MOBILE ROBOT TRACKING IN WIRELESS SENSOR NETWORKS

Yoon Kah Leow
Electrical & Computer Engineering
The University of Arizona

Ying Shang
Electrical & Computer Engineering
Southern Illinois University Edwardsville

Abstract

With the rapid development of artificial intelligent technology, robotics became a popular undergraduate and graduate engineering course. Mobile robot tracking is one of the important components in robotic classes because it has many real-world applications, such as search and rescue operations, military surveillance, and tracking moving targets in industrial warehouses. This paper addresses the mobile robot tracking problem with the help of a wireless sensor network, which consists of a number of sensor motes to collect real-time data information and transmit data packages to the mobile robot. This paper proposes an intrusion detection algorithm based on light signal sensing and a counter-based routing protocol integrated with automatic repeat request. Both the intrusion detection algorithm and the routing protocol are implemented in a prototypical wireless sensor network consisting of MICA2 sensor motes. The experiment results show that the wireless sensor network is able to report the location of the robot promptly and accurately.

Introducation

There are several robotics classes offered at Southern Illinois University Edwardsville (SIUE), such as Robotics, Introduction to Artificial Intelligence, Human-Computer Interaction, and Mobile Robotics. Mobile Robotics class, in particular, is an interdisciplinary course collaboratively taught by faculties from computer science, electrical and computer engineering, industrial engineering, and mechanical engineering. In this class, students learn about the robotics' mechanical, electrical, and computational mechanisms, as well as the hands-on experiments.

In order to enrich the Mobile Robotics course, the mobile robot tracking problem becomes a very interesting topic to add on to the existing course materials because of its practical applications in search and rescue military surveillance, military surveillance, and tracking moving targets in industrial warehouses. It is often in the case of a practical situation whereby the human intervention is not desirable in the mobile tracking. This limitation is the motivation to discover new sophisticated methodologies to achieve the kind of versatility in such a complex situation. With great advances in the realms of micro-electromagnetic systems (MEMS), technology has rendered wireless sensor networks[8] a viable solution suitable for the mobile tracking problem.

In order to better study the mobile tracking problem in a wireless sensor network prior to the final integration with the mobile robotics class, an independent study on mobile robotics and a master project on wireless sensor networks were offered in the Department of Electrical and Computer Engineering in Fall 2008. Students studied the reliable communication strategies in sensor networks, software programming for the sensor motes, and path planning of the mobile robots. The end results of the independent study and the master project include an intrusion detection algorithm and a counter-based routing protocol, both implemented on an iRobot® Create robot[5] driving in a wireless sensor network. This sensor network consists of five MICA2 sensor motes from Crossbow Technology[1]. These sensors are pre-programmed with the application software built over the TinyOS operating system to perform the tracking algorithm based on the detection of light signals and to route data packets across the wireless sensor network. Students and faculties in the Department of Electrical and Computer

Engineering at SIUE reacted to the mobile robot tracking project positively. With the knowledge of sensor networks and mobile robotics, our graduates became more competitive on the engineering job market. Moreover, the established results on the mobile robot tracking in a wireless sensor network can be integrated with the Mobile Robotics course offered at SIUE in the future.

Wireless Sensor Network

A wireless sensor network consists of spatially distributed embedded devices that are capable of acting autonomously yet collaborating among each other in a concerted fashion in achieving a common goal. These devices are well-known for its' self-configurable nature and provides a high level of automation that can often replace human roles. Collaborating with a wireless sensor network allows pursuers to gain global visibility of the pursuit grounds even without the visual contact of the evader. For instance, in Figure 1, the pursuers have limited visibility without the wireless sensor network. On the other hand, in Figure 2, with the wireless sensor network, the pursuers are able to gain information of the terrain nature and the evader's location.

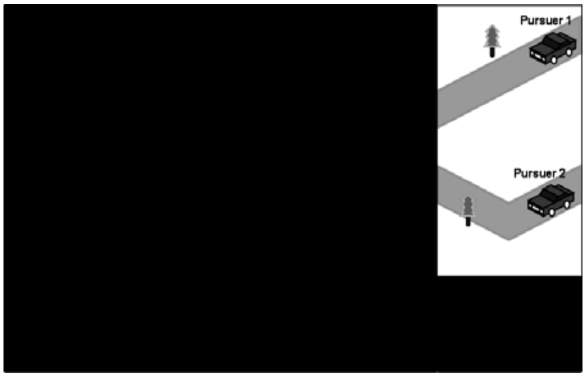


Figure 1: Pursuer visibility-
without the sensor network.

