CSE351 Computer Networks

Fall, 2025, Homework #2
Due: November 25th, 2025 (Tue), 23:59 KST

Instructor: Taesik Gong (taesik.gong@unist.ac.kr), TA: Yeji Park (yejipark@unist.ac.kr)

==

Submission Instructions. Submit this homework as a single **PDF file** named *HW2_StudentNumber_Name.pdf* (e.g., *HW2_20251234_HongGildong.pdf*) on Blackboard. All screenshots must be from the **original Wireshark interface** and clearly readable. If the file cannot be opened or is unreadable, no points will be given. **Submit only screenshots; do not attach .pcap files.**

Collaboration Policy. You are welcome to discuss the homework with your classmates to understand the concepts, but all answers, explanations, screenshots, and write-ups must be your own. Copying text or figures from others' work, or sharing your own write-up directly, is considered plagiarism and will result in no points for all involved.

Network Guidelines. Capture only **your own traffic** using Wireshark. Do not intercept others' packets, flood, or overload any host. Use normal applications (e.g., Gmail, YouTube, Naver) and generate only light, typical traffic.

==

1. Homework Overview

The goal of this homework is to observe how different Internet applications generate and exchange packets, and to connect these real traces with the concepts learned in class. Using **Wireshark**, you will capture and analyze network traffic to understand how the application, transport, and network layers interact in real environments. The purpose of this assignment is not only to test correctness but also to help you gain **hands-on experience** in real-world network analysis.

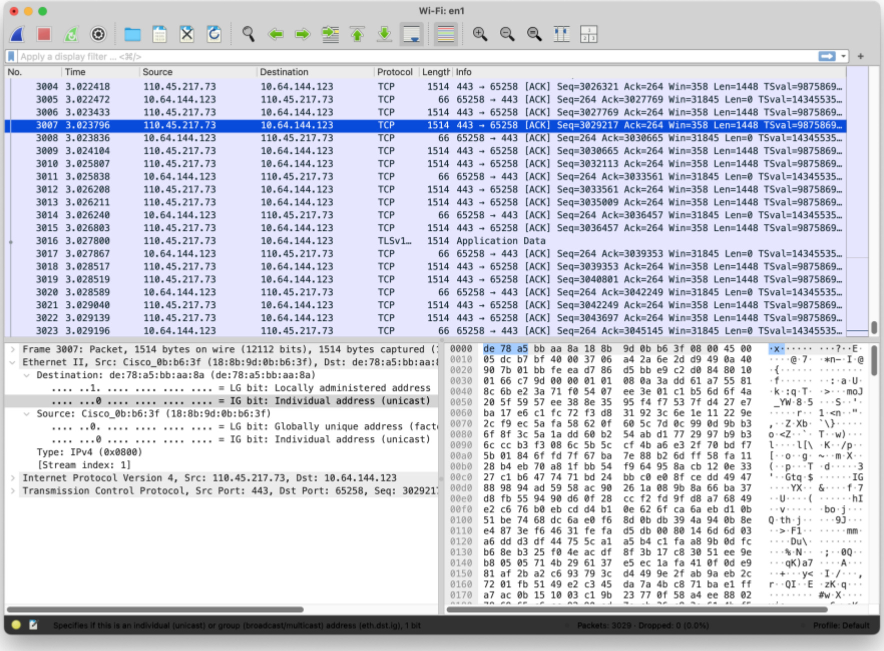
2. Wireshark

Wireshark is a free and open-source network protocol analyzer used to capture and inspect packets exchanged across a network in real time. It allows you to visualize traffic at every layer of the protocol stack, from the physical and link layers up to the application layer, and to understand how real data flows between clients and servers. By observing these packets, you can see how protocols such as TCP, UDP, IP, HTTP, and TLS operate in real environments.

In this homework, Wireshark will be used to record and analyze the actual network traffic generated by various Internet applications. Through this process, you will learn to recognize protocol behaviors such as TCP handshakes, retransmissions, or persistent HTTP connections, and to relate them to theoretical concepts learned in class.

