Verification and Evaluation of Fail-Safe Virtual Traffic Light Applications

Till Neudecker, Natalya An, and Hannes Hartenstein
Institute of Telematics & Steinbuch Centre for Computing
Karlsruhe Institute of Technology, Germany
Email: {till.neudecker, natalya.an, hannes.hartenstein}@kit.edu

***Abstract*—The purpose of Virtual Traffic Light (VTL) applications is to increase traffic efficiency without the use of conventional traffic light infrastructure. With VTL applications, vehicles self-organize for intersection crossing based on wireless communication. Although VTL applications must comply with high safety requirements, no verification of VTL's safety has been provided so far. We present a VTL protocol that is verified to be fail-safe using model checking as a verification approach. Performance evaluation through simulation showed that the verified fail-safe VTL protocol delivers—although far from optimum—decent results with respect to such traffic efficiency metrics like throughput and travel time. The investigated efficiency optimization substantially improved efficiency, yet compromised the safety of VTL. We quantify the tradeoff between efficiency and safety and show that increasing the safety level deteriorates efficiency only marginally. In particular, a safety level increase by a factor of 1,000 increases travel time by approximately 2 % in the studied scenario. Therefore, high safety requirements can be met while maintaining high efficiency gains.**

I. Introduction

One promising approach to managing intersection crossing is one in which vehicles self-organize each other based on wireless communication. The general idea is that vehicles decide between each other on an intersection crossing order, via wireless communication and without the use of infrastructure, i.e., conventional traffic lights. The decision on whether a vehicle is allowed to cross an intersection or not is displayed to the driver through an on-board unit, and is based on a certain predefined algorithm and information received over communication. This approach has been discussed in several research papers, e.g., [15], [6], and [13]. We refer to such an approach with the general term "Virtual Traffic Light" (VTL), as proposed in [6], as it eliminates the use of conventional traffic lights. The VTL approach has the potential to improve traffic efficiency at unregulated intersections, like those equipped with stop signs, because vehicles do not need to brake if they have a virtual green light or if there are no other vehicles at the intersection. Dynamic, on-demand traffic light phase shifting also offers an increase in traffic efficiency and is possible with VTL. Nevertheless, the VTL application is a very ambitious application as it requires a 100 % equipment ratio. And although preliminary studies show that VTL application improves traffic efficiency [6] together with general feasibility against adverse radio conditions [13], the VTL application is not yet verified to be a fail-safe application. If a virtual traffic light signal is not available when approaching an intersection, a driver is not aware of whether he or she can safely cross the intersection or whether to brake in order to timely stop before the intersection. Although VTL is essentially an efficiency application, strong safety requirements have to be considered because of vehicle collision risks. When it comes to vehicular traffic, there is an intuitive tradeoff between efficiency and safety: As an extreme case, vehicles are safe with respect to accidents when they are not moving. In the same way, high mobility or high efficiency might cause impaired safety.

In the current paper, we answer the question of whether a VTL protocol can be designed as a fail-safe protocol; specifically, we verify it to be safe in all situations. Typically, there are not enough protocol details available in publications that are necessary for verification, so we model a new VTL protocol. In our modeling approach we take the general idea of VTL and consider conventional traffic lights as a reference for safe intersection management. The verification of our VTL protocol is done with *model checking*, in particular using PROMELA modeling language and SPIN model checker [8]. Thus, this paper presents a modeling of a fail-safe VTL protocol as well as its verification. The verification approach that we follow is independent of any specific communication technology, communication possibilities in general, or specific vehicle speeds or intersection layouts. Next we evaluate our verified fail-safe VTL protocol, based on IEEE 802.11p communication and ns-3 simulations, with other intersection management solutions like conventional traffic lights and the first-come, first-served (FCFS) approach. Although, the verified and fail-safe VTL protocol is far from optimum, it does show decent results with respect to efficiency. In particular, it is shown that for medium to high vehicular densities, the verified fail-safe VTL protocol outperforms such approaches as FCFS, which is usually used at unregulated intersections. The fail-safe VTL protocol is shown to be less efficient when compared with conventional traffic lights; thus, two efficiency optimization techniques are discussed. Naturally, efficiency optimization compromises the fail-safe property of the VTL protocol. Finally, we quantify the tradeoff between safety and efficiency for the VTL protocol. Such an analysis allows the classification of the system's safety according to safety standards like ISO 26262 [1], which describes the safety norms to which automotive equipment has to abide.

The remainder of this paper is structured as follows: Section II provides an overview of general verification approaches as well as verification of vehicular applications. The next section describes our modeling of VTL and its safety requirements. In Section IV we present our verification method of the modeled VTL protocol as well as verification results. In Sections V and VI we discuss the performance evaluation of the VTL protocol and the tradeoff between safety and efficiency. Finally, Section VII concludes the paper.

978-1-4799-2687-9/13/$31.00 ©2013 IEEE

II. Related Work—Verification

When considering the verification of vehicular applications, three main aspects arise that influence the general safety of such applications:

- **Kinematics:** Represents continuous movement of the vehicle within physical limits, hence dealing with time, speed, acceleration, etc;
- **Control:** Represents the algorithm running on each vehicle, typically modeled as a state machine, hence operating on discrete values, states, and events;
- **Communication:** Represents the (imperfect) information exchange among vehicles that is used by the control layer for its decisions; packets and reception probabilities are the common objects.

The safety of applications must be verified for all three aspects, but the differences in the used operands (e.g., continuous values and discrete events) make the verification of vehicular applications a challenge.

