**ISE 316 Homework #2 – Due Feb 24 at 11:59 PM**

**Introduction to Networking**

**Spring 2023**

**Computer Science Department**

**Stony Brook University**

1. (10 pts) Consider an HTTP server and client. Suppose that the RTT delay between the client and server is 10 msecs; the time a server needs to transmit an object into its outgoing link is 0.5 msecs; and any other HTTP message not containing an object has a negligible (zero) transmission time. Suppose the client again makes 100 requests, one after the other, waiting for a reply to a request before sending the next request. Assume the client is using HTTP 1.1 and the IF-MODIFIED-SINCE header line. Assume 30% of the objects requested have NOT changed since the client downloaded them (before these 100 downloads are performed).

How much time elapses (in milliseconds) between the client transmitting the first request, and the completion of the last request?

1. (10 pts) Suppose a client likes to download a 400K bits long page that contains five other objects. Each object is 100,000 bits. Assume the RTT between the client browser and the server is 150 msec and the link rate is 100Mbps.
2. What is the time taken to get back the page and all objects if we assume non-­persistent HTTP with no parallel connections?
3. What is the time taken to get back the page and all objects if we assume non-­persistent HTTP with 2 parallel connections?
4. What is the time taken to get back the page and all objects if we assume persistent with no pipelining HTTP?
5. Repeat part c but with persistent pipelined HTTP.
6. **Wireshark:**

**Part 1**. The Basic HTTP GET/response interaction.

**NOTE:** The steps in italic below describe how to get a trace file. Since we cannot capture live packets on campus network (due to security concerns), to answer the homework questions, you can just open the packet trace **http-ethereal-trace-1** that was created when the steps were followed.

*The following are the steps to download a very simple HTML file - one that is very short, and contains no embedded objects:*

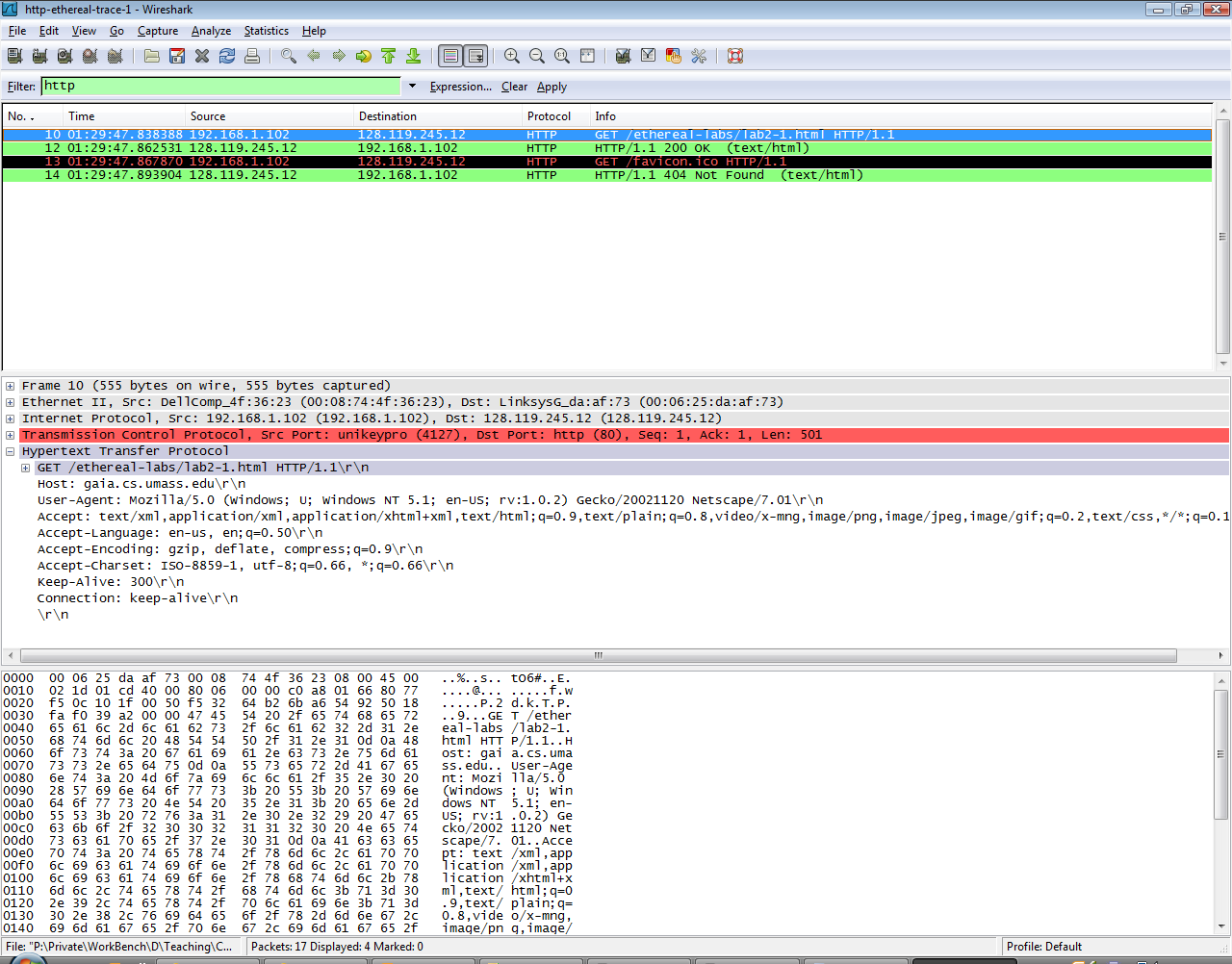
1. *Start up your web browser.*
2. *Start up the Wireshark packet sniffer, as you learned in the introductory homework. Enter “http” (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing 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, 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.*

