Self-stabilization and Sensor Networks

M Potop Butucaru, F Petit, S Tixeuil

UPMC Sorbonne Universités
prenom.nom@lip6.fr

Outline

Sensor Networks and Self-stabilization
- Model(s)
- Cached Sensornet
- Self-stabilizing Unison

TDMA
- Motivation
- Algorithm stack

Clustering
- Density
- Self-stabilizing Clustering
- Simulation Results

Conclusion

Sensor Networks

- processor + sensors + radio
- 2 AA batteries, on/off switch
- 3 LEDs for debugging

Sensor Networks

While (batteries supply power)

- Collect, aggregate and reduce data
- log into memory

In spite of numerous fault modes

- Permanent sensor failures, node failures
- restarts, radio failures
- transient faults, reconfigurations

Distributed Systems

Definition (Classical System, *a.k.a.* Non stabilizing)

Starting from a particular initial configuration, the system immediately exhibits correct behavior.

Definition (Self-stabilizing System)

Starting from any initial configuration, the system eventually reaches a configuration from with its behavior is correct.

Distributed Systems

Definition (Classical System, *a.k.a.* Non stabilizing)

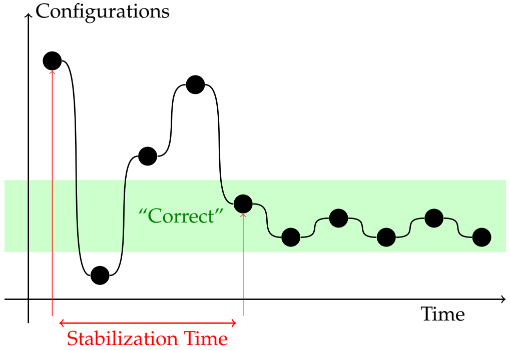
Starting from a particular initial configuration, the system immediately exhibits correct behavior.

Definition (Self-stabilizing System)

Starting from any initial configuration, the system eventually reaches a configuration from with its behavior is correct.

- Self-stabilization permits to recover from transient failures

Self-stabilization



Complexity Criteria

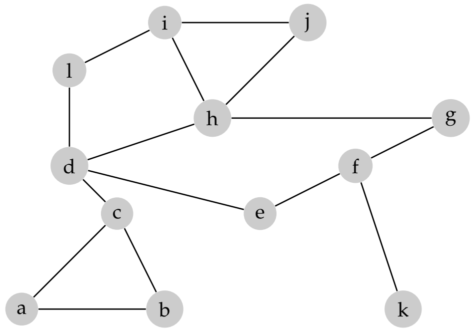
Maximize useful lifetime of system

- "maximise useful": correct quickly from illegitimate state
  - Self-stabilization, scalability
- "maximise lifetime": use minimal energy to preserve batteries
  - local vs. global preserving

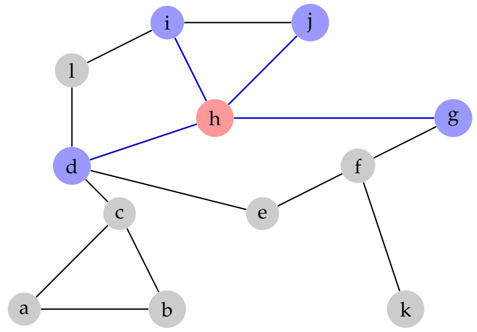
System Specifics

- only one radio frequency
- no collision detect
- access technique: CSMA/CA
- use CRC to detect collision
- no directional send/receive
- msg. are small (30 bytes)
- radio range about 1 meter
- number of neighbors < 10
- could be large number of nodes (perhaps > 100000)
- unique node IDs (probably)
- cost a few ¥(someday)
- slow processor (4 MHz)
- limited memory (4 KB RAM)
- item nodes have real-time clocks ≡ drift between 1 msec and 100 msec per second
- several power modes available

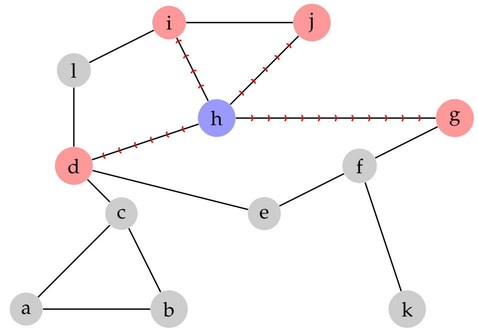
The Model(s)



