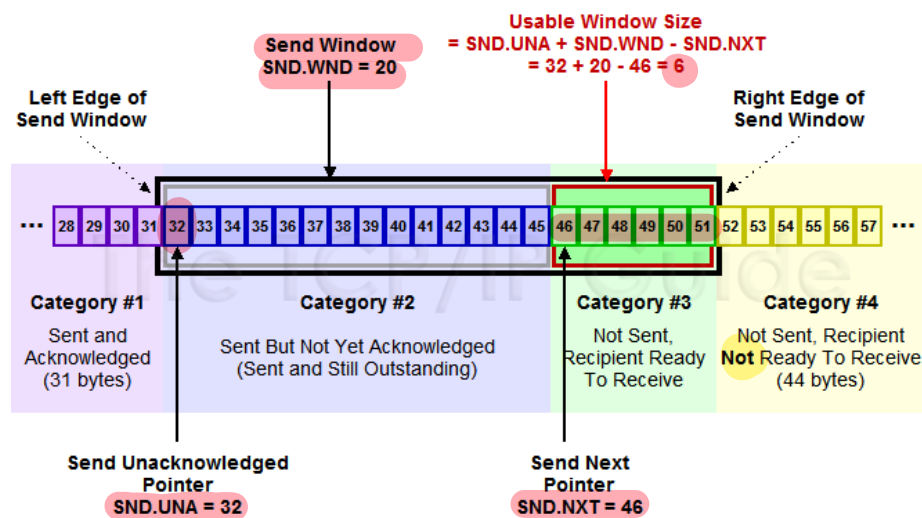


## กิจกรรมที่ 8 : TCP Window

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

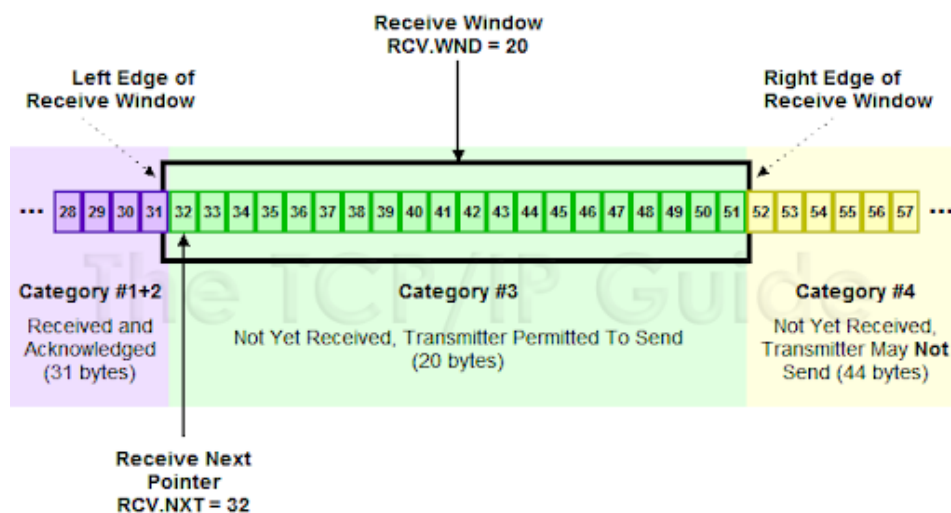
ใน send window จะแบ่งออกเป็น 4 ส่วน คือ

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

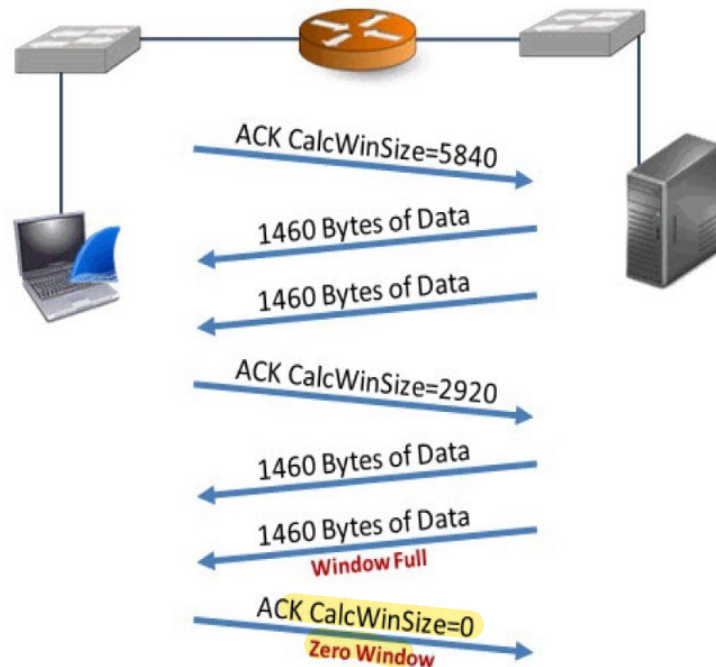


ใน receive window จะแบ่งเป็น 2 ส่วน

- ข้อมูลที่รับแล้วและ Acknowledge ไปแล้ว
- ข้อมูลพร้อมจะรับ



ในระหว่างการสื่อสารทั้ง 2 ด้านจะมีการแจ้งขนาดของ window size ที่เหลือที่ยังรับข้อมูลได้ใน 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 แล้วคลิก filter โปรแกรม wireshark จะแสดงบริเวณ packet ที่เกิด zero window ครั้งแรก ให้ขยาย TCP header field **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 ก็**จะรอฝั่งผู้รับให้เคลียร์ 