Cooperative Path-finding of Multi-robots with Wireless Multihop Communications

Sujin Park, Jin Hong Jung, Seong-Lyun Kim
Radio Resource Management & Optimization Laboratory
School of Electrical and Electronic Engineering
Yonsei University
Seodaemun-Gu, Seoul 120-749, Korea
{sjpark, jung_jinhong, slkim }@ramo.yonsei.ac.kr

***Abstract*—Path-finding by multiple robots has been studied by many researchers. In particular, when the robot is able to communicate with other colleague robots, the path-finding will be solved collaboratively among the robots. Main motivation of this paper is to apply the wireless multihop communication to the collaborative path-finding problem. For the purpose, we propose an algorithm called CPA (Cooperative Path-finding Algorithm), and report its performance by real implementation on Zigbee-based micro-robots and also by MATLAB simulations. There has been controversy over the wireless multihop communications on using it in typical wireless networks (e.g., cellular, WLAN) not only for technical but also for non-technical reasons. However, our initial study in this paper highlights a new potential of the multihop communication, in inter-robot information exchange.**

***Index Terms*—Path-finding, multi-robots, wireless multihop communications**

I. INTRODUCTION

Wireless communication has expanded to wireless *multihop* networks, which include ad hoc radio networks, sensor networks, wireless mesh networks and mobile multihop relay systems [1]. With multihop capability, wireless communication can be combined with cooperative communications [2] and network coding [3], which have attracted even more researchers to this field. In many wireless multihop networks, the merits of capacity enhancement [4] as well as coverage [1] exceed the delay caused by multihop relay. Still, however, there are unresolved issues that may not be necessarily technical; one question regards the real motivation of a relay node that allows packet relay for the other transmitting nodes by consuming its own energy. There is also a security issue in multihop communications, in the sense that one's own data transmission is received by someone else in close proximity.

On the other hand, in the networked robotics area, researchers try to realize group behaviors found in small insects or animals, in order to control and coordinate a team of robots. The multiple robots communicate with each other, sharing the same mission, naturally through wireless communications.

Main purpose of this paper is to investigate how effectively the wireless multihop communication can be used for multi-robot information exchange. For the purpose, we examine the gain of *multihopping* in a group of networked robots. Thus, our viewpoint is different from those in [1] and [4], in which the *spectral efficiency* (bits/hertz) is of primary concern. To the best of our knowledge, the potential of wireless multihop communication has not been seriously considered in the networked robotics area. At the same time, in the wireless communication field, practical applicability of multihop communication to the networked robots has not been considered either.

With this background in our mind, as initial efforts to understand the issue, we focus on the following problem: Given a random *maze* composed of walls and paths, how can a group of robots escape to the exit in the maze as quickly as possible? In the robotics area, this problem is defined as *path planning problem* [5]. The authors in [5]-[10] have suggested various algorithms on the problem but the main goal of those papers is in the efficiency of the algorithm, i.e., how fast a given algorithm can solve the problem. In this paper, we focus on using wireless multihop communication for the path planning problem. Ultimately, we seek for a simple path-finding algorithm that is more effective than any other sophisticated algorithms, when combined with the multihop communication.

Throughout the paper, each robot, equipped with sensors that detect walls and paths, can save the learned map into its memory and move toward a desired direction. Besides, each robot exchanges the map information with the others, through wireless multihop communication. Importantly, what is different from the conventional wireless multihop communications is that a relay robot (node) does not *simply* forward the map information to the next robot in the multihop routing sequence. Instead, the relay robot adds its own map information to what it has received from the others, and forwards the compound information to the next. This speeds up the process of sharing the map information among all the robots in the maze, which in turn increases the overall performance of the robots. We measure the performance of the robots by the *time duration*, during which all the robots escape from the maze.

Also, we assume a single-frequency network, in which all the robots share a single radio channel. For the multihop communication, we use the so called *random basketball routing* (BR) that was recently suggested in [11] and [12], as a multihop routing algorithm optimized for the node mobility.

