The Power of a Leader

In the Stone Age

Stephan Holzer - MIT

Yuval Emek - Technion
Roger Wattenhofer - ETH Zürich

2nd Workshop on Biological Distributed Algorithms, October 11-12, 2014 in Austin, Texas USA
The Power of a Leader

In the Stone Age

Stephan Holzer - MIT

Yuval Emek - Technion
Roger Wattenhofer - ETH Zürich

2nd Workshop on Biological Distributed Algorithms, October 11-12, 2014 in Austin, Texas USA
Stone Age Model of Distributed Computing
Computational Power of a Cell?
Computational Power of a Cell?

- Finite State Machine
Communication?
Communication?

- Transmissions:
  - Same message delivered to all neighbors
- Constant size message
- Port for each neighbor
  - Stores the last message delivered
Communication?

• Transmissions:
  – Same message delivered to all neighbors
• Constant size message
• Port for each neighbor
  – Stores the last message delivered
• Can detect if 0, 1 or 2+ ports store message m
• FSM changes state based on this sends message based on state
Stone Age Model of Distributed Computing

- All nodes run the same FSM
  - (random)
- Anonymous
- Weak communication
- Fully asynchronous
- Arbitrary network topology
  - (unknown)

- All features of the protocol are constant!
What is known?

- Can be synchronized
- Cannot elect a leader
- Cannot compute shortest paths
  - No Minimum Spanning Tree or Diameter

Edsger W. Dijkstra: Actually: I CAN COMPUTE SHORTEST PATHS!
Edsger W. Dijkstra:

Actually:
I CAN COMPUTE SHORTEST PATHS!

(The slime mold)

Actually:
Me too!
Physarum polycephalum

- Nuclei are nodes
- Tubes / plasma are edges
Computes the Shortest Path

Video can be found at: [http://www.youtube.com/watch?v=czk4xgdhdY4](http://www.youtube.com/watch?v=czk4xgdhdY4)
How a Leader can make a Difference!

- Symmetry is broken
- Can coordinate global computation
- E.g. select unique node at random
Select Unique Node at Random

- REPEAT:
  1. select random neighbor
  2. Pr[choose yourself] = 1/2
Select Unique Node at Random

- **REPEAT:**
  1. select random neighbor
  2. $\text{Pr}[\text{choose yourself}]=1/2$
Select Unique Node at Random

- REPEAT:
  1. select random neighbor
  2. \( \Pr[\text{choose yourself}] = \frac{1}{2} \)
Select Unique Node at Random

1. REPEAT:
   1. select random neighbor
   2. \( \Pr[^{\text{choose yourself}}] = 1/2 \)

Wait:

Transmissions:
- Same message delivered to all neighbors

Solution:
- Wait until there is a neighbor in a unique state among the neighbors
- Detect this via ports
- Transmit “content of this port”
Select Unique Node at Random

• REPEAT:  
  1. select random neighbor 
  2. $\Pr[\text{choose yourself}] = \frac{1}{2}$
Select Unique Node at Random

- **REPEAT:**
  1. Select random neighbor
  2. $\text{Pr}[\text{choose yourself}] = \frac{1}{2}$
Select Unique Node at Random

- REPEAT:
  1. select random neighbor
  2. $\Pr[\text{choose yourself}]=1/2$
Select Unique Node at Random

• REPEAT: 1. select random neighbor
          2. Pr[choose yourself] = 1/2
Select Unique Node at Random

- REPEAT:
  1. select random neighbor
  2. Pr[choose yourself] = 1/2
Select Unique Node at Random

• REPEAT: 1. select random neighbor
2. \( \text{Pr[choose yourself]} = 1/2 \)
Select Unique Node at Random

- **REPEAT:**
  1. select random neighbor
  2. \( \text{Pr[choose yourself]} = 1/2 \)
Select Unique Node at Random

- **REPEAT:**
  1. select random neighbor
  2. $Pr[\text{choose yourself}] = \frac{1}{2}$
Select Unique Node at Random

- REPEAT:
  1. select random neighbor
  2. Pr[choose yourself]=1/2
How a Leader can make a Difference!

- Symmetry is broken
- Can coordinate global computation
- Select unique node at random
- Check if all nodes up to a certain distance have a certain property
Check if all nodes up to a certain distance have a certain property

Determines the distance (here: 2)
Check if all nodes up to a certain distance have a certain property

Determines the distance (here: 2)
Check if all nodes up to a certain distance have a certain property

Use ping to check properties

Determines the distance (here: 2)
How a Leader can make a Difference!

• Symmetry is broken
• Can coordinate global computation
• Select unique node at random
• Check if all nodes up to a certain distance have a certain property
How a Leader can make a Difference!

- Symmetry is broken
- Can coordinate global computation
- Select unique node at random
- Check if all nodes up to a certain distance have a certain property (works for subsets as well)
How a Leader can make a Difference!

