

## 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 Protocol HTTP (ถ้าทำถูกจะมีแค่ 2 บรรทัด แต่อาจมี favicon ติดมาไม่ต้องไปสนใจ)  
(กรณีบรรทัดที่ 2 (Response) เป็น 304 Not Modified ให้ปิด Browser แล้วทำใหม่)
3. ใน Packet HTTP **Response** มีความยาวเฟรมทั้งหมดเท่าไร ประมาณ 5xx ไบต์ ให้ Capture หน้าจอส่วนที่แสดงความยาวประกอบ  
ประกอบด้วย TCP payload = 464 ไบต์ + Header Ethernet 14 ไบต์ + Header IP 20 ไบต์ + Header TCP 20 ไบต์  
  - > Transmission Control Protocol, Src Port: 80, Dst Port: 7437, Seq: 1, Ack: 388, Len: 464
  - > Hypertext Transfer Protocol
4. ใน Packet ข้อ 3 ความยาวของ Header Ethernet II เป็นเท่าไร 14 ไบต์ ให้ Capture หน้าจอส่วนที่แสดงความยาวประกอบ (Hint : ใช้ Packet Byte Pane)

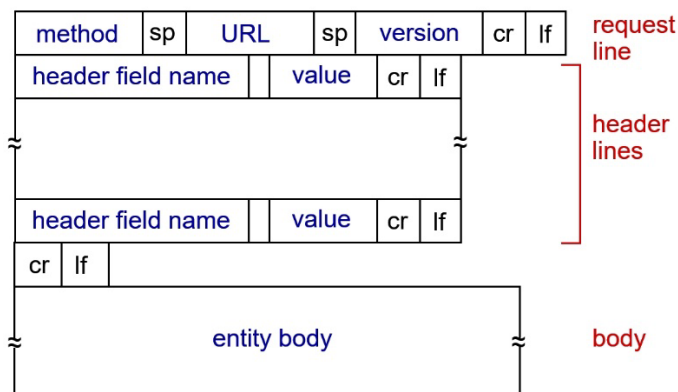
```
> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured
▼ Ethernet II, Src: Dell_02:eb:60 (18:66:da:02:eb:60), Ds
  > Destination: HuaweiTe_fb:24:d5 (c4:b8:b4:fb:24:d5)
  > Source: Dell_02:eb:60 (18:66:da:02:eb:60)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.4, Dst: 128
> Transmission Control Protocol, Src Port: 7358, Dst Port:
```

```
, 0000 c4 b8 b4 fb 24 d5 18 66 da 02 eb 60 08 00 45 00
0010 00 34 bd 47 40 00 80 06 00 00 c0 a8 01 04 80 77
0020 f5 0c 1c be 00 50 e4 9d a4 40 00 00 00 00 80 02
0030 fa f0 37 57 00 00 02 04 05 b4 01 03 03 08 01 01
0040 04 02
```

5. ใน Packet ข้อ 3 ความยาวของ TCP Header เป็นเท่าไร 20 ไบต์ ให้ Capture หน้าจอส่วนที่แสดงความยาวประกอบ

```
> Transmission Control Protocol, Src Port: 80, Dst Port: 7437, Seq: 1, Ack: 3
Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Sun, 14 Feb 2021 03:59:10 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.14 mod_perl/2.6
    Last-Modified: Sat, 13 Feb 2021 06:59:01 GMT\r\n
0020  01 04 00 50 1d 0d 5c 84 2d 06 5b 3a 67 e5 50 18  ..P.. \r\n
0030  02 2e f2 e2 00 00 48 54 54 50 2f 31 2e 31 20 32  .....HT TP/1.1 2
0040  30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 53 75 6e  00 OK\r\n Date: Sun
0050  2c 20 31 34 20 46 65 62 20 32 30 32 31 20 30 33  , 14 Feb 2021 03
```

