Evaluating Policies for Reservation-based Intersection Control[⋆]

Matteo Vasirani and Sascha Ossowski

Centre for Intelligent Information Technology
University Rey Juan Carlos, Madrid, Spain
{matteo.vasirani,sascha.ossowski}@urjc.es

Abstract. Autonomous vehicle navigation in urban road networks will be possible in the future, making traveling yet another activity that does not need the human intervention. In this paper we empirically evaluate different policies to manage a reservation-based intersection, an infrastructure facility proposed by Dresner and Stone to regulate the transit of vehicles through intersections. We evaluate two different scenarios, a single intersection and a network of intersections, comparing the original policy employed by Dresner and Stone in their work with 4 policies, inspired by the adversarial queueing theory (AQT).

Keywords: autonomous vehicle, reservation-based intersection, control policies, adversarial queueing theory

1 Introduction

The recent and promising advances in artificial intelligence and, particularly, in multiagent system technology suggest that autonomous vehicle navigation in urban road networks will be possible in the future. Safe and efficient urban automated guided vehicles (AGV) could make driving yet another activity that does not need the human intervention. At the present time, cars can be equipped with features such as cruise control [3] and autonomous steering [5]. Furthermore, there exist small-scale systems of AGVs, for example in factory transport systems. If this trend holds, one day fully autonomous vehicles will populate our road networks. In this case, given that the system will have a variable (and possibly huge) number of vehicles and an open infrastructure, central control such in today's AGV systems [4] will be impossible.

To this respect, Dresner and Stone [2] introduced a minimally centralized infrastructure facility that allows for the control of intersections. In their model, an intersection is regulated by an intelligent agent (intersection manager) that assigns reservations of space slots inside the intersection to each automated guided vehicle intending to transit through the intersection. Such an approach

[⋆] This research was partially supported by the Spanish Ministry of Science and Education through projects "THOMAS" (TIN2006-14630-C03-02) and "AT" (CONSOLIDER CSD2007-0022, INGENIO 2010).

has shown, in a simulated environment, several advantages, because it may drastically reduce delays with respect to traffic light and it makes possible the use of fine graned, vehicle-centric, control policies.

In the Dresner and Stone's original work, the intersection manager applies a simple first-come-first-served policy, evaluating the reservation requests in the same order they are received by the intersection manager. Nevertheless, allowing the intersection manager to evaluate a set of requests at the same time, it may make more informed decisions so as to optimize the intersection throughput. In this work, we make an empirical evaluation of different policies, inspired by the adversarial queueing theory (AQT) [1], comparing them with the first-come-first-served policy proposed by Dresner and Stone.

This paper is structured as follows: in section 2 we detail the Dresner and Stone's protocol that rules the interaction between the vehicles that want to cross the intersection and the control facility; section 3 introduces the different policies that can be applied for the processing of the reservation requests, evaluating them in section 4; we discuss the experimental results in section 5; finally we conclude in section 6.

2 Protocol

The reservation-based system proposed in [2] assumes the existence of two different kind of agents: *intersection managers* and *driver agents*. The intersection manager controls an intersection and schedules the transit of each vehicle. The driver agent is the entity that autonomously operates the vehicle.

Each driver agent, when approaching the intersection, contacts the intersection manager, sending a REQUEST message. The message contains the vehicle's ID, the arrival time, the arrival speed, the lane occupied by the vehicle in the road link before the intersection and the type of turn. The intersection manager simulates the vehicle's transit through the intersection and informs the driver agent whether its request has conflicts with the already confirmed reservations or not. If the transit does not have conflicts with the confirmed reservations, the intersection manager replies with a CONFIRMATION message, which implies that the driver agent implicitly accepts the reservation parameters. On the other hand, if the transit is not feasible, the intersection manager replies with a REJECTION message.

