

Exp No:- 08

Aim:- Study of packet sniffer tools Wireshark.

- a) Observer performance in promiscuous as well as non-promiscuous mode.
- b) Show the packets can be traced based on different filters.

Hardware / Software required:-

Wireshark, Ethernet & tcpdump.

Theory:-

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format.

Wireshark includes filters, color-coding & other features that let you dig deep into network traffic and inspect individual packets.

Applications:-

- Network administrators use it to troubleshoot network problems.
 - Network Security engineers use it to examine security problems.
 - Developers use it to debug protocol implementations.
 - People use it to learn network protocol internals.
- beside these examples can be helpful in many other situations too.

Features:-

- Available for UNIX and Windows.
- Capture live packet data from a network interface.

- Open files containing packet data captured with tcpdump / WinDump, Wireshark, and a 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 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 advanced features by clicking Capture Options.

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'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.

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

Wireshark 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 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 & clicking Apply (or pressing Enter). For example, type "dns" & you'll see only DNS packets. When you start typing, Wireshark will help you automatically complete your filter.

Another interesting thing you can do is right-click a packet & select Follow TCP Stream. You'll see the full conversation between the client & the server.

Choose the window & you'll find a filter has been applied automatically — Wireshark is showing you the packets that make up the conversation.

Inspecting Packets:

Click a packet to select it and you can dig down to view its details.

You can also create filter from here — just right-click one of the details & use the Apply as filter submenu to create a filter based on it.

Wireshark is an extremely powerful tool, and this is just scratching the surface of what you can do with it. Professionals use it to debug network protocol implementations, examine security

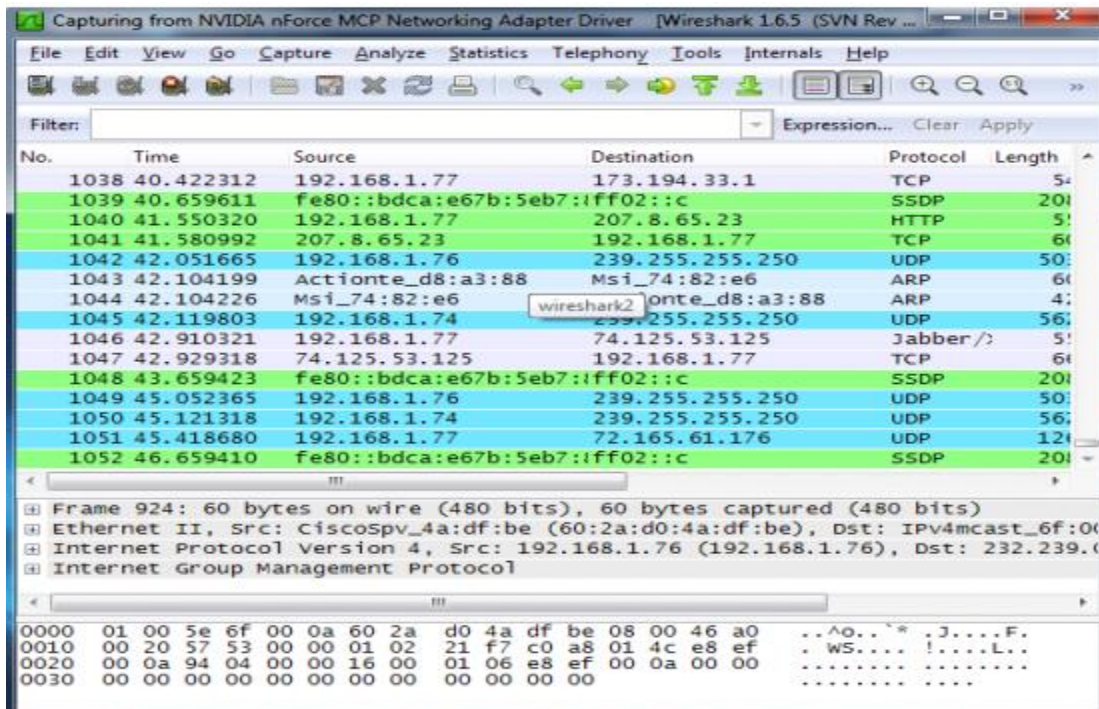
problems & inspect network protocol internal.

