กิจกรรมที่ 8 : TCP Window

Q(2) :ให้นักศึกษาตรวจสอบ zero window ระยะที่ 2 แล้วตอบคำถาม ต่อไปนี้ เกิด window full, zero window (เฉพาะครั้งแรก) และ window update ที่ packet ใด

: เกิด window full ที่ packet 4022 เกิด zero window ที่ packet 4023 และเกิด window update ที่ packet 4036

No.	Time	Source	Destination	Protocol	Length	WinSize	Info
	4021 12.316990	24.4.7.217	208.117.232.102	TCP	54	328	56770 → 80 [ACK] Seq=1270 Ack=4248122 Win=328 Len=0
\star	4022 12.679273	208.117.232.102	24.4.7.217	TCP	382	8384	[TCP Window Full] 80 → 56770 [PSH, ACK] Seq=4248122
	4023 12.889025	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248
	4024 13.366647	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1
	4025 13.366693	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248
	4026 14.362070	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1
	4027 14.362127	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248
	4028 16.240228	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1
	4029 16.240291	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248
	4030 19.945115	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1
	4031 19.945256	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248
	4032 27.344112	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1
	4033 27.344212	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248
	4034 37.364265	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4248449 Ack=1
	4035 37.364317	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1270 Ack=4248
	4036 38.319249	24.4.7.217	208.117.232.102	TCP	54	166440	[TCP Window Update] 56770 → 80 [ACK] Seq=1270 Ack=4

หลังจากมีการทำ keep alive กี่ครั้ง มีช่วงระยะเวลาเท่าไรบ้าง นับจาก zero window ครั้งก่อน

: เกิดการทำ TCP Keep-Alive ทั้งหมด 6 ครั้ง มีระยะเวลาคือ 0.477622 , 0.995377 , 1.878101 , 3.704824 , 7.398856 และ 10.020053 วินาที ตามลำดับ

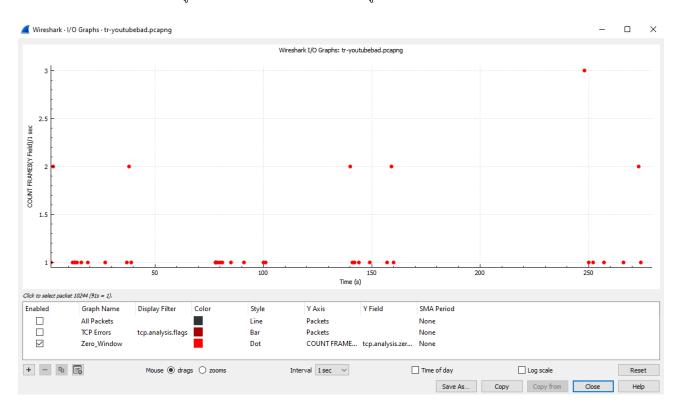
No.	Time	DeltaTime	Source	Destination	Protocol	Length	WinSize	Info
4021	12.316990	0.2006850	24.4.7.217	208.117.232.102	TCP	54	328	56770 → 80 [ACK] Seq=1270 Ack=4248122 W
4022	12.679273	0.3622830.	208.117.232.102	24.4.7.217	TCP	382	8384	[TCP Window Full] 80 → 56770 [PSH, ACK]
4023	12.889025	0.2097520	24.4.7.217	208.117.232.102	TCP	54		[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1
4024	13.366647	0.4776220	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4
4025	13.366693	0.0000460	24.4.7.217	208.117.232.102	TCP	54		[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1
4026	14.362070	0.9953770	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4
4027	14.362127	0.0000570	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1
4028	16.240228	1.8781010	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4
4029	16.240291	0.0000630	24.4.7.217	208.117.232.102	TCP	54		[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1
4030	19.945115	3.7048240	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4
4031	19.945256	0.0001410.	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1
4032	27.344112	7.3988560	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4
4033	3 27.344212	0.0001000	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1
4034	37.364265	10.020053	208.117.232.102	24.4.7.217	TCP	60	8384	[TCP Keep-Alive] 80 → 56770 [ACK] Seq=4
4035	37.364317	0.0000520	24.4.7.217	208.117.232.102	TCP	54	0	[TCP ZeroWindow] 56770 → 80 [ACK] Seq=1
4036	38.319249	0.9549320	24.4.7.217	208.117.232.102	TCP	54	166440	[TCP Window Update] 56770 → 80 [ACK] Se

