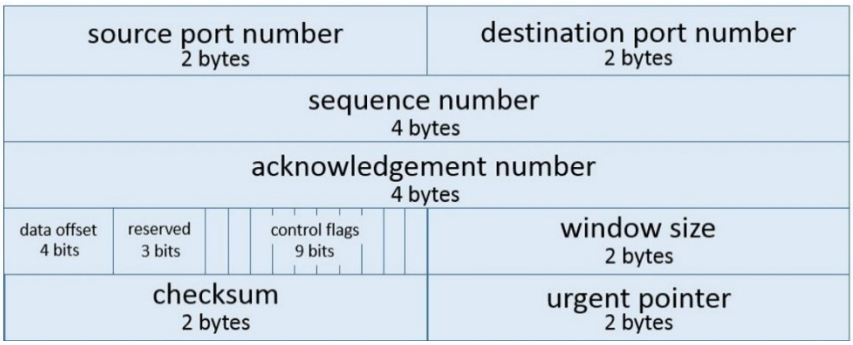
กิจกรรมที่ 6 : TCP Connection

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

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

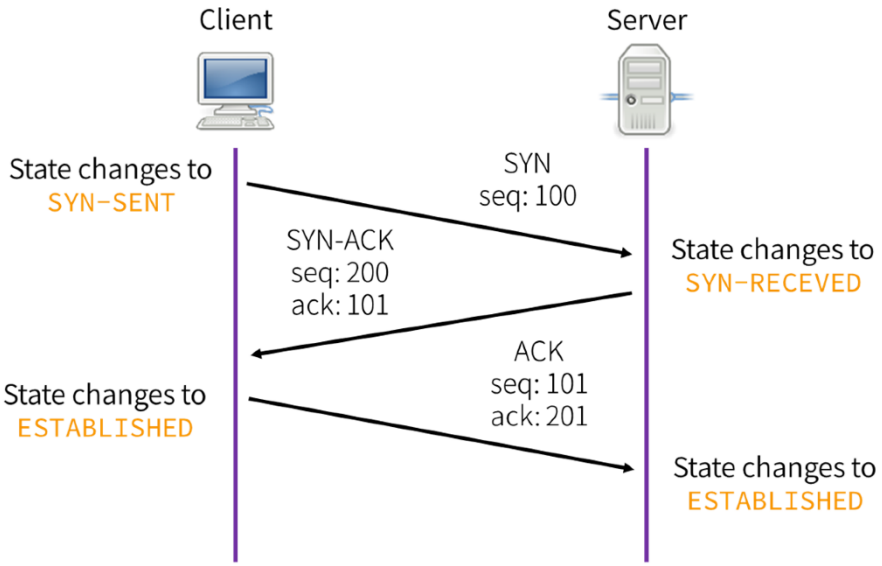


รูปแสดง TCP Header

TCP Connection Setup (TCP 3-way Handshake)

ก่อนเริ่มการส่งข้อมูลทุกครั้งของ TCP จะต้องมีการสร้าง Connection ขึ้นมาก่อนโดย Client จะเริ่มสร้างการเชื่อมต่อไปที่ Server ซึ่งประกอบด้วยการรับส่ง TCP segment ระหว่าง Client-Server จำนวน 3 TCP segments

