Packet Sniffer Analysis Report
Computer Networks HW 01

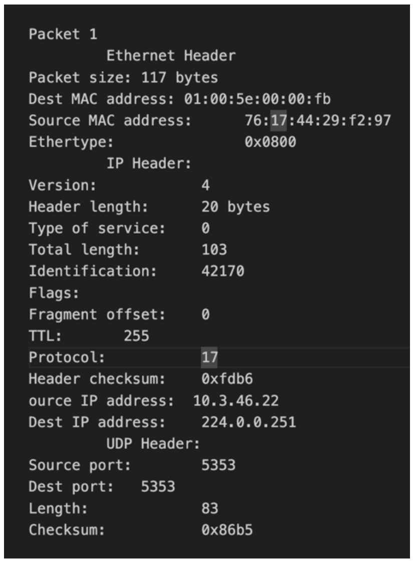
Siddhi Pandkar
sp9914@rit.edu

1. Introduction
This report presents the implementation and evaluation of a custom packet sniffer program written in Python using the Scapy library. The program reads packets from a PCAP file and extracts key header information from Ethernet, IP, TCP, UDP, and ICMP layers. The objective of this assignment was to correctly parse network packets and verify the output by comparing it with Wireshark.

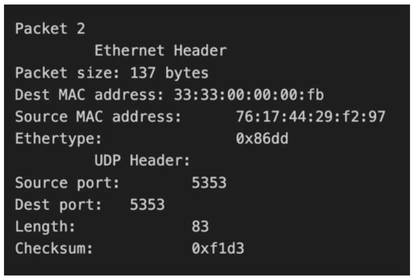
2. Methodology
The packet sniffer program was developed in Python using the Scapy library to read packets from an offline PCAP file. The program extracts and displays:

- Ethernet header fields
- IP header fields
- TCP header fields
- UDP header fields
- ICMP header fields

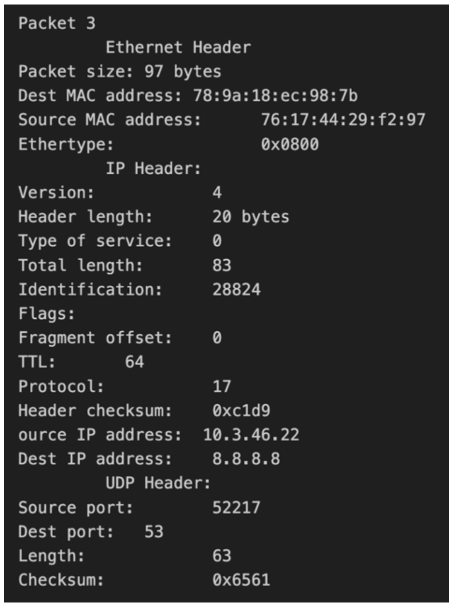
3. First 5 packets and last 5 packets by pktsniffer.py:



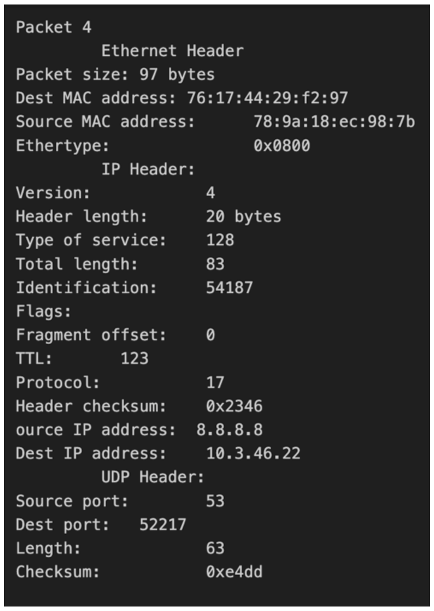
```
Packet 1
         Ethernet Header
Packet size: 117 bytes
Dest MAC address: 01:00:5e:00:00:fb
Source MAC address:      76:17:44:29:f2:97
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    0
Total length:       103
Identification:     42170
Flags:
Fragment offset:    0
TTL:       255
Protocol:           17
Header checksum:    0xfdb6
ource IP address:  10.3.46.22
Dest IP address:    224.0.0.251
         UDP Header:
Source port:        5353
Dest port:   5353
Length:             83
Checksum:           0x86b5
```