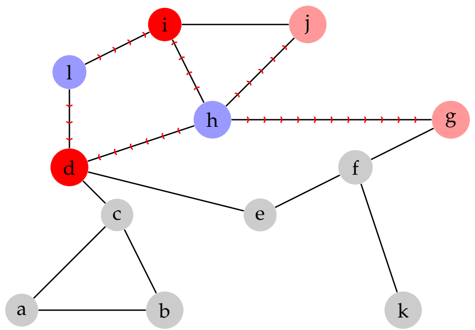
The Model(s)



The Model(s)



The Model(s)



The Model(s)

Self-stabilizing model

- Read neighborhood state,
- compute and update local state

Sensor Network model

- Read local state,
- compute and broadcast to neighborhood
- Collisions may appear

Self-stabilization in Sensor Networks

Transform (i.e. Simulate) the self-stabilizing model into the sensor networks model

- Pros: reuse existing SS algorithms
- Cons: potentially inefficient, overhead

Design self-stabilizing algorithms for the sensor networks model

- Pros: potentially efficient
- Cons: ignore previous SS work

Self-stabilization in Sensor Networks

Transform (i.e. Simulate) the self-stabilizing model into the sensor networks model

- Pros: reuse existing SS algorithms
- Cons: potentially inefficient, overhead
- [Herman 03] Cached Sensornet Transform

Design self-stabilizing algorithms for the sensor networks model

- Pros: potentially efficient
- Cons: ignore previous SS work

Self-stabilization in Sensor Networks

Transform (i.e. Simulate) the self-stabilizing model into the sensor networks model

- Pros: reuse existing SS algorithms
- Cons: potentially inefficient, overhead
- [Herman 03] Cached Sensornet Transform

Design self-stabilizing algorithms for the sensor networks model

- Pros: potentially efficient
- Cons: ignore previous SS work
- [Herman 03] Unison with collisions

Cached Sensornet Transform

Basic Algorithm

- Each node p has a variable v_p
- Each neighbor q of p has a variable $c_q v_p$
  - $c_q v_p$ is the cached value of v_p at q
- Whenever p assigns v_p, p also broadcasts the new value to the neighborhood
- Whenever a neighbor q of p receives v_p, q updates $c_q v_p$ accordingly

Cached Sensornet Transform

Definition (Cache coherence)

For all neighbors p and q, $c_q v_p = v_p$

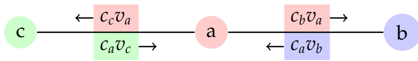
Lemma (Closure)

If started from a cache coherent state, and without collisions, the self-stabilizing model is simulated by replacing all occurrences of $c_q v_p$ by v_p

Example

Lemma (Closure)

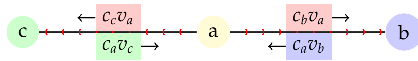
If started from a cache coherent state, and without collisions, the self-stabilizing model is simulated by replacing all occurrences of $c_q v_p$ by v_p



Example

Lemma (Closure)

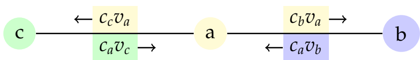
If started from a cache coherent state, and without collisions, the self-stabilizing model is simulated by replacing all occurrences of $c_q v_p$ by v_p



Example

Lemma (Closure)

If started from a cache coherent state, and without collisions, the self-stabilizing model is simulated by replacing all occurrences of $c_q v_p$ by v_p



Cached Sensornet Transform

Periodic retransmit

- Each node p periodically broadcasts v_p to its neighborhood

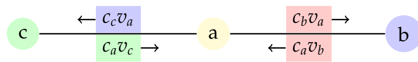
Lemma (Convergence)

If started from an arbitrary state, and without collisions, a cache coherent state is eventually reached

Example

Lemma (Convergence)

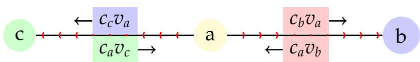
If started from an arbitrary state, and without collisions, a cache coherent state is eventually reached



Example

Lemma (Convergence)

