การศึกษาประสิทธิภาพของโอเพนซอสไฟร์วอลล์ในราฟเบอรีพาย สำหรับธุรกิจขนาดกลางและขนาดเล็ก

นพดล จินตสุนทรอุไร

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
บัณฑิตวิทยาลัย สาขาวิชาเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยี ไทย-ญี่ปุ่น
ปีการศึกษา 2561

THE PERFORMANCE STUDY OF OPEN SOURCE FIREWALLS IN RASPBERRY PI FOR SMALL AND MEDIUM ENTERPRISE (SMEs)

Noppadol Jintasunthorn-urai

A Term Paper Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology
Graduate School
Thai-Nichi Institute of Technology
Academic Year 2018

หัวข้อสารนิพนธ์	การศึกษาประสิทธิภาพของโอเพนซอสไฟร์วอลล์ในราฟเบอรีพาย สำหรับธุรกิจขนาดกลางและขนาดเล็ก
โดย	นพดล จินตสุนทรอุไร
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษาสารนิพนธ์	ดร. ภาสกร อภิรักษ์วรพินิต

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้นับสารนิพนธ์ฉบับนี้เป็น ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร.พิชิต สุขเจริญพงษ์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบสารนิพนธ์

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. อรรณพ หมั่นสกุล)

..... กรรมการ

(ดร. สรมย์พร เจริญพิทย์)

..... อาจารย์ที่ปรึกษาสารนิพนธ์

(ดร. ภาสกร อภิรักษ์วรพินิต)

นพดล จินตสุนทรอุไร : การศึกษาประสิทธิภาพของโอเพนซอสไฟร์วอลล์ในราฟเบอรีพาย สำหรับ ธุรกิจขนาดกลางและขนาดเล็ก. อาจารย์ที่ปรึกษา : ดร. ภาสกร อภิรักษ์วรพินิต, 101 หน้า.

การค้นคว้าอิสระเรื่อง การศึกษาประสิทธิภาพของโอเพนซอสไฟร์วอลล์ในราฟเบอรีพาย สำหรับธุรกิจขนาดกลาง และขนาดเล็กมีวัตถุประสงค์ เพื่อศึกษาการทำงานของบอร์ด Raspberry Pi สำหรับงานเครือข่าย โดยเน้นที่การทำงานของไฟร์วอลล์ โดยทำการเปรียบเทียบประสิทธิภาพของ Open Source Firewall ในที่นี้จะใช้ IPTable , IPFire และ OpenWRT ในด้านการใช้งานหลักๆ เช่น การส่งไฟล์ด้วย FTP การเข้าถึงเว็บไซต์ การส่ง email และ Live Streaming ทั้งนี้เพื่อหา Open Source Firewall ที่เหมาะสมที่สุดสำหรับ Raspberry Pi และนำเสนอทางเลือกใหม่ในการเลือกใช้อุปกรณ์ที่รองรับมาตรการความปลอดภัยในอินเตอร์เนตที่เหมาะสมทั้งในเรื่องลักษณะการใช้งานของหน่วยงาน และประสิทธิภาพที่เหมาะสมที่หน่วยงานธุรกิจขนาดกลาง และขนาดเล็ก (SMEs) สามารถยอมรับได้

จากงานวิจัยนี้ผลวิจัยพบว่า ทั้งในเรื่องประสิทธิภาพความง่ายในการติดตั้ง และการใช้งาน และการจัดการทรัพยากร ได้ข้อสรุปว่า OpenWRT เป็น Open Source Firewall ที่ดีที่สุดใน 3 Open Source Firewall ทั้งในเรื่องประสิทธิภาพการทำงานที่ดีที่สุดง่ายต่อการใช้งาน และการติดตั้งและการจัดสรรทรัพยากรการใช้งานที่ดีที่สุด

บัณฑิตวิทยาลัย ลายมือชื่อนักศึกษา .....

สาขาวิชา เทคโนโลยีสารสนเทศ ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2561

NOPPADOL JINTASUNTHORN-URAI: THE PERFORMANCE STUDY OF OPEN SOURCE FIREWALLS IN RASPBERRY PI FOR SMALL AND MEDIUM ENTERPRISE (SMEs) ADVISOR : DR. PASKORN APIRUKVORAPINIT, 101 PP.

This independent study investigated the performance study of open source firewalls in Raspberry Pi for small and medium enterprise (SMEs). Its purpose is to study the networking aspect of Raspberry pi in terms of the firewall functionality. The methodology of this independent study is to conduct several experiments by measuring and comparing the performance of three open source firewalls such as IPTable, IPFire and OpenWRT using real-life usage, for example, file transfer by FTP, website access by HTTP, Email by SMTP and Live Streaming. Our experiments concluded that the most suitable open source firewall for Raspberry Pi in SME businesses is OpenWRT with respect to the performance usage, ease of installation and operation, and resource management.

Graduate School Student's signature .....

Field of Information Technology Advisor's signature .....

Academic Year 2018

กิตติกรรมประกาศ

การศึกษาค้นคว้าอย่างอิสระฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ของ บุคคลหลายท่านซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด ซึ่งผู้มีพระคุณท่านแรกที่ผู้ศึกษาใคร่ขอกราบพระคุณคือ ดร. ภาสกร อภิรักษ์วรพินิต อาจารย์ที่ปรึกษาที่ได้ให้ความรู้ คำแนะนำตรวจทาน และแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ทุกขั้นตอน เพื่อให้การเขียนรายงานค้นคว้าอย่างอิสระฉบับนี้สมบูรณ์ที่สุด

ขอขอบคุณร้าน เอ็นเอ็นอิเลคทริค สำหรับการสนับสนุนสถานที่ในการจัดตั้งอุปกรณ์สำหรับการทดลองการค้นคว้าอิสระนี้ทำให้ข้อมูลที่ได้มามีความละเอียด และทำให้เนื้อหาในการศึกษาค้นคว้านี้ครบถ้วน อย่างสมบูรณ์

ขอขอบคุณสถาบันเทคโนโลยีไทย-ญี่ปุ่น สำหรับทุกสาขาวิชาที่ได้ฝึกสอนได้คำแนะนำในการจัดทำรายงานจากการศึกษาค้นคว้าอย่างอิสระฉบับนี้ที่ไม่ได้ กล่าวนาม

ขอขอบพระคุณคุณครอบครัวพ่อ และแม่ที่อยู่เบื้องหลังในความสำเร็จที่ได้ให้ความช่วยเหลือสนับสนุน และให้กำลังใจตลอดมา

นพดล จินตสุนทรอุไร

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

สภาพความเป็นมา แนวทางเหตุผล และปัญหา

การใช้งานอินเทอร์เนตในปัจจุบันนี้ มีหลากหลายรูปแบบด้วยกันเช่น เพื่อไปหาข้อมูลอ่านข่าวสาร หรือเพื่อทำการดาวน์โหลดข้อมูลใช้ในการทำธุรกรรมต่างๆ ในการเข้าใช้งานดังกล่าวอาจจะมีภัยคุกคามทางอินเทอร์เนตต่างๆ มากมายอาทิเช่น การใส่คำสั่ง SQL เพื่อไปโจมตีฐานข้อมูล (SQL injection) [1] การเปิดช่องโหว่ของการยืนยันตัวตน (Broken Authentication) การฝังคำสั่งเข้าหน้าเวปที่มีช่องโหว่ (Cross Site Scripting: XSS) จากการสำรวจข้อมูลทางสถิติของ Internet Security Threat Report ประจำปี 2018 [2] จำนวนการปิดกั้นเว็บไซต์ที่อันตรายนั้น มีจำนวนมากกว่าปีที่แล้วถึง 92% และ อัตราของการโจมตีด้วยสแปม (Spam) ของหน่วยงานธุรกิจขนาดย่อมที่มีขนาดจำนวน 1-250 คน อยู่ที่ 54.9% ซึ่งเป็นอันดับ 2 ในกลุ่มขนาดหน่วยงาน และใน 1 ผู้ใช้งานมีจำนวน Spam ที่ได้รับเฉลี่ย 45.2 เห็นได้ว่าอันตรายของภัยคุกคามทางอินเตอร์เน็ตต่อหน่วยงานธุรกิจขนาดกลางและขนาดเล็กนั้นมีมากยิ่งขึ้น และเป็นสิ่งสำคัญที่จะต้องคำนึงถึงอย่างหลีกเลี่ยงไม่ได้

มาตรการความปลอดภัยของอินเทอร์เนตในหน่วยงานธุรกิจต่างๆ มีความแตกต่างกันไป ทั้งนี้ขึ้นอยู่กับประเภทของธุรกิจ และลักษณะงานของธุรกิจด้วย สำหรับองค์กรขนาดใหญ่ทั่วไปมีการป้องกันภัยคุกคามทางอินเทอร์เน็ตที่มีประสิทธิภาพที่ดี ด้วยการลงทุนกับอุปกรณ์ต่างๆ ตัวอย่างเช่น ไฟร์วอลล์ระบบการป้องกันการบุกรุก (Intrusion Prevention Systems or IPS) แอพพลิเคชั่นเกทเว0ย์ (Application Gateway) แต่ว่าอุปกรณ์ดังกล่าวนั้น มีราคาค่อนข้างสูง เมื่อเทียบกับ รายได้ของหน่วยงานธุรกิจขนาดกลาง และขนาดย่อมโดยเฉพาะธุรกิจขนาดเล็กจะไม่สามารถลงทุนในส่วนนี้ได้เพราะจะต้องเพิ่มบุคลากรในการจัดการอุปกรณ์ดังกล่าว ซึ่งค่อนข้างซับซ้อน และเป็นการสิ้นเปลืองจนเกินไป

ไฟร์วอลล์เป็นอุปกรณ์หลักในการป้องกันความปลอดภัยของอินเทอร์เนต ที่มีความสามารถในการทำงานป้องกันมากมาย ยกตัวอย่างเช่น การตั้งกฎของไฟร์วอลล์ (Firewall Rules) เพื่อปิดกั้นไม่ให้ IP ที่ไม่พึงประสงค์เข้ามาในเครือข่าย การจัดการเส้นทางของเครือข่าย (Routing Management) การป้องกันการบุกรุก การติดตามตรวจสอบการใช้งานโดยใช้แดชบอร์ด (Monitoring by Dashboard) เป็นต้น ซึ่งเป็นการทำงานหลักๆ ของการป้องกันความปลอดภัยของอินเทอร์เนตที่ควรมี และตอบสนองกับมาตรการความปลอดภัยของอินเทอร์เนตในปัจจุบัน

ราฟเบอรี พาย (Raspberry Pi) [3] เป็นบอร์ดระบบฝังตัว ที่มีลักษณะคล้ายคอมพิวเตอร์ขนาดเล็กมีขนาดเท่ากับบัตรเครดิตซึ่งพัฒนาจาก มหาวิทยาลัยแคมบริดจ์เมื่อปี 2012 โดยราคาอยู่ที่

ประมาณ 35 เหรียญสหรัฐ โดยบอร์ด ราสพ์เบอรี พายนั้น ประกอบด้วย หน่วยประมวลผลสถาปัตยกรรมแบบอาร์ม (Advance RISC Machine : ARM) และมีหน่วยประมวลผลภาพ 3 มิติสำหรับการฉายภาพวีดีโอขนาดความละเอียดสูง มีหน่วยความจำชั่วคราว (Random Access Memory: RAM) ที่แตกต่างกันไปในแต่ละรุ่น และสามารถเพิ่มหน่วยความจำโดยใช้ เอสดีการ์ด (SD Card) ได้ พอร์ทการเชื่อมต่อประกอบด้วย เฮชดีเอ็มไอ (HDMI) ยูเอสบี (USB) และพอร์ต อีเทอร์เนต (Ethernet) ที่มีความนิยมอย่างมากในหน่วยอุตสาหกรรมระบบโครงข่ายอัจฉริยะราสพ์เบอรี พาย ทำงานด้วยระบบปฏิบัติการ ลีนุกซ์สามารถใช้ติดตั้งโปรแกรมทำเป็นเว็บเซิฟเวอร์ หรือจะเป็นหน่วยควบคุมอื่นๆ ได้ โดยที่ระบบปฏิบัติการนั้นจะเก็บไว้ในเอสดีการ์ด (SD Card) เมื่อทำการติดตั้งเสร็จเรียบร้อย ทุกหน่วยประมวลผลของ ราสพ์เบอรี พายก็จะสามารถเริ่มทำงานได้ทันที

งานวิจัยนี้เห็นว่าRaspberry Pi ซึ่งเป็น บอร์ดระบบฝังตัวขนาดเล็ก และราคาถูกรวมถึงความสามารถในการใช้งานด้านเครือข่ายสามารถจะทำการประยุกต์การใช้งานเป็นไฟร์วอลล์ (firewall) ซึ่งเป็นอุปกรณ์หลักในการป้องกันภัยคุกคามทางอินเตอร์เนต โดยนำโอเพนซอรส์ไฟร์วอลล์ (Open Source Firewall) ที่นิยมอยู่ในปัจจุบันมาทำการศึกษาเปรียบเทียบประสิทธิภาพในการใช้งาน เพื่อที่จะหา Open Source Firewall ที่เหมาะสมและคุ้มค่าที่สุดกับการใช้งานร่วมกับ Raspberry pi

จากที่กล่าวมาข้างต้นจะนำไปเป็นทางเลือกในการเลือกซื้อให้กับหน่วยงานธุรกิจขนาดกลางและขนาดย่อม ให้สามารถเลือกใช้อุปกรณ์ป้องกันความปลอดภัยของอินเตอร์เนต ในราคาย่อมเยาว์และมีประสิทธิภาพที่เทียบเคียงกับ ไฟร์วอลล์ (firewall) ในระดับองค์กรขนาดใหญ่ รวมถึงการยกระดับขีดความสามารถของ Raspberry pi ในการใช้งานในด้านเครือข่าย

วัตถุประสงค์ของงานวิจัย

1. เพื่อทำการศึกษาการทำงานของบอร์ด Raspberry pi สำหรับเครือข่าย โดยเน้นที่การทำงานของไฟร์วอลล์
2. เพื่อศึกษาคุณสมบัติและการทำงานของไฟร์วอลล์ ว่ามีความสามารถในการทำงานอย่างไร เพื่อที่จะเป็นเกณฑ์ในการวัดประสิทธิภาพของ Open Source Firewall ที่จะมาติดตั้งใน Raspberry pi
3. เพื่อทำการศึกษาเปรียบเทียบประสิทธิภาพของ Open Source Firewall ต่างๆ เพื่อที่จะหาOpen Source Firewall ที่เหมาะสมที่สุดสำหรับ Raspberry pi
4. เพื่อนำเสนอทางเลือกใหม่ในการเลือกใช้อุปกรณ์ที่รองรับ มาตรการความปลอดภัยในอินเตอร์เนตที่เหมาะสมทั้งในเรื่องรายได้ของธุรกิจ และประสิทธิภาพที่เหมาะสม ที่หน่วยงานธุรกิจขนาดกลางและขนาดเล็ก (SMEs) สามารถยอมรับได้

ขอบเขตของงานวิจัย

1. ขอบเขตของกลุ่มตัวอย่าง

1.1 Open Source Firewall สำหรับในการทดสอบประสิทธิภาพ โดยมีดังนี้

1.1.1 IPTable

1.1.2 IPFire

1.1.3 OpenWRT

1.2 กลุ่มของชุดไอพี (IP) ในงานวิจัยนี้จะเตรียมข้อมูลมา 1 วง IP เพื่อที่ทำการทดสอบการเข้าถึงของตัวอุปกรณ์ และการทดสอบประสิทธิภาพของอุปกรณ์ อย่างเช่น ความสามารถของการสร้างกฎของไฟร์วอลล์ (Firewall Rules) ความสามารถในการรับส่งปริมาณข้อมูลของโพรโตคอล (Protocol Throughput) เป็นต้น

2. ขอบเขตในด้านอุปกรณ์การใช้งาน

ในด้านอุปกรณ์ที่ใช้งาน จะมี บอร์ด Raspberry Pi ซึ่งได้ทำการ ติดตั้งระบบปฏิบัติการ Linux เรียบร้อยแล้ว

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ทางเลือกใหม่สำหรับการเลือกซื้ออุปกรณ์ที่ตอบโจทย์มาตรการรักษาความปลอดภัยให้กับหน่วยงานที่มีข้อจำกัดในด้านต่างๆ เช่น ข้อจำกัดในด้านบุคลากรการดูแล หรือข้อจำกัดในเรื่องของรายได้ในงบประมาณในการซื้อ เป็นต้น

2. สามารถเลือก Open Source Firewall ที่มีประสิทธิภาพที่เหมาะสมกับขนาดของธุรกิจและขนาดรายได้ของหน่วยงานธุรกิจ SMEs

3. ได้ Open Source Firewall ที่มีประสิทธิภาพเทียบเคียงกับไฟร์วอลล์ ในระดับองค์กรขนาดใหญ่รวมถึงการยกระดับขีดความสามารถของ Raspberry Pi

4. ได้แนวทางในการพัฒนา และต่อยอดการประยุกต์ใช้ Raspberry Pi ในแง่ของอุปกรณ์ในแง่ของเครือข่าย

นิยามศัพท์เฉพาะ

คำศัพท์ที่ 1 SMEs เป็นชื่อย่อขอหน่วยงานธุรกิจขนาดกลาง และขนาดเล็ก

คำศัพท์ที่ 2 ไฟร์วอลล์ (Firewall) คือ อุปกรณ์สำหรับการกรองการส่งข้อมูลจากอินเตอร์เน็ตภายนอก โดยการกำหนดกฎ และระเบียบของตัวอุปกรณ์ซึ่ง

คำศัพท์ที่ 3 ผู้ใช้งาน ในที่นี้หมายถึงผู้ที่ใช้งานอินเตอร์เน็ต

คำศัพท์ที่ 4 Raspberry Pi เป็นบอร์ดบอร์ดระบบฝังตัว ขนาดเล็ก และราคาถูกซึ่งในที่นี้จะเป็นอุปกรณ์หลักในการนำ Open Source Firewall เข้าไปติดตั้งกับอุปกรณ์นี้

คำศัพท์ที่ 5 Open Source Firewall คือโปรแกรมในระบบปฏิบัติการ Linux ที่มีคุณสมบัติการทำงานของไฟร์วอลล์ ในที่นี้จะเป็นกลุ่มตัวอย่างในการทดสอบประสิทธิภาพ

บทที่ 2
แนวคิด ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้อง

คุณสมบัติเบื้องต้นของ Firewall (Introduction to Firewall)

Firewall [4] เป็นอุปกรณ์ทางด้านเครือข่ายประเภทหนึ่งมีหน้าที่ในการป้องกันการเข้าถึงของข้อมูลจากต้นทางที่ไม่พึงประสงค์ทำให้ไม่สามารถเข้ามาในระบบเครือข่ายภายในได้ ทั้งนี้การป้องกันการเข้าถึงข้อมูลของ Firewall นั้น มีลักษณะที่แตกต่างกันออกไป โดยจะขึ้นอยู่กับ การประมวลผลภายใน คุณสมบัติและขีดความสามารถของอุปกรณ์นั้น ๆ

การทำงานของ Firewall เริ่มต้นที่การกำหนดจุดทีปิดกั้นอยู่หนึ่งจุด ซึ่งเปรียบเสมือนกำแพงที่ทำการป้องกัน แล้วจะทำการรายงาน และทำการตรวจสอบข้อมูลภายนอกที่เข้ามา โดยที่รายละเอียดในรายงานนั้น Firewall จะทำการจำแนกข้อมูลออกมาอย่างเช่น ข้อมูล IP ต้นทาง IP ปลายทาง เป็นต้น เพื่อรายงานกับผู้ดูแลระบบได้รับทราบ โดยที่ผู้ดูแลจะทำการกำหนดเงื่อนไขในการปิดกั้นหรือกฎ ที่จะอนุญาตให้ข้อมูลบางอย่างสามารถเข้าไปยังเครือข่าย หรือปิดกั้นไม่ให้เข้ามาในเครือข่าย

Firewall สามารถแบ่งได้ 4 ประเภทด้วยกัน ตามรายละเอียดต่อไปนี้

1. Packet Filter มีลักษณะการทำงานแบบกรองกลุ่มข้อมูล (Packet) แต่ละครั้งที่เข้ามาในเครือข่าย และส่งต่อไปยังผู้ใช้งาน ซึ่งลักษณะนี้จะทำงานได้ผลไม่ค่อยดีสำหรับ IP ที่มีการปลอมแปลงมา หรือ IP Spoofing
2. Application Gateway จะใช้กลไกทางด้านความปลอดภัยที่อนุญาตให้สามารถใช้Application บางอย่างได้ เช่น การอนุญาตให้ใช้คำสั่ง telnet เป็นต้น
3. Circuit - Level Gateway มีการทำงานโดยที่ข้อมูลจะสามารถส่งต่อระหว่างเครื่องในเครือข่ายได้ก็ต่อเมื่อสามารถสร้างการเชื่อมต่อแบบ TCP หรือ UDP ได้
4. Proxy Server จะทำการดักจับข้อความทั้งหมดที่เข้ามาในเครือข่าย และออกจากเครือข่าย ซึ่งจะมีผลดีในการซ่อนหมายเลขเครือข่ายที่แท้จริง

ระบบปฏิบัติการ Linux และ Debian

Linux Kernel [5] คือ ระบบปฏิบัติการซอฟแวร์(software) ที่อยู่ในระดับล่างที่สุดของระบบปฏิบัติการ Linux ซึ่งมีคุณสมบัติในการจัดการอุปกรณ์ของ ฮาร์ดแวร์ (hardware) เปิด

โปรแกรม และดำเนินการซ่อมแซม หรือตั้งค่าความปลอดภัย หรือองค์ประกอบอื่นๆ ของระบบซึ่งก่อตั้งโดย Linus Torvalds ในปี 1991 ซึ่ง kernel ดังกล่าวนี้มีความสัมพันธ์ กับซอฟแวร์ (software) ขนาดเล็ก ในระบบปฏิการ Linux แต่ทั้งนี้เองในส่วนนี้ยังต้องประสานงานการทำงานร่วมกันในส่วนของ core system ของระบบปฏิบัติการ

ส่วนระบบปฏิบัติการ Debian [6] เริ่มก่อตั้งเมื่อเดือนสิงหาคมปี 1993 โดย Ian Murdock เป็นระบบปฏิบัติการที่ใช้งานแพร่หลายและใช้งานได้ฟรี ซึ่งสามารถใช้ได้กับ Linux Kernel และ FreeBSD Kernel และทำงานได้ทั้งคอมพิวเตอร์ทั่วไปและบอร์ดฝังตัวอัจฉริยะ (Embedded Board) ยกตัวอย่างเช่น Raspberry Pi เป็นต้น ซึ่งเป็นที่นิยมอย่างมากในกลุ่มนักพัฒนาที่จะต้องการพัฒนา software โดยใช้เนื้อที่การใช้งานเพียงเล็กน้อยเท่านั้น เมื่อเทียบกับระบบปฏิบัติการพื้นฐานอื่นๆ

Open Source Firewall

ซอฟต์แวร์ IPTables [7]

IPTables เป็นซอฟต์แวร์โอเพ่นซอร์สที่เป็นส่วนหนึ่งของลีนุกซ์เคอร์เนล ทำงานในระดับเน็ตเวิร์กเลเยอร์ พัฒนาโดยเน็ต ฟิลเตอร์ (net filter: www.netfilter.org) มีความสามารถในการคัดกรองแพ็คเก็ตข้อมูลและรองรับการทำ NAT

Filter table เป็นตารางใน iptables ที่ใช้งานมากที่สุด เป็นจุดที่ใช้ในการตรวจสอบและควบคุมการผ่านเข้าออกของแพ็คเก็ต ซึ่ง filter table ประกอบด้วย INPUT, OUTPUT และ FORWARD chain โดยทั่วไปแล้วการใช้งาน iptables จะต้องพิมพ์คำสั่งผ่าน command line เพื่อสร้างเป็นกฎของไฟร์วอลล์ ซึ่งกฎที่เพิ่มเข้ามาใหม่จะถูกเรียงต่อท้ายกฎเดิม ในการตรวจสอบแพ็คเก็ตของไฟร์วอลล์นั้นไฟร์วอลล์จะจับคู่แพ็คเก็ตที่ตรงกับเงื่อนไขกับกฎของไฟร์วอลล์แล้วทำตามเงื่อนไขของกฎนั้นเช่น Accept, Deny, Drop

วิธีการเขียนกฎเบื้องต้นของ IPTables [8]

ปกติแล้ว IPTable นั้น จะเป็น software ที่รวมอยู่ในระบบปฏิบัติการ Linux อยู่ด้วย ซึ่งถ้าติดตั้งระบบปฏิบัติการ Linux อย่างเช่น Ubuntu จะสามารถใช้คำสั่งของ IPTable ได้จากหน้า command line

- คำสั่งพื้นฐาน

```
sudo iptables -L
```

- คำสั่งนี้จะเป็นคำสั่งที่ตรวจสอบกฎของไฟร์วอลล์ที่ได้ตั้งไว้อยู่ปัจจุบัน ซึ่งค่าเริ่มต้นนั้น จะไม่มีกฎอยู่

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- การตั้งค่าอนุญาตเซสชั่นที่ยอมรับได้ (Allow Established Session)

เพื่อที่จะสามารถอนุญาตให้เซสชั่นนั้นสามารถรับ traffic ได้ จะต้องพิมพ์คำสั่ง

```
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

เพื่อเปิดการอนุญาตเซสชั่น

- การตั้งค่าอนุญาตให้ traffic ขาเข้า เข้ามาใน port ที่ระบุไว้ เช่นเมื่อต้องการปิดกั้น traffic ทั้งหมด แต่ยังสามารถใช้งาน sshจะต้องพิมพ์คำสั่ง

```
sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

รายละเอียดของคำสั่งสามารถแบ่งเป็นตัวเลือกดังนี้

- A INPUT หมายถึง การเพิ่มกฎนี้เข้าไปใน input chain
- p tcp หมายถึง เช็คพอร์ตที่เป็น tcp
- dport ssh หมายถึง ตรวจสอบพอร์ตปลายทางว่าเป็นพอร์ต ssh
- j ACCEPT หมายถึง ถ้าข้อมูลข้างต้นนั้นเป็นจริง ก็ให้ทำการอนุญาตการทำงาน
- การปิดกั้น traffic

การปิดกั้น traffic ที่ไม่ต้องการ หรือ เมื่อใช้ในการปิดกั้นไว้ตรงท้ายของ chain rule จะต้องพิมพ์คำสั่ง

```
sudo iptables -A INPUT -j DROP
```

เพื่อทำการปิดกั้นข้อมูลที่ไม่ต้องการนั้นทิ้งไป

ซอฟต์แวร์ IPFire [9]

เป็นซอฟต์แวร์ที่ออกแบบเพื่อทำการปรับปรุงและพัฒนาเกี่ยวกับความปลอดภัยทางด้านเครือข่าย ซึ่งใช้ได้ตั้งแต่ในระดับองค์กรธุรกิจขนาดเล็ก จนถึง ธุรกิจขนาดใหญ่ ซึ่งซอฟต์แวร์ดังกล่าวนี้

มีลักษณะการทำงานที่ง่าย และมีหน้าจอการทำงานในส่วนของผู้ใช้งานค่อนข้างสะดวกต่อการใช้งาน โดยใน ซึ่งคุณสมบัติการใช้งานในภาพรวมและเกี่ยวข้องกับงานวิจัยมีดังนี้

คุณสมบัติการทำงานในด้านไฟร์วอลล์ (Firewall)

ในส่วนนี้จะอ้างอิงจากการทำงานของการตรวจสอบแพ็คเก็จแบบสถานะสัมพันธ์ (Stateful packet inspection) ซึ่งจะสร้างการทำงานบน framework ที่ชื่อว่า netfilter โดยลักษณะการทำงานของส่วนนี้จะแจ้งเตือนการทำงานถึง 4 ระดับด้วยกัน ซึ่งเปรียบได้ถึง 4 ระดับของความปลอดภัย

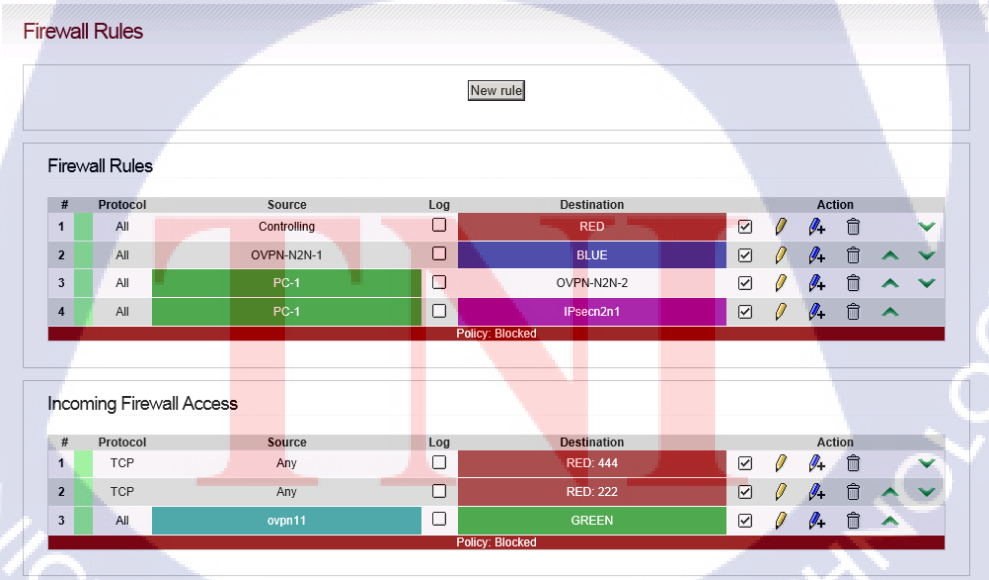
สีเขียว แสดงถึง ระดับปลอดภัย (Safe area)

สีแดง แสดงถึง ระดับอันตราย (Danger Area)

สีน้ำเงิน แสดงถึง ระดับของพื้นที่ไร้สาย หมายความว่า เป็นส่วนหนึ่งของเครือข่ายภายใน (Wireless Network)

สีส้ม แสดงถึง เขตของการควบคุมของไฟร์วอลล์ (Demilitarized Zone)

ซึ่งในระดับที่กล่าวมาข้างต้นนั้น จะเอาไปใช้ในการตั้งกฎในไฟร์วอลล์ (Firewall Rules) ซึ่งจะสามารถกำหนดให้อุปกรณ์ดังกล่าวนั้นสามารถจัดการกับ IP ที่เข้าข่ายอันตรายได้ อย่างเช่นการลบทิ้ง หรือปิดกั้นไม่ให้ผ่าน เป็นต้น



รูปที่ 2.1 รูปภาพหน้าจอการทำงานของการตั้งกฎไฟร์วอลล์ของ IPFire [9]

ซอฟต์แวร์ OpenWRT [10]

OpenWRT เป็น ซอฟต์แวร์ในระบบปฏิบัติการ Linux ซึ่งใช้สำหรับอุปกรณ์ฝังตัวอัจฉริยะ (Embedded Device) โดย OpenWRT จะทำการสร้างระบบไฟล์ขึ้นมาเพื่อให้ผู้ใช้งานสามารถปรับแต่งได้ตามความต้องการ ซึ่งเอื้ออำนวยต่อนักพัฒนาซอฟต์แวร์ที่สนใจในการสร้างอุปกรณ์เน็ตเวิร์ค โดยใช้ OpenWRT เป็นโครงสร้างหลักเพราะเป็น Open Source โดยใน OpenWRT มีการใช้งานที่ง่าย และมี User Interface ที่สามารถใช้งานได้ง่าย และเข้าใจได้ง่ายสำหรับผู้ใช้งานทั่วไปซึ่งนักพัฒนาสามารถนำ OpenWRT ไปต่อยอดกับอุปกรณ์ Network ได้หลากหลายรูปแบบซึ่งเป็นอีกหนึ่ง Open Source Firewall ที่ได้รับความนิยมอยากสูงมากในกลุ่มนักพัฒนา

รูปที่ 2.2 ตัวอย่าง User Interface ในการใช้งานไฟร์วอลล์ ของ OpenWRT [9]

ราสพ์เบอรี พาย (Raspberry Pi)

ราสพ์เบอรี พาย(raspberry pi: RPi) [11] เป็นบอร์ดระบบฝังตัว ที่มีลักษณะคล้ายคอมพิวเตอร์ขนาดเล็ก มีขนาดเท่ากับบัตรเครดิต ซึ่งพัฒนาจาก มหาวิทยาลัยแคมบริดจ์ เมื่อปี 2012 โดยราคาอยู่ที่ประมาณ 35 เหรียญสหรัฐ โดยบอร์ด ราสพ์เบอรี พาย นั้น ประกอบด้วย หน่วยประมวลผล สถาปัตยกรรมแบบอาร์ม (advance RISC machine : ARM) และมีหน่วยประมวลผลภาพ 3 มิติ สำหรับการฉายภาพวีดีโอ ขนาดความละเอียดสูง มีหน่วยความจำชั่วคราว (random access memory: RAM) ที่แตกต่างกันไปในแต่ละรุ่น ซึ่งจะกล่าวต่อไป และสามารถเพิ่ม

หน่วยความจำโดยใช้ เอสดีการ์ด (SD card) ได้ พอร์ทการเชื่อมต่อประกอบด้วย เฮชดีเอ็มไอ (HDMI) ยูเอสบี (USB) และพอร์ต อีเทอร์เนต(ethernet)ซึ่งมีความนิยมอย่างมากในหน่วยอุตสาหกรรมระบบโครงข่ายอัจฉริยะ โดยที่ราสพ์เบอรีพาย นั้น ทำงานด้วยระบบ ลีนุกซ์ ซึ่งก็หมายความว่า สามารถใช้ติดตั้งโปรแกรม ทำเป็นเว็บเซิฟเวอร์ หรือจะเป็นหน่วยควบคุมอื่นๆ ได้ โดยที่ระบบปฏิบัติการนั้นจะเก็บไว้ในเอสดีการ์ด (SD card) เมื่อทำการติดตั้งเสร็จเรียบร้อยทุกหน่วยประมวลผลของ ราสพ์เบอรีพาย ก็จะเริ่มทำงานทันที

ราสพ์เบอรี พาย โมเดล บี (raspberry pi model B)เป็นรุ่นของบอร์ดที่ได้รับความนิยมและใช้กันมากที่สุดในขนาดนี้ ซึ่งในรุ่นนี้ได้รับการพัฒนาจากรุ่น เอ และมีส่วนที่สำคัญดังต่อไปนี้

- หน่วยประมลผล ARM 700 MHz แกนเดียว
- RAM 512 MB
- พอร์ทยูเอสบี 2 ช่อง
- พอร์ทอีเทอร์เนต 100 Mb ใช่สำหรับเสียบสายแลน
- กำลังส่งไฟฟ้าอยู่ที่ 700 mA

ราสพ์เบอรี พาย 2 (raspberry pi 2) เป็นรุ่นของบอร์ดที่พัฒนาจากโมเดล บี พลัส ในปัจจุบัน ซึ่งในรุ่นนี้นั้น มีคุณสมบัติหลักเหมือนกับ โมเดล บี พลัส ทั้งหมด แต่จะพัฒนาเรื่องของประสิทธิภาพการทำงาน รวมถึงเพิ่ม ความหลากหลายเพื่อรองรับเทคโนโลยีในอนาคต เช่น รองรับการทำงานกับ วินโดวส์ 10 และการใช้งานกับกราฟฟิคระดับสูง เป็นต้น ซึ่งรายละเอียดของคุณสมบัติมีดังนี้

- หน่วยประมวลผล ARM Cortex-v7 900 MHz 4 แกน
- RAM 1 GB LPDR2 SDRAM หน่วยความจำ 2 ตัว
- พอร์ทยูเอสบี 4 ช่อง
- พอร์ทอีเทอร์เนต 100 Mb ใช่สำหรับเสียบสายแลน
- กำลังส่งไฟฟ้าสูงสุดอยู่ที่ 1000 mA

งานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้

<u>การเปรียบเทียบประสิทธิภาพของ firewall ในระดับอุปกรณ์</u>

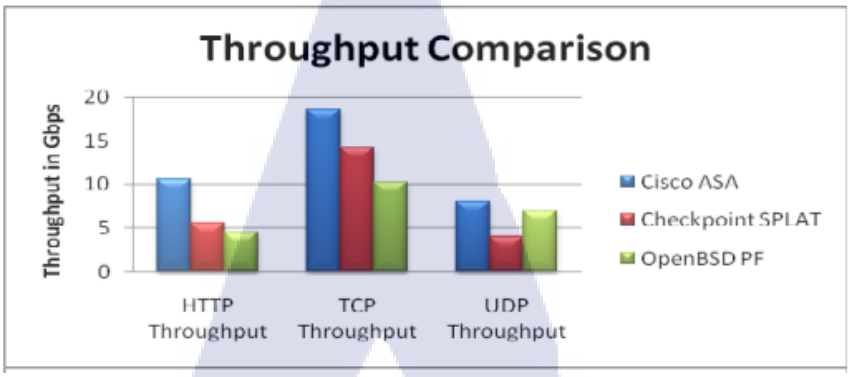
งานวิจัยเรื่องการวัดประสิทธิภาพและการวิเคราะห์แบบเปรียบเทียบของไฟร์วอลล์ ได้นำผลลัพธ์ของไฟล์วอลล์แต่ละตัวมาเปรียบเทียบกัน โดยที่อุปกรณ์ที่นำมาทดสอบนั้น มีไฟร์วอลล์จำนวนทั้งหมด 3 ตัวด้วยกัน คือ Cisco ASA , CheckPoint SPLAT และ OpenBSD PF วิธีการทดสอบใช้เครื่องมือที่ชื่อ Curl-loader สำหรับจำลองพฤติกรรมการทำงานของ HTTP/HTTPS จำนวนมาก และทดสอบประสิทธิภาพ 8 ด้านด้วยกัน [12] ซึ่งได้แก่

1. Firewall Licensing
2. Firewall Management
3. Application Intelligence
4. HTTP Throughput
5. TCP Throughput
6. Concurrent Connection
7. UDP Throughput
8. Connection Per Second

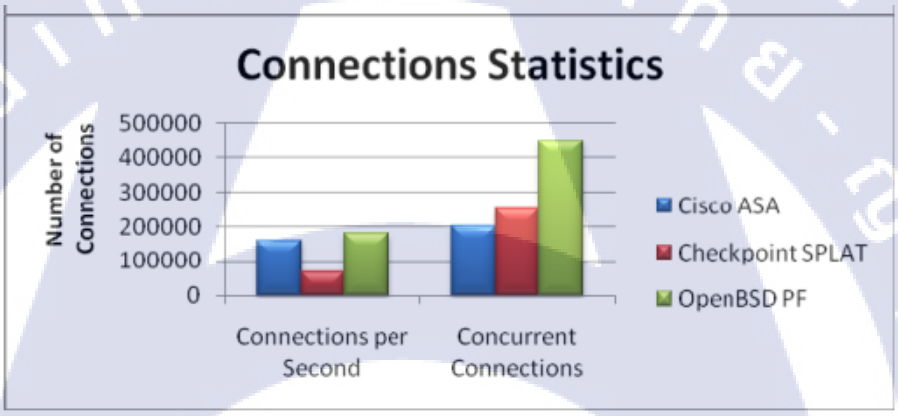
โดยที่ผลการทดสอบใน 8 ด้านจะสรุปผลออกมาในรูปแบบของตารางดังแสดงในตารางที่ 2.1 และออกมาเป็นรูปแบบกราฟ ในรูปที่ 2.3 และ 2.4

