Connectivity vs. Control: Using Directional and Positional Cues to Stabilize Routing in Robot Networks

Karthik Dantu and Gaurav S. Sukhatme

Abstract— **Various coordination algorithms have been proposed for robot networks. One of the fundamental assumptions of such algorithms is that the underlying connectivity graph be stable. Adhoc routing protocols attempt to optimize the path from source to destination and do not guarantee route stability. We bridge this gap by providing directional and locational cues to the routing protocol to provide more stable routes. We implement our ideas on Optimized Link State Routing (OLSR), a popular proactive routing protocol for robot networks. Our results show that simple directional and locational cues can achieve up to 20% fewer route switches in comparison to the basic version of OLSR.**

I. INTRODUCTION

Robotic networks have been used in various cooperative multi-robot tasks like target tracking [8], [7], mapping [6],distributed surveillance [17] or scoring goals in a game of robot soccer [5]. Significant research effort has been expended in the control of a network of robots for achieving formations [13], coverage [11] and network topology [4], [14] etc. A basic requirement in these algorithms is that the network of robots be connected and form a graph i.e. the network links (and routes) be stable with time. This is non-trivial since the routing protocol used to establish the routes has a behavioral aspect to itself. Most routing protocols are unaware of the mobile nature of the network and attempt to optimize the end to end packet delivery mechanism. The routing protocol is not concerned with the stability of the routes that it chooses as long as a reliable path exists from source to destination. If the graph provided by the routing algorithm is to be used for control, we need to bridge this gap and make the routing protocol more stable.

A. Motivating example

Consider the robot network shown in Fig. 1. The nodes are moving as shown by the arrows on each node. Node F has a choice of three edges ($F-C, F-D, F-E$) to communicate with node B if the routing metric is shortest hop-distance. However, from the direction of motion of each robot, it is clear that route $F-C-B$ is poor since nodes B and C are moving in opposite directions. Routes $F-D-B$ and $F-E-B$ are similar, but closer perusal tells us that route $F-E-B$ is likely to be most stable. Knowledge of direction of motion (more generally, speed and direction) and location of the neighboring robots can be used to make this distinction.

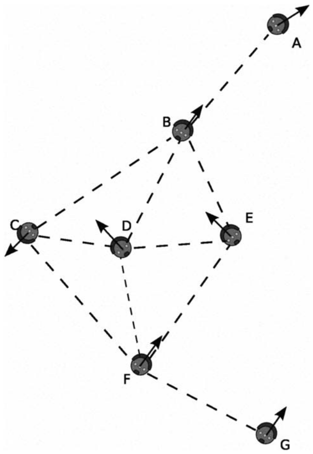


Fig. 1. Illustrating route instability

II. RELATED WORK

Connectivity in a network of robots or multi-agent systems has received close attention recently.

A. Connectivity control

Study of algebraic graph theory has given us some fundamental insights into connectivity in mobile/robot networks. [9] proposed the problem of maximizing the second smallest eigen value of the Laplacian matrix which closely follows the connectivity of the network. [4] use an exponential decay model for the connection between nodes and study a potential based control law to improve connectivity among multiple agents in a network in a decentralized fashion. [18] propose a novel control decomposition that is a hybrid of nearest neighbor potential fields to maintain existing links and coordination on the abstract topology to optimize global routes in a decentralized fashion thereby guaranteeing connectivity in the network assuming the network was connected to start with. A detailed study of the underlying graph theoretic concepts and algebraic analysis can be found in [2].

B. Topology control in mobile adhoc networks

[12] proposes a cone-based topology control algorithm under ideal radio transmission model based on the directionality of the neighbors. The claim is that if there

---

Karthik Dantu and Gaurav S. Sukhatme are with the Department of Computer Science, University of Southern California, Los Angeles, CA - 90089-02915. They can be reached at {dantu,gaurav}@usc.edu. This work was funded in part by NSF (grants CNS-0540420, CNS-0325875, CCR-0120778); ONR (grant N00014-08-1-0693); and a gift from the Okawa Foundation.

