Exploration of Networks Containing Malicious Hosts

Euripides Markou[1]

[1]Department of Computer Science & Biomedical Informatics,
University of Central Greece, Lamia, Greece.

TRANSFORM Summer School, June 2013,
FORTH, Heraklion, Greece

Outline

Outline

Mobile Agents

Mobile autonomous computational entities

- Can transport their state from one computational environment to the next and resume their execution in the new environment.
- A powerful tool for implementing distributed applications in computer networks.

Applications

- electronic commerce,
- network maintenance,
- *Personal Digital Assistants (PDAs)*

Mobile Agents

Advantages

- migration at a better location in the network,
- fewer messages exchanged,
- no need of pre-installing software at nodes of the network,
- better operation in case of dynamic change of topology,
- better operation in case of faults in the network,
- new applications.

Disadvantages

- security.

Outline

Mobile Agents

Security

- agent attacks at a node,
  - masquerading agent
  - denial of service
  - unauthorized access to data
- node attacks an agent,
  - masquerade
  - repudiation
  - unauthorized access
- agent attacks an agent.
  - masquerade
  - denial of service
  - eavesdropping
  - alteration

Mobile Agents

Security Demands

- data privacy,
- communications privacy,
- location privacy,
- integrity,
- accountability,
- availability,
- anonymity.

Mobile Agents

Actions towards security

- recording and maintaining agent's history,
- agents update their homebases,
- agents watching other agents,

Important question

Can we solve the security issues and maintain the advantages (e.g., performance)?

Outline

Mobile Agents

Towards a formal definition

- A robot with memory executing a (deterministic) algorithm and capable of:
  - having sensors to scan the environment,
  - moving in the network,
  - exchanging messages with other robots,
  - reading and writing information at nodes,
  - carrying *tokens*, etc.
- *Finite Automaton, Turing Machine*

Mobile Agents

Finite Automaton $A = (X, Y, S, \delta, \lambda, S_0)$

- $X \subseteq C_A \times C_E$:
  - C_A : a finite set consisting of all possible configurations of an agent,
  - C_E : a finite set consisting of all possible configurations of the environment as observed by the agent,
- Y : a finite set of actions that an agent can take,
- S : a finite set of states among which there is a special state S_0 called initial state,
- $\delta : S \times X \to S$: transition function,
- $\lambda : S \to Y$: defines which action an agent takes.

Mobile Agents

Finite Automaton $A = (X, Y, S, \delta, \lambda, S_0)$

- Initially, agent A resides at a node u_0 in the initial state S_0
- Agent A observes the environment and (according to functions δ, λ) enters another state $\sigma \in S$ and executes action $\lambda(S_0)$.
- When A visits a node v, observes the environment (i.e, reads *port* through which reached v and the node's state).
- The environment input, combined with its current state σ cause a transition to a new state σ' (according to δ).
- The new state σ' determines an action $\lambda(\sigma')$ etc.

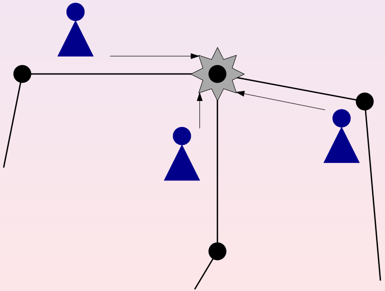
Mobile Agents

Properties

- It is usually needed that the agents terminate after a finite number of steps. This is not always achievable.
- Sometimes the agents cannot decide *when* their task has been accomplished. Or some of the agents may move forever without having done their task.
- When the agents execute the same deterministic algorithm they ‘break’ their symmetries e.g., by using their (distinct) identities, or by observing the environment, etc.

Black Hole Search (BHS)

- Network (Connected Graph G)
- Mobile agents
- Dangerous node (Black Hole) :
  destroys any agent visiting that node



Black Hole Search

Explore G and locate BH

Optimize :

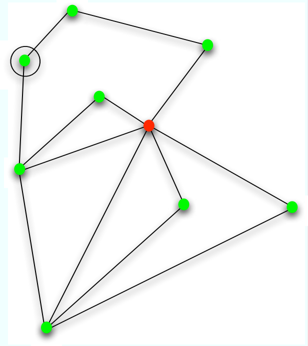
- # moves
- # agents

The Black Hole Search problem

Black Hole: A highly harmful stationary process residing at a node of a network and destroying all mobile agents visiting that node without leaving any trace.