According to [7], the commonly used verification methods include *model checking* and *theorem proving*. Model checking is a method to check whether a model meets a given constraint. The model is described as an algorithmic description using a modeling language. Theorem proving, in contrast, makes use of mathematical methods to (semi-)automatically create a proof for a given model. It is therefore more abstract than model checking as the model is specified using first or higher order logic.

Loos et al. [10] verified an intersection control system by combining the aspects of continuous driving dynamics with those of the controlling algorithm (*hybrid systems*). Their method falls under the theorem proving category in which differential dynamic logic is used to define the model; this makes it possible to express requirements on the timeliness of information that arrives at the control algorithm. However, no communication aspects have been modeled in this work. Asplund et al. [2] provided a verification for distributed vehicle coordination at intersections using Satisfiability Modulo Theories. Although communication has been modeled, strong assumptions, like all vehicles within an active area are aware of each other, have been made. Therefore, both works cover only the kinematics and control aspects of vehicular application.

The model checking tool SPIN [8] is one of the prominent verification tools and has been used to verify routing protocols [4] or privacy preserving authentication protocols for vehicular networks [3]. Both of these verified applications do not have a direct effect on the vehicle's kinematics. As model checking focuses on control and communication aspects, it is limited to discrete values, and the vehicle's continuous movement over time cannot be modeled.

The authors of [5] performed safety verification of VTL, based on the probability of disagreement, for a part of VTL protocol (the leader election process). In our approach we consider the whole VTL protocol, besides that we integrate aspect of kinematics into the control aspect by discretizing the vehicle's movement and use the SPIN model checker to verify the control and communication aspects. This is explained in Section IV.

III. Protocol Model and Safety Requirements

The idea of Virtual Traffic Lights is discussed in several research papers, e.g., [15], [6], [13]. In summary, the idea of VTL is based on the self-organization of vehicles, through wireless communication, to manage intersection crossings. Vehicles do not use conventional traffic light infrastructure, but rely on information received over communication. The virtual traffic signal or right of way is displayed to the driver through the on-board unit. The wireless communication technology that is typically envisioned for VTL is a vehicular communication technology based on IEEE 802.11p. In such networks all vehicles are sending and receiving beacon messages that contain information on their ID, speed, and so on. Upon receiving such beacons, vehicles gain neighborhood awareness and, based on some defined rules, agree on the traffic light schedule for intersection crossing. The VTL models proposed in the literature, although show potential traffic efficiency benefits, are not verified to be fail-safe. Fail-safe property is essential for such applications like VTL, as malfunctioning could lead to vehicular accidents.

The goal of this work is to present a fail-safe VTL protocol, i.e., a protocol that is verified to be safe in all situations. Typically, protocols in the related literature are not fully specified, to the degree of detail that is necessary for verification, so we model a new VTL protocol. Our model is based (1) on the general idea of decentralized self-organization of vehicles through communication for intersection crossing; (2) on the safety requirements of a conventional traffic light; and (3) on the terminology used in [6]. In [6] a vehicle temporarily takes on the role of infrastructure and controls the intersection; it is called a *VTL Leader*. When the VTL Leader leaves the intersection it performs a *Handover* of its leadership to another vehicle. The VTL Leader is elected among several *Cluster Leaders*. A Cluster Leader is a vehicle that is closest to the intersection on a road segment. Thus, a four-way intersection may have up to four Cluster Leaders.

The following design assumptions have been made: 100 % VTL equipment ratio; all vehicles obtain their position over GPS with at least lane-level precision; all vehicles possess digital maps; and all drivers are compliant and not distracted. Additionally, a vehicle is assumed to be able to "sense" whether another vehicle is driving in front or whether it is the first vehicle on a particular road segment to reach the intersection (i.e., if it is a Cluster Leader). This can be achieved either under line-of-sight (LOS) communication conditions or with the fusion of in-vehicle sensors. No assumptions on the reliability of non–line-of-sight (NLOS) communication conditions are made, i.e., there is no assumption that Cluster Leaders on each road segment can reliably communicate with each other.

A. VTL Safety Requirements

To define safety requirements that are imposed on VTL for a fail-safe operation, we consider conventional traffic lights as a reference for safe intersection management. We assume that conventional traffic lights are safe as long as drivers are compliant and not distracted. Conventional traffic lights are designed with following safety constraints [14]:

- Vehicles with a red light displayed must be able to safely stop in front of the intersection;
- Conflicting vehicles must not enter the intersection at the same time. Two vehicles are conflicting if their paths may try to occupy the same physical space at the same time [14].

The first constraint can be fulfilled by using a yellow light phase. The duration of the yellow light phase, which is dependent on the vehicle's speed v and the driver's comfortable deceleration a_{comf}, allows for safe braking in front of an intersection or a legal crossing of it. Thus, we specify the *safety distance* or D_{safe} at which the information on a traffic light has to be available for the safe crossing of an intersection or the timely stopping in front of it. The traffic light systems, conventional or virtual, have to provide the current traffic light phase to the driver at latest, at this safety distance. The safety distance is calculated as $D_{\mathrm{safe}} = v^2/(2 * a_{\mathrm{comf}})$.

The second requirement is fulfilled if a green light is only assigned to roads with no conflicting vehicles. Thus, the two main requirements of virtual traffic lights can be formulated as follows:

- Drivers get notified on the current traffic light phase at latest at distance D_{safe}; and
- Conflicting vehicles must not get the right of way (i.e. green light) at the same time.

These requirements, the general idea of self-organization of vehicles for intersection crossing, and terminology used in [6] serve as a basis for the VTL protocol that we model. The concept of the modeled VTL protocol and some protocol primitives are presented in the following subsections.

