Merging Lanes—
Fairness through Communication

D. Baselt[1], F. Knorr[2], B. Scheuermann[3], M. Schreckenberg[2], and M. Mauve[1]

[1] Department of Computer Science, Heinrich Heine University, 40225 Düsseldorf, Germany,

e-mail: SURNAME@cs.uni-duesseldorf.de

[2] Department of Physics, University of Duisburg-Essen, 47057 Duisburg, Germany,

e-mail: SURNAME@ptt.uni-due.de

[3] Department of Computer Science, Humboldt University, 10099 Berlin, Germany,

e-mail: scheuermann@informatik.hu-berlin.de

Corresponding author: Daniel Baselt, phone: +49 171 9537974

Abstract

The merging of two lanes is a common traffic scenario. In this paper we derive a formal model for the behavior of vehicles in this scenario. We discuss the question of how fairness of a merging process can be defined and introduce the notion of free-flow fairness. We first show how optimal fairness could be achieved if all vehicles were omniscient and willing to follow a given strategy. We then move to a more realistic setting, where only a subset of vehicles participates in our merging scheme and where wireless communication is limited and unreliable. By means of analysis and simulation we show that a simple beacon-based approach yields very good fairness even if only 1 % of the vehicles participate.

Index Terms

Vehicular ad hoc networks, Intelligent transportation systems, Cooperative systems, Fairness

Merging Lanes—
Fairness through Communication

1 Introduction

On-ramps are a particular critical part of any road network. Congestion or even a complete breakdown of traffic may occur, if the traffic on on-ramps is not managed properly [1], [2]. The conventional approach to solve this problem is ramp metering [3]: the inflow from the on-ramp is limited by traffic lights to prevent breakdown and to maximize the flow on the main road.

Not only has the principle of operation of common on-ramp metering algorithms recently been criticized [4], [2], this approach also has another main disadvantage. The restricted inflow from the on-ramp often causes congestion on the on-ramp which may propagate upstream into the subsequent road network. Moreover, depending on the main road's traffic flow, vehicles in the on-ramp may have to wait for a considerable and unfair amount of time before they are allowed to enter the main road. It is very hard to take the effects into account properly, when designing conventional ramp metering.

In this paper we investigate the use of car-to-car communication to manage the traffic on on-ramps while avoiding the drawbacks of conventional ramp metering. Prior work in this area, e.g., [5], [6] has focused on optimizing throughput. In the work presented here we go one step further and look at the fairness of traffic management on on-ramps when using car-to-car communication.

In a first step we will propose a fairness metric for merging algorithms based upon waiting times. We will show that zipper merge can result in arbitrarily unfair merge orders. We will then reason that, when vehicles are allowed to exchange information, fairness is achievable. We present a specific algorithm that ensures this property even when communication is restricted to unreliable single-hop beacons (e.g., via IEEE 802.11p) and under the constraint that only $1\,\%$ of the vehicles participate.

The main contributions of the work presented here are:

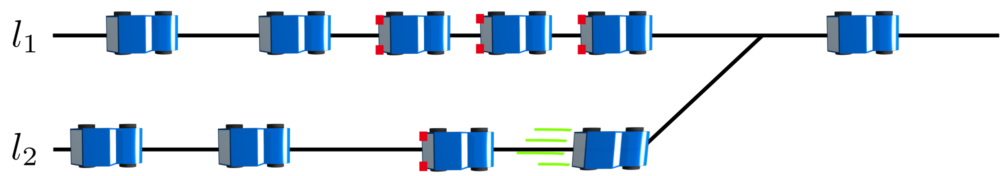

Figure 1. A sketch of a merging of two lanes.

1) The formal definition of a model describing the behavior of vehicles when two lanes merge.
2) The specification of a fairness criterion for this behavior, which we call *free-flow fairness*.
3) A distributed algorithm which results in good fairness under real-world conditions.
4) Simulation results analyzing the impact of the percentage of participating vehicles.

This paper is structured as follows. The next section discusses related work on lane merging. Section 3 describes the terms of the formal model and specifies a fairness criterion. The fairness of the zipper merge strategy is discussed in Section 4. In Section 5, we show how the vehicles should behave ideally, in order to reach a fair merge order. We then move to a real-world setting, where we drop unrealistic assumptions such as the omniscience of the involved vehicles. We propose a merging algorithm that coordinates the vehicles through unreliable beaconing. The algorithm has been implemented in the simulator ns-3 [7]; the simulation results are described in Section 6. Our findings on communication scheme design for merging scenarios are concluded in Section 7.

2 Related Work

Related work on inter-vehicle communication for on-ramp traffic control can be grouped in three categories: (1) enhancing efficiency and safety through communication, (2) adapting or evaluating communication algorithms and radio channel properties in the context of lane merging support, and (3) experiments with cars or robots that perform merging. We give examples for each of those three categories in the following.

Work on efficiency and safety focusses on metrics like disturbances in traffic flows [5] and the time cars need to drive to an intersection [8] but the merge order is not a matter in the discussions. The queue lengths upstream of a merge with and without communication are evaluated by Xu et al. [9] and the performance of cooperative autonomous cruise control (C-ACC) systems compared to non-cooperative ACC systems is discussed by Xu and Sengupta [6]. Criteria for robust merging, sets of different merging strategies, and algorithms for communicating cars are proposed by Wang, Kulik, and Ramamohanarao [10]. Further research is done on the coordination at intersections by supporting traffic light switching through communication [11]. A key term here are virtual traffic lights which are only visible to communicating cars: they help to increase intersection capacity [12], [13]. Another approach presented in a paper by Morla envisions a slot-based road usage where slots are maintained by cooperating vehicles [14]. This idea is exemplified with a road merging scenario. The paper's high-level discussion does not include details of the merge order or the communication scheme, and it is suggested that the cooperation should fail if the inbound flows are high enough that jams will emerge. In contrast, we argue that a merging coordination scheme is particularly relevant in face of a traffic jam.

The second category of related work is about communication algorithm enhancements and radio channel properties. The impact of the antenna radiation pattern with regard to the line of sight in merging scenarios is discussed by Abbas et al. [15]. Uno, Sakaguchi, and Tsugawa propose a merging control algorithm that uses communcation for reserving merging space with virtual vehicles [16]. The authors evaluate the algorithm's communication delays using simulations. Wolterink, Heijenk, and Karagiannis analyze a geocast algorithm in a merging scenario and suggest to predict future positions of vehicles for enhancing the algorithm's performance in dense traffic [17].

The third category evaluates merging behavior by means of experiments. Sakaguchi, Uno, and Tsagawa conducted experiments to show the feasibility of their merging algorithm with autonomous mobile robots [18]. Experiments with vehicles that cooperate using communication in an intersection are described by Kolodko and Vlacic [19]. A more recent experimental study, part of the AUTOPIA project [20], showed that coordinated merging at an on-ramp is possible in a real setting with cars exchanging information with a road-side unit [21]. The authors created an automated merging system with the goal of fluidly merging to a congested main road and deployed it in three production vehicles.

