Experiment No. 8

Aim: Study of packet sniffer tools: wireshark

1. Download and install wire shark and capture icmp, tcp, and http packets in promiscuous mode.
2. Explore how the packets can be traced based on different filters

Objectives:

- Understand the need for traffic analysis.
- Understand the how packet sniffing is done using wire shark.
- Trace and understand various packets from dynamic traffic.

Outcomes: The learner will be able to

- Sniff network packets and study insights of packets to get detail network information.

Hardware / Software Required: Unix/Linux/Windows, wire shark

Theory:

Wire shark, a network analysis tool formerly known as Ethereal, captures packets in real time and display the min human-readable format. Wire shark includes filters, color-coding and other features that let you dig deep into network traffic and inspect individual packets.

Features of Wire shark:

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Openfilescontainingpacketdatacapturedwithtcpdump/WinDump,Wireshark,anda
- Number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Export some or all packets in a number of capture file formats.

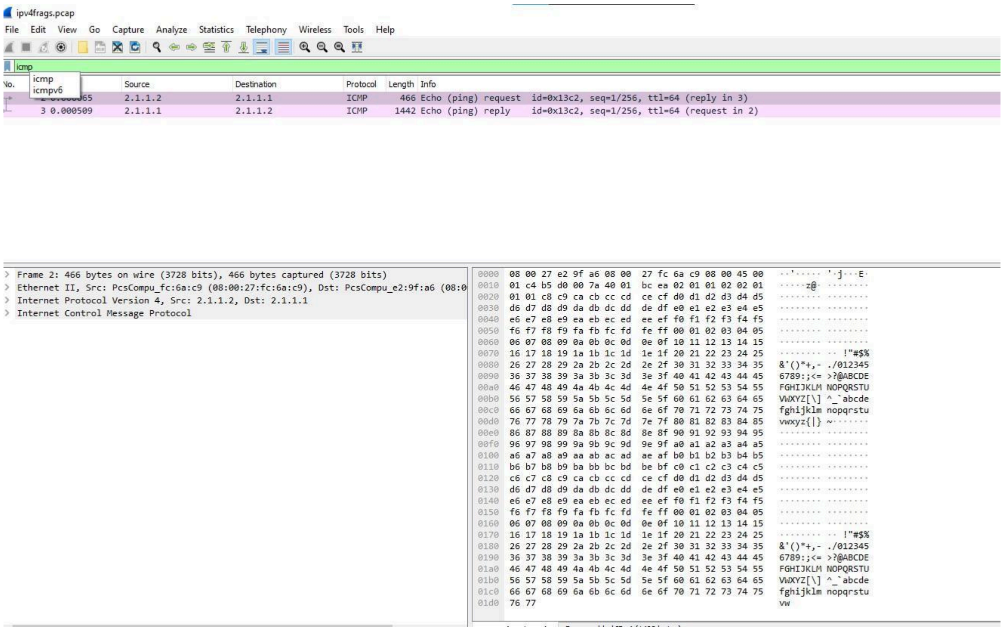
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.

Capturing Packets

After downloading and installing wire shark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

As soon as you click the interface‘s name, you‘ll see the packets start to appearing real time.

Wire shark captures each packet sent to or from your system. If you‘re capturing on a wireless interface and have promiscuous mode enabled in your capture options, you‘ll also see other the other packets on the network