B. VTL Concept

Our virtual traffic light *concept* follows these abstract steps:

- Upon approaching an intersection, vehicles try to determine whether there is already a vehicle, called a *VTL Leader*, that controls the intersection, and broadcasts traffic light information;
- If there is a VTL Leader, all vehicles receiving its messages obey the traffic signal;
- If there is no VTL Leader, vehicles determine among possibly several approaching vehicles, called *Cluster Leaders*, which one takes the VTL Leader role;
- If no VTL Leader is present and it is not possible to determine a new VTL Leader because of insufficient information, vehicles brake at D_{safe} and cross the intersection in a FCFS manner, i.e., they perform a *fallback* approach;
- A VTL Leader can, before leaving the intersection itself, perform a handover so that another vehicle becomes the VTL Leader and overtakes control of the intersection.

In a realistic intersection environment, it is possible that vehicles receive insufficient or even no information at all during the intersection approach, even though the information has been sent. For VTL application to be fail-safe, we do not make assumptions on the reliability of NLOS communication, that is why it is necessary that vehicles perform a fallback approach if information is insufficient.

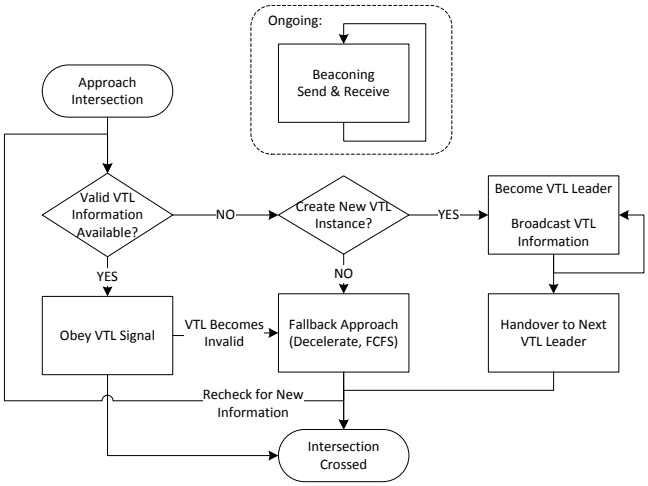

Fig. 1. Virtual Traffic Light protocol state diagram

C. VTL Protocol Primitives

Figure 1 depicts the VTL protocol presented in Section III-B as a state change diagram. The two most critical aspects are the Leader Election, i.e., whether a vehicle can establish a new VTL instance, and the Handover.

The two main requirements of the VTL protocol that are defined in Section III-A serve as the basis for the presented protocol. Because of imperfect wireless communication conditions, it cannot be guaranteed that drivers will get notified on the current traffic light phase before or at D_{safe}. However, if at D_{safe} no information on the traffic light is available, the vehicle has to brake in order to safely come to a stop before the intersection; otherwise a vehicle collision might happen. As shown in Fig. 1, this case is incorporated into the protocol design: If no valid VTL information is available, and it is not possible to create a new VTL instance (vehicle creates new VTL instance by becoming a VTL Leader and broadcasting traffic light information), the vehicle decelerates after crossing D_{safe}. The intersection crossing is then performed in an FCFS manner.

The second requirement (*consistency* of traffic light information) is easy to fulfill with a single instance that is responsible for the intersection management. For a VTL protocol, it is therefore necessary to ensure that at most one vehicle is a VTL Leader at an intersection at each moment, except during the Handover process, which is described subsequently. Hence, a vehicle may only become the VTL Leader if it can guarantee that no other vehicle becomes the VTL Leader as well. The decision *Create New VTL Instance?* shown in Fig. 1 is made as follows: The number of vehicles that can become the VTL Leader is limited to those vehicles that are Cluster Leaders. It is assumed that a vehicle can detect if it is closest to the intersection on its own road segment, and thereby determine whether it is a Cluster Leader. Therefore, the total number of vehicles that can become the VTL Leader at one intersection equals the number of approaching road segments. Using map

data, this information is known to all vehicles. In order to prevent multiple VTL instances, we specify that a vehicle can only become the VTL Leader if it possesses valid information from all other road segment's Cluster Leaders, and if it has the highest unique ID among them.

In case information from all approaching road segments is not available, a vehicle has to perform the fallback approach, i.e., brake and cross the intersection in an FCFS manner. This procedure is inefficient in case of the intersection being empty, but it is fail-safe.

After the VTL Leader decides to leave the intersection, it can handover its leadership to another vehicle by setting a specific flag in its beacons. The VTL handover is executed gradually, with a period of time, when two vehicles are VTL Leaders. It must be ensured that no inconsistent traffic light information is announced by both vehicles during that period. We therefore forbid the *new* VTL Leader to change the traffic light signal within an interval defined by the *old* VTL Leader. The VTL Leader does not need to perform a handover, and can simply leave the intersection; his virtual traffic information will then simply expire after a certain period of time.

The fallback option, the VTL requirement to possess information on Cluster Leaders from all intersecting road segments, and the limited validity of VTL information are just a few of the essential differences to the VTL protocols described in [6] and [13].

IV. Our Verification approach

In the following section we describe the approach we use to verify the VTL protocol presented in Section III. Before the presentation of our verification approach in Section IV-B and the verification results in Section IV-C, a short introduction into the used model checking method is provided.

