NETWORK TOOLS

As you are somewhat familiar with the socket programming now and have learned to build various kind of network topologies along with different network protocols using NS-3. Merely creating the networks is not a sole purpose rather we should also know that what's exactly happening in the background. Therefore, next step is to move towards the analysis of those networks. Here comes the role of the "computer networking tools" which is any type of software that lends a hand in the design, creation, monitoring, maintenance, and troubleshooting of networks. These tools play a crucial role in ensuring the efficient and secure operation of networks. Monitoring a network relies heavily on real-time information to provide value. Essential information is harvested using a variety of methods and protocols.

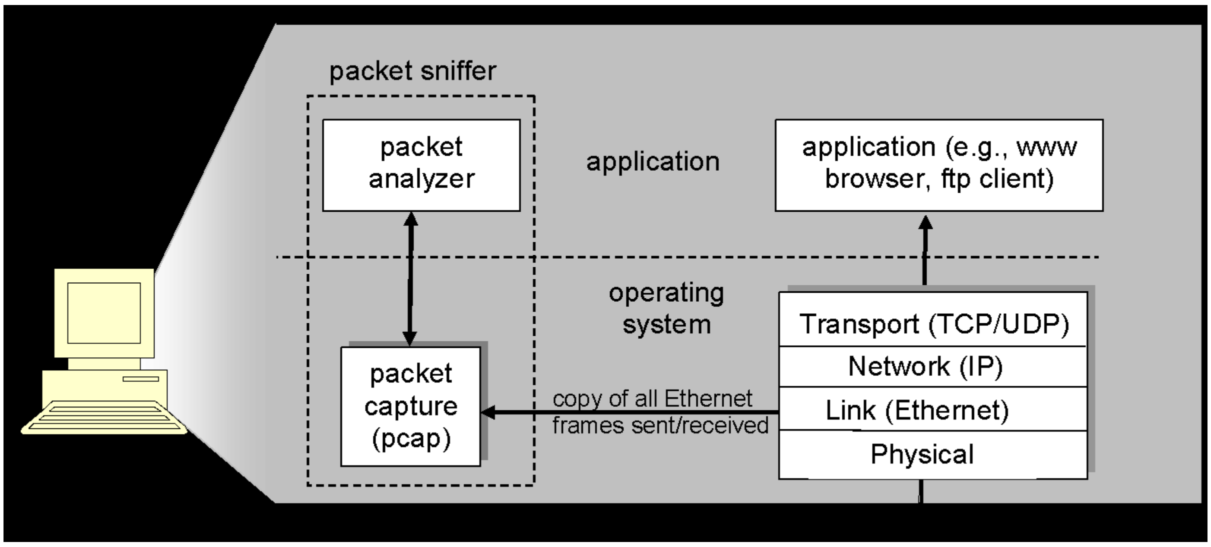
Some of the key advantages of using network tools are:

- **Troubleshooting and Problem Resolution**: Network tools help identify and diagnose network issues quickly helping administrators' pinpoint problems like connectivity issues, bottlenecks or faulty devices.
- **Performance monitoring**: Monitoring the performance of network devices, servers and applications helps in optimizing resource allocation identifying problems related to the performance and hence ensuring efficient network operation.
- **Security Analysis and Threat Detection**: Security-focused tools contribute in detecting and preventing security threats. They monitor network traffic or signs of malicious activity, unauthorized access, or potential vulnerabilities.
- **Bandwidth Management**: Monitoring and controlling the usage of network bandwidth is crucial for ensuring fair distribution of resources, enhancing performance and preventing network congestion.
- **Remote Administration**: Remote desktop software provides secure methods for administrator to access and manage remote systems. This is especially valuable for troubleshooting and performing administrative tasks on servers or devices located in different physical locations.
- **File Transfer and Collaboration**: These tools enable efficient and secure transfer of files between local machines and remote servers. This is important for collaborative work, especially in scenarios where large files or datasets need to be exchanged.
- **Network Planning and Design**: Network tools aid in planning and designing networks by providing insights into traffic patterns, resource utilization and potential points of failure which helps in creating robust and scalable network architectures.
- **Documentation and Reporting**: Many tools generate reports and logs that help in documenting network activity, these reports can be valuable for compliance, auditing and future planning.
- **Time and Cost Savings**: The efficient use of network tools can save time by automating tasks, reducing manual effort and minimizing downtime, which in turn contributes to cost savings and improved overall productivity.