In contrast to the previously described works, the paper at hand proposes fairness as a further goal for merging assistant applications. We introduce a formal model in order to understand and describe the meaning of "perfect" fairness in absolute terms. A decentralized algorithm is developed and evaluated for how close we can get to perfect fairness in simulations with realistic settings. To the author's best knowledge, there is no other approach that improves fairness in a merging with decentralized communication.

3 Merging Order Fairness

In order to analyze the behavior of vehicles when two lanes merge into one, we use a formal model. The focus of this paper is on the merging order. We therefore concentrate on this aspect in the model, and otherwise keep it as simple as possible. We consider a topology with a main lane l_1 and an on-ramp lane l_2 which merges into the main lane.

The on-ramp lane l_2 ends at a point which we call the merge point. Both lanes, as well as cars driving in the lanes, are one-dimensional. Figure 1 depicts the scenario. The longitudinal movement of the cars is restricted by a maximum speed common to all cars and a minimum speed of zero. A further simplification is that all cars have the same acceleration capabilities and the same length. To keep track of the initial order of the cars before they pass the merge point, we consider them to be totally ordered in a lane by their time of appearance.

Car c_i (for $i \in \mathbb{N}$) appears in one of the lanes at t_i^0. Only one car appears at the same time in the same lane.

The cars pass the merge point in a certain order. Each car c_i therefore has a position index $k_i \in \mathbb{N}$ in the sequence of vehicles leaving the merge. The merging scheme that is applied determines the position indices for all cars in a given traffic scenario. Our fairness criterion will refer to the order of cars leaving the merge, as given by the sequence k_i.

3.1 Free-Flow Fairness

Our objective is to enable a fair merging order. In order to reach this goal we need a better understanding of the term “fairness” in this context. Consider two cars that approach the merge point and have to decide which one drives first. An intuitive approach would be to base that decision on the cars’ distances to the merge. This is not a good solution, though: depending on the length of the queue on each line before the merge point, two cars may be similarly far from that point, while one of them has been waiting much longer than the other. It would clearly be unfair to let a car with shorter waiting time pass first. The decision should therefore be based on the time that the cars spend waiting: a car with longer waiting time should be given preference.

But how can the waiting time be measured? Or, more specifically, when does a car start waiting? We argue that the waiting time of the car should be measured starting from the point in time when the car would have arrived at the merge point if it was not hindered by any other car. This is the earliest point in time at which a given car could possibly arrive at the merge point; we term this point in time the car’s *free-flow arrival time*.

A merge order based on the free-flow arrival time lets a car pass earlier if it has the earlier free-flow arrival time. Our definition of fairness is based on this concept:

Definition 1: Free-flow fairness. Let c_1, c_2 be two arbitrary cars with free-flow arrival times of $\tilde{t}_1$ and $\tilde{t}_2$ and merging positions of k_1 and k_2, respectively. Without loss of generality let $\tilde{t}_1 < \tilde{t}_2$. The merge order is called fair if and only if it holds that $k_1 < k_2$ for any such pair of vehicles.

Free-flow fairness can be seen as a form of first-come-first-served fairness, which is also referred to as temporal fairness [22]. Such fairness definitions fit well if a resource is used for the same amount of time by each consumer. We consider this a reasonable approximation in our case: the time it takes a vehicle to traverse the merge will vary only slightly with the vehicle type.

3.2 Measuring Unfairness

Based on the notion of free-flow fairness, the unfairness of a specific merge order in a given traffic scenario can be measured. To this end, we consider the *position difference*:

the difference between a car's true merging position k_i and the position $\tilde{k}_i \in \mathbb{N}$ the car would have with a merge order given by free-flow fairness, that is, $k_i - \tilde{k}_i$. A vehicle's position difference expresses by how many vehicles it merged too early or too late. For a set G of $n = |G|$ cars, the total *unfairness* within this group of cars can be defined as

$$u := \sum_{i \in G} \left(k_i - \tilde{k}_i \right)^2 ,$$

the average unfairness $\overline{u}$ is given by

$$\overline{u} := \sqrt{u/n}.$$

We use squared differences in order to avoid problems with varying signs in the differences (cars merging too early vs. cars merging too late), and to penalize large deviations from the fair order more strongly. The squaring allows minimizing the deviations from the fair order in the spirit of a least squares optimization.

Let us for a moment assume that the merge has been unused for a time span because the queues at both incoming lanes were empty. Observe that in this case, no position interchanges between cars merging before this "gap" and cars merging afterwards can have occurred, according to the definition of free-flow fairness. Otherwise, one car would have arrived at the merge point before the gap, but merged after it, which contradicts the assumption that both queues were empty.

This constraint on permissible permutations simplifies certain computations: in particular, if such gaps occur, we can consider the groups of cars between any pair of gaps independently. Note that the unfairness of consecutive groups according to the above definition is independent and additive.

4 Fairness of the Zipper Merge

In this section, we will investigate how well classical zipper merge performs with respect to free-flow fairness as defined above. In particular, we are interested in the long-term behavior of the average unfairness $\overline{u}$.

To this end, we consider the traffic flows in the incoming lanes. We denote the average traffic flows in lanes l_1 and l_2 by q_1 and q_2, respectively. The traffic flow in a lane is limited by the cars' lengths, their maximum speed, and the safety distances between

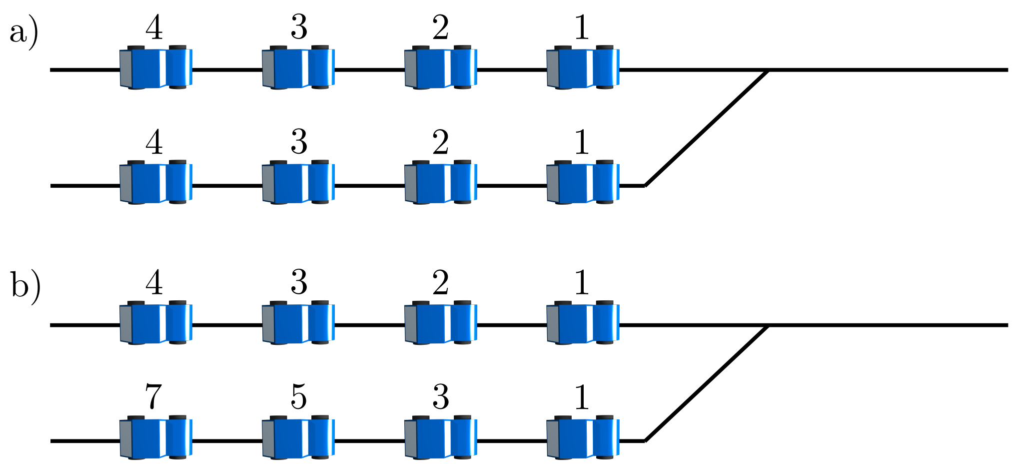

Figure 2. Two situations with different flows. The cars' free-flow arrival times are shown above them. In a), the flows are equal, whereas in b), the flow of the lower lane is halved.

the cars. It is measured in cars per second. Q is the limiting flow at the merge, i. e., the maximum number of cars per second that is able to pass the merge point.

