สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้งาน Software-Defined Networking บนเครือข่ายผู้ให้บริการอินเทอร์เน็ต

THE APPLICATION OF USING SOFTWARE-DEFINED NETWORKING ON INTERNET SERVICE PROVIDER NETWORK

โดย

มนัสชัย ใจเอื้อ

MANASCHAI JAI-EAU

อาจารย์ที่ปรึกษา

รศ.ดร.โชติพัชร์ ภรณวลัย

T144250

วพ. ว166ก 2557

เลขหมู่.....144250

เลขทะเบียน.....

วัน,เดือน,ปี.....09 มิ.ย. 2559

b.12816668

i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาการศึกษาอิสระ 2

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE APPLICATION OF USING SOFTWARE-DEFINED NETWORKING ON INTERNET SERVICE PROVIDER NETWORK

MANASCHAI JAI-EAU

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS OF THE COURSE

INDEPENDENT STUDY 2

MASTER OF SCIENCE PROGRAMIN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2015

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองการศึกษาอิสระ 2 (Independent Study 2)

เรื่อง

การประยุกต์ใช้งาน Software-Defined Networking บนเครือข่ายผู้ให้บริการอินเทอร์เน็ต

THE APPLICATION OF USING SOFTWARE-DEFINED NETWORKING ON INTERNET SERVICE PROVIDER NETWORK

นายมนัสชัย ใจเอื้อ

รหัสประจำตัว 56606053

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด
รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ
การศึกษาวิชาการศึกษาอิสระ 2 หลักสูตรวิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศ)
ภาคเรียนที่ 2 ปีการศึกษา 2557

.....**อาจารย์ที่ปรึกษา**
(รศ.ดร. โชติพัชร์ ภรณวลัย)

.....**กรรมการสอบ**
(รศ.ดร. วรพจน์ กรีสุระเดช)

.....**กรรมการสอบ**
(ผศ.ดร. สมเกียรติ วังศิริพิทักษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	การประยุกต์ใช้งาน Software-Defined Networking บนเครือข่ายผู้ให้บริการอินเทอร์เน็ต
นักศึกษา	นายมนัสชัย ใจเอื้อ
รหัสนักศึกษา	56606053
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	เทคโนโลยีเครือข่ายและระบบ
ปีการศึกษา	2557
อาจารย์ที่ปรึกษา	รศ.ดร.โชติพัชร์ ภรณวลัย

บทคัดย่อ

ปัจจุบันมีผู้ใช้งานอินเทอร์เน็ตเพิ่มมากขึ้นสอดคล้องกับปริมาณการใช้อินเทอร์เน็ตแบนด์-วิดท์ที่เพิ่มสูงขึ้นอย่างต่อเนื่องเช่นกัน ในขณะที่ผู้ให้บริการอินเทอร์เน็ตประสบกับปัญหาในการบริหารจัดการเครือข่ายซึ่งมีขนาดที่ใหญ่ขึ้น ทั้งปัญหาในด้านของอุปกรณ์ที่มีกลไกการบริหารจัดการในรูปแบบเฉพาะขึ้นอยู่กับผู้ผลิตแต่ละราย ปัญหาการขาดความยืดหยุ่นในการบริหารจัดการที่เกิดจากกลุ่มของอุปกรณ์ในเครือข่ายกรณีผู้ผลิตที่แตกต่างกัน ปัญหาด้านระบบฮาร์ดแวร์ที่มีความสามารถสูงในขณะที่ถูกจำกัดด้วยความสามารถของซอฟต์แวร์จากผู้ผลิต ส่งผลทำให้ระบบเครือข่ายอินเทอร์เน็ตไม่สามารถปรับเปลี่ยนได้อย่างรวดเร็ว และส่งผลต่อการทำงานได้ไม่เต็มประสิทธิภาพเท่าที่ควรจะเป็น ดังนั้นเพื่อให้ระบบเครือข่ายของผู้ให้บริการอินเทอร์เน็ตรองรับการใช้งานได้อย่างรวดเร็ว เพียงพอและมีประสิทธิภาพ ผู้จัดทำโครงการจึงได้ ทำการศึกษาและทดลองแนวคิดของ Software-Defined Networking (SDN) เพื่อนำผลการทดลองมาปรับปรุงและประยุกต์ใช้กับเครือข่ายผู้ให้บริการอินเทอร์เน็ต เพื่อแยกระบบควบคุมออกจากฮาร์ดแวร์ โดยควบคุมอุปกรณ์เครือข่ายทั้งหมดแบบรวมศูนย์ ในการศึกษาและทดลองจะเป็นการทดสอบบนระบบเครือข่ายผู้ให้บริการอินเทอร์เน็ตที่จำลองขึ้นมา ระบบเครือข่ายจำลองจะประกอบด้วยอุปกรณ์เครือข่ายที่รองรับการทำงานในรูปแบบ SDN โดยมีการเขียนคำสั่งสั่งงานผ่านโปรแกรมไปที่คอนโทรลเลอร์ให้ติดต่อกับโครงสร้างของเครือข่าย (Data Plane) ด้วยโพรโทคอล OpenFlow ในช่องสัญญาณที่มีความปลอดภัย ทำให้สามารถเข้าถึงกลไกการทำงานของอุปกรณ์จากทุกผู้ผลิตที่รองรับการทำงานแบบเปิด เป็นการลดการยึดติดกับกลไกการบริหารจัดการอุปกรณ์ในรูปแบบเฉพาะเจาะจง เพิ่มความยืดหยุ่นและสามารถจัดการเครือข่ายได้อย่างมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	The Application of Using Software-Defined Networking on Internet Service Provider Network
Student	Mr. Manaschai Jai-Eau
Student ID.	56606053
Degree	Master of Science
Program	Information Technology
Major	Network and Systems Technology
Academic Year	2014
Advisor	Assoc. Dr. Chotipat Pornavalai

ABSTRACT

The consistently increasing in global Internet usage leads to more demand on network bandwidth. Network administrations have been facing challenges such as managing large networks effectively and boosting customer satisfaction. The complexity of multi vendor's devices also created significant operational challenges such as lacking of flexibility in managing multi vendor's environment. The high capability of hardware is limited by vendors software, hence cannot reach its utmost performance. In this work, we study the concept of Software-Defined Networking (SDN). We also do the experiment using simulator to study the SDN concept on Internet Service Provider network in which consists of a programmable controller and OpenFlow switches. The controller is responsible for commanding the data plane of each switch, via a secure channel using an OpenFlow protocol. The separation of control and data planes reduces the complexity and increases the scalability. SDN allows us to control the underlying network via a single open interface, ignoring the proprietary issue. Thus we can manage complex networks more effectively.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานการศึกษาค้นคว้าอย่างอิสระฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ของ บุคคลหลายท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด

ขอขอบพระคุณ รศ.ดร.โชติพัชร์ ภรณวลัย อาจารย์ที่ปรึกษา ที่กรุณาสละเวลาให้ความรู้ แนวคิด คำแนะนำ รวมทั้งช่วยแก้ไขข้อผิดพลาดต่างๆ เพื่อให้รายงานการศึกษาค้นคว้าอย่างอิสระฉบับนี้มีความถูกต้องและสมบูรณ์ที่สุด

ขอขอบพระคุณ คุณเมทิศ อังคสุวรรณ และคุณสากล จันทร์เมือง Assistant Vice President (AVP) และเพื่อนพนักงานบริษัททริปเปิลทีบรอดแบนด์ จำกัด (มหาชน) ที่ได้ให้แนวคิด และคำปรึกษาเป็นอย่างดี

ขอขอบพระคุณ คุณอังครา ใจเอื้อ ซึ่งเป็นครอบครัวที่ช่วยให้แนวคิด คำปรึกษารวมทั้งเป็นผู้ให้กำลังใจเป็นอย่างดี

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อชัชพล ใจเอื้อ และคุณแม่พิกุล ใจเอื้อ ผู้เป็นเบื้องหลังของความสำเร็จในการศึกษารายงานฉบับนี้ ผู้เป็นที่รัก ที่ได้ให้ความช่วยเหลือสนับสนุนให้กำลังใจอย่างสม่ำเสมอตลอดมา

มนัสชัย ใจเอื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I
ABSTRACT II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VI
สารบัญรูป VII
บทที่ 1 บทนำ
1.1 ความเป็นมาของโครงการ 1
1.2 วัตถุประสงค์ของการพัฒนาระบบ 2
1.3 ขอบเขตในการพัฒนาระบบ 3
1.4 ขั้นตอนในการพัฒนาระบบ 3
1.5 ประโยชน์ที่คาดว่าจะได้รับ 4
บทที่ 2 ทฤษฎีพื้นฐานและเทคโนโลยีที่เกี่ยวข้อง
2.1 เครือข่ายที่กำหนดโดยซอฟต์แวร์ (SOFTWARE-DEFINED NETWORKING) 5
2.2 ชั้นควบคุม (CONTROL PLANE) 7
2.3 ชั้นข้อมูล (DATA PLANE) 8
2.4 โพรโทคอล OPENFLOW 10
2.5 คอนโทรลเลอร์ (CONTROLLER) 15
บทที่ 3 การทำงานของระบบปัจจุบัน
3.1 การศึกษาวิเคราะห์การทำงานของระบบปัจจุบัน 18
3.2 ปัญหาและข้อจำกัดที่พบในระบบปัจจุบัน 19
3.3 การให้บริการของ ผู้ให้บริการอินเทอร์เน็ต (INTERNET SERVICE PROVIDER) 21
บทที่ 4 การวิเคราะห์และออกแบบระบบงาน
4.1 องค์ประกอบของระบบเครือข่าย SOFTWARE-DEFINED NETWORKING (SDN) 23
4.2 การทำงานของระบบเครือข่าย SOFTWARE-DEFINED NETWORKING (SDN) 24
4.3 การออกแบบระบบเครือข่ายผู้ให้บริการอินเทอร์เน็ต ด้วยเทคโนโลยี SDN 25
4.4 การเชื่อมต่อระบบเครือข่ายผู้ให้บริการอินเทอร์เน็ตดาต้าเซ็นเตอร์ ด้วยเทคโนโลยี SDN 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.5 การทำงานของระบบเครือข่ายด้วยเทคโนโลยี SDN 26
บทที่ 5 การทดลอง การทำงาน และการให้บริการของระบบ SDN
5.1 สร้างเครือข่าย SDN 31
5.2 สร้างรูปแบบการสั่งงานจากคอนโทรลเลอร์ 33
5.3 การจำลองส่งทราฟฟิกบนระบบเครือข่าย SDN และระบบเครือข่ายปัจจุบัน 48
5.4 การติดต่อสื่อสารระหว่างคอนโทรลเลอร์และอุปรณ์เครือข่าย SDN 83
บทที่ 6 บทสรุป
6.1 สรุปผลการวิเคราะห์และออกแบบ 90
6.2 ปัญหาและข้อจำกัด 90
6.3 ข้อเสนอแนะ 91
บรรณานุกรม 92
ประวัติผู้เขียน 93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ข่าวสารภายในโพรโทคอล OPENFLOW 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างการทำงานของระบบ SOFTWARE-DEFINED NETWORKING 6
2.2 เปรียบเทียบชั้นควบคุมและชั้นข้อมูลก่อนมี SDN และการมี SDN 7
2.3 ชั้นควบคุม (CONTROL PLANE) และชั้นข้อมูล (DATA PLANE) 9
2.4 การแยกชั้นควบคุมและชั้นข้อมูลออกจากกันของ LINE CARD....10
2.5 การใช้โพรโทคอล OPENFLOW ระหว่างสวิตช์กับคอนโทรลเลอร์ 11
2.6 การทำงานแบบ PIPE LINE ของตารางไหลของข้อมูล 12
2.7 การสื่อสารระหว่างคอนโทรลเลอร์กับสวิตช์ 12
2.8 ส่วนประกอบภายในตารางข้อมูล OPENFLOW (FLOW TABLE) 13
2.9 ตัวอย่างการจำลองสร้างอุปกรณ์เครือข่ายและการเชื่อมต่อ ด้วยโปรแกรม MININET....16
3.1 ตัวอย่างการทำงานของอุปกรณ์เร้าเตอร์ในปัจจุบัน 19
3.2 ปัญหาด้านการไม่ทำงานร่วมกันในภาพรวมของอุปกรณ์เครือข่าย 19
3.3 รูปแบบการจัดการอุปกรณ์เครือข่ายของบริษัท CISCO SYSTEM....20
3.4 รูปแบบการจัดการของอุปกรณ์เครือข่ายของบริษัท JUNIPER....20
3.5 ตัวอย่างการทำงานของระบบเครือข่ายพื้นฐานของผู้ให้บริการอินเทอร์เน็ตปัจจุบัน 22
4.1 องค์ประกอบของระบบเครือข่าย SOFTWARE-DEFINED NETWORKING (SDN) 23
4.2 การทำงานของระบบเครือข่าย SOFTWARE-DEFINED NETWORKING (SDN) 25
4.3 การเชื่อมต่อระบบเครือข่ายผู้ให้บริการอินเทอร์เน็ตดาต้าเซ็นเตอร์ด้วยเทคโนโลยี SDN....26
4.4 การเชื่อมต่อระหว่างอุปกรณ์ SDN และคอนโทรลเลอร์ด้วย OPENFLOW โพรโทคอล....27
4.5 การจัดสรร IP ADDRESS ในเครือข่ายผู้ให้บริการอินเทอร์เน็ตดาต้าเซ็นเตอร์ 28
4.6 การไหลของทราฟฟิกของดาต้าเซ็นเตอร์ แห่งที่ 1....29
4.7 การไหลของทราฟฟิกของดาต้าเซ็นเตอร์ แห่งที่ 2 กรณีมีการใช้งาน UPLINK น้อยกว่า 80%..30
4.8 การส่งทราฟฟิกของดาต้าเซ็นเตอร์แห่งที่ 2 กรณีมีการใช้งาน UPLINK มากกว่า 80%....30
5.1 การกำหนดค่า เพื่อสร้างเครือข่าย SDN ด้วย MININET....31
5.2 การสร้างระบบเครือข่าย SDN ด้วย MININET....32
5.3 การตรวจสอบสถานะการเชื่อมต่อระบบเครือข่าย SDN ด้วย MININET 33
5.4 ผลการเชื่อมต่อจากคอนโทรลเลอร์เมื่อมีการเชื่อมต่อกับอุปกรณ์สวิตช์ผ่าน OPENFLOW โพรโทคอล....34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

5.5 ผลจากคอนโทรลเลอร์เมื่อหยุดเชื่อมต่อกับสวิตซ์ผ่าน OPENFLOW โพรโทคอล....34
5.6 ผลลัพธ์ของการติดต่อกันระหว่าง HOST ในเครือข่าย SDN....35
5.7 การกำหนดเงื่อนไขการเชื่อมต่อและการจัดการการเชื่อมต่อด้วย ARP โพรโทคอลบน สวิตซ์ SDN....35
5.8 การกำหนดเงื่อนไขในการส่งต่อข้อมูลด้วยเงื่อนไข IP ADDRESS และพอร์ต....36
5.9 แสดงค่าคอนฟิกกูเรชั่น IP ADDRESS บนเครื่อง HOST 1 และ HOST 4 ตามการกำหนดค่า จาก MININET....36
5.10 แสดงค่าคอนฟิกกูเรชั่น IP ADDRESS บนเครื่อง HOST 3 และ HOST 6 ตามการกำหนดค่า จาก MININET....37
5.11 คำสั่งงานบนคอนโทรลเลอร์สั่งงานไปที่สวิตซ์ 5 สำหรับดาต้าเซ็นเตอร์ แห่งที่ 1....38
5.12 คำสั่งงานบนคอนโทรลเลอร์สั่งงานไปที่สวิตซ์ 1 สำหรับดาต้าเซ็นเตอร์ แห่งที่ 1....38
5.13 คำสั่งงานบนคอนโทรลเลอร์สั่งงานไปที่สวิตซ์ 5 สำหรับดาต้าเซ็นเตอร์ แห่งที่ 2....40
5.14 การส่งแพ็กเก็ตจาก HOST1 ไปยัง HOST4....40
5.15 กราฟแสดงค่าข้อมูลแพ็กเก็ตที่มีการส่งออกไปจาก พอร์ต 4 ที่ สวิตซ์ 5....41
5.16 การแสดงค่าข้อมูลแพ็กเก็ตที่มีการส่งออกไปจากพอร์ต 4 ที่ สวิตซ์ 5....41
5.17 การส่งแพ็กเก็ตจาก HOST3 ไปยัง HOST6....42
5.18 การแสดงค่าข้อมูลแพ็กเก็ตที่มีการส่งออกไปจากพอร์ต 5 ที่สวิตซ์ 5....42
5.19 การแสดงค่าข้อมูลแพ็กเก็ตที่มีการส่งออกไปจากพอร์ต 5 ที่สวิตซ์ 5....43
5.20 การตรวจสอบสถานะการใช้งานของพอร์ตและกำหนดเงื่อนไข....44
5.21 การทดสอบสร้าง IPERF MODE SERVER ที่ HOST 9....44
5.22 การทดสอบสร้าง IPERF MODE CLIENT เพื่อส่งทราฟฟิกปริมาณ 1 MBPS จาก HOST 7 ไปยัง HOST 9....45
5.23 กราฟผลลัพธ์จากคอนโทรลเลอร์ที่รับมาจากสวิตซ์ 5 กรณีส่งทราฟฟิกที่ HOST 7 ไปยัง HOST 9 กรณีมีการใช้งานน้อยกว่าเงื่อนไขที่กำหนดไว้....45
5.24 ผลลัพธ์จากคอนโทรลเลอร์ที่รับมาจากสวิตซ์ 5 กรณีส่งทราฟฟิกที่ HOST 7 ไปยัง HOST 9 กรณีมีการใช้งานน้อยกว่าเงื่อนไขที่กำหนดไว้....46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

5.25 การทดสอบสร้าง IPERF MODE CLIENT และส่งทราฟฟิกปริมาณ 9 MBPS ที่ HOST7 ไปยัง HOST9 46
5.26 กราฟผลลัพธ์จากคอนโทรลเลอร์ที่รับมาจากสวิตซ์ 5 กรณีส่งทราฟฟิกที่ HOST7 ไปยัง HOST9 กรณีมีการใช้งานมากกว่าเงื่อนไขที่กำหนดไว้ 47
5.27 ผลลัพธ์จากคอนโทรลเลอร์ที่รับมาจากสวิตซ์ 5 กรณีส่งทราฟฟิกที่ HOST 7 ไปยัง HOST 9 กรณีมีการใช้งานมากกว่าเงื่อนไขที่กำหนดไว้ 47
5.28 สถานะการส่งทราฟฟิกในกรณีพื้นฐาน 48
5.29 แสดงการตรวจสอบสถานะปริมาณการใช้งานทราฟฟิก ด้วยคอนโทรลเลอร์ 49
5.30 แสดงการทดลองส่งทราฟฟิกจาก HOST 7 ไปยัง HOST 9 50
5.31 แสดงการทดลองส่งทราฟฟิกจาก HOST 3 ไปยัง HOST 10 51
5.32 แสดงการเรียกดูค่าปริมาณการใช้งานทราฟฟิก จากคอนโทรลเลอร์ไปที่ สวิตซ์ S5 51
5.33 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.1 52
5.34 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.1 53
5.35 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.1 54
5.36 การไหลของทราฟฟิกเมื่อเกิดการแบ่งทราฟฟิกจากคอนโทรลเลอร์ 55
5.37 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.2.1 55
5.38 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.2.1 56
5.39 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.2.1 57
5.40 การทดลองส่งทราฟฟิกจาก HOST 7 ไปยัง HOST 9 ตามเงื่อนไขที่ 5.3.2.1 57
5.41 การทดลองส่งทราฟฟิกจาก HOST 3 ไปยัง HOST 10 ตามเงื่อนไขที่ 5.3.2.1 58
5.42 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่าย SDN ในการทดลองที่ 5.3.2.1 58
5.43 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.2.1 59
5.44 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.2.2 60
5.45 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.2.2 60
5.46 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.2.2 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

5.47 การทดลองส่งทราฟฟิกจาก HOST 7 ไปยัง HOST 9 ตามเงื่อนไขที่ 5.3.2.2 61
5.48 การทดลองส่งทราฟฟิกจาก HOST 3 ไปยัง HOST 10 ตามเงื่อนไขที่ 5.3.2.2 62
5.49 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่าย SDN ในการทดลองที่ 5.3.2.2.....63
5.50 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่าย ปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.2.2.....64
5.51 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.2.3.....64
5.52 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.2.3.....65
5.53 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.2.3 66
5.54 การทดลองส่งทราฟฟิกจาก HOST 7 ไปยัง HOST 9 ตามเงื่อนไขที่ 5.3.2.3 66
5.55 การทดลองส่งทราฟฟิกจาก HOST 3 ไปยัง HOST 10 ตามเงื่อนไขที่ 5.3.2.3 67
5.56 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่าย SDN ในการทดลองที่ 5.3.2.3.....67
5.57 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่าย ปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.2.3.....68
5.58 สัดส่วนของทราฟฟิกในการทดลองบนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.3.1.....69
5.59 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.3.1.....70
5.60 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.3.1 70
5.61 การทดลองส่งทราฟฟิกจาก HOST 7 ไปยัง HOST 9 ตามเงื่อนไขที่ 5.3.3.1 71
5.62 การทดลองส่งทราฟฟิกจาก HOST 3 ไปยัง HOST 10 ตามเงื่อนไขที่ 5.3.3.1 71
5.63 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่าย SDN ในการทดลองที่ 5.3.3.1.....72
5.64 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่าย ปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.3.1.....73
5.65 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.3.2.....73
5.66 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.3.2.....74
5.67 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.3.2 75
5.68 การทดลองส่งทราฟฟิกจาก HOST 7 ไปยัง HOST 9 ตามเงื่อนไขที่ 5.3.3.2 75
5.69 การทดลองส่งทราฟฟิกจาก HOST 3 ไปยัง HOST 10 ตามเงื่อนไขที่ 5.3.3.2.....76
5.70 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่าย SDN ในการทดลองที่ 5.3.3.2.....76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

5.71 กราฟแสดงจำนวนเวลาที่ใช้ในการส่งข้อมูลทราฟฟิกเปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.3.2 77
5.72 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.3.3 78
5.73 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.3.3 78
5.74 สัดส่วนของทราฟฟิกในการทดลองบนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.3.3 79
5.75 การทดลองส่งทราฟฟิกจาก HOST 7 ไปยัง HOST 9 ตามเงื่อนไขที่ 5.3.3.3 79
5.76 การทดลองส่งทราฟฟิกจาก HOST 3 ไปยัง HOST 10 ตามเงื่อนไขที่ 5.3.3.3 80
5.77 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่าย SDN ในการทดลองที่ 5.3.3.3 81
5.78 กราฟแสดงจำนวนเวลาที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.3.3 82
5.79 กราฟสรุปแสดงจำนวนเวลาที่ใช้ในการส่งข้อมูลทราฟฟิกในการทดลองที่ 5.3 83
5.80 การติดต่อสื่อสารของคอนโทรลเลอร์ในกรณีทราฟฟิกมีความหนาแน่น 84
5.81 การติดต่อสื่อสารของคอนโทรลเลอร์ ในกรณีแยกช่องทางทราฟฟิกพิเศษ 85
5.82 การกระจาย หน่วยทำงานของคอนโทรลเลอร์ 87
5.83 หน่วยทำงานคอนโทรลเลอร์ บนอุปกรณ์เครือข่าย 88
5.84 หน่วยทำงานคอนโทรลเลอร์บนอุปกรณ์เครือข่ายกรณีคอลโทรลเลอร์หลักมีปัญหา 88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงการ

ตั้งแต่อดีตจนถึงปัจจุบัน โลกมีการพัฒนาขึ้นมากทั้งด้านอุปกรณ์การสื่อสาร อาทิเช่น โทรศัพท์เคลื่อนที่ ที่ผ่านมาสามารถรับ-ส่งได้เฉพาะเสียงและข้อความ แต่ตอนนี้สามารถรับ-ส่งข้อมูลได้ เรียกว่า สมาร์ทโฟน (Smart Phone) คอมพิวเตอร์แบบพกพา (Tablet) ที่มีขนาดเล็ก พกพาสะดวก ด้านบริการเครือข่ายสังคม อาทิเช่น เฟสบุ๊ค (Facebook) ทวิตเตอร์ (Twitter) และไลน์ (Line) เป็นต้น ที่ช่วยให้บุคคลสามารถติดต่อสื่อสารกันทั้งภายในประเทศและต่างประเทศผ่านเครือข่ายอินเทอร์เน็ตได้อย่างสะดวกขึ้นและด้านแอปพลิเคชัน (Application) ที่สามารถนำไปติดตั้งบนสมาร์ทโฟนและคอมพิวเตอร์แบบพกพา มีหลากหลายกลุ่ม อาทิเช่น กลุ่มข่าวสาร ที่ช่วยให้สามารถเปิดอ่านข่าวจากสำนักพิมพ์ต่างๆ แบบออนไลน์ได้แทนการอ่านจากหนังสือพิมพ์ กลุ่มแผนที่ ที่สามารถช่วยค้นหาเส้นทางไปยังจุดหมายและช่วยนำทางได้อย่างสะดวก โดยไม่ต้องอาศัยอุปกรณ์นำทางติดรถยนต์ เป็นต้น

ผลจากการคิดค้นพัฒนาที่ย่อให้โลกแคบลงและสามารถติดต่อสื่อสารกันอย่างสะดวกและรวดเร็วขึ้นผ่านเครือข่ายอินเทอร์เน็ตทำให้การใช้งานอินเทอร์เน็ตเติบโตขึ้นอย่างรวดเร็ว ข้อมูลในเดือนสิงหาคม ปี พ.ศ. 2552 มีปริมาณการใช้อินเทอร์เน็ตแบนด์วิดท์ภายในประเทศ จำนวน 272,427 Mbps และในเดือน สิงหาคม ปี พ.ศ. 2557 จำนวน 1,833,685 Mbps เพิ่มขึ้น 6.73 เท่า หรือคิดเป็น 573.09 % ส่วนปริมาณการใช้อินเทอร์เน็ตแบนด์วิดท์ไปต่างประเทศ เดือน สิงหาคม ปี พ.ศ. 2552 จำนวน 70,561 Mbps และในเดือน สิงหาคม ปี พ.ศ. 2557 จำนวน 1,008,655 Mbps เพิ่มขึ้น 14.29 เท่า หรือคิดเป็น 1,329.48 % (งานวิจัยข้อมูลอินเทอร์เน็ต, หน่วยปฏิบัติการวิจัยเทคโนโลยีเครือข่าย, ศูนย์เทคโนโลยีอิเล็กทรอนิกส์ และคอมพิวเตอร์แห่งชาติ, 2557: ออนไลน์) ซึ่งการใช้งานอินเทอร์เน็ตเพิ่มขึ้นอย่างต่อเนื่องในทศวรรษที่ผ่านมา ในด้านของผู้ให้บริการเครือข่ายมีความจำเป็นต้องขยายเครือข่ายให้รองรับปริมาณการใช้งานที่เพิ่มขึ้นเช่นเดียวกัน

ปัญหาของผู้ให้บริการเครือข่ายเพื่อให้สามารถรองรับการบริการแก่ผู้ใช้งานอย่างเพียงพอมีหลายประการ ได้แก่ การขึ้นตรงต่อผู้ผลิต ในด้านผู้ให้บริการต่างมีความประสงค์ให้เครือข่ายรองรับการตอบสนองทางธุรกิจและผู้ใช้งาน แต่อุปสรรคเกิดจากการที่ผู้ผลิตมีอุปกรณ์ที่มักจะมีวงจรการใช้งานระยะสั้นประมาณ 3 ปีหรือมากกว่านั้น การตั้งค่าของอุปกรณ์ที่แตกต่างกันตามแต่ผู้ผลิตแต่ละราย การบริหารจัดการอุปกรณ์ที่แตกต่างกัน แต่ผู้ให้บริการต้องทำให้สามารถทำงานร่วมกันได้อย่างราบรื่น อุปกรณ์มีลักษณะปิด คือ การขาดการจัดการอุปกรณ์ร่วมกันได้ ระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ผลิตที่แตกต่างกัน นอกจากการผ่านเข้าถึงผ่านโพรโทคอลเฉพาะ Telnet หรือ SSH ระบบฮาร์ดแวร์มีความสามารถในการทำงานที่สูงและมีการพัฒนาไปรวดเร็ว แต่ถูกจำกัดความสามารถจากระบบซอฟต์แวร์ของผู้ผลิต อีกทั้งการทำงานพิเศษบางอย่างจำเป็นต้องติดตั้งใบอนุญาต (License) เพิ่มและอาจมีการจำกัดอายุการใช้งานด้วย นอกจากนี้งบประมาณของผู้ให้บริการ เป็นอีกปัจจัยสำคัญที่อาจจะมีเท่าเดิมหรือลดลง ซึ่งในฐานะผู้ให้บริการมีความประสงค์ที่จะให้อุปกรณ์ที่มีอยู่สามารถทำงานได้ตามความสามารถเต็มตามที่ควรจะเป็นอย่างมากที่สุด

จากความหลากหลายของปัญหาดังกล่าว ผู้จัดทำโครงการจึงมีความประสงค์ ที่จะทำการศึกษา ทดลอง นำหลักการของเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software-Defined Networking) หรือ SDN ที่มีแนวคิดในการบริหารจัดการระบบเครือข่ายที่แยกระบบควบคุม ออกจากฮาร์ดแวร์มาประยุกต์ใช้กับระบบเครือข่าย โดยจัดทำซอฟต์แวร์ แอปพลิเคชัน ที่เรียกว่า Controller มาควบคุมการทำงานจากระยะไกลอาศัยหลักการพื้นฐานเดิมของอุปกรณ์เครือข่ายที่มีการแบ่งเป็นชั้นควบคุม (Control Plane) ที่เป็นส่วนทำหน้าที่ตัดสินใจส่งข้อมูลและชั้นของข้อมูล (Data Plane) ที่ทำหน้าที่ส่งข้อมูลไปตามการตัดสินใจของชั้นควบคุม โดย SDN ใช้แนวคิดที่ว่าชั้นควบคุมและชั้นของข้อมูลแยกการทำงานอย่างชัดเจน แล้วใช้ตัวเชื่อมการทำงาน (Application Programming Interface) เพื่อประสานการทำงานของทั้งสองส่วนนี้เข้าด้วยกัน โดยดึงเอาการทำงานของชั้นควบคุมทั้งหมดมารวมไว้ในจุดเดียวและทำการควบคุมอุปกรณ์เครือข่ายทั้งหมดแบบรวมศูนย์ โดย Controller ติดต่อกับโครงสร้างของเครือข่ายด้วยโพรโทคอล OpenFlow ในช่องสัญญาณที่มีความปลอดภัย ทำให้สามารถเข้าถึงกลไลการทำงานของอุปกรณ์จากทุกผู้ผลิตที่รองรับการทำงานแบบเปิด เป็นการลดการยึดติดกับกลไกการบริหารจัดการอุปกรณ์ในรูปแบบเฉพาะเจาะจง เพิ่มความยืดหยุ่นและสามารถจัดการเครือข่ายได้อย่างมีประสิทธิภาพได้มากขึ้น

1.2 วัตถุประสงค์ของการพัฒนาระบบ

1) เพื่อศึกษาและทดลองระบบเครือข่ายที่ทำงานในรูปแบบ Software-Defined Networking (SDN)
2) เพื่อศึกษาและทดลองแนวทางการทำงานของอุปกรณ์เครือข่ายในรูปแบบ SDN เพื่อนำมาประยุกต์ใช้ และเพิ่มประสิทธิภาพในการจัดการ การทำงานอุปกรณ์เครือข่ายแบบรวมศูนย์
3) เพื่อศึกษาและทดลองแนวทางการลดปัญหาในการทำงานร่วมกันของอุปกรณ์เครือข่ายและการตรวจสอบสถานะค่า การใช้ปริมาณทราฟฟิกที่ผ่านอุปกรณ์ จากศูนย์กลางได้อย่างมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เพื่อศึกษาและทดลองสั่งงานอุปกรณ์เครือข่าย ด้วยรูปแบบการสั่งงานของ SDN เพื่อเพิ่มประสิทธิภาพ ในการบริการจัดการอุปกรณ์บนเครือข่าย ผู้ให้บริการเครือข่ายอินเทอร์เน็ต

1.3 ขอบเขตในการพัฒนาระบบ

การประยุกต์ใช้งาน Software-Defined Networking (SDN) บนเครือข่ายผู้ให้บริการอินเทอร์เน็ต มีขอบเขตในการพัฒนาระบบดังนี้

1) ติดตั้งซอฟต์แวร์คอนโทรลเลอร์ (Controller) เพื่อทำงานบนระบบปฏิบัติการลินุกซ์ (Linux)
2) เขียนคำสั่ง สั่งงานผ่านโปรแกรมไปที่คอนโทรลเลอร์ โดยใช้โพรโทคอล Open Flow เพื่อใช้ในการติดต่อสื่อสารระหว่างชั้นควบคุม (Control Plane) และชั้นข้อมูล (Data Plane)
3) เขียนคำสั่ง สั่งงานผ่านโปรแกรมไปคอนโทรลเลอร์ (Controller) เพื่อให้แสดงสถานะของระบบเครือข่ายได้อย่างมีประสิทธิภาพ
4) เขียนคำสั่ง สั่งงานผ่านโปรแกรมไปยังคอนโทรลเลอร์ (Controller) เพื่อใช้ในการสั่งงานในเรื่องการจัดการเส้นทางการส่งทราฟฟิค บริหารจัดการอุปกรณ์เครือข่ายได้อย่างมีประสิทธิภาพ

1.4 ขั้นตอนในการพัฒนาระบบ

ศึกษาและประยุกต์ใช้งานระบบ Software-Defined Networking (SDN) บนเครือข่ายผู้ให้บริการอินเทอร์เน็ต โดยมีขอบเขตการศึกษาและพัฒนาระบบดังนี้

1) ศึกษาอุปกรณ์เครือข่ายแบบเดิม วิเคราะห์ขั้นตอน และกระบวนการทำงานของชั้นควบคุม (Control Plane) และชั้นข้อมูล (Data Plane)
2) ศึกษาเทคโนโลยี “Software-Defined Networking” โครงสร้างและกระบวนการทำงาน
   2.1) ศึกษาการทำงานของระบบประมวลผล (Controller) ที่ทำงานบน SDN
   2.2) ศึกษาการทำงานของโพรโทคอล “OpenFlow” ที่ทำงานบน SDN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ศึกษาและทดลองเขียนคำสั่ง สั่งงานผ่านโปรแกรมไปที่ระบบการประมวลผล (Controller) ทดสอบการทำงานบนอุปกรณ์เครือข่ายที่มีการแยกการทำงานกันของชั้นควบคุมและชั้นข้อมูล
4) นำเทคโนโลยี SDN มาทำการทดสอบ การสั่งงาน การส่งต่อข้อมูล และการทำงานร่วมกันของอุปกรณ์เครือข่าย บนระบบเครือข่ายจำลอง
5) แสดงให้เห็นถึงการสั่งงานและระบบตรวจสอบการใช้งานบนเครือข่ายของระบบประมวลผล (Controller) ที่มีการทำงานโดยอัตโนมัติ
6) สรุปผลการวิจัยและข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

จากการศึกษาและพัฒนาระบบ Software-Defined Networking และนำไปประยุกต์ใช้บนเครือข่ายผู้ให้บริการอินเทอร์เน็ต ก่อให้เกิดประโยชน์หลากหลายประการ ดังนี้

1) ลดปัญหาในการบริหารจัดการอุปกรณ์เครือข่ายจากหลากหลายผู้ผลิต ที่มีกลไกการทำงานในรูปแบบเฉพาะที่แตกต่างกัน
2) เพิ่มประสิทธิภาพในการทำงานให้กับอุปกรณ์เครือข่ายมากยิ่งขึ้นอันเนื่องมาจากข้อจำกัดด้านซอฟต์แวร์
3) เพิ่มประสิทธิภาพในการตรวจสอบระบบเครือข่ายให้มีประสิทธิภาพมากยิ่งขึ้น
4) เพิ่มความคล่องตัวในการบริหารจัดการด้านเส้นทางและเกิดการบริการที่ดียิ่งขึ้น
5) ลดเรื่องค่าใช้จ่ายในส่วนของอุปกรณ์เครือข่ายได้มากยิ่งขึ้น
6) เพิ่มความยืดหยุ่นและความสามารถในการขยายเครือข่ายมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

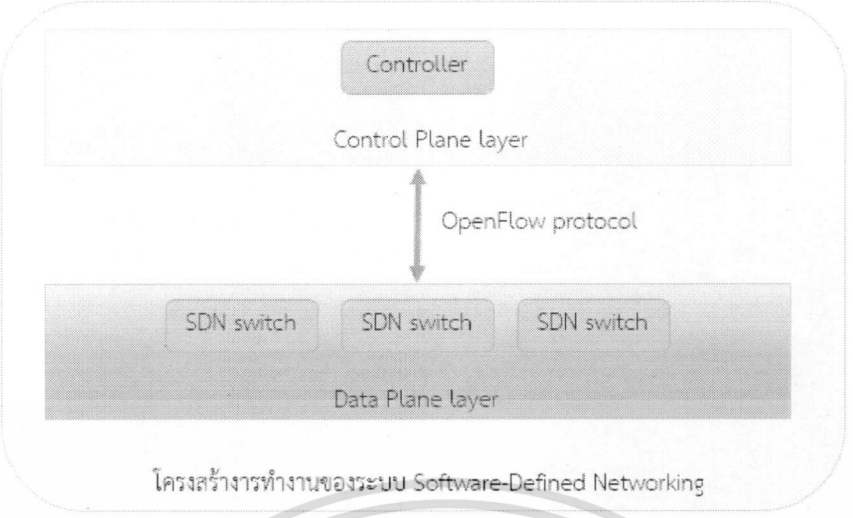
บทที่ 2
ทฤษฎีพื้นฐานและเทคโนโลยีที่เกี่ยวข้อง

2.1 เครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software-Defined Networking)

Software-Defined Networking หรือ SDN คือ วิธีการจัดการระบบเครือข่ายคอมพิวเตอร์ ที่สามารถทำให้ผู้บริหารเครือข่ายสามารถจัดการและบริหารการให้บริการในเครือข่ายได้อย่างมีประสิทธิภาพ สามารถปรับเปลี่ยนเครือข่ายได้อย่างรวดเร็วตอบสนองต่อแผนทางธุรกิจได้อย่างทันท่วงที โดยวิธีการดำเนินการบนอุปกรณ์เครือข่ายมีการแยกการทำงาน ออกเป็น 2 ชั้นการทำงาน คือ ชั้นการตัดสินใจ (Control Plane) และ ชั้นการส่งกระแสจราจรของข้อมูล (Data Plane) ชั้นที่ทำหน้าที่ในการตัดสินใจ เปรียบเทียบได้กับสมอง โดยทำการรวบรวมเส้นทางปลายทาง ของการจัดส่งกระแสจราจรของข้อมูล และตัดสินเลือกเส้นทางที่ดีที่สุด เพื่อจัดส่งกระแสจราจรของข้อมูล (Traffic)ไปยังปลายทาง โดยเรียกการทำงานนี้ว่า Control Plane และ ชั้นที่ทำหน้าที่ในการส่งกระแสจราจรของข้อมูล ไปยังปลายทาง โดยจะรับคำสั่งและข้อมูลของเส้นทางปลายทางมาจากชั้นการตัดสินใจ เพื่อใช้ในการส่งกระแสจราจร โดยสามารถเปรียบชั้นหน้าที่การส่งกระแสจราจรของข้อมูล ได้กับกล้ามเนื้อของร่างกายมนุษย์ โดยเรียกการทำงานนี้ว่า Data Plane

โครงสร้างการทำงานของระบบ Software-Defined Networking มีหน่วยทำงาน เรียกว่า Controller มีหน้าที่เปรียบได้กับ Control plane คือทำงานเป็น สมองของระบบเครือข่าย ทำหน้าที่ในการรวบรวมข้อมูล การคิด การตัดสินใจ และการสั่งงานบนระบบเครือข่ายทั้งหมด มีลักษณะรวมศูนย์ และส่งต่อคำสั่งนี้ต่อไปยังชั้นทำงาน Data Plane เพื่อทำการส่งกระแสข้อมูลต่อไป ยังปลายทางที่กำหนด ในการติดต่อสื่อสารระหว่าง Controller และ Data Plane ผู้ดูแลระบบสามารถบริหารจัดการเครือข่ายและกำหนดเส้นทาง โดยใช้ซอฟต์แวร์ที่เรียกใช้งานตัวเชื่อมการทำงาน แบบรวมศูนย์ ณ จุดเดียว หน่วยการทำงาน Controller จะติดต่อสื่อสารไปยังโครงสร้างเครือข่ายผ่านโพรโทคอล ที่เรียกว่า OpenFlow โดยโครงสร้างการทำงานของระบบ Software-Defined Networking สามารถแสดงรายละเอียดได้ดังรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 โครงสร้างการทำงานของระบบ Software-Defined Networking

