



ROYAL INSTITUTE
OF TECHNOLOGY
OF TECHNOLOGY
ROYAL INSTITUTE

Royal Institute of Technology

MSc. Software Engineering of Distributed Systems

ID2203 Distributed Systems Advanced Course

Homework 2

Andrei Shumanski
andreish@kth.se
0046707761992

Trigonakis Vasileios
vtri@kth.se
0046707694420

Stockholm 2010

TABLE OF CONTENTS

Exercise 2	4
Algorithm	4
Algorithm Explanation	5
Book's algorithm criticism & our algorithm's improvements	5
Criticism of our algorithm	6
Question 5	7
5(a)	7
5(b)	8
5(c)	10
5(d)	12
Exercise 2	13
Question 1	13

EXERCISE 2

ALGORITHM

Implements:

ProbabilisticBroadcast (pb).

Uses:

FairLossPointToPointLinks(flp2p);

UnreliableBroadcast(un).

```
1. upon event <Init> do
2.     forall  $P_i \in \Pi$  do
3.         delivered[ $P_i$ ] := 0;
4.         missing[ $P_i$ ] := 0;
5.         lsn := 0; stored := 0;
6.
7. procedure gossip (msg) is
8.     forall  $t \in \text{pick-targets (fanout)}$  do
9.         trigger <flp2pSend |  $t, \text{msg}$ >;
10.
11. upon event <pbBroadcast |  $m$ > do
12.     lsn := lsn+1; trigger <unBroadcast | [Data, self, m, lsn]>;
13.
14. upon event <unDeliver |  $P_i, [\text{DATA}, S_m, m, \text{SN}_m]$ > do
15.     if (store-threshold < random()) then
16.         stored := stored  $\cup$  { [DATA,  $S_m, m, \text{SN}_m$ ] };
17.     if ( $\text{SN}_m \geq \text{delivered}[S_m] + 1$ ) then
18.         trigger < pbDeliver |  $S_m, m$  >;
19.         forall seqnb  $\in [\text{delivered}[S_m] + 1, \text{SN}_m - 1]$  do
20.             gossip ([REQUEST, self,  $S_m, \text{seqnb}, \text{maxrounds} - 1]$ );
21.             missing[ $P_i$ ] := missing[ $P_i$ ]  $\cup$  seqnb;
22.             delivered[ $S_m$ ] :=  $\text{SN}_m$ ;
23.         else if ( $\text{SN}_m \in \text{missing}[S_m]$ ) then
24.             missing[ $S_m$ ] := missing[ $S_m$ ]  $\setminus \text{SN}_m$ ;
25.             trigger < pbDeliver |  $S_m, m$  >;
26.
27. upon event <flp2pDeliver |  $P_j, [\text{REQUEST}, P_i, S_m, \text{SN}_m, r]$ > do
28.     if ([DATA,  $S_m, m, \text{SN}_m$ ]  $\in$  stored) then
29.         trigger <flp2pSend |  $P_i, [\text{DATA}, S_m, m, \text{SN}_m]$ >;
30.     else if ( $r > 0$ ) then
31.         gossip ([REQUEST,  $P_i, S_m, \text{SN}_m, r - 1]$ );
32.
33. upon event <flp2pDeliver |  $P_j, [\text{DATA}, S_m, m, \text{SN}_m]$ > do
34.     if ( $\text{SN}_m \in \text{missing}[S_m]$ ) then
35.         missing[ $S_m$ ] := missing[ $S_m$ ]  $\setminus \text{SN}_m$ ;
36.         trigger < pbDeliver |  $S_m, m$  >;
```

ALGORITHM EXPLANATION

1. Lines 1-5: initialization of the structures that it uses. It does not use a pending array, but instead it is using an array of sets of integers where a node can keep the missing messages from the other nodes. As it appears, this structure was anyway needed by the book's algorithm, if we wanted to increase the probability of receiving messages that were delayed more than the timeout that the algorithm uses.
2. Lines 7-9: the same as the book's algorithm. Note: when a node is forwarding the request of another node, it should not pick the requester node
3. Lines 11-12: the same as the book's algorithm
4. Lines 14-25: handling of an message coming from the unreliable broadcast. Again, randomly it may store a received message or not. Then (line 17), if the node receives a new message (message with higher sequence number from the sending node), then it **immediately** delivers it, starts a gossip for each "missing" message and place the missing node to the missing structure. The changes that we made, allow us not to use timer for our algorithm to work. This is a big improvement, because we no more any timing assumptions, so our algorithm can work (almost) equally well in an asynchronous system. Finally, if we receive a message that belongs to the missing structure (the case were due to the network two messages are coming with the wrong order) then we deliver it and remove it from the missing list.
5. Lines 27-31: the same as the book's algorithm
6. Lines 33-36: when we receive a deliver from the flp2p link (coming from a gossip response), if it belongs to the missing we remove it and deliver it, else we simple ignore it.

BOOK'S ALGORITHM CRITICISM & OUR ALGORITHM'S IMPROVEMENTS

The algorithm of the book had some severe logic and performance problems. Here they are, with an explanation how we did solve them:

1. In order to call the deliver-pending procedure, it should receive a message from gossip with a s/n next to the one that it had already delivered. That means that if a message is totally lost (we have a probabilistic broadcast anyway) then the algorithm will block.
Solution: we deliver messages immediately, if the should be delivered, else we skip them
2. Combined with the problem no. 1, when it receive a timeout about a missing message, it does not handle the case where we didn't receive the proper "message". The algorithm is somehow "stubborn" and does not accepts to name some messages as lost and continue running.
Solution: we do not use timeout at all
3. If due to network delays, two messages from the same node are unDelivered with the wrong order (i.e. first message with s/n 9 and then message with s/n 8, both from process 2), then the second one is dropped by the algorithm. Instead of this it could deliver it and resolve some pending messages faster!
Solution: we handle the case when two messages come from in reverse order by checking into the missing list for the second message. If it exists, we deliver it and remove it from the missing list.
4. Except from the locking issues, the book's algorithm (tries to) implements ordered broadcast. As it is obvious, this a stronger assumption than what we need that cannot be implemented by our probabilistic algorithm. Even if it was possible, this strong assumption delays the delivery of the received messages for no reason.
Solution: we deliver the messages as soon as we receive them
5. When the algorithm receives a gossip data message, if it not the "correct" message, it is always added in the pending set. This means that pending set can contain duplicate pending messages, which is inefficient.
Solution: we add the missing messages in the list, only in one place

CRITICISM OF OUR ALGORITHM

Our algorithm is correct, faster and simpler than the book's one. Also, because of the non-use of timeouts, it is more robust and easier to work over an asynchronous system. The only issue that we identified is the growing data structures. Both stored and missing data structures are getting larger as the time passes.

