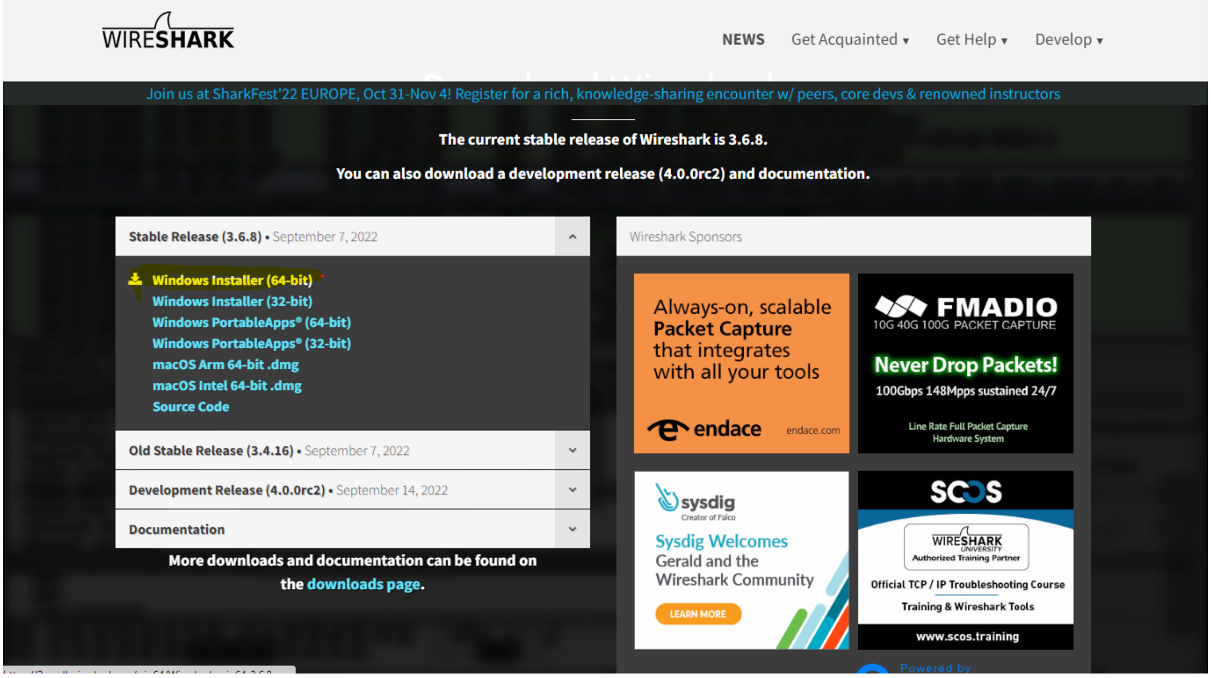
EXPERIMENT 8

AIM: Study of packets sniffer tools wireshark.

Steps:

1. Installation of wireshark in windows



Join us at SharkFest'22 EUROPE, Oct 31-Nov 4! Register for a rich, knowledge-sharing encounter w/ peers, core devs & renowned instructors

The current stable release of Wireshark is 3.6.8.

You can also download a development release (4.0.0rc2) and documentation.

Stable Release (3.6.8) • September 7, 2022

- Windows Installer (64-bit)
- Windows Installer (32-bit)
- Windows PortableApps® (64-bit)
- Windows PortableApps® (32-bit)
- macOS Arm 64-bit .dmg
- macOS Intel 64-bit .dmg
- Source Code

Old Stable Release (3.4.16) • September 7, 2022

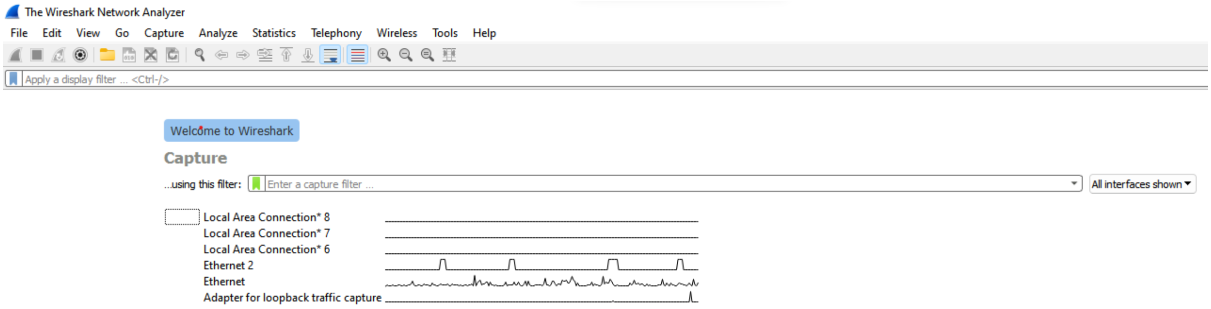
Development Release (4.0.0rc2) • September 14, 2022

Documentation

More downloads and documentation can be found on the downloads page.