ROBOCOMM 2009, 31st Mar – 2nd Apr 2009, Odense, Denmark.
ISBN 978-963-9799-51-6
DOI 10.4108/ICST.ROBOCOMM2009.5846

is a neighbor in every $5\pi/6$ cone, network connectivity is guaranteed throughout the network. They also propose further optimizations to reduce the transmit power while maintaining connectivity. [14] generalize this concept and propose Neighbor-Every-Theta graphs (NET graphs). Their result is that the connectivity is $\lfloor \frac{2\pi}{\theta} \rfloor$ when the maximum angle between adjacent neighbors is less than θ. This result holds for irregular radio models as well. The literature in topology control in mobile networks has been studied in detail in [16].

C. Location aware routing

[10] proposed location-aided routing in mobile adhoc networks. Using position information to improve reactive routing in adhoc networks. The basic idea is to use knowledge of where a node was at a given time to do targeted flooding instead of flooding throughout the network to discover a route to it. Using an extensive study with imperfect information [10] achieves good performance while reducing the amount of control traffic required to find a route to the destination. Our work is closest to [10] in principle; the key distinction is that we are attempting to achieve stability in routing as opposed to minimizing the control traffic or achieving better routes.

III. IDEA

As mentioned earlier, stable routes are essential in robot networks. Our idea is to provide direction and location cues to the routing protocol to better choose the most stable route. However, each cue has to be broadcast in the local neighborhood and is additional overhead to the control data being passed around for the routing protocol. We first study directional cues, followed by a combination of locational and directional cues. We will explain the exact metric calculated in both the cases in the following subsections.

Fig. 1 showed an example robot network to illustrate our motivation. As mentioned in Sec. I-A, a standard adhoc routing protocol that optimizes the hop-distance could choose any of $F-C-B$, $F-D-B$ or $F-E-B$ as a route from node F to node B, but an examination of node velocities suggests that route $F-E-B$ is likely superior to the others.

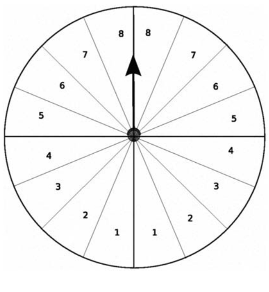


Fig. 2. Stability metric for direction cue

A. Direction Cue

The basic idea is to share the velocity of movement with the one-hop neighbors. The neighbors of each robot estimate the difference in direction w.r.t to themselves in terms of angular offset. This is translated into a number which we call the **stability metric**. This stability metric is shown in Fig. 2. We approximate the angular difference between the directions of motion of the robots (ranging from 0-180°) using a number between 1-8. This is the stability of the link. The higher the stability metric, the more stable the link. The stability metric is chosen to be symmetric about the direction of travel of the robot because the rate of separation of the robots on either side is the same. We can do better if we add location information (as illustrated in Sec. III-B).

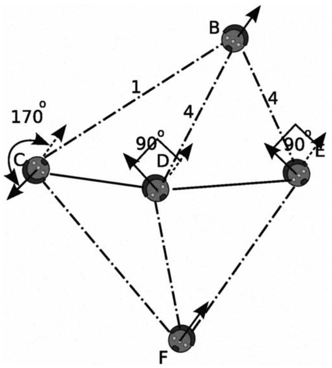


Fig. 3. Direction cue

From our earlier example in Fig. 1 routes $F-D-B$ and $F-E-B$ have the same value of the stability metric since the difference in angle between robot B and robots D and E is the same. Either one of them is better than the route $F-C-B$ and can be chosen when we just have the direction cue. This is illustrated in Fig. 3. We have superpositioned the direction vector of B on nodes C, D and E for ease of understanding. The angular difference between the direction vectors C and B is 170°. Hence, $B-C$ is given a stability metric value of 1. Angular difference between direction vectors of B and D is 90°. Hence, $B-D$ is given a stability metric value of 4. Similarly, $B-E$ is given a stability metric value of 4.

B. Position Cue

The second cue that can be given to the routing protocol is the location of the robot. In combination with the direction, this can give us an accurate estimate of how long the link will last. We propose this as the stability metric when both cues are available.

