Lab 1 – Wireshark Introduction

1 Introduction:

Wireshark is a network packet analyzer (also known as a packet sniffer). Wireshark is a computer program that can intercept, log and display the traffic passing over a digital network (or part of a network). It can be used to log data traveling over a variety of network types (cable, wireless) and, provided that the content of the data packets is unencrypted, display that data in real time.

Packet sniffers like Wireshark can be used for a variety of purposes both good...

- Analyze network problems and test network communication.
- Debug client/server communications and other network protocol communications
- Monitor network usage and bandwidth (including internal and external users and systems)
- Detect network misuse by internal and external users
- Detect network intrusion attempts (like port scanning).
- Filter suspect content from network traffic

and bad...

- Gain information for effecting a network intrusion
- Spy on other network users and collect sensitive information such as passwords (depending on any content encryption methods which may be in use)

We will be using Wireshark in order to take a closer look at network communication protocols.

1.1 Getting Wireshark:

Wireshark is free and supported on all Windows, Mac and Linux/Unix machines. It can be obtained by going to: http://www.wireshark.org/download.html

1.2 Warning:

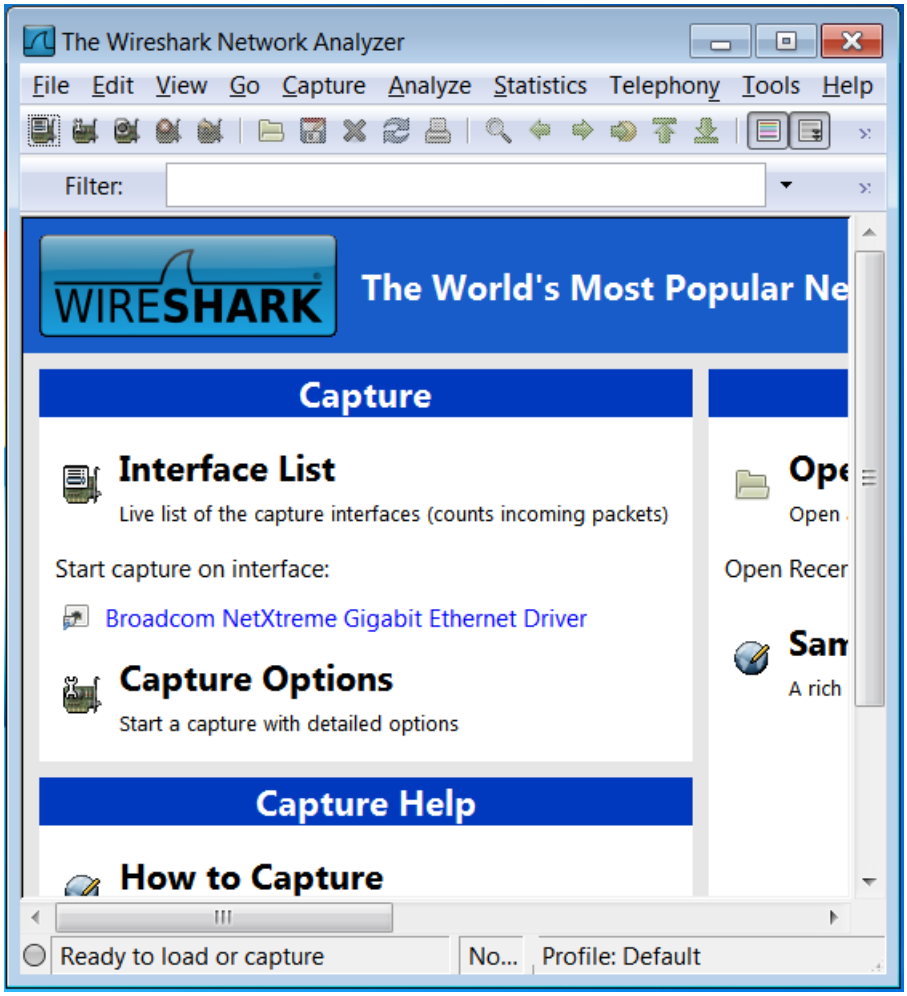
Listening, sniffing, eavesdropping on networks to which you do not have legal access is unethical and may even constitute a crime if used without permission.
On wired broadcast and wireless LANs, to capture traffic other than traffic sent directly to the machine running Wireshark software, the network adapter (network card) being used must be put into "promiscuous mode".

