Computer Networks-Lab 08

Instructor: Hurmat Hidayat

CL30001 – Computer Networks-Lab

SEMESTER Fall 2022

NATIONAL UNIVERSTIY OF COMPUTER AND EMERGING SCIENCES, FAST- PESHAWAR CAMPUS
Department of Computer Science & Software Engineering

Computer Networks - Lab 08

OBJECTIVES

After these Lab students shall be able to perform

- Introduction with Wire shark
- Wireshark User Interface
- Relation OSI and TCP/IP model
- OSI and TCP/IP Layer Analysis via Wireshark
- HTTP analysis using Wireshark

PRE-LAB READING ASSIGNMENT

Remember the delivered lecture carefully.

Computer Networks Lab 08

Table of Contents

Computer Networks Lab 08

Introduction to Wireshark

Wireshark is a network packet analyzer. 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).
In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, that has changed. Wireshark is available for free, is open source, and is one of the best packet analyzers available today.

Some intended purposes

Here are some reasons people 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

Wireshark can also be helpful in many other situations.

Features

The following are some of the many features Wireshark provides:

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and many
- other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- 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.

…and a lot more!
However, to really appreciate its power you have to start using it. Wireshark captures packets and lets you examine their contents. shows Wireshark having captured some packets and waiting for you to examine them.