Solution:

- A simple solution that can be applied is to restrict the message of the structures and shift the oldest values out of them. For example, if we say that we mostly store 40 messages, when the structure gets full, the oldest item is thrown for a new one to take place. If we use this technique, we can also implement an interesting variation of the algorithm. We could store all the new messages coming, throwing the older ones. This could help us increase the chance of delivery of a message because new messages have more chance to be asked than older one. Of course this technique would be usable only with a correct combination of message delivery frequency, data structure size and network delays.
- A more complex solution could be to have dynamic structures that increase (and decrease) their size according to the node's activity, assuming the we using the previous solution also. This could guaranty that if the node is under stress due to many losses, it will not lose messages due to full data structures, but in the common case, its structures are not big, so they do not affect its performance.

QUESTION 5

Lazy probabilistic broadcast depends on three parameters: fanout, store-threshold and maxrounds. Discuss in your report how do these parameters affect the broadcast. In your report, describe executions by varying the values of these parameters and the topology characteristics that lead to the following scenarios.

5(A)

No message is lost.

We set lost of the messages in the network to 0:

```
defaultLinks(1000, 0);
```

In this case all messages are delivered by unreliable broadcast so gossip is never executed:

```
Process 1 - 81
Terminal Process
4112 INFO (LazyPB) 1 - PB: 2 :: 18Shum-PC:22031|2
4116 INFO (LazyPB) >Delivery & till now: 100.0
4120 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 2
B3
7822 DEBUG (Application2) Broadcasting :: 3
7823 INFO (LazyPB) 1 - PB: 3 :: 18Shum-PC:22031|3
7831 INFO (LazyPB) >Delivery & till now: 100.0
7833 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 3
B4
10830 DEBUG (Application2) Broadcasting :: 4
10832 INFO (LazyPB) 1 - PB: 4 :: 18Shum-PC:22031|4
10838 INFO (LazyPB) >Delivery & till now: 100.0
10839 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 4
B5
14011 DEBUG (Application2) Broadcasting :: 5
14012 INFO (LazyPB) 1 - PB: 5 :: 18Shum-PC:22031|5
14015 INFO (LazyPB) >Delivery & till now: 100.0
14017 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 5
B6
17043 DEBUG (Application2) Broadcasting :: 6
17044 INFO (LazyPB) 1 - PB: 6 :: 18Shum-PC:22031|6
17052 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|6
17052 INFO (LazyPB) >Delivery & till now: 100.0
17052 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 6

Process 2 - 51
Terminal Process
18148 SCENARIO (Assignment2Main) Process 2 has started commands [S1].
0 DEBUG (LazyPB) lazyPBroadcast :: started
2 DEBUG (Application2) Application2 :: started
3 DEBUG (SimpleUnreliableBroadcast) unBroadcast :: started
128 INFO (Application2) Sleeping 1 milliseconds...
298 INFO (Application2) DONE ALL OPERATIONS
1684 INFO (LazyPB) >Delivery & till now: 100.0
1686 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 1
5512 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|2
5513 INFO (LazyPB) >Delivery & till now: 100.0
5514 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 2
9230 INFO (LazyPB) >Delivery & till now: 100.0
9235 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 3
12233 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|4
12233 INFO (LazyPB) >Delivery & till now: 100.0
12234 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 4
15408 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|5
15410 INFO (LazyPB) >Delivery & till now: 100.0
15410 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 5
18447 INFO (LazyPB) >Delivery & till now: 100.0
18449 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 6

Process 3 - 51
Terminal Process
18698 SCENARIO (Assignment2Main) Process 3 has started commands [S1].
0 DEBUG (LazyPB) lazyPBroadcast :: started
5 DEBUG (Application2) Application2 :: started
7 DEBUG (SimpleUnreliableBroadcast) unBroadcast :: started
10 INFO (Application2) Sleeping 1 milliseconds...
343 INFO (Application2) DONE ALL OPERATIONS
1806 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|1
1807 INFO (LazyPB) >Delivery & till now: 100.0
1808 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 1
5955 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|2
5956 INFO (LazyPB) >Delivery & till now: 100.0
5957 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 2
9299 INFO (LazyPB) >Delivery & till now: 100.0
9313 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 3
12306 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|4
12306 INFO (LazyPB) >Delivery & till now: 100.0
12307 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 4
15486 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|5
15486 INFO (LazyPB) >Delivery & till now: 100.0
15488 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 5
18516 INFO (LazyPB) >Delivery & till now: 100.0
18525 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 6

Process 4 - 51
Terminal Process
16848 SCENARIO (Assignment2Main) Process 4 has started commands [S1].
0 DEBUG (LazyPB) lazyPBroadcast :: started
0 DEBUG (Application2) Application2 :: started
0 DEBUG (SimpleUnreliableBroadcast) unBroadcast :: started
3 INFO (Application2) Sleeping 1 milliseconds...
55 INFO (Application2) DONE ALL OPERATIONS
1544 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|1
1544 INFO (LazyPB) >Delivery & till now: 100.0
1544 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 1
5298 INFO (LazyPB) >Delivery & till now: 100.0
5309 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 2
9016 INFO (LazyPB) >Delivery & till now: 100.0
9027 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 3
12017 INFO (LazyPB) >Delivery & till now: 100.0
12025 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 4
15197 INFO (LazyPB) >Delivery & till now: 100.0
15207 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 5
18240 INFO (LazyPB) 2 - Storing msg: 18Shum-PC:22031|6
18240 INFO (LazyPB) >Delivery & till now: 100.0
18240 DEBUG (Application2) Broadcast msg: 18Shum-PC:22031 : 6
```

So the result doesn't depend on *fanout*, *store-threshold* and *maxrounds* parameters because they affect only gossip phase of the algorithm. All nodes deliver all messages during unreliable broadcast and we can use only it.

