

การประเมินประสิทธิภาพของวิธีควบคุมความคับคั่งแบบ Coupled และ  
Uncoupled ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล  
ในโครงข่าย MPTCP

ผกาศิตา ศรีภักดี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม วิทยาลัยนวัตกรรมด้าน  
เทคโนโลยีและวิศวกรรมศาสตร์  
มหาวิทยาลัยธุรกิจบัณฑิตย์

พ.ศ. 2562

**Performance Evaluation of Coupled and Uncoupled Congestion Control  
Algorithm in Case of Different Throughput in MPTCP Network**

**Phakasita Sripukdee**

**A Thesis 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**



## ใบรับรองวิทยานิพนธ์

วิทยาลัยนวัตกรรมการด้านเทคโนโลยีและวิศวกรรมศาสตร์

มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อวิทยานิพนธ์ การประเมินประสิทธิภาพของวิธีควบคุมความคับคั่งแบบ Coupled และ Uncoupled ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล ในโครงข่าย MPTCP

เสนอโดย นางสาวศกาศิตา ศรีภักดี

สาขาวิชา วิศวกรรมคอมพิวเตอร์และโทรคมนาคม

อาจารย์ที่ปรึกษาวิทยานิพนธ์ อาจารย์ ดร.ธนัญ จารุวิทย์โกวิท

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบวิทยานิพนธ์แล้ว

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร.ฉัญฉกร วุฒิสิริกฤตกิจ)

..... กรรมการและอาจารย์ที่ปรึกษาวิทยานิพนธ์

(อาจารย์ ดร.ธนัญ จารุวิทย์โกวิท)

..... กรรมการ

(อาจารย์ ดร.ชัยพร เขมะภาคะพันธ์)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.มัชฌิกา อ่องแดง)

วิทยาลัยนวัตกรรมการด้านเทคโนโลยีและวิศวกรรมศาสตร์รับรองแล้ว

..... คณบดีวิทยาลัยนวัตกรรมการด้านเทคโนโลยีและวิศวกรรมศาสตร์

(ผู้ช่วยศาสตราจารย์ ดร.ณรงค์เดช กิรติพรานนท์)

วันที่ 19 เดือน ก.พ. พ.ศ. 2562

หัวข้อวิทยานิพนธ์	การประเมินประสิทธิภาพของวิธีควบคุมความคับคั่งแบบ Coupled และ Uncoupled ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลในโครงข่าย MPTCP
ชื่อผู้เขียน	ผกาศิตา ศรีภักดี
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ธนัญ จารุวิทย์โกวิท
สาขาวิชา	วิศวกรรมคอมพิวเตอร์และโทรคมนาคม
ปีการศึกษา	2561

### บทคัดย่อ

ในปัจจุบัน Multipath TCP (MPTCP) เป็นโปรโตคอลที่ช่วยในการเพิ่มประสิทธิภาพและความเร็วของการรับ-ส่งข้อมูลในเวลาเดียวกัน ด้วยการเชื่อมต่อหลากหลายเส้นทางบนอุปกรณ์สมาร์ตโฟน ทั้งการใช้งานในโครงข่าย LTE หรือ WiFi ก็ตาม ซึ่งแนวทางที่สามารถเพิ่มประสิทธิภาพของระบบได้คืออีกทางหนึ่งคือการใช้อัลกอริทึมควบคุมความคับคั่งในการจัดการและปรับอัตราการรับ-ส่งข้อมูลในละเส้นทางที่มีการเปลี่ยนแปลงสถานะเครือข่ายจนถึงสถานะที่ก่อให้เกิดความคับคั่งของข้อมูล (Congested Network) เพื่อให้สอดคล้องกับการใช้งานอินเทอร์เน็ตแบบ TCP-friendly โดยงานวิจัยนี้ศึกษาและเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมแบบ Coupled (wVegas) และ Uncoupled (CUBIC) ในโครงข่าย MPTCP (LTE และ WiFi) ด้วยการจำลองการรับ-ส่งไฟล์ข้อมูลจริงบนโปรแกรม NS3 ผลการวิจัยพบว่า ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลมากๆ ประสิทธิภาพการทำงานของอัลกอริทึมแบบ Coupled (wVegas) มี Throughput ที่ใกล้เคียงกับ TCP (ซึ่งที่มี Throughput มากสุด) และ Coupled (wVegas) ให้ Throughput ที่สูงกว่าการใช้งาน Uncoupled (CUBIC)

Thesis Title	Performance Evaluation of Coupled and Uncoupled Congestion Control Algorithm in Case of Different Throughput in MPTCP Network
Author	Phakasita Sripukdee
Thesis Advisor	Dr. Tanun Jaruvitayakowit
Department	Computer and Telecommunication Engineering
Academic Year	2018

### **Abstract**

Currently, Multipath TCP (MPTCP) is a system that increases the efficiency and speed of receiving and sending data at the same time by several paths on smartphone in an LTE network or WiFi. Well, another way is to use algorithms to manage and adjust the data-transmission rate in the path that There is a change of the network to Congested Network in order to achieve TCP-friendly Internet deployment. This research studies and compares the functions of the Coupled (wVegas) and Uncoupled (CUBIC) algorithms. In the MPTCP (LTE and WiFi) network by simulating the real-time transmission of data files on the NS3 program. The work of the Coupled (wVegas) algorithm has a throughput close to TCP (Highest Throughput side) and Coupled (wVegas), providing a higher capacity than Uncoupled (CUBIC) algorithm usage.

## กิตกรรมประกาศ

การจัดทำวิทยานิพนธ์นี้สำเร็จสมบูรณ์ได้ด้วยความเมตตากรุณาจากท่านอาจารย์ ดร. ธนัญ จารุวิทย์โกวิท อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้คำปรึกษาชี้แนะแนวทางการทำวิจัยรวมถึงการจัดทำวิทยานิพนธ์ทุกขั้นตอนและทุกกระบวนการ ตลอดทั้งอาจารย์ ดร.ชัยพร เขมะภาคะพันธ์ และผู้ช่วยศาสตราจารย์ ดร.มัชฌิมา อ่องแดง ที่ซึ่งคอยชี้แนะแนวทางการทำวิทยานิพนธ์ และรองศาสตราจารย์ ดร.ลัญจกร วุฒิสัทธาคุณกิจ มาเป็นประธานในการสอบเปิดเล่ม และสอบปิดเล่ม จนถึงคณะกรรมการทุกท่าน ที่ชี้แนะแนวทางจนสามารถสอบวิทยานิพนธ์ได้แล้วเสร็จสมบูรณ์ ผู้วิจัยรู้สึกซาบซึ้งในความช่วยเหลือ และขอขอบพระคุณอย่างสูงจากใจจริง

ขอขอบพระคุณ พ.อ.อ.ปฐมชัยพงศ์ ศรีภักดี และ นางปวิชญานินทร์ ศรีภักดี ผู้ซึ่งเป็นบิดาและมารดา ตลอดจนครอบครัวที่คอยเป็นกำลังใจและคอยให้ความรักมาโดยตลอดเวลา

ขอขอบคุณเพื่อนๆ และพี่น้อง ทั้งที่ทำงานและวิทยาลัยนวัตกรรมการด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์ ที่คอยช่วยเหลือและให้คำปรึกษา ตลอดจนเป็นกัลยาณมิตรที่ดีต่อกันเสมอมา

และขอขอบพระคุณทุกท่านที่มีได้กล่าวนามมา ณ ที่นี้ ที่มีส่วนช่วยเหลือและเป็นกำลังใจ รวมทั้งให้การสนับสนุนเป็นอย่างดีจนทำให้การศึกษาครั้งนี้สำเร็จลุล่วงด้วยดี หากข้าพเจ้า มีความบกพร่องประการใด ข้าพเจ้าขอน้อมรับไว้แต่เพียงผู้เดียว และยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ในการพัฒนางานวิจัยต่อไป

คุณค่าและประโยชน์ใดๆ ที่อาจมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบเป็นเครื่องบูชาพระคุณของบิดามารดาที่ให้กำเนิดและเลี้ยงดูให้การศึกษา ตลอดจนครูบาอาจารย์และผู้ที่มีพระคุณทุกท่านที่มีส่วนในการวางรากฐานการศึกษาให้แก่ผู้วิจัย

ผกาศิตา ศรีภักดี

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ฉ
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญตาราง.....	ช
สารบัญภาพ.....	ฎ
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	3
1.3 ขอบเขตของงานวิจัย.....	3
1.4 แผนการดำเนินงาน .....	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.6 วัสดุอุปกรณ์.....	5
1.7 ผลงานที่ตีพิมพ์.....	5
2. แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง.....	6
2.1 สถาปัตยกรรมของ LTE .....	6
2.2 Multipath Transmission Control Protocol : MPTCP .....	9
2.3 การทำงานของ MPTCP .....	11
2.4 วิธีควบคุมความคับคั่ง.....	14
2.5 การมอดูเลชันบนโครงข่ายไร้สาย .....	17
2.6 วิธีควบคุมความคับคั่งแบบ wVegas .....	21
2.7 วิธีควบคุมความคับคั่งแบบ CUBIC.....	23
2.8 โปรแกรมจำลองโครงข่าย (NS3).....	26
2.9 งานวิจัยที่เกี่ยวข้อง .....	30

## สารบัญ (ต่อ)

บทที่	หน้า
3. ระเบียบวิธีวิจัย.....	34
3.1 ขั้นตอนการออกแบบการทดสอบบนโครงข่าย LTE และ WiFi .....	34
3.2 ค่าตัวแปรของโครงข่าย LTE และ WiFi .....	36
3.3 แผนการดำเนินงาน .....	42
3.4 ขั้นตอนและวิธีการดำเนินงาน .....	43
4. ผลการศึกษา .....	47
4.1 โครงข่าย LTE และ WiFi ในการทดสอบ.....	47
4.2 ขั้นตอนในการทดสอบ .....	48
4.3 การทดสอบ.....	49
4.4 อภิปรายผลการวิจัย .....	66
5. บทสรุป และข้อเสนอแนะ .....	68
5.1 สรุปผลการวิจัย.....	68
5.2 ข้อจำกัดและแนวทางแก้ไขของงานวิจัย.....	69
5.3 ข้อเสนอแนะและแนวทางการพัฒนาในอนาคต.....	70
บรรณานุกรม.....	71
ประวัติผู้เขียน.....	75



## สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน .....	4
2.1 ช่อง CQI และ MCS .....	18
2.2 แสดง MCS บนโครงข่าย LTE.....	19
2.3 แสดง MCS บนโครงข่าย WiFi ระหว่าง Type N กับ Type AC .....	21
2.4 เปรียบเทียบคุณสมบัติงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ .....	33
3.1 ขนาดความจุของการเชื่อมต่อ และความล่าช้าระหว่างอุปกรณ์ในโครงข่าย LTE	36
3.2 ขนาดความจุของการเชื่อมต่อ และความล่าช้าระหว่างอุปกรณ์ ในโครงข่าย WiFi .....	36
3.3 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย WiFi โดยมีระยะห่างระหว่างอุปกรณ์ ของผู้ใช้งานและสถานีฐานเป็นระยะเดียว โดยใช้วิธีควบคุมความคับคั่งแบบ CUBIC .....	37
3.4 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย LTE โดยมีระยะห่างระหว่างอุปกรณ์ ของผู้ใช้งานและสถานีฐานในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธี ควบคุมความคับคั่งแบบ CUBIC .....	38
3.5 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย MPTCP โดยมีระยะห่างระหว่าง อุปกรณ์ของผู้ใช้งานและสถานีฐานบนโครงข่าย WiFi เป็นระยะเดียว และ โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานบนโครงข่าย LTE ในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ CUBIC .....	39
3.6 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย WiFi โดยมีระยะห่างระหว่างอุปกรณ์ ของผู้ใช้งานและสถานีฐานเป็นระยะเดียว โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas .....	40

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
3.7 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย LTE โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas .....	40
3.8 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย MPTCP โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานบนโครงข่าย WiFi เป็นระยะเดียวและ โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานบนโครงข่าย LTE ในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas .....	41
3.9 แผนการดำเนินงาน .....	43
4.1 แสดงข้อมูล $TCP_{Max}$ Throughput ผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว โดยใช้ Uncoupled (CUBIC) .....	49
4.2 แสดงข้อมูล อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) เฉลี่ยผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว โดยใช้ Uncoupled (CUBIC) .....	51
4.3 แสดงข้อมูล MPTCP Throughput ผ่านโครงข่าย LTE และ WiFi พร้อมกัน โดยใช้ Uncoupled (CUBIC) .....	52
4.4 แสดงข้อมูล อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) เฉลี่ยผ่านโครงข่าย LTE และ WiFi พร้อมกัน โดยใช้ Uncoupled (CUBIC).....	53
4.5 แสดงค่า Throughput ของ CUBIC MPTCP เทียบกับ Throughput ของ CUBIC $TCP_{Max}$ ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE .....	54
4.6 แสดงข้อมูล $TCP_{Max}$ Throughput ผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว โดยใช้ Coupled (wVegas) .....	56

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.7 แสดงข้อมูล อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) เฉลี่ยผ่านโครงข่าย LTE หรือWiFi เพียงเส้นทางเดียว โดยใช้ Coupled (wVegas)	58
4.8 แสดงข้อมูล MPTCP Throughput ผ่านโครงข่าย LTE และ WiFi พร้อมกัน โดยใช้ Coupled (wVegas) .....	59
4.9 แสดงข้อมูล อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) เฉลี่ยผ่านโครงข่าย LTE และ WiFi พร้อมกัน โดยใช้ Coupled (wVegas) .....	60
4.10 แสดงค่า Throughput ของ wVegas MPTCP เทียบกับ Throughput ของ wVegas TCP <sub>Max</sub> ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE .....	61
4.11 แสดงค่า TCP Throughput สูงสุด ของตามตารางแสดง MCS บนโครงข่าย WiFi และ LTE .....	63
4.12 แสดงค่า MPTCP Throughput บนโครงข่าย WiFi และ LTE ในกรณีที่ระบบให้ค่า TCP Throughput สูงสุด .....	64
4.13 แสดงข้อมูล MPTCP Throughput ระหว่าง Uncoupled (CUBIC) เทียบกับ Coupled (wVegas) ผ่านโครงข่าย LTE และ WiFi พร้อมกัน	65

สารบัญภาพ

ภาพที่	หน้า
1.1 สถาปัตยกรรมของ MPTCP.....	2
2.1 รูปแบบอ้างอิงของโครงข่าย LTE .....	7
2.2 แสดงเส้นทางการส่งข้อมูลจากการส่งระหว่าง UE และ Internet โดย (a) จาก UE ถึง Internet และ (b) จาก Internet มายัง UE .....	8
2.3 การส่งข้อมูลดำเนินการของ MPTCP .....	10
2.4 แสดงการทำงานและความแตกต่างระหว่าง SPTCP กับ MPTCP โดย Client 1, Client 3 มีการเชื่อมต่อแบบ SPTCP คือ มีเชื่อมต่อผ่านอินเทอร์เน็ตเพียงเดียว และ Client 2 มีการเชื่อมต่อแบบ MPTCP คือมีการเชื่อมต่อ มากกว่าหนึ่งอินเทอร์เน็ตเพียง โดยการสร้าง SubFlow ของ MPTCP .....	12
2.5 มีการทำงาน MPTCP โดยมีการเชื่อมต่อทั้ง 2 เส้นทาง หากการเชื่อมต่อขาดหายไป เช่นในเส้นทางที่เลข 5 และ 7 ขาดหายไป (ระหว่าง A2-B2) จะถูกนำมาจัดเรียงชุดข้อมูลอีกรอบเพื่อดำเนินการส่งกลับไปอีกเส้นทาง (ระหว่าง A1-B1) ต่อไป.....	13
2.6 แสดง Coupled Congestion Control ของ Client 2 ราย โดย Client แรก ใช้การส่งข้อมูลแบบ MPTCP และอีกราย คือ TCP ปกติ .....	17
3.1 โครงข่าย 4G LTE และ WiFi ที่ใช้ในงานวิจัย .....	35
3.2 รูปแบบ Tracing and packet traces .....	46
4.1 โครงข่าย 4G LTE และ WiFi ที่ใช้ในงานวิจัย .....	47
4.2 กราฟแสดงค่า TCP Throughput ของ CUBIC ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE .....	50
4.3 กราฟแสดงค่า Throughput ของ CUBIC MPTCP เทียบกับ Throughput ของ CUBIC TCP <sub>Max</sub> ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE .....	55

## สารบัญภาพ (ต่อ)

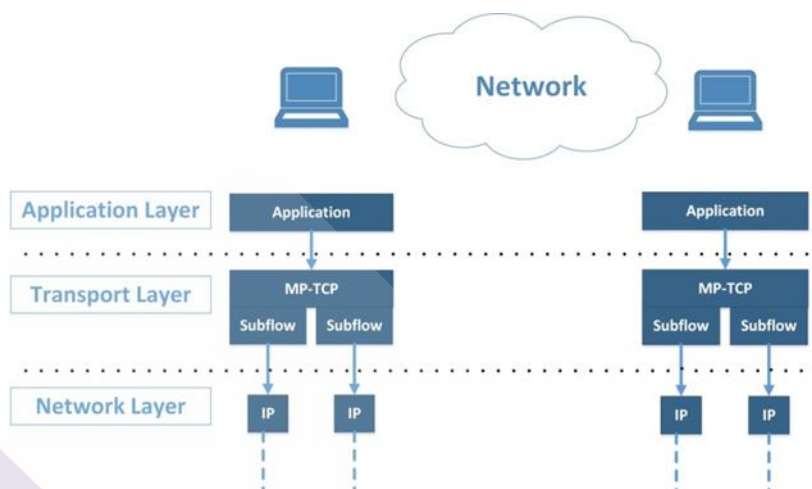
ภาพที่	หน้า
4.4 กราฟแสดงค่า TCP Throughput ของ wVegas ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE .....	57
4.5 กราฟแสดงค่า Throughput ของ wVegas MPTCP เทียบกับ Throughput ของ wVegas TCP <sub>Max</sub> ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE .....	62
4.6 แสดงตัวอย่างการส่งแพ็กเก็ตข้อมูลบนโครงข่าย LTE และ WiFi จาก UE ไปยัง Server .....	67

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีทางด้านอินเทอร์เน็ตโครงข่ายแบบไร้สาย มีความก้าวหน้าและใช้งานกันอย่างกว้างขวาง การใช้งานอินเทอร์เน็ตแบบไร้สายบนเครื่องโทรศัพท์เคลื่อนที่นั้นมีการเชื่อมต่ออินเทอร์เน็ตผ่านโครงข่ายหลัก คือ LTE และ WiFi โครงข่ายทั้งสองมีการทำงานร่วมกันด้วยเทคโนโลยี LTE-WiFi Aggregation เป็นการทำงานร่วมกันในระดับ TCP (Transmission Control Protocol) ที่เรียกว่า Multipath TCP (MPTCP) เพื่อให้ผู้ใช้งานสามารถใช้งานอินเทอร์เน็ตได้อย่างต่อเนื่อง โปรโตคอล MPTCP บนอินเทอร์เน็ตแบบไร้สาย เป็นโปรโตคอลที่ช่วยในการเพิ่มประสิทธิภาพและความเร็วของการรับ-ส่งข้อมูลในเวลาเดียวกันบนการเชื่อมต่อหลากหลายเส้นทาง MPTCP เป็นส่วนเสริมของ TCP ที่มีการอนุญาตให้มีการเชื่อมต่อเพื่อสร้าง TCP-based หลายๆ เส้นทาง ที่เรียกว่า Subflow สำหรับให้บริการการรับ-ส่งข้อมูลแบบ multi-paths ในการรับ-ส่งข้อมูล การดำเนินการเพิ่มการเชื่อมต่อย่อยเพิ่มเติมจะทำให้ three-way handshake เช่นเดียวกับการดำเนินการเริ่มต้นด้วย TCP เส้นทางใหม่ เพื่อเป็นการเลียนแบบการเชื่อมต่อ TCP แบบปกติบนเส้นทางของโครงข่าย เพื่อช่วยลดความยุ่งยากในการตั้งค่าและการเชื่อมต่อ ในการเชื่อมต่อนั้นหากมีปริมาณความหนาแน่นในการรับ-ส่งของข้อมูลบนโครงข่ายเป็นจำนวนมากเกินไป จะก่อให้เกิดความคับคั่งของข้อมูล (Congestion) ซึ่งอาจทำให้ใช้เวลาในการรับ-ส่งข้อมูลนานขึ้น หรือข้อมูลสูญหายเนื่องจากการ drop packet ได้



ภาพที่ 1.1 สถาปัตยกรรมของ MPTCP

ปัญหาหลักในการใช้งานอินเทอร์เน็ตแบบไร้สาย ทั้งโครงข่าย LTE และ WiFi คือการใช้งานและรับ-ส่งข้อมูลบนเครื่องโทรศัพท์เคลื่อนที่ โดย การนำโปรโตคอล MPTCP มาใช้เพื่อเพิ่ม Bandwidth จะเป็นตัวช่วยให้ระบบสามารถเพิ่ม throughput ได้แต่ถึงกระนั้น โปรโตคอล MPTCP ยังมีข้อจำกัดซึ่งหลายคนจะมองข้าม คือ เมื่อมีการส่งข้อมูลจากต้นทางไปยังปลายทาง ในกรณีที่มี throughput ต่างๆ กันมากๆ ตามสมมติฐาน เมื่อใช้ Congestion control แบบ Uncoupled (CUBIC) จะเกิดปัญหา Throughput โดยรวมของ MPTCP แย่กว่า Throughput ของ TCP ในฝั่งของ Throughput ที่มาก แต่ การใช้ Congestion control แบบ Coupled (wVegas) ในกรณีที่เลวร้ายสุดของ Throughput โดยรวมของ MPTCP เมื่อ Throughput 2 ฝั่งต่างกันมากๆ คือ ได้ Throughput เท่ากับฝั่งของ TCP ที่มีปริมาณ Throughput มากสุด ซึ่งในปัจจุบันยังไม่มีการศึกษาว่าความต่างของ Throughput ทั้ง 2 ฝั่งที่จำนวนเท่าใดจึงส่งผลให้ Throughput MPTCP นั้นแย่กว่า Throughput TCP ในฝั่งมาก ทั้งนี้ในการทำงานของ MPTCP นั้น มีการนำหลักการของการควบคุมความคับคั่งของข้อมูล (Congestion Control) มาใช้ซึ่ง Congestion Control สำหรับ Multipath Path ในงานวิจัยนี้ นำมาทดสอบ 2 รูปแบบ คือ Uncoupled Congestion Control เป็นรูปแบบที่ง่ายที่สุดของการควบคุมความคับคั่งสำหรับ MPTCP โดยมีการจัดการแต่ละ Subflows เหมือนการเชื่อมต่อแบบอิสระของ TCP ในรูปแบบของ TCP Congestion Control แต่เนื่องจาก Congestion Control ในแบบนี้มีการแชร์ทรัพยากรในจุดคอขวด จึงส่งผลให้ Multi-path Flow เป็นไปอย่างไม่ยุติธรรม ตัวอย่างเช่น Vegas, Reno, CUBIC, BIC, H-TCP, Hybla, Westwood เป็นต้น อีกรูปแบบคือ Coupled Congestion Control เป็นแนวคิดในการแก้ไขปัญหาความไม่ยุติธรรมของ Uncoupled Congestion Control บนการใช้ทรัพยากรร่วมกันบนจุดคอขวด โดยการจับคู่ Congestion Windows ของทุก Subflows

บนการเชื่อมต่อของ MPTCP ด้วยหลักการรวมทรัพยากร หลักการของ Coupled Congestion Control ช่วยให้โปรโตคอลในการขนส่งสามารถปรับเปลี่ยน Congestion Windows ของแต่ละ Subflow และตรวจสอบให้เป็นไปตามมาตรฐานความยุติธรรมและความเป็นธรรมในจุดคอขวดบนเครือข่าย ตัวอย่างเช่น LIA, OLIA, Balia, wVegas เป็นต้น

ผู้วิจัยมีแนวคิดที่จะทำการทดสอบการใช้งานโปรโตคอล MPTCP บนโครงข่าย LTE และ WiFi โดยจำลองสถานการณ์โดยใช้โปรแกรม NS3DCE ให้ MPTCP 2 ฟังก์ชัน มี Throughput ต่างกัน โดยให้โหลดไฟล์ข้อมูลขนาด 229.4 MB จากไฟล์ข้อมูลจริง บนโครงข่าย MPTCP โดยใช้วิธีควบคุมความคับคั่งแบบ Coupled และ Uncoupled จากนั้นใช้โปรแกรม Wireshark ในการตรวจสอบโดย Export ผลลัพธ์ออกมาเป็น Trace file เพื่อวิเคราะห์ด้วยพารามิเตอร์ Throughput , อัตราล่าช้าต่อหน่วยเวลา (Delay) และอัตราความแปรปรวนของความล่าช้าของข้อมูล (Jitter)

