Random Walks, a Paradigm to design Distributed Algorithms for Dynamic Networks

Thibault BERNARD
thibault.bernard@univ-reims.fr

Université de Reims
Champagne-Ardenne

CReSTIC - Equipe
Systèmes Communicants

October 30th 2007
Outline

1 Introduction
   - Main context
   - Tools

2 Random walks characteristics evaluation

3 Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4 Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5 Futur works
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Futur works
Framework study

Context
- Networks
- Distributed Systems (specially Dynamical networks like peer-to-peer or ad-hoc networks)

Constraints
- Management of dynamicity
- Fault tolerance
- Decentralized solutions

Motivation
Conception of solutions to classical problems of distributed algorithmic for dynamic networks
Distributed system

Model of distributed system
- Set of computing resources that communicates through channels
- \((G = (V, E))\)

Communication model:
- Messages passing model
Distributed algorithms

Definition

- Set of local algorithms
- Communication primitives (send, receive, . . .)
- Local execution on each node
Fault Tolerance

Two approaches

- Robustness
  - Lots of impossibility results
  - Requires less restrictive assumptions

- Self-stabilization
**Self-stabilization [Dijk74]**

Two properties for self-stabilizing algorithms:

- **Convergence**: starting from an illegal state and without failures, the algorithm converges to a legal state.
- **Closure**: from a legal state and without failures occurrences, the algorithm remains in legal states.
Plan

1. Introduction
   • Main context
   • Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   • Loss of token
   • Corruption of token
   • Duplication of token
   • Application to ad-hoc networks

4. Network decomposition
   • Motivations
   • Informal description
   • Main mechanisms

5. Futur works
Random walks

- Random moves in a graph
- Algorithmic procedure:
  
  **Reception of the token on the site** \( i \)
  Choose \( j \) uniformly at random among \( \text{Neigh}(i) \)
  Send token to site \( j \)
Properties and mains caracteristics

Percussion
On a finite graph, a random walk hits a node in a finite time. The mean time starting at a node \( i \) to hit node \( j \) is \( h(i,j) \).

Coverage
On a finite graph, a random walk hits all nodes in a finite time. The mean time to visit all nodes is \( C \).

Meeting
On a finite and non-bipartite graph, two random walks meet on the same node in a finite time. The mean time before two random walks meet is \( Me(x_1, x_2) \).
Adaptativity
Adaptativity
Adaptativity
Circulating word

Definition

A circulating word is a message that collects information through its moves in the network.

[Lava86]

Circulating word + random walk

= Tools to collect and broadcast information.
Spanning tree construction

\[ M = \{5\} \]
Spanning tree construction

M = \{2,5\}
Spanning tree construction

\[ M = \{4, 2, 5\} \]
Spanning tree construction

M = \{5, 4, 2, 5\}
Spanning tree construction

M = \{6,5,4,2,5\}
Spanning tree construction

\[ M = \{1, 6, 5, 4, 2, 5\} \]
Spanning tree construction

M = \{5,1,6,5,4,2,5\}
Spanning tree construction

M = \{2,5,1,6,5,4,2,5\}
Spanning tree construction

M = \{3,2,5,1,6,5,4,2,5\}
Adaptativity of the spanning tree

Circulating word moves permanently in the network
\[\Rightarrow\] topological information are alway updated
\[\Rightarrow\] Spanning tree is adaptive

At time \( T \)

\[ M_T = \{5, 3, 2, 5, 4, 5, 1\} \]

At time \( T + \Delta \)

\[ M_{T+\Delta} = \{5, 2, 1, 5, 3, 4, 5, 3, 2, 5, 4, 5, 1\} \]
Reduction of circulating word size

\[ M = \{3, 2, 5, 1, 6, 5, 4, 2, 5\} \]

Circulating word size fixed at \( n \)
Plan

1 Introduction
   - Main context
   - Tools

2 Random walks characteristics evaluation

3 Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4 Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5 Futur works
Computation of RW characteristics

Two approaches

- Markov chains
- Analogy with electrical networks [DoSn84]

Results from [CRST+97, Tet91]

- \( h(i, j) = m \times R(i, j) + \frac{1}{2} \sum_{k \in V} \deg(k) \times (R(j, k) - R(i, k)) \)
- \( m \times R \leq C \leq O(m \times R \times \log n) \)

Computation of RW characteristics \( \Rightarrow \) Computation of \( R(i, j) \)
Electricity laws

Ohm’s law

\[ U_{AB} = R(A, B) \times i_{AB} \]

Kirchhoff’s law

The sum of ingoing currents equals the sum of outgoing currents.

