

Course ISE 431: Distributed Information Systems

Recitation 12 Exercise

Michael J. May

January 5, 2010

In this lab we will further the development of the multicast application you developed last week. We will use the multicast nodes as a framework to develop a causally ordered messaging system.

To make the system we first need to put together a few tools.

1 Node Modifications

First, modify the nodes in your multicast tree application to wait a random time between 0.5 and 3 seconds between message propagation (you can use `sleep`). This will make it more interesting since messages may now arrive out of order from different parts of the tree.

2 Vector Time Stamp

Make a class or structure for vector time stamped text message. Make sure the class or structure is serializable so that it can be sent over the network easily.

Write a function to compare two vector time stamp messages with time stamps t_1 and t_2 for following conditions:

- Whether $t_1 < t_2$, $t_1 = t_2$, $t_1 > t_2$, or $t_1 \langle \rangle t_2$
- Whether t_2 is the “next in line” after t_1 (as in causally ordered multicast)

3 Causally Ordered Chat Room

3.1 Interface

Create a chatroom GUI which can display messages from different nodes on the multicast tree. The GUI window should have the following elements:

- A space to show “ready” messages (those which satisfy the COM conditions). Each message should be accompanied by the IP address of the sender and its vector time stamp.
- A textbox to enter a new message
- A “send new message” button which will submit the message to the entire multicast tree
- A space with messages that are “not ready yet”, those which do not satisfy the COM “next in line” condition.

The display should show the current vector time stamp for the client on the top or bottom of the window.

3.2 Sending

When a message is sent it should be accompanied by the COM time stamp for the sender and passed along the multicast tree.

3.3 Receiving

When a message is received along the multicast tree two things should happen:

- The message should be propagated along the tree to the parents/children as appropriate with a random delay as above
- The message should be passed to the chat client.

The chat client should put the message in the correct space:

- In the “ready space” if the message’s time stamp is ok
- In the “not ready yet space” if the message’s time stamp is not ok