CSEE 5110 Network Architecture I

Assignment 4 solution

Sri Harsha Chennavajjala (16210893)

SC9V9@mail.umkc.edu

1. **Suppose within your Web browser you click on a link to obtain a web page. The IP address for the associated URL is cached in your local host, so a DNS look-up is not necessary to obtain the IP address. Further suppose that the Web page associated with the link references ten very small objects on the same server. Let RTT0 denote the RTTs between the local host and one of the objects. Assuming zero transmission time of the object, how much time elapses from when the client clinks on the link until the client receives the full web page with**

a. Non-persistent HTTP?

b. Persistent HTTP?

Sol:

As the IP address is already cached in the local host and DNS look-up is not necessary, the client can immediately start connecting to the server using the cached IP address.

Non-persistent HTTP file transfer can be done either in serial or parallel.

a.1. Non-persistent HTTP without parallel connections:

In non-persistent HTTP, a new TCP connections has to be established for each object. Therefore the total time taken to receive an object would be:

One RTT_0 to establish TCP connection + One RTT_0 to receive the object = 2 RTT_0

It is given that the webpage has 10 small objects. Therefore, we can derive that

Time taken to transfer webpage = 2 RTT_0

Time taken to transfer 10 objects = 10 (2 RTT_0) = 20 RTT_0

Therefore

Total time taken to transfer the entire file = 2 RTT_0 + 20 RTT_0 = 22 RTT_0

a.2 Non-persistent HTTP with parallel connections:

In non-persistent HTTP with parallel connection, all objects can be sent simultaneously by establishing multiple TCP connections in parallel. Let's assume that both the client and the server can have more than 10 parallel connections at any time. Therefore:

Time taken to receive the base web page = 2 RTT_0

Time taken to receive 10 objects in parallel = 2 RTT_0

Total time taken to receive the entire web page = 2 RTT_0 + 2 RTT_0 = 4 RTT_0

b.1 Persistent HTTP without pipelining connection:

In persistent HTTP connection, a single TCP connection is setup for the entire web page transfer (along with the referred objects in the web page). There is no need to establish a new TCP connection for each object. Therefore:

Time taken to receive the base web page = 2 RTT_0

Time taken to receive one object = RTT_0

Time taken to receive web page with the referenced 10 objects = 2 RTT_0 + 10 RTT_0 = 12 RTT_0

b.2 Persistent HTTP with pipelining:

Persistent HTTP with pipelining connection enables the client and server to transfer the objects simultaneously. Let's suppose both the client and server supports more than 10 parallel connections at any time, then

Time taken to receive base web page = 2 RTT_0

Time taken to receive referred objects = RTT_0

Total time taken to receive the entire web page = 2 RTT_0 + RTT_0 = 3 RTT_0

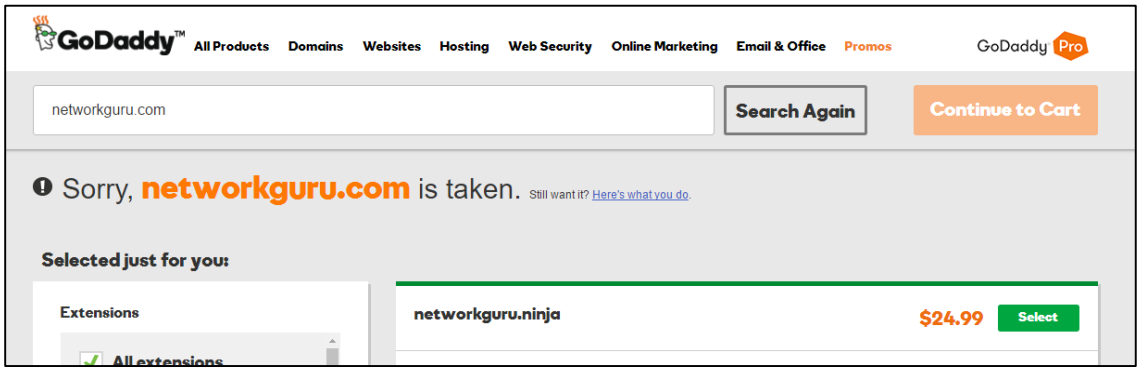
2. **Describe in detail i) what information should be added in which DNS servers for your own start-up company (say 'networkguru.com') that has a webserver and email service to its employees. ii) What are companies you can contact for domain name registration and how much are the fees?**

i) In order to register a domain name, the following steps needs to be performed:

1. First check whether the domain name networkguru.com is available or not. There are several registrar websites to check that. For e.g., GoDaddy.com, Domain.com. Choose one registrar who offers the domain for lesser price.
2. Let us suppose the local DNS server (authoritative name server) for networkguru.com is dns1.networkguru.com and its static IP address is 123.123.123.123. Let us suppose the authoritative server details for mail server as follows:
   Local DNS server for mail server: mail1.networkguru.com
   Static IP Address for mail DNS: 123.123.124.124
3. Now, we need to register these details with the selected DNS registrar.
4. Here the website networkguru.com is of type .com TLD, the registrar inserts the following Resource Records (RRs) into .com TLD server.

(networkguru.com, dns1.networkguru.com, NS) } Web Server
(dns1.networkguru.com, 123.123.123.123, A) }
(mail.networkguru.com, mail1.networkguru.com, MX) } Mail Server
(mail1.networkguru.com, 123.123.124.124, A) }

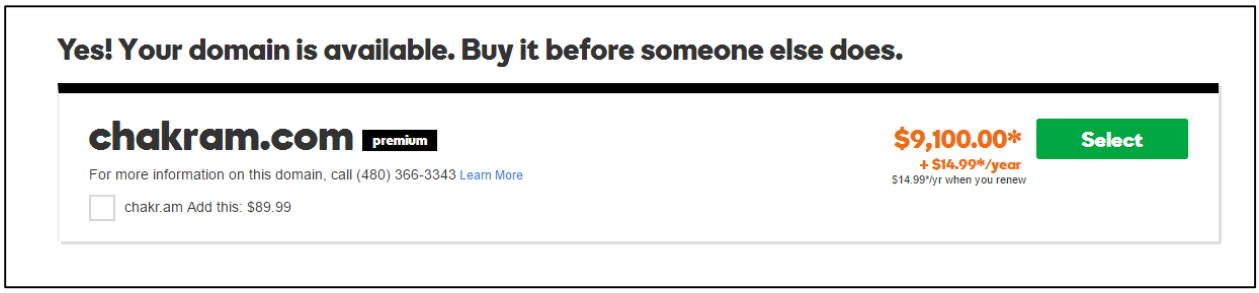
ii) There are several registrar companies available to register our domain name. Some of them are www.Domain.com, www.GoDaddy.com, www.networksolutions.com etc.
In our case, the domain name networkguru.com was already taken.



