

RESEARCH

Open Access

Evaluation of CACC string stability using SUMO, Simulink, and OMNeT++

Chenxi Lei¹, Emiel Martijn van Eenennaam^{1*}, Wouter Klein Wolterink¹, Jeroen Ploeg², Georgios Karagiannis¹ and Geert Heijnen¹

Abstract

Recent development in wireless technology enables communication between vehicles. The concept of co-operative adaptive cruise control (CACC)—which uses wireless communication between vehicles—aims at string stable behavior in a platoon of vehicles. “String stability” means any non-zero position, speed, and acceleration errors of an individual vehicle in a string do not amplify when they propagate upstream. In this article, we will discuss the string stability of CACC and evaluate its performance under varying packet loss ratios, beacon sending frequencies, and time headway settings in simulation experiments. The simulation framework is built up with a controller prototype, a traffic simulator, and a network simulator.

Keywords: CACC, string stability, packet loss ratio, beacon sending frequency, time headway

1 Introduction

In vehicular ad hoc networks (VANETs), on-board units (OBUs) give vehicles the ability of communication to make them “smart objects” more than mere transportation tools. VANETs comprise (1) communication between vehicles, often referred to as vehicle-to-vehicle (V2V), and (2) communication between vehicles and road side units (RSUs), known as vehicle-to-infrastructure (V2I). Both modes of communication can be performed using the same ad hoc wireless communication technology, such as IEEE 802.11p [1]. By using VANETs, many new services for intelligent transport systems can be enabled, and numerous opportunities for safety and efficiency improvements are created.

Traffic congestion is a growing problem in industrialized nations worldwide. Using the concept of adaptive cruise control (ACC), which has a positive impact on traffic safety and efficiency [2], can be a partial solution to this problem. By extending a cruise control system with a radar sensor, ACC allows a vehicle to maintain a pre-set speed, as well as to adapt its speed to the speed of its predecessor in order to keep a minimum distance from its predecessor [3].

However, ACC does not sufficiently improve the string stability of vehicles. A string of vehicles is said to be ‘string stable’ when any non-zero position, speed, and acceleration errors of an individual vehicle in the string do not amplify when they propagate upstream [4,5]. As a result of ACC’s lack of string stability, already at moderate traffic densities small disturbances may lead to traffic jams, negatively impacting a road’s capacity.

An enhancement to the ACC concept is the co-operative ACC (CACC), where the OBU in a vehicle is using a communication medium to communicate with OBUs available in other vehicles or RSUs. The communicated information may include a vehicles’ position, speed, acceleration, etc., which can be used to enhance the performance of the current ACC systems.

It is expected that CACC will increase vehicle traffic efficiency and traffic flow stability [2,6]. CACC can be applied in traffic applications such as co-operative following [6,7], or vehicle platooning [7,8]. A design of CACC can be found in [9].

However promising, wireless communication using IEEE 802.11p is not flawless [10]. Collisions between two or more ongoing transmissions on the wireless medium can render both useless to a receiver, causing packet loss. Especially when vehicles are gathered close together—like in a traffic jam—the performance deteriorates due to packet loss and the limited capacity of the

* Correspondence: e.m.vaneennaam@utwente.nl

¹Department of Computer Science, University of Twente, Enschede, The Netherlands

Full list of author information is available at the end of the article

channel. Since there is only so much “air time”, the rate at which the beacons are sent should ideally be minimized to leave room for transmissions by other vehicles. It is evident that both factors impact the performance of a networked control system.

The main goal of this article is to evaluate the impact of packet loss rate (PLR) and beacon sending frequency (BSF) on CACC string stability performance by means of a simulation study. The simulation environments used to accomplish this are: (1) SUMO (simulation of urban mobility) [11], (2) Simulink [12], and (3) OMNeT++ [13] together with its MiXiM (a MiXed siMulator) framework extension [14].

The research questions that are answered by this article in order to satisfy this goal are:

- How is string stability evaluated?
- What is the impact of packet loss on the string stability performance of a CACC system?
- Are there significant differences in string stability performance between the ACC and CACC systems?

This article is an extension of [15] at the 11th International Conference on ITS Telecommunications (ITST 2011). The additional contribution of this article are twofold: (1) more details on the used simulation environments are provided, and (2) a comparison between the ACC and CACC systems based on their string stability performance is performed.

In Section 2, we will briefly introduce the control theory of CACC and illustrate the concept of string stability. Then in Section 3, we will describe our simulation environment. The simulations, corresponding results, and analysis will be shown and discussed in Section 4. Finally, in Section 5, we will conclude this article and give recommendations for future study.

2 Control theory and string stability

This section describes two main concepts used in this research study, which are (1) the control theory used by the applied adaptive cruise control mechanism and (2) the concept of string stability.

2.1 Control theory

Control theory [16] deals with the behavior of dynamical systems. The control objective is to realize a desired distance to the preceding vehicle. This desired distance may be an increasing function of vehicle velocity in order to take safety aspects into account. The result is commonly referred to as a “constant time-headway spacing policy”. In order to realize the control objective, the control system acts on the desired acceleration of the vehicle by means of actively influencing the drive force, based on radar measurements and (in case of

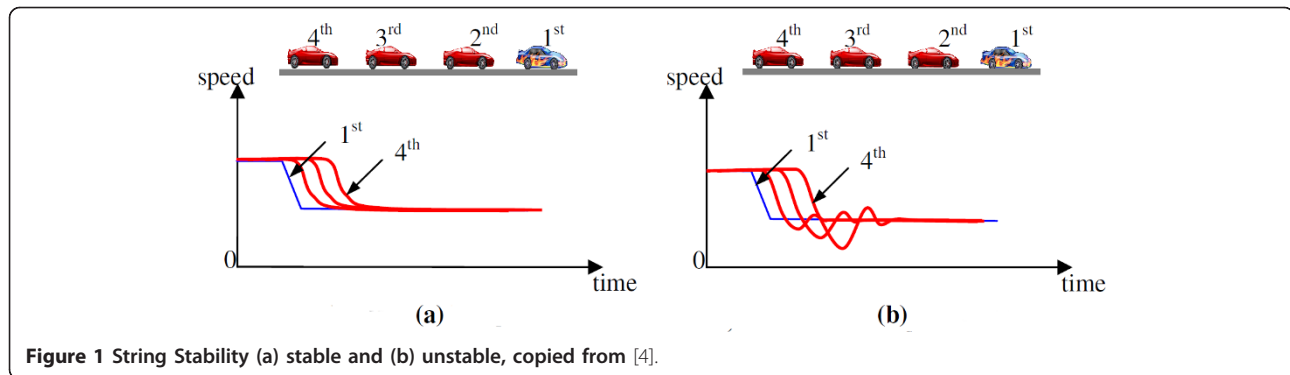
CACC) on data obtained through wireless communications. The controller’s main task is to reject disturbances caused by velocity variations of the preceding vehicles. In our study, the disturbance is caused by the behavior of other traffic, such as sudden deceleration of preceding vehicles. An ideal feedback control system should be able to cancel out all errors, effectively mitigating the effects of any forces that might or might not arise during operation and producing a response in the system that perfectly matches the designer’s wishes. In reality, this might be difficult to achieve when taking measurement errors in the sensors, delays in the controller, and imperfections in the control input into consideration. Though many solutions exist to implement a CACC controller, we will focus on a control structure that can be applied in an ad-hoc vehicle platoon scenario [9]. In this scenario, the concept of a platoon leader is not supported and all the vehicles in a platoon support the same type of one-vehicle-look-ahead CACC controller topology. The main reason of choosing this CACC controller structure is the fact that the one-vehicle-look-ahead topology is the simplest possible structure and therefore it has the highest probability of being deployed. Furthermore, a proof-of-concept implementation of this CACC controller structure has been developed within the Connect&Drive [17,18] project and shows promising results.

