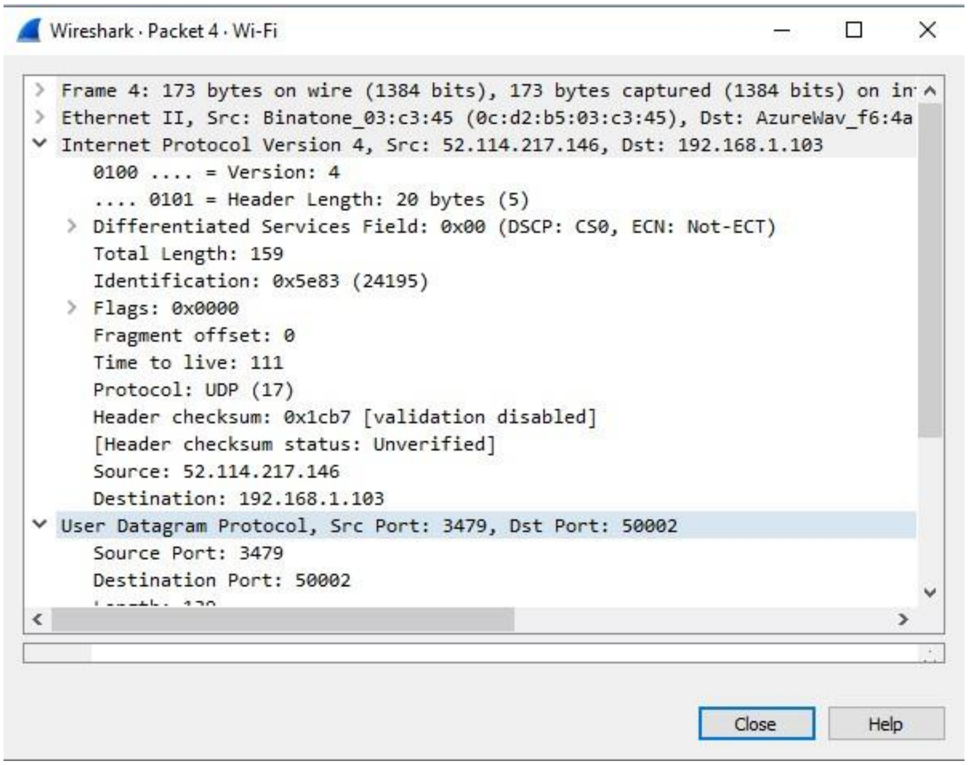
Experiment No 6

B.1: Code of performed experiment

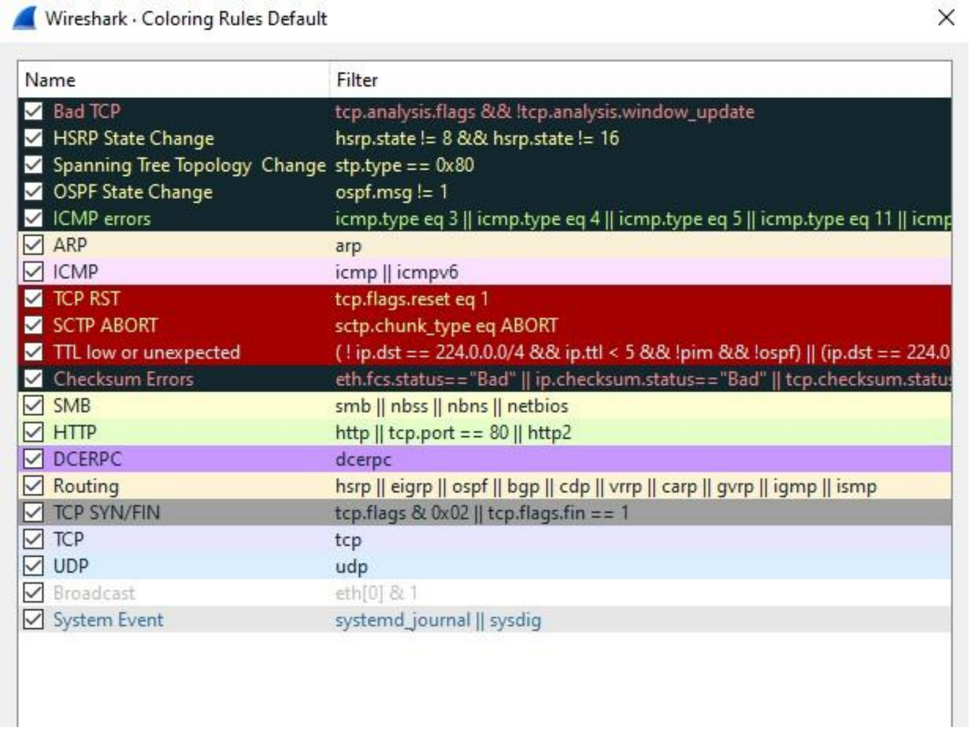
(Students are expected to write the code of performed experiment)

FRAME CONSIDERED:



```
> Frame 4: 173 bytes on wire (1384 bits), 173 bytes captured (1384 bits) on in
> Ethernet II, Src: Binatone_03:c3:45 (0c:d2:b5:03:c3:45), Dst: AzureWav_f6:4a
v Internet Protocol Version 4, Src: 52.114.217.146, Dst: 192.168.1.103
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 159
    Identification: 0x5e83 (24195)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 111
    Protocol: UDP (17)
    Header checksum: 0x1cb7 [validation disabled]
    [Header checksum status: Unverified]
    Source: 52.114.217.146
    Destination: 192.168.1.103
v User Datagram Protocol, Src Port: 3479, Dst Port: 50002
    Source Port: 3479
    Destination Port: 50002
    Length: 139
```

1. Incoming Packet
2. 52.114.217.146
3. 192.168.1.103
4. 173 Bytes
5. 14 bytes
6. 20 bytes (header) + 159 Bytes
7. 20 bytes (header) +700 Bytes 8.



Wireshark · Coloring Rules Default

<table>
<tr><th>Name</th><th>Filter</th></tr>
<tr><td>☑ Bad TCP</td><td>tcp.analysis.flags && !tcp.analysis.window_update</td></tr>
<tr><td>☑ HSRP State Change</td><td>hsrp.state != 8 && hsrp.state != 16</td></tr>
<tr><td>☑ Spanning Tree Topology Change</td><td>stp.type == 0x80</td></tr>
<tr><td>☑ OSPF State Change</td><td>ospf.msg != 1</td></tr>
<tr><td>☑ ICMP errors</td><td>icmp.type eq 3 || icmp.type eq 4 || icmp.type eq 5 || icmp.type eq 11 || icmp</td></tr>
<tr><td>☑ ARP</td><td>arp</td></tr>
<tr><td>☑ ICMP</td><td>icmp || icmpv6</td></tr>
<tr><td>☑ TCP RST</td><td>tcp.flags.reset eq 1</td></tr>
<tr><td>☑ SCTP ABORT</td><td>sctp.chunk_type eq ABORT</td></tr>
<tr><td>☑ TTL low or unexpected</td><td>(! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim && !ospf) || (ip.dst == 224.0</td></tr>
<tr><td>☑ Checksum Errors</td><td>eth.fcs.status=="Bad" || ip.checksum.status=="Bad" || tcp.checksum.statu</td></tr>
<tr><td>☑ SMB</td><td>smb || nbss || nbns || netbios</td></tr>
<tr><td>☑ HTTP</td><td>http || tcp.port == 80 || http2</td></tr>
<tr><td>☑ DCERPC</td><td>dcerpc</td></tr>
<tr><td>☑ Routing</td><td>hsrp || eigrp || ospf || bgp || cdp || vrrp || carp || gvrp || igmp || ismp</td></tr>
<tr><td>☑ TCP SYN/FIN</td><td>tcp.flags & 0x02 || tcp.flags.fin == 1</td></tr>
<tr><td>☑ TCP</td><td>tcp</td></tr>
<tr><td>☑ UDP</td><td>udp</td></tr>
<tr><td>☑ Broadcast</td><td>eth[0] & 1</td></tr>
<tr><td>☑ System Event</td><td>systemd_journal || sysdig</td></tr>
</table>

B.2: Observations and Learning’s:

In this experiment, we learnt how packets are sent between IP address. We also saw about the information contained in the packets.

B.3: Conclusion:

We conclude that we could successfully retrieve information regarding the packets. We could figure out information like: IP header, TCP header, total number of bytes in a certain frame etc.