Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis

Christoph Sommer, *Student Member*, *IEEE*, Reinhard German, and
Falko Dressler, *Senior Member*, *IEEE*

Abstract—Recently, many efforts have been made to develop more efficient Inter-Vehicle Communication (IVC) protocols for on-demand route planning according to observed traffic congestion or incidents, as well as for safety applications. Because practical experiments are often not feasible, simulation of network protocol behavior in Vehicular Ad Hoc Network (VANET) scenarios is strongly demanded for evaluating the applicability of developed network protocols. In this work, we discuss the need for bidirectional coupling of network simulation and road traffic microsimulation for evaluating IVC protocols. As the selection of a mobility model influences the outcome of simulations to a great extent, the use of a representative model is necessary for producing meaningful evaluation results. Based on these observations, we developed the hybrid simulation framework *Veins* (Vehicles in Network Simulation), composed of the network simulator OMNeT++ and the road traffic simulator SUMO. In a proof-of-concept study, we demonstrate its advantages and the need for bidirectionally coupled simulation based on the evaluation of two protocols for incident warning over VANETs. With our developed methodology, we can advance the state-of-the-art in performance evaluation of IVC and provide means to evaluate developed protocols more accurately.

Index Terms—Inter-Vehicle Communication, Vehicular Ad Hoc Network, road traffic simulation, network simulation, Veins.

1 Introduction

Realistic simulation of Inter-Vehicle Communication (IVC) protocols is one of the main challenges in the Vehicular Ad Hoc Network (VANET) research domain. In this paper, we investigate the need for bidirectional coupling of network and road traffic simulation for more realistic simulation experiments. Incident detection, e.g., traffic jam and accident detection, for optimized route planning, as well as future safety applications, e.g., enhanced lane and distance control, is the dominant driver for the development of adequate IVC protocols [1], [2], [3]. Performance evaluation of developed IVC protocols is typically accomplished by means of simulation techniques because realistic field tests are still infeasible or are limited to a few hundred cars.

It has long been established that the quality of results obtained from Mobile Ad Hoc Network (MANET) simulations is heavily influenced by the quality of the employed mobility model [4]. The impact of mobility models on VANET simulation results, as well as the inadequacy of the mobility models usually adopted in MANET simulations [5], is well-documented in the literature [6], [7]. For this reason, recent work in the field of VANET simulation commonly uses mobility traces which are more appropriate for VANETs. Such traces can be generated by real-world experiments, i.e., the observation of real road traffic behavior, or by a dedicated road traffic simulator before being fed as input to a network simulation environment.

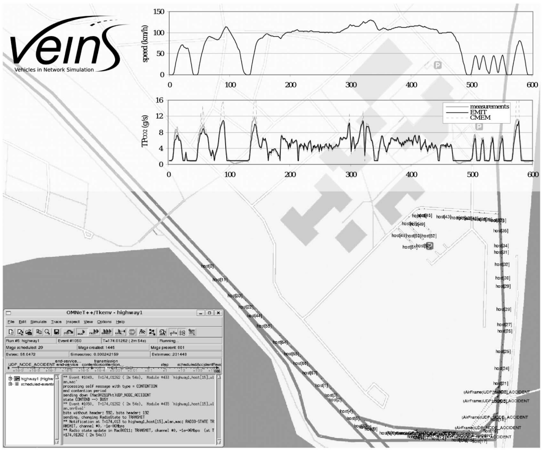
When these road traffic simulators are employed in VANET simulations, traces are commonly generated offline to speed up network simulation performance, which can then reuse generated trace files. However, one major drawback of using offline mobility traces, both pregenerated ones and those obtained from real-world measurements, is that they can only model the influence of road traffic on network traffic, but not vice versa. In order to achieve this goal, more sophisticated simulation techniques such as bidirectional coupling between a network traffic simulator and a road traffic simulator are needed [8].

We developed a special simulation framework that provides coupled network and road traffic simulation using well-established simulators from both communities. In particular, we employ OMNeT++ [9], a simulation environment free for academic use, to model realistic communication patterns of VANET nodes. Traffic simulation is performed by the microscopic road traffic simulation package SUMO [10]. Developed by German research organizations DLR and ZAIK, this simulator is in widespread use in the research community, which makes it easy to compare results from different network simulations. Availability of both simulators' C++ source code under the terms of a GPL license made it possible to integrate all needed extensions into the respective simulation cores. An overview of the resulting coupled simulation framework, which we named *Veins* (Vehicles in Network Simulation)[1] [11], is given in Fig. 1. In this figure, some network nodes, i.e., cars, are shown that are controlled by OMNeT++, which also manages all the IVC protocol events. The mobility is managed by SUMO using accurate street maps as a basis—in this example, we show the map of a part of Erlangen.

- *The authors are with Computer Networks and Communication Systems, Department of Computer Science, University of Erlangen, Martensstr 3, 91058 Erlangen, Germany.*
  E-mail: {christoph.sommer, german, dressler}@informatik.uni-erlangen.de.

Manuscript received 4 May 2009; revised 17 Dec. 2009; accepted 31 Mar. 2010; published online 6 July 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-05-0157.
Digital Object Identifier no. 10.1109/TMC.2010.133.

[1] http://www7.informatik.uni-erlangen.de/veins.



Run #5: highway1 | Event #1050 | T=174.01262 (2m 54s) | Running...

Msgs scheduled: 29 | Msgs created: 1446 | Msgs present: 801

Ev/sec: 56.0472 | Simsec/sec: 0.000242159 | Ev/simsec: 231448

```
** Event #1049.  T=174.01262 ( 2m 54s).  Module #433 `highway1.host[15].wl
an.mac'
processing self message with type = CONTENTION
end contention period
sending down (Mac80211Pkt)UDP_NODE_ACCIDENT
state CONTEND --> BUSY
** Event #1050.  T=174.01262 ( 2m 54s).  Module #435 `highway1.host[15].wl
an.snrEval'
bits without header: 592, bits header: 192
sending, changing RadioState to TRANSMIT
** Notification at T=174.013 to highway1.host[15].wlan.mac: RADIO-STATE TR
ANSMIT, channel #0, -1e-06Mbps
** Radio state update in Mac80211: TRANSMIT, channel #0, -1e-06Mbps  (at T
=174.01262 ( 2m 54s))
```

Fig. 1. Bidirectionally coupled network and road traffic simulation. Shown are vehicles controlled by Veins that use an IVC protocol for communication.

This paper addresses the need for bidirectional coupling of realistic mobility models with network simulation tools in evaluations of VANET protocols. We present such a means of bidirectional coupling, which allows the network simulation to directly control the road traffic simulation, and thus, to simulate the influence of VANET communications on road traffic. Similarly, the road traffic simulation provides information to the network simulation. For example, the vehicles' positions or planned trips can be provided to an application layer Traffic Information System (TIS). In addition, we integrated measures for the environmental impact of the road traffic by implementing a model for CO_2 emission of the cars. The key contributions of this paper can be summarized as follows:

- *Need for bidirectional coupling of network simulation and road traffic microsimulation:* Bidirectional coupling of network and road traffic simulators provides two main benefits: First, deeper insights into the impact of network protocols on the road traffic and into interactions between both can be derived. Second, the obtained results of IVC studies become more realistic. Therefore, we advocate the use of such coupled techniques.
- *Coupled simulation architecture and tool support:* Based on the OMNeT++ network simulation and the SUMO road traffic microsimulation tools, we developed an integrated VANET simulator, *Veins*, which allows dynamic interaction between both simulators. It is freely available under the terms of the GPL.
- *Proof-of-concept for studying IVC using incident warnings:* We used *Veins* to evaluate the effects of two different IVC protocols: In one set of experiments, a centralized TIS using a preinstalled infrastructure and standard MANET routing, in another set a decentralized/self-organizing TIS without such infrastructure using UDP-based broadcasting. Both IVC approaches are tested in a simple Manhattan Grid scenario as well as in a realistic environment using accurate street maps. With the help of an integrated model for CO_2 emissions, we demonstrate the feasibility of environmental impact studies.

2 Related Work

Traditionally, the mobility models used in many network simulation tools do not take into account driver behavior or specific characteristics of the urban environment (presence of stop lights, intersections, merge lanes, etc). As a result, the used mobility models, and thus, the simulation of network protocols may be unrealistic [12].

One major advancement in this domain was the concept of trace-based mobility modeling to be used in network simulation environments. Here, realistic mobility patterns are generated (offline), then used as representative models for the evaluation of network protocols. In fact, as a common practice in many simulation platforms, the mobility traces are inserted into network simulation modules as independently generated offline files. This way, the system complexity is reduced. Two methods for the generation of such trace files can be distinguished. First, real-world observations can be used, i.e., the mobility of real vehicles is observed in a city or highway environment and the resulting trace information is processed for use in network simulations [13], [14]. Similarly, mobility patterns can be extracted from these real-world observations to analytically model traffic flows [15]. Critical is that only specific scenarios can be analyzed, so the input parameter range for the network simulation is relatively fixed—even though realistic traces have been used.

Another approach is to employ traffic microsimulation tools that feed data to network simulators. An early example is based on the integration of VISSIM traces with the network simulator ns-2 [16], a frequently used simulation framework. Hybrid simulation and mathematical modeling have recently been combined in order to speed up the simulation process [17]. Also, our preliminary work facilitated network and road traffic simulation using a road traffic microsimulation model based on the IDM/MOBIL models together with the OMNeT++ network simulation tool [18].