For a first impression on how we discuss unfairness, let us assume very small traffic flows in both incoming lanes. Assume that the cars are so sparsely distributed that they do not influence each other while merging. Evidently, cars will merge in the order given by their free-flow arrival times. The unfairness metric will therefore be zero.

Looking at fairness becomes more interesting when the density of the flows is larger, so that cars start to interact and some cars must wait in order to merge. To assess such situations analytically, we assume exponentially distributed interarrival times between the cars [23], [24]. We will distinguish two cases: the total incoming traffic flow in both lanes together may be either (1) larger than or (2) at most equal to the merge capacity. In general, we are primarily interested in cases where the incoming flows are both greater than zero and differ at least minimally (i. e., $q_1 \neq q_2$). Figure 2 shows one example with equal flows (part a of the figure) and one with non-equal flows. The cars' free-flow arrival times are shown above each car. In the first part, zipper merge enables a fair merge order, while the merge order of the second example will not be fair.

If $q_1 + q_2$ exceeds the merge capacity Q, the unfairness grows to infinity if zipper merge is used. This is easy to see: with zipper merge, the same number of cars from each incoming lane will merge per time unit, as long as cars ready to merge are available on both lanes. Because the total traffic flow exceeds the merge capacity Q, an increasing backlog will build up in at least one lane (the one with the higher traffic flow). In the other lane, a backlog may or may not build up, depending on whether the traffic flow there exceeds $Q/2$. Since traffic flows are assumed to be not equal, the backlog will at least be smaller, though, and the difference in length between the backlogs will become larger and larger over time. Cars arriving in the lane with lower traffic flow will therefore be able to pass more and more cars with earlier free-flow arrival times in the other lane. Consequently, position differences in the sense introduced above increase more and more, so that the average unfairness increases without limit.

The case where the total traffic flow in both incoming lanes does not exceed the capacity of the merge is slightly more difficult to understand. We will show that the long-term average unfairness gets higher and higher, the closer the total incoming flow $q = q_1 + q_2$ approaches the merge capacity Q. To obtain this result, we model the waiting in front of the merge point with an M/D/1 queue. This queueing model describes exponential interarrival times and a constant merging time for each car at the single merge.

The so-called utilization rate of the merge is $\rho = q/Q$. Standard results from queueing theory (see, e.g., [25]) yield that the expected number of cars waiting in front of the merge is

$$\rho^2/(2(1-\rho)).$$

This is a total number for both lanes. We see that at the limit of the utilization rate (i. e., for q close to Q) the expected number of waiting cars is infinite:

$$\lim_{\rho \to 1} \rho^2/(2(1-\rho)) = \infty.$$

Figure 3 depicts this.

For $\rho < 1$, the queues in front of the merge point will, from time to time, be both empty. The expected fraction of cars that see an empty queue is $1 - \rho$. Whenever this occurs, there is a gap between two groups of cars. As argued above, such groups can

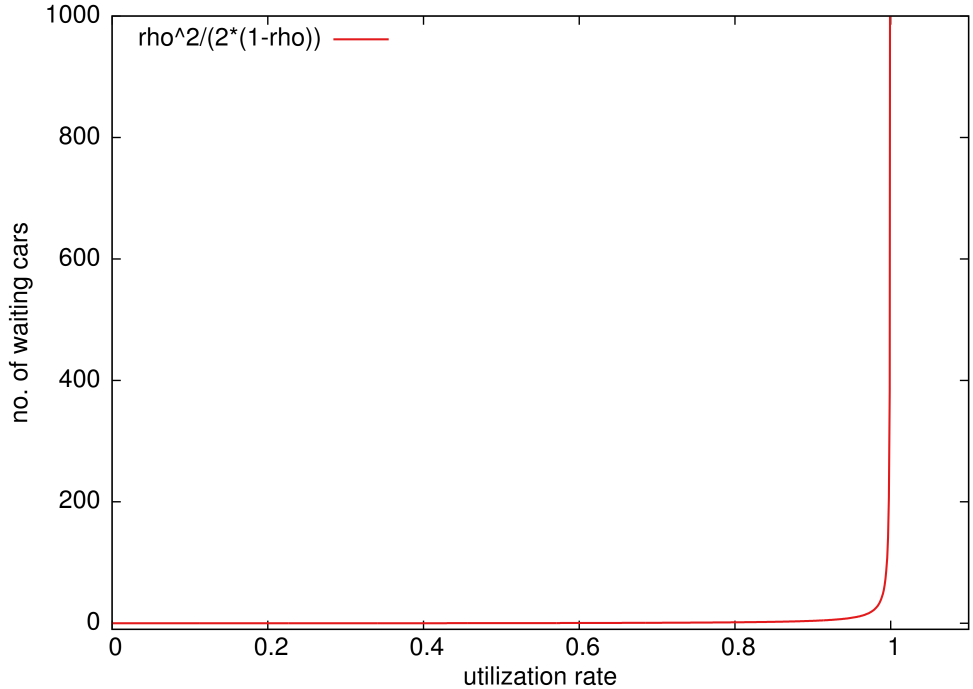

Figure 3. A plot of the number of waiting cars against the utilization rate.

be considered independently. The expected size for each of these groups is equal and finite. Since the first car of each such group will not be influenced by any other car, it merges at its free-flow arrival time. It therefore has a position difference of zero. The merge order of the other cars within the group will be a permutation of their fair merge order. For each other car of a group, the absolute value of the position difference will thus be smaller than the group size. Therefore, the average unfairness within each group is bounded above by the square of the group size. As a result, the long-term average unfairness is limited.

In general, the unfairness will be worse for higher incoming traffic flows and for higher differences between the two incoming flows. Moreover, the larger the difference between the traffic flows, the more quickly unfairness will build up.

To complement these analytical considerations, we simulated a zipper merge with different incoming flows in a simulation environment based on ns-3. We will present the details of this simulation in Section 6.

For exactly equal flows, zipper merge will be fair even over long time spans. However, equal flows are very unlikely to occur in practice—and, as dicussed above, small deviations lead to unfairness that grows arbitrarily over time. This raises the question how we can leverage communication between cars to improve the merging fairness—and which fraction of cars needs to participate in order to yield good results. We will now turn towards answering this question.

