**Dr.MAHALINGAM COLLEGE OF ENGINEERING & TECHNOLOGY**

**(An Autonomous Institution)**

**POLLACHI-642003**

**16ITL51 - Computer Networks Laboratory**

**Observation cum Record**

**Faculty Incharges**

**Dr.S.Ramakrishnan**

**Mr.J.Ramprasath**



**Department of Information Technology**

**Academic Year 2020-2021**

|  |  |
| --- | --- |
| C:\Users\student.DMLAB-44\Desktop\image.png | **Dr.MAHALINGAM COLLEGE OF ENGINEERING & TECHNOLOGY** |
| **(An Autonomous Institution)** |
| **POLLACHI-642003** |

**16ITL51 - Computer Networks Laboratory**

Name of the student : NADARAJAN S

Class & Section : III IT - A

Roll No. : 18BIT003

Certified that this is bonafied record of work done by the above student of the B.Tech – Information Technology during the year 2020-2021

Head of the Department Staff In-Charge

Submitted for the End Semester Practical examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner - 1 Internal Examiner - 2

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Experiment Name** | **Page No.** | **Marks** | **Signature** |
|  | Network trouble-shooting and performance monitoring using ipconfig, ping, netstat commands |  |  |  |
|  | Visualization of packet flow using Wireshark |  |  |  |
|  | Interpret the working principles of address resolution protocol using Wireshark |  |  |  |
|  | Examine IP traffic and its routing options using Wireshark |  |  |  |
|  | Classify the internet control message protocol messages using Wireshark |  |  |  |
|  | Analyze the TCP connection establishment and termination using Wireshark |  |  |  |
|  | Implementation of socket programming using TCP sockets |  |  |  |
|  | Configure the LAN for generating data traffic using GNS3 |  |  |  |
|  | Configure a topology and analyze traffic using riverbed |  |  |  |
|  | Experiment with NS2 to configure LAN for generating data traffic |  |  |  |
| **Average** | | |  |  |

|  |  |
| --- | --- |
| **Ex. No. : 1** | **Network trouble-shooting and performance monitoring using ipconfig, ping, netstat commands** |
| **Date :** |

**AIM:**

To troubleshoot and monitor the network performances using networks utility commands such as ipconfig, ping, netstat.

**Procedure:**

**Ipconfig:**

ipconfig (internet protocol configuration) in Microsoft Windows is a console application that displays all current TCP/IP network configuration values.

1. Open a Windows terminal by clicking **Start -> Run**. Type **cmd**, and click **OK**. Issue the **ipconfig-?** Command on the host computer, and examine the output.

|  |  |  |
| --- | --- | --- |
| S.No. | Command Name | Usage |
|  | ipconfig |  |
|  | Ipconfig /all |  |
|  | ipconfig /release |  |

1. At the command prompt, type ipconfig to view your system TCP/IP setting. Record your finding in the space provided.
   1. Write down your IP address:
   2. Write down your Subnet Mask:
   3. Write down your Default Gateway:
2. To see more info, type ipconfig /all and press ENTER. The figure shows the detailed IP configuration screen.
   1. Do all of the servers share the same network portion of the IP address as your workstation?
   2. Find the Physical Address and the NIC card model description of your system.
   3. Find the physical layer adapters associated with your system.
   4. Find the host name of your system.
   5. Find the Link local IPv6 Address of your system.
   6. Find the DHCP Enable status for Ethernet adapter.
   7. Write a command to destroy entries in DNS cache of your system.
   8. Write a command to display DNS cache entries of your system.

**Ping**:

Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network.

1. Open a Windows terminal by clicking **Start -> Run**. Type **cmd**, and click **OK**. Issue the **ping-?** Command on the host computer, and examine the output.

|  |  |  |
| --- | --- | --- |
| S.No. | Command Name | Usage |
|  | ping –t |  |
|  | ping –a |  |
|  | ping –n count |  |
|  | ping –l size |  |
|  | ping –f |  |
|  | ping –I TTL |  |
|  | ping –v TOS |  |
|  | ping –r count |  |
|  | ping –s count |  |
|  | ping –j host list |  |
|  | ping –k host list |  |
|  | ping –w timeout |  |
|  | ping –R |  |
|  | ping –S srcaddr |  |
|  | ping -4 |  |
|  | ping -6 |  |

1. Ping the loopback address to verify that TCP/IP is installed and configured correctly on the local computer.
2. Ping the IP address of the default gateway to verify that the default gateway is functioning and that you can communicate with a local host on the local network.
3. Ping a remote host to verify that you can communicate through a router.

Example: ping yahoo.co.in

1. How to increase the ping echo request to 8?
2. Set buffer size to 1500 and perform continues pining to 118.91.233.179. Write down the ping statistics.
3. Ping to neighbor node using conditions such as Set buffer size to 1500 and enable don’t fragment flag. And justify reason for not generating echo reply from target machine.
4. What would be the result of a loss of connectivity to the default gateway?
5. Determine MTU of your system using ping command.

**Netstat:**

Netstat (network statistics) is a command-line network utility tool that displays network connections for the Transmission Control Protocol, routing tables, and a number of network interface and network protocol statistics.

1. Open a Windows terminal by clicking **Start -> Run**. Type **cmd**, and click **OK**. Issue the **netstat-?** Command on the host computer, and examine the output.

|  |  |  |
| --- | --- | --- |
| **S.No** | **Command Name** | **usage** |
|  | netstat –a |  |
|  | netstat –b |  |
|  | netstat-e |  |
|  | netstat –f |  |
|  | netstat –n |  |
|  | netstat –o |  |
|  | netstat –p proto |  |
|  | netstat –r |  |
|  | netstat –s |  |
|  | netstat –t |  |

1. What command is to identify all active connections associated with your system? And document two entries for each connection status.
2. How to display Ethernet statistics of your system?
3. What command is used to display foreign address in the form of fully qualified domain names?
4. How to display port number in numerical form?
5. How to display protocol based statistics and list protocols though which data are transceivers?
6. How to display active connection under TCP protocol?
7. What command would be used to display all entries in routing table of your system?
8. How will you identify number of Unicast, & Non - Unicast packets send across your host Ethernet?
9. How can I determine whether TCP/IP is installed correctly on a Windows system?

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 2** | **Visualization of packet flow using Wireshark** |
| **Date : 05/11/2020** |

