Computer Networking
Coursework Assignment

Birkbeck, University of London

Academic Year 2019/20

The assignment requires two programs (client and server) in Java or Python, that implement the game described below.

The client has to be coded according to the following specification.

- At startup, the client prompts for three strings S_1, S_2, S_3 from the user.
- The client then:
  1. sends S_1, S_2 and S_3 to the server in three distinct UDP packets;
  2. waits for a string R and an integer n from the server (n is expected to be the sum of the lengths S_1, S_2 and S_3);
  3. after removing all characters '`#`' from R, checks that the result is the same as the concatenation of S_1, S_2 and S_3;
  4. checks if the received n is the correct sum of the lengths of S_1, S_2 and S_3;
  5. displays the results of the checks.

Example: The user enters S_1 as `net`, S_2 as `wor`, and S_3 as `king`. It calculates the sum ℓ of the lengths of the three strings, that is $\ell = 10$; it also calculates the the concatenation of S_1, S_2 and S_3, that is C=`networking`. The client sends the three strings in three packets and then it receives R as `net#wor#king` as well as n as 10. The client then removes the hash characters from R obtaining `networking`; it checks that this is the same as C; then it checks that $n = \ell$. Finally, it displays R and n, as well as the results of the two checks (this could be two `YES`/`NO` strings or any suitable message).

The server has to be coded according to the following specification.

- The server constantly waits for UDP messages.
- The server waits for *a sequence of three distinct UDP packets* containing strings S_1. S_2 and S_3.
- The server then:
  1. computes R by concatenating S_1, S_2 and S_3, adding two characters '`#`' as separators;
  2. computes n as the sum of the lengths of S_1, S_2 and S_3;
  3. sends R and n to the client as two distinct UDP packets.

Example: the server receives S_1 as `net`, S_2 as `wor`, and S_3 as `king`; computes $n = 3 + 3 + 4 = 10$ and R as `net#wor#king`. Finally, it sends R and n to the client.

The student has to produce the result of the coursework as follows.

1. The student will produce Java or Python code for the client and the server, as above specified.

2. The student will produce a brief, informal description of the code (PDF file; *no Microsoft Word or other formats*). The description should include screenshots of two execution of a client/server pair. An actual image has to be produced as a screenshot; one screenshot for each pair (two in total) will be sufficient (separate screenshots for each window are of course allowed).

3. The screenshots, accompanied by a brief explanation, will have to show a window for the client and one for the server; the client and the server will be coded so as to display suitable messages showing the execution of each step.

All the code and documentation (two source files for client and server plus a PDF file) will have to be uploaded in a *Zip* file named according to the following template: `surname_name_id.zip`.

The deadline for the submission is the 26th of April, 2020, at 23:55 London time (UTC+1). All late submissions, even if they are late by a very short time, will be capped at 40, unless a Mitigating Circumstances claim is approved. Students are strongly encouraged not to submit at the very last minute, so as to avoid technical problems which may cause a late submission.