Modeling Distributed Programming

This first practical session contains several questions, and exercises, in order to assess your understanding in the: *Basics on Modeling Distributed Systems*

1. Why is it useful to use a model for characterizing a distributed system? Give at least two arguments.

2. Say whether the following statements are **True**, **False** or **Cannot say**:

   - *Achieving cooperation towards a consistent state is one of the main difficulties in distributed programming* ;
   - *Peer-to-Peer systems require a central entity to make them work* ;
   - *In a system of N nodes, atomic commitment for such system is still achievable if only one node experience a crash.*

3. Sketch a proof of how to obtain an agreement algorithm from atomic broadcast and vice-versa. What you might conclude with that proof?

4. Explain the following terms and give one use-case where it might be useful.

   - **Best-effort Broadcast**

   - **Reliable Broadcast**

   - **Total-order Broadcast**

5. An example of a partially asynchronous system is a wired network where there exist an upper-bound in the latency of messages U_{latency} among links between nodes. Considering that every node in such network contains an implementation of an agreement algorithm, sketch a mechanism to compute/approximate U_{latency} in the network.

6. Define, in a formal way, the transition function f in the model of distributed systems seen in class.