Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

An Analysis of Random Peer-to-Peer Communication for System-Level Coordination in Decentralized Multiple-Robot Systems

Chris A. C. Parker
Department of
Computing Science
University of Alberta
Edmonton, Alberta, Canada
Email: parkerca@ieee.org

Hong Zhang
Department of
Computing Science
University of Alberta
Edmonton, Alberta, Canada
Email: zhang@cs.ualberta.ca

***Abstract*— Inter-robot communication is essential if general purpose intelligent decentralized multiple-robot systems are to become a reality. Traditionally, explicit communication amongst the robots of a MRS has been broadcasted or eliminated altogether. The limits of these two approaches prevent the widespread deployment of useful decentralized MRS. In this paper, we present and analyze, both analytically and empirically, a communication protocol that we call random peer-to-peer or RP2P. Using RP2P, a decentralized MRS can share state in logarithmic time with respect to system population. Additionally, the load placed upon the individual robots by RP2P is independent of system population size. Potential applications of random peer-to-peer communication also are provided, including state sharing, decision-making and task allocation.**

I. Introduction

Inter-robot communication is essential in a multiple-robot system (MRS) if it is to exhibit truly cooperative behaviour beyond simple reactive coordination [1]. There are three basic organizational structures for MRS: centralized, hierarchical and decentralized. In both centralized and hierarchical systems, the control is centralized in a single robot. Since control flows in one direction in these systems (from the top down), system-level coordination is relatively simple to visualize. However, these classes of MRS suffer from communication bottlenecks and both are vulnerable to single-point failures. If the central control robot becomes overloaded or damaged, the performance of the entire system will be degraded.

Decentralized systems are quite different in this regard, as none of the robots in a decentralized MRS have any authority over their teammates. As a result, these systems are relatively unaffected by the failure of individual robots. Instead of group behaviour being dictated from the top down, it emerges from the bottom up via the robots' peer-to-peer interactions. However, because there is no central controller in a decentralized MRS, the problem of system-level coordination lacks intuitive solutions.

Early decentralized MRS, such as the puck-sorting robots of Deneubourg *et al.* did not communicate directly with each other [2]. Instead, system-level coordination was achieved by carefully tuning the robots' reactions to each other and their environment so that a useful global behaviour would emerge out of their interactions. This approach to control is known as stigmergy [3]. Several works have demonstrated the power and simplicity of stigmergic control including the box-pushing work of Kube and Zhang [4] and the stick-pulling robots of Ijspeert *et al.* [5], but they also have demonstrated its primary caveat. As Holland and Melhuish point out, "[T]he success of [stigmergy] is crucially dependent on real-world physics [3]." System-level coordination via stigmergy does not allow for general purpose decentralized MRS that are not tied to specific environments or tasks. System-level coordination is important as we do not want to have to interact with each robot in a system in order to direct it. Rather, we would prefer to be able to direct a MRS as a single entity.

Without a centralized controller to schedule events within a MRS, we must accept that the robots will interact with each other stochastically. Any communication scheme that the robots might employ must at least tolerate this randomness and ideally should take advantage of it. The focus of this paper is an analysis of what can be accomplished by the random peer-to-peer exchange of messages between the members of a decentralized MRS and how we can expect such random communication to load the individual members of a system. In the next section, we describe the random peer-to-peer communication protocol as well as state the assumptions that will guide our analysis. Then, in Section II, we carry out a theoretical analysis of the protocol. In Section III, we present an experimental verification of our theory based on a set of computer simulations. We close this paper with some conclusions and lay out our future work.

