CMP2204 Introduction to Computer Networks

Term Project

Goal: The goal is to apply the concepts learned in class, through programming and hands-on practice. At the end of this project, you will have a better understanding of how a networked application operates and what are the technologies behind it.

Task: Design and implement a networked peer-to-peer chat application. All necessary protocols will be designed in class and the design document will be shared on itslearning. In your application you will need to implement the protocols as specified.

Requirements: The application should;

1. Have 4 processes: Service_Listener, Service_Advertiser, Chat_Listener, Chat_Client. These processes should work as outlined in their respective specifications.
2. Successfully detect all available users in the Local Area Network.
3. Successfully chat with any available user in the Local Area Network.
4. Display an error dialog if a message could not be delivered.
5. Output a chat log, containing timestamps and content of all messages exchanged in a chat session.

Important Notes: Deadline for the project is 23:59 on **Sunday, May 5**. Please commit all material under the "Term Project" assignment under Resources on itslearning. Your commits should include:

- All pieces of codes that you wrote. (Chat_Client, Chat_Listener, Service_Listener, Service_Advertiser).
- 4 executables for your platform of development for each of the four programs. (Please refer to http://www.pyinstaller.org/) This part is necessary for preserving code privacy.
- A README file describing how your program works, and on which platform (Windows/Linux/Unix) it has been tested.
- A file named *Requirements.txt* that specifies the versions of all Python libraries you've used in your code. I'll run *pip install Requirements.txt* before executing your code.
- A 1-page document describing which platform (Windows/Linux) you've used to develop your code, any faced challenges, and known limitations. (Also include group members' names here.)

Please name all your files as [XXX]_[filename] where XXX is your team members' initials.

You may work in groups of size **2 or 3**. You should determine a partitioning of responsibilities so that group members can work effectively in parallel.

Grading: Your commit is complete (includes 4 executables, 1 README, 1 report) **(20 pts)**. Your code runs and correctly discovers all available users **(15 pts)**, can *periodically* announce itself **(15 pts)**, can send a message to any online user **(10 pts)**, can receive message from any online user without delay **(10 pts)**, can correctly output a chat log under the same directory **(15 pts)**, easy to understand execution instructions in README, seamless user interactions (*e.g.,* displaying available users, etc.) **(15 pts)**. Anything else is a bonus.