On the Need for Bidirectional Coupling of Road Traffic Microsimulation and Network Simulation

Christoph Sommer, Zheng Yao, Reinhard German and Falko Dressler
Computer Networks and Communication Systems
Department of Computer Science, University of Erlangen, Germany
{christoph.sommer,german,dressler}@informatik.uni-erlangen.de

ABSTRACT

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 such protocols. The implemented mobility model, which defines all movement of nodes, influences the outcome of simulations to a great deal. Therefore, the use of a representative mobility model is essential for producing meaningful 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. Based on a proof-of-concept study, we demonstrate the advantages and the need for bidirectionally coupled simulation.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling Techniques; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*; I.6.m [**Simulation and Modeling**]: Miscellaneous

General Terms

Measurement, Performance

Keywords

Vehicular ad hoc networks, network simulation, road traffic microsimulation

1. INTRODUCTION

In this paper, we investigate the need for bidirectional coupling of network and road traffic simulation for more realistic Vehicular Ad Hoc Network (VANET) simulation experiments. The development of adequate Inter Vehicle Communication (IVC) protocols using VANETs is in the main focus of such simulations [2], e.g. for incident detection such as traffic jam and accident detection. In the following, we motivate the demand for more sophisticated simulation techniques by investigating the state of the art in network and road traffic simulation.

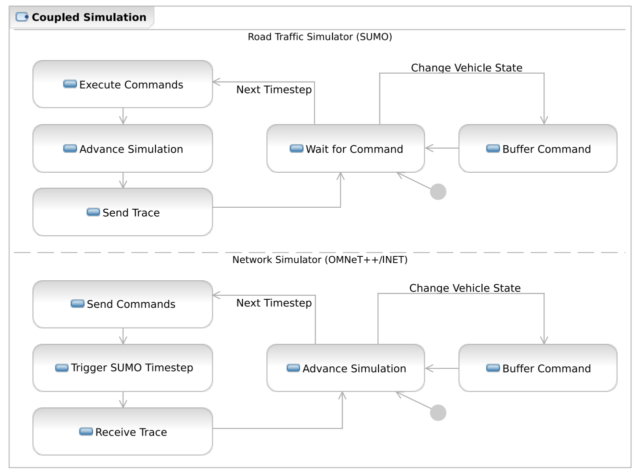


Figure 1: Overview of the coupled simulation framework. State machines of road traffic and network simulator communication modules.

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++ 3.4b2 [23], a simulation environment free for academic use, for modeling 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*[1] (Vehicles in Network Simulation), is given in Figure 1. Furthermore, we study the applicability of bidirectionally coupled network and road traffic simulation using a sample scenario evaluating the influence of IVC on road traffic.

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

MobilityModels'08, May 26, 2008, Hong Kong SAR, China.

The contributions of this paper can be summarized as follows. 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. Based on the SUMO road traffic microsimulation tool and the OMNeT++ network simulation framework, we developed the integrated VANET simulator *Veins* that allows dynamic interaction between both simulators (Section 5). As a proof of concept for incident warnings, we used the coupled simulation framework to evaluate two mechanisms for incident warnings and traffic jam prevention (Section 6).

2. 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 [6].

In most cases, network protocols are analyzed using discrete event simulation and a large number of simulation frameworks is available in this domain. Examples of such frameworks are open source tools such as the network simulator ns-2 [3], OMNeT++ [23], J-SIM [16], and JiST / SWANS [1] and commercial tools like OPNET. The working principles of all these simulators are similar and the differences lie mostly in the number of available models, i.e. typical MAC and routing protocols. Also, the support for large node numbers varies.

In our evaluation, we are using the network simulator OMNeT++, together with its *INET Framework* extension, for simulating VANET 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 similar techniques are used by the 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 usually limited to simple mobility patterns. Examples that are available in almost all network simulation frameworks are the Random Waypoint or Manhattan 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.

3. TRAFFIC MICROSIMULATION

Strictly speaking, for the most realistic simulation of moving nodes, their mobility would need to be deduced from 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.

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 [5], model traffic at a large scale, treating traffic like a liquid and often applying hydrodynamic flow theory to vehicle behavior. Mesoscopic models, like CONTRAM [20], 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 wave 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 complexity. Models that are in widespread use within the traffic science community include the Cellular Automaton (CA) model [14], the SK car-following model developed by Stefan Krauß [11], as well as the IDM/MOBIL model [21, 22]. When doing traffic simulation, each approach has its particular advantages and particular drawbacks. However, the accuracy of many of these models was evaluated in [4], which concluded with the recommendation to "take the simplest model for a particular application, because complex models likely will not produce better results". Essentially this means that, as far as network simulation is concerned, all common microsimulation approaches are of equal value as a mobility model.

