Practical-5

AIM Experiments on Packet capture tool: Wireshark

Packet Sniffer

- Sniffs messages being sent/received from/by your computer
- Store and display the contents of the various protocol fields in the messages
- Passive program
  - never sends packets itself
  - no packets addressed to it
  - receives a copy of all packets (sent/received)

Packet Sniffer Structure Diagnostic Tools

- Tcpdump
  - E.g. tcpdump -enx host 10.129.41.2 -w exe3.out
- Wireshark
  - wireshark -r exe3.out

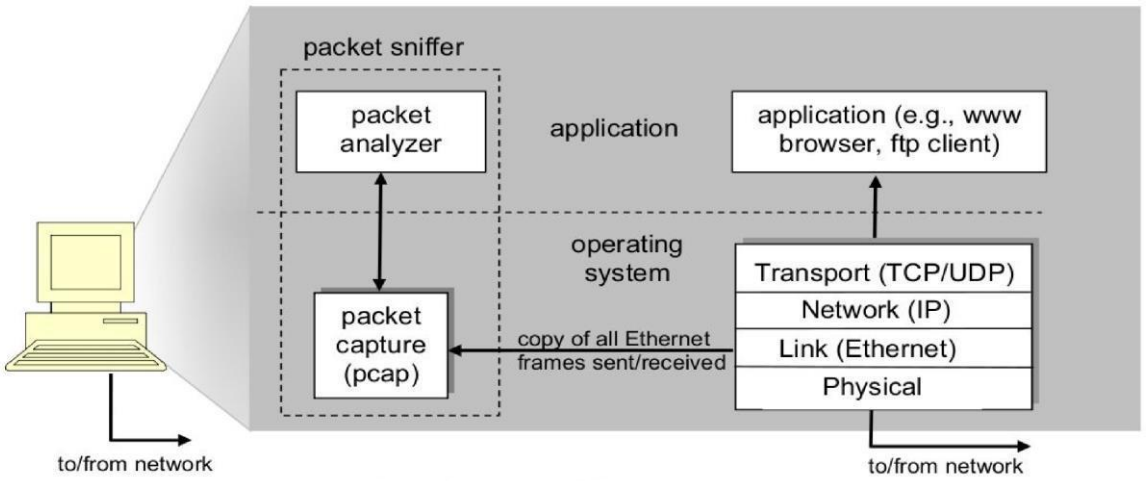


Figure 1: Packet sniffer structure

DESCRIPTION:

WIRESHARK

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, and other features that let you dig deep into network traffic and inspect individual packets. You can use Wireshark to inspect a suspicious program's network traffic, analyze the traffic flow on your network, or troubleshoot network problems.

What we can do with Wireshark:

- Capture network traffic
- Decode packet protocols using dissectors
- Define filters – capture and display
- Watch smart statistics
- Analyze problems
- Interactively browse that traffic

Wireshark used for:

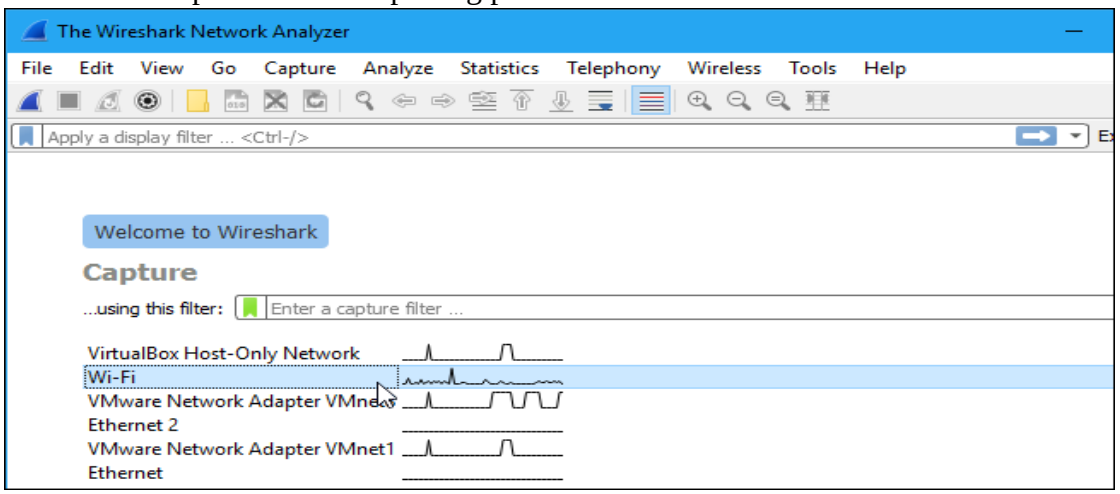
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

Getting Wireshark

Wireshark can be downloaded for Windows or macOS from its official website. For Linux or another UNIX-like system, Wireshark will be found in its package repositories. For Ubuntu, Wireshark will be found in the Ubuntu Software Center.