ตารางที่ 2.1 ตารางสรุปผลประสิทธิภาพโดยแยกออกเป็น KPI ในด้านต่างๆ

Key Performance Indicator (KPI)	**System Under Test – Firewall**		
	CISCO ASA	**CP SPLAT**	**OpenBSD PF**
Firewall Licensing	Proprietary	Proprietary	BSD
Application Intelligence	Yes	Yes	No
Firewall Management	Local	Centralize	Local
HTTP Throughput (Gbps)	10.6	5.6	4.5
TCP Throughput (Object Size = 512 kb) (Gbps)	18.6	14.2	10.2
Concurrent Connection(Gbps)	200K	250K	500K
UDP Throughput (Object Size = 512 kb) (Gbps)	8	4	7
Connection Per Second	160K	68K	180K



รูปที่ 2.3 แสดงผลประสิทธิภาพในด้านปริมาณที่รับเข้ามาและจำนวน Connection ที่สามารถเข้ามาในช่วงเวลาหนึ่ง [12]



รูปที่ 2.4 แสดงผลประสิทธิภาพ ในด้าน ปริมาณที่รับเข้ามาและจำนวน Connection ที่สามารถเข้ามาในช่วงเวลาหนึ่ง [12]

ส่วนการทดสอบในเรื่องของความปลอดภัยจะทดสอบโดยการนำเครื่องทดสอบนั้นทำการสร้าง traffic การใช้งานขึ้นมาจำนวนหนึ่งซึ่งใช้วิธีการนำ FTP Server วางไว้ที่เครือข่ายภายใน และส่งไปยังภายนอกเพื่อเข้าถึงข้อมูล และใช้เครื่องมือที่ชื่อว่า NetWag ในการช่วยประมวลผล และจับผลทดสอบ ซึ่งผลลัพธ์ที่ได้นั้น ไฟร์วอลล์ทั้ง 3 ตัวสามารถทำงานในเรื่องของการปิดกั้น port ที่เข้ามาโจมตีได้ทั้งหมด โดยที่ OpenBSD PF นั้นก็ยังสามารถทำได้เมื่อเทียบกับ Firewall ขนาดใหญ่ของ 2 ตัวที่เหลือ อีกทั้งไฟร์วอลล์ดังกล่าวนั้น ยังสามารถตรวจจับข้อมูลที่เข้ามาและสามารถสรุปผลออกมาในรูปแบบหน้าจอการทำงานที่สามารถอ่านได้ง่าย

การเปรียบเทียบประสิทธิภาพของ firewall ในระดับ Software

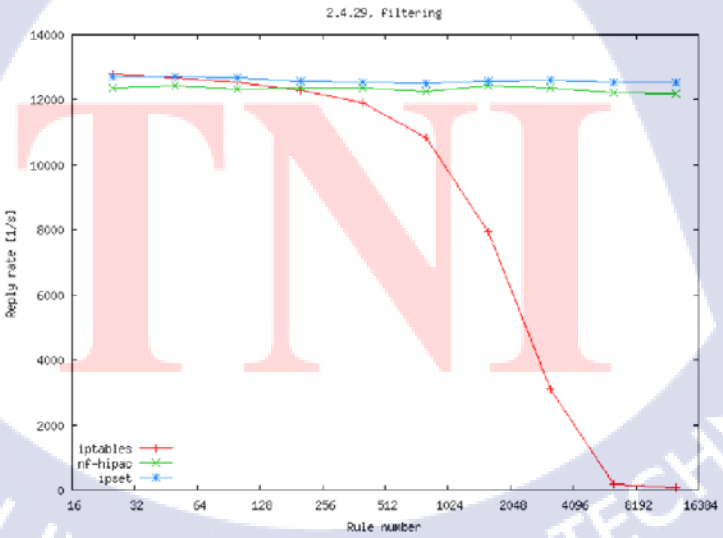
โดยการทดสอบประสิทธิภาพของ Firewall ในด้าน Software นั้นจะแสดงการเปรียบเทียบประสิทธิภาพระหว่าง netfilter กับ ซอฟต์แวร์ตัวอื่นๆ เช่น nf-hipac และ ipset โดยใช้วิธีการวัดแบบ วิธีการ plain routing , connection tracking , filtering และการ NAT โดยที่การวัดประสิทธิภาพนั้น จะมีรายละเอียดตัวอย่างที่ทดลอง โดยที่จะทดลองในการเอาข้อมูลที่เป็นโพรโตคอล TCP ในการทดสอบประสิทธิภาพ โดยมีรายละเอียดดังนี้ [13]

1. จำนวนพอร์ต (Port) ที่รองรับที่วิ่งผ่านด้วยโพรโตคอล(Protocol) TCP โดยมีจำนวนสูงสุด 1075 พอร์ท ต่อการเรียกการใช้งาน 1 วินาที

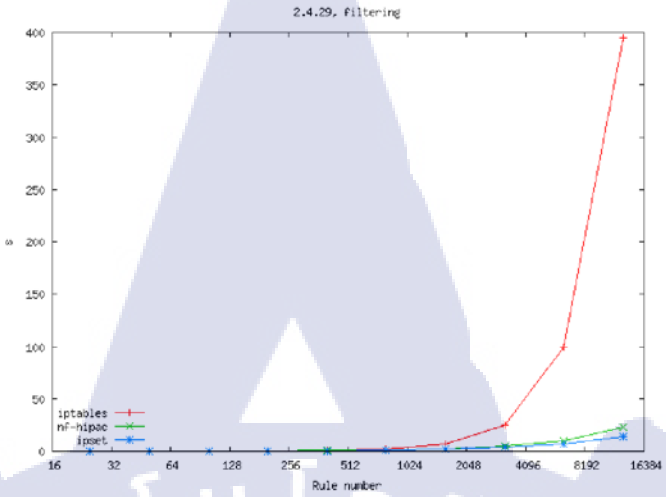
2. จำนวน descriptors ของการเปิดไฟล์ต่อ 1 process โดยปกติแล้วระบบปฏิบัติการลินุกซ์จะรองรับการเปิด descriptors 1024 ครั้งต่อ 5 วินาที และ 204 ครั้งต่อ 1 วินาที เป็นจำนวนที่รองรับได้ต่ำที่สุด

3. พื้นที่ความจำของ Socket โดยในที่นี้จะใส่ข้อมูลที่มีขนาด 16kB จากพื้นที่ที่รองรับทั้งหมด 40MB และเมื่อช่องข้อมูลเต็ม จะสามารถรองรับการเข้ามาของข้อมูล (concurrent) ได้ทั้งหมด 2560 ครั้ง และอุปกรณ์ที่นำมาใช้ในการทดลองนั้นจะใช้เครื่อง Server หน่วยประมวลผล Pentium 4 มีหน่วยความจำ 512 MB ในการทดลอง

โดยที่ผลการทดลองนั้นจะสามารถสรุปได้ว่า Netfilter มีข้อเสียอยู่ที่ประสิทธิภาพการรองรับกฎนั้น จะทำงานได้ไม่ดีเท่าที่ควรเมื่อจำนวนกฎมีจำนวนมากใน 1 chain แต่จะสามารถทำงานได้ดีกว่าไฟร์วอลล์ซอฟต์แวร์ตัวอื่นๆ เมื่อจำนวนกฎมีน้อย ซึ่งจะเป็นไปตามรูปที่ 2.5 และ 2.6



รูปที่ 2.5 แสดงถึงเวลาในการตอบสนองของกฎของไฟร์วอลล์เมื่อมีจำนวนเพิ่มขึ้นเรื่อยๆ [13]

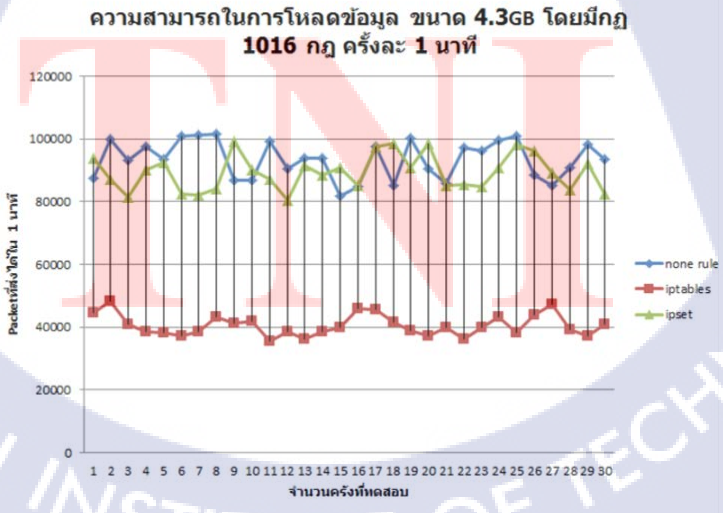


รูปที่ 2.6 แสดงถึงเวลาในการตอบสนองของการเพิ่มกฎของไฟร์วอลล์ 1 ครั้ง 4 [13]

และอีกการทดลองหนึ่ง เป็นการทดสอบโดยการนำข้อมูลขนาดใหญ่ไว้ที่ Server และใช้ client ดาวน์โหลดไฟล์ผ่านไฟร์วอลล์โดยเปรียบเทียบใน 3 กรณีด้วยกัน [7] คือ

1. การทดสอบโดยที่ไม่ได้ใช้ Software ใดๆ
2. การทดสอบโดยใช้ IPTable
3. การทดสอบโดยใช้ IPSet

ซึ่งจะใช้โปรแกรมสำหรับการ monitor เพื่อทดสอบความเร็วในดาวน์โหลดในระยะเวลา 1 นาที ซึ่งผลที่ได้นั้นจะเป็นไปตามรูปที่ 2.7



รูปที่ 2.7 แสดงกราฟเชิงเส้นของความสามารถในการดาวน์โหลดข้อมูลในกรณีที่ต่างกัน [7]

จากรูปนี้จะแสดงรายละเอียดของกราฟโดยที่

เส้นสีฟ้า จะเป็นการดาวน์โหลดไฟล์แบบไม่ใช้กฎของไฟร์วอลล์

เส้นสีแดง เป็นการดาวน์โหลดไฟล์โดยใช้ IPTable 1016 กฎ

เส้นสีเขียว เป็นการดาวน์โหลดไฟล์โดยใช้ IPset เข้ารวมยุบกฎของ IPtable จากนั้นทำการตั้งสมมุติฐานโดยใช้วิธีการทางสถิติ t-test

$H_0 : \mu_1=\mu_2$

$H_1 : \mu_1>\mu_2$

โดยที่ μ1 คือปริมาณข้อมูลโดยใช้ IPSet μ_2 คือปริมาณข้อมูลโดยใช้ IPTable

และกำหนด a = 0.05 และคำนวณ t จากสูตร $t = \frac{\sum D}{\sqrt{\frac{(n\Sigma D^2-(\sum D)^2)}{n-1}}}$ = 42.26

และเมื่อได้ค่าวิกฤต จะได้ค่า 1.699 ซึ่งอยู่ในช่วงปฏิเสธ H_0

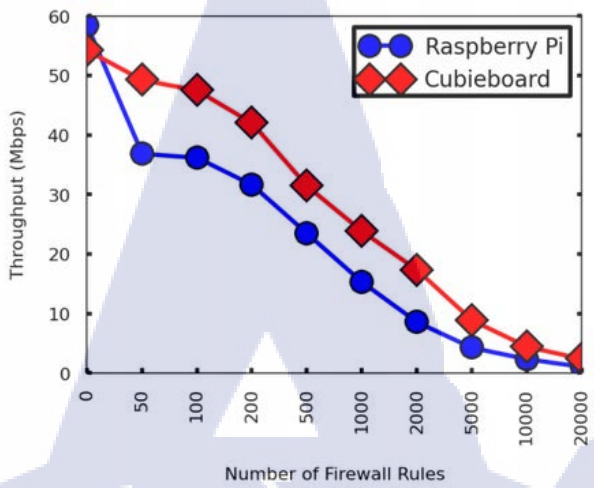
โดยจะสรุปโดยใช้ข้อมูลทางสถิติ (t-test) ว่าการใช้ ipset จะทำให้ไฟร์วอลล์มีความเร็วกว่าไฟร์วอล์ที่ใช้เฉพาะ IPTable อย่างมีนัยสำคัญที่ 0.05

<u>การเปรียบเทียบระหว่าง Raspberry Pi Firewall กับ Cubieboard Firewall</u> [14]

โดยในที่นี้จะวัดความสามารถในด้านต่างๆของการทำงานของไฟร์วอลล์อย่างเช่น การตั้งกฎไฟร์วอลล์ประสิทธิภาพในการทำงาน รวมถึง องค์ประกอบในด้านอื่นๆ อย่างเช่นการแปลงที่อยู่ของเครือข่าย (network address translation: NAT) หรือการ ดูข้อมูลที่เข้ามา เป็นต้น ซึ่งจะอธิบายเป็นส่วน ๆ ดังนี้

1. ความสามารถในการสร้างกฎไฟร์วอล์

ความสามารถในการสร้าง กฎไฟร์วอลล์ของ ราสพ์เบอรี พาย ไฟร์วอลล์มีปริมาณข้อมูลอยู่ที่ 58Mbps แตกต่างจาก cubieboard firewallที่ทำงานอยู่ที่ 54Mbps และเมื่อทำการเพิ่มจำนวนกฎเพิ่มขึ้น จำนวนทรูพุท (throughtput) cubieboard firewall จะสูงกว่าเมื่อเทียบกับ Raspberry Pi Firewall ซึ่งลักษณะความสามารถจะเป็นไปตามรูปที่ 2.8



รูปที่ 2.8 แสดงประสิทธิภาพของการสร้างกฎของไฟร์วอลล์ระหว่าง Raspberry Pi Firewall กับ Cubieboard Firewall [14]

2. คุณสมบัติ และความสามารถของการทำงานทั่วไป

คุณสมบัติภายในของระหว่าง Raspberry Pi Firewall มาเทียบกับ Cubieboard Firewallโดยที่ Raspberry Pi Firewall ทำงานด้วยหน่วยประมวลผล สถาปัตยกรรม ARM ความเร็ว 700MHz เก็บข้อมูลด้วย มัลติมีเดียการ์ด (multimedia card : MMC) ภายนอก หรือผ่าน เอสดีการ์ด (SD card) ซึ่งมีอินเตอร์เฟสการใช้งาน พอร์ทเอนกประสงค์(general-purpose input/output: GPIO) และพอร์ทสื่อสารข้อมูลแบบอนุกรม (universal asynchronous receiver/transmitter: UART) รวมอยู่ในซอฟต์แวร์ด้วย ซึ่งเป็นที่นิยมมากในการพัฒนาอุปกรณ์ด้วยระบบปฏิบัติการ ลีนุกซ์

ส่วน Raspberry Pi Firewall ทำงานด้วยสถาปัตยกรรม ARM เช่นเดียวกับราสพ์เบอรี พาย ไฟร์วอลล์ ความเร็ว 1 GHz และเก็บข้อมูลด้วยหน่วยจัดเก็บภายในขนาด 4GB NAND Flash และสามารถเพิ่มหน่วยความจำผ่าน เอสดีการ์ด (SD card) ได้ และอินเตอร์เฟสการใช้งานเพิ่มเติม อย่างเช่น ช่องเสียบยูเอสบี ช่องการสื่อสารแบบอนุกรมแบบซิงโครนัส (inter-integrated circuit: I2C) พอร์ทสถานะสัมพันธ์ (serial peripheral interface: SPI) และ พอร์ทช่องแรงดันไฟฟ้าที่แตกต่าง (low-voltage differential signaling: LVDS)

ในตารางที่ 2.2 เป็นการเปรียบเทียบคุณสมบัติระหว่าง Raspberry Pi Firewall เทียบกับ Cubieboard Firewall

ตารางที่ 2.2 คุณสมบัติการใช้งานระหว่าง ราสพ์เบอรี พาย ไฟร์วอลล์ กับ คิวบี้บอร์ด ไฟร์วอลล์ [14]

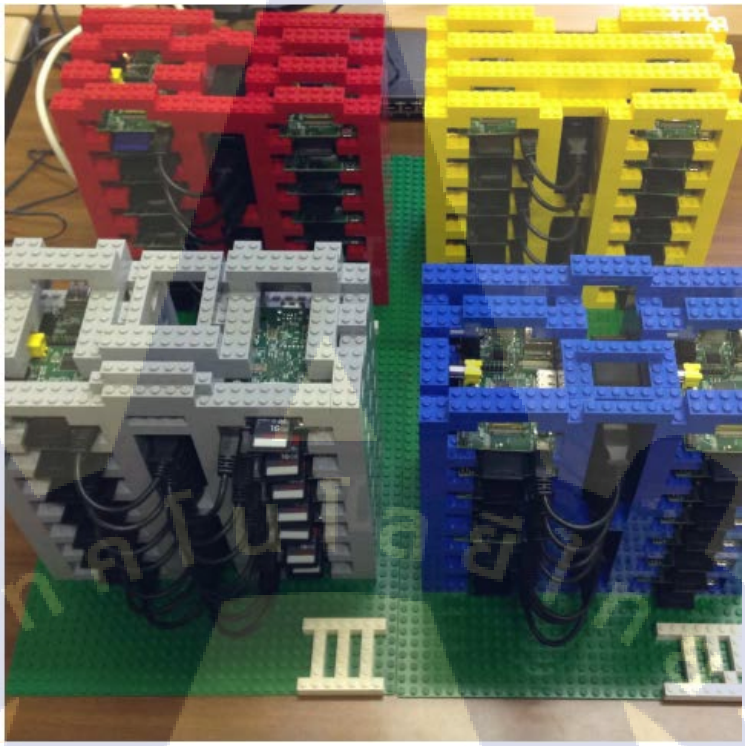
คุณสมบัติ	Raspberry Pi	Cubieboard
หน่วยประมวลผล(MHz)	700	1000
GPU	24	7.2
ความจำ	512	1024
Ethernet	10/100	10/100
ราคา ($)	35	59

การประยุกต์ใช้ Raspberry Pi มาใช้กับอุปกรณ์ทางด้านเครือข่าย

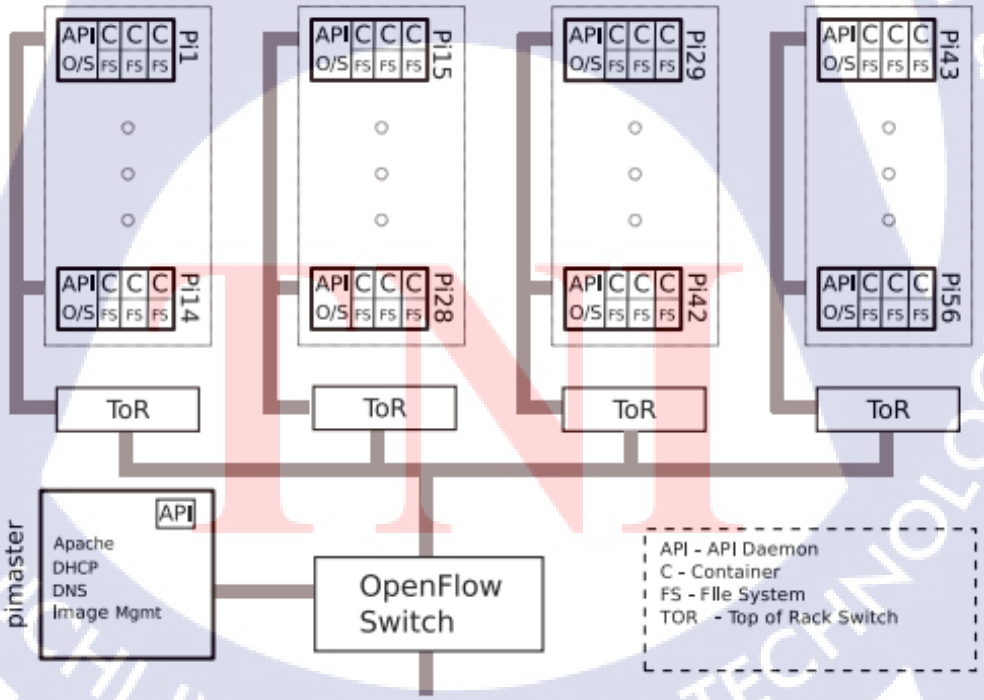
ในงานวิจัยนี้ได้นำเสนอ model ของ Cloud computing ที่สำหรับวางไว้ใน Data Center ขนาดใหญ่ [15] ซึ่งจะมีประโยชน์ในด้านของการใช้ hardware ที่มีราคาถูก และกำลังไฟที่น้อยลงด้วย รวมถึงประหยัดพื้นที่ในการใช้สอยเป็นจำนวนมาก

โดยที่นักวิจัยนั้นได้มองปัญหาว่า Cloud Data Center ทั่วไปนั้นประกอบไปด้วย server จำนวนมากมายมหาศาล ซึ่งมีราคาแพงมาก และไม่คุ้มค่ากับค่าใช้จ่ายเพื่อที่จะเอาไว้ใช้ในการศึกษาหรือทำงานวิจัย ซึ่งมีนักวิจัยบางกลุ่มได้ทำการจำลอง model สร้าง Cloud Data Center ในบางส่วนที่จำเป็นเท่านั้นแต่ก็ไม่ได้ผลเนื่องจากสาเหตุของคุณสมบัติของระบบการประมวลผลแบบกลุ่ม Cloud อาทิเช่น ปริมาณการส่ง traffic นั้น ไม่ได้คงที่เสมอไป และไม่สามารถทำนาย traffic ล่วงหน้าได้ เป็นต้น

โดยใน model นี้จะประกอบไปด้วยอุปกรณ์ Raspberry Pi Model B จำนวน 56 ตัว และประกอบต่อกัน เป็น Lego ตามรูปที่ 2.9 ซึ่งจะเป็นตัวแทนของ CloudData Center ทั้งหมด แต่มีขนาดที่เล็กกว่า ซึ่งจะมีหลักสถาปัตยกรรม ตามรูปที่ 2.10 ซึ่งจากสถาปัตยกรรมนั้น จะต่อเชื่อมโยงกันของแต่ละตัวในตึกเดียวกันและมีหัวตึก (Top of Rack) จะเป็นตัวจ่ายและกระจายข้อมูลหลักไป

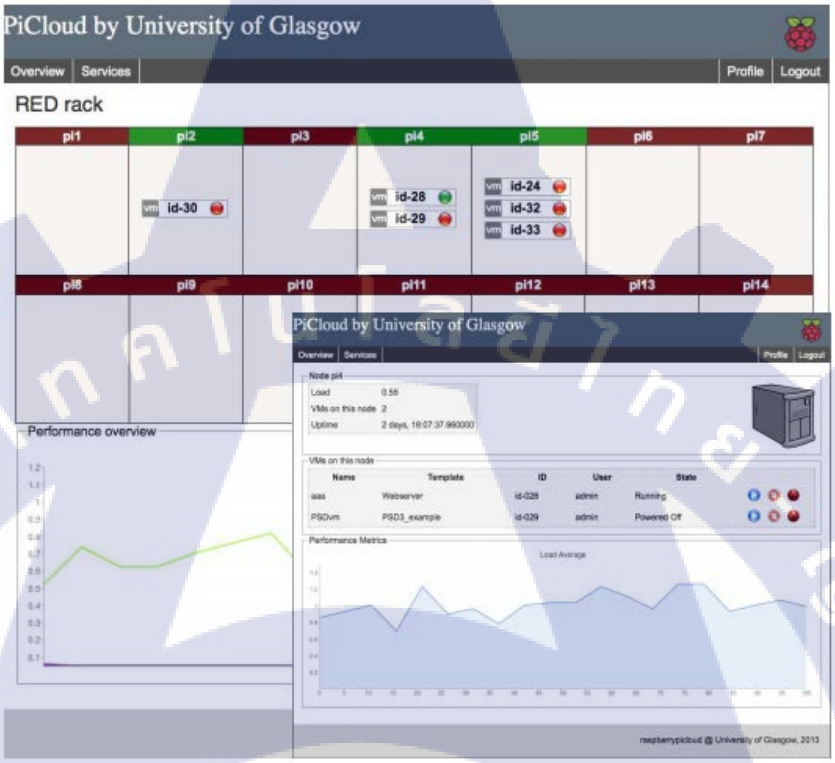


รูปที่ 2.9 อุปกรณ์ PiCloud [15]



รูปที่ 2.10 หน้าจอ สถาปัตยกรรมของการเชื่อมต่อ PiCloud [15]

โดยที่การทำงานของ PiCloud นั้น จะใช้สำหรับทำ web server หรือเก็บควบคุม data base เหมือนอุปกรณ์ web server ขนาดใหญ่รวมทั้งมี user interface การจัดการอุปกรณ์รวมอยู่ในตัวด้วย ดังรูปที่ 2.11



รูปที่ 2.11 หน้าจอ web interface การจัดการของ PiCloud [15]

ส่วนในมุมมองของราคา รวมถึงกำลังไฟฟ้าที่ใช้งานนั้น PiCloud ซึ่งผลิตมาจากบอร์ด Raspberry Pi จึงทำให้ตัวอุปกรณ์นั้น มีราคาถูก กินกำลังไฟน้อย รวมถึงไม่จำเป็นต้องใช้ระบบความเย็นเข้าช่วย ไม่เหมือนกัน server ที่อยู่ใน Datacenter ซึ่งต้องใช้ความเย็นที่เย็นพอสมควรเพื่อที่จะทำให้อุปกรณ์นั้น ทำงานได้อย่างมีประสิทธิภาพ รายละเอียดตามตารางที่ 2.3

ตารางที่ 2.3 ตารางแสดงประสิทธิภาพ ของ Testbed กับ PiCloud [15]

	Server	Power	Needs Cooling
Testbed	$112,000 (@$2,000)	10,800W/h	Yes
PiCloud	$1,960(@$35)	196W/h	No

บทที่ 3
วิธีดำเนินงานวิจัย

การศึกษาอุปกรณ์ บอร์ด Raspberry Pi

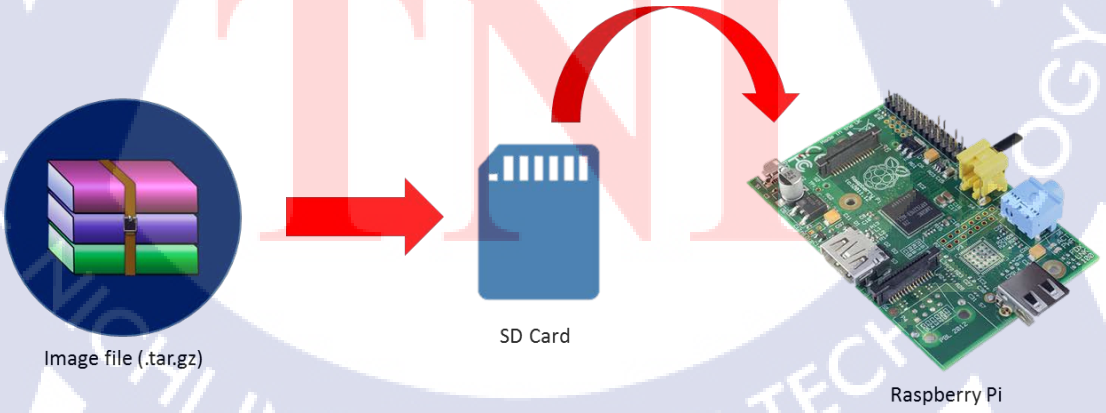
ในขั้นตอนดังกล่าวนี้จะทำการศึกษาถึงองค์ประกอบของบอร์ด Raspberry Pi รวมถึงทำการติดตั้ง Linux Kernel เข้าไปใน Raspberry Pi เพื่อเป็นขั้นตอนในการตั้งต้น อันไปสู่ขั้นตอนของการติดตั้งซอฟต์แวร์ลงในตัวอุปกรณ์ต่อไป

การติดตั้งซอฟต์แวร์ที่ใช้ในบอร์ด Raspberry Pi

ในขั้นตอนนี้เป็นการนำ ซอฟต์แวร์ในระบบปฏิบัติการลีนุกซ์ (Linux) มาติดตั้งในบอร์ด Raspberry Pi เพื่อเป็น software ที่ใช้ในการเปรียบเทียบประสิทธิภาพของ Raspberry Pi Firewall ประกอบไปด้วย IPTable IPFire PfSense IPCop ซึ่งเป็นซอฟต์แวร์ที่นำมาจัดการเกี่ยวกับการป้องกันข้อมูล และตรวจจับข้อมูลอินเตอร์เน็ตที่ผ่านเข้ามาในอุปกรณ์ซึ่งมีลักษณะคล้ายกับไฟร์วอลล์ที่มีอยู่ในท้องตลาดในปัจจุบันนี้

<u>วิธีการติดตั้ง software ลงใน raspberry pi</u>

การติดตั้ง software ลง raspberry pi นั้น จะทำการติดตั้ง image file ของซอฟต์แวร์แต่ละตัวลงใน SD Card และนำ SD Card นั้นใส่เข้าไปใน Slot SD Card ใน Raspberry Pi ตามรูปที่ 3.1 โดยที่จะทำการติดตั้ง 1 software ต่อ 1 SD Card เพื่อที่หลีกเลี่ยงภาระการทำงานภายใน และเพื่อที่จะให้ผลของประสิทธิภาพออกมาได้อย่างแม่นยำที่สุด

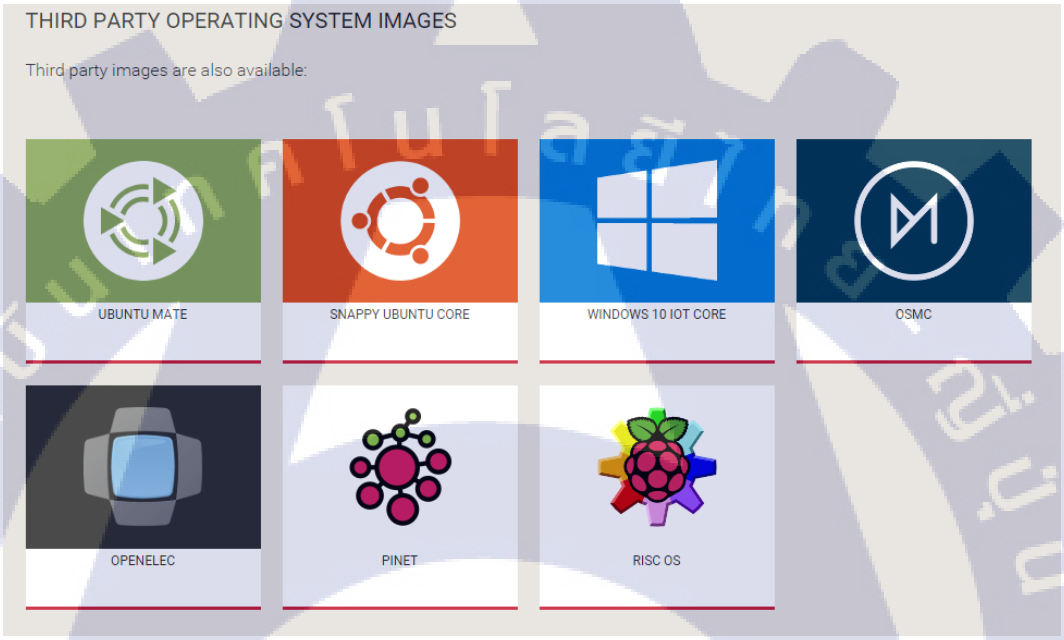


รูปที่ 3.1 แผนภาพขั้นตอนของการนำ Image File ใส่เข้าไปใน Raspberry Pi

วิธีการติดตั้ง image file และการตั้งค่าเบื้องต้นลงบน raspberry pi

1. IPTable (Netfilter)

Software นี้จะเป็น command ที่อยู่ในระบบปฏิบัติการ Linux โดยที่สามารถนำ image file ของระบบปฏิบัติการ Linux ได้ เช่น Ubuntu เป็นต้น ก็จะได้ IPTable รวมอยู่ด้วยทันที โดยสามารถ download image file ได้จาก https://www.raspberrypi.org/downloads/ ตามรูปที่ 3.2



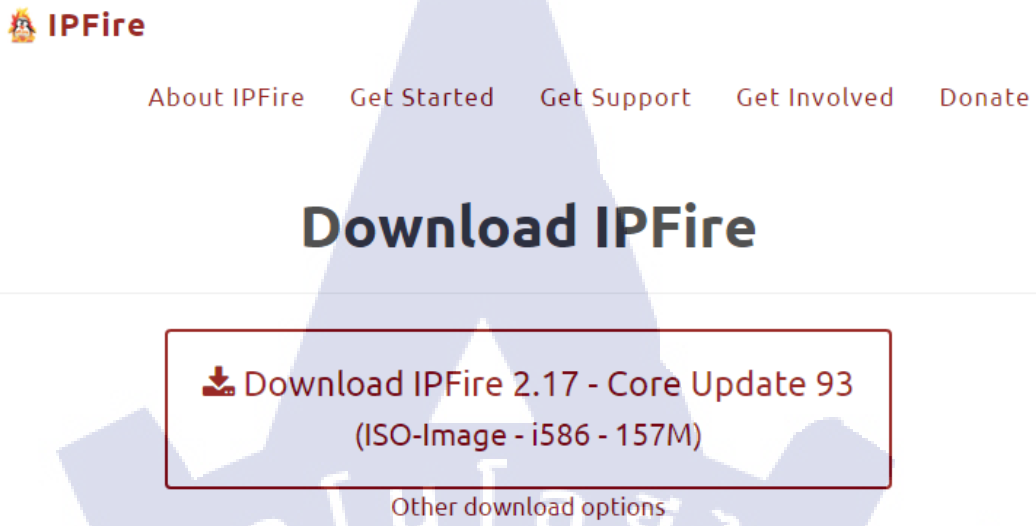
รูปที่ 3.2 หน้าจอ download ของ third party OS Image

หลังจาก download เสร็จสิ้นให้ทำการตั้งค่า IP โดยสามารถไปแก้รายละเอียดได้โดยใช้คำสั่ง nano/etc/network/interfaces และแก้รายละเอียดจากในไฟล์ดังกล่าวได้เลย

ทดสอบการเข้าถึงอินเตอร์เน็ตโดยเปิดเข้า web browser จาก server ถ้าสามารถเข้าถึงอนเตอ์เน็ตได้ แสดงว่าเป็นการเสร็จสิ้นการติดตั้ง และเข้าสู่ขั้นตอนของการทดสอบต่อไป

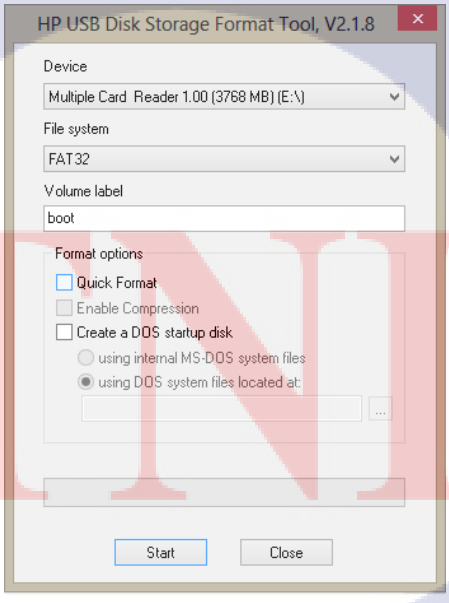
2. IPFire

2.1 สามารถ Download Image File ได้จาก http://www.ipfire.org/download แล้วเลือกที่ “Download IPFire 2.17 – Core Update 93” ตามรูปที่ 3.3 โดยที่สามารถลง image file ได้โดยที่ไม่ต้องมี OS มารองรับ



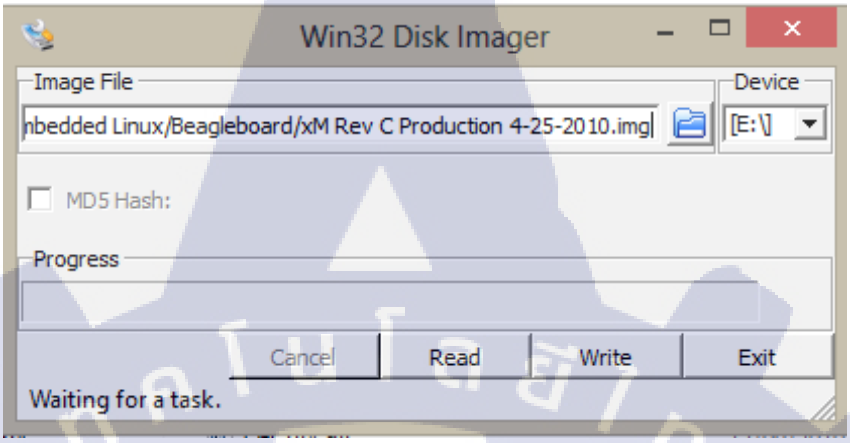
รูปที่ 3.3 รูปภาพแสดงหน้าจอ download ของ IPFire

2.2 เมื่อทำการ Download เรียบร้อยแล้ว ก็ทำการ format เครื่องโดยใช้โปรแกรม HP USB Disk Storage Format Tool (ต้องใช้ในฐานะ admin) แล้วกด start รอสักพัก จนกว่าการ format เสร็จสิ้น ตามรูปที่ 3.4



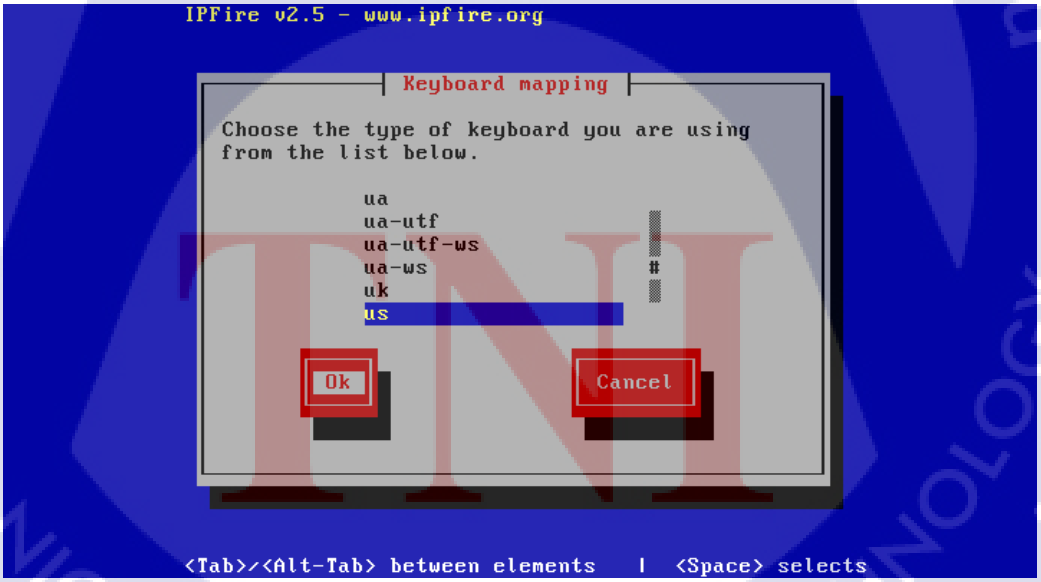
รูปที่ 3.4 รูปภาพแสดงหน้าจอ HP USB Disk Storage Format Tool

2.3 เมื่อทำการ format เสร็จเรียบร้อยแล้ว นำ image ที่ download มา mount เข้าไปใน SD Card โดยใช้โปรแกรม Win32 Disk Imager ตามรูปที่ 3.5