5 Achieving Fairness with C2C Communication

So, how should the vehicles act in order to yield good free-flow fairness? Evidently, optimal free-flow fairness will be achieved if each car merges at a position which matches the order of free-flow arrival times. If each car participated in the system (or, respectively, each driver) were willing to obey the system's instructions regarding the merge order, and if each car had perfect knowledge about all other cars and their free-flow arrival times, then perfect fairness could easily be achieved: whenever a car arrives at the front of one of the lanes leading to the merge point, it checks the free-flow arrival time of the front car in the other lane, and compares it to its own free-flow arrival time. The car with the earlier free-flow arrival time drives first. Since free-flow arrival times are, obviously, monotonically increasing within each lane, this behavior leads—in the spirit of the merge sort algorithm—to a perfectly ordered sequence of cars after the merge. In this setting, our unfairness metric is therefore zero.

In a real system, however, the above assumptions will not hold: not all cars will participate, not all drivers will always obey, and not all information will be known (due to unreliable wireless communication, limited communication range, etc.). One may therefore ask how much we lose in terms of fairness if non-equipped or non-cooperative vehicles jump the queue and/or merge too late. Throughout the rest of this paper, we will therefore drop the unrealistic assumptions one by one, and see how this changes the outcome. We will, in fact, show that this has only a minor impact on the achieved fairness. We now start with an introduction of the algorithm.

5.1 Algorithm for the merge order

The algorithm that needs to be implemented is based on a list of car identifiers. The list is sorted ascendingly by the cars' free-flow arrival times and such a list is maintained at each participating car. A map points from the list's entries to arrival times. The own car's information is initially stored in the data structures. New entries are learned by other cars' beacons. Beacons contain a sender's identifier, position in the lane, and free-flow arrival time. The free-flow arrival time is calculated at each car before its speed is influenced by other cars that brake in front of the merge. It is calculated as the time to accelerate to maximum speed and travel with this speed until the merge is reached.

An implementation of the function that is called on receiving a beacon is shown in pseudocode by Algorithm 1. The functions called in lines 2 to 4 extract the sender's

Algorithm 1 An implementation of the algorithm in pseudocode.

Require: Position $MergePoint$

1: **function** ONRECEIVEBEACON(Beacon)
2:     Identifier $id \leftarrow GetId$(Beacon)
3:     Position $x \leftarrow GetPosition$(Beacon)
4:     Time $t \leftarrow GetArrivalTime$(Beacon)
5:     **if** $x \leq MergePoint$ **then**
6:         $AddToList(id, t)$
7:     **else**
8:         $RemoveFromList(id, t)$
9:     **end if**
10: **end function**

identifier, position in the lane, and free-flow arrival time from the beacon. It is checked in line 5 if the sender is approaching the merge point. In this case, the identifier and free-flow arrival time are added to the sorted list and the map. Otherwise, the information is removed from the data structures.

The only necessary influence on the mobility model is to make the participant wait if it is not on first place of its sorted list. The merging process itself is performed by the merging rules of the underlying mobility model. That way, our proposed algorithm does not cause dangerous situations because of communication failures.

5.2 Fairness with non-participants

We will first consider the case where not all cars participate in the system and/or not all drivers follow its instructions. Observe that, for our purposes, these two are equivalent: a driver not willing to obey the guidance by the system may merge either too early or too late—just like a driver who does not receive instructions from the system. We now investigate the case of limited system participation analytically. We will prove that even with only a small fraction of participating, cooperating cars fairness is achievable. Subsequently, in the next section, we will use simulations to also investigate the case where unreliable beaconing is used and therefore perfect knowledge is no longer available.

Consider two arbitrary (participating or non-participating) cars A and B. Assume without loss of generality that A has an earlier free-flow arrival time than B, i.e., A should merge first. If A and B are in the same lane, they will merge in correct order because A has the earlier free-flow arrival time and must therefore be ahead of B. If both A and B participate in the system, they will merge in the correct order, regardless of their lanes. If, however, A and B are in different lanes and at least one of them does not participate, then it is possible that they switch order, i.e., that B merges earlier than A.

Observe, though, that A and B may no longer switch order if there are cars which participate in the system and which "force" B to wait until after A has merged. More specifically, assume that there is a participating car X in A's lane behind A, and a participating car Y in B's lane before B, such that X has an earlier free-flow arrival time than Y. Figure 4 shows this scenario. Then, B cannot merge before Y, Y will not merge before X, and X cannot merge before A; therefore, A and B are forced to merge in the correct order. The more cars enter the system with free-flow arrival times in between A and B, the more likely it becomes that participating vehicles X and Y exist which fulfill the above conditions. Consequently, grossly unfair situations, where cars switch order that are "far apart" with respect to the global fair merge order, are very

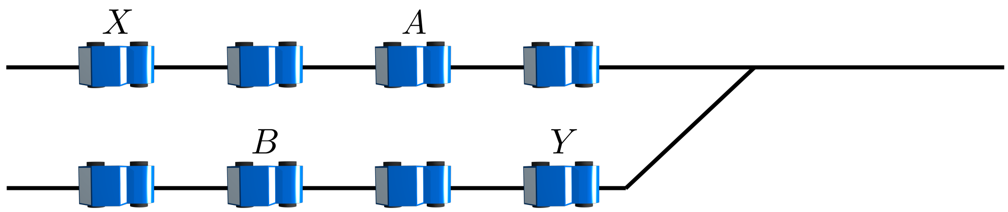

Figure 4. The influence of participants on non-participants. Cars A and B are non-participants and X and Y are participants. The fair merge order is starting with A, then X, Y, and finally B.

unlikely. Along the lines of these considerations, we now prove that high unfairness becomes very unlikely and that the expected unfairness is finite.

To this end, assume that a fraction r (with $0 < r < 1$) of cars participates in the system and obeys its directions. Furthermore, let $\tilde{k}_A$ and $\tilde{k}_B$ be the merge positions of A and B, respectively, according to the globally fair merge order. Consequently, there are

$$d = \tilde{k}_B - \tilde{k}_A - 1$$

cars with a free-flow arrival time between A and B.

We are now interested in the probability p that it is possible for A and B to merge in the wrong order, depending on the number of cars d between A and B. Those cars may be arbitrarily distributed to lanes l_1 and l_2. If we model the traffic flow like in Section 4 and assume traffic flows of q_1 and q_2 in the two lanes, we can derive an upper bound for p. Assume without loss of generality that A's lane is l_1 with average inflow q_1.

Recall that it is not possible for A and B to switch order if participating cars with the roles of X and Y as described above exist. Let $\delta = \lfloor d/2 \rfloor$. The probability that these cars exist is then bounded below by the probability that there is

1) at least one out of the first δ cars between A and B that is in A's lane l_1 and participates in the system (cf. X) and
2) at least one out of the last δ cars between A and B that is in B's lane l_2 and participates in the system (cf. Y).

