CSCI.652.01 - Distributed Systems Project Report

Deepali Kamat

Department of Computer Science

Rochester Institute of Technology

Summer 2015

I. Introduction

Target enclosure by autonomous robots is very essential and helpful as far as practical applications are concerned. Disaster surveillance, resource allocation, managing large computation load can be performed more easily using a distributed architecture rather than using a single unit. Scalability of robots or systems is of great significance in a distributed network as multiple units working together are more efficient and robust against failure or breakdown.

This project focuses on designing a distributed model for a set of ’k’ robots on a two dimensional grid such that; given a target, the robots nearest to the target enclose it from the four cardinal directions. The target is a location on the grid and has no computing or communication prowess. Additionally, the robots can only move in either horizontal or vertical directions. The project discusses three possible cases:

1. The location of the robots as well as that of the target is known beforehand;

2. The location of the robots are known but the target location is unknown; and

3. There are one or more malicious robots present in the network.

The aim of the project is to take into consideration each of these cases and find a way to complete the task of target enclosure successfully. The performance of this system is verified through computer simulation.

II. Approach Used [Outline of Project]

This project aims at creating a distributed system for enclosing a target location from the four fundamental directions using coordination and agreement algorithms. The target is enclosed by occupying the north, east, west and south directions surrounding the target. The project involves a 2 dimensional grid of length ’l’ and breadth ’b’ on which ’k’ robots will be randomly placed. Each incoming robot is considered to be independent and associated with a process that would run on the CPU. The robots communicate with their peers directly, without the help of a central command. The target spot is a fixed location on the grid and has no communication capability. The moving robots or peers can detect the presence of the target location only when it is in one of the four neighboring cells of the target. Each bot takes one unit of time to move from one block to another.

The project commences with the initialization of the 2-Dimensional Grid. The target location is randomly placed on the network. The target location is made known to all the systems in the network by the initiator system when joining. The peers or the subsequent robots join the network through the Initiator class. For each joining robot, their Manhattan distance from the target is calculated. During this process the IP-addresses and the Manhattan distances of all the robots existing in the network are stored by the joining robot and the existing robots also store the IP-address and the Manhattan distance of the robot joining the network. This way every robot knows the location of all the other robots in the network.

Consider the first case wherein the location of the target is known. The target location is made known to all the systems in the network by the initiator system. Once all the robots are in the system, and have addresses of all its peers, the distances from each bot to the target location are stored with respect to each of the four surrounding locations of the target position. These lists are then sorted in ascending order. In order to decide the surrounding robots for the target location, the individual collections are checked. The north side of the target gets the first entry by default. For all the other locations, a check is performed in order to see if that bot has already been assigned to any of the surrounding locations.The shortest(next) distance bot is assigned to the next location. The same is done for the remaining two locations surrounding the target. Once, robots are assigned for each of the four directions surrounding the target, the movement of systems is decided using consensus, where all the robots need to agree on the timing and the path taken by each of the robot to avoid collision. Once consensus is reached, the four robots move towards their respective assigned locations and the target is enclosed.
In the second case, where the location of the target is not known, the robots are strategically placed on every alternate row in such a manner that the robot is able to scan four of its adjacent (North, East, West and South) locations. The robots search concurrently and once the target is detected, the robot that detected the location, stops the search and informs every other robot about presence of the target. then, the same procedure as the first case is performed.
In the third case involving the malicious robots, select a random robot and consider it to be malicious. If the robot considered claims to have found the target then on that row send a random robot to perform a similar search; if no target was found in the initial search, assign another random robot an the random block to detect target presence. Mark both of these robots to be malicious. Pick a random robot that might not be malicious. Run that robot on the same row where the initial claim about target location was made. When the search ends, the two robots pointing at the same location are not malicious and the one pointing otherwise is the malicious bot.

A. Algorithm

1. START
2. Initialize 2-D Grid;
3. Place target on Grid;
4. Label 1:
   if(target location known){

      (a) while(ipAddress){

i. add robot to network;
ii. calculate distance between robot and target;
iii. for(each robot added){

A. update robot address and distance in each robot present;
B. update the addresses and distances of the robots present in own list;

}

}

(b) group Manhattan distances based on the 4 surrounding locations around target;

(c) sort the distances for the 4 groups in ascending order;

(d) check each group and assign the shortest distances in each group to the closest bot by checking if that bot has been assigned to any location before;

(e) selected bots undergo consensus agreement

(f) enclose target

}

5. else{

(a) place robots on grid strategically;

