**2019 COMP2410/COMP6340: Networked Information Systems**

**ASSIGNMENT 1**

**Seeing the layered network model in action (max 40 points)**

Associate Professor Hanna Suominen (Convener) and the COMP2410/COMP6340 teaching team Research School of Computer Science, *College of Engineering and Computer* *Science* (CECS), The Australian National University, Canberra, ACT, Australia

|  |  |  |
| --- | --- | --- |
|  | **Important Dates (ACT time)** | **Important Notes** |
| **Assignment 1 released on**[**Wattle**](https://wattlecourses.anu.edu.au/course/view.php?id=27233) | Fri 22 Feb 2019 at 4:30 PM |  |
| **Assignment 1 due on Wattle** | Sun 7 Apr 2019 at 11:55 PM | The assignment submission includes a report (PDF) and perhaps code. If code is included, then submit one ZIP file with the report (PDF) and all the code (e.g., txt) files. |
| **Assignment 1 graded on Wattle** | Mon 22 Apr 2019 at 5:00 PM | After familiarising yourself with your mark, please feel free to visit Hanna during her consultation hours from 4 to 5 PM on Mon 29 Apr or 6 May 2019. |

**Assignment Didactic**

Seeing in action at packet level is a concrete way to learn the layered network model. In this individual assignment, you will explore one of the most common protocols on the Internet, the *H*y*perText Transfer Protocol* (HTTP) under the hood and its relationship with the lower layers, including the use of the *Transmission Control Protocol* (TCP).

The assignment consists of the three activities: 1. Examine basic HTTP Request/Response, 2. Set up a simple web server and capture HTTP Request/Response and 3. Explore HTTP Request/Response with Wireshark (network analysis software) against the assignment web server on the Internet. Notice that the weeks 2, 3 and 5 tutorial labs have dedicated exercises to support these activities.

You may use the Linux machine (Ubuntu 16.04, 64 bit edition) in the CECS Lab or their own Windows or Mac computers for this assignment.

After completing this assignment, you should be able to 1. Understand how the Web works behind the scene, in particular, the interactions between a browser and server, 2. Explain how the application layer in one computer communicates with its matching layer in another computer, using a concrete HTTP example, and 3. Explain the concept of *encapsulation* in the layered network model.

**General Instructions**

***Grading:*** To gain the full points, you need to impress me; hard and smart work pays off.

“The trouble of opportunity is that it comes disguised as hard work.”

Herbert V. Prochnow, 1897-1998

Assignment 1 and 2 (max 40 + 60 = 100 points) are graded. In total, their marking is from 0 to 100 points and forms a hurdle; at least 40 points in total is required to pass the unit. The total assignment points will be scaled to form 30 per cent of the unit grade.

***Feedback:*** Students will be given written feedback.

***Policies:*** ANU has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University’s academic standards, and implement them. You can find the University’s education policies and an explanatory glossary at <http://policies.anu.edu.au/>.

***Academic Misconduct:*** Students are expected to have read the Academic Misconduct Rule before the commencement of their course. Other key policies include Student Assessment (Coursework) and Student Surveys and Evaluations.

***Turnitin:*** The ANU is using Turnitin to enhance student citation and referencing techniques, and to assess assignment submissions as a component of the University's approach to managing Academic Integrity. However, you may choose not to submit assessment items through Turnitin. In this instance you may be required to supplement your submission with copies of all references included in the assessment item. For additional information regarding Turnitin please visit the ANU Online website.

