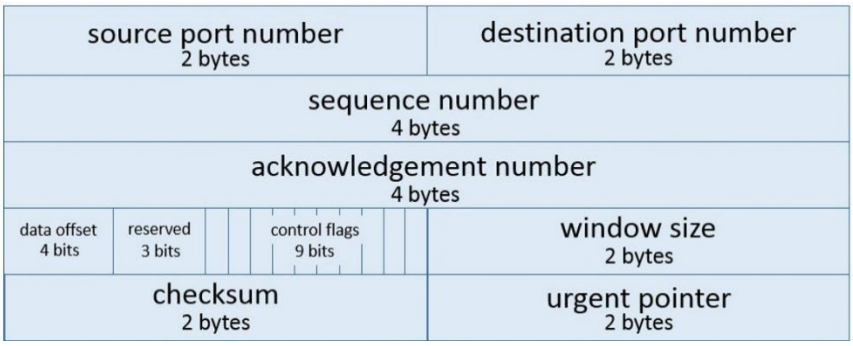
ปีระ วิริยะกุล 63010871 SEC 03
(ห้อง LAB : จ. 13.00-16.00)

กิจกรรมที่ 6 : TCP Connection

กิจกรรมครั้งนี้จะเป็นการทำความเข้าใจกับโปรโตคอล TCP (Transmission Control Protocol) ซึ่ง TCP มีคุณสมบัติในการทำงานอยู่ 5 ประการได้แก่

- Reliable, in-order delivery คือ การส่งไม่ผิดพลาดโดยข้อมูลมีการเรียงตามลำดับ
- Connection Oriented คือ ต้องมีการสร้างการเชื่อมต่อก่อน และมีการแลกเปลี่ยนข้อมูลควบคุม
- Flow Control ควบคุมการไหลของข้อมูลระหว่าง Process ทั้ง 2 ด้าน
- Congestion Control ควบคุมการไหลของข้อมูลผ่านอุปกรณ์เครือข่าย
- Full Duplex data สามารถส่งได้ทั้ง 2 ทาง ในการเชื่อมต่อเดียวกัน

Connection Setup

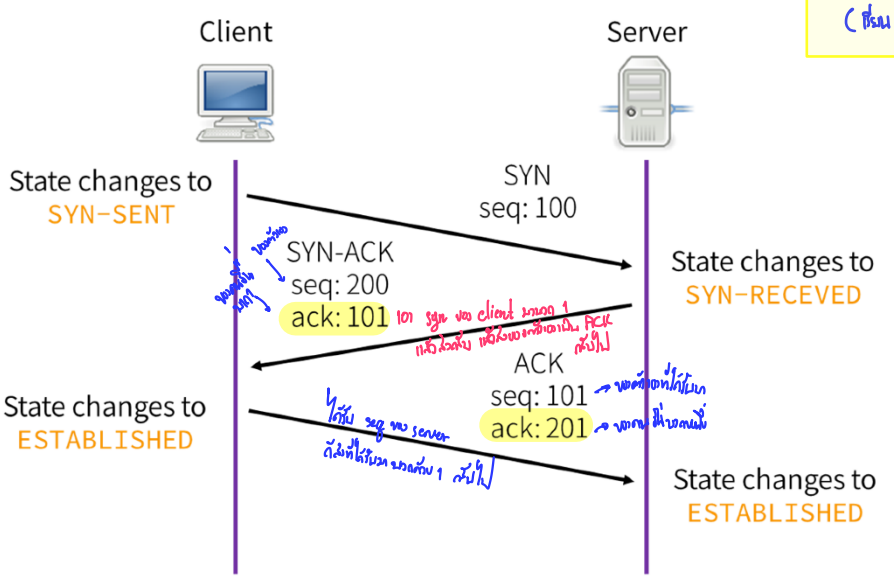


รูปแสดง TCP Header

ก่อนเริ่มการส่งข้อมูลทุกครั้งของ TCP จะต้องมีการสร้าง Connection ขึ้นมาก่อนโดย Client จะเริ่มสร้างการเชื่อมต่อไปที่ Server ซึ่งประกอบด้วย 3 ขั้นตอน