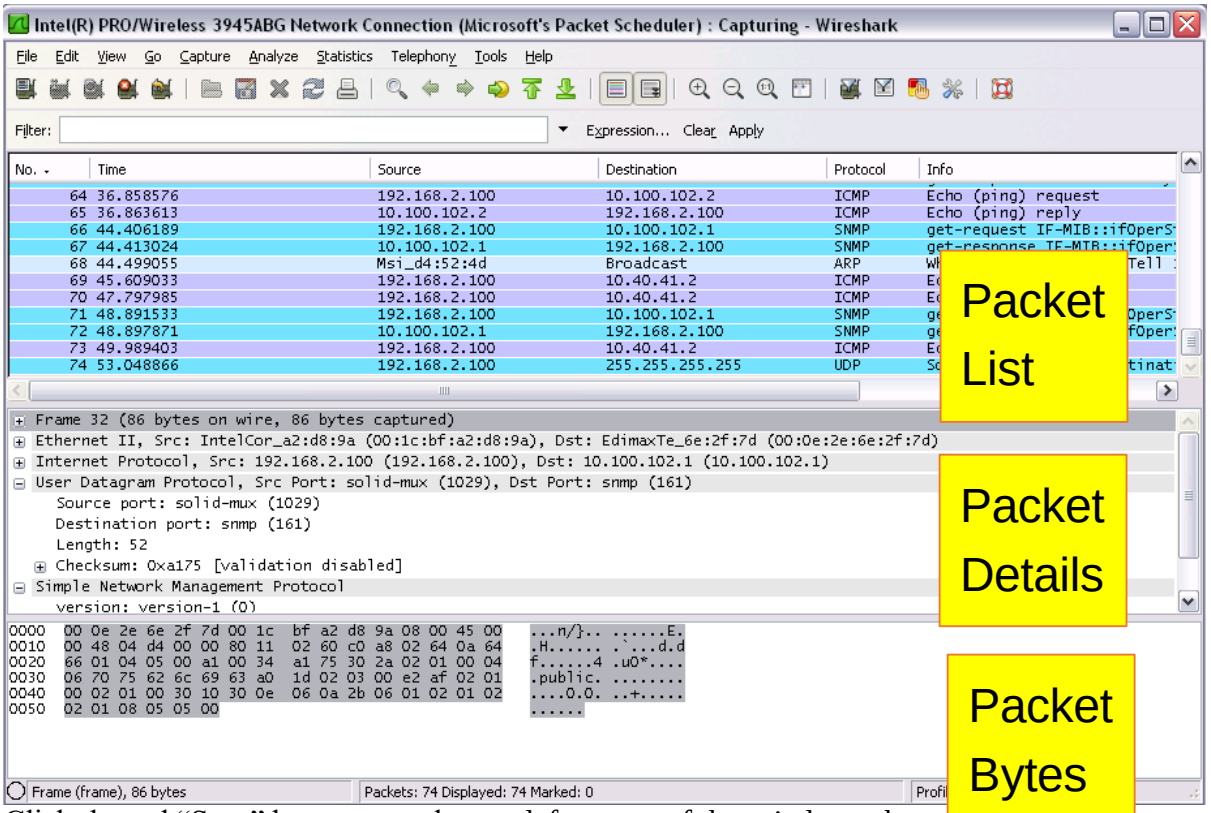
Capturing Packets

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



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 have promiscuous mode enabled—it's enabled by default—you'll also see all the other packets on the network instead of only packets addressed to your network adapter. To check if promiscuous mode is enabled, click Capture > Options and verify the "Enable promiscuous mode on all interfaces" checkbox is activated at the bottom of this window.



Click the red "Stop" button near the top left corner of the window wh e stop capturing traffic.

The "Packet List" Pane

The packet list pane displays all the packets in the current capture file. The "Packet List" pane Each line in the packet list corresponds to one packet in the capture file. If you select a line in this pane, more details will be displayed in the "Packet Details" and "Packet Bytes" panes.

The "Packet Details" Pane

The packet details pane shows the current packet (selected in the "Packet List" pane) in a more detailed form. This pane shows the protocols and protocol fields of the packet selected in the "Packet List" pane. The protocols and fields of the packet shown in a tree which can be expanded and collapsed.

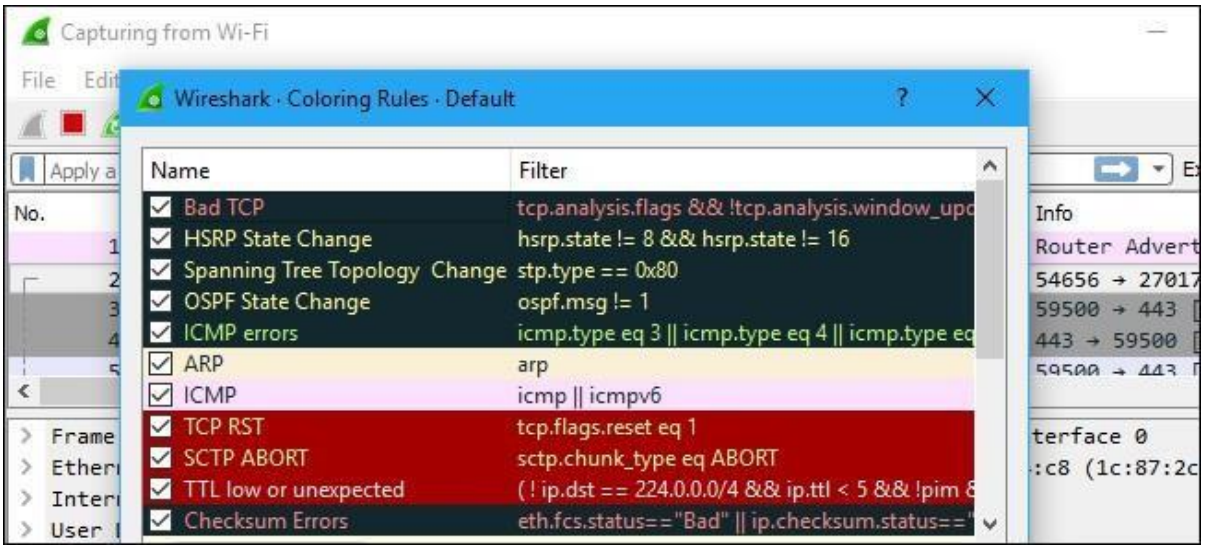
The "Packet Bytes" Pane

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

Color Coding

You'll probably see packets highlighted in a variety of different colors. Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light blue is UDP traffic, and black identifies packets with errors—for example, they could have been delivered out of order.

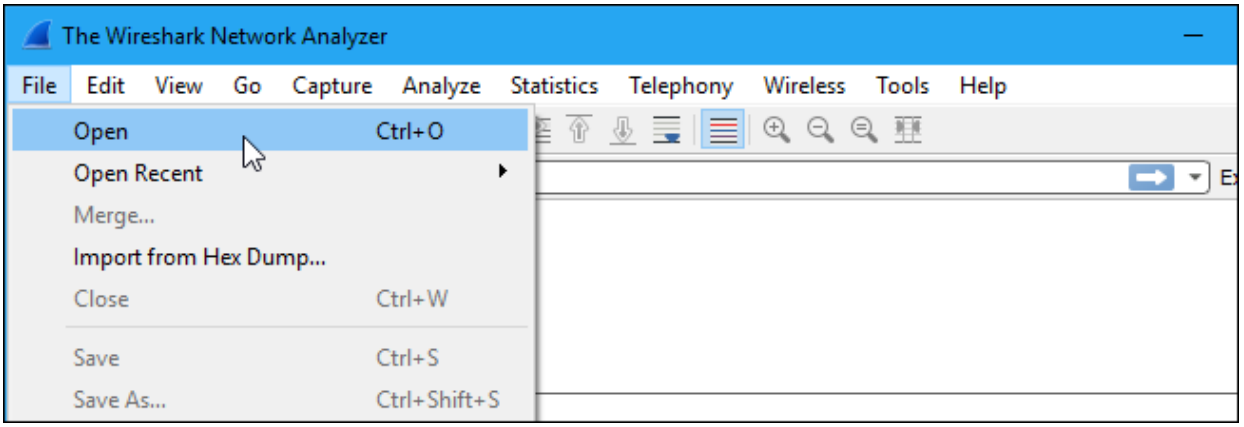
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



Sample Captures

If there's nothing interesting on your own network to inspect, Wireshark's wiki has you covered. The wiki contains a page of sample capture files that you can load and inspect. Click File > Open in Wireshark and browse for your downloaded file to open one.

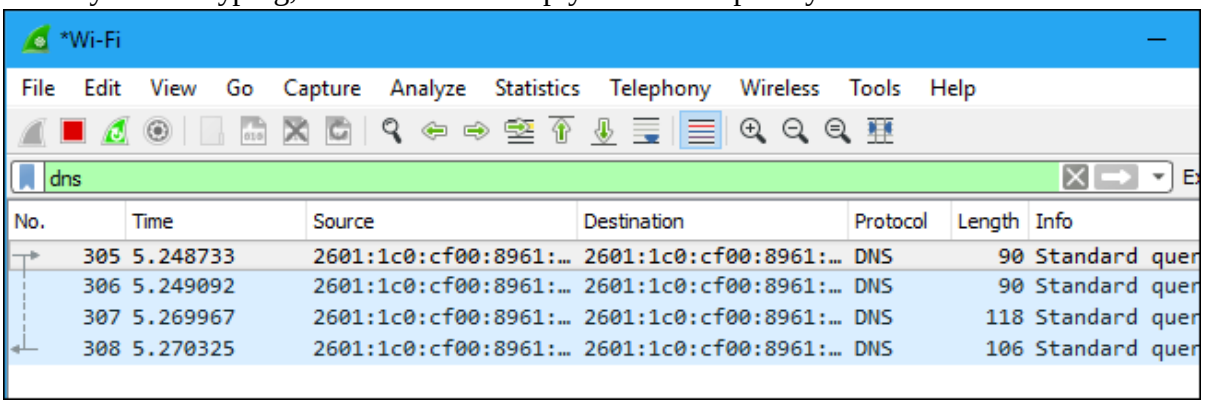
You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.