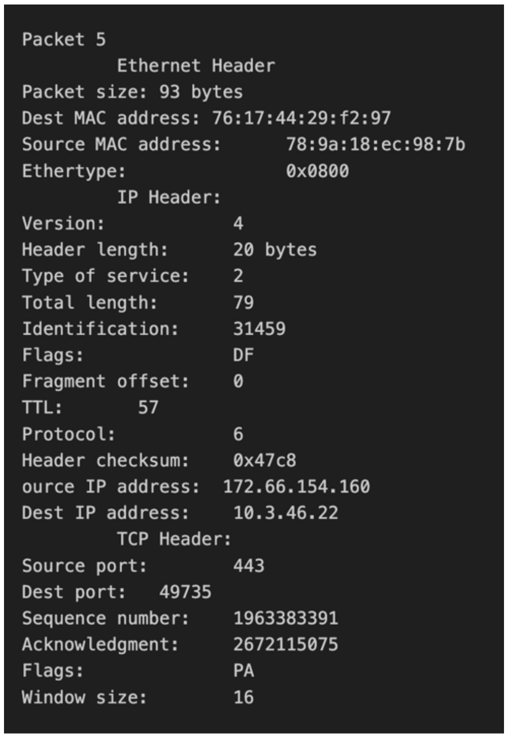
```
Packet 2
         Ethernet Header
Packet size: 137 bytes
Dest MAC address: 33:33:00:00:00:fb
Source MAC address:      76:17:44:29:f2:97
Ethertype:               0x86dd
         UDP Header:
Source port:        5353
Dest port:   5353
Length:             83
Checksum:           0xf1d3
```



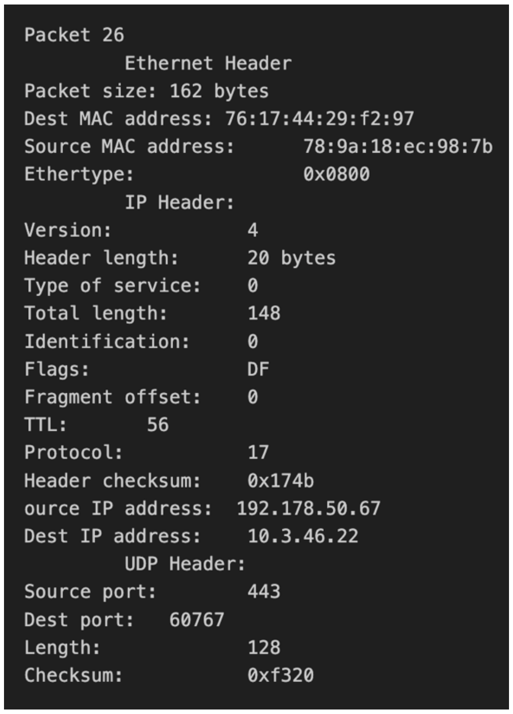
```
Packet 3
         Ethernet Header
Packet size: 97 bytes
Dest MAC address: 78:9a:18:ec:98:7b
Source MAC address:      76:17:44:29:f2:97
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    0
Total length:       83
Identification:     28824
Flags:
Fragment offset:    0
TTL:       64
Protocol:           17
Header checksum:    0xc1d9
ource IP address:  10.3.46.22
Dest IP address:    8.8.8.8
         UDP Header:
Source port:        52217
Dest port:   53
Length:             63
Checksum:           0x6561
```



