Fast Distributed Agreements and Safety-Critical Scenarios in VANETs

Gérard Le Lann

Fast Distributed Agreements and Safety-Critical Scenarios in VANETs

Gérard Le Lann
INRIA, BP 105, 78153 Le Chesnay Cedex, France

Abstract—**Longitudinal and lateral conflicting safety-critical scenarios as they arise in VANETs are investigated, as well as solutions that rest on time-bounded distributed agreements, in the presence of concurrency and unreliable inter-vehicular communications. The Fast Distributed Agreement (DA) problem and an algorithm which solves Fast DA are presented and informally specified. Analytical expressions of worst-case time bounds for reaching agreement are provided. We verify that stringent safety requirements are met through realistic examples drawn from two safety-critical scenarios.**

Keywords-Vehicular Ad Hoc Networks; Automated Vehicles; Time-Bounded and Reliable Inter-Vehicular Communications; Distributed Agreement; Safety.

I. INTRODUCTION

We consider ad hoc networks of fully automated vehicles circulating on bi-directional main roads and highways, a.k.a. VANETs (Vehicular Ad hoc Networks) [1]. Our focus is on safety-critical (SC) scenarios, where collisions are inevitable if not handled correctly. We examine longitudinal SC scenarios occurring within a (single lane) vehicular string and lateral SC maneuvers that span adjacent lanes/strings. Steep braking, changes of velocity, of inter-vehicular gaps, are examples of the former. Lane changes, on-ramp merging, overtaking, lane merging, are examples of the latter.

So far, with few exceptions, published work does not consider concurrency. In real settings, longitudinal or/and lateral SC scenarios may be undertaken simultaneously by some unknown number of vehicles unaware of impending hazardous conflicts due to concurrency. For example, a vehicle *V* brakes abruptly, forcing its followers to decelerate steeply, thus reducing their inter-vehicular gaps, while another vehicle *W* attempts a lane change for being inserted between two *V*'s followers, *V* not within sight of *W*. Being physically unfeasible, especially at medium/high velocities, that insertion must be prohibited. Another frequent scenario occurs when two vehicles, one circulating in the lane left of some lane *j*, another circulating in the lane right of *j*, undertake concurrent conflicting lane changes for being inserted in the same "slot" within a string in lane *j*.

We take the view that VANETs shall be analyzed as a collection of ad hoc strings—an isolated vehicle is a string of size 1—subject to conflicting behavioral "interferences", where safety may be jeopardized. Consequently, for addressing safety issues correctly, such "interferences" must be analyzed. Members of a string must agree explicitly on "what to do" prior to undertaking or granting SC maneuvers. This differs from current approaches with autonomous vehicles, based on guessing possible moves of other vehicles, via robotics capabilities. Given that such guesses are not risk-free, large "safety margins" are mandatory, which is antagonistic with the various goals of "efficiency" targeted by autonomous driving. Also, this differs from classical dissemination schemes, whereby a single member (typically, a platoon leader) decides unilaterally to change some global parameter, e.g., velocity, other members having to instantiate this decision. Consequences of SC events received by other members while dissemination is underway are ignored.

Issues that arise with SC scenarios shall be addressed as problems in cyber-physics. The handling of a SC scenario rests on a pair $\{A, \Phi\}$, where A stands for a solution based on inter-vehicular communications, and Φ stands for control laws that govern vehicle behaviors in the physical space, according to decisions made via A. Execution of A shall prefix activations of Φ. (Processes Φ are not examined in this paper.) SC scenarios may develop far away from a road-side unit. It follows that only vehicle-to-vehicle (V2V) communications can be considered in A.

We focus on SC scenarios where the Fast Distributed Agreement (DA) problem arises. Fast DA states that vehicles participating in a SC scenario must reach agreement, quickly. More precisely, the Time Bounded Termination (TBT) requirement states that agreement shall be reached in less than α, under worst-case conditions regarding concurrency, message losses, number of vehicles involved, and MAC-level contention. Analytical expressions of α must be given, enabling a quantification of *Fast* DA. As a rule of thumb regarding safety, vehicles shall move by less than the average vehicle size (6 m approximately) while reaching agreement, assuming no message losses. At 108 km/h, that entails α = 200 ms. Meeting TBT is far from trivial. On crowded highways (e.g., 1 vehicle every 12 m), 3 or 4 lanes each direction, radio interference range in the order of 400m, the number of transmissions that may interfere with any given vehicle may be as high as 440 approximately. Assuming a V2V activity ratio in the order of 25%, up to 110 vehicles may attempt accessing a radio channel at the same time. It is well known that bounds in the order of 200 ms cannot be achieved with current IEEE or ETSI standards under such conditions.

In Section II, a system model is given and cohorts, a formalization of strings, are presented, along with the merits of neighbor-to-neighbor (N2N) directional communications. Section III begins with a brief introduction to string-wide message dissemination, followed by an informal specification of Fast DA. In Section IV, we give an informal

specification of VAgree, an algorithm which solves fast DA, and we show how VAgree is used in two SC scenarios. Analytical expressions of worst-case time bounds are given, followed by a short discussion on naming.