Black Hole Search problem:

A graph is given. There is at most one black hole in the graph. Instruct mobile agents to locate the black hole and return to the starting point.



Outline

Communication mechanisms

Since any agent visiting a black hole is vanished without leaving any trace, the location of the black hole must be deducted by some communication mechanism implemented by the agents.

- the whiteboard model: there is a whiteboard at each node of the network where the agents can leave messages,
- the ‘pure’ token model: where the agents carry tokens which they can leave at nodes,
- the ‘enhanced’ token model: the agents can leave tokens at nodes or edges,
- the ‘time-out’ mechanism (only for synchronous networks): one agent explores a new node and then (after a pre-determined fixed time) informs another agent who waits at a safe node.

Outline

Asynchronous model

S. Dobrev, P. Flocchini, G. Prencipe, N. Santoro, 2001

Asynchronous network (no upper bound on the time needed for an edge traversal), anonymous graph, anonymous agents. At each node there is a whiteboard where the agents can leave messages.

Questions

- How many agents are needed to locate the black hole?
- What a-priori knowledge is required?
- How long does it take?

Some negative results on the asynchronous model

- The network must be 2-connected.
- It is impossible to answer the question of whether a black hole actually exists on the network.
- It is impossible to locate the black-hole if the size of the network is unknown.

On a ring topology.

- Any algorithm needs at least $2n - 4$ edge-traversals to find the black-hole regardless of the number of agents available.

Some positive results on the asynchronous model

- Two co-located agents can locate the black-hole on a ring with whiteboards within $O(n \log n)$ edge-traversals.
- $n-1$ agents can locate the black-hole on a ring of n nodes within $2n-4$ edge traversals.

$n-1$ agents in an asynchronous ring with whiteboards

1. Agents get distinct identities from the set $\{1, 2, \ldots, n-1\}$ and agree on the clockwise orientation
2. Agent i travels $i-1$ edges in clockwise direction
3. Agent i travels $n-2$ edges in counter-clockwise direction
4. Agent i returns to homebase traveling in clockwise direction
5. Agent i reports that the black hole resides at a distance i clockwise

2 agents in an asynchronous ring with whiteboards

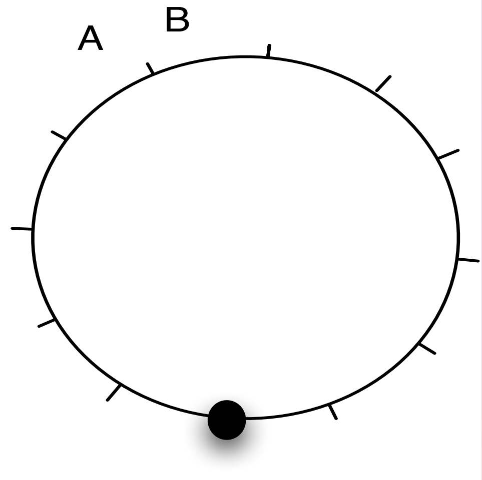
Cautious-Walk mechanism

- before leaving a node v_i going to a node v_{i+1}, the agent marks the port leading from v_i to v_{i+1} as *active*,
- immediately after visiting v_{i+1}, the agent returns to v_i, and marks the port leading from v_i to v_{i+1} as *safe*,
- the agent checks for messages at v_i and (if such a message exists) re-assigns to itself an unexplored area U'_k which has to discover and repeats from the start.

