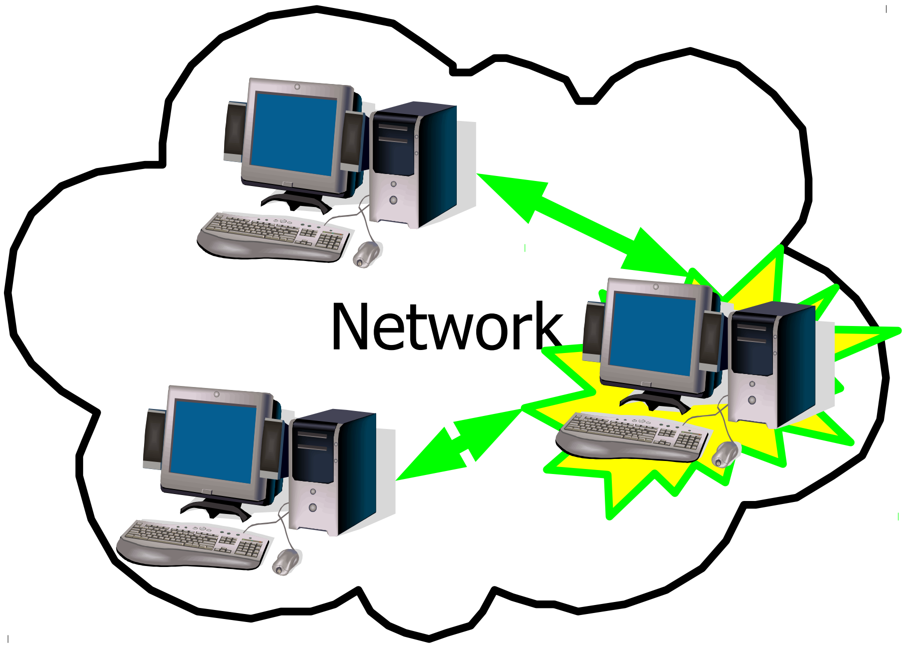
Example: Leader election

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

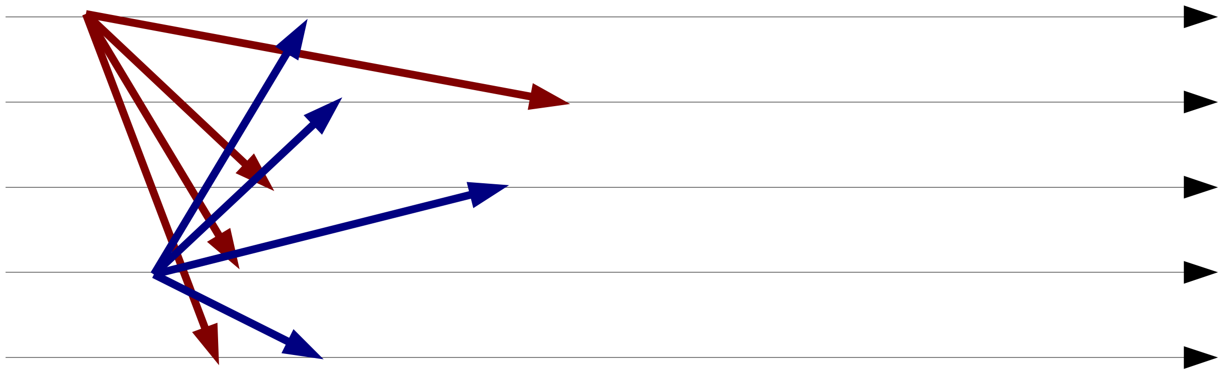


Properties

- No two processes disagree about the leader
- Every process will eventually select a leader

Simple algorithm

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



Choosing a specification

- What variables should we use to observe this system?
- What behaviors can be observed?

- Some depend on the problem being solved
  - Algorithm
- Some depend on what we assume true in the environment:
  - System model

Processing

- Set of processes:
  - Unknown
  - Known size
  - Known ids
- Process faults:
  - Crash
  - Crash and recovery
  - Misbehave arbitrarily

Communication

- Topology:
  - Fully connected (clique)
  - Connected graph:
    - Static or Dynamic
  - Partitions
    - Transient or Permanent
- Reliability:
  - From all messages lost to no messages lost
  - Can messages be created?