2 Getting Started

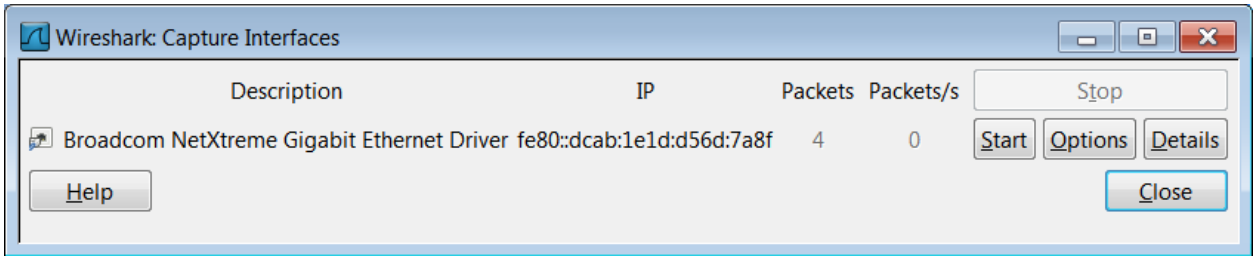
Before you start Wireshark, you will want to clear the cache (or "recent history") on your web-browser (Internet Explorer, Firefox, Safari) and close any open programs. Then locate the Wireshark in the application menus and start the program:

2.1 Select a Network Device

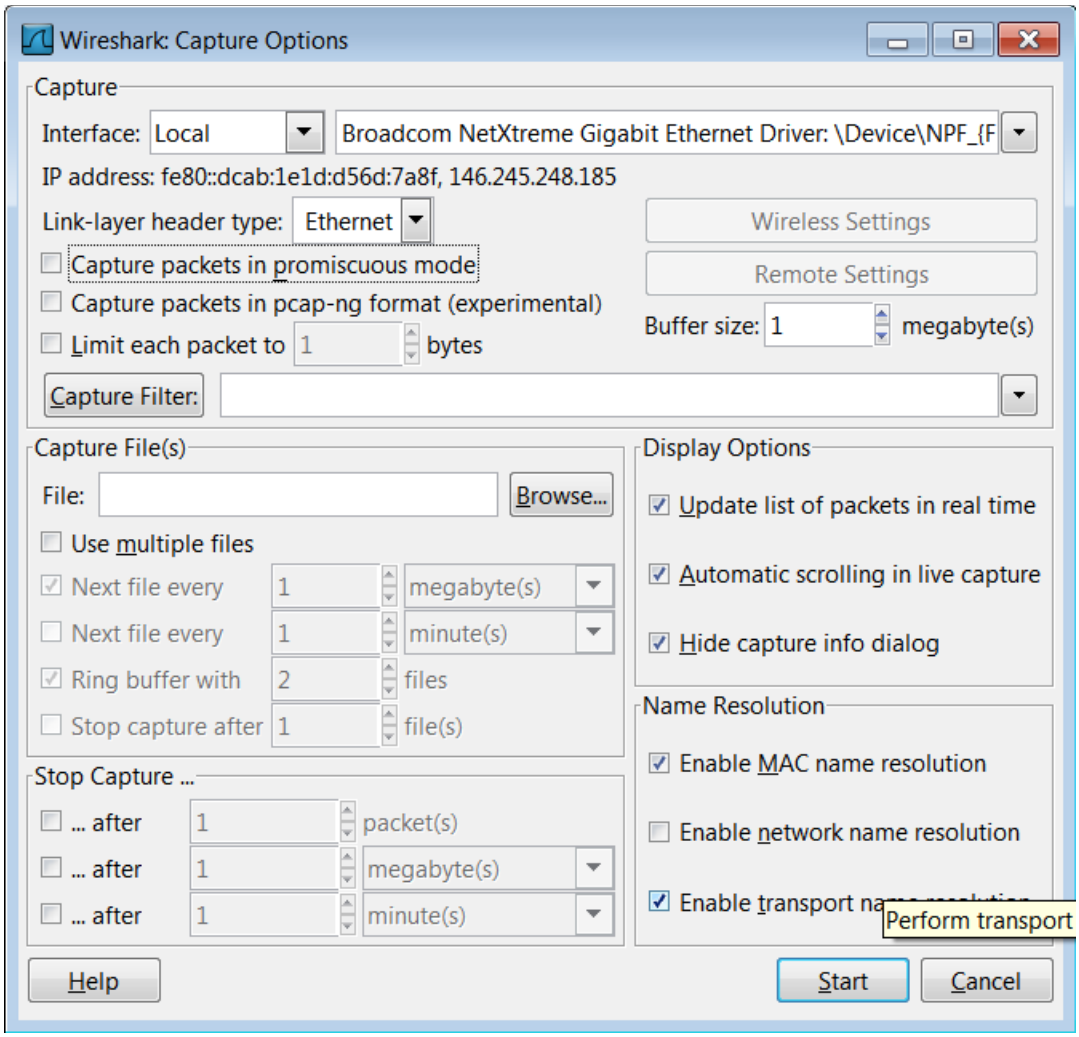
When you first start Wireshark you will be brought to the default startup screen. In order to begin a network capture you will need to choose a network device to use. There are several ways to do this: for now you can simply select the "List the Available Capture Devices" button, circled in red below.



