

Providing Over-the-horizon Awareness to Driver Support Systems

Martijn van Eenennaam and Geert Heijenk

e.m.vaneennaam@student.utwente.nl, geert.heijenk@utwente.nl

Department of Computer Science,
University of Twente, The Netherlands

Abstract—Vehicle-to-vehicle communications is a promising technique for driver support systems to increase traffic safety and efficiency. A proposed system is the Congestion Assistant [1], which aims at supporting drivers when approaching and driving in a traffic jam. Studies have shown great potential for the Congestion Assistant to reduce the impact of congestion, even at low penetration. However, these studies assumed complete and instantaneous availability of information regarding position and velocity of vehicles ahead. In this paper, we introduce a system where vehicles collaboratively build a so-called TrafficMap, providing over-the-horizon awareness. The idea is that this TrafficMap provides highly compressed information that is both essential and sufficient for the Congestion Assistant to operate. Moreover, this TrafficMap can be built in a distributed way, where only a limited subset of the vehicles have to alter it and/or forward it in the upstream direction. Initial simulation experiments show that our proposed system provides vehicles with a highly compressed view of the traffic ahead with only limited communication.

I. INTRODUCTION

VEHICLE-TO-VEHICLE (V2V) communication has great potential to make vehicular mobility more efficient and safer. Cooperating vehicles can share information on upcoming traffic conditions or planned trajectories, resulting in the orchestration of more efficient traffic flows. Various vehicle safety applications such as collision avoidance and cooperative driving are expected to result in a drop in collisions. This, in turn, will also benefit the efficiency.

A system proposed by Van Driel [1] looks promising in the effort to reduce traffic congestion on highways. It has been proven that—with a high enough degree of market penetration—a reduction of the number and effects of traffic jams is possible. This system is called the *Congestion Assistant*, a system onboard a vehicle which is based on knowledge of the situation on the road ahead. It is assumed in [1] that this knowledge is available and dependable, but no system for distributing this knowledge is proposed. We set out to devise such a system and explore possibilities.

The Congestion Assistant as described in [1] works by improving a driver's efficiency and performance in traversing a traffic jam, as illustrated in Fig. 1. The *Warning & Information* function informs the driver of

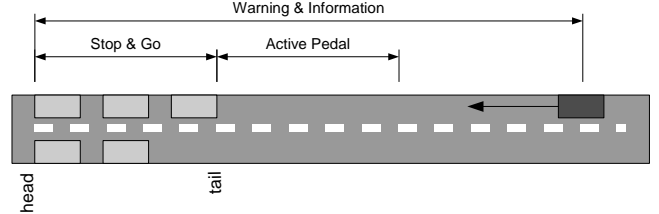


Fig. 1. A driver approaches a traffic jam

upcoming traffic situations. An *Active Pedal* function ensures a reduced inflow of vehicles at the tail of the congestion by gradually reducing the vehicle's speed. Because of the lower speed vehicles can maintain a closer following distance; as a result the stretch of road is used to effectively buffer the inflow of new vehicles into the congestion. The Active Pedal counters the unfavourable human behaviour of maintaining speed until stopped traffic is observed in front. This behaviour is found to result in a high inflow and high risk of accidents. The gradual reduction of speed implies less hard braking will occur, which benefits safety. Once a vehicle enters a congestion the *Stop & Go* function takes over longitudinal control of the vehicle, functioning as a type of Adaptive Cruise Control. The system maintains a close headway to the predecessor and—because there is no human in the loop—can react swiftly to sudden changes. Driver simulations carried out at TNO and microscopic traffic simulations using the ITS modeller [2] show great opportunities for reducing the impacts of congestion, even at low penetration (e.g. 10%) [1]. The Congestion Assistant acts upon knowledge of the position of the head and tail of a congestion.