Our cooperative path-finding by multiple robots is verified by two ways: real implementation and simulations. We imple-

This work was supported by Center for Advanced RFID/USN Technologies at Yonsei University, KOREA (IITA-2008-C1090-0801-0038).

WIOPT 2008, 1st – 3rd Apr 2008, Berlin, Germany.
ISBN 978-963-9799-18-9
DOI 10.4108/ICST.WIOPT2008.3229

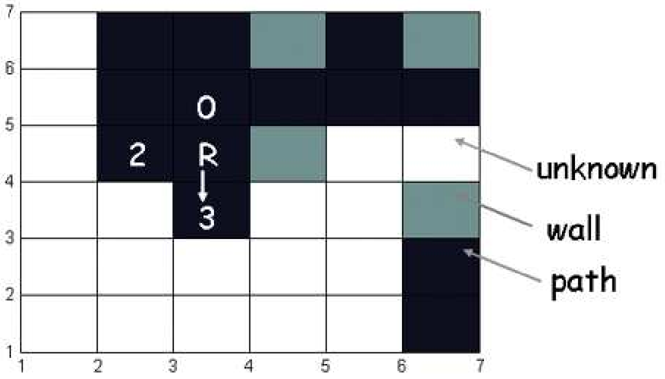

Fig. 1. Mode 1: Mobile movement with no exit information

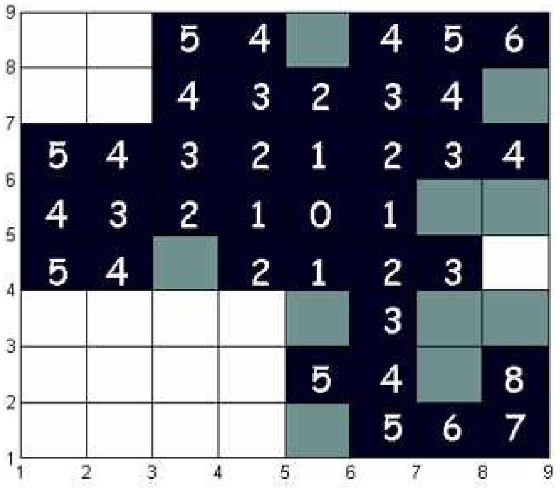

Fig. 2. Mode 2: Mobile movement with exit information

mented our cooperative path-finding on micro-robots, in which BR is embedded to the Zigbee (IEEE 802.15.4) physical layer. We also simulated a large number of robots using MATLAB, to see the gain of multihopping in solving the path planning problem. Results from both implementation and simulations are encouraging in that we can see significant gain from the multihop communications.

The rest of this paper is organized as follows: In Section 2, we propose our cooperative path-finding algorithm (CPA). In Section 3, we describe how we implement CPA on Zigbee based micro-robots, and introduce some illustrative examples from the implementation. Our simulation results are contained in Section 4. Finally, Section 5 concludes the paper with remarks on ongoing research.

II. COOPERATIVE PATH-FINDING ALGORITHM (CPA)

A. *Two Modes of Robot Movement*

The maze we use in this paper is as shown in Figure 6. A mobile robot in a maze can sense/detect a wall or a path, which is visualized in a 6×6 cell maze of Figure 1. In Figure 1, each cell denotes either a wall or a path, of which the color specifies the status. The black-colored cells are those explored by robots and known as paths. On the other hand, those grey-colored cells are detected as walls by robots. Unexplored cells are colored white. Each robot maintains such a map while it moves. A robot can move to four directions: front, back, left and right. A robot can share its map information with the others via wireless multihop communications. Throughout the paper, we assume that each robot recognizes its initial location, so that the map information constructed gradually by a robot is easy to be combined with the maps forwarded by the other robots. That is, we assume robots have their own *absolute* grid (axis) information when they start exploration.

