

# Over-the-Horizon Awareness for Advanced Driver Assistance Systems: the TrafficFilter and microSlotted 1-Persistence Flooding

Martijn van Eenennaam, Geert Heijenk, Georgios Karagiannis, Bart van Arem

**Abstract**—Vehicle-to-vehicle communications (V2V) is a promising technique for Advanced Driver Assistance Systems to increase traffic safety and efficiency. A proposed system is the Congestion Assistant, which supports drivers when approaching and driving in traffic congestion. Studies have shown great potential for such systems to reduce the impact of congestion, even at low penetration. However, these studies assumed complete and instantaneous knowledge regarding position and velocity of vehicles ahead.

This paper refines and analyses the TrafficFilter, designed to supply the required information to the Congestion Assistant. Vehicles collaboratively build a so-called TrafficMap, providing over-the-horizon awareness. To this end, an improvement to the Slotted 1-Persistence Flooding called microSlotted 1-Persistence Flooding is proposed and evaluated.

In a simulation study the TrafficFilter is found to be a viable system to build over-the-horizon awareness for future Advanced Driver Assistance Systems like the Congestion Assistant, without triggering the phenomenon known as Broadcast Storm.

**Index Terms**—vanet, v2v, flooding, intelligent transportation systems, in-network aggregation, advanced driver assistance systems, traffic information systems

## I. INTRODUCTION

The application of Advanced Driver Assistance Systems (ADAS) is an emerging trend in the automotive industry. Advances in technology enable faster, smaller and more versatile hard- and software systems while consumers demand safer and more efficient vehicles. Quite often an automated system can be used to perform computationally intensive tasks (e.g. route planning, navigation) and tasks which require great coordination (such as anti-lock braking system (ABS), and electronic stability programs (ESP)), which are beyond the capabilities of the human driver. Furthermore, automated systems may circumvent the relatively long reaction time of human beings, enabling swift (automated) response to events.

It is expected that vehicle-to-vehicle communication can lead to an improved traffic flow stability [1], because the ADAS can augment or take over some of the driver tasks.

Research on *Driver Support in Congestion* by van Driel [2] proposes a set of ADAS called the *Congestion Assistant*. The Congestion Assistant relies on over-the-horizon awareness to aid the driver in traversing traffic congestion on highways.

E.M. van Eenennaam, G. Heijenk and G. Karagiannis are with the Design and Analysis of Communication Systems group (DACS), University of Twente, The Netherlands. <http://www.utwente.nl/ewi/dacs/>  
E-mail: e.m.vaneennaam@ewi.utwente.nl

B. van Arem is with the Faculty of Civil Engineering & Geosciences, Delft University of Technology, The Netherlands

Subsystems are proposed to aid the driver and supply the driver with information. Systems like the Congestion Assistant rely on the rapid dissemination of position and velocity information over distances in the order of kilometers.

No system has been proposed in [2] to deliver this information, and generic solutions do not readily apply. We have proposed a communication system to meet the demands by ADAS on a conceptual level in [3]. This system, called the TrafficFilter, follows a multi-hop vehicular ad hoc network (VANET) approach. Information is passed upstream by means of V2V communications, against the flow of traffic, and is aggregated along the way. This method represents the over-the-horizon awareness in a structure called a TrafficMap. This is a speed profile which expresses the traffic flow speed at a certain location on the road in a highly compressed form. The TrafficMap is built by a distributed system called the TrafficFilter.

The main contributions of this paper are:

- 1) The TrafficFilter, a novel means to obtain over-the-horizon awareness in VANETs, is described in detail and an extensive evaluation is provided.
- 2) A modification to the slotted 1-Persistence Flooding [4] strategy is proposed and evaluated. This modification enables the use of an IEEE 802.11p MAC and physical layer and perform network-layer flooding in an efficient manner by exploiting standard MAC-layer scheduling properties, without causing the phenomenon known as broadcast storm.

An extension to the TrafficFilter to operate on multi-lane highways with junctions is presented in [5].

This paper is structured as follows. First, in Sec. II, the background and related work are presented. Next, the architecture of the system is discussed in Sec. III. Simulation studies are performed to evaluate the performance of the system, as presented in Sec. IV. The influence of mobility on the communication is evaluated, and the Slotted 1-Persistence flooding and the proposed improvement, the microSlotted 1-Persistence flooding, are compared. Furthermore, the quality of the communicated information with respect to error rate and latency is evaluated. Sec. V concludes this paper.

## II. BACKGROUND AND RELATED WORK

This section provides an introduction to the Congestion Assistant, and its information requirements. Next, a brief overview of related work is presented.

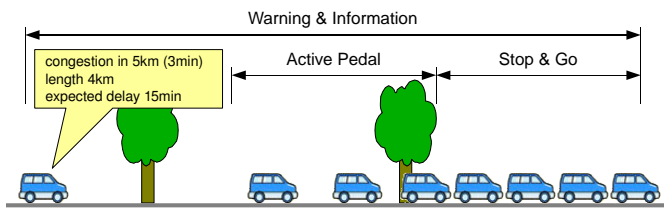


Fig. 1. A driver approaches a traffic jam

### A. The Congestion Assistant

The Congestion Assistant is an ADAS which helps drivers cope with traffic congestion on highways. It provides helpful assistance during tiresome repetitive driving tasks such as stop-and-go traffic and during potentially dangerous situations. It is expected that ADAS can have great impact on the severity of traffic congestion on highways [1], [2] by increasing the stability of the traffic flow.

In a user needs analysis study [2] it was found that drivers favour to be well-informed about upcoming traffic conditions. Based on the results of this study, a system was designed which performs three tasks; it informs, supports, and controls [2]. These tasks are present in the Warning & Information, Active Pedal, and Stop & Go functions respectively. These three functions are executed consecutively as a driver traverses a traffic jam, as illustrated in Fig. 1.

1) *Warning & Information (W&I)*: informs the driver of upcoming traffic conditions. This enables the driver to prepare for driving in the jam or choose an alternate route. Once in the jam, W&I will keep the driver updated on the situation and the progress.

In [2] there is no mention of a maximum acceptable delay or spatial-temporal accuracy of the information. The research presented in this paper provides insight into what is achievable. Further research will then have to determine if this is adequate for the Congestion Assistant to function properly in practice.

2) *Active Pedal (AP)*: applies counterpressure on the accelerator pedal starting from a certain distance prior to entering the tail of the congestion. The goal is to prevent the unfavourable behaviour of maintaining cruise speed until the tail of the jam is in sight, when suddenly very hard braking is required. This can result in accidents, but is also found to result in a high inflow of traffic, which causes the jam to grow at the tail. In microscopic traffic simulations [2] it was found that a more gradual reduction in speed reduces the inflow of traffic in the congestion because at a lower speed a closer following distance can be maintained.

3) *Stop & Go (S&G)*: engages when the vehicle enters the traffic congestion. S&G maintains a short following distance to the lead vehicle. Because no human reaction time is involved it can respond swiftly to changes in traffic flow speed. Communication is not restricted to the lead vehicle, but also includes those further down the road; as a result the system can compensate for the well-known shockwaves of braking traffic which propagate against the flow of traffic. When the vehicle leaves the traffic congestion, the S&G system disengages and manual driving recommences. An example of an implementation of the S&G functionality is the Cooperative

Adaptive Cruise Control (CACC) [6]. Note that for real-time vehicle control like CACC, the system does not use the information provided by the TrafficFilter, but rather operates on the short status messages transmitted on a periodic basis, known as beacons [7].

The Congestion Assistant was evaluated in a driver simulation study [2], and shows promising improvements in safety and efficiency. It is able to mitigate some of the unfavourable human behaviour that is part of the cause of traffic congestion. The observed driver behaviour was then simulated in a microscopic traffic analysis study for several degrees of market penetration. It was found that improvements to individual driver behaviour have positive effects on the overall traffic performance. At 10% penetration the efficiency already shows improvements, which increase as penetration goes towards 50% [2].

The general result is a more stable and homogenous traffic flow with a smaller standard deviation in speed. This is favourable because drivers sometimes tend to overreact or react too late, resulting in shockwaves of braking vehicles or—in the worst case—head-tail collisions. It will be clear that, with smaller deviations in speed, there is more time to react and less compensation is required.

### B. Related work

The Congestion Assistant needs to know the position and length of traffic congestion on the road ahead. For tens of years, radio news broadcasts in many countries have supplied traffic congestion information. This information is generally broadcast on an hourly basis and, due to airtime limitations, not always complete. This is both not timely and accurate enough for ADAS applications like the Congestion Assistant.

Traffic Message Channel - (TMC) is a service available via RDS (Radio Data System) on conventional radio broadcasts throughout Europe and North America. The information is typically digitally coded and can easily be integrated with navigation systems.

Both traditional radio broadcast and TMC rely on traditional means of detecting traffic jams (loop detectors, roadside cameras, traffic helicopters). For one, it is costly to instrument the entire road network with detectors and cameras and a helicopter only observes one area. This means that detection systems can notice a jam late, or not at all. Secondly, a centralised authority is used. This implies a long delay which is undesirable for a system which depends on knowing the location of a congestion with great temporal and geographic accuracy. Third, due to its shared-medium nature, a centralised system broadcasts information, most of which may not be of interest to a specific user.

The Congestion Assistant needs information of the road ahead, and only the road ahead. Note that this may be a unique view for every vehicle. A system to facilitate this could be based on Vehicle-to-Infrastructure (V2I) or Vehicle-to-Vehicle (V2V) communication. The recently standardised family of standards for Wireless Access in Vehicular Environments (WAVE) IEEE 1609 [8], [9] provides an architecture and a set of services and interfaces to support V2V and V2I

communication. IEEE 1609 has two important entities: the On Board Unit (OBU) and the Road Side Unit (RSU).

