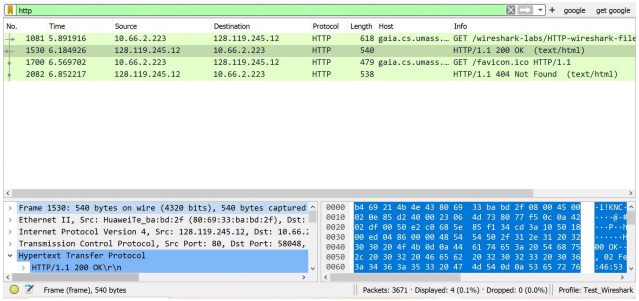
64010443 นิสาชล อยู่ดี

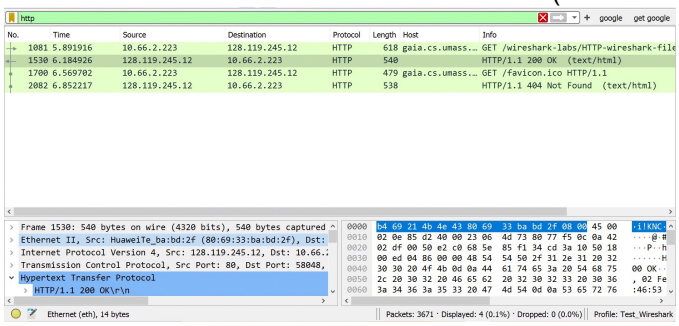
กิจกรรมที่ 4 : HTTP

ในกิจกรรมที่ผ่านมา จะเป็นการแนะนำการใช้งาน Wireshark เป็นส่วนใหญ่ในกิจกรรมครั้งนี้ จะเริ่มทำความรู้จักกับ protocol ใน Application Layer โดย protocol แรก คือ HTTP (Hypertext Transport Protocol)

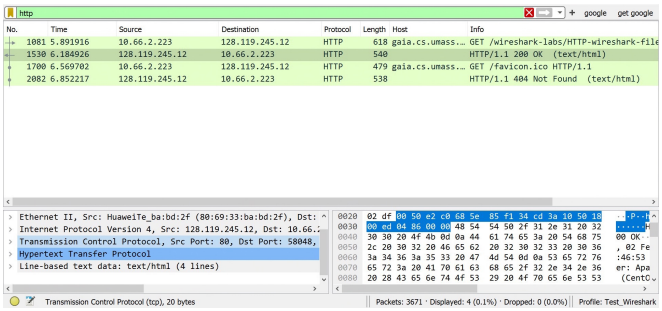
1. ให้ใช้ Wireshark เริ่มทำการ Capture และป้อน url : http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html เสร็จแล้วให้หยุด
2. ให้ใช้ display filter : http เพื่อให้แสดงเฉพาะ HTTP (ที่ถูกต้องควรจะมีแค่ 2 แพ็กเกต ในกรณีที่มีเกิน 2 แพ็คเกต อาจมาจากกรณี favicon ติดมาด้วย แต่ไม่ต้องไปสนใจแพ็คเกตที่เกินมา)
   (กรณีบรรทัดที่ 2 (Response) เป็น 304 Not Modified ให้เคลียร์ cache ของ browser แล้วทำใหม่)
3. ใน Packet List Pane ให้เลือก packet ที่เป็น HTTP Response และหาว่ามีความยาวของทั้ง frame เป็นเท่าไร 540 ให้บันทึก screenshot หน้าจอส่วนที่แสดงความยาวมาแสดง



4. ใน packet ตามข้อ 3 ความยาวเฉพาะส่วน header ของ Ethernet II เป็นเท่าไร 14 ให้บันทึก screenshot หน้าจอส่วนที่แสดงความยาวมาแสดง (Hint: หาข้อมูลจาก Packet Byte Pane)