5(B)

A broadcasted message is lost in the unreliable broadcast but recovered by gossip for some node p .

To model lost of some messages during unreliable broadcast we made network lost 50%, this is a lot but we can see how algorithm works in extreme cases:

```
defaultLinks(1000, 0.5);
```

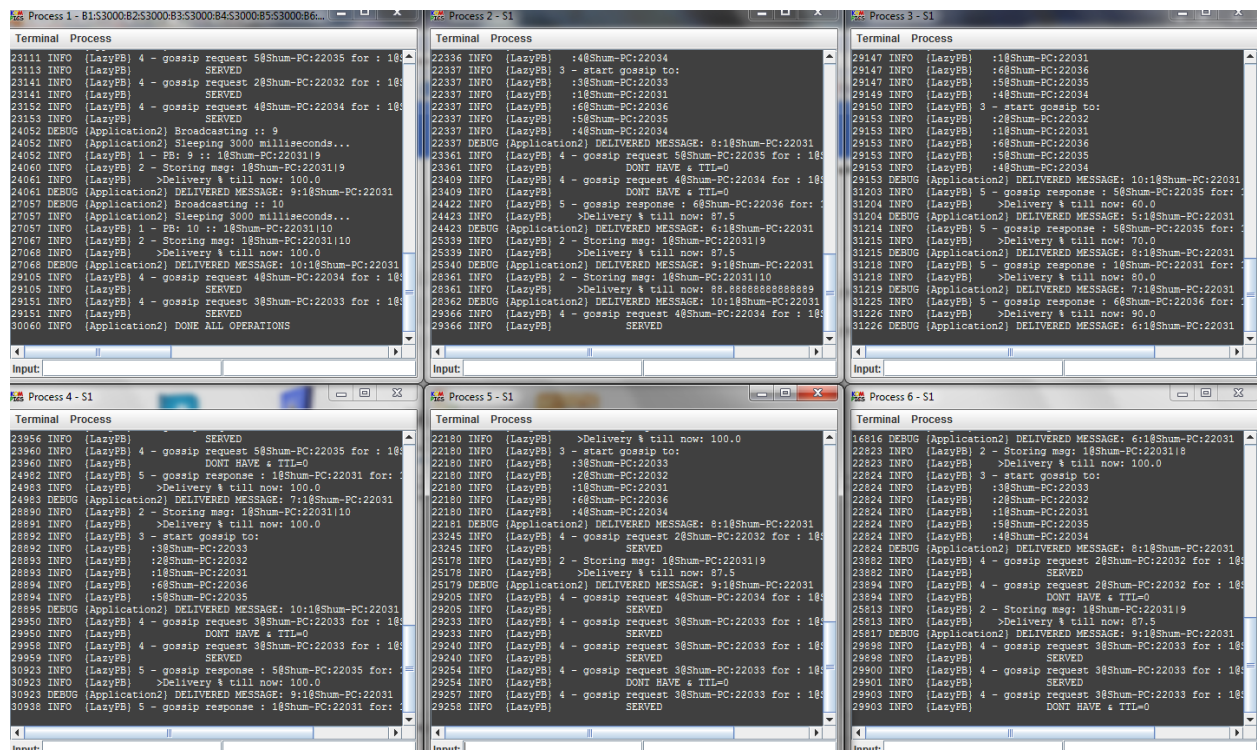
As we use fair-loss link we cannot GUARANTEE that all message will be recovered as we can lose all gossip messages.

If we want to increase chances to recover message at gossip phase we should increase *store-threshold* to be sure that if some node delivers the message it will be able to send it during the gossip. We also should increase *fanout* and *maxrounds* number of fanouts increases chance of finding the node with stored message and maxrounds increases chance to find lost message from the nodes which are not connected as fanouts.

In our experiment we used 6 nodes because with greater number of nodes it becomes difficult to analyze outputs.

```
private static final double storeTreshold = 1;
private static final int fanouts = 5;
private static final int ttl = 1;
```