- Client ส่ง TCP segment ที่เซต SYN flag ไปที่ Server โดย Client จะสร้างหมายเลข Sequence Number เรียกว่า Initial Sequence Number (ISN) ขึ้นมา (ในรูปสมมติว่า 100) ใส่ใน SEQ# แล้วส่ง
- เมื่อ Server ได้รับ TCP segment ที่เซต SYN flag แล้วจะตอบกลับไปด้วย TCP segment ที่เซต SYN-ACK flags โดย Server จะมีการสร้างหมายเลข ISN ของตนเองขึ้นมาเช่นกัน โดยใส่ใน SEQ# และนำหมายเลข SN:Client+1 แล้วใส่ใน ACK# แล้วส่ง
- เมื่อ Client ได้รับ TCP segment ที่เซต SYN-ACK flags ก็จะตอบกลับด้วย TCP segment ที่เซต ACK flag ซึ่งถือเป็น TCP segment สุดท้ายในการสร้าง TCP Connection โดย Client จะนำ SN:Client+1 ใส่ใน SEQ# และนำ SN:Server+1 ใส่ใน ACK# แล้วส่ง เมื่อส่ง TCP segment ดังกล่าวออกไปแล้ว จะถือว่าฝั่ง Client สร้างการเชื่อมต่อสำเร็จแล้ว ซึ่ง Client สามารถจะเริ่มส่งข้อมูลได้
- เมื่อ Server ได้รับ TCP segment สุดท้ายในการสร้าง TCP Connection ซึ่งมี ACK flag เซตเอาไว้ จะถือว่าฝั่ง Server สร้างการเชื่อมต่อสำเร็จแล้วเช่นกัน



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 # : 610997682	
Ack # : 0	
Flags : SYN	Window Size : 8192

SYN-ACK

Src Port : 80	Dest Port : 61598
Seq # : 4134094401	
Ack # : 610997683	
Flags : SYN/ACK	Window Size : 14300

ACK

Src Port : 61598	Dest Port : 80
Seq # : 610997683	
Ack # : 4134094402	
Flags : ACK	Window Size : 16445 [factored: 65780]

- ค่าความยาวข้อมูลของ packet ทั้ง 3 เท่ากับเท่าไรบ้าง

  Packet Length เป็น 66, 66, 54 ตามลำดับ ส่วน Payload เป็น 0 ทั้งหมด
- ใน packet ที่เซต SYN flag มีข้อมูลอื่นๆ ส่งมาด้วยหรือไม่ อะไรบ้าง (ดูในคอลัมน์ info) และข้อมูลต่างๆ เหล่านั้นมีความหมายอะไรหรือนำไปใช้อะไร (ให้ค้นหาข้อมูลเพิ่มเติมจากหนังสือ)

ข้อมูล	ความหมาย
Len = 0	ขนาดของ Payload ของ TCP Segment
MSS = 1460	Maximum Segment Size ขนาดของ Segment ที่มากที่สุด
Windows Scale = 2	ตัวคูณของ Window Size (x4)
SACK permitted	สามารถใช้ Selective ACK ได้

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

ข้อมูล	ความหมาย
Len = 0	ขนาดของ Payload ของ TCP Segment
MSS = 1430	Maximum Segment Size ขนาดของ Segment ที่มากที่สุด
Windows Scale = 6	ตัวคูณของ Window Size (x64)
SACK permitted	สามารถใช้ Selective ACK ได้

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

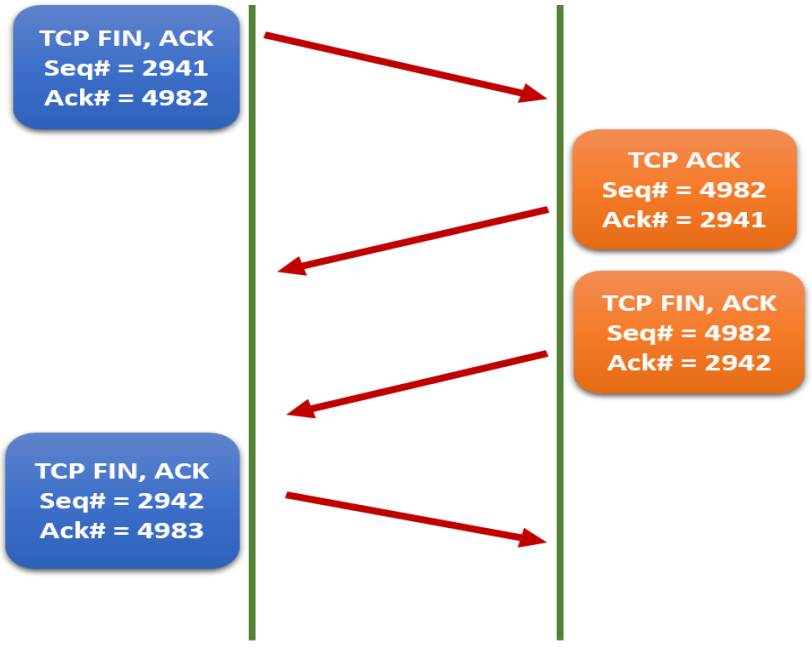
  SACK ถ้า permit ทั้งสองด้าน จะสามารถใช้ selective ack ได้