IEEE 1609 defines IEEE 802.11p for the MAC and physical layer. IEEE 802.11p has recently reached standardisation [10], but from the draft specification preliminary research was performed [4], [11], [12]. IEEE 802.11p is envisioned to be used for Dedicated Short Range Communication (DSRC) in the 5.9 GHz band. Application areas include safety and efficiency applications and toll collection.

WAVE defines a communication framework on which Intelligent Transportation Systems (ITS) can be built. Several platforms to distribute traffic information are defined in literature; for example, the CarTel [13], Trafficcopter [14] and SOTIS [15] systems.

CarTel's focus is on using a network of communicating vehicles as mobile sensor nodes which provide information to a centralised portal, where data is processed and from which it can be retrieved. The Trafficcopter system is a distributed ad hoc system which collects and distributes traffic information. Trafficcopter is a query-based system and as such retro-active. The Congestion Assistant requires pro-active information. The Self-Organizing Traffic Information System (SOTIS) provides such information, by means of a gossip-like exchange of traffic information. All nodes periodically transmit a small traffic report containing overheard information and own measurements. Over-the-Horizon information slowly trickles through the road network. SOTIS is based on UTRA TDD technology, and although it could be adapted for use with IEEE 802.11p, it was decided SOTIS' slow propagation speed (in the order of seconds per km) may be circumvented using a smart flooding scheme. Fusing Trafficcopter's idea of decentralisation and CarTel's application of roaming sensors inspired the design of the TrafficFilter.

Dissemination of information which may be of interest to every node in a network can be done using flooding. There are roughly three approaches to flooding: simple flooding (where every node repeats the message), sender-based forwarding decision (the sender tells a node to pass the message on), and receiver-based forwarding decision (a node decides for itself whether to pass information on or not).

simple flooding is known to cause a broadcast storm [16]; after reception of a frame, many nodes will simultaneously try to access the medium to perform a rebroadcast. The result is heavy contention and high collision probability. Broadcast suppression techniques can be used to minimise redundant transmissions while maintaining reachability. A sender-based scheme requires neighbour knowledge in order for a sender to designate a forwarder, and may not be the most desirable for a highly dynamic VANET. In a receiver-based scheme, a node decides whether to rebroadcast based on a certain policy. This can be probabilistic, counter-based, distance-based or neighbour-knowledge-based. Quite often a proposed scheme can use several elements, such as the Border-Aware [17] scheme which combines distance estimations with a counter in order to reach high coverage with a minimal number of transmissions. The Weighted p-Persistence scheme [4] is a combination of a probabilistic and distance-based method, where retransmission probability is computed based on the

TABLE I  
THE INFORMATION CONTAINED IN A TRAFFICMAP ENTRY.

Parameter	Description	Size
Position $p$	Lat, Longitude	8 bytes ( $D^\circ m.m'$ , $D^\circ m.m'$ )
Velocity $v$	in m/s	1 byte (0.0-35.0 in 0.2 steps)
Heading $h$	angle of travel	1 byte (255 quanta $1.4^\circ$ each)

distance. The Slotted 1-Persistence scheme [4] calculates a timeslot in which rebroadcast will be scheduled depending on the distance between sender and receiver. The Slotted p-Persistence scheme calculates a timeslot, and then performs a transmission with probability  $p$ . In [4] the Slotted 1-Persistence scheme was found to perform best w.r.t. both reachability and the reduction in redundant transmissions. This method is further discussed and an improvement is provided in Sec. III-C.

### III. ARCHITECTURE

This section discusses the design of the TrafficFilter, a distributed ad hoc system which relies on Network-layer flooding. It is assumed that every vehicle has an IEEE 802.11p OBU and is able to obtain all required local information such as its position and speed.

#### A. TrafficFilter Protocol

The TrafficFilter generates a TrafficMap in every OBU on the road. The TrafficFilter is a distributed system in which many vehicles work together to build an over-the-horizon view of the road ahead using in-network aggregation. This information is contained in a datastructure called the TrafficMap.

The TrafficMap (TM) is a collection of tuples representing speed, position, and heading information along the current route of travel up to the *Virtual Horizon*.

Position and heading can be used as a key in relation to a roadmap, to derive the local traffic flow speed at that position. For simplicity, we assume one entry consists of 10 bytes, as shown in Table I. It immediately becomes evident that disseminating this information for *every* vehicle on a stretch of road will quickly reach the limits of available channel capacity. But it is not necessary to disseminate information from *all* vehicles; an important property of a (congested) traffic flow is that the speed of a vehicle is bounded by the speed of its lead vehicle and the speed limit. As a result, speeds will be similar in a local scope, allowing the system to just *sample* the information.

1) *Sample Capturing*: Fig. 2 shows a speed-position plot of 20 km of single-lane road generated with the Intelligent Driver Model (IDM), further discussed in Sec. IV-A. Between 8 and 12 km there is a traffic congestion, visible by the lower speed values and higher density of the vehicles. Note that, as vehicles leave the congestion area, the density decreases as they accelerate.

The figure also shows the aggregation of the information, an abstraction inspired by Run-length Encoding. For the moment, consider a message, visiting every vehicle as it progresses upstream (from right to left in Fig. 2). Only when a vehicle's speed deviates *enough*—based on a threshold function  $\varepsilon$ —will

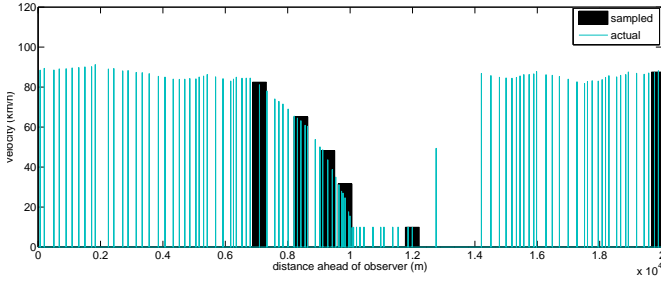


Fig. 2. A speed-position plot of 20 km of road with a congestion halfway, and its abstraction.

a new entry be placed in the TM. This is the method used to generate the abstraction in Fig. 2 and is exactly what the TrafficFilter does. Note that these few samples allow detection of the traffic jam, and can be fed into the Congestion Assistant.

The threshold function  $\varepsilon$  is defined as follows:

$$\varepsilon(v_{\text{last}}, v_{\text{own}}) = \begin{cases} \text{true} & \text{if (2) or (3);} \\ \text{false} & \text{otherwise.} \end{cases} \quad (1)$$

$$H(v_{\text{last}} - o_{\text{own}}) \cdot s_{\text{own}}(v_{\text{last}} - o_{\text{own}}) \geq v_{\text{own}} \quad (2)$$

$$H(v_{\text{own}} - o_{\text{last}}) \cdot s_{\text{last}}(v_{\text{own}} - o_{\text{last}}) \geq v_{\text{last}} \quad (3)$$

This threshold is based on the difference between the present own speed ( $v_{\text{own}}$ ) and the last-recorded speed in the received TM ( $v_{\text{last}}$ ). Eqs. (2) and (3) are simple difference functions enabled by a Heaviside step function for the own and last recorded speed in the TM, offset by  $o_{\text{own}}$  and  $o_{\text{last}}$ .

Note that these two expressions denote two areas in the  $v_{\text{last}}, v_{\text{own}}$ -plane; the situation where  $v_{\text{last}} = v_{\text{own}}$  represents the equilibrium state where the own vehicle has exactly the same speed as the last recorded speed. The offsets  $o_{\text{own}}$  and  $o_{\text{last}}$  in Eqs. (2) and (3) introduce an epsilon-environment around this equilibrium. Values outside this environment (e.g. inside the accelerating or decelerating areas as defined by Eqs. (2) and (3) and visualised in Fig. 3) cause the system to add a sample to the TrafficMap.

The function of the offsets  $o_{\text{own}}$  and  $o_{\text{last}}$  is not to add samples while in stop-and-go traffic, the two slope parameters  $s_{\text{own}}$  and  $s_{\text{last}}$  serve to tweak sensitivity to acceleration and deceleration: in case  $s_{\text{own}} > s_{\text{last}}$ , the system adds more samples on a “braking slope”, as visible in Fig. 2. Because the Congestion Assistant’s Active Pedal has to guide the vehicle’s speed as it approaches the tail of the congestion, more information on this part is desirable. For values  $(s_{\text{own}}, s_{\text{last}}) < 1$  the  $\varepsilon$  function becomes less sensitive at higher speeds, which is desirable because the goal of the system is to detect low speeds. Choosing a good threshold is key to obtaining a good representation of the road; if the threshold is too large we get few samples and might miss important details. If the threshold is too small the TM might grow explosively and contain redundancy.

By decreasing the offsets  $o_{\text{own}}, o_{\text{last}}$  (e.g. drawing the areas closer to the equilibrium) the sensitivity is increased and more samples are captured. In this case the threshold is reduced. The two areas are called “accelerating” and “decelerating” because

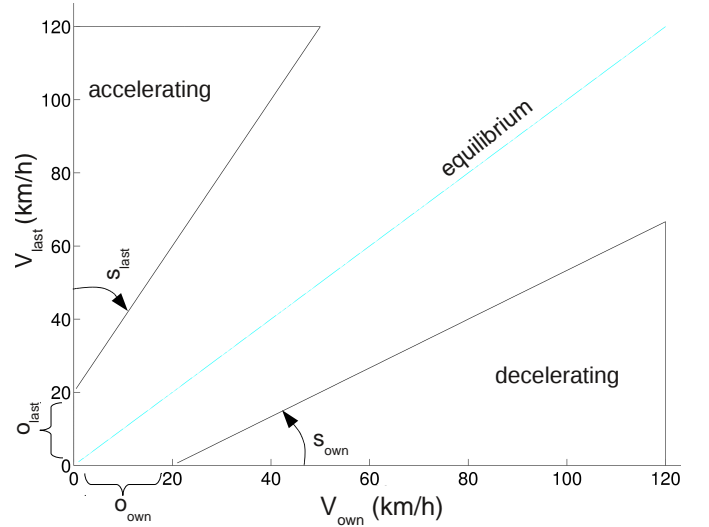


Fig. 3. The threshold function  $\varepsilon$  mapped to a 2-dimensional plane.

a large own speed  $v_{\text{own}}$  and a low last recorded speed  $v_{\text{last}}$  correspond to the tail of a jam (recall the TM is passed against the flow of traffic). Likewise, a low own speed and a high last recorded speed means the traffic downstream is accelerating. It is exactly this dynamic behaviour that justifies adding another sample to the TM.

