Sensor Network Mediated Multi Robotic Traffic Control in Indoor Environment

D. V. Karthikeya Viswanath, K. Madhava Krishna
Robotics Research Center,
International Institute of Information Technology,
Hyderabad, India
viswanath@research.iiit.ac.in , mkrishna@iiit.ac.in

Abstract— **This paper presents a methodology for coordination of multiple robotic agents moving from one location to another in an environment embedded with sensor motes. Sensor motes placed at strategic locations such as intersections coordinate robots in a way as to minimize the congestion, thus ensuring the continuous flow of robot traffic. A robot's path to its destination is computed by the network in terms of the next waypoints to reach and local navigation to the next waypoint is achieved through a reactive navigation system. The motes are capable of identifying robots in their proximity based on signal strength. A mote controls the flow of agent traffic around it with the help of the data it collects from the messages received from the robots and other surrounding motes in the network. The coordination of traffic is achieved through two methods, one that assigns priorities to pathways leading to the intersection based on the traffic density and the other based on the popular method of reservation. Extensive comparisons show the performance gain of the current method over existing ones. Transferring the burden of coordination to the network releases more computational power for the robots to engage in critical assistive activities.**

I. INTRODUCTION

Sensor network mediated robot navigation has become popular [1-3] in recent years from different viewpoints. Firstly the network acts as a computing medium thereby reducing the computational payload on-board the robot. In a manner akin to swarm robotics where each of the individual entity has limited intelligence but the group in itself behaves as a sufficiently intelligent system, sensor network allows the robotic agent to be possessed with minimal decision making capabilities but the network plus the robot behaves as a system of enhanced intelligence. Secondly the network provides for fault tolerance capabilities for if the onboard sensors fail or misbehave the robotic agent can look up to the sensor network for information about the environment. Thirdly the network supplements the computational capacity of the robot. Efficiently designed sensor fusion algorithms can agglomerate intelligence gathered through onboard as well as off board resources to come up with robust decisions.

The essential novelty of this work is that among the survey of papers on a similar theme the authors have not come across one that provides for multi agent traffic control in a world mediated by sensor network. While single robot navigation in a sensor network is well studied [1,2] there has been little in the area of multi robotic navigation in a sensor network. The performance gain of this method over existing methods of traffic coordination is also reported.

In this paper we describe the problem of coordinating multiple robot agents by sensor motes placed at critical locations such as doorways, intersections and T-junctions (figure 1). The nodes coordinate movement of robots across these junctions in two ways. In the first method the positions for the next T instants for each robot approaching the intersection along various pathways is computed. The flow of traffic is coordinated by assigning priorities to pathways based on the density and its rate of change. In this method the agent hailing from the pathway having the highest agent density and lowest rate of change is allotted the highest priority and the paths of agents with lower priority are attuned to accommodate the paths of the robot with higher priority. In the second the popular method of reservation [5,6] is dovetailed to the current situation. In this method a driver agent requests for a space-time allocation at the intersection. The intersection agent (sensor mote in this context) allows passage for the robot till the point of no conflict in the path of the vehicle through the intersection. Individually each robot's path in the environment is computed at the highest level as a sequence of waypoints to the goal, each successive waypoint one hop less than the previous.

In the area of multi-agent traffic control the work reported in [4,5] is relevant here. However the difference being in the current method coordination is achieved by considering density and rate of change of density along the incoming pathways to an intersection while [4,5] relies largely on a system of reservation of grids at an intersection based on a first come first served like policy. In [6] a mechanism for coordination between various intersection agents through an evolutionary agent paradigm was presented. The chief advantage of this method over purely multi agent based traffic controllers is that the network motes have data that can model more accurately the density and rate of change of it along a pathway. This is because robotic agents interact with motes at a more basic and active level and hence the obtained data can be used more profitably for reducing time spent at intersections when compared with methods such as [4,5] as the simulation section reveals. Moreover, in this method of traffic control the sensor network takes care of the entire routing of the robots with little involvement of the robots themselves.