Filtering Packets

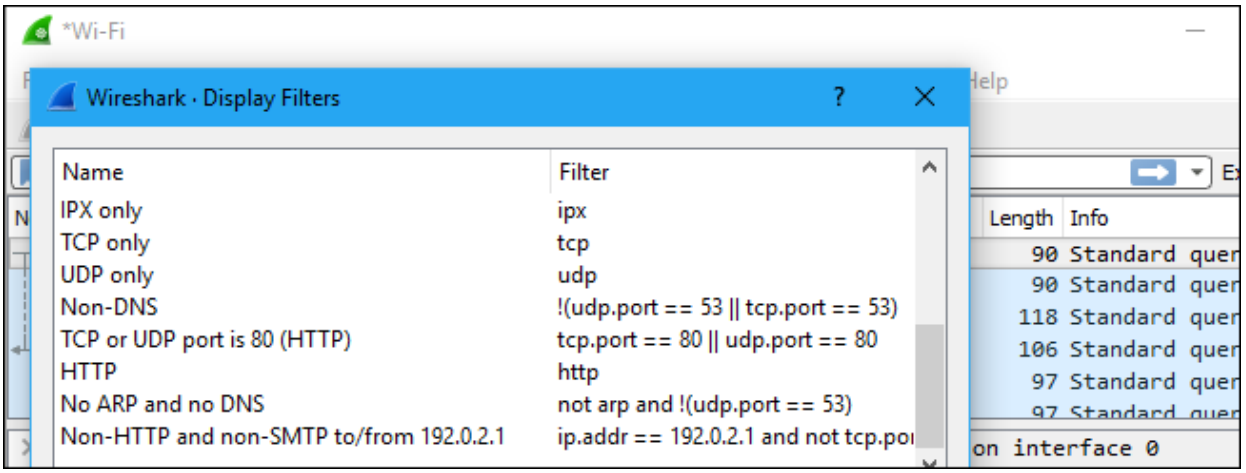
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters 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 "dns" and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.

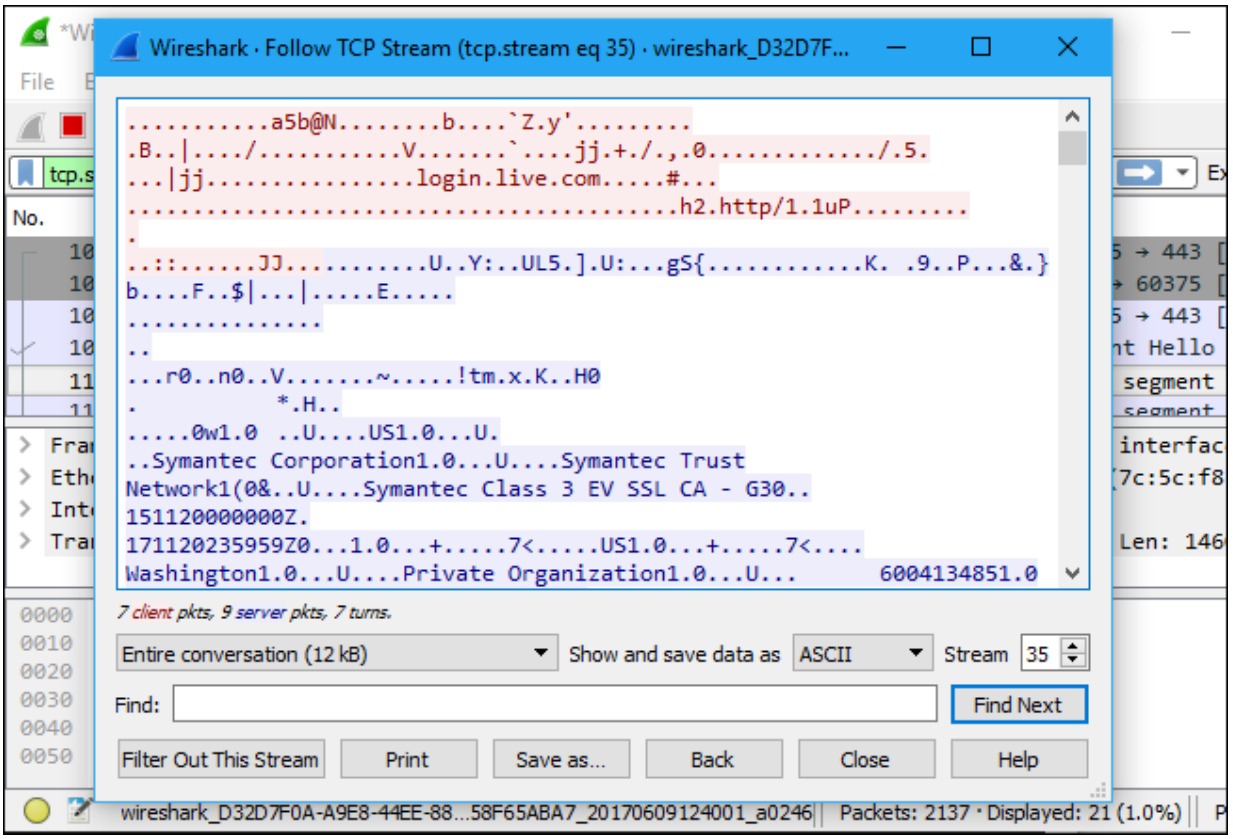


You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

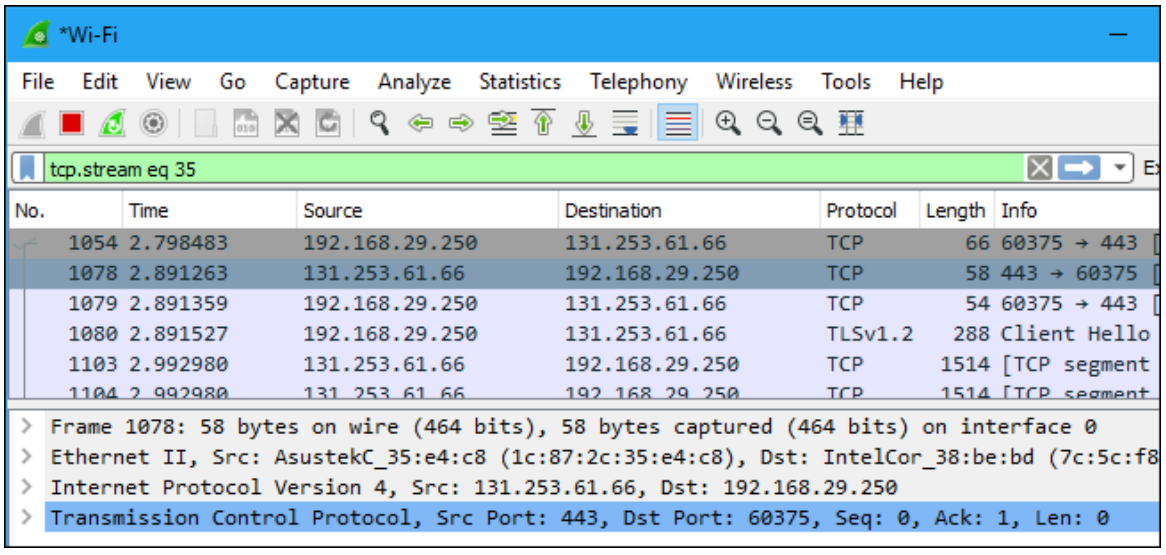
For more information on Wireshark's display filtering language, read the Building display filter expressions page in the official Wireshark documentation.