Most V2V systems proposed in literature operate on close range, e.g. with only one or a few hops in mind. Examples are a system for cooperative driving at blind intersections [3], cooperative collision warning and avoidance systems [4], [5]. These systems function by creating awareness of each other by means of beacon messages, which contain such information as speed, acceleration, etc., and are sent at regular intervals. An abiding geocast approach is proposed by Yu and Heijenk in [6] to actively notify approaching vehicles of upcoming safety events. Such approaches are either not able to carry the required

information far enough upstream or are not detailed enough or impractical because the amount of information aggregated over several kilometers of highway grows rapidly. For instance, signalling a traffic congestion by means of abiding geocast messages could cause many such messages to coexist. If beacon messages were to include information of surrounding vehicles the communication system would easily clog up because of the vast amounts of data due to the number of vehicles present on several kilometers of highway.

It becomes clear we are facing two problems: a) how do we determine there is a traffic congestion and b) how do we distribute this information across several kilometers of highway, possibly several hundreds of vehicles? An approach called CarTel as described by Hull *et al.* in [7] highlights the use of vehicles as a mobile sensor network. Data is captured by many vehicles and collected in a centralised location. This information can then be retrieved from the CarTel portal.

Empirical studies concerning spatiotemporal traffic patterns such as presented by Kerner in [8] are carried out by means of several detector loops embedded in roads. Information can then be represented as a speed profile on a road at a certain time. Such information is primarily captured to study traffic behaviour. Traffic models such as the Intelligent Driver Model by Treiber *et al.* [9] can then be tuned to mimic this behaviour for simulation of the effects of, for instance, a planned additional lane.

We reason that a vehicle equipped with the Congestion Assistant could also benefit from such a speed profile of the road ahead, because all required information can be extracted from such a profile. The difference with existing work is that this information is both captured, processed and used by the vehicles without the need for any centralised authority. To facilitate this we propose a novel approach at building an over-the-horizon view using vehicle-to-vehicle communication.

The main contribution of this paper is the introduction of a system, called the TrafficFilter, in which vehicles collaboratively build a speed profile of the road using V2V communications. The profile, called TrafficMap, represents information needed by the Congestion Assistant efficiently, and can be built with minimal upstream communication.

The remainder of this article is structured as follows. In Section II we introduce the concept of the TrafficMap, the construct that contains the over-the-horizon view. We introduce some terminology and why vehicle-to-vehicle communication is the technology of choice for this application. Next, Section III introduces the process of constructing a TrafficMap by means of a threshold-based filtering technique called the TrafficFilter. In Section IV we discuss ongoing and future research, we conclude with Section V.

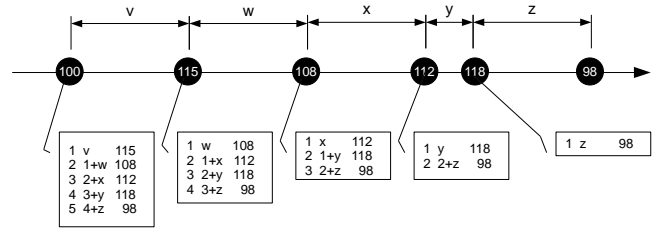


Fig. 2. Nodes on a line with a value denoting speed in km/h

II. OVER-THE-HORIZON VIEW BY MEANS OF V2V COMMUNICATIONS

A vehicle equipped with the Congestion Assistant needs to know the traffic situation several kilometers down the road. This distance is defined in [1] as five kilometers in advance of a traffic congestion, but preferably also includes the head of the jam possibly many kilometers further. Because of these great distances, the limited range of communication equipment, and the increase of interference when communicating over great distances, a V2V multi-hop communication approach is required.

It is reasoned by Jiang *et al.* in [10] that the messages produced by vehicles should be indicative; they cannot dictate how another vehicle must process and interpret a message. In line with this reasoning a vehicle publishes ‘information’, and not so much directives or commands for other vehicles.

To get a notion of the approach, vehicles are represented as tuples, composed of a means to denote a location plus the velocity of the vehicle at that location and the heading of the vehicle. A vehicle can form a representation of the road ahead, as sketched in Fig. 2. In this figure every vehicle moves on a straight line in the same direction so we will abstract from heading information. Every vehicle has knowledge of its predecessors, represented as a set of vehicle locations and velocities. We will refer to this collection of tuples as the TrafficMap. This TrafficMap gives a view on traffic flow speeds at certain locations down the road. The last entry of this list (5 in Fig. 2) defines the *virtual horizon*, the maximum defined distance captured in the TrafficMap. It is important this virtual horizon is close to or beyond the target virtual horizon as defined by the application.

