5. Übung, SS 2013
Verteilte Systeme/ Distributed Systems
Ausgegeben am 22.6.2013
Vorstellung 4.7.2013

Katinka Wolter
Benjamin Güldenring
Inst. für Informatik
Freie Universität Berlin

Assignment 1. Logical Clocks

Assume processes exchange messages as shown in Figure 1. Add a new message, that is

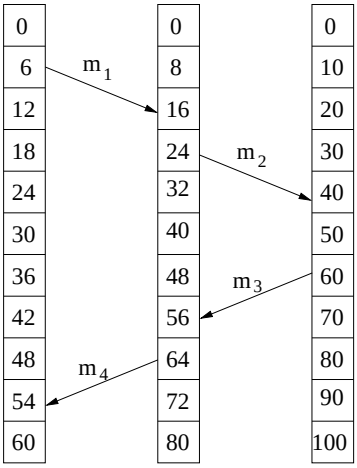

Figure 1: Three processes with each its own clock

concurrent with m_1, that is, it neither happens before m_1 nor after it.

Assignment 2. Correctness and Complexity of Leader Election algorithms

We want to verify stated properties of the leader election algorithms.

1. Prove formally, e.g. by induction, that the LCR algorithm is correct, i.e. after at most n rounds the largest ID is found and the status message sent is `LEADER`.
2. Show that the number of messages sent by the HS is at most $8n(1 + \lceil \log n \rceil)$, which means it is $O(n \log(n))$.

Assignment 3. Peersim - Leader Election

3.1 Implementation

Implement the (simplified, v.i.) *leader election protocol for wireless environments* as discussed in the lecture. **Note**: you will also find a description of the protocol in section 6.5.2 (Elections in Wireless Environments) of [1].

Your implementation should meet the following simplifications and demands:

- Use an event based simulation.
- One single (randomly chosen) node starts the election (i.e. no concurrent elections).
- The simulation ends when a best leader is found (you may omit the final broadcast).
- Each node has a `capacity` value. The best leader is found with respect to this value. A bigger value is considered to be better.

3.2 Evaluation

1. Recall that this is a protocol for wireless environments. Name at least two metrics you think are useful to evaluate this protocol. Which parameters may be interesting to vary for evaluation?
2. Pick one of those metrics and implement the necessary observer classes if necessary. You may reuse all classes peersim already provides in its source code or the examples.
3. Evaluate the protocol based on your metric using the tools we discussed in the previous peersim assignments (gnuplot, awk). The configuration of your experiments should meet the following requirements:
   - Nodes shall be initialized by a random capacity value. You may use the `LinearDistribution` or `PeakDistributionInitializer` initializers.
   - Use the `IdleProtocol` and the `WireKOut` initializers to create the initial network.

Assignment 3. References

[1] Andrew S. Tanenbaum and Steen van Maarten, Distributed Systems: Principles and Paradigms (2nd Edition) 2006, ISBN 0132392275