

# An Active Approach to Guaranteed Arrival Times Based on Traffic Shaping

Dan Marinescu, Jan Čurn, Marco Slot, Mélanie Bouroche and Vinny Cahill

**Abstract**—To address the goal of providing drivers on highways with guaranteed arrival times, we propose a traffic management system that combines virtual slots with semi-autonomous driving to shape traffic and prevent congestion. Two algorithms that address aligning vehicles into slots and efficient merging from three to two lanes are proposed. Furthermore, an implementation and evaluation of these algorithms using the VISSIM traffic simulator is presented. Our initial results indicate that a slot-based system has the potential to be used to guarantee arrival times and provide a significant overall increase in efficiency when compared against a human driving model.

## I. INTRODUCTION

Increasing road usage has led to congestion on highways as well as on rural and urban roads. In the past, congestion was commonly addressed by building new roads. Due to its financial, social and environmental implications this approach is becoming increasingly infeasible [1]. Starting from the early 1980s [2], an alternative solution to this problem has been investigated by designing and deploying intelligent transportation systems (ITS). Besides reducing congestion, ITS also addresses problems such as safety and cost reduction. In this context, our work focuses in general on how ITS can be used to reduce highway congestion by increasing traffic efficiency, and more specifically on providing guaranteed arrival times to drivers on highways.

Most previous work in this area has been focused on avoiding congestion [3], [4], increasing the capacity of the highway [5], [6] and so reducing average travel time. It has been observed, however, that drivers would prefer to have predictable travel times in preference to on-average shorter, but unpredictable, travel times [7]. The motivation for this ranges from commuters wanting to make sure they arrive at their workplace in time, to businesses wanting to make sure that their deliveries arrive in time. Despite this, to the extent of our knowledge, little attention has been paid to providing guaranteed arrival times on highways.

Current state-of-the-art approaches to obtaining reliable travel time estimates are passive in the sense that they attempt to predict travel times using both historic traffic data and real-time traffic observations [8]. However, these approaches

The work described in this paper was partly supported by the Irish Higher Education Authority Programme for Research in Third Level Institutions (as the Networked Embedded Systems Centre) and by Science Foundation Ireland grant 03/CE2/I303 1

Authors address: Distributed Systems Group, School of Computer Science and Statistics, Trinity College Dublin, Ireland. {marined, curnj, slotm, bourocm, vjcahill}@scss.tcd.ie

Vinny Cahill is also with Lero - The Irish Software Engineering Research Center.

fail in the presence of unexpected events such as accidents and road works, which cause sudden bottlenecks and may lead to prolonged congestion [9]. To address the uncertainties arising from the occurrence of such events, we propose an active approach to providing guaranteed arrival times.

Unlike passive travel-time prediction, an active approach implies some degree of actuation on vehicles so that initial arrival-time guarantees can be maintained. In the case of bottlenecks, actuation aims to maintain the same throughput in the area of the bottleneck as elsewhere on the highway, thus maintaining the arrival-time guarantee. The problem of maintaining travel-time guarantees in the presence of unexpected bottlenecks is thus reduced to the problem of maintaining the same throughput. We address this problem by designing a traffic management system (TMS) that actuates upon the vehicles by combining virtual slots with predefined behaviour generated by the infrastructure and vehicle-based semi-autonomous driving. Our hypothesis is that the predictable nature of driving in a slot with predefined behaviour allows for strong arrival time guarantees to be provided. Furthermore, predefined slot behaviours can be designed so that a cooperative driving pattern emerges that is more efficient than the current driving pattern. This slot-based TMS is implemented and evaluated on a bottleneck scenario caused by a three-lane to two-lane merging.

## II. USING VIRTUAL SLOTS

Traffic shaping is a general term that encompasses several techniques used in computer networks to optimise traffic and guarantee delivery times. One technique in particular, called “leaky bucket” [10], is used to control the rate at which packets enter a network, similarly to the way in which ramp metering is used to control access to a highway [11]. This helps prevent so-called data bursts, which are similar to vehicle platoons and can be a cause of congestion. We adapt these techniques to highway traffic by mapping vehicles to slots and show how they can be used to provide guaranteed arrival times. Unlike ramp metering, which controls the access of vehicles to the highway, we apply this traffic shaping technique on-the-fly, when necessary to prevent congestion, to vehicles that have already joined the highway.

### A. Virtual slots

In 2005, Morla proposed that congestion-free travel can be guaranteed by assigning slots to vehicles [12]. In Morla’s vision, a slot represents a time-space corridor negotiated among vehicles. The speed of slots on the same lane is the

same and vehicles are driven by human drivers, supported by a driver assistance system.

