SR (Systèmes Répartis)

Unit 2: Basic mechanisms and properties: Synchrony, Asynchrony, Reliable Channels

François Taïani

Distributed Algorithms

- Distributed Algorithms look at
  - ➔ fundamental problems of **distributed coordination**
  - ➔ for instance: agreement, mutual exclusion, leader election...
  - ➔ in an **abstract way** (abstract model of reality)
- Sometimes assuming some **adverse conditions**
  - ➔ participants may behave somewhat erratically
  - ➔ messages may get lost
- Goal of the study of distributed algorithms
  - ➔ find out **whether something is possible** under which conditions
  - ➔ for solvable problems, **prove** that a particular solution works
  - ➔ **compare** correct solutions to the same problems

Goal of this session

- A first contact with **Distributed Algorithms**
- Explore some fundamental aspects of distributed systems
  - ➔ distinguish between different kind of distributed systems
  - ➔ an example of what can and cannot be done
- 3 problems presented as metaphors
  - ➔ The cursed monastery
  - ➔ The royal wedding
  - ➔ The 2 generals
- We will discuss each of them, try to find solutions and see what this teaches us on distributed systems

Group Solving Session

The Cursed Monastery

- A visitor comes to a remote monastery and announces:
  - " *Some of the monks have been cursed by the local wizard and marked by a point on their forehead. They must all leave the monastery, or the whole community will perish.* "
- This monastery obeys a very strict rule
  - ➔ There are not mirrors in the monastery
  - ➔ Monks do not communicate in any ways
  - ➔ They only meet once a day for dinner

- The visitor makes his announcement at diner
- How many days does it take for all the cursed monks to leave the monastery and why?
  - ➔ Hint: the monks have studied distributed algorithms

The Royal Wedding

- A king would like to marry his son to the princess of a neighbouring kingdom
- By tradition, if the alliance is agreed, the wedding will take place in a remote monastery, on the border between the two kingdoms
- It is all right if the parties do not arrive at the same time at the monastery
- Messengers travel by horses, and may get lost to thugs
  - ➔ however they have a non-zero chance of getting through
- Design an algorithm that allows the wedding to take place if both parties agree (if not, nothing should happen)

The 2 Generals

- 2 allied generals have surrounded their common enemy
- Their camps are 1 day apart by horse from each other
- They want to agree on when to attack
- Each can send the other one only one message per day
- Messengers might get attacked by thugs and get lost
- Design an algorithm for the 2 generals to reach an agreement and attack simultaneously

What have we learnt?

- Whether you know or not **how long your messages will take** makes a huge difference
  - ➔ No bound on communication delays: **asynchronous** systems
  - ➔ Bounded communication delays: **synchronous** systems
- With bounded delays + global clock (monastery)
  - ➔ **Not doing something** can mean a lot
- Some problems have **no solution**
  - ➔ timely coordination with lossy channels impossible (the generals)
- If **communication** channels are **faulty**
  - ➔ possible to **make** them **perfect** (the royal wedding)
  - ➔ but a **price** to pay: communication delays can get **arbitrary long**
  - ➔ this is how **Ethernet** & **TCP/IP** work
  - ➔ does **not** work for **real time** systems