

# Lab 13

## HTTP

Prof. Kredo

Due: Start of lab Friday, May 8

Name:	
Name:	

## Introduction

In this lab you will accomplish several goals:

- Use telnet to interact with an HTTP server
- Explore how HTTP transfers content
- Research the new version of HTTP

Work in pairs for this lab using the equipment at your desk. Distribute the work evenly to make sure both group members know the material, as you will be required to know the material for evaluation.

## 1 Preliminary

Configure and connect your hosts so they can access the external network. There are several possible methods you can use from previous labs, but none require you to use a router. Test your setup by browsing a website from your hosts. Close your browser when you have verified your setup.

## 2 telnet and HTTP Servers [25 Points]

You can use `telnet` to interact with a web server by sending the commands normally sent by your browser. Open a `telnet` connection to the web server `www.ecst.csuchico.edu` on port 80 by entering the following command: `telnet www.ecst.csuchico.edu 80`.

From this point on, anything you type into `telnet` will be sent to the web server and any response from the server will appear in your terminal window.

Type in the HTTP messages your browser would normally send so that you can download the file `/~kkredo/file.html` using **HTTP 1.0**. You should type the *minimal* amount of information to form a valid HTTP request (one line).

1. Based on the notes from class (or other resources), what should you type in to retrieve the file? Adjust your answer as necessary until you are successful. Be sure to indicate any important non-printable characters.

--

When you are able to retrieve the file, answer the following questions.

2. What code and response phrase do you get from the server in the Response Line?

3. What does the server do after it sends you the file?

### 3 HyperText Transport Protocol [55 Points]

Lets now investigate some more about HTTP.

#### 3.1 HTTP/1.1 Changes

Use `telnet` to access the web server as you did in the earlier section, but change the HTTP version to 1.1. **The only change you should make to your previous command is the HTTP version.**

1. What happened? What code did the server send in its response?

2. How can you correct the error, based on our class discussion? (Hint: You need to include something for HTTP 1.1.)

Access the web server again to download the file with the corrected request.

3. What changes do you notice in the TCP connection between the successful HTTP/1.0 and successful HTTP/1.1 commands? How might this change be helpful? You may need to run each command multiple times to catch the difference.

### 3.2 HTTP Headers

Look at the headers provided by the server to answer the following questions.

1. When was the file you requested last changed?

2. How large is the file? How can you determine the file size using only the information in the HTTP packets?

3. How might an HTTP client determine the transfer is complete without this information? (If you don't know the HTTP standard, provide a logical answer.)

4. Lookup how web clients use the ETag header in relation to HTTP caches. Explain a simple example where a client may improve performance using a local cache and an ETag header.

### 3.3 Byte Serving

As mentioned in class, HTTP 1.1 allows you to download portions of a file. You can do this using the optional **Range** header. For example, including the header **Range: bytes=5000-5099** would retrieve 100 Bytes of the requested file starting at Byte 5000.

Request the file again, but only retrieve the text for the fourth amendment between the **p** tags. You will likely need multiple tries to get the correct range.

1. What was your final **Range** header for the request?

### 3.4 Cookies

Cookies have many uses in web development. Start a Wireshark capture and load a webpage for a large company.

1. How does an HTTP server give cookies to clients? How does a client tell an HTTP server about the cookies it has? Use terms from the HTTP messages in your answer.

## 4 SPDY and HTTP/2 [20 Points]

Perform some research on the next version of HTTP, HTTP/2. The standardization process is underway, so you might find more information if you also research the SPDY protocol, which forms the base for HTTP/2.

1. Explain one improvement in HTTP/2 when compared to HTTP/1.1.

**Submit your completed lab handout by the next lab.**