In the pop-up box that appears you will need to select one of the devices available on your machine. Most machines will only have one network interface available. If you have more than one device available you will want to choose an Ethernet device (cable) rather than an 802.11 (wireless) card. Once you have decided on what device to use, click Options.



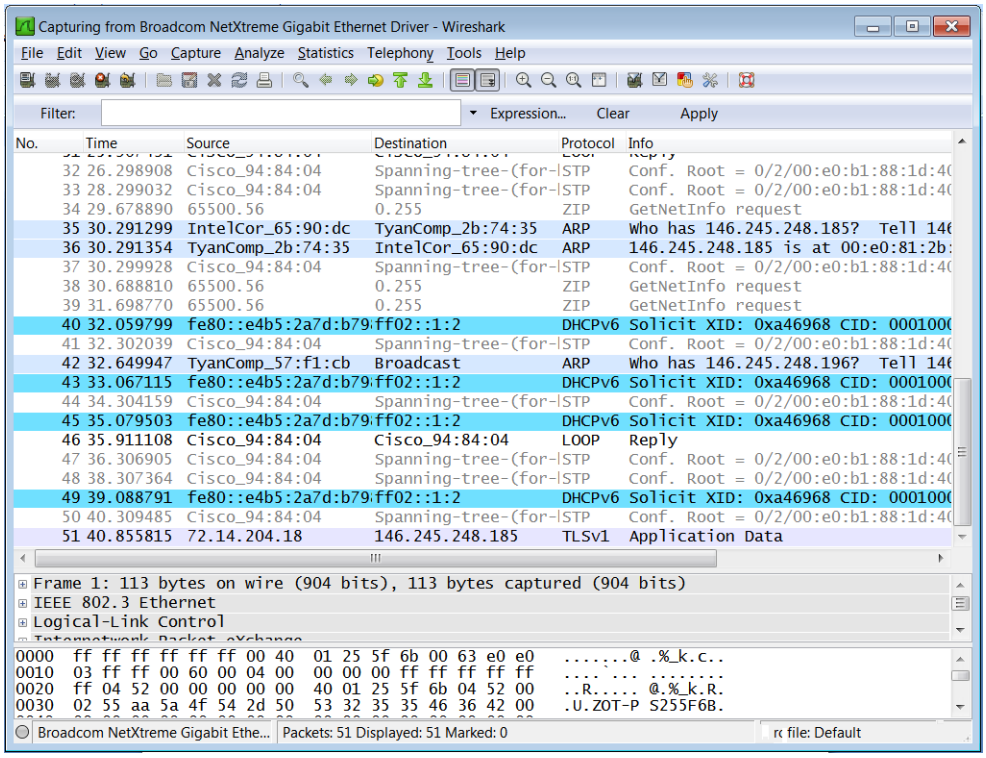
In the "Capture Options" window that appears uncheck the "Capture packets in promiscuous mode" checkbox (we only want to capture messages sent directly to your own machine). The click "Start" in both the "Capture Options" window and then in the "Capture Interfaces" window.