Today, several simulation environments exist which can generate trace files of vehicles moving according to these microsimulation models. Common tools include Daimler-Chrysler's FARSI or VISSIM by PTV AG. In the interest of comparability of research results, however, it is evidently more beneficial to use readily available simulation environments, as using the same mobility model is the easiest and sometimes the only way of accurately reproducing results obtained in related work.

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 intra-junction 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

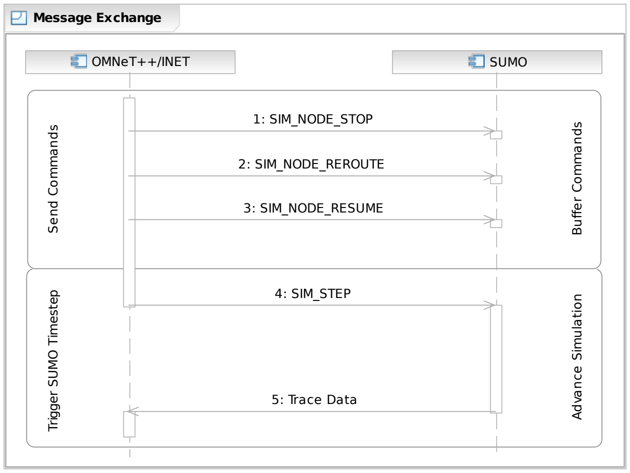


Figure 2: Messages exchanged between road traffic and network simulator communication modules.

```
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
```

Figure 3: Excerpt of the movement trace, as sent by the road traffic simulator.

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.

4. 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 simulation of network protocols may be unrealistic.

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 (off-line) and used as representative models for the evaluation of network protocols. In fact, as a common practice in many simulation platforms, the mobility traces are normally inserted into network simulation modules as independently-generated off-line files. This way, the system complexity is reduced. Two methods for the generation of 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 [7,12]. Similarly, mobility patterns can be extracted from these real-world observations to analytically model traffic flows [25].

Another approach is to employ traffic microsimulation tools coupled with network simulators. An early example is based on the integration of VISSIM traces with the network simulator ns-2 [13], a frequently used simulation framework. Similarly, the SUMO traffic microsimulation tool has been integrated with ns-2, resulting in the hybrid simulation framework TrANS [15].

Hybrid simulation and mathematical modeling have recently been combined in order to speed up the simulation process [9]. Also, our preliminary work facilitated coupled simulation using a road traffic microsimulation model based on the IDM/MOBIL model together with the OMNeT++ network simulation tool [18].

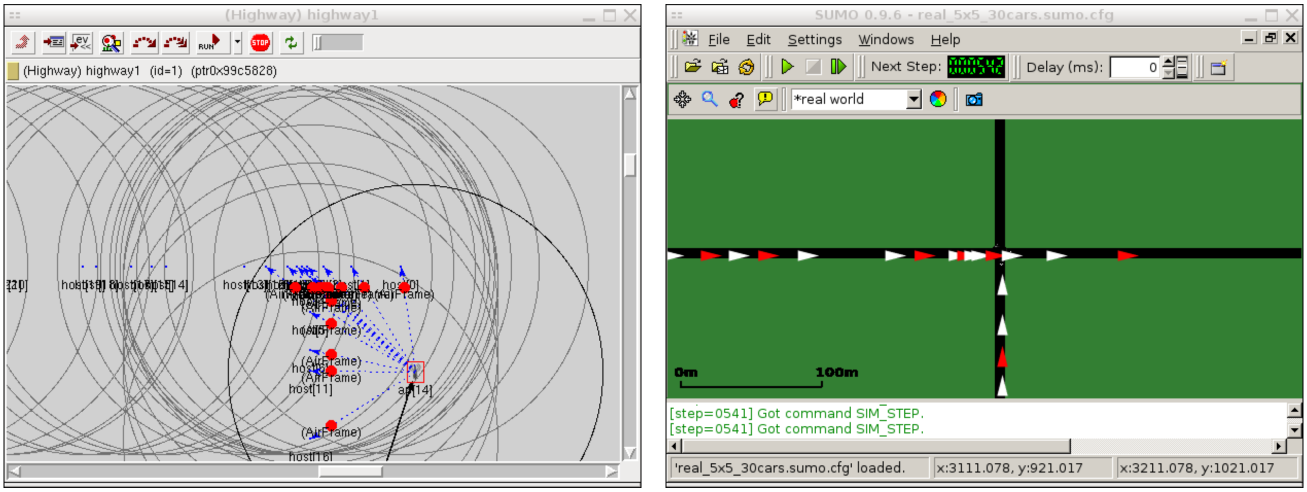
Nevertheless, such "de-coupling" design philosophy faces one dilemma: If the results from the network simulation module can affect the mobility trace, this off-line "isolated" methodology is unable to generate the real-time interaction between the mobility model simulation module and the network simulation 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.