## 1.2 วัตถุประสงค์ของงานวิจัย

1. เปรียบเทียบประสิทธิภาพของโปรโตคอล TCP และ MPTCP ในกรณีที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล ในโครงข่าย LTE และ WiFi
2. หาผลต่างของอัตราการรับ-ส่งข้อมูลที่ทำให้ MPTCP มี Throughput โดยรวมน้อยกว่า TCP
3. เพื่อที่จะเปรียบเทียบประสิทธิภาพการทำงานของ Congestion Control ระหว่าง Uncoupled และ Coupled บนโปรโตคอล MPTCP ในโครงข่าย LTE และ WiFi ในกรณีที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล

## 1.3 ขอบเขตของงานวิจัย

ในงานวิจัยนี้เป็นการจำลองสถานการณ์การใช้งานโปรโตคอล MPTCP บนโครงข่าย LTE และ WiFi ในกรณีที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล บนโครงข่าย

1. ศึกษาและจำลองสถานการณ์การใช้งานโปรโตคอล MPTCP แบบ Coupled และ Uncoupled บนโครงข่าย LTE และ WiFi ด้วยวิธีการส่งข้อมูลวิดีโอจากวิดีโอจริง บนซอฟต์แวร์ NS3 (DCE) Version 1.7
2. เปรียบเทียบประสิทธิภาพของวิธีควบคุมความคับคั่งแบบ Coupled และ Uncoupled ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล ในโครงข่าย MPTCP
3. จำลองสถานการณ์การใช้งานโปรโตคอล MPTCP แบบ Coupled ด้วย wVegas และ Uncoupled ด้วย CUBIC



4. เปรียบเทียบประสิทธิภาพของโปรโตคอล MPTCP แบบ Coupled และ Uncoupled ที่ใช้งานบนโครงข่าย LTE และ WiFi โดยตัวแปรที่ใช้ในการวิเคราะห์คือ Throughput , อัตราล่าช้าต่อหน่วยเวลา (Delay) และอัตราความแปรปรวนของความล่าช้าของข้อมูล (Jitter)

#### 1.4 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

งาน	เดือน						
	ม.ย.-ก.ค.61	ส.ค.-ก.ย.61	ต.ค.-พ.ย.61	ธ.ค. 61 -ม.ค.62	ก.พ.-มี.ค.62	เม.ย.-พ.ค. 62	
ศึกษาข้อมูลและรวบรวมงานวิจัย MPTCP แบบ Coupled และ Uncoupled							
ศึกษาหลักการทำงาน MPTCP							
ศึกษาหลักการทำงาน LTE							
ศึกษาหลักการทำงาน WiFi							
ศึกษาหลักการทำงาน MPTCP บน LTE และ WiFi บนโปรแกรมการจำลองเครือข่าย (Network Simulation 3)							
ออกแบบจำลองระบบ MPTCP							
ทดสอบ ตรวจสอบ และปรับปรุงระบบ							
สรุปผลการจำลองระบบและประโยชน์ที่ได้รับ							

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการการทำงานของโปรโตคอล MPTCP แบบ Coupled และ Uncoupled บนโครงข่าย LTE และ WiFi

2. เข้าใจขอบเขตและข้อจำกัดการทำงานของโปรโตคอล MPTCP แบบ Coupled และ Uncoupled บนโครงข่าย LTE และ WiFi

3. ทราบถึงประสิทธิภาพของ MPTCP แบบ Coupled และ Uncoupled บนโครงข่าย LTE และ WiFi

จากประโยชน์ที่คาดว่าจะได้รับดังกล่าว ผู้พัฒนาและผู้ให้บริการโครงข่าย สามารถนำไปพัฒนาบริการ เพื่อพัฒนารูปแบบและประสิทธิภาพของระบบ ให้เหมาะสมสำหรับโครงข่ายในอนาคตต่อไป

## 1.6 วัสดุอุปกรณ์

### ฮาร์ดแวร์

1. Acer Aspire E1-572G (15.6-inch) i7-4500U, CPU 1.8GHz with Turbo Boost up to 3.0 GHz, AMD Radeon™ HD 8750M with 2 GB Dedicated VRAM, 4GB DDR3 L Memory เพื่อวิเคราะห์ผลด้วยโปรแกรม Wireshark

2. MacBook Pro (15-inch, Mid 2012) 2.3GHz Quad-core Intel Core i7 Processor (Turbo Boost up to 3.3GHz) with 6MB L3 cache ,8GB of 1600MHz DDR3 Memory เพื่อจำลองการทดสอบด้วยโปรแกรม NS3

### ซอฟต์แวร์

1. Program Parallels Desktop Control Center for MAC Version 13.0.0 เพื่อใช้งาน Ubuntu Linux 14.04 Desktop 64 bit, CPU 2 Core, Memory 2 GB

2. Program network simulator NS3 DCE Version 1.7 for Linux

3. Program Wireshark version 2.0 4 for Windows 8.1 Enterprise N

โดยดำเนินการติดตั้ง Program Parallels Desktop ลงบน Mac Book Pro แบ่ง CPU 2 Core, Memory 4 GB เพื่อทำการติดตั้ง Ubuntu Linux 14.04 Desktop 64 bit หลังจากติดตั้งเสร็จดำเนินการ Install Program network simulator 3 DCE Version 1.7

## 1.7 ผลงานที่ตีพิมพ์

ผกาศิตา ศรีภักดี, “การประเมินประสิทธิภาพของวิธีควบคุมความคับคั่งแบบ *Coupled* และ *Uncoupled* ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล ในโครงข่าย *MPTCP*”, Digital Technology for Sustainable Wellbeing and Smart Society, NCITE'14

## บทที่ 2

### แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

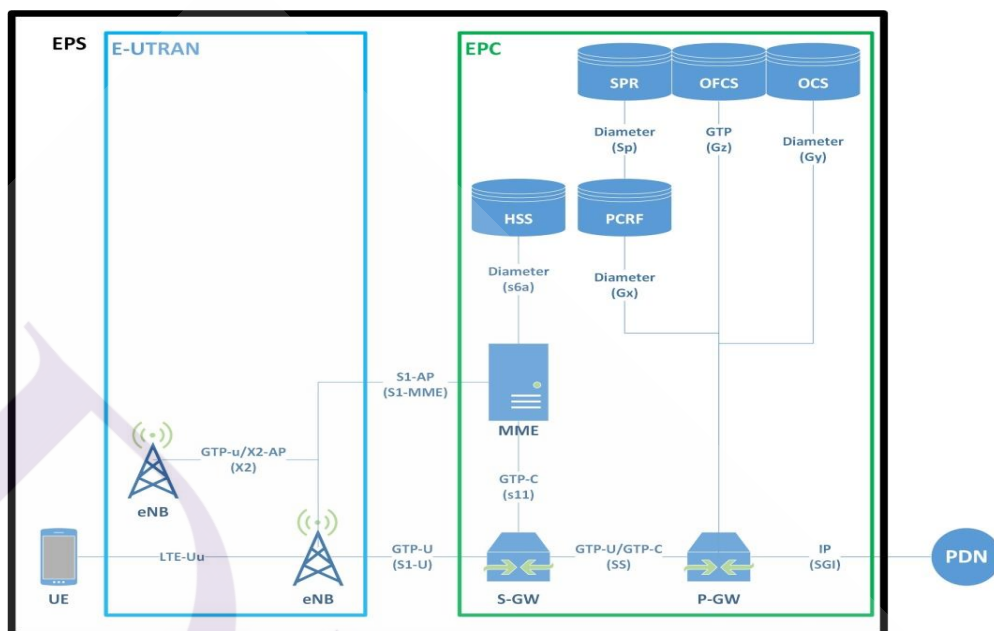
ในบทนี้จะกล่าวถึงทฤษฎีที่นำมาใช้ในงานวิจัย ซึ่งประกอบด้วย 3 สถาปัตยกรรมของ LTE, Multipath Transmission Control Protocol :MPTCP, การทำงานของ MPTCP, การทำงานของ MPTCP แบบ Uncoupled Congestion Control และ Coupled Congestion Control , Network Simulator version 3 (NS3) และงานวิจัยที่เกี่ยวข้อง

#### 2.1 สถาปัตยกรรมของ LTE<sup>1</sup>

ในปัจจุบันเทคโนโลยีมีความก้าวหน้ามากขึ้นอย่างรวดเร็ว ผู้ใช้งานสามารถติดต่อสื่อสารกันได้อย่างสะดวกและง่ายดายยิ่งขึ้น สิ่งสำคัญที่ช่วยให้การติดต่อสื่อสารนั้นเป็นเรื่องง่าย นั่นก็คือเทคโนโลยีสื่อสารไร้สาย LTE ในบทความนี้จะได้นำเสนอสถาปัตยกรรมของโครงสร้างพื้นฐานของโครงข่าย LTE ในยุคปัจจุบัน ซึ่งมีสถาปัตยกรรมโครงสร้างของระบบ ดังภาพที่ 2.1

---

<sup>1</sup> NMC Consulting Group, “LTE Network Architecture: Basic”



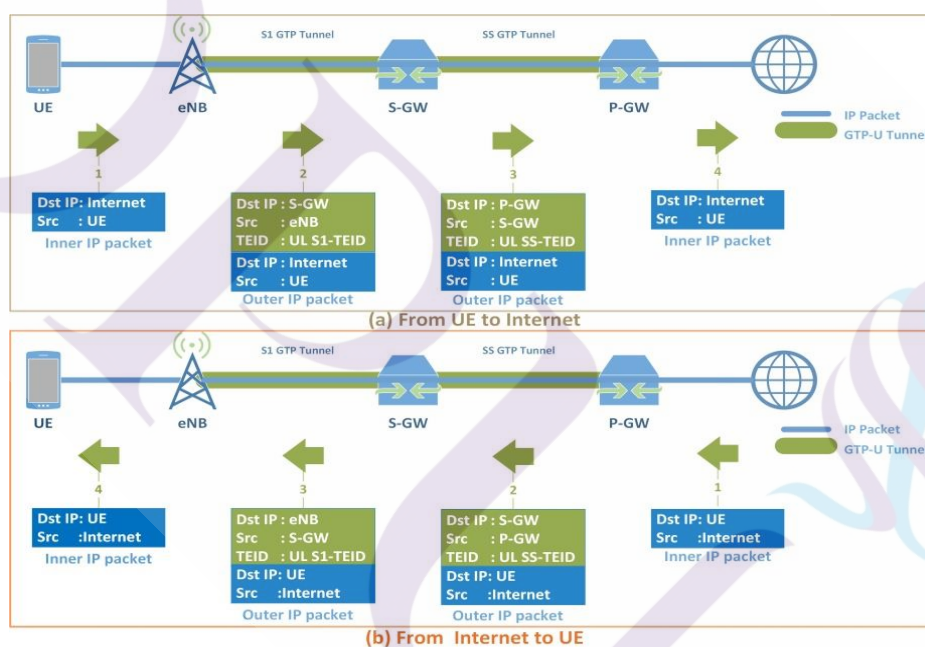
ภาพที่ 2.1 รูปแบบอ้างอิงของโครงข่าย LTE

ที่มา: LTE Network Architecture: Basic เข้าถึง 1 เมษายน 2562 จาก

<http://www.netmanias.com/en/post/techdocs/5904/architecture-lte/lte-network-architecture-basic>

จากภาพที่ 2.1 EPS (Evolved Packet System) คือสถาปัตยกรรมรุ่นต่อไปสำหรับเครือข่ายบรอดแบนด์มือถือในอนาคต EPS ประกอบด้วย the Long Term Evolution (LTE) และ Evolved Packet Core (EPC) ซึ่งการดำเนินการจะเริ่มจาก UE ทำการสื่อสารกับ eNB (evolved Node B) ใน E-UTRAN (Evolved Universal Mobile Telecommunications System Terrestrial Radio Access Network) (LTE) ทาง LTE-Uu เพื่อจัดการและเตรียมความพร้อมสำหรับการเชื่อมต่อไปยังสถานีต่อไปใน EPC โดยจะทำหน้าที่เป็นส่วนควบคุมของเครือข่าย รวมถึงสัญญาณต่าง ๆ ที่ทำงานร่วมกันบนเครือข่าย LTE หลังจากนั้นจะทำการสื่อสารกับ EPC โดยใช้โปรโตคอล S1-AP บนอินเตอร์เฟซ S1-MME สื่อสารกับ Mobility Management Entity (MME) สำหรับ control plane traffic และ ใช้โปรโตคอล GTP-U บนอินเตอร์เฟซ S1-U สื่อสารกับ Serving Gateway (S-GW) สำหรับ user plane traffic ซึ่งอินเตอร์เฟซ S1-MME และ S1-U นี้รวมเรียกว่า อินเตอร์เฟซ S1 จะทำหน้าที่เป็นตัวแทนการเชื่อมต่อจากอินเตอร์เฟซ eNB ต่ EPC

และในส่วนของ EPC จะมี MME (Mobility Management Entity) จากผู้ให้บริการเครือข่ายเป็นผู้ควบคุมการส่งสัญญาณ อาทิเช่น ความเร็ว และปริมาณข้อมูลในการส่ง ซึ่งจะดึงข้อมูลต่างๆจาก HSS (The Home Subscriber Server) เป็นฐานข้อมูลกลางที่เก็บข้อมูลเกี่ยวกับผู้ใช้บริการในเครือข่ายผู้ให้บริการ, PCRF (Policy and Charging Rules Function) จัดการปริมาณการใช้งานข้อมูลภายในโครงข่ายของผู้ใช้บริการ, SPR จัดการข้อมูลสมาชิกให้กับ PCRF, OCS การควบคุมเครดิตหรือการคำนวณค่าใช้จ่ายแบบเรียลไทม์ และ OFCS จัดการการคำนวณค่าใช้จ่ายตาม CDR-based แล้วส่งออกไปยัง PDN (Packet Data Network) ด้วยขา SGi ซึ่ง PDN ดังกล่าวคือ external IP domain ของผู้ให้บริการ ที่ UE ต้องการสื่อสารด้วยนั่นเอง โดยจะมีเส้นทางในการรับ-ส่งข้อมูลระหว่าง UE และปลายทางในโลกอินเทอร์เน็ตนี้ดังภาพที่ 2.2



ภาพที่ 2.2 แสดงเส้นทางการส่งข้อมูลจากการส่งระหว่าง UE และ Internet โดย (a) จาก UE ถึง Internet และ (b) จาก Internet มายัง UE

ที่มา: LTE Network Architecture: Basic เข้าถึง 1 เมษายน 2562 จาก

<http://www.netmanias.com/en/post/techdocs/5904/architecture-lte/lte-network-architecture-basic>

## 2.2 Multipath Transmission Control Protocol : MPTCP<sup>2</sup>

MPTCP เป็นเทคโนโลยี LTE-WiFi Aggregation ที่เป็นการนำ Wi-Fi และ LTE มาทำงานร่วมกันในเพื่อเพิ่มความเร็วในการรับส่งข้อมูล โดยเทคโนโลยีในด้าน LTE-WiFi Aggregation แบ่งออกได้เป็น 2 แบบแบบแรก คือการทำ Aggregation ระดับ Radio Link เช่นเทคโนโลยี LTE-U (Carrier Aggregation) เทคโนโลยี LWA (Link Aggregation) และแบบที่ 2 คือการทำ Aggregation ระดับ TCP นั่นคือ Multipath TCP (Path Aggregation) ในปัจจุบันนี้มีการใช้งานสมาร์ทโฟนเป็นอุปกรณ์หลักในการใช้งานอินเทอร์เน็ตไม่ว่าจะเป็นการแชร์ข้อมูลมีเดียหรือการสตรีมมิ่งวิดีโอซึ่งต้องทั้งใช้ความเร็วอินเทอร์เน็ตที่สูง เพื่อการใช้งานสื่อต่างๆในระดับคุณภาพที่สูงขึ้น

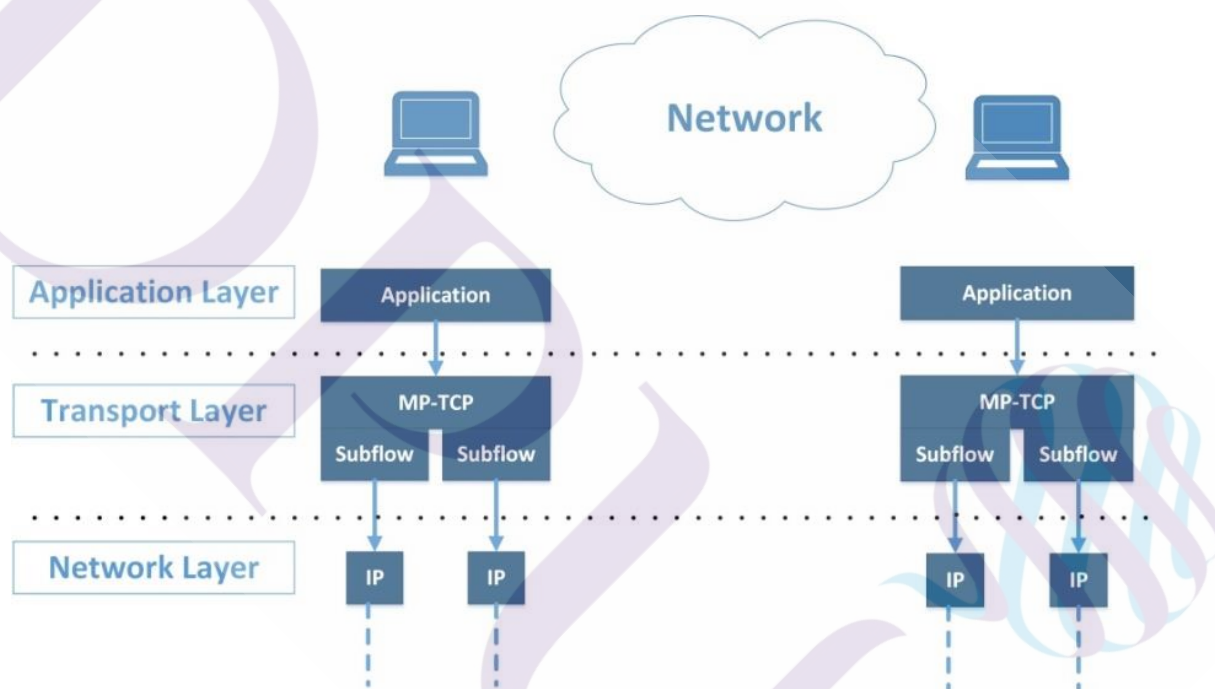
MPTCP คือชุดการทำงานที่เป็นส่วนขยายของ ransmission Control Protocol (TCP) มีหน้าที่รับผิดชอบในสร้างช่องทางในการเชื่อมต่อระหว่างอุปกรณ์ปลายทางที่ต้องการรับส่งข้อมูลซึ่งกันและกัน เมื่ออุปกรณ์ปลายทางสร้าง Connection ระหว่างกัน ซึ่งคล้ายกับวิธี three ways handshake บน TCP ปกติ จะมีการเพิ่ม Option บางอย่างเข้าไปใน Header ของ Packet TCP ที่ชื่อว่า MP\_CAPABLE เข้าไป หาก Server ที่ต้องการสร้างการเชื่อมต่อด้วยสามารถรองรับ Multipath TCP นี้ได้ก็จะตอบกลับมาด้วย Option เดียวกัน จึงเกิดการสร้าง subflow ที่อุปกรณ์ปลายทาง และแลกเปลี่ยนข้อมูลอินเทอร์เน็ตเฟสอื่นๆ เพื่อใช้เป็นช่องทางในการรับส่งข้อมูล นั่นเอง

การใช้งาน MPTCP นั้น ผู้ใช้งาน หรือ Client จะสามารถเชื่อมต่อกับโฮสต์หรือปลายทางเดียวกันได้มากกว่าหนึ่งการเชื่อมต่อหรือผ่านเครือข่ายหลายๆตัว โดยมีการสร้าง Subflow ย่อยๆหลาย Subflows เพื่อแบ่งเป็นหลายๆ IP ในการส่งข้อมูล วิธีนี้จะสร้างการเชื่อมต่อข้อมูลที่มีเสถียรภาพและมีประสิทธิภาพ ระหว่างโฮสต์และผู้ใช้งานด้วยโครงสร้างพื้นฐานของระบบเครือข่ายที่มีอยู่ การทำงานของ MPTCP นั้นมีการทำงานอยู่ในชั้น Transport Layer ซึ่งจากการเชื่อมต่อหลายอินเทอร์เน็ตเฟสหลายๆทางดังวิธีข้างต้นนั้นอาจก่อให้เกิดปัญหาในการใช้งานได้ เนื่องจากในการใช้งานแบบปกติจะสามารถใช้งานได้ทีละอินเทอร์เน็ตเฟส หากเกิดการขาดหายไปของการเชื่อมต่อ จะเกิดขบวนการดำเนินการหาเส้นทางใหม่ของการเชื่อมต่ออีกครั้ง ซึ่งจะต้องใช้เวลาในการหาเส้นทางช่วงระยะเวลาหนึ่ง จากปัญหา

---

<sup>2</sup> Costin Raiciu, Dragos Niculescu, Marcelo Bagnulo, (June 2011) Mark Handley University Politehnica of Bucharest, Universidad Carlos III de Madrid, University College London "Opportunistic Mobility with Multipath TCP" MobiArch '11: Proceedings of the sixth international workshop on MobiArch

ที่เกิดขึ้นนี้จะส่งผลอย่างเห็นได้ชัดกับการใช้งาน Application ที่ใช้งานแบบ RealTime เช่น การสตรีมมิ่งวิดีโอ เป็นต้น ดังนั้นเพื่อการใช้งานอย่างต่อเนื่องโดยไม่เกิดการกระตุกหรือค้าง จะต้องมีการนำ MPTCP เข้ามาใช้เพื่อช่วยเพิ่มความสามารถในการหาเส้นทางหลักและสำรอง ซึ่งจะต้องไม่ก่อให้เกิดปัญหาบน Socket API: Application Programming Interface ด้วยวิธีการสร้าง SubFlow ซึ่งจะแสดงให้เห็นถึงการเชื่อมต่อ, ขั้นตอนการดำเนินงาน และปัจจัยที่นำมาคิดขบวนการในการทำงานของ MPTCP โดยมีการเปรียบเทียบการทำงานระหว่าง TCP กับ MPTCP ดังภาพที่ 2.1 แสดงการทำงานโดยใช้มาตรฐานของ TCP/IP ในชั้น Transport Layer



ภาพที่ 2.3 การส่งข้อมูลดำเนินการของ MPTCP

ที่มา: Balia (Balanced linked adaptation) A new MPTCP congestion control algorithm

เข้าถึง 1 เมษายน 2562 จาก <https://www.ietf.org/proceedings/90/slides/slides-90-mptcp-0.pdf>

### 2.3 การทำงานของ MPTCP<sup>3</sup>

การทำงานของ MPTCP นั้นในการวิจัยนี้ได้ใช้การทำงานแบบ Full-MPTCP ซึ่งระบบจะมีการเชื่อมต่อผ่านอินเทอร์เน็ตเฟสทุกอินเทอร์เน็ตเฟสและทุกอินเทอร์เน็ตเฟสจะมีการเชื่อมต่อตลอดเวลาเพื่อรองรับการทำงานอย่างเต็มรูปแบบ มีการตอบรับในอัตราการส่ง - รับ ของข้อมูลที่ดีที่สุดใน แต่เนื่องจากการใช้งานเต็มประสิทธิภาพ และมีการเชื่อมต่ออินเทอร์เน็ตเฟสตลอดเวลา แม้ในช่วงที่เครื่องโทรศัพท์เคลื่อนที่ที่มีการเชื่อมต่อผ่านโครงข่ายของโทรศัพท์เคลื่อนที่เพียงเส้นทางเดียว MPTCP นี้ก็ยังมีการดำเนินการสร้าง SubFlow ตลอดเวลาเช่นเดิม การทำงานของระบบแบบ Full-MPTCP เช่นนี้จึงก่อให้เกิดปัญหาด้านพลังงานตามมา คือการสูญเสียพลังงาน, แบตเตอรี่หมดเร็วและมีค่าใช้จ่ายในการเชื่อมต่ออินเทอร์เน็ตผ่านโครงข่ายของโทรศัพท์เคลื่อนที่ ตลอดเวลา