2.2 Capturing Packets

If your Ethernet card is working you should very quickly start to see lines appearing in your packet capture window (see example below). Each line represents a packet (a unit of data) that was sent over the network that Wireshark detected.

Even though you may not be actively requesting any information (such as asking your browser to get you a webpage) your computer and the other devices on your network are constantly sending messages back and forth to each other to keep each other updated about their status, and programs on your computer may also be sending information over the network on their own (software update requests, would be one example.)



The different columns in the display window detail the number and time of packets that were captured, the source and destination for the packets, as well as protocol type and general information about the packet. You can double click on a packet to get more information about it.

2.3 Mac Addresses, IPv4 Addresses, Port Addresses

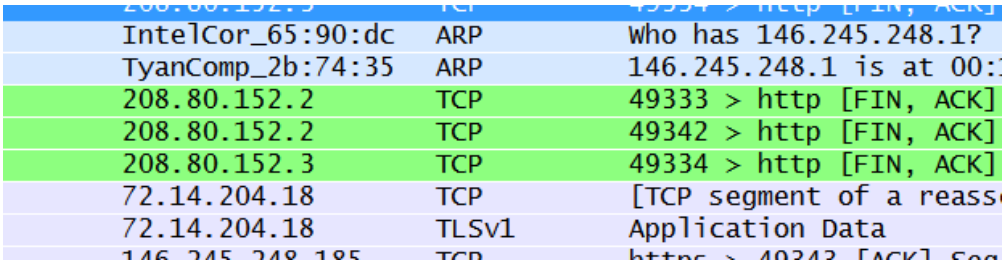
Every network device (network card) has a globally unique MAC (Machine Access Code) Address. Mac Addresses (in theory) can never me changed, so even if a machine is moved to a new network, its Mac Address will remain the same. Mac Addresses are used for communication between network devices that reside on the same physical network segment.

You can think of it like this, a Mac Address is like a person's name (or student number, which is unique). If you and I are in the same physical room, and I want to talk to you, it will be sufficient for me to call you by your name to get your attention.

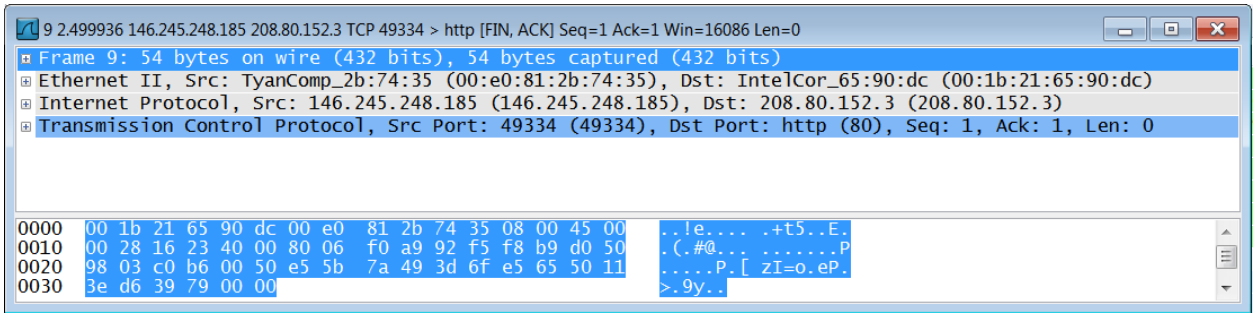
The standard (IEEE 802) format for printing MAC-48 addresses in human-friendly form is six groups of two hexadecimal digits, separated by hyphens (-) or colons (:), in transmission order.

