Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Efficient Unicast Messaging for Mobile Robots

Saumitra M. Das, Y. Charlie Hu, C.S. George Lee, and Yung-Hsiang Lu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-2035
{smdas, ychu, csglee, yunglu}@purdue.edu

Abstract— **Mobile multi-robot teams are useful in many critical applications such as search and rescue. Explicit communication among robots in such mobile multi-robot teams is useful for the coordination of such teams as well as exchanging data. Since many applications for mobile robots involve scenarios in which communication infrastructure may be damaged or unavailable, mobile robot teams frequently need to communicate with each other by using ad hoc networking. In such scenarios, energy efficient routing protocols to deliver messages among robots are a key requirement.**

In this paper, we propose and evaluate two routing protocols tailored for use in ad hoc networks formed by mobile multi-robot teams: Mobile Robot Distance Vector (MRDV) and Mobile Robot Source Routing (MRSR). Both protocols exploit the unique mobility characteristics of mobile robot networks to perform efficient routing. Our simulation study show that both MRDV and MRSR incur lower overhead while operating in mobile robot networks when compared to traditional mobile ad hoc network routing protocols such as DSR and AODV.

I. INTRODUCTION

Communication between mobile robots is useful and even critical in many applications. In applications such as search and rescue, communication infrastructure may be damaged or not present requiring the mobile robots to form an ad hoc network using each other as forwarding nodes to enable communication. Message routing among the mobile robots or from the robots to a human operator thus requires routing protocols that can operate without central control and handle dynamic topology changes due to the mobility of the mobile robots. Messaging requirements for mobile robot applications can be categorized into two major primitives: (1) Unicast messages sent from a mobile robot to another could be required by many applications for sending data and images, asking for assistance, etc., and (2) Multicast messages sent from a human operator or one mobile robot to a group of receivers which are typically used for co-ordination and control.

In this paper, we focus on the unicast primitive. Unicast messaging is useful for coordination-oriented communication. Coordination-oriented communication is defined to be the transmission related to the coordination and control of robot teams and has been extensively studied in the literature. In particular, the use of communication for robot control was studied in [9]. It has been found that communication provides certain performance improvement for particular tasks [1].

Although traditional unicast routing protocols developed for the Internet cannot be used in mobile robot applications due to the presence of dynamic topology; many unicast routing protocols proposed for mobile ad hoc networks (MANETs) can be readily applied in such scenarios. However, protocols proposed for MANETs do not take into account two unique characteristics of mobile robot applications: (1) mobile robots have significantly planned and predictable movement patterns unlike human movement which is used as a mobility model in MANET protocol studies. For example, mobile robots typically may know the velocity and direction in which they are traveling. (2) mobile robots frequently pause at various points and often the duration of the pause can be estimated or is known beforehand. For example, if a robot pauses to measure some environmental parameter, the amount of time taken for the task can typically be estimated or known in advance.

In this paper, we exploit the mobility characteristics of mobile robot applications to improve message routing for unicast facilities. We propose, design and evaluate routing protocols for efficient performance in mobile robot applications. Specifically, we propose two routing protocols, Mobile Robot Source Routing (MRSR) and Mobile Robot Distance Vector (MRDV), for mobile robot networks based on the DSR [6] and AODV [8] protocols developed for mobile ad hoc networks.

II. PROBLEM FORMULATION

We focus on an application scenario where many robots are used to form a mobile robot network. Each robot has a simple sensory ability and limited computational power. This makes it practical to build a large number of such robots. The communication among the robots is based on wireless ad hoc communication (IEEE 802.11) in which robots forward each other's packets when the source and destination nodes of a packet are not within direct reach. This reflects our mobile robot testbed in which mobile robots are equipped with laptops and PDAs and use Orinoco Wavelan wireless cards to communicate with each other.

Mobile robot networks have the following unique mobility characteristics different from traditional MANETs. Consider Figure 1. On the left side, a pausing robot is

---

[1] This work was supported in part by the National Science Foundation under Grant IIS-0329061.