จากการทำงานของ SDN ที่มีแนวคิดของการแยกการทำงานอย่างชัดเจนระหว่างชั้นควบคุมในการตัดสินใจกำหนดเส้นทางการไหลของข้อมูล (Control Plane) กับ โครงสร้างการไหลของข้อมูล (Data Plane) ผลที่ได้จากการใช้งานระบบเครือข่ายในรูปแบบ SDN กล่าวถึงได้ 3 ประการคือ

1) ผู้ดูแลระบบไม่ต้องคำนึงถึง การขึ้นกับผู้ผลิตซึ่งปัจจุบันอุปกรณ์จากแต่ละผู้ผลิตมีระบบการควบคุมจัดการที่แตกต่างกัน ระบบการทำงานแตกต่างกันโดยจะมีเพียงในส่วนที่ต้องทำงานร่วมกันระหว่างอุปกรณ์ ที่อาศัยมาตรฐานกลางหรือโพรโทคอลกลางในการสื่อสารกันเพื่อให้สามารถทำงานร่วมกันได้ เช่น การทำงานเรื่องการแลกข้อมูลซึ่งข้อมูลที่แลกเปลี่ยนกัน จะต้องตรงตามมาตรฐานที่กำหนด เพื่อนำข้อมูลมาใช้คำนวณหาเส้นทางที่ดีที่สุดในการส่งกระแสข้อมูลไปยังปลายทาง แต่เมื่อเกิดปัญหาในระบบเครือข่าย กระบวนการทำงานยังคงมีวิธีการแก้ไขที่แตกต่างกัน ดังนั้นเมื่อมีการใช้ SDN จะมีการบริหารจัดการ ณ จุดเดียวกันโดย สามารถใช้กับอุปกรณ์เครือข่ายโดยไม่ขึ้นกับผู้ผลิตด้วยการควบคุมของโพรโทคอล กลางที่เรียกว่า OpenFlow ดังรูปที่ 2.2
2) เครือข่ายมีความยืดหยุ่นและขยายตัวได้ง่ายขึ้น จากการใช้ SDN ที่ทำให้ไม่ขึ้นกับผู้ผลิตรายใดรายหนึ่ง มีการบริหารจัดการจากศูนย์กลาง ส่งผลให้สามารถปรับเปลี่ยนระบบเครือข่ายรูปแบบการเชื่อมต่อ ได้รวดเร็วตามความต้องการมากขึ้น และทำให้สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดซื้ออุปกรณ์เครือข่ายโดยมีการแข่งขันกันได้ดีขึ้น ไม่ผูกขาดอยู่กับผู้ผลิตรายใดรายหนึ่ง ช่วยให้ขยายเครือข่ายและลดต้นทุนได้มากขึ้น

3) ผู้ดูแลระบบสามารถจัดการเครือข่ายให้มีประสิทธิภาพมากขึ้น คือ สามารถสั่งงานไปยังอุปกรณ์เครือข่าย จำนวนมากได้อย่างรวดเร็ว มีความถูกต้องในการสั่งงาน การตรวจสอบสถานะการใช้งานปริมาณการไหลของข้อมูล ผ่านชั้นควบคุม (Control Plane) ที่ทำงานอยู่บน Controller ซึ่งทำงานเป็นลักษณะศูนย์กลาง โดย Controller จะติดต่อสื่อสาร ผ่านโพรโตคอล OpenFlow ไปที่ชั้นโครงสร้างการไหลของข้อมูล(Data Plane)

รูปที่ 2.2 เปรียบเทียบชั้นควบคุมและชั้นข้อมูลก่อนมี SDN และการมี SDN

2.2 ชั้นควบคุม (Control Plane)

ชั้นควบคุม (Control Plane) มีหน้าที่หลัก คือ การแลกเปลี่ยนข้อมูลเส้นทาง ระหว่างชั้นควบคุมอื่นในเครือข่าย โดยชั้นเครือข่ายสามารถอยู่บนอุปกรณ์เครือข่ายจากผู้ผลิตที่แตกต่าง หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

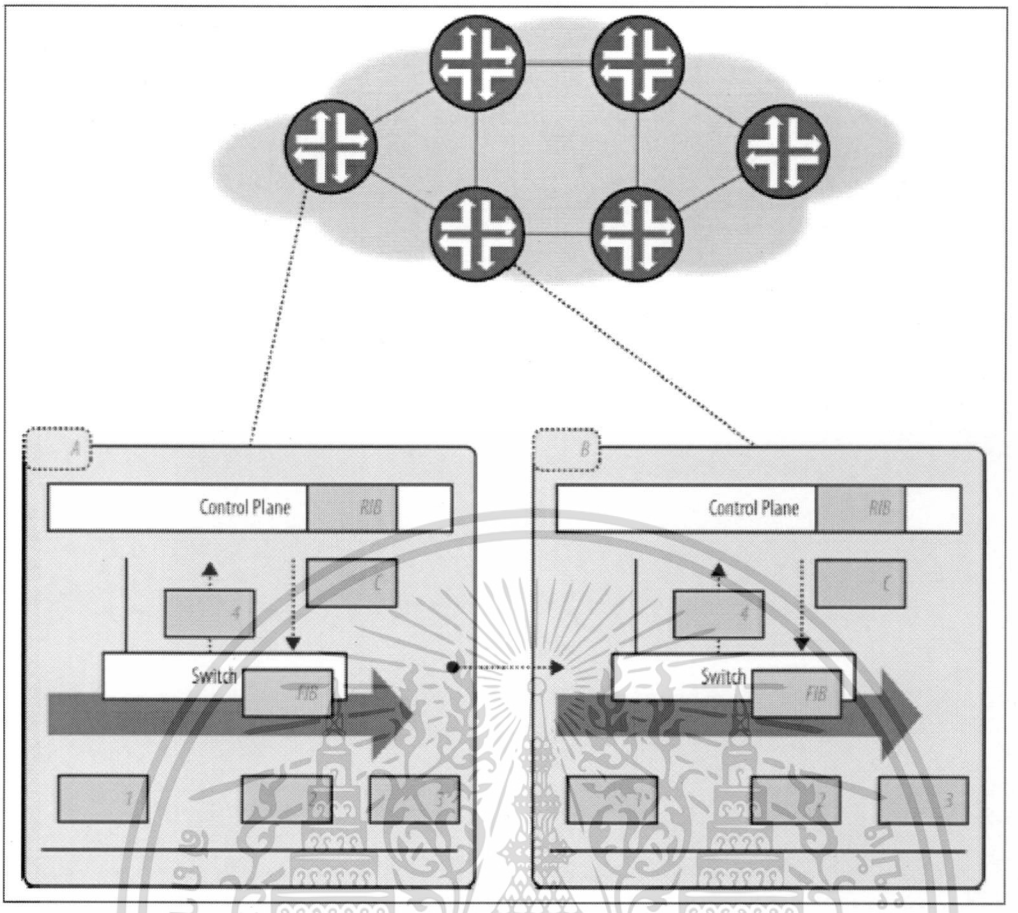
จากผู้ผลิตรายเดียวกันก็ได้ และจะทำการรวบรวมข้อมูลเส้นทางปลายทาง เพื่อเลือกเส้นทางที่ดีที่สุด ซึ่งข้อมูลดังกล่าว จะถูกเก็บเป็นข้อมูลโครงสร้างระบบเครือข่ายในอุปกรณ์เครือข่าย ข้อมูลนี้ถูกเรียกว่าเร้าติ้ง (Routing Information Base:RIB) โดยมีหน้าที่หลัก คือ เมื่อมีการไหลของข้อมูล อุปกรณ์เครือข่ายจะใช้ข้อมูลจากเร้าติ้ง ในการส่งต่อข้อมูลไปยังปลายทางซึ่งหากไม่มีข้อมูลเส้นทางปลายทางในข้อมูลเร้าติ้ง อุปกรณ์เครือข่ายก็ไม่สามารถส่งต่อข้อมูลได้ ส่งผลทำให้ข้อมูลถูกเพิกเฉย หน้าที่ของชั้นข้อมูลยังมีหน้าที่อื่นอีก เช่น การเรียนรู้ MAC Address บนอุปกรณ์สวิตซ์ การจัดการแบบง่ายบนอุปกรณ์เครือข่าย (Simple Network Management Protocol) การเข้าถึงอุปกรณ์เครือข่ายเพื่อตรวจสอบสถานะการใช้งานและสั่งงาน จากระยะไกล (Telnet, SSH)

2.3 ชั้นข้อมูล (Data Plane)

ชั้นข้อมูล (Data Plane) มีหน้าที่ในการจัดการกับ แพ็กเก็ต ที่เข้ามายัง อุปกรณ์เครือข่าย โดยผ่านมาทางช่องทางอินเทอร์เฟซ เช่น ไฟเบอร์ สายทองแดง หรือระบบไร้สาย ในกระบวนการจัดการกับแพ็กเก็ต อุปกรณ์จะต้องพิจารณาจากข้อมูล Forwarding Information Base (FIB) ซึ่งข้อมูล FIB จะนำมาจาก RIB และนำมาติดตั้งลงในชั้นข้อมูลเพื่อระบุถึงเงื่อนไขในการส่งต่อข้อมูลแพ็กเก็ตมีการระบุเงื่อนไขการส่งข้อมูลไปยังปลายทางว่าจะต้องส่งออกไปทางอินเทอร์เฟซใดบนอุปกรณ์เครือข่ายนั้นๆ

กระบวนการในการทำงานร่วมกันระหว่างชั้นควบคุมและชั้นข้อมูล จากรูปที่ 2.3 สวิตซ์ A และ B แสดงถึงกระบวนการที่ชั้นควบคุมและชั้นข้อมูลอยู่ในอุปกรณ์ตัวเดียวกัน เมื่อมีข้อมูลแพ็กเก็ตเข้ามายังอุปกรณ์ ผ่านเข้ามาในส่วนของชั้นข้อมูลจากนั้นในข้อมูลจะตรวจสอบถึงข้อมูลเลขที่อยู่ไอพีปลายทางจากตาราง FIB ว่าจะต้องส่งออกทางอินเทอร์เฟซใด จากนั้นทำการส่งต่อข้อมูลแพ็กเก็ต ไปยังอุปกรณ์ถัดไป จากรูป ข้อมูลแพ็กเก็ตจะรับเข้ามาทาง สวิตซ์ A และ ส่งต่อไปยังสวิตซ์ B ตามหมายเลข 1, 2 และ 3 แต่สำหรับกรณีในชั้นข้อมูล ไม่มีข้อมูลว่าจะต้องส่งแพ็กเก็ตออกผ่านทางอินเทอร์เฟซใด เช่น กรณีที่ไม่รู้ว่า MAC address ของอุปกรณ์เครือข่ายที่ต้องส่งข้อมูลต่อไปคือหมายเลขอะไร ชั้นข้อมูลจะส่งต่อแพ็กเก็ต ไปในชั้นควบคุมตามหมายเลข 4 เพราะกระบวนการเรื่องของ MAC address จะดำเนินการโดยชั้นควบคุม เพื่อตรวจสอบว่าจะต้องส่งต่อแพ็กเก็ต ผ่านไปด้วยเส้นทางใด อินเทอร์เฟซใด จากนั้น เมื่อกระบวนการเสร็จสิ้นชั้นควบคุมจะทำการส่งแพ็กเก็ต C กลับมายังชั้นข้อมูลเพื่อทำการส่งข้อมูลแพ็กเก็ตต่อไปเหมือนกับแพ็กเก็ตอื่นๆ และจะมีการปรับข้อมูลบนตาราง FIB เพื่อกรณีแพ็กเก็ตต่อไปที่มีลักษณะเหมือนแพ็กเก็ตที่ 4 จะสามารถส่งได้ตามปกติไม่ต้องส่งมาที่ชั้นควบคุมเพื่อทำการประมวลผลอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

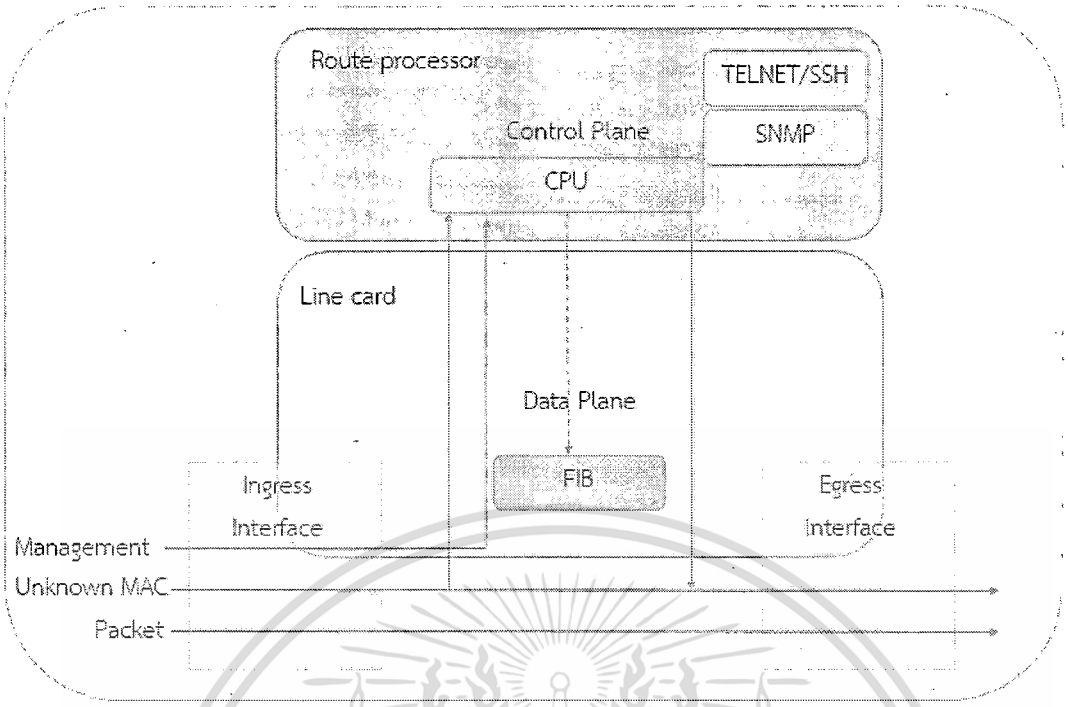


รูปที่ 2.3 ชั้นควบคุม (Control Plane) และชั้นข้อมูล (Data Plane)

(ที่มา : Thomas D. Nadeau, Ken Gray. 2013. SDN: Software Defined Networks. USA: O'Reilly Media. หน้า 13)

โดยแนวคิดในการแยกส่วนกระทำงานระหว่างชั้นควบคุมและชั้นข้อมูลออกจากกัน เป็นแนวคิดที่เกิดมาเป็นเวลานานแล้ว สามารถพิจารณาได้จาก อุปกรณ์เครือข่ายที่มีจำนวน Line Card หลายอันเพื่อแยกช่องทางการเชื่อมต่อ คือ อินเทอร์เฟซ ออกจากส่วนชั้นควบคุม (Route processor) ออกมาเพื่อทำหน้าที่โดยเฉพาะอย่าง และแยกชั้นข้อมูลออกมาเพื่อทำหน้าที่ในการจัดการแพ็กเก็ตที่เข้ามาเท่านั้น กระบวนการทำงานดังกล่าว สามารถแสดงได้ตามรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การแยกชั้นควบคุมและชั้นข้อมูลออกจากกันของ Line Card

2.4 โพรโทคอล OpenFlow

OpenFlow มีแนวคิดเริ่มต้นมาจากการใช้เป็นส่วนหนึ่งของเครือข่ายของการวิจัยที่มหาวิทยาลัย Stanford วัตถุประสงค์ คือ การสร้างโพรโทคอล ในการทดลองบนเครือข่ายมหาวิทยาลัยที่สามารถใช้สำหรับการวิจัยและการทดลอง ก่อนหน้าที่มหาวิทยาลัย จะมีการสร้างแพลตฟอร์มการทดลองของพวกเขาเองตั้งแต่เริ่มต้น วิวัฒนาการมาจากเคอร์เนลนี้โดยมีแนวคิดเริ่มต้นที่เป็นความต้องการให้ สามารถใช้ OpenFlow ทดแทน การทำงานของแบบจำลองโอเอสไอชั้น 2 และชั้น 3 อย่างสมบูรณ์ในเชิงพาณิชย์บนอุปกรณ์เครือข่ายสวิตช์ และเราเตอร์

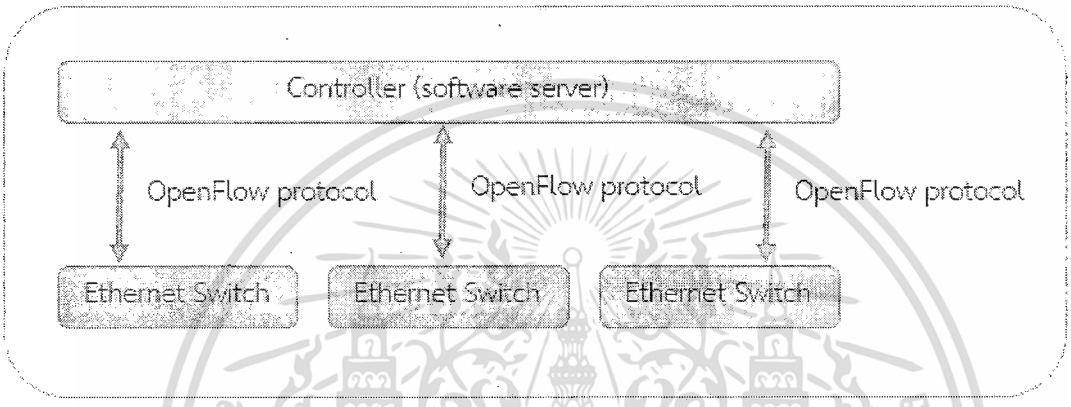
ในปี 2011 สมาคมที่ไม่แสวงหากำไรที่เรียกว่า Open Networking Foundation (ONF) เกิดขึ้นมาจากกลุ่มของผู้ให้บริการ เพื่อผลักดันการใช้ OpenFlow ในเครือข่าย ให้เป็นมาตรฐานชนิดใหม่ และเกิดเป็นมาตรฐานทางการค้า ได้มีการก่อตั้งขึ้นเพื่อทำการวิจัยและพัฒนา OpenFlow เพื่อให้เป็นมาตรฐาน และสามารถนำมาใช้งานในเชิงการค้าได้อย่างมีประสิทธิภาพ

2.4.1 ความรู้พื้นฐานของ OpenFlow

Software-Defined Networking (SDN) มีมาตรฐานกลางคือ OpenFlow เป็นการนิยามทำงานของโพรโทคอลแบบเปิด เพื่อทำให้คอนโทรลเลอร์สามารถมีปฏิสัมพันธ์กับชั้นควบคุมที่เป็นตัวตัดสินใจถึงกระบวนการ วิธีการในการจัดการกับแพ็กเก็ตที่ไหลเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในอินเทอร์เฟซ และสามารถปรับเปลี่ยนเครือข่ายและอุปกรณ์ให้ตรงกับความต้องการ และตอบสนองทางธุรกิจได้ภายใต้การทำงานของ OpenFlow โดยอุปกรณ์เครือข่ายนั้น จะต้องรองรับการทำงานของ OpenFlow ด้วยอินเทอร์เฟซมาตรฐาน ที่ทำให้คอนโทรลเลอร์สามารถทำการเปลี่ยนแปลงข้อมูลในอุปกรณ์ได้ โดยผู้ดูแลระบบสามารถควบคุมการไหลของกระแสจราจรตามความต้องการเพื่อให้เกิดประสิทธิภาพของการใช้งานสูงสุดบนเครือข่ายดังรูปที่ 2.5



รูปที่ 2.5 การใช้โพรโทคอล OpenFlow ระหว่างสวิตช์กับคอนโทรลเลอร์

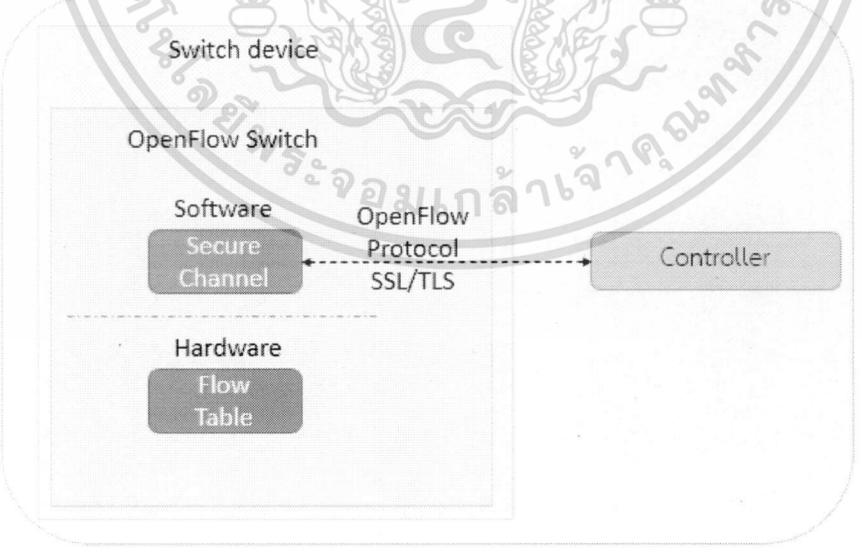
จากการที่ SDN มีการแยกระบบควบคุมที่เคยผูกติดกับระบบฮาร์ดแวร์ ซึ่ง SDN ได้จัดทำซอฟต์แวร์ แอปพลิเคชัน ที่เรียกว่า คอนโทรลเลอร์ขึ้นมาเพื่อเป็นตัวแทนในการควบคุมอุปกรณ์เครือข่าย ด้วยโพรโทคอล OpenFlow เพื่อสื่อสารกับหลากหลายอุปกรณ์บนเครือข่าย ทำให้สามารถบริหารจัดการเครือข่ายได้แบบรวมศูนย์จึงสามารถใช้ทรัพยากรเครือข่ายได้อย่างเต็มประสิทธิภาพและสามารถปรับเปลี่ยนได้เพื่อตอบสนองตามความต้องการของผู้ดูแลระบบ

OpenFlow มีการสื่อสารผ่านช่องสัญญาณที่รักษาความปลอดภัยที่ทำงานภายใต้ TLS (Transport Layer Security) ที่มีการสืบทอดมาจาก SSL (Secure Socket Layer) คอนโทรลเลอร์สามารถใช้งานโพรโทคอล OpenFlow ในการเพิ่ม ปรับเพิ่ม แก้ไข และลบข่าวสารของการไหลของกระแสจราจรในตารางได้ซึ่งมีความปลอดภัยจากการโจรกรรมข้อมูลของบุคคลที่ไม่ประสงค์ดีเพื่อเข้ามาเปลี่ยนแปลงข้อมูลและใส่เงื่อนไขการทำงานใหม่ลงไป โดย OpenFlow มีองค์ประกอบ คือ คอนโทรลเลอร์ในการควบคุมการทำงานช่องสัญญาณที่ปลอดภัยสำหรับการติดต่อสื่อสารกับสวิตช์ ตารางการไหลของข้อมูลที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานแบบ PIPE LINE โดยการทำงานในรูปแบบของ PIPE LINE คือ วิธีการทำงานโดยการส่งต่องานกันเป็นทอดๆ ดังรูปที่ 2.6 และการสื่อสารของ OpenFlow ผ่านช่องสัญญาณที่ปลอดภัย ดังรูปที่ 2.7

รูปที่ 2.6 การทำงานแบบ PIPE LINE ของตารางไหลของข้อมูล



รูปที่ 2.7 การสื่อสารระหว่างคอนโทรลเลอร์กับสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

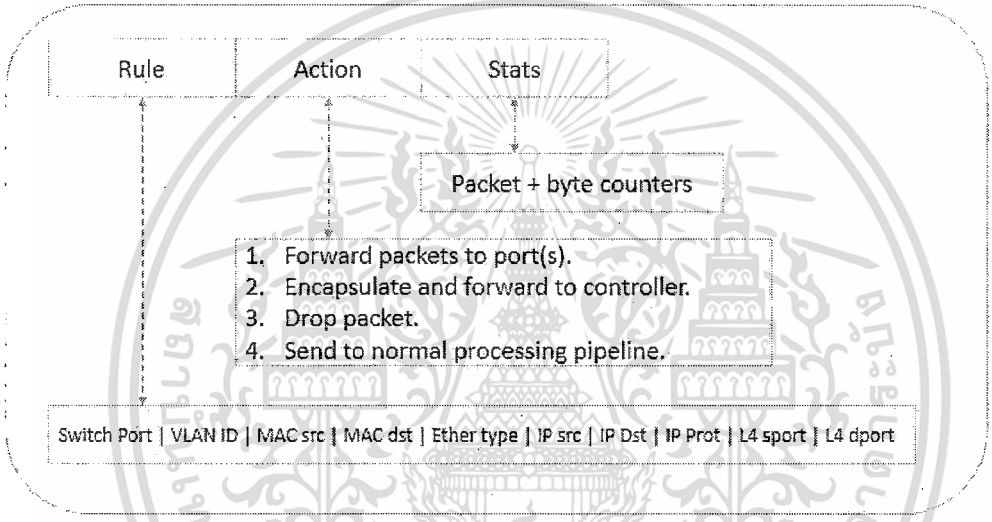
2.4.2 ข้อมูลในตาราง OpenFlow (Flow Table)

ประกอบด้วยการเก็บข้อมูล 3 ช่อง ดังนี้

1) ส่วนหัว หรือ Header ของแพ็กเก็ตที่ระบุการไหลของแพ็กเก็ต เช่น TCP Port 80

2) การกระทำ (Action) ที่ระบุว่าแพ็กเก็ตจะถูกประมวลผลอย่างไร เช่น ส่งออกไปที่ Interface GigbabitEthernet 0/1

3) ข้อมูลสถิติ (Statistics) สำหรับติดตามดูจำนวนของแพ็กเก็ตและปริมาณการใช้งานเป็นไบต์ของแต่ละการไหลของข้อมูล เช่น มีการไหลของข้อมูล 100 แพ็กเก็ต คิดเป็นจำนวน 8000 ไบต์ซึ่งสามารถตั้งค่าไว้ได้ภายในคอนโทรลเลอร์ ดังรูปที่ 2.8



รูปที่ 2.8 ส่วนประกอบภายในตารางข้อมูล OpenFlow (Flow Table)

2.4.3 โพรโทคอล OpenFlow

โพรโทคอล OpenFlow ประกอบไปด้วยข้อมูลข่าวสารที่แลกเปลี่ยนกันระหว่างคอนโทรลเลอร์และสวิตช์ทำงานอยู่บน SSL ในช่องทางที่มีความปลอดภัยและรองรับข่าวสาร 3 ชนิดคือ

1) Controller-to-Switch คอนโทรลเลอร์เป็นผู้เริ่มส่งข่าวสารนี้และมีการตอบกลับเพื่อให้คอนโทรลเลอร์สามารถบริหารจัดการสถานะทางตรรกะของอุปกรณ์ การตั้งค่าและรายละเอียดของข้อมูลภายในตารางการไหลของกระแสจราจร รวมถึงข่าวสารของแพ็กเก็ตที่ไหลออกไปด้วย มีจุดประสงค์เพื่อ

- กำหนด แก้ไข ลบ รายละเอียดของการไหลของกระแสจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ร้องขอข้อมูลที่เกี่ยวกับขีดความสามารถในการทำงานของอุปกรณ์
- ดึงข้อมูลจากอุปกรณ์ เช่น Counter
-ส่งแพ็กเก็ตกลับมาที่อุปกรณ์เพื่อประมวลผลหลังจากมีการสร้างข้อมูลการไหลของกระแสจราจรใหม่

2) Asynchronous ข่าวสารนี้ถูกส่งโดยไม่มีการส่งสัญญาณจากคอนโทรลเลอร์ ประกอบไปด้วยข่าวสารเกี่ยวกับสถานะไปยังคอนโทรลเลอร์ รวมถึงข่าวสารที่แพ็กเก็ตมีการไหลเข้ามายังอุปกรณ์ และเมื่อพบว่าแพ็กเก็ตนั้นผิดปกติจะมีการตรวจสอบเกิดขึ้น โดยจะมีการส่งแพ็กเก็ตไปที่คอนโทรลเลอร์เมื่อมีแพ็กเก็ตที่ไม่เข้ากับการไหลของกระแสจราจรในสภาวะปกติ นอกจากนี้ยังมีการแจ้งข้อมูลให้คอนโทรลเลอร์รู้ว่าการไหลของกระแสจราจรได้ถูกนำออกไป เนื่องจากขอบเขตของ TTL (Time To Live) แสดงถึงการไม่ทำงานในช่วงเวลาหนึ่ง และยังแจ้งให้คอนโทรลเลอร์รู้ว่ามีการเปลี่ยนแปลงเกี่ยวกับสถานการณ์ทำงานของพอร์ตหรือมีความผิดพลาดในการทำงานของอุปกรณ์

3) Symmetric จะถูกส่งโดยอุปกรณ์หรือคอนโทรลเลอร์ เพื่อส่งเป็น Hello Message เพื่อแสดงการคงอยู่ของตัวตนทั้งสองฝ่าย และข่าวสารที่เกี่ยวกับ Echo และ Reply ที่นำมาใช้ตรวจวัดค่าหน่วงเวลา (Delay) ระหว่างอุปกรณ์กับคอนโทรลเลอร์ เพื่อพิสูจน์ว่าอุปกรณ์สามารถทำงานได้ตามปกติ โดยข่าวสารภายในโพรโทคอล OpenFlow แสดงดังตารางที่ 2.1

ตารางที่ 2.1 ข่าวสารภายในโพรโทคอล OpenFlow

Message	Description
	Controller-to-Switch
Features	Request the capabilities of a switch. Switch responds with a features reply that specifies its capabilities.
Configuration	Set and query configuration parameters. Switch responds with parameter settings.
Modify-State	Add, delete, and modify flow/group entries and set switch port properties.
Read-State	Collect information from switch, such as current configuration, statistics, and capabilities.
Packet-out	Direct packet to a specified port on the switch.
Barrier	Barrier request/reply messages are used by the controller to ensure message dependencies have been met or to receive notifications for completed operations.
Role-Request	Set or query role of the OpenFlow channel. Useful when switch connects to multiple controllers.
Asynchronous-Configuration	Set filter on asynchronous messages or query that filter. Useful when switch connects to multiple controllers.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 (ต่อ)

Asynchronous	
Packet-in	Transfer packet to controller.
Flow-Removed	Inform the controller about the removal of a flow entry from a flow table.
Port-Status	Inform the controller of a change on a port.
Error	Notify controller of error or problem condition.
Symmetric	
Hello	Exchanged between the switch and controller upon connection startup.
Echo	Echo request/reply messages can be sent from either the switch or the controller, and they must return an echo reply.
Experimenter	For additional functions.

(ที่มา: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_16-1/161_sdn.html)

2.5 คอนโทรลเลอร์ (Controller)

จากแนวความคิดของ SDN คือ การแยกการทำงาน ระหว่าง Data Plane และ Control Plane ออกจากกัน โดยในการทำงานส่วนของชั้น Data Plane จะถูกทำงานบนอุปกรณ์ ASIC ซึ่งมีอยู่บน LINE CARD ของอุปกรณ์เครือข่ายซึ่งทำหน้าที่เป็นอินเทอร์เฟซในการจัดการกับข้อมูลที่ไหลเข้ามายังอุปกรณ์เครือข่าย และในส่วนของชั้น Control Plane จะเป็นส่วนที่แยกออกมาจากอุปกรณ์เครือข่าย จะทำงานภายนอกโดยจะมีซอฟต์แวร์ทำหน้าที่ในการจัดการชั้น Control Plane จะเรียกว่า คอนโทรลเลอร์ (Controller) โดย จะมีการทำงานในลักษณะแบบรวมศูนย์ คือ มีคอนโทรลเลอร์ที่ศูนย์กลางของระบบเครือข่ายเพียง 1 ระบบ สามารถทำหน้าที่เป็นชั้นควบคุมให้กับอุปกรณ์เครือข่ายได้ทั้งหมดของเครือข่าย การที่คอนโทรลเลอร์จะติดต่อสื่อสารกับอุปกรณ์เครือข่ายนั้น จะมีการติดต่อผ่านช่องทางที่มีความปลอดภัย คือ ใช้งานเชื่อมต่อแบบ SSL/TLS ผ่านเครือข่าย ไอพีเน็ตเวิร์คไปยังอุปกรณ์เครือข่าย SDN ซอฟต์แวร์ที่ทำหน้าที่เป็นคอนโทรลเลอร์มีตัวอย่างดังต่อไปนี้

2.5.1. Mininet

Maninet คือ โปรแกรมที่ทำหน้าจำลองระบบเครือข่ายที่มีการทำงานในรูปแบบของ SDN มีการพัฒนามาจากพื้นฐาน OS-Unix ในส่วนซอฟต์แวร์ Mininet นั้นถูกพัฒนาและได้มีการเผยแพร่แก่สาธารณะเพื่อ ให้บุคคลทั่วไปสามารถนำไปพัฒนาต่อยอดได้ อุปกรณ์เครือข่ายที่ Mininet สามารถ จำลองได้ คือ เร้าเตอร์, สวิตซ์, เครื่องคอมพิวเตอร์ และสามารถจำลองการเชื่อมต่อ ระหว่าง อุปกรณ์เครือข่ายและเครื่องคอมพิวเตอร์ จากรูปที่ 2.9 คือ ตัวอย่างการจำลอง การเชื่อมต่อของอุปกรณ์เครือข่ายและเครื่องคอมพิวเตอร์ในระบบเครือข่าย ด้วย Mininet โดยประกอบไปด้วย เครื่องคอมพิวเตอร์ จำนวน 3 เครื่อง, OpenFlow สวิตซ์ และคอนโทรลเลอร์

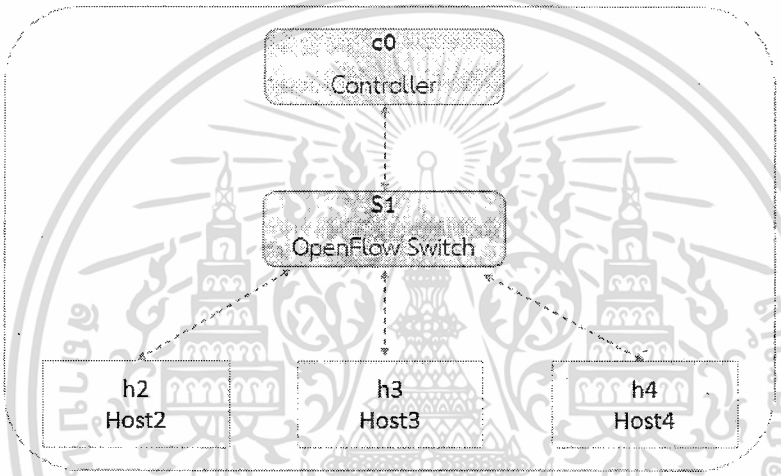
เครื่องคอมพิวเตอร์ ที่แสดงด้วยสัญลักษณ์ h2, h3 และ h4 คือ การจำลองการทำงานของเครื่องคอมพิวเตอร์บนโปรแกรม Mininet โดยสามารถจัดการได้ในระดับเซอร์วิสของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคอมพิวเตอร์ เช่น เซอร์วิสการให้บริการเว็บ รวมทั้งการจัดการเรื่อง ip address ของเครื่องคอมพิวเตอร์ในแต่ละเครื่อง

OpenFlow สวิตซ์ แสดงด้วยสัญลักษณ์ s1 คือ การจำลองอุปกรณ์เครือข่ายสวิตซ์บนโปรแกรม Mininet ที่สามารถรองรับการทำงานในลักษณะ SDN รองรับการเชื่อมต่อกับคอนโทรลเลอร์ จากภายนอกผ่าน OpenFlow โพรโตคอล

คอนโทรลเลอร์ แสดงด้วยสัญลักษณ์ c0 คือ ซอฟต์แวร์ที่ทำหน้าที่เป็น Control Plane ของอุปกรณ์เครือข่ายในระบบเครือข่ายที่ได้จำลองขึ้นมาทดสอบ ที่ทำงานอยู่บนโปรแกรม Mininet



รูปที่ 2.9 ตัวอย่างการจำลองสร้างอุปกรณ์เครือข่ายและการเชื่อมต่อ ด้วยโปรแกรม Mininet

2.5.2. NOX/POX

NOX คือ ซอฟต์แวร์ที่ทำหน้าที่คอนโทรลเลอร์ที่มีการพัฒนามาจากพื้นฐานโครงสร้างภาษา C++ มีการเปิดเผยตัวโปรแกรม (Source Code) ให้แก่สาธารณะ และรองรับการติดต่อสื่อสารด้วย Openflow protocol version 1.0 POX คือ ซอฟต์แวร์ที่ทำหน้าที่คอนโทรลเลอร์โดยเป็นการพัฒนาเพิ่มเติมมาจาก NOX แต่ใช้โครงสร้างเป็นภาษา Python และยังคงรองรับการติดต่อสื่อสารด้วย Openflow protocol version 1.0

2.5.3. Opendaylight

Opendaylight คือ ซอฟต์แวร์ที่ทำหน้าที่คอนโทรลเลอร์ที่มีการพัฒนามาจากพื้นฐานโครงสร้างภาษา JAVA มีการทำงานบนพื้นฐานของระบบปฏิบัติการลินุกซ์ (Linux) รองรับการติดต่อสื่อสารด้วย Openflow protocol version 1.0 และ 1.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4. Cisco OnePK

Cisco OnePK คือ ซอฟต์แวร์ที่ทำหน้าที่คอนโทรลเลอร์ที่ถูกพัฒนาโดยบริษัท Cisco system เพื่อใช้ในด้านอุตสาหกรรมมีการพัฒนามาจากพื้นฐานโครงสร้างภาษา JAVA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

144250

บทที่ 3
การทำงานของระบบปัจจุบัน

เป็นการศึกษาระบบโครงสร้างการทำงานของอุปกรณ์เครือข่ายในปัจจุบัน โดยจะศึกษาถึงกระบวนการทำงานของการประมวลผล (Control Plane) และกระบวนการจัดการต่อแพ็กเก็ตที่เข้ามายังอุปกรณ์เครือข่าย (Data Plane) โดยจะพิจารณาถึงข้อจำกัดบนระบบปัจจุบัน เพื่อใช้เป็นข้อมูลประกอบในการพัฒนาระบบ

3.1 การศึกษาวิเคราะห์การทำงานของระบบปัจจุบัน

อุปกรณ์เครือข่ายในปัจจุบัน ประกอบไปด้วยการทำงาน 2 ส่วนหลัก คือ ส่วนการทำงานของชั้นควบคุมและชั้นข้อมูล โดยจะทำงานร่วมกันบนอุปกรณ์เพียงตัวเดียว ซึ่งการทำงานของอุปกรณ์เครือข่ายแต่ละตัวจะเป็นการทำงานที่แยกอิสระต่อกัน ไม่มีผลเกี่ยวเนื่องของกัน ดังนั้นการจัดการระบบเครือข่ายและการกำหนดยุทธศาสตร์ในลักษณะของภาพรวมจะสามารถจัดการได้ยาก

ตัวอย่างการทำงานของอุปกรณ์เร้าเตอร์ในปัจจุบัน ดังรูปที่ 3.1

1) เมื่อเร้าเตอร์ได้รับแพ็กเก็ตเข้ามาที่อินเทอร์เฟซของเร้าเตอร์

2) เร้าเตอร์ทำการตรวจสอบข้อมูลเลขที่อยู่ไอพีปลายทางที่อยู่ในแพ็กเก็ตและนำไปค้นหาใน Routing table ของตนเอง เพื่อหาเส้นทางในการส่งต่อแพ็กเก็ต

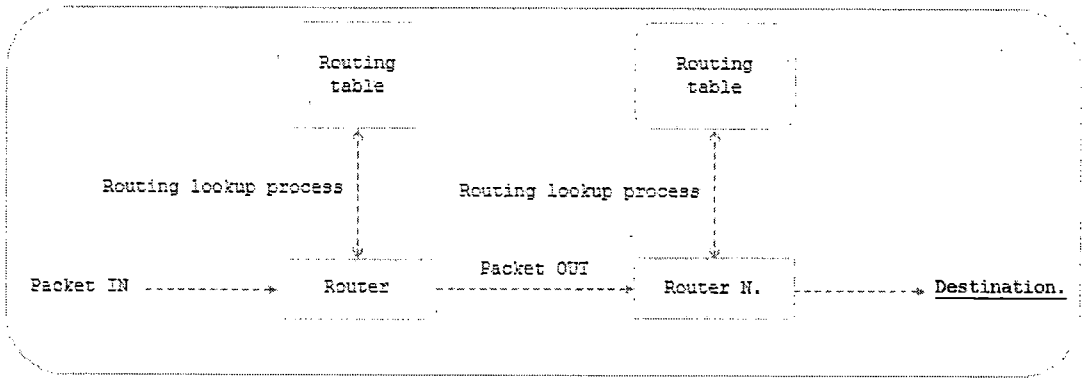
   2.1) ข้อมูลในตารางเร้าติ้งของเร้าเตอร์แต่ละตัว อาจมีข้อมูลที่ไม่เหมือนกันได้

