CS490-004 - Network Programming
Programming Project (AKA, "Project #1)

Submission Due: 12:30pm, December 11th, 2026

I. Project Objectives:

The objectives of this programming project (#1) are (1) to learn the concept of multicast network applications and (2) to experience development of network applications using IP-multicast. For this project, each of you develops a multicast time server for a multicast client (your multicast server should work for the multicast client). Your multicast time server transmits (multicast) a timestamp to all participating multicast clients (one transmission of a timestamp message should be received multiple clients no matter how many multicast clients participate in the multicast group). The source code for the multicast client is provided by the course instructor. In this system, we are going to have one BBS server and multiple BBS clients (the process for clients is identical, meaning that the same process (program) is supposed to be running at every client (see Figure 1).

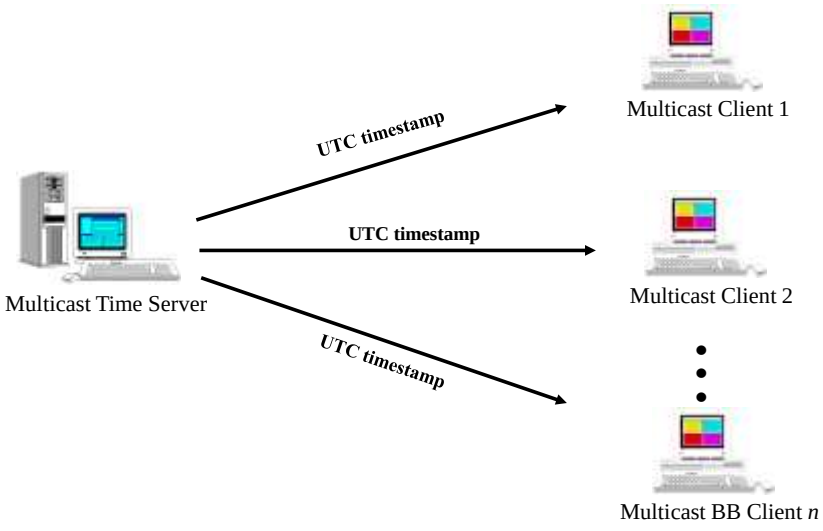

Figure 1 – Overview of the multicast time-server and client application

II. Required Process Internal Structure:

The multicast server and client are required to be structured as shown in Figure 2.

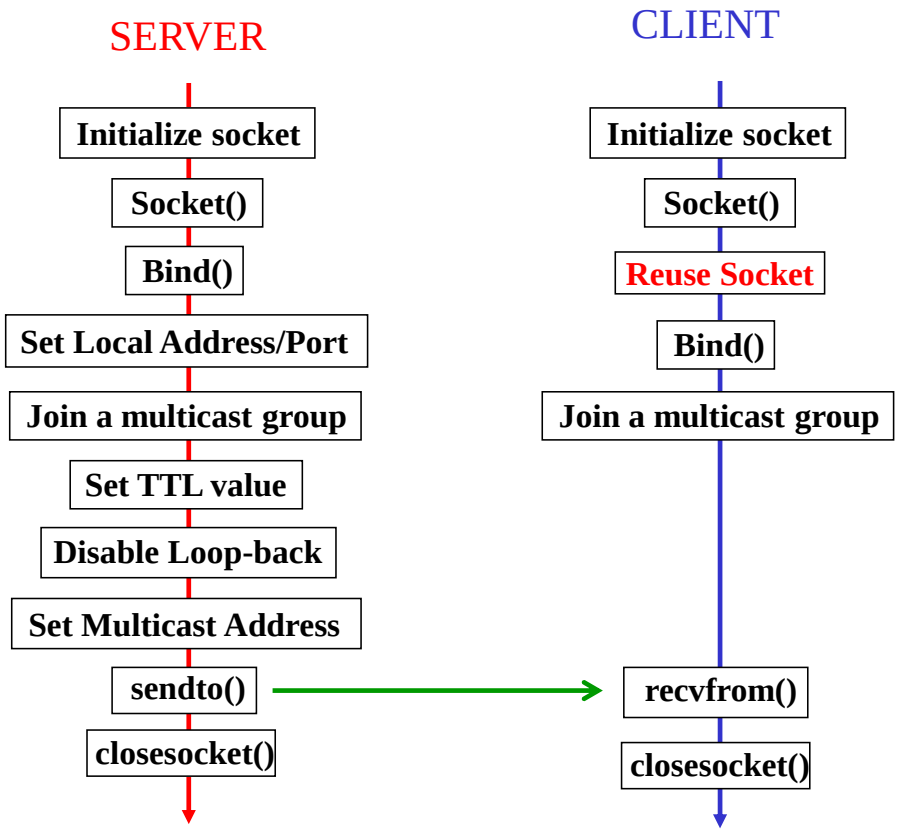

Figure 2 – Required Process Internal Structure for the multicast server and client