A. Random Peer-to-Peer Communication

Random peer-to-peer (RP2P) message passing is a very simple concept that, as we will demonstrate, allows for surprisingly efficient system-level coordination in a MRS. Because it is based upon random peer-to-peer interactions, it is perfectly suited to decentralized MRS. Individual robots simply wait for a communication channel to become available and then send a message to a randomly chosen teammate. In a very short period of time, information can be propagated across an entire MRS and will become part of a system's common knowledge. Two very simple assumptions of our robots are made in our analysis of this communication protocol, both of which are easily satisfied by current wireless communication technology. First, each robot must be uniquely identifiable by a communication port so that the messages may be addressed. Second, we assume that all messages are transmitted directly from a source robot to a destination robot in a single hop. Many MRS never spread out more than a few tens of meters; when one considers that typical 802.11 radios can communicate over distances of 100 meters, the assumption of single-hop communications is entirely realistic.

RP2P communication in a decentralized MRS is similar to the gossip algorithms proposed for wireless sensor networks by Boyd *et al.* [6]. However, unlike the sensor networks, we do not propose to use RP2P to compute numerical results directly. Robots in a decentralized MRS choose their own actions based on their knowledge of the state of the environment. RP2P, by allowing the robots in a decentralized MRS to share state more easily, will allow the robots to choose their actions based on a common set of beliefs.

B. Why not use Broadcast Communication?

In this paper, we analyze a proposed communication protocol to allow system-level coordination of a MRS. An obvious question might be why we simply do not advocate the use of broadcast communication. Via broadcast communication, a single robot can address all of the robots in its system simultaneously.

In order for broadcast communications to be viable, a very low noise channel must be present, since individual recipients cannot request the sender to resend portions of the transmission. Thus broadcast communication is error prone, range-limited and, at best, limited to very simple messages. RP2P, being peer-to-peer, allows any number of error-correction methods to be employed.

Broadcast communication is inherently limited by the power of the transmitter and therefore cannot be implemented on systems bigger than the transmitter's footprint. Although we confine our analysis in this paper to single-hop communications, there is nothing preventing RP2P from being extended to multi-hop communications. Therefore, RP2P can be scaled to geographically huge systems.

Finally, RP2P communication is *faster* than broadcast communication. Consider state-sharing. In this paper, we show that a system employing RP2P communication can share state in logarithmic time. A broadcast system can learn the state of *one* of its members in constant time, but in order to truly share state, every robot must take its turn broadcasting over the channel. Ignoring the range and error issues outlined above, it would still take a period of time linear with respect to system population for a MRS to share state via broadcast communication.

II. Theoretical Analysis

In this section, we present a theoretical analysis of RP2P communication. We will focus on two primary questions. First, how long will it take for a system to share state? Second, how many messages should we expect the individual robots to receive over a given period of time under the worst-case conditions? We assume that each robot semi-regularly transmits a message to a randomly chosen teammate and that over some time interval, we can expect each robot to transmit a message. We call this time interval a "*communication round*." It is important to note that the robots need not send their messages simultaneously; despite being undesirable, such synchronization is unrealistic in a decentralized MRS. The actual length of a communication round would depend in practice on the bandwidth of the communication channel available to a system and the number of robots sharing it.

A. Time Required to Share State in a Decentralized MRS

In order to share state, the individual robots' knowledge of their environment must be made common knowledge across their MRS. Our question can be stated as follows: Given an $n-$robot system in which a single robot initially has information "x" and in every round of communication, every robot that knows x randomly chooses another robot in the system and communicates x to it, how many communication rounds will it take until all n robots know x? Because RP2P communication is parallel, the time required for one robot to share its knowledge is the same as the time required for all robots to share their knowledge and thus for the system to share state.

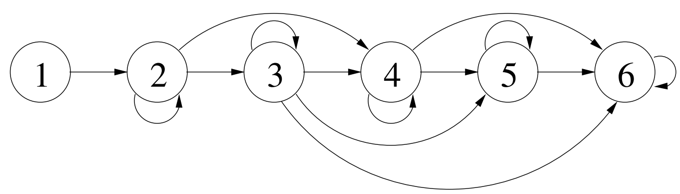

Fig. 1. This state diagram represents a 6-robot system. Each state corresponds to the number of robots in the system that know some piece of information x. Every communication round, each robot that knows x tells a randomly chosen teammate about x. Eventually, every robot will know x.