   2.2) การตัดสินใจส่งต่อแพ็กเก็ตเป็นการตัดสินใจของเร้าเตอร์ตัวนั้นๆ ซึ่งได้รับแพ็กเก็ตดังกล่าวมา

3) กรณีเร้าเตอร์ค้นหาข้อมูลไม่พบ จะไม่ทำการส่งต่อแพ็กเก็ตโดยแพ็กเก็ตจะโดนละทิ้ง

4) กรณีเร้าเตอร์ตรวจสอบเส้นทางปลายทางจากข้อมูลตารางเร้าติ้ง พบเส้นทางในการส่งข้อมูลแพ็กเก็ต จะทำการส่งต่อข้อมูลแพ็กเก็ตไปยังอุปกรณ์เครือข่ายตัวถัดไป เพื่อทำการส่งข้อมูลแพ็กเก็ตไปยังปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



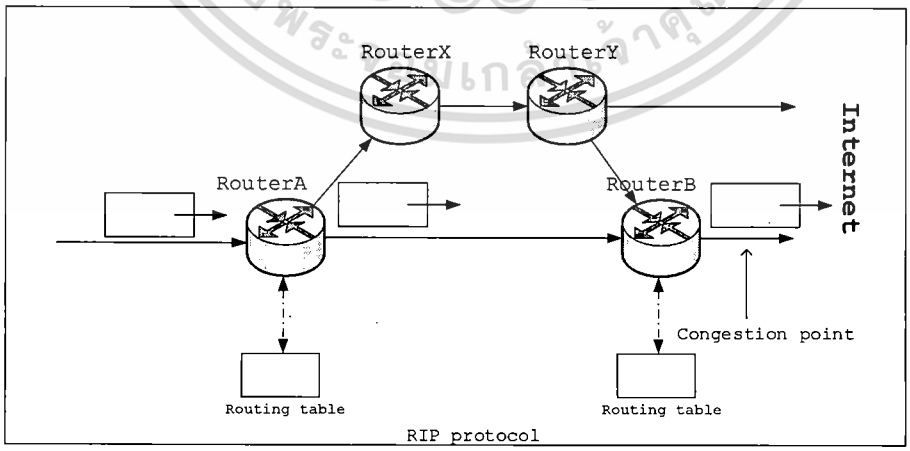
รูปที่ 3.1 ตัวอย่างการทำงานของอุปกรณ์เร้าเตอร์ในปัจจุบัน

3.2 ปัญหาและข้อจำกัดที่พบในระบบปัจจุบัน

จากการทำงานของระบบปัจจุบัน ยังคงมีปัญหาในการจัดการระบบเครือข่ายในภาพรวม การกำหนดยุทธศาสตร์ทำได้ยาก โดยพบปัญหาดังนี้

1) ปัญหาด้านการไม่ทำงานร่วมกันของอุปกรณ์เครือข่าย เนื่องจากอุปกรณ์เครือข่ายจะทำการตัดสินใจและประมวลผลด้วยตนเอง โดยไม่ได้พิจารณาในภาพรวมของระบบเครือข่าย ซึ่งไม่มีความยืดหยุ่นเพียงพอต่อความต้องการในระบบเครือข่ายของผู้ให้บริการอินเทอร์เน็ต

ตัวอย่างบนเร้าเตอร์ B link ที่ออกไปยังอินเทอร์เน็ตมีปริมาณทราฟฟิกหนาแน่นมาก และเมื่อมีแพ็กเก็ตเข้ามาที่เร้าเตอร์ A, เร้าเตอร์ A ทำการตรวจสอบจากตารางเร้าติ้ง พบว่าส่งไปทาง เร้าเตอร์ B ถือว่าเป็นเส้นทางที่ดีที่สุด เนื่องจาก เป็นเส้นทางที่มีระยะทางสั้นที่สุด แต่ว่าเร้าเตอร์ A ไม่รู้เลยว่าเส้นทางออกไปยังอินเทอร์เน็ตบนเร้าเตอร์ B มีปริมาณหนาแน่นมาก ดังรูปที่ 3.2



รูปที่ 3.2 ปัญหาด้านการไม่ทำงานร่วมกันในภาพรวมของอุปกรณ์เครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ปัญหาด้านการจัดการในด้าน การทำงานกับอุปกรณ์เครือข่าย ที่มีความแตกต่างกัน ในด้านการตั้งค่าอุปกรณ์ อันเนื่องมาจากความแตกต่างของผู้ผลิตอุปกรณ์เครือข่าย ซึ่งส่งผลให้เกิดความยุ่งยากในการบริหารจัดการระบบเครือข่ายของผู้ให้บริการอินเทอร์เน็ตที่มีอุปกรณ์เครือข่ายจำนวนมาก ดังรูปที่ 3.3 และ 3.4

```
Cisco system | ios cli pattern
SDN>en
SDN#config t
SDN(config)#ip route 1.2.3.4 255.255.255.255 10.11.11.1
```

รูปที่ 3.3 รูปแบบการจัดการอุปกรณ์เครือข่ายของบริษัท Cisco system

```
Juniper network | junos cli pattern
root@SDN>
root@SDN> configure
set routing-options static route 1.2.3.4/32 next-hop 10.11.11.1
```

รูปที่ 3.4 รูปแบบการจัดการของอุปกรณ์เครือข่ายของบริษัท Juniper

3) ปัญหาต้นทุนราคาอุปกรณ์เครือข่าย ซึ่งปัจจุบันภายในอุปกรณ์เครือข่ายจะมีทั้งการทำงานในส่วนของชั้นควบคุมและชั้นข้อมูลอยู่ด้วยกัน ดังนั้นหากตัดการทำงานในส่วนของชั้นควบคุมออก ก็จะสามารถช่วยลดราคาต้นทุนได้

4) ปัญหาด้ายประสิทธิภาพของอุปกรณ์เครือข่าย ปัจจุบันอุปกรณ์เครือข่ายทำงานในหลากหลายหน้าที่พร้อมกันส่งผลต่อประสิทธิภาพในการจัดการข้อมูลแพ็กเก็ต ทำให้ประสิทธิภาพลดน้อยลง

5) ปัญหาด้านการเลือกเส้นทางของอุปกรณ์เครือข่าย ปัจจุบันการทำงานจะเป็นการพิจารณาเลือกเส้นทางที่ดีที่สุด โดยพิจารณาจากข้อมูลเลขที่อยู่ไอพีปลายทางเท่านั้น โดยไม่ได้พิจารณาจากข้อมูลของเลขที่อยู่ไอพีต้นทาง ส่งผลให้ความสามารถในการจัดการในระบบเครือข่ายของ ผู้ให้บริการอินเทอร์เน็ตไม่คล่องตัวในการจัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การให้บริการของ ผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider)

ผู้ให้บริการอินเทอร์เน็ต หรือ เรียกว่า ISP คือ ผู้ที่ให้บริการด้านการเชื่อมต่อเพื่อให้เกิดการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ หรือเครือข่ายคอมพิวเตอร์กับระบบเครือข่ายอินเทอร์เน็ตทั่วโลก โดยผู้ให้บริการอินเทอร์เน็ตจะมีการสร้างระบบเครือข่ายขึ้นมารองรับการเชื่อมต่อระหว่างลูกค้าผู้ใช้งานเข้ากับระบบเครือข่ายของอินเทอร์เน็ตทั่วโลก ตัวอย่างกลุ่มลูกค้าผู้ใช้งาน เช่น

1) บรอดแบนด์อินเทอร์เน็ต (Broadband Internet) คือ อินเทอร์เน็ต สำหรับกลุ่มคนทั่วไป ที่มีการใช้งานตามบ้านหรือสถานที่ของตนเอง มีการเชื่อมต่อมายังผู้ให้บริการอินเทอร์เน็ต ในลักษณะ คู่สายสายทองแดง, สายเคเบิ้ล หรือ สายไฟเบอร์ ทั้งนี้ขึ้นอยู่กับเทคโนโลยีของผู้ให้บริการที่จะมีรองรับในการเชื่อมต่อ โดยอินเทอร์เน็ตประเภทนี้ จะมีการแบ่งปันแบนด์วิดท์กันในการใช้งาน ขึ้นอยู่กับเงื่อนไขที่ทางผู้ให้บริการอินเทอร์เน็ต เป็นผู้กำหนด

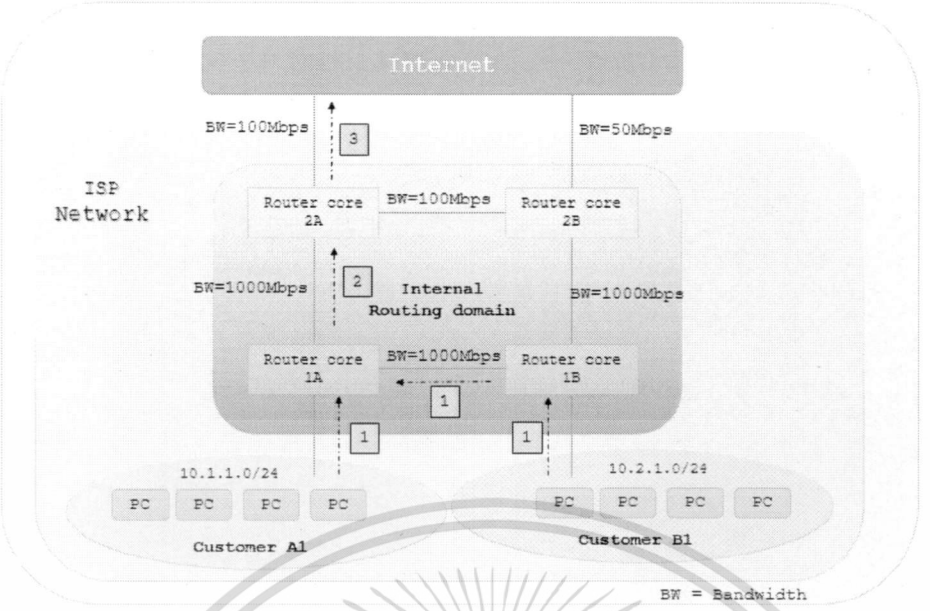
2) อินเทอร์เน็ตวงจรเช่า ส่วนบุคคล (Leased Line Internet) รูปแบบการทำงานเหมือนกับบรอดแบนด์อินเทอร์เน็ต โดยมีสิ่งที่แตกต่างกัน คือ เป็นการจัดสรรค์แบนด์วิดท์ไว้เฉพาะส่วนบุคคลไม่มีการแบ่งปันหรือมีคนร่วมใช้งาน โดยทั่วไปจะเหมาะกับองค์กรที่เป็นบริษัท มหาวิทยาลัย หรือ สถาบันต่างๆ

3) ศูนย์เครือข่ายกลางให้บริการข้อมูล อินเทอร์เน็ต (Internet Data Center) หรือ เรียกย่อว่า IDC คือ การให้บริการด้านอินเทอร์เน็ต โดยผู้ให้บริการอินเทอร์เน็ตจะมีการจัดระบบโครงสร้างพื้นฐานเพื่อเชื่อมต่ออินเทอร์เน็ตจัดเตรียมไว้ เช่น ระบบการเชื่อมต่ออินเทอร์เน็ตด้วยความเร็วสูง ระบบปรับอากาศ ระบบไฟฟ้าหลัก และ ระบบไฟฟ้าสำรองระบบรักษาความปลอดภัย รวมทั้งบริการด้านคำปรึกษาจากผู้เชี่ยวชาญด้านอินเทอร์เน็ต โดยผู้ใช้งานส่วนใหญ่จะเป็นลักษณะการนำอุปกรณ์คอมพิวเตอร์หรืออุปกรณ์เซิร์ฟเวอร์ไปติดตั้งที่ผู้ให้บริการอินเทอร์เน็ต โดยทั่วไปจะเหมาะกับผู้ที่ต้องการความมั่นคงทางด้านอินเทอร์เน็ตสูง

ทั้งนี้รูปแบบการเชื่อมต่อจากผู้ให้บริการ ไปยังผู้ใช้บริการจะมีความหลายหลาย โดยจะขึ้นอยู่กับเทคโนโลยีที่รองรับ โดยเมื่อผู้ใช้บริการมีการเชื่อมต่อเข้ามาที่เครือข่ายของผู้ให้บริการ ผู้ให้บริการจะจัดเตรียมระบบเครือข่ายพื้นฐานเพื่อให้เกิดการเชื่อมต่อระหว่างผู้ใช้บริการและระบบเครือข่ายอินเทอร์เน็ตทั่วโลก

ปัจจุบันระบบเครือข่ายพื้นฐานของผู้ให้บริการอินเทอร์เน็ต ใช้เทคโนโลยีที่เรียกว่า Packet switching โดยมีการส่งข้อมูลแพ็กเก็ต โดยการพิจารณาจากข้อมูลเลขที่อยู่ไอพีปลายทางเท่านั้น ในปัจจุบันอุปกรณ์ ที่ทำหน้าที่ในการส่งข้อมูลแพ็กเก็ตก็คือเร้าเตอร์หรือ Switch layer 3 โดยในอุปกรณ์ดังกล่าวทุกตัวจะมีข้อมูล ตารางเร้าติ้ง (Routing table) ไปยังระบบเครือข่ายอินเทอร์เน็ตทั่วโลก ดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ตัวอย่างการทำงานของระบบเครือข่ายพื้นฐานของผู้ให้บริการอินเทอร์เน็ตปัจจุบัน

ลำดับการทำงาน

1) เมื่อเร้าเตอร์ 1A และ 1B ได้รับข้อมูลแพ็กเก็ตเข้ามา จะทำการตรวจสอบข้อมูล IP address ปลายทาง จากตารางเร้าติ้งของตนเอง จากนั้นทำการส่งต่อข้อมูลแพ็กเก็ตต่อไปยังอุปกรณ์เร้าเตอร์ถัดไป โดยเร้าเตอร์ core-1A ตัดสินใจส่งต่อไปยังเร้าเตอร์ core-2A เนื่องจากเป็นระยะทางที่สั้นที่สุด และเป็นช่องทางที่มีค่าปริมาณช่องทางในการสื่อสารสูงที่สุด ในส่วนเร้าเตอร์ core-1B ตัดสินใจส่งต่อไปยังเร้าเตอร์ core-1A เนื่องจากพิจารณาค่าปริมาณช่องทางในการสื่อสารไปยังอินเทอร์เน็ต และค่าปริมาณช่องทางในการสื่อสารระหว่างเร้าเตอร์ core 2A และ core 2B มีปริมาณน้อยกว่า ระหว่างเร้าเตอร์ core 1A และ core 2A ทำให้เร้าเตอร์ core 1B ส่งต่อข้อมูลแพ็กเก็ตต่อไปยังเร้าเตอร์ core 1A ดังนั้น กรณีการส่งต่อข้อมูลแพ็กเก็ตไปยังเครือข่ายอินเทอร์เน็ต จะผ่านเร้าเตอร์ core-1A ทั้งหมด

2) เร้าเตอร์ core-1A จะส่งต่อข้อมูลแพ็กเกต ไปยังเร้าเตอร์ core-2A

3) เมื่อเร้าเตอร์ core-2A ได้รับข้อมูลแพ็กเกตเข้ามา จะทำการตรวจสอบข้อมูล IP address ปลายทาง จากตารางเร้าติ้ง จากนั้นทำการส่งต่อข้อมูลแพ็กเก็ตต่อไปยังอุปกรณ์ถัดไป คือ อุปกรณ์เครือข่ายที่เชื่อมต่อกับอินเทอร์เน็ต

ปัญหาที่พบ

ระบบเครือข่ายไม่มีความยืดหยุ่น อันเนื่องมาจาก ในการตัดสินใจเลือกเส้นทางไม่สามารถนำข้อมูล IP address ต้นทางมา ใช้ในการพิจารณาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การวิเคราะห์และออกแบบระบบงาน

4.1 องค์ประกอบของระบบเครือข่าย Software-Defined Networking (SDN)

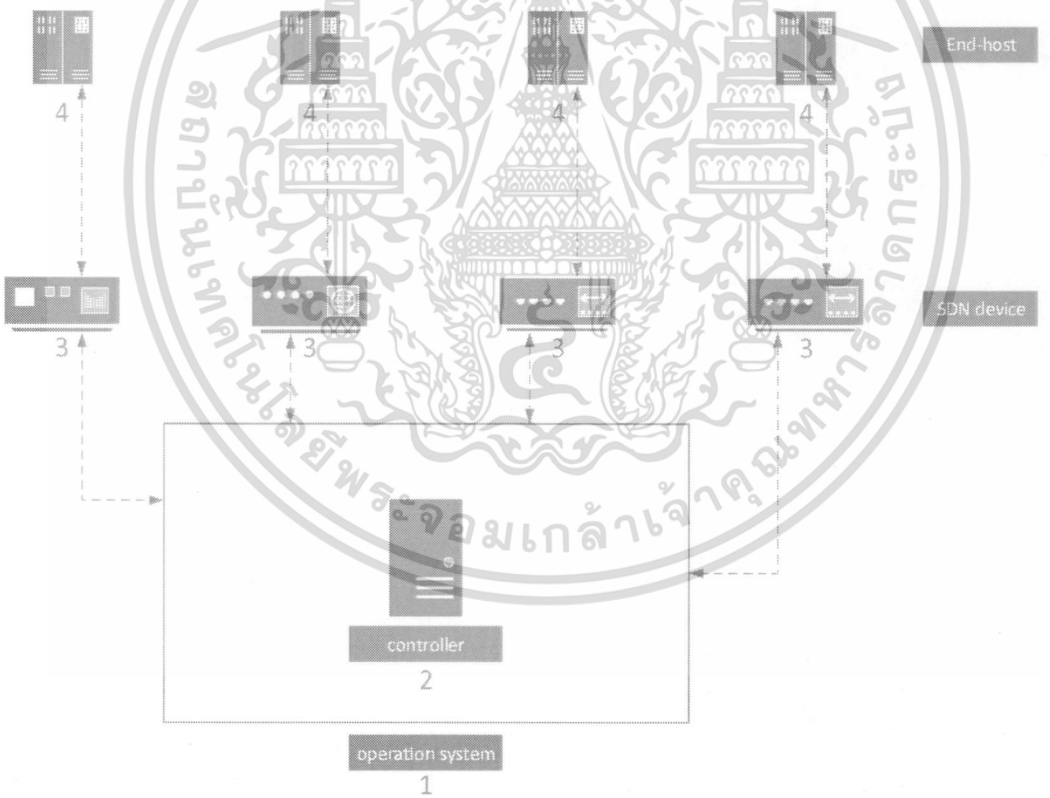
ระบบ Software-Defined Networking (SDN) มีการแยกการทำงาน ระหว่าง Control plan และ Data plan จึงสามารถแบ่งกลุ่มขององค์ประกอบได้ดังนี้ ดังรูปที่ 4.1

1) ระบบปฏิบัติการ (Operating System) พื้นฐานเพื่อรองรับการทำงานของซอฟต์แวร์คอนโทรลเลอร์ (Controller)

2) ซอฟต์แวร์ที่ทำหน้าที่เป็นคอนโทรลเลอร์ (Controller)

3) อุปกรณ์เครือข่ายที่รองรับการทำงานในรูปแบบ SDN ผ่านโพรโทคอล Openflow

4) เครื่อง (Host)



รูปที่ 4.1 องค์ประกอบของระบบเครือข่าย Software-Defined Networking (SDN)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทำงานของระบบเครือข่าย Software-Defined Networking (SDN)

ระบบเครือข่ายที่ใช้งานบนอุปกรณ์เครือข่ายมีการทำงานแยกกัน ระหว่าง Control plane และ Data plane โดยอุปกรณ์คอนโทรลเลอร์จะทำหน้าที่เป็น Control plane ทำหน้าที่บริหารจัดการเงื่อนไขในการส่งต่อข้อมูลแพ็กเก็ต และสั่งงานไปยังอุปกรณ์เครือข่ายทั้งหมด ดังรูปที่ 4.2 โดยมีการงานดังนี้

1) คอนโทรลเลอร์จะสร้างการเชื่อมต่อไปยังอุปกรณ์เครือข่ายในโครงข่าย ผ่านช่องทางที่ปลอดภัย เช่น SSL/TLS

2) คอนโทรลเลอร์จะมีการยืนยันตัวตน (Authentication) กับอุปกรณ์เครือข่ายด้วยชื่อผู้ใช้และรหัสผ่าน

3) คอนโทรลเลอร์ จะทำหน้าที่ประมวลผลในส่วนของ Control Plane ให้กับอุปกรณ์เครือข่ายในโครงข่าย

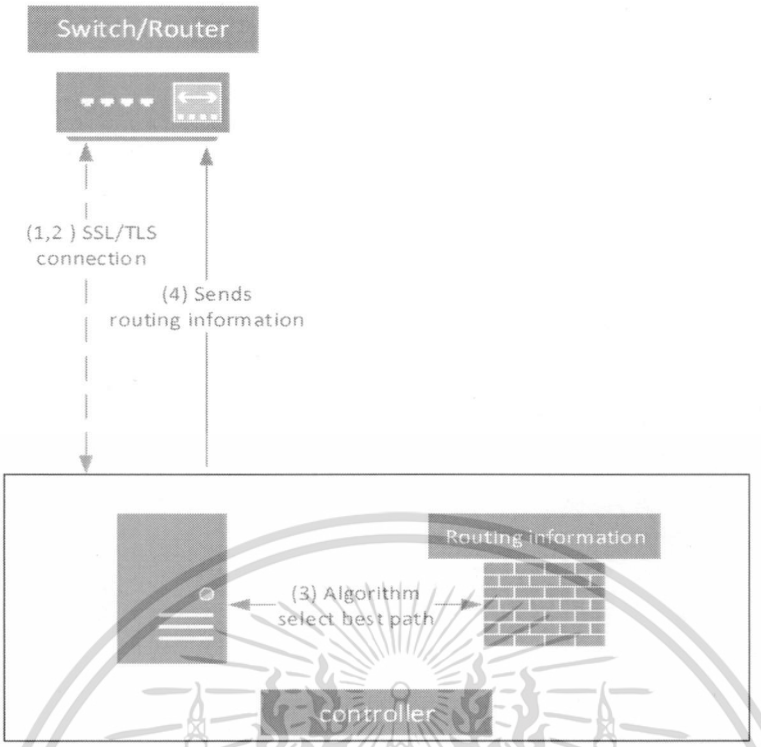
3.1) ทำหน้าที่ในการหาเส้นที่ดีที่สุด ในการส่งต่อแพ็กเก็ตจากต้นทางไปยังปลายทางพิจารณาจากเงื่อนไข IP address ปลายทาง ในการตัดสินเลือกเส้นทาง

3.2) ทำหน้าที่ในการหาเส้นที่ดีที่สุด ในการส่งต่อแพ็กเก็ตบนเงื่อนไข IP address ต้นทาง และ IP address ปลายทาง

3.3) ทำหน้าที่ในการจัดการตั้งค่าคอนฟิกูเรชันให้กับอุปกรณ์ในเครือข่าย จากศูนย์กลางเพื่อลดความผิดพลาดในการตั้งค่าเมื่ออุปกรณ์เครือข่ายมีจำนวนมาก

4) คอนโทรลเลอร์ส่งคำสั่ง และข้อมูลต่างๆ ที่จำเป็นต่อการทำงานไปที่อุปกรณ์เครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การทำงานของระบบเครือข่าย Software-Defined Networking (SDN)

4.3 การออกแบบระบบเครือข่ายผู้ให้บริการอินเทอร์เน็ต ด้วยเทคโนโลยี SDN

การออกแบบจะเป็นการนำหลักการของ SDN มาศึกษาและทดสอบบนระบบเครือข่ายจำลอง จากนั้นนำมาปรับปรุงและประยุกต์ใช้กับเครือข่ายของผู้ให้บริการเครือข่ายอินเทอร์เน็ต โดยจะมีการติดตั้งซอฟต์แวร์คอนโทรลเลอร์เพื่อทำการจัดการบริหาร อุปกรณ์ในเครือข่าย และมีการจัดการข้อมูลในระบบเครือข่ายด้วยวิธี Flow base คือ ใช้เงื่อนไข IP address ต้นทาง และ IP address ปลายทาง มาเป็นเงื่อนไขใช้ในการพิจารณาตัดสินใจ เพื่อทำการส่งต่อ ข้อมูลแพ็กเก็ตแทนที่การส่งข้อมูลด้วยรูปแบบ Packet switching ที่จะใช้เงื่อนไข IP address ปลายทางเท่านั้น ในการพิจารณาส่งต่อข้อมูลแพ็กเก็ต ทั้งนี้เพื่อให้ระบบเครือข่ายมีความยืดหยุ่นเพิ่มขึ้น เพิ่มประสิทธิภาพในการจัดสรรแบนด์วิดท์ได้เพิ่มขึ้น

การทำงานของคอนโทรลเลอร์จะสั่งงานไปยังอุปกรณ์เครือข่ายโดยใช้เงื่อนไขเป็นลักษณะ Flow base โดยพิจารณาจากตัวแปล รายละเอียดดังนี้

1) IP address หรือ กลุ่มของ IP address ต้นทาง หรือ Interface ต้นทาง

2) IP address หรือ กลุ่มของ IP address ปลายทาง

3) Action

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การเชื่อมต่อระบบเครือข่ายผู้ให้บริการอินเทอร์เน็ตดาต้าเซ็นเตอร์ ด้วยเทคโนโลยี SDN

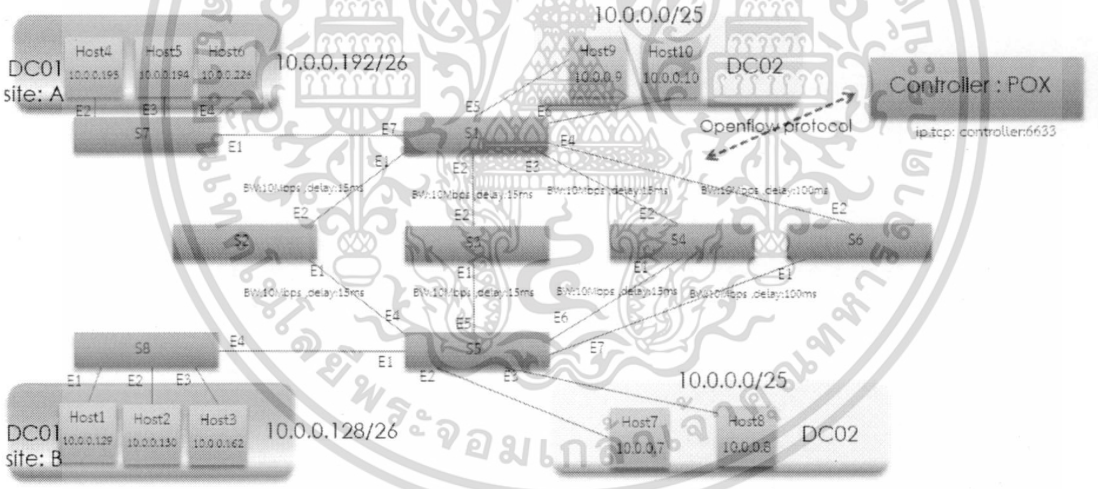
เทคโนโลยี SDN ของระบบเครือข่ายของผู้ให้บริการอินเทอร์เน็ตดาต้าเซ็นเตอร์ ประกอบด้วยรายละเอียดดังต่อไปนี้

1) IP address ที่ถูกกำหนดให้ใช้งานทั้งเครือข่ายดาต้าเซ็นเตอร์ คือ 10.0.0.0/24

2) ในระบบเครือข่ายจะประกอบไปด้วยดาต้าเซ็นเตอร์ย่อยจำนวน 2 แห่ง คือ ดาต้าเซ็นเตอร์แห่งที่ 1 (DC1) และ ดาต้าเซ็นเตอร์แห่งที่ 2 (DC2)

3) ดาต้าเซ็นเตอร์แห่งที่ 1 จะกำหนดให้ใช้ IP address ในช่วงของ 10.0.0.128/25 และยังคงใช้ Subnet เดิมตามกำหนดข้อที่ 1 และจะแบ่งย่อยออกเป็น 2 สถานที่ คือ Site: A และ Site: B ซึ่งทั้งสองแห่งตั้งอยู่ในสถานที่ตั้งคนละที่กัน

4) ดาต้าเซ็นเตอร์แห่งที่ 2 จะกำหนดให้ใช้ IP address ในช่วงของ 10.0.0.0/25 และยังคงใช้ Subnet เดิมตามกำหนดข้อที่ 1 และจะแบ่งย่อยออกเป็น 2 สถานที่ คือ Site: A และ Site: B ซึ่งทั้งสองแห่งตั้งอยู่ในสถานที่ตั้งคนละที่กัน ดังรูปที่ 4.3



รูปที่ 4.3 การเชื่อมต่อระบบเครือข่ายผู้ให้บริการอินเทอร์เน็ตดาต้าเซ็นเตอร์ด้วยเทคโนโลยี SDN

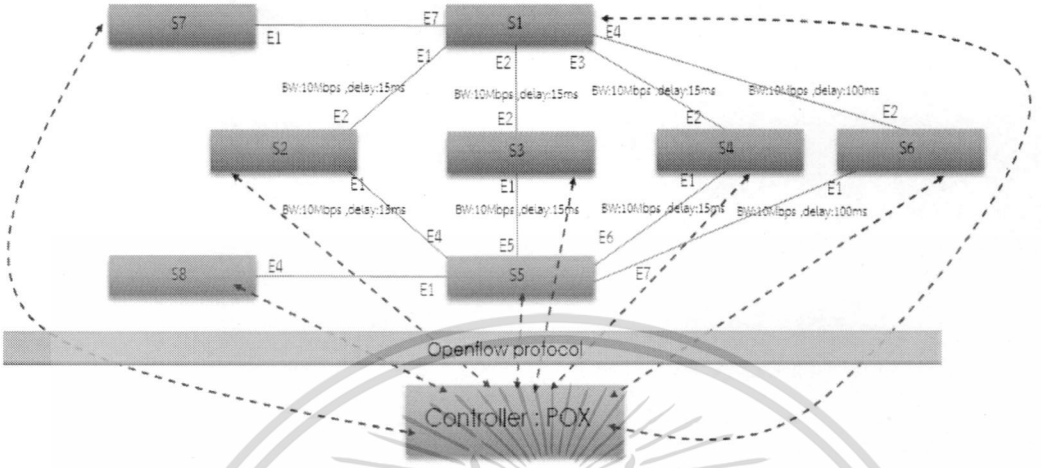
4.5 การทำงานของระบบเครือข่ายด้วยเทคโนโลยี SDN

4.5.1 การบริหารจัดการระบบเครือข่ายจากศูนย์กลาง

ระบบเครือข่าย SDN จะถูกบริหารจัดการจากศูนย์กลาง ด้วยซอฟต์แวร์คอนโทรลเลอร์ซึ่งมีความสามารถในการบริหารจัดการการส่งข้อข้อมูลบนเครือข่ายโดยไม่ติดข้อจำกัด จากการติดต่อสื่อสารในรูปแบบโพรโทคอล Openflow ซึ่งเป็นมาตรฐาน ทั้งในกรณีอุปกรณ์มีความแตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันของผู้ผลิตรูปแบบของคำสั่งในการสั่งงาน รูปแบบการตรวจสอบค่าในระบบต่างๆ บนอุปกรณ์ ดังรูปที่ 4.4



รูปที่ 4.4 การเชื่อมต่อระหว่างอุปกรณ์ SDN และคอนโทรลเลอร์ด้วย Openflow โพรโทคอล

4.5.2 การบริหารจัดการระบบเครือข่ายอินเทอร์เน็ตดาต้าเซ็นเตอร์ด้วยการจัดการทราฟฟิกในรูปแบบเงื่อนไข IP address บนอุปกรณ์ SDN สวิตช์

บนระบบเครือข่ายมีการกำหนดโครงสร้างของ IP address ที่จัดสรรเป็น Subnet 10.0.0.0/24 คือ สามารถมี จำนวน Host ได้จำนวน 254 เครื่อง โดยมีการจัดแบ่งการใช้งานเป็นดังนี้ ดังรูปที่ 4.5

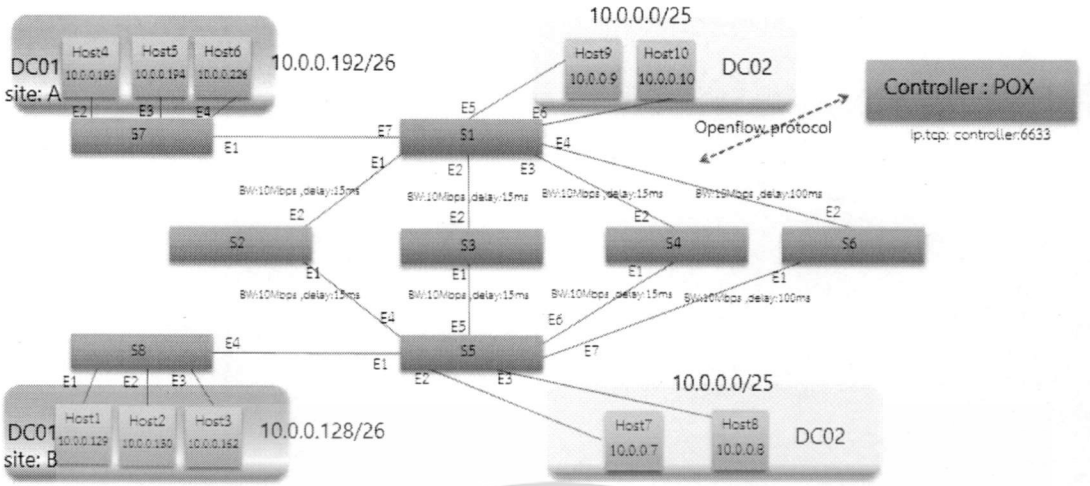
1) ดาต้าเซ็นเตอร์แห่งที่ 1 = 10.0.0.128/25

1.1) ดาต้าเซ็นเตอร์แห่งที่ 1 site A = 10.0.0.192/26

1.2) ดาต้าเซ็นเตอร์แห่งที่ 1 site A = 10.0.0.128/26

2) ดาต้าเซ็นเตอร์แห่งที่ 2 = 10.0.0.0/25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การจัดสรร IP Address ในเครือข่ายผู้ให้บริการอินเทอร์เน็ตดาต้าเซ็นเตอร์

โดยมีการกำหนดค่าการส่งทราฟฟิกเป็นดังนี้

1) การไหลของทราฟฟิกระหว่างดาต้าเซ็นเตอร์แห่งที่ 1 จะใช้งานผ่าน 2 ช่องทาง คือ

ช่องทางที่ 1 : Swtch.8 < > Switch.5 < > Switch.2 < > Switch.1 < > Switch.7

ช่องทางที่ 2 : Swtch.8 < > Switch.5 < > Switch.3 < > Switch.1 < > Switch.7

2) กำหนดเงื่อนไขการส่งข้อมูลผ่านระบบเครือข่าย SDN ด้วยรูปแบบของ IP address บนการติดต่อสื่อสารภายในเครือข่ายดาต้าเซ็นเตอร์ แห่งที่ 1 ดังนี้

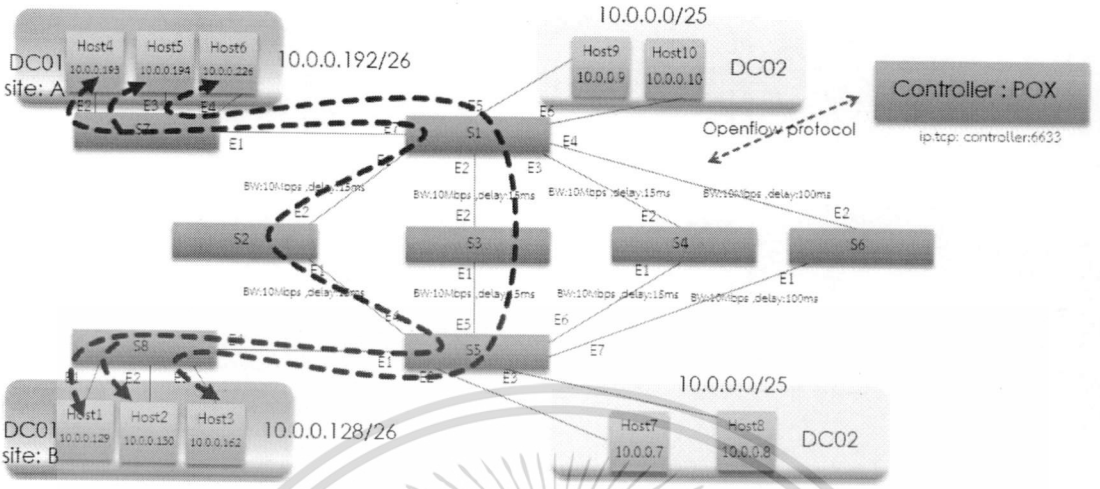
เงื่อนไขที่ 1 : การติดต่อ สื่อสารระหว่าง IP subnet = 10.0.0.128/27 และ 10.0.0.192/27 ให้ผ่านทางช่องทาง Switch.2

เงื่อนไขที่ 2 : การติดต่อ สื่อสารระหว่าง IP subnet = 10.0.0.224/27 และ 10.0.0.160/27 ให้ผ่านทางช่องทาง Switch.3

จากเงื่อนไขทั้งหมดจะสามารถแสดงเป็นรูป ดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งนี้เพื่อเป็นการบริหารจัดการ ระบบเครือข่ายให้มีการใช้งานได้อย่างมีประสิทธิภาพ



รูปที่ 4.6 การไหลของทราฟฟิกของดาต้าเซ็นเตอร์ แห่งที่ 1

4.5.3. การตรวจสอบสถานะใช้งานทราฟฟิกของระบบเครือข่ายอินเทอร์เน็ตดาต้าเซ็นเตอร์

โดยมีการตรวจสอบสถานะการใช้งานทราฟฟิกของพอร์ตเพื่อป้องกันปัญหาการคับคั่งของทราฟฟิกในระบบเครือข่าย

ระบบเครือข่ายของ ดาต้าเซ็นเตอร์แห่งที่ 2 จะมีการส่งทราฟฟิกระหว่างดาต้าเซ็นเตอร์ผ่านเส้นทางสวิตซ์.4 เป็นหลักและมีเส้นทางผ่านสวิตซ์6 เป็นช่องทางสำรองในกรณีที่มีการใช้งานทราฟฟิกในปริมาณมาก โดยมีการกำหนด ค่าการส่งทราฟฟิกในสถานการณ์ปกติ เป็นดังนี้

เส้นทางการส่งทราฟฟิก Swtch.5 < > Switch.4 < > Switch.1 ดังรูปที่ 4.7

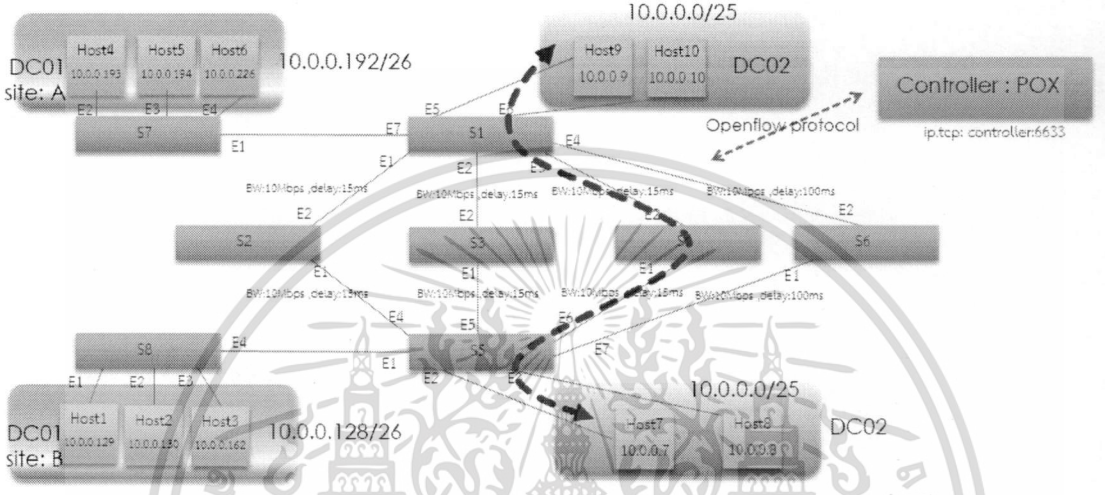
จากการที่ระบบการจัดการเครือข่ายถูกจัดการโดยคอนโทรลเลอร์จากศูนย์กลาง ดังนั้นในการจัดการทราฟฟิกของดาต้าเซ็นเตอร์ แห่งที่ 2 จะมีการตรวจสอบปริมาณการใช้งานระหว่าง ดาต้าเซ็นเตอรแห่งที่ 2 เพื่อที่จะป้องกัน การเกิดการคับคั่งของปริมาณทราฟฟิก โดยกำหนดเงื่อนไขที่ คอนโทรลเลอร์เพื่อทำการตรวจสอบ ดังนี้