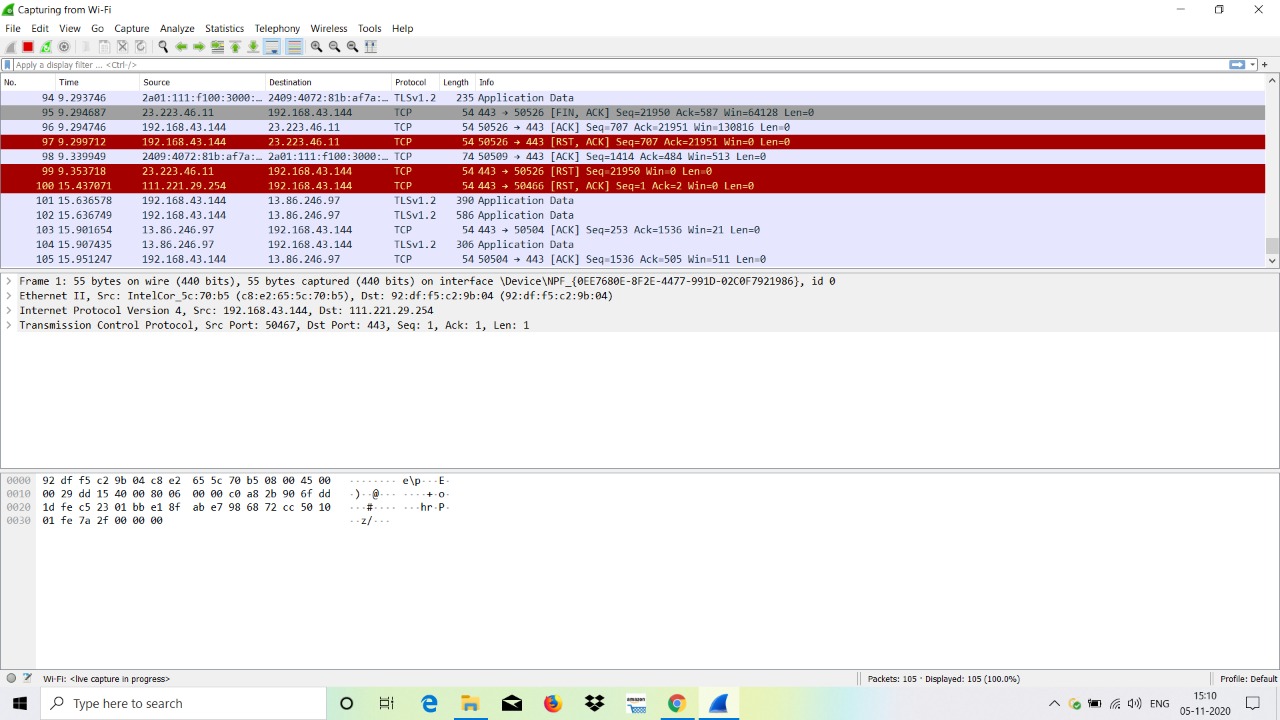
**Aim:**

To visualization of data flow and filter the traffic based on protocols and IPs using Wireshark

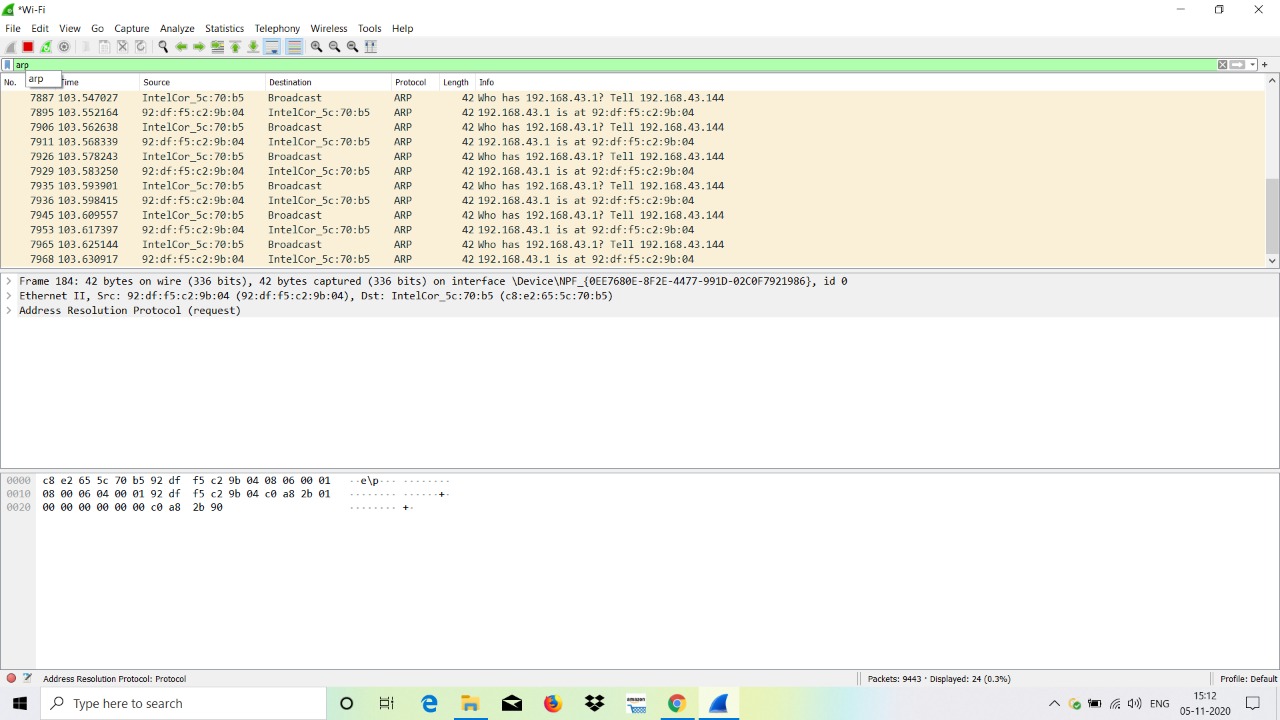
**Procedure:**

**Capturing Traffic using Wireshark:**

1. Capture traffic using following steps.
   1. Open a Wireshark
   2. You can start capture the Ethernet traffic by choosing *Capture*->*Interfaces*->*Ethernet (or)Wireless Network Connection.*
   3. View the traces in Wireshark packet list window. If a trace is unavailable in your Wireshark packet list window means then you can generate traffic using ping command by pining to your adjacent system. Illustrate the traffic generated by your system.



1. Examine the traffic generated by your system. By default traces contains following details - S.No, Time, Source & Destination IP address, Protocol, Length, Information. Expand any one trace by double clicking the trace. Illustrate the expended trace.



1. Match the expended trace with TCP/IP layering.

**Frame - Network**

**Ethernet II – Data Link**

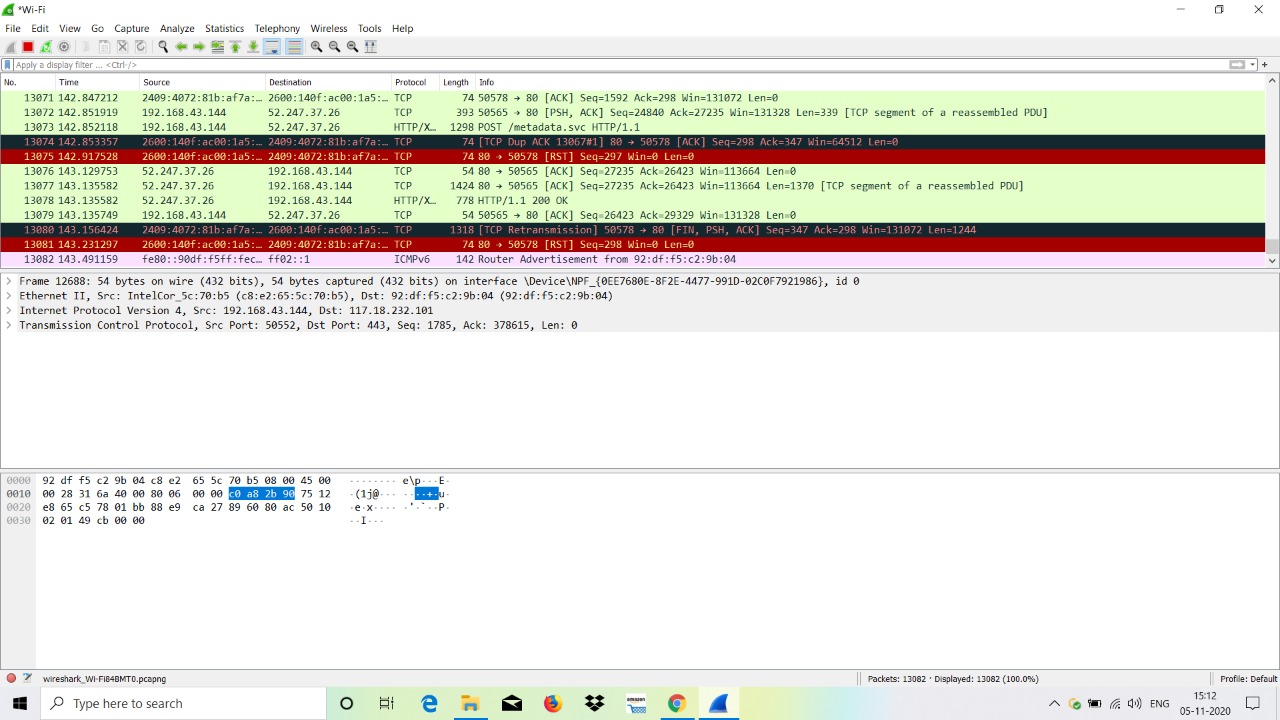
**Address Resolution Protocol - Physical**

1. Examine any one of traces in 1(c) by expanding its protocols and find the size of all the protocols associated with traces 1(c).