In this configuration all nodes are connected to all others as fanouts and store-treshhold is 100% so we should lose messages only if it was lost during the unreliable broadcast and gossip phase.



```
Process 1 - b153000825300083530008453000853300086...
Terminal Process
23111 INFO (LazyFB) 4 - gossip request 58Shum-PC:22035 for : 18
23113 INFO (LazyFB) SERVED
23144 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
23141 INFO (LazyFB) SERVED
23152 INFO (LazyFB) 4 - gossip request 48Shum-PC:22034 for : 18
23153 INFO (LazyFB) SERVED
24052 DEBGS (Application2) Broadcasting : 9
24052 INFO (LazyFB) 1 - FB: 9 : 18Shum-PC:22031
24060 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031
24061 INFO (LazyFB) >Delivery & till now: 100.0
24061 DEBGS (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
27057 DEBGS (Application2) Broadcasting : 10
27057 INFO (Application2) Sleeping 3000 milliseconds...
27057 INFO (LazyFB) 1 - FB: 10 : 18Shum-PC:22031
27057 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031
27068 INFO (LazyFB) >Delivery & till now: 100.0
27068 DEBGS (Application2) DELIVERED MESSAGE: 10:18Shum-PC:22031
28105 INFO (LazyFB) 4 - gossip request 48Shum-PC:22034 for : 18
28105 INFO (LazyFB) SERVED
29151 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29151 INFO (LazyFB) SERVED
30060 INFO (Application2) DONE ALL OPERATIONS

Process 2 - S1
Terminal Process
22336 INFO (LazyFB) 4:48Shum-PC:22034
22337 INFO (LazyFB) 3 - start gossip to:
22337 INFO (LazyFB) 18Shum-PC:22033
22337 INFO (LazyFB) 18Shum-PC:22031
22337 INFO (LazyFB) 68Shum-PC:22036
22337 DEBGS (Application2) DELIVERED MESSAGE: 8:18Shum-PC:22031
22337 INFO (LazyFB) 48Shum-PC:22034
23361 INFO (LazyFB) 4 - gossip request 58Shum-PC:22035 for : 18
23361 INFO (LazyFB) DONT HAVE & TTL=0
23409 INFO (LazyFB) 4 - gossip request 48Shum-PC:22034 for : 18
23409 INFO (LazyFB) DONT HAVE & TTL=0
24422 INFO (LazyFB) 5 - gossip response : 68Shum-PC:22036 for :
24423 INFO (LazyFB) >Delivery & till now: 87.5
24423 DEBGS (Application2) DELIVERED MESSAGE: 6:18Shum-PC:22031
25339 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031
25339 INFO (LazyFB) >Delivery & till now: 87.5
25340 DEBGS (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
28361 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031
28361 INFO (LazyFB) >Delivery & till now: 88.88888888888889
28362 DEBGS (Application2) DELIVERED MESSAGE: 10:18Shum-PC:22031
29366 INFO (LazyFB) 4 - gossip request 48Shum-PC:22034 for : 18
29366 INFO (LazyFB) SERVED

Process 3 - S1
Terminal Process
29147 INFO (LazyFB) 18Shum-PC:22031
29147 INFO (LazyFB) 68Shum-PC:22036
29147 INFO (LazyFB) 18Shum-PC:22035
29149 INFO (LazyFB) 48Shum-PC:22034
29150 INFO (LazyFB) 3 - start gossip to:
29153 INFO (LazyFB) 28Shum-PC:22032
29153 INFO (LazyFB) 18Shum-PC:22031
29153 INFO (LazyFB) 68Shum-PC:22036
29153 INFO (LazyFB) 98Shum-PC:22035
29153 INFO (LazyFB) 48Shum-PC:22034
29153 DEBGS (Application2) DELIVERED MESSAGE: 10:18Shum-PC:22031
31203 INFO (LazyFB) 5 - gossip response : 58Shum-PC:22035 for :
31204 INFO (LazyFB) >Delivery & till now: 60.0
31204 DEBGS (Application2) DELIVERED MESSAGE: 5:18Shum-PC:22031
31214 INFO (LazyFB) 5 - gossip response : 58Shum-PC:22035 for :
31215 INFO (LazyFB) >Delivery & till now: 70.0
31215 DEBGS (Application2) DELIVERED MESSAGE: 8:18Shum-PC:22031
31218 INFO (LazyFB) 5 - gossip response : 18Shum-PC:22031 for :
31218 INFO (LazyFB) >Delivery & till now: 80.0
31219 DEBGS (Application2) DELIVERED MESSAGE: 7:18Shum-PC:22031
31225 INFO (LazyFB) 5 - gossip response : 68Shum-PC:22036 for :
31226 INFO (LazyFB) >Delivery & till now: 90.0
31226 DEBGS (Application2) DELIVERED MESSAGE: 6:18Shum-PC:22031

Process 4 - S1
Terminal Process
23956 INFO (LazyFB) SERVED
23960 INFO (LazyFB) 4 - gossip request 58Shum-PC:22035 for : 18
23960 INFO (LazyFB) DONT HAVE & TTL=0
24982 INFO (LazyFB) 5 - gossip response : 18Shum-PC:22031 for :
24983 INFO (LazyFB) >Delivery & till now: 100.0
24983 DEBGS (Application2) DELIVERED MESSAGE: 7:18Shum-PC:22031
28890 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031
28891 INFO (LazyFB) >Delivery & till now: 100.0
28892 INFO (LazyFB) 3 - start gossip to:
28892 INFO (LazyFB) 18Shum-PC:22033
28893 INFO (LazyFB) 28Shum-PC:22032
28893 INFO (LazyFB) 18Shum-PC:22031
28894 INFO (LazyFB) 68Shum-PC:22036
28894 INFO (LazyFB) 18Shum-PC:22035
28895 DEBGS (Application2) DELIVERED MESSAGE: 10:18Shum-PC:22031
29950 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29950 INFO (LazyFB) DONT HAVE & TTL=0
29950 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29950 INFO (LazyFB) 18Shum-PC:22032
29950 INFO (LazyFB) 5 - gossip response : 58Shum-PC:22035 for :
29950 INFO (LazyFB) >Delivery & till now: 100.0
29950 DEBGS (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
29950 INFO (LazyFB) 5 - gossip response : 18Shum-PC:22031 for :

Process 5 - S1
Terminal Process
22180 INFO (LazyFB) >Delivery & till now: 100.0
22180 INFO (LazyFB) 3 - start gossip to:
22180 INFO (LazyFB) 18Shum-PC:22033
22180 INFO (LazyFB) 28Shum-PC:22032
22180 INFO (LazyFB) 18Shum-PC:22031
22180 INFO (LazyFB) 68Shum-PC:22036
22180 INFO (LazyFB) 48Shum-PC:22034
22181 DEBGS (Application2) DELIVERED MESSAGE: 8:18Shum-PC:22031
23245 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
23245 INFO (LazyFB) SERVED
25178 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031
25178 INFO (LazyFB) >Delivery & till now: 87.5
25179 DEBGS (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
29205 INFO (LazyFB) 4 - gossip request 48Shum-PC:22034 for : 18
29205 INFO (LazyFB) SERVED
29233 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29233 INFO (LazyFB) SERVED
29240 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29240 INFO (LazyFB) SERVED
29254 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29254 INFO (LazyFB) DONT HAVE & TTL=0
29257 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29257 INFO (LazyFB) SERVED
29258 INFO (LazyFB) SERVED

Process 6 - S1
Terminal Process
16816 DEBGS (Application2) DELIVERED MESSAGE: 6:18Shum-PC:22031
22223 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031
22223 INFO (LazyFB) >Delivery & till now: 100.0
22224 INFO (LazyFB) 3 - start gossip to:
22224 INFO (LazyFB) 18Shum-PC:22033
22224 INFO (LazyFB) 28Shum-PC:22032
22224 INFO (LazyFB) 18Shum-PC:22031
22224 INFO (LazyFB) 68Shum-PC:22036
22224 INFO (LazyFB) 48Shum-PC:22034
22224 DEBGS (Application2) DELIVERED MESSAGE: 8:18Shum-PC:22031
23882 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
23882 INFO (LazyFB) SERVED
23894 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
23894 INFO (LazyFB) DONT HAVE & TTL=0
25813 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031
25813 INFO (LazyFB) >Delivery & till now: 87.5
25817 DEBGS (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
29898 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29898 INFO (LazyFB) SERVED
29900 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29901 INFO (LazyFB) SERVED
29903 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
29903 INFO (LazyFB) DONT HAVE & TTL=0
```

