CSC358H5: Principles of Computer Networking — Winter 2025

Wireshark Lab 0: Introduction to Wireshark

Due Date: No submission is required

Welcome to CSC358H5. Apart from lectures, problem sets, and Programming Assignments, we'll be delving into a series of Wireshark labs. The primary aim of these tasks is to provide students with an opportunity to see protocols in action, which greatly deepens their understanding of network protocols. Throughout the course, there will be three Wireshark Labs, in which you sniff into packets being transmitted in real network environment. It's noteworthy that the Wireshark Labs are modeled after similar labs in "Computer Networking: A Top-Down Approach," by Jim Kurose and Keith Ross.

The main purpose of this lab, *i.e.* WL0, is to get you started with Wireshark and making some simple packet capture and observations.

WL0 is will not be graded and it contributes 0% towards your course grade.

Collaboration

All Wireshark Labs should be done individually. You should not share your solutions with others or see other student solutions. You can also ask your questions on the course discussion board publicly, as long as you do not reveal the solution. If there is a specific question related to a part of lab that needs revealing the answer, either ask it in a tutorial session or an office hour from a TA or via a private question on the discussion website.

0 Lab Environment

Before starting this lab, please download and install Wireshark if you had not done so already. You can find the link to download Wireshark (directly to your computer) `https://www.wireshark.org/download.html`. You do **not** need to run Wireshark on a VM (such as in Kali Linux) in this course, though you are welcome to do so.

1 What is Wireshark?

Wireshark is a free, open-source packet analysis tool used to record and analyze network traffic. The tool provides a convenient interface for visualizing network traffic and viewing information about traffic flows and individual packets.

Wireshark is mainly used for: i) Capturing network traffic in transit; ii) Viewing and analyzing previously-captured network traffic via capture files (including .cap and .pcap files). This functionality can be useful for a variety of tasks, such as detecting problems with a network, debugging programs, examining security issues, verifying network applications, and education.

Note, however, that Wireshark cannot be directly used for intrusion detection purposes. Wireshark has no system of alerts that would notify users of suspicious network activity, though data provided by Wireshark can be used to find such activity. Wireshark likewise cannot be used to update network settings or send packets over a network.

Wireshark is capable of operating in three main modes, each serving distinct purposes:

- **Normal Mode:** Captures packet to and from your machine.
- **Promiscuous Mode:** Captures all non-encrypted packets within a given network and requires hardware support.
- **Monitor Mode:** Captures all encrypted and unencrypted packets within a given network and requires hardware support. Note that this mode is only available for Unix/Linux systems.

1.1 Wireshark Interface

Figure 1 illustrates an example of a Wireshark main window. This window contains three main panes, namely Packet List Pane, Packet Details Pane, and Packet Bytes Pane.

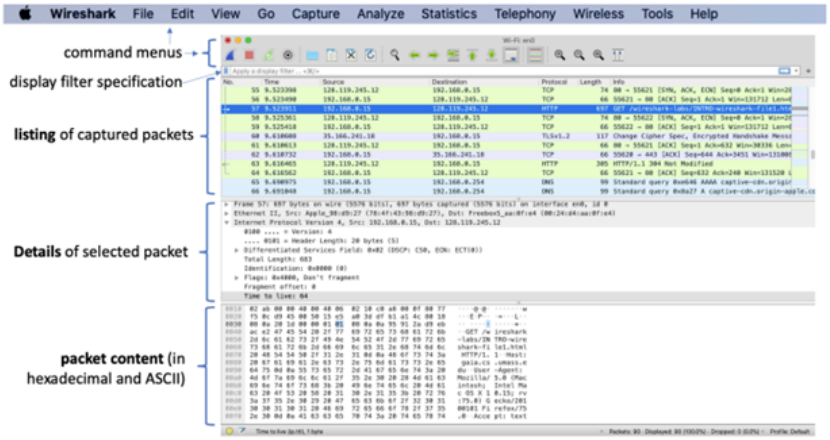

Figure 1: Wireshark window, during and after capture.