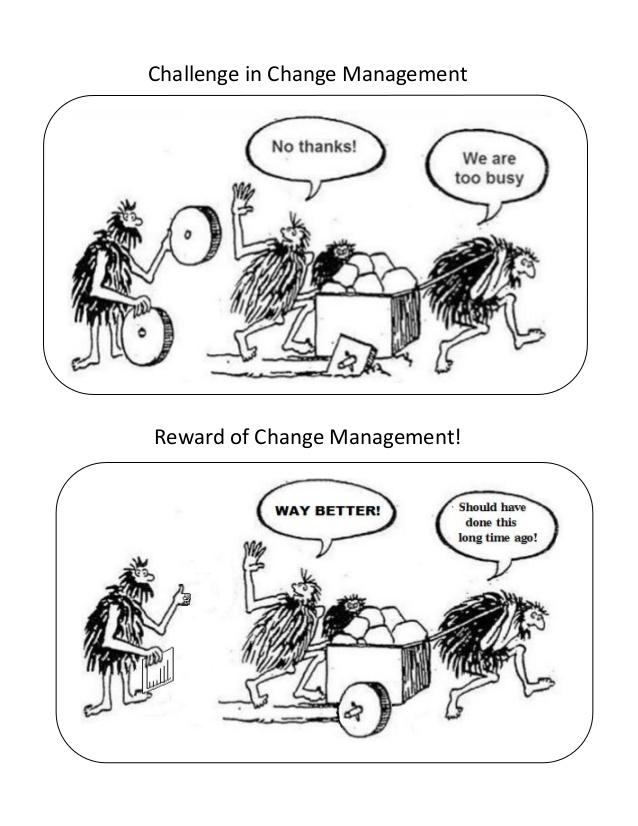
***Referencing requirements:*** Appropriate referencing is required (see <https://academicskills.anu.edu.au/taxonomy/term/142>). Please follow the Harvard style (see <http://www.anu.edu.au/students/learning-development/academic-integrity/style-guides>).

***Assignment submission:*** Assignments are submitted using the course Wattle site. You will be required to press a **submit button** to electronically sign a declaration as part of your submission. Please keep a copy of the submitted content for your records. No submission (or resubmission) after the due date will be permitted. If an assessment task is not submitted by the due date, a mark of 0 will be awarded.

***Help is available:*** Remember to challenge yourself but at the same time, be realistic with your ability to deliver on time. We do provide help and support in tutorial labs and also on Wattle and Piazza. Do not hesitate to ask for advice in order to be efficient. However, we are here to support your learning journey; doing the hard and smart work is your responsibility.

***Time management:*** Also, remember that the first go at something never works and writing takes time. Schedule yourself to spend, for example, Weeks 1 and 2 on the first version of your solution, Weeks 3 and 4 on reporting it, Week 5 on improving your solution, and Week 6 on finalising your report.