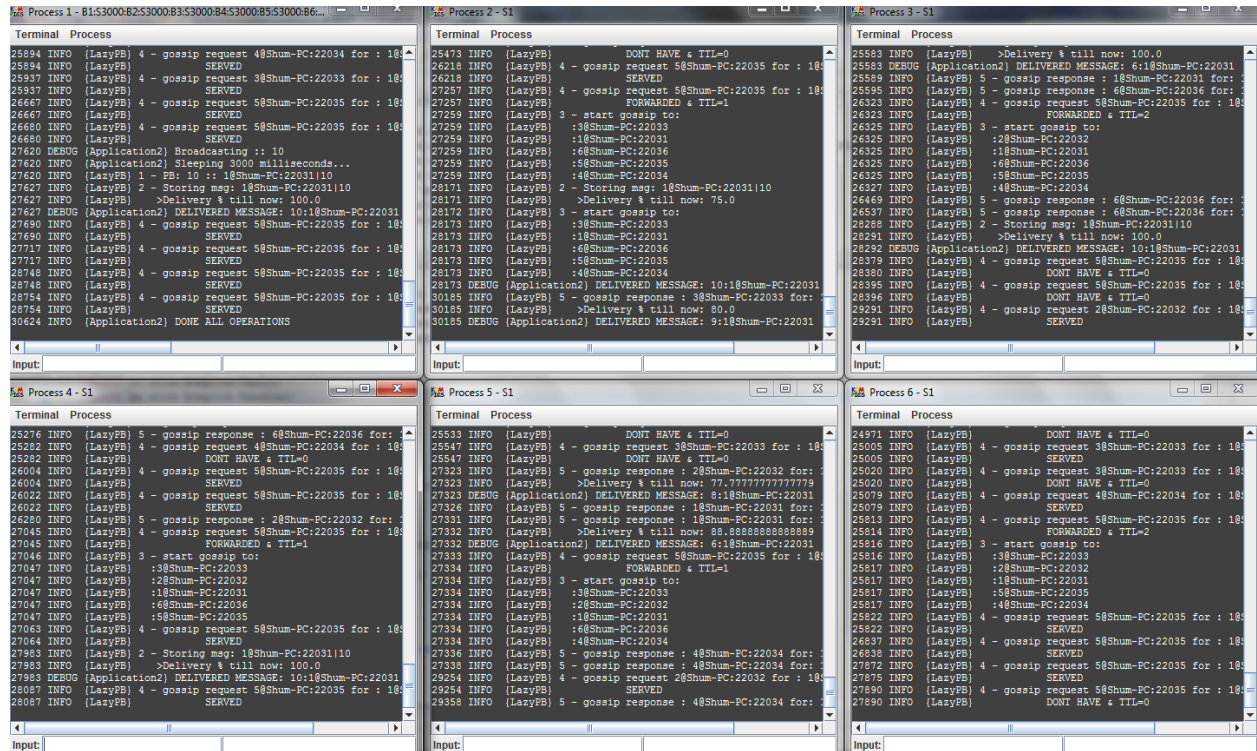

2 nodes delivered 100% of messages and 4 nodes delivered ~90% of messages.

Let's increase maxrounds to 2, it should increase the chances:

```
private static final double storeTreshold = 1;

private static final int fanouts = 5;

private static final int ttl = 3;
```



4 nodes delivered 100% of messages, one node – 80% and one node – 90%, so delivery rate really increased.

5(C)

A broadcasted message is lost such that although it is stored on some node(s) in the network, a node *p* missed it in the unreliable broadcast and furthermore, *p* could not retrieve it via gossiping as well.

To model such situation we keep the network lost rate 50%:

```
defaultLinks(1000, 0.5);
```