To use Wireshark, first install it from the official website (https://www.wireshark.org/download.html) and launch the program. Choose the network interface currently in use, such as Wi-Fi or Ethernet, and click the Start Capture button to begin recording packets. Perform the desired network activity, for example, browsing a webpage or sending a message and then click Stop to end the capture. You can apply display filters (e.g., http, tcp.port == 443, or ip.addr == 8.8.8.8) to focus on relevant traffic. Saved capture files can be exported in .pcapng format for later analysis.

Using Wireshark will help you connect theory what you've learn in the course with real Internet behavior while also developing valuable practical skills for debugging, security analysis, and performance evaluation.



```
No.       Time      Source          Destination     Protocol Length Info
     3004 3.022418  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3026321 Ack=264 Win=358 Len=1448 TSval=9875869…
     3005 3.022472  10.64.144.123   110.45.217.73   TCP        66  65258 → 443 [ACK] Seq=264 Ack=3027769 Win=31845 Len=0 TSval=14345535…
     3006 3.023433  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3027769 Ack=264 Win=358 Len=1448 TSval=9875869…
     3007 3.023796  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3029217 Ack=264 Win=358 Len=1448 TSval=9875869…
     3008 3.023836  10.64.144.123   110.45.217.73   TCP        66  65258 → 443 [ACK] Seq=264 Ack=3030665 Win=31845 Len=0 TSval=14345535…
     3009 3.024104  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3030665 Ack=264 Win=358 Len=1448 TSval=9875869…
     3010 3.025807  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3032113 Ack=264 Win=358 Len=1448 TSval=9875869…
     3011 3.025838  10.64.144.123   110.45.217.73   TCP        66  65258 → 443 [ACK] Seq=264 Ack=3033561 Win=31845 Len=0 TSval=14345535…
     3012 3.026208  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3033561 Ack=264 Win=358 Len=1448 TSval=9875869…
     3013 3.026211  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3035009 Ack=264 Win=358 Len=1448 TSval=9875869…
     3014 3.026240  10.64.144.123   110.45.217.73   TCP        66  65258 → 443 [ACK] Seq=264 Ack=3036457 Win=31845 Len=0 TSval=14345535…
     3015 3.026803  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3036457 Ack=264 Win=358 Len=1448 TSval=9875869…
     3016 3.027800  110.45.217.73   10.64.144.123   TLSv1…   1514  Application Data
     3017 3.027867  10.64.144.123   110.45.217.73   TCP        66  65258 → 443 [ACK] Seq=264 Ack=3039353 Win=31845 Len=0 TSval=14345535…
     3018 3.028517  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3039353 Ack=264 Win=358 Len=1448 TSval=9875869…
     3019 3.028519  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3040801 Ack=264 Win=358 Len=1448 TSval=9875869…
     3020 3.028589  10.64.144.123   110.45.217.73   TCP        66  65258 → 443 [ACK] Seq=264 Ack=3042249 Win=31845 Len=0 TSval=14345535…
     3021 3.029040  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3042249 Ack=264 Win=358 Len=1448 TSval=9875869…
