Lab: Using a Protocol Analyzer (Wireshark)

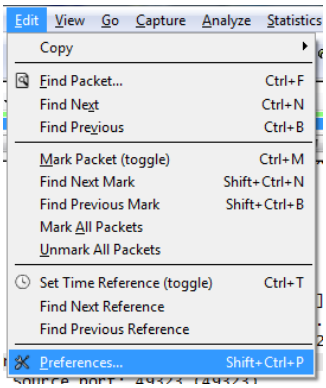
Purpose: The purpose of this lab is to become familiar with a protocol analyzer such as Wireshark, formally know as Ethereal. Wireshark is available for various operating systems including Windows, Linux and Macintosh. Wireshark can be downloaded free, at http://www.wireshark.org/

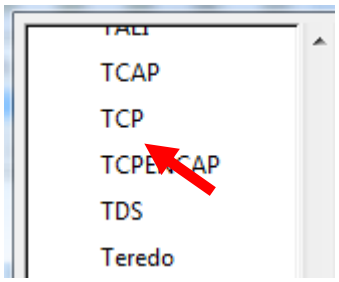
Version used for this assignment: Version 0.99.6a (SVN Rev 22276

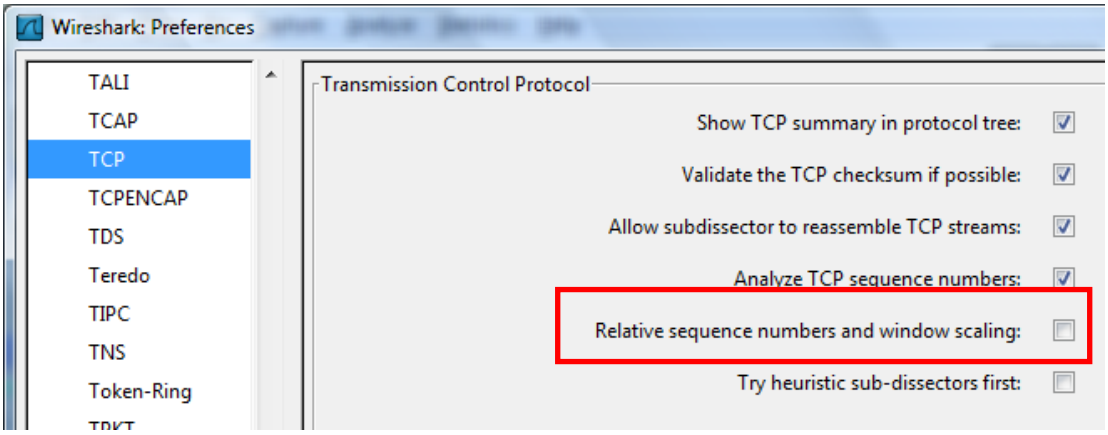
Step 1: Setting up TCP preferences.

By default, Wireshark will show us relative TCP Sequence and Acknowledgement numbers starting with 1. To view the actual TCP Sequence and Acknowledgement numbers sent and received we need to disable this option.

- Choose Edit >> Preferences
- Click on Protocols
- Click on TCP
- Disable "Relative sequence number and window scaling" (be sure it is not checked)

