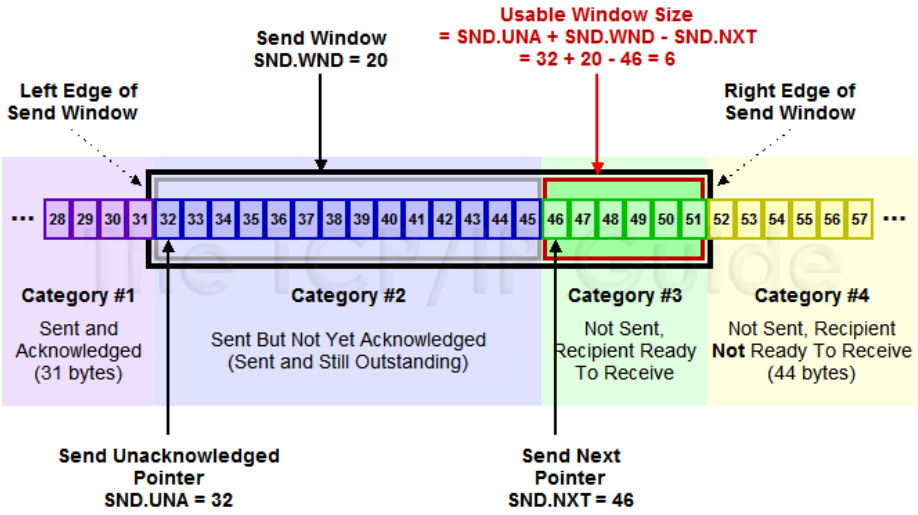
นายสิริทรัพย์ สิริพัฒนากุล sec 3 6301097 8

กิจกรรมที่ 8 : TCP Window

กิจกรรมครั้งนี้จะเป็นการทำความเข้าใจกับโปรโตคอล TCP (Transmission Control Protocol) ให้มากยิ่งขึ้น โดยเน้นเรื่องของ TCP Window โดย TCP Window จะแบ่งออกเป็น send Window และ receive Window

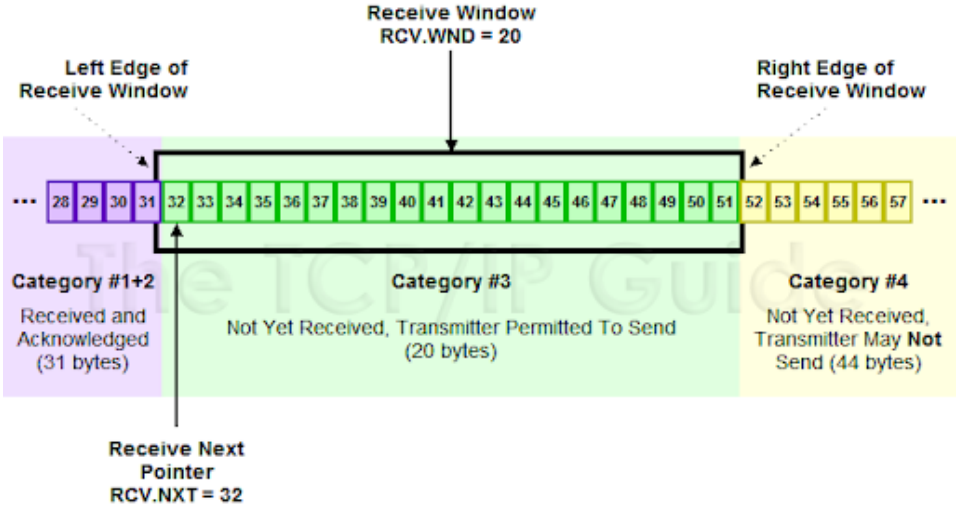
ใน send window จะแบ่งออกเป็น 4 ส่วน คือ

- ข้อมูลที่ส่งแล้วและได้รับ Acknowledge ไปแล้ว
- ข้อมูลที่ส่งไปแล้วแต่ยังไม่ได้รับ Acknowledge (ใน Wireshark จะเรียกว่า byte in flight)
- ข้อมูลที่ยังไม่ได้ส่ง และ ฝั่งรับสามารถรับได้ (ตามขนาดของ receive window)
- ข้อมูลที่ยังไม่ได้ส่ง และ ฝั่งรับไม่พร้อมจะรับเนื่องจากขนาดของ receive window