การใช้งาน MPTCP นั้น สามารถใช้งานได้ ทั้ง IPv4 และ IPv6 ในระบบปฏิบัติการ(OS) ของแต่ละอุปกรณ์<sup>4</sup> โดยในการทำงานจะมีการสร้างและลบ SubFlow เพื่อบันทึกค่า TCP SubFlow ของแต่ละตัว ในช่วงเริ่มต้นของการเชื่อมต่อนี้จะทำให้เกิดการค้างหรือกระตุกของบริการขึ้นได้ เนื่องจากระบบจะค้นหาอินเทอร์เน็ตเฟสที่สามารถใช้งานในการเชื่อมต่อแล้วดำเนินการถาม SYN ไปหา Server ปลายทางเพื่อเริ่มการเชื่อมต่อ แต่หากมีการสร้าง SubFlow อยู่ ระบบจะไม่ดำเนินการถาม SYN อีกครั้ง เพราะสามารถดึงการควบคุมในการส่งแพ็กเก็ตของการส่งข้อมูลครั้งแรก หรือเพิ่ม ลบ แก้วไข ของ TCP ที่ถูกเก็บไว้ยัง TCP SubFlow มาใช้งานต่อไปได้เลย

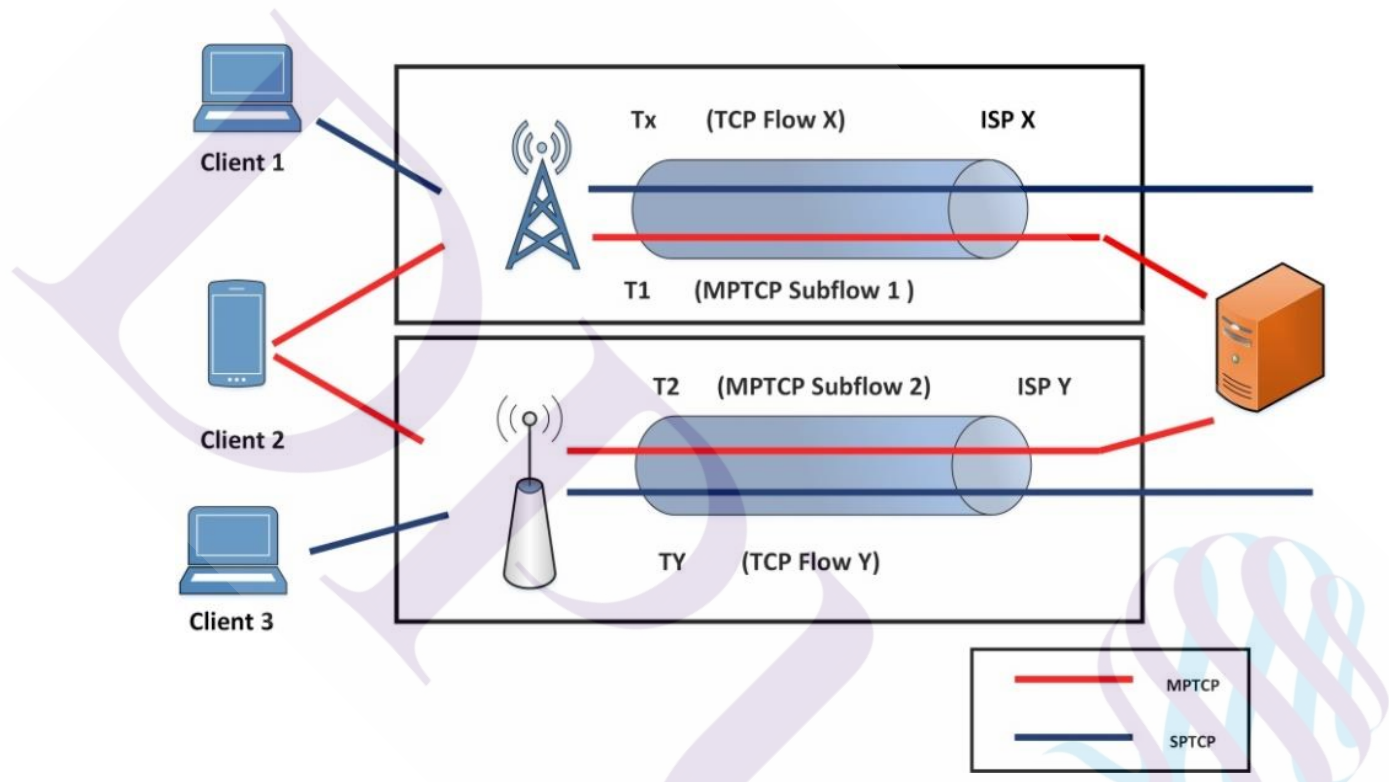
---

<sup>3</sup> Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, Olivier Bonaventure: Université catholique de Louvain, University Politehnica of Bucharest (August 2012) " Exploring Mobile/WiFi Handover with Multipath TCP " CellNet '12: Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design

<sup>4</sup> Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure and Mark Handley: Universitatea Politehnica Bucuresti, ‡Universite Catholique de Louvain Keio University, University College London " How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP " (April 2012) NSDI'12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation



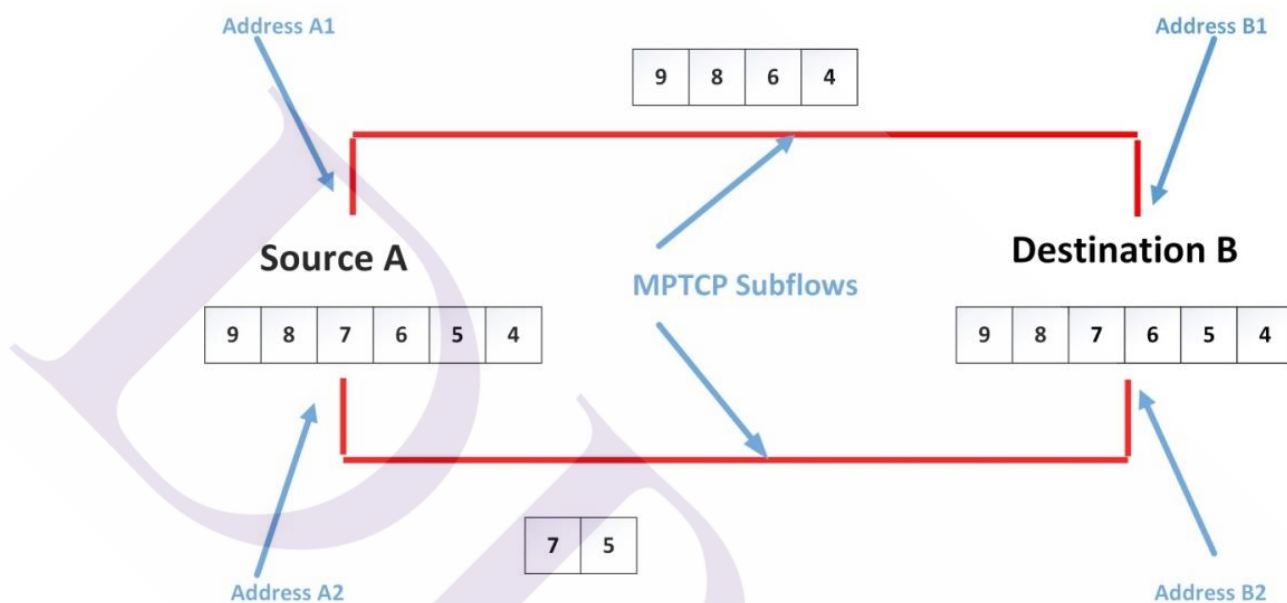
ในการใช้งานระหว่าง SPTCP : Single Path TCP กับ MPTCP : Multipath Path TCP จะเห็นข้อเปรียบเทียบและความแตกต่างในการทำงานในชั้น Transport Layer เนื่องจากการสร้าง subflow และรูปแบบการทำงานของ MPTCP ในการรับและส่งข้อมูลจะเห็นการทำงาน ดังภาพที่ 2.4 ขอเปรียบเทียบในการเชื่อมต่อการทำงานของเชื่อมต่อแบบ SPTCP และ MPTCP ซึ่งแสดงการทำงานและความแตกต่างระหว่าง SPTCP กับ MPTCP ดังภาพที่ 2.4



ภาพที่ 2.4 แสดงการทำงานและความแตกต่างระหว่าง SPTCP กับ MPTCP โดย Client 1, Client 3 มีการเชื่อมต่อแบบ SPTCP คือ มีเชื่อมต่อผ่านอินเทอร์เน็ตเพียงเดียว และ Client 2 มีการเชื่อมต่อแบบ MPTCP คือมีการเชื่อมต่อ มากกว่าหนึ่งอินเทอร์เน็ต โดยการสร้าง SubFlow ของ MPTCP

ที่มา: Multipath TCP speeds up the internet so much that security breaks เข้าถึง 1 เมษายน 2562 จาก [http://www.theregister.co.uk/2014/07/31/multipath\\_tcp\\_will\\_bork\\_your\\_network\\_probes\\_flum\\_mox\\_your\\_firewalls/](http://www.theregister.co.uk/2014/07/31/multipath_tcp_will_bork_your_network_probes_flum_mox_your_firewalls/)

การทำงานของ MPTCP นั้น เมื่อขาดการเชื่อมต่อจะมีการส่งข้อมูล และการนำ packet ที่เหลือที่ยังไม่สามารถส่งต่อไปได้ ดำเนินการส่งต่อไปยังอีกเส้นทางหนึ่งและนำมาจัดเรียงชุดข้อมูลอีกครั้งรอบเพื่อดำเนินการส่งต่อไปดังภาพที่ 2.5



ภาพที่ 2.5 มีการทำงาน MPTCP โดยมีการเชื่อมต่อทั้ง 2 เส้นทาง หากการเชื่อมต่อขาดหายไป เช่นในเส้นทางที่เลข 5 และ 7 ขาดหายไป (ระหว่าง A2-B2) จะถูกนำมาจัดเรียงชุดข้อมูลอีกครั้งรอบเพื่อดำเนินการส่งกลับไปอีกเส้นทาง (ระหว่าง A1-B1) ต่อไป

ที่มา: รู้จักกับ Multipath TCP เข้าถึง 1 เมษายน 2562: <http://www.dpu.ac.th/mect/mptcp.html>

## 2.4 วิธีควบคุมความคับคั่ง<sup>5</sup>

ในการใช้งานโปรโตคอล MPTCP บนอินเทอร์เน็ตแบบไร้สาย เป็นโปรโตคอลที่ช่วยในการเพิ่มประสิทธิภาพและความเร็วของการรับ-ส่งข้อมูลในเวลาเดียวกันบนการเชื่อมต่อหลากหลายเส้นทาง

MPTCP เป็นส่วนเสริมของ TCP ที่มีการอนุญาตให้มีการเชื่อมต่อเพื่อสร้าง TCP-based หลากๆ เส้นทาง ที่เรียกว่า Subflow สำหรับให้บริการการรับ-ส่งข้อมูลแบบ multi-paths ในการรับ-ส่งข้อมูล เมื่อมีปริมาณความหนาแน่นการรับ-ส่งของข้อมูลบนโครงข่ายเป็นจำนวนมากเกินไป จะก่อให้เกิดความคับคั่งของข้อมูล (Congestion) ซึ่งอาจทำให้ใช้เวลาในการรับ-ส่งข้อมูลนานขึ้น หรือ ข้อมูลสูญหายเนื่องจากการ drop packet ได้ การทำงานของ MPTCP เป็นการรับ-ส่งข้อมูลแบบเต็มรูปแบบ (Full-Mesh) ประกอบด้วย การเชื่อมต่อแบบ TCP หลากๆ Subflow บนเส้นทางโครงข่ายที่พร้อมใช้งานที่แตกต่างกันไปยังปลายทางเดียวกัน โดยเริ่มต้นจากการตั้งค่าการเชื่อมต่อ TCP แบบปกติ ระหว่าง ต้นทางและปลายทางผ่านเส้นทางบนโครงข่ายเส้นทางใดเส้นทางหนึ่ง เมื่อเริ่มต้นการเชื่อมต่อแล้ว จะดำเนินการเพิ่มการเชื่อมต่อย่อยเพิ่มเติมโดยการทำ three-way handshake เช่นเดียวกับการดำเนินการเริ่มต้นด้วย TCP เส้นทางใหม่ เพื่อเป็นการเลียนแบบการเชื่อมต่อ TCP แบบปกติ กับ Middleboxes บนเส้นทางของโครงข่าย เพื่อช่วยลดความยุ่งยากในการตั้งค่าและการเชื่อมต่ออินเทอร์เน็ตในความเป็นจริง ทำให้การรับ-ส่งข้อมูลสามารถดำเนินการได้อย่างรวดเร็วและมีประสิทธิภาพดียิ่งขึ้น โปรโตคอล MPTCP นั้น แตกต่างจาก SPTCP (Single Path Transmission Control Protocol) ซึ่งเป็นโปรโตคอลในการรับ-ส่งข้อมูล ที่ไม่ได้รับการสนับสนุนจากอุปกรณ์ Middleboxes ในปัจจุบัน

โปรโตคอล MPTCP มีการใช้ Congestion Control เพื่อปรับอัตราการรับ-ส่ง ข้อมูลในแต่ละเส้นทางที่มีการเปลี่ยนแปลงสถานะเครือข่ายจนถึงสถานะที่ก่อให้เกิดความคับคั่งของข้อมูล โดยดำเนินการปรับ Congestion Windows ภายใน SubFlow เพื่อควบคุมอัตราการรับ-ส่งข้อมูลของแต่ละ

---

<sup>5</sup> Fa Fu, Xing Zhou, Thomas Dreiholz, Keying Wang, Feng Zhou and Quan Gan, "Performance Comparison of Congestion Control Strategies for Multi-Path TCP in the NORNET Testbed", Hainan University, College of Information Science and Technology, Haikou, China, IEEE/CIC ICC 2015 Symposium on Next Generation Networking

Subflow ให้สอดคล้องกับการใช้งานอินเทอร์เน็ตแบบ TCP-friendly โดยเป้าหมายของ Congestion Control มีกฎ 3 ข้อดังนี้

**กฎข้อที่ 1 : (“Improve Throughput”)** การส่งข้อมูลแบบ Multipath Path Flow นั้น จะต้องมีประสิทธิภาพได้อย่างน้อยที่สุดเท่ากับการส่งข้อมูลแบบ Single Path Flow ที่ดีที่สุดในจำนวนเส้นทางทั้งหมดที่มีอยู่

**กฎข้อที่ 2 : (“Do no Harm”)** การส่งข้อมูลแบบ Multipath Path Flow นั้น จะต้องไม่เพิ่ม Capacity ในทรัพยากรที่มีการใช้ข้อมูลร่วมกัน ไปมากกว่า การใช้ทรัพยากรแบบ Single Path Flow ซึ่งการดำเนินการเช่นนี้จะสามารถรับประกันได้ว่า การส่งข้อมูลแบบ Multipath Path Flow จะไม่ส่งผลกระทบต่อเส้นทางการรับ-ส่งข้อมูลอื่นๆ อย่างไม่เหมาะสม

**กฎข้อที่ 3 : (“Balance Congestion”)** การส่งข้อมูลแบบ Multipath Path Flow ควรจะย้ายหรือปรับการรับ-ส่งข้อมูลจากเส้นทางที่เกิดความคับคั่งของข้อมูล (Congestion) ขึ้น ไปยังเส้นทางใหม่ที่ดีกว่าให้ได้มากที่สุด ทั้งนี้การเลือกเส้นทางจะต้องเป็นไปตามกฎข้อที่ 1 และ 2 ด้วยเช่นกัน

การใช้หลักการตามกฎข้อที่ 1 และ 2 รวมกัน จะส่งผลให้เกิด Fairness กับจุดคอขวดในส่วน of กฎข้อที่ 3 สามารถใช้หลักการของการแชร์ทรัพยากรมาใช้ร่วมได้ ในการส่งข้อมูลแบบ Multipath Path Flow เมื่อมีการรับ-ส่งข้อมูลผ่านเส้นทางที่มีความคับคั่งน้อย ทราฟฟิกหรือการรับ-ส่งข้อมูลในเครือข่ายนั้น จะเคลื่อนที่ห่างจาก Congestion Area หรือ พื้นที่ที่มีความคับคั่ง ซึ่งสิ่งนี้จะช่วยเพิ่มประสิทธิภาพของระบบและผลรวมของ Throughput ให้มากขึ้นอีกด้วย

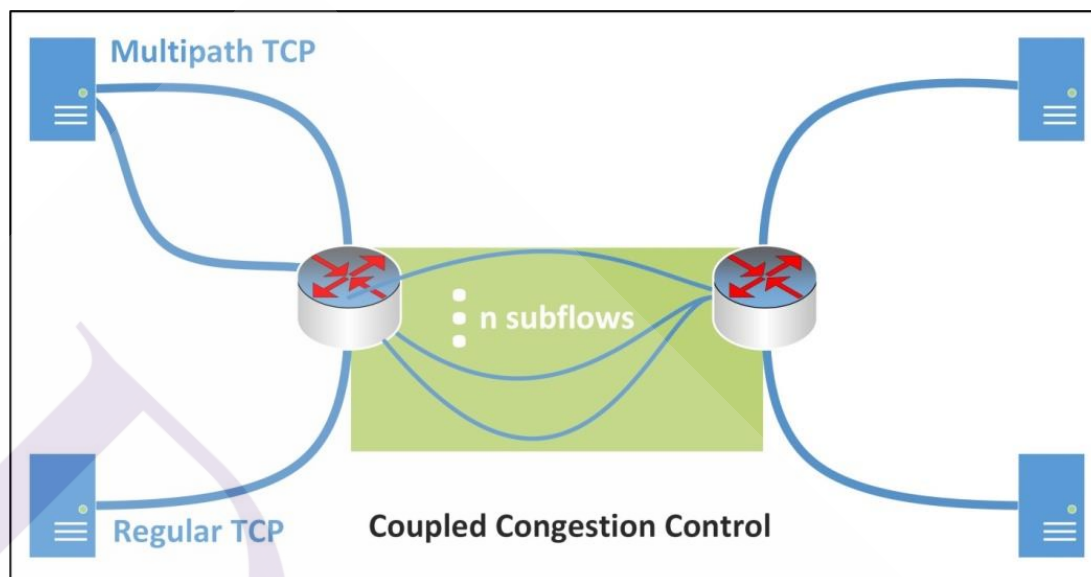
#### 2.4.1 วิธีควบคุมความคับคั่งสำหรับการรับ-ส่งข้อมูลแบบหลายเส้นทาง

วิธีควบคุมความคับคั่งสำหรับการรับ-ส่งข้อมูลแบบหลายเส้นทาง แบ่งเป็น 2 รูปแบบ ดังนี้

1. วิธีควบคุมความคับคั่งแบบ Uncouple คือ รูปแบบที่ง่ายที่สุดของการควบคุมความคับคั่งสำหรับ MPTCP โดยมีการจัดการแต่ละ Subflows เหมือนการเชื่อมต่อแบบอิสระของ TCP ในรูปแบบของ TCP Congestion Control แต่เนื่องจาก Congestion Control ในแบบนี้มีการแชร์ทรัพยากรในจุดคอขวด จึงส่งผลให้ Multi-path Flow เป็นไปอย่างไม่ยุติธรรม ตัวอย่าง Uncouple Congestion Control คือ Vegas, Reno, CUBIC, BIC, H-TCP, Hybla, Westwood เป็นต้น ซึ่งแต่ละรูปแบบมีความแตกต่างกันทั้งข้อดี-ข้อเสีย ที่แตกต่างกันไป ในงานวิจัยนี้ ได้ศึกษาและทดสอบอัลกอริทึมในรูปแบบ ของ CUBIC ซึ่งเป็น Congestion Control ที่พัฒนามาจาก BIC และเป็นตัวอัลกอริทึมพื้นฐานตัวปัจจุบันสำหรับการใช้งาน TCP บน Linux โดยจะมีการใช้หลักการของ Congestion Control คือการกำหนดจำนวน Segment ที่ TCP ส่งได้พร้อมๆ กัน (ลักษณะเดียวกับ Sliding Window) ซึ่งระบบปฏิบัติการจะส่งข้อมูลผ่าน

เครือข่ายได้เป็นจำนวน Segment ไม่เกินขนาดของ Congestion Window (CWND) แล้วกำหนดกระบวนการเพิ่มหรือลดขนาดของของ CWND ให้สอดคล้องกับสภาพความคับคั่งของเครือข่าย สำหรับขนาดที่เล็กที่สุดและเป็นค่าตั้งต้นของ Congestion Window คือ 1 MSS (Maximum Segment Size) ซึ่งการปรับขนาดของ Congestion Windows Size จะขึ้นอยู่กับค่า RTT (Round trip Time) โดย CUBIC จะเพิ่มขนาดของ Congestion Windows Size ในปริมาณมากอย่างรวดเร็วในขณะที่อยู่ห่างจากจุดอิ่มตัว (Saturation Point) แล้วค่อยๆ ช้าลงเมื่ออยู่ใกล้จุดอิ่มตัว คุณลักษณะนี้ช่วยให้ CUBIC ขยายขีดความสามารถของการรับ-ส่งข้อมูลและลดความล่าช้าลงได้ ซึ่งในทางเดียวกันก็สามารถเพิ่มเสถียรภาพ และความเป็นธรรมตามมาตรฐานของ TCP Flow ได้เช่นกัน

2. วิธีควบคุมความคับคั่งแบบ Coupled คือแนวคิดพื้นฐานในการแก้ไขปัญหาความไม่ยุติธรรมของ Uncouple Congestion Control บนการใช้ทรัพยากรร่วมกันบนจุดคอขวด โดยการจับคู่ Congestion Windows ของทุก Supflows บนการเชื่อมต่อของ MPTCP ด้วยหลักการรวมทรัพยากร การตรวจสอบการแอ่งที่จุดคอขวดนั้นเป็นเรื่องที่ยาก ซึ่งสิ่งนี้เป็นส่วนหนึ่งของการจัดการแบนวิธส์ สำหรับ Multi-path เพื่อไม่ให้เกิดการแอ่งที่จุดคอขวดขึ้น หลักการของ Coupled Congestion Control ช่วยให้โปรโตคอลในการขนส่งสามารถปรับเปลี่ยน Congestion Windows ของแต่ละ Subflow และตรวจสอบให้เป็นไปตามมาตรฐานความยุติธรรมและความเป็นธรรมในจุดคอขวดบนเครือข่าย ตัวอย่าง Coupled Congestion Control คือ LIA, OLIA, Balia, wVegas เป็นต้น รูปแบบของ Coupled Congestion Control ใช้กับขั้นตอนการเพิ่มขึ้นหรือเข้าใกล้ของสถานะ Congestion Avoidance และเพื่อระบายขยายตัวเพิ่มขึ้นของ Congestion Windows เมื่อได้รับ Acknowledgement ส่วนในขั้นตอนอื่นๆ ดำเนินการเช่นเดียวกับมาตรฐานของ TCP



ภาพที่ 2.6 แสดง Coupled Congestion Control ของ Client 2 ราย โดย Client แรก ใช้การส่งข้อมูลแบบ MPTCP และอีกราย คือ TCP ปกติ

ที่มา: Multipath TCP เข้าถึง 1 เมษายน 2562 จาก <https://queue.acm.org/detail.cfm?id=2591369>

## 2.5 การมอดูเลชันบนโครงข่ายไร้สาย

### 2.5.1 มอดูเลชันบนโครงข่าย LTE<sup>6</sup>

การส่งสัญญาณโครงข่าย LTE การจาก eNodeB ถึง UE จะต้องมีการทำมอดูเลชันมีการคำนวณ รายละเอียดดังนี้

#### 1. การคำนวณแบนด์วิดท์

ช่องสัญญาณ LTE ที่สามารถใช้งานได้มี 1.4, 3, 5, 10, 15, 20 MHz โดยมีคลื่นความถี่ Number of Resource Block (RB) แบ่งออกเป็น 6, 15, 25, 50, 75 และ 100 ตามลำดับ ซึ่งแบนด์วิดท์มากเท่าไร Throughput ก็จะยิ่งสูงขึ้นเท่านั้น

<sup>6</sup> [http://anisimoff.org/eng/lte\\_throughput.html](http://anisimoff.org/eng/lte_throughput.html)

## 2. การคำนวณคุณภาพของช่องสัญญาณ (Channel quality)

การรับ - ส่งระหว่าง eNodeB ถึง UE มีเงื่อนไขที่ส่งผลกระทบต่ออัตราการส่งข้อมูลและผลลัพธ์ของ throughput นั้นขึ้นกับค่า MCS ซึ่งจะมีค่าแปรผันตามสภาพแวดล้อมและเงื่อนไขในการรับ - ส่ง มาตรฐานในการส่งของช่องส่งสัญญาณและการส่ง CQI (Channel Quality Indicator) เพื่อให้ eNodeB เลือก MCS ดังแสดงในตารางที่ 2.1 และตารางที่ 2.2

