

MSCS631_WireShark_5

Wireshark – Lab 5: Wireshark: IP Protocol, Fragmentation, and IPv6

Samrat Baral

University of the Cumberlands

2025 Spring – Advanced Computer Networks (MSCS-631-M40) – Full Term

Dr. Yousef Nijim

March 14, 2025

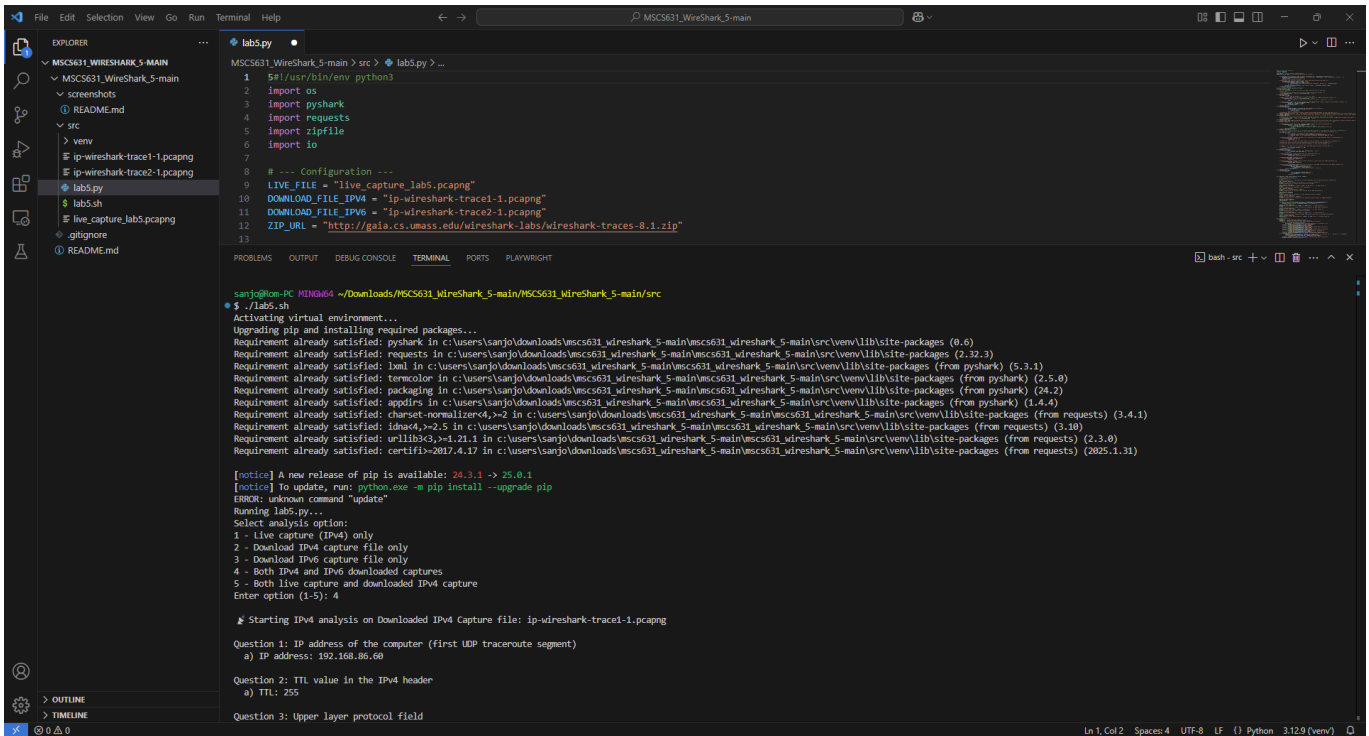
Lab Overview

In this lab, you will investigate the IPv4 and IPv6 protocols using packet traces. You will analyze:

- IPv4 datagrams captured during a traceroute session,
- The behavior of UDP and ICMP packets,
- IP fragmentation in large UDP segments,
- IPv6 DNS and traffic details.

The lab requires you to inspect packet headers in Wireshark, answer questions about header fields, fragmentation, and IPv6 addressing, and use display filters to focus on specific traffic.

