

TCP/UDP using Wireshark

Install Wireshark

Follow the link to download and install the official package of wireshark.

https://www.wireshark.org/docs/wsug_html_chunked/ChBuildInstallWinInstall.html

Capture Packets in Wireshark

Follow the given steps to capture and analyze packets using wireshark. Answer the questions present. Keep screenshots in a folder to support your answers.

Experiment 2: Wireshark as a Network Protocol Analyzer

Learning Objectives:

- To become familiarized with the Wireshark application environment
- To perform basic PDU capture using Wireshark
- To perform basic PDU analysis
- To experiment with Wireshark features and options such as PDU capture, display filtering and following TCP streams
- To define the purpose of network protocol analyzers, such as Wireshark

Background

Wireshark is a software protocol analyzer, or "packet sniffer" application, used for network troubleshooting, analysis, software and protocol development, and education. Before June 2006, Wireshark was known as Ethereal.

It has a rich and powerful feature set and runs on most computing platforms including Windows, OS X, Linux, and UNIX. Network professionals, security experts, developers, and educators around the world use it regularly. It is freely available as open source, and is released under the GNU General Public License version 2.

Wireshark can read live data from Ethernet, Token-Ring, FDDI, serial (PPP and SLIP) (if the OS on which it's running allows Wireshark to do so), 802.11 wireless LAN (if the OS on which it's running allows Wireshark to do so), ATM connections (if the OS on which it's running allows Wireshark to do so), and the "any" device supported on Linux by recent versions of libpcap.

While some people use the advantage of Wireshark for network monitoring, others use Wireshark to capture and analyze telnet and FTP logins and passwords, web traffic, including mail transactions to steal private passwords and personal information from the internet. For security and safety reasons, it is strictly advised that Wireshark should be used responsibly.

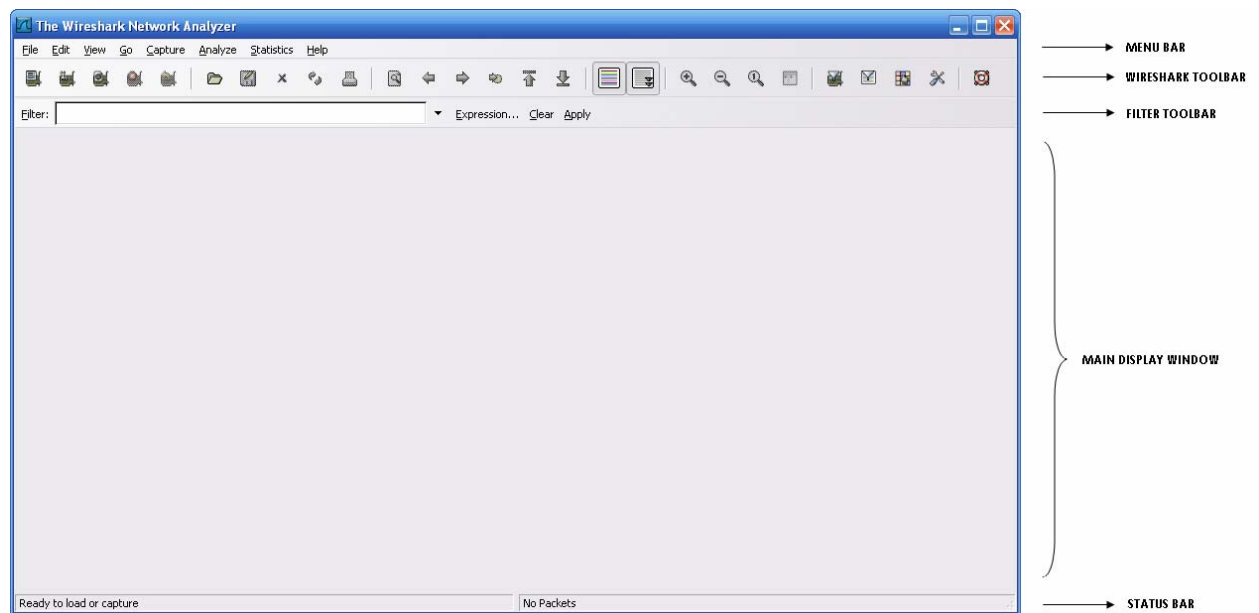
For further information and software download, visit
<http://www.wireshark.org>

For more network security concerns, visit
<http://www.cromwell-intl.com/security/monitoring.html>

PART 1: Wireshark Environment

To capture Protocol Data Units (PDUs), the computer on which Wireshark is installed must have an active connection to a network. Wireshark must be running before any data can be captured.

Open Wireshark application. As the Wireshark is launched, a window similar to the one shown below is displayed.



Explore the Wireshark Environment and answer the following questions:

What selection in the Menu Bar enables one to open and merge capture files, save/print/export capture files in whole or in part, and to quit from Wireshark?






What selection in the Menu Bar contains items to find a packet, time reference or mark one or more packets, and set your preferences?

What selection in the Menu Bar allows one to start and stop captures and to edit capture filters?

What selection in the Menu Bar contains menu-items to display various statistic windows, including a summary of the packets that have been captured, including display protocol hierarchy statistics?

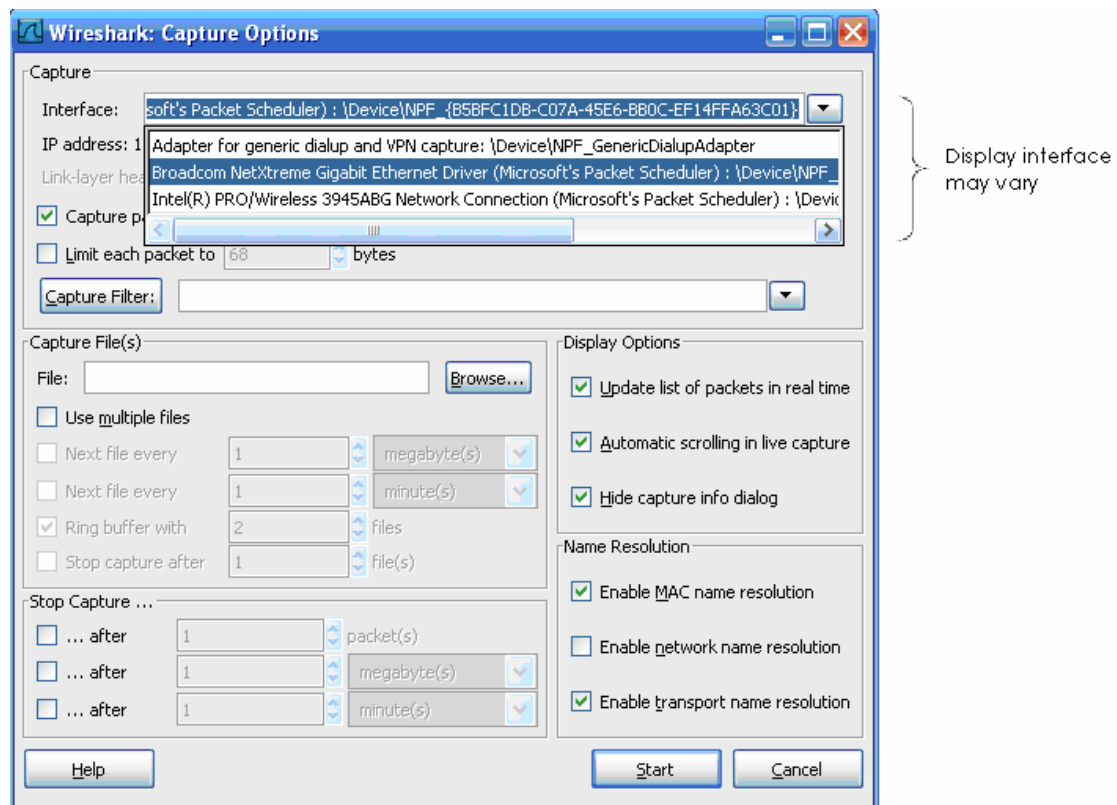
What selection in the Menu Bar contains items to help the user, like access to some basic help, a list of supported protocols, manual pages, online access to some of the webpages, and the usual 'about' dialog?

