Constrained Geocast to Support Cooperative Adaptive Cruise Control (CACC) Merging

Wouter Klein Wolterink, Geert Heijenk, Georgios Karagiannis
Department of Computer Science, University of Twente, The Netherlands
{w.kleinwolterink, geert.heijenk, g.karagiannis}@utwente.nl

Abstract—In this paper we introduce a new geocasting concept to target vehicles based on where they will be in the direct future, in stead of their current position. We refer to this concept as *constrained geocast*. This may be useful in situations where vehicles have interdependencies based on (future) maneuvers. We have developed a first version of such a protocol in the context of an automated merging application, and tested it using simulations. Results show that the protocol is able to meet the requirements of such applications. Compared to a common geo-broadcast protocol this protocol becomes more reliable as road traffic densities increase, but in other aspects the performance is so far lacking. Based on our experiences with implementing the protocol however we see plenty of room for further improvement.

Index Terms—Cooperative Adaptive Cruise Control (CACC), geocast, V2V, VANET

I. INTRODUCTION

Cooperative Adaptive Cruise Control (CACC) is a traffic efficiency system in which the speed of vehicles is automatically controlled in a cooperative matter using vehicular communication. Because of the short reaction time of CACC compared to human drivers, vehicles can drive relatively close together (<1s) forming so-called *platoons*. They are also able to react significantly faster to speed changes of their preceding vehicles, minimizing unnecessary braking and accelerating and in this way increasing overall traffic flow stability (see [1]). The goals of CACC include increasing the capacity of the road network and decreasing vehicle emissions. We are currently involved in the Connect & Drive project which' goal is to design, implement and test a CACC system using short-range vehicle-to-vehicle (V2V) communication.

A requirement of any CACC system is that it must be able to support the merging of vehicles inside an existing platoon. Vehicles may for instance simply wish to join a platoon, or are forced to do so at a road narrowing or at a merging junction (see Fig. 1). The first and the second case concern situations in which vehicles drive in adjacent lanes, and can be executed using direct one-hop communication: when a
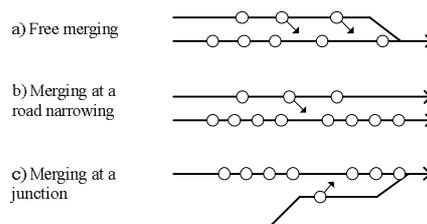

Figure 1. Different types of merging. The fat lines are road lanes, the arrows denote the direction of travel.

vehicle inside a platoon receives a request from another vehicle to join the platoon it will create a so-called *merging gap* by gradually decreasing its speed, thereby increasing the headway to its preceding vehicle. When the merging gap is large enough the merging vehicle aligns with it and joins the platoon. Afterwards normal CACC operation is resumed.

Creating a merging gap and aligning with it takes time, and this becomes problematic in the case of merging at a junction. Especially in metropolitan areas the merging vehicle will not be able to communicate directly with the platoon in which it wants to join until it has reached the merge area. Taking the length of the merge area into account, as well as realistic vehicle speeds, this will often not allow the merging vehicle to join the platoon. For this reason we require a communication system that is able to warn any vehicle inside a platoon in advance, using indirect multi-hop communication, that it needs to create a merging gap for a merging vehicle at a junction.

(Abiding) geocast is a form of routing in which messages are routed through a network based on spatiotemporal *constraints*. Normally these constraints directly specify the destination region and destination time span, see Section II. In our case however the constraint is different, since the location of the vehicle that must create a gap for the merging vehicle is unknown. Stated generically the constraint that is used by the geocast protocol is as follows:

"Any node[1] that will be within a certain region during a certain future time period is a destination node and should receive a message."

---

[1] In this paper a node is defined as a network-entity, so any communication-capable vehicle is also a node.

Here the region being the merge area, and the time period being the period when the merging vehicle will reach this area.

We refer to this communication concept as *constrained geocast*. We believe that this may prove a valid geocast approach for types of intelligent traffic systems that wish to target approaching vehicles, e.g., warning applications or traffic information applications. In this paper we evaluate a protocol that is able to perform constrained geocast in the context of our merging application.

This paper presents our research so far on constrained geocast within the context of our CACC merging application. We have developed a first version of the protocol and have evaluated its performance by means of simulation. The main research questions that are answered by this paper are:

1) Can the constrained geocast protocol be used to reliably warn platoon vehicles to create a gap?
2) How is the performance of the constrained geocast protocol affected by different network densities and traffic scenarios (i.e., the number of lanes on a road)?
3) Can the constrained geocast protocol outperform a common geo-broadcast protocol?

The remainder of this paper is organized as follows. In Section II some background is given on vehicular networking, automated merging and geocast. Section III presents the merging application and fully specifies its spatiotemporal routing constraint. Section IV presents our constrained geocast protocol and discusses its distinctive traits. The performance evaluation of the protocol and its results are discussed in Section V. We give our conclusions in Section VI, along with our directions for future work.

