Lab 07: TCP Flow Control and Congestion Control

ในปฏิบัติการส่วนนี้เราจะสำรวจพฤติกรรมการทำงานของ TCP โดยมุ่งเน้นไปที่เรื่องการควบคุมอัตราการส่งข้อมูล เราจะได้เห็นเราจะได้เห็นกลไกควบคุมการไหล (Flow Control) ของ TCP ซึ่งช่วยหลีกเลี่ยงการที่ฝั่งส่งจะส่งข้อมูลเกินความพร้อมรับของฝั่งผู้รับ รวมถึงอัลกอริทึมการควบคุมความคับคั่ง (Congestion Control Algorithm) ของ TCP

A. Adjusting TCP receive window

เนื่องด้วยขนาดของ receive window ของ TCP โดยทั่วไปจะถูกบริหารจัดการโดยระบบปฏิบัติการ การศึกษา TCP Flow Control ในส่วนนี้จึงปรับใช้ socket programming เพื่อระบุขนาด receive buffer เป็นการเฉพาะเพื่อช่วยให้เห็นถึงกลไกการทำงานที่ชัดเจนขึ้น โดยให้ทำการทดลองดังต่อไปนี้

1. สั่งรัน TCPServer_FlowControl.py เพื่อสร้าง server process ที่จะส่งข้อความบางส่วนที่ Vinton Gray Cert ผู้ที่ออกแบบ TCP/IP เคยกล่าวเอาไว้ โดยข้อความเหล่านั้นจะถูกส่งให้กับฝั่ง client ทันทีที่สร้างการเชื่อมต่อสำเร็จ
2. เปิด Wireshark และเริ่มทำการ capture packet โดยเลือก Adapter for loopback traffic capture และใช้ Capture filter ต่อไปนี้

```
tcp port 12000
```

3. สั่งรัน TCPClient_FlowControl.py เพื่อสร้าง client process ที่จะเชื่อมต่อไปยัง server process ผ่าน client socket ซึ่งกำหนดขนาดของ receive buffer ตามค่าที่ระบุผ่านการรับค่าทางคีย์บอร์ด
   ในที่นี้ให้ระบุค่า receive buffer เป็น 1048576
4. หลังจากที่ socket เชื่อมต่อสำเร็จ server process จะเริ่มส่ง TCP segment ที่มี payload ให้กับ client process โดยเมื่อข้อมูลถูกส่งถึงเครื่อง client ก็จะถูกนำมาเก็บพักไว้ที่ receive buffer ก่อนที่จะ client process จะมาอ่านข้อมูลจาก receive buffer ไปประมวลผล ซึ่งโปรแกรมที่กำหนดให้จะให้ระบุขนาดของข้อมูลที่ client process ต้องการจะอ่านผ่านทางการป้อนค่าทางคีย์บอร์ด ในที่นี้ให้ระบุค่า 1048576
5. โปรแกรมจะวนซ้ำเพื่อให้ client process อ่านข้อความจาก receive buffer ในที่นี้ให้วนอ่านไปเรื่อย จนพบว่าอ่านข้อมูลที่ server process ส่งมาครบทั้งหมด แล้วจึงป้อนค่า 0 ทางคีย์บอร์ดเพื่อจบการทำงาน client process
6. ให้ copy ผลลัพธ์จากการรัน TCPClient_FlowControl.py เก็บเอาไว้ในไฟล์ Lab07-A1.txt เพื่อใช้ตอบคำถาม

7. สลับไปหน้า Wireshark และสั่งให้หยุด capture และให้ save ไฟล์ไว้ด้วยชื่อ Lab07-A1.pcapng
8. ทำการทดลองซ้ำตั้งแต่ขั้นตอนที่ 2 จนถึงขั้นตอนที่ 7 เพิ่มอีกหลายรอบ โดยในแต่ละรอบให้เปลี่ยนแปลงค่าขนาด receive buffer และค่าจำนวน bytes ที่ client process อ่านจาก socket จากการเรียกฟังก์ชันในแต่ละครั้ง โดยหากรวมการทดลองครั้งแรกด้วยแล้ว ให้ใช้ค่าและใช้ชื่อไฟล์ตามตารางต่อไปนี้

Receive buffer size (bytes)	Number of bytes per reading (bytes)	Filename
1048576	1048576	Lab07-A1
1048576	64	Lab07-A2
16777216	16777216	Lab07-A3
32	128	Lab07-A4
32	16	Lab07-A5