Note that the sample capture policy proposed here is rather simple. A more advanced version of the  $\varepsilon$ -function might include dynamic adaptation based on the density of samples, the (local) traffic density, entropy of the information in the TM, distances between samples etc.

2) *Sample Averaging*: Every vehicle has to decide whether to add an entry to the TrafficMap. Although vehicles are influenced by factors such as speed limits and other traffic it is clear that there can be deviations, even in free-flowing traffic. An example is the difference between a car and a heavy truck it is overtaking. We would like a sample not to be a potentially locally deviating value, but a good representation of the average velocity in the immediate vicinity of the node that adds it. In order to make a sample representative for the general area around the vehicle we could introduce elaborate majority-voting schemes, but a simple averaging also suffices. The idea is as follows:

- 1) A node decides to add its measurement to the TrafficMap because it is allowed to do so by the  $\varepsilon$ -function.
- 2) The TrafficMap is rebroadcast.
- 3) A vehicle a short distance upstream receives it. Its  $\varepsilon$ -function does not allow it to add a new sample. It might, however, slightly alter the last entry  $(v_{\text{last}}, p_{\text{last}})$  if it is within the averaging distance  $\Delta$ .
- 4) The TrafficMap is rebroadcast if allowed by the rebroadcast mechanism (see Sec. III-C).

The result is that a sample is like a drop of paint, it gradually hardens and does not accept adjustment after a certain amount of time, or distance in this case, expressed as the averaging distance  $\Delta$ . The averaging is expressed in the following equation:

$$TM_{\text{new}} = \begin{cases} [(v_{\text{last}}^*, p_{\text{last}}), TM_{\text{old}}(2 \dots n)] & \text{if } \delta < \Delta; \\ TM_{\text{old}} & \text{otherwise,} \end{cases} \quad (4)$$

where

$$v_{\text{last}}^* = \frac{v_{\text{last}} + (v_{\text{own}} \times \theta(\delta))}{1 + \theta(\delta)} \quad (5)$$

and

$$\delta = |(p_{\text{own}} - p_1)|. \quad (6)$$

In words, the resulting value  $v_{\text{last}}^*$  is composed of the previous value of  $v_{\text{last}}$  plus a weighted amount of  $v_{\text{own}}$  at distance  $\delta$  from the location where  $v_{\text{last}}$  was captured. The weighing is handled by  $\theta(\delta)$  which is defined as follows:

$$\theta(\delta) = \left( \frac{\Delta - \delta}{\Delta} \right). \quad (7)$$

Eq. (7) gives a value between 0 and 1 for any  $\delta$  between 0 and  $\Delta$ , the averaging interval. Depending on  $\Delta$  and the vehicle density a sample  $v_{\text{last}}$  is made by one or multiple vehicles. Presently we use a set value for  $\Delta$  but it could be directly based (e.g. inversely proportional) on the density of traffic, the effects can be researched.

Eq. (5) ensures  $\frac{1}{1+\theta}$  of the original sample  $v_{\text{last}}$  is carried on in  $v_{\text{last}}^*$ . The result is an average calculated over an a priori unknown number of values.

Whether or not such averaging is of interest and what the effects are has to result from simulation or field studies. One can imagine that a car overtaking a truck in free-flowing highway traffic must not trigger addition of a sample to the TrafficMap. In this case, the difference of  $120 - 80 = 40$  km/h should not be allowed to trigger adding a sample and the  $\varepsilon$ -function should be calibrated accordingly. However, the averaging of several cars and one truck will result in a lower value due to the truck. Nevertheless it is argued that this gives a value well above speeds that might indicate traffic congestion.

3) *Sample Removal*: When a sample is captured it represents the situation at the location *where* it was captured, at the moment *when* it was captured. As soon as the information travels upstream, its relation to the actual situation diminishes, i.e. the confidence intervals, both spatial and temporal, increase. The traffic situation close ahead needs to be represented in detail because the Congestion Assistant's *Active Pedal* function needs precise information on where the congestion begins. Further away, the traffic situation does not need to be represented in detail and an aggregation can be performed to reduce the number of samples in the TM. Every node that rebroadcasts performs the following reduction operations:

- Two consecutive remote samples that are somewhat the same (as defined by a threshold  $\omega$ ) are reduced to one, the most remote one.
- A distant set of samples indicating a drop in speed (tail of a jam) resembles a set of stairs. They can be reduced to the first and last sample of these stairs. An observer can then interpolate between the samples.
- A sample is beyond the target virtual horizon. Samples beyond a certain distance are discarded to ensure information only flows as far as defined by the target virtual

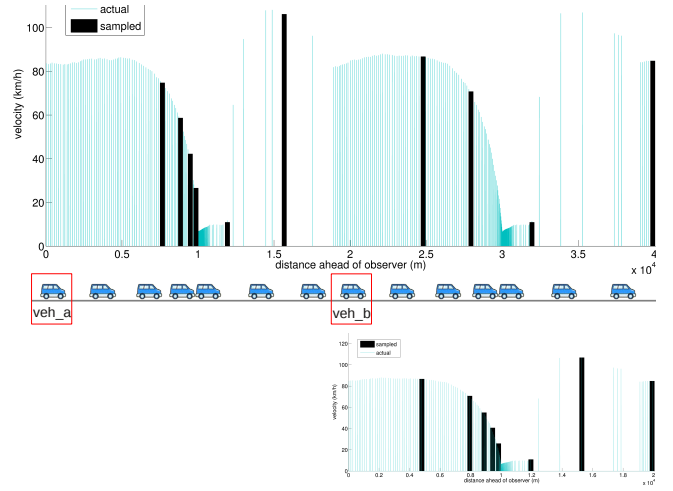


Fig. 4. A TrafficMap is passed upstream. Some redundancy is removed along the way.

horizon relative to host vehicle. As the information travels it ages, losing its relation to the actual present situation, rendering it obsolete.

- In order to meet demands of a maximum message size, remote samples might be discarded when there simply are too many samples in the TrafficMap. This could be the result of turbulent dynamics in traffic. An implication might be that the actual virtual horizon draws nearer.

Redundant samples generated because of a generous  $\varepsilon$ -function can be removed or merged based on a complete overview of the redundant sample's up- and downstream conditions. Whether to keep, remove or merge a sample is also threshold-dependent. This threshold, denoted as  $\omega$ , depends on traffic dynamics, just like the  $\varepsilon$ -function used as a capturing threshold, but also on the distance to the current node as confidence intervals increase with the distance.

The goal is to remove only redundant samples and reduce the size of the TrafficMap. This will be beneficial when the aim is to reach a large virtual horizon. Fig. 4 shows the TM present in two vehicles, one at position 0 (veh\_a) and the other further down the road (veh\_b). As can be seen, veh\_a sees two traffic jams up ahead. veh\_b only sees one (and has probably just passed the other one). Note that the bottom TrafficMap's 8 samples have been reduced to 4 in the top TrafficMap, without too much loss of detail. As veh\_a approaches the second congestion area, it will receive more accurate TM messages.

## B. States of the TrafficFilter

The TrafficFilter disseminates information against the flow of traffic, as discussed in the previous section. Because this information needs to be distributed pro-actively, it is considered best to send this information unsolicited, i.e. TMs are continuously flowing. Using a timer  $\tau$ , every node ensures TMs are disseminated on a periodic basis defined by the Maximum Inter-TrafficMap Time (MIT). In this work we

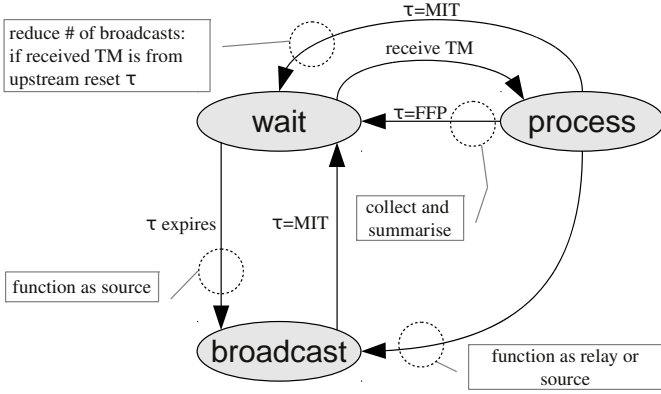


Fig. 5. The TrafficFilter protocol states.

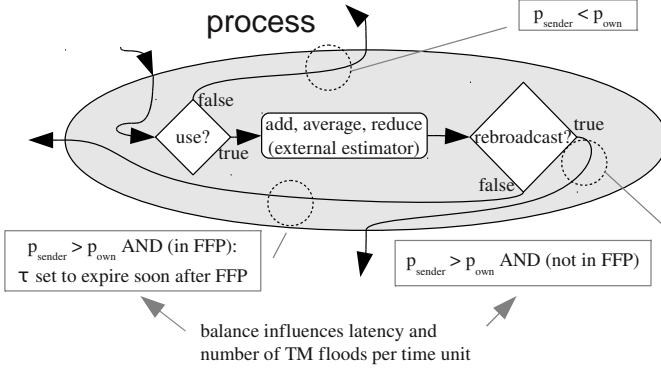


Fig. 6. The Process state in more detail.

consider a MIT of  $3 \text{ s}^{-1}$ .