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 ฟังผู้รับจึงส่ง packet ACK ซึ่งมีขนาดของ window size = 243820 ซึ่งไม่เท่ากับ 0 ซึ่งหมายความว่า receive window ของฝั่งผู้รับว่างแล้ว พร้อมรับข้อมูลใหม่ ณ จุดนี้ ถือว่าเหตุการณ์ zero window สิ้นสุดลง โดย wireshark จะสร้างข้อมูลแจ้งเตือน window update

packet ที่ 4022

2. ให้นักศึกษาตรวจสอบ zero window ระยะเวลาที่ 2 แล้วตอบคำถาม ต่อไปนี้

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

window full ; packet ที่ 4022      window update ; packet ที่ 4026

zero window ; packet ที่ 4023

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

3 keep alive 6 ครั้ง

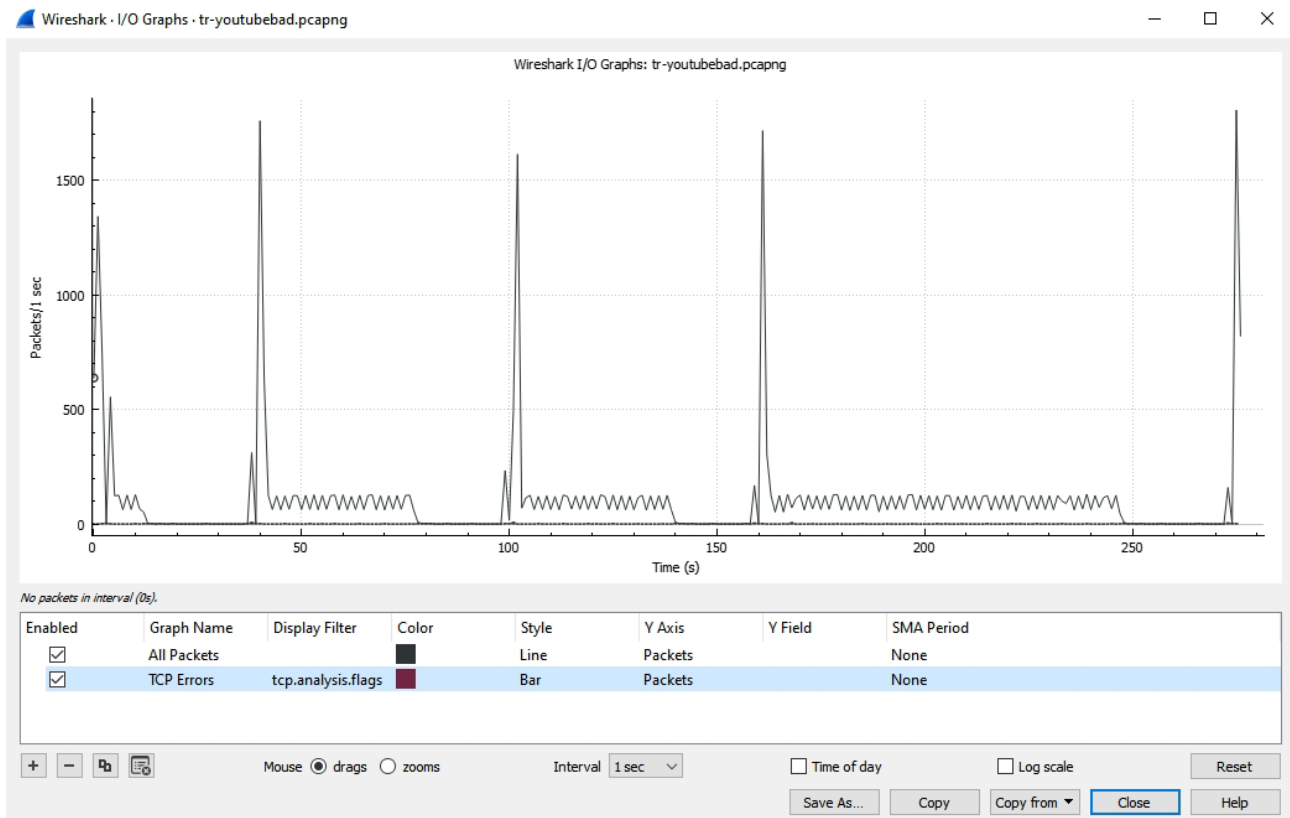
เวลา 0.422 , 0.995 , 1.476 , 2.704 , 7.396 , 10.02 นาที

