Fleet-Level Communication for Self-Driving Vehicles

Andrea F. Daniele
TTI-Chicago, USA
afdaniele@ttic.edu

Noah A. Hirsch
University of Chicago, USA
nashirsch@uchicago.edu

Max X. Liu
University of Chicago, USA
maxliu@uchicago.edu

***Abstract*—As autonomous vehicles become more prevalent, a large concern is ensuring safety. The most used perception system on autonomous vehicles is vision. Even though cameras are cheap and easy to integrate, they are prone to failure in low-visibility scenarios (e.g., foggy weather). An important consideration for safety is how autonomous vehicles communicate with each other in order to reduce the risk of collisions in hazardous and unforeseeable situations. In this work, we want to investigate the optimal communication strategy for self-driving vehicles in a low-visibility hazardous scenario. Obviously, communicating with as many vehicles as possible would be ideal, but in practice, the presence of both physical and technological limitations such as wireless communication range limits and low-bandwidth communication channels impose a more structured and optimized communication strategy. Our study identifies some of the critical aspects of fleet-level communication, such as message broadcasting, message propagation and reaction time aware communication strategy. Tests on a realistic model of a town show that our communication strategy can reduce the number of collisions by about 61%.**

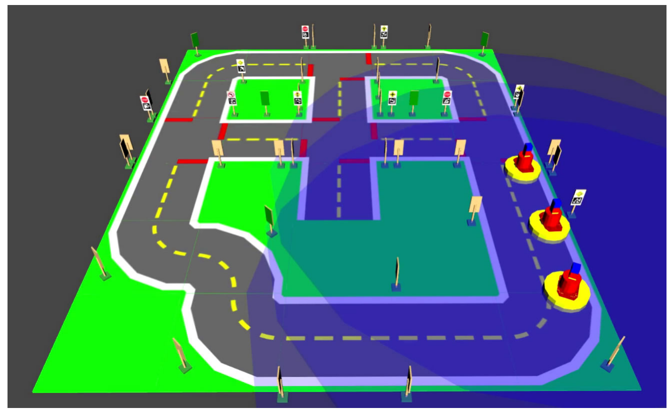

Fig. 1: 3D view of Duckietown (no simulation) showing vehicles stopping in line without colliding due to an accident located at the North-East 3-way intersection (POI). The *yellow* circles indicate the position of the vehicles after they stopped. The *blue* circles indicate their communication ranges. The *red* cylinders indicate the location advertised by the vehicles as *dangerous*. The small *blue boxes* indicate that a vehicle is broadcasting messages about known dangerous locations.

I. Introduction

In order to increase autonomous vehicle safety and optimize route planning, vehicles must be able to effectively communicate with one another. Specifically in hazardous situations, it is critical to communicate useful data to relevant vehicles. One of the main challenges in fleet-level communication is the choice of a network infrastructure suitable for the task. Wi-Fi networks are fast and cheap to deploy but have a limited range. Communication via satellites offer an unlimited range but it is expensive to deploy. Another challenge is that of deciding how much effort each car should put in trying to advertise a dangerous situation to nearby vehicles. Even though self-driving vehicles are expected to be capable of processing a lot of information directly on-board, network infrastructure technologies still require engineers to minimize the amount of data exchanged on shared communication infrastructures such as cellular or Wi-Fi network. We are interested in finding a set of functionalities that a fleet-level communication pipeline should exhibit such that we can improved the safety of self-driving vehicles while minimizing the impact on on-board computers and network infrastructures. Tests on a realistic model of a town show that our communication strategy can reduce the number of collisions by about 61%.

II. Background

For over a decade, the feasibility and applications of communication between autonomous vehicles have been well defined and studied. The biggest limitations at the time were network infrastructure and autonomous vehicle technology; but modern networks and autonomous vehicle technology have made communication applications not only feasible but also necessary. The applications of communication between autonomous vehicles span beyond road vehicles and have many practical benefits to other transportation methods such as trains and planes, but for the purposes of this work the discussion will center around road vehicles.