Draw five buttons present on the Wireshark toolbar and define its functions by having the mouse pointer positioned over the button for a period of time:

Toolbar	Function
	
	
	
	Open capture file
	Getting Help

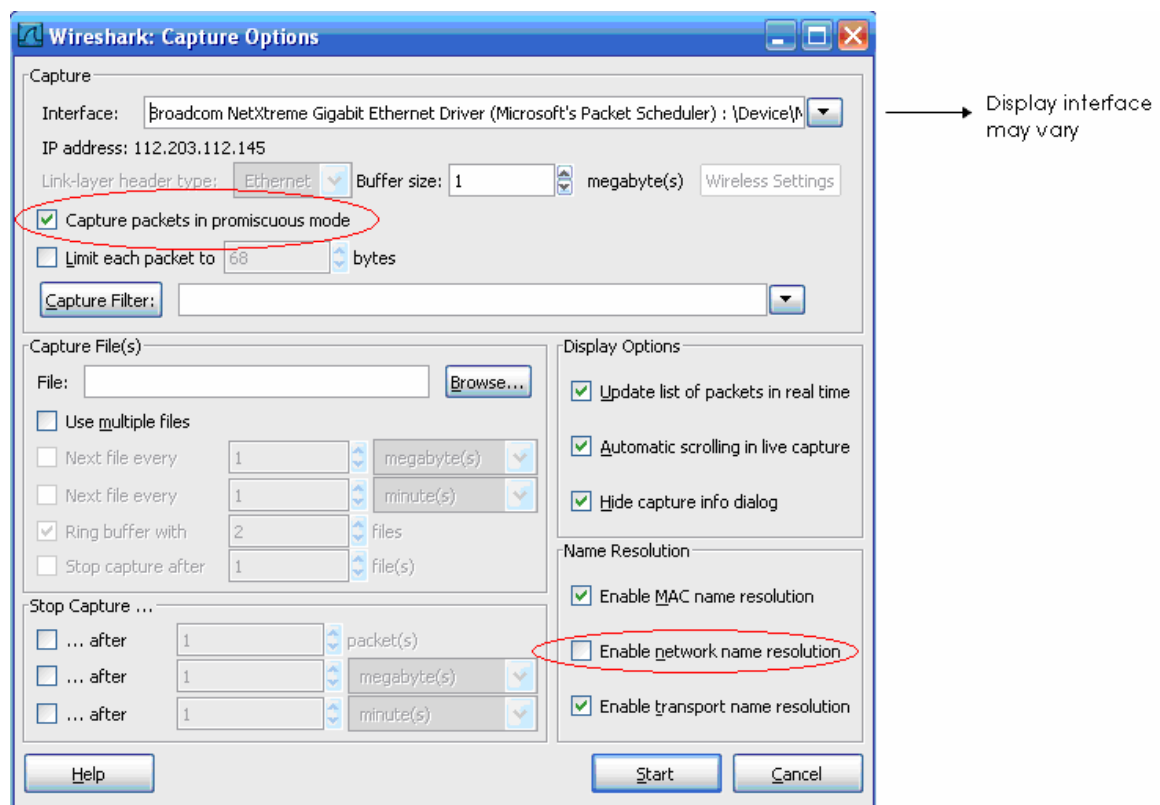
PART II: PDU Capture

To start PDU capture, go to Capture on the Menu bar and select Options. The Capture Options dialog box provides a range of settings and filters which determines which and how much data traffic is captured.



First, it is necessary to ensure that Wireshark is set to monitor the correct interface. From the Interface drop down list, select the network adapter in use. Typically, a computer would be connected to an Ethernet Adapter.

Among those settings available in the Capture Options dialog box, the two selections encircled below are worth examining.



Setting Wireshark to capture packets in promiscuous mode:

If this feature is NOT checked, only PDUs destined for this computer will be captured.

If this feature is checked, all PDUs destined for this computer AND all those detected by the computer NIC on the same network segment are captured.

NOTE: The capturing of these other PDUs depends on the intermediary device connecting the end device computers on the network. As one implement different intermediary devices (hub, switches and routers), different Wireshark results are obtained.

Should Wireshark be set into promiscuous mode when analyzing one's own network traffic? Why?

What do you think would be the drawback of having Wireshark in promiscuous mode when analyzing one's own network traffic?

Setting Wireshark for network name resolution:

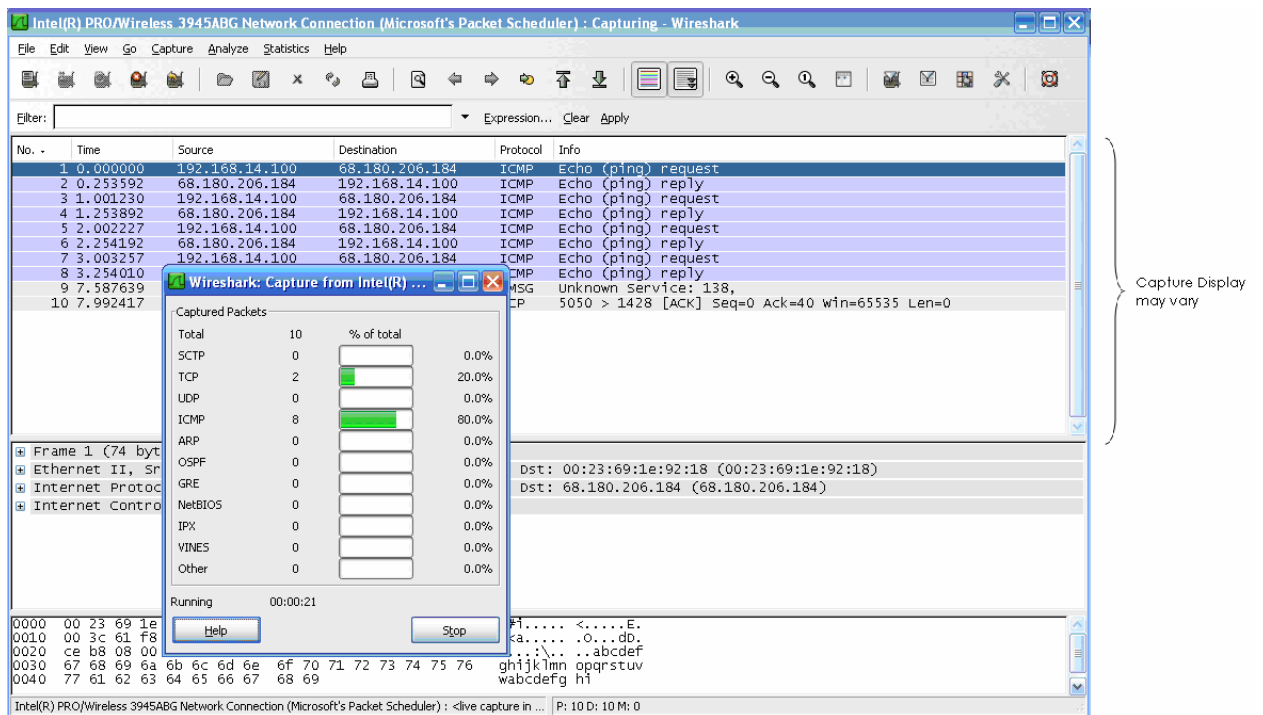
This option allows one to control whether or not Wireshark translates network addresses found in PDUs into names. Although this is a useful feature, the name resolution process may add extra PDUs the captured data, perhaps distorting the analysis.