II. RELATED WORK

Leading European projects in the field of vehicular networking include COMeSafety[2], SAFESPOT[3], GeoNET[4] and CVIS[5]. We use COMeSafety's European ITS VANET Protocol (EIVP) [11] as network level protocol, and based our geocast protocol on SAFESPOT's positioning interface [3].

Automated driving, especially in platoon formation, has long since been subject of research, see [1], [8], [9]. Research on merging maneuvers within platoons can be found in e.g., [8] and [10]. However, their goal was to optimize the merging procedure from the point of the merger's benefits. Our approach focuses on the realization of a merging manoeuvre where the disturbances on the highway are minimized.

Geobroadcast (or geocast, see e.g. [6], [7]), supports the dissemination of information in a larger geographical area. The sender of the information defines the geographical area where the data message should be disseminated and attaches it to the message. Information is sent once. In contrast,

abiding geocast [4] is a dissemination approach where the information is geocasted to all nodes that are inside a destination region within a certain period of time. We generalize these principles into constrained geocast, as explained in Section I.

III. THE MERGING APPLICATION

We consider a CACC system in which communication is fully based on the periodical (1-10 Hz) broadcast of network-level EIVP beacons using 802.11p [12]. Our constrained geocast protocol is therefore also constrained to the use of these beacons, although their timing may be altered.

Earlier research on automated merging focused on adapting the merging vehicle's speed to match existing gaps in the flow of freeway traffic. However, since the goal of our CACC system is to have vehicles drive in platoon-wise fashion, with little room in between individual vehicles, such an approach may cause situations where merging vehicles will not be able to find a gap to merge in. More important even, since we assume that our system must work in an environment where vehicles are a mix of automated and non-automated vehicles, our system needs to be able to cope with non-automated vehicles. For these reasons we have chosen to adapt the speed of the approaching freeway vehicles, to ensure that a merging vehicle will always have room to merge. We refer to the room needed as the required *merging gap*. Later on we may add functionality, so that when the merging vehicle is similarly automated its speed may also be controlled.
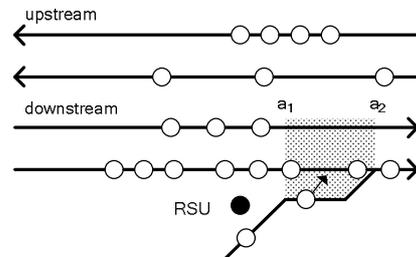

Figure 2. The merging scenario. The arrows denote the direction of travel: we refer to the lower lanes as the downstream lanes and the upper lanes as the upstream lanes. The merge area has been shaded and runs from a_1 to a_2.

Figure 2 depicts our merging scenario, showing (in this case) a four-lane freeway (two downstream and two upstream lanes) and a merging ramp connected to the slowest of the downstream lanes. The merge area has been shaded, and its starting point and end point are referred to as a_1 and a_2, respectively. Note that we consider all downstream lanes as part of the merge area. An RSU is located in such a way that it can sense and (if applicable) communicate with approaching merging vehicles, and communicate with any freeway vehicles up to a certain distance away.

The RSU continually senses the merging ramp for approaching merging vehicles. For every sensed merging vehicle the RSU estimates:

1) the moment when the merging vehicle will reach the start of the merge area a_1 (denoted as $t_{a,1}$);
2) the moment when the merging vehicle will reach the end of the merge area a_2 ($t_{a,2}$);
3) the required size of the merge gap within a platoon to let the merging vehicle merge safely.

Sensing and measuring may be performed in any way, as long as the RSU is able to calculate the described estimates. These estimates, together with some other information, are bundled together in a message called a *merge request* (MR). Every time the RSU senses the merging ramp it discards any previous MRs and creates a MR for every approaching merging vehicle. The information contained in a MR fully defines (i) the position of the merging vehicle during the period [$t_{a,1}$, $t_{a,2}$] that it is in the merge area, and (ii) the position and size of the required merging gap during that period. This is also illustrated in Fig. 3. Note that we assume that the merging vehicle has constant speed during that period, so that its position within the merge area can be calculated as:

$$s_M(t) = a_1 + (a_2 - a_1) \cdot (t_{a,2} - t_{a,1}), \qquad (1)$$

for $t \, \varepsilon \, [t_{a,1}, t_{a,2}]$. Table I shows the entire structure of a MR.

TABLE I
STRUCTURE OF THE MERGE REQUEST

Name	Type	Description
id	identifier	Identifies a merging vehicle
t_{create}	timestamp	Set upon creation of the MR.
a_1	location (GPS coordinates)	Start of the merge area.
a_2	location (GPS coordinates)	End of the merge area.
$t_{a,1}$	timestamp	Moment when the merging vehicle is estimated to reach a_1.
$t_{a,2}$	timestamp	Moment when the merging vehicle is estimated to reach a_2.
μ	distance (m.)	Half the required merging gap.