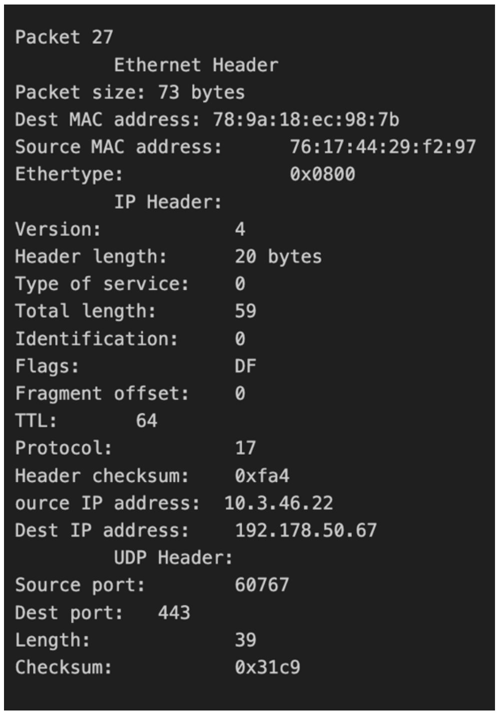
```
Packet 4
         Ethernet Header
Packet size: 97 bytes
Dest MAC address: 76:17:44:29:f2:97
Source MAC address:      78:9a:18:ec:98:7b
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    128
Total length:       83
Identification:     54187
Flags:
Fragment offset:    0
TTL:       123
Protocol:           17
Header checksum:    0x2346
ource IP address:  8.8.8.8
Dest IP address:    10.3.46.22
         UDP Header:
Source port:        53
Dest port:   52217
Length:             63
Checksum:           0xe4dd
```



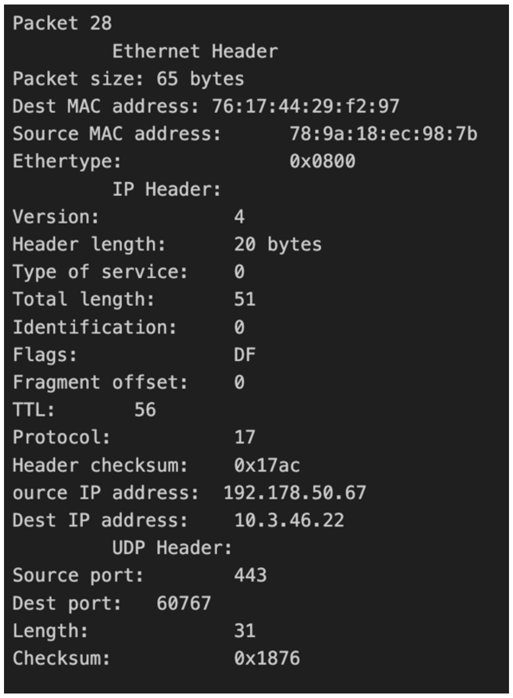
```
Packet 5
         Ethernet Header
Packet size: 93 bytes
Dest MAC address: 76:17:44:29:f2:97
Source MAC address:      78:9a:18:ec:98:7b
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    2
Total length:       79
Identification:     31459
Flags:              DF
Fragment offset:    0
TTL:       57
Protocol:           6
Header checksum:    0x47c8
ource IP address:  172.66.154.160
Dest IP address:    10.3.46.22
         TCP Header:
Source port:        443
Dest port:   49735
Sequence number:    1963383391
Acknowledgment:     2672115075
Flags:              PA
Window size:        16
```



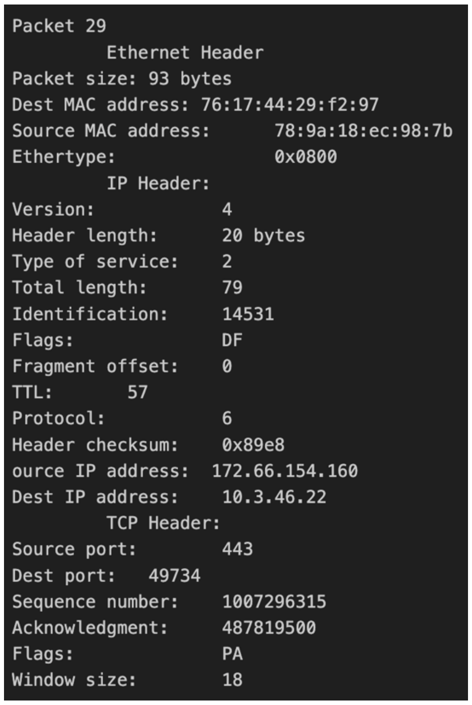
```
Packet 26
         Ethernet Header
Packet size: 162 bytes
Dest MAC address: 76:17:44:29:f2:97
Source MAC address:      78:9a:18:ec:98:7b
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    0
Total length:       148
Identification:     0
Flags:              DF
Fragment offset:    0
TTL:       56
Protocol:           17
Header checksum:    0x174b
ource IP address:  192.178.50.67
Dest IP address:    10.3.46.22
         UDP Header:
Source port:        443
Dest port:   60767
Length:             128
Checksum:           0xf320
```



