In *The Fifth Workshop on Agents in Traffic and Transportation Multiagent Systems* (ATT 08), pp. 94-101, Estoril, Portugal, May 2008.

Replacing the Stop Sign: Unmanaged Intersection Control for Autonomous Vehicles

Mark VanMiddlesworth
Harvard University
Elec. Eng. and Comp. Sci. Department
mvandmidd@fas.harvard.edu

Kurt Dresner
University of Texas at Austin
Department of Computer Sciences
kdresner@cs.utexas.edu

Peter Stone
University of Texas at Austin
Department of Computer Sciences
pstone@cs.utexas.edu

ABSTRACT

As computers inevitably begin to replace humans as the drivers of automobiles, our current human-centric traffic management mechanisms will give way to hyper-efficient systems and protocols specifically designed to exploit the capabilities of fully autonomous vehicles. We have introduced such a system for coordinating large numbers of autonomous vehicles at intersections [4, 5]. Our experiments suggest that this system could alleviate many of the dangers and delays associated with intersections by allowing vehicles to "call ahead" to an agent stationed at the intersection and reserve time and space for their traversal. Unfortunately, such a system is not cost-effective at small intersections, as it requires the installation of specialized infrastructure. In this paper, we propose an intersection control mechanism for autonomous vehicles designed specifically for low-traffic intersections where the previous system would not be practical, just as inexpensive stop signs are used at intersections that do not warrant a full traffic light installation. Our mechanism is based on purely peer-to-peer communication and thus requires no infrastructure at the intersection. We present experimental results demonstrating that our system, while not suited to large, busy intersections, can significantly outperform traditional stop signs at small intersections: vehicles spend less time waiting and consume less fuel.

1. INTRODUCTION

Recent advances in technology have made it possible to construct a fully autonomous, computer-controlled vehicle capable of navigating a closed obstacle course. The DARPA Urban Challenge [1], at the forefront of this research, aims to create a full-sized driverless car capable of navigating alongside human drivers in heavy urban traffic. It is feasible that, in the near future, many vehicles will be controlled without direct human involvement. Our current traffic control mechanisms, designed for human drivers, will be upgraded to more efficient mechanisms, taking advantage of cutting-edge research in the field of Multiagent Systems (MAS). Previously, we introduced an MAS-based traffic management system that has the potential to vastly outperform current traffic signals [4, 5]. In this system, vehicles negotiate with an agent stationed at the intersection, which grants each vehicle a specific time and space for its traversal. However, the high infrastructure costs associated with this system make it uneconomical at low-traffic intersections. For these situations, we propose a new control mechanism, based on peer-to-peer interaction, that requires no specialized infrastructure at the intersection.

1.1 A Managed Intersection Control Mechanism

Previously, we proposed an intersection control mechanism to direct autonomous agents safely through an intersection [5]. This system is based on the interaction of two classes of agents: *intersection managers* and *driver agents*. Driver agents "call ahead" to an intersection manager at the intersection, reserving the time and space needed to cross. Specifically, when approaching an intersection, a driver agent sends a request message containing a predicted arrival time and velocity, along with basic information about the vehicle it is controlling. The intersection manager responds with either a confirmation message containing details of the approved reservation, or a denial message, signaling that the parameters sent by the driver agent are unacceptable. In the case of confirmation, the driver agent will attempt to meet the parameters of the reservation, and will cancel the reservation if it cannot. In the case of denial, the driver agent must try to make a different reservation.

Intersection managers base their decisions on the supplied parameters and an *intersection control policy*. The most efficient policies, including FCFS or "first come, first served", simulate the trajectory of the vehicle through the intersection. At each stage in the simulation, the intersection manager checks whether the vehicle is within a certain buffer distance of any other vehicle in the intersection. If the requesting vehicle can cross the intersection without entering any space-time reserved by another vehicle, the policy creates the reservation, and the intersection manager approves the request. Otherwise, the policy does not create a reservation, and the intersection manager denies the request. By integrating these policies with traditional traffic light systems, we have also demonstrated that the system can accommodate human traffic [6]. This multiagent approach offers substantial safety and efficiency benefits as compared to existing mechanisms, such as traffic lights and stop signs. Vehicles pass through the intersection faster, and congestion at intersections is significantly reduced.

Although at the city level this system is mostly decentralized, at each individual intersection, traffic is coordinated by a single arbiter agent, the intersection manager. We therefore designate this system a *managed* intersection control mechanism. An intersection controlled by a traffic light is also a managed intersection—the traffic light being the arbiter agent. Conversely, we designate intersection control mechanisms without an arbiter agent, such as stop signs and traffic circles, *unmanaged* intersection control mechanisms.