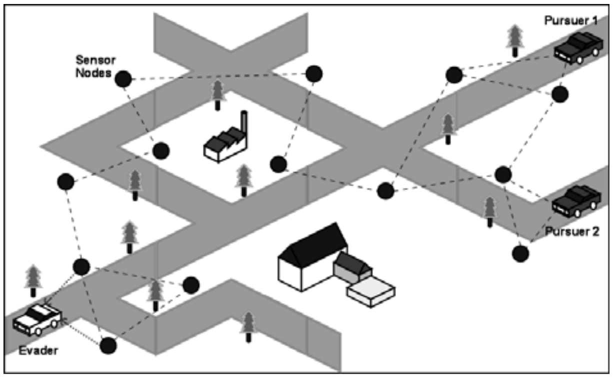


Figure 2: Pursuer visibility –
with the sensor network.

Intrusion Detection And Routing Protocol

In order to facilitate the research on the mobile tracking problem, we assume a statically deployed homogeneous sensor network and that data collision is negligible to interfere with the operation. This research problem can be broken down into the intrusion detection mechanism design and the data routing protocol design.

Intrusion Detection Mechanism

In the mobile tracking, sensor motes need to be able to detect intruders in the region. The challenge is the versatility for the sensor to differentiate between a positive detection and a false detection. Moreover, in a realistic situation, it is very likely that a sensor network will be deployed over a long period of time. The sensors need to realize the ever-changing surrounding environment, such as the light ambience and background noise, and adapt themselves constantly in order to provide accurate detections.

This paper focuses on the intrusion detection[3, 9], that is based on the changes of the light signals. On one hand, in order to handle the changing light ambience in the environment, the desired motion detection algorithm needs to tune itself to the background light ambience constantly. On the other hand, it is a requirement for the sensors to be deployed for a long period of time without any human intervention. Hence, the intrusion detection algorithm needs to be able to perform the fine-tuning process autonomously.

The light sensor is programmed to acquire light values at two frequencies 10 Hz and 0.67 Hz. The two frequencies are selected based on experimental calibrations after running exhaustive tests with a robot moving at a fixed velocity. In order to take care of energy constraints on these embedded devices, the faster frequency of 10 Hz is only used when a large difference in light ambience has been observed, that is, a motion is suspected within the sensor's sensing range. Hence, the motion detection algorithm will sense the light values based on the slower 0.67 Hz while operating under the normal light ambience conditions.

During the intrusion detection, a light value that is just retrieved will be compared with the previously retrieved light value. A motion is suspected if the difference between them is more than 1.0 count. If a motion is suspected, the algorithm will increase the light sampling frequency to 10 Hz and a second set of light value will be retrieved. Hence, this second set of light value will replace the current light value and will be compared with the previous set of light value. Positive motion detections will be reported if there are two consecutive differences in the light ambience that are more than 1.0 count. By buffering the light values, it gives allowance for slight changes in the light ambience throughout the monitoring process to achieve a fine-tuning effect. Hence, the sensors can be deployed in the field without any further calibrations as the light ambience changes over a long period of time. The flowchart in Figure 3 illustrates the motion detection algorithm.