2 agents in an asynchronous ring with whiteboards

1. The agents get distinct identities and agree on the clockwise orientation
2. Let $X := h$
3. **While** $(|E| < n - 1)$
   - Divide U into two continuous disjoint parts U^L (starting counter-clockwise of node X) and U^R (starting clockwise of node X) of almost equal sizes
   - Agent 1(2) leaves a message at X saying that she will explore $U^L(U^R)$
   - Agent 1(2) explores $U^L(U^R)$ using *Cautious Walk*
   - Agent 1(2) traverses clockwise (counter-clockwise) the explored area until it reaches a node u with a not safe port
   - Update E and U
   - Let $X := u$
4. Report the black-hole location

2 agents in an asynchronous ring with whiteboards



Outline

1. Mobile Agents
   - Introduction
   - Security
   - Definitions
2. Black Hole Search problem
   - Communication mechanisms
   - Asynchronous network
   - Synchronous network
3. Black-Hole Search using DFAs and tokens
4. Gray holes

Synchronous model

Synchronous network (at least partially, i.e. there is an upper bound on the time needed by an agent for traversing any edge).

- At most one black hole.
- Two agents having a complete map of the graph.
- The agents have distinct labels.
- They can communicate only when they meet (and not by leaving messages at nodes).

Observations

- Two agents are needed to discover one black hole
- The agents can decide whether there is a black hole or not

We define the cost of a black hole search scheme as the number of edge-traversals taken under the worst-case location of the black hole (or when it does not exist in the network).

Some questions and results on the synchronous model

Questions

- How fast can you locate the black hole (or decide that there is no black hole)?
- Find an algorithm which, given a graph and a starting point, returns a black hole search scheme of a minimum cost.

Some results [J. Czyzowicz, D. Kowalski, M, A. Pelc, 2004-2007], [R. Klasing, M, T. Radzik, F. Sarracco, 2005-2008]

- $5/3$ approximation algorithm for arbitrary trees
- optimal algorithms for special tree families
- NP-hard in arbitrary graphs
- $7/2$ approximation algorithm for arbitrary graphs

Terminology

A black hole search scheme (BHS-scheme) for the input (G,s) is a pair of sequences of edge traversals (moves) of each of the two agents with the following properties:

- Each move takes one time unit.
- Upon completion of the scheme there is at least one surviving agent which knows the location of the black hole.
- The surviving agents must return to s.

We consider a node explored if all the remaining agents know whether it is safe or not.
The explored territory increases at meeting points.
We call the sequence of moves between meeting points, a phase.

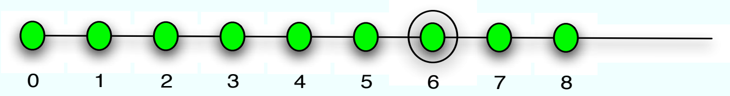
Basic properties and exploration tactics

- An unexplored node cannot be traversed by both agents
- During a phase an agent can traverse at most one unexplored node
- At the end of each phase, the explored territory is increased by 1 or 2 nodes
- Exploration tactics:
  - Probe: one agent visits exactly one unexplored node and returns to meet the other agent who has remained in the explored territory.
  - Split: both agents exit the explored territory, visit exactly one unexplored node each, and return to the explored territory to meet.

The case of the line

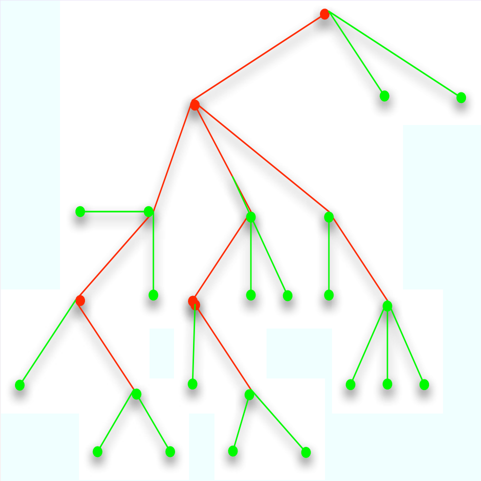
An **optimal algorithm** for a black hole search in at most $4n - 8$ time units.

