In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS 05), pp. 471-477, Utrecht, The Netherlands, July 2005.

Multiagent Traffic Management: An Improved Intersection Control Mechanism

Kurt Dresner and Peter Stone
University of Texas at Austin
Department of Computer Sciences
Austin, TX 78712 USA
{kdresner, pstone}@cs.utexas.edu

ABSTRACT

Traffic congestion is one of the leading causes of lost productivity and decreased standard of living in urban settings. Recent advances in artificial intelligence suggest vehicle navigation by autonomous agents will be possible in the near future. In a previous paper, we proposed a reservation-based system for alleviating traffic congestion, specifically at intersections. This paper extends our prototype implementation in several ways with the aim of making it more implementable in the real world. In particular, we 1) add the ability of vehicles to turn, 2) enable them to accelerate while in the intersection, and 3) augment their interaction capabilities with a detailed protocol such that the vehicles do not need to know anything about the intersection control policy. The use of this protocol limits the interaction of the driver agent and the intersection manager to the extent that it is a reasonable approximation of reliable wireless communication. Finally, we describe how different intersection control policies can be expressed with this protocol and limited exchange of information. All three improvements are fully implemented and tested, and we present detailed empirical results validating their effectiveness.

1. INTRODUCTION

Traffic congestion is one of the leading causes of lost productivity and decreased standard of living in urban settings. According to a recent study of 85 U.S. cities [17], annual time spent waiting in traffic has increased from 16 hours per capita to 46 hours per capita since 1982. In the same period, the annual financial cost of traffic congestion has swollen from $14 billion to more than $63 billion (in 2002 US dollars). Each year, Americans burn approximately 5.6 billion gallons of fuel while idling in heavy traffic. Recent advances in artificial intelligence suggest that autonomous vehicle navigation will be possible in the near future. Individual cars can now be equipped with features of autonomy such as cruise control, GPS-based route planning [13, 15], and autonomous steering [9, 11]. Once individual cars become autonomous, many of the cars on the road will have such capabilities, thus opening up the possibility of autonomous interactions among multiple vehicles.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

Multiagent Systems (MAS) is the subfield of AI that aims to provide both principles for construction of complex systems involving multiple agents and mechanisms for coordination of independent agents' behaviors [16]. In an earlier paper, we proposed an MAS-based approach to alleviating traffic congestion, specifically at intersections [3]. In this paper, we describe several ways in which we have transformed that system into a more realistic and implementable system.

Current methods for enabling traffic to flow through intersections include building overpasses and installing traffic lights. However, the former is very expensive and forbids turning, while the latter can be quite inefficient, often requiring cars to remain stopped even when no cars are present on the intersecting road.

At this time, it is possible to create a small-scale system in which all cars are piloted by a central computer. Consider, for example, the task of controlling ten vehicles on an open factory floor. However, growing such a system to handle an intersection in which a city's worth of cars might turn up would involve prohibitively expensive and inefficient communication and control infrastructure. Here we aim to maximize the efficiency of moving cars through intersections with minimal centralized infrastructure. We assume that intersections can be outfitted with a simple wireless communication system and a protocol (which we introduce here) for communicating with oncoming traffic and giving permission for cars to pass.

In our system, vehicles must traverse intersections according to a set of parameters agreed upon by the vehicle and the intersection manager (as they do today by obeying red and green lights), but otherwise are free to decide for themselves how to drive. Each car is an autonomous agent, and in particular need not surrender control to any centralized decision maker.

Given the above assumptions, we have proposed a novel reservation-based system by which cars request and receive time slots from the intersection during which they may pass [3]. While this system showed the potential for a reservation-based system to drastically improve the efficiency of intersections, it required driving agents to maintain a constant velocity in the intersection and forbade turning (a very important part of intersections). Furthermore, it did not adequately specify how they should interact. In this paper, we take three large steps towards making the system implementable in the real world. First, we augment it to allow turning. Second, we make acceleration in the intersection possible, which allows us to subsume the stop sign policy within the reservation framework. Third, we specify a protocol to govern the interactions of the vehicles and the intersection such that the vehicles do not need to know anything about the intersection control policy. The use of this protocol limits the interaction of the driver agent and the intersection manager to the extent that it is a reasonable approximation of reliable wireless communication. Using this protocol, we detail how many everyday intersection control policies, such as the traffic light and the stop sign can be encoded.