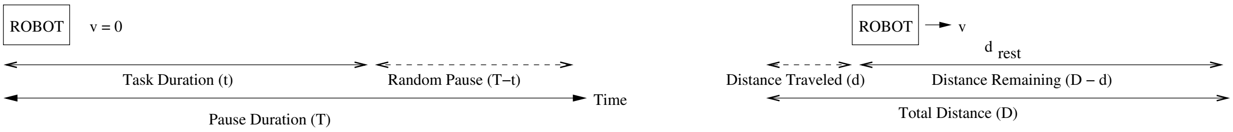


Fig. 1. Mobile robot model and assumptions.

depicted. A robot could be paused because it is awaiting further instructions from a human operator or is busy doing a task like detecting a gas. The pause duration (T) is thus composed of a time t to complete the current task followed by a random period of time $T - t$ after which the robot moves again. Typically, a small set of tasks can be performed by a robot and most such tasks take a predetermined amount of time. We thus assume a set of k tasks that the robot can undertake each with a different task duration. On the right side, a moving robot is depicted. In this case, most commands to a robot specify an amount of distance D to travel in a particular direction. Thus, even without GPS or localization support a robot can at the very least measure with some error the distance d already traveled and thus the distance $D - d$ which we denote the distance-to-rest (d_{rest}). Thus we assume that the robot can estimate its d_{rest} when mobile, and the time remaining to complete a task t. In this paper, we do not try to characterize the random pause duration after task completion. Simple heuristics like the exponentially weighted moving average (EWMA) of past random pause times can potentially be used for further improving the performance of routing protocols. We also assume that the robot has some means of calculating its current velocity. This is a valid assumption and has already been implemented in our mobile robot testbed. Additionally, the velocity may in many cases be determined by an energy efficient path planning algorithm and thus known to the control software.

III. PROTOCOL DESIGN

A. *Mobile Robot Source Routing (MRSR)*

MRSR is based on Dynamic Source Routing (DSR) [6], a well known multi-hop routing protocol for MANETs which is based on the concept of source routing in contrast to hop-by-hop routing. MRSR incorporates three mechanisms, route discovery, route construction, and route maintenance.

1) Route Discovery: Route discovery is the process by which a source robot discovers a route to a destination robot for which it does not already have a route in its cache. The process broadcasts a ROUTE REQUEST packet that is flooded across the network in a controlled manner. In addition to the address of the original initiator of the request and the target of the request, each ROUTE REQUEST packet contains a route record, which records the sequence of hops taken by the ROUTE REQUEST packet as it propagates through the network. ROUTE REQUEST packets use sequence numbers to prevent duplication.

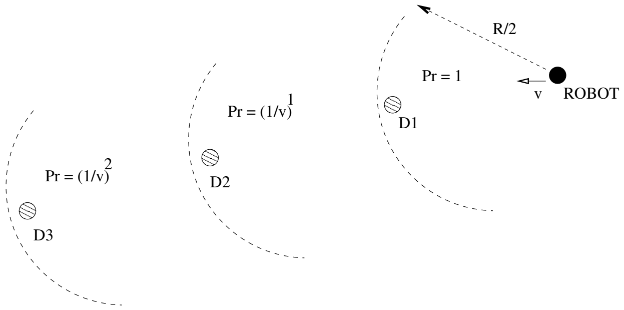


Fig. 2. Probabilistic stationarity based route discovery.

Robots probabilistically rebroadcast ROUTE REQUEST packets based on a probability value p_r. This mechanism is used to reduce routing overhead and energy consumption by limited the propagation of ROUTE REQUEST packets. In addition, this mechanism ensures that highly mobile robots become part of a route less frequently so as to increase the route lifetime. MRSR exploits the mobility knowledge present in the robots to calculate p_r. Once the ROUTE REQUEST is received by the robot, it calculates p_r and rebroadcasts the ROUTE REQUEST with a probability of p_r. If the robot is currently stationary, p_r is set to 1. Otherwise p_r is calculated using a combination of the current velocity v and the distance weighting factor γ of the robot. Since we assume that the robot knows how much distance it has to travel and the distance it has already traveled, it can calculate its distance to rest d_{rest} by subtracting these quantities. Given a wireless transmission range of R, the distance weighting factor is given by:

$$\gamma = \frac{d_{rest}}{R/2}$$

Finally, p_r is defined to be inversely proportional to the current velocity of the robot. Thus robots that are highly mobile are likely to rebroadcast a ROUTE REQUEST packet rarely. In addition, γ is used to exponentially decay the value of p_r. This is done because a higher value of γ indicates that the robot is more likely to move out of range. Thus p_r is finally calculated as:

$$p_r = (\frac{1}{v})^{\gamma}$$

Thus, MRSR exploits the fact that robots know how much they are traveling beforehand in many applications by using the distance weighting factor. If γ is 0, the robot is going to stop such that the route will remain valid, and p_r is correctly calculated as 1.0. Thus, even if the robot is traveling at a high velocity, γ takes into account wireless range and route validity to adjust the rebroadcast probability p_r. Figure 2 shows how p_r is calculated. D1, D2 and D3 denote different possible values of d_{rest} and the corresponding values of p_r calculated by the protocol are shown next to them.

Once the ROUTE REQUEST is received, it is answered by a ROUTE REPLY packet either from the destination node or an intermediate node that has a cached route to the destination. The ROUTE REPLY contains the reversed route record (source route) used by the ROUTE REQUEST which is used to reach back to the source. MRSR requires each robot encountered along the ROUTE REPLY path to encode its mobility information into the ROUTE REPLY packet. This information consists of an estimated timeout value which indicates when the robot will move out of the range of the route being set up and a 1-bit flag to indicate if the robot is moving or paused (M/P). If the robot is paused, it estimates its task completion time and inserts that into the ROUTE REPLY as the timeout. If the robot is moving, it encodes the timeout as the time taken to move out of radio range based on the estimate of its current speed. This timeout is calculated assuming that the robot is on average $\frac{R}{2}$ away from the previous node on the route. If the robot estimates it will stop before moving out of radio range, the timeout value is set to a high value (e.g. 25 seconds), since this means that route will be maintained by this robot for a longer period of time. This mobility information (i.e. estimated timeouts and flag assigned by each robot) is learned by each node that forwards the ROUTE REPLY back. This timeout value associated with a node is used to efficiently manage and utilize routes before they break as explained in the next section.

2) Route Construction: Similar to DSR, MRSR also uses aggressive caching to reduce the frequency and propagation of route discoveries. MRSR uses a graph cache similar in structure to that proposed in [5]. However, the algorithm used to manage the graph cache is modified for use in mobile robot networks. MRSR stores individual links of routes to build a topological graph of the network. Specifically, each mobile robot in the network maintains a graph cache, i.e., a topological view of the network based on discovered and forwarded link information. A single-source shortest path algorithm like Dijkstra can then be used to compute source routes to other mobile robots to whom paths exist in the graph.

MRSR expires links based on timeouts learned from ROUTE REPLY packets. The timeout of a link $l_{i,j}$ in the graph is calculated as the minimum of the timeout values learned for the endpoint nodes i and j of that link. If the timeout of a link $l_{i,j}$ is found to be 0, it is assumed to be broken. When a link $l_{i,j}$ breaks as indicated by the reception of a ROUTE ERROR packet, the link is immediately removed. Since there could be another link $l_{i,k}$ which may still be valid although $l_{i,j}$ is broken, instead of removing all links containing node i, MRSR marks only the link $l_{i,j}$ as broken and multiplicatively reduces the timeout associated with endpoints of the broken link, i.e., nodes i and j. This achieves the desired effect: the link in error is removed from the graph, and all other links containing either endpoint of the link in error have reduced timeouts reflecting the recent mobility of those endpoints.

When a route needs to be found to a destination node, MRSR first tries to use Dijkstra's algorithm to construct a route using only paused robot nodes in the graph. If such a route cannot be constructed, it tries to find a path with the longest lifetime based on the mobility information provided by the other robots. Specifically, the route that has the highest minimum timeout among all of its links is chosen. If there are multiple such routes, the shortest one is chosen.

3) Route Maintenance: Route maintenance is similar to DSR and consists of monitoring the operation of the route and informing the sender of any routing errors. If a route breaks due to a link failure, the detecting host sends a ROUTE ERROR packet to the source which upon receiving it, removes all routes in the host's cache that use the hop in error.