Wireshark Sponsors
Always-on, scalable Packet Capture that integrates with all your tools
endace
endace.com
FMADIO
10G 40G 100G PACKET CAPTURE
Never Drop Packets!
100Gbps 148Mpps sustained 24/7
Line Rate Full Packet Capture Hardware System
sysdig
Creator of Falco
Sysdig Welcomes Gerald and the Wireshark Community
LEARN MORE
SCOS
WIRESHARK UNIVERSITY
Authorized Training Partner
Official TCP / IP Troubleshooting Course
Training & Wireshark Tools
www.scos.training
Powered by

2.After downloading and installing wireshark, 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 advance features by clicking capture options.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark

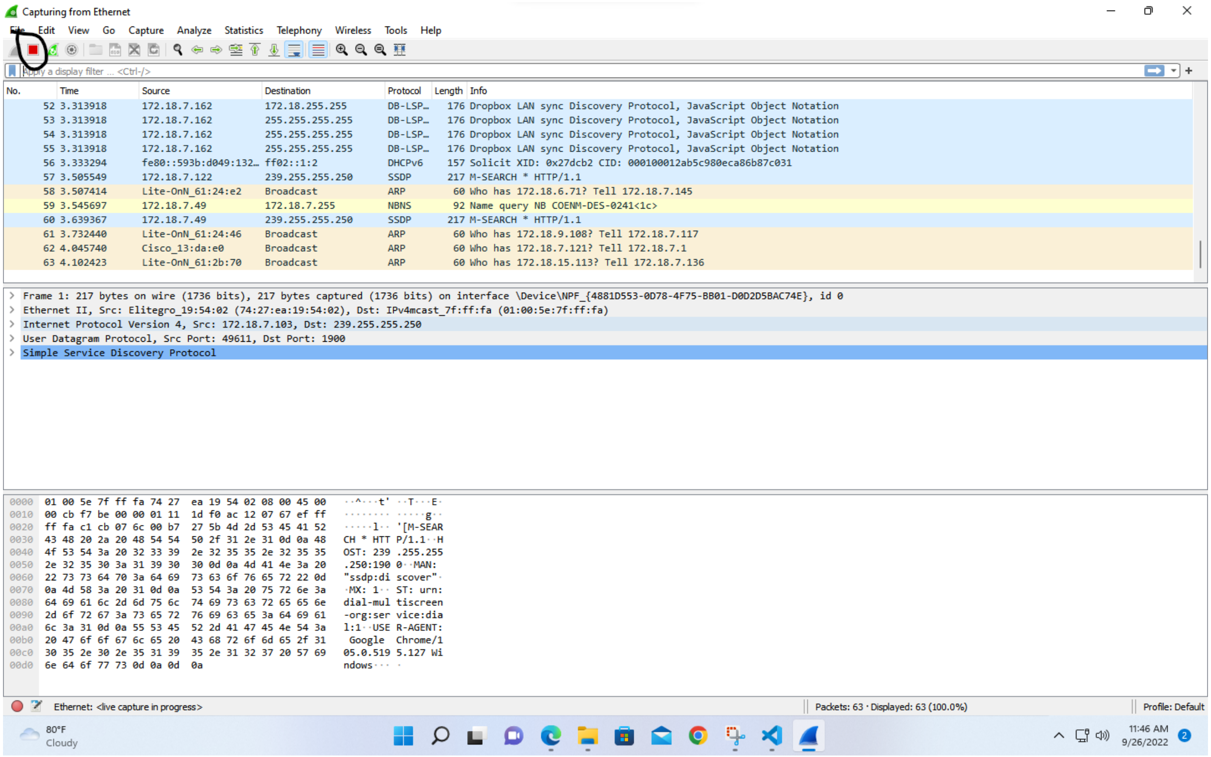
Capture

...using this filter: Enter a capture filter ... All interfaces shown

Local Area Connection* 8
Local Area Connection* 7
Local Area Connection* 6
Ethernet 2
Ethernet
Adapter for loopback traffic capture

3.As soon as you click the interface name, you’ll see the packets start to appear in real time. Wireshark captures each packets sent to or from your system. If you are capturing on a wireless interface and have promiscuous mode enabled in your capture options, you will also see other packets on the network.