  MSS เป็นค่าที่แลกเปลี่ยนระหว่างคู่สนทนา เพื่อแจ้งไปยังอีกฝั่งว่ารองรับ MSS เท่าใด ทั้งสองฝั่งไม่ได้จำเป็นต้องใช้ค่าเดียวกัน ไม่ได้เป็นการตกลงค่าเดียวกัน (RFC879 section 3)

  Window Size เป็นค่าที่แลกเปลี่ยนระหว่างคู่สนทนา เพื่อแจ้งไปยังอีกฝั่งว่าตนมีพื้นที่ว่างพร้อมรับข้อมูลเท่าใด **ถ้าตอบ SACK ถูกให้ถือว่าเป็นคำตอบที่ถูกต้อง**

TCP Connection Termination (หรือ TCP Connection Teardown)

เมื่อสิ้นสุดการส่งข้อมูลแล้ว ใน TCP จะมีการปิด Connection ซึ่งประกอบด้วย 4 ขั้นตอน



- 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 : 61598	Dest Port : 80
Seq # : 323	
Ack # : 1127	
Flags : FIN/ACK	Window Size : 16163 [factored: 64652]

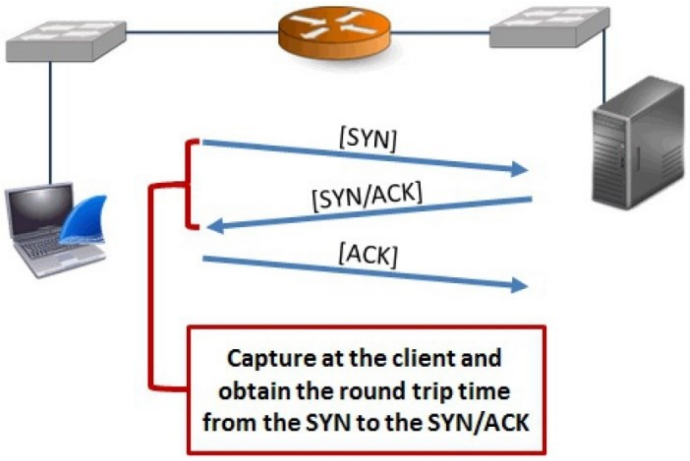
Packet# 1664	
Src Port : 80	Dest Port : 61598
Seq # : 1127	
Ack # : 324	
Flags : FIN/ACK	Window Size :241 [factored: 15424]

Packet# 1665	
Src Port : 61598	Dest Port : 80
Seq # : 324	
Ack # : 1128	
Flags : ACK	Window Size : 16163 [factored: 64652]

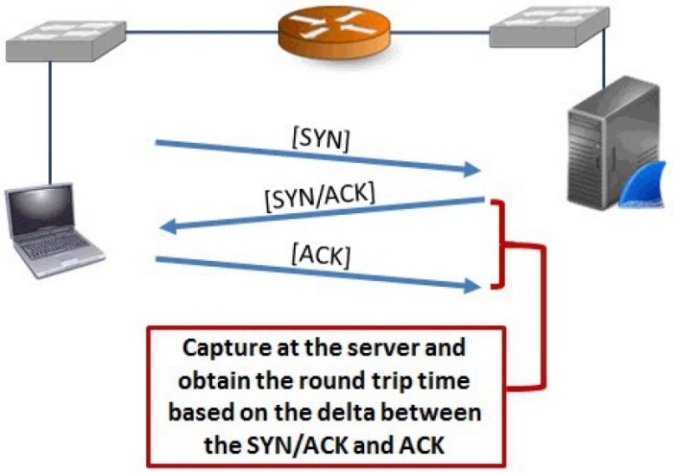
วิธีค้นหา

__ใช้ follow TCP Stream หรือใช้ Statistics | Conversation หรือใส่ display filter หา IP ผู้รับและผู้ส่ง แล้วตรวจสอบ 3 ถึง 4 packets สุดท้ายของ Stream__