B. MRDV

MRDV is based on the well known AODV [8] routing protocol. MRDV shares on-demand behavior with MRSR/DSR and the use of hop-by-hop routing and destination based sequence numbers with another ad hoc routing protocol DSDV [7].

1) Route Discovery: Route discovery is performed via a process similar to MRSR. However, MRDV stores routing information as one entry per destination in contrast to MRSR which caches multiple entries per destination. Like MRSR, a node satisfies the ROUTE REQUEST by sending a ROUTE REPLY back to the source or by increasing the hop count and re-broadcasting to its neighbors. As the ROUTE REQUEST propagates from the source to various nodes, a reverse path is set up from these nodes back to the source. When the ROUTE REPLY travels backward to the source using the set up reverse path, a forward path is also set up towards the destination.

MRDV also probabilistically rebroadcasts ROUTE REQUEST packets. Each node along the ROUTE REQUEST sets a timeout value to indicate how long it is likely to remain as part of the route (i.e. before breaking the route). An estimated timeout with respect to the route being established is calculated similarly as in MRSR, and the timeout is inserted into the ROUTE REQUEST packet. Note that unlike MRSR, no source routing is done so only one timeout value can be inserted into the packet. However, since the weakest link in the route (i.e. the robot with the lowest timeout) decides the time of link breakage, one worst case value is sufficient to accurately characterize the lifetime of the route. Each subsequent node on forwarding the ROUTE REPLY checks the timeout set by the previous node. If the forwarding node's timeout is lower (it is more mobile than all preceding nodes on the route), it modifies the timeout encoded in the packet to its own timeout value and forwards the packet. If its timeout value is higher, the forwarding node just rebroadcasts the packet. When the ROUTE REQUEST finally reaches the destination, the final timeout value is encoded in the resulting ROUTE REPLY back to the source.

2) Route Maintenance: Similar to AODV, MRDV uses timers to expire routes that have not been used recently. However, in MRDV, these timeouts are set using the encoded value in the ROUTE REPLY unlike the statically assigned values used by AODV. When a ROUTE REPLY sets up a forward path to a destination node, each node along the way assigns the route a timeout based on the timeout value returned in the ROUTE REPLY packet.

We note that unlike in MRSR, only one route will be discovered in MRDV. This route may not have the longest lifetime out of all the possible routes. Instead, MRDV avoids route error by using the discovered lifetime for this route to expire it at the appropriate time.

The destination sequence numbers in control packets ensure loop freedom and freshness of routing information. MRDV ensures wider propagation of ROUTE ERRORS by using a per destination predecessor list at each node, similarly as in AODV. The MRDV version in this study uses link layer feedback for detection of broken links similar to AODV-LL [2].

IV. METHODOLOGY

We use the Glomosim simulator [10] to evaluate the performance of MRDV and MRSR. We implemented both protocols in Glomosim and compare them to the existing implementations of DSR and AODV.

Mobility pattern: The mobility model used is as follows. As the simulation starts, each robot chooses a random destination in the given area and starts moving towards the chosen destination with a speed chosen uniformly between 1 and 5 meters/second. Note that the speed chosen is lower than those considered for MANET routing protocol comparison studies (typically 10-20 m/s) since they assume faster participants (e.g. cars). Once the robot reaches the destination, it chooses a random task from a set of tasks with a maximum task duration of p (pause time) seconds and pauses for the task's duration. After pausing it chooses the next random destination. This effectively models the movement of a group of robots performing tasks: each robot moves towards a particular area, spends some time at the location performing a task, and then moves to the next position. Although the mobility of robots in the simulation is random, the robots know their velocity, task duration and d_{rest}. We have considered pause times ranging from 900s to 0s and a simulation duration of 900s. Note that the random task selection is not done for p=900s and 0s. A pause time of 900s in a simulation of duration 900s signifies that no robots move. A pause time of 0 second signifies that the robots are in constant motion and do not stop to perform any tasks.

Traffic pattern: Each robot initiates data connections to other randomly chosen member robots. Each packet sent is 64 bytes in size and the packet rate is 1 packet every second per connection.

Energy Model: Since our wireless communication is based on IEEE 802.11, we adopt the energy model and measurements for ad hoc networks proposed in [4]. We use a wireless network interface with a 2 Mbps bandwidth.