In this course we will look into some of the Network tools like tcpdump, Wireshark and Filezilla.

The understanding of network protocols can be enhanced by actually "seeing protocols in action" and by "playing around with protocols" – observing and analyzing the sequence of messages exchanged between two protocol entities, digging into the details of protocol operation, and causing protocols to perform certain actions and then observing these actions and their consequences. Such things can be done either in simulated scenarios or in a "real" network environment like the Internet.

The basic tool for observing the messages exchanged between executing protocol entities is called a **packet sniffer**. A packet sniffer captures ("sniffs") messages being sent/received from/by your computer; it will typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer observes the messages being sent and received by applications and protocols running on your computer, but never will it send a packet by itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent/received from/by application and protocols executing on your machine.



The figure above the structure of a packet sniffer. At the right part of the figure, there are protocols (Internet protocols) and applications (web browser or ftp client). The packet sniffer, shown within the dashed rectangle in Figure is an addition to the usual software in your computer, and consists of two parts. The **packet capture library** receives a copy of every link-layer frame that is sent from or received by your computer. The messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In this figure, the assumed physical media is Ethernet, and so all upper layer protocols are eventually encapsulated within a Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/received from/by all protocols and applications executing in your computer.

The second component of a packet sniffer is the **packet analyzer**, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must "understand" the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in the figure. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Also, it understands the HTTP protocol and so, like, it knows that the first bytes of an HTTP message will contain the string "GET," "POST," or "HEAD".

Unveiling the Network: A Look at tcpdump and Wireshark.

The digital world thrives on the constant flow of information, carried by packets traversing the intricate web of networks. To understand and troubleshoot this unseen traffic, network professionals rely on specialized tools like tcpdump and wireshark; we'll be looking into and perform a couple of exercises related to these both in this lab course. Both are one of the popular tools used for capturing and analyzing network traffic, but they differ in their interface, capabilities and use cases.

Tcpdump

Tcpdump is an open source and widely used command-line data-network packet sniffing and analysing tool for Unix-like operating systems, including Linux and macOS. It allows the system administrator to capture and analyze network traffic in real-time or to save the captured packets to a file for later analysis. It is one of the powerful tools for troubleshooting network issues, provides a low-level view of network communication helps in diagnosing network-related problems and inspecting the characteristics of network traffic. While powerful, tcpdump's interface might seem daunting for beginners. It requires typing commands to capture, filter and analyze network traffic.

Some of the advantages of using tcpdump:

- **Lightweight and efficient**: Runs smoothly even on resource- constrained systems.
- **Flexible filtering**: Lets you specify criteria to capture only relevant packets, saving storage space and analysis time.
- **Scriptable**: Can be integrated into scripts for automated tasks.

Key features and functionalities of 'tcpdump' include:

1. **Packet Capture**: 'tcpdump' captures packets from a network interface, allowing users to view and analyze the details of each packet. This includes information such as source and destination IP addresses, protocol types, flags and payload data.

2. **Filtering**: Users can apply filters to capture specific types of traffic based on criteria such as source or destination IP addresses, port numbers, or specific protocols. This enables focused analysis on particular aspects of network communication.
3. **Display Formats**: ‘tcpdump’ can display packet information in various formats, including human-readable, hexadecimal, and machine-readable formats. This flexibility makes it suitable for a wide range of users, from network administrators to security professionals.
4. **Real-time and Offline Analysis**: ‘tcpdump’ can capture packets in real-time as they traverse the network, providing immediate insights into network activity. Additionally, it can read packets from a saved capture file for offline analysis.
5. **Promiscuous Mode**: ‘tcpdump’ can be used in promiscuous mode, allowing the capture of all packets on a network segment, regardless of whether they are destined for the capturing machine. This is useful for analyzing overall network traffic.
6. **Integration with Other Tools**: Captured packet data can be saved in the pcap format, which is a standard format for packet capture file. This allows users to analyze captured data with other tools also for an in-depth analysis and visualization.

Usage example:

Type the following commands in the terminal.

