ROBOTRAK: A Centralized Real-time Monitoring, Control, and Coordination System for Robot Swarms

Ming Li[1], Anthony Alvarez[1], Francesco De Pellegrini[2], B. Prabhakaran[3], Imrich Chlamtac[2]
[1] Department of Computer Science, California State University
Fresno, CA 93740, USA. {mingli, anthonyalvarez}@csufresno.edu
[2] Create-Net, Via Solteri 38
38100 Trento, Italy. {francesco.depellegrini, chlamtac}@create-net.org
[3] Department of Computer Science, The University of Texas at Dallas
Richardson, TX 75083, USA. praba@utdallas.edu

Abstract— **Robotic swarm has been a hot topic in recent years. In robotic swarms, a team of network enabled bots are dispatched to some areas to fulfill certain tasks, such as military actions and chemical substance tracking. However, how to monitor, control, and coordinate swarms in a real-time manner such that robots can collaborate efficiently and effectively is a challenging issue. In this paper, a real-time software on enforcing the monitoring, control, and coordination of intelligent robotic swarms (ROBOTRAK) has been developed. All the designed features and functionalities have been successfully implemented with user friendly graphical user interfaces (GUI) and extensively tested in various scenarios such as multiple robots and multiple servers running the ROBOTRAK software. Results show that ROBOTRAK is user friendly and can help monitor, control, and coordinate robotic swarms timely, effectively, and efficiently.**

Keywords-robot swarm, software development, security, GUI

I. INTRODUCTION

Robot swarms have gained lots of attention in recent years. Despite of many complicated artificial intelligent algorithms proposed to improve the correctness and efficiency of decision making of a single robot, using a team of robots has many advantages. Although several softwares [1][5][6] have been developed, they are either designed to monitor a single robot, or does not perform control and coordination, or are non-real time and used just for the purpose of simulation. To meet the software needs resulting from the increasing deployment of network enabled robot swarms, we have designed and developed ROBOTRAK, a secure software for monitoring, control, and coordination of intelligent robotic swarms. Using TCP connections through the wireless medium, the ROBOTRAK server can exchange information with the robotic swarm reliably and continuously. For *monitoring purpose*, all the bots collect and report wireless signal strength, interference, neighboring bots list, and location information. For *control purpose*, a new robot can be dispatched to join a swarm or an existing robot in a swarm can be guided to move to a specific destination. For *coordination purpose*, the server maintains the connectivity of the swarm under various situations. Furthermore, in order to maintain network privacy and avoid outsider intrusion, *multi-security levels* and a *dynamic password technique* were implemented.

The major advantages of ROBOTRAK are:

- Flexibility with comprehensive features on robot monitoring, control, and coordination;
- User friendliness with nice graphical user interface;
- Multi-level security at both application layer and transport layer;
- Robustness with the implementation of reliable socket communication.

ROBOTRAK can run on any server with Internet connection and requires the robots in a target swarm to be equipped with wireless adaptors (such as WiFi) and global positioning system (GPS). All the messages can be transmitted/received through typical wired/wireless multi-hop network connections. The software was implemented with *Microsoft Visual Basic* and can be modified to provide web-based versions. All the designed features and functionalities have been implemented with user friendly graphical user interfaces (GUI) and extensively tested in various scenarios such as multiple robots and multiple instances of running ROBOTRAK software. Results show that ROBOTRAK is user friendly and can help monitor, control, and coordinate robotic swarms timely, effectively, and efficiently.

II. ASSUMPTIONS AND MAJOR SOFTWARE MODULES

In this section, we describe the three major components of ROBOTRAK: *Monitoring module*, *Control module*, *and Coordination module*. Some basic assumptions are made on the design and development of the ROBOTRAK software. From the networking aspect, we assume the following:

- All robots are equipped with wireless adaptors (such as WiFi) and can be accessed by the central server running ROBOTRAK through Internet.
- Every robot is assigned a unique IP address and port number for the communication with the software. The software can connect to any robot by sending packets to the corresponding IP address and port number.
- All robots are equipped with global positioning system (GPS) so that they can report their locations to the ROBOTRAK server.

ROBOCOMM 2007, 14th – 16th Oct 2007, Athens, Greece.

ISBN 978-963-9799-08-0
DOI 10.4108/ICST.ROBOCOMM2007. 2171

- Wireless access is ubiquitously available for at least one mobile robot. In case some robots cannot access any wireless access point, they can forward this information to the one that is connected to the Internet wirelessly.

From the robot intelligence aspect, we assume the following:

- All robots have the necessary intelligence to find a path from one location to another by itself.
- There is a certain basic coordination scheme (such as [2][3][4]) running to maintain the connectivity of the swarm under normal situation. The purpose of ROBOTRACK is to provide assistance in unusual situations.
- Each swarm has one leader and at least one member for coordination purpose. This leader can be either specified or determined by some voting algorithms.