This problem has been addressed with the NCTUns simulation environment [24]. This tool is similar to TrANS but 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 to 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.

5. BIDIRECTIONAL COUPLING

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 exchanged 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 Figure 1, 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. 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 environment conditions.



(a) OMNeT++/INET (b) SUMO

Figure 4: Screenshots of simulators' graphical user interfaces running network and road traffic simulations in parallel.

Figure 2 shows the commands sent by OMNeT++ to the road traffic simulator, allowing it to influence vehicles' behavior. Using these commands, vehicles in SUMO can be stopped, resumed, and rerouted to avoid arbitrary road segments. Also illustrated in Figure 2 are the alternating two phases of coupled simulation which result from this approach. In the first phase, commands are sent to SUMO, in the second phase their execution is triggered and the resulting mobility trace received.

Figure 3 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.

Figure 4 shows screenshots of the GUI versions of both simulators running a coupled simulation of IVC in traffic streams merging at an intersection.

6. PROOF-OF-CONCEPT EXAMPLES

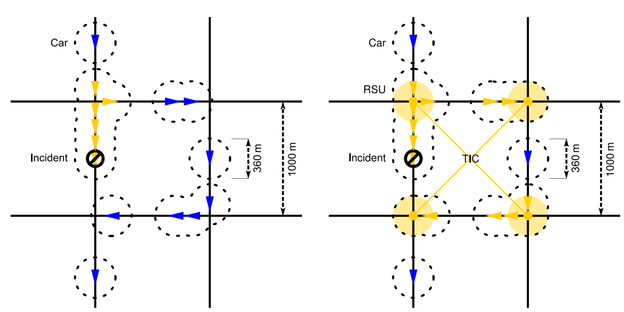
In the following, we demonstrate the advantages of bidirectionally coupled network and road traffic simulation using *Veins* in a proof-of-concept example. In particular, we use the scenarios depicted in Figure 5 for evaluating the influence of IVC protocols on road traffic using bidirectionally coupled simulators. Detailed information on the used protocol and an in-depth performance evaluation can be found in [19].

6.1 First Scenario and Setup

The scenario used in our simulations consists of a number of simulated single-lane roads. The roads are laid out in a grid with a cell size of 1 km. Simulations are performed for grid sizes ranging from 5×5 roads to 16×16 roads. 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.

Traffic obstructions are introduced by stopping the lead vehicle for 60 s 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.

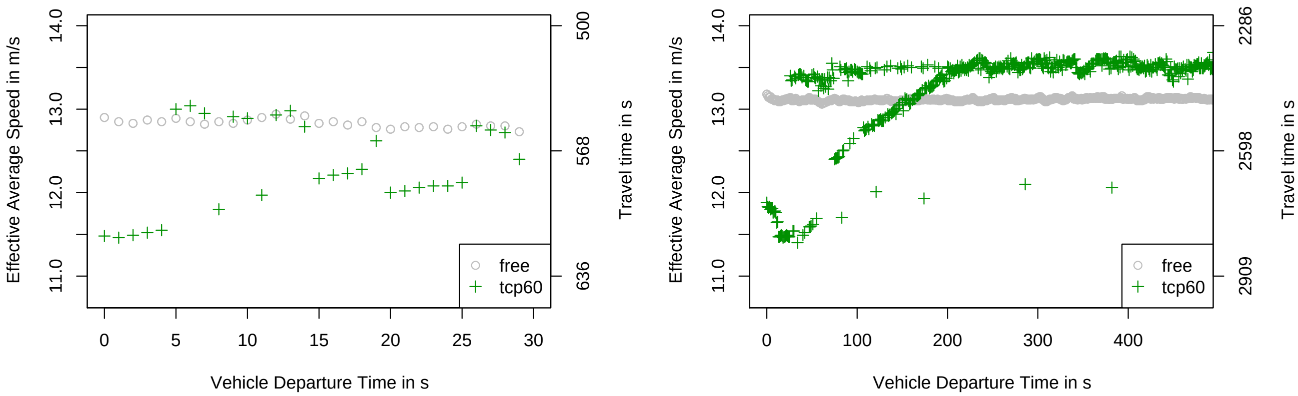
To provide ad hoc routing among the nodes, we use our implementation [17] of the Dynamic MANET On Demand (DYMO) routing protocol as an application-layer module of the *INET Framework* module set. As per the specification, it uses a node's UDP module to communicate with other instances of DYMO, to discover and maintain routes and thus establish a VANET.