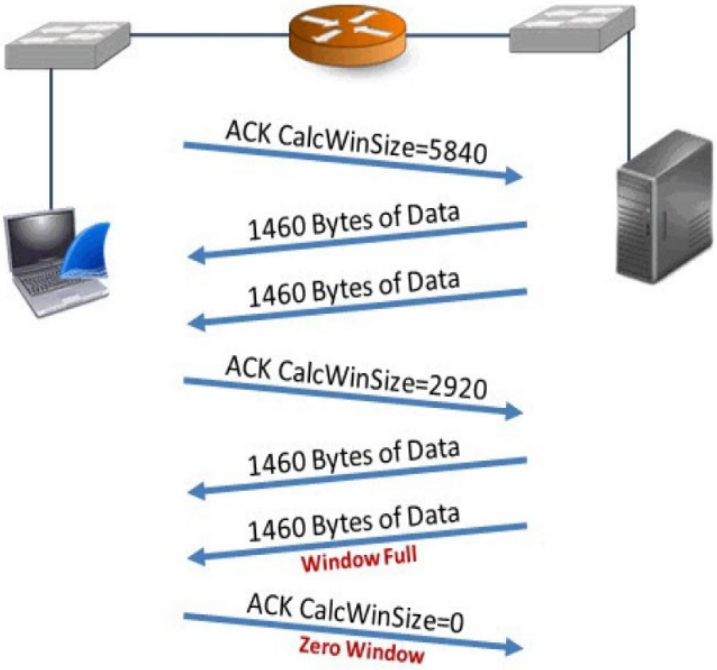


ใน receive window จะแบ่งเป็น 2 ส่วน

- ข้อมูลที่รับแล้วและ Acknowledge ไปแล้ว
- ข้อมูลพร้อมจะรับ



ในระหว่างการสื่อสารทั้ง 2 ด้านจะมีการแจ้งขนาดของ window size ที่เหลือที่ยังรับข้อมูลได้มาใน header ของ TCP โดยมีขนาด 2 ไบต์ โดยมีค่าสูงสุด คือ 65,535 ไบต์ โดยมี Scaling Factor เป็นตัวคูณ ซึ่งหากฝั่งรับไม่สามารถนำข้อมูลออกจาก receive window ได้เร็วพอจะทำให้ Buffer เต็มและเกิด zero window ตามรูป (หมายเหตุข้อมูล window full และ zero window นี้เป็นข้อมูลที่ wireshark สร้างขึ้น เพื่อให้สะดวกต่อการใช้งาน)



