

Self-stabilizing Population Protocols

Hong Jiang

Theory Clique, Oct. 14, 2005

Adapted from OGST(Oct 4, 2005) slides

October 14, 2005

- 1 Introduction
 - Population Protocols
 - A simple example
 - Self-stabilization
- 2 Protocols
 - Token circulation
 - Distance-2 coloring
 - Leader election
- 3 Future research

Basic model of computation

- An ad hoc network of finitely many **anonymous** finite-state agents.
- Each agent receives an initial input which decides its initial state.
- Two agents interact when they are close enough.
- Each interaction rule is of the form: $(a, b) \mapsto (c, d)$, in which a , b , c , and d are states.
- Each agent has a current output decided by its current state.

Basic model (continued)

- A configuration C is a multiset of all agents' states. $C \rightarrow C'$ means C goes to C' after one transition.
- A configuration C is output-stable, if the multiset of outputs does not change in all configurations reachable from C .
- Let M be the multiset corresponding to the inputs, protocol A computes predicate P , if in every fair execution the system enters some output-stable configuration in which every agent outputs $P(M)$.

interaction graph

Complete Every pair of nodes have chance to interact.

General Two nodes can interact only if they are connected by an edge.

- Generally edges are directed. A transition of the form $(a, b) \mapsto$ can occur iff edge $a \rightarrow b$ exists.
- In an undirected interaction graph, each edge goes both ways.

Characteristics of population protocols

Memory constraint Each agent is a finite-state machine. (constant memory)

Strong anonymity Nodes do not have unique IDs, **and do not have innate ability to tell different nodes apart.**

Global fairness An infinite execution is fair if for all global configurations C and C' s.t. $C \rightarrow C'$, if C occurs infinitely often, so does C' .

A motivating example

- A flock of birds, each carrying a device that can measure the bird's body temperature.
- The device can signal other devices within a small distance.
- Each device wants to compute the parity of the number of feverish birds in the flock.
- Protocol: (2-bit state)
 - Input: normal $\mapsto (1, 0)$; high $\mapsto (1, 1)$
 - $((1, x), (1, y)) \rightarrow ((0, x \oplus y)(1, x \oplus y))$
 - $((1, x), (0, y)) \rightarrow ((1, x)(0, x))$
 - Output: $(x, y) \rightarrow y$
- Demo:
<http://www.cs.yale.edu/homes/jiang-hong/demo/parity.html>

Behaviors

- Not all useful tasks can be modeled as computation.
- We look at the following behavioral tasks:
 - Round-robin token circulation
 - Local addressing (distance-2 coloring)
 - Leader election in rings

Self-stabilization

Definition

A distributed system is **self-stabilizing** if it can start at any possible global configuration and regain consistency by itself without any external intervention.

- Regaining consistency means entering configurations that satisfy a set of safety conditions (such as the existence of a unique leader).
- A self-stabilizing system can tolerate **arbitrary transient faults**.

Token circulation

- Assumptions:
 - A unique leader exists. (We remove the assumption later.)
 - The ring is directed. (We can remove this too!)
- Requirement: A unique token circulates around the ring, without skipping any node.
- Protocol: Each node has a label $\in \{0, 1\}$; two node a and b do the following:
 - If b is leader, it sets its label to the complement of a 's label; Otherwise b copies a 's label.
 - If either node changes label, the token is passed from a to b . A new token is generated if a does not have one.
- Demo:

<http://www.cs.yale.edu/homes/jiang-hong/demo/token.html>

Distance-2 coloring

- Assumption: Known degree bound D
- Requirement: Any pair of nodes with 2-edges between them have different colors.
 - In other words, the neighbors of each node have different colors.
- Protocol sketch:
 - Each node has one bit corresponding to each color.
 - when a red node interacts with a blue node, the blue bit of the red node and the red bit of the blue node are flipped synchronously.
 - If the red node has another blue neighbor, its blue bit will eventually lose sync with one of the blue neighbor.
 - The blue node that detects the conflict changes its color.

- Demo:

<http://www.cs.yale.edu/homes/jiang-hong/demo/d2coloring.html>

Leader election in rings

- Requirement: Exactly one fixed node declares itself the leader, and all other nodes declare themselves nonleaders.
- Protocol sketch
 - k is any integer that does not divide the size of the ring.
 - Each node has a **label** and a **leader** mark, and can hold a **bullet** token and a **probe** token.
 - A maximal sequence of consecutive (modular k) labels is called a **segment**. (such as $0 \rightarrow 1 \rightarrow 2 \rightarrow 0 \rightarrow 1$ for $k = 3$)
 - An edge at the end of a segment is called a **barrier**. Every barrier moves counter-clockwise around the ring.
 - When 2 barriers collide, one of them is eliminated.
 - There exists at least one barrier, and eventually exactly one barrier.

Leader election in rings (cont.)

- A barrier sends out **probes** to the counter-clockwise direction.
- **Probes** are absorbed by any **leader** they run into.
- A **probe** is converted to a **leader** if it reaches a barrier.
- A **leader** fires **bullets** clockwise around the ring.
- **Bullets** are absorbed by any barrier they run into, and kill any **leader** along the way.

Demo:

http://www.cs.yale.edu/homes/jiang-hong/demo/leader_election.html

Future research

- Uniform self-stabilizing leader election protocols.
two known impossibility results:
 - impossible in rings, in models with local fairness.
 - impossible without constraint on topology, in population protocol model.
- Relation between different fairness conditions in distributed systems.