การเฝ้าระวังการใช้บริการทางด่วนข้อมูลผ่านทางเครือข่าย

นายธีรชัย สันติวิชัยพานิช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2558

UniNet Express Lane Monitoring

Mr. Theerachai Suntiwichaipanich

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF BACHELOR OF COMPUTER ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

ACADEMIC YEAR 2015

ปริญญานิพนธ์เรื่อง	: การเฝ้าระวังการใช้บริการทางด่วนข้อมูลผ่านทางเครือข่าย
ชื่อ	: นายธีรชัย สันติวิชัยพานิช
สาขาวิชา	: วิศวกรรมคอมพิวเตอร์
ภาควิชา	: วิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะ	: วิศวกรรมศาสตร์
อาจารย์ที่ปรึกษา	: รองศาสตราจารย์ ดร.วรา วราวิทย์
ปีการศึกษา	: 2558

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ อนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....	หัวหน้าภาควิชาวิศวกรรมไฟฟ้า
(ผู้ช่วยศาสตราจารย์ ดร. นภดล วิวัชรโกเศศ)	และคอมพิวเตอร์
.....	ประธานกรรมการ
(รองศาสตราจารย์ ดร.วรา วราวิทย์)	
.....	กรรมการ
(รองศาสตราจารย์ ดร.ณชล ไชยรัตนะ)	
.....	กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วรัญญู วงษ์เสรี)	

ลิขสิทธิ์ของภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

Projected Report Title	: UniNet Express Lane Monitoring
Name	: Mr. Theerachai Suntiwichaipanich
Major Field	: Computer Engineering
Department	: Electrical and Computer Engineering
Faculty	: Engineering
Project Advisor	: Assoc. Prof. Dr. Vara Varavithya
Academic Year	: 2015

Accepted by the Faculty of Engineering, King Mongkut's University of Technology North Bangkok in Partial Fulfillment of the Requirements for the Degree of Bachelor of Computer Engineering.

…….	Chairperson of Department of Electrical
(Asst. Prof. Dr. Noppadol Wiwatcharagoses)	and Computer Engineering
…….	Chairperson
(Assoc. Prof. Dr. Vara Varavithya)	
…….	Member
(Assoc. Prof. Dr. Nachol Chaiyaratana)	
…….	Member
(Asst. Prof. Dr. Waranyu Wongseree)	

บทคัดย่อ

การให้บริการทางด่วนข้อมูล (UninNet ExpressLane) ผ่านทางเครือข่าย มักพบปัญหาต่างๆ มากมาย อาทิเช่น ไม่สามารถตรวจสอบสถานะของโหนดต่าง ๆ ได้ว่าสามารถใช้งานได้อยู่หรือไม่ หรือการใช้งานทรัพยากรมาก ซึ่งอาจเป็นสาเหตุทำให้ระบบเกิดความล้มเหลว ผู้จัดทำจึงได้พัฒนาระบบเฝ้าระวังทางเครือข่าย (UniNet ExpressLane Monitoring) มาประยุกต์ทำงานร่วมกับระบบการให้บริการทางด่วนข้อมูล เพื่อช่วยในการเก็บข้อมูลสถานะของอุปกรณ์ทางเน็ตเวิร์คต่าง ๆ เพื่อนำมาใช้ในการวิเคราะห์ เพื่อแก้ไขปัญหาดังกล่าว รวมทั้งสามารถดูเส้นทางการไหลข้อมูลได้อีกด้วย ซึ่งสามารถจำแนกออกเป็น 3 ส่วน ดังนี้ 1. ตรวจสอบสถานะของโหนดต่าง ๆ ที่เชื่อมต่ออยู่กับคอนโทรลเลอร์, 2. ตรวจสอบทรัพยากรต่าง ๆ ของแต่ละโหนด, 3. ตรวจสอบรายละเอียดของเส้นทางการไหลข้อมูล โดยใช้เครื่องมือสำหรับเฝ้าระวัง Nagios XI สำหรับเฝ้าระวังทรัพยากรของโหนด เนื่องจากมี Backend API ให้นำข้อมูลมาประยุกต์ใช้งานได้ และใช้ Ryu Rest API ซึ่งเป็นเซอร์วิสหนึ่งในแอพพลิเคชั่นของคอนโทรลเลอร์สำหรับตรวจสอบสถานะของโหนดที่เชื่อมต่อกับคอนโทรลเลอร์ และยังสามารถดูรายละเอียดของเส้นทางการไหลข้อมูลได้อีกด้วย จากการทดสอบระบบเฝ้าระวังจากการต่อเชื่อมในกรณีต่าง ๆ พบว่าระบบเฝ้าระวังสามารถทำงานได้ถูกต้องและมีการอัพเดทสถานะได้ตรงตามเวลาที่กำหนดไว้ ซึ่งสามารถใช้เป็นข้อมูลในการวิเคราะห์หาสาเหตุในกรณีที่ระบบล้มเหลวหรือใช้ในการวิเคราะห์เพื่อปรับปรุงระบบให้บริการทางด่วนข้อมูลให้มีประสิทธิภาพมากยิ่งขึ้น

Abstract

There are a lot of problem in UniNet ExpressLane Service such as a can not monitoring flow, node connected or monitoring resource which are cause the system fail. So we attend to apply network monitoring to the UniNet ExpressLane Service to monitor node connected or resource usage or service usage divide into three part : 1. Monitoring node connected, 2. Monitoring node resource, 3. Monitoring detail of flow with monitoring tools, use Ryu rest API to monitoring node connected and flow of Openflow Switch, Nagios XI to monitoring resource and service on node. The result of test in each case found UniNet ExpressLane Monitoring can monitor and display correctly and on time. Use to analysis for ploblem when system fail and improve system to the highest efficiency.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้ข้าพเจ้าได้ทุ่มเททักษะ ความสามารถ เพื่อทำให้ผลงานนี้มีคุณภาพและมีประโยชน์สูงสุดต่อผู้ที่ได้ศึกษา ปริญญานิพนธ์นี้ไม่อาจสำเร็จได้โดยปราศจากบุคคลที่คอยให้ความช่วยเหลือต้องขอขอบพระคุณบุคคลเหล่านั้นมา ณ โอกาสนี้

กราบขอบพระคุณ รองศาสตราจารย์ ดร.วรา วราวิทย์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ที่ได้ให้ความรู้ คำปรึกษา คำแนะนำ ข้อคิดเห็น ข้อมูล และการสนับสนุนอย่างเต็มที่ ซึ่งเป็นประโยชน์อย่างยิ่งสำหรับปริญญานิพนธ์เล่มนี้

กราบขอบพระคุณอาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ทุกท่านที่ได้ให้วิชาความรู้ในวิชาชีพด้านวิศวกรรมและกำลังใจในการแก้ไขปัญหาในด้านการศึกษาตลอด 4 ปีที่ผ่านมา ทำให้ข้าพเจ้าสามารถนำความรู้ที่เรียนมาและทักษะการดำเนินชีวิตประจำวันมาประยุกต์ใช้ในปริญญานิพนธ์ได้เป็นอย่างดี

กราบขอบพระคุณ คุณพ่อและคุณแม่ที่ให้การสนับสนุนและความห่วงใยต่อข้าพเจ้าตลอดมา ทำให้ข้าพเจ้าประสบความสำเร็จในการศึกษาไปได้ด้วยดี

ขอบคุณรุ่นพี่ รุ่นน้อง และเพื่อนทุกคนสำหรับคำแนะนำและความช่วยเหลือที่ทำให้ข้าพเจ้าทำงานได้อย่างมีประสิทธิภาพ

ธีรชัย สันติวิชัยพานิช

สารบัญ

สารบัญ

สารบัญตาราง

สารบัญภาพ

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

บทที่ 1
บทนำ

การเฝ้าระวังหรือการดูแลระบบเป็นงานสำคัญอย่างหนึ่งสำหรับผู้ดูแลระบบเครือข่าย เนื่องจากเป็นตัวช่วยในการวิเคราะห์และแก้ไขปัญหาเมื่อเกิดปัญหาขึ้น ซึ่งการใช้เครื่องมือช่วยในการเฝ้าระวังระบบนั้นจะทำให้ผู้ดูแลระบบสามารถรับรู้ เข้าใจถึงปัญหาที่เกิดขึ้นกับระบบได้ง่ายขึ้น ช่วยลดเวลาในการแก้ไขปัญหาเช่น สามารถตรวจสอบได้ว่าขณะนั้นระบบยังทำงานอยู่หรือไม่ หรือระบบล้มเหลวเวลาใดบ้าง สามารถตรวจสอบทรัพยากรหรือการให้บริการเซอร์วิสต่างๆ ในระบบเป็นต้น และนำมาวิเคราะห์ประเมินผล, พยากรณ์, แก้ไขปัญหาหรือปรับปรุงให้ระบบมีความเสถียรมากยิ่งขึ้น

การให้บริการทางด่วนข้อมูลผ่านทางเครือข่าย (UniNet Express Lane Services) ของสำนักงานบริหารเทคโนโลยีสารสนเทศเพื่อพัฒนาการศึกษา (UniNet) เพื่อทำให้สามารถรับส่งข้อมูลได้อย่างเต็มประสิทธิภาพ โดยไม่ผ่านอุปกรณ์ทางเน็ตเวิร์ค เช่น Firewall ซึ่งจะทำให้แพ็กเก็ตถูกตรวจสอบจนเกิดความล้าช้าในการรับส่ง การให้บริการทางด่วนข้อมูลจะทำให้การรับส่งข้อมูลได้เต็มแบนด์วิดท์ โดยอาศัยแนวคิดของการนิยามเครือข่ายโดยซอฟแวร์ (Software Define Network) มาควบคุมโฟล์วหรือเส้นทางการไหลของข้อมูลให้เป็นไปตามความต้องการของผู้ใช้งาน ซึ่งจะมีหน้าเว็บสำหรับให้ผู้ใช้งานขอใช้บริการหรือตรวจสอบสถานะการใช้บริการ และเพื่อให้ระบบให้บริการทางด่วนข้อมูลมีประสิทธิภาพจึงควรจะมีระบบเฝ้าระวังทางเครือข่ายเข้ามาร่วมในระบบเพื่อตรวจสอบสถานะของโฟล์วของข้อมูลจากการรับส่งระหว่างคู่สนทนาและแสดงผลผ่านทางหน้าเว็บเพื่อให้ผู้ใช้งานหรือผู้ดูแลระบบสามารถตรวจสอบด้วย

ผู้จัดทำได้เล็งเห็นความสำคัญของระบบเฝ้าระวังจึงมีการนำมาพัฒนาร่วมกับระบบการให้บริการทางด่วนข้อมูล (UniNet Express Lane Services) เพื่อให้ทำงานเป็นฟังก์ชันหนึ่งในการตรวจสอบสถานะต่าง ๆ ของระบบ อาทิเช่น ตรวจสอบการทำงานของเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดที่ติดตั้งอยู่ตามจุดต่าง ๆ โดยใช้ Nagios XI ตรวจสอบเส้นทางไหลข้อมูลผ่านบอร์ด Netfpga และตรวจสอบการเชื่อมต่อระหว่างโหนดต่าง ๆ กับเครื่องคอนโทรลเลอร์ ซึ่งจะช่วยให้ผู้ดูแลระบบสามารถนำข้อมูลจากการเฝ้าระวังไปใช้ในการตรวจสอบการทำงานของระบบและปรับปรุงประสิทธิภาพของระบบต่อไป

ปริญญานิพนธ์ฉบับนี้ประกอบด้วยเนื้อหาทั้งหมด 5 บท โดยบทที่ 1 จะกล่าวถึงหลักการเหตุผลและขอบเขตของปริญญานิพนธ์นี้ ซึ่งหลักการและเหตุผลต่าง ๆ ในปริญญานิพนธ์นี้คือระบบให้บริการที่ขาดระบบเฝ้าระวัง จึงเป็นสาเหตุที่ทำให้ต้องมีการนำองค์ความรู้ในของการเฝ้าระวังทางเครือข่ายมาประยุกต์ร่วมกับระบบดังกล่าว เพื่อลดปัญหาความล่าช้าในการแก้ไขปัญหาเมื่อระบบเกิดล่มเหลวและในบทที่ 1 ยังกล่าวถึงภาพรวมของระบบเฝ้าระวังทางเครือข่ายในการให้บริการทางด่วนข้อมูลที่จำเป็นต่อการทำปริญญานิพนธ์นี้

บทที่ 2 ได้กล่าวถึงระบบเฝ้าระวังทางเครือข่ายและองค์ความรู้โดยละเอียด เช่น เครื่องมือที่ใช้ในการเฝ้าระวังรูปแบบในการเฝ้าระวัง และ โปรโตคอลในการเฝ้าระวังทางเครือข่ายขั้นตอนในการเตรียมเครื่องเพื่อติดตั้งเครื่องมือที่ใช้ในการเฝ้าระวังอย่างละเอียด เป็นต้น และยังกล่าวถึงเซอร์วิสสำคัญต่างๆ ที่ใช้ในระบบเฝ้าระวัง ได้แก่ ตารางการทำงานอัตโนมัติ (Crontab) และเป็นคอมมานไลน์ส่วนเสริมที่ติดมากับ Openflowใช้ในการตรวจสอบสถานะของพอร์ตและการไหลของข้อมูลรวมทั้ง Nagios Backend API และ Ryu Rest API ในการเข้าถึงสถานะของโหนดต่าง ๆ

บทที่ 3 นั้นกล่าวถึงการออกแบบระบบเฝ้าระวังทางเครือข่ายในการให้บริการทางด่วนข้อมูลซึ่งแบ่งขั้นตอนการออกแบบได้ 4 ขั้นตอน ดังนี้