Nevertheless, such "decoupling" design philosophy faces one dilemma: If the results from the network simulation module can affect the mobility trace, this methodology is unable to generate the real-time interaction between the mobility model simulation module and the network simulation module. For example, in vehicular safety applications, vehicles will generate alert messages to change the mobility patterns of other vehicles. In this case, the network simulation model and the mobility simulation model need to interact with each other in a real-time manner. Whenever IVC does not influence the mobility of the vehicles, the use of traces is sufficient. In most cases, however, IVC will be used to distribute traffic related information. Thus, the vehicles will probably change their speed or route according to the received information.

This problem has been addressed with the NCTUns simulation environment [19]. This tool allows integrated network and traffic simulation. The main problem of this tool, which has been developed from scratch, is that the models in both domains (network and road traffic microsimulation) are hard to compare with well-tested models using standard simulation environments. Additionally, the manifold implementations of models for various network protocols, available for, e.g., ns-2 or OMNeT++, cannot be used.

For these reasons, some of the presented simulation frameworks have recently started to adopt a generalized traffic control interface. For example, the SUMO traffic microsimulation tool has been integrated with ns-2, resulting in the hybrid simulation framework TraNS [20]. TraNS used an active loop to send control commands to SUMO. This allows to stop vehicles or to change their routes. However, in the early definition of TraNS, the simulation time control has not been considered.

Veins contributes to the state-of-the-art with a fine-grained control interface between both simulation domains. This not only supports the active exchange of control and statistics data but also the real-time interaction between the network simulation and the road traffic microsimulation as well as the collection of simulation results on the fly. For these reasons, some of the presented simulation frameworks have recently started to adopt a generalized traffic control interface, which we briefly discuss in Section 3.4.

3 Veins—Vehicles In Network Simulation

The developed simulation framework *Veins* incorporates all the benefits from state-of-the-art simulation techniques of both the network simulation and the road traffic microsimulation domains. Before we outline the architecture and behavior of the bidirectionally coupled simulation using *Veins*, we briefly introduce the mentioned simulation domains and motivate our selection of adequate candidate tools.

3.1 Network Simulation

Network simulation is commonly used to model computer network configurations long before they are deployed in the real world. Through simulation, the performance of different network setups can be compared, making it possible to recognize and resolve performance problems without the need to conduct potentially expensive field tests. Network simulation is also widely used in research, in order to evaluate the behavior of newly developed network protocols [21].

In most cases, network protocols are analyzed using discrete event simulation and a large number of simulation frameworks are available in this domain. Examples of such frameworks are open source tools such as the network simulator ns-2 [22], OMNeT++ [9], J-SIM [23], and JiST/SWANS [24] and commercial tools like OPNET. In our evaluation, we are using the network simulator OMNeT++, together with its *INET Framework* extension, for simulating VANET protocols. Aside from the level of support for a large number of nodes, the working principles of all these simulators are similar and the differences lie mostly in the number of available models, i.e., typical MAC, routing, and other Internet protocols. Thus, in this section, we provide a short overview to event-based network simulation using OMNeT++. Without losing generality, it can be said that the same techniques apply to other simulation tools as well.

Scenarios in OMNeT++ are represented by a hierarchy of reusable modules written in C++. Modules' relationships and their communication links are stored as Network Description (NED) files and can be modeled graphically. Simulations are either run interactively, in a graphical environment, or are executed as command-line applications. The *INET Framework* provides a set of OMNeT++ modules that represent various layers of the Internet protocol suite, e.g., the TCP, UDP, IPv4, and ARP protocols. It also provides modules that allow the modeling of spatial relations of mobile nodes and IEEE 802.11 transmissions between them.

Mobility support for network simulations is limited to simple mobility patterns. Examples that are available in OMNeT++ are the Random Waypoint or mass-based mobility models. It is widely accepted that such simple mobility patterns cannot be used for experiments in VANET scenarios as road traffic patterns strongly differ from such simple mobility models. Alternatively, trace-driven simulation can be employed for the evaluation of IVC protocols. In this case, node mobility patterns are either obtained by observations of real traffic or by using road traffic microsimulation tools described below. The problem of such a trace-driven simulation is that here node movement is statically computed and cannot be influenced by IVC, which means that, e.g., a hazard warning will have absolutely no effect on traffic density and driver behavior.

3.2 Road Traffic Microsimulation

As mentioned before, for the most realistic simulation of moving nodes, their mobility would need to be modeled according to trace files obtained in real-world measurements. However, even if such trace files could be readily created for a specific scenario, simulations could still only be performed for exactly the scenario one was able to gather movement traces for. Varying only a single parameter, e.g., traffic density, and keeping all other parameters unchanged, would be infeasible with this approach. Full control over all aspects of the scenario can, however, be achieved if movement traces are generated by traffic simulation tools. This also opens up the possibility to generate movement traces on the fly, a prerequisite for closing the feedback loop and allowing network simulations to influence traffic simulations, as is commonly desirable in settings where information relevant to the traffic situation is being exchanged between nodes.

Traditionally, road traffic simulation models are classified into Macroscopic, Mesoscopic, and Microscopic models, according to the granularity with which traffic flows are examined. Macroscopic models, like METACOR [25], model traffic at a large scale, treating traffic like a liquid and often applying hydrodynamic flow theory to vehicle behavior. Mesoscopic models, like CONTRAM [26], are concerned with the movement of whole platoons, using, e.g., aggregated speed-density functions to model their behavior. Simulations of VANET scenarios, however, are concerned with the accurate modeling of single radio transmissions between nodes, and therefore, require exact positions of simulated nodes. Both Macroscopic and Mesoscopic models cannot offer this level of detail, so only Microscopic simulations, which model the behavior of single vehicles and interactions between them, will be considered as mobility models for simulated VANET nodes.

Transportation and traffic science has developed a number of microsimulation models, each taking a different approach and thus each resulting in simulations of different

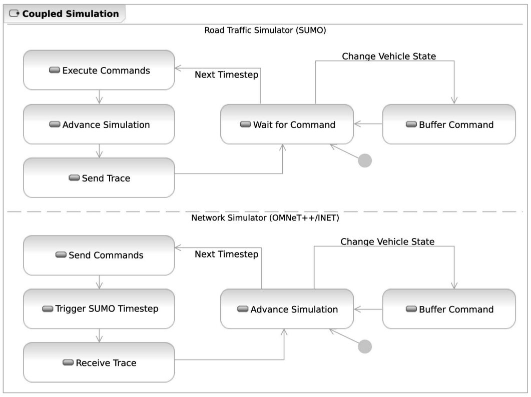


Fig. 2. Overview of the coupled simulation framework. State machines of road traffic and network simulator communication modules.

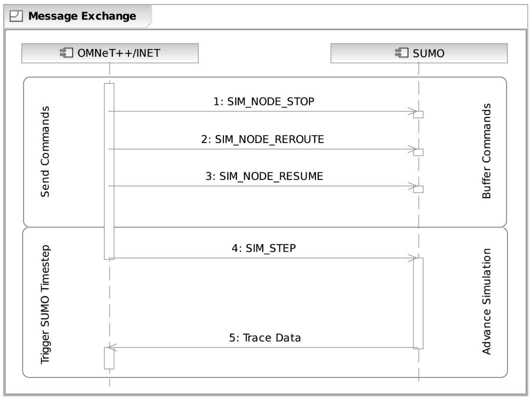


Fig. 3. Sequence diagram of messages exchanged between network and road traffic simulator communication modules. Command execution is delayed until the next road traffic simulation timestep is triggered.

complexity. Models that are in widespread use within the traffic science community include the Cellular Automaton (CA) model [27], [28], the car-following model by Stefan Krauß (SK) [29], and the IDM/MOBIL models [30], [31]. When doing road traffic simulation, each approach has its particular advantages and particular drawbacks. However, the accuracy of many of these models was evaluated in [32], which concluded that, as far as network simulation is concerned, all common road traffic microsimulation approaches are of equal value as a mobility model.

Traffic simulation in *Veins* is performed by the microscopic road traffic simulation package SUMO, which uses the aforementioned SK mobility model, can perform simulations both running with and without a GUI, and imports city maps from a variety of file formats. SUMO allows high-performance simulations of huge networks with roads consisting of multiple lanes, as well as of intrajunction traffic on these roads, either using simple right-of-way rules or traffic lights. Vehicle types are freely configurable with each vehicle following statically assigned routes, dynamically generated routes, or driving according to a configured timetable.

The use of such microscopic road traffic simulation in combination with IVC protocol analysis using a state-of-the-art network simulator can provide deeper insights into the behavior of VANET protocols than is possible with one alone. This is especially the case if IVC can directly influence the road traffic, e.g., through incident warnings or other traffic messages. Such an evaluation requires a bidirectional coupling of both simulators.

3.3 Bidirectionally Coupled Simulation

We achieved bidirectional coupling of both frameworks, the network simulator OMNeT++ and the road traffic simulator SUMO, by extending each with a dedicated communication module. During simulation runs, these communication modules exchange commands, as well as mobility traces, via TCP connections.