1.2 One Size Does Not Fit All

Managed intersection control mechanisms have a major drawback: cost. An arbiter agent of some sort must be stationed at the intersection, and our previously proposed managed system, this agent must have sufficient computational resources and communications bandwidth to rapidly negotiate a high volume of requests. Although the throughput benefits in large intersections would certainly warrant this expense, the system would be uneconomical for small intersections. Stop signs are a low-overhead, unmanaged system designed for low-traffic intersections, complementing larger intersections managed by traffic lights. In this paper, we propose an unmanaged intersection control mechanism for autonomous vehicles, designed specifically for low-traffic intersections. Our system—based on peer-to-peer communication and requiring no specialized infrastructure—is a similar complement to the managed intersection we previously proposed [5]. We make similar assumptions about the driver agent, such that a driver agent capable of using the managed system can be modified to use both systems seamlessly. We also present empirical data comparing our system to both traffic lights and stop signs. We focus our analysis primarily on the comparison between our system and the class of intersections that would currently be managed by a stop sign (low-traffic intersections), as these are the intersections for which our system is intended.

The remainder of this paper is organized as follows. In Section 2 we introduce the goals of our system, state our assumptions about the agents' world knowledge, and outline the protocol of our system. Section 3 describes the behavior of each individual driver agent. In Section 4, we present and discuss the empirical results of our system. Section 5, contains a discussion of current related work and presents some directions for further research. We summarize and conclude in Section 6.

2. AN UNMANAGED AUTONOMOUS INTERSECTION

To address the issue of high cost associated with managed autonomous intersections, we have created a low-cost alternative for low-traffic intersections. In this section, we introduce our unmanaged autonomous intersection control mechanism. First, we specify the goals of our system. Next, we describe our assumptions about the driver agents. We then outline the protocol for communication between vehicles, and describe the rules that each vehicle must follow.

2.1 Goals Of The System

For an unmanaged intersection control mechanism for autonomous vehicles to be both effective and economically viable, we believe it should have the following properties:

- Vehicles using the system should get through the intersection more quickly than they do using current mechanisms (i.e. stop signs).
- The protocol should have minimal (ideally none) per-intersection infrastructure costs.
- The protocol should guarantee the safety of the vehicles using it. Specifically, if all vehicles follow the protocol correctly, no collisions should result.

2.2 Assumptions

To safely navigate an intersection, a driver agent needs access to specific information: the layout and location of the intersection, any speed limits, and a variety of other parameters. As with our managed system, we assume that vehicles have access to this information either on board the vehicle or via a remote database. We assume that each vehicle is outfitted with a wireless communication device with sufficient range to communicate with other vehicles approaching the intersection. This range is approximately 200 meters in our scenario, but could vary based on the size of the intersection. We assume that this communication device has sufficient bandwidth to handle vehicle-to-vehicle communication, although our implementation relies on very small data packets, and we do not expect bandwidth to be a serious constraint. Finally, we assume that the latency of this device is sufficiently low. In our testing, we simulate a 20ms latency, but this is not a strict requirement of our system, as the parameters of the protocol can be adjusted to suit the environment (see Section 2.3).

In addition to these intersection-specific assumptions, we also assume that each vehicle has all the abilities required of autonomous open-road driving. These include access to a GPS-like navigation system that can provide an accurate and precise position, as well as laser range finders or short-wave radar capable of reliably sensing other vehicles in the immediate vicinity.

Finally, we assume that driver agents have access to information about the vehicle they are controlling, including its current velocity, position, and heading.

By analyzing the physical layout of the intersection, agents can determine which of the paths through it are *compatible*. That is, which paths can safely be followed simultaneously without the risk of a collision. For example, right turns from the rightmost lanes in any direction are always compatible, whereas any paths that intersect are not. Rather than having each agent independently find these paths, we assume that the list of compatible trajectories is part of the agent's knowledge of the intersection. Because driver agents may use this information to plan their trajectory through the intersection, possibly allowing two vehicles to cross simultaneously, it is important that each agent have the same notion of which paths are compatible.

2.3 Communication Protocol

Unlike the protocol for our managed intersection [3], our protocol for unmanaged autonomous intersection control is designed for communication among only one type of agent: driver agents. In our system, each agent sends and receives information to and from each other agent, maintaining up-to-date information about every vehicle approaching the intersection. Dropped packets and limited transmission distance may cause agents to have outdated or inconsistent information. Because data transmission is largely asynchronous in an ad-hoc wireless network of mobile agents, this protocol cannot rely on a dialogue between agents. As such, the protocol is simple, consisting only of broadcast messages. There are two types of messages: CLAIM and CANCEL.