2. THE ORIGINAL SYSTEM

Previously, we proposed a novel reservation-based multi-agent approach to alleviating traffic, specifically at intersections. This system consisted of two types of agents: *intersection managers* and *driver agents*. Each system consists of an intersection manager for each intersection and a driver agent for each vehicle. Intersection managers are responsible for directing the vehicles through the intersection, while the driver agents are responsible for controlling the vehicles to which they are assigned. To improve the throughput and efficiency of the system, the driver agents "call ahead" to the intersection manager and request space-time in the intersection. The intersection manager then determines whether or not these requests can be met. Depending on the decision the intersection manager makes, the driver agent either records the parameters of the request (the *reservation*) and attempts to meet them, or it makes another request at a later time.

To determine whether or not a request can be met, the reservation manager simulates the journey of the vehicle across the intersection, which it divides into a grid of $n \times n$ tiles. The parameter n is called the *granularity* of the reservation manager. At each time step of the simulation, it determines which tiles the vehicle occupies. If throughout this simulation, no required tile is occupied by another vehicle (from a previous reservation), the manager reserves the tiles for this vehicle.

After creating a custom simulator, we evaluated the performance of the reservation system against two other *intersection control policies* - the overpass and the traffic light. An intersection control policy is a method the intersection managers use to determine when specific vehicles are allowed in the intersection. Using the simulator, we showed that using the reservation-based policy, vehicles crossing an intersection experience much lower *delay* (increase in travel time from the optimal) versus the traffic light. Furthermore, we showed that the reservation-based policy drastically increases the throughput of the intersection. For any realistic intersection control policy, there exists an amount of traffic above which vehicles arrive at the intersection more frequently than they can go through the intersection. At this point, the average delay experienced by vehicles travelling through the intersection grows without bound. They demonstrated that compared to the traffic light, this amount of traffic is much higher for the reservation system. In addition to our simulator applets[1] Garcia and Vidal have implemented applets reproducing the results[2].

3. IMPROVING THE ORIGINAL MODEL

The results described in the previous section are very encouraging. In this section, we offer several ways to improve the system with regard to flexibility, efficiency, and making it implementable in the real world.

3.1 Desirable Properties

In order for the reservation-based mechanism to be both realistic and practical, we believe that the following properties ought to hold.

1. The agents should only communicate information which is necessary for the system to function properly.
2. The agents should only have access to information that can be reliably obtained with current technology.
3. Communication failure (dropped messages) should not violate the system's safety properties.
4. The vehicles should be treated as individual agents, and no centralized controller should have any more control over them than necessary.
5. The system should incorporate a simple communication protocol that allows agents to know only a minimal amount about each other. As long as agents obey and understand the protocol, no extra information exchange or other interaction should be required.
6. Every vehicle should eventually make it through the intersection (i.e. no deadlocks or starvation).

3.2 Acceleration in the Intersection

Our previous implementation of the reservation system made reservations for vehicles only at a constant velocity. This property is partly responsible (along with others discussed in Section 6) for the deadlocks their system experienced. With this restriction, if a vehicle made a reservation at a low velocity, it would consume a large amount of space-time in the intersection. This, in turn, would cause other vehicles to be delayed making their reservations (which would also be at low velocities). These slow-downs often led to permanent deadlocks. By allowing acceleration in the intersection, our system always eventually recovers from slowdowns caused by heavy traffic.

Because the reservation manager can now return reservations with accelerations, the problem becomes determining what those accelerations should be. By varying its accelerations just right, a vehicle may be able to fit through a small opening in the intersection. Somehow, the intersection manager must choose the correct accelerations. We chose to use a very simple heuristic: the intersection manager first tries to have the entering vehicle accelerate to the maximum allowed velocity. If such a reservation is not possible, it attempts to make a constant-velocity reservation. If the constant-velocity reservation also fails, the request is rejected. Using acceleration in the intersection, along with the protocol presented in Section 4, allows us to implement the stop sign policy within this reservation framework.