OMNeT++ is an event-based simulator, so it handles mobility by scheduling node movements at regular intervals. This fits well with the approach of SUMO, which also advances simulation time in discrete steps. As can be seen in Fig. 2, the control modules integrated with OMNeT++ and SUMO were able to buffer any commands arriving in-between timesteps to guarantee synchronous execution at defined intervals. At each timestep, OMNeT++ would then send all buffered commands to SUMO and trigger the corresponding timestep of the road traffic simulation. Upon completion of the road traffic simulation timestep, SUMO would send a series of commands and the position of all instantiated vehicles back to the OMNeT++ module. This allows OMNeT++ to react to the received mobility trace by introducing new nodes, by deleting nodes that had reached their destination, and by moving nodes according to their road traffic simulation counterpart. After processing all received commands and moving all nodes according to the mobility information, OMNeT++ would then advance the simulation until the next scheduled timestep, allowing nodes to react to altered environmental conditions, i.e., the IVC is influencing their speed and routes.

Fig. 3 shows the interaction between both simulators in the form of a message sequence chart. Using a simple request/response protocol, road traffic in SUMO can be influenced by OMNeT++ in a whole number of ways. Most importantly, time steps are generated to advance the simulation in SUMO. Furthermore, vehicles can be stopped to create artificial traffic jams, they can be resumed to resolve those jams, and each simulated vehicle can be individually rerouted around arbitrary road segments. This way, Veins accurately reflects how drivers that know about a traffic obstruction will try to avoid it. More recently, commands have been included to model vehicles slowing down, e.g., in front of hazards, as well as commands that allow vehicles and network components to interact with traffic lights.

Also illustrated in Fig. 3 are the alternating two phases of coupled simulation which result from this approach. In the first phase, commands are sent to SUMO, and in the second phase, their execution is triggered and the resulting mobility trace received. This way, both simulators are tightly coupled and SUMO is only able to perform a simulation step after all events within a time step have been processed in the network simulation. The network simulator advances the road traffic microsimulation only at fixed intervals. That means that the granularity of these intervals needs to be sufficiently fine-grained to obtain realistic results. From a performance point of view, this is not a

```
tsp 0
add host[0000] Car;i=car0_vs;r=0,,#707070,1
mov host[0000] 998.35  4995.00   0.00  0901
tsp 8
add host[0002] Car;i=car1_vs;r=0,,#707070,1
mov host[0002] 998.35  4993.42   0.00  0901
mov host[0001] 998.35  4976.32   6.74  0901
mov host[0000] 998.35  4943.28   9.83  0901
[...]
tsp 529
del host[0000]
mov host[0003] 3786.65  998.35  13.89  0404
mov host[0002] 3911.91  998.35  13.90  0404
mov host[0001] 3954.35  998.35  13.89  0404
```

Fig. 4. Excerpt of the movement trace, as sent by the road traffic simulator.

problem because the road traffic microsimulation can be processed much faster compared to the simulation of wireless networks.

Fig. 4 shows a small sample of the command and mobility trace stream sent by SUMO to the network simulation. To guarantee synchronicity of both simulators, each timestep is signaled by one `tsp` command containing the current simulation time. Using the `add` command, the road traffic simulation is able to introduce new vehicles entering the road traffic simulation, to be represented by an arbitrary OMNeT++ module. In the example shown, new modules of type "Car" using images "car0_vs" and "car1_vs" are instantiated at timesteps 0 and 8. Similarly, the road traffic simulation is able to remove from the network simulation all vehicles that have reached their destination by issuing `del` commands. Mobility traces are communicated by transmitting the current speed and position of all instantiated nodes as a series of `mov` commands, the position being expressed as both OMNeT++ simulation coordinates and SUMO lane identifier.

Fig. 5 shows screenshots of the GUI versions of both simulators running a coupled simulation of IVC among traffic streams merging at an intersection. As can be seen, the positions of the nodes, which are represented as small dots in OMNeT++, are exactly the same as provided by the vehicles' position in SUMO. The OMNeT++ screenshot also shows an active radio broadcast from a Roadside Unit (RSU) to all neighboring cars.

3.4 Steps Toward a Common Interface

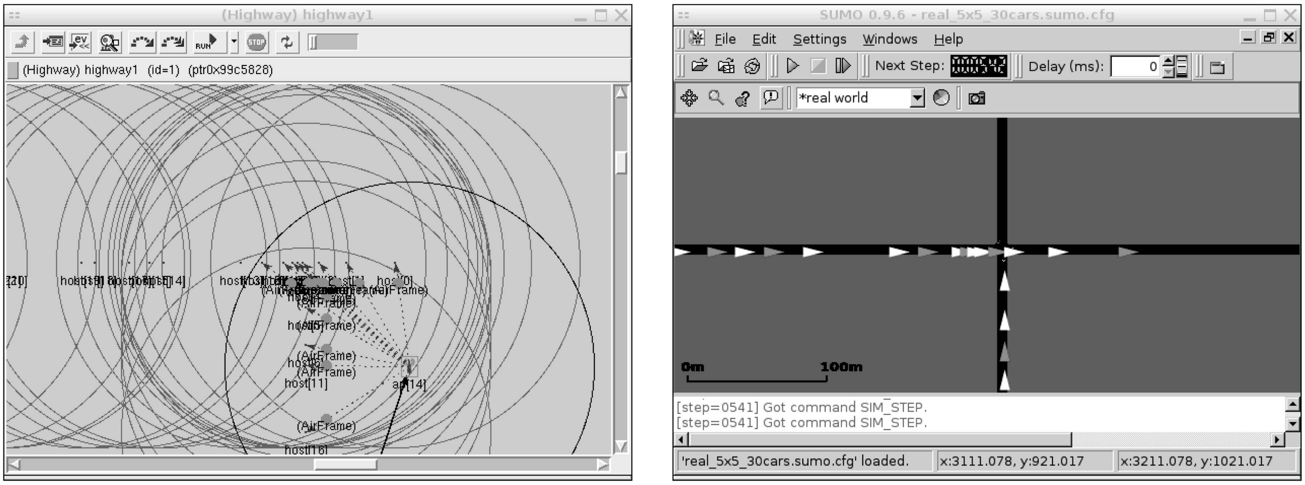
For all reasons outlined in the last chapters, coupled simulation of network traffic and road traffic is a technique that is being increasingly adopted for research in the VANET domain.

In order to foster comparability of research results and to allow realistic simulations of VANET protocols with any of a whole range of network simulators, a common interface between road traffic and network traffic simulators is highly desired.

The presented approach has therefore been integrated together with other approaches into a generalized traffic control interface, TraCI [33], which uses a very similar command-response approach and a TCP connection. Clients of a road traffic simulation can rely on a flexible, subscription-based system to receive information about events in the mobility model. There are ongoing implementation efforts to integrate this interface for the coupling of SUMO and AnSim mobility models with the simulators ns-2, Shawn, JiST/SWANS, and OMNeT++.

4 The Impact of IVC on Mobility—Selected Proof-of-Concept Examples

In the following, we demonstrate the advantages of bidirectionally coupled network and road traffic simulation based on a number of proof-of-concept examples using *Veins*. In particular, we evaluate the effects of two different IVC protocols: a centralized TIS using a preinstalled infrastructure and standard MANET routing versus a decentralized/self-organizing TIS without such infrastructure relying on UDP-based broadcasting. IVC is analyzed in two scenarios: a simple Manhattan Grid topology versus a



[step=0541] Got command SIM_STEP.
[step=0541] Got command SIM_STEP.

'real_5x5_30cars.sumo.cfg' loaded. x:3111.078, y:921.017 x:3211.078, y:1021.017

(a)

(b)

Fig. 5. Screenshots of simulators' graphical user interfaces running network and road traffic simulations in parallel. While the graphical representation requires substantial computational resources, it is very helpful for debugging the simulation scenario. (a) OMNeT++/INET. (b) SUMO.

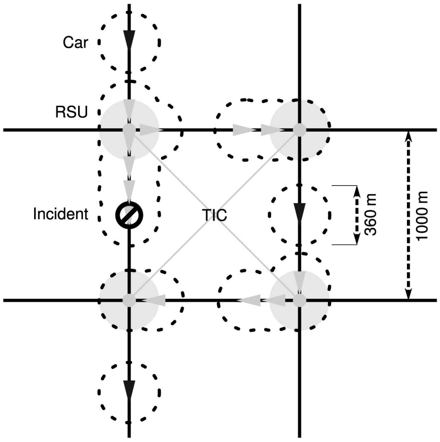


Fig. 6. TCP IVC scenario. Communication is supported by RSUs.

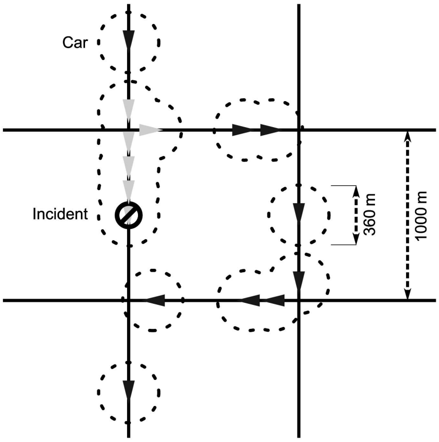


Fig. 7. UDP IVC scenario. Communication relies on the VANET alone.

more realistic setup based on real street maps of the city of Erlangen. The simulation results clearly outline the impact of IVC on the mobility of the vehicles.