If the requested domain name is not available, then we will have two options to proceed with:

- Communicate with the domain owner to buy it
- Wait until the domain expires and participate in the domain auction

Generally domain names cost from $1 to $40 (for the first year). This amount will increase by $7 to $10 if we want to make our domain private. Some premium domain names will cost from $1000 to millions of dollars.



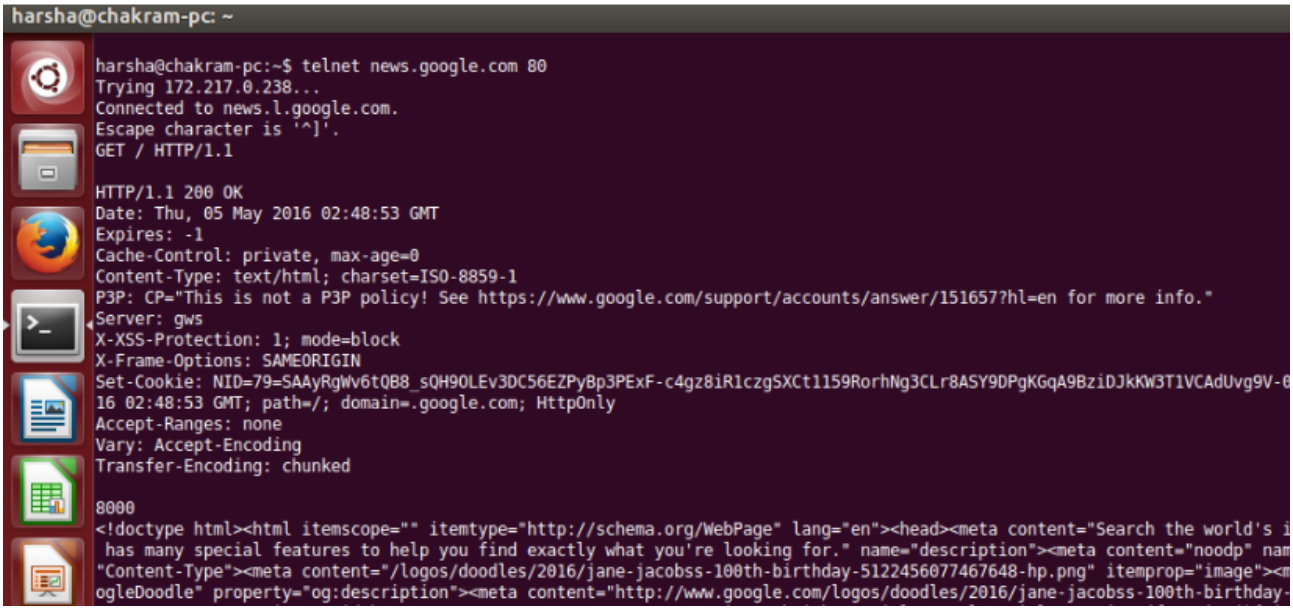
Laboratory Homework

Part 1: Telnet experiments

1. Try HTTP request (GET, HEAD, or POST) without using a web-browser. You can do this on command line using '> telnet webserver 80'. (for example, www.umkc.edu) Record the HTTP responses from the server – retrieve at least two different response status from the server.