3.3 Excess Information

Our previous work relied on the assumption that vehicles knew each others' positions and reservation statuses at all times. However, it is not immediately obvious how any vehicle would get this information in the real world. While exact position information would be hard to come by, there is no reason to believe that vehicles would have any access at all to the internal state of other vehicles around it (even ones in close proximity). An older model vehicle interacting with a new model vehicle can not be expected to understand the newer model's inner workings. Additionally, the manufacturer of the driver agent may not want other vehicles to know what goes on "under the hood."

3.4 Unspecified Communication Between Driver Agents and Intersection Managers

Our previous paper [3] specified which agents govern which aspects of their system, but they do not specify exactly *how* the agents

[1] `http://www.cs.utexas.edu/users/kdresner/2004aamas`

[2] `http://jmvidal.cse.sc.edu/netlogomas/TrafficManagementMendoza.html`

coordinate their efforts. Additionally, in their work, any driver agent would have to understand what kind of intersection control policy the intersection manager was using in order to interact with it. To address these issues, we created a detailed communication protocol to govern and restrict the interactions of driver agents and intersection managers.

This protocol solved three problems at once. First, all information between the agents goes through one monitorable channel, which makes it much easier to reason about. Second, by limiting the interactions of the agents to a few message types, we can ensure that no agent has an unrealistic amount of control over another. Third, the agents now have a way to communicate that is identical for any intersection management policy or driver agent policy. A vehicle can cross an intersection using a traffic light without knowing it is a traffic light. The traffic light speaks the same language as a stop sign and a reservation system. The driver agent thus must have a behavior that works with all sorts of intersection control policies — that is, the driver agent must view the intersection as a black box, and vice versa.

4. PROTOCOL

We have created a protocol by which the agents can communicate the bare minimum of information necessary to function appropriately. The protocol consists of several message types for each kind of agent, as well as some rules governing when the messages should be sent and what sorts of guarantees accompany them. A detailed specification of the protocol including full syntax and semantics is available in our technical report [2]. In this section we present those aspects that are essential to understanding the remainder of the paper.

4.1 Message Types

The vehicles and intersection manager are each restricted to a few types of messages with which they must coordinate.

4.1.1 Vehicle → Intersection

There are four types of messages that can be sent from vehicles to the intersection.

1. REQUEST — This is the message a vehicle sends when it does not have a reservation and wishes to make one. It contains the properties of the vehicle (ID number, performance, size, etc.) as well as some properties of the proposed reservation (arrival time, arrival velocity, type of turn, arrival lane, etc.).

2. CHANGE-REQUEST — This is the message a vehicle sends when it has a reservation, but would like to switch to a different set of parameters.

3. CANCEL — This is the message a vehicle sends when it no longer desires its current reservation.

4. RESERVATION-COMPLETED — This message is used when the vehicle has completed its traversal of the intersection. This message can be used to collect statistics for each vehicle, which can be recorded in order to analyze and improve the performace of the intersection manager.

4.1.2 Intersection → Vehicle

There are three types of messages that can be sent from the intersection to the individual vehicles.

1. CONFIRMATION — This message is a response to a vehicle's REQUEST (or CHANGE-REQUEST) message. It can contain a counter-offer by the intersection. The reservation parameters in this message are implicitly accepted by the vehicle, and must be explicitly cancelled if the driver agent of the vehicle does not approve. Note that this is safe to faulty communication — the worst that can happen is that the intersection reserves space that does not get used.

2. REJECTION — By sending this message, an intersection can inform a vehicle that the parameters sent in the latest REQUEST (or CHANGE-REQUEST) were not acceptable, and that the intersection either could not or did not want to make a counter-offer. This message also contains a field indicating whether or not the rejection was because the reservation manager requires the vehicle to stop at the intersection before entering. This lets the driver agent know that it should not attempt any more reservations until it reaches the intersection.

3. ACKNOWLEDGMENT — This message acknowledges the receipt of a CANCEL or RESERVATION-COMPLETED message.

4.2 Protocol Actions

In addition to message types, the agents involved (the vehicles and the intersection) must obey a set of rules. These are not entirely unlike the rules that human drivers follow when driving.

4.2.1 Vehicle Actions

These are the rules that the vehicles are expected to follow in order to allow the intersection to function efficiently.