2.2 String stability

The term “string stability” is often used interchangeably with “platoon stability” in this field, which means that any non-zero position, speed, and acceleration errors of an individual vehicle in a string do not amplify when they propagate upstream [4,5]. According to [19], the vehicle speed should be taken as a basis for string stability, which is more relevant than distance error in view of traffic analysis. A simple scenario which can be used to explain string stability is illustrated in Figure 1a,b.

Figure 1a,b shows a string of vehicles, moving from left to right. For a clear illustration we show only four vehicles, but the concept also holds for strings of more vehicles. The leading vehicle is denoted as 1st while the last vehicle is denoted as 4th. In each of these figures, below the shown string of vehicles, a speed versus time coordinate graph for each vehicle is shown. As time goes by, the leading vehicle decelerates linearly and we can see different responses of the following vehicle in the platoon depending on whether the platoon is string stable or not.

In Figure 1a, the situation is shown where the platoon is string stable: the effect of the changing speed for the leading vehicle is not amplified through the following vehicles and the deceleration of following vehicles is smooth without any fluctuation of the speed. In Figure



1b, the platoon is considered not string stable (string unstable): the following vehicles decelerate even more than the leading vehicle. Though finally, the speeds of the following vehicles approach the leading vehicle's speed, the speed fluctuates significantly. These fluctuations are amplified from vehicle to vehicle in the upstream direction. Actually, during the period of fluctuation, the distance between neighboring vehicles also fluctuates. As a result, collisions between vehicles are more likely to happen. This example shows a decelerating lead vehicle, but the concept also holds for an accelerating lead vehicle.

String stability can be improved if the information of the preceding vehicle (e.g., acceleration and velocity) is used in the feedback loop of the Cruise Controller. This information can be collected by a low latency communication medium [9]. The most distinctive difference between ACC and CACC is that besides the preceding vehicle's speed and position used as inputs in ACC, the acceleration of the preceding vehicle transmitted through the wireless channel is also adopted as input in CACC, see Figure 2 and [9]. Therefore, CACC is treated as a solution to achieve a desired following distance with string stability. Alternatively, the preceding vehicle's acceleration can be estimated from the distance and relative velocity as measured by the radar. This leads, however, to additional phase delay in the estimated acceleration due to the estimation algorithm, which potentially causes string instability. For this reason, this method is not pursued here.

3 Simulation model

The system is evaluated in a discrete event simulation framework composed of several simulation environments and models: (1) the vehicle behavior (the controller prototype), including the ACC and CACC models, which have been implemented using Simulink [12]; (2) the mobility behavior of vehicles, which has been modeled using SUMO (Simulation of Urban Mobility) [11]; (3) the communication networking behavior, which has

been modeled using OMNeT++ [13] together with its MiXiM (a MiXed simulator) Framework extension.

3.1 Simulink model

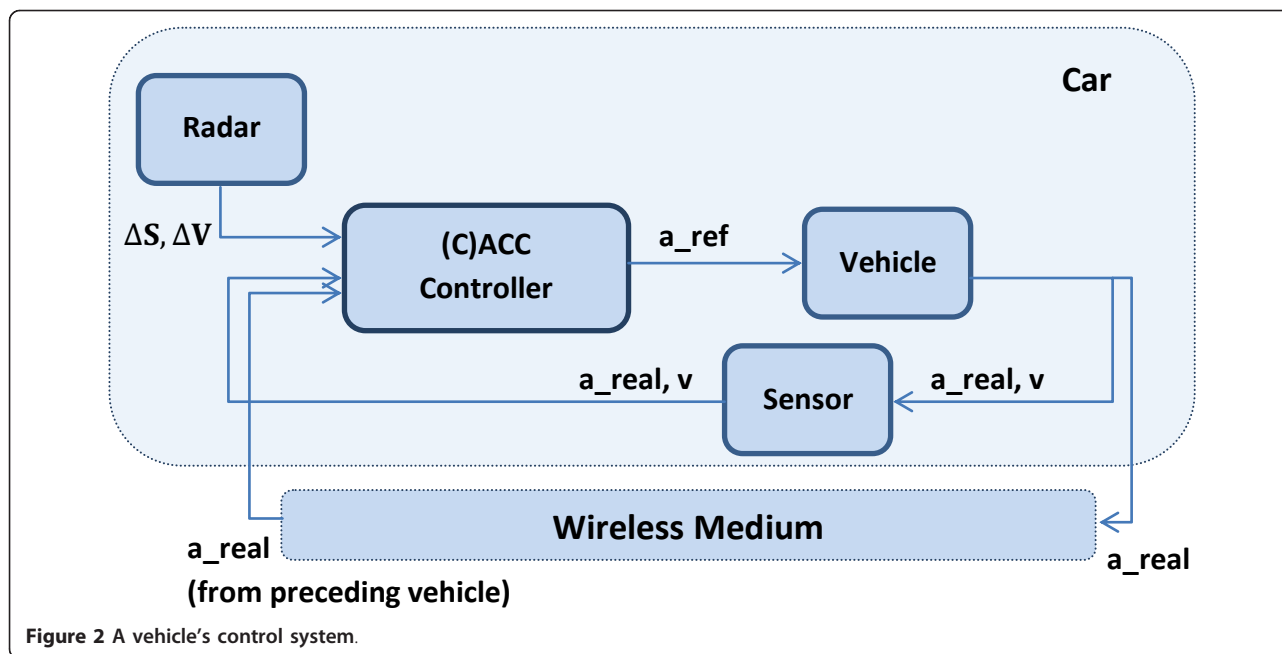
In the "Car" part of Figure 2, the module "(C)ACC Controller" together with the module "Vehicle" provides the prototype of the CACC and ACC controllers. At the beginning of each simulation step, the module "(C)ACC Controller" reads relative speed and distance to the preceding vehicle from the "Radar" module. The host vehicle's acceleration and speed are read from the module "sensor" as inputs. In addition, CACC would read the acceleration of the preceding vehicle from the "Wireless Medium" by Wi-Fi interface, which is not necessary for ACC. The desired time headway, desired distance at standstill, and cruise speed are pre-set before the simulation starts. The control objective is to realize a desired distance, taking into account a pre-defined maximum speed, referred to as the cruise speed. Note that the cruise speed is a maximum speed when the vehicle operates in (C)ACC mode. If there is no target (preceding) vehicle, the system switches to a cruise control mode, in which case the cruise speed becomes the target speed.

The time headway is the time it takes for vehicle "i" to reach the current position of its preceding vehicle "i-1" when continuing to drive with a constant velocity [9]. The primary control objective is to follow the preceding vehicle at a desired distance $D(t)$:

$$D(t) = D_{\text{standstill}} + h * V(t) \quad (1)$$

Here, " $D_{\text{standstill}}$ " denotes the desired distance to the preceding vehicle at standstill; " h " denotes the desired time headway and " $V(t)$ " is the current host vehicle velocity.

Based on these inputs, the CACC/ACC controller can calculate a reference acceleration " a_{ref} ". The "Vehicle" module takes this reference acceleration as input and mimics the response of a real vehicle, taking into account for instance the vehicle's inertia. The resulting



(or real) acceleration that is the “Vehicle” module’s output is referred to as “a real”. This acceleration is used to model the behavior of the vehicle, and to calculate the vehicle’s resulting speed “v” and position “s” at the *next* simulation time step. Module “Sensor” always outputs the *current* position and speed of a vehicle. Furthermore, “a real” is sent to module “Wireless Medium”, to model its wireless transmission. Note that when due to impairments of the wireless communication medium a packet loss occurs and an updated acceleration value is lost, then the CACC controller uses a previously received and stored acceleration value. This holds in general, as evaluation of the controller happens at a higher rate than reception of input through the wireless medium.