- Client การส่ง packet SYN ไปที่ Server โดย Client จะมีการสร้างหมายเลข Sequence Number เรียกว่า ISN : Initial Sequence Number ขึ้นมา (ในรูปสมมติว่า 100) ใส่ใน SEQ# แล้วส่ง
- เมื่อ Server ได้รับ packet SYN จะตอบกลับโดย packet SYN-ACK โดย Server จะมีการสร้างหมายเลข ISN ของตนเองขึ้นมาเช่นกัน โดยใส่ใน SEQ# และนำหมายเลข SN:Client+1 แล้วใส่ใน ACK# แล้วส่ง
- เมื่อ Client ได้รับ packet SYN-ACK ก็จะตอบกลับโดย packet ACK สุดท้าย โดย Client จะนำ SN:Client+1 ใส่ใน SEQ# และนำ SN:Server+1 ใส่ใน ACK# แล้วส่ง เมื่อถึงตรงนี้จะถือว่าฝั่ง Client สร้างการเชื่อมต่อสำเร็จแล้ว ซึ่ง Client สามารถจะเริ่มส่งข้อมูลได้
- เมื่อ Server ได้รับ packet ACK สุดท้าย จะถือว่าฝั่ง Server สร้างการเชื่อมต่อสำเร็จแล้วเช่นกัน

ปิยะ วิริยะกุล 63010871 SEC 03
(เรียน LAB : จ. 13.00-16.00)



1. ให้เปิดไฟล์ http-browse101d.pcapng ค้นหา 3 way handshake แรกในไฟล์แล้ว บันทึกข้อมูลลงในตารางด้านล่าง (ทั้ง Seq# และ Ack# ให้ใช้แบบ raw ในช่อง Flag ให้บอกว่ามี Flag ใดที่ Set บ้าง

SYN

Src Port :	61598	Dest Port :	80
Seq # : (raw)	610997682		
Ack # : (raw)	0		
Flags : (set)	Syn	TCP Flags :	.....s.

SYN-ACK

Src Port :	80	Dest Port :	61598
Seq # : (raw)	4134094401		
Ack # : (raw)	610997683 (เพิ่มจาก client ขึ้น 1 เพื่อบอกสัญญาณ ACK)		
Flags : (set)	Syn, Ack	TCP Flags :	....A..s.

ACK

Src Port :	61598	Dest Port :	80
Seq # : (raw)	610997683 (ตัวเลขเท่าเดิมจากที่ส่งไป)		
Ack # : (raw)	4134094402 (ตัวเลข server เพิ่มขึ้น 1 แล้วส่งกลับ)		
Flags : (set)	Ack	TCP Flags :	....A....

- ค่าความยาวข้อมูลของ packet ทั้ง 3 เท่ากับเท่าไรบ้าง 66B, 66B, 54B ตามลำดับ
- ใน packet SYN มีข้อมูลอื่นๆ ส่งมาด้วยหรือไม่ อะไรบ้าง (ดูในคอลัมน์ info) และข้อมูลต่างๆ เหล่านั้นมีความหมายอะไรหรือนำไปใช้อะไร (ให้ค้นหาข้อมูลเพิ่มเติมจากหนังสือ)

ธีรระ วิริยะกุล 63010871 SEC 03
(เรียน LAB : พ. 13.00-16.00)

ข้อมูล	ความหมาย
Win = 8192 (2^{13})	Window size : จำนวน Byte ที่ผ่านไปได้ก่อนที่จะทำการตอบกลับ / จำนวน Byte ที่มันส่งมาได้ ก่อนที่จะรอการตอบกลับ
MSS = 1460 Bytes	Maximum segment size : ขนาดข้อมูลสูงสุดที่ client กับ server รับรู้ร่วมกันว่า สามารถ LOAD / TRASFER ได้
WS = 4 ?	Window scaling : scale window ฝั่ง Sender
SACK_PERM = 1 ?	Selective Ack permitted :

มากกว่าค่านี้ทำให้เกิน ขนาดของ MTU ใน L2 ถ้ามันเกิน packet จะโดน Drop หายไประหว่างทาง (traceroute ก็ใช้สังเกตได้ (ไม่เห็นว่า loss))

MTU : Maximum Transmission Unit.

