SDN-based Misbehavior Detection System for Vehicular Networks

Abdelwahab Boualouache, Ridha Soua, and Thomas Engel
SnT, University of Luxembourg, Luxembourg
Email: {abdelwahab.boualouache, ridha.soua, thomas.engel}@uni.lu

***Abstract*—Vehicular networks are vulnerable to a variety of internal attacks. Misbehavior Detection Systems (MDS) are preferred over the cryptography solutions to detect such attacks. However, the existing misbehavior detection systems are static and do not adapt to the context of vehicles. To this end, we exploit the Software-Defined Networking (SDN) paradigm to propose a context-aware MDS. Based on the context, our proposed system can tune security parameters to provide accurate detection with low false positives. Our system is Sybil attack-resistant and compliant with vehicular privacy standards. The simulation results show that, under different contexts, our system provides a high detection ratio and low false positives compared to a static MDS.**

***Index Terms*—Vehicular Networks; Software Defined Networks; Security; Privacy; Misbehaving Detection Systems**

I. Introduction

Vehicular networks offer interesting applications ranging from safety-related applications to comfort applications. However, vehicular networks are vulnerable to many types of internal and external attacks such as message droppings and false information injections that can lead to hazardous situations for drivers and passengers. While external active attacks can easily be avoided using cryptographic solutions, internal attacks are difficult to avoid using these same solutions since internal attackers are authenticated members in the considered network [1]. Alternatively, using Misbehavior Detection Systems (MDSs) is considered as an efficient way to detect internal attacks. MDSs generally use two detection mechanisms [2]: (i) node-centric: this mechanism is primarily interested in nodes (vehicles or RSUs). For example, an MDS can continuously monitor the forwarding behavior of nodes and calculate the ratio between the receiving packets and forwarded packets of each node, and thereby it can decide on the trustworthiness of nodes; and (ii) data-centric: this mechanism is primarily interested in data rather than nodes. For example, an MDS can check the correctness using plausibility and consistency methods and, thereby evaluate the trustworthiness of these messages. Most proposed MDSs adopt a combined approach where a node-centric mechanism is used to evaluate nodes according to the correctness of the exchanged data, while the correctness of data is verified using a data-centric mechanism.

Although the significant number of proposed MDSs for vehicular networks, several questions are still open: (i) Most of existing MDSs are not compliant with current vehicular privacy standards which expect that vehicles use a set of pseudonyms (temporal identifiers) to protect their location privacy [3]. Indeed, these MDSs assume that nodes use fixed identifiers; (ii) Although pseudonyms are beneficial for location privacy, they could be used to generate Sybil attacks, which can significantly affect the performances of MDSs. For instance, an attacker can use pseudonyms as Sybils to influence on a vote decision on the trustworthiness of nodes; and more importantly (iii) all existing MDS are static and do not take into the account the context where vehicles are evolving such as mobility, number of attackers, and the network performance.

To address these issues, we exploit the Software-Defined Networking (SDN) paradigm to propose a context-aware MDS. The control plane in this system is responsible for the secure clustering, the election and deployment of the watchdogs, the dynamic adjusting of different security parameters like the threshold of detection according to context to provide an accurate detection with low false positives. Our system is also based on a generic and flexible trust model that can be adapted to detect any attacks on vehicular networks. In contrast to previous proposed MDSs, our system is Sybil attack-resistant and compliant with vehicular privacy standards.

II. Related Work

As previously mentioned, most of the existing MDSs use a hybrid approach that combines node-centric and data-centric detection mechanisms. The authors of [4, 5] propose a scheme to evaluate the reputation of vehicles based on their behaviors and the quality of provided information. The reputation values provided by each Cluster Member (CM) are periodically reported to the Cluster Head (CH). Each detected attacker is also reported to the CH. When a CH receives an attacker detection report, it leverages on a vote decision and the received reputation values to check the correctness of this detection. The proposed scheme also periodically calculates the trust levels of vehicles based on their reputation. However, this solution suffers from a significant overhead due to clusters management complexity and a large number of monitoring vehicles. In addition, the security parameters such as trust thresholds and the evaluation period, are not dynamically adjusted. Moreover, this scheme is not privacy-preserving and vulnerable to Sybil attacks. The authors of [6] outline the importance of considering the context where vehicles are evolving to increase the efficiency of MDSs. Indeed, their results show that the performance of MDSs varies from one