We model our system as a Markov chain, a $6-$robot example of which is shown in Figure 1, wherein the state corresponds to the number of robots that know x. The initial state of the system is state 1. A system's state can at most double (if every single robot that knows x contacts a different robot that does not know x) and at least stay the same (if every robot that knows x contacts a robot that already knows x) after each communication round.

To compute the expected time to propagate the knowledge of x across a MRS, we compute the state-transition probabilities of our Markov chain system representation and then compute the expected number of state transitions required to reach the system's absorbing state. First, we calculate the probability that $m\epsilon[0, i]$ of the i robots that know x will contact some of the $n-i$ robots that do not know x. This probability follows the binomial distribution and is given by Equation 1.

$$P_{\text{contact}}(i,m,n) = \binom{i}{m}\left(\frac{n-i}{n-1}\right)^{m}\left(\frac{i-1}{n-1}\right)^{i-m} \quad (1)$$

Equation 1 tells us how many of the knowledgeable robots we can expect to contact unknowledgeable robots, but not how many unique unknowledgeable robots will be contacted. To determine how many unique unknowledgeable robots will learn of x, we must ask ourselves: Assuming that j robots each randomly choose one of k robots and send a message to it, how many ways can the messages be addressed so that exactly $l\epsilon[1, \min(j,k)]$ robots will receive a message?

Let us assume that the function $f(l,k,j)$ provides this result. There are l^j ways in which j robots can contact up to l robots and $\binom{k}{l}$ ways in which the l robots that get contacted can be chosen. However, the l^j ways in which up to l robots can be contacted includes all of the permutations in which less than l unique robots are contacted[1]. We must subtract all of the permutations in which fewer than l unique robots receive a message. The number of permutations to be subtracted is provided by the function $f()$: $\sum_{a=1}^{l-1}\binom{l}{a}f(a,a,j)$. Finally, we can see that $f(1,k,j) = k$ by inspection[2], which provides and anchor condition for our recursive definition of $f(l,k,j)$, which shown in Equation 2.

$$f(l,k,j) = \begin{cases} l>1: & \binom{k}{l}\left[l^j - \sum_{a=1}^{l-1}\left(\binom{l}{a}f(a,a,j)\right)\right] \\ l=1: & k \end{cases} \quad (2)$$

In total, there are k^j ways in which j robots can contact robots from the pool of k robots, so the probability of exactly l robots receiving a message assuming that j robots each address a message to a robot randomly chosen from k robots = $\frac{f(l,k,j)}{k^j}$. Using this result and Equation 1, we can calculate the state transition probabilities of the Markov chain representation of our decentralized MRS by Equation 3. The characteristic form of these transition matrices can be seen in Figure 2.

$$P_{transition}(S_i \rightarrow S_{i+h}) = \sum_{q=h}^{i}\frac{P_{contact}(i,q,n)\cdot f(h,n-i,q)}{(n-i)^q} \quad (3)$$

Markov chains of the form illustrated in Figure 1 have exactly one absorbing state: the state in which all of the robots know x. The expected number of communication rounds required for a piece of information to propagate across a decentralized MRS using RP2P communication is the same as the expected number of state transitions required to reach the system's absorbing state from state 1. This can be calculated by first decomposing the transition matrix P as shown in Equation 4.

[1]For example, one of the l^j ways in which j robots can contact up to l robots includes the case where all j robots send a message to the same robot.

[2]There are $\binom{k}{1} = k$ ways in which one robot can be chosen from k robots, and only one way in which j robots can all contact the same robot.