II. PROBLEM FORMULATION

Given: A planar world, typically indoors, embedded with sensor motes. Robot agents crisscross this world. The map of the environment is unknown.

Objective: Guiding the robots to their respective destinations. During the process of navigation they often cross intersections, meeting points and junctions. The objective is to have a network mote placed at such intersections that coordinate the traffic such that the sum over the time spent by each agent while crossing the junctions is minimized.

Assumptions:

a. The robotic agents through inter agent coordination reach the intersections in a single file.

b. Number of agents is not fixed and they can be introduced in any pathway till such time there is no place to spawn any further due lo lack of space or congestion in that pathway.

c. The motes are capable of identifying robots in their proximity based on the strength of signal received from the robot.

d. The motion of robot agents is modeled as integer multiple of the resolution of a cell for every time sample. The cell distance is such that the agents can modify their kinematics to move by that distance or multiple of it between any two time samples. The time interval between successive samples is the same throughout.

e. Each mote is programmed to store the IDs and directions of its surrounding motes that are one hop count away from it.

Assumption 'a' has more to do with that indoor setting for which the problem is considered where it's difficult to have more than one line of traffic for one direction along corridors, hallways and at entrance through doors. It is evident from subsequent sections that the coordination mechanism can be scaled up to handle multiple lanes of traffic along a direction.

Assumption 'b' is often used in agent community [5,6]. It serves as a yardstick for evaluating the control mechanism. It is a welcome assumption more than anything.

Assumption 'c' is routinely used in sensor network community [1,2] to detect the event if the robot has come close enough to a sensor mote to send the next action from the mote.

Assumption'd' is used to reduce the search space over the possible velocities of robots. It is once again a common theme in several discrete time optimization problems that involve discretizing a large state space in both Multi robotic [7] & single robotic planning setting [8]. Moreover it provides an easy way to test collision by looking for space time overlays in cells without compromising the original philosophy of the coordination algorithm. Path discretization for collision checking is not uncommon either [7, 8].

Assumption 'e' comes into picture when a robot is to be directed by the mote to its next waypoint and when the mote should calculate the density of robots coming its way.

III. THE METHODOLOGY

A robot can be guided to its destination in an environment embedded with sensor motes using the hop count distance method [1]. Packets are sent from the mote closest to the goal area and are received by motes that are within one hop of the sent mote. The sender packet consists of the mote's ID and hop count. The receiver mote updates its hop count if the hop count of the received packet is less than its current hop value and only then does it broadcast it to others after incrementing the updated hop-count by 1. Thus each mote stores its hop count distance from the Goal. The robot when wanting to reach a destination sends a query message to the network, the mote which receives this query message guides the robot in the direction of a surrounding mote closest to the robot's destination. This process repeats till the robot reaches its destination. Figure 1 shows the path of a single robot from its starting position START to its destination GOAL.

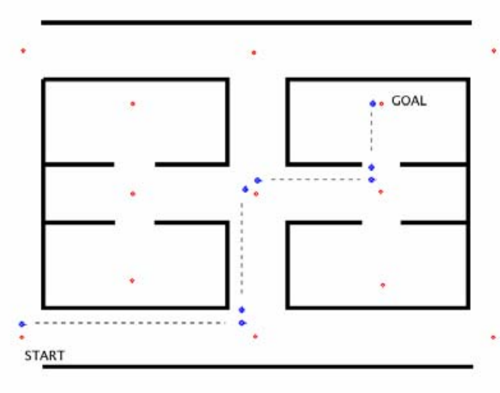

Figure 1: The path of a single robot guided from its starting position to its goal by the sensor network. The red dots indicate the motes deployed. The blue circle is the robot.