Conclusion:-

In this experiment, we analyze various packet sniffer tools that monitor network traffic transmitted between legitimate users or in the network. The packet sniffer is the network monitoring tool. It is used for network monitoring, traffic analysis, troubleshooting, packet capturing, message, protocol analysis, penetration testing and many other purposes.

Output:

Capturing Packets



No.	Time	Source	Destination	Protocol	Length
1038	40.422312	192.168.1.77	173.194.33.1	TCP	54
1039	40.659611	fe80::bdca:e67b:5eb7::c	207.8.65.23	SSDP	201
1040	41.550320	192.168.1.77	207.8.65.23	HTTP	51
1041	41.580992	207.8.65.23	192.168.1.77	TCP	60
1042	42.051665	192.168.1.76	239.255.255.250	UDP	50
1043	42.104199	Actionte_d8:a3:88	Msi_74:82:e6	ARP	60
1044	42.104226	Msi_74:82:e6	Actionte_d8:a3:88	ARP	41
1045	42.119803	192.168.1.74	239.255.255.250	UDP	56
1046	42.910321	192.168.1.77	74.125.53.125	Jabber/	51
1047	42.929318	74.125.53.125	192.168.1.77	TCP	60
1048	43.659423	fe80::bdca:e67b:5eb7::c	207.8.65.23	SSDP	201
1049	45.052365	192.168.1.76	239.255.255.250	UDP	50
1050	45.121318	192.168.1.74	239.255.255.250	UDP	56
1051	45.418680	192.168.1.77	72.165.61.176	UDP	121
1052	46.659410	fe80::bdca:e67b:5eb7::c	207.8.65.23	SSDP	201

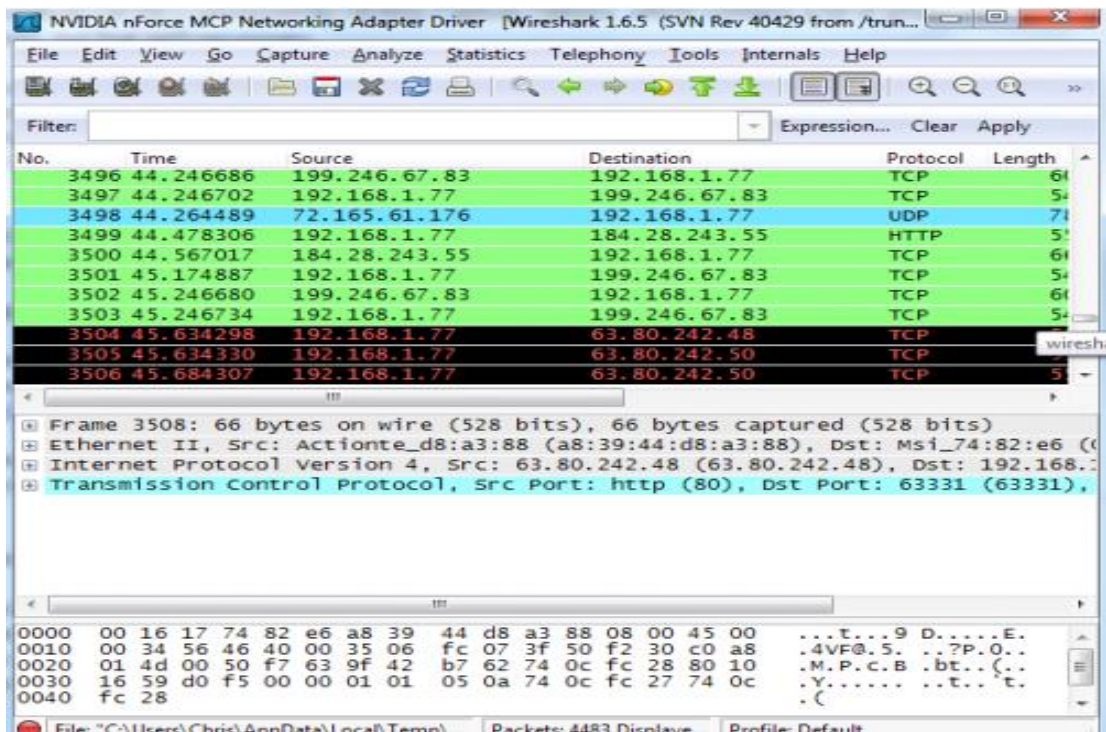
Frame 924: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)