II. System Model

A. Introduction

Behaviors of vehicles are under the control of on-board (OB) systems, which include robotics devices (e.g., sensors, cameras, front-looking, rear-looking, side-looking radars, lidars, actuators), radio communication devices, associated software, e-maps, and GNSS receivers (e.g., GPS, Galileo). Examples of functions available via robotics are lane-level positioning, safe longitudinal and lateral spacing. We assume lane numbering (consecutive integers), known via e-maps. Robotics technology is limited to line-of-sight, and devices may fail, temporarily or permanently. Thus the need for V2V communications, which serve (1) to extend or/and to back-up robotics-centric functions, (2) to enable non-line-of-sight inter-vehicular coordination. V2V communications are defined by standards such as IEEE 802.11p or ETSI ITS-G5. Here, we only need to know that V2V communications rest on omnidirectional transmissions, radio range in the order of 250 m, interference range in the order of 400 m, and 10 MHz wireless channels [2]. CSMA/CA is the standardized MAC protocol. V2V communications are known to be unreliable [3], [4]. Since successful deliveries of V2V messages are not guaranteed with such standards, they are inappropriate for SC communications. Likewise, the problem of how to guarantee small bounded channel access delays (BCAD) under realistic assumptions (e.g., numerous contenders, variable numbers of lanes, inaccurate GPS coordinates, message/reservation losses) remains unsolved. For example, see [5] where MAC delays achieved with the IEEE 802.11p protocol are given, resorting to analytical modeling and NS-2 simulations. For various channel loads, assuming 1 vehicle every 12 m, highest stochastic delays range between 75.3 ms and 211.8 ms. Exact worst-case delays are theoretically unbounded.

Clearly, other communication protocols are needed for meeting the TBT requirement. Given that accidents involve vehicles necessarily sufficiently close to each other, short-range communications should suffice for the handling of SC scenarios. An interesting approach consists in addressing MAC problems arising with lateral communications separately from MAC problems as they arise with longitudinal inter-neighbor communications in strings.

B. Cohorts and N2N Directional Communications

A cohort is an ad hoc string of vehicles, referred to as members, circulating in the same lane, bound to meet unambiguous specifications [6], [7]. To the exception of head *CH* and tail *CT*, every cohort member has two neighbors (see Fig. 1). A cohort specification states the highest possible number of members (n^*) and velocity dependent safe spacing intervals, denoted s_{xy} in Fig. 1. With $n^* \approx 30$, a cohort would span in the order of 250 m (resp., 450 m) at very low (resp., high) velocities. Safe inter-cohort gaps (which depend on velocities), denoted $S_{ct/ch}$ in Fig. 1, are also specified. Due to these gaps, cohorts do not "interfere" with each other. In particular, occurrence of a "brick wall" phenomenon within some cohort *C* does not result in rear-end collisions with and within cohorts following *C*. Cohort-wide coordination can be accomplished via algorithms resting on N2N directional communications. There is no need to "pollute" the ether with 360° antennas over, e.g., 250 m, for coordinating consecutive members of a given cohort. Very short-range (e.g., 30 m), small beamwidth (e.g., 30°) front-looking and rear-looking unidirectional radio antennas suffice [8]. Observe that they would complement radars and lidars regarding spacing, as well as omnidirectional V2V communications, thereby instantiating diversified redundancy, which is essential regarding safety.

Neighbors exchange N2N messages and N2N beacons, upstream and downstream. Beacons carry ranks. This is how members assign themselves consecutive ranks, 1 for *CH*, and $n \leq n^*$ for *CT*. An isolated vehicle assigns itself rank 1. A vehicle coming from behind would assign itself rank 2 upon receiving a beacon from its predecessor, and so on.

Cohort topology knowledge (CTK) is an important feature enabled by N2N communications. Periodically, string members generate N2N beacons carrying their respective intrinsic attributes (e.g., size and type), their ranks, and their current physical parameters (e.g., lane number, geolocation in lane, velocity, spacing with predecessor and successor). These messages are propagated throughout a cohort via a dissemination algorithm (see Section III). Therefore, besides current *n*, every member of a cohort is knowledgeable of those intrinsic attributes and current parameters specific to other members forming a group Γ of *g* contiguous neighbors, $g \leq n$, as well as whether Γ is located upstream or downstream relative to a member's rank—see further for how CTK can be used.

With short-range directional antennas, radio interferences may occur only among a limited set of nearby vehicles, which permits to solve BCAD—see [9] for a detailed discussion of existing solutions. It follows that λ, the highest 1-hop N2N link delay incurred with transmitting a longest N2N message, in the absence of failures, is known. Delay λ includes (1) the delay incurred when transmitting a N2N message, worst-case MAC access delay included, (2) a receiver's OB processing time. Beacons and acknowledgements being shorter than N2N messages, their N2N link delays are smaller than λ.