A. Model Checking Basics

As mentioned in Section II, several tools and approaches for model checking exist. We chose SPIN with its modeling language PROMELA because of its communication support, powerful process modeling and mature technology. The VTL protocol described in Section III is implemented in PROMELA and verified using the SPIN model checker.[1]

SPIN provides two variations for verification: exhaustive search and bitstate hashing. An exhaustive search verification begins with an initial state, evaluates every possible next system state, and checks whether the defined invariant holds true. The *system state* of a model consists of all processes' state variables and their current instruction, as well as the state of all global message channels. To avoid multiple evaluations of the same state, every evaluated state must be stored. Although this is feasible for small models, the verification of larger models can require a lot of memory for the state storage. For example, a model that requires 100 Byte to store one system state and has a total number of 10^9 reachable system state requires over 93 GByte of memory. Although compression mechanisms can reduce the memory consumption, larger models easily reach the memory limits of today's hardware.

[1]The model is provided at http://dsn.tm.kit.edu/english/vtl.php

For models that are too large to be exhaustively verified, a bitstate search can be used. Instead of storing each system state completely, a hash function is used to map each system state to a single number between 0 and $2^w - 1$. The constant w is chosen so that 2^w Bit fit in the computer's memory. During verification, the corresponding bit of the calculated hash value of each evaluated state will be set in a bitmap. To check for already evaluated states, the verification algorithm hashes the current state and checks whether the corresponding bit in the bitmap is already set. Therefore, the storage of each state requires only one Bit of memory. The big drawback is that it is possible for so-called *hash collisions* to occur (i.e., two different states are mapped to the same hash value). If this happens during verification, the algorithm will treat a new state as a known state and therefore will not evaluate it. Hence, the search is not necessarily exhaustive, but a partial, randomized search. The coverage (the number of reached systems states divided by the number of reachable system states) of a bitstate search strongly depends on the size of the bitmap, as the likeliness of hash collisions decreases with larger bitmap sizes. Hence, choosing a large value of w is recommended for best coverage. The bitmap's occupancy ratio (the number of visited states divided by the total number of bits, also called hash factor, h) serves as an indicator for the search coverage. A hash factor close to 1 indicates a full bitmap, which causes too many hash collisions, whereas a large hash factor ($h > 100$) is an indicator of good coverage [8].

B. Verification Method

In this section we describe the approach we use to verify the VTL protocol presented in Section III. Model checking only covers the control and the communication aspect. In order to provide a verification of the kinematic aspect, we integrated the continuous movement of the vehicle into the control algorithm: The VTL model implemented in PROMELA models a vehicle either before or after D_{safe}. Crossing D_{safe} is implemented as an indeterministic statement. Recall the two VTL requirements defined in the previous section: (1) Drivers must get notified on the current traffic light phase at latest, at D_{safe}; and (2) conflicting vehicles must not get the right of way at the same time. Integrating the safe braking distance D_{safe} into the VTL model eliminates the need for an additional verification of kinematics as long as the control layer is able to correctly sense the distance to the intersection. Therefore, only requirement 2 (consistency of traffic light signals) must be verified.

The consistency of traffic light signals is ensured if vehicles approaching an intersection either possess the same traffic light signal set or do not possess any traffic light signal at all, as vehicles perform a fallback approach in this case. A *traffic light signal set* consists of the traffic light phase for each approaching lane of the intersection. We will define this formally as it is the invariant of our verification: Let V be the set of all vehicles at one intersection and $v \in V$ be a vehicle. A traffic light signal set $s = (l_1, ..., l_n)$ is a tuple that indicates the current light phase for each lane and direction ($l_{1,...,n} \in \{\text{red}, \text{green}\}$; n is the number of approaching lanes); and S denotes the set of all possible traffic light signal sets.[2]

[2]S includes signal sets that must not occur (e.g., all lanes green). We assume that VTL Leaders do not broadcast such signal sets with the help of map data.

For each vehicle v, $s_v \in S \cup \{\perp\}$ denotes the traffic light signal set possessed by v. If a vehicle does not possess any valid traffic light signal set, then $s_v = \perp$.

The consensus invariant that must hold is then:

$$\exists \sigma \in S \; \forall v \in V : s_v = \sigma \vee s_v = \perp$$

The goal of our verification is to ensure that in no reachable system state the defined invariant is violated.

C. Verification Results

Verification runs for the following scenarios with one intersection and varying numbers of vehicles and approaching road were performed: Two vehicles and two one-way roads, three vehicles and two one-way roads, three vehicles and three one-way roads, and four vehicles and two one-way roads.[3] For the initial state, all vehicles are located at a distance larger than D_{safe} away from the intersection and do not possess any traffic light information. Table I lists the number of evaluated system states as well as the required memory for each scenario. Most importantly, no violation of the consistency constraint or deadlock could be found by SPIN in any scenario.

TABLE I. VERIFICATION RESULTS

Setup		Output			
Vehicles	**Roads**	**States**	**Errors**	**Memory**	**Comment**
2	2	$267,441$	0	34 MB	Exhaustive
3	2	$5.42 * 10^9$	0	216 GB	Exhaustive
3	3	$1.91 * 10^{10}$	0	256 GB	Bitstate, $h = 115$
4	2	$> 10^{11}$	0	256 GB	Bitstate, $h < 22$

When the number of vehicles increases, the verification complexity increases drastically: Whereas the scenario with two vehicles only required 34 MByte of memory and 1.37 s of computation time, adding one vehicle to the scenario caused the verification to require 216 GByte of memory and 7.2 h of computation time. Because of the hardware memory limit of 512 GByte, only a bitstate verification could be performed for larger scenarios. The scenario with three vehicles and three roads could be verified with sufficient coverage as indicated by a hash factor of 115. However, scenarios with four vehicles or more could not be verified with sufficient coverage because of memory and computation time restrictions. This also indicates general limitations of the used model checking approach.