1.1.1 Packet List Pane

Figure 2 illustrates an example of a packet list pane, which allows you to view all captures packets. By default, the following information is displayed:

- **Packet relations:** The first column (to the left of the packet number) displays how each packet is related to the selected packet. Packets of the same conversation are connected using a line. See Figure 3 for more details.
- **No.:** The number of the packet in the capture file
- **Time:** Packet timestamp. By default, displays the time since packet capture started.
- **Source:** Source address.
- **Destination:** Destination address.
- **Protocol:** Abbreviated version of packet protocol name.
- **Length:** Packet length.
- **Info:** Additional information about packet content.

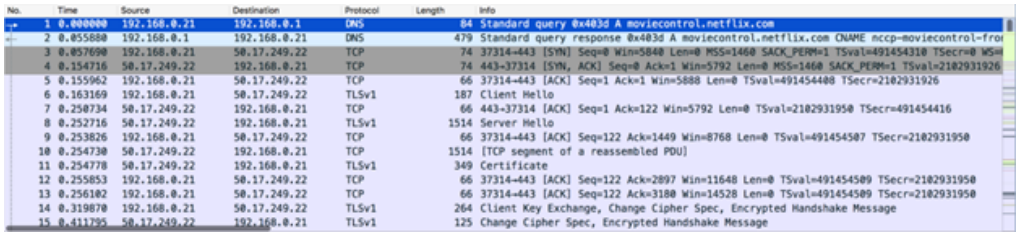

Figure 2: Packet List Pane.

Clicking on each individual packet allows you to view that packet's details in the packet details pane. Right mouse click on a packet results in a context menu with several functions. For instance, "Follow Stream" function allows you to view an entire conversation related to a given packet.

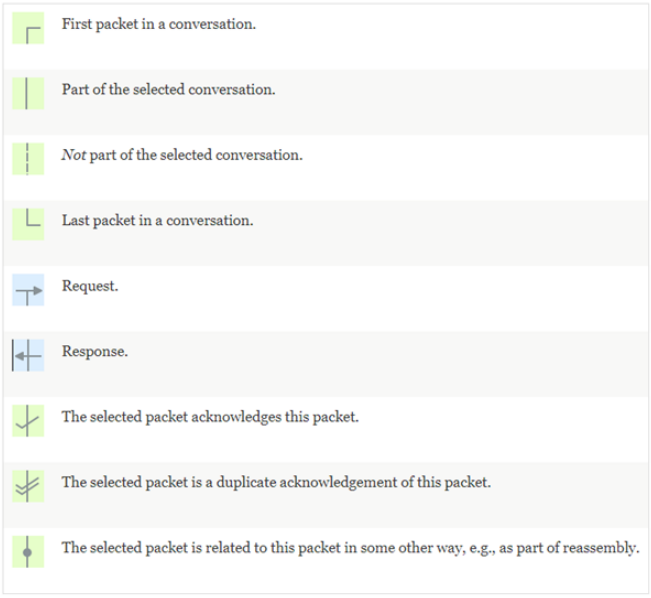

Figure 3: Related packet symbols.

1.1.2 Packet Details Pane

Figure 4 illustrates an example of a packet details pane that shows the protocols and protocol fields of the selected packet. Note that information enclosed in square brackets (*e.g.*, [Time: 055880000 seconds] in Figure 4) is not present in the captured data and is instead **generated by Wireshark**. Possible generated information by Wireshark includes response times, TCP analysis, IP geolocation information, and checksum validation. Moreover, Wireshark detects related packets (such as packets part of the same conversation) and includes links to these packets (*e.g.*, [Request In: 1] in Figure 4).

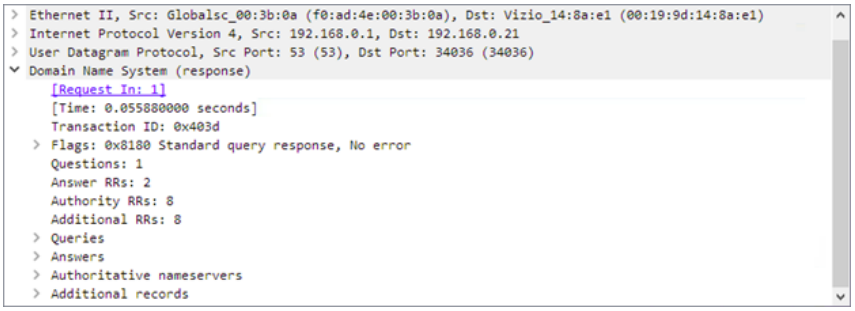

Figure 4: Packet details pane.