Another interesting thing you can do is right-click a packet and select Follow > TCP Stream. You'll see the full TCP conversation between the client and the server. You can also click other protocols in the Follow menu to see the full conversations for other protocols, if applicable.

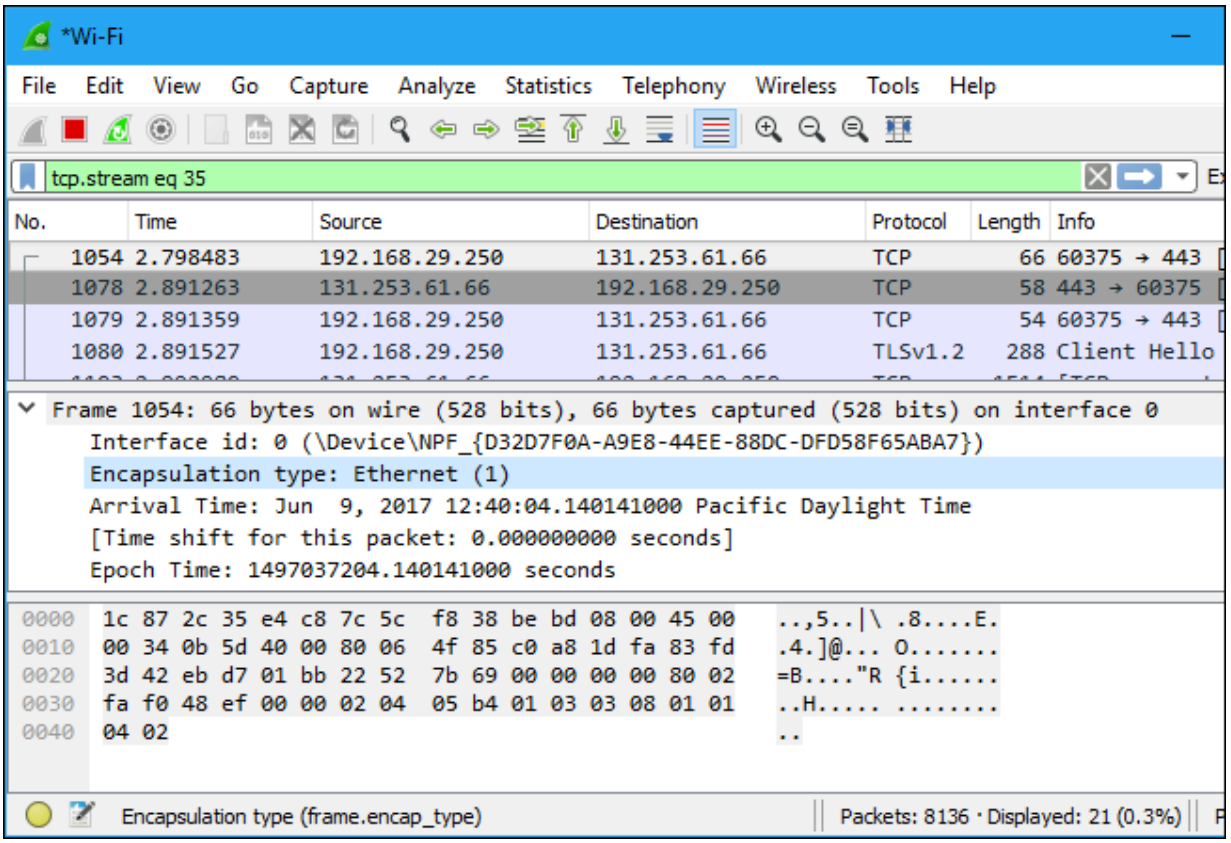


Close the window and you'll find a filter has been applied automatically. Wireshark is showing you the packets that make up the conversation.