Examples:
01-23-45-67-89-ab
01:23:45:67:89:ab

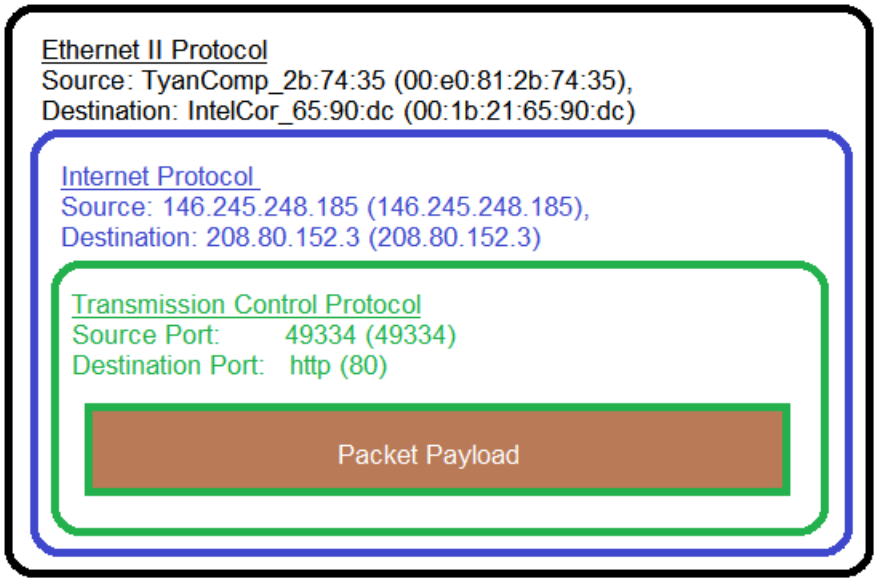
Look in your packet capture window for a packet that is using the TCP protocol.



Double click on the packet line to obtain more details about that specific packet:



In the window above you can see that the frame that I captured and opened to examine what was actually an Ethernet II packet, which contained within it two other packets. The packet selected above is actually a nesting of 3 different protocol packets, as shown in the picture below.



The Ethernet Protocol uses MAC addresses to pass messages between machines that are on the same network.

Not all machines of course are on the same physical network. Two machines that are using the Internet to communicate may be on opposite sides of the world and therefore will need to use **IP (Internet Protocol) Addresses** in order to communicate.

An IP address is a unique number that can be assigned to a network card. The Internet (using routing tables) keeps track of where particular sets of IP addresses are located and how *to route messages to get to those IP addresses.*

You can think of it like this, if you and I are not in the same room and I want to get a message to you, I could simply send you a letter. In order for the letter to reach you, I need to have your address (street, town, postcode). With your address I can post my letter from anywhere in the world and it will eventually get to you. Just as you can move and change your physical address, a computer can also move and change its IP address.

There are two general formats for IP addresses: IPv4 and IPv6. Most computers in the world currently use IPv4 addresses. IPv6 is the successor to the Internet Protocol version 4 (IPv4). In contrast to IPv4, which defined an IP address as a 32-bit number, IPv6 addresses have a size of 128 bits, vastly expanding the addressing capability of the Internet Protocol.

IPv4 addresses are most often written in dot-decimal notation, which consists of the four octets of the address expressed separately in decim**al an**d separated by periods. Example:
192.168.1.1

IPv6 address is represented as eight groups of four hexadecimal digits, each group representing 16 bits (two octets). The groups are separated by colons (:) Leading 0's are sometime omitted. An example of both formats is shown below:
2001:0db8:85a3:0000:0000:8a2e:0370:7334
2001:db8:85a3:0:0:8a2e:370:7334.

A single computer may at any given time be running dozens of different programs (applications). Consequently, when a network packet arrives at a machine, we would like to know what program should receive the information in the packet (if it's a webpage it needs to go to the web-browser, if it's a anti-virus update, it needs to go to the antivirus program). We can make sure that data packets reach the proper programs by using Ports. Many protocols such as Transmission Control Protocol (TCP) and Universal Datagram Protocol (UDP) allow packets to be addressed to unique numbers called ports. These ports, in turn are linked (by convention, and by the operating system) to certain programs. Examples:
Port 80: Used by web-servers
Port 21: Used by FTP Servers
Port 53: Used by DNS Servers

*You can think of it like this, some people live in blocks of flats. Consequently if I want to send a letter to you, it may not be enough to just kn*ow your address; I may also need your flat number.