The intersection manager also uses the *reservation distance* as a criterion for filtering out reservation requests that could generate deadlock situations [2]. The reservation distance is approximated as $v_a \cdot (t_a - t)$, where v_a is the arrival speed of the vehicle, t_a is the arrival time of the vehicle (both contained in the REQUEST message), and t is the current time. For each lane i, the intersection manager has a variable d_i, initialized to ∞. For each reservation request r in lane i, the intersection manager computes the reservation distance, $d(r)$. If $d(r) > d_i$, r is rejected. If, on the other hand, $d(r) \leq d_i$, r is processed as normal. If r is rejected after being processed as normal, $d_i \leftarrow min(\ d_i, d(r)\ )$. Otherwise, $d_i \leftarrow \infty$. Although the use of the reservation distance does not guarantee that a vehicle only gets a reservation if all the vehicles in front of it already have a reservation, it makes it more probable.

3 Reservation-based policies

In the intersection control mechanism proposed originally by Dresner and Stone, the intersection manager processes the incoming requests with a first-come-first-served policy (FCFS). This means that if two vehicles send requests that require the same space-time in the intersection, the vehicle that sends the request first will obtain the reservation. This policy in extreme case could result being quite inefficient. Consider the case in which a set of n vehicles, v_1, v_2, …, v_n, such that v_1's request has conflicts with every other vehicle, but that v_2, …, v_n do not have conflicts with one another. If v_1 sends its request first, its reservation will be granted and all other vehicles' requests will be rejected. On the other hand, if it sends its request last, the other $n-1$ vehicles will have their requests confirmed, whilst only v_1 will have to wait.

Still, other policies for processing the requests, inspired by the research on adversarial queueing theory (AQT), can be employed. The AQT [1] model has been used in the latest years to study the stability and performance of packet-switched networks. In this model, the arrival of packets to the network (i.e., the traffic pattern) is controlled by an adversary that defines, for each packet, the place and time in which the packet joins the system. Each node in the network has a reception buffer for every incoming edge, an output queue for every outgoing edge, and a packet dispatcher that dispatches each incoming packet into the corresponding output queue (or removed, if this is the final node of the packet), using a specific policy. Under these assumptions, the stability of the network system is studied, where stability is the property that at any time the maximum number of packets present in the system is bounded by a constant that may depend on system parameters.

The AQT model and the request processing of the reservation-based control mechanism share some similarities. In the same way a packet dispatcher decides which packet from the reception buffer will be dispatched to the corresponding output queue, so an intersection manager may decide in which order to process a set ot reservation requests, assigning priorities to requests accordingly with its scheduling policy. Taking inspiration from the AQT model, we compared the FCFS policy with 4 universally stable policies, namely *longest-in-system* (LIS), *shortest-in-system* (SIS), *farthest-to-go* (FTG) and *nearest-to-source* (NTS). The LIS policy gives priority to the request of the vehicle which earliest joined the system. The SIS policy gives priority to the request of the vehicle which latest joined the system. The FTG policy gives priority to the request of the vehicle which still has to traverse the longest path until reaching its destination. The NTS policy gives priority to the he request of the vehicle which is closest to its origin, i.e., which has traversed the less portion of its whole route.

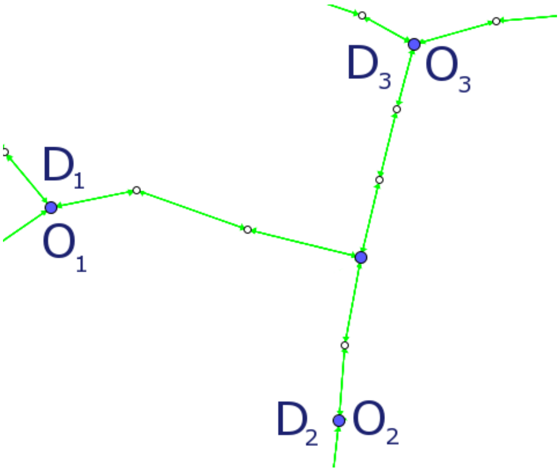

Fig. 1. Scenario 1: single intersection

4 Experimental results

In order to implement the 4 policies, the REQUEST message sent to the intersection manager must contain the necessary additional information: the time stamp when the vehicle joined the system, i.e., when it started to travel, an identifier of the origin location, and an identifier of the destination location.

We evaluated two different scenarios: i) a single intersection scenario (section 4.1) and ii) a network of intersections scenario (section 4.2).