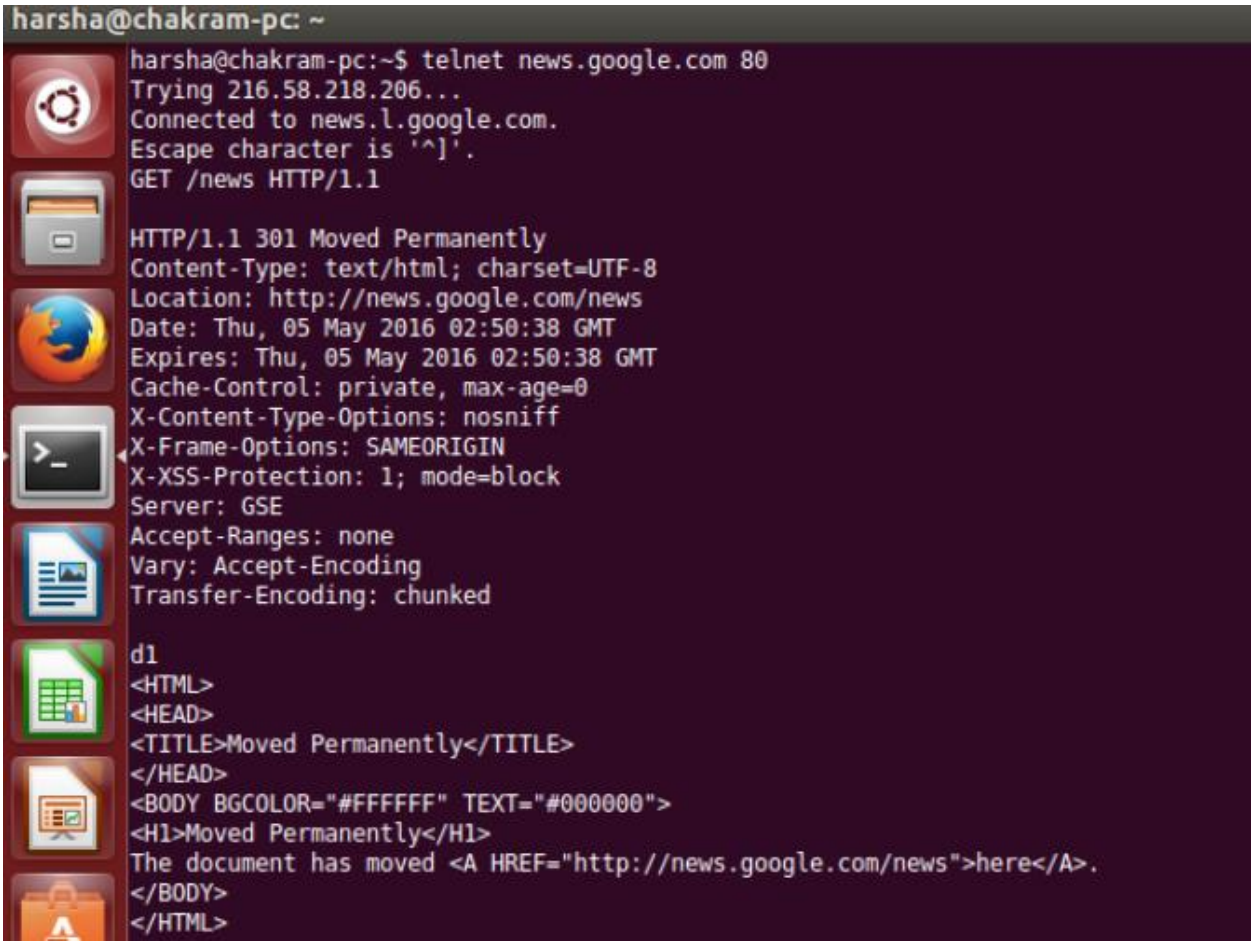
GET HTTP status: OK

Code: 200



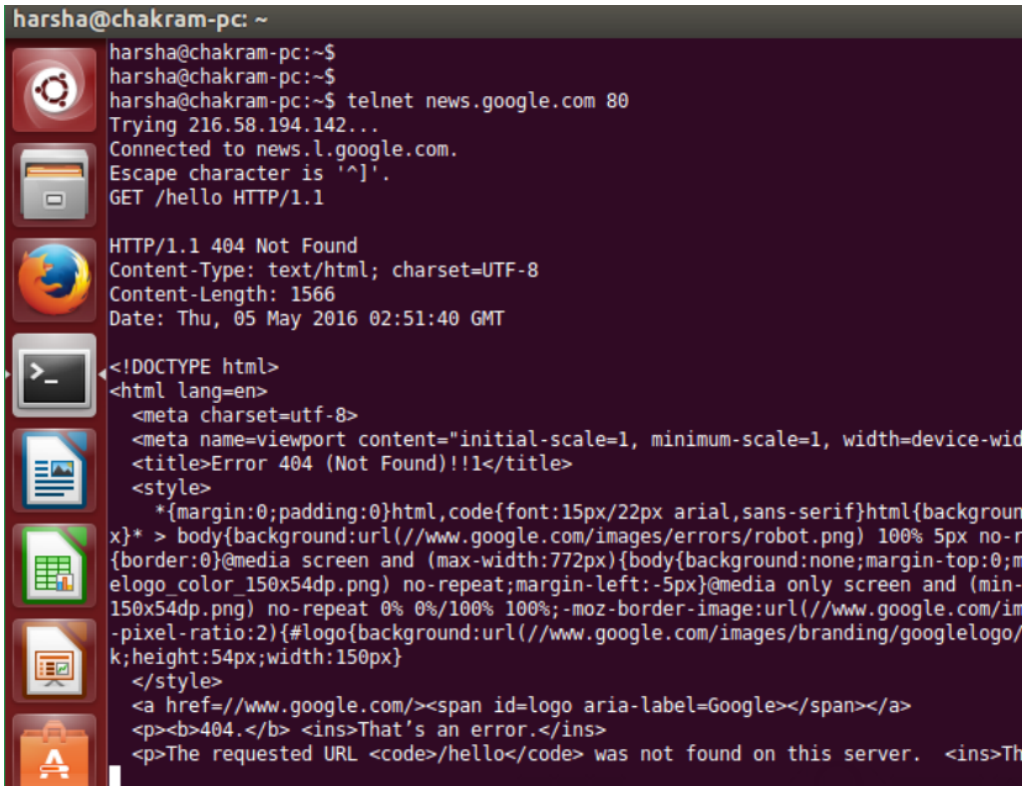
GET HTTP Status: Moved permanently

Code: 301



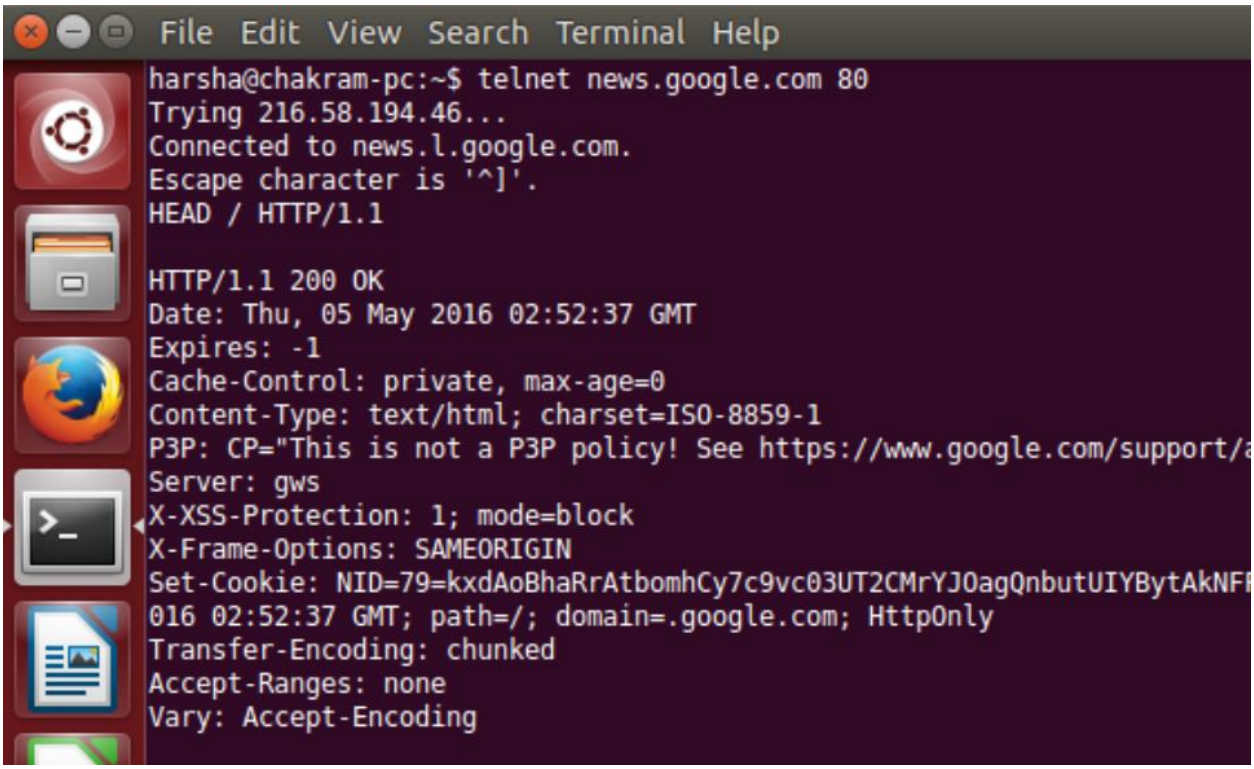
GET HTTP Status: Not found

Code: 404



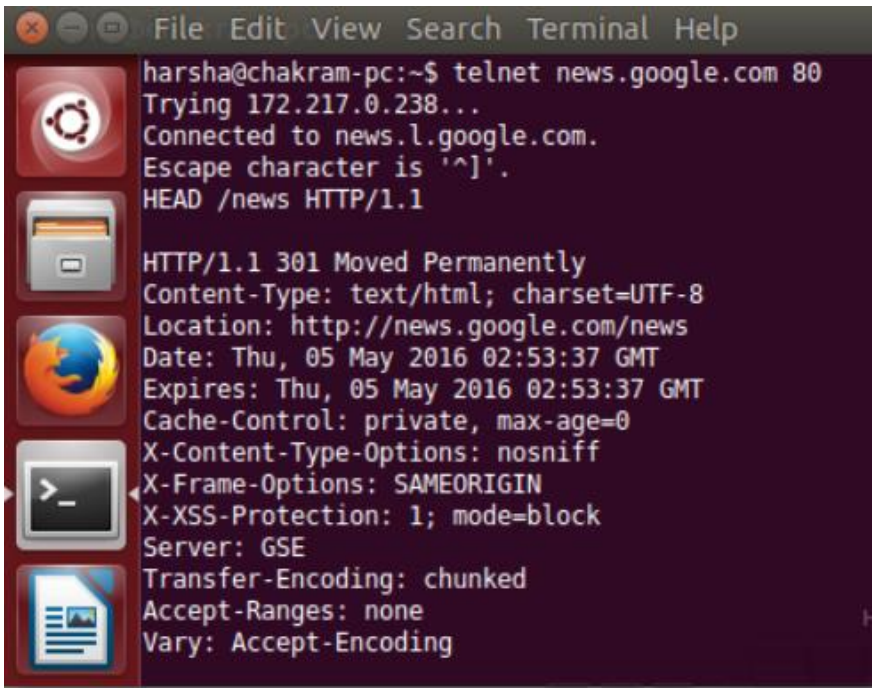
HEAD HTTP Status: OK

Code: 200



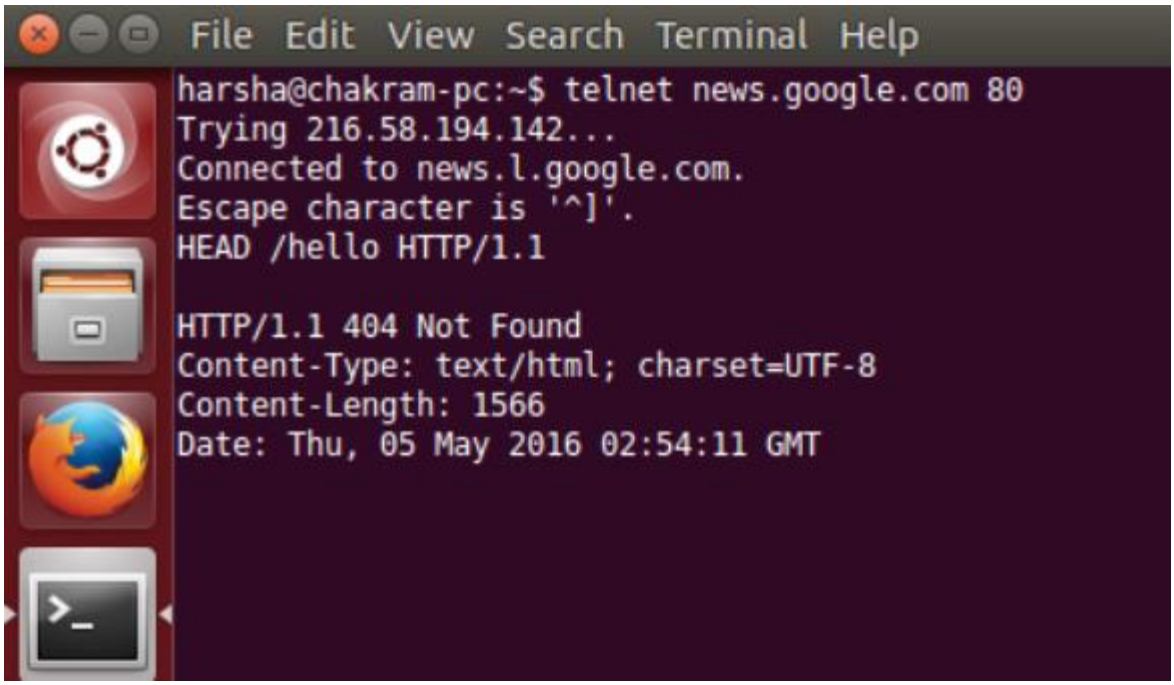
HEAD HTTP Status: Permanently moved

Code: 301



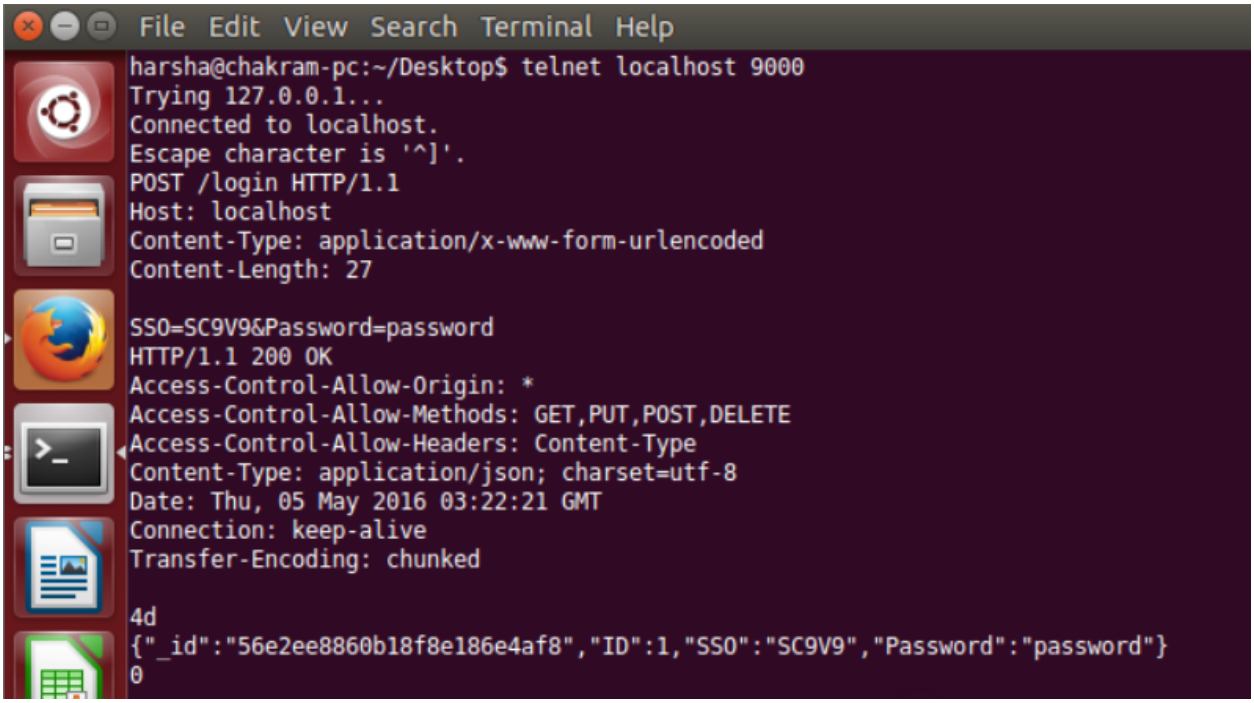
HEAD HTTP Status: Not found

Code: 404



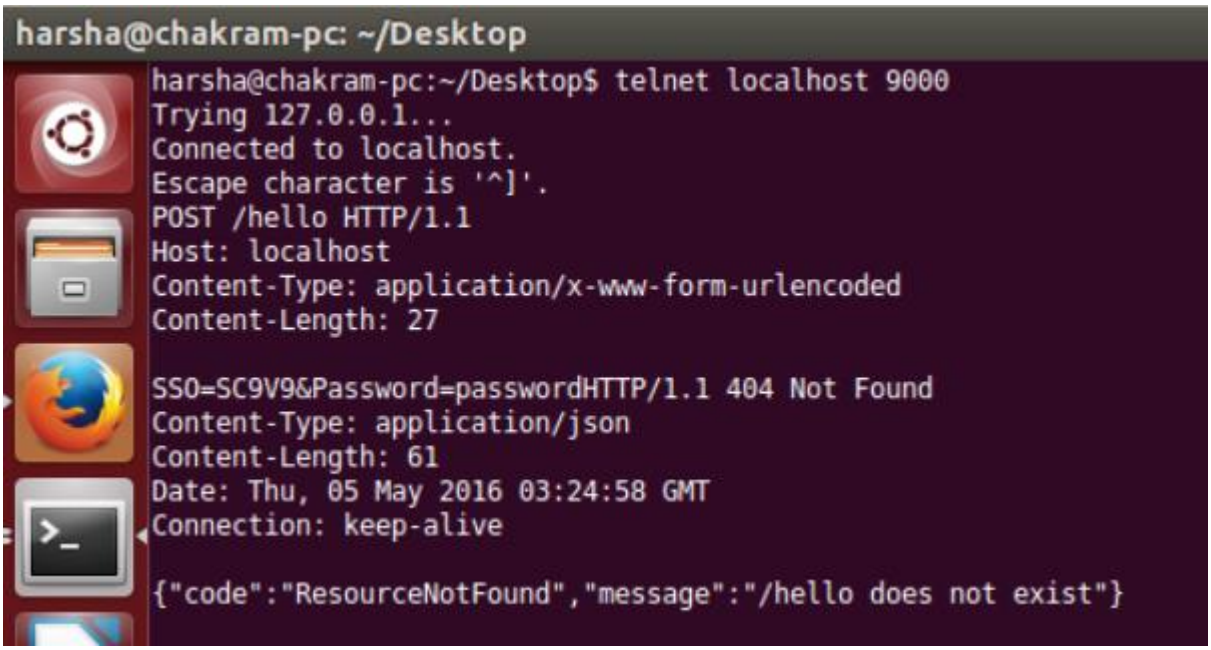
POST HTTP Status: OK

Code: 200



POST HTTP Status: Not found

