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?

   3, 3, 3, 3, 3
   ทำการเพิ่มcodeให้printจำนวนครั้งออกมา

2) จากไฟล์ Lab07-A1.txt พบว่าฝั่ง client มีการวนซ้ำเพื่ออ่านข้อมูลจาก receive buffer ด้วยคำสั่ง clientSocket.recv เป็นจำนวนกี่ครั้ง? แต่ละครั้งได้ข้อความใดบ้าง?

   1
   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.
   นับจากที่ตัวpython printออกมา

3) จากไฟล์ Lab07-A2.txt พบว่าฝั่ง client มีการวนซ้ำเพื่ออ่านข้อมูลจาก receive buffer ด้วยคำสั่ง clientSocket.recv เป็นจำนวนกี่ครั้ง? แต่ละครั้งได้ข้อความใดบ้าง?

   3
   The Internet is literally a network of networks.The Internet liv
   es where anyone can access it.The idea that you can somehow eras
   e the Internet is silly.
   นับจากที่ตัวpython printออกมา

4) จากข้อ 1) ข้อ 2) และข้อ 3) ในการส่งข้อมูลผ่าน TCP ฝั่งผู้รับข้อมูลทราบหรือไม่ว่าฝั่งผู้ส่งเรียกฟังก์ชัน send เพื่อส่งข้อมูลเป็นจำนวนกี่ครั้ง? และผู้รับทราบหรือไม่ว่าผู้ส่งเรียกฟังก์ชัน send แต่ละครั้งส่งข้อมูลเท่าใดและสิ้นสุดลงที่ใด?

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	32768	1048576	32
Lab07-A2.pcapng	1048576	32768	1048576	32
Lab07-A3.pcapng	16777216	32768	16777216	512
Lab07-A4.pcapng	32	32	32	1
Lab07-A5.pcapng	32	32	32	1

ดูได้จาก TCP protocol ใน packet detail pane

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

ส่งไปตอนส่งSYNไปขอเปิดconnection ดูได้จาก TCP protocol ใน packet detail pane

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

Calculated window size = Window * Window size scaling factor

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?

Packetที่4 ขนาด 32 bytes

ดูได้จาก info ใน packet list pane

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

A4 packet 9, 11, 13, 18
A5 packet 9, 11, 13, 15, 20, 22, 24, 26
tcp.window_size == 0

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

ส่ง TCP Zero Window Probe โดยมีpayloadเป็นตัวc

ดูจาก packet detail paneและinfoในpacket list pane

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

ไม่เท่ากันในแต่ละครั้งมีการเว้นมากกว่าเดิม2เท่า

ในกรณีของผู้เรียนใช้การSet time refเพื่อเทียบเวลาพบว่าครั้งแรกเป็น0.6sครั้งที่2เป็น1.2s

12) ฝั่งผู้ส่งทราบได้อย่างไรว่าฝั่งผู้รับพร้อมที่จะรับข้อมูลต่อแล้ว?

ผู้รับส่ง TCP Window Update

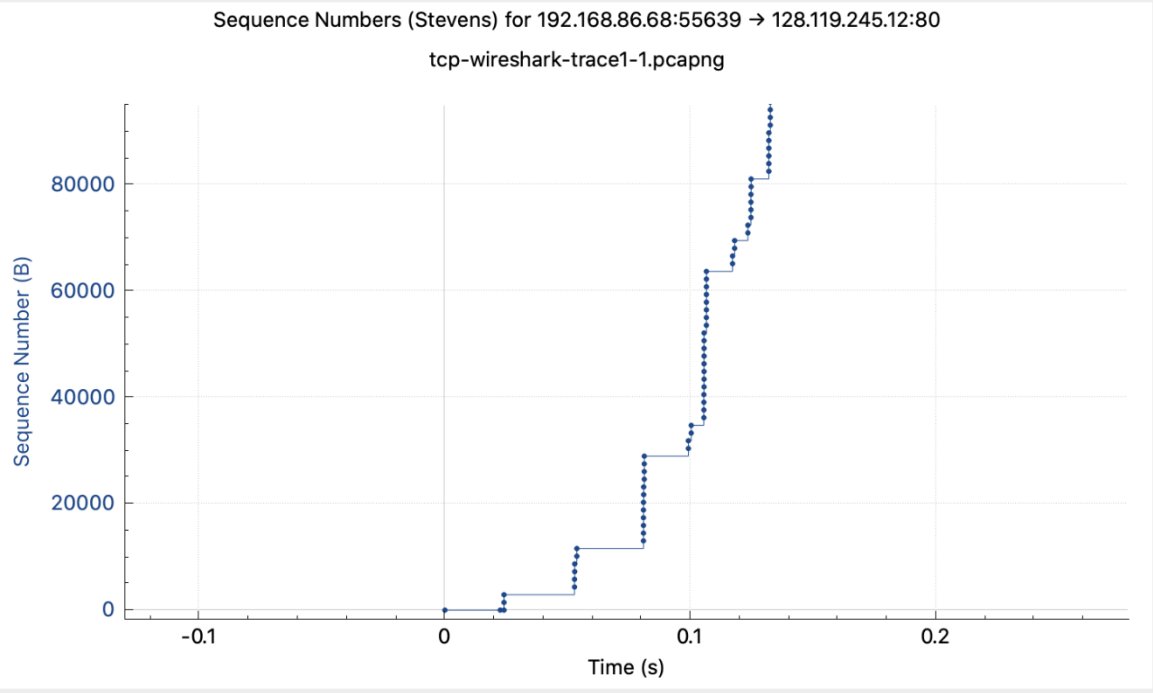
ดูได้จากลำดับการส่งในinfoของpacket list pane