In both scenarios, vehicles with a speed of zero, after some time (in the simulations, we used about 1 s for this delay), try to inform other vehicles of a potential incident on the current lane. If the network simulation determines that such an incident warning is eventually received by another vehicle, it will store information about the timestamp and contents of the warning message. Using the bidirectional coupling to the road traffic simulator, it will also trigger for this vehicle an adjustment of the affected road segment's estimated travel time. In addition, if somewhere along its route the vehicle would have passed this road segment, a route recomputation is performed using estimated travel times as the cost metric of Dijkstra's shortest path algorithm. Vehicles participating in the VANET are thus given the opportunity to adjust their routes to avoid an incident. When the originating vehicle resumes its journey, it notifies other vehicles that the lane can be used again, allowing them to restore their original estimated travel times.

4.1 Analyzed IVC Protocols

We modeled two different protocol variants for evaluating VANETs in the *Veins* environment. These two variants represent extreme settings for possible IVC solutions. We briefly introduce both variants in the following.

4.1.1 Centralized TIS Scenario Relying on TCP and Standard MANET Protocols

Conventional TISs are organized in a centralistic way. Usually, sensor-based traffic monitoring systems are deployed directly at the roadside to collect information about current traffic conditions. Additionally, cars may participate as sensors in this scenario. The monitoring data is transferred to a central TIC, where the current road situation is analyzed. The result of this situation analysis is transmitted to all participating cars using a broadcast medium. Characteristic of a centralized TIS is that the traffic information is processed in one or more dedicated (centralized) traffic processing entities, e.g., a TIC.

For our experiments, we created such a centralized scenario as depicted in Fig. 6. In this scenario, a number of RSUs, each connected to a central traffic information service, are added to each intersection to support IVC. In this scenario, vehicles maintain a TCP connection to the central server, which is used to publish and revoke incident information. In intervals of 60 s or 180 s, depending on the scenario, vehicles also use the TCP connection to retrieve a list of incident warnings from the central server. The ad hoc routing is provided by a typical MANET routing protocol. In the following, we refer to this scenario as the "TCP scenario."

4.1.2 Distributed/Self-Organized TIS Using UDP Broadcast Communication

An alternative and completely different approach for monitoring the traffic situation and distributing the traffic messages to vehicle drivers is to employ a decentralized self-organizing TIS. An example is the decentralized SOTIS approach [34], in which vehicles inform each other of the local traffic situation by means of IVC. The traffic situation analysis is performed locally in each car. No communication/sensor infrastructure is required. For a global route optimization, the SOTIS information for the local area (e.g., for a radius of 50-100 km) can additionally be combined with traffic information provided by roadside access points or conventional centralized systems.

We realized such distributed behavior by flooding incident warnings through the VANET, as shown in Fig. 7. This notification is being flooded over 5 hops or 25 hops, depending on the scenario. Upon receiving an incident warning, a vehicle would query the originating node if the warning was current and, if it received a positive reply, try and avoid the lane in question. In the following, we refer to this distributed scenario as the "UDP scenario."

4.2 Simulation Setup

We configured vehicles to drive at a maximum speed of 14 m/s and modeled dense inner-city traffic with inattentive drivers. Table 1 lists the values used to parameterize the vehicles of the road traffic microsimulation.

For all communications, the complete network stack, including ARP, is simulated and wireless modules are configured to closely resemble IEEE 802.11b network cards transmitting at 11 Mbit/s with RTS/CTS disabled. For the

TABLE 1
Road Traffic Microsimulation Setup

Parameter	Value
Maximum vehicle speed	14 m/s
Maximum vehicle acceleration	$2.6\,\mathrm{m/s^2}$
Maximum desired deceleration	$4.5\,\mathrm{m/s^2}$
Assumed vehicle length	5 m
Driver imperfection σ ("dawdling")	0.5

TABLE 2
INET Framework Module Parameters

Parameter	Value
TCP.mss	1024 Byte
TCP.advertisedWindow	14 336 Byte
TCP.tcpAlgorithmClass	TCPReno
ARP.retryTimeout	1 s
ARP.retryCount	3
ARP.cacheTimeout	100 s
mac.address	auto
mac.bitrate	11 Mbit/s
mac.broadcastBackoff	31 slots
mac.maxQueueSize	14 Pckts
mac.rtsCts	false
decider.bitrate	11 Mbit/s
decider.snirThreshold	4 dB
snrEval.bitrate	11 Mbit/s
snrEval.headerLength	192 bit
snrEval.snrThresholdLevel	3 dB
snrEval.thermalNoise	−110 dB
snrEval.sensitivity	−85 dB
snrEval.pathLossAlpha	1.9
snrEval.carrierFrequency	2.4 GHz
snrEval.transmitterPower	2 mW
channelcontrol.carrierFrequency	2.4 GHz
channelcontrol.pMax	2 mW
channelcontrol.sat	−80 dBm
channelcontrol.alpha	1.9

simulation of radio wave propagation, a plain free-space model is employed, with the transmission ranges of all nodes adjusted to a fixed value of 180 m, a trade-off between varying real-world measurements described in related work [35], [36].

To provide ad hoc routing among the nodes, we use our implementation [37] of the Dynamic MANET On Demand (DYMO) routing protocol as an application layer module of the *INET Framework* module set. As per the specification, it uses a node's UDP module to communicate with other instances of DYMO, to discover and maintain routes, and thus, establish a VANET.

All simulation parameters used to parameterize the modules of the *INET Framework* are summarized in Table 2.

4.3 Manhattan Grid Scenario

The basic scenario that we use for the evaluation is illustrated in Fig. 8. Single-lane roads are laid out in the form of a Manhattan Grid with the intersections of roads spaced 1 km apart. This scenario is frequently used for evaluating protocols designed for urban scenarios—a very good and representative example is described in [28]. Simulations are performed for grid sizes ranging from 5×5 roads to 16×16 roads (using 30 and 1,000 vehicles, respectively). In each simulation, all vehicles start, one by one, at a fixed source node in the top left corner of the grid. If no IVC takes place, vehicles then travel along the shortest route to a fixed sink node located in the bottom right corner of the grid.

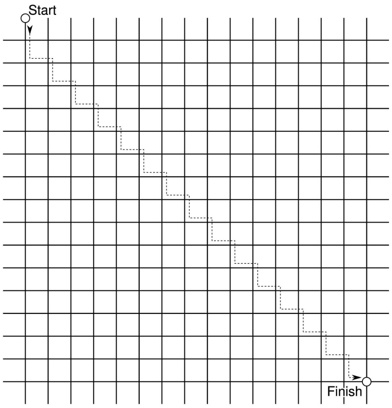


Fig. 8. Overview of the simulated Manhattan Grid scenario: single-lane roads are laid out in a grid with a cell size of 1 km. Start and finish node positions are fixed. With no IVC, cars always pick the shortest route.

Traffic obstructions are introduced by stopping the lead vehicle for 60 or 240 s, depending on the scenario. As each road offers a single lane per driving direction, nodes cannot overtake each other, and hence, need to find a way around blocked roads by means of IVC, or get stuck in traffic.

In order to examine the impact of different IVC setups on communication performance, we measured the number of packet collisions on the wireless channel per packet sent. This measure is often used in the context of analyzing the efficiency of MANET routing protocols as it describes the effective utilization of the wireless channel. Figs. 9a and 9b show the results of this evaluation for small-scale and large-scale simulations, respectively. As can be seen, collision ratios in the TCP scenarios always remained under a tolerable 10 percent packet loss, which could easily be compensated by TCP retry mechanisms. Collision ratios in UDP scenarios, however, exceeded 25 percent in large-scale simulations, which significantly hindered packet exchanges.

These results are also reflected in the impact different IVC setups had on road traffic performance. Plotted in Figs. 10a and 10b is the effective average speed of vehicles, measured in small and large-scale simulations, respectively. It was obtained by dividing the length of the shortest route by each vehicle's total travel time.

Two simulation scenarios were configured with no IVC taking place. As can be seen, in the case of free flowing traffic, the travel time distribution among simulated vehicles in both scenarios is almost homogeneous, as could be expected, but speeds average at well below the maximum speed of 14 m/s. This is due to cars decelerating at every intersection, which, in combination with high traffic densities on the single, shortest route shared by all vehicles, leads to micro jams. In the second case, where the lead vehicle stopped for a short time duration, e.g., due to an accident, without IVC taking place, the average node speed is reduced by both this stop and by the traffic jam left behind.

Using a traffic incident warning protocol, we expect the road traffic being influenced by the IVC protocol. As stated before, we examined the effects of two different protocols.