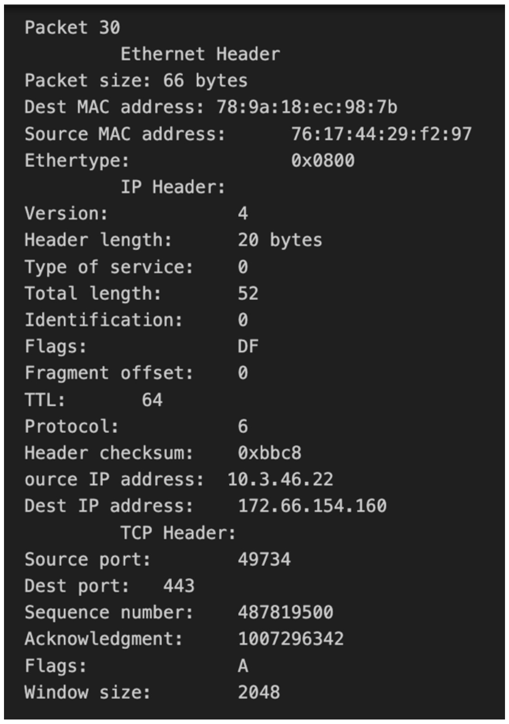
```
Packet 27
         Ethernet Header
Packet size: 73 bytes
Dest MAC address: 78:9a:18:ec:98:7b
Source MAC address:      76:17:44:29:f2:97
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    0
Total length:       59
Identification:     0
Flags:              DF
Fragment offset:    0
TTL:       64
Protocol:           17
Header checksum:    0xfa4
ource IP address:  10.3.46.22
Dest IP address:    192.178.50.67
         UDP Header:
Source port:        60767
Dest port:   443
Length:             39
Checksum:           0x31c9
```



```
Packet 28
         Ethernet Header
Packet size: 65 bytes
Dest MAC address: 76:17:44:29:f2:97
Source MAC address:      78:9a:18:ec:98:7b
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    0
Total length:       51
Identification:     0
Flags:              DF
Fragment offset:    0
TTL:       56
Protocol:           17
Header checksum:    0x17ac
ource IP address:  192.178.50.67
Dest IP address:    10.3.46.22
         UDP Header:
Source port:        443
Dest port:   60767
Length:             31
Checksum:           0x1876
```

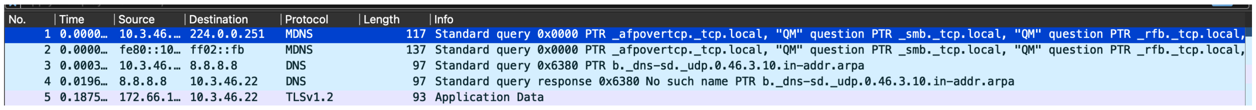


```
Packet 29
         Ethernet Header
Packet size: 93 bytes
Dest MAC address: 76:17:44:29:f2:97
Source MAC address:      78:9a:18:ec:98:7b
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    2
Total length:       79
Identification:     14531
Flags:              DF
Fragment offset:    0
TTL:       57
Protocol:           6
Header checksum:    0x89e8
ource IP address:  172.66.154.160
Dest IP address:    10.3.46.22
         TCP Header:
Source port:        443
Dest port:   49734
Sequence number:    1007296315
Acknowledgment:     487819500
Flags:              PA
Window size:        18
```

