

Rahul Madala

Assignment 1 Report

February 24, 2020

Assignment 1

Task 1:

Application Layer Protocol:

Types of messages exchanged: request-response, where the request was made by the server and the responses were made by the client.

Message Syntax: In order to separate the sentence input and options input, print statements were used to indicate to the user what input to type. These inputs were stored in separate variables and then concatenated when the socket sent this information to the server. The inputs stored in these variables were separated by a line break so that the server could differentiate the sentence from the options input. The server then takes the two inputs that were combined in one string, and separates the string into the sentence and list of options to execute which is then split up into separate options. The server then sends the modified sentence in each option as one whole string back to the client. This information is printed out by the client.

Message Semantics: The fields used in this protocol to represent the messages to be sent were sentence and options on the client side, and textInput, sentence, optionsList, and finalOutput on the server side. On the client side, the sentence variable represents the sentence part of the user input which will be modified by the server. The options variable represents the collection of operations to be used by the server to modify the sentence. On the server side, textInput was used to store the entire user input of the client. This was then broken up into sentence and optionsList, with the sentence representing the original sentence, and optionsList representing the collection of operations to be used on the sentence. As these options were executed, each part of the modified sentence was added to the finalOutput variable, which is the variable sent back to the client and printed out. This finalOutput represents the result of all the operations used in the program.

Rules:

When the server program initiates, the server creates the socket needed for communication and prints out "The server is ready to receive". This line dictates when it is acceptable for the client to send messages to the server. When the client initiates and the user enters its input, the message is

sent to the server which validates the input, modifies it, and returns it back to the client. Once the client program receives the message and prints out the modified sentence to the user, the program terminates, however, the server is still active and can accept a new run of the client program.

Task 2:

Question 1: 224 packets were detected by Wireshark during the server and client application run. The average packet size was 501.69. The average rate of number of packets captured per second by Wireshark was 11.2 packets/second.

Question 2: In this run of Wireshark there was no packet that had the HTTP or Hypertext protocol.

Question 3: The other IP addresses found in this Wireshark protocol besides 10.0.0.77 were IP addresses 13.83.149.67, 65.55.44.109, 2a01:111:f307:1790, and 2601:5c1:4402:d4e0.

Question 4:

Source Port: 49830

Destination Port: 443

Source IP: 10.0.0.77

Destination IP: 65.55.44.109

IP Header length: 20 bytes

Sequence Number: 611

TCP Payload: 1164 bytes.

Total Length: 1218 bytes

Question 5: TCP Stream Output (see TCP stream images)