WIRESHARK

BASICS OF WIRESHARK

ABHISHEK GUATAM
LUCIDEUS

INTRODUCTION

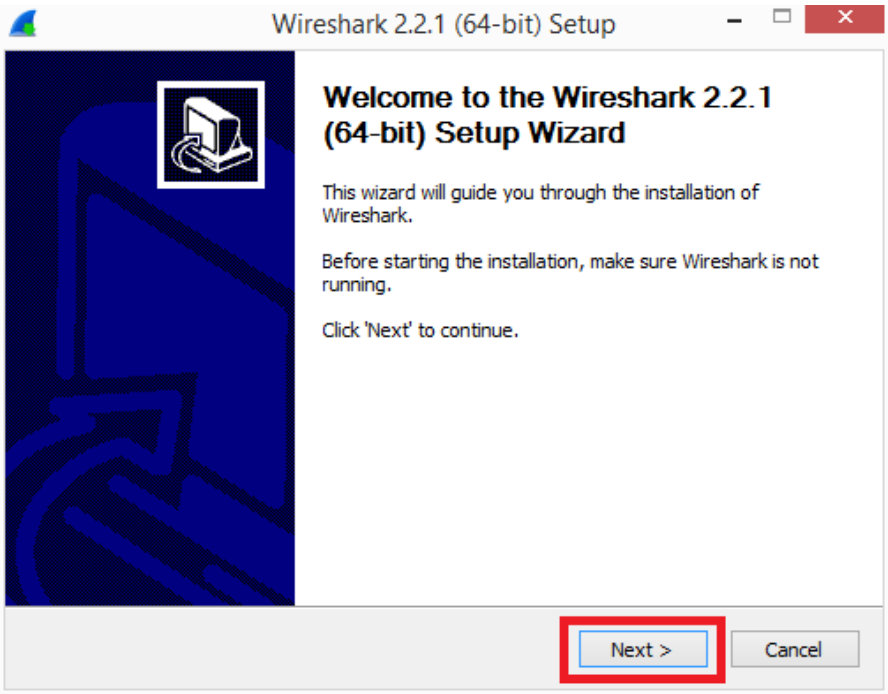
Wireshark is a **network packet analyzer**. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.
Wireshark is perhaps one of the best open source packet analyzers available today.

Features of WIRESHARK

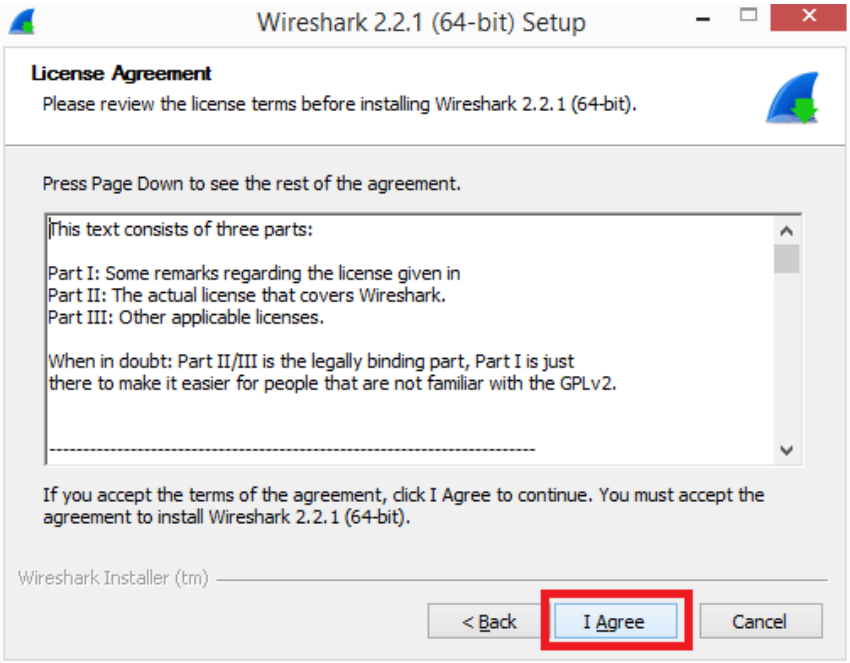
Available for UNIX and Windows.
Capture live packet data from a network interface.
Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
Import packets from text files containing hex dumps of packet data.
Display packets with very detailed 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.
Wireshark captures packets and lets you examine their contents.
Live capture from many different network media
Import files from many other capture programs
Export files for many other capture programs
Many protocol dissector
Open Source Software

INSTALLING WIRESHARK

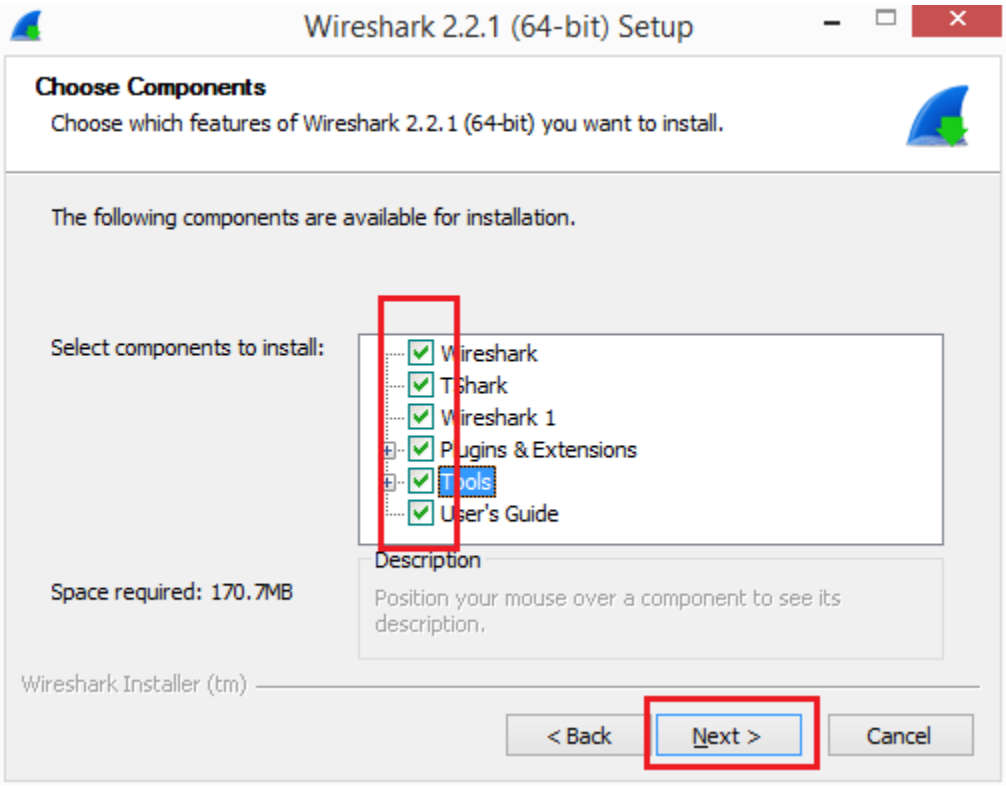
STEP 1: Double click on the downloaded .exe file of Wireshark. Click yes if UAC is prompted and then click “NEXT”.



