

ExNo: 5 Experiment on Packet Capture

DATE: 27/7 TOOL: WIRESHARK.

Aim:

To capture and analyze network packets using wireshark and apply filters to display specific protocols.

Packet sniffer:

→ sniffs message being sent/receive from/by your computer

→ store and display the contents of the various protocol fields in the message.

Description

Wireshark, a network analysis tool formerly known as Ethereal, captures packet in real time and display them in human-readable format. Wireshark includes filters, color coding and other features that let you dig deeper into network traffic and inspect individual packets and inspect individual packets and analyse packets using capturing and analysing packets using Wireshark tool.

* To filter, capture view packets in Wireshark tool.

* capture 100 packets from the eth0 interface and IEEE 802.3 LAN Interface and save it.

Getting Wireshark :-

Wireshark can be downloaded for windows or macOS from its official website.

Capturing packets :

After downloading and installing Wireshark launch it and double-click the name of a network interface under capture to start capturing packets on that interface.

The "Packet Bytes" Pane

The packet bytes pane shows the data of the current packets I selected in the "Packet List" pane in a hexdump style.

Sample Captures

If there's nothing interesting on your network to inspect, Wireshark's wiki has you covered.

Time	Source	Destination	Protocol	Length	Info
1.468313	7e00:0f8:c965:3513..	f662::fb	HTTP	319	Standard query response 0x0000 PTR DELL::DESKTOP-C...
2.059622	142.251.221.202	172.16.75.53	UDP	1115	443 - 50988 Len=1873
3.071491	172.16.75.53	142.251.221.202	UDP	75	50988 + 443 Len=33
4.095581	CloudNetwork_77:ff:bf	Broadcast	ARP	42	ARP Announcement for 172.16.75.53
4.823698	172.16.75.53	142.251.221.202	UDP	71	50988 + 443 Len=29
4.869364	142.251.221.202	172.16.75.53	UDP	67	443 - 50988 Len=25
4.295469	172.16.75.53	142.251.221.202	UDP	73	50988 + 443 Len=29
4.314443	142.251.221.202	172.16.75.53	UDP	67	443 - 50988 Len=25
4.408713	172.16.75.247	239.8.0.8	UDP	1878	41173 + \$1721 Len=1816
4.488748	172.16.75.139	236.8.0.1	UDP	92	57441 + 6666 Len=59
4.516508	172.16.75.53	142.251.221.202	UDP	71	50988 + 443 Len=29
4.532225	172.16.75.53	74.125.208.188	TCP	64	50988 - 9228 [FIN, ACK] Seq=1 Win=512 Len=6
4.537791	74.125.208.188	172.16.72.1	TCP	66	509249 + 53 [SYN] Seq=0 Win=65535 Len=64
4.541281	172.16.75.53	172.16.75.53	TCP	65	509249 - 53 [SYN, ACK] Seq=1 Win=29208 Len=6
4.541295	172.16.75.53	172.16.72.1	TCP	54	509249 + 53 [ACK] Seq=2 Win=131072 Len=6
4.541301	172.16.75.53	172.16.72.1	TCP	56	509249 + 53 [SYN, ACK] Seq=1 Ack=1 Win=131072 Len=6
4.541410	172.16.75.53	172.16.72.1	DNS	68	Standard query 0x2460 A stali.google.com
4.543585	172.16.72.1	172.16.75.53	TCP	54	53 - 509249 [ACK] Seq=1 Ack=3 Win=29208 Len=6
4.543591	142.251.221.202	172.16.75.53	UDP	97	443 - 50988 Len=25
4.543597	172.16.72.1	172.16.75.53	DRX	135	Standard query response 0x2460 A stali.google.com
4.544139	172.16.75.53	172.253.318.188	TCP	66	509538 - 53220 [SYN] Seq=0 Win=65535 Len=64
4.544143	172.16.75.53	172.16.72.1	TCP	64	509249 + 53 [SYN, ACK] Seq=17 Ack=2 Win=512 Len=6

Time	Source	Destination	Protocol	Length	Info
172.16.75.53					
4.546223					
4.537761					
4.541150					
4.541195					
4.541381					
4.541410					
4.545505					
4.545505					
4.545557					
4.545799					
4.546138					