1. วิเคราะห์และออกแบบระบบรวม
2. วิเคราะห์และออกแบบระบบเฝ้าระวังคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดต่าง ๆ
3. วิเคราะห์และออกแบบระบบเฝ้าระวังบอร์ด NetFPGA
4. วิเคราะห์และออกแบบระบบเฝ้าระวังคอนโทรลเลอร์

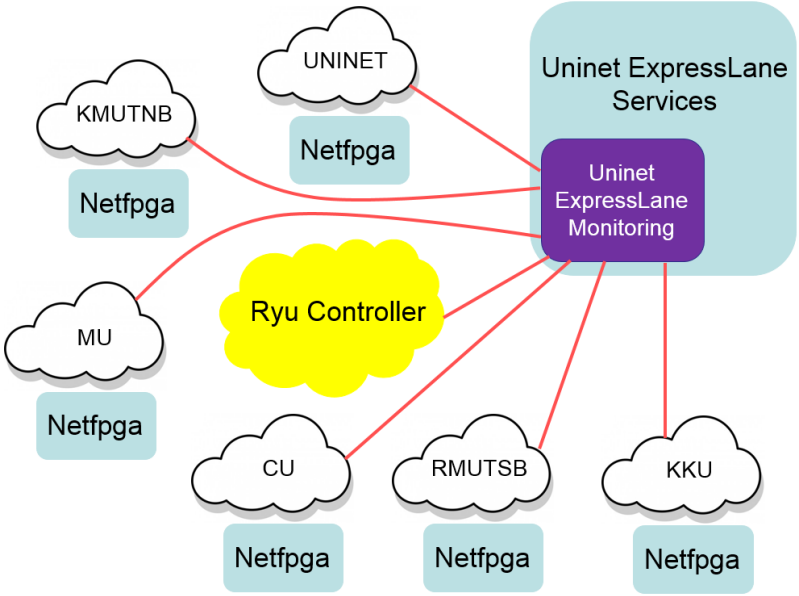
บทที่ 4 กล่าว ถึงผลจากการทดลองตามการออกแบบระบบในบทที่ 3 ซึ่งสามารถแบ่งเป็น 3 ส่วนคือ

1. การเฝ้าระวังสถานะและทรัพยากรของคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดต่างๆ
2. การเฝ้าระวังสถานะและเส้นทางการไหลข้อมูลของบอร์ด NetFPGA
3. การเฝ้าระวังสถานะและเส้นทางการไหลข้อมูลผ่านคอนโทรลเลอร์

อีกทั้งในบทที่ 4 ยังกล่าวถึงการทดสอบประสิทธิภาพของระบบเฝ้าระวังเพื่อให้ระบบมีประสิทธิภาพสูงสุดในทรัพยากรที่กำหนด มีการทดสอบในกรณีต่างๆ ทั้งเหตุการณ์ปกติและเหตุการณ์ไม่ปกติ เพื่อทดสอบว่าระบบยังสามารถทำงานได้ปกติ

และในบทที่ 5 กล่าวถึงบทสรุปของการทำปริญญานิพนธ์ โดยใจความสำคัญจะกล่าวถึงข้อดีของการนำวิธีการที่ปริญญานิพนธ์นี้ได้นำเสนอว่าก่อให้เกิดผลอย่างไรต่อผู้ที่มีส่วนเกี่ยวข้องและมีส่วนช่วยในการแก้ปัญหาอย่างไรดังที่ได้กล่าวมาแล้วข้างต้นว่าในปัจจุบันนั้นมีปัญหาเช่นไรเกี่ยวกับระบบการให้บริการที่ปราศจากระบบเฝ้าระวังทางเครือข่าย ซึ่งเป็นการชี้ให้เห็นถึง

ประโยชน์ที่ได้จากการนำปริญญานิพนธ์นี้ไปประยุกต์ใช้ในระบบการใช้งานจริง อีกทั้งยังบอกถึงข้อเสียและข้อจำกัดที่ยังจะต้องดำเนินการแก้ไขต่อไป พร้อมบอกถึงปัญหา อุปสรรค และขั้นตอนในการแก้ไขปัญหาอุปสรรคนั้นให้สามารถทำงานได้ตามที่วางแผนไว้ในการทำปริญญานิพนธ์นี้



ภาพที่ 1-1 ภาพรวมของระบบเฝ้าระวังการให้บริการทางด่วนข้อมูลผ่านทางเครือข่ายของ UniNet ซึ่งจะประกอบไปด้วย 3 ส่วนหลัก ๆ คือ 1. UninetExpressLaneService 2. RyuController 3. โหนดต่าง ๆ จำนวน 6 โหนดซึ่งทั้ง 3 ส่วนจะถูกเฝ้าระวังโดระบบ UniNetExpressLane Monitoring ที่ได้พัฒนาขึ้น

บทที่ 2
ระบบเฝ้าระวังทางเครือข่าย

ในบทนี้จะกล่าวถึงงานวิจัยหรือบทความที่เกี่ยวข้องกับระบบเฝ้าระวังเพื่อใช้เป็นแหล่งอ้างอิงหรือเป็นแนวคิดในการพัฒนา รวมถึงองค์ความรู้และข้อควรระวังในการติดตั้งซึ่งจะประกอบไปด้วยส่วนหลัก ๆ คือ 1. ระบบเฝ้าระวังคืออะไร มีโครงสร้างของระบบเฝ้าระวังว่าประกอบด้วยส่วนย่อย ๆ อย่างไรบ้าง และมีการทำงานของการเฝ้าระวังอย่างไร 2. เครื่องมือที่ใช้ในการเฝ้าระวังทางเครือข่าย (Nagios XI) รวมทั้งขั้นตอนในการเตรียมเครื่องและการติดตั้ง 3. โปรโตคอลเอ็สเอ็นเอ็มพีที่ใช้ในการติดต่อสื่อสารภายในระบบ 4. คำสั่ง DPCTL ในการตรวจดูสภาพการไหลของข้อมูล 5. เซอร์วิส Crontabในการกำหนดการทำงานอัตโนมัติเพื่ออัพเดทสถานะต่างๆ ไปยังฐานข้อมูล

2.1 งานวิจัยและบทความที่เกี่ยวข้อง

The Network Monitoring Base On Cacti for EAST ได้ใช้เครื่องมือในการเฝ้าระวังทางเครือข่ายชื่อ Cacti ในการเฝ้าระวังระบบ EAST (Experimental Advanced Superconducting Tokamak) ซึ่งเป็นระบบที่ใช้ผลิตพลังงานสะอาดโดยใช้เทคโนโลยีนิวเคลียร์ฟิวชันหรือการจำลองปฏิกิริยาของดวงอาทิตย์ โดยประกอบด้วยระบบย่อย ๆ ที่ทำงานผ่านทางอินเตอร์เน็ตซึ่งต้องการความเชื่อถือ และทำให้แน่ใจว่าระบบนั้นยังสามารถทำงานได้ปกติ เช่น ดูสถานะของอุปกรณ์ต่าง ๆ หรือแบนด์วิดท์ของเครือข่าย และไม่ก่อให้เกิดการล่มเหลวของระบบจนนำไปสู่การเสียเวลาและเสียทรัพยากร ทำให้ได้มีการนำเครื่องมือเฝ้าระวังทางเครือข่ายมาเฝ้าระวัง โดยผู้วิจัยได้เลือกใช้ Cacti โดยให้เหตุผลว่า

1. เป็นเครื่องมือที่สามารถใช้งานได้ฟรี (Open Source) และง่ายต่อการติดตั้ง
2. มีอินเตอร์เฟสที่ใช้งานง่ายสำหรับผู้ใช้
3. มีฟังก์ชันในการจัดการกับผู้ใช้งานและข้อมูลที่ดี

ผลจากงานวิจัยชี้ให้เห็นว่า การใช้เครื่องมือในการเฝ้าระวังทางเครือข่าย Cacti มาช่วยในการเฝ้าระวังระบบ EAST ทำให้ระบบมีประสิทธิภาพมากขึ้น เนื่องจากสามารถนำข้อมูลที่ได้จากการเฝ้าระวังมาวิเคราะห์เพื่อเตรียมรับมือกับความล้มเหลวที่อาจเกิดขึ้นได้

Distributed Event Monitoring for Software Defined Network ได้พัฒนาระบบเฝ้าระวังเพื่อเก็บสถานะของอุปกรณ์ต่าง ๆ ที่เกิดขึ้นภายในระบบ SDN เช่น สถานะ Openflow Switch, สถานะของ Host ซึ่งปกติ Controller จะมีการแลกเปลี่ยนข้อความในการตรวจสอบสถานะของ Switch และ Host อยู่ตลอดเวลา ดังนั้นเมื่อมีเหตุการณ์ผิดปกติเกิดขึ้น เช่น Host offline จะมีการเก็บเหตุการณ์นั้น ๆ ไว้ที่ Controller และส่งไปยัง Logs Server ผ่านทาง syslog protocol เพื่อให้ผู้ดูแลระบบนำไปวิเคราะห์เพื่อปรับปรุงระบบต่อไปโดยผู้วิจัยใช้ Rsyslog มาติดตั้งเป็น Logs Server และคอนฟิคเพื่อให้สามารถเก็บ log จากอุปกรณ์ต่าง ๆ อีกทั้งยังสามารถติดตั้งร่วมกับฐานข้อมูล เช่นMySQL หรือ PostgreSQL ได้ โดยแบ่งตามระดับความรุนแรง ได้แก่ WARN, INFO, ERROR เพื่อลดขนาดข้อมูลในการเก็บ และง่ายต่อการนำมาวิเคราะห์ และทดสอบโดยการปิดการเชื่อมต่อของ Switch หรือ Host ขณะระบบกำลังทำงาน ซึ่งผลสรุปจากงานวิจัยชี้ให้เห็นว่าสามารถเก็บ log จาก Openflow Switch หรือ Host ได้อย่างอัตโนมัติและมีประสิทธิภาพ

2.2 Network Monitoring System

Network Monitoring System (NMS) เป็นระบบที่ทำหน้าที่ในการเฝ้ามองและดูแลระบบพร้อมทั้งเก็บประวัติการใช้งานหรือสถานะต่าง ๆ ของระบบ และนำมาใช้ในการวิเคราะห์ตรวจสอบย้อนหลัง เมื่อเกิดข้อผิดพลาดขึ้นกับระบบ หรือนำมาปรับปรุงการทำงานของระบบให้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งเครื่องมือที่ทำหน้าที่เป็น NMS ที่นิยมใช้กันอย่างแพร่หลาย ได้แก่ Nagios, Cacti, Open NMS ซึ่งในปริญญานิพนธ์นี้ได้เลือกใช้ Nagios ซึ่งเป็นเครื่องมือที่มี API ให้ดึงค่าที่สามารถเฝ้าระวังได้มาแสดงในระบบเฝ้าระวังที่ผู้จัดทำได้พัฒนาขึ้น

2.2.1 Nagios

เป็นซอฟต์แวร์ที่ใช้ในการเฝ้าระวังหรือดูแลระบบเน็ตเวิร์ค (Network Monitoring) ซึ่งสามารถตรวจสอบทรัพยากรพื้นฐานของระบบปฏิบัติการ มีทั้งแบบที่เป็นโอเพ่นซอร์สสามารถใช้งานได้ฟรีคือ Nagios Core และแบบมีค่าใช้จ่าย เช่น Nagios XI, Nagios Network Analyzer, Nagios Log Server เป็นต้นซึ่งมีความแตกต่างกันทางด้านฟีเจอร์และประสิทธิภาพในการจัดการ (ตารางที่ 2-1)

ตารางที่ 2-1 ตารางเปรียบเทียบฟีเจอร์การใช้งานระหว่าง Nagios XI กับ NAgios Core

Feature	Nagios XI	Nagios Core
Web Configuration Interface	✓	✘
Configuration Wizards	✓	✘
Database Backend API	✓	✘

ความสามารถของ Nagios XI

1. สามารถตรวจสอบการให้บริการต่าง ๆ ของเครื่องคอมพิวเตอร์หรืออุปกรณ์ทางด้านเน็ตเวิร์คได้ เช่น PING, HTTP, SSH, TELNET, FTP, SNMP

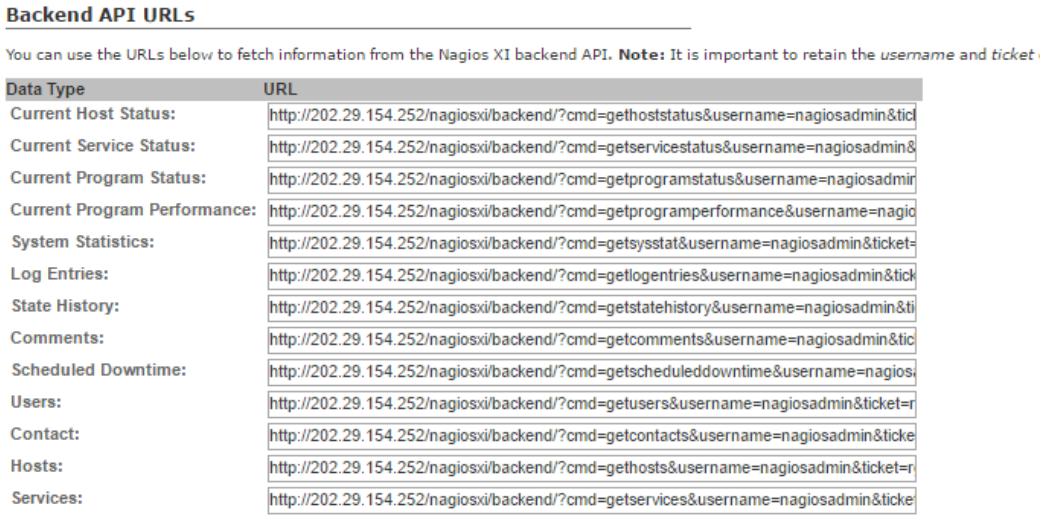
2. สามารถตรวจสอบสถานะการใช้งานทรัพยากรได้ เช่น CPU Usage, Memory, Disk Usage

