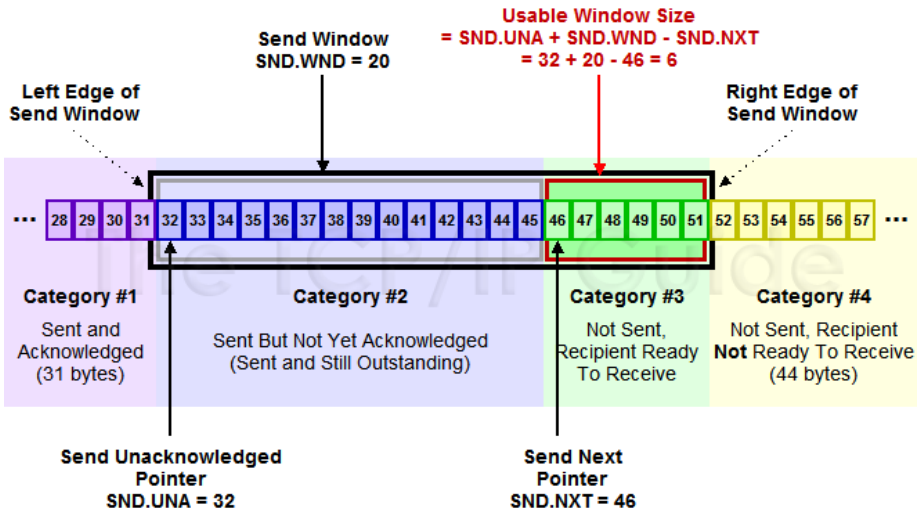
กิจกรรมที่ 8 : TCP Window

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

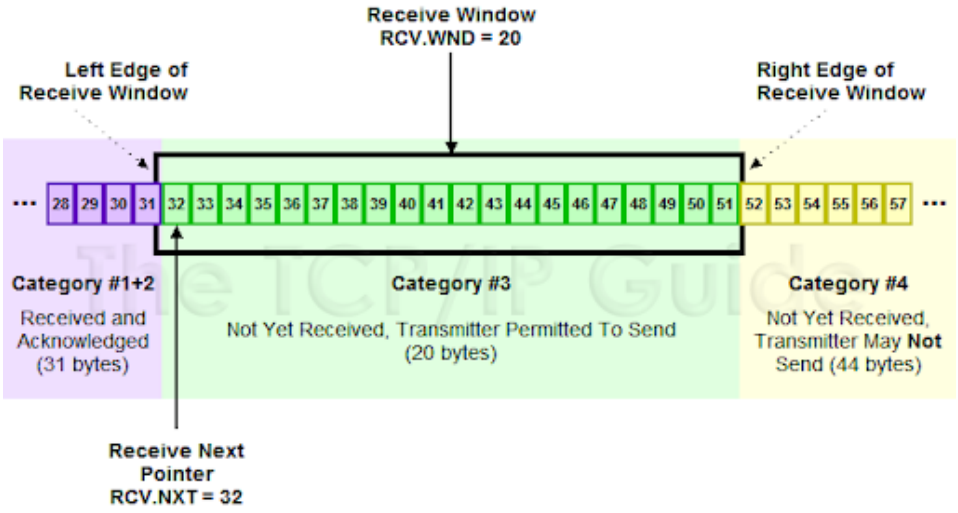
ใน send window จะแบ่งออกเป็น 4 ส่วน คือ

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

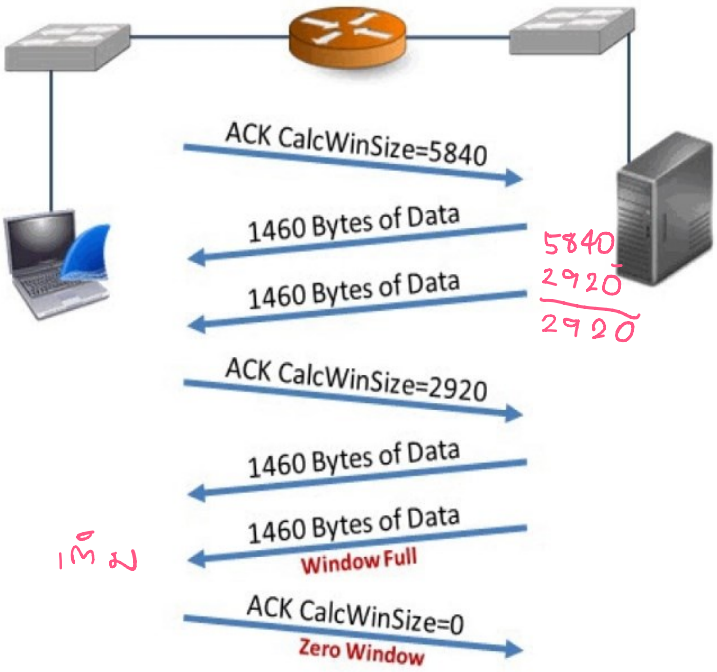


ใน receive window จะแบ่งเป็น 2 ส่วน

- 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 หมายเลข 2718 ซึ่งเป็น packet ที่ host 24.4.7.217 ส่ง ACK กลับมา โดยมี window size เหลือเพียง 1,460 ไบต์
    - ต่อมาใน packet หมายเลข 2719 พบว่า host 208.117.232.102 มีการส่งข้อมูลไปอีก 1,460 ไบต์ ซึ่งจะทำให้เต็ม receive window พอดี และทำให้ wireshark สร้างข้อมูลแจ้งเตือนว่า window full
    - เมื่อถึง packet หมายเลข 2720 พบว่า host 24.4.7.217 ส่ง packet ACK กลับมา โดยมีค่า window size เป็น 0 ทำให้ wireshark สร้างข้อมูลแจ้งเตือนว่า zero window
    - ให้สังเกตช่วงเวลาระหว่าง packet หมายเลข 2720 และ 2721 จะเห็นว่ามีระยะห่างมากกว่าปกติ หมายความว่าฝั่งผู้ส่งเมื่อพบ zero window ก็จะรอฝั่งผู้รับให้เคลียร์ receive window เสียก่อน
    - ใน packet หมายเลข 2721 จะมีการส่ง packet keep alive (คือ packet ACK ที่ไม่มีข้อมูล จากฝั่งผู้ส่ง ซึ่งจะเกิดขึ้นเมื่อ keepalive timer expire)
    - จากนั้นใน packet หมายเลข 2722 ผู้รับจะส่ง ACK กลับมา โดยมี window size เป็น 0 เช่นเดิม และเกิดซ้ำอีกครั้งใน packet หมายเลข 2723 และ 2724
    - จนกระทั่ง packet หมายเลข 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 สิ้นสุดตอนที่ ws ไม่ใช่ 0

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

zero window ที่ packet 4023

window update ที่ packet 4036

4022	12.679273	208.117.232.102	24.4.7.217	HTTP	382	4248122	4248450	1270	8384	[TCP Window Full] Continuation
4023	12.889025	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4024	13.366647	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4025	13.366693	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4026	14.362070	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4027	14.362127	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4028	16.240228	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4029	16.240291	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4030	19.945115	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4031	19.945256	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4032	27.344112	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4033	27.344212	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4034	37.364265	208.117.232.102	24.4.7.217	TCP	60	4248449	4248449	1270	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1270 Win=8384 Len=0
4035	37.364317	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=0 Len=0
4036	38.319249	24.4.7.217	208.117.232.102	TCP	54	1270	1270	4248450	166440	[TCP Window Update] 56770 → 80 [ACK] Seq=1270 Ack=4248450 Win=166440 Len=0

- หลังจากมีการทำ keep alive กี่ครั้ง มีช่วงระยะเวลาเท่าไรบ้าง นับจาก zero window ครั้งก่อน