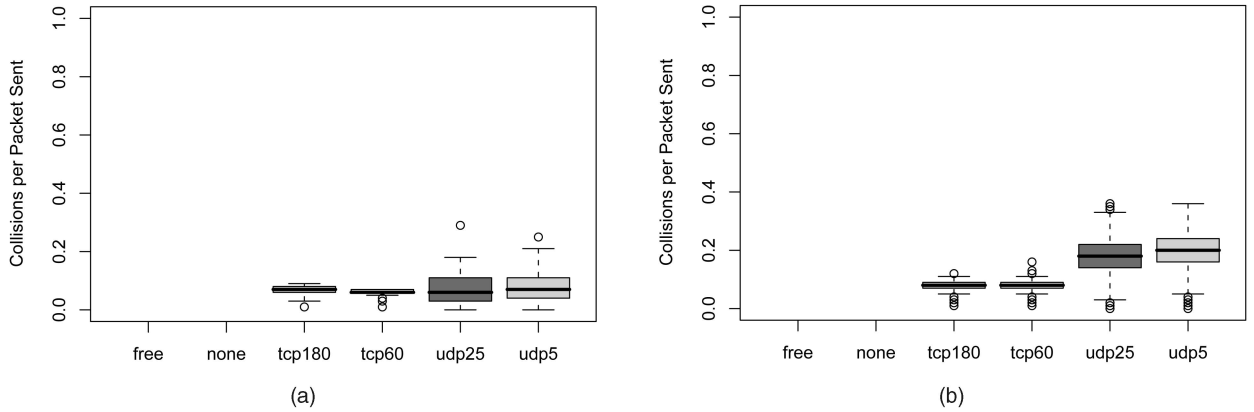


Fig. 9. Packet collisions on wireless channel per packet sent. One scenario with free flowing traffic, one with no IVC, four scenarios with VANET communications. (a) Thirty vehicles on a 5×5 grid; lead vehicle stops for 60 s. (b) One thousand vehicles on a 16×16 grid; lead vehicle stops for 240 s.

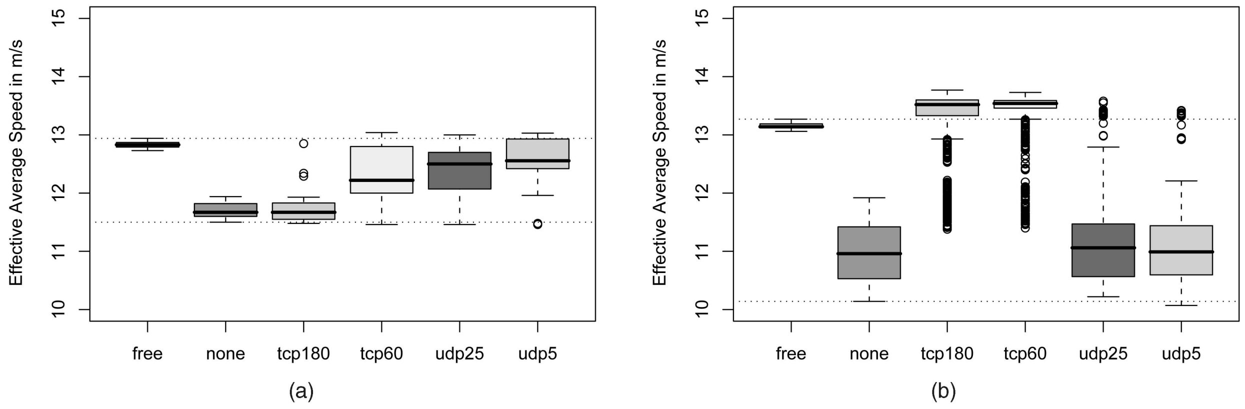


Fig. 10. Vehicle speed averaged over all vehicles and complete route. One scenario with free flowing traffic, one with no IVC, and four scenarios with VANET communications. (a) Thirty vehicles on a 5×5 grid; lead vehicle stops for 60 s. (b) One thousand vehicles on a 16×16 grid; lead vehicle stops for 240 s.

Depending on the scale of the simulation, different IVC scenarios performed differently at helping vehicles avoid the artificially generated incident.

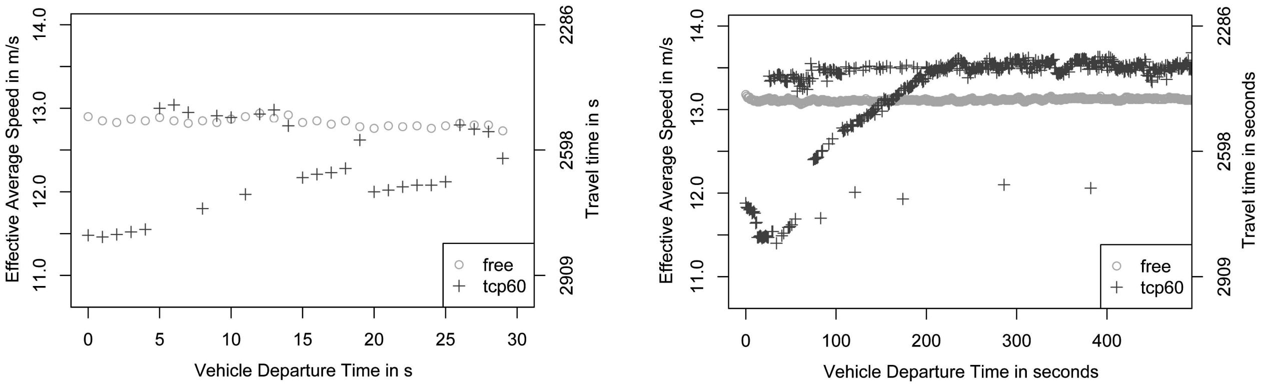
In the small-scale simulation of Fig. 10a, a polling interval of 180 s for TCP communications proved too long to significantly influence road traffic performance, but a polling interval of 60 s already led to a noticeable improvement. Performance was even better for UDP communication scenarios, where almost a quarter of vehicles did not suffer increased travel times due to the simulated incident if the TTL was reduced to 5 hops.

In the large-scale simulations of Fig. 10b, results were almost reversed. Here, UDP communications could only insignificantly improve road traffic performance and TCP communication scenarios fared far better. When a small polling interval was used, almost all vehicles reached their goal even faster than they could in the case of unobstructed traffic without IVC, thanks to a large number of vehicles taking alternate routes, which reduced traffic densities and helped avoid micro jams.

In order to provide a more detailed look into traffic effects in this scenario, Figs. 11a and 11b show the effective average speed of vehicles, but present measurements separated by vehicles' departure times. Plotted is one single example run each, for both the case of free flowing traffic and the case of IVC with an artificially triggered incident.

According to the scheduled incident, the lead vehicle is delayed by 60 s and 240 s, respectively. In the 5×5 scenario depicted in Fig. 11a, the cars following immediately behind are forced into a traffic jam and delayed accordingly. If the IVC message that is sent by the stopped car can be received by following cars, they can reroute to a free road and bypass the jam area. These cars can even drive faster than the cars in an incident-free scenario because they do not get delayed at street corners.

Similarly, the incident stopping the leading car involves all cars following immediately behind it in a traffic jam in the 16×16 scenario shown in Fig. 11b. This time some of them are delayed even further. The first cluster of cars that was more than one road away from the incident, however, already had enough time to receive and process the incident warning early enough to be able to find alternative routes to the destination, allowing vehicles to reach it even faster than they could when they just followed the shortest route in the incident-free scenario. As can be seen, IVC managed to prevent permanent delays on the affected road segment, so even vehicles that



(a)

(b)

Fig. 11. Average speed of individual vehicles, ordered by time of departure. One scenario with free flowing traffic and one scenario with incident and IVC. Vehicles poll the TIC every minute. (a) Thirty vehicles on a 5×5 grid; lead vehicle stops for 60 s. (b) One thousand vehicles on a 16×16 grid; lead vehicle stops for 240 s.

were unaware of the incident were able to continue on their route shortly after the lead vehicle continued its journey: Up to a departure time of just over 240 s, their time spent in the jam linearly decreased toward zero.

Vehicles starting later than approximately 65 s and 250 s, respectively, were completely unaffected by the incident having taken place, the only noticeable delays being caused because of merging traffic streams as vehicles approached their destination.

Measuring the runtime performance of simulations, we achieved similar results to those obtained in unidirectionally coupled simulations [18]. Depending on the exact parameterization, runs using bidirectionally coupled simulations took only insignificantly longer—or, in corner cases, ran even faster—compared to those using a random waypoint mobility model.

4.4 A More Realistic Example—Street Map of Erlangen

Building on the first proof-of-concept example in which vehicles traveled on an artificial grid of roads, we now used the coupled simulation environment to model IVC among traffic in the city of Erlangen, Germany. More specifically, we simulated 200 cars leaving the parking lot of the computer science building, on average one every 6 s, then heading to a business park along an individual, dynamically chosen route.

Serving as the basis for the road layout in this scenario was map data publicly available from the OpenStreetMap project. This project unites data from a multitude of free data sources like the US Census Bureau's TIGER geographic database, together with community-generated map data obtained by volunteers capturing GPS tracks using handheld devices, then postprocessing these tracks to obtain detailed maps. The collected data are available under the permissive "Creative Commons Attribution-Share Alike" license, allowing the direct modification, as well as free use of aggregated map data by interested parties.

A rendered representation of the map data, overlaid with the locations of traffic source and sink nodes, is given in Fig. 12. This data modeled the particular section of the required road network in great detail, accurately reflecting road attributes such as road type, access restrictions, lane counts, and speed limits. We successfully converted the raw map data to form a SUMO network, preserving the road layout, as well as all pertinent road attributes.

To facilitate comparability between results obtained in this scenario to results from the Manhattan Grid scenario, we first discuss the effective average speed. In a second step, as an example of evaluations beyond basic travel time comparisons, we will discuss results obtained from calculations of simulated vehicles' CO_2 emissions.

