Foundations of Distributed Systems

José Orlando Pereira

HASLab / Departamento de Informática
Universidade do Minho

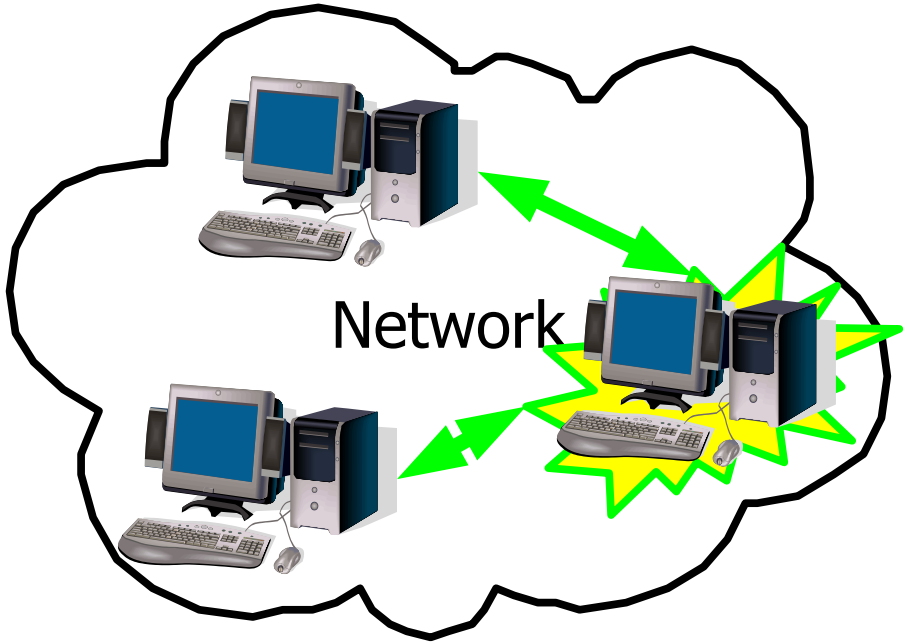
2019/2020

Motivation

- What is truly unique about distributed systems?

Case study: Leader election

- Select a unique leader in a distributed system
- Useful for:
  - Coordination
  - Efficiency
  - …

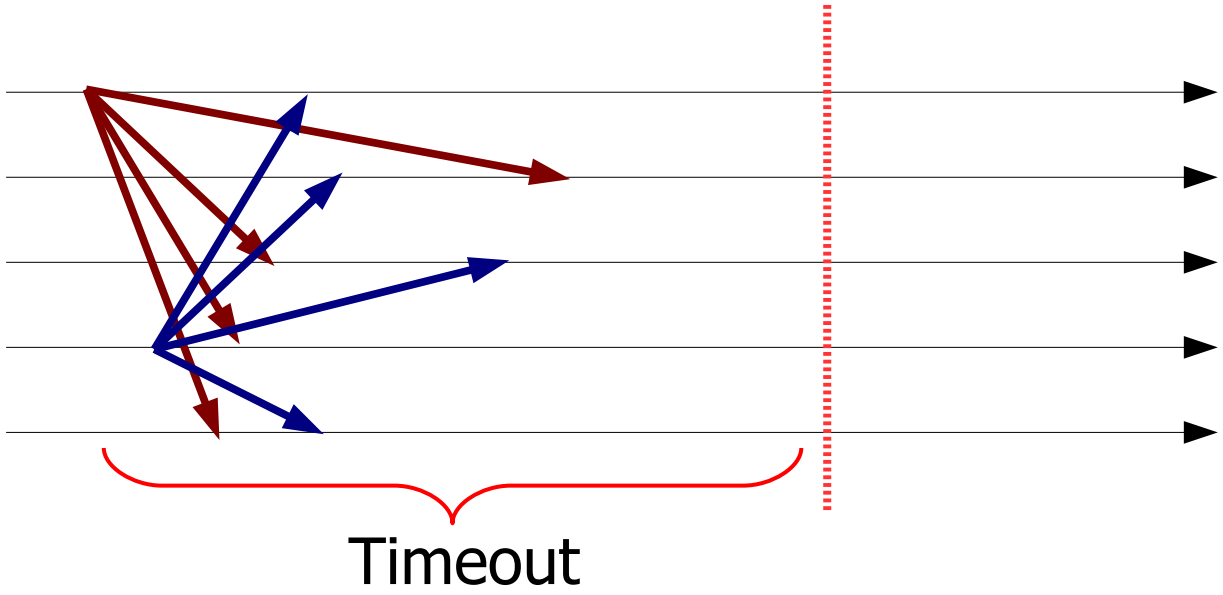


Abstraction

- No sockets/clients/servers/ byte buffers/threads/...
- Reasoning in terms of:
  - *n* processes: sequences of discrete computation events
  - n^2 channels: connect send/receive events in pairs of processes

