CS F303 Computer Networks: 2019-20 semester II

Assignment 3

Do this assignment in groups of 5. This grouping may be different from the other assignments. Only one group member is required to submit. The deadline for this assignment is 22nd April, 2020, 23:59 hours.

You are developing an application which can't tolerate the growth and shrinking of the TCP congestion window, but requires reliable delivery. Therefore, you are required to use UDP as the transport protocol, but build reliability in the application layer.

1. Design your own application layer protocol which uses UDP sockets to transfer packets between a client and a server (or a P2P setup, your choice). Your protocol should be able to take care of sequence numbers, acknowledgements, retransmissions, and what ever else is required to run your application smoothly. **You are required to submit a pdf file with your protocol specification.** The specification document should contain the assumptions about the application, the network, the strategies to tackle different situations in the network, etc. Maximum length of this document is 2-3 pages, 11 point font for non header text.
2. Write a Java class to implement the protocol you designed. Other Java programs should be able to instantiate your class and use different methods in the class. Some example methods can be:
    1. `MyReliableUDPSocket.create()`
    2. `MyReliableUDPSocket.read()`
    3. `MyReliableUDPSocket.write()`
    4. etc…

    You need to submit this java file
3. Write a toy network application in Java (chat, messaging, file transfer, etc.) which uses the class you designed in the previous step. You can choose your own application, it does not need to be elaborate, it should be able to use and demonstrate the reliability features of your protocol. **You need to submit this(ese) java file(s)**
4. **Include a readme.txt file with instructions to compile and run your application**
5. **Submit a pdf file with plots of the performance of your protocol under different network conditions.** e.g. throughput under various packet loss, delays, reordering, packet corruption conditions (use the netem tool to emulate different network conditions on your localhost, see the Evaluation heading below for *netem*).

Evaluation: Submit a single zip file with all the above asks. Only one group member is required to submit. Your application will be tested under various network conditions using the *netem* tool (https://wiki.linuxfoundation.org/networking/netem). e.g. 50% packet loss, 90% packet loss, varying delays, jitter, packet corruption, reordering, rate, etc. Your protocol should be able to work under these conditions, or inform the source about the failure to deliver. You will be required to demo the assignment to the TAs or instructors or both. The modalities of the demo will be decided depending on the Covid-19 pandemic situation.

Collaboration within a group: Given the extraordinary situation we are operating under, group assignments can be a challenge but not impossible to execute. Use Google Meet, Google Docs, Gitlab, etc. to collaborate. There are enough sub-tasks in the assignment on which the team members can work independently. (The initial plan for the assignment was to ask two groups to design a common protocol but implement them independently and test the inter-compatibility of the two implementations. Given the time and geographical constraints, we are foregoing this part)

Collaboration outside a group: The class is encouraged to discuss and collaborate, but cheating in any form is strongly discouraged. Use piazza for discussions so that all discussions are in the open. **Include a file with the names of people you have discussed with to maintain transparency.** There is almost no scope of overlap between two groups. Your submission will be passed through code similarity and document similarity checkers. Any form of cheating will be dealt with along declared policies. The bottom of this page has a good reference on cheating vs. collaboration: http://www.cse.iitd.ac.in/~mausam/courses/col772/spring2019/

Start early, the deadline is a hard deadline as the institute has a hard deadline for pre-comprehensive evaluations. Divide the days into protocol design, protocol implementation and application implementation based on how much effort each of these components requires. Hint: a clear protocol covering all corners is easier to implement.