C. Dependability Issues and Cohort Split

Fading, interferences and channel collisions which contribute to garbling medium-range omnidirectional V2V communications are less of a problem with short-range directional N2N communications. Nevertheless, safety requires acknowledged communications. Returning an acknowledgement (ack) for every correct delivery of a N2N message is trivially feasible in cohorts, since neighbors know each other. With V2V omnidirectional communications, it is notoriously difficult to specify worst-case conditions, such as highest number of repetitions needed to cope with a loss. Conversely, with N2N communications, worst-cases can be

easily specified. Neighbors exchange beacons periodically, every π, whenever there is no messaging activity. Let p stand for a small integer, $p > 1$. A N2N link that has been inactive longer than $p\pi$ time units is declared failed. Failure of an OB system translates in a N2N link failure. (A vehicle with a failed OB system must stop without creating hazards, typically reaching an emergency lane as soon as possible, hazard lights activated.) With $p = 2$ and $\pi = 250$ ms, vehicles would travel less than 12.5 m at 90 km/h until a N2N link failure is detected. Safety is not necessarily sacrificed, since safe inter-neighbor gaps can be maintained via robotics. However, given our goal in this paper, we must address the network partitioning issue. Many impossibility results have been established regarding agreement in the presence of partitioning in wired networks [10]. Such results do not hold with cohorts, for the following fundamental reason: *communication network partitioning leads to physical network partitioning*.

This is called a cohort split. Consider cohort C and two neighbors X and Y, Y following X. Assume a failed X/Y link. When Y detects that failure, Y broadcasts a "cohort split" V2V message (to be received by all or some members of C and other cohorts), and Y activates a dissemination of that message as a N2N message within the new cohort which is being created. Y decelerates until safe spacing $S_{ct/ch}$ is instantiated between X and Y. X becomes tail of truncated cohort. When aware of the X/Y link failure, conditions permitting, X would accelerate so as to expedite the split maneuver.

For any fault-tolerant distributed algorithm, one must specify integer $f, f \geq 0$, the highest number of (message, ack) losses and OB system failures that may be experienced in the course of execution. We can safely write $f \leq (p\text{-}1)\ (n\text{-}1)$. Within a split-free cohort, less than p consecutive losses may impact a N2N link. This condition is necessarily violated in the presence of more than $(p\text{-}1)\ (n\text{-}1)$ failures.

In the sequel, cohort and string are used interchangeably.

III. AGREEMENT VS. DISSEMINATION

A. Message Dissemination

The merits of disseminating V2V messages as intra-string N2N messages can be illustrated with velocity changes in platoons. Triggered solely by a lead vehicle, such a change is broadcast via a V2V message, possibly forwarded by platoon members—the Cooperative Adaptive Cruise Control (CACC) paradigm. A major concern is string stability—avoidance of sequences of successive amplifying accelerations or decelerations [11]. However, in analyses of CACC, MAC access delays are sometimes ignored, and message losses are not always considered, with some notable recent exceptions [12].

N2N message dissemination is essential for coping with V2V message losses. Consider 2 consecutive cohorts C_i and C_j, C_j following C_i, and imagine that an "emergency slow down" V2V message is broadcast by some member of C_i, which message is received by some members of C_j, to the exception of C_j's CH (CH is the leader in the case of platoons). Only these members are able to launch the desired deceleration process within C_j. This is just one example of settings showing that it is necessary to address issues arising with SC scenarios assuming that any string member may be the triggering vehicle.

In [13], we have presented an algorithm (named Π) that achieves cohort-wide reliable message dissemination. Assume Π is triggered by a member Y other than CH or CT, k hops away from CH, k' hops away from CT, $k+k' = n-1$. Let $h = \max\{k, k'\}$. Π terminates within the following time bounds:

$$\Delta_h(f) < 4\lambda\,[h + 3(f+2)] \qquad (1)$$

(Formulae of smaller tightest bounds are more involved.)

Bound denoted Δ(f) holds when Π is initiated by CT or CH ($h = n\text{-}1$). Assuming a N2N channel bandwidth in the order of 15 Mbits/s, values of λ would range between 0.3 ms and 1.2 ms. Let us choose $\lambda = 1$ ms. Consider $n = 20$, $f = n/5$, and $h = 12$. Bounds (in ms) are:

$\Delta_{12}(0) < 48$ $\Delta_{12}(4) < 120$ $\Delta(0) < 100$ $\Delta(4) < 148$

Distances travelled at 108 km/h until Π terminates would be smaller than 4.44 m (Δ(4)) or 1.44 m ($\Delta_{12}(0)$). Note that bound Δ(0) achieved for a string of 20 vehicles is smaller than the 100 ms figure frequently quoted for delivering a single "critical" V2V message assuming no failures—a wish, not a proven bound. As for velocity changes, thanks to Π, all string members know the new targeted velocity faster than via V2V messaging in most realistic conditions (MAC contention, message losses). This results in string stability higher than achieved with CACC solutions.

B. The Fast DA Problem

Most results established for CACC approaches hold in the absence of concurrent lateral maneuvers (e.g., string insertions) triggered by adjacent vehicles. Consider some string or some string subset of g contiguous members, referred to as a group, denoted Γ. Γ's member of smallest (resp., highest) rank is denoted F (resp., L), for first and last, respectively. Agreement is needed whenever Γ is posted concurrent and conflicting events. SC events carried in V2V messages may originate from road-side clouds, from vehicles external to Γ, or from within Γ.