3. สามารถตั้งระบบแจ้งเตือนเมื่อระบบเกิดความผิดพลาดกับการให้บริการต่าง ๆ เพื่อช่วยให้ผู้ดูแลสามารถแก้ไขได้ทัน ก่อนจะเกิดความเสียหาย

4. สามารถใช้ NagiosXIBackend API ในการดึงข้อมูล-สถานะ ในการเฝ้าระวังอุปกรณ์ขณะนั้นมาใช้งานร่วมกับแอพพิเคชั่นอื่น ๆ ได้

NagiosXI Backend API

Nagios XI ได้พัฒนา API ให้ผู้ใช้สำหรับนำเอาสถานะต่าง ๆ ที่ได้จากการเฝ้าระวังไปประยุกต์ใช้งานหรือแสดงผลร่วมกับแอพพิเคชั่นอื่น ๆ ในลักษณะของ Third-Party ผ่านทางรูปแบบของไฟล์ XML โดยผู้ใช้สามารถเลือกประเภทของข้อมูลเพื่อนำไปประยุกต์ใช้ได้ เช่น Current Host Status จะแสดงสถานะของ Host ที่ได้เฝ้าระวังอยู่ขณะนั้น หรือ Current Service Status จะแสดงสถานะของทรัพยากรและเซอร์วิสต่าง ๆ ที่ได้เฝ้าระวังอยู่เครื่องนั้นๆ ตามภาพที่ เป็นต้น ซึ่งการใช้งาน Nagios XI Backedn API ร่วมกับระบบ UninetExpressLane Monitoring จะกล่าวถึงในบทที่ 3



Backend API URLs

You can use the URLs below to fetch information from the Nagios XI backend API. **Note:** It is important to retain the *username* and *ticket*

Data Type	URL
Current Host Status:	http://202.29.154.252/nagiosxi/backend/?cmd=gethoststatus&username=nagiosadmin&tic
Current Service Status:	http://202.29.154.252/nagiosxi/backend/?cmd=getservicestatus&username=nagiosadmin&
Current Program Status:	http://202.29.154.252/nagiosxi/backend/?cmd=getprogramstatus&username=nagiosadmir
Current Program Performance:	http://202.29.154.252/nagiosxi/backend/?cmd=getprogramperformance&username=nagio
System Statistics:	http://202.29.154.252/nagiosxi/backend/?cmd=getsysstat&username=nagiosadmin&ticket=
Log Entries:	http://202.29.154.252/nagiosxi/backend/?cmd=getlogentries&username=nagiosadmin&tick
State History:	http://202.29.154.252/nagiosxi/backend/?cmd=getstatehistory&username=nagiosadmin&ti
Comments:	http://202.29.154.252/nagiosxi/backend/?cmd=getcomments&username=nagiosadmin&tic
Scheduled Downtime:	http://202.29.154.252/nagiosxi/backend/?cmd=getscheduleddowntime&username=nagios
Users:	http://202.29.154.252/nagiosxi/backend/?cmd=getusers&username=nagiosadmin&ticket=r
Contact:	http://202.29.154.252/nagiosxi/backend/?cmd=getcontacts&username=nagiosadmin&ticke
Hosts:	http://202.29.154.252/nagiosxi/backend/?cmd=gethosts&username=nagiosadmin&ticket=r
Services:	http://202.29.154.252/nagiosxi/backend/?cmd=getservices&username=nagiosadmin&ticke

ภาพที่ 2-1 แสดงประเภทข้อมูลรวมทั้ง URL ในการเข้าถึง Nagios XI Backend API เพื่อนำค่าไปประยุกต์ใช้และแสดงผลร่วมกับแอพพลิเคชั่นอื่น

แพลตฟอร์มที่รองรับการทำงานของ Naigos

- RHEL 5 & 6 32-bit and 64-bit, and RHEL 7 (requires RHN registration)
- CentOS 5 & 6 32-bit and 64-bit, and CentOS 7

ขั้นตอนการเตรียมเครื่องคอมพิวเตอร์และติดตั้ง Nagios XI

ในการติดตั้ง Nagios XI คอมพิวเตอร์ที่จะติดตั้งจะต้องเป็นเครื่องที่ใหม่ คือยังไม่มีการติดตั้งหรือปรับปรุงแก้ไขแพ็คเก็ตต่างๆ ควรจะเป็นเครื่องที่เพิ่งจะลงระบบปฏิบัติการใหม่ซึ่งมีขั้นตอนการติดตั้งดังนี้

ขั้นตอนที่ 1 : ติดตั้งระบบปฏิบัติการ Ubuntu Server 14.04

ขั้นตอนที่ 2 : ดาวน์โหลด Nagios XI เวอร์ชันล่าสุด

wgethttp://assets.nagios.com/downloads/nagiosxi/xi-latest.tar.gz

ขั้นตอนที่ 3: แตกไฟล์

tar xzf xi-latest.tar.gz

ขั้นตอนที่ 4 : หลังจากแตกไฟล์จะได้โฟลเดอร์ nagiosxi จากนั้นย้ายไดเรคทอรี่ข้าไปในโฟเดอร์nagiosxiโดยพิมพ์

cd nagiosxi

ขั้นตอนที่ 5: เริ่มติดตั้ง Nagois Xi โดยพิมพ์

./fullinstall

ขั้นตอนที่ 6 : รอจนกระทั่งติดตั้งเสร็จ อาจใช้เวลา 5-10 นาที เนื่องจาก Nagios XI จะต้องติดตั้งแพ็คเก็ตเพิ่มเติมที่จำเป็นต้องใช้ในการ Monitor

ขั้นตอนที่ 7 : หลังจากการติดตั้งเสร็จสมบูรณ์ สามารถเริ่มต้นการใช้งานโดย

เข้าไปที่ URL http://<ipnagiosxiserver>/

ขั้นตอนที่ 8 : หน้าแรกเมื่อเข้ามาจะพบหน้าให้กำหนดรหัสผ่านใหม่เพื่อเข้าใช้งาน

2.3 SNMP

SNMP ซึ่งเป็นโปรโตคอลที่ใช้ในการจัดการดูแลอุปกรณ์บนระบบเครือข่าย เช่น Switch, Router, Computer เป็นต้น ซึ่งในการเฝ้าระวังระบบจะสื่อสารกันระหว่างเครื่องที่ทำหน้าที่เป็น Server (Nagios XI) กับเครื่องที่เป็น Client (Host) ผ่านทางโปรโตคอล SNMP ดังนั้นจึงจำเป็นจะต้องติดตั้ง SNMP หลังจากที่ติดตั้ง Naigos XI เสร็จเรียบร้อยแล้ว

2.3.1 โครงสร้างส่วนประกอบและการทำงานของ SNMP ประกอบด้วย 3 ส่วน ได้แก่

1. SNMP Manager : เป็นส่วนของซอฟต์แวร์ที่อยู่ในอุปกรณ์ที่เป็นตัวจัดการแลบริหารเครือข่าย
2. SNMP Agent : เป็นส่วนของซอฟต์แวร์ที่อยู่ในอุปกรณ์เป้าหมายที่ต้องการเฝ้าดูแล
3. SNMP MIB : เป็นชุดข้อมูลการทำงานที่ใช้ในการแลกเปลี่ยนระหว่าง Agent กับ Manager

2.3.2 ขั้นตอนการทำงานของ SNMP

1. SNMP Manager จะสร้างคำร้องขอ (Getrequest) ประกอบด้วย ชื่อ MIB และส่งไปยัง SNMP Agent
2. SNMP Agent รับคำร้องขอ และนำชื่อ MIB มาตรวจสอบและค้นหาข้อมูลของ MIB นั้น
3. SNMP Agent สร้างข้อมูลการตอบกลับ (Getresponse) และส่งกลับไปยัง SNMP Manager

2.3.3 ขั้นตอนในการติดตั้ง SNMP

ขั้นตอนที่ 1 : # sudo apt-get install net-snmp net-snmp-utils

ขั้นตอนที่ 2 : # mv /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.orig

ขั้นตอนที่ 3 : # vim /etc/snmp/snmpd.conf

ขั้นตอนที่ 4 : # rocommunitypublic
syslocation Bangkok
syscontact trachiiz@gmail.com

ขั้นตอนที่ 5 : # service snmp restart

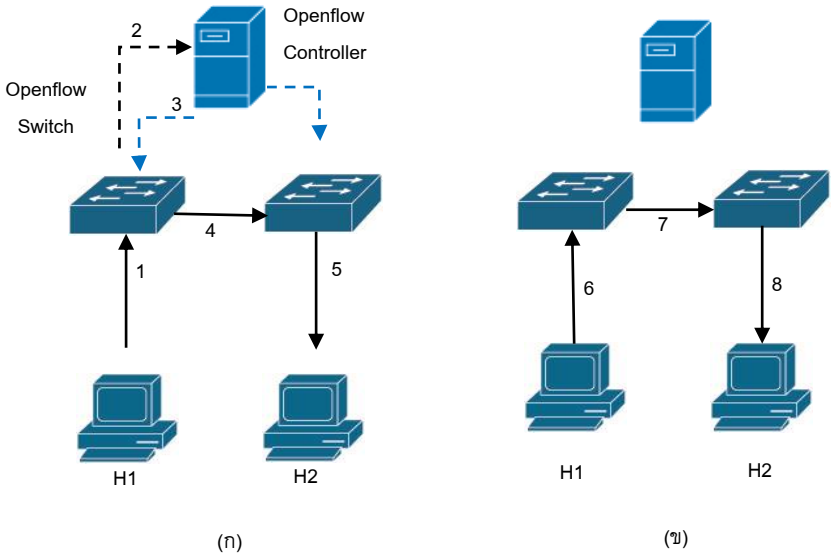
ในการเชื่อมต่อระหว่าง Server กับ Client ทั้งสองฝั่งจะต้องติดตั้ง Service snmp ด้วย ดังนั้นจึงต้องติดตั้ง snmpให้กับเครื่อง Client ซึ่งวิธีการในการติดตั้งนั้นเหมือนกับการติดตั้งในฝั่ง Server โดยจะต้องกำหนด rocommunity ให้ตรงกันจึงจะสามารถสื่อสารกันได้

ทดสอบการสื่อสารผ่าน SNMP

snmpwalk -v1 –c public localhost

2.4 Openflow 1.0

OpenFlow คือโปรโตคอลในเลเยอร์ 2 หรือดาต้าลิ้งเลเยอร์ (OSI Model) ดังนั้น OpenFlow จะทำหน้าที่ในการส่งต่อเฟรมจากสวิตช์หนึ่งไปยังอีกสวิตช์ โดยที่เราไม่จำเป็นต้องเข้าไปตั้งค่าสวิตช์แต่ละตัวโดยตรง แต่จะทำการตั้งค่าจากส่วนกลางหรือคอนโทรลเลอร์ สามารถอธิบายให้เข้าใจได้ดังภาพที่ 2-2 อีกทั้ง OpenFlow ยังถูกออกแบบมาเพื่อให้ใช้งานได้กับสวิตช์จากผู้ผลิตใด ๆ ก็ได้ที่รองรับ Openflow โดยไม่ยึดติดกับ ยี่ห้อหรือรุ่นของสวิตช์ เพราะอุปกรณ์ทั้งหมดสามารถคุยเข้าใจกันได้ผ่านโปรโตคอลเดียวกันทำให้โปรโตคอล Openflow ถูกนำมาใช้ในระบบ Software Define Networkโดยทำหน้าที่เป็นตัวการสื่อสารระหว่างเครื่องควบคุมกับ Openflow Switch ในปัจจุบัน Openflow ได้รู้จักอย่างแพร่หลายมากยิ่งขึ้นและมีแนวโน้มที่ระบบ Openflow จะถูกนำมาเป็นมาตรฐานใหม่ในระบบเครือข่ายในอนาคต Openflow มีการพัฒนาออกมาหลายเวอร์ชั่นตั้งแต่ 1.0–1.5 ซึ่งในปริญญานิพนธ์นี้ได้เลือกใช้เวอร์ชั่น 1.0 เนื่องจากเป็นเพียงเวอร์ชั่นเดียวที่สามารถทำงานร่วมกับ NetFPGA 1G ได้



(ก) (ข)

ภาพที่ 2-2 ขั้นตอนการทำงานของโอเพ่นโฟลว์ซึ่งบ่งบอกขั้นตอนต่าง ๆ ในการที่โอเพ่นโฟลว์สวิตช์ได้รับตารางการไหลใหม่ๆเข้ามาจะต้องทำการติดต่อไปยังเครื่องควบคุมเพื่อให้พิจารณาว่าจะกระทำอย่างไรกับตารางการไหลที่ได้รับมา (ตามภาพ ก) จากนั้นจึงเพิ่มโฟลว์ หรือเส้นทางการไหลของข้อมูลไปยังโอเพ่นโฟลว์สวิตช์ทั้งสองตัว เมื่อสวิตช์ทั้งสองตัวมีโฟล์วแล้วเมื่อมีแพ็คเก็ตส่งมาจึงสามารถผลักแพ็คเก็ตตผ่านไปยังโฮสต์ต่อไปได้เลย โดยไม่ต้องผ่านคอนโทรลเลอร์ (ตามภาพ ข)

2.5 DPCTL

เป็นคอมมานไลน์ส่วนเสริมที่ติดมากับ Openflow มีประโยชน์เพื่อใช้ในการตรวจสอบสถานะของพอร์ตและการไหลของข้อมูลผ่านทาง Openflow message อีกทั้งยังสามารถเพิ่มเส้นทางการไหลของข้อมูลได้ทำให้สะดวกสบายต่อผู้ดูแลในการเฝ้าระวังระบบ (Monitoring) มากยิ่งขึ้น โดยมีรูปแบบของคำสั่งดังนี้

dpctl [OPTIONS] SWITCH COMMAND [ARG...]