ระยะเวลาตั้งแต่เกิด zero window ครั้งแรกจนถึง window update ใช้เวลาเท่าไร

: เกิด Zero Window ครั้งแรกที่ packet 4023 ที่เวลา 12.889025 วินาทีและเกิด Window Update ที่ packet 4036 ที่ เวลา 38.319249 วินาที ดังนั้นเราจะได้ระยะเวลาตั้งแต่เกิด Zero Window จนถึง Window Update เท่ากับ 38.319249 – 12.889025 = 25.430224 วินาที

Q(3) : จากการวิเคราะห์ข้อมูลจาก Statistics I/O Graph กราฟบอกข้อมูลอะไร

A(3) : จากกราฟที่ได้เป็นการบอกข้อมูลจำนวนครั้งที่เกิด Zero Window อยู่ในแนวแกน Y และเทียบกับเวลาในแนวแกน X



Q(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



Q(4.1): กราฟแสดงอะไร

A(4.1) : จากกราฟ bar สีเขียว แสดงจำนวนครั้งที่พบการเกิด window full (แกน Y) โดยเทียบกับเวลา (แกน X) และ จากกราฟ bar สีน้ำเงิน แสดงจำนวนครั้งที่พบการเกิด window update (แกน Y) โดยเทียบกับเวลา (แกน X)

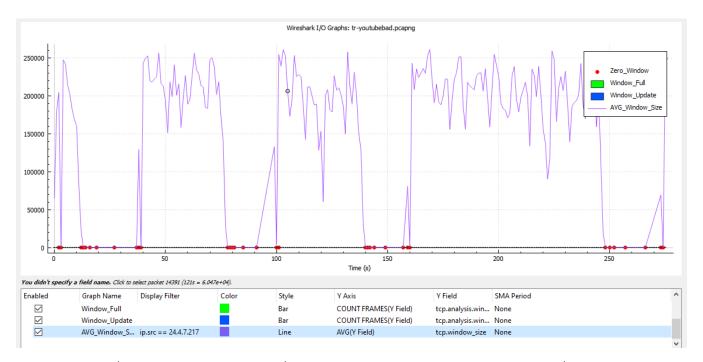
Q(4.2) : จากกราฟสามารถบอกได้หรือไม่ว่ามี window full กี่ครั้ง ให้ Capture รูปประกอบด้วย

A(4.2) : สามารถบอกได้จากกราฟ bar สีเขียว ที่เกิดขึ้นบนกราฟจำนวน 6 ครั้ง แสดงถึงการเกิด window full



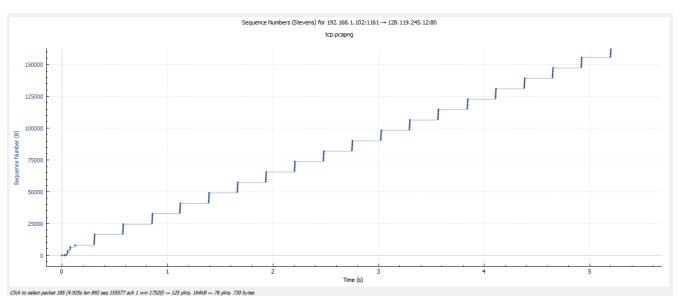
Q(5) : ให้สร้าง I/O Graph ใหม่ดังนี้

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

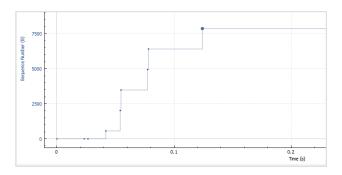


A(5) : จากกราฟที่ได้เป็นการบอกข้อมูลจำนวนค่าเฉลี่ยของขนาด window หรือ window size ของ packet ที่เกิดจาก src ip คือ 24.4.7.217 อยู่ในแนวแกน Y และเทียบกับเวลาในแนวแกน X สามารถนำมาใช้วิเคราะห์ความสามารถของการรับ-ส่ง ข้อมูลระหว่าง send window และ receive window ได้