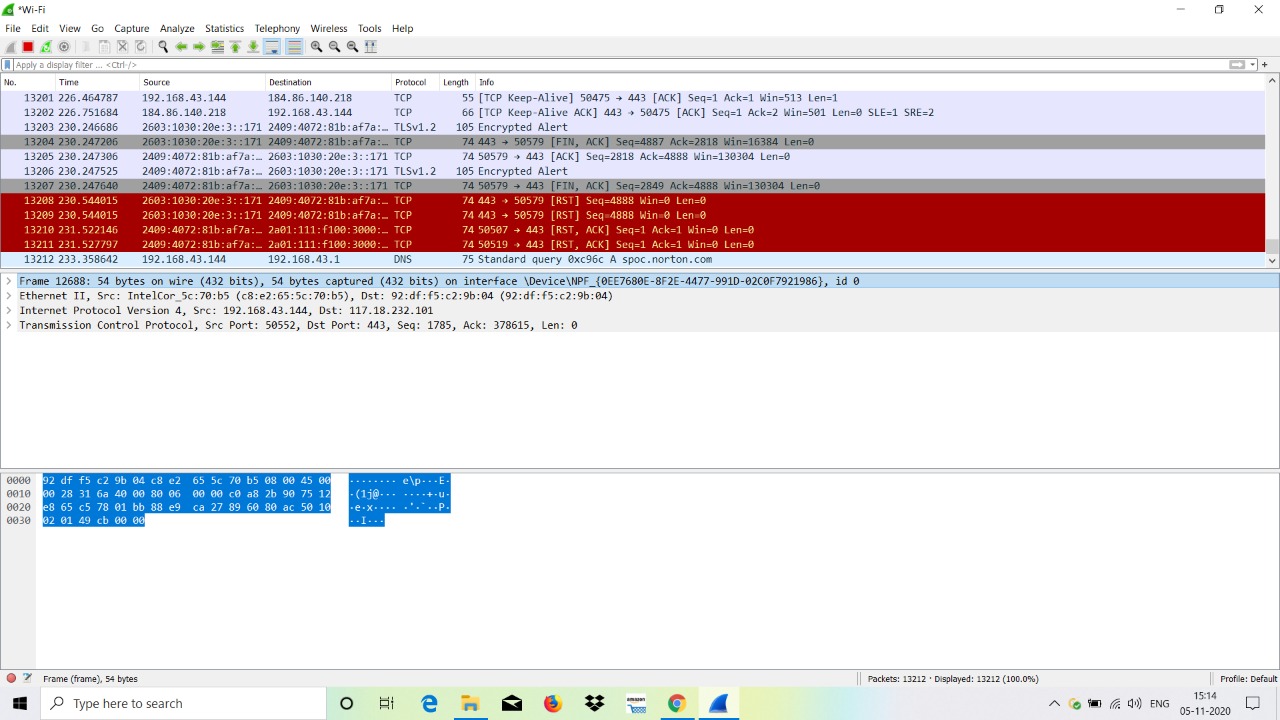
**Frame - 42 bytes**

**Ethernet II - 14 bytes**

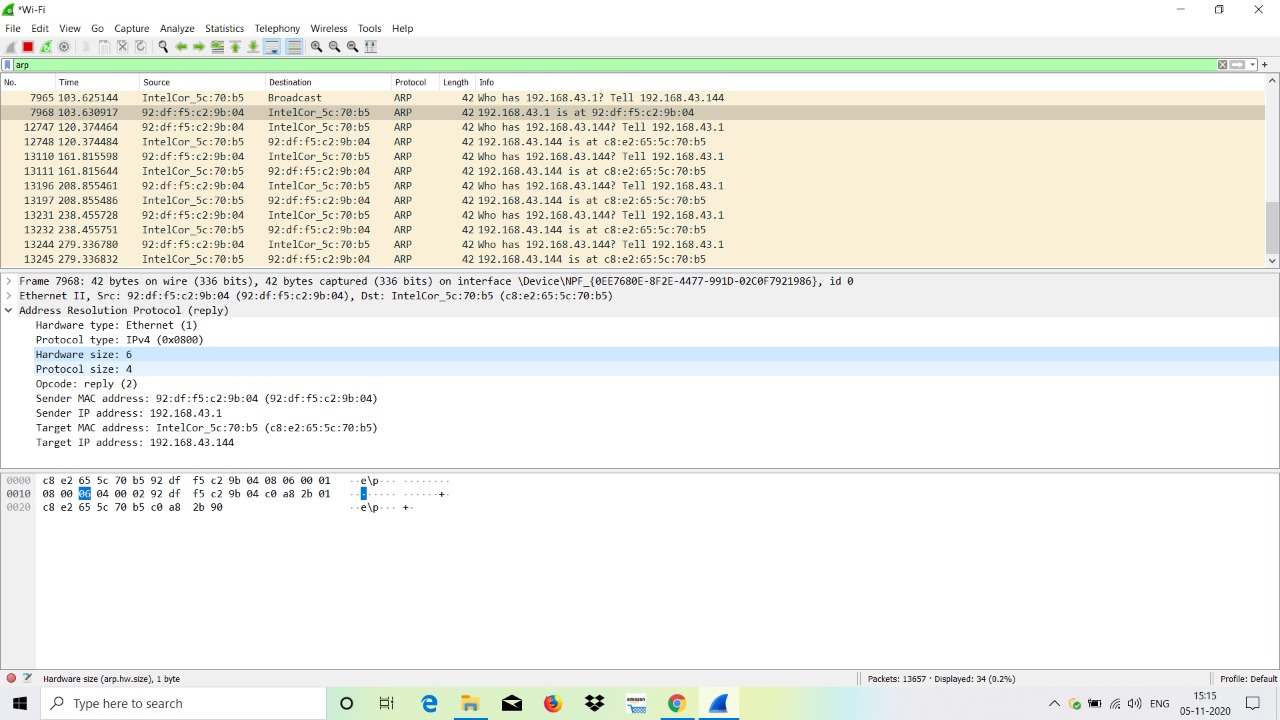
**ARP – 28 bytes**



1. Examine how traces are converted into hexadecimal format. Find the hexadecimal format of traces 2 by click over all the protocol associated with traces 2. View the hexadecimal format of trace in Wireshark packet bytes window.