The applications of vehicle communication have been well thought out, one of the key challenges that remains is implementing a system infrastructure that is able to effectively deploy such features while keeping in mind the network requirements [10]. Features such as emergency warnings, collision avoidance, as well as motion planning, all require different degrees of network speed and reliability that range from extreme reliability and low latency to more relaxed requirements. With modern networks, this is becoming less of an issue but is still a key factor we keep in mind while simulating our communication network. Recent work has been done to make the implementation of communication infrastructure much easier by providing a standardized framework for coordination among a fleet of vehicles [3]. This framework would remove

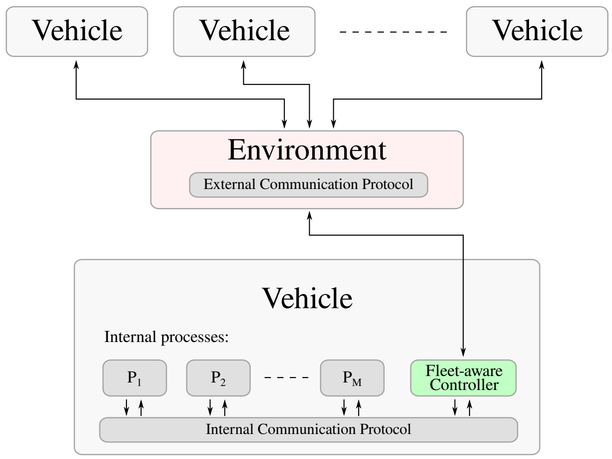

Fig. 2: Block diagram of a fleet of autonomous vehicles

the complications of needing to program individual vehicles and abstracts the global state of the system when vehicles are deployed. This will allow users to quickly build upon single vehicle tasks and deploy concurrent, sequential, or event based tasks to a fleet of vehicles. This would be extremely useful in road scenarios that require a large degree of coordination between vehicles in order to safely and efficiently navigate (ex. a large pile up of cars on a highway during rush hour).

Improving the efficiency of vehicles on roads (especially highways) has long been a very obvious application for autonomous vehicle communication [4]. There has been both recent research and previous research done in this area where the question considered was essentially how to best model vehicles following one another to best improve safety and efficiency [6, 9]. Primary variables considered are velocity and distance with findings that support flocking behavior of vehicles where velocities and distances between vehicles all converge towards a value. In this work, we focus on studying the safety of different communication models. Specifically, we look at scenarios where traditional methods may be ineffective and propose a solution that takes into account variables such as network latency and reaction speed. We present five unique communication features and detail the effectiveness of each one of them in the context of collision ratio, and safety distance.

III. Approach

Figure 2 shows a block-diagram of a system of self-driving vehicles acting on and reacting to a partially unknown environment. In this work we enable vehicles to explicitly communicate with each other. Our contribution is indicated in Figure 2 by the green block "Fleet-aware Controller". A fleet-aware controller process in a self-driving vehicle has two objectives: notifying other vehicles about hazardous situations that the vehicle is aware of, and stopping the vehicle at a safe distance from a dangerous location. A fleet-aware controller must have access both the internal and the external communication pipeline. The internal communication pipeline is usually a centralized message-based communication architecture that allows processes within the same vehicle to exchange messages. The external communication pipeline is usually a distributed message-based communication architecture that allow vehicles that are physically far from each other to exchange information. Both these modules are fundamental to our approach but neither the definition nor the implementation of any of them is within the scope of this work. We are interested in which information to bridge from the internal communication network to the external and viceversa as well as how often the same information should be broadcasted in or out of the vehicle. Our fleet-aware controller is comprised of five main features:

- Local knowledge and emergency stop
- Single message transmission
- Message propagation
- Message broadcasting
- Vehicle-chain-aware communication strategy

A. Local knowledge and emergency stop

A vehicle maintains a database of known dangerous locations. The vehicle gently stops when the distance to the closest dangerous location is approaching the safety distance. This feature does not require or subsume any communication strategy. It is a mere safety feature with no knowledge about other vehicles. The following features will be responsible for updating and sharing such a database.

B. Single message transmission

When a vehicle comes within observation distance of an obstruction on the road (e.g., an accident), it adds the location to its internal database and sends out a single danger message to every vehicle within its communication range. This message contains the location of the danger (GPS coordinates) and the time of the observation. When a vehicle receives a message from another vehicle, it adds the location of the danger to its internal database.

C. Message propagation

When a vehicle receives a message from another vehicle, it also retransmits the same message to all the vehicles within its communication range. This creates a propagation effect throughout the network of vehicles.