If every vehicle were to add its information to the TrafficMap and (re)broadcast it, several problems would arise. Firstly, the potentially large number of vehicles on the road causes the list to grow rapidly, exceeding the maximum packet size. It is important the TrafficMap fits in one packet because no state will have to be maintained between consecutive packets and loss of one packet will not harm the information transferred in other packets. Secondly, the aggregate amount of data transferred would require a large amount of bandwidth, which needs to be shared with other applications or simply is not available, and thirdly, all these broadcasts would result

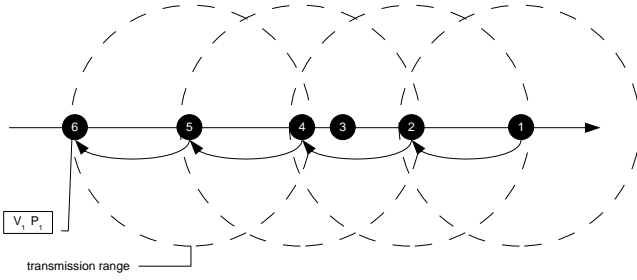


Fig. 3. From the perspective of node 6 (an observer) node 1 is a source (assuming 1 added an entry in the TrafficMap) and nodes 2, 4 and 5 are relay nodes. Node 3 is a latent node.

in heavy contention and many transmission collisions.

Several smart broadcasting techniques exist to combat the phenomenon known as *Broadcast Storm* [11] in MANETS or VANETS [12]. Such approaches rely on a means to suppress rebroadcasts by means of border awareness [13], location awareness or probability of rebroadcast [12], [11]. We reason that, besides using an efficient broadcasting technique that mitigates Broadcast Storms we should also limit the amount of data to only relevant data. This is what the TrafficFilter introduced in Section III does.

In the remainder of this article we will discuss *source*, *observer*, *relay* and *latent* nodes. The type of a node is based on the relevance of the information a node has and its role in the dissemination process. We define a source node (see Fig. 3) as a mobile node that broadcasts its own information (e.g. add an entry to the TrafficMap). An observer node is a (potentially) mobile node that receives this information. Relay nodes do not add information but merely pass it on. A latent node does not publish or relay information, but will receive it. The information functions as a means to observe traffic, hence any receiver is an observer. The moment an observer passes the information on it becomes a relay node itself if no new information is added, or a source node when it also injects its own information. The TrafficMap is disseminated by means of a geocast-like broadcast scheme that propagates against the flow of traffic, thus carrying information upstream.

In the next Section we will propose a means to collaboratively build a TrafficMap by all equipped vehicles on the road in an efficient way.

III. THE TRAFFICFILTER

The TrafficFilter ensures only relevant information is added to the TrafficMap. It is a system similar to Run-Lenght Encoding or Pulse Code Modulation with a variable hold time. The aim is to make a set of samples that best represents the actual speed-position relations on a road. Because a vehicle is influenced for a great deal by factors such as speed limit and other traffic many vehicles will show roughly the same speed, a relation often used in both macroscopic [14], [15] and microscopic

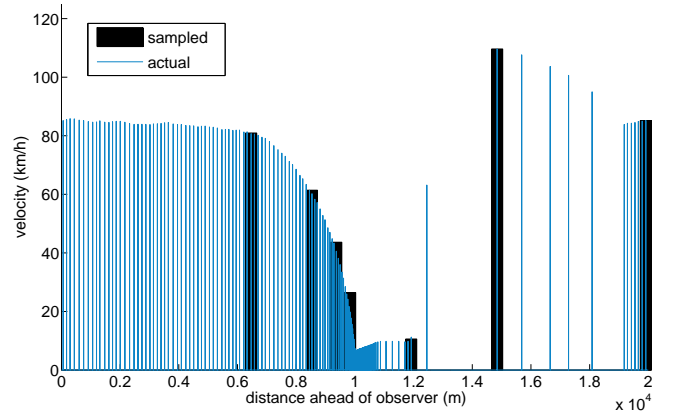
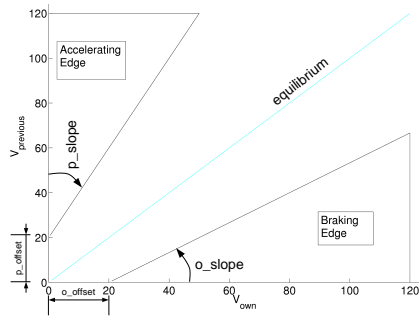