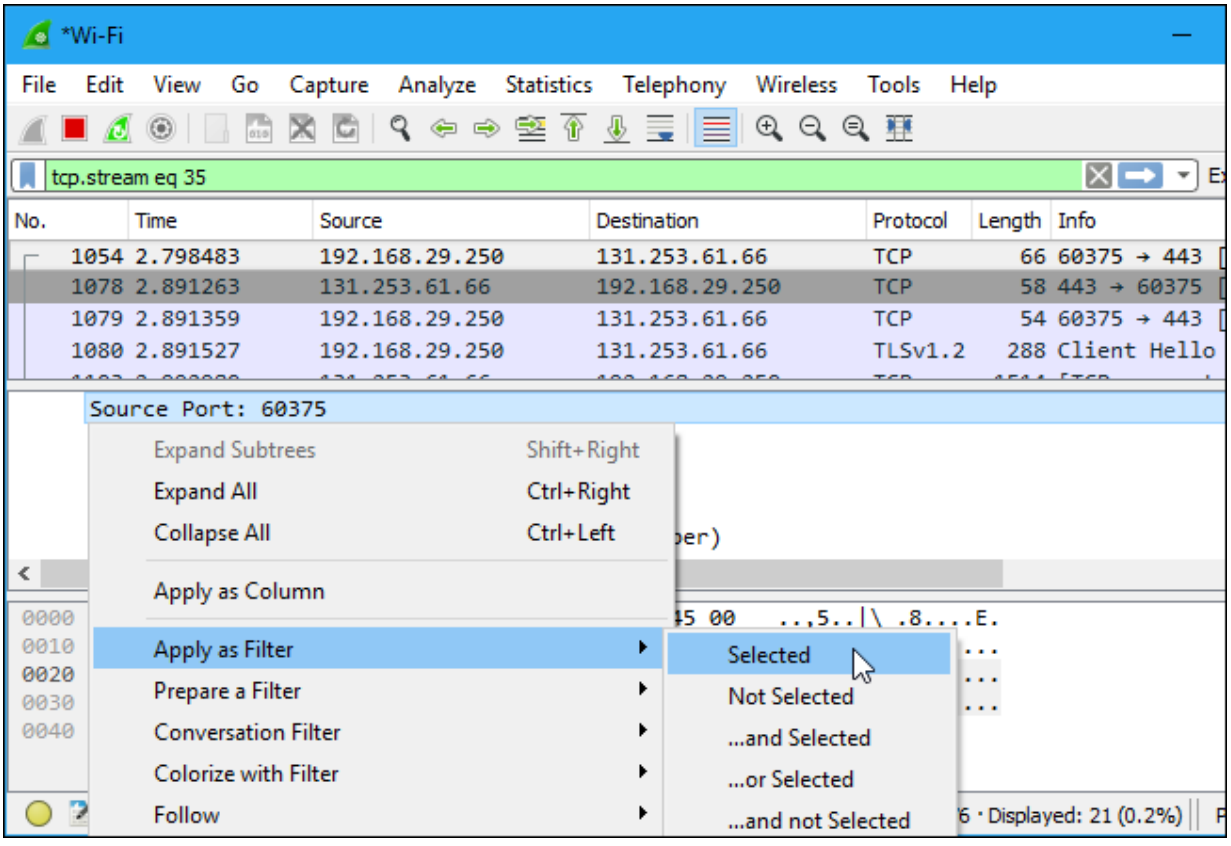


Inspecting Packets

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

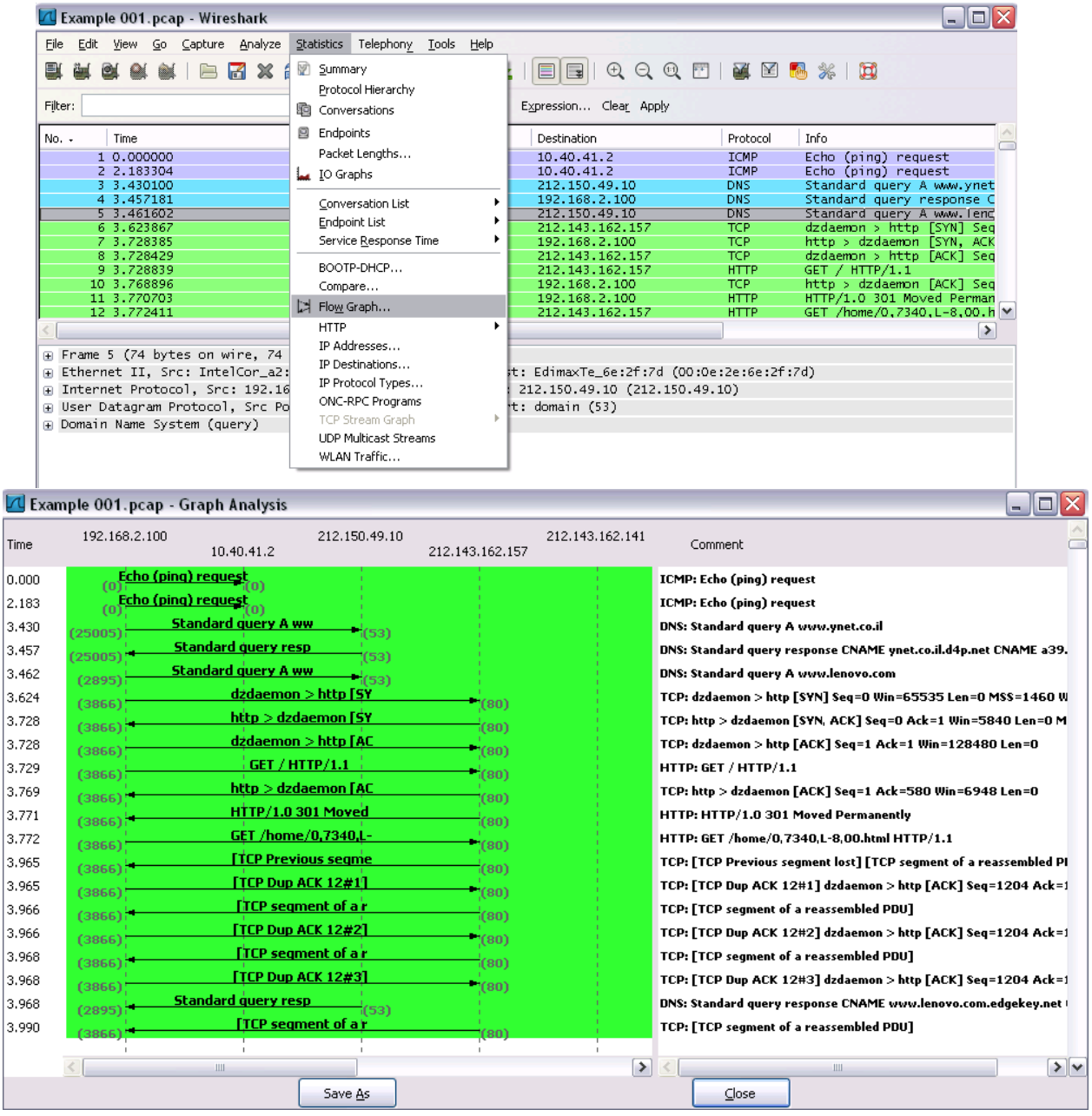


You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



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

Flow Graph: Gives a better understanding of what we see.



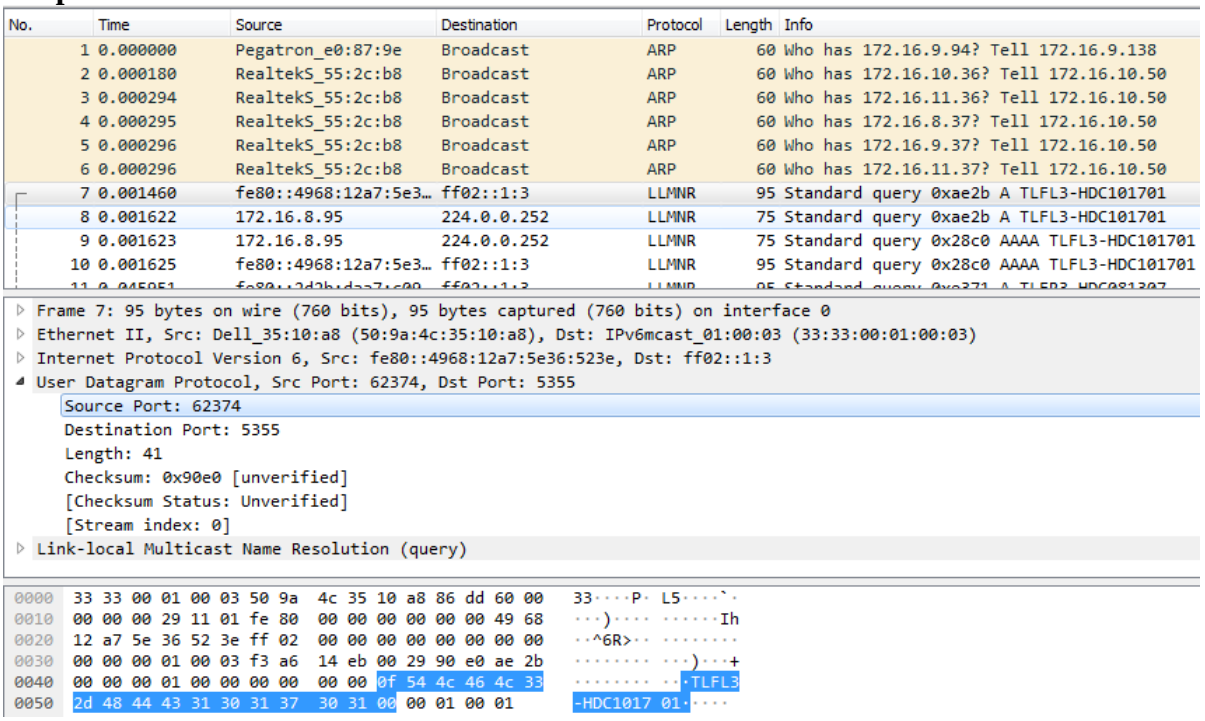
CAPTURING AND ANALYSING PACKETS USING WIRESHARK TOOL

To filter, capture, 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.