Time

- Synchronous:
  - Known upper bounds on message delays, processing delays, and clock differences
- Partially synchronous:
  - Unknown fixed bounds
  - Known eventual bounds
- Asynchronous:
  - No assumptions on bounds and clocks

Example: Leader election

Static known participants

Synchronous
Reliable static

Synchronous
Reliable dynamic

Synchronous
Reliable clique

Synchronous
Unreliable clique

Synchronous
Bounded unreliable
Clique

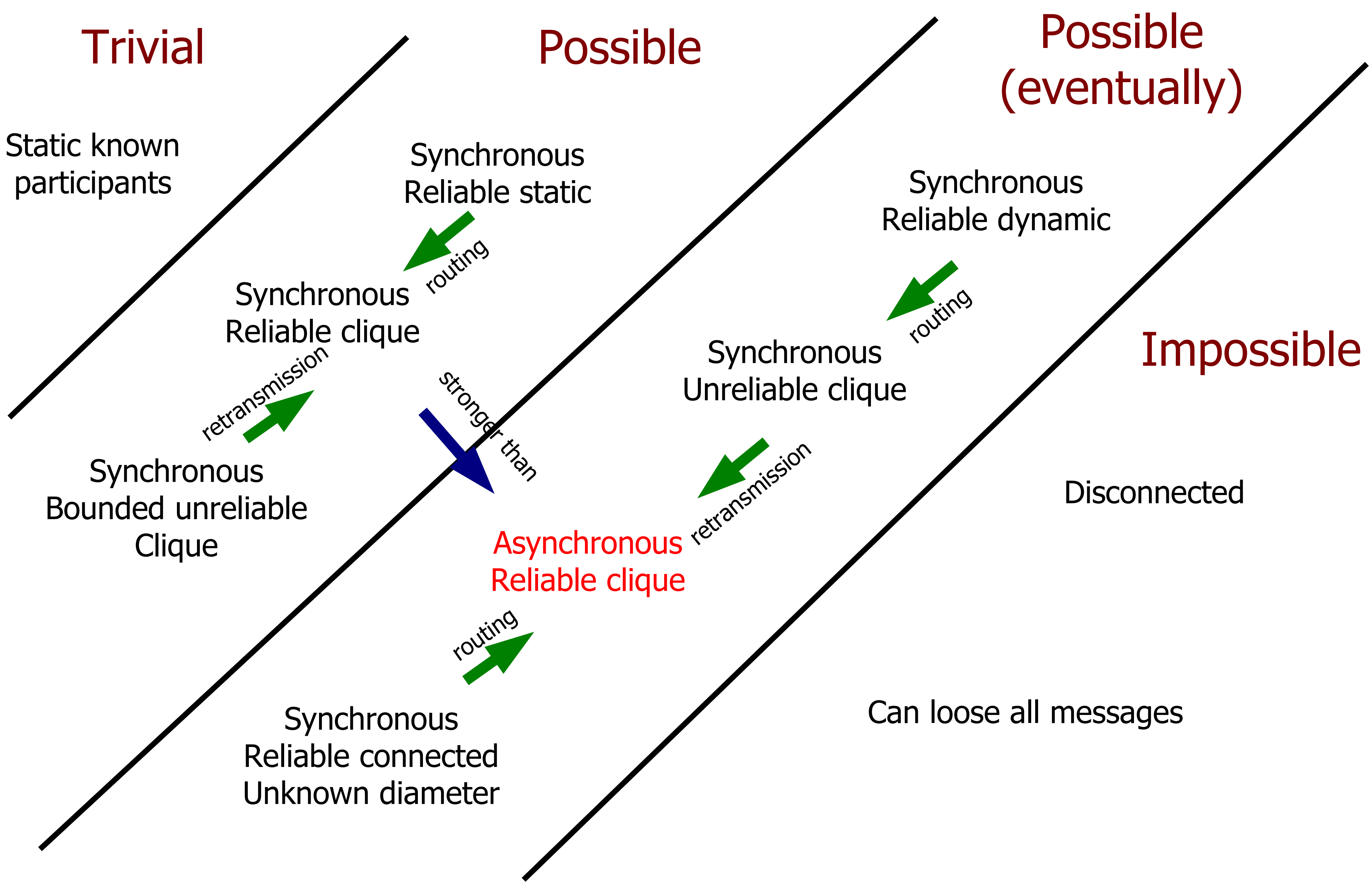
Disconnected

Asynchronous
Reliable clique

Synchronous
Reliable connected
Unknown diameter

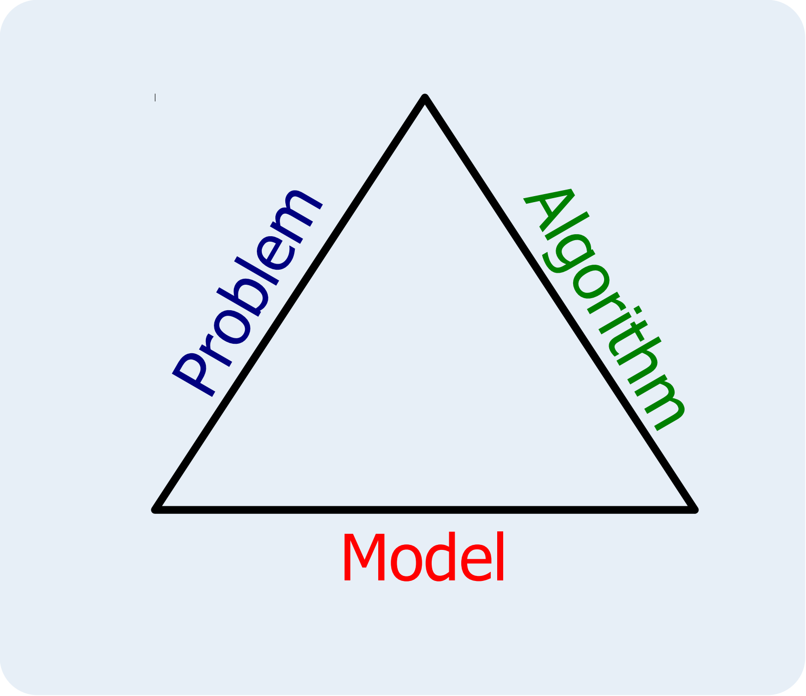
Can loose all messages

Example: Leader election

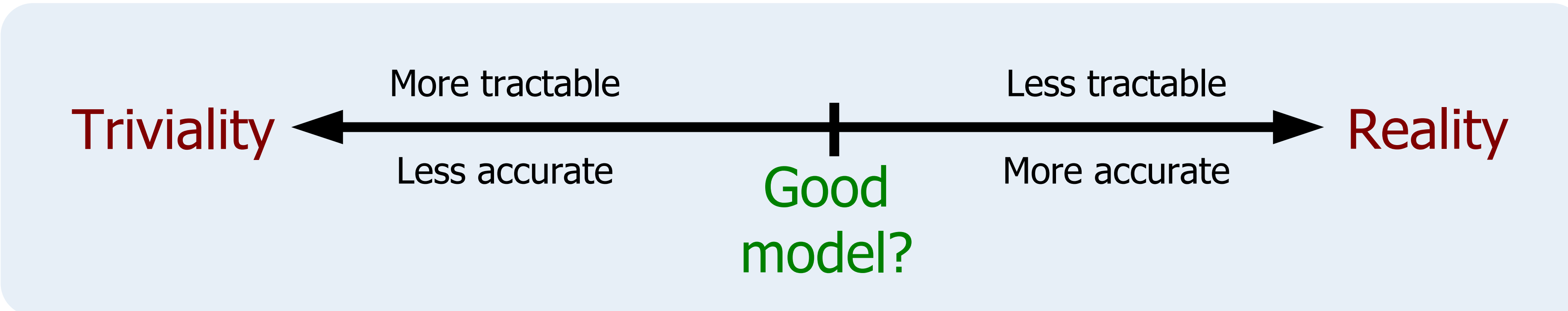


"What good are models?"