(a) UDP IVC scenario. Communication relies on VANET alone.

(b) TCP IVC scenario. Communication supported by RSUs.

Figure 5: The two types of examined IVC scenarios.



(a) 30 vehicles on a 5×5 grid; lead vehicle stops for 60 s (b) 1000 vehicles on a 16×16 grid; lead vehicle stops for 240 s

Figure 6: Average speed of individual vehicles, ordered by time of departure. One scenario with free flowing traffic, one scenario with incident and IVC. Vehicles poll the Traffic Information Center every minute.

Two different types of IVC, illustrated in Figure 5, are examined. In both scenarios vehicles with a speed of zero, after some time, start to inform other vehicles of a potential incident on the current lane, causing them to avoid this lane. When the originating vehicle resumes its journey, it notifies other vehicles that the lane can be used again.

Figure 5(a) displays the UDP scenario, in which this notification was realized by flooding incident warnings through the VANET 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.

Figure 5(b) displays the TCP scenario, in which a number of Roadside Units (RSUs), each connected to a central traffic information service, were added to each intersection to support IVC. In this scenario, vehicles maintained a TCP connection to the central server, which was used to publish and revoke incident information. In intervals of 60 s or 180 s, depending on the scenario, vehicles also used the TCP connection to retrieve a list of incident warnings from the central server.

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

Table 1: Road Traffic Microsimulation Setup

Parameter	Value
Maximum vehicle speed	14 m/s
Maximum vehicle acceleration	2.6 m/s^2
Maximum desired deceleration	4.5 m/s^2
Assumed vehicle length	5 m
Driver imperfection σ (“dawdling”)	0.5

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 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 [8, 26]. All simulation parameters used to parameterize the modules of the *INET Framework* are summarized in Table 2.

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

6.2 First Simulation Results

In order to evaluate the performance of the IVC protocols, we measured the average speed of the vehicles within our scenario.

Two simulation scenarios were configured with no IVC taking place. In the case of free flowing traffic, i.e. simulations without any incidents, the speed distribution among simulated vehicles is almost homogeneous. Vehicles' average speed is 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 the lead vehicle stopped for a short amount of time, e.g. due to an accident. Here, 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

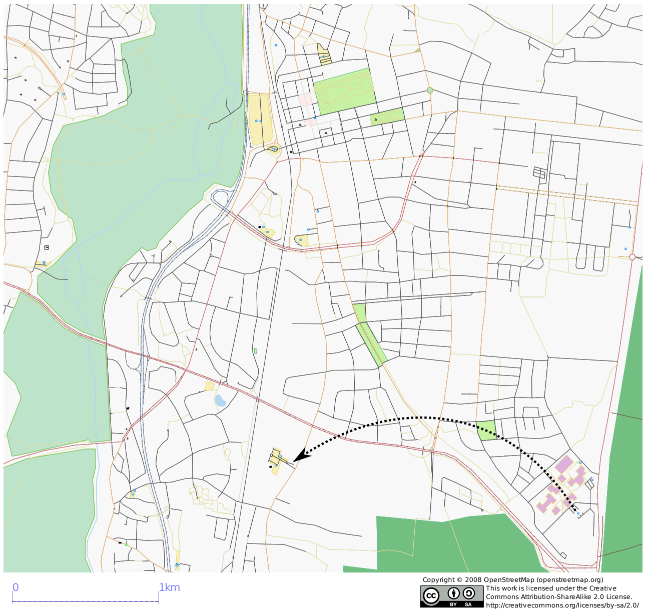


Figure 7: Map of Erlangen, Germany as available from the OpenStreetMap project

before, we examined the effects of two different protocols. Depending on the scale of the simulation, different IVC scenarios performed differently at helping vehicles avoid the artificially-generated incident.

