Impact of Probabilistic Vehicle Estimates on Communication Reliability at Intelligent Crossroads

Daniel Markert[a], Philip Parsch[b] and Alejandro Masrur[a]
[a] Department of Computer Science, TU Chemnitz, Germany,
[b] Department of Computer Science, The University of Chicago, USA

***Abstract*—Due to the safety-critical nature of intelligent crossroads, ensuring communication reliability between vehicles and the corresponding road side unit (RSU) is of utmost importance. This requires knowledge of the maximum number of vehicles in the system to be able to assess interference. However, due to the open-ended nature of the application, i.e., vehicles can enter and leave at arbitrary points in time, it becomes difficult and inefficient to perform an analysis based on deterministic methods. For this reason, this paper instead proposes a probabilistic technique to estimate the maximum vehicle count at crossroads. To this end, we investigate how different factors like traffic protocol, vehicle density, and vehicle length influence probabilistic estimates of the maximum vehicle count. We then highlight how these estimates impact communication reliability by deriving guarantees towards packet loss in an exemplary crossroad vehicular ad-hoc network (VANET). As shown in a detailed case study and simulations using OMNeT++, pessimism and overdesign can significantly be reduced compared to deterministic approaches, while still maintaining high safety levels.**

***Index Terms*—Intelligent Transportation Systems, Vehicular Ad Hoc Networks, Vehicular and Wireless Technologies, Communication Network Reliability, Cyber-Physical Systems**

1 Introduction

Constant progress in the field of autonomous driving is making intelligent crossroads become increasingly important. Such systems aim to replace conventional traffic lights by coordinating vehicles to prevent unnecessary braking and, thus, optimize traffic flow. To this end, a road side unit (RSU) periodically transmits new speed values to approaching vehicles so that they can cross the intersection safely and efficiently. Since communication is safety-critical, a special vehicular ad-hoc network (VANET) protocol must be implemented to ensure reliable communication between RSU and vehicles. In this paper, we focus on such an intelligent intersection and the communication reliability in the corresponding VANET.

In particular, the communication reliability is directly affected by the maximum number of vehicles in the intersection — more vehicles lead to more data traffic and, therefore, increase packet loss. Since an intelligent crossroad is an open-ended cyber-physical system, i.e., participants can enter and exit the system at any time, it becomes difficult to provide any kind of guarantees. Especially deterministic approaches (e.g., assuming that the intersection is completely filled with vehicles) are not suitable here, since these lead to high pessimism and overdesign. For this reason, this work focuses on probabilistic approaches to estimate the maximum number of vehicles instead, as these better fit the random nature of such systems. More precisely, many properties of intelligent crossroads are of stochastic nature, e.g., traffic density, turn direction, vehicle types and packet loss, etc.

On the other hand, probabilistic methods cannot guarantee full safety, since there is always a small but non-zero probability of exceeding the estimated values, which violates safety. To circumvent this problem, additional fail-safe mechanisms are required, for example, switching to conventional traffic lights when the system exceeds the estimated maximum number of vehicles. As soon as the number of vehicles returns to the original one, safety is restored and the system can switch back to its intended automated operation. While such fail-safe mechanisms maintain safety, they also reduce the quality of service by disrupting normal operation. Hence, their use must be kept at a minimum, which can be achieved by selecting the appropriate probabilistic estimates.

Contributions: In this paper, based on our previous work [20], we present a probabilistic approach to estimate the maximum number of vehicles at an intelligent crossroad. To this end, we investigate the likeliness that vehicles require a certain space to cross the intersection, which is used to calculate how many vehicles fit within its range. For this we consider the following factors:

- traffic protocol, i.e., actions (e.g., turn right/left, drive through) that vehicles can perform at the given intersection,
- vehicle density, i.e., how many vehicles there are at the intersection,
- vehicle length, i.e., what types of vehicles (trucks, cars, motorcycles) are present at the intersection, and
- secondary factors, i.e., other indirect influences such as weather, time of day, day of the week, etc.

We then design a specialized VANET by incorporating probabilistic vehicle estimates and deriving guarantees on communication reliability, which we validate by extensive simulations based on OMNeT++ [30]. These guarantees are also of probabilistic nature, i.e., they hold a residual risk of not being met. Therefore, they need to be backed up by fail-safe mechanisms, as mentioned above. Finally, we discuss how to extend the presented analysis to different crossroad settings.

Structure of this work: Section 2 discusses the state of the art, while Section 3 covers the underlying models and assumptions and introduces the crossroad example used for later analysis.

In Section 4, the different factors influencing vehicle counts are discussed and results are used in Section 5 to derive our probabilistic estimates. Next, Section 6 evaluates the influence of these different factors, while Section 7 introduces the VANET protocol and discusses the impact on communication reliability. Finally, Section 8 concludes the paper.

2 Related Work

This section provides a brief overview of literature concerned with estimating vehicle numbers and their impact on communication reliability. To this end, since there are only few works that combine these two topics, we first discuss approaches that deal with estimating vehicle numbers in general and then talk about works that analyze the impact of vehicle traffic, etc. on communication reliability. Finally, we mention works combining both.

A number of approaches have been presented that estimate vehicle numbers by using fixed sensors such as cameras [38], magnetic coils [31], etc. These typically count the number of vehicles passing through a small section of a road, where the sensors are mounted, and use different methods to estimate the number of vehicles outside the monitored segments — sensor deployments are usually costly and, hence, only certain parts of the road are monitored. For example, in [38], a correlation model is proposed that uses heuristics and machine learning techniques to estimate vehicle numbers outside the monitored areas. Similarly, in [31], different filters and estimation algorithms are used with the same purpose.

Other approaches also use mobile sensors, e.g., specially equipped probe vehicles. For example, the approach in [2] estimates traffic density based on the travel times by probe vehicles (the higher the traffic density, the longer the travel times). In [23], vehicles' trajectories are used for calculations, i.e., the information of where vehicles drive to and whether they have to stop or not is an indicator for congested areas or free flowing traffic correspondingly. In summary, such monitoring approaches can provide accurate vehicle estimates, however, they are usually expensive and need to be deployed first to provide data. In contrast, our method does not require an actual deployment, but vehicle numbers can already be estimated with existing information such as intersection layout as discussed later in more detail.

Another important metric, which is commonly found in the literature and closely related to the vehicle count at a crossroad, is vehicle queue length. This describes the number of vehicles on a road that are approaching the intersection with the intent to cross. Since the queue itself can extend beyond the range of the intersection and therefore beyond the VANET itself, only the part of the queue within range of the intersection is relevant. In other words, the vehicle count at a crossroad, which we aim to estimate in this paper, is only a subset of the vehicle queues leading towards the intersection.

One such queue-based approach is presented in [15]. Here, the number of vehicles entering the intersection (obtained from sensors) is used to more accurately predict traffic and accordingly modify traffic-light plans a for higher efficiency. Similarly, [18] estimates the queue length by analyzing communication traffic instead of the traditional input/output approach, i.e., estimates are not derived from real-world data gathered by sensors. This allows them to estimate time-dependent queue lengths even when the queues exceed past the physical sensors typically used for these estimations. Another approach towards queue length is presented in [17]. Here, data from loop detectors is processed by convolutional neural networks to estimate queue lengths at traffic junctions, which is shown to increase the accuracy of estimations.

While queue-based approaches allow estimating vehicle numbers with great accuracy, they still have a number of drawbacks in the context of our work. That is, since they depend on live data, they can only output data after being deployed, but cannot predict data for settings that have not been installed yet. In addition, these focus on roads and not on intersections in particular. As a result, they do not consider intersection specific information such as layout, turn behavior, etc., which might lead to reduced accuracy.

Similarly, in [32], traffic density is estimated by using information about vehicle spacing obtained from the VANET's infrastructure. While this avoids needing extra sensors and reduces overall costs, it has the same disadvantages as the queue-based approaches mentioned before. For example, it does not include crossroad-dependent factors and does not allow for predictions or estimates in advance, but only after deployment.

To upper-bound the maximum number of vehicles at a crossroad, deterministic approaches have been used in the past [21][22]. However, these approaches do not consider the effects of crossroad protocols, vehicle lengths, travel directions etc. on the number of vehicles. On the contrary, they are based on restrictive simplifications that aim to guarantee safety at the cost of high pessimism.

Regarding communication reliability, a number of papers have been presented that analyze the effect of vehicle counts in intelligent transportation systems. However, most of them focus on improving physical models for more accurate simulations, e.g., shadowing [14], multi-path fading [28], etc., and only few account for traffic-related impact at intersections.

In [6], a coordinated routing for traffic nodes (i.e., vehicles) is proposed to reduce the overall load on safety-critical communication channels. This presents an meaningful option to distribute the overall vehicle load in an area between different intersections and their respective communication channels. However, it cannot be applied to an individual intersection with a single communication channel, as it is the case in this paper.