1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on

link II, Src: CloudNetwork_77:ff:bf (4c:82:a9:77:ff:bf), Dst: Brodcast (ff:ff:ff:ff:ff:ff)

2: 0.000000

3: 0.098875

4: 1.097894

5: 1.098581

6: 1.098581

7: 1.098581

8: 1.098581

9: 1.098581

10: 1.098581

11: 1.098581

12: 1.098581

13: 1.098581

14: 1.098581

15: 1.098581

16: 1.098581

17: 1.098581

18: 1.098581

19: 1.098581

20: 1.098581

21: 1.098581

22: 1.098581

23: 1.098581

24: 1.098581

25: 1.098581

26: 1.098581

27: 1.098581

28: 1.098581

29: 1.098581

30: 1.098581

31: 1.098581

32: 1.098581

33: 1.098581

34: 1.098581

35: 1.098581

36: 1.098581

37: 1.098581

38: 1.098581

39: 1.098581

40: 1.098581

41: 1.098581

42: 1.098581

43: 1.098581

44: 1.098581

45: 1.098581

46: 1.098581

47: 1.098581

48: 1.098581

49: 1.098581

50: 1.098581

51: 1.098581

52: 1.098581

53: 1.098581

54: 1.098581

55: 1.098581

56: 1.098581

57: 1.098581

58: 1.098581

59: 1.098581

60: 1.098581

61: 1.098581

62: 1.098581

63: 1.098581

64: 1.098581

65: 1.098581

66: 1.098581

67: 1.098581

68: 1.098581

69: 1.098581

70: 1.098581

71: 1.098581

72: 1.098581

73: 1.098581

74: 1.098581

75: 1.098581

76: 1.098581

77: 1.098581

78: 1.098581

79: 1.098581

80: 1.098581

81: 1.098581

82: 1.098581

83: 1.098581

84: 1.098581

85: 1.098581

86: 1.098581

87: 1.098581

88: 1.098581

89: 1.098581

90: 1.098581

91: 1.098581

92: 1.098581

93: 1.098581

94: 1.098581

95: 1.098581

96: 1.098581

97: 1.098581

98: 1.098581

99: 1.098581

100: 1.098581

101: 1.098581

102: 1.098581

103: 1.098581

104: 1.098581

105: 1.098581

106: 1.098581

107: 1.098581

108: 1.098581

109: 1.098581

110: 1.098581

111: 1.098581

112: 1.098581

113: 1.098581

114: 1.098581

115: 1.098581

116: 1.098581

117: 1.098581

118: 1.098581

119: 1.098581

120: 1.098581

121: 1.098581

122: 1.098581

123: 1.098581

124: 1.098581

125: 1.098581

126: 1.098581

127: 1.098581

128: 1.098581

129: 1.098581

130: 1.098581

131: 1.098581

132: 1.098581

133: 1.098581

134: 1.098581

135: 1.098581

136: 1.098581

137: 1.098581

138: 1.098581

139: 1.098581

140: 1.098581

141: 1.098581

142: 1.098581

143: 1.098581

144: 1.098581

145: 1.098581

146: 1.098581

147: 1.098581

148: 1.098581

149: 1.098581

150: 1.098581

151: 1.098581

152: 1.098581

153: 1.098581

154: 1.098581

155: 1.098581

156: 1.098581

15

Filtering packet :-

If you're trying to impact something specific, such as the traffic a program sends when phoning home, it helps to close down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

		Time	Source IP	Destination IP	Comment
4.541410	59249		172.16.75.53	172.16.72.1	Standard query 0x2460 A mtalk.google.com C 53
4.545557	59249				Standard query response 0x2460 A mtalk.google.com C 53

You can also click Analyze > Display Filters to choose a filter from among the default filters included in wireshark close the window and you'll find a filter has been applied automatically wireshark is showing you the packets that make up the conversion.

