On the Design of Synchronous Traffic Protocols for Intelligent Intersections

Dissertation
Submitted in Fulfillment of the
Requirements for the Academic Degree

Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

M.Sc. Daniel Markert
born 1989 in Karl-Marx-Stadt (now Chemnitz), Germany

Dept. of Computer Science
Computer Architectures and -Systems
Chemnitz University of Technology

Date of submission: June 20th 2025
Date of defense: January 12th 2026
Year of publication: 2026

Reviewers:
Prof. Dr. Alejandro Masrur
Jun.-Prof. Dr. Stefan Reitmann

Cite links:
(URL) `https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa2-1015238`
(DOI) `https://doi.org/10.60687/2026-0021`

Acknowledgments

This acknowledgement is dedicated to all those whose input, guidance and support have been invaluable during the development of this work.

To Professor Alejandro Masrur — your unfaltering persistence in encouraging me forward on this doctoral journey has been instrumental in its completion. I am deeply grateful for your unwavering support, mentorship, guidance, and friendship. Your example inspires me to strive toward the same spirit of excellence and compassion.

To my colleagues at CAS, over the years and now — our superb mix of long discussions, mutual support and appropriate banter has contributed immensely to keeping me sane and motivated. Sharing this quest with you has been an absolute pleasure and I look forward to whatever comes next. You haven't heard the last of me.

To my parents — you made sure to provide me early on with all the tools I use every day and without which this entire path would not have been possible. I am where I am today because of your sacrifices, dedication, and support, and I will be forever grateful.

To my family — by blood, by choice, and by circumstance — through every high and low, you have been a steady anchor and a safe place. I feel immensely fortunate to have you in my life.

To my friends (you know who you are) — even during the most difficult stretches of this ride, the sheer level of support, love and cohesion I have received from you is immeasurable. You made the hard times easier and the good times even better. I could not wish for more exquisite company.

And finally, to my wife Linda — this is as much yours as it is mine. Thank you for putting up with my endless ramblings, formatting battles and existential crises. I could not have done any of this without your love, unconditional support, encouragement and patience. You are the method to my madness, and I love you.

Thank you all.

Abstract

Intelligent intersections aim to replace conventional traffic lights with traffic protocols that dynamically schedule connected and automated vehicles (CAVs) in all directions, as opposed to greatly static traffic lights that are currently in use. Traffic protocols can be synchronous or asynchronous in nature, with synchronous such traffic protocols enforcing efficient crossing patterns to achieve a higher throughput.

However, due to the open-ended nature of realistic intersections, current synchronous traffic protocols are generally confined to specific types of traffic or infrastructure, which greatly restricts their applicability in the real world. To address this issue, this thesis proposes several design concepts to increase the real-world applicability of synchronous traffic protocols to a wider range of settings. More specifically, three points are covered.

First, deterministic methods fail to provide meaningful estimates of the maximum number of vehicles at the intersection, which is paramount to assess communication reliability and, in the end, guarantee safety. In contrast, probabilistic estimates can greatly reduce pessimism and overdesign compared to deterministic approaches while still retaining safety. These and other benefits or the proposed approach are illustrated by means of a detailed case study and simulations using OMNeT++.

Second, to guarantee safety, traffic protocols must enforce sufficiently large gaps between vehicles on different lanes, taking their dimensions into account. In particular, existing such protocols are designed for the longest possible vehicle resulting in *space-hungry* intersections, that require modifications in the infrastructure (in particular, broader roads/lanes). Moreover, these do not allow for vehicles that are exceptionally longer than the ones considered at design time (e.g., extra long trucks, or buses, etc.). In this thesis, to overcome this limitation, all proposed approaches handle overlength vehicles as exceptions to compensate for their low probability of occurrence, relaxing space requirements on the intersections. This is implemented in a single-crossing traffic protocol called SV-LTR (Single-Vehicle LTR) and further extended by a two-speed scheme to account for realistic driving and turning behavior and augmented with a platooning mode called PB-LTR (Platooning-Based LTR), where vehicles on opposing lanes cross with reduced inter-vehicle distances while considering the delay to perpendicular lanes.

Finally, FleXS-TP (Flexible Synchronous Traffic Protocol) dynamically forms crossing patterns without requiring specific vehicle arrival orders. By maintaining a synchronous core while scheduling vehicles individually, FleXS-TP combines the efficiency of synchronous protocols under well-behaved traffic with the flexibility of asynchronous approaches under randomized traffic, achieving substantial throughput in both scenarios, as demonstrated by realistic SUMo simulations.

These results illustrate the proposed techniques' effectiveness in enhancing safety, throughput, and space efficiency, while broadening the applicability of synchronous traffic protocols to diverse and realistic traffic conditions.