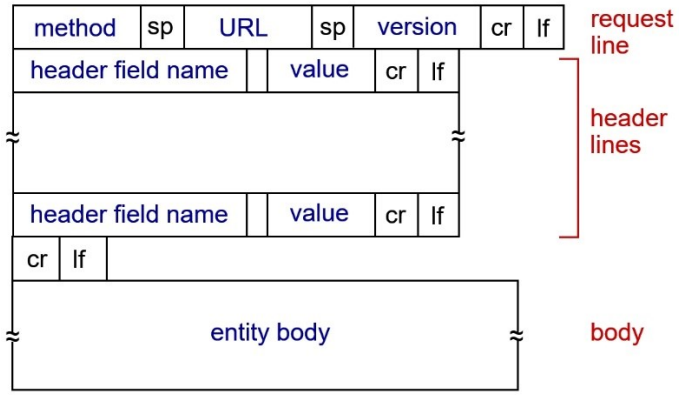
5. ใน packet ตามข้อ 3 ความยาวเฉพาะส่วน header ของ Transmission Control Protocol เป็นเท่าไร 20 ให้บันทึก screenshot หน้าจอส่วนที่แสดงความยาวมาแสดง



6. เพราะเหตุใด header ของ packet ต้องซ้อนเป็นชั้นๆ จงอธิบายเหตุผล

เพื่อจัดลำดับข้อมูลให้เป็นระเบียบเพื่อให้ง่ายต่อการอ่าน เวลาที่มีการเปลี่ยนแปลงข้อมูลของแต่ละ Layer เวลามีการเปลี่ยนแปลงข้อมูลของแต่ละ Layer หากไม่มี Header อาจทำให้เวลาที่ต้องการแก้ไขข้อมูล อาจทำให้ต้องใช้เวลานานในการไล่หาส่วนที่ต้องการแก้ไข

7. จากรูปแบบของ HTTP Message ตามรูป และ HTTP Request และ Response ที่ดักจับได้ ให้ตอบคำถามต่อไปนี้ (สามารถใช้วิธี capture แล้ว highlight ข้อมูลเพื่อตอบคำถามได้)



- browser และ server ใช้ HTTP version ใด HTTP / 1.1
- browser เป็นโปรแกรมอะไร Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom/109.0.0.0 Safari/537.36 Edg/109.0.1518.70\r\n
- server เป็นโปรแกรมอะไร Apache/2.4.6 OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
- ภาษาที่ browser ระบุว่าสามารถรับจาก server ได้ en-US, en;q=0.9 \r \n
- status code ที่ส่งกลับมาจาก server มายัง browser Status Code : 200 [Status Code Description : OK]
- ค่าของ Last-Modified ของไฟล์ที่ server Thu, 02 Feb 2023 06:46:01 GMT \r \n
- มีข้อมูลกี่ไบต์ที่ส่งมายัง browser 128 byte

- ให้สรุปว่า header field name ตาม HTTP message format ของข้อมูลที่ส่งกลับมีอะไรบ้าง

Version ของ browser server, โปรแกรมที่ browser และ server ใช้, ภาษาที่ browser รองรับ , Status code , วันที่เวลาที่แก้ไขล่าสุด , จำนวน byte ลักษณะการเชื่อมต่อ , ชนิดของข้อมูล