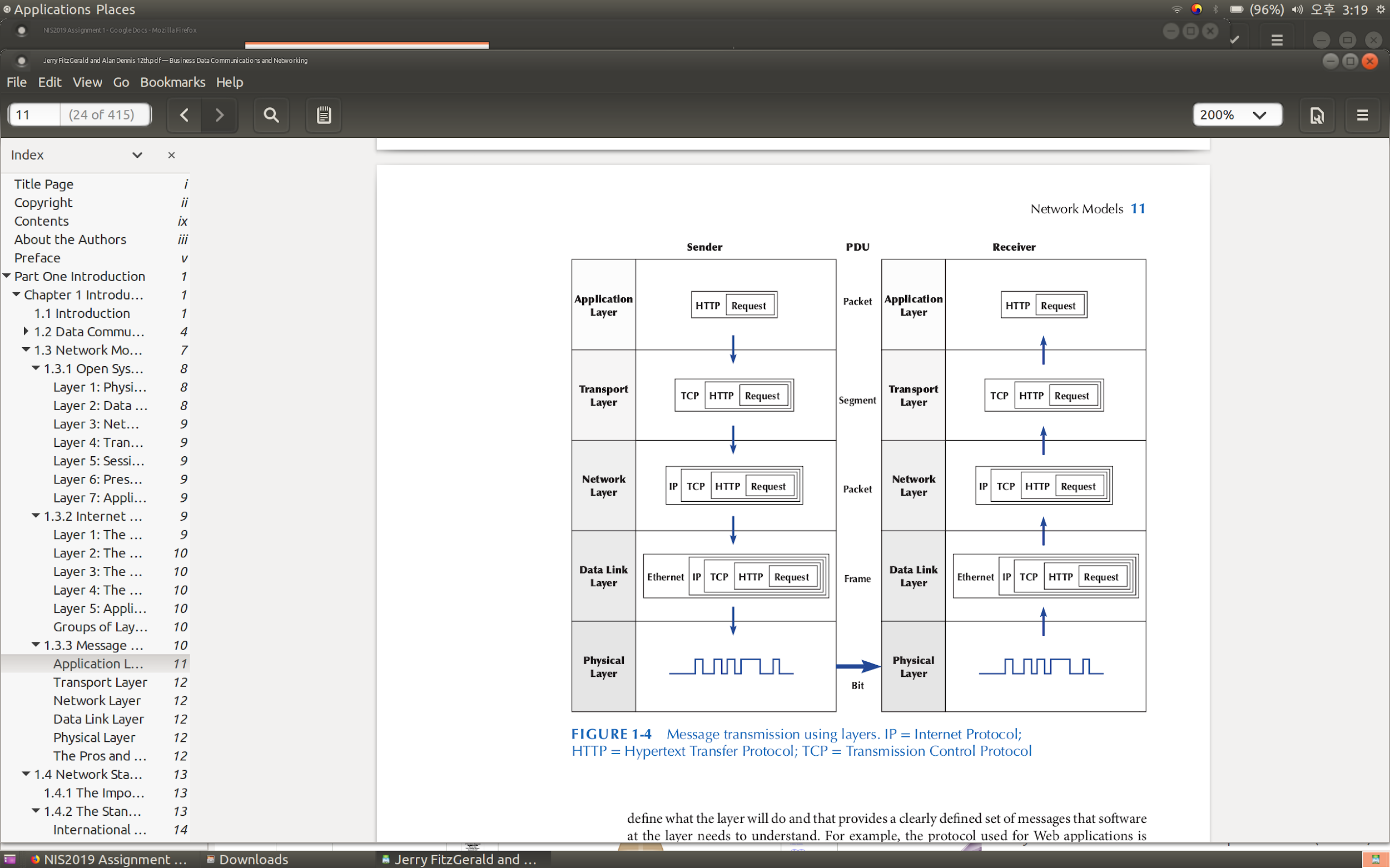
***Marking:*** Please see the rubric below for our indicative grading guidelines. They supplement the general marking guidelines of The ANU Research School of Computer Science.



Source: <https://www.slideshare.net/TonyFarias/cartoon-no-thanks-too-busy>

## **Basic HTTP Request and Response**

The Textbook Figure 1-4 shows message transmission using layers. More specifically, it shows how a message requesting a Web page (HTTP Request) would be sent on the Internet.



**Q1.** Modify Figure 1-4 to represent HTTP Response that shows message transmission using layers and describe briefly the modified figure (max 5 points).

## **Setting up a Simple Web Server**

Now, you run a simple web server based on the http.server module in Python 3 (or SimpleHTTPServer in Python 2). Note that the Python package is already installed on the Linux machines in the CECS Labs. On your own computer, you might have to install the [Python](http://www.python.org) package and the module.

Create a subdirectory “httpsrv” at your home directory and a text file called index.html at the subdirectory. Please write the following content in the index.html file. Notice that you should use your own uni id instead of *uxxxxxxx* in this file.

|  |
| --- |
| <HTML>  <HEAD>  <TITLE>NIS2019 Assignment Page</TITLE>  </HEAD>  <BODY>  <H1>NIS2019 Assignment Test Page for uxxxxxxxx</H1>  <HR>  </BODY>  </HTML> |

Change your current directory to “httpsrv” and run on of the following commands:

$ python -m http.server 8000

or

$ python -m SimpleHTTPServer 8000

Notice that you might have to run python2 (or 3) instead of python, depending on the Python version.

Alternatively, you may wish to install and run a production quality web server (for example, Apache HTTP Server) on your own computer. In this case, you might have to create index.html at a different subdirectory (not “httpsrv”).