2.3.1 Claim

A CLAIM message is sent by an agent in order to announce its intentions to use a specific space and time in the intersection. CLAIM contains information describing both the vehicle's intended path through the intersection, as well as when it believes its traversal will take place. Once the agent has chosen these parameters, it broadcasts its CLAIM repeatedly. The message contains seven fields:

- `vehicle_id`—The vehicle's unique Vehicle Identification Number (VIN).
- `message_id`—A monotonically increasing counter specific to this message. Other agents will use `message_id` to identify the most recent message from this vehicle. This number is not changed when a specific message is rebroadcast; it is incremented only when a vehicle generates a *new* message to broadcast.
- `stopped_at_intersection`—A boolean value representing whether the vehicle is stopped at the intersection.
- `lane`—The lane in which the vehicle will be when it arrives at the intersection. Each lane incident to the intersection has an absolute index available as part of the intersection's layout information.
- `turn`—The direction in which this vehicle will turn.
- `arrival_time`—The time at which this vehicle will enter the intersection.
- `exit_time`—The time at which this vehicle will exit the intersection.

2.3.2 CANCEL

An agent sends a CANCEL message to release any currently held reservation. This message cancels any pending reservation; even if other agents have differing or outdated information about an agent's reservation, the agent can still cancel. The CANCEL message is broadcast repeatedly, with the same period as CLAIM, to ensure it is received by all other agents. This message has two fields:

- `vehicle_id`—This vehicle's VIN.
- `message_id`—A monotonically increasing number specific to this message. This is the same as the `message_id` field in CLAIM.

2.3.3 Message Broadcast

Because each message contains all the latest relevant information about the sending vehicle, agents need only pay attention to the most recent message from any other vehicle. Each message is also broadcast repeatedly with a set period to ensure its eventual delivery, should a new vehicle enter the transmission range of the sender. As a result, although occasional dropped messages may increase the delay in communications between vehicles, they should not pose a significant threat to the safety of vehicles in our system. In situations with higher rates of packet loss, messages may need to be broadcast more frequently to compensate. Conversely, in low-latency, high-reliability scenarios, messages can be sent less frequently.

For security purposes, we also assume that each message is digitally signed, ensuring that driver agents cannot falsify the `vehicle_id` parameter. Messages that do not conform to the protocol or are not digitally signed are ignored.

2.3.4 Conflict, Priority, and Dominance

In order to facilitate the discussion of agent behavior and protocol analysis, we define the following relations on CLAIM messages.

Two CLAIM messages are said to *conflict* if all of the following are true:

- The paths determined by the `lane` and `turn` parameters of the CLAIM messages are not compatible
- The time intervals specified in the CLAIM messages are not disjoint

We define the relative *priority* of two CLAIM messages based on the following rules, presented in order from most significant to least significant:

1. If neither CLAIM specifies that the sending vehicle is stopped at the intersection, the CLAIM with the earliest `exit_time` has priority.
2. If both CLAIM messages specify that the respective sending vehicles are stopped at the intersection, the CLAIM whose `lane` is "on the right" has priority. Here, "on the right" is defined similarly to current traffic laws regarding four-way stop signs. This binary relation on the incident lanes is globally available as a characteristic of the intersection.
3. If neither message's `lane` can be established as being "on the right," the CLAIM whose `turn` parameter indicates the sending vehicle is not turning has priority.
4. If priority cannot be established by the previous rules, the CLAIM with the lowest `vehicle_id` has priority.

.

Finally, given two claims c_1 and c_2, we say that c_1 *dominates* c_2 if either of the following rules is true:

- The `stopped_at_intersection` field of c_1 is `true` and the `stopped_at_intersection` field of c_2 is false.
- The `stopped_at_intersection` fields of c_1 and c_2 are identical, c_1 and c_2 conflict, and c_1 has priority over c_2.

2.4 Required Agent Actions

The consequences of failure in a traffic management system can be disastrous. As such, in addition to a communication protocol, a rigid set of rules must govern the interaction of agents within the system. With human drivers, traffic laws serve this purpose: if every driver obeys traffic laws, there is little or no potential for automobile accidents. Our multiagent system relies on an analogous set of rules. While there is nothing physically preventing an agent from ignoring them, the safety of each agent's vehicle can only be guaranteed if that agent follows the rules. Note that the rules restrict only how the agent behaves while in the intersection; driver agents have full autonomy everywhere else. The rules are as follows:

1. A vehicle may not enter the intersection if its own Claim is dominated by any other current Claim.
2. A vehicle may not enter the intersection without first broadcasting an Claim for at least T_p seconds. In our implementation, $T_p = .4$.
3. A vehicle must vacate the intersection at or before the `exit_time` specified in its most recent Claim message.
4. If a vehicle is going to traverse the intersection, it must follow a reasonable path from the point of entry to the point of departure. This means, for example, that a vehicle going straight through the intersection must remain within its lane, and that a vehicle turning right must not enter any other lanes.
5. The `stopped_at_intersection` field of an agent's Claim must be set to `true` if and only if the agent is stopped at the intersection.
6. The agent may not broadcast unless it is within a certain distance of the intersection. This distance is called the *lurk distance*. In our implementation, the lurk distance is 75 meters.