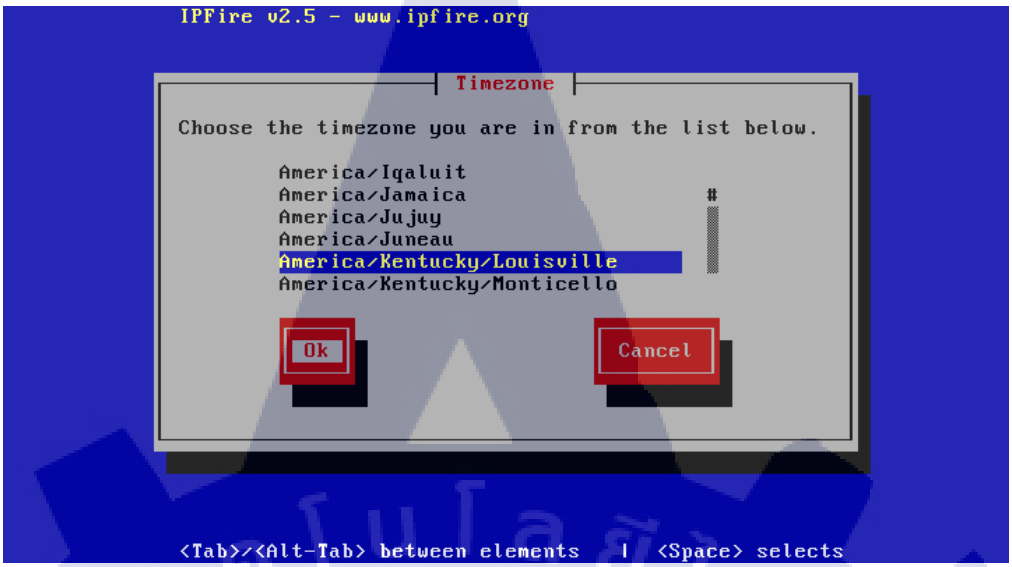


รูปที่ 3.5 รูปภาพแสดงหน้าจอ Win32 Disk Imager

2.4 ทำการเปิด Raspberry Pi เพื่อเข้าสู่การตั้งค่าเบื้องต้น โดยทำการเลือกภาษา keyboard และเลือก time zone ที่ต้องการ ตามรูปที่ 3.6 และ รูปที่ 3.7

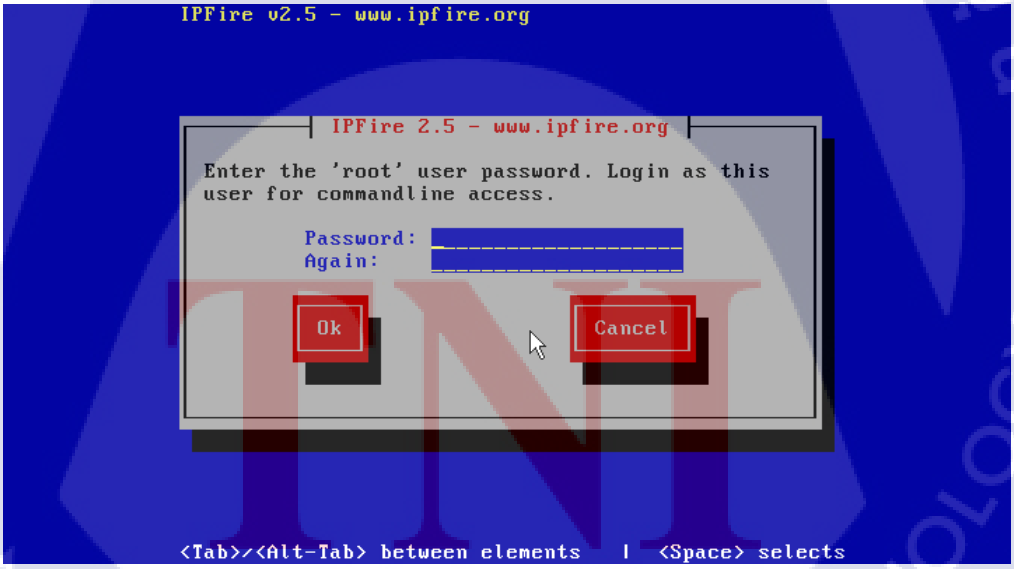


รูปที่ 3.6 รูปภาพแสดงการตั้งค่า keyboard



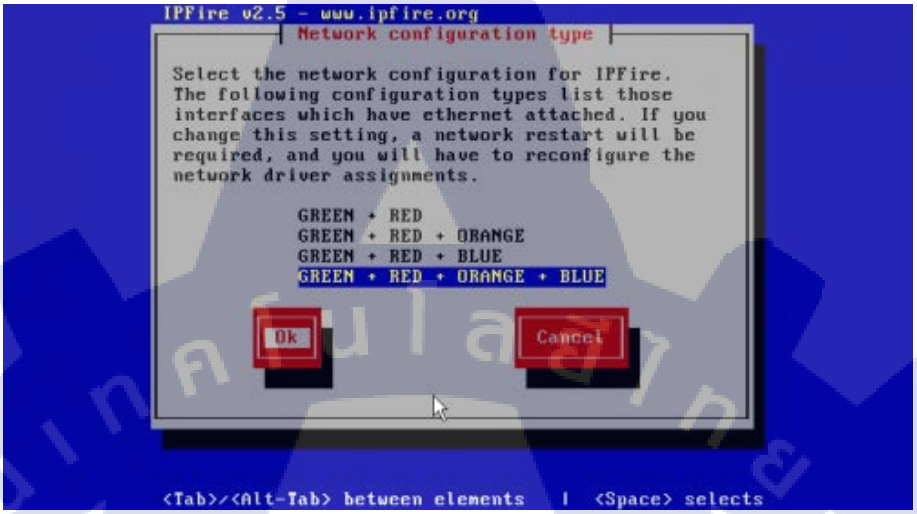
รูปที่ 3.7 รูปภาพแสดงการตั้งค่า time zone

2.5 ทำการตั้งค่าชื่อ hostname ของไฟร์วอลล์ และตั้งค่าชื่อผู้ใช้งานและรหัสผ่านในการเข้าถึงทั้งในรูปแบบ web interface และผ่าน ssh protocol ตามรูปภาพที่ 3.8



รูปที่ 3.8 รูปภาพแสดงการตั้งค่าชื่อผู้ใช้งานและรหัสผ่าน

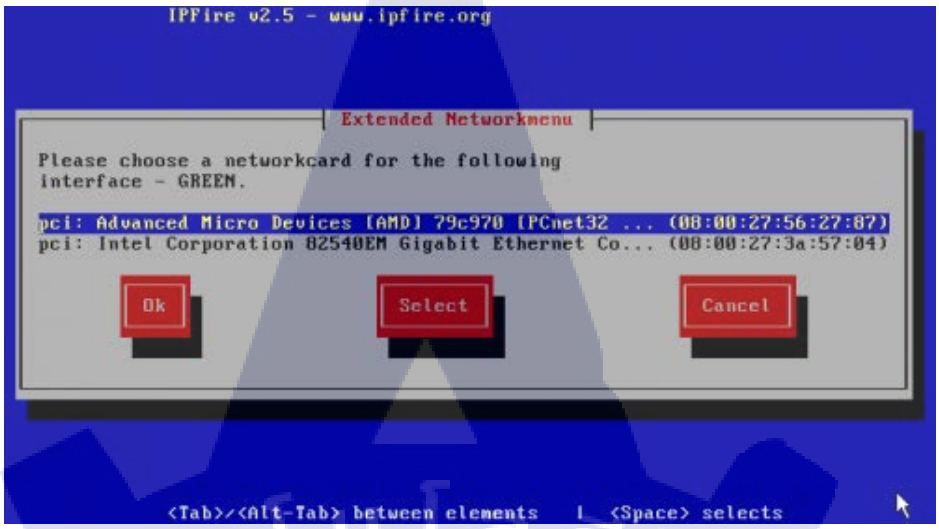
2.6 ขั้นตอนต่อไปเป็นการตั้งค่า Network Configuration Type สำหรับการใช้งานไฟร์วอลล์ โดยในงานวิจัยนี้จะทำการเลือกเป็น Green+Red หลังจากนั้นเลือก Ok ตามตัวอย่างในรูปที่ 3.9



รูปที่ 3.9 รูปภาพแสดงการตั้งค่า Network Configuration Type

2.7 ขั้นตอนต่อไปเป็นการจับคู่ interface เข้ากับ interface จริงที่มีอยู่ใน Raspberry Pi (Driver Card Assignment) ซึ่งในที่นี้จะใช้เป็น port ที่มาจาก USB to LAN ทั้งหมด ทำการจับคู่เข้ากับ interface ตามรูปที่ 3.10 และ 3.11

รูปที่ 3.10 รูปภาพแสดงการตั้งค่า Driver Card Assignment



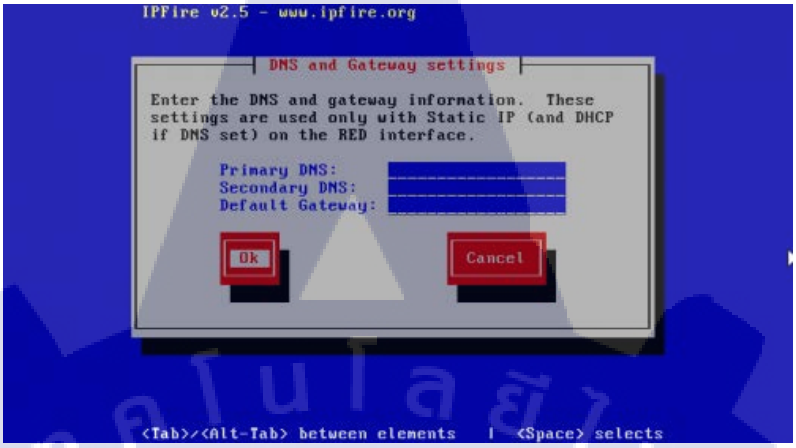
รูปที่ 3.11 รูปภาพแสดงการตั้งค่า Driver Card Assignment

2.8 ขั้นตอนต่อไปเป็นการกำหนด network address ให้กับแต่ละ interface โดยข้อกำหนดของแต่ละ interface ตามนิยามที่กล่าวไว้ในบทที่ 2 นั้น ซึ่ง IP Address ของแต่ละ interface (something is missing) และกำหนด IP Address ตามรูปที่ 3.12 โดยในแต่ละ interface นั้นมีลักษณะดังนี้

Red คือ Interface ที่เชื่อมต่อกับอินเตอร์เน็ต หรือ Network ภายนอก และ Green คือ Interface ที่เชื่อมต่อกับ Network ภายใน

รูปที่ 3.12 รูปภาพแสดงการตั้งค่า Network Address ของแต่ละ Interface

2.8 ขั้นตอนต่อไปเป็นการกำหนด DNS และ Gateway เพื่อใช้ในการเชื่อมต่อกับ network ภายนอก ตามรูปที่ 3.13



รูปที่ 3.13 รูปภาพแสดงการตั้งค่า DNS และ Gateway

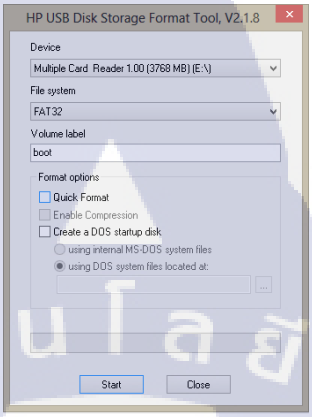
2.9 ทำการรอเพื่อให้ firewall ทำการอ่านค่าที่ได้จากการตั้งค่าเบื้องต้น และทำการทดสอบการใช้งานโดยเข้าไปที่ IP ที่ทำการตั้งค่าจากขั้นตอนของการตั้งค่า network interface โดยลงท้ายด้วย port 444 ตัวอย่างเช่น https://192.168.4.2:444 เป็นต้น และทำการทดสอบการเข้าถึงอินเตอร์เน็ตโดยการเปิด web browser จาก server เป็นการเสร็จสิ้นการติดตั้ง พร้อมที่จะเข้ากระบวนการการทดสอบประสิทธิภาพต่อไป

3. OpenWRT

3.1 ดาวน์โหลด Image File ได้จาก http://downloads.openwrt.org/ ในงานวิจัยนี้ทำการเลือกเวอร์ชั่น 17.01.4 ตามรูปที่ 3.14 แล้วเลือก brcm2708 และเลือก subtarget bcm2708

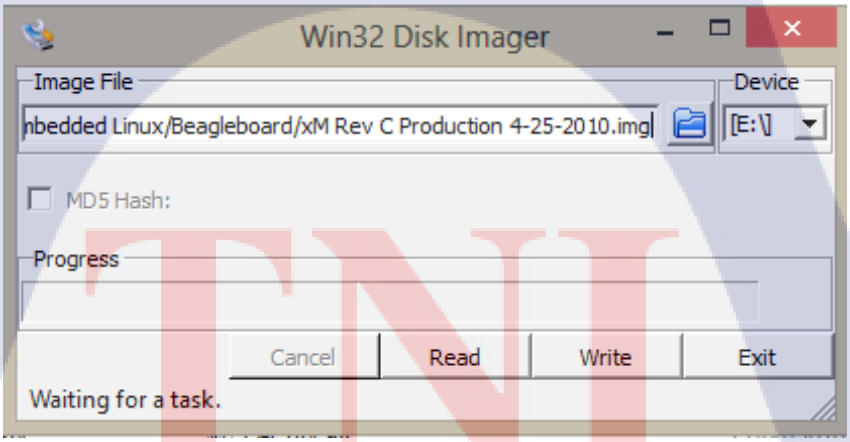
รูปที่ 3.14 รูปภาพหน้าดาวน์โหลดของ OpenWRT

3.2 เมื่อทำการ Download เรียบร้อยแล้ว ก็ทำการ format เครื่องโดยใช้โปรแกรม HP USB Disk Storage Format Tool (ต้องใช้ในฐานะ admin) แล้วกด start รอสักพัก จนกว่าการ format เสร็จสิ้น ตามรูปที่ 3.15



รูปที่ 3.15 รูปภาพแสดงหน้าจอ HP USB Disk Storage Format Tool

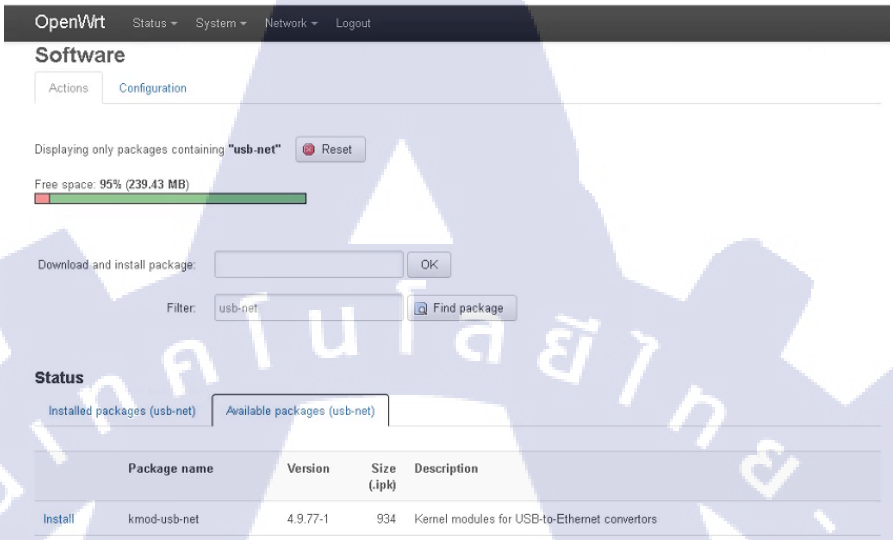
3.3 เมื่อทำการ format เสร็จเรียบร้อยแล้ว นำ image ที่ download มา mount เข้าไปใน SD Card โดยใช้โปรแกรม Win32 Disk Imager ตามรูปที่ 3.16



รูปที่ 3.16 รูปภาพแสดงหน้าจอ Win32 Disk Imager

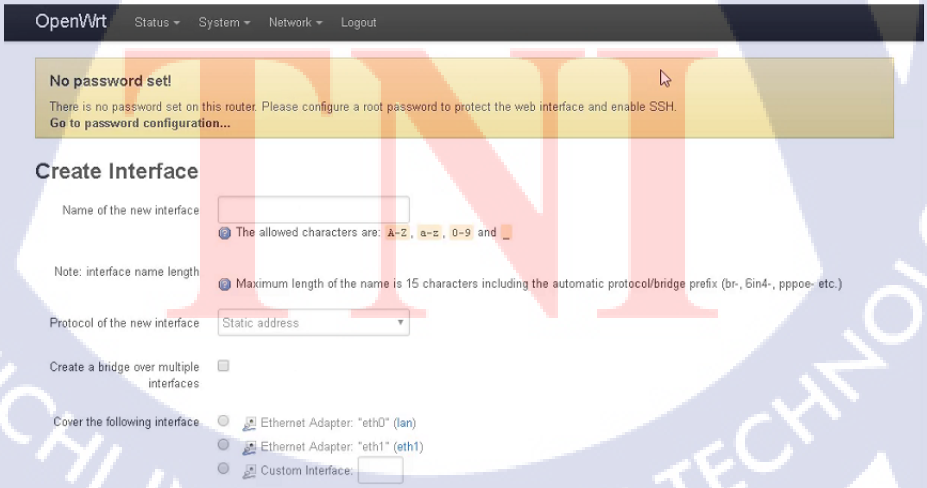
3.4 หลังจากนั้นนำ SD Card ดังกล่าวไปใส่ใน Raspberry Pi เพื่อทำการเปิดการใช้งานหน้าเว็บอินเตอร์เฟส (Web Interface) โดยนำสาย LAN ไปเสียบเข้ากับพอร์ท Ethernet แล้วเข้าไปที่ 192.168.1.1 จากหน้าเว็บเบราวเซอร์ (Web Browser)

3.5 ทำการติดตั้ง Package File ที่ชื่อว่า kmod-usb-net จากเมนู Sysyem > Software แล้ว ใส่คำว่า usb-net จากช่อง filter ตามรูปที่ 3.17 ทั้งนี้เพื่อให้ Openwrt ได้รู้จักกับอินเตอร์เฟสที่เป็น USB



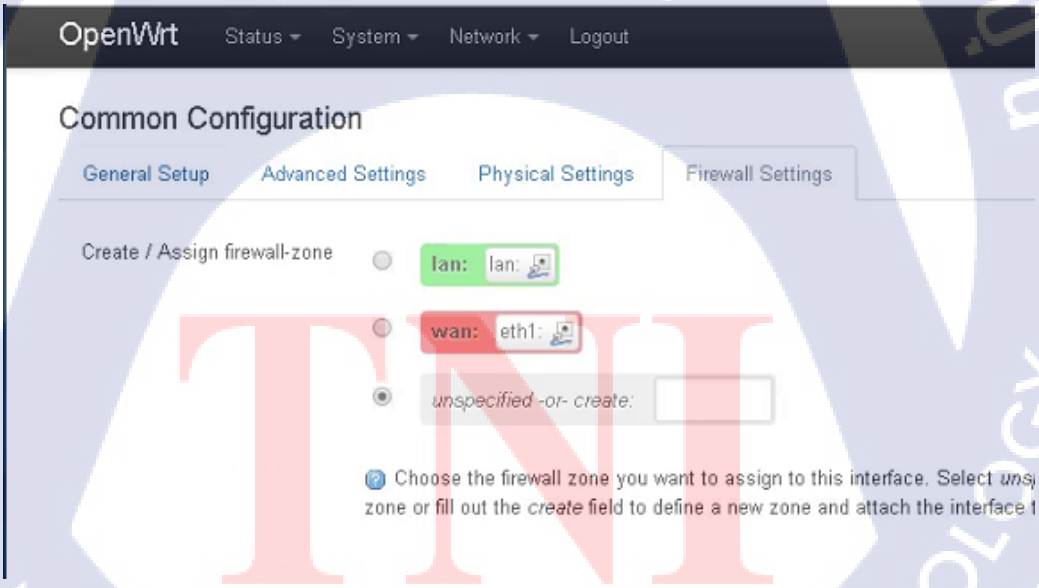
รูปที่ 3.17 รูปภาพแสดงหน้าจอ ของ Software ในหน้า web interface ของ Openwrt

3.6 จากนั้นจะแสดง port Ethernet ที่ใช้งานทั้งหมดทั้งที่ฝังอยู่ใน Raspberry Pi และ USB จากเมนู Network > Interfaces และทำการตั้งค่า IP Address, Gateway และ Zone ใช้งาน ตามรูปที่ 3.18 รูปที่ 3.19 และรูปที่ 3.20 เป็นการเสร็จสิ้นการตั้งค่า



รูปที่ 3.18 รูปภาพแสดงเมนูการตั้งค่า Interface ในการสร้างอินเตอร์เฟส

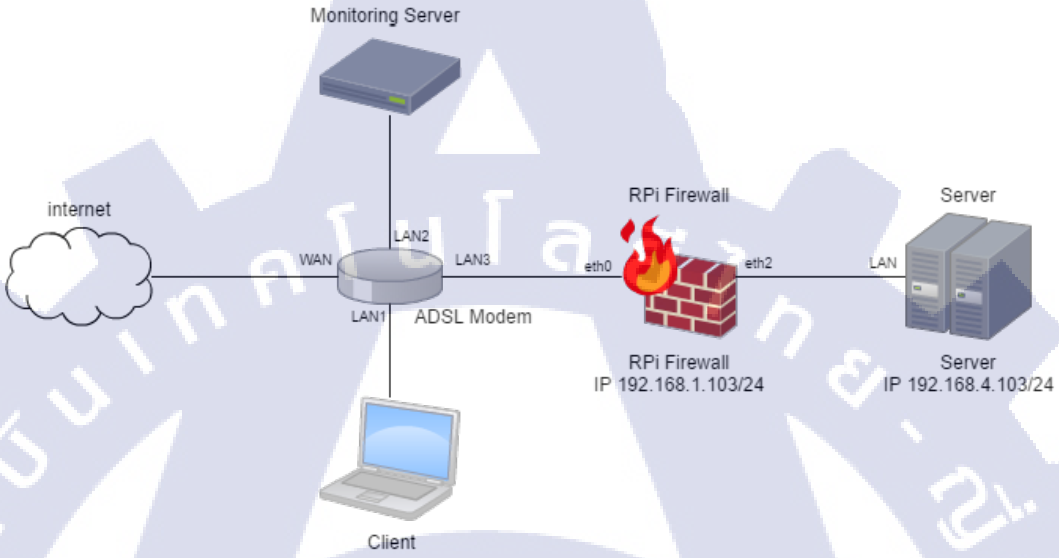
รูปที่ 3.19 รูปภาพแสดงเมนูการตั้งค่า Interface ในการตั้งค่า IP Address



รูปที่ 3.20 รูปภาพแสดงเมนูการตั้งค่า Interface ในการตั้งค่า Zone ของอินเตอร์เฟส

การทดสอบประสิทธิภาพของไฟร์วอลล์

ขั้นตอนดังกล่าวจะเป็นขั้นตอนของการเปรียบเทียบประสิทธิภาพของไฟร์วอลล์ในแต่ละ software ที่กล่าวในบทที่ 1 โดยในที่นี่จะกล่าวถึง แผนภาพเครือข่าย (Network Diagram) ของงานวิจัยซึ่งจะใช้ในการทดสอบประสิทธิภาพนี้ซึ่งเป็นไปตามรูปที่ 3.21



รูปที่ 3.21 Network Diagram ของการทดสอบประสิทธิภาพไฟร์วอลล์

<u>อุปกรณ์ที่จะใช้วัดประสิทธิภาพนั้น ประกอบไปด้วยดังนี้</u>

1. Client ในที่นี้ได้ทำการจำลองเป็น traffic simulator เป็นการจำลองเป็นข้อมูลที่รับเข้ามา (Throughput) เพื่อทำการทดสอบประสิทธิภาพของ Raspberry Pi Firewall ซึ่งในแต่ละการทดลองจะทำการจำลองข้อมูล packet ที่แตกต่างกันไปในแต่ละการทดลอง ซึ่งจะกล่าวในหัวข้อถัดไป

2. Monitoring Server ในที่นี้จะใช้ Open Source ที่ชื่อว่า PRTG ในการตรวจสอบและตรวจจับปริมาณข้อมูล traffic ออกมาในรูปแบบกราฟ สำหรับการทดสอบปริมาณข้อมูลที่รับและออกของแต่ละ open source firewall

3. PRTG เป็นเครื่องมือสำหรับตรวจสอบ และเก็บข้อมูลต่างๆ แล้วแสดงผลออกมาในรูปแบบกราฟ ซึ่งในที่นี่ PRTG มีการรองรับโพรโตคอล SNMP สำหรับ การสร้างกราฟการใช้งานทราฟฟิคการใช้งาน CPU และอื่นๆ ซึ่งสอดคล้องกับข้อมูลสำหรับการประมวลผลและวิเคราะห์ผลในงานวิจัยนี้

4. Firewall ในที่นี้จะใช้ Raspberry Pi ในการทดสอบประสิทธิภาพ และทำการติดตั้ง Software ดังนี้

4.1 IPTable

4.2 IPFire

4.3 OpenWRT

โดยที่จะแยกการติดตั้ง 1 Software / 1 SD Card เพื่อที่จะได้ประสิทธิภาพออกมาได้แม่นยำที่สุด และไม่สร้างภาระ Load ของการใช้งานหลาย process ใน OS จนเกินไป

ซึ่งในแต่ละ Software นั้นจะต้องมีการจำลองเพื่อทดสอบประสิทธิภาพ โดยการตั้งกฎของไฟร์วอลล์ขึ้นมารอเอาไว้ เพื่อที่จะทดสอบภาระของการทำงานซึ่งจะกล่าวในข้อต่อไปในส่วนการเปรียบทียบวิเคราะห์ผล และประสิทธิภาพ

1. Server ในที่นี้จะเป็นอุปกรณ์สำหรับการรับส่งข้อมูลเพื่อทำการประมวลผลลัพธ์เมื่อข้อมูลแพ็คเกตนั้นผ่าน Raspberry Pi Firewall มาแล้ว อุปกรณ์ที่ใช้จะเป็น Android Box ซึ่งเหมาะสมกับการใช้งานในทุกแอพพลิเคชั่น ในการใช้งานที่หลากหลาย รวมถึงราคาย่อมเยาว์

ซึ่งในการทดสอบประสิทธิภาพของไฟร์วอลล์ของงานวิจัยนี้ จะทำการทดสอบประสิทธิภาพในด้านต่าง ๆ โดยทำการทดสอบประสิทธิภาพแบบแยกเป็นแอพพลิเคชั่น และ การรวมการใช้งานแอพพลิเคชั่นมากกว่า 1 อย่างเข้าด้วยกัน รายละเอียดของการทดสอบมีดังต่อไปนี้

<u>ชุดที่ 1 การทดสอบประสิทธิภาพการทำงานแบบแยกทีละประเภทของแอพพลิเคชั่น (Application)</u>

การทดสอบในชุดนี้เป็นการทดสอบการใช้งานโดยแยกเป็นแอพพลิเคชั่น โดยในที่นี้จะแยกเป็นการใช้งานทั้งการรับส่งข้อมูลด้วย FTP, การใช้งานการเข้าถึงเว็บไซต์ด้วย HTTP และ HTTPS, การใช้งาน e-mail และการใช้งานผ่านไลฟ์สตรีมมิ่ง (live streaming)

1. การทดสอบประสิทธิภาพการรับ-ส่ง ข้อมูลด้วย โพรโตคอล FTP โดยในที่นี้จะทำการจำลอง Server เป็น FTP Server เพื่อใช้เป็นศูนย์กลางในการรับ-ส่งข้อมูล โดยจะทำการเปิดพอร์ตที่จำเป็นสำหรับการใช้งานนี้ โดยจะทำการติดตั้ง FTP Server ลงใน Server ปลายทาง และทำการเปิด port สำหรับการส่ง FTP เท่านั้น ในที่นี้จะทำการ forward port 2221 สำหรับการใช้งาน FTP และ port 2300-2399 เป็น port passive ในการส่งข้อมูล

ตารางที่ 3.1 Policy Rule สำหรับการทดสอบของการทดลอง FTP

IP ต้นทาง	IP ปลายทาง	Service/Port	Action
Any	192.168.1.103 nat 192.168.4.103	2221	NAT
Any	192.168.1.103 nat 192.168.4.103	2300-2399	NAT

<u>โดยวิธีการวัด และทดสอบประสิทธิภาพ จะทำการทดสอบตามรายละเอียดดังต่อไปนี้</u>

1. ทำการส่งไฟล์ขนาด 2 GB ด้วยโพรโตคอล FTP เพื่อทดสอบความสามารถในการส่งไฟล์

2. นำไฟล์ 2GB จากข้อที่ 1 แบ่งเป็น 3 ไฟล์ย่อย (ขนาดไฟล์ประมาณ 900 MB) และทำการส่งพร้อมกันเพื่อทดสอบการทำงานของ FTP ในการส่งมากกว่า 1 session

2.1 การทดสอบประสิทธิภาพในการเข้าถึงข้อมูลเว็บไซต์ด้วย โพรโตคอล HTTP และ HTTPS โดยจะใช้ โพรโตคอลร่วมกันระหว่าง HTTP , HTTPS และ DNS ในการเข้าถึงข้อมูลเว็บไซต์ รวมถึงการดาวน์โหลดข้อมูลโดยใช้โพรโตคอลดังกล่าว

เว็บไซต์ที่ทำการทดสอบนั้นเป็นการรวมรวมสถิติจากเว็บไซต์ alexa.com ซึ่งเว็บไซต์ ที่นำมาทดสอบนั้นเป็นเว็บไซต์ที่มีความนิยมสูงที่สุด และมีขนาด content ในหน้าเว็บไซต์มากเพียงพอ โดยเว็บไซต์ที่นำมาทดสอบนั้นมีดังต่อไปนี้

- youtube.com
- google.com
- pantip.com
- lazada.co.th
- movie2free.com
- sanook.com
- yahoo.com
- kapook.com
- dek-d.com
- mthai.com

โดยการทดสอบนี้จะแบ่งเป็นสถานการณ์ย่อยดังนี้

1. การทดสอบการเข้าเว็บไซต์โดยไม่มีการปิดกั้นใดๆ ทั้งสิ้น

ในที่นี้จะทำการทดสอบการเข้าเว็บไซต์ทั่วไป โดยที่เปิดพอร์ตสำหรับ โพรโตคอล DNS, HTTP, HTTPS สำหรับการเข้าถึงเว็บไซต์ และทำการทดสอบการเข้าเว็บไซต์โดยใช้ Server เป็นเครื่องสำหรับทดสอบ

ตารางที่ 3.2 Policy Rule สำหรับการทดสอบการเข้าถึงเว็บไซต์แบบไม่มีการปิดกั้น

IP ต้นทาง	IP ปลายทาง	Service/Port	Action
192.168.4.103	Any	53	Accept
192.168.4.103	Any	80	Accept
192.168.4.103	Any	443	Accept

หลังจากนั้นทำการสรุปและวัดผลในระหว่างการเข้าเว็บไซต์

2. การทดสอบการเข้าเว็บไซต์โดยมีเงื่อนไขในการปิดกั้นเว็บไซต์บางส่วน

ในที่นี้จะทำการทดสอบการเข้าเว็บไซต์ทั่วไป โดยที่เปิดพอร์ตสำหรับ โพรโตคอล DNS, HTTP, HTTPS และรวมถึงสร้าง rule สำหรับการปิดกั้นเว็บไซต์บางส่วน สำหรับการเข้าถึงเว็บไซต์และทำการทดสอบการเข้าเว็บไซต์โดยใช้ Server เป็นเครื่องสำหรับทดสอบ

ตารางที่ 3.3 Policy Rule สำหรับการทดสอบการเข้าถึงเว็บไซต์แบบมีการปิดกั้นเว็บไซต์บางส่วน

IP ต้นทาง	IP ปลายทาง	Service/Port	Action
192.168.4.103	Any	53	Accept
192.168.4.103	Any	80	Accept
192.168.4.103	Any	443	Accept

หลังจากนั้นทำการสรุปและวัดผลในระหว่างการเข้าเว็บไซต์และทำการเปรียบเทียบการทำงานกับการทดสอบการเข้าถึงเว็บไซต์ทั้ง 2 กรณี

- การทดสอบประสิทธิภาพในการรับและส่งอีเมล์ (e-mail) โดยในที่นี้จะทำการจำลองพฤติกรรมการรับส่งอีเมล์ไปยังปลายทาง โดยใช้โปรแกรมจำลอง traffic ที่ชื่อว่า iperf และทำการปิดกั้นโดยใช้กฎของไฟร์วอลล์ โดยทำการตั้งเงื่อนไขการส่ง email จำนวน 10MB ทุกๆ 1 วินาที ซึ่งการส่งอีเมล์นั้นจะทำงานด้วย โพรโตคอล SMTP สำหรับการส่งเมล์

โดยการทดสอบนี้จะทำการส่งอีเมล์ไปยังปลายทางโดยใช้โพรโตคอล SMTP อ้างอิงตาม rule ดังนี้

ตารางที่ 3.4 Policy Rule สำหรับการทดสอบการใช้งานรับและส่งอีเมล์

IP ต้นทาง	IP ปลายทาง	Service/Port	Action
192.168.4.103	Any	25	Accept

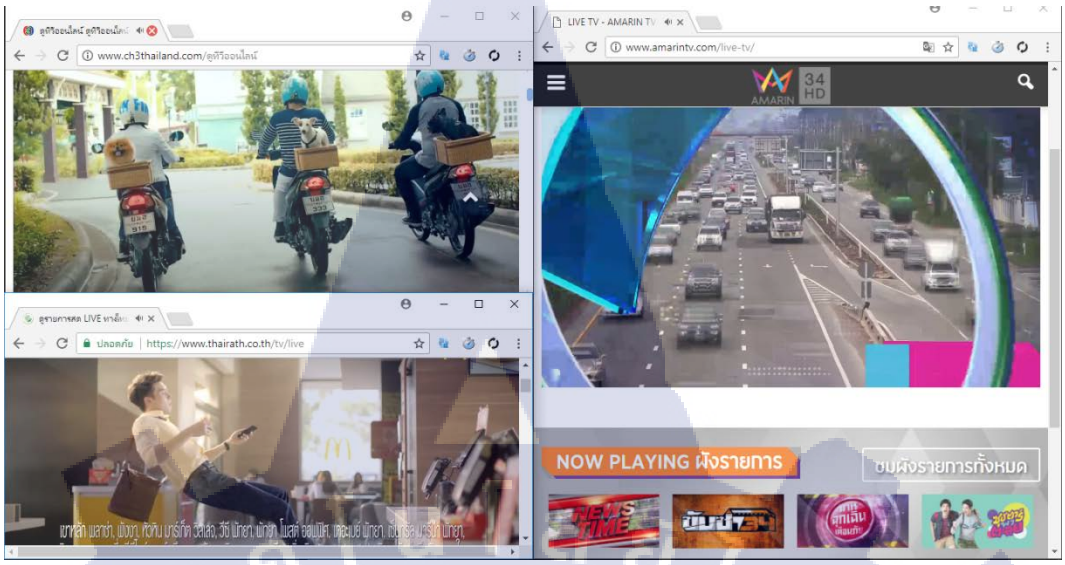
หลังจากนั้นจะทำการสรุปและวัดผลหลังจากการทดลองดังกล่าว

- การทดสอบประสิทธิภาพการใช้งาน Live Streaming สำหรับการใช้งานแอพลิเคชั่นในช่วงระยะเวลาหนึ่งแบบต่อเนื่อง โดยในที่นี้จะทำการจำลองโดยการเปิดสตรีมมิ่งรายการโทรทัศน์ของช่องไทยรัฐทีวีผ่านเว็บไซต์ www.thairath.co.th/tv/live, http://www.ch3thailand.com และ http://www.amarintv.com/live-tv/ และปรับคุณภาพของภาพเป็น 720p (1280x720p)

ตารางที่ 3.5 Policy Rule สำหรับการทดสอบการใช้งาน Live Streaming

IP ต้นทาง	IP ปลายทาง	Service/Port	Action
192.168.4.103	Any	Special port TCP	Accept
192.168.4.103	Any	Special port UDP	Accept

จากนั้นทำการเปิดการทำงานพร้อมกันในหน้าจอเดียวกัน ลักษณะเดียวกันกับรูปภาพที่ 3.22 ทดสอบการเข้าใช้งานดังกล่าวเป็นเวลา 30 นาที เพื่อให้ผลลัพธ์ที่ออกมาเป็นไปอย่างมีประสิทธิภาพและง่ายต่อการเปรียบเทียบ โดยในที่นี้จะทำการทดสอบทั้งการเปิดการใช้งาน streaming ครบทั้ง 3 ช่อง และทำการ ปิดกั้นการใช้งานการ streaming ช่อง 3 (http://www.ch3thailand.com) จากนั้นทำการเปรียบเทียบประสิทธิภาพระหว่างกราฟของแต่ละ open source firewall ในการทดลองย่อยต่างๆ เพื่อสรุปความแตกต่าง



รูปที่ 3.22 ลักษณะการทำงานของการทดสอบ Streaming

<u>ชุดที่ 2 การทดสอบประสิทธิภาพการทำงานพร้อมกันของแอพพลิเคชั่นมากกว่า 1 ประเภท</u>

โดยในชุดนี้เป็นการทดสอบการทำงานแอพพลิเคชั่นมากกว่า 1 ประเภท เพื่อทดสอบความสามารถของ Raspberry pi firewall ในการใช้งานหลากหลาย โดยทำการรวมแอพพลิเคชั่นที่กล่าวไว้ในชุดที่ 1 เพื่อทดสอบขีดความสามารถของไฟร์วอลล์ในการทำงานพร้อมกันโดยมีการทดลองดังต่อไปนี้

1. การทดสอบการใช้งานเว็บไซต์ อีเมล์ และ Live Streaming จากสถิติของ Digital in 2018 ไตรมาสที่ 1 ได้กล่าวถึงลักษณะการใช้งานอินเตอร์เน็ตของประชากรไทย พบว่าแอพพลิเคชั่นที่ใช้งานมากที่สุดจากสถิตินั่นก็คือ การเข้าใช้งานเว็บไซต์ อีเมล์ และ Live Streaming ที่เพิ่งเข้ามาใหม่ โดยในที่นี้จะทำการจำลองเพื่อให้สอดคล้องกับสถิติดังกล่าว เพื่อทดสอบประสิทธิภาพของ Raspberry Pi Firewall โดยนำ rule จากการทดสอบการใช้งานเว็บไซต์แบบมีการปิดกั้นเว็บไซต์บางส่วน การทดสอบการส่ง email และการทดสอบ live streaming แบบมีการปิดกั้นช่องทีวี มารวมกัน และใช้งานพร้อมกัน และทำการสรุปผลในแต่ละ open source firewall

2. การทดสอบประสิทธิภาพการทำงานแบบแอพลิเคชั่นทุกอย่างเข้าด้วยกัน โดยนำ policy ของการทดลองที่ 2 ในแต่ละแอพพลิเคชั่น ได้แก่ การใช้งาน FTP แบบแบ่งไฟล์ 3 ไฟล์ทำการส่งพร้อมกัน การใช้งานเว็บไซต์แบบมีการปิดกั้นเว็บไซต์บางส่วน การทดสอบการส่ง email และการทดสอบ live streaming แบบมีการปิดกั้นช่องทีวี มารวมกันและใช้งานพร้อมกัน จากนั้นทำการสรุปผล ซึ่งในการทดสอบนี้เป็นการจำลองการใช้งานตามพฤติกรรมของผู้ใช้งานทั่วไป และทดสอบขีดจำกัดของ open source firewall