1. Find the fields and its size of the protocols associated with traces 2.



**Hardware type – 2 bytes**

**Protocol type – 2 bytes**

**Hardware size – 1 byte**

**Protocol size – 1 byte**

**Opcode – 2 bytes**

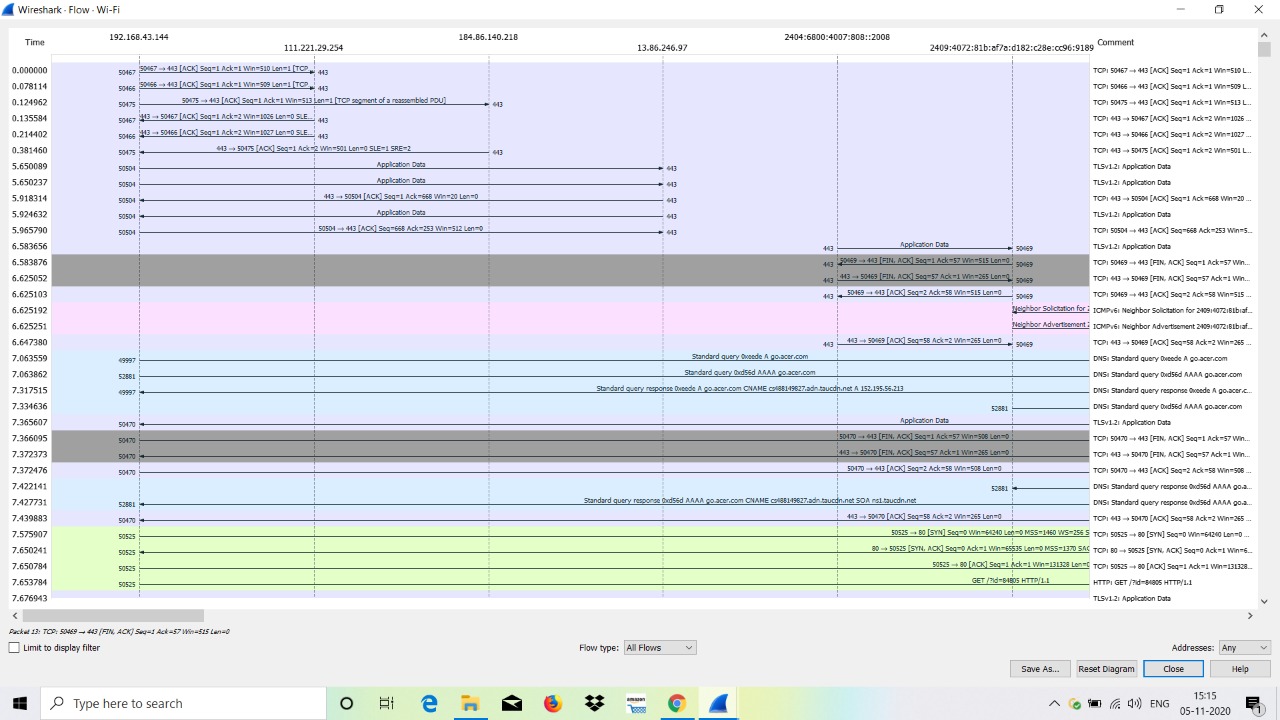
**Sender MAC Address – 6 bytes**

**Sender IP Address – 4 bytes**

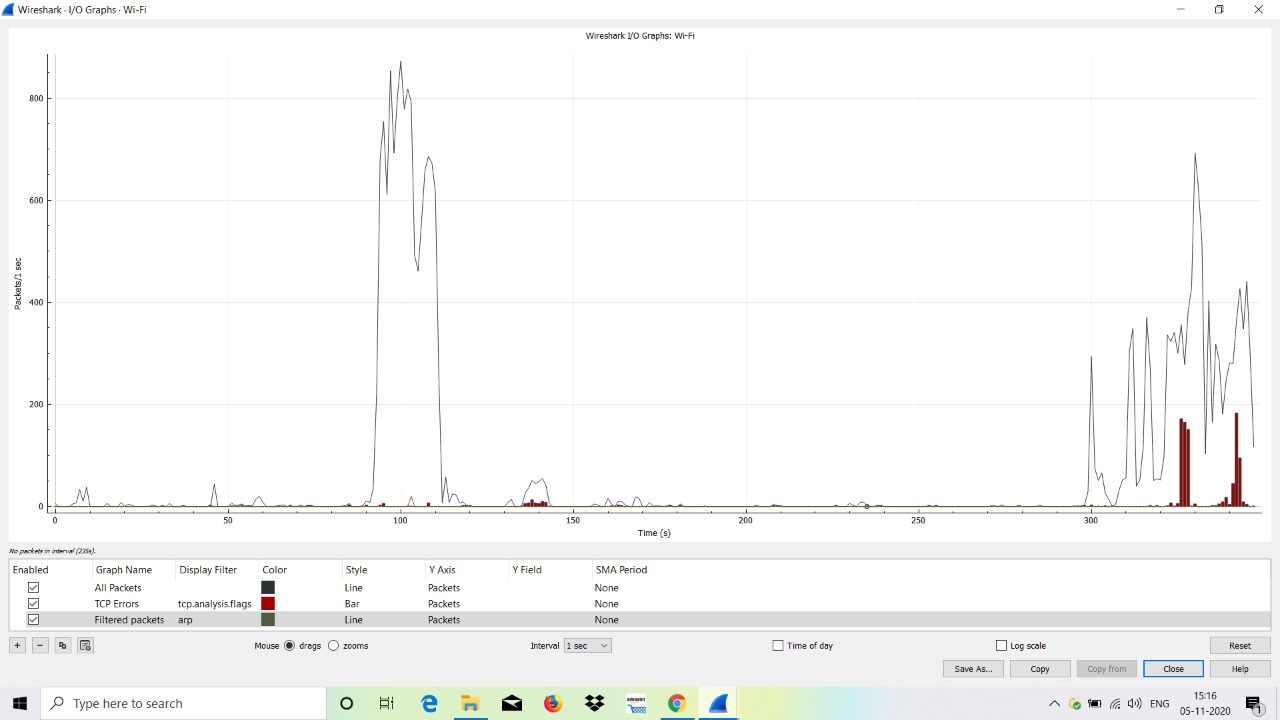
**Target MAC Address – 6 bytes**

**Target IP Address – 4 bytes**

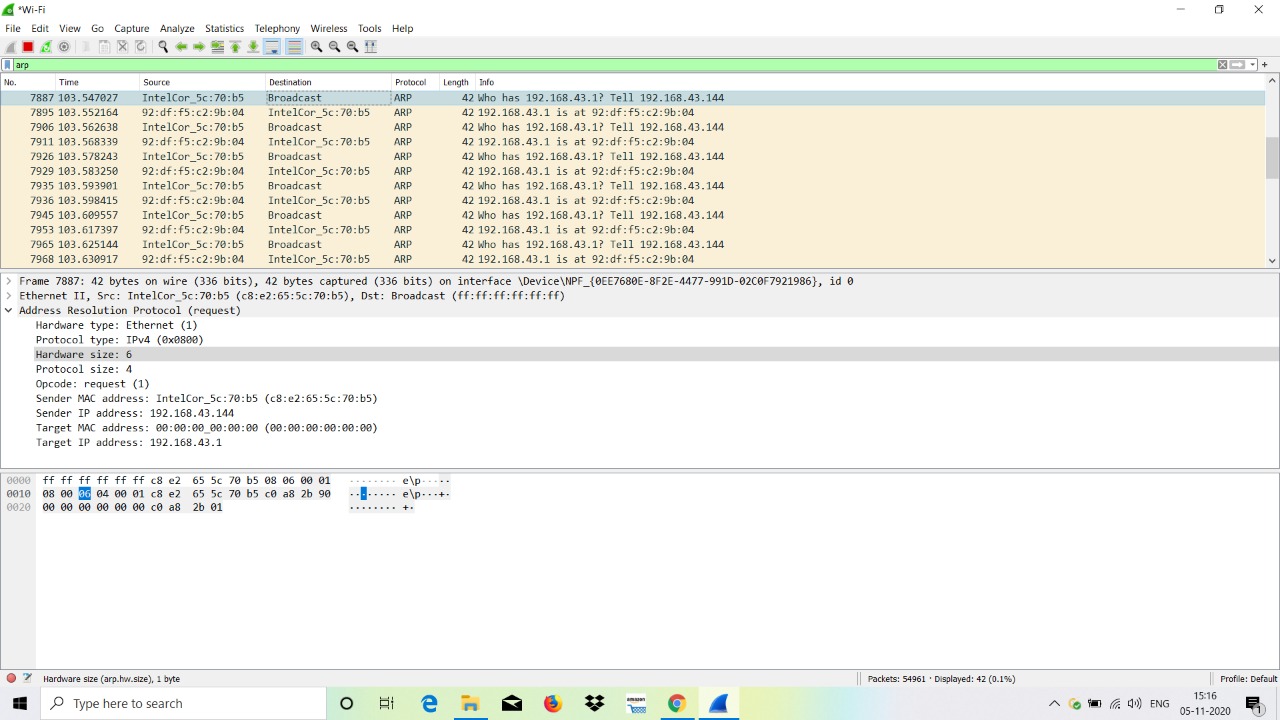
1. Appraise traces 1(c) using flow graph. To generate flow graph use following steps.
   1. In Wireshark click Statistics -> Flow Graph, Choose packets->Displayed packets, Choose flow type->General flow, Choose node address type->Standard source/destination addresses.
   2. Illustrate flow graph.