2.5 Selfish and Malicious Agents

Agents in our system are assumed to be self-interested—they may take any possible legal action in order to ensure they traverse the intersection in as little time possible. Agents have little incentive to lie about their lane, path, or exit time, because lying about any of these puts the vehicle at risk for collision. However, an agent may have an incentive to falsely claim that it is stopped at the intersection. While there is a chance this may slow down the traffic in front of the offending vehicle, if there is no such traffic exists, an agent may gain some advantage by falsely claiming that it is stopped at the intersection, allowing its Claim to dominate the Claims of other moving vehicles. This may result in the vehicle crossing the intersection earlier. This type of behavior is not currently disincentivized by our protocol, but if it were to become a problem, could be tested at random intersections to ensure compliance. This is analogous to current traffic enforcement, which relies on sporadic monitoring and associated penalties to decrease rule violations.

As with any multiagent system, malicious agents are a potential problem. In current traffic scenarios, nothing prevents someone from deliberately crashing into another vehicle, or disabling traffic signals. Similarly, a malicious driver agent could flood the network with useless traffic, preventing the system from operating properly. While nothing can be done to stop a determined saboteur, the fact that all messages are signed makes it impossible for vehicles to conceal their identity while using the protocol.

3. DRIVER AGENT BEHAVIOR

Our proposed unmanaged intersection control mechanism relies not only on the communication protocol defined in Section 2.3, but also on the existence of driver agents that can abide by the protocol. Our prototype driver agent's behavior is comprised of three phases: lurking, making a reservation, and intersection traversal.

3.1 Lurking

As the vehicle approaches the intersection, it begins to receive messages from other agents. However, it may not broadcast a reservation until it is within the *lurk distance*. The lurk distance is calculated to ensure that an agent is within transmission range of other vehicles long enough to be reasonably sure that it is aware of every pending Claim. Claims are broadcast repeatedly at a set frequency; more frequent broadcasts reduce the amount of time an agent must spend within transmission range to assemble all pending Claims. Therefore, lurk distance depends on both transmission range and broadcast frequency. In our simulations, we set lurk distance to 75 meters—a reasonable approximation given current communication technology.

3.2 Making a Reservation

The most important part of our driver agent behavior starts when vehicle reaches the lurk distance. At this point, it needs to let the other driver agents know how it intends to cross the intersection. We call this part of the process "making a reservation," as an analogue to our managed system, which also uses a reservation paradigm [5]. During this time, the vehicle needs to compute its expected arrival time, arrival velocity, departure time, and given the messages it has accumulated from other vehicles, determine the soonest time at which the intersection will be available. This behavior is shown in Algorithm 1.

Algorithm 1 Behavior of the driver agent from coming within lurk distance of the intersection to entering the intersection.

1: **loop**
2:   **if** do not have a current Claim **then**
3:     generate a new Claim
4:   **end if**
5:   **if** not at the intersection and another vehicle is **then**
6:     broadcast Cancel
7:   **else**
8:     **if** arriving estimate changes or Claim is dominated **then**
9:       generate a new Claim
10:     **end if**
11:     broadcast the Claim
12:   **end if**
13: **end loop**

As an agent approaches the intersection, it generates a Claim based on predictions of its arrival time, arrival velocity, and path through the intersection (line 3). To predict the time required to cross the intersection, the agent must know its arrival velocity. Initially, the agent calculates the earliest possible arrival time, and the predicted velocity of the vehicle at this time based on the speed limit and its own acceleration constraints (the physical constraints of the vehicle, in addition to the constraints imposed by traffic front of it) Based on this arrival velocity, the agent predicts the time at which it will exit the intersection, assuming that it can accelerate as needed within the intersection. If the agent has received no Claims from other vehicles that dominate this Claim, the agent will begin to broadcast this Claim (line 11).

Otherwise, the agent generates a new Claim at the earliest possible time such that it will not be dominated by any existing Claim of another vehicle (line 9). To do so, the agent searches through existing Claims to find the next block of time that it could potentially dominate, assuming it can arrive at the highest legal velocity. After finding a suitable block, the agent predicts its arrival velocity based on arrival time (which is generally lower than the maximum legal velocity), which it uses to determine the actual time required to cross the intersection. If the agent can traverse the intersection in the available time, it begins broadcasting a CLAIM; if not, it searches for the next suitable block and repeats these calculations.

3.3 Intersection Traversal

Once a vehicle has made a reservation, it needs only to broadcast the CLAIM continually and to arrive at the intersection in accordance with its reservation. However, sometimes the vehicle may want to change an existing claim in order to take advantage of an unexpected early arrival (line 8). On the other hand, traffic patterns may occasionally cause a vehicle to arrive late. If a vehicle predicts that it cannot fulfill the parameters of its CLAIM message, it must either send a CANCEL a new CLAIM. Similarly, if a new CLAIM message arrives that dominates the driver agent's CLAIM, the driver agent must also make a new reservation.