เกิด keep alive 6 ครั้ง

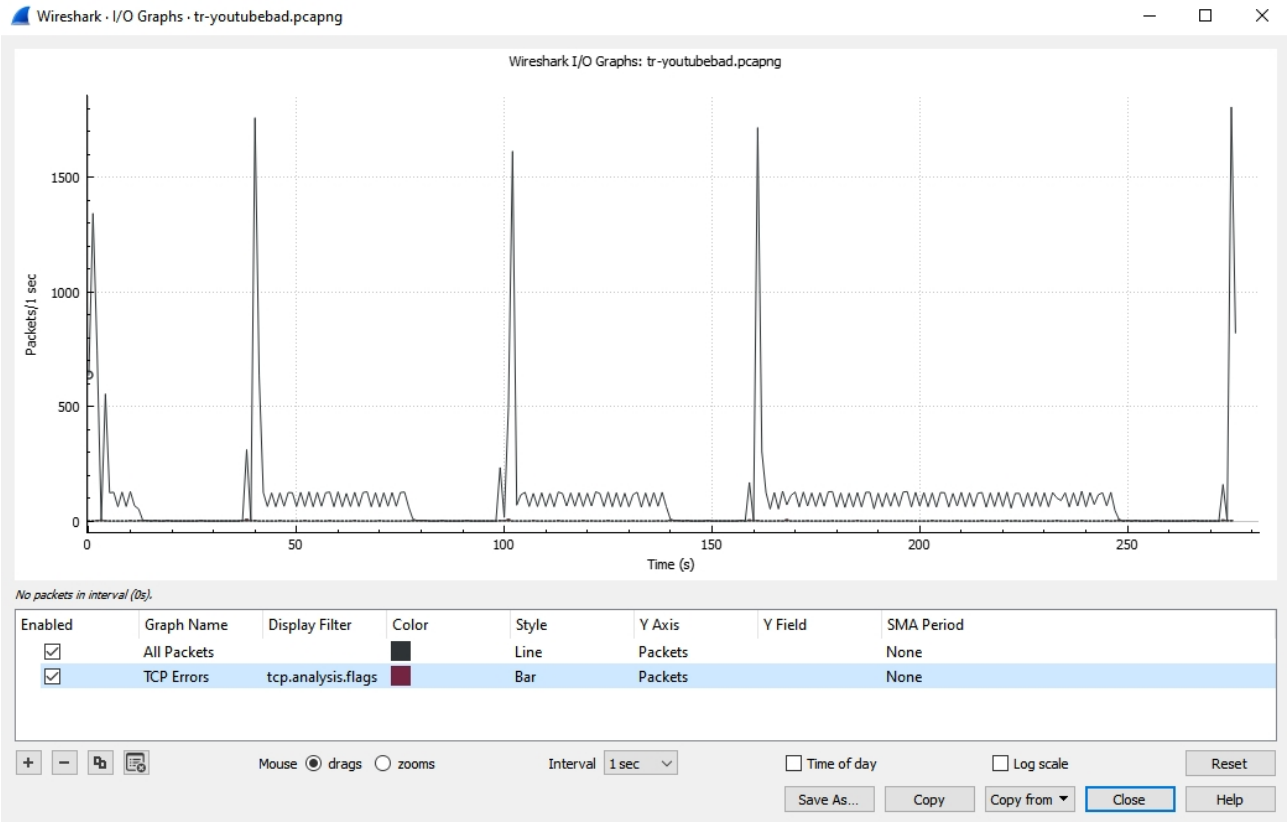
4022	12.679273
4023	*REF*
4024	0.477622
4025	*REF*
4026	0.995377
4027	*REF*
4028	1.878101
4029	*REF*
4030	3.704824
4031	*REF*
4032	7.398856
4033	*REF*
4034	10.020053

1) 0.477622 วินาที
2) 0.995377 วินาที
3) 1.878101 วินาที
4) 3.704824 วินาที
5) 7.398856 วินาที
6) 10.020053 วินาที