V. EVALUATION

In the current section we present a simulation-based performance evaluation of the verified VTL protocol. The goal of this evaluation is to assess how the verified VTL protocol performs in a close-to-real-world environment and in comparison with other intersection management approaches.

A. Scenario

To evaluate the VTL protocol's performance in realistic environments, various models are required to represent the reality. Particular models of interest are the networking model, the mobility model, the radio prorogation model, and the human model. The verified VTL protocol was implemented

[3]All verification runs were performed on a 2.67 GHz Intel Xeon E7-8837 machine equipped with 512 GByte of main memory.

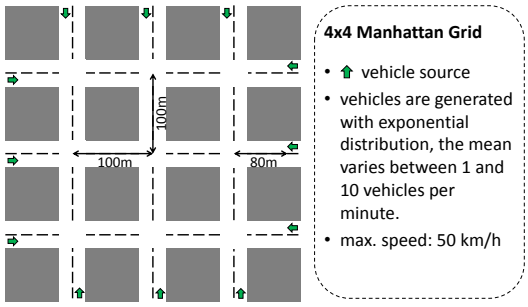

Fig. 2. Manhattan Grid 4x4 scenario.

in the ns-3 network simulator,[4] the vehicles' movement was modeled by the Intelligent Driver Model (IDM) [9]. Radio propagation on the intersection is complex and requires sophisticated models. For our evaluation we chose the Virtual-Source11p [11] radio propagation model because it is based on real-world measurements in the intersection environment. The *interbuilding distance* (IBD) is one of the main parameters that affect radio propagation at intersections, especially for communication between intersecting roads (NLOS conditions). Obviously, larger IBDs improve radio propagation conditions, whereas smaller IBDs impair them. The human model was not fully considered for simplicity reasons and because of limited data on human driver behavior. We modeled an obedient, not distracted driver, who is able to respond immediately to traffic light information.

The analyzed road layout represents a 4x4 Manhattan grid, cf. Fig. 2. Generated vehicles simply drive straight until they cross the whole grid and disappear at the other end. We compared the performance of verified VTL with the following conventional intersection management approaches:

- *All-way stop*, which realizes the first-come, first-served (FCFS) principle; and
- *Conventional traffic light (CTL)*, which represents dedicated traffic lights that are installed at each intersection and perform phase switching in a static, pretimed manner (fixed phase duration of 30 s). This is done either *asynchronously*, i.e., random switch at each intersection, or *synchronously*, i.e., green or red wave for all horizontal, or all vertical roads.

B. Efficiency Evaluation

For the evaluation of VTL we looked at the *throughput* and *travel times* of various intersection management approaches, including VTL support with respect to different vehicular densities.

Figure 3 shows the throughput, represented as the number of outflowing vehicles with respect to inflowing vehicles. In other words, we evaluated whether the number of vehicles that leave the grid is the same as the number of vehicles entering the grid. Obviously, if the inflow is higher than the outflow, the road network reached its maximum throughput capacity. As shown in Fig. 3, all intersection management approaches perform similarly for vehicular densities below five vehicles per minute, per source. As vehicular density grows, FCFS

[4]http://www.nsnam.org, version 3.15.

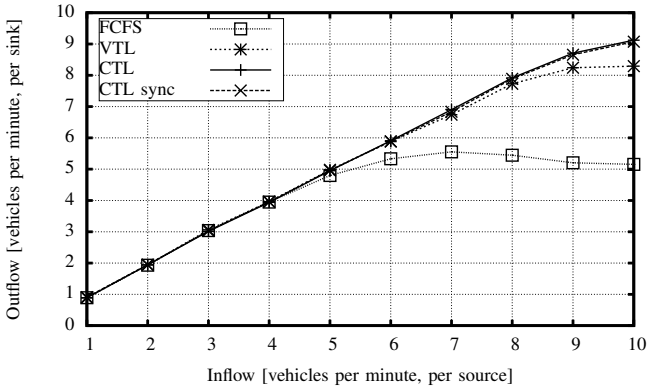

Fig. 3. Throughput vs. vehicle density. The IBD is 22 m, TxRate is 10 Hz, and $v = 50\,\mathrm{km/h}$

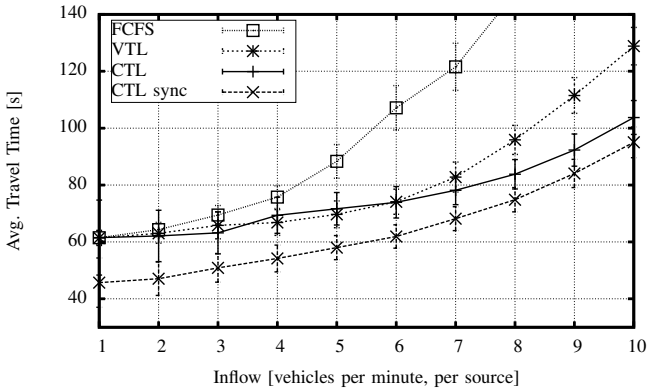

Fig. 4. Average travel time vs. vehicle density with 95 % confidence intervals. The IBD is 22 m, TxRate is 10 Hz, and $v = 50\,\mathrm{km/h}$

starts to throttle, whereas VTL still performs close to optimum until vehicular density is approximately seven vehicles per minute, per source. With the increased density, VTL, as well as conventional traffic light approaches, begin to deteriorate because the road network is now saturated, although VTL is slightly worse. These results indicate that VTL can outperform FCFS approaches and perform close to CTL with respect to throughput.