– ใน packet SYN-ACK มีข้อมูลอื่นๆ ส่งมาด้วยหรือไม่ อะไรบ้าง (ดูในคอลัมน์ info) และข้อมูลต่างๆ เหล่านั้นมีความหมายอะไรหรือนำไปใช้อะไร

ข้อมูล	ความหมาย
Win = 14600	Window size : จำนวน Byte ที่ผ่านไปได้ก่อนที่จะทำการตอบกลับ / จำนวน Byte ที่มันส่งมาได้ ก่อนที่จะรอการตอบกลับ
MSS = 1460 Bytes	Maximum segment size : ขนาดข้อมูลสูงสุดที่ client กับ server รับรู้ร่วมกันว่า สามารถ LOAD / TRASFER ได้
WS = 64 ?	Window scaling : scale window ฝั่ง Receiver
SACK_PERM = 1 ?	Selective Ack permitted :

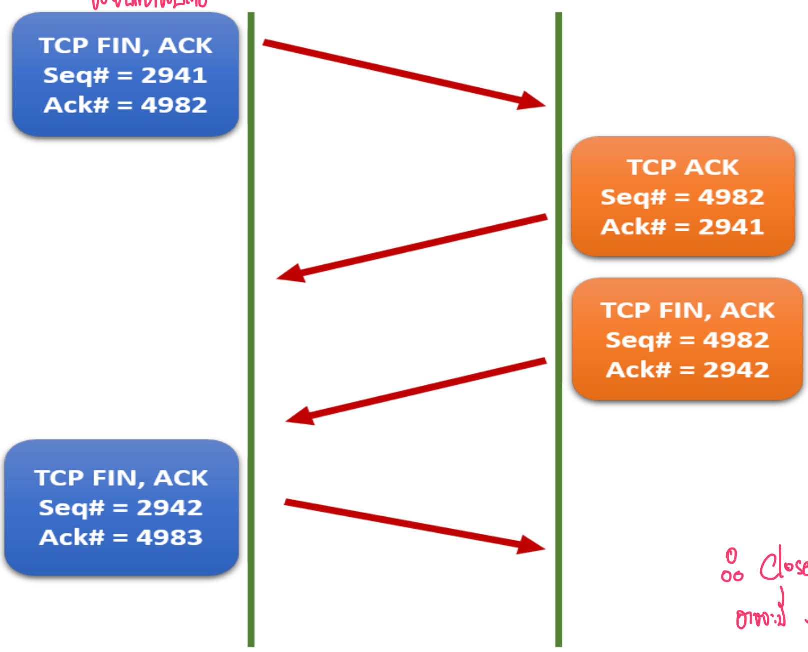
– ให้ดู packet ที่ส่งข้อมูล packet แรก (หรือ packet อื่นก็ได้) ให้ตอบว่าในข้อมูลที่ไม่เท่ากันของ Client กับ Server ในการเลือกใช้ข้อมูลหนึ่ง (เนื่องจากทั้ง 2 ด้านต้องใช้พารามิเตอร์เดียวกันในการส่งข้อมูล) คิดว่ามีหลักในการเลือกอย่างไร

เลือกตามความต้องการของ sender, เช่น packet ที่มีขนาดเล็ก เมื่อได้เลือก eff line สูงสุด

Connection Terminated

เมื่อสิ้นสุดการส่งข้อมูลแล้ว ใน TCP จะมีการปิด Connection ซึ่งประกอบด้วย 4 ขั้นตอน