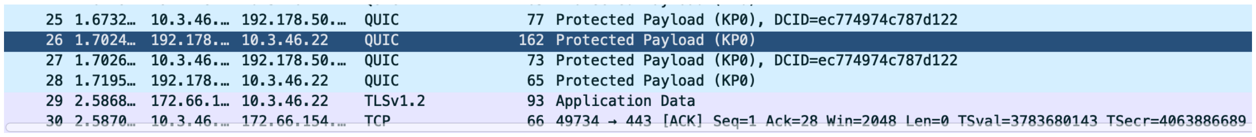


```
Packet 30
         Ethernet Header
Packet size: 66 bytes
Dest MAC address: 78:9a:18:ec:98:7b
Source MAC address:      76:17:44:29:f2:97
Ethertype:               0x0800
         IP Header:
Version:            4
Header length:      20 bytes
Type of service:    0
Total length:       52
Identification:     0
Flags:              DF
Fragment offset:    0
TTL:       64
Protocol:           6
Header checksum:    0xbbc8
ource IP address:  10.3.46.22
Dest IP address:    172.66.154.160
         TCP Header:
Source port:        49734
Dest port:   443
Sequence number:    487819500
Acknowledgment:     1007296342
Flags:              A
Window size:        2048
```

4. Wireshark first 5 and last 5 entries:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000…	10.3.46.…	224.0.0.251	MDNS	117	Standard query 0x0000 PTR _afpovertcp._tcp.local, "QM" question PTR _smb._tcp.local, "QM" question PTR _rfb._tcp.local,
2	0.0000…	fe80::10…	ff02::fb	MDNS	137	Standard query 0x0000 PTR _afpovertcp._tcp.local, "QM" question PTR _smb._tcp.local, "QM" question PTR _rfb._tcp.local,
3	0.0003…	10.3.46.…	8.8.8.8	DNS	97	Standard query 0x6380 PTR b._dns-sd._udp.0.46.3.10.in-addr.arpa
4	0.0196…	8.8.8.8	10.3.46.22	DNS	97	Standard query response 0x6380 No such name PTR b._dns-sd._udp.0.46.3.10.in-addr.arpa
5	0.1875…	172.66.1…	10.3.46.22	TLSv1.2	93	Application Data



25	1.6732…	10.3.46.…	192.178.50.…	QUIC	77	Protected Payload (KP0), DCID=ec774974c787d122
26	1.7024…	192.178.…	10.3.46.22	QUIC	162	Protected Payload (KP0)
27	1.7026…	10.3.46.…	192.178.50.…	QUIC	73	Protected Payload (KP0), DCID=ec774974c787d122
28	1.7195…	192.178.…	10.3.46.22	QUIC	65	Protected Payload (KP0)
29	2.5868…	172.66.1…	10.3.46.22	TLSv1.2	93	Application Data
30	2.5870…	10.3.46.…	172.66.154.…	TCP	66	49734 → 443 [ACK] Seq=1 Ack=28 Win=2048 Len=0 TSval=3783680143 TSecr=4063886689

5. Packet Comparison Results

First Packets Comparison
he first few packets from the program were compared with the same packets in Wireshark. The Ethernet, IP, and UDP details such as addresses, protocol, and ports were the same in both. This shows that the program reads and displays packet information correctly.

Last Packets Comparison
The last few packets of the capture file were similarly analyzed in the custom tool and Wireshark as well. The header information was identical for both tools. This shows that   packet sniffer is functioning correctly with packets from the entire capture file.

6. Filtering Functionality

Filtering feature of the program was tested for filters with TCP, UDP ICMP\textunderscore and port 80. If we set the TCP filter this would be displayed and drop everything down one level. The UDP filter did also only display UDP packets, and your ICMP filter showed only ICMP packets, so everything seems to work in a sense of detecting the different prorocols. When port 80 filter was applied, there were output only for TCP packets associated with HTTP traffic and it showed all of the packets with a source or destination port 80. These program illustrates that packets can be filtered right down to the protocol type and port number.

7. Interesting Observation:

What's interesting in these packets is that in the captured packets, there are a lot of multicast DNS (mDNS) targeting destination address 224.0.0.251 on port 5353 from one of the participants.. Tthese packets are applicable to local network service discovery, e.g., printers, media devices or computers around you. This means that several devices on the network were looking for services during the period of capture.

8. Conclusion

The custom packet sniffer successfully parsed Ethernet, IP, TCP, UDP, and ICMP headers from the provided PCAP file. The results were validated against Wireshark and were found to be consistent.
The filtering functionality worked as expected, correctly isolating packets based on protocol type and port number.
The analysis of the captured traffic revealed common network activities such as mDNS service discovery, DNS queries, and encrypted HTTPS communication.
Overall, the implementation meets the assignment requirements and demonstrates correct packet parsing and filtering behavior.