6. เหตุผลที่ Header ของข้อมูลต้องซ้อนเป็นชั้นๆ คือ  
 เป็นการ Encapsulation เนื่องจาก packet มีการส่งลงมาทีละ Layer ในแต่ละ Layer ก็จำเป็นต้องมีข้อมูลสำหรับใช้ใน Layer เดียวกันที่ปลายทาง จึงมีการซ้อนเป็น Layer โดยเมื่อถึงปลายทาง ก็จะมีการถอด Header ออกไป คล้ายกับเปิดซองจดหมาย
7. จากรูปแบบของ HTTP Message ตามรูป และ HTTP Request และ Response ที่ดักจับได้ ให้ตอบคำถามต่อไปนี้ (สามารถใช้วิธี Capture แล้ว Highlight ข้อมูลเพื่อตอบคำถามได้)

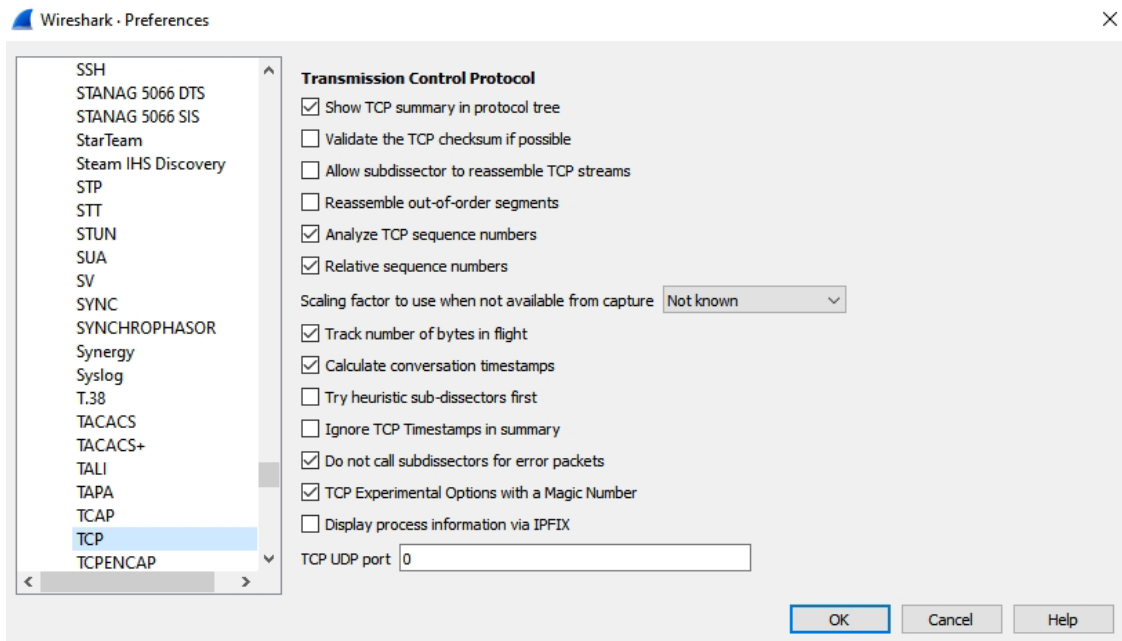


- Browser และ Server ใช้ HTTP version ไต **HTTP 1.1**
- Browser เป็นโปรแกรมอะไร **โปรแกรมอะไรก็ได้ที่เป็นชื่อ Browser**
- Server เป็นโปรแกรมอะไร **Apache 2.4.6**
- ภาษาที่ Browser ระบุว่าสามารถรับจาก Server ได้ **ไม่แน่นอนขึ้นกับ Browser**
- Status Code ที่ส่งกลับมาจาก Server มายัง Browser **200 OK**
- ค่าของ Last-Modified ของไฟล์ที่ Server **Tue 02 Feb 2021** ขึ้นกับวันที่ดึงข้อมูล
- มีข้อมูลกี่ไบต์ที่ส่งมายัง Browser **128 ไบต์ (Content Length)**
- ให้สรุปว่า header field name ตาม HTTP message format ของข้อมูลที่ส่งกลับมีอะไรบ้าง

Date, Server, Last-Modified, ETAG, Accept-Ranges, Content-Length, Keep-Alive, Connection,  
Content-Type

```
Date: Tue, 09 Feb 2021 12:36:14 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.14 mod_perl/2.0.11 Perl/v5.16.3\r\n
Last-Modified: Tue, 09 Feb 2021 06:59:01 GMT\r\n
ETag: "80-5bae1d2479c57"\r\n
Accept-Ranges: bytes\r\n
> Content-Length: 128\r\n
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=UTF-8\r\n
```