Fig. 4 shows the same robot network from our earlier example. Let robots F, B, D, E be at positions (x_f, y_f), (x_b, y_b), (x_d, y_d) and (x_e, y_e) respectively. Let their respective velocities be v_f, v_b, v_d and v_e. Given the instantaneous velocities, we can compute the relative velocity between the robot pairs (B, D) and (B, E). Given the radius of communication for node B (say R), we can predict link duration associated with each of the links $B-D$ and $B-E$ as the time taken to travel the distance (d_1 and d_2 respectively) given their velocity such that they go out of the range of communication of robot B (assuming a simple disc model

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5846
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5846

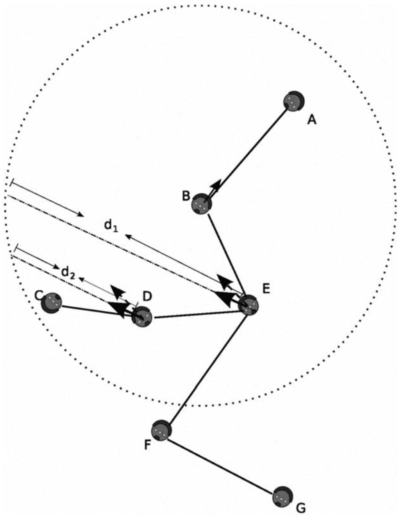
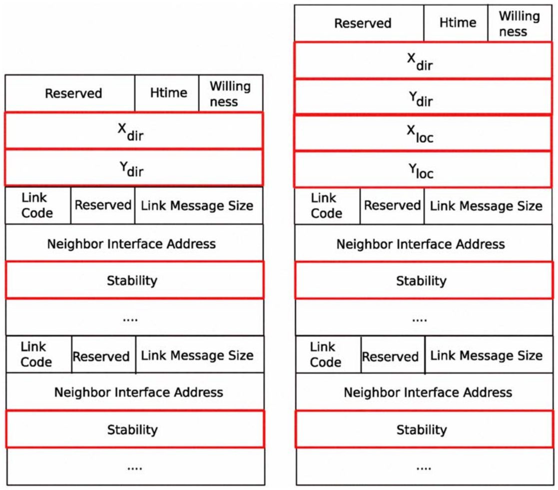


Fig. 4. Location and direction cue



(a) Direction (b) Location

Fig. 5. Hello message augmentation

for the radio). We consider the duration of the existence of the link as its stability metric.

IV. MODIFYING OLSR

We choose Optimized Link State Routing (OLSR) [3] as the routing protocol to implement our idea. We first describe the protocol and then our additions to the protocol.The additions made for both direction cues and location cues are very similar.

A. Protocol details

OLSR is a link state routing protocol with some optimizations built in to decrease the flooding of control messages. The protocol maintains the following information repositories.

- *Multiple Interface Association information base* is for networks where nodes have more than one local network interface.
- *Link information base* has information about the links of the current node.
- *Neighbor information base* has information about the neighbors with symmetric links
- *2-Hop Neighbor information base* has a set of 2-hop neighbors of the current node that have symmetric links with the neighbor set of this node
- *MPR information base* is the set of neighbors that are chosen as relays for the control messages sent by the current node. This is the chief optimization in OLSR.

All these information bases are built by broadcasting hello and topology control messages. We add a direction information base to this list to store the direction information of neighbors. For the location cues, we add a location information base. Let us assume that robot A received a hello message from its neighbor robot B. A stores the direction of B in a direction tuple corresponding to B. It then computes a stability metric w.r.t the direction of robot B and stores this value in the link information base. When node A sends out its hello messages, it adds this stability value along with the neighbor interface addresses to tell every neighbor how stable its link is with that neighbor. Location augmentation happens in a similar fashion. The hello message augmentation is shown in Fig. 5.

We add a stability parameter to the 2-hop neighbor base. Every time a robot receives a hello message, it updates its information bases correspondingly. The additional function we add is to update the link stability value in the 2-hop neighbor base. This information is used in the routing table computation.