1.1.3 Packet Bytes Pane

Figure 5 illustrates an example of a packet bytes pane that shows shows the data of a given packet as hexadecimal and ASCII representations. Moreover, Wireshark allows plugins and additional protocols to create context-specific tabs. For instance, Figure 6 shows a packet bytes pane with an additional tabs that contains data from reassembling multiple TCP packets.

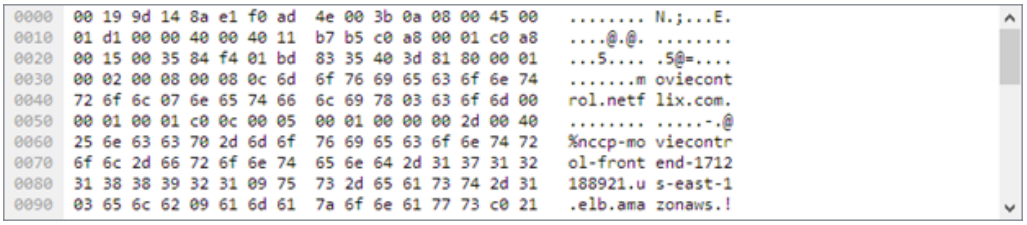

Figure 5: Packet bytes pane.

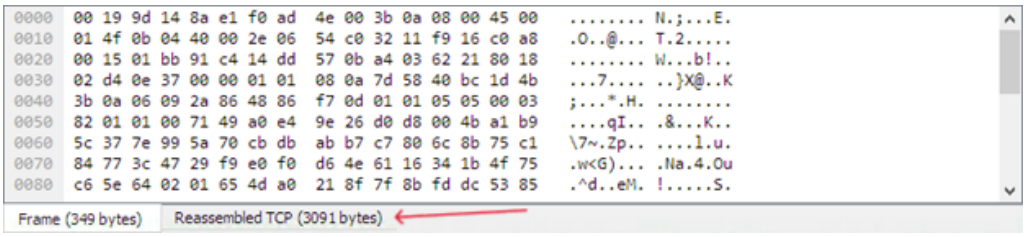

Figure 6: Packet bytes pane with additional tab.

1.2 Wireshark Statistics

Wireshark finds a variety of statistics about the network, packets, and users. The program creates charts and other data visualizations for convenient interpretation. In what follows, we review some of these statistics.

1.2.1 Protocol Hierarchy

This statistic provides a tree of all the protocols in the capture. Figure 7 illustrates an example of a protocol hierarchy window, where Each row contains the statistical values of one protocol.

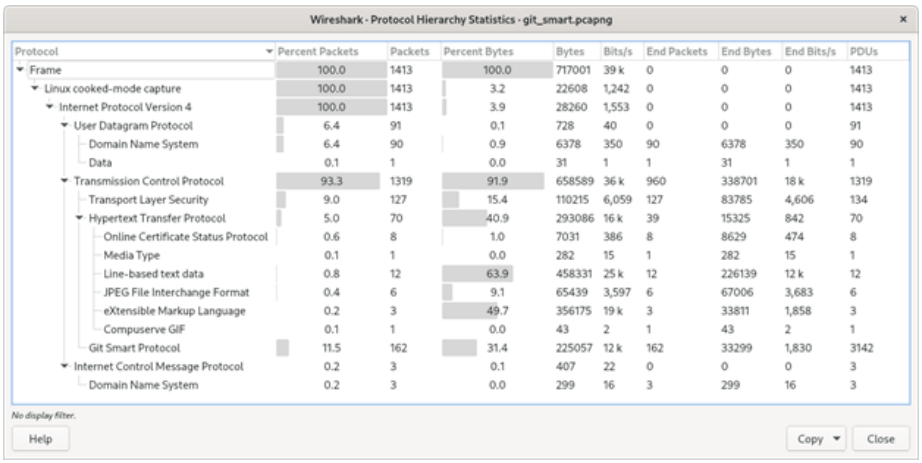

Figure 7: Protocol hierarchy window.

1.2.2 Conversations

This statistic provides a detailed view of communication between endpoints (*e.g.*, conversation start times, conversation duration, IP addresses, MAC addresses, number of packets exchanged, number of bytes exchanged). Figure 8 illustrates an example of a conversations window.