8. ให้นักศึกษาหาวิธี clear cache ของ Browser ที่ตนเองใช้อยู่ แล้วจัดการ clear ให้เรียบร้อย
9. เปิด Wireshark ใหม่แล้ว Capture ที่ url <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html> จากนั้นให้กด Refresh เพื่อโหลดหน้าอีกครั้ง จากนั้นให้หยุด Capture
10. ให้ใช้ display filter : http เพื่อให้เห็นเฉพาะ Protocol HTTP (ถ้าทำถูกจะมีแค่ 4 บรรทัด บรรทัด แต่อาจมี favicon ติดมาไม่ต้องไปสนใจ) 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 **Browser จะมีการ Cache ข้อมูลเอาไว้ ดังนั้นเมื่อ Browser ทราบว่าเป็นการโหลดหน้าเดิม ก็จะส่งวันและเวลาที่มีการโหลดหน้าเดิมไปให้กับ Server เพื่อให้ Server ตัดสินใจว่าจะส่ง Content มาให้ใหม่หรือไม่**
  - ในการตอบกลับของ Server ครั้งที่ 2 มีการส่งไฟล์มาด้วยหรือไม่ จะอธิบายอย่างไร **ไม่มีการส่งไฟล์ เนื่องจากนับจากเวลาใน IF-MODIFIED-SINCE ยังไม่มีการเปลี่ยนแปลง Content ใหม่ ยังเป็น Content เดิม เมื่อ Server ตรวจสอบพบจึงรู้ว่าข้อมูลเป็นข้อมูลเดิม จึงไม่มีการส่งมาใหม่**
11. ให้ไปที่ Edit | Preference... | Protocol | TCP ตามรูป



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

12. ให้ทำตามข้อ 8 อีกครั้ง และเปิด Wireshark ใหม่แล้ว Capture ที่ url <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html> จากนั้นให้หยุด Capture
13. ให้ใช้ display filter : http เพื่อให้แสดงเฉพาะ Protocol HTTP (ถ้าทำถูกจะมี 5 บรรทัด) ซึ่งจะเห็นว่าหลังจากข้อมูล HTTP/1.1 200 OK แล้ว ยังมีข้อมูลตามมามาก 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

มี HTTP GET 1 ครั้ง และ packet ที่ 2 ที่มี status code คือ Code 200 ok
14. ให้ทำตามข้อ 5 อีกครั้ง และเปิด Wireshark ใหม่แล้ว Capture ที่ url <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html> จากนั้นให้หยุด Capture
  - ให้ใช้ display filter : http เพื่อให้แสดงเฉพาะ Protocol HTTP และให้ตอบคำถามต่อไปนี้
  - มี HTTP GET กี่ครั้ง จาก url ใดบ้าง

3 ครั้งจาก

<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>.

<http://gaia.cs.umass.edu/pearson.png>,

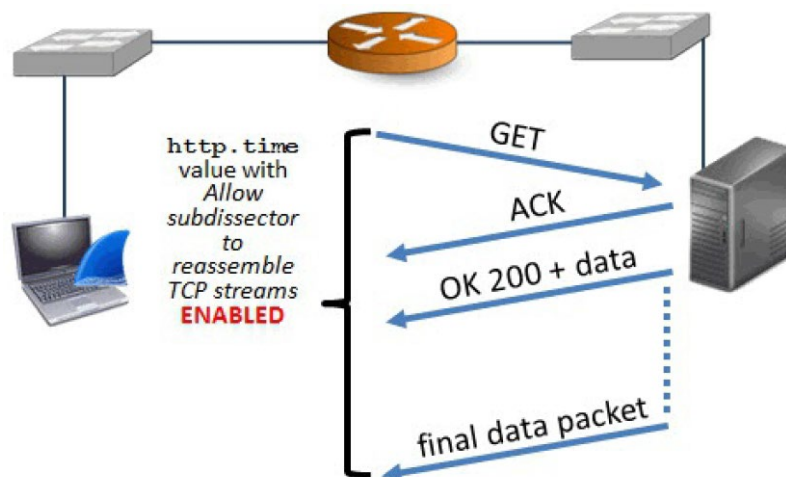
[http://kurose.cslash.net/8E\\_cover\\_small.jpg](http://kurose.cslash.net/8E_cover_small.jpg)