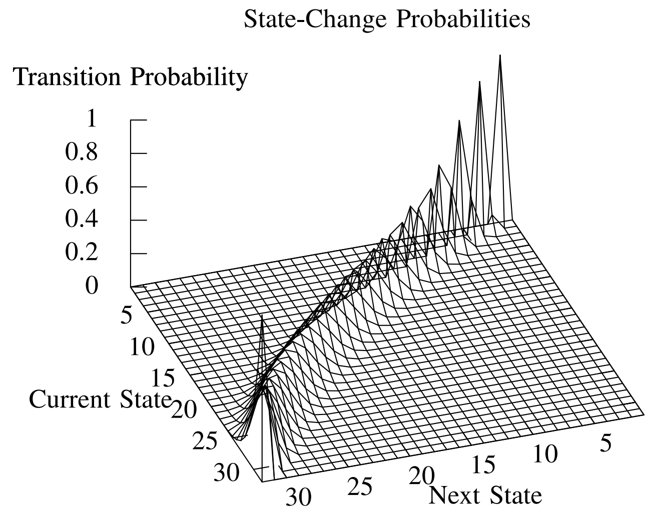

Fig. 2. This figure depicts the state transition matrix that describes how a piece of information x spreads from a single robot to the rest of its 32-robot system via RP2P communication. The elements of the matrix, p(current state, next state), are the probability of the number of robots that know x will increase from *current state* to *next state* in the next communication round. The matrix's narrow ridge of non-zero probability leads to small confidence intervals in the overall transition time - random robot behaviours lead to highly predictable system-level behaviour.

$$P = \begin{bmatrix} S & T \\ 0 & I \end{bmatrix} \quad (4)$$

The expected number of state transitions is equal to the sum of the elements in the first row of the matrix Q, which is computed from the decomposition of P as shown in Equation 5.

$$Q = (I-S)^{-1} \quad (5)$$

Figure 3 plots the expected number of communication rounds required to spread a piece of information across a MRS via RP2P versus the robotic population. Note that the trend is logarithmic: doubling a system's population increases the time required for the robots to share their knowledge by a constant amount (~ 1.7 communication rounds for every doubling of population). Furthermore, the absolute variance in the expected time converges to a constant for system populations above 16 robots.

B. Message Traffic in an n-Robot System

Based on the analysis of the previous section, RP2P communication allows robots in a decentralized MRS to pool their knowledge very quickly. We now turn our attention to the load that RP2P communication would place on the individual robots. If we do not consider how messages are generated, a communication protocol can load its participants via either transmission or reception. In RP2P, robots transmit

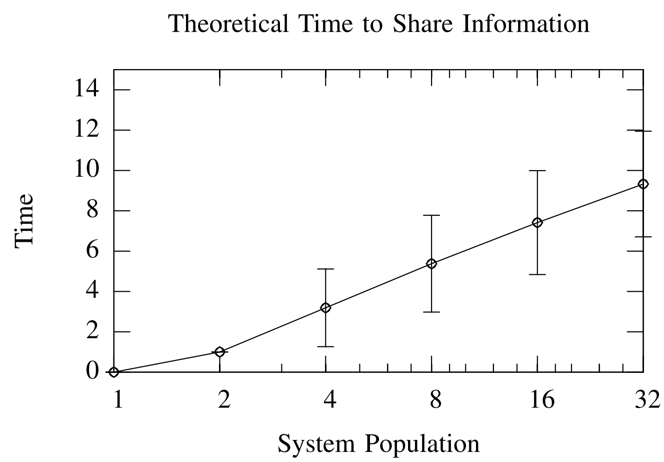

Fig. 3. Here we plot the time required for a decentralized MRS to share their knowledge with each other versus system population. Notice the logarithmic relationship and that the magnitudes of the 95% confidence bounds converge to a constant.

one message per communication round. Thus the transmission load per robot is fixed and well behaved with respect to system population. On the other hand, a robot conceivably could receive from zero to $n-1$ messages in a single communication round. For RP2P to be practical, we must calculate the probability of a robot receiving various numbers of messages per round and how this result will scale with system population. Our analysis begins with Equation 6, which describes the probability of a robot receiving i messages over a single communication round in an n-robot system.

$$P(\text{receive } i \text{ messages}) = \binom{n-1}{i}\left(\frac{1}{n-1}\right)^{i}\left(\frac{n-2}{n-1}\right)^{n-i-1} \tag{6}$$