Code: 404



Part 2: Wireshark experiments

In this part of the homework, you will use Wireshark to investigate HTTP protocol in operation. In this lab, you'll explore a couple of aspects of the HTTP protocol: the basic GET/response interaction, and retrieving large HTML files.

Download a packet trace file (http-wireshark-trace-1) of this homework from the Blackboard, Assignment Section. Or create a trace file on your own following the instruction at the footnote1 from you home (not at University!).

Then, open the trace file on the Wireshark, and answer to the questions below.

Part 2-1: The Basic HTTP GET/response interaction

Once you open the trace file, it will show in the packet-listing window that two 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 packetcontents 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 is later labs), so make sure the boxes at the far left of the

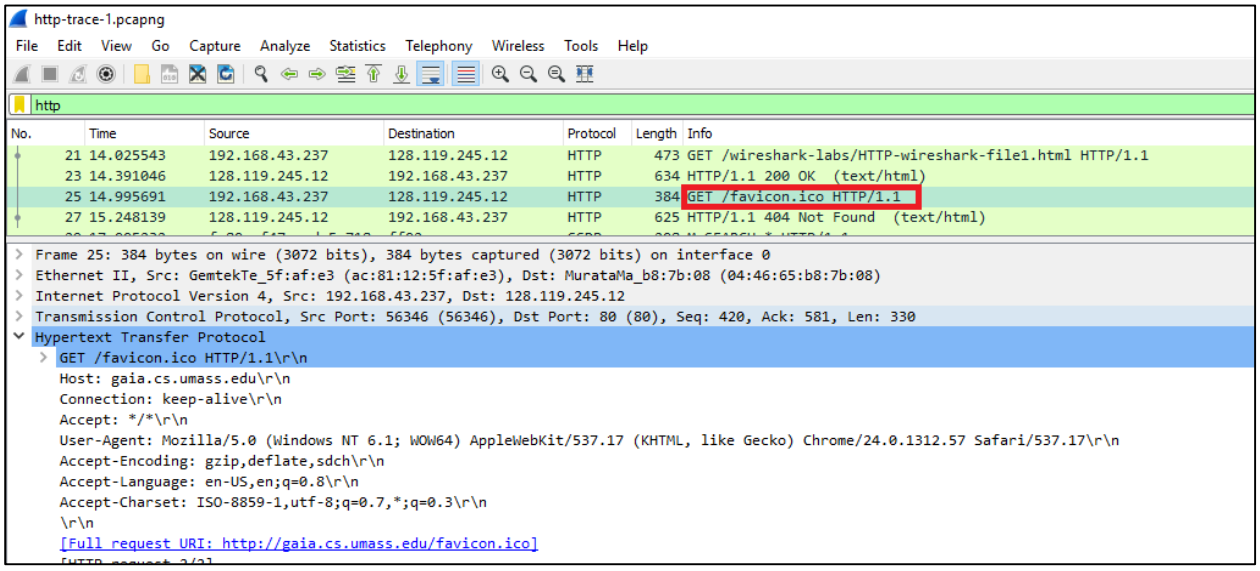
Frame, Ethernet, IP and TCP information have a plus sign (which means there is hidden, undisplayed information), and the HTTP line has a minus sign (which means that all information about the HTTPmessage is displayed).