  - นักศึกษาคิดว่า ภาพทั้ง 2 ภาพในไฟล์ มีการ download ที่ละไฟล์ (serial) หรือทำพร้อมๆ กัน (parallel) ให้อธิบาย

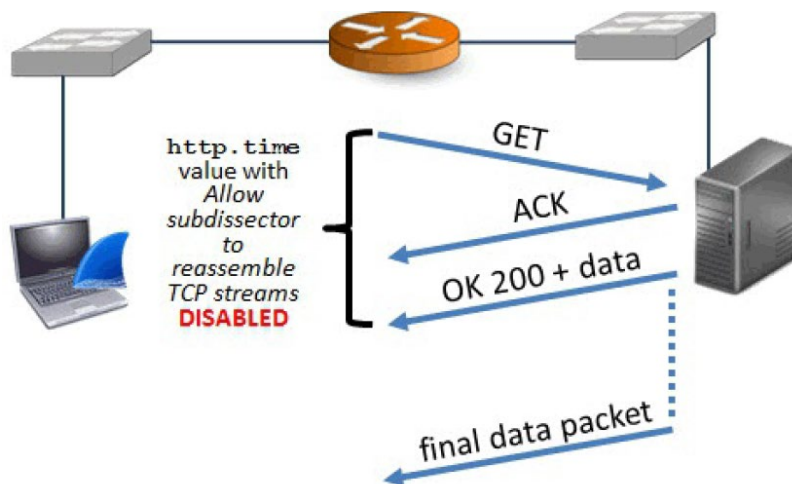
parallel ซึ่งถ้าดูจาก Wireshark จะเห็นว่ามี Request ในเวลาใกล้ๆ กัน และมี Response กลับมาไม่เป็นไปตามลำดับ (อาจต้องโหลดดูหลายๆ ครั้งจึงจะเห็นว่าบางครั้งภาพที่ 2 มาก่อนภาพแรก)

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

Wireshark จะไม่แสดงคำว่า Continue และจะรวมข้อมูลของ Response ใน Stream เดียวกันเป็น packet เดียว



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** ตาม

รูป คือ หาก Disable จะคิดเฉพาะ packet HTTP OK 200 แต่ถ้า Enable ก็จะเป็นเวลาที่นับรวมถึงการโหลดข้อมูลทั้งหมด ดังนั้นให้ disable **Allow subdissector to reassemble TCP streams** ก่อน

15. ให้ไปที่ บรรทัดที่เป็น 200 OK แล้วไปที่ Hypertext Transfer Protocol แล้ว Expand Subtrees ออกมาทั้งหมด แล้วไปที่บรรทัด **Time since request** แล้วเลือก **Apply as Column** ให้ตั้งชื่อว่า HTTP Delta จากนั้นให้ Sort จะพบ packet ที่ใช้เวลามากที่สุด
16. ให้นักศึกษาตรวจสอบ RTT ของเว็บ [www.ce.kmitl.ac.th](http://www.ce.kmitl.ac.th), [www.reg.kmitl.ac.th](http://www.reg.kmitl.ac.th), [www.kmitl.ac.th](http://www.kmitl.ac.th) และเว็บอื่นอีก 1 เว็บ (นักศึกษาเลือกเอง) ให้ออกว่าค่า RTT ของแต่ละเว็บมีค่าใด ให้เรียงลำดับน้อยไปมาก ให้นักศึกษาแสดงขั้นตอนการทำงาน (เขียนอธิบายย่อๆ และ Capture รูปประกอบ) และเปรียบเทียบกับเพื่อนอีก 1 คน

ข้อมูลของแต่ละคนจะไม่เท่ากันขึ้นกับปัจจัยหลายอย่าง เช่น ความช้าเร็ว ใกล้เคียง