ID2203 Tutorial 2 - Broadcast

Cosmin Arad Tallat M. Shafaat
icarad(@)kth.se tallat(@)kth.se

Handout: February 12, 2010
February 19, 2010

Introduction

The goal of this tutorial is to get familiar with the broadcast abstraction used in distributed systems.

Exercise 1

Consider the Lazy Probabilistic Broadcast given in section 3.8.5 of the course text book. The algorithm has some errors, some of which have been informally specified below:

- The *deliver-pending* method is called in the event $flp2pDeliver$. This means that when the gossip phase of the algorithm is active, the algorithm will progress only when it receives a missed message m such that $sn_m = delivered[s_m] + 1$. Since the algorithm is probabilistic, it might be that the node cannot recover the message m, even in the gossip phase. In such a case, the progress of the algorithm is blocked.
- On event Timeout, the case is not handled where a node has still not received all the missed messages, i.e. $sn > delivered[s] + 1$.

In this exercise, you have to do the following:

1. Try to find any other errors.
2. Correct the algorithm. Since the algorithm is probabilistic, your correction should increase the probability of a message to be delivered compared to the algorithm in the book.

3. Provide your new algorithm in your report along with discussion on the errors fixed.

4. Implement your algorithm in Kompics. To implement Unreliable Broadcast that lazy probabilistic broadcast uses, you can implement a simple broadcast algorithm such as Algorithm 1.

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.

   (a) No message is lost (all nodes deliver a broadcasted message m).

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

   (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.

   (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.

   For all of the above executions, use non-zero values for the parameters.

Exercise 2

The lazy probabilistic broadcast given in section 3.8.5 of the text book assumes that the topology is fully connected, i.e. there is a link between all pairs of nodes. In reality, a node may not always know about all the nodes in the system. Consider a case where the topology is connected but not fully connected, i.e. there is a *path* between any pair of nodes but not a link between all pairs of nodes. Answer the following in your report.

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

Algorithm 1 Unreliable Broadcast

Implements:

UnreliableBroadcast (un).

Uses:

FairLossPointToPointLinks (flp2p).

1: **upon event** $\langle\ unBroadcast \mid m\ \rangle$ **do**
2: **for all** $p_i \in \Pi$ **do**
3: **trigger** $\langle flp2pSend \mid p_i, m\ \rangle$;
4: **end for**
5: **end event**

6: **upon event** $\langle\ flp2pDeliver \mid p_i, m\ \rangle$ **do**
7: **trigger** $\langle unDeliver \mid p_i, m\ \rangle$;
8: **end event**

2. Can you devise an alternative implementation of Algorithm 1 which would make LazyPB work correctly for such a topology? Explain your answer.