ตารางที่ 2.1 ช่อง CQI และ MCS

CQI	Before Rel.12		Rel.12 and beyond	
	Modulation	Code rate	Modulation	Code rate
0	Out of range			
1	QPSK	0.0762	QPSK	0.0762
2	QPSK	0.1172	QPSK	0.1885
3	QPSK	0.1885	QPSK	0.4385
4	QPSK	0.3008	16QAM	0.3691
5	QPSK	0.4385	16QAM	0.4785
6	QPSK	0.5879	16QAM	0.6016
7	16QAM	0.3691	64QAM	0.4551
8	16QAM	0.4785	64QAM	0.5537
9	16QAM	0.6016	64QAM	0.6504
10	64QAM	0.4551	64QAM	0.7539
11	64QAM	0.5537	64QAM	0.8525
12	64QAM	0.6504	256QAM	0.6943
13	64QAM	0.7539	256QAM	0.7783
14	64QAM	0.8525	256QAM	0.8634
15	64QAM	0.9258	256QAM	0.9258

ที่มา: How to calculate LTE throughput เข้าถึง 1 เมษายน 2562 จาก

[http://anisimoff.org/eng/lte\\_throughput.html](http://anisimoff.org/eng/lte_throughput.html)

ตารางที่ 2.2 แสดง MCS บนโครงข่าย LTE

MCS Index	Modulation	TBS Index	MCS Index	Modulation	TBS Index
0	QPSK	0	17	64QAM	15
1		1	18		16
2		2	19		17
3		3	20		18
4		4	21		19
5		5	22		20
6		6	23		21
7		7	24		22
8		8	25		23
9		9	26		24
10	16QAM	10	27	25	
11		11	28	26	
12		12	29	QPSK	Reserved
13		13	30	16QAM	
14		14	31	64QAM	
15		15			
16		16			

**ที่มา:** How to calculate LTE throughput เข้าถึง 1 เมษายน 2562 จาก  
[http://anisimoff.org/eng/lte\\_throughput.html](http://anisimoff.org/eng/lte_throughput.html)

### 3. การคำนวณปริมาณทรัพยากรในโครงข่าย

ทรัพยากรที่มีอยู่จะถูกแบ่งออกให้กับสมาชิกที่ใช้งาน หากมีสมาชิกเพิ่มขึ้น จะต้องมีการแบ่งจัดสรรทรัพยากรให้กับสมาชิกทุกคน นอกจากนั้นยังขึ้นกับลำดับความสำคัญระหว่างผู้ให้บริการกับสมาชิกด้วย



ดังนั้นจะสามารถคำนวณ Throughput ของโครงข่าย LTE ได้ตามขั้นตอนดังนี้

1. กำหนด แบนด์วิดท์ ที่จะใช้งานว่าอยู่ช่วงใด 1.4, 3, 5, 10, 15, 20 MHz และตรงกับ Resource blocks แบบใด

2. เลือกและค้นหาช่องสัญญาณที่ดีที่สุดให้กับการเชื่อมต่อ ซึ่งในการคำนวณต้องทราบว่าใช้งาน MCS ช่องไหน MCS จะขึ้นอยู่กับ CQI โดยความสัมพันธ์ระหว่าง MCS กับ CQI เป็นการกำหนดที่ขึ้นอยู่กับผู้ให้บริการ และ ตารางที่ 2.1 และ 2.2 ต่อไปนี้สามารถนำมาใช้ในการเลือกดัชนี MCS ส่วนช่อง TBS (Transport Block Size) คือจำนวนบิตที่สามารถส่งต่อ 1 TTB = 1 มิลลิวินาที

3. ดูข้อมูลที่อยู่ใน Transport Block Size (TBS) เพื่อตรวจสอบว่ามีการส่งผ่านข้อมูลไปที่บิตต่อ 1 TTB (1 มิลลิวินาที) และทำการคูณด้วย 1,000 เท่ากับ bps (bit per second)

#### 2.5.2 มอดูเลชันบนโครงข่าย WiFi

การส่งสัญญาณบนโครงข่าย WiFi มีหลายปัจจัยในการนำมาคิดเพื่อ รับ - ส่งข้อมูลระหว่าง Access Point (AP) ถึง UE ประกอบด้วยดังนี้

- Spatial Streams (SS) เสาในการ รับ - ส่ง สัญญาณ อยู่ที่ตั้งแต่ 1 - 8
- Guard Interval (GI) ช่วงเวลาระหว่าง AP ถึง UE อยู่ที่ 400, 800 ns
- Channel Bonding (CB) ช่องสัญญาณในการ รับ - ส่ง อยู่ที่ 20, 40, 80, 160 MHz
- Modulation and Coding Scheme (MCS) มีค่าอยู่ในช่วง 0 - 9

ในส่วนของความเร็วกับ MCS และตัวแปรอื่นๆ สามารถดูเปลี่ยนได้ดังตารางที่ 2.3

ตารางที่ 2.3 แสดง MCS บนโครงข่าย WiFi ระหว่าง Type N กับ Type AC<sup>7</sup>

MCS Index - 802.11n and 802.11ac					802.11n		802.11ac					
HT VHT		SS	Modulation	Coding	20MHz		40MHz		80MHz		160MHz	
MCS	MCS				No SGI	SGI	No SGI	SGI	No SGI	SGI	No SGI	SGI
0	0	1	BPSK	1/2	6.5	7.2	13.5	15	29.3	32.5	58.5	65
1	1	1	QPSK	1/2	13	14.4	27	30	58.5	65	117	130
2	2	1	QPSK	3/4	19.5	21.7	40.5	45	87.8	97.5	175.5	195
3	3	1	16-QAM	1/2	26	28.9	54	60	117	130	234	260
4	4	1	16-QAM	3/4	39	43.3	81	90	175.5	195	351	390
5	5	1	64-QAM	5/6	52	57.8	108	120	234	260	468	520
6	6	1	64-QAM	3/4	58.5	65	121.5	135	263.3	292.5	526.5	585
7	7	1	64-QAM	5/6	65	72.2	135	150	292.5	325	585	650
8	1	256-QAM	3/4		78	86.7	162	180	351	390	702	780
9	1	256-QAM	5/6		n/a	n/a	180	200	390	433.3	780	866.7

## 2.6 วิธีควบคุมความคับคั่งแบบ wVegas<sup>8</sup>

เนื่องจากการคำนวณภายใน MPTCP ไม่ได้คำนึงถึงเฉพาะขนาดของช่องสัญญาณการเชื่อมต่อ แต่ยังมี MPTCP Congestion Control เข้ามาเกี่ยวข้องด้วย ซึ่งงานวิจัยนี้เลือกใช้ Congestion control แบบ wVegasสำหรับการควบคุมความคับคั่งแบบ Coupled Congestion Control ซึ่งจะคำนวณเส้นทางในการรับ-ส่งข้อมูลจากค่า Round-Trip Time (RTT) และ Delay ว่าช่องทางใดเหมาะสำหรับการรับ-ส่งข้อมูลมากกว่ากัน ตามสมการดังนี้

$$\begin{aligned}
 DIFF &= \left( \frac{CWND}{BASE_{RTT}} - \frac{CWND}{RTT} \right) BASE_{RTT} - RTT \quad \text{---- (1)} \\
 &= CWND (RTT - BASE_{RTT}) / RTT
 \end{aligned}$$

<sup>7</sup> <http://mcsindex.com>

<sup>8</sup> Wenlan Guo<sup>1</sup>, Zhijia Wang, Yun Zhang, School of Computer Science and Technology, Harbin Univ. Sci. Tech., Harbin, China (2013.31.28) " Delay-based Congestion Control for Multipath TCP "

โดยจาก (1)

DIFF = แพคเกจที่ค้างค้างที่อยู่ในคิวที่จะบอกความยาวของคิว และมีการคำนวณทุกครั้งหลังจากสิ้นสุดของแต่ละรอบ

BASE\_RTT = ค่าต่ำสุดของ RTT ที่วัดได้ ซึ่งจะไม่มีการหน่วงเวลาในการจัดคิวรับ - ส่ง ข้อมูล โดยเอาค่าเฉลี่ยในรอบก่อนหน้า

RTT = เวลาตั้งแต่เริ่มส่งแพ็คเกจข้อมูลจนเสร็จสิ้น

CWND = การควบคุมความคับคั่งของข้อมูล

**ขั้นตอนการทำงาน การควบคุมความแออัดควบคุมของ wVegas**

equilibrium\_rates[r] ทำหน้าที่ในการจัดเก็บอัตราการส่งผ่านของเส้นทาง r ทำการคำนวณน้ำหนักของ sub-flow

queue\_delay[r] ทำหน้าที่ในการจัดเก็บความล่าช้าที่ต่ำสุดของเส้นทาง

alpha[r] ทำหน้าที่ในการจัดเก็บแพคเกจที่คาดหวังว่าจะค้างค้างในเส้นทาง r

โดยการวิเคราะห์อัลกอริทึมที่เฉพาะเจาะจงจะแสดงดังต่อไปนี้

**ระยะเริ่มต้น**

total\_alpha <- fixed number ---- เครือข่ายจะกำหนดค่าคงที่ของการไหลมาของแพคเกจที่ค้างค้างและบันทึกไว้ใน total\_alpha

Equilibrium\_rates[r] <- 0, queue\_delay[r] <- 0, alpha[r] <- random number

Base\_RTT[r] ---- การวัด RTT ต่ำสุดของเส้นทาง โดยไม่มีการจัดคิว

**ขั้นตอนการหลีกเลี่ยงความความแออัดหลังของแต่ละรอบของเส้นทาง**

RTT <- Sampled\_rtts[r]/Sampled\_num[r] ---- การคำนวณค่าเฉลี่ยของ RTT รอบก่อนหน้า

Diff <- cwnd[r] (RTT-Base\_RTT[r]) / RTT---- จำนวนตัวเลขจริงของแพคเกจที่ค้างค้างในเส้นทาง r และก็เป็นความล่าช้าของเข้าคิวทำให้บ่งบอกถึงการขยายตัวของความแออัด

If Diff >= alpha[r] then

Equilibrium\_rates[r] <- cwnd[r]/RTT

```

Adjust weight[r]<-Equilibrium_rates[r]/total_rate
alpha[r]<-weight[r] multiply total_alpha
alpha[r]<-max (alpha[r], 2) end if

```

---- คือการปรับปรุงการส่งข้อมูล อัตราน้ำหนักและ  
 ค่าการณั้แพคเกจที่คั่งค้างของเส้นทาง r

```

If Diff<alpha[r] then cwnd[r] <- cwnd[r]+1
If Diff>alpha[r] then cwnd[r] <- cwnd[r]-1

```

---- คือการปรับปรุงขนาดของหน้าต่างส่ง  
 ตามความสัมพันธ์ระหว่างค่าว่าแพคเกจที่คั่งค้าง  
 และแพคเกจที่คั่งค้างที่เกิดขึ้นจริง

```

q<-RTT - Base_RTT
if queue_delay[r]=0 or queue_delay[r]>q then queue_delay[r] <- q
if q>2 multiply queue_delay[r]
backoff_factor <- 0.5 multiply Base_RTT[r]/RTT
queue_delay[r]<-0

```

---- ค่า RTT จะได้รับจากค่าเฉลี่ยของ RTT รอบที่ผ่านมา แต่ละค่า RTT  
 จะได้รับเมื่อแต่ละแพคเกจในรอบจะถูกส่งต่อไป ถ้าค่า RTT มีขนาดสูงขึ้นจะเห็นถึงค่าความ  
 แตกต่างที่เห็นชัด ดังนั้นค่าต่ำสุดในความล่าช้าของการจัดคิวขั้นต่ำ จะได้รับการบันทึกไว้  
 การควบคุมจะลดความแออัดของเส้นทางเพื่อระบายข้อมูลออกจากแพคเกจที่คั่งค้างต่อไป

```

cwnd[r] <- max (cwnd[r], 2)

```

--- ตรวจสอบว่าเส้นทางไม่ถูกยกเลิกทั้งหมด

**หากเกิดการสูญเสียที่เกิดขึ้นในแพคเกจเส้นทาง r**

equilibrium\_rates[r] <- 0, queue\_delay[r] <- 0 ---- การสูญเสียเกิดขึ้นเมื่อเส้นทางถูกยกเลิก  
 และส่งข้อมูลทั้งหมด จะถูกเลื่อนไปอีกเส้นทาง น้ำหนักของเส้นทางที่ถูกเลือก ยังคงเหมือนเดิม

## 2.7 วิธีควบคุมความคับคั่งแบบ CUBIC<sup>9</sup>

งานวิจัยนี้เลือกใช้ Congestion control แบบ CUBIC<sup>8</sup> สำหรับการควบคุมความคับคั่งแบบ

---

<sup>9</sup> Sangtae Ha, Injong Rhee, Dept of Computer Science North Carolina State University Raleigh, NC 27695 and Lisong Xu, Dept of Comp. Sci. and Eng. University of Nebraska Lincoln, Nebraska 68588, “CUBIC: A New TCP Friendly HighSpeed TCP Variant”

Uncoupled Congestion Control ซึ่งคุณสมบัติที่สำคัญของ CUBIC คือการเติบโตของ window size ซึ่งขึ้นอยู่กับเวลาจริงระหว่าง congestion events สองเหตุการณ์ที่ติดต่อกัน เรียกว่า congestion epoch ดังนั้นการเติบโตของ window size จึงไม่ขึ้นกับ RTT ซึ่งมี window growth function ตามสมการดังนี้

$$W(t) = C(t - K)^3 + W_{max} \quad \text{--- (1)}$$

โดยจาก (1)

C = พารามิเตอร์ CUBIC,

t = เวลาที่ผ่านไปจากการลดหน้าต่างสุดท้าย,

K = ช่วงเวลาที่ฟังก์ชันข้างต้นใช้เพื่อเพิ่ม W เป็น Wmax เมื่อไม่มีเหตุการณ์การสูญเสียเพิ่มเติมและคำนวณโดยใช้สมการดังนี้

$$K = \sqrt[3]{\frac{W_{max}\beta}{C}} \quad \text{--- (2)}$$

และ  $\beta$  เป็นการคูณแบบลดลงคงที่

ขั้นตอนการทำงาน การควบคุมความแออัดควบคุมของ<sup>10</sup>

**SLOW START** ---ปรับเปลี่ยนช่วง Slow start เมื่อเริ่มต้นการใช้งาน (เมื่อ cwnd โตขึ้น)

ระยะเริ่มต้น

last max = 0; --- ค่าเริ่มต้นของค่าสูงสุดที่ผ่านมา

loss cwnd = 0 --- ค่าเริ่มต้นของอัตราการสูญเสียของ congestion window

epoch start = 0, ssthresh = 100,  $\beta = 0.2$ , b = 2.5; C = 0.4

<sup>10</sup> Gulshan Amin Gilkar, Syed Yasmeen Sahdad Department of CSE Department of CSE

### ขั้นตอนการหลีกเลี่ยงความแออัดหลังของแต่ละรอบของเส้นทาง

$\text{delay min} = \min(\text{RTT}, \text{delay min})$  --- ค่าตัวต่ำสุดของค่าความล่าช้าและบันทึกไว้ใน

$\text{delay min}$

if  $\text{cwnd} < \text{ssthresh}$  then  $\text{cwnd}++$  else //slow start

if epoch start = 0 then

epoch start = current time

$K = \max(0, 3p(b - \text{last max} - \text{cwnd}))$

origin point =  $\max(\text{cwnd}, \text{last max})$

$t = \text{current time} + \text{delay min} - \text{epoch start}$  --- จำนวนเวลา  $t$  ที่ผ่านไปจากการลดหน้าต่างสุดท้าย

target = origin point +  $c \cdot (t - K)^3$  --- จำนวนค่า target จากจุด original point

เมื่อได้รับ ACK ในระหว่างการ congestion avoidance, CUBIC จะคำนวณอัตราการขยายตัวของ Window size ในช่วง RTT ถัดไป โดยใช้ สมการ (1) เพื่อกำหนดให้เป็น target ของ congestion window

หากเกิดการสูญเสียที่เกิดขึ้นในแพคเกจเส้นทาง  $r$

epoch start = 0

if  $\text{cwnd} < \text{last max}$  then  $\text{last max} = 0.9 \cdot \text{cwnd}$

else  $\text{last max} = \text{cwnd}$  end if

loss  $\text{cwnd} = \text{cwnd}$

$\text{cwnd} = 0.8 \cdot \text{cwnd}$  // backoff cwnd by 0.8

การสูญเสียเกิดขึ้นเมื่อเส้นทางถูกยกเลิก และส่งข้อมูลทั้งหมด จะถูกเลื่อนไปอีกเส้นทางนำหน้าของเส้นทางที่ถูกเลือก ยังคงเหมือนเดิม

## 2.8 โปรแกรมจำลองโครงข่าย (NS3)<sup>11</sup>

ในการพัฒนาระบบเครือข่ายนั้น จะต้องมีการวางโครงสร้างและระบบ ผ่านการติดตั้งอุปกรณ์ต่างๆ ที่จะต้องใช้ในการพัฒนาเครือข่าย ในการดำเนินการพัฒนาระบบนั้นจำเป็นจะต้องมีการทดสอบก่อนการใช้งานทุกครั้ง ซึ่งเป็นการยากที่จะทดสอบการทำงานผ่านอุปกรณ์ต่างๆเหล่านั้น ทั้งเนื่องด้วยความเสี่ยงด้านการลงทุน งบประมาณ ความผิดพลาดของการวางระบบ เหล่านี้ทำให้เกิดปัญหาต่างๆตามมาอีกมากมาย จึงเกิดความจำเป็นที่จะต้องมีการจำลองเครือข่าย เพื่อทำการศึกษาการทำงานของระบบเครือข่ายในด้านต่างๆ ก่อนการติดตั้งระบบโครงข่ายเพื่อใช้งานจริง ส่งผลให้เกิดโปรแกรมที่ช่วยในการจำลองเครือข่ายขึ้นมากมาย เช่น NS3, Swans, Omnet++ เป็นต้น

การทดสอบอุปกรณ์หรือพัฒนาระบบเครือข่ายในอดีตนั้นจะต้องใช้งบประมาณในการดำเนินการที่เป็นจำนวนมาก ไม่คุ้มต่อการลงทุน และยังเสียเวลาในการติดตั้งอุปกรณ์เหล่านั้นอีก แต่ในปัจจุบันได้นำระบบ NS3 มาทดสอบและยังเป็น open-source สามารถที่จะทดสอบ บนระบบปฏิบัติการ Linux, MAC OS X ฯลฯ การจำลองบน NS3 รองรับการใช้งานทั้งในเครือข่าย IP และ non-IP ซึ่งผู้ใช้งานส่วนใหญ่มุ่งเน้นไปที่การจำลองแบบไร้สาย wireless/IP simulations ซึ่งเกี่ยวข้องกับโมเดลแบบ Wi-Fi, WiMAX หรือ LTE สำหรับเลเยอร์ 1 และ 2 และโปรโตคอลการจัดเส้นทางแบบคงที่หรือไดนามิกเช่น OLSR และ AODV เป็นต้น ซึ่งภาษาที่ใช้งานบน NS3 คือ ภาษา C++ ซึ่งง่ายต่อการเขียนและการทำงานในปัจจุบัน

ตัวอย่างการเขียน NS3 : tcp-pcap-nanosec-example.cc File Reference<sup>12</sup>

Network topology

```
//
// n0 ----- n1
//      500 Kbps
//      5 ms
```

<sup>11</sup> <https://www.nsnam.org/doxygen/index.html>

<sup>12</sup> [https://www.nsnam.org/doxygen/dir\\_600cab24fe22a606254421806d2e08fa.html](https://www.nsnam.org/doxygen/dir_600cab24fe22a606254421806d2e08fa.html)

โดยกำหนดให้ จาก n0 to n1 ใช้ BulkSendApplication.

และ Tracing ของ queues และ packet ร่องรับด้วยไฟล์ "tcp-pcap-nanosec-example.pcap"

เมื่อ tracing มีการ turned on จะทำให้ Trace file timestamps ถูกบันทึกใน nanoseconds

เริ่มต้นจากการประกาศ module

```
#include <string>
#include <fstream>
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/internet-module.h"
#include "ns3/applications-module.h"
#include "ns3/network-module.h"
```

ประกาศ namespace และ ค่าตัวแปรเริ่มต้น

```
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("TcpPcapNanosecExample");
int main (int argc, char *argv[])
{
    bool tracing = false;
    bool nanosec = false;
    uint32_t maxBytes = 0;
```

ดำเนินการ run-time ผ่าน command-line arguments

```
CommandLine cmd;
cmd.AddValue ("tracing", "Flag to enable tracing", tracing);
cmd.AddValue ("nanosec", "Flag to use nanosecond timestamps for pcap as default", nanosec);
cmd.AddValue ("maxBytes",
    "Total number of bytes for application to send", maxBytes); cmd.Parse (argc, argv);
```



ตรวจสอบการเรียกใช้ nanosec cmdline flag, เพื่อสร้าง nanosecond timestamp สำหรับ pcap traces

```
if (nanosec)
{
    Config::SetDefault ("ns3::PcapFileWrapper::NanosecMode", BooleanValue (true));
}
```

สร้างโหนดเพื่อการใช้งาน (กำหนดไว้ 2 nodes)

```
NS_LOG_INFO ("Create nodes.");
NodeContainer nodes;
nodes.Create (2);
NS_LOG_INFO ("Create channels.");
```

สร้าง point-to-point link ด้วย topology ข้างต้น

```
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("500Kbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("5ms"));
```

```
NetDeviceContainer devices;
devices = pointToPoint.Install (nodes);
```

ดำเนินการ Install the internet stack บน the nodes

```
InternetStackHelper internet;
internet.Install (nodes);
```

ดำเนินการเพิ่ม IP addresses.

```
NS_LOG_INFO ("Assign IP Addresses.");
Ipv4AddressHelper ipv4;
```

```

ipv4.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign (devices);
NS_LOG_INFO ("Create Applications.");

```

สร้าง a BulkSendApplication และ install บน node 0

```

uint16_t port = 9; // well-known echo port number
BulkSendHelper source ("ns3::TcpSocketFactory",
                       InetSocketAddress (i.GetAddress (1), port));

```

ตั้งค่าปริมาณข้อมูลที่จะส่งเป็น bytes

```

source.SetAttribute ("MaxBytes", UintegerValue (maxBytes));
ApplicationContainer sourceApps = source.Install (nodes.Get (0));
sourceApps.Start (Seconds (0.0));
sourceApps.Stop (Seconds (10.0));

```

สร้าง a PacketSinkApplication และ install บน node 1

```

PacketSinkHelper sink ("ns3::TcpSocketFactory",
                       InetSocketAddress (Ipv4Address::GetAny (), port));
ApplicationContainer sinkApps = sink.Install (nodes.Get (1));
sinkApps.Start (Seconds (0.0));
sinkApps.Stop (Seconds (10.0));

```

ตั้งค่า tracing เมื่อถูก enabled

```

if (tracing)
{
    AsciiTraceHelper ascii;
    pointToPoint.EnablePcapAll ("tcp-pcap-nanosec-example", false);
}

```

ดำเนินการ run และ clear บน simulation

```

NS_LOG_INFO ("Run Simulation.");
Simulator::Stop (Seconds (10.0));

Simulator::Run ();

Simulator::Destroy ();

NS_LOG_INFO ("Done.");

Ptr<PacketSink> sink1 = DynamicCast<PacketSink> (sinkApps.Get (0));

std::cout << "Total Bytes Received: " << sink1->GetTotalRx () << std::endl;
}

```

## 2.9 งานวิจัยที่เกี่ยวข้อง

### 2.9.1 Performance Comparison of Congestion Control Strategies for Multi-Path TCP in the NORNET Testbed

ในบทความนี้เปรียบเทียบประสิทธิภาพของ TCP Congestion Control Multi-Path บน Fiber-Based Internet Connections และ ADSL-Based Internet Connections TCP และ เปรียบเทียบการทำงานของ Congestion Controls for MPTCP ในแบบต่างๆ ของ Uncoupled และ Coupled โดยการจำลองจากการทำงานบนโครงข่าย Internet Service Providers (ISP) จริง โดยสามารถรองรับการทำงานได้ดี

สรุปงานวิจัยมีข้อดีดังนี้

1) มีการทดสอบ MPTCP Congestion Control ที่หลากหลาย ทั้งแบบ Uncoupled และแบบ Coupled

2) เปรียบเทียบประสิทธิภาพของ TCP Congestion Control บน Fiber-Based Internet Connections และ และ ADSL-Based Internet Connections และ บนอุปกรณ์จริง

ข้อจำกัด

1) ไม่มีการทดสอบ MPTCP ในการใช้งานโครงข่าย LTE/ WiFi

### 2.9.2 Delay-based Congestion Control for Multipath TCP<sup>13</sup>

ในบทความนี้ทดสอบการใช้งาน MPTCP บน MATLAB ในโครงข่าย LTE และ WiFi โดยเพิ่มการใช้งานการควบคุมความคับคั่งของข้อมูลแบบ wVegas โดยการจำลองจากโปรแกรม NS3

สรุปงานวิจัยมีข้อดีดังนี้

1) มีการทดลองที่ชัดเจนกับการใช้งานจริงด้วย network topologies ที่มีการใช้ทรัพยากรร่วมกัน

2) มีการเปรียบเทียบ Congestion Control (แบบ Coupled) ใน 2 รูปแบบ คือ delay-based (wVegas) และ loss-based (Linked Increases)

ข้อจำกัด

1) ไม่มีการเปรียบเทียบการแออัดของข้อมูล ระหว่าง MPTCP Congestion Control แบบ Uncoupled และ Coupled

2) ไม่มีการทดสอบ MPTCP ในการใช้งานโครงข่าย LTE และ WiFi

### 2.9.3 Performance Evaluation of MPTCP Protocol over LTE and WiFi Network<sup>14</sup>

ในบทความนี้ เปรียบเทียบการทำงานของ MPTCP กับ TCP บนโครงข่าย LTE และ Wifi โดยคำนึงถึงการควบคุมความคับคั่งของข้อมูล และเปรียบเทียบจากค่า Throughput, อัตราล่าช้าต่อหน่วยเวลา (Delay), อัตราการสูญหายของข้อมูล (Packet Loss Rate) และค่าความเบี่ยงเบนของการหน่วงเวลา (Jitter) โดยสามารถรองรับการทำงานได้ดี

สรุปงานวิจัยมีข้อดีดังนี้

1) มีการทดสอบ MPTCP ในการใช้งานโครงข่าย LTE และ Wifi

2) มีการเปรียบเทียบการทำงานของ MPTCP กับ TCP บนโครงข่าย LTE และ WiFi โดยคำนึงถึงการควบคุมความคับคั่งของข้อมูล (wVegas)

<sup>13</sup> Yu Cao\*t, Mingwei Xu\*+, Xiaoming Fu\$, \*Tsinghua University, tNational Data Switch Center, China Tsinghua National Laboratory for Information Science and Technology (TNList), Institute of Computer Science, University of Goettingen

<sup>14</sup> ภาณุวง เมฆไพบุลย์ และธนัญ จารุวิทย์โกวิท. (2559) “Performance Evaluation of MPTCP Protocol over LTE and WiFi Network” การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้ามหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่ 8 (น.) ภูเก็ต: มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์.