The same concept was advocated two years later by Ravi et al., in a more business-centric vision [13]. Their approach permits the coexistence of a slot-based system with existing driving patterns by providing a high-priority lane. In similar work [14], Cahill et al. proposed a time division multiple access (TDMA) inspired approach to allocate slots to vehicles on the highway.

Our traffic shaping approach builds on this idea of a slot. We extend the previous notion of a slot moving at a constant speed in a single lane to a more flexible one, in which a slot can accelerate, decelerate and change lanes. A vehicle driving in a slot should replicate the behaviour of its slot. In this paper we propose the use of slots with predefined behaviour. By predefined behaviour we understand an a priori known speed profile and trajectory. As such, we define a slot  $S$ :

$$S = \{z, p, t, b\},$$

where  $z$  represents the size of the slot,  $p$  represents its position at time  $t$  and  $b$  the predefined behaviour as a sequence of accelerate, decelerate and lane changing manoeuvres.

### B. Requirements for driving in slots

Based on our previous definition of a slot we can see that, for a vehicle to drive in a specific slot, it needs to be aware of both *local information*:

- the vehicle's own speed
- the vehicle's own acceleration
- the vehicle's own position
- the lane in which the vehicle is currently driving
- the presence of other vehicles ahead.

as well as *slot information*:

- the slot's position and lane at a specific point in time
- the slot's size
- the slot's behaviour.

The local information can be easily obtained using technology already available in most vehicles. Speed and acceleration measurements are very accurate, and global navigation satellite system (GNSS) positioning devices such as differential GPS (DGPS) have a accuracy of typically 0.2-0.5m. Error in the longitudinal direction can be compensated by the size of the slot. For example, a positioning error of up to one meter means that the slot needs to be at least two meters larger in length than the car. The error in the lateral direction is not relevant since we assume that a vehicle can determine that lane on which it's currently driving, possibly using a combination of stereo vision and digital maps.

For a vehicle to decide whether it can move into a slot it first needs to determine whether the slot is empty or not. This can be achieved using radar- or lidar-based sensing technologies such as the ones used by adaptive cruise control (ACC) systems [15].

The slot information is provided by the infrastructure and needs to be communicated to the vehicle. This form of communication is generally referred to as vehicle-to-infrastructure (V2I) communication. Depending on the scenario, V2I communication might have stringent real-time

requirements or not. For example, for a merging scenario that was caused by road works, the slot information could be provided in advance using already widely-deployed 3G networks, as well as 4G technologies such as WiMAX and LTE that are in the process of being deployed on a wide scale. For other scenarios, such as when dealing with accidents, the information might need to be communicated in real-time and so real-time vehicular communication would be required. A considerable research effort is currently targeting the development of real-time vehicular communication [16]. Since the merging scenario described in this paper does not have stringent real-time communication requirements, the use of 3G communication with a latency of around 600 ms [17] is sufficient.

The accuracy of the slot positioning depends on the accuracy of the initial slot position, the accuracy of the digital maps and the accuracy of the dead-reckoning method used. Since the slot can be generated at a well-known position and has a predefined behaviour, this accuracy would be very good and the eventual error range neglectable.

Once a vehicle has the required local and slot information it can use the local information to generate its own behaviour, matching the behaviour of the slot provided in the slot information. The vehicle needs to use this information to assist the driver. Such assistance can range from fully autonomous driving to simply providing the driver with the information necessary to replicate the behaviour of the slot. While fully autonomous driving has already been demonstrated on previous occasions [5], [6], it is now commonly believed that this form of driving will not be widely deployed in the near future. On the other hand, relying solely on drivers following the advice of the vehicle to perfection is also unrealistic. As such, a balance needs to be obtained between autonomous driving and manual driving. In general, we refer to this as semi-autonomous driving.

From the perspective of the longitudinal (throttle) and lateral (steering) control, several forms of semi-autonomous driving can be designed. While in this paper the focus is more on the slot coordination/scheduling aspects rather than low-level vehicle control, we believe that a combination of autonomous longitudinal control and assisted lateral control such as a head-up display can be used for slot-based driving. Such forms of semi-automated vehicle control, in particular longitudinal [18] but also lateral control [19] have been designed, implemented and proven to work. Furthermore, technologies such as ACC and lane departure warning (LDW) [20] have already been deployed in production vehicles.