The TrafficFilter protocol entity comprises a state machine, as shown in Fig. 5. The system is initiated in the `wait` state and will leave this state when a TM is received or upon the expiration of  $\tau$ . If a TM is received, control moves to the `process` state, where the received TM is processed for use by the Congestion Assistant. Based on the relation between the own position ( $p_{\text{own}}$ ) and the position of the sender ( $p_{\text{sender}}$ ), a broadcast will be executed if required, after which  $\tau$  will be reset to MIT and the system will return to `wait`, or the system will reset  $\tau$  and immediately return to `wait`. A special case is when a TM is received while the system is in the Flood Free Period (FFP). In this case the received TM will be processed but not rebroadcast, and  $\tau$  will be set to FFP. This will collect and summarise received information, as shown in Fig. 5, and prevents successive floods from following too close behind each other.

Fig. 6 shows the `process` state in more detail. If the received message originates from downstream (e.g. ahead of the host vehicle) it is used for further processing, otherwise  $\tau$  is reset and control moves back to `wait`. Next, the message goes through the `external estimator`, which compares the received TM to the local TM, the over-the-horizon awareness

<sup>1</sup>At highway speeds, a vehicle travels 250 m ( $R$ ) in  $\frac{250\text{m}}{33\text{m/s}} = 7.57\text{s} > 2\text{-MIT}$ . If an anomaly in traffic flow occurs, a vehicle receives notice before it arrives at the geographic location denoted in the message.

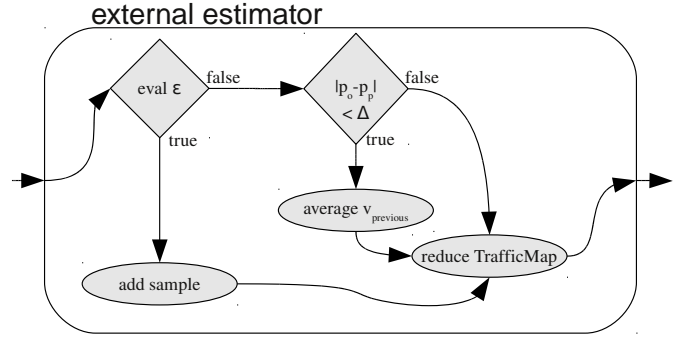


Fig. 7. The External Estimator in more detail.

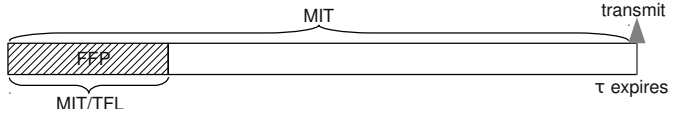


Fig. 8. The timing parameters MIT and FFP.

cached in the host vehicle. The external estimator decides to add, average, or remove samples as described in Sec. III-A. This process is the core of the TrafficFilter and is illustrated in Fig. 7. The threshold function Eq. (1) is evaluated and a sample will be added if necessary. If the  $(v_{\text{own}}, v_{\text{last}})$  tuple does not lie in either the accelerating or decelerating area the algorithm will check if it can still contribute by averaging its measurements with the last sample in the TM, as expressed in Eq. (4). Note that this serves the purpose of making one sample represent an average flow speed and not the speed of a single vehicle. Finally, the algorithm checks if the TM can be compressed by merging distant samples or removing those beyond the Virtual Horizon. The aim is to keep the TM as small as possible.

After the `external estimator` completes, the decision is made whether or not to rebroadcast the TM by means of the flooding scheme, see Fig. 6. In order to prevent creation of too many floods, several floods can be combined using a ‘collect and summarise’ approach, represented by the transition from the `process` to the `wait` state in Fig. 5. This function ensures that there is a certain maximum number of TM floods per MIT-period (defined as the TrafficMap Flood Limit, TFL). This relation is illustrated in Fig. 8.

The `broadcast` state in Fig. 5 consists of two components, as depicted in Fig. 9: a CSMA/CA broadcast provided by the 802.11 MAC DCF that will transmit as soon as the medium is observed idle (and the backoff counter reaches zero), and a flooding scheme for disseminating TM information. This latter event happens in reaction to reception of a TM message, the former happens upon expiration of  $\tau$ , a new flood will then be initiated.

The flooding mechanism is a method to break synchronised rebroadcast in response to reception of the same message at a group of nodes in a certain area, it works essentially by spreading the delay of the transition into the CSMA/CA broadcast. The rebroadcast will take place in the node furthest

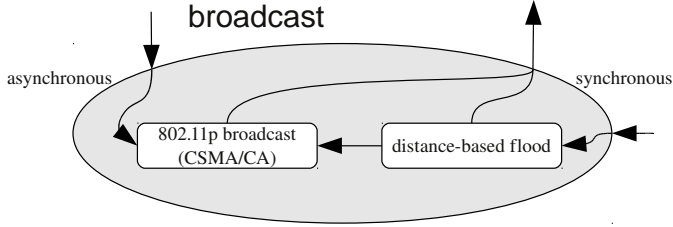


Fig. 9. The Broadcast state in more detail.

removed from the sender, in order to rapidly cover many kilometers. Two methods to do this are presented in section III-C.

The asynchronous, direct broadcast is used when the  $\tau$ -timer has expired and when to satisfy MIT a new flood needs to be initiated. This means a MIT-period has passed during which no TM flood was observed by the vehicle. This can be caused by gaps in the network; existing floods have died out because the chain of propagation is broken. The node will inject a new flood.

It is envisioned that during traffic congestion the majority of transmissions will take place as synchronous transmissions (i.e. mediated by the flooding scheme) for maximum efficiency. A broadcast triggered by an event does not imply *immediate* broadcast, as more safety-critical application may get priority in the MAC layer<sup>2</sup>.

### C. Flooding

The broadcast state as shown in Fig. 9 relies on a distance-based flooding scheme as illustrated in Fig 10. Flooding is the act of injecting a message into the network, which will then be propagated to all nodes or a subset. When a node receives a message, it will forward it. It is important that a message fits in one frame because then no state has to be maintained and the flooding will be more robust; reception of one frame does not depend on correct reception of prior or subsequent frames.

In the case of the TrafficFilter, the flood needs to be propagated against the flow of traffic, and the contents of the message may be altered at every hop. In order to prevent a broadcast storm [16], a broadcast suppression technique needs to be employed so only specific nodes rebroadcast. In [4] Wisitpongphan *et al.* evaluate several flooding strategies for use in VANETs; the Slotted 1-Persistence Flooding was shown to be the best trade-off between efficiency and latency. This scheme is explained below, after that an improvement is provided.

1) *Slotted 1-Persistence*: When a broadcast is received by a node, transmission of the rebroadcast will be scheduled in timeslot  $S_{ij}$ . Scheduling a rebroadcast occurs only when the packet has been received for the first time, this can be recognised by the FloodID. If a duplicate is received before the

<sup>2</sup>because of 802.11p's EDCA QoS based on 802.11e, the TrafficFilter is designed to use lowest priority because it does not convey safety-critical information. The result is a larger contention window and AIFS.

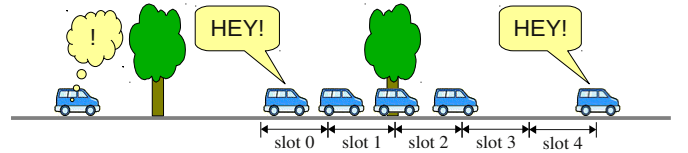


Fig. 10. Nodes rebroadcast based on the distance to the sender, and refrain from broadcasting if they overhear a rebroadcast from a node further away (e.g. vehicles in slots 1 and 2).

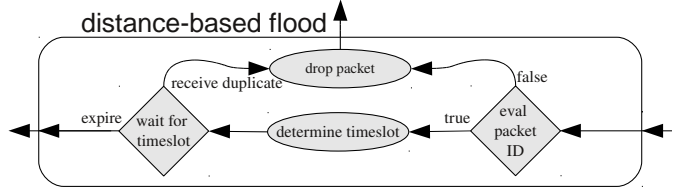


Fig. 11. Distance-based flooding by means of Slotted 1-Persistence Flooding.

rebroadcast is executed, the packet will be discarded and the rebroadcast will be cancelled; this node will not rebroadcast because an other node already did. This is illustrated in Fig. 11, which shows the distance-based flood component of Fig. 9.

The timeslot chosen by node  $i$  to rebroadcast a packet follows from (8) and depends on  $D_{ij}$  (the distance between the nodes  $i$  and  $j$  where  $j$  is the node which transmitted the message and node  $i$  is the host vehicle), the (estimated) transmission range  $R$  and an a priori determined number of slots  $N_s$ . Because a transmission range cannot be deterministically derived because of erratic channel properties,  $R$  is chosen such that it corresponds to the average transmission range.

The time a node has to wait before rebroadcasting, expressed as  $T_{S_{ij}}$ , can be calculated as follows<sup>3</sup>:

$$T_{S_{ij}} = t_s \cdot \left\lfloor N_s \cdot \left( 1 - \frac{\min(D_{ij}, R)}{R} \right) \right\rfloor \quad (8)$$

where  $t_s$  (the one-hop delay or slot time) is the sum of medium access, transmission and propagation delay. This allows a node to completely receive a message transmitted by another node in a previous slot, and decide whether it still needs to rebroadcast in the next slot. In [4] this value is defined as 5 ms. We set  $N_s$  to 5, which also conforms to [4]. The result is that a node for which  $D_{ij}$  is larger will pick an earlier timeslot and will sooner be able to rebroadcast. However, this will only be done if no other node rebroadcasts in the mean time.