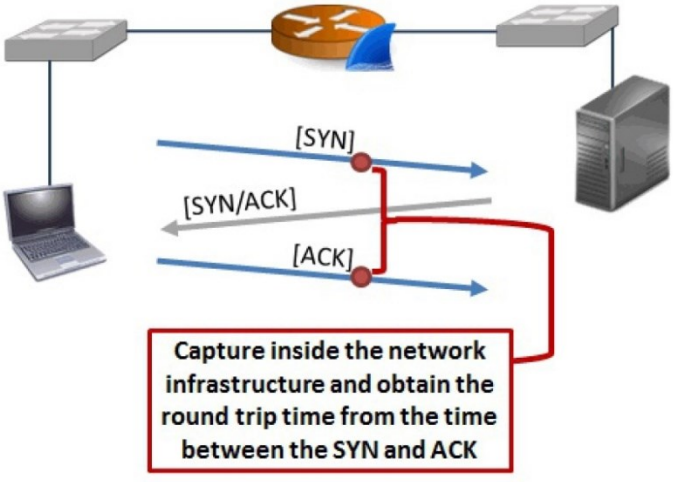
3. ใน Wireshark เราสามารถจะหา packet ที่มีคุณลักษณะของ flags เฉพาะได้ โดยใช้ display filter tcp.flags เช่น **tcp.flags.syn==1** หรือ **tcp.flags.ack==1** ซึ่งเราสามารถใช้หา RTT ของ TCP handshake ได้ โดยการหา RTT ของ TCP handshake มี 3 แบบ คือ วัดจากฝั่ง Client จะใช้เวลาระหว่าง SYN และ SYN-ACK



และวัดจากฝั่ง Server จะใช้เวลาระหว่าง SYN/ACK กับ ACK



แต่ในกรณีที่วัดจากอุปกรณ์ ควรใช้ระหว่าง SYN และ ACK ตามรูป



4. จากไฟล์ http-browse101d.pcapng ให้สร้าง display filter ที่สามารถแสดงเฉพาะ packet ต่อไปนี้ โดยไม่มี packet อื่นๆ มาปน (นักศึกษาพยายามคิดด้วยตนเอง)
    - 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)

    หา SYN และ SYN/ACK ใช้ tcp.flags.syn==1

    หา SYN/ACK และ ACK ใช้ (tcp.flags.syn==1 && tcp.flags.ack==1) || (tcp.seq==1 && tcp.ack==1) && (tcp.flags.fin==0) && (tcp.len==0)

    หา SYN และ ACK ใช้ (tcp.flags.syn==1 && tcp.flags.ack==0) || (tcp.seq==1 && tcp.ack==1) && (tcp.flags.fin==0) && tcp.len ==0

    สามารถตอบ display filter อื่นๆ ได้ที่ให้ผลตรงกัน

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

URL	เวลา
www.kmitl.ac.th	0.007318
www.reg.kmitl.ac.th	0.005622
www.google.com	0.021372

คำตอบแต่ละคนไม่เท่ากัน แต่จะเห็นว่าเว็บที่อยู่ใกล้จะมี RTT ที่น้อยกว่า

- ให้ตอบว่าระหว่าง RTT ที่วัดในครั้งนี้ กับ HTTP RTT ที่วัดในครั้งก่อนหน้านี้ บอกถึงอะไร และแตกต่างกันอย่างไร

    RTT ที่วัดครั้งนี้วัดจากการทำงานของ TCP/IP Stack ของระบบปฏิบัติการ ซึ่งไม่นับรวมเวลาในการให้บริการเว็บนั้นๆ ในขณะที่ HTTP RTT เป็นการทำงานของที่นับรวมเวลาในการให้บริการเว็บซึ่งให้บริการโดย Web Server Software เข้าไปด้วย