Questions (A)

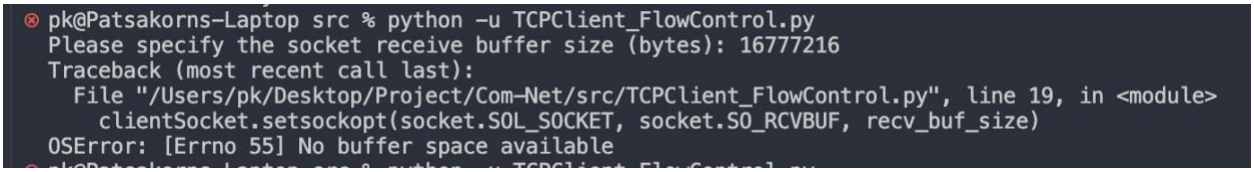
หลังจากทดลองสร้างการเชื่อมต่อระหว่าง client process และ server process ตามขั้นตอนข้างต้นแล้ว ให้ศึกษา source code ของโปรแกรมภาษา Python ที่ให้ไว้ทั้ง 2 ไฟล์ รวมถึงศึกษาไฟล์ packet capture ที่ดักจับได้เพื่อตอบคำถามต่อไปนี้

1) จากการศึกษาไฟล์ TCPServer_FlowControl.py หลังจากตอบรับการเชื่อมต่อจากฝั่ง client ในแต่ละ connection ฝั่ง server จะวนซ้ำเพื่อเรียกคำสั่ง connectionSocket.send เป็นจำนวนกี่ครั้ง?
    a. 3 ครั้ง
    b. "The Internet is literally a network of networks."
    c. "The Internet lives where anyone can access it."
    d. "The idea that you can somehow erase the Internet is silly."
2) จากไฟล์ Lab07-A1.txt พบว่าฝั่ง client มีการวนซ้ำเพื่ออ่านข้อมูลจาก receive buffer ด้วยคำสั่ง clientSocket.recv เป็นจำนวนกี่ครั้ง? แต่ละครั้งได้ข้อความใดบ้าง?
    a. 1 ครั้ง
    b. #0 | Read from buffer: The Internet is literally a network of networks.The Internet lives where anyone can access it.The idea that you can somehow erase the Internet is silly.
3) จากไฟล์ Lab07-A2.txt พบว่าฝั่ง client มีการวนซ้ำเพื่ออ่านข้อมูลจาก receive buffer ด้วยคำสั่ง clientSocket.recv เป็นจำนวนกี่ครั้ง? แต่ละครั้งได้ข้อความใดบ้าง?
    a. 1 ครั้ง
    b. #0 | Read from buffer: The Internet is literally a network of networks.The Internet liv
4) จากข้อ 1) ข้อ 2) และข้อ 3) ในการส่งข้อมูลผ่าน TCP ฝั่งผู้รับข้อมูลทราบหรือไม่ว่าฝั่งผู้ส่งเรียกฟังก์ชัน send เพื่อส่งข้อมูลเป็นจำนวนกี่ครั้ง? และผู้รับทราบหรือไม่ว่าผู้ส่งเรียกฟังก์ชัน send แต่ละครั้งส่งข้อมูลเท่าใดและสิ้นสุดลงที่ใด?

a. ไม่ทราบ

5) จากไฟล์ packet capture แต่ละไฟล์ จงพิจารณา TCP segment ที่สามจาก 3-way handshake ซึ่งเป็นการส่ง ACK จากฝั่ง client ไปยัง server จงตรวจสอบว่า Window ใน TCP header มีค่าเป็นเท่าใด? Wireshark คำนวณ Calculated window size ออกมาเป็นค่าเท่าใด? Window size scaling factor มีค่าเป็นเท่าใด?