During the entire merging procedure a merging vehicle will have the same unique *id* in its MRs, so that vehicles can be tracked independently. Field t_{create} is used to indicate the freshness of the MR and is set to the moment of creation. The MR has a restricted lifetime after which it should be discarded; we have taken this lifetime to be a system parameter and have therefore not included it in the MR. Fields a_1, a_2, $t_{a,1}$, and $t_{a,2}$ are set as described before. One should note that the values for $t_{a,1}$ and $t_{a,2}$ are estimates which contain some margin of error. This margin is higher when the merging vehicle is further away.

The value of field μ is still subject to research but has two goals: to define the width of the required merging gap, and to account for the margin of error introduced by the estimates $t_{a,1}$ and $t_{a,2}$. Eq. (2) shows the resulting calculation performed by the RSU:

$$\mu = \frac{l}{2} + h \cdot v_{freeway} + f_i \cdot v_M \cdot (t_{a,1} - t_0) \cdot \qquad (2)$$

Parameter l (in meters (m)) is the width of the vehicle, h (s) the standard CACC time headway, $v_{freeway}$ (m/s) the speed of the freeway traffic (measured by the RSU), f_i (unitless, $0 \le f_i \le 1$) the inaccuracy factor, $t_{a,1}$ (s) the estimated arrival time of the merging vehicle at a_1, and t_0 (s) the current time. The first two parts of the equation account for the required merging gap; the last part for the margin of error. Note that the value for μ increases as $t_{a,1} - t_0$ increases: the further away the merging vehicle is, the larger the expected error is, and therefore also μ.

Once the RSU has created a MR for every merging vehicle it bundles all MRs into a single Merge Message (MM) and geo-casts the MM using the following constraint:

"A freeway vehicle is part of a MM's destination set if for at least one merging vehicle (whose MR has been included) the vehicle is estimated to be within μ meters of that merging vehicle during the period [$t_{a,1}$, $t_{a,2}$]."

Since we want our geocast protocol to work in a distributed manner, freeway vehicles must themselves be able to decide whether they are estimated to fall within μ meters of a merging vehicle during the period [$t_{a,1}$, $t_{a,2}$]. For this we use the formula of linear motion with zero acceleration to calculate a freeway vehicle's future position on the road:

$$s_F(t) = s_0 + v_0 \cdot (t - t_0), \qquad (3)$$

where t_0 (s) is the current time, s_0 (m) is the vehicle's current location, and v_0 (m/s) its current speed. A vehicle F is thus part of a MR's destination set if the following holds:

$$\left| s_F(t) - s_M(t) \right| \le \mu \quad t \in \left[t_{a,1}, t_{a,2} \right], t_0 \le t_{a,2} \cdot \qquad (4)$$

Figure 3 illustrates how both the position of the merging gap and the expected trajectories of a freeway vehicle are a function of time.

A geocasted MM will be forwarded by nodes in the upstream direction until it has reached all destination nodes. When a destination CACC vehicle receives a MM, the message is forwarded to its CACC module. The CACC module will then take action to ensure that the merging vehicle will have a gap to merge in. What kind of action needs to be taken is out of scope here, and is still being researched. The CACC module will only take action if it has valid information about approaching merging vehicles; if the MR times out and no new MR has arrived, it will revert back to normal CACC operation. For an approaching vehicle to be able to continuously track a merging vehicle it therefore needs to receive a continuous stream of MRs (and thus MMs), each containing fresh information and obsoleting any previously received MRs. A destination vehicle should therefore receive a new MR before the previous MR becomes

invalid. We are currently researching the minimum amount of time that a destination vehicle should receive its first MR before the gap has to be ready, as well as the minimum distance. So far we have no reliable numbers on these minima but initial calculations show that the use of multi-hop communication will be required.
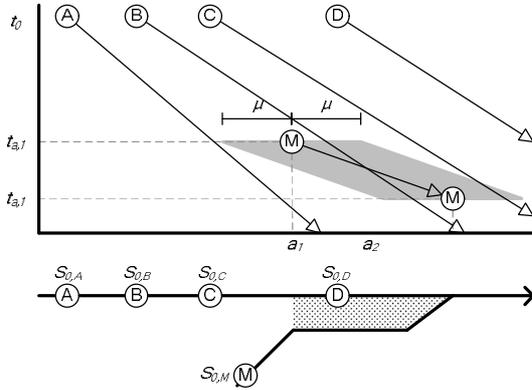

Figure 3. A spatiotemporal diagram of the required merging gap (the parallelogram) and the expected trajectories of freeway vehicles A-D, calculated at t_0 using Eq. (3). Nodes B and C are destination nodes, since they cross the parallelogram.

IV. CONSTRAINED GEOCAST

In this section the constrained geocast protocol, designed in the context of a CACC merging application, is described.

A. The Message Format

As was already mentioned in Section III we use the EIVP message format, see [11]. Similar to the CACC messages, a MM (which is simply a set of MRs) is encapsulated as a transport PDU (TPDU) in the EIVP network layer packet. The size of a MR is about 50 bytes. We neglect the TPDU overhead (which is only a few bytes), so the size of a MM in bytes is the number of enclosed MRs times 50.