Several papers address the impact of vehicle count on communication reliability. For example, in [26], the impact of vehicle counts on communication reliability is analyzed in the context of platooning by predicting packet reception rate. Results show that increasing vehicle densities lead to a decrease in packet reception rate. Similarly, [37] discusses the negative impact of an increasing vehicle count on vehicle to infrastructure (V2I)-performance, focusing on the maximum data volume that can be transmitted. Likewise, [1] analyzes how the vehicle count impacts packet loss rates in hand-off algorithms, recognizing it as a major influencing factor on communication reliability.

Lastly, the approach in [13] connects high vehicle counts to safety issues caused by congested wireless channels. To mitigate this problem, the authors propose a strategy in which the RSU modifies communication parameters (such as transmission range, transmission rate and contention window size) to increase communication reliability in the case of congestion. While this is an interesting strategy to adapt to high vehicle counts, the presented version again depends on sensors such as cameras, etc. that need to be deployed first to deliver data. Moreover, like the other approaches already mentioned, they do not address the problem of deterministic pessimism or propose probabilistic models for the vehicle count.

Finally, to the best of our knowledge, probabilistic approaches for estimating the maximum number of vehicles at a crossroad in conjunction with their impact on communication reliability have been first used in [20]. However, in this approach, no turn behavior or traffic density is considered and the crossroad was restricted to a single lane only, which clearly does not reflect the real world. In this paper, our goal is to lift these restrictions and extend the work in [20] to improve the accuracy of vehicle estimates at the crossroad. To this end, we consider not only primary factors as the mentioned above, but also secondary ones such as weather and time that also influence our estimates. As a result, we are able to provide better guarantees on the VANET's reliability, which we demonstrate based on OMNeT++ simulations.

3 Background Knowledge and Definitions

As mentioned earlier, the idea behind an intelligent intersection is to replace traditional traffic lights with an RSU that coordinates vehicles to cross the intersection safely and efficiently. Fig. 1 shows such an exemplary intersection consisting of four lanes (north, east, south and west), which will be used in the following as a basis for our analysis.[1] Note that, for the sake of exposition, we assume that vehicles arrive only in the west/east lane, while the north/south lane is for exiting only. However, our analysis can easily be extended for more complex intersection types with multiple lanes as shown later.

Whenever a vehicle comes within a range R of the intersection, it enters the *arrival* zone and must first register at the RSU — in our example, we set $R = 150\,\text{m}$. To this end, the vehicle sends an identifier containing its vehicle type, mass, length, etc. as well as speed and desired direction, i.e., whether it plans to turn left, right or drive through. Once vehicles are registered, the RSU keeps tracking them and periodically broadcasting control messages containing speed values for all vehicles that are currently within range.

[1]We assume a fully connected and automated traffic in this paper. In case of mixed-traffic with conventional vehicles, clearly, infrastructure support is needed in form of sensors (like radars, lidars, etc.) and intelligent panels to communicate with drivers. On the other hand, conventional vehicles do not affect the communication reliability at the crossroad and, hence, we do not elaborate on this any further.

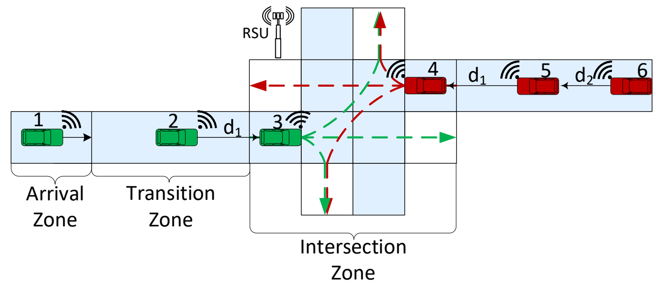


Fig. 1: An exemplary intelligent four-lane crossroad in which new vehicles arrive periodically at the west/east lane.

Upon registration, a vehicle enters the *transition* zone where it drives at the given speed and keeps a certain distance d to its front vehicle. This way, all approaching vehicles can be synchronized to pass the intersection efficiently and collision-free [4][19]. Lastly, at the *intersection* itself, vehicles can either drive through or turn left or right — this is denoted by T, L and R respectively. Note that once a vehicle leaves the intersection, it is automatically de-registered by the RSU and not regarded anymore.

Each vehicle has to keep a minimum distance d_{min} to its front vehicle, which depends on the crossroad layout and protocol used, as explained in the following section in detail.[2] To this end, we divide the intersection into sectors to create a grid-like structure, which is a common procedure for centralized crossroads [7][24][29]. The length S of each sector is a multiple of d_{min} and stands for the distance each vehicle travels within a given unit of time we call *cycle*. This allows us to design crossroad protocols independent of speed, provided that all vehicles have the same constant speed. For simplicity, we assume that S is equal to a standard vehicle's length (e.g., 5m). In addition, we also assume that the longest vehicle — a truck — is at most 10m long, i.e., it fits into $2S$.

Note that all vehicles travel at the same constant speed and, as a result, this should be chosen to allow for safe driving, in particular, turning left/right. On the other hand, we can extend this approach to vehicles that adapt their speeds depending on their directions (e.g., a turning vehicle may reduce its speed to $30km/h$, whereas vehicles driving through continue at $50km/h$), with slower vehicles introducing some delay/penalty. This is, however, beyond the scope of this paper — we plan to further investigate this in the future.

Unlike deterministic approaches, probabilistic ones cannot guarantee absolute safety, i.e., there is always a residual chance that estimates are not met. For this reason, fail-safe mechanisms must be added to avoid potential accidents, e.g., the crossroad can switch to a conventional traffic light's behavior, etc. Clearly, the system should be designed properly such that fail-safe mechanisms are not overused, but remain reserved

[2]Note that we define this distance d_{min} from the front bumper of one vehicle to the front bumper of the next vehicle and, hence, it includes the length of the leading vehicle [19].

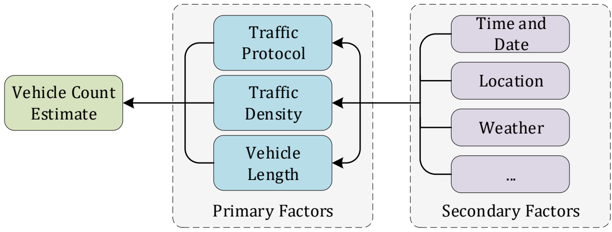


Fig. 2: Vehicle counts are influenced by different factors, which can be categorized as primary (direct influence) and secondary (indirect influence).

for rare exceptions. This way, probabilistic estimates can be used to guarantee safety during normal operation while, at the same time, reducing overdesign.

4 Influencing Factors

In order to derive estimates for the number of vehicles at a crossroad, we must first examine the various factors that influence these estimates. In general, these can be divided into two categories as displayed in Fig. 2: primary and secondary factors. Primary factors influence vehicle estimates directly and therefore have the strongest impact. These include the traffic protocol (i.e., how vehicles behave at the intersection), the traffic density (i.e., how many vehicles are present) and the vehicle length (i.e., how much space they need). The secondary factors, on the other hand, influence vehicle estimates indirectly by affecting the primary factors. They include weather, date and time, location and many others. In the following, we will first discuss the primary factors, starting by briefly introducing the traffic protocol from [19] and applying it to our crossroad example from Fig. 1. To this end, we derive a list of possible directions vehicles can take at the crossroad and assign a cycle cost and probability to each. We then add traffic density and vehicle length to improve these values and make them more precise. Finally, we discuss the influence of secondary factors such as date and time, etc. These findings are then used in the next chapter to derive probabilistic vehicle estimates.

4.1 Traffic protocol

A major influencing factor and the first step in our analysis is the selection of a suitable traffic protocol. This specifies how vehicles must behave in order to pass the intersection safely and efficiently and, as a result, fixes parameters such as throughput, inter-vehicle separation, etc. needed for the following analysis. In this paper, we select the traffic protocol from [19], which describes a centralized, RSU-based approach.

Let us first assume that all vehicles have a standard length of S. According to [19], vehicles on opposing lanes form synchronized sets that cross the intersection simultaneously. In our example from Fig. 1, this means that two vehicles (one from the west and and one from east lane) cross at a time. Since each vehicle can take a different direction (e.g., drive through, turn right or left), this leads to different combinations, each requiring a certain amount of cycles to be executed, which we refer to as *cycle cost* g in this paper. Note that the cycle cost defines the distance d to the following vehicle set, see again Fig. 1. This ensures that each set has left the intersection center before a new set arrives. This distance d allows us to draw conclusions about how many vehicles fit within the intersection of radius R. However, for an accurate estimate, other factors have be accounted for, in particular, the turn direction as discussed next.

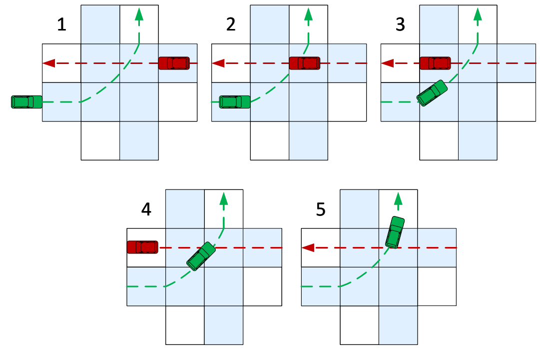


Fig. 3: A LT/TL combination in detail: Since the trajectories of both vehicles overlap, the left-turning vehicle must be delayed by one cycle to avoid a collision.