Just like in the previous example, four sets of simulation runs were performed. One set of runs simulated uninhibited road traffic. In this set of runs, again a fixed route was computed for each vehicle prior to starting the simulation, but this time a dynamic user assignment model [38] was employed for this computation. This could be shown to lead to an equilibrium in route distribution among

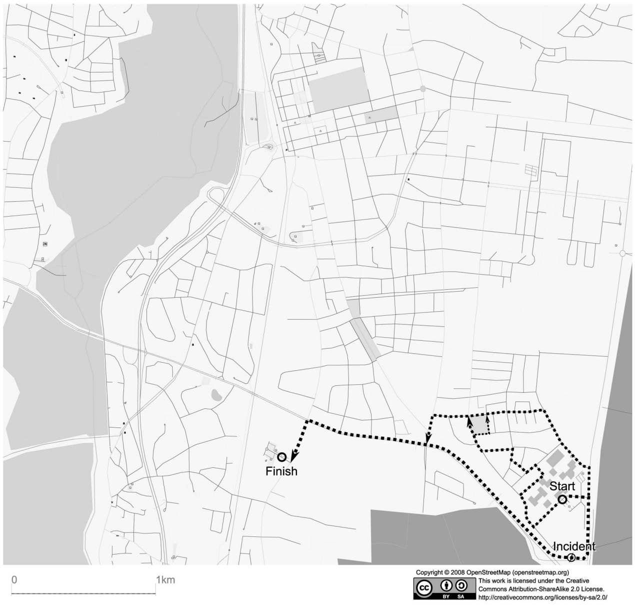

Finish
Start
Incident
0
1km
Copyright © 2008 OpenStreetMap (openstreetmap.org)
This work is licensed under the Creative Commons Attribution-ShareAlike 2.0 License.
http://creativecommons.org/licenses/by-sa/2.0/

Fig. 12. Map of Erlangen, Germany, as available from the OpenStreetMap project. Overlaid are the scenario's start and finish positions, the fastest route between them, and popular alternative routes around the artificial incident.

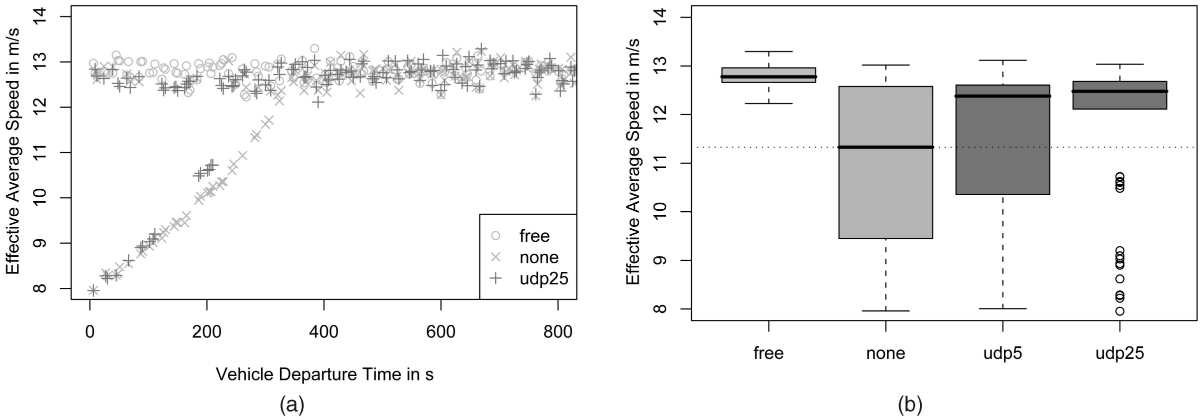


Fig. 13. Average speed of individual vehicles for free flowing traffic, traffic with an incident, and for UDP-based IVC. (a) Average speed ordered by time of departure. (b) Average speed grouped by scenario.

vehicles and roads and provide near-optimal travel times for every user. As in the previous scenario, this route assignment served as the basis for individual dynamic recomputation of routes, should a vehicle encounter a traffic incident.

In the second set of runs, such an incident was simulated by stopping the lead vehicle of cars traveling along the major artery connecting the university campus and the business park. In the final two sets of simulation runs, all vehicles were equipped with IVC technology, so stopped vehicles could disseminate information about congested road segments through a VANET. The two IVC-enabled sets of simulation runs differed in the range over which information dissemination was allowed—being configured to 5 hops and 25 hops, respectively. Vehicles that received such notifications could then often completely avoid traffic incidents.

Because IVC via a TCP-based protocol was not likely to provide additional insights, simulations in this example were only performed using the UDP-based IVC protocol introduced in Section 4.1.

Plotted in Fig. 13a are exemplary results of these three sets of simulation runs, showing the effective average speed of each vehicle in relation to the time it entered the simulation. As can be seen, the variance of travel times in the first scenario ("free") was much greater than in the previous example, due to the simulated vehicles now traveling to their destination along a multitude of different routes, as computed by the dynamic user assignment. Therefore, not all, yet still a major portion of the vehicles, were involved in the incident on the major artery that took place in the second scenario, where no IVC took place ("none"). Enabling IVC over 25 hops in the third scenario ("udp25") led to a significant increase of vehicles' speeds, as vehicles that were not too close to the incident when it happened, and thus were caught in the resulting jam, were now able to turn around before they reached the affected road segment, delaying them only slightly. Other cars managed to avoid the incident altogether.

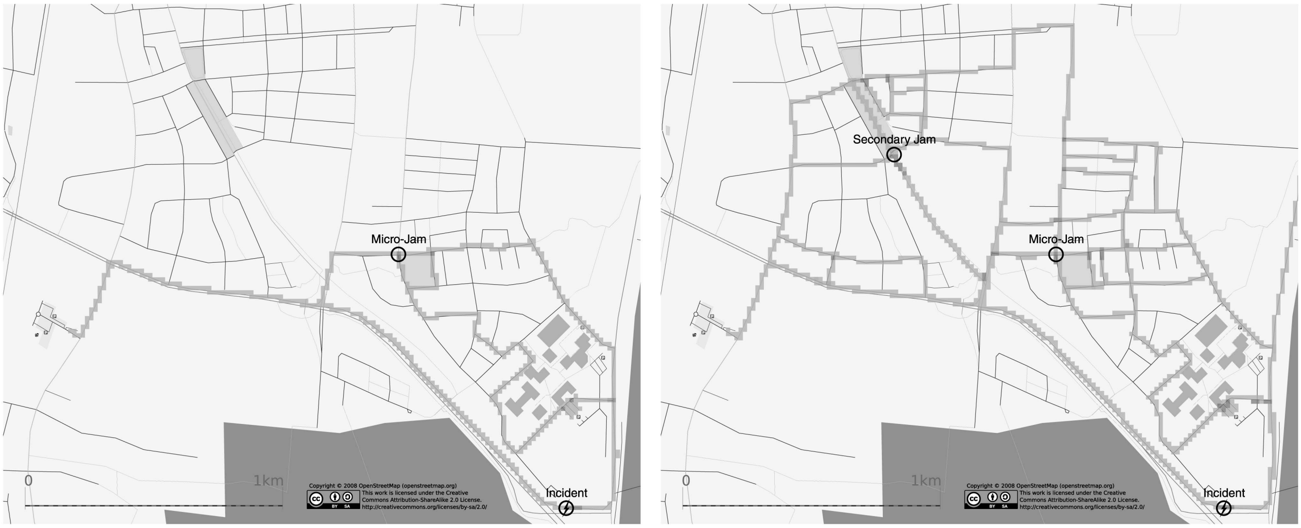
However, closer examination of the travel times reveals a small number of vehicles, even among those that started driving long after the original incident was resolved, which experienced noticeable delays compared to those in runs where no IVC took place. Those vehicles were found to have been caught in secondary microjams which occurred on detours, in particular, at intersections where popular alternative routes merge. This effect is a direct consequence of vehicles considering only the currently-known traffic state in rerouting decisions, and thus, often favoring detours using low-capacity roads. It is illustrated in Fig. 14a in the form of a 2D heatmap, where we indicate the maximum length of stop and mark (micro)jams. As can be seen, about three microjams occurred in our example.

In order to verify the observed effects, we added a second traffic flow that crosses the detours resulting from the first example. We carefully modeled this second traffic flow so as not to influence traffic on the original route between the shown start and finish points. In this extended scenario, the effect of microjams was even more pronounced, as illustrated in Fig. 14b. The main reason is that the low-capacity roads taken for the detours were already used to almost their maximum. This way, a small fraction of additional traffic led to further road congestion, and therefore, to additional microjams and even a much longer, secondary jam, as illustrated in Fig. 14b.

The increased variance and improvements of the average speed are summarized in the boxplot in Fig. 13b. In this plot, only individual vehicles starting earlier than 400 s are considered to outline the characteristics of free flowing traffic, traffic queuing after an incident, and the advantages of UDP-based IVC over 5 hops and 25 hops, respectively. The latter scenarios again outline the need for bidirectional coupling of the simulation tools.