2) *microSlotted 1-Persistence*: The Slotted 1-Persistence flooding scheme described above breaks the synchronisation of simple flooding, which would otherwise result in all nodes trying to access the medium simultaneously. It is identified, however, that a similar synchronisation—albeit on a smaller scale—can occur within one slot when vehicle densities are high (as is very likely in congested traffic conditions). As such, the Slotted 1-Persistence flooding does not completely solve

<sup>3</sup>The formulation in [4] appears to be incorrect, as ceiling operators are placed around  $\frac{\min(D_{ij}, R)}{R}$  which would result in rounding a value between 0.0 and 1.0 to 1 which would result in  $S_{ij} = 0$  for all  $D_{ij}$ .

the broadcast storm problem. A means to address this problem is using a probability of transmission lower than 1, as in the p-Persistence scheme [4] or choosing a larger number of slots  $N_s$ . In [4] it was shown that the first scheme does not perform as well as the Slotted 1-Persistence and that the second option will also deteriorate performance: the slot time  $t_s$  will remain constant because propagation and MAC processing delay are constant, but with more slots the cumulative delay can be much higher.

Recently, this problem has been identified by Blum and Eskandarian [18] as the *Timeslot Boundary Synchronization Problem*. It is the result of 1) backoff counter synchronisation because multiple nodes freeze their backoff counter due to a transmission by another node or 2) simultaneous message creation. Messages may be created simultaneously in response to a rebroadcast operation, as is the case in the Slotted 1-persistence flooding. When nodes simultaneously perform Clear Channel Assessment (CCA) they find the medium empty and immediately start transmitting. If the medium is not empty, they draw a delay from a contention window  $[0, CW - 1]$ . The probability of two nodes choosing the same delay is  $\frac{1}{CW}$ , with more nodes collisions become more likely. It is recommended in [18] that the synchronisation be staggered by means of some (random) delay.

We propose a solution which relies on the CSMA/CA in the IEEE 802.11 MAC layer, and alter the Slotted 1-Persistence flooding scheme accordingly. The mandatory time between transmissions is defined in the standard [19] as DIFS (Distributed Inter Frame Space). If every node waits a multiple of DIFS ( $64\mu s$  in IEEE 802.11p [10]), nodes will not all sense the medium simultaneously. As a result, in most cases one node will start transmitting first and the others will find the medium busy. Because this scheme is a slotting scheme at a fine granularity it is referred to as the microSlotted 1-Persistence Flooding scheme. This scheme is based on the Slotted 1-Persistence Flooding and adds a small delay of  $T_{ms_{ij}}$  (i.e. an integer multiple of DIFS) to every  $T_{s_{ij}}$  defined by Eq. (8). Note that we still preserve the slots as in the Slotted scheme, Eq. (8) still holds and a rebroadcast can still be cancelled if a duplicate is received before the rebroadcast is executed.  $T_{ms_{ij}}$  is defined as:

$$T_{ms_{ij}} = t_{ms} \cdot \left\lceil N_{ms} \left( 1 - \frac{D_{ij} \bmod S}{S} \right) \right\rceil \quad (9)$$

where  $t_{ms}$  is the duration of a microSlot, taken as DIFS.  $N_{ms}$  is the number of microSlots per slot based on the estimated transmission range  $R$ , number of slots  $N_s$  and average vehicle length  $\bar{l}$ :

$$N_{ms} = \frac{\left(\frac{R}{N_s}\right)}{\bar{l}} \quad (10)$$

For an  $R$  of 250 m,  $N_s$  of 5 and  $\bar{l} = 5$  m,  $N_{ms}$  is set<sup>4</sup> to 10. The modulo operator in Eq. (9) scales the domain  $D_{ij}$  to the size  $S$  of the timeslot associated with the geographical slot within which the node is located. This prioritises the most

remote node within a slot. The geographical size of a slot,  $S$  is defined as  $\frac{R}{N_s}$ . As a result, the total wait time for node  $i$  after receiving a packet from node  $j$  is the sum of (8) and (9):

$$T_{wait} = T_{s_{ij}} + T_{ms_{ij}}. \quad (11)$$

Node  $i$  will schedule to hand the message over to the MAC layer after  $T_{wait}$ . Behaviour is just like the Slotted 1-Persistence Flooding: if, before handing the message to the MAC layer, a packet is received with the same FloodID, the pending transmission is cancelled; another node performed a successful broadcast and node  $i$  does not need to contribute.

This flooding strategy assumes an 802.11-compatible MAC and PHY because it relies on CSMA/CA, assuming higher layers have no (or limited) control over its operations. This means that, once the Network layer sends a message down to the MAC for broadcast, it cannot be recalled. Within one slot as defined in Eq. (8) several values for  $T_{ms_{ij}}$  may exist. The result is that the transmissions by separate nodes are executed serially by means of backoff instead of parallel (which would result in collisions). A benefit is that the probability that at least one broadcast in a slot  $t_s$  is successful, increases.

A drawback is that all messages scheduled in the MAC layer still have to be executed, resulting in a larger medium busy time or channel utilisation. An evaluation of the performance of the microSlotted and the original Slotted 1-Persistence Flooding scheme is provided in Section IV and shows this has little adverse effect.

#### D. Extensions for multi-lane

It is possible to extend the TrafficFilter and the underlying dissemination mechanism for use on multi-lane, bi-directional roads and junctions, as described in more detail in [5]. In short, several fields are added to the TM message. A TM entry has been extended with a lane number, a road segment identifier (derived from a map), and also is aware of the position and type of junctions. The Sample Capturing, Averaging and Removal procedures have been modified accordingly and the underlying dissemination mechanism (the microSlotted 1-Persistence flooding) has been modified to give priority to nodes which have added an important sample. The multi-lane TM could be visualised as several parallel single-lane TMs as shown in Fig. 4.

## IV. PERFORMANCE

The performance of the TrafficFilter system is evaluated in the discrete event network simulator OMNeT++ [20], using the Mobility Framework [21]. This section will evaluate a) the performance of the proposed addition of microSlots to the Slotted 1-Persistence Flooding, b) the performance of the TrafficFilter on top of a flooding dissemination mechanism, and c) the influence of mobility on TrafficMap accuracy and flooding efficiency.

#### A. Network model and assumptions

This paper considers a single-lane, unidirectional highway of 10 km, which is modelled in OMNeT++. An extension for

<sup>4</sup>the geographic size of a slot is 50 m. The size of a microSlot then equals  $\frac{50m}{10} = 5m$ , the size of an average vehicle.



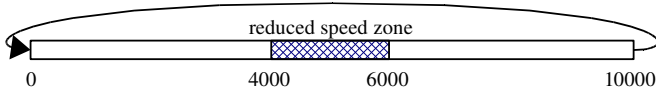


Fig. 12. Road model used in the simulations, a traffic congestion is simulated between 4000 and 6000 m.

multi-lane scenarios is provided and evaluated in [5]. When a vehicle reaches the end of the road it wraps around to the beginning, as illustrated in Fig. 12. Vehicle densities (denoted with  $\rho$ ) from 10 to 150 veh./km are evaluated. No information is collected while the simulation is in its warmup period, in order to analyse steady-state behaviour. The experiments were performed using the method of independent replications. To achieve statistical significance, every scenario is repeated 50 times, and one run constitutes 100 TM disseminations. The mean and 95% confidence intervals are provided in the results.

The TrafficFilter system is tested using Slotted 1-Persistence flooding and microSlotted 1-Persistence flooding. Both scenarios are evaluated with and without node mobility to separate mobility-induced effects from the flooding schemes ability to deal with the broadcast storm problem. Because the Congestion Assistant needs to be informed of upcoming congestions, we model a congestion halfway down the road in case of mobility.

Vehicles know their exact location (we assume a perfect global positioning system (GPS)) and all vehicles can communicate. For all scenarios, vehicles are initially placed according to a uniform distribution so as to guarantee a fully connected network (i.e. no two nodes are further removed from each other than the transmission range). Furthermore, vehicles remain stationary during the experiments where no mobility is involved. This way a flooding mechanism's ability to deal with the broadcast storm problem can be researched because the results are not affected by mobility-induced effects such as network partitioning.

When mobility is involved, vehicles move according to the Intelligent Driver Model (IDM) [22], implemented in OMNeT++'s Mobility Framework and parameterised as shown in Table III. The IDM is a microscopic car-following model, i.e. it models longitudinal behaviour of individual vehicles based on velocity, distance to the vehicle in front and the velocity difference with the vehicle in front. This means the speed of a vehicle depends on two factors: 1) the speed limit  $v_0$  and the time headway  $T$  between this vehicle and the one in front. Time headway at a constant speed results in a distance: at 80km/h and  $T = 1.6$ , the distance to the vehicle in front would be approximately 35.5 meters. If this distance decreases, the vehicle slows down (to maintain its time headway). If the distance increases, the vehicle speeds up (but does not exceed the speed limit).

Before conducting the experiment the system runs for 300 s in order for traffic to stabilise. A traffic congestion is introduced by setting the maximum speed at 20 km/h between 4 and 6 km, see Fig. 12. As the speed drops on the stretch of road between 4 and 6 km, the IDM will maintain a shorter following distance. Vehicles will manifest the typical stop-and-go behaviour while in the congestion area. The result is a high

TABLE II  
PARAMETERS OF THE TRAFFICFILTER

Parameter	Description	Value
$MIT$	Max. Inter-TM Time	3 s
$FFP$	Flood-Free Period	0.5 s
$\Delta$	averaging interval	500 m
$o_{own}$	offset w.r.t. own speed	5 m/s
$o_{last}$	offset w.r.t. last speed in TM	7 m/s
$s_{own}$	sensitivity to braking	8/9
$s_{last}$	sensitivity to acceleration	5/6
Virtual Horizon	when to drop samples	10 km
$R$	estimated communication range	250 m
$N_s$	number of slots	5
$N_{ms}$	number of microSlots	10
$t_s$	slot time	5 ms
$t_{ms}$	microSlot time	DIFS

TABLE III  
PARAMETERS OF THE INTELLIGENT DRIVER MOBILITY MODEL

Parameter	Description	Value
$a$	acceleration	0.73 m/s <sup>2</sup>
$b$	deceleration	1.67 m/s <sup>2</sup>
$T$	time headway	1.6 s
$s_0, s_1$	jam distance	2 m, 0 m
$v_0$	speed limit	130
$\delta$	acceleration exponent	4
$dt$	timesteps	0.1 s
$\bar{l}$	vehicle length	5 m

local vehicle density in the congestion, and low density at the head of the congestion: vehicles will accelerate again.