3.2 SUMO model

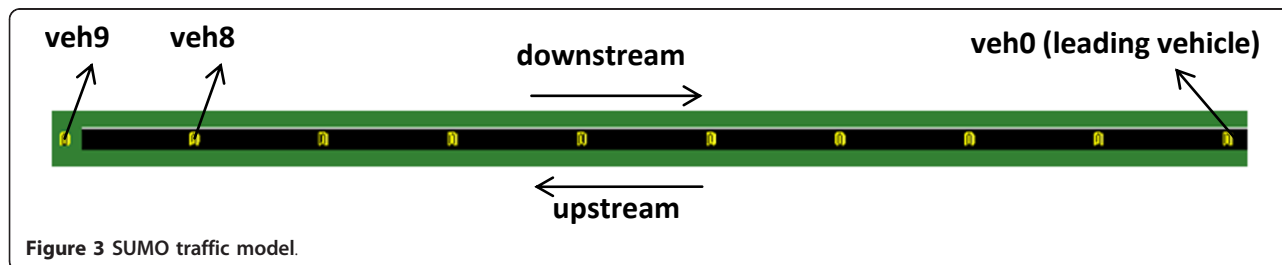
Figure 3 shows a part of the generated road network and a platoon of 10 vehicles. We mark each vehicle with an ID, where the leading vehicle’s ID is “veh0”, that of the first following vehicle is “veh1” and the last vehicle’s ID is “veh9”. The leading vehicle moves from left to

right (the downstream direction) and the other 9 vehicles equipped with ACC/CACC controllers follow the leading vehicle.

3.3 OMNeT++/MiXiM model

OMNeT++/MiXiM is used to simulate the wireless communication between vehicles in a platoon. Vehicles are simulated in the form of communication nodes, which have the same positions as the vehicles and are able to exchange information by periodically broadcasting short status messages called beacons. In this way, every vehicle (communication node) can get its preceding vehicle’s acceleration.

The OMNeT++/MiXiM model applies a Cooperative Awareness mechanism using beaconing [20]. The beaconing procedure is using a simple timer, which means that a node transmits a beacon every τ seconds (i.e., at a rate of $1/\tau$ Hz), with a small, randomly chosen variation or offset. By tuning the value of τ , the Beacon Sending Frequency (BSF) can be varied. The MAC and Physical modules used in the OMNeT++/MiXiM model are based on the IEEE 802.11p standard [1]. The model



used in this research was realized by modifying the currently available IEEE 802.11 MiXiM example, i.e., Mac80211, such that it could operate as an 802.11p model. Changes relevant for this research primarily concern different modulation types and corresponding timing parameters (such as bitrate, slotTime σ and Inter-Frame Spacings) and MAC parameters such as contention window and enhanced distributed channel access (EDCA) functionality. In order to remain tractable the physical layer uses a stochastic error model (PLR is drawn from a uniform distribution). In these experiments we consider only a one-vehicle-lookahead controller topology and vehicles are following each other, establishing line-of-sight (LoS) communication paths between relevant communication peers. Within the range of distances (headways) used in the experiments (7.7-40 m), the PLR is not expected to vary significantly with distance for LoS communication. When studying scenarios with multiple-vehicle-lookahead, or non-communicating vehicles mixed with CACC-equipped vehicles, the wireless communication needs to be performed with a more detailed model in order to account for non-Line-of-Sight situations.

3.4 Complete simulation model

The complete experiment structure can be seen in Figure 4a, while the corresponding simulation model is shown in Figure 4b. As shown in Figure 4a, corresponding to Figure 2, "Cars" equipped with "Controllers" (CACC) communicate with each other through their "Wi-Fi" interfaces. In the simulation structure shown in Figure 4b, "Cars" are simulated by the SUMO model and "Controllers" are originally built in the form of a Simulink model. MiXiM simulates the wireless transmission.

In order to allow the Simulink model to be used by vehicles implemented in SUMO, it is first converted

into a C++ shared library by using the Real-Time Workshop tool in Simulink so that it can be called in the source code of SUMO.

In this study, the method described in [21,22] is used for bidirectional coupling between OMNeT++/MiXiM and SUMO, where the two simulators communicate with each other through a traffic control interface (TraCI) by transmitting TCP messages. Here OMNeT++/MiXiM acts as the TraCI client and SUMO acts as the TraCI server.

For the simulation of CACC, four steps are noted in Figure 4b:

1. At the beginning of each simulation time step, MiXiM sends the information received from other communication nodes (i.e., preceding vehicle's acceleration) to SUMO. This information is collected by each communication node from the latest received beacon sent by its preceding vehicle, essentially implementing a sample-and-hold mechanism.
2. In the SUMO part, the acceleration received from MiXiM and the other parameters are used as inputs for the CACC controller for each vehicle as described in Section 3.1 to calculate a real acceleration and velocity ("a real" and "v" in Figure 2).
3. The resulting velocity and position are used to simulate the movement of vehicles in SUMO.
4. After moving the vehicles, SUMO will send a trace back to MiXiM which comprises the vehicles' acceleration, velocity and position generated by the CACC controller and MiXiM moves its communication nodes according to the vehicles' position information from SUMO, followed by the transmission of a beacon by each communication node if its beacon timer has expired. Recall that these events are scheduled at intervals of τ . Note that the received information is buffered before the start of the next simulation time-step.

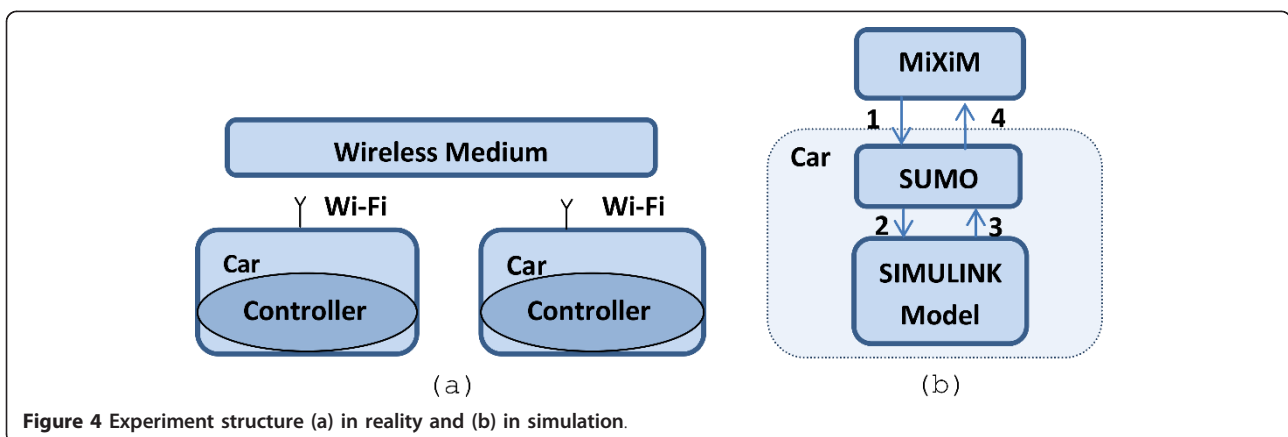


Figure 4 Experiment structure (a) in reality and (b) in simulation.