context to another. However, this study only considers two parameters of the context: the mobility of both vehicles and attackers. The authors of [7] propose an MDS for software-defined vehicular networks where vehicles analyze the incoming traffic and forward some selected data flows to the SDN controller. On the basis of these data flows, the SDN controller trains a multi-classifier model based on SVM. The parameters of the trained model are forwarded to vehicles to be used in the detection of misbehaving vehicles. However, many details on how the model is trained are missing. In addition, once the model is trained, it cannot be updated according to the context of vehicles. Finally, the performance evaluation is based on the KDD data set, which does not reflect the intrinsic characteristics of vehicular networks. Recently, the authors of [8], propose an SDN-based framework for 5G vehicular networks where the control plane consists of two modules: an authentication module and a misbehavior detection module. The proposed MDS uses a data-centric mechanism to verify the consistency of the data. However, the proposed MDS is again static and not context-aware.

III. System model and MDS description

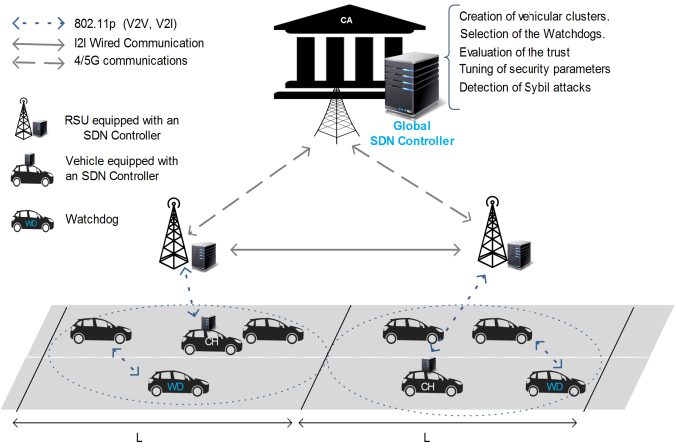


Fig. 1: Software defined vehicular network architecture for MDS

As illustrated in Figure 1, we consider a software-defined vehicular network architecture consisting of vehicles, Road Side Units (RSUs), and the Certification Authority (CA). This architecture has three levels of SDN control: (i) Local SDN controllers, which are installed on each Cluster Head (CH). The role of these controllers is to select the Watchdogs according to the strategy described in Section IV-B and to calculate the trust level of Cluster Members (CMs); (ii) Regional SDN controllers, which are installed on RSUs. These controllers calculate trust levels of local SDN-controllers and aggregate the trust level of vehicles; and (iii) Global SDN controller, which is installed at the CA and has global knowledge of the software-defined vehicular network. The global SDN-controller creates the vehicular clusters, selects CHs (see Section IV-A) tunes the security parameters of the MDS. Vehicles except the CHs belong to the forwarding plane. Each vehicle is equipped with an IEEE 802.11p interface to communicate with other vehicles. Each vehicle is also equipped with an SDN controller and an SDN agent. This agent is always activated. However, the SDN controller is initially deactivated and will only be activated when the vehicle becomes a CH and deactivated again if the vehicle reverts to a CM. Each RSU is equipped with two interfaces: wired link to communicate with the neighboring RSUs, and an LTE/5G interface to communicate with the global SDN controller. We assume that the RSUs are trusted nodes and the communication links between the local SDN controllers, the vehicles, and between the three types of SDN controllers are secured.

In our proposed SDN-based MDS system, the control plane five main control functions: (i) Creation of vehicular clusters; (ii) Selection of the Watchdogs; (iii) Evaluation of the trust; (iv) Detection of Sybil attacks; and (v) Tuning of security parameters including not only the parameters and thresholds of the trust, but also the number of Watchdogs. A Watchdog monitors the neighboring vehicles and sends its reports to the local SDN-controller. The local SDN controller monitors all CMs and calculates their trust levels leveraging on their monitoring reports and the reports received from Watchdogs. Finally, The local SDN controller sends its report to the regional SDN controller. This latter monitors local SDN-controllers and calculates their trust levels. Then, the regional SDN controllers aggregate all the trust values of vehicles and send the final report to the global SDN-controller. This process is periodically executed during the evaluation period.