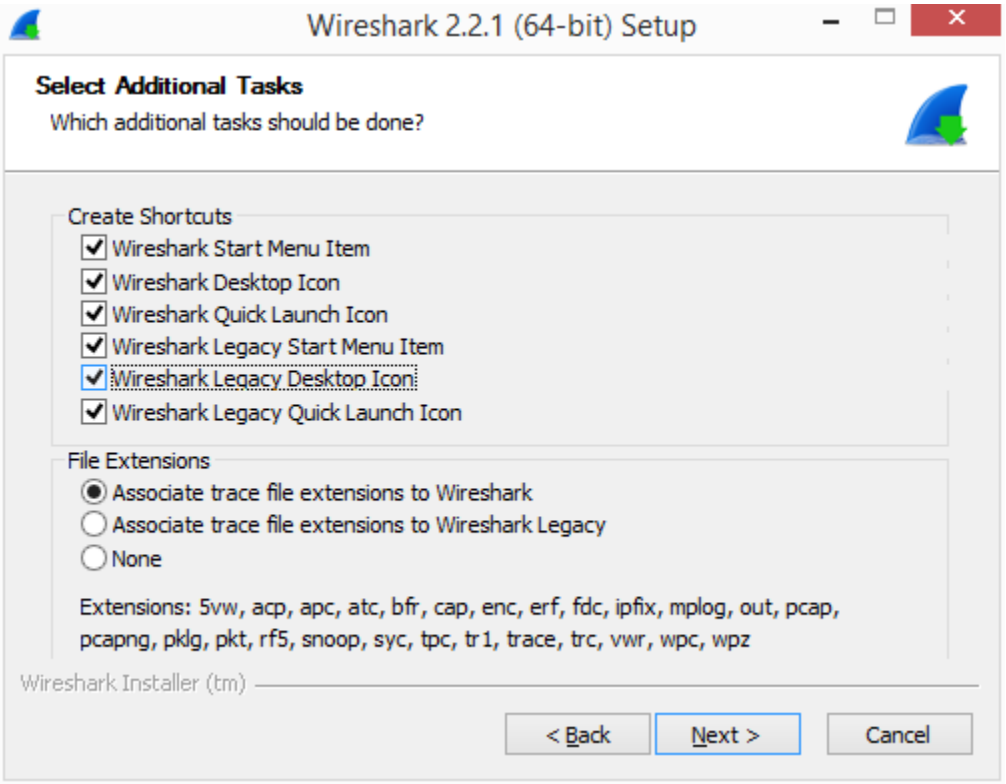
STEP 2: Accept the License Agreement by clicking “I Agrre”



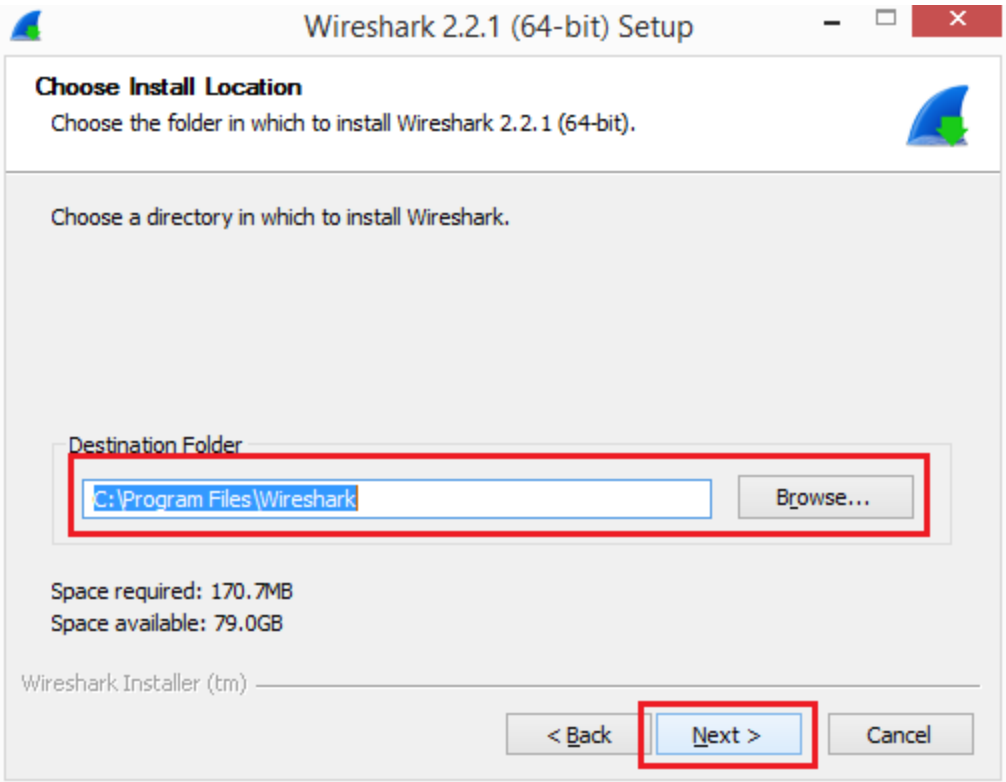
STEP 3: Tick all the check box and then click “NEXT”.



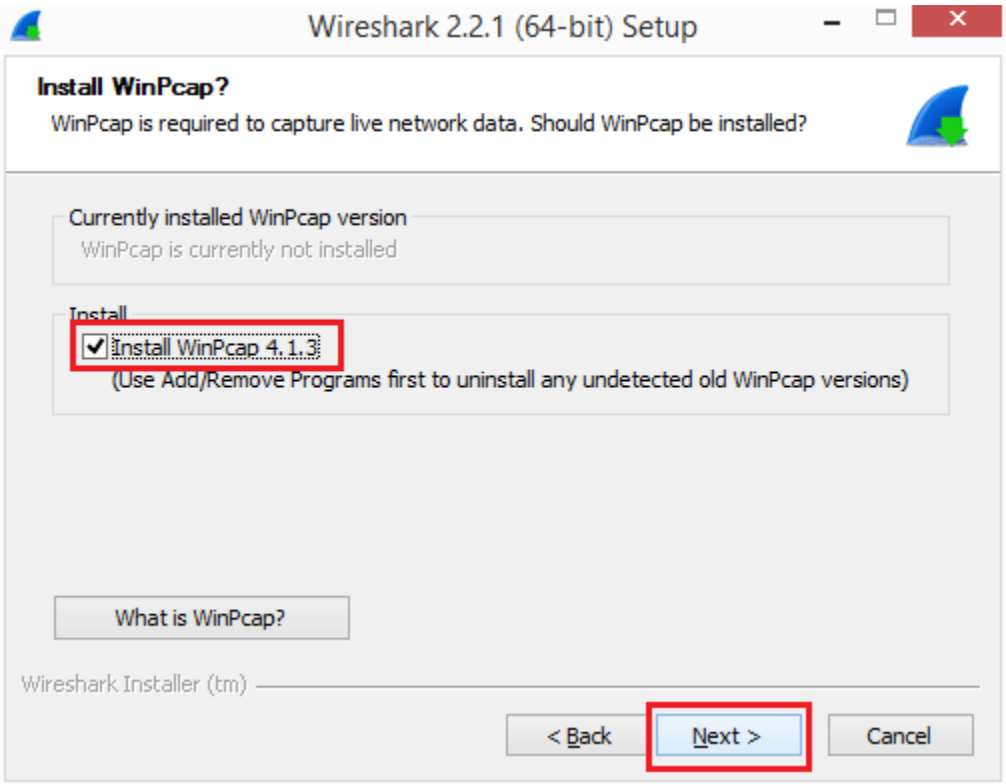
STEP 4: In the Following Screen select the options according to your own choice.



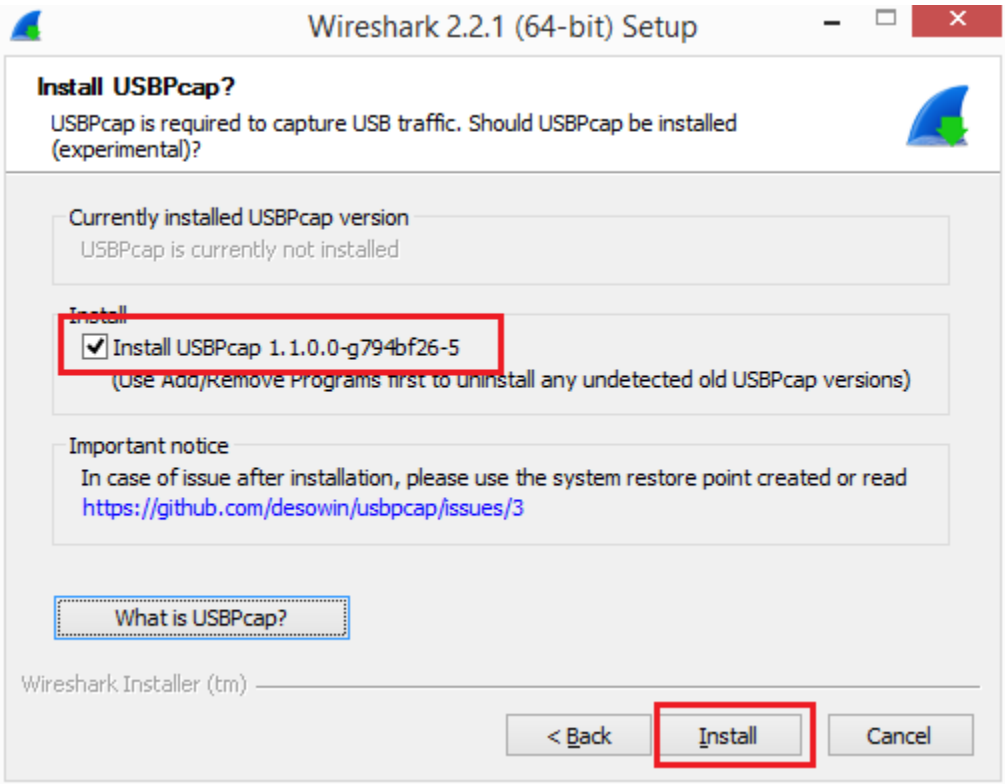
STEP 5: Select the directory in which you want to install but it is recommended to install the program in the default directory.