1. Appraise traces 1(c) using IO graph. To generate IO graph use following steps.
   1. In Wireshark click Statistics -> IO Graph

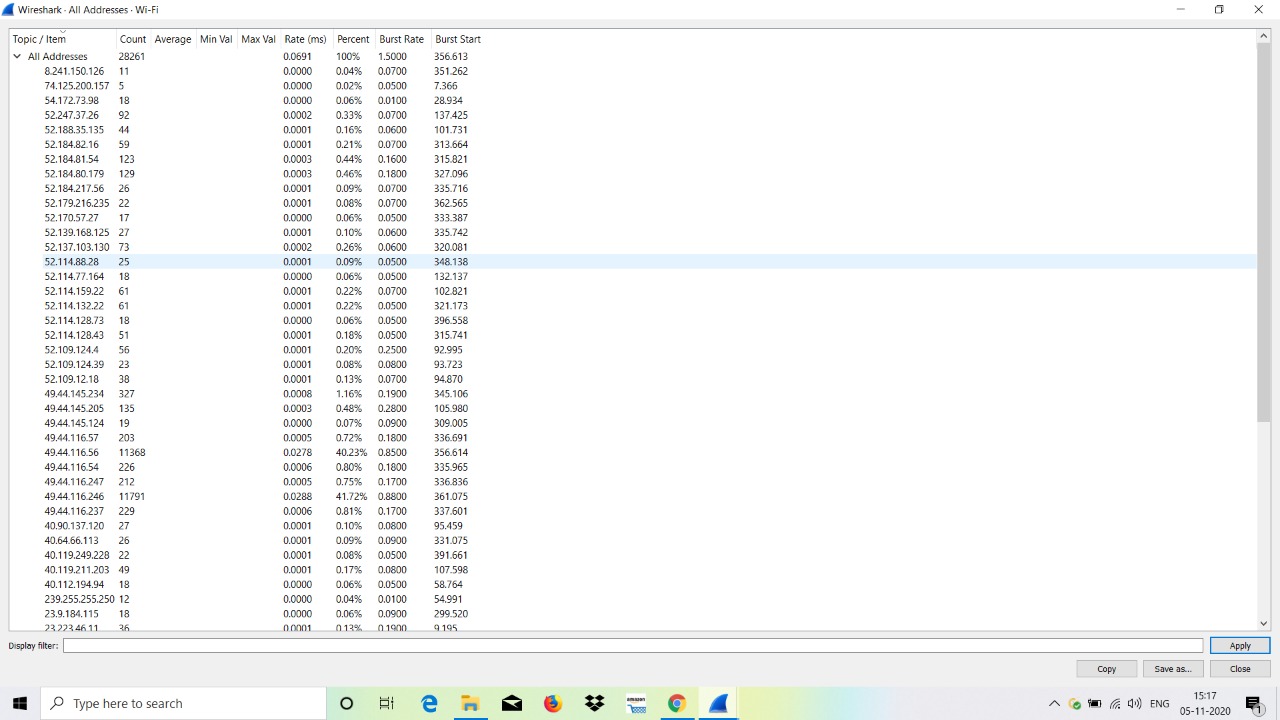


1. Filter a traffic based on following parameter:
   1. Protocol based filtration

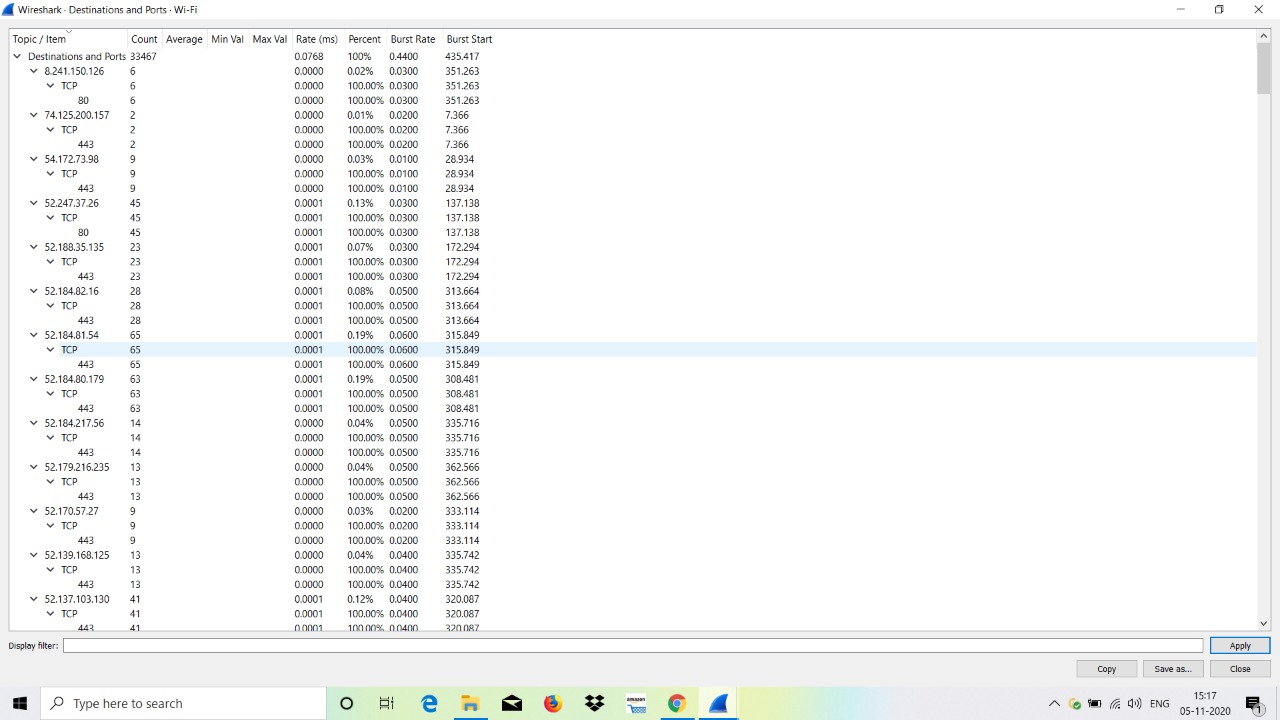


* 1. Source & destination IP based filtration

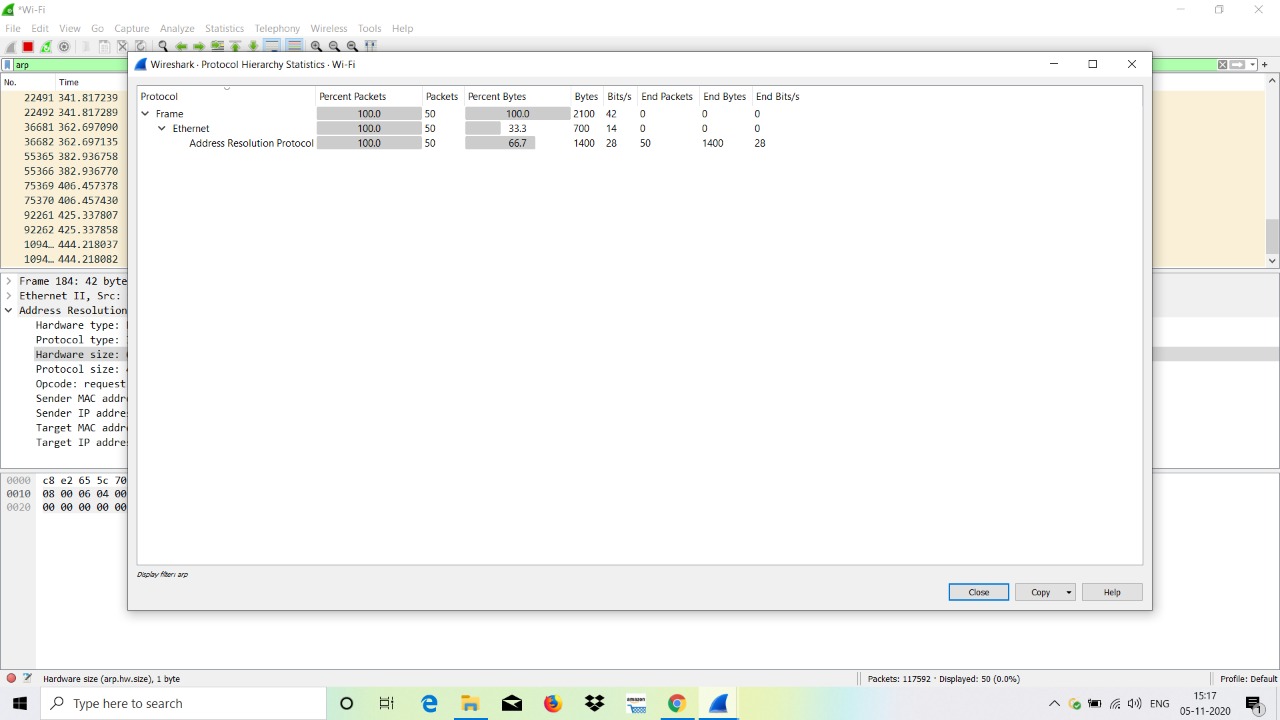
1. Examine the traces based on IP address using following steps.
   1. In Wireshark click Statistics ->IPv4 Statistics -> All addresses based traffic



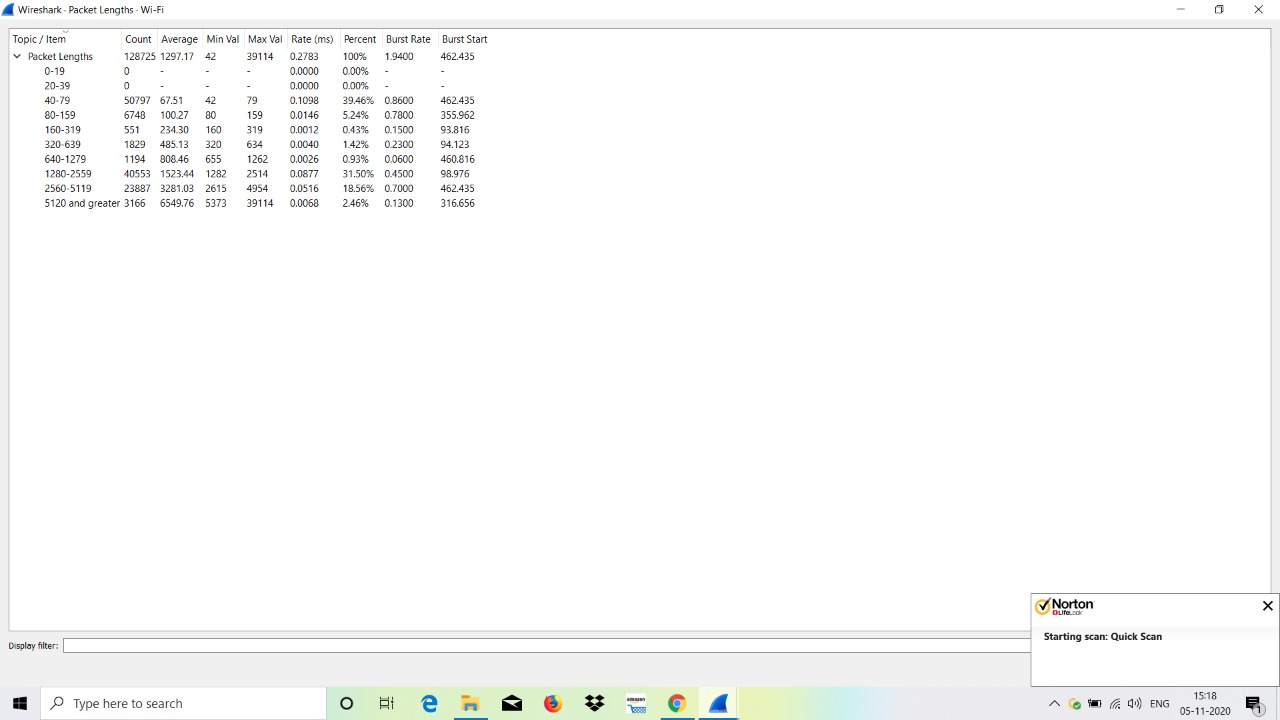
* 1. In Wireshark click Statistics ->IPv4 Statistics -> Destination and port based traffic



1. Examine the traces based on protocol hierarchy using following steps.
   1. In Wireshark packet list window click one trace.
   2. Then click Statistics -> protocol hierarchy
   3. You can view protocol hierarchy it contains detail statistics of protocols.
   4. List down protocols by expending the protocol hierarchy.



1. Classify the packet based on length using following steps.
   1. In Wireshark click Statistics -> packet length.
   2. You can view packet length.
   3. Find the maximum range of the packet length.



**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 3** | **Interpret the working principles of address resolution protocol using Wireshark** |
| **Date :** |

**Aim:**

To analyze the ARP traffic and its cache table using Wireshark and ARP dump command.

**Procedure:**

**ARP cache table analysis:**

1. Open a Windows terminal by clicking **Start -> Run**. Type **cmd**, and click **OK**. Issue the **arp-?** Command on the host computer, and examine the output.

|  |  |  |
| --- | --- | --- |
| S.No. | Command | Usage |
|  | arp -a | Displays the list of ARP entries available at that instant |