The TrafficFilter is implemented as a Network-layer module on top of the Mobility Framework MAC and physical layer modules. The TrafficFilter is parameterised as discussed in Sec. III and summarised in Table II. The MAC and physical layer modules are adapted to conform to IEEE 802.11p specifications [10], as provided in Table IV. Signal propagation is modelled with the Friis Free Space formula with  $\alpha = 3.5$  and the collision model is based on Signal to Noise Ratio using Bit Error Rate curves, as is default in the Mobility Framework. We set the power such that the estimated transmission range  $R$  is 250 m. The carrier frequency is 5.87 GHz and bandwidth is 10 MHz. This is one of the DSRC channels allocated for road safety and traffic efficiency [23]. This research does not consider influence of mobility on the physical layer and signal propagation, such as Doppler effects and time-varying channel conditions.

A TrafficFilter flood is triggered every MIT seconds at the vehicle closest to the 10 km-point, this is the flood initiator. The flood then travels through the network (against the flow of traffic). The payload and headers are padded (if necessary) to reach a MAC service data unit (MSDU) of 300 bytes, resulting in the same transmission delay for each transmission. 300 bytes are sufficient to carry approximately 25 entries in the TM.

In the experiments, the vehicle density  $\rho$  is varied from 10 to 150. This has impact on the average speed of vehicles, forcing drivers to reduce their speed.

### B. Evaluation of the Flooding Schemes

In these experiments, the Slotted and microSlotted flooding schemes both aim to disseminate TM information. The flood-

TABLE IV  
PARAMETERS OF THE 802.11P MAC AND PHY

Parameter	Value
MAC Frame size	300 bytes
slotTime $\sigma$	16 $\mu$ s
DIFS	64 $\mu$ s
Contention Window $CW$	16
Carrier frequency	5.87 GHz
Bandwidth	10 MHz
Bitrate	6 Mbps

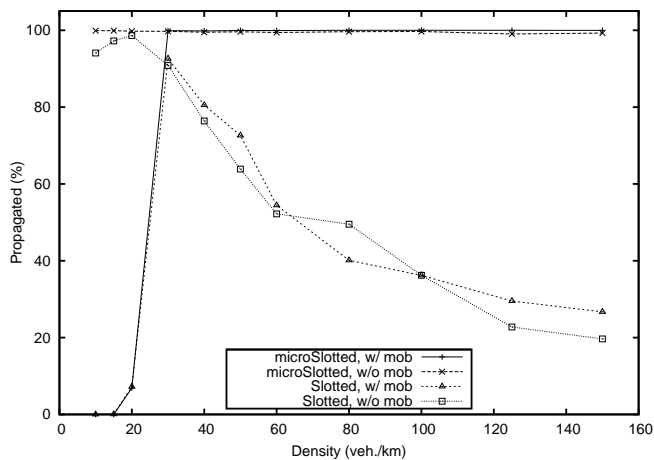


Fig. 13. Reachability: the average flood propagation percentage across 10 km of highway.

ing schemes are compared based on reachability, delay, hop count, and medium utilisation.

1) *Reachability*: This metric measures the percentage of the floods launched by the flood initiator (at 10 km) which eventually reach the trailing vehicle (closest to 0 km). It is a measure of the ability of a flooding scheme to disseminate information over great distances.

The reachability of the Slotted scheme (see Fig. 13) rapidly drops as  $\rho$  increases, but some floods still get through. Even though the Slotted 1-Persistence flooding scheme ameliorates the broadcast storm, a considerable degree of broadcast storm remains under high vehicle densities. This causes an increasing number of collisions as  $\rho$  increases. The microSlotted scheme maintains a delivery ratio close to 100%. The reason why the microSlotted scheme does not maintain exactly 100% delivery is because the probability that a flood will not fully propagate—however minute—is still present because collisions can still occur. For densities  $\leq 20$  veh./km gaps larger than the transmission range occur when mobility is involved, these gaps stop dissemination.

2) *Delay*: Delay is defined as the time needed for a TrafficMap flood to traverse the highway from the 10 km point (the flood initiator) to the vehicle closest to the 0 km point. It is a measure of the age of the information received from 10 km ahead. This is only calculated for fully propagated floods, which accounts for the relatively large confidence intervals for the Slotted scheme in Fig. 14.

The microSlotted scheme relies on adding a small extra delay to the wait time defined by the Slotted scheme. This delay is in the order of  $[0-9]$  DIFS periods (0–576 $\mu$ s) per hop.

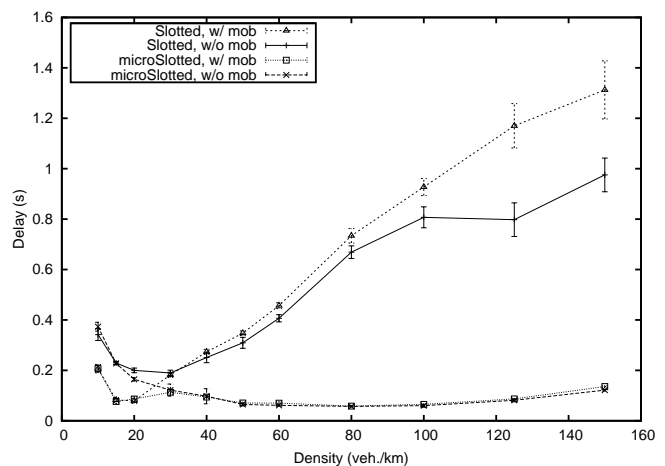


Fig. 14. Delay: time needed to travel 10 km.

This results in a tiny extra delay per hop for the microSlotted scheme, barely visible in Fig. 14. As  $\rho$  increases beyond 20 veh./km, the two schemes clearly behave differently; the original scheme shows an increasing delay as the number of nodes increases, the modified scheme exhibits a diminishing delay. This behaviour of the Slotted scheme can be attributed to an increasing number of collisions within the first slot, which means a rebroadcast in the next slot is executed one slotTime  $t_s$  later. If that broadcast collides, one  $t_s$  later the nodes in the next slot get a chance to broadcast, until a successful broadcast occurs or a flood dies out completely (reducing the reachability, as seen in Fig. 13).

At low densities a high delay is observed. This is due to the fact that, with fewer nodes, the nodes are not always at optimum distance from each other. For instance, it could very well be that there are no vehicles in the first three slots but there is one in the fourth (i.e. close to the rebroadcaster). This would incur three  $t_s$  of delay just for this hop, resulting in a 15 ms penalty to the end-to-end delay. With increasing density it becomes increasingly more probable that there is a vehicle in the extremes of the estimated transmission range, resulting in covering a large distance per hop and thus covering the full 10 km in fewer hops. Both contribute to a lower delay. The first because rebroadcast takes place immediately in the first timeslot, the second because fewer hops are needed altogether. Beyond 80 veh./km the delay of the microSlotted scheme also starts to rise. This is because, even with the microSlots, probability of collision increases with the vehicle density and rebroadcast by nodes in later slots is required. The additional delay of the microSlots is small compared to the overall delay. With an average of 50 hops (see next metric) the delay incurred by the microSlots amounts to at most  $50 \cdot 9 \cdot \text{DIFS} = 28.8$  ms whereas the overall delay is in the order of 100 ms. This is not visible in Fig. 14 because as  $\rho$  increases, it becomes more probable there is a vehicle in earlier microSlots, a behaviour also observed for the timeslots of  $t_s$ . Both the Slotted and microSlotted schemes achieve a greater propagation speed than the SOTIS approach [15] which takes  $\sim 3.5$  s/km, hence the received over-the-horizon awareness has better freshness.

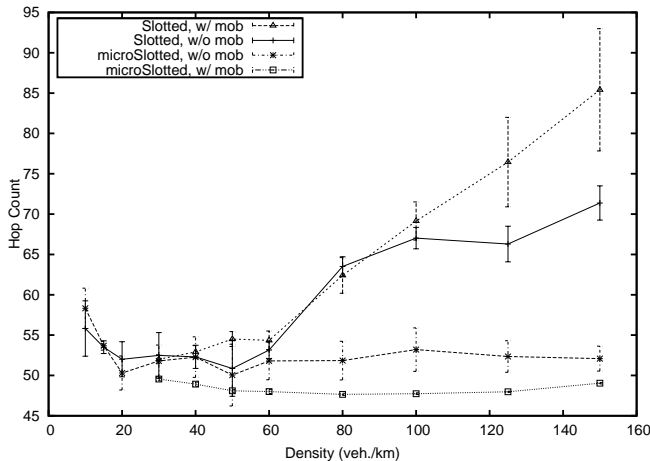


Fig. 15. Hop Count: average number of hops required to travel 10 km.

3) *Hop Count*: Hop count is the number of rebroadcasts (“hops”) that are required for a message to travel from the flood initiator to the 0 km point, i.e. the number of hops needed to traverse 10 km. Hop count gives insight in the performance of the flooding scheme; because every broadcast consumes resources a low hop count is favourable. This metric is only calculated for fully propagated floods. Hop count is shown in Fig. 15. Note the relatively large confidence intervals for the slotted scheme, this is due to the reduction in fully propagated floods, also visible in Fig. 14.

After  $\rho = 15$  veh./km, the probability that a slot contains more than one vehicle increases. In the slotted scheme this was found to result in a larger probability of collision because the vehicles in one slot will be synchronised when performing their rebroadcast. If a collision occurs in the first slot, the vehicles in the second slot will not discard their rebroadcasts (they have not successfully recognised a rebroadcast by a more distant node) and will rebroadcast when it is their time. If in the second slot there are also more than one node, a collision can occur. In the extreme, collisions occur in all successive slots and the flood will die out, resulting in a deleterious effect on reachability, as evident in Fig. 13.

