ประสิทธิภาพการทำงานของ Multipath TCP ในระบบโครงข่ายส่วนบุคคล

สรสิช ศรีเอี่ยมสะอาด

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมมาคม วิทยาลัยนวัตกรรม
ด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์
พ.ศ. 2562

Performance of Multipath TCP in Personal Network

Sorasich Sriaimsa-ard

A Thematic Paper Submitted in Partial Fulfillment of the Requirements

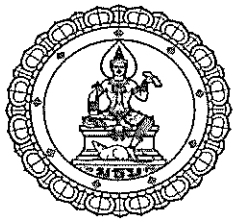
For the Degree of Master of Engineering

Department of Computer and Telecommunication Engineering

College of Innovative Technology and Engineering,

Dhurakij Pundit University

2019



ใบรับรองสารนิพนธ์

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อสารนิพนธ์ ประสิทธิภาพการทำงานของ Multipath TCP ในระบบโครงข่ายส่วนบุคคล

เสนอโดย นายสรสิช ศรีเอี่ยมสะอาด

สาขาวิชา วิศวกรรมคอมพิวเตอร์และโทรคมนาคม

อาจารย์ที่ปรึกษาสารนิพนธ์ อาจารย์ ดร.ธนัญ จารุวิทยโกวิท

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบสารนิพนธ์แล้ว

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ)

.....กรรมการและอาจารย์ที่ปรึกษาสารนิพนธ์

(อาจารย์ ดร.ธนัญ จารุวิทยโกวิท)

.....กรรมการ

(อาจารย์ ดร.ชัยพร เขมะภาตะพันธ์)

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์รับรองแล้ว

.....คณบดีวิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

(ผู้ช่วยศาสตราจารย์ ดร.ณรงค์เดช กีรติพรานนท์)

วันที่ ....9.... เดือน ....ม.ค.... พ.ศ. ....2562....

หัวข้อสารนิพนธ์ ประสิทธิภาพการทำงานของ Multipath TCP ในระบบโครงข่ายส่วนบุคคล
ชื่อผู้เขียน สรสิช ศรีเอี่ยมสะอาด
อาจารย์ที่ปรึกษา ดร.ธนัญ จารุวิทยโกวิท
สาขาวิชา วิศวกรรมคอมพิวเตอร์และโทรคมนาคม
ปีการศึกษา 2561

บทคัดย่อ

TCP (Transport Control Protocol) ทำงานที่ชั้น Transport Layer ของ OSI ซึ่งทำหน้าที่ควบคุมการรับส่งข้อมูลระหว่างอุปกรณ์ต้นทาง (Source) และอุปกรณ์ปลายทาง (Destination) ของโครงข่าย เพื่อให้สามารถรับ/ส่งข้อมูลได้อย่างมีประสิทธิภาพมากที่สุด TCP ได้พัฒนาให้สามารถใช้งานหลายเส้นทาง (Multipath) ต่อ 1 การเชื่อมต่อ เพื่อเพิ่มความเร็วในการรับส่งข้อมูลให้สูงขึ้น โดยมีการกำหนดเป็นมาตรฐาน MPTCP (Multipath TCP)

งานวิจัยนี้ทดสอบประสิทธิภาพการทำงาน MPTCP บนอุปกรณ์จริงบนระบบโครงข่ายส่วนบุคคล โดยจะใช้งาน MPTCP ผ่านโครงข่าย LAN และ WiFi ในเวลาเดียวกัน โดยตั้งสมมุติฐานว่าเครื่องแม่ข่ายปลายทางไม่รองรับ MPTCP จึงจำเป็นต้องมีการใช้งาน MA-GW (Multinet Aggregate GateWay) เพื่อให้ MA-GW สร้างการเชื่อมต่อ TCP ไปยังเครื่องแม่ข่ายปลายทางแทน งานวิจัยนี้ตั้งสมมุติฐานว่าขนาดของบัฟเฟอร์ของ MA-GW และ UE (User Equipment) ตลอดจนวิธีควบคุมความคับคั่ง (Congestion control) มีผลต่อประสิทธิภาพการทำงานของ MPTCP งานวิจัยนี้จึงดำเนินการทดสอบโดยการปรับแต่งค่าขนาดของบัฟเฟอร์ และวิธีการควบคุมความคับคั่งของ UE และ MA-GW โดยวัดประสิทธิภาพการทำงานจากความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter)

ผลการทดสอบแสดงให้เห็นว่าการปรับขนาดบัฟเฟอร์ขนาด 16 MB และใช้วิธีควบคุมความคับคั่งแบบ CUBIC ที่ UE และ MA-GW จะให้ประสิทธิภาพของ MPTCP ดีที่สุด เมื่อเปรียบเทียบกับค่าตั้งต้นที่กำหนดจากระบบปฏิบัติการ Linux Mint 18 และ Linux CentOS 7 โดยสามารถเพิ่ม Throughput ได้ 32% และลด Download Time ได้ 18% ส่วน Delay และ Jitter มีความต่างแบบไม่มีนัยสำคัญ

Thematic Paper Title	Performance of Multipath TCP in Personal Network
Auther	Sorasich Sriaimsa-ard
Thematic Paper Advisor	Dr.Tanun Jaruvitayakovit
Department	Computer and Telecommunication Engineering
Academic Year	2018

Abstract

TCP (Transport Control Protocol) is one of the main protocols in the Transport Layer in OSI. It control receive/send data between Source and Destination for Network. The goal is to receive/send data as efficiently as possible. TCP has developed to be able to use multiple paths per session. This helps to increase link bandwidth to Multipath TCP.

This research examines the performance of MPTCP in private networks, IE. LAN and WiFi networks, by assuming that the destination server does not support MPTCP, therefore the MA-GW (Multinet Aggregate GateWay) is required. In order for MA-GW to establish a TCP connection to the destination server instead. This research hypothesized that the buffer size of MA-GW and UE (User Equipment) as well as the congestion control method had an effect on MPTCP's performance. This research aims to study the effect of the buffer size, congestion control scheme of the UE (User Equipment) and MA-GW on the performance of MPTCP. Measure performance from Throughput, Download Time, Delay and Jitter.

These research results show that buffer size to 16 MB and congestion control scheme to Cubic of the UE (User Equipment) and MA-GW on the increase performance of MPTCP. When compared to the default settings from Linux Mint 18 and Linux CentOS 7. Which can increase the throughput by 32% and reduce the download time by 18%. The delay and the jitter are insignificant differences.

กิตติกรรมประกาศ

การจัดทำสารนิพนธ์ฉบับนี้สำเร็จลุล่วงและบรรลุตามจุดประสงค์ได้อย่างสมบูรณ์ โดยได้รับความอนุเคราะห์อย่างสูงยิ่งจาก ดร.ธนัญ จารุวิทยโกวิท อาจารย์ที่ปรึกษา ที่ได้กรุณาให้การช่วยเหลือ ทั้งในด้านข้อคิดเห็น คำแนะนำ แนวทางการแก้ปัญหา และตรวจสอบแก้ไขข้อบกพร่องที่เกิดขึ้น ที่รวมไปถึงการสละเวลาอันมีค่าเพื่อให้คำปรึกษาในทุกๆ ช่วงเวลาการดำเนินงาน นับตั้งแต่เริ่มต้นสืบค้นเพื่อจัดตั้งหัวข้องานวิจัยจนกระทั่งงานเสร็จสมบูรณ์ ซึ่งข้าพเจ้ารู้สึกมีความซาบซึ้งเป็นอย่างยิ่งในความเอาใจใส่ที่ท่านมีให้ ทั้งในบทบาทของอาจารย์ที่ปรึกษา และในบทบาทของอาจารย์ประจำภาควิชา จึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

ขอขอบคุณคณะกรรมการทุกท่าน ที่ได้สละเวลาทำหน้าที่เป็นกรรมการสอบสารนิพนธ์ เพื่อให้คำแนะนำที่เป็นประโยชน์เป็นอย่างยิ่งทำให้งานวิจัยครั้งนี้มีคุณค่ามากยิ่งขึ้น และขอขอบคุณเจ้าหน้าที่ทุก ๆ ท่าน ซึ่งได้ช่วยอำนวยความสะดวกในการดำเนินเรื่องต่าง ๆ ให้เป็นไปตามระเบียบปฏิบัติได้อย่างถูกต้องตามขั้นตอนเป็นอย่างดี

คุณค่าและประโยชน์ใดๆ ที่อาจมีจากสารนิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบเป็นเครื่องบูชาพระคุณของบิดามารดาที่ให้กำเนิดและเลี้ยงดูให้การศึกษา ตลอดจนครูบาอาจารย์ และผู้ที่มีพระคุณทุกท่านที่มีส่วนในการวางรากฐานการศึกษาให้แก่ผู้วิจัย ตลอดจนผู้แต่งหนังสือหรือตำราทุกท่านที่ข้าพเจ้าได้นำมาอ้างอิงในสารนิพนธ์ฉบับนี้ และขอขอบคุณทุกๆ ความช่วยเหลือจากเพื่อนร่วมชั้นเรียนของข้าพเจ้าทุกท่าน โดยเฉพาะคุณปิยพงษ์ เคนเหลี่ยม และคุณไกรลาศ ศิลกุล ที่ได้เสียสละเวลาเพื่อแนะนำกลวิธีการดำเนินงาน รวมไปถึงการสนับสนุนข้อมูลทั้งด้านทฤษฎี และด้านปฏิบัติ ซึ่งนับเป็นประโยชน์ให้การทำงานสามารถมุ่งไปข้างหน้าอย่างสะดวกรวดเร็วขึ้น ท้ายสุดนี้ข้าพเจ้ามุ่งหวังเป็นอย่างยิ่งว่างานค้นคว้าฉบับนี้ จะสามารถสร้างประโยชน์ให้แก่ผู้ที่สนใจได้ไม่มากก็น้อย และหากมีข้อบกพร่องประการใดข้าพเจ้าขอน้อมรับไว้แต่เพียงผู้เดียว

สรสิช ศรีเอี่ยมสะอาด

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ฆ
บทคัดย่อภาษาอังกฤษ ..... ง
กิตติกรรมประกาศ ..... จ
สารบัญ ..... ฉ
สารบัญตาราง ..... ซ
สารบัญภาพ ..... ฌ
บทที่
1. บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 3
1.3 ขอบเขตของงานวิจัย 3
1.4 ประโยชน์ที่คาดว่าจะได้รับ 4
1.5 ตารางแผนการดำเนินงาน 4
2. แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง 5
2.1 ความแตกต่างระหว่าง TCP กับ MPTCP 5
2.2 การทำงานของ MPTCP 8
2.3 ตัวอย่างการทำงานของ MPTCP 9
2.4 SOCKS Proxy 11
2.5 FTP (File Transfer Protocol) 12
2.6 Wireshark 13
2.7 TCP Congestion Control 13
2.8 งานวิจัยที่เกี่ยวข้อง 14
3. ระเบียบวิจัย 18
3.1 แนวทางการวิจัยและพัฒนา 18
3.2 แผนการดำเนินงาน 19
3.3 ขั้นตอนและวิธีการดำเนินงาน 20

สารบัญ(ต่อ)

บทที่ **หน้า**

4. ผลการดำเนินงาน 28
4.1 วิเคราะห์ผลการทดสอบปรับ Buffer ที่ UE 29
4.2 วิเคราะห์ผลการทดสอบปรับ Buffer ที่ MPTCP Proxy 33
4.3 วิเคราะห์ผลการทดสอบปรับ Congestion Control 37
4.4 สรุปผลการวัดประสิทธิภาพการใช้งาน MPTCP 41
5. บทสรุป และข้อเสนอแนะ 45
5.1 สรุปผลการวิจัย 45
5.2 ข้อจำกัดของงานวิจัย และแนวทางแก้ไข 46
5.3 ข้อเสนอแนะและแนวทางการพัฒนาในอนาคต 47
บรรณานุกรม 48
ภาคผนวก 52
ประวัติผู้เขียน 60

สารบัญตาราง

ตารางที่ **หน้า**

1.1 แผนการดำเนินงาน 4
2.1 รายละเอียดของ MPTCP Options 9
2.2 ผลิตภัณฑ์ของ Shadowsocks Socks5 12
2.3 เปรียบเทียบความสามารถของ TCP Westwood และ TCP Cubic 14
2.4 เปรียบเทียบงานวิจัยนี้และงานวิจัยที่เกี่ยวข้อง 17
4.1 ผลการทดสอบเมื่อปรับ Buffer ที่ UE 29
4.2 ผลการทดสอบเมื่อปรับ Buffer ที่ MA-GW 33
4.3 ผลการทดสอบเมื่อปรับ Congestion Control 37

สารบัญภาพ

ภาพที่ **หน้า**

1.1 การเชื่อมต่อจาก UE ไปยังเครื่องแม่ข่ายโดยใช้ MA-GW 2

2.1 ตัวอย่างการทำงานของ TCP..... 6

2.2 กระบวนการทำงานของ MPTCP 6

2.3 ตัวอย่างการทำงานของ MPTCP..... 7

2.4 การสื่อสารระหว่าง MPTCP กับ TCP โดยใช้ MA-GW เป็นตัวช่วย 8

2.5 โครงสร้างการทำงานของ MPTCP..... 8

2.6 ตัวอย่างการทำงานของ MPTCP..... 9

3.1 การเชื่อมต่อจาก UE ไปยัง Server โดยผ่าน MPTCP Proxy 22

3.2 จำลองระบบสำหรับทดสอบบนอุปกรณ์จริง..... 22

3.3 Calculate conversation timestamps..... 24

3.4 แสดงคอลัมน์ timestamps 25

3.5 ตั้งค่า Packet Delay ให้แสดงในโปรแกรม Wireshark 25

3.6 Column ที่ใช้ในโปรแกรม Wireshark..... 26

3.7 ลำดับการวัดประสิทธิภาพของ UE และ MA-GW 27

4.1 แสดงความเร็วในการรับส่งข้อมูล (Mbps) กรณีปรับขนาด Buffer ของ UE 30

4.2 แสดงระยะเวลาในการดาวน์โหลด (Sec) กรณีปรับขนาด Buffer ของ UE 30

4.3 แสดงค่าความล่าช้าทางเวลา (μs) กรณีปรับขนาด Buffer ของ UE..... 31

4.4 แสดงค่าความแปรปรวนของความล่าช้าทางเวลา (ns)
กรณีปรับขนาด Buffer ของ UE 31

4.5 แสดงประสิทธิภาพความเร็วในการรับส่งข้อมูล (%)
กรณีปรับขนาด Buffer ของ UE 32

4.6 แสดงประสิทธิภาพระยะเวลาในการดาวน์โหลด (%)
กรณีปรับขนาด Buffer ของ UE 32

4.7 แสดงความเร็วในการรับส่งข้อมูล (Mbps)
กรณีปรับขนาด Buffer ของ MA-GW 34

4.8 แสดงระยะเวลาในการดาวน์โหลด (Sec)
กรณีปรับขนาด Buffer ของ MA-GW 34

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

4.9 แสดงค่าความล่าช้าทางเวลา (μs) กรณีปรับขนาด Buffer ของ MA-GW..... 35

4.10 แสดงค่าความแปรปรวนของความล่าช้าทางเวลา (ns)
กรณีปรับขนาด Buffer ของ MA-GW 35

4.11 แสดงประสิทธิภาพความเร็วในการรับส่งข้อมูล (%)
กรณีปรับขนาด Buffer ของ MA-GW 36

4.12 แสดงประสิทธิภาพระยะเวลาในการดาวน์โหลด (%)
กรณีปรับขนาด Buffer ของ MA-GW 36

4.13 แสดงความเร็วในการรับส่งข้อมูล (Mbps) กรณีปรับ Congestion Control..... 38

4.14 แสดงระยะเวลาในการดาวน์โหลด (Sec) กรณีปรับ Congestion Control 38

4.15 แสดงค่าความล่าช้าทางเวลา (μs) กรณีปรับ Congestion Control..... 39

4.16 แสดงค่าความแปรปรวนของความล่าช้าทางเวลา (ns)
กรณีปรับ Congestion Control..... 39

4.17 แสดงประสิทธิภาพความเร็วในการรับส่งข้อมูล (%)
กรณีปรับ Congestion Control..... 40

4.18 แสดงประสิทธิภาพระยะเวลาในการดาวน์โหลด (%)
กรณีปรับ Congestion Control..... 40

4.19 แสดง Throughput ของ LAN และ WiFi
กรณีทำงานแบบ TCP เปรียบเทียบกับ MPTCP 41

4.20 แสดงความเร็วในการรับส่งข้อมูล (%) จากทุกการทดสอบ 41

4.21 แสดงระยะเวลาในการดาวน์โหลด (%) จากทุกการทดสอบ 42

4.22 แสดง Buffer 16, 32 MB Retransmission บนระบบจริง 43

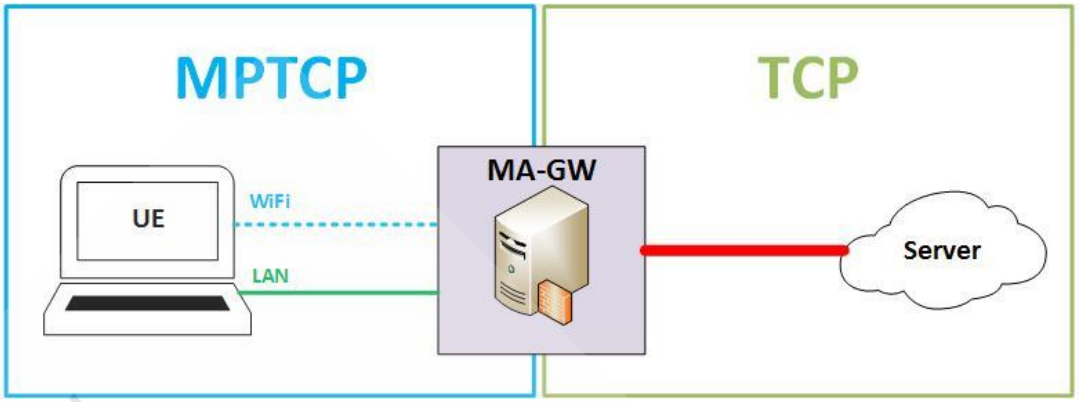
4.23 แสดงประสิทธิภาพการ Retransmit และ Fast Recovery ของ Cubic 44

4.24 แสดงประสิทธิภาพการ Retransmit และ Fast Recovery ของ Westwood..... 44

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การเชื่อมต่ออุปกรณ์เข้าสู่ระบบเครือข่ายในปัจจุบันสามารถเชื่อมต่อแบบใช้สายสัญญาณ เช่น สายทองแดง (RJ-45, RJ-11, Coaxial), สายใยแก้วนำแสง (FTTx) หรือการเชื่อมต่อแบบไร้สาย (WiFi, LTE) ซึ่งอุปกรณ์ IT ต่างๆ มักจะมีอินเตอร์เฟสสำหรับเชื่อมต่อระบบเครือข่ายอยู่หลายเส้นทาง (path) เพื่อเป็นทางเลือกในการใช้งาน เช่น โทรศัพท์เคลื่อนที่สามารถเชื่อมต่อระบบเครือข่ายได้โดยใช้ LTE หรือ WiFi แต่การสื่อสารระหว่างอุปกรณ์ต้นทาง (UE: โทรศัพท์เคลื่อนที่ หรือเครื่องคอมพิวเตอร์) และเครื่องแม่ข่าย (Server) มักใช้งานเพียงเส้นทางเดียว เพราะข้อจำกัดของโพรโตคอล TCP ที่ถูกออกแบบมาให้ใช้งานได้เพียงเส้นทางเดียวต่อ 1 การเชื่อมต่อ (session) ทำให้ช่องสัญญาณ (Bandwidth) ที่ได้รับจะขึ้นอยู่กับความสามารถของอินเตอร์เฟสที่อุปกรณ์นั้นๆ ได้ทำการเชื่อมต่ออยู่ นักวิจัยจึงมีแนวคิดที่จะเพิ่มช่องสัญญาณโดยการทำให้อุปกรณ์สามารถใช้งานอินเตอร์เฟสได้มากกว่า 1 เส้นทาง จึงทำให้เกิดโพรโตคอล Multipath TCP (MPTCP) ทำให้สามารถใช้งานได้หลายเส้นทางต่อ 1 การเชื่อมต่อ (session) ทำให้การเชื่อมต่อระหว่าง UE และเครื่องแม่ข่ายไม่ถูกจำกัดด้วยจำนวนเส้นทางการเชื่อมต่ออีกต่อไป แต่ MPTCP ยังมีข้อจำกัดทางในการใช้งานอยู่ เนื่องจาก UE และเครื่องแม่ข่าย จะต้องรองรับโพรโตคอล MPTCP ด้วย ปัญหาหลักในปัจจุบันคือ เครื่องคอมพิวเตอร์แม่ข่ายใน Internet ไม่รองรับโพรโตคอล MPTCP นักวิจัยจึงคิดวิธีการแก้ไขปัญหาโดยการนำอุปกรณ์ที่สามารถแลกเปลี่ยนการสื่อสารระหว่าง MPTC และ TCP มาติดตั้งระหว่างกลางของเครือข่าย MPTCP และ TCP ทำให้เกิดเป็นอุปกรณ์ MPTCP Proxy หรืออีกชื่อหนึ่งว่า Multinet Aggregate GateWay (MA-GW) ขึ้น โดยใช้เป็น Gateway ของอุปกรณ์ที่รองรับ MPTCP ดังแสดงในภาพที่ 1.1 เพื่อให้สามารถใช้งานช่องสัญญาณ (Bandwidth) ได้อย่างเต็มประสิทธิภาพ MA-GW ยังทำหน้าที่ในการ NAT (Network Address Translation) และคอยตรวจสอบเครื่องแม่ข่ายที่รองรับ MPTCP เพื่อให้สามารถใช้งาน MPTCP จาก UE จนไปถึงเครื่องแม่ข่ายได้เต็มประสิทธิภาพอีกด้วย



ภาพที่ 1.1 การเชื่อมต่อจาก UE ไปยังเครื่องแม่ข่ายโดยใช้ MA-GW

เนื่องจากความนิยมในปัจจุบันผู้ใช้งาน Internet นิยมเข้าถึง Content ประเภท Video เป็นหลักยิ่งคุณภาพของ Content สูงเท่าไหร่ ก็ยิ่งสามารถเพิ่มอรรถรสได้มากขึ้นเท่านั้น ในขณะเดียวกันขนาดไฟล์ที่ต้องดาวน์โหลดก็เพิ่มขึ้นเช่นกัน ซึ่งทำให้จำเป็นต้องใช้ขนาดช่องสัญญาณ (Bandwidth) จำนวนมาก โดยเฉพาะอย่างยิ่งวีดีโอ 4K ที่สามารถแสดงรายละเอียดของภาพและเสียงได้อย่างหน้าตื่นตาตื่นใจ และในปัจจุบันก็มีอุปกรณ์และ Content 4K เพิ่มมากขึ้น วิธีที่จะทำให้ Content เหล่านี้สามารถใช้งานได้อย่างลื่นไหล ผู้ให้บริการโทรศัพท์เคลื่อนที่จึงได้นำโพรโทคอล MPTCP มาใช้งานเพื่อเพิ่มประสิทธิภาพการทำงานให้กับระบบเครือข่ายของตนเอง โดยใช้งาน 3G/4G ร่วมกับ WiFi และใช้งานผ่าน MA-GW ทำให้สามารถลดปริมาณทราฟฟิค ของเครือข่าย 3G/4G ลงได้ และผู้ใช้งาน MPTCP ก็สามารถเข้าถึง Content ได้อย่างรวดเร็วขึ้น ผู้วิจัยจึงมีแนวคิดที่จะนำ MPTCP มาใช้กับเครือข่ายส่วนบุคคลเพื่อเพิ่มขนาดช่องสัญญาณ (Bandwidth) ให้กับอุปกรณ์ที่มีอินเตอร์เฟสสำหรับการเชื่อมต่อระบบเครือข่าย 2 เส้นทางขึ้นไป เช่น Notebook, Server เพื่อเพิ่มประสิทธิภาพการทำงานของอุปกรณ์ดังกล่าว

งานวิจัยนี้ทดสอบเพื่อประเมินประสิทธิภาพการทำงาน MPTCP บนอุปกรณ์จริงของระบบโครงข่ายส่วนบุคคล โดยใช้งาน MPTCP ผ่านโครงข่าย LAN และ WiFi ในเวลาเดียวกันเพื่อเพิ่มขนาดช่องสัญญาณ ทำให้สามารถเข้าถึง Content ได้อย่างรวดเร็ว และใช้งาน MA-GW ในการทดสอบระบบ เพราะในปัจจุบันเครื่องแม่ข่ายส่วนใหญ่ยังไม่รองรับการทำงานของ MPTCP และทดสอบปรับแต่งพารามิเตอร์ Buffer, Congestion Control ที่ส่งผลกระทบต่อประสิทธิภาพการทำงานของ MPTCP เพื่อเพิ่มประสิทธิภาพการทำงานให้ได้มากที่สุด โดยวัดประสิทธิภาพการทำงานของ MPTCP จากความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้า

ทางเวลา (Jitter) เปรียบเทียบหา Buffer, Congestion Control ค่าใดให้ประสิทธิภาพการทำงานของ LAN และ WiFi ได้ดีที่สุด

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 ศึกษาและใช้งาน MPTCP, MA-GW บนอุปกรณ์จริง โดยใช้งานอินเตอร์เฟส LAN และ WiFi ในเวลาเดียวกัน

1.2.3 เพื่อวัดประสิทธิภาพการใช้งาน MPTCP จากการดาวน์โหลดไฟล์ วัดค่า ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) ที่เกิดขึ้นในระบบ