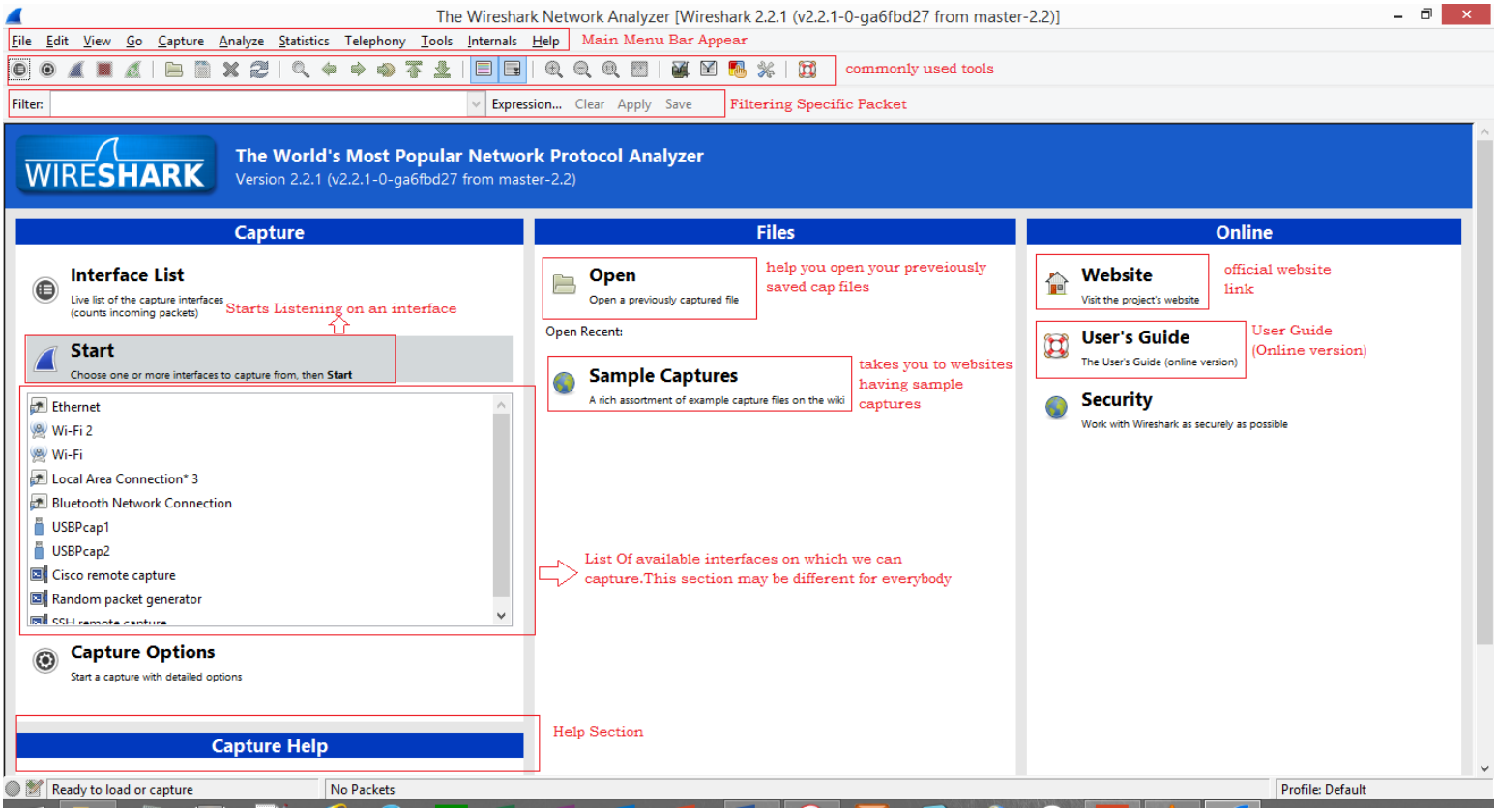
STEP 6: CLICK “Install WinPcap”. winPcap is the application needed for capturing purpose. (For windows users only)

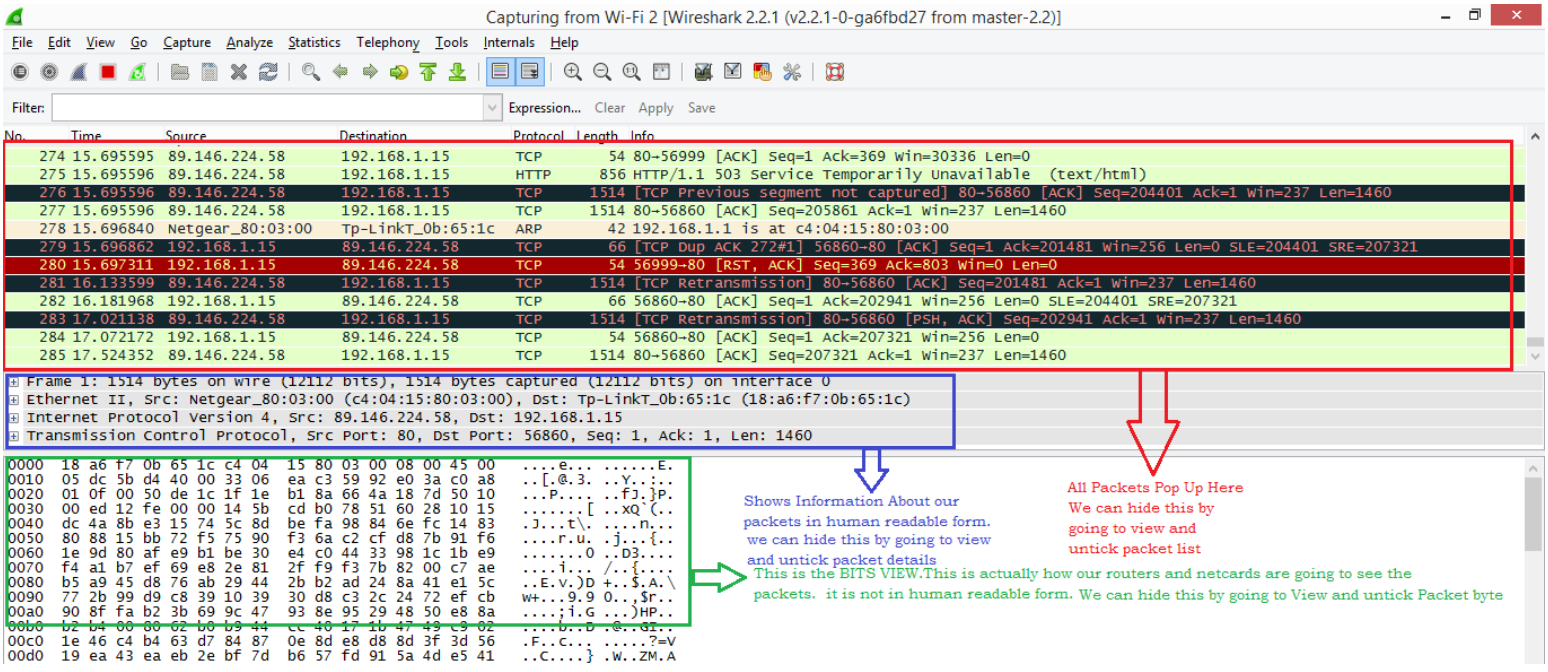


STEP 7: CHECK install “Install USBPcap”. USBPcap is an open-source USB sniffer for Windows.