ขอจบการเชื่อมต่อ



บางครั้ง 2 packet ที่จะถูกยุบรวมกัน ดังนั้น ลำดับไม่แน่นอน

ดังนั้น Close connection อาจจะมี 3-4 packets

[illegible] 63010871 SEC 03
(เรียน LAB : ศ. 13.00–16.00)

- ฝ่ายใดฝ่ายหนึ่งที่ต้องการปิด Connection (ต่อไปจะเรียก A และเรียกอีกฝั่งว่า B) จะส่ง packet ที่มี FIN/ACK flag มา โดยใช้ SEQ# และ ACK# เท่ากับ packet สุดท้ายก่อนจะปิด connection
- ฝั่ง B จะตอบด้วย packet ที่มี ACK flag โดยใช้ SEQ# เท่ากับ ACK# ของ FIN/ACK ก่อนหน้า และใช้ ACK# เท่ากับของ SYN# ของ packet ล่าสุด โดยเมื่อ A ได้รับ packet นี้ จะถือว่าเป็นการสิ้นสุด connection ของฝั่ง A (**หมายเหตุ** บางครั้งอาจไม่มีการส่ง packet นี้ โดยอาจรวมไปกับ packet ที่ 3
- ฝั่ง B จะเริ่มปิด Connection บ้าง โดยจะส่ง packet ที่มี FIN/ACK flag โดยใช้ SEQ# เท่ากับ ACK# ของ FIN/ACK ก่อนหน้า และใช้ ACK# เท่ากับของ SYN# ของ packet ล่าสุด +1
- ฝั่ง A จะตอบกลับการปิด Connection โดยจะส่ง packet ที่มี FIN/ACK flag โดยใช้ SEQ# เท่ากับ ACK# ของ FIN/ACK ก่อนหน้า และใช้ ACK# เท่ากับของ SYN# ของ packet ล่าสุด +1 เมื่อถึงจุดนี้จะถือว่าเป็นการสิ้นสุด Connection ของ B

2. ให้หา Packet ที่ปิด Connection ของ Connection ในข้อ 1 โดยให้บอกขั้นตอนการหาและป้อนรายละเอียดลงในตาราง (ข้อมูล Seq# และ Ack # ให้ใช้แบบ Relative)

Packet# 1663	
Src Port : 80	Dest Port : 61598
Seq # : (relative) 923	
Ack # : (relative) 1127	
Flags : (set) Fin, Ack	TCP Flags : ....A...F

Packet# 1664	
Src Port : 80	Dest Port : 61598
Seq # : (relative) 1127	
Ack # : (relative) 924	
Flags : (set) Fin, Ack	TCP Flags : ....A...F

Packet# 1665	
Src Port : 61598	Dest Port : 80
Seq # : (relative) 924	
Ack # : (relative) 1128	
Flags : (set) Ack	TCP Flags : ....A....

วิธีค้นหา

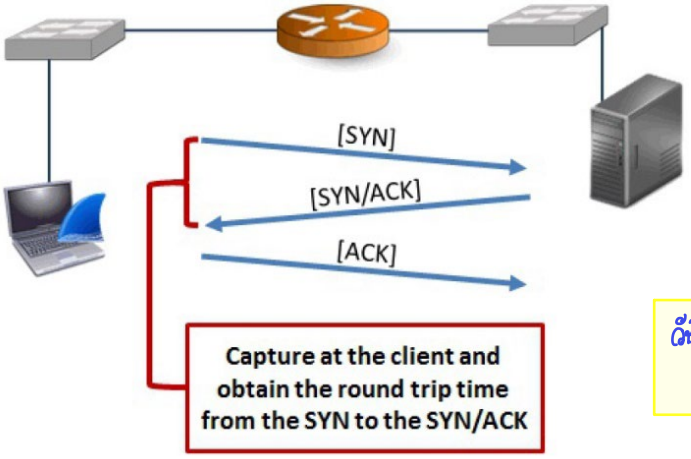
follow tcp stream ของ packet ที่เป็น packet เริ่ม connection จากนั้นก็ทำการ filout this stream

แล้วจะได้ filter : !(tcp.stream eq 0) แต่ถ้าทำการแก้ไข filter ให้แสดงเฉพาะ stream นี้ต้องเอา ! ออก ได้เป็น tcp.stream eq 0

↳ เอา stream นี้ออกไปหมด ↑ แสดงเฉพาะ stream นี้.