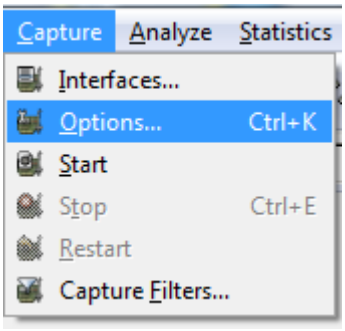


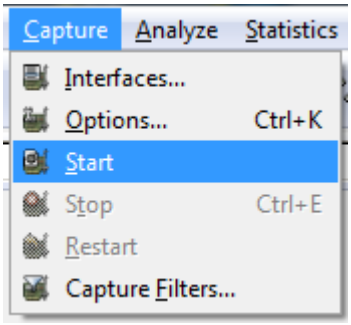




Step 2: Starting the Wireshark Capture

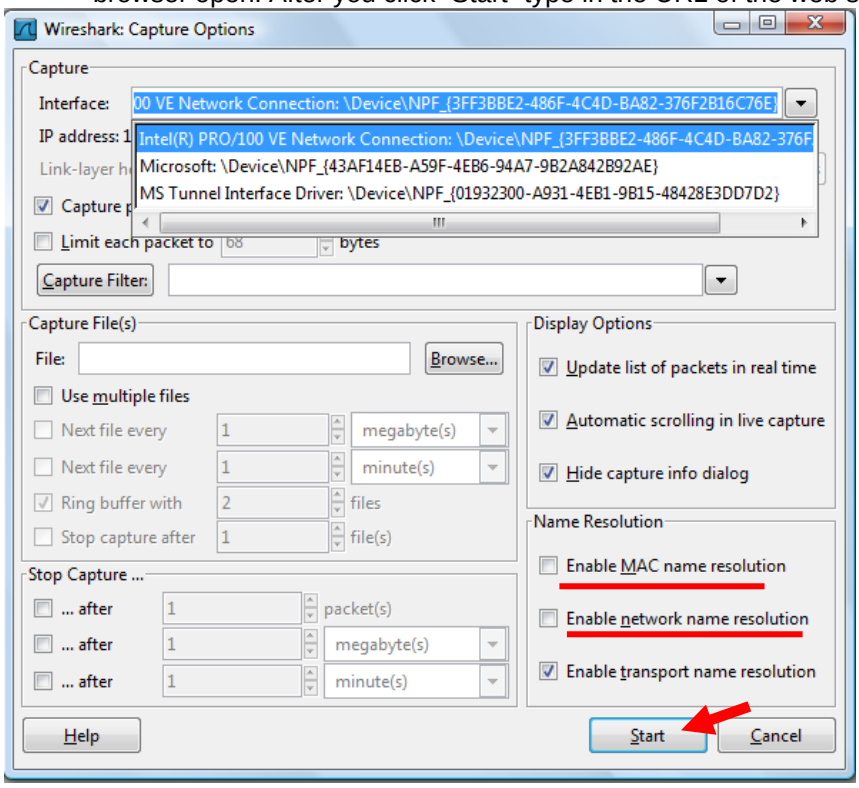
- First time choose::
  - **Capture → Options**
- Subsequent times choose::
  - **Capture → Start**



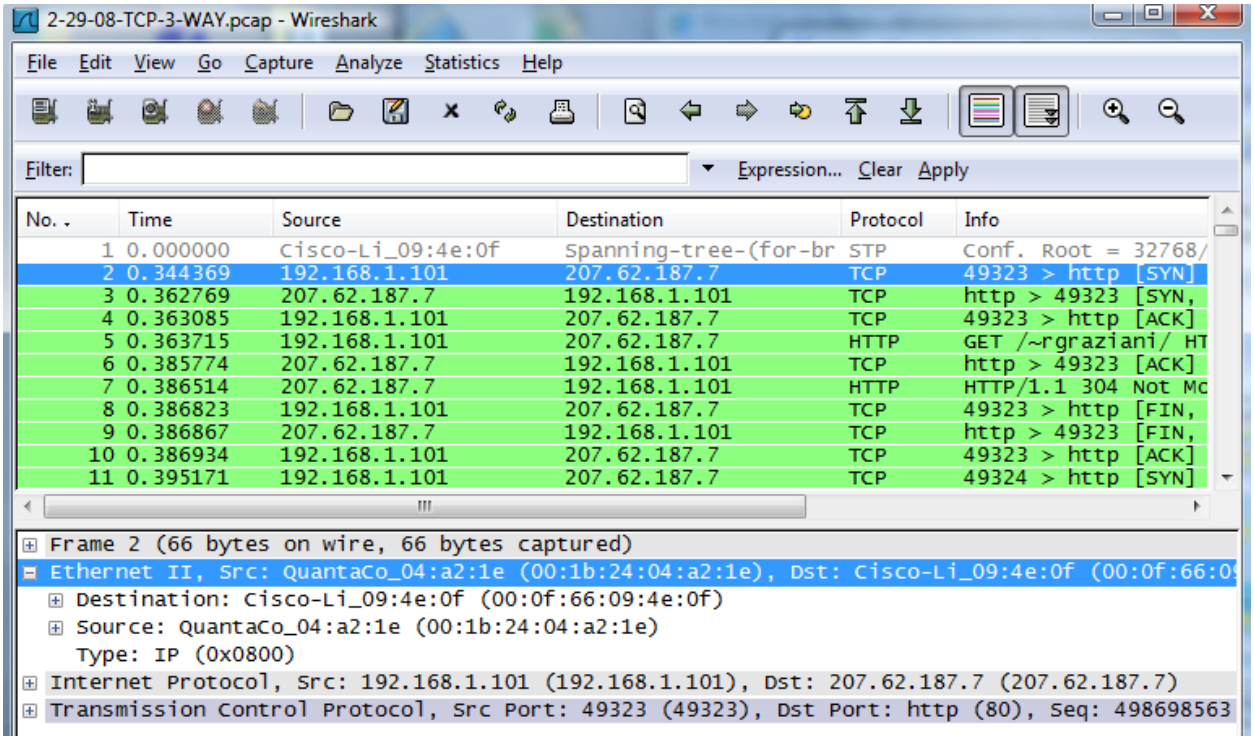


Beginning the capture of frames and packets:

- If you have more than one Ethernet NIC card installed, be sure to choose the proper interface.
- Disable "MAC name resolution" (be sure it is not checked)
- Disable "Network name resolution" (be sure it is not checked)
- Click "Start" to begin the capture (Note: Before clicking on "Start" you may wish to have a web browser open. After you click "Start" type in the URL of the web site you wish to download.)



At this point Wireshark begins capturing packets including unicasts directed for the MAC address of your Ethernet NIC, broadcasts for all devices, multicasts, and unknown unicasts (unicasts flooded by the switch when the Destination MAC address is not in its MAC Address Table).



If you want to capture specific types of packets such as HTTP, FTP or ICMP, perform that operation now.

Examples:

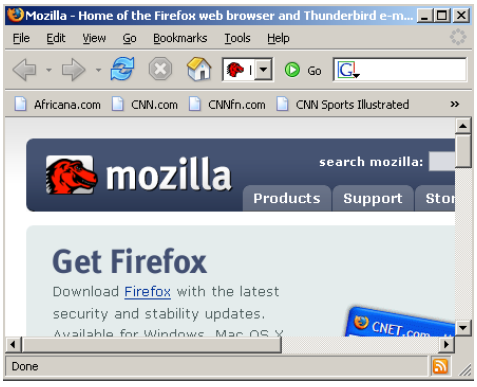
Ping (ICMP Echo and Echo Reply)

```
C:\> ping 192.168.10.33   (Windows)
```

or

```
# ping 192.168.10.33    (Linux/Unix)
```

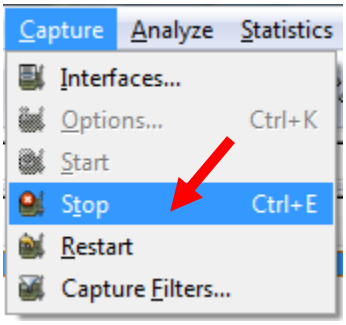
Web traffic



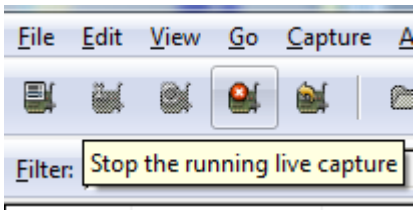
Using a web browser, go to a web site of your choosing,

Step 3: Stopping the Wireshark Capture

Click the Stop button or icon to end the capture.