1) คอนโทรลเลอร์จะทำการตรวจสอบปริมาณทราฟฟิกของพอร์ต uplink ระหว่างดาต้าเซ็นเตอร์แห่งที่ 2 (ผ่าน สวิตซ์4)

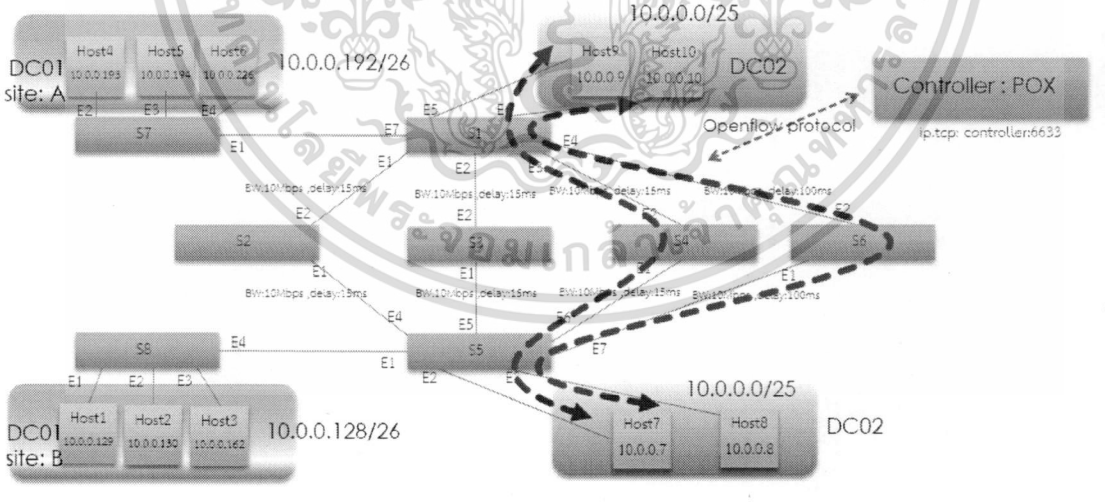
2) หากตรวจสอบพบว่าปริมาณการใช้งานของพอร์ต uplink มีปริมาณมากกว่า 80% ของขนาด uplink จะมีการสั่งงานจากคอนโทรลเลอร์ให้มีการแบ่งทราฟฟิกบางส่วนออกไปจาก uplink เดิม เพื่อเป็นการลดความคับคั่งของพอร์ต uplink ลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) คอนโทรลเลอร์สั่งงานไปยัง SDN สวิตซ์เพื่อทำการแบ่งทราฟฟิกออกจากพอร์ต uplink เดิม โดยมีเงื่อนไขการแบ่งปริมาณของทราฟฟิกด้วย IP address ต้นทางและปลายทาง คือ การติดต่อสื่อสารระหว่างคู่ IP address ของ Host7 และ Host9 จะมีการกำหนดให้ส่งทราฟฟิกผ่าน Switch4, การติดต่อสื่อสารระหว่างคู่ IP address ของ Host8 และ Host10 จะมีการกำหนดให้ส่งทราฟฟิกผ่าน Switch6 ดังรูปที่ 4.8



รูปที่ 4.7 การไหลของทราฟฟิกของดาต้าเซ็นเตอร์ แห่งที่ 2 กรณีมีการใช้งาน Uplink น้อยกว่า 80%



รูปที่ 4.8 การส่งทราฟฟิกของดาต้าเซ็นเตอร์แห่งที่ 2 กรณีมีการใช้งาน Uplink มากกว่า 80%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทำงาน และการให้บริการของระบบ SDN

จากการศึกษาและออกแบบระบบการทำงานของ SDN จากบทที่ 4 จึงได้มีการสร้างระบบเครือข่ายจำลอง ที่อุปกรณ์เครือข่ายรองรับการทำงานในรูปแบบ SDN เพื่อทำการจำลอง การทำงาน ทดลองการทำงานของระบบเครือข่ายที่มีการบริหารจัดการด้วยระบบซอฟต์แวร์ เพื่ออธิบายการทำงาน ดังต่อไปนี้

5.1 สร้างเครือข่าย SDN

เป็นการสร้างระบบเครือข่ายที่มีความสามารถรองรับการทำงานในรูปแบบของ SDN ผ่าน Openflow โพรโทคอลโดยสร้างระบบเครือข่ายจำลองด้วยโปรแกรม Mininet โดยมีองค์ประกอบของอุปกรณ์ ดังนี้ SDN สวิตซ์, เครื่องคอมพิวเตอร์ และการเชื่อมต่อในรูปแบบต่างๆ

5.1.1. สร้างเครือข่ายบน Mininet

ขั้นตอนการสร้างเครือข่ายบน Mininet ดังรูปที่ 5.1

```
class MyTopo(Topo):
    "Single switch connected to n hosts."
    def __init__(self):
        Topo.__init__(self)
        s1=self.addSwitch('s1',listenPort=6681)
        s2=self.addSwitch('s2',listenPort=6682)
        s3=self.addSwitch('s3',listenPort=6683)
        s4=self.addSwitch('s4',listenPort=6684)
        s5=self.addSwitch('s5',listenPort=6685)
        s6=self.addSwitch('s6',listenPort=6686)
        s7=self.addSwitch('s7',listenPort=6687)
        s8=self.addSwitch('s8',listenPort=6688)
        h1=self.addHost('h1')
        h2=self.addHost('h2')
        h3=self.addHost('h3')
        h4=self.addHost('h4')
        h5=self.addHost('h5')
        h6=self.addHost('h6')
        h7=self.addHost('h7')
        h8=self.addHost('h8')
        h9=self.addHost('h9')
        h10=self.addHost('h10')
        self.addLink(h1, s8, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(h2, s8, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(h3, s8, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s8, s5, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(h7, s5, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(h8, s5, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s5, s2, bw=10, delay='15ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s5, s3, bw=10, delay='200ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s5, s4, bw=10, delay='15ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s5, s6, bw=10, delay='100ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s2, s1, bw=10, delay='15ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s3, s1, bw=10, delay='15ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s4, s1, bw=10, delay='15ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s6, s1, bw=1, delay='100ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s1, h9, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s1, h10, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s1, s7, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s7, h4, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s7, h5, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
        self.addLink(s7, h6, bw=100, delay='1ms', loss=0, max_queue_size=1000, use_htb=True)
```

รูปที่ 5.1 การกำหนดค่า เพื่อสร้างเครือข่าย SDN ด้วย Mininet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2. เริ่มการทำงานของระบบเครือข่าย SDN ด้วย Mininet

จากขั้นตอนการสร้างระบบเครือข่าย ด้วย Mininet เสร็จสิ้นแล้ว จะเป็นการนำ code ดังกล่าวมาเริ่มทำงานเพื่อสร้างระบบเครือข่าย SDN ดังแสดงรายละเอียดได้ดังรูปที่ 5.2 จะเป็นการแสดงให้เห็นถึงกระบวนการสร้างระบบเครือข่ายด้วย Mininet และเมื่อสร้างระบบเครือข่ายเสร็จ Mininet จะขึ้นหน้าจอเพื่อรอรับคำสั่งต่อไป

```
mininet@mininet-vm:~/hat$ sudo ./is2-mininet.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8
*** Adding links:
(100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) (h1, s8) (100.00M
oss) (100.00Mbit 1ms delay 0% loss) (h3, s8) (100.00Mbit 1ms delay 0% loss) (100
 0% loss) (h5, s7) (100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss)
Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) (h8, s5) (100.00Mbit 1ms
00.00Mbit 1ms delay 0% loss) (h10, s1) (10.00Mbit 15ms delay 0% loss) (10.00Mbit
ss) (s1, s3) (10.00Mbit 15ms delay 0% loss) (10.00Mbit 15ms delay 0% loss) (s1,
ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) (s1, s7) (10.00Mbit 15ms delay
bit 200ms delay 0% loss) (s3, s5) (10.00Mbit 15ms delay 0% loss) (10.00Mbit 15ms
 (s5, s6) (100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) (s5, s8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controller
*** Starting 8 switches
s1 (10.00Mbit 15ms delay 0% loss) (10.00Mbit 15ms delay 0% loss) (10.00Mbit 15ms delay 0
t 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) s2 (10.00Mbit 15ms delay 0% loss) (1
delay 0% loss) s4 (10.00Mbit 15ms delay 0% loss) (10.00Mbit 15ms delay 0% loss) s5 (100.
% loss) (10.00Mbit 15ms delay 0% loss) (10.00Mbit 200ms delay 0% loss) (10.00Mbit 15ms d
 (1.00Mbit 100ms delay 0% loss) s7 (100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay
0Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) (
Dumping host connections
h1 h1-eth0:s8-eth1
h2 h2-eth0:s8-eth2
h3 h3-eth0:s8-eth3
h4 h4-eth0:s7-eth2
h5 h5-eth0:s7-eth3
h6 h6-eth0:s7-eth4
h7 h7-eth0:s5-eth2
h8 h8-eth0:s5-eth3
h9 h9-eth0:s1-eth5
h10 h10-eth0:s1-eth6
*** Starting CLI:
mininet>
```

รูปที่ 5.2 การสร้างระบบเครือข่าย SDN ด้วย Mininet

5.1.3. ตรวจสอบการเชื่อมต่อของระบบเครือข่าย Mininet

การตรวจสอบสถานะการเชื่อมต่อของระบบเครือข่าย SDN โดยจะประกอบไปด้วยอุปกรณ์ Host, อุปกรณ์สวิตช์ และการเชื่อมต่อระหว่างอุปกรณ์ภายในเครือข่ายที่ได้สร้างขึ้น ดังรูปที่ 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mininet> net
h1 h1-eth0:s8-eth1
h2 h2-eth0:s8-eth2
h3 h3-eth0:s8-eth3
h4 h4-eth0:s7-eth2
h5 h5-eth0:s7-eth3
h6 h6-eth0:s7-eth4
h7 h7-eth0:s5-eth2
h8 h8-eth0:s5-eth3
h9 h9-eth0:s1-eth5
h10 h10-eth0:s1-eth6
s1 lo:  s1-eth1:s2-eth2 s1-eth2:s3-eth2 s1-eth3:s4-eth2 s1-eth4:s6-eth2 s1-eth5:h9-eth
s2 lo:  s2-eth1:s5-eth4 s2-eth2:s1-eth1
s3 lo:  s3-eth1:s5-eth5 s3-eth2:s1-eth2
s4 lo:  s4-eth1:s5-eth6 s4-eth2:s1-eth3
s5 lo:  s5-eth1:s8-eth4 s5-eth2:h7-eth0 s5-eth3:h8-eth0 s5-eth4:s2-eth1 s5-eth5:s3-eth
s6 lo:  s6-eth1:s5-eth7 s6-eth2:s1-eth4
s7 lo:  s7-eth1:s1-eth7 s7-eth2:h4-eth0 s7-eth3:h5-eth0 s7-eth4:h6-eth0
s8 lo:  s8-eth1:h1-eth0 s8-eth2:h2-eth0 s8-eth3:h3-eth0 s8-eth4:s5-eth1
c0
mininet>
mininet> nodes
available nodes are:
c0 h1 h10 h2 h3 h4 h5 h6 h7 h8 h9 s1 s2 s3 s4 s5 s6 s7 s8
mininet>
```

รูปที่ 5.3 การตรวจสอบสถานะการเชื่อมต่อระบบเครือข่าย SDN ด้วย Mininet

5.2 สร้างรูปแบบการสั่งงานจากคอนโทรลเลอร์

บนระบบเครือข่าย SDN จะอาศัยการสั่งงานจากคอนโทรลเลอร์ โดยมีลักษณะการทำงานเป็นแบบรวมศูนย์ การสั่งงานทั้งหมดจะถูกกำหนดจากศูนย์กลาง โดยคอนโทรลเลอร์ที่ใช้ในการสั่งงานมีชื่อว่า POX มีรูปแบบการจัดการ การสั่งงานด้วยภาษา Python

5.2.1. การสั่งงานจากคอนโทรลเลอร์ศูนย์กลางไปยัง SDN สวิตซ์ในเครือข่าย

หลังจากที่ได้ทำการสร้างเครือข่ายด้วย Mininet จะเป็นขั้นตอนการสร้างรูปแบบการควบคุมจากคอนโทรลเลอร์เพื่อทำการสั่งงานไปยังอุปกรณ์สวิตช์ในเครือข่ายทั้งหมด ซึ่งสามารถเปรียบได้กับเมื่อในระบบเครือข่ายของผู้ให้บริการอินเทอร์เน็ตมีอุปกรณ์เครือข่ายจำนวนมาก และมาจากผู้ผลิตที่แตกต่างกัน โดยต้องมาทำงานร่วมกัน โดยจะมีการสั่งให้คอนโทรลเลอร์เริ่มทำงานเพื่อสร้างการเชื่อมต่อระหว่างคอนโทรลเลอร์และอุปกรณ์สวิตช์ในเครือข่ายด้วย Openflow โพรโทคอล และพร้อมที่จะสั่งงาน, เรียกดูค่าต่างๆ จากอุปกรณ์สวิตซ์ที่มีการเชื่อมต่อสมบูรณ์แล้ว ดังรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mininet@mininet-vm:~/pox$
mininet@mininet-vm:~/pox$ ./pox.py forwarding.spanning_tree
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
[core                  ] POX 0.2.0 (carp) is up.
[openflow.of_01        ] [00-00-00-00-00-08 3] connected
[openflow.of_01        ] [00-00-00-00-00-04 6] connected
[openflow.of_01        ] [00-00-00-00-00-01 4] connected
[openflow.of_01        ] [00-00-00-00-00-02 2] connected
[openflow.of_01        ] [00-00-00-00-00-06 8] connected
[openflow.of_01        ] [00-00-00-00-00-07 7] connected
[openflow.of_01        ] [00-00-00-00-00-03 5] connected
[openflow.of_01        ] [00-00-00-00-00-05 1] connected
```

รูปที่ 5.4 ผลการเชื่อมต่อจากคอนโทรลเลอร์เมื่อมีการเชื่อมต่อกับอุปกรณ์สวิตซ์ผ่าน Openflow โพรโทคอล

เมื่ออุปกรณ์ SDN สวิตซ์หยุดการเชื่อมต่อกับคอนโทรลเลอร์ บนคอนโทรลเลอร์จะมีการแสดงข้อมูล เพื่อบอกถึงสถานะการเชื่อมต่อ ดังรูปที่ 5.5

```
[openflow.of_01        ] [00-00-00-00-00-05 1] closed
[openflow.spanning_tree] 3 ports changed
[openflow.of_01        ] [00-00-00-00-00-02 2] closed
[openflow.spanning_tree] 3 ports changed
[openflow.of_01        ] [00-00-00-00-00-08 3] closed
[openflow.of_01        ] [00-00-00-00-00-01 4] closed
[openflow.of_01        ] [00-00-00-00-00-03 5] closed
[openflow.of_01        ] [00-00-00-00-00-04 6] closed
[openflow.of_01        ] [00-00-00-00-00-07 7] closed
[openflow.of_01        ] [00-00-00-00-00-06 8] closed
```

รูปที่ 5.5 ผลจากคอนโทรลเลอร์เมื่อหยุดเชื่อมต่อกับสวิตซ์ผ่าน Openflow โพรโทคอล

เมื่อคอนโทรลเลอร์เริ่มสถานะการทำงานและมีการสั่งงานมายัง SDN สวิตซ์จะส่งผลให้ระบบเครือข่ายสามารถส่งข้อมูลระหว่างกันได้ถูกต้อง โดยสามารถทำการตรวจสอบสถานะการเชื่อมต่อรวมทั้งสถานะการใช้งานของ Host ในเครือข่ายได้จาก Mininet ดังรูปที่ 5.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mininet>
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Results: 0% dropped (90/90 received)
mininet>
mininet>
```

รูปที่ 5.6 ผลลัพธ์ของการติดต่อกันระหว่าง Host ในเครือข่าย SDN

จากรูปแสดงถึงการทดสอบสถานะการเชื่อมต่อระหว่างกันภายในเครือข่าย ด้วยการสั่งงานคำสั่ง Pingall จาก Mininet ซึ่งหมายถึง การสั่งให้ Host แต่ละเครื่อง Ping ไปยังทุกเครื่องในระบบเครือข่าย และเมื่อดำเนินการเสร็จจะมีผลลัพธ์แสดง คือ 0% dropped แปลว่า ทุก Host ในเครือข่ายสามารถติดต่อสื่อสารระหว่างกันได้อย่างสมบูรณ์ ด้วยการสั่งงานจากคอนโทรลเลอร์

5.2.2. การสั่งงานจากคอนโทรลเลอร์ไปยังอุปกรณ์ SDN ด้วยเงื่อนไข IP address

คอนโทรลเลอร์จะสั่งงานไปยัง อุปกรณ์ SDN สวิตซ์ ผ่านช่องทางการเชื่อมต่อของ Openflow โพรโทคอล โดยจะมีการกำหนดเงื่อนไขในการสั่งงาน ตามมาตรฐานของ Openflow โพรโทคอล โดยการสั่งงาน จะใช้เงื่อนไขข้อมูล IP address เพื่อใช้ในการกำหนดรูปแบบของทราฟฟิก และส่งคำสั่งไปยังอุปกรณ์สวิตซ์ดังแสดงตามรูปที่ 5.7 - 5.12

```
if event.connection.dpid==s8_dpid:
    a=packet.find('arp')
    if a and a.protodst=="10.0.0.4":
      msg = of.ofp_packet_out(data=event.ofp)
      msg.actions.append(of.ofp_action_output(port=4))
      event.connection.send(msg)

    if a and a.protodst=="10.0.0.5":
      msg = of.ofp_packet_out(data=event.ofp)
      msg.actions.append(of.ofp_action_output(port=4))
      event.connection.send(msg)
```

รูปที่ 5.7 การกำหนดเงื่อนไขการเชื่อมต่อและการจัดการการเชื่อมต่อด้วย ARP โพรโทคอลบนสวิตซ์ SDN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.7 แสดงถึงการกำหนดเงื่อนไขในการเชื่อมโยงระหว่าง IP address ที่ Host ใช้งาน และพอร์ตที่ Host ทำการเชื่อมต่ออยู่กับอุปกรณ์สวิตซ์ SDN เป็นการทำงานด้วย ARP โพรโทคอล

```
msg = of.ofp_flow_mod()
msg.priority =100
msg.idle_timeout = 0
msg.hard_timeout = 1
msg.match.dl_type = 0x0800
msg.match.nw_dst = "10.0.0.4"
msg.actions.append(of.ofp_action_output(port = 4))
event.connection.send(msg)

msg = of.ofp_flow_mod()
msg.priority =100
msg.idle_timeout = 0
msg.hard_timeout = 1
msg.match.dl_type = 0x0800
msg.match.nw_dst = "10.0.0.5"
msg.actions.append(of.ofp_action_output(port = 4))
event.connection.send(msg)
```

รูปที่ 5.8 การกำหนดเงื่อนไขในการส่งต่อข้อมูลด้วยเงื่อนไข IP address และพอร์ต

จากรูปที่ 5.8 แสดงถึงการกำหนดเงื่อนไข สำหรับการส่งต่อข้อมูลบนอุปกรณ์สวิตซ์ SDN ด้วยเงื่อนไข IP address และพอร์ต ที่ต้องการส่งต่อข้อมูล

รูปที่ 5.9 และ 5.10 แสดงถึงการตรวจสอบค่า IP address ของ Host ด้วยการใช้คำสั่ง ชื่อ Host ที่ต้องการตามด้วยคำสั่ง ifconfig พร้อมทั้งระบุพอร์ตที่ต้องการตรวจสอบ

```
mininet> h1 ifconfig h1-eth0
h1-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.129  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:1156 (1.1 KB)

mininet> h4 ifconfig h4-eth0
h4-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:04
          inet addr:10.0.0.193  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:1066 (1.0 KB)

mininet>
```

รูปที่ 5.9 แสดงค่าคอนฟิกกูเรชั่น IP address บนเครื่อง Host 1 และ Host 4 ตามการกำหนดค่าจาก Mininet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mininet> h3 ifconfig h3-eth0
h3-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:03
          inet addr:10.0.0.162  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:1156 (1.1 KB)

mininet> h6 ifconfig h6-eth0
h6-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:06
          inet addr:10.0.0.226  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:906 (906.0 B)

mininet>
```

รูปที่ 5.10 แสดงค่าคอนฟิกกูเรชั่น IP address บนเครื่อง Host 3 และ Host 6 ตามการกำหนดค่าจาก Mininet

คอนโทรลเลอร์จะสามารถสั่งงานไปยัง SDN สวิตซ์ด้วยเงื่อนไข คือ IP addrsss และ Ethernet-type ได้แสดงดังรูปที่ 5.11, 5.12 และ 5.13

โดยรูปที่ 5.11 5.12 และ 5.13 แสดงถึงการสั่งงานด้วยเงื่อนไข IP address ต้นทาง IP address ปลายทางและเงื่อนไข Ethernet-type ซึ่งมีการกำหนดค่าเป็น 0x0800 มีความหมายว่า IP version4 และมีการสั่งงานไปยัง SDN สวิตซ์ 5 และ SDN สวิตซ์ 1 สำหรับการส่งข้อมูลของดาต้าเซ็นเตอร์แห่งที่ 1, ดาต้าเซ็นเตอร์ แห่งที่ 2 จากการสั่งงานด้วยรูปแบบนี้ถือว่าเป็นสิ่งที่แตกต่าง จากการทำงานของระบบเครือข่ายในปัจจุบันที่มีการ ส่งต่อข้อมูลด้วยเงื่อนไข IP address ปลายทางเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
msg = of.ofp_flow_mod()
msg.command=of.OFPFC_MODIFY_STRICT
msg.priority =100
msg.idle_timeout = 0
msg.hard_timeout = 0
msg.match.dl_type = 0x0800
msg.match.nw_src = "10.0.0.128/27"
msg.match.nw_dst = "10.0.0.192/27"
msg.actions.append(of.ofp_action_output(port = 4))
core.openflow.getConnection(s5_dpid).send(msg)

msg = of.ofp_flow_mod()
msg.command=of.OFPFC_MODIFY_STRICT
msg.priority =100
msg.idle_timeout = 0
msg.hard_timeout = 0
msg.match.dl_type = 0x0800
msg.match.nw_src = "10.0.0.160/27"
msg.match.nw_dst = "10.0.0.224/27"
msg.actions.append(of.ofp_action_output(port = 5))
core.openflow.getConnection(s5_dpid).send(msg)
```

รูปที่ 5.11 คำสั่งงานบนคอนโทรลเลอร์สั่งงานไปที่สวิตซ์ 5 สำหรับดาต้าเซ็นเตอร์ แห่งที่ 1

```
msg = of.ofp_flow_mod()
msg.command=of.OFPFC_MODIFY_STRICT
msg.priority =100
msg.idle_timeout = 0
msg.hard_timeout = 0
msg.match.dl_type = 0x0800
msg.match.nw_src = "10.0.0.192/27"
msg.match.nw_dst = "10.0.0.128/27"
msg.actions.append(of.ofp_action_output(port = 1))
core.openflow.getConnection(s1_dpid).send(msg)

msg = of.ofp_flow_mod()
msg.command=of.OFPFC_MODIFY_STRICT
msg.priority =100
msg.idle_timeout = 0
msg.hard_timeout = 0
msg.match.dl_type = 0x0800
msg.match.nw_src = "10.0.0.224/27"
msg.match.nw_dst = "10.0.0.160/27"
msg.actions.append(of.ofp_action_output(port = 2))
core.openflow.getConnection(s1_dpid).send(msg)
```

รูปที่ 5.12 คำสั่งงานบนคอนโทรลเลอร์สั่งงานไปที่สวิตซ์ 1 สำหรับดาต้าเซ็นเตอร์ แห่งที่ 1

จากรูป 5.11 เป็นการสั่งงานไปยังสวิตซ์ 5 เพื่อให้ทำงาน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงาน จากเงื่อนไขที่ 1

1) หาก Flow ที่เขามาเป็น Ehternet type = 0x0800 = IP version4

2) หาก Flow ที่เข้ามามี IP address ปลายทาง = 10.0.0.192/27

3) หาก Flow ที่เข้ามามี IP address ต้นทาง = 10.0.0.128/27

4) หากทั้ง 3 ข้อ AND กันเป็นจริง จะส่งข้อมูลออกไปยัง Local พอร์ตที่ 4 บน สวิตซ์ 5

การทำงาน จากเงื่อนไขที่ 2

1) หาก Flow ที่เขามาเป็น Ehternet type = 0x0800 = IP version4

2) หาก Flow ที่เข้ามามี IP address ปลายทาง = 10.0.0.224/27

3) หาก Flow ที่เข้ามามี IP address ต้นทาง = 10.0.0.160/27

4) หากทั้ง 3 ข้อ AND กันเป็นจริง จะส่งข้อมูลออกไปยัง Local พอร์ตที่ 5 บน สวิตซ์ 5

จากรูป 5.12 เป็นการสั่งงานไปยังสวิตซ์ 1 เพื่อให้ทำงาน ดังนี้

การทำงาน จากเงื่อนไขที่ 1

1) หาก Flow ที่เขามาเป็น Ehternet type = 0x0800 = IP version4

2) หาก Flow ที่เข้ามามี IP address ปลายทาง = 10.0.0.128/27

3) หาก Flow ที่เข้ามามี IP address ต้นทาง = 10.0.0.192/27

4) หากทั้ง 3 ข้อ AND กันเป็นจริง จะส่งข้อมูลออกไปยัง Local พอร์ตที่ 1 บน สวิตซ์ 1

การทำงาน จากเงื่อนไขที่ 2

1) หาก Flow ที่เขามาเป็น Ehternet type = 0x0800 = IP version4

2) หาก Flow ที่เข้ามามี IP address ปลายทาง = 10.0.0.160/27

3) หาก Flow ที่เข้ามามี IP address ต้นทาง = 10.0.0.224/27

4) หากทั้ง 3 ข้อ AND กันเป็นจริง จะส่งข้อมูลออกไปยัง Local พอร์ตที่ 2 บน สวิตซ์ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
msg = of.ofp_flow_mod()
msg.command=of.OFPFC_MODIFY_STRICT
msg.priority =100
msg.idle_timeout = 0
msg.hard_timeout = 0
msg.match.dl_type = 0x0800
msg.match.nw_src = "10.0.0.224/27"
msg.match.nw_dst = "10.0.0.160/27"
msg.actions.append(of.ofp_action_output(port = 5))
core.openflow.getConnection(s5_dpid).send(msg)
```

รูปที่ 5.13 คำสั่งงานบนคอนโทรลเลอร์สั่งงานไปที่สวิตซ์ 5 สำหรับดาต้าเซ็นเตอร์ แห่งที่ 2

จากรูป 5.13 เป็นการสั่งงานไปยัง สวิตซ์ 5 เพื่อให้ทำงาน ดังนี้

1) หาก Flow ที่เขามาเป็น Ehternet type = 0x0800 = IP version4

2) หาก Flow ที่เข้ามามี IP address ปลายทาง = 10.0.0.160/27

3) หาก Flow ที่เข้ามามี IP address ต้นทาง = 10.0.0.224/27

4) หากทั้ง 3 ข้อ AND กันเป็นจริง จะส่งออกไปยัง Local พอร์ตที่ 5 บน สวิตซ์ 5

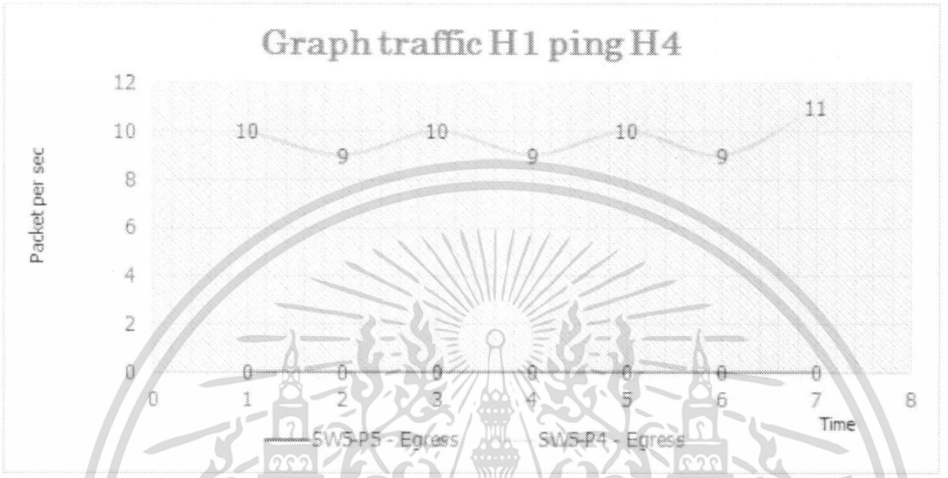
เมื่อมีการทดสอบส่งทราฟฟิกจาก Host 1 ไปยัง Host 4 จะพบว่าแพ็กเก็ตจะถูกส่งตรงกับเงื่อนไขที่มีการสั่งงานจากคอนโทรลเลอร์ ไปยัง SDN สวิตซ์โดยใช้เงื่อนไข IP address ต้นทาง คือ 10.0.0.128/27, IP Address ปลายทาง คือ 10.0.0.192/27 และเป็นการส่งทราฟฟิกในรูปแบบของ IP version4 จะทำการส่งทราฟฟิกผ่านทางพอร์ต 4 บนสวิตซ์ 5 จะได้ผลแสดงดังรูปที่ 5.14 ซึ่งเป็นการแสดงให้เห็นถึงการสั่งงานจากคอนโทรลเลอร์มายัง SDN สวิตซ์ สามารถทำงานร่วมกันได้อย่างถูกต้อง

```
mininet> h1 ping -i 0.1 -c 10 h4
PING 10.0.0.193 (10.0.0.193) 56(84) bytes of data.
64 bytes from 10.0.0.193: icmp_seq=2 ttl=64 time=74.4 ms
64 bytes from 10.0.0.193: icmp_seq=3 ttl=64 time=72.4 ms
64 bytes from 10.0.0.193: icmp_seq=4 ttl=64 time=70.1 ms
64 bytes from 10.0.0.193: icmp_seq=5 ttl=64 time=75.3 ms
64 bytes from 10.0.0.193: icmp_seq=6 ttl=64 time=76.7 ms
64 bytes from 10.0.0.193: icmp_seq=7 ttl=64 time=70.1 ms
64 bytes from 10.0.0.193: icmp_seq=8 ttl=64 time=72.5 ms
64 bytes from 10.0.0.193: icmp_seq=9 ttl=64 time=80.7 ms
64 bytes from 10.0.0.193: icmp_seq=10 ttl=64 time=73.0 ms

--- 10.0.0.193 ping statistics ---
10 packets transmitted, 9 received, 10% packet loss, time 910ms
rtt min/avg/max/mdev = 70.113/73.961/80.749/3.169 ms
mininet>
```

รูปที่ 5.14 การส่งแพ็กเก็ตจาก Host1 ไปยัง Host4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟแสดงปริมาณจำนวนแพ็กเก็ตเมื่อมีการส่งทราฟฟิกจาก Host 1 ไปยัง Host 4 ผลลัพธ์คือสวิตซ์ 5 จะส่งแพ็กเก็ตออกผ่านทางพอร์ต 4 ผ่านทาง สวิตซ์ 2 เท่านั้น ตรงตามเงื่อนไข ที่มีการเขียนโปรแกรมสั่งงานผ่านทางคอนโทรลเลอร์ ดังรูปที่ 5.15 และ 5.16 โดยผลลัพธ์ดังกล่าวมีการเรียกดูค่ามาจากอุปกรณ์ SDN สวิตซ์ด้วยคอนโทรลเลอร์ ปัจจุบันยังคงไม่มีระบบแสดงภาพในการสื่อสารถึงค่าต่างๆ ในส่วนการแสดงผลจากคอนโทรลเลอร์



รูปที่ 5.15 กราฟแสดงค่าข้อมูลแพ็กเก็ตที่มีการส่งออกไปจาก พอร์ต 4 ที่ สวิตซ์ 5

```
Switch.5_port:5 -> S3:  0 tx packets
Switch.5_port:4 -> S2:  9 tx packets

Switch.5_port:5 -> S3:  0 tx packets
Switch.5_port:4 -> S2:  10 tx packets

Switch.5_port:5 -> S3:  0 tx packets
Switch.5_port:4 -> S2:  9 tx packets

Switch.5_port:5 -> S3:  0 tx packets
Switch.5_port:4 -> S2:  10 tx packets

Switch.5_port:5 -> S3:  0 tx packets
Switch.5_port:4 -> S2:  9 tx packets

Switch.5_port:5 -> S3:  0 tx packets
Switch.5_port:4 -> S2:  10 tx packets
```

รูปที่ 5.16 การแสดงค่าข้อมูลแพ็กเก็ตที่มีการส่งออกไปจากพอร์ต 4 ที่ สวิตซ์ 5

เมื่อมีการส่งทราฟฟิกจาก Host 3 ไปยัง Host 6 จะพบว่าแพ็กเก็ตถูกส่งตรงกับเงื่อนไขที่มีการสั่งงานจากคอนโทรลเลอร์ไปยังสวิตซ์ 5 โดยมี IP Address ต้นทาง คือ 10.0.0.160/27 มีการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

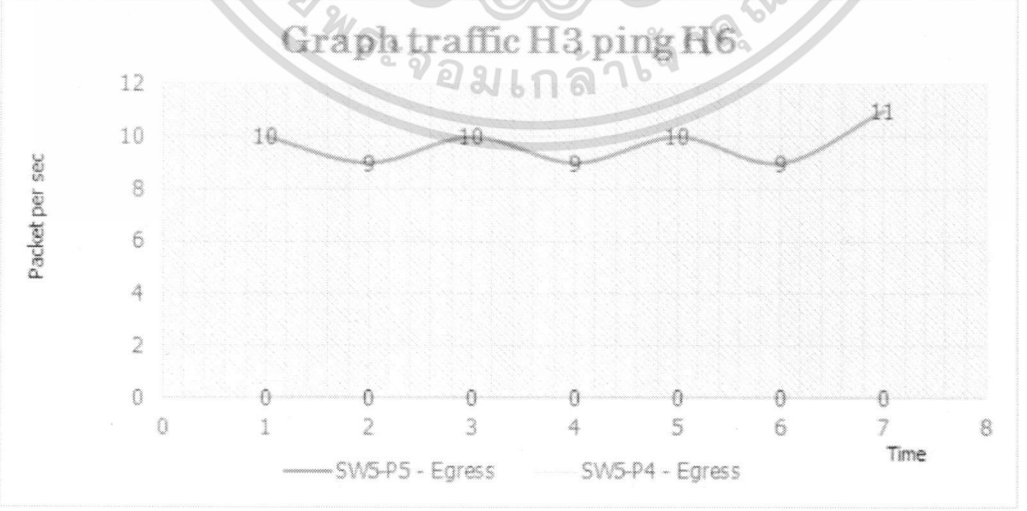
แพ็กเก็ตไปยัง IP Address ปลายทาง คือ 10.0.0.224/27 จะทำการส่งทราฟฟิกผ่านทางพอร์ต 5 บนสวิตซ์ 5 ดังรูปที่ 5.17 , 5.18 และ 5.19

การทดสอบส่งทราฟฟิกระหว่าง Host 3 และ Host 6 เพื่อเป็นการทดสอบเงื่อนไขการทำงานของ SDN สวิตซ์ ที่มีการสั่งงานการทำงานมาจากคอนโทรลเลอร์ ดังรูปรูปที่ 5.17

```
mininet> h3 ping -i 0.1 -c 10 h6
PING 10.0.0.226 (10.0.0.226) 56(84) bytes of data.
64 bytes from 10.0.0.226: icmp_seq=2 ttl=64 time=440 ms
64 bytes from 10.0.0.226: icmp_seq=3 ttl=64 time=440 ms
64 bytes from 10.0.0.226: icmp_seq=4 ttl=64 time=439 ms
64 bytes from 10.0.0.226: icmp_seq=5 ttl=64 time=446 ms
64 bytes from 10.0.0.226: icmp_seq=6 ttl=64 time=441 ms
64 bytes from 10.0.0.226: icmp_seq=7 ttl=64 time=443 ms
64 bytes from 10.0.0.226: icmp_seq=8 ttl=64 time=440 ms
64 bytes from 10.0.0.226: icmp_seq=9 ttl=64 time=443 ms
64 bytes from 10.0.0.226: icmp_seq=10 ttl=64 time=442 ms

--- 10.0.0.226 ping statistics ---
10 packets transmitted, 9 received, 10% packet loss, time 939ms
rtt min/avg/max/mdev = 439.794/441.894/446.049/1.979 ms, pipe 5
mininet>
```

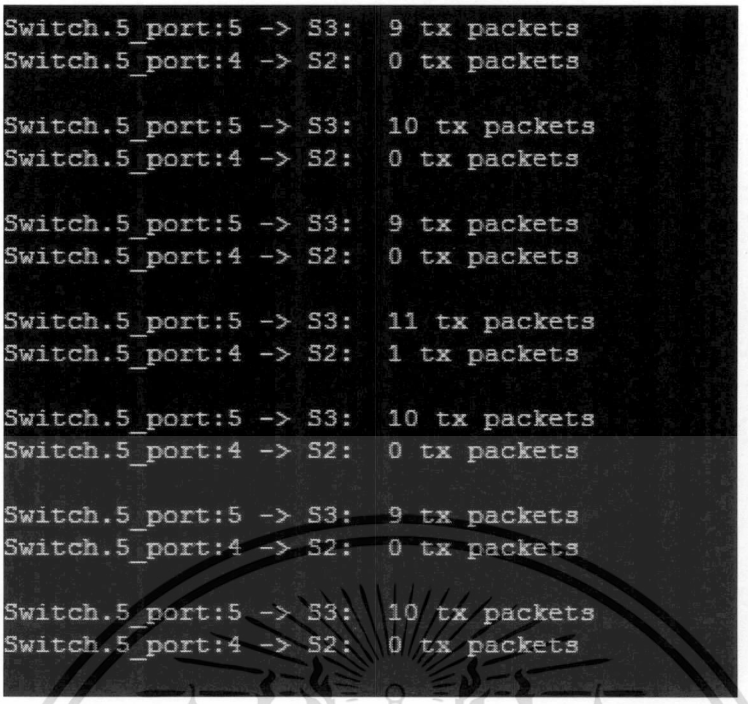
รูปที่ 5.17 การส่งแพ็กเก็ตจาก Host3 ไปยัง Host6

กราฟแสดงปริมาณจำนวนแพ็กเก็ต เมื่อมีการส่งทราฟฟิกจาก Host 3 ไปยัง Host 6 ผลลัพธ์คือ สวิตซ์ 5 จะการส่งแพ็กเก็ตออกผ่านทางพอร์ต 5 ผ่านทาง สวิตซ์ 5 เท่านั้น ตรงตามเงื่อนไข ที่มีการเขียนโปรแกรมสั่งงานผ่านทางคอนโทรลเลอร์ ดังรูปที่ 5.18 และ 5.19



รูปที่ 5.18 การแสดงค่าข้อมูลแพ็กเก็ตที่มีการส่งออกไปจากพอร์ต 5 ที่สวิตซ์ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.19 การแสดงค่าข้อมูลแพ็กเก็ตที่มีการส่งออกไปจากพอร์ต 5 ที่สวิตซ์ 5

5.2.3. การตรวจสอบปริมาณการใช้งานของอุปกรณ์ SDN ด้วยคอนโทรลเลอร์และการกำหนดเงื่อนไข

หากมีปริมาณทราฟฟิกคับคั่งคอนโทรลเลอร์จะสั่งงานไปยังอุปกรณ์ SDN สวิตซ์ด้วยเงื่อนไข IP address เพื่อทำการแบ่งปริมาณทราฟฟิกส่งผลให้ลดโอกาสเกิดการคับคั่งของปริมาณทราฟฟิกโดยอัตโนมัติ