As Eq. (4) shows a vehicle only needs it current position and speed (and the current time) to calculate whether it is part of the destination set of a given MR. Since this information is also included in the network header of the regularly beaconed EIVP messages, a node also knows for each of its one-hop neighbours whether they are part of the destination set.

B. The Destination Set

In this section we analyse the destination set of our constrained geocast protocol. It will be shown that this set is much more dynamic than is the case with regular geocast. We first focus on the destination set of a single MR.

Due to Eq. (4) destination vehicles that move at significantly different speeds are by definition geographically dispersed, so there is no homogenous destination region. Destination vehicles and non-destination vehicles may be mixed together. Vehicles can furthermore at any moment enter, leave, or re-enter the destination set, depending on their own behaviour (i.e., speed profile). They must therefore continuously check whether (4) holds for any recently received MR. Other nodes in the network see the status of a node as it was last beaconed, and will therefore only react to a node's changed status if they receive a beacon that proclaims these changes. Nodes should store received messages at the network layer for as long as they are valid, so that they may immediately be passed up to the application layer when the node joins its destination set. Although this is not a scalable approach when the number of messages (and their respective lifetimes) rises, we do not foresee any such problems within the scope of our merging application.

To explain some of the dynamicities of the destination set suppose that all downstream traffic travels at a constant speed. Let the *destination region* be the region in which every vehicle that travels at this specific constant speed is part of the destination set of a given MR. As time progresses the destination region will linearly move downstream (towards the merge area) until it reaches the region $[a_2 - \mu, a_2 + \mu]$ (assuming the MM hasn't timed out by then). Should the speed of the traffic suddenly become higher, then the destination region will move upstream, since downstream traffic will be able to travel a greater distance during the period $[t_0, t_{a,2}]$. Likewise the destination region moves downstream if the speed of the traffic decreases.

Now suppose that the RSU repeatedly sends out new MRs for a certain merging vehicle, each MR obsoleting the previous one. If the merging vehicle behaves as was estimated beforehand the estimations for $t_{a,1}$ and $t_{a,2}$ will remain the same for each consecutive MR. The value μ will become smaller with each MR however, since $t_{a,1}$-t_0 in Eq. (2) will linearly decrease. As a result of this, the destination region will also become smaller with each update (due to Eq. (4)). Figure 4 illustrates how the destination region nears the merge area as time progresses, and how it becomes smaller as well.
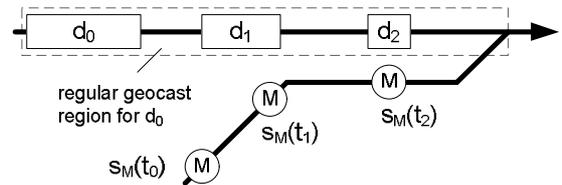

Figure 4. The merging vehicle and the resulting constrained destination regions at different time steps. The regions become smaller with every time step because μ decreases.

If the merging vehicle does *not* behave as was previously estimated then the destination region will shift either downstream or upstream. If the new estimate predicts that the merging vehicle will arrive at the merge area sooner than was previously expected then the destination region will move downstream. The region moves upstream if the merging vehicle arrives later.

Although we have now continually used the term destination regions, keep in mind that the RSU still only sends out a MR (without any specific addressing), and that the nodes forward the MR in a distributed fashion towards any existing destination nodes. Destination nodes will often lie outside the transmission range of the RSU and can therefore not be directly addressed.

Should the RSU wish to disseminate a MR using normal geocast then it would have to use a destination region that covers the entire area from a_2 up to a distance upstream of the merge area that a vehicle travelling at the maximum speed limit can cover in the period $[t_0, t_{a,1}]$. This has also been illustrated in Fig. 4. The normal geocast destination set will always be bigger than our constrained destination set: to ensure that everyone is reached the RSU will target a region that may be larger than necessary, while in our constrained approach the message will only be forwarded upstream as far as is needed (see Section IV.B). Combined with a more restricted mode of forwarding outside the destination set, the aim of our protocol is to increase efficiency. This effect increases as the average speed decreases while the speed limit remains the same; a typical dense road traffic scenario. In this scenario the normal geocast destination region will overshoot considerably. As the number of merging vehicles increases however the efficiency of the constrained geocast protocol decreases, since in that case multiple constrained destination sets are spread out over the downstream road, see Figure 5.
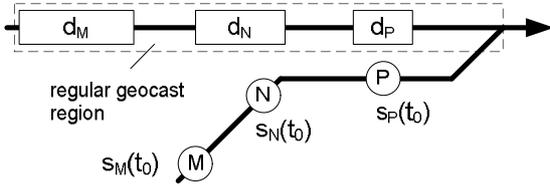

Figure 5. The merging vehicles M, N, and P and their resulting constrained destination regions at t_0.