(Note: You should ignore any HTTP GET and response for favicon.ico. If you see a reference to this file, 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 lab.).

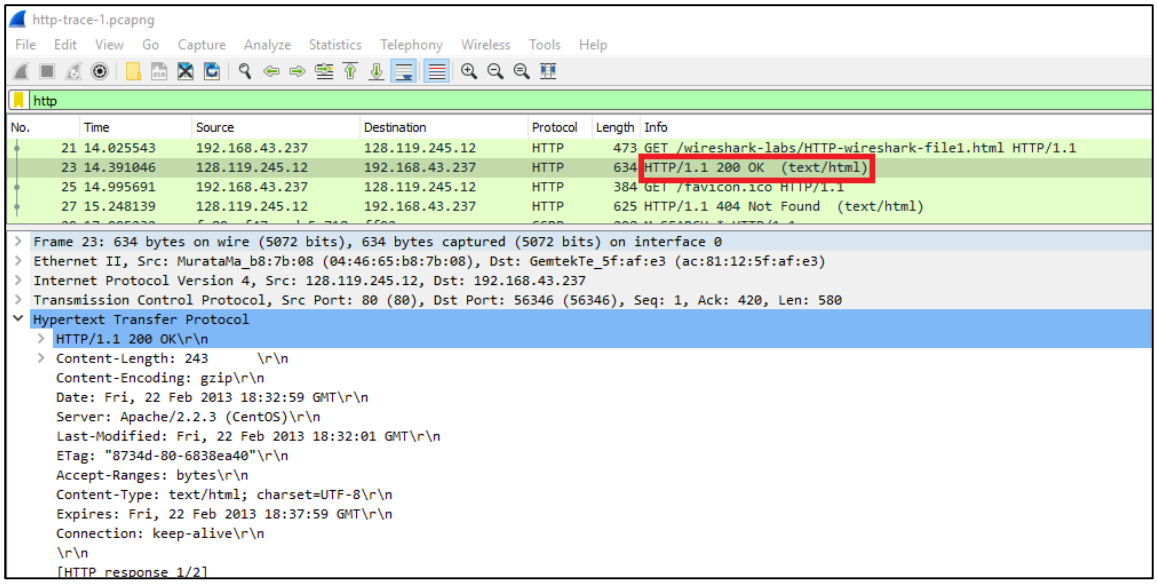
By looking at the information in the HTTP GET and response messages, answer the following questions. When answering the following questions, you should print out the GET and response messages (see the introductory Wireshark lab for an explanation of how to do this) and indicate where in the message you've found the information that answers the following questions.

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

From the below figure, it is found that my browser running HTTP version 1.1

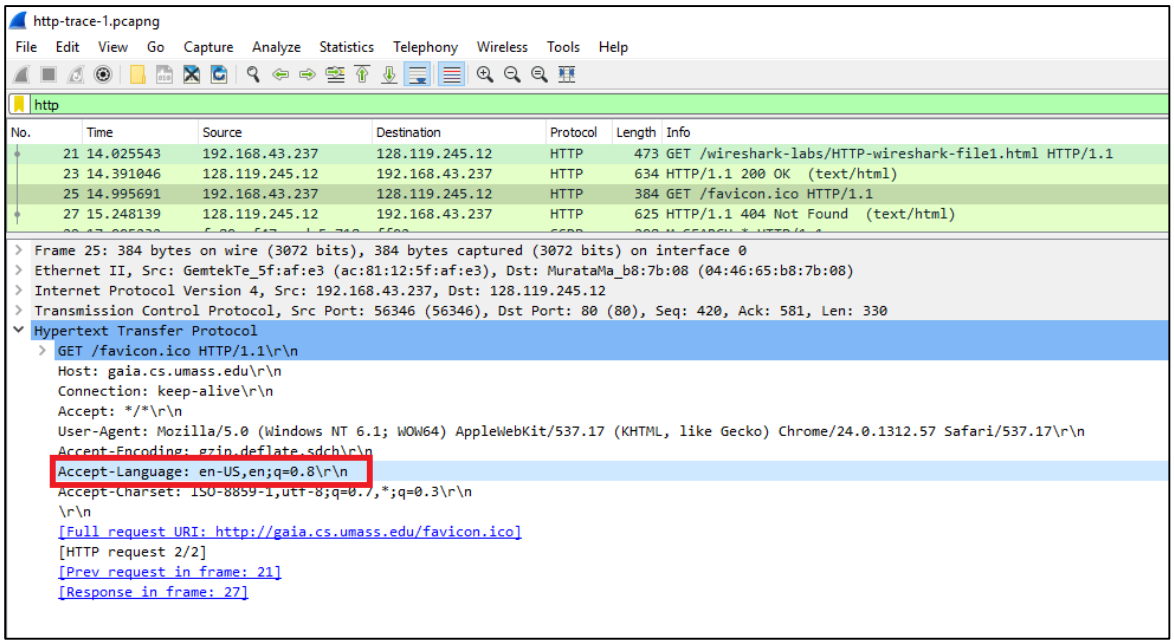


Server is running on HTTP version 1.1



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

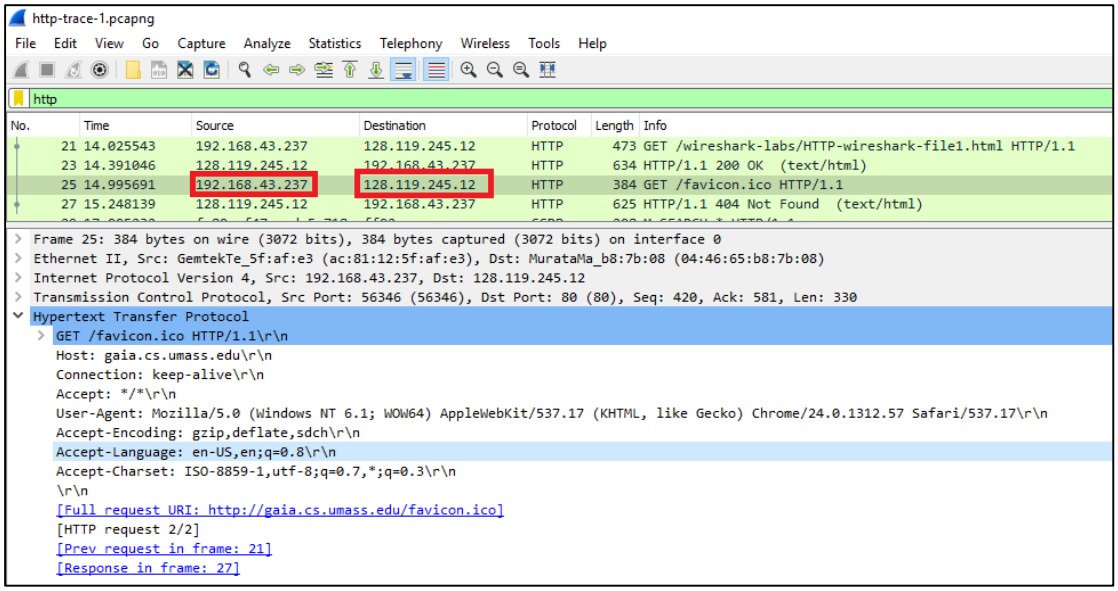
The language that the browser can accept is en-us.