4022	12.6792	208.117.232.102	24.4.7.217	TCP	382	4248122	4248450	1270	8384	0.362283000	[TCP Window Full] 80 → 56770 [PSH, A
4023	12.8890	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0.209752000	[TCP ZeroWindow] 56770 → 80 [ACK] Se
4024	13.3666	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	0.477622000	[TCP Keep-Alive] 80 → 56770 [ACK] Se
4025	13.3666	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0.000046000	[TCP ZeroWindow] 56770 → 80 [ACK] Se
4026	14.3620	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	0.995377000	[TCP Keep-Alive] 80 → 56770 [ACK] Se
4027	14.3621	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0.000057000	[TCP ZeroWindow] 56770 → 80 [ACK] Se
4028	16.2402	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	1.878101000	[TCP Keep-Alive] 80 → 56770 [ACK] Se
4029	16.2402	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0.000063000	[TCP ZeroWindow] 56770 → 80 [ACK] Se
4030	19.9451	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	3.704824000	[TCP Keep-Alive] 80 → 56770 [ACK] Se
4031	19.9452	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0.000141000	[TCP ZeroWindow] 56770 → 80 [ACK] Se
4032	27.3441	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	7.398856000	[TCP Keep-Alive] 80 → 56770 [ACK] Se
4033	27.3442	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0.000100000	[TCP ZeroWindow] 56770 → 80 [ACK] Se
4034	37.3642	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	10.020053000	[TCP Keep-Alive] 80 → 56770 [ACK] Se
4035	37.3643	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	0.000052000	[TCP ZeroWindow] 56770 → 80 [ACK] Se
4036	38.3192	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	166440	0.954932000	[TCP Window Update] 56770 → 80 [ACK]

- ระยะเวลาตั้งแต่เกิด zero window ครั้งแรกจนถึง window update ใช้เวลาเท่าไร

ใช้เวลา 95.4302 sec

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



- ข้อมูลแกน Y คือ packet/sec แกน x คือเวลา ซึ่งจะเห็นว่าข้อมูลมีการส่งได้ดี (กราฟพุ่งสูง จำนวน 5 ครั้ง) จากนั้นก็ลดลงอย่างมาก