Metrics: The following metrics are evaluated for the routing protocols: (1) Routing Overhead – The total number of control packets transmitted, with each hop-wise transmission counted as one transmission; (2) Packet Delivery Ratio – The ratio of the data received at the destination over the number of data packets transmitted by source robots; (3) Average Delay – The average delay incurred in sending a data packet; and (4) Energy Consumed – The amount of energy consumed by the the robot team for communication. This includes energy spent during sending and receiving both data and control packets.

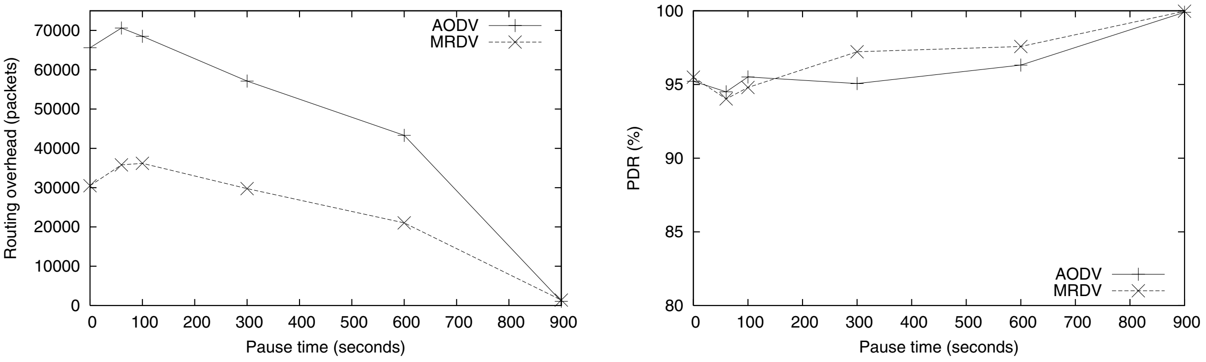
V. PERFORMANCE EVALUATION

In this section, we first compare MRDV with AODV and MRSR with DSR using the four metrics discussed above for a network of 50 nodes with varying mobility. The area chosen for 50 nodes is 1500m x 300m, similar to the studies in [2], [3] which evaluated the performance of DSR and AODV.

A. MRDV Performance

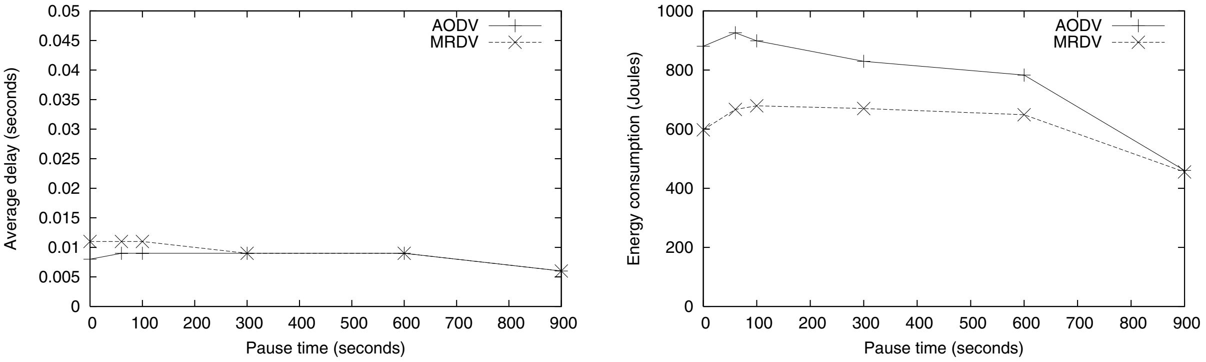
The routing protocol performance for MRDV and AODV are depicted in Figure 3. The metrics are measured by varying the pause time and thus the extent of mobility in the network. A higher pause time results in a network with lower mobility and a pause time of 900 seconds (same as the simulation duration) results in a completely static network.