In the multi-robot case, it is possible for two robots to collide in the maze. However, this is not considered in our simulation, so that multiple robots can coexist in a same location. This is to avoid the complexity of our simulation but in real implementation, each robot regards the others as moving obstacles.

In Figure 1, the number in each black cell denotes the number of unexplored white cells around the cell. For example, the cell with the number 3 has three white cells out of the four directions. In the figure, the current location of the robot is the "R" cell, and the robot moves to the black cell (direction), of which the number is highest. The number in each black cell is being continuously updated as the robot moves and gets map information from the other robots. Also, if there is a tie-break among the cells, a robot moves to the black cell that has the smallest number of visits by the same robot. This means that each robot counts/memorizes how many times it has visited each black cell.

When the exit of a given maze is found/detected, the robot updates the location of the exit on its map, and tries to forward the map to the others. A robot, once it has the exact location of the exit on its map, moves forward to the direction to the exit, using the so called *contour line methods*, which is illustrated in Figure 2. In the figure, the number on each black cell denotes the distance (in terms of cells) to the exit, and robot moves to the direction that decreases the distance. This means that there are two modes in the robot movement: A robot, without knowledge of the location of the exit, moves according to Figure 1 (Mode 1). On the other hand, once it knows the exit, robot movement will follow the contour line method of Figure 2 (Mode 2).

B. *Random Basketball Multihop Routing*

The multihop communication protocols designed for mobile robots (nodes) should be not only light in respect of overhead, but also self-configurable according to the robot movement. One direction for meeting the above requirements is a per-hop-based (i.e., hop-by-hop) multihop routing, where each node (robot) forwards its packet to its neighboring nodes under *favorable* positions. The term, "favorable" can be defined differently, depending on which objective is set for the network performance (e.g., energy, delay, jitter and throughput, etc.). This kind of multihop communication is known as the *opportunistic routing* (see [13] and literature therein). In this paper, we use the random basketball multihop routing (BR) [11], [12], for inter-robot multihop communication, which is also an opportunistic routing. In this section, for explanation purpose, we borrowed some text and a figure from the paper [11].

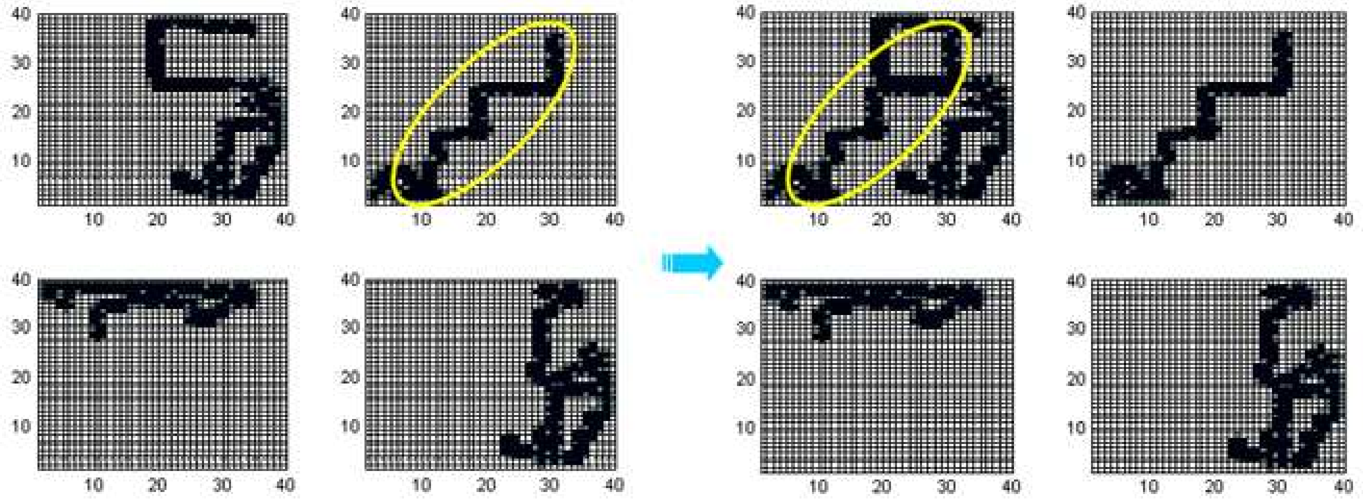

Fig. 3. Sharing the map via random basketball routing among four robots in our simulation. The robot in the northeast transmits its map to the one in the northwest.