Tcpdump continues to capture packets until it receives an interrupt signal. Use ctrl+C to interrupt.

Capture packets on the eth0 interface:

```
$ tcpdump -i eth0
```

Capture HTTP traffic on eth0 without resolving hostnames:

```
$ tcpdump -n -i eth0 port 80
```

To see which interfaces are available for capture, use the command:

```
$ sudo tcpdump –list-interfaces
```

Or

```
$ sudo tcpdump -D
```

Output:

```
1.eth0 [Up, Running]

2.lo [Up, Running, Loopback]

3.any (Pseudo-device that captures on all interfaces) [Up, Running]

4.lxcbr0 [Up]
```

5.bluetooth-monitor (Bluetooth Linux Monitor) [none]

6.nflog (Linux netfilter log (NFLOG) interface) [none]

7.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]

To capture all packets in any interface:

```
$ sudo tcpdump --interface any
```

To limit the number of packets captured and stop tcpdump, use -c:

```
$ sudo tcpdump -i any -c 5
```

where c is for count.

Tcpdump can capture and decode many different protocols like TCP, UDP, ICMP and many more.

Various logical and Boolean operations can also be performed for the packet matching and filtering purposes.

For further details, type the command below on the terminal:

```
$ sudo man tcpdump
```

Essentially it will open the manual for tcpdump.

Understanding the output format:

```
16:13:28.476112 IP 152.195.38.76.80 > 172.16.5.137.38592: Flags [P.], seq 1:738, ack 425, win 131, length 737
```

The fields may differ depending on the type of packet being sent, but this is the general format.

The first field, 16:13:28.476112 represents the timestamp of the received packet as per the local clock.

The IP represents the network layer protocol here IPv4. For IPv6 the value is IP6.

The next field, 152.195.38.76.80 is the source IP address and port, which is followed by the destination IP address and port, represented by 172.16.5.137.38592.

After the source and destination, you will find the TCP flags Flags [.]

Typical values for this field include:

Value	Flag Type	Description
S	SYN	Connection Start
F	FIN	Connection Finish
P	PUSH	Data push
R	RST	Connection reset
.	ACK	Acknowledgement

This field can also be a combination of other values, such as [S.] for a SYN-ACK packet.

Next is the sequence number of the data contained in the packet. For the first packet captured, this is an absolute number. Subsequent packets use a relative number to make it easier to follow. Here the sequence is seq 1:738, which contains bytes 1 to 738 of this flow.

This is followed by the acknowledgement number: ack 425.

The next field is the window size win 131 which represents the number of bytes available in the receiving buffer, followed by TCP options like Maximum Segment Size or Window Scale.

Finally, the packet length, length 737, which represents the length, in bytes, of the payload data. The length is the difference between the last and first bytes in the sequence number.

(Note: There may be very minor variations in the format of the output of tcpdump depending on the version of tcpdump on your machine.)

Wireshark

Wireshark emerges as a powerful tool for network professionals, enthusiasts, and anyone curious about the inner workings of data communication. This free and open-source packet analyzer allows you to capture, analyze, and understand the digital conversations happening on your network. A network packet analyzer presents captured packet data in as much detail as possible. You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable (but at a higher level, of course).

Here are some reasons to use Wireshark:

- Network administrators use it to *troubleshoot network problems*
- Network security engineers use it to *examine security problems*
- QA engineers use it to *verify network applications*
- Developers use it to *debug protocol implementations*
- People use it to *learn network protocol* internals

What can Wireshark do?

Wireshark offers a comprehensive suite of features to unlock the mysteries of network traffic:

- **Capture live traffic:** Witness the real-time flow of packets across your network interface, like watching a live stream of data packets.

- **Import captured files:** Analyze previously captured traffic stored in various file formats.
- **Deep packet inspection:** Deconstruct each packet, revealing its individual layers (header and data) and their specific details, allowing you to understand the communication process at a granular level.
- **Protocol dissection:** Wireshark understands and decodes various protocols like HTTP, TCP/IP, and DNS, providing in-depth information about the data being exchanged.
- **Powerful filtering:** Narrow down captured traffic to specific criteria, focusing only on relevant packets based on source/destination IP address, port number, protocol, or other characteristics.
- **Visualize the flow:** Wireshark presents captured packets in a structured and color-coded format, making it easier to identify different types of traffic and follow the conversation flow between devices.
- **Search and statistics:** Quickly find specific packets based on keywords or filter criteria, and gain insights into overall network activity through statistics.
- **Customization and extensibility:** Wireshark can be customized through plugins and capture filters to cater to specific needs and extend its functionalities.