Figure 4 depicts the average travel time that vehicles require to cross the grid from the moment they enter at the source until they reach the sink. For small vehicle densities, i.e., 1–2 vehicles per minute, per source, VTL's performance is similar to that of FCFS and asynchronous CTL. Because there are only a few vehicles, no VTL instance can be established in most cases, as it requires information from all four roads and VTL is switching to the fallback option. Because of low vehicle density, the benefit of asynchronous CTL is also not noticeable. Synchronized CTL perform by far the best. For comparison, the absolute best travel time for one vehicle is 25.7 s (360 m/14 m/s). With the increase in traffic density, FCFS performs noticeably worse than VTL, because VTL vehicles are now able to establish VTL instances and manage intersections without going into fallback. VTL performs similarly to asynchronous CTL; synchronized CTL, however, still performs best.

The results show that the verified VTL protocol performs better than FCFS with respect to throughput and travel time.

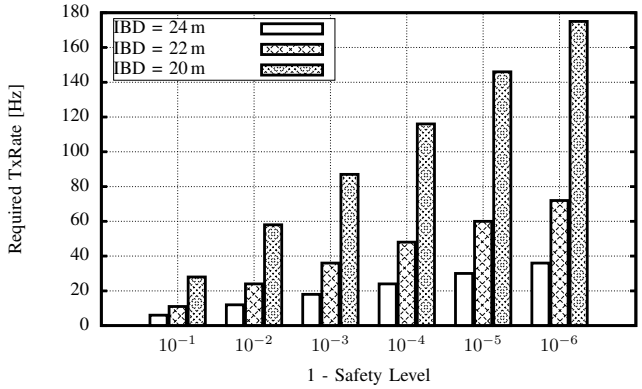

Fig. 5. Required TxRate vs. safety level at $D_{\mathrm{safe}} = 49\,\mathrm{m}$ for various inter building distances, $v = 50\,\mathrm{km/h}$, and $a_{\mathrm{comf}} = 2m/s^2$

However, for low vehicle densities VTL is worse than the synchronized CTL approach because of the lack of vehicles on all other road segments. Not detecting vehicles on other road segments could be the result of lost messages, or the intersection is really empty. It is possible to optimize the VTL protocol's efficiency by detecting that an intersection is, with high probability, in fact empty and there is no need to brake.

VI. Tradeoff: Safety vs. Efficiency

We investigated two modifications to the fail-safe VTL protocol that aimed to improve its efficiency while relaxing the safety constraint: *Adaptive Transmission Rate* and *Adaptive Braking Distance*. In low vehicle densities, high performance gains are possible if more VTL instances could be created, and vehicles would not have to brake at every single intersection because of a fallback FCFS approach. A vehicle that crosses an intersection without braking, however, must ensure that there is no conflicting vehicle approaching the intersection. It is therefore essential for vehicles to be able to determine whether a conflicting road is in fact empty, or whether all beacons originating from vehicles on a conflicting road got lost (e.g., result of bad channel conditions or packet collisions).

With the help of the VirtualSource11p [11] channel model, it is possible to determine the probability of packet reception, which allows us to calculate a *safety level*, i.e., a confidence level stating the probability that a road is really empty, depending on the sender's transmission rate (TxRate) and the sender's and receiver's positions. The safety level is calculated by approaching vehicles as the cumulative reception probability for at least one beacon, assuming that a present sender vehicle is approaching the intersection.

Adaptive Transmission Rate: To maximize efficiency by avoiding unnecessary braking maneuvers, it is desirable that approaching vehicles reach a sufficient safety level at D_{safe}. This could be achieved by adapting the sender's TxRate, depending on the desired safety level and the radio channel conditions that are strongly influenced by the intersection layout.

Figure 5 shows the required TxRate for various safety levels and three different, but common, intersection layouts. Naturally, the stricter the safety level requirement, the higher is the required TxRate. However, even the smallest presented safety level of 90 % requires a TxRate above 20 Hz for an IBD

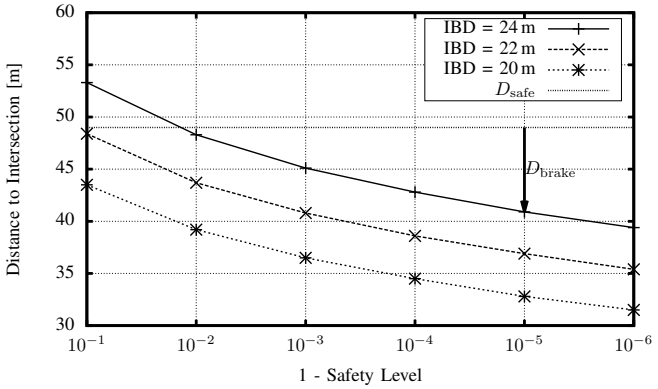

Fig. 6. Distance to intersection at which a safety level value can be reached for a fixed TxRate (10 Hz) after braking at D_{safe}

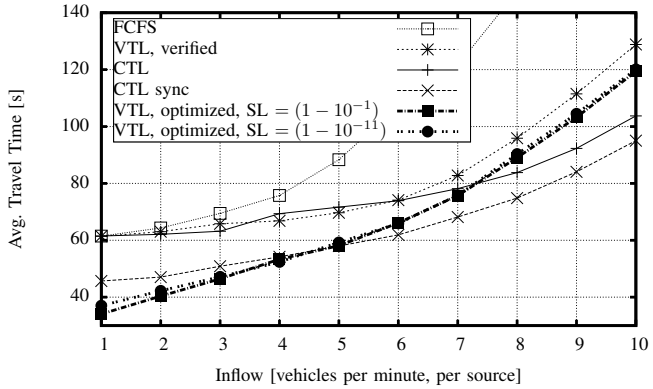

Fig. 7. Average travel time vs. vehicle density. The IBD is 22 m, TxRate is 10 Hz, and $v = 50\,\text{km/h}$. VTL includes three configurations: verified and performance optimized with two safety levels (90 % and 99.999999999 %)

of 20 m. The highest presented safety level requires a TxRate well above typically assumed communication capabilities, even for wide intersections. Another issue is that a slight change in the IBD has a nonlinear effect on the required transmission rate. This could cause large errors in required TxRate estimation in case of slight errors in IBD information. Although this approach would maximize efficiency, it is not feasible.