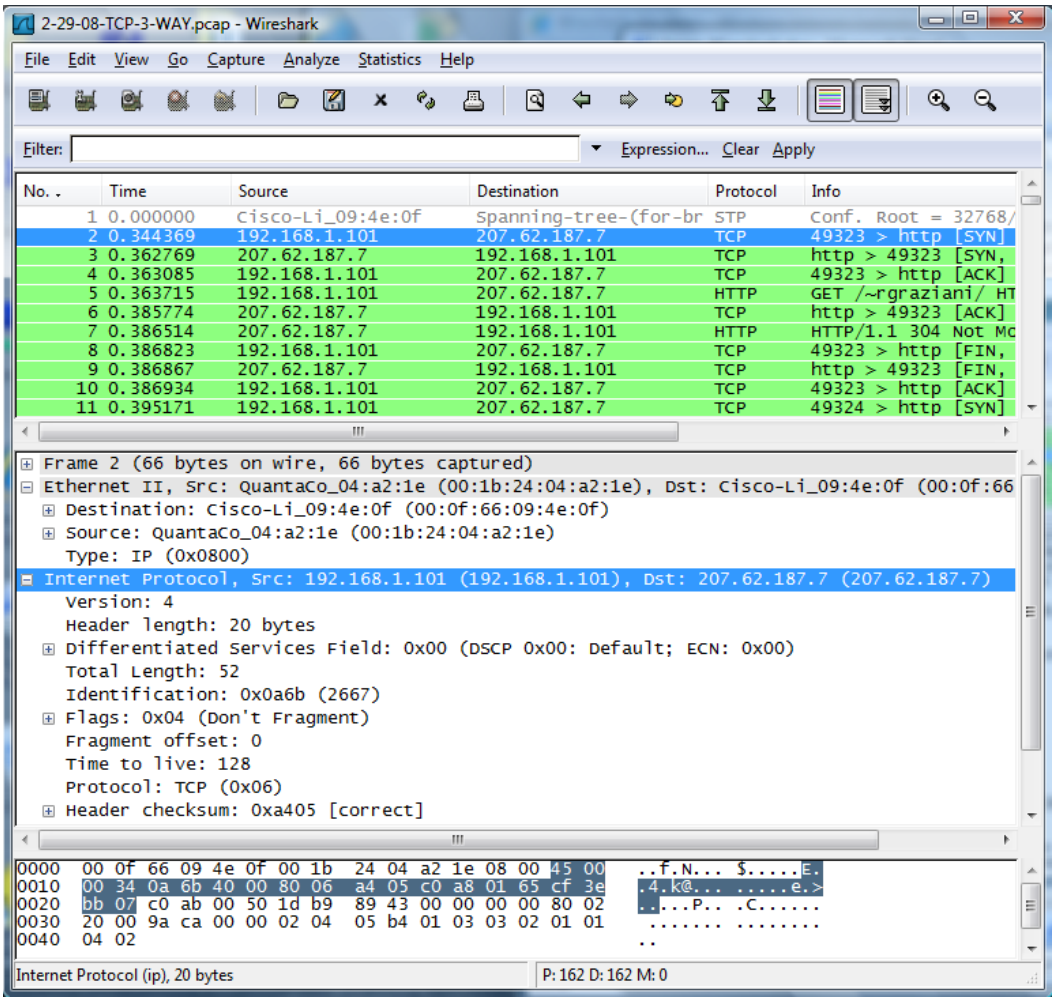


OR



Step 4: Looking at a Wireshark Frame

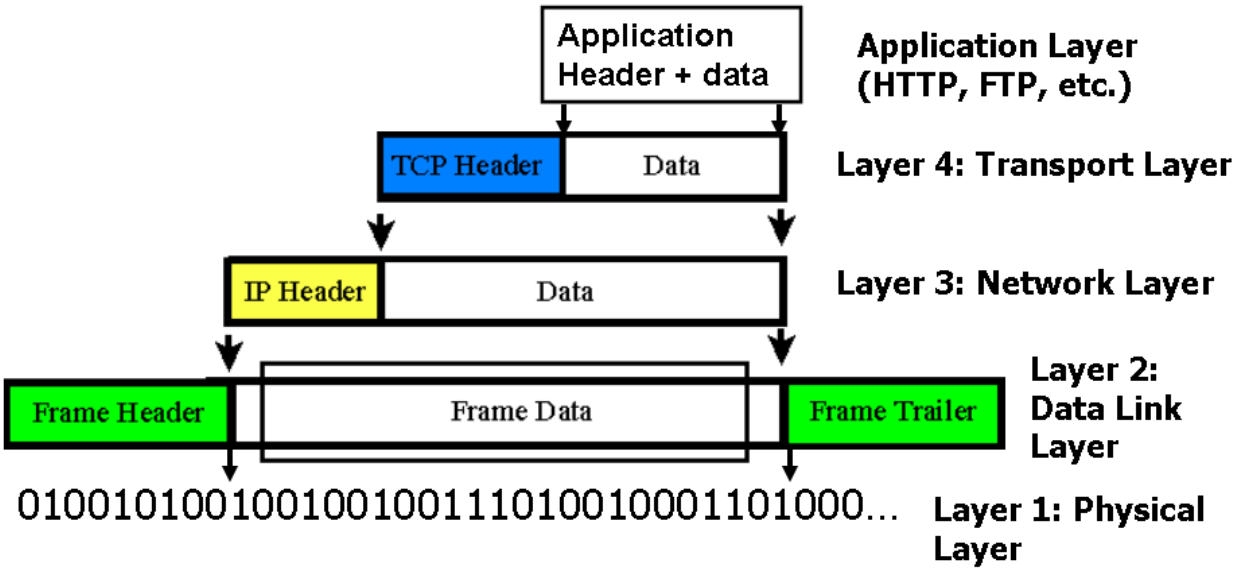
The amount of information captured can be overwhelming. You can use the “Capture Filter” option in the Options screen where you selected the interface. The amount of frames captured can also be limited by capturing frames in a lab with only a few computers instead of a production network.