Equation 6 can be expanded and then factored into a more useful form, given by Equation 7.

$$P(\text{receive k messages}) = \frac{1}{k!}\left(\frac{\prod_{i=3}^{k}(n-i)}{(n-1)^{(k-2)}}\right)\left(\frac{n-2}{n-1}\right)^{n-k} \tag{7}$$

Of the three terms of Equation 7, only two are significant. The middle term takes the form of Equation 8 which converges to unity as n increases, removing it from the equation.

$$\frac{n^{k-2} + a_{k-3}n^{k-3} + a_{k-4}n^{k-4} + \ldots + a_1 n + a_0}{n^{k-2} + b_{k-3}n^{k-3} + b_{k-4}n^{k-4} + \ldots + b_1 n + b_0} \tag{8}$$

This leaves the first term - a scaling constant - and the third term. Upon closer inspection, we can see that the third term is a variation on the limit definition of the root of the natural logarithm, e, given by Equation 9.

$$\lim_{n\to\infty}\left(\frac{n}{n-1}\right)^{n} = e \tag{9}$$

Thus we can approximate Equation 6 with Equation 10. We can see that this equation is valid by inspection, as the infinite sum of the reciprocals of the factorials $= e$, and multiplying this by $\frac{1}{e}$ yields unity as the sum of the probabilities of receiving 0 to n messages per communication round.

$$\lim_{n\to\infty} P(\text{receive i messages}) = \frac{1}{i!e} \tag{10}$$

Figure 4 plots Equation 6 for the $i = [0 : 3]$ against system population for up to $n = 32$ robots. The graph clearly illustrates the rapid convergence of the probabilities to their predicted limits. Given the rate at which $\frac{1}{i!e}$ decreases, we should not expect individual robots to receive large numbers of messages regardless of their system's population. Thus we conclude that the load placed on individual robots by RP2P communication is minimal and independent of the robotic population of a decentralized MRS.

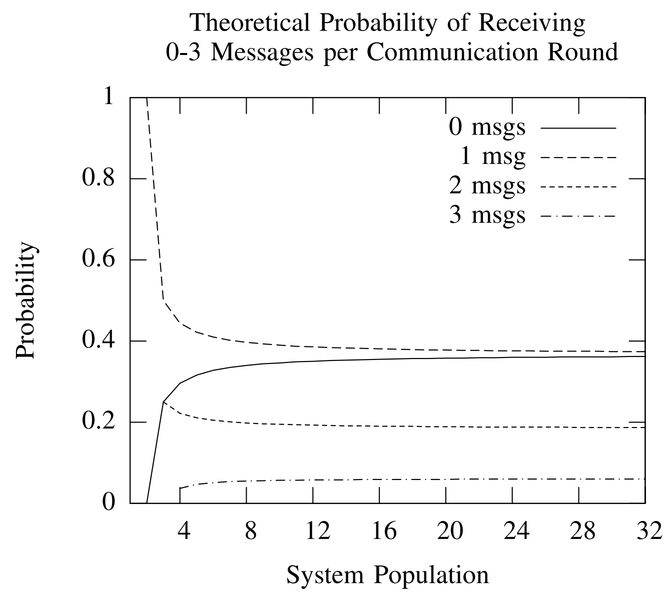

Fig. 4. This graph plots the probability of a robot receiving 0, 1, 2 or 3 messages in a single communication round versus system population. The probability of receiving i messages quickly converges to its theoretical limit of $\frac{1}{i!e}$, which is independent of system population.

III. Experimental Verification

Having conducted a theoretical analysis of the behaviour of the RP2P communication protocol, we now present the results of simulations that were carried out in order to validate our theory. We present our experimental verification in the same order as the theory was developed in Section II.

A. Time Required to Disperse Information

In order to measure the actual time required for a piece of information to spread to every robot in a decentralized MRS using RP2P communication, numerical simulations of n robots were carried out in which initially only one robot knew some piece of information. In every round of communication, each robot that knew the piece of information randomly chose a teammate and sent it a message containing the information. $10,000$ trials were conducted for each system population. The results of these simulations are plotted in Figure 5.