These two events are stochastically independent. Their probabilities are given by

$$p_X = 1 - \left(1 - \frac{q_1}{q_1 + q_2} \cdot r\right)^{\delta}$$

and

$$p_Y = 1 - \left(1 - \frac{q_2}{q_1 + q_2} \cdot r\right)^{\delta},$$

respectively. If both conditions hold, then A and B cannot possibly merge in the wrong order. As a result, the probability p is bounded above as follows:

$$p \leq 1 - p_X \cdot p_Y.$$

With $q_{\min} := \min\{q_1, q_2\}$ it is easy to derive that

$$p \leq 2 \cdot \left(1 - \frac{q_{\min}}{q_1 + q_2} \cdot r\right)^{\delta} = 2 \cdot \left(1 - \frac{q_{\min}}{q_1 + q_2} \cdot r\right)^{\lfloor d/2 \rfloor}.$$

Therefore, the probability that it is possible for A and B to switch order decreases exponentially with increasing "distance" d between them.

Now, what does this mean for the expected (un)fairness if a large number of vehicles merge? The total order in which cars actually merge is a permutation of the fair merge order. Assume that car c_i merges x positions too early (or too late) with respect to the totally fair merge order. Then, c_i's contribution to the total unfairness is x^2. In this situation, there must be a vehicle c_j which merges in the wrong order with respect to c_i and for which $|\tilde{k}_i - \tilde{k}_j| \geq x$; this follows from basic mathematical properties of permutations. The number of cars between c_i and c_j in the totally fair merge order is

$$d = |\tilde{k}_i - \tilde{k}_j| - 1 \geq x - 1.$$

By the arguments above, the probability that c_i and c_j were able to merge in wrong order is bounded above by

$$\tilde{p}_x = 2 \cdot \left(1 - \frac{q_{\min}}{q_1 + q_2} \cdot r\right)^{\lfloor (x-1)/2 \rfloor}.$$

This is the probability to switch places for the case that c_j is x positions away. Before switching with c_j, c_i needs to switch with all vehicles in the other lane up to a distance of $x - 1$ positions. If c_j is further away, the probability to switch is lower because the number of vehicles c_i has to switch with is larger. So this equation bounds the probability for c_i to switch positions with any vehicle c_j that is x or more positions away regarding the fair order. To obtain an (admittedly coarse, yet sufficient) bound for the probability to be off by x positions *either* too early *or* too late, we can double this expression:

$$p_x \leq 2 \cdot \tilde{p}_x = 4 \cdot \left(1 - \frac{q_{\min}}{q_1 + q_2} \cdot r\right)^{\lfloor (x-1)/2 \rfloor}.$$

As a result, the expected contribution of c_i to the total unfairness is bounded above by

$$E[u_i] \leq \sum_{x=1}^{\infty} x^2 \cdot p_x.$$

As can be seen by d'Alambert's ratio test, this series converges, i.e., there is a finite upper bound. Note that this upper bound for the unfairness contribution does neither depend on properties of the specific car nor on the total number of cars. Since the expected unfairness contribution of any car is bounded above by the same finite value, the expected average unfairness with our merging scheme is also finite—even for small fractions r of participating vehicles and for arbitrarily unbalanced flows q_1 and q_2.

For zipper merge, as discussed in the previous section, the unfairness grows without bounds even for slightly unbalanced inflows in the two lanes. When merging is supported by inter-vehicle communication in the way outlined here, in contrast, all vehicles—including non-participating ones—are kept from merging at grossly unfair positions. The resulting unfairness is finite even when only a small fraction r of cars actively participates in the system.

We consider this a very interesting trait of the mechanisms proposed here: they lay the foundation for one of the very few applications of inter-vehicle communication where substantial benefit results from even just a small technology penetration ratio. In the following section, we will show that this trait does not only exist in the asymptotic limits as considered in the preceding section and here, but can likewise be shown in realistic simulation scenarios.

6 Simulation and Evaluation

We implemented our merging algorithm in the network simulator ns-3 and evaluated the influence of the incoming flows and the number of participants on the resulting unfairness. Before presenting the results, the setup of the simulation environment is explained, including the employed car-following model to support merging operations.

6.1 Simulation setup

The communication scheme and the car acceleration behavior for coordinating the merging of two lanes have been evaluated using the network simulator ns in version 3.13 and the vehicle movement model of the Intelligent Driver Model (IDM [26]). For the latter, we used the implementation of the Highway Mobility 2.0 framework [27] that enhances ns-3 with a model for highways, cars, and commonly used car-following models.

We created two different simulation setups. The first investigates the impact of the traffic flows, it keeps the number of participating cars at a constant low fraction of the total population. The second shows the impact of the rate of participating cars by keeping the traffic flows constant and varying the number of participants. Both setups use the same car-following model. Cars, if equipped, communicate via an unreliable channel. The road network consists of two parallel lanes with the same length and a third lane that represents the road after the merge. The maximum speed in all lanes is $36\,\mathrm{m/s}$ which equals $129.6\,\mathrm{km/h}$. The interarrival times of the cars are exponentially distributed and the probability for a car to become a participant of the coordination system is independently randomized.

For communication, we implemented vehicle-to-vehicle beaconing with randomized sending intervals: after sending a beacon, the next one is sent after a timespan that is uniformly random between $1\,s$ and $2\,s$. The random intervals avoid synchronization effects [28]. The sending rate is not a major parameter for the fairness in our simulations: the main task of the beaconing algorithm is to distribute each car's free-flow arrival time just once to other cars, since this does not change over time. We used the 802.11a channel and network device implementation of ns-3. The parameters are chosen to match the 802.11p standard for sending in the control channel, e.g., we chose OFDM at a rate of $6\,\mathrm{Mb/s}$ and a bandwidth of $10\,\mathrm{MHz}$. The transmit power is $33\,\mathrm{dBm}$. A constant speed propagation delay is added to the channel, as well as a loss model using Friis with Nakagami-m propagation. The sensitivity thresholds are based on Atheros 802.11a chipsets and the 802.11a standard: the energy detection threshold is set to $-71\,\mathrm{dBm}$ and

Table 1

Communication parameters of the simulations. The string values describe classes and constants of ns-3.

WifiHelper	WIFI_PHY_STANDARD_80211a
Data mode	OfdmRate6MbpsBW10MHz
MAC type	AdhocWifiMac
Delay model	ConstantSpeedPropagationDelayModel
Loss model	NakagamiPropagationLossModel
TxPowerStart	33
TxPowerEnd	33
TxPowerLevels	1
RxGain	0
TxGain	0
EnergyDetectionThreshold	-71 dBm
CcaMode1Threshold	-91 dBm

the CCA mode 1 threshold to -91 dBm [29]. Table 1 lists the communication parameters.