Select a specific frame/packet. For example the packet with the encapsulated application protocol **HTTP**.

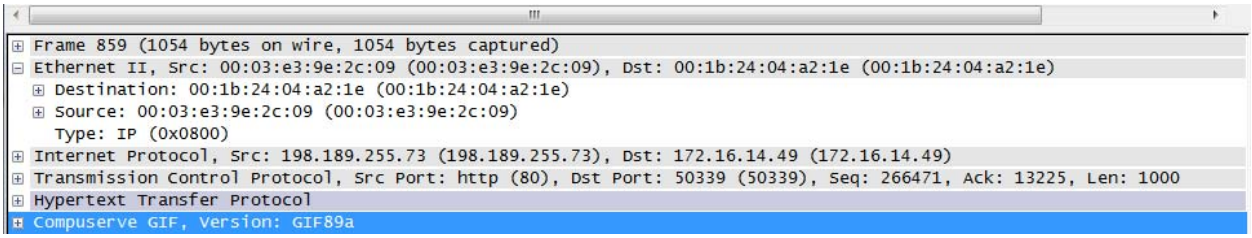
No.	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco-Li_09:4e:0f	Spanning-tree-(for-br	STP	Conf. Root = 32768/00:0f:6
2	0.344369	192.168.1.101	207.62.187.7	TCP	49323 > http [SYN] Seq=498
3	0.362769	207.62.187.7	192.168.1.101	TCP	http > 49323 [SYN, ACK] Se
4	0.363085	192.168.1.101	207.62.187.7	TCP	49323 > http [ACK] Seq=498
5	0.363715	192.168.1.101	207.62.187.7	HTTP	GET /~graziani/ HTTP/1.1
6	0.385774	207.62.187.7	192.168.1.101	TCP	http > 49323 [ACK] Seq=219
7	0.386514	207.62.187.7	192.168.1.101	HTTP	HTTP/1.1 304 Not Modified
8	0.386823	192.168.1.101	207.62.187.7	TCP	49323 > http [FIN, ACK] Se
9	0.386867	207.62.187.7	192.168.1.101	TCP	http > 49323 [FIN, ACK] Se
10	0.386934	192.168.1.101	207.62.187.7	TCP	49323 > http [ACK] Seq=498
11	0.395171	192.168.1.101	207.62.187.7	TCP	49324 > http [SYN] Seq=325

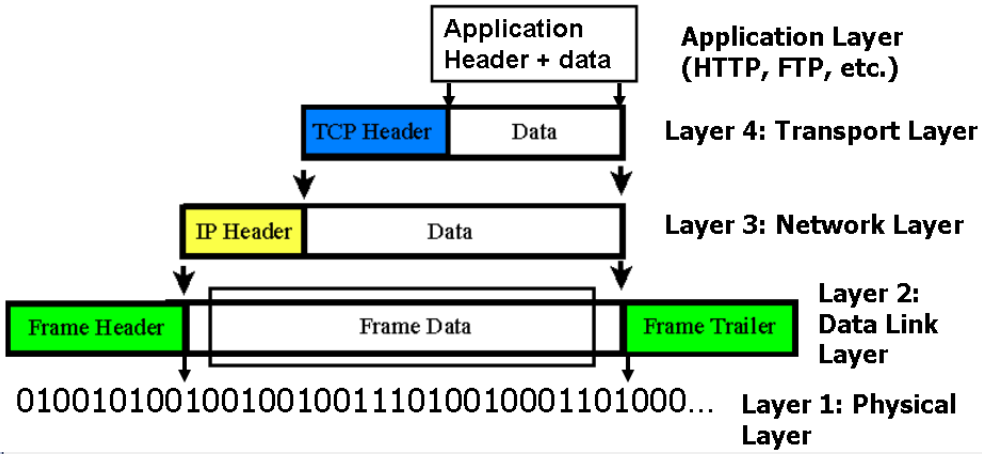
Inside the Wireshark frame is the data, which is usually another protocol with data. This encapsulation process most likely started with an application header (HTTP, etc.) with the original data. Below is an example.