Longitudinal agreements are needed for numerous reasons. Here, we have chosen Velocity Agreement, which can be illustrated with the following scenario: at about the same time, a V2V message carrying "lane ahead is congested, velocity smaller than 55 km/h" is received by a member of Γ, and a V2V message carrying "icy conditions ahead, velocity smaller than 30 km/h" is received by another member. Lateral agreements are needed with SC scenarios involving lane changes. We have chosen Lane Change Agreement, which arises whenever only 1 risk-free change can be granted, out of multiple conflicting requests. The notion of conflict is defined and illustrated.

Approximate agreement and exact agreement, a.k.a. consensus, are among the most studied algorithmic problems in distributed computing [10], [14]. However, neither the problem specifications nor the solutions devised for models of wired systems are applicable to cyber-physical systems such as VANETs. Consensus (see below) is a good example.

Assumptions
- Asynchronous or synchronous system model.
- Set of g processes, $g > 1$.
- Less than g incorrect (failed) processes.
- Reliable inter-process communications.
- Every process proposes a value.

Properties
- *Validity*: Decision value is some value proposed.
- *Agreement*: No two correct processes decide differently.
- *Eventual Termination*: Every correct process eventually decides.

There are five major differences with our Fast DA problem (see Table I). First, communications are unreliable. Second, regarding Validity, a very specific decision value ought to be chosen among proposed values, since a decision value must match physical constraints (e.g., positioning, velocities). For example, in the case of on-ramp merging, which pair of vehicles is to be chosen for insertion cannot be "some" pair on a highway. Third, Eventual Termination is replaced by Time-Bounded Termination (TBT), a stronger requirement. Fourth, Acceptability serves to strengthen TBT: realistic values of bound α shall be small enough. Finally, Synchronicity is mandatory: UTC times at which members activate processes Φ must not differ by more than ε.

Process failures (our OB system failures) and failures of N2N links lead to cohort splits. If algorithm VAgree is running when a cohort split occurs, VAgree is aborted, and restarted within newly formed cohorts.

IV. Solving Fast DA

We begin with presenting the VAgree algorithm. Then, we show how VAgree is used in Velocity Agreement and Lane Change Agreement. Every scenario and triggering events are assigned a type, e.g., *va* for Velocity Agreement, *lca* for Lane Change Agreement. Also, since VAgree may be run in the course of concurrent and conflicting SC scenarios, ties must be broken *uniformly* (i.e. identically by all vehicles involved). Tie-breaking between concurrent and conflicting SC scenarios of different types rests on priorities which depend on time or on road/traffic conditions. For example, a lane change of low priority may be prohibited or delayed whenever appropriate, whereas whenever mandatory (e.g., lane merging due to closed lane ahead), a lane change request is tagged with a high priority. Due to space restrictions, use of priorities cannot be detailed in this paper.

A. VAgree

Recall that VAgree is an element of A, the cyber part of a solution for a given SC scenario. The informal specification of VAgree given in Table II holds for a given type. Values are drawn from a set associated with that type. Solving Fast DA is relatively easy from an algorithmic standpoint. The challenge of interest is to find an efficient solution, leading to tightest worst-case termination times. VAgree is a 2-phase algorithm which rests on Π for downstream and upstream dissemination of N2N messages throughout Γ (see Fig. 1). Names of N2N messages are init, collect, and decisive, underlined in the presentation below.

TABLE I. Informal specification of Fast DA

■ Assumptions
- Synchronous system model.
- Group Γ of g contiguous members, $g > 1$.
- OB system failures and N2N link failures either are tolerated or result in a group split.
- Every member may propose a value.

■ Properties
- *Validity*: Decision value $D = \Psi\{\text{proposed values}\}$.
- *Agreement:* No two members decide differently.
- *Time-Bounded Termination:* Every member decides in less than α time units.
- *Acceptability*: Distances travelled in α are smaller than average vehicle size.
- *Synchronicity*: Times at which D is posted to OB systems are comprised within a small time interval ε. Distance travelled during ε by the member earliest to post D until the latest member does so is an order of magnitude smaller than vehicle sizes.

1) Overview

Let v_k stand for a value proposed by member K, referred to as a proposer. D is any appropriate function Ψ of proposed values, e.g., smallest value, relative majority, Boolean disjunction. The Agreement requirement is met by providing every member with common knowledge (all proposals or D).

a) Notations and variables

- init: sent in phase 1
- collect and decisive: sent in phase 2
- [message]: contents of message
- T*: termination time of VAgree
- u: time needed for computing D
- Timer TT: time left until reaching UTC time T*
- Boolean P (true ≡ I have a proposal)
- Boolean FI (true ≡ I have forwarded init before)
- $D = \Psi\{[1^{st}\ \text{collect}] \cup [2^{nd}\ \text{collect}]\}$

b) States

- *listen*: state entered upon termination of VAgree; entering this state resets Booleans P and FI to false.
- *prop*: state after a collect message has been forwarded, or after creating a collect message (the latter done by F or L).
- *waitT*: OB system is inactive or runs algorithms other than VAgree, until timer TT awakes.

c) Events

e_1: issuance of proposal as requested by local OB system, in response to some external event or internal state transition
e_2: init received from a neighbor
e_3: collect received from a neighbor
e_4: decisive received from a neighbor
e_5: awakening of timer TT

Key to achieving tightest worst-case termination bounds is the choice of sending init messages in both directions during phase 1.