By definition nodes are either inside or outside the destination set of a MR. If they are outside the set they have a certain position to the set. A node that reaches the merge area too soon to fall inside the MR's merging gap parallelogram is said to be *in front of* the destination set, and a node that reaches the merge too late *behind* the destination set. Nodes know their own position to the MR destination set (inside, in front of or behind) and that of their neighbors (based on the most recently received beacon). In Figure 3 node A is behind the destination set, nodes B and C are inside the destination set, and D is in front of the destination set.

A node is part of a MM's destination set if it is part of the destination set of at least one of the included MR's.

C. Constrained geo-routing

In this section the routing protocol to support our constrained geocast is fully specified. We consider the routing of a MM containing multiple MRs. Forwarding of MMs is only performed by including it in a node's beacon, which is then broadcasted. Beacons are never rescheduled. For two vehicles x and y that are driving in the downstream direction toward the merge area, with x situated closer to the merge area than y, x is said to be in front of y, and y is said to be behind x.

We assume that nodes beacon at such a frequency that two nodes that drive in opposite directions on opposite lanes (downstream vs. upstream) should each receive multiple beacons as they pass each other (ignoring lost beacons due to transmission failures).
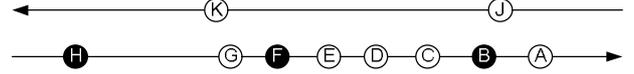

Figure 6. Nodes travelling downstream (lower lane) and upstream. The merge area has been left out. The dark nodes are destination nodes. Downstream travelling nodes can only communicate with the nodes directly in front or behind. Nodes J and K can respectively communicate with B and G.

Figure 6 is used as a reference situation. For the sake of the example we assume a constant speed for all the downstream travelling nodes in the figure.

Routing when travelling upstream
When a node x that is travelling upstream receives a valid (i.e., its lifetime has not expired) MM it will store the MM for as long as it is valid, unless it gets replaced by a newer MM. Node x will not include the MM in one of its beacons *unless* x has a one-hop neighbor that is a destination node and that did not include the MM in the beacon that was most recently received by x. See for example nodes J and K in Fig. 6: they will only include a MM if they just received a beacon from either B, F, or H that did *not* include a MM.

Routing when travelling downstream
Now consider a node x travelling downstream that receives a MM m. We define the counter R as a repeat counter, and R_{MAX} as the upper limit for that counter. R is set to 0 for each newly received MM.

If x is a destination node of m then x will include m in all of its beacons as long as m is valid, to indicate that it has received m. See nodes B, F, and H in Fig. 6.

If non-destination node x has a one-hop destination node that did not include m in the beacon most recently received by x, then x will include m in its next beacon. See e.g. nodes A and C in Fig. 6: they will include m in their next beacon if they just received a beacon from B that did not include m.

If (i) non-destination node x is behind the destination set of at least one of m's MRs, and (ii) has at least one downstream travelling neighbour driving in front that is also positioned behind at least one MR, and (iii) x has not yet received m from any downstream travelling vehicle driving in front, and (iv) $R < R_{MAX}$, *then* x will include m in its next beacon and increase R with 1. This situation would occur in Fig. 6 if node D receives m from node E, while it hasn't yet received m from

anyone in front.

If (i) non-destination node x is in front of the destination set of at least one of m's MRs, and (ii) has at least one downstream travelling neighbour driving behind it that is also positioned in front of at least one MR, and (iii) x has not yet received m from any downstream travelling vehicle driving behind it, and (iv) $R < R_{MAX}$, *then* x will include m in its next beacon and increase R with 1. This situation would occur in Fig. 6 if node D receives m from node C, while it hasn't yet received m from anyone behind.

If (i) non-destination node x is in front of the destination set of at least one of m's MRs, and (ii) x has not yet received m from any downstream travelling vehicle driving behind it, and (iii) $R < R_{MAX}$, *but* x has no downstream travelling neighbours behind it (see node G in Fig. 6), *then* x is the last of a group of downstream travelling nodes and it has the responsibility to get the message to the next group of downstream nodes. It does this by including m every n^{th} beacon, where n is chosen such that any upstream travelling node that passes x at max speed should receive at least 2 beacons containing m (not considering transmission errors). A similar method is used in abiding geocast to determine the repetition rate of geocast messages being broadcasted, see [4]. The value of n is calculated as follows:

$$n = \left\lfloor \frac{\lambda_q \frac{d}{v_{MAX}+v_x}}{N} \right\rfloor = \left\lfloor \frac{\lambda_g d}{N(v_{MAX}+v_x)} \right\rfloor, \quad (4)$$

with N (unitless) being the number of beacons one wants to receive (here 2), v_{MAX} (m/s) the speed limit of the upstream lane, v_x (m/s) the speed of x, λ_g (Hz) the beacon generation rate (assumed to be equal for all nodes), and d (m) the maximum distance that two nodes driving on opposite sides of a road will be in each other's transmission range assuming a straight road and equal transmission ranges (here set to 480 meters, which is about twice the expected transmission range).