การเปรียบเทียบวิเคราะห์และสรุปผลประสิทธิภาพ

โดยรายละเอียดประสิทธิภาพที่นำมาทำการวิเคราะห์นั้นมีรายละเอียดดังนี้

1. การใช้งานในหน่วยประมวลผลซีพียู (CPU Processing usage) เป็นการทดสอบความสามารถในการประมวลผลซีพียูของ Raspberry Pi Firewall หน่วยที่ได้จะเป็นเปอร์เซนต์

2. แบนด์วิธที่ใช้งานในแต่ละอินเตอร์เฟส (Bandwidth Interface usage) เป็นการทดสอบประสิทธิภาพในการส่งข้อมูลในแต่ละการทดสอบทั้งขาเข้า และขาออกหน่วยที่ได้จะเป็น kilobits per second (kbps)

3. ค่าเฉลี่ยภาระงานของหน่วยประมวลผล (Load Average) เป็นการทดสอบค่าภาระงานที่ใช้งานในหน่วยประมวลผลในการทำงานแต่ละช่วงเวลา ซึ่งในระบบปฏิบัติการ linux จะมีค่านี้ปรากฏออกมาจากคำสั่ง top เพื่อให้เห็นแนวโน้มการทำงานของหน่วยประมวลผล และอัตรา task ที่ค้างเหลืออยู่ในช่วงเวลานั้นๆ ค่านี้จะไม่มีหน่วย แต่ค่าที่ได้จะต้องมีค่าไม่เกินจำนวน CPU ที่มีอยู่ โดยในงานวิจัยนี้ Raspberry Pi Firewall มี CPU เพียง 1 core ดังนั้นค่าเฉลี่ยภาระงานควรมีค่าไม่เกิน 1 ถึงจะทำงานได้เต็มประสิทธิภาพ ถ้าค่าที่ได้เกิน 1 นั่นหมายความว่ายังมี process อื่นที่ยังรอทำงานอยู่ทำให้การทำงานเป็นไปอย่างล่าช้า ซึ่งในแต่ละการทดลองแต่ละครั้งเป็นการทดลองที่ใช้เวลาในการทดสอบพอสมควร ในงานวิจัยนี้จึงแสดงผลกราฟที่เป็นค่าเฉลี่ยภาระงานของหน่วยประมวลผลที่เป็น 15 นาทีย้อนหลัง ตัวอย่างเช่น กราฟในจุดนาทีที่ 15 จะนำค่าเฉลี่ยของ ค่า Load Average ในนาทีที่ 0-14 มาทำการเฉลี่ยและได้ค่าของนาทีที่ 15 เป็นต้น

เมื่อได้ข้อมูลมาครบทั้ง 3 Open Source Firewall แล้ว ทำการวิเคราะห์และสรุปผล และแสดงผลในรูปแบบด้วยตารางและกราฟ รวมทั้ง สรุปผลข้อดีข้อเสีย เพื่อที่จะหา Open Source Firewall ที่เหมาะสมในการติดตั้งใน Raspberry Pi

บทที่ 4
ผลการทดลอง

จากวิธีการทดลองที่กล่าวมาในบทที่ 3 ในที่นี้จะแสดงผลการทดลองของการทดสอบประสิทธิภาพ ทั้งในรูปแบบการทดลองแบบแยกทีละประเภทของ application การทดสอบการทำงานพร้อมกันของ Application มากกว่า 1 ประเภท โดยแจกแจงรายละเอียดในแต่ละการทดลอง โดยมีรายละเอียดดังนี้

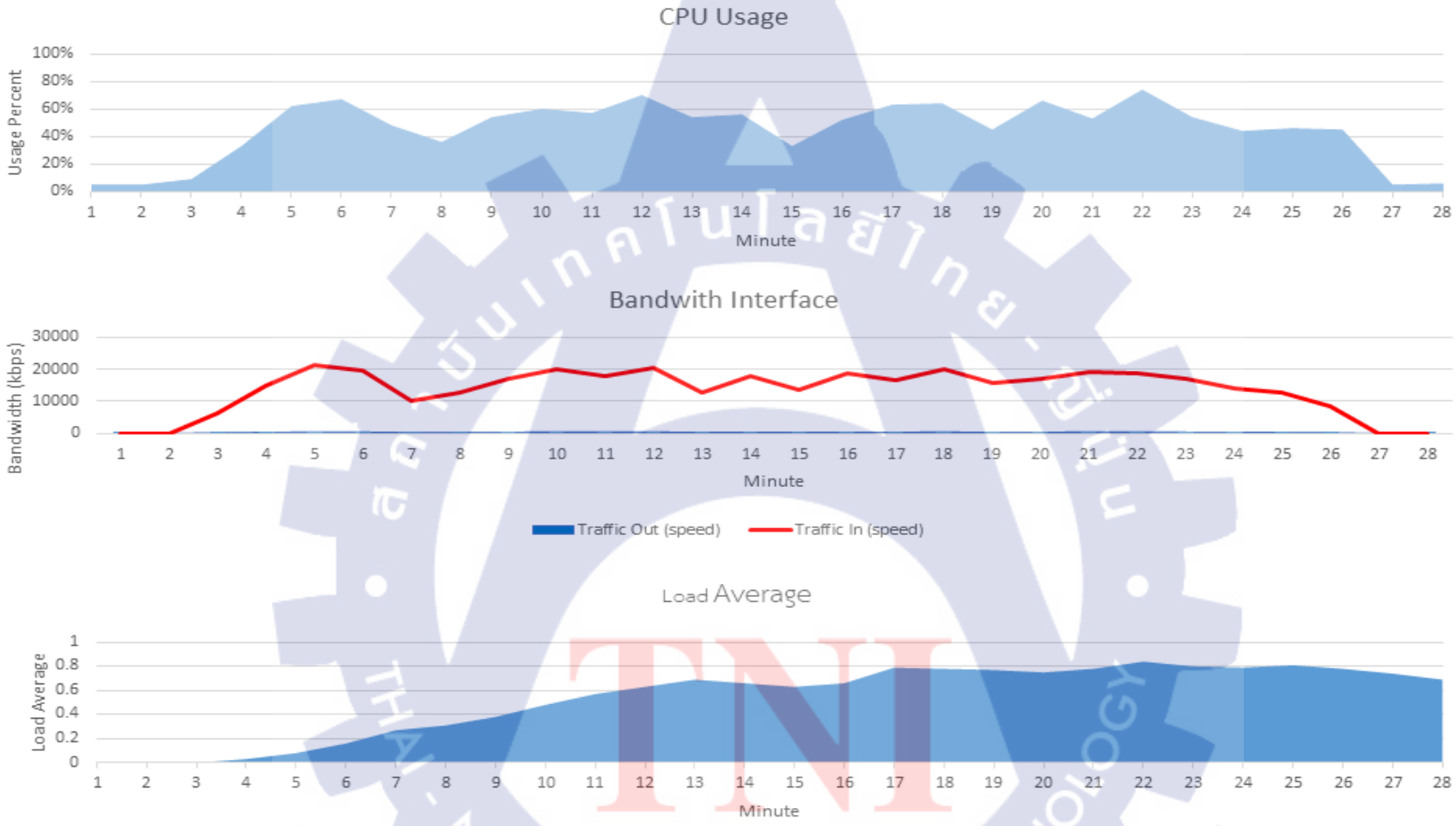
โดยในการทดสอบแต่ละครั้งจะนำกราฟที่ได้จาก monitoring tools ที่ชื่อว่า PRTG ซึ่ง Raspberry Pi Firewall จะทำการส่งค่าด้วยโพรโตคอล SNMP ไปยัง Monitoring Server ซึ่งชุดข้อมูลที่ส่งไปนั้น จะส่งในรูปแบบข้อความ โดยมีเลข OID ที่แสดงถึงข้อมูลค่าต่างๆกำกับอยู่ด้วย โดยแบ่งเป็นตามลำดับ เช่น CPU usage, Load Average, Memory, Interface Bandwidth ตามลำดับ จากนั้น นำชุดข้อมูลดังกล่าวมาทำการ generate graph ทุกๆ 1 นาที โดยแยกเป็นกราฟตามหมายเลข OID

โดยในงานค้นคว้านี้ จะแสดงผลการทดสอบประสิทธิภาพ 3 ด้าน คือ CPU Processing Usage , Bandwidth Interface usage และ Load Average

การทดสอบประสิทธิภาพ Firewall โดยใช้ IPTable

การทดสอบการส่งข้อมูลด้วย FTP

1. การทดสอบด้วย FTP แบบส่งไฟล์ครั้งเดียว ขนาด 2GB จำนวน 1 ไฟล์



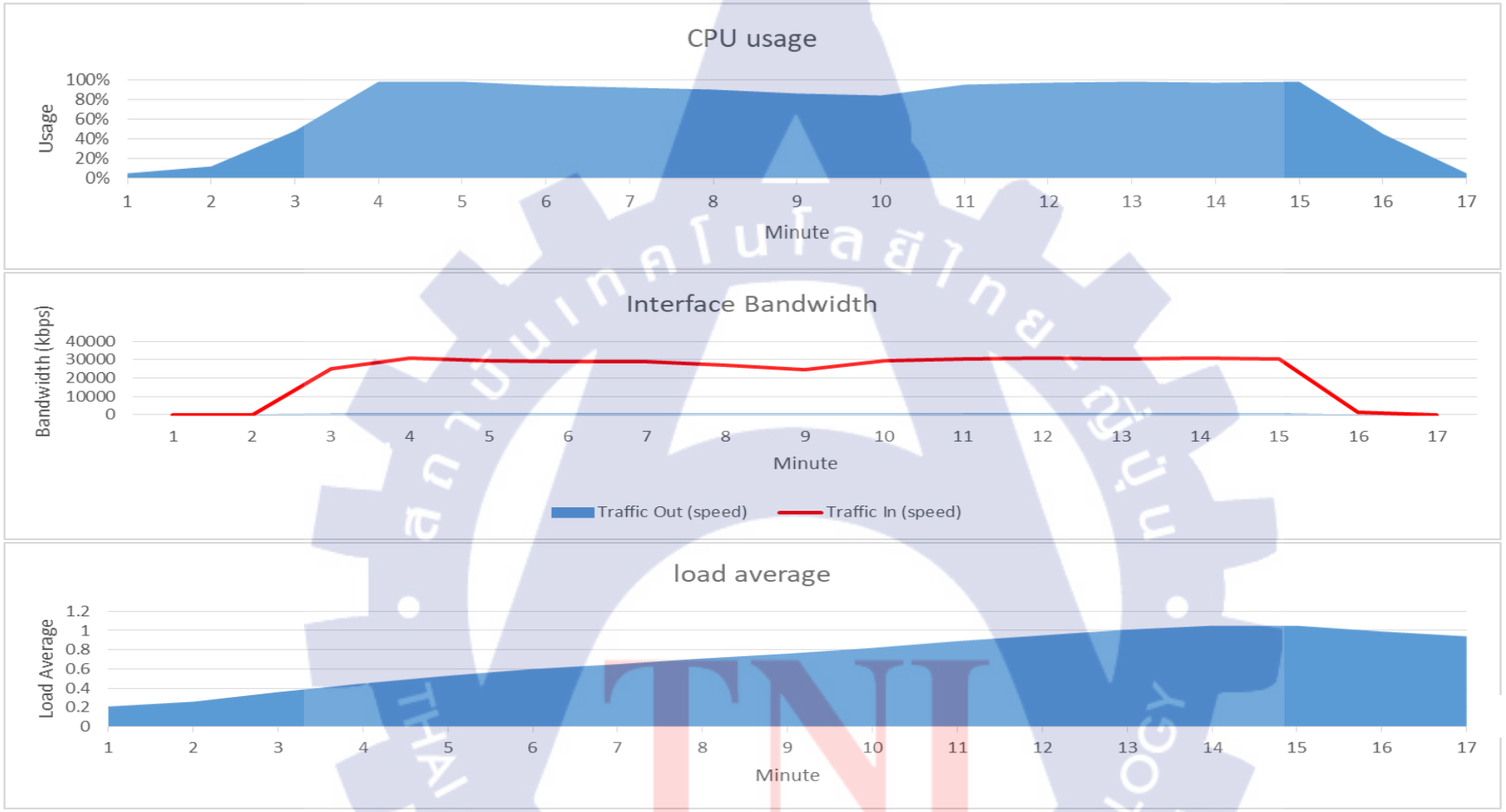
รูปที่ 4.1 กราฟแสดงผลการทดสอบการทดสอบด้วย FTP แบบส่งครั้งเดียว 1 ไฟล์ ของ IPTable

จากรูปที่ 4.1 เมื่อเริ่มมีการส่งไฟล์ด้วย FTP การทำงานของ CPU จะเริ่มการทำงานในช่วงนาทีที่ 3 โดยจะเพิ่มขึ้นเรื่อย ๆ และมีการเพิ่มขึ้นของ CPU usage โดยเฉลี่ย 45 % ซึ่งในกรณีนี้เพิ่มขึ้นตาม bandwidth interface ที่เพิ่มด้วย โดยจะสังเกตเห็นว่าการทำงานของ FTP ยังสามารถทำงานได้ตามปกติโดยที่ การใช้งานของ CPU ไม่ได้ขึ้นจนถึง 100%

ส่วนการใช้งานอินเตอร์เฟส สอดคล้องกับการทำงานของ CPU เมื่อแบนด์วิธการใช้งานเพิ่มขึ้น CPU ทำงานสูงขึ้นตามลำดับ โดยเฉลี่ยแล้วแบนด์วิธที่ใช้งานสำหรับการส่ง FTP ในการทดลองโดยใช้ IPTable สามารถรับส่งได้สูงสุด 20,000 kbps (20Mbps) โดยประมาณ และส่งได้เฉลี่ย 13,700 kbps (13.7 Mbps)

การใช้งานค่าเฉลี่ยภาระในการประมวลผลซึ่งจะแสดงการทำงานภาระการทำงานย้อนหลังของ CPU 15 นาที เพื่อเห็นว่าในช่วงเวลาดังกล่าวมี process ที่รอประมวลผลอยู่มากน้อยเพียงใดและส่งผลทำให้ CPU Process ในรูปที่ 4.1 ทำงานหนักขึ้น จากกราฟดังกล่าวจะแสดงให้เห็นว่า Load ของการทำงานโดยเฉลี่ยไม่ได้ทำงานหนักมากจนเกินไป โดยที่ค่าสูงสุดของ ค่าเฉลี่ยภาระในการประมวลผลนั่น อยู่ที่ประมาณ 0.85 ซึ่งสอดคล้องกับการทำงานและประสิทธิภาพของ CPU

จากผลของการทดสอบการส่งด้วยโพรโตคอล FTP ทั้งการใช้งาน CPU แบนด์วิธที่ใช้งานและค่าเฉลี่ยภาระของหน่วยประมวลผล ในช่วงที่กำลังส่งข้อมูลอยู่ แบนด์วิธที่ใช้งาน เพิ่มมากขึ้นจนถึงขีดสูงสุดของอุปกรณ์ที่รับได้ หรือ bandwidth อินเตอร์เน็ตสูงสุดที่ได้รับ การใช้งาน CPU ยังไม่ถึง 100% และ ทำการส่งไฟล์เสร็จสมบูรณ์ด้วยเวลา 27 นาที



รูปที่ 4.2 กราฟแสดงผลการทดสอบการทดสอบด้วย FTP แบบส่งครั้งเดียว 3 ไฟล์ พร้อมกัน ของ IPTable

จากรูปที่ 4.2 เมื่อเริ่มมีการส่งไฟล์ด้วย FTP การทำงานของ CPU มีการใช้งานเฉลี่ย 73% ซึ่งในกรณีนี้มาจากการทำงานของ session ที่เพิ่มมากขึ้น และพร้อมกันทำให้มีการใช้งานของ CPU ที่สูง โดยเฉลี่ยแล้วแบนด์วิธที่ใช้งานสำหรับการส่ง FTP ใน iptable จะสามารถรับส่งได้สูงสุด 31,000 kbps (31 Mbps) และส่งได้เฉลี่ย 22,000 kbps (22 Mbps) และทำการส่งไฟล์เสร็จสมบูรณ์ด้วยเวลา 17 นาที

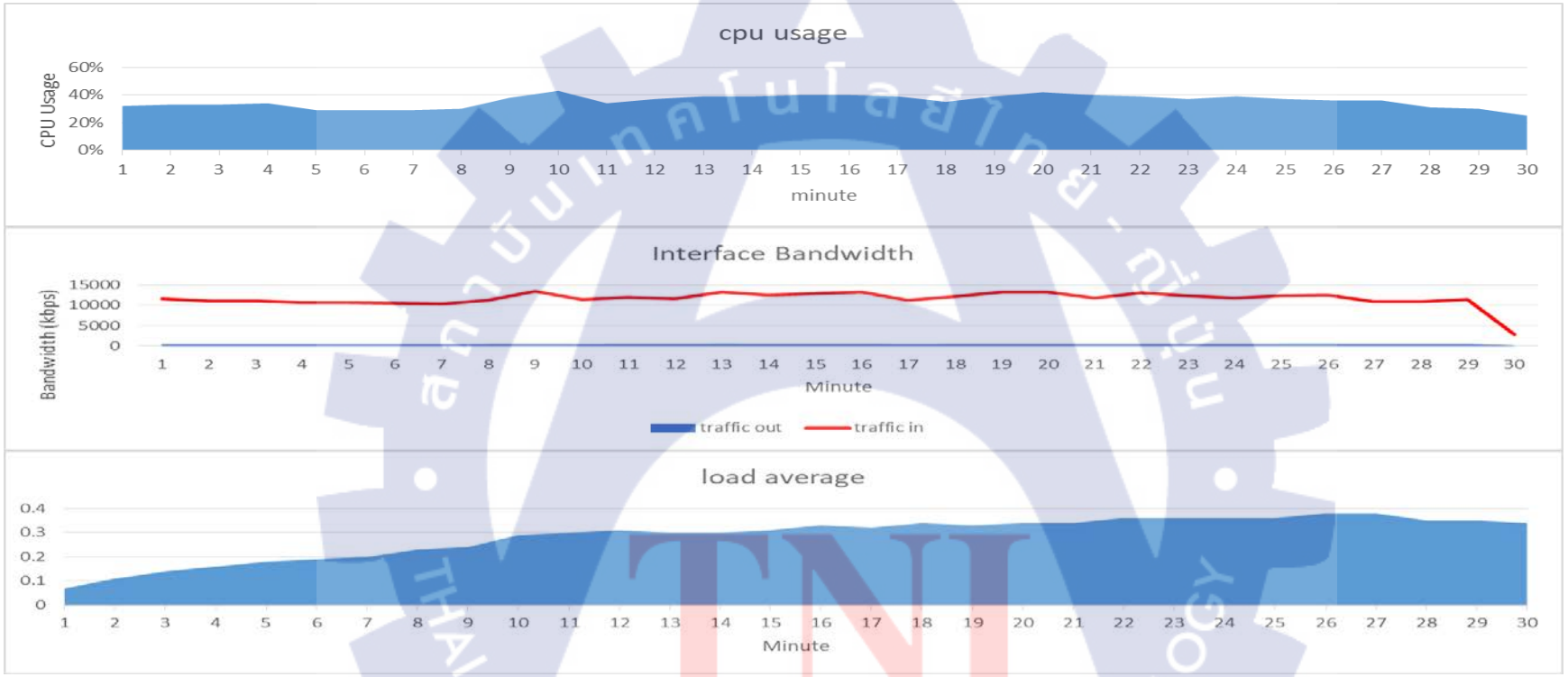
ค่าภาระการประมวลผลของรูปที่ 4.2 นี้เมื่อเทียบกับกรณีส่งไฟล์เดียวในกราฟค่าเฉลี่ยภาระการประมวลผลในรูปที่ 4.1 จะมีค่ามากขึ้น เพราะ CPU Process นั้นมีค่ามากขึ้นด้วย แต่ทั้งนี้การทำงานยังสามารถทำงานได้ตามปกติ ไม่มีการสูญหายของข้อมูล เพียงแต่เป็นการรอการทำงานของ process นั้น เนื่องจากมี session การทำงานที่เพิ่มขึ้น

ในกรณีที่มีการส่ง FTP พร้อมกัน 3 ครั้งจะทำให้การทำงานของ CPU นั้น ทำงานหนักมากยิ่งขึ้น และในแต่ละไฟล์จะเฉลี่ย bandwidth การส่งให้เท่าๆ กันตามขีดความจำกัดของ bandwidth ของไฟร์วอลล์ที่มี และปัจจัยอื่น อย่างเช่นการทำงานโหลดเฉลี่ย เมื่อมีการส่ง FTP พร้อมกัน 3 ไฟล์ จะทำให้ โหลดการทำงานได้ต่อเนื่องมากกว่าการส่งแบบไฟล์เดียวซึ่งค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 1.05

โดยจากการทดลองของ FTP ในรูปที่ 4.1 และ 4.2 จะเห็นได้ว่าการใช้งานแบบส่ง 3 ไฟล์พร้อมกันจะใช้การใช้งาน CPU และค่าเฉลี่ยภาระงานมากกว่าการส่ง FTP แบบส่งครั้งเดียว 1 ไฟล์ เพราะว่านอกจากที่จะต้องใช้แบนด์วิธที่ต่อเนื่องกัน และใช้ปริมาณสูงในการทดลองแบบส่งพร้อมกัน 3 ไฟล์จะมีจำนวนเซสชั่น (Session) เข้ามาเกี่ยวข้องด้วยจึงทำให้มีการใช้ปริมาณภาระงานที่เพิ่มมากขึ้น

การทดสอบประสิทธิภาพในการเข้าถึงข้อมูลเว็บไซต์

1. การทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยไม่มีการปิดกั้นใดๆ ทั้งสิ้น



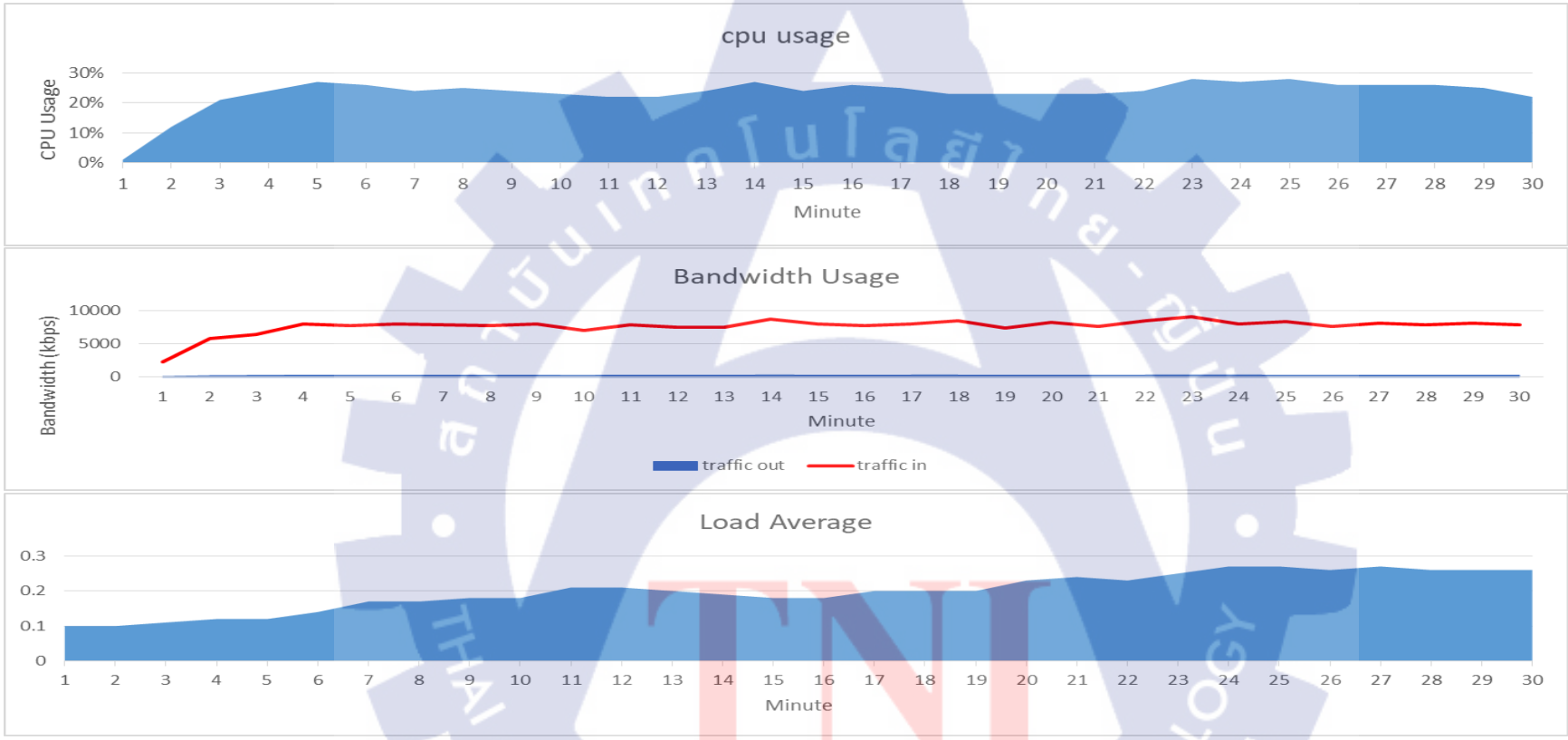
รูปที่ 4.3 กราฟแสดงผลการทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยไม่มีการปิดกั้นใดๆ ทั้งสิ้น ของ IPTable

จากรูปที่ 4.3 จะแสดงการทำงานของการใช้งานในหน่วยประมวลผล CPU ของการเข้าถึงเว็บไซต์ จำนวน 10 เว็บไซต์ การใช้งาน CPU โดยเฉลี่ย 34% ส่วนการใช้งานแบนด์สูงสุด 13,440 kbps (13.44 Mbps) ซึ่งเป็น ปริมาณของ content ทั้งหมด 10 เว็บไซต์มารวมกัน โดยที่การทำงานนั้นสอดคล้องกับการทำงานของ CPU เนื่องจาก bandwidth มีการใช้งานไม่เยอะมาก ทำให้การใช้งาน CPU ไม่ได้มากขึ้น

ในส่วนของค่าเฉลี่ยภาระงานของหน่วยประมวลผล เนื่องจากค่าของ CPU อยู่ในระดับที่ไม่ถึง 100% ทำให้กราฟค่าเฉลี่ยภาระงานของหน่วยประมวลผลมีค่าไม่เกิน 1 ซึ่งแสดงว่า process ในการทำงานการเข้าถึงเว็บไซต์นั้น มี process ที่ค้างอยู่เป็นจำนวนไม่มาก เพราะเนื่องจากอัตราการส่งและ ปริมาณของข้อมูลของการทดลองที่น้อย ดังนั้นค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 0.38

ทั้งนี้การที่กราฟ interface bandwidth ในรูปที่ 4.3 มีค่าไม่คงที่ขึ้นอยู่กับปริมาณ content ของในแต่ละเว็บไซต์รวมถึงในช่วงจังหวะที่ทำการ refresh หน้าเว็บไซต์ซึ่งในบางครั้งอาจจะดึงข้อมูลที่อยู่ในเว็บไซต์ที่ไม่เหมือนกัน ยกตัวอย่างเช่น หน้าโฆษณา

2. <u>การทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยมีการปิดกั้นเว็บไซต์บางส่วน</u>



รูปที่ 4.4 กราฟแสดงผลการทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยมีการปิดกั้นเว็บไซต์บางส่วน ของ IPTable

การทดลองนี้เป็นการแสดงการทำงานของการใช้งานในหน่วยประมวลผล CPU ของการเข้าถึงเว็บไซต์ จำนวน 10 เว็บไซต์ และทำการปิดกั้นเว็บไซต์จำนวน 5 เว็บไซต์ ได้แก่ Lazada.com, movie2free.com, sanook.com, dek-d.com และ kapok.com ซึ่งการใช้งาน CPU โดยเฉลี่ย 23%

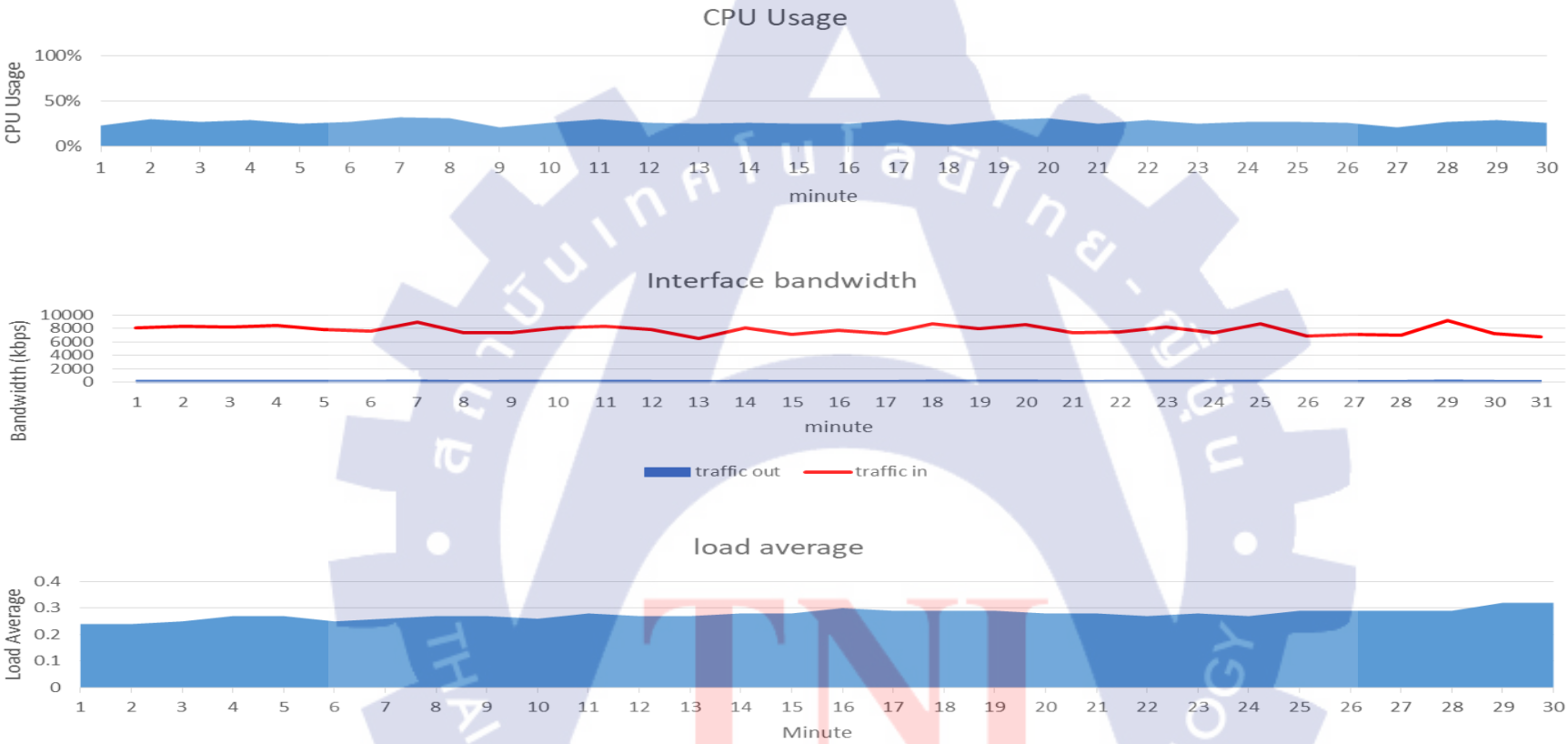
ส่วนของการทำงานของการใช้แบนด์วิธ ของการเข้าถึงเว็บไซต์ เมื่อทำการปิดกั้นเว็บไซต์ไปบางส่วน การใช้งานแบนด์วิธโดยเฉลี่ย 7,630 kbps (7.63 Mbps) และค่าสูงสุดของแบนด์วิธอยู่ที่ 9,100 kbps (9.1 Mbps) ซึ่งเป็น ปริมาณของ content ทั้งหมดที่ไม่ได้ทำการปิดกั้น มารวมกัน โดยที่การทำงานนั้นสอดคล้องกับการทำงานของ CPU ในกรณีที่ bandwidth มีการใช้งานไม่เยอะมาก ทำให้การใช้งาน CPU ไม่ได้มากขึ้นตามไป

ในส่วนของค่าเฉลี่ยภาระงานของหน่วยประมวลผล เนื่องจากค่าของ CPU อยู่ในระดับที่ไม่ถึง 100% ทำให้กราฟค่าเฉลี่ยภาระงานของหน่วยประมวลผลมีค่าไม่เกิน 1 ซึ่งแสดงว่า process ในการทำงานการเข้าถึงเว็บไซต์นั้น มี process ที่ค้างอยู่เป็นจำนวนน้อย เพราะเนื่องจากอัตราการส่งและ ปริมาณของข้อมูลที่น้อย ดังนั้นค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 0.27

โดยจากการทดลองนี้ผลที่ได้ออกมาจากการที่ทำการปิดกั้นจำนวน 5 website ทำการปิดกั้นสำเร็จ ทั้งหมดโดยไม่มี content ของเว็บไซต์ที่ทำการปิดกั้นปนอยู่ในผลการทดลอง และเนื่องจากทั้งนี้การที่กราฟ interface bandwidth ในรูปที่ 4.4 มีค่าไม่คงที่ขึ้นอยู่กับปริมาณ content ของในแต่ละเว็บไซต์ รวมถึงในช่วงจังหวะที่ทำการ refresh หน้าเว็บไซต์ ซึ่งในบางครั้งอาจจะดึงข้อมูลที่อยู่ในเว็บไซต์ไม่เหมือนกัน ยกตัวอย่างเช่น หน้าโฆษณา

สรุปแล้วจากการทดลองของการเข้าถึงเว็บไซต์ทั้ง 2 การทดลองนั้น การทดลองแบบไม่มีการปิดกั้นใดๆ เป็นการทดลองที่ใช้ประสิทธิภาพการใช้งาน CPU แบนด์วิธ และค่าเฉลี่ยภาระงานที่มากกว่า การทดลองแบบทำการปิดกั้น เนื่องจาก content ที่อนุญาตผ่านไปยัง Raspberry Pi Firewall ที่มีมากกว่า ซึ่งในการทดสอบนี้สรุปได้ว่าปริมาณของ rule ที่ทำการปิดกั้นนั้น มีจำนวนไม่มาก แต่ สาเหตุหลักที่การใช้งาน CPU มีค่าที่สูงเพราะมาจากปริมาณ content ที่อนุญาตผ่านจาก Raspberry Pi firewall มีผลมากกว่าจำนวน rule ของ firewall ที่ทำการปิดกั้น

3. <u>การทดสอบประสิทธิภาพในการรับส่ง email ด้วย โพรโตคอล SMTP</u>



รูปที่ 4.5 กราฟแสดงผลการประสิทธิภาพในการรับส่ง email ด้วย โพรโตคอล SMTP ของ IPTable

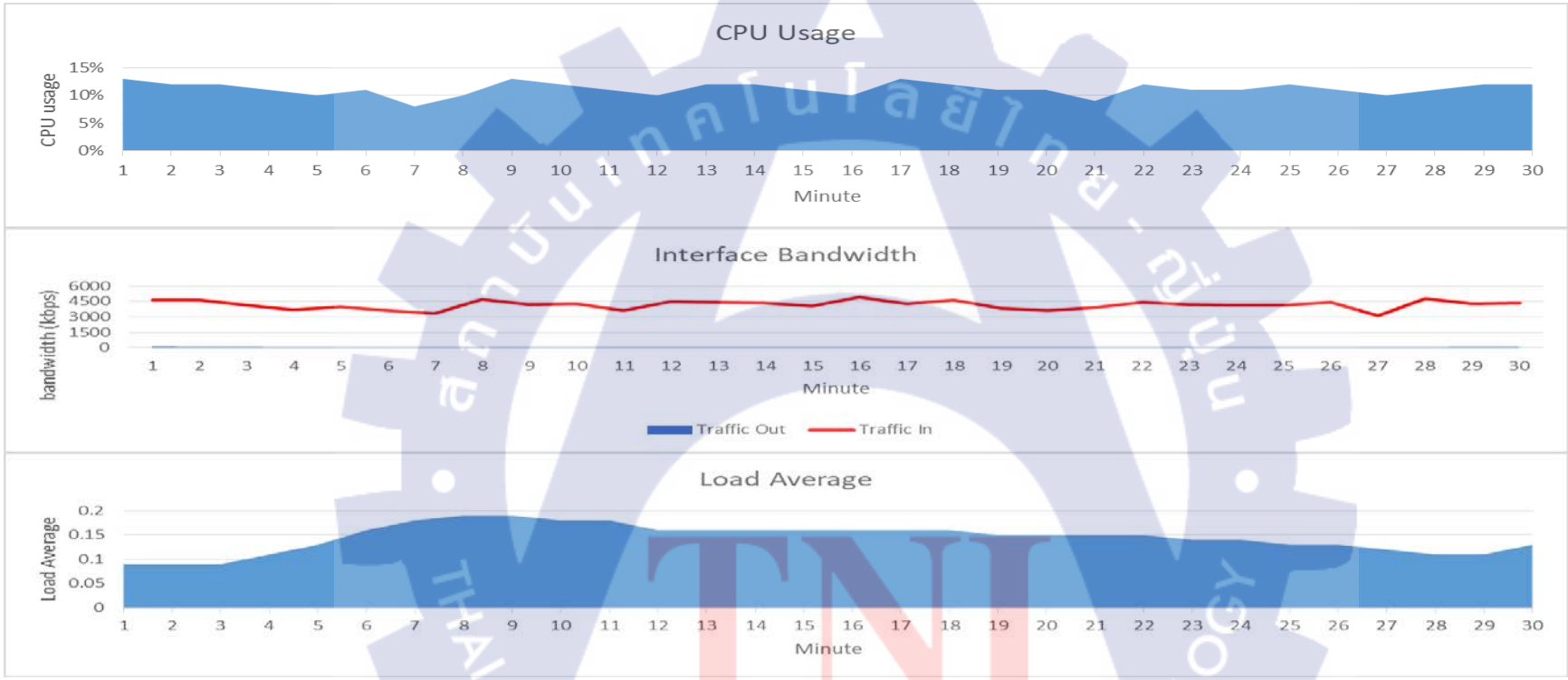
จากกราฟรูปที่ 4.5 เนื่องด้วยการทำงานของ email ตามที่กล่าวไว้แล้วในบทที่ 3 จะเป็นการส่งข้อมูลเพียง 10MB ต่อ 1 วินาที โดยที่การใช้งาน CPU สูงที่สุดในการทดลองนี้อยู่ที่ 32% และค่าเฉลี่ยอยู่ที่ประมาณ 27% ซึ่งในบางครั้งจะมีบางช่วงที่ไม่สามารถส่งได้ตามที่ตั้งค่าเอาไว้คือ 10MB เนื่องจากความเสถียรของ Internet ในช่วงเวลานั้น และจะมีการทบการส่งไปยังช่วงวินาทีต่อไป ทำให้กราฟ interface bandwidth ในรูปที่ 37 ที่ได้นั้น ไม่ได้คงที่เสมอไป

สำหรับแบนด์วิธที่ใช้งานนั้นจะสอดคล้องกับคำอธิบายของการใช้งาน CPU โดยในการทดลองจะเป็นการส่งข้อมูลจำนวน 10MB ทุกๆ 1 วินาที ซึ่งจากกราฟ interface bandwith ในรูปที่ 4.5 IPTable สามารถทำงานได้ และ Raspberry Pi สามารถส่ง email ได้ตลอดทุกช่วงเวลา โดยที่การใช้งานแบนด์วิธสูงสุดอยู่ที่ 9,240 kbps (9.24 Mbps) และจากค่าเฉลี่ยภาระงานในการทำงานของ CPU สูงสุดอยู่ที่ 0.32 ซึ่งสอดคล้องกับการใช้งานของ CPU