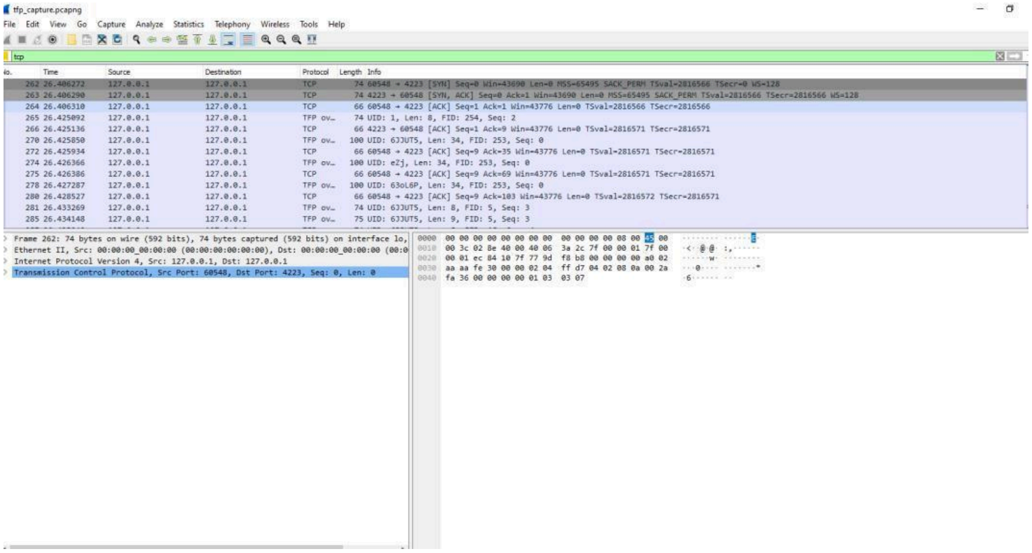


icmp

```
No.          Source       Destination   Protocol  Length Info
  2 0.000065 2.1.1.2      2.1.1.1       ICMP       466 Echo (ping) request  id=0x13c2, seq=1/256, ttl=64 (reply in 3)
  3 0.000509 2.1.1.1      2.1.1.2       ICMP      1442 Echo (ping) reply    id=0x13c2, seq=1/256, ttl=64 (request in 2)
```

```
> Frame 2: 466 bytes on wire (3728 bits), 466 bytes captured (3728 bits)
> Ethernet II, Src: PcsCompu_fc:6a:c9 (08:00:27:fc:6a:c9), Dst: PcsCompu_e2:9f:a6 (08:0
> Internet Protocol Version 4, Src: 2.1.1.2, Dst: 2.1.1.1
> Internet Control Message Protocol
```

```
0000  08 00 27 e2 9f a6 08 00  27 fc 6a c9 08 00 45 00   ··'·····'·j···E·
0010  01 c4 b5 d0 00 7a 40 01  bc ea 02 01 01 02 02 01   ·····z@·········
0020  01 01 c8 c9 ca cb cc cd  ce cf d0 d1 d2 d3 d4 d5   ················
0030  d6 d7 d8 d9 da db dc dd  de df e0 e1 e2 e3 e4 e5   ················
0040  e6 e7 e8 e9 ea eb ec ed  ee ef f0 f1 f2 f3 f4 f5   ················
0050  f6 f7 f8 f9 fa fb fc fd  fe ff 00 01 02 03 04 05   ················
0060  06 07 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   ················
0070  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ·········· !"#$%
0080  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0090  36 37 38 39 3a 3b 3c 3d  3e 3f 40 41 42 43 44 45   6789:;<= >?@ABCDE
00a0  46 47 48 49 4a 4b 4c 4d  4e 4f 50 51 52 53 54 55   FGHIJKLM NOPQRSTU
00b0  56 57 58 59 5a 5b 5c 5d  5e 5f 60 61 62 63 64 65   VWXYZ[\] ^_`abcde
00c0  66 67 68 69 6a 6b 6c 6d  6e 6f 70 71 72 73 74 75   fghijklm nopqrstu
00d0  76 77 78 79 7a 7b 7c 7d  7e 7f 80 81 82 83 84 85   vwxyz{|} ~·······
00e0  86 87 88 89 8a 8b 8c 8d  8e 8f 90 91 92 93 94 95   ················
00f0  96 97 98 99 9a 9b 9c 9d  9e 9f a0 a1 a2 a3 a4 a5   ················
0100  a6 a7 a8 a9 aa ab ac ad  ae af b0 b1 b2 b3 b4 b5   ················
0110  b6 b7 b8 b9 ba bb bc bd  be bf c0 c1 c2 c3 c4 c5   ················
0120  c6 c7 c8 c9 ca cb cc cd  ce cf d0 d1 d2 d3 d4 d5   ················
0130  d6 d7 d8 d9 da db dc dd  de df e0 e1 e2 e3 e4 e5   ················
0140  e6 e7 e8 e9 ea eb ec ed  ee ef f0 f1 f2 f3 f4 f5   ················
0150  f6 f7 f8 f9 fa fb fc fd  fe ff 00 01 02 03 04 05   ················
0160  06 07 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   ················
0170  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ·········· !"#$%
0180  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0190  36 37 38 39 3a 3b 3c 3d  3e 3f 40 41 42 43 44 45   6789:;<= >?@ABCDE
01a0  46 47 48 49 4a 4b 4c 4d  4e 4f 50 51 52 53 54 55   FGHIJKLM NOPQRSTU
01b0  56 57 58 59 5a 5b 5c 5d  5e 5f 60 61 62 63 64 65   VWXYZ[\] ^_`abcde
01c0  66 67 68 69 6a 6b 6c 6d  6e 6f 70 71 72 73 74 75   fghijklm nopqrstu
01d0  76 77                                              vw
```