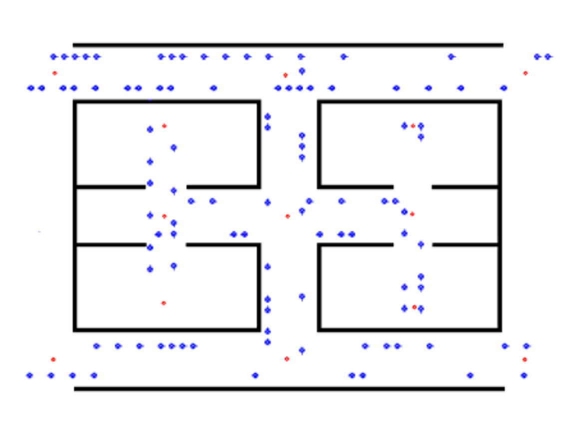

Figure 2: Multiple robots being guided by the sensor network.

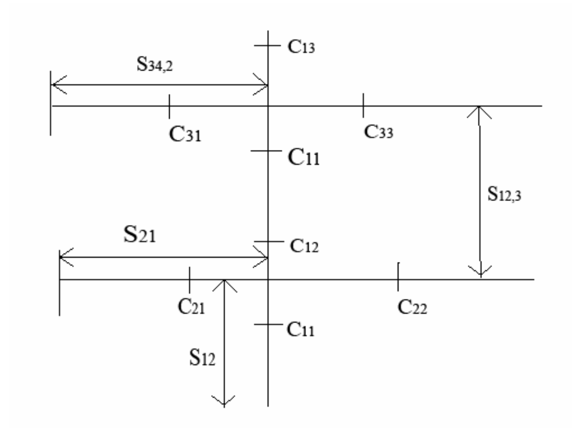

Figure 4: An Intersection

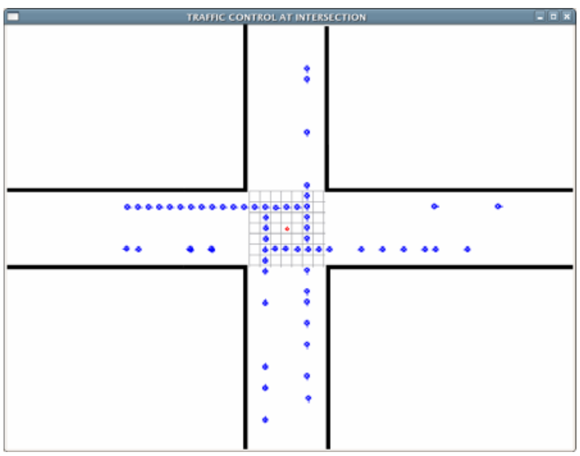

Figure 3: Multiples robots approaching an intersection continuously leading to congestion.

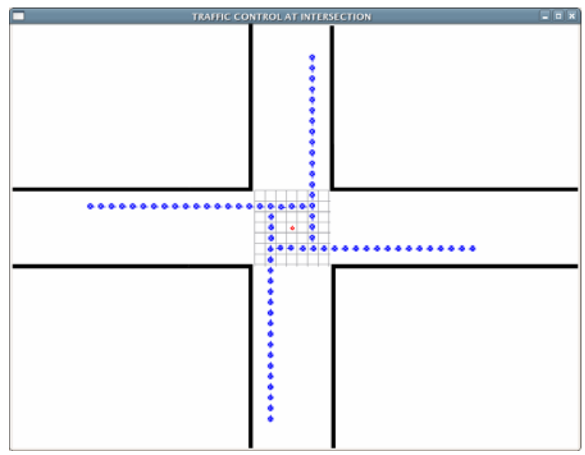

Figure 5: A Deadlock Scenario.