Once the vehicle reaches the intersection, it crosses in accordance with its CLAIM. While in the intersection, for safety purposes, the vehicle continues to broadcast its CLAIM, however this CLAIM cannot be dominated, as the vehicle is already executing the intersection traversal, which is clear from the fact that the current time is after the `arrival_time` in the CLAIM. After a vehicle has vacated the intersection, it stops transmitting its CLAIM.

3.3.1 Vehicle Control

The driving actions taken by a vehicle to complete its reservation are very similar to those of the driver agent in our managed mechanism [3]. If a vehicle predicts that it will arrive late, it accelerates. If a vehicle predicts that it will arrive early, it slows down (unless it believes it can make an earlier CLAIM). The vehicle must also ensure that it arrives with sufficient velocity to traverse the intersection within the constraints of its reservation.

3.3.2 Canceling "Bad" Reservations

In some situations, a vehicle is unable to reach the intersection at the proper time and velocity. To detect these situations, the vehicle is constantly predicting its arrival time. As with the driver agent presented in our work on managed intersections, this agent calculates its arrival time and velocity either *optimistically* or *pessimistically* [5]. If a vehicle detects no vehicles in front of it, it will make an optimistic projection of arrival time, assuming it can accelerate as needed before it arrives. However, if a vehicle is obstructed by traffic, it will make a pessimistic projection of arrival time based on the assumption that it cannot accelerate before it arrives at the intersection. If the vehicle's predicted arrival time is later than that of its reservation, the vehicle will cancel its current reservation and attempt to make a reservation for a later time.

3.3.3 Improving Reservations

If a driver agent predicts that it will arrive at the intersection before the time specified in its reservation, it may be able to improve its reservation before reaching the intersection. To accomplish this, the agent looks for blocks of intersection time between its predicted arrival time and the arrival time specified in its reservation. If the vehicle determines that it can broadcast a suitably large CLAIM that will not be dominated, it will immediately begin broadcasting this CLAIM. As specified by the communication protocol, this implicitly cancels any previous reservation held by the vehicle.

If a vehicle arrives at the intersection before the time specified in its reservation, it changes its CLAIM to reflect that it is stopped and waiting to cross (as required by the protocol). As a result, this agent's CLAIM will now dominate the CLAIM of any vehicle not stopped at the intersection. The stopped agent will then begin broadcasting the earliest possible non-dominated CLAIM. If no other vehicles are stopped, this will be T_p seconds from the current time, as the vehicle must broadcast its claim for at least this amount of time before entering the intersection. If other vehicles are stopped at the intersection, the agent will broadcast a CLAIM for the earliest block of time not dominated by the CLAIM of any stopped vehicles.

4. EMPIRICAL RESULTS

This section presents empirical results comparing our unmanaged autonomous intersection to intersections outfitted with four-way stop signs and traffic lights. After describing our metrics and experimental setup, we compare the average delay induced by each of these control policies. We then use these results estimate the amounts of traffic for which a stop sign outperforms a traffic light. This range is the primary focus of the analysis of our system, as we consider it to be the range over which an unmanaged policy is more appropriate than a managed policy. We also compare the relative fuel consumption associated with the stop sign and unmanaged autonomous policies. Finally, we discuss the effects of dropped messages on our unmanaged autonomous control policy.

4.1 Metrics

In our analysis, we examine two key metrics: *average delay* and *average cumulative acceleration*. The primary metric is the average of the delay experienced by each vehicle as it crosses the intersection. The baseline for delay is the time it would take a vehicle to traverse a completely empty intersection. Because a vehicle must slow down to turn, the baseline is different for left turns, right turns, and straight passages through the intersection. We measured the trip time for an unobstructed vehicle following these three paths, giving us an accurate baseline for comparison. Delay is measured as actual trip time minus baseline trip time, which isolates the effect of the intersection control policies and allows us to accurately compare the among them.

The second metric we use is the average of the the cumulative acceleration of each vehicle during its trip through the intersection. We define the *cumulative acceleration* of a vehicle, denoted $\overline{a}$, as:

$$\overline{a} = \sum_{i=0}^{s} |a_i|$$

where s is the trip length of the vehicle measured in simulator steps, and a_i is the acceleration of the vehicle at simulator step i. Note that the baseline for $\overline{a}$ is nonzero in turning vehicles, as vehicles must slow down to turn and accelerate again to the speed limit afterwards. We chose to compare the average cumulative acceleration to examine the relative fuel efficiency of each system. Although not a direct measure of fuel efficiency, a vehicle's cumulative acceleration provides a reasonable approximation of gasoline usage, because substantially more fuel is required to accelerate than to maintain a constant velocity. Average delay is also an indicator of fuel efficiency, as the delay experienced by a vehicle correlates with the amount of fuel consumed while the vehicle was not accelerating (either idling at the intersection or traveling at a constant velocity). Thus, we can compare the relative fuel efficiency of each system by comparing both average delay and average cumulative acceleration.