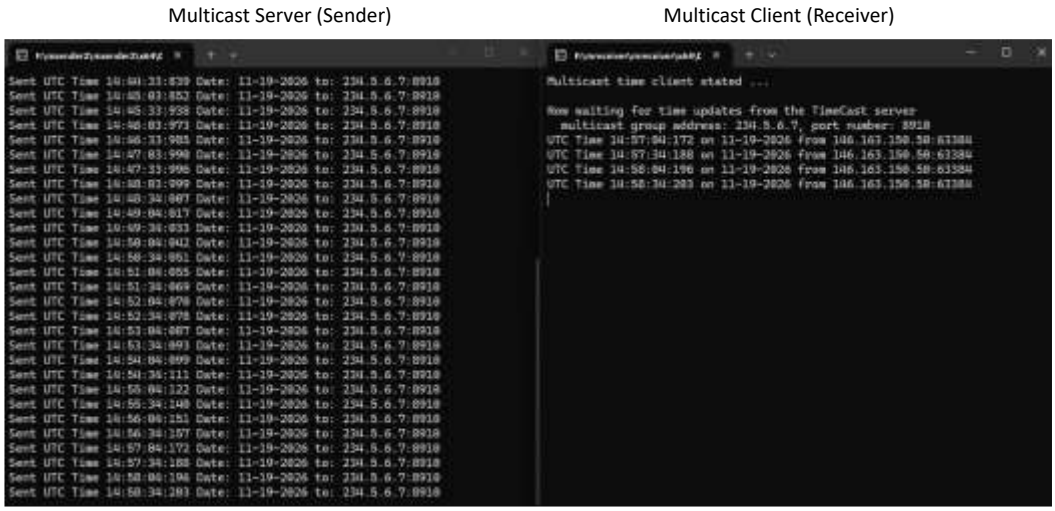

Figure 3 – Required Outputs

III. Project Requirements

(a) The source code file for your multicast-server should be implemented as a C++ (*.cpp) file which should be compiled by a MS Visual Studio (or Visual Studio Code) on a Windows-PC.

(b) The source code file for your multicast-server should be developed as a console-mode project.

(c) Your multicast timestamp server should display "Sent UTC Time: hh:mm:ss" (only) once for each time it transmits a timestamp (each timestamp should be a character string up to 256 ASCII characters).

(d) Each of you is assigned a class-D IP address (a.k.a., "multicast group address"). Everyone is expected not to use any class-D address other than the one assigned to each of you. Everyone is expected to use the same UDP port number (port: 8910).

(e) The TTL (Time-To-Live) parameter should be set to "2" (do not use TTL > 2 for overloading a network).

(f) Although this is not an essential component of this project, but just for simplification, everyone is expected to use the same timestamp generator (UTC timestamp generator). The source code of the UTC timestamp generator will be posted to the CS490-004 course home).

(g) This project is an individual programming project (the programming resources posted to the course website can be used in your source code file).

(h) Function-level error detection should be implemented (you do not have to implement recovery from errors from functions, but your program should display an error message when an error happened for every major function).

IV. Submission:

Submit your C++ source code file ("mt_server_*nnn*.cpp", where "*nnn*" is the last three digits of your student ID) by emailing it to hfujino@siue.edu.

V. Testing:

Your C++ source code file will be tested for the following points:

(1) Outputs must be correct.
(2) Programs should not crash during execution.
(3) All data transfers are completed without any missing data.
(4) Multicast data should be received by all the clients (the server does <u>not</u> have to receive the multicast data). Your server will be tested by changing the number of the multicast clients.

VI. Grading:

- Compile error: -80 points
- Serious runtime error: -10 points for each
- Minor runtime error: -5 points for each

VII. Suggested Reference:

- https://gist.github.com/hostilefork/f7cae3dc33e7416f2dd25a402857b6c6