D. Message broadcasting

Each vehicle is responsible for maintaining an updated located database of dangerous locations. By using perception they can declare a location as safe. With message broadcasting, each vehicle will republish all the known dangerous locations in its database to the nearby vehicles with fixed frequency F. This will enable vehicles that enter the communication range of another vehicles at any given time to receive updated information about the safety of the surroundings.

(a) (b)

Fig. 3: Duckietown at the Toyota Technological Institute at Chicago (TTIC) and Duckiebot robot used in this work

E. Vehicle-chain-aware communication strategy

When a vehicle stops due to the presence of an obstruction, it suddenly becomes an obstruction itself. Vehicles stopping due to an emergency will add their location to the database of dangerous locations.

IV. Evaluation and Results

We evaluate our communication pipeline on a fleet of robots (i.e., Duckiebots) driving autonomously in a realistic model of a town called Duckietown [7]. Figures 3a and 3b show respectively our test-bed Duckietown, and one of the Duckiebots used in our experiments.

A. Implementation

Duckietown is a robotics educations and outreach effort. For our experiments we use a 3-by-4 meters Duckietown with three 3-way and one 4-way intersections. The combination of hardware and software allows for highly modular autonomous vehicles and smart-cities research. The vehicles used in Duckietown are called Duckiebots (Figure 3b) and they use a battery-powered 3-wheeled chassis and are controlled by an on-board Raspberry Pi 3 Model B. Duckiebots feature a on-board camera for lane detection and a 2.4GHz b/g/n Wi-Fi module. The Duckietown software stack uses ROS (Robot Operating System) [8] as internal communication protocol, though it does not provide any support or implementation for fleet-level communication. As for the external communication protocol, we use LCM (Lightweight Communications and Marshalling) [2]. LCM is a multi-platform library that allows simple low-latency messaging between processes and devices. All the vehicles are connected to a central Wi-Fi Access Point.

Self-driving vehicles in the real world can exchange information about locations of interests without any ambiguity since they all share the same reference frame, that is the geographic coordinate system used by GPS (Global Positioning System) satellites. Duckietown does not use the standard GPS technology for two reasons: (i) the GPS signal is not available indoor; (ii) the GPS provide a localization error (about $3-4$ meters) that is bigger than the size of the whole Duckietown. In order to provide the Duckiebots a GPS-like service, we equipped each Duckiebot with an AprilTag visual fiducial marker [5] with the normal to the tag orthogonal to the road and pointing to the ceiling of the room. A camera with wide field-of-view is attached to the ceiling of the room, at the center of the town looking down. This allows us to estimate the pose of each Duckiebot within Duckietown simply by detecting the fiducial markers and estimating their poses relative to the camera. In this setup, the marker-based localization error is about $2cm$, which corresponds to about 4 meters when scaled to the real world. The localization service is provided to the Duckiebots via LCM messages with a frequency of about $8Hz$.

B. Experiments

We test our communication pipeline on a common hazardous situations, that is a car accident at a 3-way intersection in the case of low visibility conditions. For the remainder of this document we will refer to the location of the accident (its GPS coordinates) as the *Point of Interest* (POI). In our scenario, after the accident occurred, a number of vehicles N will approach the same intersection from different directions and with different speed. We test our model on a variable number of vehicles between 3 and 6. We observe consistent results across all the values of N that we consider. One experiment starts with N vehicles start moving at the same time and from different locations and driving towards the POI. The experiment ends when all the vehicles stop (either after they crashed or stopped safely). In our experiments, due to the lack of visibility, the first vehicle approaching the intersection will inevitably crash into the damaged vehicle. Immediately after the crash, the communication pipeline of such vehicle takes over. We are interested in reducing the ratio C of cars colliding with each other, as well as increasing the average distance D_p from the POI (how far away from the POI a vehicle will stop) and the safety distance D_s, that is the average distance between two consecutive vehicles.