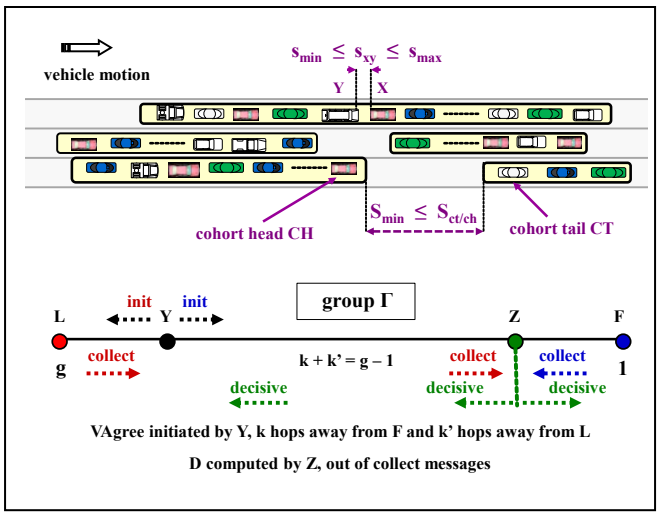

Figure 1. Cohorts and principles of VAgree illustrated ($Y \neq F$, $Y \neq L$)

2) The VAgree algorithm

Phase 1 starts when some member sends an init message to its neighbor(s), message forwarded upstream and downstream so as to "awake" F and L, in case neither F nor L "awakes" first. The specification in Table II is given for member Y other than F and L, assuming $g > 2$ (the case $g = 2$ derives trivially from Table II). Upon entering phase 1, a member is in state *listen*. To the possible exception of F or L, a member must have issued or relayed an init message prior to receiving a collect message (e_3), which triggers switching to state *prop*. A member sends or forwards an init message only once. Only 1 proposal may be issued by a member while VAgree is being run. There is no phase 1 when F or L "awakes" spontaneously.

Phase 2 is started by F (resp., L), or by both of them concurrently. Phase 2 consists in disseminating a collect message that will carry all proposals from visited members until reaching L (resp., F), or until "crossing" another collect message (see Fig. 1). In both cases, the member where this "crossing" occurs knows all proposals, thus D can be computed. In case two neighbors are provided each with both collect messages, both would compute the same D. When a collect message is created by F or L at UTC time t, termination time $T = t+\alpha+u$ is computed and recorded in the collect message. Bound α depends on n, which is known to every member through CTK (Subsection II.B). Therefore, ignoring possible minor discrepancies ε relative to UTC readings (see Synchronicity), F and L shall compute the same T, except when string membership is being updated—F and L may use different values for n. In order to avoid any ambiguity, the member which computes D also computes time $T^* = \max\{T, T'\}$. D and T^* are recorded in the decisive message. Upon receiving a decisive message at current UTC time t_d, a member computes $\delta T = T^* - t_d$, and sets its timer TT to δT. Thus, all members instantiate D at the same UTC time, when state transition $waitT \otimes e_5$ fires.

Recall that integer f which appears in bounds α cannot be greater than $(p\text{-}1)$ $(n\text{-}1)$. If not spread uniformly across a string, losses may lead to a N2N link failure whenever $f > p$.

TABLE II. INFORMAL SPECIFICATION OF VAGREE

State/Event Transitions for $Y \neq F$, $Y \neq L$

listen $\otimes$ e_1: **if** P false **then** {v_y:= proposal; P:= true; send init to both neighbors} **else** {**if** new proposal of priority higher than proposal recorded in v_y **then** v_y:= new proposal}
listen $\otimes$ e_2: **if** FI false **then** {forward init to opposite neighbor; FI:= true} **else** discard init
listen $\otimes$ e_3: *%1st collect received%*
if P true **then** {add v_y to [collect]}; store [collect]; forward collect to opposite neighbor; switch to *prop*
prop $\otimes$ e_3: *%2nd collect received%*
compute $D = \Psi${values in both collect}; compute T^* and δT; store D and T^* in decisive; forward decisive to both neighbors; set timer TT to δT; switch to *waitT*
prop $\otimes$ e_4: forward decisive to opposite neighbor; compute δT; set timer TT to δT; switch to *waitT*
waitT $\otimes$ e_5: post D to OB system; switch to *listen*
prop $\otimes$ e_1: % VAgree is in progress%
new proposal put on hold
waitT $\otimes$ e_1: % VAgree is in progress%
new proposal put on hold

Integer p is assigned a value known to all vehicles (likely, a standardized value). Since n also is common knowledge, except in transitory conditions (see above), the same f, and hence the same α, can be computed by all members if some publicly known rule links f and n. Note that such a rule may be modified at will, members using VAgree for deciding on the same new rule.