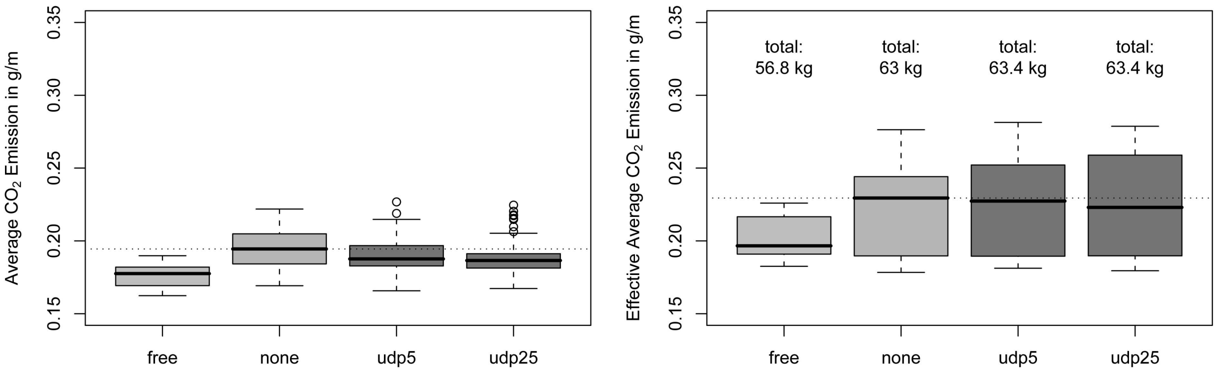
These results clearly demonstrate the negative impact the artificial traffic incident had on travel times, as well as how an IVC can alleviate this impact. This reduction in travel times is due to vehicles taking alternative, noncongested, but longer routes. This might raise the question of whether IVC might, while being beneficial for travel times and safety, on the other hand harm the environment. We will use this question as an example of how to take the presented evaluation one step further, now examining the CO_2 emissions of vehicles participating in an IVC system.

In order to estimate CO_2 emissions, we have integrated the EMIT model [39] of vehicle emissions with our OMNeT++ mobility model. EMIT calculates emissions depending on vehicle speed and acceleration, taking into


Micro-Jam
Incident
0
1km
Copyright © 2008 OpenStreetMap (openstreetmap.org) This work is licensed under the Creative Commons Attribution-ShareAlike 2.0 License. http://creativecommons.org/licenses/by-sa/2.0/
Secondary Jam
Micro-Jam
Incident
0
1km
Copyright © 2008 OpenStreetMap (openstreetmap.org) This work is licensed under the Creative Commons Attribution-ShareAlike 2.0 License. http://creativecommons.org/licenses/by-sa/2.0/

(a) (b)

Fig. 14. Heatmap of vehicle stop times in $25 \text{ m} \times 25 \text{ m}$ regions of the scenario. (a) Baseline scenario. (b) Scenario with additional traffic flow crossing detours.



(a) (b)

Fig. 15. Average $\mathrm{CO_2}$ emission of vehicles for free flowing traffic, traffic with an incident, and for UDP-based IVC. (a) $\mathrm{CO_2}$ emission based on actual distance covered. (b) $\mathrm{CO_2}$ emission normalized to distance to-be-covered.

account vehicle characteristics such as total mass, engine, and installed catalytic converter. Tailpipe emission of CO_2 is calculated according to the following simple polynomial:

$$EMIT_{CO_2} = \begin{cases} \alpha + \beta v + \delta v^3 + \zeta a v, & \text{if } P_{tract} > 0, \\ \alpha', & \text{else.} \end{cases}$$

Values of α to ζ and the calculation of P_{tract}, the tractive power requirement at a vehicle's wheels, were fitted to match what authors termed a *category 9 vehicle,*[2] e.g., a 1994 Dodge Spirit.

The values obtained from CO_2 emission calculations for all simulated scenarios are plotted in Fig. 15. For better comparability, the first figure, Fig. 15a, shows vehicles' CO_2 emissions per distance traveled. As can be seen, simulations that included an artificial traffic incident, but no IVC, recorded a much larger amount of CO_2 emitted per meter traveled, reflecting the large number of cars caught in the jam and emitting CO_2, but not traveling. In scenarios that included IVC among vehicles, this figure improved vastly, which reflects these simulations' vehicles no longer burning fuel while standing still. However, this evaluation does not take into account the fact that, in order to avoid a jam, vehicles have to travel along detours.

This fact was accounted for when plotting Fig. 15b, which again shows the amount of CO_2 emitted by vehicles, but normalized to the length of the shortest route between start and destination, instead of the actual route taken.

This measure can be used to finally gauge the environmental impact of IVC, which can be seen to be almost nonexistent. Even in the examined scenario, which only allowed for very long detours around the artificial incident, IVC did not lead to a significant increase of CO_2 emissions: in all cases, about 63 kg CO_2 were produced in total by all the cars. From this experiment, we can learn how much environmental impact needs to be considered for a given improvement in terms of traveling time and average speed.

5 Conclusion

In conclusion, it can be said that bidirectionally coupled road traffic and network simulation provides major

[2] http://hdl.handle.net/1721.1/1675.

advantages compared to uncoupled or purely trace-driven simulation. Thus, for all protocol evaluations that cannot afford to disregard network or radio effects and for all evaluations where IVC would likely influence driver behavior, we strongly advocate the use of bidirectionally coupled simulation.

The following findings support this observation:

- simple traffic models are inappropriate for road traffic simulation [5],
- road traffic simulation and bidirectional coupling have a negligible impact on simulation runtime,
- traces of road traffic, either generated using road traffic microsimulation or by observing real-world traffic, allow realistic traffic modeling, but IVC protocols cannot be tested completely as no feedback can be provided to the mobility model [16], [18],
- using bidirectional coupling, the impact of IVC on road traffic can be directly evaluated [11], [19].

The simulation framework *Veins*, which we developed, provides all necessary functionality to perform this bidirectional coupling. Using our methodology, we provide means to evaluate developed protocols more accurately. The simulation framework relies on state-of-the-art simulators from both domains, thus, it incorporates well-known models for road traffic microsimulation with a comprehensive selection of models of network protocols. The presented proof-of-concept study demonstrated not only the applicability but also the need for bidirectional coupling of road traffic microsimulation and network simulation. Furthermore, the integrated model for CO_2 emission allows not only to investigate the effects IVC will have on the road traffic, e.g., in terms of average speed and covered distance, but also on the environment, i.e., to decide whether the advantages for drivers will have a higher impact on the environment.

ACKNOWLEDGMENTS

This work was partially supported by the German CoCar project funded by the BMBF under grant number 01BU0692. This paper is based on earlier work on Veins that was presented at the workshops of ACM MobiHoc 2008 (bidirectional coupling) and IEEE INFOCOM 2008 (IVC analysis).

REFERENCES

[1] J.J. Blum, A. Eskandarian, and L.J. Hoffman, “Challenges of Intervehicle Ad Hoc Networks,” *IEEE Trans. Intelligent Transportation Systems,* vol. 5, no. 4, pp. 347-351, Dec. 2004.

[2] H. Hartenstein and K.P. Laberteaux, “A Tutorial Survey on Vehicular Ad Hoc Networks,” *IEEE Comm. Magazine,* vol. 46, no. 6, pp. 164-171, June 2008.

[3] M.L. Sichitiu and M. Kihl, “Inter-Vehicle Communication Systems: A Survey,” *IEEE Comm. Surveys and Tutorials,* vol. 10, no. 2, pp. 88-105, 2008.

[4] T. Camp, J. Boleng, and V. Davies, “A Survey of Mobility Models for Ad Hoc Network Research,” *Wireless Comm. and Mobile Computing,* special issue on mobile ad hoc networking: research, trends and applications, vol. 2, no. 5, pp. 483-502, 2002.

[5] J. Yoon, M. Liu, and B. Noble, “Random Waypoint Considered Harmful,” *Proc. IEEE INFOCOM,* vol. 2, pp. 1312-1321, Mar. 2003.

[6] A. Mahajan, N. Potnis, K. Gopalan, and A. Wang, “Urban Mobility Models for VANETs,” *Proc. Second IEEE Int'l Workshop Next Generation Wireless Networks (IEEE WoNGeN '06),* Dec. 2006.

[7] A.K. Saha and D.B. Johnson, “Modeling Mobility for Vehicular Ad-Hoc Networks,” *Proc. First ACM Int'l Workshop Vehicular Ad Hoc Networks (VANET '04),* pp. 91-92, Oct. 2004.

[8] C. Sommer, Z. Yao, R. German, and F. Dressler, “On the Need for Bidirectional Coupling of Road Traffic Microsimulation and Network Simulation,” *Proc. ACM MobiHoc: First Int'l Workshop Mobility Models for Networking Research (MobilityModels '08),* pp. 41-48, May 2008.

[9] A. Varga, “The OMNeT++ Discrete Event Simulation System,” *Proc. European Simulation Multiconf. (ESM '01),* June 2001.

[10] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, “SUMO (Simulation of Urban MObility); An Open-Source Traffic Simulation,” *Proc. Fourth Middle East Symp. Simulation and Modelling (MESM '02),* pp. 183-187, Sept. 2002.

[11] C. Sommer, Z. Yao, R. German, and F. Dressler, “Simulating the Influence of IVC on Road Traffic Using Bidirectionally Coupled Simulators,” *Proc. IEEE INFOCOM: Mobile Networking for Vehicular Environments (MOVE '08),* Apr. 2008.