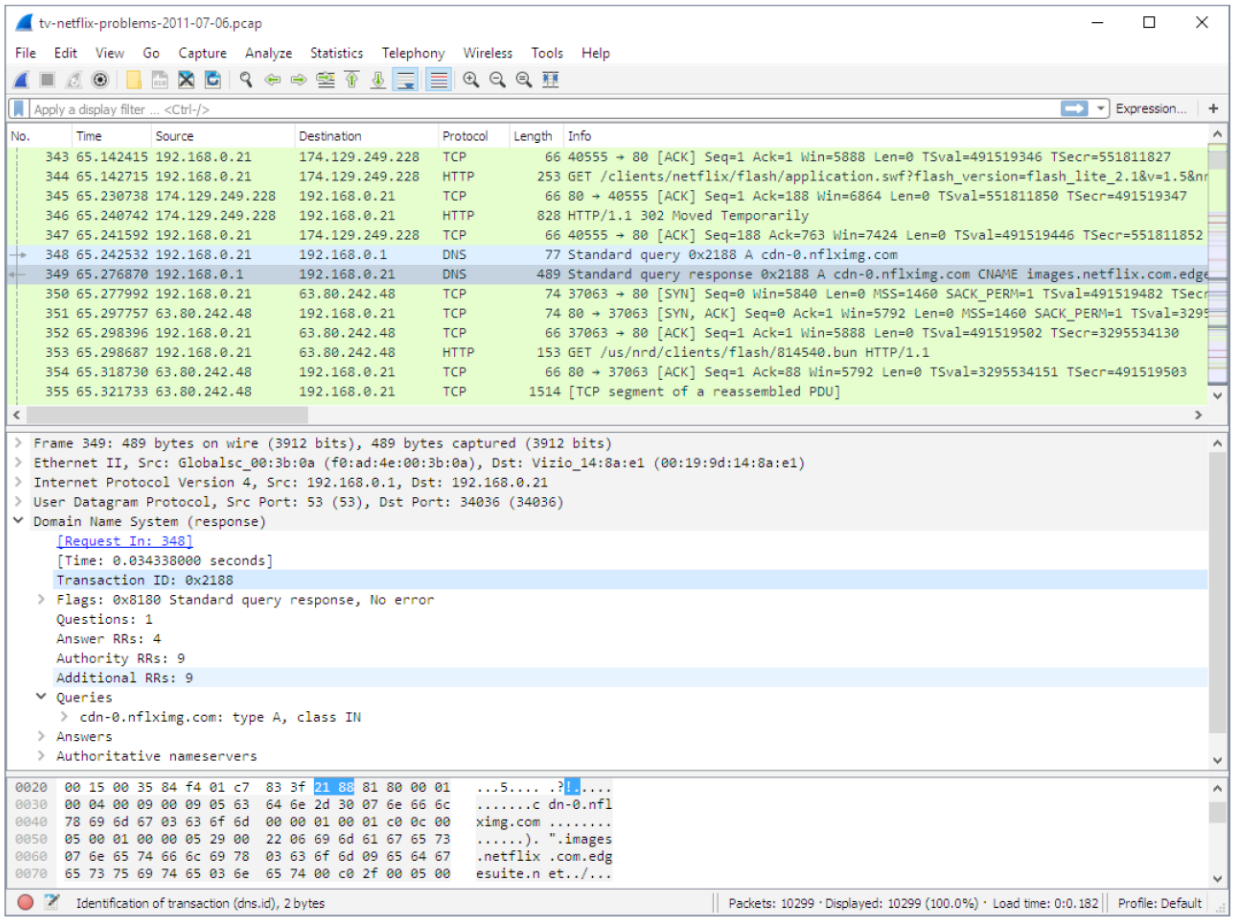


Figure 1. Wireshark captures packets and lets you examine their contents.

Live capture from many different network media

Wireshark can capture traffic from many different network media types, including Ethernet, Wireless LAN, Bluetooth, USB, and more. The specific media types supported may be limited by several factors, including your hardware and operating system.

Where To Get Wireshark

You can get the latest copy of the program from the Wireshark website at https://www.wireshark.org/download.html. The download page should automatically highlight the appropriate download for your platform and direct you to the nearest mirror. Official Windows and macOS installers are signed by the Wireshark Foundation.

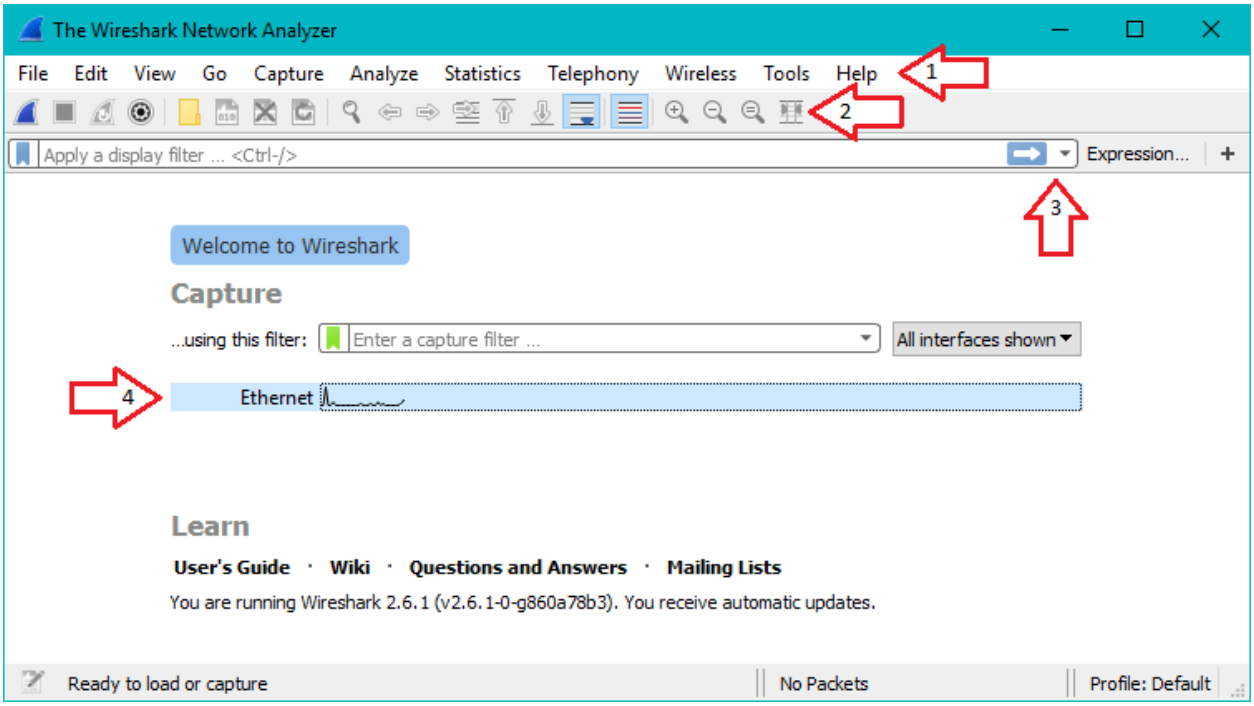
Wireshark User Interface (GUI) Overview

Since Wireshark is built for several different platforms using several different window managers, styles, and versions of the GUI toolkit there may be slight variations in your Wireshark's screen display. Rest assured, primary functionality remains the same so this tutorial should still be easy to understand.