If a collision occurs in the first slot, a successful rebroadcast can occur in the second slot, or in the third and so on. This implies that, although a rebroadcast does take place, the geographically covered distance is smaller than the theoretical optimum. As a result, more hops are needed to traverse the full 10 kilometers. This is reflected in the rise of the number of hops and can be attributed to collisions of transmissions scheduled in earlier slots.

The microSlotted scheme appears to suffer to a lesser degree from an increase in hop count due to collisions. This has two reasons:

- 1) Because of the microSlots, collisions in the first slot are less likely because there is less synchronisation, a rebroadcast in later slots is hardly needed.
- 2) With increasing density, the probability of a collision within one microSlot increases. However, if a collision is to occur in the first microSlot the CSMA/CA mechanism of the MAC layer will ensure the transmission scheduled

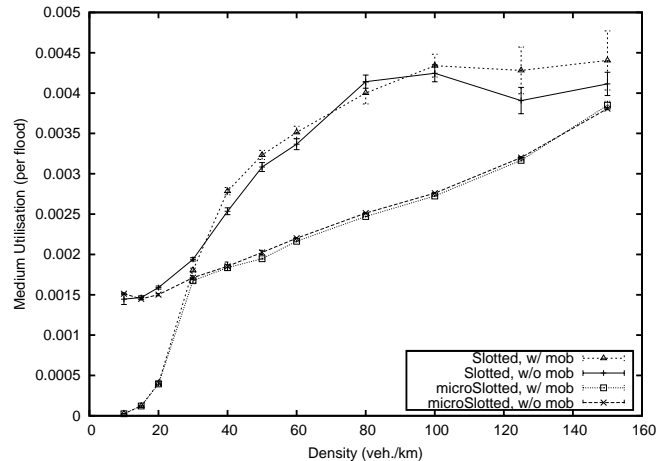


Fig. 16. Medium Utilisation: seconds of busy time per flood.

in a later microSlot by a different node can still go through, albeit with a slight delay due to backoff.

With the microSlotted scheme, a vehicle density has to become so high as to guarantee multiple vehicles per microSlot before the effects of collisions start to have serious effects. With the microSlot size of  $\frac{50\text{m}}{10} = 5$  m as employed in this study, this becomes a possibility when multiple lanes have high vehicle densities, such as during rush hours.

4) *Medium Utilisation*: Medium utilisation (or channel busy time) is defined as the mean time in seconds (normalised per flood) a node sees the channel busy (e.g. carrier sense detects energy on the medium). This metric is calculated for all nodes and all floods, including those which did not fully propagate. The medium utilisation is shown in Fig. 16.

The microSlotted scheme uses the available resources more efficiently and hence is able to maintain a lower medium utilisation. Interesting to see is that the microSlotted scheme rises linearly while the slotted scheme eventually levels as  $\rho$  is increased. This can be explained by the fact that many floods no longer fully propagate. As a result the average medium utilisation per node is lower because many nodes (those furthest away from the flood initiator) observe no transmissions at all.

Floods are initiated every 3 seconds, hence transmissions are of a bursty nature. For  $\rho = 150$ , the microSlotted scheme has a utilisation of 3.8 ms per 3 s, or 0.00127%. This means the channel is idle and available for other traffic most of the time.

5) *Slot Allocation*: Choosing the correct estimate for the transmission range  $R$  in (9) is of great impact on the efficacy of the flooding scheme. Intuitively it is clear that, in order to achieve a low end-to-end delay, it is paramount the geographical footprint of the first slot is correctly aligned with the most remote receivers of a transmission. The flooding scheme, then, must see to it that most of the successful transmissions are carried out in the first timeslot whenever a capable node is present. This can become a challenge when multiple nodes exist within this single slot. The behaviour discussed in the previous sections on performance is reflected in the distribution of the slot utilisation, as depicted in Fig.

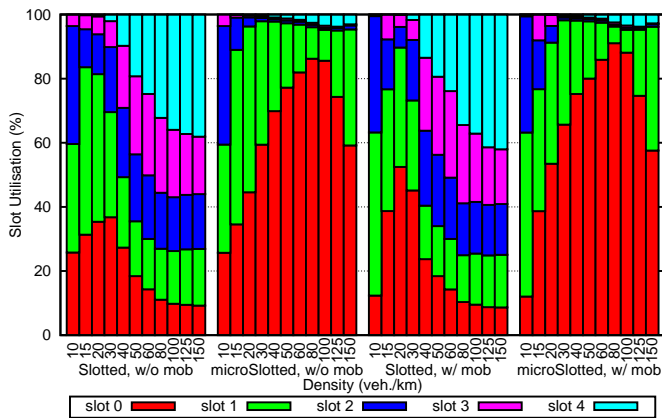


Fig. 17. Slot Utilisation - shares of slots [0...4] are stacked.

17. This figure shows the slot allocation for the two flooding strategies with and without mobility. The slot number is logged only when a node sends the TM message down to the MAC layer. As a result, both colliding and successful transmissions are recorded. In the following, we will first consider the static case, where nodes are placed with spacings drawn from a uniform distribution.

*a) Static Network:* At 10 veh./km  $\sim 25\%$  of the transmissions is carried out in slot 0 while another  $\sim 35\%$  occurs in slot 1, 35% in slot 2 and 5% in slot 3. Because the probability for a slot to contain multiple vehicles is low, this results in the same behaviour for both schemes. The reason why some transmissions occur in slot 1 or 2 and not in slot 0 is quite simple. There exists a probability that there simply is no vehicle present in the geographic area covered by slot 0. With more vehicles per kilometer it becomes more likely that a vehicle occupies the area covered by slot 0. Because of the uniform distribution of the nodes, every slot is equally likely to have at least one vehicle in it.

The interplay of the probability of correct reception (w.r.t. bit errors in the packet) and cumulative probability a node is present in slot  $n$  but none is present in previous slots results in an approximately even share for the first three slots. As the vehicle density increases, it becomes clear that more transmissions are scheduled in later slots when using the Slotted scheme, indicated by the left-most histogram in Fig. 17. This is because a growing share of the transmissions per slot collides, so the rebroadcast task is implicitly handed over to vehicles in later slots. The vehicles in later slots have not received a duplicate with the same FloodID, and do not cancel their own transmission attempt. This explains the increase in delay, the increase in hops required to cover the full 10 km and why some floods do not make it across the entire road at all. These floods suffer consecutive collisions in all slots.

The microSlotted scheme, however, shows an increasing utilisation of slot 0 up to around 80 veh./km. The microSlotted scheme has a low probability of collision within one slot because rebroadcast synchronisation is broken at the network layer. As a result, when a transmission does not succeed in slot 0, it is likely to succeed in slot 1. This explains the low utilisation of slots 2 through 4.

*b) Mobile Network:* In the mobile network (the two right-most histograms in Fig. 17) the situation for  $\rho \in \{10, 15, 20\}$  differs from the static case. This can be attributed to the distribution of cars on the road, plus the very low percentage of fully propagated floods at these densities. For  $\rho > 30$  the slot allocation resembles that of the static counterparts. This is also reflected in the results for reachability, delay, hop count and medium utilisation.

Node mobility seems to have little effect on the flooding scheme's ability to propagate a message. The largest threat to a flood in a VANET is network partitioning, as reflected in the results for low vehicle densities. Differences between the static and mobile scenarios can be explained by the way vehicles are distributed on the road; being uniform in the static case and variable under the IDM. As a result, there may be many vehicles per slot in the traffic jam and few at the head of the jam. As we saw previously, an increasing vehicle density makes it harder for the flooding scheme and the CSMA/CA to coordinate transmissions.

### C. Quality of Communicated Information

As information travels, it loses its relation to the situation where and when the information was collected. For example, information in the TM could indicate that 8 km downstream the traffic flow speed is 50 km/h. The degree to which this information correlates with reality depends on three factors: 1) measurement accuracy (e.g. GPS accuracy), 2) aging of information while being processed and in transit, and 3) inaccuracy incurred by the sampling approach. These factors have different sources:

- 1) is inherent to the GPS system which has an accuracy of 20 m with 99 per cent confidence [24]. This factor is not considered in these experiments.
- 2) is dominated by the rate at which the information ages. During the average flood dissemination period ( $\sim 100$  ms), traffic moves at most 3.3 m (considering highway speeds).
- 3) is the degree with which the node performing the rebroadcast is also the *best* node to do so. This could occur, for example, when several vehicles are overtaking a slower vehicle but the slower vehicle adds a sample to the TM. The reported traffic flow speed will then be lower than the actual. This factor is determined by TrafficFilter parameters (such as the  $\epsilon$ -function which determines thresholds).

The simulator is instrumented to record the position and speed of all nodes and the communicated TM at the moment the node closest to the 0-km point receives a TM message. This approach measures the effects of 2) and 3) and results in an average deviation in TM information for a certain traffic density. This experiment measures how well TM entries match the actual situation (factor 2 in the list above) and how well an interpolated TM (a representation of reality) matches the actual situation (simulated reality itself). The resulting mean and 95% confidence intervals over 50 runs of 100 floods each are shown in Fig. 18. In these calculations, only fully propagated floods were considered.

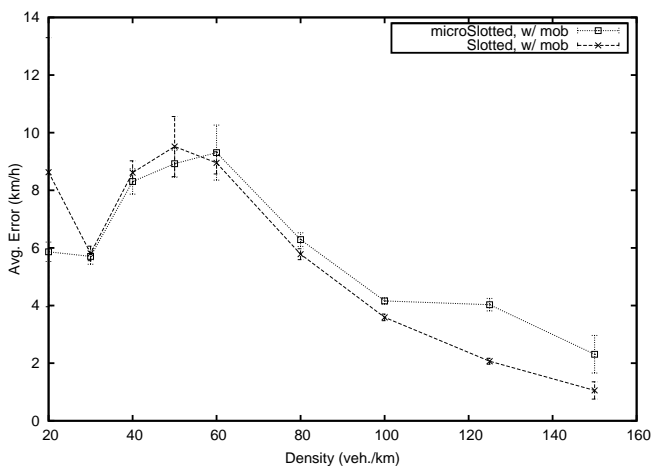


Fig. 18. Errors in communicated speed information.