- Split(s-1, s+1);
- Split(s-2, s+2);
- Walk(s-1);
- Walk-and-Probe(1);
- Split(0, s+3);
- Walk(s+2);
- Walk-and-Probe(n);



An optimal algorithm for a bushy tree: agents always split

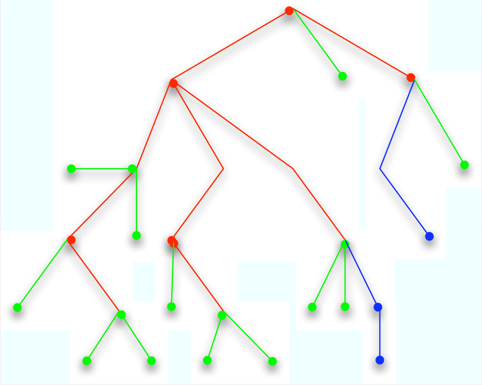
- *red edge:* ≥ 6 traversals
- *green edge:* ≥ 2 traversals



An arbitrary tree

A 5/3 approximation algorithm

- *blue branch:* ≥ 6 traversals



Black hole search in an arbitrary graph

Hamiltonian Circuit problem

Instance : cubic planar graph $G = (V, E)$, and an edge $(x, y) \in E$;

Question : does G contain a Hamiltonian cycle that includes edge (x, y)?

Black hole search problem (decision version)

Instance : graph $G' = (V', E')$, with a starting node $s \in V'$, and a positive integer X;

Question : does there exist a BHS scheme σ for G' starting from s, such that the BHS based on σ has cost at most X?

BHS: NP-hardness and Approximation

Finding a fastest BHS-scheme for arbitrary graphs is NP-hard

The graph G has a hamiltonian cycle, iff the constructed graph G' has a BHS scheme with cost at most $5n + 2$.

Approximation algorithms

- There is an easy 4-approximation algorithm by taking any spanning tree and probe.
- Better approach (minimize waiting time, maximize splits) giving a $\frac{7}{2}$-approximation algorithm:
    - find a ‘good’ spanning tree
    - generate a BHS scheme for the tree

Black-Hole Search using DFAs and tokens

The agents:

- are anonymous (same deterministic algorithm)
- have constant memory
- carry a constant number of tokens
- see other agents or tokens when they are at the same node.
- are synchronous
- are initially scattered in the network
- have no knowledge about the size of the network or the total number and initial positions of the agents

Network Exploration

Dangerous Network

- Team of agents
- Co-located, distinct identities
- Synchronous or Asynchronous
- Communication:
  - whiteboards
  - tokens-on-nodes
  - tokens-on-edges

Safe Network

- Single agent
- Restricted Memory (logarithmic)
- Constant memory + Tokens
- Undirected / Directed
- How much memory?
- How many tokens?

BHS with constant memory agents

Question

Can agents having constant memory explore dangerous graphs and locate BH?

Yes if :

- they are co-located and synchronous
- the graph can be explored (e.g., a ring)

Time-out mechanism

next node is safe

next node is the BH

BHS with constant memory agents

Question

Can agents having constant memory explore dangerous graphs and locate BH?

Yes if :

- they are co-located and synchronous
- the graph can be explored (e.g., a ring)

Time-out mechanism

next node is safe

next node is the BH

BHS with constant memory agents

Question

Can agents having constant memory explore dangerous graphs and locate BH?

Yes if :

- they are co-located and synchronous
- the graph can be explored (e.g., a ring)

Time-out mechanism

next node is safe

next node is the BH

BHS with constant memory agents

Question

Can agents having constant memory explore dangerous graphs and locate BH?

Yes if :

- they are co-located and synchronous
- the graph can be explored (e.g., a ring)

Time-out mechanism

next node is safe

next node is the BH

BHS with constant memory agents

Question

Can agents having constant memory explore dangerous graphs and locate BH?

Yes if :

- they are co-located and synchronous
- the graph can be explored (e.g., a ring)

Time-out mechanism

next node is safe   next node is the BH

BHS with constant memory agents

Question

Can agents having constant memory explore dangerous graphs and locate BH?

Yes if :

- they are co-located and synchronous
- the graph can be explored (e.g., a ring)

Time-out mechanism

next node is safe

next node is the BH

BHS with constant memory agents

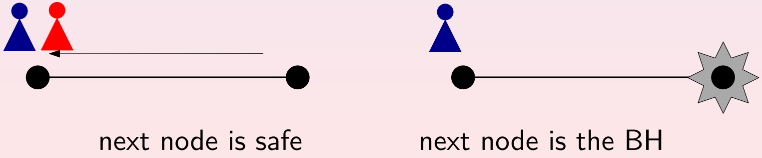
Question

Can agents having constant memory explore dangerous graphs and locate BH?

