Causality and Pseudo-Code for Distributed Systems (2nd practical session)

This practical session contains exercises to assess your understanding in:

- **Causality in Distributed Systems;**
- **Lamport clocks and vector clocks;**
- **Writing pseudo code for distributed algorithms.**

1. Give informal definitions of *liveness* and *safety*. What is the relation of these two properties with *correctness* in the specification of a system?

2. What is causality in distributed systems?

3. Explain with your own words:
   - What are Lamport clocks?
   - What are vector clocks?
   - What are the differences between Lamport clocks and vector clocks?

4. The Event-based Component Model (ECM) relies on *requests* and *indications*. Explain both concepts. What is the difference between them ?

5. Follow the ECM to give an implementation of vector clocks.

Best effort broadcast with Controlled Flooding

The goal of *best-effort global broadcast* (hereafter referred simply as broadcast) is to send a message to all nodes in the network. This form of dissemination is not necessary *reliable*. Nodes *may* forward a message using local broadcast upon its first reception. We denote such nodes as *relay* nodes.

Controlled flooding is an implementation of broadcast that relies on a simple counter-based approach, which is efficient in a variety of practical situations; Figure 1 shows an example of this implementation.

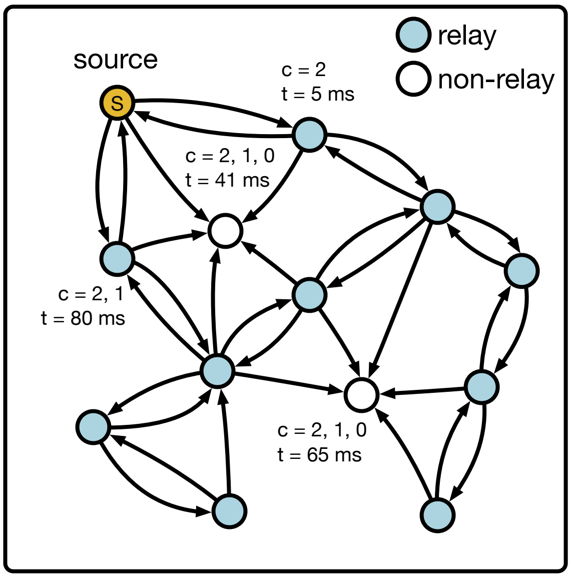

Figure 1: Example of best effort broadcast using controlled flooding.

Controlled flooding works as follows:

- Upon the first reception of a message, a node sets up a timer t and a counter c;
- Every reception of a duplicate decreases c by one;
- When t expires, the node only acts as a relay if $c > 0$.

Exercise. Follow the ECM to give the specification of controlled flooding.