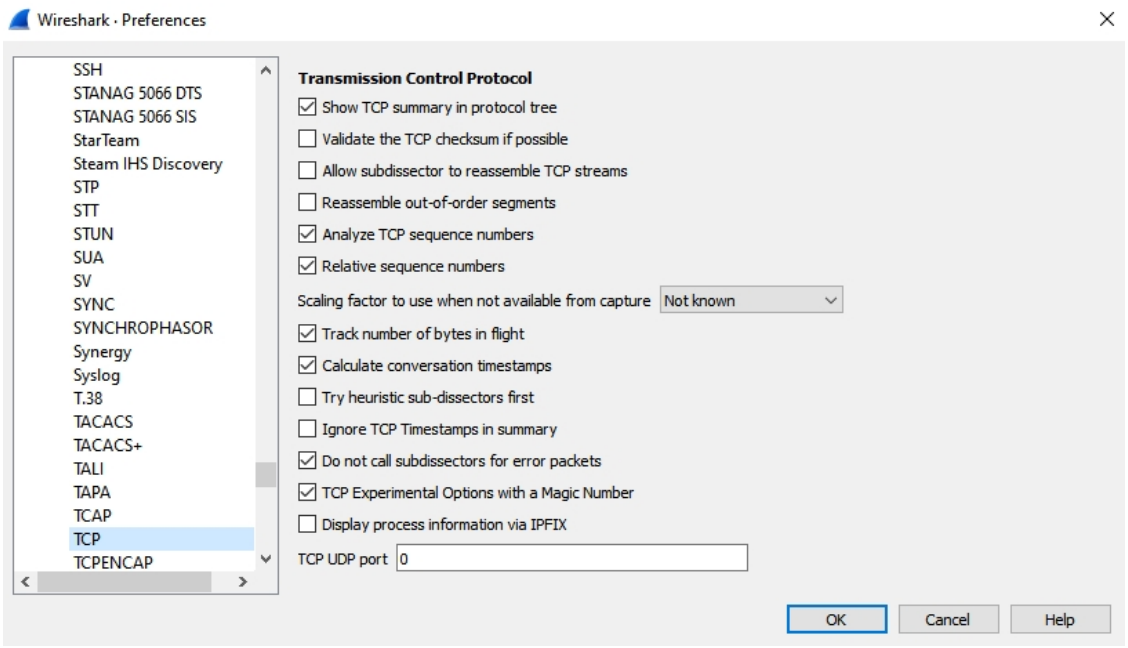
8. ให้นักศึกษาหาวิธี clear cache ของ browser ที่ตนเองใช้อยู่ แล้วจัดการ clear ให้เรียบร้อย

9. เปิด Wireshark ใหม่แล้ว capture การเรียกหน้าเว็บเพจไปยัง url http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html จากนั้นให้กด refresh เพื่อโหลดหน้าอีกครั้ง จากนั้นให้หยุด capture
10. ให้ใช้ display filter : http เพื่อให้แสดงเฉพาะ HTTP (ที่ถูกต้องควรจะมีแค่ 4 แพ็กเกต ในกรณีที่มีเกิน 4 แพ็คเกต อาจมาจากกรณี favicon ติดมาด้วย แต่ไม่ต้องไปสนใจแพ็คเกตที่เกินมา) และตอบคำถามต่อไปนี้
    - ใน HTTP GET ครั้งที่ 1 มีคำว่า IF-MODIFIED-SINCE หรือไม่ ไม่มี
    - ใน HTTP GET ครั้งที่ 2 มีคำว่า IF-MODIFIED-SINCE หรือไม่ มี
    - **(ถ้ามี) ข้อมูลที่ต่อจาก** IF-MODIFIED-SINCE มีความหมายอย่างไร

      หมายถึง มีการเปลี่ยนแปลง ณ เวลานี้หรือยัง

      เช่น ตามการทดลอง จะได้ If-Modified-Since: Thu, 02 Feb 2023 06:59:01 GMT\r\n
    - ในการตอบกลับของ server ครั้งที่ 2 มีการส่งไฟล์มาด้วยหรือไม่ สามารถอธิบายได้ว่าอย่างไร

      ไม่มี เนื่องจากว่ามี if-modified-since ใน get ครั้งที่ 2 ทำให้ไม่มีการส่งไฟล์มา
11. ให้ไปที่ Edit | Preference… | Protocol | TCP ตามรูป



ให้แน่ใจว่า ไม่ติ๊กที่ **Allow subdissector to reassemble TCP streams**

12. ให้ทำตามข้อ 8 อีกครั้ง และเปิด Wireshark ใหม่แล้ว capture การเรียกหน้าเว็บเพจไปยัง url http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html จากนั้นให้หยุด capture
13. ให้ใช้ display filter : http เพื่อให้แสดงเฉพาะ HTTP (ถ้าทำถูกจะมี 5 บรรทัด) ซึ่งจะเห็นว่าหลังจากข้อมูล HTTP/1.1 200 OK แล้ว ยังมีข้อมูลตามมาอีก เนื่องจากไฟล์ html มีความยาวมาก (มากกว่า 4000 ไบต์) ทำให้ไม่สามารถส่งมาใน 1 packet ได้ จึงมีการแบ่งเป็นหลายๆ ส่วน (โดย TCP) ดังนั้นใน Wireshark จึงแสดงคำว่า Continuation ให้นักศึกษาตอบคำถามต่อไปนี้