The following observations can be made from the results. Firstly, as Figure 3(a) shows, MRDV reduces the overhead due to control messages significantly when compared to AODV by exploiting the mobility characteristics of mobile robot networks. As expected, when the network is static, both MRDV and AODV perform similarly. In all cases in which the network is mobile, MRDV provides a 50% reduction in overhead due to control messages when compared to AODV. More importantly, this reduction is achieved without any loss in packet delivery performance for MRDV (Figure 3(b)). In fact, for medium mobility scenarios, MRDV has a slightly higher packet delivery ratio than AODV. A high PDR is desirable since successful coordination of mobile robots require reliable delivery of commands. In addition, the average delay depicted in Figure 3(c) is also similar for MRDV and AODV. Figure 3(d) shows the energy consumption due to communication for MRDV and AODV. Due to its lower control overhead, MRDV reduces the radio usage during the transmit mode as well as the receive mode. This reduction in radio usage translates to energy savings ranging from 30% for high mobility scenarios to 18% for medium to low mobility scenarios. As expected, for a static network, both protocols consume similar amount of energy.



(a) Routing overhead

(b) PDR



(c) Avg. delay

(d) Energy consumption

Fig. 3. Comparison of distance vector routing protocols.

B. MRSR Performance

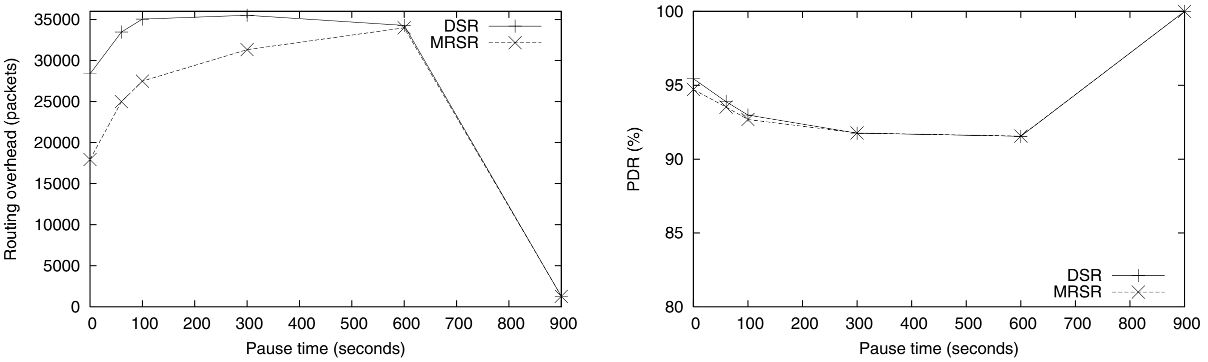
The routing protocol performance for MRSR and DSR are depicted in Figure 4. Similar to the previous section, the metrics are measured by varying the pause time and thus the extent of mobility in the network.

The following observations can be made from the results. Firstly, as Figure 4(a) shows, MRSR reduces the overhead due to control messages when compared to DSR. Note that DSR employs very aggressive caching and also uses a graph cache [5] and thus typically has low overhead. Despite that, MRSR results in a further reduction in overhead. In high mobility scenarios, MRSR provides a 30% reduction in overhead while for medium mobility scenarios the reduction is 12% when compared to DSR. As expected, when the network is static, both MRSR and DSR perform similarly. However, due to efficient caching, both protocols perform similarly for pause times ranging from 600s to 900s unlike the comparison in the previous section. MRSR achieves this reduction while maintaining a PDR similar to that of DSR (Figure 4(b)). In addition, Figure 4(c) shows that MRSR has a lower average delay in many scenarios. Figure 4(d) shows the energy consumption due to communication for both MRSR and DSR. Due to its lower control overhead and increased route lifetimes, MRSR reduces the radio usage. This energy savings obtained range from 12% for high mobility scenarios to 5% for medium mobility scenarios. As expected, for a static network, both protocols consume similar amount of energy.

