Poster: A Simulation-based Testbed for Vehicular Collision Detection

G. Avino, M. Malinverno, F. Malandrino, C. Casetti, C.-F. Chiasserini
Politecnico di Torino
Torino, Italy

G. Nardini
Università di Pisa
Pisa, Italy

S. Scarpina
TIM
Torino, Italy

***Abstract*—This paper presents a simulation-based testbed for the study of collision detection in vehicular networks. We combine the Omnet++ discrete-event simulator with the SUMO mobility simulator and the SimuLTE framework, in order to study an intersection scenario where the road-side unit also integrates a collision detector. Through this set of tools, we are able to simulate the exchange of cooperative awareness messages (CAMs) and collision alerts, and evaluate the timeliness with which the latter are received.**

I. Introduction

Safety is a prominent class of applications of vehicular networks; indeed, in 2015 road accidents accounted for over 35,000 deaths in the United States alone [1], and over one million worldwide [2]. Among the safety applications that have been envisioned by ETSI and SAE, *collision detection* is one of the most significant. Vehicles periodically [3] (and anonymously [4]) report their position, direction and speed to a *detector*, by sending a *cooperative awareness message* (CAM). CAM messages coming from different vehicles are then combined together by a *collision detector*, in charge of determining whether any two vehicles are set on a collision course, and, if so, it alerts their drivers.

Collision detection is especially important in conditions of low visibility, e.g., due to the presence of obstacles like buildings that prevent drivers from timely realizing the danger. The importance and relevance of collision detection has been acknowledged by transportation regulators: as recently as December 2016, the U.S. Department of Transportation (DOT) published a Notice of Proposed Rulemaking (NPRM) for vehicular communications [5]. The document proposed to establish a new Federal Motor Vehicle Safety Standard (FMVSS), No. 150, to make vehicular networking technology compulsory, with 50% of newly-made vehicles equipped with such a technology in 2018, 75% in 2019, and 100% in 2020.

CAM and collision alert messages can be transmitted through two main technologies: 802.11p/DSRC and LTE. Our high-level goal is to compare these two alternatives from the viewpoint of both latency (i.e., how long it takes to detect a collision and alert the involved drivers) and reliability (e.g., the fraction of CAMs and alerts that get lost due to collisions or interference). In this paper, we present the simulation testbed we will use to this end, along with our reference collision detection algoritm.

Fig. 1. Simulated scenario, in the version where the LTE technology is used and RSUs are eNodeBs.

Specifically, Sec. II details our reference scenario and the tools we use to simulate it, while Sec. III presents the algorithm we implement in our detector. Finally, Sec. IV outlines the next steps of our work.

II. Scenario and simulation tools

We simulate the simple scenario detailed in Fig. 1, where different types of vehicles, e.g., cars and motorbikes, have to traverse a crossing, that can be regulated by a semaphore. Depending on the technology used, the RSU is an LTE eNodeB or an 802.11p access point. The chosen technology also impacts the location of the collision detector: with 802.11p, it can be integrated with the RSU; if LTE is adopted, it must be connected to the packet gateway (PGW) of the cellular network. The latter issue can be targeted through the so-called *multi-access edge computing* (MEC) paradigm, by integrating a virtualized packet core (vEPC) within eNodeBs and allowing collision detectors to be located therein.

The simulator we selected is Omnet++ [6], a popular, open-source, discrete-event simulator. More specifically, we use the SimuLTE-Veins distribution [7], where Omnet++ is combined with:

- the SUMO mobility simulator [8];

Algorithm 1 Collision detection

Require: $\vec{x_0}, \vec{v}, \mathcal{B}$

1: $\mathcal{C} \leftarrow \emptyset$
2: $\vec{x}(t) \leftarrow \vec{x_0} + \vec{v}t$
3: **for all** $b \in \mathcal{B}$ **do**
4: $\quad \vec{x^b}(t) \leftarrow \vec{x_0^b} + \vec{v^b} \cdot t$
5: $\quad \vec{d}(t) \leftarrow \vec{x}(t) - \vec{x^b}(t)$
6: $\quad D(t) \coloneqq |\vec{d}(t)|^2 \leftarrow (\vec{v} - \vec{v^b}) \cdot (\vec{v} - \vec{v^b})t^2 + 2(\vec{x_0} - \vec{x_0^b}) \cdot (\vec{v} - \vec{v^b})t + (\vec{x_0} - \vec{x_0^b}) \cdot (\vec{x_0} - \vec{x_0^b})$
7: $\quad t^\star \coloneqq t\colon \frac{\mathrm{d}}{\mathrm{d}t} D(t) = 0 \leftarrow \frac{-(\vec{x_0} - \vec{x_0^b}) \cdot (\vec{v} - \vec{v^b})}{|\vec{v} - \vec{v^b}|^2}$
8: $\quad$ **if** $t^\star < 0$ **then**
9: $\quad\quad$ **continue**
10: $\quad d^\star \leftarrow \sqrt{D(t^\star)}$
11: $\quad$ **if** $d^\star \leq d_{\min}$ **then**
12: $\quad\quad \mathcal{C} \leftarrow \mathcal{C} \cup \{b\}$
return $\mathcal{C}$

- the Veins framework [9], allowing SUMO and Omnet++ to interact with each other;
- the SimuLTE framework [10], adding LTE support to Omnet++.