- มี HTTP GET กี่ครั้ง และมี packet ใดบ้างที่มี Status Code และเป็น Status Code ใด

  1 ครั้ง , Packet HTTP OK 200

14. ให้ทำตามข้อ 8 อีกครั้ง และเปิด Wireshark ใหม่แล้ว capture การเรียกหน้าเว็บเพจไปยัง url http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html จากนั้นให้หยุด capture
    - ให้ใช้ display filter : http เพื่อให้แสดงเฉพาะ HTTP และให้ตอบคำถามต่อไปนี้
    - มี HTTP GET กี่ครั้ง และไปยัง url ใดบ้าง

      Get 3 ครั้ง

      [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]

      [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]

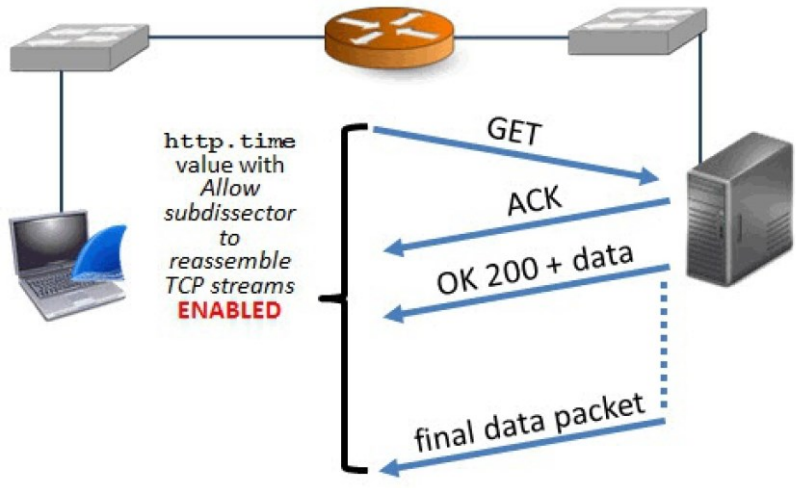
      [Full request URI: http://clients3.google.com/generate_204]

    - ผู้เรียนคิดว่า ภาพทั้ง 2 ภาพในไฟล์ ถูกทำการ download ทีละไฟล์ (serialize) หรือถูก download ไปพร้อมๆ กัน (parallelize) ให้อธิบาย

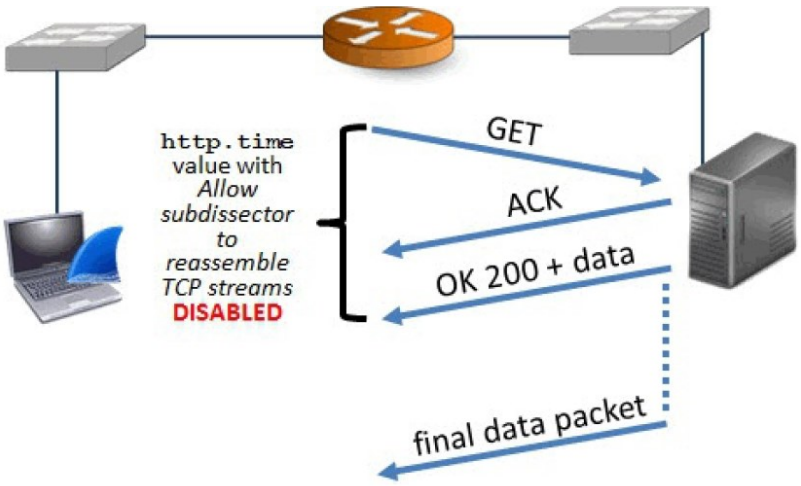
      Serialize เนื่องจากรูปทั้ง 2 Host ไม่เหมือนกัน เลยทำให้เกิดการ request ไป 2 รอบ