[12] C. Sommer and F. Dressler, “Progressing Towards Realistic Mobility Models in VANET Simulations,” *IEEE Comm. Magazine,* vol. 46, no. 11, pp. 132-137, Nov. 2008.

[13] D. Li, H. Huang, X. Li, M. Li, and F. Tang, “A Distance-Based Directional Broadcast Protocol for Urban Vehicular Ad Hoc Network,” *Proc. Int'l Conf. Wireless Comm. Networking and Mobile Computing (WiCom '07),* pp. 1520-1523, Sept. 2007.

[14] H.-Y. Huang, P.-E. Luo, M. Li, D. Li, X. Li, W. Shu, and M.-Y. Wu, “Performance Evaluation of SUVnet With Real-Time Traffic Data,” *IEEE Trans. Vehicular Technology,* vol. 56, no. 6, pp. 3381-3396, Nov. 2007.

[15] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, “Routing in Sparse Vehicular Ad Hoc Wireless Networks,” *IEEE J. Selected Areas in Comm.,* vol. 25, no. 8, pp. 1538-1556, Oct. 2007.

[16] C. Lochert, A. Barthels, A. Cervantes, M. Mauve, and M. Caliskan, “Multiple Simulator Interlinking Environment for IVC,” *Proc. Second ACM Int'l Workshop Vehicular Ad hoc Networks (VANET '05), Poster Session,* pp. 87-88, Sept. 2005.

[17] M. Killat, F. Schmidt-Eisenlohr, H. Hartenstein, C. Rössel, P. Vortisch, S. Assenmacher, and F. Busch, “Enabling Efficient Accurate Large-Scale Simulations of VANETs for Vehicular Traffic Management,” *Proc. ACM MobiCom: Fourth Int'l Workshop Vehicular Ad Hoc Networks (VANET '07),* pp. 29-38, Sept. 2007.

[18] C. Sommer and F. Dressler, “The DYMO Routing Protocol in VANET Scenarios,” *Proc. 66th IEEE Vehicular Technology Conf. (VTC '07-Fall),* pp. 16-20, Sept./Oct. 2007.

[19] S.Y. Wang, C.L. Chou, Y.H. Chiu, Y.S. Tseng, M.S. Hsu, Y.W. Cheng, W.L. Liu, and T.W. Ho, “NCTUns 4.0: An Integrated Simulation Platform for Vehicular Traffic Communication, Network Researches,” *Proc. First IEEE Int'l Symp. Wireless Vehicular Comm. (WiVec '07),* Oct. 2007.

[20] M. Piorkowski, M. Raya, A.L. Lugo, M. Grossglauser, and J.-P. Hubaux, “Joint Traffic and Network Simulator for VANETs,” *Proc. Mobile Information and Comm. Systems (MICS '06), Poster Session,* Oct. 2006.

[21] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K.-c. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan, “Effects of Detail in Wireless Network Simulation,” *Proc. SCS MultiConf. Distributed Simulation,* pp. 3-11, Jan. 2001.

[22] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, “Advances in Network Simulation,” *Computer,* vol. 33, no. 5, pp. 59-67, May 2000.

[23] A. Sobeih, J.C. Hou, L.-C. Kung, N. Li, H. Zhang, W.-P. Chen, H.-Y. Tyan, and H. Lim, “J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks,” *IEEE Wireless Comm.,* vol. 13, no. 4, pp. 104-119, Aug. 2006.

[24] R. Barr, Z.J. Haas, and R. van Renesse, “JiST: Embedding Simulation Time into a Virtual Machine,” *Proc. EuroSim Congress on Modelling and Simulation,* 2004.

[25] N. Elloumi, H. Haj-Salem, and M. Papageorgiou, “METACOR: A Macroscopic Modeling Tool for Urban Corridors,” *Proc. Triennal Symp. Transportation Analysis (TRISTAN II),* vol. 1, pp. 135-150, June 1994.

[26] N.B. Taylor, “The CONTRAM Dynamic Traffic Assignment Model,” *Networks and Spatial Economics,* vol. 3, pp. 297-322, 2003.

[27] K. Nagel and M. Schreckenberg, "A Cellular Automaton Model for Freeway Traffic," *J. Physique I France,* vol. 2, pp. 2221-2229, 1992.

[28] O.K. Tonguz, W. Viriyasitavat, and F. Bai, "Modeling Urban Traffic: A Cellular Automata Approach," *IEEE Comm. Magazine,* vol. 47, no. 5, pp. 142-150, May 2009.

[29] S. Krauß, "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics," PhD thesis, Univ. of Cologne, 1998.

[30] M. Treiber, A. Hennecke, and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical Rev. E,* vol. 62, p. 1805, 2000.

[31] M. Treiber and D. Helbing, "Realistische Mikrosimulation Von Straßenverkehr Mit Einem Einfachen Modell," *Proc. 16th Symp. Simulationstechnik (ASIM '02),* Sept. 2002.

[32] E. Brockfeld and P. Wagner, "Testing and Benchmarking of Microscopic Traffic Flow Simulation Models," *Proc. 10th World Conf. Transport Research (WCTR),* July 2004.

[33] A. Wegener, M. Piorkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An Interface for Coupling Road Traffic and Network Simulators," *Proc. 11th Comm. and Networking Simulation Symp. (CNS '08),* Apr. 2008.

[34] L. Wischhof, A. Ebner, and H. Rohling, "Information Dissemination in Self-Organizing Intervehicle Networks," *IEEE Trans. Intelligent Transportation Systems,* vol. 6, no. 1, pp. 90-101, Mar. 2005.

[35] F. Hui and P. Mohapatra, "Experimental Characterization of Multi-Hop Communications in Vehicular Ad Hoc Network," *Proc. Second ACM Int'l Workshop Vehicular Ad Hoc Networks (VANET '05),* Sept. 2005.

[36] H. Wu, M. Palekar, R. Fujimoto, J. Lee, J. Ko, R. Guensler, and M. Hunter, "Vehicular Networks in Urban Transportation Systems," *Proc. Nat'l Conf. Digital Gov't Research (DGO '05),* May 2005.

[37] C. Sommer, I. Dietrich, and F. Dressler, "A Simulation Model of DYMO for Ad Hoc Routing in OMNeT++," *Proc. First ACM/ICST Int'l Conf. Simulation Tools and Techniques for Comm. Networks and Systems (SIMUTools '08): First ACM/ICST Int'l Workshop OMNeT++ (OMNeT++ '08),* Mar. 2008.

[38] C. Gawron, "Simulation-Based Traffic Assignment—Computing User Equilibria in Large Street Networks," PhD thesis, Univ. of Cologne, 1999.

[39] A. Cappiello, I. Chabini, E. Nam, A. Lue, and M. Abou Zeid, "A Statistical Model of Vehicle Emissions and Fuel Consumption," *Proc. Fifth IEEE Int'l Conf. Intelligent Transportation Systems (ITSC),* pp. 801-809, 2002.

Christoph Sommer received the MSc degree in computer science from the University of Erlangen, Germany, in 2006. He is currently a PhD student working with the Electrical and Computer Engineering Department of Carnegie Mellon University and the Computer Networks and Communication Systems Group at the Department of Computer Science, University of Erlangen. His research is focused on questions regarding efficiency and security aspects of Car-to-X communication in heterogeneous environments. He is a student member of the IEEE.

Reinhard German received a diploma in computer science in 1991, the PhD degree in 1994, and the habilitation degree in 2000 from the Computer Science Department, Technical University of Berlin. Thereafter, he joined the Department of Computer Science at the University Erlangen-Nuremberg. First, he was an associate professor (system simulation), then he became a full professor in 2004 (computer networks and communication systems), and now he is the head of the department. His research interests include model-based and measurement-based performance analysis, modeling and simulation paradigms and tools, numerical analysis of Markovian and non-Markovian models, vehicular communications, and autonomous sensor/actuator networks.

Falko Dressler received the MSc and PhD degrees from the Department of Computer Sciences, University of Erlangen, in 1998 and 2003, respectively. In 2003, he joined the Computer Networks and Internet Group at the University of Tuebingen. Since 2004, he has been with the Computer Networks and Communication Systems Group at the University of Erlangen, where he is currently a professor. He teaches classes on self-organizing sensor and actor networks, network security, and communication systems. He is an editor for journals such as Elsevier's *Ad Hoc Networks* and ACM/Springer's *Wireless Networks (WINET)*. He was guest editor of special issues on self-organization, autonomic networking, and bioinspired computing and communication for the *IEEE Journal on Selected Areas in Communications (JSAC)*, Elsevier's *Ad Hoc Networks*, and Springer's *Transactions on Computational Systems Biology (TCSB)*. Besides chairing a number of workshops associated to high-level conferences, he regularly acts on the program committees of leading networking conferences such as IEEE INFOCOM, IEEE ICC, IEEE Globecom, IEEE MASS, and others. He has published two books, including *Self-Organization in Sensor and Actor Networks* (Wiley, 2007). He is a senior member of the IEEE (Communications Society, Computer Society, and Vehicular Technology Society) as well as a senior member of the ACM (SIGMOBILE), and a member of the GI (KuVS and Real-time). His research activities are focused on self-organizing networks addressing issues in wireless ad hoc and sensor networks, intervehicular communication systems, bioinspired networking, and adaptive network security techniques.