The simulation is simplified to let participants send beacons within a distance of 1 km to the merge point. This is not a necessary parameter for the algorithm to work but reduces communication in the lanes far from the merge. This distance was chosen such that informing other cars takes place far enough from the merge point where a reaction regarding the merging behavior is possible. The participants merge in the order determined by their free-flow arrival times if they know about each other (that is, if beacons are successfully received). A participant learns via beaconing when a participant from the other lane traverses the merge point. Packet losses may cause a participant to wait at the merge point although all participants to wait for have already merged. A timeout avoids starvation in such situations. It is calculated dynamically: if a car receives the status of a participant that has to merge first, it is estimated how long it will take the sender until it traverses the merge point in a congested setting. Cars that do not participate use zipper merge.

Table 2
IDM car behavior parameters.

Desired speed	$36\,\mathrm{m/s}$
Acceleration	$3.0\,\mathrm{m/s^2}$
Deceleration	$3.0\,\mathrm{m/s^2}$
Minimum gap	$2.0\,\mathrm{m}$
Time headway	$1.5\,\mathrm{s}$
Delta exponent	4.0
Vehicle length	$4\,\mathrm{m}$
Simulation step	$1\,\mathrm{s}$

6.1.1 Car-following algorithm

To simulate car movement, we used the Intelligent Driver Model, a car-following model for a single lane that calculates a car's acceleration on the microscopic level, i.e., based on individual cars. The calculation uses knowledge about the distance to the car ahead and the speed difference to that car. IDM is a lightweight model that suffices for our purpose of simulating cars with realistic acceleration rates and safety distances. Several studies show its suitability to these demands [30], [31]. The model parameters were chosen conservatively to create reasonably realistic car following. The desired velocity equals the maximum speed of $36\,\mathrm{m/s}$. The maximum acceleration and deceleration are set to $3.0\,\mathrm{m/s^2}$. The minimum gap between cars is $2.0\,\mathrm{m}$, the time headway is $1.5\,\mathrm{s}$, the delta exponent is 4.0, and the length of vehicles is $4\,\mathrm{m}$. The simulation steps at which cars recalculate their accelerations is set to $1\,\mathrm{s}$. Table 2 lists the IDM parameters.

The merging behavior was also implemented using IDM. Since IDM is designed to follow a single car driving directly ahead, it is used twice for merging: for the car that is ahead in the own lane as well as for the car that has to be ahead after merging. When a car c_i has decided to merge behind a car c_j from the other lane, it calculates an acceleration value for c_j in addition to that for the car directly ahead of c_i. From both values, the minimum is chosen to ensure safe following. If c_j is far upstream of the merge point, c_i treats it as if it had stopped in the merge point. The acceleration value calculated by c_i in view of c_j then causes c_i to wait in front of the merge point. After c_j merged, its true speed and location are used for calculation. If c_j is in the proximity of the merge point, its speed value is gradually increased to enable a smooth transition between the two states.

6.1.2 Varying incoming flows

In the simulations with varying incoming flows, multiple flow levels were simulated for $11 \cdot 10^3$ s of simulation time. The lowest flow in lane l_1 was 0.025 vehicles per second and it was increased in steps of 0.025 vehicles/s up to 0.225 vehicles/s. The mean flow in lane l_2 was always twice that of l_1's flow; that is, $q_1/q_2 = 0.5$. Each parameter set was simulated 100 times with different random seeds. The lanes were long enough to avoid jam effects at the insertion point of cars (which might otherwise cause undesirable simulation artifacts, like lower initial speeds or smaller flows when inserting cars).

6.1.3 Varying participation ratios

The simulations with varying ratios of participating cars were run with flows of 0.15 vehicles/s and 0.3 vehicles/s in lanes l_1 and l_2, respectively, and the average participation ratio was varied. The flow sizes were chosen to cause a jam upstream of the merge point. Each setup was run for 100 times, each until 3000 cars merged. The lanes were chosen long enough to avoid a jam at the insertion point.

6.2 Evaluation

6.2.1 Different incoming flows

Figure 5 shows the average unfairness in the simulations, plotted over the sum of incoming flows, i.e., vehicles/s, on the x axis. We ran the simulations with no participants, i.e., all cars do a standard zipper merge, and with a participation ratio of $1\,\%$. The boxes show the 10th percentile, the median, and the 90th percentile. The whiskerbars describe the minimum and maximum values. The results of runs with $0\,\%$ participating cars for flows larger than 0.45 vehicles/s are not plotted; the results are far beyond the bounds of the figure. The resulting unfairness with these parameters was merely limited by the fixed simulation duration and the consequently finite number of cars.

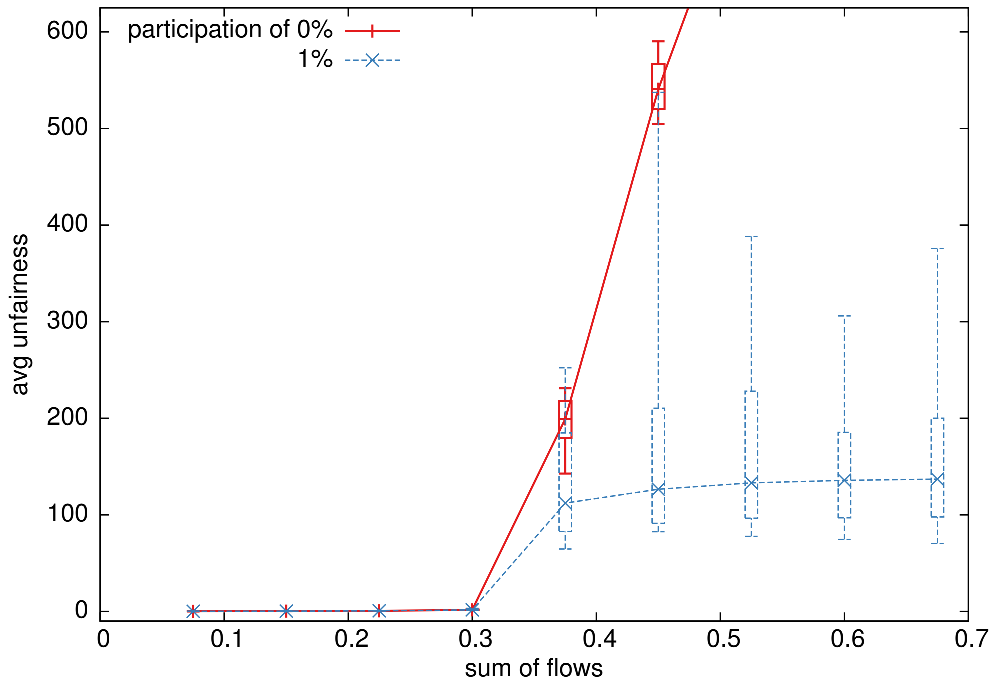

Figure 5. Average unfairness after long (but finite) runs without participation and with $1\,\%$ participation over different sums of flows in the incoming lanes.