At the center of the intersection, each vehicle can either go straight or turn left or right (T, L, R). Considering that two vehicles are crossing at a time, this leads to $3^2 = 9$ different combinations: LL, LT, LR, TL, TT, TR, RL, RT, RR. However, since vehicles are synchronized and their lengths and speeds are assumed to be the same, some of these combinations only mirror. That is, with respect to cycle costs, LT equals TL, LR equals RL, and RT equals TR. As a result, there are 6 distinct combinations in total.

Fig. 3 shows the the LT/TL combination, which is the only combination with overlapping trajectories between vehicles. Here, the left-turning vehicle must be delayed by a full cycle before entering the center of the intersection to avoid colliding with the vehicle driving through. The whole crossing procedure ends after 5 cycles when the left-turning vehicle leaves the intersection — note that all vehicles must have left the intersection before a new set can enter.

For the remaining combinations (LL, RR, TT, LR, TR), the crossing vehicles do not influence each other, i.e., they don't have overlapping trajectories. This allows us to consider each drive maneuver independently, which facilitates determining its cycle cost. For a right turn, vehicles can exit the crossroad after 2 cycles, driving through requires 3 cycles and making a left turn has a cycle cost of 4. Whichever maneuver in the combination has the highest cycle cost dominates the cost of the entire combination. Table I shows all driving combinations and their corresponding cycle costs.

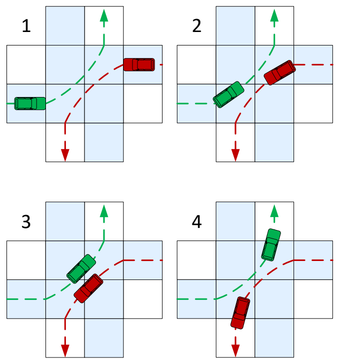


Fig. 4: During a LL combination, vehicles cross simultaneously in front of each other.

	West		
East	**R**	**T**	**L**
R	2	3	4
T	3	3	5
L	4	5	4

Table I: Driving combinations and their cycle costs

Note that in case of the LL combination, as shown in Fig. 4, we also have to consider the vehicle width. That is, depending on the sector size S, vehicles might come very close to each other. However, as long as the width of both vehicles is below a critical value, no modifications are necessary [19]. For simplicity, we disregard exceptionally wide vehicles in this paper.

Now that each driving combination has been assigned a cycle cost, let us determine their probabilities next. To this end, we first need the probability of each individual drive direction, which can be calculated using prediction algorithms such general traffic forecasting [34][35] or be derived from statistical data, e.g., by observing the traffic flow for a sufficiently long period of time. For example in [9], the westbound traffic of the city of Redmond was observed for a year and probabilities were determined to be $p_L = 0.3$, $p_R = 0.1$ and $p_T = 0.6$ for right turns, left turns and driving through respectively.

The overall probability of a combination can then be easily calculated by multiplying individual turn probabilities (since vehicles are independent of each other). For example, the probability of a driving combination given by TT is that both vehicle 1 and 2 drive through, i.e., $p_D = p_T \cdot p_T = 0.36$. In the case of mirrored combinations, e.g., LT/TL, RL/LR, etc., we have to consider both cases, e.g., $p_D = p_T \cdot p_R + p_R \cdot p_T = 0.12$.

Knowing the probability of each driving combination allows us to estimate the likeliness with which a certain cycle cost will occur and thus enables us to draw conclusions about the maximum number of vehicles. However, further factors are needed for a more accurate estimation, such as the traffic density, which is discussed next.

4.2 Traffic Density

Another primary factor with strong influence on vehicle estimates is the traffic density, i.e., how many vehicles are currently traveling towards the intersection. Previously, it was assumed that the traffic density is sufficiently high such that vehicles always cross in full sets — in our case, two at a time. However, during periods of low traffic, i.e., when less vehicles arrive at the intersection than exit, this might not be the case. As a result, we have to account for the possibility of having only a single vehicle cross at a time, as well as having no vehicle crossing at all. We denote these cases as *half set* and *empty set* respectively. In the following, we describe these cases in more detail, highlight the conditions causing them and derive their impact on cycle costs and probabilities.

In the case of only one vehicle crossing, i.e., a half set, there are no overlapping trajectories. In this case, the crossing is executed as efficiently as the drive direction allows. In the case of empty sets, we cannot use the drive direction of vehicles to determine the cycle cost, since no vehicles are crossing. Here, we propose that empty sets have a fixed cycle cost of $g = 2$, which is equal to the cycle cost of the shortest possible driving combination (RR) in the crossroad. Note that having a minimum cycle cost for empty sets is important for the following analysis, since it avoids having excessive large numbers of very small empty sets. However, choosing $g = 2$ still allows for more conservative estimates without negatively impacting results.

Considering that vehicles can be present or absent, we have additional combinations at the crossroad. More specifically, the previous 9 combinations of Table I can be extended, as shown in Table II.

	West			
East	**R**	**T**	**L**	**A**
R	2	3	4	2
T	3	3	5	3
L	4	5	4	4
A	2	3	4	2

Table II: Cycle costs for different driving combinations (A denotes an absent vehicle)

To determine what leads to full, half or empty sets and what their occurrence probabilities are, a more detailed analysis is needed.

Recall that the intersection processes vehicles in sets of two. That is, whenever a new vehicle reaches the arrival zone (see Fig. 1), the RSU tries to synchronize it with the vehicle on the opposing lane to form a set. To this end, the vehicle first speeds up or down so as to reach the intersection in a synchronized manner. Note that this also implies that vehicles in a set maintain a constant distance to preceding vehicles on their corresponding lanes. This distance is equal to the cycle cost of the preceding set. Now, if the new vehicle arrives too late, it cannot catch up anymore, and a half set is formed with only the vehicle on the opposing lane. The new vehicle is then *scheduled* to reach the intersection at a time t_{new} equal to

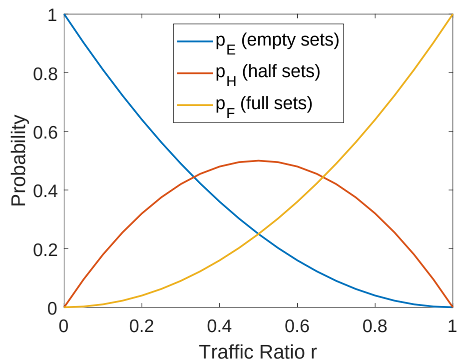


Fig. 5: Probabilities for half, empty and full sets in relation to the traffic ratio $r = \frac{r_{arr}}{r_{exit}}$. Note that $p_E + p_H + p_F = 1$.

that of the opposing vehicle plus the opposing vehicle's cycle cost. If the new vehicle is unable to reach the intersection at that time (due to, for example, a speed limit, etc.), an empty set takes place (i.e., no vehicle crosses) and the new vehicle will be scheduled at $t_{new} + 2S$. This procedure is repeated, i.e., empty sets will take place delaying the new vehicle by a multiple of $2S$, until this can reach the intersection at the scheduled time.

To obtain the probability of a full, a half or an empty set, we need to analyze the relation between the *arrival rate* and the *exit rate*. The arrival rate specifies how many vehicles arrive at the crossroad per time unit. The exit rate specifies how many vehicles can be processed and leave the intersection per time unit. Clearly, the exit rate greatly depends on the traffic protocol and intersection layout. In particular, the required inter-vehicle distance d is an indicator of how many vehicles can cross during a specified amount of time.

Let us define the *traffic ratio* r between arrival and exit rate as follows:

$$r = \frac{r_{arr}}{r_{exit}}. \quad (1)$$

If $r = 1$, the intersection is completely filled. Similarly, if $r = 0$, the intersection is empty. We hence model by $p_A = 1 - r$ the probability of a vehicle to be absent.

For the case of empty sets, both vehicles have to be absent leading to the following probability:

$$p_E = p_A^2 = (1 - r)^2 . \quad (2)$$

For vehicles to form a half set, one vehicle has to be absent while the other is present:

$$p_H = p_A \cdot (1 - p_A) + (1 - p_A) \cdot p_A = 2 \cdot r - 2 \cdot r^2 . \quad (3)$$

Finally, for a full set, both vehicles are present, resulting in:

$$p_F = (1 - p_A) \cdot (1 - p_A) = r^2 . \quad (4)$$

These probabilities are displayed in Fig. 5 in relation to the traffic ratio r. Since full, half and empty sets are mutually exclusive events, the sum of their probabilities has to be $p_E + p_H + p_F = 1$.

In order to derive values for the traffic ratio r, let us first consider the crossroad-specific exit rate r_{exit}, i.e., the maximum number of vehicles that can drive through the intersection per time unit. For this purpose, we assume that the vehicles have an average speed of $50km/h$ entering and leaving the crossroad with $R = 150m$ in $10.8s$.