จากผลการทดลองดังกล่าวสามารถสรุปได้ว่า IPTables สามารถใช้งานในด้านการรับและส่ง email ได้ เพราะว่า แบนด์วิธการใช้งาน email นั้นไม่ได้สูงมาก ทำให้ Raspberry pi firewall ยังสามารถทำงานได้อย่างมีประสิทธิภาพ และยังสามารถรองรับการใช้งานอื่นๆ ได้อีก โดยดูจากประสิทธิภาพการใช้งาน CPU เป็นหลัก

4 การทดสอบประสิทธิภาพการใช้งาน Live Streaming

ก. การทดสอบประสิทธิภาพการใช้งาน Live Streaming โดยไม่ได้มีการปิดกั้นใดๆ



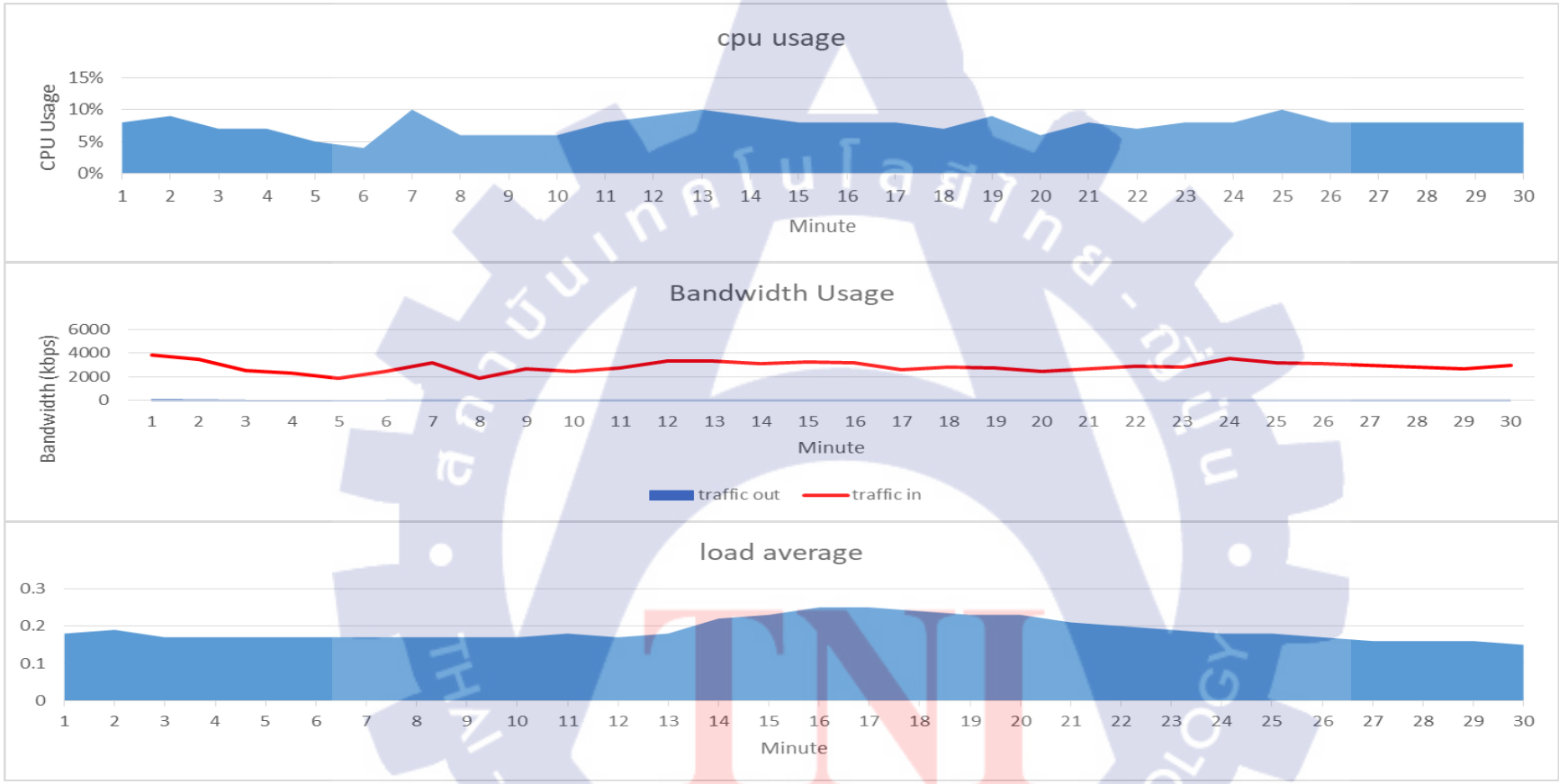
รูปที่ 4.6 กราฟแสดงผลประสิทธิภาพการใช้งาน Live Streaming แบบไม่ได้ทำการปิดกั้นของ IPTable

ผลการทดสอบแบบ Live Streaming นี้เป็นผลการทดสอบ Live Streaming โดยทำการเปิด Streaming ช่องทีวี 3 ช่องได้แก่ Thaitv3, Thairath TV และ Amarin TV พร้อมกันด้วยความละเอียด 720p ซึ่งค่าเฉลี่ย ของการใช้งาน CPU ของการทดสอบนี้อยู่ที่ 14% และค่าสูงสุดของการใช้งาน CPU อยู่ที่ 19% จากผลของ CPU Usage แสดงว่า Bandwidth ที่ใช้งานโดยเฉลี่ย 1 ช่องจะใช้ CPU Usage โดยประมาณ 4.6%

ส่วนค่าเฉลี่ยของแบนด์วิธอยู่ที่ 4,200 kbps (4.2Mbps) และค่าสูงสุดของการใช้งานแบนด์วิธอยู่ที่ 4,950 kbps (4.95 Mbps) จากกราฟ Interface bandwidth ของรูปที่ 4.6 สรุปได้ว่าโดยเฉลี่ย 1 ช่องจะใช้แบนด์วิธ โดยประมาณ 1,650 kbps (1.65 Mbps) ซึ่งตรงตามหลักการของการ streaming ในช่วงความละเอียด 720p จะต้องใช้แบนด์วิธอย่างน้อยต้องอยู่ในช่วง 2-4 Mbps ทั้งนี้ขึ้นอยู่กับ content และภาพที่ส่งมาในช่วงเวลานั้นๆ ว่าในรายการนั้นๆ ส่งภาพมาในรูปแบบใด โดยในแต่ละช่วงเวลาจะมีปริมาณของ content ไม่เท่ากัน

ส่วน ค่าสูงสุดของค่าเฉลี่ยภาระงาน อยู่ที่ 0.19 ซึ่งสอดคล้องกับการใช้งาน CPU ที่มีค่าไม่เกิน 1 นั่นหมายความว่าการทำงานของ Live Streaming ของ IPTable ยังสามารถทำงานได้อย่างเต็มประสิทธิภาพ และสามารถรองรับการใช้งานอื่นๆ ได้อีก

ข. การทดสอบประสิทธิภาพการใช้งาน Live Streaming โดยมีการปิดกั้นบางช่อง



รูปที่ 4.7 กราฟแสดงผลประสิทธิภาพการใช้งาน Live Streaming แบบมีการปิดกั้นบางช่องของ IPTable

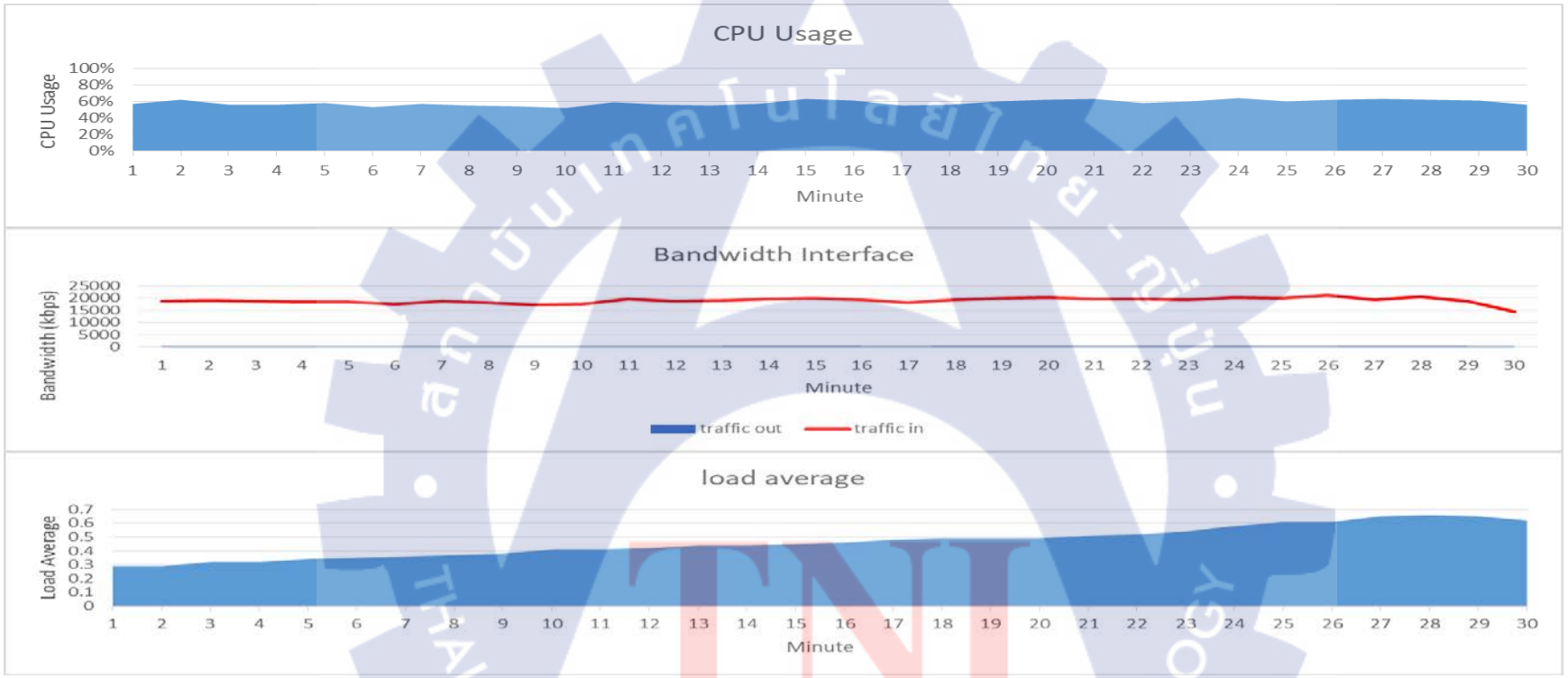
ผลการทดสอบแบบ Live Streaming นี้เป็นผลการทดสอบ Live Streaming โดยทำการเปิดช่องทีวี 3 ช่องและทำการปิดกั้นด้วย IPTable จำนวน 1 ช่อง นั่นคือ thaitv3.com ด้วยความละเอียด 720p ซึ่งค่าเฉลี่ยการใช้งาน CPU ของการทดสอบนี้อยู่ที่ 8% และค่าสูงสุดของการใช้งาน CPU อยู่ที่ 10% ซึ่งสอดคล้องกับ Bandwidth ที่ใช้งานโดยเฉลี่ย 1 ช่องเหมือนที่กล่าวไปในกรณีเปิดพร้อมกัน 3 ช่อง แต่ว่าในการทดลองนี้จะเหลือ traffic ช่องรายการ ที่ผ่านได้จำนวน 2 ช่องเท่านั้น จึงทำให้ค่าการใช้งาน CPU จึงลดลง โดยทั้งนี้ขึ้นอยู่กับปริมาณ content และภาพที่ส่งมาในช่วงเวลานั้นๆ ว่า ในรายการนั้นๆ ส่งภาพมาในรูปแบบใด โดยในแต่ละช่วงเวลาจะมีปริมาณของ content ไม่เท่ากัน

ค่าสูงสุดของการใช้งานแบนด์วิธ อยู่ที่ 3,820 kbps (3.82 Mbps) การใช้งานโดยเฉลี่ย 1 ช่องเหมือนที่กล่าวไปในกรณีเปิดพร้อมกัน 3 ช่อง แต่ว่าในการทดลองนี้จะเหลือ traffic ที่ผ่านได้จำนวน 2 ช่องเท่านั้น ซึ่งผลที่ได้นั้นก็สอดคล้องกับการใช้งาน CPU ที่ลดลงไปเพราะมีการถูกปิดกั้น traffic ด้วย

ค่าสูงสุดของค่าเฉลี่ยภาระงาน อยู่ที่ 0.25 ซึ่งสอดคล้องกับการใช้งาน CPU และ แบนด์วิธที่ลดลง และการใช้งาน CPU ที่มีค่าไม่เกิน 1 นั่นหมายความว่าการทำงานของ IPTable ยังสามารถทำงานได้อย่างเต็มประสิทธิภาพ และยังสามารถรองรับการใช้งานอื่นๆ ได้อีก

ซึ่งจากการทดสอบการใช้งาน Live Streaming จากรูปที่ 4.6 และ 4.7 กรณีที่ไม่ได้ทำการปิดกั้นช่องจะใช้ประสิทธิภาพการใช้งาน CPU แบนด์วิธ และค่าเฉลี่ยภาระงานที่มากกว่า ทั้งนี้เพราะขนาดของ content ทั้งหมดของกรณีแรกมีจำนวนมากกว่า และการประมวลผลการตรวจสอบ Rule ของ firewall ก็ไม่ได้ส่งผลต่อการใช้งานประสิทธิภาพมากนักจนสังเกตเห็นความแตกต่างได้

5. การทดสอบประสิทธิภาพการทำงานพร้อมกันของแอพพลิเคชั่นมากกว่า 1 ประเภท ได้แก่ การเข้าถึง website การส่ง email และ live streaming



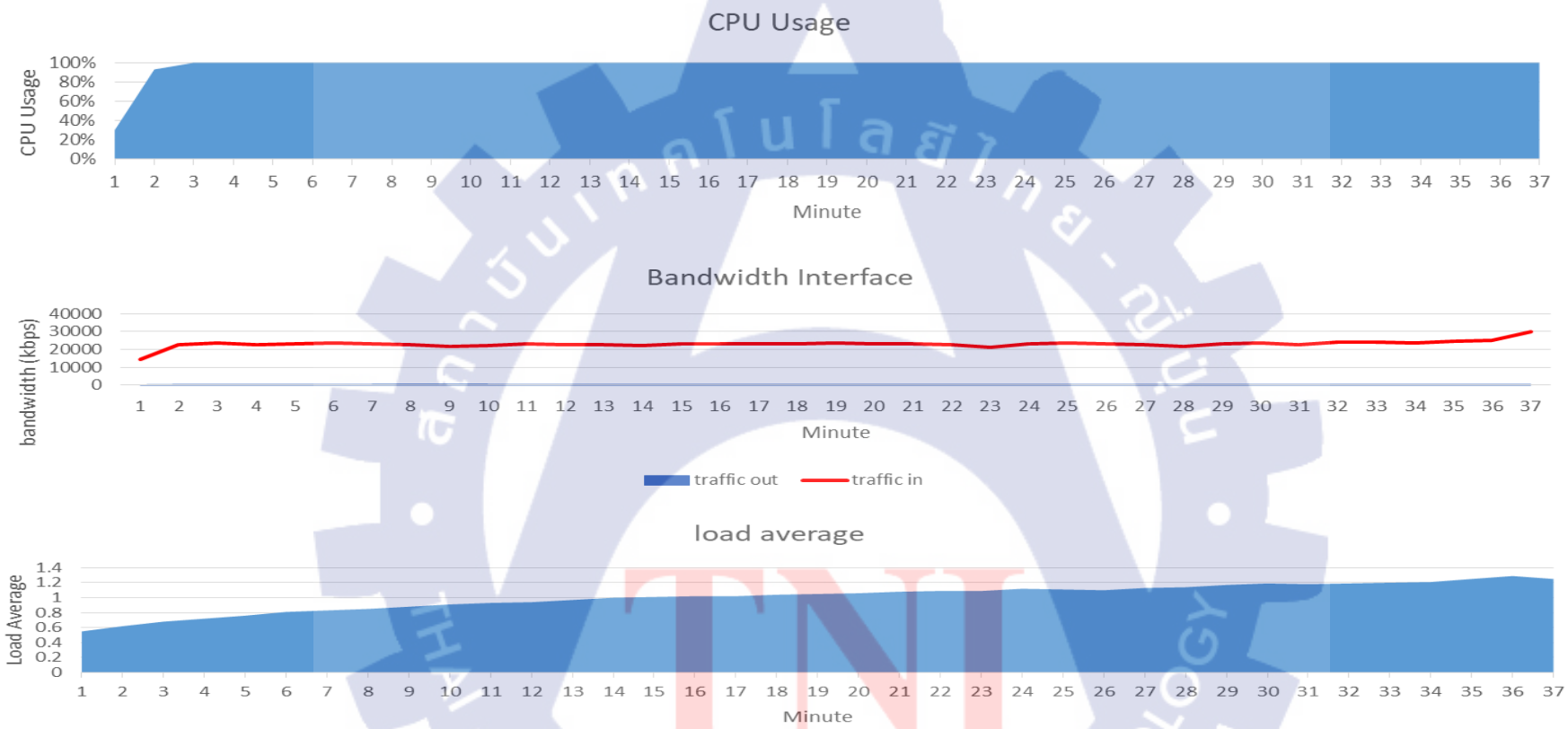
รูปที่ 4.8 กราฟแสดงผลประสิทธิภาพการทำงานแบบแอพลิเคชั่นมากกว่า 1 อย่างของ IPTable

จากรูปที่ 4.8 เป็นกราฟโดยการทดสอบทำงานแบบหลายแอพพลิเคชั่น โดยอ้างอิงจากสถิติ ได้แก่ การใช้งานการเข้าถึงเว็บไซต์ การส่ง email และ Live Streaming ซึ่งค่าเฉลี่ยของ CPU Usage การทดลองนี้อยู่ที่ 57% โดยค่าสูงสุดของการทดลองนี้อยู่ที่ 64% แบนด์วิธสูงสุดในการใช้งานอยู่ที่ 21,220 kbps (21.22 Mbps)

ส่วน ค่าเฉลี่ยภาระงาน ซึ่งจะสังเกตเห็นว่าค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 0.66 เพราะฉะนั้นประสิทธิภาพการทำงาน ยังสามารถทำงานได้ดี เพราะว่า CPU ที่ใช้งานยังไม่ถึง 100% และไม่มีการสะดุดหรือเกิดการขัดข้อง หรือเรียกว่า buffering ในระหว่างการใช้งาน

โดยที่สามารถใช้งานได้เต็มประสิทธิภาพ สามารถส่ง email ได้ตลอดช่วงเวลา เข้าถึงเว็บไซต์ที่ทำการเปิดให้ใช้งานได้ รวมถึงสามารถใช้งาน Live Streaming พร้อมกันได้โดยไม่มีการติดขัดระหว่างรับชมรายการ

6. การทดสอบประสิทธิภาพการทำงานแบบแอพลิเคชั่นทุกอย่างเข้าด้วยกันทั้ง FTP การเข้าถึงเว็บไซต์ Email และ Livestreaming



รูปที่ 4.9 กราฟแสดงผลประสิทธิภาพ การทำงานแบบแอพลิเคชั่นทุกอย่างเข้าด้วยกัน ของ IPTable

จากการทดสอบแบบการนำทุกแอพพลิเคชั่นทำงานพร้อมกัน ทั้ง FTP การเข้าถึงเว็บไซต์ Email และ Livestreaming จะสังเกตว่าในระหว่างการทดลอง ประสิทธิภาพการใช้งาน CPU ขึ้นไปถึง 100% ซึ่งการใช้งาน FTP ทำการ upload ไฟล์เสร็จสิ้นเป็นเวลา 37 นาที ซึ่งแตกต่างจากการทดสอบการใช้งาน FTP เพียงอย่างเดียว ดังรูปที่ 34 ในส่วน CPU Usage จะใช้งานโดยเฉลี่ย 73% และส่งไฟล์สมบูรณ์เพียง 17 นาทีเท่านั้น ซึ่งเร็วกว่าการทดสอบนี้

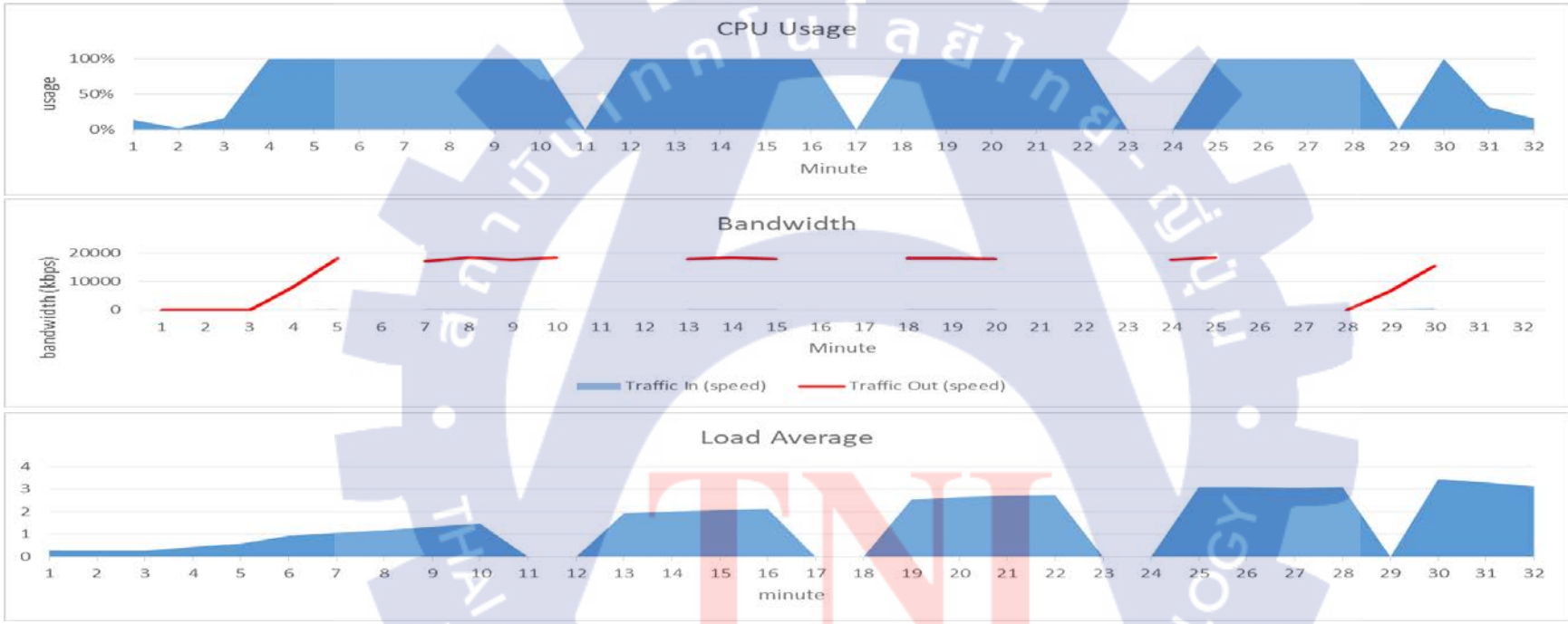
ในด้าน bandwidth ที่ใช้งาน แบนด์วิธการใช้งานสูงสุดอยู่ที่ 29,990 kbps (29.9 Mbps) ซึ่งเป็นค่าสูงสุดของความสามารถของ IPTable ที่สามารถดาวน์โหลด content ได้ และค่าเฉลี่ยภาระงานสูงสุดของการทดลองนี้อยู่ที่ 1.29 นั่นแสดงว่าในบางช่วงเวลามีภาระงานค้างอยู่ ทำให้การใช้งานเว็บไซต์ อีเมล์ และ live streaming ทำงานได้อย่างไม่เต็มประสิทธิภาพ เพราะมี traffic จาก FTP ดึงไปเป็นจำนวนมาก

เมื่อเทียบกับการใช้งานแบบการนำแอพพลิเคชั่นมากกว่า 1 อย่างในรูปที่ 4.8 นั้น ประสิทธิภาพของการทำงานของการทดลองนี้ แย่กว่าการทดลองในรูปที่ 4.8 เพราะว่าเป็นการใช้ CPU ที่สูง และค่าเฉลี่ยภาระงานที่เยอะกว่า เมื่อเทียบกับการทดลองแบบไม่มี FTP เพราะ traffic ของ การทดลองแบบการส่ง FTP แบบส่ง 3 ครั้งพร้อมกัน มีปริมาณมาก ดังรูปที่ 4.2 ทำให้การใช้งาน CPU มีค่อนข้างสูง และBandwidth ไม่เพียงพอต่อการใช้งานอื่นๆ ตามอัตราส่ง traffic ของแอพพลิเคชั่น ส่งผลให้การใช้งานบางอย่าง เช่น การใช้งานเว็บไซต์ที่สามารถเข้าได้ช้าบ้างในบางช่วงเวลา การใช้งาน email ที่ไม่สามารถส่งได้เต็ม 10 Mbps ในบางครั้ง และทำให้การ streaming ภาพของบางช่อง มีการติดขัดระหว่างรับชมช่องรายการ

การทดสอบประสิทธิภาพ Firewall โดยใช้ IPFire

1. <u>การทดสอบการส่งข้อมูลด้วย FTP</u>

   ก. การทดสอบด้วย FTP แบบส่งไฟล์ครั้งเดียว ขนาด 2GB จำนวน 1 ไฟล์



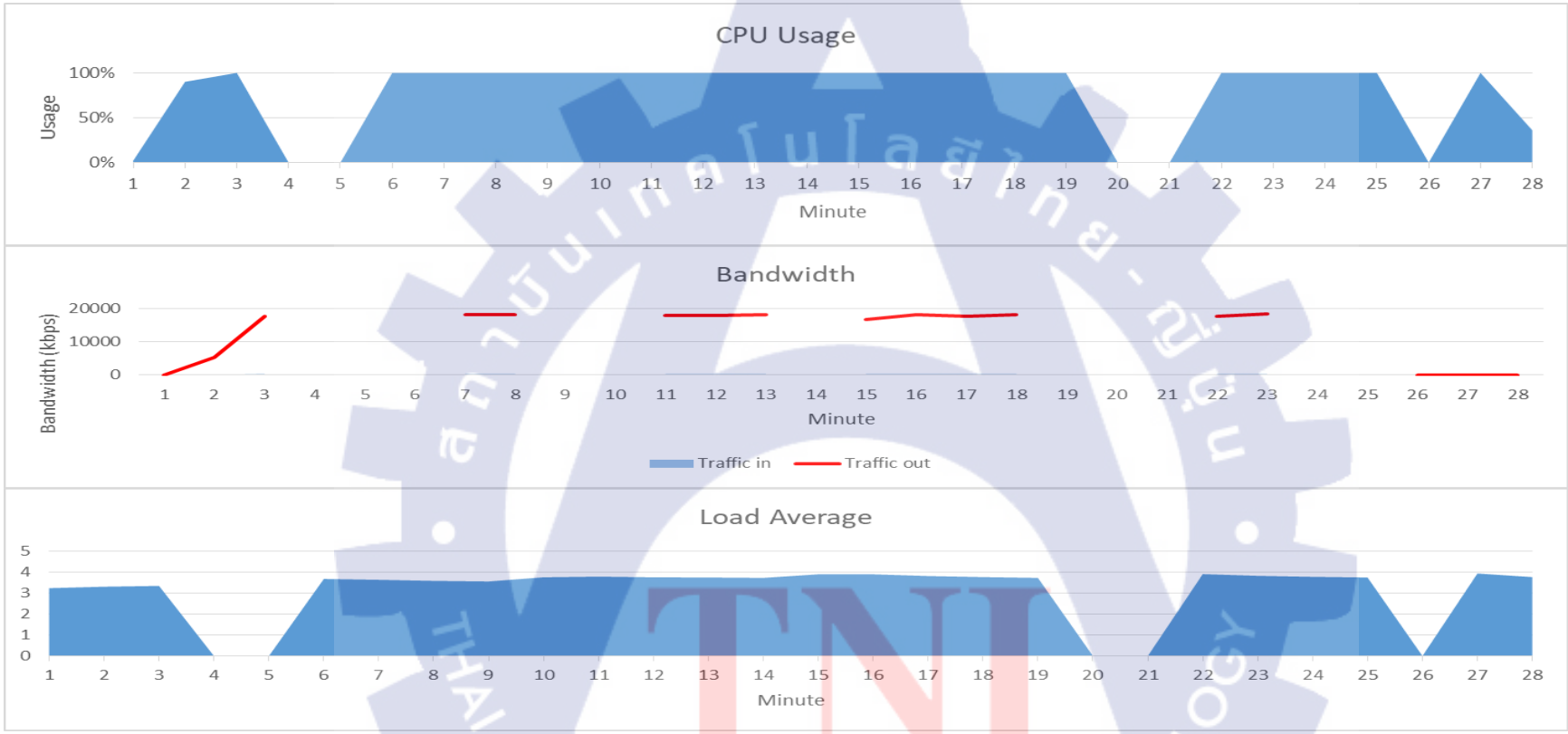
รูปที่ 4.10 กราฟแสดงผลการทดสอบการทดสอบด้วย FTP แบบส่งครั้งเดียว 1 ไฟล์ ของ IPFire

จากรูปที่ 4.10 ค่าของกราฟ CPU Usage ในระหว่างการส่งไฟล์อยู่ที่ 100% และมีบางช่วงเวลามีข้อมูลของกราฟที่ขาดหายไป เพราะว่าในระหว่างช่วงเวลาที่กำลังส่งข้อมูลของ FTP CPU ของ Raspberry Pi Firewall ทำงานเกินขีดความสามารถ และมีค่าภาระการทำงานของ CPU ที่สูงเกินไป ดังกราฟแสดงค่าเฉลี่ยภาระงานในรูปที่ 4.10 จึงไม่สามารถส่งค่ามายัง Monitoring Server กราฟจึงมีการขาดช่วงในบางส่วน ดังในนาทีที่ 11 ,17, 23, 29 ซึ่งมีการขาดหายพร้อมกันทั้ง 3 กราฟ ส่วนในกราฟ Bandwidth Usage ในนาทีที่ 6 ,12 , 16, 21, 22, 26 และในกราฟค่าเฉลี่ยภาระงานในนาทีที่ 27,12, 18 และ 24 เนื่องจากชุดข้อมูลจากโพรโตคอล SNMP ที่ส่งมายัง Monitoring Server โดยที่ค่า CPU Usage เป็นข้อมูล ในบรรทัดแรกๆ จึงทำให้กราฟ CPU Usage นั้นแสดงค่าได้มากกว่า รองลงมาเป็นค่าเฉลี่ยภาระงาน และ การใช้งาน Bandwidth ซึ่งเป็นข้อมูลในบรรทัดท้ายๆ ซึ่งในช่วงเวลาที่ภาระการทำงานของ CPU ทำงานหนัก ข้อมูลของ SNMP จะทำการหยุดส่งข้อมูลทันที ทำให้ข้อมูลของ SNMP นั้นจะได้รับไม่ครบถ้วน ในบางช่วงเวลา

เช่นเดียวกับกราฟของ Interface Bandwidth เนื่องจากการทำงานของ CPU นั้นทำงานเกินขีดความสามารถ และค่าเฉลี่ยภาระงานที่สูงเกินขีดความสามารถของ CPU ซึ่งมีแค่ 1 core ทำให้กราฟการใช้งานบางส่วนมีการขาดหายไป โดยในการทดสอบนี้อัตราการส่งข้อมูลมากที่สุดอยู่ที่ 18,000 kbps (18 Mbps) ซึ่งเป็น bandwidth สูงสุดที่ ณ ขนาดนั้น และค่าเฉลี่ยภาระงานค่าสูงสุดอยู่ที่ 3.5

ส่วนระยะเวลาของการส่ง FTP ของการทดลองนี้อยู่ที่ 27 นาที ซึ่งระยะเวลาเท่ากับของ IPTables แต่ประสิทธิภาพการทำงานของ CPU ของ IPTables จะทำงานได้ดีกว่า โดยที่ IPTables ใช้ CPU Usage โดยเฉลี่ยเพียงแค่ 45%

ข. การทดสอบด้วย FTP แบบส่ง 3 ครั้งพร้อมกัน โดยทำการแบ่งไฟล์ 2GB ออกเป็น 3 ไฟล์ และทำการส่งพร้อมกันด้วย FTP



รูปที่ 4.11 กราฟแสดงผลการทดสอบการทดสอบด้วย FTP แบบส่งครั้งเดียว 3 ไฟล์ พร้อมกัน ของ IPFire

จากรูปที่ 4.11 เมื่อ session เพิ่มขึ้น ทำให้การใช้งานของ CPU เพิ่มมากขึ้น โดยที่การใช้งาน CPU ขึ้นไป 100% ตลอดเวลา และในบางช่วงเวลามีช่วงของกราฟขาดหายไป เพราะเนื่องจากการใช้งาน process ที่สูงผิดปกติ ทำให้ Raspberry Pi Firewall ส่งข้อมูลได้ไม่ครบถ้วน ดังในนาทีที่ 4, 5, 20, 21 และ 26 ซึ่งมีการขาดหายพร้อมกันทั้ง 3 กราฟ ส่วนในกราฟ Bandwidth Usage ในนาทีที่ 6, 9, 10, 14, 19, 24 และ 25 เนื่องจากชุดข้อมูลจากโพรโตคอล SNMP ที่ส่งมายัง Monitoring Server โดยที่ค่า CPU Usage เป็นข้อมูล ในบรรทัดแรกๆ จึงทำให้กราฟ CPU Usage นั้นแสดงค่าได้มากกว่า รองลงมาเป็นค่าเฉลี่ยภาระงาน และ การใช้งาน Bandwidth ซึ่งเป็นข้อมูลในบรรทัดท้ายๆ ซึ่งในช่วงเวลาที่ภาระการทำงานของ CPU ทำงานหนัก ข้อมูลของ SNMP จะทำการหยุดส่งข้อมูลทันที ทำให้ข้อมูลของ SNMP นั้นจะได้รับไม่ครบถ้วน ในบางช่วงเวลา

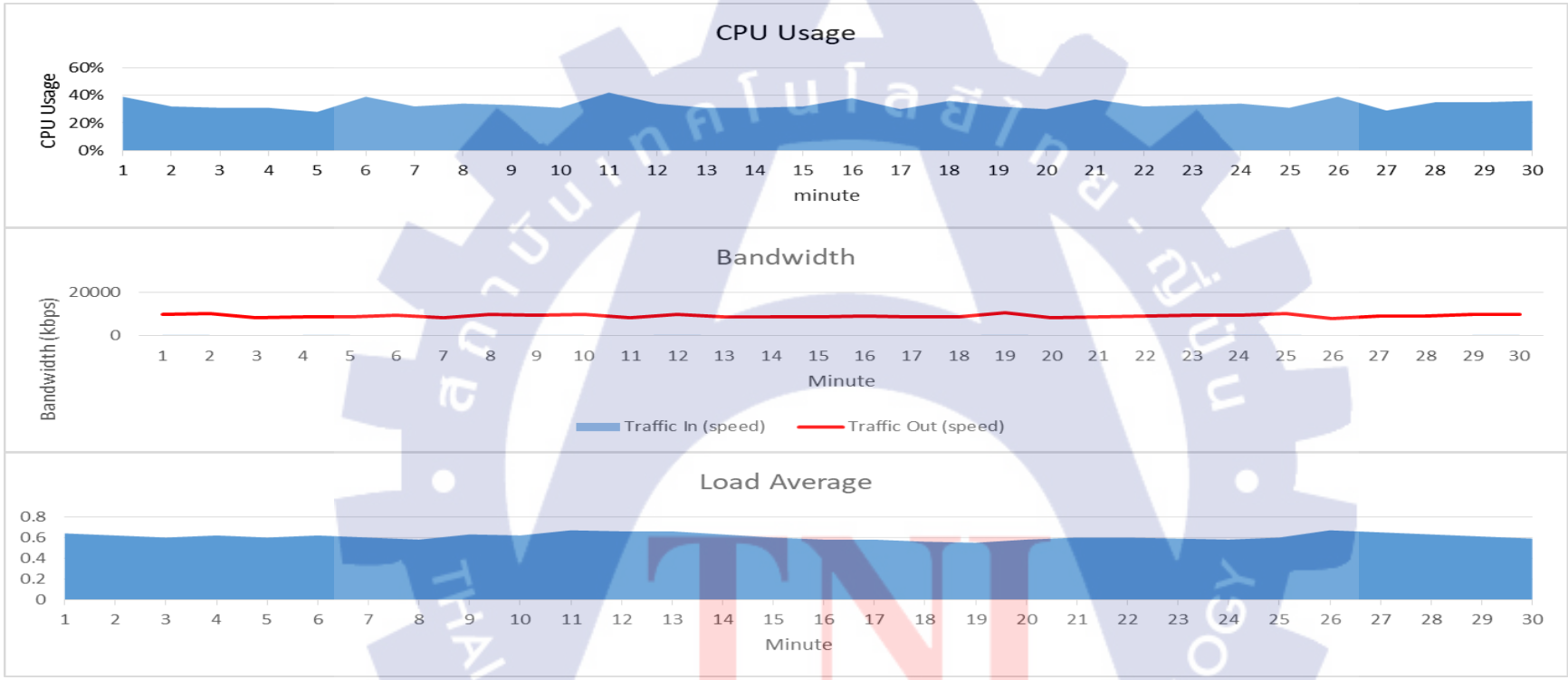
ส่วนกราฟ Interface Bandwidth สังเกตเห็นว่ากราฟจะมีการขาดตอนเช่นเดียวกับกราฟของ CPU Usage ส่วนสาเหตุที่เกิดขึ้นนั้น เนื่องจากการทำงานของ CPU นั้นทำงานเกินขีดความสามารถ เช่นเดียวกับการทดลองการทดลองส่งไฟล์ด้วย FTP 1 ครั้ง ของ IPFire ส่วนอัตราการส่งข้อมูลมากที่สุดอยู่ที่ 18,000 kbps (18 Mbps) ซึ่งเป็น bandwidth สูงสุดที่ ณ ขนาดนั้น

เนื่องจากการใช้งาน CPU ที่สูงผิดปกติ และแบนด์วิธที่ใช้งานเกินขีดสูงสุด ทำให้ค่าเฉลี่ยภาระงานของหน่วยประมวลผลมีค่าเกินจำนวนของ core ที่มีอยู่เป็นปริมาณมากค่าสูงสุดของค่าเฉลี่ยภาระงานของการทดลองนี้อยู่ที่ 3.8

จากการทดลองนี้เนื่องจากการทำงานของการส่ง FTP ของ IPFire นั้น ทำงานมากจนเกิดขีดความสามารถของ Raspberry Pi ทำให้การทำงานของ CPU นั้น มีค่า 100% ตลอดการทำงาน และในบางช่วงเวลา raspberry Pi ไม่สามารถส่งค่าทั้งการทำงานของ CPU แบนด์วิธที่ใช้งาน และค่าเฉลี่ยภาระของหน่วยประมวลผล มายัง Monitoring Server ได้ ทำให้กราฟนั้นมีการขาดหายไปบางช่วง ซึ่งอาจจะสรุปได้ว่าการใช้งานของ IPFire อาจจะไม่เหมาะสมกับการใช้งานประเภทนี้ เมื่อเทียบกับ Open Source Firewall ประเภทอื่นๆ ในงานวิจัยนี้