Lastly, we add a stability value to the topology control base and the topology control messages that are exchanged throughout the network. Whenever a node receives a topology control message, it updates its stability value to be the minimum of the stability values of that route and the stability value of the link to the robot that forwarded this message.

B. Routing Table Computation

From [3], the routing table computation occurs every time there is a modification to any of the local information bases. We modify step 3 of [3] where the routing table computation is based on the 2-hop neighbors. Instead of picking the first neighbor that has a link to the 2-hop neighbor we run through the list of neighbors to pick the neighbor with the highest stability metric as the next hop to route to that 2-hop neighbor. If there was a prior route to this 2-hop destination, we only pick a new route if the stability value is below a certain minimum threshold.

We also modify the step 3.1 of [3] where routes are calculated/updated based on the topology control information base. Similar to the 2-hop case, we only modify a route if the stability value of that path has fallen below a minimum threshold.

V. SIMULATION RESULTS

We studied the performance of our algorithm in ns-2 - a network simulator([1]). We used the OLSR implementation

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5846
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5846

from the University of Murcia, Spain [15] and modified it as described in the previous section.

A. Simulation setup

We deploy robots in a 500x500 area. We vary the number of robots to vary the connectivity in the network. Mobility of the robots is assumed to be using the random way point mobility model. Each robot assumes a random direction motion and a uniformly distributed random speed between 0 and max speed. We vary this max speed to study the effect of the speed of movement on route stability. We average each result shown over ten iterations. The propagation model for the channel is assumed to be two ray ground. We set the transmit and receive thresholds so as to have a communication of approximately 50 units. The metric we measure is the difference in route switches. We calculate this as a percentage of the number of route switches in the basic OLSR protocol. This gives us a good measure of how much our cues are helping in stabilizing the routing. We vary the max speed of travel and the number of nodes in the system.

B. Results

Fig. 6 shows us the results when we just use directional cues. Fig. 6(a) shows the effect of density on route stability. As the network becomes denser, the number of possible choices for routes is higher and our cues help pick the more stable route. From our simulations, we can get an average of up to 10% fewer route switches. Further increase in density does not necessarily improve the route stability by much. Fig. 6(b) shows the effect of change in max speed of robot travel on route stability. The faster robots can move, the lesser the benefit of the directional cues. This is expected since the routes last for much smaller durations when the robots are moving faster. Choosing one route over another has little benefit if the robots are moving rapidly. The benefit rapidly deteriorates and in networks where the robots are moving at high speeds, there is little benefit in using directional cues.

Fig. 7 is a graph illustrating the benefits of providing both positional and directional cues. With both these pieces of information, we can more accurately determine the stability of a link. This results in better choice for the routes and correspondingly more stable routes. This is shown in Fig. 7(a) and Fig. 7(b). We can achieve up to an average of 20% fewer route switches with both these cues.

C. Effect of error

We introduced error in the announced direction of motion and position for robots to understand the repercussions of inaccurate localization and direction information. We introduce additive white Gaussian noise with a variance of a percentage of the radius of communication for both the 'x' and 'y' for position error. For direction error, the error introduced is additive white Gaussian noise with a percentage of each component as the variance.

The effect of error is shown in Fig. 8. Fig. 8(a) shows the effect when only positional error is introduced. The x-axis shows the percentage error as described above. The effect of position error is minimal on the route stability.

However, the effect of direction error is significant. From Fig. 8(b), the benefit of the cues rapidly falls off with increase in direction error. This is understandable as the direction is the primary basis of our calculations in picking a more stable route. Fig. 9 and Fig. 10 show the effect of 10% direction error on various settings for max speed and density. Route stability goes down a little as is to be expected with the introduction of error.

D. Discussion

Directional cues on their own appear to give us about 2-8% decrease in route switching. If we want to achieve better stability, we will need localization of the robots in a global frame. This is a more stringent constraint and needs additional sensors like GPS or some external mechanism using which we can accurately position the robots. However, it is promising that by providing simple cues we can stabilize routing and reduce route switching by up to one-fifth.