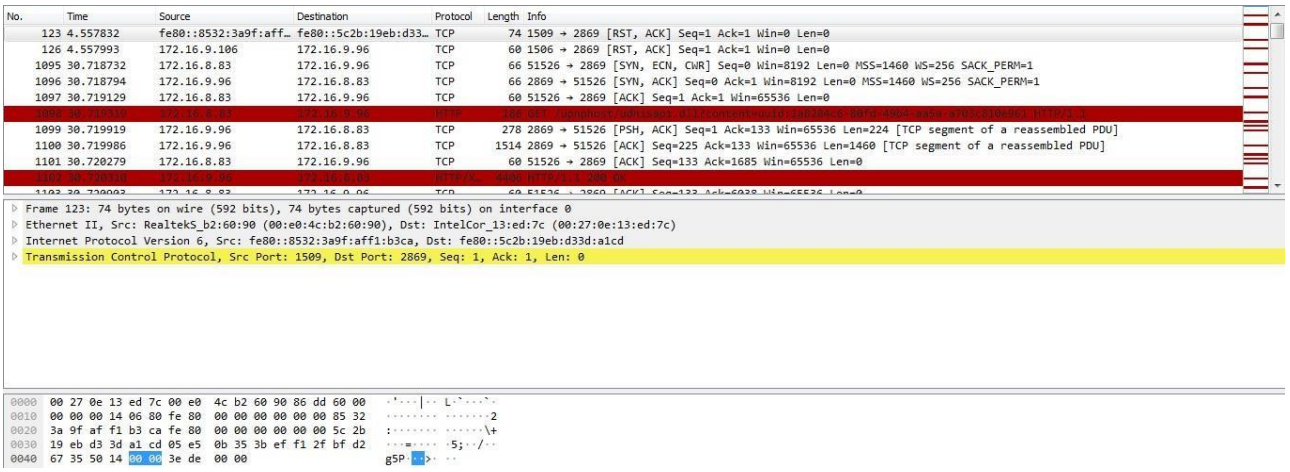
Output



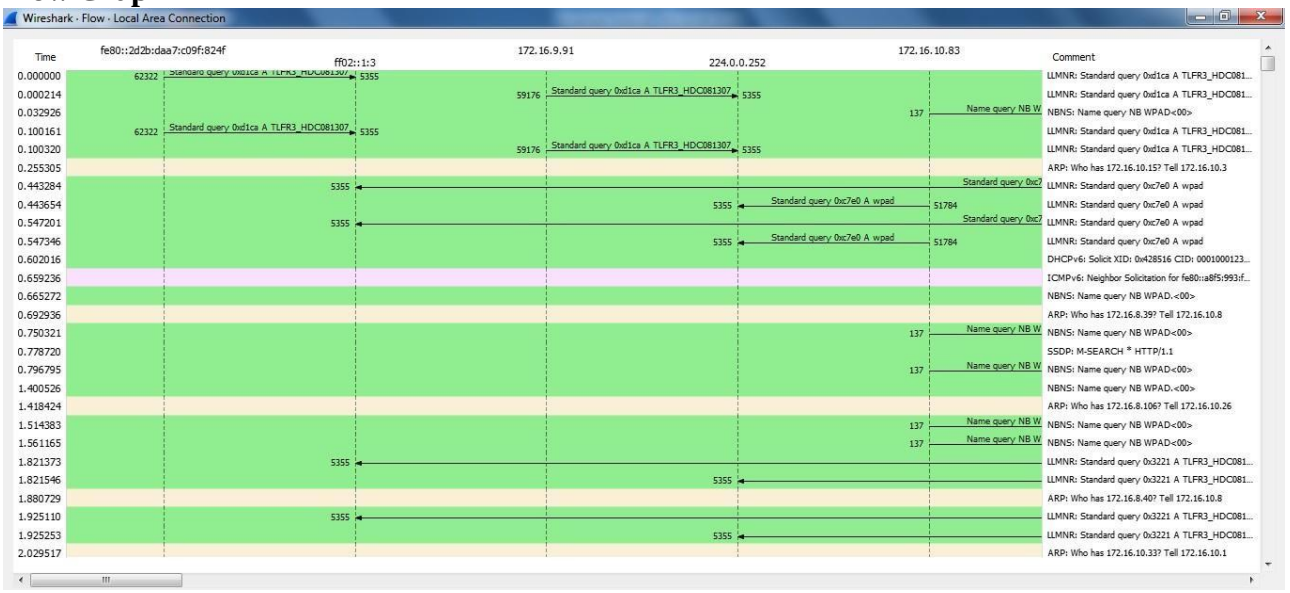
1. **Create a Filter to display only TCP/UDP packets, inspect the packets and provide the flow graph**

Procedure

- Select Local Area Connection in Wireshark.
- Go to capture → option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search TCP packets in search bar.
- To see flow graph click Statistics→Flow graph.
- Save the packets.



Flow Graph

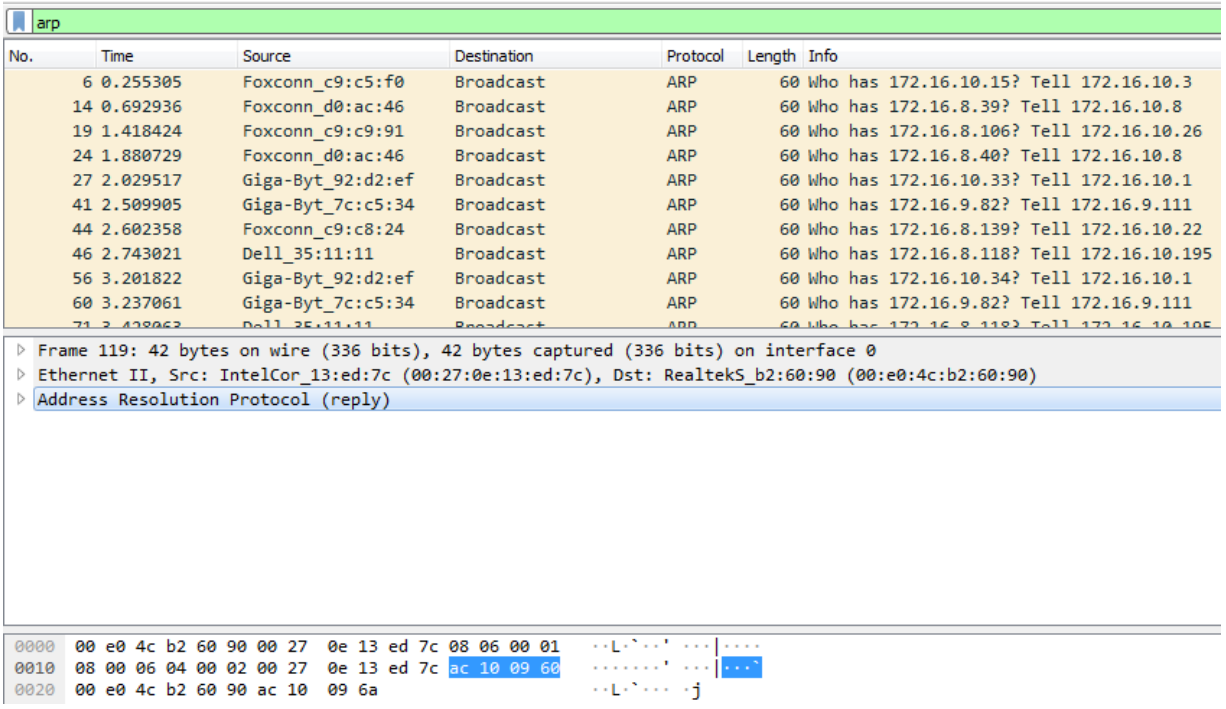


2. Create a Filter to display only ARP packets and inspect the packets.

Procedure

- Go to capture → option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search ARP packets in search bar.
- Save the packets.