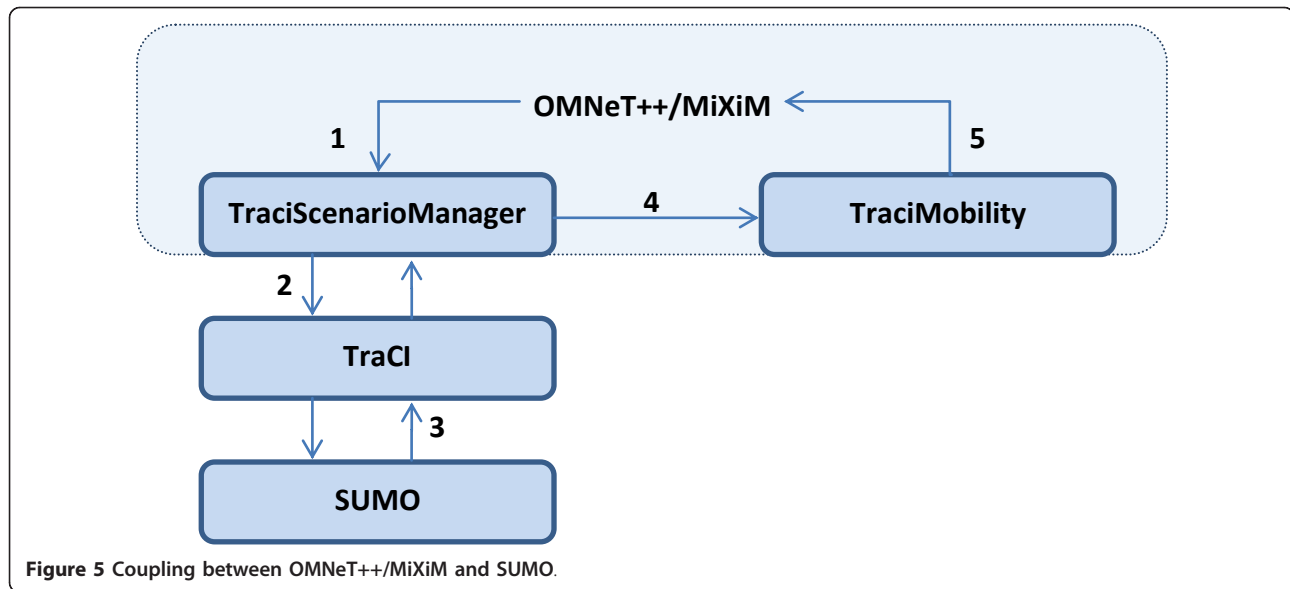


Figure 5 Coupling between OMNeT++/MiXiM and SUMO.

The functionality blocks used for the coupling between OMNeT++/MiXiM and SUMO are shown in Figure 5, and comprise:

- **TraCI:** is a traffic control interface, which is a TCP-based client/server architecture which provides access to run a roadtraffic simulation. During simulation runs, both SUMO and OMNeT++ use their communication modules to exchange commands by using TCP connections. As a TraCI client, OMNeT++/MiXiM uses the TraCIScenarioManager to communicate with the TraCI server “SUMO”.
- **TraCIMobility:** is an OMNeT++/MiXiM function that is able to move corresponding communication nodes. The communication between interacting modules in OMNeT++ is accomplished by exchanging internal messages. The mobility of communication nodes in MiXiM is realized by the TraCIMobility function once vehicles in the traffic simulation environment update their position and speed.
- **TraCIScenarioManager:** it connects OMNeT++ to a TraCI server (SUMO) running road traffic simulations. It sets up and controls the simulation experiments, moving nodes with the help of the TraCIMobility module. It makes the initialization of the stages in the simulation as well as controls the connection to the TraCI server (SUMO). The communication between OMNeT++/MiXiM and SUMO is based on the exchange of TraCI messages. The TraCIScenarioManager as being the “Manager” can for example (1) request and retrieve from SUMO the parameters such as acceleration, vehicle speed, position, and (2) request to execute commands such as creating an object. This

can however be accomplished only if these parameters and commands are defined in the “TraCIConstants.h” header file. “TraCIConstants.h” is a header file that defines all parameters allowed to be transmitted between SUMO and OMNeT++/MiXiM (e.g., acceleration, position, velocity). Moreover, this header file contains all the allowed program commands that can be executed and can be used on these parameters, e.g., “get”, “set”. The function “queryTraCI” specifies exactly which parameters (and commands) will be exchanged between SUMO and MiXiM. In this research activity the exchanged parameters are acceleration, position, velocity, and time headway. The “queryTraCI” command is sent in a message that is buffered within the “TraCIBuffer” module until the beginning of each time step. During the simulation, SUMO would execute as “queryTraCI”, executing commands and sending back information through TraCI back to OMNeT++/MiXiM.

At the beginning of each timestep, OMNeT++ would send buffered commands to SUMO by using the TraCIScenarioManager (steps 1 and 2 in Figure 5). SUMO uses this information as described in the previous subsection. Once the received information is used, SUMO sends a trace to MiXiM (step 3 in Figure 5), which includes the traffic information such as position, speed and acceleration of vehicles. In MiXiM, the communication nodes are moved discretely according to the positions received from the SUMO trace. This movement of the communication nodes is implemented by the MiXiM mobility module TraCIMobility (step 4 in Figure 5). Then the communicating nodes’ state of each vehicle is modified,

followed by the procedure of transmitting a beacon (step 5 in Figure 5). Note that the received information is buffered before the start of next simulation timestep.

For ACC simulation, we just need the SUMO model and the shared library converted from the Simulink model, as no communication is involved. More details on the implemented and used integrated simulation environment can be found in [23].

4 Experiments, results, and analysis

In order to investigate the impact of packet loss ratio and beacon sending frequency on ACC and CACC string stability performance, a simulation study was performed. By observing the speed and acceleration of following vehicles it can be investigated whether the disturbance of the leading vehicle is amplified upstream through the platoon, as was described in Section 2.2. Therefore, the vehicle speed as well as its undershoot or overshoot in situations of traffic disturbances, can be used as string stability performance measures. Vehicle speed undershoot (or overshoot) can be defined as the absolute difference between the lowest (or highest) vehicle speed of the last following vehicle and the (target) speed of the leading vehicle.

4.1 Experiment setup

The topology that is used in all our experiments is illustrated in Figure 3. A platoon of ten vehicles is placed on a straight single lane road 5,000 m long. We use a pre-defined time headway of 0.7 s. The distance at standstill is set to 7.7 m and vehicle length is 4.46 m. Varying $D_{\text{standstill}}$ is not of interest to the dynamic platoon behavior because it has no effect on the dynamic behavior, $D_{\text{standstill}}$ could be seen as a “virtual extension” of the vehicle. It is a margin a vehicle uses around its preceding vehicle. The distance between vehicles will change with different $D_{\text{standstill}}$ setting, which may impact communication. However, variations in the range [5,10] m are not of that much influence.

The upper limit of the vehicle’s acceleration is specified to be 2 m/s^2 and the minimal acceleration is specified to be -9 m/s^2 , i.e., the deceleration does not go below -9 m/s^2 . These parameters apply to all experiments in our study except for those where we investigate the influence of different time headway values. In order to guarantee a high statistical accuracy of the obtained results, multiple runs have been performed and 90% confidence intervals have been calculated. For all performed experiments, the largest calculated confidence interval is $\pm 3.1052\%$ of the shown calculated mean values.