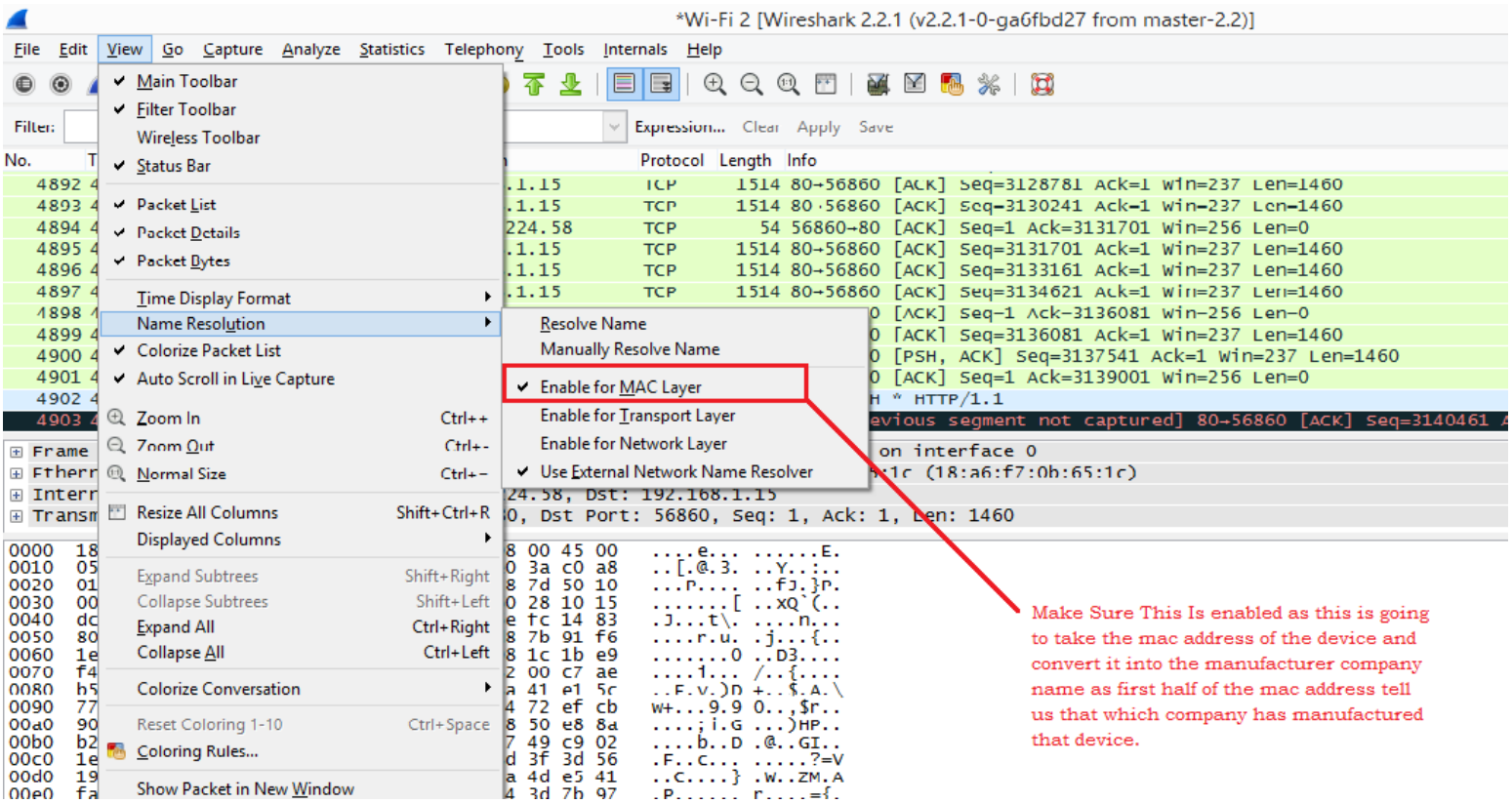


OVERVIEW OF INTERFACE





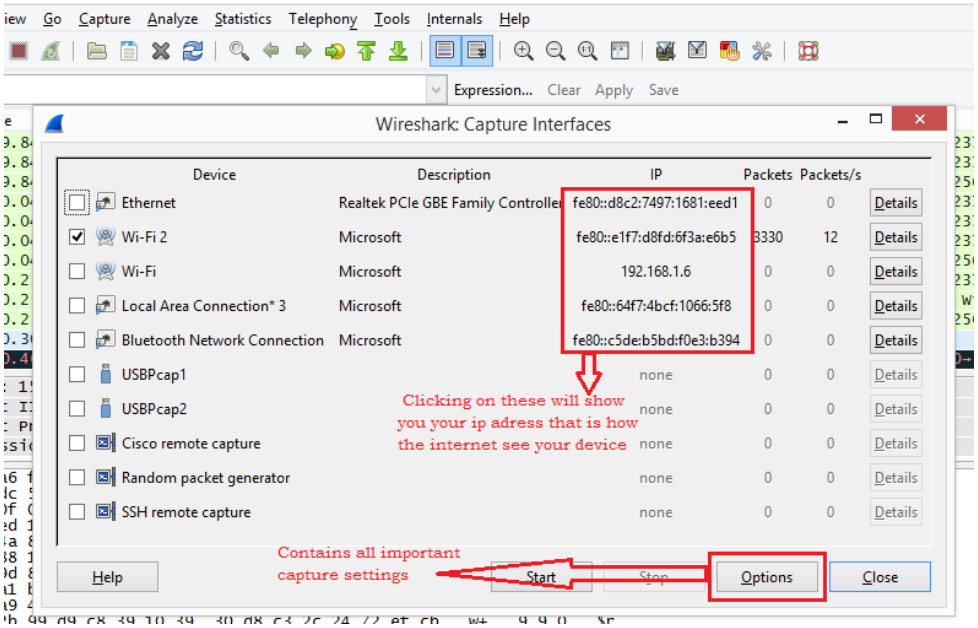
NOTE WE CAN CHANGE THE LAYOUT OF WIRESHARK BY GOING TO EDIT AND THEN TO PREFERNCES AND SET IT ACCORDING TO OUR OWN WISH.