เงื่อนไขการจัดการทราฟฟิกจะโดนสั่งงานจาก คอนโทรลเลอร์ไปที่อุปกรณ์ SDN สวิตซ์ที่กำหนด โดยการสั่งการจะเป็นในรูปแบบของ Dynamic คือ คอนโทรลเลอร์ จะทำการตรวจสอบปริมาณการใช้งานของพอร์ตที่กำหนดตลอดเวลา ด้วยการเรียกดูค่าปริมาณการใช้งานทราฟฟิกจากอุปกรณ์สวิตซ์ในช่วงเวลาแต่ละวินาที เพื่อตรวจสอบความหนาแน่นของพอร์ตดังกล่าว ในการส่งข้อมูลแพ็กเก็ตเพื่อเป็นการช่วยให้ เพิ่มโอกาสในการส่งข้อมูลแพ็กเก็ตได้ในปริมาณเพิ่มขึ้น เนื่องจาก สวิตซ์สามารถส่งข้อมูลได้จำนวน 2 ช่องทาง คือช่องทางหลักและช่องทางเสริมตามเงื่อนไขกำหนด ดังนั้นเมื่อเครือข่ายเกิดความหนาแน่นของข้อมูลแพ็กเก็ต จะมีการแบ่งปริมาณการใช้งานข้อมูลแพ็กเก็ตไปยังเส้นทางเสริม จึงส่งผลให้สามารถเพิ่มโอกาสในการใช้งานส่งข้อมูลได้มากขึ้นโดยปริมาณการใช้งานจะพิจารณาจากค่าทราฟฟิกบนสวิตซ์ โดยจะนำมาคำนวณเป็นจำนวนเปอร์เซ็นต์การใช้งานและเมื่อค่าจำนวนเปอร์เซ็นต์การใช้งานมากกว่าค่าที่กำหนดไว้จะดำเนินการสั่งงานไปยังสวิตซ์เพื่อทำการแบ่งทราฟฟิก ดังรูปที่ 5.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if event.connection.dpid==s5_dpid:
  for f in event.stats:
    if int(f.port_no)<65534:

if f.port_no==6:
  pre_s5_p6=s5_p6
  s5_p6=f.tx_bytes
  pre_s5_p6_packet=s5_p6_packet
  s5_p6_packet=f.tx_packets
  check_usage_s5_p6  = (((((-(pre_s5_p6-s5_p6))*8) / 1024000.00)/10)*100)
  if check_usage_s5_p6  >= 80.00:
     print "S5-P6-exceed 80%"
     _share_dc2 ();

if f.port_no==7:
  pre_s5_p7=s5_p7
  s5_p7=f.tx_bytes
  pre_s5_p7_packet=s5_p7_packet
  s5_p7_packet=f.tx_packets
  check_usage_s5_p7 = (((((-(pre_s5_p7-s5_p7))*8) / 1024000.00)/10)*100)
  if check_usage_s5_p7 >= 80.00:
     print "S5-P7-exceed 80%"
     _share_dc2 ();
  print ""
```

รูปที่ 5.20 การตรวจสอบสถานะการใช้งานของพอร์ตและกำหนดเงื่อนไข

การแสดงถึงกระบวนการในการจำลองการส่งทราฟฟิก โดยใช้โปรแกรม iperf ทำงานในรูปแบบ server โดยให้ทำงานที่ Host 9 เพื่อรอรับการส่งทราฟฟิกจาก client คือ เครื่อง Host อื่นๆ แสดงดังรูปที่ 5.21

```
mininet>
mininet>
mininet> h9 iperf -s -u &
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
mininet>
mininet>
```

รูปที่ 5.21 การทดสอบสร้าง iperf mode server ที่ Host 9

การแสดงถึงการส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 ด้วยโปรแกรม iperf ด้วยปริมาณทราฟฟิก ขนาด 1 Mbps ดังรูปที่ 5.22

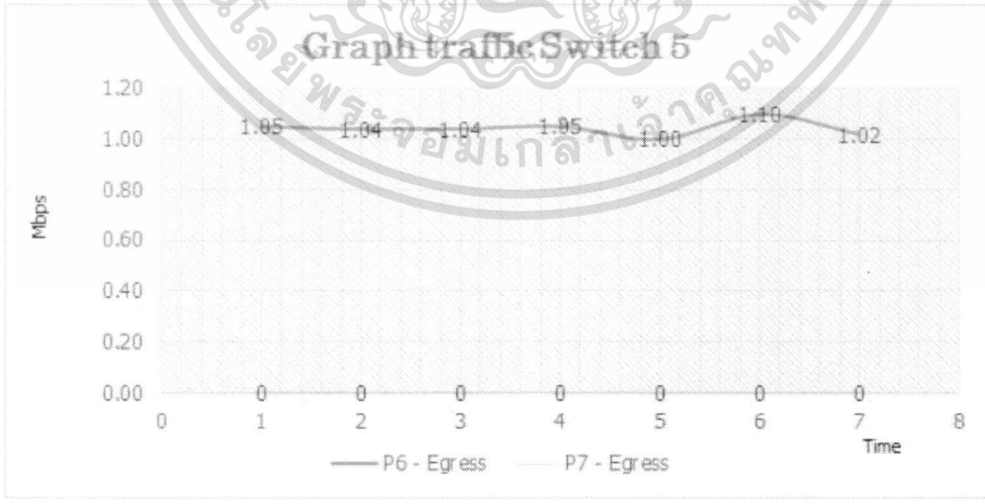
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mininet> h7 iperf -c h9 -b 1m -w 128k
WARNING: option -b implies udp testing
------------------------------------------------------------
Client connecting to 10.0.0.9, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  256 KByte (WARNING: requested  128 KByte)
------------------------------------------------------------
[  3] local 10.0.0.7 port 53643 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.19 MBytes  1000 Kbits/sec
[  3] Sent 852 datagrams
[  3] Server Report:
[  3]  0.0-370.5 sec  1.19 MBytes  27.0 Kbits/sec   0.603 ms   23/  852 (2.7%)
[  3]  0.0-370.5 sec  23 datagrams received out-of-order
mininet>
```

รูปที่ 5.22 การทดสอบสร้าง iperf mode client เพื่อส่งทราฟฟิกปริมาณ 1 Mbps จาก Host 7 ไปยัง Host 9

ผลลัพธ์ที่แสดงจากคอนโทรลเลอร์ที่มีการเรียกดูค่าจาก SDN สวิตซ์จะเห็นได้ว่ามีการใช้งานทราฟฟิกอยู่เพียง 1 พอร์ตเท่านั้น และแสดงถึงปริมาณทราฟฟิกที่ตรงกับขนาดทราฟฟิกที่ส่งออกจาก Host7 ดังรูปที่ 5.22 คือ 1000 Kbits/sec หรือ 1Mbps

กราฟแสดงปริมาณทราฟฟิกบนดาต้าเซ็นเตอร์ 2 กรณีมีการใช้งานปริมาณทราฟฟิกในช่องทางหลักน้อยกว่า 80 เปอร์เซ็นต์ ผลลัพธ์คือ สวิตซ์ 5 จะการส่งแพ็กเก็ต ออกผ่านทางพอร์ต 6 ไปยัง สวิตซ์ 4 เท่านั้น ซึ่งตรงตามเงื่อนไข ที่มีการเขียนโปรแกรมสั่งงานผ่านทางคอนโทรลเลอร์ ดังรูปที่ 5.23



รูปที่ 5.23 กราฟผลลัพธ์จากคอนโทรลเลอร์ที่รับมาจากสวิตซ์ 5 กรณีส่งทราฟฟิกที่ Host 7 ไปยัง Host 9 กรณีมีการใช้งานน้อยกว่าเงื่อนไขที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Switch.5-port_6(bw:10Mbps) -> S4:  1.05 MBits used 10.51 %
Switch.5-port_7(bw:10Mbps) -> S4:  0.00 MBits used 0.00 %

Switch.5-port_6(bw:10Mbps) -> S4:  1.04 MBits used 10.40 %
Switch.5-port_7(bw:10Mbps) -> S4:  0.00 MBits used 0.00 %

Switch.5-port_6(bw:10Mbps) -> S4:  1.04 MBits used 10.40 %
Switch.5-port_7(bw:10Mbps) -> S4:  0.00 MBits used 0.00 %

Switch.5-port_6(bw:10Mbps) -> S4:  1.04 MBits used 10.40 %
Switch.5-port_7(bw:10Mbps) -> S4:  0.00 MBits used 0.00 %
```

รูปที่ 5.24 ผลลัพธ์จากคอนโทรลเลอร์ที่รับมาจากสวิตซ์ 5 กรณีส่งทราฟฟิกที่ Host 7 ไปยัง Host 9 กรณีมีการใช้งานน้อยกว่าเงื่อนไขที่กำหนดไว้

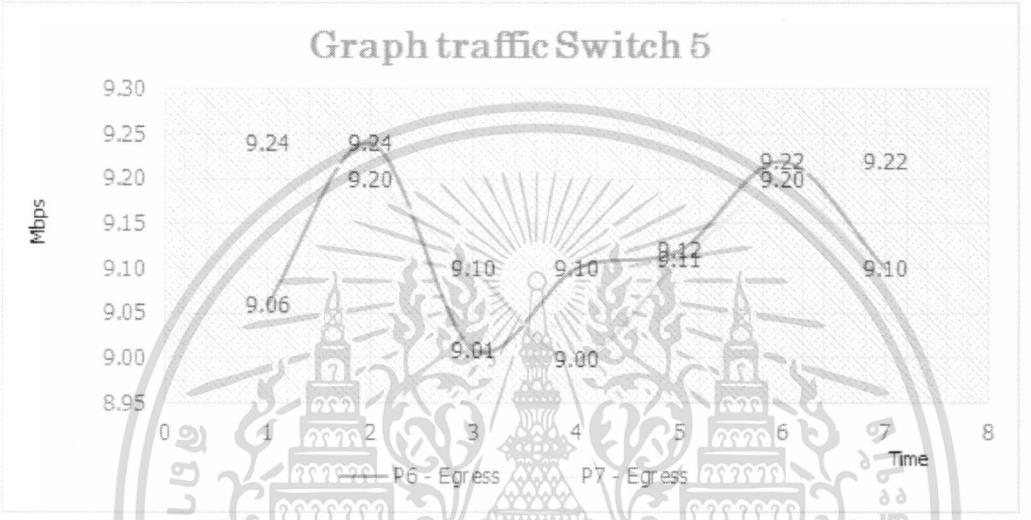
การแสดงถึงการส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 ผ่ายโปรแกรม iperf โดยส่งทราฟฟิกที่ขนาด 9 Mbps ซึ่งคิดเป็น 90 เปอร์เซ็นต์ ของขนาด Link เพื่อเป็นการทดสอบการทำงานของ SDN สวิตซ์ ตามเงื่อนไขจากการสั่งงานจากคอนโทรลเลอร์ ดังรูปที่ 5.25

```
mininet> h7 iperf -c h9 -b 9m -w 128k -i 1 -m
WARNING: option -b implies udp testing
------------------------------------------------------------
Client connecting to 10.0.0.9, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  256 KByte (WARNING: requested  128 KByte)
------------------------------------------------------------
[  3] local 10.0.0.7 port 59647 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  1.07 MBytes  9.01 Mbits/sec
[  3]  1.0- 2.0 sec  1.07 MBytes  9.01 Mbits/sec
[  3]  2.0- 3.0 sec  1.07 MBytes  9.01 Mbits/sec
[  3]  3.0- 4.0 sec  1.07 MBytes  9.00 Mbits/sec
[  3]  4.0- 5.0 sec  1.07 MBytes  9.01 Mbits/sec
[  3]  5.0- 6.0 sec  1.07 MBytes  9.01 Mbits/sec
[  3]  6.0- 7.0 sec  1.07 MBytes  9.00 Mbits/sec
[  3]  7.0- 8.0 sec  1.07 MBytes  9.01 Mbits/sec
[  3]  8.0- 9.0 sec  1.07 MBytes  9.01 Mbits/sec
[  3]  9.0-10.0 sec  1.07 MBytes  9.00 Mbits/sec
[  3]  0.0-10.0 sec  10.7 MBytes  9.00 Mbits/sec
[  3] Sent 7658 datagrams
[  3] Server Report:
[  3]  0.0-958.1 sec  6.91 MBytes  60.5 Kbits/sec  433.330 ms 2721/ 7652 (36%)
[  3]  0.0-958.1 sec  913 datagrams received out-of-order
mininet>
```

รูปที่ 5.25 การทดสอบสร้าง iperf mode client และส่งทราฟฟิกปริมาณ 9 Mbps ที่ Host7 ไปยัง Host9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟแสดงปริมาณทราฟฟิกบนดาต้าเซ็นเตอร์ 2 กรณีที่มีการใช้งานปริมาณทราฟฟิกในช่องทางหลักมากกว่า 80 เปอร์เซ็นต์ ผลลัพธ์คือ สวิตซ์ 5 จะแบ่งข้อมูลแพ็กเก็ต ส่งออกผ่านทางพอร์ต 6 และ พอร์ต 7 ตรงตามเงื่อนไขที่มีการเขียนโปรแกรมสั่งงานผ่านทางคอนโทรลเลอร์ ดังรูปที่ 5.26 และ 5.27 ส่งผลทำให้ สามารถเพิ่มช่องทางในการส่งทราฟฟิกได้มากขึ้น ทำให้ปริมาณการส่งข้อมูลแพ็กเก็ตเพิ่มจำนวนมากขึ้น เพราะ สามารถส่งข้อมูลผ่านทางช่องทางหลัก และ ช่องทางสำรองได้ในเวลาเดียวกัน ทำให้จำนวนข้อมูลแพ็กเก็ตโดยรวม สามารถส่งได้มีปริมาณเพิ่มขึ้น



รูปที่ 5.26 กราฟผลลัพธ์จากคอนโทรลเลอร์ที่รับมาจากสวิตซ์ 5 กรณีส่งทราฟฟิกที่ Host7 ไปยัง Host9 กรณีมีการใช้งานมากกว่าเงื่อนไขที่กำหนดไว้

```
Switch.5-port_6(bw:10Mbps) -> S4:  9.06 MBits used 90.61 %
Switch.5-P6-main-exceed 80%
Switch.5-port_7(bw:10Mbps) -> S4:  9.24 MBits used 92.37 %
Switch.5-P7-backup-exceed 80%


Switch.5-port_6(bw:10Mbps) -> S4:  9.24 MBits used 92.37 %
Switch.5-P6-main-exceed 80%
Switch.5-port_7(bw:10Mbps) -> S4:  9.23 MBits used 92.26 %
Switch.5-P7-backup-exceed 80%
```

รูปที่ 5.27 ผลลัพธ์จากคอนโทรลเลอร์ที่รับมาจากสวิตซ์ 5 กรณีส่งทราฟฟิกที่ Host 7 ไปยัง Host 9 กรณีมีการใช้งานมากกว่าเงื่อนไขที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การจำลองส่งทราฟฟิกบนระบบเครือข่าย SDN และระบบเครือข่ายปัจจุบัน

ในการทดลองหัวข้อนี้ มีการกำหนดพื้นฐาน การไหลของทราฟฟิกในเครือข่ายจำลอง SDN เป็นดังนี้

1) ทราฟฟิกระหว่าง DC01 Site B และ DC02 Site A จะมีการส่งทราฟฟิก ดังนี้

DC01 Site B < > Switch S8 < > Switch S5 < > Switch S6 < > Switch S1 < > DC02 Site A.

ในการส่งทราฟฟิกจะเป็นการส่งผ่านทาง ช่องทางที่ 2 โดยกำหนดขนาดความสามารถในการส่งทราฟฟิก ของช่องทางที่ 2 มีค่าเท่ากับ 100 เมกะบิตต่อวินาที

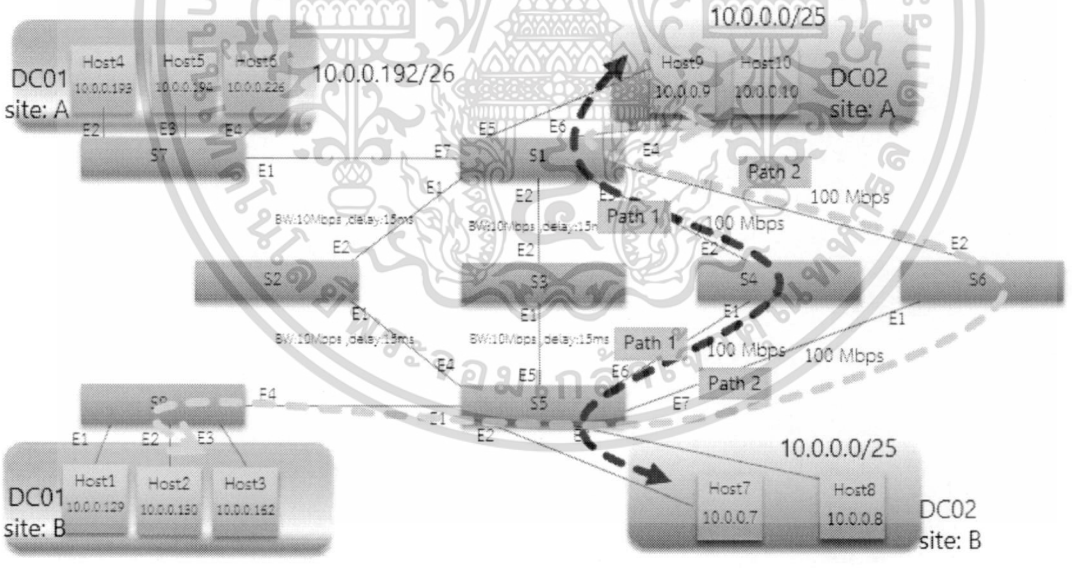
2) ทราฟฟิกระหว่าง DC02 Site B และ DC02 Site A จะมีการส่งทราฟฟิก ดังนี้

DC02 Site B < > Switch S5 < > Switch S4 < > Switch S1 < > DC02 Site A.

ในการส่งทราฟฟิกจะเป็นการส่งผ่านทาง ช่องทางที่ 1 โดยกำหนดขนาดความสามารถในการส่งทราฟฟิกได้ ของช่องทางที่ 1 มีค่าเท่ากับ 100 เมกะบิตต่อวินาที

3) ทราฟฟิกในกรณีที่ไม่ได้ระบุในการทดลองนี้จะไม่นำมาพิจารณา

จากข้อกำหนดพื้นฐานทั้งหมด สามารถแสดงได้ดังรูปที่ 5.28



รูปที่ 5.28 สถานะการส่งทราฟฟิกในกรณีพื้นฐาน

คอนโทรลเลอร์จะมีการตรวจสอบปริมาณการใช้งาน ของเครือข่าย DC02 Site B กรณีส่งทราฟฟิกไปยัง DC02 Site A โดยจะพิจารณาปริมาณทราฟฟิกของพอร์ตบนอุปกรณ์สวิตซ์ S5 โดยในเงื่อนไขของการแบ่งทราฟฟิก จะพิจารณาจากปริมาณทราฟฟิกบนพอร์ตของอุปกรณ์สวิตซ์ S5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเส้นทางที่ 1 คือ หากมีปริมาณทราฟฟิกมากกว่า 80 เปอร์เซ็นต์ ของความสามารถในการส่งทราฟฟิก จะทำการแบ่งทราฟฟิกไปยังเส้นทางที่ 2 ผ่านอุปกรณ์สวิตซ์ S6 โดยจะทดสอบบนระบบเครือข่ายที่ทำงานในรูปแบบ SDN และ ระบบเครือข่ายที่ทำงานในรูปแบบปัจจุบัน ซึ่งจะพิจารณาในส่วนของการส่งทราฟฟิกเท่านั้นเพื่อเป็นการทดสอบความสามารถ ในการจัดการทราฟฟิกในสถานการณ์ที่แตกต่างกัน โดยมีเงื่อนไขในการดำเนินการ ดังนี้

บนเครือข่าย SDN อุปกรณ์คอนโทรลเลอร์จะทำการตรวจสอบปริมาณทราฟฟิกของพอร์ตบนอุปกรณ์สวิตซ์ S5 บนช่องทางการส่งทราฟฟิกระหว่าง DC02 เส้นที่ 1 คือ พอร์ต E6 โดยจะพิจารณาค่าปริมาณทราฟฟิก คือ

1) มีปริมาณทราฟฟิกน้อยกว่า 80 เปอร์เซ็นต์ จะไม่มีการแบ่งทราฟฟิก
2) มีปริมาณทราฟฟิกมากกว่า 80 เปอร์เซ็นต์ จะพิจารณาแบ่งทราฟฟิก โดยใช้เงื่อนไข IP address ต้นทาง ในการแบ่งทราฟฟิก ให้ไปใช้งานผ่านเส้นทางที่ 2

5.3.1 การตรวจสอบปริมาณทราฟฟิกของพอร์ต บนอุปกรณ์สวิตซ์ S5 บนเส้นที่ 1 และ 2

คอนโทรลเลอร์ ในเครือข่าย SDN จะตรวจสอบปริมาณการใช้งานปัจจุบัน บนสวิตซ์ S5 พอร์ต E6 และ E7 เพื่อตรวจสอบความหนาแน่นของทราฟฟิกบนเส้นทางที่ 1 และ 2 ดังรูปที่ 5.29

```
S5-port_E6(BW:100 Mbps) -> S4:  0.00 Mbps used 0.00 %
S5-port_E7(BW:100 Mbps) -> S6:  0.00 Mbps used 0.00 %

S5-port_E6(BW:100 Mbps) -> S4:  0.00 Mbps used 0.00 %
S5-port_E7(BW:100 Mbps) -> S6:  0.00 Mbps used 0.00 %

S5-port_E6(BW:100 Mbps) -> S4:  0.00 Mbps used 0.00 %
S5-port_E7(BW:100 Mbps) -> S6:  0.00 Mbps used 0.00 %

S5-port_E6(BW:100 Mbps) -> S4:  0.00 Mbps used 0.00 %
S5-port_E7(BW:100 Mbps) -> S6:  0.00 Mbps used 0.00 %

S5-port_E6(BW:100 Mbps) -> S4:  0.00 Mbps used 0.00 %
S5-port_E7(BW:100 Mbps) -> S6:  0.00 Mbps used 0.00 %

S5-port_E6(BW:100 Mbps) -> S4:  0.00 Mbps used 0.00 %
S5-port_E7(BW:100 Mbps) -> S6:  0.00 Mbps used 0.00 %

S5-port_E6(BW:100 Mbps) -> S4:  0.00 Mbps used 0.00 %
S5-port_E7(BW:100 Mbps) -> S6:  0.00 Mbps used 0.00 %
```

รูปที่ 5.29 แสดงการตรวจสอบสถานะปริมาณการใช้งานทราฟฟิก ด้วยคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบจำลองส่งทราฟฟิกจาก Host 7 ไป Host 9 ผ่านโปรแกรม iPref ด้วยปริมาณทราฟฟิก 50 เมกะบิตต่อวินาที ดังรูปที่ 5.30 และ ส่งทราฟฟิกจาก DC01 site B Host 3 ไปยัง Host 10 ด้วยปริมาณทราฟฟิก 50 เมกะบิตต่อวินาที ส่งผลให้มีปริมาณทราฟฟิกใช้งานบนเส้นทางที่ 1 และ 2 อยู่ที่ประมาณ 50 เปอร์เซ็นต์ ของความสามารถในการส่งทราฟฟิก ดังรูปที่ 5.30 – 5.31

```
mininet> h7 iperf -c h9 -t 10 -i 1 -b50m
WARNING: option -b implies udp testing
------------------------------------------------------------
Client connecting to 10.0.0.9, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.7 port 48702 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  1.0- 2.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  2.0- 3.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  3.0- 4.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  4.0- 5.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  5.0- 6.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  6.0- 7.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  7.0- 8.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  8.0- 9.0 sec  5.97 MBytes  50.0 Mbits/sec
[  3]  9.0-10.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  0.0-10.0 sec  59.6 MBytes  50.0 Mbits/sec
[  3] Sent 42552 datagrams
read failed: Connection refused
[  3] WARNING: did not receive ack of last datagram after 2 tries.
mininet>
```

รูปที่ 5.30 แสดงการทดลองส่งทราฟฟิกจาก Host 7 ไปยัง Host 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mininet> h3 iperf -c h10 -t 10 -i 1 -b50m
WARNING: option -b implies udp testing
------------------------------------------------------------
Client connecting to 10.0.0.10, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 41656 connected with 10.0.0.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  5.95 MBytes  49.9 Mbits/sec
[  3]  1.0- 2.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  2.0- 3.0 sec  5.65 MBytes  47.4 Mbits/sec
[  3]  3.0- 4.0 sec  5.99 MBytes  50.3 Mbits/sec
[  3]  4.0- 5.0 sec  5.94 MBytes  49.8 Mbits/sec
[  3]  5.0- 6.0 sec  5.95 MBytes  49.9 Mbits/sec
[  3]  6.0- 7.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  7.0- 8.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  8.0- 9.0 sec  5.96 MBytes  50.0 Mbits/sec
[  3]  9.0-10.0 sec  5.95 MBytes  49.9 Mbits/sec
[  3]  0.0-10.0 sec  59.3 MBytes  49.7 Mbits/sec
[  3] Sent 42307 datagrams
read failed: Connection refused
[  3] WARNING: did not receive ack of last datagram after 2 tries.
mininet>
```

รูปที่ 5.31 แสดงการทดลองส่งทราฟฟิกจาก Host 3 ไปยัง Host 10

อุปกรณ์คอนโทรลเลอร์ จะเรียกดูค่าข้อมูลปริมาณการใช้งานทราฟฟิกของพอร์ต 7 และ พอร์ต 8 บนสวิตช์ S5 โดยจะแสดงเป็นค่าปริมาณทราฟฟิก ในหน่วยเมกะบิตต่อวินาที และแสดงเป็นจำนวนเปอร์เซ็นต์การใช้งาน สามารถแสดงได้ดังรูปที่ 5.32

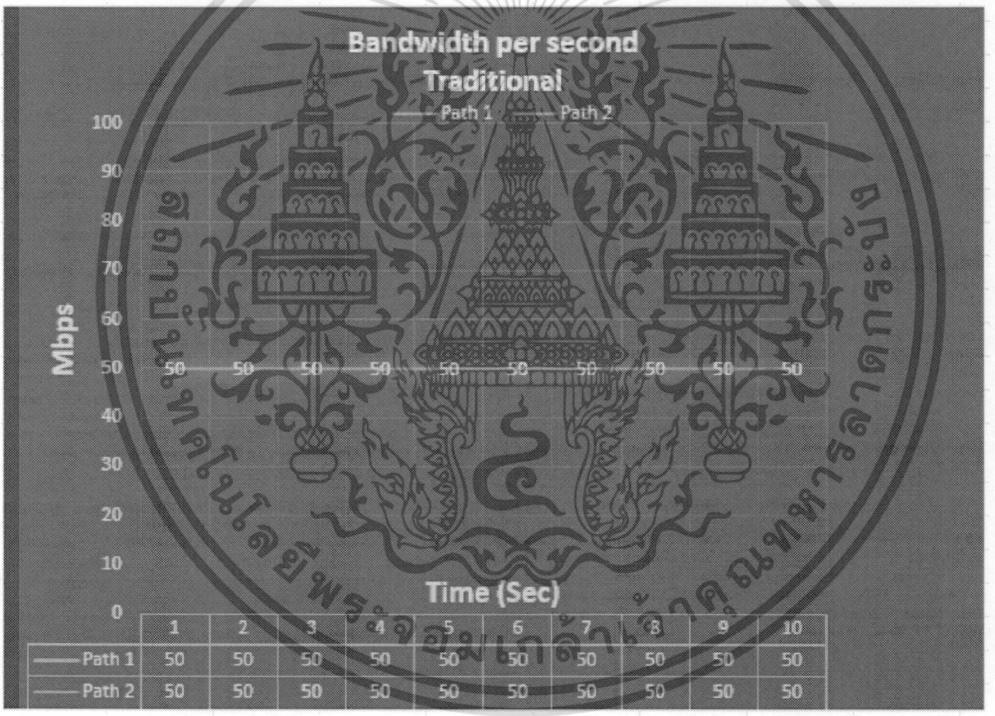
```
S5-port_E7(BW:100 Mbps) -> S6:  50.01 Mbps used 50.01 %
S5-port_E6(BW:100 Mbps) -> S4:  50.99 Mbps used 50.99 %

S5-port_E7(BW:100 Mbps) -> S6:  48.15 Mbps used 48.15 %
S5-port_E6(BW:100 Mbps) -> S4:  50.19 Mbps used 50.19 %

S5-port_E7(BW:100 Mbps) -> S6:  48.70 Mbps used 48.70 %
S5-port_E6(BW:100 Mbps) -> S4:  46.69 Mbps used 46.69 %

S5-port_E7(BW:100 Mbps) -> S6:  34.99 Mbps used 34.99 %
S5-port_E6(BW:100 Mbps) -> S4:  34.91 Mbps used 34.91 %

S5-port_E7(BW:100 Mbps) -> S6:  48.55 Mbps used 48.55 %
S5-port_E6(BW:100 Mbps) -> S4:  49.09 Mbps used 49.09 %
```

รูปที่ 5.32 แสดงการเรียกดูค่าปริมาณการใช้งานทราฟฟิก จากคอนโทรลเลอร์ไปที่ สวิตช์ S5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีเปรียบเทียบปริมาณทราฟฟิกที่สามารถส่งได้ต่อวินาที ระหว่างเครือข่าย SDN และเครือข่ายปัจจุบันพบว่า ผลลัพธ์ที่ได้มีค่าใกล้เคียงกัน เนื่องจากขนาดของช่องทางที่ใช้ในการส่งทราฟฟิกสามารถรองรับต่อปริมาณของทราฟฟิกได้ และสำหรับกรณีเครือข่ายที่ทำงานด้วยระบบ SDN การใช้งานทราฟฟิกบนเส้นทางที่ 1 มีปริมาณไม่หนาแน่น มีปริมาณไม่ถึงค่าที่มีการกำหนดไว้ ทำให้การไหลของทราฟฟิกเหมือนกับ ระบบเครือข่ายที่ทำงานในปัจจุบัน

กราฟแสดงถึงผลการทดลองที่ 5.3.1 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่ายปัจจุบันเมื่อความสามารถในการส่งทราฟฟิกของระบบเครือข่ายปัจจุบัน คือ 100 เมกะบิตต่อวินาที ความต้องการในการส่งทราฟฟิก อยู่ที่ 50 เมกะบิตต่อวินาที จำนวน 10 วินาที ส่งผลให้ระบบเครือข่ายปัจจุบันสามารถส่งข้อมูลเสร็จสิ้นใช้ระยะเวลา 10 วินาที โดยต่อวินาที ระบบใช้เวลาในการส่งทราฟฟิก 0.5 วินาที สามารถแสดงได้ดังรูปที่ 5.33



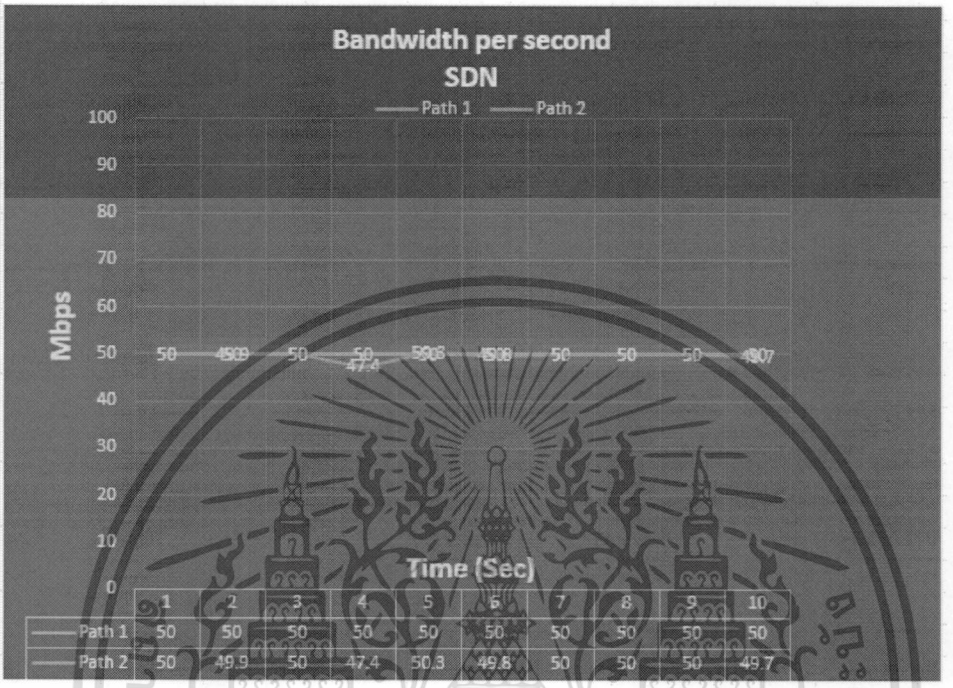
	1	2	3	4	5	6	7	8	9	10
Path 1	50	50	50	50	50	50	50	50	50	50
Path 2	50	50	50	50	50	50	50	50	50	50

รูปที่ 5.33 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.1

กราฟแสดงถึงผลการทดลองที่ 5.3.1 การส่งทราฟฟิกต่อหน่วยเวลา บนระบบเครือข่ายที่ทำงานด้วย SDN เมื่อความสามารถในการส่งทราฟฟิกของระบบเครือข่าย SDN คือ 100 เมกะบิตต่อวินาที ความต้องการ ในการส่งทราฟฟิก อยู่ที่ 50 เมกะบิตต่อวินาที จำนวน 10 วินาที ส่งผลให้ระบบเครือข่าย SDN สามารถส่งข้อมูลเสร็จสิ้นเป็น ระยะเวลา 10 วินาที โดยต่อวินาที ระบบใช้เวลาในการส่งทราฟฟิก 0.5 วินาที ในการทำงานของคอนโทรลเลอร์ จะตรวจสอบปริมาณทราฟฟิกพบว่ามี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

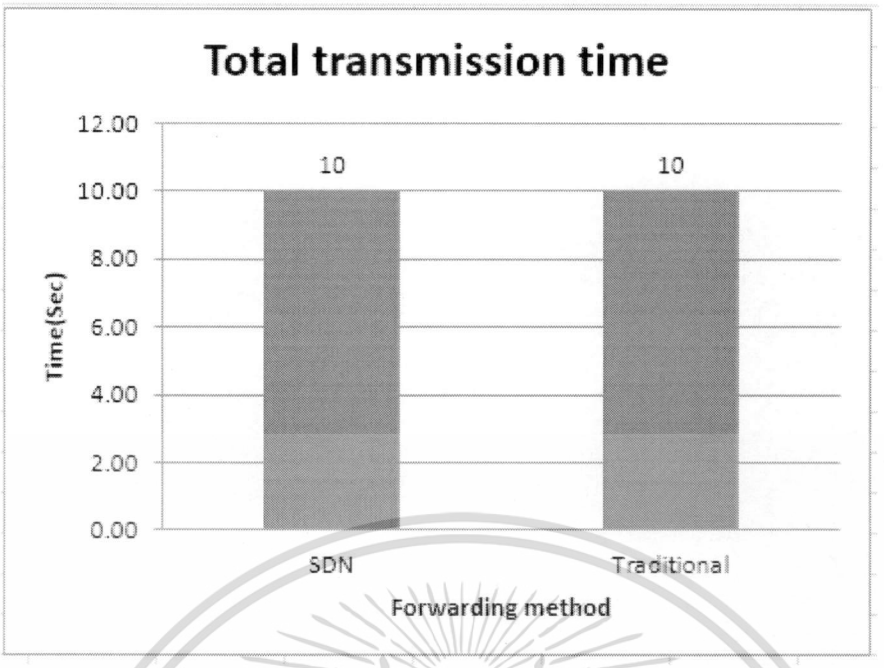
ปริมาณน้อยกว่า 80 เปอร์เซ็นต์ ไม่เข้าเงื่อนไขที่กำหนดไว้ในการแบ่งทราฟฟิก ส่งผลให้การส่งทราฟฟิกทำงานในรูปแบบปกติเช่นเดียวกับระบบเครือข่ายในปัจจุบัน สามารถแสดงได้ดังรูปที่ 5.34



รูปที่ 5.34 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.1

จากการทดสอบส่งทราฟฟิกบนเครือข่ายที่ทำงานด้วยระบบ SDN และบนระบบเครือข่ายที่ทำงานด้วยระบบปัจจุบัน ผลที่ได้ออกมา ไม่แตกต่างกัน ใช้เวลาในการส่งเป็นเวลา 10 วินาที เนื่องด้วยปริมาณความต้องส่งทราฟฟิกมีค่าน้อยกว่าความสามารถในการส่งทราฟฟิก ส่งผลให้ไม่มีความหนาแน่นเกิดขึ้นในระบบ และระบบเครือข่ายสามารถส่งข้อมูลทราฟฟิกได้ทั้งหมดในแต่ละวินาที โดยไม่มีการต้องรอคิวในการส่ง สามารถแสดงได้ดังรูปที่ 5.35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

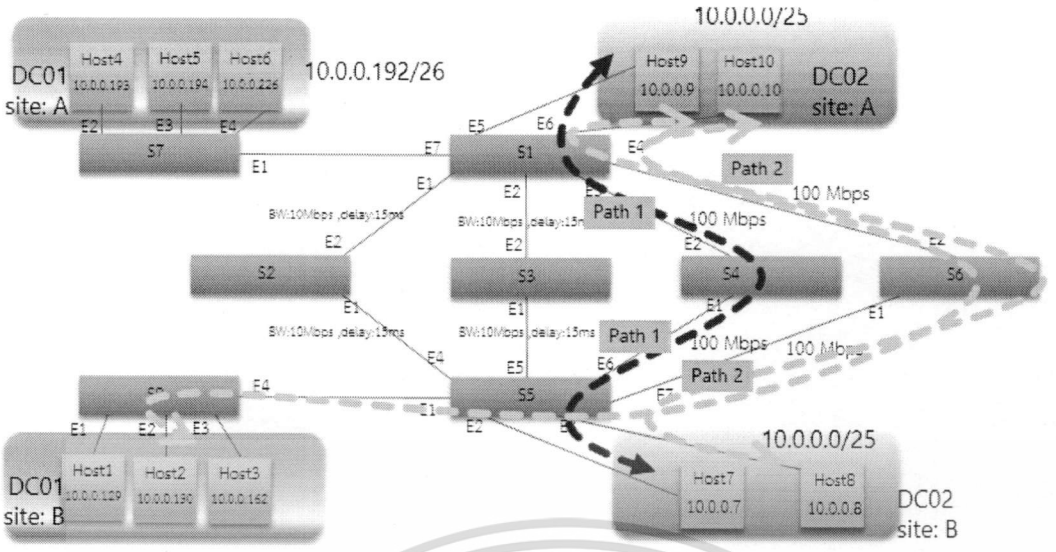


รูปที่ 5.35 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.1

5.3.2. คอนโทรลเลอร์แบ่งทราฟฟิก เมื่อเส้นทางที่ 1 มีการใช้งานมากกว่า 80% และปริมาณทราฟฟิกที่ต้องการส่งผ่านช่องทางที่ 1 มีน้อยกว่าความสามารถในการส่งทราฟฟิก

คอนโทรลเลอร์ตรวจสอบปริมาณการใช้งานของทราฟฟิก และเมื่อพบว่ามีการใช้งานทราฟฟิกหนาแน่นบนเส้นทางที่ 1 คือ มีปริมาณมากกว่า 80 เปอร์เซ็นต์ ของความสามารถในการส่งทราฟฟิก โดยเมื่อพิจารณาจากเครือข่ายที่ได้จำลองขึ้นมา ค่าการส่งทราฟฟิกที่มากกว่า 80 เมกะบิตต่อวินาที ถือว่าเส้นทางการส่งทราฟฟิกมีความหนาแน่น หรือมีโอกาสเกิดความหนาแน่นสูง จากนั้นจะดำเนินการแบ่งทราฟฟิก ไปยังเส้นทางที่ 2 โดยเงื่อนไขในการแบ่งทราฟฟิกไปยังเส้นทางที่ 2 คือ พิจารณาจาก IP address ต้นทาง โดยมีการกำหนดเงื่อนไขว่า ทราฟฟิกที่มาจาก Host 8 จะทำการส่งไปยังเส้นทางที่ 2 โดย ทราฟฟิกที่มาจาก Host 7 ยังคงส่งผ่านทางช่องทางที่ 1 และทราฟฟิกระหว่าง DC01 Site B และ DC02 Site A ยังคงมีการส่งผ่านเส้นทางที่ 2 สามารถแสดงได้ดังรูปที่ 5.36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.36 การไหลของทราฟฟิกเมื่อเกิดการแบ่งทราฟฟิกจากคอนโทรลเลอร์