These results confirm what we expected: with a zipper merge, the average unfairness grows without bound. In contrast, runs with $1\,\%$ participation converge to a finite level of unfairness. The absolute position differences have also been calculated to examine how many positions a car is away from its fair place. In the simulations with $1\,\%$ participation and higher flows than the merge capacity, the average absolute position difference is about 100 cars.

6.2.2 Different participation ratios

Figure 6 shows the average unfairness with a logarithmic scale on the y axis. For each run, the results of the first 3000 cars traversing the merge are shown in steps of 100 mergings. Again, the boxes show the 10th percentile, the median, and the 90th percentile, while the whiskerbars describe the minimum and maximum values. The unfairness of the runs with $100\,\%$ participants is constantly zero and is thus not visible here. Cars begin to build up a jam quickly after a few mergings. The average unfairness then

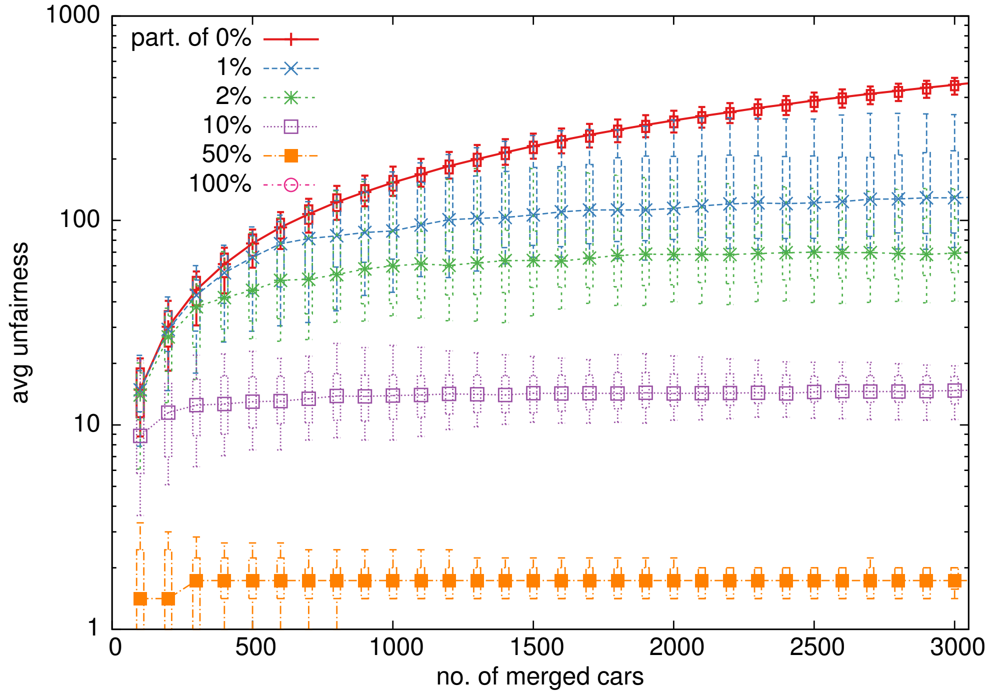

Figure 6. Average unfairness on y for a car merging at position x with different participation ratios, log scale.

grows monotonically with each merging car unless there are participants that meet at the merge point. For all runs with participants, the average unfairness reaches a steady state. Higher participation ratios stabilize at lower states of average unfairness.

Figure 7 shows how position differences evolve over time from an initially fair state.

Single runs with varying participation ratios were chosen. The plot depicts the position differences on the y axis over the cars' true merging positions on the x axis. I.e., a point at x coordinate 600 and y coordinate 200 means that the car that should have merged as the 600th car has merged 200 positions too early or too late. Only cars from lane l_1 are shown and the position differences are neither squared nor averaged. The plotted range is limited to the first 1500 mergers. The figure shows that the differences return to a fairer state from time to time which is due to the meeting of participants in different lanes. However, it can be noticed that perfect fairness is not always maintained.

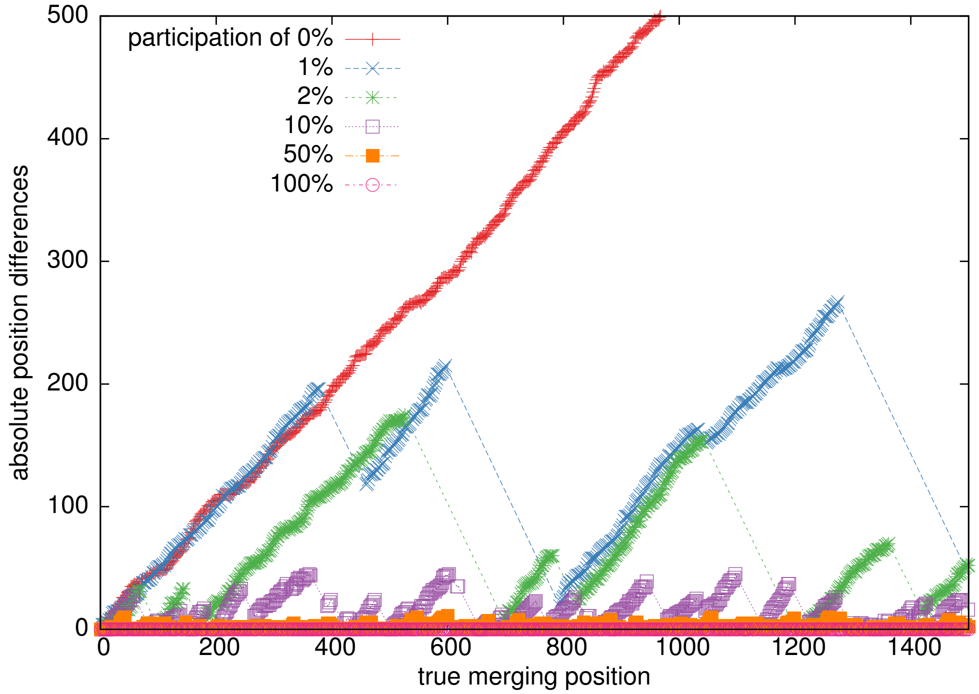

Figure 7. Position differences of cars coming from lane l_1.

This is because participating vehicles can only ensure a perfectly fair merge order among themselves. Position changes may still occur if non-participating vehicles are involved.

7 Conclusion

We have proposed a distributed algorithm for enhancing the fairness at the merging of two lanes. Our analytical model of the setting results that the algorithm enables a fairer merge order than the zipper merge in congested situations. It only uses periodic broadcasts exchanged with car-to-car communication. The algorithm has been implemented in ns-3 and simulated with the IDM car-following model and unreliable communication. The simulations show that the algorithm enhances fairness already at very low ratios of participating vehicles. These results are very promising and encourages future work: the good performance of the algorithm motivates to test it in real-world experiments. Besides experiments, the algorithm can be extended in future work in several directions. The described scenario is simple as it only consists of a single merge. More complex road networks are easy to imagine with more lanes and multiple merge points. It is important to understand how the algorithm influences fairness there. Another direction for research is to evaluate how the algorithm can preserve privacy with anonymized communication.

