The Black Hole Search Problem
Facing the harmful host threats in distributed mobile computing environments

The Problem: distributed mobile computing

Network sites (host)

Local processes (stationary)

Mobile agents
• Navigate from host to host
• Perform computations (same protocol)
• Asynchronous
• Use a whiteboard provided by the host, Accessed in mutual exclusion

The Problem: security

Security issues
1. Harmful agent (presence of malicious processes) -- acute in unregulated non-cooperative settings (Internet)
2. Harmful host (presence at a network site of harmful stationary Process) -- exists also in regulated environment where agents cooperate (hardware or software failure)
The Problem

Identify the **harmful** host in the network

Harmful host (**black hole**): stationary process that disposes of visiting agents upon their arrival, leaving **no observable trace** of such a destruction......

The goal: **find and report the location of the black hole**

i.e., at least one agent must survive and the location of the black hole must be known

Previous Results - Protecting Agents Against Host Attacks

**Systems approach**
- Hohl '98, '00
- Sander, Tschudin '98
- Vitek, Castagna '99

**Algorithmic approach: DFPS**
- BH search in rings (DISC 01)
- BH search in arbitrary networks (PODC 02)
  1. different bounds depending on the amount of topological information available to the agents

Major Players - **Agents**

Can move from node to node
Have computing capabilities
Have bounded storage
Have the same behavior
Take unpredictable amount of time in computing and moving
Communicate through whiteboards, access through mutual exclusion

Major Players - **Black Hole**

Stationary process located at a node
Destroys any agent arriving at that node
Does not leave any trace of destruction
Its location is unknown to the agents
Solving the Problem

The Black Hole Search (BHS) is solved if:

1. At least one agent survives
2. The location of the black hole is known
   a) All the surviving agents know the location of the black hole or
   b) The neighborhood of the black hole is marked so that no other agent will enter it

Basic Limits

- It is impossible to find the black hole if the size of the network is unknown to the agents
- It is impossible to verify whether or not there is a black hole
- The BHS problem can be solved only if the network is biconnected

We assume n is known, there is a black hole and the network is biconnected

Basic Limits on Size

Since one agent may immediately wander into the black hole,

Obs1 At least ?? agents are needed

Complexity Measures

Number of Agents - size
Number of Moves - cost

Worst case, with respect to
- the network
- location of the black hole
- asynchrony
Basic Limits on Size

Since one agent may immediately wander into the black hole,

**Obs1** At least two agents are needed

Furthermore,

**Obs2** With full topological knowledge, two agents suffice
Major Players - Agents and Black Hole

Can move from node to node

Have computing capabilities

Have bounded storage \((O(\log n))\) bits

Have the same behavior

Asynchronous: unpredictable time in computing and moving

Communicate through whiteboards (at each node -- mutual exclusion), \(O(\log n)\) bits

Stationary process located at a node

Destroys any agent arriving at that node

Does not leave any trace of destruction

Its location is unknown to the agents

Terminology

Unexplored: no agent has been sent/received via this port
**Terminology**

Unexplored: no agent has been sent/received via this port

Active: an agent has been sent through it, but no agent has been received through it

**Terminology**

Unexplored: no agent has been sent/received via this port

Active: an agent has been sent through it, but no agent has been received through it

**Terminology**

Unexplored: no agent has been sent/received via this port

Active: an agent has been sent through it, but no agent has been received through it

**Terminology**

Unexplored: no agent has been sent/received via this port

Active: an agent has been sent through it, but no agent has been received through it

Explored: an agent has been received via this port
Terminology

A port of a node can be:
- **unexplored** (no agent has been sent/received via this port)
- **explored** (an agent has been received via this port)
- **active** (an agent has been sent through it, but no agent has been received through it)

We will require that any **active** port not leading to the **BH** be made **explored** as soon as possible. This is achieved with **Cautious Walk**.

Cautious Walk Technique

Whenever an agent $a$ leaves a node $u$ through an **unexplored** port $p$ (transforming it into **active**), upon its arrival to the node $v$, and before proceeding somewhere else, $a$ returns to $u$ (transforming that port into **explored**).

Cautious Walk Technique

Example for a line:

- Used in all our algorithms
- Limits the number of agents entering the black hole
- Any size-optimal algorithm must use Cautious Walk
Each node has:

- bounded amount of storage (whiteboard)
  \( O(\log n) \) bits

(if the labeling is consistent, the ring has orientation, otherwise it is unoriented)

---

**Fact 1**: impossible distinguish between slow link and black hole

---

**Q**: is it possible to determine (with explicit termination) whether or not there is a black hole?
Fact 1: impossible distinguish between slow link and black hole

Obs 1.a: it is impossible to determine (with explicit termination) whether or not there is a black hole

Knowledge of $n$?