Computer Networks Lab 08

When you first open Wireshark you'll be presented with the start screen.

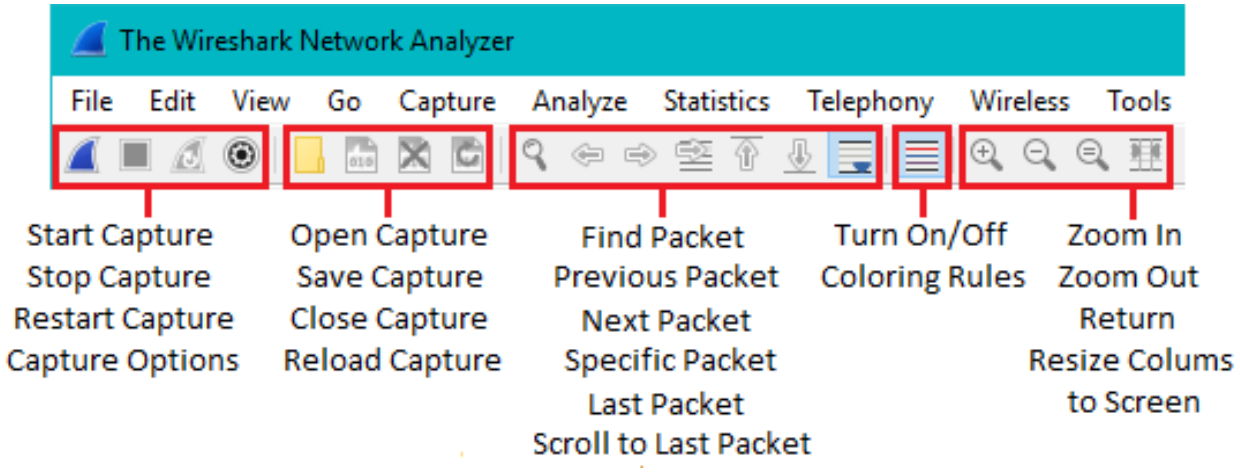
There are four primary areas to the start screen, some of which will carry over into the working screen once you pick an interface to work capture traffic from.