There are also a number of other capture filtering process settings available under the Wireshark: Capture Options.

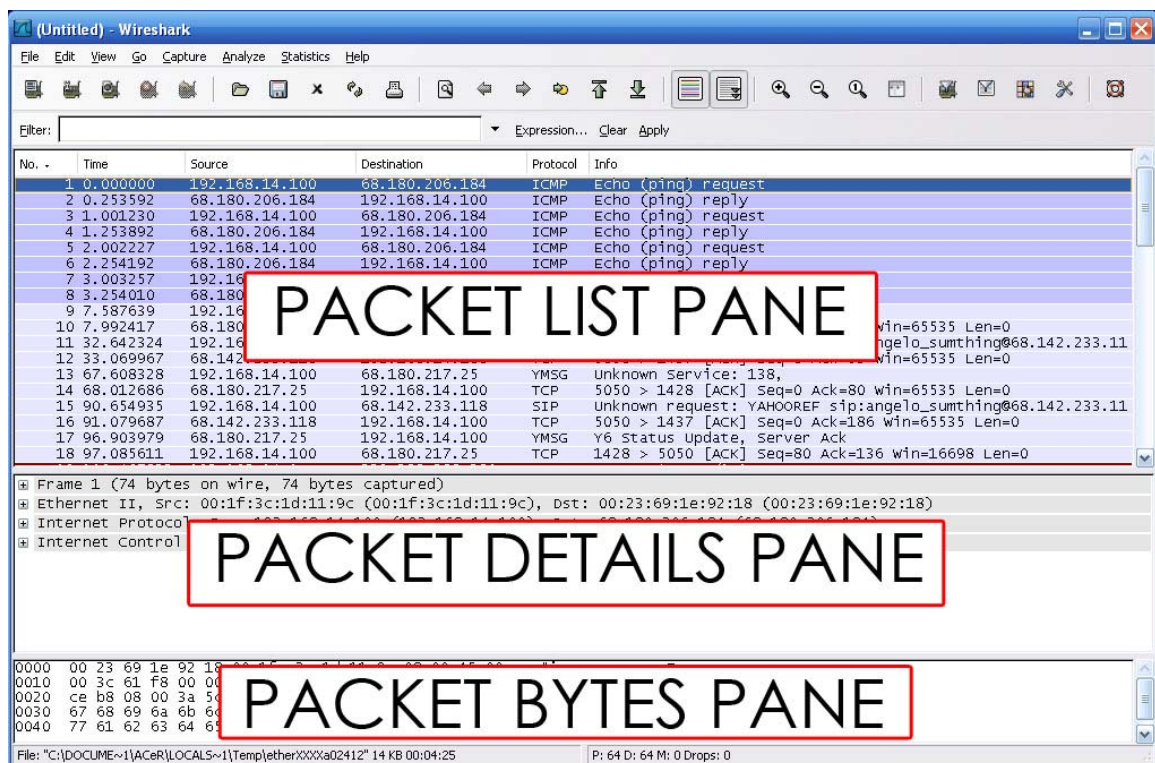
Make sure that the "Capture packets in promiscuous mode" and "Hide capture info dialog" options are unchecked.

Begin Capture Process:

To start data capture process, click the Start button on the Wireshark: Capture Options window. This would show a Capture Information Dialog, and the Main Display Window would then be divided into different panes similar to the window shown below shown below:



When the Stop button is clicked, the capture process is terminated, and the main screen is displayed. This Main Display Window of Wireshark has three panes.



You may explore these three panes later.

Each line in the Packet List corresponds to one PDU or packet of the captured data. If you select a line in this pane, more details will be displayed in the "Packet Details" and "Packet Bytes" panes. The example above shows the PDUs captured when the ping utility was used and [http:// www.yahoo.com](http://www.yahoo.com) was accessed. Packet number 1 is selected in this pane.

The Packet Details pane shows the protocols and protocol fields of the selected packet. The protocols and fields of the packet are displayed using a tree, which can be expanded and collapsed.

The Packet Bytes pane shows the data of the current packet in what is known as “hexdump” style. When a more in-depth analysis is required, this displayed information is useful for examining the binary values and content of PDUs.

The data PDU capture information can be saved in a file. This file can be opened in Wireshark for future analysis without the need to re-capture the same data traffic again. When closing a data capture screen or exiting Wireshark, you are prompted to save the captured PDUs, similar to the image shown below:



Clicking on Continue without Saving closes the file or exits Wireshark without saving the displayed captured data.

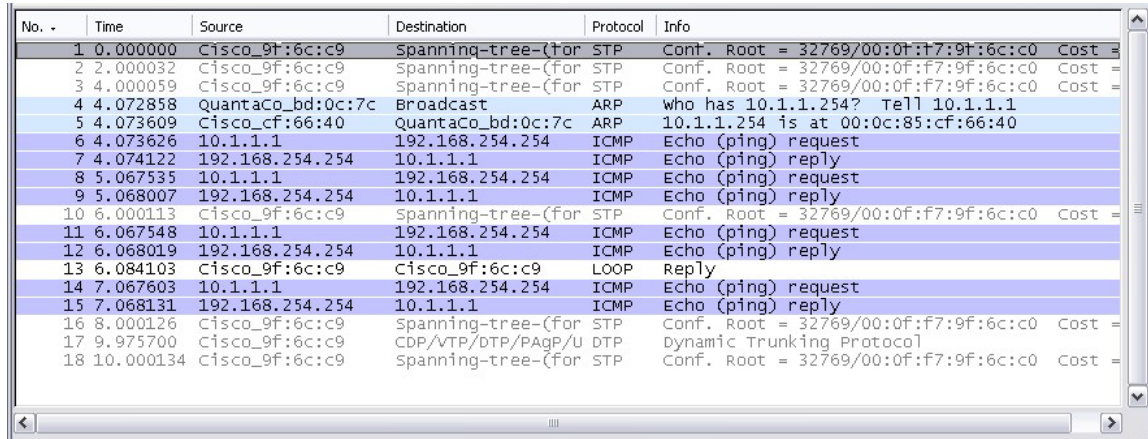
PART III: Analyzing Sample PDU Captures

A. Ping PDU Capture

Note: Your instructor will be providing a sample Wireshark capture of a Ping PDU. Analyze the file to answer the following questions:

Step 1: **Examine the Packet List pane.**

The Packet List pane on Wireshark should now look something like this:



No.	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
2	2.000032	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
3	4.000059	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
4	4.072858	QuantaCo_bd:0c:7c	Broadcast	ARP	who has 10.1.1.254? Tell 10.1.1.1
5	4.073609	Cisco_cf:66:40	QuantaCo_bd:0c:7c	ARP	10.1.1.254 is at 00:0c:85:cf:66:40
6	4.073626	10.1.1.1	192.168.254.254	ICMP	Echo (ping) request
7	4.074122	192.168.254.254	10.1.1.1	ICMP	Echo (ping) reply
8	5.067535	10.1.1.1	192.168.254.254	ICMP	Echo (ping) request
9	5.068007	192.168.254.254	10.1.1.1	ICMP	Echo (ping) reply
10	6.000113	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
11	6.067548	10.1.1.1	192.168.254.254	ICMP	Echo (ping) request
12	6.068019	192.168.254.254	10.1.1.1	ICMP	Echo (ping) reply
13	6.084103	Cisco_9f:6c:c9	Cisco_9f:6c:c9	LOOP	Reply
14	7.067603	10.1.1.1	192.168.254.254	ICMP	Echo (ping) request
15	7.068131	192.168.254.254	10.1.1.1	ICMP	Echo (ping) reply
16	8.000126	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
17	9.975700	Cisco_9f:6c:c9	CDP/VTP/DTP/PagP/U	DTP	dynamic Trunking Protocol
18	10.000134	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =

Look at the packets listed above; we are interested in packet numbers 6, 7, 8, 11, 12, 14 and 15.