We use this machinery to simulate three cases, namely:

- the baseline case, where no communication happens;
- the case where the 802.11p technology is used to transmit CAM messages and collision alerts;
- the case when LTE is used for the same purpose.

As mentioned earlier, our main metric of interest is *delay*. Specifically, we are interested in assessing whether collision alerts are issued and received by the interested vehicles in time for their driver to act upon them. Considering typical reaction and braking times, an alert received at least one second before the collision can be considered timely.

III. The collision detection algorithm

The algorithm, which is based on [11], takes as an input the position and speed of the current vehicle (Line 0), respectively identified by vectors $\vec{x_0}$ and $\vec{v}$ [1], as well as the previous CAMs in $\mathcal{B}$. We start by initializing the set $\mathcal{C}$ of vehicles, with which the current vehicle will collide, to the empty set (Line 1), and we compute how the position of the current vehicle will change over time (Line 2). Then, for every vehicle that generated a CAM $b \in \mathcal{B}$ recently received by the detector, we compute its position over time (Line 4) and the difference $\vec{d}(t)$ between the positions of such vehicle and the current vehicle (Line 5). The scalar $D(t) \coloneqq |\vec{d}(t)|^2$, computed in Line 6, represents the square [2] of the distance over time. We are interested in the minimum value that this quantity will take over time; to this end, in Line 7 we compute the time $t^\star$ at which $D(t)$ will take its minimum value. If $t^\star < 0$, then the vehicles are actually getting farther apart and no action is required (Line 8). Otherwise, in Line 10 we compute the minimum distance $d^\star$ the two vehicles will be at; if such a value is lower than a threshold value $d_{\min}$ (Line 11), then we need to send an alert, and thus add b to $\mathcal{C}$ (note that b essentially identifies the vehicle which sent the CAM).

In summary, Alg. 1 returns the set $\mathcal{C}$ of vehicles with which the current vehicle is set to collide. This set (along with additional information such as the time of collision) is transmitted back to the vehicles that sent the CAMs included in $\mathcal{C}$. The vehicles will therefore alert their drivers or, if appropriate, directly take action, e.g., brake before the collision happens.

[1] Note that the speed vector also includes information on the direction.

[2] Using the squared distance instead of the distance itself simplifies computations.

IV. Next steps

Although running a complete vehicular simulation as described above is a nontrivial accomplishment in itself, we have a long way to go before we can provide a comprehensive performance evaluation of vehicular collision detection. A first step we plan to make is deriving network delay results for both the 802.11p and LTE cases. In the latter case, we are also interested in how much delay is incurred within the EPC, or the vEPC network if MEC is applied. These results will be present in the final version of our poster.

A second area we are planning to improve in is the collision detection algorithm itself. Indeed, the algorithm we discussed in Sec. III is very general, but also fairly simplistic. In particular, it does not account for vehicles that may (or will) change their direction in the future, e.g., have already signaled the intention of turning at a crossing. This can cause both false positives and false negatives, and can be addressed by providing more complete input data to the detection algorithm.

Acknowledgement

This work is partially supported by TIM through the research contract "Multi-access Edge Computing", and by the EC through the H2020 5G-TRANSFORMER project (Project ID 761536).

References

[1] National Highway Traffic Safety Administration. Fatality Analysis Reporting System. https://www-fars.nhtsa.dot.gov/Main/index.aspx.
[2] World Health Organization. Global status report on road safety. http://www.who.int/violence_injury_prevention/road_safety_status/2015/GSRRS2015_data/en/.
[3] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. Lo Cigno, and F. Dressler, "How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help," *IEEE Transactions on Mobile Computing*, 2015.
[4] F. Malandrino, C. Borgiattino, C. Casetti, C. F. Chiasserini, M. Fiore, and R. Sadao, "Verification and Inference of Positions in Vehicular Networks through Anonymous Beaconing," *IEEE Transactions on Mobile Computing*, 2014.
[5] NHTSA. U.S. DOT advances deployment of Connected Vehicle Technology to prevent hundreds of thousands of crashes. http://bit.ly/2hMmtSk.
[6] Omnetpp. https://omnetpp.org.
[7] SimuLTE and Veins. http://simulte.com/add_veins.html.
[8] DLR Institute. The SUMO Mobility Simulator. http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/.
[9] Veins. http://veins.car2x.org.
[10] SimuLTE. http://simulte.com.
[11] F. Gong, B. Gao, and Q. Niu, "An Algorithm for Rapidly Computing the Minimum Distance between Two Objects Collision Detection," in *2008 Congress on Image and Signal Processing*, 2008.