Contents

List of Figures

List of Tables

List of Acronyms

AIM	Autonomous Intersection Management (Traffic Protocol)
BRIP	Ballroom Intersection Protocol
CAV	Connected Autonomous Vehicle
CPS	Cyber-Physical System
CSIP	Configurable Synchronous Intersection Protocol
CSMA	Carrier Sense Multiple Access
DSIP	Distributed Synchronous Intersection Protocol
FCFS	First Come, First Served
FleXS-TP	Flexible Synchronous Traffic Protocol
LTR	Left, Through, Right (Traffic Protocol)
MAC	Medium Access Control
OL	Overlength
OMNeT++	Objective Modular Network Testbed in C++
PB-LTR	Platooning-Based Left, Through, Right (Traffic Protocol)
PHY	Physical Layer (of a Communication Protocol)
PPDU	Protocol Packet Data Unit
RSU	Roadside Unit
STIP	Spatio-Temporal Intersection Protocol
SUMo	Simulation of Urban Mobility
SV-LTR	Single-Vehicle Left, Through, Right (Traffic Protocol)
TDMA	Time Division Multiple Access
VANET	Vehicular Ad-Hoc Network
V2I	Vehicle to Infrastructure (Communication)
V2V	Vehicle to Vehicle (Communication)

Chapter 1.

Introduction

There is an increasing number of vehicles on roads and highways, which makes it necessary to control traffic flow in a more efficient manner. With intersections being a main source of congestion in a road network [87], there is an ongoing trend to improve their traffic efficiency. This includes features such as broadcasting the so-called time-to-green of traffic lights [90] or prioritizing emergency vehicles [34] or public transportation [57]. In the future, with connected and automated vehicles (CAVs) becoming increasingly widespread, conventional traffic lights are expected to further evolve into *intelligent intersections* that dynamically coordinate CAVs in all possible directions with the aim of achieving a higher throughput, by performing dynamic adaptations depending on traffic.

To guarantee safety, intelligent intersections rely on traffic protocols that *schedule* vehicles into collision-free *crossing patterns*. Traffic protocols can be classified as either asynchronous [26] [36] or synchronous [11] [51] [52], depending on their scheduling strategy.[1]

However, applying these concepts to real-world intersections poses several challenges as no two intersections are the same. In particular, the existing infrastructure as well as traffic volume and composition can greatly vary, not only between different intersections, but also at the same intersection throughout the day [80].

This thesis is concerned with the challenges of implementing traffic protocols in practice as detailed next.

1.1. Scope and Motivation

The following aspects of intelligent intersections are covered:

1. communication reliability and assumptions made when assessing it,
2. space efficiency and infrastructure-depending applicability of traffic protocols,
3. balancing fairness and efficiency via platooning, and
4. bridging the gap between idealistic, well-behaved and realistic, random traffic.

As mentioned before, traffic protocols can be of synchronous and asynchronous nature. Asynchronous traffic protocols schedule vehicles in a reactive manner as they arrive, while avoiding collisions. As such, they are completely insensitive to the order in which vehicles enter the intersection. However, if two or more vehicles have conflicting trajectories, they will potentially have to stop and wait for their turn. Note that this is similar to the behavior by

[1] Note how this is not necessarily indicative of their implementation, which can be either centralized and decentralized/distributed.

traffic lights, but with the right of way being decided on individual vehicles dynamically — independent of their driving directions and/or lanes on which they arrive.

Synchronous traffic protocols, on the other hand, actively coordinate vehicles' arrival times at the intersection border following predetermined patterns. That is, vehicles are sped up or slowed down — observing speed limits and avoiding unnecessary stopping — such that they reach the intersection border with a desired timing. This allows conflicts to be solved ahead of time (i.e., before reaching the intersection border), in the end, resulting in higher throughput.

Further, synchronous approaches are typically based on a roadside unit (RSU), which coordinates vehicles as they enter the intersection by assigning them speeds so as to synchronize the points in time at which they arrive at and cross the intersection. This requires periodically collecting data from vehicles and computing new speed values to guarantee a collision-free crossing and, at the same time, avoid unnecessarily braking and accelerating vehicles at the intersection with the aim of minimizing waiting time and energy consumption.

Clearly, since communication is safety-critical, these computations need to be performed in real time, for which it is paramount to guarantee a reliable communication between vehicles and the RSU. To this end, special vehicular ad-hoc network (VANET) protocol must be implemented to ensure reliable communication between the RSU and vehicles. In this case, note that reliability strongly depends on the number of vehicles at the intersection. In particular, as the number of vehicles increases, interference rises and, as consequence, VANET reliability decreases.[2]

Since an intelligent intersection 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 overly short vehicles) are not suitable here, since these lead to high pessimism and overdesign. Alleviating this pessimism by making realistic assumptions about vehicle lengths and traffic composition, which are variables of rather stochastic nature, is the **first motivation** of this thesis.

