Project-1 Report

Introduction:

Basically we have tried to build an android application which can send messages to other android applications over network. Messages are delivered in a “Total-Causal” order.

Design Details:

1. **Algorithm**

   At each process P:-

   - When each process needs to multicast a message **m**, with message ID **m_ID** to group **g**, will b-multicast its message with “**normal**” tag to all members including sequencer.
   - Message to sequencer should be sent after it’s been sent to all others.
   - When message arrives at a process:-
     - First message (**normal or order**)corresponding to a **m_id** id is put into **holdback queue**
     - When second message (**order or normal**) corresponding to that **m_id** is received then this message id popped from this hold back queue and put in a **delivery queue**
     - This delivery queue (priority queue) ensure that messages are delivered in correct order irrespective of the order in which **order** messages are received

   At sequencer S:-

   - When message arrives, sequencer increments it’s sent **order** message count and attaches it to new message and b-multicasts to group **g**.

2. **Pseudo code**

   At each process p:

```
On initialization: rev_seq_id = -1
To To_multicast message m to group g
    m.tag = “normal”
    FIFO_multicast (m, g – sequencer)
    FIFO_unicast (m, sequencer)
On FIFO_deliver message m
    If (m.m_id != in hold back queue)
        Place m in holdback queue, with key m.m_id
    Else
        Fetch message m from holdback queue with key = m.m_id
        Place m in delivery queue after updating its seq_id
TO_deliver(M)
While(delivery queue .size != 0 && delivery queue.head.seq_id == rev_seq_id + 1)
    rev_seq_id = rev_seq_id + 1;
    remove head and deliver message.
```

   At Sequencer S:

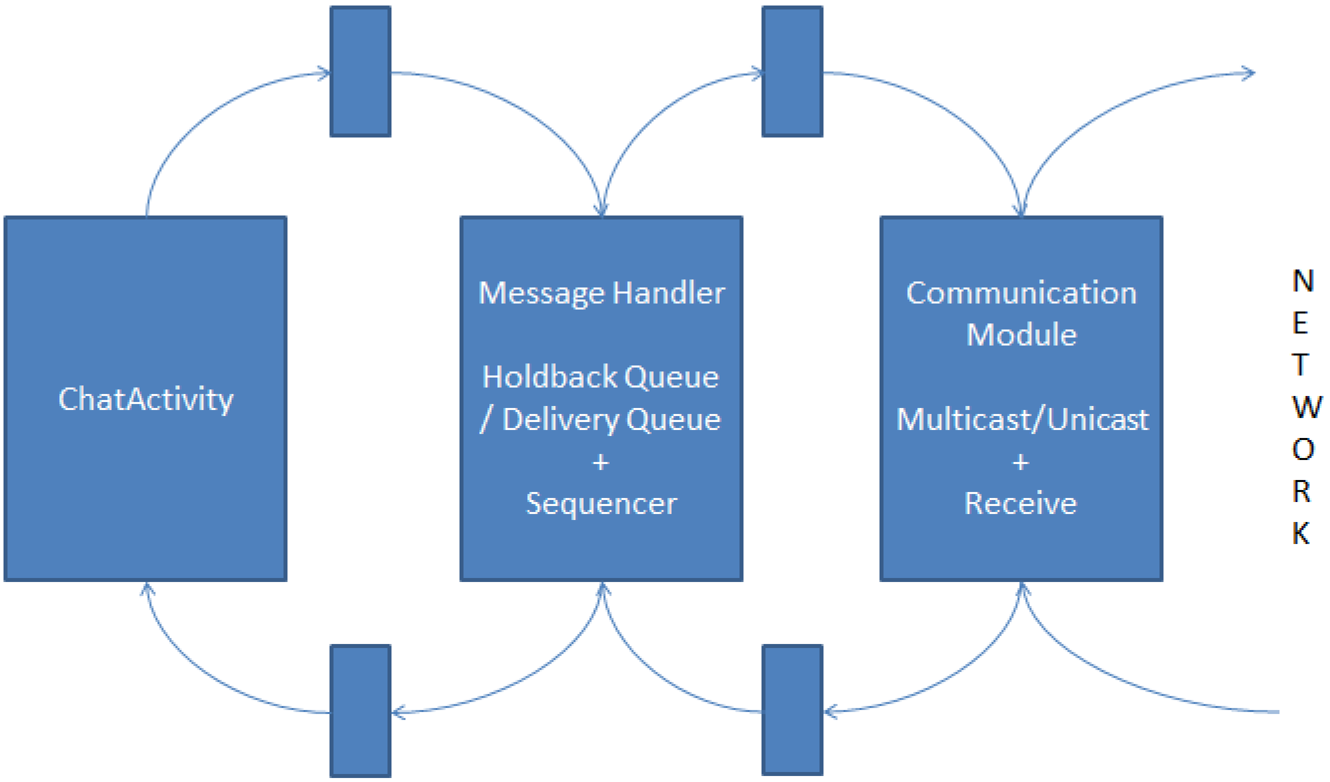
```
On initialization: seq_id = -1;
On FIFO_deliver message m
    seq_id = seq_id + 1
    m.seq_id = seq_id
    m.tag = “order”
    FIFO_multicast(m, g)
```