### III. A TMS BASED ON SLOTS

The nature of human driving is inherently competitive. While this is not a major problem under free flow conditions, when the volume increases or the highway suffers a drop in capacity, the competitive nature of human driving becomes an impediment [21]. This competitive nature is somehow limited by the set of driving rules. Unfortunately, these rules were chiefly designed to address safety issues. Too often we find drivers driving on the fast lane instead of the slow

lane, not making space for faster cars or cars that intend to exit the highway. No legally binding driving rules exist forbidding such behaviour or, where they exist, they are seldom enforced. The result is the same: an inefficient form of driving, that is competitive rather than cooperative.

### A. Approach

To address this problem we have designed a TMS based on the slot model described in the previous section. Unlike current driving behaviour, the predefined nature of a slot means that one can design a set of behaviours that, when assigned to different slots, work together in a cooperative manner. As stated in the introduction, our hypothesis is that such behaviour is not only predictable, thus allowing for strong arrival time guarantees to be provided, but can also be more efficient than the current driving pattern. Since driving in a slot reduces the freedom of choice of the driver we advocate the use of slots only in situations where the competitive nature of human driving leads to congestion - such as highway bottlenecks - and not under free flow conditions. This way, drivers will have to relinquish their driving freedom only for a limited period of time, which we believe that would increase the driver acceptance of the system.

Furthermore, the nature of driving within a slot implies transferring the longitudinal control of the vehicle from the human driver to the vehicle, which poses a great advantage. Currently, the minimal distance between two vehicles is limited by slow human reaction times, which varies between 1.2 and 1.7 seconds [22]. When the human driver is removed from the control loop the reaction times are significantly reduced. This means that slots can be packed more tight together on the highway and so the capacity of the highway increases. Part of this increase in capacity is used to counter the negative effect that positioning errors have on the size of the slot, and therefore on the capacity of the highway, as in the previous section.

Our TMS comprises of three different components: the *Manager* and the *Road side units* representing the infrastructure on one side, and the *Vehicles* on the other side. This is shown in Fig. 1. These entities work together to accomplish two main tasks: assist the drivers to align vehicles into slots and, once this is achieved, assist the drivers to drive within their respective slots.

The *Manager* comprises of a *Slot generator* and a *Communication layer*. As the name suggests, the *Slot generator* is responsible for generating slots with an associated predefined behaviour. The behaviour of a slot is assigned from a set of predefined slot behaviours. This set of behaviours is scenario-specific and is computed by the TMS to address the particular scenario. A detailed description of the slot behaviour generation process is beyond the scope of this paper.

For a vehicle to be able to replicate the behaviour of a slot, it needs to be able to know the current position, speed, acceleration, trajectory, as well as the set of manoeuvres the slot is going to perform in the future. Since the behaviour of

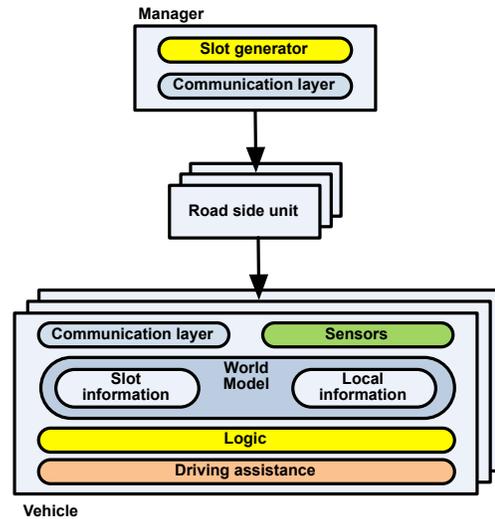


Fig. 1. Components of the TMS

a slot is predefined, the current position can be determined by knowing this behaviour and the position of the slot at a certain point in time. This information is broadcast to the Vehicles by the Communication layer via the Road side units.

The Communication layer of a Vehicle receives this information and stores it into the *World Model* as *Slot information*. This is augmented with the *Local information* gathered by *Sensors*. The *Logic* of the vehicle uses the information in the Knowledge base to perform two tasks: mapping the vehicle to an empty slot and, once this is achieved, assisting the driver into replicating the behaviour of the slot. The interfacing with the human driver as well as the direct control over the vehicles takes place through the *Driving assistance*.

### B. Getting vehicles into slots

Before a vehicle can drive within a slot, it needs to move into the slot, driving at the same speed as the slot. The slot can either be assigned by the infrastructure or the vehicle can find the empty slot that suits it best. We call this the vehicle-to-slot mapping (VTS) algorithm and differentiate between mappings performed when vehicles enter the highway and mappings performed on the fly with vehicles being informed when they need to move into slots.