Ethernet II, Src: CiscoSpv_4a:df:be (60:2a:d0:4a:df:be), Dst: IPv4mcast_6f:00:00:00:00:00 (01:00:5e:00:00:00)

Internet Protocol Version 4, Src: 192.168.1.76 (192.168.1.76), Dst: 232.239.252.251 (01:00:5e:00:00:00)

Internet Group Management Protocol

0000 01 00 5e 6f 00 0a 60 2a d0 4a df be 08 00 46 a0 ..^o..* .J....F.
0010 00 20 57 53 00 00 01 02 21 f7 c0 a8 01 4c e8 ef ..WS....!....L..
0020 00 0a 94 04 00 00 16 00 01 06 e8 ef 00 0a 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00



No.	Time	Source	Destination	Protocol	Length
3496	44.246686	199.246.67.83	192.168.1.77	TCP	60
3497	44.246702	192.168.1.77	199.246.67.83	TCP	54
3498	44.264489	72.165.61.176	192.168.1.77	UDP	71
3499	44.478306	192.168.1.77	184.28.243.55	HTTP	51
3500	44.567017	184.28.243.55	192.168.1.77	TCP	60
3501	45.174887	192.168.1.77	199.246.67.83	TCP	54
3502	45.246680	199.246.67.83	192.168.1.77	TCP	60
3503	45.246734	192.168.1.77	199.246.67.83	TCP	54
3504	45.634298	192.168.1.77	63.80.242.48	TCP	54
3505	45.634330	192.168.1.77	63.80.242.50	TCP	54
3506	45.684307	192.168.1.77	63.80.242.50	TCP	54

Frame 3508: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

Ethernet II, Src: Actionte_d8:a3:88 (a8:39:44:d8:a3:88), Dst: Msi_74:82:e6 (08:00:27:74:0c:fc)

Internet Protocol Version 4, Src: 63.80.242.48 (63.80.242.48), Dst: 192.168.1.77 (01:00:5e:00:00:00)

Transmission Control Protocol, Src Port: http (80), Dst Port: 63331 (63331), Seq: 3456789012, Win: 65535, Len: 0

0000 00 16 17 74 82 e6 a8 39 44 d8 a3 88 08 00 45 00 ...t...9 D....E.
0010 00 34 56 46 40 00 35 06 fc 07 3f 50 f2 30 c0 a8 ..4VF@.5. .?P.O..
0020 01 4d 00 50 f7 63 9f 42 b7 62 74 0c fc 28 80 10 ..M.P.c.B .bt..(
0030 16 59 d0 f5 00 00 01 01 05 0a 74 0c fc 27 74 0c ..Y..... .t..t..
0040 fc 28

Filtering Packets

The screenshot shows the Wireshark interface with a packet capture of DNS traffic. The packet list on the left shows several DNS queries and responses. The packet details pane on the right shows the structure of a DNS query packet, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Domain Name System (query).

No.	Time	Source	Destination	Protocol	Length	Info
30	8.516189	89.101.160.5	192.168.0.220	DNS	191	Standard query response 0x7eaf A ogs.google.com CNAME www3.l.google.com A 209.85.202.101 A 209.85.202.113 A...
31	8.516190	89.101.160.5	192.168.0.220	DNS	90	Standard query response 0x6267 A www.google.com A 216.58.208.228
34	8.521575	89.101.160.4	192.168.0.220	DNS	89	Standard query response 0xf619 A www.google.ie A 216.58.208.195
35	8.528190	89.101.160.4	192.168.0.220	DNS	192	Standard query response 0xe9e7 A apis.google.com CNAME plus.l.google.com A 209.85.202.100 A 209.85.202.139 A...
36	8.532226	89.101.160.5	192.168.0.220	DNS	107	Standard query response 0x36df A www.gstatic.com A 209.85.202.94 A 209.85.202.120
279	10.453605	192.168.0.220	89.101.160.5	DNS	77	Standard query 0xe66f A www.cctmoodle.com
280	10.466955	89.101.160.5	192.168.0.220	DNS	107	Standard query response 0xe66f A www.cctmoodle.com CNAME cctmoodle.com A 52.50.203.224
316	11.293038	192.168.0.220	89.101.160.5	DNS	79	Standard query 0x2828 A clients4.google.com
317	11.306645	89.101.160.5	192.168.0.220	DNS	199	Standard query response 0x2828 A clients4.google.com CNAME clients.l.google.com A 209.85.202.100 A 209.85.2...
330	11.364563	192.168.0.220	89.101.160.5	DNS	84	Standard query 0x8c9a A www.google-analytics.com
332	11.378914	89.101.160.5	192.168.0.220	DNS	224	Standard query response 0x8c9a A www.google-analytics.com CNAME www.google-analytics.l.google.com A 209.85...
388	11.563762	192.168.0.220	89.101.160.5	DNS	82	Standard query 0x9fbc A cctsm2016.appspot.com
389	11.564521	192.168.0.220	89.101.160.5	DNS	70	Standard query 0x8c67 A www.cct.ie
390	11.564521	192.168.0.220	89.101.160.5	DNS	70	Standard query 0xb488 A sarcloth.ie
391	11.576765	89.101.160.5	192.168.0.220	DNS	129	Standard query response 0x9fbc A cctsm2016.appspot.com CNAME appspot.l.google.com A 209.85.202.141