OPTIONS จะประกอบไปด้วย

- show switch : show basic imfomation
- status switch [KEY] : report statistics
- dump-desc switch : print switch description

- dump-tables switch : print table stats
- dump-flows switch : print all flow entries
- add-flow switch flow : add flow described by flow
- del-flows switch flow : delete matching flow

ซึ่งหากต้องการหาคำสั่งเพิ่มเติม สามารถพิมพ์คำสั่ง dpctl –h เพื่อแสดงคำสั่งทั้งหมดได้
วิธีการเชื่อมต่อประกอบไปด้วย 3 วิธี ได้แก่

1. tcp:HOST[:PORT] PORT (default: 6633) on remote TCP HOST
2. unix:FILE Unix domain socket named FILE
3. fd:NFile descriptor N

2.6 Crontab

เป็นคำสั่งที่ติดมาพร้อมกับระบบปฏิบัติการลีนุกซ์ ช่วยในการกำหนดเวลาให้ทำงานต่าง ๆ อัตโนมัติ โดยไม่ต้องสั่งงานนั้นซ้ำ ๆ ซึ่งเหมาะสำหรับงานที่ต้องทำเป็นประจำทุกๆ นาที, ทุก ๆ ชั่วโมง หรือ ทุก ๆ วันเช่น การสั่งรันสคริปในการ monitoring ทุกๆ 1 นาที
คำสั่งพื้นฐาน Crontab
crontab –e ใช้สำหรับเพิ่มหรือแก้ไข เวลา และงาน
crontab –l ใช้สำหรับแสดงงานทั้งหมดที่กำหนดเอาไว้ใน crontab

รูปแบบของคำสั่ง crontab ประกอบไปด้วย 6 ส่วนดังนี้

** * * * ** command to be executed

* = minute มีค่า 0 - 59 เวลาเป็นนาที จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงนาทีที่กำหนด
* = hour มีค่า 0 - 23 เวลาเป็นชั่วโมง จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงชั่วโมงที่กำหนด
* = day มีค่า 1 - 31 เวลาเป็นวัน จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงวันที่กำหนด
*= month มีค่า 1 - 12 เวลาเป็นเดือน จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงเดือนที่กำหนด
*= weekday มีค่า 0 - 6 วันของแต่ละสัปดาห์ มีค่าดังนี้ อาทิตย์ = 0, จันทร์ = 1, อังคาร = 2, พุธ = 3, พฤหัส = 4, ศุกร์ = 5,เสาร์ = 6
*= command คือ งานหรือ script ต่าง ๆ ที่ต้องการทำเมื่อถึงเวลาที่กำหนดไว้

ทุกครั้งที่มีการเพิ่มหรือแก้ไขงานใน crontab ขั้นตอนสุดท้ายคือ การรีสตาร์ทเซอร์วิสเพื่อเริ่มดำเนินการงานที่ได้เพิ่มหรือแก้ไขใหม่โดยการพิมพ์คำสั่ง

```
# service crond restart
```

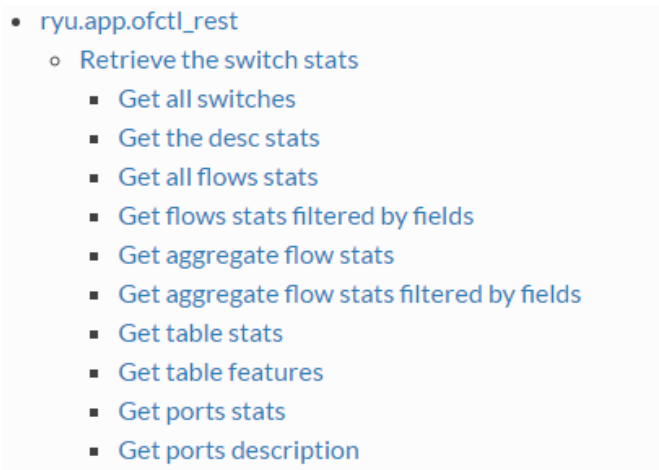
2.7 Ryu Controller

Ryu ออกเสียงว่า “รี-ยูว” เป็นภาษาญี่ปุ่น ซึ่งมีความหมายว่า “การไหล” Ryu เป็น Controller ที่ใช้ในการจัดการเครือข่ายตามแนวคิดของ Software Define Network (SDN) ซึ่งจะทำหน้าที่ในการควบคุมและจัดการการไหลของข้อมูลในระบบเครือข่าย เช่น เมื่อมี packet วิ่งผ่านเข้ามาใน Switch Ryu จะตรวจสอบว่า packet นี้จะต้องส่งต่อไปยัง port ไหนของ Switch เป็นต้น ในการพัฒนาการจัดการเครือข่ายโดยใช้ภาษา Python

Ryu รองรับการทำงานร่วมกับProtocol ต่างๆมากมาย เช่น Openflow, Netconf, OF-config เป็นต้น Ryu Controller สามารถรองรับ Openflow เวอร์ชัน 1.0, 1.1, 1.2, 1.3, 1.4 และ 1.5

2.7.1 Ryu Rest API

Ryu ยังมีการแอพพลิเคชั่นที่รองรับการทำงาน REST API คือ ofctl_rest เพื่อให้สามารถตรวจสอบสถานะของอุปกรณ์ที่เชื่อมต่อกับ Controller และการอัพเดทสถานะต่าง ๆ ช่วยให้การ Debug ทำได้ง่ายขึ้น ตามภาพที่ 2-3 โดยจะรองรับการทำงานร่วมกับ Openflow เวอร์ชัน 1.0, 1.1, 1.2, 1.3 เท่านั้นซึ่งจะกล่าวถึงรายละเอียดของแต่ละพารามิเตอร์และการใช้งานร่วมกับ UniNetExpressLane Monitoring ในบทที่ 3



ภาพที่ 2-3 ตัวอย่างของ Rest API ของ Ryu Controller ในการตรวจสอบสถานะของอุปกรณ์ที่เชื่อมต่ออยู่กับคอนโทรลเลอร์

บทที่ 3
การออกแบบระบบเฝ้าระวัง

ในการพัฒนาระบบสำหรับเฝ้าระวังการให้บริการผ่านเครือข่ายอินเตอร์เน็ตในโครงการ UniNet Express Lane สามารถแบ่งวิธีการดำเนินงานได้เป็น 4 ส่วนดังนี้

1. วิเคราะห์และออกแบบระบบรวม
2. วิเคราะห์และออกแบบระบบเฝ้าระวังคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดต่าง ๆ
3. วิเคราะห์และออกแบบระบบเฝ้าระวังบอร์ด NetFPGA
4. วิเคราะห์และออกแบบระบบเฝ้าระวังคอนโทรลเลอร์

3.1 วิเคราะห์และออกแบบระบบรวม

ระบบสำหรับเฝ้าระวังในโครงการ UniNet Express Lane Serviceนั้นประกอบด้วย 3 ส่วนหลัก ตามภาพที่ 3-1 คือ

3.1.1 เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดต่าง ๆ

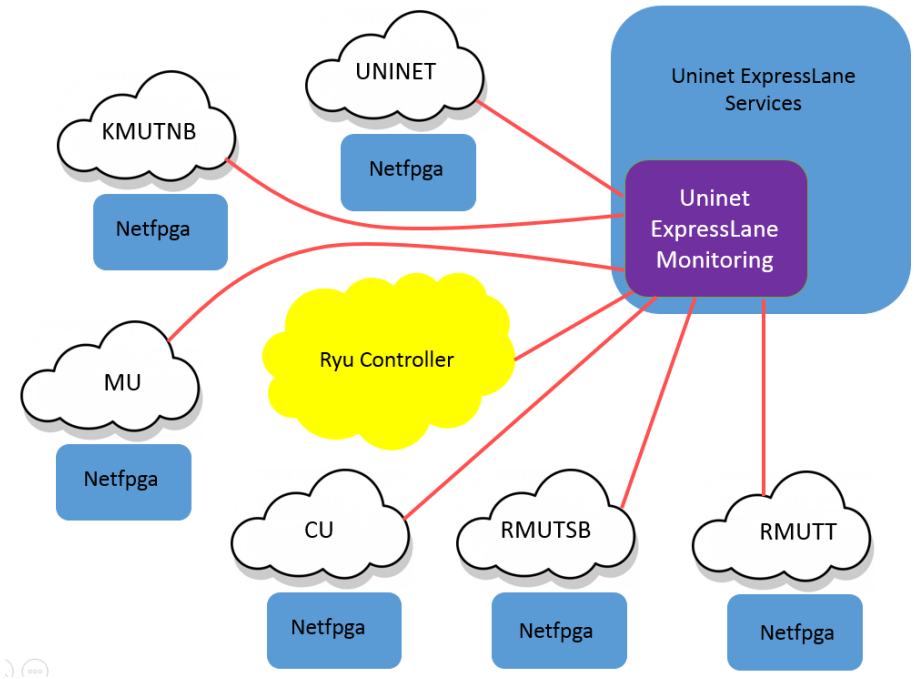
เป็นคอมพิวเตอร์ที่ไว้สำหรับติดตั้งบอร์ด Netfpgaโดยจะต่อเชื่อมกับคอนโทรลเลอร์เพื่อทำหน้าที่เป็นโอเพ่นโฟล์วสวิชต์ และรับส่งข้อมูลผ่านทางพอร์ตทั้ง 4 พอร์ตของบอร์ด Netfpga ตามหลักการของ Software Define Network (SDN) โดยเฝ้าระวังการทำงานของเซอร์วิสต่าง ๆ รวมถึงทรัพยากรต่าง ๆ ในเครื่องคอมพิวเตอร์ เพื่อตรวจสอบว่าสามารถทำงานได้ตามปกติหรือไม่ ซึ่งแต่ละโหนดจะติดตั้งซอฟต์แวร์เพื่อใช้ในการเฝ้าระวังเส้นทางการไหลของข้อมูล และซอฟต์แวร์ในการอัพเดทสถานะไปยังเครื่องคอนโทรลเลอร์ ตามภาพที่ 3-2

3.1.2 บอร์ด NetFPGA

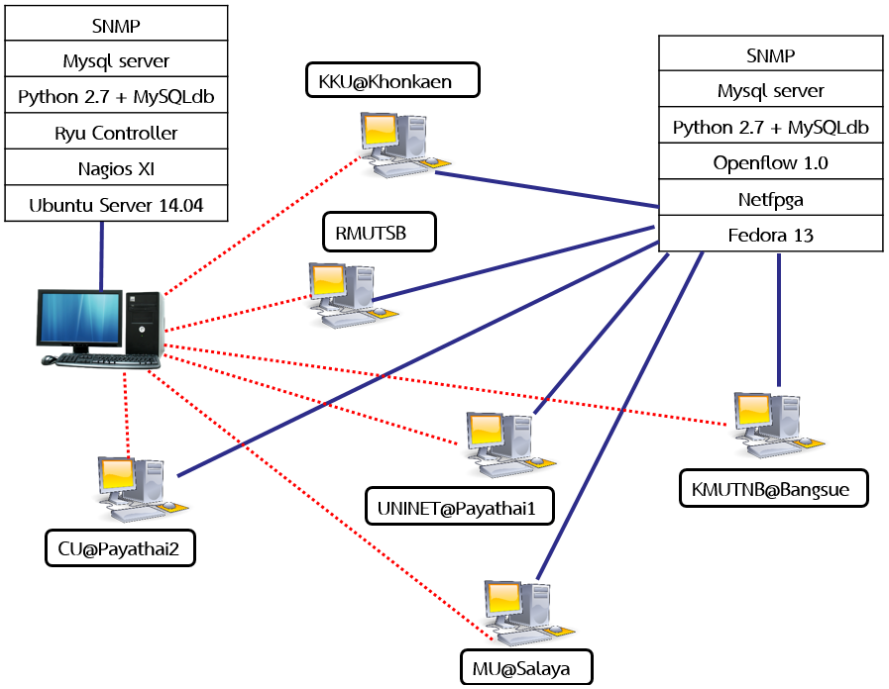
บอร์ดที่ติดตั้งอยู่ในเครื่องคอมพิวเตอร์ที่เป็นโหนดทำหน้าที่เป็นโอเพ่นโฟล์วสวิตช์เพื่อรับส่งข้อมูลจากเครื่องต้นทางไปยังเครื่องปลายทาง โดยการเฝ้าระวัง Mac Address ต้นทางและปลายทาง พอร์ตที่ใช้ในการรับส่งข้อมูล และจำนวนแพ็คเก็ตที่รับส่งระหว่างการส่งข้อมูล

3.1.3 คอนโทรลเลอร์

ซอฟต์แวร์ที่ทำหน้าที่เป็นคอนโทรลเลอร์ในการจัดการเส้นทาง อัพเดทตารางเส้นทางให้กับบอร์ด Netfpga โดยจะเฝ้าระวังจำนวนของโหนดที่เชื่อมต่อกับคอนโทรลเลอร์ ขณะเวลานั้น ๆ รวมทั้ง Mac Address ต้นทางและปลายทาง และจำนวนแพ็คเก็ตที่รับส่งข้อมูลด้วย



ภาพที่ 3-1 ภาพรวมของระบบเฝ้าระวังทางเครือข่าย โดยมีระบบเฝ้าระวังทางเครือข่าย (UninetExpressLane Monitoring) ทำงานอยู่ภายในระบบทางด่วนข้อมูลผ่านทางเครือข่ายของ UniNetอีกทีหนึ่งและทำหน้าที่ในการเฝ้าระวังสถานะของโหนดต่างๆ จำนวน 6 โหนด และคอนโทรลเลอร์



ภาพที่ 3-2 แสดงซอฟต์แวร์และฮาร์ดแวร์ที่ติดตั้งอยู่บนโหนดต่าง ๆ เพื่อใช้ในการตรวจสอบเส้นทางการไหลข้อมูล (DPCTL) ซึ่งเป็นคอมมานไลน์ที่ติดมากับ Openflow 1.0 และอัพเดทไปยังเครื่องคอนโทรลเลอร์ผ่านทาง Mysql server