As baseline, we used an intersection regulated by traffic lights with 3 phases (one per incoming road link) of 30 seconds each.

The experiments have been done using a custom, discrete-time, mesoscopic-microscopic simulator. This simulator models the traffic flow on the roads at mesoscopic level [7], where the dynamic of a vehicle is governed by the average traffic density on the link it traverses rather than the behaviour of other vehicles in the immediate neighbourhood as in microscopic models.

Since the mesoscopic model does not offer the necessary level of detail to model a reservation-based intersection, when a vehicle enters an intersection its dynamic switches into a microscopic, cellular-based, simulator, whose update rules follow the Nagel-Schreckenberg [6] model. The cell size is set to 5 meters, and for simplicity we assume that the vehicles cross the intersection at a constant speed, so that any additional tuning of parameters, such as slowdown probability or acceleration/deceleration factors, is not necessary.

4.1 Scenario 1: single intersection

In this scenario we simulate a single intersection with three road links of three lanes each, that connect the origin set $\mathcal{O} = \{O_1, O_2, O_3\}$ with the destination

Table 1. Traffic demands for scenario 1

	λ								
	1	5	10	15	20	25	30	35	40
total spawned vehicles	29	136	285	438	633	716	832	1063	1183

set $\mathcal{D} = \{D_1, D_2, D_3\}$ (see figure 1). When a vehicle is spawned, we assign it an origin $o_i \in \mathcal{O}$ and a destination $d_j \in \mathcal{D}$, with $i \neq j$. We simulate different traffic demands by varying the expected number of vehicles (λ) that, for every O-D pair, are spawned in an interval of 60 seconds. We spawned vehicles for a total time of 10 minutes. Table 1 summarizes the global traffic demands for different values of λ.

The metrics we used to evaluate the performance of the different policies were the *average delay* (sec.), the *average queue time* (sec.) and the *average rejected requests* (% of sent requests).

The average delay measures the increase in travel time due to the presence of the intersection (be it reservation-based or regulated by traffic lights). It is measured running two types of simulation: in the first one, the intersection is regulated by the control mechanism under evaluation and the vehicles must obey the norms that the control mechanism imposes; in the second one, the vehicles travel as if they could transit through the intersection unhindered. The difference between the two average travel times gives us the average delay. Formally:

$$\frac{\displaystyle\sum_{i \in \mathcal{V}} (t_f^i - t_0^i) \; - \; \sum_{i \in \mathcal{V}} (\widehat{t_f^i} - \widehat{t_0^i})}{N}$$

where $\mathcal{V}$ is the set of vehicles, N is the number of vehicles, t_f^i and t_0^i are respectively the time when vehicle i arrives at its destination and when it leaves its origin in the simulation with the intersection regulated by a control mechanism, while $\widehat{t_f^i}$ and $\widehat{t_0^i}$ are respectively the time when vehicle i arrives at its destination and when it leaves its origin if we make the vehicles transit through the intersection unhindered.

The average queue time is the time spent by the vehicles at the intersection queue. Formally:

$$\frac{\displaystyle\sum_{i \in \mathcal{V}} (t_{q_f}^i - t_{q_0}^i)}{N}$$

where $t_{q_f}^i$ is the time when vehicle i leaves the queue of the intersection and $t_{q_0}^i$ is the time when it enters the queue of the intersection.

The average rejected requests is a metric that applies only to the reservation-based policies, and is measured as the ratio between the rejected requests and the sent requests. Formally:

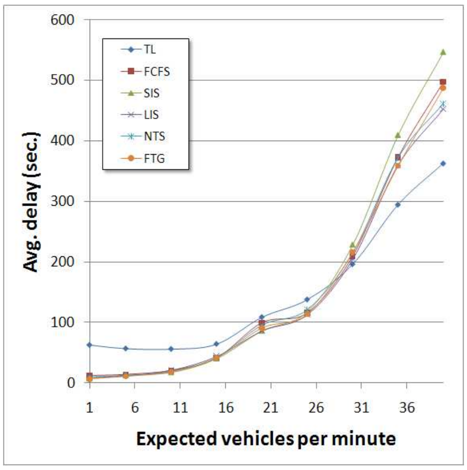

Fig. 2. Average delay

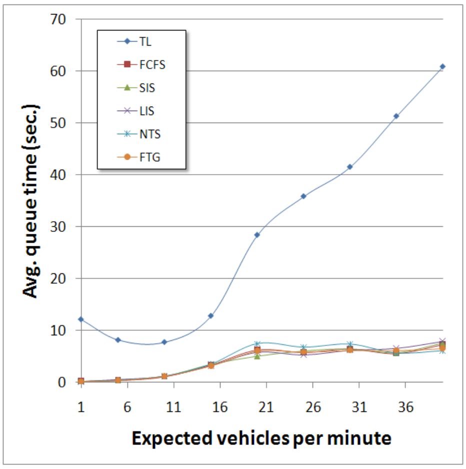

Fig. 3. Average time in queue

$$\frac{\sum_{i\in\mathcal{V}} r^i/s^i}{N}$$

where r^i is the number of rejected requests of vehicle i and s^i is the number of requests sent by vehicle i.

Figure 2 plots the average delay for different traffic demands ($\lambda \in [1, 40]$). When the traffic demand is low ($\lambda \in [1, 15]$), all the reservation-based policies (FCFS, LIS, SIS, FTG, NTS) tend to behave in the same manner, reducing the average delay of about the 65% with respect to traffic lights (TL). Still, when the traffic demand reach a critical value (around $\lambda = 30$), the reservation-based intersection performs worse than the traffic light intersection, with an increase of the average delay between 24% and 50%. Among the reservation-based policies, with high traffic densities the LIS policy is the best one (the increase of the average delay with respect to traffic lights is “only” 24.65%), while the SIS policy is the worst one (with an increase of 50.57%). The reservation-based intersection outperforms a traffic light intersection particularly when the traffic demand is below a certain threshold, because few requests are rejected and the majority of vehicles can transit through the intersection without waiting for a green phase such in traffic light intersections.

Although the experiments have been performed with a custom simulator that is different from that used in [2], the above results seem consistent with the results that we can find in [2]. In fact, in the original work by Dresner and Stone, the reservation-based intersection with FCFS policy outperforms the traffic-light intersection when the traffic density is in the range $[0, 1]$ $vehicles/sec$, while the authors didn’t gave any results for higher traffic demands. In our experiments, the reservation-based intersection starts to perform worse than the traffic-light

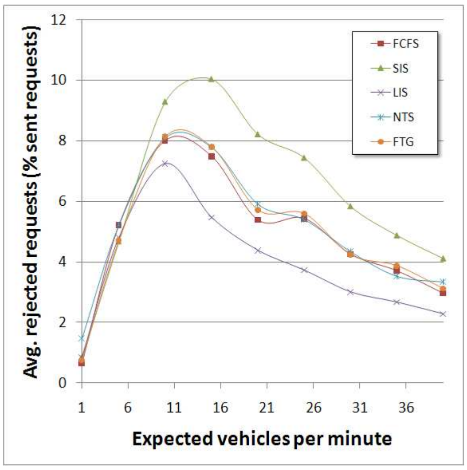

Fig. 4. Average rejected requests

when the expected number of vehicles (λ) is 30, or, given table 1, when the traffic demand is 1.38 *vehicles/sec*, beyond the maximum value evaluated in [2].

Figure 3 plots the average time spent at the intersection queue. Here are noticeable two very distinct dynamics. With a traffic light intersection, the time spent by the vehicles at the intersection queue grows linearly and constantly with the traffic demand. On the other hand, with a reservation-based intersection, the queue time settles around about 7 seconds, independently of the policy in use. This plot gives us an idea of the vehicle's behaviour when approaching the two different types of intersection. If the intersection is regulated by traffic lights, the vehicle proceeds at the speed permitted by the traffic conditions and, once it reaches the intersection, if the traffic light is red it enters the intersection queue. In this way, the more the vehicles approaching the intersection, the longer the waiting time at the intersection queue. With a reservation-based intersection the dynamic of the vehicle approaching the intersection is different. If the vehicle holds a valid reservation, it maintains its speed because it has safety guarantees about its transit through the intersection. On the other hand, if it does not have such reservation, it reduces its speed for safety reasons, and it keeps requesting a reservation to the intersection manager. Thus the collective behaviour is a slower, smoother, traffic flow through the intersection, with few time spent stuck at the intersection.

