Distributed algorithms

Pierre Colson

mercredi 05 janvier

Contents

Markdown verion on *github*

General

- The distributed system is made of a finite set of **processes** : each process models a **sequencial** program
- Every pair of processes is connected by a **link** through which the processes exchange **messages**
- **Safety** is a property which states that nothing bad should happen
- **Liveness** is a property which states that something good should happen
- Twos kinds of failures are mainly considered
  - **Omissions** : The process omits to send messages it is supposed to send
  - **Arbitrary** : The process sends messages it is not supposed to send
- A **correct** process is a process that does not fail (that does not crash)
- **Fair-loss links**
  - **FL1. Fai-loss** : If a message is sent infinitely often by p_i to p_jn and neither p_i or p_j crashes then m is delivered infinitely often by p_j
  - **FL2. Finite duplication** : If a message m is sent a finite number if times by p_i to p_j, m is delivred a finite number of times by p_j
  - **FL3. No creation** : No message is delivered unless it was sent
- **Stubborn links**
  - **SL1. Stubborn delivery** : If a process p_i sends a message m to a correct process p_j, and p_i does not crash, then p_j delivers m an infinite number of times
  - **SL2. No creation** : No message is delivered unless it was sent

```
Implements: StubbornLinks (sp2p)
Uses : FairLossLinks (flp2p)
upon event <sp2pSend, dest, m> do
  while (true) do
    trigger <flp2pSend, dest, m>
upon event <flp2pDeliver, src, m> do
  trigger <sp2pDeliver, src, m>
```