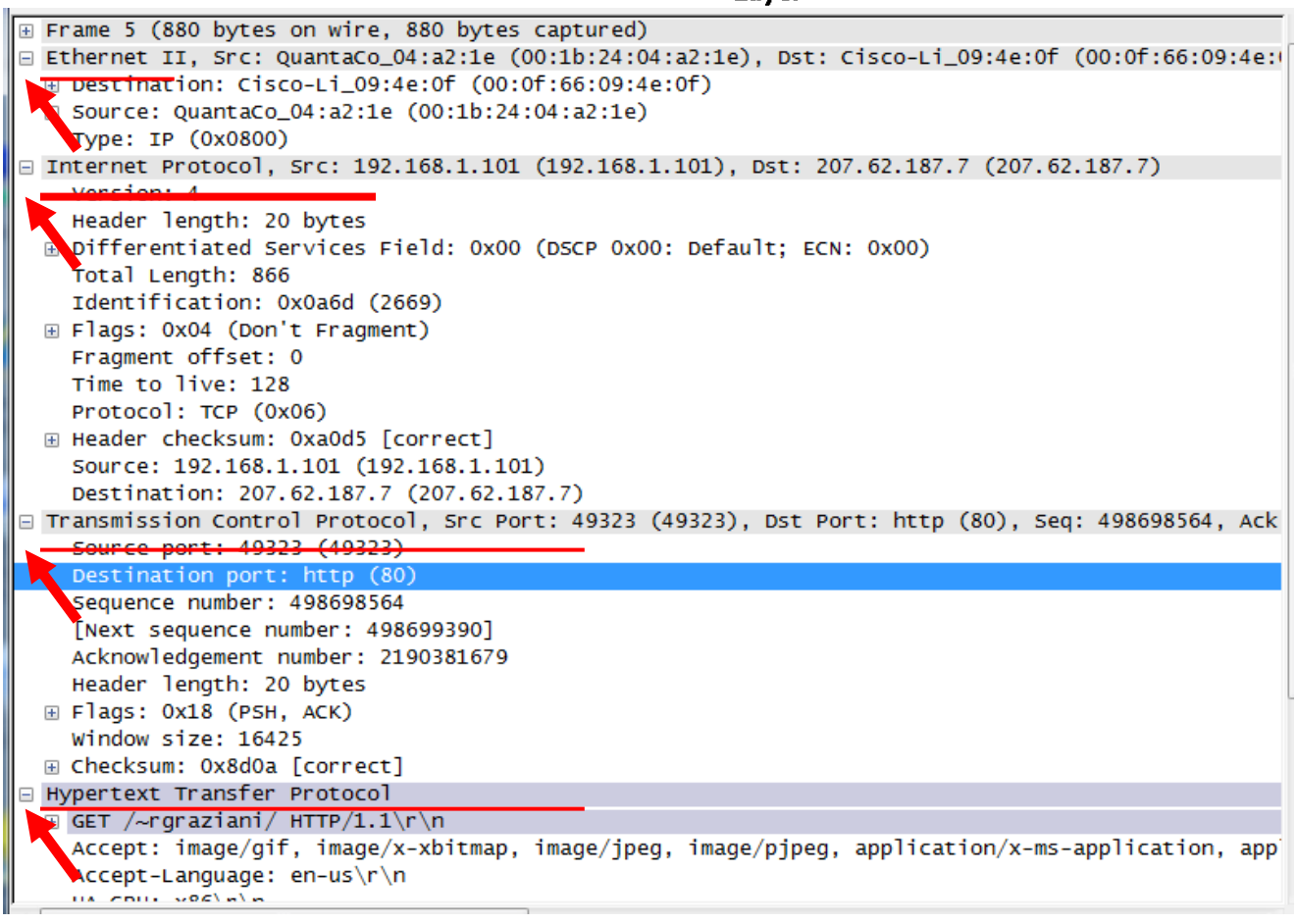


To view this information for each protocol, the Ethernet frame and the encapsulated upper layer protocols, click on the "+" sign next to that protocol. For example, to see the fields within the Ethernet frame click the **"+"** sign next to Ethernet II (the type of Ethernet frame). The **"+"** sign will turn into a **"-"** sign and display the particular information.