C. Comparison among the protocols

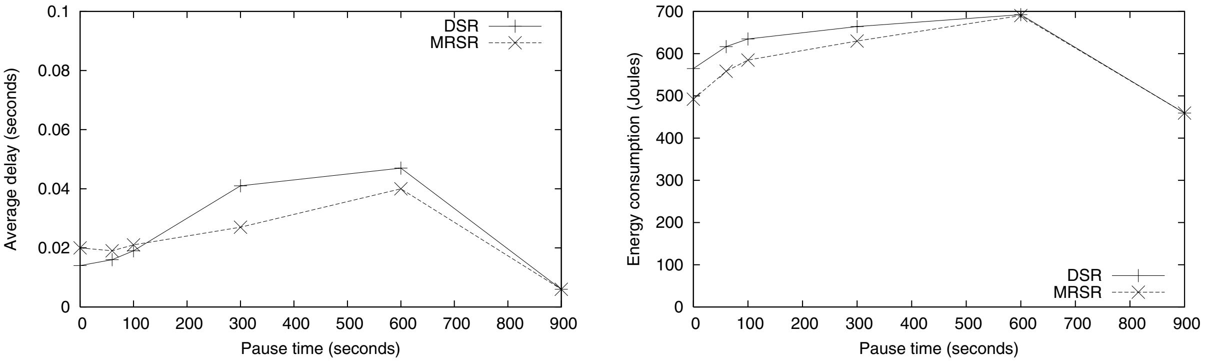
In this section, we compare the relative performance of all 4 protocols: AODV, MRDV, DSR and MRSR. For smaller mobile robot networks such as 50 nodes (evaluated in the previous sections), both MRDV and MRSR have similar performance. MRSR has a lower overhead than MRDV due to the use of aggressive caching. However, MRSR also has a higher delay than MRDV due to the use of larger packets for encoding source routes. MRDV is as efficient in energy consumption and overhead as DSR and MRSR while AODV has a much higher overhead and energy consumption. MRDV also has a slightly higher PDR than MRSR.

When the performance of the protocols in larger networks (100 nodes) was studied (not depicted due to lack of space), it was found that distance vector protocols are superior in packet delivery performance. This has been observed in earlier studies such as in [3] and has been attributed to not using source routing which increases packet sizes and congestion as the network size increases. It was also found that the overhead and energy savings from MRDV as compared to AODV increase with the network size.



(a) Routing overhead

(b) PDR



(c) Avg. delay

(d) Energy consumption

Fig. 4. Comparison of source routing protocols.

In summary, for smaller mobile robot networks, both MRSR and MRDV are attractive options with MRSR providing higher energy savings at the cost of slightly increased delay. For larger mobile robot networks, MRDV provides a superior packet delivery performance with increased energy savings compared to AODV and MRSR.

VI. CONCLUSION

In this paper, we designed and evaluated two new protocols for unicast messaging in mobile robot networks. Both protocols exploit the fact that abundant mobility information exists in a mobile robot network. Unlike typical MANETs, in which mobility cannot be predicted easily, mobile robots have specific instructions and perform specific tasks which govern their mobility and pausing behavior. Both protocols, MRSR and MRDV exploit this information in the following ways: (1) constructing longer lifetime routes through the use of robots that are likely to remain as part of the route, (2) intelligently assigning timeouts to links and routes based on the mobility information, and (3) reducing the large overhead of route discoveries by probabilistic rebroadcast of ROUTE REQUEST packets. These features allow MRSR and MRDV to provide unicast routing in mobile robot networks with lower overhead without incurring any reduction in packet delivery performance in comparison to the state-of-the-art MANET routing protocols.

Our future work consists of comparing these protocols in our testbed. We also plan to study modifications to location based unicast routing protocols for use in mobile robot networks.

REFERENCES

[1] T. Balch and R. C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52, 1994.
[2] J. Broch. et al., A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of ACM MobiCom*, October 1998.
[3] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. of IEEE INFOCOM*, March 2000.
[4] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proc. of IEEE INFOCOM*, April 2001.
[5] Y.-C. Hu and D. B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *Proc. of ACM MobiCom*, August 2000.
[6] D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic, 1996.
[7] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. of ACM SIGCOMM*, August 1994.
[8] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Proc. of IEEE WMCSA*, February 1999.
[9] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. Papanikolopoulos. Effects of Limited Bandwidth Communications Channels on the Control of Multiple Robots. In *IEEE IROS*, October 2001.
[10] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. In *PADS Workshop*, May 1998.