Now not all nodes that deliver message should store it, so we decrease *store-threshold* to 30% and we also decrease *fanouts* to 3 to make chances of gossip failure higher:

```
private static final double storeThreshold = 0.3;
```

```
private static final int fanouts = 3;
```

```
private static final int ttl = 1;
```

```
Process 1 - B1:53000:B2:53000:B3:53000:B4:53000:B5:53000:B6:
Terminal Process
21134 DEBUG (Application2) Broadcasting : 8
21134 INFO (Application2) Sleeping 3000 milliseconds...
21136 INFO (LazyFB) 1 - FB: 8 :: 18Shum-PC:22031|8
21150 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031|8
21151 INFO (LazyFB) >Delivery & ttl now: 100.0
21151 DEBUG (Application2) DELIVERED MESSAGE: 8:18Shum-PC:22031
23209 INFO (LazyFB) 4 - gossip request 48Shum-PC:22034 for : 18
23209 INFO (LazyFB) DONT HAVE & TTL=0
24136 DEBUG (Application2) Broadcasting : 9
24136 INFO (Application2) Sleeping 3000 milliseconds...
24136 INFO (LazyFB) 1 - FB: 9 :: 18Shum-PC:22031|9
24140 INFO (LazyFB) >Delivery & ttl now: 100.0
24140 INFO (LazyFB) >Delivery & ttl now: 100.0
24143 DEBUG (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
27137 DEBUG (Application2) Broadcasting : 10
27137 INFO (Application2) Sleeping 3000 milliseconds...
27137 INFO (LazyFB) 1 - FB: 10 :: 18Shum-PC:22031|10
27142 INFO (LazyFB) >Delivery & ttl now: 100.0
27144 DEBUG (Application2) DELIVERED MESSAGE: 10:18Shum-PC:22031
29191 INFO (LazyFB) 4 - gossip request 68Shum-PC:22036 for : 18
29191 INFO (LazyFB) DONT HAVE & TTL=0
29213 INFO (LazyFB) 4 - gossip request 68Shum-PC:22036 for : 18
29214 INFO (LazyFB) SERVED
50139 INFO (Application2) DONE ALL OPERATIONS

Process 2 - 51
Terminal Process
17850 INFO (LazyFB) 3 - start gossip to:
17850 INFO (LazyFB) :18Shum-PC:22031
17850 INFO (LazyFB) :68Shum-PC:22036
17850 INFO (LazyFB) :48Shum-PC:22034
17851 DEBUG (Application2) DELIVERED MESSAGE: 6:18Shum-PC:22031
19915 INFO (LazyFB) 5 - gossip response : 18Shum-PC:22031 for :
19915 INFO (LazyFB) >Delivery & ttl now: 33.33333333333333
19915 DEBUG (Application2) DELIVERED MESSAGE: 4:18Shum-PC:22031
20821 INFO (LazyFB) >Delivery & ttl now: 33.33333333333333
20822 DEBUG (Application2) DELIVERED MESSAGE: 7:18Shum-PC:22031
24887 INFO (LazyFB) 4 - gossip request 48Shum-PC:22034 for : 18
24888 INFO (LazyFB) DONT HAVE & TTL=0
26849 INFO (LazyFB) >Delivery & ttl now: 42.857142857142854
26850 INFO (LazyFB) :48Shum-PC:22034
26850 INFO (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
26850 INFO (LazyFB) :68Shum-PC:22036
26850 INFO (LazyFB) :58Shum-PC:22035
26850 INFO (LazyFB) :48Shum-PC:22034
26853 DEBUG (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
27846 INFO (LazyFB) 4 - gossip request 38Shum-PC:22033 for : 18
27846 INFO (LazyFB) DONT HAVE & TTL=0
29839 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031|10
29839 INFO (LazyFB) >Delivery & ttl now: 44.444444444444444
29839 DEBUG (Application2) DELIVERED MESSAGE: 10:18Shum-PC:22031

Process 3 - 51
Terminal Process
11231 INFO (LazyFB) 4 - gossip request 68Shum-PC:22036 for : 18
11232 INFO (LazyFB) DONT HAVE & TTL=0
13149 INFO (LazyFB) >Delivery & ttl now: 50.0
13149 DEBUG (Application2) DELIVERED MESSAGE: 5:18Shum-PC:22031
14266 INFO (LazyFB) 4 - gossip request 58Shum-PC:22035 for : 18
14267 INFO (LazyFB) SERVED
16160 INFO (LazyFB) >Delivery & ttl now: 60.0
16163 DEBUG (Application2) DELIVERED MESSAGE: 6:18Shum-PC:22031
17251 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
17251 INFO (LazyFB) DONT HAVE & TTL=0
19157 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031|7
19157 INFO (LazyFB) >Delivery & ttl now: 66.66666666666666
19158 DEBUG (Application2) DELIVERED MESSAGE: 7:18Shum-PC:22031
23205 INFO (LazyFB) 4 - gossip request 58Shum-PC:22035 for : 18
23205 INFO (LazyFB) SERVED
25181 INFO (LazyFB) >Delivery & ttl now: 71.42857142857143
25183 INFO (LazyFB) 3 - start gossip to:
25183 INFO (LazyFB) :28Shum-PC:22032
25183 INFO (LazyFB) :68Shum-PC:22036
25186 INFO (LazyFB) :48Shum-PC:22034
25188 DEBUG (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
29249 INFO (LazyFB) 4 - gossip request 68Shum-PC:22036 for : 18
29249 INFO (LazyFB) DONT HAVE & TTL=0

Process 4 - 51
Terminal Process
17253 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
17253 INFO (LazyFB) DONT HAVE & TTL=0
17255 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
17255 INFO (LazyFB) DONT HAVE & TTL=0
22183 INFO (LazyFB) 2 - Storing msg: 18Shum-PC:22031|8
22183 INFO (LazyFB) >Delivery & ttl now: 100.0
22199 INFO (LazyFB) 3 - start gossip to:
22200 INFO (LazyFB) :28Shum-PC:22032
22201 INFO (LazyFB) :18Shum-PC:22031
22201 INFO (LazyFB) :68Shum-PC:22036
22202 INFO (LazyFB) 3 - start gossip to:
22202 INFO (LazyFB) :18Shum-PC:22031
22202 INFO (LazyFB) :68Shum-PC:22036
22202 INFO (LazyFB) :58Shum-PC:22035
22203 DEBUG (Application2) DELIVERED MESSAGE: 8:18Shum-PC:22031
23180 INFO (LazyFB) 4 - gossip request 58Shum-PC:22035 for : 18
23180 INFO (LazyFB) DONT HAVE & TTL=0
25174 INFO (LazyFB) >Delivery & ttl now: 75.0
25177 DEBUG (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
26206 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
26207 INFO (LazyFB) SERVED
29238 INFO (LazyFB) 4 - gossip request 68Shum-PC:22036 for : 18
29239 INFO (LazyFB) SERVED

Process 5 - 51
Terminal Process
19046 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
19046 INFO (LazyFB) DONT HAVE & TTL=0
23935 INFO (LazyFB) >Delivery & ttl now: 60.0
23937 INFO (LazyFB) 3 - start gossip to:
23937 INFO (LazyFB) :18Shum-PC:22033
23937 INFO (LazyFB) :68Shum-PC:22036
23937 INFO (LazyFB) :48Shum-PC:22034
23940 INFO (LazyFB) 3 - start gossip to:
23941 INFO (LazyFB) :38Shum-PC:22033
23941 INFO (LazyFB) :18Shum-PC:22031
23941 INFO (LazyFB) :68Shum-PC:22036
23943 DEBUG (Application2) DELIVERED MESSAGE: 8:18Shum-PC:22031
25970 INFO (LazyFB) 5 - gossip response : 38Shum-PC:22033 for :
25971 INFO (LazyFB) >Delivery & ttl now: 62.5
25971 DEBUG (Application2) DELIVERED MESSAGE: 7:18Shum-PC:22031
29942 INFO (LazyFB) >Delivery & ttl now: 62.5
29944 INFO (LazyFB) 3 - start gossip to:
29951 INFO (LazyFB) :18Shum-PC:22032
29951 INFO (LazyFB) :18Shum-PC:22031
29953 INFO (LazyFB) :68Shum-PC:22036
29958 DEBUG (Application2) DELIVERED MESSAGE: 10:18Shum-PC:22031
31013 INFO (LazyFB) 4 - gossip request 68Shum-PC:22036 for : 18
31013 INFO (LazyFB) DONT HAVE & TTL=0

Process 6 - 51
Terminal Process
26798 INFO (LazyFB) DONT HAVE & TTL=0
26813 INFO (LazyFB) 4 - gossip request 28Shum-PC:22032 for : 18
26813 INFO (LazyFB) DONT HAVE & TTL=0
26780 INFO (LazyFB) >Delivery & ttl now: 83.33333333333334
26785 INFO (LazyFB) 3 - start gossip to:
26792 INFO (LazyFB) :18Shum-PC:22031
26796 INFO (LazyFB) :58Shum-PC:22035
26796 INFO (LazyFB) :48Shum-PC:22034
26797 INFO (LazyFB) 3 - start gossip to:
26797 INFO (LazyFB) :28Shum-PC:22032
26797 INFO (LazyFB) :18Shum-PC:22031
26797 INFO (LazyFB) :48Shum-PC:22034
26798 INFO (LazyFB) 3 - start gossip to:
26798 INFO (LazyFB) :38Shum-PC:22033
26798 INFO (LazyFB) :18Shum-PC:22031
26798 INFO (LazyFB) :58Shum-PC:22035
26801 DEBUG (Application2) DELIVERED MESSAGE: 10:18Shum-PC:22031
29800 INFO (LazyFB) 4 - gossip request 58Shum-PC:22035 for : 18
29800 INFO (LazyFB) DONT HAVE & TTL=0
30852 INFO (LazyFB) 5 - gossip response : 18Shum-PC:22031 for :
30852 INFO (LazyFB) >Delivery & ttl now: 70.0
30852 DEBUG (Application2) DELIVERED MESSAGE: 9:18Shum-PC:22031
30855 INFO (LazyFB) 5 - gossip response : 48Shum-PC:22034 for :
```