Depending on the inter-vehicle separations, there can be different numbers of vehicles at the crossroad, meaning that the exit rate has a maximum and minimum value. To obtain the maximum exit rate, we assume that vehicles only perform right turns (RR), which leads to the shortest inter-vehicle separation of $2S$ (i.e., $10m$ for $S = 5m$). This allows for up to to 15 vehicles per lane or 30 for the entire crossroad. The resulting maximum exit rate is 30 vehicles every $10.8s$, or 10,000 per hour. The minimum exit rate, on the other hand, can be obtained assuming that overlength vehicles are present (see next section) and only perform LT/TL maneuvers. These lead to inter-vehicle distances of $6S$, resulting in 10 vehicles per 10.8 seconds, or 3,333 vehicles per hour. Now, in order to select a more realistic value for the exit rate, we propose using the weighted average cycle cost of $3.75S$ — this is calculated in the next chapter. This leads to an exit rate of $r_{exit} = 5{,}333 \frac{\text{veh}}{h}$.

Regarding the arrival rate, we again use the values provided in the traffic data set from [9], where traffic flow ranged from 500 to 5300 vehicles per hour. Inserting the values for arrival and exit rates in (1), we obtain a traffic ratio r of 0.1 to 0.99, and a vehicle absence probability p_A of 0.01 to 0.9.

To summarize, taking traffic density into account allows for more realistic vehicle estimates, since crossroads are not always completely full nor empty. As a result, there are different combinations with different cycle cost, i.e., not all are full sets. The corresponding probabilities of these cases can be derived based on the crossroad's exit and arrival rate at the specific crossroad.

4.3 Vehicle lengths

So far, we have assumed that all vehicles have the same length equal to one sector size S. This section removes this restriction and analyzes the impact of different vehicle lengths on our probabilistic estimates.

Fig. 6 shows the market share of various vehicle lengths based on a car sales statistics from [20]. As we can see, there are three main groups of vehicles: motorcycles, cars and trucks/buses, with cars being the most common. Consequently, it is very likely to encounter a car at the intersection, whereas it is relatively unlikely to come across motorbikes or long trucks.

If we were to follow a deterministic approach to estimate vehicle count, we would have to assume that there are only overly short vehicles at the intersection, i.e., motorbikes, even though the probability is very low. Clearly, this leads to very high pessimism and overdesign. Probabilistic estimates, however, can significantly reduce this pessimism, as discussed next.

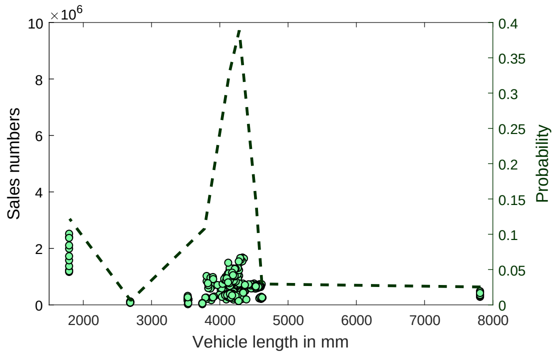


Fig. 6: Probability distribution of vehicle lengths.

	West			
East	**R**	**T**	**L**	**A**
R	3	4	5	3
T	4	4	6	4
L	5	6	5	5
A	3	4	5	2

Table III: Cycle costs for all driving combinations considering an overlength penaltiy of one S.

In order to analyze the impact of different vehicle lengths on our probabilistic estimates, we have to first study the effects on the traffic protocol from [19]. Here, vehicles longer than the sector length S — which we have set to 5 m before — are considered to be overlength (OL) vehicles. Whenever there is an overlength vehicle (which has the probability of p_{OL}), the protocol adds a number of extra cycles (called overlength penalty) to the current combination to prevent potential collisions with the subsequent vehicle set. More specifically, for every S that the OL vehicle is longer than the standard length of one S[3], one cycle is added to the cycle cost regardless of the drive direction. For example, in case of a truck with a length of $2S$, one cycle has to be added to the cycle cost independent of its drive direction. If there are multiple overlength vehicles within a set, the penalty is added only once based on the longest vehicle. Finally, we can update cycle costs to consider overlength penalty, leading to Table III.

Now that we have considered all primary factors (traffic protocol, traffic density and vehicle length), we can assign each cycle cost a probability. Using exemplary values of $p_A = 0.1$, $p_L = 0.3$, $p_T = 0.6$, $p_R = 0.1$, $p_{OL} = 0.025$ and $R = 150\,m$, this results in Table IV.

4.4 Secondary factors

Unlike primary factors, which influence vehicle count estimates directly, secondary factors have a smaller, indirect influence. More precisely, they typically alter primary factors by changing their probability distributions, for example, the probability of drive direction, etc. In the following, we will discuss some exemplary secondary factors and explain how these can be integrated in our analysis to improve the accuracy of our estimates.

[3]Note that the OL amount is rounded up to the next multiple of S for safety reasons. If a vehicle is $7m$ long (and again $S = 5m$), it will count as a $2S$ vehicle for the traffic protocol, i.e., its OL amount is one S.

Cycle cost	Vehicles in set	Probability
2	0	0.01
	1	0.017544
	2	0.007895
3	1	0.105720
	2	0.379156
4	1	0.055368
	2	0.128271
5	1	0.001368
	2	0.287291
6	2	0.007387

Table IV: Cycle costs and their probabilities depending on the number of vehicles per set. In total, the weighted average is 3.75 cycles and 1.8 vehicles per set.

A relevant secondary factor is date and time, which affects all three primary factors. For example, during rush hours in the morning, more vehicles travel downtown, while in the evening, these travel back to the suburbs. Similarly, holidays can cause some places to be crowded while others are empty. As a result, traffic density and drive directions (thereby also the traffic protocol) are affected. In addition, date and time can also change the vehicle length probabilities. For example, since several European countries have implemented truck bans on Sundays and holidays, the chance of encountering overlength vehicles is greatly reduced on those days.

There exist many further secondary factors such as weather and location. For example, traffic is usually higher during bad weather when it is snowing or raining — according to a survey in [3], 24% of respondents cited the weather as a reason for driving a car instead of riding/walking. Similarly, the location influences traffic. For example, in the previously mentioned rush hour scenario, the location (e.g., downtown or suburbs) is important and can completely change the traffic situation of the intersection. In general, secondary factors can be obtained from traffic observations or forecasting based on weather, time, etc., and the scheduling of large-scale events in the area [12][34][35].

Note that secondary factors change the probability to encounter certain cycle costs, but not the cycle costs themselves — these depend only on primary factors. Now, in order to include secondary factors in our analysis, we simply replace the individual probabilities of the primary factors with updated ones. To illustrate this, let us again consider the work in [9], where Redmond's traffic flow was monitored for a sufficiently long time. Here, for the eastbound traffic at the intersection West Lake Sammamish Parkway and Leary Way, turn probabilities were $p_L = 0.06$, $p_T = 0.43$ and $p_R = 0.51$ in the morning and changed to $p_L = 0.19$, $p_T = 0.51$ and $p_R = 0.30$ in the afternoon. The cycle costs, however, remain the same, since the time of the day only influences the probabilities of having a certain turn, but not the turn itself. Vehicle estimates can then be easily derived as shown in the next section.

5 Deriving vehicle estimates

To estimate the number of vehicles at a crossroad, we have to consider all different vehicle set combinations within the crossroad's range R. As previously discussed, each set has a certain cycle cost g which defines the number of cycles required for turns, driving through, etc. and, therefore, defines the distance to the following set of vehicles. These distances can be used to calculate how many sets physically fit into R. Note that, since lanes are synchronized, the inter-vehicle separations on each lane are the same — see again Fig. 1.

First, let us derive the upper and lower bounds for the number of vehicle sets within the crossroad's range. For the upper bound n_{max}, we assume having only sets with the shortest possible cycle cost of $g = 2$ (i.e., right turns without overlength vehicles or empty sets only). This results in:

$$n_{max} = \left\lceil \frac{R}{\min\limits_{\forall i}(g_i) \cdot S} \right\rceil = \left\lceil \frac{R}{2S} \right\rceil, \tag{5}$$

with $\min\limits_{\forall i}(g_i)$ being the minimum of all cycle costs g_i and $0 \leq i < y$, where y is the number of different cycle costs — in our example in Table IV there are $y = 5$ different cycle costs (2 to 6).

To determine the lower bound on the number of sets n_{min}, we consider that all vehicles are overlength vehicles (e.g., trucks) and have the highest cycle costs, i.e., TL/LT with $g = 6$. This results in:

$$n_{min} = \left\lceil \frac{R}{\max\limits_{\forall i}(g_i) \cdot S} \right\rceil = \left\lceil \frac{R}{6S} \right\rceil, \tag{6}$$

with $\max\limits_{\forall i}(g_i)$ being the maximum of all cycle costs g_i and again $0 \leq i < y$.