A. Motivation for Traffic Control

When the sensor network based navigation is scaled to multiple robots then the situation would be as shown in Figure 2. It is evident from the figure that the traffic needs coordination so that there are no collisions and congestion is minimized. Consider vehicles approaching an intersection with maximum speeds and without respite/continuously. Clearly such a situation would lead to congestion at the intersection (Figure 3) thus curbing the free flow robots. It is an undesirable situation leading to considerable loss in efficiency and productivity. If there is absolutely no control over traffic then the worst case scenario arises in which all the robots are trapped in a deadlock and there wouldn't be any movement in the traffic at all. (Figure 5) Consider an intersection such as the one in Figure 4, in indoor environments where the question of an overpass doesn't arise with vehicles moving at maximum speeds *v*.

The vehicles should arrive at the area common to both Δt apart, where Δt is the time taken to cross by vehicle along pathway 1(2) from position marked C_{11} (C_{12}) to C_{12} (C_{21}). This allows for a continuous flow rate and describes the best possible situation. The computation of the collision area and positions C_{11}, C_{12} etc is described in [9]. This constraints the vehicles along the same pathway to be at least $v\Delta t$ apart. Any future placement of intersection such as between 1 & 3 requires the distances to be an odd integer multiple of $v\Delta t$ apart for allowing for unhindered flow of max velocity traffic. In other words the distance

$$S_{34,2} = S_{12,3} \pm (2n+1)\,\Delta t,\ n = 0, 1, 2\ldots\ldots$$

Evidently these constraints are often impossible to meet for several reasons that lead to a coordination mechanism. In the presence of turns this mechanism is all the more inevitable.

B. Priority Ordering

The intersection receives messages about the robots which are approaching it from other motes which have been the previous waypoints of a robot's route to its destination. The intersection mote thus knows the density of robots approaching it in each of the pathways. The mote maintains the list of robots corresponding to a pathway; the list is updated every time a new robot comes its way. The intersection mote also calculates the rate of change in densities from the list of robots it has. Having aggregated info from all the pathways it assigns priorities to them. First the pathways are clustered based on the density values as high density and low density clusters. Among the clusters with high density the pathways are ranked on increasing order of rate of change of density. This process is repeated for clusters classified as low density clusters. Thus the pathway with highest density and lowest rate of change of it gets the top most rank or priority since this is a situation corresponding to congestion. Within a pathway the agents are ranked based on their closeness to the intersection. The first n_a number of them are ranked and then ranking of agent in the next pathway is proceeded. All the agents in a lower ranked pathway have ranks lower than those in a higher priority pathway. We say agent a_x has a lower rank than agent a_y if the value of the rank of a_x, $r(a_x)$ is actually higher than the value of rank of a_y, $r(a_y)$.

The path of the agent with highest priority is left as such for the next T instants if it does not collide with those already crossing the intersection. If there collisions the path of highest priority agent is adapted with those already at the intersection. The paths of those with lower priority are then modified to avoid collision with those higher in the priority. Paths & collisions are computed for a look ahead duration of T samples. However this process is repeated every τ samples, $\tau < T$ by the mote. Computed paths at end of every τ samples is transmitted to the robots by the mote.

C *Reservation*

In the reservation method the priorities are assigned based on a first come first served basis. The intersection mote receives request for space reservation. The agent whose request is first received is allotted the highest priority. The agent whose request is received next becomes the second highest. Whenever a request is received the mote computes the path of that robot and sees if it is collision free. If no collisions are detected it grants the request, else it computes the path till the cell just ahead of collision. The intersection mote thus computes collision free paths based on this order and guides agents through the intersection. However if a request is received from an agent in the pathway before the agent in which is ahead of it in that pathway the intersection manager rejects the request. This makes sure within a pathway the agent first in the file always receives the highest priority in that pathway. In this policy also the path of an agent is calculated for a look-ahead duration of T samples.

IV. SIMULATION

A. The Simulator

We have developed a simulator which can precisely simulate the mote-mote communication as well as mote-robot communication which is essential for the proper coordination of robot traffic. Using the assumptions mentioned in the earlier sections the simulator was built to model the discrete motion of the robots. The simulator contains two main modules, one that simulates the global robot guidance and the second that simulates the traffic at a single intersection.