4. Click the stop capture button near the top left corner of the window when you want to stop capturing traffic.


Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

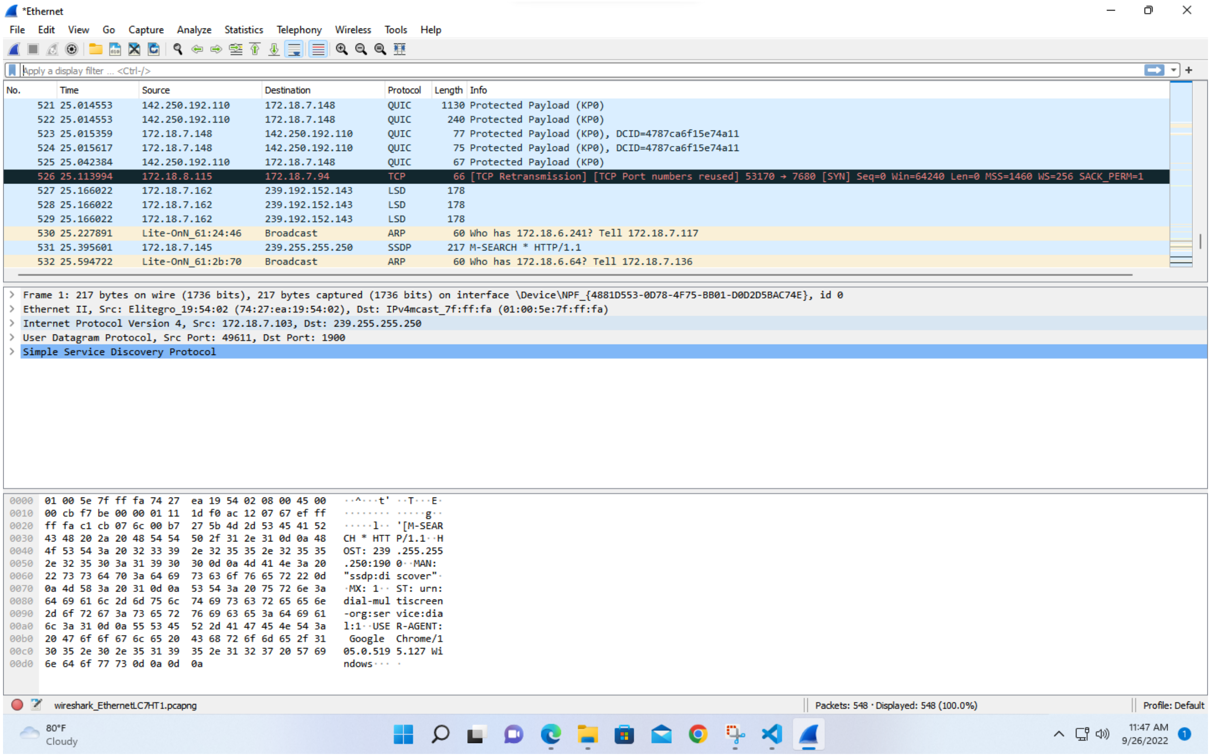
```
No.  Time      Source               Destination      Protocol Length Info
  52 3.313918  172.18.7.162         172.18.255.255   DB-LSP…     176 Dropbox LAN sync Discovery Protocol, JavaScript Object Notation
  53 3.313918  172.18.7.162         255.255.255.255  DB-LSP…     176 Dropbox LAN sync Discovery Protocol, JavaScript Object Notation
  54 3.313918  172.18.7.162         255.255.255.255  DB-LSP…     176 Dropbox LAN sync Discovery Protocol, JavaScript Object Notation
  55 3.313918  172.18.7.162         255.255.255.255  DB-LSP…     176 Dropbox LAN sync Discovery Protocol, JavaScript Object Notation
  56 3.333294  fe80::593b:d049:132… ff02::1:2        DHCPv6      157 Solicit XID: 0x27dcb2 CID: 000100012ab5c980eca86b87c031
  57 3.505549  172.18.7.122         239.255.255.250  SSDP        217 M-SEARCH * HTTP/1.1
  58 3.507414  Lite-OnN_61:24:e2    Broadcast        ARP          60 Who has 172.18.6.71? Tell 172.18.7.145
  59 3.545697  172.18.7.49          172.18.7.255     NBNS         92 Name query NB COENM-DES-0241<1c>
  60 3.639367  172.18.7.49          239.255.255.250  SSDP        217 M-SEARCH * HTTP/1.1
  61 3.732440  Lite-OnN_61:24:46    Broadcast        ARP          60 Who has 172.18.9.108? Tell 172.18.7.117
  62 4.045740  Cisco_13:da:e0       Broadcast        ARP          60 Who has 172.18.7.121? Tell 172.18.7.1
