CS 505 - Programming Assignment 2
Reliable Broadcast
Gowtham Kaki , Praseem Banzal, Suvidha Kancharla
31st March, 2014.

1. Changes in Reliable Channel layer:

According to the previous HW implementation, Reliable channel is a one to one channel between 2 processes. The channel was initiated with sender port, destination IP and destination port. Hence, the channel knew where to route the messages.

To implement reliable broadcast, we changed Reliable Channel implementation to a one to many channel, because we cannot have multiple channels listening on the same port (1 process should have 1 port according to the Process Interface). Ie, we use the same channel to send messages to multiple processes. To achieve this, each process maintain 2 mappings <ProcessID ,SendSeqNumbers> and <ProcID, RecvSeqNumbers>. Each Message maintains fields to which destination IP:Port it has to be routed to. Thus, the Sender/Receiver of Reliable Channel uses the sequence numbers from the mappings to send/accumulate the message to/from a particular destination IP:Port.

2. Reliable Broadcast:

Reliable Broadcast uses UDP Reliable Channels to broadcast/deliver messages.

Reliable Broadcast implementation:

- On Broadcast, broadcaster uses reliable channel to send the message to all the processes registered prior to broadcast, before delivering to itself.
- All processes register the callback of Reliable Channel to a method that checks if the message has been previously delivered. If not, it delivers the message (to a callback of Reliable Broadcast Receiver) and broadcasts to all the other processes.

Code

Edu.purdue.cs505.RChannel. * : Reliable Channel Implementation.
Edu.purdue.cs505.* : Code specific to RB, FIFO-RB, FIFO-SRB.

- ReliableBroadcastClass : Implements ReliableBroadcast Interface.
- ReliableChannelReceiver : Callback of RChannel (Implementation of Reliable Channel Interface).
- TestBroadcastReceiverClass : Our test Callback of RB. *<< You create your own callback class instead of this one while testing RB>>*
- Main.java: test1 method explains how to test RB. It creates 2 processes on 2 different machines. Each adds the other process to its broadcast group with addProcess. Each process broadcasts 50K messages. The receive of TestBroadcastReceiverClass prints a done message on success.

3. FIFO Reliable Broadcast

FIFO Reliable Broadcast uses Reliable Broadcast to broadcast/deliver messages.

FIFORB implementation:

- On FIFO RB, RB to all processes.
- All processes register callback of RB to the same method that takes care of FIFO order.
- The callback of RB maintains a message bag per process and waits until it receives the messages in FIFO order. It then delivers to the callback of FIFO RB.

Code

- FIFOReliableBroadcastClass : Implements FIFOReliableBroadcast Interface.

- FIFOReliableBroadcastReceiver : Callback of RB.
- TestFIFOBroadcastReceiverClass: Our test Callback of FIFO RB. *<< You create your own callback class instead of this one while testing FIFORB>>*
- Main.java : test2/test3 method explains how to test FIFORB . They create multiple processes on the same/different machines. One process FIFORB's messages. All others deliver in the FIFO order. The receive of TestFIFOBroadcastReceiverClass prints Done on success.

4. Semantic FIFO Reliable Broadcast

Semantic FIFO RB uses FIFO RB to broadcast/deliver messages.

SemanticFIFORB implementation

- SemanticFIFORB uses FIFORB to broadcast messages.
- All processes register callback of FIFORB to the same method that takes care of Semantic order.
- The callback of FIFORB waits for deliverydelay ms until it stashes the obsolete messages. Non-obsolete messages are delivered to the callback of SemanticFIFORB.

Code

- SrbMessage : Extends Message. Its similar to Message with a few extra fields specific to SemanticFIFORB. To clearly separate concerns and avoid clutter in Message, we have created this class.
- We didn't want you to take the trouble of remembering to set SrbOn = true for testing SemanticFIFORB or set SrbOn=false for testing FIFORB. Hence, we created separate classes for testing Semantic RB.
- FIFOSemanticReliableBroadcastClass : Implements FIFOSemanticReliableBroadcast Interface.
- FIFOSrbReceiverClass : Callback of FIFO RB.
- TestFIFOSrbReceiverClass : Our Callback of SemanticFIFORB. *<< You create your own callback class instead of this one while testing Semantic FIFO RB.>>*
- Main.java : test4 method tests SemanticFIFORB. A broadcaster broadcasts multiple messages and a few successive messages that make the previous messages obsolete. The receive of TestFIFOSrbReceiverClass verifies if all messages other than obsolete messages are received and prints Done on success.

Note :
To test Semantic FIFO RB:

** Please use SrbMessage to create messages instead of Message class.
** Please use FIFOSemanticReliableBroadcastClass to create an instance of the implementation of FIFOReliableBroadcast instead of using FIFOReliableBroadcastClass with SrbOn = true.