We have studied OLSR which is a pro-active routing protocol. We envision that we will get at least similar and possibly better results on a reactive routing protocol. This is because there is no constant state being maintained in all the nodes in reactive routing protocols. If a route is broken, the source has to reissue a route request and discover the path again. If position and direction information of neighbors is available, the robot whose edge was disconnected could find a route to the neighor that just got disconnected instead of having to re-discover the whole path.

VI. CONCLUSIONS

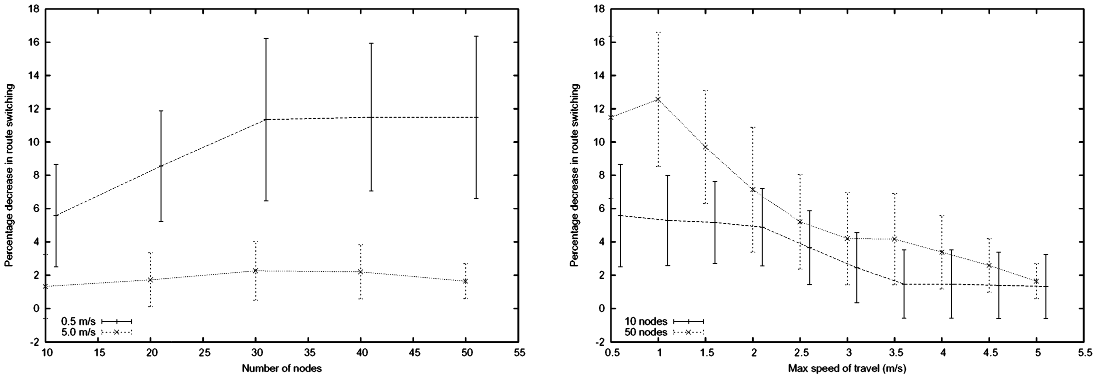
We identify the problem of route stability in robot networks. We suggest directional and positional cues as information that can be used by the routing protocol to achieve more stable routes. We have incorporated our idea into OLSR, a popular routing protocol for wireless networks. We show the exact changes needed to be made to the routing protocol to instantiate our idea. Our preliminary investigation shows that providing directional and positional cues are beneficial in stabilizing routing. We can achieve up to 20% fewer route switches using both directional and positional information.

Our instantiation of OLSR makes changes to the way routes are calculated. This is in direct conflict with route optimality. Our future work is going to investigate how we can ameliorate this conflict and choose the best possible routes (for any routing metric) while providing stable routes for other purposes like connectivity and topology control. Another future work is to test this routing in real robot networks and evaluate the benefit. This will validate the generality of our idea going beyond the mobility models used here.

REFERENCES

[1] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. Advances in network simulation. *Computer*, 33(5):59–67, 2000.

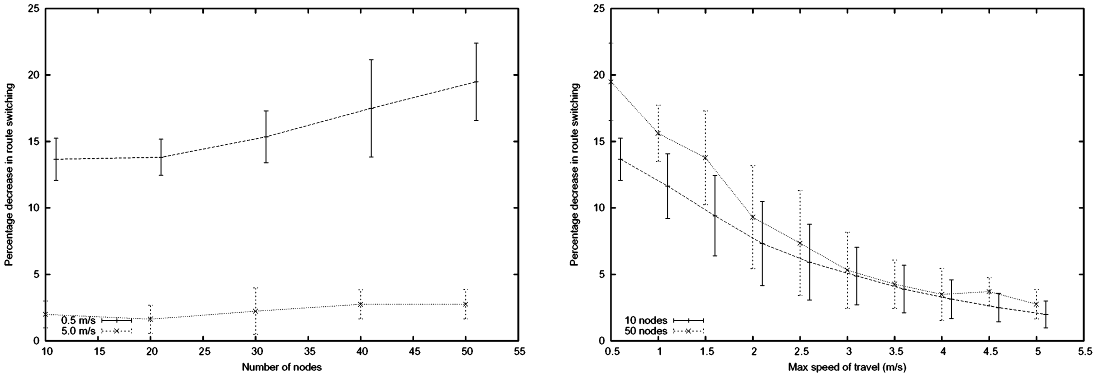
Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5846
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5846