3. Brief description of source files,

MsgAppV2Activity.java	Primary purpose of this activity is get number of applications you want to connect with.
ChatActivity.java	Takes user input – normal message, test1 and test2
MsgHndlngModule.java	It takes user input form ChatActivity and passes it to

	Communication module after adding suitable header to support Total-Causal order
CommModule.java	Provides functions for multicating, unicasting, initiating sequencer election, etc.
Sequencer.java	Assigns sequence number to message and multicast
MsgPasser.java	Used to pass message between different threads of Chat activity, MsgHndlngModule and CommModule
Message.java	Defines message object format
MsgDbAdapter.java	Content provider
PeerAddress.java	Peer address format
Utility.java	It contains static functions and constants to be used by all

Major Components



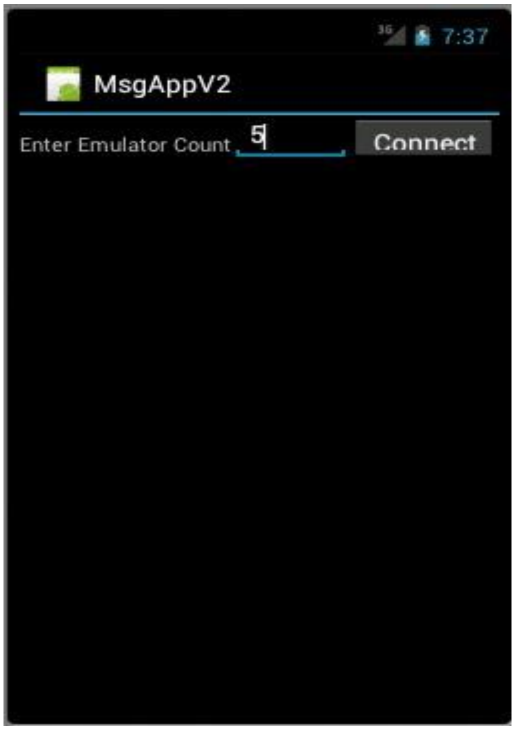
References

1. Tutorials on,
   http://developer.android.com/
   http://thinkandroid.wordpress.com/2010/01/13/writing-your-own-contentprovider/
2. Discussion on Piazza and all-time favorite Google.

Steps To Run:

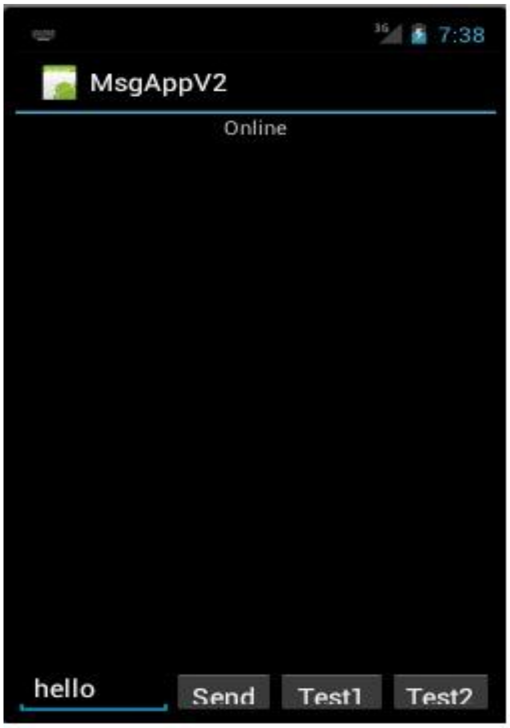
1. Launch activity
2. Enter number of devices (Screen shot 1)