Fig. 4. TrafficMap created with Threshold Sampling approach

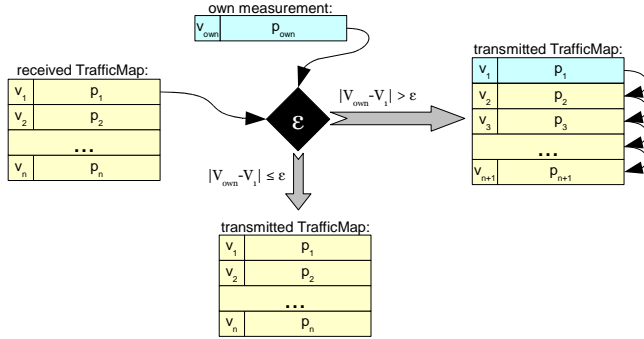
[9] traffic simulations. This means there will be a lot of similar speeds on our road, mostly clustered. We reduce this redundancy by only adding a sample to the TrafficMap when a certain deviation from the previous sample occurs. Fig. 4 shows a one-dimensional road, 20 kilometers in length, generated with the Intelligent Driver Model proposed by Treiber *et al.* in [9]. The IDM is used to illustrate the working of the TrafficFilter because it provides a quick way to obtain a detailed speed profile. Note that our implementation of the IDM is not calibrated to loopdetector data and is merely used for proof of concept. A vehicle's speed is mapped against its position on the road at a certain point in time, providing a snapshot of a stretch of road. The black bars are the samples in the TrafficMap from a vehicle approaching a traffic congestion, located at position 0.

When interpreting the TrafficMap, a sample remains valid until—in the direction *against* the flow of traffic—a new sample is added. Correlating vehicle position, speed and heading information with road map information allows a vehicle to deduct whether to add a sample and aids in interpreting the samples in the TrafficMap. In our description we will consider a TrafficMap for a single stretch of road, and hence focus on the position and speed information. The TrafficMap contains all the information the Congestion Assistant needs to know about upcoming traffic conditions: in the case of Fig. 4 the tail of the jam is located 10km away and the curve of braking vehicles can be interpolated. The head of the congestion is at 12km. The Warning & Information function can derive information such as the time to the jam, expected incurred delay and—when the vehicle is in the congestion—the progress within the congestion.

The TrafficFilter consists of three functions which operate on the TrafficMap: a capturing function ensures only relevant samples are added. An averaging function ensures a sample represents a small area on the road rather than a single vehicle. A reduction function removes redundancy in remote samples and removes those samples beyond the virtual horizon. These functions



(a) The threshold function ε



(b) Adding a sample to the TrafficMap

Fig. 5. Capturing a sample

work on the TrafficMap which we will now formally define.

Let TM be a set of tuples (v_i, p_i) where p is the position and v the velocity of vehicle i :
 $TM = [(v_1, p_1), (v_2, p_2), \dots, (v_n, p_n)]$ where $p_i \neq p_j$ and $1 \leq i \leq n$ and n denotes the number of tuples currently in the TrafficMap.