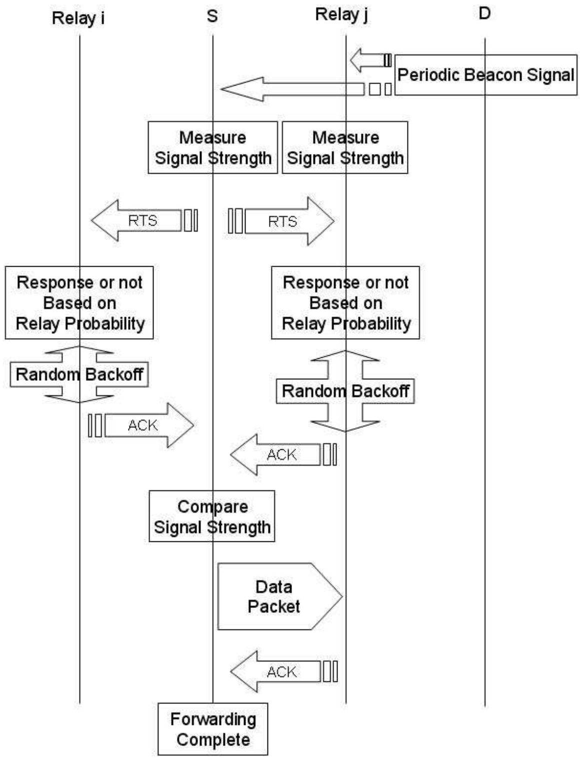

Fig. 4. Signaling for choosing a relay

BR has a key parameter p, which is called the *relay (receiving) probability*. For a given time slot, a source node (robot) having data (map information), is either in transmission mode with probability $1-p$ or in receiving mode with probability p. When a source node is in transmission mode, it sends its data either to a relay node (robot) or directly to the destination node (robot) if the destination is within one-hop communication range. When a source node is in receiving mode, the node listens to the other transmitting nodes as a candidate relay node.

Figure 4 shows the signaling process of BR for a source node in transmission mode to choose a relay. Destination node D periodically broadcasts a beacon signal so that the strength of the signal is measured and stored by each receiving node. When node S wants to transmit some packets in transmission mode, it sends out Request-to-Send (RTS) to its neighboring candidate relay nodes (determined by the relay probability p) within the radio range. Each neighboring candidate relay node, after receiving RTS, waits short random time slots to avoid collision, and then sends out acknowledgement (ACK). In the ACK packet header, there should be at least two information: one is the measured signal strength of the periodic beacon signal from the destination and the other is the ID of the candidate relay node (i.e., the owner of ACK). After receiving ACK packets from neighboring candidate relay nodes, node S compares them and chooses a relay node (to pass data packets) that reports the strongest received beacon signal from D.

Each robot (S), once it updates its map, determines destination robot (D) to send the map. In this paper, we assume that each robot *randomly* chooses its destination robot out of its one-hop communication range. The relay node (robot) on the multihop routing sequence, will add its own map information to what it has received, and forward the new map to the next robot. This procedure will be repeated until the map reaches the destination robot. Note that this map delivery is executed simultaneously and in parallel among the robots. Figure 3 illustrates how four robots located remotely, share the map information, by using BR.