|  | arp –g | It also displays the list of ARP entries available at that instant as arp -a |
|  | arp –v | Displays current ARP entries in verbose mode. All invalid  entries and entries on the loop-back interface will be shown. |
|  | inet\_addr | It specifies an Internet address |
|  | -N if\_addr | Displays the ARP entries for the network interface specified by the address |
|  | arp –d | It Deletes the host specified by given IP address.It may be  ended with \* to delete all hosts. |
|  | arp –s | It adds the host and associates the Internet address with the Physical address. The Physical address is given as 6 hexadecimal bytes separated by hyphens |
|  | eth\_addr | It specifies a physical address |

1. Issue the command that displays ARP entries. What are the results?

C:\Users\Nadev>arp -a

Interface: 192.168.43.144 --- 0x3

Internet Address Physical Address Type

192.168.43.1 92-df-f5-c2-9b-04 dynamic

192.168.43.255 ff-ff-ff-ff-ff-ff static

224.0.0.22 01-00-5e-00-00-16 static

224.0.0.251 01-00-5e-00-00-fb static

224.0.0.252 01-00-5e-00-00-fc static

239.255.255.250 01-00-5e-7f-ff-fa static

255.255.255.255 ff-ff-ff-ff-ff-ff static

1. Issue the command to perform delete single entry from the ARP cache?

**C:\Users\Nadev>arp -d 123.12.11.123**

1. What command would be used to delete all ARP cache entries (flush ARP cache)?

**C:\Users\Nadev>arp -d \***

1. How was the Static ARP entry added to the ARP cache?

**C:\Users\Nadev>arp -s 11.11.11.11 nat-nat-nat-nat**

**ARP packet analysis using Wireshark:**

1. To generate ARP traffic using following steps.
   1. Open a Wireshark and start capturing Ethernet traffic by choosing *Capture*->*Interfaces*->*Ethernet*.
   2. In Wireshark filter: type **arp** and press enter.
   3. Open a terminal window by clicking on Start -> Run. Type cmd, and press OK.
   4. View ARP cache entries issue arp-a command in cmd.