Inspecting packets :-

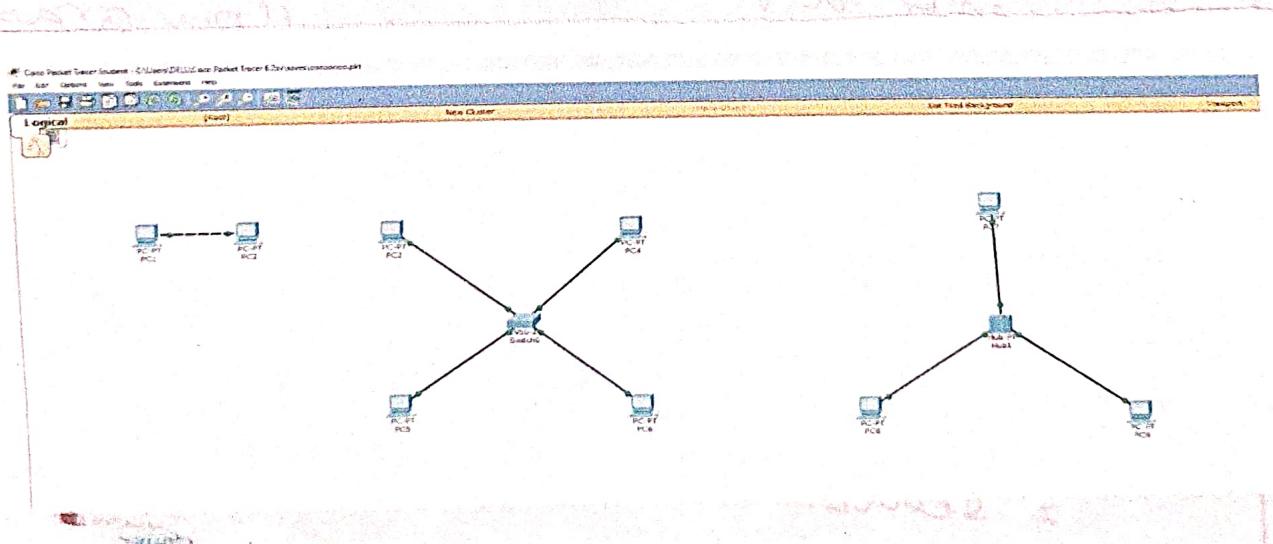
click a packet to select it

and you can dig down to view its details

No.	Time	Source	Destination	Protocol	Length	Info
19273	2008.088756	0.0.0.0	255.255.255.255	DHCP	340	DHCP Discover - Transaction ID 0x73626813
15878	270.200004	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x73626813
15137	329.009535	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x1d470563
20110	453.427083	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x2c889008
39418	484.148567	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x71442f13
31218	482.582826	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x73537966
31850	501.217956	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x585ffeb2
33433	538.900519	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0x9c224a08
34801	587.214951	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x7d1d4554
36462	622.067521	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x73937f5d
47871	810.467653	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x1105f52
47976	813.533985	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x87531696
48582	820.898239	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x68d45b47
49105	840.979214	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x7a116a96
49356	854.353885	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x2318a6e3
49944	861.460416	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x378a378e
50294	875.695584	0.0.0.0	255.255.255.255	DHCP	378	DHCP Request - Transaction ID 0x78a378e

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

Flow Graph :- gives a better understanding of what we see.



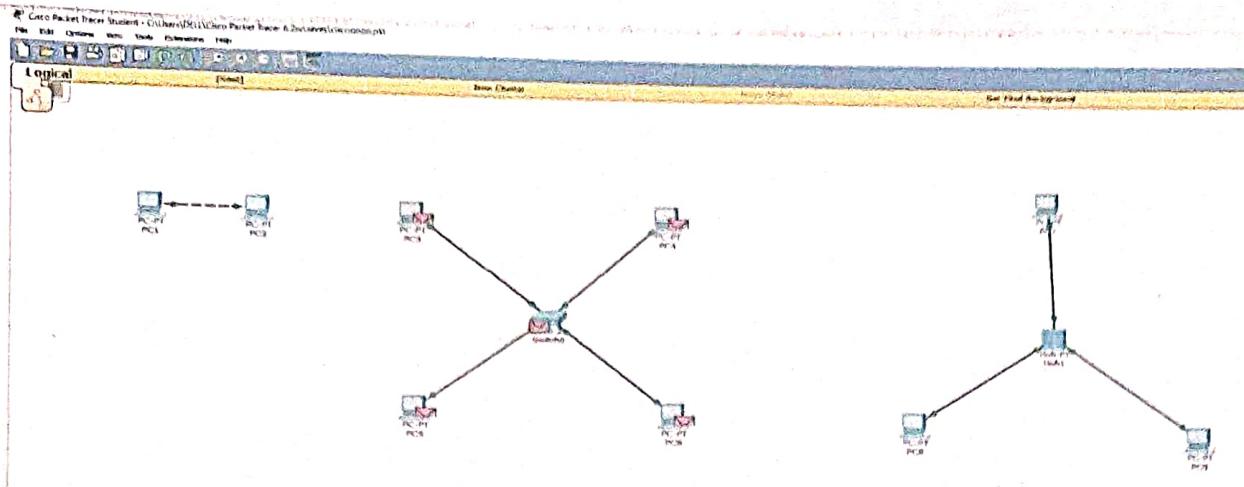
~~capturing and analysing packets using Wireshark tool.~~