III. IMPLEMENTATION OF COOPERATIVE PATH-FINDING ALGORITHM

We have implemented our cooperative path-finding algorithm (CPA) on micro-robots (Figure 5) with Zigbee-based communication functionality, which conforms to the IEEE 802.15.4 physical layer specifications.[1] In addition, each robot is embedded with six infrared-sensors, sound sensors, light

---

[1] All hard- and softwares in our experiments are based on Bioloid robots manufactured by Robotis Inc., Korea (http://www.robotis.com)

Parameter	Value
Processor clock	16 Mhz
Internal_communication_speed	57,600 bps
Maximum_velocity	161 mm/sec
Avg_beacon_interval	960 ms
Avg_rts_interval	1280 ms
Response_wait_time	480 ms
Map_transmission_time	350 ms
Ack_wait_time	450 ms
Transmission range	10 M
Radio frequency	2.4 Ghz
Maximum data rate (Zigbee)	250 Kbps

TABLE I
EXPERIMENT PARAMETERS

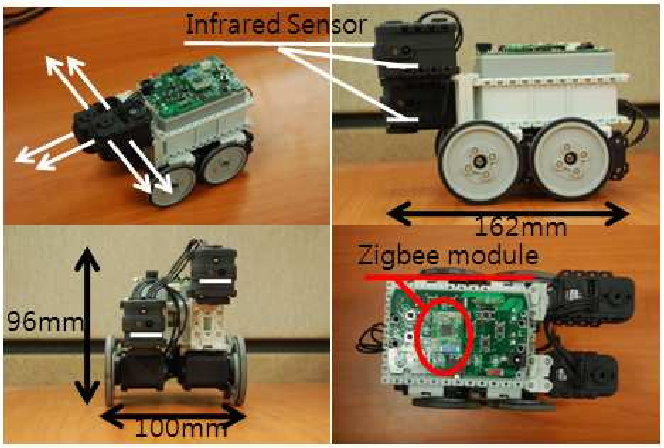

Fig. 5. Micro-robot with embedded with infrared-sensors and Zigbee

sensors, buzzers, four-step-motors and a microprocessor, ATmega128 in its size of $162 \times 100 \times 96\ mm^3$. By using infrared-sensors, robots can detect and explore the random environment and other sensors are turned off in our experiment. Infrared-sensors are set up to detect three directions from a given robot (front, right and left).

Key experimental parameters are specified in Table I. The internal communication speed of the robot is $57,600$ bps. A robot can move up to 161 mm per second. Beacons and RTS are periodically broadcast by destination robots and source robots, of which intervals are specified in the table. BR waits 480 ms for responses after broadcasting RTS. Also, it waits another 450 ms for ACK after map transmission. The entire map information is composed of 33 packets, which takes 350 ms for transmission to the other robot.

To examine the performance of CPA, we compare the algorithm with a counter part, where each robot does not communicate and moves forward according to the two modes described in Section 2-A. Our evaluation is done by two robots in a random maze of 6×11 cells, as shown in Figures 6 and 7.

Figure 6 shows how the robots move according to CPA and Figure 7 contains the traces of the robots. In both figures, (a) shows the initial position of a leading robot that moves to the front cell. The second robot appears in (b), while the leading robot explores the right side of the first T junction. The second robot follows the leading robot. In (c), the second robot arrives at the same T junction which is already explored by the leading robot. And the leading robot is now exploring the second T junction. In (d), the second robot is able to skip the right side of the T junction due to the map information transmitted by the leading robot with BR. Meanwhile, the leading robot escapes from the dead-end of the second T junction. In (e), the second robot takes a different path. It moves to unvisited cells by the leading robot. At last, both robots completely explore the environment effectively and escape from the maze. The leading robot takes 38 sec and the second robot, 30 sec.[2] The average time for escaping