8 Acknowledgments

Funding for this work has been provided by the state of North Rhine-Westphalia and the European Union within the NRW-EU Ziel 2 program 2007-2013 (EFRE).

References

[1] D. Helbing, M. Treiber, A. Kesting, and M. Schönhof, "Theoretical vs. empirical classification and prediction of congested traffic states," *Eur. Phy. J. B*, vol. 69, no. 4, pp. 583–598, 2009.

[2] B. S. Kerner, *Introduction to Modern Traffic Flow Theory and Control: The Long Road to Three-Phase Traffic Theory*. Berlin: Springer, 2009.

[3] M. Papageorgiou and A. Kotsialos, "Freeway ramp metering: an overview," in *ITSC '00*, vol. 3, no. 4, Oct. 2000, pp. 271–281.

[4] B. S. Kerner, "Control of spatiotemporal congested traffic patterns at highway bottlenecks," in *ITSC '05*, vol. 8, no. 2, Sep. 2005, pp. 308–320.

[5] A. Kanaris, E. Kosmatopoulos, and P. Loannou, "Strategies and spacing requirements for lane changing and merging in automated highway systems," *IEEE Trans. on Veh. Technol.*, vol. 50, no. 6, pp. 1568–1581, Nov. 2001.

[6] Q. Xu and R. Sengupta, "Simulation, analysis, and comparison of ACC and CACC in highway merging control," in *IV '03*, Jun. 2003, pp. 237–242.

[7] NSNAM, "ns-3: network simulator 3." [Online]. Available: http://www.nsnam.org

[8] G. Raravi, V. Shingde, K. Ramamritham, and J. Bharadia, "Merge algorithms for intelligent vehicles," in *Next Gener. Des. and Verif. Methodol. for Distrib. Embed. Control Syst.*, S. Ramesh and P. Sampath, Eds. Springer Netherlands, 2007, pp. 51–65.

[9] Q. Xu, K. Hedrick, R. Sengupta, and J. VanderWerf, "Effects of vehicle-vehicle/roadside-vehicle communication on adaptive cruise controlled highway systems," in *VTC '02-Fall*, Sep. 2002, pp. 1249–1253.

[10] Z. Wang, L. Kulik, and K. Ramamohanarao, "Robust traffic merging strategies for sensor-enabled cars using time geography," in *GIS '09*, Nov. 2009, pp. 362–371.

[11] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode, "Adaptive traffic lights using car-to-car communication," in *VTC '07-Spring*, Apr. 2007, pp. 21–25.

[12] C. Avin, M. Borokhovich, Y. Haddad, and Z. Lotker, "Optimal virtual traffic light placement," in *FOMC '12*, Jul. 2012, pp. 6:1–6:10.

[13] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, and O. K. Tonguz, "Self-organized traffic control," in *VANET '10*, Sep. 2010, pp. 85–90.

[14] R. Morla, "Sentient future competition: Vision of congestion-free road traffic and cooperating objects," in *EWSN '06*, Feb. 2006.

[15] T. Abbas, L. Bernadó, A. Thiel, C. F. Mecklenbräuker, and F. Tufvesson, "Radio channel properties for vehicular communication: Merging lanes versus urban intersections," *IEEE Veh. Technol. Mag.*, vol. PP, no. 99, 2013.

[16] A. Uno, T. Sakaguchi, and S. Tsugawa, "A merging control algorithm based on inter-vehicle communication," in *ITSC '99*, Oct. 1999, pp. 783–787.

[17] W. K. Wolterink, G. J. Heijenk, and G. Karagiannis, "Constrained geocast to support cooperative adaptive cruise control (cacc) merging." in *VNC '10*, Dec. 2010, pp. 41–48.

[18] T. Sakaguchi, A. Uno, and S. Tsugawa, "Inter-vehicle communications for merging control," in *IVEC '99*, vol. 1, Sep. 1999, pp. 365–370.

[19] J. Kolodko and L. Vlacic, "Cooperative autonomous driving at the intelligent control systems laboratory," *IEEE Intell. Syst.*, vol. 18, no. 4, pp. 8–11, Jul. 2003.

[20] V. Milanés, E. Onieva, J. Pérez, J. Villagrá, J. Godoy, J. Alonso, C. González, T. de Pedro, and R. García, "Autopia program advances: How to automate the traffic?" in *EUROCAST '11*, vol. 2, Feb. 2011, pp. 374–381.

[21] V. Milanés, J. Godoy, J. Villagrá, and J. Pérez, "Automated on-ramp merging system for congested traffic situations," in *ITSC '11*, Jun. 2011, pp. 500–508.

[22] A. Wierman, "Fairness and scheduling in single server queues," *Surv. in Oper. Res. and Manag. Sci.*, vol. 16, no. 1, pp. 39–48, Jan. 2011.

[23] A. J. Miller, "A queueing model for road traffic flow," *J. R. Stat. Soc. Ser. B (Statistical Methodology)*, vol. 23, no. 1, pp. 64 –90, 1961.

[24] M. Rudack, M. Meincke, and M. Lott, "On the dynamics of ad hoc networks for inter vehicle communications (IVC)," in *ICWN '02*, Jun. 2002.

[25] R. B. Cooper, *Introduction to queueing theory*, 2nd ed. London : Arnold, 1981, ISBN: 0-7131-3435-6.

[26] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000.

[27] H. Arbabi and M. C. Weigle, "Highway mobility and vehicular ad-hoc networks in ns-3," in *WSC '10*. Society for Computer Simulation International, Dec. 2010.

[28] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *MobiCom '00*, Aug. 2000, pp. 243–254.

[29] J. Lee, J. Ryu, S.-J. Lee, and T. T. Kwon, "Improved modeling of IEEE 802.11a PHY through fine-grained measurements," *Comp. Netw.*, vol. 54, no. 4, pp. 641–657, Mar. 2010.

[30] E. Brockfeld, R. Kühne, A. Skabardonis, and P. Wagner, "Toward benchmarking of microscopic traffic flow models," *Transp. Res. Rec.*, vol. 1852, no. 1, pp. 124–129, 2003.

[31] E. Brockfeld and P. Wagner, "Calibration and validation of microscopic traffic flow models," in *Traffic and Granular Flow '03*, S. P. Hoogendoorn, S. Luding, P. H. L. Bovy, M. Schreckenberg, and D. E. Wolf, Eds. Springer Berlin Heidelberg, 2005, pp. 67–72.