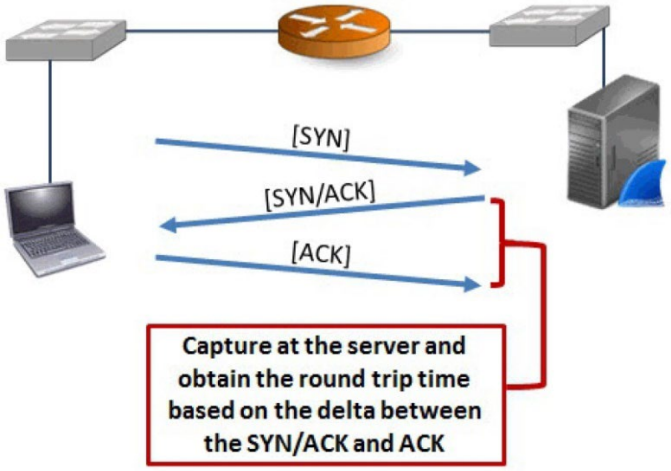
โดย การ close tcp connection จะอยู่ที่ 3-4 packet สุดท้าย เลือกจาก seq, Ack ตามทฤษฎี

3. ใน Wireshark เราสามารถจะหา packet ที่มีคุณลักษณะของ flags เฉพาะได้ โดยใช้ display filter tcp.flags เช่น **tcp.flags.syn==1** หรือ **tcp.flags.ack==1** ซึ่งเราสามารถใช้หา RTT ของ TCP handshake ได้ โดยการหา RTT ของ TCP handshake มี 3 แบบ คือ วัดจากฝั่ง Client จะใช้เวลาระหว่าง SYN และ SYN-ACK

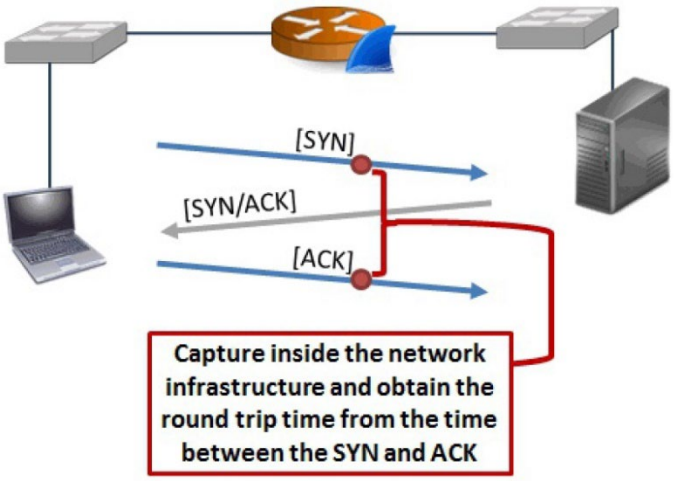


ปิยะ ศิริมงคล 63010871 SEC 03
(เรียน LAB : จ. 13.00-16.00)