Frame 13: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
 Ethernet II, Src: IntelCor_e8:bf:2c (a0:88:b4:e8:bf:2c), Dst: CompalBr_c9:87:4d (54:67:51:c9:87:4d)
 Internet Protocol Version 4, Src: 192.168.0.220, Dst: 89.101.160.5
 User Datagram Protocol, Src Port: 57092, Dst Port: 53
 Domain Name System (query)

0000 54 67 51 c9 87 4d a0 88 b4 e8 bf 2c 00 00 45 00 TgQ..M..E.
 0010 00 3b 01 74 00 00 00 11 7e 4f c0 a8 00 dc 59 65 .:t... +0....Ye
 0020 a0 05 df 04 00 35 00 27 45 c3 f6 19 01 00 00 015.' E.....
 0030 00 00 00 00 00 00 03 77 77 77 06 67 6f 6f 67 6cw ww.googl
 0040 65 02 69 65 00 00 01 00 01 e.ie.....

The screenshot shows the 'Follow TCP Stream' window in Wireshark, displaying the content of a specific TCP stream. The stream content is shown in a text area, and the bottom of the window has buttons for 'Find', 'Save As', 'Print', and radio buttons for 'ASCII', 'EBCDIC', 'Hex Dump', 'C Arrays', and 'Raw'.

Stream Content

Accept-Encoding: gzip, deflate
 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
 Connection: keep-alive
 Referer: http://ca.linkedin.com/pub/geds-dead/21/192/326

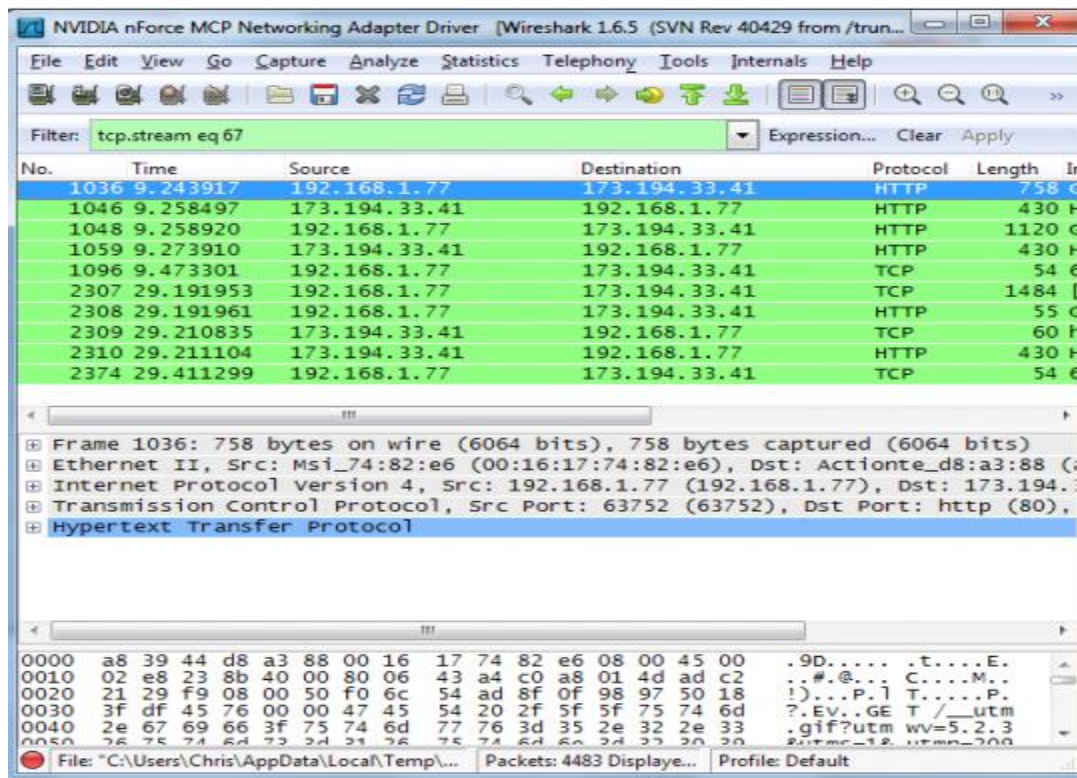
HTTP/1.1 200 OK
 Date: Thu, 26 Jan 2012 03:03:47 GMT
 Content-Length: 35
 X-Content-Type-Options: nosniff
 Pragma: no-cache
 Expires: wed, 19 Apr 2000 11:43:00 GMT
 Last-Modified: wed, 21 Jan 2004 19:51:30 GMT
 Content-Type: image/gif
 Cache-Control: private, no-cache, no-cache=Set-Cookie, proxy-revalidate
 Age: 210285
 Server: GFE/2.0

