Distributed Algorithm on AHCv2: Waves: Tarry's Traversal and Tree Algorithms, Release V1.0.0

Berkay Şahin

May 15, 2024

Outline

1. Introduction
2. Tarry's Algorithm
3. Tree Algorithm
4. Implementation and Methodology
5. Results
6. Discussion
7. Conclusion

Introduction

Context:

- Introduction to the challenges and importance of efficient traversal algorithms in distributed systems.
- Essential for managing complex networks involved in distributed systems.

Problem Statement

Problem Statement:

- The need for effective network traversal and spanning tree construction.
- Importance of designing protocols that minimize communication overhead.
- Ensuring completion even in dynamic network topologies.

Significance and Necessity

Significance and Necessity:

- Why these algorithms are crucial for system robustness and performance.
- Impact on efficient resource management and fault diagnosis.

Overview of Tarry's Algorithm

Tarry's Traversal Algorithm:

- Designed for undirected graphs.
- Ensures every node is visited exactly once.

Principles and Mechanism

Principles and Mechanism:

- Simple token-passing mechanism.
- Each channel is visited twice to ensure completeness.

Pseudocode of Tarry's Algorithm

Pseudocode:

```
if node is initiator then
    send token to an arbitrary neighbor;
end
while token is received from neighbor do
    mark self as visited;
    if there are unvisited neighbors then
        choose an unvisited neighbor and send the token;
    else
        return token to sender;
    end
end
```

Algorithm 1: Tarry's Algorithm

Benefits and Evaluation

Benefits and Evaluation:

- Ensures complete network coverage.
- Minimizes message overhead by avoiding revisits.

Overview of Tree Algorithm

Tree Algorithm:

- Designed for acyclic networks.
- Focuses on quick spanning tree formation.

Principles and Mechanism

Principles and Mechanism:

- Waits for messages from all neighbors except one.
- Decision making by exactly two nodes in the network.

Pseudocode of Tree Algorithm

Pseudocode:

```
foreach node in network do
    Listen for messages from all neighbors except one;
    if messages received from all but one neighbor then
        Select the neighbor with no message as parent;
        Send message to selected parent;
    end
    if message received from parent then
        Finalize decision;
    end
end
```

Algorithm 2: Tree Algorithm

Benefits and Evaluation

Benefits and Evaluation:

- Efficient message utilization.
- Quick decision-making process.

Simulation Setup

Simulation Setup:

- Description of the AHCv2 simulation environment.
- Details on network topologies used: linear, tree, and random graphs.

Methodological Approach

Methodological Approach:

- Nodes initialized with specific algorithms.
- Monitoring and capturing efficiency and coverage metrics.

Theoretical Results

Theoretical Results:

- Expected message usage and network traversal completeness for Tarry's Algorithm.
- Speed and message overhead implications for the Tree Algorithm.

Implications of Findings

Implications of Findings:

- Suitability of Tarry's Algorithm for detailed network exploration.
- Applicability of the Tree Algorithm for rapid deployment in structured networks.

Practical Recommendations

Practical Recommendations:

- Application scenarios for each algorithm based on their strengths.

Conclusion

Summary of Findings:

- Comparative analysis of Tarry's and the Tree Algorithms.
- Future research directions for reducing overhead and improving adaptability.

Final Thoughts

Final Thoughts:

- Necessity of empirical validation of theoretical predictions.
- Importance of continued research in distributed systems traversal algorithms.

References

Questions

Thank you! Any questions?