• Symmetry is broken
• Can coordinate global computation
• Select unique node at random
• Check if all nodes up to a certain distance have a certain property (works for subsets as well)
• Iterate through all nodes (for $v \in V$ do TASK)
Iterate through all nodes (for $v \in V$ do)

$v_{\text{max}} := \text{leader}$;

while not all nodes at distance $\leq \text{dist(leader, } v_{\text{max}} \text{)} + 1$
are marked do

select random node $u$;  // $u$ marks itself
if $\text{dist(leader, } u \text{)} > \text{dist(leader, } v_{\text{max}} \text{)}$ then

$v_{\text{max}} := u$;

$u$ performs TASK;
Iterate through all nodes (for $\nu \in V$ do)

\[
\nu_{\text{max}} := \text{leader};
\]

while not all nodes at distance $\leq \text{dist(leader, } \nu_{\text{max}}) + 1$ are marked do

select random node $u$;  // $u$ marks itself

if dist(leader, $u$) > dist(leader, $\nu_{\text{max}}$) then

\[
\nu_{\text{max}} := u;
\]

$u$ performs TASK;
Iterate through all nodes $(\text{for } v \in V \text{ do})$

$v_{\text{max}} := \text{leader}$;

while not all nodes at distance $\leq \text{dist}(\text{leader}, v_{\text{max}}) + 1$ are marked do

select random node $u$; // $u$ marks itself

if $\text{dist}(\text{leader}, u) > \text{dist}(\text{leader}, v_{\text{max}})$ then

$v_{\text{max}} := u$;

u performs TASK;
Iterate through all nodes (for $\nu \in V$ do)

$n_{max} := leader$;

while not all nodes at distance $\leq dist(leader, n_{max}) + 1$ are marked do

select random node u;  // u marks itself

if $dist(leader, u) > dist(leader, n_{max})$ then

$n_{max} := u$;

u performs TASK;
Iterate through all nodes (for $v \in V$ do)

$v_{max} \ := \ leader;$

**while not** all nodes at distance $\leq \text{dist}($leader,$v_{max}) + 1$ are marked **do**

**select** random node $u;$ \ // $u$ marks itself

**if** $\text{dist}($leader,$u) > \text{dist}($leader,$v_{max})$ **then**

$v_{max} \ := \ u;$

$u$ performs TASK;
Iterate through all nodes (for $v \in V$ do)

$v_{\text{max}} := \text{leader}$;

while not all nodes at distance $\leq \text{dist(leader, } v_{\text{max}}) + 1$ are marked do

select random node $u$; // $u$ marks itself

if $\text{dist(leader, } u) > \text{dist(leader, } v_{\text{max}})$ then

$v_{\text{max}} := u$;

$u$ performs TASK;
Iterate through all nodes (for $v \in V$ do)

$v_{max} := leader$;

while not all nodes at distance $\leq \text{dist}(\text{leader}, v_{max}) + 1$ are marked do

select random node $u$;  // $u$ marks itself

if $\text{dist}(\text{leader}, u) > \text{dist}(\text{leader}, v_{max})$ then

$v_{max} := u$;

$u$ performs TASK;
Iterate through all nodes (for $v \in V$ do)

\[
\nu_{max} := leader;
\]
\[
\textbf{while not} \text{ all nodes at distance } \leq \text{dist}(\text{leader}, \nu_{max})+1 \text{ are marked } \textbf{do}
\]
\[
\textbf{select} \text{ random node } u; \quad \text{// u marks itself}
\]
\[
\textbf{if} \text{dist}(\text{leader},u) > \text{dist}(\text{leader}, \nu_{max}) \textbf{then}
\]
\[
\nu_{max} := u;
\]
\[
\text{u performs TASK;}
\]
Iterate through all nodes (for $v \in V$ do)

$v_{\text{max}} := \text{leader}$;

while not all nodes at distance $\leq \text{dist(leader, } v_{\text{max}}) + 1$
are marked do

select random node $u$;  // $u$ marks itself

if $\text{dist(leader, } u) > \text{dist(leader, } v_{\text{max}})$ then

$v_{\text{max}} := u$;

$u$ performs TASK;
Iterate through all nodes (for $\nu \in V$ do)

$v_{max} := leader$;

while not all nodes at distance $\leq \text{dist}(\text{leader}, v_{max}) + 1$ are marked do

select random node $u$;  // $u$ marks itself

if $\text{dist}(\text{leader}, u) > \text{dist}(\text{leader}, v_{max})$ then

$v_{max} := u$;

$u$ performs TASK;
Iterate through all nodes (for $v \in V$ do)

$v_{\text{max}} := \text{leader};$

while not all nodes at distance $\leq \text{dist}(\text{leader}, v_{\text{max}}) + 1$ are marked do

select random node $u$;  // $u$ marks itself

if $\text{dist}(\text{leader}, u) > \text{dist}(\text{leader}, v_{\text{max}})$ then