Millman’s theorem

\[ \frac{V - V_1}{r_1} + \frac{V - V_2}{r_2} + \cdots + \frac{V - V_n}{r_n} = 0 \]
**Computation of $R(1, 2)$**

\[
\begin{align*}
V_1 &= 1 \\
V_2 &= 0 \\
V_3 &= \frac{1}{4} \\
V_4 &= \frac{1}{2} \\
V_5 &= \frac{1}{4}
\end{align*}
\]

\[
\begin{align*}
i_2 &= V_1 - V_2 = 1 \\
i_3 &= V_1 - V_4 = \frac{1}{2} \\
i_1 &= i_2 + i_3 = \frac{3}{2}
\end{align*}
\]

\[
R(1, 2) = \frac{1}{3} = \frac{2}{3}
\]
Computation of meeting time

Case of 2 RW
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Futur works
Token circulation

**Primitive for:**
- Election
- Efficient broadcast information
- Resources allocation
- Structures Maintenance
- ...

**Existing solutions**
Based on the construction and maintenance of a virtual spanning structure [ChWe02]
Communication failures

Self-stabilization in messages passing model

- loss of token

- Corruption of token

- Duplication of token
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Futur works
Loss of token

Solution: Timeout

- At the reception of the token, a node reset its timeout
- If the timeout trigger, its node produce a new token
Timeout
Timeout
Timeout

T=7

Diagram of a network showing nodes connected by lines with T=7 indicating the time out value.
Timeout

T=1
Timeout
Timeout
Timeout

Random walk policy as token’s moves

[DoSW02] : $T = C \times i$

$\implies$ Token creation still possible

There is no bound on the visiting time for a given node.
Reloading Wave

Our approach: a solution decided by the token
Broadcast a reset timeout order to all nodes

Principle
- A node receiving this wave, reset its timeout and continue the propagation.
- The wave is propagated periodically through an adaptive spanning tree stored in a circulating word inside the token.
When propagate the wave?

Remarks:
- After a visit, a node produces a new token in $T$ time units.
- The propagation of the wave takes in the worst case $n$ time units.

The token should propagate the reloading wave each $T - n$ time units.
Example
Example

T - n = hop of the token
=> Reloading wave
Example

```
1
|
|
 T=3
|
4
|
5
```

reset msg

```
2
|
|
```

```
3
|
|
T=3
```

Th. Bernard (URCA)

Hagen

October 30th 2007

30 / 54
Example

Wave arriving
timeout reseted
T=7, no token creation

T = 1
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Future works
Corruption of token

Solution

Test realized locally on each visited nodes.

\[ M = \{5, 1, 6, 5, 3, 2, 5\} \]

Here \( 3 \notin \text{Neigh}(5) \), node 5 corrects the word: \( M = \{5, 1, 6, 5\} \).
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Future works
Duplication of token

A solution [IsJa90] (state model)

Meeting property

Mergure of topological information
Scheme of proof

Legal state $\mathcal{EL}$
- One complete and consistent token $\mathcal{JUCC}$
- All nodes have been visited $\mathcal{TSV}$

Lemmas
- A visited node can not produce new tokens
- The number of visited nodes increases
- All nodes become visited nodes: $\mathcal{C} \rightarrow \mathcal{TSV}$
- All tokens become consistent
- There exist at least one complete token
- For all configurations satisfying $\mathcal{C} \in \mathcal{TSV}, \mathcal{C} \rightarrow \mathcal{JUCC}$

