INDIAN INSTITUTE OF INFORMATION TECHNOLOGY KALYANI
Autonomous institution under MHRD, Govt. Of India
&
Department of Information Technology & Electronics, Govt. of West Bengal
WEBEL IT Park Campus (Near Buddha Park), Kalyani -741235, West Bengal
Tel : 033 2582 2240, website : www.iiitkalyani.ac.in

Lab Assignment #02
Submit on or before 23/01/2018

Weekly contact	**: 0 – 0 – 3 (L – T – P)**
Course No.	**: CS 612**
Course Title	**: Networking**
Instructor-In-Charge	**: Dr. SK Hafizul Islam (hafi786@gmail.com)**

Aim

Explaining HTTP Traffic using Wireshark.

Objectives

i) To learn capturing live packets using Wireshark, which is a network protocol analysis tool.
ii) Analysing HTTP traffic using Wireshark.

Network Traffic Analyzer

Wireshark: It is an open-source and foremost network protocol analyzer. It lets you see what's happening on your network at a microscopic level. It is used for network analysis, troubleshooting and to assist communications protocol development and education. Wireshark does not manipulate packets on the network, but can only analyze those already present, with minimal overhead. Wireshark has a rich feature set which includes the following:
i) Deep inspection of hundreds of protocols, with more being added all the time
ii) Live capture and offline analysis
iii) Standard three-pane packet browser
iv) Multi-platform: Runs on Windows, Linux, OS X, Solaris and many others
v) Captured network data can be browsed via a GUI
vi) The most powerful display filters.
vii) Capture files compressed with gzip can be decompressed on the fly
viii) Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)
ix) Colouring rules can be applied to the packet list for quick, intuitive analysis

Important Links and References (Further reading)

i) http://is.gd/RazB76) ii) http://www.wireshark.org/about.html

Installing Wireshark

For Windows OS (Windows): i) Download the latest stable version of WireShark (Available at https://www.wireshark.org/download.html) ii) Choose all components for installation, including WinPcap iii) Proceed until completion.
(https://www.youtube.com/watch?v=SbpDTggwmPU&t=43s)

For Linux OS (Ubuntu)
i) sudo apt-get istall wireshark ii) press y iii) select yes and press enter iv) Open wireshark v) sudo apt-get --reinstall --no-install-recommends install wireshark-common →sudo apt-get --reinstall --no-install-recommends install wireshark-common→sudo chmod +x /usr/bin/dumpcap (to install dumpcap package)

Procedure (How to use Wireshark):

1. Start Wireshark by starting the executable from the installed directory.
2. Select proper interface for capturing packets (See Figure-1).