As long as state *prop* is not entered, new local proposals are accepted. In case a waiting queue would build up, the proposal of highest priority is stored in the first collect message seen. Proposals put on hold are serviced by reactivating VAgree once the current instantiation is over. Members do not defer the relaying or the processing of N2N messages unrelated to VAgree while VAgree is running.

3) Time-bounded termination and acceptability

Bounds α are easily derived from bounds Δ given in (1), where $k+k' = g\text{-}1$ and $h = \max\{k, k'\}$. Here, $f = f_1 + f_2$, f_1 (resp., f_2) standing for the number of failures occurring during phase 1 (resp., phase 2). Since $\alpha_h(f) = \Delta_h(f_1) + \Delta(f_2)$, one finds:

$$\alpha_h(f) < 4\lambda\,[h+g\text{-}1+3(f+4)] \qquad (2)$$

Highest bounds, denoted $\alpha(f)$, are reached when VAgree is initiated by F or L. Trivially:

$$\alpha(f) < 4\lambda\,[2(g+5)+3f] \qquad (3)$$

Acceptability is checked below for VAgree executions in Velocity Agreement and Lane Change Agreement scenarios.

4) Synchronicity

OB systems activate processes Φ when local timers TT awake, at the same UTC time T^*, inaccuracy ε. Thanks to OB GNSS receivers, possibly backed up by OB clocks, discrepancies in the reading of UTC time by any 2 vehicles

are less than a few ms, which translates in a few decimeters for vehicles moving at very high velocities. The Synchronicity requirement is met.

B. Velocity Agreement and Lane Change Agreement

Here, we examine complete solutions A for selected SC scenarios. For Velocity Agreement, A is VAgree. For Lane Change Agreement, A comprises phases of V2V messaging in addition to VAgree. Highest termination bounds denoted θ_x hold for a type x SC scenario.

1) Velocity agreement

Group Γ is an entire cohort, size n. Proposed values for type *va* are new velocities. With our illustrative example (Subsection III.B), $\Psi = \min\{\text{new velocity values}\}$. VAgree is as shown in Table II, $Y \neq CH$, $Y \neq CT$. Bounds $\theta_{va,h}(f)$ and $\theta_{va}(f)$ are bounds $\alpha_h(f)$ and $\alpha(f)$ in (2) and (3) respectively, with g replaced by n. Using the numerical figures introduced in Subsection III.A, we find:

$\theta_{va,12}(0) < 172 \quad \theta_{va,12}(4) < 220 \quad \theta_{va}(0) < 200 \quad \theta_{va}(4) < 248$

In the absence of failures (OB systems and message losses), agreement is reached in less than 200 ms. Distances travelled at 108 km/h until agreement is reached would be smaller than 7.44 m ($\theta_{va}(4)$) or 5.16 m ($\theta_{va,12}(0)$). The TBT and the Acceptability requirements are met.

2) Lane change agreement

A lane change scenario is triggered by some vehicle referred to as a Requestor, denoted R. Let Ξ stand for a string in lane j, and assume that R in adjacent lane j' intends to perform an insertion within Ξ. A lane change scenario is composed of 3 phases, shown in Table III. In phases 1 and 3, vehicles exchange V2V messages via Lateral Geocast (LGcast) and Unicast (Ucast) primitives. In phase 1, R only needs to "talk to" nearby vehicles in lane j. Therefore, LGcast—Geocast [15] aimed at laterally positioned vehicles, suffices. To the best of our knowledge, the problem of achieving reliable V2V omnidirectional communications is still open. Since solving this problem is out of the scope of this paper, we assume the following (hypothesis H):

- When LGcast$\{M,V\}$ is performed by vehicle V, at least 1 vehicle among those targeted by V is delivered message M correctly and in time,
- When several Ucast$\{M,V\}$ are performed directed at the same vehicle V, V is delivered at least 1 message M correctly and in time.

In our system model, a vehicle only needs to know its geolocation *in its lane*, with a precision better than average vehicle size. In addition to data provided by GNSS receivers, which can be inaccurate at times (loss of satellite signals), that can be obtained via dead reckoning, i.e. distances measured to or from a landmark (found in e-maps). Nearby vehicles in adjacent lanes have comparable geolocations in their respective lanes.

a) Phases in lane change agreement

In phase 1, R does LGcast$\{Q,R\}$, Q standing for a V2V message typed *lca*, unique identity $id(Q)$, which carries parameters $\omega(R)$ such as velocity, size, and geolocation in j'. Let $\Gamma(Q)$ stand for a group of g contiguous members in Ξ which match $\omega(R)$, referred to as Participants. For the sake of generality, assume $g > 1$. Most often, g is not higher than 5 (4 possible insertion slots). LC*test* stands for a procedure which identifies $\Gamma(Q)$ from CTK and $\omega(R)$, or which returns "nil" (not a Participant). Very briefly, in LCtest run by Y member of Ξ, Y's and R's geolocations are compared as well as whether spacing with Y's predecessor and/or successor need be increased, for accommodating R's insertion.