3.2 วิเคราะห์และออกแบบระบบเฝ้าระวังคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดต่าง ๆ

3.2.1 ข้อกำหนดการเลือกใช้อุปกรณ์

ระบบปฏิบัติการ	:	Fedora 13 with NetFPGA
NetFPGA	:	ความเร็ว 1 GB จำนวน 4 พอร์ต
Mysql Server	:	สำหรับเชื่อมต่อกับ Server ในการอัพเดทข้อมูล
Python 2.7	:	สำหรับเขียนสคริปในการรันเพื่ออัพเดทสถานะ
MySQLdb	:	ไลบรารี่ของ Python สำหรับเชื่อมต่อดาต้าเบส
SNMP	:	โปรโตคอลในการสื่อสารระหว่างโหนดกับ Nagios XI เพื่ออัพเดทสถานะของเซอร์วิส และทรัพยากรในเครื่องโหนด
Nagios XI	:	สำหรับเฝ้าระวังทรัพยากรและเซอวิสต่าง ๆ ของเครื่อง

3.2.2 การเข้าถึงข้อมูลผ่านทาง Nagios XI Backend API

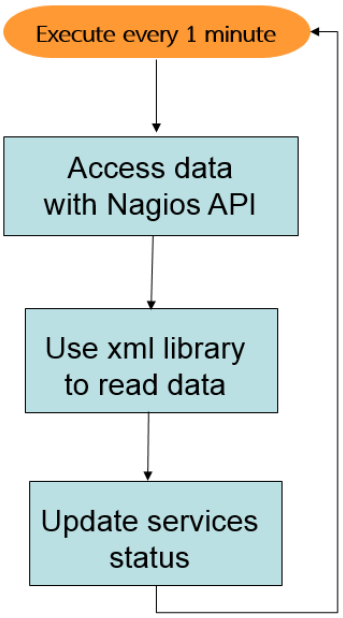
ในการเฝ้าระวังเครื่องโหนดโดยใช้ Nagios XI ผ่านทางโปรโตคอล SNMP ซึ่งจะต้องติดตั้งแพ็คเก็ต SNMP ทั้งสองฝั่งทั้งเครื่อง Server และ Client (โหนด) จากนั้นเลือกเซอร์วิสที่ต้องการจะเฝ้าระวัง เช่น Disk Usage, Memory Usageและใช้ Nagios Backend API ในการเข้าถึงสถานะข้อมูลในรูปของไฟล์ XML โดยสามารถเลือกประเภทของข้อมูลที่ต้องการจะดึงข้อมูลเพื่อนำมาแสดงผลบนหน้าเว็บแอพพลิเคชั่นตามภาพที่ 3-3 ซึ่งในการพัฒนาจะใช้ Current Service Status เพื่อดึงข้อมูลที่เป็นสถานะของเซอร์วิสที่เฝ้าระวังอยู่ขณะนั้นมาใช้งาน

Backend API URLs

You can use the URLs below to fetch information from the Nagios XI backend API. **Note:** It is important to retain the *username* and *ticket*

Data Type	URL
Current Host Status:	http://202.29.154.252/nagiosxi/backend/?cmd=gethoststatus&username=nagiosadmin&tic
Current Service Status:	http://202.29.154.252/nagiosxi/backend/?cmd=getservicestatus&username=nagiosadmin&
Current Program Status:	http://202.29.154.252/nagiosxi/backend/?cmd=getprogramstatus&username=nagiosadmir
Current Program Performance:	http://202.29.154.252/nagiosxi/backend/?cmd=getprogramperformance&username=nagio
System Statistics:	http://202.29.154.252/nagiosxi/backend/?cmd=getsysstat&username=nagiosadmin&ticket=
Log Entries:	http://202.29.154.252/nagiosxi/backend/?cmd=getlogentries&username=nagiosadmin&tick
State History:	http://202.29.154.252/nagiosxi/backend/?cmd=getstatehistory&username=nagiosadmin&ti
Comments:	http://202.29.154.252/nagiosxi/backend/?cmd=getcomments&username=nagiosadmin&tic
Scheduled Downtime:	http://202.29.154.252/nagiosxi/backend/?cmd=getscheduleddowntime&username=nagios
Users:	http://202.29.154.252/nagiosxi/backend/?cmd=getusers&username=nagiosadmin&ticket=r
Contact:	http://202.29.154.252/nagiosxi/backend/?cmd=getcontacts&username=nagiosadmin&ticke
Hosts:	http://202.29.154.252/nagiosxi/backend/?cmd=gethosts&username=nagiosadmin&ticket=r
Services:	http://202.29.154.252/nagiosxi/backend/?cmd=getservices&username=nagiosadmin&ticke
Hostgroups:	http://202.29.154.252/nagiosxi/backend/?cmd=gethostgroups&username=nagiosadmin&ti
Servicegroups:	http://202.29.154.252/nagiosxi/backend/?cmd=getservicegroups&username=nagiosadmin
Contactgroups:	http://202.29.154.252/nagiosxi/backend/?cmd=getcontactgroups&username=nagiosadmin
Hostgroup Members:	http://202.29.154.252/nagiosxi/backend/?cmd=gethostgroupmembers&username=nagios
Servicegroup Members:	http://202.29.154.252/nagiosxi/backend/?cmd=getservicegroupmembers&username=nagi
Contactgroup Members:	http://202.29.154.252/nagiosxi/backend/?cmd=getcontactgroupmembers&username=nagi

ภาพที่ 3-3 Nagios Backend API URLs ที่สามารถนำไปประยุกต์ใช้งานได้ โดยจะมีให้เลือกใช้งานมากมายขึ้นอยู่กับความต้องการใช้งานของผู้นำไปใช้ ซึ่งในปริญญานิพนธ์นี้ได้เลือกใช้ Current Service Status เพื่อตรวจสอบสถานะของเซอร์วิสที่ได้ทำการเฝ้าระวังอยู่



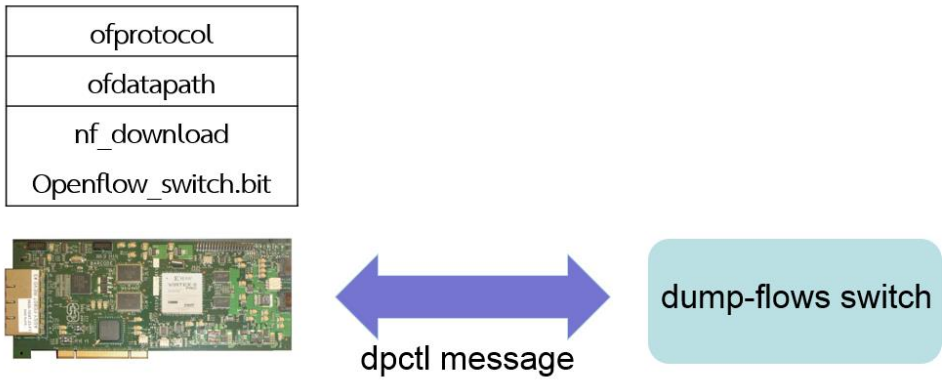
ภาพที่ 3-4 แผนภาพการทำงานเฝ้าระวังคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดต่าง ๆ

จากภาพที่ 3-4 แสดงให้เห็นการทำงานของฟังก์ชันการเฝ้าระวังโหนดต่างๆ ผ่านทางเครื่องเมือ Nagios XI ซึ่งติดต่อสื่อสารผ่านทางโปรโตคอล SNMP โดยการทำงานจะเริ่มด้วยการเข้าถึงข้อมูลที่เฝ้าระวังผ่าน Nagios XI ในรูปของไฟล์ XML และใช้ไลบรารี่ในการอ่านข้อมูลไฟล์ XML ในภาษาไฟธอน ในการเข้าถึงข้อมูลและนำค่าของเซอร์วิสหรือ สถานะที่ต้องการ เพื่อเก็บข้อมูลลงฐานข้อมูล โดยจะทำซ้ำขั้นตอนดังกล่าวทุกๆ 1 นาที ซึ่งกำหนดโดยคำสั่ง Crontab

3.3 วิเคราะห์และออกแบบระบบเฝ้าระวังบอร์ด Netfpga

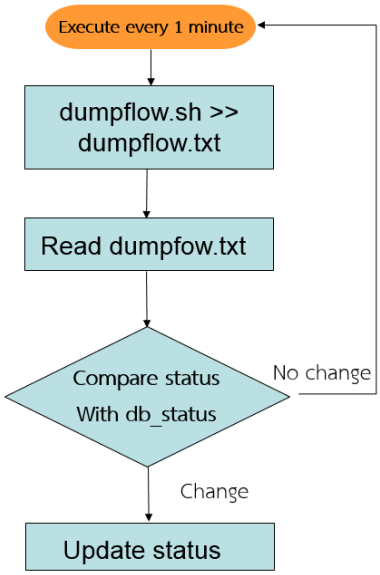
3.3.1 วิเคราะห์และออกแบบระบบ

เมื่อดาวน์โหลดไฟล์ Openflow_switch.bit ลงไปในบอร์ด Netfpga จะทำให้บอร์ด Netfpga ทำหน้าที่เป็นสวิตช์ที่ทำงานตามแนวคิด SDN (Software Define Network) สามารถทำงานร่วมกับคอนโทรลเลอร์ได้ ซึ่งในโอเพ่นโฟล์วนั้นมีคอมมานไลน์ในการตรวจสอบสถานะอุปกรณ์ที่ทำหน้าที่เป็นโอเพ่นโฟล์วสวิตช์ เรียกว่า dpctl command โดยประกอบไปด้วยคำสั่งต่าง ๆ มากมาย เช่น dump-flows เพื่อดูเส้นทางการไหลของข้อมูล จำนวนแพ็คเก็ต add-flows เพื่อเพิ่มเส้นทางการรับส่งข้อมูลและ del-flows ที่สามารถลบเส้นทางได้ผ่านทาง dpctl command (ภาพที่ 3-5)



ภาพที่ 3-5 ภาพรวมของระบบเฝ้าระวังบอร์ด NetFPGA

การทำงานในการเฝ้าระวังบอร์ด NetFPGA เริ่มจากรันไฟล์สคิป dumpflow.sh เพื่อดำเนินการคำสั่งในการดูเส้นทางการไหลของข้อมูล (Flow) ที่ไหลผ่านพอร์ตต่าง ๆ ของบอร์ด NetFPGA และเก็บผลไว้ในไฟล์ dumpflow.txt จากนั้นรันไฟล์สำหรับอ่านค่าจากไฟล์ dumpflow.txt เพื่อนำค่าที่ได้มาเปรียบเทียบกับค่าหรือเส้นทางการไหลของข้อมูล (Flow) ที่อยู่ในฐานข้อมูลซึ่งหากมีการเปลี่ยนแปลงรายละเอียด เช่น Mac Address, พอร์ต, จำนวนแพ็คเก็ต จะเกิดเหตุการณ์อัพเดทข้อมูลในฐานข้อมูล ซึ่งการทำงานทั้งหมดที่กล่าวมาจะถูกทำงานในทุก ๆ นาทีผ่านตารางการทำงานอัตโนมัติ Crontab (ภาพที่ 3-6)



ภาพที่ 3-6 แผนภาพการทำงานเฝ้าระวังบอร์ด NetFPGA

3.4 วิเคราะห์และออกแบบระบบเฝ้าระวังการทำงานของคอนโทรลเลอร์

3.4.1 การเข้าถึงสถานะของโหนดที่เชื่อมต่ออยู่กับคอนโทรลเลอร์ผ่านทาง RestFul API

ในการวิเคราะห์เพื่อเฝ้าระวังการทำงานของโหนดที่เชื่อมต่อกับคอนโทรลเลอร์ใช้หลักการของ RestFul API ซึ่งเป็น API ของ Ryu Controller ในการร้องขอไปยังคอนโทรลเลอร์และตอบกลับมาในรูปของ JSON หรือ XML ซึ่งเป็นไฟล์ขนาดเล็ก โดยที่ฝั่งคอนโทรลเลอร์จะต้องรันแอพพลิเคชั่นที่รองรับการทำงานของ RestFul API คือ ofctl_rest.py ที่อยู่ในโฟล์เดอร์ของ Ryu เพื่อรอรับการร้องขอสถานะต่าง ๆ ผ่านเมธอด เช่น GET, PUSH, PUT และ DEL ซึ่งพารามิเตอร์ในการร้องขอสถานะต่าง ๆ ประกอบไปด้วย

Switches : แสดงสวิตช์ทั้งหมดที่เชื่อมต่ออยู่กับคอนโทรลเลอร์

Method	URL
GET	/stats/switches

Response Message :

Attribute	Description	Example
dpid	Datapath ID	1

Description : แสดงรายละเอียดของสวิตช์ตัวนั้น ๆ ที่เชื่อมต่ออยู่กับคอนโทรลเลอร์

Method	URL
GET	/stats/desc/<dpid>

Response Message :

Attribute	Description	Example
dpid	Datapath ID	“1”
mfr_desc	Manufacturer description	“Stanford University”
hw_desc	Software description	“ Reference Userspace Switch”
sw_desc	Software description	“1.0.0”
serial_num	Serial number	“None”
dp_desc	Human readable description of datapath	“None”

Switch Table : แสดงรายละเอียดของตารางเส้นทางข้อมูลของสวิตช์ตัวนั้น ๆ ที่เชื่อมต่ออยู่กับคอนโทรลเลอร์

Method	URL
GET	/stats/table/<dpid>

Response Message :

Attribute	Description	Example
dpid	Datapath ID	“1”
table_id	Table ID	0
name	Name of Table	“Classifier”
max_entries	Max number of entries supported	1e+06
active_count	Number of active entries	0
lookup_count	Number of packets looked up in table	8