The intersection area can accommodate up to 25 robot agents at a time. This number is dependent upon the size of the intersection and the size of the robots. These parameters could be accurately modeled in our simulator. For deriving the results shown we have set the size of the intersection and robots such that the number of robots which can be in an intersection is 25. However, it doesn't mean that 25 robots could always move in a continuous manner at the intersection, it can be done only with a proper traffic coordination policy.

When a robot's path is blocked it has to be halted and the robots that are dependent on the current robot's movement are halted as well. So the obvious metric for evaluating the two policies is the average number of robots stopped over a period of time. The more the number of robots stopped the lesser is the efficiency of the policy. Statistics were collected after a simulation of 10,000 time samples.

B. Simulation Results

Figure 6 shows the statistics for the two policies for different look-ahead times. The 'Avg' column in this figure shows the average number of robots that were halted in their path per sample to accommodate robots of higher priority when averaged over 10,000 samples. The 'Max' column shows the highest number of robots that were halted in a single time sample over 10,000 samples. Column 1 shows the numbers of robots approaching the intersection from the various pathways (here four) for which the 'Avg' and 'Max' values are computed. It is evident from the statistics that the priority ordering policy fares better than the reservation policy of traffic control. The simulation tests have shown that the robot density based priority ordering policy minimizes the congestion at the intersection reducing considerably the possibility of a deadlock.

	T = 1				T = 3				T = 5			
Number of Robots	Reservation		Priority Ordering		Reservation		Priority Ordering		Reservation		Priority Ordering	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
10	0	1	0	1	0	1	0	1	0	1	0	1
20	0.4	4	0.4	4	0.3	4	0.2	3	0.26	4	0.18	3
30	8.3	12	5.1	8	6.2	10	2	6	5.3	8	1.76	5
40	15.7	20	7.5	16	10.8	15	3	10	9.5	13	2.5	8
50	30.1	40	15.2	25	20.8	32	8	20	15.2	28	5.9	15
60	DEADLOCK		30.6	40	35	40	15.3	35	24.8	35	12	29
70			DEADLOCK		DEADLOCK		25	50	39.2	52	18	40
80							40	70	DEADLOCK		35	60

Figure 6: The Results obtained for different look-ahead time samples T and for different number of robots approaching the intersection. The 'Avg' column shows the average number of robots that were halted in their path to accommodate robots of higher priority. The 'Max' column shows the highest number of robots that were halted in a single time sample.

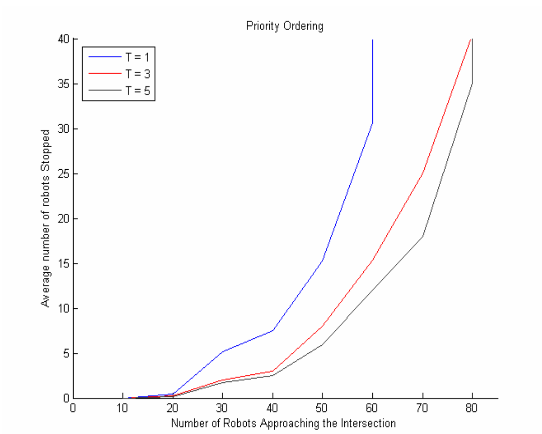

Figure 7: Graph showing the average number of robots stopped at the intersection for different look-ahead time samples.

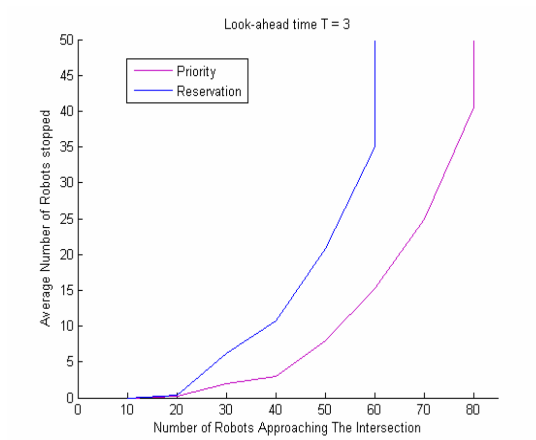

Figure 8: Graph illustrating the efficiency of priority ordering based policy over the reservation based policy.