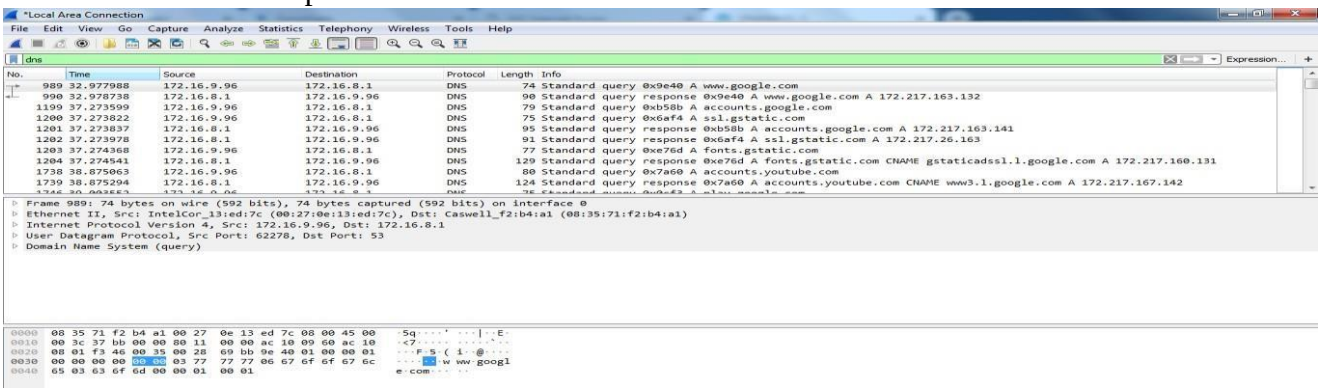
Output

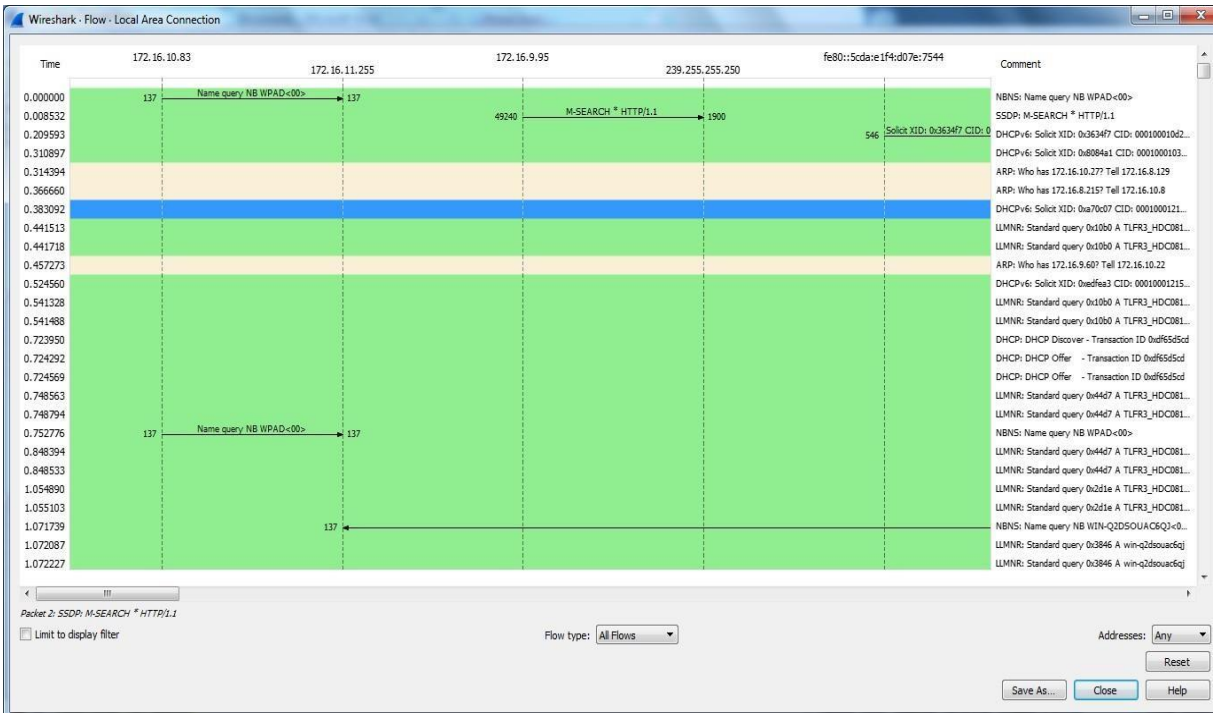


3. Create a Filter to display only DNS packets and provide the flow graph.

Procedure

- Go to capture → option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search DNS packets in search bar.
- To see flow graph click Statistics→Flow graph.
- Save the packets.