From the Wireshark Packet List answer the following:

What protocol is used by ping? Do not give your answer in acronym form.

What are the names of the two ping messages?

Step 2: Select (highlight) the first echo request packet on the list with the mouse.

The Packet Detail pane will now display something similar to:

```
+ Frame 6 (74 bytes on wire, 74 bytes captured)
+ Ethernet II, Src: QuantaCo_bd:0c:7c (00:c0:9f:bd:0c:7c), Dst: Cisco_cf:66:40 (00:0c:85:cf:66:40)
+ Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 192.168.254.254 (192.168.254.254)
+ Internet Control Message Protocol
```

Click on each of the four "+" to expand the information.

The packet Detail Pane will now be similar to:

```
- Frame 6 (74 bytes on wire, 74 bytes captured)
  Arrival Time: Jan 10, 2007 01:54:07.860436000
  [Time delta from previous packet: 0.000017000 seconds]
  [Time since reference or first frame: 4.073626000 seconds]
  Frame Number: 6
  Packet Length: 74 bytes
  Capture Length: 74 bytes
  [Frame is marked: False]
  [Protocols in frame: eth:ip:icmp:data]
  [Coloring Rule Name: ICMP]
  [Coloring Rule String: icmp]
- Ethernet II, Src: QuantaCo_bd:0c:7c (00:c0:9f:bd:0c:7c), Dst: Cisco_cf:66:40 (00:0c:85:cf:66:40)
  + Destination: Cisco_cf:66:40 (00:0c:85:cf:66:40)
  + Source: QuantaCo_bd:0c:7c (00:c0:9f:bd:0c:7c)
  Type: IP (0x0800)
- Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 192.168.254.254 (192.168.254.254)
  Version: 4
  Header length: 20 bytes
  + Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 60
  Identification: 0x0bf7 (3063)
  + Flags: 0x00
  Fragment offset: 0
  Time to live: 128
  Protocol: ICMP (0x01)
  + Header checksum: 0x6421 [correct]
  Source: 10.1.1.1 (10.1.1.1)
  Destination: 192.168.254.254 (192.168.254.254)
- Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x2a5c [correct]
  Identifier: 0x0300
  Sequence number: 0x2000
```

As you can see, the details for each section and protocol can be expanded further. Spend some time scrolling through this information. At this stage of this experiment, you may not fully understand the information displayed but make a note of the information you do recognize.

As you select a line in the Packets Detail pane all or part of the information in the Packet Bytes pane also becomes highlighted.

For example, if the second line (+ Ethernet II) is highlighted in the Details pane the Bytes pane now highlights the corresponding values.

0000	00 0c 85 cf 66 40 00 c0 9f bd 0c 7c 08 00 45 00	...f@... ...E.
0010	00 3c 0b f7 00 00 80 01 64 21 0a 01 01 01 c0 a8	.<..... d!.....
0020	fe fe 08 00 2a 5c 03 00 20 00 61 62 63 64 65 66	...*\... .abcdef
0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	ghijklmn opqrstuv
0040	77 61 62 63 64 65 66 67 68 69	wabcdefg hi

This shows the particular binary values that represent that information in the PDU. At this stage of the course, it is not necessary to understand this information in detail.

Locate the two different types of 'Source' and 'Destination'. What do these addresses refer to?

Analyze the frames with the first echo request and echo reply and complete the table below.

	First Echo Request	First Echo Reply
Frame Number		
Source IP Address		
Destination IP Address		
ICMP Type value		
ICMP Code value		
Source Ethernet Address		
Destination Ethernet Address		
Internet Protocol version		
Time to Live (TTL) value		

B. HTTP PDU Capture

Note: Your instructor will be providing a sample Wireshark capture of a Ping PDU. Analyze the file to answer the following questions:

The sample captured file shows the interaction of a host device accessing a website with a web browser.

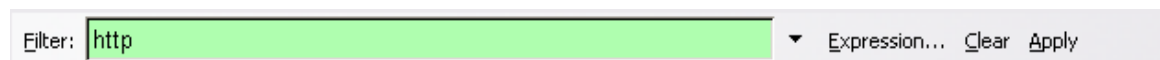
What do you think is the name of the web site accessed by the host?

What protocol was used in resolving the website name to a corresponding IP address by doing a standard name query?

Using Wireshark's Filter feature:

When analyzing a capture file, one might prefer filtering the captured packets concerning specific protocols. Filtering of packets according to the protocols associated with them can be done using the Wireshark's Filter Toolbar.

On the Filter Toolbar, type-in http and press Enter as shown below:



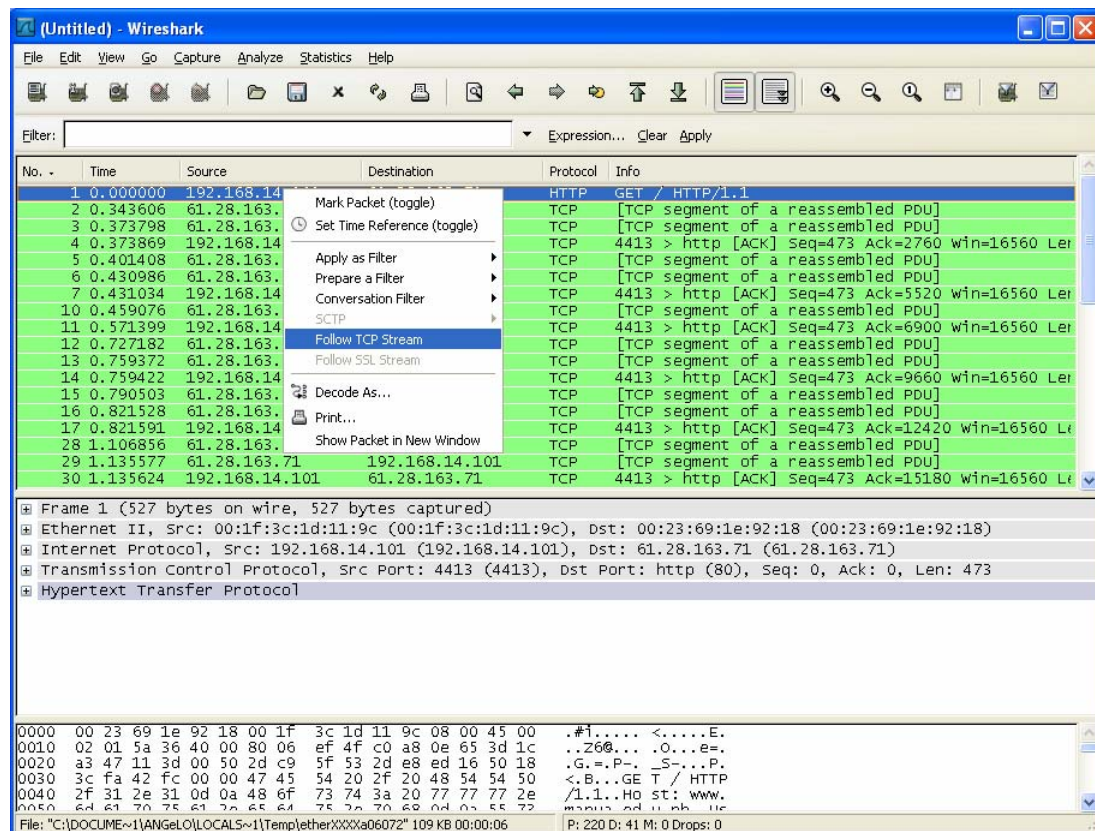
The first frame shows the interaction of the host to the web server and the second frame shows the response of the server to the client. By analyzing the filtered frames, complete the table below:

	Host to Web Server	Web Server to Host
Frame Number		
Source port		
Destination port		
Source IP Address		
Destination IP Address		
Source Ethernet Address		
Destination Ethernet Address		

Using Wireshark's Follow TCP Stream

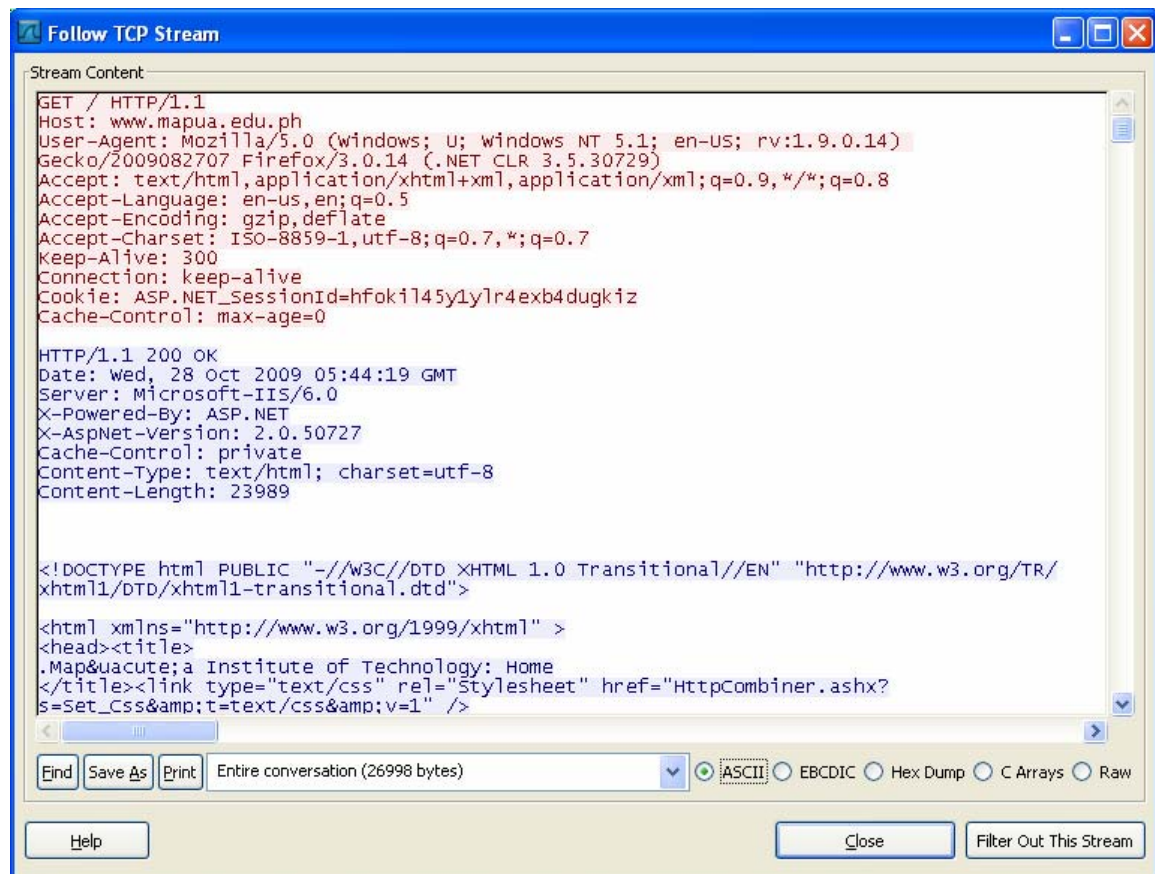
If you are working with TCP based protocols it can be very helpful to see the data from a TCP stream in the way that the application layer sees it. Perhaps you are looking for passwords in a Telnet stream, or you are trying to make sense of a data stream. Maybe you just need a display filter to show only the packets of that TCP stream. If so, Wireshark's ability to follow a TCP stream will be useful to you.

Make sure the filter toolbar is blank. Right-click any packet inside the Packet List Pane, then select "Follow TCP Stream." For the demonstration purposes, a packet containing the HTTP GET request "GET / HTTP/1.1" was the one right-clicked below.



(capture display may vary)

Upon Following a TCP Stream, a window similar to the one below is shown:



The stream content is displayed in the same sequence as it appeared on the network. By default, traffic coming from source to destination is marked in red, while traffic coming from destination to source is marked in blue. One can change these colors in the Edit/Preferences "Colors" page.

NOTE: The stream content won't be updated while doing a live capture. To get the latest content one shall have to reopen the dialog. Non-printable characters will be replaced by dots.

Choose frame 19, then click Follow TCP stream from the Analyze tab. Explore the Stream Content window and answer the following questions:

Based on the color coding explained earlier, what color represents network traffic coming from your computer terminal by default?

Based on the color coding explained earlier, what color represents network traffic coming from the web server?

Based on the Stream Content obtained, what can be observed regarding the information highlighted in red?

Based on the Stream Content obtained, what can be observed regarding the information highlighted in blue?

In the Packet List pane, highlight an HTTP packet that has the notation "(text/html)" in the Info column. **In the Packet Detail pane click on the "+" next to "Line-based text data: html"**

When this information expands what is displayed?

Under Follow TCP Stream, one can also choose to view the data in one of the following formats:

ASCII	In this view you see the data from each direction in ASCII.
EBCDIC	For viewing IBM codes representing characters as numbers.
HEX Dump	This allows you to see all the data. This will require a lot of screen space and is best used with binary protocols.
C Arrays	This allows you to import the stream data into your own C program.
Raw	This allows you to load the unaltered stream data into a different program for further examination. The display will look the same as the ASCII setting, but "Save As" will result in a binary file.

Which format is best for viewing ASCII based protocols such as HTTP?

Would the Raw format have the same display with the ASCII format?

If the Raw format would look just the same as in ASCII format, then what would be the difference in using the Raw format?

PART IV: Experimenting with Wireshark

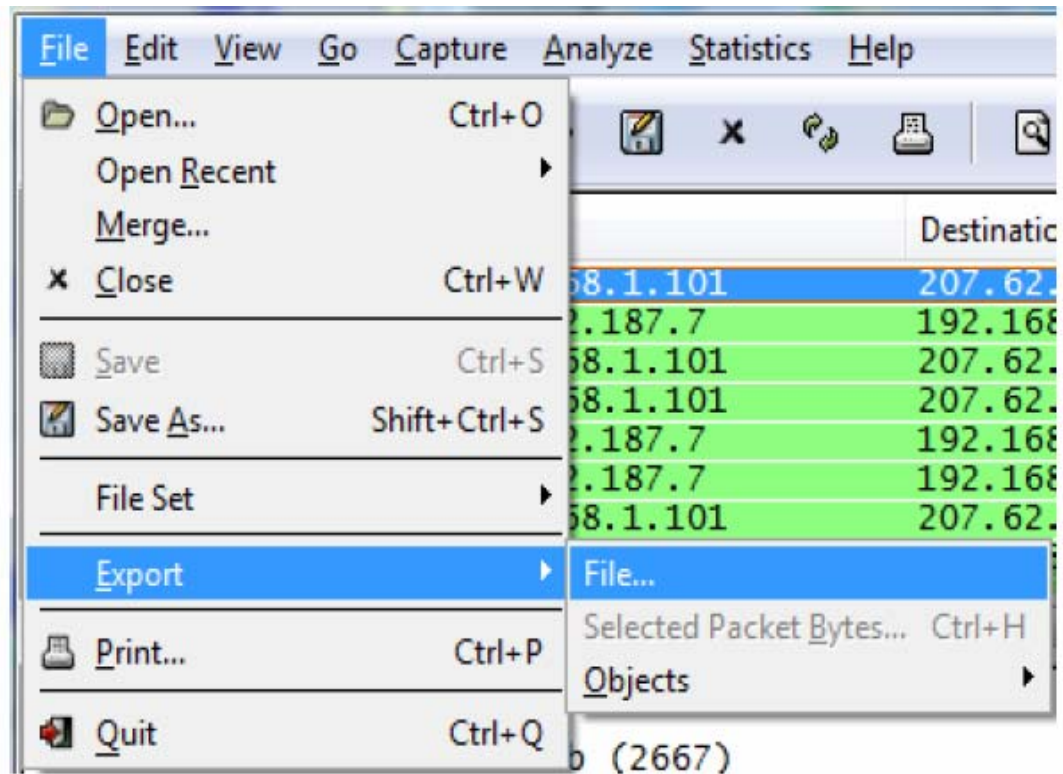
Setup a simple switched network of 3 PCs with one PC acting as web server. Your instructor will assign the IP addresses for PCs and web server.