Filename	Receive buffer (bytes)	Window	Calculated window size	Window size scaling factor
Lab07-A1.pcapng	1048576	65535	1048560	16
Lab07-A2.pcapng	1048576	65535	1048560	16
Lab07-A3.pcapng	16777216	-	-	-
Lab07-A4.pcapng	32	40830	326640	8
Lab07-A5.pcapng	32	40830	326640	8



Lab07-A3.pcapng space buffer ไม่พอ Mac user M2

6) จากข้อ 5) ฝั่ง client process มีการสื่อสารค่า Window size scaling factor ไปยังฝั่ง server process เมื่อใด? ค่าดังกล่าวอยู่ใน field ใดของ TCP header?

a. TCP three-way handshake.
b. Window size scaling factor เป็นส่วนหนึ่งของ TCP options ที่อยู่ใน TCP header และถูกใช้เพื่อเพิ่มขนาดของ receive window ที่สามารถรองรับได้มากกว่า 65,535 bytes (ซึ่งเป็นขีดจำกัดสูงสุดของค่า window size ที่สามารถระบุได้ด้วยฟิลด์ window size 16 บิตใน TCP header)

7) จากข้อ 5) จงพิจารณาค่า Window ค่า Calculated window size และค่า Window size scaling factor ค่าทั้งสามมีความสัมพันธ์กันอย่างไร? จงอธิบาย

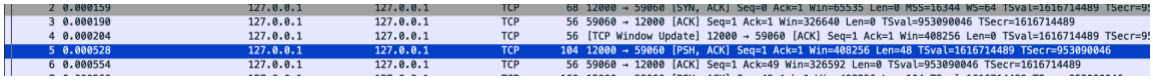
a. ค่า Window (Window Size Value): นี่คือค่าขนาดของ receive window ที่ส่งไปใน TCP header, และมันถูกจำกัดด้วยฟิลด์ 16 บิต, ซึ่งหมายความว่ามันสามารถมีค่าสูงสุดได้ถึง 65535 bytes.

b. ค่า Calculated window size: นี่คือค่าขนาดของ receive window ที่ถูกคำนวณจากค่า Window ที่ส่งมาใน TCP header โดยที่มีการปรับขนาดตาม Window size scaling factor. ในที่นี้, ค่า Calculated window size เป็นผลลัพธ์ของการคูณค่า Window ใน header ด้วยค่า Window size scaling factor.

c. ค่า Window size scaling factor: นี่คือตัวคูณที่ใช้สำหรับ scaling ค่า window size เพื่อให้สามารถมี receive window ที่ใหญ่กว่าขีดจำกัด 65535 bytes ของฟิลด์ window size ใน TCP header ได้. มันถูกตกลงกันในระหว่าง three-way handshake ของ TCP และเป็นค่าประจำที่ใช้สำหรับการเชื่อมต่อนั้นๆ.

8) ตรวจสอบไฟล์ Lab07-A4.pcapng และไฟล์ Lab07-A5.pcapng พบว่า TCP segment แรกที่มีการส่ง payload (the first data-delivery TCP segment) อยู่ใน packet หมายเลขใด? และนำส่ง TCP payload ขนาดเท่าใด?

a. No.5

b. 

c. TCP payload (48 bytes)

9) ตรวจสอบไฟล์ทั้ง 5 ในไฟล์ใดบ้าง ที่เกิดเหตุการณ์ฝั่งผู้รับประกาศไปยังฝั่งผู้ส่งข้อมูลว่าขนาด TCP Window มีค่าเป็น 0? และเกิดเหตุการณ์ดังกล่าวเกิดขึ้นครั้งแรกที่ packet หมายเลขใดในแต่ละไฟล์

a. ไม่เจอ

b. การคาดเดาเกี่ยวกับไฟล์ที่อาจเกิด Zero Window:

i. Lab07-A1: มี buffer size เท่ากับ 1,048,576 bytes และการอ่านที่ 1,048,576 bytes, น่าจะไม่มี Zero Window เพราะ buffer มีขนาดใหญ่พอที่จะรองรับข้อมูลที่ส่งมาทั้งหมดในครั้งเดียว.

ii. Lab07-A2: มี buffer size เท่ากับ 1,048,576 bytes แต่การอ่านที่เพียง 64 bytes, ไม่น่าจะเกิด Zero Window ในกรณีนี้เนื่องจากการอ่านมีขนาดเล็กมากเมื่อเทียบกับ buffer.

iii. Lab07-A3: มี buffer size เท่ากับ 16,777,216 bytes และการอ่านที่ 16,777,216 bytes, น่าจะไม่เกิด Zero Window เพราะ buffer มีขนาดใหญ่พอที่จะรองรับข้อมูลที่ส่งมาทั้งหมดในครั้งเดียว.

iv. Lab07-A4: มี buffer size เท่ากับ 32 bytes แต่การอ่านที่ 128 bytes, มีโอกาสสูงที่จะเกิด Zero Window เพราะการอ่านมีขนาดใหญ่กว่า buffer.