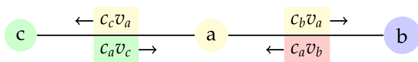
If started from an arbitrary state, and without collisions, a cache coherent state is eventually reached



Example

Lemma (Convergence)

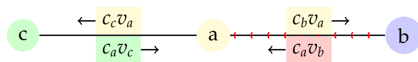
If started from an arbitrary state, and without collisions, a cache coherent state is eventually reached



Example

Lemma (Convergence)

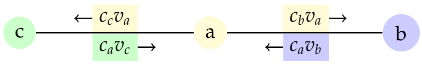
If started from an arbitrary state, and without collisions, a cache coherent state is eventually reached



Example

Lemma (Convergence)

If started from an arbitrary state, and without collisions, a cache coherent state is eventually reached



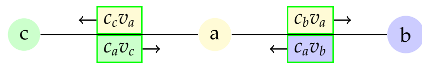
Cached Sensornet Transform

Message Corruption

- Each neighbor q of p has a Boolean variable $b_q v_p$
- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to A$; for all neighbors q of p, $b_p v_q$ becomes false

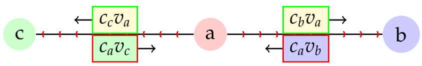
Example

- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to A$; for all neighbors q of p, $b_p v_q$ becomes false



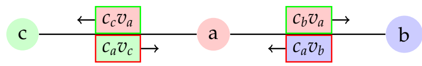
Example

- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to A$; for all neighbors q of p, $b_p v_q$ becomes false



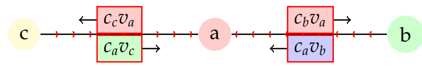
Example

- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to A$; for all neighbors q of p, $b_p v_q$ becomes false



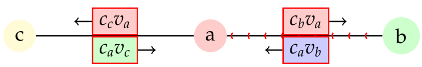
Example

- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to A$; for all neighbors q of p, $b_p v_q$ becomes false



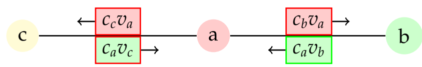
Example

- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to A$; for all neighbors q of p, $b_p v_q$ becomes false



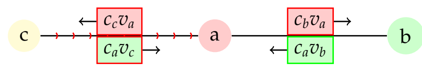
Example

- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to A$; for all neighbors q of p, $b_p v_q$ becomes false



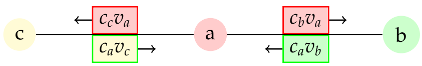
Example

- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to A$; for all neighbors q of p, $b_p v_q$ becomes false



Example

- If q receives v_p correctly, $b_q v_p$ becomes true
- $G \to A$ becomes
  for all neighbors q of p, $b_p v_q$ and $G \to$
  A; for all neighbors q of p, $b_p v_q$ becomes false



Cached Sensornet Transform

Periodic Retransmit

Message Corruption

Lemma (Self-stabilization)

If started from an arbitrary state, the self-stabilizing model is eventually simulated

Self-stabilizing Unison

Specification

- Each node p has a clock variable v_p
- For every neighbors p and q, $|v_p - v_q| \leqslant 1$

Self-stabilizing Unison

Specification

- Each node p has a clock variable v_p
- For every neighbors p and q, $|v_p - v_q| \leqslant 1$

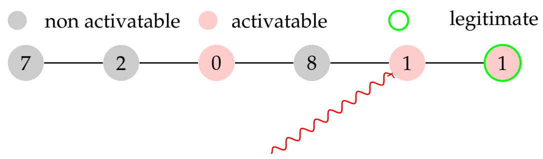
Self-stabilizing Unison

- for every neighbor q, $v_q \geqslant v_p \to v_p := v_p + 1$

Example

Self-stabilizing Unison

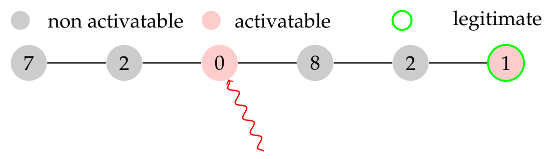
- for every neighbor q, $v_q \geqslant v_p \to v_p := v_p + 1$