Set the Capture Options as described above in the overview. On the address bar of the client hosts, input the IP address of the web server on a browser and start the capture process.

Outcome: Save an “expanded” Ethernet frame (http) to a text file and print out the file. This Ethernet frame can include:

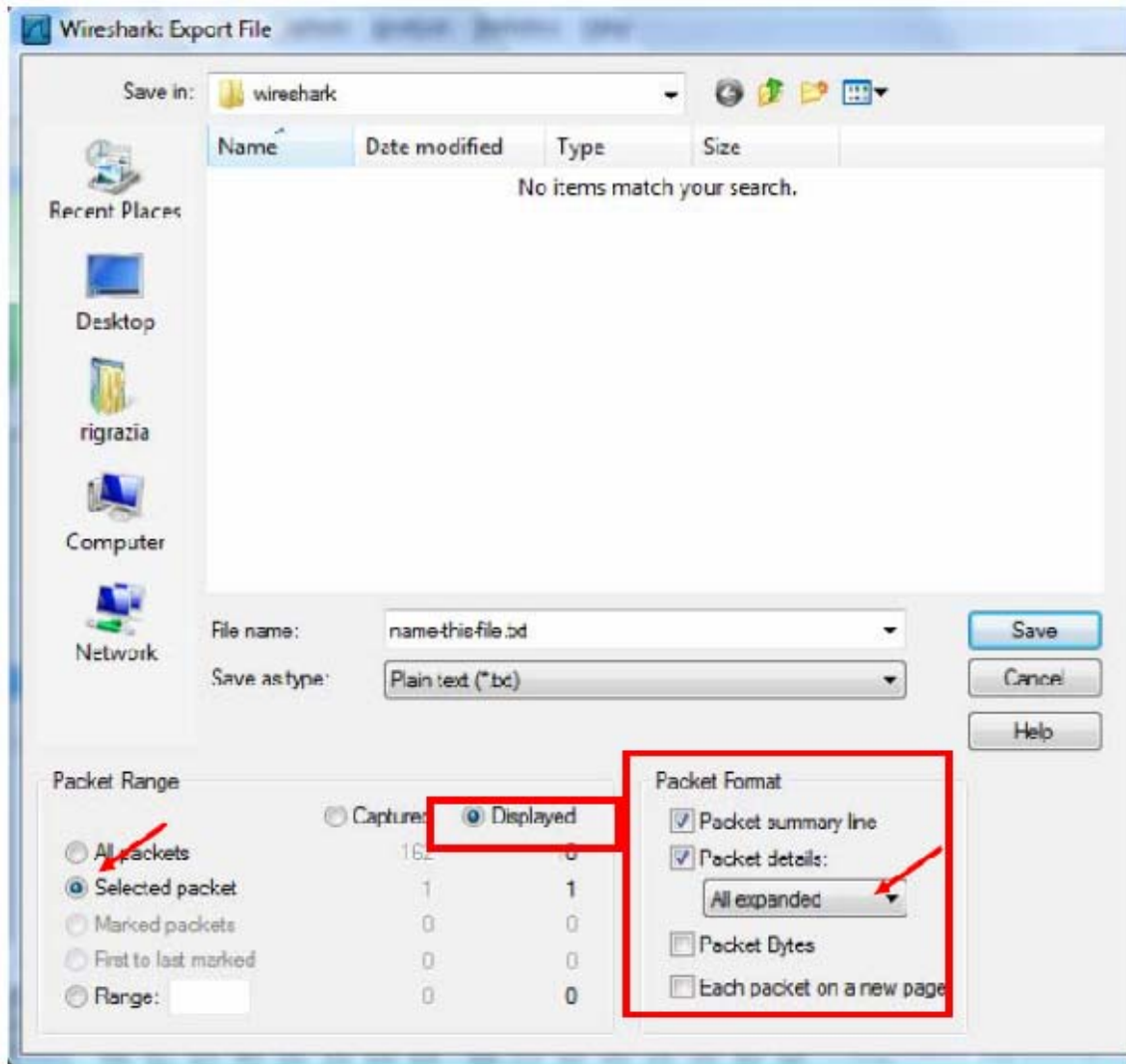
- Ethernet frame
- IP packet
- TCP/UDP header
- Application header and/or Data

To save the Ethernet frames to a text file, choose File >> Export >> File. .



To save a single expanded frame/packet:

- File name: Be sure to use the file extension .txt
- Packet Range: Selected packet
- Click on Displayed (This will be the current frame/packet selected in the display)
- Packet Format:
- Be sure Packet Details is clicked (check in the box)
- Choose: All expanded
- Click "Save"



To save a range of expanded frame/packets:

- File name: Be sure to use the file extension .txt
- Packet Range: Range
- Click on Captured
- Range: First frame/packet – last frame packet. (Example: 2-4)
- Packet Format:
- Be sure Packet Details is clicked (check in the box)
- Choose: All expanded
- Click "Save"

Sample Output

No. Time Source Destination Protocol Info

2 0.344369 192.168.1.101 207.62.187.7 TCP 49323 > http [SYN] Seq=498698563 Len=0 MSS=1460 WS=2

Frame 2 (66 bytes on wire, 66 bytes captured)

Arrival Time: Mar 1, 2008 14:11:23.257549000

[Time delta from previous captured frame: 0.344369000 seconds]

[Time delta from previous displayed frame: 0.344369000 seconds]

[Time since reference or first frame: 0.344369000 seconds]

Frame Number: 2

Frame Length: 66 bytes

Capture Length: 66 bytes

[Frame is marked: False]

[Protocols in frame: eth:ip:tcp]

[Coloring Rule Name: HTTP]

[Coloring Rule String: http || tcp.port == 80]

Ethernet II, Src: QuantaCo_04:a2:1e (00:1b:24:04:a2:1e), Dst: Cisco-Li_09:4e:0f (00:0f:66:09:4e:0f)

Destination: Cisco-Li_09:4e:0f (00:0f:66:09:4e:0f)

Address: Cisco-Li_09:4e:0f (00:0f:66:09:4e:0f)

....0 = IG bit: Individual address (unicast)

....0 = LG bit: Globally unique address (factory default)

Source: QuantaCo_04:a2:1e (00:1b:24:04:a2:1e)

Address: QuantaCo_04:a2:1e (00:1b:24:04:a2:1e)

....0 = IG bit: Individual address (unicast)

....0 = LG bit: Globally unique address (factory default)

Type: IP (0x0800)

Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 207.62.187.7 (207.62.187.7)

Version: 4

Header length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)

0000 00.. = Differentiated Services Codepoint: Default (0x00)

....0. = ECN-Capable Transport (ECT): 0

....0 = ECN-CE: 0

Total Length: 52

Identification: 0x0a6b (2667)

Flags: 0x04 (Don't Fragment)