v. Lab07-A5: มี buffer size เท่ากับ 32 bytes แต่การอ่านที่ 16 bytes, อาจเกิด Zero Window ถ้าการส่งข้อมูลเกิดขึ้นอย่างรวดเร็วและผู้รับไม่สามารถประมวลผลข้อมูลที่ได้รับได้ทันท่วงที.

c. แต่ผมไม่เจอ ;-;

10) หลังจากเกิดเหตุการณ์ตามข้อ 9) ฝั่งผู้ส่งมีการส่ง TCP segment ที่มีลักษณะอย่างไรออกไป เพื่อสอบถามความพร้อมรับข้อมูลของฝั่งผู้รับ? (คำใบ้: โปรดสังเกตขนาด TCP payload ของ TCP segment เหล่านี้ว่ามีขนาดเท่าใด) Wireshark มีการระบุข้อความเฉพาะในคอลัมน์ Info ของ packet เหล่านี้ เป็นข้อความว่าอะไร?

a. เมื่อเกิดเหตุการณ์ Zero Window ซึ่งหมายความว่าฝั่งผู้รับได้ประกาศว่ามี TCP window size เป็น 0 และไม่สามารถรับข้อมูลเพิ่มเติมได้ในขณะนั้น, ฝั่งผู้ส่งจะหยุดส่งข้อมูลต่อไปและเริ่มกระบวนการสอบถามเพื่อตรวจสอบว่าเมื่อใดที่ฝั่งผู้รับพร้อมที่จะรับข้อมูลเพิ่มเติม.

b. TCP Zero Window Probe: ลักษณะ TCP Segment:

   i. ฝั่งผู้ส่งจะส่ง TCP segment ที่เรียกว่า Zero Window Probe. TCP segment เหล่านี้มีลักษณะเฉพาะคือมี TCP payload ขนาดเล็กมาก (โดยปกติจะเป็น 1 byte) เพื่อสอบถามความพร้อมของผู้รับ. ส่วนใหญ่จะไม่มีข้อมูลจริงที่จะส่ง, แต่เป็นการส่งเพื่อกระตุ้นให้ฝั่งผู้รับตอบกลับ.

   ii. การระบุใน Wireshark: ใน Wireshark, TCP segment ที่ส่งเป็น Zero Window Probe อาจจะถูกระบุในคอลัมน์ Info ด้วยข้อความเฉพาะ เช่น "ZeroWindowProbe" หรือ "TCP ZeroWindowProbe".

   iii. การตอบกลับ: ฝั่งผู้รับจะตอบกลับด้วย TCP ACK ซึ่งจะยังคงมี window size เป็น 0 หากยังไม่พร้อมที่จะรับข้อมูลเพิ่มเติม, หรือจะปรับ window size เป็นค่าที่มากกว่า 0 เพื่อบ่งบอกว่ามีพื้นที่ว่างใน buffer แล้วและพร้อมที่จะรับข้อมูลต่อไป.

11) จากกรณีที่ฝั่งผู้ส่งข้อมูลส่ง packets ตามข้อ 10) โปรดสังเกตว่าฝั่งผู้ส่งมีการเว้นช่วงระยะเวลาในการส่ง packet เป็นเวลาเท่าใดบ้าง? เว้นช่วงระยะเวลาเท่ากันหรือไม่ในแต่ละครั้ง? หากไม่เท่ากัน การเว้นช่วงระยะเวลาในแต่ละครั้งมีการเพิ่มหรือมีการลดในลักษณะอย่างไร? จงอธิบาย

a. การเว้นช่วงระยะเวลาการส่ง Zero Window Probes:

   i. Exponential Back-off: โดยปกติ, การเว้นช่วงระยะเวลาของ probes จะเริ่มต้นที่ค่าหนึ่ง (เช่น, 1 วินาที) และจะเพิ่มขึ้นเป็นเท่าตัวในแต่ละครั้งที่ probe ถูกส่งโดยไม่ได้รับคำตอบ. นี้อาจเป็น 1, 2, 4, 8, 16 วินาที และอื่นๆ เป็นต้น.

   ii. ความไม่เท่ากันของระยะเวลา: ระยะเวลาเหล่านี้ไม่เท่ากันและจะเพิ่มขึ้นตามกลไก exponential back-off เพื่อลดภาระการเชื่อมต่อที่อาจเกิดจากการส่ง packets ที่ไม่จำเป็นเมื่อทราบว่าฝั่งตรงข้ามยังไม่พร้อมรับข้อมูล.

   iii. การปรับลดระยะเวลา: มีการกำหนดเพดานสำหรับระยะเวลา timeout นี้เพื่อไม่ให้มันเพิ่มขึ้นไปอย่างไม่สิ้นสุด. หาก window ขยายตัวและผู้รับพร้อมที่จะรับข้อมูลอีกครั้ง, ฝั่งผู้ส่งอาจรีเซ็ตระยะเวลานี้กลับไปเป็นค่าเริ่มต้น.