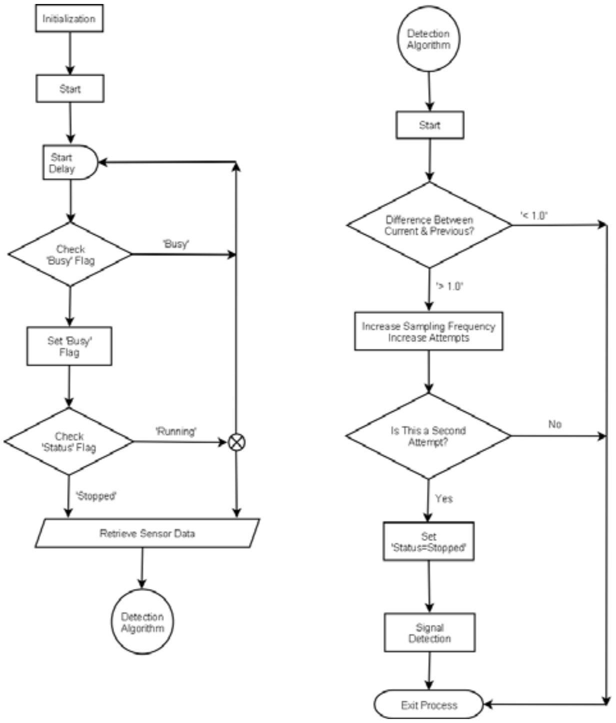


Figure 3: Motion detection algorithm flowchart.

Counter-Based (CB) Routing Protocol

After the intrusion detection has taken place, the sensor which initiates the detection needs to convey the message to the surrounding sensors. The surrounding sensors need to propagate the message throughout the entire sensor network until the user is informed of the detection. Hence, an efficient routing protocol[2, 4, 7, 10, 11] needs to be devised.

Wireless sensor network applications can be classified into the following four types of data flow: event-driven, query-driven, continuous, and hybrid [8]. Our research project focuses on the event-driven application. One of the requirements of such an application is that, the event has to be reported to the data sink in a timely fashion, so that a corresponding action can be carried out. Hence, it is desirable for the data to travel via the shortest path to minimize the propagation delay. Furthermore, a reliable propagation path is essential to ensure a high Quality of Service (QoS) during the data delivery.

In order to minimize the power wastage, the counter-based (CB) routing protocol is utilized[6]. A pure flooding-based broadcasting protocol is modified by inserting a randomized delay before transmitting a data packet. Each sensor node retains a copy of the last routed packet. Figure 4 illustrates the CB routing protocol. Assuming that only node A and B are within communication range of the sender. The sender has to route a data packet via node A or node B in order to reach node C. Hence, by introducing a random delay before each transmission, nodes A and B will transmit the relayed data packet at a different time depending on which random timer expires first. Therefore, if node A transmits first and intercepts node B before transmission, this will prevent a redundant data transmission as node B will detect the data duplication and drops the packet silently thereby saving the transmission power and reducing network traffic.

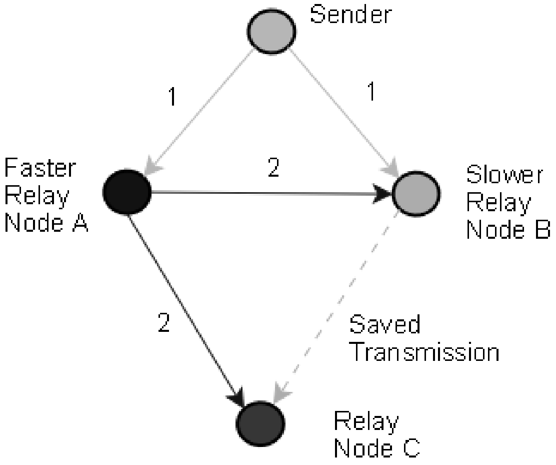


Figure 4: Routing the same packet via the same sensor node.

Automatic Repeat Request in Counter-Based Routing Protocol

The CB routing protocol is especially advantageous in a network deployment that has at least 3-connectivity. However, due to its broadcasting nature, one disadvantage is the lack of feedback from a receiving node to ensure QoS. Furthermore, in a practical scenario, the network deployment is often randomly deployed with a network model that closely approximates a Poisson distribution leading to deployments that are 2-connectivity or 1-connectivity. Due to the indeterministic nature of the sensors' location, a CB routing algorithm might result in a network partition despite that sensors are within the communication range. This problem is referred to as the virtual network partition (VNP) problem.

Figure 5 illustrates a randomly deployed sensor network that will create a potential VNP. Two clusters are realized based on the deployment. Thereafter, the left cluster can relay data packets to the data sink only via the critical node situated between the two clusters. As all the nodes in a cluster are within the communication range, the other two receiving sensors will receive the data packet from the transmitting sensor (i.e., source). Based on the CB routing protocol, both sensors will attempt to relay the same data packet at a different time. Assuming the sensor node connected to the critical node is the slower one. The faster node will intercept the slower node and cause it to drop the data packet even before the packet is relayed. Hence, the message will be lost even though the sensors are fully connected.