     3022 3.029139  110.45.217.73   10.64.144.123   TCP      1514  443 → 65258 [ACK] Seq=3043697 Ack=264 Win=358 Len=1448 TSval=9875869…
     3023 3.029196  10.64.144.123   110.45.217.73   TCP        66  65258 → 443 [ACK] Seq=264 Ack=3045145 Win=31845 Len=0 TSval=14345535…
```

```
> Frame 3007: Packet, 1514 bytes on wire (12112 bits), 1514 bytes captured (
v Ethernet II, Src: Cisco_0b:b6:3f (18:8b:9d:0b:b6:3f), Dst: de:78:a5:bb:aa:
  v Destination: de:78:a5:bb:aa:8a (de:78:a5:bb:aa:8a)
      .... ..1. .... .... .... .... = LG bit: Locally administered address
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  v Source: Cisco_0b:b6:3f (18:8b:9d:0b:b6:3f)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (fact
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
    [Stream index: 1]
> Internet Protocol Version 4, Src: 110.45.217.73, Dst: 10.64.144.123
> Transmission Control Protocol, Src Port: 443, Dst Port: 65258, Seq: 302921
```

```
0000  de 78 a5 bb aa 8a 18 8b  9d 0b b6 3f 08 00 45 00   ·x······ ···?··E·
0010  05 dc b7 bf 40 00 37 06  a4 2a 6e 2d d9 49 0a 40   ····@·7· ·*n-·I·@
0020  90 7b 01 bb fe ea d7 86  d5 bb e9 c2 d0 84 80 10   ·{······ ········
0030  01 66 c7 9d 00 00 01 01  08 0a 3a dd 61 a7 55 81   ·f······ ··:·a·U·
0040  8c 6b e2 3a 71 f0 54 07  ee 3e 01 c1 b5 6d 6f 4a   ·k·:q·T· ·>···moJ
0050  20 5f 59 57 ee 38 8e 35  95 f4 f7 53 7f d4 27 e7    _YW·8·5 ···S··'·
0060  ba 17 e6 c1 fc 72 f3 d8  31 92 3c 6e 1e 11 22 9e   ·····r·· 1·<n··"·
0070  2c f9 ec 5a fa 58 62 0f  60 5c 7d 0c 99 0d 9b b3   ,··Z·Xb· `\}·····
0080  6f 8f 3c 5a 1a dd 60 b2  54 ab d1 77 29 97 b9 b3   o·<Z··`· T··w)···
0090  6c cc b3 f3 08 6c 5b 5c  cf 4b a6 e3 2f 70 bd f7   l····l[\ ·K··/p··
00a0  5b 01 84 6f fd 7f 67 ba  7e 88 b2 6d ff 58 fa 11   [··o··g· ~··m·X··
00b0  28 b4 eb 70 a8 1f bb 54  f9 64 95 8a cb 12 0e 33   (··p···T ·d·····3
00c0  27 c1 b6 47 74 71 bd 24  bb c0 e0 8f ce dd 49 47   '··Gtq·$ ······IG
00d0  88 98 94 ad 59 58 ac 90  26 1a 08 9b 8a 66 ba 37   ····YX·· &····f·7
00e0  d8 fb 55 94 90 d6 0f 28  cc f2 fd 9f d8 a7 68 49   ··U····( ······hI
00f0  e2 c6 76 b0 eb cd d4 b1  0e 62 6f ca 6a eb d1 0b   ··v····· ·bo·j···
0100  51 be 74 68 dc 6a e0 f6  8d 0b db 39 4a 94 0b 8e   Q·th·j·· ···9J···
0110  e4 87 3e f6 46 31 fe fa  d5 db 00 80 14 6d 6d 03   ··>·F1·· ·····mm·
0120  a6 dd d3 df 44 75 5c a1  a5 b4 c1 fa a8 9b 0d fc   ····Du\· ········
0130  b6 8e b3 25 f0 4e ac df  8f 3b 17 c8 30 51 ee 9e   ···%·N·· ·;··0Q··
0140  b8 05 05 71 4b 29 61 37  e5 ec 1a fa 41 0f 0d e9   ···qK)a7 ····A···
0150  81 af 2b a2 c6 93 79 3c  d4 49 9e 2f ab 9a eb 2c   ··+···y< ·I·/···,
0160  72 01 fb 51 49 e2 c3 45  da 7a 4b c8 71 ba e1 ff   r··QI··E ·zK·q···
0170  a7 ac 0b 15 10 03 c1 9b  23 77 0f 58 a4 ee 88 02   ········ #w·X····
```

Specifies if this is an individual (unicast) or group (broadcast/multicast) address (eth.dst.ig), 1 bit
Packets: 3029 · Dropped: 0 (0.0%)
Profile: Default

Fig. 1 Wireshark running example

You may refer to the official Wireshark User's Guide and Wireshark Wiki for additional help while completing this homework. The User's Guide provides installation, setup, and filtering instructions in detail, while the Wiki includes sample captures.