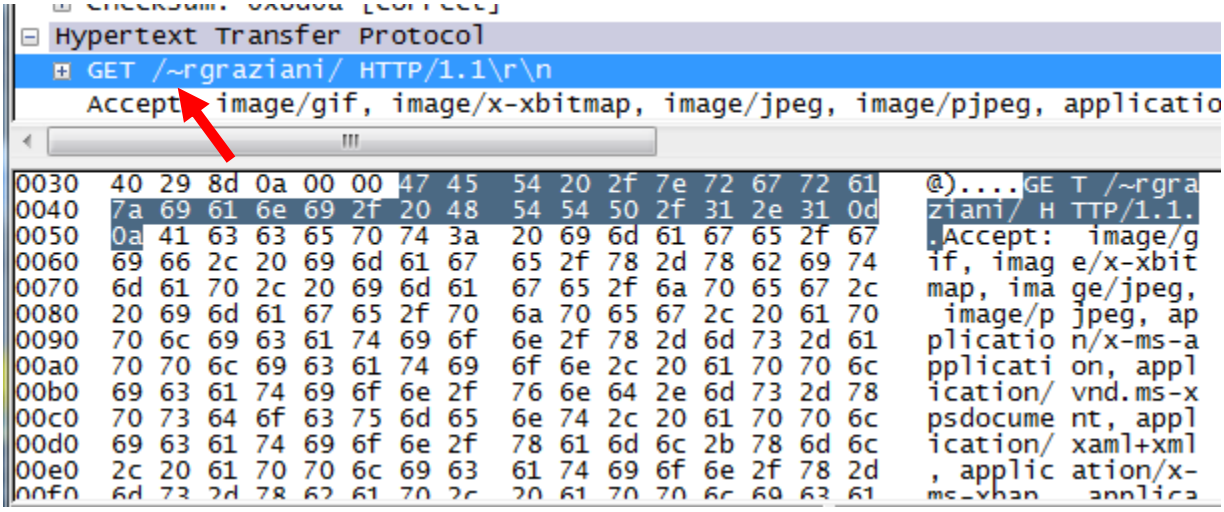
To see other layers, click the appropriate **"+"** sign next to the protocol.



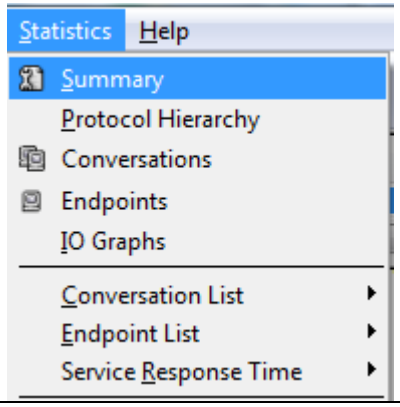


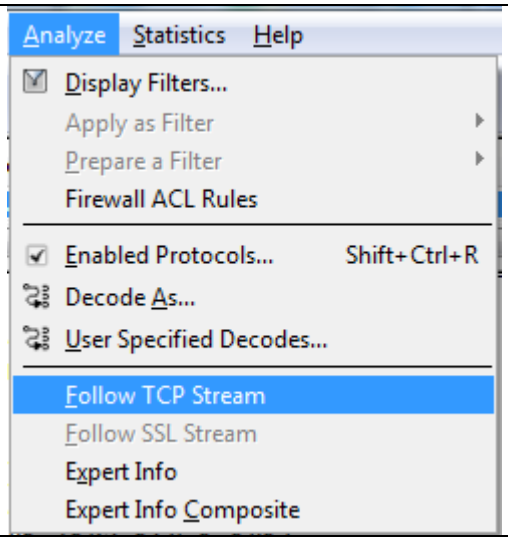


To view the data, look at the information in both Hexadecimal and ASCII.



Various statistics and graphs can be displayed from the Statistics menu.



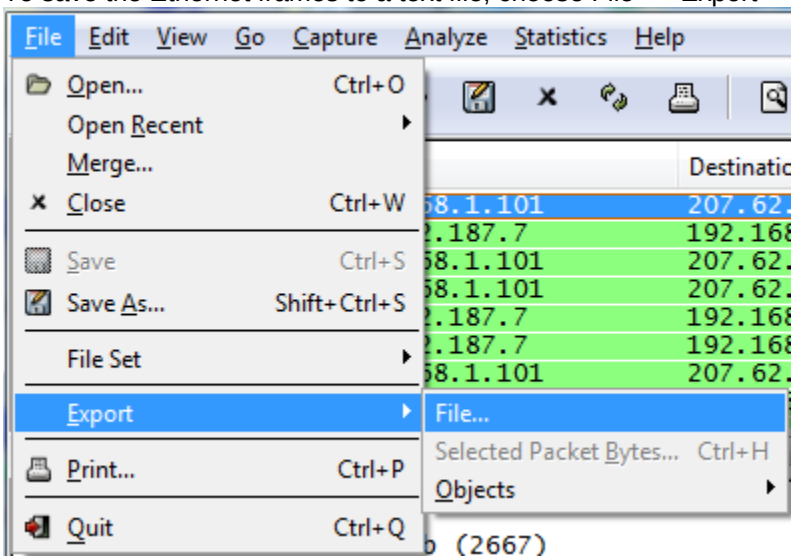
To view only the packets associated with a specific TCP session, such as downloading a web page, select Analyze >> Follow TCP Stream. Be sure you have one of the packets selected for that stream before applying this option.	