Benefits of using Wireshark:

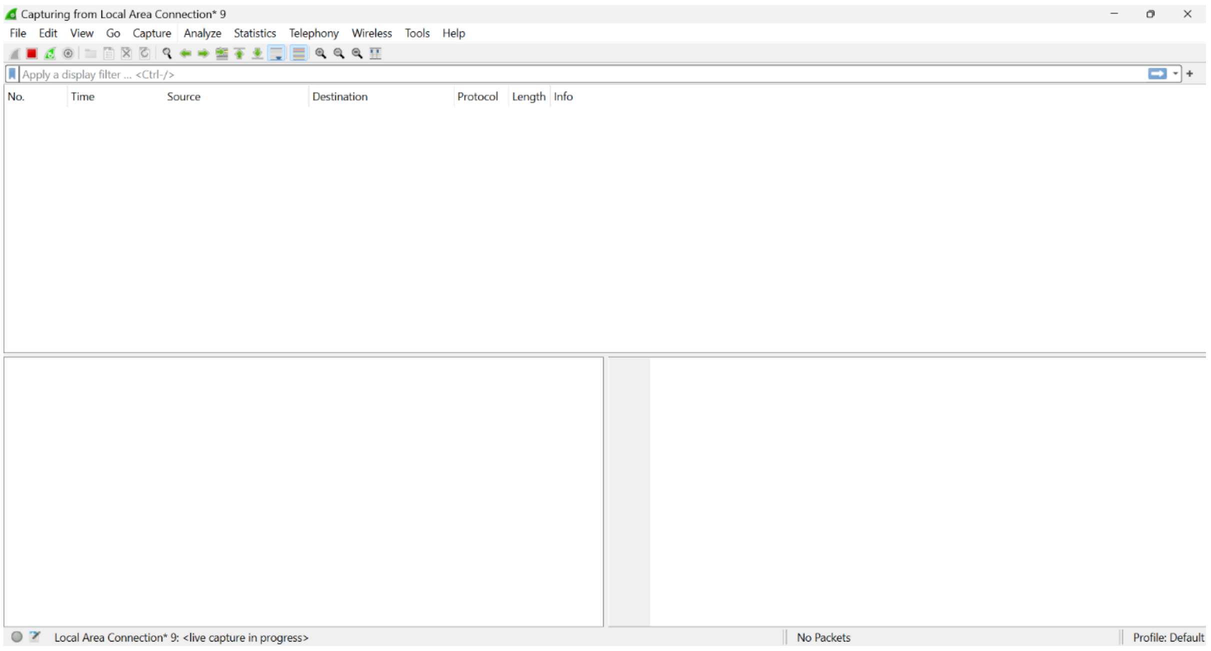
- **Troubleshooting network issues:** Identify bottlenecks, diagnose connectivity problems, and pinpoint the source of performance issues.
- **Security analysis:** Investigate suspicious activity, identify potential vulnerabilities, and monitor network security protocols.
- **Software and protocol development:** Gain insights into how applications communicate over the network, aiding in software development and protocol testing.
- **Education and learning:** Understand how network protocols work, visualize data exchange, and gain valuable knowledge about network communication.

Getting started with Wireshark:

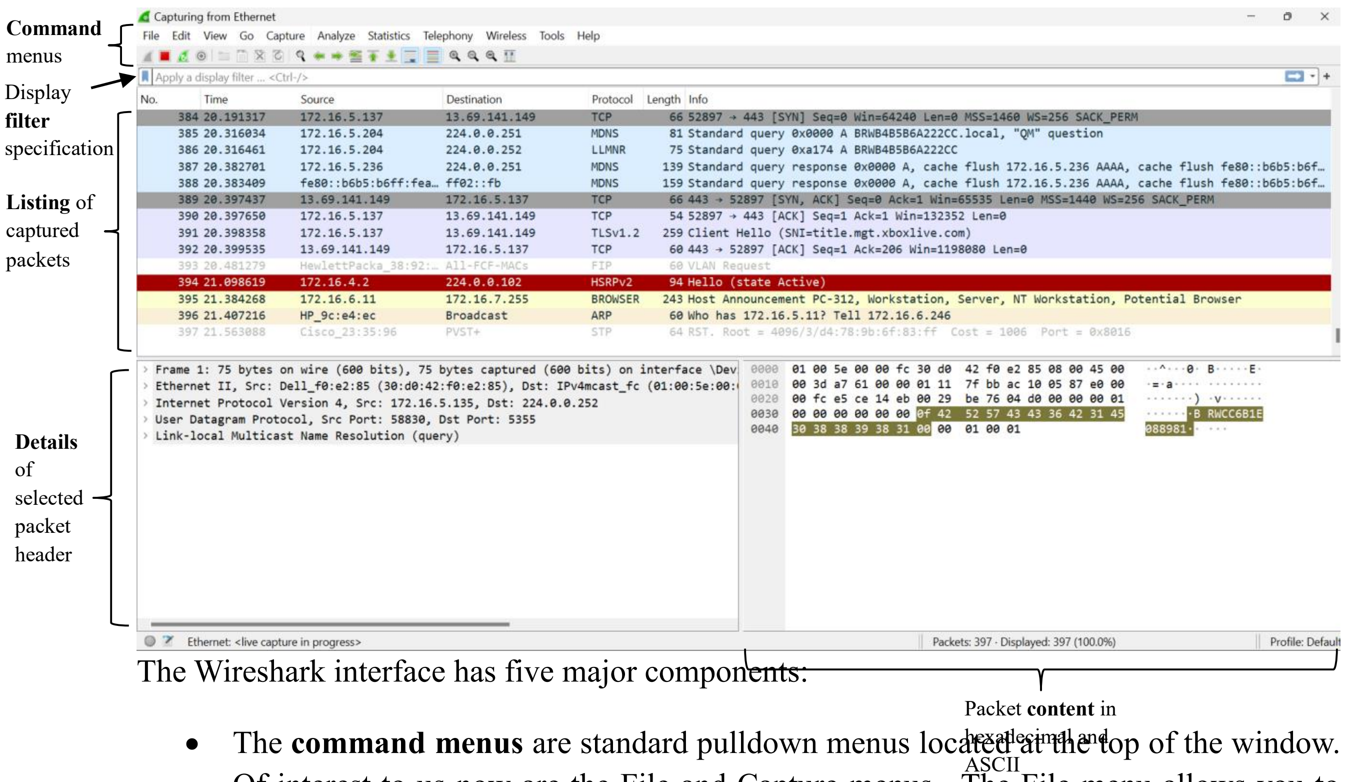
Wireshark is available for download on various platforms including Windows, macOS, and Linux. While it offers a user-friendly interface, understanding network protocols is crucial for interpreting the captured data effectively. Numerous online resources, tutorials, and communities can help you navigate the world of Wireshark and unlock its full potential.

Running Wireshark:

When you run the Wireshark program, the Wireshark GUI will open and initially, no data will be displayed in the various windows.


Capturing from Local Area Connection* 9
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
Local Area Connection* 9: <live capture in progress>
No Packets
Profile: Default