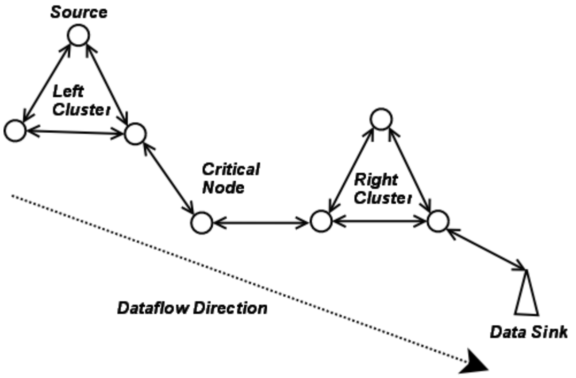


Figure 5: Virtual network partition.

In order to address the problem of message loss, an Automatic Repeat reQuest (ARQ) is integrated into the CB routing protocol. By virtue of a broadcasting based protocol, each receiving sensor node is responsible for broadcasting a relayed data packet after a random delay. Hence, if a pair of sender and receiver sensors is within the communication range, the sender should expect to receive the same packet from the receiver after a certain elapsed time. This elapsed time is equivalent to the summation of the maximum random delay, time taken to transmit the data packet, and the round-trip propagation delay. Hence, the sender will wait for a reply in the form of a transmitted data packet based on the period defined as *Total Delay*:

$$\begin{aligned}\text{Total Delay} = \text{Max Random Delay} &+ \\ \text{Data Transmission Time} &+ \\ \text{Round - trip Propagation Delay}&.\end{aligned}$$

The ARQ timer is reset when the sender receives any packet from its neighboring sensors. A sender will retransmit the previously routed data packet if it does not receive any data packets after an ARQ timeout period of total delay. In order to solve the virtual network partition problem, an extra data field, *override*, is also introduced into the sensor data payload. The override field instructs the receiver mote to allow the retransmitted data packet to be relayed even though it is a duplicated packet. However, the receiving node will reset the override field before the data packet is relayed. This will ensure that the data packet is only overridden for the first layer of sensor motes with respect to the node where it is first set. The flowchart in Figure 6 and the pseudo-code in Figure 7 illustrate the counter based routing protocol.

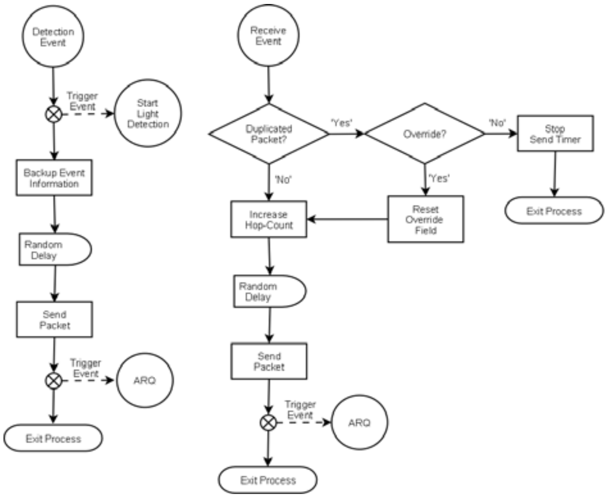


Figure 6: Routing protocol flowchart.

```
DataMsg* current_pkt;
int prev_hop_count;

Function Receive (DataMsg* data_pkt) {
  Resend_timer_stop();
  int hop_count = data_pkt->hop_count;

  if (data_pkt is duplicated) {
    if (hop_count => prev_hop_count) {
      if (override) {
        data_pkt->override = 0;
      }
      else {
         Send_timer_stop();
        return;
      }
    }
  }
  prev_hop_count = data_pkt->hop_count;
  data_pkt->hop_count++;
  current_pkt = data_pkt;
  Send_timer_fire_once(random_delay);
}

Function Send_timer_fired() {
  Transmit(current_pkt);
  Resend_timer_fire_once(round_trip_delay);
}

Function Resend_timer_fired() {
  current_pkt->override = 1;
  Transmit(current_pkt);
}
```

Figure 7: Routing protocol pseudo-code.