0... = Reserved bit: Not set

.1.. = Don't fragment: Set

..0. = More fragments: Not set

Fragment offset: 0

Time to live: 128

Protocol: TCP (0x06)

Header checksum: 0xa405 [correct]

[Good: True]

[Bad : False]

Source: 192.168.1.101 (192.168.1.101)

Destination: 207.62.187.7 (207.62.187.7)

Transmission Control Protocol, Src Port: 49323 (49323), Dst Port: http (80), Seq: 498698563, Len: 0

Source port: 49323 (49323)

Destination port: http (80)

Sequence number: 498698563

Header length: 32 bytes

Flags: 0x02 (SYN)

0... = Congestion Window Reduced (CWR): Not set

.0.. = ECN-Echo: Not set

..0. = Urgent: Not set

...0 = Acknowledgment: Not set

.... 0... = Push: Not set

.... .0.. = Reset: Not set

.... ..1. = Syn: Set

.... ...0 = Fin: Not set

Window size: 8192

Checksum: 0x9aca [correct]

[Good Checksum: True]

[Bad Checksum: False]

Options: (12 bytes)

Maximum segment size: 1460 bytes

NOP

Window scale: 2 (multiply by 4)

NOP
NOP
SACK permitted

TCP/UDP using Wireshark

PART-II - Check TCP Handshaking using Wireshark

- Before starting to analyze the TCP handshake from Wireshark, make sure you choose an internet enabled interface to connect to the internet (which is how we will connect to the website). Also, you will need to click on the stop button once the connection is been established, to monitor only the initial packets separately without mixing things up.
- You can filter the results according to the
 - **protocol** (Eg *tcp*)
 - **protocol and port** (either source or destination port) (Eg *tcp.port eq 80*)
 - **IP address** (Eg *ip.src==192.168.0.103*) (Eg *ip.dst==192.168.0.103*)
 - Based on byte sequence in the payload , use the **contains** filter with the **protocol** name and **byte** sequence.(Eg *tcp contains 00:01:02*)
 - Other complex conditions using **or** or **and** (Eg *tcp and dns*)
 - Adding **not** in the front of a statement negates it
- Now lets narrow it down a little bit to only monitor the tcp traffic
-

Step-1 - All Traffic

The image shows a Wireshark network traffic capture window titled "Wi-Fi". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar with icons for file operations, capture, analysis, and display. A display filter bar at the top shows "Apply a display filter ... <Ctrl-/>".

The main packet list pane displays 13 captured packets. The selected packet (No. 7) is a TCP SYN packet from 192.168.8.102 to 52.213.14.58. The packet details pane shows the structure of this packet: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.8.102	104.18.27.25	UDP	1287	60888 → 443 Len=1245
2	0.000152	192.168.8.102	104.18.27.25	UDP	1069	60888 → 443 Len=1027
3	0.062966	104.18.27.25	192.168.8.102	UDP	67	443 → 60888 Len=25
4	0.063556	104.18.27.25	192.168.8.102	UDP	1242	443 → 60888 Len=1200
5	0.063556	104.18.27.25	192.168.8.102	UDP	326	443 → 60888 Len=284
6	0.070785	192.168.8.102	104.18.27.25	UDP	87	60888 → 443 Len=45
7	0.441260	192.168.8.102	52.213.14.58	TCP	66	60561 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PE...
8	0.571830	192.168.8.102	34.252.240.59	TLSv1.2	226	Application Data
9	0.572029	192.168.8.102	34.252.240.59	TLSv1.2	100	Application Data
10	0.572124	192.168.8.102	34.252.240.59	TLSv1.2	1845	Application Data
11	0.761400	192.168.8.102	13.227.254.28	TLSv1.2	167	Application Data
12	0.761495	192.168.8.102	13.227.254.28	TLSv1.2	93	Application Data
13	0.764722	192.168.8.102	13.227.254.28	TLSv1.2	166	Application Data

Packet details for Frame 7:

- Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{D151C30D-D6EC-40F8-9310-81115D651983}.
- Ethernet II, Src: IntelCor_f8:b4:c0 (58:96:1d:f8:b4:c0), Dst: HuaweiTe_bc:59:0f (10:44:00:bc:59:0f)
- Internet Protocol Version 4, Src: 192.168.8.102, Dst: 52.213.14.58
- Transmission Control Protocol, Src Port: 60561, Dst Port: 443, Seq: 0, Len: 0

The packet bytes pane shows the raw data: 0000 10 44 00 bc 59 0f 58 96 1d f8 b4 c0 08 00 45 00 -D..Y.X.E..

The status bar at the bottom indicates: wireshark_Wi-FiOW6401.pcapng | Packets: 137 · Displayed: 137 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

Step-2 Filter TCP traffic

The image shows a Wireshark network traffic capture. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. A filter bar at the top of the packet list contains the text "hot udp and not tis". The packet list table below shows several TCP packets. Packet 7 is highlighted, and its details are shown in the bottom pane. The details pane shows the frame structure: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
7	0.441260	192.168.8.102	52.213.14.58	TCP	66	60561 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PE...
15	0.779663	52.213.14.58	192.168.8.102	TCP	66	443 → 60561 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1400 SAC...
16	0.779805	192.168.8.102	52.213.14.58	TCP	54	60561 → 443 [ACK] Seq=1 Ack=1 Win=65792 Len=0
18	0.780283	34.252.240.59	192.168.8.102	TCP	54	443 → 60538 [ACK] Seq=1 Ack=2010 Win=626 Len=0
22	0.780335	192.168.8.102	34.252.240.59	TCP	54	60538 → 443 [ACK] Seq=2010 Ack=460 Win=256 Len=0
24	0.882062	13.227.254.28	192.168.8.102	TCP	54	443 → 60554 [ACK] Seq=1 Ack=114 Win=133 Len=0
25	0.882687	13.227.254.28	192.168.8.102	TCP	54	443 → 60554 [ACK] Seq=1 Ack=153 Win=133 Len=0
26	0.882687	13.227.254.28	192.168.8.102	TCP	54	443 → 60554 [ACK] Seq=1 Ack=265 Win=133 Len=0
27	0.882687	13.227.254.28	192.168.8.102	TCP	54	443 → 60554 [ACK] Seq=1 Ack=374 Win=133 Len=0
29	0.928865	192.168.8.102	13.227.254.28	TCP	54	60554 → 443 [ACK] Seq=374 Ack=40 Win=256 Len=0
30	0.984451	52.213.14.58	192.168.8.102	TCP	54	443 → 60561 [ACK] Seq=1 Ack=518 Win=28160 Len=0
33	0.985740	192.168.8.102	52.213.14.58	TCP	54	60561 → 443 [ACK] Seq=518 Ack=5358 Win=65792 Len=0
36	1.087663	34.252.240.59	192.168.8.102	TCP	54	443 → 60538 [ACK] Seq=460 Ack=2052 Win=626 Len=0

Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{D151C30D-D6EC-40F8-9310-81115D651983}, Ethernet II, Src: IntelCor_f8:b4:c0 (58:96:1d:f8:b4:c0), Dst: HuaweiTe_bc:59:0f (10:44:00:bc:59:0f)
Internet Protocol Version 4, Src: 192.168.8.102, Dst: 52.213.14.58
Transmission Control Protocol, Src Port: 60561, Dst Port: 443, Seq: 0, Len: 0

0000 10 44 00 bc 59 0f 58 96 1d f8 b4 c0 08 00 45 00 -D-Y-X-E-

wireshark_Wi-FiOW6401.pcapng Packets: 137 · Displayed: 62 (45.3%) · Dropped: 0 (0.0%) Profile: Default

Meaning of Control bits

Also, the control bits are as follows:

- SYN: Synchronize sequence numbers
- ACK: Acknowledgment field significant
- FIN: No more data from sender
- URG: Urgent Pointer field significant
- PSH: Push Function
- RST: Reset the connection

Step-3 Connection Establishment

First the top most packet is sent from the client (192.168.8.102) to the server (52.213.14.58). It is a SYN segment as mentioned under the **Info** section. Basically, a SYN request is used to synchronize the sequence numbers with the server. In this, the **initial sequence number** (ISN) is specified. Along with each packet transferred, the ISN is incremented by 1 when sent to the server. To start a connection, the client and server must synchronize each other's sequence numbers. The **Acknowledgment field** (ACK: 0) is set to zero because it's the first part of the three-way handshake.