V. Performance Evaluation

In this section the performance evaluation of our constrained geocast protocol – in the context of the CACC merging application described in Section III – is presented. Evaluation was done using simulations. The simulation model is described in Section V.A. We have also implemented and analyzed a simple geo-broadcast protocol to support our merging application, see Section V.B. The simulated scenarios and measured metrics are described in Section V.C and V.D, respectively. Finally Section V.E discusses the simulation results.

A. The simulation model

We have implemented our protocol in the OMNET++/MiXiM[6] network simulator, versions 4.1 and 1.2 respectively. To model the behaviour of the 802.11p model as close as possible we have altered the IEEE 802.11 medium access module in such a way that all parameters follow the 802.11p specification [12], although details in the 802.11p physical layer (OFDM) were abstracted from. Signal propagation was modeled using MiXiM's simple path loss model, with α set to 3.5, the center frequency set to 5.87 MHz, and the SNR threshold set to 0.1259. In the MAC layer we set the access category to 0. Transmission power was set to 50 mW (which gives a range of about 250 m), propagation delay was ignored, and thermal noise was set to -110 dBm. All switching times were set to 0. Network- and transport layer packet overhead was set to 3200 bits, including security overhead.

Mobility was modeled using the intelligent driver model (IDM, see [13]), which we have implemented in MiXiM. Mobility spacing was set to 6.66, acceleration to 0.73 m/s^2, deceleration to 1.67 m/s^2, and the acceleration exponent to 4. Vehicle lengths were set to 4 m and nodes were updated every 0.1 s. The desired velocity of vehicles was adapted to meet the speed limits of a road lane.

B. Simple broadcast protocol

The simple broadcast protocol is fully defined by its destination region and some timing parameters. The start of the destination region is defined as the distance upstream of the start of the merge area that a vehicle traveling at the maximum allowable speed limit can cover in 20 s. The end of the destination region is equal to the end of the merge area.

Every node inside the destination region that receives a beacon containing a new (i.e., never seen before) MM will include this MM in the next beacon it transmits. The transmission time of the next beacon is defined as follows:

$$t_{i+1} = \max(t_r, t_i + T_{\min}) + T_u, \quad (4)$$

where t_r is the moment when the MM was received, t_i is the moment the previous beacon was sent, T_{min} (s) is the required minimum inter-beacon period and T_u (ms) is a random value drawn from the range [0,10] according to a uniform distribution. Based on [6] we have set T_{min} to 100 ms.

C. Simulation scenarios

Figure 6 shows the different layouts of our traffic simulation scenarios. One, two or four lanes of 4km length were modeled with an RSU next to it at the start of the merge area, and a merge area of 100 m. length. The merging lane and merging vehicle were abstracted from: the RSU continually generated virtual merging vehicles with an estimated arrival time at the merge area. At any time there were 4 approaching merging vehicles: every 7 s a new vehicle was generated, that arrived at the merge area between 15-20 s later. The speed of a merging vehicle in the merge area is

uniformly chosen from [v_{MAX}-10, v_{MAX}+10], where v_{MAX} is the maximum speed of the slowest downstream lane.

For the single lane and two lanes scenarios the densities vary in the range {5,10,15,20,25} vehicles/km/lane, and the maximum speed limit was varied between 80 and 120 km/h. For the four lanes scenario the density of the two fast lanes was set to 5 vehicles and the speed limit to 120 km/h, The density of the two slower lanes was again varied over the range {5,10,15,20,25} veh./km/lane, with the maximum speed limit set to 80 km/h.

The RSU transmits a MM by including it once in its beacon. A new MM is transmitted every I_R beacons – I_R was varied between 1 and 2, while at the same time the lifetime of a MM was varied between 2 and 4 seconds. The beacon frequency was set to 1 Hz for all scenarios.

D. Performance metrics

This section lists all metrics used for measuring performance.

Medium busy time fraction. Measures the fraction of time that an arbitrary node experienced the medium as busy during the entire simulation, either because it was itself transmitting or because someone else was. We use this as an indication for the load on the network. Busy time was only measured inside the logging region (see Fig. 6) which is equal to the simple broadcast protocol's destination region, except that it extends another 100m downstream.
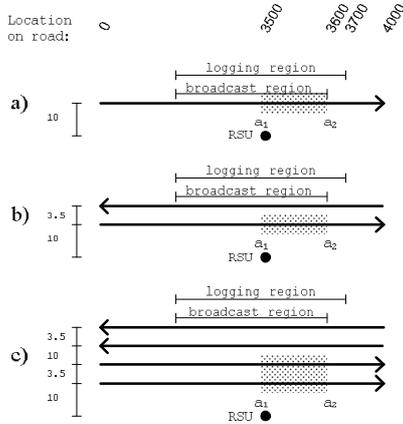

Figure 6. The traffic simulation scenarios.