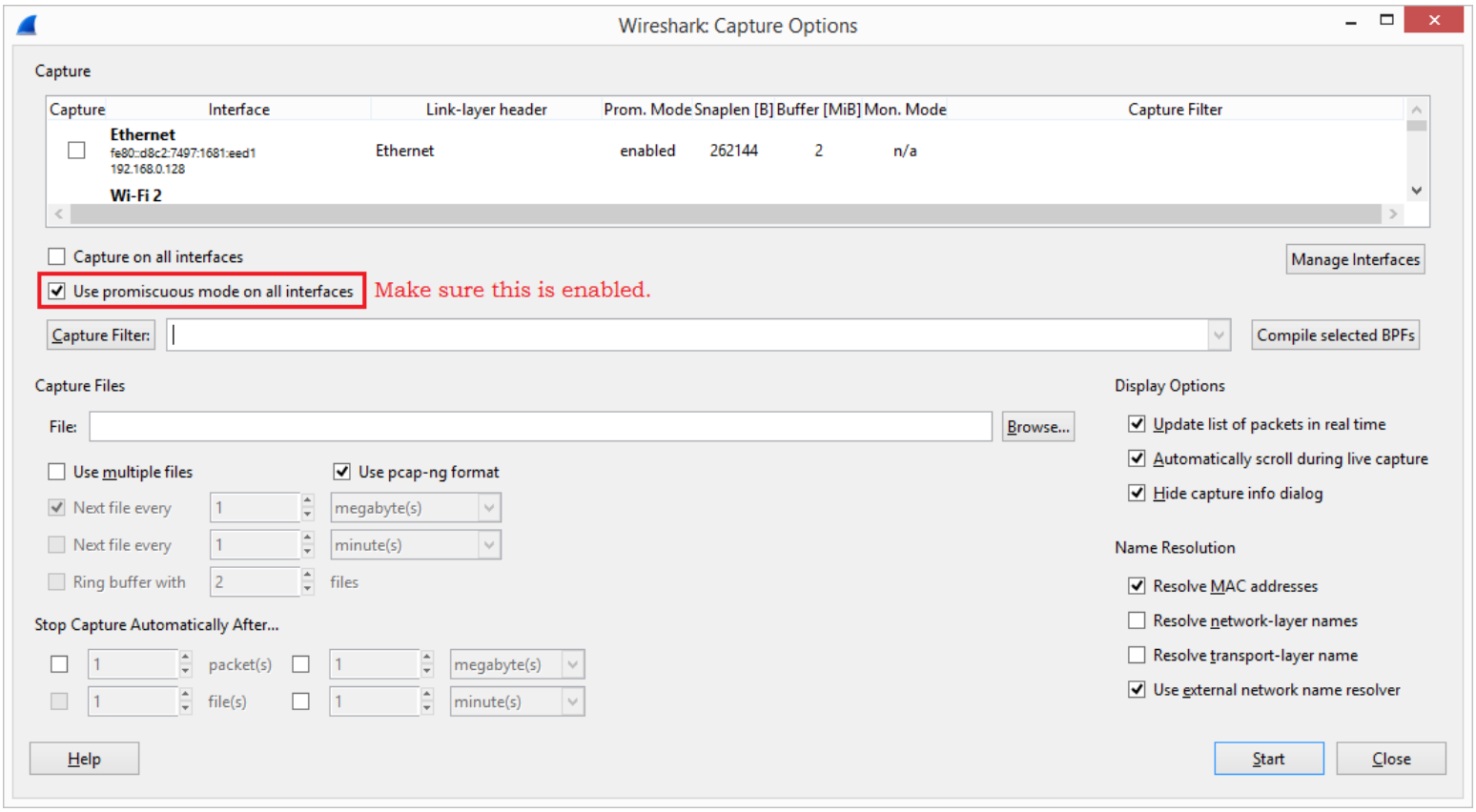
Capture Options

These settings are very important **as it tells the wireshark what we want to listen on the network.**

Step 1: Goto to capture options and click on interfaces the keyboard shortcut for this is ctrl+I. A screen similar to following screen will popup.



Step 2: now click on options. It will give more capture options and will give us more control.



Make sure listening is enabled in promiscuous mode. Now let's understand what promiscuous mode is. Lets suppose you went to a coffee shop and you wanted to sniff everyone's traffic although that illegal and you should never do it, but you start capturing and something weird started you noticed that you were only able to view your own traffic. This happened because all the packets which came to our machine which was not meant for us were ignored by our machine and this is how it build. So what promiscuous mode does is that it enables the device to listen to everyone else traffic but sometimes the promiscuous mode is default blocked in our interface device so for this we need to buy a device which supports promiscuous mode.

TOOLBAR ICONS

 : This shows the interface list

 : This shows capture options.

 : Start Capture

 : Stop Capture

 : Refresh or Restart

 : Open a capture File

 : Save the capture File

 : Close the capture file

 : Reload the current capture file

 : Find a packet

 : Go back in packet History

 : Go forward in packet History

 : Goto packet with number(Packet Id)

 : Goto first packet

 : Goto Last packet

 :Edit preference

 : Show some help

 : Colorize packet list

 : Auto scroll packet list in live capture

 : Zoom in

 : Zoom out

 : Zoom 100%

 : Resize all columns

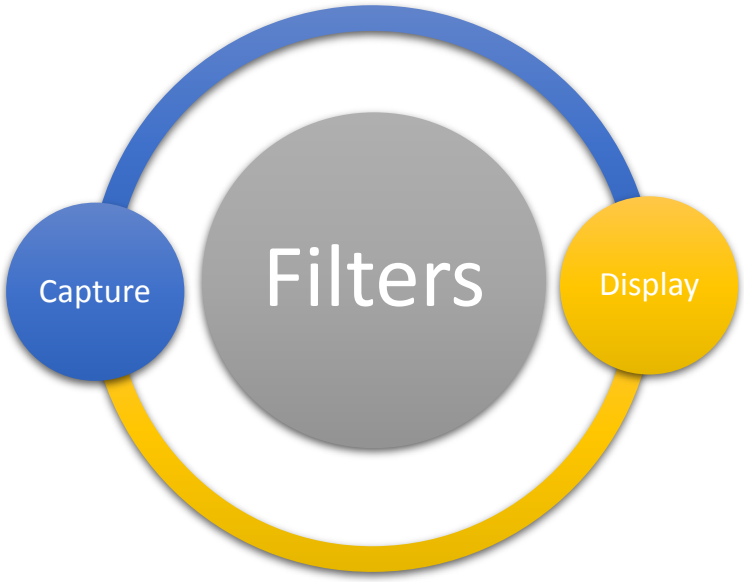
 : Edit capture file

 : Edit display filter

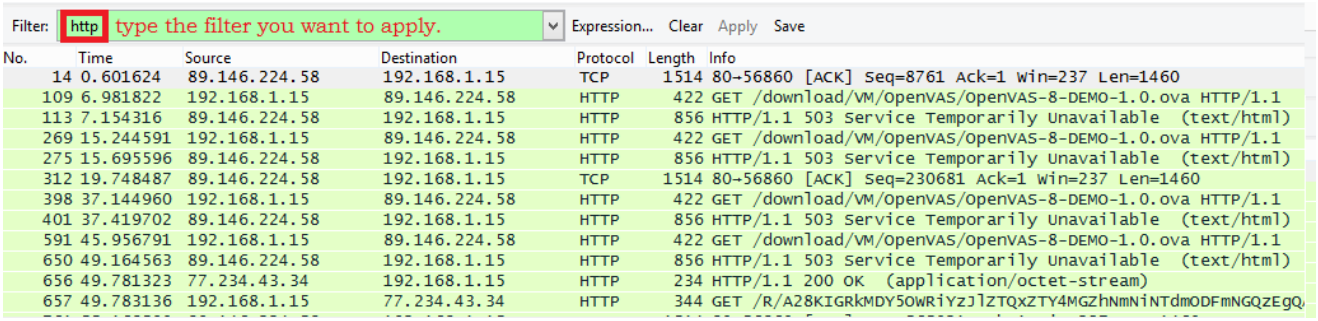
 : Edit packet coloring rule

FILTERS

it is a way to filter out our packets. When we do capturing the packets we get lots of packets out of which we might not worry about a lot of packets like all UDP Packets and Some TCP Packets so here filter proves to be a great feature for us . There are two different types of filters:

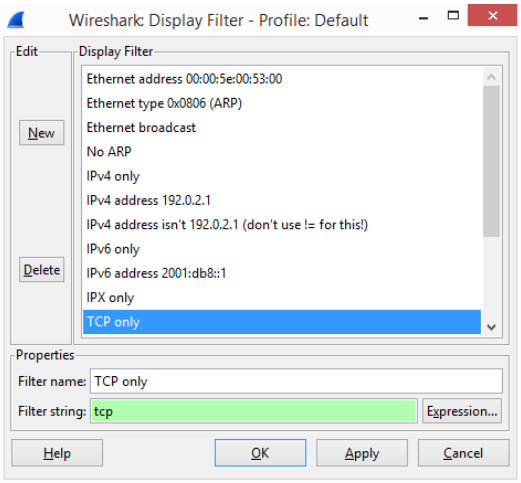


1) Display Filters: This is a display filter.
Example we want to view only the http
Packets so we can type http and press enter and then it will only display the http packets.



2) Capture Filters: It is in the capture settings. If we put http in the capture filter it would not even log any other packet except http during the capturing process.