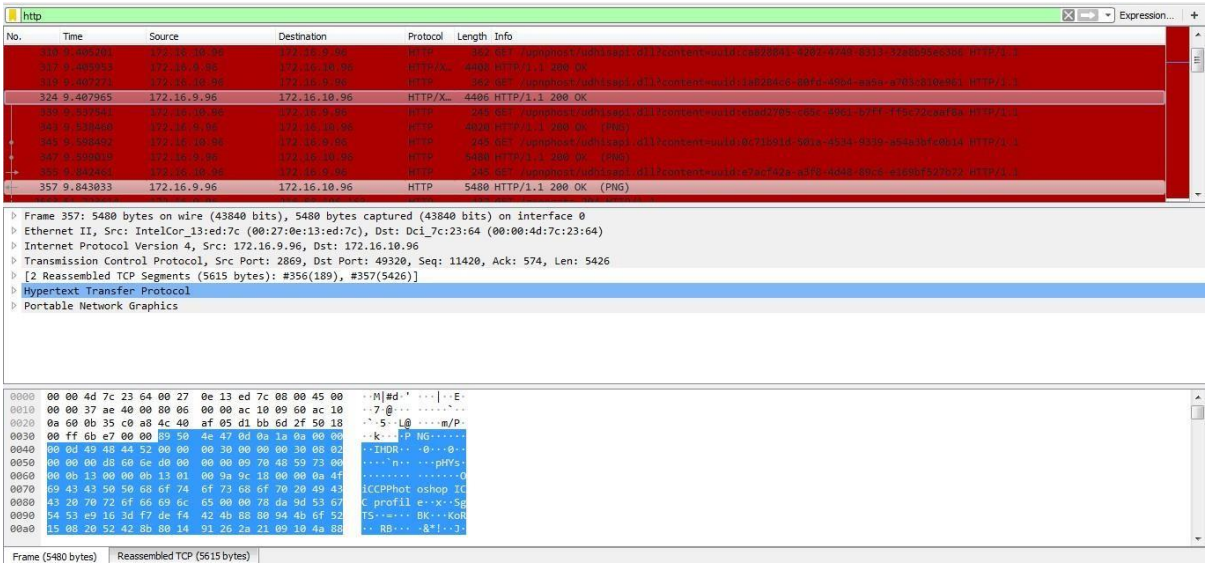




4. Create a Filter to display only HTTP packets and inspect the packets

Procedure

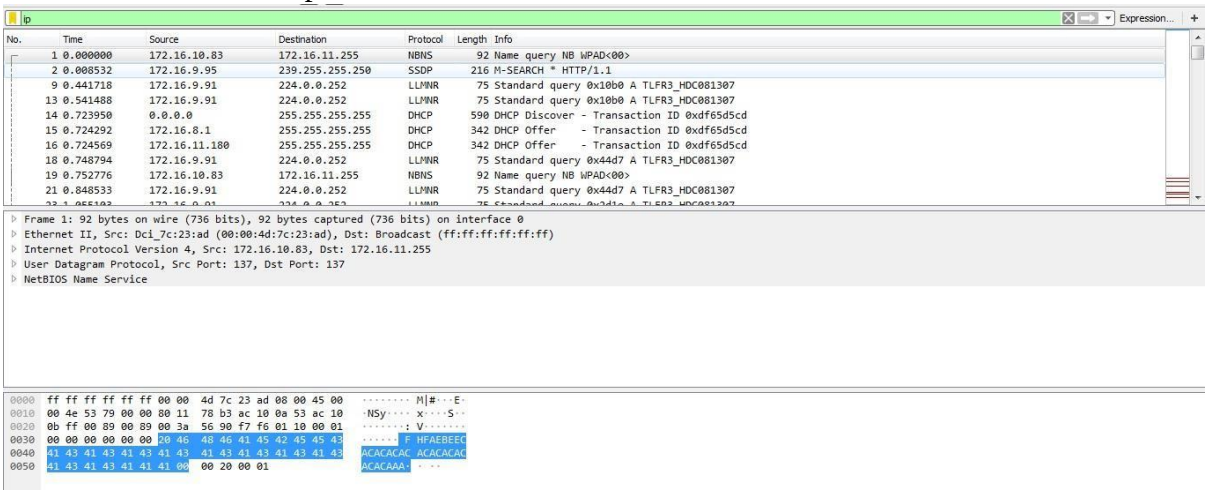
- Select Local Area Connection in Wireshark.
- Go to capture → option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search HTTP packets in search bar.
- Save the packets.



5. Create a Filter to display only IP/ICMP packets and inspect the packets.

Procedure

- Select Local Area Connection in Wireshark.
- Go to capture → option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search ICMP/IP packets in search bar.
- Save the packets

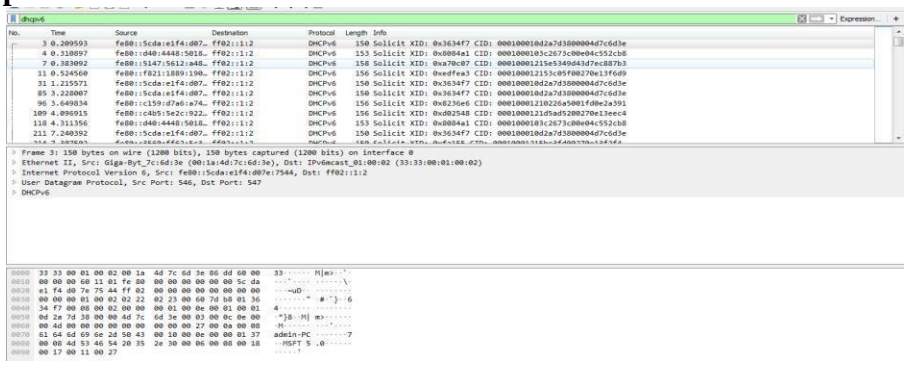


6. Create a Filter to display only DHCP packets and inspect the packets.

Procedure

- Select Local Area Connection in Wireshark.
- Go to capture → option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search DHCP packets in search bar.
- Save the packets

Output



Student observation:

1. **What is promiscuous mode?**
2. **Does ARP packets has transport layer header? Explain.**
3. **Which transport layer protocol is used by DNS?**
4. **What is the port number used by http protocol?**
5. **What is a broadcast ip address?**

1. What is promiscuous mode?
   Promiscuous mode is a network interface card (NIC) setting where the NIC passes all packets it receives to the computer's network stack, regardless of the destination MAC address. Normally, a NIC only processes frames addressed directly to it, broadcast addresses, or multicast addresses. When switched to promiscuous mode, it intercepts all network traffic on the segment, which is particularly useful for network monitoring, intrusion detection, and packet sniffing. It enables network administrators and security professionals to analyze traffic in detail, but it also poses security risks if misused, because it exposes all network data passing through the interface. Some operating systems and hardware require administrative privileges to enable this mode, and in switched Ethernet networks, additional configurations like port mirroring are often necessary to view all traffic.

2. Does ARP packets have transport layer header? Explain.
   ARP (Address Resolution Protocol) packets do not contain transport layer headers because ARP operates directly on the network layer (Layer 3) with a specific payload structure. ARP is used to map IP addresses to MAC addresses so that subsequent data packets can be delivered correctly on the local network. Its message structure includes hardware type, protocol type, hardware and protocol addresses, and the operation type (request or reply), but it doesn't utilize protocols from the TCP or UDP transport layers, which reside above the network layer in the OSI model. Therefore, ARP packets are independent of the transport layer and do not have transport layer headers like TCP or UDP headers.

3. Which transport layer protocol is used by DNS?
   The Domain Name System (DNS) primarily uses the User Datagram Protocol (UDP) at the transport layer. DNS queries and responses commonly operate on UDP port 53 because it provides a fast, connectionless method suitable for the typically small query/response message size. However, DNS can also use TCP (also on port 53) when the response data exceeds DNS over UDP's size limit or if reliable transmission is needed, such as during zone transfers between DNS servers.

4. What is the port number used by HTTP protocol?
   The HTTP protocol traditionally uses port number 80. When a client sends a request to a web server without specifying a port, it defaults to port 80 for HTTP traffic. In secure scenarios, HTTPS (HTTP over SSL/TLS) uses port 443. These well-known port numbers are standardized and recognized universally for web traffic.

5. What is a broadcast IP address?
   A broadcast IP address is a special IP address used to send data to all hosts within a specific subnet simultaneously. It allows a device to transmit a message that all other devices in the subnet will receive. The broadcast address is derived by taking the subnet's network address and setting all host bits to 1. For example, in a subnet with network address 192.168.1.0 and subnet mask 255.255.255.0 (/24), the broadcast address would be 192.168.1.255. This address ensures that any packet sent to it over the network will be received by all hosts within that subnet, making it useful for network-wide announcements or management packets.