Components in Wireshark Graphical User Interface


Command menus
Display filter specification
Listing of captured packets
Details of selected packet header
Packet content in hexadecimal and ASCII
Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Ethernet: <live capture in progress>
Packets: 397 · Displayed: 397 (100.0%)
Profile: Default

```
No.  Time       Source              Destination     Protocol Length Info
 384 20.191317  172.16.5.137        13.69.141.149   TCP       66 52897 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
 385 20.316034  172.16.5.204        224.0.0.251     MDNS      81 Standard query 0x0000 A BRWB4B5B6A222CC.local, "QM" question
 386 20.316461  172.16.5.204        224.0.0.252     LLMNR     75 Standard query 0xa174 A BRWB4B5B6A222CC
 387 20.382701  172.16.5.236        224.0.0.251     MDNS     139 Standard query response 0x0000 A, cache flush 172.16.5.236 AAAA, cache flush fe80::b6b5:b6f…
 388 20.383409  fe80::b6b5:b6ff:fea… ff02::fb       MDNS     159 Standard query response 0x0000 A, cache flush 172.16.5.236 AAAA, cache flush fe80::b6b5:b6f…
 389 20.397437  13.69.141.149       172.16.5.137    TCP       66 443 → 52897 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
 390 20.397650  172.16.5.137        13.69.141.149   TCP       54 52897 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0
 391 20.398358  172.16.5.137        13.69.141.149   TLSv1.2  259 Client Hello (SNI=title.mgt.xboxlive.com)
 392 20.399535  13.69.141.149       172.16.5.137    TCP       60 443 → 52897 [ACK] Seq=1 Ack=206 Win=1198080 Len=0
 393 20.481279  HewlettPacka_38:92:… All-FCF-MACs   FIP       60 VLAN Request
 394 21.098619  172.16.4.2          224.0.0.102     HSRPv2    94 Hello (state Active)
 395 21.384268  172.16.6.11         172.16.7.255    BROWSER  243 Host Announcement PC-312, Workstation, Server, NT Workstation, Potential Browser
 396 21.407216  HP_9c:e4:ec         Broadcast       ARP       60 Who has 172.16.5.11? Tell 172.16.6.246
 397 21.563088  Cisco_23:35:96      PVST+           STP       64 RST. Root = 4096/3/d4:78:9b:6f:83:ff  Cost = 1006  Port = 0x8016
```

```
> Frame 1: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \Dev
> Ethernet II, Src: Dell_f0:e2:85 (30:d0:42:f0:e2:85), Dst: IPv4mcast_fc (01:00:5e:00:
> Internet Protocol Version 4, Src: 172.16.5.135, Dst: 224.0.0.252
> User Datagram Protocol, Src Port: 58830, Dst Port: 5355
> Link-local Multicast Name Resolution (query)
```

```
0000  01 00 5e 00 00 fc 30 d0  42 f0 e2 85 08 00 45 00   ··^···0· B·····E·
0010  00 3d a7 61 00 00 01 11  7f bb ac 10 05 87 e0 00   ·=·a···· ········
0020  00 fc e5 ce 14 eb 00 29  be 76 04 d0 00 00 00 01   ·······) ·v······
0030  00 00 00 00 00 00 0f 42  52 57 43 43 36 42 31 45   ·······B RWCC6B1E
0040  30 38 38 39 38 31 00 00  01 00 01                  088981·· ···
```

The Wireshark interface has five major components:

- The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.

- In the **packet display filter field**, a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows).
- The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.
- The **packet-header details window** provides details about the packet selected (highlighted) in the packet listing window (To select a packet in the packet listing window, place the cursor over the packet's one-line summary in the packet listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the plus-or-minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.
- The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