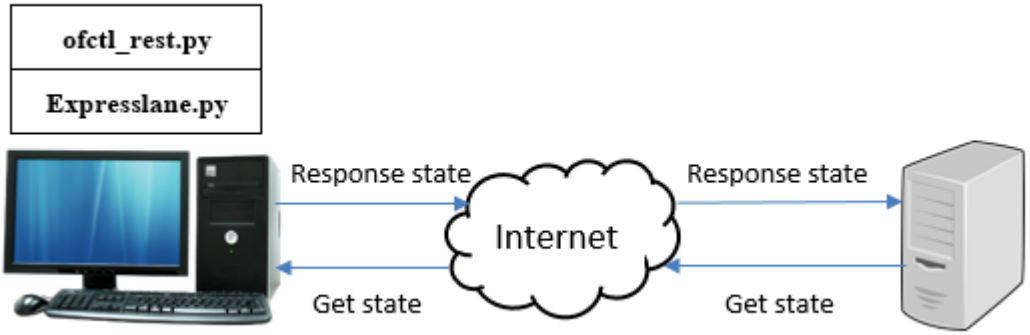
flow stats : แสดงรายละเอียดของเส้นทางไหลข้อมูลของสวิตช์ตัวนั้น ๆ ที่เชื่อมต่ออยู่กับคอนโทรลเลอร์

Method	URL
GET	/stats/flow/<dpid>

Response Message :

Attribute	Description	Example
dpid	Datapath ID	“1”
length	Length of this entry	88
table_id	Software description	0
duration_sec	Software description	2
priority	Priority of the entry	11111

Attribute	Description	Example
packet_count	Number of packets in flow	0
match	Field to match	{“in_port”:1}
actions	Instruction set	[“OUTPUT:2”]



ภาพที่ 3-7 ภาพรวมการทำงานของระบบเฝ้าระวังคอนโทรลเลอร์

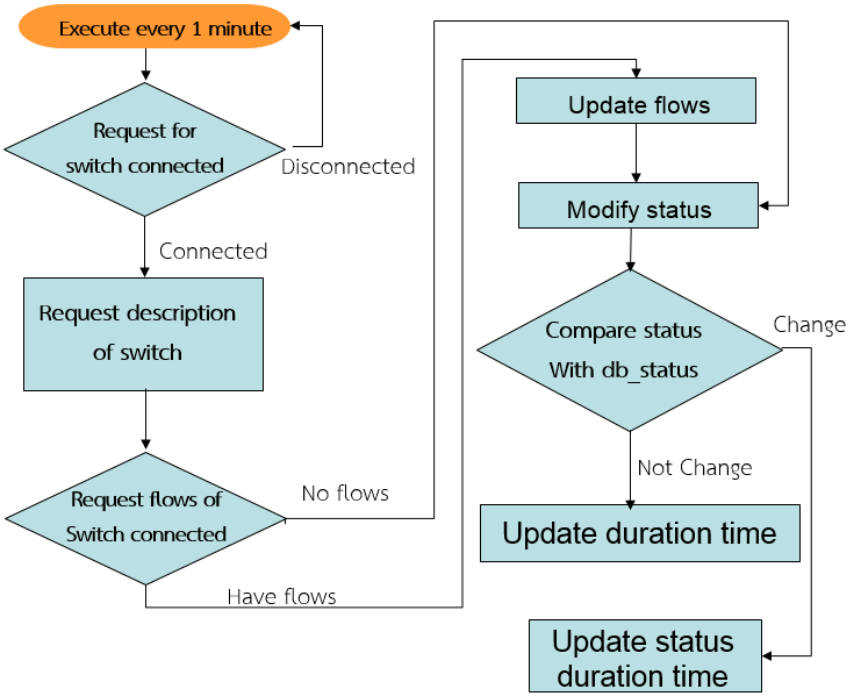
```
$ curl -X GET http://localhost:8080/stats/switches
```

```
[
  1,
  2,
  3
]
```

ภาพที่ 3-8 แสดงการทดสอบร้องขอสถานะของ switch ที่เชื่อมต่ออยู่กับคอนโทรลเลอร์ผ่านทาง Rest API ด้วย method GET จากการทดสอบจะเห็นได้ว่ามี switch ที่เชื่อมต่ออยู่ทั้งหมด 3 โหนด ได้แก่ 1, 2 และ 3 ซึ่งเป็นเลขของ Datapath ID ที่กำหนดให้กับ switch แต่ละตัวตามลำดับ

3.4.2 การดำเนินการในการอัพเดทสถานะผ่านทาง Rest API

การดำเนินการเริ่มต้นด้วยการร้องขอ Rest API เมธอด Get ไปยังคอนโทรลเลอร์เพื่อร้องขอจำนวนโหนดทั้งหมดที่เชื่อมต่ออยู่ขณะนั้นด้วยคำสั่ง http://202.29.154.252:8080/stats/switches ถ้าหากไม่มีโหนดเชื่อมต่ออยู่จะจบการทำงานและไม่มีการอัพเดทข้อมูลในฐานข้อมูล แต่ถ้าหากมีการเชื่อมต่ออยู่อย่างน้อย 1 โหนด ขั้นต่อไปจะเป็นการร้องขอรายละเอียดของสวิตช์ตัวนั้นเพื่อตรวจสอบว่าเป็นสวิชต์จากโหนดใดด้วย http://202.29.154.252:8080/stats/desc/1 จากนั้นจะร้องขอเส้นทางการไหลของข้อมูลในสวิตช์ตัวนั้น ๆ เพื่อดูรายละเอียดการรับส่งด้วย http://202.29.154.252:8080/stats/flow/1 ขึ้นอยู่กับขณะนั้นว่ามีการรับส่งข้อมูลเกิดขึ้นหรือไม่ และมีการคำนวณระยะเวลาที่โหนดเชื่อมต่อหรือไม่เชื่อมต่อด้วย จากนั้นนำข้อมูลทั้งหมดไปอัพเดทในฐานข้อมูล (ภาพที่ 3-7) ซึ่งการทำงานทั้งหมดที่กล่าวมาจะถูกทำงานในทุก ๆ นาทีผ่านตารางการทำงานอัตโนมัติ Crontab (ภาพที่ 3-8)



ภาพที่ 3-9 แผนภาพการทำงานของระบบเฝ้าระวังคอนโทรลเลอร์

บทที่ 4
ผลการนำระบบเฝ้าระวังมาใช้กับระบบทางด่วนข้อมูล

การทดสอบประสิทธิภาพของระบบเฝ้าระวังทางเครือข่าย มีเป้าหมายเพื่อทดสอบว่าระบบสามารถทำงานร่วมกับระบบทางด่วนข้อมูลได้จริง สามารถรายงานสถานะของโหนดต่าง ๆ รวมถึงเส้นทางการไหลข้อมูลได้อย่างถูกต้อง ซึ่งผลการดำเนินการสามารถแบ่งออกเป็น 3 ส่วนหลัก คือ

1. การเฝ้าระวังคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดต่าง ๆ
2. การเฝ้าระวังสถานะของบอร์ด Netfpga
3. การเฝ้าระวังสถานะของคอนโทรลเลอร์ และการทดสอบประสิทธิภาพการทำงานของระบบเฝ้าระวังที่ติดตั้งอยู่บนเครือข่ายของ UniNet จำนวน 6 โหนด

4.1 การเฝ้าระวังคอมพิวเตอร์ที่ทำหน้าที่เป็นโหนดต่าง ๆ

ใช้ไลบราลี่ของไฟธอนเพื่อเข้าถึงข้อมูลสถานะของเซอร์วิสที่อยู่ในรูปแบบไฟล์ XML ผ่านทาง REST API (ภาพที่ 4-1) จะได้ข้อมูลไฟล์ XML (ภาพที่ 4-2) จากนั้นให้เลือกรายละเอียด (Parameter) เฉพาะส่วนที่ต้องการจะนำมาแสดงผลบนหน้าเว็บ โดยในปริญญานิพนธ์นี้ได้เลือกพารามิเตอร์ ตามภาพที่ 4-1 ดังนี้

host_name : ชื่อโหนดต่าง ๆ ได้แก่ BSU, PYT1, PYT2, SLY, RMUTT, RMUTSB
name : ชื่อเซอร์วิสที่เฝ้าระวัง เช่น PING, CPU_Usage, Memory_Usage
status : สถานะของแต่ละเซอร์วิส
status_update_time : เวลาในการอัพเดทสถานะของเซอร์วิส

และนำไปเปรียบเทียบกับข้อมูลในฐานข้อมูลเพื่อจะได้ทราบว่ามีการเปลี่ยนแปลงสถานะหรือไม

```xml
<?xml version="1.0" encoding="utf-8"?>
<servicestatuslist>
  <recordcount>28</recordcount>
  <servicestatus id='336'>
    <instance_id>1</instance_id>
    <service_id>175</service_id>
    <host_id>171</host_id>
    <host_name>UNINET</host_name>
    <host_alias>UNINET</host_alias>
    <name>Swap Usage</name>
    <host_display_name></host_display_name>
    <host_address>202.29.226.166</host_address>
    <display_name>Swap Usage</display_name>
    <status_update_time>2016-06-07 18:58:09</status_update_time>
```

ภาพที่ 4-1 ตัวอย่างข้อมูลสถานะเซอร์วิสที่ได้จากการเข้าถึงผ่านทาง Backend API ซึ่งจะอยู่ในรูปแบบของไฟล์ XML ซึ่งประกอบด้วยพารามิเตอร์ต่าง ๆ เช่น host_name, name, status_update_time เป็นต้น

```python
response = requests.get("http://202.29.154.252/nagiosxi/backend/
            ?cmd=getservicestatus&username=nagiosadmin&ticket=rgkk3bo4")
tree = ET.fromstring(response.content)
el = tree.findall("servicestatus")
index = [0,1,2,3,4,5,6,7,10,11,12,13,14,17,18,19,20,23,
        27,28,30,32,33,34,36,35,37,38,39,41,42,43,44]
db = MySQLdb.connect("localhost","root","root","UniNetExpressLane")
cur = db.cursor()

for i in range(len(index)):
    ch        = el[index[i]].getchildren()
    host_name = ch[3].text
    service   = ch[5].text
    timestamp = ch[9].text
    statuss   = ch[10].text
```

ภาพที่ 4-2 โค้ดการเข้าถึงค่าสถานะของเซอร์วิสต่าง ๆ โดยใช้ไลบรารี่ภาษาไพธอนดึงค่าจาก Nagios Backend API ตาม Index ของแต่ละเซอร์วิส

โดยการทำงานดังกล่าวจะถูกกำหนดไว้ในตารางการทำงานอัตโนมัติ (Crontab) และเริ่มทำงานทุก ๆ นาที (ภาพที่ 4-3) จากภาพจะเป็นการกำหนดให้มีการรันไฟล์ update_nagios_statuss.py เพื่อรันสคริปสำหรับอ่านค่าจาก Nagios Backend API โดยใช้ไลบรารี่สำหรับเข้าถึงข้อมูลไฟล์ XML จากนั้นเก็บค่าของสถานะเซอร์วิสที่ต้องการและนำข้อมูลเหล่านั้นมาเปรียบเทียบกับค่าในฐานข้อมูลและอัพเดทค่าพร้อมทั้งแสดงผลบนหน้าเว็บแอพพลิเคชั่น (ภาพที่ 4-4)

```
* * * * * sudo python /usr/lib/python2.7/xml/etree/update_nagios_statuss.py
> /usr/lib/python2.7/xml/etree/error.txt 2>&1
```

ภาพที่ 4-3 ภาพแสดงการกำหนดตารางการทำงานอัตโนมัติในโหนดต่าง ๆ

Nagios Status

Hostname	Timestamp	Service	Status
BSE	2016-06-13 10:28:31	Swap Usage	Swap space: 0%used(0MB/4000MB) (<80%) : OK
BSE	2016-06-13 10:24:23	Ping	OK - 202.29.226.170: rta 0.179ms, lost 0%
BSE	2016-06-06 16:43:48	Memory Usage	Physical memory: 59%used(1196MB/2016MB) (<80%) : OK
BSE	2016-06-13 10:28:31	CPU Usage	2 CPU, average load 1.0% < 80% : OK

ภาพที่ 4-4 ตัวอย่างหน้าแสดงผลสถานะของโหนดต่างๆบนหน้าเว็บแอพพลิเคชั่น

4.2 การเฝ้าระวังบอร์ด NetFPGA

4.2.1 ทดสอบคำสั่ง dpctl command โดยใช้การ dump-flows เพื่อดูข้อมูลเส้นทางการไหลของข้อมูลผ่านพอร์ตของบอร์ด Netfpga หากไม่มีการรับส่งข้อมูลผลลัพธ์จะว่าง คือไม่มีรายละเอียดของเส้นทาง (ภาพที่ 4-5) แต่ถ้ามีการรับส่งข้อมูลจะแสดงรายละเอียดของเส้นทางการไหล (ภาพที่ 4-6) เช่น Mac Address, In port, Out port, Packet count เป็นต้น

```
[root@Project-SDN ~]# dpctl dump-flows unix:/var/run/dp0
stats_reply (xid=0x99483f0e): flags=none type=1(flow)
[root@Project-SDN ~]#
```

ภาพที่ 4-5 ตัวอย่างใช้คำสั่ง dump-flows ในขณะที่ไม่มีการส่งข้อมูลผ่านบอร์ด Netfpga

```
[root@Project-SDN ~]# dpctl dump-flows unix:/var/run/dp0
stats_reply (xid=0xbfa74873): flags=none type=1(flow)
  cookie=0, duration_sec=167s, duration_nsec=282000000s, table_id=0, priority=32
768, n_packets=334, n_bytes=32732, idle_timeout=0,hard_timeout=300,in_port=3,dl_
src=00:23:ae:9c:b3:c8,dl_dst=00:23:ae:9c:92:fd,actions=output:2
  cookie=0, duration_sec=167s, duration_nsec=294000000s, table_id=0, priority=32
768, n_packets=334, n_bytes=32732, idle_timeout=0,hard_timeout=300,in_port=2,dl_
src=00:23:ae:9c:92:fd,dl_dst=00:23:ae:9c:b3:c8,actions=output:3
[root@Project-SDN ~]#
```