1.2.3 ทดลองปรับแต่งค่าพารามิเตอร์เพื่อเพิ่มประสิทธิภาพการทำงานของ MPTCP

1.3 ขอบเขตของการวิจัย

1.3.1 งานวิจัยนี้จำลองระบบโครงข่ายส่วนบุคคลโดยใช้งาน MPTCP ผ่านโครงข่าย LAN และ WiFi ในเวลาเดียวกัน

1.3.2 งานวิจัยนี้ติดตั้งอุปกรณ์ MA-GW ในการสื่อสารกับเครื่องคอมพิวเตอร์แม่ข่ายที่รองรับแต่โปรโตคอล TCP

1.3.3 งานวิจัยนี้ศึกษาผลกระทบของขนาด Buffer ของอุปกรณ์ปลายทางด้านผู้ใช้งาน (User Equipment – UE) และอุปกรณ์ MA-GW โดยจะปรับค่าขนาดของ Buffer ด้านรับและด้านส่งของ MA-GW และค่าขนาดของ Buffer ด้านรับของ UE เพื่อหาค่าที่ทำให้ระบบสามารถทำงานได้ประสิทธิภาพสูงสุด

1.3.4 งานวิจัยนี้ศึกษาผลกระทบของวิธีควบคุมความคับคั่งที่มีผลต่อการทำงานของ MPTCP ในโครงข่ายส่วนบุคคล โดยเลือกใช้วิธีควบคุมความคับคั่งแบบ Cubic และ Westwood

1.3.5 งานวิจัยนี้วัดประสิทธิภาพการทำงานของ MPTCP โดยใช้โปรแกรม Wireshark วัดค่าผลรวมของอัตราการรับส่งข้อมูล (Throughput), ระยะเวลาในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) ที่เกิดขึ้นในระบบ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการวิจัยนี้คือ

1.4.1 ระบบโครงข่ายส่วนบุคคล ที่สามารถใช้งานโครงข่าย LAN และ WiFi ได้พร้อมกัน

1.4.2 ทราบถึงพารามิเตอร์ที่ส่งผลกระทบต่อประสิทธิภาพการทำงานของ MPTCP

1.4.3 สามารถเพิ่มประสิทธิภาพการทำงานของ MPTCP ได้

1.5 ตารางแผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

เดือน / งาน	มิ.ย.-ต.ค.60	ต.ค. -พ.ย.60	ธ.ค. 60 -พ.ค.61	มิ.ย. -ต.ค.61	พ.ย. -ธ.ค. 61
ศึกษาข้อมูลและทดลองทำระบบ MPTCP บน Virtual Machine					
จัดหาอุปกรณ์ที่จะจำลองระบบ					
ติดตั้งระบบ					
ทดสอบและวัดประสิทธิภาพการทำงานของ MPTCP					
สรุปผลการทดสอบการทำงานระบบ และจัดทำเล่มสารนิพนธ์					

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

MPTCP มีจุดมุ่งหมายให้ อุปกรณ์ IT ต่างๆ สามารถใช้งานได้หลายเส้นทางพร้อมกัน เพื่อใช้งานทรัพยากรที่มีได้อย่างเต็มประสิทธิภาพ และเป็นระบบสำรองเส้นทาง (Redundancy) ในบทนี้จะกล่าวถึงทฤษฎีที่นำมาใช้ในงานวิจัยและงานวิจัยที่เกี่ยวข้อง

2.1 ความแตกต่างระหว่าง TCP กับ MPTCP[1 2 3]

TCP เป็นโพรโทคอลที่ทำงานอยู่ในชั้น Transprot Layer ของ TCP/IP มีหน้าที่ในการสร้างการเชื่อมต่อแบบ End-to-End เพื่อให้คอมพิวเตอร์สารมารถรับส่งข้อมูลกันได้ ซึ่ง TCP จะทำงานอยู่ 3 อย่าง คือ

2.1.1 Connection Oriented สร้างการเชื่อมต่อระหว่างต้นทางและปลายทาง และควบคุมการส่งข้อมูลให้ถูกต้องครบถ้วน

2.1.2 Flow Control ควบคุมความเร็วในการรับส่งข้อมูลเพื่อลดการ Drop Packet ของอุปกรณ์ฝั่งรับ

2.1.3 Congestion Control ตรวจสอบความคับคั่งในเครือข่าย และควบคุมการส่งข้อมูลเพื่อลดการ Drop Packet ของอุปกรณ์ระหว่างทางให้มากที่สุด

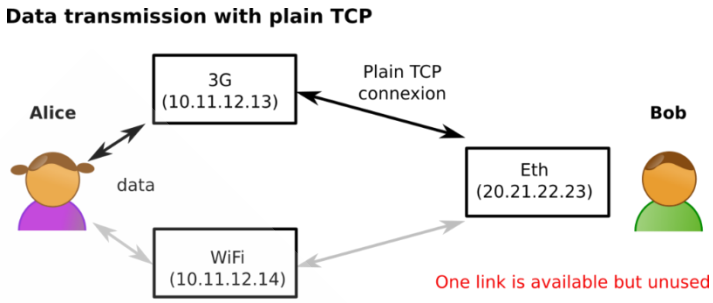
ข้อจำกัดของ TCP คือ การทำงานจะผูกกับ Interface ของอุปกรณ์ ตัวอย่างเช่น ดังภาพที่ 2.1 Alice มี 3G เป็นเส้นทางหลัก และ WiFi เป็นเส้นทางรอง ถ้า Alice ต้องการส่งข้อมูลไปหา Bob Alice จะต้องส่งข้อมูลผ่านทาง 3G เพียงเส้นทางเดียว เส้นทางของ WiFi จะไม่ถูกใช้งานจนกว่าจะปิดการใช้งาน 3G จากตัวอย่างสรุปได้ว่า TCP สามารถใช้งานได้เพียงแค่เส้นทางเดียว วิธีที่จะทำให้ Alice สามารถใช้งานได้หลายเส้นทางคือ นำโพรโทคอล MPTCP มาใช้งานแทน TCP

---

[1] adslthailand.com. (2560). เทคโนโลยี LTE-WiFi Aggregation. สืบค้นเมื่อ 25 กรกฎาคม 2560, จาก http://www.adslthailand.com/post/introduction-to-multipath-tcp-lte-WiFi-aggregation

[2] WIKIPEDIA. (2018). Multipath TCP. Retrieved October 19, 2018, from https://en.wikipedia.org/wiki/Multipath_TCP

[3] Olivier Bonaventure. (2013). Decoupling TCP from IP with Multipath TCP. Retrieved October 25, 2017, from http://multipath-tcp.org/data/MultipathTCP-netsys.pptx

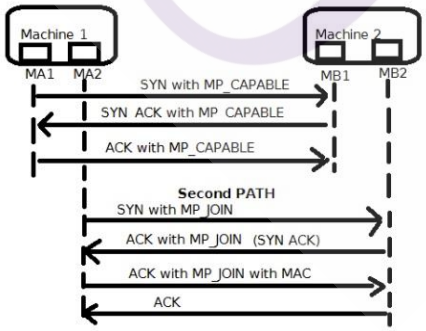


ภาพที่ 2.1 ตัวอย่างการทำงานของ TCP

ภาพประกอบจากเว็บไซต์เมื่อวันที่ 25 กรกฎาคม 2560 :

https://en.wikipedia.org/wiki/Multipath_TCP#/media/File:DifferenceTCP_MPTCP-en.png

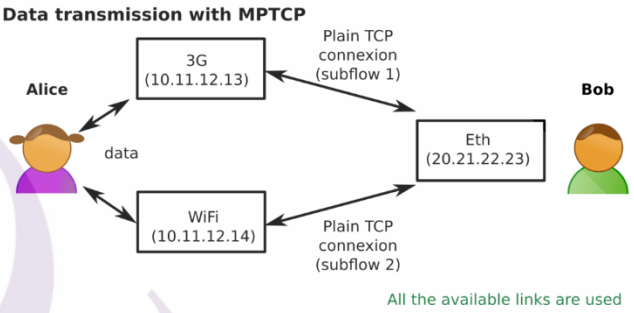
Multipath TCP (MPTCP) เป็นโพรโทคอลที่พัฒนามาจาก TCP ซึ่งทำงานอยู่ในชั้น Transport Layer ของ TCP/IP การทำงานจะสนใจแค่อุปกรณ์ต้นทางและอุปกรณ์ปลายทางเท่านั้น จึงเรียกได้ว่าเป็นการทำงานแบบ End-to-End โดย MPTCP จะเพิ่มในเรื่องของการใช้งานหลายเส้นทางโดยสามารถอธิบายการทำงานได้ดังนี้ ในระหว่างที่ อุปกรณ์ต้นทางและปลายทางสร้างการเชื่อมต่อ MPTCP จะเพิ่ม Option “MP_CAPABLE” เข้าไปใน Header ของ Segment TCP ดังภาพที่ 2.2 และส่งไปยังปลายทาง หากอุปกรณ์ปลายทางรองรับ MPTCP อุปกรณ์ปลายทางจะตอบกลับด้วย Option “MP_CAPABLE” หลังจากได้รับ ACK ของ Option “MP_CAPABLE” ทั้งสองฝั่งจะสร้าง Flow และ Subflow ของแต่ละเส้นทางขึ้น ทำให้ อุปกรณ์ต้นทางและปลายทางสามารถเพิ่มความเร็วในการรับส่งข้อมูลได้ตามความสามารถของเส้นทางนั้น ๆ หากอุปกรณ์ฝั่งใดไม่รองรับ MPTCP จะไม่มีการสร้าง subflow เกิดขึ้น



ภาพที่ 2.2 กระบวนการทำงานของ MPTCP

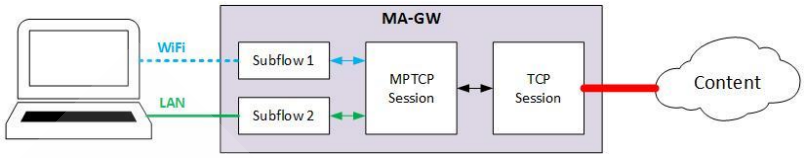
ภาพประกอบจากเว็บไซต์เมื่อวันที่ 25 กรกฎาคม 2560 :
http://www.adslthailand.com/post/introduction-to-multipath-tcp-lte-WiFi-aggregation

ตัวอย่าง ดังภาพที่ 2.3 Alice มี 3G เป็นเส้นทางหลัก และ WiFi เป็นเส้นทางรองและมีการใช้งาน MPTCP ถ้า Alice ต้องการส่งข้อมูลไปหา Bob Alice จะสามารถส่งข้อมูลผ่านทาง 3G และ WiFi ได้พร้อมกันทำให้ Alice มี Bandwidth เพิ่มขึ้นความเร็วในการส่งข้อมูลเพิ่มขึ้น



ภาพที่ 2.3 ตัวอย่างการทำงานของ MPTCP
ภาพประกอบจากเว็บไซต์เมื่อวันที่ 25 กรกฎาคม 2560 :
https://en.wikipedia.org/wiki/Multipath_TCP#/media/File:DifferenceTCP_MPTCP-en.png

MPTCP เป็นโพรโทคอลที่ใช้งานการเชื่อมต่อกับระบบเครือข่ายหลายๆ เส้นทางพร้อมกัน เพื่อเพิ่มความเร็วในการรับส่งข้อมูล แต่มีข้อจำกัดที่สำคัญคือ อุปกรณ์ต้นทางและปลายทางจะต้องรองรับการเชื่อมต่อแบบ MPTCP ในกรณีที่อุปกรณ์ต้นทางหรือปลายทางตัวใดตัวหนึ่งไม่รองรับ MPTCP จำเป็นจะต้องใช้อุปกรณ์ที่ทำหน้าที่เป็นตัวกลางในการสื่อสารของอุปกรณ์ที่ใช้งาน MPTCP และอุปกรณ์ที่ใช้งาน TCP คือ MPTCP Proxy หรืออีกชื่อหนึ่งว่า Multinet Aggregate GateWay (MA-GW) ดังภาพที่ 2.4 ทำหน้าที่จัดการเรื่อง NAT (Network Address Translation) และคอยตรวจสอบเครื่องแม่ข่ายที่รองรับ MPTCP เพื่อส่งผ่าน MPTCP ทำงานทำงานได้อย่างมีประสิทธิภาพ



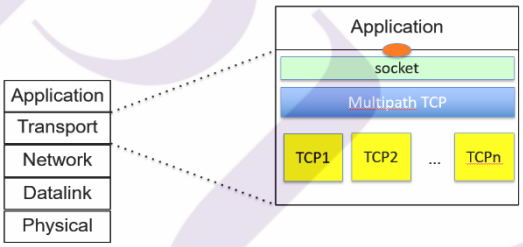
ภาพที่ 2.4 การสื่อสารระหว่าง MPTCP กับ TCP โดยใช้ MA-GW เป็นตัวช่วย

ภาพประกอบจากเว็บไซต์เมื่อวันที่ 25 กรกฎาคม 2560 :

https://www.citrix.com/blogs/2013/08/30/mptcp-netscaler-way/

2.2 การทำงานของ MPTCP[4 5 6]

การทำงานของ MPTCP ดังภาพที่ 2.5 ในชั้น Transport จะมีโพรโทคอล Multipath TCP (MPTCP) ซึ่งทำหน้าที่ในการตรวจสอบอุปกรณ์ต้นทางกับอุปกรณ์ปลายทางว่าทั้งคู่รองรับโพรโทคอล MPTCP หรือไม่ ถ้าทั้งคู่รองรับ MPTCP จะสร้างการเชื่อมต่อแบบ Multipath, จัดการ Subflow เส้นทางที่เพิ่มเข้ามา และจัดการยกเลิก Subflow ของเส้นทางที่หลุดการเชื่อมต่อ การทำงานของแต่ละ Subflow จะทำงานโดยใช้ โพรโทคอล TCP ในการสื่อสารระหว่างอุปกรณ์ต้นทางและอุปกรณ์ปลายทาง และโพรโทคอล MPTCP ยังทำหน้าที่ประสานงานกับ Socket เพื่อระบุ Session ที่เป็นเจ้าของข้อมูลอีกด้วย



ภาพที่ 2.5 โครงสร้างการทำงานของ MPTCP

ภาพประกอบจากเว็บไซต์เมื่อวันที่ 25 กรกฎาคม 2560 :

http://multipath-tcp.org/data/MultipathTCP-netsys.pptx

การตรวจสอบและสร้างการเชื่อมต่อ MPTCP จำเป็นต้องใช้ MPTCP Option ในการสื่อสารระหว่างอุปกรณ์ต้นทางและอุปกรณ์ปลายทาง ดังตารางที่ 2.1

---

[4] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for multipath TCP development", RFC6182 2011. http://multipath-tcp.org/data/MultipathTCP-netsys.pptx

[5] WIKIPEDIA. (2018). Multipath TCP. Retrieved October 19, 2018, from https://en.wikipedia.org/wiki/Multipath_TCP

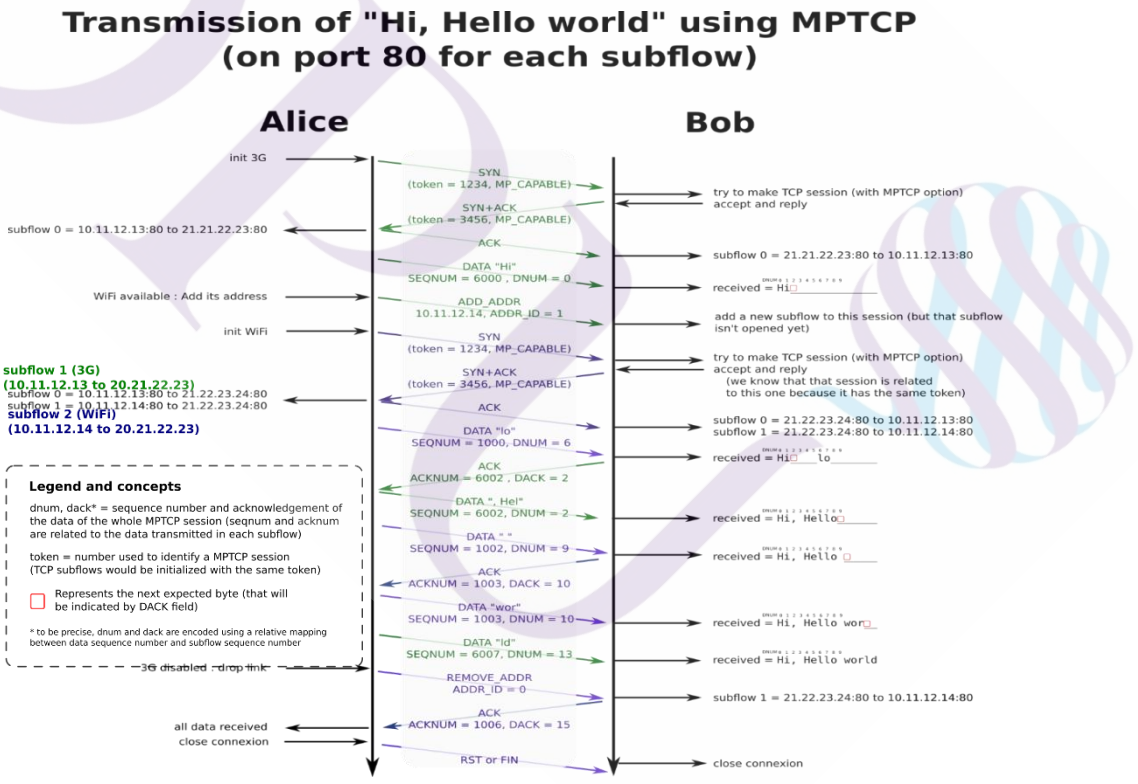
[6] Olivier Bonaventure. (2013). Decoupling TCP from IP with Multipath TCP. Retrieved October 25, 2017, from http://multipath-tcp.org/data/MultipathTCP-netsys.pptx

ตารางที่ 2.1 รายละเอียดของ MPTCP Options

Value	Symbol	Name
0x0	MP_CAPABLE	Multipath Capable
0x1	MP_JOIN	Join Connection
0x2	DSS	Data Sequence Signal (Data ACK and data sequence mapping)
0x3	ADD_ADDR	Add Address
0x4	REMOVE_ADDR	Remove Address
0x5	MP_PRIO	Change Subflow Priority
0x6	MP_FAIL	Fallback
0x7	MP_FASTCLOSE	Fast Close
0xf	(PRIVATE)	Private Use within controlled testbeds

จากเว็บไซต์เมื่อวันที่ 28 กรกฎาคม 2560 : https://en.wikipedia.org/wiki/Multipath_TCP

2.3 ตัวอย่างการทำงานของ MPTCP [7]



ภาพที่ 2.6 ตัวอย่างการทำงานของ MPTCP

[7] WIKIMEDIA. (2017). Transmission of "Hi, Hello world" using MPTCP. Retrieved July 28, 2017, from https://upload.wikimedia.org/wikipedia/commons/f/f3/MPTCP-session-en.png

ภาพประกอบจากเว็บไซต์เมื่อวันที่ 28 กรกฎาคม 2560 :
https://upload.wikimedia.org/wikipedia/commons/f/f3/MPTCP-session-en.png

ตัวอย่างการทำงานของ MPTCP จากภาพที่ 2.6 Alice ซึ่งเป็น UE ต้องการติดต่อกับ Bob โดยใช้ MPTCP ซึ่ง Alice จะมี 2 เส้นทาง คือ 3G และ WiFi จากตัวอย่าง 3G จะเป็นเส้นทางหลัก ซึ่งสามารถอธิบายได้ ดังนี้

2.3.1 เริ่มต้นการสื่อสาร 3G ของ Alice ทำ three-way handshake กับ Bob โดยใช้ SYN ที่มี Option “MP_CAPABLE” และ token ไว้ระบุ Session MPTCP เพื่อสอบถาม Bob ว่ารองรับ MPTCP หรือไม่

2.3.2 เมื่อ Bob ได้รับ SYN ที่มี Option “MP_CAPABLE” จาก Alice, Bob จะตอบ SYN+ACK ที่มี Option “MP_CAPABLE” และ token ที่ Bob สุ่มกลับไปหา Alice

2.3.3 เมื่อ Alice ได้รับ SYN+ACK ที่มี Option “MP_CAPABLE” ของ Bob, Alice จะสร้าง Subflow 0 ขึ้นมา และส่ง ACK ไปหา Bob เพื่อให้ Bob สร้าง Subflow 0 เช่นกัน

2.3.4 หลังจากกระบวนการ three-way handshake เสร็จ Alice สามารถส่ง ข้อมูลให้ Bob ได้ทันทีโดยข้อมูลที่ส่งไปจะมีค่า SEQNUM (sequence number ของ Subflow) ของเส้นทาง 3G และค่า DNUM (sequence number ของ MPTCP)

2.3.5 ระหว่างส่งข้อมูล MPTCP ของ Aliceจะส่ง Option ADD_ADDR เพื่อแจ้งให้ Bob ทราบว่าจะเพิ่ม Subflow ในการสื่อสาร

2.3.6 Alice ทำ three-way handshake ที่เส้นทาง WiFi เพื่อสร้างการเชื่อมต่อ โดยใช้ SYN ที่มี Option “MP_CAPABLE” และใช้ token เป็นเลขเดียวกับของเส้นทาง 3G

2.3.7 เมื่อ Bob ได้รับ Segment ที่มี token เดียวกับเส้นทาง 3G ของ Alice, Bob จะส่ง SYN+ACK โดยใช้ token เดิมที่ใช้ติดต่อกับ Alice

2.3.8 เมื่อ Alice ได้รับ SYN+ACK, Alice จะสร้าง subflow 1 และส่ง ACK ไปยัง Bob เพื่อให้ Bob สร้าง subflow 1 เช่นกัน หลังจากนี้เส้นทาง 3G และ WiFi ของ Alice สามารถใช้ Bandwidth ทั้งหมดเพื่อสื่อสารบน Session เดียวกันได้ทันที โดยที่ค่า SEQNUM จะแยกเป็นของแต่ละเส้นทาง แต่ค่า DNUM จะใช้งานร่วมกันเพื่อระบุ sequence number ของ MPTCP

2.3.9 เมื่อ Bob ต้องการขอ sequence number ของข้อมูลจาก Alice, Bob จะส่ง DACK ไป Alice จะสามารถส่งข้อมูลให้ Bob ได้อย่างถูกต้อง

2.3.10 สมมุติว่าเส้นทาง 3G ของ Alice Down เส้นทาง WiFi จะส่ง Option REMOVE_ADDR และเลขของ subflow ที่ Down ให้ Bob เมื่อ Bob ได้รับข้อความ Bob จะยกเลิก subflow ตามที่ Alice ส่งมา

2.4 SOCKS5 [8 9]

SOCKS5 เป็นโพรโตคอลที่ทำหน้าที่เป็น Proxy Server ซึ่งทำงานในระดับต่ำกว่า HTTP Proxy จึงทำให้ SOCKS5 มีความยืดหยุ่นในการใช้งานมากกว่า การทำงานของ SOCKS5 จะใช้ Handshake Protocol ในการแลกเปลี่ยนข้อมูลการเชื่อมต่อกับเครื่องลูกข่าย และพยายามคงสภาพ Header ที่เครื่องลูกข่ายสร้างไว้ ก่อนส่งต่อไปยังอินเทอร์เน็ต SOCKS5 จะใช้ Socket ในการติดตามการเชื่อมต่อ ทำให้ Header มีขนาดเล็ก อีกทั้งยังสามารถรับส่งข้อมูลประเภท UDP ได้อีกด้วย ในงานวิจัยนี้จะใช้โปรแกรม Shadowsocks Socks5 ใช้งานเป็น MA-GW ทำหน้าที่แยกและรวม Traffic ระหว่างระบบที่รองรับ MPTCP กับระบบ TCP เพื่อให้สามารถใช้งาน MPTCP ได้ในขณะที่ เครื่องแม่ข่ายไม่รองรับการทำงานของ MPTCP

การใช้งาน Shadowsocks Socks5 สามารถเลือกใช้งานการเชื่อมต่อกับเครื่องแม่ข่ายได้ 3 โหมด ดังนี้

- Direct ใช้ shadowsocks จัดการ Packet ทั้งหมด
- Redirect ใช้ Netfilter ของ Linux จัดการ Packet ทั้งหมด
- Tunnel ใช้ Tunnel ที่สร้างไว้กับเครื่องแม่ข่ายจัดการ Packet ทั้งหมด

Shadowsocks Socks5[10] มีการพัฒนาผลิตภัณฑ์ของตนเอง ออกมาหลายเวอร์ชั่น แสดงดังตารางที่ 2.2

2.5.1 shadowsocks: พัฒนาโดยใช้ภาษา Python ปัจจุบัน ค.ศ. 2018 ทาง Shadowsocks ได้ปิดบริการ Source Code เป็นที่เรียบร้อยแล้ว เพื่อให้ผู้สนใจหันไปใช้ผลิตภัณฑ์ ตัวอื่น

2.5.2 shadowsocks-libev พัฒนาโดยใช้ภาษา C สามารถรองรับทุกฟังก์ชั่นต่างๆ ที่ Shadowsocks Socks5 ได้ออกแบบมา แต่มีข้อจำกัดที่ ภาษา C ที่นำมาพัฒนาไม่สามารถทำงานบน CPU Multicore ได้อย่างเต็มประสิทธิภาพ

2.5.3 shadowsocks-go พัฒนาโดยใช้ภาษา Go เพื่อแก้ปัญหา การทำงานของ CPU Multicore ให้สามารถทำงานได้อย่างเต็มประสิทธิภาพ ซึ่งฟังก์ชั่นในการทำงาน จะมีแค่ส่วนจัดการผู้ใช้เป็นหลัก

2.5.4 go-shadowsocks2 พัฒนาโดยใช้ภาษา Go ซึ่งฟังก์ชั่นในการทำงาน จะเน้นไปทาง Core Network ที่จำเป็น เช่น Redirect mode, Tunnel mode

---

[8] WIKIPEDIA. (2018). SOCKS. Retrieved September 30, 2018, from https://en.wikipedia.org/wiki/SOCKS

[9] Margaret Rouse. (2005). SOCKS. Retrieved September 30, 2018, from http://whatis.techtarget.com/definition/socks

[10] shadowsocks.org. (2018). shadowsocks. Retrieved October 19, 2018, from https://shadowsocks.org/en/spec/Implementations.html

จากผลิตภัณฑ์ทั้งหมดของ Shadowsocks Socks5 ผู้วิจัยได้เลือก go-shadowsocks2 มาใช้งาน เนื่องจากรองรับเครื่องคอมพิวเตอร์ใช้ CPU Multicore และในงานวิจัยนี้จะทดลองโดยใช้ Redirect mode เนื่องจากสามารถควบคุมการทำงานได้โดยใช้ iptables ในการจัดการ

ตารางที่ 2.2 ผลิตภัณฑ์ของ Shadowsocks Socks5

	ss	ss-libev	ss-go	go-ss2
TCP Fast Open	✓	✓	✗	✗
Multiuser	✓	✓	✓	✗
Management API	✓	✓	✗	✗
Redirect mode	✗	✓	✗	✓
Tunnel mode	✓	✓	✗	✓
UDP Relay	✓	✓	✓	✓
AEAD ciphers	✓	✓	✗	✓
Plugin	✗	✓	✗	✗

จากเว็บไซต์เมื่อวันที่ 1 สิงหาคม 2561 : https://shadowsocks.org/en/spec/Implementations.html

2.5 FTP(File Transfer Protocol) [11] [12]

FTP(File Transfer Protocol) ถูกออกแบบมาเพื่อใช้ในการโอนถ่ายไฟล์เป็นหลัก มีความปลอดภัยเพราะมีการตรวจสอบสิทธิ์ก่อนใช้งาน เหมาะสมกับการโอนถ่ายไฟล์จำนวนมาก แต่มีข้อเสียคือ การโอนถ่ายไฟล์ไม่มีการเข้ารหัส ทำให้มีโอกาสถูกดักจับข้อมูลได้ตลอดเวลา FTP จะมีการใช้งาน Port อยู่ 2 Port คือ Port 21 ใช้สำหรับรับส่งคำสั่งควบคุมการใช้งาน FTP (Control Connection) และ Port 20 หรือ Random Port ใช้สำหรับรับส่งข้อมูล (DATA Connection)

งานวิจัยนี้ใช้ FTP ในการทดสอบดาวน์โหลดข้อมูลเพื่อวัดประสิทธิภาพการทำงานของ MPTCP ในด้านความเร็วในการรับส่งข้อมูล, เวลาที่ใช้ในการดาวน์โหลด, ค่าความล่าช้าทางเวลา และค่าความแปรปรวนของความล่าช้าทางเวลา

---

[11] พิศาล พิทยาธุรวิวัฒน์. (2555). ติดตั้งระบบเครือข่าย Linux Server ภาคปฏิบัติ. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

[12] จตุชัย แพงจันทร์, อนุโชต วุฒิพรพงษ์, และ อรรณพ ขันธิกุล (บรรณาธิการ). (2551). เจาะระบบ Network 2nd Edition. นนทบุรี : ไอดีซีฯ.