1. ให้เปิดไฟล์ tr-youtubebad.pcapng จากนั้นให้ค้นหาเหตุการณ์ zero window โดยใช้ display filter tcp.analysis.zero_window จะเห็นว่ามี zero window เกิดขึ้นจำนวนมาก ให้เลือกบรรทัดแรก แล้วยกเลิก filter โปรแกรม wireshark จะแสดงบริเวณ packet ที่เกิด zero window ครั้งแรก ให้ขยาย TCP หาฟิลด์ calculated window size แล้วสร้างเป็นคอลัมน์ โดยกำหนดให้ Align Center และตั้งชื่อเป็น **WinSize**
   - ให้สังเกตที่ packet 2718 ซึ่งเป็น packet ที่ host 24.4.7.217 ส่ง ACK กลับมา โดยมี window size เหลือเพียง 1,460 ไบต์
   - ต่อมาใน packet 2719 host 208.117.232.102 มีการส่งข้อมูลไปอีก 1,460 ไบต์ ซึ่งจะทำให้เต็ม receive window พอดี และทำให้ wireshark สร้างข้อมูลแจ้งเตือนว่า window full
   - เมื่อถึง Packet 2720 host 24.4.7.217 ก็ส่ง Packet ACK กลับมา โดยมีค่า window size เป็น 0 ทำให้ wireshark สร้างข้อมูลแจ้งเตือนว่า zero window
   - ให้สังเกตช่วงเวลาระหว่าง packet 2720 และ 2721 จะเห็นว่ามีระยะห่างมากกว่าปกติ หมายความว่าฝั่งผู้ส่งเมื่อพบ zero window ก็จะรอฝั่งผู้รับให้เคลียร์ receive window เสียก่อน
   - ใน packet 2721 จะมีการส่ง packet keep alive (คือ packet ACK ที่ไม่มีข้อมูล จากฝั่งผู้ส่ง ซึ่งจะเกิดขึ้นเมื่อ keepalive timer expire)
   - จากนั้นใน packet 2722 ผู้รับจะส่ง ACK กลับมา โดยมี window size เป็น 0 เช่นเดิม และเกิดซ้ำอีกครั้งใน packet 2723 และ 2724

- จนกระทั่ง packet 2725 ฝั่งผู้รับจึงส่ง packet ACK ซึ่งมีขนาดของ window size = 243820 ซึ่งไม่เท่ากับ 0 ซึ่งหมายความว่า receive window ของฝั่งผู้รับว่างแล้ว พร้อมรับข้อมูลใหม่ ณ จุดนี้ ถือว่าเหตุการณ์ zero window สิ้นสุดลง โดย wireshark จะสร้างข้อมูลแจ้งเตือน window update

2. ให้นักศึกษาตรวจสอบ zero window ระยะที่ 2 แล้วตอบคำถาม ต่อไปนี้

- เกิด window full, zero window (เฉพาะครั้งแรก) และ window update ที่ packet ใด

window full ที่ packet 4022 | window update ที่ packet 4036

window zero (ครั้งแรก) ที่ packet 4023

- หลังจากมีการทำ keep alive กี่ครั้ง มีช่วงระยะเวลาเท่าไรบ้าง นับจาก zero window ครั้งก่อน ให้แสดงรูป capture จาก wireshark ที่แสดงเวลาของ keep alive แต่ละครั้ง มาด้วยใน 1 รูป

มีการทำ keep alive 6 ครั้ง มีช่วงระยะเวลา 0.477622 0.995377 1.878101 3.704824 7.398856 และ 10.020053 วินาที ตามลำดับ