2. การทดสอบประสิทธิภาพในการเข้าถึงข้อมูลเว็บไซต์

ก. การทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยไม่มีการปิดกั้นใดๆ ทั้งสิ้น



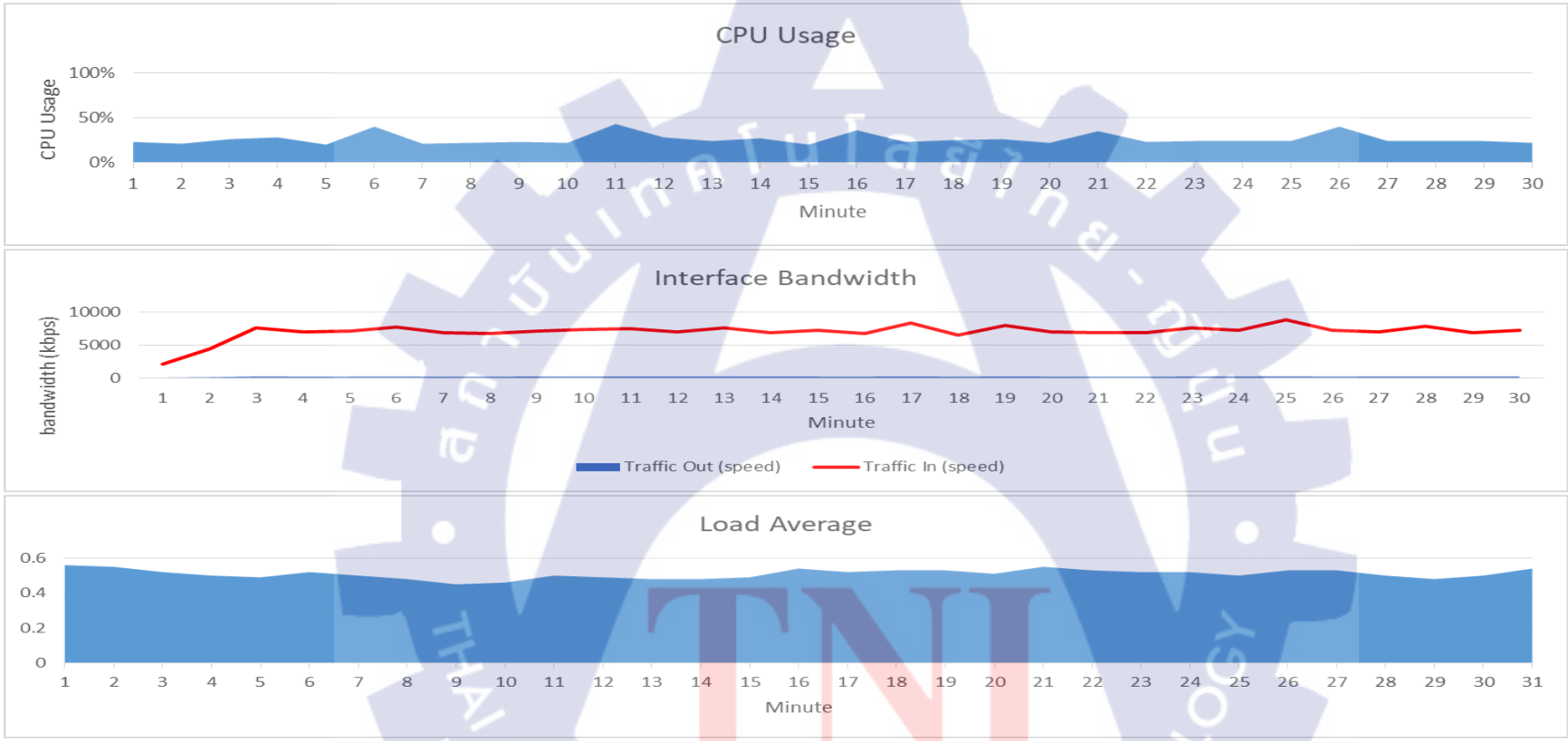
รูปที่ 4.12 กราฟแสดงผลการประสิทธิภาพการเข้าถึงเว็บไซต์โดยไม่มีการปิดกั้นใดๆ ทั้งสิ้น ด้วย โพรโตคอล HTTP และ HTTPS ของ IPFire

จากรูปที่ 4.12 แสดงการทำงานของการใช้งานในหน่วยประมวลผล CPU ของการเข้าถึงเว็บไซต์ จำนวน 10 เว็บไซต์ การใช้งาน CPU โดยเฉลี่ย 33% ส่วนการใช้งานแบนด์วิธโดยเฉลี่ย 8,900 kbps (8.9 Mbps) ซึ่งเป็น ปริมาณของ content ทั้งหมด 10 เว็บไซต์มารวมกัน โดยที่การทำงานนั้นสอดคล้องกับกราฟของ CPU Usage ในรูปที่ 4.12 เนื่องจาก bandwidth การใช้งานไม่เยอะมาก ทำให้การใช้งาน CPU ไม่ได้มากขึ้น

ในส่วนของค่าเฉลี่ยภาระงานของหน่วยประมวลผล เนื่องจากค่าของ CPU อยู่ในระดับที่ไม่ถึง 100% ทำให้กราฟค่าเฉลี่ยของหน่วยประมวลผลมีค่าไม่เกิน 1 ซึ่งแสดงว่า process ในการทำงานการเข้าถึงเว็บไซต์นั้น มี process ที่ค้างอยู่เป็นจำนวนไม่มาก เพราะเนื่องจากอัตราการส่ง และขนาดของข้อมูลของการทดลอง ใช้ปริมาณน้อย และ ค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 0.67

ทั้งนี้การที่กราฟ Interface Bandwidth ในรูปที่ 4.12 มีค่าไม่คงที่ ขึ้นอยู่กับปริมาณ content ของในแต่ละเว็บไซต์ รวมถึงในช่วงจังหวะที่ทำการ refresh หน้าเว็บไซต์ ซึ่งในบางครั้งอาจจะดึงข้อมูลที่อยู่ในเว็บไซต์ไม่เหมือนกันยกตัวอย่างเช่น หน้าโฆษณา เป็นต้น

ข. การทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยมีการปิดกั้นเว็บไซต์บางส่วน



รูปที่ 4.13 กราฟแสดงผลการประสิทธิภาพการเข้าถึงเว็บไซต์โดยมีการปิดกั้นเว็บไซต์บางส่วนด้วยโพรโตคอล HTTP และ HTTPS ของ IPFire

การทดลองนี้เป็นการแสดงการทำงานของการใช้งานในหน่วยประมวลผล CPU ของการเข้าถึงเว็บไซต์ จำนวน 10 เว็บไซต์ และทำการปิดกั้นเว็บไซต์จำนวน 5 เว็บไซต์ ซึ่งกล่าวไว้ในการทดลองของ Open Source Firewall ที่แล้ว โดยค่า CPU Usage ของการทดลองนี้โดยเฉลี่ยอยู่ที่ 27%

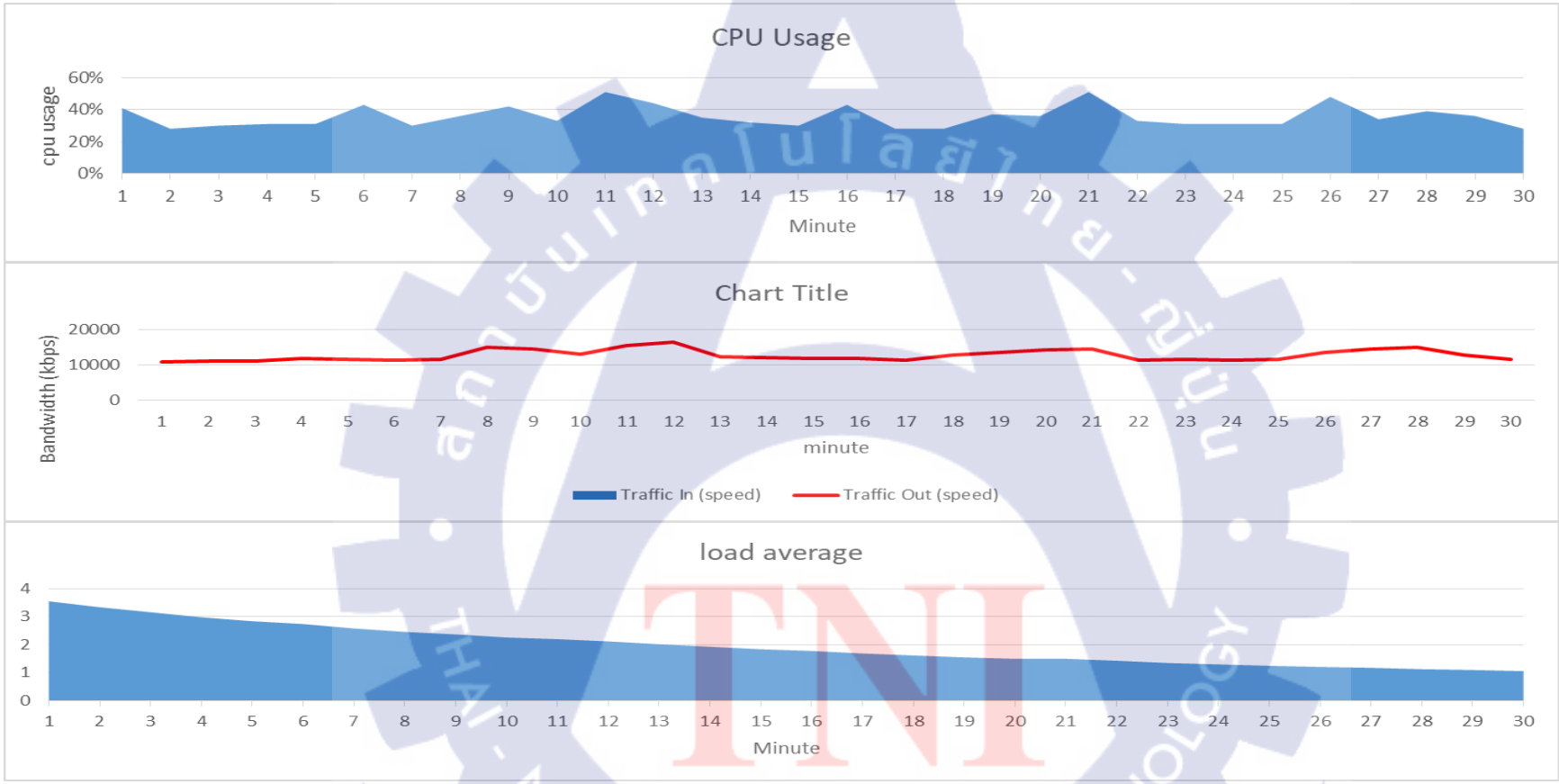
ส่วนของการทำงานของการใช้แบนด์วิธ ของการเข้าถึงเว็บไซต์ เมื่อทำการปิดกั้นเว็บไซต์ไปบางส่วน การใช้งานแบนด์วิธโดยเฉลี่ย 7,000 kbps (7Mbps) และค่าสูงสุดของ Interface Bandwidth อยู่ที่ 8,810 kbps (8.81 Mbps) ซึ่งเป็น ปริมาณของ content ทั้งหมดที่ไม่ได้ทำการปิดกั้น มารวมกัน โดยที่การทำงานนั้นสอดคล้องกับการทำงานของ CPU ในกรณีที่ bandwidth การใช้งานไม่เยอะมาก ทำให้การใช้งาน CPU ไม่ได้มากขึ้นไป

ในส่วนของค่าเฉลี่ยภาระงานของหน่วยประมวลผล เนื่องจากค่าของ CPU อยู่ในระดับที่ไม่ถึง 100% ทำให้กราฟค่าเฉลี่ยของหน่วยประมวลผลมีค่าไม่เกิน 1 ซึ่งแสดงว่า process ในการทำงานการเข้าถึงเว็บไซต์นั้น มี process ที่ค้างอยู่เป็นจำนวนไม่มาก เพราะเนื่องจากอัตราการส่ง และขนาดของข้อมูลของการทดลอง ใช้ปริมาณน้อยกว่า ส่วนค่าเฉลี่ยของภาระงานของการทดลองนี้อยู่ที่ 0.5 และ ค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 0.55

โดยจากการทดลองนี้ผลที่ได้ออกมาจากการที่ทำการปิดกั้นจำนวน 5 website ทำการปิดกั้นสำเร็จทั้งหมด โดยไม่มี content ของเว็บไซต์ที่ทำการปิดกั้นผ่านเข้ามาใน Raspberry Pi Firewall และ เนื่องจากทั้งนี้การที่กราฟ Interface Bandwidth ในรูปที่ 4.13 นั้นมีค่าไม่คงที่ ขึ้นอยู่กับปริมาณ content ของในแต่ละเว็บไซต์ รวมถึงในช่วงจังหวะที่ทำการ refresh หน้าเว็บไซต์ ซึ่งในบางครั้งอาจจะดึงข้อมูลที่อยู่ในเว็บไซต์ไม่เหมือนกัน ยกตัวอย่างเช่น หน้าโฆษณา เป็นต้น

สรุปแล้วจากการทดสอบของการเข้าถึงเว็บต์ทั้ง 2 การทดสอบ ในรูปที่ 4.12 และ 4.13 นั้น การทดสอบการเข้าถึงเว็บไซต์แบบไม่มีการปิดกั้นใดๆ เป็นการทดสอบที่ใช้ CPU Usage, Interface Bandwidth และค่าเฉลี่ยภาระงานที่มากกว่า การทดสอบการเข้าถึงเว็บไซต์แบบทำการปิดกั้น เนื่องจาก content ที่อนุญาตผ่านไปยัง Raspberry Pi Firewall ที่มีมากกว่า ซึ่งในการทดสอบนี้สรุปได้ว่า ปริมาณของ rule ที่ทำการปิดกั้นนั้น มีจำนวนไม่มาก แต่ สาเหตุหลักที่การใช้งาน CPU มีค่าที่สูงเพราะมาจากปริมาณ content ที่อนุญาตผ่านจาก Raspberry Pi firewall มีผลมากกว่าจำนวน rule ของ firewall ที่ทำการปิดกั้น

3. การทดสอบประสิทธิภาพในการรับและส่ง email ด้วยโพรโตคอล SMTP



รูปที่ 4.14 กราฟแสดงผลการประสิทธิภาพในการรับส่ง email ด้วย โพรโตคอล SMTP ของ IPFire

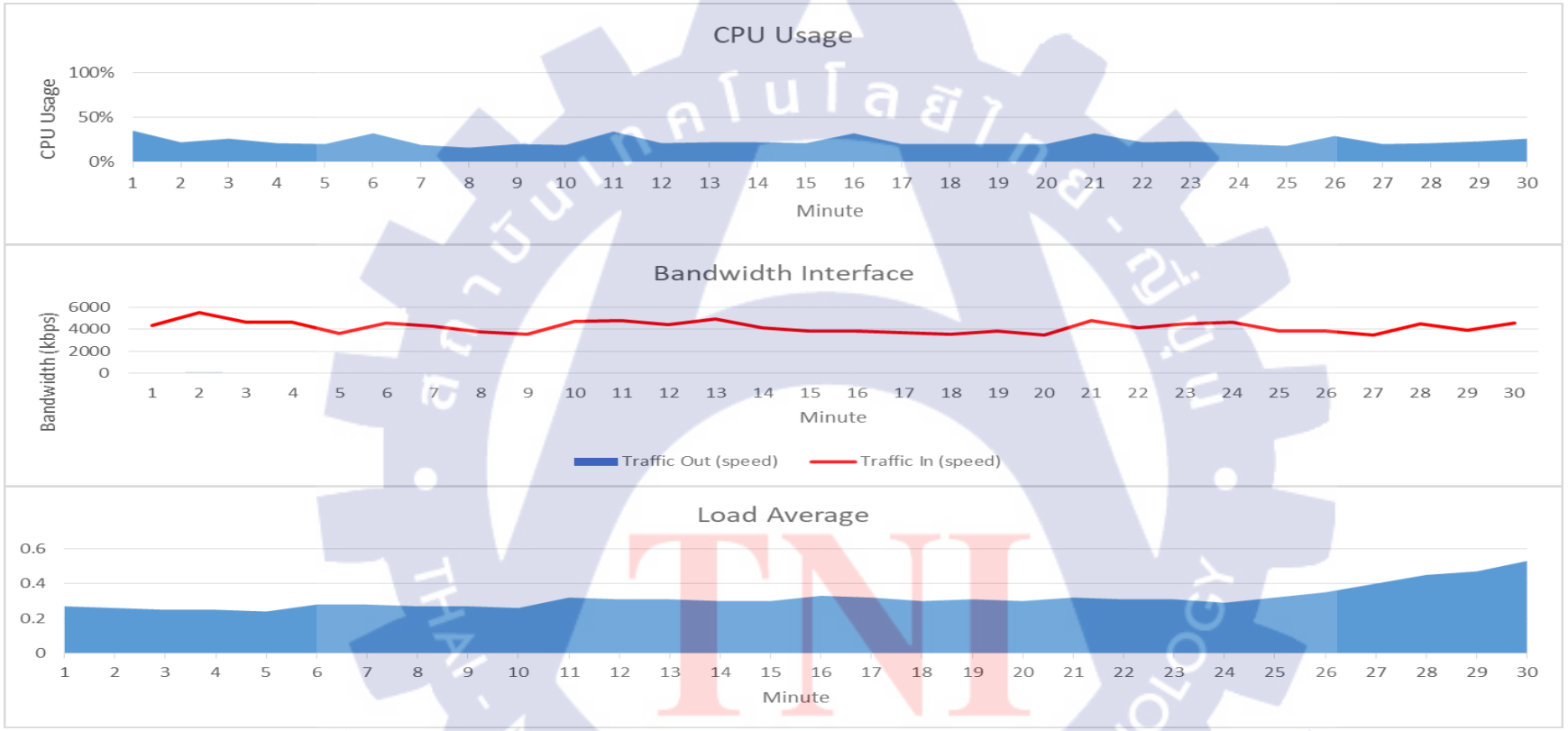
เนื่องด้วยการทำงานของ email ตามที่กล่าวไว้แล้วในบทที่ 3 จะเป็นการส่งข้อมูลเพียง 10MB ต่อ 1 วินาที ทำให้ โดยที่ค่าเฉลี่ยของ CPU Usage อยู่ที่ประมาณ 39% ซึ่งในบางครั้งจะมีบางช่วงที่ไม่สามารถส่งได้ตามที่ตั้งค่าเอาไว้คือ 10MB เนื่องจากความเสถียรของ Internet ในช่วงเวลานั้น และจะมีการทบการส่งไปยังช่วงวินาทีต่อไป ทำให้กราฟ Interface Bandwidth ในรูปที่ 4.14 ไม่ได้คงที่เสมอไป

สำหรับแบนด์วิธที่ใช้งานนั้นจะสอดคล้องกับคำอธิบายของการใช้งาน CPU ก่อนหน้านี้ โดยในการทดลองจะเป็นการส่งข้อมูลจำนวน 10MB ทุกๆ 1 วินาที ซึ่งจากรูปที่ 4.14 IPFire สามารถทำงานได้ และ Raspberry Pi สามารถส่ง email ได้ตลอดทุกช่วงเวลา โดยที่การใช้งานแบนด์วิธการใช้งานสูงสุดอยู่ที่ 16,440 kbps (16.44 Mbps) และจากค่าเฉลี่ยภาระงานในการทำงานของ CPU สูงสุดอยู่ที่ 1.96

จากผลการทดลองดังกล่าวสามารถสรุปได้ว่า IPFire สามารถใช้งานในด้านการรับและส่ง email ได้ เพราะว่า แบนด์วิธการใช้งาน email นั้นไม่ได้สูงมาก ทำให้ Raspberry pi firewall ยังสามารถทำงานได้อย่างเต็มประสิทธิภาพ แต่ในช่วงเวลาในการส่ง email จะมีการส่งได้ไม่ครบ 10MB ในช่วงวินาทีนั้น และจะยกยอดการส่งไปยังวินาทีต่อไป จึงทำให้มี Bandwidth บางช่วงสูงกว่า 10Mbps และจากประสิทธิภาพในภาพรวมดังกล่าวยังสามารถรองรับการใช้งานอื่นๆ ได้อีก โดยดูจากประสิทธิภาพการใช้งาน CPU เป็นหลัก

4. การทดสอบประสิทธิภาพการใช้งาน Live Streaming

ก. การทดสอบประสิทธิภาพการใช้งาน Live Streaming โดยไม่ได้มีการปิดกั้นใดๆ



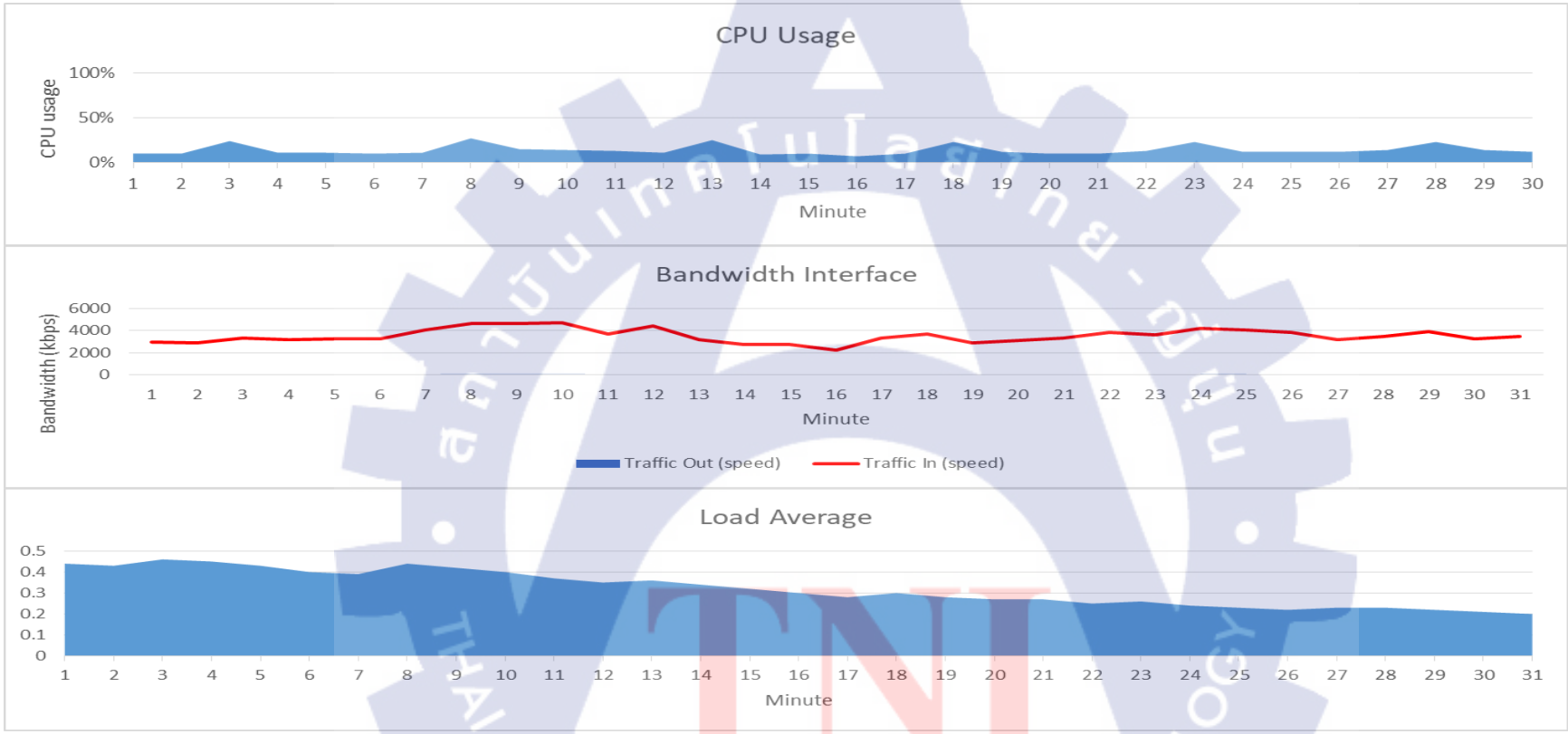
รูปที่ 4.15 กราฟแสดงผลประสิทธิภาพการใช้งาน Live Streaming แบบไม่ได้ทำการปิดกั้นของ IPFire

ผลการทดสอบแบบ Live Streaming ดังรูปที่ 4.15 นี้เป็นผลการทดสอบ Live Streaming โดยทำการเปิดช่องทีวี 3 ช่องพร้อมกันด้วยความละเอียด 720p ซึ่งค่าเฉลี่ยของการทดสอบนี้อยู่ที่ 23% และค่าสูงสุดของการใช้งาน CPU อยู่ที่ 35% ซึ่งสอดคล้องกับ Bandwidth ที่ใช้งานโดยเฉลี่ย 1 ช่องจะใช้ CPU โดยประมาณ 7.6%

ส่วนค่าสูงสุดของการใช้งานแบนด์วิธ อยู่ที่ 10,550 kbps (10.55 Mbps) นั่นหมายความว่าโดยเฉลี่ย 1 ช่องจะใช้แบนด์วิธ โดยประมาณ 3,510 kbps (3.51 Mbps) ซึ่งตรงตามหลักการของการ streaming ในช่วงความละเอียด 720p จะต้องใช้แบนด์วิธอย่างน้อยต้องอยู่ในช่วง 2-4 Mbps ทั้งนี้ขึ้นอยู่กับ content และภาพที่ส่งมาในช่วงเวลานั้นๆ ว่า ในรายการนั้นๆ ส่งภาพมาในรูปแบบใด โดยในแต่ละช่วงเวลาจะมีขนาดของ content ไม่เท่ากัน

ส่วนค่าสูงสุดของค่าเฉลี่ยภาระงาน อยู่ที่ 0.67 ซึ่งสอดคล้องกับการใช้งาน CPU ที่มีค่าไม่เกิน 1 นั่นหมายความว่าการทำงานของ Live Streaming ของ IPFire ยังสามารถทำงานได้อย่างเต็มประสิทธิภาพ และสามารถรองรับการใช้งานอื่นๆ ได้

ข. การทดสอบประสิทธิภาพการใช้งาน Live Streaming โดยมีการปิดกั้นบางช่อง



รูปที่ 4.16 กราฟแสดงผลประสิทธิภาพการใช้งาน Live Streaming โดยมีการปิดกั้นบางช่อง ของ IPFire

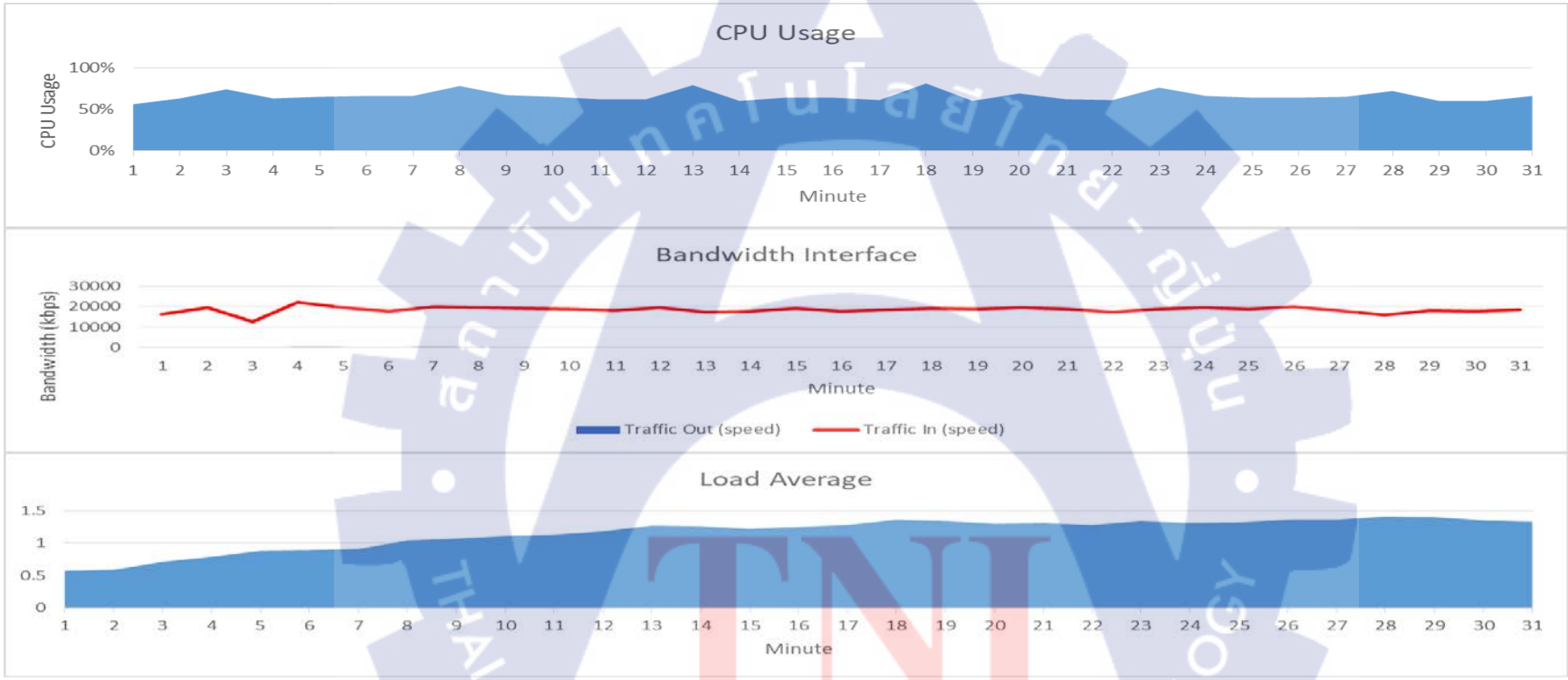
ผลการทดสอบแบบ Live Streaming นี้เป็นผลการทดสอบ Live Streaming โดยทำการเปิดช่องทีวี 3 และทำการปิดกั้นด้วย firewall จำนวน 1 ช่องด้วยความละเอียด 720p ซึ่งค่าเฉลี่ยของการใช้งาน CPU อยู่ที่ 14% และค่าสูงสุดของการใช้งาน CPU อยู่ที่ 27% ซึ่งสอดคล้องกับ Bandwidth ที่ใช้งานโดยเฉลี่ย 1 ช่องเหมือนที่กล่าวไปในกรณีเปิดพร้อมกัน 3 ช่อง แต่ว่าในการทดลองนี้จะเหลือ traffic ที่ผ่านได้จำนวน 2 ช่องเท่านั้น ค่าการใช้งาน CPU จึงลดลง โดยทั้งนี้ขึ้นอยู่กับ content และภาพที่ส่งมาในช่วงเวลานั้นๆ ว่าในรายการนั้นๆ ส่งภาพมาในรูปแบบใด โดยในแต่ละช่วงเวลาจะมีขนาดของ content ไม่เท่ากัน

ค่าสูงสุดของการใช้งานแบนด์วิธ อยู่ที่ 4,730 kbps (4.73 Mbps) ที่ใช้งานโดยเฉลี่ย 1 ช่องเหมือนที่กล่าวไปในกรณีเปิดพร้อมกัน 3 ช่อง แต่ว่าในการทดลองนี้จะเหลือ traffic ที่ผ่านได้จำนวน 2 ช่องเท่านั้น ซึ่งผลที่ได้นั้นก็สอดคล้องกับการใช้งาน CPU ที่ลดลงไปเพราะมีการถูกปิดกั้น traffic ด้วย

ค่าสูงสุดของค่าเฉลี่ยภาระงาน อยู่ที่ 0.46 ซึ่งสอดคล้องกับการใช้งาน CPU และ แบนด์วิธที่ลดลง และการใช้งาน CPU ที่มีค่าไม่เกิน 1 นั่นหมายความว่าการทำงานของ IPFire ยังสามารถทำงานได้อย่างเต็มประสิทธิภาพ และสามารถรองรับการใช้งานอื่นๆ ได้อีก

ซึ่งจากการทดสอบการใช้งาน Live Streaming จากรูปที่ 4.15 และรูปที่ 4.16 กรณีที่ไม่ได้ทำการปิดกั้นช่องจะใช้ประสิทธิภาพการใช้งาน CPU แบนด์วิธ และค่าเฉลี่ยภาระงานที่มากกว่าทั้งนี้ก็เพราะว่าปริมาณ content ทั้งหมดของกรณีแรกมีจำนวนมากกว่า และการประมวลผลการตรวจสอบ Rule ของ firewall ก็ไม่ได้ส่งผลต่อการใช้งานประสิทธิภาพมากนักจนสังเกตเห็นความแตกต่างได้

5. การทดสอบประสิทธิภาพการทำงานพร้อมกันของแอพพลิเคชั่นมากกว่า 1 ประเภท ได้แก่ การเข้าถึง website, การส่ง email และ live streaming



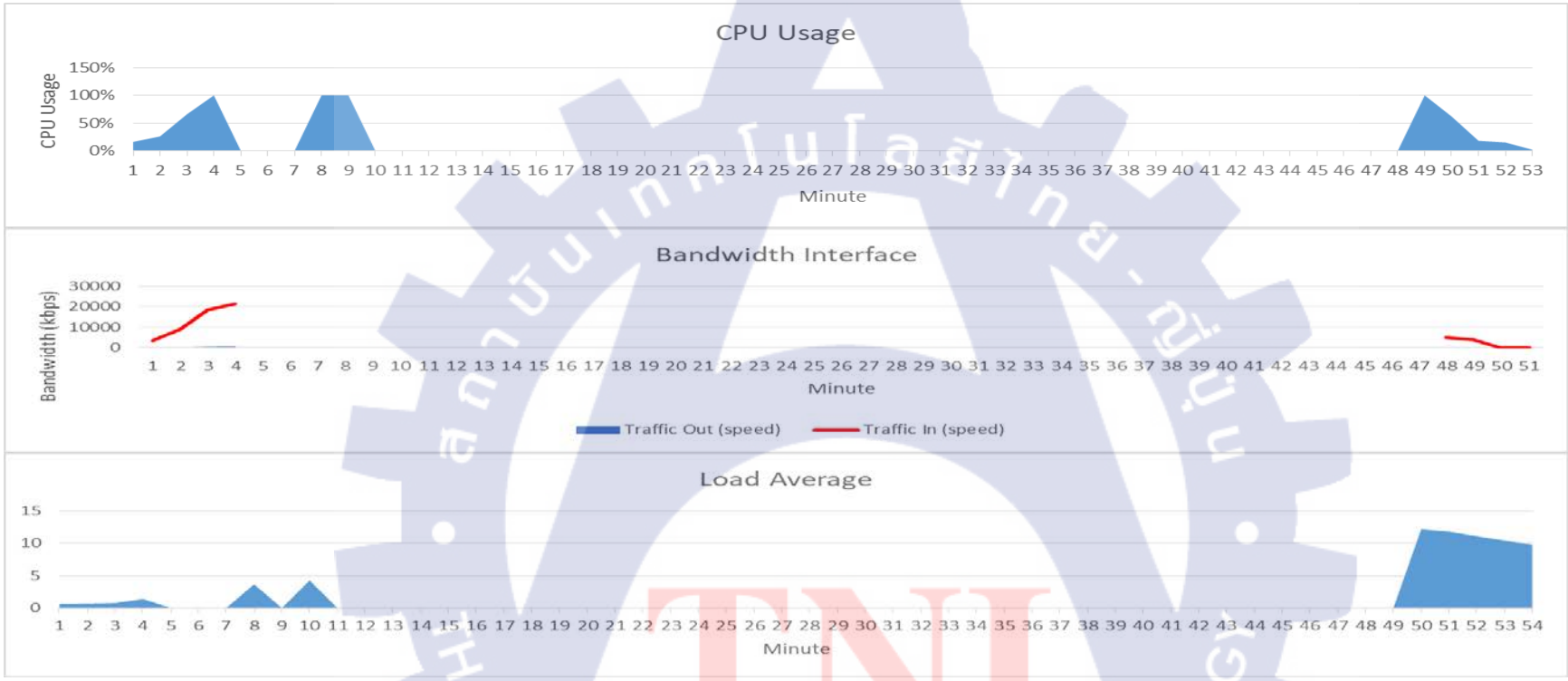
รูปที่ 4.17 กราฟแสดงผลประสิทธิภาพ การทำงานแบบแอพลิเคชั่นมากกว่า 1 อย่างของ IPFire

จากรูปที่ 4.17 เป็นกราฟการทดสอบทำงานแบบหลายแอพพลิเคชั่น โดยอ้างอิงจากสถิติ ได้แก่ การใช้งานการเข้าถึงเว็บไซต์ การส่ง email และ Live Streaming ซึ่งค่าเฉลี่ยของการใช้งาน CPU อยู่ที่ 57% โดยค่าสูงสุดของการทดลองนี้อยู่ที่ 66% แบนด์วิธสูงสุดในการใช้งานอยู่ที่ 22,070 kbps (22.07 Mbps) โดยในช่วงนาทีที่ 3 มีการข้อข้องในการ download website ทำให้กราฟการใช้งานลงไปที่ 12,810 kbps (12.81 Mbps) และกลับมา download ได้ปกติอีกครั้งในนาทีที่ 4 หลังจากนั้น ค่าดังกล่าวจะอยู่ในช่วงค่าเฉลี่ยการใช้งาน ที่ 18,480 kbps (18.48 Mbps)

ส่วนกราฟค่าเฉลี่ยภาระงานในรูปที่ 4.17 จะสังเกตเห็นว่าค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 1.41 ซึ่งเมื่อค่าเฉลี่ยภาระงานที่เกิน 1 นั่นหมายความว่ามี process ที่มีค้างสะสมอยู่ในช่วงเวลาสั้นๆ แต่ในแง่ประสิทธิภาพการทำงาน ยังสามารถทำงานได้ดี เพราะว่า CPU Usage ที่ใช้งานยังไม่ถึง 100%

สำหรับการทดสอบนี้ สามารถใช้งานได้เต็มประสิทธิภาพ สามารถส่ง email ได้ตลอดช่วงเวลา เข้าถึงเว็บไซต์ที่ทำการเปิดให้ใช้งานได้ รวมถึงสามารถใช้งาน Live Streaming พร้อมกันได้โดยไม่มีการติดขัดระหว่างรับชมรายการ

6. การทดสอบประสิทธิภาพการทำงานแบบแอพลิเคชั่นทุกอย่างเข้าด้วยกันทั้ง FTP การเข้าถึงเว็บไซต์ Email และ Livestreaming



รูปที่ 4.18 กราฟแสดงผลการทดสอบประสิทธิภาพการทำงานแบบแอพลิเคชั่นทุกอย่างเข้าด้วยกัน ของ IPFire

การทดสอบดังกล่าวนี้ ผลลัพธ์ที่ได้จะคล้ายคลึงกับการใช้งาน FTP เพียงอย่างเดียว ในรูปที่ 4.10 และ 4.11 โดยมีบางช่วงเวลาไม่สามารถดึงค่าออกมาได้ เนื่องด้วยมีการใช้งานอย่างอื่นเข้ามาผสมอยู่ด้วย ทำให้การใช้งาน CPU และค่าเฉลี่ยภาระงาน มีค่าสูงมาก และทำให้ไม่สามารถดึงค่าออกมาได้ในช่วงเวลาที่ใช้งานสูงๆ รวมถึงทำให้ การใช้งานทั้งเว็บไซต์ อีเมล์ live streaming ทำงานได้อย่างไม่เต็มประสิทธิภาพ เพราะมี traffic จาก FTP ดึงไปเป็นจำนวนมาก ซึ่งการใช้งาน FTP ทำการ upload ไฟล์เสร็จสิ้นเป็นเวลา 41 นาที ซึ่งแตกต่างจากการใช้งาน FTP เพียงอย่างเดียว ซึ่งในรูปที่ 43 ใช้เวลาเพียง 27 นาทีเท่านั้น