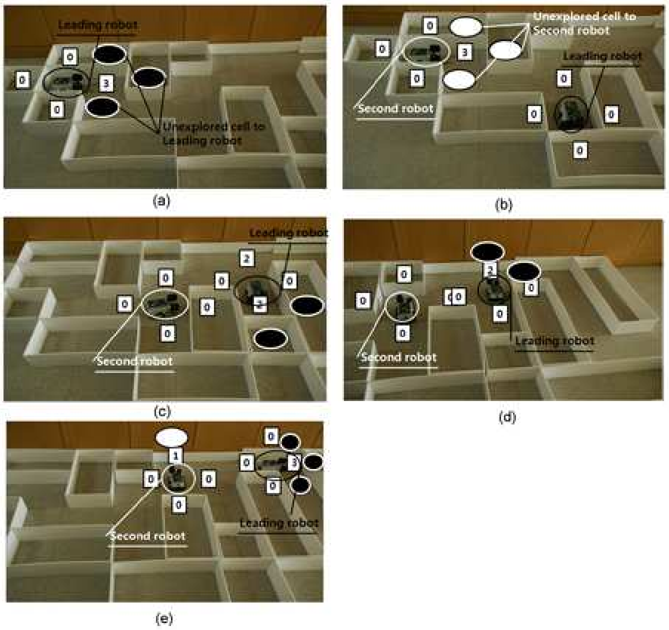

Fig. 6. Cooperative path-finding by two robots

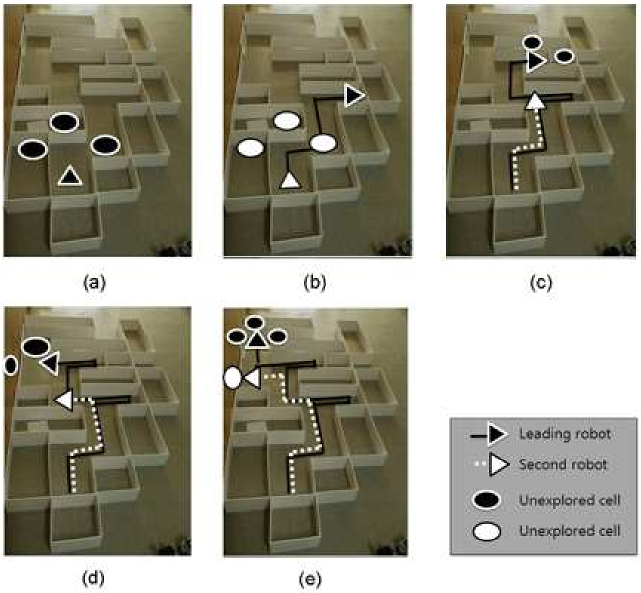

Fig. 7. Traces of two robots

---

[2] Moving pictures of the experiments can be downloaded from http://ramo.yonsei.ac.kr/cooperation.avi

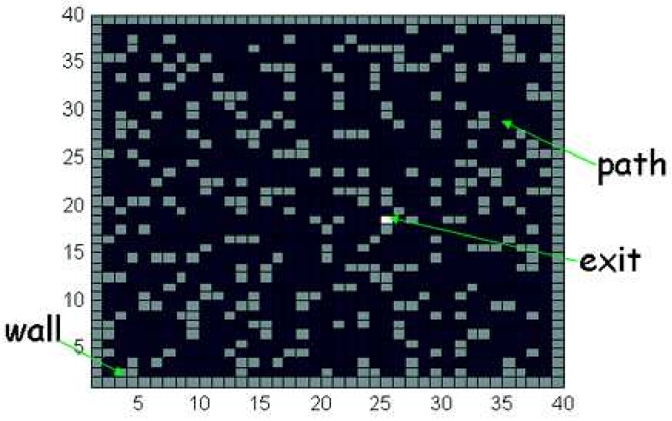

Fig. 8. 40 × 40 cell maze for our simulation

by a robot is 34 sec in this example. On the other hand, if inter-robot communication might be turned off, then it would take 38 sec on average for one robot to escape from the maze. This explains the gain of inter-robot communication. To further investigate the benefit of the multihop communication, we simulate the robots on a large sized maze, which is contained in the next section. This experiment demonstrated that exploring unknown environment with multi robots can be done effectively with our approach. Thus this experiment also imply efficient cooperative exploration can occur when all robots share the same map without complex exploring algorithm.