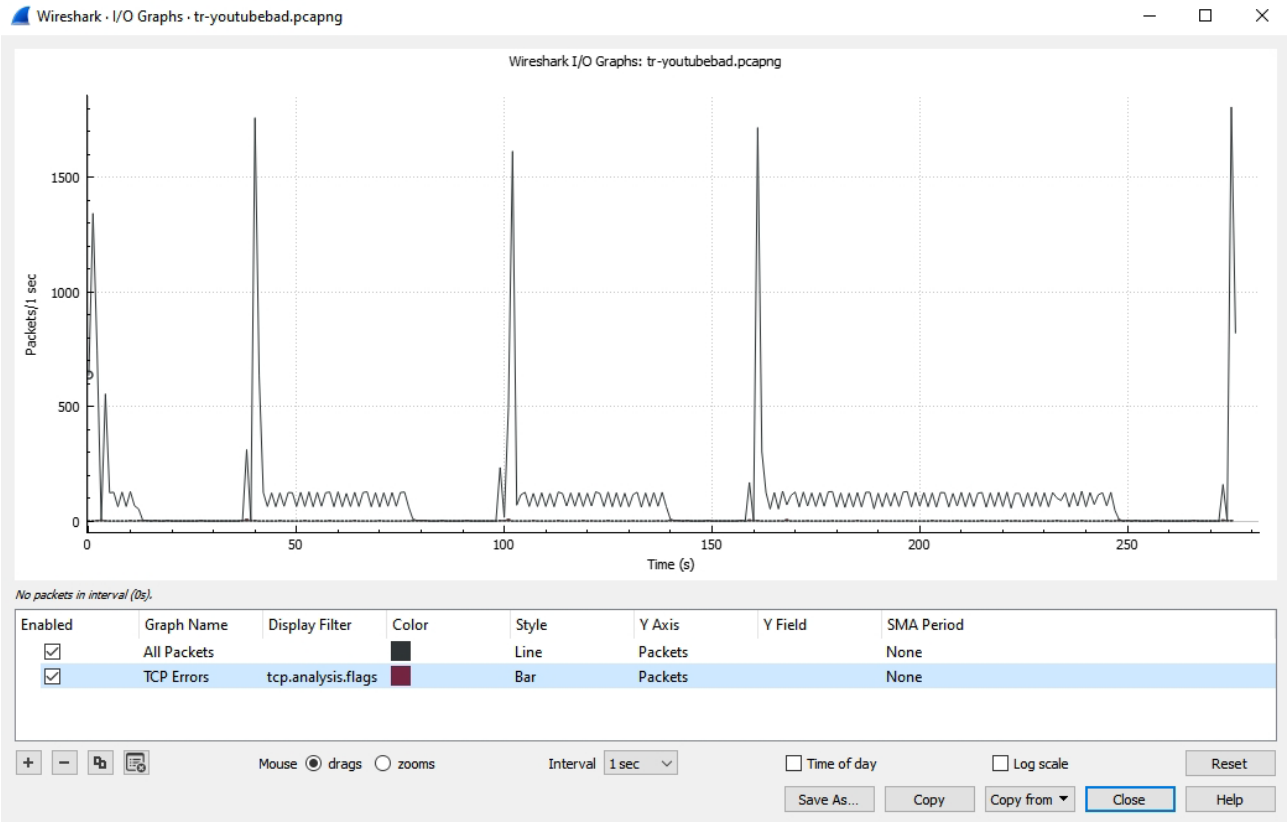
Toolbar → View → Time Display Format → Second Since Previous Displayed Packet

No.	Time	Source	Destination	Protocol	Length	#SEQ	#NEXTSEQ	#ACK	TCP Segment Len	WinSize	Info
4020	0.000765	208.117.232.102	24.4.7.217	TCP	1514	4246662	4248122	1270	1460	8384	80 → 56770 [ACK] Seq=4246662 Ack=1270 Win=8384 Len=1460 [TCP segment of a reassembled P
4021	0.200685	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248122	0	328	56770 → 80 [ACK] Seq=1270 Ack=4248122 Win=328 Len=0
4022	0.362283	208.117.232.102	24.4.7.217	TCP	382	4248122	4248450	1270	328	8384	[TCP Window Full] 80 → 56770 [PSH, ACK] Seq=4248122 Ack=1270 Win=8384 Len=328 [TCP segm
4023	0.209752	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4024	0.477622	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	0	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4025	0.000046	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4026	0.995377	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	0	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4027	0.000057	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4028	1.878101	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	0	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4029	0.000063	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4030	3.704824	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	0	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4031	0.000141	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4032	7.398856	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	0	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4033	0.000100	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4034	10.020053	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	0	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4035	0.000052	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4036	0.954932	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	166440	[TCP Window Update] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=166440 Len=0
4037	0.018973	208.117.232.102	24.4.7.217	TCP	1514	4248450	4249910	1270	1460	8384	80 → 56770 [ACK] Seq=4248450 Ack=1270 Win=8384 Len=1460 [TCP segment of a reassembled P

- ระยะเวลาตั้งแต่เกิด zero window ครั้งแรกจนถึง window update ใช้เวลาเท่าไร

  คลิกขวาที่ packet 4023 → Set/Unset Time Reference → ค่า Time ที่ packet 4036 = 25.430224 วินาที