MM miss rate. For every MM transmitted by the RSU the nodes were logged which were, for at least one beacon interval, part of the destination set. If a node did not receive the MM it was counted as a miss. The MM miss rate was calculated by dividing the per node average number of missed MMs by the average total number of MMs that a node *should* have received.

Notification distance. This is the average distance at which a vehicle received its first MR for a merging vehicle.

For all shown results confidence intervals have been left out but lie within a few percent of the shown means.

E. Simulation results

We start by judging whether our constrained protocol is able to meet requirements. This proved positive: in all situations except for the single lane case nodes received a warning well in advance to be able to react properly. Figure 9 shows the notification distance for one and for two lanes. Messages were generated when vehicles were about 400 meters away. For the scenario with four lanes the figures are quite similar. At 120 km/h distances were larger but trends were similar.

The results show that the performance of the constrained geocast mainly suffers from network fragmentation and hardly from transmission failures due to a high network load. Although the upstream vehicles carry any received MM with them, the lifetime of a MM is too short for this rule to bridge network gaps. As densities increase, performance also increases. Since network fragmentation implies that there is plenty of room for a merging vehicle to merge, we deem this a desirable property of the protocol. At one lane, with very low densities, the constrained geocast as well as the simple broadcast protocol operate as if the RSU was only broadcasting single hop.

Performance results of the simple broadcast protocol are somewhat different: although it also suffers from network fragmentation at very low densities, its performance improves faster because upstream nodes will immediately forward the MM. Already at 15 veh./km/lane however we can see that reliability drops again due to transmission failures, which is in contrast with the constrained geocast. This effect increases as densities go up, as can be seen in Fig 8 and 10. For Fig. 8 miss ratios are higher at 120 km/h but trends are similar.

In all cases the simple broadcast protocol has a higher notification distance. This is due to the fact that MMs are rebroadcasted immediately, in stead of having to wait for the next scheduled beacon. In this way the MMs can be disseminated much faster throughout the network.

Figure 7 shows the impact of both protocols on the network load w.r.t. beaconing. In both cases the impact is relatively low, even with a beaconing rate of only 1 Hz. It can be seen that as the road traffic density increases, the overhead of the constrained geocast becomes larger than the simple broadcast. This is due to the fact that as the road traffic density increases, the number of destination nodes that continuously include the MM to indicate its receipt goes up. Recall that we continuously have four concurrent merging vehicles, each with their own destination set, and see again Fig. 4.

VI. CONCLUSIONS AND FUTURE WORK

In this work we presented *constrained geocast* as a geocast paradigm in which the destination set is defined by the future position of the nodes involved. We have developed and tested a constrained geocast protocol in the context of a CACC merging application, and have compared its performance with a simple geo-broadcast protocol. The aim of the protocol is to route messages selectively but reliable w.r.t. transmission failures due to high network load. Unreliability due to

network fragmentation is not seen as a significant problem, since this implies that there is plenty of room to merge.

Both protocols were able to meet the merging application requirements unless densities were so low that they caused frequent network fragmentation. The simple broadcast protocol is more reliable when densities are low, while the constrained geocast is more reliable at higher densities since it is better able to cope with transmission failures.

Network overhead of the constrained geocast protocol increases as densities increase. Although it is still low compared to the beaconing load, it does surpass the overhead for the simple broadcast protocol for higher densities. Our first next step is to lessen this overhead.

Obviously it is not very efficient to include the entire MM only to indicate its receipt. In future work we will therefore only include the identifier of the MM (or MR, see next remark) to indicate this. We furthermore want to make inclusion and removal of the MRs in the MM dynamic, so that MRs need only be disseminated to where they are needed. In Fig. 4 for instance this would mean that only the MR belonging to node M would be included in the MM reaching destination region d_M. Together these alterations should significantly improve performance.

REFERENCES

[1] G. Naus, R. Vugts, J. Ploeg, M. van de Molengraft, and M. Steinbuch, *Cooperative adaptive cruise control, design and experiments*. American Control Conference, Baltimore, MD, United States, accepted, 2010.

[2] Rattaphol Pueboobpaphan, Fei Liu, Bart van Arem, *The Impacts of a Communication based Merging Assistant on Traffic Flows of Manual and Equipped Vehicles at an On-ramp Using Traffic Flow Simulation*. To appear in Proceedings of the 13th International IEEE Conference on Intelligent Transporation systems (ITSC 2010), September 2010.

[3] A. Yuen et al., *D3.3.3 Local dynamic map specifications*, SP3 SINTECH SAFESPOT, Cooperative Vehicles and Road Infrastructure for Road Safety, April 2008.

[4] C. Maihöfer, T. Leinmüller, E. Schoch, *Abiding Geocast: Time-Stable Geocast for Ad Hoc Networks*. Proceedings of VANET '05, 2005.

[5] S. Ni, Y. Tseng, Y. Chen, J. Sheu, *The broadcast storm problem in a mobile ad hoc network*. Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), 1999.