A. Sample Capturing

When a vehicle receives a TrafficMap it evaluates whether its speed (v_{own}) deviates *enough* from that of the last source node's speed ($v_{previous} = v_1$) and if so, add a new sample, i.e.,

$$TM_{new} = \begin{cases} [(v_{own}, p_{own}), TM_{old}] & \text{if } |v_{own} - v_1| > \varepsilon; \\ TM_{old} & \text{otherwise.} \end{cases} \quad (1)$$

Here ε is a function of the two speeds, which decides if the deviation between the two is large enough. It justifies adding a sample to the TrafficMap when the $(v_{own}, v_{previous})$ point falls in either of the two areas (accelerating edge, braking edge) shown in Fig. 5(a). This simple ε -function consists of two straight lines, the offset and slope of which can be configured. The equilibrium-line represents free-flowing traffic, most of the speed differences will be close to this line. It is deviation from this line which is of interest because the flow speed of traffic changes. The exact definition of the ε -function needs to be determined using simulation or field experiments.

Fig. 5(b) illustrates how a sample is added to the TrafficMap. When the own measurements—derived from onboard sensors or GPS—differ significantly from v_1 (as determined by the ε -function and (1)) the own measurement is *pushed* onto the stack and rebroadcast, i.e. the vehicle is a source node. If it does not differ the TrafficMap is either rebroadcast unaltered (relay node) or not broadcast at all (latent node), if upstream vehicles are known to rebroadcast the TrafficMap.

B. Sample Averaging

A single vehicle is responsible for adding 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 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 probably 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 ε -function.
- 2) The TrafficMap is rebroadcast.
- 3) A vehicle a short distance upstream receives it. Its ε -function does not allow it to add a new sample. It might, however, slightly alter the last entry (v_1, p_1) if it is within the averaging distance Δ .
- 4) The TrafficMap is rebroadcast.

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 Δ . The averaging is expressed in the following equation:

$$TM_{new} = \begin{cases} [(v_1^*, p_1), TM_{old}(2 \dots n)] & \text{if } \delta < \Delta; \\ TM_{old} & \text{otherwise,} \end{cases} \quad (2)$$

where

$$v_1^* = \frac{v_1 + (v_{own} \times \theta(\delta))}{1 + \theta(\delta)} \quad (3)$$

and

$$\delta = |(p_{own} - p_1)|. \quad (4)$$

In words, the resulting value v_1^* is composed of the previous value of v_1 (the velocity-component of the entry at index 1 in the TrafficMap) plus a weighted amount of v_{own} at distance δ from the location where v_1 was captured. The weighing is handled by $\theta(\delta)$ which is defined as follows:

$$\theta(\delta) = \left(\frac{\Delta - \delta}{\Delta} \right). \quad (5)$$

Eq. (5) gives a value between 0 and 1 for any δ between 0 and Δ , the averaging interval. Depending on Δ and the vehicle density a sample v_1 is made by one or multiple vehicles. Presently we use a set value for Δ but it could be directly based (i.e. inversely proportional) on the density of traffic, the effects can be researched.

Eq. (3) ensures $\frac{1}{1+\theta}$ of the original sample v_1 is carried on in v_1^* . The result is an average calculated over an a priori unknown number of values. The exact definition of (3) and (5) is still subject of research.

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 as discussed in Section III-A. In this case the difference of $120 - 80 = 40\text{km/h}$ should not be allowed to trigger adding a sample and the ε -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.

C. Reducing Redundancy

When a sample is captured it represents the actual situation at the current location. As soon as the information travels upstream, its relation to the actual situation diminishes, e.g. 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 so much detail and a summarisation can be performed to reduce the number of samples. Every node that rebroadcasts (either a relay or a source node) can perform such reduction operations with a couple of assumptions:

- Two consecutive remote samples that are somewhat the same (as defined by a threshold ω) can be reduced to one, the most remote one. The idea is that the confidence intervals overlap and the sample represents an area.
- 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.

Redundant samples generated because of a generous ε -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 ω depends on traffic dynamics, just like the ε -function used as a capturing threshold, but also on the distance to the current node as confidence intervals increase with the distance.