ในการทดลองจะมีรูปแบบในการส่งทราฟฟิก ดังต่อไปนี้

5.3.2.1.ทราฟฟิกจาก DC02 = 90 เมกะบิตต่อวินาที และ ทราฟฟิกจาก DC01 = 50 เมกะบิตต่อวินาที สัดส่วนทราฟฟิก รูปแบบที่ 1

1) ทราฟฟิกของ DC02 = 90 เมกะบิตต่อวินาที ประกอบไปด้วย Host 7 = 90 เมกะบิตต่อวินาที, Host 8 ไม่มีการส่งทราฟฟิก

2) ทราฟฟิกของ DC01 = 50 เมกะบิตต่อวินาที ประกอบไปด้วย Host 3 = 50 เมกะบิตต่อวินาที

ความสามารถในการส่งทราฟฟิกของเส้นทางที่ 1 และ 2 มีขนาดของแต่ละช่องทางอยู่ที่ 100 เมกะบิตต่อวินาที โดยสามารถแสดงสัดส่วนของข้อมูลทราฟฟิก ในแต่ละเส้นทาง เมื่อมีการส่งทราฟฟิบนระบบเครือข่ายปัจจุบัน ได้ดังรูปที่ 5.37

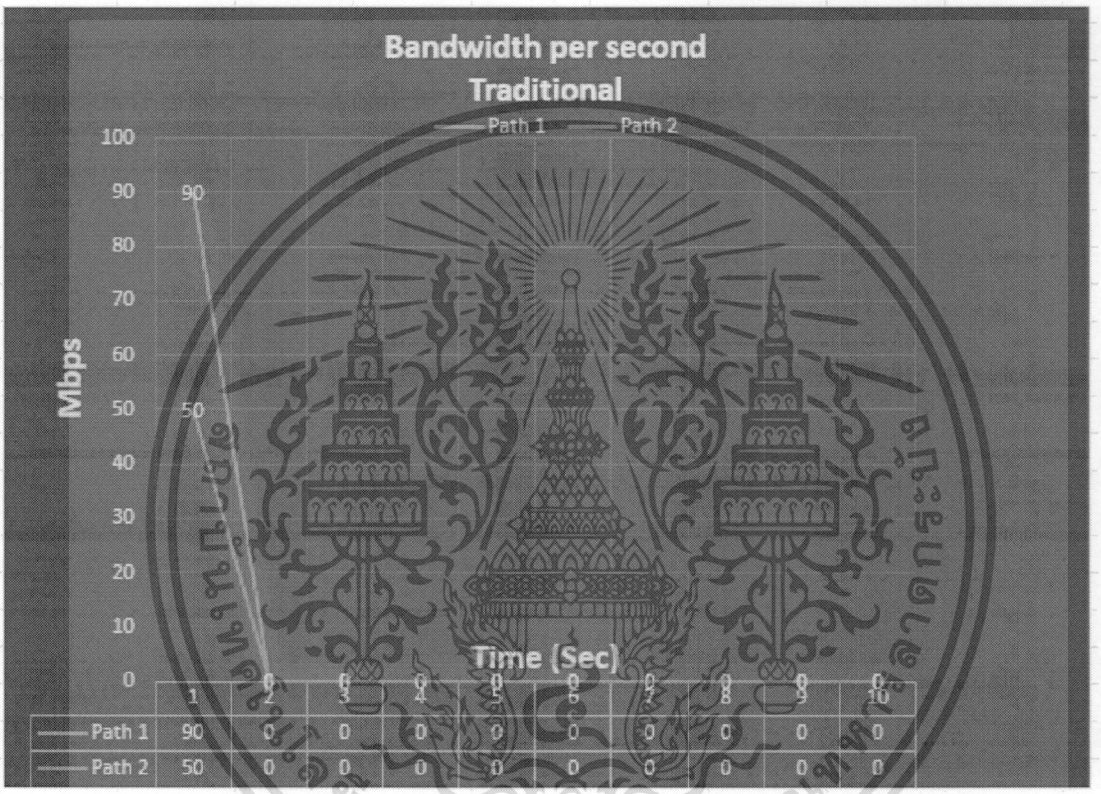
Traditional								
Path 1					Path2			
Host.7	Host.8	total	capacity	Time(sec)	DC01	total	capacity	Time(sec)
90	0	90	100	0.9	50	50	100	0.5

รูปที่ 5.37 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.2.1

กราฟแสดงถึงผลการทดลองที่ 5.3.2.1 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่ายปัจจุบัน ความต้องการในการส่งทราฟฟิกบนเส้นทางที่ 1 มีอยู่ที่ 90 เมกะบิตต่อวินาที เมื่อพิจารณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 90 /100 เมกะบิตต่อวินาที = 0.9 วินาที และ ความต้องการ ในการส่งทราฟฟิก บนเส้นทางที่ 2 อยู่ที่ 50 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 50 /100 เมกะบิตต่อวินาที = 0.5 วินาที จากการส่งข้อมูลทั้ง 2 เส้นทางพร้อมๆกันส่งผลให้ระบบเครือข่ายที่ทำงานในรูปแบบปัจจุบัน สามารถส่งข้อมูลเสร็จสิ้น ใช้เวลา 0.9 วินาที แสดงได้ดังรูปที่ 5.38



รูปที่ 5.38 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.2.1

ทดสอบส่งทราฟฟิกแบบเดียวกัน บนระบบเครือข่ายที่ทำงานด้วย SDN จากเงื่อนไขการส่งทราฟฟิก จะส่งผลให้ปริมาณการใช้งานทราฟฟิกบนเส้นทางที่ 1 มีปริมาณ มากกว่า 80 เมกะบิตต่อวินาที ตรงตามเงื่อนไขบนคอลโทรลเลอร์ ทำให้เกิดการสั่งงาน แบ่งทราฟฟิกของ Host 8 ไปยังเส้นทางที่ 2 ร่วมกับการส่งทราฟฟิกของ DC01 ดังแสดงตามรูปที่ 5.39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SDN								
Path 1				Path 2				
Host.7	total	capacity	Time(sec)	Host.8	DC01	total	capacity	Time(sec)
90	90	100	0.9	0	50	50	100	0.5

รูปที่ 5.39 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.2.1

ทดสอบส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 จำนวน 1 แพ็กเก็ต โดยมีขนาด 11.25M คือ 11.25 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 11.25x8 = 90 เมกะบิตต่อวินาที และมีการกำหนดให้แสดงผลเวลาที่ใช้ในการส่งด้วยการกำหนดค่า –i ให้มีค่าเท่ากับ 1 คือ แสดงผลทุกๆรอบ 1 วินาที จากการทดสอบ ดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 0.9 วินาที และในการทดลองครั้งนี้เนื่องด้วยเป็นการทดลองบนระบบเครือข่ายจำลอง อาจมีค่าผลลัพธ์ในบางส่วนที่มีความผิดพลาดบ้าง อันเนื่องจากปัจจัยด้านประสิทธิภาพในการทำงานของเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผล ดังแสดงตามรูปที่ 5.40

```
mininet>
mininet> h7 iperf -c h9 -t 10 -i 1 -n 11.25M
------------------------------------------------------------
Client connecting to 10.0.0.9, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.7 port 47205 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.9 sec  11.2 MBytes   111 Mbits/sec
mininet>
mininet>
```

รูปที่ 5.40 การทดลองส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 ตามเงื่อนไขที่ 5.3.2.1

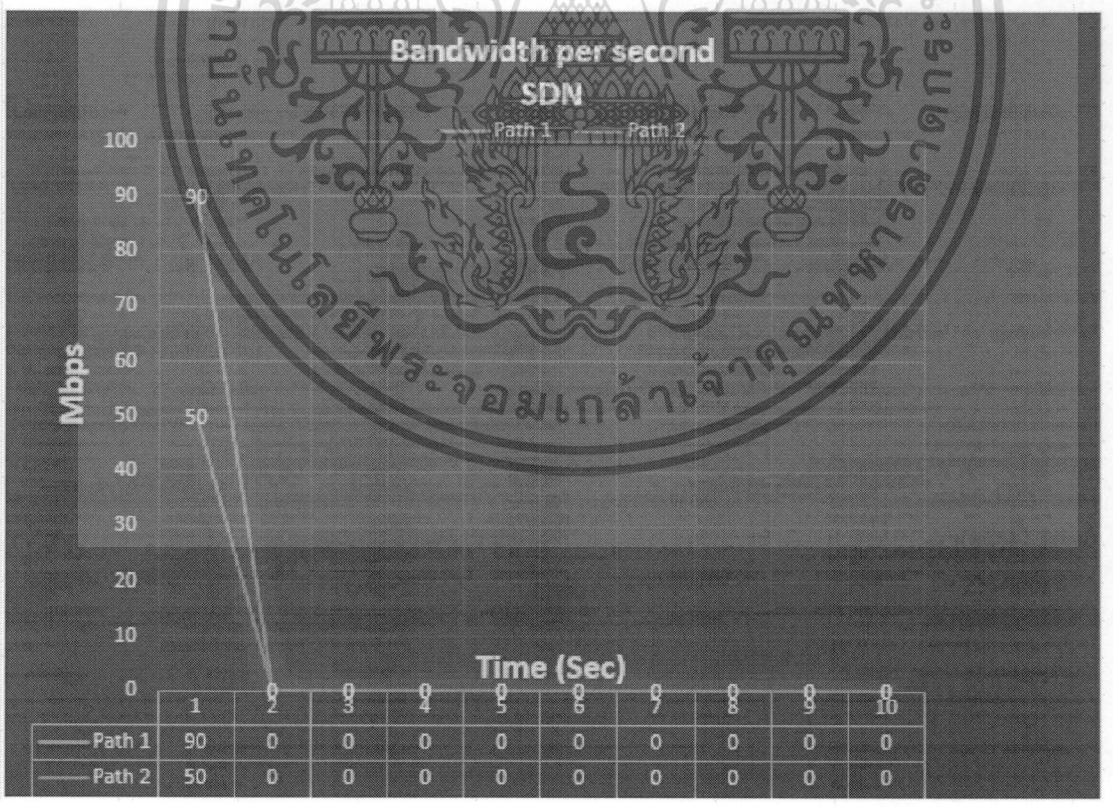
ทดสอบส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 จำนวน 1 แพ็กเก็ต โดยมีขนาด 6.25M คือ 6.25 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 6.25x8 = 50 เมกะบิตต่อวินาที และมีการกำหนดให้แสดงผลเวลาที่ใช้ในการส่ง ด้วยการกำหนดค่า –i ให้มีค่าเท่ากับ 1 คือ แสดงผลทุกๆรอบ 1 วินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 0.5 วินาที ดังแสดงตามรูปที่ 5.41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mininet>
mininet> h3 iperf -c h10 -t 10 -i 1 -n 6.25M
------------------------------------------------------------
Client connecting to 10.0.0.10, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 41992 connected with 10.0.0.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.5 sec  6.25 MBytes   102 Mbits/sec
mininet>
```

รูปที่ 5.41 การทดลองส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 ตามเงื่อนไขที่ 5.3.2.1

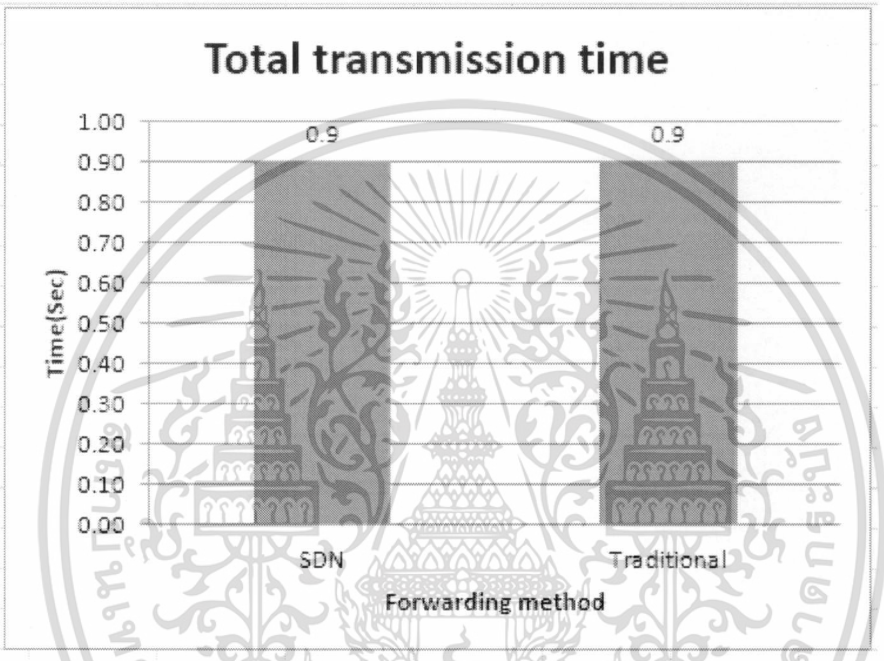
กราฟแสดงถึงผลการทดลองที่ 5.3.2.1 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่าย SDN โดยมีความต้องการในการส่งทราฟฟิกบนเส้นทางที่ 1 อยู่ที่ 90 เมกะบิตต่อวินาที บนเส้นทางที่ 2 อยู่ที่ 50 เมกะบิตต่อวินาที ส่งผลให้เข้าเงื่อนไขที่กำหนดไว้ที่คอลโทรลเลอร์ ทำให้มีการแบ่งทราฟฟิกจาก Host 8 ให้ไปทางเส้นทางที่ 2 โดยในกรณีนี้พบว่าที่ Host 8 ไม่มีการส่งทราฟฟิกเลย จึงส่งผลให้การไหลของทราฟฟิกบนระบบเครือข่าย SDN ไม่มีการเปลี่ยนแปลง ดังแสดงตามรูปที่ 5.42



รูปที่ 5.42 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่าย SDN ในการทดลองที่ 5.3.2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการสอบส่งทราฟฟิกบนเครือข่ายที่ทำงานด้วยระบบ SDN และบนระบบเครือข่ายที่ทำงานด้วยระบบปัจจุบัน ผลลัพธ์ที่ได้ คือ ไม่มีความแตกต่างกัน ใช้เวลาในการส่งเป็นเวลา 0.9 วินาที เนื่องด้วยเงื่อนไขในการแบ่งทราฟฟิกของคอนโทลเลอร์ทำงานในรูปแบบของ IP address ต้นทาง ซึ่งในกรณีนี้การส่งทราฟฟิกมาจาก Host7 เท่านั้น เมื่อเกิดการแบ่งทราฟฟิกไปยังเส้นทางที่ 2 ของ Host 8 ซึ่งไม่มีการส่งทราฟฟิกจึงไม่มีผลต่อระบบเครือข่าย SDN บนเส้นทางที่ 1 และ 2 ส่งผลให้ระยะเวลาในการส่งข้อมูลทราฟฟิกยังคงเดิม สามารถแสดงได้ดังรูปที่ 5.43



รูปที่ 5.43 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.2.1

5.3.2.2.ทราฟฟิกจาก DC02 = 90 เมกะบิตต่อวินาที และ ทราฟฟิกจาก DC01 = 50 เมกะบิตต่อวินาที สัดส่วน ทราฟฟิก รูปแบบที่ 2

1) ทราฟฟิกของ DC02 = 90 เมกะบิตต่อวินาที ประกอบไปด้วย Host 7 = 50 เมกะบิตต่อวินาที, Host 8 = 40 เมกะบิตต่อวินาที

2) ทราฟฟิกของ DC01 = 50 เมกะบิตต่อวินาที ประกอบไปด้วย Host 3 = 50 เมกะบิตต่อวินาที

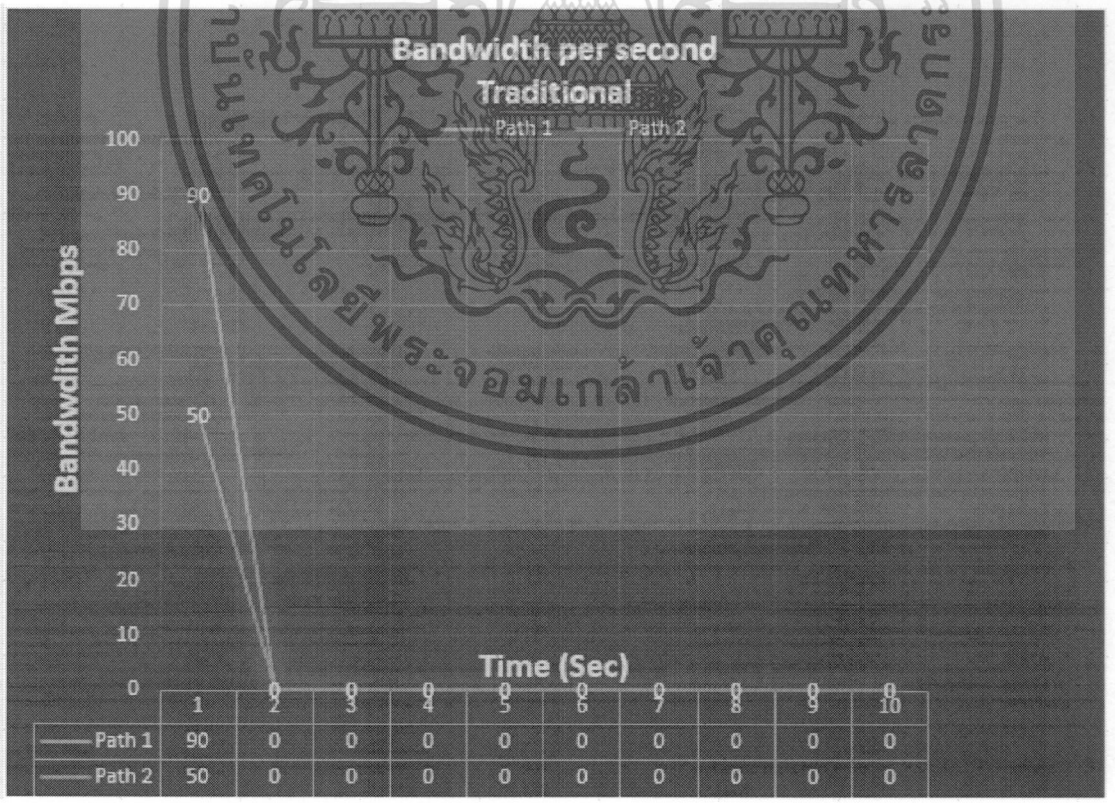
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถในการส่งทราฟฟิกของเส้นทางที่ 1 และ 2 มีขนาดของแต่ละช่องทางอยู่ที่ 100 เมกะบิตต่อวินาที โดยสามารถแสดงสัดส่วนของ ข้อมูลของทราฟฟิก ในแต่ละเส้นทาง เมื่อมีการส่งทราฟฟิกบนระบบเครือข่ายปัจจุบัน ได้ดังรูปที่ 5.44

	Traditional								
	Path 1					Path 2			
	Host.7	Host.8	total	capacity	Time(sec)	DC01	total	capacity	Time(sec)
2	50	40	90	100	0.9	50	50	100	0.5

รูปที่ 5.44 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.2.2

กราฟแสดงถึงผลการทดลองที่ 5.3.2.2 มีการพิจารณาถึงสัดส่วนของทราฟฟิกที่มีการเปลี่ยนแปลง การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่ายปัจจุบัน ยังคงเดิม เช่นเดียวกับการทดลองที่ 5.3.2.1 อันเนื่องมาจาก ผลรวมของสัดส่วนทราฟฟิก มีค่าคงเดิม ส่งผลให้ระยะเวลาสูงสุดในการส่งข้อมูลทราฟฟิก คงเดิมที่ 0.9 วินาที บนเส้นทางที่ 1 สามารถแสดงได้ดังรูปที่ 5.45



รูปที่ 5.45 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบส่งทราฟฟิกแบบเดียวกัน บนระบบเครือข่ายที่ทำงานด้วย SDN จากเงื่อนไขการส่งทราฟฟิก จะส่งผลให้ปริมาณการใช้งานบนเส้นทางที่ 1 มีปริมาณ มากกว่า 80 เมกะบิตต่อวินาที ตรงตามเงื่อนไขบนคอลโทรลเลอร์ ทำให้เกิดการสั่งงานไปที่อุปกรณ์สวิตซ์ เพื่อทำการแบ่งทราฟฟิกของ Host 8 ไปยังเส้นทางที่ 2 ร่วมกับการส่งทราฟฟิกของ DC01 ส่งผลให้จำนวนทราฟฟิกที่ต้องส่งผ่านเส้นทางที่ 1 มีค่า 50 เมกะบิตต่อวินาที และ บนเส้นทางที่ 2 มีค่า 90 เมกะบิตต่อวินาที ดังแสดงตามรูปที่ 5.46

SDN								
Path 1				Path 2				
Host.7	total	capacity	Time(sec)	Host.8	DC01	total	capacity	Time(sec)
50	50	100	0.5	40	50	90	100	0.9

รูปที่ 5.46 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.2.2

ทดสอบส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 จำนวน 1 แพ็กเก็ต โดยมีขนาด 6.25M คือ 6.25 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 6.25x8 = 50 เมกะบิตต่อวินาที และมีการกำหนดให้แสดงผลเวลาที่ใช้ในการส่ง ด้วยการกำหนดค่า –i ให้มีค่าเท่ากับ 1 คือ แสดงผลทุกๆ รอบ 1 วินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 0.5 วินาที ดังแสดงตามรูปที่ 5.47

```
mininet>
mininet> h7 iperf -c h9 -t 10 -i 1 -n 6.25M
------------------------------------------------------------
Client connecting to 10.0.0.9, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.7 port 47223 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.5 sec  6.25 MBytes  95.8 Mbits/sec
mininet>
mininet>
```

รูปที่ 5.47 การทดลองส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 ตามเงื่อนไขที่ 5.3.2.2

ทดสอบส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 จำนวน 1 แพ็กเก็ต โดยมีขนาด 11.25M คือ 11.25 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 11.25x8 = 90 เมกะบิตต่อวินาที และมีการกำหนดให้แสดงผลเวลาที่ใช้ในการส่ง ด้วยการกำหนดค่า –i ให้มีค่าเท่ากับ 1 คือ แสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

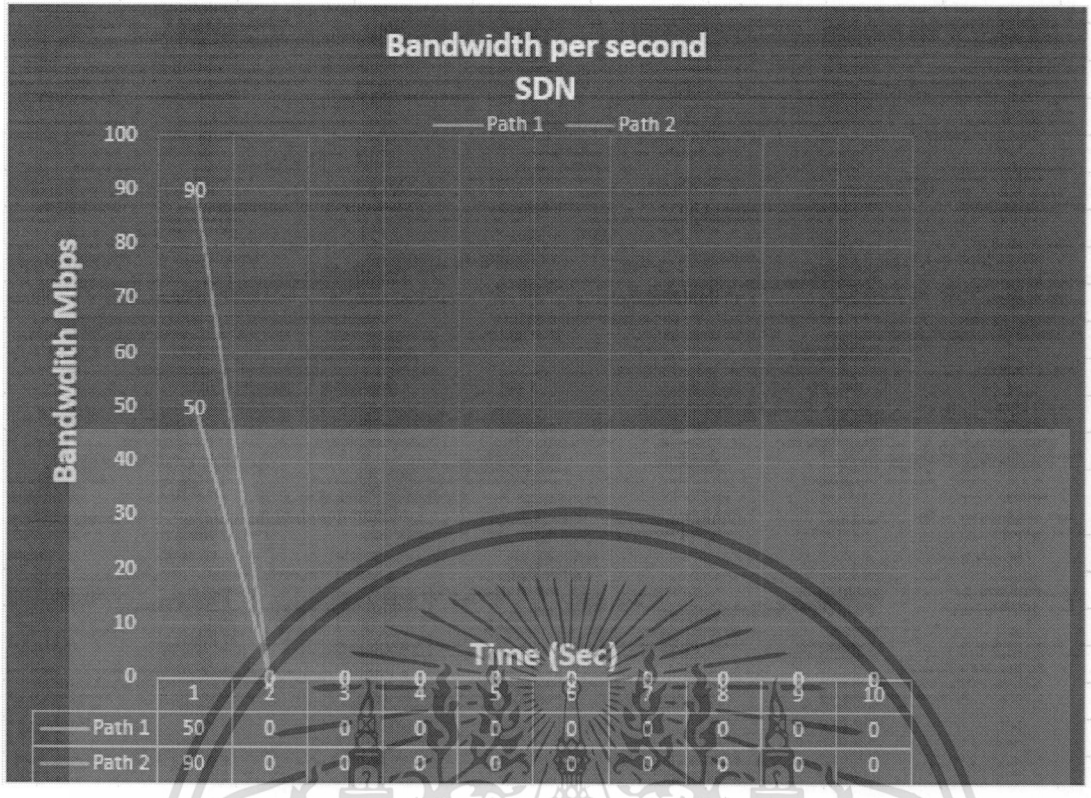
ทุกๆรอบ 1 วินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 0.9 วินาที ดังแสดงตามรูปที่ 5.48

```
mininet>
mininet>
mininet> h3 iperf -c h10 -t 1 -i 1 -n 11.25M
------------------------------------------------------------
Client connecting to 10.0.0.10, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 42026 connected with 10.0.0.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.9 sec  11.2 MBytes   111 Mbits/sec
mininet>
```

รูปที่ 5.48 การทดลองส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 ตามเงื่อนไขที่ 5.3.2.2

กราฟแสดงถึงผลการทดลองที่ 5.3.2.2 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่าย SDN โดยมีความต้องการในการส่งทราฟฟิกบนเส้นทางที่ 1 อยู่ที่ 90 เมกะบิตต่อวินาที บนเส้นทางที่ 2 อยู่ที่ 50 เมกะบิตต่อวินาที ส่งผลให้เข้าเงื่อนไขที่กำหนดไว้ที่คอลโทรลเลอร์ ทำให้มีการแบ่งทราฟฟิกจาก Host 8 ให้ไปทางเส้นทางที่ 2 โดยในกรณีนี้พบว่าที่ Host 8 มีการส่งทราฟฟิกจำนวน 40 เมกะบิตต่อวินาที ส่งผลให้ปริมาณทราฟฟิกบนเส้นทางที่ 1 มีค่า 50 เมกะบิตต่อวินาที และ บนเส้นทางที่ 2 มีค่า 90 เมกะบิตต่อวินาที ส่งผลให้ระยะเวลามากสุดในการส่งข้อมูลทราฟฟิก ยังคงเดิมที่ 0.9 วินาที บนเส้นทางที่ 2 สามารถแสดงได้ดังรูปที่ 5.49

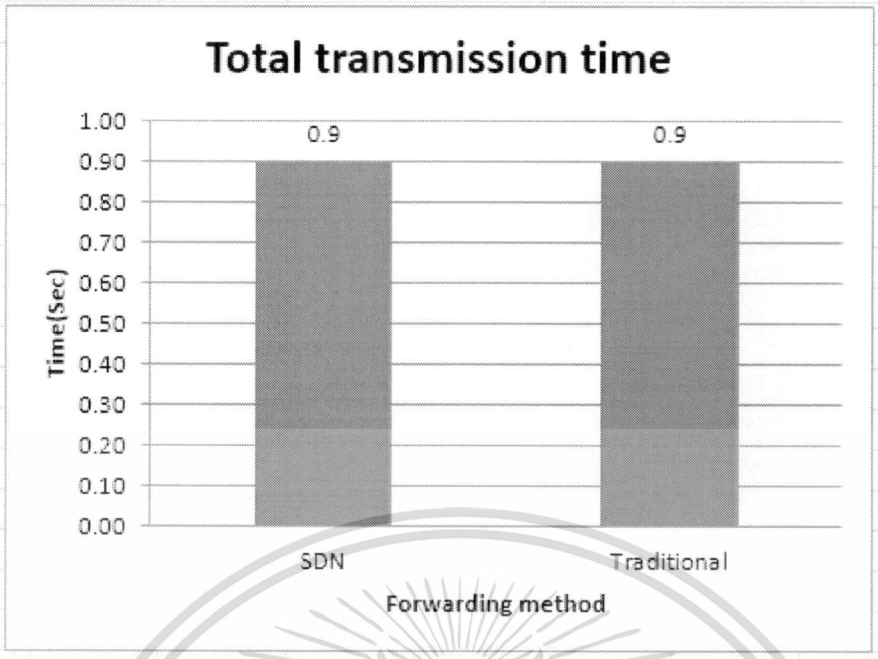
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.49 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่าย SDN ในการทดลองที่ 5.3.2.2

จากการทดสอบส่งทราฟฟิกบนเครือข่ายที่ทำงานด้วยระบบ SDN และบนระบบเครือข่ายที่ทำงานด้วยระบบปัจจุบัน ผลที่ได้ออกมาไม่มีความแตกต่างกัน ใช้เวลามากสุดในการส่งเป็นเวลา 0.9 วินาที เนื่องด้วยเงื่อนไขในการแบ่งทราฟฟิกของคอนโทลเลอร์ทำงานในรูปแบบของเงื่อนไข IP address ต้นทาง ซึ่งในกรณีนี้การแบ่งทราฟฟิกมาจาก Host8 ไปยังเส้นทางที่ 2 ส่งผลให้ปริมาณทราฟฟิกบนเส้นทางที่ 1 ลดลง และปริมาณทราฟฟิกบนเส้นทางที่ 2 เพิ่มขึ้น แต่ยังน้อยกว่าความสามารถในการส่งทราฟฟิกของเส้นทางที่ 2 และเมื่อพิจารณาถึงผลรวมของทราฟฟิกเมื่อมีการแบ่งทราฟฟิกจาก Host8 ไปบนเส้นทางที่ 2 จะทำให้ปริมาณการส่งทราฟฟิกบนเส้นทางที่ 2 ของระบบเครือข่าย SDN มีค่าเท่ากับปริมาณการส่งทราฟฟิกบนเครือข่ายปัจจุบันบนเส้นทางที่ 1 จึงทำให้ได้ผลของระยะเวลาในการส่งข้อมูล มีค่าเท่ากันที่ 0.9 วินาที สามารถแสดงได้ดังรูปที่ 5.50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.50 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.2.2

5.3.2.3.ทราฟฟิกจาก DC02 = 90 เมกะบิตต่อวินาที และ ทราฟฟิกจาก DC01 = 50 เมกะบิตต่อวินาที สัดส่วนทราฟฟิก รูปแบบที่ 3

1) ทราฟฟิกของ DC02 = 90 เมกะบิตต่อวินาที ประกอบไปด้วย Host 7 = 30 เมกะบิตต่อวินาที, Host 8 = 60 เมกะบิตต่อวินาที

2) ทราฟฟิกของ DC01 = 50 เมกะบิตต่อวินาที ประกอบไปด้วย Host 3 = 50 เมกะบิตต่อวินาที

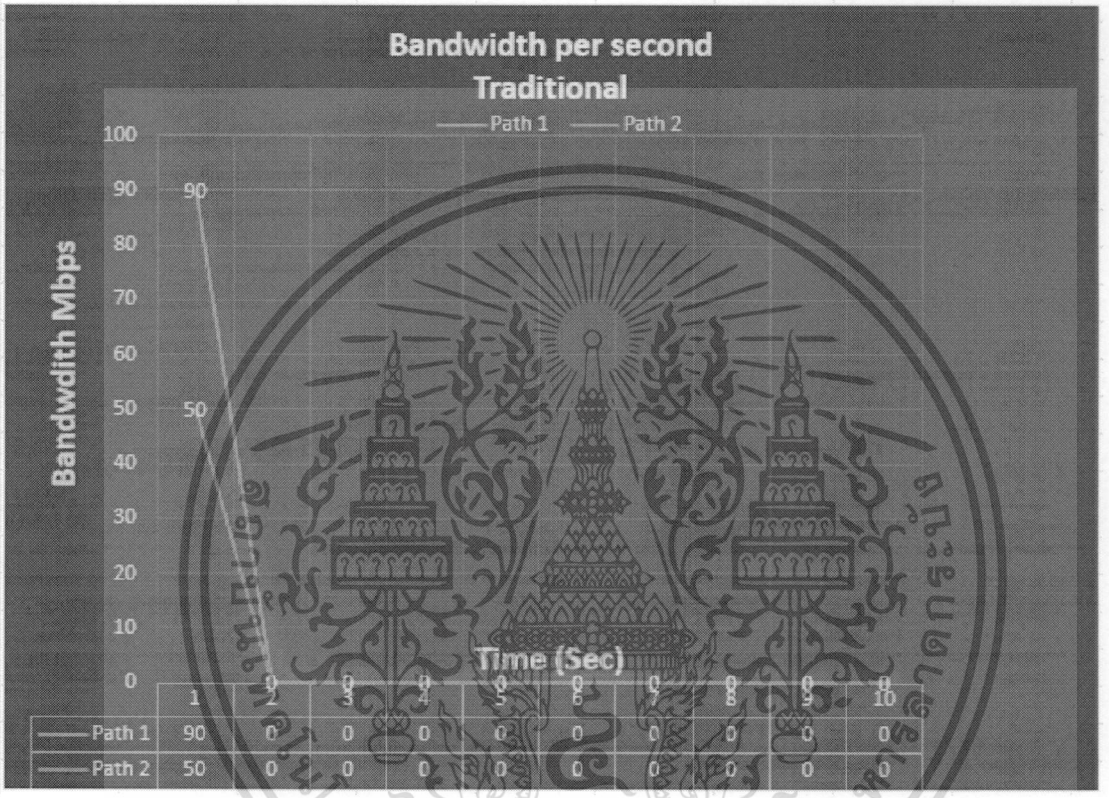
ความสามารถในการส่งทราฟฟิกของเส้นทางที่ 1 และ 2 มีขนาดของแต่ละช่องทางอยู่ที่ 100 เมกะบิตต่อวินาที โดยสามารถแสดงสัดส่วนของ ข้อมูลของทราฟฟิก ในแต่ละเส้นทาง เมื่อมีการส่งทราฟฟิบนระบบเครือข่ายปัจจุบัน ได้ดังรูปที่ 5.51

	Traditional								
	Path 1					Path 2			
	Host.7	Host.8	total	capacity	Time(sec)	DC01	total	capacity	Time(sec)
3	30	60	90	100	0.9	50	50	100	0.5

รูปที่ 5.51 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟแสดงถึงผลการทดลองที่ 5.3.2.3 มีการพิจารณาถึงสัดส่วนของทราฟฟิกที่มีการเปลี่ยนแปลงการส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่ายปัจจุบันยังคงเดิม เช่นเดียวกับการทดลองที่ 5.3.2.1 และ 5.3.2.2 อันเนื่องมาจาก ผลรวมของสัดส่วนทราฟฟิก มีค่าคงที่ ส่งผลให้ระยะเวลามากสุดในการส่งข้อมูลทราฟฟิก ยังคงเดิมที่ 0.9 วินาที บนเส้นทางที่ 1 สามารถแสดงได้ดังรูปที่ 5.52



รูปที่ 5.52 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.2.3

ทดสอบส่งทราฟฟิกแบบเดียวกันบนระบบเครือข่ายที่ทำงานด้วย SDN จากเงื่อนไขการส่งทราฟฟิก จะส่งผลให้ปริมาณการใช้งานทราฟฟิกบนเส้นทางที่ 1 มีปริมาณ มากกว่า 80 เมกะบิตต่อวินาที ตรงตามเงื่อนไขบนคอลโทรลเลอร์ ทำให้เกิดการสั่งงาน แบ่งทราฟฟิกของ Host 8 ไปยังเส้นทางที่ 2 ร่วมกับการส่งทราฟฟิกของ DC01 ส่งผลให้จำนวนทราฟฟิกที่ต้องส่งผ่านเส้นทางที่ 1 มีค่า 30 เมกะบิตต่อวินาที และ บนเส้นทางที่ 2 มีค่า 110 เมกะบิตต่อวินาที ดังแสดงตามรูปที่ 5.53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SDN								
Path 1				Path 2				
Host.7	total	capacity	Time(sec)	Host.8	DC01	total	capacity	Time(sec)
30	30	100	0.3	60	50	110	100	1.1

รูปที่ 5.53 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.2.3

ทดสอบส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 จำนวน 1 แพ็กเก็ต โดยมีขนาด 3.75M คือ 3.75 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 3.75x8 = 30 เมกะบิตต่อวินาที และมีการกำหนดให้แสดงผลเวลาที่ใช้ในการส่ง ด้วยการกำหนดค่า –i ให้มีค่าเท่ากับ 1 คือ แสดงผลทุกๆ รอบ 1 วินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 0.5 วินาที ดังแสดงตามรูปที่ 5.54

```
mininet>
mininet> h7 iperf -c h9 -t 10 -i 1 -n 3.75M
------------------------------------------------------------
Client connecting to 10.0.0.9, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.7 port 47207 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.3 sec  3.75 MBytes   123 Mbits/sec
mininet>
```

รูปที่ 5.54 การทดลองส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 ตามเงื่อนไขที่ 5.3.2.3

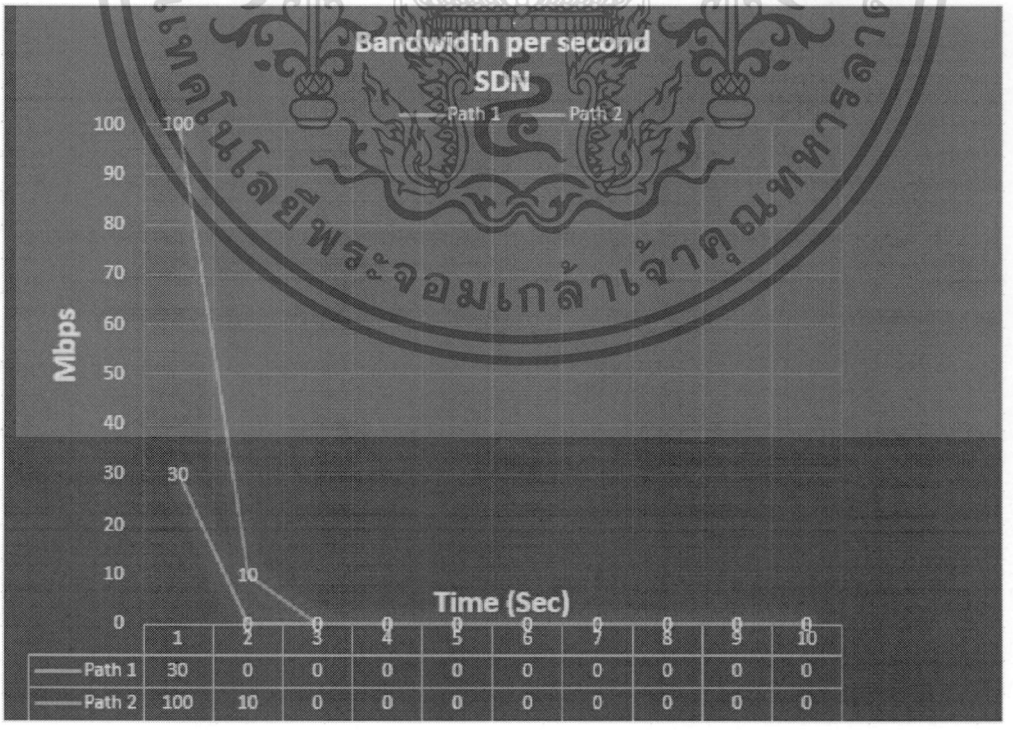
ทดสอบส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 จำนวน 1 แพ็กเก็ต โดยมีขนาด 13.75M คือ 13.75 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 13.75x8 = 110 เมกะบิตต่อวินาที ในการทดลองจะส่งทราฟฟิกที่เป็นผลรวมของทั้งหมด และมีการกำหนดให้แสดงผลเวลาที่ใช้ในการส่ง ด้วยการกำหนดค่า –i ให้มีค่าเท่ากับ 1 คือ แสดงผลทุกๆรอบ 1 วินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 1.1 วินาที ดังแสดงตามรูปที่ 5.55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mininet>
mininet> h3 iperf -c h10 -t 1 -i 1 -n 13.75M
------------------------------------------------------------
Client connecting to 10.0.0.10, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 42128 connected with 10.0.0.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  13.5 MBytes   113 Mbits/sec
[  3]  0.0- 1.1 sec  13.8 MBytes   110 Mbits/sec
mininet>
mininet>
```

รูปที่ 5.55 การทดลองส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 ตามเงื่อนไขที่ 5.3.2.3