1. *Stop Wireshark packet capture*

The example in the Figure below shows in the packet-listing window that four HTTP messages were captured: the GET message (from your browser to the gaia.cs.umass.edu web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP GET message, which is highlighted in the packet-listing window).



Recall that since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well.

We want to minimize the amount of non-HTTP data displayed (we’re interested in HTTP here, and will be investigating these other protocols in later homework), so make sure the boxes at the far left of the Frame, Ethernet, IP and TCP information have a plus sign (which means there is hidden information), and the HTTP line has a minus sign (which means that all information about the HTTP message is displayed).

(*Note:* You should ignore any HTTP GET and response for **favicon.ico**. It is your browser automatically asking the server if it (the server) has a small icon file that should be displayed next to the displayed URL in your browser. We’ll ignore references to this pesky file in this homework.).

By looking at the information in the HTTP GET and response messages, **answer the following questions.**

1. Is the client browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

Both client browser and server are running version 1.1.

2. What languages (if any) does client browser indicate to the server that it can accept?

Accept-Language: en-us, en;q=0.50\r\n

So, American English.

3. What is the status code returned from the server to the client browser?

HTTP/1.1 200 OK

So, code returned is 200.

4. For the HTML file returned to the client, when is it last modified at the server?

Last-Modified: Tue, 23 Sep 2003 05:29:00 GMT\r\n

5. How many bytes of content are returned to the client browser?

It’s 73 bytes as seen in two places:

* Content-Length: 73\r\n
* File Data: 73 bytes

**Part 2**. The HTTP CONDITIONAL GET/response interaction

Note: again, the following messages in italic describe how the trace was get. Since we cannot run Wireshark on a live network connection, you can open the packet trace **http-ethereal-trace-2** that was created when the steps below were followed.

*Most web browsers perform object caching and thus perform a conditional GET when retrieving an HTTP object. Before performing the steps below, make sure your browser’s cache is empty. (To do this under Netscape 7.0, select Edit->Preferences->Advanced->Cache and clear the memory and disk cache. For Firefox, select Tools->Clear Private Data, or for Internet Explorer, select Tools->Internet Options->Delete File; these actions will remove cached files from your*

*browser’s cache.*

* *Start up your web browser, and make sure your browser’s cache is cleared, as discussed above.*
* *Start up the Wireshark packet sniffer*
* *Enter the following URL into your browser*

*http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html*

*Your browser should display a very simple five-line HTML file.*

* *Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)*
* *Stop Wireshark packet capture and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.*

**Answer the following questions:**

6. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?

No, I don’t see that.

7. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

Yes, server did return the contents. We can see at the low part in line-base text data:

A screenshot of a computer screen

Description automatically generated with medium confidence

8. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header? Where does the client computer get this information?

Yes, I see it.

If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT\r\n

**Info retrieved from /ethereal-labs/lab2-2.html HTTP/1.1**

9. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file?

HTTP/1.1 304 Not Modified

Server did NOT return contents since no modification was made.

**Part 3**. HTML Documents with Embedded Objects

The steps below describe how the traces are obtained. Since we cannot run Wireshark on a live network connection, you can open the packet trace **http-ethereal-trace-3** that was created when the steps below were followed.

*Now that we’ve seen how Wireshark displays the captured packet traffic for large HTML files, we can look at what happens when your browser downloads a file with embedded objects, i.e., a file that includes other objects (in the example below, image files) that are stored on another server(s).*

* *Start up your web browser, and make sure your browser’s cache is cleared, as discussed above.*
* *Start up the Wireshark packet sniffer*
* *Enter the following URL into your browser*

*http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html*

*Your browser should display a short HTML file with two images. These two images are referenced in the base HTML file. That is, the images themselves are not contained in the HTML; instead the URLs for the images are contained in the downloaded HTML file. As discussed in the textbook, your browser will have to retrieve these logos from the indicated web sites. Our publisher’s logo is retrieved from the www.aw-bc.com web site. The image of our book’s cover is stored at the manic.cs.umass.edu server.*

* *Stop Wireshark packet capture and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed.*

**Answer the following question:**

10. How many HTTP GET request messages were sent by your browser? To which

Internet addresses were these GET requests sent? What was returned in the three responses?

There were 3 and each response in next page as screenshots.   
  
A screenshot of a computer

Description automatically generated  
  
  
  
  
  
FIRST  
  
Text

Description automatically generated

SECOND

Text

Description automatically generated

THIRD

Graphical user interface, text

Description automatically generated