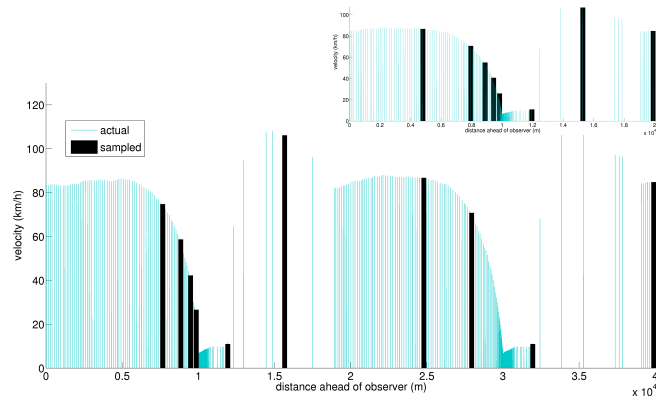


Fig. 6. A TrafficMap is passed upstream. Some redundancy is removed along the way

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. This is illustrated in Fig. 6. Here the original stretch of road is presented together with its representation twice as far away on the same road. The former is placed to align with the latter. As can be seen, the observer of the TrafficMap on the bottom sees two traffic jams up ahead. The observer of the top TrafficMap only sees one (and has probably just passed the other one). Note that the top TrafficMap's 8 samples have been reduced to 4 in the bottom TrafficMap, without too much loss of detail.

The reduction step will also remove samples in the following two cases:

- 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 horizon. 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.

The Congestion Assistant acts upon information from kilometers away. As such it is not a delay-critical application and should leave enough bandwidth for other, more delay-critical applications [16].

IV. ONGOING AND FUTURE WORK

In Section III we have introduced a system in which vehicles collaboratively build a TrafficMap. This is an efficient representation of the road traffic on a certain stretch of road. The communication protocols for actually distributing a TrafficMap still need to be defined but a hint of the direction is provided here.

Initial simulation experiments, such as those used to generate Fig. 4 and Fig. 6, show that our proposed

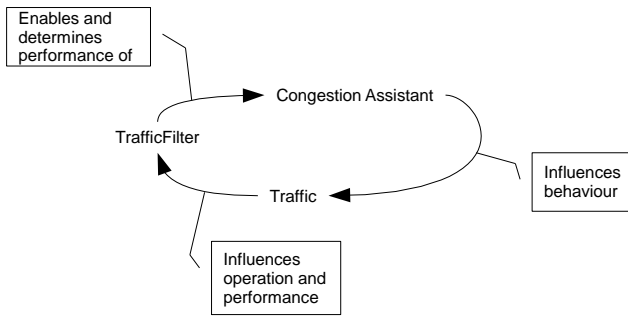


Fig. 7. Interplay between Traffic, TrafficFilter and Congestion Assistant

system provides vehicles with a highly compressed view of the traffic ahead with only limited communication. Future work will include a detailed simulation study to evaluate the performance of the system. We also plan to study the impact of communication system performance (packet loss, latency) and data reduction on the performance of the Congestion Assistant.

It is expected that, to deal with the great variation of dynamics in highway traffic, the thresholds and configuration variables used by the TrafficFilter should be dynamic as well. When the dynamics of traffic are such that a lot of samples are generated, thresholds might be adjusted. When vehicle density decreases, the averaging interval Δ might be increased. A target virtual horizon needs to be defined (as required by the application) and thresholds need to be set to achieve this.

The Congestion Assistant and the TrafficFilter function in a highly dynamic environment; that of highway traffic. The performance of the TrafficFilter depends on the presence of vehicles and hence on traffic, as illustrated in Fig. 7. This traffic, in turn, is influenced by the Congestion Assistant. The Congestion Assistant acts upon information provided by the TrafficFilter. These dependency issues are all for further study.

The TrafficFilter relies on the presence of instrumented vehicles on the road. Using a realistic communication range of 200–300 meters [17], [6] it is quite possible there are gaps in the network. A means to overcome such a gap is by allowing vehicles on the opposite lane to carry this information.

So far only the spatial side of the TrafficFilter has been described: using a snapshot at a point in time the algorithms of the capture, average and reduction functions have been introduced. Here we will briefly introduce the temporal side of the TrafficFilter, which is still topic of research.

A vehicle enters a highway and it listens for ongoing TrafficMap communications. When TrafficMap communications are observed the vehicle participates, otherwise it initiates TrafficMap communications. There are three factors here: firstly, we need to ensure a certain number of messages per time unit to keep the system alive and allow newcomers to engage, secondly we must ensure a

received TrafficMap is passed on swiftly and, lastly, observed deviations in speed must be rapidly communicated upstream.