Figure 7 illustrates that efficiency improves when the look-ahead time for planning increases. It shows data collected for different look-ahead time samples for the priority ordering policy. The average number of robots stopped is minimal in the case of maximum T (5 in this case). Figure 7 is a comparison graph which plots the data collected for priority ordering policy. Comparison is done between the average number of robots halted when the look-ahead time is 1, 3 and 5 respectively. Clearly the system fares better i.e. congestion is less when motion planning is done for a higher look-ahead time. Figure 8 compares the data collected for the two control policies for the same look-ahead time. It is evident that the priority ordering policy shows a much better performance than the reservation policy.

It is to be noted that while the current method guarantees a better policy of congestion management and deadlock prevention it does not per se rule out the occurrence of such phenomena.

V. CONCLUSIONS and SCOPE

A new methodology has been proposed for coordinating robotic agent traffic by a sensor network in an indoor environment. The method is based on allotting priorities to pathways depending on the density of agents and rate of change of it in those pathways. The present method coordinates traffic flow better and reaches deadlock situations for a far higher number of robots in the pathways over the existing method of reservation adapted to sensor network framework. The number of robots need to be halted at the intersection is also lesser in the current method consistently over several runs of simulation. The paper describes a complete system in which multiple robots are guided to their respective destinations while reducing the congestion in the robot traffic at intersections. No topological map of the environment is used in the system. In a situation where a number of robots are crisscrossing an area such as in ant colonies the current method of reducing congestion finds utility. Outside a robotic domain it also shows a means better than the current methods of traffic control such as on roads if a suitable mechanism of measuring densities is possible.

In this paper we proposed ideas for minimizing congestion at intersections; congestion control is being done at a local level, control mechanism at one intersection is not affecting the control mechanism in another intersection. If we look at the traffic control at a global perspective and include the global traffic movement information at each intersection then the efficiency of the system can be improved a lot. Since a sensor network works in a decentralized manner it helps our case in which traffic information of one intersection has to be sent to another intersection. Our on-going research is being carried out in similar lines.

REFERENCES

[1] Q. Li, M. DeRosa, and D. Rus, "*Distributed algorithms for guiding navigation across a sensor network*," in The 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03), 2003.

[2] Maxim Batalin, Gaurav S. Sukhatme, and Myron Hattig, "Mobile Robot Navigation using a Sensor Network," In *IEEE International Conference on Robotics and Automation*, Apr 2004.

[3] Peter Corke, Ron Peterson and Daniela Rus, "Localization and Navigation Assisted by Cooperating Networked Sensors and Robots" In *International Journal of Robotics Research,* 2005.

[4] Kurt Dresner and Peter Stone, "Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism" In *The Third International Joint Conference on Autonomous Agents and Multiagent Systems*

[5] Kurt Dresner and Peter Stone, "Multiagent Traffic Management: An Improved Intersection Control Mechanism" In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*

[6] A. L. C. Bazzan, "A distributed approach for coordination of traffic signal agents." *Autonomous Agents and Multi-Agent Systems*, 10(2):131–164, March 2005.

[7] Bennewitz M., Burgard W. and Thrun S., (2002). "Finding and Optimizing Solvable Priority Schemes for Decoupled Path Planning Techniques for Teams of Mobile Robots." *Robotics and Autonomous Systems*, 41 (2), 89-99

[8] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[9] K Madhava Krishna, Henry Hexmoor and S Chellappa, "Reactive Navigation of Multiple Moving Agents by Collaborative Resolution of Conflicts", *accepted in Journal of Robotic Systems*, 22(5): 249-269, 2005.