Finally, we evaluated the reservation-based policies in terms of average rejected requests (as a percentage of the sent requests). Here with rejected request we refer to a request that cannot be granted due to *conflicts* with the already confirmed reservations. Figure 4 plots the results for the different traffic demands under evaluation. With low traffic density, all the policies perform quite similarly, and the rejected requests increase linearly with the number of vehicles

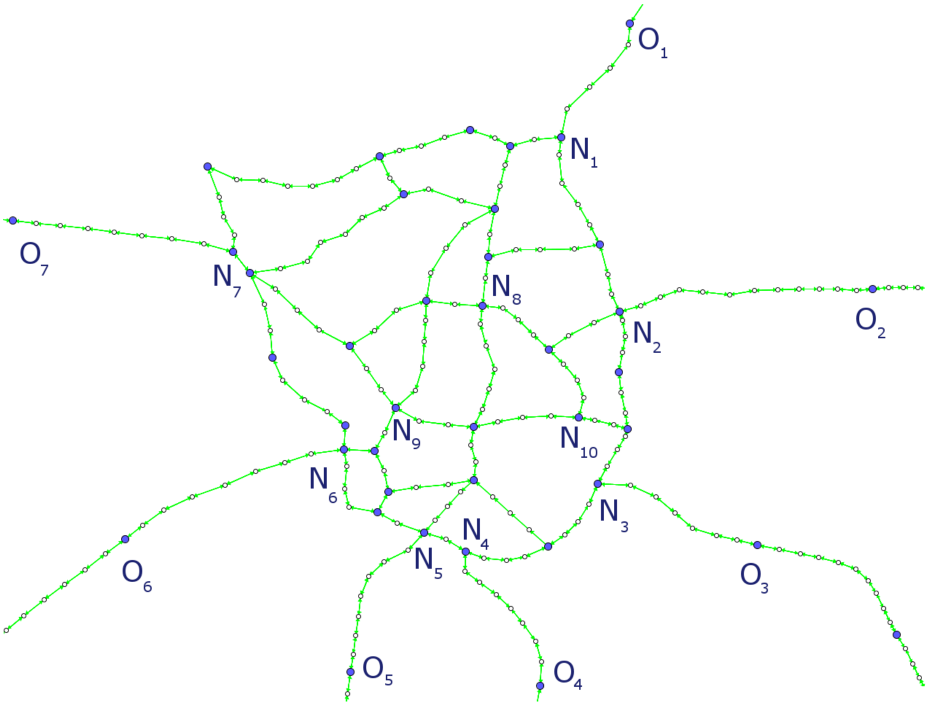

Fig. 5. Scenario 2: network of intersections

approaching the intersection. When the traffic demand reaches a critical point (around $\lambda = 15$), the rejected requests tend to decrease with the traffic demand. The reason of this counterintuitive dynamic is the effect of the *reservation distance*. As the number of rejected requests increases, the reservation distance tends to become smaller, because it is updated with the distance of the closest vehicle whose request has been rejected. With high traffic density, the effect of the reservation distance becomes predominant, filtering out the majority of the reservation requests and processing only those of the nearest vehicles. Since less requests are processed, less conflicts are detected, so that the average rejected requests decrease.

4.2 Scenario 2: network of intersections

In this scenario we simulate an entire network of intersections (figure 5). We defined several locations that serve as origins and destinations for the traffic demand. The vehicles that commute from/to locations $\in \mathcal{O} = \{O_1, O_2, O_3, O_4, O_5, O_6, O_7\}$ form the traffic under evaluation. The vehicles that commute from/to locations $\in \mathcal{N} = \{N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, N_9, N_{10}\}$ serve to add "noise" and to populate the network more realistically.

We aimed at recreating a typical morning peak, with 2 different traffic demands, namely *low* (3986 vehicles) and *high* (11601 vehicles).