The VTS algorithm for vehicles entering the highway is fairly straight forward when the access to the highway is controlled, e.g., by means of ramp metering. However, due to the likely reluctance of drivers to give up their driving freedom, we believe that slots should not be used in free flow traffic conditions, but only when this is mandatory to prevent congestion and so maintain the arrival time guarantees. For this reason we chose to design an on-the-fly VTS algorithm. The algorithm is based on the idea of generating slots with a headway equal to the typical headway between vehicles driving on the highway and a speed equal to the average speed of the fastest lane, aiming to maintain the same throughput after the vehicles have been aligned into slots as before the alignment. A possible problem of this approach

is when dealing with traffic patterns that consist of platoons of vehicles driving at small intra-platoon headways but large inter-platoon headways. To avoid having empty slots between vehicles the VTS algorithm defines a start and end point between which the alignment takes place. Vehicles that have empty slots ahead of them will accelerate to the maximum speed and pass as many empty slots as they can before reaching another vehicle or the end point of the VTS, when they need to be aligned to a slot. The distance between the start and end points of the VTS algorithm as well as the difference between the maximum speed of a car and the speed of the slot are parameters that need to be configured to counterbalance the maximum size of a platoon, therefore preventing the build up of queues of vehicles. This tuning of the algorithm is subject to the evaluation. As such, while some vehicles will need to slow down and others will need to speed up, overall the throughput is the same and so the guarantees are maintained. The VTS algorithm is presented in 1.

```

begin VTS
  Vehicle receives slot information and the start and end points of
  the VTS;
  if Vehicle can sense another Vehicle ahead then
    while Vehicle can get into the next free slot before the end
    point of VTS do
      control the speed to be at exactly 1 inter-slot headway
      behind Vehicle in front ;
    end
    if the distance to the Vehicle ahead is more than 1 inter-slot
    headway then
      get into the first slot ahead;
    end
  end
else
  while Vehicle cannot sense a Vehicle ahead and Vehicle can
  get into the next free slot within the border of VTS do
    accelerate to maximum speed and maintain this speed;
  end
  if Vehicle can sense another Vehicle ahead then
    while Vehicle can get into the next free slot within the
    border of VTS do
      control the speed to be at exactly 1 inter-slot
      headway behind Vehicle in front ;
    end
    if the distance to the Vehicle ahead is more than 1
    inter-slot headway then
      get into the first slot ahead;
    end
  end
else
  get into the first slot ahead;
end
end

```

**Algorithm 1:** VTS algorithm

The VTS algorithm is applied by each vehicle separately. It uses local and slot information to determine the positions of vehicles and slots ahead of the vehicle and based on this information executes one of the following three low-level control loops:

- control the speed to be at exactly one inter-slot headway behind Vehicle in front
- get into the first slot ahead
- accelerate to maximum speed and maintain this speed

### C. Driving within a slot

While the algorithm for aligning vehicles into slots is generic, driving within a slot simply means following the predefined behaviour of the slot. This predefined behaviour is not generic but rather depends on the traffic scenario that needs to be addressed. Such a scenario is presented in the next section, along with a set of predefined slot behaviours that together form a cooperative and efficient form of driving.

## IV. SCENARIO AND VISSIM IMPLEMENTATION

To evaluate the impact of our slot-based TMS system, we chose a scenario where a three lane highway experiences a drop in capacity from three to two lanes. This creates a bottleneck which, under high traffic volume, will cause congestion. Such bottlenecks can be caused by road works or accidents and is a common cause for congestion on highways [21].

There are two factors that contribute to the occurrence of congestion in our scenario. First of all, when the combined traffic volume of the three-lane segment exceeds the capacity of the two-lane segment, the drop in capacity alone can be held responsible for the resulting congestion. However, congestion can still arise when the total traffic volume is lower than the capacity of the two-lane segment, due to the inefficient merging performed by human drivers. We propose a lane merging algorithm (LMA) that addresses both factors. It reduces the inter-vehicle headway by transferring the longitudinal control of the vehicle from the driver to the vehicle itself, thus increasing the capacity of the highway. Furthermore, it assists the driver to perform a coordinated merging procedure which is more efficient than the unassisted, competitive merging procedure performed by human drivers.

### A. The lane merging algorithm

The LMA is specific to our scenario and builds on the alignment of vehicles into slots as a result of the VTS algorithm. A visualisation of the LMA is shown in Figure 2.