  63 4.102423  Lite-OnN_61:2b:70    Broadcast        ARP          60 Who has 172.18.15.113? Tell 172.18.7.136
```

```
> Frame 1: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF_{4881D553-0D78-4F75-BB01-D0D2D5BAC74E}, id 0
> Ethernet II, Src: Elitegro_19:54:02 (74:27:ea:19:54:02), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
> Internet Protocol Version 4, Src: 172.18.7.103, Dst: 239.255.255.250
> User Datagram Protocol, Src Port: 49611, Dst Port: 1900
> Simple Service Discovery Protocol
```

```
0000  01 00 5e 7f ff fa 74 27  ea 19 54 02 08 00 45 00   ··^···t' ··T···E·
0010  00 cb f7 be 00 00 01 11  1d f0 ac 12 07 67 ef ff   ········ ·····g··
0020  ff fa c1 cb 07 6c 00 b7  27 5b 4d 2d 53 45 41 52   ·····l·· '[M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1··H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:190 0··MAN: 
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:di scover"·
0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20 75 72 6e 3a   ·MX: 1·· ST: urn:
0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63 72 65 65 6e   dial-mul tiscreen
0090  2d 6f 72 67 3a 73 65 72  76 69 63 65 3a 64 69 61   -org:ser vice:dia
00a0  6c 3a 31 0d 0a 55 53 45  52 2d 41 47 45 4e 54 3a   l:1··USE R-AGENT:
00b0  20 47 6f 6f 67 6c 65 20  43 68 72 6f 6d 65 2f 31    Google  Chrome/1
00c0  30 35 2e 30 2e 35 31 39  35 2e 31 32 37 20 57 69   05.0.519 5.127 Wi
00d0  6e 64 6f 77 73 0d 0a 0d  0a                        ndows··· ·
```