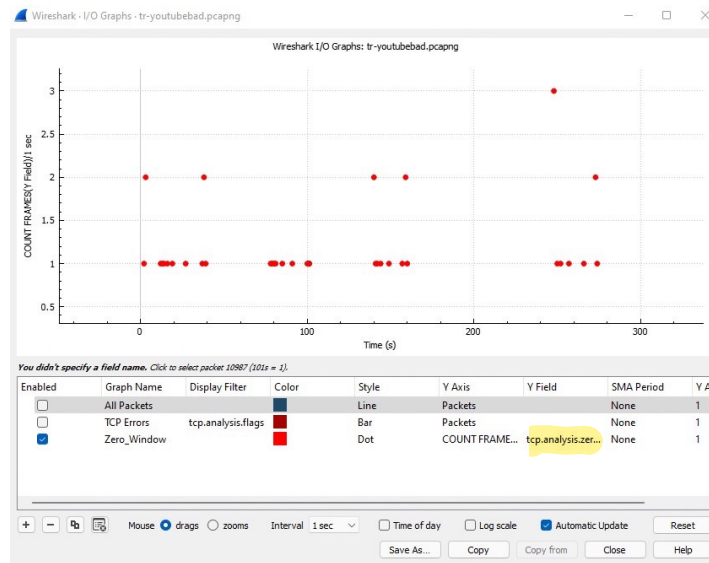
- ในช่องด้านล่าง เราสามารถสร้างกราฟขึ้นมาใหม่ได้ ให้กด + แล้วกำหนดข้อมูลดังนี้

- 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)

กราฟแสดง zero-window ในรูปแบบ Dot ตามช่วงเวลาที่

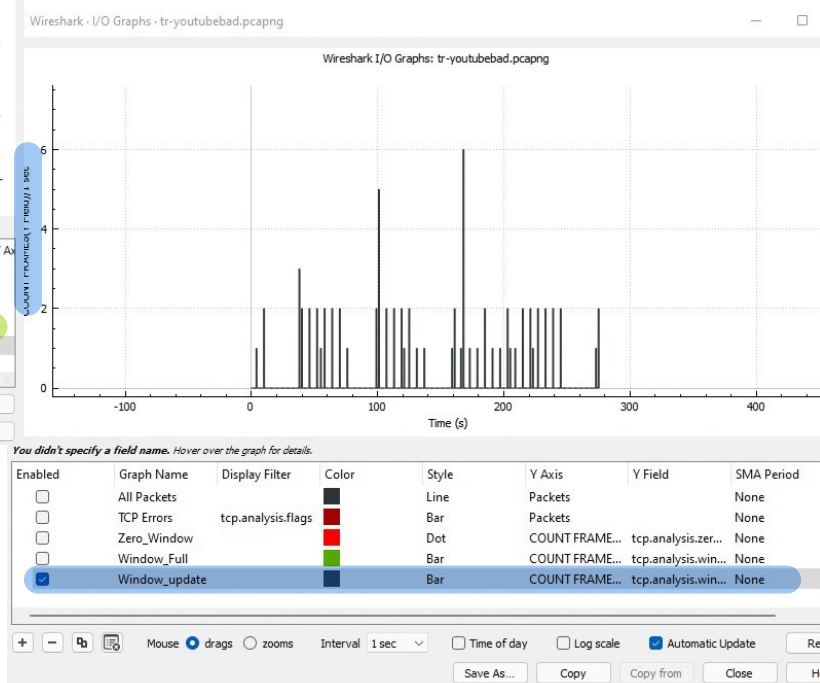
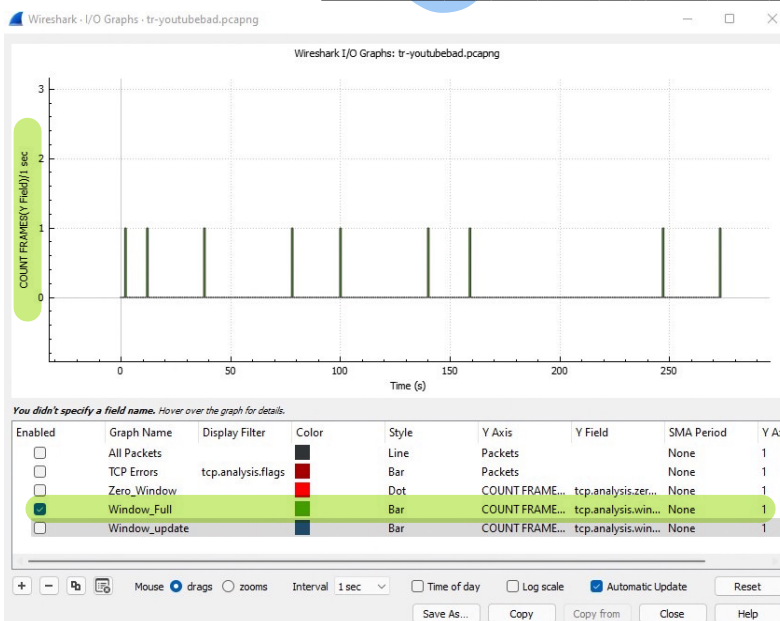