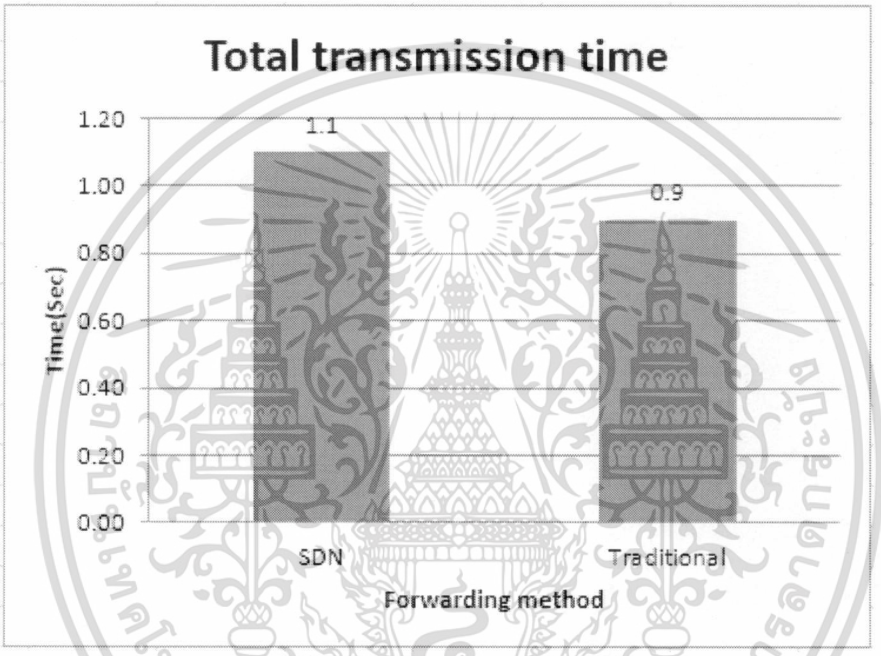
กราฟแสดงถึงผลการทดลองที่ 5.3.2.3 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่าย SDN โดยมีความต้องการ ในการส่งทราฟฟิก บนเส้นทางที่ 1 อยู่ที่ 90 เมกะบิตต่อวินาที บนเส้นทางที่ 2 อยู่ที่ 50 เมกะบิตต่อวินาที ส่งผลให้เข้าเงื่อนไขที่กำหนดไว้ที่คอลโทรลเลอร์ทำให้มีการแบ่งทราฟฟิกจาก Host 8 ให้ไปทางเส้นทางที่ 2 โดยในกรณีนี้พบว่าที่ Host 8 มีการส่งทราฟฟิก จำนวน 60 เมกะบิตต่อวินาที ส่งผลให้ ปริมาณทราฟฟิกบนเส้นทางที่ 1 มีค่า 30 เมกะบิตต่อวินาที และ บนเส้นทางที่ 2 มีค่า 110 เมกะบิตต่อวินาที ส่งผลให้ระยะเวลามากสุดในการส่งข้อมูลทราฟฟิก อยู่ที่ 1.1 วินาที บนเส้นทางที่ 2 สามารถแสดงได้ดังรูปที่ 5.56



รูปที่ 5.56 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่าย SDN ในการทดลองที่ 5.3.2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบส่งทราฟฟิกบนเครือข่ายที่ทำงานด้วยระบบ SDN และบนระบบเครือข่ายที่ทำงานด้วยระบบปัจจุบัน ผลที่ได้พบว่าการทำงานของเครือข่ายปัจจุบันยังคงได้ผลเหมือนเดิม ในขณะที่การทำงานของระบบเครือข่าย SDN ใช้เวลาในการส่งทราฟฟิกเพิ่มขึ้น เนื่องมาจากผลกระทบจากการแบ่งทราฟฟิกของคอนโทรลเลอร์ ด้วยเงื่อนไข IP address ต้นทาง ทำให้ทราฟฟิกของเส้นทางที่ 1 โดยรวมลดลง แต่ส่งผลกระทบต่อเส้นทางที่ 2 เพราะผลรวมของทราฟฟิกที่เส้นทางที่ 2 ต้องจะต้องทำการส่ง คือ 110 เมกะบิตต่อวินาที ทำให้เครือข่าย SDN ในผลการทดลองนี้ ใช้เวลามากกว่า เครือข่ายในปัจจุบัน แสดงได้ดังรูปที่ 5.57



รูปที่ 5.57 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.2.3

5.3.3. คอนโทรลเลอร์แบ่งทราฟฟิก เมื่อเส้นทางที่ 1 มีการใช้งานมากกว่า 80% และปริมาณทราฟฟิกที่ต้องการส่งผ่านช่องทางที่ 1 มีมากกว่าความสามารถในการส่งทราฟฟิก

คอนโทรลเลอร์จะทำงานเหมือนกับข้อที่ 5.3.2 ทุกประการ โดยในหัวข้อนี้จะเป็นการทดสอบบนเงื่อนไข เมื่อปริมาณทราฟฟิกที่ต้องการส่งบนเส้นทางที่ 1 มีค่ามากกว่าความสามารถในการส่งของเส้นทางที่ ซึ่งจะส่งผลให้คอนโทรลเลอร์ทำการแบ่งทราฟฟิก ของ Host 8 ไปที่เส้นทางที่ 2 ตามเงื่อนไขที่กำหนด ในการทดลองจะมีรูปแบบในการส่งทราฟฟิก ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3.1.ทราฟฟิกจาก DC02 = 150 เมกะบิตต่อวินาที และ ทราฟฟิกจาก DC01 = 70 เมกะบิตต่อวินาที สัดส่วนทราฟฟิก รูปแบบที่ 1

1) ทราฟฟิกของ DC02 = 150 เมกะบิตต่อวินาที ประกอบไปด้วย Host 7 = 130 เมกะบิตต่อวินาที, Host 8 = 20 เมกะบิตต่อวินาที

2) ทราฟฟิกของ DC01 = 70 เมกะบิตต่อวินาที ประกอบไปด้วย Host 3 = 70 เมกะบิตต่อวินาที

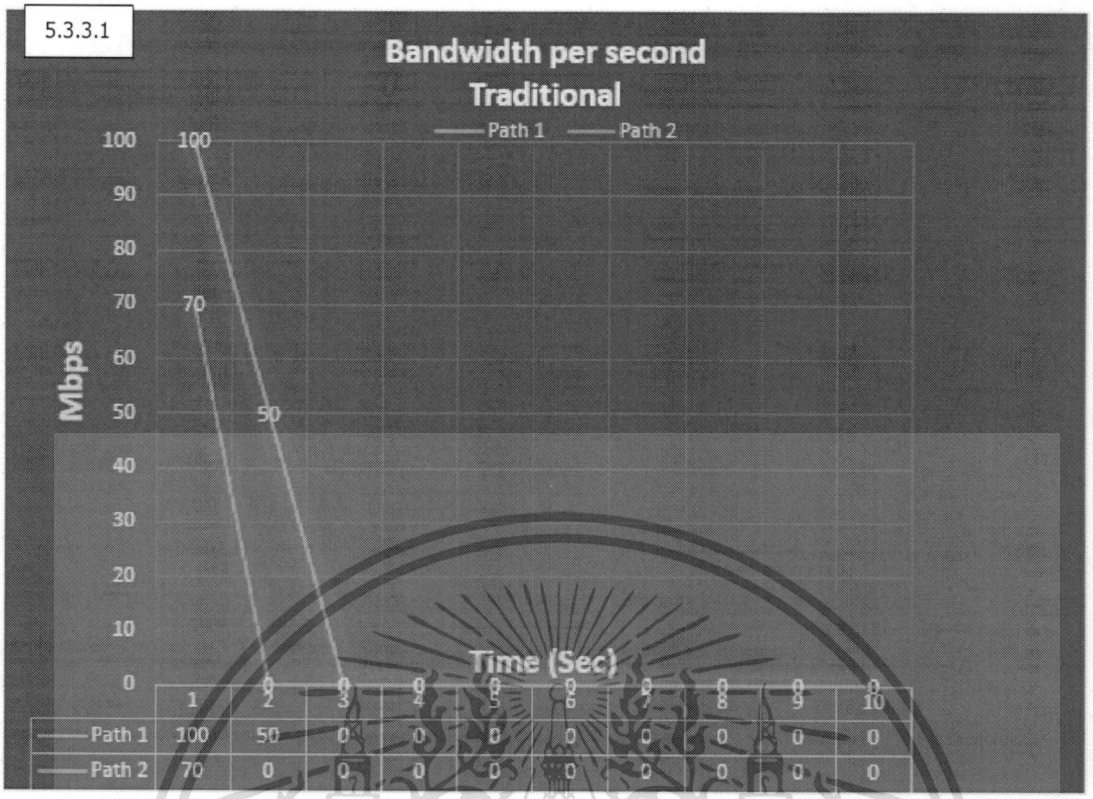
ความสามารถในการส่งทราฟฟิกของเส้นทางที่ 1 และ 2 มีขนาดของแต่ละช่องทางอยู่ที่ 100 เมกะบิตต่อวินาที โดยสามารถแสดงสัดส่วนของ ข้อมูลของทราฟฟิก ในแต่ละเส้นทาง เมื่อมีการส่งทราฟฟิก บนระบบเครือข่ายปัจจุบัน ได้ดังรูปที่ 5.58

	5.3.3.1	Traditional							
	Path 1					Path 2			
	Host.7	Host.8	total	capacity	Time(sec)	DC01	total	capacity	Time(sec)
1	130	20	150	100	1.5	70	70	100	0.7

รูปที่ 5.58 สัดส่วนของทราฟฟิกในการทดลองบนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.3.1

กราฟแสดงถึงผลการทดลองที่ 5.3.3.1 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่ายปัจจุบัน ความต้องการ ในการส่งทราฟฟิก บนเส้นทางที่ 1 อยู่ที่ 150 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 150 /100 เมกะบิตต่อวินาที = 1.5 วินาที และความต้องการในการส่งทราฟฟิกบนเส้นทางที่ 2 อยู่ที่ 70 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 70 /100 เมกะบิตต่อวินาที = 0.7 วินาที การส่งข้อมูลบนทั้ง 2 เส้นทางจะเริ่มพร้อมกันส่งผลให้ ระบบเครือข่ายที่ทำงานในรูปแบบปัจจุบัน สามารถส่งข้อมูลเสร็จสิ้น ใช้ระยะเวลา 1.5 วินาที สามารถแสดงได้ดังรูปที่ 5.59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.59 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.3.1

ทดลองส่งทราฟฟิกแบบเดียวกัน บนระบบเครือข่ายที่ทำงานด้วย SDN จากเงื่อนไขการส่งทราฟฟิก จะส่งผลให้ปริมาณการใช้งานทราฟฟิกบนเส้นทางที่ 1 มีปริมาณ มากกว่า 80 เมกะบิตต่อวินาที ตรงตามเงื่อนไขบนคอลโทรลเลอร์ ทำให้เกิดการสั่งงานแบ่งทราฟฟิกของ Host 8 ไปยังเส้นทางที่ 2 ร่วมกับการส่งทราฟฟิกของ DC01 ส่งผลให้จำนวนทราฟฟิกที่ต้องส่งผ่านเส้นทางที่ 1 มีค่า 130 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 130 /100 เมกะบิตต่อวินาที = 1.3 วินาที และ บนเส้นทางที่ 2 มีค่า 90 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 90 /100 เมกะบิตต่อวินาที = 0.9 วินาที ดังแสดงตามรูปที่ 5.60

5.3.3.1			SDN					
Path 1				Path 2				
Host.7	total	capacity	Time(sec)	Host.8	DC01	total	capacity	Time(sec)
130	130	100	1.3	20	70	90	100	0.9

รูปที่ 5.60 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 จำนวน 1 แพ็กเก็ต โดยมีขนาด 16.25M คือ 16.25 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 16.25x8 = 130 เมกะบิตต่อวินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 1.3 วินาที ดังแสดงตามรูปที่ 5.61

5.3.3.1

```
mininet>
mininet> h7 iperf -c h9 -t 1 -i 1 -n 16.25M
------------------------------------------------------------
Client connecting to 10.0.0.9, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.7 port 47354 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  13.1 MBytes   110 Mbits/sec
[  3]  0.0- 1.3 sec  16.2 MBytes   106 Mbits/sec
mininet>
```

รูปที่ 5.61 การทดลองส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 ตามเงื่อนไขที่ 5.3.3.1

ทดลองส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 จำนวน 1 แพ็กเก็ต โดยมีขนาด 11.25M คือ 11.25 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 11.25x8 = 90 เมกะบิตต่อวินาที โดยในการทดลองจะใช้ทราฟฟิกรวม ในการส่งทดลอง จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 0.9 วินาที ดังแสดงตามรูปที่ 5.62

5.3.3.1

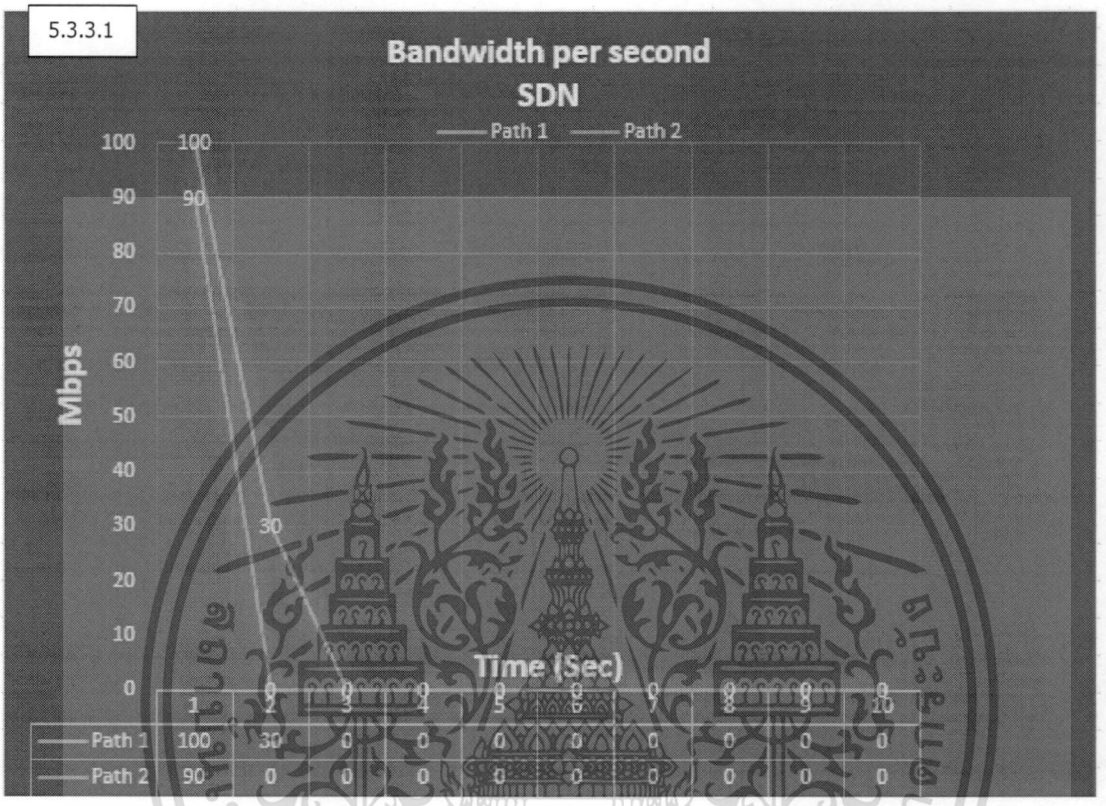
```
mininet>
mininet> h3 iperf -c h10 -t 1 -i 1 -n 11.25M
------------------------------------------------------------
Client connecting to 10.0.0.10, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 42158 connected with 10.0.0.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.9 sec  11.2 MBytes   106 Mbits/sec
mininet>
```

รูปที่ 5.62 การทดลองส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 ตามเงื่อนไขที่ 5.3.3.1

กราฟแสดงถึงผลการทดลองที่ 5.3.3.1 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่าย SDN โดยมีความต้องการ ในการส่งทราฟฟิกบนเส้นทางที่ 1 อยู่ที่ 150 เมกะบิตต่อวินาที บนเส้นทางที่ 2 อยู่ที่ 70 เมกะบิตต่อวินาที ส่งผลให้เข้าเงื่อนไขที่กำหนดไว้ที่คอลโทรลเลอร์ ทำให้มีการแบ่งทราฟฟิกจาก Host 8 ให้ไปทางเส้นทางที่ 2 โดยในกรณีนี้พบว่าที่ Host 8 มีการส่งทราฟฟิก จำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

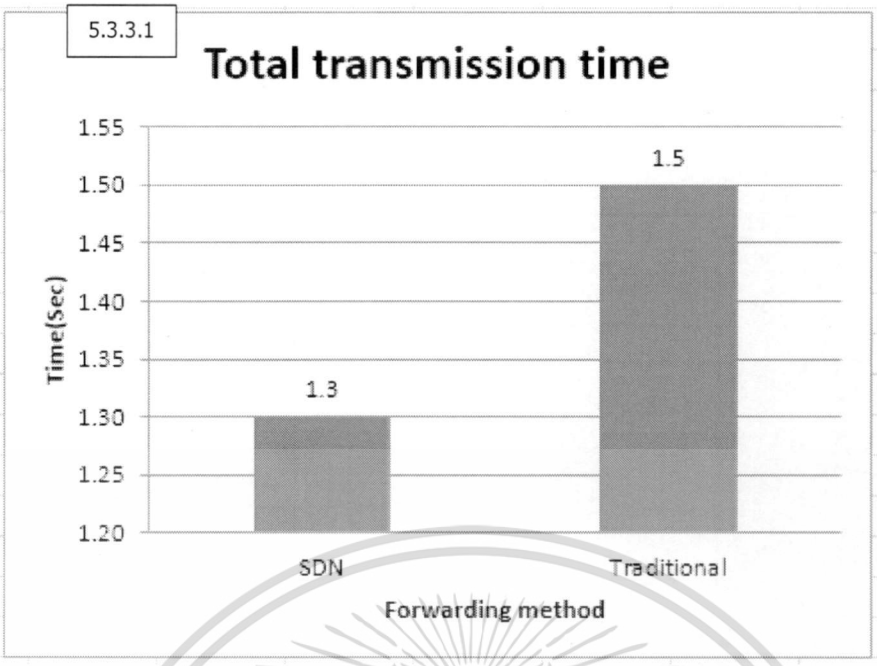
20 เมกะบิตต่อวินาที ส่งผลให้ ปริมาณทราฟฟิกบนเส้นทางที่ 1 มีค่า 130 เมกะบิตต่อวินาที และ บนเส้นทางที่ 2 มีค่า 90 เมกะบิตต่อวินาที ส่งผลให้ระยะเวลามากสุดในการส่งข้อมูลทราฟฟิก อยู่ที่ 1.3 วินาที บนเส้นทางที่ 1 สามารถแสดงได้ดังรูปที่ 5.63



รูปที่ 5.63 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่าย SDN ในการทดลองที่ 5.3.3.1

จากการทดลองส่งทราฟฟิกบนเครือข่ายที่ทำงานด้วยระบบ SDN และบนระบบเครือข่ายที่ทำงานด้วยระบบปัจจุบัน ผลที่ได้พบว่าการทำงาน ของระบบเครือข่าย SDN ใช้เวลาในการส่งข้อมูลน้อยกว่าระบบเครือข่ายปัจจุบัน เนื่องด้วยการแบ่งทราฟฟิกของคอนโทรลเลอร์ สามารถช่วยลดความหนาแน่นที่เกิดขึ้นบนเส้นทางที่ 1 และปริมาณทราฟฟิกรวมหลังการแบ่งทราฟฟิก บนเส้นทางที่ 2 ยังมีค่าน้อยกว่า ค่าความสามารถในการส่งทราฟฟิก สามารถแสดงได้ดังรูปที่ 5.64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.64 กราฟแสดงจำนวนเวลา ที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.3.1

5.3.3.2.ทราฟฟิกจาก DC02 = 150 เมกะบิตต่อวินาที และ ทราฟฟิกจาก DC01 = 70 เมกะบิตต่อวินาที สัดส่วนทราฟฟิก รูปแบบที่ 2

1) ทราฟฟิกของ DC02 = 150 เมกะบิตต่อวินาที ประกอบไปด้วย Host 7 = 110 เมกะบิตต่อวินาที, Host 8 = 40 เมกะบิตต่อวินาที

2) ทราฟฟิกของ DC01 = 70 เมกะบิตต่อวินาที ประกอบไปด้วย Host 3 = 70 เมกะบิตต่อวินาที

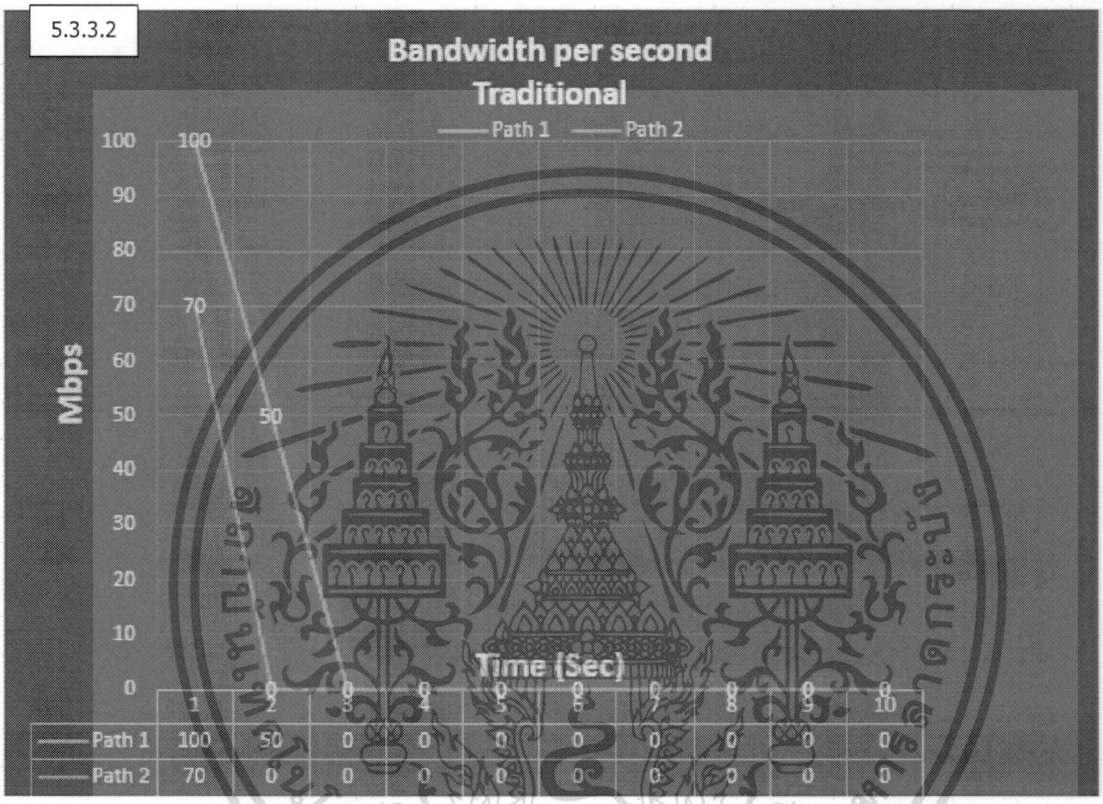
โดยสามารถแสดงสัดส่วนของข้อมูลของทราฟฟิกในแต่ละเส้นทาง เมื่อมีการส่งทราฟฟิกบนระบบเครือข่ายปัจจุบัน ได้ดังรูปที่ 5.65

	5.3.3.2	Traditional							
	Path 1					Path 2			
	Host.7	Host.8	total	capacity	Time(sec)	DC01	total	capacity	Time(sec)
2	110	40	150	100	1.5	70	70	100	0.7

รูปที่ 5.65 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟแสดงถึงผลการทดลองที่ 5.3.3.2 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่ายปัจจุบัน ในการทดลองนี้มีการเปลี่ยนแปลงสัดส่วนของทราฟฟิกบนเส้นทางที่ 1 แต่ผลรวมปริมาณทราฟฟิกยังคงเดิม ที่ 150 เมกะบิตต่อวินาที และบนเส้นทางที่ 2 ไม่มีการเปลี่ยนแปลงของปริมาณทราฟฟิก ผลรวมปริมาณทราฟฟิก ยังคงเดิม ที่ 70 เมกะบิตต่อวินาที ทำให้ผลกาทดลองได้ผลลัพธ์เหมือนกับการทดลองที่ 5.3.2.1 สามารถแสดงได้ดังรูปที่ 5.66



รูปที่ 5.66 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.3.2

ทดสอบส่งทราฟฟิกแบบเดียวกันบนระบบเครือข่ายที่ทำงานด้วย SDN จากเงื่อนไขการส่งทราฟฟิก จะส่งผลให้ปริมาณการใช้งานทราฟฟิกบนเส้นทางที่ 1 มีปริมาณ มากกว่า 80 เมกะบิตต่อวินาที ตรงตามเงื่อนไขบนคอลโทรลเลอร์ ทำให้เกิดการสั่งงานแบ่งทราฟฟิกของ Host 8 ไปยังเส้นทางที่ 2 ร่วมกับการส่งทราฟฟิกของ DC01 ส่งผลให้จำนวนทราฟฟิกที่ต้องส่งผ่านเส้นทางที่ 1 มีค่า 110 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 110 /100 เมกะบิตต่อวินาที = 1.1 วินาที และ บนเส้นทางที่ 2 มีค่า 110 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 110 /100 เมกะบิตต่อวินาที = 1.1 วินาที ดังแสดงตามรูปที่ 5.67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3.2	SDN							
Path 1				Path 2				
Host.7	total	capacity	Time(sec)	Host.8	DC01	total	capacity	Time(sec)
110	110	100	1.1	40	70	110	100	1.1

รูปที่ 5.67 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.3.2

ทดสอบส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 จำนวน 1 แพ็กเก็ต โดยมีขนาด 13.75M คือ 13.75 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 13.75x8 = 110 เมกะบิตต่อวินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 1.1 วินาที ดังแสดงตามรูปที่ 5.68

```
5.3.3.2 >
mininet>
mininet> h7 iperf -c h9 -t 1 -i 1 -n 13.75M
------------------------------------------------------------
Client connecting to 10.0.0.9, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.7 port 47378 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  13.4 MBytes   112 Mbits/sec
[  3]  0.0- 1.1 sec  13.8 MBytes   109 Mbits/sec
mininet>
```

รูปที่ 5.68 การทดลองส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 ตามเงื่อนไขที่ 5.3.3.2

ทดสอบส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 จำนวน 1 แพ็กเก็ต โดยมีขนาด 13.75M คือ 13.75 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 13.75x8 = 110 เมกะบิตต่อวินาที โดยในการทดลองจะใช้ทราฟฟิกรวมในการส่งทดลอง จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 1.1 วินาที ดังแสดงตามรูปที่ 5.69

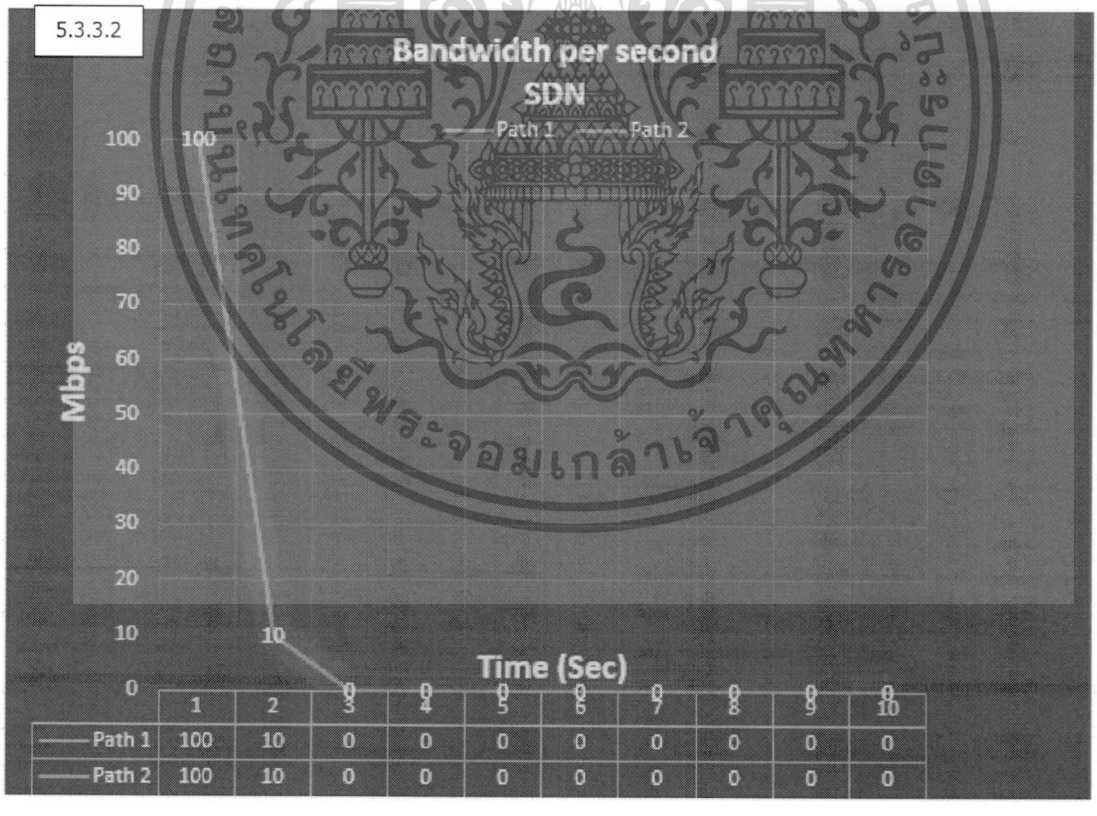
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3.2

```
mininet> h3 iperf -c h10 -t 1 -i 1 -n 13.75M
------------------------------------------------------------
Client connecting to 10.0.0.10, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 42173 connected with 10.0.0.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  13.6 MBytes   114 Mbits/sec
[  3]  0.0- 1.1 sec  13.8 MBytes   108 Mbits/sec
mininet>
mininet>
```

รูปที่ 5.69 การทดลองส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 ตามเงื่อนไขที่ 5.3.3.2

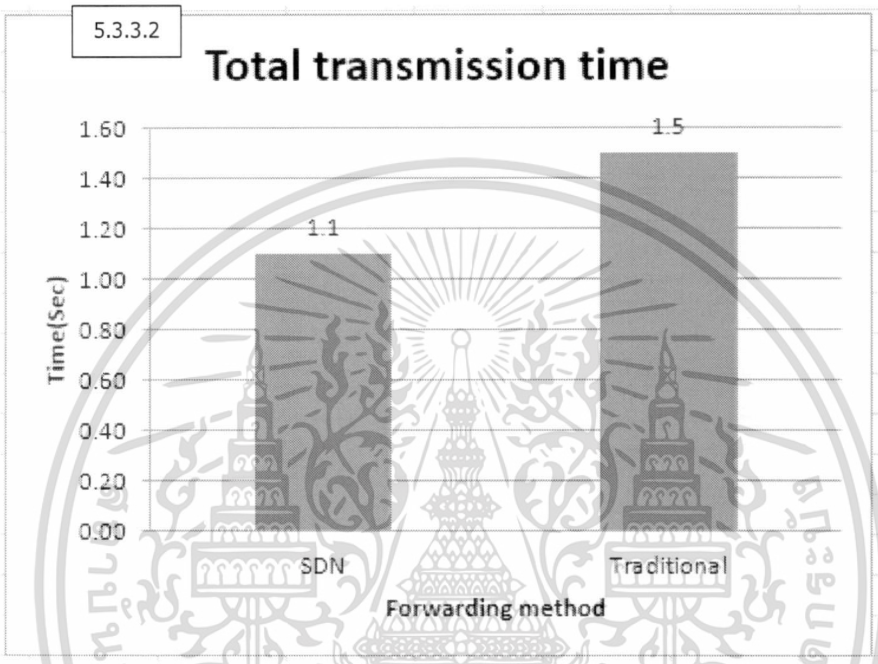
กราฟแสดงถึงผลการทดลองที่ 5.3.3.2 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่าย SDN ในการทดลองนี้เมื่อคอนโทรลเลอร์ทำการแบ่งทราฟฟิกของ Host 8 ออกไปยังเส้นทางที่ 2 ทำให้ ผลรวมของทราฟฟิกของแต่ละเส้นทางการส่งทราฟฟิก มีค่าเท่ากันที่ 110 เมกะบิตต่อวินาที สามารถแสดงได้ดังรูปที่ 5.70



รูปที่ 5.70 กราฟแสดงการส่งข้อมูลทราฟฟิก บนระบบเครือข่าย SDN ในการทดลองที่ 5.3.3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลอง ส่งทราฟฟิกบนเครือข่ายที่ทำงานด้วยระบบ SDN และบนระบบเครือข่ายที่ทำงานด้วยระบบปัจจุบัน ผลที่ได้พบว่าการทำงานของระบบเครือข่าย SDN ใช้เวลาในการส่งข้อมูลน้อยกว่า ระบบเครือข่ายปัจจุบัน เนื่องด้วยการแบ่งทราฟฟิกของคอนโทรลเลอร์ สามารถช่วยลดความหนาแน่นที่เกิดขึ้นบนเส้นทางที่ 1 และสามารถแบ่งทราฟฟิกไปยังเส้นทางที่ 2 ได้เหมาะสม สามารถแสดงได้ดังรูปที่ 5.71



รูปที่ 5.71 กราฟแสดงจำนวนเวลาที่ใช้ในการส่งข้อมูลทราฟฟิกเปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.3.2

5.3.3.3.ทราฟฟิกจาก DC02 = 150 เมกะบิตต่อวินาที และ ทราฟฟิกจาก DC01 = 70 เมกะบิตต่อวินาที สัดส่วนทราฟฟิก รูปแบบที่ 3

1) ทราฟฟิกของ DC02 = 150 เมกะบิตต่อวินาที ประกอบไปด้วย Host 7 = 10 เมกะบิตต่อวินาที, Host 8 = 140 เมกะบิตต่อวินาที

2) ทราฟฟิกของ DC01 = 70 เมกะบิตต่อวินาที ประกอบไปด้วย Host 3 = 70 เมกะบิตต่อวินาที

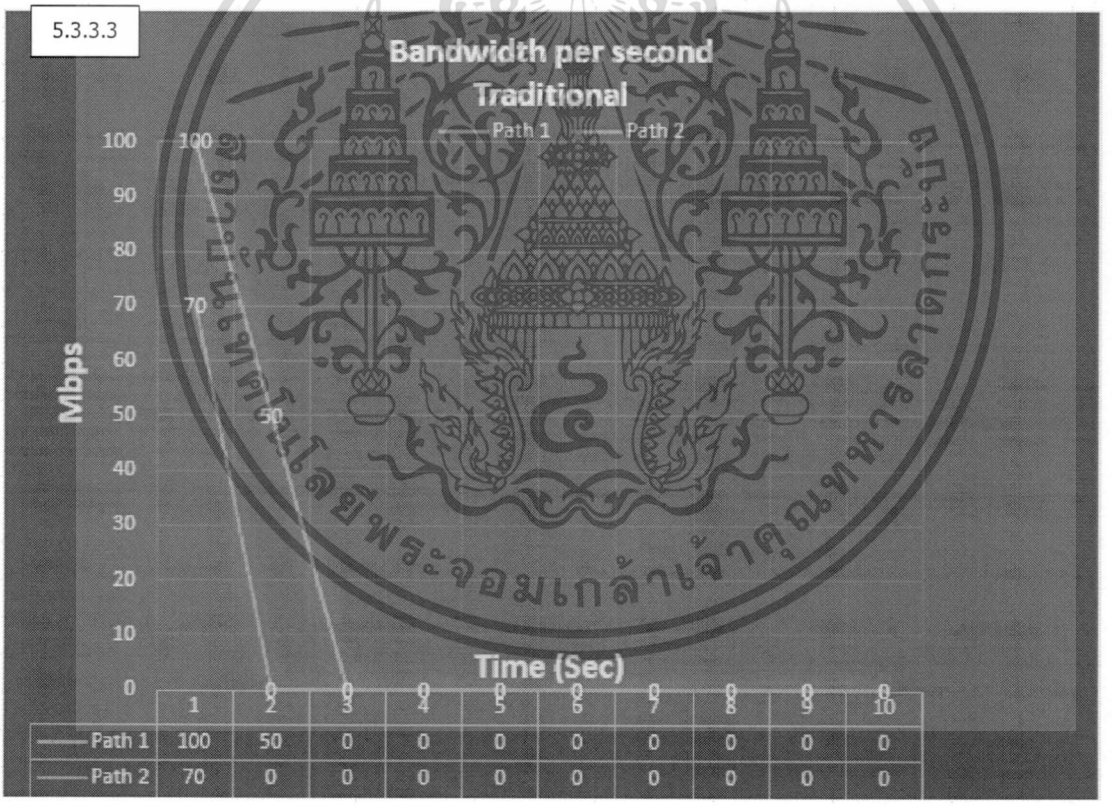
โดยสามารถแสดงสัดส่วนของ ข้อมูลของทราฟฟิก ในแต่ละเส้นทาง เมื่อมีการส่งทราฟฟิกบนระบบเครือข่ายปัจจุบัน ได้ดังรูปที่ 5.72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	5.3.3.3	Traditional							
	Path 1					Path 2			
	Host.7	Host.8	total	capacity	Time(sec)	DC01	total	capacity	Time(sec)
3	10	140	150	100	1.5	70	70	100	0.7

รูปที่ 5.72 สัดส่วนของทราฟฟิกในการทดลอง บนระบบเครือข่ายปัจจุบัน เงื่อนไขที่ 5.3.3.3

กราฟแสดงถึงผลการทดลองที่ 5.3.3.3 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่ายปัจจุบัน ในการทดลองนี้มีการเปลี่ยนแปลงสัดส่วนของทราฟฟิกบนเส้นทางที่ 1 แต่ผลรวมปริมาณทราฟฟิกยังคงเดิม ที่ 150 เมกะบิตต่อวินาที และบนเส้นทางที่ 2 ไม่มีการเปลี่ยนแปลงของปริมาณทราฟฟิก ผลรวมปริมาณทราฟฟิก ยังคงเดิม ที่ 70 เมกะบิตต่อวินาที ทำให้ผลการทดลองได้ผลลัพธ์เหมือนกับการทดลองที่ 5.3.2.1 และ 5.3.2.2 สามารถแสดงได้ดังรูปที่ 5.73



รูปที่ 5.73 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่ายปัจจุบัน ในการทดลองที่ 5.3.3.3

ทดลองส่งทราฟฟิกแบบเดียวกันบนระบบเครือข่ายที่ทำงานด้วย SDN จากเงื่อนไขการส่งทราฟฟิกจะส่งผลให้ปริมาณการใช้งานทราฟฟิกบนเส้นทางที่ 1 มีปริมาณ มากกว่า 80 เมกะบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อวินาที ตรงตามเงื่อนไขบนคอลโทรลเลอร์ทำให้เกิดการสั่งงานแบ่งทราฟฟิกของ Host 8 ไปยังเส้นทางที่ 2 ร่วมกับการส่งทราฟฟิกของ DC01 ส่งผลให้จำนวนทราฟฟิกที่ต้องส่งผ่านเส้นทางที่ 1 มีค่า 10 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 10 /100 เมกะบิตต่อวินาที = 0.1 วินาที และ บนเส้นทางที่ 2 มีค่า 210 เมกะบิตต่อวินาที เมื่อพิจารณาจากความสามารถในการส่งที่ 100 เมกะบิตต่อวินาที จะใช้เวลาทั้งหมดในการส่งอยู่ที่ 210 /100 เมกะบิตต่อวินาที = 2.1 วินาที ดังแสดงตามรูปที่ 5.74

	5.3.3.3	SDN						
	Path 1				Path 2			
Host.7	total	capacity	Time(sec)	Host.8	DC01	total	capacity	Time(sec)
10	10	100	0.1	140	70	210	100	2.1

รูปที่ 5.74 สัดส่วนของทราฟฟิกในการทดลองบนระบบเครือข่าย SDN เงื่อนไขที่ 5.3.3.3

ทดสอบส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 จำนวน 1 แพ็กเก็ต โดยมีขนาด 1.25M คือ 1.25 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 1.25x8 = 10 เมกะบิตต่อวินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 0.1 วินาที ดังแสดงตามรูปที่ 5.75

```
m  5.3.3.3
mininet>
mininet>
mininet> h7 iperf -c h9 -t 1 -i 1 -n 1.25M
------------------------------------------------------------
Client connecting to 10.0.0.9, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.7 port 47379 connected with 10.0.0.9 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.1 sec  1.25 MBytes   152 Mbits/sec
mininet>
```

รูปที่ 5.75 การทดลองส่งทราฟฟิกจาก Host 7 ไปยัง Host 9 ตามเงื่อนไขที่ 5.3.3.3

ทดสอบส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 จำนวน 1 แพ็กเก็ต โดยมีขนาด 26.25M คือ 26.25 เมกะไบต์ต่อวินาที ซึ่งเมื่อมีการแปลงเป็นหน่วยบิต จะได้เป็น 26.25x8 = 210 เมกะบิตต่อวินาที จากการทดลองดังกล่าวส่งผลให้ใช้เวลาในการส่งทราฟฟิก อยู่ที่ 2.1 วินาที ดังแสดงตามรูปที่ 5.76