1. A vehicle may not enter the intersection without a reservation.

2. If a vehicle is going to cross the intersection, it must do everything reasonable within its power to cross in accordance with the parameters included in the most recent CONFIRMATION message it has received from the intersection.

3. If a vehicle sends another message before the intersection manager has sent a response, the intersection manager may choose to ignore it. Thus, a vehicle should only send a message if it has received a response to its previous message.

4. If a vehicle has not yet entered the intersection and does not have a reservation, it may send a REQUEST message. If it has not yet entered the intersection and does have a reservation, it may send either a CHANGE-REQUEST or CANCEL message. If it sends any of these messages when it is not allowed to, the intersection may choose to ignore them.

5. If a vehicle has a reservation and has successfully crossed the intersection, it may send a RESERVATION-COMPLETED message.

6. If a vehicle receives a CONFIRMATION message, it is considered to have a reservation.

4.2.2 Intersection Actions

These are the rules representing the obligations the intersection manager is expected to fulfill.

1. When an intersection receives a REQUEST message, it must respond with either a CONFIRMATION or a REJECTION message. If it responds with a CONFIRMATION message, it is

guaranteeing that no cross-traffic will interfere with the vehicle if it crosses the intersection in accordance with the parameters in the message.

2. When an intersection receives a CHANGE-REQUEST message, it must respond with either a CONFIRMATION or a REJECTION message. If it responds with a CONFIRMATION message, it is guaranteeing that no cross-traffic will interfere with the vehicle if it crosses the intersection in accordance with the parameters in the message. Any previous guarantees are nullified.

3. When an intersection receives a CANCEL message, it must respond with an ACKNOWLEDGMENT message. Any guarantee that had been made to the sending vehicle is nullified.

5. INTERSECTION CONTROL POLICIES

Using this protocol, we can express the control policies from our prior work as well as a new one, the stop sign.

5.1 Overpass

The overpass accepts all REQUEST and CHANGE-REQUEST messages exactly as they are, sending corresponding CONFIRMATION messages (with reasonably large error values). This is good for testing purposes, but implementing the overpass with this protocol is only an academic exercise - there would be no reason for it in a real system (in fact it would be quite dangerous).

5.2 Reservation System

message, the intersection simulates the journey of the vehicle with the supplied parameters. If the vehicle can make it through the intersection without using space-time reserved by another vehicle (or near another vehicle), the intersection generates a unique reservation ID, records the reservation, and sends a CONFIRMATION message to the vehicle. If the vehicle cannot make it, the intersection responds with a REJECTION message.

On receiving a CHANGE-REQUEST, the intersection again simulates the journey of the vehicle with the revised parameters. If the vehicle can make it through, the intersection removes the old reservation, generates a new ID, records the new reservation, and sends a CONFIRMATION message to the vehicle. If the vehicle cannot make it, the intersection responds with a REJECTION message (and the vehicle keeps its old reservation).

On receiving a CANCEL or RESERVATION-COMPLETED message, the reservation system deletes the reservation associated with the reservation ID in the message, and responds with an ACKNOWLEDGMENT message.

5.3 Stop Sign

The stop sign is exactly like the a reservation system, except that it only accepts reservations from vehicles that are stopped at the intersection. Any other reservation requests are rejected with a message indicating the vehicle must stop at the intersection.

5.4 Traffic Light

When the traffic light receives a REQUEST message, it examines the arrival time in the message. It then calculates the next time after this that the light for the direction, turn, and lane of the sending vehicle will be green and responds with a CONFIRMATION message that reflects this information (including errors that correspond to the beginning and end of the green light).

6. NEW DRIVER AGENT

The above protocol is designed to place minimal restrictions on vehicle control. As a result, there remains a lot of freedom in creating driver agents. Though our system does not depend on any specific driver agent implementation, we need at least one concrete instantiation in order to test it empirically. In this section we discuss our extensions to our driver agent [3].

Previously, once a driver agent made a successful reservation (at its current velocity), it was forced to maintain that velocity until it reached the intersection. This is a major weakness for the system. If vehicles ever made reservations at very low velocities, not only did they consume a lot of valuable space-time in the intersection, but they also slowed down traffic behind them the rest of the way to the intersection. Repeated iterations of this scenario eventually contribute to deadlocking the system. In fact, the authors point out that their system did deadlock under certain circumstances for this very reason. The other part of this problem (that vehicles cannot accelerate while in the intersection) is addressed via the protocol presented in Section 4.