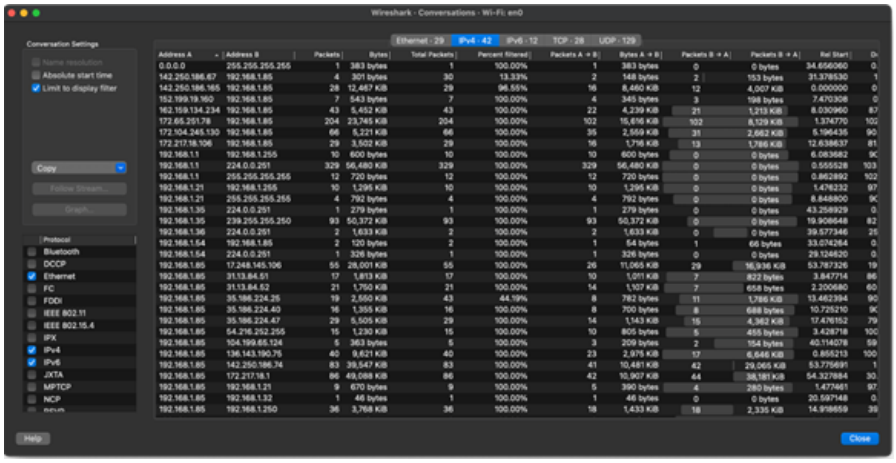

Figure 8: Protocol hierarchy window.

1.2.3 Flow Graph

This statistic Shows a sequence diagram of packet exchanges between devices, providing insights into communication flows and handshakes (*e.g.*, TCP connections). It creates a timeline of network activity, showing connections between hosts, including requests and responses. It provides packets time, direction, ports, and comments for each captured connection. Connections can be filtered by ICMP flows, ICMPv6 flows, UIM flows, and TCP flows. Figure 9 illustrates an example of a flow graph window. Vertical lines represent specific hosts and rows at the very left represent the time of each packet (since start of capture by default). The numbers at the end of each error display port numbers.

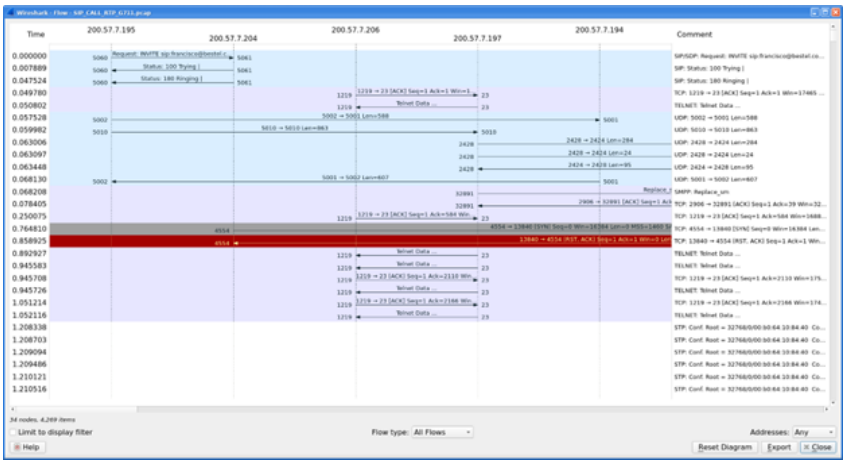

Figure 9: Flow graph window.

2 A Test Run

This testrun will showcase capturing and analyzing packets in Wireshark.

2.1 Capturing Packets

When loading Wireshark a window similar to Figure 10 is shown. This is where the interface where the packet capture will take place is specified.

1. eth0/en0: This is the default interface selected by Wireshark, and it should be selected if packets over the computer and external networks such as the internet need to be captured.
2. lo: This is the localhost interface and should be selected if packets sent over the loopback address 127.0.0.1 need to be captured.

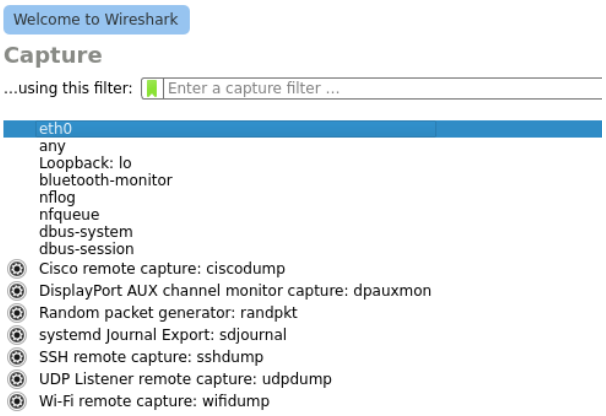

Figure 10: Wireshark interface selection window.