### ข้อจำกัด

1) ไม่มีการเปรียบเทียบการแออัดของข้อมูล ระหว่าง MPTCP Congestion Control แบบ Uncoupled และ Coupled

2.9.4 A First Study on Using MPTCP to Reduce Latency for Cloud Based Mobile Applications<sup>15</sup>

ในบทความนี้ เปรียบเทียบ Packet Latency ของการทำงานของ MPTCP กับ TCP ตามการใช้งานแอปพลิเคชันที่มีการใช้ความหนาแน่นของกราฟฟิคต่างกัน ในระดับ สูง, กลาง และ ต่ำ บน โครนข่าย

สรุปงานวิจัยมีข้อดีดังนี้

- 1) มีการทดสอบ MPTCP ในการใช้งาน โครนข่าย LTE
- 2) มีการเปรียบเทียบ Packet Latency ของการทำงานของ MPTCP กับ TCP ตามการใช้งานแอปพลิเคชันที่มีการใช้ความหนาแน่นของกราฟฟิคต่างกัน โดยคำนึงถึงการควบคุมความคับคั่งของข้อมูล (LIA)

### ข้อจำกัด

- 1) ไม่มีการทดสอบ MPTCP ในการใช้งาน โครนข่าย WiFi
- 2) ไม่มีการเปรียบเทียบการแออัดของข้อมูล ระหว่าง MPTCP Congestion Control แบบ Uncoupled และ Coupled

---

<sup>15</sup> Karl-Johan Grinnemo and Anna Brunstrom, 2015 7th IEEE International Workshop on Performance Evaluation of Communications in Distributed Systems and Web based Service Architectures

ตารางที่ 2.4 เปรียบเทียบคุณสมบัติงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ

คุณสมบัติ	Performance Comparison of Congestion Control Strategies for Multi-Path TCP in the NORNET Testbed	Delay-based Congestion Control for Multipath TCP	Performance Evaluation of MPTCP Protocol over LTE and WiFi Network	A First Study on Using MPTCP to Reduce Latency for Cloud Based Mobile Applications	งานวิจัยที่นำเสนอ
1. มีการทดสอบจากการใช้งานจริง	✓	✗	✗	✓	✗
2. พารามิเตอร์ในการเปรียบเทียบคุณภาพของ Congestion Control	Throughput	Packet Loss, Packet Queuing Delay	Throughput, Delay, Packet Loss	Packet Latency, Packet Loss	Throughput, Delay, Jitter
3. เครื่องมือในการทดสอบ	NORNET Testbed	NS3	NS3	Linux kernel	NS3
4. โครงข่ายในการทดสอบ	Fiber-Based และ ADSL-Based	✗	LTE, WiFi	LTE	LTE, WiFi
5. MPTCP Congestion Control ที่ใช้ทดสอบ	BALIA, CUBIC, LIA, OLIA, Reno, wVegas	Linked Increases, wVegas	wVegas	LIA	wVegas, CUBIC
6. การเปรียบเทียบ MPTCP Congestion Control ระหว่าง Uncoupled/ Coupled	✗	✗	✗	✗	✓

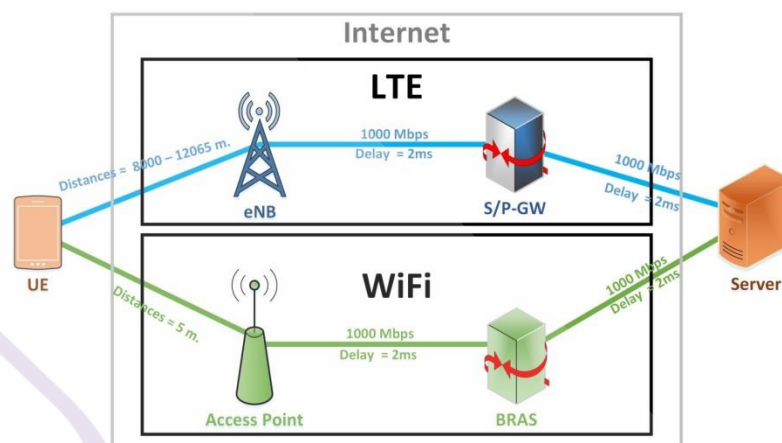
## บทที่ 3

### ระเบียบวิธีวิจัย

ในบทนี้จะกล่าวถึง ขั้นตอนการจำลองโครงข่าย LTE / WiFi เพื่อรองรับการทำงานของ MPTCP (Multiple TCP) พร้อมทั้งอธิบายแนวทางการวิจัยและพัฒนา เครื่องมือที่ใช้ในงานวิจัย แผนการดำเนินงาน ขั้นตอนและวิธีการดำเนินงาน

ปัจจุบันมีการใช้งานระบบอินเทอร์เน็ตโดยผ่านระบบโครงข่าย LTE และ WiFi โดยมีลักษณะการใช้งานแบบสลับโครงข่ายไปมาตลอดเวลา ส่งผลให้ Application ที่ใช้งานมีการรับ-ส่งข้อมูลแบบไม่ต่อเนื่อง วิทยานิพนธ์ฉบับนี้จึงมีแนวความคิดที่จะนำโปรโตคอล MPTCP มาประยุกต์ใช้งานร่วมกันระหว่างโครงข่าย LTE และ WiFi เพื่อช่วยปรับปรุงการเชื่อมต่อดังกล่าวไม่ให้เกิดการขาดหาย และช่วยเพิ่มความเร็วอินเทอร์เน็ตและการทำงานของสูงขึ้นไป แต่กระนั้นยังคงต้องคำนึงถึงข้อจำกัดอีกอย่างคือ เมื่อมีการใช้โปรโตคอล MPTCP ส่งข้อมูลจากต้นทางไปยังปลายทาง ในกรณีที่มี Throughput ต่างกันมากๆ ความต่างของ Throughput ทั้ง 2 ฝั่งที่นี้อาจส่งผลให้ Throughput MPTCP นั้นต่ำกว่า Throughput TCP ในฝั่งมาก โดยในบทความนี้จะจำลองระบบโครงข่าย โดยใช้โปรแกรม Network Simulator Version 3 (NS3) ในการเปรียบเทียบประสิทธิภาพการใช้งาน โปรโตคอล TCP และ โปรโตคอล MPTCP ที่มีการใช้งานผ่านทั้งโครงข่าย LTE / WiFi ด้วยการใส่ Congestion Control แบบ Uncoupled (CUBIC) และ Coupled (wVegas) เพื่อหาขอบเขตและข้อจำกัดข้างต้นต่อไป

#### 3.1 ขั้นตอนการออกแบบการทดสอบบนโครงข่าย LTE และ WiFi



ภาพที่ 3.1 โครงข่าย 4G LTE และ WiFi ที่ใช้ในงานวิจัย

จากภาพที่ 3.1 แสดงให้เห็นถึงโครงข่าย LTE / WiFi ที่ใช้ในการทดสอบในงานวิจัยนี้จะประกอบไปด้วย 2 โครงข่ายเชื่อมต่อกัน ซึ่งมีรายละเอียดดังต่อไปนี้

3.1.1 โครงข่าย LTE (Long Term Evolution) จะประกอบไปด้วยโทรศัพท์เคลื่อนที่ของผู้ใช้งาน (User Equipment – UE) เชื่อมต่อโครงข่ายผู้ให้บริการ โดยโทรศัพท์เคลื่อนที่ของผู้ใช้งานนั้นจะมีการเปลี่ยนแปลงตามรูปแบบการจำลองทางโครงข่าย LTE ตามรูปที่ 3.1 โดย UE สามารถที่จะใช้บริการผ่านโครงข่าย LTE เพื่อรับ-ส่งข้อมูลไปกับอุปกรณ์ปลายทางที่อยู่โครงข่ายอินเทอร์เน็ต

3.1.2 โครงข่าย WiFi (Wireless Fidelity) จะประกอบไปด้วยอุปกรณ์กระจายสัญญาณ (Access Point – AP) ซึ่งเชื่อมต่อกับ UE โดยการเชื่อมต่อของผู้ใช้งานนั้นจะเปลี่ยนแปลงระยะตามรูปแบบการจำลองการทางโครงข่าย WiFi ตามรูปที่ 3.1 โดย UE สามารถที่จะใช้บริการผ่านโครงข่าย WiFi เพื่อรับ-ส่งข้อมูลกับอุปกรณ์ปลายทางที่อยู่โครงข่ายอินเทอร์เน็ต

การเชื่อมต่อระหว่างโครงข่ายทั้ง 2 นั้นจะเป็นการเชื่อมต่อโดยใช้โปรโตคอลของ TCP และ MPTCP ซึ่งจะคำนึงถึงการเชื่อมต่อในการให้บริการ โดยไม่ส่งผลกระทบต่อการใช้งานของ Application และไม่มีการเคลื่อนที่ของโทรศัพท์เคลื่อนที่ของผู้ใช้งาน รวมถึงเครื่องแม่ข่ายและอุปกรณ์ภายในโครงข่าย LTE /WiFi



### 3.2 ค่าตัวแปรของโครงข่าย LTE และ WiFi

ในการออกแบบโครงข่าย LTE / WiFi นั้นจะต้องมีการคำนึงถึงค่าตัวแปรที่ใช้ในการทดลองการทำงานของโครงข่าย มีรายละเอียดดังต่อไปนี้

3.2.1 ศึกษาการใช้งาน บนโครงข่าย LTE โดยใช้ NS 3 เป็นซอฟต์แวร์ในการจำลองการทำงาน โดยกำหนด ค่า LTE แบบ SISO, Bandwidth (20MHz)

ตารางที่ 3.1 ขนาดความจุของการเชื่อมต่อ และความล่าช้าระหว่างอุปกรณ์ในโครงข่าย LTE

การเชื่อมต่อ	ความจุของการเชื่อมต่อ (Mbps)	ค่าความล่าช้าทางเวลา (msec)
1. โทรศัพท์เคลื่อนที่ของผู้ใช้งานเชื่อมต่อกับอุปกรณ์สถานีฐาน (eNB)	100	แปรผันตามระยะทาง
2. อุปกรณ์สถานีฐานเชื่อมต่อกับอุปกรณ์ S/P-GW	1000	2
3. อุปกรณ์ S/P-GW เชื่อมต่อกับเครื่องแม่ข่าย	1000	2

3.2.2 ศึกษาการใช้งานบนโครงข่าย WiFi โดยใช้ NS 3 เป็นซอฟต์แวร์ในการจำลองการทำงาน และเลือกใช้ Wifi มาตรฐาน 802.11ac โดยกำหนด ค่า WiFi แบบ MIMO, Spatial Streams (SS) = 1, Channel bonding = 80MHz, Guard Interval (GI) = 400ns, Modulation and Coding Scheme (MCS) = 0-9

ตารางที่ 3.2 ขนาดความจุของการเชื่อมต่อ และความล่าช้าระหว่างอุปกรณ์ในโครงข่าย WiFi

การเชื่อมต่อ	ความจุของการเชื่อมต่อ (Mbps)	ค่าความล่าช้าทางเวลา (msec)
1. โทรศัพท์เคลื่อนที่ของผู้ใช้งานเชื่อมต่อกับอุปกรณ์สถานีฐาน (Access Point)	แปรผันตามค่า mcs	แปรผันตามระยะทาง

2.อุปกรณ์สถานีฐานเชื่อมต่อกับอุปกรณ์ BRAS	1000	2
3.อุปกรณ์ BRAS เชื่อมต่อกับเครื่องแม่ข่าย	1000	2

3.2.3 ศึกษาการใช้งานบนโครงข่าย เพื่อที่จะประเมินและวิเคราะห์ประสิทธิภาพการใช้งาน โพรโทคอล MPTCP และ TCP บนโครงข่าย LTE และ WiFi ด้วยการ ใช้ Congestion Control แบบ Uncoupled (CUBIC) และ Coupled (wVegas)

3.2.4 รูปแบบการส่งข้อมูลในโครงข่ายและการเชื่อมต่อระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐาน มีการทดสอบแบ่งเป็น 6 ประเภท ดังนี้

3.2.4.1. มีการทดสอบการรับ-ส่งข้อมูลขนาด 229.4 MB บนโครงข่าย WiFi เพียงเส้นทางเดียว โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (Access Point) เป็นระยะทาง 5 เมตร, ขนาด Bufsizes เท่ากับ 1048576 Bytes และกำหนดค่า MCS (Modulation and Coding Scheme) ของ WiFi ในช่วง 0 - 9 โดยใช้วิธีควบคุมความคับคั่งแบบ CUBIC โดยมีรายละเอียดของข้อมูลที่ใช้ทดสอบเป็นไปตามตาราง 3.3

ตารางที่ 3.3 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย WiFi โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานเป็นระยะเดียว โดยใช้วิธีควบคุมความคับคั่งแบบ CUBIC

ประเภทของข้อมูล (Protocol)	ขนาดไฟล์	ระยะห่างระหว่าง สถานีฐาน	ขนาด Bufsizes (Bytes)	วิธีควบคุม ความคับคั่ง	MCS
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	0
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	1
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	2
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	3
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	4
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	5
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	6
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	7

TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	8
TCP (data)	229.4 MB	5 เมตร	1048576	CUBIC	9

3.2.4.2 มีการทดสอบการรับ-ส่งข้อมูลขนาด 229.4 MB บนโครงข่าย LTE เพียงเส้นทางเดียว โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (eNB) เป็นระยะทางในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ CUBIC โดยมีรายละเอียดของข้อมูลที่ใช้ทดสอบเป็นไปตามตาราง 3.4

**ตารางที่ 3.4** ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย LTE โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ CUBIC

ประเภทของข้อมูล (Protocol)	ขนาดไฟล์	ระยะห่างระหว่างสถานี ฐาน	วิธีควบคุมความคับคั่ง
TCP (data)	229.4 MB	8000 - 12065 เมตร	CUBIC

ในการทดสอบที่ 3.2.4.1, 3.2.4.2 เป็นการทดสอบเพื่อหา Throughput ของ TCP ที่มีอัตราการรับ - ส่งข้อมูล TCP มากสุด ผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว และหาผลต่างในอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE โดยผลต่างดังกล่าว จะถูกนำมาวิเคราะห์ในรูปแบบของจำนวนเท่า โดยเลือกอัตราที่จำนวน 1 ถึง 9 เท่า

3.2.4.3 มีการทดสอบการรับ-ส่งข้อมูลขนาด 229.4 MB บนโครงข่าย MPTCP (ซึ่งมีการรับ-ส่งข้อมูลบนโครงข่าย WiFi และ LTE พร้อมกันทั้ง 2 เส้นทาง) โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (Access Point) บนโครงข่าย WiFi และระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (eNB) เท่ากับการทดสอบที่ 3.2.4.1 และ 3.2.4.2 โดยใช้วิธีควบคุมความคับคั่งแบบ CUBIC โดยมีรายละเอียดของข้อมูลที่ใช้ทดสอบเป็นไปตามตาราง 3.5

**ตารางที่ 3.5** ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย MPTCP โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานบนโครงข่าย WiFi เป็นระยะเดียว และ โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานบนโครงข่าย LTE ในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ CUBIC

ประเภทของข้อมูล (Protocol)	ระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และ สถานีฐาน (Access Point)	MCS	ระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และ สถานีฐาน (eNB)	วิธีควบคุมความคับคั่ง
MPTCP	5 เมตร	0	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	1	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	2	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	3	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	4	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	5	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	6	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	7	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	8	8000 - 12065 เมตร	CUBIC
MPTCP	5 เมตร	9	8000 - 12065 เมตร	CUBIC

เพื่อเปรียบเทียบความแตกต่างระหว่าง Throughput ของ TCP ที่มีอัตราการรับ - ส่งข้อมูล TCP มากสุดในโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว เทียบกับค่า Throughput ของ MPTCP ที่มีการรับ-ส่งข้อมูลบนโครงข่าย WiFi และ LTE พร้อมกัน โดยใช้พารามิเตอร์เดียวกับ 3.2.4.1 และ 3.2.4.2

3.2.4.4. มีการทดสอบการรับ-ส่งข้อมูลขนาด 229.4 MB บนโครงข่าย WiFi เพียงเส้นทางเดียว โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (Access Point) เป็นระยะทาง 5 เมตร, ขนาด Bufsizes เท่ากับ 1048576 Bytes และกำหนดค่า MCS(Modulation and Coding Scheme) ของ WiFi ในช่วง 0 - 9 โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas โดยมีรายละเอียดของข้อมูลที่ใช้ทดสอบเป็นไปตามตาราง 3.6

ตารางที่ 3.6 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย WiFi โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน และสถานีฐานเป็นระยะเดียว โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas

ประเภทของข้อมูล (Protocol)	ขนาดไฟล์	ระยะห่างระหว่าง สถานีฐาน	ขนาด Bufsizes (Bytes)	วิธีควบคุม ความคับคั่ง	MCS
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	0
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	1
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	2
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	3
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	4
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	5
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	6
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	7
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	8
TCP (data)	229.4 MB	5 เมตร	1048576	wVegas	9

3.2.4.5 มีการทดสอบการรับ-ส่งข้อมูลขนาด 229.4 MB บนโครงข่าย LTE เพียงเส้นทางเดียว โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (Access Point) เป็นระยะทางในช่วงคือ 8000 ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas โดยมีรายละเอียดของข้อมูลที่ใช้ทดสอบเป็นไปตามตาราง 3.7

ตารางที่ 3.7 ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย LTE โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน และสถานีฐานในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas

ประเภทของข้อมูล (Protocol)	ขนาดไฟล์	ระยะห่างระหว่างสถานี ฐาน	วิธีควบคุมความคับคั่ง
TCP (data)	229.4 MB	8000 - 12065 เมตร	wVegas

ในการทดสอบที่ 3.2.4.4, 3.2.4.5 เป็นการทดสอบเพื่อหา Throughput ของ TCP ที่มีอัตราการรับ - ส่งข้อมูล TCP มากสุด ผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว และในการทดสอบเพื่อการเปรียบเทียบนั้น ผู้วิจัยจะเลือกใช้ชุดพารามิเตอร์เดียวกับแบบ CUBIC เพื่อนำมาวิเคราะห์ในการเปรียบเทียบกับ การทดสอบที่ 3.2.4.6 ต่อไป

3.2.4.6 มีการทดสอบการรับ-ส่งข้อมูลขนาด 229.4 MB บนโครงข่าย MPTCP (ซึ่งมีการรับ-ส่งข้อมูลบนโครงข่าย WiFi และ LTE พร้อมกันทั้ง 2 เส้นทาง) โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (Access Point) และระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (Access Point) เท่ากับการทดสอบที่ 3.2.4.5 และ 3.2.4.6 โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas โดยมีรายละเอียดของข้อมูลที่ใช้ทดสอบเป็นไปตามตาราง 3.8

**ตารางที่ 3.8** ข้อมูลที่ใช้ในการทดสอบบนโครงข่าย MPTCP โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานบนโครงข่าย WiFi เป็นระยะเดียว และ โดยมีระยะห่างระหว่างอุปกรณ์ของผู้ใช้งานและสถานีฐานบนโครงข่าย LTE ในช่วงคือ 8000 เมตร ถึง 12065 เมตร โดยใช้วิธีควบคุมความคับคั่งแบบ wVegas

ประเภทของข้อมูล (Protocol)	ระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (Access Point)	MCS	ระยะห่างระหว่างอุปกรณ์ของผู้ใช้งาน (UE) และสถานีฐาน (eNB)	วิธีควบคุมความคับคั่ง
MPTCP	5 เมตร	0	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	1	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	2	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	3	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	4	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	5	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	6	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	7	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	8	8000 - 12065 เมตร	wVegas
MPTCP	5 เมตร	9	8000 - 12065 เมตร	wVegas

### 3.3 แผนการดำเนินงาน

#### 3.3.1 ศึกษา ค้นคว้า และรวบรวมงานวิจัยที่เกี่ยวข้องกับ MPTCP

ศึกษาทฤษฎีและรูปแบบของ TCP และ MPTCP ที่มี Congestion Control แบบ Uncoupled (CUBIC) และ Coupled (wVegas) สำหรับการจำลองระบบและ เพื่อการเปรียบเทียบประสิทธิภาพระหว่างการทำงานของโปรโตคอล

#### 3.3.2 ศึกษาอุปกรณ์และ โปรแกรมที่ใช้ทำการจำลองระบบ

ศึกษาค้นหาและรวบรวมโปรแกรมที่จะทำการจำลองระบบเพื่อที่จะสามารถประยุกต์ใช้ในการจำลองระบบของ MPTCP บนระบบโครงข่าย เพื่อการนำเสนอได้อย่างถูกต้อง ซึ่งงานวิจัยนี้ใช้โปรแกรม Network Simulator 3 (NS3) ซึ่งเป็น Open Source (Source ที่ให้ใช้งานได้โดยไม่มีค่าใช้จ่าย)

#### 3.3.3 ศึกษาหลักการงานหลักการทำงานของ Multipath TCP

เพื่อค้นหาและทดสอบโครงข่ายมีประสิทธิภาพที่สามารถใช้งาน MPTCP ได้

#### 3.3.4 จำลองระบบที่จะทดสอบ

จำลองระบบโดยใช้โปรแกรม NS 3 ที่ศึกษาจากข้างต้น พร้อมทั้งรวบรวมข้อมูลในการออกแบบ ในการทดสอบว่าโครงข่ายมีประสิทธิภาพและสามารถใช้งาน MPTCP ได้อย่างถูกต้อง