Note that each such set can have 0, 1 or 2 vehicles. As a result, the lower bound on the number of vehicles c_{min} is zero, i.e., there are only empty sets. In case of the upper bound on the number of vehicles c_{max}, we assume having the maximum number of vehicle sets n_{max} with each set being full, i.e., two vehicles crossing at a time — this represents the deterministic worst case, i.e., when the intersection is as tightly packed with vehicles as physically possible. This leads to:

$$c_{min} = 0, \tag{7}$$

$$c_{max} = 2 \cdot n_{max}. \tag{8}$$

When generating different sequences of vehicle sets, we must ensure that these fit into the crossroad's range R according to the used traffic protocol. Following [20], a *valid* sequence must meet the following condition:

$$R - 2S <= \sum_{\forall g_i}^{\max\limits_{\forall i}(g_i)} k_{g_i} \cdot g_i \cdot S < R, \tag{9}$$

where k_{g_i} represents the number of sets in the sequence with an inter-vehicle separation of $g_i \cdot S$. Note that a valid sequence must also be greater than $R - 2S$, since another set could otherwise fit into R (increasing the number of vehicles).

In order to estimate the likeliness for a certain set l to occur, we have to consider the probabilities leading to that case. More specifically, we have to consider the probability of having overlength vehicles p_{OL}, the probability of having a certain driving combination p_D (i.e., drive through, turn right/left, etc.), and the probability that the set is either a full set p_F, a half set p_H or an empty set p_E. Combining these, we can calculate the probability p_{set} of a given set to occur by:

$$p_l^{set} = p_D \cdot \{p_F \vee p_H \vee p_E\} \cdot \{p_{OL} \vee (1 - p_{OL})\} \tag{10}$$

where p_D, p_{OL}, p_F, p_H and p_E are obtained as explained before in Section 4. Note that these probabilities are associated with the corresponding set l, however, we omit index l for ease of exposition. In addition, $\{p_F \vee p_H \vee p_E\}$ denotes that either p_F or p_H or p_E should be used depending on whether the set l is a full, a half or an empty set. Similarly, $\{p_{OL} \vee (1-p_{OL})\}$ denotes that either p_{OL} or $(1-p_{OL})$ should be used depending on whether there is an overlength vehicle in the set or not.

Finally, to derive the probability of a given sequence, we multiply the probabilities of all sets that comprise the sequence:

$$p_{seq} = \prod_{\forall l} p_l^{set}. \tag{11}$$

Let us now calculate the different probabilistic estimates for our crossroad example of Fig. 1. Assuming an intersection range of $R = 150m$ and a sector size of $S = 5m$, the maximum number of sets at the crossroad ranges between 5 and 15. This leads to a vehicle count between 0 and 30, with 30 being equivalent to the deterministic worst case. Analyzing all k_{g_i} that constitute valid set sequences with $n \in [5, 15]$, we can estimate the vehicle count c with their corresponding probabilities p_c, as shown in table V.

Note that these results clearly show the pessimism of deterministic approaches. That is, the probability of having $c = 30$ is 2.92×10^{-33}, i.e., it will effectively never occur. The probabilistic estimate, on the other hand, can be chosen according to a given safety level weighing the chances that the estimate is exceeded in reality. For example, when selecting $c = 20$, the chance that there is a higher c at the crossroad is less than 4.12×10^{-7}.

5.1 *Algorithm and complexity*

In the following, we present Alg. 1 to compute all possible vehicle counts c at the intersection and their corresponding occurrence probabilities p_c.

As described previously, a valid sequence at the crossroad consists of n_{min} to n_{max} sets which fulfill (9). Therefore, after initializing probability values (see line 1 to 3), we iterate over all possible sequences out of $n \in [n_{min}, n_{max}]$ sets (see lines 4 to 5), where each of the n sets performs one of the k driving combinations shown in Table IV (requiring a given number of cycles and, hence a given inter-vehicle separation).

Veh. Count c	Probability p_c
0	3.10910E-27
1	5.04206E-23
2	2.09829E-19
3	3.79077E-16
4	2.48871E-13
5	5.75976E-11
6	1.60102E-08
7	1.76790E-06
8	7.43245E-05
9	0.001313942
10	0.010975936
11	0.048099038
12	0.115049521
13	0.159819801
14	0.156309103
16	0.171482817
17	0.154131984
18	0.064850191
19	0.000546925
20	0.000133040
21	3.71764E-07
22	4.05141E-08
23	1.06723E-11
24	7.06784E-13
25	3.83482E-17
26	1.81339E-18
27	2.12428E-23
28	7.80047E-25
29	9.73839E-32
30	2.92152E-33

Table V: Resulting maximum number of vehicles with their corresponding probabilities p_c for $p_A = 0.1$, $p_L = 0.3$, $p_T = 0.6$, $p_R = 0.1$, $R = 150m$ and $S = 5m$.

Algorithm 1: Calculating vehicle count estimates

Result: Vehicle estimates c with corresponding probabilities p_c as in Table V

1 **for** c = 0 *to* c_{max} **do**
2     $p_c(c) = 0$
3 **end**
4 **for** n = n_{min} *to* n_{max} **do**
5   **forall** *different sequences with n sets* **do**
6   **if** *sequence is valid as per* (9) **then**
7   get sequence vehicle count c_{seq};
8   get sequence probability p_{seq} as per (11);
9             $p_c(c_{seq}) = p_c(c_{seq}) + p_{seq}$;
10   **else**
11   continue;
12   **end**
13   **end**
14 **end**

Each of these sequences is then tested for validity, following (9), see line 6. If it is valid, the vehicle count c_{seq} is determined in line 7 by summing up the numbers of all empty, half and full sets, and the occurrence probability p_{seq} is calculated in line 8. Finally, in line 9, the occurrence probability p_{seq} is added to the total probability $p_c(c_{seq})$ for that specific vehicle count c_{seq}.

If the current sequence is not valid, no action is taken and we move to the next sequence (lines 10 and 11). Once all possible sequences have been processed, the values of p_c correspond to the occurrence probabilities of the different vehicle counts, as displayed in Table V.

Regarding the complexity of the algorithm, iterating through all possible sequences out of $n \in [n_{min}, n_{max}]$ sets, each of which being one out of $k = 10$ driving combinations (see Table IV), leads to the total amount of n_{max}^k sequences. As a result, Alg. 1 shows exponential complexity of $\mathcal{O}(n_{max}^k)$. Since the algorithm has exponential complexity that depends on the number of different crossing combinations k, which in turn depends on the complexity of the crossroad itself, having a more complex crossroad can greatly increase the number of combinations k and therefore the runtime of the algorithm. In our test system with an Intel Core i7-8086k at 4.00 GHz and with 16 GB RAM, the algorithm had a runtime of approximately 180 to 200 seconds — more experiments can be found in Section 6.4. Since this algorithm runs offline at design time and typically only reruns from time to time for adjustments/ maintenance, having a runtime of several minutes is acceptable.

On the other hand, the computation can be sped up at the cost of accuracy by introducing approximations. In particular, the impact of one or more primary factors can be partially or entirely disregarded. For example, in the four-way crossroad from before, dismissing the probability of absent vehicles eliminates all sets with less than 4 vehicles (i.e., 4 out of 5 possible vehicle sets with 0, 1, 2 and 3 vehicles), which reduces the number of combinations by 80%. Similarly, at the cost of a lesser throughput, we can disregard overlength vehicles, which reduces the overall number of combinations by 50%, i.e., there will be only one possible length instead of two. Finally, we can disregard slow maneuvers in the vehicle count estimation and only considering those with a high throughput like right turns, which leads to more vehicles at the crossroad. Note that these approximations do not impact the behavior of the crossroad or the crossroad protocol, but only the vehicle count estimation, which becomes more pessimistic.

5.2 Generalization

All modifications/extensions either affect primary or secondary factors. Changes affecting primary factors could be switching to a different traffic protocol, i.e., how vehicles behave at the intersection, considering another intersection layout, or adding new vehicle types such as trains. Changes affecting secondary factors could be to account for date and time, weather and location. Now in order to illustrate the process of incorporating such changes to our theory, let us first revisit the different steps to derive vehicle estimates.

The first step is to assign a cycle cost to each action that the traffic protocol allows at the given intersection. For example, in our crossroad example of Fig. 1, each vehicle can either turn left, right or drive through. This results in $3^2 = 9$ combinations in total with each having a specific cycle cost. Including other primary factors such as traffic density and vehicle length increases the number of driving combinations at the intersection and leads to additional cycle costs — see also Table III. The second step is to assign each cycle cost a probability leading to Table IV. For this, we use all primary factors and enhance their probabilities with secondary factors if needed. Lastly, the third step is to use Alg. 1 to calculate vehicle counts. As discussed before, this algorithm generates different combinations/sequences of all actions (such as turning etc.) and their cycle costs and sorts outs those that do not physically fit into the intersection's range R. This calculates how many sequences lead to a given vehicle count and derives its probability.