2.6 Wireshark[13] [14]

Wireshark คือ โปรแกรมสำหรับวิเคราะห์ packet ในระบบเครือข่าย จะทำงานโดยการดักจับข้อมูลโดยใช้ pcap บันทึกข้อมูลจาก NIC (Network Interface Card) นำข้อมูลมาแสดงผลเพื่อให้เห็นแผนภาพการทำงานโดยรวม โปรแกรม Wireshark มีข้อมูลโครงสร้างโพรโทคอลที่มีอยู่ในปัจจุบันเป็นจำนวนมาก ทำให้สามารถแสดงข้อมูลออกมาได้อย่างละเอียด สามารถจับข้อมูลในระบบเครือข่าย และอ่านข้อมูล packet จากไฟล์ดิบมาวิเคราะห์ได้ สามารถดักจับข้อมูลได้ทั้ง Ethernet, IEEE 802.11, PPP และ loopback ใช้งานได้ทั้งบน GUI และ command line (TShark)และ filter ข้อมูลได้ โปรแกรมสามารถติดตั้งได้หลาย platform ทั้ง Linux, Unix หรือ Window

งานวิจัยนี้ใช้ Wireshark ร่วมกับโปรแกรม LiberOffice Calc ในการดักจับข้อมูลวิเคราะห์ประสิทธิภาพการทำงานของ MPTCP ในด้านความเร็วในการรับส่งข้อมูล, เวลาที่ใช้ในการดาวน์โหลด, ค่าความล่าช้าทางเวลา และค่าความแปรปรวนของความล่าช้าทางเวลา

2.7 TCP Congestion Contorl[15] [16] [17] [18]

TCP Westwood เป็น Congestion Control ที่ออกแบบมาเพื่อแก้ไขการกู้คืน three duplicate ACKs เพื่อหลีกเลี่ยงการลด cwnd เกิดความจำเป็น โดยดูจาก Bandwidth ของระบบที่มีอยู่ ซึ่งเหมาะกับการใช้งานบนระบบการเชื่อมต่อแบบไร้สาย (WiFi, LTE) TCP Westwood ใช้หลักการทำงาน Slow Start และ Congestion Avoidance แบบเดียวกับ TCP Reno คือ

- Slow Start มีการเพิ่ม cwnd แบบ exponential
- Congestion Avoidance มีการเพิ่ม cwnd แบบ linear

ในช่วง Congestion Avoidance เมื่อเกิด Duplicate ACKs หรือ Nois จะพยายามไม่ลด cwnd ให้ต่ำกว่า transit capacity แต่เมื่อเกิด Timeout จะเริ่มส่ง cwnd ใหม่จาก 1

TCP Cubic พัฒนามาจาก TCP BIC มีการนำ additive increase และ binary search increase มาผสมกันเพื่อให้เกิดความสามารถในการปรับขนาด cwnd ได้ดีขึ้นและเพิ่ม TCP

---

[13] Ripmilla. (2553). มารู้จักกับตัวดักจับข้อมูลหรือ Sniffer กันเถอะ. สืบค้นเมื่อ 27 มีนาคม 2018, จาก https://www.freeware.in.th/windows/1756

[14] Saixiii. (2560). Wireshark-sniffer. สืบค้นเมื่อ 27 มีนาคม 2018, จาก https://saixiii.com/wireshark-sniffer/

[15] CLAUDIO CASETTI, MARIO GERLA, MARIO GERLA, M.Y. SANADIDI and REN WANG. TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks. in: Kluwer Academic Publishers. Manufactured in The Netherlands (2002).

[16] ผศ.ดร.มัชฌิกา อ่องแตง. (2017) CT518: Advanced Internet Engineering Class 7: Newer TCP Variants.

[17] Wei Bao, Vincent W.S. Wong, and Victor C.M. Leung. A Model for Steady State Throughput of TCP CUBIC. in: IEEE Globecom 2010.

[18] ผศ.ดร.มัชฌิกา อ่องแตง. (2017) CT518: Advanced Internet Engineering Class 6: Newer TCP Variants.

friendliness, RTT fairness ซึ่ง TCP Cubic สามารถทำงานได้ดีกับ large bandwidth-delay product นอกจากนี้การเติบโตของ cwnd ยังกำหนดขึ้นแบบ Real-time เพื่อให้การเติบโตของ cwnd ไม่ขึ้นกับ RTT เพื่อให้เกิด RTT fairness

เปรียบเทียบการทำงานของ Congestion Control ในด้าน Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery เกี่ยวกับอังกอริทึมที่นำมาใช้ระหว่าง TCP Cubic และ TCP Westwood ดังตารางที่ 2.3

ตารางที่ 2.3 เปรียบเทียบความสามารถของ TCP Westwood และ TCP Cubic

	TCP Cubic	TCP Westwood
Slow Start	additive increase and binary search increase	exponential
Congestion Avoidance	additive increase and binary search increase	linear
Fast Retransmit	additive increase and binary search increase	ssthresh ประมาณ transit capacity
Fast Recovery	additive increase and binary search increase	ssthresh ประมาณ transit capacity

2.8 งานวิจัยที่เกี่ยวข้อง

2.8.1 A Measurement-based Study of MultiPath TCP Performance over Wireless Networks[19]

ทดสอบวัดประสิทธิภาพจาก UE ซึ่งในกรณีนี้คือโทรศัพท์เคลื่อนที่ ใช้งาน MPTCP สื่อสารกับเครื่องแม่ข่ายโดยตรง อุปกรณ์ในการทดสอบจะใช้งาน WiFi ส่วนบุคคล, WiFi สาธารณะ และ LTE จากผู้ให้บริการใน USA จำนวน 3 ค่าย จำนวน 2 การเชื่อมต่อ โดยการทดสอบจะรับส่งข้อมูลด้วยไฟล์ขนาดที่แตกต่างกัน และเปรียบเทียบประสิทธิภาพการทำงานระหว่าง Single Path กับ Multipath สรุปได้ว่ากรณี Single Path: WiFi ส่วนบุคคลและ WiFi สาธารณะ ให้ค่า Latency ต่ำกว่า LTE ทั้ง 3 ค่าย กรณี Multipath ให้ค่า Latency น้อยที่สุดเมื่อเปรียบเทียบกับ Single

[19] Yung-Chih Chen, Yeon-sup Lim, Richard J. Gibbens, Erich M. Nahum, Ramin Khalili and Don Towsley. A Measurement-based Study of MultiPath TCP Performance over Wireless Networks. In Proceedings of the 2013 conference on Internet measurement conference, pages 455-468. ACM, 2013.

Path ทั้งหมด และไฟล์ขนาดเล็ก Single Path จะทำงานได้ดีกว่า Multipath ส่วนไฟล์ขนาดใหญ่ Multipath ทำงานได้ดีกว่า

ข้อดีของงานวิจัย

- ทดสอบใช้งาน WiFi และ LTE บนอุปกรณ์จริง
- ทดสอบใช้งานทั้ง WiFi ส่วนบุคคล และ WiFi สาธารณะ
- ทดสอบใช้งาน LTE กับผู้ให้บริการ 3 ค่าย
- มีการเปรียบเทียบระหว่าง Single Path และ Multipath
- มีการทดลองปรับ Congestion Control

ข้อจำกัดของงานวิจัย

- ไม่มีการใช้ MA-GW เมื่อนำมาใช้งานอาจจะไม่ได้ Bandwidth ตามที่ทดสอบมา
- ขนาดไฟล์ที่ทดสอบใหญ่สุดที่ 32 MB ทำให้ไม่สามารถวิเคราะห์ประสิทธิภาพการโหลดไฟล์ระยะยาวได้

2.8.2 An In-depth Understanding of Multipath TCP on Mobile Devices: Measurement and System Design[20]

ทดสอบวัดประสิทธิภาพอุปกรณ์ Mobile ใช้งาน MPTCP สื่อสารผ่าน MA-GW ไปยัง CDN (Content Delivery Network) ทดสอบด้วยข้อมูลขนาดเล็กและข้อมูลขนาดใหญ่โดยสนใจที่ RTT, Bandwidth และลักษณะของเส้นทางที่แตกต่างกัน (LTE กับ WiFi) สรุปได้ว่า การดาวโหลดไฟล์ขนาดเล็ก ถ้าอินเทอร์เฟสหลักมีค่า RTT น้อยจะให้ประสิทธิภาพดีที่สุด การดาวโหลดไฟล์ขนาดใหญ่ ต้องดูที่ค่า RTT ของทุกอินเทอร์เฟส ถ้าค่า RTT ใกล้เคียงกันจะให้ประสิทธิภาพดีที่สุด เพราะถ้าค่า RTT แตกต่างกันมากข้อมูลที่ส่งไปจะสะสมในอินเทอร์เฟสที่ RTT น้อยเป็นหลักซึ่งถ้าเส้นทางนั้นเป็นเส้นทางที่ให้ Bandwidth น้อย ประสิทธิภาพจะตกลงทันที

ข้อดีของงานวิจัย

- การทดสอบมีการแยก Gateway ที่ให้ Latency ของ LTE ดี กับ Gateway ที่ให้ Latency ของ WiFi ดี ในการทดสอบ
- การทดสอบมีการแยก CDN ที่ให้ Latency ของ LTE ดี กับ CDN ที่ให้ Latency ของ WiFi ดี ในการทดสอบ

ข้อจำกัดของงานวิจัย

---

[20] Ashkan Nikravesh, Yihua Guo, Feng Qian, Z. Morley Mao and Subhabrata Sen. An In-depth Understanding of Multipath TCP on Mobile Devices: Measurement and System Design. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, pages 189-201. ACM, 2016.

- ขนาดไฟล์ที่ทดสอบใหญ่สุดที่ 50 MB ทำให้ไม่สามารถวิเคราะห์ประสิทธิภาพการโหลดไฟล์ระยะยาวได้

2.8.3 More is Better? Measurement of MPTCP based Cellular Bandwidth Aggregation in the Wild[21]

ทดสอบประสิทธิภาพ MPTCP ในฝั่งชนโดยใช้ 4G และ 3G ของผู้ให้บริการ 4 ราย ใช้ USB Hub และ Air Card ในการทดสอบโดยงานวิจัยนี้จะติดตั้ง MA-GW บน Cloud Providers จำนวน 3 รายสำหรับเปรียบเทียบระหว่าง Cloud ภายในประเทศ กับ Cloud ที่มีผู้ใช้งานยอมรับจากทั่วโลก และตั้ง Content Server เก็บไฟล์ขนาดต่างๆ สำหรับใช้ในการทดสอบความเร็วในการรับส่งข้อมูล ในการทดสอบจะมีการเปรียบเทียบ MPTCP กับ TCP ปกติ สรุปได้ว่าเครือข่าย 4G ใช้งานแบบ 4-MPTCP และ 3G ใช้งานแบบ 4-MPTCP ทั้งคู่เมื่อมีการใช้งาน MPTCP ในการรับส่งข้อมูลขนาดเล็กจะให้ประสิทธิภาพการทำงานที่ช้ากว่าการใช้งาน TCP ปกติ เนื่องจากเครือข่าย 4G และ 3G มีค่า Latency ที่ค่อนข้างสูง ทำให้กระบวนการสร้างการเชื่อมต่อแบบ MPTCP กลายเป็นการเพิ่ม Latency ให้กับระบบโดยรวมแทน การรับส่งข้อมูลที่มีขนาดใหญ่ ขนาดที่ทดสอบสูงสุดคือ 64MB MPTCP ยังทำงานช้ากว่า TCP ปกติอยู่ แต่ในภาพรวมความเร็วในการทำงานของ MPTCP เริ่มใกล้เคียงกับ TCP ปกติ ซึ่งสามารถประเมินได้กว่า ถ้าขนาดไฟล์ใหญ่กว่า 64MB จะทำให้ MPTCP ทำงานได้เร็วกว่า TCP ปกติ การเปรียบเทียบระหว่าง Cloud ภายในประเทศ กับ Cloud ที่มีผู้ใช้งานยอมรับจากทั่วโลกสรุปได้ว่า Bandwidth จาก USB Hub และ Air Card ไปถึง Proxy มีผลต่อประสิทธิภาพการทำงานโดยรวม และโดยภาพรวมการใช้งาน HTTP Proxy มีส่วนช่วยในการเข้าถึงข้อมูลมากที่สุด และการเปลี่ยน Congestion Control ก็มีส่วนช่วยให้ทำงานเร็วขึ้นเช่นกัน

ข้อดีของงานวิจัย

- ทดสอบใช้งาน LTE กับผู้ให้บริการ 4 ค่าย
- ทดสอบโดยใช้ Cloud ภายในประเทศและ ต่างประเทศ
- มีการทดสอบในสถานที่แตกต่างกัน (เดินในสถานที่ผู้คนแออัด, รถเมล์ รถไฟใต้ดิน)
- มีการทดลองปรับ Congestion Control

ข้อจำกัดของงานวิจัย

- ขนาดไฟล์ที่ทดสอบใหญ่สุดที่ 64 MB ทำให้ไม่สามารถวิเคราะห์ประสิทธิภาพการโหลดไฟล์ระยะยาวได้

---

[21] Zhixiong Niu, Zhi Wang, Hong Xu, Chuan Wu and Francis C.M. Lau. More is Better? Measurement of MPTCP based Cellular Bandwidth Aggregation in the Wild. In 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). IEEE, 2016

ตารางที่ 2.4 เปรียบเทียบงานวิจัยนี้และงานวิจัยที่เกี่ยวข้อง

	งานวิจัยที่ 1	งานวิจัยที่ 2	งานวิจัยที่ 3	งานวิจัยนี้
พารามิเตอร์ที่ใช้วัดประสิทธิภาพ				
วัดประสิทธิภาพจาก Delay	✓	✓	✓	✓
วัดประสิทธิภาพจาก Jitter				✓
วัดประสิทธิภาพจาก Bandwidth	✓	✓	✓	✓
Interface ที่ใช้ทดสอบ				
ใช้งาน 3G/4G	✓	✓	✓	
ใช้งาน WiFi ส่วนบุคคล	✓	✓		✓
ใช้งาน LAN ส่วนบุคคล				✓
ทดสอบระบบ				
ทดสอบโดยใช้ MA-GW		✓	✓	✓
ทดสอบโดยใช้ Cloud สาธารณะ	✓	✓	✓	
ทดสอบ Download File < 100 MB	✓	✓	✓	
ทดสอบ Download File > 100 MB				✓
ทดสอบเพิ่มประสิทธิภาพจาก Buffer				✓
ทดสอบเปลี่ยน Congestion Control	✓		✓	✓

จากตารางที่ 2.4 งานวิจัยที่เกี่ยวข้องที่ 1, 2 และ 3 จะมีการทดสอบประสิทธิภาพ MPTCP โดยใช้เครือข่าย 3G/4G กับ WiFi บนโครงข่ายอินเทอร์เน็ตเพื่อวัดประสิทธิภาพของโครงข่ายสาธารณะเป็นหลัก แต่ยังไม่มีการปรับแต่งค่า Buffer เพื่อให้ใช้งาน MPTCP มีประสิทธิภาพที่ดีขึ้น และไฟล์ที่นำมาทดสอบมีขนาดเล็ก ทำให้ไม่สามารถวิเคราะห์ประสิทธิภาพการโหลดไฟล์ระยะยาวได้ งานวิจัยนี้จะทดสอบประสิทธิภาพการทำงานของ MPTCP ในระบบโครงข่ายส่วนบุคคลโดยใช้งาน LAN และ WiFi เพื่อเพิ่มประสิทธิภาพของ MPTCP ด้วยวิธีปรับเปลี่ยน Buffer และ Congestion Control งานวิจัยทำทดสอบระบบโดยใช้อุปกรณ์จริง โดยไฟล์ที่นำมาทดสอบจะใช้ขนาด 5.4 GB เพื่อทำให้สามารถวิเคราะห์ประสิทธิภาพการโหลดไฟล์ระยะยาวได้ ผลที่ได้จากการทดสอบ จะสามารถนำไปใช้ในการตัดสินใจในการทำระบบ MPTCP บนระบบโครงข่ายส่วนบุคคลได้

บทที่ 3
ระเบียบวิธีวิจัย

บทที่ 3 จะอธิบายแนวทางการวิจัย, แผนการดำเนินงาน และขั้นตอนในการวิจัย โดยมุ่งเน้นที่จะใช้งาน โพรโทคอล MPTCP ให้มีประสิทธิภาพที่ดีที่สุด

3.1 แนวทางการวิจัยและพัฒนา

งานวิจัยนี้จะวัดประสิทธิภาพการทำงานของ MPTCP ของเครือข่ายส่วนบุคคลซึ่งเป็นการจำลองระบบขึ้นมาบนอุปกรณ์จริงในส่วนของโพรโทคอล MPTCP และอุปกรณ์เสมือน (VMware) ในส่วนโพรโทคอล TCP ซึ่งประกอบด้วยอุปกรณ์ต้นทาง (UE), Gateway (MA-GW), อุปกรณ์ปลายทาง (FTP Server) ทดสอบโดยให้ UE ดาวน์โหลดไฟล์ขนาด 5.4 GB จากเครื่องแม่ข่าย ผ่านโพรโทคอล FTP และทดสอบปรับแต่งค่า Buffer และ Congestion Control ของอุปกรณ์ โดยมีความคาดหวังว่าจะสามารถเพิ่มประสิทธิภาพการทำงานของ MPTCP ได้ สำหรับค่าที่จะนำมาเปรียบเทียบเพื่อวัดประสิทธิภาพ ได้แก่ ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) การจำลองระบบจะเป็นดังภาพที่ 1.1

ในปัจจุบันเครื่องแม่ข่ายของผู้ให้บริการ Service ต่างๆ มักไม่รองรับโพรโทคอล MPTCP เพราะฉะนั้น การออกแบบระบบของงานวิจัยนี้จะคำนึงถึงกรณีนี้ด้วย การจำลองระบบของงานวิจัยนี้จึงตั้งติดตั้ง MA-GW เพื่อเป็นตัวกลางในการสื่อสารของอุปกรณ์ UE ที่ใช้งาน MPTCP และเครื่องแม่ข่าย (FTP Server) ที่ใช้งาน TCPโดยระบบที่จำลองขึ้นมาจะมีความต้องการของอุปกรณ์ดังนี้

3.1.1 UE จำลองบนอุปกรณ์จริง

- ระบบปฏิบัติการ: Linux Mint 18 64 bit ติดตั้ง MPTCP-kernal
- Network
  - WiFi AC speed 867 Mbps
  - LAN speed 1,000 Mbps
- Software: go-shadowsocks2, iptables, File zilla, wireshark

3.1.2 MA-GW จำลองบนอุปกรณ์จริง

- ระบบปฏิบัติการ: Linux CentOS 7 64 bit ติดตั้ง MPTCP-kernal
- Network
  - LAN speed 1,000 Mbps จำนวน 2 path
  - LAN Virtual Switch 2,000 Mbps
- Software: go-shadowsocks2

3.1.3 FTP Server จำลองบนอุปกรณ์เสมือน (VMware)

- ระบบปฏิบัติการ: Linux CentOS 7 64 bit ติดตั้ง MPTCP-kernal
- Network: LAN Virtual Switch 2,000 Mbps
- Software: vsftpd

3.1.4 Access Point สามารถรองรับการทำงานมาตรฐาน WiFi AC speed 867 Mbps

ขั้นตอนการเก็บผลจะเก็บผลการใช้งาน MPTCP จำนวน 3 รอบ ของการปรับแต่ง 1 ครั้งโดยจะปรับแต่งค่า Buffer และ Congestion Control และใช้โปรแกรม Wireshark ในการดักจับข้อมูลการทำงาน และใช้ LiberOffice Calc ในการคำณวนเพื่อหาค่า ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) เพื่อนำมาวิเคราะห์ สรุปความสามารถในการทำงานของ MPTCP สามารถเพิ่มประสิทธิภาพการทำงานของ MPTCP ได้หรือไม่ ค่าพารามิเตอร์ตัวไหนที่ส่งผลต่อประสิทธิภาพในการทำงานและสรุปผลการทดลองของงานวิจัย

3.2 แผนการดำเนินงาน

3.2.1 ศึกษาข้อมูลและทดลองทำระบบ MPTCP บน Virtual Machine

ค้นหาข้อมูลเกี่ยวกับ MPTCP, MA-GW และ Sock5 เพื่อหาวิธีในการสื่อสารระหว่าง UE และ FTP Server, กระบวนการทำงาน, การสร้างการเชื่อมต่อ, การ Implement, ข้อจำกัด, พารามิเตอร์ที่สามารถเพิ่มประสิทธิการสื่อสาร, กำหนดวิธีในการทดสอบระบบและจำลองระบบ MPTCP บน Virtual Machine ให้สามารถทำงานได้ โดยมีความคาดหวังว่าจะสามารถเพิ่มประสิทธิภาพการทำงานของ MPTCP ได้

3.2.2 จัดหาอุปกรณ์

จัดหาอุปกรณ์ที่ต้องใช้ในงานวิจัยเพื่อจำลองระบบที่สื่อสารแบบ MPTCP โดยมี ความต้องการพื้นฐานตามเกณฑ์ที่กำหนด ดังนี้

- UE ต้องรองรับ WiFi AC 867 Mbps บนคลื่น 5 GHz และ LAN Gigabit

- MA-GW และFTP Server ต้องรองรับ LAN Gigabit จำนวน 2 port และสร้าง Virtual Machine เชื่อมต่อระหว่าง MA-GW และ FTP Server ต้องมี Bandwidth มากกว่า 2 Gbps

- Access Point ต้องรองรับ WiFi AC AC 867 Mbps

3.2.3 ติดตั้งระบบ

จัดทำระบบตามแผนที่ได้ออกแบบไว้ ประกอบด้วย การเชื่อมต่อระหว่างอุปกรณ์, การติดตั้งโปรแกรมที่ต้องใช้ในการทดสอบระบบ, การตั้งค่าโปรแกรม และทดสอบการทำงานเพื่อ ให้สามารถวัดประสิทธิภาพการทำงานได้ตามที่วางแผนไว้

3.2.4 วัดประสิทธิภาพการทำงานของ MPTCP

ทดสอบโดยให้ UE ดาวน์โหลดไฟล์ขนาด 5.4 GB จาก FTP Server ผ่านโพรโทคอล FTP และทดสอบปรับค่า Buffer และ Congestion Control ของอุปกรณ์โดยมีความคาดหวังว่าจะสามารถเพิ่มประสิทธิภาพการทำงานของ MPTCP ได้ สำหรับค่าที่จะนำมาเปรียบเทียบเพื่อวัดประสิทธิภาพ ได้แก่ ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter)