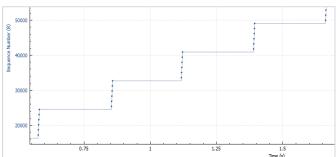
Q(6) :ในการควบคุม congestion control ของ TCP จะมีหลักอยู่ 2 ข้อ คือ Slow Start และ Congestion Avoidance ให้เปิดไฟล์ tcp.pcapng แล้วดูที่ Statistics->TCP Stream Graph-> Time-Sequence-Graph(Stevens) โดยแต่ละจุดแสดงถึง การส่งในแต่ละ segment ร่วมกับ Statistics-> Flow Graph นักศึกษาสามารถบอกได้หรือไม่ว่า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



เป็นกราฟที่ทำการ Switch Direction โดยจากกราฟในไฟล์ tcp.pcapng เราสามารถเห็นกระบวนการของ Slow Start และ Congestion Avoidance ได้ ดังนี้



กราฟแสดงการเกิด Slow Start จะสังเกตุเห็นได้จากค่าของ sequence Number ที่มีค่าเพิ่มขึ้นแบบ exponential (เช่น 1 , 2 , 4 , 8 , 16 , ...)



กราฟแสดงการเกิด Congestion Avoidance จะสังเกตุ เห็นได้จากค่าของ sequence Number ที่มีค่าเพิ่มขึ้น แบบอย่างคงที่ หรือแบบ Linear (เช่น +1, +2, +3, ...)

เป็นกราฟที่ได้จาก Flow Graph ซึ่งหากเรานำมาวิเคราะห์อาจใช้ parameter "Win" ที่มาจาก Window Size มาหาช่วงต่างๆ ได้ เช่น ในช่วงแรก win มีการเพิ่มขึ้นอย่างต่อเนื่อง เช่น 6780 -> 8760 -> 11680 -> ... -> 62780 อาจแสดงถึงช่วง Slow Start ได้

Time	192.168	1.1.102	128.119.245.12
0.000000	1161	1161 — 80 [5YN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1	▶ 80
0.023172	1161	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1	80
0.023265	1161	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0	▶ 80
0.026477	1161	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP segment of a reassembled PDU]	▶ 80
0.041737	1161	1161 — 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	▶ 80
0.053937	1161	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0	80
0.054026	1161	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	▶ 80
0.054690	1161	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	▶ 80
0.077294	1161	80 — 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0	80
0.077405	1161	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	● 80
0.078157	1161	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	▶ 80
0.124085	1161	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0	80
0.124185	1161	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147 [TCP segment of a reassembled PDU]	
0.169118	1161	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0	80
0.217299	1161	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0	80
0.267802	1161	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0	80
0.304807	1161	80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0	80
0.305040	1161	1161 → 80 [ACK] Seq=9013 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
0.305813	1161	1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
0.306692	1161	1161 — 80 [ACK] Seq=11933 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
0.307571	1161	1161 → 80 [ACK] Seq=13393 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
0.308699	1161	1161 — 80 [ACK] Seq=14853 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
0.309553	1161	1161 → 80 [PSH, ACK] Seq=16313 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]	→ 80
0.356437	1161	80 → 1161 [ACK] Seq=1 Ack=10473 Win=26280 Len=0	80
0.400164	1161	80 → 1161 [ACK] Seq=1 Ack=11933 Win=29200 Len=0	80
0.448613	1161	80 → 1161 [ACK] Seq=1 Ack=13393 Win=32120 Len=0	80
0.500029	1161	80 → 1161 [ACK] Seq=1 Ack=14853 Win=35040 Len=0	80
0.545052	1161	80 → 1161 [ACK] Seq=1 Ack=16313 Win=37960 Len=0	80
0.576417	1161	80 → 1161 [ACK] Seq=1 Ack=17205 Win=37960 Len=0	80
0.576671	1161	1161 → 80 [ACK] Seq=17205 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
0.577385	1161	1161 → 80 [ACK] Seq=18665 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
0.578329	1161	1161 → 80 [ACK] Seq=20125 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	▶ 80
0.579195	1161	1161 → 80 [ACK] Seq=21585 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
0.580149	1161	1161 → 80 [ACK] Seq=23045 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	▶ 80
0.581074	1161	1161 — 80 [PSH, ACK] Seq=24505 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]	▶ 80
0.626496	1161	80 1161 [ACK] Seq=1 Ack=18665 Win=40880 Len=0	80
0.672796	1161	80 → 1161 [ACK] Seq=1 Ack=20125 Win=43800 Len=0	80
0.730684	1161	80 1161 [ACK] Seq=1 Ack=21585 Win=46720 Len=0	80
0.772990	1161	80 → 1161 [ACK] Seq=1 Ack=23045 Win=49640 Len=0	80
0.820622	1161	80 → 1161 [ACK] Seq=1 Ack=24505 Win=52560 Len=0	80
0.853186	1161	80 → 1161 [ACK] Seq=1 Ack=25397 Win=52560 Len=0	80
0.853405	1161	1161 → 80 [ACK] Seq=25397 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	▶ 80
0.854076	1161	1161 → 80 [ACK] Seq=26857 Ack=1 Wi=17520 Len=1460 [TCP segment of a reassembled PDU]	▶ 80