To capture packets the blue fin icon can be clicked in the top ribbon seen in Figure 11. Wireshark will start capturing packets and display them in the main window.

Figure 11: Wireshark Capturing packets.

To capture more interesting packets, simply visit a website through the browser. Ideally this website would be an http website, since https websites are encrypted and packets may not be captured.

1. Wireshark will begin capturing HTTP packets.
2. These packets include information such as the type of request (GET, POST) as well as the responses from the server.

2.2 Packet Analysis

After visiting the website the capture can be stopped by clicking the red square, and the packets can be analyzed.

1. The HTTP conversation can be located by searching for http in the search bar.
2. To find the request to the website locate a GET request in the filtered packets.
3. Right click and follow the HTTP stream to see more details.

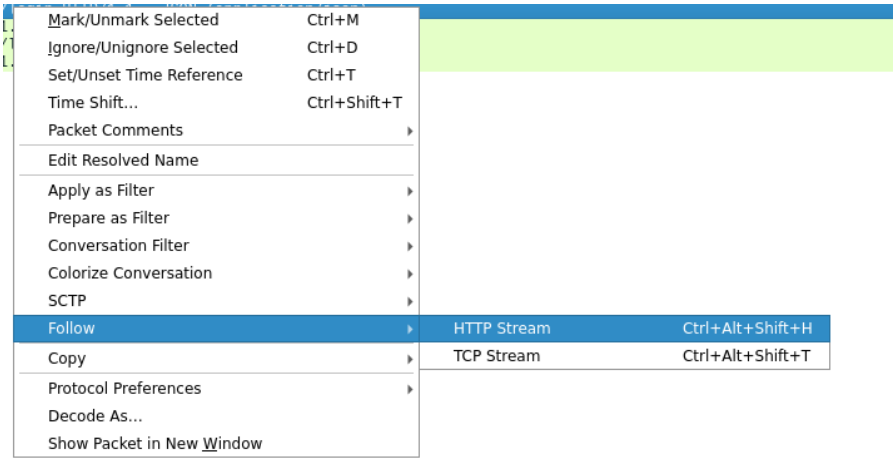

Figure 12: Following HTTP Stream.

Additionally, the packets display the encapsulated headers shown in Figure 13

1. Frame and Ethernet II: This part of the Link Layer including information such as the source and destination MAC address.
2. Internet Protocol (IPv4): This part of the Network Layer including information such as source and destination IP address.
3. Transmission Control Protocol (TCP): This part of the Transport Layer including information such as source and destination ports, sequence and acknowledgment numbers, and segment length.
4. Hypertext Transfer Protocol (HTTP): This is part of the Application Layer including the data being exchanged, such as HTTP requests or responses.

```
▸ Frame 459: 376 bytes on wire (3008 bits), 376 bytes captured (3008 bits) on interface eth0, id 0
▸ Ethernet II, Src: Microsoft_f1:bc:a0 (00:15:5d:f1:bc:a0), Dst: Microsoft_b7:e4:05 (00:15:5d:b7:e4:05)
▸ Internet Protocol Version 4, Src: 172.25.219.36, Dst: 34.107.221.82
▸ Transmission Control Protocol, Src Port: 53650, Dst Port: 80, Seq: 1, Ack: 1, Len: 310
▸ Hypertext Transfer Protocol
```

Figure 13: Encapsulated packet headers.

2.3 Supplementary Questions

1. What is the IP address of the website you visited, and what is the IP address of your computer?
2. Filter by DNS packets and determine the query name, type and class. Hint: look under the Domain Name System (query) header in the packet.
3. Determine the TTL (time to live) value of some of the transmitted packets, and determine if they were fragmented or not. Hint: this information is associated with the internet protocol (IP).

3 Submitting The Lab

The primary goal of WL0 is to help you get started with Wireshark. This lab is ungraded, does not contribute to your course grade, and requires no submission.