Output Screenshots



The screenshot shows a VS Code editor with the file explorer on the left displaying the project structure for 'MSCS631_Wireshark_5-main'. The main editor window shows the 'lab5.py' file with the following code:

```
1 #!/usr/bin/env python3
2 import os
3 import pyshark
4 import requests
5 import zipfile
6 import io
7
8 # --- Configuration ---
9 LIVE_FILE = "live_capture_lab5.pcapng"
10 DOWNLOAD_FILE_IPV4 = "ip-wireshark-trace1-1.pcapng"
11 DOWNLOAD_FILE_IPV6 = "ip-wireshark-trace2-1.pcapng"
12 ZIP_URL = "http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces-8.1.zip"
13
```

The terminal window shows the execution of the script, which includes pip installation logs and a series of questions and answers related to network analysis:

```
sanjo@om-PC: ~/Downloads/MSCS631_Wireshark_5-main/MSCS631_Wireshark_5-main/src
$ ./lab5.sh
Upgrading pip and installing required packages...
Requirement already satisfied: pyshark in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (0.6)
Requirement already satisfied: requests in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (2.32.3)
Requirement already satisfied: lxml in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (from pyshark) (5.3.1)
Requirement already satisfied: termcolor in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (from pyshark) (2.5.0)
Requirement already satisfied: packaging in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (from pyshark) (24.2)
Requirement already satisfied: appdirs in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (from pyshark) (1.4.4)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (from requests) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (from requests) (2.3.0)
Requirement already satisfied: certifi<2017.4.17 in c:\users\sanjo\downloads\mscs631_wireshark_5-main\mscs631_wireshark_5-main\src\venv\lib\site-packages (from requests) (2025.1.31)

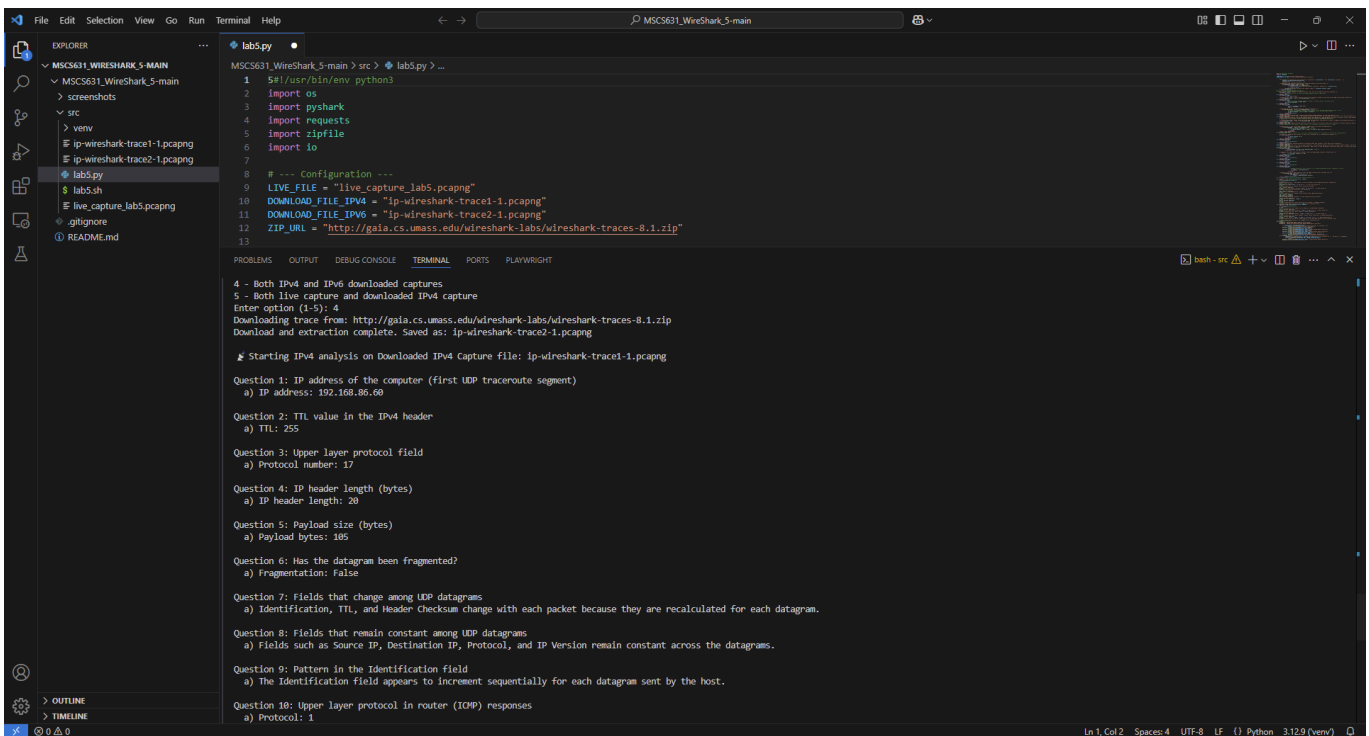
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
ERROR: unknown command "update"
Running lab5.py...
Select analysis option:
1 - Live capture (IPv4) only
2 - Download IPv4 capture file only
3 - Download IPv6 capture file only
4 - Both IPv4 and IPv6 downloaded captures
5 - Both live capture and downloaded IPv4 capture
Enter option (1-5): 4

$ Starting IPv4 analysis on Downloaded IPv4 Capture file: ip-wireshark-trace1-1.pcapng

Question 1: IP address of the computer (first UDP traceroute segment)
a) IP address: 192.168.86.60

Question 2: TTL value in the IPv4 header
a) TTL: 255

Question 3: Upper layer protocol field
```