Ethernet: <live capture in progress> Packets: 63 · Displayed: 63 (100.0%) Profile: Default

5. Wireshark uses color to help you identify the types of traffic at a glance. By default, green is the TCP traffic, dark blue is DNS traffic, light blue is UDP traffic and black identifies TCP packets with problems - for eg They could not have been delivered out-of-order.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time       Source             Destination        Protocol Length Info
  521 25.014553  142.250.192.110    172.18.7.148       QUIC     1130 Protected Payload (KP0)
  522 25.014553  142.250.192.110    172.18.7.148       QUIC      240 Protected Payload (KP0)
  523 25.015359  172.18.7.148       142.250.192.110    QUIC       77 Protected Payload (KP0), DCID=4787ca6f15e74a11
  524 25.015617  172.18.7.148       142.250.192.110    QUIC       75 Protected Payload (KP0), DCID=4787ca6f15e74a11
  525 25.042384  142.250.192.110    172.18.7.148       QUIC       67 Protected Payload (KP0)
  526 25.113994  172.18.8.115       172.18.7.94        TCP        66 [TCP Retransmission] [TCP Port numbers reused] 53170 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  527 25.166022  172.18.7.162       239.192.152.143    LSD       178
  528 25.166022  172.18.7.162       239.192.152.143    LSD       178
  529 25.166022  172.18.7.162       239.192.152.143    LSD       178
  530 25.227891  Lite-OnN_61:24:46  Broadcast          ARP        60 Who has 172.18.6.241? Tell 172.18.7.117
  531 25.395601  172.18.7.145       239.255.255.250    SSDP      217 M-SEARCH * HTTP/1.1
  532 25.594722  Lite-OnN_61:2b:70  Broadcast          ARP        60 Who has 172.18.6.64? Tell 172.18.7.136
```

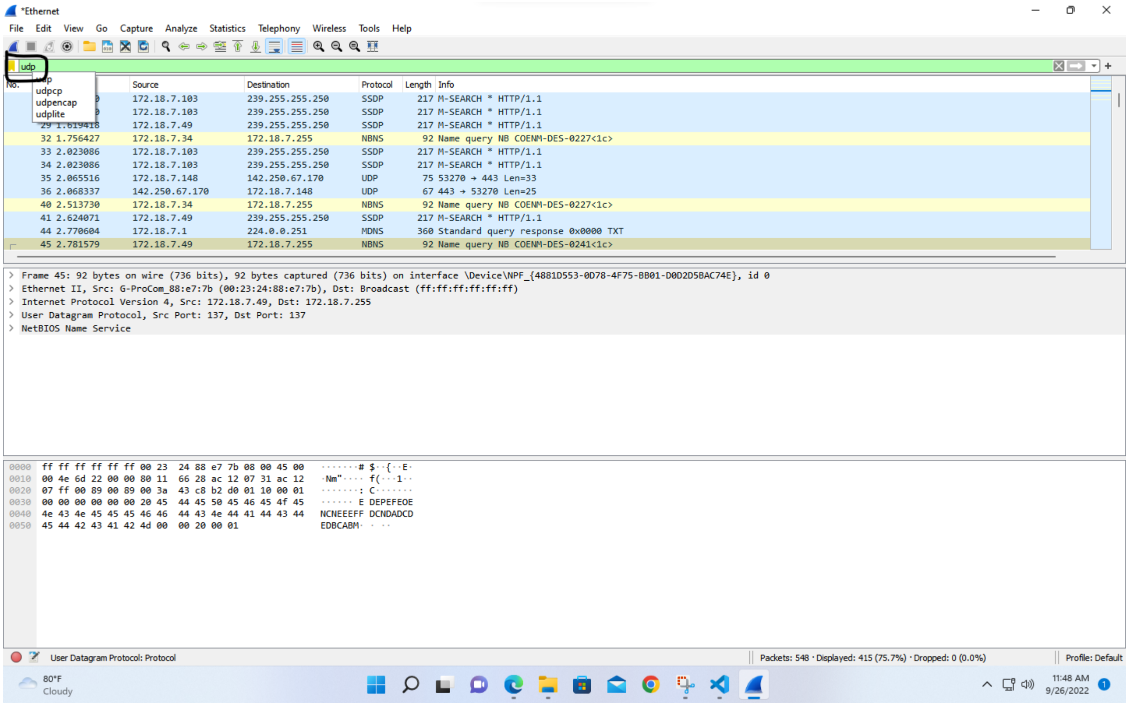
```
> Frame 1: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF_{4881D553-0D78-4F75-BB01-D0D2D5BAC74E}, id 0
> Ethernet II, Src: Elitegro_19:54:02 (74:27:ea:19:54:02), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
> Internet Protocol Version 4, Src: 172.18.7.103, Dst: 239.255.255.250
> User Datagram Protocol, Src Port: 49611, Dst Port: 1900
> Simple Service Discovery Protocol
```