In order to validate the controller model and traffic model, we simulate the performance of ACC (without communication), and CACC with perfect communication, where for CACC each vehicle can always get its preceding vehicle’s acceleration within SUMO without

loss and delay. The results (not shown here) are very similar to the results obtained in [9] and it is concluded in [23] that the original Simulink model and the combined SUMO-Simulink model are satisfactorily equivalent. In these baseline experiments, CACC outperforms ACC on string stability. Details of this experiment and its corresponding results and analysis can be found in [23]. This article covers a comparison between the string stability of an ACC system and of a CACC system which uses a non-perfect communication medium.

4.1.1 Simulation scenarios

The leading vehicle starts with an initial speed of 20 m/s that is kept constant until $t = 80 \text{ s}$ (i.e., up to the 8,000th simulation time step, where one time step = 10 ms). During this period each following vehicle has a stable speed (no fluctuations) of 20 m/s and distance between any two neighboring vehicles is also stable.

The leading vehicle performs a pre-set mobility pattern depending on the scenario—it either accelerates or decelerates, enabling study of these events in isolation. The metrics of interest depend heavily on the behavior of the leading vehicle and are calculated relative to the behavior of veh0 (e.g., under/overshoot of velocity). Using a step change of the lead vehicle acceleration heavily excites the system, such that the important dynamics become clearly visible.

For the first scenario, at time step 8,000, we let the leading vehicle decelerate with an acceleration of -9 m/s^2 , until the leading vehicle reaches the speed of 15 m/s. For the second scenario, at time step 8,000, we let the leading vehicle accelerate with acceleration 2 m/s^2 until the speed of the leading vehicle reaches the value of 25 m/s. For experiments in this section, the packet loss ratio (PLR) and beacon sending frequency (BSF) are varied. The chosen values of packet loss ratio are 0%, 10%, 20%, 30%, 40%, 50% and that of beacon sending frequency are 25 Hz, 20 Hz, 15 Hz, 10 Hz, 5 Hz. We also simulate the case with a default beacon sending frequency and packet loss ratio of 15 Hz and 20%, and different time headway (TH) values: 2 s, 1.5 s, 1.0 s, 0.9 s, 0.8 s, 0.7 s, 0.6 s, and 0.5 s. Note that in all these experiments packet loss is artificially accomplished in a random fashion according to a uniform distribution, see [23]. Moreover, the dropping of a packet is independent from that of other packets.

In these experiments, we are only observing the velocity response of the last following vehicle, because when the platoon is not string stable it is this vehicle that will experience the strongest disturbances.

4.2 Simulation results and analysis

For the first scenario, due to space limitations, we only show a subset of the results obtained during this

research activity. The other simulation results of vehicles with respect to both velocity and acceleration, and two-sided 90% confidence intervals for all the simulation results can be found in [23].

4.2.1 Deceleration

The curves from bottom up at the 9,000th time step of Figure 6 indicate packet loss ratio in descending order. It can be seen from Figure 6 that for a constant value of beacon sending frequency (10 Hz) and time headway (0.7 s), as the packet loss ratio increases, the velocity fluctuations of veh9 are increasing, which means that the disturbance of the leading vehicle is amplified more through the platoon upstream. In other words, the platoon is more string unstable. Moreover, the undershoot of the velocity is also getting larger as the packet loss ratio increases according to Figure 6. Figure 7 shows the acceleration corresponding to Figure 6; higher PLR results in delayed but larger response.

The undershoot of velocity for the last vehicle is shown in Figure 8. The undershoot is shown for different combinations of selected beacon sending frequencies and packet loss ratios.

From Figure 8 it follows that, with a selected value of beacon sending frequency and time headway (0.7 s), the undershoot of velocity for the last vehicle increases as the packet loss ratio increases, which means that the platoon becomes more string unstable. It can also be observed that for a selected value of packet loss ratio, the string stability becomes worse as the beacon sending frequency decreases. A vehicle always uses the acceleration value which is most recently received from its preceding vehicle as the input of the CACC controller. Therefore, a higher beacon sending frequency for the preceding vehicle results in a higher possibility of receiving fresh information under a constant packet loss ratio. Besides, lower packet loss ratio can also result in a

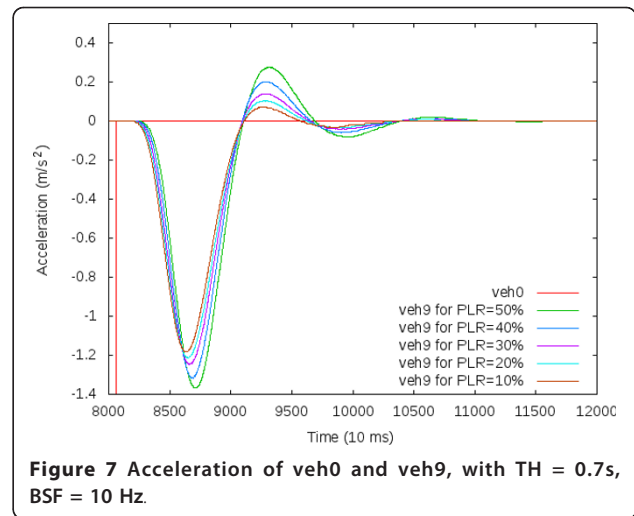


Figure 7 Acceleration of veh0 and veh9, with TH = 0.7s, BSF = 10 Hz.

higher possibility of receiving fresh information under a constant beacon sending frequency. For a BSF of 25 Hz packet loss has little effect because vehicles can still easily receive sufficiently fresh information. It should be noted, however, that at 25 Hz the wireless channel capacity will become a limiting factor when considering larger numbers of vehicles [10], so a low PLR is not always achievable.

4.2.2 Varying time headway

Figure 9 shows the velocity of the last vehicle for varied time headway settings for the BSF of 15 Hz and 20% PLR. Note that the curves from left to right at a velocity of 18 m/s of Figure 9 show the headway in ascending order. It can be seen from Figure 9 that with our selected beacon sending frequency and packet loss ratio, as time headway increases, the platoon becomes more string stable, i.e., the velocity of the last vehicle decreases with less fluctuations, findings also reported

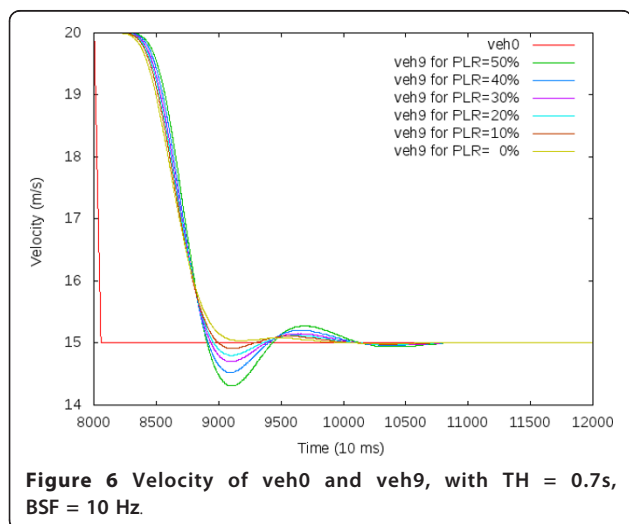


Figure 6 Velocity of veh0 and veh9, with TH = 0.7s, BSF = 10 Hz.