The screenshot shows the continuation of the VS Code editor with the 'lab5.py' file and the terminal output. The terminal shows the script downloading a trace file and performing a series of questions and answers related to network analysis:

```
4 - Both IPv4 and IPv6 downloaded captures
5 - Both live capture and downloaded IPv4 capture
Enter option (1-5): 4
Downloading trace from: http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces-8.1.zip
Download and extraction complete. Saved as: ip-wireshark-trace2-1.pcapng

$ Starting IPv4 analysis on Downloaded IPv4 Capture file: ip-wireshark-trace1-1.pcapng

Question 1: IP address of the computer (first UDP traceroute segment)
a) IP address: 192.168.86.60

Question 2: TTL value in the IPv4 header
a) TTL: 255

Question 3: Upper layer protocol field
a) Protocol number: 17

Question 4: IP header length (bytes)
a) IP header length: 20

Question 5: Payload size (bytes)
a) Payload bytes: 105

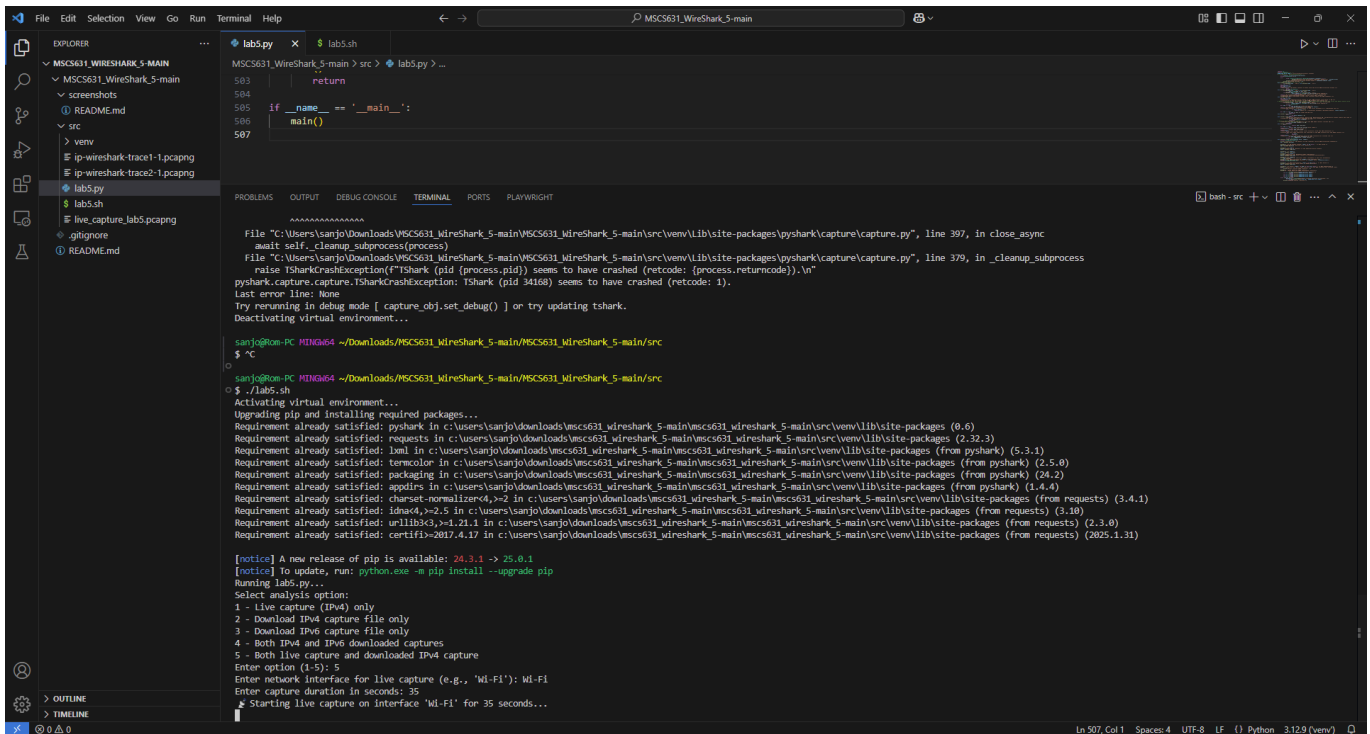
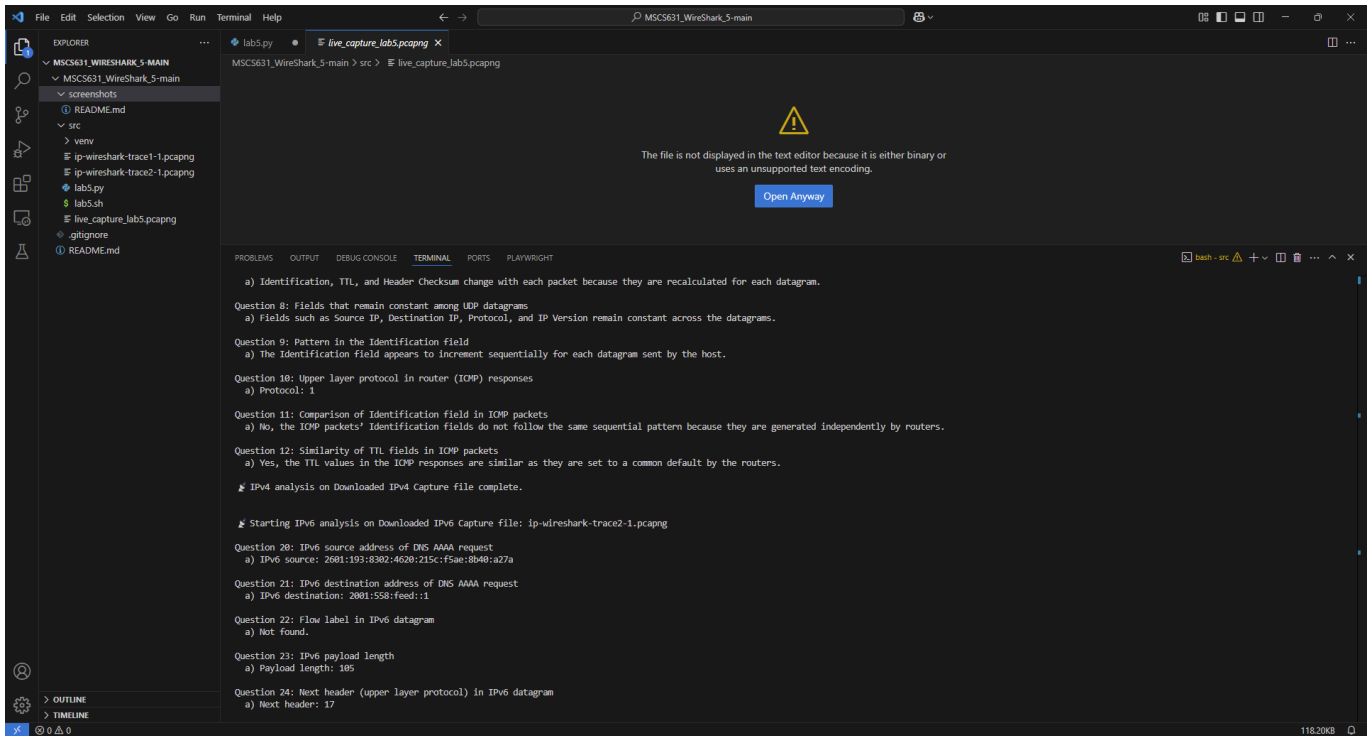
Question 6: Has the datagram been fragmented?
a) Fragmentation: False

Question 7: Fields that change among UDP datagrams
a) Identification, TTL, and Header checksum change with each packet because they are recalculated for each datagram.

Question 8: Fields that remain constant among UDP datagrams
a) Fields such as Source IP, Destination IP, Protocol, and IP Version remain constant across the datagrams.

Question 9: Pattern in the Identification field
a) The Identification field appears to increment sequentially for each datagram sent by the host.

Question 10: Upper layer protocol in router (ICMP) responses
a) Protocol: 1
```