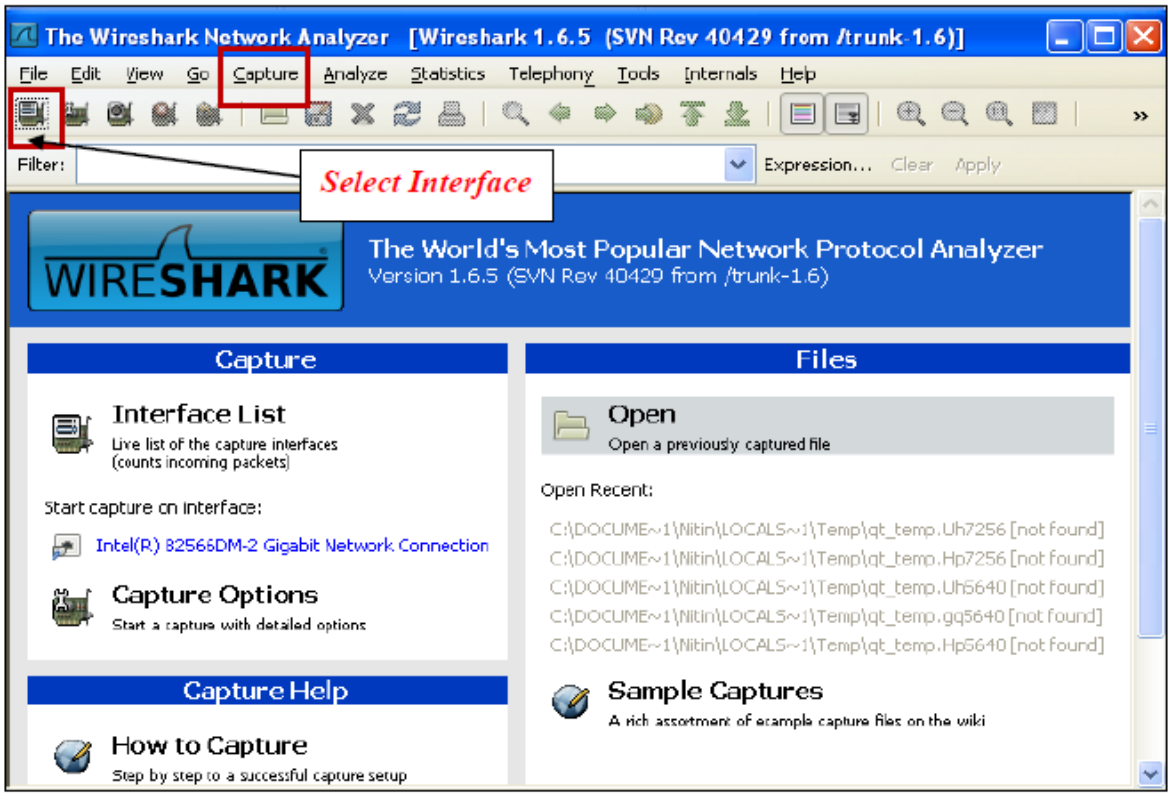

Figure 1

3. You will see a dynamic list of packets being captured by WireShark. In order to stop a running capture, press CTRL+E or from the menu, select Capture –> Stop (See Figure-2).