GIF89a.....D.;GET /__utm.gif?
 utmmv=5.2.3&utms=1&utmn=637238821&utmhn=www.theglobeandmail.com
 &utmcs=UTF-8&utmsr=1280x1024&utmvp=1263x893&utmsc=24-
 bit&utmilen-us&utmie=1&utmf1=11 1%20c102&utmdr=ululemon%20ad%

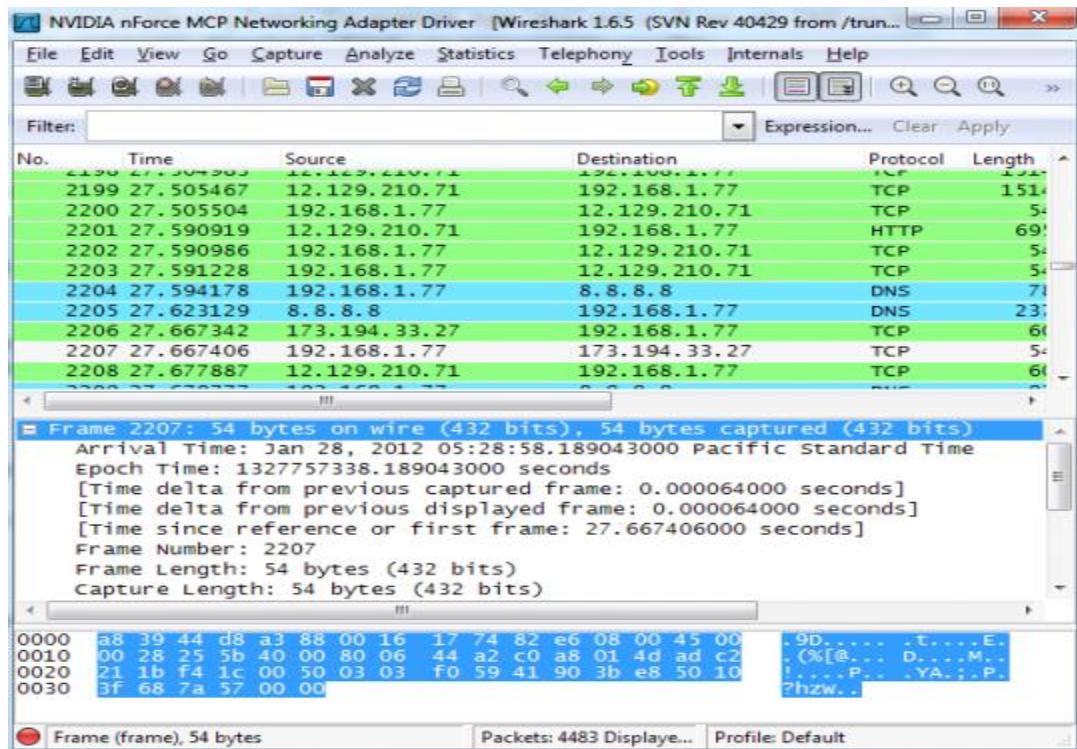
Entire conversation (4329 bytes)