3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?

IP address of source system is: 192.168.43.237

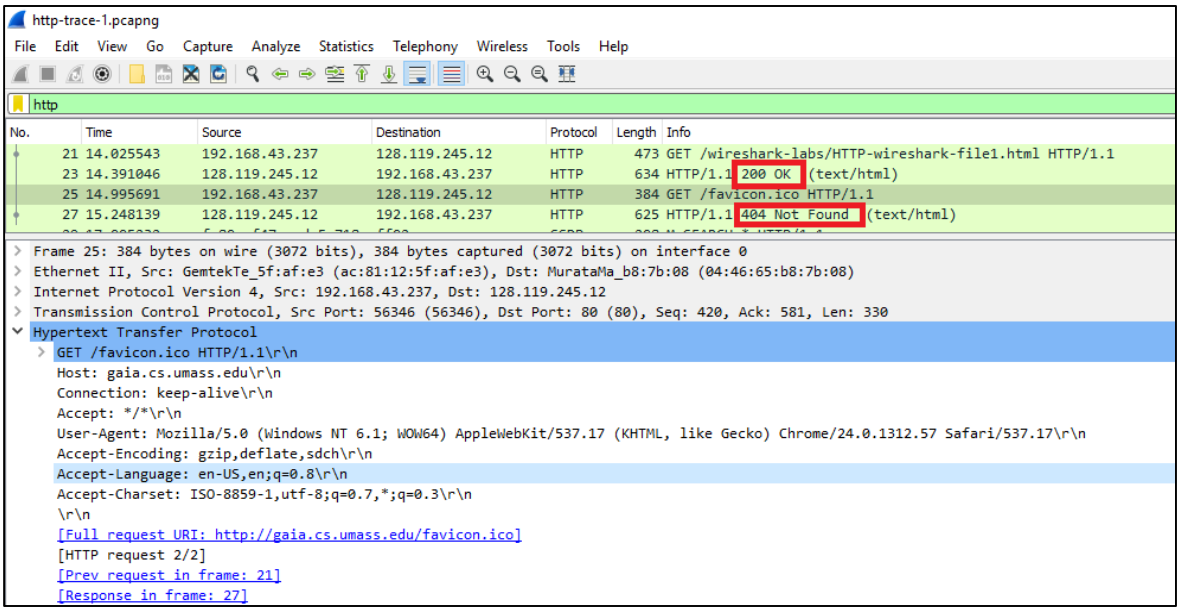
IP address of gaia.cs.umass.edu is: 128.119.245.12



4. What is the status code returned from the server to your browser? There are two return status codes sent by the server