- Partition problems according to:
  - <u>Feasibility</u>: What classes of problems can be solved?
  - <u>Cost</u>: How expensive must the solution be?



“What models are good?”



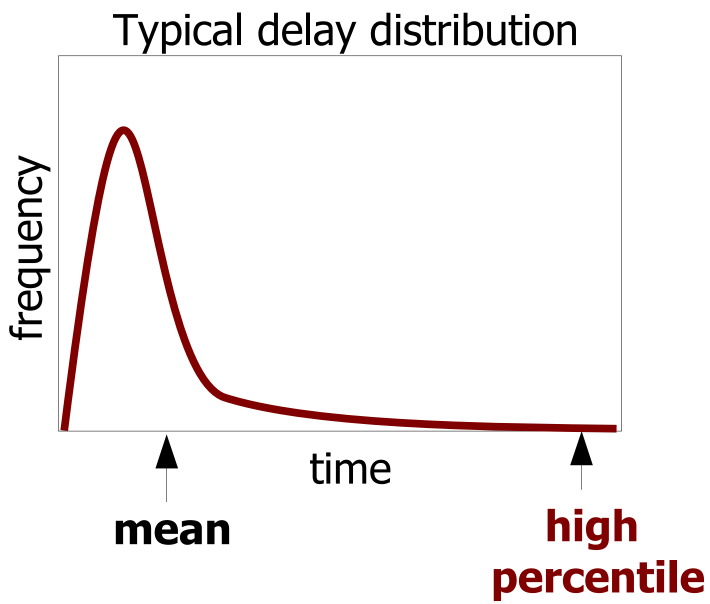
- Accurate:
  - Analysis of the model yields truths about the object
- Tractable:
  - Analysis of the model is possible

Models and engineering

- Coverage is the probability that reality exhibits only behaviors that are captured by the model
- Unfortunately, coverage < 1.0 in all interesting models (both tractable and accurate)

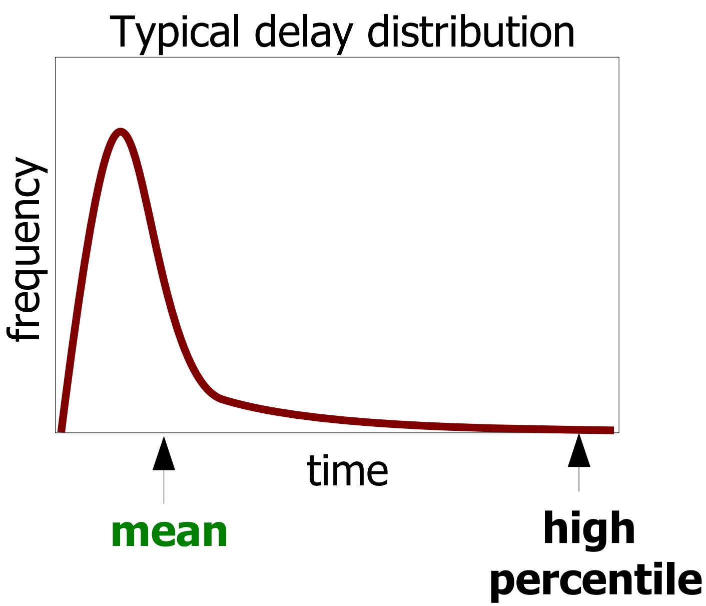
Model vs performance

- Tight synchronous limits are dangerous:
  - Timeouts proportional to mean delay
  - Low coverage or expensive systems
- Large synchronous limits are not useful:
  - Timeouts proportional to high percentile delay
  - Taking advantage of synchrony causes a very large performance penalty



Model vs performance

- Solutions for asynchronous systems might have better performance:
  - Progress after mean delay
  - Even if more message exchanges are necessary



Which models when?

- Weak models lead to general solutions
  (when solutions exist...)
- Weak models lead to expensive solutions
  (but a strong model has a cost too...)
- Considering multiple models simultaneously:
  - Layered abstractions
  - Bounds to real systems

Which models when?

- The universal model for distributed systems:
  - Asynchronous
  - Reliable message passing
  - Statically connected clique
- Provides solutions that are universally valid
- Strong enough to solve (most) problems