The LMA defines a set of predefined slot behaviours which together perform a more efficient merging procedure. As can be observed from Figure 2, the behaviours are classified in two main categories: A and B. Category A slots are immediately followed, within the same lane, by category B slots. Each of the two categories has three subcategories, numbered 1, 2 and 3 according to the lane in which the slots are initially moving. This results in six different slot behaviours. At the beginning the slots are aligned as a result of the VTS algorithm. From this starting position the B-slots reduce the distance to A-slots, leaving an empty space between these newly formed pairs (Phase 1). This space is then used by A2 slots to move to lane 1 (Phase 2). Similarly, A3 and B3 slots use the remaining space to change to lane 2 (Phase 3), leaving no slot and thus no car on lane 3.

While slots in different positions have different behaviour, their combined emergent behaviour is a merging from three to two lanes that does not require slots to slow down below their initial speed and thus maintains the throughput through

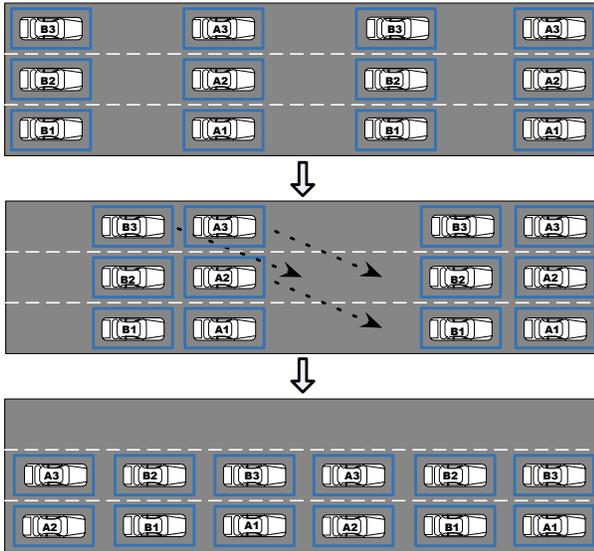


Fig. 2. Visualisation of the lane merging algorithm

the bottleneck at the same level as elsewhere on the highway. The LMA defines start and end points for each of the three phases and is shown in Algorithm 2.

```

begin LMA
  while Vehicle has not reached the end point of the LMA do
    if Vehicle is within Phase 1 then
      if Vehicle is in a slot of category B then
        Reduce the gap to the slot ahead to a predefined
          minimum headway;
      end
    else
      Maintain speed ;
    end
  end
  else if Vehicle is within Phase 2 then
    if Vehicle is in a slot of category A2 then
      Change to lane 1;
    end
  end
  else if Vehicle is within Phase 3 then
    if Vehicle is in a slot of subcategory 3 then
      Change to lane 2;
    end
  end
end
end

```

Algorithm 2: Lane merging algorithm

Depending on the slot category and the phase it is in, a vehicle might need to perform one of the following control loops to stay within its slot:

- reduce the gap to the slot ahead to a predefined minimum headway
- maintain speed
- change to the lane to the right of the current lane

Note that at no point during the LMA will vehicles travel at a speed slower than the the initial speed of the slot set by the VTS algorithm. This means that the overall throughput during the LMA and after the LMA is completed is the same as that of the VTS, which in turn was equal to the throughput on the three-lane segment before the vehicles were aligned

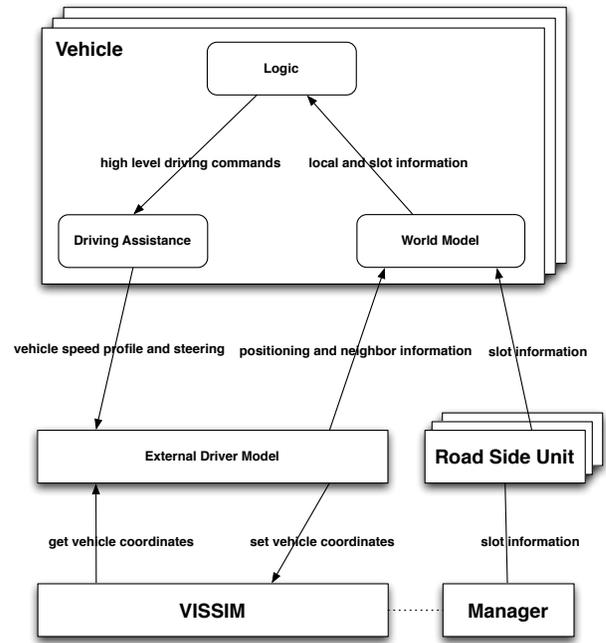


Fig. 3. VISSIM-based framework for the TMS

into slots. This means that the combination of VTS and LMA maintains the throughput in the bottleneck area and so arrival time guarantees can be provided.