(a) Density vs. route switching

(b) Speed of travel vs. route switching

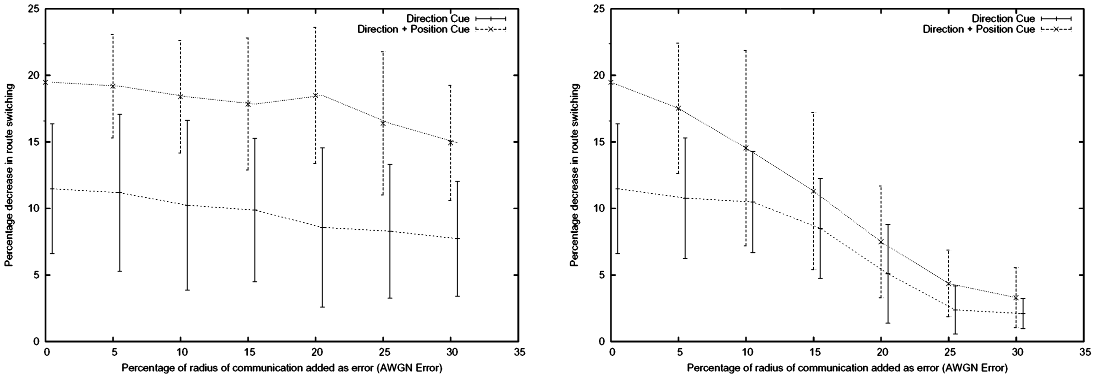
Fig. 6. Effect of density and speed of movement on route stability (direction cue only)



(a) Density vs. route switching

(b) Speed of travel vs. route switching

Fig. 7. Effect of density and speed of movement on route stability (direction and location cues)

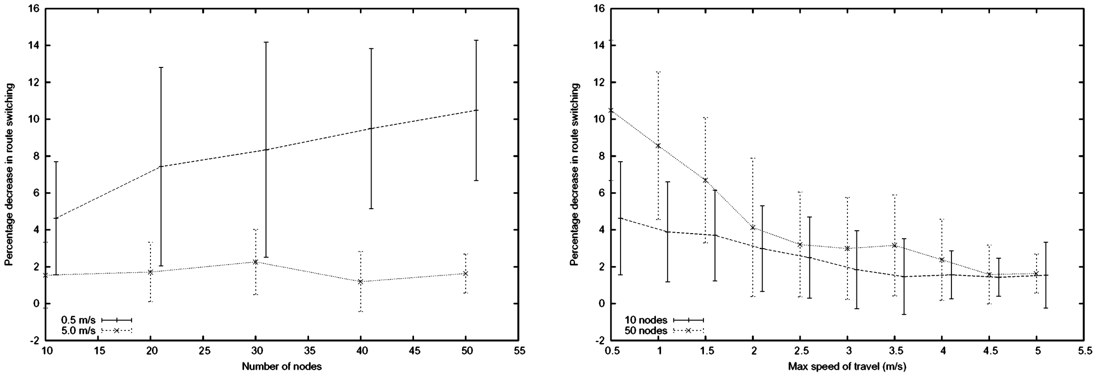


(a) Route switching with location error only (error added is zero mean Gaussian noise with variance of a percentage of the radius of communication as shown on x-axis)

(b) Route switching with direction error only (error added is zero mean Gaussian noise with variance of a percentage of the velocity components in each of the x and y directions)

Fig. 8. Effect of error on route stability (Number of nodes=50, max speed=0.5 m/s)