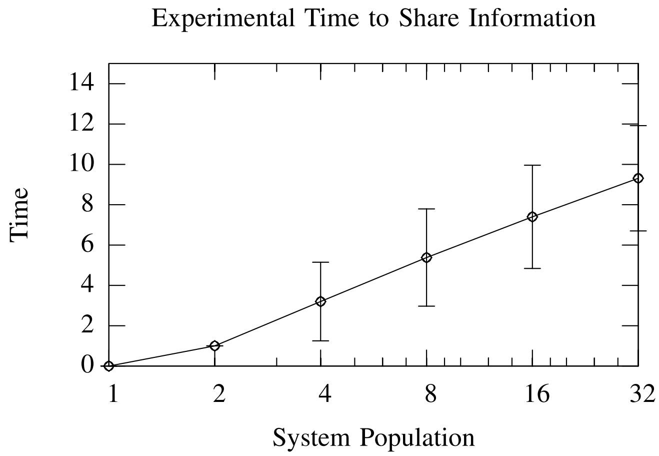

Fig. 5. This figure plots the actual time required for a robot to distribute its knowledge to the rest of its system as measured from a simulation of robots exchanging messages with each other. Our measured data is almost identical to the theoretical predictions presented in Figure 3.

This figure and its theoretical counterpart, Figure 3, are in complete agreement. In Figure 6, we plot the magnitudes of the 95% confidence intervals of the experimentally determined information distribution times against system population. Their convergence to a constant ($\sim$ 2.6 communication rounds) clearly is evident.

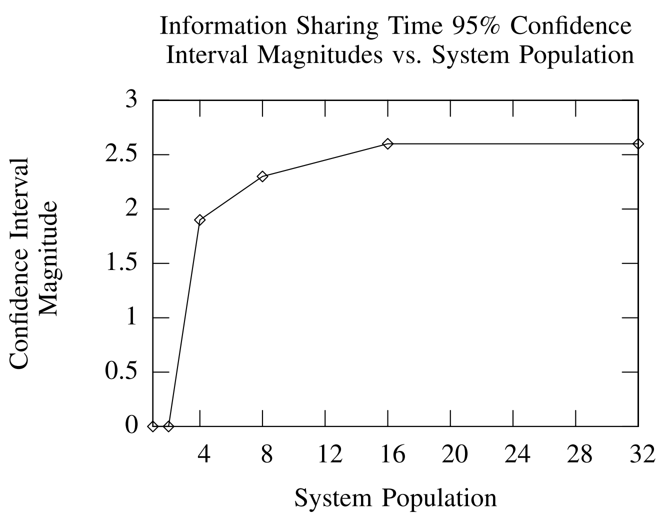

Fig. 6. In this graph, we plot the magnitudes of the 95% confidence intervals of the data from Figure 5 for systems up to 32 robots. Despite the increase in system population, the absolute confidence in the absorption time remains constant for MRS with populations greater than $n = 16$ robots.

B. Message Traffic in an $n-$Robot System

In order to measure the communication load on the individual robots in terms of the number of messages that each should expect to receive per communication round, n simulated robots randomly passed messages amongst themselves. From their communication logs, we computed the probabilities of receiving 0, 1, 2 or 3 messages per round. This data is plotted against system population in Figure 7.