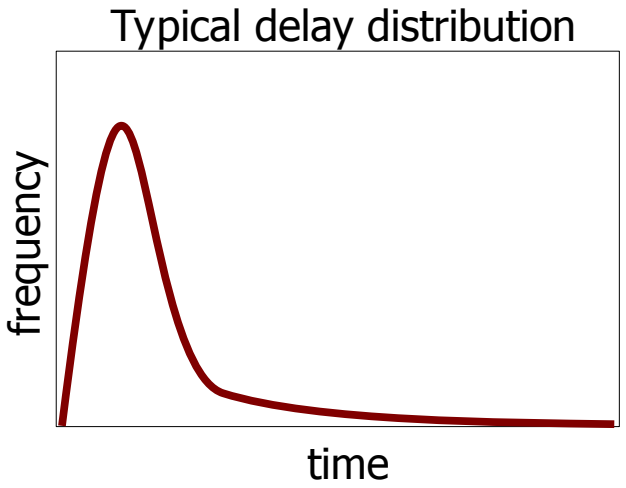
Case study: Simple algorithm

- Each process trying to be the leader sends its network address to all others
- Each process considers the process with the highest address to be the leader



In practice

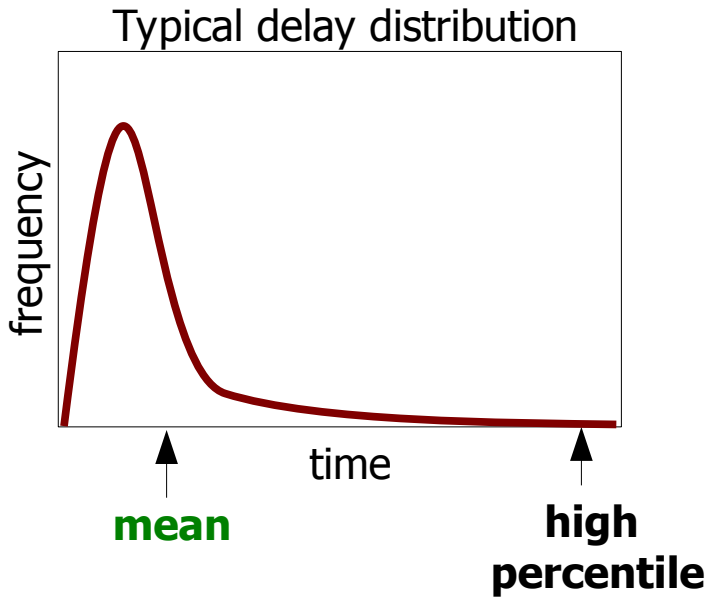
- Tight timeouts are dangerous:
  - E.g., proportional to mean delay
  - Means low coverage



- Large timeouts are not useful:
  - E.g., proportional to high percentile
  - Taking advantage of time causes a very large performance penalty

In practice

- Solutions that do not use time might have better performance:
  - Run time proportional to mean delay
  - Even if more message exchanges are necessary