A timer can be used to ensure a maximum time between TrafficMap communication is satisfied. If no TrafficMap has been received for a certain period the vehicle decides to broadcast. When a TrafficMap is received from downstream (thus ahead in traffic) the information is used by the capture, average and reduction functions and the TrafficMap is sent for transmission using a distance aware (slotted p-persistent) flooding scheme modified from that described by Wisitpongphan *et al.* in [12]. This scheme propagates TrafficMap disseminations upstream, favouring rebroadcast by nodes further away to minimise the number of hops over great distances and achieve low latency. The exact detail of the flooding scheme is subject to further research but the idea is that on a highway several TrafficMap floods are being passed upstream simultaneously (at geographic disjoint locations). The system ensures a minimum of such floods but also a maximum by means of collecting and summarising several TrafficMaps received in rapid succession.

V. CONCLUSIONS

The Congestion Assistant described by Van Driel [1] provides a means to alleviate the effects of traffic congestion. It is recognised that a system such as the Congestion Assistant will only work if it is backed by an adequately provisioned communication service provider. We believe that a distributed ad hoc solution using multi-hop V2V communications is the best way to see to the communication needs of the Congestion Assistant system. A possible solution has been presented in this article. It consists of a TrafficMap that only contains essential velocity information, and that is built in a distributed and collaborative way using a so-called TrafficFilter. The follow-up research will evaluate if this approach is a viable one.

REFERENCES

- [1] C. van Driel, "Driver support in congestion - an assessment of user needs and impacts on driver and traffic flow," Ph.D. dissertation, University of Twente, Nov 2007.
- [2] E. Versteegt, G. Klunder, and B. Van Arem, "Its modeller - irvin wp1.1," TNO Mobility and Logistics, Delft, The Netherlands, Tech. Rep. Report 2005-16, 2005.
- [3] L. Li and F.-Y. Wang, "Cooperative driving at adjacent blind intersections," *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 1, pp. 847–852 Vol. 1, 10–12 Oct. 2005.
- [4] T. ElBatt, S. K. Goel, G. Holland, H. Krishnan, and J. Parikh, "Cooperative collision warning using dedicated short range wireless communications," in *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*. New York, NY, USA: ACM, 2006, pp. 1–9.
- [5] 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.

- [6] Q. Yu and G. Heijenk, "Abiding geocast for warning message dissemination in vehicular ad hoc networks," *to be published in Proceedings of the IEEE Vehicular Networks and Applications Workshop 2008 (VehiMobi 2008)*, 2008.
- [7] 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," in *4th ACM SenSys*, Boulder, CO, November 2006.
- [8] B. S. Kerner, "Empirical macroscopic features of spatial-temporal traffic patterns at highway bottlenecks," *Phys. Rev. E*, vol. 65, no. 4, p. 046138, Apr 2002.
- [9] 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.
- [10] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Hertrich, "Design of 5.9 GHz DSRC-based vehicular safety communication," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 13, no. 5, pp. 36–43, October 2006.
- [11] 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.
- [12] 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 [see also IEEE Personal Communications]*, vol. 14, no. 6, pp. 84–94, December 2007.
- [13] 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.
- [14] D. Helbing and M. Treiber, "Gas-kinetic-based traffic model explaining observed hysteretic phase transition," *Phys. Rev. Lett.*, vol. 81, no. 14, pp. 3042–3045, Oct 1998.
- [15] D. Helbing, "Improved fluid-dynamic model for vehicular traffic," *Phys. Rev. E*, vol. 51, no. 4, pp. 3164–3169, Apr 1995.
- [16] S. Rezaei, R. Sengupta, and H. Krishnan, "Reducing the communication required by DSRC-based vehicle safety systems," *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 361–366, Sept. 30 2007–Oct. 3 2007.
- [17] F. Bai and H. Krishnan, "Reliability analysis of DSRC wireless communication for vehicle safety applications," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 355–362, 2006.