#### 3.3.5 เปรียบเทียบ วิเคราะห์ผลที่ได้ และสรุป

เมื่อทำการจำลองระบบที่ทำการนำเสนอเสร็จแล้ว และทำการวิเคราะห์เปรียบเทียบถึงประสิทธิภาพของระบบที่ทำการนำเสนอ เพื่อที่จะสรุปผลการจำลองระบบว่าประสิทธิภาพของระบบนั้นเป็นอย่างไร มีขอบเขตหรือขีดจำกัดเป็นเช่นไร

#### 3.3.6 รวบรวมข้อมูลที่ได้ทั้งหมดจัดทำวิทยานิพนธ์

ทำการรวบรวมข้อมูลของระบบที่ทำการนำเสนอ ที่ได้ทั้งหมดเพื่อจัดทำเป็นวิทยานิพนธ์ แผนการดำเนินงานที่ได้วางแผนไว้ แสดงอยู่ในตารางที่ 3.3

ตารางที่ 3.9 แผนการดำเนินงาน

งาน	เดือน					
	มิ.ย.-ก.ค.61	ส.ค.-ก.ย.61	ต.ค.-พ.ย.61	ธ.ค. 61 -ม.ค.62	ก.พ.-มี.ค.62	เม.ย.-พ.ค. 62
ศึกษาข้อมูลและรวบรวมงานวิจัย TCP และ MPTCP แบบ Coupled และ Uncoupled						
ศึกษาหลักการทำงาน MPTCP						
ศึกษาหลักการทำงาน LTE						
ศึกษาหลักการทำงาน WiFi						
ศึกษาหลักการทำงาน MPTCP บน LTE และ WiFi บนโปรแกรมการจำลองเครือข่าย (Network Simulation 3)						
ออกแบบจำลองระบบ MPTCP						
ทดสอบ ตรวจสอบ และปรับปรุงระบบ						
สรุปผลการจำลองระบบและประโยชน์ที่ได้รับ						

### 3.4 ขั้นตอนและวิธีการดำเนินงาน

ขั้นตอนการดำเนินงานวิจัยนี้เป็นการจำลองสถานการณ์และวิเคราะห์ประสิทธิภาพการใช้งานของ MPTCP ที่มี Congestion Control แบบ Uncoupled (CUBIC) และ Coupled (wVegas) ทดสอบโดยโปรแกรม NS3 โดยมีสถานะที่กำหนดไว้ในสมมติฐานการวิจัย

#### 1. Congestion Control แบบ Uncoupled (CUBIC)

1.1 จำลองและวิเคราะห์ผลของการใช้งานโปรโตคอล TCP บนโครงข่าย WiFi และดำเนินการเก็บค่าการใช้งาน ต่างๆ เพื่อทำการข้อมูลเพื่อนำมาเปรียบเทียบ

1.2 จำลองและวิเคราะห์ผลของการใช้งานโปรโตคอล TCP บนโครงข่าย LTE และดำเนินการเก็บค่าการใช้งาน ต่างๆ เพื่อทำการข้อมูลเพื่อนำมาเปรียบเทียบ



1.3 จำลองและวิเคราะห์ผลของการใช้งานโปรโตคอล MPTCP บนโครงข่าย LTE และ WiFi ดำเนินการเก็บค่าการใช้งาน ต่างๆ เพื่อทำการข้อมูลเพื่อนำมาเปรียบเทียบ

## 2. Congestion Control แบบ Coupled (wVegas)

2.1 จำลองและวิเคราะห์ผลของการใช้งานโปรโตคอล TCP บนโครงข่าย WiFi และดำเนินการเก็บค่าการใช้งาน ต่างๆ เพื่อทำการข้อมูลเพื่อนำมาเปรียบเทียบ

2.2 จำลองและวิเคราะห์ผลของการใช้งานโปรโตคอล TCP บนโครงข่าย LTE และดำเนินการเก็บค่าการใช้งาน ต่างๆ เพื่อทำการข้อมูลเพื่อนำมาเปรียบเทียบ

2.3 จำลองและวิเคราะห์ผลของการใช้งานโปรโตคอล MPTCP บนโครงข่าย LTE และ WiFi ดำเนินการเก็บค่าการใช้งาน ต่างๆ เพื่อทำการข้อมูลเพื่อนำมาเปรียบเทียบ

3. เปรียบเทียบเพื่อหาขอบเขตของ Throughput ในสถานการณ์ที่มีความแตกต่างของอัตราการรับส่งข้อมูล ในโครงข่าย MPTCP ที่ส่งผลให้ Throughput MPTCP นั้นน้อยกว่า Throughput TCP

4. นำข้อมูลพารามิเตอร์ที่ใช้เปรียบเทียบประสิทธิภาพของโครงข่าย MPTCP และ TCP เพื่อนำค่า Throughput , อัตราล่าช้าต่อหน่วยเวลา (Delay) และอัตราความแปรปรวนของความล่าช้าของข้อมูล (Jitter) ของระบบมาเปรียบเทียบ

## วิธีการดำเนินงานการนำซอฟต์แวร์ NS3 มาประยุกต์ใช้งานร่วมกับโครงข่าย LTE/WiFi

ซอฟต์แวร์ NS3 นั้นใช้ภาษาทางโปรแกรมในการพัฒนา 2 ภาษาคือ ภาษา C++ และภาษา Python โดย NS3 เป็นซอฟต์แวร์ที่ใช้ในการจำลองการทำงานของโครงข่ายโดย NS3 สามารถจำลองการทำงานของโปรโตคอลต่างๆ ได้มากมาย อาทิเช่น โปรโตคอล TCP/IP เป็นต้น ในการใช้ซอฟต์แวร์ NS3 ในการจำลองโครงข่าย LTE/WiFi

ขั้นตอนในการประยุกต์ใช้ซอฟต์แวร์ NS3 จะมี อยู่ 2 แบบคือ 1. NS3 AllinOne 2. NS3 DCE (Direct Code Execution) ในงานวิจัยนี้เลือกใช้ NS3 DCE กับโครงข่าย LTE/WiFi นั้นจะประกอบไปด้วยขั้นตอนการติดตั้งของ NS3 DCE<sup>1</sup> มีขั้นตอนการติดตั้งอยู่ 2 โหมด ดังต่อไปนี้

## 1. โหมดพื้นฐาน (The basic mode)

### 1.1 โหลดตั้งค่าตัวแปรบางอย่างของ Bake

```
hg clone http://code.nsnam.org/bake bake
export BAKE_HOME=`pwd`/bake
export PATH=$PATH:$BAKE_HOME
export PYTHONPATH=$PYTHONPATH:$BAKE_HOME
```

### 1.2 สร้างไคเรกทอรีสำหรับ DCE และติดตั้งโดย Bake

```
mkdir dce
cd dce
bake.py configure -e dce-ns3-|version|
bake.py download
bake.py build
```

ในการวิจัยนี้ใช้ version 1.7 , dce-ns3-1.7

### 1.3 ตรวจสอบว่าการติดตั้งสมบูรณ์หรือไม่

## 2. โหมดขั้นสูง (The advanced mode ของ Linux kernel)

### 2.1 สร้างไคเรกทอรีสำหรับ DCE และติดตั้งโดย Bake

```
mkdir dce
cd dce
bake.py configure -e dce-linux-|version|
bake.py download
bake.py build
```

ในการวิจัยนี้ใช้ version 1.7 , dce-ns3-1.7

### 2.2 ติดตั้ง DCE using WAF ที่ใช้สำหรับการกำหนดค่าการรวบรวมและการติดตั้งโปรแกรม

สคริปต์การกำหนดค่าจะถูกเขียนในไฟล์หลามชื่อ wscript เรียกกรอบ WAF และเรียกได้ว่าเป็นปฏิบัติการ waf

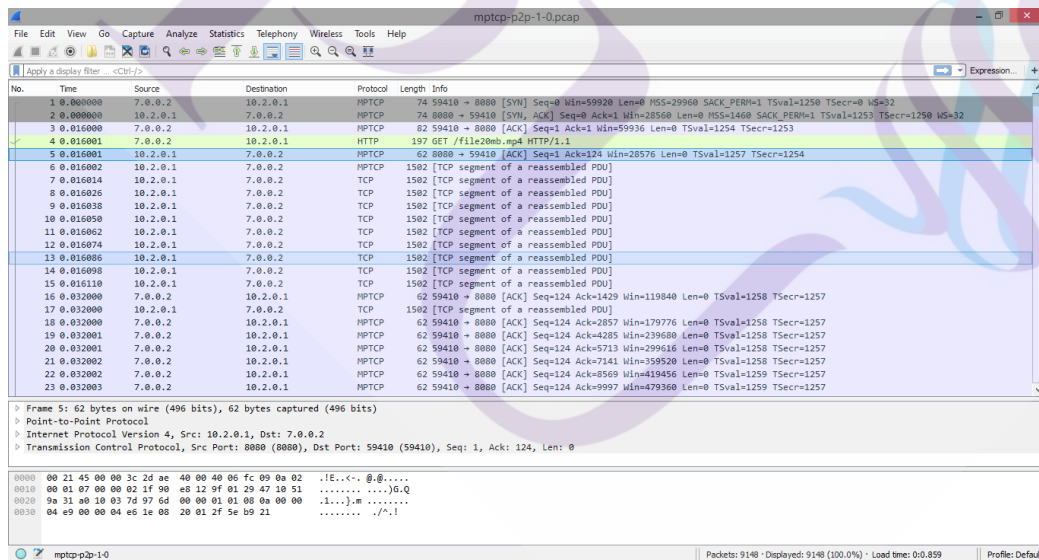
- # Download pybindgen (optional)
- # Download ns-3
- # Configure
- # Build and install in the directory specified by
- # --prefix parameter

2.3 คำนวณโหลดและติดตั้ง net-next-sim and DCE (net-next-sim includes the linux stack module):

- # Clone net-next-sim
- # Select a kernel version
- # Configure and build
- # Download, configure, build and install DCE

**Output NS3**

- NS3 แสดงผลออกมาได้ ในรูปแบบ Tracing and packet traces โดยสร้างไฟล์มาใช้งาน เป็น pcap ที่สามารถจะนำไปเปิด กับ wireshark ตามรายละเอียด ภาพที่ 3.2



ภาพที่ 3.2 รูปแบบ Tracing and packet traces

ในการวิจัยนี้ได้ใช้งาน file นามสกุล pcap โดยใช้ wireshark ในการคำนวณหาค่า Throughput, Loss, Delay, Jitter

- การหา Throughput<sup>1</sup> หรือ อัตราการรับ-ส่งข้อมูลบนโครงข่าย
- การหาค่า Delay หรือ อัตราล่าช้าต่อหน่วยเวลา ต้องดูที่ค่า Timestamp<sup>2</sup> เพื่อนำข้อมูลมาหาค่า Delay กำหนดค่า Timestamp จะต้อง เพิ่ม column ที่ wireshark และทำการ Export Packet Dissection เป็น AS CSV.. ทำการเปิด Excal เพื่อการทำการคำนวณหาค่า Delay
- การหาค่า Jitter หรือ อัตราความแปรปรวนของความล่าช้าของข้อมูล ซึ่งได้จากค่า Delay โดย กำหนดสูตรการหาค่าใน excal = STDEV ก็จะได้ค่า Jitter

---

<sup>1</sup> ภาณุวง เมฆไพบูลย์.(2559).การประเมินประสิทธิภาพโปรโตคอล MPTCP บนโครงข่าย LTE และ WiFi.EENET.ประเทศไทย

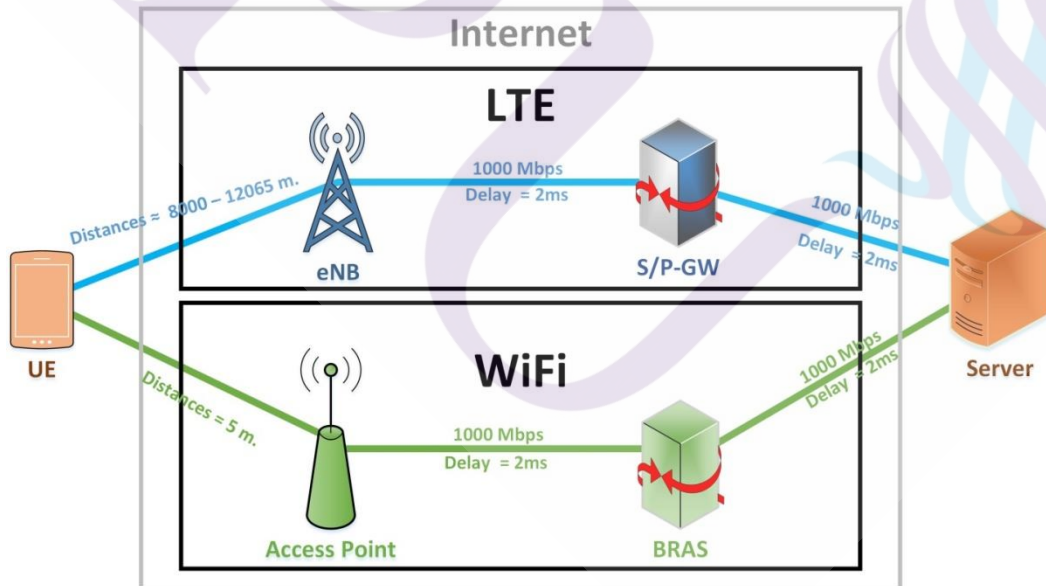
<sup>2</sup> <https://www.youtube.com/watch?v=QqKAnZnHss0>

## บทที่ 4

### ผลการศึกษา

#### 4.1 โครงข่าย LTE และ WiFi ในการทดสอบ

ในบทนี้จะกล่าวถึงผลการศึกษาวิจัยและการอภิปรายผลการวิจัย โดยผู้วิจัยทำการทดสอบการทำงานของ MPTCP เทียบกับ TCP โดยการทดสอบการทำงานมีการใช้วิธีการควบคุมความคับคั่งของข้อมูล (Congestion Control Algorithm) แบบ Coupled ด้วยอัลกอริทึม wVegas และ Uncoupled ด้วยอัลกอริทึม CUBIC ซึ่งใช้ Application เป็น wget เพื่อ Download ข้อมูลจากเครื่องแม่ข่าย (Server) ไปให้เครื่องปลายทาง(UE) โดยมีการส่งข้อมูลผ่าน 2 โครงข่าย (ในที่นี้คือโครงข่าย LTE และ WiFi) โดยไม่มี ทราฟฟิก Background ทางโครงข่าย LTE และ WiFi) ซึ่งได้ทดสอบการรับ-ส่งข้อมูลตามการใช้งาน ดังรูปที่ 4.1



ภาพที่ 4.1 โครงข่าย 4G LTE และ WiFi ที่ใช้ในงานวิจัย

การเชื่อมต่อระหว่างโครงข่ายทั้ง 2 นั้นจะเป็นการเชื่อมต่อโดยใช้โปรโตคอลของ MPTCP และ TCP ซึ่งจะคำนึงถึงการเชื่อมต่อในการให้บริการ โดยไม่ส่งผลกระทบต่อการใช้งานของ Application และไม่มีการเคลื่อนที่ของโทรศัพท์เคลื่อนที่ของผู้ใช้งาน รวมถึงเครื่องแม่ข่ายและอุปกรณ์ภายในโครงข่าย LTE และ WiFi การเชื่อมต่อระหว่างโครงข่ายทั้ง 2 นั้นจะเป็นการเชื่อมต่อโดยใช้โปรโตคอลของ MPTCP และ TCP ซึ่งจะคำนึงถึงการเชื่อมต่อในการให้บริการ โดยไม่ส่งผลกระทบต่อการใช้งานของ Application และไม่มีการเคลื่อนที่ของโทรศัพท์เคลื่อนที่ของผู้ใช้งาน รวมถึงเครื่องแม่ข่ายและอุปกรณ์ภายในโครงข่าย LTE และ WiFi

#### 4.2 ขั้นตอนในการทดสอบ

ในการทดสอบงานวิจัยแบ่งเป็น 2 ขั้นตอน ดังนี้

**ขั้นตอนที่ 1.** การทดสอบเพื่อหา Throughput ของ  $TCP_{Max}$

ซึ่งนิยามของ  $TCP_{Max}$  คืออัตราการรับ - ส่งข้อมูล TCP ผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว โดยจะเป็นค่า Throughput ของ TCP ฟังก์ชันที่มี Throughput มากที่สุด และดำเนินการเก็บค่าดังกล่าวนี้เป็นค่า  $TCP_{Max}$  เพื่อหาจำนวนเท่าของความแตกต่างในอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE และเปรียบเทียบความแตกต่างระหว่าง  $TCP_{Max}$  เทียบกับค่า Throughput ของ MPTCP

**ขั้นตอนที่ 2.** การทดสอบการทำงานของ MPTCP

โดยการส่งข้อมูลผ่านโครงข่าย LTE กับ WiFi พร้อมกัน และคำนวณค่า Sum of Throughput ของ MPTCP ซึ่งนั่นก็คือค่า Throughput ของ MPTCP

ในการทดสอบนั้นได้เลือกใช้ Congestion Control แบบ Uncoupled (CUBIC) และ Coupled (wVegas) และทดสอบด้วยการส่งข้อมูลวิดีโอขนาด 229.4 MB (ความคมชัด 720P (1280×720)) ซึ่งการเปรียบเทียบความแตกต่างระหว่าง  $TCP_{Max}$  เทียบกับค่า Throughput ของ MPTCP นั้น จะถูกนำมาวิเคราะห์ในรูปแบบของจำนวนเท่า โดยเลือกอัตราที่จำนวน 1 ถึง 9 เท่า

โดยผลการศึกษาวิจัยจะวัดผลจากค่าประสิทธิภาพโครงข่าย ต่อไปนี้

- ค่าอัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput)
- ค่าความล่าช้าทางเวลา (Delay)
- ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter)

### 4.3 การทดสอบ

การทดสอบ แบ่งเป็น 3 กรณี คือ

การทดสอบที่ 1: การประเมิน Throughput ของ MPTCP เทียบกับ Throughput ของ TCP<sub>Max</sub> โดยใช้ Uncoupled (CUBIC)

การทดสอบที่ 2: การประเมิน Throughput ของ MPTCP ระหว่าง Uncoupled (CUBIC) เทียบกับ Coupled (wVegas)

การทดสอบที่ 3: การประเมิน Throughput ของ MPTCP เทียบกับ Throughput ของ TCP<sub>Max</sub> โดยใช้ Coupled (wVegas)

โดยมีรายละเอียดดังนี้

การทดสอบที่ 1: การประเมิน Throughput ของ MPTCP เทียบกับ Throughput ของ TCP<sub>Max</sub> โดยใช้ Uncoupled (CUBIC)

การทดสอบที่ 1.1 การทดสอบเพื่อหา Throughput ของ TCP<sub>Max</sub> โดยรับ - ส่งข้อมูลผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว

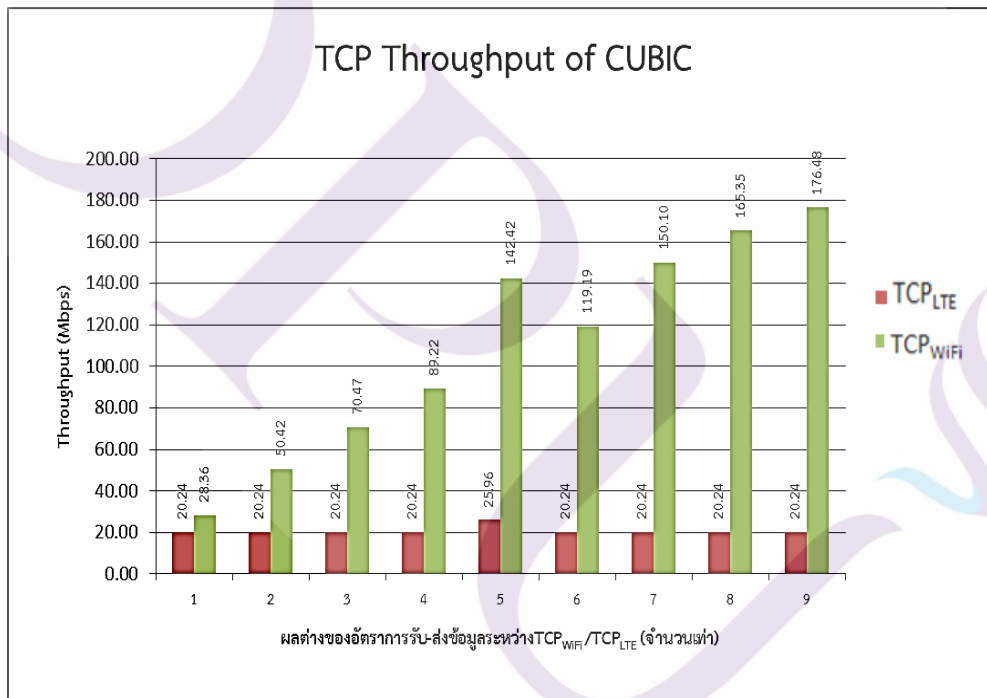
ในการทดสอบนี้ ผู้วิจัยได้มีการกำหนดค่า MCS ของฝั่ง WiFi โดยดำเนินการปรับค่า MCS ที่ 0-9 ไปตามลำดับ เพื่อให้ได้ผลต่าง ที่ 1 ถึง 9 เท่า ตามระบุไว้ข้างต้น และเมื่อยังไม่ได้อัตราการความแตกต่าง (จำนวนเท่า) ที่ต้องการ ลำดับต่อไปจึงได้ดำเนินการปรับระยะห่างระหว่าง UE-AP หรือ UE-eNB เป็นลำดับต่อไป โดยจากการทดสอบนั้นได้จำนวนเท่าของความแตกต่างในอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE ตามแสดงดังตารางที่ 4.1

ตารางที่ 4.1 แสดงข้อมูล TCP<sub>Max</sub> Throughput ผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว โดยใช้ Uncoupled (CUBIC)

ผลต่าง (จำนวนเท่า)	WiFi MCS	ระยะห่าง UE-AP (m.)	ระยะห่าง UE-eNB (m.)	ค่าเฉลี่ย Throughput (Mbps)		
				WiFi	LTE	TCP <sub>Max</sub>
1	0	5	12065	28.36	20.24	28.36
2	1	5	12065	50.42	20.24	50.42
3	2	5	12065	70.47	20.24	70.47
4	3	5	12065	89.22	20.24	89.22

5	4	5	8000	142.42	25.96	142.42
6	4	5	12065	119.19	20.24	119.19
7	6	5	12065	150.10	20.24	150.10
8	8	5	12065	165.35	20.24	165.35
9	9	5	12065	176.48	20.24	176.48

ซึ่งจากตารางที่ 4.1 จะได้ว่าซึ่งค่า  $TCP_{Max}$  โดยข้อมูลจากตารางนั้น แสดงให้เห็นว่า ในการใช้ Uncoupled (CUBIC) ในการทดสอบนั้นส่งผลให้ TCP Throughput ฝั่งที่ให้ค่ามากที่สุด คือฝั่งของโครงข่าย WiFi ดังภาพที่ 4.2



ภาพที่ 4.2 กราฟแสดงค่า TCP Throughput ของ CUBIC ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE



จากภาพที่ 4.2 แสดงค่า Throughput ที่ได้ ของฝั่งโครงข่าย WiFi และ โครงข่าย LTE ที่ทำให้เกิดผลต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE ที่ 1- 9 เท่า และในการทดสอบนี้จะใช้ค่า TCP Throughput ที่มากที่สุด เพื่อนำไปเปรียบเทียบกับ Throughput ของ MPTCP โดยใช้ Uncoupled (CUBIC)

จากการทดสอบที่ 4.1 ผลของ TCP<sub>Max</sub> มี Throughput, Delay , Jitter ตามแสดงดังตารางที่ 4.2

**ตารางที่ 4.2** แสดงข้อมูล อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) เฉลี่ย ผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว โดยใช้ Uncoupled (CUBIC)

ผลต่าง (จำนวนเท่า)	ค่าเฉลี่ย TCP <sub>Max</sub> Throughput (Mbps)	ค่าเฉลี่ย Delay (ms)	ค่าเฉลี่ย Jitter (ms)
1	28.36	0.293	3.923
2	50.42	0.162	2.244
3	70.47	0.116	1.847
4	89.22	0.092	1.507
5	142.42	0.069	1.696
6	119.19	0.069	1.696
7	150.10	0.055	1.646
8	165.35	0.050	2.522
9	176.48	0.047	2.001

จากตารางที่ 4.2 เป็นการแสดงผลของ ค่าอัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) โดยเฉลี่ย ของการรับ-ส่งข้อมูลบนโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว

**การทดสอบที่ 1.2** การทดสอบเพื่อหา Throughput ของ MPTCP โดยรับ - ส่งข้อมูลผ่านโครงข่าย LTE และ WiFi พร้อมกัน

ในการทดสอบนั้น ผู้วิจัยได้มีการกำหนดค่า MCS ของฝั่ง WiFi และ ระยะห่างเช่นเดียวกันกับการทดสอบที่ 1.1 ซึ่งมีความแตกต่างกันที่ ในการทดสอบที่ 1.2 นี้ มีการประเมินประสิทธิภาพของ TCP ที่มีการรับ-ส่งข้อมูลผ่านโครงข่าย LTE หรือ WiFi พร้อมกัน เพื่อหา อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput) ของ MPTCP โดยใช้ Uncoupled (CUBIC) ตามแสดงดังตารางที่ 4.3

**ตารางที่ 4.3** แสดงข้อมูล MPTCP Throughput ผ่านโครงข่าย LTE และ WiFi พร้อมกัน โดยใช้ Uncoupled (CUBIC)

ผลต่าง (จำนวนเท่า)	WiFi MCS	ระยะห่าง UE-AP (m.)	ระยะห่าง UE-eNB (m.)	ค่าเฉลี่ย Throughput (Mbps)		
				WiFi	LTE	MPTCP
1	0	5	12065	26.14	19.94	46.05
2	1	5	12065	41.28	19.41	60.64
3	2	5	12065	45.02	19.68	65.26
4	3	5	12065	45.63	19.37	64.31
5	4	5	8000	64.88	15.40	101.57
6	4	5	12065	76.06	17.00	81.78
7	6	5	12065	77.19	11.90	89.81
8	8	5	12065	78.05	11.78	87.70
9	9	5	12065	86.34	11.39	88.43

ซึ่งจากผลของ MPTCP มี Throughput, Delay , Jitter ตามแสดงดังตารางที่ 4.4

**ตารางที่ 4.4** แสดงข้อมูล อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) เฉลี่ย ผ่านโครงข่าย LTE และ WiFi พร้อมกัน โดยใช้ Uncoupled (CUBIC)

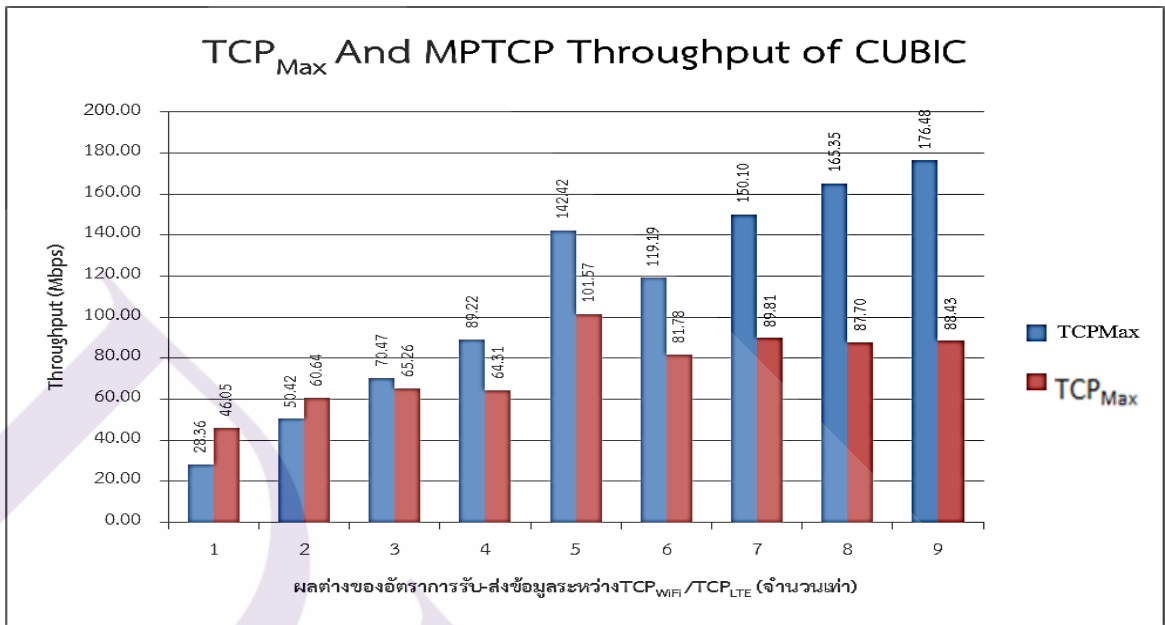
ผลต่าง (จำนวนเท่า)	ค่าเฉลี่ย MPTCP Throughput (Mbps)	ค่าเฉลี่ย Delay (ms)		ค่าเฉลี่ย Jitter (ms)	
		LTE	WiFi	LTE	WiFi
1	46.05	0.373	0.379	1.498	3.255
2	60.64	0.400	0.231	10.186	1.979
3	65.26	0.411	0.192	2.002	1.503
4	64.31	0.425	0.185	3.625	1.394
5	101.57	0.456	0.133	4.861	1.188
6	81.78	0.474	0.142	3.880	3.015
7	89.81	0.667	0.116	4.287	1.217
8	87.70	0.691	0.110	4.183	0.940
9	88.43	0.723	0.104	4.255	1.685

ซึ่งผลการทดสอบให้อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput) ของ MPTCP เทียบกับ TCP Throughput โดยใช้ Uncoupled (CUBIC) เป็นดังตารางที่ 4.5

ตารางที่ 4.5 แสดงค่า Throughput ของ CUBIC MPTCP เทียบกับ Throughput ของ CUBIC TCP<sub>Max</sub> ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE

ผลต่าง (จำนวนเท่า)	ค่าเฉลี่ย MPTCP Throughput (Mbps)	ค่าเฉลี่ย TCP <sub>Max</sub> Throughput (Mbps)
1	46.05	28.36
2	60.64	50.42
3	65.26	70.47
4	64.31	89.22
5	101.57	142.42
6	81.78	119.19
7	89.81	150.10
8	87.70	165.35
9	88.43	176.48

และภาพที่ 4.3



ภาพที่ 4.3 กราฟแสดงค่า Throughput ของ CUBIC MPTCP เทียบกับ Throughput ของ CUBIC TCP<sub>Max</sub> ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE

จากภาพที่ 4.3 แสดงให้เห็นว่า การใช้งาน Uncoupled (CUBIC) ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลบนโครงข่าย MPTCP ที่ 1 เท่า และ 2 เท่า นั้นมี Throughput เฉลี่ยที่สูงกว่า TCP<sub>Max</sub> โดย MPTCP มี Throughput เฉลี่ยอยู่ที่ประมาณ 46.05 Mbps และ 60.64 Mbps ตามลำดับ และ TCP<sub>Max</sub> มี Throughput เฉลี่ยอยู่ที่ประมาณ 28.36 Mbps และ 50.42 Mbps ตามลำดับและตั้งแต่ 3 เท่าขึ้นไป MPTCP มี Throughput ที่ต่ำกว่า TCP<sub>Max</sub> โดย MPTCP มี Throughput เฉลี่ยอยู่ที่ประมาณ 65.26, 64.31, 101.57, 81.78, 89.81, 87.70 และ 88.43 Mbps ตามลำดับ และ TCP<sub>Max</sub> มี Throughput เฉลี่ยอยู่ที่ประมาณ 70.47, 89.22, 142.22, 119.19, 150.10, 165.35 และ 176.48 Mbps ตามลำดับ โดยค่าเฉลี่ยของ Throughput มีความห่างไกลจากค่า TCP<sub>Max</sub> และมีอัตราที่เพิ่มมากขึ้นตามอัตราความแตกต่างของการรับ-ส่งข้อมูล

**การทดสอบที่ 2 :** การประเมิน Throughput ของ MPTCP เทียบกับ Throughput ของ TCP<sub>Max</sub> โดยใช้ Coupled (wVegas)

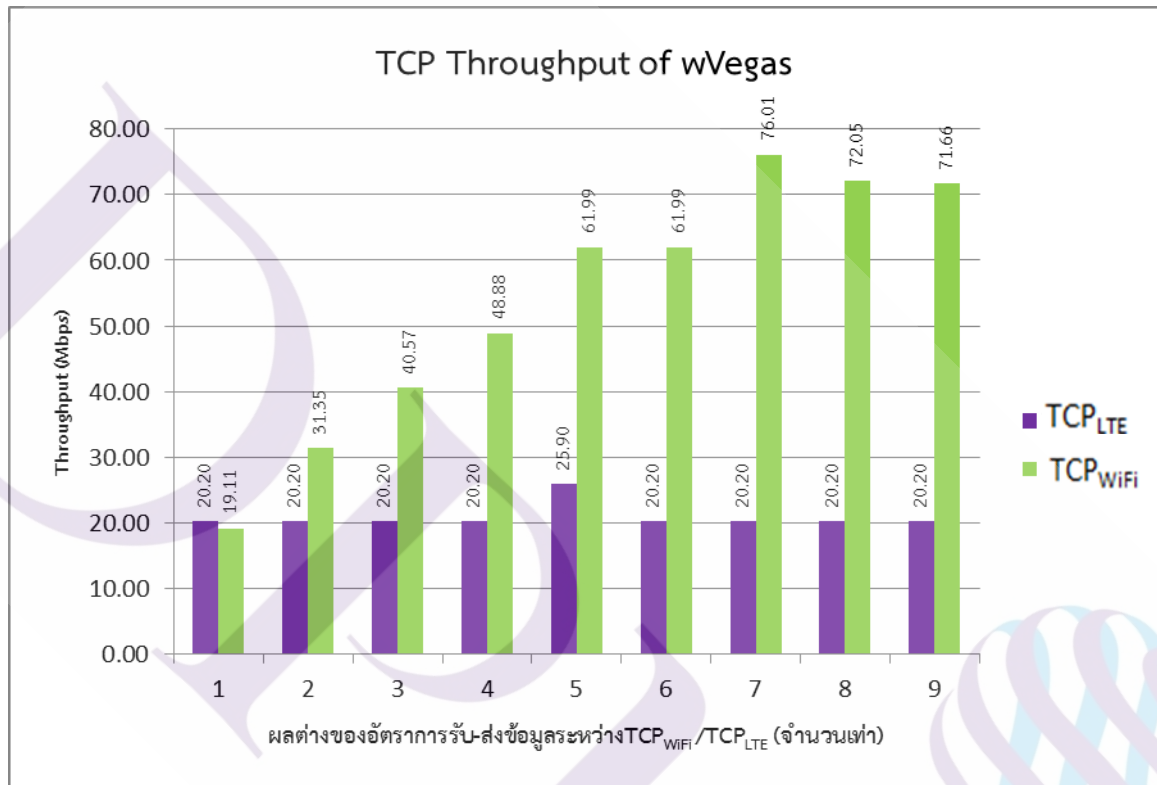
**การทดสอบที่ 2.1** การทดสอบเพื่อหา Throughput ของ TCP<sub>Max</sub> โดยรับ - ส่งข้อมูลผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว

ในการทดสอบนั้น ผู้วิจัยได้มีการกำหนดค่า MCS ของฝั่ง WiFi และ ระยะห่างเช่นเดียวกันกับการทดสอบที่ 1.1 โดยมีการประเมินประสิทธิภาพของ TCP ที่มีการรับส่งข้อมูลบนโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว แต่เปลี่ยนวิธีการควบคุมความคับคั่งเป็นแบบ Coupled (wVegas) เพื่อหาอัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput) ของ TCP<sub>Max</sub> ตามแสดงดังตารางที่ 4.5

**ตารางที่ 4.6** แสดงข้อมูล TCP<sub>Max</sub> Throughput ผ่านโครงข่าย LTE หรือ WiFi เพียงเส้นทางเดียว โดยใช้ Coupled (wVegas)

ผลต่าง (จำนวนเท่า)	WiFi MCS	ระยะห่าง UE-AP (m.)	ระยะห่าง UE-eNB (m.)	ค่าเฉลี่ย Throughput (Mbps)		
				WiFi	LTE	TCP <sub>Max</sub>
1	0	5	12065	19.11	20.20	20.20
2	1	5	12065	31.35	20.20	31.35
3	2	5	12065	40.57	20.20	40.57
4	3	5	12065	48.88	20.20	48.88
5	4	5	8000	61.99	25.90	61.99
6	4	5	12065	61.99	20.20	61.99
7	6	5	12065	76.01	20.20	76.01
8	8	5	12065	72.05	20.20	72.05
9	9	5	12065	71.66	20.20	71.66

ซึ่งจากตารางที่ 4.6 จะได้ว่าซึ่งค่า  $TCP_{Max}$  โดยข้อมูลจากตารางนั้น แสดงให้เห็นว่า ในการใช้ Uncoupled (CUBIC) ในการทดสอบนั้นส่งผลให้ TCP Throughput ฝั่งที่ให้ค่ามากที่สุด โดยส่วนมาก คือฝั่งของ โครงข่าย WiFi ดังภาพที่ 4.4



ภาพที่ 4.4 กราฟแสดงค่า TCP Throughput ของ wVegas ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE

ซึ่งจากผลของ  $TCP_{Max}$  มี Throughput, Delay, Jitter ตามแสดงดังตารางที่ 4.6

**ตารางที่ 4.7** แสดงข้อมูล อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) เฉลี่ย ผ่านโครงข่าย LTE หรือWiFi เพียงเส้นทางเดียว โดยใช้ Coupled (wVegas)

ผลต่าง (จำนวนเท่า)	ค่าเฉลี่ย TCP <sub>Max</sub> Throughput (Mbps)	ค่าเฉลี่ย Delay (ms)	ค่าเฉลี่ย Jitter (ms)
1	20.20	0.405	1.461
2	31.35	0.261	0.929
3	40.57	0.202	0.619
4	48.88	0.168	0.613
5	61.99	0.133	0.860
6	61.99	0.133	0.860
7	76.01	0.108	1.240
8	72.05	0.114	1.272
9	71.66	0.115	1.393

**การทดสอบที่ 2.2** การทดสอบเพื่อหา Throughput ของ MPTCP โดยรับ - ส่งข้อมูลผ่านโครงข่าย LTE กับ WiFi พร้อมกัน

ในการทดสอบนั้น ผู้วิจัยได้มีการกำหนดค่า MCS ของฝั่ง WiFi และ ระยะห่างเช่นเดียวกันกับการทดสอบที่ 2.1 โดยมีการประเมินประสิทธิภาพของ TCP ที่มีการรับส่งข้อมูลบนโครงข่าย LTE หรือ พร้อมกัน และใช้วิธีการควบคุมความคับคั่งเป็นแบบ Coupled (wVegas) เพื่อหา อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput) ของ MPTCP ตามแสดงดังตารางที่ 4.8



ตารางที่ 4.8 แสดงข้อมูล MPTCP Throughput ผ่านโครงข่าย LTE และ WiFi พร้อมกัน โดยใช้ Coupled (wVegas)

ผลต่าง (จำนวนเท่า)	WiFi MCS	ระยะห่าง UE-AP (m.)	ระยะห่าง UE-eNB (m.)	ค่าเฉลี่ย Throughput (Mbps)		
				WiFi	LTE	MPTCP
1	0	5	12065	17.11	19.82	36.92
2	1	5	12065	27.33	19.61	46.90
3	2	5	12065	35.75	17.98	53.69
4	3	5	12065	42.26	17.05	59.27
5	4	5	8000	50.74	25.46	76.15
6	4	5	12065	54.15	16.36	70.46
7	6	5	12065	64.30	18.04	82.23
8	8	5	12065	70.97	17.93	88.76
9	9	5	12065	73.85	17.54	91.27

ซึ่งจากผลของ MPTCP มี Throughput, Delay , Jitter ตามแสดงดังตารางที่ 4.9

**ตารางที่ 4.9** แสดงข้อมูล อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput), ค่าความล่าช้าทางเวลา (Delay), ค่าความแปรปรวนของค่าความล่าช้าทางเวลา (Jitter) เฉลี่ย ผ่านโครงข่าย LTE และ WiFi พร้อมกัน โดยใช้ Coupled (wVegas)

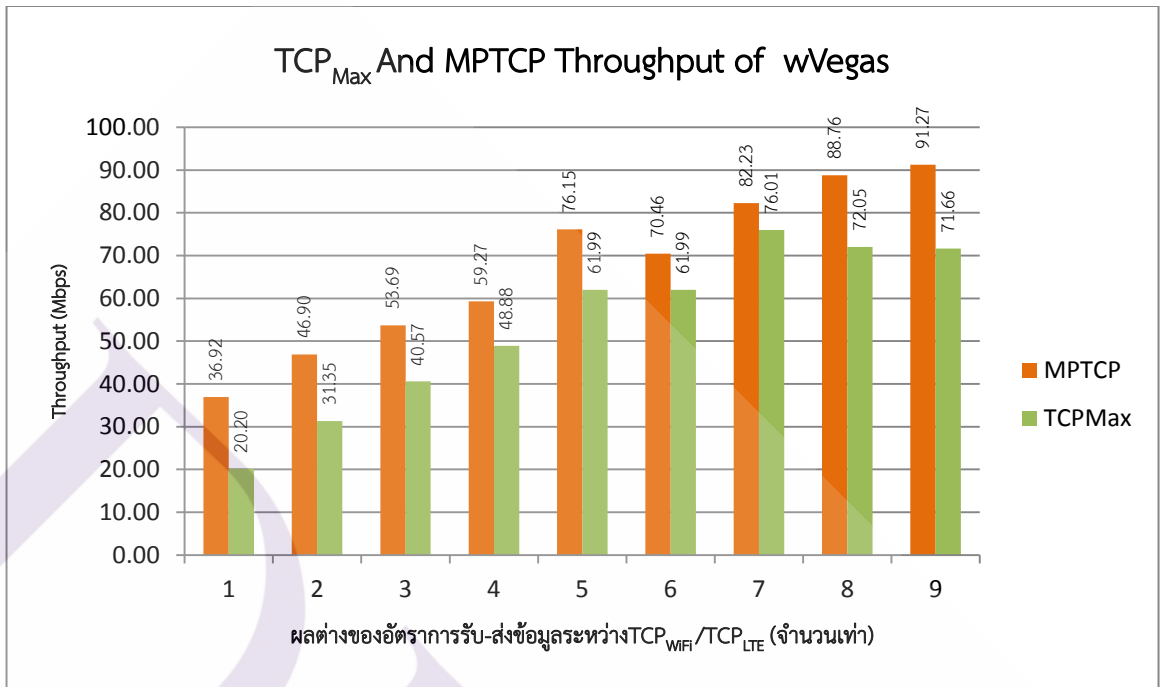
ผลต่าง (จำนวนเท่า)	ค่าเฉลี่ย MPTCP Throughput (Mbps)	ค่าเฉลี่ย Delay (ms)		ค่าเฉลี่ย Jitter (ms)	
		LTE	WiFi	LTE	WiFi
1	36.92	0.387	0.720	1.452	1.095
2	46.90	0.409	0.449	1.458	0.969
3	53.69	0.450	0.347	1.596	0.724
4	59.27	0.481	0.297	1.585	0.826
5	76.15	0.323	0.237	1.460	0.617
6	70.46	0.513	0.233	1.607	0.483
7	82.23	0.471	0.188	1.642	1.171
8	88.76	0.469	0.170	1.773	1.275
9	91.27	0.480	0.163	1.857	1.375

ซึ่งผลการทดสอบให้อัตราการรับ-ส่งข้อมูลบนโครงข่าย (Throughput) ของ MPTCP โดยใช้ wVegas เป็นตารางที่ 4.10

**ตารางที่ 4.10** แสดงค่า Throughput ของ wVegas MPTCP เทียบกับ Throughput ของ wVegas TCP<sub>Max</sub> ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/ LTE

ผลต่าง (จำนวนเท่า)	ค่าเฉลี่ย MPTCP Throughput (Mbps)	ค่าเฉลี่ย TCP <sub>Max</sub> Throughput (Mbps)
1	36.92	20.20
2	46.90	31.35
3	53.69	40.57
4	59.27	48.88
5	76.15	61.99
6	70.46	61.99
7	82.23	76.01
8	88.76	72.05
9	91.27	71.66

และภาพที่ 4.5



ภาพที่ 4.5 กราฟแสดงค่า Throughput ของ wVegas MPTCP เทียบกับ Throughput ของ wVegas TCP<sub>Max</sub> ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลระหว่างโครงข่าย WiFi/LTE

จากภาพที่ 4.5 แสดงให้เห็นว่า การใช้งาน Coupled (wVegas) ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล ของ wVegas นั้น บนโครงข่าย MPTCP มี Throughput เฉลี่ย ที่สูงกว่า TCP<sub>Max</sub> โดย MPTCP มี Throughput เฉลี่ยอยู่ที่ประมาณ 36.92, 46.90, 53.69, 59.27, 76.15, 70.46, 82.23, 88.76 และ 91.27 Mbps ตามลำดับ และ TCP<sub>Max</sub> มี Throughput เฉลี่ยอยู่ที่ประมาณ 20.20, 31.35, 40.57, 48.88, 61.99, 61.99, 76.01, 72.05 และ 71.66 Mbps ตามลำดับ โดยมีเมื่ออัตราความแตกต่างของการรับ-ส่งข้อมูลที่เพิ่มมากขึ้น Throughput บน MPTCP นั้นยังคงให้ค่าที่ใกล้เคียง TCP<sub>Max</sub>

**การทดสอบที่ 3:** การประเมิน Throughput ของ MPTCP ระหว่าง Uncoupled (CUBIC) เทียบกับ Coupled (wVegas)

การทดสอบเพื่อเปรียบเทียบ Throughput ของ MPTCP เมื่อมีการใช้งาน Congestion Control ระหว่าง Uncoupled (CUBIC) เทียบกับ Coupled (wVegas) โดยรับ – ส่งข้อมูลผ่านโครงข่าย LTE กับ WiFi พร้อมกัน ซึ่งในการเปรียบเทียบนี้จะเป็นการเปรียบเทียบกับตาราง MCS ของ LTE และ WiFi โดยอ้างอิงจากรายที่ 2.1, 2.2 และ 2.3

เนื่องจากการกำหนด MCS ของ LTE ของระบบเป็นแบบอัตโนมัติ โดยจากผลการทดสอบจะมี MCS =12 ซึ่งอยู่ในช่วง 16 QAM (มีค่า MCS = 10 ถึง 16 อ้างอิงจากรายที่ 2.1 และ 2.2 ) และเมื่อมีการรับ-ส่งข้อมูลในช่วง 20 MHz แบบ SISO จะส่งผลให้มีค่า Throughput สูงสุดอยู่ที่ 30.576 Mbps และในส่วน MCS ของ WiFi จะมีการกำหนดค่าการทดสอบที่ 0 – 9 เพื่อให้เห็นความแตกต่างของช่วง Throughput ที่จะเกิดขึ้นอ้างอิงตามตารางที่ 2.3

รายละเอียดในการใช้ MCS ของ WiFi และ LTE นั้นจะมีค่า Throughput สูงสุด เป็นไปตาม ตารางที่ 4.11 ซึ่งข้อมูลนี้จะใช้ในการเปรียบเทียบการทดสอบในลำดับต่อไป

ตารางที่ 4.11 แสดงค่า TCP Throughput สูงสุด ของตามตารางแสดง MCS บนโครงข่าย WiFi และ LTE

WiFi		LTE	
MCS	Throughput สูงสุด (Mbps)	MCS	Throughput สูงสุด (Mbps)
9	433.3	12	30.576
8	390	12	30.576
7	325	12	30.576
6	292.5	12	30.576
5	260	12	30.576
4	195	12	30.576
3	130	12	30.576
2	97.5	12	30.576
1	65	12	30.576
0	32.5	12	30.576

จากตารางที่ 4.11 จะส่งผลให้ MPTCP มี Throughput มีใกล้เคียงหรือเท่ากับ  $TCP_{Max}$  หรือ Throughput ของ TCP ที่มากที่สุด โดยมี MPTCP Throughput เป็นไปตามตารางที่ 4.12 ซึ่งจาก TCP Throughput ดังกล่าวนั้นทำให้เกิดผลต่าง TCP Throughput ทั้งสองฝั่ง ดังแสดงในตารางที่ 4.12

ตารางที่ 4.12 แสดงค่า MPTCP Throughput บนโครงข่าย WiFi และ LTE ในกรณีที่ระบบให้ค่า TCP Throughput สูงสุด

WiFi MCS	ผลลัพธ์ Throughput (Mbps)			ผลต่าง TCP (WiFi/ LTE)
	WiFi	LTE	MPTCP	
9	433.3	30.576	433.3	14
8	390	30.576	390	13
7	325	30.576	325	11
6	292.5	30.576	292.5	10
5	260	30.576	260	9
4	195	30.576	195	6
3	130	30.576	130	4
2	97.5	30.576	97.5	3
1	65	30.576	65	2
0	32.5	30.576	32.5	1