IV. Clustering and Watchdogs Election

A. Clustering Strategy

We assume that the road is divided into equal static segments as shown in Figure 1. The length of the segment (L) is less than the communication range of vehicles (R). We assume that the global SDN controller periodically creates the vehicular clusters, which are restrained to these segments. Indeed, at a given time t, all vehicles within a given segment are all considered members of the same cluster and the cluster head is selected according to the Selection Factor (SF), which is given by the formula (1):

$$SF_i = \alpha * Trust_i + \beta * (Ndistance_i * Nspeed_i) \quad (1)$$

$$Ndistance_i = \frac{Maxdistance - distance_i}{Maxdistance} \quad (2)$$

$$Nspeed_i = 1 - \frac{|speed_i - Avgspeed|}{Maxspeed - Minspeed} \quad (3)$$

Formula (1) selects the most honest and stable vehicle to become the CH. A vehicle *i* is stable if it is close to the center of the cluster and its speed is close to the average speed of all CMs of the same cluster. For this reason, the selection of the CH is based on two criteria: trust ($Trust_i$) and mobility.

The impact of each of these criteria is weighted by α and β (α + β = 1, α,$\beta \in [0, 1]$). The mobility is measured according to: (i) $Ndistance_i$ (calculated by the formula (2)), which is the normalized value of the distance between the vehicle and the center of the segment, and (ii) $Nspeed_i$ (calculated by the formula (3)), which is the normalized value of the difference between the vehicle's speed and the average speed of the CMs. The vehicle with the highest SF value is selected as a CH.

We assume that the cluster management (the creation and the update of clusters) is performed by the global SDN-controller.

B. Watchdogs Election

The evaluation of the trust of a vehicle is computed based on the opinions collected from his neighbor vehicles, namely watchdogs. However, it is crucial to ensure that opinions are not collected from misbehaving Watchdogs. In addition, a significant overhead could be generated if a large number of vehicles plays the role of a watchdog. For these reasons, the local SDN-controller should carefully select the Watchdogs according to their trust level and their distance to vehicles. To this end, we propose that the number of Watchdogs should be determinate by the formula (4) where z is the size of the cluster and ρ_w is the density of the watchdogs. The density of watchdogs is determined as functions of the presence the misbehaving vehicles.

$$nbr_{watchdogs} = \frac{z}{\rho_w} \qquad (4)$$

After the calculation of the number of Watchdogs, we deploy them according to the number of road lanes and the distribution of vehicles on the considered segment. The segment is thus divided into zones whose number (nbr_{zone}) is calculated using the following formula:

$$(5) \qquad \begin{cases} nbr_{zone} = nbr_{lanes} & if (nbr_{lanes})\%2 = 0 \\ nbr_{zone} = nbr_{lanes} + 1 & else \end{cases}$$

with nbr_{lanes} denotes the number of lanes. The number of Watchdogs that can be deployed at each zone ($nbr_{watchdogs/zone}$) can be calculated using the formula (6). The vehicles with high trust values in each zone are selected as Watchdogs.

$$nbr_{watchdogs/zone} = \frac{nbr_{vehicle/zone}}{\rho_w} \qquad (6)$$

V. Trust computation and Sybil attack resistance

A. Trust computation

The trust of Vehicles (CMs) is evaluated by the local SDN controller based on their actions. The trust level of a vehicle is divided into two parts: the direct and the indirect trust. The direct trust is calculated based on the interactions between the vehicle and the local SDN controller (CH), whereas indirect trust is calculated based on interactions of the vehicle and the Watchdogs. The trust level of a given vehicle *v* ($Trust_v$) is thus given by the following formula:

$$Trust_v = (1 - \frac{1}{(\gamma * I_v) + 1}) * DT_v + (\frac{1}{(\gamma * I_v) + 1}) * IT_v \quad (6)$$

Where I_v is the number of direct interactions between the vehicle and the local SDN controller. Since the direct trust is more important in the calculation of the trust, we assign more weight to it ($1 - \frac{1}{(\gamma * I) + 1}$), which rapidly increases with the number of direct interactions (I_v). However, it is controlled by the parameter $\gamma \in \mathbb{R}^+$.