**C:\Users\Nadev>arp -a**

**Interface: 192.168.43.144 --- 0x3**

**Internet Address Physical Address Type**

**192.168.43.1 92-df-f5-c2-9b-04 dynamic**

**192.168.43.255 ff-ff-ff-ff-ff-ff static**

**224.0.0.22 01-00-5e-00-00-16 static**

**224.0.0.251 01-00-5e-00-00-fb static**

**224.0.0.252 01-00-5e-00-00-fc static**

**239.255.255.250 01-00-5e-7f-ff-fa static**

**255.255.255.255 ff-ff-ff-ff-ff-ff static**

* 1. Ping to your neighbor node and make sure that neighbor node IP is not available in your arp cache table.

**ping mcet.in**

The ping command can be used to test network connectivity. By accessing other devices, ARP associations are dynamically added to ARP cache.

* 1. View arp traces in Wireshark packet list window.

Text

Description automatically generated

1. Using above traces 6(f), list the ARP request message fields and its size.

**Hardware type – 2 bytes**

**Protocol type – 2 bytes**

**Hardware size – 1 byte**

**Protocol size – 1 byte**

**Opcode – 2 bytes**

**Sender MAC Address – 6 bytes**

**Sender IP Address – 4 bytes**

**Target MAC Address – 6 bytes**

**Target IP Address – 4 bytes**

**Graphical user interface, text, application, email

Description automatically generated**

1. Using above traces 6(f), list the ARP Reply message fields and its size

**Hardware type – 2 bytes**

**Protocol type – 2 bytes**

**Hardware size – 1 byte**

**Protocol size – 1 byte**

**Opcode – 2 bytes**

**Sender MAC Address – 6 bytes**

**Sender IP Address – 4 bytes**

**Target MAC Address – 6 bytes**

**Target IP Address – 4 bytes**

**Graphical user interface, text, application, email

Description automatically generated**

1. Using above traces 6(f), find the target MAC address and target IP address of your destination computer in the ARP request packet.

Graphical user interface, text, application, email

Description automatically generated

1. Using above traces 6(f), find the Sender MAC address and Sender IP address of your computer in the ARP request packet.

Graphical user interface, text, application, email

Description automatically generated

1. Give the hexadecimal value for the sender IP address and target IP address in the ARP reply packet using traces 6(f).

**Sender IP Address**

Graphical user interface, application, Teams

Description automatically generated

**Target IP Address**

**Graphical user interface, application, email, Teams

Description automatically generated**

1. Appraise traces 6(f) using flow graph. To generate flow graph use following steps and make sure that Wireshark filter contains only arp traffic.
   1. In Wireshark click Statistics -> Flow Graph, Choose packets->Displayed packets, Choose flow type->General flow, Choose node address type->Standard source/destination addresses.

Graphical user interface, application, table, Excel

Description automatically generated

1. Appraise traces 6(f) using IO graph. To generate IO graph use following steps and make sure that Wireshark filter contains only arp traffic.
   1. In Wireshark click Statistics -> IO Graph
   2. In Wireshark IO Graphs window set filter under arp and IO graph will be automatically generated for ARP traffic.

A picture containing timeline

Description automatically generated

1. What is the peek time in the above generated IO graph?

A picture containing indoor, sitting, looking, window

Description automatically generated

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 4** | **Examine IP traffic and its routing options using Wireshark** |
| **Date :** |

**Aim:**

To examine IP header and its routing options using wireshark.

**Procedure:**

**IP & its routing options analysis using Wireshark:**

1. **Loose Source Routing**
2. To generate IP loose source routing using following steps.
   1. Open a Wireshark and start capturing Ethernet traffic by choosing *Capture*->*Interfaces*->*Ethernet*.
   2. In Wireshark filter: type **icmp**and press enter.
   3. Open a terminal window by clicking on Start -> Run. Type cmd, and press OK.
   4. Issue ping-? Command in cmd and view routing option commands.
   5. Ping to node and using loose source routing.
   6. View icmp traces in Wireshark packet list window.
3. Using above traces 1(f), list the IP request message fields and its size.
4. **Strict Source Routing**
5. How do you enforce strict source routing using ping command.
6. To generate IP strict source routing using following steps.
   1. Open a Wireshark and start capturing Ethernet traffic by choosing *Capture*->*Interfaces*->*Ethernet*.
   2. In Wireshark filter: type **icmp**and press enter.
   3. Open a terminal window by clicking on Start -> Run. Type cmd, and press OK.
   4. Issue ping-? Command in cmd and view routing option commands.
   5. Ping to node and using strict source routing.
   6. View icmp traces in Wireshark packet list window.
7. Using above traces 4(f), list the IP request message fields and its size.
8. **IP Record Route**
9. How do you enforce record route option using ping command?
10. To generate IP record route using following steps.
    1. Open a Wireshark and start capturing Ethernet traffic by choosing *Capture*->*Interfaces*->*Ethernet*.
    2. In Wireshark filter: type **icmp**and press enter.
    3. Open a terminal window by clicking on Start -> Run. Type cmd, and press OK.
    4. Issue ping-? Command in cmd and view routing option commands.
    5. Ping to node and using record route.
    6. View icmp traces in Wireshark packet list window.
11. Write down routes list displayed with a help of 7(e).

1. Find the recorded route IPs in the traces 7(f)?
2. **IP Fragmentation**
3. How will you fragment packets using command line?
4. Examine IP fragmentation using offline traces. Find the fields and its size associated with IP fragmentation using offline traces.
5. Appraise the fragmented datagrams using flow diagram.
6. **Tracert**

Tracert is a route tracing utility that displays a list of near-side router interfaces of the routers along the path between a source host and a destination. Tracert uses the IP TTL field in ICMP Echo Requests and ICMP Time Exceeded messages to determine the path from a source to a destination through an IP internetwork.

1. Open a Windows terminal by clicking **Start -> Run**. Type **cmd**, and click **OK**. Issue the tracert **-?** Command on the host computer, and examine the output.

|  |  |  |
| --- | --- | --- |
| **S.No** | **Command Name** | Usage |
|  | tracert –d |  |
|  | tracert –h maximum\_hops |  |
|  | tracert –j host-list |  |
|  | tracert –w timeout |  |
|  | tracert –R |  |
|  | tracert –S srcaddr |  |
|  | tracert -4 |  |
|  | tracert -6 |  |

1. Find the route for 192.168.2.21?
2. Appraise the trace route using flow diagram.

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 5** | **Classify the internet control message protocol messages using Wireshark** |
| **Date :** |

**Aim:**

To analyze ICMP header format and its types using wireshark.