Primary Areas of the Wireshark Start Screen

1. The Menu
2. The Main Toolbar
3. The Filter Toolbar
4. The Interface List

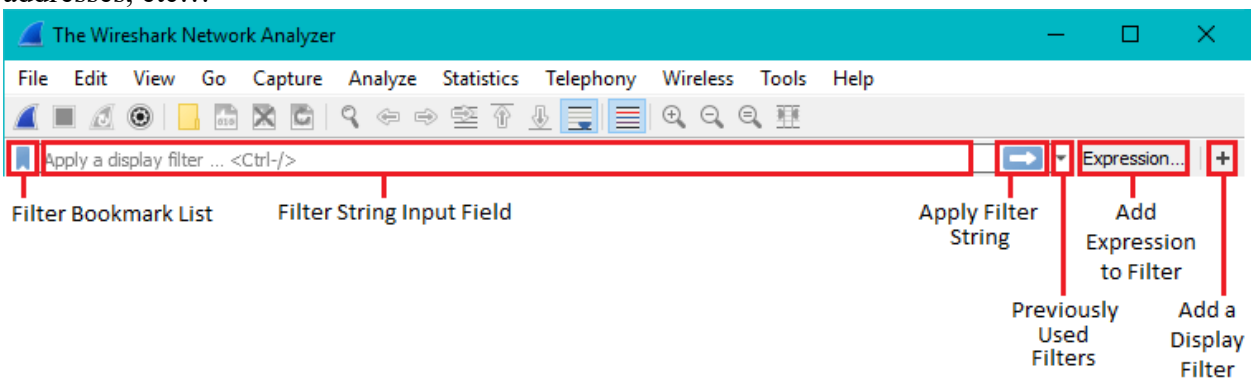
Wireshark Main Toolbar



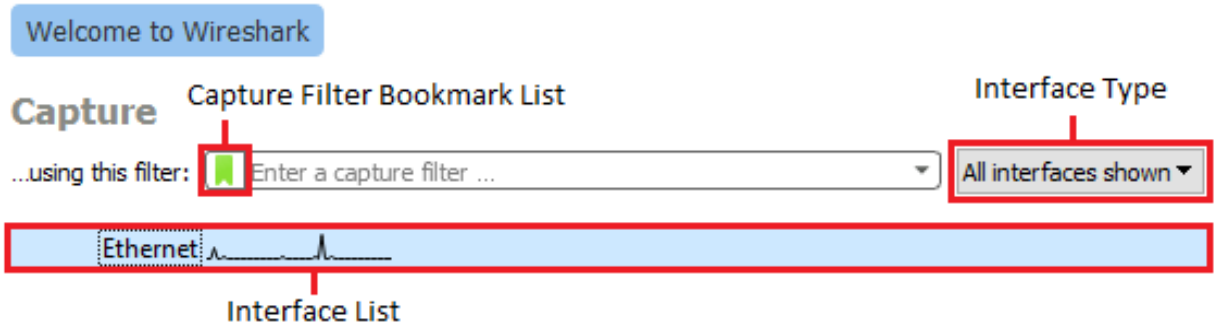
This is a quick access toolbar providing easy to use buttons for the most common functions of the main menu. Most of these buttons become active only after you've selected an interface to monitor.

Wireshark Filter Toolbar

This toolbar allows you to quickly edit and apply display filters to your capture. Display filters allow you to narrow down the packets that you've captured to only those that are relevant to what you're trying to see such as specific IP address sources and destinations, protocols, MAC addresses, etc…

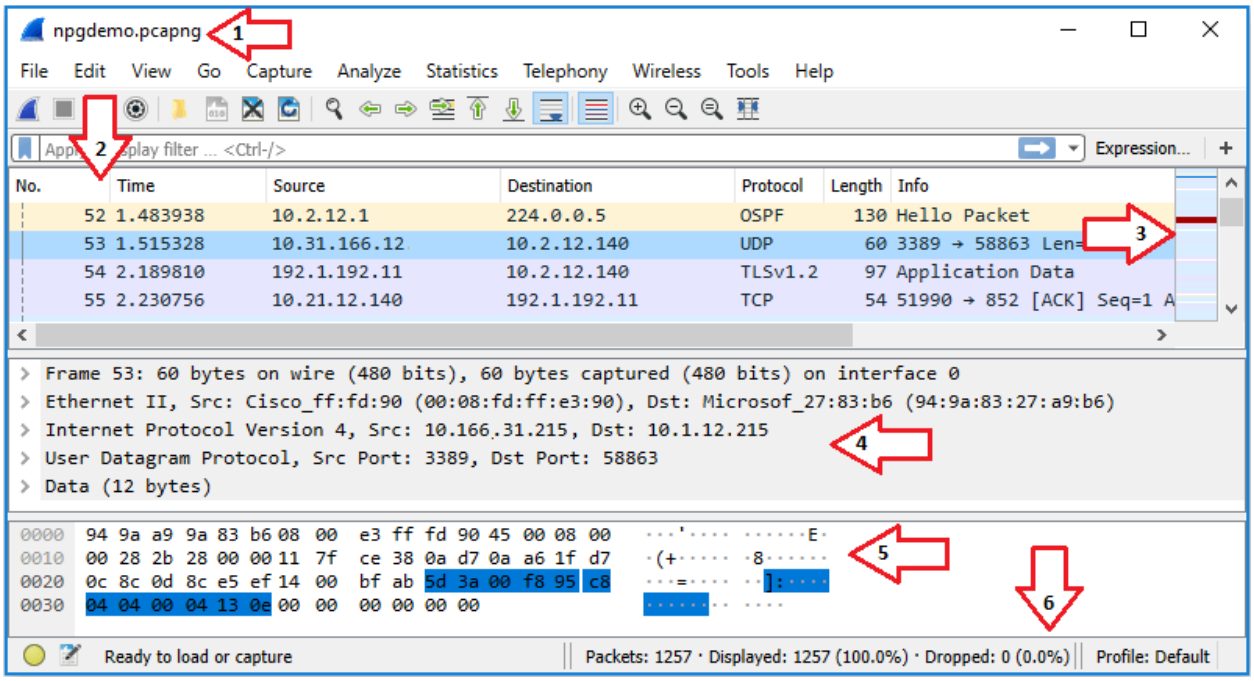


Wireshark Interface List



The Interface List is the area where the interfaces that your device has installed will appear. Before you can see packet data you need to pick one of the interfaces by clicking on it. You can choose a capture filter and type of interface to show in the interfaces lists at this screen as well. Clicking on and interface or opening an existing capture file will take you to the working screen:

Computer Networks Lab 08



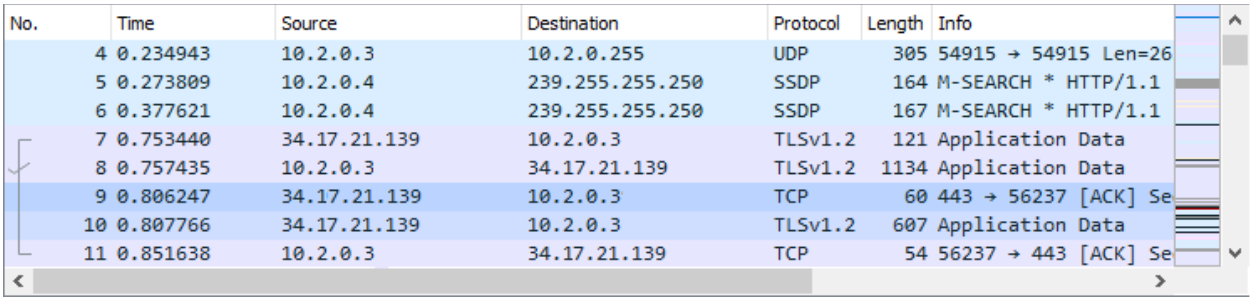
Primary Areas of the Wireshark Working Screen:

1. Title Bar
2. Packet List Pane
3. Intelligent Scrollbar
4. Packet Details Pane
5. Packet Bytes Pane
6. The Statusbar

Wireshark Title Bar

This bar shows the name of the interface you're capturing until you save your capture. Then it will show the name of the capture dump file. If you open a saved capture file its' name will be displayed here.

Wireshark Packet List Pane



Every line in this pane represents one packet. By default, the pane is broken up into 7 columns, each of which provides useful identification data for each packet and can be sorted to help you

better dissect the data. You can remove, add, and reorder the columns to suit your needs. Selecting a packet will show more details in the Packet Details Pane and Packet Bytes Pane.

No.

The No. column assigns a unique number to each packet. It can also display a symbol to help identify the relationship between packets if you click on a packet.

Time

Displays the timestamp for when the packet was captured. The format of this timestamp is customizable.

Source

Displays the source IP or MAC address that the packet originated from.

Destination

Displays the destination IP or MAC address that the packet was heading to.

Protocol

Displays abbreviated protocol information for the packet.

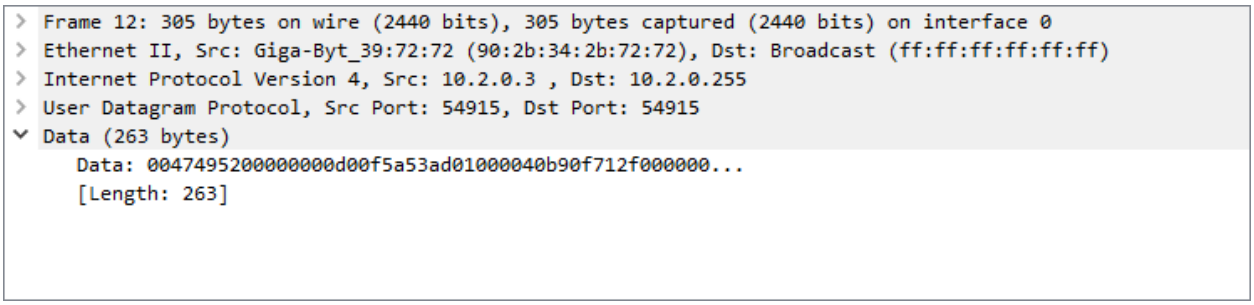
Length

Displays the packet length.

Info

Displays additional information related to the packet.

Wireshark Packet Details Pane



When you click on a packet in the Packet List Pane it loads data about that packet in the Packet Details Pane. This pane displays the packet's different protocols and protocol fields. This is list is displayed as a tree that can be expanded to show even more detail.

The details can also include a couple special fields that Wireshark generates on its' own by analyzing the packets. The two fields are Generated Fields and Links.

Generated Fields

This information is enclosed in brackets ([ ]) and contains info such as TCP analysis, response time, checksum validation, and IP geolocation.

Links

Wireshark will generate a link if it detects relationships between packets. These links will be formatted blue with an underline. Double-clicking on the link will jump you to the related packet.

There is also a context menu which you can access by right clicking within the pane.