ภาพที่ 4-6 ตัวอย่างใช้คำสั่ง dump-flows ในขณะที่มีการส่งข้อมูลผ่านบอร์ด Netfpga

4.2.2 เปรียบเทียบค่าและอัพเดทข้อมูลในฐานข้อมูล

การทำงานของไฟล์ updateflow.py เป็นการอ่านค่าจากไฟล์ dumpflow.txt ซึ่งเป็นเส้นทางข้อมูลขณะนั้น ๆ ที่มีการอัพเดทในทุก ๆ นาที จากนั้นเลือกเอาเฉพาะค่าที่ต้องการจะนำมาแสดงผล เช่น Mac address, n_packet, inport, outport เป็นต้น และนำมาเปรียบเทียบกับค่าเดิมในฐานข้อมูลเพื่ออัพเดทค่าใหม่ หากมีการเปลี่ยนแปลง และแสดงผลผ่านหน้าเว็บแอพพลิเคชั่น (ภาพที่ 4-7) โดยการทำงานดังกล่าวจะถูกกำหนดไว้ในตารางการทำงานอัตโนมัติ (Crontab) และเริ่มทำงานทุก ๆ นาที

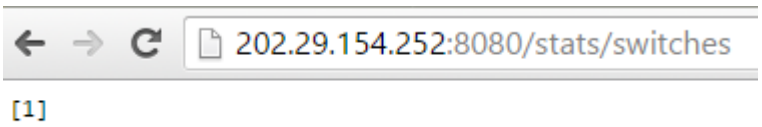
Netfpga Status

Node	Timestamp	Inport	Dest_MAC	Output	Dest_MAC	packet_count
BSE	2016-06-13 10:30:01	-	-	-	-	-
KKN	2016-06-15 16:17:01	3	00:23:ae:9c:b3:c8	2	00:23:ae:9c:92:fd	95
PYT1	2016-06-15 16:25:01	-	-	-	-	-
PYT2	2016-06-13 11:44:01	-	-	-	-	-
RMUTSB	2016-06-13 11:45:01	-	-	-	-	-
SLY	2016-06-15 16:09:01	-	-	-	-	-

ภาพที่ 4-7 ตัวอย่างหน้าแสดงผลเส้นทางไหลข้อมูลต่างๆบนหน้าเว็บแอพพลิเคชั่น

4.3 การเฝ้าระวังคอนโทรลเลอร์

4.3.1 ทดสอบใช้งาน REST API ร้องขอสถานะของโหนดที่เชื่อมต่อกับคอนโทรลเลอร์



ภาพที่ 4-8 ผลตัวอย่างการร้องขอสถานะ การเชื่อมต่อของโหนดกับคอนโทรลเลอร์

จากภาพที่ 4-8 จะเห็นว่าขณะที่ร้องขอสถานะ มีการเชื่อมต่อของโหนดกับคอนโทรลเลอร์เพียง 1 โหนดเท่านั้น โดยเลขที่แสดงจะเป็น Datapath ID ของสวิตช์ตัวนั้น ๆ ที่ถูกกำหนดในขั้นตอนการกำหนด Datapath เช่น 00000000001

4.3.2 เปรียบเทียบและอัพเดทสถานะ

เริ่มต้นจากการร้องขอสถานะของโหนดที่เชื่อมต่ออยู่กับคอนโทรลเลอร์ในขณะนั้นด้วย REST API ซึ่งผลลัพธ์ที่ตอบกลับมาจะอยู่ในรูปของ JSON จึงต้องใช้ไลบรารี่ JSON ในภาษาไพธอน เพื่ออ่านข้อมูลถ้าหากมีข้อมูลอย่างอย่างน้อย 1 โหนดเชื่อมต่ออยู่ขั้นตอนต่อไปจะเป็นการร้องขอรายละเอียดและเส้นทางการไหลของข้อมูลของโหนดนั้น ๆ (ภาพที่ 4-9) และคำนวณช่วงระยะเวลาของแต่ละสถานะ เพื่อนำมาอัพเดทในฐานข้อมูล และแสดงผลผ่านหน้าเว็บแอพพลิเคชั่นต่อไป (ภาพที่ 4-11) ซึ่งถ้าหากสถานะของโหนดเป็น Online จะสามารถกดเพื่อดูรายละเอียดเส้นทางการไหลของข้อมูลได้ (ภาพที่ 4-12)

```
url_sw = "http://202.29.154.252:8080/stats/switches"
response = urllib.urlopen(url_sw)
sw_data = json.loads(response.read())
desc_sw = "http://202.29.154.252:8080/stats/desc/"+str(num)
flow_sw = "http://202.29.154.252:8080/stats/flow/"+str(num)
```

ภาพที่ 4-9 โค้ดส่วนการร้องข้อสถานะต่างๆผ่าน REST API

โดยการทำงานดังกล่าวจะถูกกำหนดไว้ในตารางการทำงานอัตโนมัติ (Crontab) และเริ่มทำงานทุก ๆ นาที (ภาพที่ 4-10

```
* * * * * sudo python /home/expresswayserver/Expressway/update_ryu_status.py
 > /home/expresswayserver/Expressway/log.txt 2>&1
```

ภาพที่ 4-10 ภาพแสดงการกำหนดตารางการทำงานอัตโนมัติบนเครื่องคอนโทรลเลอร์

Node_Connected

ID	Node Name	Status	Update	Duration
1	PYT1	Offline	Sat Jun 11 2016 23:58:01 GMT+0700 (ICT)	3 days, 16:42:01.01
2	BSE	Offline	Mon Jun 13 2016 09:55:02 GMT+0700 (ICT)	2 days, 6:45:00.01
3	SLY	Offline	Wed Jun 08 2016 10:57:14 GMT+0700 (ICT)	7 days, 5:42:48.01
4	PYT2	Offline	Wed Jun 15 2016 09:34:01 GMT+0700 (ICT)	7:05:00.61
7	RMUTSB	Offline	Mon Jun 13 2016 12:02:02 GMT+0700 (ICT)	2 days, 4:36:59.61
8	KKN	Online	Mon Jun 13 2016 12:04:01 GMT+0700 (ICT)	2 days, 4:35:00.61

ภาพที่ 4-11 ภาพแสดงสถานะของโหนดต่าง ๆ ที่เชื่อมต่อยู่กับคอนโทรลเลอร์พร้อมทั้งเวลาที่เริ่มมีการเปลี่ยนแปลงสถานะและช่วงเวลาของสถานะ

Detail

Src Mac	Dest Mac	Inport	Outport	Packet Count
00:23:ae:9c:92:fd	00:23:ae:9c:b3:c8	2	3	394

Cancel

ภาพที่ 4-12 ภาพแสดงรายละเอียดของเส้นทางการไหลของข้อมูลที่ได้จากการร้องขอผ่าน REST API

4.4 การทดสอบประสิทธิภาพของระบบ

ในการทดสอบประสิทธิภาพของระบบเฝ้าระวังมีเป้าหมายการทดสอบการทำงานของระบบเฝ้าระวังที่ติดตั้งอยู่บนเครือข่ายของ UniNet จำนวน 6 โหนด ได้แก่ BSE, PYT1,PYT2, SLY, RMUTSB, RMUTT ว่าสามารถใช้งานได้จริง และมีความเร็วในการตอบสนองที่ดีโดยในการทดสอบระบบเฝ้าระวังเครือข่ายในการให้บริการทางด่วนข้อมูลนั้น จะแบ่งการเฝ้าระวังเป็น 3 ส่วนหลัก คือ

1. เฝ้าระวังคอนโทรลเลอร์
2. เฝ้าระวังโหนดต่าง ๆ
3. เฝ้าระวังNetfpga (Openflow Switch)

เมื่อระบบทำงานการเฝ้าระวังในส่วนต่าง ๆ จะทำการเก็บข้อมูลและส่งค่าไปยัง Server ที่อยู่ในเครือข่ายจริง (Public IP) เพื่อเก็บค่าสถานะลงในฐานข้อมูลในทุก ๆ 1 นาที และส่วนแอพพลิเคชั่นจะดึงข้อมูลจากฐานข้อมูลเพื่อนำไปแสดงผลบนหน้าเว็บได้อย่างถูกต้อง หรือไม่ถูกต้อง หรือไม่ตอบสนองเลย

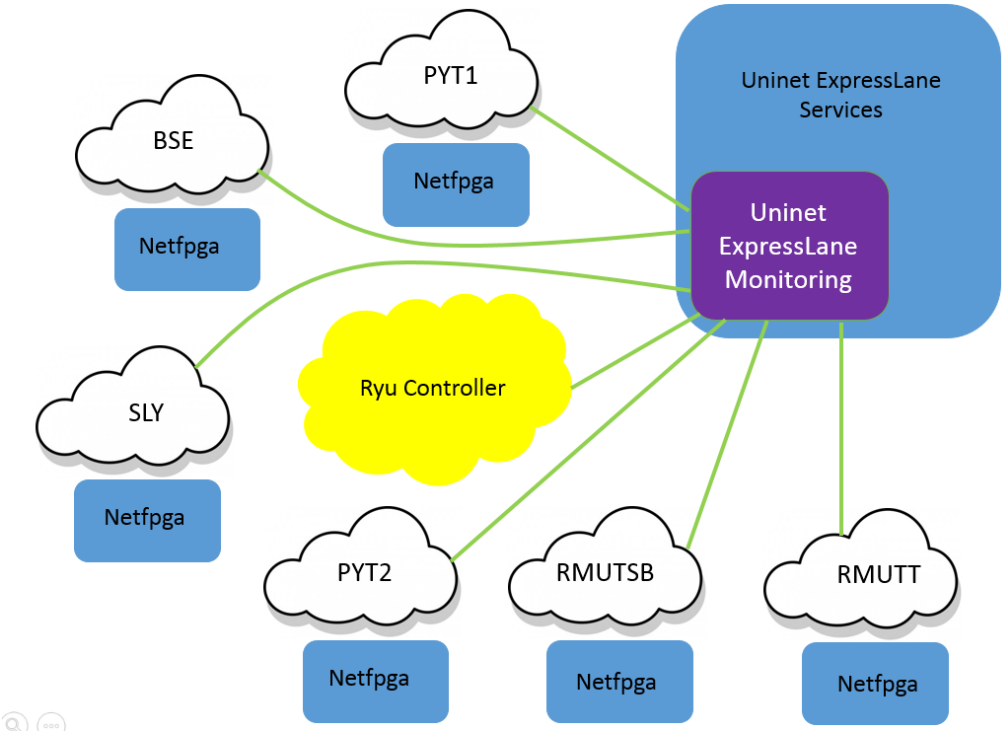
ในการเลือกตัววัดประสิทธิภาพของระบบเฝ้าระวังการให้บริการทางด่วนข้อมูลนี้จะสังเกตที่ความเร็วในการอัพเดทสถานะว่าอัพเดทได้ตามระยะเวลาที่กำหนดหรือไม่และสังเกตความถูกต้องในการแสดงผลว่ามีความผิดพลาดหรือไม่พารามิเตอร์ต่าง ๆ ในการทดสอบประสิทธิภาพของระบบเฝ้าระวังในด้านของระบบ ได้แก่ จำนวนคอนโทรลเลอร์ จำนวนโหนด ในด้านของภาระงานนั้นผู้จัดทำต้องการจะทดสอบว่าภายหลังจากระบบเฝ้าระวังเริ่มทำงานแล้ว การเก็บค่าของสถานะต่าง ๆ รวมถึงการแสดงผลผ่านทางหน้าเว็บถูกต้องและเป็นอัตโนมัติหรือไม่ และในกรณีที่มีโหนดใดโหนดหนึ่งล้มเหลว จะมีการอัพเดทสถานะโดยอัตโนมัติและถูกต้องหรือส่งผลกระทบต่อระบบเฝ้าระวังหรือไม่

การเลือกแฟคเตอร์ในการวัดประสิทธิภาพ ผู้จัดทำได้เลือกพารามิเตอร์ จำนวนโหนดในการเชื่อมต่อกับคอนโทรลเลอร์ (Online) คือ เมื่อมีบางโหนดล้มเหลวจะมีการอัพเดทสถานะและแสดงผลได้อย่างถูกต้อง

เทคนิคที่เลือกใช้เป็นตัววัดจากระบบจริงที่สร้างขึ้นเอง โดยได้ทำการติดตั้งโหนดจำนวน 6 โหนดไว้ตามสถานที่ต่าง ๆ และกำหนด Public IP ไว้สำหรับรีโมทเข้าไป จากนั้นเชื่อมต่อเข้าไปยัง

คอนโทรลเลอร์ ตามที่ได้ออกแบบไว้โดยผลการทดลองในกรณีต่าง ๆ จะถูกแสดงแยกตามกรณีดังนี้

กรณี 1 ทุกโหนดเชื่อมต่อกับระบบเฝ้าระวัง (เหตุการณ์ปกติ)



ภาพที่ 4-13 แสดงโครงสร้างการเชื่อมต่อของทุกโหนด เพื่อทดสอบการทำงานของระบบเฝ้าระวัง