เช่นเดียวกันกราฟ Interface Bandwidth ในรูปที่ 4.18 มีบางช่วงเวลาไม่สามารถดึงค่าออกมาได้ โดยที่ในช่วงนาทีที่ 4 ที่แบนด์วิธอยู่ที่ 21,250 kbps (21.25 Mbps) ซึ่งเป็นจุดสูงสุดของกราฟนี้ หลังจากนั้น Raspberry Pi ก็ไม่ได้ทำการส่งค่ามายัง monitoring server เลย จึงทำให้สามารถอ่านค่าเฉลี่ยที่แน่นอนได้ เพราะว่าการใช้งาน CPU และค่าเฉลี่ยภาระงานที่สูงเกิดขีดจำกัด

เนื่องด้วยค่าเฉลี่ยภาระงานที่มีมากเกินขีดจำกัด ทำให้การกราฟที่แสดงผลออกมา มีบางส่วนที่มีการขาดหายไป ซึ่งจากกราฟนั้น ค่าเฉลี่ยภาระงานมีค่าสูงสุดที่ 12.14 ซึ่งถือว่าค่าที่ได้นั้นมีค่าสูงมาก และทำให้การทดสอบนี้ทำงานได้อย่างไม่เต็มประสิทธิภาพ

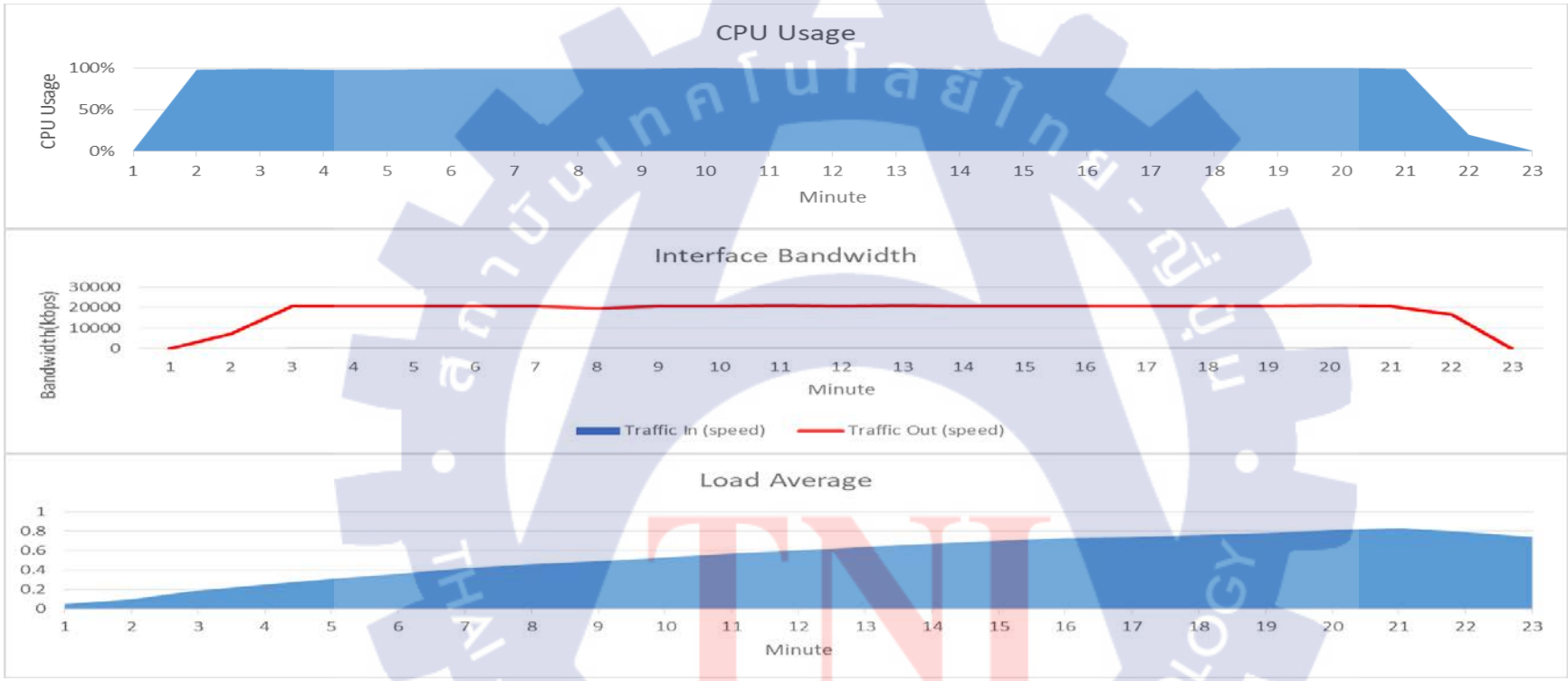
เมื่อเทียบกับการใช้งานแบบการนำแอพพลิเคชั่นมากกว่า 1 อย่างในรูปที่ 4.17 นั้น ประสิทธิภาพของการทำงานของการทดลองนี้ แย่กว่าการทดลองดังกล่าว เพราะว่าเป็นการใช้ CPU ที่สูง และค่าเฉลี่ยภาระงานที่เยอะกว่า เมื่อเทียบกับการทดลองแบบไม่มี FTP

จากการทดสอบก่อนหน้านี้ traffic ของ การทดสอบด้วย FTP แบบส่ง 3 ครั้งพร้อมกัน ของ IPFire ในรูปที่ 4.11 การใช้งาน CPU มีค่อนข้างสูง และไม่เพียงพอต่อการใช้งานอื่นๆ ตามอัตราส่วนของแอพพลิเคชั่น ทำให้การใช้งานบางอย่าง อย่างเช่น การใช้งานเว็บไซต์ที่สามารถเข้าได้ช้าบ้างในบางช่วงเวลา การใช้งาน email ที่ไม่สามารถส่งได้เต็ม 10 Mbps ในบางครั้ง และทำให้การ streaming ภาพของบางช่อง มีการขัดข้องระหว่างรับชมรายการ

การทดสอบโดยใช้ OpenWRT

1. <u>การทดสอบการส่งข้อมูลด้วย FTP</u>

   ก. การทดสอบด้วย FTP แบบส่งไฟล์ครั้งเดียว ขนาด 2GB จำนวน 1 ไฟล์



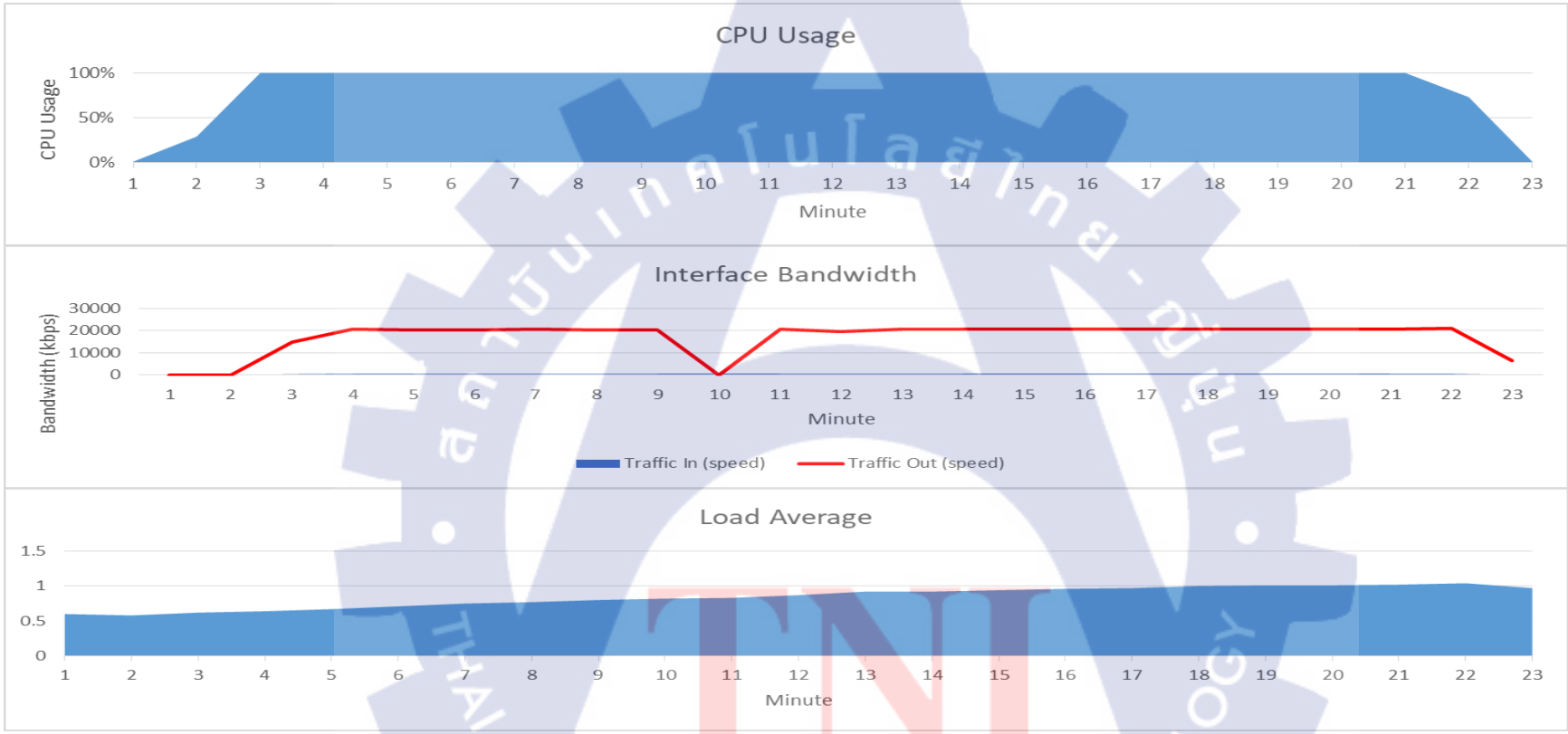
รูปที่ 4.19 กราฟแสดงผลการทดสอบประสิทธิภาพการทำงาน FTP แบบส่งครั้งเดียว 1 ไฟล์ ของ Openwrt

จากรูปที่ 4.19 เมื่อเริ่มมีการส่งไฟล์ด้วย FTP การทำงานของ CPU จะเริ่มการทำงานในช่วงนาทีที่ 3 โดยจะเพิ่มขึ้นเรื่อยๆ และมีการเพิ่มขึ้นของ CPU Usage โดยเฉลี่ย 88 % ซึ่งในกรณีนี้จะเพิ่มขึ้นตามกราฟของ Interface Bandwidth ในรูปที่ 4.19 ด้วย โดยจะสังเกตเห็นว่าการทำงานของ FTP ยังสามารถทำงานได้ตามปกติโดยที่ การใช้งานของ CPU มีการใช้งานถึง 100% ในบางช่วงเวลาส่วนกราฟ Interface Bandwidth ในรูปที่ 4.19 สอดคล้องกับการทำงานของ CPU เมื่อแบนด์วิธการใช้งานเพิ่มขึ้น CPU Usage จะทำงานสูงขึ้นตามลำดับ โดยเฉลี่ยแบนด์วิธที่ใช้งานสำหรับการส่ง FTP ในการทดลองนี้ สามารถรับส่งได้สูงสุด 20,000 kbps (20 Mbps)

การใช้งานค่าเฉลี่ยภาระในการประมวลผลซึ่งจะแสดงการทำงานภาระการทำงานย้อนหลังของ CPU 15 นาที เพื่อเห็นว่าในช่วงเวลาดังกล่าวมี process ที่รอประมวลผลอยู่มากน้อยเพียงใด และส่งผลทำให้กราฟของ CPU Usage ในรูปที่ 4.19 ทำงานหนักขึ้น จากกราฟดังกล่าว แสดงให้เห็นว่า Load ของการทำงานโดยเฉลี่ยไม่ได้ทำงานหนักมากจนเกินไป โดยที่ค่าสูงสุดของ ค่าเฉลี่ยภาระในการประมวลผลนั่น อยู่ที่ประมาณ 0.83 ซึ่งสอดคล้องกับการทำงานและประสิทธิภาพของ CPU

จากการทดลองขั้นต้นเป็นผลของการทดสอบการส่งด้วยโพรโตคอล FTP ทั้งการใช้งาน CPU แบนด์วิธที่ใช้งานและค่าเฉลี่ยภาระของหน่วยประมวลผล ซึ่งในช่วงที่กำลังส่งข้อมูลอยู่ แบนด์วิธที่ใช้งาน เพิ่มมากขึ้น จนถึงขีดสูงสุดของอุปกรณ์ที่รับได้ รวมทั้งการใช้งาน CPU ยังไม่ถึง 100% ตลอดช่วงเวลา และ ทำการส่งไฟล์เสร็จสมบูรณ์ด้วยเวลา 20 นาที

ข. การทดสอบด้วย FTP แบบส่ง 3 ครั้งพร้อมกัน โดยทำการแบ่งไฟล์ 2GB ออกเป็น 3 ไฟล์ และทำการส่งพร้อมกันด้วย FTP



รูปที่ 4.20 กราฟแสดงผลการทดสอบประสิทธิภาพการทำงาน FTP แบบส่งพร้อมกัน 3 ไฟล์ ของ Openwrt

จากรูปที่ 4.20 เมื่อเริ่มมีการส่งไฟล์ด้วย FTP การทำงานของ CPU มีการใช้งานเฉลี่ย 100% ซึ่งในกรณีนี้มาจากการทำงานของ session ที่เพิ่มมากขึ้น และทำงานพร้อมกัน ทำให้มีการใช้งานของ CPU ที่สูง โดยเฉลี่ยแล้ว แบนด์วิธที่ใช้งานสำหรับการส่ง FTP ใน OpenWRT ของการทดสอบนี้ สามารถรับส่งได้สูงสุด 21,090 kbps (21.09 Mbps) และทำการส่งไฟล์เสร็จสมบูรณ์ด้วยเวลา 18 นาที

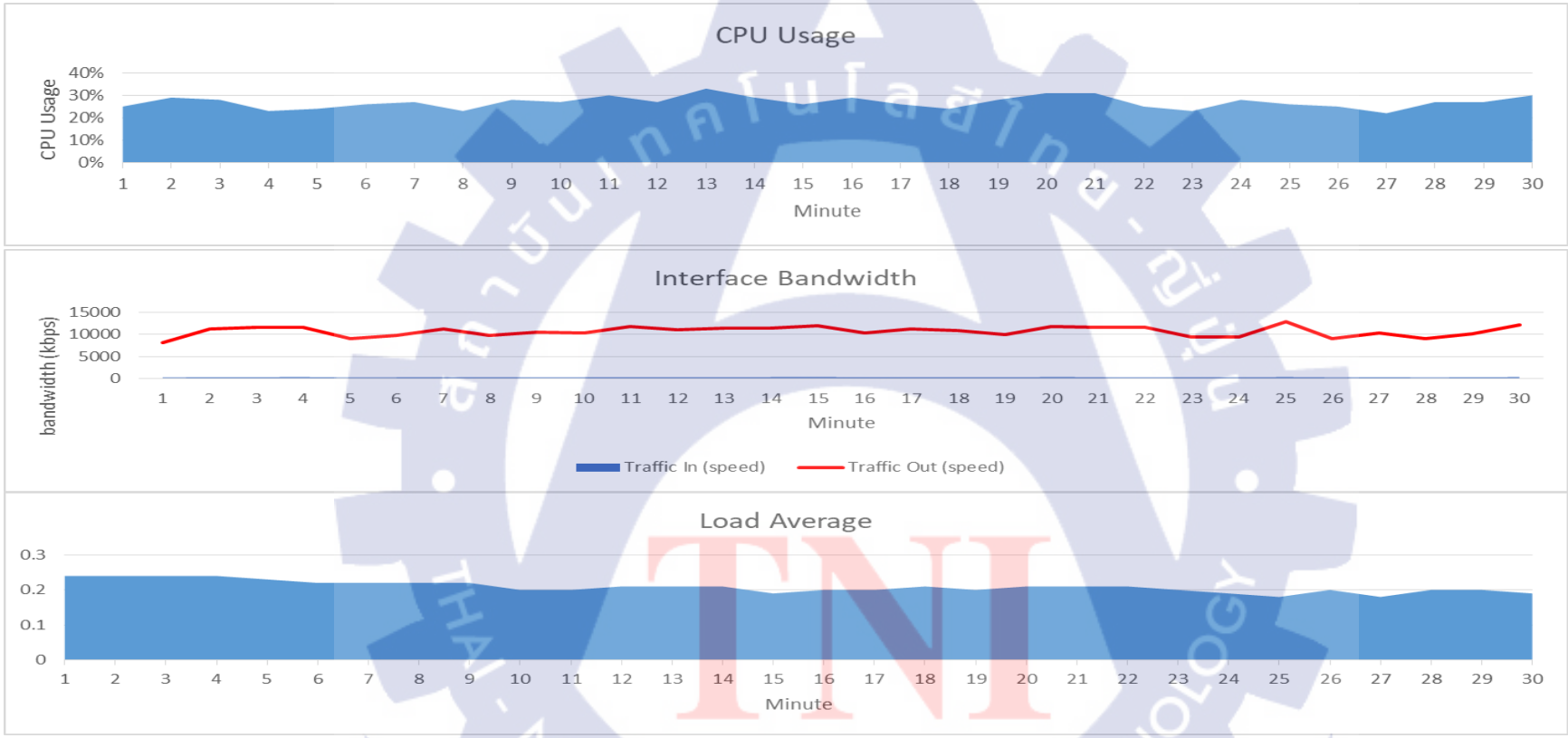
ค่าเฉลี่ยภาระการประมวลผลของรูปที่ 4.20 เมื่อเทียบกับกรณีส่งไฟล์เดียวในกราฟค่าเฉลี่ยภาระงานในรูปที่ 4.19 จะมีค่ามากขึ้น จึงทำให้ CPU Process นั้นมีค่ามากขึ้นด้วย แต่ทั้งนี้การทำงานยังสามารถทำงานได้ตามปกติ ไม่มีการสูญหายข้อมูล เพียงแต่เป็นการรอการทำงานของ process นั้น เนื่องจากมี session การทำงานที่เพิ่มขึ้น

ในกรณีที่มีการส่ง FTP พร้อมกัน 3 ครั้งจะทำให้การทำงานของ CPU นั้น ทำงานหนักมากยิ่งขึ้น และในแต่ละไฟล์จะเฉลี่ย bandwidth การส่งให้เท่าๆ กันตามขีดความจำกัดของ bandwidth ของไฟร์วอลล์ที่มี และปัจจัยอื่นๆ เช่นการทำงานโหลดเฉลี่ยเมื่อมีการส่ง FTP พร้อมกัน 3 ไฟล์ จะทำให้ โหลดการทำงานได้ต่อเนื่องมากกว่า การส่งแบบไฟล์เดียวซึ่งค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 1.04

โดยจากการทดลองของ FTP ใน รูปที่ 4.19 และ รูปที่ 4.20 จะเห็นได้ว่าการใช้งานแบบส่ง 3 ไฟล์พร้อมกันจะใช้การใช้งาน CPU และค่าเฉลี่ยภาระงานมากกว่าการส่ง FTP แบบส่งครั้งเดียว 1 ไฟล์ เพราะว่านอกจากที่จะต้องใช้แบนด์วิธที่ต่อเนื่องกัน และใช้ปริมาณสูงในการทดลองแบบส่งพร้อมกัน 3 ไฟล์จะมีเซสชั่น (Session) เข้ามาเกี่ยวข้องด้วย จึงทำให้มีการใช้ปริมาณภาระงานที่เพิ่มมากขึ้น

2. การทดสอบประสิทธิภาพในการเข้าถึงข้อมูลเว็บไซต์

ก. การทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยไม่มีการปิดกั้นใดๆ ทั้งสิ้น



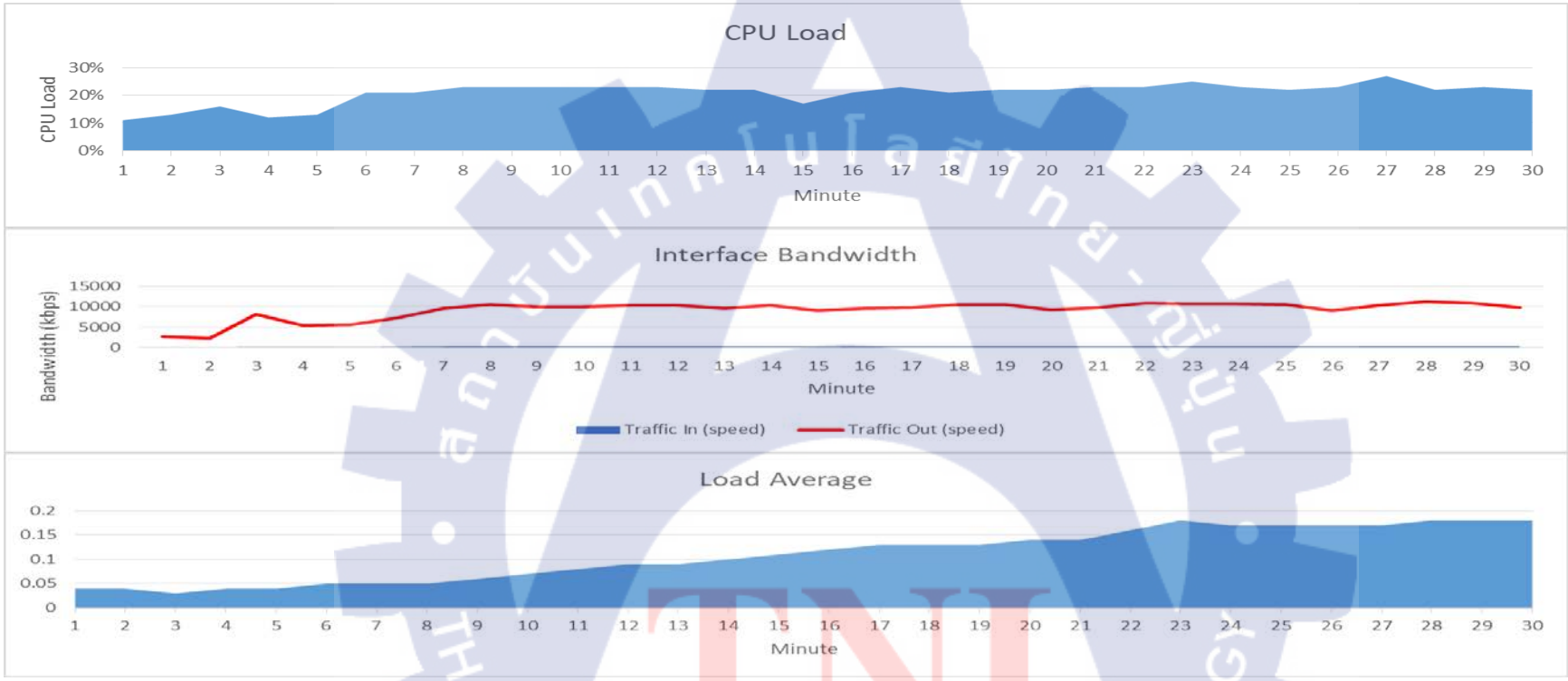
รูปที่ 4.21 กราฟแสดงผลการทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยไม่มีการปิดกั้นใดๆ ทั้งสิ้น

จากรูปที่ 4.21 จะแสดงการทำงานของการใช้งานในหน่วยประมวลผล CPU ของการเข้าถึงเว็บไซต์ จำนวน 10 เว็บไซต์ การใช้งาน CPU โดยเฉลี่ย 27% ส่วนค่า Interface Bandwidth สูงสุดอยู่ที่ 12,850 kbps (12.85 Mbps) ซึ่งเป็น ปริมาณของ content ทั้งหมด 10 เว็บไซต์มารวมกัน โดยที่การทำงานนั้นสอดคล้องกับการทำงานของ CPU เพราะว่า bandwidth มีการใช้งานไม่เยอะมาก ทำให้การใช้งาน CPU ไม่ได้มากขึ้นไป

ในส่วนของค่าเฉลี่ยภาระงานของหน่วยประมวลผล เนื่องจากค่าของ CPU อยู่ในระดับที่ไม่ถึง 100% ทำให้กราฟค่าเฉลี่ยภาระงานของหน่วยประมวลผลมีค่าไม่เกิน 1 ซึ่งแสดงว่า process ในการทำงานการเข้าถึงเว็บไซต์นั้น มี process ที่ค้างอยู่เป็นจำนวนไม่มาก เพราะเนื่องจากอัตราการส่งและ ปริมาณของข้อมูลของการทดลอง มีปริมาณที่น้อย ดังนั้นค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 0.25

ทั้งนี้การที่กราฟ Interface Bandwidth ในรูปที่ 4.21 นั้นมีค่าไม่คงที่ ขึ้นอยู่กับปริมาณ content ของในแต่ละเว็บไซต์ รวมถึงในช่วงจังหวะที่ทำการ refresh หน้าเว็บไซต์ ซึ่งในบางครั้งอาจจะดึงข้อมูลที่อยู่ในเว็บไซต์ที่ไม่เหมือนกัน ยกตัวอย่างเช่น หน้าโฆษณา เป็นต้น

ข. การทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยมีการปิดกั้นเว็บไซต์บางส่วน



รูปที่ 4.22 กราฟแสดงผลการทดสอบประสิทธิภาพการเข้าถึงเว็บไซต์โดยมีการปิดกั้นเว็บไซต์บางส่วนของ Openwrt

การทดลองนี้เป็นการแสดงการทำงานของการใช้งานในหน่วยประมวลผล CPU ของการเข้าถึงเว็บไซต์ จำนวน 10 เว็บไซต์ และทำการปิดกั้นเว็บไซต์จำนวน 5 เว็บไซต์ การใช้งาน CPU โดยเฉลี่ย 22%

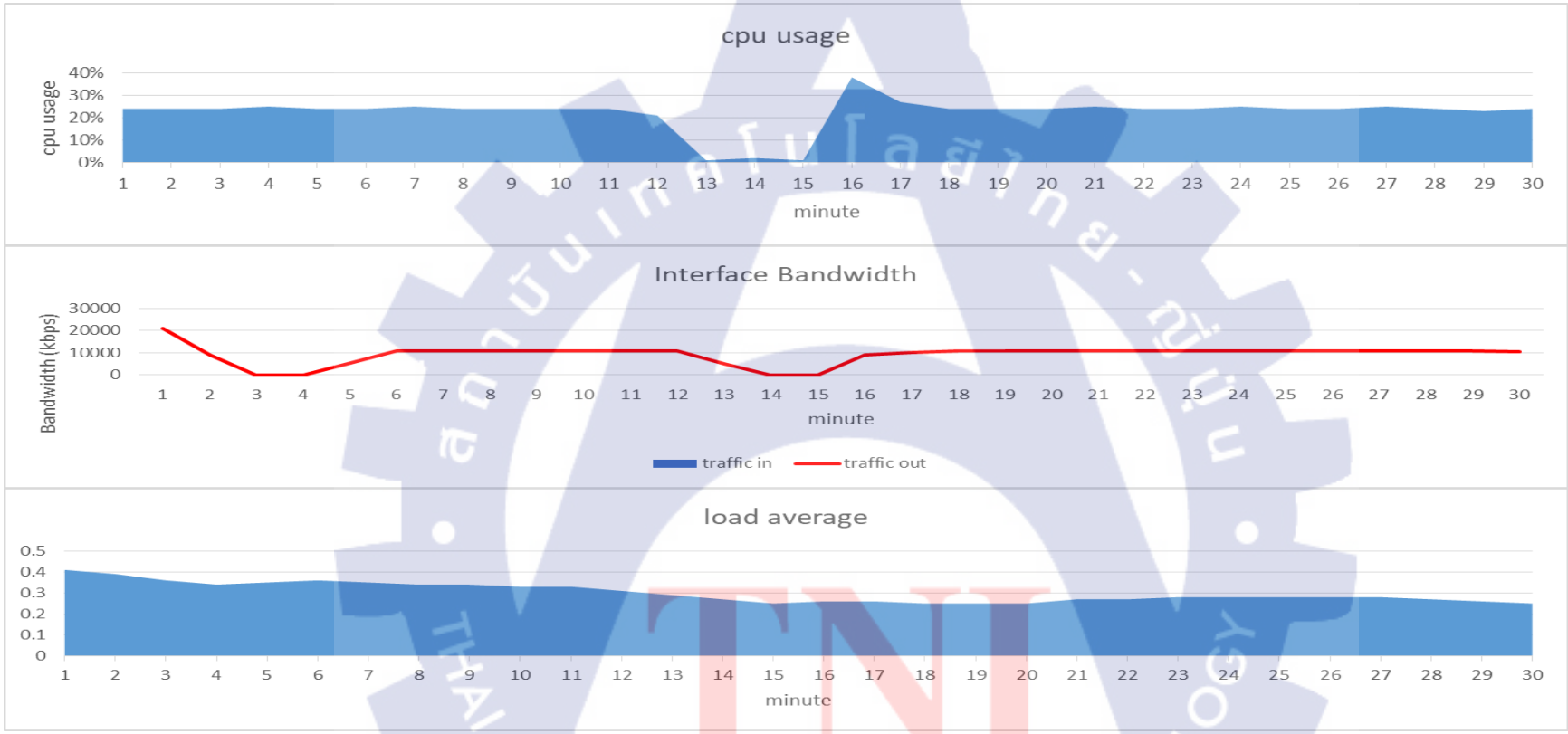
ส่วนของการทำงานของการใช้แบนด์วิธ ของการเข้าถึงเว็บไซต์ เมื่อทำการปิดกั้นเว็บไซต์ไปบางส่วน การใช้งานแบนด์วิธโดยเฉลี่ย 7,000 kbps (7 Mbps) และค่าสูงสุดของ Interface Bandwidth อยู่ที่ 11,380 kbps (11.38 Mbps) ซึ่งเป็น ปริมาณของ content ทั้งหมดที่ไม่ได้ทำการปิดกั้น มารวมกัน โดยที่การทำงานนั้นสอดคล้องกับการทำงานของ CPU เพราะว่า bandwidth มีการใช้งานไม่เยอะมาก ทำให้การใช้งาน CPU ไม่ได้มากขึ้นไป

ในส่วนของค่าเฉลี่ยภาระงานของหน่วยประมวลผล เนื่องจากค่าของ CPU อยู่ในระดับที่ไม่ถึง 100% ทำให้กราฟค่าเฉลี่ยภาระงานของหน่วยประมวลผลมีค่าไม่เกิน 1 ซึ่งแสดงว่า process ในการทำงานการเข้าถึงเว็บไซต์นั้น มี process ที่ค้างอยู่เป็นจำนวนไม่มาก เพราะเนื่องจากอัตราการส่งและ ขนาดของข้อมูลของการทดลองที่ใช้ปริมาณน้อย ส่วนค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 0.2

โดยจากการทดลองนี้ผลที่ได้ออกมาจากการที่ทำการปิดกั้นจำนวน 5 website ทำการปิดกั้นสำเร็จเพียงแค่ 3 website เท่านั้น เพราะว่า OpenWRT มีการปิดกั้นในรูปแบบ IP Address จึงทำให้ในบาง website ที่มีหลาย IP ไม่สามารถปิดกั้นได้ และจากการที่กราฟ Interface Bandwidth ในรูปที่ 4.22 นั้นมีค่าไม่คงที่ ขึ้นอยู่กับปริมาณ content ของในแต่ละเว็บไซต์ รวมถึงในช่วงจังหวะที่ทำการ refresh หน้าเว็บไซต์ ซึ่งในบางครั้งอาจจะดึงข้อมูลที่อยู่ในเว็บไซต์ไม่เหมือนกัน ยกตัวอย่างเช่น หน้าโฆษณา เป็นต้น

สรุปแล้วจากการทดสอบของการเข้าถึงเว็บไซต์ทั้ง 2 การทดสอบนั้น การทดสอบการเข้าถึงเว็บไซต์แบบไม่มีการปิดกั้นใดๆ เป็นการทดสอบที่ใช้ CPU Usage, Interface Bandwidth และค่าเฉลี่ยภาระงานที่มากกว่าการทดสอบการเข้าถึงเว็บไซต์แบบทำการปิดกั้น เพราะว่า content ที่อนุญาตผ่านไปยัง Raspberry Pi Firewall มีมากกว่า ซึ่งในการทดสอบนี้สรุปได้ว่า ปริมาณของ rule ที่ทำการปิดกั้นนั้น มีจำนวนไม่มาก แต่ สาเหตุหลักที่การใช้งาน CPU มีค่าที่สูงเพราะมาจากปริมาณ content ที่อนุญาตผ่านจาก Raspberry Pi firewall มีผลมากกว่าจำนวน rule ของ firewall ที่ทำการปิดกั้น

3. การทดสอบประสิทธิภาพในการใช้งานรับและส่ง email ด้วย โพรโตคอล SMTP



รูปที่ 4.23 กราฟแสดงผลการทดสอบประสิทธิภาพในการใช้งานรับและส่งอีเมล์ ของ Openwrt

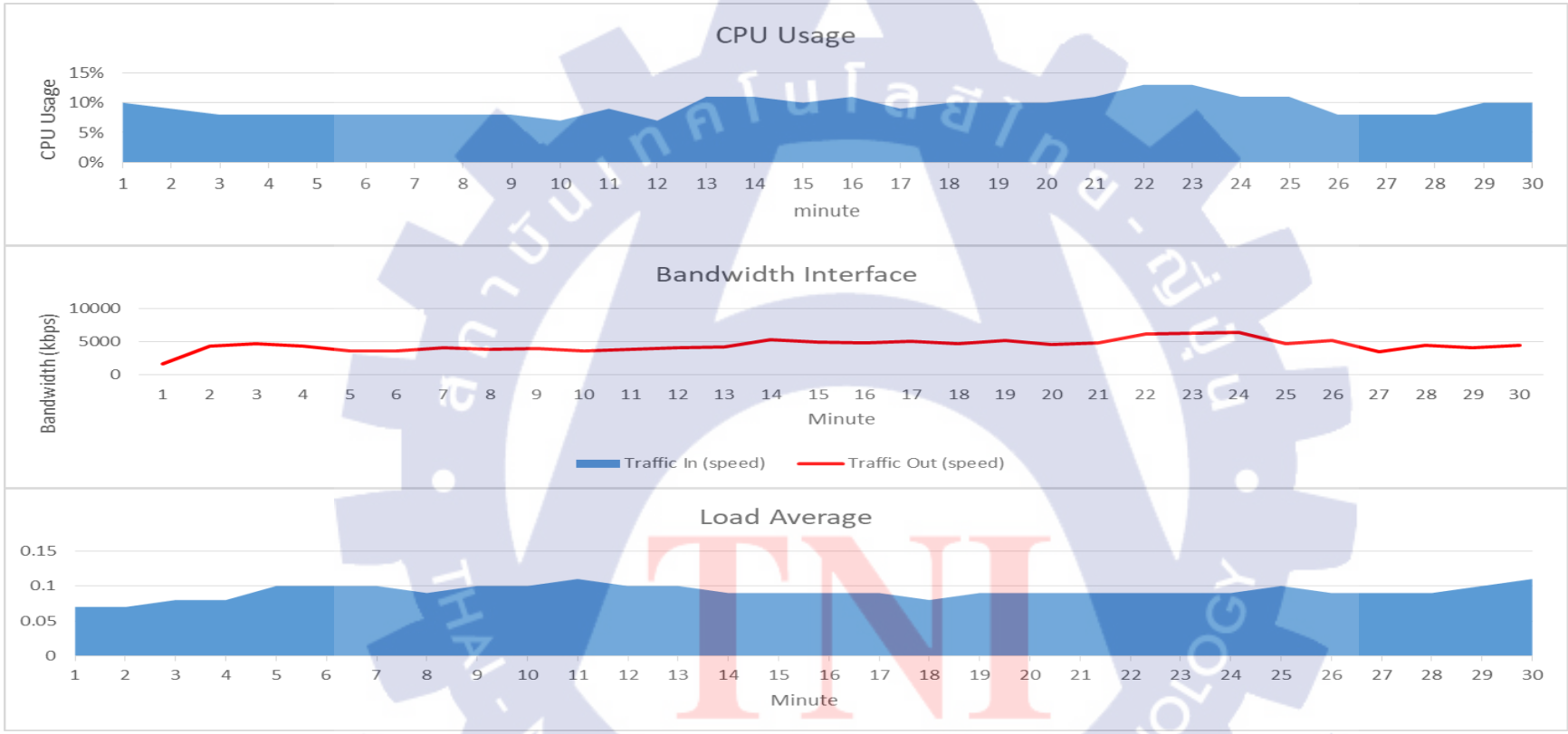
จากกราฟรูปที่ 4.23 เนื่องด้วยการทำงานของ email ตามที่กล่าวไว้แล้วในบทที่ 3 จะเป็นการส่งข้อมูลเพียง 10MB ต่อ 1 วินาที ไม่ได้สูงมากเท่าของการทดลองของ FTP โดยที่การใช้งาน CPU สูงที่สุดในการทดลองนี้อยู่ที่ 37% และ ค่าเฉลี่ยอยู่ที่ประมาณ 22% ซึ่งในบางครั้งจะมีบางช่วงที่ไม่สามารถส่งได้ตามที่ตั้งค่าเอาไว้คือ 10MB เนื่องจากความเสถียรของ Internet ในช่วงเวลานั้น และจะมีการทบการส่งไปยังช่วงวินาทีต่อไป ทำให้กราฟที่ได้นั้น ไม่ได้คงที่เสมอไป

สำหรับแบนด์วิธที่ใช้งานนั้นจะสอดคล้องกับคำอธิบายของการใช้งาน CPU ก่อนหน้านี้ โดยในการทดลองจะเป็นการส่งข้อมูลจำนวน 10MB ทุกๆ 1 วินาที ซึ่งจากกราฟ OpenWRT สามารถทำงานได้ และ Raspberry Pi สามารถส่ง email ได้ทุกช่วงเวลา โดยที่การใช้งานแบนด์วิธสูงสุดอยู่ที่ 20,880 kbps (20.88 Mbps) และจากค่าเฉลี่ยภาระงานในการทำงานของ CPU สูงสุดอยู่ที่ 0.41 ซึ่งสอดคล้องกับกราฟการใช้งานของ CPU Usage ในรูปที่ 4.23

จากผลการทดลองดังกล่าวสามารถสรุปได้ว่า OpenWRT สามารถใช้งานในด้านการรับและส่ง email ได้เพราะว่าแบนด์วิธการใช้งาน email นั้นไม่ได้สูงมาก ทำให้ Raspberry pi firewall ยังสามารถทำงานได้อย่างเต็มประสิทธิภาพ และยังสามารถรองรับการใช้งานอื่นๆ ได้อีก โดยดูจากประสิทธิภาพการใช้งาน CPU เป็นหลัก