- ระยะเวลาตั้งแต่เกิด zero window ครั้งแรกจนถึง window update ใช้เวลาเท่าไร

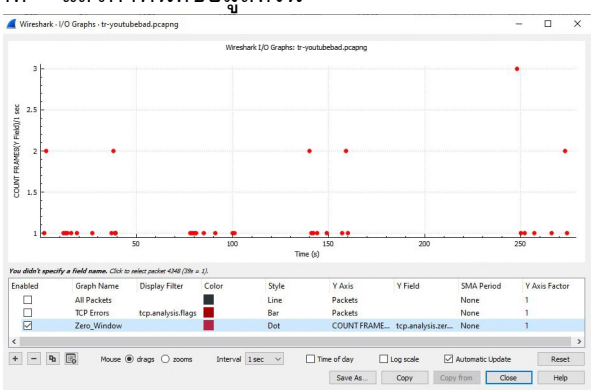
25.430224 วินาที

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

- ให้ Disable กราฟเดิมที่มีอยู่ทุกกราฟ โดยคลิกที่ช่องสี่เหลี่ยมในคอลัมน์ Enabled ของแต่ละกราฟเพื่อนำเครื่องหมายถูกออก
- 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



- กราฟใหม่ที่พึ่งสร้างขึ้นบอกข้อมูลอะไร

แสดงการเกิด zero window ในช่วงเวลาต่าง ๆ

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

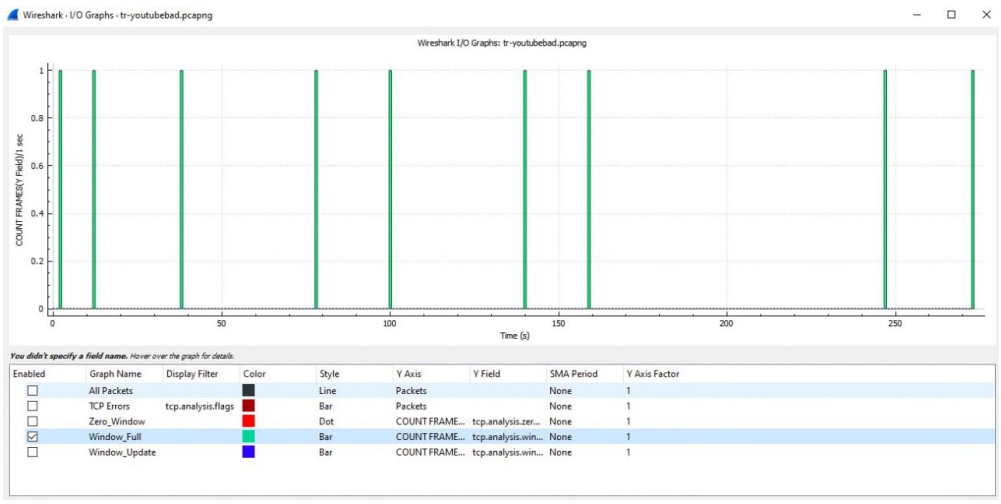
สีแดงคือการเกิด zero window

สีเขียวคือการเกิด window full

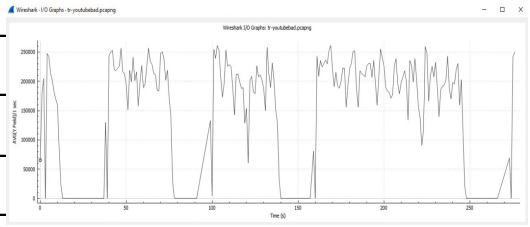
สีน้ำเงินการเกิด window update

- จากกราฟสามารถบอกได้หรือไม่ว่ามี window full กี่ครั้ง ให้บันทึกภาพ screenshot ประกอบด้วย