In order to provide a more detailed look into traffic effects in this scenario, Figures 6(a) and 6(b) 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 Figure 6(a), 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 re-route 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 Figure 6(b). This time some of them are delayed even further. The first cluster of cars that were 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 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 towards zero.

Measuring the run-time 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 – than those using a random waypoint mobility model.

6.3 A More Realistic Example

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.

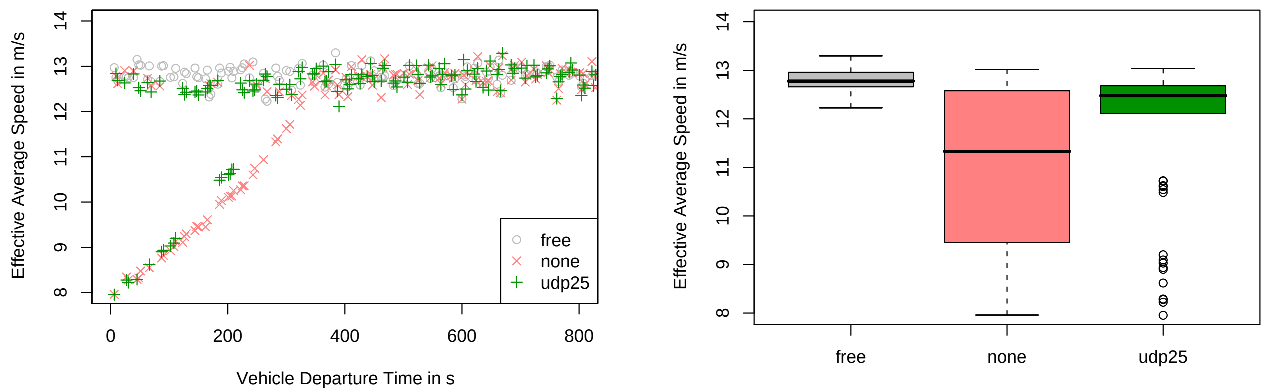
Serving as the basis for the road layout in this scenario was map data publicly available from the OpenStreetMap project. This project unites data from a multitude of free data sources like the U.S. Census Bureau's TIGER geographic database, together with community-generated map data obtained by volunteers capturing GPS tracks using handheld devices, then post-processing these tracks to obtain detailed maps. The collected data is 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 Figure 7. This data modeled the particular section of the required road network in great detail, accurately reflecting road attributes such as road type, access restricions, 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.

Just like in the previous example, three sets of simulation runs were performed. One set of runs simulated uninhibited road traffic. In the second set of runs, a traffic incident was simulated by stopping the lead vehicle of cars travelling along the major artery connecting the university campus and the business park. In the final set of simulation runs, all vehicles were equipped with IVC technology, so stopped vehicles could disseminate information about congested road segments through a VANET. Vehicles that received such notifications could then often completely avoid traffic incidents.

Plotted in Figure 8(a) 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. 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 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, as well as other congested road segments, altogether.

The variance of the average speed is also shown in the boxplot in Figure 8(b). In this plot, only individual vehicles



(a) Average speed ordered by time of departure

(b) Average speed grouped by scenario

Figure 8: Average speed of individual vehicles for the free flowing traffic, traffic with an incident, and for UDP-based IVC

starting earlier than 400 seconds are considered to outline the characteristics of free flowing traffic, traffic queuing after an incident, and the advantages of UDP-based IVC. The latter scenario again outlines the need for bidirectional coupling of the simulation tools.

7. CONCLUSION

In conclusion, it can be said that bidirectionally coupled road traffic and network simulation provides major advantages compared to uncoupled or purely trace-driven simulation. The following findings support this observation:

- simple traffic models are inappropriate for road traffic simulation [27]
- 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 supplied to the mobility model [13, 18]
- using bidirectional coupling, the impact of IVC on road traffic can be directly evaluated [19, 24]

The simulation framework we developed, *Veins* (Vehicles in Network Simulation), provides all necessary functionality to perform this bidirectional coupling. It 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.

8. REFERENCES

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

[2] J. J. Blum, A. Eskandarian, and L. J. Hoffman. Challenges of Intervehicle Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):347–351, December 2004.

[3] 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. *IEEE Computer*, 33(5):59–67, May 2000.

[4] E. Brockfeld and P. Wagner. Testing and Benchmarking of Microscopic Traffic Flow Simulation Models. In *10th World conference on transport research (WCTR)*, Istanbul, Turkey, July 2004.