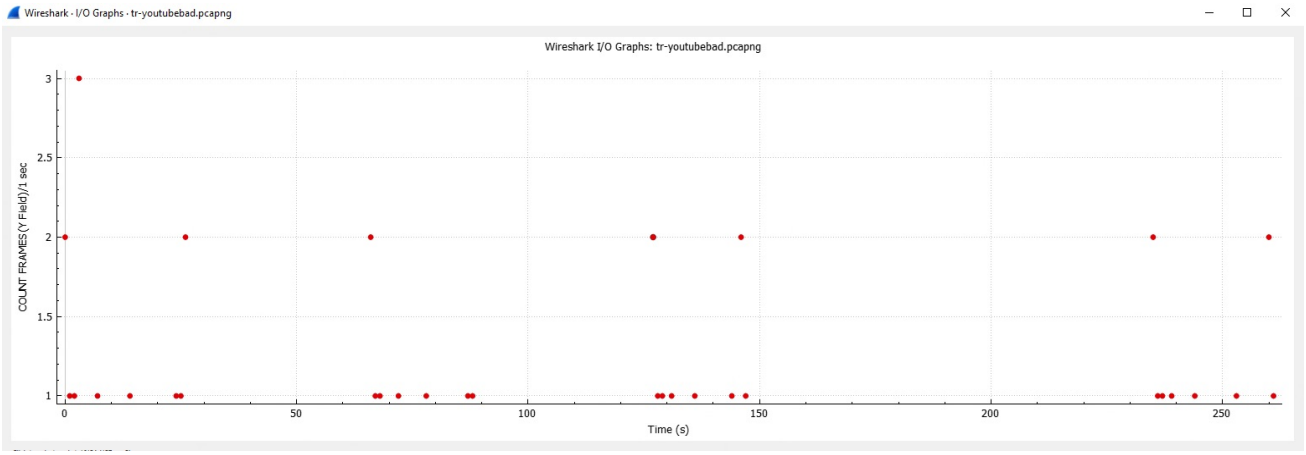
3. การวิเคราะห์ข้อมูลนอกจากจะทำในหน้าต่าง Packet List และ Packet Detail แล้ว ใน wireshark ยังให้เครื่องมือประเภทกราฟมาด้วย จากไฟล์เดิม ให้นักศึกษาเรียกเมนู Statistics | I/O Graph จะปรากฏหน้าจอดังนี้



- ข้อมูลแกน Y คือ packet/sec แกน x คือเวลา ซึ่งจะเห็นว่าข้อมูลมีการส่งได้ดี (กราฟพุ่งสูง จำนวน 5 ครั้ง) จากนั้นก็ลดลงอย่างมาก
- ในช่องด้านล่าง เราสามารถสร้างกราฟขึ้นมาใหม่ได้ ให้กด + แล้วกำหนดข้อมูลดังนี้
  - Graph Name : Zero_Window
  - Display filter : ว่าง
  - Color : แดง
  - Style : Dot
  - Y Axis : COUNT FRAMES(Y Field)
  - Y Field : tcp.analysis.zero_window
- ให้ Disable กราฟเดิมทั้ง 2 กราฟ

- กราฟบอกข้อมูลอะไร (แสดงรูป capture ของกราฟด้วย)

