IIT Goa CS 212 (Computer Networks LAB)
Instructor: Dr. Neha Karanjkar

LAB 5: The Web, HTTP and DNS

Instructions:

- You can complete this lab exercise individually OR in teams of 2 persons (choose your teammate). If working in a team, all persons in the team should work on each problem together. Do not divide up the questions among team members.
- You need to show a demo of the working solution to the TAs for questions: **2 and 4**

Submission Instructions

- Submit a brief report (pdf file) containing sreenshots and descriptions for your solutions on Google classroom. Also attach code for your webserver. Naming conventions must be followed for all uploaded files.

PART 1: Creating Webpages using HTML and Javascript

1. Choose a name for your team. Create a website for your team focussed on any theme/topic of your choice, using **only raw HTML** and a text editor. (Some suggested themes: Internet memes, Networking-related jokes, Networking history...)

    - The website should consist of atleast 2 webpages, where one of the pages should be named “index.html”
    - Each webpage should contain a hyper-link to the other.
    - The webpages should contain:
        - alteast one link to some external website
        - atleast one image
        - some formatted text such as headings and bold text.

          Do not include inappropriate content in the webpages.

    Store the objects (.html files, images etc) in a single folder on the local machine and test your webpages by opening the html files in a browser.

2. The essential components/technologies that go into a modern webpage are:

    - HTML: to define the content of web pages
    - CSS: to specify the style/layout of web pages and
    - JavaScript: to program the behavior of web pages

    Tutorials and references for all three can be found here:
    https://www.w3schools.com/default.asp.

- Go through the initial tutorials for Javascript at https://www.w3schools.com/js/default.asp.
- Using Javascript, add a **button** to your webpage (created in q1) such that each time the user clicks this button, one of the images in your webpage changes to some other image.

PART 2: Creating a Web-server and observing HTTP traffic

3. Create a web-server for your website using Python. The webserver process should accept a TCP connection from a Client process on port 9090 and generate responses to GET requests for objects contained in your website's folder. (You can modify and use the template for a simple Python Webserver provided on Google Classroom.)

    - Test your webserver by entering the following url in a browser's address bar:
      "http://localhost:9090" on the same machine,
    - Check if your website can be accessed from another machine in the lab. Use the private IP address of the webserver (10.xx.xx.xx) instead of 'localhost'.

4. View the HTTP Request and Response traffic generated by the web browser using the browser's developer/Network tool. (For Google-Chrome, press F12 to view the Network Tools)

    - Check if you can see the "raw" HTTP requests and responses
    - Find out whether persistent or non-persistent connection was used by the browser for viewing your webpages
    - Open another webpage hosted remotely, such as https://www.iitgoa.ac.in/~nehak/example.html

      View the HTTP traffic for this webpage in the Network tool. Does the browser download the webpages components (images etc) in a **sequential or pipelined or parallel** manner? **[find out and show a demo to the TAs]**

- Observe the HTTP requests in Wireshark. Look out for the TCP handshaking (SYN/ SYN-ACK/ACK sequence). Is there one persistent TCP connection per webpage?

PART 3: DNS

5. You can use the dig command to send DNS queries to any specified DNS server.

   General format of the command:
   `$ dig @<DNS server> <name to be looked up> <type>`
   Example: $ `dig @a.root-servers.net com NS`

   - Go through this short tutorial for the dig command

     https://www.hostinger.in/tutorials/how-to-use-the-dig-command-in-linux/

   - Find out the appropriate command options (for dig) to send a DNS query to the root server "**`a.root-servers.net`**", requesting for the address of all name-servers (NS) for the "**`com`**" top-level domains.
   - Try running dig with the "+trace" option to see all the steps performed in iteratively querying the hierarchy of DNS servers. Try to make sense of the output.

     Example: **`dig +trace  @8.8.8.8 stackoverflow.com`**
   - Open Wireshark and observe the DNS traffic on your computer using the filter "dns"
   - Find address of your local DNS server
   - Find out addresses of any 3 root-level DNS servers

----------End----------