Theorem

$$\forall \mathcal{C}, \mathcal{C} \rightarrow \mathcal{EL}$$
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Future works
Model

Communication model

- Nodes communicate through radio waves of different range
- Oriented graph

Mobility

- Dynamic graph
- Nodes moves are a natural behavior
- Alternative solutions to flooding
Adaptation of circulating word management

Permanent algorithm $\implies$ size grows infinitely

Our reduction technique

- based on useful information.
- Allows to renew topological information used.
Adaptation of circulating word management

Remarks
The word $M = \langle 5, 4, 1, 3, 1, 2, 3, 4, 2, 5, 1 \rangle$ allows the construction of spanning tree enrooted on an arbitrary node.

Definition
A cycle $C(i, j)$ in the word $M$ is called constructor if:

$$\forall k \in \text{identities}(M), \exists l \in \{i, \ldots, j\} | M[l] = k$$
Results

Our algorithm

- maintains an adaptive image of the network
- manages connexions and disconnexions
- bounds the size of the circulating word

Lemma

The size of the circulating word is bounded by \( \frac{n^2 + 8n}{4} \)

Simulations results

<table>
<thead>
<tr>
<th>Size of the network</th>
<th>10</th>
<th>100</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average size of the word</td>
<td>17</td>
<td>225</td>
<td>1342</td>
</tr>
<tr>
<td>Deviation</td>
<td>1.9</td>
<td>20</td>
<td>192</td>
</tr>
</tbody>
</table>
**k exclusion for ad-hoc network**

**k exclusion**
- Extension of mutual exclusion
- At most $k$ nodes can get critical section at a given time

**k distinct tokens will circulate (Set of colors $\mathcal{K}$)**

**Convergence**
- Preliminary phase: deletion of corrupted tokens
- Production phase: produce at least one token by colors of $\mathcal{K}$
- Mergure phase: deletion of duplicated tokens
Mobility assumption

Self-stabilizing if:

An edge chosen to belong to the spanning tree should permit the propagation of the reloading wave during the time it belongs to this tree.
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Future works
Motivation

**Limitation of Random Walks**

- In the most general case, Bounds on hitting, cover and meeting time are in $O(n^3)$
- Acceptable in theory but not for practical application.

**Differents kinds of solutions (depending of the task to achieve)**

- Increasing the number of random walks
- Build and maintain a hierarchical structure over the network
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Future works
Goal

Get a decomposition of the network into partition. Each of them should have a number of node comprised between $m_{min}$ and $m_{max}$. 
Goal
Goal
Goal
Goal
Goal
Goal
Goal
Goal
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
     - Main mechanisms

5. Future works
Annexion

**Condition**
The local random walk annexes a new node and

\[ Nb_{\text{nodes}} \leq m_{\text{max}} \]

**Operations**
- Mark the new node with the LRW color.
- Update LRW information
Division

Condition

The local random walk annexes a new node and

\[ Nb_{\text{nodes}} > m_{\text{max}} \]

Operations

- Update LRW information
- Send a wave on the spanning structure to split it into two parts
Submission

Condition

The local random walk visits another partition and

\[ Nb_{nodes} < m_{\min} \]

Operations

- Kill the LRW
- Propagate a wave through the partition to change the color partition
Plan

1. Introduction
   - Main context
   - Tools

2. Random walks characteristics evaluation

3. Fault tolerant Token Circulation
   - Loss of token
   - Corruption of token
   - Duplication of token
   - Application to ad-hoc networks

4. Network decomposition
   - Motivations
   - Informal description
   - Main mechanisms

5. Futur works
Futur works

Adaptive hierarchization
- Formal proof of the algorithm
- Self-stabilizing behaviour?

Experimentations
- Topology of the partitions created
- Stability of the partition regarding the mobility of the network

Applications
Hybridation of random walks and *local flooding* for ad-hoc networks
Random Walks, a Paradigm to design Distributed Algorithms for Dynamic Networks

Thibault BERNARD
thibault.bernard@univ-reims.fr

Université de Reims
Champagne-Ardenne

CReSTIC - Équipe Systèmes Communicants

October 30th 2007