(b) make all the robots search grid simultaneously;

(c) if(target found){
• stop search, inform peers; • go to step 4 (Label 1);

}

6. Case 3 : Malicious Robot

(a) Select random robot

(b) consider the robot is the malicious one

(c) if target found{

• select random(target not found, wrong block number);

(a random robot is selected and run on the row where target was found)

}

(d) else{

• random(block number);
(a random robot is assigned to search the block)

}

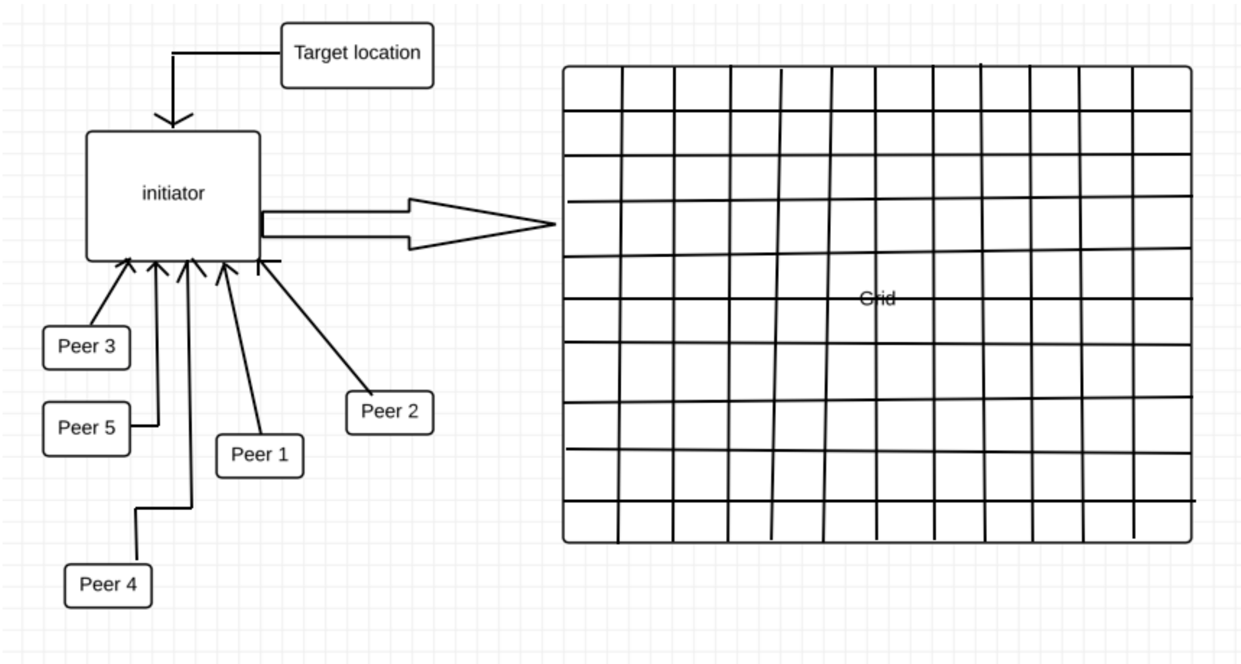
(e) Mark the two robots to be malicious;

(f) pick one random robot which is not possibly malicious;

(g) run them to check the row where the target was detected;

(h) the two robots pointing at the same block are not malicious;

B. Block Diagram



The bots join the network through the initiator

The initiator places the target location.

The robots are placed; Manhattan distances are calculated.

The four ideal robots are chosen, consensus is obtained.

The robots are ordered to enclose the target;

Target is enclosed.

III. Results

The simulation project for the robot attack was implemented by incorporating remote method invocation using the Java programming language for the three cases. Algorithms were used such that the robots reach consensus before approaching towards the target. The system has been optimized to enclose target using minimum time by assigning only the closest bots to enclose the target.

IV. Conclusions

The target on the grid was enclosed by implementing the concept of distributed systems. It is observed that applying a distributed approach helps achieve the goals faster and the amount of work to be done is distributed among all the participating peers. A distributed model for a set of ’k’ robots on a two dimensional grid was designed and the nearest robots enclosed it from the four fundamental directions. The performance of the system was verified through computer simulation.

V. References

[1] http://www.cs.utexas.edu/users/AustinVilla/legged/papers/consensus.pdf

[2] http://link.springer.com/chapter/10.1007%2F978-3-642-33932-5_75#page-1

[3] https://courses.cs.washington.edu/courses/cse452/13sp/resources/Intro.pdf