[6] C. Maihöfer, "A Survey of Geocast Routing Protocols", IEEE Communications Surveys, Vol. 6, No. 2, 2004, pp. 32- 42.

[7] B. Karp and HT Kung, "GPSR: greedy perimeter stateless routing for wireless networks. In Proceedings of the 6th annual international conference on Mobile computing and networking", 2000.

[8] Ann Hsu, Sonia Sachs, Farokh, Eskafi, Pravin Varaiya, "The Design of Platoon Maneuvers for IVHS", American Control Conference, 1991.

[9] L. Dhevi Baskar, B. De Schutter, H. Hellendoorn, "Hierarchical Traffic Control and management with Intelliugent Vehicles", Proc. of IEEE Intelligent Vehicles Symposium, 2007, pp. 834 -839.

[10] S. Halle, B. Chaib-draa, J. Laumonier, "Car platoons simulated as a multiagent system", Proc. 4th Worksshop on Agent-Based Simulation, 2003, pp. 57-63.

[11] Information Society Technologies, D31 European ITS Communication Architecture Overall Framework Proof of Concept Implementation. Version 2.0, March 2009.

[12] IEEE Computer Society, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. July, 2010.

[13] M. Treiber, A. Hennecke, D. Helbing. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E, 62(2), Aug 2000.
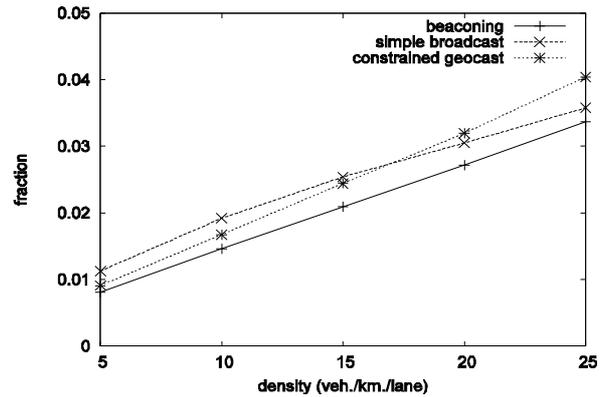
Figure 7.Avg. medium busy time fraction experienced by a vehicle for two lanes.
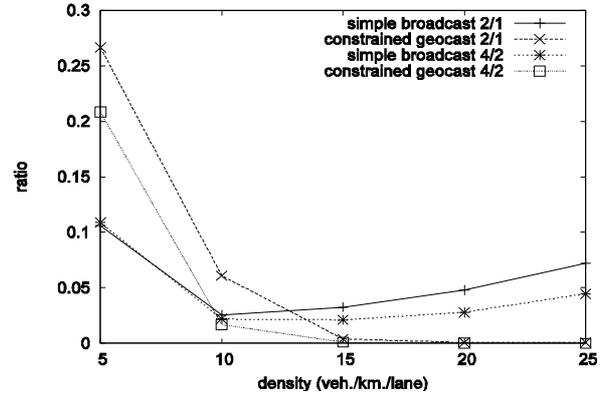

Figure 8. Merge Message (MM) miss rates, for two lanes and for different lifetime/include values.
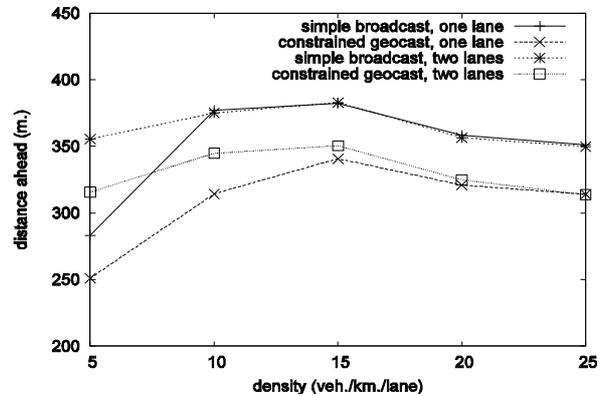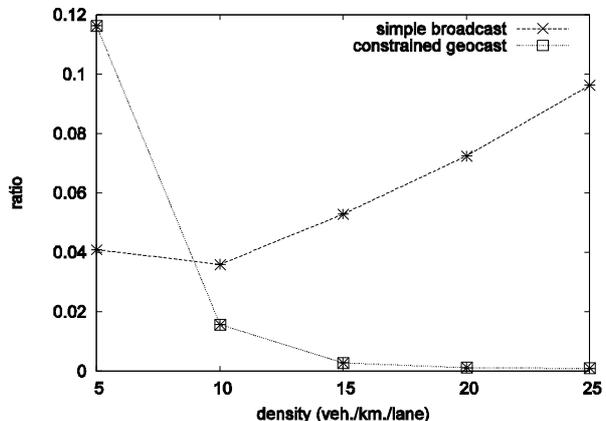

Figure 9. Advance notification distance for one and two lanes at 80 km/h.

Figure 10. The MM miss rate for four lanes.