### B. Implementation of the slot-based TMS in VISSIM

To validate our concept we implement the slot-based TMS and the two algorithms using the VISSIM traffic simulator [23]. We used VISSIM's external driver model DLL to replace VISSIM's build-in driver models with our slot-based driving model.

VISSIM is a time step microscopic simulator. As such, the external driver model is called at each time step for each of the vehicles in the network. VISSIM provides the external driver model with information about each vehicle, such as the vehicle's own speed, current lane, acceleration and relative steering, as well as information about the vehicle's neighbours, such as relative distance and speed. The task of the external driver model is to compute and return to VISSIM variables such as the desired speed and lane. This basically allows the driver model to control both the longitudinal and the lateral tasks of the vehicles.

Based on the previously described interaction model between VISSIM and the external driver model DLL we designed a framework which implements the slot-based TMS. The VTS and LMA algorithms can be plugged in the framework as well as replaced with other scenario-specific algorithms. This framework is shown in Figure 3.

Each *Vehicle* receives local information from VISSIM as well as slot information from the *Manager* through the *External Driver Model* and *Road Side Units* respectively. The *World Model* component organises and provides a higher level API to the *Logic* component. Besides this, the *World Model* also filters the exhaustive local information received

from VISSIM by discarding information which is outside the sensing range of the vehicle’s sensors, emulating the sensing capabilities of a vehicle. No artificial errors are introduced with respect to positioning measurements since the size of the slot as defined in our system model caters for these errors. The World Model is a multi-purpose, complex component that we are currently developing and its detailed description is outside the purpose of this paper.

The Logic component uses a combination of VTS followed by LMA to address the specific problem of our scenario. Using the high-level API provided by the World Model, the Logic determine which control loop needs to be applied currently and informs the *Driving Assistance* component. This in turn implements the low-level control loop, which results in speed, acceleration and steering commands. Since we are not using real cars, these commands are passed to the External Driver Model and implicitly to VISSIM instead of the human driver or the vehicle. For the purpose of this paper we reduced the complexity of the problem by assuming an ideal form of communication (no packets are lost or delayed) between the Vehicles and the Road Side Units or the Manager and the Road Side Units.

## V. EVALUATION

For the evaluation of the slot-based TMS we modeled the three-to-two lane merging scenario within VISSIM. Vehicles join the highway at the beginning of a three-lane segment, which is 2472 meters long. This segment then merges into a 235 meters long two-lane segment. The length of the three-lane segment was chosen to sufficiently accommodate both the VTS and LMA algorithms. When vehicles initially join the highway, their behaviour is controller by the Wiedemann driver model. After 372 meters, the VTS algorithm starts. A length of 1200 meters was chosen to sufficiently accommodate the VTS algorithm. 1572 meters after joining the highway, vehicles start the LMA procedure, for which 900 meters were assigned.

Since the focus of this work is on high-level optimization rather than low-level vehicle control, we reduced the complexity of the problem by assuming all vehicles have identical capabilities. The desired speed of the vehicles was set at 30  $m/s$ , while the acceleration and deceleration that can be achieved by a vehicle at any time are constant at 4  $m/s^2$  and -6  $m/s^2$ . The maximum speed a vehicle can achieve was set at 40  $m/s$ , while the headway between vehicles oscillates around 1.5  $m$ .

The travel time of each vehicle is measured from the time the vehicle joins the three-lane segment to the time it leaves the two-lane segment. The average travel time of vehicles during one hour is used as performance metric. We evaluated our slot-based TMS against the Wiedemann 1999 driver model, which is used by VISSIM to simulate human driving on highway [23]. For this purpose, we performed 10 different runs for traffic volumes of 2000, 3000, 4000 and 5000 vehicles per hour. The results of our evaluation are shown in Figure 4 and Figure 5.