4.2 Experimental Setup

To test these policies, we use a custom simulator which simulates a four-way intersection with one lane of traffic in each direction (see Figure 1). This small, symmetrical intersection is representative of those intersections currently configured as a four-way stop, and thus provides the best test case for unmanaged control mechanisms. We control traffic levels via a Poisson process governed by the probability of creating a new vehicle in a given lane at each time step. We simulate traffic levels between 0 and 0.5 vehicles per second, with 15% of vehicles turning left and 15% turning right. Each data point represents the average of 20 simulations, with each run consisting of 30 minutes of simulated time. All data are shown with error bars indicating a 95% confidence interval.

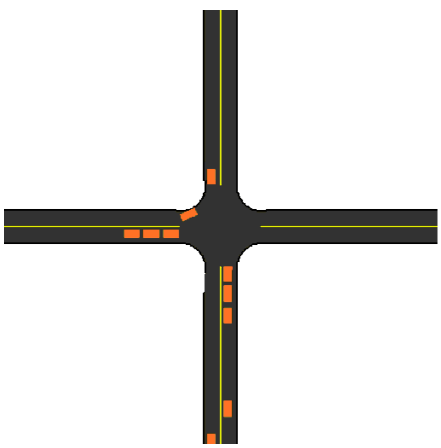

Figure 1: A screenshot of the simulator.

The traffic light timing is configured such that, in succession, each direction receives a green light for 10 seconds, followed by 3 seconds of yellow. There is a large body of theory and empirical evidence concerning the timing of traffic lights, but this work is largely irrelevant to our simulated scenario for two reasons. First, much of the theory deals with the timing of lights across multiple intersections, whereas we are examining one intersection in isolation. Second, our simulator generates symmetric traffic, which greatly simplifies light timing by eliminating the need to account for higher traffic levels in a particular direction or lane. For these reasons, we established a reasonable timing pattern experimentally by evaluating 10 different candidate patterns and selecting the one that led to the lowest average delay.

It should be noted that our four-way stop sign policy does not allow multiple vehicles to inhabit the intersection simultaneously. In the real world, stop signs can allow a limited sharing of the intersection. This is most apparent in intersections with multiple lanes of traffic in each direction: in this situation, cars traveling parallel to one another can cross the intersection at the same time. There is significantly less potential for sharing the intersection when there is only one lane of traffic in each direction. A human driver may observe the vehicle currently crossing the intersection and predict the vehicle's actions for the remainder of its journey (although this prediction is not always accurate!). If the other vehicle's path does not conflict with the intended path of the human driver, he or she may enter the intersection slightly before the other vehicle has exited. However, the benefits of this behavior are significantly reduced in small intersections. Therefore, we believe that our four-way stop sign policy is a reasonable approximation of a real-world four-way stop.

4.3 Delay

As shown in Figure 2, our system significantly reduces the average delay experienced by each vehicle. When traffic flow is below 0.35 vehicles per second, the four-way stop is a more effective policy than the traffic light. Because an unmanaged mechanism performs best over this domain, we consider [0,0.35] vehicles per second to be the target domain of our system.

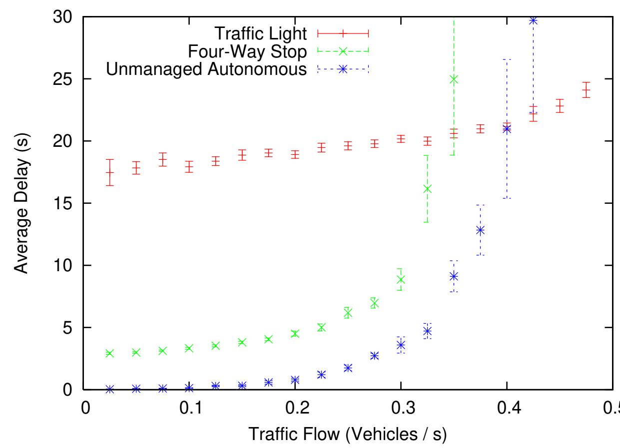

Figure 2: A comparison of average delay of the traffic light, four-way stop, and our unmanaged mechanism. The x-axis represents the traffic level, expressed in vehicles per second. The y-axis represents the average of each vehicle's delay, in seconds.

Our unmanaged system results in near-zero delay at traffic levels below 0.2 vehicles per second. In these situations, most agents are able to cross the intersection without slowing down to wait for other vehicles. With the four-way stop sign, each vehicle must stop even if no others are present, resulting in a baseline average delay of approximately 3 seconds. The traffic light system has a higher baseline average delay, around 18 seconds.