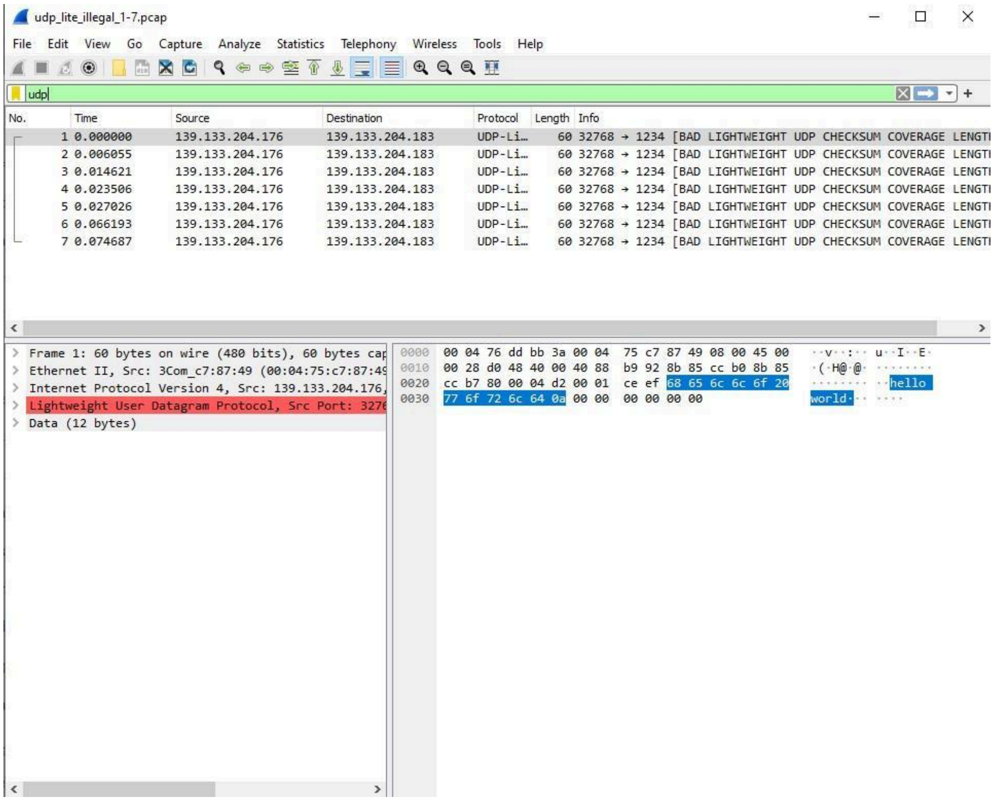


tfp_capture.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

```
No.  Time        Source      Destination  Protocol  Length Info
262 26.406272   127.0.0.1   127.0.0.1    TCP        74 60548 → 4223 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM TSval=2816566 TSecr=0 WS=128
263 26.406290   127.0.0.1   127.0.0.1    TCP        74 4223 → 60548 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_PERM TSval=2816566 TSecr=2816566 WS=128
264 26.406310   127.0.0.1   127.0.0.1    TCP        66 60548 → 4223 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=2816566 TSecr=2816566
265 26.425092   127.0.0.1   127.0.0.1    TFP ov…    74 UID: 1, Len: 8, FID: 254, Seq: 2
266 26.425136   127.0.0.1   127.0.0.1    TCP        66 4223 → 60548 [ACK] Seq=1 Ack=9 Win=43776 Len=0 TSval=2816571 TSecr=2816571
270 26.425850   127.0.0.1   127.0.0.1    TFP ov…   100 UID: 6JJUT5, Len: 34, FID: 253, Seq: 0
272 26.425934   127.0.0.1   127.0.0.1    TCP        66 60548 → 4223 [ACK] Seq=9 Ack=35 Win=43776 Len=0 TSval=2816571 TSecr=2816571
274 26.426366   127.0.0.1   127.0.0.1    TFP ov…   100 UID: eZj, Len: 34, FID: 253, Seq: 0
275 26.426386   127.0.0.1   127.0.0.1    TCP        66 60548 → 4223 [ACK] Seq=9 Ack=69 Win=43776 Len=0 TSval=2816571 TSecr=2816571
278 26.427287   127.0.0.1   127.0.0.1    TFP ov…   100 UID: 63oL6P, Len: 34, FID: 253, Seq: 0
280 26.428527   127.0.0.1   127.0.0.1    TCP        66 60548 → 4223 [ACK] Seq=9 Ack=103 Win=43776 Len=0 TSval=2816572 TSecr=2816571
281 26.433269   127.0.0.1   127.0.0.1    TFP ov…    74 UID: 6JJUT5, Len: 8, FID: 5, Seq: 3
285 26.434148   127.0.0.1   127.0.0.1    TFP ov…    75 UID: 6JJUT5, Len: 9, FID: 5, Seq: 3
```

```
> Frame 262: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo,
> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:0
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 60548, Dst Port: 4223, Seq: 0, Len: 0
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
0010  00 3c 02 8e 40 00 40 06  3a 2c 7f 00 00 01 7f 00   ·<··@·@· :,······
0020  00 01 ec 84 10 7f 77 9d  f8 b8 00 00 00 00 a0 02   ······w· ········
0030  aa aa fe 30 00 00 02 04  ff d7 04 02 08 0a 00 2a   ···0···· ·······*