1) Direct Trust: Direct trust is computed based on the actions of the vehicle during its journey. An action can be either honest or misbehaving. A misbehaving action in our model is defined as a malicious action performed by misbehaving vehicles such as a message drop, false information injection, message replay and channel jamming. The impact of these misbehaving actions is different. For example, injecting false information is more harmful than replaying a message [6]. For this reason, we introduce a weight $s_j \in \{1$: Low, 2: Medium, 3: High, 4: Lethal$\}$ for each misbehaving action to reflect its impact on the safety. To this end, we denote by (A_v^h) and A_v^m, the number of honest actions performed by a vehicle *v* and the number of weighted misbehaving actions given by the formula (7), respectively. The total number of weighted actions A_v is the sum of honest actions and weighted misbehaving actions as given in the formulas (8).

$$A_v^m = \sum_{j=1}^{n} s_j * A_j \qquad (7)$$

$$A_v = A_v^h + A_v^m \qquad (8)$$

The direct trust is thus calculated using the following formula.

$$DT_v = (\frac{A_v^m}{A_v}) * (1 - \frac{1}{(\gamma * A_v^m) + 1}) \qquad (9)$$

2) Indirect Trust: The indirect trust of a vehicle *v* is the average of trust levels calculated by the watchdogs who were interacting with them. The indirect trust of a vehicle is thus calculated using the formulas (10), where nbr_w is the number of Watchdogs who have interacted with *v* and $DT_v^{w_k}$ is the direct trust of a vehicle *v* calculated by a Watchdog *k* using the formula (9).

$$IT_v = \frac{1}{nbr_w} \sum_{k=1}^{nbr_w} DT_v^{w_k} \qquad (10)$$

B. Trust computation of local SDN-controllers

Local SDN controllers (CHs) are also evaluated by the regional SDN controller based on their actions. The trust level of a local SDN controller (LT) is thus the average of direct trust levels of regional SDN controllers who have interacted with it. It is hence given by the formula (11), where nbr_{rc} is the number of regional SDN controllers and $DT_v^{rc_i}$ is a direct

trust level reported by a regional SDN controller rc_i. $DT_v^{rc_i}$ is defined by the formula (12).

$$LT = \frac{1}{nbr_{rc}} \sum_{i=1}^{nbr_{rc}} DT_{lc}^{rc_i} \qquad (11)$$

$$DT_{lc} = (\frac{A_{lc}^m}{A_{lc}}) * (1 - \frac{1}{(\gamma * A_{lc}) + 1}) \qquad (12)$$

Where A_{lc}^m is the number of weighted misbehaving actions performed by the local SDN controller, while A_{lc} is the total number of weighted actions.

C. Aggregation, privacy and Sybil attack resistance

In our MDS, the trust levels of vehicles are regularly calculated according to a fixed time period Δ, which is dynamically adjusted by the global SDN controller. During Δ, the local SDN controller (CH) calculates the direct trust levels of its cluster members and each Watchdog calculates the trust levels of its neighboring CMs. At the end of Δ, each Watchdog reports the calculated direct trust levels to the local SDN controller. As soon as these reports are received, the local SDN controller calculates the final trust level of all its CMs. These calculated trust levels (trust report) are sent to the global SDN-controller (CA) via regional SDN-controllers (RSUs). The CA thus decides the truthiness of vehicles if the trust of the vehicle is below a trust threshold $\sigma \in [0, 1]$. This threshold is dynamically adjusted by the SDN-controllers to provide high detection accuracy and to decrease the false positive.

Algorithm 1: Sybil attack detection

```
Data: Trust Report (TR)
Result: Sybil Attacker set (SA)
foreach pseudo ps_i ∈ TR do
    if ! notified (ID_v, ps_i) then
        SA ← SA ∪ ID_v;
    end
end
```

However, as vehicles frequently change their pseudonyms, different trust values associated with the same vehicle could be reported to the CA. In addition, misbehaving vehicles could use their pseudonyms as Sybils to avoid being detected. To overcome this problem, we propose that each vehicle notifies its local SDN-controller before changing its pseudonym. This notification is forwarded to the global SDN controller (CA). Each time the CA receives a trust report, it runs the Sybil attack detection algorithm as described in Algorithm 1. For each reported trust level entry, the CA checks if the used pseudonym ps_i was reported or not using its long-term identity ID_v. If a vehicle *v* changes its pseudonym without informing the CA, it is considered as a misbehaving vehicle and added to the Sybil attacker list.