- ให้สร้างกราฟเพิ่มอีก 2 กราฟ ดังนี้

- ✓ 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 สีนํ้าเงิน
- กราฟแสดงอะไร (แสดงรูป capture ของกราฟด้วย)

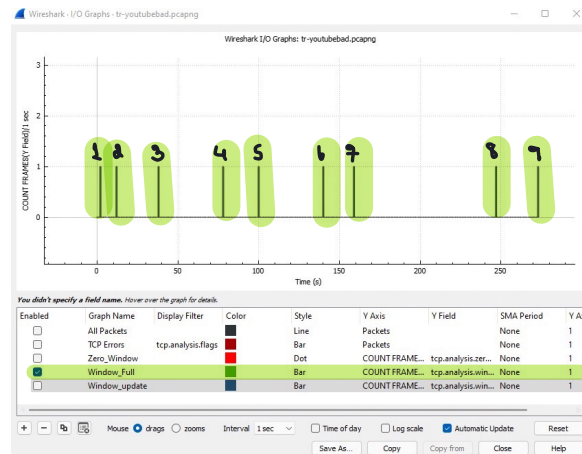
กราฟเขียวแสดง จำนวนครั้ง window full ต่อ 1 sec

กราฟฟ้าแสดง จำนวนครั้ง window-update ต่อ 1 sec



- จากกราฟสามารถบอกได้หรือไม่ว่ามี window full ที่ครั้ง ให้ Capture รูปประกอบด้วย

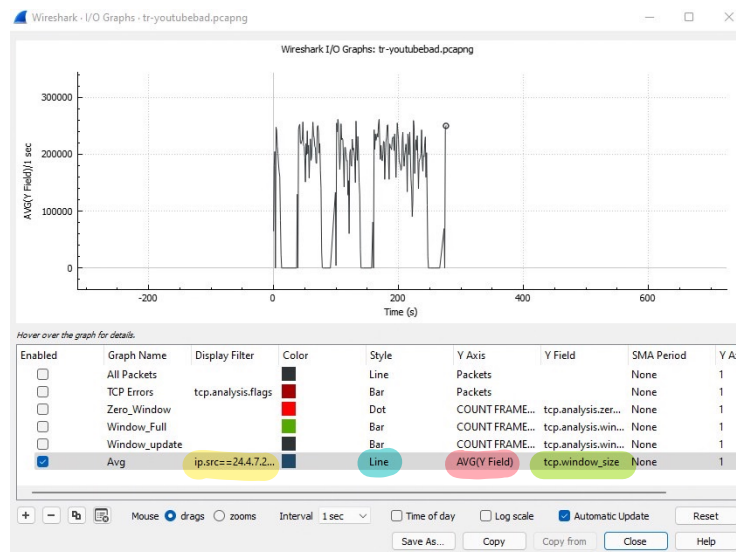
ลองหาให้ได้ มี window full ทั้งหมด ๗ ครั้ง