Now, whenever a change is introduced into a primary factor, both cycle costs and their probabilities change. As a result, we have to rerun the above calculation from step 1. For example, let us extend the crossroad from Fig. 1 such that vehicles can now also arrive from the north and south lanes. In that case, vehicles can still turn left/right or drive through, however, with up to 4 vehicles crossing in a set, turning combinations change to $3^4 = 81$ in total. The following steps then remain the same, i.e., we include overlength vehicles and traffic density to obtain the total number of possible cycle costs, assign them a probability, and finally use Alg. 1 to calculate vehicle counts.

Modifying a secondary factor, on the other hand, does not change the cycle costs, but only their probabilities as discussed previously in Section 4.4. As a result, we can skip step 1 and continue at step 2, where we replace the probabilities of cycle costs with updated values, e.g., new time-dependent turn rates, weather forecast data, etc. Again, the remaining steps are the same, i.e., we use Alg. 1 to calculate vehicle counts.

5.3 *Fail-safe behavior*

Using probabilistic estimates brings the residual risk that they may not hold. For example, when selecting $c = 20$ as per Table V, the chance of having a higher vehicle count is $\approx 4.12 \times 10^{-7}$. Even though this value is very small, it does not guarantee full safety. To overcome this problem, additional fail-safe mechanisms are required that detect such cases and prevent potential accidents. For example, if more vehicles are present at a crossroad than initially accounted for, communication might break. In that case, packets from or to the RSU do not arrive at the same rate anymore (see Section 7.1), which clearly compromises safety. To counteract this effect, the RSU could reduce vehicles' cruise speed, potentially forcing them to stop and wait until they can be served or it could even switch directly to a classic traffic-light operation. Similarly, vehicles should also implement a fail-safe behavior independent of the RSU such that, if communication breaks, i.e., no updates are received for a certain time, safety can still be guaranteed. For example, vehicles can automatically reduce their speed to keep a certain distance to each other and even perform an emergency braking to prevent crashes. However, since these mechanisms disrupt normal operation and, hence, jeopardize utility, the system should be designed not to overuse them.

6 Impact of crossroad characteristics

In this section, we evaluate the impact of the crossroad's characteristics (drive direction, probability of absent vehicles and overlength penalty) on the probabilistic estimates.

6.1 *Drive direction*

As discussed before in Section 4.1, different combinations of drive directions have different cycle costs and, therefore, affect the estimated maximum number of vehicles within the crossroad. More specifically, right turns lead to a higher vehicle count, since they have lower cycle costs, while left turns and through decrease the vehicle count due to the low efficiency of LL and LT combinations.

Our example from Fig. 1 uses $p_L = 0.3$, $p_T = 0.6$ and $p_R = 0.1$ from [16]. Now, if we keep $p_T = 0.6$ and shift the remaining 0.4 between p_L and p_R, the previously discussed behavior can be observed in Fig. 7a. That is, the more right turns there are, the higher the chance of having high vehicle counts. For $p_L = 0$ and $p_R = 0.4$, i.e., no left turns, it can even be observed that vehicle counts below 12 cannot occur unless vehicles are absent (i.e., having less than 12 vehicles is impossible for $p_A = 0$). In addition, for $p_L = 0.4$ and $p_R = 0$, i.e., no right turns, it is not possible to have more than 20 vehicles at the crossroad. For all other values of p_L, p_T and p_R (with $p_L > 0$ and $p_R > 0$), the range of possible vehicle counts c is within $[0, 30]$ as shown in Table V.

6.2 *Vehicle density*

The impact of variable traffic density, i.e., different probabilities p_A of vehicles being absent, is depicted in Fig. 7b. Again, $p_A = 0$ means that no vehicles are absent and each crossing set is full, whereas $p_A = 1$ results in empty sets only. As expected, the higher p_A, the more likely it is to have a lower vehicle count. On the other hand, very high vehicle counts, e.g., $c = 30$ are still possible, however, very unlikely to occur.

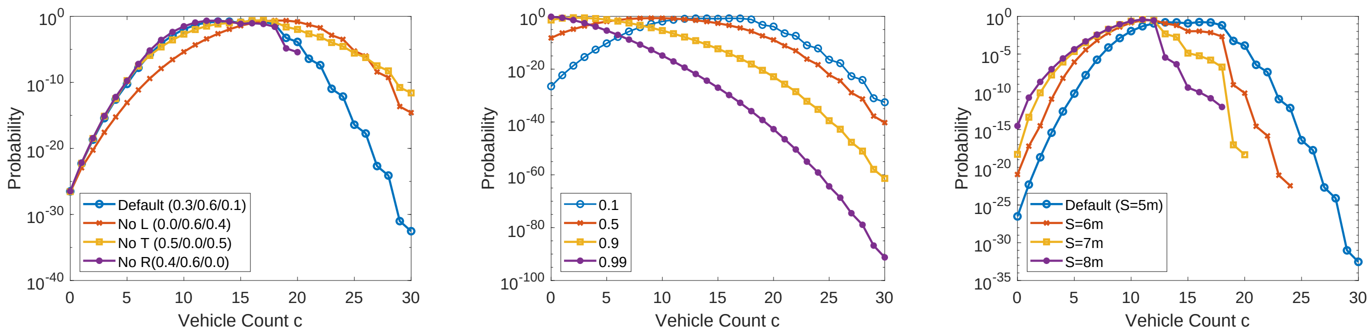
6.3 *Vehicle length*

Changes to the sector size S impact the probability p_{OL} of having overlength vehicles, since only vehicles longer than S are regarded as overlength ones. Effectively, larger sector sizes decrease p_{OL}, which leads to less extra cycles due to overlength penalty. However, on the other hand, all cycle costs are given in multiples of S and therefore increase for larger S. As a result, larger sector sizes effectively reduce the number of vehicles that can be processed at a time and are therefore not meaningful.

To illustrate the effects on our vehicle estimates c, let us now have a look at Fig. 7c. It can be seen that a larger S reduces the maximum c as expected. This complies with (5), which is directly proportional to the upper bound of c (since each set can have up to 2 vehicles). Inversely, a smaller S results in possibly larger c. Considering that larger S values reduce the throughput at the intersection, it is meaningful to set S to the length of the vehicle that occurs the most often at the intersection, i.e., $S = 5$ m.

6.4 *Algorithm performance*

The performance of the algorithm greatly depends on the sector size S. This is due to its impact in the calculation of n_{min} and n_{max} in (6) and (5). Since the algorithm has a complexity of $\mathcal{O}(n_{max}^k)$, having a reduced n_{max} reduces the runtime substantially — from approximately 180 to 200 s for $S = 5\,m$ ($n_{max} = 15$) to below 100 ms for $S = 8\,m$ ($n_{max} = 10$), see Table VI. Changing other parameters such as turn direction, vehicle absence, etc. had no impact on the runtime of the algorithm — it remained between around 180 and 200 s.



(a) Impact of drive direction (brackets denote $(p_L/p_T/p_R)$)

(b) Impact of vehicles being absent p_A

(c) Impact of sector size S

Fig. 7: Impact of drive direction probabilities, absent vehicles and sector size (i.e., overlength penalty) on our probabilistic estimate of c. If not specified otherwise, parameters are set to $p_A = 0.1$, $p_L = 0.3$, $p_T = 0.6$, $p_R = 0.1$, $R = 150m$ and $S = 5m$.

S [m]	Runtime
5	180 to 200s
6	8 to 11s
7	880 to 950ms
8	90 to 100ms

Table VI: Algorithm runtime in relation to sector size S.

7 Impact on Crossroad Communication

This section evaluates how vehicle estimates impact communication reliability at the crossroad. For this purpose we first introduce the VANET communication protocol used and then assess its performance with an OMNeT++ simulation.

7.1 VANET communication protocol

The communication scheme used for the intelligent crossroad is based on the crossroad VANET from [20], which offers high performance and provides analytical models that facilitate analysis. Note that we deliberately do not chose IEEE 802.11p-based protocols such as WAVE or ITS-G5, since these are not suited for high traffic, safety-critical applications such as intelligent intersections — this is further elaborated at the end of this section. On the other hand, observations remain valid and can be translated to other communication protocols as well.

According to the VANET protocol from [20], vehicles and the road-side unit (RSU) periodically exchange data in cycles which have the following structure as shown in Fig. 8.

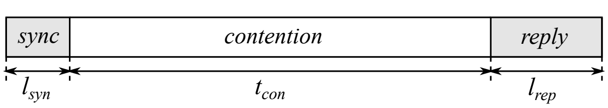


Fig. 8: Structure of a communication cycle

Each cycle starts with a sync field in which the RSU broadcasts a service advertising (SA) packet. This contains information about the intersection type, traffic load, etc. and is received by all vehicles within range R. Once a vehicle has received a SA packet, it replies by sending a request message in the contention phase. These contain vehicle data such as current speed, position, vehicle ID, etc. The RSU collects these messages, calculates new speed values for each vehicle according to the traffic protocol described in Section 3 and communicates these during the reply phase. Note that after a cycle is complete, a new cycle starts immediately. The cycle interval or length is determined by the *physical resolution*, i.e., the maximum distance that a vehicle may travel before it requires an update from the RSU. For example, if we set the resolution to 1 m and assume a speed of $50km/h$ the cycle length is set to $\frac{1\,m}{50\,km/h} = 72$ ms.