ส่วนต่อมาค่า win มีการเพิ่มขึ้นแบบค่าคงที่ที่ 62780 ก็อาจวิเคราะห์ได้ว่าช่วงนี้เป็น Congestion Avoidance

4.651858	1161	1161 → 80 (ACK) Seq=145925 Ack=1 Win=17520 Len=1460 (TCP segment of a reassembled PDU)	■ 80
4.652735	1161	1161 → 80 [PSH, ACK] Seq=147385 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]	■ 80
4.747988	1161	80 → 1161 [ACK] Seq=1 Ack=143005 Win=62780 Len=0	80
4.844598	1161	80 → 1161 [ACK] Seq=1 Ack=145925 Win=62780 Len=0	80
4.920051	1161	80 → 1161 [ACK] Seq=1 Ack=148277 Win=62780 Len=0	- 80
4,920310	1161	1161 — 80 [ACK] Seq=148277 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	■ 80
4.921025	1161	1161 — 80 [ACK] Seq=149737 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	■ 80
4.921916	1161	1161 — 80 [ACK] Seq=151197 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	■ 80
4.922820	1161	1161 — 80 [ACK] Seq=152657 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	₩ 80
4.923863	1161	1161 — 80 [ACK] Seq=154117 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	8 0
4.924667	1161	1161 — 80 [PSH, ACK] Seq=155577 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]	₩ 80
5.019189	1161	80 → 1161 (ACK) Seq=1 Ack=151197 Win=62780 ten=0	80
5.104175			
5.105060			1
5.106121			
5.125019	1161	80 — 1161 [ACK] Seq=1 Ack=154117 Win=62780 Len=0	80
5.197286	1161	80 → 1161 (ACK) Seq=1 Ack=156469 Win=62780 Len=0	80
5.197508	1161	1161 — 80 [ACK] Seq=156469 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	₩ 80
5.198388	1161	1161 — 80 [ACK] Seq=157929 Ack=1 Win=17520 Len=1460 (TCP segment of a reassembled PDU)	⇒ 80
5.199275	1161	1161 — 80 [ACK] Seq=159389 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	₩ 80
5.200252	1161	1161 — 80 [ACK] Seq=160849 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	→ 80
5.201150	1161	1161 — 80 [ACK] Seq=162309 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]	⇒ 80
5.202024	1161	1161 — 80 [PSH, ACK] Seq=163769 Ack=1 Win=17520 Len=272 [TCP segment of a reassembled PDU]	■ 80
5.297257	1161	80 — 1161 [ACK] Seq=1 Ack=159389 Win=62780 Len=0	80
5.297341	1161	POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)	₩ 80
5.389471	1161	80 → 1161 (ACK) Seq=1 Ack=162309 Win=62780 Len=0	80
5.447887	1161	80 — 1161 [ACK] Seq=1 Ack=164041 Win=62780 Len=0	80
5.455830	1161	80 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0	80
5.461175	1161	HTTP/1.1 200 OK (text/html)	80
5.598090			
5.599082			
5.651141	1161	1161 — 80 [ACK] Seq=164091 Ack=731 Win=16790 Len=0	⊷ 80
6.101044			
6.102069			
6.600152			
6.601063			
7.102852			
7.103780			