12) ฝั่งผู้ส่งทราบได้อย่างไรว่าฝั่งผู้รับพร้อมที่จะรับข้อมูลต่อแล้ว?

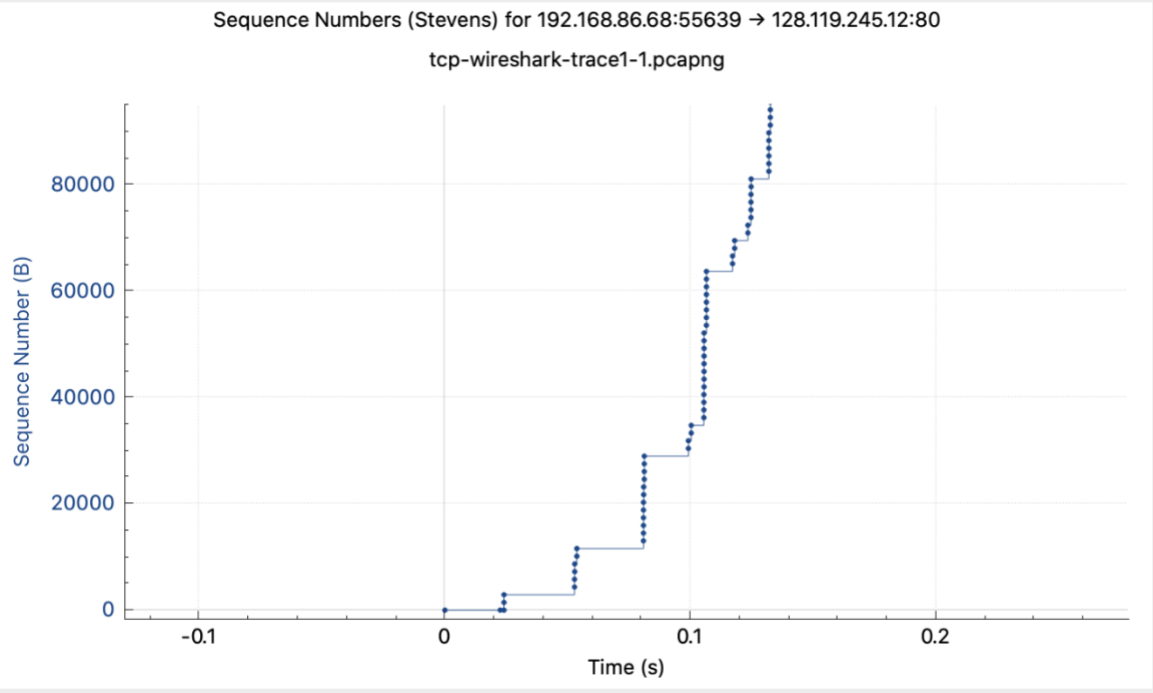
a. การรับ TCP ACK ที่มี Window Size ไม่เป็น 0: หลังจากที่ผู้ส่งได้รับ TCP ACK ที่มีค่า window size ใหม่ที่มากกว่า 0, นี่บ่งบอกว่าผู้รับมีพื้นที่ว่างใน buffer และพร้อมที่จะรับข้อมูลต่อไป. ค่า window size ที่ระบุใน TCP ACK นี้จะกำหนดปริมาณข้อมูลที่ผู้ส่งสามารถส่งได้ก่อนที่จะต้องรอ ACK ถัดไป.
b. การปรับใช้ Window Size Scaling: หากได้รับการตกลงใช้ window scaling ในการตั้งค่าเชื่อมต่อ TCP, ค่า window size ที่แสดงใน ACK จะถูกปรับขยายตาม window scaling factor ที่ตกลงกันไว้ในขั้นตอนการเชื่อมต่อ.
c. Zero Window Probes และการตอบรับ: ถ้าผู้ส่งได้ส่ง Zero Window Probes ไปยังผู้รับเพื่อตรวจสอบว่าผู้รับพร้อมที่จะรับข้อมูลหรือไม่, การได้รับ ACK ที่มีค่า window size ที่มากกว่า 0 จะหมายความว่าผู้รับพร้อมแล้ว. ในกรณีที่ผู้รับยังไม่พร้อม, ผู้ส่งจะได้รับ ACK ที่ยังคงมีค่า window size เป็น 0, ซึ่งหมายความว่าผู้ส่งควรที่จะต้องรอก่อนที่จะส่งข้อมูลต่อไป.
d. การตรวจสอบและปรับเวลา: ผู้ส่งจะต้องตรวจสอบอย่างสม่ำเสมอโดยการส่ง probes หรือตรวจสอบ ACKs ที่รับมาเพื่อระบุความพร้อมของผู้รับอย่างต่อเนื่อง.
e. การจัดการระยะเวลา: ในระหว่างที่ window size เป็น 0, ผู้ส่งจะจัดการระยะเวลาการส่งข้อมูลและ probes อย่างระมัดระวังเพื่อไม่ให้เกิดการแออัดหรือ overload ที่ผู้รับ.