```
0000  01 00 5e 7f ff fa 74 27  ea 19 54 02 08 00 45 00   ··^···t' ··T···E·
0010  00 cb f7 be 00 00 01 11  1d f0 ac 12 07 67 ef ff   ········ ·····g··
0020  ff fa c1 cb 07 6c 00 b7  27 5b 4d 2d 53 45 41 52   ·····l·· '[M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1··H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:190 0··MAN:
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:di scover"·
0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20 75 72 6e 3a   ·MX: 1·· ST: urn:
0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63 72 65 65 6e   dial-mul tiscreen
0090  2d 6f 72 67 3a 73 65 72  76 69 63 65 3a 64 69 61   -org:ser vice:dia
00a0  6c 3a 31 0d 0a 55 53 45  52 2d 41 47 45 4e 54 3a   l:1··USE R-AGENT:
00b0  20 47 6f 6f 67 6c 65 20  43 68 72 6f 6d 65 2f 31    Google  Chrome/1
00c0  30 35 2e 30 2e 35 31 39  35 2e 31 32 37 20 57 69   05.0.519 5.127 Wi
00d0  6e 64 6f 77 73 0d 0a 0d  0a                        ndows··· ·
```

wireshark_EthernetLC7HT1.pcapng
Packets: 548 · Displayed: 548 (100.0%)
Profile: Default
80°F
Cloudy
11:47 AM
9/26/2022

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 Wireshark'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 (or pressing Enter). For example, type -dnsl and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp
udp
udpcp
udpencap
udplite

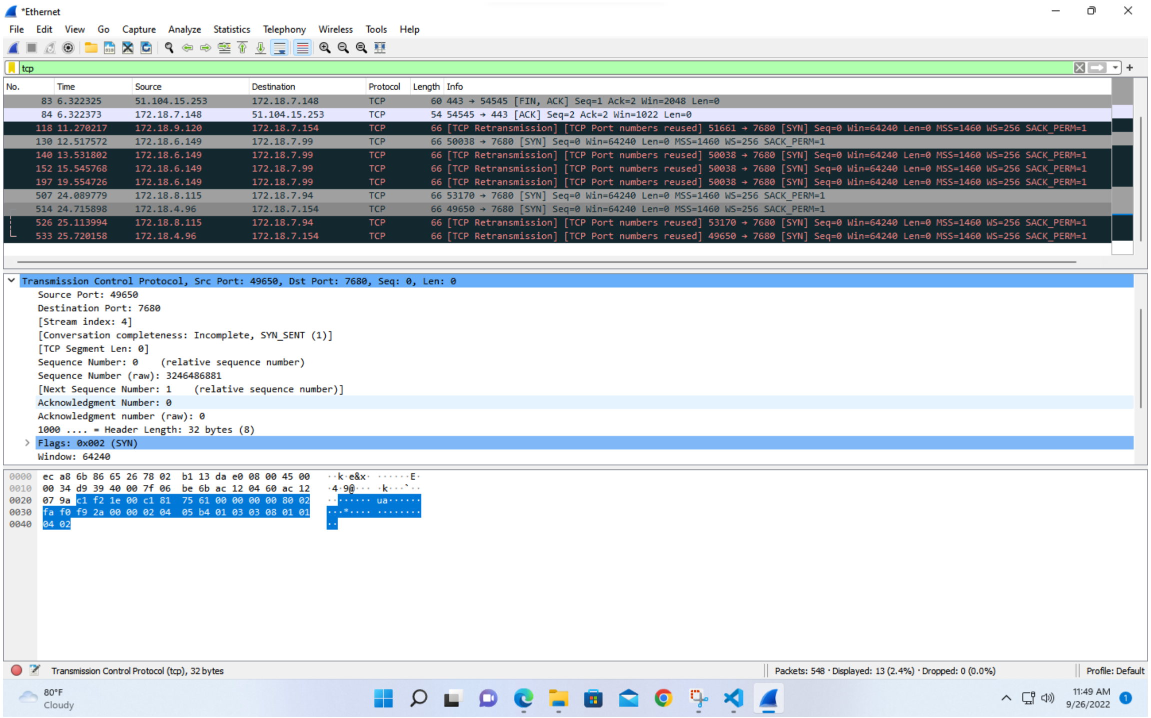
```
No.                  Source          Destination       Protocol Length Info
                  0  172.18.7.103    239.255.255.250   SSDP     217 M-SEARCH * HTTP/1.1
                  0  172.18.7.103    239.255.255.250   SSDP     217 M-SEARCH * HTTP/1.1
     29 1.619418     172.18.7.49     239.255.255.250   SSDP     217 M-SEARCH * HTTP/1.1
     32 1.756427     172.18.7.34     172.18.7.255      NBNS      92 Name query NB COENM-DES-0227<1c>
     33 2.023086     172.18.7.103    239.255.255.250   SSDP     217 M-SEARCH * HTTP/1.1
     34 2.023086     172.18.7.103    239.255.255.250   SSDP     217 M-SEARCH * HTTP/1.1
     35 2.065516     172.18.7.148    142.250.67.170    UDP       75 53270 → 443 Len=33
     36 2.068337     142.250.67.170  172.18.7.148      UDP       67 443 → 53270 Len=25
     40 2.513730     172.18.7.34     172.18.7.255      NBNS      92 Name query NB COENM-DES-0227<1c>
     41 2.624071     172.18.7.49     239.255.255.250   SSDP     217 M-SEARCH * HTTP/1.1
     44 2.770604     172.18.7.1      224.0.0.251       MDNS     360 Standard query response 0x0000 TXT
     45 2.781579     172.18.7.49     172.18.7.255      NBNS      92 Name query NB COENM-DES-0241<1c>
```