The metrics we used to evaluate the performance of the different policies for a given O-D pair (o, d), with $o, d \in \mathcal{O}$, was the average delay per crossed intersection (sec/intersection):

$$\frac{\displaystyle\sum_{(o,d)\in\mathcal{OD}} avgDelay(o,d)/intersections(o,d)}{||\mathcal{OD}||}$$

where $\mathcal{OD}$ is the set of all the O-D pairs (o, d), $avgDelay(o, d)$ is the average delay of the given O-D pair, $intersections(o, d)$ is the number of intersections of the route from o to d, and $||\mathcal{OD}||$ is the number of O-D pairs in the set $\mathcal{OD}$.

Low traffic demand. Table 2 shows the average delay per crossed intersection and the relative delay (made 100 the best policy). At first sight, it seems confirmed that, as for scenario 1, the traffic light intersection is the intersection policy that causes more delay for low traffic demand. This confirms the results of scenario 1: the reservation-based intersection with a good policy takes advantage of low traffic demands, reducing drastically the delay with respect to a traffic lights intersection. With traffic lights, vehicles stop at the intersection even if there are no vehicles on the road link with the green phase. On the other hand, with a reservation-based intersection, few vehicles mean few reservations that are rejected, so that the transit through the intersection speeds up.

The LIS policy is the policy that causes less delay, 6.68 seconds per intersection, about the 21% less than the FCFS. The traffic light intersection is the worst one: a vehicle is delayed 38 seconds when it transits through each intersection it finds on its route, with respect to the $7-8$ seconds of a reservation-based intersection.

Still, it is noticeable how the traffic light intersection has a low standard deviation with respect to its average delay. This is a desirable property from the point of view of the quality of service of the system, which in this way does not penalize nor favour excessively a specific O-D pair, introducing the same delay for every trip through the network.

Table 2. Average delay per crossed intersection (low traffic demand)

	Average delay (sec/intersection)	**stdev.**	**Relative delay**
TL	38.00	8.39	568.77
FCFS	8.13	7.43	121.64
LIS	6.68	5.63	100.00
SIS	7.63	5.68	114.24
FTG	7.12	5.58	106.56
NTS	7.65	5.35	114.44

High traffic demand. Finally, we evaluated the reservation-based policies and the traffic light intersection using a high traffic demand, with a total of 11601 vehicles traveling through the road network. To assess which policy is the best one in reducing delays, we rely again on the average delay per crossed intersection (table 3). When the traffic demand is high, the traffic light intersection (TL) turns out to be the best policy, with 194.38 seconds of delay per crossed intersection. The reservation-based policies perform all slightly worse than the traffic light intersection, with about 230 seconds of delay per crossed intersection. It is interesting to notice that also in this case the TL has the lowest standard deviation, almost the half of the standard deviation of the best performing reservation-based policy. This is a hint that as the demand increases, the performance of a reservation-based intersection becomes more volatile: in some part of the network it could speed up the transit through an intersection, while in other parts it may slow down the transit even more than a traffic light, which behaviour is, on the other hand, more predictable and stable.

Table 3. Average delay per crossed intersection (high traffic demand)

	Average delay (sec/intersection)	**stdev.**	**Relative delay**
TL	194.38	95.72	100.00
FCFS	226.48	154.06	116.51
LIS	229.69	162.04	118.16
SIS	230.33	161.36	118.49
FTG	233.63	159.14	120.19
NTS	225.08	160.67	115.79

5 Discussion

In the experiments described in section 4.1 and 4.2 we evaluated the performance of different policies that a reservation-based intersection manager can employ to process the reservation requests that it receives. One (FCFS) is the policy used in the original work by Dresner and Stone, the others (SIS, LIS, NTS, FTG) are policies inspired by the theory of adversarial queueing. As baseline, we used an intersection regulated by traffic lights (TL) with n phases (one per incoming road link) of 30 seconds each. From the experimental results of scenario 1 (single intersection) and scenario 2 (network of intersections) we can conclude the following:

The reservation-based intersection is suited for low-load situations.
As seen in the experimental results, the reservation-based intersection reduces drastically the average delay when the traffic demand is low. The vehicles are able to transit through the intersection unhindered, they almost do not stop at the intersections, and the allocation capacity of the intersection is maximized. Still, when the demand increases, the throughput of

the reservation-based intersection becomes closer to that of a traffic light intersection, and for high traffic demand it performs even worse than a traffic light intersection. This is because a reservation-based intersection is less robust than a traffic light and its performance is very sensitive to the traffic demand. With many vehicles approaching the intersection, the correct arrival time at the intersection becomes harder to estimate and more sensitive to traffic variations, so that many confirmed requests are cancelled by the vehicles, thus reducing the intersection throughput.