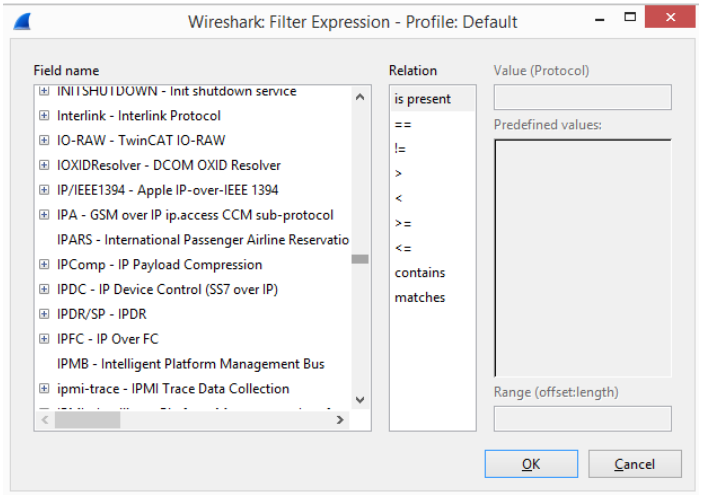
***NOTE*: if we click on the filter button in the left most corner we get a list of most common filters and we can apply any of it or add our own filter in that list. And one more point to note is that whenever we are typing any display filter that is not a valid filter it will show in red color.**



Now suppose we want to filter out only GET packets then we can proceed as follow by typing http.request.method == "GET"

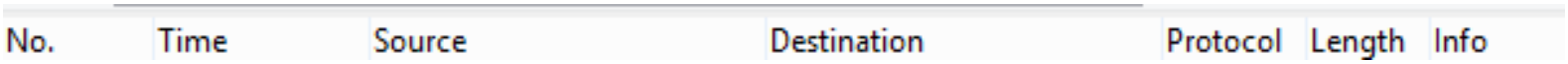
Now lets understand it we are saying we want to filter http packet which are having a request and we want to filter it by method and that method should be GET.

IF we click on expression button it will show us different types of filter we can apply by the use of expressions.



We can also combine two or more filter like we want to view the http GET AND POST method packets we can type

(**http.request.method == "GET"**)||(**http.request.method == "POST"**) those familiar to programming will understand it better. This is a way of saying show me all packets which are using GET or POST. We can also use && which will do a logical and between the two filters that is it will display only those packets which satisfy both the conditions, although if we use it here in this example it won't make any sense. We can also save our filters by clicking the save for future use.



INTERFACE CONTROLS

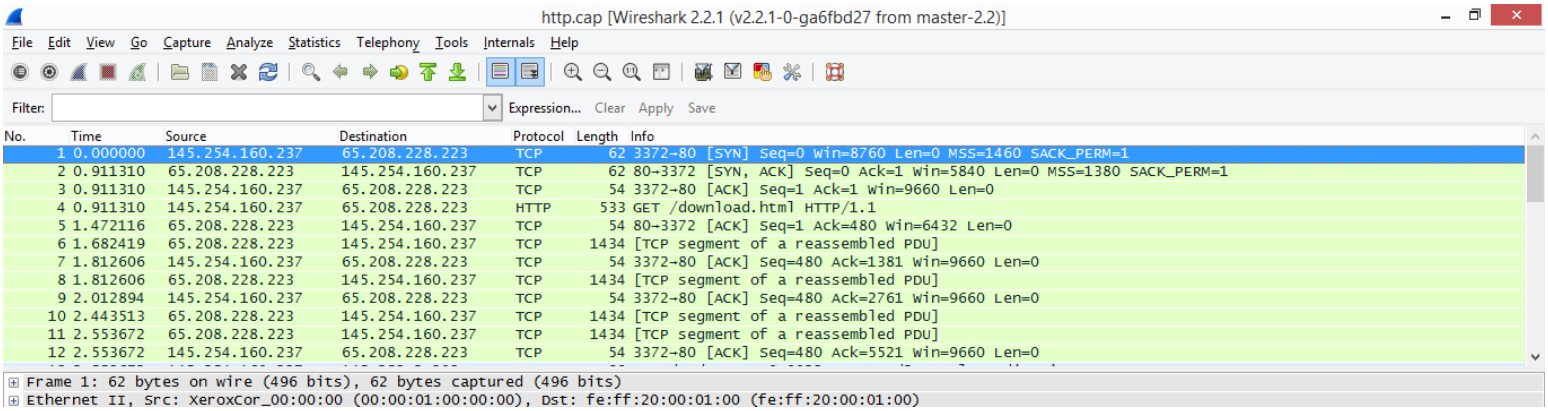
1)The No. tells us the packet number it acts like a Id for a packet.
2)The Time tells us the time at which the packet was captured. The first packets has always a time 0.000000 so if we see the time of last packets it will give us the time till which the capturing was done. We can also change the time display format by going to view and then navigating to Time Display format and select according to our own wish. Here we have options like second, Deci second, millisecond, microsecond, nanosecond.
3) Source tell that where the packet came from.
4) Destination tells us where the packet is going.
5)Info gives us a small summary of the packet.

We can also change their positions of columns by clicking on and dragging to the desired position. We can also hide these columns by right clicking and going to display column and here we can untick the column we want to hide.

If we want to see all packet details then we can right click in the packet details area and then click “Expand all”.

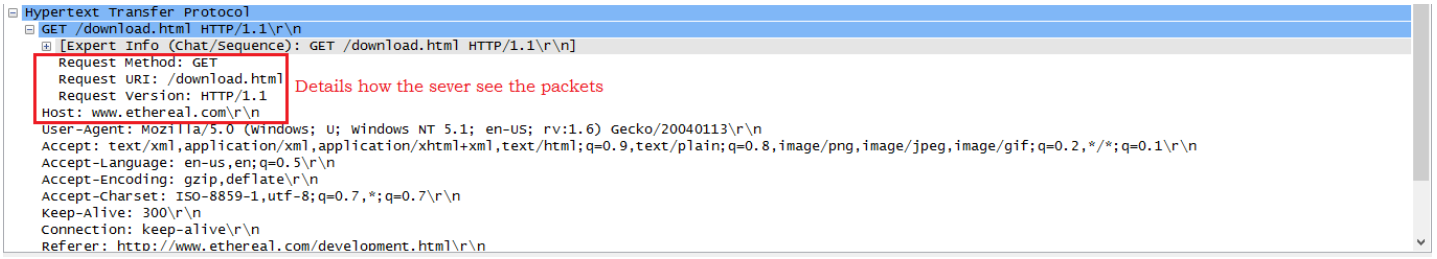
WHAT IS A PACKET?

It is a package of information that we want to send to a different location. Now let’s examine some packets. I have downloaded a sample http cap file and opened it in the wireshark.