```
> Frame 45: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface \Device\NPF_{4881D553-0D78-4F75-BB01-D0D2D5BAC74E}, id 0
> Ethernet II, Src: G-ProCom_88:e7:7b (00:23:24:88:e7:7b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 172.18.7.49, Dst: 172.18.7.255
> User Datagram Protocol, Src Port: 137, Dst Port: 137
> NetBIOS Name Service
```

```
0000  ff ff ff ff ff ff 00 23  24 88 e7 7b 08 00 45 00   ·······# $··{··E·
0010  00 4e 6d 22 00 00 80 11  66 28 ac 12 07 31 ac 12   ·Nm"···· f(···1··
0020  07 ff 00 89 00 89 00 3a  43 c8 b2 d0 01 10 00 01   ·······: C·······
0030  00 00 00 00 00 00 20 45  44 45 50 45 46 45 4f 45   ······ E DEPEFEOE
0040  4e 43 4e 45 45 45 46 46  44 43 4e 44 41 44 43 44   NCNEEEFF DCNDADCD
0050  45 44 42 43 41 42 4d 00  00 20 00 01               EDBCABM· · ··
```

User Datagram Protocol: Protocol | Packets: 548 · Displayed: 415 (75.7%) · Dropped: 0 (0.0%) | Profile: Default

80°F
Cloudy
11:48 AM
9/26/2022


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

```
No.     Time         Source          Destination     Protocol Length Info
     83 6.322325     51.104.15.253   172.18.7.148    TCP       60 443 → 54545 [FIN, ACK] Seq=1 Ack=2 Win=2048 Len=0
     84 6.322373     172.18.7.148    51.104.15.253   TCP       54 54545 → 443 [ACK] Seq=2 Ack=2 Win=1022 Len=0
    118 11.270217    172.18.9.120    172.18.7.154    TCP       66 [TCP Retransmission] [TCP Port numbers reused] 51661 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    130 12.517572    172.18.6.149    172.18.7.99     TCP       66 50038 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    140 13.531802    172.18.6.149    172.18.7.99     TCP       66 [TCP Retransmission] [TCP Port numbers reused] 50038 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    152 15.545768    172.18.6.149    172.18.7.99     TCP       66 [TCP Retransmission] [TCP Port numbers reused] 50038 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    197 19.554726    172.18.6.149    172.18.7.99     TCP       66 [TCP Retransmission] [TCP Port numbers reused] 50038 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    507 24.089779    172.18.8.115    172.18.7.94     TCP       66 53170 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    514 24.715898    172.18.4.96     172.18.7.154    TCP       66 49650 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    526 25.113994    172.18.8.115    172.18.7.94     TCP       66 [TCP Retransmission] [TCP Port numbers reused] 53170 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    533 25.720158    172.18.4.96     172.18.7.154    TCP       66 [TCP Retransmission] [TCP Port numbers reused] 49650 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
```

```
v Transmission Control Protocol, Src Port: 49650, Dst Port: 7680, Seq: 0, Len: 0
    Source Port: 49650
    Destination Port: 7680
    [Stream index: 4]
    [Conversation completeness: Incomplete, SYN_SENT (1)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3246486881
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window: 64240
```

```
0000  ec a8 6b 86 65 26 78 02  b1 13 da e0 08 00 45 00   ··k·e&x· ······E·
0010  00 34 d9 39 40 00 7f 06  be 6b ac 12 04 60 ac 12   ·4·9@··· ·k···`··
0020  07 9a c1 f2 1e 00 c1 81  75 61 00 00 00 00 80 02   ········ ua······
0030  fa f0 f9 2a 00 00 02 04  05 b4 01 03 03 08 01 01   ···*···· ········
0040  04 02                                              ··
```

Transmission Control Protocol (tcp), 32 bytes | Packets: 548 · Displayed: 13 (2.4%) · Dropped: 0 (0.0%) | Profile: Default

Conclusion:

Packet sniffing method is used by network administrator to troubleshoot the problem by understand the logs generated by the tool used as packet sniffer. Thus, we have successfully installed sniffing tool such as wireshark to sniff the traffic ongoing in the network.