และวัดจากฝั่ง Server จะใช้เวลาระหว่าง SYN/ACK กับ ACK



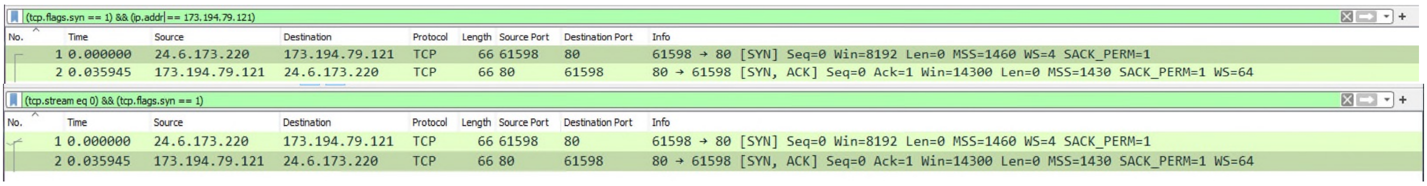
แต่ในกรณีที่วัดจากอุปกรณ์ ควรใช้ระหว่าง SYN และ ACK ตามรูป



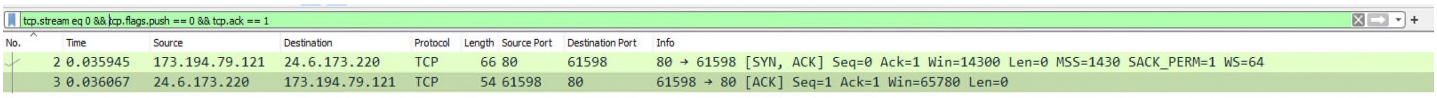
ธีรัช วีระมงคล 63010871 SEC 03
(ชั้น LAB : พ. 13.00-16.00)

4. จากไฟล์ http-browse101d.pcapng ให้สร้าง display filter ที่สามารถแสดงเฉพาะ packet ที่เป็น Open Connection (3 way handshake) คู่ที่กำหนด ของทุกๆ TCP Stream โดยไม่มี packet อื่นๆ มาปน (นักศึกษาพยายามคิดด้วยตนเอง) ให้เขียนวิธีการหา และ display filter ของแต่ละอัน
   - packet SYN และ SYN/ACK ของ 3 way handshake (packet ที่ 1 และ 2)
   - packet SYN/ACK และ ACK ของ 3 way handshake (packet ที่ 2 และ 3)
   - packet SYN และ ACK 3 way handshake (packet ที่ 1 และ 3)

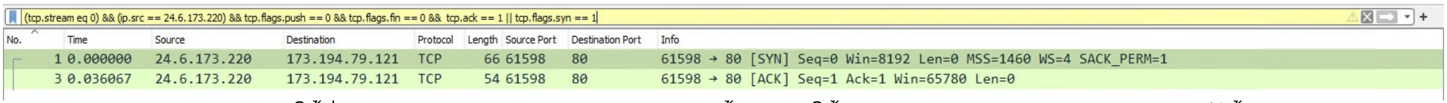
1,2



2,3



1,3



5. เราสามารถใช้ค่า RTT ของ TCP handshaking ตามข้อ 4 มาใช้วัดประสิทธิภาพของ Web Server ได้เช่นกัน โดย Server ที่มีค่า RTT น้อย แสดงถึงการตอบสนองที่รวดเร็ว ดังนั้นให้ capture ข้อมูลจากเว็บ และใช้ display filter ตามข้อ 4 (ให้นักศึกษาเลือกใช้ตัวที่เหมาะสม) เพื่อหาค่า RTT ของเว็บต่างๆ จำนวน 3 เว็บ แล้วนำค่ามาใส่ตาราง

URL	เวลา
www.apple.com	0.030552 sec.
www.reg.kmitl.ac.th	0.019127 sec.
www.ce.kmitl.ac.th	0.290526 sec.

   - ให้ตอบว่าระหว่าง RTT ที่วัดในครั้งนี้ กับ HTTP RTT ที่วัดในครั้งก่อนหน้านี้ บอกถึงอะไร และแตกต่างกันอย่างไร

RTT ในแต่ละครั้งเป็นระยะเวลาระหว่าง อุปกรณ์ 3 ตัว transfer
แต่ HTTP RTT สื่อถึงระยะเวลาที่ใช้ในการ request จนกว่าจะได้เว็บไซต์

งานครั้งที่ 6

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