Prerequisites

- **Python 3.x**
- **Tshark:** Ensure that Tshark is installed and available in your system's PATH.
Download from [Wireshark](#).
- **Pyshark:** Install via pip:

```
pip install pyshark
python3 lab5.py
```

Features

- Analysis of IPv4 header fields (TTL, Identification, etc.)
- Examination of UDP, ICMP, and fragmented datagrams
- IPv6 packet inspection and DNS AAAA request analysis

Lab Overview

In this lab, you will investigate the IPv4 and IPv6 protocols using packet traces. You will analyze:

- IPv4 datagrams captured during a traceroute session,
- The behavior of UDP and ICMP packets,
- IP fragmentation in large UDP segments,
- IPv6 DNS and traffic details.

The lab requires you to inspect packet headers in Wireshark, answer questions about header fields, fragmentation, and IPv6 addressing, and use display filters to focus on specific traffic.

```
pip install pyshark
```

- Download the required trace files (e.g., `ip-wireshark-trace1-1.pcapng` and `ip-wireshark-trace2-1.pcapng`) from the Wireshark Labs repository:

Wireshark Labs Trace Files

Lab Analysis and Answers

Part 1: Basic IPv4

Question 1: Select the first UDP segment sent by your computer via the traceroute command to `gaia.cs.umass.edu`. Expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?

IP address: 192.168.86.60

Question 2: What is the value in the time-to-live (TTL) field in this IPv4 datagram's header?

TTL: 255

Question 3: What is the value in the upper layer protocol field in this IPv4 datagram's header? [Note: the answers for Linux/macOS differ from Windows here].

Protocol number: 17

Question 4: How many bytes are in the IP header?

IP header length: 20

Question 5: How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

Payload bytes: 105

Question 6: Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

Fragmentation: False

Question 7: Which fields in the IP datagram always change from one datagram to the next within this series of UDP segments sent by your computer destined to 128.119.245.12, via traceroute? Why?

Identification, TTL, and Header Checksum change with each packet because they are recalculated for each datagram.

Question 8: Which fields in this sequence of IP datagrams (containing UDP segments) stay constant? Why?

Fields such as Source IP, Destination IP, Protocol, and IP Version remain constant across the datagrams.

Question 9: Describe the pattern you see in the values in the Identification field of the IP datagrams being sent by your computer.

The Identification field appears to increment sequentially for each datagram sent by the host.

Question 10: What is the upper layer protocol specified in the IP datagrams returned from the routers? [Note: the answers for Linux/macOS differ from Windows here].

Protocol: 1

Question 11: Are the values in the Identification fields (across the sequence of all of ICMP packets from all of the routers) similar in behavior to your answer to question 9 above?

No, the ICMP packets' Identification fields do not follow the same sequential pattern because they are generated independently by routers.

Question 12: Are the values of the TTL fields similar, across all of the ICMP packets from all of the routers?

Yes, the TTL values in the ICMP responses are similar as they are set to a common default by the routers.

Part 2: Fragmentation

Question 13: Find the first IP datagram containing the first part of the segment sent to 128.119.245.12 by your computer (using traceroute with a packet length of 3000 bytes). Has that segment been fragmented across more than one IP datagram?

Yes, the segment has been fragmented.

Question 14: What information in the IP header indicates that this datagram has been fragmented?

Fragment Offset: 185, Flags: 0x01

Question 15: What information in the IP header for this packet indicates whether this is the first fragment versus a latter fragment?

This is the first fragment (Fragment Offset is 0).

Question 16: How many bytes are there in this IP datagram (header plus payload)?

56

Question 17: Inspect the datagram containing the second fragment of the fragmented UDP segment. What information in the IP header indicates that this is not the first datagram fragment?

Fragment Offset is 185, indicating it is not the first fragment.

Question 18: What fields change in the IP header between the first and second fragment?

Between the first and second fragments, the Fragment Offset and Total Length fields change, while the Identification, Source IP, Destination IP, and Protocol fields remain the same.

Question 19: Find the IP datagram containing the third fragment of the original UDP segment. What information in the IP header indicates that this is the last fragment of that segment?

The absence of the 'More Fragments' (MF) flag indicates this is the last fragment.

Part 3: IPv6

Question 20: What is the IPv6 address of the computer making the DNS AAAA request? Give the IPv6 source address for this datagram in the exact same form as displayed in the Wireshark window.

IPv6 source: 2601:193:8302:4620:215c:f5ae:8b40:a27a

Question 21: What is the IPv6 destination address for this datagram? Give this IPv6 address in the exact same form as displayed in the Wireshark window.

IPv6 destination: 2001:558:feed::1

Question 22: What is the value of the flow label for this datagram?

Not found.

Question 23: How much payload data is carried in this datagram?

Payload length: 105

Question 24: What is the upper layer protocol to which this datagram's payload will be delivered at the destination?

Next header: 17

Question 25: How many IPv6 addresses are returned in the response to the AAAA request?

Not determined.

Question 26: What is the first of the IPv6 addresses returned by the DNS for youtube.com? Give this IPv6 address in the exact same shorthand form as displayed in the Wireshark window.

Not found