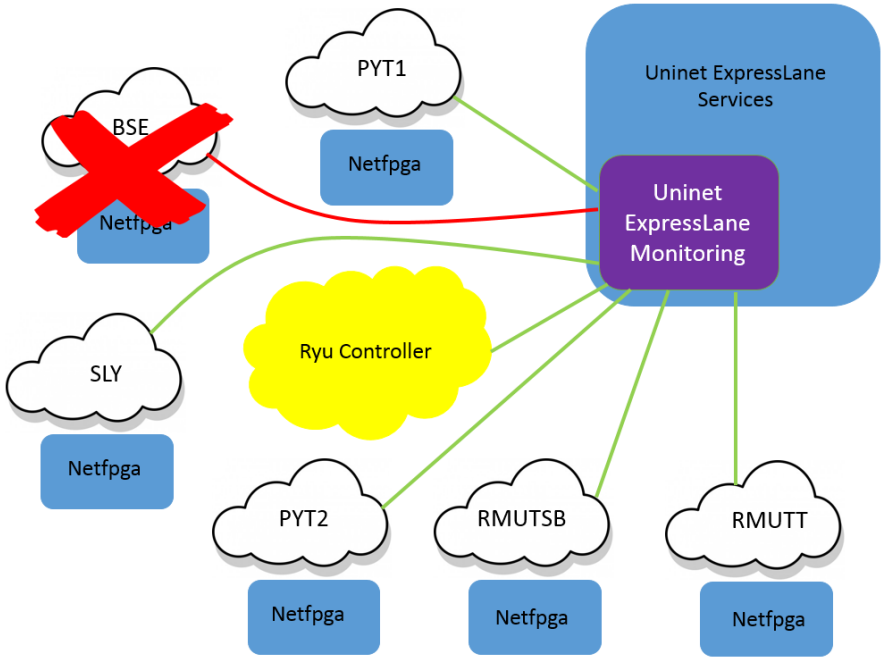
Node_Connected				
ID	Node Name	Status	Update	Duration
1	PYT1	Online	2016/09/07 15:47:01	1:15:00.43
2	BSE	Online	2016/09/07 17:02:01	0 0:0:0
3	SLY	Online	2016/09/07 15:49:01	1:13:00.43
4	PYT2	Online	2016/09/07 15:51:01	1:11:00.43
5	RMUTSB	Online	2016/09/07 15:54:02	1:07:59.43
6	RMUTT	Online	2016/09/07 15:55:02	1:06:59.43

ภาพที่ 4-14 แสดงสถานะของโหนดที่เชื่อมต่อกับคอนโทรลเลอร์จำนวน 6 โหนด

Nagios Status			
Hostname	Timestamp	Service	Status
BSE	2016-09-07 16:49:52	Swap Usage	Swap space: 0%used(0MB/4032MB) (<80%) : OK
BSE	2016-09-07 16:49:59	Ping	OK - 202.29.226.170: rta 0.213ms, lost 0%
BSE	2016-09-07 16:49:59	Memory Usage	Physical memory: 64%used(1279MB/2003MB) (<80%) : OK
BSE	2016-09-07 16:49:59	CPU Usage	2 CPU, average load 1.0% < 80% : OK
Controller	2016-08-23 19:59:52	SSH	SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.6 (protocol 2.0)
Controller	2016-08-23 19:58:13	HTTP	HTTP OK: HTTP/1.1 200 OK - 3239 bytes in 0.001 second response time
Controller	2016-09-07 16:46:13	Current Users	USERS OK - 3 users currently logged in
Controller	2016-09-07 16:49:22	Service Status - mysqld	(No output on stdout) stderr:
PYT1	2016-09-07 16:49:59	Ping	OK - 202.29.226.166: rta 0.676ms, lost 0%
PYT1	2016-09-07 16:49:59	Memory Usage	Physical memory: 51%used(1034MB/2016MB) (<80%) : OK

ภาพที่ 4-15 แสดงสถานะของเซอร์วิสที่ได้จากการเฝ้าระวังโหนด โดยใช้ Nagios XI และแสดงผลผ่านหน้าเว็บ จะเห็นได้ว่าทุกโหนดที่เฝ้าระวังจะอยู่ในสภาวะปกติ

กรณี 2 มีบางโหนดล้มเหลว เช่น โหนดBSU ล้มเหลว (เหตุการณ์ผิดปกติ)



ภาพที่ 4-16 แสดงโครงสร้างการเชื่อมต่อของโหนดต่าง ๆ ยกเว้นโหนด BSE เพื่อทดสอบการทำงานของระบบเฝ้าระวัง

Node_Connected

ID	Node Name	Status	Update	Duration
1	PYT1	Online	2016/09/07 15:47:01	0:09:00.97
2	BSE	Offline	2016/08/25 17:26:01	12 days, 22:30:00.97
3	SLY	Online	2016/09/07 15:49:01	0:07:00.97
4	PYT2	Online	2016/09/07 15:51:01	0:05:00.97
5	RMUTSB	Online	2016/09/07 15:54:02	0:01:59.97
6	RMUTT	Online	2016/09/07 15:55:02	0:00:59.97

ภาพที่ 4-17 แสดงสถานะของโหนดที่เชื่อมต่อกับคอนโทรลเลอร์จำนวน 5 โหนด ล้มเหลว 1 โหนด ได้แก่ โหนด BSE

Nagios Status

Hostname	Timestamp	Service	Status
BSE	2016-09-07 16:27:58	Swap Usage	ERROR: Description/Type table : No response from remote host "202.29.226.170".
BSE	2016-09-07 16:27:56	Ping	OK - 202.29.226.170: rta 0.212ms, lost 0%
BSE	2016-09-07 16:27:58	Memory Usage	ERROR: Description/Type table : No response from remote host "202.29.226.170".
BSE	2016-09-07 16:27:58	CPU Usage	ERROR: Description table : No response from remote host "202.29.226.170".
Controller	2016-08-23 19:59:52	SSH	SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.6 (protocol 2.0)
Controller	2016-08-23 19:58:13	HTTP	HTTP OK: HTTP/1.1 200 OK - 3239 bytes in 0.001 second response time
Controller	2016-09-07 16:26:22	Current Users	USERS OK - 3 users currently logged in
Controller	2016-09-07 16:24:31	Service Status - mysqld	(No output on stdout) stderr:
PYT1	2016-09-07 16:27:58	Ping	OK - 202.29.226.166: rta 0.675ms, lost 0%
PYT1	2016-09-07 16:27:58	Memory Usage	Physical memory: 51%used(1033MB/2016MB) (<80%) : OK

ภาพที่ 4-18 แสดงสถานะของเซอร์วิสที่ได้จากการเฝ้าระวังโหนดกรณีโหนด BSE ล้มเหลวโดยใช้ Nagios XI และแสดงผลผ่านหน้าเว็บ จะเห็นได้ว่าทุกโหนดที่เฝ้าระวังจะอยู่ในสภาวะปกติ ยกเว้นโหนด BSE เนื่องจากโหนดล้มเหลว

จากการทดสอบประสิทธิภาพของระบบเฝ้าระวังตามกรณีต่าง ๆ ทั้งแบบกรณีปกติและกรณีมีบางโหนดล้มเหลว พบว่าระบบเฝ้าระวังสามารถทำงานได้อย่างถูกต้อง การประมวลผลและแสดงผลได้ตรงตามเวลาที่กำหนดไว้ และมีประสิทธิภาพ

บทที่ 5
สรุปผลการนำระบบเฝ้าระวังมาใช้งานร่วมกับการให้บริการทางด่วนข้อมูล

ในปริญญานิพนธ์นี้ได้นำเสนอการพัฒนาระบบเฝ้าระวังทางเครือข่ายประยุกต์ใช้งานร่วมกับระบบให้บริการทางด่วนข้อมูล (UniNet Express Lane Service) ใช้ชื่อว่า UniNet Express Lane Monitoring ซึ่งเป็นความร่วมมือระหว่างสาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือกับสำนักงานบริหารเทคโนโลยี เพื่อให้บริการเส้นทางอินเตอร์เน็ตทางด่วนผ่านทางหน้าเว็บแอพพลิเคชั่นซึ่งระบบเฝ้าระวังทางเครือข่ายมีหน้าที่สำคัญในการตรวจสอบสถานะของโฟล์ว และสถานะของโหนดที่ติดตั้งอยู่ตามมหาวิทยาลัยต่าง ๆ จำนวน 6 แห่งที่ต่อเชื่อมถึงกันผ่านทางเครือข่ายของ UniNet ประกอบด้วย การเฝ้าระวังสถานะโหนดที่เชื่อมต่ออยู่กับคอนโทรลเลอร์, การเฝ้าระวังโฟล์วของบอร์ด NetFPGA และการเฝ้าระวังเซอร์วิสของเครื่องคอมพิวเตอร์ที่ติดตั้งอยู่ตามโหนดต่าง ๆ และนำมาแสดงผลบนหน้าเว็บแอพพลิเคชั่นผ่านทาง Nodejs จากการทดลองแสดงให้เห็นว่าระบบเฝ้าระวังทางเครือข่ายสามารถทำงานร่วมกับระบบให้บริการทางด่วนข้อมูลได้อย่างดี สามารถอัพเดทสถานะได้ถูกต้องและตรงตามเวลาที่กำหนด รวมทั้งสามารถเก็บ log ได้เมื่อมีการเปลี่ยนแปลงสถานะของเซอร์วิสเพื่อไว้สำหรับให้ผู้ดูแลระบบได้สามารถนำไปวิเคราะห์ปรับปรุงระบบต่อไป ซึ่งจะทำให้เกิดประโยชน์แก่กลุ่มคน 2 กลุ่ม คือ ผู้ใช้งานบริการทางด่วนข้อมูลและผู้ดูแลระบบ โดยผู้ใช้งานสามารถตรวจสอบสถานะความพร้อมใช้งานของโหนดต่าง ๆ ได้ก่อนจะจองเส้นทาง อีกทั้งยังช่วยให้ผู้ดูแลระบบสามารถใช้ประโยชน์จากระบบเฝ้าระวังในการตรวจสอบระบบย้อนหลังในกรณีที่เกิดเหตุการณ์ระบบล้มเหลว หรือนำไปใช้เป็นข้อมูลในการวิเคราะห์เพื่อปรับปรุงให้ระบบมีความเสถียรเพิ่มมากยิ่งขึ้นได้

ตลอดการพัฒนาระบบเฝ้าระวังทางเครือข่ายผู้พัฒนาได้พบปัญหาและอุปสรรคต่างๆ จึงได้ทำการวบรวมไว้ดังนี้ เครื่องมือที่ใช้ในการเฝ้าระวังเป็นซอฟแวร์ที่มีลิขสิทธิ์ จึงมีข้อจำกัดในการใช้งาน การติดตั้ง Nagios XI จำเป็นจะต้องติดตั้งในระบบปฏิบัติการที่ยังไม่มีการติดตั้งแพ็คเก็ตเพิ่มเติม หรือควรจะติดตั้งหลังจากที่ลงระบบปฏิบัติการเสร็จสิ้น มิฉะนั้นอาจทำให้ระบบติดตั้งไม่สำเร็จ ข้อจำกัดของเซอร์วิส Crontab ซึ่งเป็นคำสั่งในการกำหนดเวลาการทำงานของงานต่างๆ สามารถกำหนดการทำงานขั้นต่ำได้เพียง 1 นาทีเท่านั้น ทำให้การอัพเดทข้อมูลมีความล่าช้าเล็กน้อย รวมทั้งเมื่อมีการเพิ่มหรือลดจำนวนโหนดจะต้องทำการติดตั้งแพ็คเก็ตสำหรับการเฝ้าระวังเอง ซึ่งหากมีการพัฒนาต่อไปอาจจัดทำให้ระบบสามารถเรียนรู้เองอัตโนมัติ และการแสดงผลของโฟล์วที่

ไหลผ่านบอร์ด NetFPGA หรือ Openflow Switch ให้อยู่ในรูปแบบเมทริกซ์มีความระเอียดมากยิ่งขึ้น

เอกสารอ้างอิง

1. *C.C. Li,Z.S. Ji,F. Wang, P.Wang,Y. Wang,Z.C. Zhang. " The Network Monitoring Base On Cacti for EAST"* [Online]. Available: http://ieeexplore.ieee.org/document/7543086/
2. Quan Vuong, Ha Manh Tran, Son Thanh Le. *"Distributed Event Monitoring for Software Defined Network"* [Online]. Available: http://ieeexplore.ieee.org/document/7422379/
3. Nagios [Online]. Available: https://www.nagios.com
4. *Install and Configure SNMP On Centos* [Online]. Available: https://www.liquidweb.com/kb/how-to-install-and-configure-snmp-on-centos/
5. Ctrontab [Online]. Available: https://support.hostatom.com/knowledgebase.php?action=displayarticle&id=117
6. Install Ryu Controller [Online]. Available: https://osrg.github.io/ryu/
7. *Ryu Controller Ofctl Rest API* [Online]. Available: http://ryu.readthedocs.io/en/latest/app/ofctl_rest.html
8. *Nagios XI – Backend API Access* [Online]. Available: https://assets.nagios.com/downloads/nagiosxi/docs/Accessing_The_XI_Backend_API.pdf

ประวัติผู้แต่ง

ปริญญานิพนธ์เรื่อง : การเฝ้าระวังการใช้บริการทางด่วนข้อมูลผ่านทางเครือข่าย
สาขาวิชา : วิศวกรรมคอมพิวเตอร์
ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะ : วิศวกรรมศาสตร์
ชื่อ : นายธีรชัย สันติวิชัยพานิช
ประวัติ

เกิดเมื่อวันที่ 11 เดือนธันวาคม พ.ศ. 2536 อยู่บ้านเลขที่ 49/1290 ถนนเอกชัย ตำบลโคก-ขาม อำเภอเมือง จังหวัดสมุทรสาคร 74000 สำเร็จการศึกษาในระดับมัธยมศึกษาตอนปลายจากโรงเรียนสมุทรสาครบูรณะ จังหวัดสมุทรสาคร สาขาวิทยาศาสตร์-คณิตศาสตร์ ปีการศึกษา 2555 และสำเร็จการศึกษาในระดับปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2558