The reservation-based intersection produces a smoother traffic flow. From the analysis done for scenario 1, we can conclude that a reservation-based intersection affects the pattern of the traffic flow. Although the average delay increases with the traffic density, the reservation-based intersection reduces drastically the time spent by the vehicles at the intersection queue, especially in worst case situations: the queue time with high traffic demand ($\lambda \in [20, 40]$) is reduced up to a 80% with the reservation-based intersection. These two metrics suggest that a reservation-based intersection producing a slower, smoother, flow through the intersection, with the vehicles spending less time stuck at the intersection.

FCFS is the simplest policy but it is quite efficient. In spite of its simple behaviour, the FCFS does not perform much worse than the other, more complex, policies. Thus, it can be considered the best choice, since it needs less information: SIS and LIS need to know when the vehicle joined the system, while NTS and FTG need the information about where the vehicles is coming from and where is going to. This fact suggests also that probably we cannot expect great improvements of the efficiency of a single intersection using even more sophisticated policies than those evaluated in this paper. Still, we remark the performnce of the LIS policy. As as shown in the above experiments, it performs better than the FCFS in reducing delays and queue time, especially in low-load situations, when the reservation-based intersection outperforms the traffic light intersection. Furthermore it can be implemented with no much more effort with respect to the FCFS and the extra information it needs cannot be manipulated by the driver agent. For example, when the driver agent starts up, it has no time stamp, so the very first reservation that it will request won't have any time stamp. The intersection manager, detecting that the request has no time stamp, could manage the request with a default time stamp (e.g., the actual time) and then it could "stamp" the driver agent with the actual time so that, for the rest of its trip, it will provide a good approximation of the time when it joined the system.

6 Conclusions

Autonomous vehicle navigation in urban road networks will be possible in the future. To this respect, Dresner and Stone introduced an infrastructure facility that allows reservation-based control of intersection to regulate the transit of vehicles. In this paper we evaluated different policies, inspired by the adversarial queueing theory, that can be employed by this kind of facility, comparing them with the first-come-first-served policy used in the original work.

We showed that the reservation-based intersection is best suited for low-load situations, compared with a traffic light intersection, generating a smoother traffic flow with less time spent at the intersection queue. Furthermore, by empirical evaluation, we showed that, albeit simple, the first-come-first-served (FCFS) policy is quite efficient. Still, some improvements can be obtained using the longest-in-system (LIS) policy, which reduces delays and queue time more effectively than the FCFS, especially in low-load situations.

References

1. M. Blesa, D. Calzada, A. Fernández, L. López, A. Martínez, A. Santos, and M. Serna. Adversarial queueing model for continuous network dynamics. In *Proceeding of Mathematical Foundations of Computer Science*, volume 3618 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2005.
2. K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008.
3. P. Ioannou and C. C. Chien. Autonomous intelligent cruise control. *IEEE Transactions on Vehicular Technology*, 42(4):657–672, 1993.
4. C. W. Kim and J. M. A. Tanchoco. Conflict-free shortest-time bidirectional agv routeing. *International Journal of Production Research*, 29(12):2377–2391, 1991.
5. B. Krogh and C. Thorpe. Integrated path planning and dynamic steering control for autonomous vehicles. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:1664–1669, Apr 1986.
6. K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France 2*, pages 2221–2229, 1992.
7. T. Schwerdtfeger. Dynemo: A model for the simulation of traffic flow in motorway networks. In *Proceedings of the 9th International Symposium on Transportation and Traffic Theory*, pages 65–87. VNU Science Press, 1984.