To the best of our knowledge, this work is the first effort to integrate the ARQ into a CB routing protocol to address the problem of the VNP. Furthermore, an event-based interface facilitates handshaking between the new routing protocol and the proposed lightweight motion detection algorithm is defined to realize the intrusion detection application.

Hardware Implementation

The implementation of the mobile tracking in a wireless sensor network consists of the motion detection system, the routing protocol, the path planning algorithm, and the system integration. The entire system requires collaborations between sensor hardware, robot hardware, and system software for both wireless sensors and the mobile robot.

Experiment Hardware

The sensors used in the project are MICA2 sensors from the Crossbow Technology[1], shown in Figure 8. On the *MPR400CB*, the processor onboard is the *Atmel ATmega128L* micro-controller. It has 128KBytes of instruction *EEPROM* and another 4KBytes of data *EEPROM* for data storage. The communication module onboard is the *Chipcon CC1000* radio. It also has a 51 pin I/O connector to interface to the *MTS400CA* sensor board shown in Figure 9. The hardware available for use on the sensor board are the humidity and temperature sensor, barometric pressure and temperature sensor, light sensor, and 2-axis accelerometer[1]. This project makes use of the light sensor to implement the motion detection algorithm.

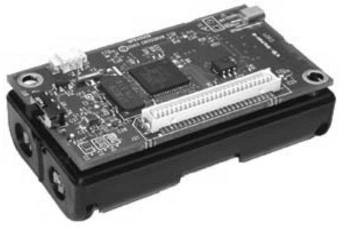


Figure 8: MICA2 MPR400CB sensor mote [1].

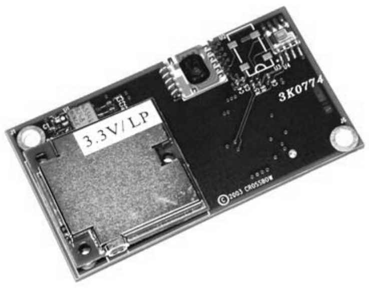


Figure 9: MTS400CA sensor board [1].

The implementation of the motion detection algorithm and the counter-based routing protocol are built over the TinyOS operating system. Experiments are carried out to analyze the efficiency of the algorithms with an objective to project an accurate simulation that is close to the actual mobile tracking problem. The system architecture of the experiment is shown in Figure 10.

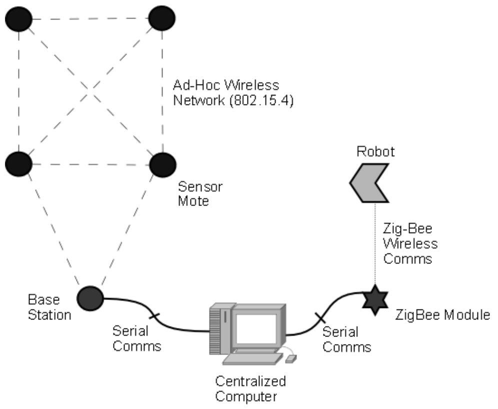


Figure 10: System architecture.

A simple test-bed is constructed to simulate a sensor field while a robot is utilized to simulate an intruder in the sensor field. Figure 11 shows the “iRobot® Create” robot as a moving robot in the sensor field. The robot provides an electronic and software interface called the iRobot® Create Open Interface for controlling Create’s behavior and reading its sensors[5].

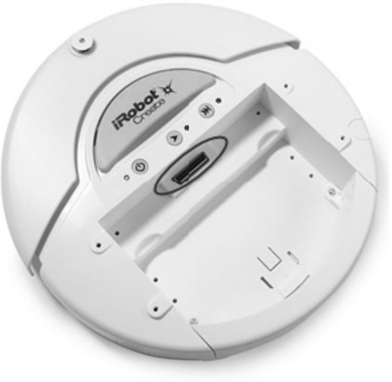


Figure 11: “iRobot® Create” robot.

The oval circle locates the 25-pin Cargo Bay Connector in which the external controller can interface with. In this experiment, a ZigBee module is interfaced to this connector, so that the centralized computer is able to control the robot wirelessly. The main components on the interface circuit are listed and the connections on the breadboard are illustrated in Figure 12.