Wireshark Packet Bytes Pane

```
0000  80 2a a8 5e 18 ad 90 2b  34 39 72 2d 08 00 45 00   ·*·^···+ 49r-··E·
0010  00 28 58 0e 40 00 80 06  60 49 0a 02 00 27 22 c5   ·(X·@··· `I···'"·
0020  15 8b db ad 01 bb 06 b9  68 32 52 75 f4 49 50 10   ········ h2Ru·IP·
0030  00 fe 42 93 00 00                                  ··B···
```

When you click on a packet in the Packet List Pane it loads data about the packet in the Packet Bytes Pane as well. This data is in a hexdump style with each line displaying the data offset, 16 hexadecimal bytes, and 16 ASCII bytes. The periods represent non-printable bytes.
If you mouse over a specific piece of data Wireshark will highlight the corresponding data which you see in the example above in blue where the hex bytes are highlighted along with the associated ASCII bytes.
Occasionally, when Wireshark reassembles some packets into a single chunk or decrypts data, there will be multiple pages tabbed at the bottom of the Packet Bytes Pane.
There is also a context menu which you can access by right clicking within the pane.

Wireshark Statusbar

The statusbar contains informational messages.
Note: Color Coding: You'll probably see packets highlighted in green, blue, and black. 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

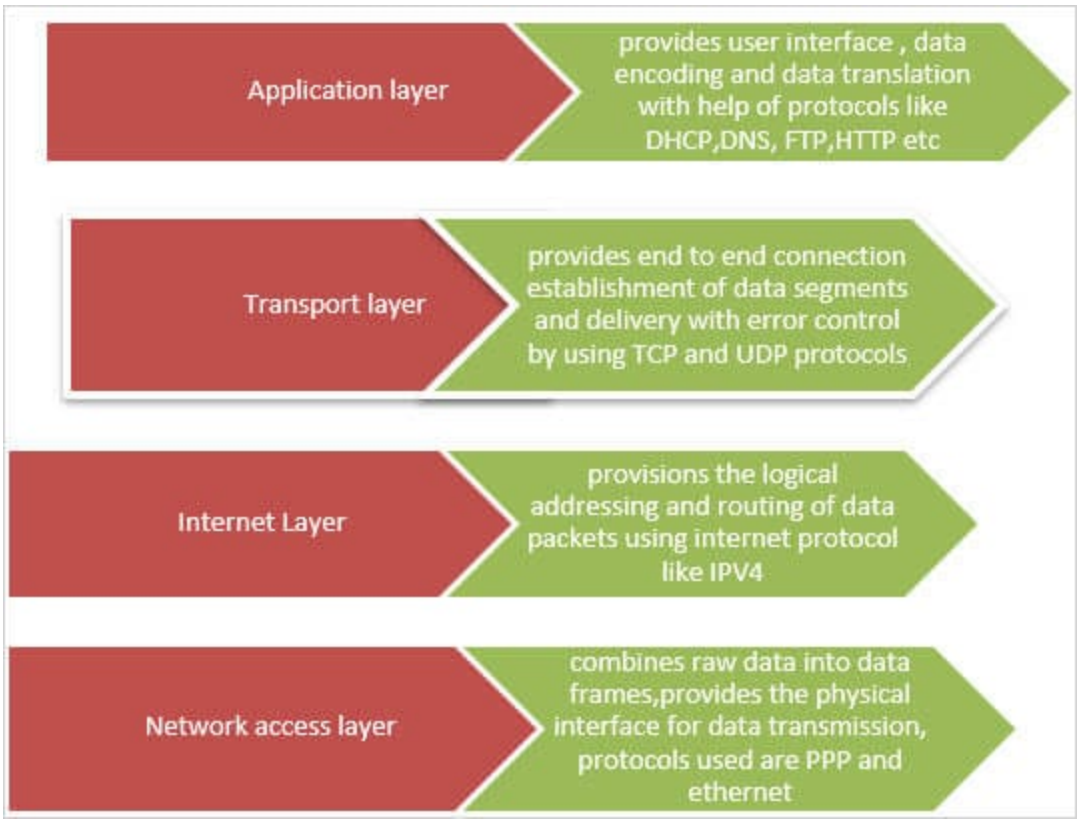
Computer Networks Lab 08

OSI Network Layer Analysis via Wireshark

We all know that OSI (Open Systems Interconnection) is a reference model for how applications communicate over a network. Here are the 7 layers according to OSI model:

7 Application	**Key Responsibilities -** User Application Services **Common Protocols -** DNS; NFS; BOOTP; DHCP; SNMP; RMON; FTP; TFTP; SMTP; POP3; IMAP; NNTP; HTTP; Telnet **Scope-** Application data **Data Type can Handled -** User Data
6 Presentation	**Key Responsibilities -** Data Translation; Compression and Encryption **Common Protocols-** SSL; Shells and Redirectors; MIME **Scope-** Application data representations **Data Type can Handled -** Encoded User Data
5 Session	**Key Responsibilities -** Session Establishment, Management and Termination **Common Protocols-** NetBIOS, Sockets, Named Pipes, RPC **Scope-** Sessions between local or remote devices **Data Type can Handled -** Session
4 Transport	**Key Responsibilities** - Process-Level Addressing; Multiplexing/Demultiplexing; Connections; Segmentation and Reassembly Acknowledgments and Retransmissions, Flow Control **Common Protocols**- TCP and UDP; SPX; NetBEUI/NBF **Scope-** Communication between software processes **Data Type can Handled -** Datagram and Packets
3 Network	**Key Responsibilities-**Logical Addressing; Routing; Datagram Encapsulation; Fragmentation and Reassembly; Error Handling and Diagnostics **Common Protocols-** IP; IPv6; IP NAT; IPsec; Mobile IP; ICMP; IPX; DLC; PLP; Routing protocols such as RIP and BGP **Scope-** Messages between local or remote devices **Data Type can Handled-** Datagram and Packets
2 Datalink	**Key Responsibilities-** Logical Link Control; Media Access Control; Data Framing; Addressing; Error Detection and Handling; Defining Requirements of Physical Layer **Common Protocols-** EEE 802.2 LLC, Ethernet Family; Token Ring; FDDI and CDDI; IEEE 802.11 (WLAN, Wi-Fi); HomePNA; HomeRF; ATM; SLIP and PPP **Scope-** Low-level data messages between local devices **Data Type can Handled -** Frames
1 Physical	**Key Responsibilities-** Encoding and Signaling; Physical Data Transmission; Hardware Specifications; Topology and Design **Common Protocols-** (Physical layers of most of the technologies listed for the data link layer) **Scope-** Electrical or light signals sent between local devices **Data Type can Handled** - Bits

There is another network model which is TCP/IP. Here are the 4 layers according to TCP/IP model:



Relation OSI and TCP/IP model:

Below is the relation between OSI model and TCP/IP model.

OSI Model	**TCP/IP Model**
Application Layer	**Application Layer**
Presentation Layer	
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	Network access Layer
Physical Layer	

Now the question comes, in **Wireshark what model we should be expecting?** Actually in Wireshark we observe below layers

Application Layer	**[Layer 5]**
Transport Layer	**[Layer 4]**
Network Layer	**[Layer 3]**
Data Link Layer	**[Layer 2]**
Physical Layer	**[Layer 1]**

Now we understand that the above layers are not exactly OSI or TCP/IP but a combination of both models.

HTTP analysis using Wireshark

In this lab, we'll explore several aspects of the HTTP protocol: the basic GET/response interaction, HTTP message formats, and retrieving large HTML files.

The Basic HTTP GET/response interaction

Let's begin our exploration of HTTP by downloading a very simple HTML file - one that is very short, and contains no embedded objects. Do the following:

1. Start up your web browser.
2. Start up the Wireshark packet sniffer. Enter "http" (just the letters, not the quotation marks, and in lower case) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We're only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).
3. Wait a bit more than one minute, and then begin Wireshark packet capture.
4. Enter the following to your browser
   http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html
   Your browser should display the very simple, one-line HTML file.
5. Stop Wireshark packet capture

Your Wireshark window should look similar to the window shown in Figure 1. If you're unable to run Wireshark on a live network connection, you can download a packet trace that was created when the steps above were followed.