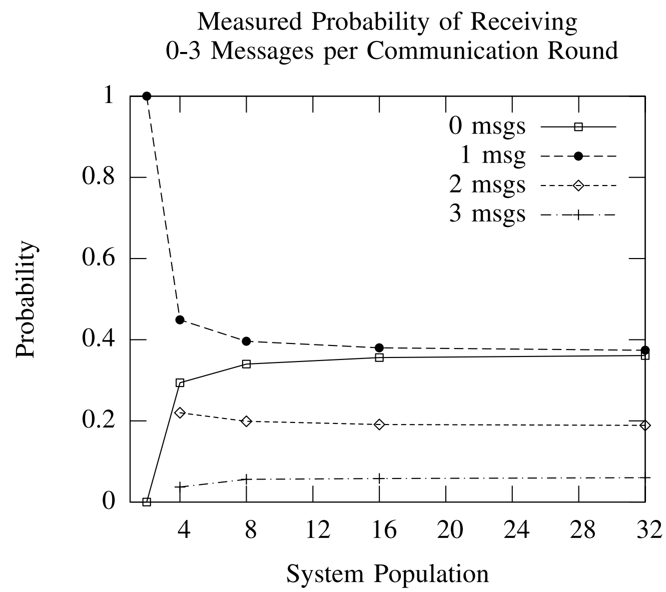

Fig. 7. Here we plot the probability of a robot receiving 0 to 3 messages per communication round as measured from our simulations of MRS. These measurements are nearly identical to our analytical predictions presented in Figure 4

Again, our experimental results (Figure 7) agree with our theoretical predictions (Figure 4).

IV. Conclusions and Future Work

In this paper, we have presented and analyzed, both analytically and empirically, a communication protocol for decentralized MRS that we have called random peer-to-peer, or RP2P. Through our analysis, we have demonstrated that RP2P represents a powerful coordination mechanism despite the protocol's inherent simplicity. Via RP2P, we have shown that information can be shared amongst the robots of a decentralized MRS in logarithmic time. Additionally, the communication load placed on the individual robots is independent of the number of robots that compose a system. Thus RP2P will scale well to large systems.

Many papers have advocated the use of decentralized MRS due to their apparent robustness, yet most decentralized MRS to date have been single-purpose systems tuned to specific environments due to the difficulties encountered in deploying explicit inter-robot communication in them. Our work has shown that explicit communication can be used effectively in decentralized MRS and we have provided a protocol for its implementation.

Our next tasks are threefold. We currently are developing physical MRS that will use RP2P to coordinate the individual behaviours of the member robots. In particular, we will employ RP2P to enable state sharing and emergency notification. Second, we will combine the results of this paper along with those of our earlier decision-making work [7] to solve the collective relocation task in a physical environment. Finally, in this work, we have concentrated on single-hop communications. In order to make RP2P practical for a larger variety of MRS, we intend to investigate the use of multi-hop message routing to deliver the peer-to-peer messages. Multi-hop communications will increase the communication load on the individual robots, as they will have to accommodate the messages intended not only for themselves, but also those that they would be expected to relay on to their teammates, but it will decrease the load on the communication channel, as robots could reduce the range of their transmissions. How much these factors will increase the load on the individual robots and affect the overall performance of our communication scheme needs to be investigated further.

REFERENCES

[1] G. Dudek, M. Jenkin, and E. Milios, *Robot Teams*. A K Peters, Ltd., 2002, ch. A Taxonomy of Multiple Robot Systems, pp. 3–22.

[2] J. L. Deneubourg, S. Goss, N. R. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, "The dynamics of collective sorting: Robot-like ants and ant-like robots," in *Proceedings of First International Conference on Simulation of Adaptive Behavior*, 1990, pp. 356–365.

[3] O. Holland and C. Melhuish, "Stigmergy, self-organisation, and sorting in collective robotics," *Journal of Adaptive Behaviour*, vol. 5, no. 2, pp. 173–202, 1999.

[4] C. R. Kube and H. Zhang, "The use of perceptual cues in multi-robot box pushing," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, 1996, pp. 2085–2090.

[5] A. J. Ijspeert, A. Martinoli, and A. Billard, "Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment," *Autonomous Robots*, vol. 11, no. 2, pp. 149–171, 2001.

[6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *Proceedings of IEEE Infocom 2005*, vol. 3, March 2005, pp. 1653–1664.

[7] C. A. C. Parker and H. Zhang, "Biologically inspired decision making for collective robotic systems," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, September 2004, pp. 375–380.