๑ 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



แสดง calculate windowsize โดยเฉลี่ย เฉพาะ ip 24.4.7.217 ในช่วงเวลาต่างๆ

สามารถวิเคราะห์หา window full, zero window และ window update ได้

โดยที่ช่วงที่กราฟเป็น 0 แสดงว่าเกิด zero window

ช่วงที่กราฟกำลังลดเข้าใกล้ 0 แสดงว่าเกิด window full

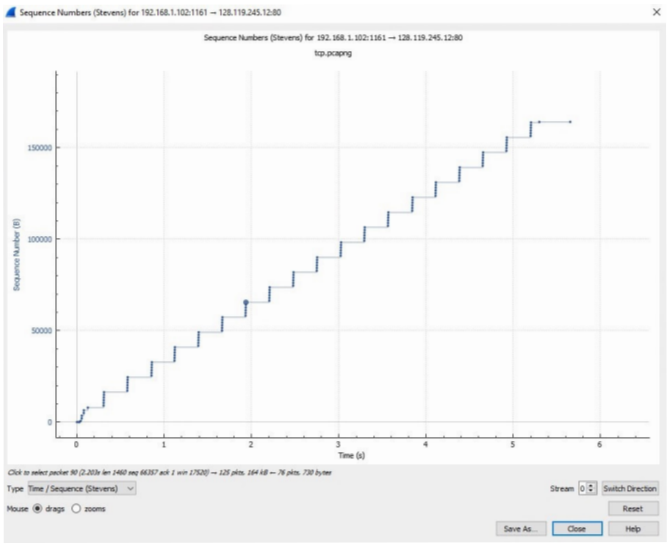
ช่วงที่กราฟเพิ่มขึ้นหลังจากเป็น 0 แสดงว่าเกิด window update

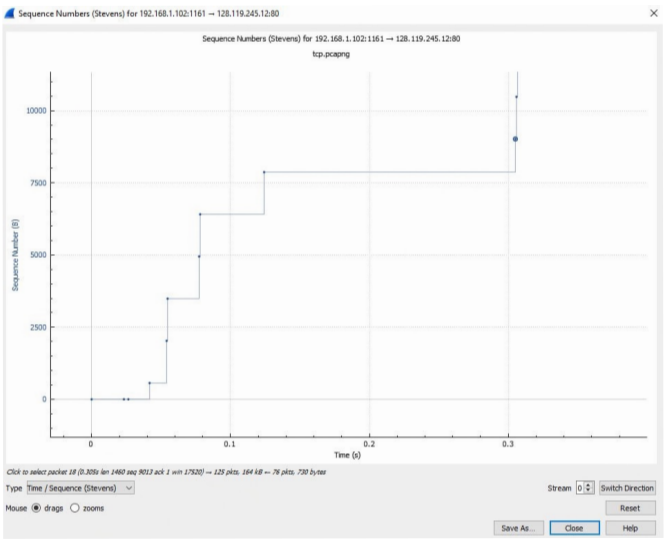
6. ในการควบคุม congestion control ของ TCP จะมีหลักอยู่ 2 ข้อ คือ Slow Start และ Congestion Avoidance ให้เปิดไฟล์ tcp.pcapng แล้วดูที่ Statistics->TCP Stream Graph-> Time-Sequence-Graph(Stevens) จากนั้นคลิกที่ปุ่ม Switch Direction เพื่อเปลี่ยนทิศทางให้เป็นทิศที่ส่งจาก host 192.168.1.102 ส่งไปยัง host 128.119.245.12 โดยแต่ละจุดแสดงถึงการส่งในแต่ละ TCP segment ให้พิจารณากราฟนี้ร่วมกับกราฟจาก Statistics-> Flow Graph นักศึกษาสามารถบอกได้หรือไม่ว่า Slow Start เริ่มต้นและสิ้นสุดที่ใด และมี Congestion Avoidance เกิดขึ้นหรือไม่

หน้าถัดไป

งานครั้งที่ 8

- การส่งงาน เขียนหรือพิมพ์ลงในเอกสารนี้ และส่งเป็นไฟล์ PDF เท่านั้น
- ตั้งชื่อไฟล์โดยใช้รหัสนักศึกษา ตามด้วย section และ _lab08 ตามตัวอย่างต่อไปนี้ 64019999_sec20_lab08.pdf
- กำหนดส่ง ภายในวันที่ 24 มีนาคม 2566 โดยให้ส่งใน Microsoft Teams ของรายวิชา