Example

Self-stabilizing Unison

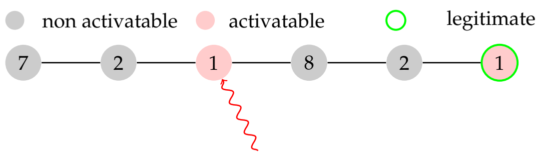
- for every neighbor q, $v_q \geqslant v_p \to v_p := v_p + 1$



Example

Self-stabilizing Unison

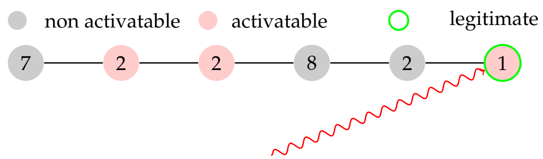
- for every neighbor q, $v_q \geqslant v_p \to v_p := v_p + 1$



Example

Self-stabilizing Unison

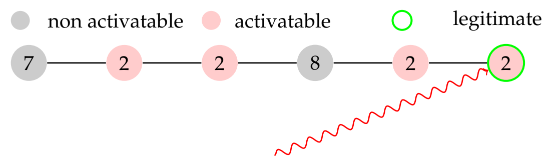
- for every neighbor q, $v_q \geqslant v_p \to v_p := v_p + 1$



Example

Self-stabilizing Unison

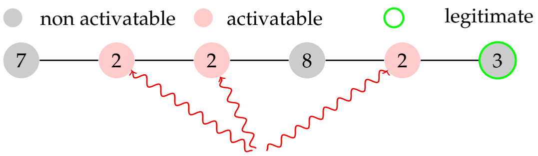
- for every neighbor q, $v_q \geqslant v_p \to v_p := v_p + 1$



Example

Self-stabilizing Unison

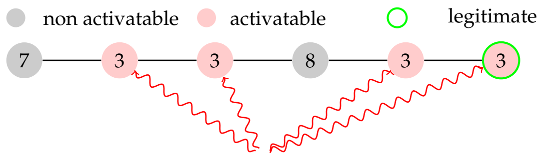
- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Example

Self-stabilizing Unison

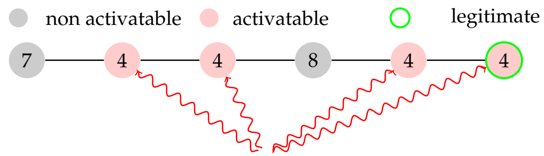
- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Example

Self-stabilizing Unison

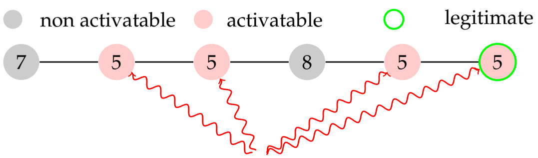
- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Example

Self-stabilizing Unison

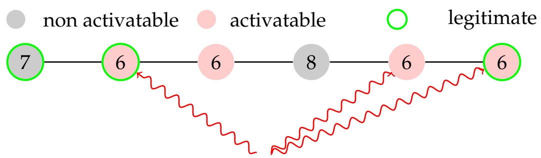
- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Example

Self-stabilizing Unison

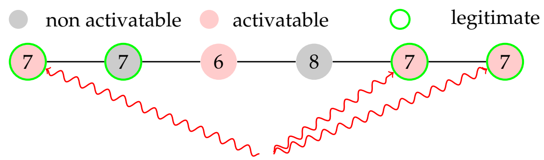
- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Example

Self-stabilizing Unison

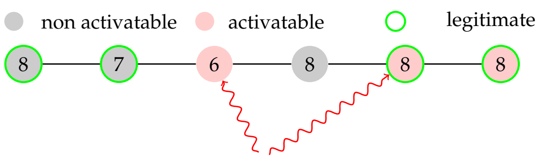
- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Example

Self-stabilizing Unison

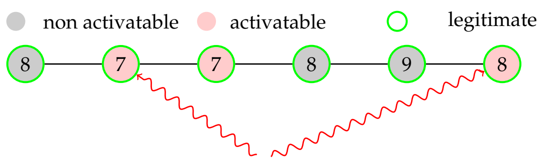
- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Example

Self-stabilizing Unison

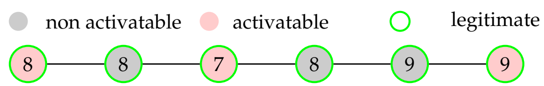
- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Example