**Procedure:**

**ICMP Traffic Analysis using Wireshark:**

1. To generate ICMP traffic using following steps.
   1. Open a Wireshark and start capturing Ethernet traffic by choosing *Capture*->*Interfaces*->*Ethernet*.
   2. In Wireshark filter: type **icmp**and press enter.
   3. Open a terminal window by clicking n Start -> Run. Type cmd, and press OK.
   4. Issue ping-? Command in cmd and view routing option commands.
   5. Ping to node.
   6. View icmp traces in Wireshark packet list window.
2. Examine one of the ping request packets sent by your host using 1(f) traces. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?
3. Examine the corresponding ping reply packet using 1(f) traces. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?
4. Record information from the *first* echo request packet for 1(f) .

|  |  |
| --- | --- |
| **Field** | **Value** |
| Type |  |
| Code |  |
| Checksum |  |
| Identifier |  |
| Sequence number |  |
| Data |  |

1. Record information from the *first* echo reply packet for 1(f).

|  |  |
| --- | --- |
| **Field** | **Value** |
| Type |  |
| Code |  |
| Checksum |  |
| Identifier |  |
| Sequence number |  |
| Data |  |

1. To generate ICMP host unreachable traffic using following steps.
   1. Open a Wireshark and start capturing Ethernet traffic by choosing *Capture*->*Interfaces*->*Ethernet*.
   2. In Wireshark filter: type **icmp**and press enter.
   3. Open a terminal window by clicking n Start -> Run. Type cmd, and press OK.
   4. Issue ping-? Command in cmd and view routing option commands.
   5. Ping to unavailable node.
   6. View icmp traces in Wireshark packet list window.
2. Examine one of the ping requests & reply packets sent by your host using 6(f) traces. What are the Type/Code values for host unreachable?
3. How to generate ICMP Time-to-live exceeded message using ping command and demonstrate it?
4. Examine one of the ping requests & reply packets sent by your host for TTL exceed traces. What are the Type/Code values for TTL exceed message?
5. How long is the ICMP header of a TTL Exceeded packet? Select different parts of the header in Wireshark to see how they correspond to the bytes in the packet.
6. Is the data in the echo reply the same as in the echo request or different?
7. Which ICMP message type is used to return information to the sender?
8. Appraise the flow diagram for 1(f) traces.
9. Appraise the IO Graph for 1(f) traces.

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 6** | **Analyze the TCP connection establishment and termination using Wireshare** |
| **Date :** |

**Aim:**

Analyze the TCP Header format & TCP connection establishment and termination using Wireshare.

**Procedure:**

**TCP Traffic Analysis using Wireshark:**

1. To generate TCP traffic using following steps.
   1. Open a Wirkshark and start capturing Ethernet traffic by choosing *Capture*->*Interfaces*->*Ethernet*.
   2. In Wireshark filter: type **TCP** and press enter.
   3. Open a web browser.
   4. Type 118.91.233.179 & press enter
   5. View TCP traces in Wireshark packet list window.
2. How many transactions was involved 1(e) to make successful connection establishment?
3. What are the control flags used for connection establishment and write down trace numbers of connection establishment using 1(e) traces?
4. Using above traces 1(e), list the TCP fields and its size.
5. Using above traces 1(e), list the IP fields and its size.
6. What is the source port address for establishment of TCP connection & find it using the trace 1(e)?
7. Appraise traces 1(e) using flow graph. To generate flow graph use following steps and make sure that Wireshark filter contains only**TCP** traffic.
   1. In Wireshark click Statistics -> Flow Graph, Choose packets->Displayed packets, Choose flow type->TCP flow, Choose node address type->Standard source/destination addresses.
   2. Illustrate TCP connection establishment using flow graph.
8. Appraise traces 1(e) using IO graph. To generate IO graph use following steps and make sure that Wireshark filter contains only **TCP** traffic.
   1. In Wireshark click Statistics -> IO Graph
   2. In Wireshark IO Graphs window set filter under TCP and IO graph will be automatically generated for TCP traffic.
9. What is the peek time in the above generated IO graph?
10. What is the control flags used for connection termination?
11. Appraise connection termination using flow graph. To generate flow graph use following steps and make sure that Wireshark filter contains only **TCP** traffic.
    1. In Wireshark click Statistics -> Flow Graph, Choose packets->Displayed packets, Choose flow type->TCP flow, Choose node address type->Standard source/destination addresses.
    2. Illustrate TCP connection termination using flow graph. Include the following features on your diagram:
12. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between your system and 118.91.233.179? What is it in the segment that identifies the segment as a SYN segment?
13. What is the sequence number of the SYNACK segment sent by 118.91.233.179 to your computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? What is it in the segment that identifies the segment as a SYNACK segment?
14. Generate round trip time and throughput for TCP traffic by choosing Statistics->TCP Stream Graph in Wireshare window.
15. Use the Time-Sequence-Graph (Stevens) plotting tool to view the sequence number versus time plot of segments being sent from your system to the Telnet server
16. Using the Wireshark capture of the first TCP session start-up (SYN bit set to 1), fill in information about the TCP header:

1. Using the Wireshark capture of the second TCP session, From 118.91.233.179 to your computer (only SYN and ACK bits are set to 1), fill in information about the TCP header:

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 7** | **Implementation of socket programming using TCP sockets** |
| **Date :** |

**Aim:**

To implement any one below give client server application using TCP socket.

1. Implementation of TCP for Client Server Chat
2. Implementation of TCP for Client Server Echo
3. Implementation of TCP for Client Server Chat and convert the output to upper case.
4. Implementation of sliding window protocol to send one character at a time using simplex mode.
5. Implementation of sliding window protocol to send string at a time using simplex mode.
6. Implementation of sliding window protocol to send one character at a time using simplex mode and convert the output to upper case.
7. Implementation of sliding window protocol to send string at a time using simplex mode and convert the output to upper case.

**Procedure:**

**Server**

**Client**

**Implementation**

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 8** | **Configure the LAN for generating data traffic using GNS3** |
| **Date :** |

**Aim:**

To create a LAN and analyze data traffic using GNS3 with wireshark.

**Procedure:**

1. Select the connectivity nodes and place it in workspace.
2. Select the end nodes and place it in workspace.
3. Make a link between end nodes with connectivity nodes.
4. Start all the nodes using run button.
5. Configure IP address, subnet mask and default gateway for all the nodes.
6. Generate data traffic.
7. Analyze the data traffic using wireshark.

**Implementation and Screenshots:**

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 9** | **Configure a topology and analyze traffic using riverbed** |
| **Date :** |

**Aim:**

To create a LAN and analyze data traffic using riverbed.

**Procedure:**

1. Create an empty scenario for topology and select the network scale as office.
2. Specify the size of X and Y span.
3. Select technology as sm\_int\_model\_list.
4. Under rapid configuration set center node model as simple switch and peripheral node as simple workstation and Link as Ethernet.
5. Select following things from open object palette.
   1. Server
   2. Simple application
   3. Simple profile and
   4. Connect server to topology using Ethernet.
6. Configure the application server by setting the number of rows as 1(server containing single application), application name as database and traffic as high load.
7. Configure profile application by setting the database as an application name.
8. Configure the topology by setting support service as all.
9. Configure the server by setting support service as single application and profile name as sm\_int\_profile.
10. Measure the Ethernet delay for the entire network.
11. Measure the load of the server for the packet transfer per second.

**Implementation and Screenshots:**

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |

|  |  |
| --- | --- |
| **Ex. No. : 10** | **Experiment with NS2 to configure LAN for generating data traffic** |
| **Date :** |

**Aim:**

Configure a Local Area Network using NS2 and analyze traffic generated in LAN.

**Procedure:**

1. Wired node creation
   1. Create 4 node as n0, n1, n2, n3
   2. Make a link between n0 and n1 as duplex connection with 10Mbs speed
   3. Make a link between n2 and n0 as duplex connection with 10Mbs speed
   4. Make a link between n3 and n0 as duplex connection with 10Mbs speed
   5. Arrange the nodes in following order
      1. Node n0 and n1 in right side
      2. Node n0 and n2 in left upper side
      3. Node n0 and n3 in left down side
2. Generate UPD traffic
3. The CBR agents have to be connected to the Null agent
4. Set packet size as 512 bytes
5. Set data transfer rate as 200Kb
6. Set application start time as 1.0 and stop time as 5.0

**Implementation**

**Interpretation and Outcomes:**

|  |  |
| --- | --- |
| **Marks** | |
| Preparation | /20 |
| Observation & Results | /25 |
| Viva Voce | /10 |
| Record | /20 |
| Total | /75 |