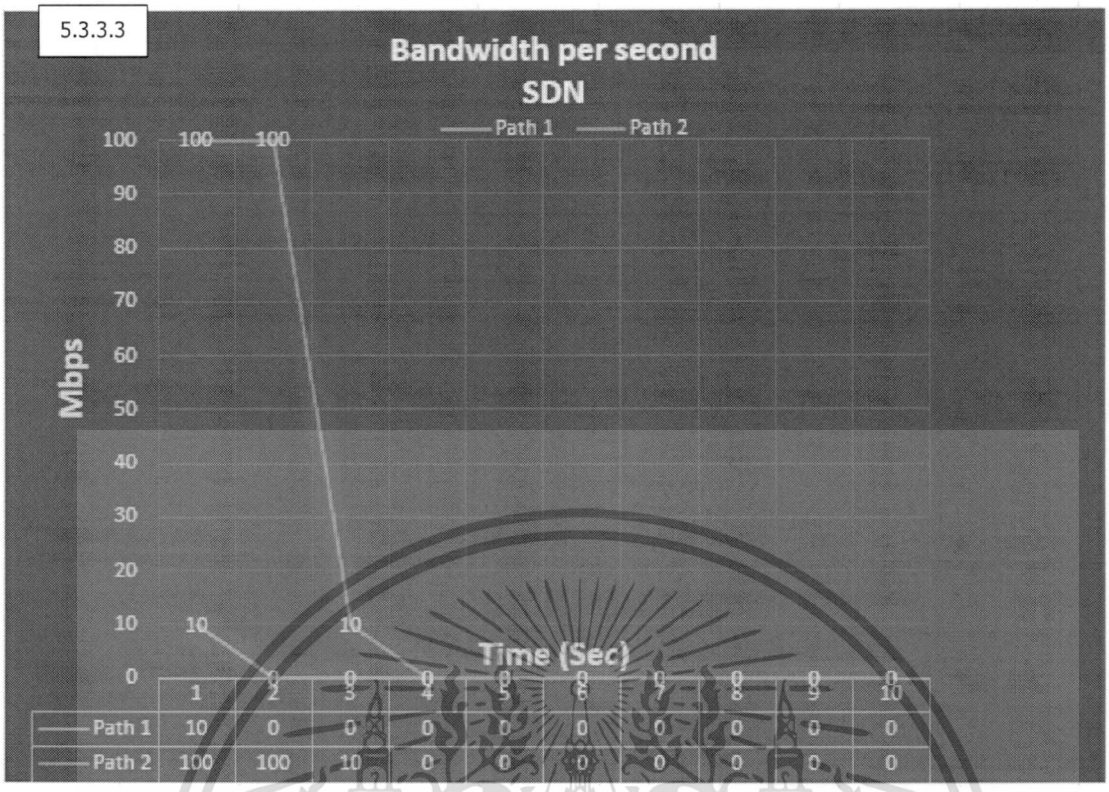
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
5.3.3.3 et>
        et>
mininet> h3 iperf -c h10 -t 1 -i 1 -n 26.25M
------------------------------------------------------------
Client connecting to 10.0.0.10, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 42253 connected with 10.0.0.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  13.9 MBytes   116 Mbits/sec
[  3]  1.0- 2.0 sec  11.0 MBytes  92.3 Mbits/sec
[  3]  0.0- 2.1 sec  26.2 MBytes   107 Mbits/sec
mininet>
mininet>
```

รูปที่ 5.76 การทดลองส่งทราฟฟิกจาก Host 3 ไปยัง Host 10 ตามเงื่อนไขที่ 5.3.3.3

กราฟแสดงถึงผลการทดลองที่ 5.3.3.3 การส่งทราฟฟิกต่อหน่วยเวลาบนระบบเครือข่าย SDN โดยมีความต้องการในการส่งทราฟฟิก บนเส้นทางที่ 1 อยู่ที่ 150 เมกะบิตต่อวินาที บนเส้นทางที่ 2 อยู่ที่ 70 เมกะบิตต่อวินาที ส่งผลให้เข้าเงื่อนไขที่กำหนดไว้ที่คอลโทรลเลอร์ ทำให้มีการแบ่งทราฟฟิกจาก Host 8 ให้ไปทางเส้นทางที่ 2 โดยในกรณีนี้พบว่าที่ Host 8 มีการส่งทราฟฟิกจำนวน 140 เมกะบิตต่อวินาที ส่งผลให้ ปริมาณทราฟฟิกบนเส้นทางที่ 1 มีค่า 10 เมกะบิตต่อวินาที และ บนเส้นทางที่ 2 มีค่า 210 เมกะบิตต่อวินาที ส่งผลให้ปริมาณทราฟฟิกระหว่างเส้นทางที่ 1 และ 2 มีความแตกต่างกันมาก จึงทำให้ระยะเวลามากสุดในการส่งข้อมูลทราฟฟิก อยู่ที่ 2.1 วินาที บนเส้นทางที่ 2 สามารถแสดงได้ดังรูปที่ 5.77

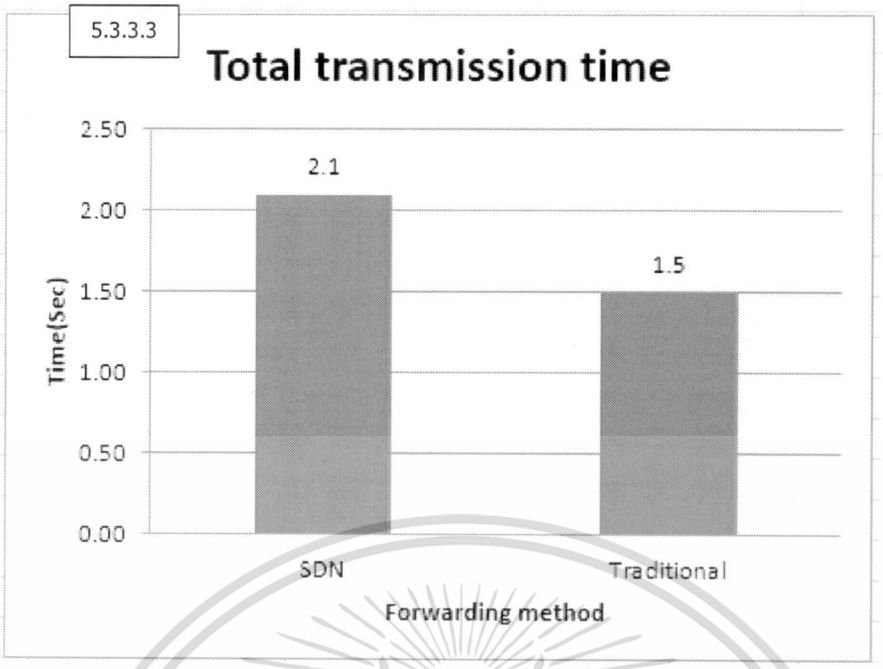
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.77 กราฟแสดงการส่งข้อมูลทราฟฟิกบนระบบเครือข่าย SDN ในการทดลองที่ 5.3.3.3

จากการทดลองส่งทราฟฟิกบนเครือข่ายที่ทำงานด้วยระบบ SDN และบนระบบเครือข่ายที่ทำงานด้วยระบบปัจจุบัน ผลที่ได้พบว่าการทำงานของระบบเครือข่าย SDN ใช้เวลาในการส่งข้อมูลมากกว่าระบบเครือข่ายปัจจุบัน สาเหตุเกิดจากการแบ่งทราฟฟิกของคอนโทรลเลอร์ที่ดำเนินการด้วยเงื่อนไข IP address ต้นทาง ประกอบกับปริมาณทราฟฟิกที่ต้องการส่งของ IP address ดังกล่าวมีปริมาณมาก ดังนั้นเมื่อแบ่งทราฟฟิกไปยังเส้นทางที่ 2 จึงส่งผลให้เกิดความหนาแน่นบนเส้นทางที่ 2 โดยในภาพรวม ส่งผลให้การทำงานของเครือข่าย SDN ใช้เวลาในการส่งทราฟฟิกมากกว่าเครือข่ายในปัจจุบัน สามารถแสดงได้ดังรูปที่ 5.78

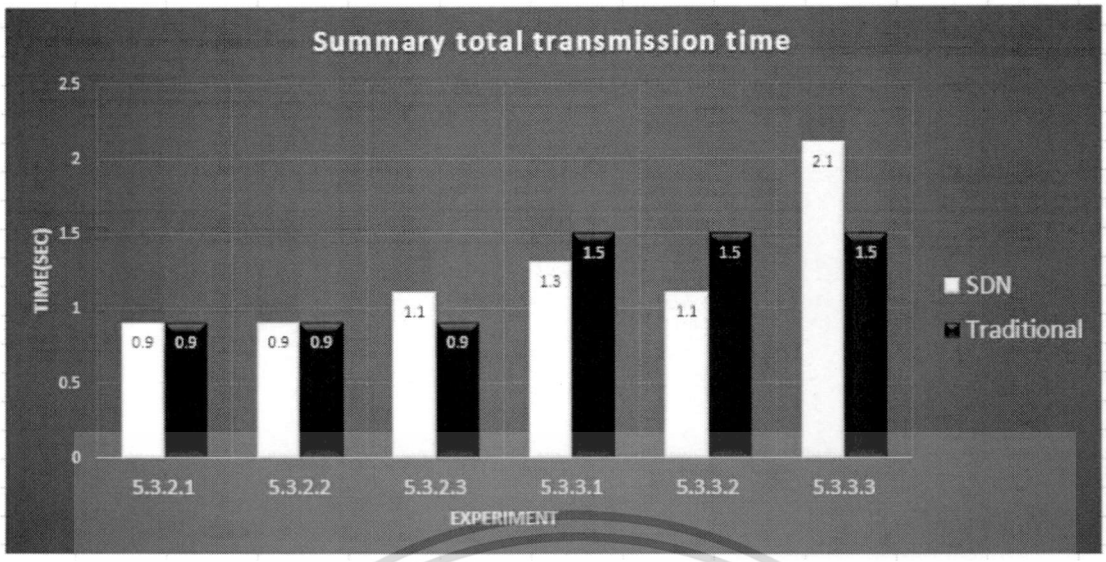
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.78 กราฟแสดงจำนวนเวลาที่ใช้ในการส่งข้อมูลทราฟฟิก เปรียบเทียบระหว่างเครือข่ายปัจจุบันและเครือข่าย SDN ในการทดลองที่ 5.3.3.3

จากรูปแบบการทดลองที่มีการปรับเปลี่ยนสัดส่วนของทราฟฟิกในรูปแบบต่างๆ เพื่อใช้ในการทดลอง บนระบบเครือข่ายที่ทำงานในรูปแบบปัจจุบัน และเครือข่ายที่ทำงานในรูปแบบ SDN ผลลัพธ์ที่ได้ คือ ระบบเครือข่ายที่ทำงานในรูปแบบ SDN ในกรณีทดลองที่ 5.3.2.3 และ 5.3.3.3 พบว่า สามารถทำงานได้ช้ากว่าระบบเครือข่ายที่ทำงานในรูปแบบปัจจุบันแต่ไม่ทั้งหมด ระบบเครือข่ายที่ทำงานในรูปแบบ SDN ในกรณีทดลองที่ 5.3.3.1 และ 5.3.3.2 พบว่าสามารถทำงานได้เร็วกว่าระบบเครือข่ายที่ทำงานในรูปแบบปัจจุบัน ซึ่งจากผลการทดลองทั้งหมดทำให้พบว่าเงื่อนไขในการแบ่งทราฟฟิกของเครือข่าย SDN และปริมาณสัดส่วนของทราฟฟิก เป็นปัจจัยที่มีผลต่อการทำงานของเครือข่ายทั้ง 2 รูปแบบ สามารถแสดงได้ดังรูปที่ 5.79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.79 กราฟสรุปแสดงจำนวนเวลาที่ใช้ในการส่งข้อมูลทราฟฟิกในการทดลองที่ 5.3

5.4 ข้อเสนอแนะในการปรับปรุงการติดต่อสื่อสารระหว่างคอนโทรลเลอร์และอุปรณ์เครือข่าย SDN

บนระบบเครือข่าย SDN องค์ประกอบหลักที่สำคัญและขาดไม่ได้คือ คอนโทรลเลอร์ ซึ่งมีหน้าที่หลัก คือ บริหารจัดการอุปกรณ์ในเครือข่าย ดังนั้นการติดต่อสื่อสารจึงเป็นสิ่งสำคัญ เพื่อให้สามารถจัดการเครือข่ายได้ด้วยความรวดเร็วทันต่อเหตุการณ์ที่เกิดขึ้น โดยการติดต่อสื่อสารระหว่างคอนโทรลเลอร์และอุปกรณ์เครือข่าย ปัจจุบันใช้การส่งข้อมูลแบบแพ็กเก็ตสวิตช์ชิง หมายความว่าช่องทางการสื่อสารระหว่างคอนโทรลเลอร์และอุปกรณ์เครือข่ายจะมีข้อมูลแพ็กเก็ตอื่นๆร่วมใช้งานด้วยไปพร้อมกันไม่มีการแยกช่องทางทราฟฟิกให้ใช้เฉพาะงานนั้นๆ ดังนั้น จึงอาจส่งผลกระทบต่อการติดต่อสื่อสารของคอนโทรลเลอร์ได้ เช่น กรณีมีปริมาณทราฟฟิกหนาแน่นอาจส่งผลให้การติดต่อสื่อสารระหว่างคอนโทรลเลอร์ และอุปกรณ์เครือข่ายเกิดความล่าช้าหรือไม่สามารถติดต่อกันได้จนอาจส่งผลกระทบต่อการทำงานของระบบเครือข่ายทั้งหมด แต่ในการศึกษาครั้งนี้ทางผู้จัดทำได้ทดสอบระบบเครือข่าย SDN ผ่านซอฟต์แวร์จำลองช่องทางในการติดต่อสื่อสารระหว่างคอนโทรลเลอร์และอุปกรณ์เครือข่าย จะผ่านช่องทางระบบภายในของซอฟต์แวร์จำลอง จึงไม่สามารถทำการทดลองได้ ผู้จัดทำจึงขอนำเสนอแนวทางในการทดลองและ ปัญหาที่อาจเกิดขึ้นเพื่อเป็นประโยชน์ในการศึกษาครั้งต่อไป ดังนี้

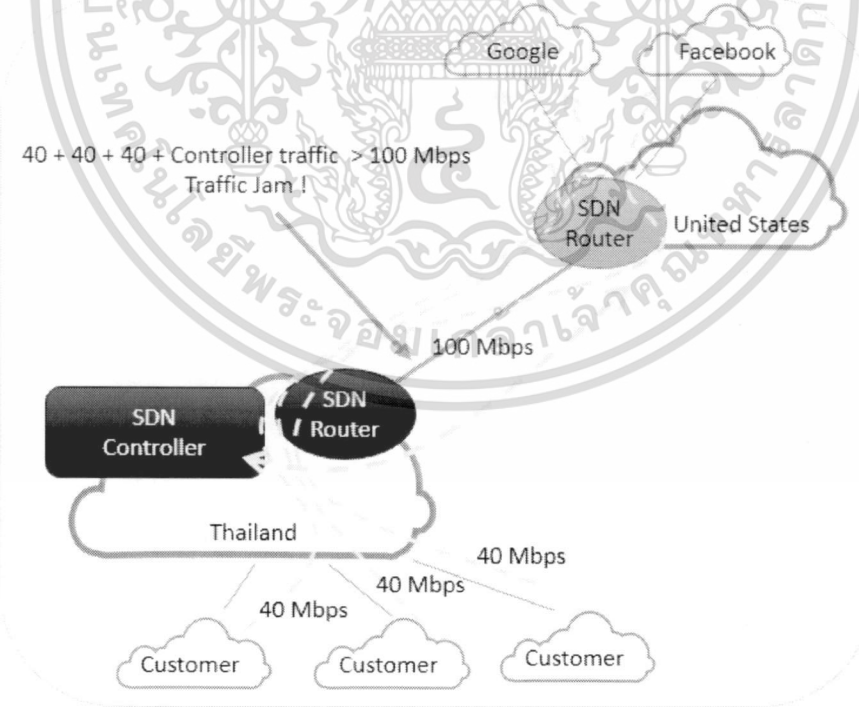
5.4.1. ปัญหาที่เกิดจาก ช่องทางการส่งทราฟฟิกมีความหนาแน่น

เนื่องจากปัจจุบันการส่งข้อมูลเป็นแบบแพ็กเก็ตสวิตช์ชิง ช่องทางในการส่งข้อมูลจึงมีการแบ่งปันการใช้งาน เพื่อให้เกิดการใช้งานได้ปริมาณสูงสุด ด้วยเหตุนี้หากกรณีช่องสัญญาณในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากปัจจุบันการส่งข้อมูลเป็นแบบแพ็กเก็ตสวิตซ์ชิง ช่องทางในการส่งข้อมูลจึงมีการแบ่งปันการใช้งาน เพื่อให้เกิดการใช้งานได้ปริมาณสูงสุด ด้วยเหตุนี้หากกรณีช่องสัญญาณในการส่งทราฟฟิก มีความหนาแน่น ก็จะส่งผลต่อข้อมูลแพ็กเก็ต คือ ทำให้ส่งไปถึงปลายทางได้ล่าช้า หรือไม่ถึงปลายทาง

ตัวอย่าง จากรูปที่ 5.80 ระบบเครือข่าย SDN มีที่ตั้งจำนวน 2 แห่ง คือ Thailand และ United states โดยทั้ง 2 แห่งนี้มีช่องทางในการส่งทราฟฟิกเชื่อมต่อหากัน ด้วยขนาด 100 เมกะบิตต่อวินาที (Mbps) และระบบเครือข่าย Thailand มีการเชื่อมต่อกับระบบเครือข่ายของลูกค้า ระบบเครือข่าย United states มีการเชื่อมต่อกับผู้ให้บริการ Google และ Facebook โดยอุปกรณ์เครือข่าย SDN Router ที่ United states มีการทำงานในรูปแบบ SDN และมีการติดต่อสื่อสารกับคอนโทรลเลอร์ที่ Thailand ดังนั้นในกรณีที่ระบบเครือข่ายของลูกค้าที่ Thailand จำนวน 3 กลุ่ม ต้องการส่งทราฟฟิกกลุ่มละ 40 Mbps ไปยัง Google และ Facebook จะส่งผลให้ทราฟฟิกรวมทั้งหมดจากกลุ่มลูกค้าคือ 40 Mbps x 3 = 120 Mbps มีปริมาณมากกว่า ความสามารถในการส่งทราฟฟิกที่ 100 Mbps และหากในเวลาเดียวกันอุปกรณ์คอนโทรลเลอร์ต้องการติดต่อกับอุปกรณ์เครือข่าย SDN ที่ United states ก็จะส่งผลให้มีความล่าช้าในการติดต่อสื่อสารระหว่างกัน หรือในกรณีร้ายแรงอาจจะไม่สามารถติดต่อสื่อสารกันได้ ในช่วงเวลาดังกล่าวอันเนื่องมาจากความหนาแน่นของทราฟฟิกมีสูง

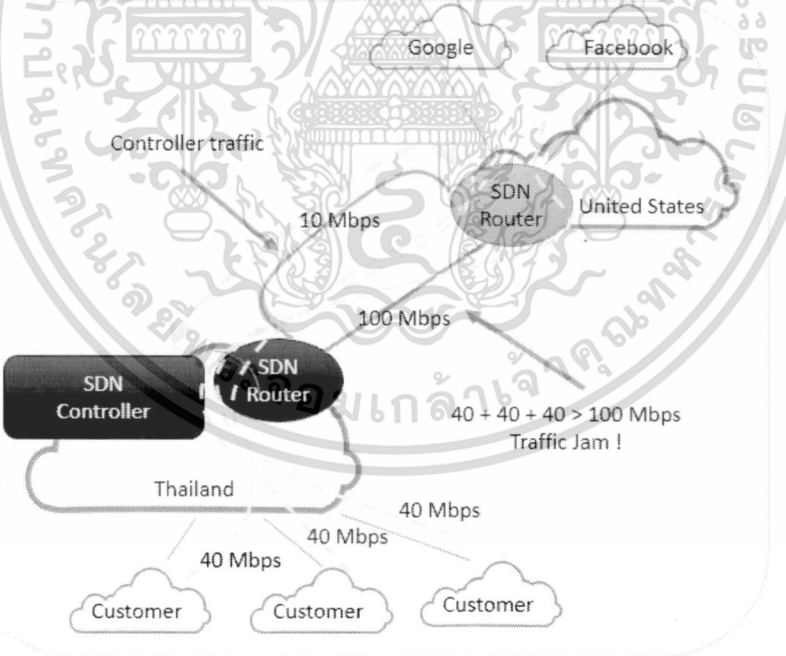


รูปที่ 5.80 การติดต่อสื่อสารของคอนโทรลเลอร์ในกรณีทราฟฟิกมีความหนาแน่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการทดลอง เสนอเพื่อใช้ในการแก้ไข

1) ดำเนินการทำเรื่องคุณภาพของการให้บริการ (Quality of service) และ การจัดลำดับของความสำคัญของข้อมูล สำหรับกรณีนี้ต้องพิจารณาทราฟฟิกที่เป็นการติดต่อสื่อสารระหว่างคอนโทรลเลอร์และอุปกรณ์ SDN ให้มีความสำคัญสูงสุด และทำการจัดส่งข้อมูลเหล่านี้ก่อนข้อมูลอื่น เพื่อทำให้การบริหารจัดการบนระบบเครือข่าย SDN สามารถดำเนินการได้เป็นปกติ
2) ดำเนินการสร้างช่องทางการเชื่อมต่อที่แยกออกจากระบบงานเดิม เพื่อใช้ส่งทราฟฟิกเฉพาะที่เกี่ยวข้องกับการบริหารจัดการระบบเครือข่าย SDN เท่านั้น เช่น ใช้ในการรับ-ส่งทราฟฟิกของการติดต่อสื่อสารระหว่างคอนโทรลเลอร์ อุปกรณ์ SDN และทราฟฟิก อื่นๆที่เกี่ยวข้องกับการบริหารจัดการระบบเครือข่าย SDN ทั้งนี้วิธีการนี้จะต้องลงทุนเพิ่มเติมจากระบบเครือข่ายเดิม อันเนื่องมาจาก การสร้างช่องทางเชื่อมต่อใหม่ ดังรูปที่ 5.81



รูปที่ 5.81 การติดต่อสื่อสารของคอนโทรลเลอร์ ในกรณีแยกช่องทางทราฟฟิกพิเศษ

5.4.2. ปัญหาที่เกิดจาก ระยะทางของติดต่อสื่อสาร (Latency time)

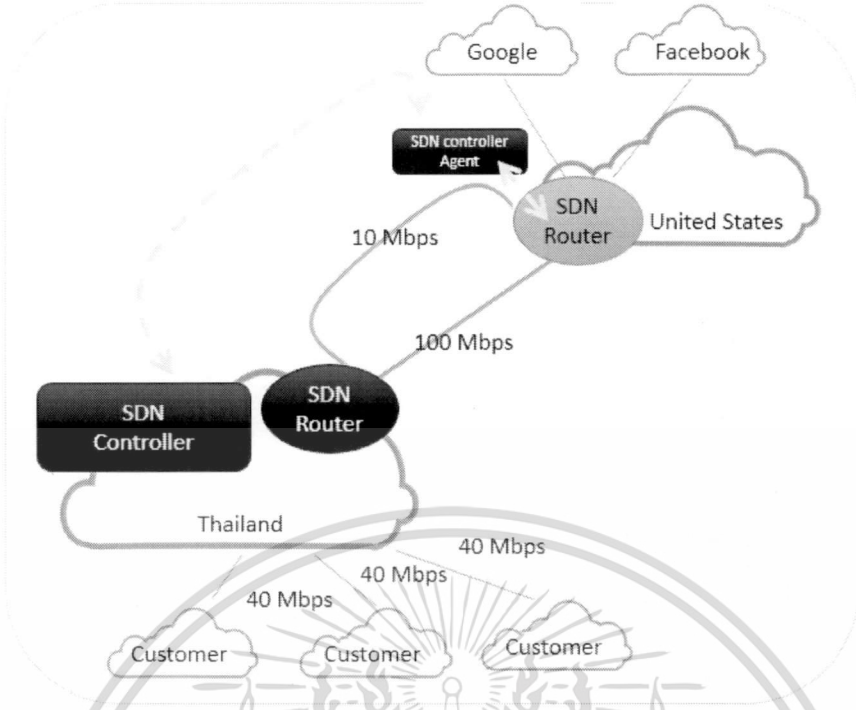
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องด้วยระบบเครือข่ายของผู้ให้บริการอินเทอร์เน็ต ต้องมีการสร้างการเชื่อมต่อไปยังพื้นที่ต่างๆ หลายๆ จุดเพื่อทำการเชื่อมต่อกันเป็นระบบเครือข่าย โดยในแต่ละจุดการเชื่อมต่อจะมีระยะทางที่แตกต่างกันออกไป เช่น การเชื่อมต่อระหว่างชั้นภายในอาคารเดียวกัน การเชื่อมต่อระหว่างประเทศ ประเทศไทยกับประเทศจีน จากปัจจัยดังกล่าวจะส่งผลต่อความเร็วในการส่งข้อมูลต่อ 1 หน่วยเวลา คือ การส่งข้อมูลระหว่างภายในอาคารจะทำได้เร็วกว่าการส่งข้อมูลระหว่างประเทศ อันเนื่องปัจจัยด้านระยะทางหรือสามารถเปรียบเทียบได้กับการขับรถไปกลับ ด้วยความเร็วเท่ากัน ระหว่างกรุงเทพ-นนทบุรี และกรุงเทพ-เชียงใหม่ ผลที่ได้คือ ในระยะเวลาเท่ากัน จำนวนรอบการขับรถของกรุงเทพ-นนทบุรีสามารถทำได้มากกว่า อันเนื่องด้วยระยะทางที่สั้นกว่า จากปัจจัยนี้ส่งผลต่อความรวดเร็วในการบริหารจัดการเครือข่าย SDN เพราะระบบประมวลผลการทำงานอยู่ไกลจากอุปกรณ์เครือข่ายที่ทำงาน

แนวทางเสนอเพื่อใช้แก้ไข

เนื่องด้วยปัญหาด้านระยะทางส่งผลต่อระยะเวลาในการส่งข้อมูล ทำให้เกิดผลกระทบต่อความเร็วในการประมวลผลการบริหารจัดการระบบเครือข่าย SDN ผู้จัดทำจึงขอเสนอแนวคิดว่า ให้มีการกระจายหน่วยทำงานคอนโทรลเลอร์ให้ใกล้อุปกรณ์เครือข่ายปลายทางมากที่สุด โดยมีคอนโทรลเลอร์ซึ่งทำหน้าที่เป็นศูนย์กลางอยู่เช่นเดิม แต่เพิ่มคอนโทรลเลอร์ย่อยกระจายไปตามอุปกรณ์เครือข่ายที่มีความห่างไกล และให้คอนโทรลเลอร์ย่อยสามารถสั่งงานไปยังอุปกรณ์เครือข่ายได้โดยตรง ทำหน้าที่รับคำสั่งจากคอนโทรลเลอร์ศูนย์กลางเพื่อนำไปประมวลผล และคอนโทรลเลอร์ย่อยจะสามารถประมวลผล บริหารจัดการระบบเครือข่าย SDN ที่รับผิดชอบได้ด้วยตนเอง เพื่อเพิ่มความรวดเร็วในการบริหารเครือข่ายสามารถแสดงได้ดังรูปที่ 5.82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

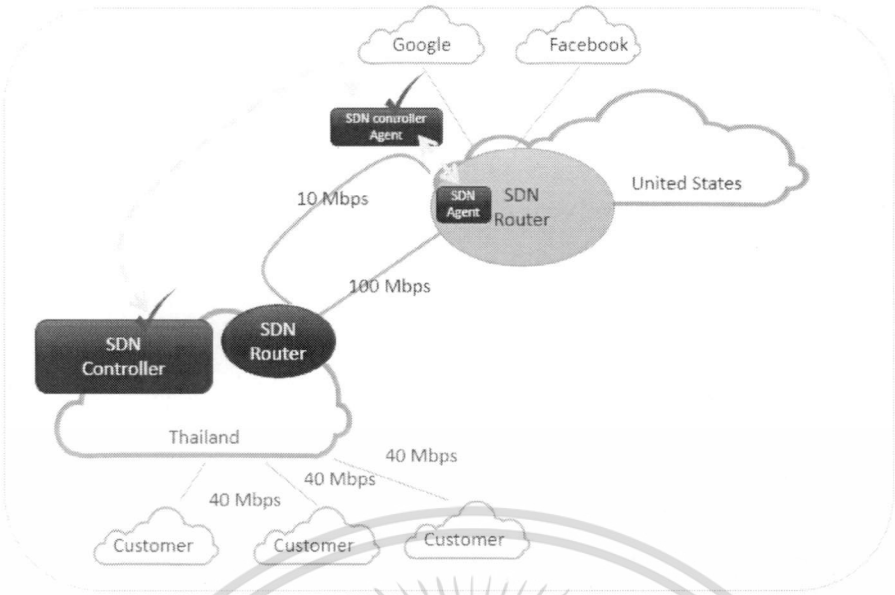


รูปที่ 5.82 การกระจาย หน่วยทำงานของคอนโทรลเลอร์

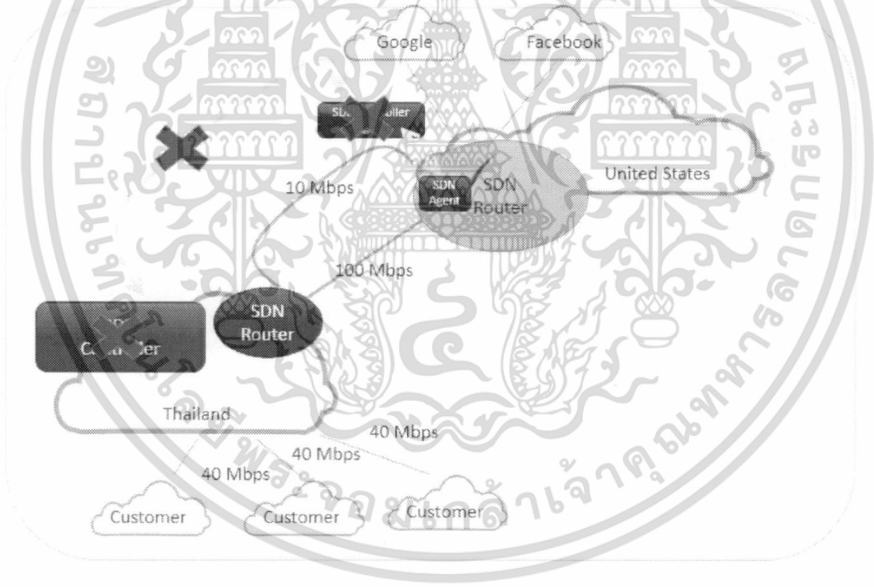
5.4.3. ปัญหาที่เกิดจากคอนโทรลเลอร์ไม่สามารถติดต่อกับอุปกรณ์เครือข่าย SDN

การติดต่อสื่อสารระหว่างอุปกรณ์เครือข่าย SDN และคอนโทรลเลอร์ถือว่าเป็นสิ่งสำคัญ เพราะหากอุปกรณ์เครือข่าย SDN ไม่มีคอนโทรลเลอร์ก็อาจจะไม่สามารถทำงานได้หรือประสิทธิภาพในการทำงานลดลงอย่างมาก ซึ่งปัญหาที่ส่งผลให้เกิดเหตุการณ์ดังกล่าวมีจำนวนมากและไม่สามารถคาดเดาได้ เช่น ช่องทางในการส่งข้อมูลเสียหายทำให้ไม่สามารถติดต่อสื่อสารกันได้ระหว่างคอนโทรลเลอร์และอุปกรณ์เครือข่าย SDN เกิดความผิดพลาดในการส่งข้อมูล ทำให้ข้อมูลเกิดการสูญหายหรือข้อมูลไม่ถูกต้อง เกิดความเสียหายกับฮาร์ดแวร์ที่ทำหน้าที่เป็นคอนโทลเลอร์ทำให้ระบบคอนโทรลเลอร์ไม่สามารถทำงานได้ จากเหตุการณ์ดังกล่าวผู้จัดทำขอเสนอแนวคิด คือ การเพิ่มหน่วยทำงานคอนโทรลเลอร์ลงไปที่ อุปกรณ์เครือข่ายเพื่อวัตถุประสงค์กรณีขาดการติดต่อจากคอนโทรลเลอร์อุปกรณ์เครือข่ายนั้นๆ ยังคงสามารถทำงานได้เป็นปกติ โดยคงสถานะการทำงาน เป็นสถานะล่าสุดที่มีการรับคำสั่งมาจาก คอนโทรลเลอร์ สามารถแสดงได้ดังรูปที่ 5.83 และ 5.84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.83 หน่วยทำงานคอนโทรลเลอร์ บนอุปกรณ์เครือข่าย



รูปที่ 5.84 หน่วยทำงานคอนโทรลเลอร์บนอุปกรณ์เครือข่ายกรณีคอลโทรลเลอร์หลักมีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
บทสรุป

6.1 สรุปผลการวิเคราะห์และออกแบบ

เทคโนโลยี SDN คือ การทำงานของอุปกรณ์เครือข่ายซึ่งมีการแยกการทำงานกันระหว่างการทำงานของ Data plane และ Control plane โดยจะเน้นการสั่งงานจากศูนย์กลางด้วยอุปกรณ์คอนโทรลเลอร์ส่งไปยัง ในส่วนของ Data plane หรือ อุปกรณ์เครือข่ายในหลากหลาย ที่ตั้ง ผ่านระบบเครือข่ายดาต้า โดยรูปแบบการสั่งงานจะอยู่ในรูปแบบมาตรฐานซึ่งเรียกว่า Openflow โพรโทคอลซึ่งการสั่งงานลักษณะนี้จะเป็นการช่วยในการส่งคำสั่งและรับค่าคอนฟิกูเรชันในรูปแบบมาตรฐานไปยังอุปกรณ์เครือข่ายที่มีความแตกต่างกันในด้านของผู้ผลิต และในส่วนของระบบการทำงาน จะเป็นการส่งคำสั่งในรูปแบบเดียวกันทั้งหมดทั้งเครือข่ายที่รองรับเทคโนโลยี SDN

ระบบเครือข่ายของผู้ให้บริการอินเทอร์เน็ตนั้น จะประกอบไปด้วยอุปกรณ์เครือข่ายจำนวนมาก และมีความแตกต่างกันของเจ้าของผลิตภัณฑ์อุปกรณ์เครือข่าย รวมทั้งความแตกต่างกันในส่วนของระบบการทำงาน ซึ่งจะส่งผลกระทบต่อไปยังการบริหารจัดการอุปกรณ์เหล่านั้น เนื่องมาจากรูปแบบของการสั่งงานที่มีความแตกต่างกัน จะส่งผลต่อในการตรวจสอบแก้ไขปัญหาในกรณีเหตุเสียหรือเหตุการณ์อื่นๆที่เกี่ยวข้องกับอุปกรณ์เครือข่ายเหล่านั้น ดังนั้น SDN จึงมีบทบาทเข้ามาช่วยจัดการแก้ไขปัญหาดังกล่าว อันเกิดมาจากอุปกรณ์เครือข่ายที่มีความหลากหลายรูปแบบ ของการสั่งงาน การทำงานร่วมกัน ด้วยการนำ Openflow โพรโทคอลมาบริหารจัดการกลุ่มของอุปกรณ์ดังกล่าว ด้วยการสั่งงานจากคอนโทรลเลอร์ที่อยู่ศูนย์กลางออกไปที่อุปกรณ์เครือข่ายในเครือข่าย ส่งผลให้ระบบการบริหารจัดการ บริการหรือสั่งงานไปยังอุปกรณ์เครือข่ายได้สะดวกขึ้น และสามารถเพิ่มประสิทธิภาพให้กับระบบตรวจสอบสถานะและค่าต่างๆของอุปกรณ์เครือข่ายในระบบเครือข่ายได้สะดวกมากด้วยการตรวจสอบจากอุปกรณ์ คอนโทรลเลอร์ที่ศูนย์กลางของระบบเครือข่าย SDN แต่สิ่งหนึ่งที่ต้องทำการวิเคราะห์ต่อไป คือ ความรวดเร็วในการรับส่งข้อมูลโดยใช้ SDN จะเพิ่มขึ้นหรือลดลง เนื่องจาก ประสิทธิภาพโดยรวมของระบบจะดีขึ้นหรือไม่ ขึ้นอยู่กับปริมาณข้อมูลและความสามารถในการรองรับข้อมูลของเครือข่าย ณ ขณะนั้น

6.2 ปัญหาและข้อจำกัด

1) อุปกรณ์ ที่เป็นฮาร์ดแวร์ที่รองรับ SDN และ Openflow โพรโทคอล ปัจจุบันมีบริษัทที่จัดจำหน่ายจำนวนน้อย โดยส่วนมากบริษัทผู้ผลิตจะแจ้งว่าอยู่ระหว่างการทดลอง เพื่อให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รองรับและเป็นลักษณะ Roadmap ที่ทางบริษัทตั้งกำหนดการณ์ไว้ในอนาคต ด้วยปัญหานี้จึงต้องใช้ซอฟต์แวร์จำลองระบบเครือข่าย SDN คือ Mininet มาช่วยในการทดสอบ

2) ซอฟต์แวร์คอนโทรลเลอร์ปัจจุบันยังมีความหลากหลายมาก และยังคงไม่มีการกำหนดมาตรฐานที่ชัดเจน ส่งผลให้การหาข้อมูลในรายละเอียดที่เกี่ยวข้องได้น้อย

3) ซอฟต์แวร์ Mininet เนื่องด้วยมีการทำงานในลักษณะของ Virtualization ที่ชื่อว่า Virutal box ส่งผลให้ ค่าผลการทดลองจาก Mininet มีความไม่เสถียร

6.3 ข้อเสนอแนะ

จากอุปสรรคที่พบในการศึกษาเรื่องเทคโนโลยี SDN พบว่า SDN ในปัจจุบันขอบเขตการทำงานจะอยู่ในส่วนของอุปกรณ์สวิตซ์เป็นส่วนมาก ในส่วน SDN ที่ทำงานกับเร้าเตอร์ส่วนมากจะเป็นลักษณะซอฟต์แวร์ทดสอบหรือเป็นลักษณะซอฟต์แวร์สำหรับนักพัฒนาเพื่อใช้ในการทดสอบ ในส่วนของบริษัทเชิงพาณิชย์ก็เช่นกันจะมีการทำซอฟต์แวร์เพื่อทำการจำลองระบบเครือข่ายที่เป็นอุปกรณ์เร้าเตอร์ เช่น CiscoOnePK

ดังนั้นการศึกษาเรื่องเทคโนโลยี SDN ในครั้งต่อไป ควรมีการศึกษา ควบคู่กับไประหว่าง SDN ที่ทำงานบนอุปกรณ์สวิตซ์และอุปกรณ์เร้าเตอร์ ในเชิงของพาณิชย์และรูปแบบเปิดเผยให้มีการใช้งานแบบไม่มีค่าใช้จ่าย เพื่อเป็นการศึกษาแนวการทำงาน แนวคิด ในระบบเครือข่าย SDN มากยิ่งขึ้นจะเป็นประโยชน์ในการวิเคราะห์และปรับใช้กับระบบเครือข่ายได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

วิรินทร์ เมฆประดิษฐสิน. 2014. “รู้จักกับโปรโตคอล Openflow.” **ไมโครคอมพิวเตอร์**. 347: 81-90.

Chih-Heng Ke. **Software Defined Network (SDN) Mininet Learning Guide.** [Online] Available: http://csie.nqu.edu.tw/smallko/sdn/sdn.htm

Mark Gates. **What is Iperf?.** [Online] Available: https://iperf.fr/

Murphy McCauley. 2015. **POX Wiki.** [Online] Available: https://openflow.stanford.edu/display/ONL/POX+Wiki

Nick Feamster. 2014. **Software Defined Networking.** [Online] Available: https://www.coursera.org/course/sdn1

Nick McKeown. 2008. **OpenFlow: Enabling Innovation in Campus Networks.** [Online] Available: http://archive.openflow.org/documents/openflow-wp-latest.pdf

Opennetwork Organization. 2012. **OpenFlow Switch Specification.** [Online] Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.1.pdf

Thomas D. Nadeau, Ken Gray. 2013. **SDN: Software Defined Networks**. USA: O'Reilly Media.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายมนัสชัย ใจเอื้อ
วันเดือนปีเกิด	24 กุมภาพันธ์ 2527
สถานที่เกิด	ฉะเชิงเทรา
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตรบัณฑิต (คอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีที่สำเร็จการศึกษา	2548
การทำงาน	วิศวกร ส่วนงานอินเทอร์เน็ตเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้