IV. SIMULATION RESULTS

In our simulation, we use a 40×40 cell maze as illustrated in Figure 8, where we have simulated 500 independent initial positions of robots to collect the results. During a time slot each robot performs sensing, transmitting and moving. The map information of a robot can be transmitted up to two hops in a slot. The relay probability p is affected by some network factors such as node density. We fixed p as 0.72 following the results in [12].

Figure 9 shows the average escaped time as a function of the number of robots, where we choose the communication range of a robot among 0, 8 and 16 cells. Though the radio channel fluctuates as the robots move, we assume that all robots have the fixed communication range as above, to simplify our simulation. Since our purpose is to examine the impact of wireless multihop communication on the path-finding, this simplification is acceptable. In the figure, the zero communication range stands for the case that the robot does not communicate with each other. In this case, since we assume that multiple robots may coexist in the same cell in our simulations, the average escaped time of a robot is independent of the number of the robots in the maze. However, we see that the escaped time decreases rapidly as the number of robots increases, when the robots communicate with the others. This can be interpreted as the gain of the cooperative path-finding by communication between robots.

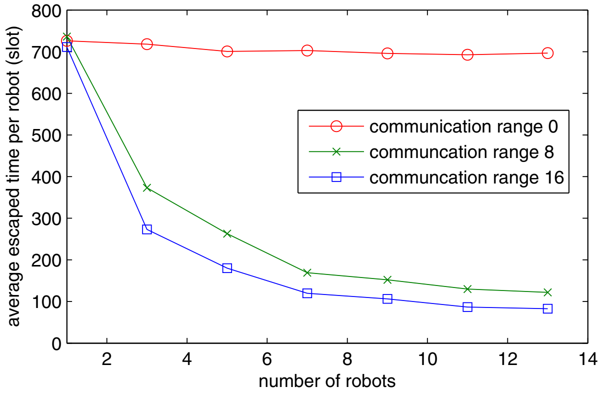

Fig. 9. Average escaped time of a robot as a function of the number of robots in the maze.

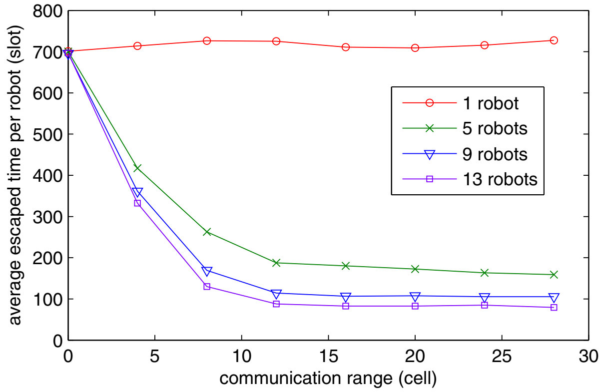

Fig. 10. Average escaped time of a robot as a function of the communication range (in cells) of a robot.

Next, we check how the communication range affects the performance of the path-finding. Figure 10 shows the average escaped time as a function of the communication range of a robot. Note that the curves start at the almost same point in the zero communication range case, as discussed in Figure 9. However, they decrease rapidly as the communication range increases. The increase of the communication range has both pros and cons. When it increases the mobile robots can make single-hop communication with the other robots located remotely. Because the map information of remotely located robots are uncorrelated, the increased communication range will have positive effects on the performance of the path-finding. On the other hand, as the communication range increases, inter-robot communication will more likely be done by single-hop transmission. As mentioned in Introduction, the gain of multihopping comes from merging of multiple maps from different robots in the multihop routing sequence. So when communication is done solely by single-hop transmission, such multihopping gain may not be expected. In Figure 10, we see that the curve rapidly decreases until the communication range reaches up to 12. This is because the positive side of communication-range increase, exceeds its the negative side. However, beyond the level of 12, the average escaped time is decreasing rather slowly. This is due to the fact that single-hop communication is rather widespread and the network has little chance to get benefits of multihopping.