When we started this packet capture we turned off promiscuous mode, so the packet we are looking at had to have been one that contained a message between our machine and some other machine.
Fill in all of the boxes below:

Source Machine	Destination Machine
Mac Address (Ethernet Protocol):	Mac Address (Ethernet Protocol):
IP Address (IP Protocol):	IP Address (IP Protocol):
Port Address (TCP Protocol):	Port Address (TCP Protocol):

2.4 Determining your actual IP address

2.4.1 On Windows 7:

Step 1: Click Start / Run and type: cmd or command to open a Windows command line window.
Step 2: From the prompt, type ipconfig and press enter.

This should give you information similar to what is shown below.

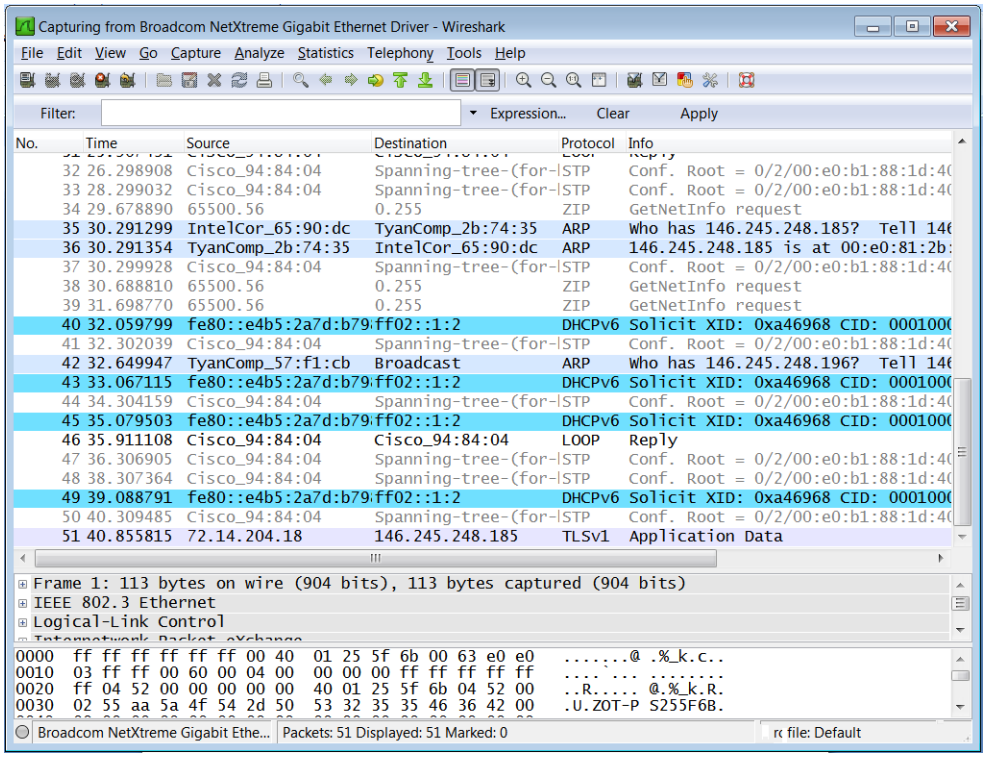
Windows 7 IP Configuration
*Ethernet adapter Local Area Connection: ********
*Connection-specific DNS Suffix . : *********
IP Address: 192.168.1.101
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.1.1

2.4.2 On Linux or Mac:

Step 1: Open the Linux or Unix shell if you are utilizing a GUI interface for your Linux or Unix machine.
Step 2: From the prompt, type "ifconfig" (without the quotes) and press enter.
This should give you a listing of network information.