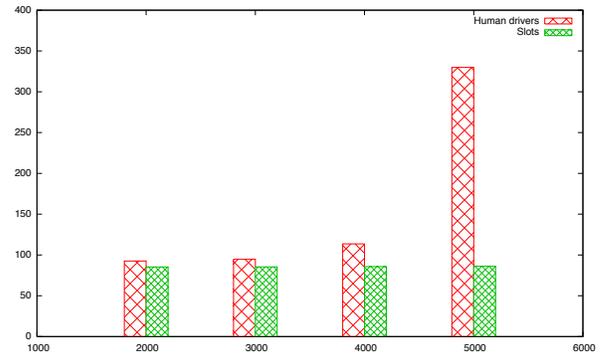


Fig. 4. Average travel times over 10 runs

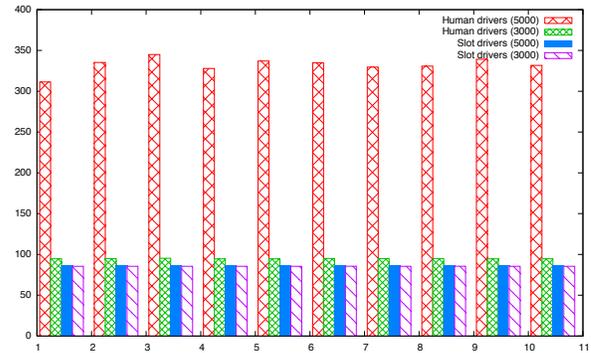


Fig. 5. Average travel times for each of the 10 runs

Figure 4 shows the average travel time over 10 simulation runs of our slot-based TMS compared against human drivers as simulated by the Wiedemann 1999 driver model. From this graph it can be observed that there is little difference in the average travel times of human drivers for traffic volumes of 2000 and 3000 vehicles. However, for traffic volumes of 4000 and 5000 vehicles, a dramatic increase in the average travel times of human drivers can be observed. The explanation for this is that the increase in traffic volumes amplifies the effects of inefficient merging, leading to congestion. When the slot-based TMS is used, the average travel times of the vehicles are consistent and always lower than those of the human drivers for each of the tested traffic volumes. This proves our hypothesis that the predictable nature of slot-based TMS can be used to provide travel time guarantees in the presence of bottlenecks.

Note that, for traffic volumes of 2000 and 3000 vehicles, the lower average travel time of the slot-based TMS compared to human drivers is due to the nature of the VTS algorithm, in which vehicles that cannot sense another vehicle ahead of them accelerate to their maximum speed and maintain this speed until they either sense another vehicle ahead or they approach the limit of the VTS and they need to move into a slot.

Figure 5 shows the variation in average travel times over 10 simulation runs for traffic volumes of 3000 vehicles and 5000 vehicles for both human drivers and the slot-based TMS. For 3000 vehicles, the average times of human

drivers and slot-based TMS are constant. However, this is not the case for situations where congestion occurs, as can be observed from the graph. For 5000 vehicles, a significant variance occurs in the average travel times of human drivers. Unlike human drivers, the travel times corresponding to the slot-based TMS are constant. These results indicate that slots should be used to provide arrival time guarantees in situations where congestion could arise, but are not mandatory in free flow traffic.

## VI. CONCLUSION AND FUTURE WORK

Our work extends the idea of vehicles driving in slots by introducing the concept of predefined slot behaviour. Based on this, we have designed a TMS that shapes traffic by mapping vehicles to slots. When driving in a slot, a vehicle follows the predefined behaviour of the slot. Within the framework of this TMS we have designed and implemented in VISSIM one generic algorithm for aligning vehicles to slots and one scenario-specific algorithm for efficient lane merging.

The results of our evaluation indicate that the slot concept can be used to provide guaranteed arrival times to vehicles driving on highways. For the lane merging scenario, the slot-based system helps prevent congestion that would otherwise occur when a high volume of traffic needs to pass the bottleneck.

To the extent of our knowledge, this is the first experimental evaluation of the benefit of using slots. In the future, we will build on this initial work by focusing on lower-level aspects such as vehicle control and coordination. The next step will be implementing the low-level vehicle control required for slot driving in car-like robots.

Furthermore, in the process of moving from simulation to actual vehicles we plan to address issues which, for the sake of reducing complexity, were left out of this initial work, such as failure modes and heterogeneous vehicles.