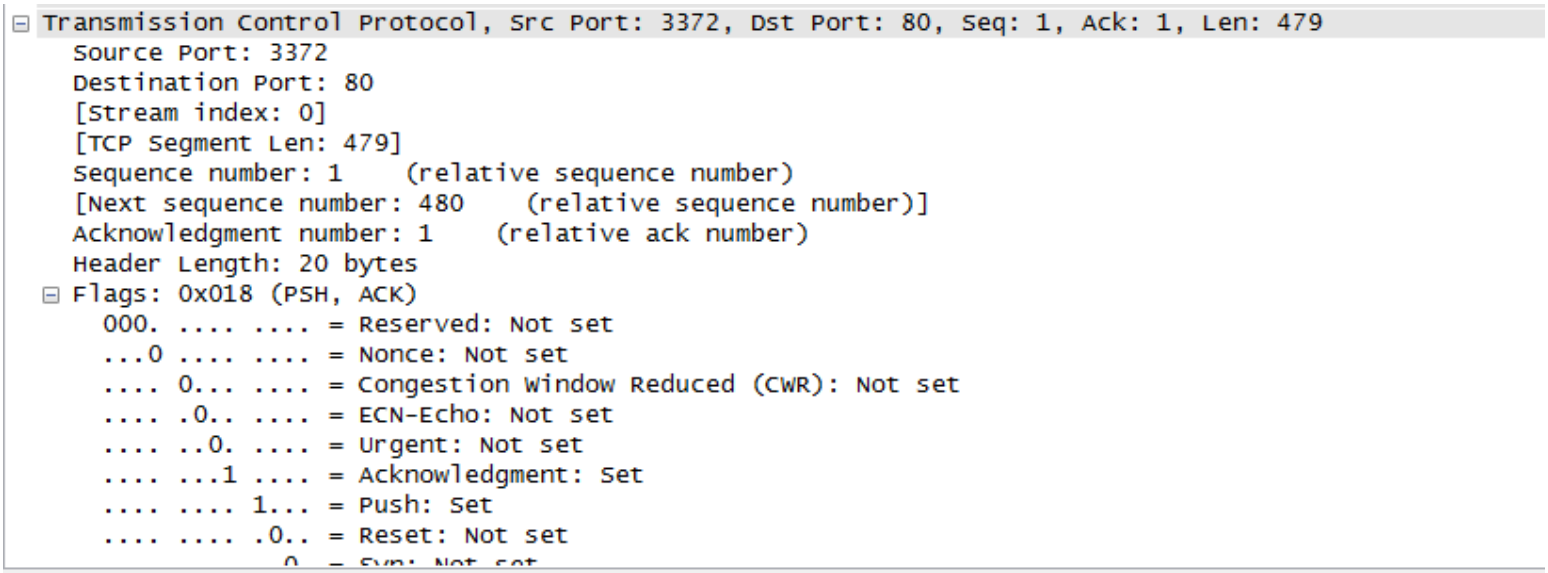


Now lets examine the packet 4 which is a http GET packet. So lets see it details

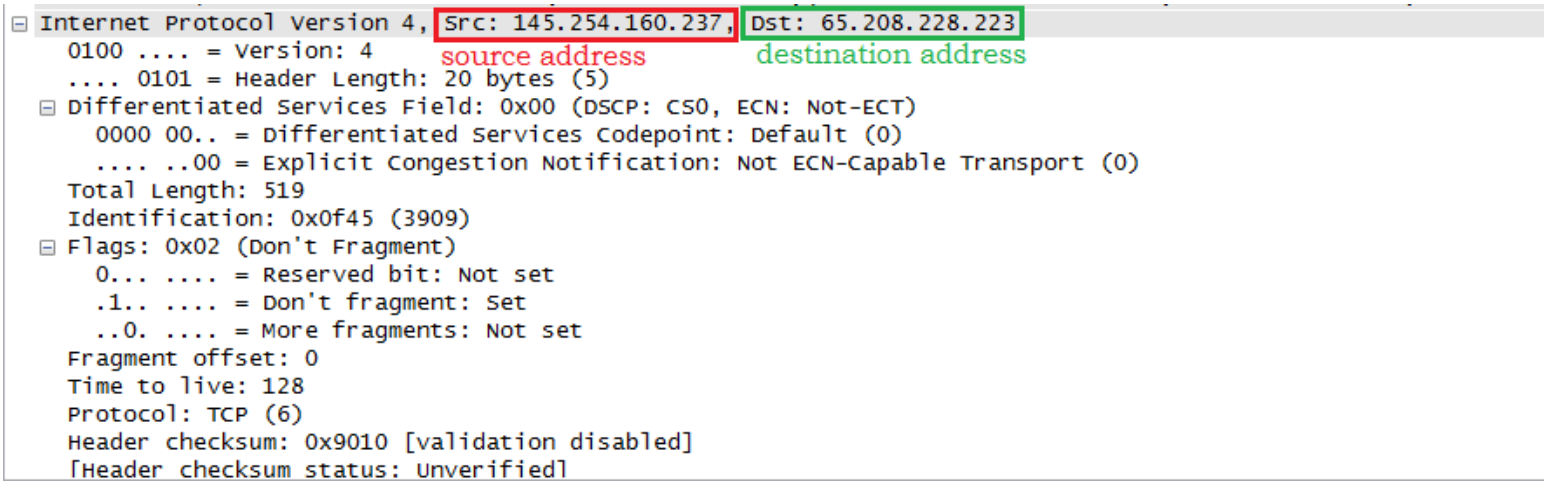
1)



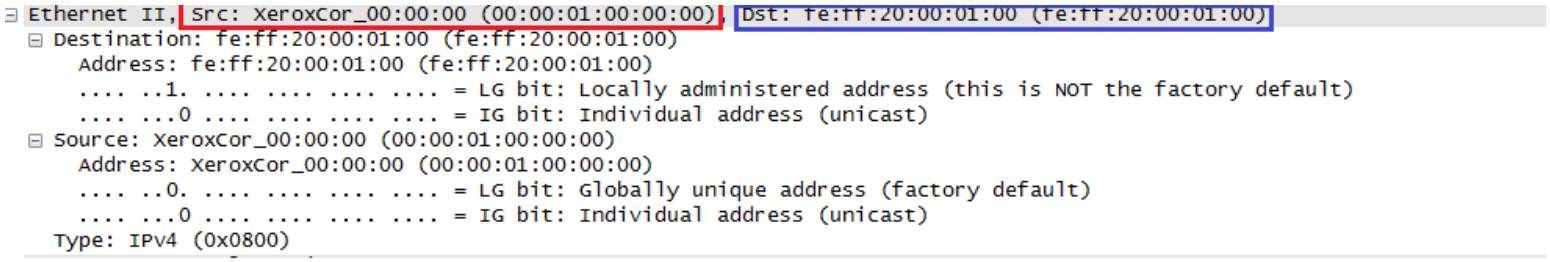
So when server see this packet it thinks that ok it is some http request which means they are looking some files ir pictures or other web content then it checks the get method and which url is being requested. We also need to include basic tcp information in our packet.



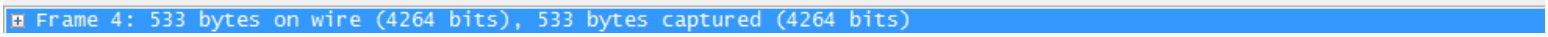
Like here in the above picture we can see the tcp information related to our packet like what ports we are using in our communication. We will also include ipv4 information since we are using ipv4 address.



And this holds the destination and the return address information. Now we will see the Ethernet information also as any data we send goes through some route. It is the information to connect our computer to the router and it uses the mac addressing.