← มีทั้งหมด ๗ 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

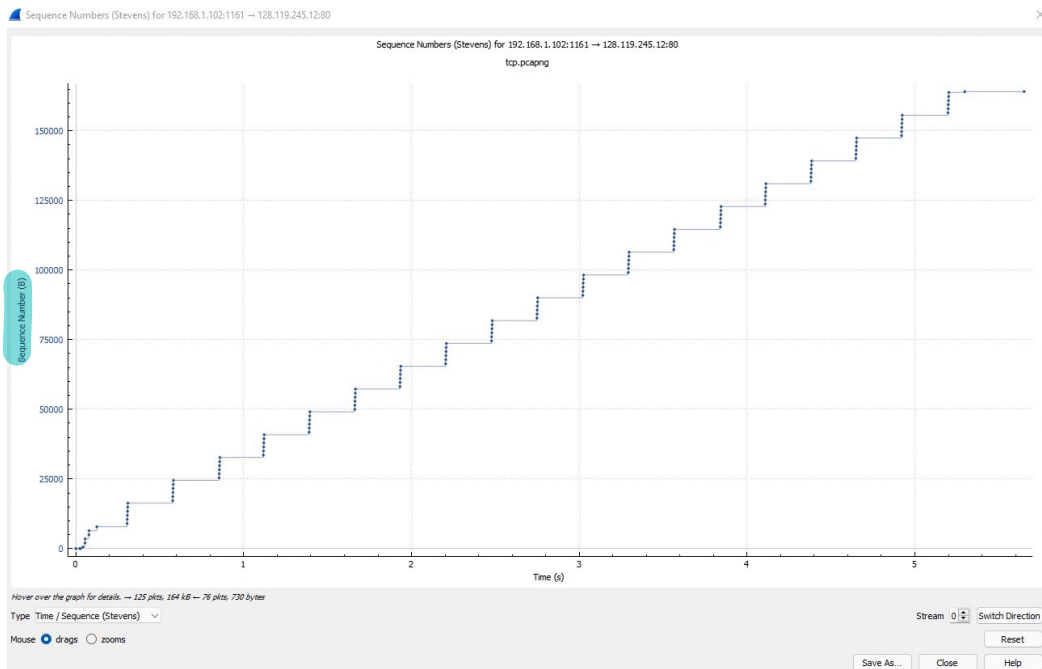
อัตราเน็ตเวิร์กประมาณ window size ตั้งแต่ 0 ถึง 300,000 เมื่อกราฟแสดงออกมา  
 ข้อสังเกต window size = 0 เมื่อ window update จะเกิดขึ้นในแนวตั้งไปประมาณ 260,000



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



1. slow start เริ่มส่ง packet ที่ 2 และ 1460 bytes (Congestion window)  
 2. มีค่า window size (1460 byte) ขยับได้ 62,780 bytes  
 3. ค่าของ window size ไม่ใช่ว่า Congestion Avoidance หมดแล้ว