Yes if :

- they are co-located and synchronous
- the graph can be explored (e.g., a ring)

Time-out mechanism



BHS with scattered constant memory agents

Question

Can we solve the BHS problem with scattered constant memory agents using tokens?

Yes in : [J. Chalopin, S. Das, A. Labourel, M, 2011-2012]

- Anonymous Rings (cycles $n \geq 3$)
  - oriented : all agents agree on the ring's orientation
  - non-oriented : agents may disagree on the orientation
- Anonymous Torus (toroidal grids $n \times m$, $n, m \geq 3$)
  The agents agree on the cardinal directions (N,S,W,E)

BHS with scattered constant memory agents

Question

Can we solve the BHS problem with scattered constant memory agents using tokens?

Yes in : [J. Chalopin, S. Das, A. Labourel, M, 2011-2012]

- Anonymous Rings (cycles $n \geq 3$)
  - oriented : all agents agree on the ring's orientation
  - non-oriented : agents may disagree on the orientation
- Anonymous Torus (toroidal grids $n \times m$, $n, m \geq 3$)
  The agents agree on the cardinal directions (N,S,W,E)

Minimizing the # of agents and tokens

Our goals [J. Chalopin, S. Das, A. Labourel, M, 2011-2012]

- Find the minimal resources to solve the problem: # of agents and # of tokens per agent
- Find trade-offs between these two resources
- Evaluate the impact of constant memory on the resources needed
- Observe the differences between movable (reusable) tokens and unmovable tokens.

Our results on the ring (*SIROCCO '11*)

		Resources necessary and sufficient	
Tokens are	**Ring is**	**# agents**	**# tokens**
Movable	Oriented	3	1
	Unoriented		
Unmovable	Oriented	4	2
	Unoriented	5	2

- Lower bounds work even if agents have non-constant memory.
- No trade-off between # of agents and # of tokens.

Our results on the torus (*DISC '11*)

- **Impossibility results :**
  - No algorithm for any constant number of agents with any constant number of unmovable tokens
  - No algorithm for 3 agents with 1 movable token each
  - No algorithm for 2 agents (with any number of tokens), even if they have unlimited memory
- **Algorithms :**
  - An algorithm for ≥ 3 agents with 3 tokens
  - An algorithm for ≥ 4 agents with 2 tokens
  - An algorithm for exactly 3 agents with 2 tokens

Gray-hole

A Byzantine black hole at a node u...

- may kill an agent A before/while/after A has access to u's whiteboard,
- can transmit the visiting agent to a different destination node than the agent requested or even without the agent requests a transmission,
- may also serve incoming agents in different order than the order they arrived,
- may copy an incoming agent A and forward A more than once.

R. Kralovic, S. Miklik, Periodic Data Retrieval Problem in Rings Containing a Malicious Host, SIROCCO 2010.

Gray-hole - More severe behavior

A Byzantine black hole at a node u...

- may change the contents of u's whiteboard,
- may change the memory of a visiting agent.

The Periodic Data Retrieval problem in spite of a gray hole.

The Periodic Data Retrieval problem

The agents operate in the network and their aim is to deliver the data from any safe node to the home-base infinitely many times.

The model

- A labeled oriented ring network of n nodes (hosts) connected by bidirectional, FIFO links.
- In each host there is a whiteboard.
- The mobile agents are identical, asynchronous computational processes with distinct identities and memory and they have a map of the network. They are initially located at the same node.

The Periodic Data Retrieval problem in spite of a gray hole

New results

- No algorithm can solve the problem in rings containing a gray-hole with reliable whiteboard using less than four agents (previously best: three agents).
- The problem in rings containing a gray-hole with reliable whiteboard can be solved using four agents (previously best: nine agents).
- No algorithm can solve the problem in rings containing a gray-hole with an unreliable whiteboard using less than five agents (previously best: three agents).