Computer Networks Lab 08

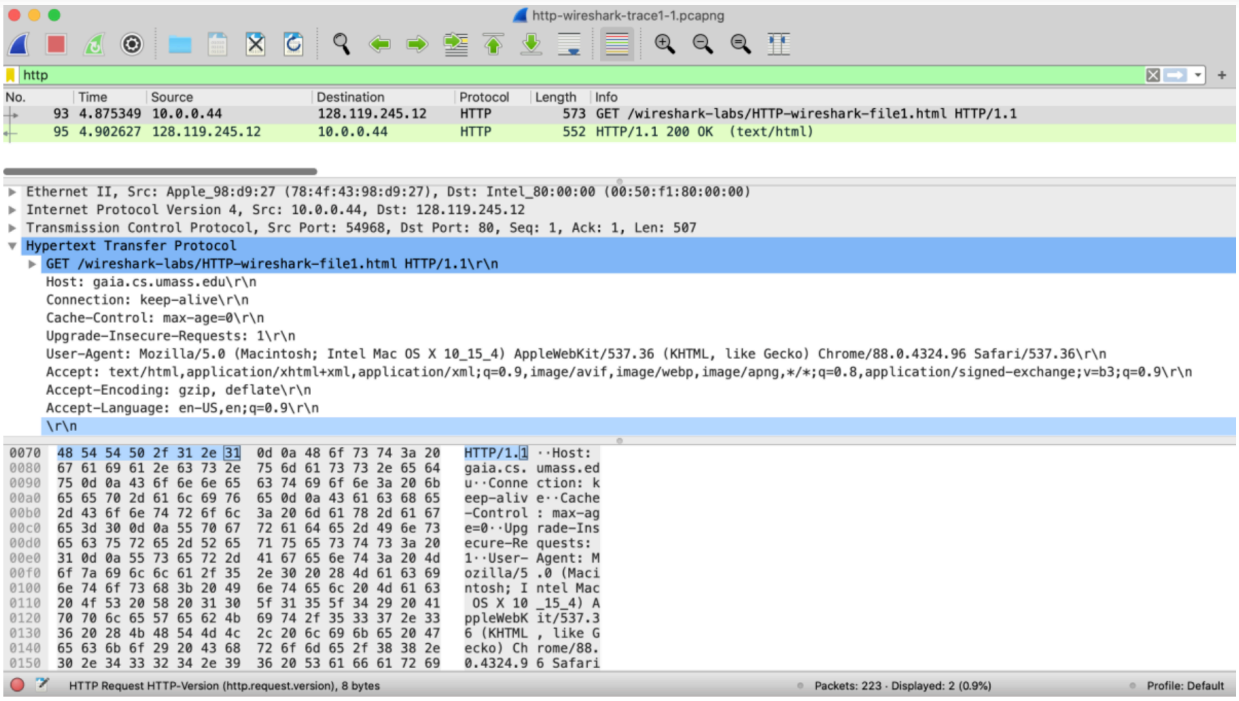


Figure1:Wireshark Display after http://gaia.cs.umass.edu/wireshark-labs/ HTTPwireshark-file1.html has been retrieved by your b

The example in Figure 1 shows in the packet-listing window that two HTTP messages were captured: the GET message (from your browser to the gaia.cs.umass.edu web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP OK message, which is highlighted in the packet-listing window). Recall that since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well. We want to minimize the amount of non-HTTP data displayed (we're interested in HTTP here, and will be investigating these other protocols is later labs), so make sure the boxes at the far left of the Frame, Ethernet, IP and TCP information have a plus sign or a right-pointing triangle (which means there is hidden, undisplayed information), and the HTTP line has a minus sign or a down-pointing triangle (which means that all information about the HTTP message is displayed).

Task: By looking at the information in the HTTP GET and response messages, answer the following questions

1. Is your browser running HTTP version 1.0, 1.1, or 2?
2. What version of HTTP is the server running?
3. What languages (if any) does your browser indicate that it can accept to the server?
4. What is the IP address of your computer? What is the IP address of the gaia.cs.umass.edu server?
5. What is the status code returned from the server to your browser?
6. When was the HTML file that you are retrieving last modified at the server?
7. How many bytes of content are being returned to your browser?
8. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one

Computer Networks Lab 08

The HTTP CONDITIONAL GET/response interaction

Recall from Section 2.2.5 of the text, that most web browsers perform object caching and thus often perform a conditional GET when retrieving an HTTP object. Before performing the steps below, make sure your browser's cache is empty.
Now do the following:

- Start up your web browser, and make sure your browser's cache is cleared, as discussed above.
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser : http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
- Your browser should display a very simple five-line HTML file.
- Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)
- Stop Wireshark packet capture, and enter "http" (again, in lower case without the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.

Task: Answer the following questions:

1. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?
2. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?
3. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET6 ? If so, what information follows the "IF-MODIFIED-SINCE:" header?
4. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

See https://www.howtogeek.com/304218/how-to-clear-your-history-in-any-browser/ for instructions on clearing your browser cache.