## REFERENCES

- [1] P. Ioannou, Y. Wang, and H. Chang, "Integrated roadway / adaptive cruise control system: Safety, performance, environmental and near term deployment considerations," Tech. Rep., July 2007.
- [2] L. Figueiredo, I. Jesus, J. A. T. Machado, J. R. Ferreira, and Martins, "Towards the development of intelligent transportation systems," in *Intelligent Transportation Systems Proceedings*, 2001, pp. 1206–1211. [Online]. Available: <http://dx.doi.org/10.1109/ITSC.2001.948835>
- [3] A. Ghods, A. Kian, and M. Tabibi, "Adaptive freeway ramp metering and variable speed limit control: a genetic-fuzzy approach," *Intelligent Transportation Systems Magazine, IEEE*, vol. 1, no. 1, pp. 27–36, 2009. [Online]. Available: <http://dx.doi.org/10.1109/ITS.2009.932718>
- [4] D. de Bruin, J. Kroon, R. van Klaveren, and M. Nelisse, "Design and test of a cooperative adaptive cruise control system," in *Intelligent Vehicles Symposium*, 2004, pp. 392–396. [Online]. Available: <http://dx.doi.org/10.1109/IVS.2004.1336415>
- [5] R. Horowitz and P. Varaiya, "Control design of an automated highway system," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 913–925, 2000. [Online]. Available: <http://dx.doi.org/10.1109/5.871301>
- [6] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, no. 3, pp. 155–161, 2002. [Online]. Available: <http://dx.doi.org/10.1109/TITS.2002.802929>
- [7] W. Recker, H. M. Zhang, L. Chu, A. Chen, and M. McNally, "Development of methods and tools for managing traffic congestion in freeway corridors," in *Intelligent Transportation Systems Conference Proceedings*, 2006, pp. 30–37. [Online]. Available: <http://dx.doi.org/10.1109/ITSC.2006.1706714>
- [8] J. D. Margulici and X. Ban, "Benchmarking travel time estimates," *Intelligent Transport Systems, IET*, vol. 2, no. 3, pp. 228–237, 2008. [Online]. Available: <http://dx.doi.org/10.1049/iet-its:20080002>
- [9] P. Yulong and D. Leilei, "Study on intelligent lane merge control system for freeway work zones," October 2007, pp. 586–591. [Online]. Available: <http://dx.doi.org/10.1109/ITSC.2007.4357778>
- [10] J. Truner, "New directions in communications (or which way to the information age?)," vol. 24, no. 10, pp. 8–15, October 1986.
- [11] M. Papageorgiou and A. Kotsialos, "Freeway ramp metering: an overview," in *Intelligent Transportation Systems Proceedings*, 2000, pp. 228–239. [Online]. Available: <http://dx.doi.org/10.1109/ITSC.2000.881058>
- [12] R. Morla, "Vision of congestion-free road traffic and cooperating objects," Tech. Rep., November 2005.
- [13] N. Ravi, S. Smaldone, L. Iftode, and M. Gerla, "Lane reservation for highways (position paper)," in *Intelligent Transportation Systems Conference Proceedings*, 2007, pp. 795–800. [Online]. Available: <http://dx.doi.org/10.1109/ITSC.2007.4357746>
- [14] V. Cahill, A. Senart, D. C. Schmidt, S. Weber, A. Harrington, and B. Hughes, "The managed motorway: real-time vehicle scheduling: a research agenda," in *HotMobile '08: Proceedings of the 9th workshop on Mobile computing systems and applications*. New York, NY, USA: ACM, 2008, pp. 43–48. [Online]. Available: <http://dx.doi.org/10.1145/1411759.1411771>
- [15] R. Abou-Jaoude, "Acc radar sensor technology, test requirements, and test solutions," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 4, no. 3, pp. 115–122, 2003. [Online]. Available: <http://dx.doi.org/10.1109/TITS.2003.821286>
- [16] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 2, pp. 3–20, June 2009. [Online]. Available: <http://dx.doi.org/10.1109/SURV.2009.090202>
- [17] C. Wewetzer, M. Caliskan, K. Meier, and A. Luebke, "Experimental evaluation of umts and wireless lan for inter-vehicle communication," August 2007, pp. 1–6. [Online]. Available: <http://dx.doi.org/10.1109/ITST.2007.4295880>
- [18] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 4, no. 3, pp. 143–153, 2003. [Online]. Available: <http://dx.doi.org/10.1109/TITS.2003.821292>
- [19] S. Kumarawadu and T. T. Lee, "Neuroadaptive combined lateral and longitudinal control of highway vehicles using rbf networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 4, pp. 500–512, December 2006. [Online]. Available: <http://dx.doi.org/10.1109/TITS.2006.883113>
- [20] R. Bishop, *Intelligent Vehicle Technology And Trends (Artech House Its Library)*. Artech House Publishers, May 2005. [Online]. Available: <http://www.worldcat.org/isbn/1580539114>
- [21] T. Vanderbilt, *Traffic: Why We Drive the Way We Do (and What It Says About Us)*, 1st ed. Knopf, July 2008.
- [22] D. B. Maciuga and K. J. Hedrick, "Brake dynamics effect on ahs lane capacity," in *Future Transportation Technology Conference & Exposition*, August 1995.
- [23] PTV, "Vissim traffic simulator." [Online]. Available: <http://www.ptvag.com/software/transportation-planning-traffic-engineering/software-system-solutions/vissim/>