2.5 Stopping/Saving a Capture

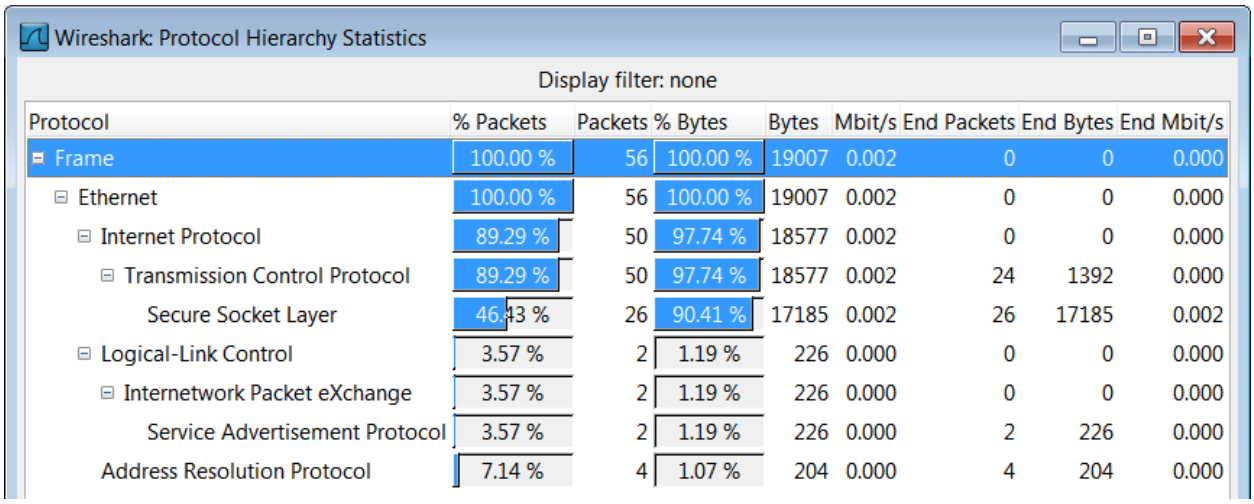
By now your capture (which we left running) should have acquired a lot of data. Let's stop our capture and then save it so that we can use it later. We do this by hitting the "stop capture" button (red cross).



After you have stopped your capture save it by clicking on the "save capture button" (found to the right of the "stop capture" button. Give the saved capture a name you will remember and save it in your class folder.

3 Analyzing a Saved Capture

Wireshark has many different tools to help users filter and analyze the contents of a capture. Open a previous capture file (you may have one already open) and then in the menu bar click on **Statistics** and then **Protocol Hierarchy.** You should get a window that looks something like this:



In this window you can see the types of kinds of all protocols that were being used during the capture. Using the example above we can see that:

89.29% of all messages were using the IP protocol
3.57% were using Logic-Link Control Protocols
4.14% were using the ARP protocol

Note that 100% of the messages that used the IP protocol (had IP headers) also contained the TCP protocol (had TCP headers) [89.29% = 89.29%].

ARP and other "Logical-Link" protocols are protocols that are used by the routers and other Internet devices to keep their routing tables up to date.

Close the "Hierarchy Statistics" window and then in the menu bar click on **Statistics** and then **Conversations.** The window that opens will have tabs across the top that allow you to examine what conversations were taking place during your capture based on protocol and based on source and destination.

4 Examining Captured Data

Make sure you cleared the cache (or "recent history") on your web-browser (Internet Explorer, Firefox, Safari) and that you have closed any other open programs. Do the following:

Step 1: Start a new packet capture.

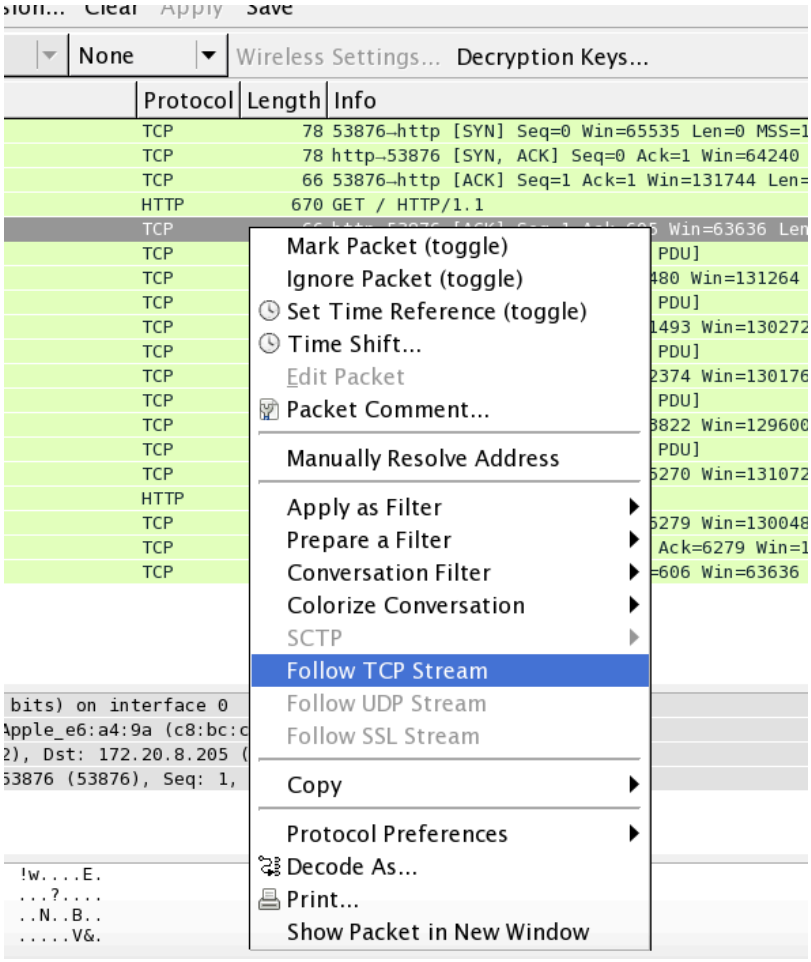
Step 2: Open a browser.