3.2.5 สรุปผลการจำลองระบบและประโยชน์ที่ได้รับ

ประเมินผลการทดลอง และสรุปผล พร้อมชี้แจ้งเกี่ยวกับประสิทธิภาพการทำงานของ MPTCP, การปรับแต่งค่า Buffer และ Congestion Control สามารถเพิ่มประสิทธิภาพการทำงานได้หรือไม่ และลักษณะงานที่เหมาะสมในการใช้งาน

3.3 ขั้นตอนและวิธีการดำเนินงาน

3.3.1 ศึกษาข้อมูลและทดลองทำระบบ MPTCP บน Virtual Machine

วัตถุประสงค์ของการวิจัยนี้ เพื่อศึกษาการใช้งาน MPTCP, ทราบถึงข้อจำกัดการทำงานของ MPTCP, วัดประสิทธิภาพของ MPTCP และเพิ่มประสิทธิภาพการทำงานของ MPTCP ขั้นตอนนี้เป็นการศึกษาหาองค์ความรู้ต่างๆ งานวิจัยนี้ได้รวบรวม แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง ที่จำเป็นต้องรู้ไว้ในบทที่ 2 เมื่อได้องค์ความรู้ต่างๆ ผู้วิจัยจึงได้จำลองระบบขึ้นมาบนอุปกรณ์เสมือน (Virtual Machine) ใช้ Topology ดังภาพที่ 3.1 เพื่อทดสอบใช้งาน MPTCP, MA-GW และ Sock5 ในการสื่อสารระหว่าง UE และ FTP Server เพื่อหาวิธีที่ดีที่สุดในการทดสอบระบบและการเก็บข้อมูลเพื่อนำมาวิเคราะห์ ก่อนที่จะทดสอบบนอุปกรณ์จริง ซึ่งวิธีที่ใช้ในการทดสอบสามารถสรุปได้ดังนี้

- วิธีทดสอบระบบและเก็บข้อมูล จะทดสอบโดยให้ UE ดาวน์โหลดไฟล์ขนาด 5.4 GB จากเครื่องแม่ข่าย ผ่านโพรโทคอล FTP และเก็บข้อมูลขณะดาวน์โหลดไฟล์ที่ UE โดยใช้โปรแกรม Wireshark บันทึกข้อมูลเป็นไฟล์ pcap สำหรับใช้ในการวิเคราะห์ข้อมูลต่อไป

- วิธีการวิเคราห์ข้อมูล จะนำไฟล์ pacp ที่ได้มาคำนวณเพื่อหา ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) เปรียบเทียบ ระหว่างก่อนและหลังปรับแต่งค่า Buffer และ Congestion Control เพื่อวัดประสิทธิภาพการทำงานของ MPTCP ว่าสามารถทำงานได้ดีขึ้นหรือแย่ลง และสรุปผลเกี่ยวกับปัจจัยที่ส่งผลต่อการทำงานของ MPTCP

3.3.2 สั่งซื้ออุปกรณ์ที่จะจำลองระบบ

อุปกรณ์ที่ใช้จำลองระบบต้องเป็นไปตามความต้องการพื้นฐานตามเกณฑ์ที่กำหนดในหัวข้อ 3.2.2 จัดหาอุปกรณ์ โดยอุปกรณ์ที่ใช้ในงานวิจัยมีดังนี้

1. Labtop acer aspire VX5[1]
   - CPU: Core i7-7700-HQ
   - RAM: 20 GB
   - Hard disk: SSD M.2 Samsung 960 EVO 250 GB
   - OS: Mint Linux 18 (UE)
   - LAN: 1 Gbps
   - Wireless: Qualcomm QCA61x4A 802.11ac
2. PC
   - CPU: Core i5-3570
   - RAM: 8 GB
   - Hard disk: SSD Samsung 840 PRO 128 GB
   - OS: Cent OS linux 7 Desktop 64bit (MA-GW)
     - Install VMware Workstation Pro 14
     - VMware: Cent OS linux 7 Desktop 64bit (FTP Server)
   - LAN: 1 Gbps * 2
3. Access Point UBIQUITI UniFi (UAP-AC-LITE)[2]

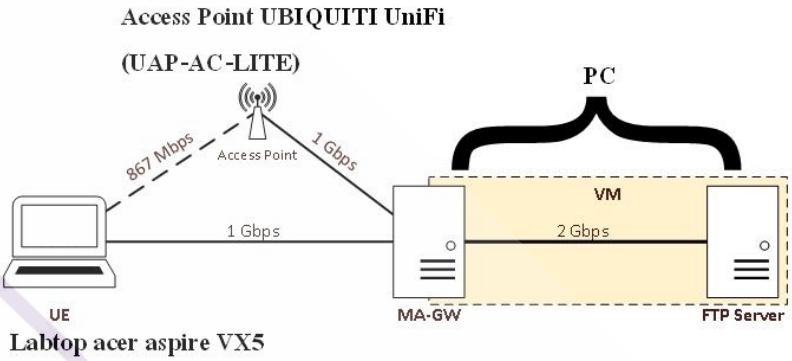
[1] Acer Inc. (2018). Aspire VX 15. สืบค้นเมื่อ 27 ตุลาคม 2561, จาก https://www.acer.com/ac/th/TH/content/series/aspirevx15

[2] Ubiquiti Networks, Inc. (2018). UniFi AP AC LITE. สืบค้นเมื่อ 27 ตุลาคม 2561, จาก https://www.ubnt.com/unifi/unifi-ap-ac-lite/

- Network Interface: 10/100/1000 Ethernet Port

- Standard: 802.11ac 867 Mbps

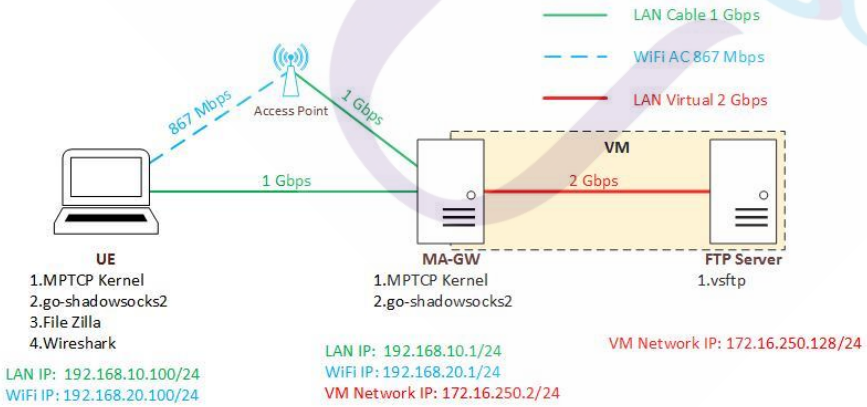
การเชื่อมต่อระหว่างอุปกรณ์เป็นดังภาพที่ 3.1



ภาพที่ 3.1 อุปกรณ์ที่ใช้ในการจำลองระบบ

3.3.3 ติดตั้งระบบ

จัดทำระบบสำหรับทดสอบระบบบนอุปกรณ์จริงจะประกอบด้วยเครื่องคอมพิวเตอร์จำนวน 2 เครื่อง คือ UE และ MA-GW สำหรับทดสอบการใช้งานของ MPTCP ที่ MA-GW จะมีการใช้งาน VMware จำลองคอมพิวเตอร์เสมือน 1 เครื่อง สำหรับใช้งานเป็น FTP Server ซึ่งไม่มีการติดตั้ง MPTCP เพื่อทดสอบกรณีที่ อุปกรณ์ปลายทางไม่รองรับการทำงานของ MPTCP การเชื่อมต่อระหว่างอุปกรณ์ต่างๆ และการติดตั้งโปรแกรมของแต่ละอุปกรณ์ IP ของแต่ละ Interface สามารถสรุปได้ดังภาพที่ 3.2



ภาพที่ 3.2 จัดทำระบบสำหรับทดสอบระบบบนอุปกรณ์จริง

UE จะใช้งาน Laptop acer aspire VX5 ซึ่งมี Interface สำหรับเชื่อมต่อระบบเครือข่ายอยู่ 2 เส้นทาง คือ LAN มีความเร็วในการรับส่งข้อมูล 1 Gbps เชื่อมต่อระบบเครือข่ายโดยใช้ IP: 192.168.10.100/24 และ WiFi AC มีความเร็วในการรับส่งข้อมูล 867 Mbps เชื่อมต่อระบบเครือข่ายโดยใช้ IP: 192.168.20.100/24 ติดตั้ง MPTCP Kernel เพื่อให้สามารถใช้งาน MPTCP ได้ ติดตั้ง go-shadowsocks2 เพื่อใช้งาน Sock5 ในการเชื่อมต่อกับ MA-GW ติดตั้ง File Zilla เพื่อใช้ทดสอบดาวน์โหลดไฟล์ขนาด 5.4 GB และติดตั้ง Wireshark เพื่อใช้ในการเก็บบันทึกข้อมูลการทำงานของ MPTCP เพื่อนำมาวิเคราห์การทำงานในภายหลัง

MA-GW จะใช้งาน PC ซึ่งมี Interface สำหรับเชื่อมต่อระบบเครือข่ายอยู่ 3 เส้นทาง คือ LAN มีความเร็วในการรับส่งข้อมูล 1 Gbps จำนวน 2 เส้นทางเชื่อมต่อระบบเครือข่ายโดยใช้ IP: 192.168.10.1/24 และ IP: 192.168.20.1/24 และ LAN Virtual กำหนดความเร็วในการรับส่งข้อมูล 2 Gbps เชื่อมต่อระบบเครือข่ายโดยใช้ IP: 172.16.250.2/24 ติดตั้ง MPTCP Kernel เพื่อให้สามารถใช้งาน MPTCP ได้ ติดตั้ง go-shadowsocks2 เพื่อใช้งาน Sock5 โดย MA-GW จะเป็นตัวแทนในการสื่อสารระหว่างอุปกรณ์ใช้งาน MPTCP กับ อุปกรณ์ใช้งาน TCP

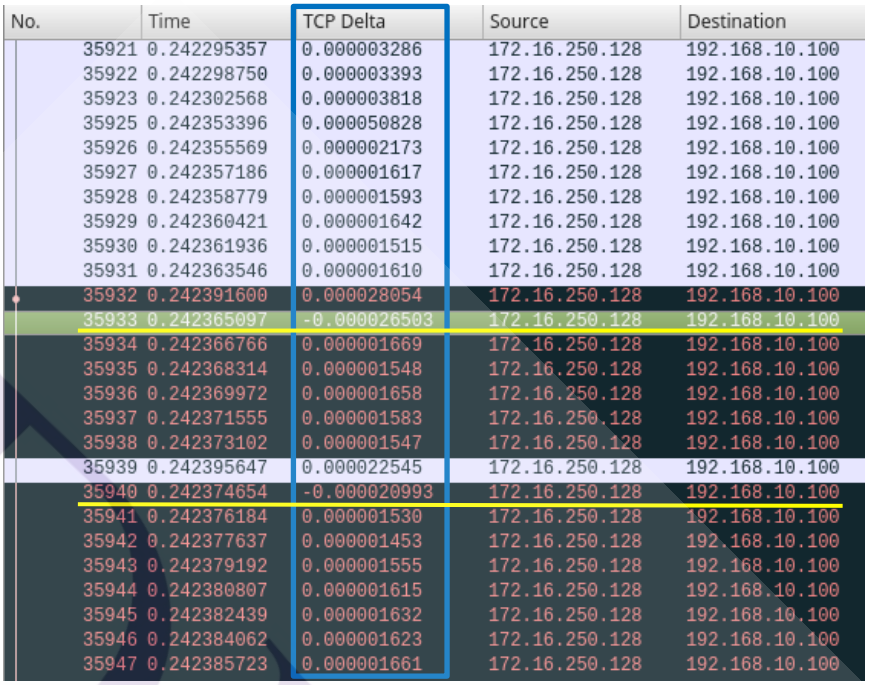
FTP Server จะใช้งาน VM ซึ่งมี Interface สำหรับเชื่อมต่อระบบเครือข่ายอยู่ 1 เส้นทาง คือ LAN Virtual กำหนดความเร็วในการรับส่งข้อมูล 2 Gbps เชื่อมต่อระบบเครือข่ายโดยใช้ IP: 172.16.250.128/24 ติดตั้ง vsftp สำหรับใช้เป็น FTP Server และมีการสร้างไฟล์ขนาด 5.4 GB สำหรับให้ UE ดาวน์โหลด

Access Point จะใช้งาน UBIQUITI UAP-AC-LITE ซึ่งมี Interface สำหรับเชื่อมต่อระบบเครือข่ายอยู่ 2 เส้นทาง คือ LAN มีความเร็วในการรับส่งข้อมูล 1 Gbps และ WiFi มีความเร็วในการรับส่งข้อมูล 867 Mbps ซึ่งทั้ง 2 เส้นทางเชื่อมต่อระบบเครือข่าย 192.168.20.0/24 โดยที่ UE จะเชื่อมต่อผ่านทาง WiFi และ MA-GW จะเชื่อมต่อผ่านทาง LAN

3.3.4 วัดประสิทธิภาพการทำงานของ MPTC

การวัดประสิทธิภาพการทำงานในงานวิจัยนี้จะใช้งานโปรแกรม Wireshark ในการดักจับข้อมูลการสื่อสาร และบันทึกข้อมูลเป็นไฟล์ pcap โดยทางผู้วิจัยจะกรองข้อมูล Packet Retransmission ออกทั้งหมดเพราะบาง Packet โปรแกรม Wireshark คำนวณ TCP Delta เป็นจำนวนติดลบ ดังภาพที่ 3.3 ซึ่งทำให้ไม่สามารถนำไปคำนวน Jitter ได้ โดย Packet ที่ตัดออกคิดเป็นประมาณ 0.0025% ของ Packet ทั้งหมด

การกรองข้อมูลที่นำมาวิเคราะห์ จะกรองเฉพาะข้อมูล FTP เท่านั้น ที่ Packet “FTP-Data” ใช้คำสั่ง Conversation Filter -> TCP และเพิ่ม Filter “ not tcp.analysis.retransmission and not tcp.analysis.fast_retransmission” เพื่อนำ Packet Retransmission ออกทั้งหมด



No.	Time	TCP Delta	Source	Destination
35921	0.242295357	0.000003286	172.16.250.128	192.168.10.100
35922	0.242298750	0.000003393	172.16.250.128	192.168.10.100
35923	0.242302568	0.000003818	172.16.250.128	192.168.10.100
35925	0.242353396	0.000050828	172.16.250.128	192.168.10.100
35926	0.242355569	0.000002173	172.16.250.128	192.168.10.100
35927	0.242357186	0.000001617	172.16.250.128	192.168.10.100
35928	0.242358779	0.000001593	172.16.250.128	192.168.10.100
35929	0.242360421	0.000001642	172.16.250.128	192.168.10.100
35930	0.242361936	0.000001515	172.16.250.128	192.168.10.100
35931	0.242363546	0.000001610	172.16.250.128	192.168.10.100
35932	0.242391600	0.000028054	172.16.250.128	192.168.10.100
35933	0.242365097	-0.000026503	172.16.250.128	192.168.10.100
35934	0.242366766	0.000001669	172.16.250.128	192.168.10.100
35935	0.242368314	0.000001548	172.16.250.128	192.168.10.100
35936	0.242369972	0.000001658	172.16.250.128	192.168.10.100
35937	0.242371555	0.000001583	172.16.250.128	192.168.10.100
35938	0.242373102	0.000001547	172.16.250.128	192.168.10.100
35939	0.242395647	0.000022545	172.16.250.128	192.168.10.100
35940	0.242374654	-0.000020993	172.16.250.128	192.168.10.100
35941	0.242376184	0.000001530	172.16.250.128	192.168.10.100
35942	0.242377637	0.000001453	172.16.250.128	192.168.10.100
35943	0.242379192	0.000001555	172.16.250.128	192.168.10.100
35944	0.242380807	0.000001615	172.16.250.128	192.168.10.100
35945	0.242382439	0.000001632	172.16.250.128	192.168.10.100
35946	0.242384062	0.000001623	172.16.250.128	192.168.10.100
35947	0.242385723	0.000001661	172.16.250.128	192.168.10.100

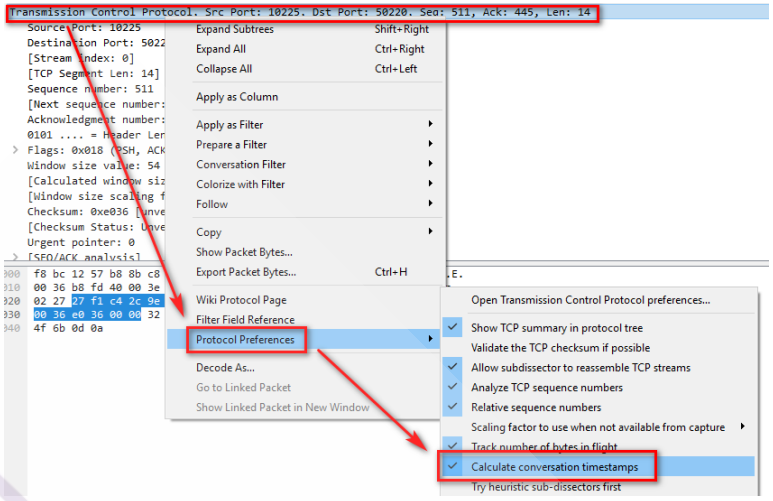
ภาพที่ 3.3 Packet Retransmission คำนวณ Delay ผิดพลาด

หลังจาก Filter ข้อมูลเสร็จ Packet แรกที่เริ่ม SYN จะตั้งค่า Set Time Reference เพื่อตั้งค่าเวลาในแต่ละ Packet ให้เป็น 0 สำหรับคำนวณ ระยะเวลาในการดาวน์โหลดไฟล์ทั้งหมดได้ง่ายขึ้น

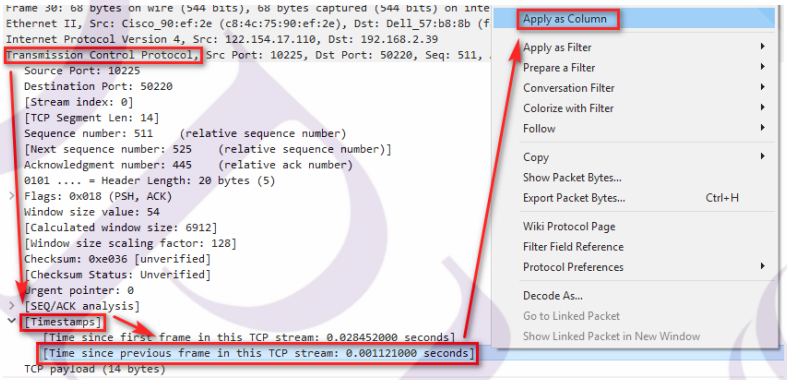
ในการวัด Delay[3] ของแต่ละ Packet จะวัดจาก Timestamps หลักการคือ เมื่ออุปกรณ์ต้นทางส่ง Packet จะมีการลง Timestamps1 เมื่ออุปกรณ์ปลายทางได้รับ Packet จะมีการลง Timestamps2 เมื่อนำ Timestamps2 - Timestamps1 = Packet Delay (TCP Delta) ซึ่งโปรแกรม Wireshark มีฟังก์ชั่นคำนวณอยู่โดยเปิดที่ Transmission Control Protocol -> Protocol Preferences -> Calculate conversation timestamps ดังภาพที่ 3.4 จะทำให้ Transmission Control Protocol มี Timestamps แสดงขึ้น ในหัวข้อย่อยให้เลือก Time since previous frame in this TCP stream -> Apply as Column ดังภาพที่ 3.5 และเปลี่ยนชื่อเป็น TCP Delta เพื่อแสดง Packet Delay

---

[3] Laura Chappell. (2013). Wireshark Tip 22: Find Delays with TCP "Calculate Conversation Timestamps". Retrieved September 10, 2017, from https://www.youtube.com/watch?v=QqKAnZnHss0



ภาพที่ 3.4 แสดงคอลัมน์ timestamps



ภาพที่ 3.5 ตั้งค่า Packet Delay ให้แสดงในโปรแกรม Wireshark

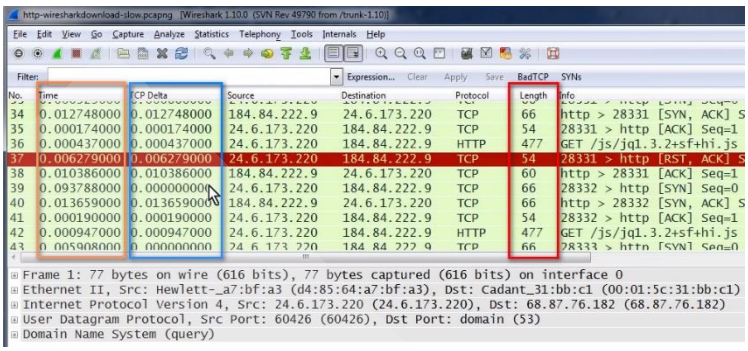
ข้อมูลที่จำเป็นสำหรับนำมาคำนวณในงานวิจัยนี้ได้แก่ข้อมูลใน Column: Time, TCP Delta, Length(Packet Length:Byte) ดังภาพที่ 3.6 โดยจะใช้ในการคำนวณหา ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) ซึ่งมีสูตรในการทำนวณดังนี้

$$\text{Throughput} = \frac{(\text{Total Packet Length} * 8)}{\text{Download time}}$$

$$\text{Download time} = \text{Last Time} - \text{First Time}$$

$$\text{Delay} = \frac{\text{Total TCP Delta}}{\text{Number of packets}} \qquad \text{... (3.1)}$$

$$\text{Jitter} = \frac{\left(\sum_{i=First\ TCP\ Delta}^{j=Last\ TCP\ Delta}[\text{i}- \text{Delay}]^2\right)}{(\text{Number of packets}-1)}$$

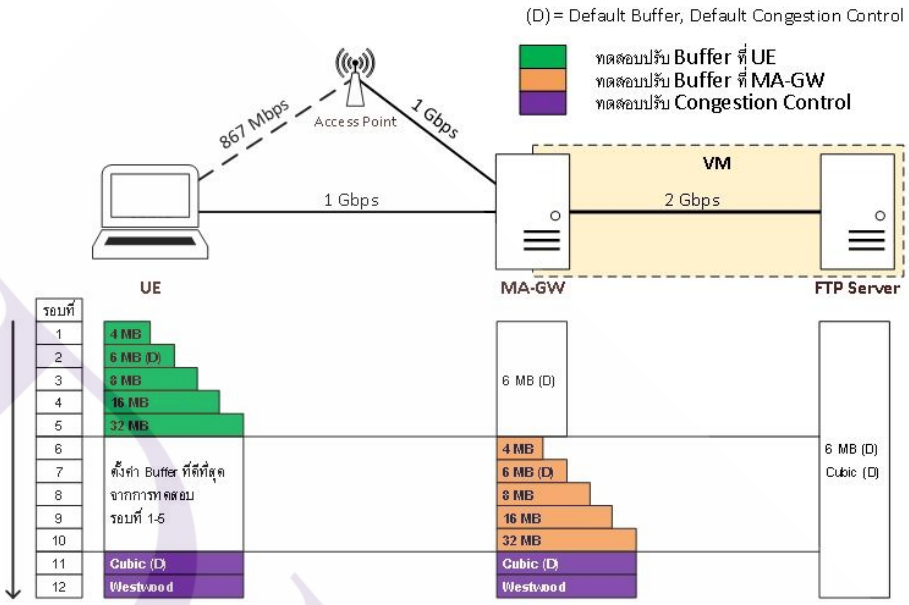


ภาพที่ 3.6 Column ที่ใช้ในโปรแกรม Wireshark

ขั้นตอนในการวัดประสิทธิภาพ จะทดสอบโดยการ

- ปรับ Buffer UE ขนาด 4MB, 6MB(Default), 8MB, 16MB, 32MB
- ตั้งค่า Buffer UE ที่ให้ประสิทธิภาพดีที่สุด
- ปรับ Buffer MA-GW ขนาด 4MB, 6MB(Default), 8MB, 16MB, 32MB
- ตั้งค่า Buffer MA-GW ที่ให้ประสิทธิภาพดีที่สุด
- ปรับ Congestion Control ระหว่าง Cubic (Default) กับ westwood
- สรุปหาประสิทธิภาพที่ได้จากการปรับ Buffer ว่าการทำงานเป็นอย่างไรเมื่อเปรียบเทียบกับก่อนปรับ
- เปรียบเทียบว่า Congestion Control ประเภทไหนให้ประสิทธิภาพดีกว่า

โดยสามารถสรุปขั้นตอนการทดสอบได้ดังภาพที่ 3.7



ภาพที่ 3.7 ลำดับการวัดประสิทธิภาพของ UE และ MA-GW

3.3.5 สรุปผลการจำลองระบบและประโยชน์ที่ได้รับ

ประเมินผลการทดลอง และสรุปผล พร้อมชี้แจ้งเกี่ยวกับประสิทธิภาพการทำงานของ MPTCP, การปรับแต่งค่า Buffer และ Congestion Control สามารถเพิ่มประสิทธิภาพการทำงานได้หรือไม่และ ลักษณะงานที่เหมาะสมในการใช้งาน