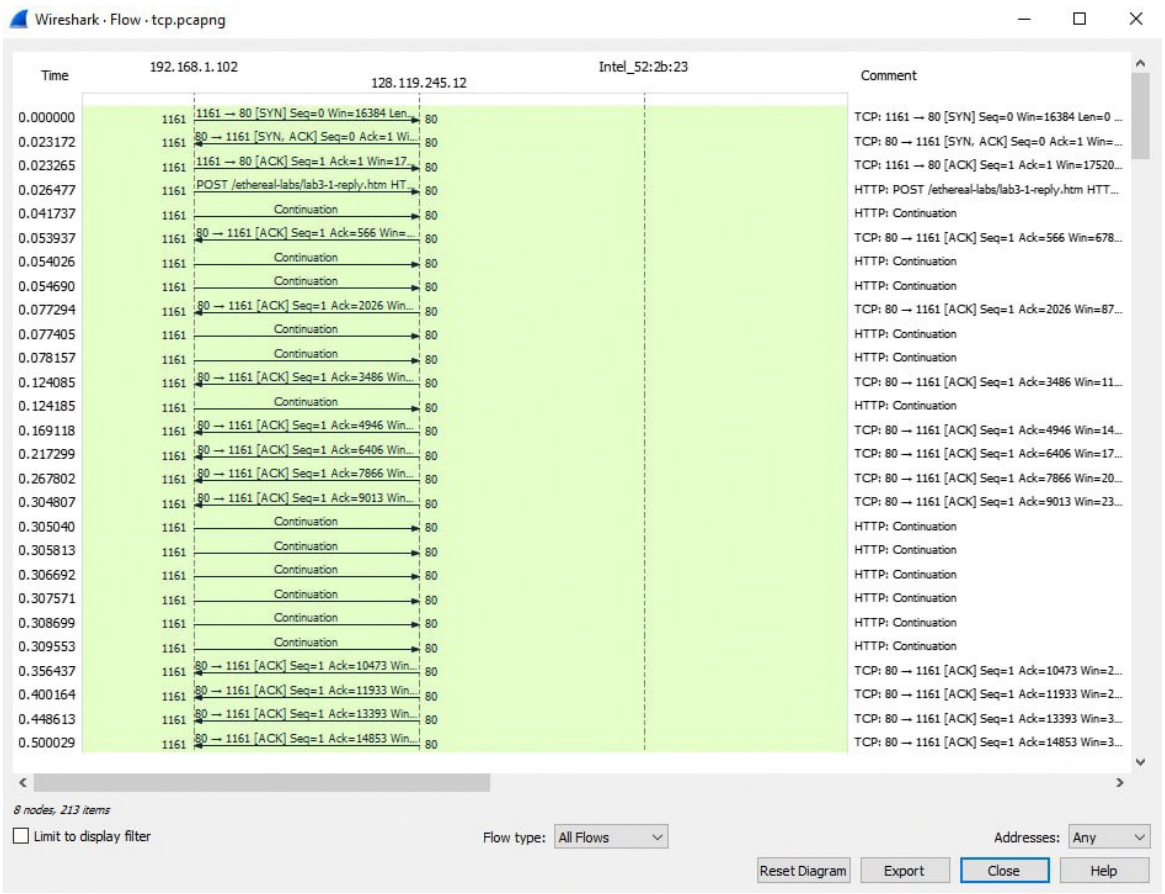




ช่วงที่เกิด slow start คือช่วงวินาทีที่ 0.0 - 0.1

การเพิ่มขึ้นของ window size ที่ดี จะมีการเพิ่มขึ้นแบบ exponential increment

และช่วงที่เกิด congestion avoidance คือหลังจากวินาทีที่ 0.1 ขึ้นไป



หากนำ flow graph มาวิเคราะห์จะทำให้ได้เวลาที่เกิด slow start หรือ congestion avoidance ได้ชัดเจน โดยดูจากขนาดของ window size ในช่วงเวลา 0-0.1 การเพิ่มขึ้นของ window size เป็นแบบ exponential increment

