cs425-mp1

C++ Multicast implementation. Option for simulated network delay and causal-ordered or total-ordered multicasting.

Directory Purposes

`src` — Holds all of the `.cpp` files that we wrote

`include` — Holds all of the `.h` files that we wrote

`build` — Holds all of the `.o` files that are used when compiling

`bin` — Holds all of the `.exe` files that are used to execute the search program

How to Use

The following command will start a process with supplied `pid` using the `port` and `ip` listed in `multicast.config`. The second argument chooses the ordering scheme (causal or total):

```
./bin/runner.exe <pid> <ordering_scheme=[causal,total]>
```

Total ordering requires the user to start up a sequencer along with the n processes as follows:

```
./bin/runner.exe <pid> total s
```

Implementation Details

Network delay simulation was accomplished by spawning n-1 threads (one for each process receiving the message) and sleeping them for a random duration of time bounded by `min_delay` and `max_delay` before sending. Note that sending to yourself doesn't incur any simulated delay. For causal ordering, we implemented the vector timestamp based algorithm outlined in the textbook on page 657. This method uses a hold-back queue to avoid delivering a message until it has delivered all messages causally preceding it. To multicast a message to group g, process p adds 1 to its entry in the timestamp and multicasts the message with the timestamp to group g. To meet the causal constraints, p_i will wait until it has delivered any earlier message sent by p_j and it has delivered any message that p_j had delivered at the time it multicast the message. Total ordering was implemented with the help of a sequencer. A process wishing to multicast a message will append a unique identifier (id) and pass the message along to the sequencer as well as members of g. The sequencer is implemented as its own process, and it is responsible for maintaining a group specific sequence number and assigning increasing and consecutive sequence numbers to messages it delivers. The sequencer announces messages by multicasting the ordered messages to group g.