Time	192.168.1.102	128.119.245.12	Intel_52:2b:23	Broadcast	Comment
0.000000	1161	1161 -> 80 [SYN] Seq=0 Win=16384 Len=0	80		TCP: 1161 -> 80 [SYN] Seq=0 Win=16384 Len=0 ...
0.023172	1161	80 -> 1161 [SYN, ACK] Seq=0 Ack=1 Win=...	80		TCP: 80 -> 1161 [SYN, ACK] Seq=0 Ack=1 Win=...
0.023265	1161	1161 -> 80 [ACK] Seq=1 Ack=1 Win=17520...	80		TCP: 1161 -> 80 [ACK] Seq=1 Ack=1 Win=17520...
0.026477	1161	1161 -> 80 [PSH, ACK] Seq=1 Ack=1 Win=...	80		TCP: 1161 -> 80 [PSH, ACK] Seq=1 Ack=1 Win=...
0.041737	1161	1161 -> 80 [PSH, ACK] Seq=566 Ack=1 Win=...	80		TCP: 1161 -> 80 [PSH, ACK] Seq=566 Ack=1 Win=...
0.053937	1161	80 -> 1161 [ACK] Seq=1 Ack=566 Win=678...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=566 Win=678...
0.054026	1161	1161 -> 80 [ACK] Seq=2026 Ack=1 Win=17...	80		TCP: 1161 -> 80 [ACK] Seq=2026 Ack=1 Win=17...
0.054690	1161	1161 -> 80 [ACK] Seq=3486 Ack=1 Win=17...	80		TCP: 1161 -> 80 [ACK] Seq=3486 Ack=1 Win=17...
0.077294	1161	80 -> 1161 [ACK] Seq=1 Ack=2026 Win=87...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=2026 Win=87...
0.077405	1161	1161 -> 80 [ACK] Seq=4946 Ack=1 Win=17...	80		TCP: 1161 -> 80 [ACK] Seq=4946 Ack=1 Win=17...
0.078157	1161	1161 -> 80 [ACK] Seq=6406 Ack=1 Win=17...	80		TCP: 1161 -> 80 [ACK] Seq=6406 Ack=1 Win=17...
0.124085	1161	80 -> 1161 [ACK] Seq=1 Ack=3486 Win=11...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=3486 Win=11...
0.124185	1161	1161 -> 80 [PSH, ACK] Seq=7866 Ack=1 Win=...	80		TCP: 1161 -> 80 [PSH, ACK] Seq=7866 Ack=1 Win=...
0.169118	1161	80 -> 1161 [ACK] Seq=1 Ack=4946 Win=14...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=4946 Win=14...
0.217299	1161	80 -> 1161 [ACK] Seq=1 Ack=6406 Win=17...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=6406 Win=17...
0.267802	1161	80 -> 1161 [ACK] Seq=1 Ack=7866 Win=20...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=7866 Win=20...
0.304807	1161	80 -> 1161 [ACK] Seq=1 Ack=9013 Win=23...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=9013 Win=23...
0.305040	1161	1161 -> 80 [ACK] Seq=9013 Ack=1 Win=17...	80		TCP: 1161 -> 80 [ACK] Seq=9013 Ack=1 Win=17...
0.305813	1161	1161 -> 80 [ACK] Seq=10473 Ack=1 Win=1...	80		TCP: 1161 -> 80 [ACK] Seq=10473 Ack=1 Win=1...
0.306692	1161	1161 -> 80 [ACK] Seq=11933 Ack=1 Win=1...	80		TCP: 1161 -> 80 [ACK] Seq=11933 Ack=1 Win=1...
0.307571	1161	1161 -> 80 [ACK] Seq=13393 Ack=1 Win=1...	80		TCP: 1161 -> 80 [ACK] Seq=13393 Ack=1 Win=1...
0.308699	1161	1161 -> 80 [ACK] Seq=14853 Ack=1 Win=1...	80		TCP: 1161 -> 80 [ACK] Seq=14853 Ack=1 Win=1...
0.309553	1161	1161 -> 80 [PSH, ACK] Seq=16313 Ack=1 Win=...	80		TCP: 1161 -> 80 [PSH, ACK] Seq=16313 Ack=1 Win=...
0.356437	1161	80 -> 1161 [ACK] Seq=1 Ack=10473 Win=2...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=10473 Win=2...
0.400164	1161	80 -> 1161 [ACK] Seq=1 Ack=11933 Win=2...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=11933 Win=2...
0.448613	1161	80 -> 1161 [ACK] Seq=1 Ack=13393 Win=3...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=13393 Win=3...
0.500029	1161	80 -> 1161 [ACK] Seq=1 Ack=14853 Win=3...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=14853 Win=3...
0.545052	1161	80 -> 1161 [ACK] Seq=1 Ack=16313 Win=3...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=16313 Win=3...
0.576417	1161	80 -> 1161 [ACK] Seq=1 Ack=17205 Win=3...	80		TCP: 80 -> 1161 [ACK] Seq=1 Ack=17205 Win=3...
0.576671	1161	1161 -> 80 [ACK] Seq=17205 Ack=1 Win=1...	80		TCP: 1161 -> 80 [ACK] Seq=17205 Ack=1 Win=1...
0.577385	1161	1161 -> 80 [ACK] Seq=18665 Ack=1 Win=1...	80		TCP: 1161 -> 80 [ACK] Seq=18665 Ack=1 Win=1...
0.578329	1161	1161 -> 80 [ACK] Seq=20125 Ack=1 Win=1...	80		TCP: 1161 -> 80 [ACK] Seq=20125 Ack=1 Win=1...

Packet 32: TCP: 1161 -> 80 [ACK] Seq=20125 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]

☐ Limit to display filter

Flow type: All Flows

Addresses: Any

Reset Diagram Export Close Help

งานครั้งที่ 8

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