B. TCP Congestion Control in Action

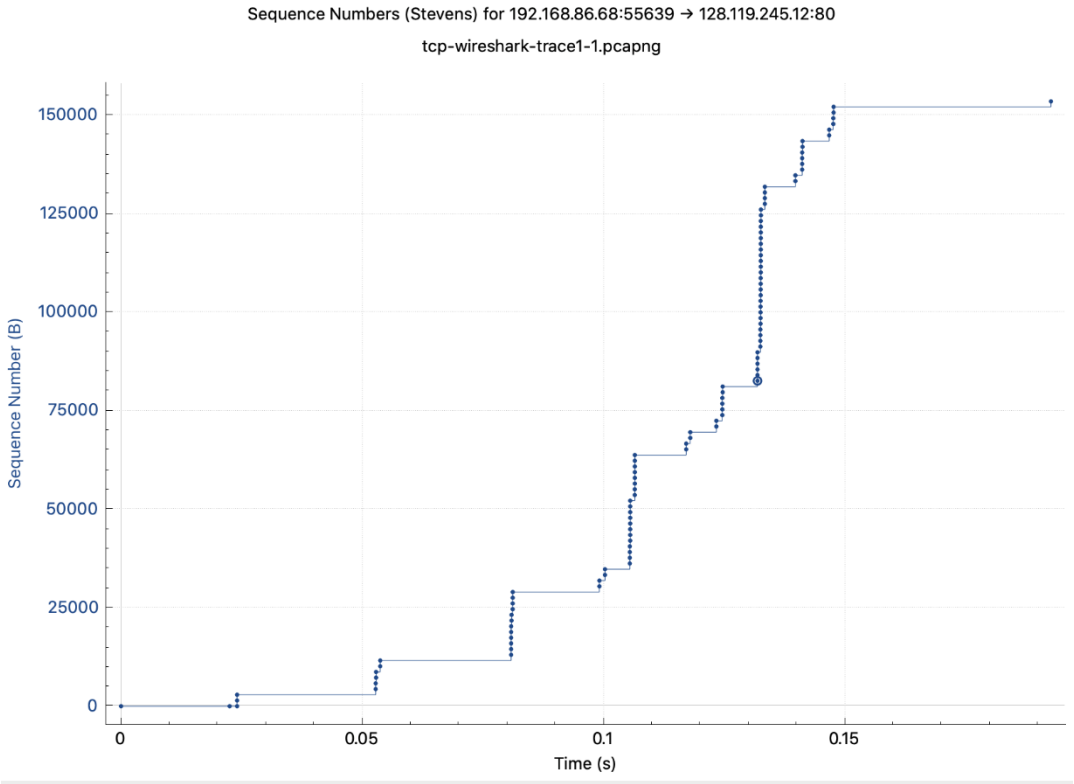
ในส่วนต่อไปนี้จะเป็นการศึกษาการทำงานของ Congestion Control ใน TCP โดยแนวทางการศึกษาจะเป็นการสังเกตปริมาณข้อมูลที่ส่งจาก client ไปยัง server ต่อหน่วยเวลา แทนที่จะคำนวณเองจากข้อมูลดิบจาก packets ผู้เรียนจะได้ใช้หนึ่งในความสามารถของ Wireshark ในการสร้างกราฟรูปแบบ Time-Sequence-Graph (Stevens)

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?

    1460 และ 1460

    ดูได้จาก packet list pane

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)

    Client เป็นผู้ส่ง ดูได้จากpacket HTTPสังเกตใน packet detail pane มีการระบุไว้ว่ารวมมาจากframeใดบ้าง

    แต่ละ segment มีpayloadขนาด 1448 bytes มีความสัมพันธ์กับค่าMSSแต่ไม่ใช่ค่าMSSค่า

    ค่า MSS เป็น 1460 แต่เนื่องจากมีoptionเสริมจึงเหลือเพียง 1448

15) จากข้อ 14) สามารถสรุปความสัมพันธ์ระหว่างของค่า Maximum Segment Size ค่าขนาดความยาวของ TCP header (TCP header length) และขนาด TCP payload ได้ว่าอย่างไร?

    MSS = (TCPHeader – 20) + Payload

    สืบค้นจากinternet(กล่าวว่าMTU(1500)=IPHeader(20) + TCPHeader(20) + Payload(1460) หากTCP Header มี option TCP Payload ต้องลดขนาดลง

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?

    slow start phase เนื่องจากมีการเพิ่มขึ้นเป็น2เท่าเรื่อยๆ

17) การส่ง TCP segment แต่ละชุดจากที่ปรากฏในกราฟ สามารถสังเกตได้ว่ามีรอบการส่งออกไปเป็นระยะๆ ช่วงรอบระยะเวลาดังกล่าวสามารถบ่งบอกถึงอะไรได้?

    RTT

18) หลังจากสร้างการเชื่อมต่อสำเร็จ ฝั่งผู้ส่งข้อมูล ส่งข้อมูลต่อเนื่องออกไปเป็นจำนวนกี่ segment โดยที่ไม่ต้องรอ acknowledgement?

3 segments ดูได้จากชุดแรกที่ส่งไป(t=0.024)

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