Results for nodes are:

Node	Delivery rate
1	100%
2	44%
3	66%
4	75%
5	62%
6	70%

Node 1 has high delivery rate because it sends broadcast messages and delivers all of them so we can exclude it from comparison. We can see that comparing to previous test chances to fail gossip increased because some messages were not stored.

5(D)

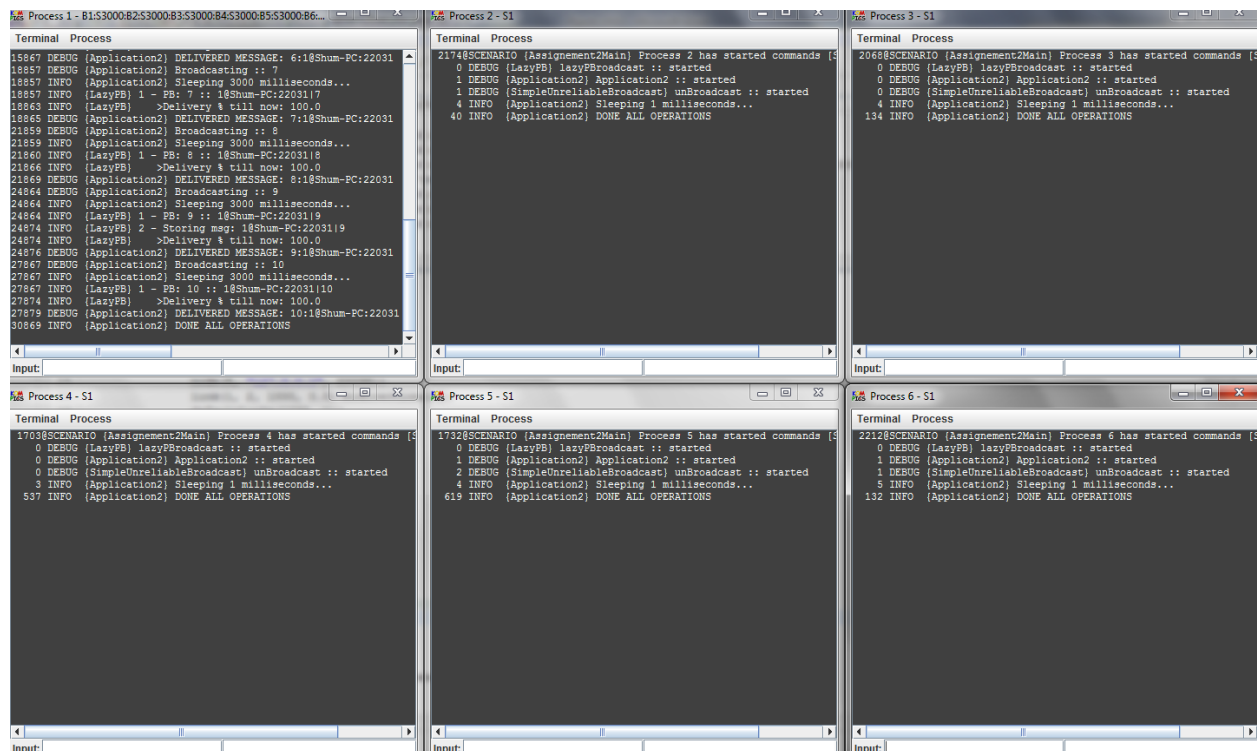
A broadcasted message, after being delivered by some node(s) and missed by a node p, is completely lost such that p can never retrieve it through gossiping.

We can guarantee that message will be never retrieved through gossiping by setting to 0 configuration parameters but according to PDF we can't do that.