To filter captured, view, packets in Wireshark Tool capture 100 packets

from the Ethernet : IEEE 802.3 LAN Interface and save it.

Procedure:

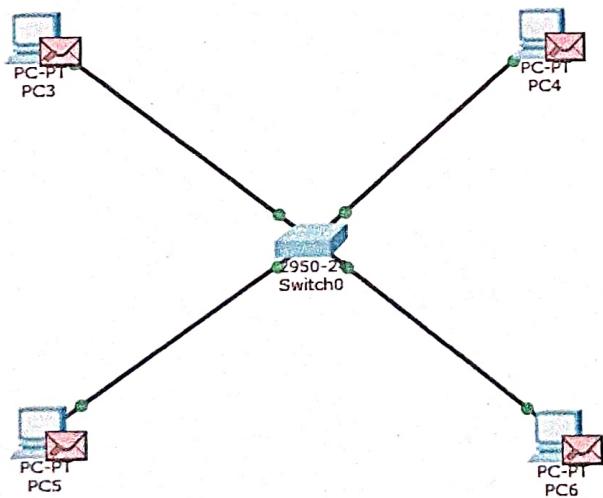
- Select Local Area connection in wireshark
 - Go to capture → option
 - Select stop capture automatically after 100 packets.
 - Then click start capture & save the packets.
1. Create a filter to display only TCP / UDP packets, inspect the packets and provide the flow graph,
 2. Create a filter to display only ARP packets and inspect the packets
 3. Create a filter to display only DNS packets and provide the flow graph.
 - * Go to captured → option
 - * Select stop capture automatically after 100 packets.
 - * Then click start capture.
 - * Search DNS packets in search bar
 - * To see flow graph click statics → Flow graph
 - * Save the packets



4. Create a filter to display only HTTP packets and inspect the packets.

procedure:

- select Local Area connection in wireshark
- Go to capture → option
- select stop capture automatically after 100 packet
- Then click start capture
- search HTTP packets and save packets.



5. create a filters to display only IP / ICMP packets to inspect the packets.

- Select local Area connection in wireshark
 - Go to capture → option
 - Select stop capture automatically after 100 packets.
 - Then click start capture
 - Select search ICMP IP packets in search box
 - Save the packets.
6. Create a filter to display only DHCP packets and inspect the packets

student Observation :-

1) what is Promiscuous mode?

Promiscuous mode is a setting for a network interface card (NIC) where it captures all network packets that pass through it, not just the ones addressed to it. It is used in packet switching and network Monitoring.

2) Does ARP packets have transport layer header ? Explain

No, ARP (Address resolution protocol) packets do not have a transport layer header, ARP works at Datalink layer to map an IP address to MAC address.

3) Which transport layer protocol is used by DNS ?

DNS can use: UDP on port 53 for most queries (factor) TCP on port 53 for tasks like zone transfers or responses exceeding 512 bytes.

4) What is the port number used by ~~HTTP~~ protocol?

HTTP uses port 80 (TCP) for secure HTTP (HTTPS) the port is 443 (TCP).

5. What is a broadcast IP address?

A broadcast IP address is an address used to send data to all hosts in a network simultaneously.

In IPv4, it's highest address in a subnet.

Eg:-

For network 192.168.1.0/24, the broadcast is 192.168.1.255.

Result:

Experiments on packet capture tool, wireShark was successfully studied.