6.1 Optimism and Pessimism

Unlike our previous implementation of the driver agent, our new agent does not calculate its reservation times using only its current velocity. In the prior work, the driver agent always made requests by calculating the time to get to the intersection at its current velocity, after which, it maintained that velocity until it was through the intersection. It does not matter *how* the vehicle reaches the intersection, as long as the vehicle arrives as scheduled. The behavior as originally proposed can lead to serious problems when, for example, a vehicle makes a reservation while stuck behind a slower-moving vehicle. If the vehicle in front eventually accelerates, the other vehicle should be able to accelerate as well (possibly switching to an earlier reservation).

To utilize this flexibility, we introduce the notion of an *optimistic* or *pessimistic* driver agent. An optimistic agent makes a reservation assuming it will immediately get to accelerate to full speed. An agent which no longer finds itself stuck behind a slower vehicle will become optimistic and attempt to make a new, earlier reservation. A pessimistic agent assumes it will be stuck at its current velocity until it reaches the intersection. If an agent has to cancel its reservation because there is no way for it to arrive on time, it becomes pessimistic. Due to the relatively infrequent and smooth transitions through these situations, our driver agent can take advantage of improving circumstances without causing it to send excessive numbers of CHANGE-REQUEST messages when things change.

6.2 Cancellation and Communication Complexity

Another change, very closely related to the previous section, is an improvement in the communication complexity of the model. In the initial model, the agent determined whether or not it could honor a reservation assuming it kept its present velocity for the remainder of the journey to the intersection. While this might keep things more up-to-date, it often caused a decelerating agent to make and cancel new reservations in rapid succession until it stopped decelerating. In order to prevent this, the new agent only cancels a reservation if there is absolutely no physical way it could reach the intersection on time. If a person were a few minutes late in leaving for the airport, that person would not immediately cancel his or her flight entirely. On the contrary, that person would hope to make up lost time at some point before the flight left. Only when there was no hope of making it to the jetway on time would the person actually cancel the reservation.

Reducing the communication complexity of the system is very important for two reasons. First, if fewer total messages are sent, the bandwidth required to send messages is lower; thus, given the available bandwidth, messages are much less likely to be delayed or lost — events which might negatively affect the system's efficiency. Second, many of the messages (like the REQUEST and CHANGE-REQUEST messages) directly result in intense computation by the intersection manager. Because the resources of the intersection manager are limited, it can only process these messages at some fixed rate. In order to regulate the driver agents, we envision that some sort of charge (perhaps a micropayment) will be levied for each message. In this case, reducing the number of messages sent will be a priority for driver agents.

7. EMPIRICAL RESULTS

In this section, we evaluate the performance of our improved reservation system for varying amounts of traffic and varying percentages of turning vehicles. Additionally, we show results for the new stop sign control policy as implemented under our protocol. We then compare these to results from an earlier paper regarding standard traffic lights. Finally, we experiment with allowing vehicles to turn from any lane — something that would be extremely dangerous without the reservation-based mechanism.

For each experiment, the simulator simulates 3 lanes in each of the 4 cardinal directions. The total area modelled is a square with sides of 250 meters. The speed limit in all lanes is 25 meters per second. Figure 1 shows a screenshot of the graphical display. Each time step in the simulator represents .02 seconds of real time. During each time step, a vehicle is spawned with the given probability, each driver is given sensor input and a decision-making phase, the positions of each vehicle are updated based on the decisions of the driver, and finally any vehicles that have left the area of the simulation are removed. Every configuration shown is run for 100,000 steps in the simulator, which corresponds to approximately half an hour. Vehicles that are spawned in any given direction turn both right and left with probability .05. Unless otherwise specified, vehicles turning right are spawned in the right lane, whereas vehicles turning left are spawned in the left lane. Vehicles that are not turning are distributed probabilistically amongst the lanes such that the traffic in each lane is as equal as possible. The reservation system in these simulations has a granularity of 24 and ensures that no two vehicles occupy the same tile within half a second of each other. Videos of the simulator running can be seen at `http://www.cs.utexas.edu/users/kdresner/2005aamas/`.