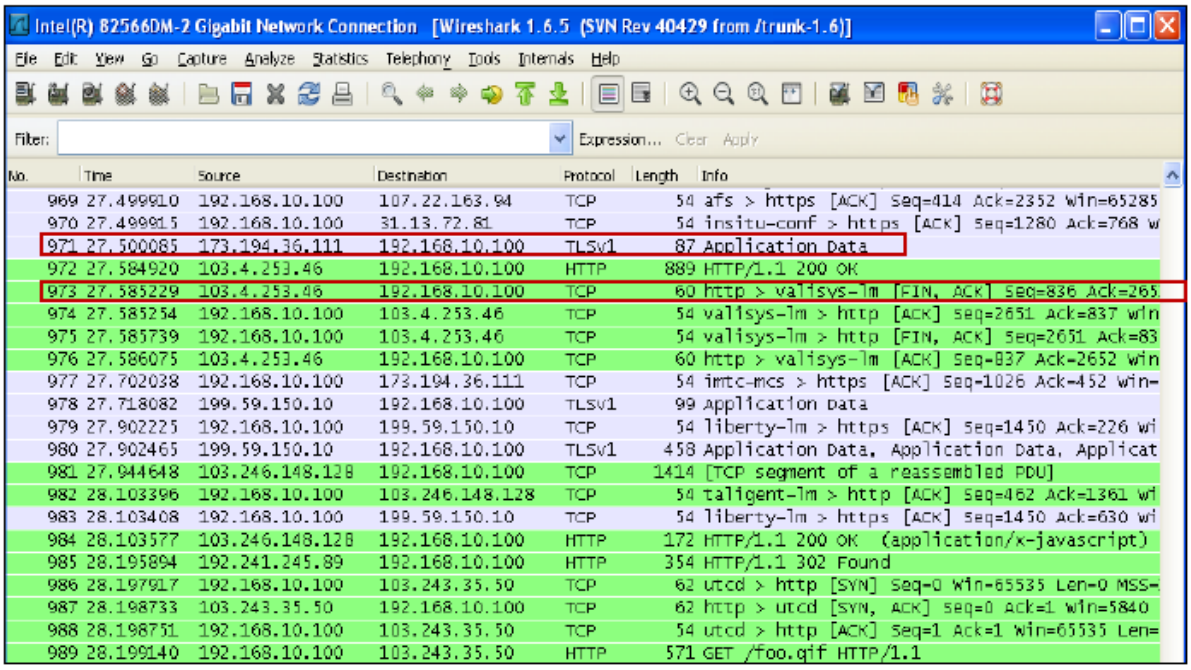

Figure 2

4. Various packets may be filtered. For instance, if you would only like to see HTTP packets enter HTTP in the Filter input-box and press Apply (See Figure-3 & 4).

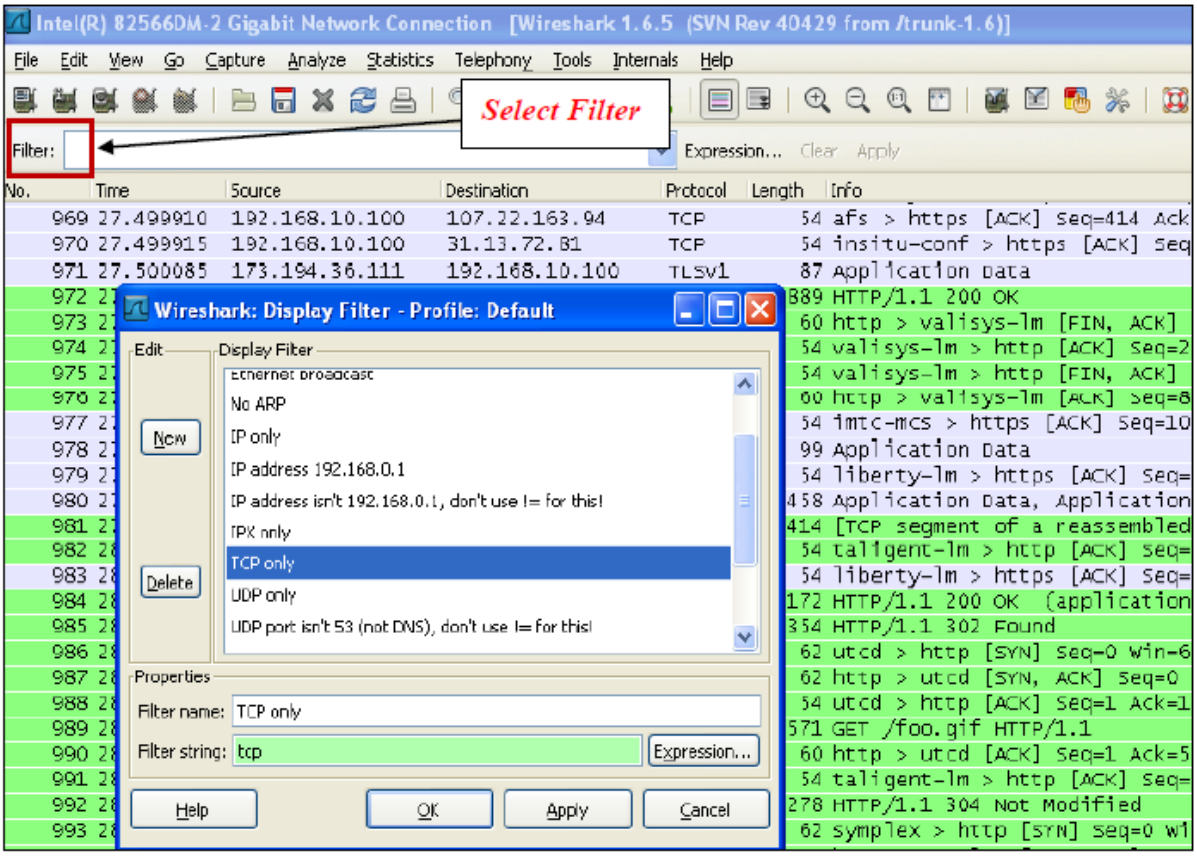

Figure 3

Ref: https://wiki.wireshark.org/Hyper_Text_Transfer_Protocol

Hyper Text Transfer Protocol (HTTP)