The image shows a Wireshark packet capture window titled "Wi-Fi". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar. A filter bar at the top shows "not udp and not tls". The packet list pane displays several packets, with packet 7 selected. The packet details pane shows the structure of the selected packet, and the packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
7	0.441260	192.168.8.102	52.213.14.58	TCP	66	60561 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PE...
15	0.779663	52.213.14.58	192.168.8.102	TCP	66	443 → 60561 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1400 SAC...
16	0.779805	192.168.8.102	52.213.14.58	TCP	54	60561 → 443 [ACK] Seq=1 Ack=1 Win=65792 Len=0
18	0.780283	34.252.240.59	192.168.8.102	TCP	54	443 → 60538 [ACK] Seq=1 Ack=2010 Win=626 Len=0
22	0.780335	192.168.8.102	34.252.240.59	TCP	54	60538 → 443 [ACK] Seq=2010 Ack=460 Win=256 Len=0
24	0.882062	13.227.254.28	192.168.8.102	TCP	54	443 → 60554 [ACK] Seq=1 Ack=114 Win=133 Len=0

Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{D151C30D-D6EC-40F8-9310-81115D651983}, Ethernet II, Src: IntelCor_f8:b4:c0 (58:96:1d:f8:b4:c0), Dst: HuaweiTe_bc:59:0f (10:44:00:bc:59:0f) Internet Protocol Version 4, Src: 192.168.8.102, Dst: 52.213.14.58 Transmission Control Protocol, Src Port: 60561, Dst Port: 443, Seq: 0, Len: 0

```
0000  10 44 00 bc 59 0f 58 96 1d f8 b4 c0 08 00 45 00  -D-Y-X-...-E-
0010  00 34 d1 89 40 00 80 06 00 00 c0 a8 08 66 34 d5  -4-@-...-f4-
0020  0e 3a ec 91 01 bb f8 cc 1e 71 00 00 00 00 80 02  -.:.....-q-...
0030  fa f0 0c 44 00 00 02 04 05 b4 01 03 03 08 01 01  -...D-...-...
0040  04 02  ..
```

Wireshark_Wi-FiOW64O1.pcapng | Packets: 137 · Displayed: 62 (45.3%) · Dropped: 0 (0.0%) | Profile: Default

Step-4

The second frame is the act of the server sending back the ACK and SYN segment back to the client. The server is acknowledging the request of the client for synchronization and is also sending its request to the client for synchronization of its sequence numbers. The ACK number is proof to the client that the ACK is specific to the SYN the client initiated. The process of acknowledging the client's request allows the server to increment the client's sequence number by one and use it as its acknowledgment number.

The image shows a Wireshark packet capture window titled "Wi-Fi". The packet list pane displays several packets, with packet 15 selected. The packet details pane shows the structure of the selected packet, and the packet bytes pane shows the raw data.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
7	0.441260	192.168.8.102	52.213.14.58	TCP	66	60561 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PE...
15	0.779663	52.213.14.58	192.168.8.102	TCP	66	443 → 60561 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1400 SAC...
16	0.779805	192.168.8.102	52.213.14.58	TCP	54	60561 → 443 [ACK] Seq=1 Ack=1 Win=65792 Len=0
18	0.780283	34.252.240.59	192.168.8.102	TCP	54	443 → 60538 [ACK] Seq=1 Ack=2010 Win=626 Len=0
22	0.780335	192.168.8.102	34.252.240.59	TCP	54	60538 → 443 [ACK] Seq=2010 Ack=460 Win=256 Len=0
24	0.882062	13.227.254.28	192.168.8.102	TCP	54	443 → 60554 [ACK] Seq=1 Ack=114 Win=133 Len=0

Packet Details (Frame 15):

- Frame 15: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{D151C30D-D6EC-40F8-9310-81115D651983}
- Ethernet II, Src: HuaweiTe_bc:59:0f (10:44:00:bc:59:0f), Dst: IntelCor_f8:b4:c0 (58:96:1d:f8:b4:c0)
- Internet Protocol Version 4, Src: 52.213.14.58, Dst: 192.168.8.102
- Transmission Control Protocol, Src Port: 443, Dst Port: 60561, Seq: 0, Ack: 1, Len: 0

Packet Bytes:

```
0000  58 96 1d f8 b4 c0 10 44 00 bc 59 0f 08 00 45 00  X.....D..Y...E.
0010  00 34 00 00 40 00 e7 06 87 a6 34 d5 0e 3a c0 a8  -4-.-.-.-4-:.-.
0020  08 66 01 bb ec 91 7a dd dc 77 f8 cc 1e 72 80 12  -f-.-.-.-Z-.-w-.-r-
0030  69 03 9d 3a 00 00 02 04 05 78 01 01 04 02 01 03  i.-.-.-.-x-.-.-.-
0040  03 08  ..
```

Wireshark Status: wireshark_Wi-FiOW6401.pcapng | Packets: 137 · Displayed: 62 (45.3%) · Dropped: 0 (0.0%) | Profile: Default

Step-5

In the third frame, the client sends an ACK segment, acknowledging the request from the server for synchronization. This completes the process of establishing a reliable connection and the three-way handshake.

The image shows a Wireshark packet capture window titled "Wi-Fi". The packet list on the left shows several frames. Frame 16 is selected, showing a TCP ACK segment from 192.168.8.102 to 52.213.14.58. The packet details pane shows the frame structure: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
7	0.441260	192.168.8.102	52.213.14.58	TCP	66	60561 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PE...
15	0.779663	52.213.14.58	192.168.8.102	TCP	66	443 → 60561 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1400 SAC...
16	0.779805	192.168.8.102	52.213.14.58	TCP	54	60561 → 443 [ACK] Seq=1 Ack=1 Win=65792 Len=0
18	0.780283	34.252.240.59	192.168.8.102	TCP	54	443 → 60538 [ACK] Seq=1 Ack=2010 Win=626 Len=0
22	0.780335	192.168.8.102	34.252.240.59	TCP	54	60538 → 443 [ACK] Seq=2010 Ack=460 Win=256 Len=0
24	0.882062	13.227.254.28	192.168.8.102	TCP	54	443 → 60554 [ACK] Seq=1 Ack=114 Win=133 Len=0

Frame 16: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{D151C30D-D6EC-40F8-9310-81115D651983}

Ethernet II, Src: IntelCor_f8:b4:c0 (58:96:1d:f8:b4:c0), Dst: HuaweiTe_bc:59:0f (10:44:00:bc:59:0f)

Internet Protocol Version 4, Src: 192.168.8.102, Dst: 52.213.14.58

Transmission Control Protocol, Src Port: 60561, Dst Port: 443, Seq: 1, Ack: 1, Len: 0

0000 10 44 00 bc 59 0f 58 96 1d f8 b4 c0 08 00 45 00 .D..Y.X.E.
0010 00 28 d1 8a 40 00 80 06 00 00 c0 a8 08 66 34 d5 .(. @... ..f4.
0020 0e 3a ec 91 01 bb f8 cc 1e 72 7a dd dc 78 50 10 .!:.....rZ..xP.
0030 01 01 0c 38 00 00 ...8..

Questions

- Find the Sequence number, window size and Length for the sender
- Find the sequence number, window size and Length for the receiver