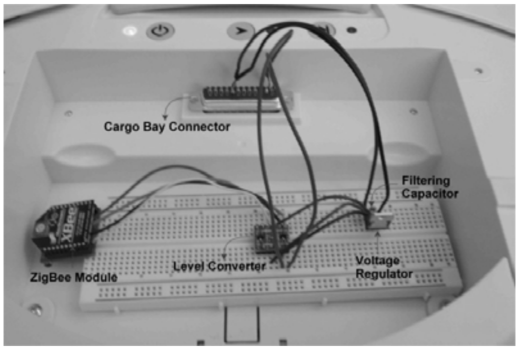


Fig. 12: Breadboard circuit mounted
on iRobot® Create.

Figure 13 illustrates the wireless sensors that are setup beneath the test-bed. Each sensor is assigned with a unique node ID with node ID 1 dedicated to the base station. Due to the deployment strategy of the sensors, the test-bed is divided into four cells which are identified by their corresponding wireless sensor's node ID numbers found beneath it. Figure 14 illustrates the four portions on the test-bed with an iRobot® Create robot deployed over it.

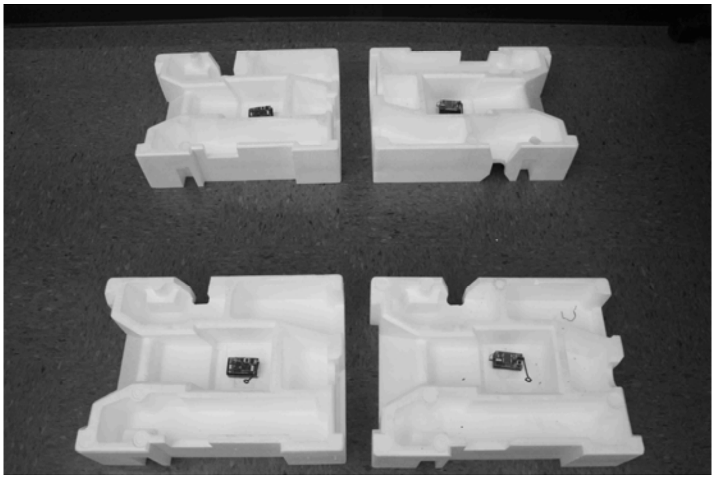


Figure 13: Experiment test-bed sensors.

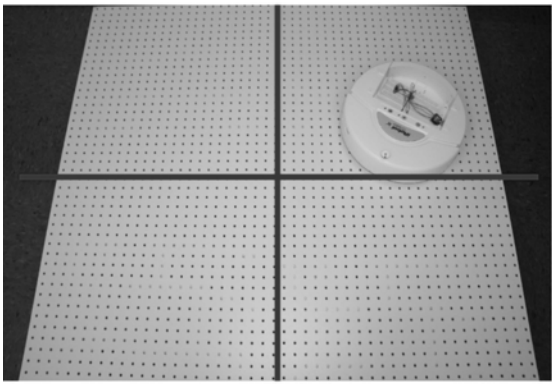
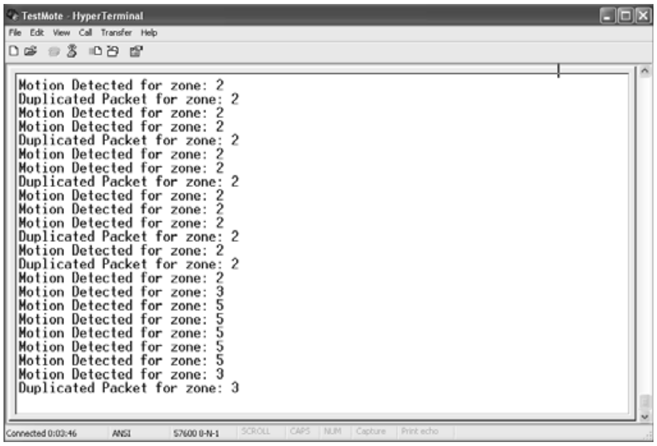


Figure 14: Experiment test-bed.