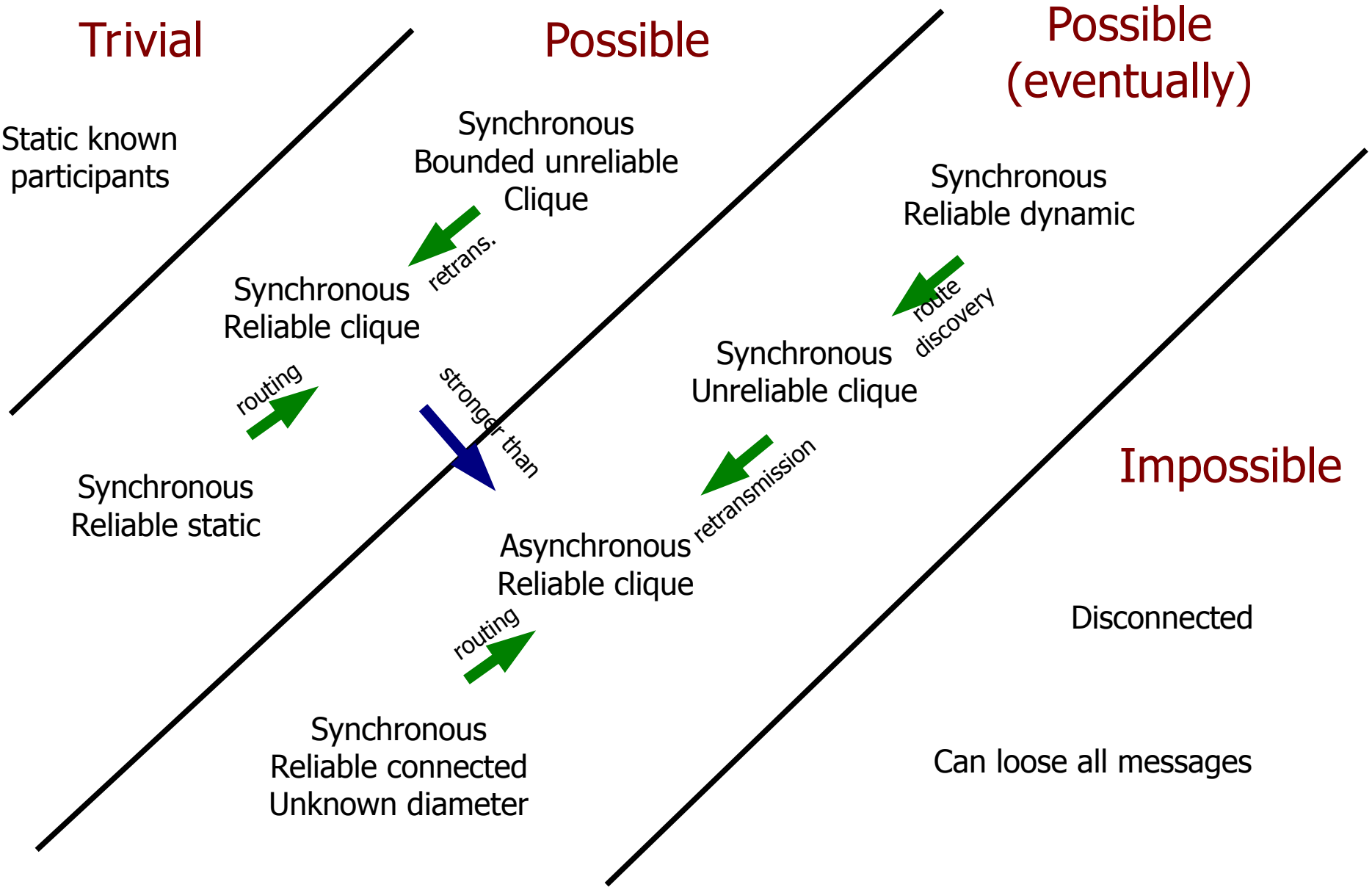
Asynchronous system model

- Assume no global time reference
- Assume no bounds on:
  - clock drift
  - processing time
  - message passing time
- Can we still solve the problem?

Example: Approach

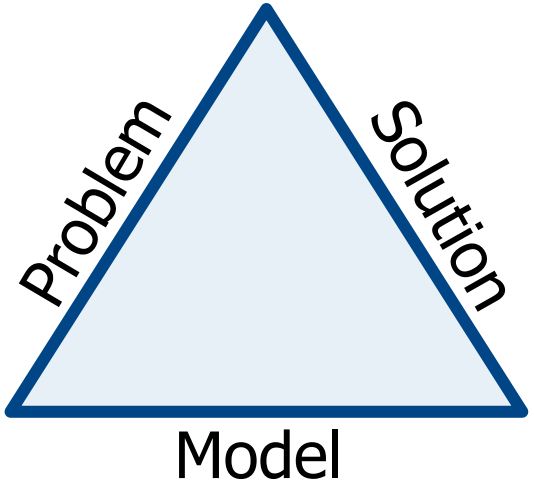
- Start with a synchronous reliable fully connected network
- Relax the system model:
  - Unbounded message loss
  - Large/unknown graph diameter
  - Dynamic graph
- Example: Leader election

Example: Leader election



Summary

- A system model is a set of assumptions:
  - "what we believe about the world"
- A system model is an abstraction
- An algorithm solves a problem in a system model:



Summary

- Asynchronous system model abstracts:
  - Heterogeneity
  - Dynamics
  - **Uncertainty**
- Much simpler than handling them explicitly
- Leads to widely applicable solutions

Summary

- What is truly unique about distributed systems?
  - **Uncertainty!**
- Challenge:
  - Perform a computation in spite of uncertainty