Self-stabilizing Unison

- for every neighbor q, $v_q \geqslant v_p \rightarrow v_p := v_p + 1$



Unison with Collisions

Specification

- Each node p has a clock variable v_p
- For every neighbors p and q, $|v_p - v_q| \leqslant 1$

Self-stabilizing Unison

- for every neighbor q, $v_q \geqslant v_p \to v_p := v_p + 1$

Unison with Collisions

Specification

- Each node p has a clock variable v_p
- For every neighbors p and q, $|v_p - v_q| \leqslant 1$

Self-stabilizing Unison with Collisions

- for every neighbor q, $c_p v_q \geqslant v_p \to v_p := v_p + 1$

Unison with Collisions

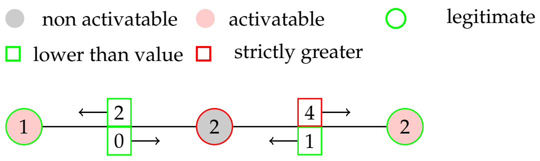
Specification

- Each node p has a clock variable v_p
- For every neighbors p and q, $|v_p - v_q| \leqslant 1$

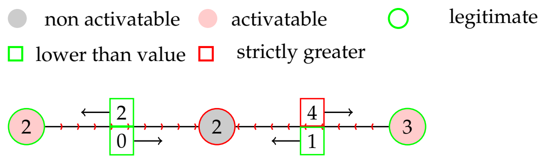
Self-stabilizing Unison with Collisions

- for every neighbor q, $c_p v_q \geqslant v_p \to v_p := v_p + 1$
- Only correctly received messages update cached variables

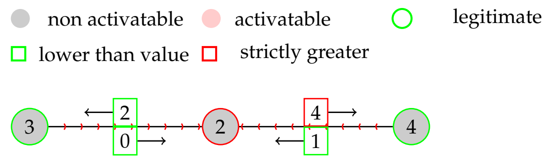
Example



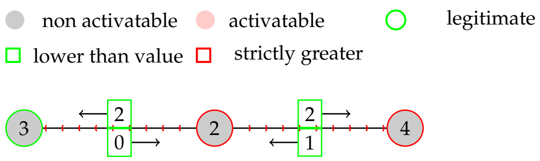
Example



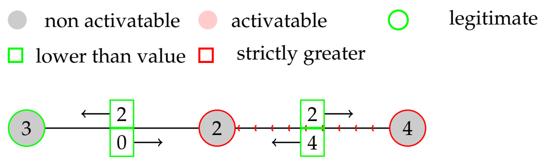
Example



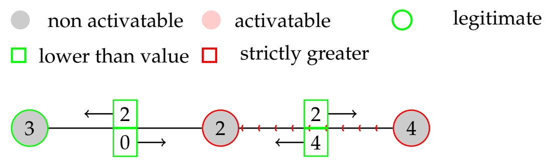
Example



Example



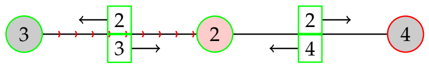
Example



Example

non activatable   activatable   legitimate

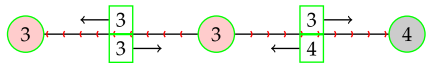
lower than value   strictly greater



Example

non activatable   activatable   legitimate

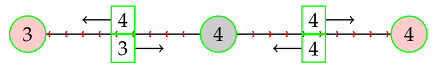
lower than value   strictly greater



Example

non activatable   activatable   legitimate

lower than value   strictly greater



Unison with Collisions

Cache coherence weakening

- For every neighbors p and q, $c_p v_q \leqslant v_q$

Self-stabilizing Unison with collisions

- Unison and Weak cache coherence are preserved by program executions
- Unison and Weak cache coherence eventually hold
- Some extra work is expected to get bounded clock values

Self-stabilization in Sensor Networks

Transform (i.e. Simulate) the self-stabilizing model into the sensor networks model

- [Herman 03] Cached Sensornet Transform
- Overhead is not upper bounded

Design self-stabilizing algorithms for the sensor networks model

- [Herman 03] Unison with collisions
- Proof in the model is specific to the problem