[5] N. Elloumi, H. Haj-Salem, and M. Papageorgiou. METACOR: A macroscopic modeling tool for urban corridors. *Triennal Symposium on Transportation Analysis*, 1:135–150, 1994.

[6] 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. In *SCS Multiconference on Distributed Simulation*, pages 3–11, January 2001.

[7] 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 Transactions on Vehicular Technology (ToVT)*, 56(6):3381–3396, November 2007.

[8] F. Hui and P. Mohapatra. Experimental Characterization of Multi-hop Communications in Vehicular Ad Hoc Network. In *2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005)*, Cologne, Germany, September 2005.

[9] M. Killat, F. Schmidt-Eisenlohr, H. Hartenstein, C. Rössel, P. Vortisch, S. Assenmacher, and F. Busch. Enabling Efficient and Accurate Large-Scale Simulations of VANETs for Vehicular Traffic Management. In *13th ACM International Conference on Mobile Computing and Networking (ACM MobiCom 2007): 4th ACM International Workshop on Vehicular Ad hoc Networks (VANET 2007)*, pages 29–38, Montreal, Quebec, Canada, September 2007.

[10] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner. SUMO (Simulation of Urban MObility);

An open-source traffic simulation. In *4th Middle East Symposium on Simulation and Modelling (MESM2002)*, pages 183–187, Sharjah, United Arab Emirates, September 2002.

[11] S. Krauß. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. PhD thesis, 1998.

[12] D. Li, H. Huang, X. Li, M. Li, and F. Tang. A Distance-Based Directional Broadcast Protocol for Urban Vehicular Ad Hoc Network. *International Conference on Wireless Communications, Networking and Mobile Computing 2007 (WiCom 2007)*, pages 1520–1523, 21-25 September 2007.

[13] C. Lochert, A. Barthels, A. Cervantes, M. Mauve, and M. Caliskan. Multiple simulator interlinking environment for IVC. In *2nd ACM International Workshop on Vehicular Ad hoc Networks (VANET 2005), Poster Session*, pages 87–88, Cologne, Germany, September 2005. ACM Press.

[14] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221–2229, 1992.

[15] M. Piorkowski, M. Raya, A. L. Lugo, M. Grossglauser, and J.-P. Hubaux. Joint Traffic and Network Simulator for VANETs. In *MICS 2006, Poster Session*, Zurich, Switzerland, October 2006.

[16] 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 Communications*, 13(4):104–119, 2006.

[17] C. Sommer, I. Dietrich, and F. Dressler. A Simulation Model of DYMO for Ad Hoc Routing in OMNeT++. In *1st ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008): 1st ACM/ICST International Workshop on OMNeT++ (OMNeT++ 2008)*, Marseille, France, March 2008. ACM.

[18] C. Sommer and F. Dressler. The DYMO Routing Protocol in VANET Scenarios. In *66th IEEE Vehicular Technology Conference (VTC2007-Fall)*, pages 16–20, Baltimore, Maryland, USA, September/October 2007. IEEE.

[19] C. Sommer, Z. Yao, R. German, and F. Dressler. Simulating the Influence of IVC on Road Traffic using Bidirectionally Coupled Simulators. In *27th IEEE Conference on Computer Communications (IEEE INFOCOM 2008): Mobile Networking for Vehicular Environments (MOVE 2008)*, Phoenix, AZ, April 2008. IEEE.

[20] N. B. Taylor. The CONTRAM Dynamic Traffic Assignment Model. *Networks and Spatial Economics*, 3:297–322, 2003.

[21] M. Treiber and D. Helbing. Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell. In *16. Symposium Simulationstechnik (ASIM 2002)*, Rostock, Germany, September 2002.

[22] M. Treiber, A. Hennecke, and D. Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, 62:1805, 2000.

[23] A. Varga. The OMNeT++ Discrete Event Simulation System. In *European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic, 2001.

[24] 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, and Network Researches. In *1st IEEE International Symposium on Wireless Vehicular Communications (WiVec 2007)*, Baltimore, MD, USA, October 2007. IEEE.

[25] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz. Routing in Sparse Vehicular Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, 25(8):1538–1556, October 2007.

[26] H. Wu, M. Palekar, R. Fujimoto, J. Lee, J. Ko, R. Guensler, and M. Hunter. Vehicular Networks in Urban Transportation Systems. In *National Conference on Digital Government Research (dg.o2005)*, Atlanta, GA, USA, May 2005.

[27] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, 2:1312–1321, 30 March - 3 April 2003.