The error between TM entries and the actual situation (how well a sample represents the actual situation at that point) were found to be negligible when compared to the error between the interpolation of a TM and the actual situation. As vehicles move faster, this error increases. Note that this evaluation only considered the validity of the samples at the moment a TM arrives. Due to the periodic nature of this system (paced by  $\tau$ ) a vehicle receives new TM information every  $\tau$  seconds. We did not evaluate the impact of aging once a TM has been received (e.g. what happens in the 3 s between consecutive TM receptions).

How well the interpolated TM (using a connect-the-dots approach) matches the actual situation at the moment of reception is highly dependent on the parameterisation of the TrafficFilter. These need to be matched to the vehicle density  $\rho$  and the resulting traffic flow speed (as these are inherently related under the IDM or any other realistic driver behaviour). Because the sampling is carried out by means of thresholds, a new entry is only added to the TM if the threshold is exceeded. This implies that, in between entries, the actual traffic flow can show ripples which go undetected by the TrafficFilter because no threshold is exceeded.

The difference between the two flooding schemes results from the position and number of broadcast nodes selected by the flooding schemes, and the reachability associated with each scheme. In Fig. 15 we saw that the Slotted scheme requires more hops to traverse the 10 km. Fig. 17 explains why, and the impact on reachability is evident in Fig. 13. These observations can be used to explain the different behaviour of the microSlotted scheme in Fig. 18 after  $\rho$  exceeds 100. The Slotted scheme takes small hops at high vehicle densities, because often collisions occur in remote slots. Because the road model is only single lane, the speed of every vehicle depends on that of its predecessor by means of the IDM. As such, a small cluster of vehicles will not show large speed deviations. The Slotted scheme “selects” a rebroadcaster closer to the broadcaster. Because of the physical influence vehicles have on each other, the deviations in speed are small. The microSlotted scheme, on the other hand, is able to coordinate rebroadcasts under a much larger vehicular density, and as

such rebroadcasts in the most remote slot are more common ( $\sim 75\%$  and  $\sim 57\%$  at  $\rho = 125, 150$  respectively, see Fig. 17). This means that the rebroadcasting nodes can have a larger deviation under the microSlotted scheme than with the slotted scheme, as reflected in Fig. 18.

From the discussion above it becomes clear that the accuracy of the over-the-horizon awareness generated by the TrafficFilter depends to a great extent on the parameterisation; in this work performed by a simple threshold-based selection criterion. Especially the  $\varepsilon$ -function and  $\tau$  are important parameters to tune the TrafficFilter, and need to be matched to traffic flow dynamics. In this work we used the IDM to obtain some qualitative indications, but the exact parameterisation is left as future work.

## V. CONCLUSIONS & FUTURE WORK

Over-the-horizon awareness can be provided to Advanced Driver Assistance Systems by means of an ad hoc multi-hop V2V solution. The system designed in this research, the TrafficFilter, has two core tasks: 1) selection of information in a datastructure called the TrafficMap, and 2) the efficient dissemination of this information by means of V2V communication.

A system to gather the required information in a distributed manner has been described. This information is then disseminated using network-layer flooding and in-network aggregation. Two flooding schemes were evaluated, being the Slotted 1-Persistence approach and a modified Slotted 1-Persistence scheme which staggers the rebroadcast timers in all involved nodes by means of microSlots. It is shown that the microSlotted 1-Persistence scheme provides significant improvement over the Slotted scheme when used for flooding in a VANET:

- The time and number of hops required to traverse 10 km of highway is lower (100 ms VS.  $>1$  s).
- The delivery ratio (reachability) of the microSlotted scheme is higher ( $\sim 100\%$  VS.  $\sim 20\%$  at high densities).
- The microSlotted scheme suffers fewer collisions and has better utilisation of spectrum resources.
- The additional delay incurred per hop ( $T_{ms_{ij}}$ ) has negligible impact on end-to-end delay.

In short, the microSlotted 1-Persistence flooding is better at preventing broadcast storms. Together with the microSlotted 1-Persistence flooding scheme, the TrafficFilter system can deliver an accurate view on the road ahead, with minor temporal and spatial errors:

- The influence of mobility is small. Traffic is relatively slow moving w.r.t. communication. As a result, information received from the Virtual Horizon has not aged significantly upon arrival (at 10 km, the positional inaccuracy is at most a car-length).
- The sampling approach employed by the TrafficFilter retains important samples.
- Deviations in speed are in the order of several kilometers per hour and diminish as traffic dynamics decrease.

Choosing a good flooding strategy with correct settings (based primarily on topology and node density) can make

VANET flooding possible, even at high vehicle densities. The absence of communication-capable vehicles (gaps) is a problem of a more fundamental nature. It is expected that vehicles in the opposite lane, Vehicle-to-Infrastructure communication (V2I) or cellular technology can be used to bridge partitioned networks.

The TrafficFilter is designed to measure an accumulation of vehicles. As vehicle density increases, so does the probability a connected network exists that can carry this information upstream. This combination gives a good outlook on a practical application: when the need to communicate is there, so is the network.

Adaptation of the TrafficFilter to operate on multi-lane roads with junctions has been performed and is described and evaluated in a separate publication [5]. Parameterisation of the TrafficFilter based on actual data collected on highways is left as future work.

#### ACKNOWLEDGMENT

The authors wish to thank Ramon S. Schwartz for discussions on flooding in VANETs. This work is supported by the Dutch NL Agency/HTAS (High Tech Automotive Systems) Project Connect&Drive, Project no. HTASD08002.

#### REFERENCES

- [1] R. Pueboobpaphan and B. van Arem, "Understanding the relation between driver/vehicle characteristics and platoon/traffic flow stability for the design and assessment of cooperative adaptive cruise control," *Proc. Transportation Research Board 89th Annual Meeting, Washington DC, USA*, 2010.
- [2] C. van Driel and B. van Arem, "Traffic flow impacts of a congestion assistant," *Proc. of TRB 87th Annual Meeting, Washington DC, USA*, Jan 2008.
- [3] E. M. van Eenennaam and G. J. Heijenk, "Providing over-the-horizon awareness to driver support systems," *Proc. of 4th IEEE Workshop on Vehicle to Vehicle Communications (V2VCOM)*, June 2008.
- [4] N. Wisitpongphan, O. Tonguz, J. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 84–94, December 2007.
- [5] R. S. Schwartz, E. M. van Eenennaam, G. Karagiannis, G. Heijenk, W. Klein Wolterink, and H. Scholten, "Using v2v communication to create over-the-horizon awareness in multiple-lane highway scenarios," *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 998–1005, jun. 2010.
- [6] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft, and M. Steinbuch, "String-stable cacc design and experimental validation: A frequency-domain approach," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 4268–4279, Nov 2010.
- [7] E. M. van Eenennaam, W. Klein Wolterink, G. Karagiannis, and G. Heijenk, "Exploring the Solution Space of Beaconing in VANETs," *First IEEE Vehicular Networking Conference, VNC2009, Tokyo, Japan*, Oct 2009.
- [8] "IEEE 1609 - family of standards for wireless access in vehicular environments (WAVE)," <http://www.standards.its.dot.gov/>, Jan 2006.
- [9] S. Grafting, P. Mahonen, and J. Riihijarvi, "Performance evaluation of IEEE 1609 wave and IEEE 802.11p for vehicular communications," in *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on*, June 2010, pp. 344–348.
- [10] Task Group P, *IEEE 802.11p Draft 9.0*, Sept 2009.
- [11] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. New York, NY, USA: ACM, 2004, pp. 1–9.
- [12] S. Eichler, "Performance evaluation of the IEEE 802.11p WAVE communication standard," *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pp. 2199–2203, 30 2007-Oct. 3 2007.
- [13] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: A distributed mobile sensor computing system," *4th ACM SenSys*, November 2006.
- [14] A. Moukas, K. Chandrinou, and P. Maes, "Trafficopter: A distributed collection system for traffic information," in *CIA '98: Proceedings of the Second International Workshop on Cooperative Information Agents II, Learning, Mobility and Electronic Commerce for Information Discovery on the Internet*. London, UK: Springer-Verlag, 1998, pp. 33–43.
- [15] L. Wischoff, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "SOTIS - a self-organizing traffic information system," in *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, vol. 4, April 2003, pp. 2442–2446 vol.4.
- [16] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wirel. Netw.*, vol. 8, no. 2/3, pp. 153–167, 2002.
- [17] C. Zhu, M. Lee, and T. Saadawi, "A border-aware broadcast scheme for wireless ad hoc network," *Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE*, pp. 134–139, 5-8 Jan. 2004.
- [18] J. J. Blum and A. Eskandarian, "Avoiding Timeslot Boundary Synchronization for Multihop Message Broadcast in Vehicular Networks," *VTC Spring 2009*, 2009.
- [19] "IEEE std 802.11-2007 (revision of IEEE std 802.11-1999)," pp. C1–1184, June 12 2007.
- [20] OMNeT++, "discrete event simulation framework," <http://www.omnetpp.org>, 2010.
- [21] M. Löbbers and D. Willkomm, "A mobility framework for OMNeT++ user manual," Jan 2007.
- [22] 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.
- [23] "C2C-CC Manifesto: Overview of the C2C-CC system version 1.1," Car 2 Car Communication Consortium, Tech. Rep., 2007.
- [24] W. Ochieng and K. Sauer, "Urban road transport navigation: performance of the global positioning system after selective availability," *Transportation Research Part C 10*, pp. 171–187, 2002.