Step 5: Saving output to a text file

Outcome: Save an "expanded" Ethernet frame to a text file and print out the file. This Ethernet frame can include:

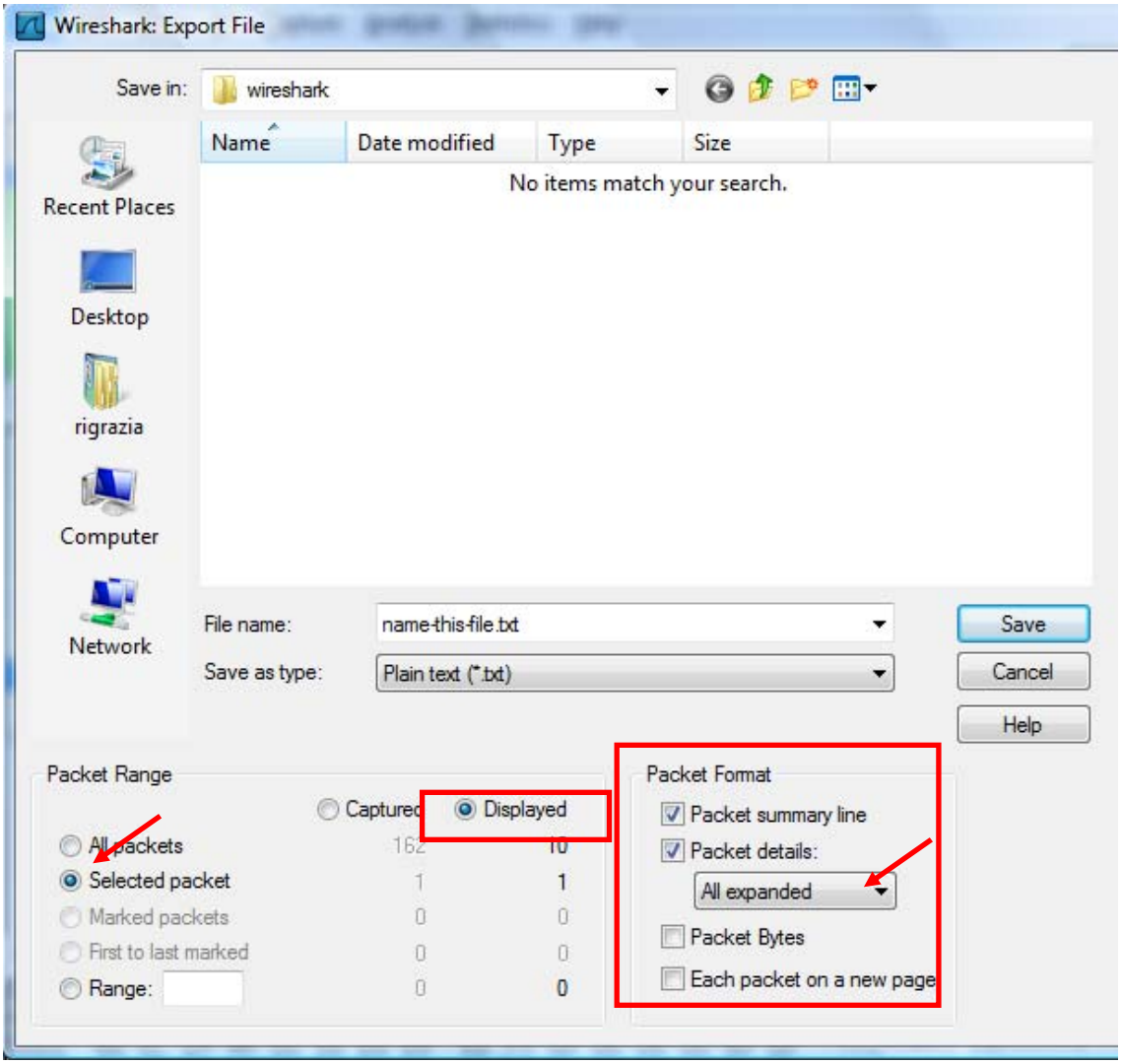
- Ethernet frame
- IP packet
- TCP/UDP header
- Application header and/or Data

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
```