Another way to create such execution is to set network lost to 100%

```
defaultLinks(1000, 1);
```

In this case only sending node delivers the message and all other nodes p do not deliver it.



```
Process 1 - 81:53000:82:53000:83:53000:84:53000:85:53000:86:53000
Terminal Process
18867 DEBUG (Application2) DELIVERED MESSAGE: 6:1@Shum-PC:22031
18867 DEBUG (Application2) Broadcasting :: 7
18867 INFO (Application2) Sleeping 3000 milliseconds...
18867 INFO (LazyPB) 1 - PB: 7 :: 1@Shum-PC:22031/7
18867 INFO (LazyPB) >Delivery & till now: 100.0
18867 DEBUG (Application2) DELIVERED MESSAGE: 7:1@Shum-PC:22031
18867 DEBUG (Application2) Broadcasting :: 8
18867 INFO (Application2) Sleeping 3000 milliseconds...
18867 INFO (LazyPB) 1 - PB: 8 :: 1@Shum-PC:22031/8
18867 INFO (LazyPB) >Delivery & till now: 100.0
18867 DEBUG (Application2) DELIVERED MESSAGE: 8:1@Shum-PC:22031
18867 DEBUG (Application2) Broadcasting :: 9
18867 INFO (Application2) Sleeping 3000 milliseconds...
18867 INFO (LazyPB) 1 - PB: 9 :: 1@Shum-PC:22031/9
18867 INFO (LazyPB) >Delivery & till now: 100.0
18867 DEBUG (Application2) DELIVERED MESSAGE: 9:1@Shum-PC:22031
18867 DEBUG (Application2) Broadcasting :: 10
18867 INFO (Application2) Sleeping 3000 milliseconds...
18867 INFO (LazyPB) 1 - PB: 10 :: 1@Shum-PC:22031/10
18867 INFO (LazyPB) >Delivery & till now: 100.0
18867 DEBUG (Application2) DELIVERED MESSAGE: 10:1@Shum-PC:22031
18867 INFO (Application2) DONE ALL OPERATIONS

Process 2 - S1
Terminal Process
21746SCENARIO (Assignment2Main) Process 2 has started commands [5
0 DEBUG (LazyPB) lazyPBroadcast :: started
1 DEBUG (Application2) Application2 :: started
1 DEBUG (SimpleUnreliableBroadcast) unBroadcast :: started
4 INFO (Application2) Sleeping 1 milliseconds...
40 INFO (Application2) DONE ALL OPERATIONS

Process 3 - S1
Terminal Process
20688SCENARIO (Assignment2Main) Process 3 has started commands [5
0 DEBUG (LazyPB) lazyPBroadcast :: started
0 DEBUG (Application2) Application2 :: started
0 DEBUG (SimpleUnreliableBroadcast) unBroadcast :: started
4 INFO (Application2) Sleeping 1 milliseconds...
134 INFO (Application2) DONE ALL OPERATIONS

Process 4 - S1
Terminal Process
1703SCENARIO (Assignment2Main) Process 4 has started commands [5
0 DEBUG (LazyPB) lazyPBroadcast :: started
0 DEBUG (Application2) Application2 :: started
0 DEBUG (SimpleUnreliableBroadcast) unBroadcast :: started
3 INFO (Application2) Sleeping 1 milliseconds...
537 INFO (Application2) DONE ALL OPERATIONS

Process 5 - S1
Terminal Process
1732SCENARIO (Assignment2Main) Process 5 has started commands [5
0 DEBUG (LazyPB) lazyPBroadcast :: started
1 DEBUG (Application2) Application2 :: started
2 DEBUG (SimpleUnreliableBroadcast) unBroadcast :: started
4 INFO (Application2) Sleeping 1 milliseconds...
619 INFO (Application2) DONE ALL OPERATIONS

Process 6 - S1
Terminal Process
22128SCENARIO (Assignment2Main) Process 6 has started commands [5
0 DEBUG (LazyPB) lazyPBroadcast :: started
1 DEBUG (Application2) Application2 :: started
1 DEBUG (SimpleUnreliableBroadcast) unBroadcast :: started
5 INFO (Application2) Sleeping 1 milliseconds...
132 INFO (Application2) DONE ALL OPERATIONS
```

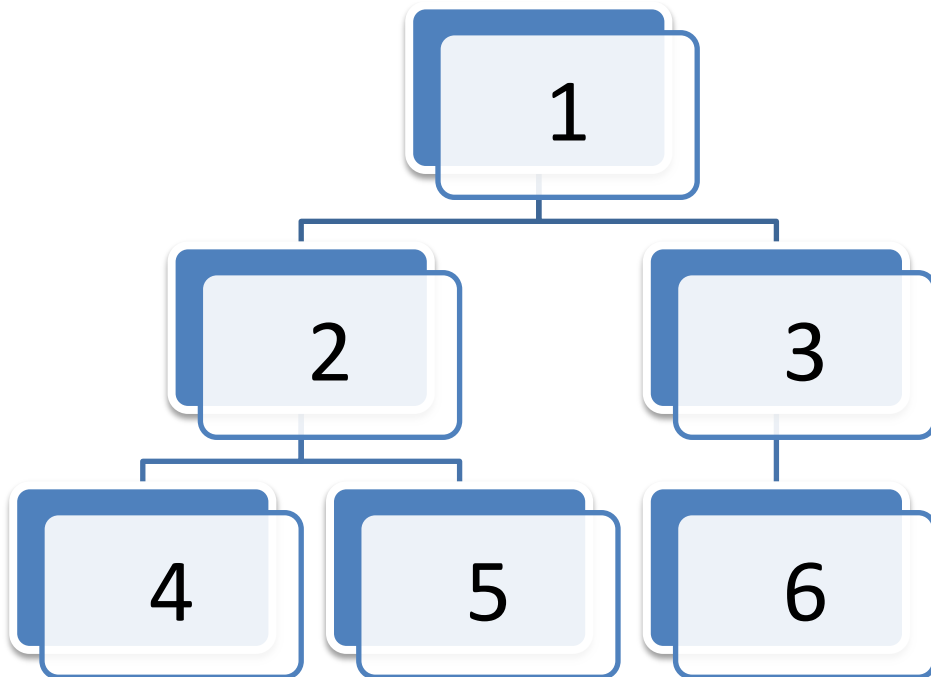
The first node delivers 100% messages and all others deliver nothing.

EXERCISE 2

QUESTION 1

Do you think the lazy probabilistic broadcast algorithm will work in such a topology? What implications will the afore-mentioned topology have on the algorithm?

No, the lazy probabilistic broadcast will not work in not fully connected topology.



The problem is that unreliable broadcast described in Algorithm 1 sends messages only to neighbors so if node 1 broadcasts something it is delivered only by nodes 2 and 3. Nodes 4, 5, and 6 never deliver broadcast messages from 1 and they do not know if they missed some message from node 1 or node 1 never send any messages so they cant start gossip.