We evaluate the effectiveness of our pipeline by comparing the values of C, D_p, and D_s with those achieved by the baseline. In real life scenarios, when no autonomy is involved, the outcome of such an event heavily depends on the drivers' ability, experience, attention, and responsiveness [1] as well as visibility conditions (e.g., lens flare at sunset and sunrise, fog, occlusions). This makes it hard to define a baseline to compare against. Although, we can all agree on two points: (i) the worst case (i.e., all approaching cars crashing) is not unlikely to happen; (ii) the difference between worst and best case scenario is not a mere number when people's lives is involved. For these reason, we consider the worst case as a baseline.

C. Technological and Physical limitations

We are interested in the effect of our model in real life scenarios. In order to reduce the differences between our lab setup and the real world, we simulate physical effects such as wireless communication range limits, communication instability due to packets loss, and GPS-based localization latency. We artificially simulate the wireless communication limit to $100m$ (scaled to real world) by ignoring all the messages that are sent from a distance that is higher than the limit. Communication instability is simulated by exchanging information via UDP protocol, that does not attempt re-transmission in case of lost packets (unlike TCP). We also introduce a 1 second delay between the GPS location we broadcast to the robots and their actual location.

D. Ablation tests

As explained in Section III, our communication pipeline is comprised of five key features. We believe that these features are fundamental for improving self-driving vehicles' safety via explicit communication. In order to study the contribution of each feature, we run an ablation test, in which we run the same number of experiments on a version of our pipeline obtained by disabling one or multiple features. In particular, we consider four different models: **baseline**, **simple**, **propagation**, and **full-model**.

	Model			
Feature	**baseline**	**simple**	**propagation**	**full-model**
Single message transmission	NO	YES	YES	YES
Message propagation	NO	NO	YES	YES
Message broadcasting	NO	NO	NO	YES

We run 6 tests for each communication model. For a given vehicle, initial position, speed, and direction are kept unchanged throughout all the tests.

E. Results

Figure 4 shows the collision ratio observed for different communication models in the case of $N = 3$ vehicles approaching the POI. We observe that in the absence of good visual perception (**baseline**), as we consider the worst case, the collision rate is 100% as all the vehicles will end up

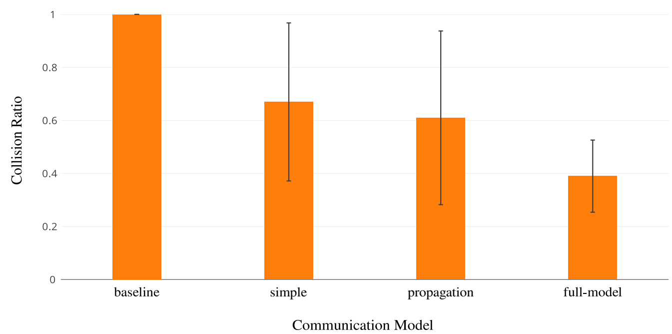

Fig. 4: Collision ratio (C) for different communication models

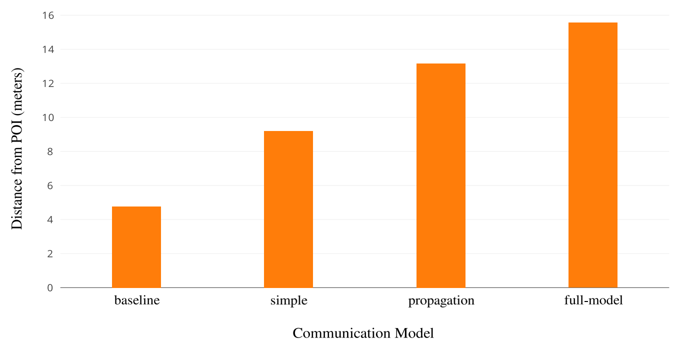

Fig. 5: Distance from the POI (D_p) for different communication models

colliding with each other at the POI. By enabling the first vehicle that reaches the POI (and collides) to warn all the vehicles within the communication range (**simple**), we found out that on average, only one vehicle out of 3 is close enough to receive the message and stop safely. It is worth noting that any other vehicle entering the communication range after the message was sent will not receive any message because the vehicle will not keep broadcasting the message. This means that whether the vehicles will collide or not, strictly depends on their position when the message was generated. A natural extension of this model would be to allow all the vehicles that received the message to propagate it to other vehicles nearby (**propagation** model). We notice that even though the average collision ratio decreases, the scenario can always degenerate to the case where all vehicles are too far from each other to exchange messages, hence they will all collide. Our complete communication pipeline (**full-model**) features all these communication strategies as well as a broadcasting mechanism that allow all the vehicles that receive a message to keep publishing it at a frequency F. We empirically found that in our model of town, a frequency of $H = 1Hz$ ensures safety while minimizing the load on the communication channel.