In Figure 10, the speed of the decrease is boosted by the number of the robots. However, when the communication range becomes larger than 12, there are only subtle differences. So it is better not to expand the communication range beyond a certain level. Instead, increasing the number of robots will bring more effective results.

V. CONCLUDING REMARKS

In this paper, we proposed a cooperative path-finding algorithm for a collaborative robot system. By measuring how fast one robot can escape from a random maze, we illustrated the effects of wireless multihop communications on the cooperation of multiple robots. The amount of map information shared by all the robots will be exponentially increasing as time goes with the multihop communications.

On the other hand, there might be some errors in delivering map information. In this case, the so called *bad news* will be quickly spread out to the entire group of robots, which would affect the performance of the robots in a destructive way. Throughout the paper, we assume that there is no communication error but this should be considered for more realistic simulations. The simulation results are also affected by the assumption that more than one robots can coexist in the same location at a time. If we do not accept this assumption, the increased number of robots would have both positive- and negative impact: increasing the escape time (due to collision) and decreasing it (due to cooperation). Thus net effect of the number of robots needs careful simulations. These are the points that our current research is heading towards.

REFERENCES

[1] IEEE 802.16j/D1-2008, "Air interface for fixed and mobile broadband wireless access systems - multihop relay specification," August, 2007.

[2] J. N. Laneman, D. N. C. Tse and G. W. Wornell, "Cooperative diversity in wireless networks: efficiency protocols and outage behavior," *IEEE Trans. Info. Thoery*, Vol. 50 (12), pp. 3062-3080, 2004.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Trans. Info. Theory*, Vol. 46 (4), pp. 1204-1216, 2000.

[4] S. Toumpis and A. J. Goldsmith, "Capacity regions for wireless ad hoc networks," *IEEE Trans. Wireless Communications*, Vol. 2 (4), pp. 736-748, 2003.

[5] A. Stentz, "Optimal and efficient path planning for unknown and dynamic enviroments," *Technical report*, CMU-RI-TR-93-20, The Robotics Institute, Carnegie Mellon University, PA, USA, 1993.

[6] H. Noborio, K. Fujimura and Y. Horiuchi, "A comparative study of sensor-based path-planning algorithms in an unknown maze," Proc. *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 909-916, 2000.

[7] B. H. Kazerouni, M. B. Moradi and P. H. Kazerouni, "Variable priorities in maze-solving algoithms for robot's movement," Proc. *IEEE Int. Conf. Industrtial Informatics*, pp. 181-186, 2003.

[8] V. S. Gordon and Z. Matley, "Evolving sparse direction maps for maze pathfinding," Proc. *Congress on Evolutionary Computation*, pp. 835-838, 2004.

[9] L. Podsedkowski, J. Nowakowski, M. Idzikowski and I. Vizvary, "A new solution method for path planning in partially known or unkonwn environment for nonholonomic mobile robots," *Elsevier Robotics and Autonomous Systems*, Vol. 34, pp. 145-152, 2001.

[10] M. Szymanski, T. Breitling, J. Seyfried and H. Wörn, "Distributed shortest-path finding by a mirco-robot swarm," *Springer Ant Colony Optimization and Swarm Intelligence*, LNCS 4150, pp. 404-411, 2006.

[11] D. M. Kim, Y. J. Hwang, S.-L. Kim, "Random basketball routing for ZigBee based sensor networks," Proc. 4th *IEEE VTS Asia Pacific Wireless Communcations Symposium*, Hsinchu, Tiwan, 2007; extended version submitted to *Elsevier Computer Communications*, 2008.

[12] Y. J. Hwang and S.-L. Kim, "The capacity of random wireless networks," to appear in *IEEE Trans. Wireless Communications*, 2008 (http://ramo.yonsei.ac.kr/br.pdf).

[13] R. C. Shah, S. Wietholter, A. Wolisz and J. M. Rabaey, "When does opportunistic routing make sense?," Proc. *IEEE Int. Conf. Pervasive Computing and Communications Workshops*, 2005.