When traffic flow is between 0.2 and 0.35 vehicles per second, our system shows a somewhat increased delay. In these cases, cars must often slow down to accommodate other vehicles, but but only rarely will a vehicle need to make a complete stop. With the stop sign policy, vehicles begin to queue at the intersection, and must often wait for vehicles in front of them to cross. The traffic light policy shows almost no increase in delay at these levels.

At traffic levels above 0.35 vehicles per second, the stop sign policy deadlocks. At these traffic levels, our system is similar to a four-way stop: because there is almost always at least one vehicle waiting to cross, agents must wait until they are stopped at the intersection to make a reservation (as described in Section 2.4). However, the intersection sharing in our system (allowing four simultaneous right turns, for example) provides a noticeable benefit at these traffic levels. Our unmanaged system can safely handle traffic levels up to approximately 0.4 vehicles per second, at which point traffic begins to back up. The traffic light shows only a slight increase in delay at these traffic levels. In these situations, our data suggest that a managed mechanism is more appropriate.

4.4 Average Acceleration

Another benefit of our system is reduced average acceleration, as shown in Figure 3. With the stop sign policy, every vehicle must come to a complete stop at the intersection and accelerate to the speed limit after crossing. If vehicles are queued at the intersection, each vehicle must stop at the back of the queue. As the queue moves forward, each vehicle accelerates for a brief period of time, then decelerates to a stop until another car leaves the front of the queue. This behavior results in a very high average acceleration for the stop sign policy.

For low levels of traffic, our system allows most vehicles to pass directly through the intersection without slowing or stopping. Even at high traffic levels, when our system is essentially a modified four-way stop, our system results in lower average acceleration than a four-way stop. This is because our system causes shorter queues than a stop sign, reducing the amount of acceleration and braking required for each vehicle to reach the front of the queue. Combined with the data on average delay, these results suggest that our unmanaged autonomous system would allow significantly reduced fuel consumption.

4.5 Dropped Messages

We designed our system to be resistant to occasional communication failures such as dropped messages. In our previously proposed managed intersection, the vehicles must wait for a response from the intersection manager before entering the intersection [5]. Because of this, dropped packets may increase the delay of the system, but will not cause a collision. In our system, we have found no statistically significant correlation between dropped packets and delay. Rather, dropped packets introduce a possibility of failure that increases with the percentage of packets dropped.

To quantify this effect, we varied the proportion of dropped messages between 0 and 0.7 at intervals of 0.1, running 400 thirty-minute simulations at each level. The traffic level in these simulations was 0.3 vehicles per second. When fewer than 40% of messages were dropped, the system behaved normally. Between 40% and 60% packet loss, the system began to experience safety failures—five of the 1200 simulations in this range resulted in collisions. At 70% packet loss, the frequency of collisions is significantly higher, with collisions occurring in seven of 200 simulations.

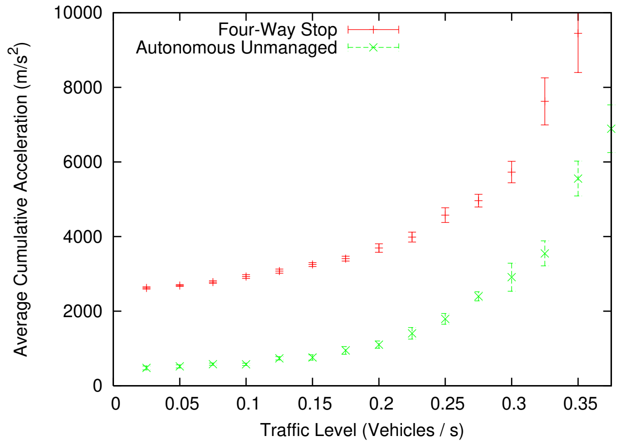

Figure 3: A comparison of average acceleration of the four-way stop and our unmanaged mechanism. The x-axis represents the traffic level, expressed in vehicles per second. The y-axis represents the average of each vehicle's cumulative acceleration, in meters per second per second.

These results suggest that, as proposed, our peer-to-peer protocol can tolerate moderate levels of packet loss with no ill effects, but that serious communication issues might make it unsafe. While a thorough analysis of communication failures is beyond the scope of this paper, research in distributed systems has shown that fast and reliable information dissemination in ad-hoc wireless networks such as the kind we are simulating is possible [2]. We thus leave further communication analysis to future work.

5. DISCUSSION AND RELATED WORK

We have presented a system which allows autonomous agents to coordinate their safe passage through an intersection without an intersection manager, and demonstrated that it outperforms the current mechanisms for both managed and unmanaged intersections over its target traffic levels. We have specified a detailed protocol meeting the constraints of vehicle-to-vehicle communication, which adds few assumptions on top of those in the managed autonomous intersection. Because of this, it would be easy to create a driver agent that can utilize both managed and unmanaged intersections. As this driver agent approaches the intersection, it determines whether the intersection is managed using previous experience or, if the agent has never encountered the intersection, by attempting to communicate with the intersection manager. If the agent receives a response, it uses the appropriate managed intersection protocol; if not, it uses our unmanaged intersection protocol.