Helpful resources:

1. Wireshark User's Guide: https://www.wireshark.org/docs/wsug_html_chunked/
2. Wireshark Wiki (Sample Captures): https://wiki.wireshark.org/SampleCaptures
3. Wireshark Tutorial for Beginners (Video): https://www.youtube.com/watch?v=qTaOZrDnMzQ
4. Packet Analysis Guide: https://medium.com/@cyberengage.org/master-wireshark-tool-like-a-pro-the-ultimate-packet-analysis-guide-for-real-world-analysts-981fb9024e7d

3. Application Traffic Experiment

In this Homework, you will perform experiments on **five different application categories**, capturing traffic from one example in each category. The categories are as follows:

1. **E-mail**: Use a client or web service such as *Gmail, Outlook, or Thunderbird* to send and receive e-mails while capturing the related packets.
2. **Web Browsing**: Use a browser such as *Chrome, Edge, or Safari* to visit any website of your choice.
3. **Instant Messaging**: Use a messaging application such as *KakaoTalk, Discord, Slack, WhatsApp, or Telegram*.
4. **Video Streaming**: Play a short video (at least 30 seconds) using services such as *YouTube, Netflix, Twitch, or Naver TV.*
5. **Online or Web Game**: Choose any online or browser-based game that you enjoy and play for a short time.

For each category, generate typical user actions (such as sending a message, loading a webpage, or streaming a short video) and capture the corresponding packets. Then analyze the data to identify protocol behavior, performance characteristics, and key observations such as congestion control, encryption, or caching.

1. **Traffic Capturing Experiment**: For each application category, plan how to generate representative traffic, determine appropriate capture conditions, and design an analysis strategy to extract meaningful patterns. The experiment should be reproducible and systematic, connecting real packet behavior with theoretical network concepts learned in class.

2. **Analyze the result**: For every application, describe how the traffic was generated, include representative screenshots showing packet exchanges or flows, and provide detailed analysis of the observed behavior. Connect your findings to the concepts learned in lectures, *including topics such as TCP handshakes, retransmissions, congestion control, error recovery, or persistent connections.* Discuss what kinds of data are exchanged, how the protocols interact across layers, and how the design of each system balances reliability, latency, and throughput**.**

3. **Discussion**: Summarize and compare your observations. Highlight similarities and differences in how each category structures its communication, manages timing and reliability, and utilizes transport-layer mechanisms. *Discuss what these findings reveal about protocol design and performance trade-offs in modern Internet services*. Through this experiment, you will gain a deeper appreciation of how real-world network traffic reflects the principles of Internet architecture and how theoretical models manifest in practical communication systems.

4. Report Guide

***** First, all required report elements are listed in the uploaded report template. Read it carefully and write your report following the instructions. *****

***** Also, please read carefully the guide below. *****

Please upload your **report in PDF format only** and do not include the raw .pcapng file in your submission. The main body of the report **should not exceed five pages**. Additional screenshots, tables, or supporting materials may be included in the Appendix (no page limit). **However, all required elements and key analysis results must appear in the main body, not only in the appendix.** If essential information is missing from the main content, points may be deducted. Use figures, screenshots, and tables effectively to support your explanations. Whenever you include supplementary material in the appendix, clearly reference it in the text (e.g., "see Appendix A"). **Any materials not referenced in the main text will be considered unnecessary and may not be reviewed by the TA.**

Each section of your report must clearly include all required components. Screenshots and analysis results should be well-organized and easy to read. Concise, clear explanations are preferred over lengthy paragraphs. For detailed formatting requirements, refer to the uploaded report template.

5. Evaluation Criteria (Total 100 points)

(1) Completeness of five application analyses (60 points)

(2) Connection with class concepts (30 points)

(3) Summary and insights (10 points)