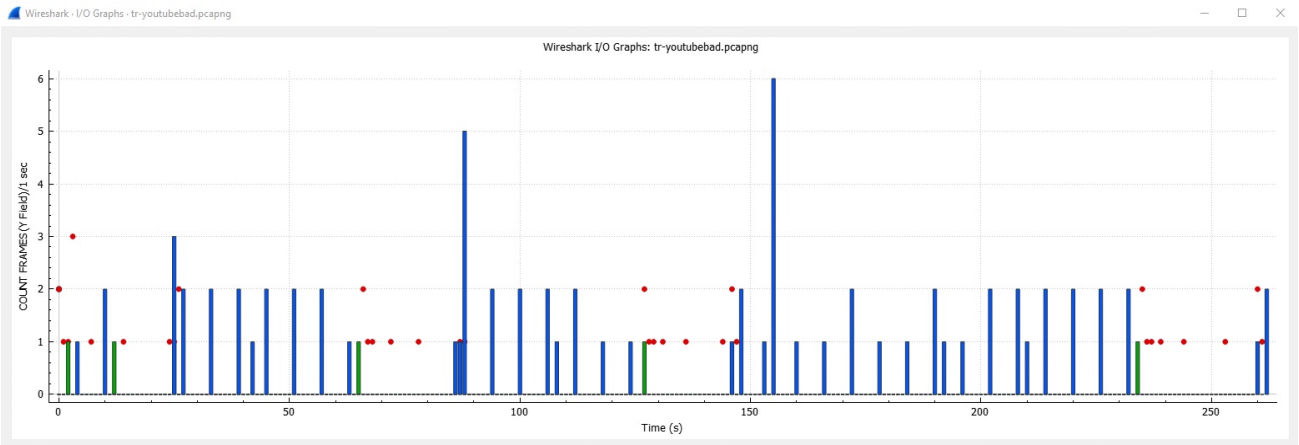
แกน y คือ จำนวน packet ที่เกิด zero window / 1 วินาที แกน x คือ เวลา



4. ให้สร้างกราฟเพิ่มอีก 2 กราฟ ดังนี้
   - ชื่อ Window_Full โดยใน Y(AXIS) ใช้ COUNT FRAMES(Y Field) และช่อง Y Field ใช้ tcp.analysis.window_full กำหนดประเภทเป็น Bar สีเขียว
   - ชื่อ Window_Update โดยใน Y(AXIS) ใช้ COUNT FRAMES(*) และช่อง Y Field ใช้ tcp.analysis.window_update กำหนดประเภทเป็น Bar สีน้ำเงิน
   - กราฟแสดงอะไร (แสดงรูป capture ของกราฟด้วย)

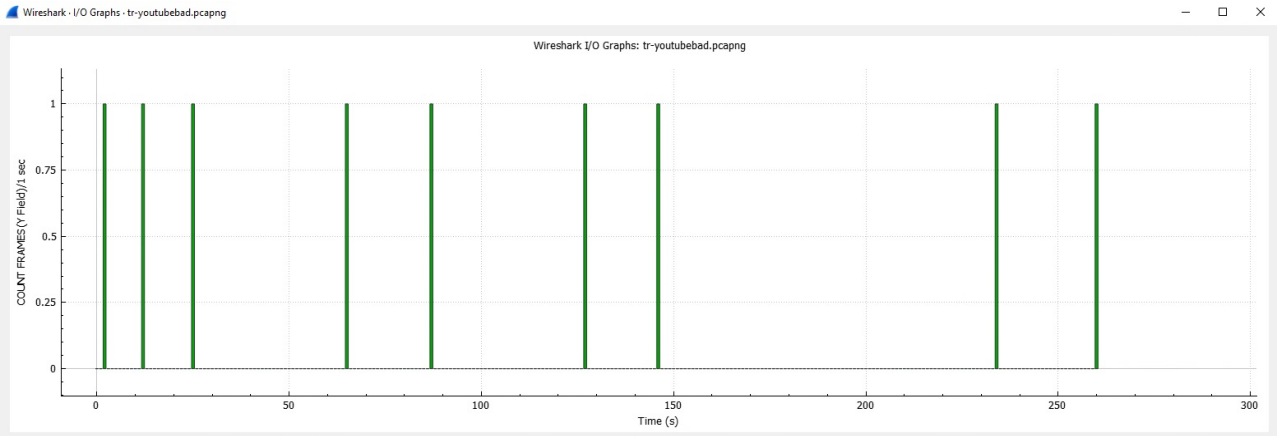
แกน y คือ จำนวน packet ที่เกิด window full และ window update / 1 วินาที