$v_{\text{max}} := u;$

$u$ performs TASK;
Iterate through all nodes (for $v \in V$ do)

$v_{\text{max}} := \text{leader};$

while not all nodes at distance $\leq \text{dist(leader,} v_{\text{max}}) + 1$
are marked do

select random node $u$; // $u$ marks itself

if $\text{dist(leader,} u) > \text{dist(leader,} v_{\text{max}})$ then

$v_{\text{max}} := u;$

$u$ performs TASK;
Iterate through all nodes \((\text{for } v \in V \text{ do})\)

\[ v_{\text{max}} := \text{leader}; \]

\textbf{while not} all nodes at distance \(\leq \text{dist(leader, } v_{\text{max}}) + 1\) are marked \textbf{do}

\textbf{select} random node \(u; \quad //\ u \text{ marks itself}

\textbf{if} \ \text{dist(leader, } u) > \text{dist(leader, } v_{\text{max}}) \textbf{then}

\[ v_{\text{max}} := u; \]

\(u\) performs \(\text{TASK};\)
Iterate through all nodes (for $\nu \in \mathcal{V}$ do)

$v_{\text{max}} := \text{leader}$;

while not all nodes at distance $\leq \text{dist}(\text{leader}, v_{\text{max}}) + 1$ are marked do

select random node $u$;  // $u$ marks itself

if $\text{dist}(\text{leader}, u) > \text{dist}(\text{leader}, v_{\text{max}})$ then

$v_{\text{max}} := u$;

$u$ performs TASK;
How a Leader can make a Difference!

• Symmetry is broken
• Can coordinate global computation
• Select unique node at random
• Check if all nodes up to a certain distance have a certain property (works for subsets as well)
• Iterate through all nodes (\texttt{for } u \in V \texttt{ do TASK})
How a Leader can make a Difference!

- Symmetry is broken
- Can coordinate global computation
- Select unique node at random
- Check if all nodes up to a certain distance have a certain property (works for subsets as well)
- Iterate through all nodes (\textbf{for } u \in V \textbf{ do TASK})
- Decide dist(u,v) > dist(u’,v’)
Decide $\text{dist}(u,v) > \text{dist}(u',v')$

Perform BFS
Decide $\text{dist}(u,v) > \text{dist}(u',v')$

- **leader**
  - Acts as synchronizer

Perform BFS

- $u$
- $u'$
- $v$
- $v'$
Decide $\text{dist}(u,v) > \text{dist}(u',v')$

- Leader: Acts as synchronizer
- Perform BFS:
  - $u$
  - $u'$
  - $v$
  - $v'$
Decide $\text{dist}(u,v) > \text{dist}(u',v')$
Decide $\text{dist}(u,v) > \text{dist}(u',v')$
Decide $\text{dist}(u,v) > \text{dist}(u',v')$
How a Leader can make a Difference!

- Symmetry is broken
- Can coordinate global computation
- Select unique node at random
- Check if all nodes up to a certain distance have a certain property (works for subsets as well)
- Iterate through all nodes \((\text{for } u \in V \text{ do TASK})\)
- Decide \(\text{dist}(u,v) > \text{dist}(u',v')\)
How a Leader can make a Difference!

• Symmetry is broken
• Can coordinate global computation
• Select unique node at random
• Check if all nodes up to a certain distance have a certain property (works for subsets as well)
• Iterate through all nodes (for \( u \in V \) do TASK)
• Decide \( \text{dist}(u,v) > \text{dist}(u',v') \)

• Constant combination of all these
Shortest Path

**INPUT:** nodes u, v  
(assume v is leader)

While $u \neq v$ do

u marks itself to be on the shortest path

for each neighbor w of u do

if $\text{dist}(w, v) < \text{dist}(u, v)$ then

$u := w$;

**OUTPUT:** marked nodes
Shortest Path

• Example:
Shortest Path

• Example:
Shortest Path

• Example:
Shortest Path

• Example:
Shortest Path

• Example:
Shortest Path

• Example:
Shortest Path

• Example:
Shortest Path

• Example:

Diagram of a network with nodes and connections, illustrating the concept of shortest path.
Shortest Path

• Example:
Shortest Path

• Example:
Different Model for the Slime Mold?

• Keep algorithms simple

• How does communication work?
  – Strength of flow through tubes indicates length.
  – We assume additional communication through tubes/plasma.
  – Communicate through moving nuclei?

• Nuclei:
  – How long do they live? Robustness?
  – Do they move?
  – How many are there?
  – Do they communicate / process information?
These are possibility results
- which weak assumptions work
- which don’t
Thanks!

holzer@mit.edu
yemek@ie.technion.ac.il
wattenhofer@ethz.ch