During contention phase, vehicles transmit data using the probabilistic medium access control (MAC) protocol from [25]. This is an asynchronous protocol similar to CSMA, which has been optimized to be better suited for high data traffic scenarios. That is, it offers a modified backoff scheme that better distributes transmissions within the contention phase and reduces overhead by omitting carrier sensing and acknowledgments — these are more a burden than of help during high data traffic [25]. In the following, we briefly explain its working principle and show how to configure it.

Whenever a vehicle transmit a request message, these are not only sent once, but multiple times x, whereby the time between transmissions is randomly selected in an interval $[t_{min}, t_{max}]$. Given the transmission time of a request message l_{req} and the number of vehicles c, it is possible to determine the worst-case reliability of the system, i.e., the probability that at least one out of x transmissions reaches the RSU:

$$p = 1 - \left(\frac{2(c-1)l_{req}}{t_{max} - t_{min}} \right)^x . \tag{12}$$

As can be seen in (12), the larger the packet length l_{req} or the number of transmitting vehicles c, the lower reliability becomes. Clearly, this is due to a higher channel load, which increases the chance of packet collisions. On the other hand, reliability increases for a larger time interval $t_{max} - t_{min}$ or a higher number of transmissions x — the higher x, the less likely it is to lose all packets. Before we select values for each of these parameters, let us first adapt the above equation to include probabilistic estimates of the vehicle count c and their occurrence probabilities p_c:

$$\bar{p} = \sum_{c=c_{min}}^{c_{max}} p_c \left(1 - \left(\frac{2(c-1)l_{req}}{t_{max} - t_{min}}\right)^x\right). \tag{13}$$

Here, $\bar{p}$ is the weighted reliability, i.e., the sum of all reliabilities of all different c multiplied by their occurrence probabilities p_c as per Table V. Further, c_{min} and c_{max} are the lower and upper bound of the vehicle estimate as per (7) and (8). Note that for the evaluation presented in the next section, we selected $l_{req} = 80\,\mu$s, which corresponds to 30 bytes payload at 6 Mbit/s — see also [20]. Further, in order to optimize reliability, we set $x = 3$ as per [20] and assumed $t_{min} = \frac{t_{max}}{2}$ and $t_{max} = \frac{t_{con} - l_{req}}{x}$ with t_{con} being the communication interval as shown in Fig. 8 [25].

Suitability of IEEE 802.11p: Let us briefly discuss why we do not select an 802.11p-based protocol for the intelligent intersection. The reason is that these protocols have a number of drawbacks that make them unsuitable for applications with real-time requirements and high data traffic [10]. First, they are based on carrier-sense multiple access (CSMA), which performs poorly at high traffic loads due to large overhead and an unsuited backoff scheme [11][25]. Second, since transmission time is divided in control channel (CCH) and service channel (SCH) intervals, channel efficiency is poor and cannot exceed 50 % [5]. Third, transmission intervals are only 100 ms [5], which severely lowers the intended physical resolution of $1m$ as discussed above (recall that vehicles have to send x messages within $t_{con} \approx 72ms$).

These drawbacks make 802.11p-based protocols unsuitable for applications such as intelligent crossroads. Approaches have been presented to mitigate these problems, for example, a modified MAC layer that automatically adapts CCH and SCH lengths [33]. However, the aim of this work is not the protocol itself, but the impact of vehicle estimates on communication reliability, hence, we forgo to further investigate these highly specialized solutions.

Suitability of 5G: There are many other protocols that can be used for intelligent intersections. One of the most promising is 5G, which is gaining attention due to its high performance, i.e., low latency and high transmission speeds [8]. In contrast to 802.11p, 5G uses a centralized, nonorthogonal multiple access (NOMA) approach where data transfers are parallelized and controlled by a base station [27]. This leads to better performance especially at high density traffic [8]. However, 5G requires expensive infrastructure, which is partly not available in most countries and cannot be just deployed for a crossroad. In contrast to this, 802.11p is much easier to deploy in one isolated location and currently associated with less costs than 5G

On the other hand, note that the focus of this paper is not the VANET protocol itself, but rather how to estimate vehicle counts with less pessimism compared to classic, deterministic approaches. The VANET protocol serves as a means to illustrate how performance is affected by these estimates and is independent from our theory. As a result, other protocols such as 802.11p or 5G could be chosen instead of [20]. Further analysis on this is, however, beyond the scope of the paper.

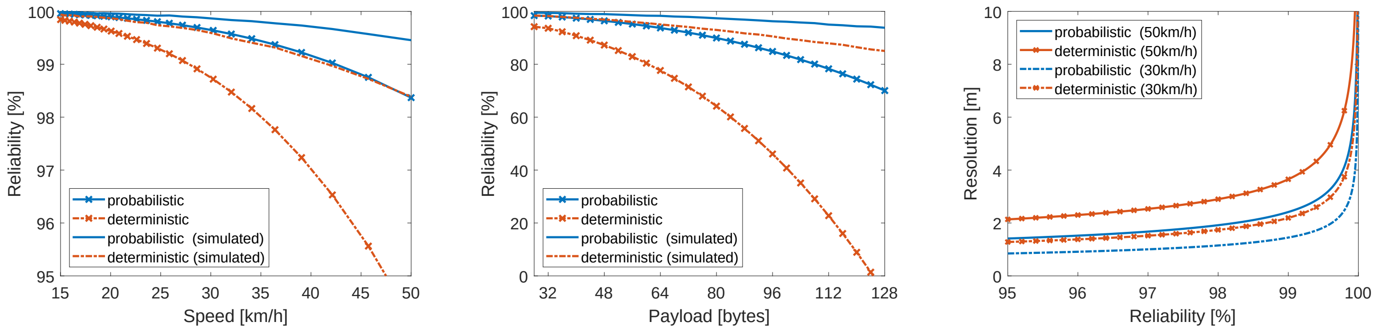
7.2 Evaluating performance impact

Next, let us examine the impact of probabilistic vehicle estimates on the communication reliability of the VANET introduced in Section 7.1. To this end, we performed simulations based on the OMNeT++ simulation framework [30], which allowed us to record statistical data of a very large number of transmissions — for each of the presented curves in Fig. 9, at least 100,000 communication cycles were simulated. We use the channel models and parameters from [36] and assume that there is no external interference present — however, this can be easily added as described in [25]. Note that we selected $c = 30$ for the deterministic approach, since this represents the worst-case number of vehicles at the intersection as per Eq. (8). For our proposed estimate, we selected $c = 20$ in the following experiments, which is a reasonable safe value, i.e., the chance of having a higher c is very small with $\approx 4.12 \times 10^{-7}$.

Fig. 9a shows the calculated (as per (13)) and the simulated (average) transmission reliability in relation to the vehicle's cruise speed at the crossroad. Recall that the vehicle speed defines the cycle length and, therefore, the communication interval t_{con} in which vehicles have to send their request messages. The higher the speed, the less time each node has to transmit its request message, resulting in a higher channel load and, therefore, less reliability. Also note that the simulated reliabilities are always higher than the calculated worst-case values in Fig. 9a.

Next, in Fig. 9b, we analyze how different payload lengths of the request message affect communication reliability. As it can be seen, reliability decreases for larger payload lengths. This is because larger payloads increase the time it takes to transmit a packet and, therefore, the chance of collision with other packets. For example, changing the payload from 30 bytes to 128 bytes, the reliability decreases from 99.45 % to 93.8 % for the probabilistic (simulated) and from 98.3 % to 85 % for the deterministic (simulated) approach. Consequently, is is meaningful to keep the payload as short as possible for the given application.

Fig. 9c shows the physical resolution (the distance a vehicle travels at a given speed) in relation to the achievable reliability p. The larger the physical resolution, the more time a vehicle has available to communicate with the RSU and, hence, the reliability increases. It can be observed from Fig. 9c that the resolution increases very slowly at first for lower p. This makes it possible to strongly increase reliability at the cost of a slightly larger resolution. For example, by increasing the resolution from 1 m to 2 m for the 50km/h case, we can increase reliability from 89 % to $\approx$ 99 % with the probabilistic



(a) Calculated communication reliability for varying vehicle speed

(b) Simulated communication reliability for varying payload length

(c) Physical resolution for varying communication reliability

Fig. 9: Comparing the proposed probabilistic with deterministic vehicle count estimates

technique. This is meaningful, if the application tolerates it. Note that the steep increase for very high p close to $100\,\%$ is due to the fact that the used MAC protocol cannot ensure full reliability (e.g., $p = 100\,\%$) — it requires increasingly more time (i.e., a higher resolution) as it approaches $100\,\%$ [25].

Further, note that a lower speed improves resolution and/or reliability, since the vehicles have again more time to transmit data — see again Fig. 9a. For example, reducing the speed from 50km/h to 30km/h and assuming $99\,\%$ reliability, the resolution improves from 2.2 m to 1.8 m. Further, it can be seen in Fig. 9c that the probabilistic technique at 50km/h still has a slightly better (lower) resolution than the deterministic one at 30km/h.