แกน x คือ เวลา



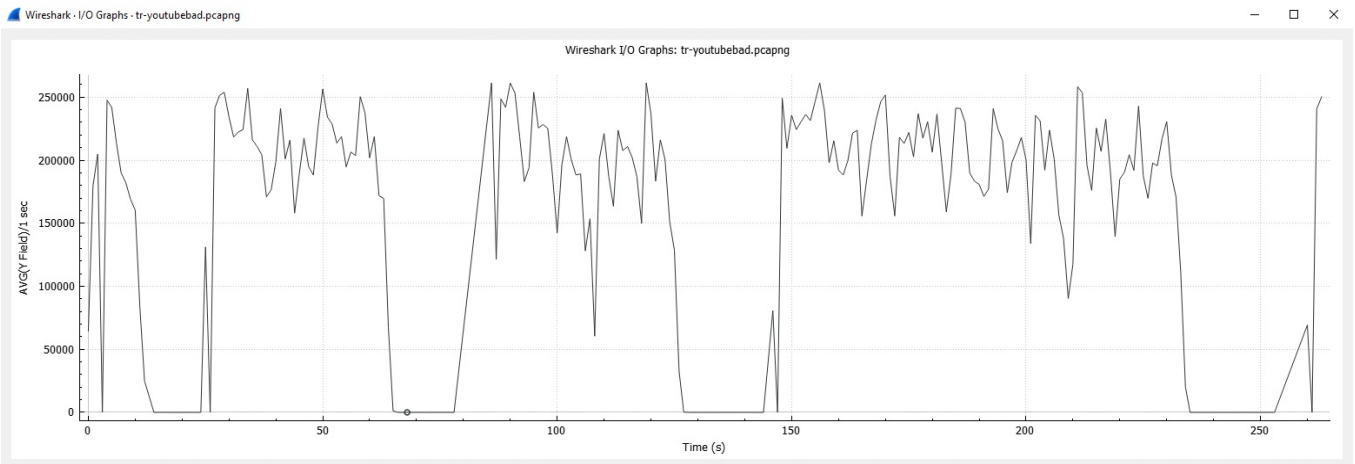
- จากกราฟสามารถบอกได้หรือไม่ว่ามี window full กี่ครั้ง ให้ Capture รูปประกอบด้วย

มี window full 9 ครั้ง



5. ให้สร้าง I/O Graph ใหม่ โดยในช่อง Display Filter ให้ใส่ ip.src==24.4.7.217 ใน Y(AXIS) ใช้ AVG(*) และช่อง Y Field ใช้ tcp.window_size กำหนดประเภทเป็น Line ให้ capture รูป และ อธิบายว่าเราสามารถวิเคราะห์ข้อมูลอะไรจากกราฟนี้ ให้ Capture รูปประกอบด้วย