The Hyper Text Transport Protocol is a text-based request-response client-server protocol. A HTTP client (e.g. a web browser such as Mozilla/Chrome) performs a HTTP request to a HTTP server (e.g. the Apache HTTP server), which in return will issue a HTTP response. The HTTP protocol header is text-based, where headers are written in text lines.

HTTP/1.1 allows for client-server connections to be pipelined, whereby multiple requests can be sent (often in the same packet), without waiting for a response from the server. The only restriction is the server MUST return the responses in the same order as they were received. This enables greater efficiency, especially on revalidation.

An encrypted variant named HTTPS is also available. This is often used where privacy of data is necessary, e.g. when using online banking. The HTTPS protocol is in fact two protocols running on top of each other. The first protocol is a security protocol like SSL, TLS or PCT. The second protocol, which runs on top of this security protocol, is HTTP. The URLs starting with https:// really are only a shorthand notation for the end user. The web browser will read the URI scheme (https://), initiate the security protocol to the server, and once this secure connection is established, issue a HTTP request over it with the URI specified in the request.

History

The Hyper Text Transfer Protocol (HTTP) was initiated at the CERN in Geneve (Switzerland), where it emerged (together with the HTML presentation language) from the need to exchange scientific information on a computer network in a simple manner. The first public HTTP implementation only allowed for plain text information, and almost instantaneously became a replacement of the GOPHER service. One of the first text-based browsers was LYNX which still exists today; a graphical HTTP client appeared very quickly with the name NCSA Mosaic. Mosaic was a popular browser back in 1994. Soon the need for a more rich multimedia experience was born, and the markup language provided support for a growing multitude of media types.

Protocol Dependencies

1) MIME_multipart: HTTP uses MIME_multipart to encode its messages.
2) TCP: Typically, HTTP uses TCP as its transport protocol. The well known TCP port for HTTP traffic is 80. A HTTP proxy often uses a different port; typical values are 81, 3128, 8000 and 8080. However, HTTP can use other transport protocols as well.

Capturing HTTP Protocol Packets

Here are the steps for capturing and analysing FTP Protocol:

Step: 1 Start a Wireshark capture

- Close all unnecessary network traffic, such as the web browser, to limit the amount traffic during the Wireshark capture (not necessarily).
- Start the Wireshark capture.

Step: 2 Start a HTTP Session: By opening a website using your internet browser.

Step 3: Stop the Wireshark capture (after some time).

Step 4: View the Wireshark Main Window.

- Wireshark captured many packets during the HTTP session
- You can see the various protocols working underneath HTTP in Wirshark (i.e. TCP, IP etc).
- OR go to Statistics→Protocol Hierarchy to find the same.

Step 5: Select a HTTP Stream in Wireshark Main Window and analyse it. (See the section given below)

Analyzing HTTP Protocol

Consider following figure for capturing HTTP protocol packets using filter (circled red in the Figure- 4)