Once turns are allowed, delay does not work very well as a metric. There are many different paths through the intersection and amongst them are several different total distances. In addition, vehicles that are turning must slow down before making their turns, so they may take longer than the minimum time to go through the intersection, even under optimal conditions. Because of this, we have decided to simply measure the average time it takes a vehicle to go from a fixed start point to a fixed destination point. We refer to this time as the *trip time*.

Note that in the previous work, the traffic light was shown to have trip times of at least 5 seconds longer than optimal, even in scenarios with extremely light traffic. The absolute shortest time to go from start to finish in this scenario is 10 seconds, which means that the average trip time for the traffic light would be at least 15 seconds.

7.1 The Overpass

In our last paper [3], we presented the overpass as the optimal solution to the intersection control problem. With the addition of turns, a traditional overpass does not make sense. However, we would like an ideal-case solution in which cross-traffic does not affect the time it takes a vehicle to complete its journey. Thus, although it does not represent a true overpass, we still refer to this solution as "the overpass." Vehicles are granted reservations at any time and they can pass through one another, however vehicles turning may have to slow down in order to make the turn.

Although a lower bound on the trip time of a vehicle is 10 seconds, turning vehicles must slow to make the turn. Thus the average time for the overpass system as shown in Figure 2 is just above 10 seconds.

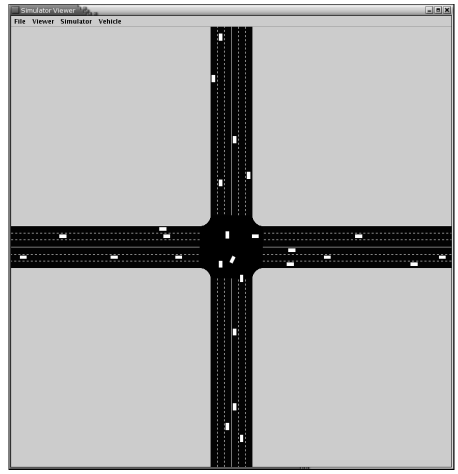

Figure 1: A screenshot of our simulator in action.

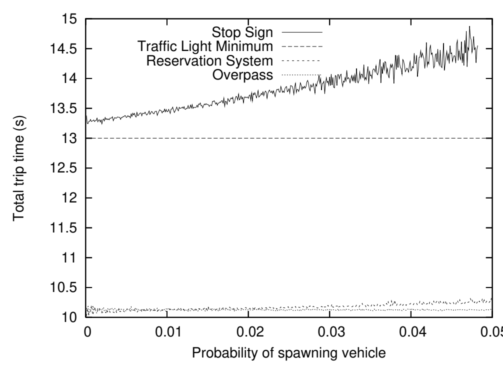

Figure 2: Trip times for varying amounts of traffic for the reservation system, the stop sign, and the optimal "overpass".

7.2 The Reservation System

The reservation system performs very well, nearly matching the performance of the overpass system. At higher levels of traffic, the average trip time for a vehicle gets as high as 10.35 seconds, but is never more than 1 second above optimal. Under none of the tested conditions does the reservation system approach the trip times of the traffic light system in our previous work.

7.3 The Stop Sign

Small intersections with slow-moving traffic tend not to be amenable to control by traffic lights. Light traffic can usually regulate itself fairly effectively. For example, consider an intersection with a stop sign - all vehicles must come to a stop, but afterwards may proceed if the intersection is clear. In these situations, a stop sign is often much more efficient than a traffic light, because vehicles are never stuck waiting for a light to change when there is no cross-traffic. Because our new protocol enables us to define such a control policy, we test how it compares to the other systems as well. Note that this system is much more efficient than an actual stop sign, because once the vehicle has stopped at the intersection, the driver agent and intersection can determine when the car may safely proceed more precisely than a human driver. As shown in Figure 2, the stop sign does not perform as well as the reservation system or the overpass, but for low amounts of traffic, it still performs fairly well, with average trip times only about 3 seconds greater than optimal. As the traffic level increases, however, performance degrades.