In summary, it can be seen that our probabilistic estimates for the maximum number of vehicles can greatly reduce pessimism compared to deterministic approaches. In particular, the VANET protocol benefits from it and can achieve a much higher communication reliability. Note that the safety/confidence for the chosen estimate is very high, i.e., it is very unlikely to encounter a larger vehicle count c at in the intersection ($\approx 4.12 \times 10^{-7}$ for $c = 20$). Our results show that it is more likely that the communication fails rather than c is exceeded in this example.

8 Concluding Remarks

In this paper, we proposed a method to derive probabilistic estimates for the number of vehicles at a crossroad. To this end, we analyzed the space requirements of different actions such as turn left/right, driving through, etc. and calculated their likeliness using different probabilistic factors such traffic density, traffic protocol, vehicle lengths, and others. This allows calculating how many vehicles physically fit within the range of the crossroad and, as a result, estimate how likely it is to encounter a certain number of vehicles at the intersection.

To illustrate the benefits of the proposed approach, we simulated an exemplary VANET using the OMNeT++ framework. Our results show that probabilistic estimates can greatly reduce pessimism compared to deterministic approaches, while still maintaining a high level of safety.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – MA 6480/3-1 and PA 3507/1-1.

References

[1] O. AlFarraj, A. Tolba, S. Alkhalaf, and A. AlZubi. Neighbor Predictive Adaptive Handoff Algorithm for Improving Mobility Management in VANETs. *Computer Networks*, 151:224–231, 2019.

[2] M. A. Aljamal, H. M. Abdelghaffar, and H. A. Rakha. Real-time Estimation of Vehicle Counts on Signalized Intersection Approaches Using Probe Vehicle Data. *arXiv preprint arXiv:2001.01119*, 2020.

[3] L. Aultman-Hall, D. Lane, and R. R. Lambert. Assessing Impact of Weather and Season on Pedestrian Traffic Volumes. *Transportation research record*, 2140:35–43, 2009.

[4] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige. Ballroom Intersection Protocol: Synchronous Autonomous Driving at Intersections. In *Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2015.

[5] A. Bazzi, G. Cecchini, M. Menarini, B. M. Masini, and A. Zanella. Survey and Perspectives of Vehicular Wi-Fi versus Sidelink Cellular-V2X in the 5G Era. *Future Internet*, 11:122, 2019.

[6] T. Blazek, C. F. Mecklenbräuker, C. Backfrieder, and G. Ostermayer. Improving Communication Reliability in Intelligent Transport Systems Through Cooperative Driving. In *IFIP Wireless and Mobile Networking Conference (WMNC)*, 2017.

[7] L. Chen and C. Englund. Cooperative Intersection Management: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 17:570–586, 2016.

[8] E. Dahlman, S. Parkvall, and J. Skold. *5G NR: The Next Generation Wireless Access Technology*. Academic Press, 2018.

[9] Data.Redmond.Gov - Turning Movement Counts. https://data.redmond.gov/Infrastructure/Turning-Movement-Counts/r9jv-kb54/data, February 2019.

[10] S. Eichler. Performance Evaluation of the IEEE 802.11p WAVE Communication Standard. In *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, 2007.

[11] Y. P. Fallah, C.-L. Huang, R. Sengupta, and H. Krishnan. Analysis of Information Dissemination in Vehicular Ad-Hoc Networks With Application to Cooperative Vehicle Safety Systems. *IEEE Transactions on Vehicular Technology*, 60(1):233–247, 2010.

[12] E. J. Horvitz, J. Apacible, R. Sarin, and L. Liao. Prediction, Expectation, and Surprise: Methods, Designs, and Study of a Deployed Traffic Forecasting Service. *arXiv preprint arXiv:1207.1352*, 2012.

[13] M. A. Javed, S. Zeadally, and E. B. Hamida. Data analytics for Cooperative Intelligent Transport Systems. *Vehicular communications*, 15:63–72, 2019.

[14] Y. Kikuchi and Y. Yamao. Propagation Loss Characteristic of V2V Communication for Right-Turn Accident Prevention Scenario. In *Progress in Electromagnetics Research Symposium (PIERS-Toyama)*, 2018.

[15] R. D. Komguem, R. Stanica, M. Tchuente, and F. Valois. WARIM: Wireless sensor networks architecture for a reliable intersection monitoring. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[16] J. Krumm. Where will they turn: predicting turn proportions at intersections. *Personal and Ubiquitous Computing*, 14:591–599, 2010.

[17] S. Lee, K. Xie, D. Ngoduy, and M. Keyvan-Ekbatani. An Advanced Deep Learning Approach to Real-Time Estimation of Lane-Based Queue Lengths at a Signalized Junction. *Transportation research part C: emerging technologies*, 109:117–136, 2019.

[18] H. X. Liu, X. Wu, W. Ma, and H. Hu. Real-time Queue Length Estimation of Signalized Intersections Based on RFID Data. *Transportation research part C: emerging technologies*, 17:412–427, 2009.

[19] D. Markert and A. Masrur. Space-Efficient Traffic Protocols for Intelligent Crossroads. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[20] D. Markert, P. Parsch, and A. Masrur. Using Probabilistic Estimates to Guarantee Reliability in Crossroad VANETs. In *Proceedings of the ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet)*, 2017.

[21] A. Masrur, M. Kit, T. Bures, and W. Hardt. Towards Component-Based Design of Safety-Critical Cyber-Physical Applications. In *Proceedings of the Euromicro Conference on Digital Systems Design (DSD)*, 2014.

[22] A. Masrur, M. Kit, V. Matena, T. Bures, and W. Hardt. Component-Based Design of Cyber-Physical Applications with Safety-Critical Requirements. *Microprocessors and Microsystems (MICPRO)*, 42, 2016.

[23] Y. Mei, W. Gu, E. C. Chung, F. Li, and K. Tang. A Bayesian Approach for Estimating Vehicle Queue Lengths at Signalized Intersections using Probe Vehicle Data. *Transportation Research Part C: Emerging Technologies*, 109:233–249, 2019.

[24] A. I. Morales Medina, N. van de Wouw, and H. Nijmeijer. Cooperative Intersection Control Based on Virtual Platooning. *IEEE Transactions on Intelligent Transportation Systems*, 19:1727–1740, 2018.

[25] P. Parsch and A. Masrur. Accounting for Reliability in Unacknowledged Time-Constrained WSNs. *ACM Transactions on Cyber-Physical Systems*, 3:1–28, 2019.

[26] G. Pathak, H. Li, C. B. Math, and S. H. de Groot. Modelling of Communication Reliability for Platooning Applications for Intelligent Transport System. In *Proceedings of the IEEE Vehicular Technology Conference (VTC-Fall)*, 2016.

[27] S. Redana et al. 5G PPP Architecture Working Group: View on 5G Architecture, 2019. Version 3.0.

[28] H. H. Saleh and S. T. Hasson. Improving Communication Reliability in Vehicular Networks Using Diversity Techniques. *Journal of Computational and Theoretical Nanoscience*, 16:838–844, 2019.

[29] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti. Revisiting Street Intersections Using Slot-Based Systems. *PLOS ONE*, 11:1–9, 2016.

[30] A. Varga. The OMNeT++ Discrete Event Simulation System. In *Proceedings of the European Simulation Multiconference (ESM)*, 2001.

[31] G. Vigos and M. Papageorgiou. A Simplified Estimation Scheme for the Number of Vehicles in Signalized Links. *IEEE Transactions on intelligent transportation systems*, 11:312–321, 2010.

[32] J. Wang, Y. Huang, Z. Feng, C. Jiang, H. Zhang, and V. C. Leung. Vehicular Network Based Reliable Traffic Density Estimation. *IEEE Transactions on Vehicular Technology*, 67:6424–6437, 2018.

[33] K. Xiong, X. Chen, L. Rao, X. Liu, and Y. Yao. Solving the Performance Puzzle of DSRC Multi-Channel Operations. In *IEEE International Conference on Communications (ICC)*, 2015.

[34] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with Knowledge from the Physical World. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2011.

[35] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-Drive: Driving Directions Based on Taxi Trajectories. In *Proceedings of the International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, 2010.

[36] R. Zerod. 5.9 GHz V2X Modem Performance Challenges With Vehicle Integration. In *Proceedings of the Intelligent Transportation Systems (ITS) World Congress*, 2014.

[37] J. Zhou, D. Tian, Y. Wang, Z. Sheng, X. Duan, and V. C. Leung. Reliability-Optimal Cooperative Communication and Computing in Connected Vehicle Systems. *IEEE Transactions on Mobile Computing*, 19:1216–1232, 2020.

[38] P. Zhuang, Y. Shang, and B. Hua. Statistical Methods to Estimate Vehicle Count Using Traffic Cameras. *Multidimensional Systems and Signal Processing*, 20:121–133, 2009.