนำไป Migrate	
24 พ.ย 2560	ช่วยเหลือ Site งานฝั่ง System Data Cenetr ในการสอนการ ใช้งานระบบและอุปกรณ์ต่างๆ	
27 พ.ย 2560	Standby ที่ Site งานลูกค้าเพื่อตรวจสอบความถูกต้อง	

ภาพกิจกรรมการทำงาน

ภาพปฏิบัติงาน CRS3 – 16 Slot



ภาพปฏิบัติงาน ASR 9001 ที่ไซต์งาน



ภาพเข้าอบรม PreCCNA



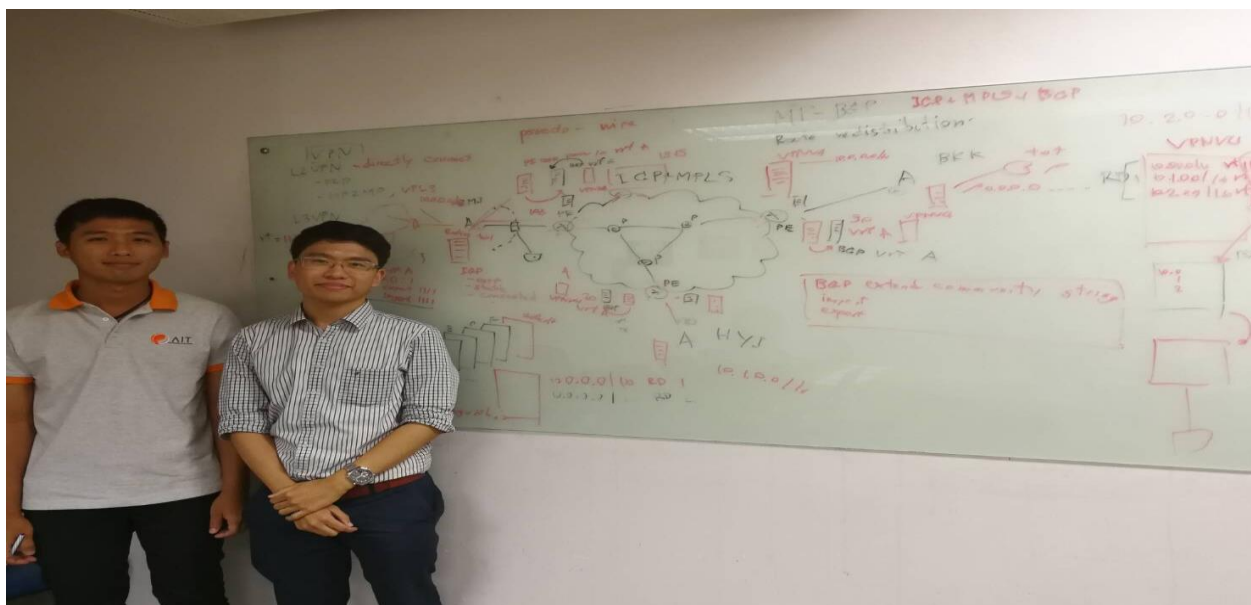
ภาพ Present งานสทกิจให้กับบริษัทและทางผู้บริหาร



ภาพปฏิบัติงานตรวจสอบ Router 2960



ภาพอบรม MPLS BGP ISIS ของในไซต์งาน



ภาพคอร์สอบรม Presentation & Personality Class



ภาพโครง AIT Excellent Award



ประวัติผู้เขียน

ชื่อ – นามสกุล	จรัสสิน ปัญญาวิสุทธิชัย
วัน เดือน ปี เกิด	22 มิถุนายน 2539
ที่อยู่	515/137 ถนน เจริญราษฎร์ แขวงบางโคล่ เขตบางคอแหลม จังหวัด กรุงเทพมหานคร 10120
ประวัติการศึกษา	กำลังศึกษาในหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยี สารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง