Wireshark : UDP

Aim

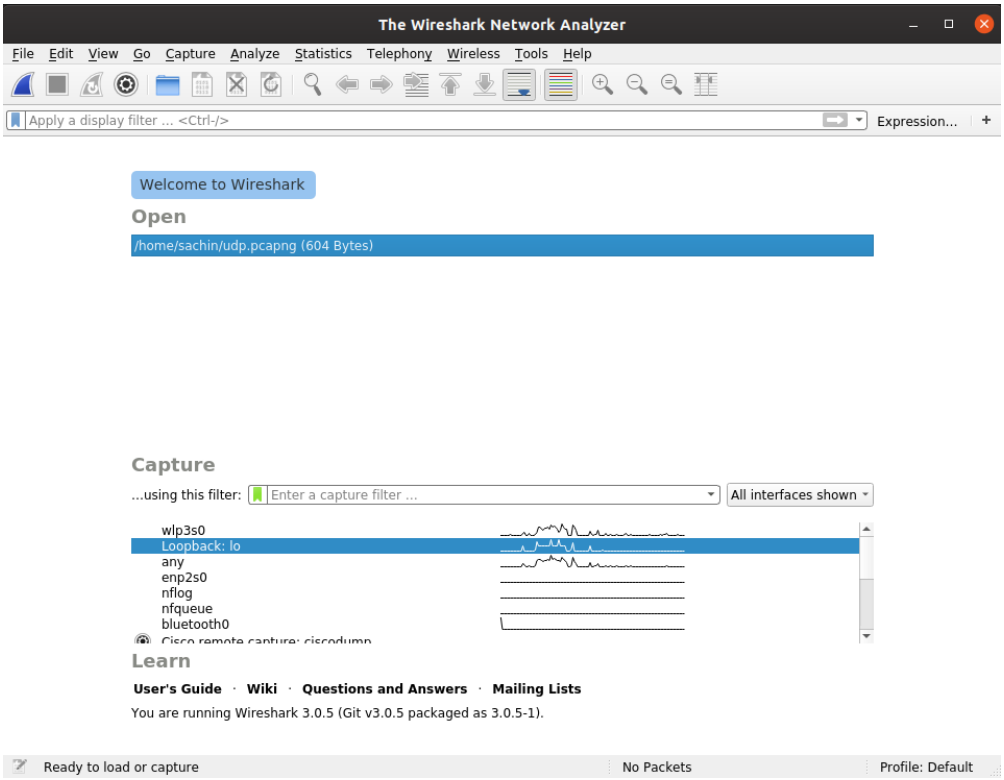
Using Wireshark observe data transferred in client server communication using UDP and identify the UDP datagram.

Theory

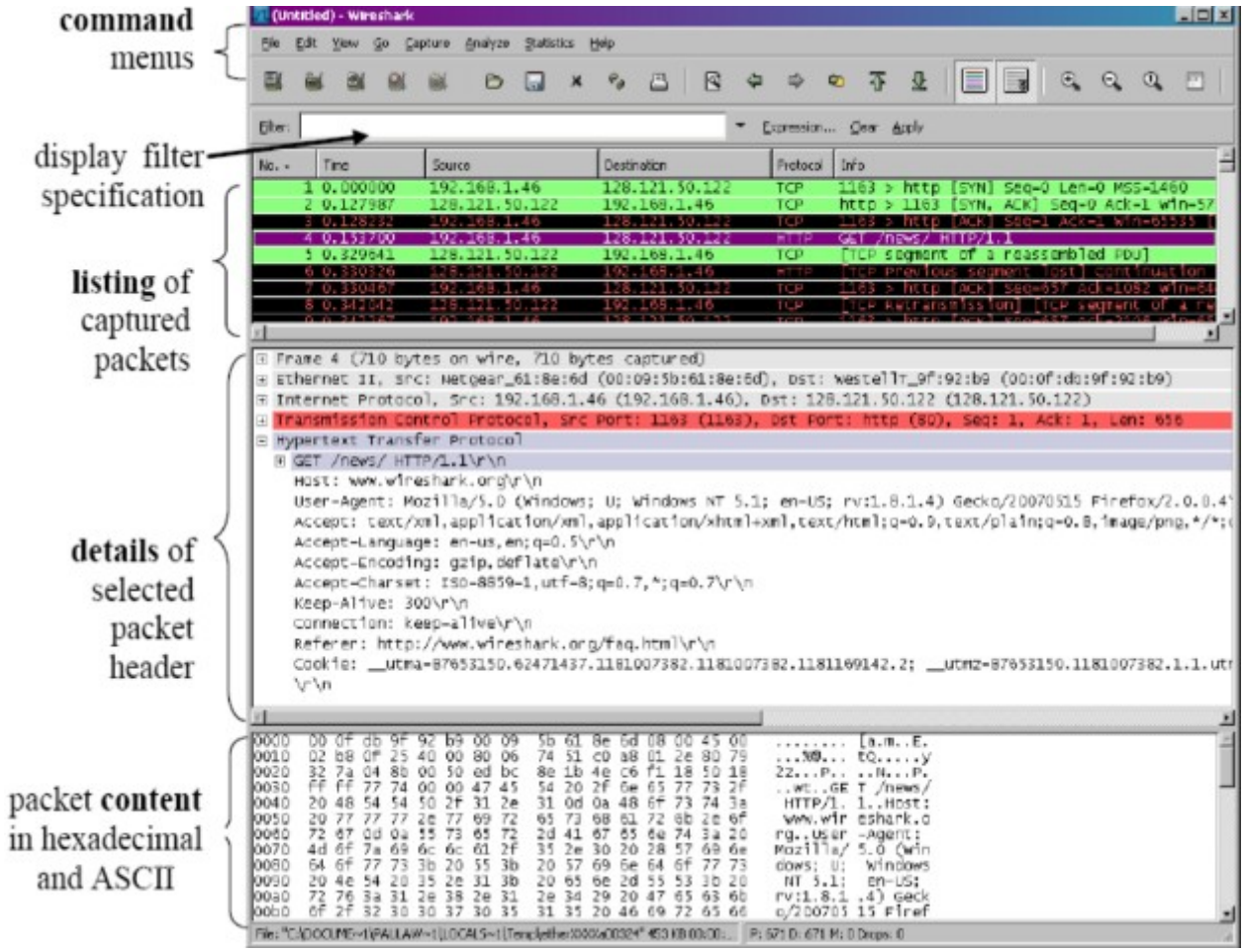
Wireshark

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. Some of the main uses include:

- Network administrators use it to troubleshoot network problems
- Developers use it to debug protocol implementations
- QA engineers use it to verify network applications
- Network security engineers use it to examine security problems
- People use it to learn network protocol internals



Wireshark Interface

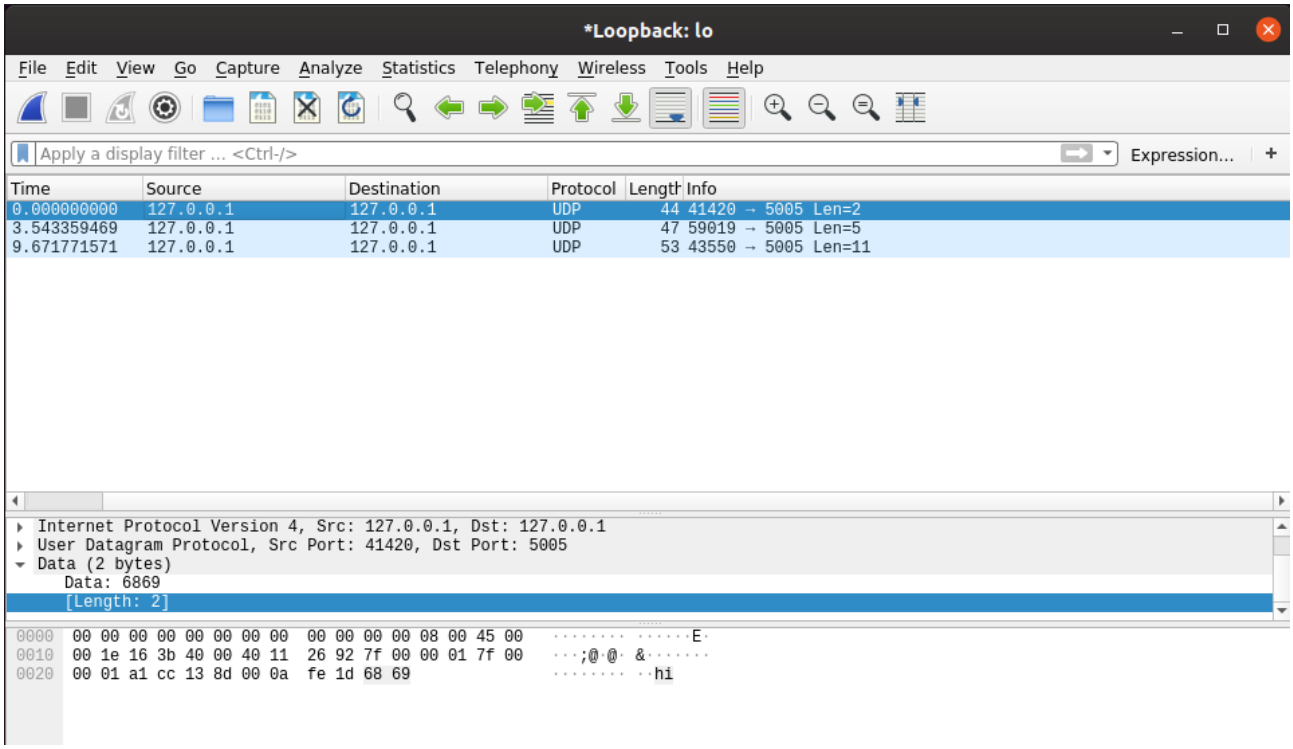


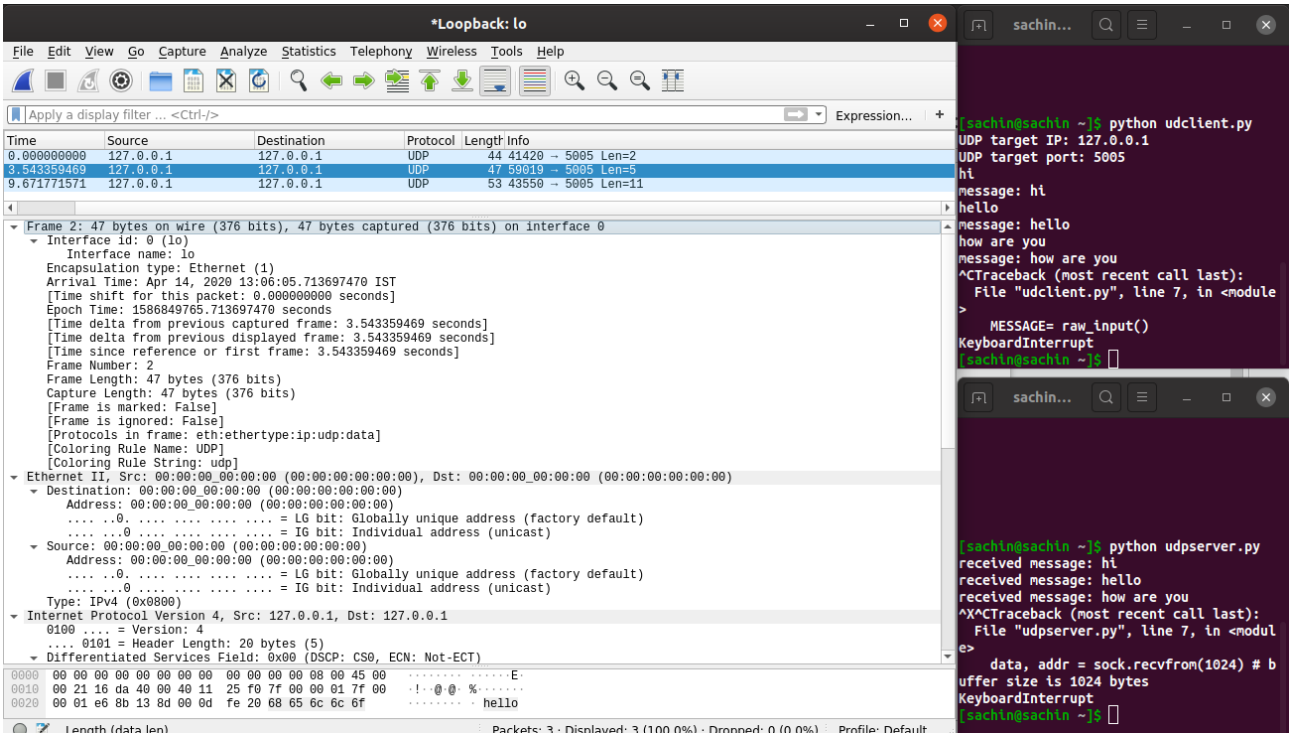
Wireshark components

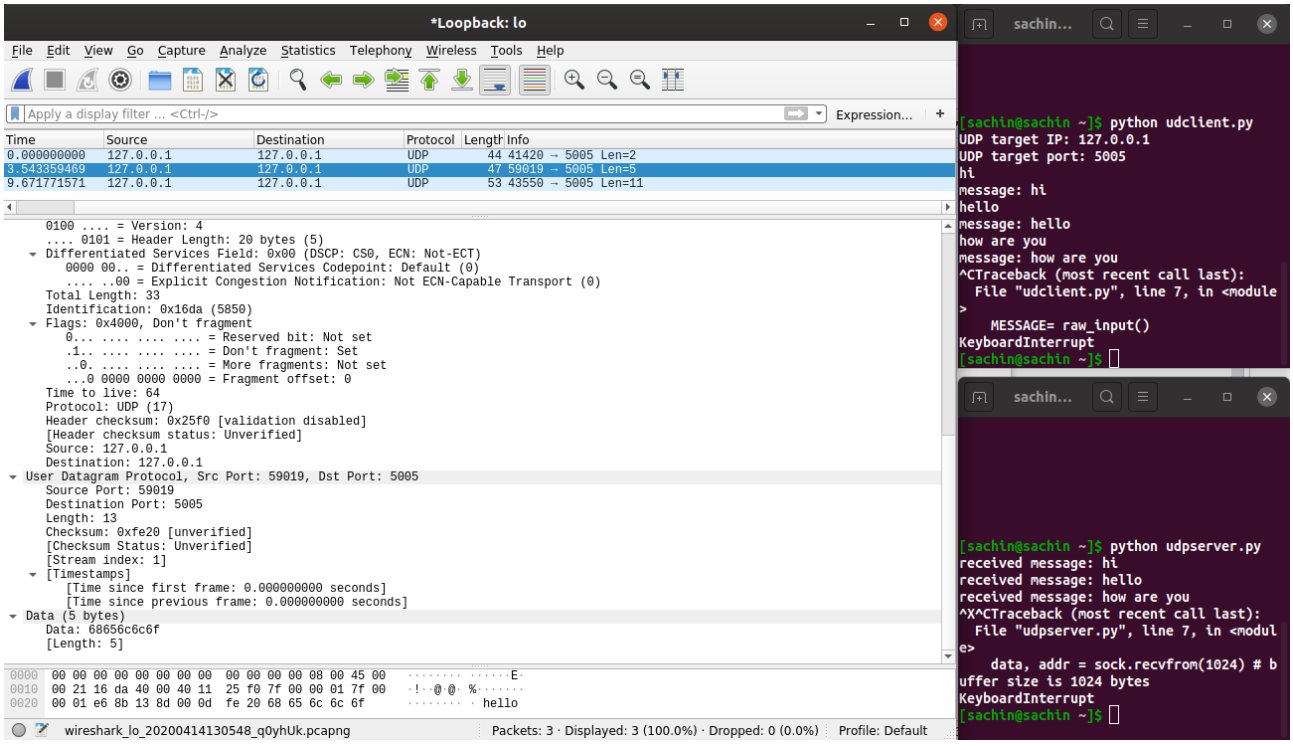
Capturing Packets: After downloading and installing Wireshark, one can launch and select a network interface under Capture to start capturing packets on that interface. That is, if you want to capture traffic on your bluetooth interface, select the bluetooth interface. Advanced features can be configured by clicking Options in Capture tab. As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system. If you have promiscuous mode enabled(default), you'll also see all the other packets on the network instead of only packets addressed to your network adapter.

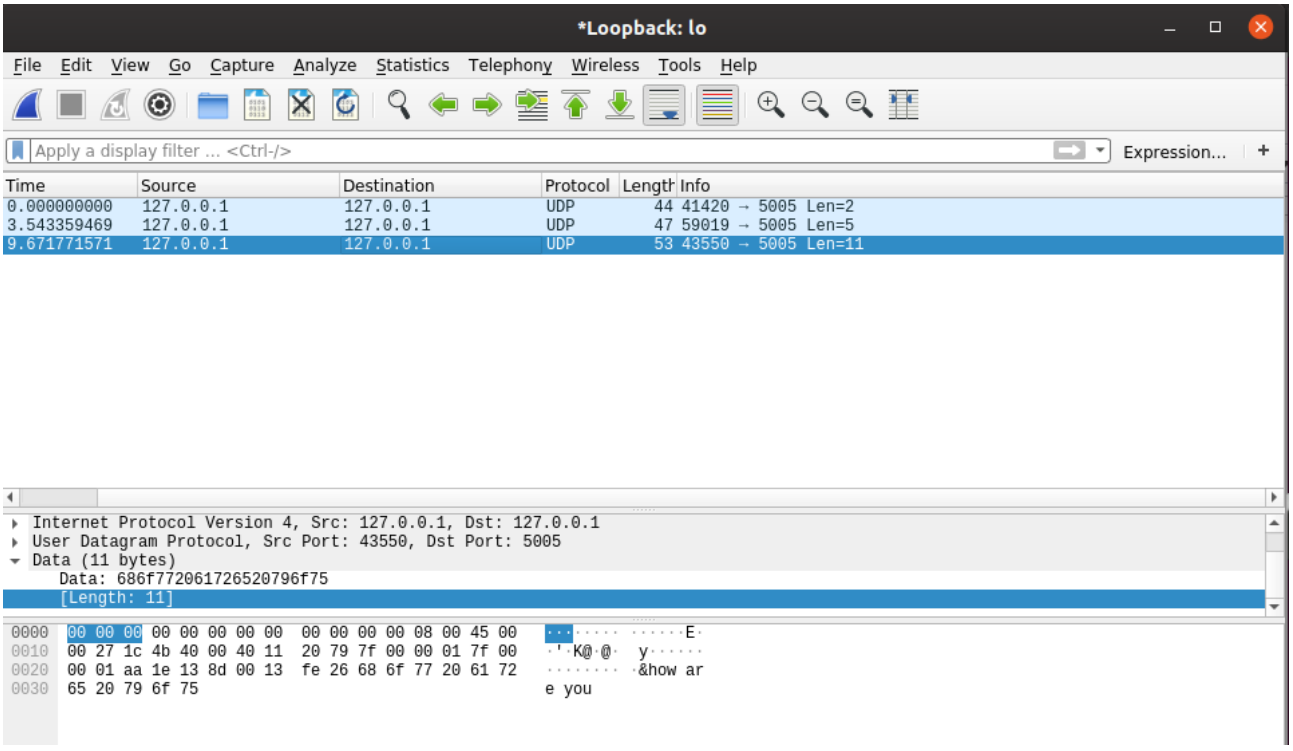
Filtering Packets: For inspecting something specific, filtering comes in handy. The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns" and you'll see only DNS packets.

Output


*Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Time Source Destination Protocol Length Info
0.000000000 127.0.0.1 127.0.0.1 UDP 44 41420 → 5005 Len=2
3.543359469 127.0.0.1 127.0.0.1 UDP 47 59019 → 5005 Len=5
9.671771571 127.0.0.1 127.0.0.1 UDP 53 43550 → 5005 Len=11
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
User Datagram Protocol, Src Port: 41420, Dst Port: 5005
Data (2 bytes)
Data: 6869
[Length: 2]
0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
0010 00 1e 16 3b 40 00 40 11 26 92 7f 00 00 01 7f 00
0020 00 01 a1 cc 13 8d 00 0a fe 1d 68 69
hi


*Loopback: lo
sachin...
[sachin@sachin ~]$ python udclient.py
UDP target IP: 127.0.0.1
UDP target port: 5005
hi
message: hi
hello
message: hello
how are you
message: how are you
^CTraceback (most recent call last):
File "udclient.py", line 7, in <module>
MESSAGE= raw_input()
KeyboardInterrupt
[sachin@sachin ~]$
[sachin@sachin ~]$ python udpserver.py
received message: hi
received message: hello
received message: how are you
^X^CTraceback (most recent call last):
File "udpserver.py", line 7, in <module>
data, addr = sock.recvfrom(1024) # buffer size is 1024 bytes
KeyboardInterrupt
[sachin@sachin ~]$





Result

Wireshark was installed and Packet transfer using UDP was observed .