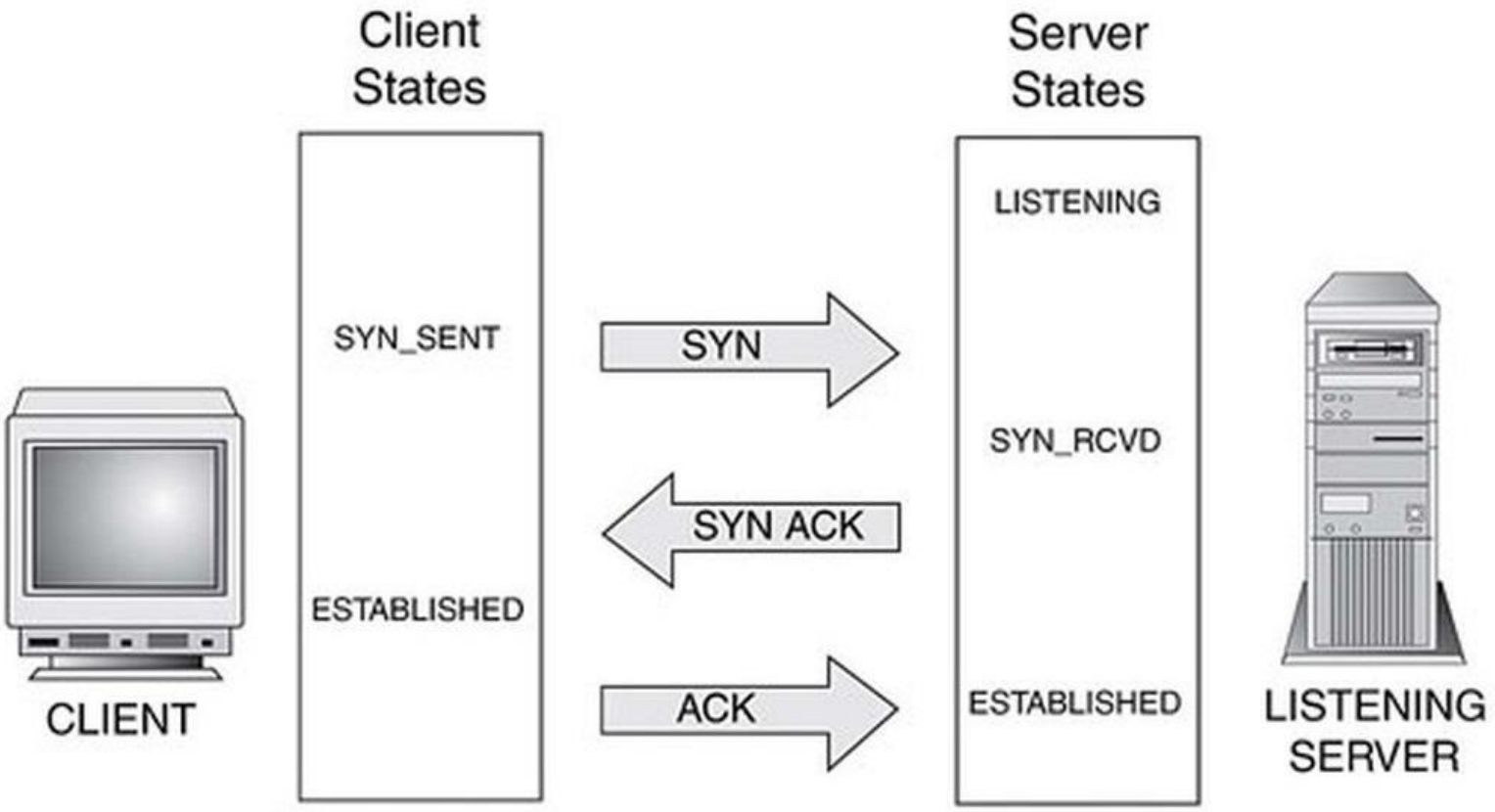


Now lets see what A frame is . A frame is a point where our softwares becomes real so here all these ones and zeros need to be converted into some signal so that it can get transmitted. Eg if we are sending data in Ethernet then it will generate electric pulses according to the ones and the zeros , if we are using fibre optics then light pulses will get generated. Once all the electrical pulses reach to server then the server interprete it and analyze the request.



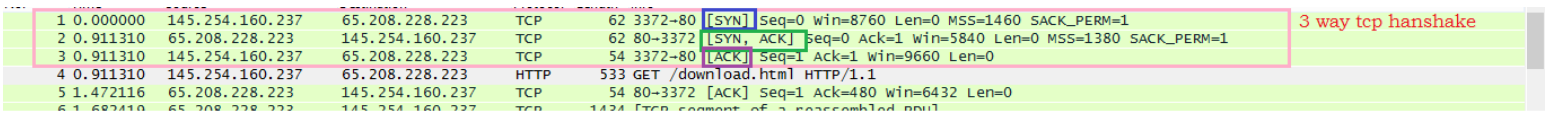
TCP 3 WAY HANDSHAKE

http says what kind of data we are sending and how it should be formatted and read and in that way a server can easily read a packet. TCP says that “This is how we are going to connect” It is not related to actual data or what we are sending.

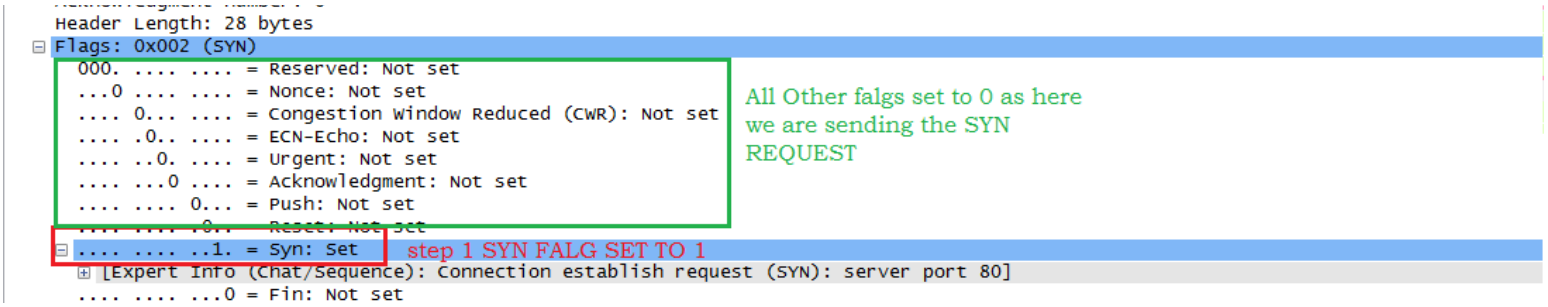


First we will send a SYN request which is Synchronization request it is a way of saying to server that can I connect to you?

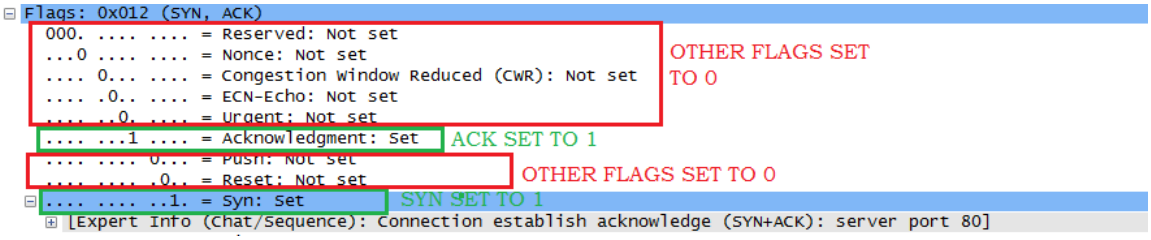
If the server allows the connection a SYN ACK(Synchronization Acknowledgement) is generated which is like saying I acknowledge that you made a request and yes you can connect to me. Now last part of this handshake is that this clients sends one last ACK . After the handshake we are now connected to the server and yes we can now send a http request. Now lets see the handshake in the following picture.



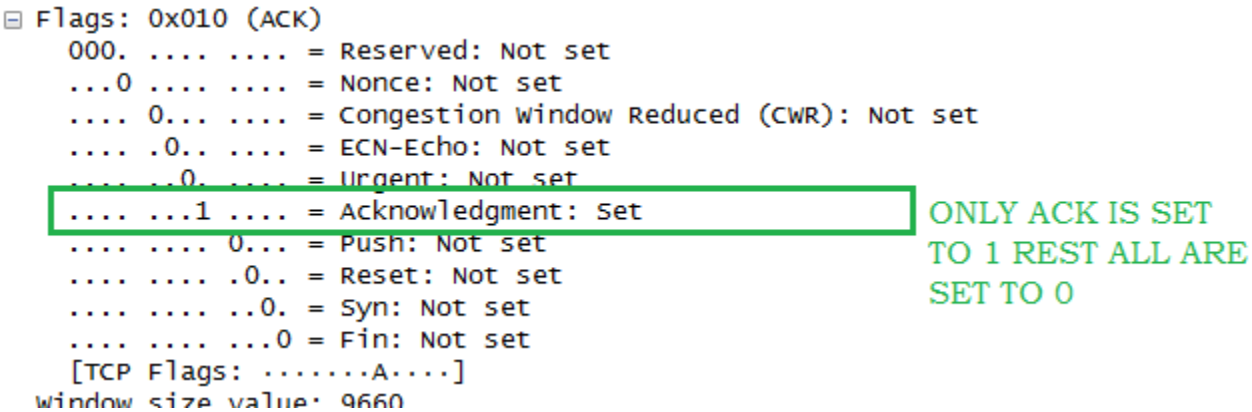
STEP1: Sending SYN request.



STEP2: Sending SYN ACK request



STEP 3: GENETRATING ACK



IF WE ARE NOT GETTING ANY SYN ACK PACKETS THEN THE SERVER MIGHT BE DOWN OR YOU MAY BE BLOCKED BY A FIREWALL.AT THE END OF CONNECTION WE NEED TO SSAY GOOD BYE TO THE SERVER. SO TO TERMINATE WE SEND A FIN PACKET AND THE SERVER RESPONDS BY GENERATING TCP FYN ACK AND AGAIN THE CLIENT GENERATES A ACK FOR THE LAST TIME.