หลังจากนั้นเป็นการเพิ่มขึ้นแบบ additional เป็นการเพิ่มขึ้นด้วยค่าที่เท่าๆกันไปเรื่อยๆ

ซึ่ง กราฟทั้งสองแบบสามารถหาได้ทั้ง slow start และ congestion avoidance แต่ถ้านำมาวิเคราะห์ร่วมกันจะได้เวลาที่ชัดเจนยิ่งขึ้น

No.	Time	Source	Destination	Protocol	Length	Host	Info
2712	2.483887	208.117.232.102	24.4.7.217	TCP	1514		80 → 56770 [ACK] Seq=2874802 Ack=1270 Win=8384 Len=1460 [TCP segment of a re
2713	2.484052	24.4.7.217	208.117.232.102	TCP	54		56770 → 80 [ACK] Seq=1270 Ack=2876262 Win=7300 Len=0
2714	2.485234	208.117.232.102	24.4.7.217	TCP	1514		80 → 56770 [ACK] Seq=2876262 Ack=1270 Win=8384 Len=1460 [TCP segment of a re
2715	2.485237	208.117.232.102	24.4.7.217	TCP	1514		80 → 56770 [ACK] Seq=2877722 Ack=1270 Win=8384 Len=1460 [TCP segment of a re
2716	2.485241	208.117.232.102	24.4.7.217	TCP	1514		80 → 56770 [ACK] Seq=2879182 Ack=1270 Win=8384 Len=1460 [TCP segment of a re
2717	2.485244	208.117.232.102	24.4.7.217	TCP	1514		80 → 56770 [ACK] Seq=2880642 Ack=1270 Win=8384 Len=1460 [TCP segment of a re
2718	2.485341	24.4.7.217	208.117.232.102	TCP	54		56770 → 80 [ACK] Seq=1270 Ack=2882102 Win=1460 Len=0
2719	2.486110	208.117.232.102	24.4.7.217	TCP	1514		[TCP Window Full] 80 → 56770 [ACK] Seq=2882102 Ack=1270 Win=8384 Len=1460 [T
2720	2.688431	24.4.7.217	208.117.232.102	TCP	54		[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=2883562 Win=0 Len=0
2721	3.095692	208.117.232.102	24.4.7.217	TCP	60		[TCP Keep-Alive] 80 → 56770 [ACK] Seq=2883561 Ack=1270 Win=8384 Len=0
2722	3.095741	24.4.7.217	208.117.232.102	TCP	54		[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=2883562 Win=0 Len=0
2723	3.882335	208.117.232.102	24.4.7.217	TCP	60		[TCP Keep-Alive] 80 → 56770 [ACK] Seq=2883561 Ack=1270 Win=8384 Len=0
2724	3.882383	24.4.7.217	208.117.232.102	TCP	54		[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=2883562 Win=0 Len=0
2725	4.094206	24.4.7.217	208.117.232.102	TCP	54		[TCP Window Update] 56770 → 80 [ACK] Seq=1270 Ack=2883562 Win=243820 Len=0
2726	4.114835	208.117.232.102	24.4.7.217	TCP	1514		80 → 56770 [ACK] Seq=2883562 Ack=1270 Win=8384 Len=1460 [TCP segment of a re
2727	4.115674	208.117.232.102	24.4.7.217	TCP	1514		80 → 56770 [ACK] Seq=2885022 Ack=1270 Win=8384 Len=1460 [TCP segment of a re
2728	4.115750	24.4.7.217	208.117.232.102	TCP	54		56770 → 80 [ACK] Seq=1270 Ack=2886482 Win=240900 Len=0