Experiment Observations

The light that passes through the test-bed surface holes illuminates the individual cells. Experiments are conducted under the normal room condition in which the light ambience within each cell is approximately 20 counts. As the robot travels across the sensor motes, the light ambience will drop gradually until a minimum value of approximately 3 counts. When the sensors report a positive detection based on the difference in the light ambience, the horizontal distance from the position of the sensor to the point where the first drop of light ambience can be detected denotes the sensing range of a sensor. The nearest sensor node ID numbers to the moving robot are shown on the user's computer shown in Figure 15. In all, the wireless sensor network is able to report the location of the robot promptly and accurately. The results justify the advantage of employing wireless sensor network in the mobile tracking problem.



```
Motion Detected for zone: 2
Duplicated Packet for zone: 2
Motion Detected for zone: 2
Motion Detected for zone: 2
Duplicated Packet for zone: 2
Motion Detected for zone: 2
Motion Detected for zone: 2
Duplicated Packet for zone: 2
Motion Detected for zone: 2
Motion Detected for zone: 2
Motion Detected for zone: 2
Duplicated Packet for zone: 2
Motion Detected for zone: 2
Duplicated Packet for zone: 2
Motion Detected for zone: 2
Motion Detected for zone: 3
Motion Detected for zone: 5
Motion Detected for zone: 5
Motion Detected for zone: 5
Motion Detected for zone: 5
Motion Detected for zone: 5
Motion Detected for zone: 3
Duplicated Packet for zone: 3
```

Figure 15: The nearest sensor node IDs to the intruder robot.

Conclusion

This paper studies the mobile robot tracking problem, where users can discover the moving robot's location with the help of a wireless sensor network. Based on the available MICA2 sensor mote hardware, an energy and time efficient motion detection algorithm is designed and a new counter-based routing protocol is proposed by making use of the automatic repeat request in solving the problem of the virtual network partition. This provides a lightweight solution while preserving the advantage of being an easily implementable protocol. Both the intrusion detection algorithm and the routing protocol are implemented in a prototypical wireless sensor network consisting of five MICA2 sensor motes.

In summary, the mobile tracking project can serve as an additional course material, an undergraduate senior design project, and a master thesis topic at Southern Illinois University Edwardsville. More future research in this area can be done based on our preliminary results, for instance, extending the motion detection and the routing protocol to multiple robots tracking.

References

1. Crossbow, "MTS/MDA Sensor Board Users Manual," http://www.xbow.com.

2. T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor Systems*, pp. 171-180, 2003.

3. O. Dousse, C. Tavoularis, and P. Thiran, "Delay of intrusion detection in wireless sensor networks," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pp. 155-165, 2006.

4. A. El-Hoiydi and J.-D. Decotignie., "Wisemac: An ultra low power mac protocol for multi-hop wireless sensor networks," *Algorithmic Aspects of Wireless Sensor Networks*, vol. 3121, pp. 18-31, 2004.

5. iRobot® Create from iRobot Corporation. http://www.iRobot.com.

6. S. Ni, Y. Tseng, Y. Chen and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 151-162, 1999.

7. J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 95-107, 2004.

8. S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, “A taxonomy of wireless micro-sensor network models,” *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 6, no. 2, pp. 28-36, 2002.

9. Y. Wang, X. Wang, B. Xie, D. Wang, and D.P. Agrawal, “Intrusion detection in homogeneous and heterogeneous wireless sensor networks,” in *IEEE Transactions on Mobile Computing*, vol. 7, no. 6, pp. 698-711, 2008.

10. W.B. Heinselman, A.P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” in *IEEE Transactions On Wireless Communications*, vol. 1, No. 4, pp. 660-670, 2002.

11. W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” in *Proceedings of the IEEE Infocom*, pp. 1567-1576, 2002.

Biographical Information

Yoon Kah Leow received the B.Eng degree in Electrical and Electronic Engineering from Nanyang Technological University, Singapore, in 2004, and the M.S. degree in Electrical Engineering from Southern Illinois University Edwardsville, Illinois, in 2009. He is currently pursuing his Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Arizona, Tucson. His research interests include Wireless ad hoc and sensor network, reconfigurable computing systems, and high performance distributed computing.

Ying Shang received the B.S. in Control Engineering from Shandong University, Jinan, China in 1998, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Notre Dame, in 2003 and 2006, respectively. She is currently an assistant professor in the Department of Electrical and Computer Engineering at Southern Illinois University Edwardsville. Her research interests include discrete event and hybrid systems, sensor communication networks, and power systems.