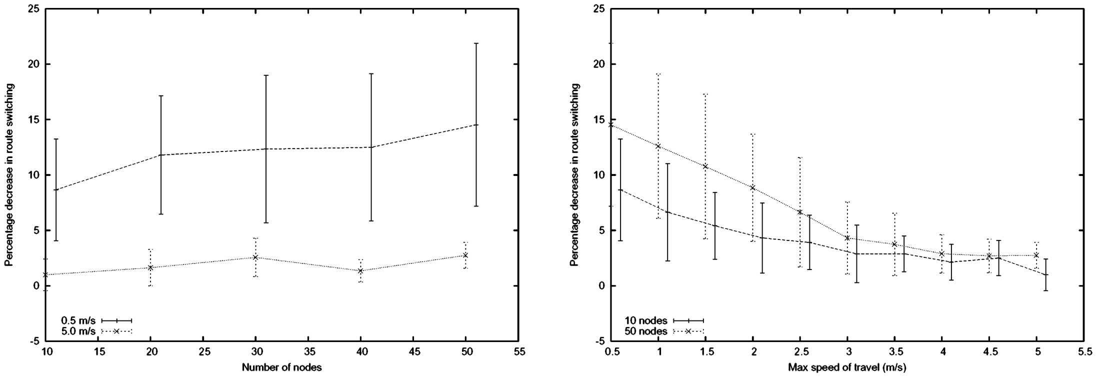
Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5846
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5846



(a) Density vs. route switching

(b) Speed of travel vs. route switching

Fig. 9. Effect of error (direction cue only): Error added is zero mean Gaussian noise with variance of 10% percentage of the velocity components in each of the x and y directions



(a) Density vs. route switching

(b) Speed of travel vs. route switching

Fig. 10. Effect of error (direction and location cue): Error added is zero mean Gaussian noise with variance of 10% percentage of the velocity components in each of the x and y directions

[2] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. To appear. Electronically available at http://coordinationbook.info.

[3] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). RFC 3626, October 2003.

[4] M. De Gennaro and A. Jadbabaie. Decentralized control of connectivity for multi-agent systems. In *45th IEEE Conference on Decision and Control*, pages 3628–3633, Dec. 2006.

[5] D. S. G. Lakemeyer, E Sklar and T. Takahashi. *RoboCup 2006 Robot Soccer World Cup X*. Springer, 2006.

[6] A. Howard. Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, 25(12):1243–1256, 2006.

[7] A. Howard, L. E. Parker, and G. S. Sukhatme. Experiments with large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research*, 25(5):431–447, May 2006.

[8] B. Jung and G. S. Sukhatme. Tracking targets using multiple robots: The effect of environment occlusion. *Autonomous Robots*, 13(3):191–205, Nov 2002.

[9] Y. Kim and M. Mesbahi. On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. *Automatic Control, IEEE Transactions on*, 51(1):116–120, Jan. 2006.

[10] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[11] K. Li and J. Baillieul. Problems in decentralized sensor-actuator networks. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 3207–3212, Dec. 2005.

[12] L. Li, J. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. A cone-based distributed topology-control algorithm for wireless multi-hop networks. *Networking, IEEE/ACM Transactions on*, 13(1):147–159, Feb. 2005.

[13] N. Moshtagh, N. Michael, A. Jadbabaie, and K. Daniilidis. Bearing-only control laws for balanced circular formations of ground robots. In *Proceedings of Robotics: Science and Systems*, Zurich, Switzerland, June 2008.

[14] S. Poduri, S. Pattem, B. Krishnamachari, and G. S. Sukhatme. Using local geometry for tunable topology control in sensor networks. *IEEE Transactions on Mobile Computing*, Feb 2009.

[15] F. Ros. Um-olsr: Reference olsr implementation from u-m, 2005.

[16] P. Santi. *Topology Control in Wireless Ad Hoc and Sensor Networksk*. Wiley, 2005.

[17] M. Saptharishi, C. Spence Oliver, C. Diehl, K. Bhat, J. Dolan, A. Trebi-Ollennu, and P. Khosla. Distributed surveillance and reconnaissance using multiple autonomous atvs: Cyberscout. *Robotics and Automation, IEEE Transactions on*, 18(5):826–836, Oct 2002.

[18] M. Zavlanos and G. Pappas. Distributed formation control with permutation symmetries. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2894–2899, Dec. 2007.