Step 3: Type in the address of a webpage into your browser window:
http://www.somewherenice.com/

Step 4: Go back to the packet capture window and look for a section of green packets that look like the following:

137	4.832247000	www.rgu.ac.uk	172.20.8.205	TCP	78 http→53877 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=
138	4.832359000	172.20.8.205	www.rgu.ac.uk	TCP	66 53877→http [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=646760486
139	4.832618000	172.20.8.205	www.rgu.ac.uk	HTTP	702 GET /js/jquery-1.7.2.min.js HTTP/1.1
140	4.833772000	www.rgu.ac.uk	172.20.8.205	TCP	1514 [TCP segment of a reassembled PDU]
141	4.834265000	www.rgu.ac.uk	172.20.8.205	TCP	1514 [TCP segment of a reassembled PDU]
142	4.834306000	172.20.8.205	www.rgu.ac.uk	TCP	66 53875→http [ACK] Seq=671 Ack=2897 Win=129600 Len=0 TSval=6467(
143	4.850263000	www.rgu.ac.uk	172.20.8.205	TCP	78 http→53878 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=

Locate the packet that uses the HTTP protocol, and is a "GET" request message to the IP address of the webpage. Right click on that conversation and choose the "Follow TCP Stream" option.



You should see the following (see picture below):

- Red Text indicating the request message (packet header and data).
  Questions:
  What browser are you using?
  Do you see that information in the Get Request?
  What operating system are you using?
  Do you see that information in the Get Request?

- Blue Text indicating the web-servers HTTP response to your Get request.
  Questions:
  What is the type of web-server that is running the Kali Linux site?
  What operating system is the Kali Linux Server using?

- The complete XTHML code to the web-page that you requested!!!

The web-server that you requested information from did not encrypt the data packets that it sent back. As a consequence you should be able to see quite clearly the webpage that was sent to your browser. Any webpage, email, text message (etc.)

that is sent unencrypted over a network can be intercepted and viewed by anyone using a packet sniffer like wire shark.

4.1 Other things to notice.

The TCP Stream window only shows two different messages, but in reality the conversation (request and delivery of a webpage) between your machine and the server took place many messages to complete.

4.1.1 Message Size

Occasionally data files that are very large will be too large to send in one packet. When that happens a large file may be broken up into many smaller packets each of which are sent separately. These packets will all be numbered so that when they arrive at their destination (and they may arrive out of order) the can be put back together.

4.1.2 SYN & ACK

TCP was designed to provide reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. TCP does this using special response in the message headers. In brief two computers having a conversation will label their messages. You can see an example of this in the circled section below.

Source	Destination	Protocol	Info
146.245.250.147	146.245.248.185	TCP	http > 49519 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK
146.245.248.185	146.245.250.147	TCP	49519 > http [ACK] Seq=1 Ack=1 Win=65700 Len=0
146.245.248.185	146.245.250.147	HTTP	GET /collegenow/ HTTP/1.1

5. Next Tasks

5.1 New Captures

Capture traces for the following traffic.

1. Ping the a classmates PC from your PC .