บทที่ 4
ผลการดำเนินงาน

การดำเนินงานครั้งนี้มีวัตถุประสงค์เพื่อวัดประสิทธิภาพการใช้งาน MPTCP จากดาวน์โหลดไฟล์ ขนาด 5.4 GB และทดสอบปรับ Buffer ขนาด 4 MB, 6 MB (Default), 8 MB, 16 MB และ 32 MB ที่ UE จากนั้น Fix Buffer ที่ให้ประสิทธิภาพดีที่สุดที่ UE และปรับ Buffer ที่ MA-GW ขนาด 4 MB, 6 MB (Default), 8 MB, 16 MB และ 32 MB เพื่อสรุปหาประสิทธิภาพที่ได้จากการปรับ Buffer ว่าการทำงานดีขึ้นหรือแย่ลง เมื่อเปรียบเทียบกับก่อนปรับ ทดสอบโดยการปรับ Congestion Control ระหว่าง Cubic (Default) ออกแบบมาสำหรับ เพิ่มความเร็วบนเครื่อข่ายที่มี large bandwidth-delay product กับ westwood ออกแบบมาเพื่อเพิ่มความเร็วในการกู้คืนเมื่อเกิด Duplicate ACKs ของ large bandwidth-delay product เพื่อเปรียบเทียบว่า Congestion Control ประเภทไหนให้ประสิทธิภาพมากกว่ากัน โดยมีความคาดหวังว่าจะสามารถเพิ่มประสิทธิภาพการทำงานของ MPTCP ได้ สำหรับค่าที่จะนำมาเปรียบเทียบเพื่อวัดประสิทธิภาพ ได้แก่ ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter)

ขั้นตอนในการเก็บผลการทดสอบเมื่อติดตั้งอุปกรณ์สำหรับทดสอบตามที่ได้ออกแบบไว้ในบทที่ 3 ที่อุปกรณ์ UE จะใช้โปรแกรม Wireshark ในการดักจับ Packet ตั้งแต่ก่อนเริ่มกระบวนการส่งข้อมูล จนกระทั่งกระบวนการส่งข้อมูลเสร็จสิ้น ซึ่งการปรับพารามิเตอร์ 1 ครั้งจะดักจับ Packet 3 รอบ และนำมาหาค่าเฉลี่ย เพื่อให้ผลการทดสอบที่ได้เป็นค่าถูกต้องที่สุด หลังจากนั้นจะนำผลที่ได้มาคำนวณซึ่งโปรแกรม Wireshark สามารถคำนวณ Throughput และ Download time ของการดาวน์โหลดไฟล์ได้ แต่มีข้อจำกัดที่ไม่สามารถคำนวณ Delay และ Jitter ของการดาวน์โหลดไฟล์ได้ เพราะฉะนั้น ผู้วิจัยจึงใช้วิธีในการ Export ข้อมูลของโปรแกรม Wireshark เป็นไฟล์ CSV เพื่อคำนวณหา ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) โดยใช้สูตรในการคำนวณตามสมการที่ 3.1

เมื่อคำนวณเสร็จจะนำผลการทดสอบไปเปรียบเทียบ เพื่อหาว่าปรับพารามิเตอร์แบบไหนสามารถให้ผลลัพธ์การทำงานได้ดีที่สุด โดยในบทนี้จะแบ่งเป็นหัวข้อดังนี้

4.1 วิเคราะห์ผลการทดสอบปรับ Buffer ที่ UE

4.2 วิเคราะห์ผลการทดสอบปรับ Buffer ที่ MA-GW

4.3 วิเคราะห์ผลการทดสอบปรับ Congestion Control

4.4 สรุปผลการวัดประสิทธิภาพการใช้งาน MPTCP

4.1 วิเคราะห์ผลการทดสอบปรับ Buffer ที่ UE

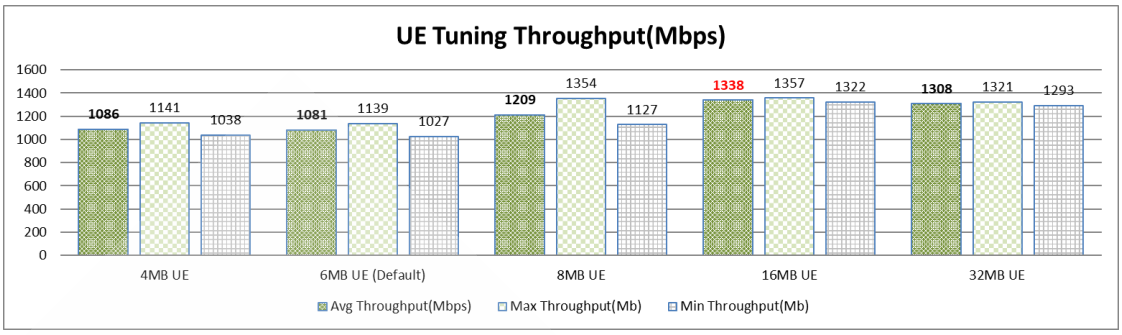
การทดสอบจะทดสอบปรับ Buffer ขนาด 4 MB, 6 MB, 8 MB, 16 MB และ 32 MB โดย Linux Mint 18 จะกำหนด TCP buffer sizes ค่า Default ดังนี้ receive socket memory (rmem = 6 MB) และ send socket memory (wmem = 4 MB) ผลการทดสอบปรับ Buffer ที่ UE ซึ่งจะปรับที่ receive socket memory เท่านั้น สามารถสรุปผลได้ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดสอบเมื่อปรับ Buffer ที่ UE

UE Tuning TCP_MEM														
UE Tuning	MA-GW		UE		Avg Throughput (Mbps)	Max Throughput (Mbps)	Min Throughput (Mbps)	Download Time (Sec)	Delay (µs)	Min Delay (µs)	Max Delay (µs)	Jitter (ns)	Min jitter (ns)	Max jitter (ns)
	rmem	wmem	rmem	wmem										
4MB UE	6MB	4MB	4MB	4MB	1086	1141	1038	36.20	9.82	1.324	393,755	192.68	0	155,034,920
6MB UE (Default)	6MB	4MB	6MB	4MB	1081	1139	1027	35.87	9.94	1.318	398,635	252.91	0	158,901,994
8MB UE	6MB	4MB	8MB	4MB	1209	1354	1127	33.51	9.02	1.347	394,383	120.64	1.87E-10	155,530,760
16MB UE	6MB	4MB	16MB	4MB	1338	1357	1322	32.54	8.27	1.426	202,294	5.17	1.11E-10	40,919,481
32MB UE	6MB	4MB	32MB	4MB	1308	1321	1293	33.24	8.45	1.312	84,376	3.61	0	7,117,945

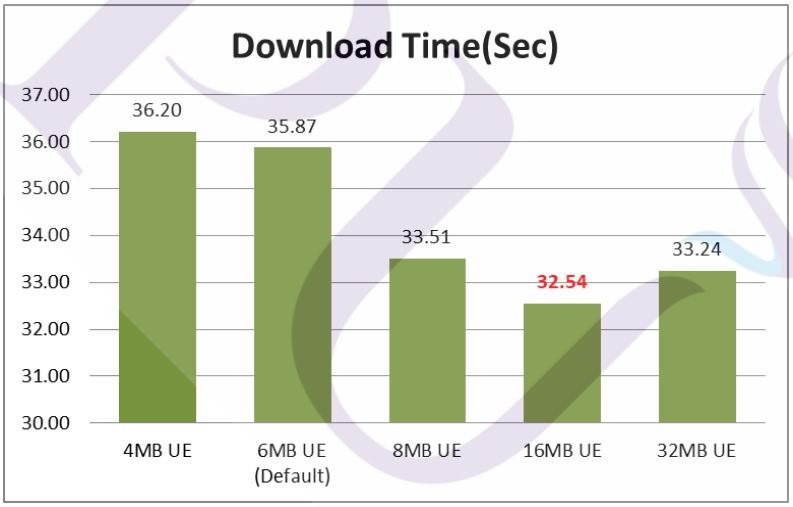
ค่าเริ่มต้นของระบบปฏิบัติการ
ผลการทดสอบหลักที่ดีที่สุด
ผลการทดสอบรองที่ดีที่สุด

ความเร็วในการรับส่งข้อมูล (Throughput) เมื่อปรับ Buffer ขนาด 16 MB สามารถเพิ่มประสิทธิภาพของความเร็วในการรับส่งข้อมูลได้มากที่สุด เฉลี่ยที่ 1,338 Mbps รองลงมาคือ 32 MB และ8 MB สามารถเพิ่มประสิทธิภาพได้ 1,308 และ 1,209 Mbps ตามลำดับ ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่า การปรับ Buffer 16 MB สามารถให้ ความเร็วในการรับส่งข้อมูลดีที่สุด แต่การปรับ Buffer เป็น 32MB ก็สามารถใช้งานได้เช่นกันเนื่องจาก ความเร็วนในการรับส่งข้อมูลไม่ได้ต่างกันอย่างมีนัยสำคัญ ดังภาพที่ 4.1



ภาพที่ 4.1 แสดงความเร็วในการรับส่งข้อมูล (Mbps) กรณีปรับขนาด Buffer ของ UE

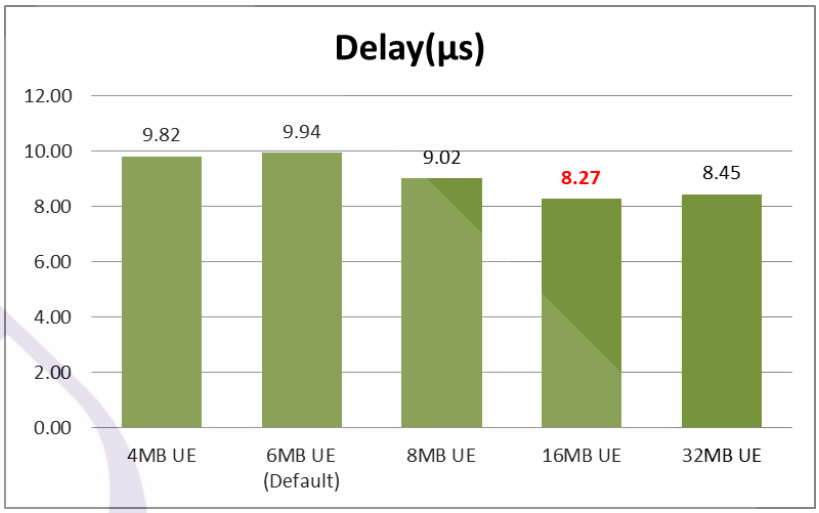
ระยะเวลาในการดาวน์โหลด (Download Time) เมื่อปรับ Buffer ขนาด 16 MB ประสิทธิภาพของการดาวน์โหลดได้เร็วที่สุด เฉลี่ยที่ 32.54 Sec รองลงมาคือ 32 MB และ8 MB ซึ่งมีระยะเวลาดาวน์โหลด 33.24 และ 33.51 Sec ตามลำดับ ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่าการปรับ Buffer 16 MB สามารถให้ผลการดาวน์โหลดที่เร็วที่สุด แต่การปรับ Buffer เป็น 32 MB และ 8 MB ก็สามารถใช้งานได้เช่นกันเนื่องจาก สามารถลดระยะเวลาในการดาวน์โหลดได้ใกล้เคียงกับการปรับ Buffer เป็น 16 MB เช่นกัน ดังภาพที่ 4.2



ภาพที่ 4.2 แสดงระยะเวลาในการดาวน์โหลด (Sec) กรณีปรับขนาด Buffer ของ UE

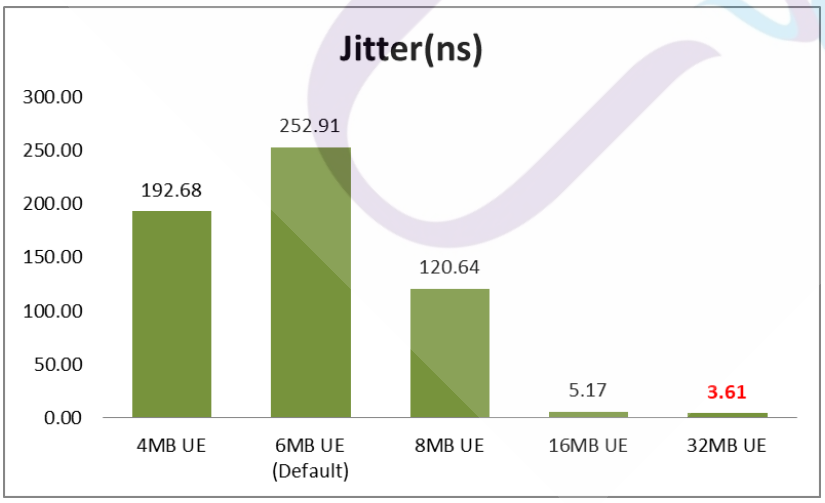
ค่าความล่าช้าทางเวลา (Delay) เมื่อปรับ Buffer ขนาด 32MB ค่าความล่าช้าทางเวลาน้อยที่สุด เฉลี่ยที่ 8.45 μs รองลงมาคือ 32MB และ 8MB สามารถทำให้ค่าความล่าช้าทางเวลาน้อยลงได้ 8.45 และ 9.02 μs ตามลำดับ ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่า การปรับ Buffer

16MB สามารถให้ผลการทำงานดีที่สุด แต่การปรับ Buffer เป็น 32MB ก็สามารถใช้งานได้เช่นกัน เนื่องจากค่าความล่าช้าทางเวลาไม่ได้ต่างกันอย่างมีนัยสำคัญ ดังภาพที่ 4.3



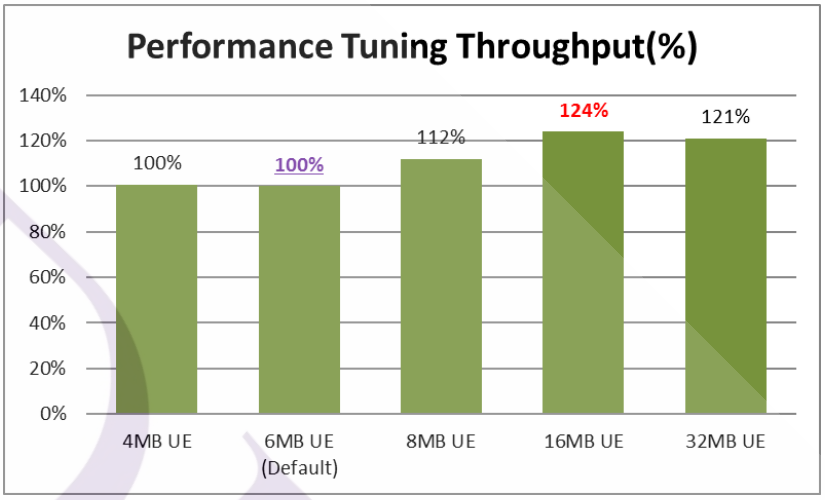
ภาพที่ 4.3 แสดงค่าความล่าช้าทางเวลา (µs) กรณีปรับขนาด Buffer ของ UE

ค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) เมื่อปรับ Buffer ขนาด 32MB ค่าความแปรปรวนของความล่าช้าทางเวลาน้อยที่สุด เฉลี่ยที่ 3.61 ns รองลงมาคือ 16MB สามารถทำให้ค่าความแปรปรวนของความล่าช้าทางเวลาน้อยลงได้ 5.17 ns แต่ผลการทดสอบออกมาเป็นหน่วย ns ซึ่งทำให้ผลที่ออกมาไม่แตกต่างกันอย่างมีนัยสำคัญ ดังภาพที่ 4.4

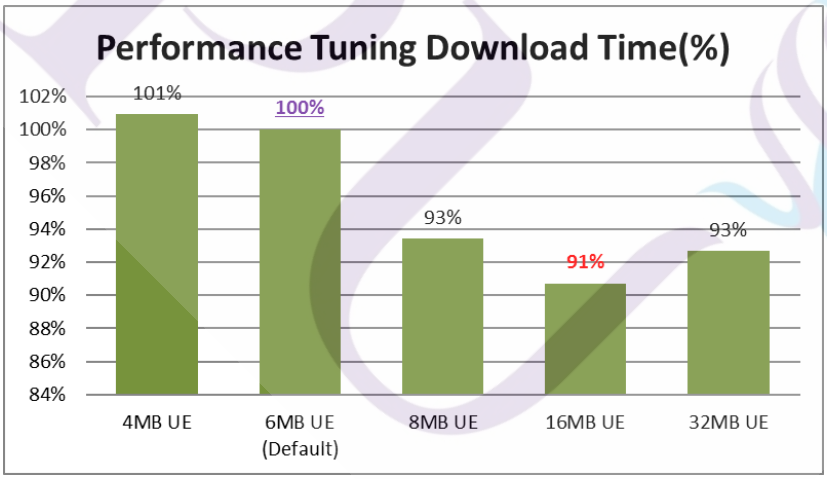


ภาพที่ 4.4 แสดงค่าความแปรปรวนของความล่าช้าทางเวลา (ns) กรณีปรับขนาด Buffer ของ UE

การทดสอบปรับ Buffer UE สามารถเพิ่มประสิทธิภาพการทำงานได้จริงโดยสามารถเพิ่มความเร็วในการรับส่งข้อมูลได้ถึง 24% และลดระยะเวลาในการดาวน์โหลดได้ถึง 9% จาก Buffer มาตรฐานของ Linux Mint18 และ Linux CentOS 7 [receive socket memory (rmem = 6MB) และ send socket memory (wmem = 4MB)] ดังภาพ 4.5 และ 4.6



ภาพที่ 4.5 แสดงประสิทธิภาพความเร็วในการรับส่งข้อมูล (%) กรณีปรับขนาด Buffer ของ UE



ภาพที่ 4.6 แสดงประสิทธิภาพระยะเวลาในการดาวน์โหลด (%) กรณีปรับขนาด Buffer ของ UE

4.2 วิเคราะห์ผลการทดสอบปรับ Buffer ที่ MA-GW

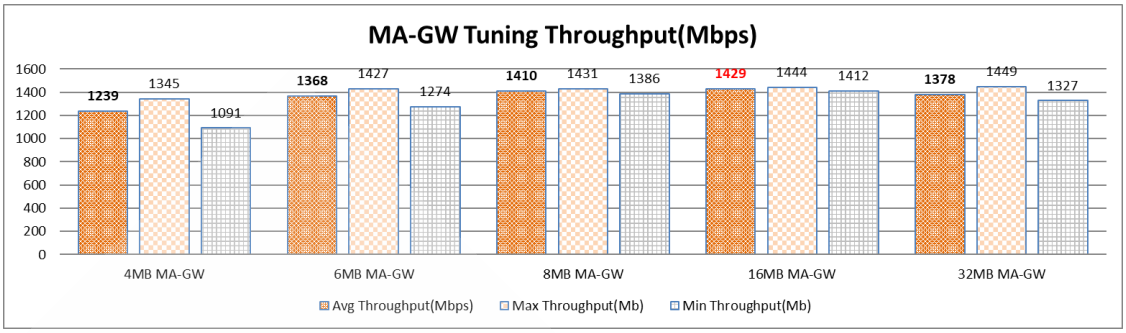
การทดสอบจะทดสอบปรับ Buffer ขนาด 4MB, 6MB, 8MB, 16MB, 32MB โดย Linux CentOS 7 จะกำหนด TCP buffer sizes ค่า Default ดังนี้ receive socket memory (rmem = 6MB) และ send socket memory (wmem = 4MB) ผลการทดสอบปรับ Buffer ที่ MA-GW ซึ่งจะปรับที่ receive socket memory และ send socket memory สามารถสรุปผลได้ดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดสอบเมื่อปรับ Buffer ที่ MA-GW

MA-GW Tuning TCP_MEM															
MA-GW Tuning	MA-GW		UE		Avg Throughput (Mbps)	Max Throughput (Mbps)	Min Throughput (Mbps)	Download Time (Sec)	Delay (µs)	Min Delay (µs)	Max Delay (µs)	Jitter (ns)	Min jitter (ns)	Max jitter (ns)
	rmem	wmem	rmem	wmem										
4MB MA-GW	4MB	4MB	16MB	4MB	1239	1345	1091	33.54	8.79	1.317	376,662	93.62	9.35E-11	141,867,273
6MB MA-GW	6MB	6MB	16MB	4MB	1368	1427	1274	30.74	8.045519	1.390	1	199.6613	4.74E-11	1
8MB MA-GW	8MB	8MB	16MB	4MB	1410	1431	1386	30.13	7.7	1.322	194,865	15.97	2.72E-11	37,969,370
16MB MA-GW	16MB	16MB	16MB	4MB	1429	1444	1412	29.41	7.71	1.391	1,194,586	226.1	3.06E-12	1,427,017,346
32MB MA-GW	32MB	32MB	16MB	4MB	1378	1449	1327	30.86	7.81	1.335	1,640,558	301.31	1.34E-11	2,691,405,418

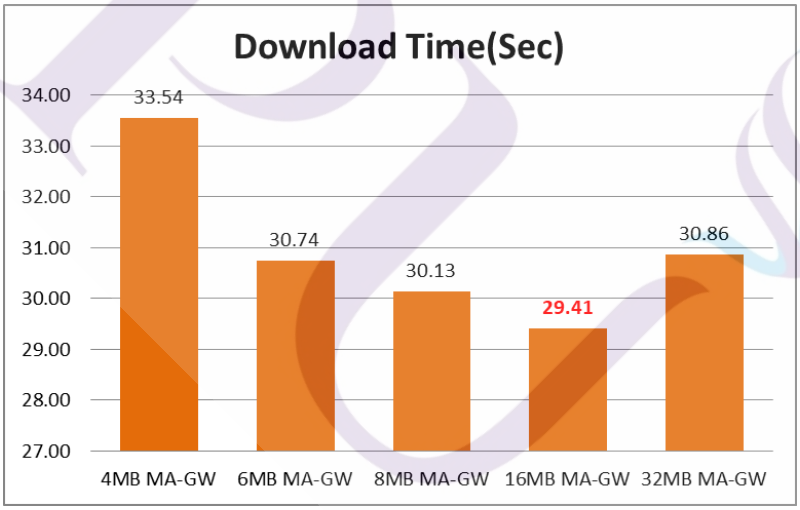
ค่าเริ่มต้นของระบบปฏิบัติการ
ผลการทดสอบหลักที่ดีที่สุด
ผลการทดสอบรองที่ดีที่สุด

ความเร็วในการรับส่งข้อมูล (Throughput) เมื่อปรับ Buffer ขนาด 16MB สามารถเพิ่มประสิทธิภาพของความเร็วในการรับส่งข้อมูลได้มากที่สุด เฉลี่ยที่ 1,429 Mbps รองลงมาคือ 8MB สามารถเพิ่มประสิทธิภาพได้ 1,410 ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่า การปรับ Buffer 16MB สามารถให้ความเร็วในการรับส่งข้อมูลดีที่สุด แต่การปรับ Buffer เป็น 8MB ก็สามารถใช้งานได้เช่นกันเนื่องจากความเร็วในการรับส่งข้อมูลไม่ได้ต่างกันอย่างมีนัยสำคัญ ดังภาพที่ 4.7



ภาพที่ 4.7 แสดงความเร็วในการรับส่งข้อมูล (Mbps) กรณีปรับขนาด Buffer ของ MA-GW

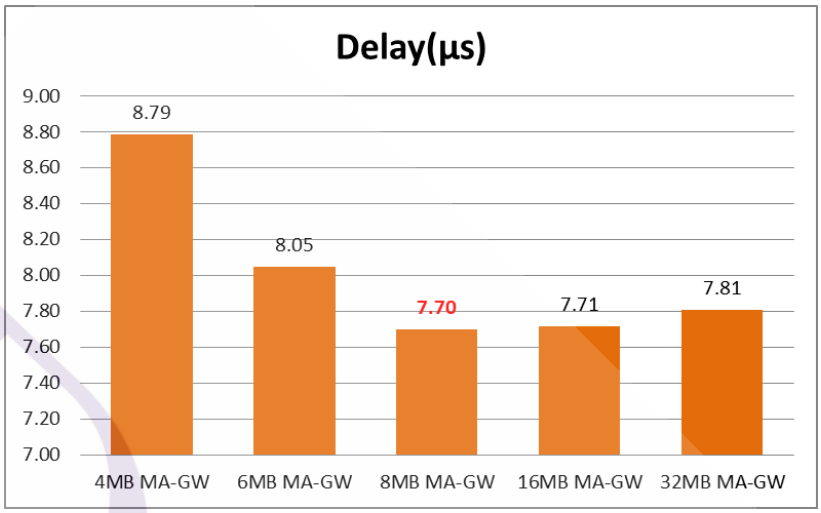
ระยะเวลาในการดาวน์โหลด (Download Time) เมื่อปรับ Buffer ขนาด 16MB ประสิทธิภาพของการดาวน์โหลดได้เร็วที่สุด เฉลี่ยที่ 29.41 Sec รองลงมาคือ 8MB และ32MB สามารถลดระยะเวลาในการดาวน์โหลดได้ 30.13 และ 30.86 Sec ตามลำดับ ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่า การปรับ Buffer 16MB สามารถให้ผลการดาวน์โหลดที่เร็วที่สุด แต่การปรับ Buffer เป็น 8MB และ 32MB ก็สามารถใช้งานได้เช่นกันเนื่องจาก สามารถลดระยะเวลาดาวน์โหลดได้ใกล้เคียงกับการปรับ Buffer เป็น 16MB เช่นกัน ดังภาพที่ 4.8