0040  fa 36 00 00 00 00 01 03  03 07                     ·6······ ··
```



udp_lite_illegal_1-7.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp

```
No.  Time       Source            Destination       Protocol  Length Info
  1 0.000000   139.133.204.176   139.133.204.183   UDP-Li…     60 32768 → 1234 [BAD LIGHTWEIGHT UDP CHECKSUM COVERAGE LENGTH
  2 0.006055   139.133.204.176   139.133.204.183   UDP-Li…     60 32768 → 1234 [BAD LIGHTWEIGHT UDP CHECKSUM COVERAGE LENGTH
  3 0.014621   139.133.204.176   139.133.204.183   UDP-Li…     60 32768 → 1234 [BAD LIGHTWEIGHT UDP CHECKSUM COVERAGE LENGTH
  4 0.023506   139.133.204.176   139.133.204.183   UDP-Li…     60 32768 → 1234 [BAD LIGHTWEIGHT UDP CHECKSUM COVERAGE LENGTH
  5 0.027026   139.133.204.176   139.133.204.183   UDP-Li…     60 32768 → 1234 [BAD LIGHTWEIGHT UDP CHECKSUM COVERAGE LENGTH
  6 0.066193   139.133.204.176   139.133.204.183   UDP-Li…     60 32768 → 1234 [BAD LIGHTWEIGHT UDP CHECKSUM COVERAGE LENGTH
  7 0.074687   139.133.204.176   139.133.204.183   UDP-Li…     60 32768 → 1234 [BAD LIGHTWEIGHT UDP CHECKSUM COVERAGE LENGTH
```

```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes cap
> Ethernet II, Src: 3Com_c7:87:49 (00:04:75:c7:87:49
> Internet Protocol Version 4, Src: 139.133.204.176,
> Lightweight User Datagram Protocol, Src Port: 3276
> Data (12 bytes)
```

```
0000  00 04 76 dd bb 3a 00 04  75 c7 87 49 08 00 45 00   ··v··:·· u··I··E·
0010  00 28 d0 48 40 00 40 88  b9 92 8b 85 cc b0 8b 85   ·(·H@·@· ········
0020  cc b7 80 00 04 d2 00 01  ce ef 68 65 6c 6c 6f 20   ········ ··hello 
0030  77 6f 72 6c 64 0a 00 00  00 00 00 00               world··· ····
```

Wire shark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

Filtering Packets

If you‘re trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you‘ll likely have a large amount of packets to sift through. That‘s where Wire shark‘s filters come in.

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 (oppressing Enter).

For example, type—dns ¦and you‘ll see only DNS packets. When you start typing, Wire shark will help you auto complete your filter.

Conclusion:

For Faculty Use:

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				