VI. Performance Evaluation

We have carried out a set of simulations to evaluate the performance of our proposed MDS. These simulations are conducted using Veins Simulation Framework [9]. Table I summarizes the simulation parameters.

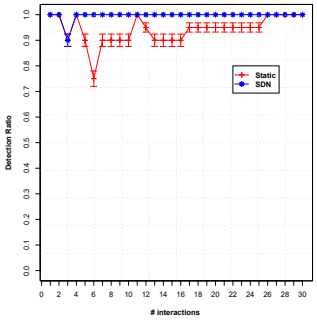
TABLE I: Simulation Parameters

Parameter	**Value**
Simulation duration	60 s
Transmission Range	500 m
The size of the cluster	{20, 30}
Ratio of misbehaving vehicles	{10%, 30%}
Number of watchdogs	{1, 2}
α,β	0.5
s_i	1

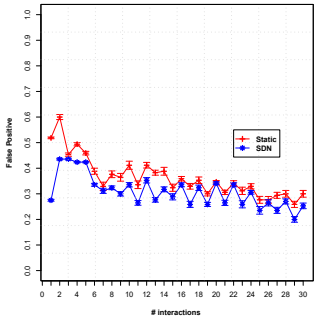
We considered the case of a freeway road. We simulated a 2-lane straight road section of 3 Km. The mobility of vehicles is generated using SUMO. As shown in Table I, we considered the case of medium clusters (20 to 30 vehicles). We also considered low (10%) and high (30%) ratio of misbehaving vehicles. The parameters α and β are fixed to 0.5, while the weight of all misbehaving actions (s_j) equals to 1.

We studied the efficiency of the proposed MDS in general, but in particular, we evaluated the merit of introducing the SDN in our proposed system. For this reason, we considered, two versions of our proposed MDS: (i) the **Static-MDS**: this is a free-SDN version, which uses a default configuration (nbr_w = 2, σ = 0.5, γ = 1) that does not change over time and do not adapt to the context of vehicles; and (ii) **SDN-MDS**: which was described in the previous sections and ensures the implementation of an adaptive MDS. In this version, the security parameters of the MDS (nbr_w, σ, and γ) are changed according to the context of vehicles. By mixing up the ratio of attackers and the the size of cluster, we came up with 4 different contexts. During the evaluation period, 30 interactions between the local SDN controllers and the Watchdogs were performed. We run simulation several times with different random seeds and calculate the average value with a 95% confidence interval.

Figures 2, 3, 4, and 5 compare the performances of **Static-MDS** and **SDN-MDS** in terms of detection ratio and false positives in each considered context. It is clear that our proposed MDS provides higher detection ratio. In addition, we can see that SDN-MDS adapts the security parameters (nbr_w, σ, and γ) according to the context to enhance the detection ratio and decrease the false positive as the number of interactions increase. As shown in Table II, the SDN controller deploys 2 Watchdogs for the cluster with 20 vehicles, because the attackers were distributed over the cluster. However, the values assigned to the parameter γ show that the SDN controller puts much consideration on the direct trust evaluation provided by the local SDN controller compared to the indirect trust