Q: assume agents anonymous and co-located (they all start from the same node) → what can happen?
Fact2: agents anonymous and co-located behave as one agent

Obs2: it is impossible to locate BH if the agents are both anonymous and co-located

Agents \( r \) and \( l \), starting from the home base always move using cautious walk.

Proceed in phases.

At phase \( i \)

- divide the unexplored area in two contiguous disjoint parts of almost equal size

- agents explore (using cautious walk) the different parts
- one of them (say, r) will successfully terminate its part
  (--- the black hole is on the other side ! ---)
  and comes back

  (--- the other agent could be dead
  or simply slow ---)

- r follows the safe links already
  traveled by l until it reaches the
  node with the active link (the last
  safe node).

- r calculates the size of the still
  unexplored area. If it consists of
  only one node ...that's the black hole
  (--- l is dead ---)

- If not, r enters the next phase,
  leaving a msg for l
  (in case it is not dead) ...  

- ... and r goes back to
  traverse half of the unexplored
  area on the other side.

- If l is not dead it will return, and find r’s message;
  it will start executing the new phase (exploring its
  own assigned half)

---

**Co-located Agents in the Ring**

Number of moves: $2n \log n + O(n)$

**Theorem:** The algorithm is size and cost optimal

<table>
<thead>
<tr>
<th></th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>$(n-1) \log(n-1) + O(n)$</td>
<td>$2n \log n + O(n)$</td>
</tr>
</tbody>
</table>
Dispersed Anonymous Agents: Lower Bounds

Theorem: At least three agents are needed in unoriented rings

Theorem: The cost of locating the black hole in the oriented ring (and, thus, also in unoriented ring) is at least $\Omega(n \log n)$

Dispersed Anonymous Agents: Upper Bound (oriented)

A cost-optimal algorithm for any size $k \geq 2$
(size-optimal when $k = 2$)
The algorithm is composed by three parts: *pairing, elimination, resolution*
Main idea: form pairs of agents, have the pairs search for the black hole

- Form pairs of Agents
- Reduce the number of pairs to a SINGLE PAIR
- Apply the Algorithm for two co-located agents

Dispersed Anonymous Agents: Upper Bound (oriented)

Pairing Algorithm

- Move clockwise (cautious walk) until find a node visited by another agent
- Chase that agent until find
  1) either a node visited by two agents
  2) or the last safe node marked by the chased agent
In case 1) become ALONE (passive status)
In case 2) leave a mark JOIN_ME, form a pair and become PAIRED-LEFT
- Encountering a mark JOIN_ME, clear the mark and become PAIRED-RIGHT
- Meeting a paired agent, become ALONE (passive status)

Dispersed Anonymous Agents: Upper Bound (oriented)

When an agent is paired, it starts the

Elimination Algorithm

Election-like process
Only one pair is selected

(notice that both nodes and agents are anonymous! ....still a pair will be uniquely identified)
**Theorem:** The black hole can be found in \(O(n \log n)\) cost with \(k \geq 2\) agents. This is size and cost optimal.

**Theorem:** The black hole can be found in \(O(n \log n)\) cost with \(k \geq 3\) agents. This is size and cost optimal.

<table>
<thead>
<tr>
<th></th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>oriented</td>
<td>2</td>
<td>oriented</td>
</tr>
<tr>
<td>unoriented</td>
<td>3</td>
<td>unoriented</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>(\Omega(n \log n))</td>
<td>(O(n \log n))</td>
</tr>
</tbody>
</table>

**Other special networks**

- **Complete Knowledge:** two agents suffice to locate BH in \(\Theta(n \log n)\) moves

**Question:** can \(O(n \log n)\) be improved for special topologies?

**No** for rings, where \(\Omega(n \log n)\) applies for any two agents solution.

**Fortunately,**

this answer does not generalize!!!!

We presented technique for 2 agents BH search with complete knowledge, which results in \(O(n)\)

BH location in frequently used interconnection networks:

- Hypercubes*
- Cube-related networks
- Star graphs
- Multidimensional tori* and meshes*

* provided that \(\text{diam}(G) \in O(n/ \log n)\)
Setting

Simple, unoriented 2-connected n-node graph $G$

All agents start at the same node $h$ (home base)

Structural information available to agents:
- Complete Topological Knowledge

Traversal Pair

The key of the technique is the notion of

**Traversal Pair (TP)**

$O_G$: arbitrary fixed total ordering $v_1,v_2,...,v_n$ of the nodes

$\pi_a,\pi_b$: walks in $G$ starting from $v_1$ and $v_n$, and visiting the nodes in the order $v_1,v_2,...,v_n$ and $v_n,v_{n-1},...,v_1$ respectively.