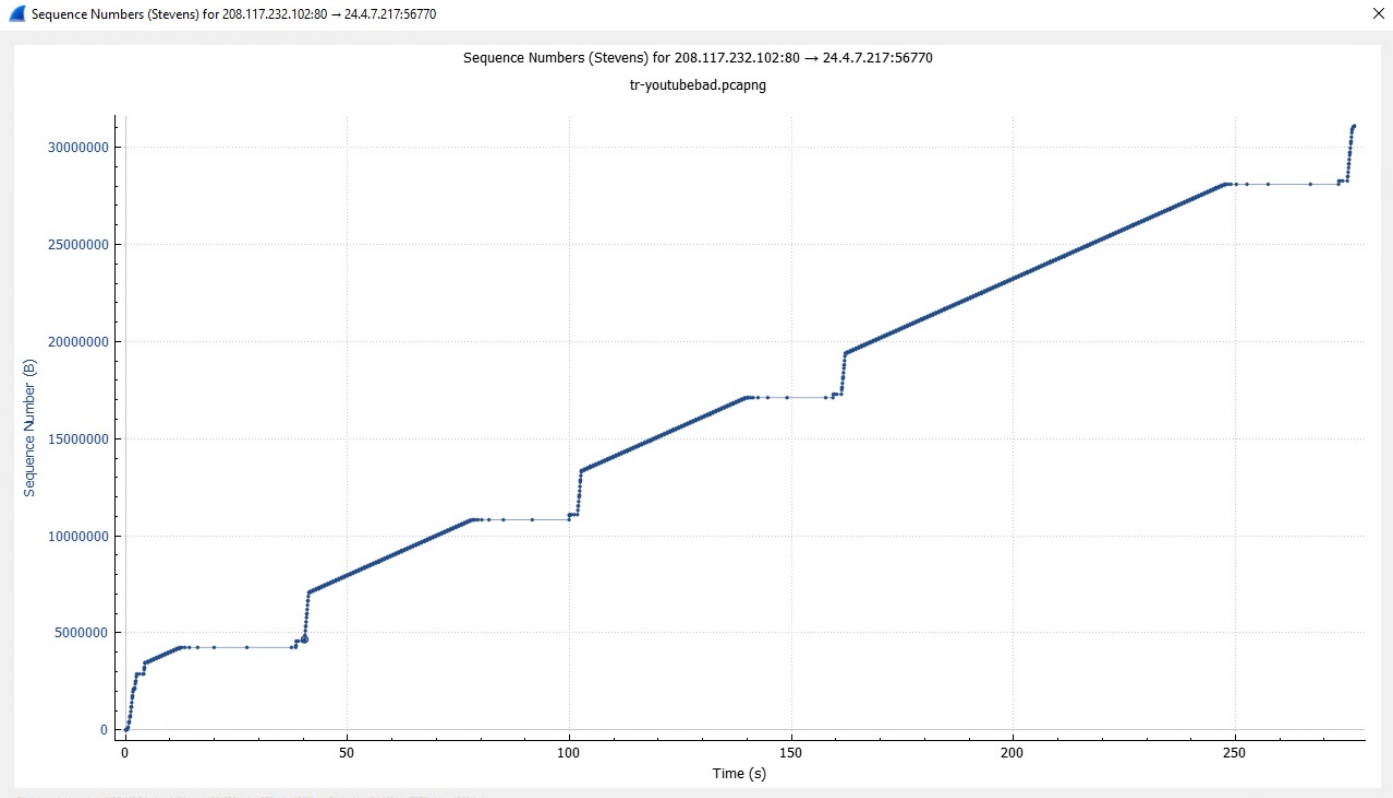
แกน Y คือ ค่าเฉลี่ยของ window size ของ packet ที่ถูกส่งมาจาก IP 24.4.7.217 แกน X คือ เวลา สามารถวิเคราะห์เวลาการเกิด zero window ได้



6. ในการควบคุม congestion control ของ TCP จะมีหลักอยู่ 2 ข้อ คือ Slow Start และ Congestion Avoidance ให้เปิดไฟล์ tcp.pcapng แล้วดูที่ Statistics->TCP Stream Graph-> Time-Sequence-Graph(Stevens) โดยแต่ละจุดแสดงถึงการส่งในแต่ละ segment ร่วมกับ Statistics-> Flow Graph นักศึกษาสามารถบอกได้หรือไม่ว่า Slow Start เริ่มต้นและสิ้นสุดที่ใด และมี Congestion Avoidance เกิดขึ้นหรือไม่ ให้อธิบาย พร้อมรูปประกอบ

Slow Start → ช่วงที่ค่าในแกน Y เกือบขนานกับแกน เวลา คือ packet ส่งในระยะเวลาห่างกัน

Congestion Avoidance → ช่วงที่ค่าในแกน Y เพิ่มขึ้นแบบ exponential



งานครั้งที่ 8

- การส่งงาน เขียนหรือพิมพ์ลงในเอกสารนี้ และส่งโดยเป็นไฟล์ PDF เท่านั้น
- ตั้งชื่อไฟล์โดยใช้รหัสนักศึกษา และ _Lab8 เช่น 63010789_Lab6.pdf
- กำหนดส่ง ภายในวันที่ 23 มีนาคม 2565