และจากผลในตารางที่ 4.5 การใช้งาน Uncoupled (CUBIC) ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลบนโครงข่าย MPTCP ตั้งแต่ 3 เท่าขึ้นไป บน MPTCP มี Throughput ที่ต่ำกว่า  $TCP_{Max}$  โดยผลต่าง 3 เท่าที่เกิดขึ้นนั้น เกิดจากการใช้งาน WiFi ด้วย MCS 2 ซึ่งเมื่อเทียบกับการใช้งาน Coupled (wVegas) ด้วย MCS 2 นั้น ผลของ MPTCP Throughput ยังคงทำงานได้ดีกว่า  $TCP_{Max}$  Throughput ดังแสดงในตารางที่ 4.13

ตารางที่ 4.13 แสดงข้อมูล MPTCP Throughput ระหว่าง Uncoupled (CUBIC) เทียบกับ Coupled (wVegas) ผ่านโครงข่าย LTE และ WiFi พร้อมกัน

ผลลัพธ์ Throughput (Mbps)					
WiFi	ตาม Index	จากการทดสอบด้วย CUBIC		จากการทดสอบด้วย wVegas	
MCS	MPTCP	TCP <sub>MAX</sub>	MPTCP	TCP <sub>MAX</sub>	MPTCP
9	433.3	176.48	88.43	71.66	91.27
8	390	165.35	87.70	72.05	88.76
7	325	157.49	89.81	74.50	84.25
6	292.5	150.10	81.78	76.01	82.23
5	260	142.42	25.54	72.43	78.43
4	195	119.19	101.57	69.99	70.91
3	130	89.22	64.31	48.88	59.27
2	97.5	70.47	65.26	40.57	53.69
1	65	50.42	60.64	31.35	46.90
0	32.5	28.36	46.05	19.11	36.92

จากผลในตารางที่ 4.5 และ ตารางที่ 4.13 พบว่า การใช้งาน Uncoupled (CUBIC) ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลบนโครงข่าย MPTCP ตั้งแต่ 3 เท่าขึ้นไปบน MPTCP มี Throughput ที่ต่ำกว่า TCP<sub>Max</sub> โดย MPTCP ให้ Throughput เท่ากับ 65.26 และ TCP<sub>Max</sub> ให้ Throughput เท่ากับ 70.47 โดยผลต่างของ Throughput ฝั่ง WiFi/ LTE ที่เกิดขึ้นนี้ มีอัตราที่ 3 เท่า ซึ่งเกิดจากการใช้งาน WiFi ด้วย MCS 2 ซึ่งเมื่อเทียบกับการใช้งาน Coupled (wVegas) ด้วย MCS 2 นั้น ผลของ MPTCP Throughput ยังคงทำงานได้ดีกว่า TCP<sub>Max</sub> Throughput

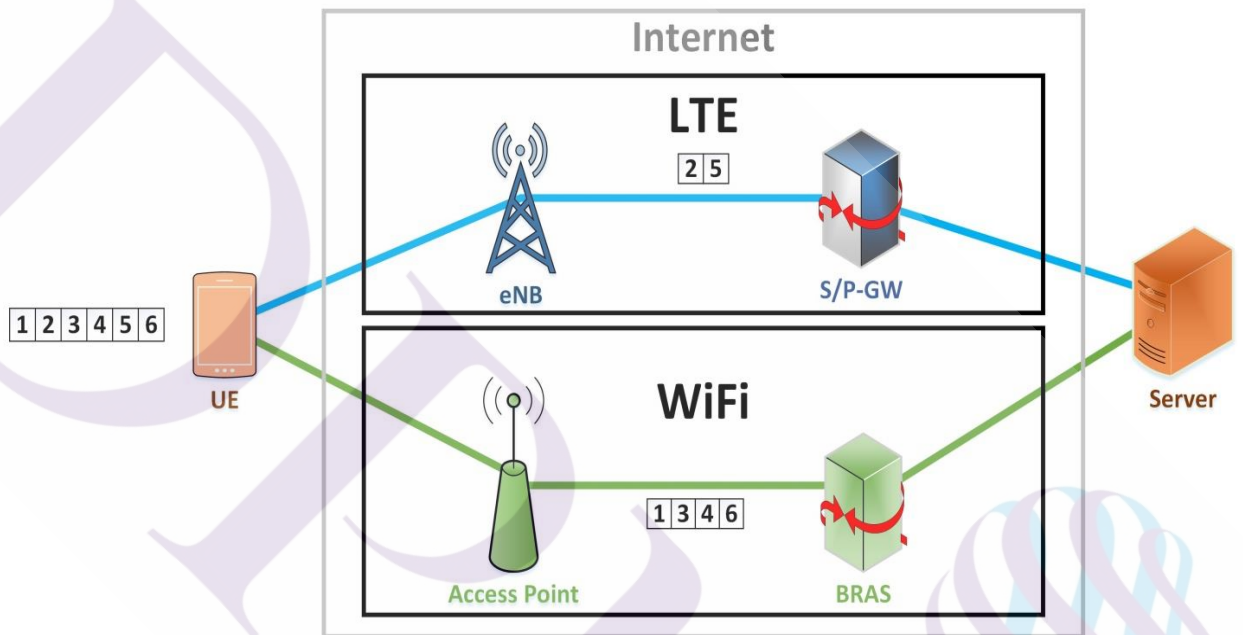
#### 4.4 อภิปรายผลการวิจัย

การรับ-ส่งข้อมูลบน TCP เป็นแบบเส้นทางเดียว แต่บน MPTCP นั้นเป็นการรับ-ส่งแบบหลายเส้นทาง (Subflows) เมื่อเกิดความแตกต่างของอัตราการรับ-ส่งข้อมูลบนโครงข่ายมากๆ ในขั้นตอนการรวมแพ็คเก็ตของข้อมูลที่ปลายทางจะต้องใช้เวลาในการรวมข้อมูลทั้งหมด จึงส่งผลต่อประสิทธิภาพของระบบและผลรวมของ Throughput บน MPTCP

จากผลการศึกษาพบว่า เมื่อเกิดสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลบนโครงข่ายระหว่างบน MPTCP บนการใช้งาน Uncoupled (CUBIC) ที่สูงเกิน 3 เท่าขึ้นไป จะส่งผลให้ MPTCP มี Throughput ที่ต่ำกว่า TCP ฟังก์ชันที่มี Throughput มากที่สุด เนื่องจากอัลกอริทึมแบบ Uncoupled (CUBIC) มีการเติบโตของ window size ที่ไม่ขึ้นกับ RTT (Round Trip Time) แต่จะขึ้นอยู่กับเหตุการณ์ความแออัดค่าสุดเท่านั้น และด้วยมาตรฐาน TCP การใช้งาน CUBIC จึงมี RTT ที่สั้นมาก ซึ่งส่งผลให้มีการได้รับ ACKs จะเร็วขึ้น ดังนั้นความแออัดของหน้าต่าง (congestion windows) จึงเติบโตได้เร็วกว่าการใช้งานวิธีควบคุมความคับคั่งแบบอื่นๆ เมื่อถึงขั้นตอนการรวมแพ็คเก็ตที่ปลายทาง ระบบจึงเกิดการรอแพ็คเก็ตข้อมูลที่ยังค้างอยู่ในโครงข่ายที่จุดปลายทาง (หรือจุดคอขวด) จนเกิด Congestion Area ที่ส่งผลให้ประสิทธิภาพของระบบและผลรวมของ Throughput ลดลงและส่งผลให้ค่า Sum of Throughput ของ MPTCP ใน CUBIC นั้นมีค่า Throughput ที่มีความก้าวร้าว (Aggressively) ซึ่งทำให้ค่าเฉลี่ยของ Throughput มีความห่างไกลจากค่า  $TCP_{Max}$  (หรือ TCP ฟังก์ชันที่มี Throughput มากที่สุด) และมีอัตราที่เพิ่มมากขึ้นตามอัตราความแตกต่างของการรับ-ส่งข้อมูล ดังภาพตัวอย่างในการรับ-ส่งข้อมูลภาพที่ 4.7 ซึ่งชุดข้อมูลประกอบด้วยแพ็คเก็ต 1,2,3,4,5 และ 6 ถูกแบ่งการรับ-ส่งเป็น 2 เส้นทาง คือบนโครงข่าย LTE มีการส่งแพ็คเก็ต 2 และ 5 บนโครงข่าย WiFi มีการส่งแพ็คเก็ต 1,3,4 และ 6 เมื่อแพ็คเก็ตที่ถูกส่งผ่านโครงข่าย WiFi มาถึงปลายทางแล้ว ระบบจะต้องรอรวมแพ็คเก็ตข้อมูลที่ถูกส่งผ่านโครงข่าย LTE ด้วยเพื่อนำข้อมูลเหล่านั้นมาจัดเรียงชุดข้อมูลใหม่อีกครั้งก่อนสิ้นสุดการดำเนินการ ลักษณะเช่นนี้จึงส่งผลให้ประสิทธิภาพของระบบและผลรวมของ Throughput ให้ลดลงไปกว่าแบบ การใช้งาน Coupled (wVegas) นั่นเอง และ บนการใช้งาน Coupled (wVegas) มี MPTCP Throughput ที่สูงกว่า TCP ฟังก์ชันที่มี Throughput มากที่สุด ( $TCP_{Max}$ ) เมื่อมีความแตกต่างของอัตราการรับ-ส่งข้อมูลที่สูงมากขึ้น Throughput บน MPTCP ยังคงมีค่า MPTCP Throughput ที่สูงกว่า  $TCP_{Max}$  และมีค่าที่ใกล้เคียง  $TCP_{Max}$  ด้วยเช่นกันซึ่งแตกต่างจาก Uncoupled (CUBIC) เพราะมีการจัดการ Congestion Windows ของทุก Subflows บนการเชื่อมต่อให้ เป็นไปตามมาตรฐานความยุติธรรมและความเป็นธรรมในจุดคอขวดบนเครือข่าย จึงทำให้ลดเวลาการรอเพื่อรวมแพ็คเก็ตที่ปลายทาง โดย wVegas จะคำนวณเส้นทางในการรับ-ส่งข้อมูลจากค่า Round-Trip



Time (RTT) พร้อมตรวจสอบแพ็คเก็ตที่ค้างค้างที่อยู่ในคิวโดยจะมีการคำนวณทุกครั้งหลังจากสิ้นสุดการดำเนินการของแต่ละรอบเพื่อตรวจสอบว่าช่องทางใดเหมาะสำหรับการรับ - ส่งข้อมูล ส่งผลให้ประสิทธิภาพของระบบและผลรวมของ MPTCP Throughput สูงกว่าแบบ Uncoupled (CUBIC) ซึ่งในกรณีที่เลวร้ายสุดของ Coupled (wVegas) นั้น จะให้ Throughput ที่ใกล้เคียงกับ TCP ฟังก์ชันที่มี Throughput มากสุดนั่นเอง



ภาพที่ 4.6 แสดงตัวอย่างการส่งแพ็คเก็ตข้อมูลบนโครงข่าย LTE และ WiFi จาก UE ไปยัง Server

โดยจากผลการทดสอบในงานวิจัยนี้ เมื่อเปรียบเทียบประสิทธิภาพของ MPTCP Throughput ระหว่างการใช้ Congestion Control แบบ Uncoupled (CUBIC) เทียบกับ Coupled (wVegas) แล้วพบว่าเมื่อมีความแตกต่างของอัตราการรับ-ส่งข้อมูล ที่ 3 เท่าขึ้นไป การใช้งานแบบ Uncoupled (CUBIC) ให้ Throughput ที่ต่ำกว่า  $TCP_{Max}$  Throughput ซึ่งเมื่อเทียบกับการใช้งาน Coupled (wVegas) แต่ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลที่สูงที่ 3 เท่า การใช้งาน Coupled (wVegas) นั้น จะให้ Throughput และประสิทธิภาพของระบบที่สูงกว่า  $TCP_{Max}$  Throughput

## บทที่ 5

### บทสรุป และข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

งานวิจัยนี้ประเมินประสิทธิภาพของวิธีกควบคุมความคับคั่งแบบ Coupled และ Uncoupled ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล ในโครงข่าย MPTCP (บนเส้นทางโครงข่าย LTE และ WiFi โดยไม่มี ทราฟฟิค Background ทางโครงข่าย LTE และ WiFi) ด้วยการรับ-ส่งไฟล์ วิดีโอขนาด 229.4 MB ผลการศึกษาพบว่า ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล บนโครงข่ายหลายๆ เมื่อถึงขั้นตอนการรวมแพ็คเกจของข้อมูลที่ปลายทางจะต้องใช้เวลาในการรวม ข้อมูลทั้งหมดเพื่อจัดเรียงชุดข้อมูลใหม่ จึงส่งผลให้ผลของ TCP ฟังก์ชันที่มี Throughput มากสุด มี Throughput มากกว่าผลรวมของ MPTCP

ในการใช้งาน Uncoupled (CUBIC) บนโครงข่าย MPTCP ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลบนตั้งแต่ 3 เท่าขึ้นไป จะส่งผลให้ผลรวมที่มาจากค่า Sum of Throughput ของ MPTCP บนฝั่ง LTE และ WiFi มี Throughput ที่ต่ำกว่า TCP ฟังก์ชันที่มี Throughput มากสุด โดยค่าเฉลี่ยของ Throughput จะมีความห่างไกลจากค่า TCP ฟังก์ชันที่มี Throughput มากสุด และเพิ่มมากขึ้นตามอัตราความแตกต่างของการรับ-ส่งข้อมูล

ในการใช้งาน Coupled (wVegas) บนโครงข่าย MPTCP ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล จะให้ค่าผลรวม (Sum of Throughput) ของ MPTCP บนฝั่ง LTE และ WiFi มี Throughput ที่ใกล้เคียงกับค่า TCP ฟังก์ชันที่มี Throughput มากสุด

และจากงานวิจัยนี้พิสูจน์ให้เห็นว่าในกรณีของสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลในโครงข่าย MPTCP เมื่อมีความแตกต่างของอัตราการรับ-ส่งข้อมูลที่สูงเกิน 3 เท่าขึ้นไป การใช้งาน wVegas นั้น จะให้ Throughput และประสิทธิภาพของระบบที่ดีกว่า การใช้งาน Uncoupled CUBIC จึงสามารถสรุปได้ว่าการใช้งานวิธีกควบคุมความคับคั่งแบบ Coupled ให้ประสิทธิภาพของ Throughput ที่ดีกว่าแบบ Uncoupled

### 5.1.1 สรุปผลตามวัตถุประสงค์ของงานวิจัย

1. ได้ดำเนินการเปรียบเทียบประสิทธิภาพของโปรโตคอล TCP และ MPTCP ในกรณีที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูลในโครงข่าย LTE และ WiFi
2. ได้ทราบขอบเขตความต่างของอัตราการรับ-ส่งข้อมูลที่ทำให้ MPTCP มี Throughput โดยรวมน้อยกว่า TCP
3. ได้เปรียบเทียบประสิทธิภาพการทำงานของ Congestion Control ระหว่าง Uncoupled และ Coupled บนโปรโตคอล MPTCP ในโครงข่าย LTE และ WiFi ในกรณีที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล

### 5.1.2 สรุปผลตามขอบเขตของงานวิจัย สามารถสรุปผลได้ดังนี้

1. ทำการจำลองสถานการณ์การใช้งานโปรโตคอล MPTCP แบบ Couple และ Uncoupled บนโครงข่าย LTE และ WiFi ด้วยวิธีการส่งข้อมูลวีดีโอจากวีดีโอจริง บนซอฟต์แวร์ NS3 (DCE) Version 1.7
2. ดำเนินการเปรียบเทียบประสิทธิภาพของวิธควบคุมความคับคั่งแบบ Coupled และ Uncoupled ในสถานการณ์ที่มีความแตกต่างของอัตราการรับ-ส่งข้อมูล ในโครงข่าย MPTCP
3. จำลองสถานการณ์การใช้งานโปรโตคอล MPTCP แบบ Couple ด้วย wVegas และ Uncoupled ด้วย CUBIC
4. ดำเนินการเปรียบเทียบประสิทธิภาพของโปรโตคอล MPTCP แบบ Couple และ Uncoupled ที่ใช้งานบนโครงข่าย LTE และ WiFi โดยตัวแปรที่ใช้ในการวิเคราะห์ก็คือ Throughput (อัตราการรับ-ส่งข้อมูลบนโครงข่าย), Delay (อัตราล่าช้าต่อหน่วยเวลา), Jitter (อัตราความแปรปรวนของความล่าช้าของข้อมูล)

## 5.2 ข้อจำกัดและแนวทางแก้ไขของงานวิจัย

1. เนื่องจาก โปรแกรม NS3 เป็นโปรแกรมที่มีการพัฒนาอยู่เสมอ ซึ่งเป็นโปรแกรมที่ฟรีค่าใช้จ่าย (Opensource) ซึ่งต้องใช้เวลาในการศึกษาเป็นเวลานาน รวมถึงการเรียกใช้งานโมดูลหรือตัวแปรต่างๆ ให้เหมาะสมกับการใช้งานอีกด้วย
2. ในการวิเคราะห์ผลการจำลองด้วยโปรแกรม NS3 มีความยุ่งยาก เนื่องจากผลการจำลองจะเป็นไฟล์นามสกุล pcap ฉะนั้นจะต้อง export ออกมาเปิดด้วยโปรแกรม Wireshark และคำนวณด้วยโปรแกรม MS Excel เพื่อประเมินผลต่อไป

### 5.3 ข้อเสนอแนะและแนวทางการพัฒนาในอนาคต

เนื่องจาก โปรแกรม NS3 เป็นโปรแกรมที่มีการพัฒนาอยู่เสมอ ผู้วิจัยจึงจะขอเสนอแนะแนวทางในการพัฒนาต่อไป ดังนี้

1. ทดสอบระบบ MPTCP ผ่านระบบโครงข่าย LTE และ WiFi ที่สามารถทำงานได้หลายผู้ใช้งานในช่วงเวลาเดียวกัน เพื่อทดสอบว่าระบบสามารถรองรับการทำงานได้ โดยไม่ส่งผลกระทบต่อการใช้งาน
2. ทดสอบระบบ MPTCP ผ่านระบบโครงข่าย LTE และ WiFi โดยการจำลองที่โดยการทำการเคลื่อนไหวของผู้ใช้งาน โดยเคลื่อนที่ เข้า - ออก จากผู้ให้บริการ และกำหนดความเร็วในการเคลื่อนที่ว่ามีผลกระทบต่อการทำงาน MPTCP ระดับใด
3. ทดสอบ MPTCP ผ่านระบบโครงข่าย LTE และ WiFi โดยการจำลอง เปลี่ยน Congestion Control ที่รองรับการทำงาน MPTCP เช่น Lia, oLia, Westwood เป็นต้น



บรรณานุกรม

## บรรณานุกรม

### ภาษาไทย

ภาณุวง เมฆไพบุลย์ และธัญญ์ จารุวิทย์โกวิท. (2559). *Performance Evaluation of MPTCP Protocol over LTE and WiFi Network*. การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้ามหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่ 8 (น.) ภูเก็ต : มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์. รู้จักกับ *Multipath TCP* เข้าถึง 1 เมษายน 2562: <http://www.dpu.ac.th/mect/mptcp.html>

### ภาษาต่างประเทศ

*Balia (Balanced linked adaptation) A new MPTCP congestion control algorithm* เข้าถึง 1 เมษายน 2562 จาก <https://www.ietf.org/proceedings/90/slides/slides-90-mptcp-0.pdf>

Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, Olivier Bonaventure: Université catholique de Louvain, University Politehnica of Bucharest (August 2012) "*Exploring Mobile/WiFi Handover with Multipath TCP*" CellNet '12: Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design

Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure and Mark Handley: (2012) Universitatea Politehnica Bucuresti, †Université Catholique de Louvain Keio University, University College London " *How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP* " (April 2012) NSDI'12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation

Costin Raiciu, Dragos Niculescu, Marcelo Bagnulo, (June 2011) Mark Handley University Politehnica of Bucharest, Universidad Carlos III de Madrid, University College London "*Opportunistic Mobility with Multipath TCP*" MobiArch '11: Proceedings of the sixth international workshop on MobiArch

## บรรณานุกรม (ต่อ)

### ภาษาต่างประเทศ

*Direct Code Execution (DCE) Manual* เข้าถึง 1 มีนาคม 2562 จาก

<https://www.nsnam.org/docs/dce/release/1.4/manual/singlehtml/index.html>

Fa Fu, Xing Zhou, Thomas Dreibholz, Keying Wang, Feng Zhou and Quan Gan,(2015) ”

*Performance Comparison of Congestion Control Strategies for Multi-Path TCP in the NORNET Testbed*”, Hainan University, College of Information Science and Technology,

Haikou, China, IEEE/CIC ICC 2015 Symposium on Next Generation Networking

Gulshan Amin Gilkar, Syed Yasmeen Sahdad Department of CSE Department of CSE SITM. “TCP

*CUBIC- Congestion Control Transport Protocol*”

*How to calculate LTE throughput* เข้าถึง 1 เมษายน 2562 จาก

[http://anisimoff.org/eng/lte\\_throughput.html](http://anisimoff.org/eng/lte_throughput.html)

Karl-Johan Grinnemo and Anna Brunstrom, 2015 , “*A First Study on Using MPTCP to Reduce*

*Latency for Cloud Based Mobile Applications*”, 7th IEEE International Workshop on

Performance Evaluation of Communications in Distributed Systems and Web based

Service Architectures

*LTE Network Architecture: Basic* เข้าถึง 1 เมษายน 2562 จาก

<http://www.netmanias.com/en/post/techdocs/5904/architecture-lte/lte-network-architecture-basic>

*MCS : Index* เข้าถึง 15 มีนาคม 2562 จาก

<http://mcsindex.com>

*Multipath TCP speeds up the internet so much that security breaks* เข้าถึง 1 เมษายน 2562 จาก

[http://www.theregister.co.uk/2014/07/31/multipath\\_tcp\\_will\\_bork\\_your\\_network\\_probes\\_flummo\\_x\\_your\\_firewalls/](http://www.theregister.co.uk/2014/07/31/multipath_tcp_will_bork_your_network_probes_flummo_x_your_firewalls/)

*Multipath TCP* เข้าถึง 1 เมษายน 2562 จาก

<https://queue.acm.org/detail.cfm?id=2591369>

## บรรณานุกรม (ต่อ)

### ภาษาต่างประเทศ

*ns-3, tcp Directory Reference* เข้าถึง 1 มีนาคม 2562 จาก

[https://www.nsnam.org/doxygen/dir\\_600cab24fe22a606254421806d2e08fa.html](https://www.nsnam.org/doxygen/dir_600cab24fe22a606254421806d2e08fa.html)

*ns-3 Documentation* เข้าถึง 1 มีนาคม 2562 จาก

<https://www.nsnam.org/doxygen/index.html>

Sangtae Ha, Injong Rhee,(2008) Dept of Computer Science North Carolina State University Raleigh, NC 27695 and Lisong Xu, Dept of Comp. Sci. and Eng.University of Nebraska Lincoln, Nebraska 68588, “*CUBIC: A New TCPFriendly HighSpeed TCP Variant*”

Wenlan Guo<sup>1</sup>, Zhijia Wang, Yun Zhang, (2013) School of Computer Science and Technology,Harbin Univ.Sci.Tech.,Harbin,China " *Delay-based Congestion Control for Multipath TCP* "

Wireshark Tip 22: Find Delays with TCP "Calculate Conversation Timestamps" เข้าถึง 10 เมษายน 2562 จาก <https://www.youtube.com/watch?v=QqKAnZnHss0>

Yu Cao<sup>\*t</sup>, Mingwei Xu<sup>\*+</sup>, Xiaoming Fu<sup>§</sup>,(2012) “*Delay-based Congestion Control for Multipath TCP*”, <sup>\*</sup>Tsinghua University, <sup>t</sup>National Data Switch Center, China Tsinghua National Laboratory for Information Science and Technology (TNList), Institute of Computer Science, University of Goettingen



### ประวัติผู้เขียน

ชื่อ-นามสกุล

นางสาวพกาศิตา ศรีภักดี

ประวัติการศึกษา

ปริญญาตรี สำนักวิชาวิศวกรรมศาสตร์และทรัพยากร  
มหาวิทยาลัยวลัยลักษณ์ ปีการศึกษา 2551

ตำแหน่งและสถานที่ทำงานปัจจุบัน

วิศวกร

หน่วยธุรกิจคลาวด์และดิจิทัล

บริษัท ทีไอที จำกัด (มหาชน) (สำนักงานใหญ่)