    - ให้คลิ๊กขวาที่ Transmission Control Protocol | Protocol Preferences แล้วติ๊กที่ **Allow subdissector to reassemble TCP streams** เกิดอะไรขึ้น

      packet ที่มีการ continue หายไป ไม่แสดงผล



ค่า http.time เมื่อ Enable Allow subdissector to reassemble TCP streams



ค่า http.time เมื่อ Disable Allow subdissector to reassemble TCP streams

ในการตรวจสอบความล่าช้าในการทำงานของ Web Server เราจะใช้ค่า RTT (Round Trip Time) ซึ่งเป็นค่าเวลาตั้งแต่ GET จนถึงตอบกลับ (OK 200) ซึ่งจะบอกได้ถึงการตอบสนองต่อการเรียกใช้ของ Web Server ตัวนั้น ซึ่งสำหรับ Wireshark จะมีผลกระทบจาก การกำหนดค่า **Allow subdissector to reassemble TCP streams** ตามรูป คือ หาก disable จะคิดเฉพาะ packet HTTP OK 200 แต่ถ้า Enable ก็จะเป็นเวลาที่นับรวมถึงการโหลดข้อมูลทั้งหมด ดังนั้นให้ disable **Allow subdissector to reassemble TCP streams** ก่อน มีผลในเรื่องของเวลา

15. ให้ไปที่ บรรทัดที่เป็น 200 OK แล้วไปที่ Hypertext Transfer Protocol แล้วขยาย subtrees ออกมาทั้งหมด แล้วไปที่บรรทัด **Time since request** แล้วเลือก **Apply as Column** ให้ตั้งชื่อว่า HTTP Delta จากนั้นให้ sort เพื่อหา packet ที่มีเวลา HTTP Delta มากที่สุด
16. ให้นักศึกษาตรวจสอบ RTT ของ 3 เว็บดังนี้ 1) http://example.com/ 2) http://www.http2demo.io/ 3) http://www.vulnweb.com/ และเว็บอื่นอีก 1 เว็บ (ผู้เรียนเลือกเอง) ให้บอกว่าค่า RTT ของแต่ละเว็บมีค่าใด ให้เรียงลำดับน้อยไปมาก ให้นักศึกษาแสดงขั้นตอนการทำงาน (เขียนอธิบายย่อๆ และบันทึก screenshot ประกอบ) และเปรียบเทียบค่ากับเพื่อนอีก 1 คน ว่าลำดับเหมือนกันหรือไม่ อย่างไร

ทำการ capture url ที่ต้องการ ใส่ display filter : http ทำการ sort ที่ คอลัมน์ http delta ดูว่าค่ามากสุด คือค่าใด

RTT ของเรา =>

1. http://datastruc.ce.kmitl.ac.th : 0.046270000
2. http://www.http2demo.io/ : 0.065791000
3. http://example.com/ : 0.206425000
4. http://www.vulnweb.com/ : 0.398066000

RTT ของเพื่อน

1. http://datastruc.ce.kmitl.ac.th = 0.121040000
2. http://example.com/ = 0.213822000
3. http://www.http2demo.io/ = 0.233378000
4. http://www.vulnweb.com/ = 0.234044000

15

http

google get google

No.	Time	Source	Destination	Protocol	Length	Host	HTTP delta	Info
777	13.521530	128.119.245.12	10.66.2.223	HTTP	1355		0.315569000	HTTP/1.1 200 OK (text/
875	14.138648	178.79.137.164	10.66.2.223	HTTP	225		0.304440000	HTTP/1.1 301 Moved Perm
2277	27.137344	128.119.245.12	10.66.2.223	HTTP	294		0.284255000	HTTP/1.1 304 Not Modifi
823	13.848468	128.119.245.12	10.66.2.223	HTTP	1434		0.278879000	HTTP/1.1 200 OK (PNG)[
2213	26.853089	10.66.2.223	128.119.245.12	HTTP	645	gaia.cs.umass.…		GET /wireshark-labs/HTT

งานครั้งที่ 4

- การส่งงาน เขียนหรือพิมพ์ลงในเอกสารนี้ และส่งเป็นไฟล์ PDF เท่านั้น
- ตั้งชื่อไฟล์โดยใช้รหัสนักศึกษา ตามด้วย section และ _lab04 ตามตัวอย่างต่อไปนี้
  64019999_sec20_lab04.pdf
- กำหนดส่ง ภายในวันที่ 10 กุมภาพันธ์ 2566 โดยให้ส่งใน Microsoft Teams ของรายวิชา