5.1 Future Work

After introducing the reservation-based protocol for managed intersections based on the assumption that all cars are autonomous, we later presented a policy which allows both computer- and human-controlled vehicles to safely interact at the same intersection [6]. Our protocol for unmanaged intersections can be similarly adapted to accommodate human drivers using traffic signs. The human drivers would be directed to behave as if they were stopped at a two-way stop, yielding to all approaching vehicles (this also assumes that the computer-controlled vehicles have some signal identifying them as autonomous). Because our system is designed for low-traffic intersections, human drivers could generally expect to wait for no more than a few seconds. Our proposed system for accommodating human drivers and the corresponding managed system both put human-controlled vehicles at somewhat of a disadvantage–an incentive for human drivers to transition to fully computer-controlled vehicles. Future research could formalize and optimize a policy for accommodating human drivers in our unmanaged autonomous intersection.

Another potential area for future research is allowing the system to adapt to asymmetric traffic flow. Many intersections consistently receive higher traffic in some lanes than others. In these intersections, a two-way stop is often more efficient than a four-way stop. In our current system, all agents stopped at the intersection are given equal priority, regardless of the number of vehicles queued behind them. This approximates the behavior at a four-way stop. However, by granting priority to lanes with longer queues, our system could alleviate congestion in high-traffic lanes. This would allow our system to function like a two-way stop in situations with asymmetric traffic flow, while functioning like a four-way stop in situations with more symmetrical traffic.

5.2 Related Work

Intersection management—especially for intersections of autonomous vehicles—is an exciting and promising area of research for autonomous agents and multiagent systems. Many projects in AI and intelligent transportation systems address this increasingly important problem. Using techniques from computer networking, Naumann and Rasche created an algorithm in which drivers attempt to obtain *tokens* for contested parts of the intersection, without which they cannot cross [8]. While this allows vehicles to cross unimpeded in very light traffic, the system has no notion of "planning ahead"; only one vehicle may hold a token at any given time, no agent can plan to have the token in the future if another agent has it currently. Kolodko and Vlacic have created a system very similar to ours on golf cart–like IMARA vehicles [7]. However, their system requires all vehicles to come to a stop, irrespective of traffic conditions.

In the context of video games and animation, Reynolds has developed autonomous steering algorithms that attempt to avoid collisions in intersections that do not have any signaling mechanisms [9]. While such a system does have the enormous advantage of not requiring any special infrastructure or agent at the intersection, it has two fatal drawbacks that make it unsuitable for use with real-life traffic. First, the algorithm does not let driver agents choose which path they will take out of the intersection; a vehicle may even find itself exiting the intersection the same way it came in, due to efforts to avoid colliding with other vehicles. Second, the algorithm only *attempts* to avoid collisions—it does not make any guarantees about safety.

6. CONCLUSION

Recent research has already produced fully autonomous, computer-controlled vehicles. As these vehicles become more common, we will be able to phase out human-centric traffic control mechanisms in favor of vastly more efficient computer-controlled systems. This will be especially beneficial at intersections, which are a major cause of delays. For a transition of this magnitude, infrastructure cost will be a central, if not primary, concern. This paper presents a novel, unmanaged intersection control mechanism requiring no specialized infrastructure at the intersection. We have described in detail a protocol for our unmanaged autonomous intersection, and created a prototype driver agent capable of utilizing this protocol. As illustrated by our empirical results, our protocol can significantly reduce both delay and fuel consumption as compared to a four-way stop. Unsignalized intersections far outnumber those that are sufficiently large or busy to warrant the cost of a managed solution. Whereas busier intersections may need to wait for the funding and installation of requisite infrastructure, our proposed mechanism has the potential to open every one of these unsignalized intersections to be used safely and efficiently by autonomous vehicles.

7. REFERENCES

[1] DARPA. The DARPA urban challenge, 2007. `http://www.darpa.mil/grandchallenge`.

[2] V. Drabkin, R. Friedman, G. Kliot, and M. Segal. Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks. In *The 26th IEEE International Symposium on Reliable Distributed Systems*, Beijing, China, October 2007.

[3] 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.

[4] 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, NY, USA, July 2004.

[5] K. Dresner and P. Stone. Multiagent traffic management: An improved intersection control mechanism. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 471–477, Utrecht, The Netherlands, July 2005.

[6] K. Dresner and P. Stone. Sharing the road: Autonomous vehicles meet human drivers. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1263–68, Hyderabad, India, January 2007.

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

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