Then, $(\pi_a,\pi_b)$ is a $v_1$-$v_n$ TP of $G$ w.r.t. $O_G$

**Traversal Pair: size and radius**

Let $\pi=(\pi_a,\pi_b)$ be a $u$-$v$ TP of $G$:

- **Size** of $\pi$, $s_\pi(G)=\max(|\pi_a|,|\pi_b|)$
- **Radius** of $\pi$, $r_\pi(G)=\max_{i=1\ldots n}(r(G_{v_i}),r(G_{v_{n-i}}))$

with $r(G_{v_i})$ the depth of the BFS tree of the subgraph of $G$ induced by $v_1,v_2,...,v_i$ rooted at $v_i$
Consider a mesh with at least one side of even length.

Example: 2D Mesh

Computing the walks...

...and the corresponding BFS.
Consider a mesh with at least one side of even length

Computing the walks...

...and the corresponding BFS

Consider a mesh with at least one side of even length

Computing the walks...

...and the corresponding BFS

Consider a mesh with both sides of odd length

$s_\pi(\mathcal{G}) = |\pi_a| = |\pi_b| = 11$

Computing the walks...

$r_\pi(\mathcal{G}) = 4$

...and the corresponding BFS
Consider a mesh with both sides of odd length.

Algorithm SplitWork

IDEAS????

Divide and Conquer:
Algorithm SplitWork

- divide the unexplored area in two contiguous disjoint parts of almost equal size (local computation, always possible, because the network is biconnected)
Divide and Conquer:

Algorithm SplitWork

- divide the unexplored area in two contiguous disjoint parts of almost equal size
- agents explore (using cautious walk) the different parts

- one of them (say, \(a\)) will successfully terminate its part (the black hole is on the other side); it comes back, using Sorthest Safe Path
- then follows the trace of \(b\) until it comes to the last safe node \(v\) explored by \(b\)
- if there is single unexplored node remaining, the black hole is there

Algorithm SplitWork (cont.)

- otherwise \(a\) divides the remaining unexplored area into two parts of almost equal size and leaves a message for \(b\) about that
Algorithm SplitWork (cont.)

- otherwise \( a \) divides the remaining unexplored area into two parts of almost equal size and leaves a message for \( b \) about that
- then \( a \) leaves to explore its part using the SSP
- when (if) \( b \) returns to \( v \), it finds the message
- and proceeds by exploring its part

The process is repeated at most \( \log(n) \) times,

Total cost is \( O(s_n(G) + r_n(G) \log n) \)
### Results for particular topologies

<table>
<thead>
<tr>
<th>Topology</th>
<th>Size</th>
<th>Radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>d-dimensional hypercube*</td>
<td>$4n$</td>
<td>$\text{diam}(G)$</td>
</tr>
<tr>
<td>d-dimensional mesh*</td>
<td>$4n$</td>
<td>$\text{diam}(G)$</td>
</tr>
<tr>
<td>d-dimensional torus*</td>
<td>$4n$</td>
<td>$\text{diam}(G)$</td>
</tr>
<tr>
<td>Wrapped Butterfly, WBF(d)</td>
<td>$4n$</td>
<td>$O(d^2)$</td>
</tr>
<tr>
<td>Cube Connected Cycles, CCC(d)</td>
<td>$4n$</td>
<td>$O(d^2)$</td>
</tr>
<tr>
<td>Star graph</td>
<td>$4n$</td>
<td>$2^{d+1} + 1$</td>
</tr>
</tbody>
</table>

* provided that $\text{diam}(G) \in O(n/ \log n)$

**Theorem:** Two agents can locate the BH in $O(n)$ moves in all of the above topologies*.

### Relaxing complete topological knowledge

Until now, we assumed to have not only the

**Topological awareness:**
- Knowledge of the class of the network
- Knowledge of the labelling of the ports

But the stronger

**Complete topological knowledge:**
- Topological awareness, plus
  - Size of the network
  - Location of the source

### Universal protocols

i.e. working for arbitrary biconnected networks

<table>
<thead>
<tr>
<th>Structural Information</th>
<th>Size</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topological Ignorance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta \leq n-4$</td>
<td></td>
<td>$\Theta(n^2)$</td>
</tr>
<tr>
<td>$n-3 \leq \Delta$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sense of Direction</td>
<td>2</td>
<td>$\Theta(n^2)$</td>
</tr>
<tr>
<td>Full Topological Knowledge</td>
<td>2</td>
<td>$\Theta(n \log n)$</td>
</tr>
</tbody>
</table>

$\Delta$: maximum degree, excluding the home base

* - assuming optimal size
Conclusions

The existence of \textbf{BH} is not uncommon!!

Undetectable crash failure of a site in an asynchronous network = \textbf{Black Hole}

\textbf{Black Hole} search = locating an undetectable crash failure