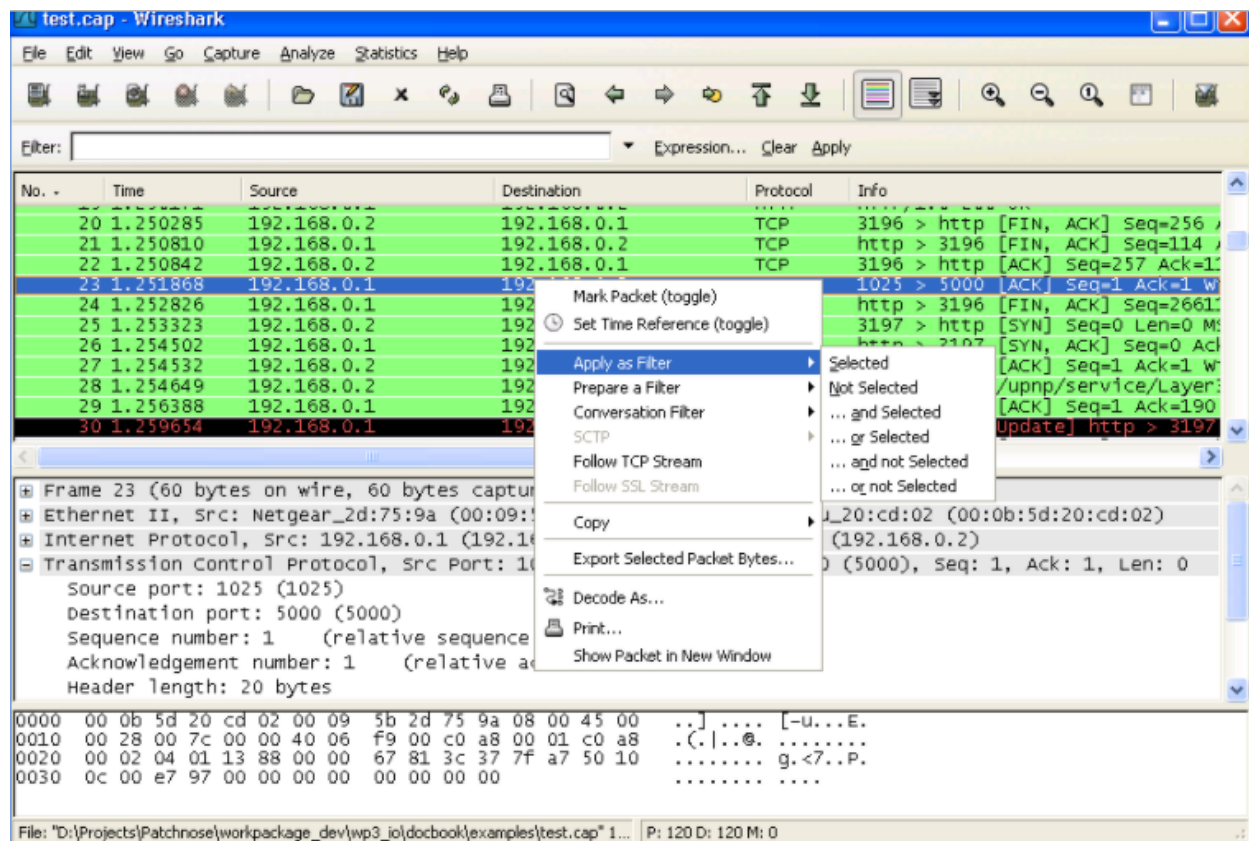
Find Save As Print ASCII EBCDIC Hex Dump C Arrays **Raw**

Help Filter Out This Stream Close



Inspecting Packets :





Conclusion:

In this experiment we analyze various packet sniffing tools that monitor network traffic transmitted between legitimate users or in the network. The packet sniffer is network monitoring tool. It is opted for network monitoring, traffic analysis, troubleshooting, Packet grapping, message, protocol analysis, penetration testing and many other purposes.