B. TCP Congestion Control in Action

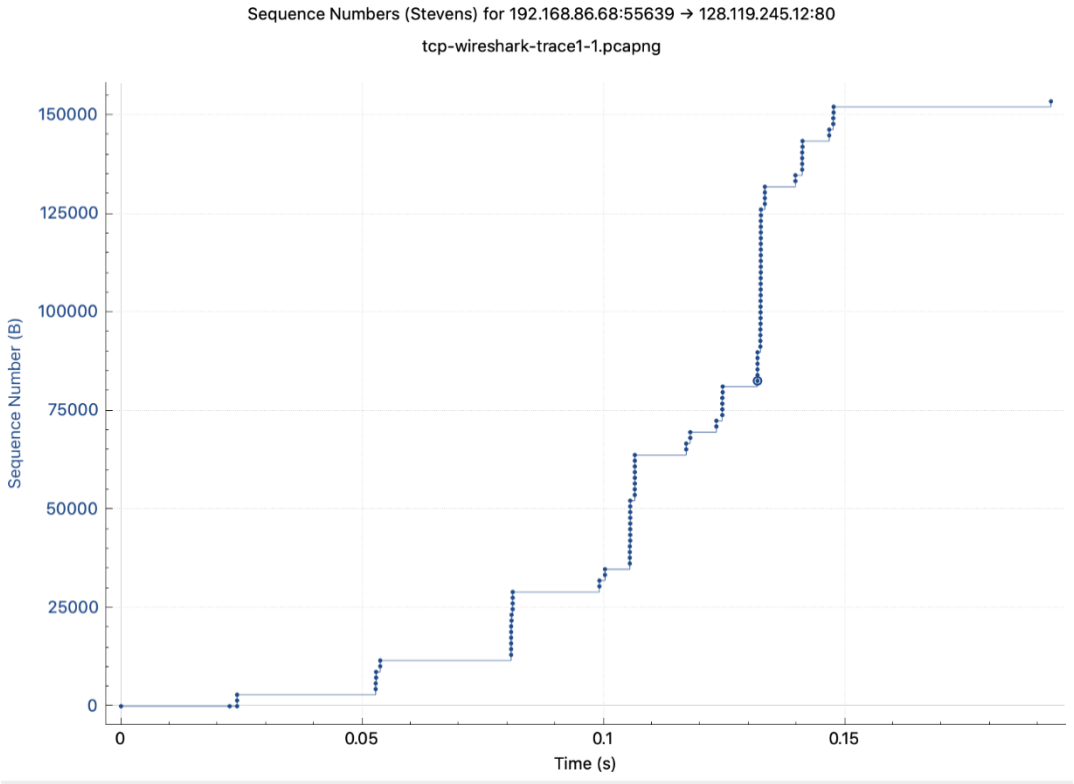
ในส่วนต่อไปนี้จะเป็นการศึกษาการทำงานของ Congestion Control ใน TCP โดยแนวทางการศึกษาจะเป็นการสังเกตปริมาณข้อมูลที่ส่งจาก client ไปยัง server ต่อหน่วยเวลา แทนที่จะคำนวณเองจากข้อมูลดิบจาก packets ผู้เรียนจะได้ใช้หนึ่งในความสามารถของ Wireshark ในการสร้างกราฟรูปแบบ Time-Sequence-Graph (Stevens)