In phase 2, Participants run VAgree. Proposed values are typed *lca*, and they carry the identity $id(.)$ of the request being answered. More precisely, Y proposes "OK for lane change $id(Q)$, my value is v_y = [Y's size, geolocation in j, spacing $s_{y,x}$ (resp., $s_{y,z}$) with predecessor (resp., successor)]". There are various ways of computing D, ranging from elementary and sub-optimal to elaborate. Here is an example of the latter with triple (X,Y,Z) where Y precedes Z and follows X. Let $d_R(V,W)$ stand for the distance to be covered by R for an insertion between V and W, and $\delta_R(V,W)$ stand for the additional spacing to be created between V and W for R's insertion. The objective is to minimize overall energy consumption due to accelerations and decelerations. Then, function $\Psi = \min_{\Gamma(Q)}\{d_R(V,W)+\delta_R(V,W)\}$ makes sense. Let *D(R)* stand for the agreed decision. *D(R)* consists of a pair of "elected" neighbors (names and related data, such as e.g. geolocations in j). All Ucast messages carry the same *D(R)*.

In phase 3, members of $\Gamma(Q)$ do Ucast$\{id(Q),D(R),R\}$. Elected vehicles create an insertion slot for R and R starts moving toward that slot upon receiving a Ucast message. Under H, since R and members of $\Gamma(Q)$ have seen every V2V message exchanged in phases 1 and 3, R and members of $\Gamma(Q)$ are in agreement. Note that *D(R)* could be computed by R, relieving Participants from the need to run VAgree. However, since Ucast messages would not have identical contents (individual proposals differ), the loss of a Ucast message in phase 2 could lead to an incorrect decision, even under hypothesis H. Moreover, that would entail a 3rd round of V2V messaging, since R must "tell" Participants which of them are elected, correctly or incorrectly.

b) Termination times

Agreement over $\Gamma(Q)$ is reached within bounds $\theta_{lca,h}(f)$ and $\theta_{lca}(f)$ equal to bounds $\alpha_h(f)$ and $\alpha(f)$ in (2) and (3), respectively, necessarily smaller than the θ_{va}'s when $g < n$. Using the same numerical figures as above, $g = 4$ (thus $f = 0$ or 1), one finds the following highest bounds (F or L is the initiator):

$\theta_{lca}(0) < 72 \quad \text{and} \quad \theta_{lca}(1) < 84$

Let σ stand for a successful transmission and delivery latency of a V2V message, MAC access delay included.

TABLE III. PHASES IN LANE CHANGE AGREEMENT

- Phase 1: R does LGcast$\{Q,R\}$.
- Phase 2: Upon receiving Q from R, Y runs LC*test*. If "nil" is returned, no further phases. If a Participant, Y runs VAgree, proposal v_y.
- Phase 3: When VAgree delivers decision *D(R)*, Y does Ucast$\{id(Q),D(R),R\}$.

Compared to σ, OB system latencies due to running the code of VAgree are negligible, thus ignored here. Let Λ (in ms) stand for the time needed for running the 3 phases of *A*. Under *H*, losses of V2V messages can be ignored. Thus:

$$\Lambda(f) < 2\ \sigma_{max} + \theta_{lca}(f)$$

The key parameter is σ_{max} which, strictly speaking, may take unbounded values due to contention and message losses. Regarding contention, deterministic MAC protocols based on radio or optical communications are being researched for solving the BCAD problem as it arises with V2V messaging.

Let us now revisit hypothesis *H*. To the best of our knowledge, there are no solutions to the lane change problem in case *H* would be repeatedly violated when LGcast is performed. This can be tolerated by resorting to algorithm Π, under a more permissive hypothesis: at least 1 member of Ξ (say *Z*) not a member of Γ(*Q*) receives a V2V message issued via LGcast. Let $z \neq 0$ stand for the hop distance between *Z* and *F* or *L*. Dissemination of such a message from *Z* to *F* or *L* is then needed, which entails the following additional latency:

$$\Delta(f_d) < 4\lambda\ [z+3(f_d+2)] \qquad \text{(see equation (1)).}$$

Thus, $\Lambda(f) < 2\ \sigma_{max} + \Delta(f_d) + \theta_{lca}(f_a)$, where f_d (resp., f_a) stands for the number of losses experienced during dissemination (resp., agreement). Finally:

$$\Lambda(f) < 2\ \sigma_{max} + 4\lambda\ [3(z+f)+16], \quad f = f_a + f_d.$$

Z cannot be too far away from *R*, thus from *F* or *L*. Besides other numerical figures, let us have $z = 2$. We find:

$$\Lambda(0) < 2\ \sigma_{max} + 88 \quad \text{and} \quad \Lambda(1) < 2\ \sigma_{max} + 100$$

Additional latencies due to relaxing hypothesis *H* are quite small. Dissemination as per Π improves dependability in SC scenarios that involve V2V messaging, without sacrificing timeliness.

c) Lane change agreement and concurrency

Within a given string, multiple insertions can be performed quasi simultaneously provided that they do not conflict, i.e. they are physically feasible and risk-free. With Requestors sufficiently apart from each other, concurrency is conflict-free. Conversely, consider 2 lane changes requested at about the same time by requestors *V* and *W* via LGcast{*P*,*V*} and LGcast{*Q*,*W*}. *V* and *W* want to move to lane *j*. In one scenario, they are in lane $j' = j-1$. In another scenario, *V* is in lane *j'* and *W* in lane $j'' = j+1$, particularly interesting when *V* and *W* are not in mutual line-of-sight. Let us have $\Gamma^* = \Gamma(P) \cup \Gamma(Q)$, $\Gamma^* \neq \varnothing$.