It should be noted that for a robot to be accessed from the Internet through access points and to communicate with other robots in the same swarm simultaneously require a minimum of two wireless adaptors. One adaptor is configured to operate in *infrastructure* mode and connect with a nearby access point and another adaptor is configured to operate in *ad hoc* mode and connect with all robots within the transmission range.

A. Monitoring module

The monitoring module basically analyzes the messages from each robot and decides the desired information the administrator intends to track such as physical location. Also, connectivity within the swarm is critical for the performance of a swarm. Thus, each robot should report its neighboring robots within the same swarm. This neighboring information can be obtained through communication among mobile robots in a self-organized wireless ad hoc network. In summary, each robot will report following basic information to the server for monitoring purpose:

- *Physical location*: The coordinates obtained from the GPS system is sent from each robot to the server.
- *Neighboring list*: Whenever a robot receives or overhears messages from another robot in the same swarm, it will add the robot to its neighboring list. This list is periodically refreshed to remove old neighbors and add new neighbors.

It should be noted that more information may be required for monitoring, depending on the specific application. The applications of ROBOTRACK has some analogy with the case of wireless Internet Service Providers (ISPs) or network administrators of commercial and military wireless networks. There, the most important issue is how to place mesh routers such that all targeted areas are covered sufficiently and efficiently, i.e., all areas are covered with good signal strength with minimum number of mesh routers. Typically, some areas may experience high interference and some areas may experience weak or no signal, due to inappropriate installation of many mesh routers. Furthermore, it is possible that an outsider may install their mesh routers and thus has significant interference with the commercial wireless network. In the case of ROBOTRACK, a robot team can be dispatched to report the *Wi-Fi signal strength* and *SSID* of all nearby Wi-Fi access points for potential intrusion detection.

B. Control module

The control module is designed to only provide assistance to the swarm in certain unusual situations. As mentioned earlier, the control from the server is not supposed to replace the basic operations of each individual robot or the whole swarm, but to help them. In ROBOTRAK, we consider three situations where outside assistance is desirable:

- *Robot Joining*: dispatching a new robot to join the swarm by providing the destination. While the robot is moving towards the team, the center location of swarm is periodically updated for guidance.
- *Robot Leaving*: requesting one robot in a swarm to leave the swarm. The robot may go back to storage, or go to a location specified by the server.
- *Swarm Movement*: requesting the whole swarm to move to another location to perform some other tasks. There may be several reasons for this. The administrator may identify certain dangerous situations at the current location and thus wants the robots to move immediately. Or the administrator may have another urgent task for the swarm to fulfill.

C. Coordination module

Similar to the control module, the coordination module is also designed to only provide minimum assistance for swarm connectivity. The basic coordination is still performed by the swarm itself independently with certain intelligent algorithms. However, with the convenient monitoring module, ROBOTRAK can easily identify certain swarm partition and acts promptly. In ROBOTRAK, we consider the following three representative situations:

- *Isolation of a regular robot*: If one robot is isolated from the rest of the team due to various reasons such as entering a building or the other side of an obstacle, it will report its neighbor list as empty to the server. Based on this information, the server can quickly direct the robot to move towards the swarm with higher speed.
- *Isolation of a leader robot*: We assume that the leader should guide the whole swarm to move. However, in case the leader gets isolated due to similar reasons for a regular robot, the server can request all other robots to move to the leader and maintain the swarm connectivity.
- *Swarm partition*: In the worst case, swarm may be divided to multiple isolated partitions and cannot perform any meaningful task for quite some time.

We can design a simple algorithm for fixing swarm partitioning. In this algorithm, it is assumed that every robot has a unique identifier from 1 to *N*. When the server receives the neighboring information from each robot, it will run the algorithm. Initially, there are N partitions with robot i included partition i. Then, two partitions P_i and P_j are merged to P_i if $i < j$ and robot j is in the neighboring list of i. Thereafter, this merging will continue until all partitions have been examined. Finally, if there is only one partition that includes all robots in the swarm, the server knows that the swarm is connected. Otherwise, the swarm identifies a major partition that includes the maximum number of robots, get its center location, and requests robots in all other partitions to move towards that location.

III. Security Enforcements

There are two possible security issues with robot swarm monitoring, control, and coordination. On the one hand, the ROBOTRAK administrator should posses the appropriate permission for certain functionalities. On the other hand, it is important to identify malicious outsiders who may steal the passwords and try to control, hack and eventually damage the swarm. To address these issues, we enforce three security policies:

A. *Multi-level password checking*

We classify the access levels of administrators to *general*, *high*, *highest*, which are determined by the passwords provided. Then, we restrict their permissions as follows.