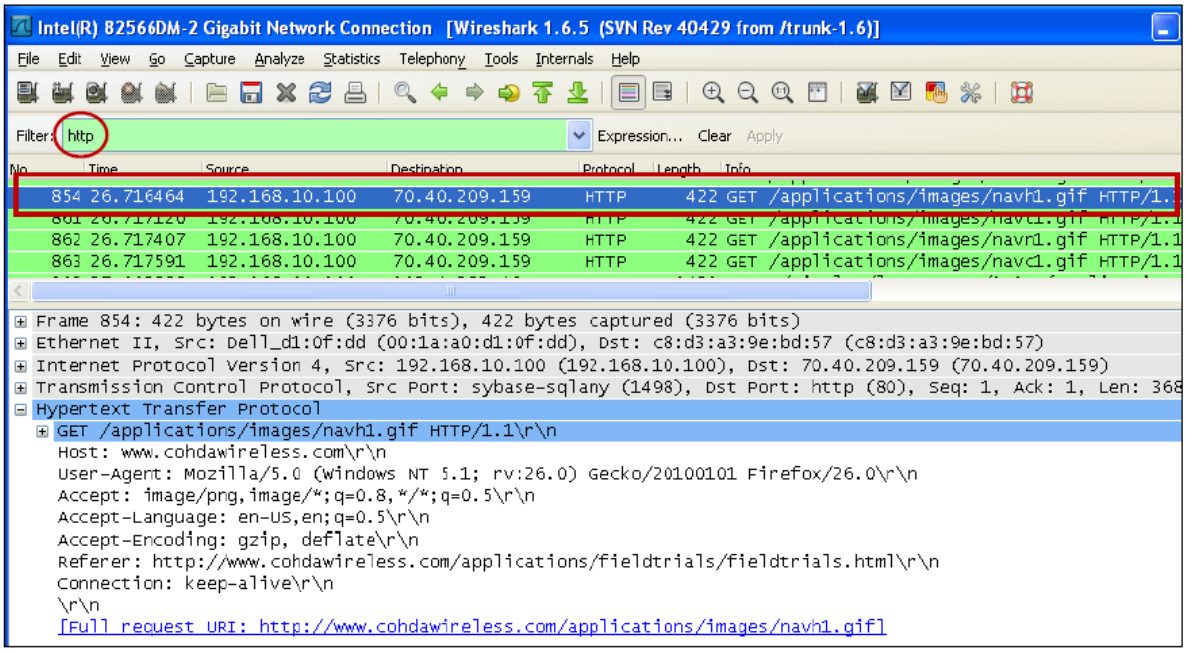

Figure 4

Example Traffic

Request by an end-users browser

This user wants to access the web site " http://wiki.wireshark.org/Hyper_Text_Transfer_Protocol", so they type in http:// http://wiki.wireshark.org/Hyper_Text_Transfer_Protocol into their browser and hit enter.

After the usual DNS resolution to find the IP address for:

http://wiki.wireshark.org/Hyper_Text_Transfer_Protocol.org, a connection is initiated via TCP to the web server (SYN; SYN & ACK; ACK). The very next thing to be sent to the web server by the browser/client is the following plain text request:

Figure 5

You can see this request in Wireshark as shown in below figure:

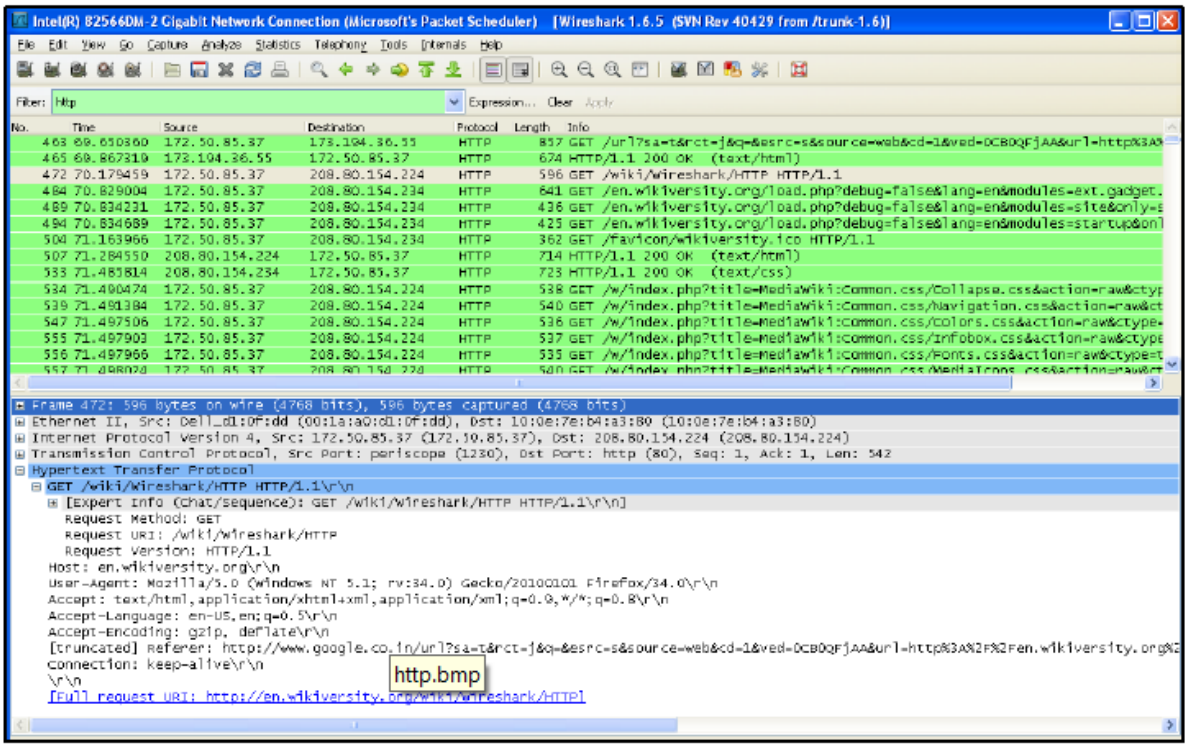

Figure 6

The server knows the browser/client is done with its traffic when it receives a blank line with a carriage return + line feed (\r\n).

Response from the server

The response is also in plain text:

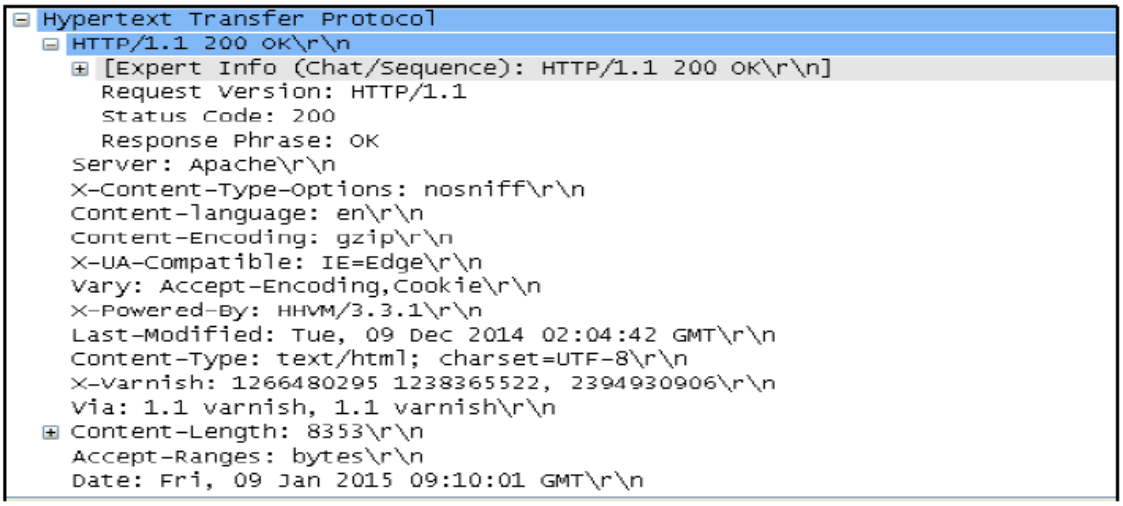

Figure 7

You can see this response in Wireshark as shown in below figure: Thus client and server start communication.