   Note: Currently ports for number of devices are fixed. That is, if number of devices is 1 then emulator number 5554 must be on. If it is 3 then emulator number 5554, 5556 and 5558 must be on. And So on.
3. Take user input (Screen shot 2)



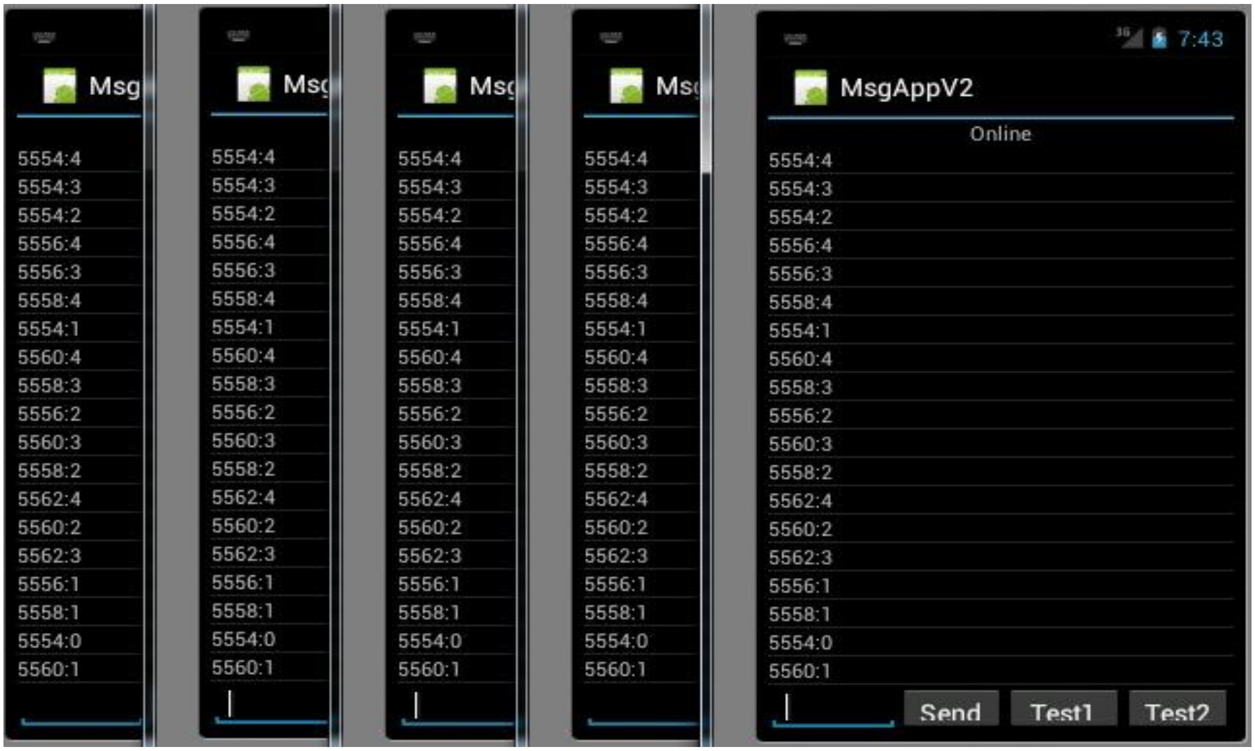
Enter Emulator Count 5

Screen shot 1



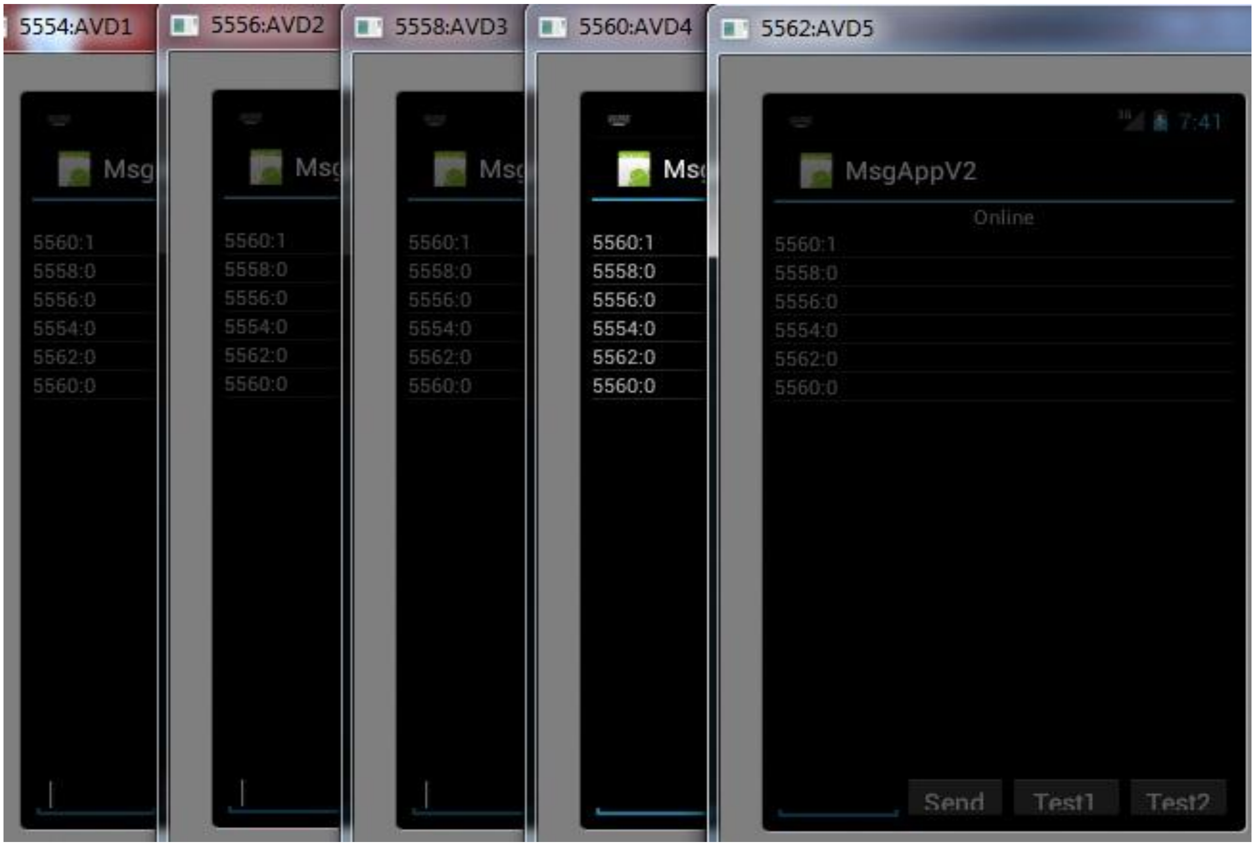
hello

Screen shot 2



```
5554:4
5554:3
5554:2
5556:4
5556:3
5558:4
5554:1
5560:4
5558:3
5556:2
5560:3
5558:2
5562:4
5560:2
5562:3
5556:1
5558:1
5554:0
5560:1
```

Screen shot 3 (TestCase1)



5554:AVD1

```
5560:1
5558:0
5556:0
5554:0
5562:0
5560:0
```

5556:AVD2

```
5560:1
5558:0
5556:0
5554:0
5562:0
5560:0
```

5558:AVD3

```
5560:1
5558:0
5556:0
5554:0
5562:0
5560:0
```

5560:AVD4

```
5560:1
5558:0
5556:0
5554:0
5562:0
5560:0
```

5562:AVD5

```
5560:1
5558:0
5556:0
5554:0
5562:0
5560:0
```

Screen shot 4 (TestCase 1)