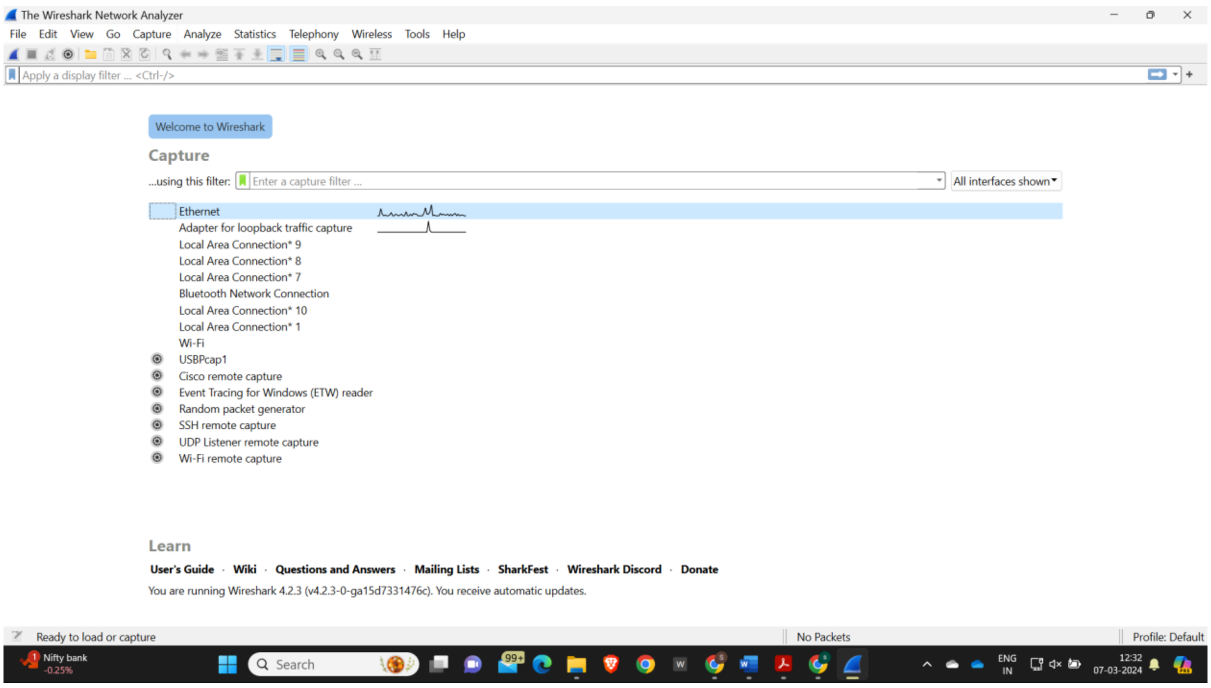
Test Run in Wireshark:

So, after getting familiar with the components and fields used in wireshark, the best way to learn about any new software is to try it out! Assuming that your computer is connected to the Internet via a wired Ethernet interface. Do the following:

1. Start up a web browser, which will display the selected homepage.
2. Start up the Wireshark software. You may initially see a window where you can select the network interface or if not select the capture from the menu above.
3. To begin packet capture, after selecting the capture a pull-down menu will be displayed and select the *options*. This will cause the "Wireshark: Capture Options" window to be displayed.
4. In case your computer has more than one active network interface (e.g., if you have both a wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets (mostly likely the wired interface). After selecting the network interface (or using the default interface chosen by

Wireshark) the packet capture will begin now- all packets being sent/received from/by your computer are now being captured.

5. While wireshark is running, enter the URL: https://cse.iitism.ac.in/ and have that displayed in your browser. In order to display this page, your browser will contact the HTTP server at cse.iitism.ac.in and exchange HTTP messages with the server. The Ethernet frames containing these HTTP messages will be captured by Wireshark.
6. After your browser has loaded the home page, stop wireshark packet capture by selecting stop in the Wireshark capture window. This will display all packets captured and now there are live packet data that contains all protocol messages exchanged between your computer and other entities. The HTTP message exchanges with the cse.iitism.ac.in web server should appear somewhere in the listing of packets captured. But there will also be many other types of packets displayed as well shown in the *Protocol* column.
7. Type in “http” (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select Apply (to the right of where you entered “http”). This will cause only HTTP message to be displayed in the packet-listing window.
8. Select the first http message shown in the packet-listing window. This should be the HTTP GET message that was sent from your computer to the cse.iitism.ac.in HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window. By clicking plus and minus boxes to the left side of the packet details window, minimize the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. Maximize the amount information displayed about the HTTP protocol.
9. Exit Wireshark.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Search
Nifty bank
-0.25%
ENG
IN
12:32
07-03-2024

Capture

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

Ethernet
Adapter for loopback traffic capture
Local Area Connection* 9
Local Area Connection* 8
Local Area Connection* 7
Bluetooth Network Connection
Local Area Connection* 10
Local Area Connection* 1
Wi-Fi
USBPcap1
Cisco remote capture
Event Tracing for Windows (ETW) reader
Random packet generator
SSH remote capture
UDP Listener remote capture
Wi-Fi remote capture

Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

You are running Wireshark 4.2.3 (v4.2.3-0-ga15d7331476c). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default

References:

1. https://www.tcpdump.org/manpages/tcpdump.1.html
2. https://opensource.com/article/18/10/introduction-tcpdump
3. http://www.alexonlinux.com/tcpdump-for-dummies#introduction
4. https://www.wireshark.org/docs/
5. https://wiki.wireshark.org/