Figures 5 and 6 show the average distance between the POI and the place where the vehicles stopped, and the average distance between two consecutive vehicles respectively (both scaled to the real world). The distance D_p is close to 4.5 meters for the **baseline** model since all car collide with each

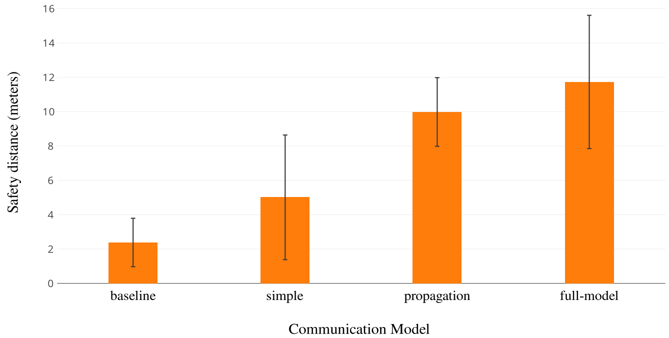

Fig. 6: Safety distance (D_s) between two consecutive vehicles

other around the POI. The models **simple** and **propagation** increase both D_p and D_s compared to the **baseline** model, but they fail when vehicles are outside the communication range(s). The complete pipeline **full-model** succeed in warning vehicles that enter the communication range(s) later. Unlike D_p, D_s does not depend on the number of vehicles involved in the experiment, hence it is an absolute indicator of safety. Figure 1 shows a 3D visualization of Duckietown in which three Duckiebots stopped in a straight line without colliding into each other after successfully exchanging messages about an accident located at the North-East 3-way intersection (POI). The *yellow* circles indicate the position of the vehicles after they stopped. The *blue* circles indicate their communication ranges. The *red* cylinders indicate the location advertised by the vehicles as *dangerous*.

The proposed communication model achieves a collision ratio reduction of about 61%, while increasing the distance from the POI and the safety distance between vehicles by a factor of 3 and 5 respectively, compared to the baseline. It is important to notice that Figure 4 present a bias of $1/N$ because we assumed that the first vehicle must collide with the damaged vehicle to detect it. This assumption is necessary to define a baseline to compare against. Giving the first vehicle to time to perceive the obstacle and stop safely would raise the question why the others cannot do the same.

V. Conclusion

In this work we presented a set of functionalities that a fleet-level communication pipeline should exhibit such that we can improved the safety of self-driving vehicles while minimizing the impact on on-board computers and network infrastructures. Tests on a realistic model of a town show that our communication strategy can reduce the number of collisions by about 61%. Our model could be improved by giving the vehicles the ability to discard messages that are not relevant to their current paths or the ability to re-plan so that the danger is avoided.

VI. Individual Contributions

//TODO

References

[1] D. W. Eby. An analysis of crash likelihood: age versus driving experience. 1995.

[2] A. S. Huang, E. Olson, and D. C. Moore. Lcm: Lightweight communications and marshalling. In *Intelligent robots and systems (IROS), 2010 IEEE/RSJ international conference on*, pages 4057–4062. IEEE, 2010.

[3] K. Lima, E. R. B. Marques, J. Pinto, and J. B. Sousa. Dolphin: a task orchestration language for autonomous vehicle networks.

[4] R. M. Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, 2007.

[5] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, May 2011.

[6] H. Ou and T.-Q. Tang. An extended two-lane car-following model accounting for inter-vehicle communication. *Physica A: Statistical Mechanics and its Applications*, 2017.

[7] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, et al. Duckietown: an open, inexpensive and flexible platform for autonomy education and research. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1497–1504. IEEE, 2017.

[8] M. Quigley, J. Faust, T. Foote, and J. Leibs. Ros: an open-source robot operating system.

[9] H. Tanner, A. Jadbabaie, and G. J. Pappas. Coordination of multiple autonomous vehicles.

[10] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk. A survey of inter-vehicle communication protocols and their applications. *IEEE Communications Surveys & Tutorials*, 11(2).