- *General*: under this access level, the software can *only monitor* the basic information of the swarm and cannot issue control or coordination commands to any robot;
- *High*: under this access level, the software can *monitor and coordinate* the robot swarm, and *watch the image/video* from each individual robot, if applicable;
- *Highest*: under this access level, the software can perform *all functionalities including robot control.*

With this multi-level password checking, we can avoid unauthorized interruption to the swarm. This policy can be changed according to the application and user requirements.

B. *Dynamic password update*

One special situation is that an outside intruder may get the password somewhere and try to maliciously connect and hack a robot that has been connected with an authorized server running ROBOTRAK. In this case, we adopt a simple approach to avoid simultaneous connection of an individual robot from two or more administrators with highest access levels. Each time after an administrator with highest access level gets connected to a robot with a predefined password, it will assign a new password to the robot periodically (every minute). So, when a outsider attempts to connect to the same robot with the predefined password, the robot will reject the connection and is still safely under the control of the authorized administrator.

Of course, it is possible that a hacker may get connected to the swarm with highest access level and prevents the authorized user from connecting. In this case, the authorized user will know the situation and take certain actions. However, given the assumption that ROBOTRAK is running continuously at the authorized server, this does not occur frequently.

C. *Server information reporting*

To further enforce security, it is still desirable to prevent those hackers who are not trying to control the swarm, but simply monitor robots position. To address this potential loop, each server running the software will require every connected robot to report all servers it is communicating with. Then, authorized users can identify possible intruders by checking the reported servers.

IV. Feature Demonstration

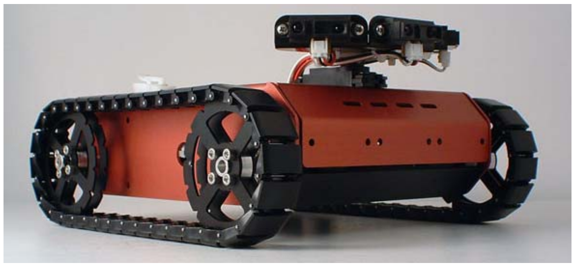

Figure 1. The Traxster Robot.

In this section, we demonstrate the features of ROBOTRAK through distributed simulation. In this simulation, a robot agent program is running to emulate each individual robot. We will test ROBOTRAK with Traxster Robot ([7] and Figure 1) that will run on Window CE system and is equipped with multiple Wi-Fi adaptors and GPS receivers. However, since these agent programs have exactly the same message format and operating system requirements as real robots, the simulation does provide an accurate demonstration of the features of ROBOTRAK.

A. *Swarm monitoring*

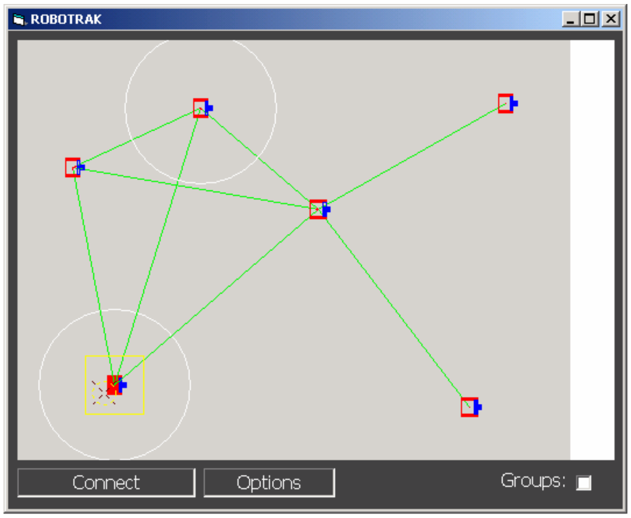

Figure 2. Illustration of swarm monitoring

In this section, we illustrate the swarm monitoring feature. Figure 2 shows the monitored topology and sonar ranges of selected robots. We can see that the swarm is strongly connected. The neighborhood is established when two robots are within 250 meters from each other. In real scenarios, each robot can send broadcast message to discover its actual neighbors and report the list to the administrator for displaying the topology information. Also, the sonar can detect the distance from obstacles such as walls and provide specific information about the environment.

B. *Swarm control*

In this subsection, we demonstrate two types of control: *robot movement* and *swarm movement*. Figure 3 shows the robot movement where each individual robot is separately selected and directed to a specific destination. The tails shows the trail of the robot in the last a few seconds, which is inspired by SwisTrack [1]. When the robot arrives the destination, it will stop and the trail gradually disappears. Figure 3 also shows the swarm movement. When the “Groups” option is checked, the swarm can be selected by entering “CTRL+A” command from the keyboard and directed to move to around the same destination by right-clicking the mouse at the specific location on the screen.