7.4 Allowing Turns from Any Lane

In traditional traffic systems, especially those with traffic lights, vehicles wishing to turn onto the cross street must do so from specially designated turning lanes. This helps prevent cars that want to turn from holding up non-turning traffic. However, with a system like the reservation system, this restriction is no longer necessary. There is nothing inherent in the reservation system that demands vehicles turn from any specific lane, and thus we investigated these effects[3]. As seen in Figure 3, relaxing this restriction in fact worsens performance. While one might think this allows the vehicles more flexibility, it on average increases the resources used by any one turning vehicle. By making left turns from the left lane and right turns from the right lane, vehicles both travel a shorter distance and use reservation tiles that are less heavily used.

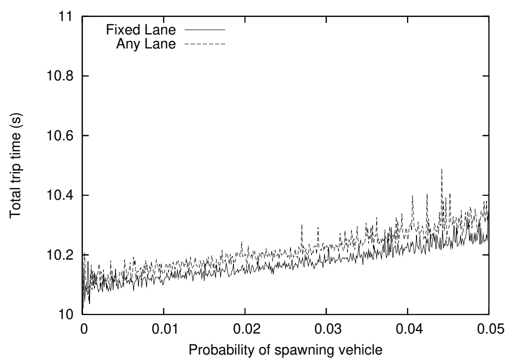

Figure 3: Comparison of the normal reservation system with turns to one allowing turning from any lane.

7.5 Changes to the Driver Agent

As shown in Figure 4, the improvements to the driver agent drastically reduced both the average number of reservations made as well as the average number of messages transmitted. These data were collected using the same simulator settings as the rest of this section, but with a vehicle spawning probability of .02 (approximately 2000 vehicles). For lower amounts of traffic, the effect was less pronounced.

8. DISCUSSION AND RELATED WORK

	Messages	Reservations
Before	560.85	165.89
After	5.97	1.02

Figure 4: For a moderate amount of traffic, the average number of messages sent and reservations made by driver agents before and after the improvements described in Section 6.

We have shown that our reservation system can be extended naturally to incorporate turning and accelerating in the intersection. Furthermore, we have shown that the reservation system can outperform the stop sign, approaching optimal, at a wide range of traffic densities. Our communication protocol, which allows the system to subsume both the stop sign and the traffic light, solves some major concerns posed as detailed in our previous work [3].

One of these concerns was allowing the system to work with human drivers, pedestrians, or cyclists. One can imagine a system that shifts to a traffic-light-like control policy (with physical lights) when it detects vehicles or pedestrians that cannot participate in the reservation system. These individuals could then interact with the intersection the way they do currently. Once the traffic consisted only of participating vehicles, the intersection manager could switch back to a more efficient reservation-based policy.

8.1 Future Work

There are still many challenges and interesting questions to be answered in this domain. For example, we investigated the effects of allowing the vehicle to turn from any lane, but we did not investigate what happens when vehicles are allowed to turn *into* any lane. Furthermore, with the creation of a communication protocol, we can create more interesting driver agents and intersection managers. Both could involve machine learning. The inherent multi-agent nature of the domain makes it a good testbed for multi-agent learning research. The agents can be heterogenous, and the different types of agents (intersection managers and drivers) have different, but not necessarily opposing, goals.

We also see a large opportunity for more research in designing more intelligent reservation systems and driver agents. Currently both of these use heuristics to find available reservations and reservation times, respectively. Applying machine learning techniques to these issues could increase the efficiency of the system even further.

8.2 Related Work

Rasche and Naumann have worked extensively on decentralized solutions to intersection collision avoidance problems [8, 10]. Many approaches focus on improving current technology (systems of traffic lights). For example, Roozemond allows intersections to act autonomously, sharing the data they gather [14]. The intersections then use this information to make both short- and long-term predictions about the traffic and adjust accordingly. This approach still assumes human-controlled vehicles. Bazzan has used an approach using both MAS and evolutionary game theory which involves multiple intersection managers (agents) that must focus not only on local goals, but also on global goals [1].

Work is also being done with regard to the control of the individual vehicles. Hallé and Chaib-draa have taken a MAS approach to collaborative driving by allowing vehicles to form *platoons*, groups of varying degrees of autonomy, that then coordinate using a hierarchical driving agent architecture [4]. While not focusing on intersections, Moriarty and Langley have shown that reinforcement

[3]Videos of this can be seen at `http://www.cs.utexas.edu/users/kdresner/2005aamas/`.

learning can train efficient driver agents for lane, speed, and route selection during freeway driving [7].