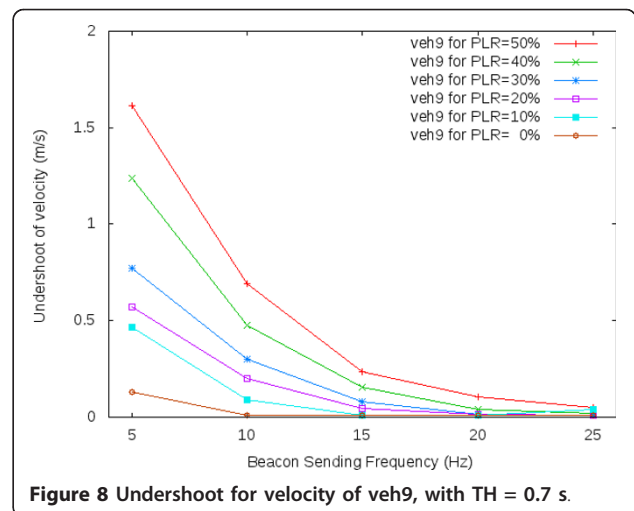
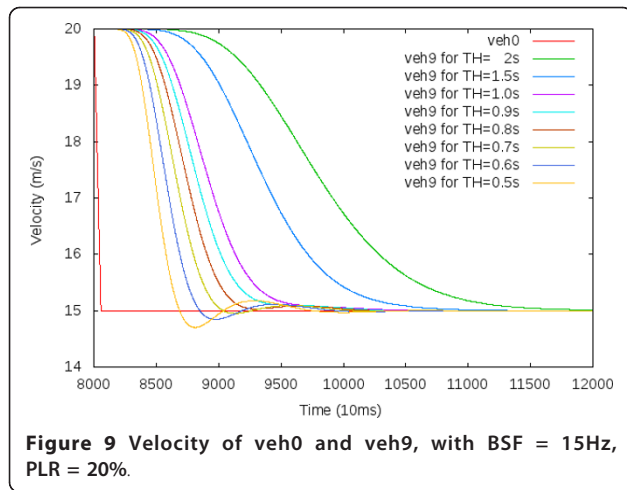
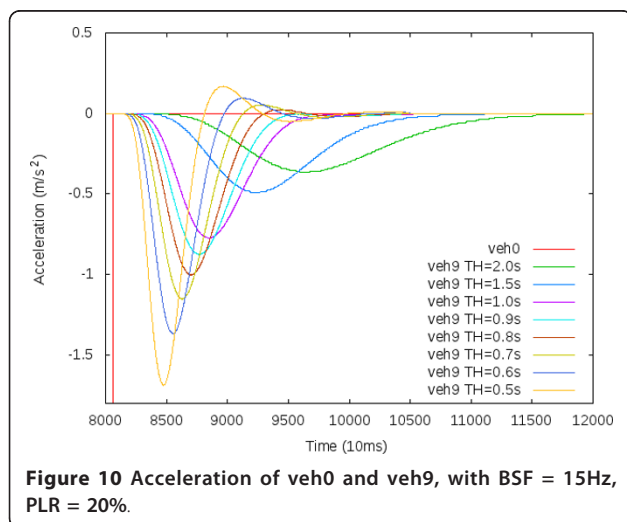


Figure 8 Undershoot for velocity of veh9, with TH = 0.7 s.



in [9,18]. Figure 10 shows the acceleration of veh9 in response to the sudden deceleration of the lead vehicle. It clearly shows that, with shorter time headway, reaction of veh9 is sooner but more aggressive than with larger time headways, even to such a degree that deceleration resembles an emergency stop.

Furthermore, with larger time headways, the relative distance between vehicles is larger. when a disturbance occurs on a leading vehicle, the following vehicles do not react as aggressively as when small time headways are used. Though large time headways may be beneficial for string stability, they are detrimental to road throughput and capacity. Therefore, it is important though challenging to find the smallest time headway to guarantee string stability, while maximizing the road capacity, especially in the face of traffic and network dynamics.



4.2.3 Acceleration

For the second scenario, we again observe the velocity of the last vehicle. The results of the simulation are plotted in similar fashion as the first scenario.

Note that the curves from top down at the 9,100th time step of Figure 11 indicate packet loss ratio in descending order. For PLR = 0%, there is hardly any overshoot. The associated acceleration is shown in Figure 12 and shows that higher PLR results in larger acceleration.

Different from Figures 8 and 13 depicts the overshoot of the velocity associated with the last vehicle. From Figures 11 and 13 it can be seen that for a given value of beacon sending frequency and time headway, the CACC controller's performance on string stability is decreasing with a higher packet loss ratio. Accordingly, for a given value of packet loss ratio and time headway, the string stability gets worse with a lower beacon sending frequency. The acceleration of veh9, plotted in Figure 12, shows that with a higher PLR reaction is later and more aggressive.

4.2.4 Varying time headway

Note that the curves from left to right at a velocity of 22 m/s of Figure 14 indicate time headway in ascending order for a BSF of 15 Hz and 20% PLR. From Figure 14, the same conclusions can be derived as the ones derived from Figure 9 in the deceleration scenario. In particular, it can be observed that string stability is improving when the time headway is increased. Figure 15 shows the corresponding acceleration curves. It becomes clear that, even in an acceleration scenario, string instability may result in sudden *deceleration* of vehicles, clearly visible for TH = 0.5 s

4.3 ACC versus CACC

The third and fourth experiment scenarios are used to compare the ACC and CACC string stability performance.

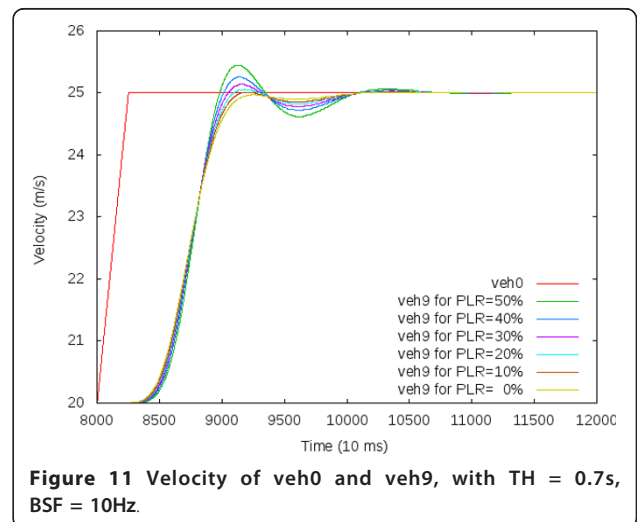
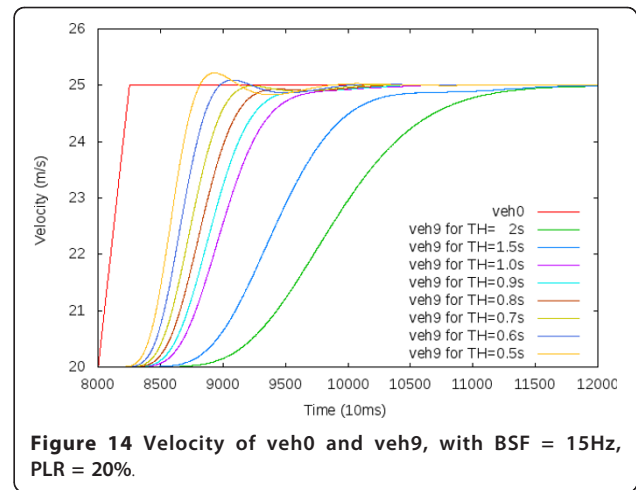
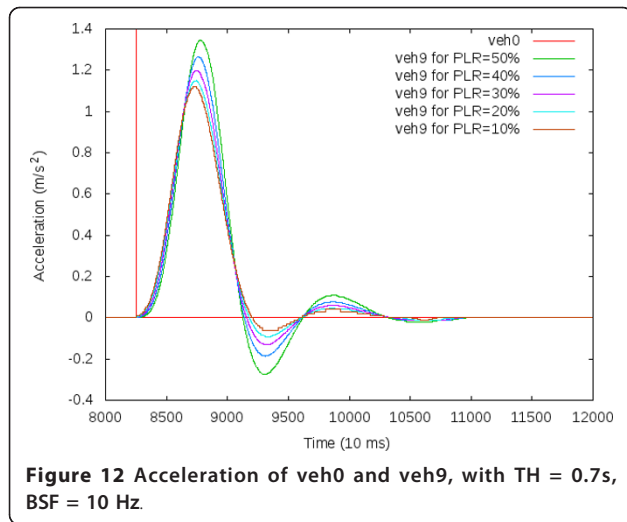


Figure 10 Acceleration of veh0 and veh9, with BSF = 15Hz, PLR = 20%.