ในการสร้างกราฟดังกล่าว ให้คลิกเลือก TCP segment ที่ client ส่งออกไปอันหนึ่งอันใดจากในหน้า Packet List Pane จากนั้นให้เลือกเมนู Statistics -> TCP Stream Graph -> Time Sequence-Graph (Stevens) ผู้เรียนควรจะเห็นหน้าต่างแสดงกราฟดังรูป 1 ปรากฏขึ้นบนหน้าจอ โดยผู้เรียนสามารถขยาย ย่อช่วงระยะเวลาที่แสดงในแต่ละแกนได้

แต่ละจุดที่พล็อตในกราฟแทนการส่งแต่ละ TCP segment ซึ่งแสดงให้เห็นถึงหมายเลข sequence number ของแต่ละ TCP segment เทียบกับเวลาที่ส่งแต่ละ TCP segment ซึ่งกลุ่มจุดที่ซ้อนเหนือจุดอื่นๆ ในแนวตั้งในแต่ละชุด แทนการส่ง packets เป็นชุดต่อเนื่องกัน (series of packets)



รูป 1 กราฟรูปแบบ Time-Sequence (Stevens) ซึ่งเป็นการพล๊อตระหว่างหมายเลข sequence number กับเวลา



รูป 2 อีกหนึ่งมุมมองของกราฟซึ่งใช้ข้อมูลชุดเดียวกันกับ รูป 1

Questions (B)

จากไฟล์ tcp-wireshark-trace1-1.pcapng พิจารณาข้อมูลการรับส่ง packets รวมถึงข้อมูลจากกราฟรูปแบบ Time-Sequence (Stevens) เพื่อตอบคำถามต่อไปนี้

13) ในขั้นตอน 3-way handshake เพื่อสร้างการเชื่อมต่อ ฝั่ง client process ประกาศว่ารองรับ Maximum Segment Size (MSS) ค่าเท่าใด? server process ประกาศว่ารอบรับ Maximum Segment Size ค่าเท่าใด?
    a. TCP Option - Maximum segment size: 1460 bytes
        i. Kind: Maximum Segment Size (2)
        ii. Length: 4
        iii. MSS Value: 1460
14) ในกรณีนี้ ระหว่าง client และ server ฝั่งใดเป็นผู้ส่งข้อมูล? หากพิจารณา TCP segment ที่นำส่งข้อมูล (data-delivering TCP segment) แต่ละ segment มีขนาดของ TCP payload เป็นเท่าใด? ค่าดังกล่าวเท่ากับ MSS ที่ฝั่งผู้รับประกาศไว้ในข้อ 13) หรือไม่? กรณีที่มีค่าไม่เท่ากันโปรดระบุว่าเป็นเพราะสาเหตุใด? (คำใบ้: โปรดสังเกต TCP header ว่ามีการใช้ options ใดหรือไม่)
    a. client ส่ง
    b. TCP payload (1448 bytes)
    c. Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
        i. TCP Option - No-Operation (NOP)
        ii. TCP Option - No-Operation (NOP)
        iii. TCP Option - Timestamps
15) จากข้อ 14) สามารถสรุปความสัมพันธ์ระหว่างของค่า Maximum Segment Size ค่าขนาดความยาวของ TCP header (TCP header length) และขนาด TCP payload ได้ว่าอย่างไร?
    a. Maximum Segment Size (MSS) คือค่าที่กำหนดขนาดสูงสุดของ TCP payload ในแต่ละ TCP segment ที่สามารถส่งได้
    b. TCP payload (1448 bytes)
    c. Options: (12 bytes) + Header (20 bytes)
16) จากกราฟรูปแบบ Time-Sequence (Stevens) จงพิจารณาชุดของ packets ที่ส่งต่อเนื่องกันในช่วงเวลาใกล้เคียงกับเวลาดังนี้ t = 0.024, t = 0.053, t = 0.081 และ t = 0.1 ในแต่ละช่วงเวลาที่ส่งสามารถอนุมานว่า TCP ทำงานอยู่ใน slow start phase, congestion avoidance phase หรือ phase อื่นใด?

a. t = 0.024: หากเราเห็นว่า sequence numbers เพิ่มขึ้นอย่างรวดเร็วในช่วงเวลานี้, นั่นอาจหมายถึงการที่ TCP อยู่ใน slow start phase.
b. t = 0.053, t = 0.081: ถ้าการเพิ่มขึ้นของ sequence numbers ยังคงมีแนวโน้มที่รวดเร็วและเป็นแบบ exponential, นั่นหมายความว่า TCP ยังคงอยู่ใน slow start phase.
c. t = 0.1: หากเริ่มมีการเพิ่มขึ้นของ sequence numbers ที่ชะลอลงและเปลี่ยนเป็นแบบ linear มากกว่า exponential, นี่อาจเป็นสัญญาณของการเปลี่ยนไปเป็น congestion avoidance phase.