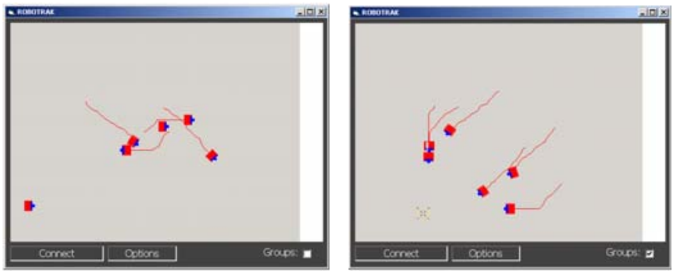

Figure 3. Illustration of swarm control. Left: robot movement; Right: swarm movement

C. *Swarm coordination*

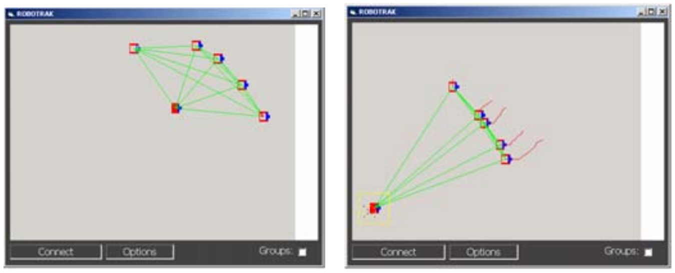

Figure 4. Leader (dark color) movement coordination. Left: before movement; Right: after movement.

To demonstrate the feature of swarm coordination, we consider two situations: (i) *Leader movement* where all other robots need to follow the leader to avoid swarm partitioning. (ii) *Member movement* where a specific robot cannot move two far away from the rest of the swarm. Figure 4 illustrates the leader (dark colored) movement. Initially, all the swarms are within the range of the leader. When the leader moves towards the left bottom corner, all other robots follow the leader and moves towards the same direction to maintain the swarm connectivity. Figure 5 illustrates the coordination on a swarm member movement. It can be seen that as the robot moves towards the left, more and more other robots get disconnected from it. However, after it moves out of the range of all other robots, it is “dragged” back to a destination within the transmission range of the closest robot by the software with the coordination algorithm, in order to maintain connectivity of the swarm. More demonstrations of the swarm coordination are omitted due to limited space.

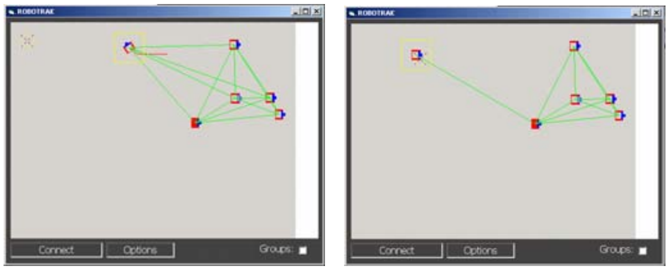

Figure 5. Swarm member movement coordination. Left: before movement; Right: after movement.

V. CONCLUSTION

How to monitor, control, and coordinate robot swarms in real time is a challenging and imperative issue. In this paper, we introduced our work on the design and development of a real-time software, ROBOTRAK, to perform this task. The software can effectively track the location, connectivity, and other important information of robots, provide useful assistance to robot swarms such as dispatching new robots to a swarm, and maintain the connectivity of the swarm. Several security measurements are enforced to make sure the software does not introduce unwanted outside intrusion. The software is user friendly, efficient, robust, and thus can be used in practice. In the future, we will work on loading the maps to the software to improve the visual effect of robot locations. Also, we will test ROBOTRAK extensively with real robot swarms for further improvements.

REFERENCES

[1] N. Correll, G. Sempo, Y. Lopez de Meneses, J. Halloy, J.-L. Deneubourg, and A. Martinoli, “SwisTrack: A Tracking Tool for Multi-Unit Robotic and Biological Systems”, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2185-2191, 2006.

[2] V. Trianni and M. Dorigo, “Self-organisation and communication in groups of simulated and physical robots,” Biological Cybernetics, 2006.

[3] Groß R., Bonani, M., Mondada, F. and Dorigo, M. (2006) Autonomous Self-assembly in Mobile Robotics. IEEE Transactions on Robotics. In press.

[4] Poduri, S. and Sukhatme, G. S. (2004). Constrained Coverage for Mobile Sensor Networks. IEEE International Conference on Robotics and Automation, New Orleans, LA, USA.

[5] MobileEyes: http://www.mobilerobots.com/MobileEyes.html

[6] James McLurkin, Jennifer Smith, James Frankel, David Sotkowitz, David Blau, Brian Schmidt. "Speaking Swarmish: Human-Robot Interface Design for Large Swarms of Autonomous Mobile Robots", AAAI Spring Symposium, March 28, 2006.

[7] Traxster Robot: http://www.roboticsconnection.com/pc-15-3-traxster-robot-kit.aspx