On real autonomous vehicles, Kolodko and Vlacic have created a primitive system for intersection control which is very similar to the granularity-1 reservation system [6].

Actual systems in practice (not MAS) for traffic light optimization include TRANSYT [12], which is an off-line system requiring extensive data gathering and analysis, and SCOOT [5], which is an advancement over TRANSYT, responding to changes in traffic loads on-line. However, almost all of the methods in practice or discussed above still rely on traditional signalling systems.

9. CONCLUSION

This paper makes four main contributions. First, it augments a proposed intersection control mechanism to allow for more flexible vehicle control, including turning and accelerating while in the intersection. Second, it introduces a detailed protocol by which vehicles and intersection managers can effectively and efficiently communicate and coordinate their actions. Third, it describes a driver agent that makes good use of this protocol. Finally, it demonstrates how this augmented system, using the protocol, can still drastically outperform both the traffic light and the stop sign.

The mechanism is currently limited by the use of straightforward heuristics to calculate reservation parameters, both on the part of the intersection manager and the driver agents. However, this limitation is a focus of our ongoing research. Once autonomous vehicles become common, this mechanism may be useful for controlling real traffic.

Acknowledgments

This research is supported in part by NSF CAREER award IIS-0237699.

10. REFERENCES

[1] 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.

[2] K. Dresner and P. Stone. Multiagent traffic management: A protocol for defining intersection control policies. Technical Report UT-AI-TR-04-315, The University of Texas at Austin, Department of Computer Sciences, AI Laboratory, December 2004.

[3] K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *The Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 530–537, New York, New York, USA, July 2004.

[4] S. Hallé and B. Chaib-draa. A collaborative driving system based on multiagent modelling and simulations. *Journal of Transportation Research Part C (TRC-C): Emergent Technologies*, 13:320–345, 2005.

[5] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton. SCOOT - a traffic responsive method of co-ordinating signals. Technical Report TRRL-LR-1014, Transport and Road Research Laboratory, 1981.

[6] J. Kolodko and L. Vlacic. Cooperative autonomous driving at the intelligent control systems laboratory. *IEEE Intelligent Systems*, 18(4):8–11, July/August 2003.

[7] D. Moriarty and P. Langley. Learning cooperative lane selection strategies for highways. In *Proceedings of the Fifeenth National Conference on Artificial Intelligence*, pages 684–691, Madison, WI, 1998. AAAI Press.

[8] R. Naumann and R. Rasche. Intersection collision avoidance by means of decentralized security and communication management of autonomous vehicles. In *Proceedings of the 30th ISATA - ATT/IST Conference*, 1997.

[9] D. A. Pomerleau. *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Publishers, 1993.

[10] R. Rasche, R. Naumann, J. Tacken, and C. Tahedl. Validation and simulation of decentralized intersection collision avoidance algorithm. In *Proceedings of IEEE Conference on Intelligent Transportation Systems (ITSC 97)*, 1997.

[11] C. W. Reynolds. Steering behaviors for autonomous characters. In *Proceedings of the Game Developers Conference*, pages 763–782, 1999.

[12] D. I. Robertson. TRANSYT — a traffic network study tool. Technical Report TRRL-LR-253, Transport and Road Research Laboratory, 1969.

[13] S. Rogers, C.-N. Flechter, and P. Langley. An adaptive interactive agent for route advice. In O. Etzioni, J. P. Müller, and J. M. Bradshaw, editors, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 198–205, Seattle, WA, USA, 1999. ACM Press.

[14] D. A. Roozemond. Using intelligent agents for urban traffic control control systems. In *Proceedings of the International Conference on Artificial Intelligence in Transportation Systems and Science*, pages 69–79, 1999.

[15] T. Schonberg, M. Ojala, J. Suomela, A. Torpo, and A. Halme. Positioning an autonomous off-road vehicle by using fused DGPS and inertial navigation. In *2nd IFAC Conference on Intelligent Autonomous Vehicles*, pages 226–231, 1995.

[16] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, July 2000.

[17] Texas Transportation Institute. 2004 urban mobility report, September 2004. Accessed at `http://mobility.tamu.edu/ums` in December 2004.