[Curl](https://curl.haxx.se/) is a command line toolfor transferring data with *Universal Resource Locators* (URLs). It supports a variety of application layer protocols and it is widely available on Linux, Mac, and Windows. In particular, this tool allows you to send HTTP Requests to a web server and to receive HTTP Responses from the server. For example, on a terminal session, you can run

$ curl -v <http://13.211.159.241:80>

**Q2.** Find ‘*the\_IP\_address*’ of the computer on which you run your web server. Run the command below with *the\_IP\_address* on another computer and then describe briefly the curl output (max 5 points):

$ curl -v http://*the\_IP\_address*:8000

Notice that for the question Q2, you should use two communicating computers in the CECS Labs (or your own devices), physical or virtual, and include two IP addresses of your web server and curl client computer) in your report. You could also choose to use, for example, Apache HTTP Server here.

**Q3.** Use your web browser and enter http://*the\_IP\_address*:8000 at the address field. Capture the screenshot of the web browser and describe briefly what happens between your web browser and web server (max 5 points).

## **Exploring HTTP with Wireshark**

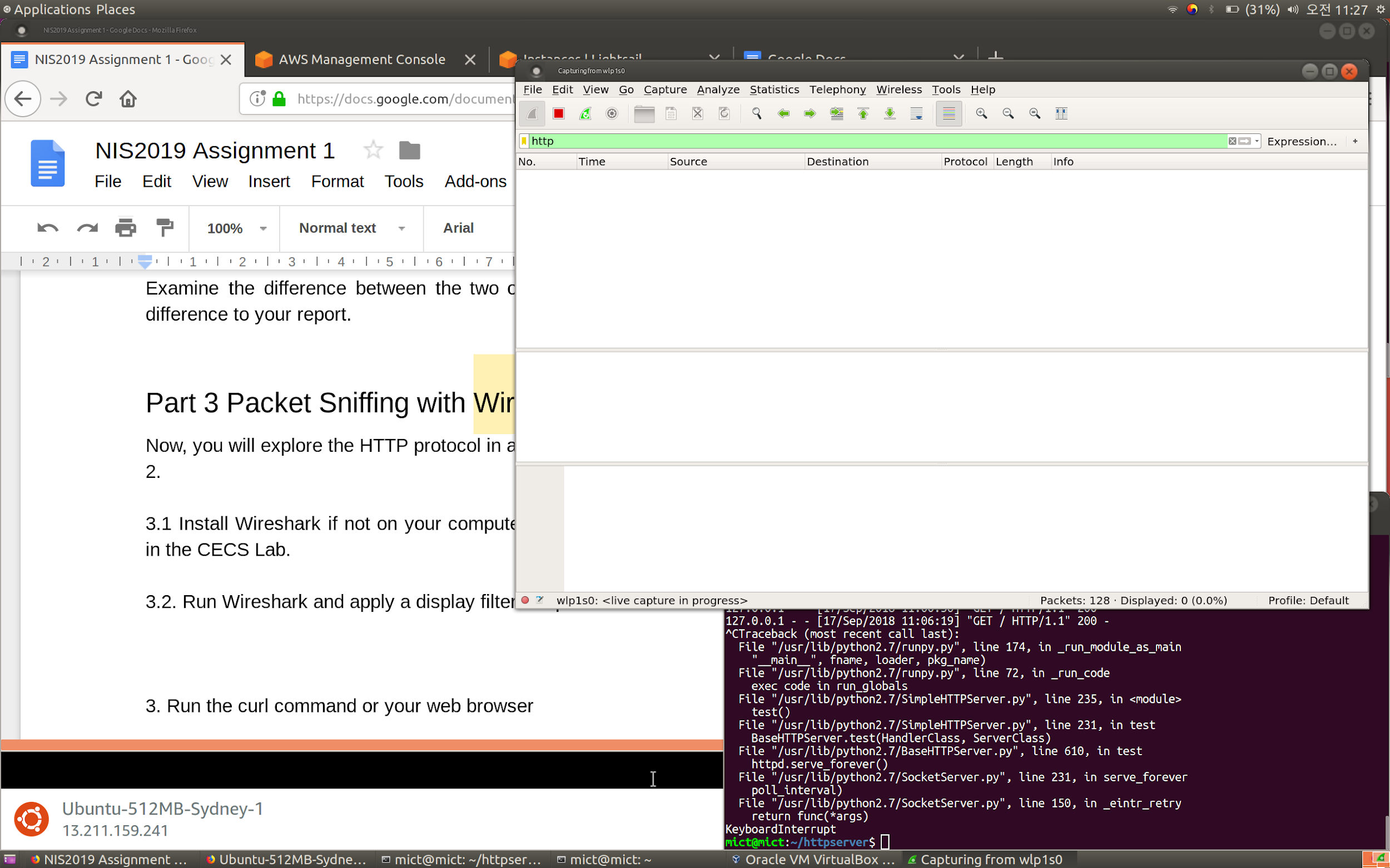
[Wireshark](http://www.wireshark.org) is a free and open source network protocol analyser widely used in industry and academia. Wireshark allows you to monitor what your computer sends to another on the network and what your computer receives at each layer from the application layer down to the physical layer. More specifically, you will monitor network traffic between your computer and the designated web server on the Internet.