4. <u>การทดสอบประสิทธิภาพการใช้งาน Live Streaming</u>

ก. การทดสอบประสิทธิภาพการใช้งาน Live Streaming โดยไม่ได้มีการปิดกั้นใดๆ



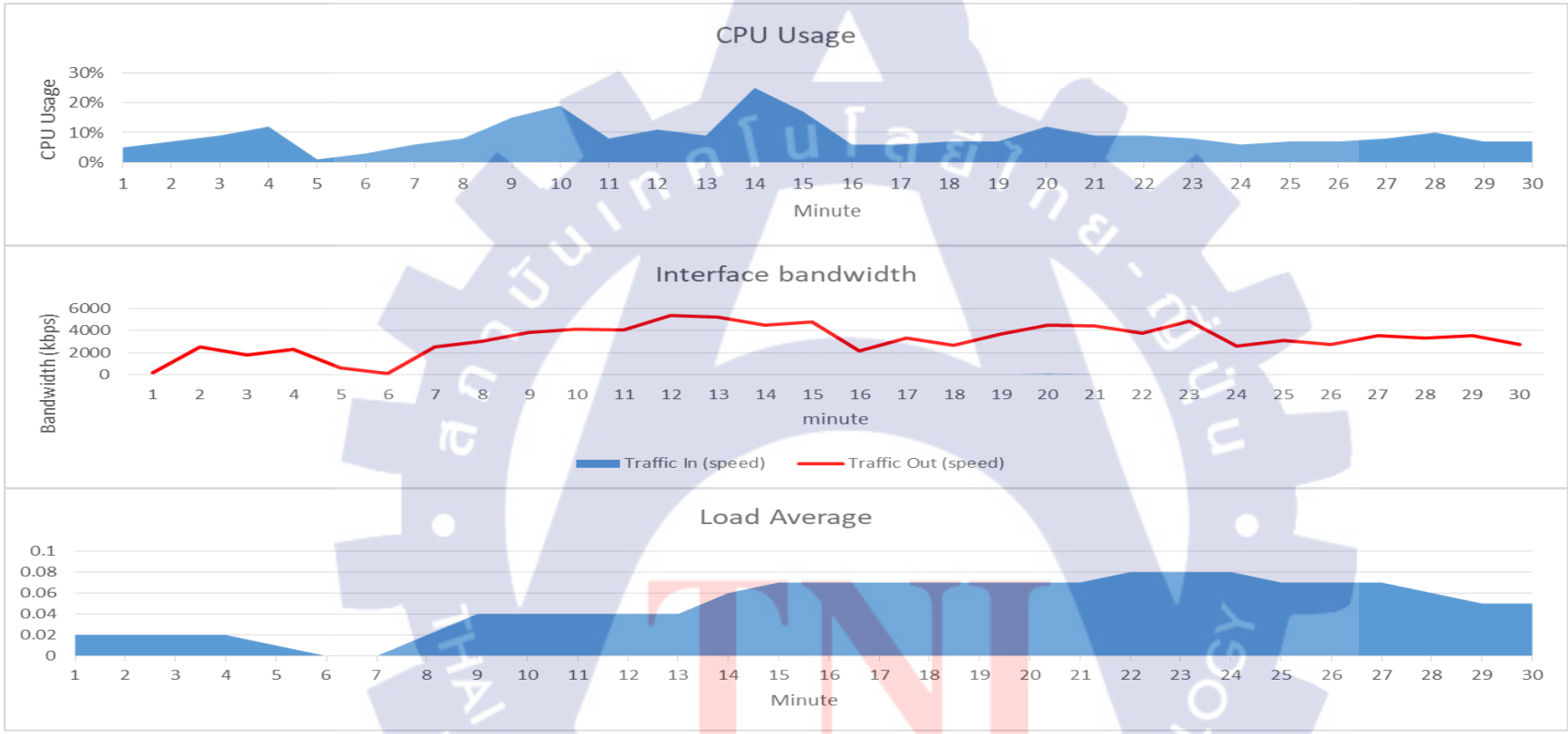
รูปที่ 4.24 กราฟแสดงผลการทดสอบประสิทธิภาพการใช้งาน Live Streaming โดยไม่ได้มีการปิดกั้นใดๆ ของ Openwrt

ผลการทดสอบแบบ Live Streaming นี้เป็นผลการทดสอบ Live Streaming โดยทำการเปิดช่องทีวี 3 ช่องพร้อมกันด้วยความละเอียด 720p ซึ่งค่าเฉลี่ยของการใช้งาน CPU ของการทดสอบนี้อยู่ที่ 10% และค่าสูงสุดของการใช้งาน CPU อยู่ที่ 23% ซึ่งสอดคล้องกับ Bandwidth ที่ใช้งานโดยเฉลี่ย 1 ช่องจะใช้ CPU โดยประมาณ 3.3%

ส่วนค่าสูงสุดของการใช้งานแบนด์วิธ อยู่ที่ 6,400 kbps (6.4 Mbps) ซึ่งหมายความว่าโดยเฉลี่ย 1 ช่องจะใช้แบนด์วิธ โดยประมาณ 2,130 kbps (2.13 Mbps) ซึ่งตรงตามหลักการของการ streaming ในช่วงความละเอียด 720p จะต้องใช้แบนด์วิธอย่างน้อยต้องอยู่ในช่วง 2-4 Mbps ทั้งนี้ขึ้นอยู่กับ content และภาพที่ส่งมาในช่วงเวลานั้นๆ ว่า ในรายการนั้นๆ ส่งภาพมาในรูปแบบใด โดยในแต่ละช่วงเวลาจะมีปริมาณของ content ไม่เท่ากัน

ส่วนค่าสูงสุดของค่าเฉลี่ยภาระงาน อยู่ที่ 0.12 ซึ่งสอดคล้องกับการใช้งาน CPU นั่นหมายความว่าการทำงานของ Live Streaming OpenWRT ยังสามารถทำงานได้อย่างเต็มประสิทธิภาพ และยังสามารถรองรับการใช้งานอื่นๆ ได้อีก

ข. การทดสอบประสิทธิภาพการใช้งาน Live Streaming โดยมีการปิดกั้นบางช่อง



รูปที่ 4.25 กราฟแสดงผลการทดสอบประสิทธิภาพการใช้งาน Live Streaming โดยมีการปิดกั้นบางช่อง ของ Openwrt

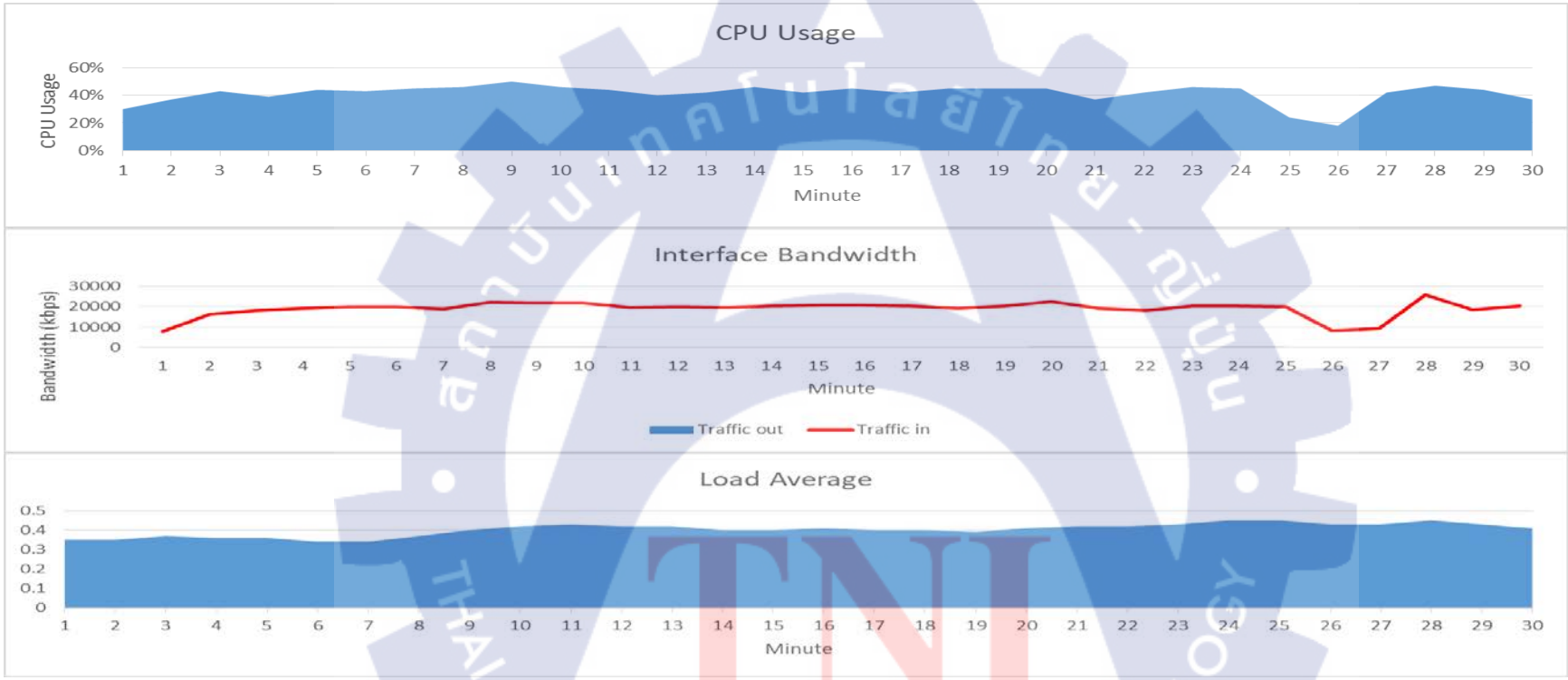
ผลการทดสอบแบบ Live Streaming ในรูปที่ 4.25 นี้เป็นผลการทดสอบ Live Streaming โดยทำการเปิดช่องทีวี 3 และทำการปิดกั้นด้วย firewall จำนวน 1 ช่องด้วยความละเอียด 720p ซึ่งค่าเฉลี่ยของการใช้งานหน่วยประมวลผลอยู่ที่ 8% และค่าสูงสุดของการใช้งาน CPU อยู่ที่ 26% ซึ่งสอดคล้องกับ Bandwidth ที่ใช้งานโดยเฉลี่ย 1 ช่องเหมือนที่กล่าวไปในกรณีเปิดพร้อมกัน 3 ช่อง แต่ว่าในการทดลองนี้จะเหลือ traffic ที่ผ่านได้จำนวน 2 ช่องเท่านั้น ค่าการใช้งาน CPU จึงลดลง โดยทั้งนี้ขึ้นอยู่กับ content และภาพที่ส่งมาในช่วงเวลานั้นๆ ว่า ในรายการนั้นๆส่งภาพมาในรูปแบบใด โดยในแต่ละช่วงเวลาจะมีปริมาณของ content ไม่เท่ากัน

ค่าสูงสุดของการใช้งานแบนด์วิธ อยู่ที่ 5,340 kbps (5.34 Mbps) ที่ใช้งานโดยเฉลี่ย 1 ช่องเหมือนที่กล่าวไปในกรณีเปิดพร้อมกัน 3 ช่อง ในรูปที่ 4.24 แต่ว่าในการทดลองนี้จะเหลือ traffic ที่ผ่านได้จำนวน 2 ช่องเท่านั้น ซึ่งผลที่ได้นั้นก็สอดคล้องกับการใช้งาน CPU ที่ลดลงไปเพราะมีการถูกปิดกั้น traffic ด้วย

ค่าสูงสุดของค่าเฉลี่ยภาระงาน อยู่ที่ 0.11 ซึ่งสอดคล้องกับการใช้งาน CPU และแบนด์วิธที่ลดลง นั่นหมายความว่าการทำงานของ Live Streaming ของ OpenWRT ยังสามารถทำงานได้อย่างเต็มประสิทธิภาพ และยังสามารถรองรับการใช้งานอื่นๆ ได้อีก

จากการทดสอบการใช้งาน Live Streaming ของทั้ง 2 กรณี ในรูปที่ 4.24 และรูปที่ 4.25 กรณีที่ไม่ได้ทำการปิดกั้นช่องจะใช้ประสิทธิภาพการใช้งาน CPU แบนด์วิธ และค่าเฉลี่ยภาระงานที่มากกว่า เพราะว่าปริมาณของ content ทั้งหมดของกรณี streaming พร้อมกัน 3 ช่องมีจำนวนมากกว่า และปริมาณของ rule ที่ทำการปิดกั้นนั้น มีจำนวนไม่มาก แต่สาเหตุหลักที่การใช้งาน CPU มีค่าที่สูงมาจากปริมาณ content ที่อนุญาตผ่านจาก Raspberry Pi firewall มีผลมากกว่าจำนวน rule ของ firewall ที่ทำการปิดกั้น

5. การทดสอบประสิทธิภาพการทำงานพร้อมกันของแอพพลิเคชั่นมากกว่า 1 ประเภท ได้แก่ การเข้าถึง website, การส่ง email และlive streaming



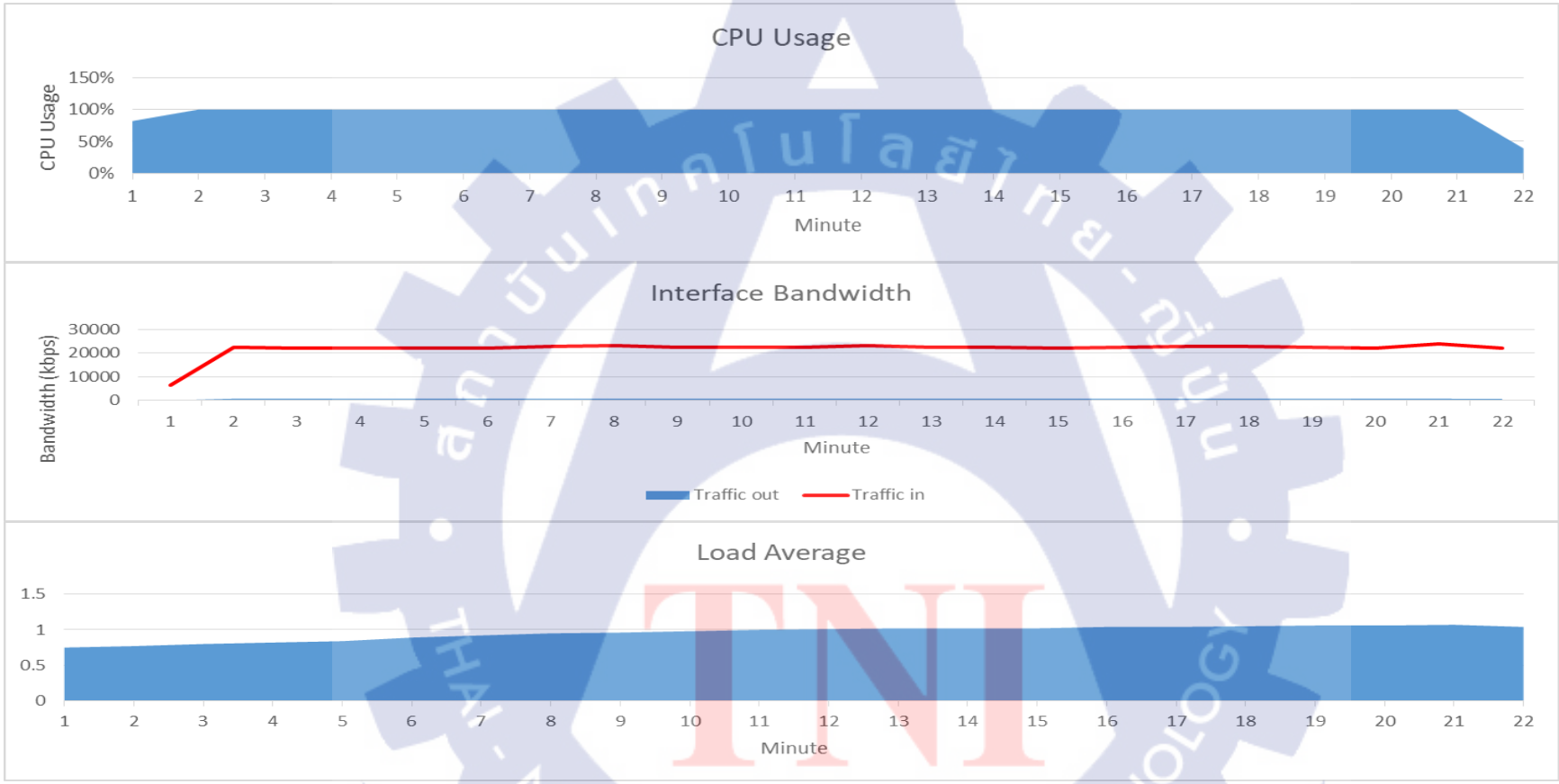
รูปที่ 4.26 กราฟแสดงผลประสิทธิภาพ การทำงานแบบแอพลิเคชั่นมากกว่า 1 อย่างของ OpenWRT

จากรูปที่ 4.26 เป็นกราฟการทดสอบประสิทธิภาพการทำงานพร้อมกันของแอพพลิเคชั่นมากกว่า 1 ประเภท ได้แก่ การเข้าถึง website, การส่ง email และlive streaming โดยการประมวลผล CPU เมื่อทำงานแบบหลายแอพพลิเคชั่นพร้อมๆกัน ซึ่งค่าเฉลี่ยของการทดลองนี้อยู่ที่ 41% โดยค่าสูงสุดของการทดลองนี้อยู่ที่ 50% แบนด์วิธสูงสุดในการใช้งานอยู่ที่ 25,881 kbps (25.88 Mbps)

ส่วน ค่าเฉลี่ยภาระงาน ซึ่งจะสังเกตเห็นว่าค่าสูงสุดของค่าเฉลี่ยภาระงานอยู่ที่ 0.45 เพราะฉะนั้นประสิทธิภาพการทำงาน ยังสามารถทำงานได้ดี เพราะว่า CPU ที่ใช้งานยังไม่ถึง 100% และไม่มีการสะดุดหรือเกิดการขัดข้อง หรือเรียกว่า buffering ในระหว่างการใช้งาน

การทดลองนี้สามารถใช้งานได้เต็มประสิทธิภาพ สามารถส่ง email ได้ตลอดช่วงเวลา เข้าถึงเว็บไซต์ที่ทำการเปิดให้ใช้งานได้ รวมถึงสามารถใช้งาน Live Streaming พร้อมกันได้โดยไม่มีการติดขัดระหว่างรับชมรายการ

6. การทดสอบประสิทธิภาพการทำงานแบบแอพลิเคชั่นทุกอย่างเข้าด้วยกันทั้ง FTP การเข้าถึงเว็บไซต์ Email และ Livestreaming



รูปที่ 4.27 กราฟแสดงผลประสิทธิภาพ การทำงานแบบแอพลิเคชั่นทุกอย่างเข้าด้วยกัน ของ OpenWRT

จากการทดสอบแบบการนำทุกแอพพลิเคชั่นทำงานพร้อมกัน ทั้ง FTP การเข้าถึงเว็บไซต์ Email และ Livestreaming จะสังเกตว่าในระหว่างการทดลอง ประสิทธิภาพการใช้งาน CPU ขึ้นไปถึง 100% ซึ่งการใช้งาน FTP ทำการ upload ไฟล์เสร็จสิ้นเป็นเวลา 22 นาที ซึ่งแตกต่างจากการทดสอบการใช้งาน FTP เพียงอย่างเดียว ดังรูปที่ 4.20 ซึ่งส่งไฟล์สมบูรณ์เพียง 18 นาทีเท่านั้น ซึ่งเร็วกว่าการทดสอบนี้

ในด้าน bandwidth ที่ใช้งาน แบนด์วิธการใช้งานสูงสุดอยู่ที่ 23,920 kbps (23.92 Mbps) ซึ่งเป็นค่าสูงสุดของความสามารถของ OpenWRT ที่สามารถดาวน์โหลด content ได้และค่าเฉลี่ยภาระงานสูงสุดของการทดลองนี้อยู่ที่ 1.07 นั่นแสดงว่าในบางช่วงเวลามีภาระงานค้างอยู่ทำให้การใช้งาน เว็บไซต์ อีเมล์ และ live streaming ทำงานได้อย่างไม่เต็มประสิทธิภาพ เพราะมี traffic จาก FTP ดึงไปเป็นจำนวนมาก

เมื่อเทียบกับการใช้งานแบบการนำแอพพลิเคชั่นมากกว่า 1 อย่างในรูปที่ 4.26 นั้น ประสิทธิภาพของการทำงานของการทดลองนี้ แย่กว่าการทดลองในรูปที่ 4.26 เพราะว่าเป็นการใช้ CPU ที่สูง และค่าเฉลี่ยภาระงานที่เยอะกว่า เมื่อเทียบกับการทดลองแบบไม่มี FTP เพราะว่า traffic ของ การทดลองแบบการส่ง FTP แบบส่ง 3 ครั้งพร้อมกัน มีปริมาณมาก ทำให้การใช้งาน CPU มีค่อนข้างสูง และไม่เพียงพอต่อการใช้งานอื่นๆ ตามอัตราส่ง traffic ของแอพพลิเคชั่น ส่งผลให้การใช้งานบางอย่าง เช่น การใช้งานเว็บไซต์ที่สามารถเข้าได้ช้าบ้างในบางช่วงเวลา การใช้งาน email ที่ไม่สามารถส่งได้เต็ม 10 Mbps ในบางครั้ง และทำให้การ streaming ภาพของบางช่องมีการติดขัดระหว่างรับชมช่องรายการ

บทที่ 5
บทสรุปผลการทดลอง และข้อเสนอแนะ

บทสรุปผลการทดลอง

จากผลการทดลองในบทที่ 4 จะได้ข้อสรุป ออกมาเป็นตัวเลขทั้งค่าเฉลี่ยการใช้งาน CPU การใช้งานแบนด์วิธสูงสุด และค่าสูงสุดของค่าเฉลี่ยภาระงานออกมาเป็นตาราง ในแต่ละการทดลองต่างๆ และ ตาม Open Source Firewall ดังตารางต่อไปนี้

ตารางที่ 5.1 ตารางค่าสรุปผล ค่าเฉลี่ยการใช้งาน CPU การใช้งานแบนด์วิธสูงสุด และค่าสูงสุดของค่าเฉลี่ยภาระงาน

	Iptable			IP Fire			Openwrt		
	CPU	Bw.	Load avg.	CPU	Bw.	Load avg.	CPU	Bw.	Load avg.
FTP Protocol									
1. Single Thread	45	20	0.85	100	18	3.5	88	20	0.83
2. Multi Thread	73	31	1.08	100	18	3.8	100	21.09	1.04
HTTP protocol									
1. unblock website (Allow All)	34	13.44	0.38	33	10.55	0.67	27	12.85	0.25
2. block 5 website	23	9.1	0.27	27	8.81	0.55	22	11.38	0.2
SMTP Protocol									
1. sent email	27	9.24	0.32	39	16.44	1.96	22	20.88	0.41
Live Streaming									
1. stream 3 channel	11	4.95	0.19	23	5.4	0.53	10	6.4	0.12
2. Stream 2 channel	8	3.82	0.25	14	4.73	0.46	8	5.34	0.11
MixUse									
HTTP+SMTP+Live Streaming	57	21.22	0.66	66	22.07	1.41	41	25.88	0.45
HTTP+SMTP+Live Streaming + FTP	98	29.9	1.29	100	n/a	12.14	96	23.92	1.07

ซึ่งในงานวิจัยนี้จะยกประเด็นที่สำคัญๆ ในการสรุปผลเพื่อหา open source firewall ที่ดีที่สุดและข้อดี ข้อเสียของการใช้งานในแต่ละ open source firewall

ประเด็นที่ 1 ประสิทธิภาพการใช้งานจากสถิติ

จากตารางที่ 5.1 แสดงการสรุปผล ค่าเฉลี่ยการใช้งาน CPU การใช้งานแบนด์วิธสูงสุดและค่าสูงสุดของค่าเฉลี่ยภาระงาน ของการทดลองต่างๆ โดยที่ในงานวิจัยนี้จะให้ความสำคัญกับการใช้งาน CPU ที่น้อยกว่า เป็นอันดับแรก ส่วน ค่าเฉลี่ยภาระงานที่น้อยกว่า และแบนด์วิธการใช้งานที่สูงกว่าจะทำการพิจารณา เป็นลำดับถัดไป

การทดสอบการใช้งานส่งไฟล์ด้วย FTP การส่งไฟล์พร้อมกัน 3 ไฟล์ (Mult ithread) จะใช้ CPU, แบนด์วิธ และ ค่าเฉลี่ยภาระงานสูงกว่าการส่งไฟล์พร้อมกันไฟล์เดียว (Single thread) เมื่อสังเกตจากการใช้งาน CPU ผลที่ได้คือ IPTable มีประสิทธิภาพการใช้งานที่ดีที่สุดเพราะค่าการใช้งาน CPU 73% ซึ่งเป็นค่าน้อยที่สุดใน Open Source Firewall ทั้ง 3 ตัว ส่วน OpenWRT และ IPFire มีค่าการใช้งาน CPU 100% ทั้ง 2 ตัว ซึ่งมากกว่า IPTable

การทดสอบการใช้งานเว็บไซต์ เมื่อไม่มีการปิดกั้น จะใช้ CPU, แบนด์วิธ และค่าเฉลี่ยภาระงานสูงกว่า เมื่อทำการปิดกั้นเว็บไซต์ เมื่อสังเกตจากการใช้งาน CPU ของ OpenWRT อยู่ที่ 22% หมายความว่าประสิทธิภาพการใช้งานของ OpenWRT ดีที่สุดเพราะค่าการใช้งาน CPU น้อยที่สุด รองลงมาเป็น IPTable ที่มีการใช้งาน CPU อยู่ที่ 23% และ IPFire การใช้งาน CPU อยู่ที่ 27% ตามลำดับ

การทดสอบการใช้งานอีเมล์เมื่อสังเกตุจากการใช้งาน CPU OpenWRT มีประสิทธิภาพการใช้งานที่ดีที่สุดเพราะค่าการใช้งาน CPU น้อยที่สุด อยู่ที่ 22% รองลงมาเป็น IPTable ค่าการใช้งาน CPU อยู่ที่ 27% และ IPFire ค่าการใช้งาน CPU อยู่ที่ 39% ตามลำดับ

การทดสอบการใช้งาน Live Streaming เมื่อไม่มีการปิดกั้นช่องรายการ จะใช้ CPU, แบนด์วิธ และค่าเฉลี่ยภาระงานสูงกว่าเมื่อทำการปิดกั้นเว็บไซต์ เมื่อสังเกตจากการใช้งาน CPU IPTable กับ OpenWRT มีประสิทธิภาพการใช้งาน CPU ที่เท่ากันที่ 8% แต่ OpenWRT จะมีค่าเฉลี่ยภาระงานที่ 0.11 ซึ่งน้อยกว่า IPTable ที่มีค่าเฉลี่ยภาระงานที่ 0.25 ส่วน IPFire จะมีค่าเฉลี่ยภาระงานที่มากที่สุดที่ 0.46 เพราะฉะนั้นในการใช้งาน Live Streamimg OpenWRT มีประสิทธิภาพการใช้งานได้ดีที่สุด

ส่วนการทดสอบประสิทธิภาพการทำงานพร้อมกันของแอพพลิเคชั่นมากกว่า 1 ประเภท ที่รวมการใช้งานทั้ง FTP, การใช้งานเว็บไซต์ อีเมล์ และ Live Streaming จะใช้ CPU, แบนด์วิธ และค่าเฉลี่ยภาระงานสูงกว่า การทดสอบประสิทธิภาพการทำงานพร้อมกันของแอพพลิเคชั่นมากกว่า 1 ประเภท แบบไม่มี FTP เมื่อสังเกตจากการใช้งาน CPU ของ OpenWRT อยู่ที่ 96% หมายความว่าประสิทธิภาพการใช้งานของ OpenWRT ดีที่สุดเพราะค่าการใช้งาน CPU น้อยที่สุด รองลงมาเป็น IPTable การใช้งาน CPU อยู่ที่ 98% และ IPFire การใช้งาน CPU อยู่ที่ 100% ตามลำดับ

จากตารางที่ 5.1 สรุปได้ว่าประสิทธิภาพการทำงานรวมของ OpenWRT ทำงานได้ดีที่สุด ยกเว้น การส่งข้อมูล FTP ประสิทธิภาพการใช้งานของ IPTable ทำงานได้ดีที่สุด ซึ่งจากตารางที่ 5.1 การใช้งาน CPU ในการทดสอบการส่งไฟล์ด้วย FTP พร้อมกัน 3 ไฟล์ ของ IPTable อยู่ที่ 73% ส่วน OpenWRT อยู่ที่ 100% ส่วน IPFire นั้นมีประสิทธิภาพในภาพรวมเป็นอันดับที่ 3 ซึ่งก็กล่าวได้ว่า OpenWRT มีประสิทธิภาพการทำงานได้ดีเมื่อจำนวน traffic มีปริมาณที่น้อย แต่ในกรณีของ Traffic ที่มีจำนวนมาก IPTable ทำงานได้ดีกว่า

ประเด็นที่ 2 ความง่ายในการติดตั้งและการใช้งาน

สำหรับ IPTable จากที่กล่าวไปในบทที่ 1 เนื่องด้วยการใช้งานทั้งหมดอยู่ภายใต้การทำงานแบบ command line จึงทำให้การใช้งานนั้นเป็นไปค่อนข้างยาก ถ้าผู้ที่ใช้งานนั้นไม่มีพื้นฐานการใช้งาน Linux ที่เพียงพอรวมถึงกับการสร้าง rule ที่ต้องอาศัยความเข้าใจเป็นอย่างมาก

สำหรับ IPFire เป็นการใช้งานแบบ Web user interface ทั้งในการติดตั้ง และการใช้งาน จึงมีการใช้งานที่ง่าย และสะดวกสำหรับผู้ใช้งานใหม่ แต่การติดตั้งการใช้งานของ IPFire ไม่สามารถปรับแต่งการใช้งานในระดับ kernel ได้ และผู้พัฒนาไม่ได้มีการใช้ทรัพยากรในส่วนของ port ethernet ของ Raspberry Pi ได้ จึงจำเป็นต้องใช้ USB-LAN ในการใช้งานทั้ง 2 interface ซึ่งการใช้งาน USB นั้นก็ไม่สามารถทำงานได้เต็มประสิทธิภาพเทียบเท่ากับ port ethernet ที่มาพร้อมกับ raspberry Pi ได้

สำหรับ OpenWRT เป็นการใช้งานแบบ Web user interface ทั้งในการติดตั้งและการใช้งานจึงมีการใช้งานที่ง่าย และสะดวกสำหรับผู้ใช้งานใหม่ แต่ในเรื่องของการสร้าง port ethernet เพื่อทำเป็นพอร์ตที่ 2 ในการใช้งานนั้น มีข้อจำกัดการใช้งานได้แค่บางรุ่นเท่านั้น ซึ่งจะแตกต่างกับ IPTable และ IPFire ที่สามารถใช้งานได้ทุกอุปกรณ์

จากที่กล่าวมานั้นสรุปในด้านความง่ายในการติดตั้งและการใช้งานสรุปได้ว่า IPFire และ OpenWRT มีความง่ายในการติดตั้งและการใช้งานเท่ากัน เพราะว่าเป็นการใช้งานแบบ Web user interface ซึ่งแตกต่างจาก IPTable ซึ่งทำงานในรูปแบบ command line.

ประเด็นที่ 3 การจัดสรรทรัพยากรและ process ระหว่างการทำงาน

สำหรับ IPTables เป็น open source firewall ซึ่งมีขนาดการใช้งานและการใช้ทรัพยากรในการใช้งาน (Environment) น้อย จึงทำให้มีข้อได้เปรียบในด้านประสิทธิภาพการใช้งานทั้งการส่งไฟล์แบบ FTP การเข้าถึงเว็บไซต์ การส่ง email และการใช้งาน Live Streaming โดยสังเกตจากค่าเฉลี่ยภาระงานในตารางรูปที่ 5.1

ส่วนทรัพยากรในการใช้งานของ IPFire มีค่อนข้างมากและมี feature การใช้งานที่หลากหลาย ซึ่งก็ต้องแลกกันการกินทรัพยากรบางส่วนให้กับ feature ที่ใช้งาน และ process การทำงานที่เป็น interval มีการใช้ประสิทธิภาพการทำงานของ CPU ที่สูง ในทุกๆ 5 นาทีจะเห็น process การทำงานของ collector ในทุกๆ 5 นาที ซึ่งจะสังเกตได้จากกราฟของ IPFire เกือบทุกกราฟจะมียอดเขาจำนวนถี่ๆ เท่าๆ กับทุก 5 นาที และส่งผลทำให้การทำงานของ FTP ในบางครั้งมีการใช้งาน CPU ที่สูงเกินขีดจำกัด

และ OpenWRT นั้นมีการใช้ทรัพยากรที่น้อยและ Process การทำงานที่น้อยเช่นเดียวกับ IPTable รวมถึงการจัดสรรทรัพยากรอย่างเป็นระบบทำให้ประสิทธิภาพการทำงานที่ตามมานั้นมีประสิทธิภาพในการทำงานในขั้นดีกว่า IPTables และ IPFire โดยสังเกตจากค่าเฉลี่ยภาระงานในตารางที่ 5.1 ค่าเฉลี่ยภาระงานของทุกการทดลองจะมีค่าน้อยที่สุดเสมอยกเว้นการทดลองส่งไฟล์ด้วย FTP

เพราะฉะนั้น ในด้านการจัดสรรทรัพยากร และprocess ในการทำงาน สรุปได้ว่า OpenWRT สามารถจัดสรรทรัพยากร และใช้ process การทำงานน้อยที่สุด ส่วน IPTable สามารถจัดสรรทรัพยากรและใช้ process การทำงานน้อยเป็นอันดับที่ 2 ส่วน IPFire จัดสรรทรัพยากรและใช้ process การทำงาน ได้แย่ที่สุด

<u>สรุปจากทั้ง 3 ประเด็นเพื่อหา Open Source Firewall ที่ดีที่สุด</u>

จาก 3 ประเด็นที่ได้กล่าวมา ทั้งในเรื่องประสิทธิภาพ ความง่ายในการติดตั้ง และการใช้งานและการจัดการทรัพยากร ได้ข้อสรุปว่า OpenWRT เป็น Open Source Firewall ที่ดีที่สุดใน 3 Open Source Firewall ทั้งในเรื่องประสิทธิภาพการทำงานที่ดีที่สุดง่ายต่อการใช้งาน และการติดตั้งและ การจัดสรรทรัพยากรการใช้งานที่ดีที่สุด

สำหรับ IPTables เป็น open source firewall ซึ่งมีขนาดการใช้งานและการใช้ทรัพยากรณ์ในการใช้งาน (Environment) น้อย จึงทำให้มีข้อได้เปรียบในด้านประสิทธิภาพการใช้งานทั้งการส่งไฟล์แบบ FTP การเข้าถึงเว็บไซต์ การส่ง email และการใช้งาน Live Streaming

สำหรับ IPFire ถึงแม้จะเป็น open source firewall ที่มี Web user interface การใช้งาน และการจัดการเรื่องการ setting firewall ในด้านต่างๆได้ดีที่สุด แต่ในด้านประสิทธิภาพการใช้งานทั้งการส่งไฟล์แบบ FTP การเข้าถึงเว็บไซต์ การส่ง email และการใช้งาน Live Streaming ทำได้แย่ที่สุดใน 3 open source firewall จะสังเกตได้จากตัวเลขผลสรุปในด้าน CPU และค่าเฉลี่ยภาระงานที่มีตัวเลขมากที่สุด ทั้งนี้มีปัจจัยสำคัญที่ทำให้ประสิทธิภาพในการใช้งานลดลง อันเนื่องมาจากทรัพยากรในการใช้งานของ IPFire มีค่อนข้างมาก และมี feature การใช้งานที่หลากหลาย ซึ่งก็ต้องแลกกันการกินทรัพยากรบางส่วนให้กับ feature ที่ใช้งาน

ข้อเสนอแนะและแนวทางงานวิจัยในอนาคต

เนื่องจากปัจจุบันนี้ Raspberry Pi ได้ทำการจำหน่าย Raspberry Pi 3 ซึ่งทำการอัพเกรดการทำงานไปจาก Raspberry Pi 1 เพื่อให้ผลทดสอบเป็นไปอย่างปัจจุบัน และสามารถสร้างทางเลือกในการเลือกซื้อ Firewall ในราคาที่เหมาะสมกับหน่วยงานธุรกิจขนาดกลาง และขนาดเล็กได้ ทางผู้จัดทำแนะนำให้ทำการทดลองด้วย Raspberry Pi 3 เพื่อที่ให้ทางหน่วยงานที่ตัดสินใจในการเลือกซื้อได้ทำการใช้ข้อมูลดังกล่าวในการเลือกซื้ออุปกรณ์รักษาความปลอดภัยที่เป็นปัจจุบัน และยังสามารถต่อยอดได้ในอนาคต

ส่วนในเรื่องขององค์ประกอบในการทดสอบประสิทธิภาพ ให้ทำการศึกษาการทดสอบประสิทธิภาพจาก NSS Labs [16] ซึ่งเป็นหน่วยงานวิจัยด้านความปลอดภัย ซึ่งรายละเอียดจะเป็นการทดสอบประสิทธิภาพการทำงานในหลากหลายสถานการณ์ และหลากหลายรูปแบบซึ่งการทดสอบประสิทธิภาพในงานวิจัยนี้เป็นแค่การทดสอบเบื้องต้นเท่านั้น

บรรณานุกรม

บรรณานุกรม

[1] The OWASP Foundation,*"The OWASP Top 10 - 2017,"* [Online]. Avaliable : https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. [Accessed: February 12, 2018].

[2] Symantec Corporation, "*Internet Security Threat Report 2018,"* California: Symantec Corporation, 2018.

[3] M. Joshi and C. Gohel, "Agent base network traffic monitoring," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 2, no. 5, pp. 1799–1803, May 2013.

[4] O. Evroro, "Internet Security: the role of firewall system," M.S. Thesis (Computer Science), Delta State Polytechnic, Otefe-Oghara, Nigeria, 2011.

[5] J. Corbet et al., *Linux Kernel Development*, San Francisco: A Linux Foundation Publication, 2015.

[6] Debian Company, "*About Debian,"* [Online]. Avaliable : https://www.debian.org /intro/about. [Accessed: September 8, 2017].

[7] ศุภพร รัฐอาจ และคณะ,*"การปรับแต่งไฟร์วอลล์โอเพ่นซอร์สให้มีประสิทธิภาพสูงสุด,"* วิทยานิพนธ์ วท.ม. (วิทยาการคอมพิวเตอร์), มหาวิทยาลัยมหาสารคาม, มหาสารคาม, 2553.

[8] Ubuntu, "*Iptables How To*," [Online]. Avaliable: https://help.Ubuntu.com/com munity/IptablesHowTo. [Accessed: September 10, 2017].

[9] IPFire, *"About IPFire,"* [Online]. Avaliable: http://www.ipfire.org/features. [Accessed: September 10, 2017].

[10] OpenWRT, *"About OpenWRT,"* [Online]. Avaliable: https://openwrt.org/. [Accessed: 10 September 10, 2017].

[11] Raspberry Pi Foundation, *"Raspberry PiHardware,"* [Online]. Avaliable: https://www.raspberrypi.org/documentation/hardware/raspberrypi/. [Accessed: September 12, 2017].

[12] C. Sheth and R. Thakker, "Performance Evaluation and Comparative Analysis of Network Firewalls," *Devices and Communications, ICDeCom*, Mesra, Ranchi, India, February 24-25, 2011, pp.1-5.

[13] J. Kadlecsik and G. Pásztor, *"Netfilter Performance Testing,"* [Online]. Avaliable: http://people.netfilter.org/kadlec/nftest.pdf. [Accessed: September 14, 2017]

[14] Z. Nabi, *"A $35 Firewall for the Developing World,"* Dublin: IBM Research, 2014.

[15] F. P. Tso et al., "*The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures,"* M.S. Thesis (Computing Science), University of Glasgow, Glasgow, UK, 2013.

[16] NSS Labs, Inc.,*"Test Methodology – Next Generation Firewall v8.0,"* Austin: NSS Labs Inc., 2017.