ภาพที่ 4.8 แสดงระยะเวลาในการดาวน์โหลด (Sec) กรณีปรับขนาด Buffer ของ MA-GW

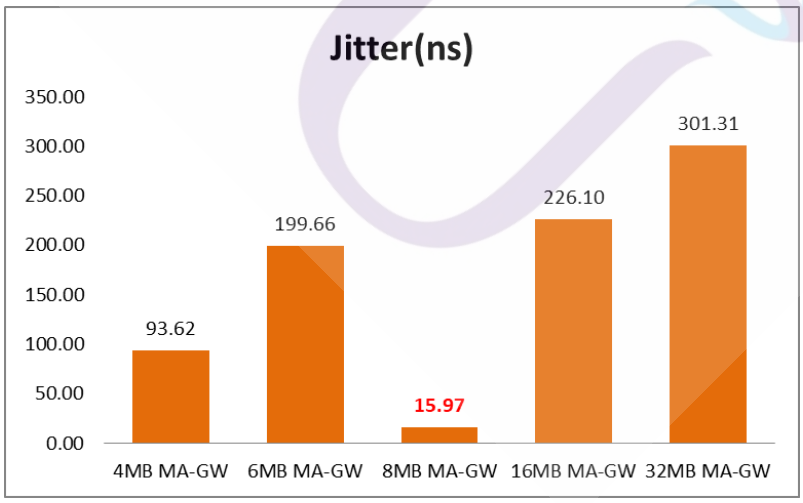
ค่าความล่าช้าทางเวลา (Delay) เมื่อปรับ Buffer ขนาด 8MB ค่าความล่าช้าทางเวลาน้อยที่สุด เฉลี่ยที่ 7.70 μs รองลงมาคือ 16MB และ 32MB สามารถทำให้ค่าความล่าช้าทางเวลาน้อยลงได้ 7.71 และ 7.81 μs ตามลำดับ ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่า การปรับ Buffer 8MB

สามารถให้ผลการทำงานดีที่สุด แต่การปรับ Buffer เป็น 16MB และ 32MB ก็สามารถใช้งานได้เช่นกันเนื่องจากค่าความล่าช้าทางเวลาไม่ได้ต่างกันอย่างมีนัยสำคัญ ดังภาพที่ 4.9



ภาพที่ 4.9 แสดงค่าความล่าช้าทางเวลา (µs) กรณีปรับขนาด Buffer ของ MA-GW

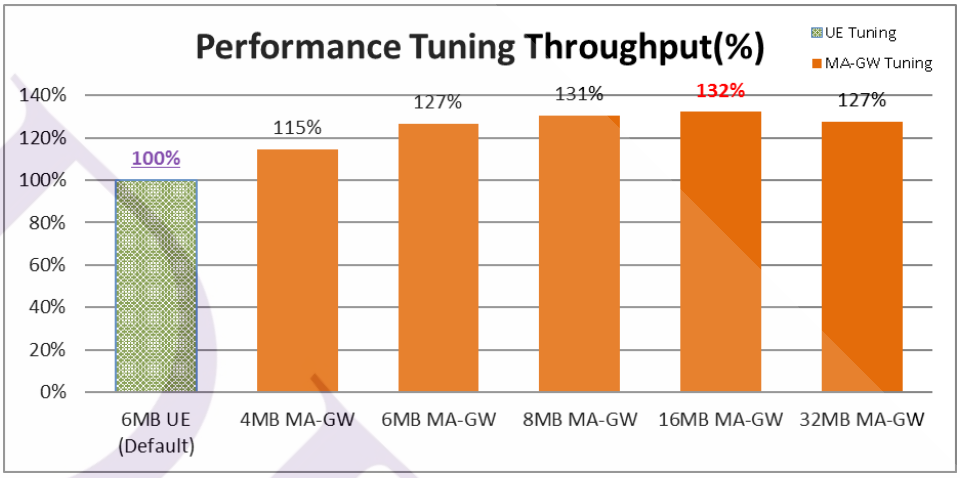
ค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) เมื่อปรับ Buffer ขนาด 8MB ค่าความแปรปรวนของความล่าช้าทางเวลาน้อยที่สุด เฉลี่ยที่ 15.97 ns ผลการทดสอบสามารถบ่งบอกได้ว่า การปรับ Buffer 8MB สามารถให้ผลการทำงานดีที่สุด แต่ผลการทดสอบออกมาเป็นหน่วย ns ซึ่งทำให้ผลที่ออกมาไม่แตกต่างกันอย่างมีนัยสำคัญ ดังภาพที่ 4.10



ภาพที่ 4.10 แสดงค่าความแปรปรวนของความล่าช้าทางเวลา (ns)

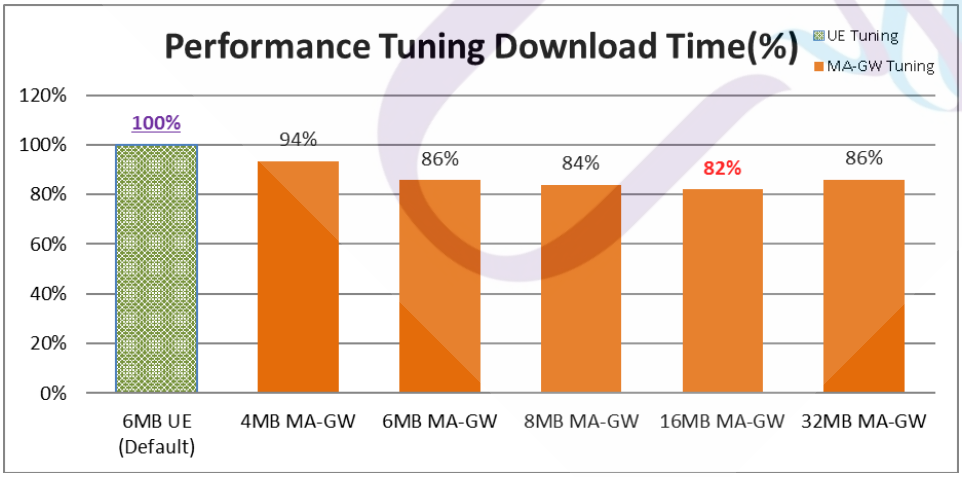
กรณีปรับขนาด Buffer ของ MA-GW

การทดสอบปรับ Buffer MA-GW สามารถเพิ่มประสิทธิภาพการทำงานได้จริงโดยสามารถเพิ่มความเร็วในการรับส่งข้อมูลได้ถึง 32% และระยะเวลาในการดาวน์โหลดได้ถึง 18% จาก Buffer มาตรฐานของ Linux Mint18 และ Linux CentOS 7 [receive socket memory (rmem = 6MB) และ send socket memory (wmem = 4MB)] ดังภาพ 4.11 และ 4.12



ภาพที่ 4.11 แสดงประสิทธิภาพความเร็วในการรับส่งข้อมูล (%)

กรณีปรับขนาด Buffer ของ MA-GW



ภาพที่ 4.12 แสดงประสิทธิภาพระยะเวลาในการดาวน์โหลด (%)

กรณีปรับขนาด Buffer ของ MA-GW

4.3 วิเคราะห์ผลการทดสอบปรับ Congestion Control

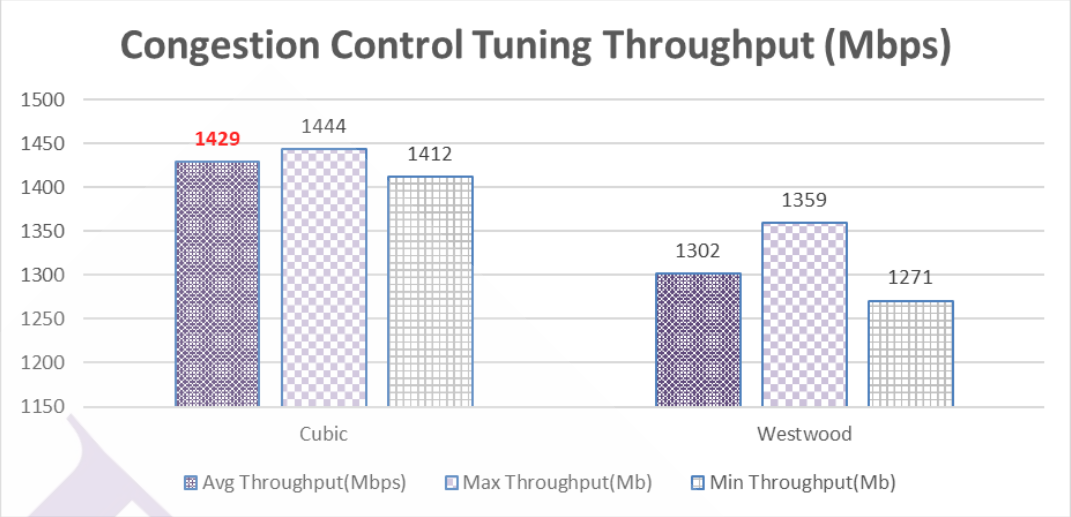
การทดสอบจะทดสอบปรับ Congestion Control ระหว่าง Cubic(Default) ออกแบบมาสำหรับ เพิ่มความเร็วบนเครื่อข่ายที่มี large bandwidth-delay product กับ westwood ออกแบบมาเพื่อเพิ่มความเร็วในการกู้คืนเมื่อเกิด Duplicate ACKs ของ large bandwidth-delay product เพื่อเปรียบเทียบว่า Congestion Control ประเภทไหนให้ประสิทธิภาพมากกว่ากัน โดยจะกำหนด TCP buffer sizes เป็นค่าที่ดีที่สุดจากการทดสอบ Buffer ดังนี้ UE (rmem = 16MB) / (wmem = 4MB) และ MA-GW (rmem = 16MB) / (wmem = 16MB) ผลการทดสอบปรับ Congestion Control ที่ UE และ MA-GW พร้อมกัน สามารถสรุปผลได้ดังตารางที่ 4.3

ตารางที่ 4.3 ผลการทดสอบเมื่อปรับ Congestion Control

Congestion Control Tuning	MA-GW		UE		Avg Throughput (Mbps)	Max Throughput (Mbps)	Min Throughput (Mbps)	Download Time (Sec)	Delay (µs)	Min Delay (µs)	Max Delay (µs)	Jitter (ns)	Min jitter (ns)	Max jitter (ns)
	rmem	wmem	rmem	wmem										
Cubic (Default)	16MB	16MB	16MB	4MB	1429	1444	1412	29.41	7.71	1.391	1,194,586	226.1	3.06E-12	1,427,017,346
Westwood	16MB	16MB	16MB	4MB	1302	1359	1271	33.02	8.4	1.399	248,393	26.52	0.00E+00	61,694,674

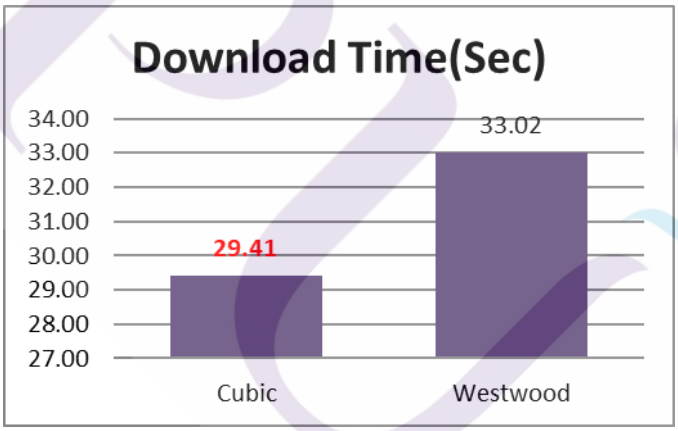
ค่าเริ่มต้นของระบบปฏิบัติการ
ผลการทดสอบหลักที่ดีที่สุด
ผลการทดสอบรองที่ดีที่สุด

ความเร็วในการรับส่งข้อมูล (Throughput) การใช้งาน Cubic สามารถแสดงประสิทธิภาพของ ความเร็วในการรับส่งข้อมูลเฉลี่ยที่ 1,429 Mbps ซึ่งได้มากกว่า Westwood เฉลี่ยที่ 1,302 Mbps ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่า Cubic สามารถให้ความเร็วในการรับส่งข้อมูลดีที่สุด ดังภาพที่ 4.13



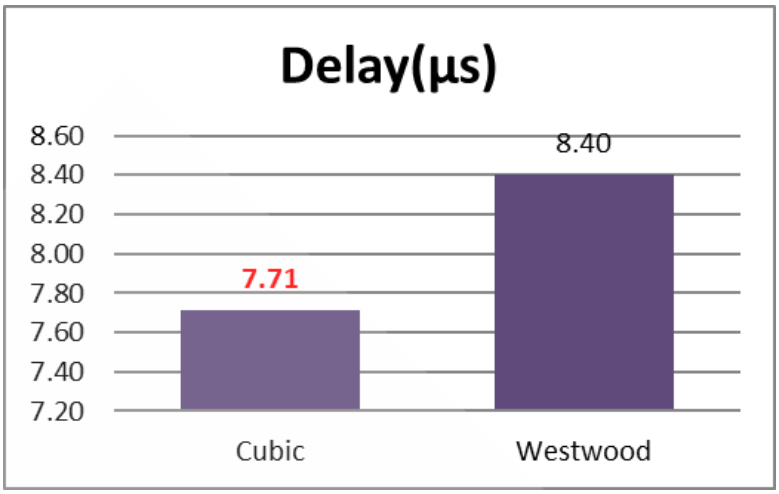
ภาพที่ 4.13 แสดงความเร็วในการรับส่งข้อมูล (Mbps) กรณีปรับ Congestion Control

ระยะเวลาในการดาวน์โหลด (Download Time) การใช้งาน Cubic ประสิทธิภาพของการดาวน์โหลดได้เร็วที่สุด เฉลี่ยที่ 29.41 Sec ซึ่งทำได้ดีกว่า Westwood เฉลี่ยที่ 33.02 Sec ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่า Cubic สามารถให้ผลการดาวน์โหลดได้เร็วที่สุด ดังภาพที่ 4.14



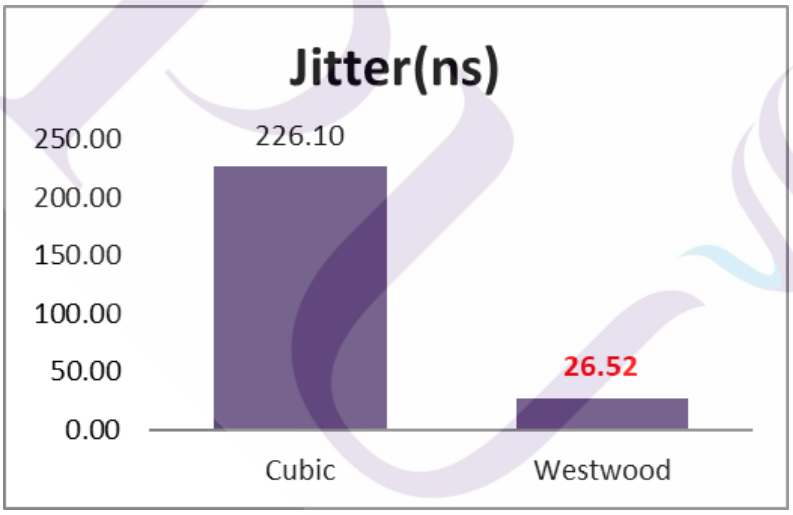
ภาพที่ 4.14 แสดงระยะเวลาในการดาวน์โหลด (Sec) กรณีปรับ Congestion Control

ค่าความล่าช้าทางเวลา (Delay) การใช้งาน Cubic ค่าความล่าช้าทางเวลาน้อยที่สุด เฉลี่ยที่ 7.71 μs ซึ่งทำได้ดีกว่า Westwood เฉลี่ยที่ 8.40 μs ซึ่งผลการทดสอบสามารถบ่งบอกได้ว่า Cubic สามารถให้ผลการทำงานดี ดังภาพที่ 4.15



ภาพที่ 4.15 แสดงค่าความล่าช้าทางเวลา (µs) กรณีปรับ Congestion Control

ค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) การใช้งาน Westwood ค่าความแปรปรวนของความล่าช้าทางเวลาน้อยที่สุด เฉลี่ยที่ 26.52 ns ซึ่งทำได้ดีกว่า Cubic เฉลี่ยที่ 226.10 ns ผลการทดสอบสามารถบ่งบอกได้ว่า Westwood สามารถให้ผลการทำงานดีที่สุด ดังภาพที่ 4.16

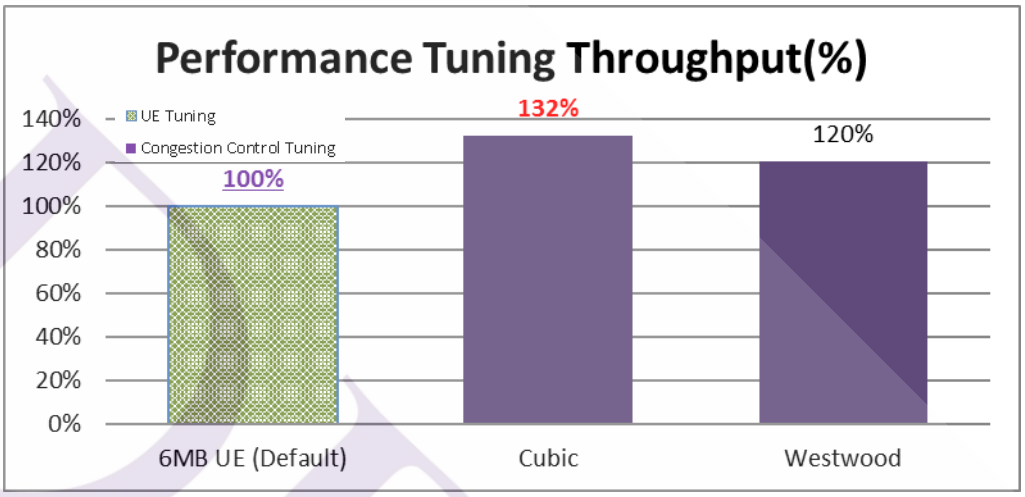


ภาพที่ 4.16 แสดงค่าความแปรปรวนของความล่าช้าทางเวลา (ns)

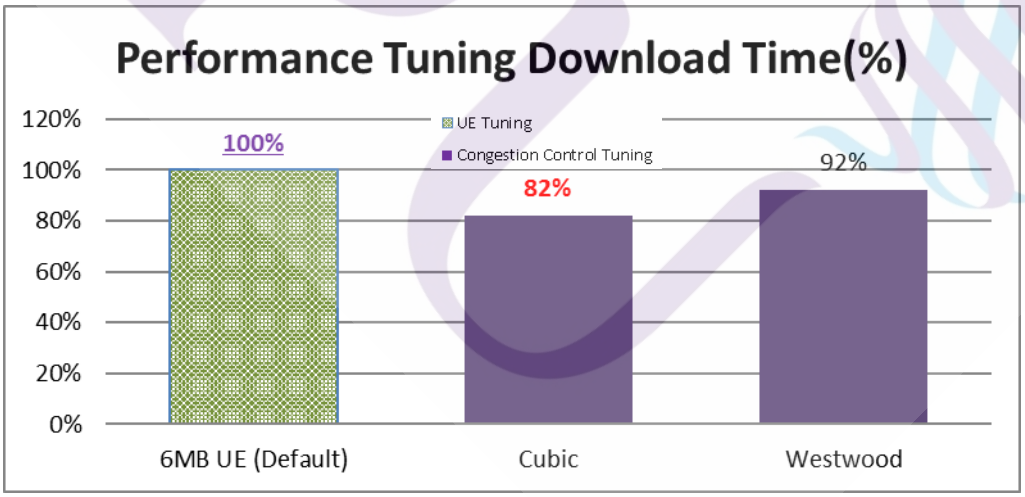
กรณีปรับ Congestion Control

การทดสอบปรับ Congestion Control สามารถแสดงประสิทธิภาพการทำงาน โดยที่ Westwood จะให้ประสิทธิภาพด้านค่าความแปรปรวนของความล่าช้าทางเวลาได้ดีกว่าโดยสามารถ

เพิ่มความเร็วในการรับส่งข้อมูลได้ 20% และระยะเวลาในการดาวน์โหลดได้ถึง 8% แต่ Cubic สามารถให้ประสิทธิภาพด้าน ความเร็วในการรับส่งข้อมูล, ระยะเวลาในการดาวน์โหลด, ค่าความล่าช้าทางเวลา ได้ดีกว่า โดยสามารถเพิ่มความเร็วในการรับส่งข้อมูลได้ถึง 32% และลดระยะเวลาในการดาวน์โหลดได้ถึง 18% จึงสรุปได้ว่าควรเลือกใช้ Congestion Control Cubic จึงจะสามารถใช้งาน MPTCP ได้ประสิทธิภาพได้ดีที่สุด ดังภาพ 4.17 และ 4.18



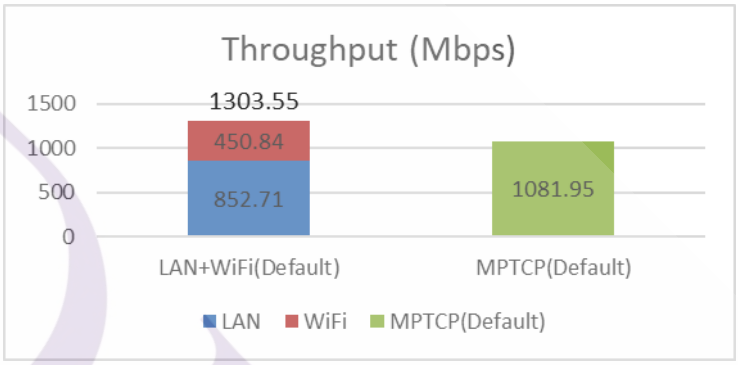
ภาพที่ 4.17 แสดงประสิทธิภาพความเร็วในการรับส่งข้อมูล (%) กรณีปรับ Congestion Control



ภาพที่ 4.18 แสดงประสิทธิภาพระยะเวลาในการดาวน์โหลด (%) กรณีปรับ Congestion Control

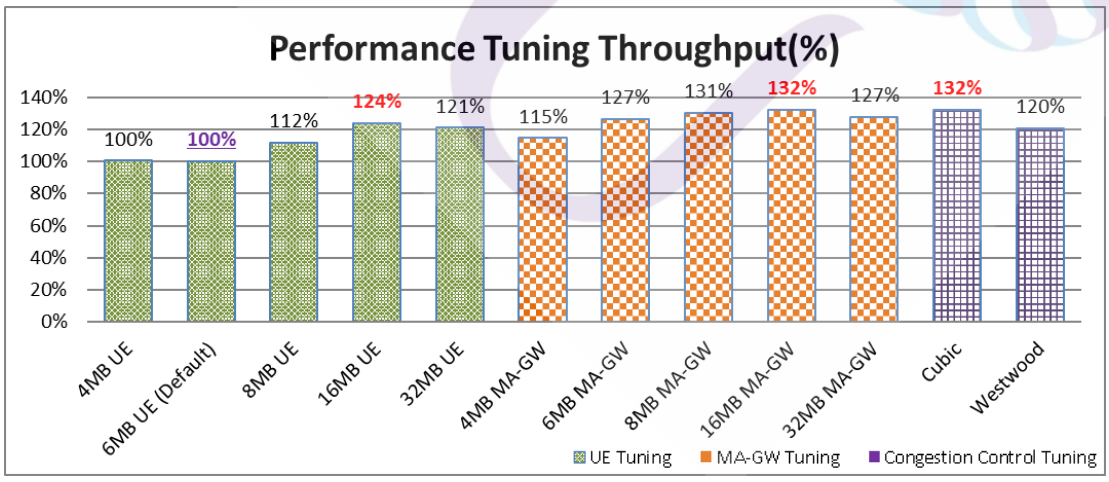
4.4 สรุปผลการวัดประสิทธิภาพการใช้งาน MPTCP

ประสิทธิภาพการทำงานกรณีทำงานแบบ TCP และไม่ปรับ Buffer ของ LAN จะได้ Throughput 852.71 Mbps และWiFi จะได้ Throughput 450.84 Mbps เพราะฉะนั้น LAN และ WiFi รวมจะได้ Throughput 1,302.55 Mbps เมื่อเปรียบเทียบกับ MPTCP กรณีไม่ปรับ Buffer จะได้ Throughput 1,081.95 Mbps โดยสามารถสรุปได้ว่า Environment ที่ได้ทดสอบ สามารถให้ Throughput ได้ประมาณ 83% จาก Througput ทั้งหมด ดังภาพที่ 4.19

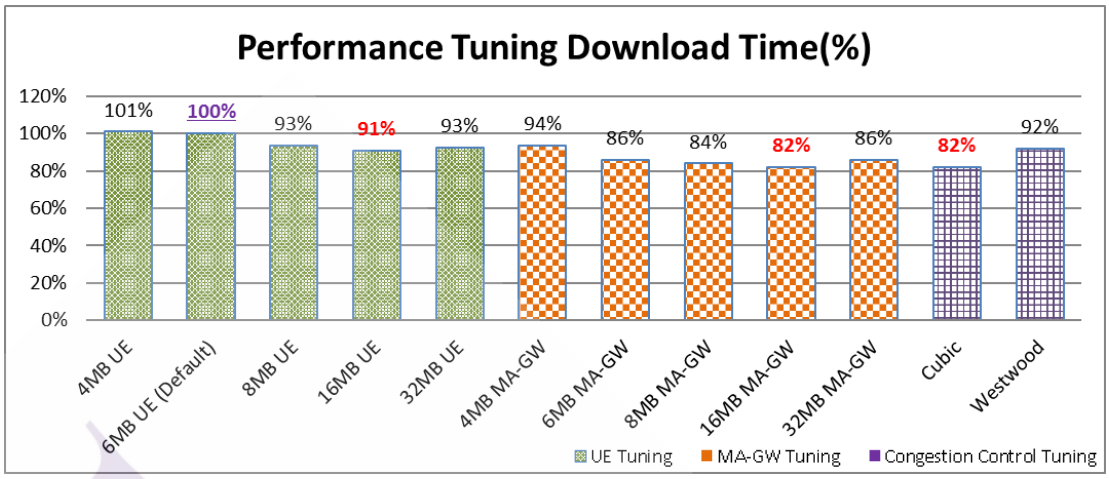


ภาพที่ 4.19 แสดง Throughput ของ LAN และ WiFi กรณีทำงานแบบ TCP เปรียบเทียบกับ MPTCP

จากการจำลองระบบบนอุปกรณ์จริงตาม Environment ที่ได้ทดสอบ ทั้งด้าน Buffer Tuning และ Congestion Control Tuning สามารถสรุปผลการทดสอบได้ดังนี้