Wireshark is installed on the computers in the CECS Labs. It is widely available on Linux, Mac and Windows so you may wish to install/use it on your own computer. However, the CECS IT has kindly enabled super user (sudo) rights for all Networked Information Systems students in the CECS Labs for the purposes of this assignment.

In the CECS Labs, run the command in a terminal session:

$ gksudo wireshark

In Wireshark, apply a display filter (in green) - enter http in the display filter.



While Wireshark is running, run the curl command:

$ curl http://13.211.159.241:80

**Q4.** Examine Wireshark output panels and identify in the HTTP Request (max 15 points):

1. User-Agent:
2. Host:
3. What is the length of HTTP Request (including the lower layers)?
4. What is the size of TCP payload for the HTTP Request?

Identify in the HTTP Response message in Wireshark:

1. Server:
2. What is the length of HTTP Response (including the lower layers)?
3. What is the size of TCP payload for the HTTP Response?

Notice that you should include the Wireshark screenshots for 1-7 in your report.

**Q5.** Describe briefly the concept of encapsulation in the context of Q4 (max 10 points).

Now, when you have completed all the five activities of Q1-Q5, it is time to perfect your report. Please give evidence of your academic writing skills by following the straight line writing and key messaging instructions in your report. Pay careful attention to concise information delivery; be brief but comprehensive and think about your report structure. Remember to save it as one PDF document for its submission.

**Indicative Assessment Rubric**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Max  points | Exemplary  (80%-100% of the points) | Excellent  (70%-79% of the points) | Good  (60%-69% of the points) | Acceptable  (50%-59% of the points) | Unsatisfactory  (0%-49% of the points) |
| Q1 | 5 | Exceptional report | Excellent figure;  Excellent description | Correct figure;  Good description | Correct figure;  Partial or incorrect description | Incorrect or no figure |
| Q2 | 5 | Production quality web server used  (e.g., Apache HTTP server);  Exceptional report | Correct curl output;  Excellent description of curl output;  Two IP addresses given | Correct curl output;  Good description | Correct curl output;  Partial or incorrect description | localhost or 127.0.0.1 used |
| Q3 | 5 | Exceptional report | Correct screen capture with uid;  Excellent description | Correct screen dump with uid;  Good description | Correct screen capture  with uid;  Partial or incorrect description | No screen capture of a web browser or  no uid |
| Q4 | 15 | Exceptional report | Excellent answers;  Wireshark screen captures to support answers | Good answers to Q4.3,4.4, 4.6,4.7 | Partial screen captures;  Partial or incorrect answers | No screen dump of Wireshark |
| Q5 | 10 | Exceptional report | Excellent description;  Q4.4 and Q4.7 answers used | Good description | Concept explained |  |
| Total | 40 |  |  |  |  |  |