Adaptive Braking Distance: As sufficient safety levels cannot be achieved at a distance of D_{safe} (e.g., 49 m at $50\,\text{km/h}$), vehicles must decelerate at D_{safe} in order to maintain a safe state, but could continue to drive at their desired speed once a sufficient safety level is reached. The efficiency of this mechanism strongly depends on the braking distance, i.e., the distance between D_{safe} and the point where a sufficient safety level is reached.

We assume that all vehicles communicate at a fixed TxRate of 10 Hz and that all vehicles are aware of that fact. Figure 6 shows the distance to intersection at which a vehicle reaches a desired safety level. The length of the arrow labeled D_{brake} represents the braking distance (approximately 8 m), which is necessary to reach the desired safety level of $1 - 10^{-5}$ for an IBD of 24 m. Obviously, higher safety levels cause vehicles to reach a safety level later and, therefore, lead to longer braking distances that reduce efficiency. However, in contrast to the Adaptive Transmission Rate method, it is possible to achieve high safety levels at a reasonable efficiency loss. In addition, a variation of the IBD now has a limited effect on the distance that vehicles need to brake.

We implemented this mechanism to compare its efficiency gain compared with the verified and nonoptimized VTL protocol and the other intersection management methods evaluated in Section V. Two extreme safety levels, 90 % and 99.999999999 %, are depicted for comparison. As shown in Fig. 7, the efficiency-optimized VTL protocol outperforms synchronized CTL for low vehicle densities and outperforms the nonoptimized VTL protocol for all investigated vehicle densities. The effect of the two shown safety levels on efficiency turns out to be minimal, in the range of $< 3s$ or $< 9\%$ of additional travel time in the evaluated scenario. This is nonintuitive, as one could expect a higher efficiency penalty for such a high safety level gain, considering the reduced risk of falsely not detecting a conflicting vehicle by a factor of 10 billion. Further simulations showed that the efficiency penalty for a safety level increase by a factor of 1,000 is only approximately 2% for safety levels between $1 - 10^{-1}$ and $1 - 10^{-11}$.

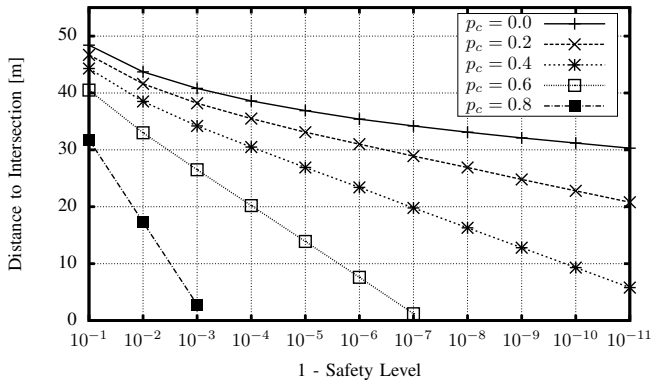

Fig. 8. Distance to intersection at which a safety level value can be reached for various packet collision probabilities $p_c$for a fixed TxRate (10 Hz) and constant speed ($50\,\text{km/h}$)

A. Discussion

Although the proposed mechanism improves VTL's efficiency, several limitations should be mentioned:

- The safety level calculation assumes independent probabilities of packet reception without accounting for the impact of burst errors;
- The radio channel model only accounts for path loss and fading, and not for packet collisions;
- Because of channel congestion, vehicles might have to reduce their TxRate.

We leave the first issue to future work, but briefly address the other two. Figure 8 shows the effect of independent packet collision probabilities ranging from 0 to 80 % on the distance at which a given safety level can be reached. Obviously, higher packet-collision rates worsen the performance of this optimization mechanism. However, up to a packet collision rate of 40%, all safety levels can be achieved before reaching the intersection. The occurrence of even higher packet-collision rates is unlikely during an approach of a presumably empty intersection.

Figure 9 shows the effect of various transmission frequencies on the braking distance. By defining a minimum TxRate that must be obeyed by the congestion control algorithm, it

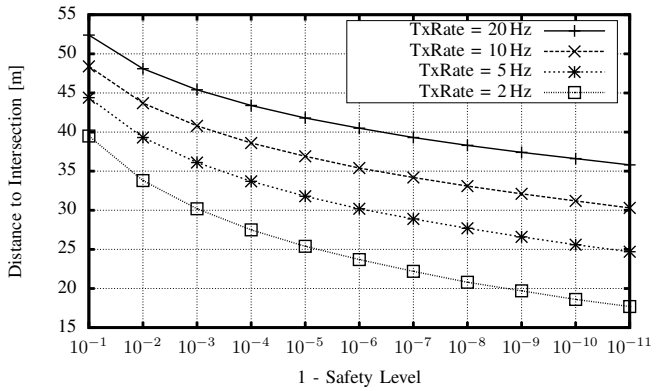

Fig. 9. Distance to intersection at which a safety level value can be reached for various TxRates and constant speed (50 km/h) and an IBD of 22 m

would still be possible to calculate a lower bound on the safety level.[5] Channel sensing approaches could also be integrated into VTL in order to estimate the real TxRate of other vehicles. Although there is more room for efficiency optimization, the evaluated mechanism does improve VTL's performance, while maintaining high and configurable safety levels.