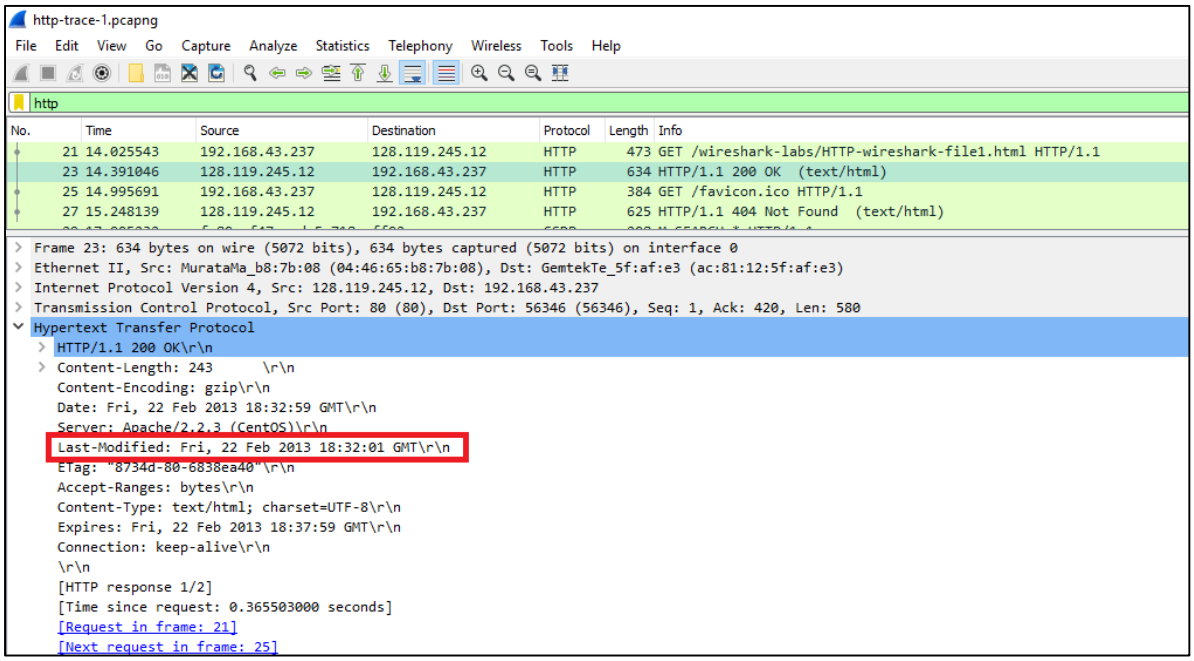
200 OK

404 Not Found



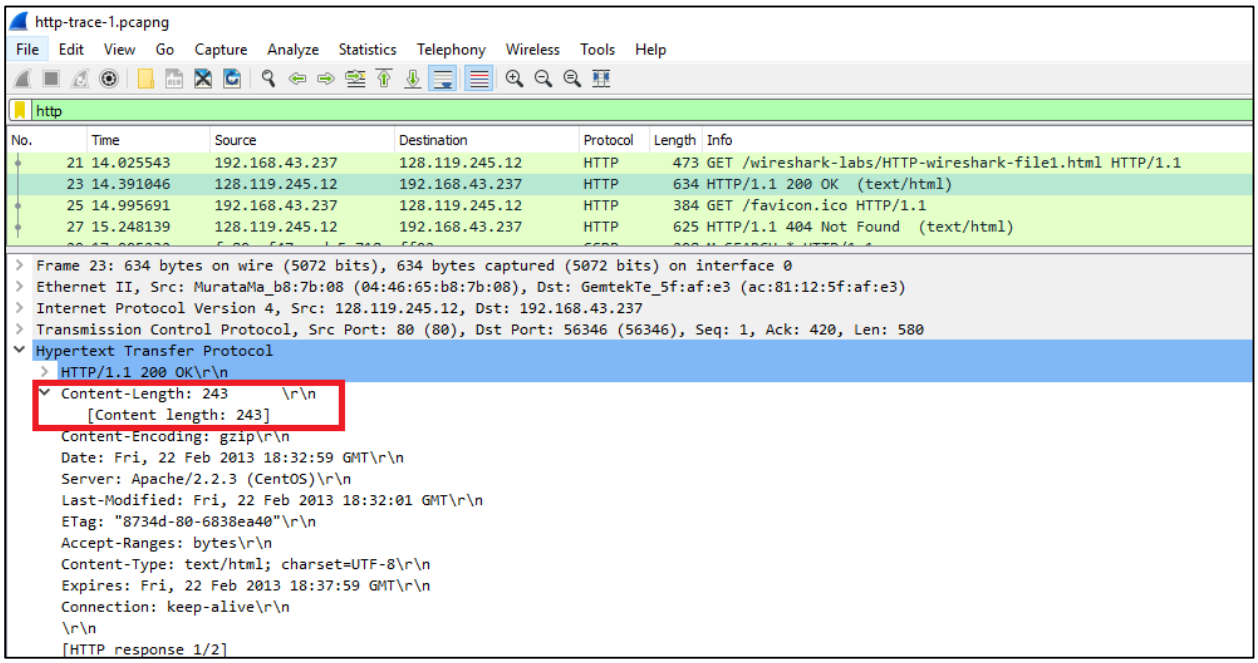
5. When the HTML file that you are retrieving was last modified at the server?

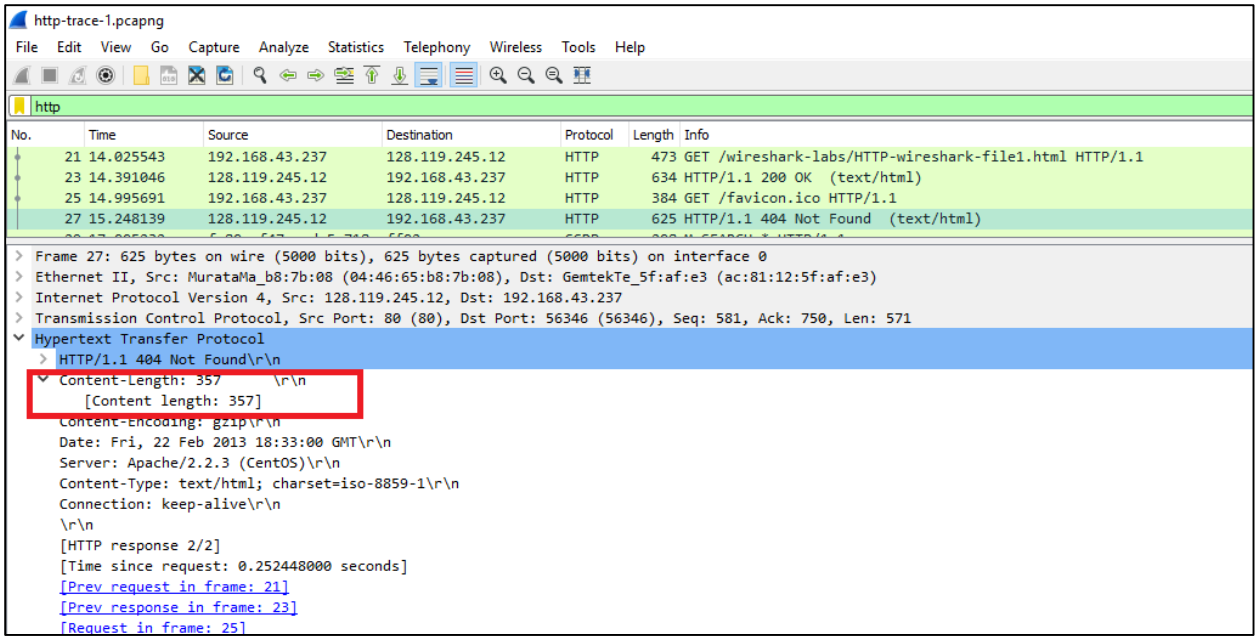
The last modified date and time for the HTML file is Fri, 22 Feb 2013 18:32:01 GMT



6. How many bytes of content are being returned to your browser?

There are two responses from server. Content lengths for these two responses are 243 bytes and 357 bytes




http-trace-1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
21 14.025543 192.168.43.237 128.119.245.12 HTTP 473 GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
23 14.391046 128.119.245.12 192.168.43.237 HTTP 634 HTTP/1.1 200 OK (text/html)
25 14.995691 192.168.43.237 128.119.245.12 HTTP 384 GET /favicon.ico HTTP/1.1
27 15.248139 128.119.245.12 192.168.43.237 HTTP 625 HTTP/1.1 404 Not Found (text/html)
Frame 27: 625 bytes on wire (5000 bits), 625 bytes captured (5000 bits) on interface 0
Ethernet II, Src: MurataMa_b8:7b:08 (04:46:65:b8:7b:08), Dst: GemtekTe_5f:af:e3 (ac:81:12:5f:af:e3)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.237
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 56346 (56346), Seq: 581, Ack: 750, Len: 571
Hypertext Transfer Protocol
HTTP/1.1 404 Not Found\r\n
Content-Length: 357 \r\n
[Content length: 357]
Content-Encoding: gzip\r\n
Date: Fri, 22 Feb 2013 18:33:00 GMT\r\n
Server: Apache/2.2.3 (CentOS)\r\n
Content-Type: text/html; charset=iso-8859-1\r\n
Connection: keep-alive\r\n
\r\n
[HTTP response 2/2]
[Time since request: 0.252448000 seconds]
[Prev request in frame: 21]
[Prev response in frame: 23]
[Request in frame: 25]