ภาพที่ 4.20 แสดงความเร็วในการรับส่งข้อมูล (%) จากทุกการทดสอบ



ภาพที่ 4.21 แสดงระยะเวลาในการดาวน์โหลด (%) จากทุกการทดสอบ

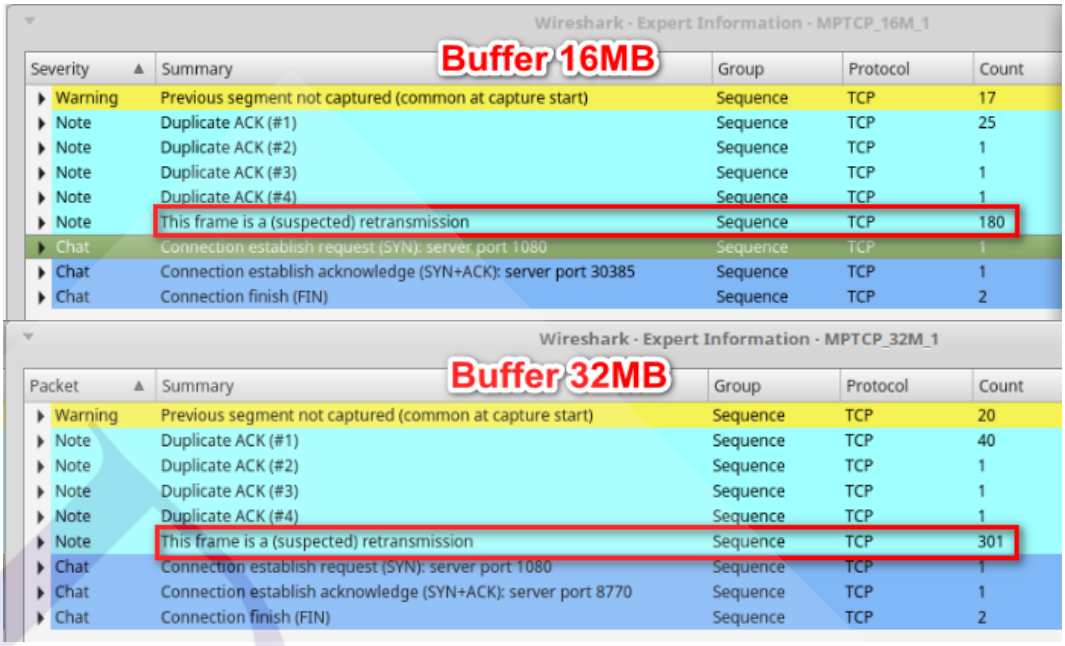
จากผลในตารางที่ 4.1 ถึง 4.3 สามารถวิเคราะห์ผลได้ดังนี้

Buffer Tuning เมื่อทำการเพิ่มขนาด Buffer เป็น 6MB, 8MB, 16MB, 32MB สามารถทำให้ความเร็วในการรับส่งข้อมูล (Throughput) มากขึ้น ส่งผลให้สามารถลดระยะเวลาในการดาวน์โหลด (Download Time) และ ค่าความล่าช้าทางเวลา (Delay)น้อยลง จากผลลัพธ์นี้ สามารถสรุปได้กว่าการปรับ Buffer ทำให้เพิ่มประสิทธิภาพในการทำงานได้จริง ส่วนค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) มีอาการแปรปวนเพราะมีการใช้งาน WiFi บนอุปกรณ์จริงในการทดสอบ ทำให้ยากต่อการควบคุมคลื่นรบกวน แต่โดยภาพรวม Jitter ที่ได้มามีหน่วยเป็น ns ซึ่งเป็นความเร็วที่ยอมรับได้ในปัจจุบัน ข้อสังเกต เมื่อทดสอบปรับ Buffer 32MB ประสิทธิภาพการทำงานที่ได้ออกมาจะแย่กว่าการปรับ Buffer 16MB เล็กน้อย ซึ่งทางผู้วิจัยได้ตรวจสอบพบว่า Buffer 32MB จะมี Packet Retransmission มากกว่า Buffer 16MB ดังภาพที่ 4.21 สิ่งที่งานวิจัยนี้คาดว่าเป็นเหตุที่ Buffer 32MB ทำงานได้แย่กว่า Buffer 16MB คือ

4.4.1 Buffer มีใหญ่เกิน ไม่สอดคล้องกับความเร็วในการประมวลผล

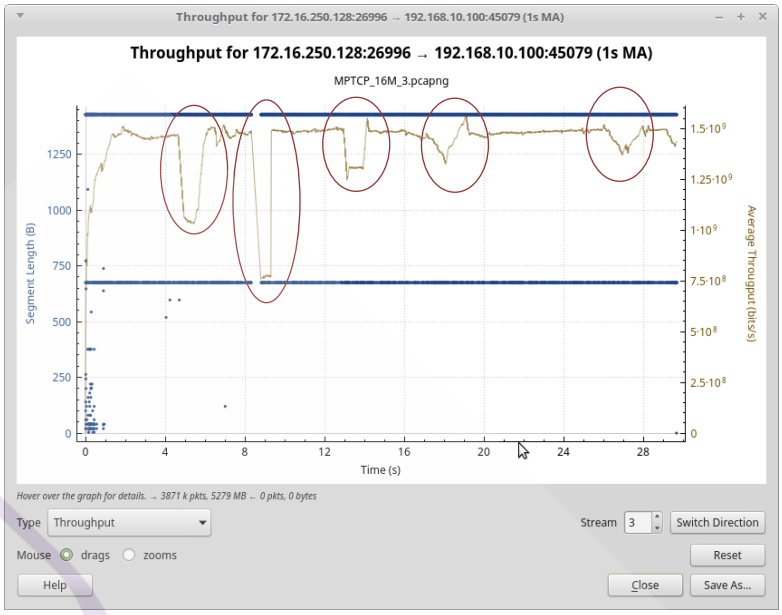
4.4.2 Buffer มีใหญ่เกิน ไม่สอดคล้องกับขนาด Windows size

จึงสามารถสรุปได้ว่า การปรับ Buffer สามารถเพิ่มประสิทธิภาพการทำงานได้จริง (ผลการทดสอบของงานวิจัยนี้สามารถเพิ่มความเร็วในการรับส่งข้อมูลสูงสุดถึง 32% ดังภาพที่ 4.19 และ ลดระยะเวลาในการดาวน์โหลดได้ถึง 18% ดังภาพที่ 4.20) แต่การปรับ Buffer มากเกินเกิด Packet Retransmission ในระบบเยอะขึ้นดังภาพที่ 4.21 และประสิทธิภาพการทำงานจะลดลง

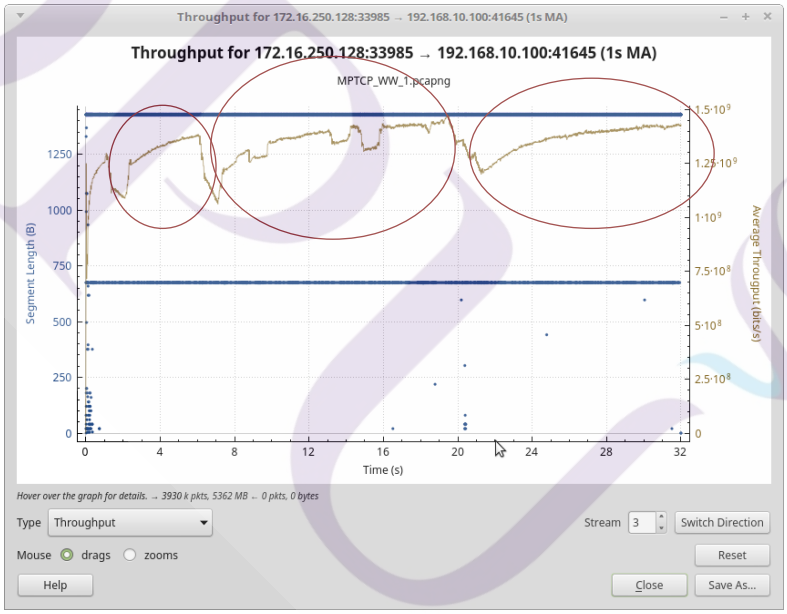


ภาพที่ 4.22 แสดง Buffer 16, 32 MB Retransmission บนระบบจริง

Congestion Control Tuning การใช้งาน Cubic สามารถให้ประสิทธิภาพด้าน ความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) ได้ดีกว่าเนื่องจาก Cubic มีการใช้ Algorithm Binary Search เข้ามาช่วยตั้งแต่กระบวนการ Slow Start, Congestion Avoidance, Fast Retransmit และ Fast Recovery ทำให้ Cubic สามารถเข้าถึงความเร็วในการรับส่งข้อมูลสูงสุดได้เร็วกว่าดังภาพที่ 4.23 Westwood ปรับปรุงในส่วน Fast Retransmit และ Fast Recovry เพื่อให้ความเร็วในการรับส่งข้อมูลใกล้เคียง Transit Capacity มากที่สุด เพื่อใช้งานกับเครือข่ายไร้สายโดยเฉพาะ จะต้องในเวลานานกว่าเพื่อเข้าถึงความเร็วในการรับส่งข้อมูลสูงสุดดังภาพที่ 4.24 ด้วยเหตุผลนี้จึงทำให้ Cubic สามารถทำงานได้เต็มประสิทธิภาพมากกว่า (ผลการทดสอบ Congestion Control ของงานวิจัยนี้ Cubicสามารถเพิ่มความเร็วในการรับส่งข้อมูลจาก Westwood ถึง 12% ดังภาพที่ 4.19 และ ลดระยะเวลาในการดาวน์โหลดจาก Westwood ถึง 10% ดังภาพที่ 4.20)



ภาพที่ 4.23 แสดงประสิทธิภาพการ Retransmit และ Fast Recovery ของ Cubic



ภาพที่ 4.24 แสดงประสิทธิภาพการ Retransmit และ Fast Recovery ของ Westwood

บทที่ 5
บทสรุป และข้อเสนอแนะ

ในบทนี้จะสรุปผลการดำเนินงาน ข้อจำกัดของงานวิจัย ข้อเสนอแนะและแนวทางการพัฒนาในอนาคต

5.1 สรุปผลการวิจัย

งานวิจัยได้จำลองการใช้งาน MPTCP ขึ้นบนอุปกรณ์จริง โดยใช้ระบบปฏิบัติการ Linux ในการทดสอบ สามารถใช้งาน MPTCP รวม Bandwidth ระหว่าง LAN และ WiFi ได้ตามมาตรฐาน rfc6182[1] ซึ่งสถานการณ์จริง Server ส่วนใหญ่ของผู้ให้บริการต่างๆ ไม่รองรับ MPTCP ระบบที่ทดสอบจึงมีการติดตั้ง MPTCP Proxy เพื่อเป็นตัวแทนในการสื่อสารระหว่าง MPTCP และ TCP โดยงานวิจัยนี้ได้ใช้ go-shadowsocks2 ในการทำ MPTCP Proxy สามารถทำงานได้ตาม Draft MPTCP proxy mechanisms[2] ของ IETF (Internet Engineering Task Force) ในการจำลองระบบนี้ทางผู้วิจัยได้พบปัญหา Driver WiFi ของอุปกรณ์ UE ไม่สามารถทำงานด้าน Upload ได้เต็มประสิทธิภาพ แต่ไม่มีผลต่อการทดสอบ เพราะ UE ทดสอบเพียงการ Download เท่านั้น

งานวิจัยจะเก็บผลการทดลองด้วยโปรแกรม Wireshark การทดลองจะดาวน์โหลดไฟล์ขนาด 5.4 GB โดยสามารถสรุปผลการทดลองได้ดังนี้

5.1.1 การปรับ Buffer สามารถทำให้เพิ่มประสิทธิภาพการทำงานได้จริง โดยการปรับ Buffer ที่ 16MB จะให้ประสิทธิภาพการทำงานดีที่สุด เมื่อปรับ Buffer เป็น 32MB ประสิทธิภาพในการทำงานไม่เพิ่มขึ้นเนื่องจากมี Packet Retransmission มากกว่า การปรับ Buffer ควรปรับที่ UE และ MA-GW จะเพิ่มประสิทธิภาพได้ดีที่สุด ความเร็วในการรับส่งข้อมูลก่อนปรับ Buffer ที่ 1,081 Mbps เป็น 1,429 Mbps ดีขึ้น 32% เทียบกับการใช้ค่า default ของ Linux Mint เวอร์ชั่น 18 (UE) และ CentOS เวอร์ชั่น 7 (MA-GW), ระยะเวลาในการดาวน์โหลดก่อนปรับ Buffer ใช้เวลา 35.87

---

[1] A. Ford, C. Raiciu, M. Handley, S. Barre, J.Iyengar. (2011). "Architectural Guidelines for Multipath TCP Development". Retrieved September 10, 2017, from https://www.rfc-editor.org/rfc/rfc6182.txt

[2] X.Wei, C.Xiong, E. Lopez, Fortinet. (2015). MPTCP proxy mechanisms. Retrieved September 10, 2017, from https://datatracker.ietf.org/doc/draft-wei-mptcp-proxy-mechanism/

วินาที เป็น 29.41 วินาที ใช้เวลาเร็วขึ้น 18%, ค่าความล่าช้าทางเวลา และค่าความแปรปรวนของความล่าช้าทางเวลาเนื่องจากผลลัพธ์ที่ได้ออกมา ค่าความล่าช้าทางเวลา < 4 μs และค่าความแปรปรวนของความล่าช้าทางเวลาหน่วยเป็น Nano Sec จึงแตกต่างกันอย่างไม่มีนัยสำคัญ

5.1.2 การปรับ Congestion Control ผลการทดสอบ Cubic ใช้ Algorithm Binary Search เข้ามาช่วยตั้งแต่กระบวนการ Slow Start, Congestion Avoidance, Fast Retransmit และ Fast Recovery ทำให้ Cubic สามารถเพิ่มประสิทธิภาพการทำงานได้ดีกว่า Westwood ที่ปรับปรุงแค่ในส่วน Fast Retransmit และ Fast Recovry เพื่อให้ความเร็วในการรับส่งข้อมูลใกล้เคียง Transit Capacity มากที่สุด ความเร็วในการรับส่งข้อมูล 12%, ระยะเวลาในการดาวน์โหลด 10% และค่าความล่าช้าทางเวลา และ ค่าความแปรปรวนของความล่าช้าทางเวลาเนื่องจากผลลัพธ์ที่ได้ออกมา ค่าความล่าช้าทางเวลา < 4 μs และ ค่าความแปรปรวนของความล่าช้าทางเวลาหน่วยเป็น Nano Sec จึงแตกต่างกันอย่างไม่มีนัยสำคัญ

MPTCP สามารถใช้ประโยชน์ได้ดีกับอุปกรณ์ที่มีอินเทอร์เฟสเชื่อมต่อด้าน Bandwidth, Delay, Jitter ที่แตกต่างกัน เช่น Smartphone มีอินเทอร์เฟส LTE และ WiFi หรือใช้งานกับระบบการส่งข้อมูลในระยะทางไกล เช่น ระบบสำรองข้อมูลต่างๆ หรือ DR Site, ใช้งานกับพวกอุปกรณ์ IOT (อาจจะต้องดูเรื่องการใช้พลังงานเพิ่มเติม) หรือสำหรับให้บริการ Internet LTE + WiFi ให้กับ Smartphone ฯลฯ บริการเหล่านี้เหมาะกับการใช้งาน MPTCP มากที่สุด

5.2 ข้อจำกัดของงานวิจัย และแนวทางแก้ไข

5.2.1 ข้อจำกัดเรื่องอุปกรณ์ ที่นำมาทดสอบ ดังนี้

- จำนวนเครื่อง UE มีแค่เครื่องเดียว ไม่สามารถทดสอบกรณีแย่ง Traffic ได้แนวทางแก้ไข ต้องมีทุนวิจัยเพิ่ม หรือ Simulator แทน

- ปัญหาเรื่อง Driver WiFi ของ Laptop บน OS Linux ทำให้การ Upload ของอุปกรณ์มีข้อจำกัดอยู่ที่ 6 Mbps แต่ Download สามารถทำ 867 Mbps เต็มประสิทธิภาพ แนวทางแก้ไข ต้องหาอุปรณ์ WiFi AC ที่สามารถรองรับการทงานของ Linux ได้ 100%

- ไม่สามารถควบคุมเรื่องสัญญาณรบกวนของ WiFi ได้ เนื่องจากบริเวณที่ทดสอบเป็นคอนโด ซึ่งมีการแย่งช่องสัญญาณเยอะทำให้การทดสอบบางครั้งไม่เสถียรต้องทดสอบหลายครั้ง แนวทางการแก้ไข เลือกช่วงเวลาทดสอบที่คนส่วนใหญ่ปิดอุปกรณ์กระจาย WiFi หรือ เปลี่ยนสถานที่ทดสอบ

5.2.2 ข้อจำกัดของ MPTCP ไม่รองรับบน Windows ต้องรอ Version Windows ที่รองรับ

5.2.3 การใช้งาน MPTCP อุปกรณ์ต้นทาง และอุปกรณ์ปลายทางต้องติดตั้ง MPTCP ไม่เช่นนั้นอุปกรณ์จะทำงานผ่าน TCP แทน ต้องใช้ MA-GW เพื่อช่วยสื่อสารระหว่าง โพรโทคอล TCP และ MPTCP

5.2.4 การใช้งาน MA-GW เส้นทางฝั่ง TCP จำเป็นต้องมี Bandwidth มากกว่าฝั่ง MPTCP ไม่เช่นนั้นจะกลายเป็นคอขวด (Bottleneck) ของระบบ

5.2.5 ข้อจำกัดของ Coupled Congestion Control ของ kernel 4.14.24.mptcp ไม่สามารถใช้งานได้ แนวทางแก้ไขปัญหา ต้องรอ MPTCP Version ที่แก้ไขปัญหา

5.3 ข้อเสนอแนะและแนวทางการพัฒนาในอนาคต

5.3.1 ทดสอบการเพิ่มจำนวน UE ในระบบเพื่อหาประสิทธิภาพกรณี ที่มีจำนวน UE เยอะ

5.3.2 ทดสอบประสิทธิภาพการทำงานของ Coupled Congestion Control กับ Decoupled Congestion Control

5.3.3 ทดสอบประสิทธิภาพ MPTCP เปรียบเทียบระหว่าง มี MPTCP Proxy กับไม่มี MPTCP Proxy ให้ประสิทธิภาพต่างกันหรือไม่

5.3.4 ทดสอบประสิทธิภาพของ MPTCP ระหว่าง Local-Server Mode กับ Redirect Mode

5.3.5 ทดสอบหาข้อสรุปว่าเหตุใดเมื่อ Buffer size มากขึ้น จึงทำให้ Throughput น้อยลง

บรรณานุกรม

บรรณานุกรม

ภาษาไทย

จตุชัย แพงจันทร์, อนุโชต วุฒิพรพงษ์, และ อรรณพ ขันธิกุล (บรรณาธิการ). (2551). เจาะระบบ Network 2nd Edition. นนทบุรี : ไอดีซีฯ.

ผศ.ดร.มัชฌิกา อ่องแตง. (2017) CT518: Advanced Internet Engineering Class 7: Newer TCP Variants.

ผศ.ดร.มัชฌิกา อ่องแตง. (2017) CT518: Advanced Internet Engineering Class 6: Newer TCP Variants.

พิศาล พิทยาธุรวิวัฒน์. (2555). ติดตั้งระบบเครือข่าย Linux Server ภาคปฏิบัติ. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

Acer Inc. (2561). Aspire VX 15. สืบค้นเมื่อ 27 ตุลาคม 2561, จาก https://www.acer.com/ac/th/TH/content/series/aspirevx15

adslthailand.com. (2560) เทคโนโลยี LTE-WiFi Aggregation. สืบค้นเมื่อ 25 กรกฎาคม 2560, จาก http://www.adslthailand.com/post/introduction-to-multipath-tcp-lte-wifi-aggregation

Mindphp. (2560). buffer คืออะไร. สืบค้นเมื่อ 27 มีนาคม 2018, จาก http://www.mindphp.com/คู่มือ/73-คืออะไร/2061-buffer-คืออะไร.html

Ripmilla. (2553). มารู้จักกับตัวดักจับข้อมูลหรือ Sniffer กันเถอะ. สืบค้นเมื่อ 27 มีนาคม 2018, จาก https://www.freeware.in.th/windows/1756

Saixiii. (2560). Wireshark-sniffer. สืบค้นเมื่อ 27 มีนาคม 2018, จาก https://saixiii.com/wireshark-sniffer/

Ubiquiti Networks, Inc. (2561). UniFi AP AC LITE. สืบค้นเมื่อ 27 ตุลาคม 2561, จาก: https://www.ubnt.com/unifi/unifi-ap-ac-lite/

ภาษาต่างประเทศ

A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, “Architectural guidelines for multipath TCP development", RFC6182 2011. http://multipath-tcp.org/data/MultipathTCP-netsys.pptx

Ashkan Nikravesh, Yihua Guo, Feng Qian, Z. Morley Mao and Subhabrata Sen. An In-depth Understanding of Multipath TCP on Mobile Devices: Measurement and System Design. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, pages 189-201. ACM, 2016.

CLAUDIO CASETTI, MARIO GERLA, MARIO GERLA, M.Y. SANADIDI and REN WANG. TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks. in: Kluwer Academic Publishers. Manufactured in The Netherlands (2002).

Wei Bao, Vincent W.S. Wong, and Victor C.M. Leung. A Model for Steady State Throughput of TCP CUBIC. in: IEEE Globecom 2010.

Yung-Chih Chen, Yeon-sup Lim, Richard J. Gibbens, Erich M. Nahum, Ramin Khalili and Don Towsley. A Measurement-based Study of MultiPath TCP Performance over Wireless Networks. In Proceedings of the 2013 conference on Internet measurement conference, pages 455-468. ACM, 2013.

Zhixiong Niu, Zhi Wang, Hong Xu, Chuan Wu and Francis C.M. Lau. More is Better? Measurement of MPTCP based Cellular Bandwidth Aggregation in the Wild. In 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). IEEE, 2016

A. Ford, C. Raiciu, M. Handley, S. Barre, J.Iyengar. (2011). "Architectural Guidelines for Multipath TCP Development". Retrieved September 10, 2017, from https://www.rfceditor.org/rfc/rfc6182.txt

Laura Chappell. (2013). Wireshark Tip 22: Find Delays with TCP "Calculate Conversation Timestamps". Retrieved September 10, 2017, from https://www.youtube.com/watch?v=QqKAnZnHss0

Margaret Rouse. (2005). SOCKS. Retrieved September 30, 2018, from http://whatis.techtarget.com/definition/socks

Olivier Bonaventure. (2013). Decoupling TCP from IP with Multipath TCP. Retrieved October 25, 2017, from http://multipath-tcp.org/data/MultipathTCP-netsys.pptx

shadowsocks.org. (2018). shadowsocks. Retrieved October 19, 2018, from https://shadowsocks.org/en/spec/Implementations.html

sparkst@cs.unc.edu. (1969). How To: Network/TCP/UDP Tuning. Retrieved October 19, 2017, from https://wwwx.cs.unc.edu/~sparkst/howto/network_tuning.php

WIKIMEDIA. (2017). Transmission of "Hi, Hello world" using MPTCP. Retrieved July 28, 2017, from https://upload.wikimedia.org/wikipedia/commons/f/f3/MPTCP-session-en.png

WIKIPEDIA. (2018). Multipath TCP. Retrieved October 19, 2018, from https://en.wikipedia.org/wiki/Multipath_TCP

WIKIPEDIA. (2018). SOCKS. Retrieved September 30, 2018, from https://en.wikipedia.org/wiki/SOCKS

X.Wei, C.Xiong, E. Lopez, Fortinet. (2015). MPTCP proxy mechanisms. Retrieved September 10, 2017, from https://datatracker.ietf.org/doc/draft-wei-mptcp-proxy-mechanism/

ภาคผนวก

ภาคผนวก ก

บทความวิจัยเรื่อง ประสิทธิภาพการทำงานของ Multipath TCP ในระบบโครงข่ายส่วนบุคคล

บทความวิจัย

การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้า ครั้งที่ 11

11th Conference of Electrical Engineering Network 2019 (EENET 2019)

ประสิทธิภาพการทำงานของ Multipath TCP ในระบบโครงข่ายส่วนบุคคล

Performance of Multipath TCP in Personal Network

สรสิช ศรีเอี่ยมสะอาด[1] และธนัญ จารุวิทยโกวิท

[1]วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรม สาขาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

110/1-4 ถ.ประชาชื่น แขวง ทุ่งสองห้อง เขต หลักสี่ กรุงเทพฯ 10210 โทรศัพท์ 0-2954-7300 ถึง 29 ต่อ 498, 594, 601 E-mail: sorasichmail@gmail.com

บทคัดย่อ

Multipath TCP (MPTCP) เป็นโพรโตคอลที่จะช่วยให้อุปกรณ์ IT ต่างๆ สามารถสื่อสารกับเครื่องแม่ข่ายได้หลายเส้นทางต่อ 1 การเชื่อมต่อ (session) ทำให้ Bandwidth เพิ่มมากขึ้น การรับส่งข้อมูลจึงเสร็จเร็วกว่าการสื่อสารเส้นทางเดียว งานวิจัยนี้ทดสอบประสิทธิภาพการทำงานของ MPTCP ในโครงข่ายส่วนบุคคลผ่านโครงข่าย LAN และ WiFi โดยมีการใช้งาน MPTCP ผ่าน MA-GW (Multinet Aggregate GateWay) และศึกษาผลกระทบของการปรับเปลี่ยนค่าพารามิเตอร์ Buffer size รวมถึง Congestion Control Scheme ของ UE (User Equipment) และ MA-GW ต่อประสิทธิภาพการทำงานของ MPTCP

คำสำคัญ: MPTCP, MA-GW, Congestion Control, Buffer

Abstract

Multipath TCP (MPTCP) is the protocol that allows IT devices communicate with servers using multiple paths per one session. This helps to increase link bandwidth, as the result, data transmission is completed faster than a single path communication. This research examines the performance of MPTCP in private networks, ie. LAN and WiFi networks, using MPTCP via the MA-GW (Multinet Aggregate GateWay). This research aims to study the effect of the buffer size, congestion control scheme of the UE (User Equipment) and MA-GW on the performance of MPTCP.