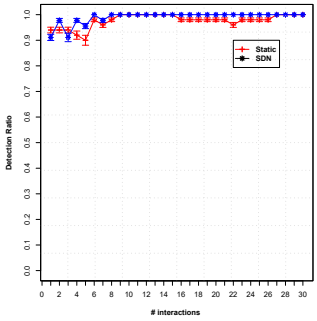


(a) Detection ratio

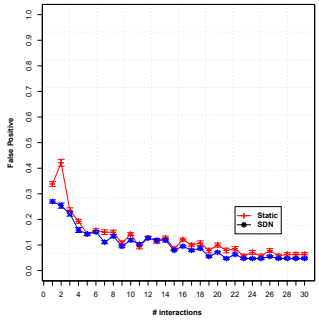


(b) False positive

Fig. 2: Context 1

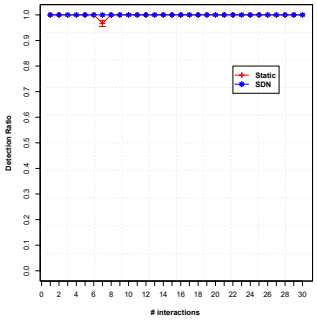


(a) Detection ratio

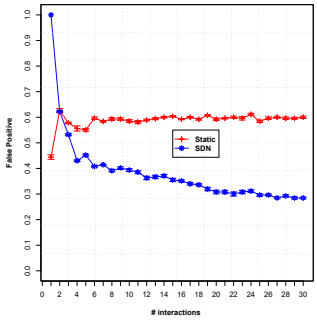


(b) False positive

Fig. 3: Context 2

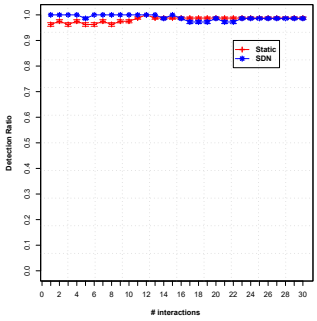


(a) Detection ratio

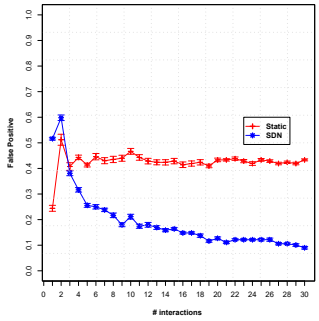


(b) False positive

Fig. 4: Context 3



(a) Detection ratio



(b) False positive

Fig. 5: Context 4

TABLE II: Context Parameters

	Size of Cluster	Ratio of misbehaving Vehicles	nbr_w	σ	γ
Context 1	20	10%	2	0.63	5
Context 2		30%	2	0.55	100
Context 3	30	10%	1	0.58	0.7
Context 4		30%	1	0.68	1

the evaluation provided by the Watchdogs as the number of attackers increases. On the other hand, for the cluster with 30 vehicles, only 1 Watchdog is deployed because the attackers are grouped only on one side of the cluster. However, the values assigned to the parameter γ show that the global SDN controller gives more importance of the indirect trust evaluation given by this Watchdog compared to the case of the cluster with 20 vehicles. Table II also shows that the trust threshold (σ) is also adapted in each considered context to provide high detection radio with low positive rate.

VII. Conclusion

The failure in detecting misbehaving nodes in a vehicular network could jeopardize the safety of users. In this paper, we have proposed an adaptive misbehavior detection system that, leveraging on the Software-Defined Networking paradigm, adjusts its security parameters according to the context of vehicles. Our system is privacy-preserving and Sybil attack resistant.

Acknowledgment

This work was supported by the H2020 5G-DRIVE project (ID: 814956).

References

[1] C. A. Kerrache, C. T. Calafate, J.-C. Cano, N. Lagraa, and P. Manzoni, "Trust management for vehicular networks: An adversary-oriented overview," *IEEE Access*, vol. 4, pp. 9293–9307, 2016.

[2] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 779–811, 2018.

[3] ETSI TR 103 415, "Intelligent transport systems (ITS); security; pre-standardization study on pseudonym change management," *ETSI standards*, 2018.

[4] H. Sedjelmaci and S. M. Senouci, "A new intrusion detection framework for vehicular networks," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 538–543.

[5] ——, "An accurate and efficient collaborative intrusion detection framework to secure vehicular networks," *Computers & Electrical Engineering*, vol. 43, pp. 33–47, 2015.

[6] F. Ahmad, V. N. Franqueira, and A. Adnane, "Team: A trust evaluation and management framework in context-enabled vehicular ad-hoc networks," *IEEE Access*, vol. 6, pp. 28 643–28 660, 2018.

[7] M. Kim, I. Jang, S. Choo, J. Koo, and S. Pack, "Collaborative security attack detection in software-defined vehicular networks," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2017, pp. 19–24.

[8] S. Garg, K. Kaur, G. Kaddoum, S. H. Ahmed, and D. N. K. Jayakody, "Sdn based secure and privacy-preserving scheme for vehicular networks: A 5g perspective," *IEEE Transactions on Vehicular Technology*, 2019.

[9] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan 2011.