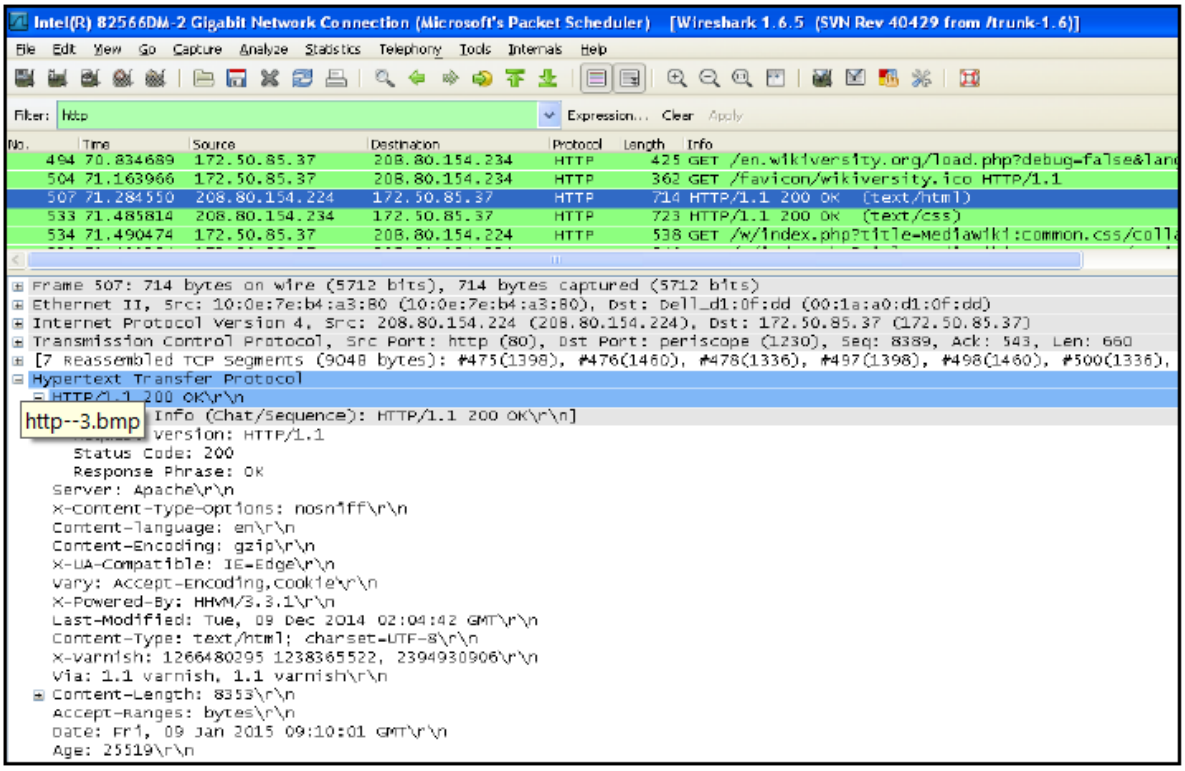

Figure 8

Assignment

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, as described in the Lab. Enter "http" (just the letters, not the quotation marks) 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 (we'll see why shortly), 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.

By looking at the information in the HTTP GET and response messages, answer the following questions. When answering the following questions, you should print out the GET and response messages and indicates where in the message you've found the information that answers the following questions. When you hand in your assignment, annotate the output so that it's clear where in the output you're getting the information for your answer (e.g., for our classes, we ask that students markup paper copies with a pen, or annotate electronic copies with text in a colored font).

- Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?
- Is HTTP used TCP or UDP for connection establishment? Identified the control packets (SYN, ACK, etc) with sequence numbers.
- What is the port number HTTP protocol used to setup the connection?
- What languages (if any) does your browser indicate that it can accept to the server?
- What is the IP address of your computer? Of the gaia.cs.umass.edu server?

- What is the status code returned from the server to your browser?
- When was the HTML file that you are retrieving last modified at the server?
- How many bytes of content are being returned to your browser?
- 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.
- What are the other protocols involved in this HTTP request and response, list them and write why they are used?