Keywords: MPTCP, MA-GW, Congestion control, Buffer

1. บทนำ

ในปัจจุบันอุปกรณ์ IT ต่างๆ มักจะมีจะมีอินเตอร์เฟสสำหรับการเชื่อมต่อระบบเครือข่ายอยู่หลายเส้นทาง (path) แต่การสื่อสารระหว่างอุปกรณ์ต้นทาง (UE: User Equipment) และเครื่องแม่ข่าย มักใช้งานเพียงเส้นทางเดียว เพราะข้อจำกัดของโพรโตคอล TCP ที่ถูกออกแบบมาให้ใช้งานได้เพียงเส้นทางเดียวต่อ 1 การเชื่อมต่อ (session) ซึ่งวิธีที่จะทำให้สามารถใช้งานได้หลายเส้นทางต่อ 1 การเชื่อมต่อ ได้นั้นต้องใช้โพรโตคอล Multipath TCP (MPTCP) ทำให้การเชื่อมต่อระหว่าง UE และเครื่องแม่ข่ายไม่ถูกจำกัดด้วยจำนวนเส้นทางการเชื่อมต่ออีกต่อไป แต่ MPTCP ยังมีข้อจำกัดทางในการใช้งานอยู่ เนื่องจาก UE (โทรศัพท์เคลื่อนที่ หรือเครื่องคอมพิวเตอร์) และ เครื่องแม่ข่าย จะต้องรองรับโพรโตคอล MPTCP ถ้าไม่รองรับโพรโตคอล MPTCP จะต้องใช้ MPTCP Proxy หรืออีกชื่อหนึ่งว่า Multinet Aggregate GateWay (MA-GW) มาติดตั้งระหว่างกลางของเครือข่ายที่ใช้ TCP และ MPTCP

งานวิจัยนี้ทดสอบเพื่อประเมินประสิทธิภาพการทำงาน MPTCP บนอุปกรณ์จริงของระบบโครงข่ายส่วนบุคคล โดยจะใช้งาน MPTCP ผ่านโครงข่าย LAN และ WiFi ในเวลาเดียวกัน เพื่อเพิ่มขนาดช่องสัญญาณ และใช้ MA-GW ในการทดสอบ

งานวิจัยจะทดสอบปรับแต่งพารามิเตอร์ Buffer size, Congestion Control Scheme ที่ส่งผลกระทบต่อประสิทธิภาพการทำงานของ MPTCP เพื่อเพิ่มประสิทธิภาพการทำงานให้ดีที่สุด โดยวัดประสิทธิภาพการทำงานของ MPTCP จาก Throughput (ความเร็วในการรับส่งข้อมูล), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) เปรียบเทียบหา Buffer, Congestion Control ค่าใดให้ประสิทธิภาพการทำงานของ LAN และ WiFi ได้ดีที่สุด

2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

[1],[2],[3] MPTCP คือ โพโทคอลที่ช่วยให้ สามารถใช้งานเส้นทางที่เชื่อมต่อกับระบบเครือข่ายหลายๆ เส้นทางได้พร้อมกันต่อ 1 การเชื่อมต่อ เพื่อเพิ่ม Throughput และสามารถสำรองเส้นทางได้ ซึ่งมีเงื่อนไขในการใช้งานคือ UE และเครื่องแม่ข่าย จะต้องรองรับโพรโตคอล MPTCP

[4] MA-GW เป็นอุปกรณ์ที่ทำหน้าที่เป็นตัวกลางในการสื่อสารของอุปกรณ์ที่ใช้งาน MPTCP และอุปกรณ์ที่ใช้งาน TCP, จัดการเรื่อง NAT (Network address translation) และคอยตรวจสอบเครื่องแม่

ข่ายที่รองรับ MPTCP เพื่อส่งผ่าน MPTCP ทำงานทำงานได้อย่างมีประสิทธิภาพ โดยใช้ MA-GW คิดตั้งระหว่าง UE กับ Content ดังรูปที่ 1

รูปที่ 1 การสื่อสารระหว่าง MPTCP กับ TCP โดยใช้ MPTCP Proxy

[6],[7],[8] SOCKS Proxy เป็นโพรโตคอลที่ทำหน้าที่เป็น Proxy Server ซึ่งทำงานในระดับต่ำกว่า HTTP Proxy จึงทำให้ SOCKS มีความยืดหยุ่นในการใช้งานมากกว่า การทำงานของ SOCKS จะใช้ Handshake Protocol แลกเปลี่ยนข้อมูลการเชื่อมต่อกับเครื่องลูกข่าย และพยายามคงสภาพ Header ที่เครื่องลูกข่ายสร้างไว้ ก่อนส่งต่อไปยังอินเทอร์เน็ต SOCKS จะใช้ Socket ในการติดตามการเชื่อมต่อ ทำให้ Header มีขนาดเล็ก อีกทั้งยังสามารถรับส่งข้อมูลประเภท UDP ได้อีกด้วย ในงานวิจัยนี้จะใช้โปรแกรม Shadowsocks Socks5 ซึ่งเป็นโปรแกรม SOCKS Proxy สำหรับใช้งานเป็น MA-GW ทำหน้าที่แยกและรวม Traffic ระหว่างระบบที่รองรับ MPTCP กับระบบ TCP ปกติ เพื่อให้สามารถใช้งาน MPTCP ได้

2.2 งานวิจัยที่เกี่ยวข้อง

[5] ทดสอบวัดประสิทธิภาพจาก UE ซึ่งในกรณีนี้คือโทรศัพท์เคลื่อนที่ ใช้งาน MPTCP สื่อสารกับเครื่องแม่ข่ายโดยตรง อุปกรณ์ในการทดสอบจะใช้งาน WiFi ส่วนบุคคล, WiFi สาธารณะ และ LTE จากผู้ให้บริการใน USA จำนวน 3 ค่าย จำนวน 2 การเชื่อมต่อ โดยการทดสอบจะรับส่งข้อมูลด้วยไฟล์ขนาดที่แตกต่างกัน และเปรียบเทียบประสิทธิภาพการทำงานระหว่าง Single Path กับ Multipath สรุปได้ว่า กรณี Single Path: WiFi ส่วนบุคคลและ WiFi สาธารณะ ให้ค่า Latency ต่ำกว่า LTE ทั้ง 3 ค่าย กรณี Multipath ให้ค่า Latency น้อยที่สุดเมื่อเปรียบเทียบกับ Single Path ทั้งหมด และไฟล์ขนาดเล็ก Single Path จะทำงานได้ดีกว่า Multipath ส่วนไฟล์ขนาดใหญ่ Multipath ทำงานได้ดีกว่า

3 ขั้นตอนการดำเนินงานวิจัย

3.1 ขั้นตอนการออกแบบ

การทดสอบบนอุปกรณ์จริงจะประกอบด้วยเครื่องคอมพิวเตอร์ จำนวน 2 เครื่อง คือ UE และ MA-GW การสื่อสารประกอบด้วย 2 เส้นทาง คือ สาย LAN ความเร็ว 1 Gbps และ WiFi ความเร็ว 867 Mbps ใช้ทดสอบการงานของ MPTCP ที่ MA-GW จะใช้งาน VMware จำลองคอมพิวเตอร์เสมือน 1 เครื่อง การสื่อสารใช้ Host-only ความเร็ว 2 Gbps สำหรับใช้เป็น FTP Server ซึ่งใช้โพรโทคอล TCP เพื่อทดสอบกรณีเครื่องแม่ข่ายไม่รองรับการทำงานของ MPTCP การเชื่อมต่อระหว่างอุปกรณ์ต่างๆ สามารถสรุปได้ดังรูปที่ 2

รูปที่ 2 การเชื่อมต่อระหว่างอุปกรณ์ของระบบที่ออกแบบ

3.2 การทดสอบ

ทดสอบโดยให้ UE ดาวน์โหลดไฟล์ขนาด 5.4 GB (เพื่อแสดงประสิทธิภาพของ MPTCP ได้เต็มที่) จากเครื่องแม่ข่าย ผ่านโพรโทคอล FTP โดยใช้โปรแกรม Wireshark ในการดักจับข้อมูลการสื่อสารที่ UE โดยงานวิจัยนี้จะกรองข้อมูล Packet Retransmission ออกทั้งหมดเพราะบาง Packet โปรแกรม Wireshark คำนวณ TCP Delta เป็นจำนวนติดลบ ซึ่งทำให้ไม่สามารถนำไปคำนวน Jitter ได้ โดย Packet ที่ตัดออกคิดเป็น 0.0025% ของ Packet ทั้งหมด

งานวิจัยนี้จะวัดประสิทธิภาพการทำงานของ MPTCP จากความเร็วในการรับส่งข้อมูล (Throughput), ระยะเวลาที่ใช้ในการดาวน์โหลด (Download Time), ค่าความล่าช้าทางเวลา (Delay) และค่าความแปรปรวนของความล่าช้าทางเวลา (Jitter) จากโปรแกรม Wireshark โดยใช้สูตรคำนวณ ดังนี้

$$\text{Throughput} = \frac{(\text{Total Packets Length} \times 8)}{\text{Download time}} \quad (1)$$

Throughput คำนวณจาก ขนาดข้อมูลที่ได้รับทั้งหมด (หน่วยเป็น Byte) คูณ 8 หารด้วยระยะเวลาในการดาวน์โหลด (หน่วยเป็น Second)

$$\text{Download Time} = \text{Finish Time} - \text{Start Time} \quad (2)$$

Download Time คำนวณจาก เวลาที่ได้รับ Packet สุดท้าย ลบด้วย เวลาที่ได้รับ Packet แรก (หน่วยเป็น Second)

$$\text{Delay} = \frac{\text{Total TCP Delta}}{\text{Number of packets}} \quad (3)$$

Delay คำนวณจาก ผลรวมของ TCP Delta (หน่วยเป็น Second) หารด้วย จำนวน Packets ทั้งหมด

$$\text{Jitter} = \frac{\left(\sum_{i=First\ TCP\ Delta}^{i=Last\ TCP\ Delta} [i - \text{Delay}]^2\right)}{(\text{Number of packets} - 1)} \quad (4)$$

Jitter คำนวณจาก ผลรวมของ [("TCP Delta แต่ละ Packet" ลบด้วย Delay) ยกกำลัง 2] และหารด้วย (All Packet ลบ 1) หน่วยเป็น Second

บทความวิจัย

การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้า ครั้งที่ 11

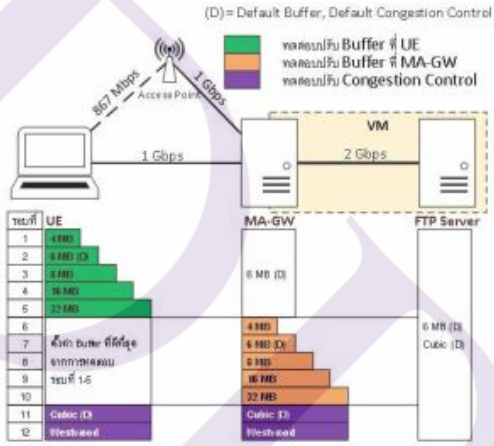
11th Conference of Electrical Engineering Network 2019 (EENET 2019)

ขั้นตอนในการวัดประสิทธิภาพ สามารถแบ่งได้ดังนี้

1) วัดประสิทธิภาพจาก Buffer UE ขนาด 4MB, (Default) 6MB, 8MB, 16MB, 32MB ตั้งค่า Buffer UE ที่ให้ประสิทธิภาพดีที่สุด

2) วัดประสิทธิภาพจาก Buffer MA-GW ขนาด 4MB, (Default) 6MB, 8MB, 16MB, 32MB ตั้งค่า Buffer MA-GW ที่ให้ประสิทธิภาพดีที่สุด

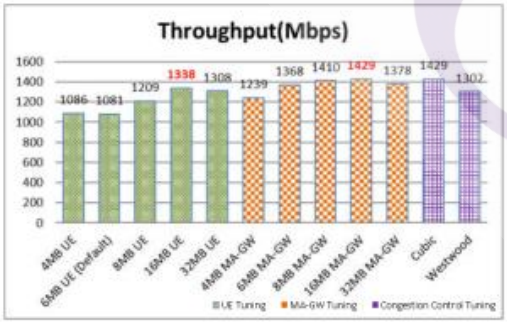
3) วัดประสิทธิภาพจาก Congestion Control ระหว่าง Cubic (Default) กับ Westwood สามารถสรุปได้ดังรูปที่ 3



รูปที่ 3 ลำดับในการวัดประสิทธิภาพของ UE และ MA-GW

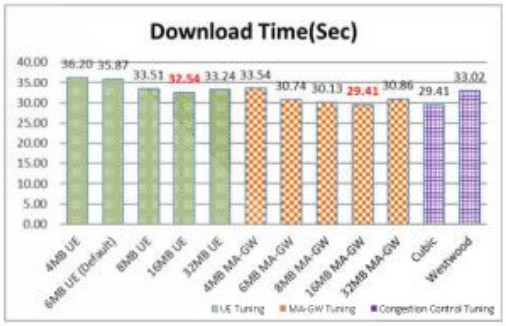
4. ผลการดำเนินงาน

Throughput (ความเร็วในการรับส่งข้อมูล) เมื่อปรับ Buffer UE และ MA-GW สามารถเพิ่มประสิทธิภาพการทำงานได้ดีที่สุดที่ 16MB, ประสิทธิภาพลดลงเล็กน้อยที่ 32MB และปรับ Congestion Control เป็น Cubic ให้ประสิทธิภาพดีที่สุด ดังแสดงในรูปที่ 4



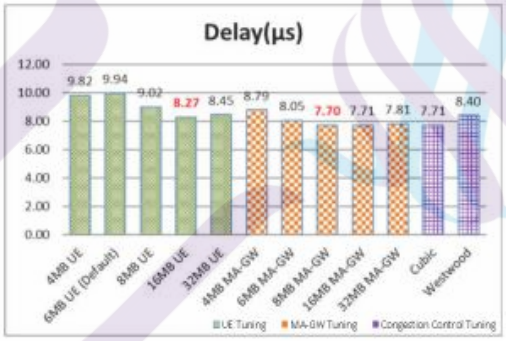
รูปที่ 4 ความเร็วในการรับส่งข้อมูล (Mbps)

Download Time (ระยะเวลาในการดาวน์โหลด) เมื่อปรับ Buffer UE และ MA-GW สามารถลด Download Time ได้ดีที่สุดที่ 16MB, Download Time เพิ่มขึ้นเล็กน้อยที่ 32MB และปรับ Congestion Control เป็น Cubic ให้ประสิทธิภาพดีที่สุด ดังแสดงในรูปที่ 5



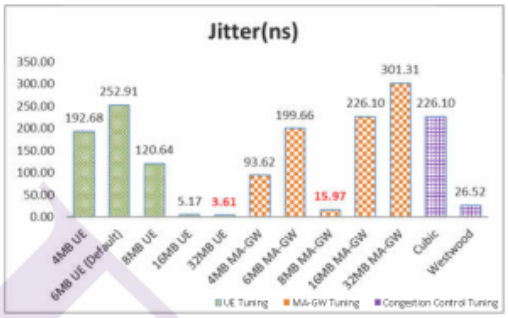
รูปที่ 5 ระยะเวลาในการดาวน์โหลด (Sec)

Delay (ค่าความล่าช้าทางเวลา) เมื่อปรับ Buffer สามารถลด Delay ที่ UE ได้ดีที่สุดที่ 16MB, Delay เพิ่มขึ้นเล็กน้อยที่ 32MB และ MA-GW ได้ดีที่สุดที่ 8MB, 16MB, 32MB โดยไม่ต่างกันอย่างมีนัยสำคัญ และปรับ Congestion Control เป็น Cubic ให้ประสิทธิภาพดีที่สุด ดังแสดงในรูปที่ 6



รูปที่ 6 ค่าความล่าช้าทางเวลา (µs)

Jitter (ค่าความแปรปรวนของความล่าช้าทางเวลา) ผลที่ได้มีความไม่แน่นอน เนื่องจากหน่วยวัดที่ออกมาเป็น (ns) จึงไม่ต่างกันอย่างมีนัยสำคัญ ดังแสดงในรูปที่ 7



รูปที่ 7 ค่าความแปรปรวนของความล่าช้าทางเวลา (ns)

จากการจำลองระบบบนอุปกรณ์จริงตาม Environment ที่ได้ทดสอบ สรุปได้ว่าการปรับขนาด Buffer 16MB ที่ UE และ MA-GW จะทำให้ได้ Throughput, Download Time, Delay ดีที่สุด ทำให้ได้ประสิทธิภาพการทำงานของ MPTCP ดีที่สุด ส่วนค่า Jitter มีค่าสวิงเนื่องจากมีการใช้งาน WiFi บนอุปกรณ์จริงในการทดสอบ ทำให้ยากต่อการควบคุมคลื่นรบกวน แต่โดยภาพรวม Jitter ที่ได้มามีหน่วยเป็น ns ซึ่งเป็นค่าที่อยู่ในเกณฑ์ที่ยอมรับได้

เมื่อทดสอบปรับ Buffer 32MB ประสิทธิภาพการทำงานที่ได้ออกมาจะแย่กว่าการปรับ Buffer 16MB เล็กน้อย ซึ่งงานวิจัยนี้ตรวจสอบพบว่ากรณี Buffer 32MB ระบบจะมี Packet Retransmission มากกว่า Buffer 16MB สิ่งที่งานวิจัยนี้คาดว่าเป็นเหตุที่ Buffer 32MB ทำงานได้แย่กว่า Buffer 16MB อาจจะเนื่องมาจาก

1) Buffer มีใหญ่เกิน ไม่สอดคล้องกับความเร็วในการประมวลผลของ CPU

2) Buffer มีใหญ่เกิน ไม่สอดคล้องกับขนาด Windows size ซึ่งเป็นสิ่งที่ผู้วิจัยตั้งสันนิษฐานไว้

การปรับ Congestion Control แบบ Cubic ที่ UE และ MA-GW จะทำให้ Throughput, Download Time, Delay มีประสิทธิภาพการทำงานของ LAN และ WiFi ได้ดีกว่าการปรับ Congestion Control แบบ Westwood เนื่องจาก Cubic ใช้งานอัลกอริทึมแบบ binary search ที่ขั้นตอน Slow Start, Congestion Avoidance, Fast Retransmit และ Fast Recovery จึงทำให้ทำงานได้เร็วกว่า Westwood ที่ใช้งานอัลกอริทึมแบบ exponential ที่ขั้นตอน Slow Start, ทำงานแบบ linear ที่ขั้นตอน Congestion Avoidance และตั้ง ssthresh ประมาณ transit capacity ที่ขั้นตอน Fast Retransmit และ Fast Recovery

5. สรุปผลการศึกษา

การปรับขนาด Buffer size ของ UE และ MA-GW ภายใต้ Environment ที่ทดสอบในงานวิจัยนี้มีผลต่อประสิทธิภาพการทำงานของ MPTCP โดยจากผลการทดสอบพบว่าการปรับ Buffer UE 16MB และ MA-GW 16MB สามารถเพิ่ม Throughput สูงสุดถึง 32% เทียบกับการใช้ค่า default ของ Linux Mint เวอร์ชั่น 18 (UE) และ CentOS เวอร์ชั่น 7 (MA-GW) และลด Download Time ได้ถึง 18% แต่การปรับ Buffer มากเกินจะทำให้เกิด Packet Retransmission ในระบบเยอะขึ้น และประสิทธิภาพการทำงานจะลดลง การปรับ Congestion Control แบบ Cubic จะให้ประสิทธิภาพการทำงานดีกว่า Westwood โดยผลการทดสอบของงานวิจัยนี้ Cubic สามารถเพิ่ม Throughput ได้ถึง 12% และลด Download Time ได้ถึง 10%

เอกสารอ้างอิง

[1] MPTCP And LTE-WiFi Aggregation. http://www.adslthailand.com/post/introduction-to-multipath-tcp-lte-wifi-aggregation [Accessed: 25-07-2017].

[2] MPTCP. https://en.wikipedia.org/wiki/Multipath_TCP [Accessed: 19-10-2018].

[3] Decoupling TCP from IP with Multipath TCP. http://multipath-tcp.org/data/MultipathTCP-netsys.pptx [Accessed: 25-10-2017].

[4] MPTCP Proxy. https://www.citrix.com/blogs/2013/08/30/mptcp-netscaler-way [Accessed: 19-10-2018].

[5] Yung-Chih Chen, Yeon-sup Lim, Richard J. Gibbens, Erich M. Nahum, Ramin Khalili and Don Towsley. A Measurement-based Study of MultiPath TCP Performance over Wireless Networks. In Proceedings of the 2013 conference on Internet measurement conference, pages 455-468. ACM, 2013.

[6] SOCKS. https://en.wikipedia.org/wiki/SOCKS [Accessed: 30-08-2018].

[7] SOCKS. http://whatis.techtarget.com/definition/socks [Accessed: 30-08-2018].

[8] shadowsocks.org. (2018). shadowsocks. Retrieved October 19, 2018, from https://shadowsocks.org/en/spec/Implementations.html [Accessed: 19-10-2018].

นายสรชิช ศรีเอี่ยมสะอาด

ตำแหน่ง: Network Operation Center

บริษัท นิภาเทคโนโลยี จำกัด

การศึกษา: กำลังศึกษาปริญญาโทวิศวกรรมคอมพิวเตอร์และโทรคมนาคม คณะวิศวกรรมศาสตร์ วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรม มหาวิทยาลัยธุรกิจบัณฑิตย์

งานวิจัยที่สนใจ: SDN, NFV, Proxmox VE tuning, Ceph Storage tuning

ภาคผนวก ข

ผลการทดสอบการทำงานของ MPTCP

UE Tuning TCP_MEM															
UE Tuning	MA-GW		UE		Avg Throughput(Mbps)	Max Throughput(Mb)	Min Throughput(Mb)	Total Packet Length(B)	Download Time(Sec)	Delay(µs)	Min Delay(µs)	Max Delay(µs)	Jitter(ns)	Min jitter(ns)	Max jitter(ns)
	rmem	wmem	rmem	wmem											
4MB UE	6M	4M	4M	4M	1086	1141	1038	5148562371	36.20	9.82	1.324	393,755	192.68	0	155,034,920
6MB UE (Default)	6M	4M	6M	4M	1081	1139	1027	5067495456	35.87	9.94	1.318	398,635	252.91	0	158,901,994
8MB UE	6M	4M	8M	4M	1209	1354	1127	5296020550	33.51	9.02	1.347	394,383	120.64	1.86961E-10	155,530,760
16MB UE	6M	4M	16M	4M	1338	1357	1322	5706449165	32.54	8.27	1.426	202,294	5.17	1.11035E-10	40,919,481
32MB UE	6M	4M	32M	4M	1308	1321	1293	5695845228	33.24	8.45	1.312	84,376	3.61	0	7,117,945

MA-GW Tuning TCP_MEM															
MA-GW Tuning	MA-GW		UE		Avg Throughput(Mbps)	Max Throughput(Mb)	Min Throughput(Mb)	Total Packet Length(B)	Download Time(Sec)	Delay(µs)	Min Delay(µs)	Max Delay(µs)	Jitter(ns)	Min jitter(ns)	Max jitter(ns)
	rmem	wmem	rmem	wmem											
4MB MA-GW	4M	4M	16M	4M	1239	1345	1091	5436262432	33.54	8.79	1.317	376,662	93.62	9.34622E-11	141,867,273
6MB MA-GW	6M	6M	16M	4M	1368	1427	1274	5508347114	30.74	8.05	1.390	1	199.66	4.73633E-11	1
8MB MA-GW	8M	8M	16M	4M	1410	1431	1386	5568773055	30.13	7.70	1.322	194,865	15.97	2.71933E-11	37,969,370
16MB MA-GW	16M	16M	16M	4M	1429	1444	1412	5506097046	29.41	7.71	1.391	1,194,586	226.10	3.05557E-12	1,427,017,346
32MB MA-GW	32M	32M	16M	4M	1378	1449	1327	5567140126	30.86	7.81	1.335	1,640,558	301.31	1.34298E-11	2,691,405,418

Congestion Control Tuning															
Congestion Control Tuning	MA-GW		UE		Avg Throughput(Mbps)	Max Throughput(Mb)	Min Throughput(Mb)	Total Packet Length(B)	Download Time(Sec)	Delay(µs)	Min Delay(µs)	Max Delay(µs)	Jitter(ns)	Min jitter(ns)	Max jitter(ns)
	rmem	wmem	rmem	wmem											
Cubic	16M	16M	16M	4M	1429	1444	1412	5506097046	29.41	7.71	1.391	1,194,586	226.10	3.05557E-12	1,427,017,346
Westwood	16M	16M	16M	4M	1302	1359	1271	5629572262	33.02	8.40	1.399	248,393	26.52	0	61,694,674

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายสรสิช ศรีเอี่ยมสะอาด
ประวัติการศึกษา	ปริญญาตรีวิทยาศาสตรบัณฑิต สาขา เทคโนโลยีสารสนเทศและการสื่อสาร (แขนง เครือข่ายและการสื่อสาร) มหาวิทยาลัยราชภัฏนครปฐม
ตำแหน่งและสถานที่ทำงานปัจจุบัน	Network Operation Center บริษัท นิภา เทคโนโลยี จำกัด