Figure 11 Velocity of veh0 and veh9, with TH = 0.7s, BSF = 10Hz.



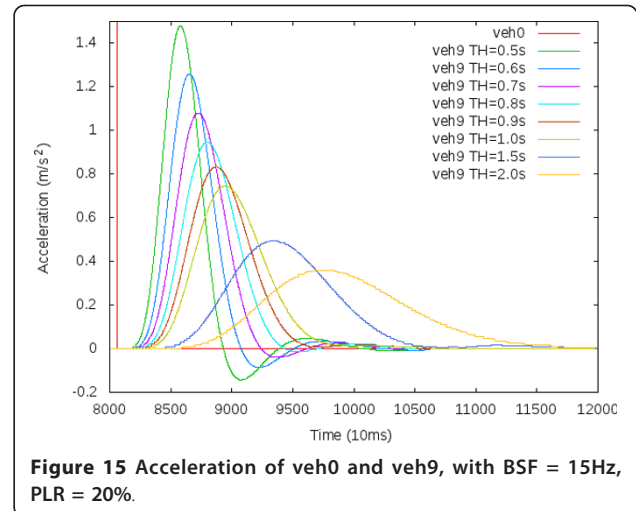
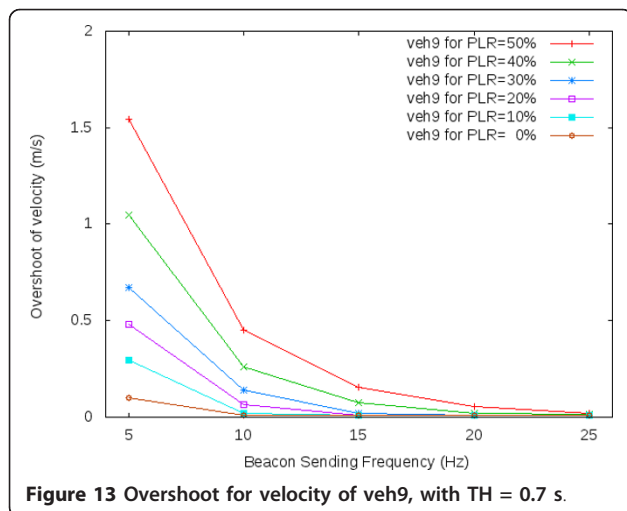
In particular, the third experiment scenario is similar to the first experiment scenario, where the lead vehicle suddenly decelerates. The undershoot of velocity for the last vehicle is shown for different combinations of selected beacon sending frequencies and packet loss ratios, see Figure 16. The fourth experiment is similar to the second scenario, where the lead vehicle suddenly accelerates. The overshoot of velocity for the last vehicle is shown for different combinations of selected beacon sending frequencies and packet loss ratios, see Figure 17. Note that in both experiments the string stability performance of the ACC system is not affected by the beacon sending frequencies and packet loss ratios, since ACC does not use beaconing, but radar measurements for the calculation of, e.g., time headway and acceleration, as modeled in the “Radar” block in Figure 2.

In both Figures 16 and 17, it can be observed that from the point of string stability performance, CACC

strongly outperforms ACC. In particular, for a beaconing packet loss ratio of 0% the CACC string stability overshoot and undershoot are more than 10 times smaller than those measured on the ACC system. Even for a beaconing packet loss ratio of 50% the CACC string stability overshoot and undershoot are more than 5 times smaller than those measured on the ACC system.

High packet loss ratio or low beacon sending frequency can result in a low refresh rate of inputs at the controller. Though differences between, e.g., 0 and 50% PLR are large, the resulting CACC still outperforms ACC, reinforcing the conclusion of [9] that the CACC feed forward controller enables smaller time headways than the ACC feedback controller, while still maintaining string stability.

This research evaluated PLR and BSF. The delay of the communicated information is minimal, as 10 vehicles do not load the wireless channel to such an extent that contention delay increases significantly. It should be



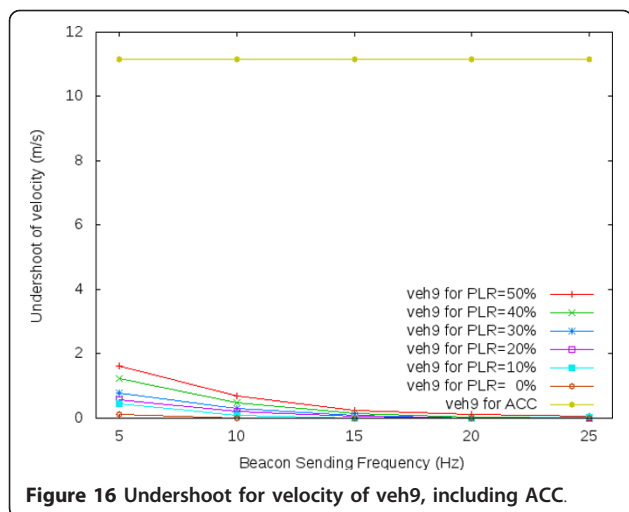


Figure 16 Undershoot for velocity of veh9, including ACC.

noted that in an 802.11p system high PLR is usually accompanied by increased delay due to its carrier sense multiple access with collision avoidance (CSMA/CA) medium access mechanism [24].

5 Conclusions and future study

In this article, the string stability of a CACC controller in the presence of imperfect communication has been investigated. For that purpose, a simulation environment integrating a time-driven controller and traffic simulations (in Simulink and SUMO, respectively) has been combined with event-driven wireless communication simulation (in MiXiM/OMNeT++). We observed that beacon sending frequency and packet loss ratio have significant influence on the performance of the evaluated CACC controller. Lower beacon sending frequencies and/or higher packet loss ratios, which prevent vehicles from receiving fresh information from preceding vehicles, will lower the CACC controller’s performance on

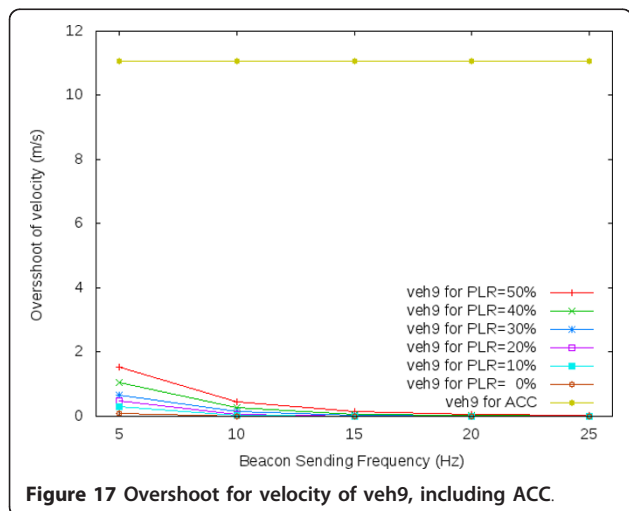


Figure 17 Overshoot for velocity of veh9, including ACC.

string stability. Therefore, given a required time headway, strict requirements with respect to beacon sending frequency and packet loss ratio have to be set in order to guarantee string stability.