17) การส่ง TCP segment แต่ละชุดจากที่ปรากฏในกราฟ สามารถสังเกตได้ว่ามีรอบการส่งออกไปเป็นระยะๆ ช่วงรอบระยะเวลาดังกล่าวสามารถบ่งบอกถึงอะไรได้?
    a. สภาพแวดล้อมในเครือข่าย: การสังเกตช่วงรอบการส่งของ TCP segments อาจช่วยให้เห็นภาพรวมเกี่ยวกับสภาพแวดล้อมในเครือข่าย เช่น ความหนาแน่นของการส่งข้อมูล, ปริมาณข้อมูลที่ถูกส่งไปยังหรือมาจากเครือข่าย, หรือการเปลี่ยนแปลงของความเร็วในการส่งข้อมูล (throughput).
    b. การทำงานของ TCP: การสังเกตช่วงรอบการส่งของ TCP segments อาจช่วยให้เห็นการทำงานของ TCP congestion control algorithm ได้ดีขึ้น เช่น การเริ่มต้นใน slow start phase โดยการเพิ่มขึ้นของ congestion window แบบ exponential, หรือการเปลี่ยนไปยัง congestion avoidance phase โดยการเพิ่มขึ้นของ congestion window แบบเชิงเส้น.
    c. การแปรผันของสถานะของเครือข่าย: การสังเกตช่วงรอบการส่งของ TCP segments อาจช่วยให้เห็นการเปลี่ยนแปลงในสถานะของเครือข่าย เช่น การสูญเสียข้อมูล, การเพิ่มหรือลดความเร็วในการส่งข้อมูล (congestion window size), หรือการปรับปรุง latency.

18) หลังจากสร้างการเชื่อมต่อสำเร็จ ฝั่งผู้ส่งข้อมูล ส่งข้อมูลต่อเนื่องออกไปเป็นจำนวนกี่ segment โดยที่ไม่ต้องรอ acknowledgement?
    a. การส่งข้อมูลโดยไม่ต้องรอ acknowledgement เป็นส่วนหนึ่งของขั้นตอนใน TCP ที่เรียกว่า "fast transmission".
    b. จำนวน segment ที่ส่งต่อเนื่องโดยไม่ต้องรอ ack จะขึ้นอยู่กับ congestion window size, advertised window size, และความเสี่ยงในการสูญเสียข้อมูลในเครือข่าย

Submission

จงตอบคำถามในส่วนที่ระบุหัวข้อ Question ตั้งแต่ (A) ไปจนถึง (B) ซึ่งมีคำถามร่วมทั้งหมด 18 ข้อ โดยในคำตอบของแต่ละข้อด้วยให้อธิบายด้วยว่าหาคำตอบมาได้อย่างไร ตัวอย่างเช่น อธิบายว่าสามารถค้น packet ตามที่โจทย์ระบุได้ด้วยวิธีการใด หรือค่าที่นำมาตอบ นำมาจาก field ใดของ header ตาม protocol ใด

ในกรณีที่คัดลอกคำตอบของคนอื่นมา ให้ระบุชื่อของบุคคลที่เป็นต้นฉบับมาด้วย หากตรวจพบว่ามีการลอกมาแต่ไม่มีการระบุชื่อบุคคลที่เป็นต้นฉบับ ผู้สอนจะถือว่าทุจริตและอาจพิจารณาลงโทษให้ตกเกณฑ์รายวิชาในทันที

การส่งงาน ให้เขียนหรือพิมพ์หมายเลขข้อและคำตอบของข้อนั้นๆ และส่งเป็นไฟล์ PDF เท่านั้น กรุณาตั้งชื่อไฟล์โดยใช้รหัสนักศึกษา ตามด้วย section และ _lab07 ตามตัวอย่างต่อไปนี้ 64019999_sec20_lab07.pdf