A conflict is defined as follows: at least 1 member of Γ* has issued a proposal "OK for lane change *id*(*P*)" and at least 1 member of Γ* has issued a proposal "OK for lane change *id*(*Q*)". Safety requires 1 acceptance at most, which can be enforced very easily: no lane changes, both requests rejected. (This is frequently experienced with existing autonomous vehicles, since robotics capabilities achieve reactive safety only.) With VAgree, arriving at such a poor outcome can be avoided. It suffices to break ties deterministically, for example by defining some total order (ω) on the set of identities *id*(.), as well as some choice function relative to ω (e.g., $\min_\omega$ or $\max_\omega$), leading to a unique *id** over Γ*. Values retained as inputs for Ψ are those tagged with *id**. It follows that calculations of *D*(.) are relative to a unique request. For example, with $\max_\omega$ as a choice function, *W* would change lane first if $id(Q) >_\omega id(P)$, followed by *V* afterwards.

d) Naming

Names such as *X*, *Y*, or *R* used to designate vehicles must be unique. Most often, naming rests on IP addresses (OB systems), which is problematic regarding privacy. Ideally, we need a scheme whereby vehicles can assign themselves names that are unique, anonymous (no linkage with plate numbers or IP addresses), and long-lived if so desired, with no reliance on V2V communications (which would be contradictory with our objectives). Such names would be used as unambiguous sender/receiver addresses in V2V messages, as well as for creating unique message identities (our *id*(.)s). It turns out that such schemes exist, based on the cohort concept, to be presented in forthcoming publications.

REFERENCES

[1] G. Karagiannis et al., "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions", IEEE Comm. Surveys & Tutorials, vol. 13, 4, 4th quarter 2011.

[2] M.L. Sichitiu and M. Kihl, "Inter-vehicle communication systems: A survey", IEEE Comm. Surveys & Tutorials, vol. 10, 2, 2008, pp. 88-105.

[3] C. Bergenhem et al., "V2V communication quality: Measurements in a cooperative automotive platooning application", SAE Intl. J. Passeng. Cars – Electron. Elec. Syst., vol. 7, 2, Aug. 2014, 9 p.

[4] K. Karlsson, C. Bergenhem, and E. Hedin, "Field measurements of IEEE 802.11p communication in NLOS environments for a platooning application", IEEE VTC Fall-2012, pp. 1-5.

[5] Y. Yao et al., "Delay analysis and study of IEEE 802.11p based DSRC safety communication in a highway environment", Proc. IEEE Infocom 2013, pp. 1591-1599.

[6] G. Le Lann, "Cohorts and groups for safe and efficient autonomous driving on highways", 3rd IEEE Vehicular Networking Conference (VNC), Amsterdam (NL), Nov. 2011, pp. 1-8.

[7] G. Le Lann, "On the power of cohorts - Multipoint protocols for fast and reliable safety-critical communications in intelligent vehicular networks", ACM/IEEE ICCVE-2012, Beijing, Dec. 2012, pp. 35-42.

[8] R. Ramanathan et al., "Ad hoc networking with directional antennas: A complete system solution", IEEE Journal Selected Areas in Communications, vol. 23, 3, March 2005, pp. 496-506.

[9] A.A. Abdullah, L. Cai, and F. Gebali, "DSDMAC: dual sensing directional MAC protocol for ad hoc networks with directional antennas", IEEE Trans. Vehicular Technology, vol. 61, 3, March 2012, pp. 1266-1275.

[10] N.A. Lynch, Distributed Algorithms. Morgan Kaufmann. ISBN 1-55860-348-4 (1996), 872 p.

[11] G.J.L. Naus et al., "String-stable CACC design and experimental validation: A frequency-domain approach", IEEE Trans. Vehicular Technology, vol. 59, 9, Nov. 2010, pp. 4268-4279.

[12] C. Lei, and al., "Impact of packet loss on CACC string stability performance", 11th Intl. Conference on ITS Telecommunications (ITST 2011), Aug. 2011, pp. 381-386.

[13] G. Le Lann, "Safety in vehicular networks—On the inevitability of short-range directional communications", Proc. 14th Intl. Conference on Ad Hoc, Mobile, and Wireless Networks (AdHoc-Now 2015), Athens, June-July 2015, Springer LNCS 9143, pp. 347-360.

[14] M. Biely, U. Schmid, and B. Weiss, "Synchronous consensus under hybrid process and link failures", Theoretical Computer Science, 412(40), 2011, Elsevier, pp. 5602–5630.

[15] R. J. Hall, "An Improved Geocast for Mobile Ad Hoc Networks", IEEE Trans. Mobile Computing, vol. 10, 2, Feb. 2011, pp. 254-266.