The ACC and CACC systems are compared based on their string stability performance. From this comparison it can be deduced that from the point of the string stability performance, CACC strongly outperforms ACC. In particular, for a beaconing packet loss ratio of 0% the CACC string stability overshoot and undershoot are more than 10 times smaller than the ones measured on the ACC system. Even for a beaconing packet loss ratio of 50% the CACC string stability overshoot and undershoot are more than 5 times smaller than the ones measured on the ACC system. Regarding future study, we give the following recommendations: (1) study the impact of correlated (burst) losses; (2) study the impact of losses that are caused by real propagation problems or channel overload, instead of artificially generating these losses. This will also include associated delays; (3) investigate string stability by using other performance measures; (4) investigate also other types of CACC controller topologies, in addition to the one-vehicle-look-ahead topology; (5) investigate which overshoot/undershoot thresholds are acceptable for different types of traffic scenarios.

Abbreviations

ACC: Adaptive Cruise Control; BSF: Beacon Sending Frequency; CACC: Cooperative Adaptive Cruise Control; IEEE: Institute of Electrical and Electronics Engineers; OBU: On-board Unit; OMNeT++: Simulation of Urban MObility; PLR: Packet Loss Ratio; RSU: Road Side Unit; SUMO: vehicular ad hoc Network; VANET: Vehicular Ad Hoc Network; Wi-Fi: Wireless Fidelity

Acknowledgements

This study was supported by the Dutch NL Agency/HTAS (High Tech Automotive Systems) Project Connect&Drive, Project no. HTASD08002.)

Author details

¹Department of Computer Science, University of Twente, Enschede, The Netherlands ²Integrated Vehicle Safety Department, TNO Technical Sciences, Helmond, The Netherlands

Competing interests

The authors declare that they have no competing interests.

Received: 22 November 2011 Accepted: 23 March 2012

Published: 23 March 2012

References

1. IEEE80211p: 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, (2010)
2. IR Wilmink, GA Klunder, B van Arem, Traffic flow effects of integrated full-Range speed assistance (IRSA). in *IEEE Intelligent Vehicles Symposium (IV07)*, Istanbul, Turkey, 1-3, 1078-1084 (2007)
3. I Vreeswijk, B van Arem, K Malone, C van Driel, Deployment Scenarios for Speed Assistance Systems, in *the 15th World Congress on Intelligent Transport Systems*, New York, USA, 1-10 (2008)

4. R Pueboobpaphan, B van Arem, Understanding the relation between driver and vehicle characteristics and platoon and traffic flow stability for design and assessment of cooperative adaptive cruise control. *Transportation Research Board (TRB) 89th Annual Meeting*, Washington DC, USA, 1–17 (2010)
5. A Bose, P Ioanno, Analysis of traffic flow with mixed manual and intelligent cruise control (ICC) vehicles: Theory and experiments, in *California PATH Res. Rep. UCB-ITS-PRR-2001-13*, 6 i–63 (2001). ISSN 1055-1425
6. B van Arem, C Tampere, K Malone, Modelling traffic flows with intelligent cars and intelligent roads, in *IEEE Intelligent Vehicle Symposium (IV03)*, Columbus, OH, USA, 456–461 (2003). ISBN 0-7803-7848-2
7. P Ioannou, *Automated Highway Systems*, (Plenum Press, New York, 1997)
8. D Reichardt, M Miglietta, L Moretti, P Morsink, W Schulz, CarTALK 2000: safe and comfortable driving based upon inter-vehicle-communication, in *IEEE Intelligent Vehicle Symposium (IV02)*, 2, 545–550 (2002). ISBN 0-7803-7346-4
9. GJL Naus, RPA Vugts, J Ploeg, MJGv Molengraaf, M Steinbuch, String-Stable CACC design and experimental validation: a frequency-domain approach. *IEEE Trans Veh Technol.* **59**, 4268–4279 (2010)
10. EM van Eenennaam, W Klein Wolterink, G Karagiannis, GJ Heijen, Exploring the Solution Space of Beaconing in VANETS. *IEEE Vehicular Networking Conference, VNC*, Tokyo, Japan, 1–8 (2009). ISBN 978-1-4244-5685-7
11. D Krajzewicz, G Hertkorn, C Rössel, P Wagner, SUMO (Simulation of Urban MObility); An open-source traffic simulation, in the *4th Middle East Symposium on Simulation and Modeling (MESM2002)*, Sharjah, United Arab Emirates, 183–187 (2002). ISBN 907-7-039-090
12. Simulink introduction in mathworks official website (visited in May 2011) <http://www.mathworks.com/products/simulink/description1.html>
13. A Varga, The OMNeT++ discrete event simulation system, in *Proceedings of the European Simulation Multiconference ESM2001*, Prague, Czech Republic, 319–324 (2001)
14. MiXiM - Simulation Framework for Wireless and Mobile Networks. <http://mixim.sourceforge.net>
15. C Lei, EM van Eenennaam, W Klein Wolterink, G Karagiannis, GJ Heijen, J Ploeg, Impact of Packet Loss on CACC String Stability Performance, in *Proceedings of the Eleventh International Workshop on ITS Telecommunications*, IEEE Communications Society, Saint-Petersburg, Russia, 381–386 (2011). ISBN 978-1-61284-668-2
16. S Kumarawadu, *Control systems: Theory and Implementation*, (Alpha Science International, Oxford, UK, 2010)
17. J Ploeg, A Serrarens, G Heijen, Connect & Drive: design and evaluation of cooperative adaptive cruise control for congestion reduction. *J Modern Transport.* **19**(3), 207–213 (2011)
18. J Ploeg, B Scheepers, E van Nunen, N van de Wouw, H Nijmeijer, Design and experimental evaluation of cooperative adaptive cruise control, in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington DC, USA, 260–265 (2011). ISBN 978-1-4577-2198-4
19. B van Arem, CJG van Driel, R Visser, The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Trans Intell Transport Syst.* **7**, 429–436 (2006)
20. EM van Eenennaam, G Karagiannis, G Heijen, Towards scalable beaconing in VANETS, in the *Fourth ERCIM workshop on eMobility*, Luleå, Sweden, 103–108 (2010). ISBN 978-91-7439-103-9
21. C Sommer, Z Yao, R German, F Dressler, On the need for bidirectional coupling of road traffic microsimulation and network simulation, in *1st ACM SIGMOBILE International Workshop on Mobility Models for Networking Research (MobilityModels'08)*, Hong Kong, China, 44–48 (2008). ISBN 978-1-60558-111-8
22. C Sommer, R German, F Dressler, Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Trans Mobile Comput.* **10**, 3–15 (2011)
23. C Lei, Cooperative Adaptive Cruise Control model study based on traffic & network simulation. http://www.utwente.nl/ewi/dacs/assignments/completed/bachelor/reports/2011-lei_chenxi.pdf
24. R Reinders, EM van Eenennaam, G Karagiannis, GJ Heijen, Contention window analysis for beaconing in VANETS, in *Seventh IEEE International Wireless Communications and Mobile Computing conference, IWCMC 2011*, IEEE Computer Society, Istanbul, Turkey, 1481–1487 (2011). ISBN 978-1-4244-9539-9

doi:10.1186/1687-1499-2012-116

Cite this article as: Lei et al.: Evaluation of CACC string stability using SUMO, Simulink, and OMNeT++. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:116.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