VII. Conclusion

In the current paper, we analyzed whether a VTL protocol can be a fail-safe protocol and in case it is safe, how efficient is it. We first modeled a VTL protocol with safety requirement constraints and then verified it to be fail-safe using model checking. Although the verification could only be successfully performed for small scenarios, the results indicate that the presented protocol is indeed fail-safe. Simulation results show how efficient the verified fail-safe VTL protocol is when compared with other intersection management solutions, such as conventional traffic light and FCFS. Applying the efficiency optimization technique that detects empty roads based on the nonreception of beacons to the protocol resulted in a significant efficiency gain (up to 44 %). However, the efficiency increase led to the fact that safety can only be guaranteed to a certain degree. A parameter *safety level* is used as a quantification of the VTL system's safety. Further simulations revealed that the safety level's effect on efficiency, i.e., the tradeoff between safety and efficiency, is only approximately 2 % of the efficiency penalty for a safety level increase by a factor of 1,000 (for the shown scenario setup). This allows the use of extremely high safety levels as specified by automotive industry standards [1] with only marginal loss in efficiency.

Further efficiency optimization techniques can be also investigated, e.g., incorporation of channel congestion and dynamic transmission rate adaptation. Traffic light phase shift optimization has a potential to improve efficiency as well, but has to consider a fairness among all vehicles (also considering cheating vehicles). The protocol itself could be further optimized by e.g., allowing several crossing vehicles on the intersection area simultaneously, as long as they will not collide. In our analytical studies we assumed independent packet reception probability, but the effect of burst errors on application should also be studied. Additionally, we verified a VTL protocol modeled with the PROMELA modeling language and transferred its implementation into the ns-3 simulator. An approach as, e.g., *equivalence checking* could be performed to verify the ns-3 implementation.

As it has already been stated, the VTL application is a very ambitious application that requires a 100 % equipment ratio. Other traffic players, such as bicyclists and pedestrians, as well as nonequipped vehicles could be integrated through, e.g., smartphones [12]. Future work can consider more complex scenarios, e.g., realistic driving patterns, such as left/right turns, priority roads as well as the impact of human errors. Thus, VTL is still a visionary and ambitious research issue.

[5]The minimum TxRate could be defined per intersection and stored in map data available for all vehicles.

Acknowledgements

The work on VTL initially began during a student exchange program of Till Neudecker at the research group of Prof. Ozan Tonguz at Carnegie Mellon University. Natalya An acknowledges the support of the Ministry of Science, Research and the Arts of Baden-Württemberg (Az: Zu 33-827.377/19,20), and the Klaus Tschira Stiftung for the junior research group on Traffic Telematics. The authors would like to thank Jens Mittag and the anonymous reviewers for their valuable comments and to acknowledge the use of the InstitutesCluster II at the Steinbuch Centre for Computing (SCC) as well as thank SCC's staff for their technical support.

References

[1] ISO 26262: Road Vehicles–Functional Safety, http://www.iso.org, 2011.

[2] M. Asplund, A. Manzoor, M. Bouroche, S. Clarke, and V. Cahill. A formal approach to autonomous vehicle coordination. In *FM 2012: Formal Methods*. Springer, 2012.

[3] B. K. Chaurasia, S. Keshari, S. Verma, and G. Tomar. Verification of privacy preserving authentication protocol for vanets. In *Proceedings IEEE CICN*, 2010.

[4] R. de Renesse and A. Aghvami. Formal verification of ad-hoc routing protocols using spin model checker. In *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference, MELECON*, 2004.

[5] N. Fathollahnejad, E. Villani, R. Pathan, R. Barbosa, and J. Karlsson. On reliability analysis of Leader Election protocols for Virtual Traffic Lights. In *Proceedings IEEE/IFIP DSN*, 2013.

[6] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, and O. K. Tonguz. Self-organized traffic control. In *Proceedings of the 7th ACM International Workshop on VANET*, 2010.

[7] O. Grumberg and H. Veith. *25 Years of Model Checking: History, Achievements, Perspectives*. Lecture Notes in Computer Science. Springer, 2008.

[8] G. J. Holzmann. *Design and Validation of Computer Protocols*. Prentice Hall, 1990.

[9] Longitudinal Traffic model: The Intelligent-Driver Model (IDM). http://www.vwi.tu-dresden.de/~treiber/MicroApplet/IDM.html.

[10] S. M. Loos and A. Platzer. Safe intersections: At the crossing of hybrid systems and verification. In *Proceedings IEEE ITSC*, 2011.

[11] T. Mangel, O. Klemp, and H. Hartenstein. 5.9 GHz inter-vehicle communication at intersections: a validated non-line-of-sight path-loss and fading model. *EURASIP Journal on Wireless Communications and Networking*, 2011.

[12] M. Nakamurakare, W. Viriyasitavat, and O. K. Tonguz. A prototype of Virtual Traffic Lights on android-based smartphones. In *Proceedings IEEE SECON*, 2013.

[13] T. Neudecker, N. An, O. K. Tonguz, T. Gaugel, and J. Mittag. Feasibility of Virtual Traffic Lights in non-line-of-sight environments. In *Proceedings of the 9th ACM International Workshop on VANET*, 2012.

[14] R. Roess, E. Prassas, and W. McShane. *Traffic Engineering*. Pearson Education, 2010.

[15] J. Wu, A. Abbas-Turki, and A. E. Moudni. Discrete methods for urban intersection traffic controlling. In *Proceedings IEEE VTC Spring*, 2009.