Similarly, due to the coordinated crossing under synchronous traffic protocols, longer vehicles impact the inter-vehicle distances as separation between two vehicles traveling from north to south should be sufficiently large for a vehicle to cross from east to west in between them, if interweaving patterns are implemented. Existing approaches, such as the Ballroom Intersection Protocol (BRIP) [11], are parameterized by the longest vehicle allowed at the intersection, which in turn leads to excessive size requirements towards the intersection infrastructure, limiting applicability to real, existing intersections. Overcoming this limitation is the **second motivation** of this thesis.

A key component of synchronous strategies to achieve high efficiency in terms of vehicle throughput is to have vehicles cross in all directions simultaneously. However, increasing the inter-vehicle distance to allow for this type of synchronization can offset the increase in throughput. Here, having vehicles cross on opposing, conflict-free lanes only can drastically reduce the required inter-vehicle distance, effectively following a platooning strategy where a platoon leader determines the behavior of the entire platoon, with following vehicles mimicking their

[2]Note that synchronizing access to the communication channel to avoid interference between vehicles, e.g., using TDMA (Time Division Multiple Access), yields a huge overhead due to the constantly changing operation conditions and is not best suited for this context.

behavior. Investigating the performance and applicability of such a technique in the context of intelligent intersections is the **third motivation** of this thesis.

However, this circles back to a recurring theme — while synchronous traffic protocols are highly efficient under idealistic conditions, realistic traffic does not necessarily match these predetermined patterns, leading to a degraded performance of synchronous traffic protocols in realistic settings. Specifically, as such approaches group vehicles into *sets*, which then cross in synchrony, having randomized, irregular traffic can lead to these sets not being filled entirely.

In general, synchronous approaches outperform asynchronous techniques under *well-behaved* traffic, in which crossing patterns can effortlessly be enforced, e.g., all vehicles drive through, or vehicles driving through are always followed by vehicles turning right/left, etc.[3] However, under realistic, *random* traffic, this situation reverses and asynchronous approaches become more suitable instead since synchronous approaches struggle to adapt. Overcoming this limitation of synchronous traffic protocols by increasing their flexibility towards random traffic while still retaining the efficiency under well-behaved traffic is the **fourth motivation** of this thesis.

Following these motivations, the following contributions are made in the context of this thesis.

1.2. Contributions

As mentioned before, this thesis is concerned with designing intelligent intersections. In particular, synchronous traffic protocols have been shown to achieve a better throughput under *well-behaved*, i.e., ordered traffic. However, asynchronous protocols can better deal with random traffic. As a result, this work focuses on bridging the gap between these two paradigms.

Specifically, existing traffic protocols — both synchronous and asynchronous —have limited real-world applicability with respect to communication reliability, traffic composition (well-behaved vs. random), and infrastructure requirements. To address this, several techniques are proposed to broaden the applicability of these protocols.

Namely, this thesis proposes both a probabilistic modeling of communication reliability to reduce deterministic pessimism, as well as the following traffic protocols: i) Single-Vehicle Left, Through, Right (SV-LTR) to increase space efficiency of traffic protocols, ii) Platooning-Based Left, Through, Right (PB-LTR) to extend intelligent intersections via platooning and iii) Flexible Synchronous Traffic Protocol (FleXS-TP) to increase flexibility towards traffic composition. These techniques were validated by extensive simulations using OMNeT++ [82] for communication reliability and SUMo [46] for the traffic protocols.

Detailed explanations and clarifications of each contribution are provided next.

[3] The idea of enforcing crossing patterns may appear unrealistic at a first glance. However, current intersections worldwide already require drivers to select lanes according to their intended directions, typically having separated drive-through/turn-right and turn-left lanes. These already enforce crossing patterns, but in much less dynamic fashion, resulting in less throughput.

Reducing Deterministic Pessimism for Communication Reliability

Due to the safety-critical nature of intelligent intersections, ensuring communication reliability between vehicles and the corresponding road side unit is of utmost importance. To assess potential interference, this requires knowledge of the maximum number of vehicles in the system. 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. This thesis introduces a probabilistic approach to estimating the maximum number of vehicles at intersections, accounting for factors such as traffic protocol, vehicle density, and vehicle length. These estimates are then used to derive probabilistic guarantees for communication reliability by analyzing packet loss in an exemplary VANET scenario at an intersection.

Specifically, using a realistic probabilistic vehicle length distribution can lead to a stepwise, probabilistic worst-case estimation instead of the deterministic worst-case assumption, where the longest possible vehicle is assumed to be the default case with no regard for its actual occurrence rate. This greatly reduces deterministic pessimism and overdesign.

Infrastructure-Agnostic Design

Existing synchronous approaches use the length of the longest possible vehicle at the intersection to constrain its inter-vehicle distances, assuming mostly homogeneous traffic. However, for realistic, heterogeneous traffic where exceptionally long vehicles are rare but possible, this leads to unnecessarily strict requirements for the intersection infrastructure.

Instead, treating extraordinarily large vehicles as exceptions rather than the norm by including overlength as a measure to handle vehicles exceeding a specific length can facilitate the *uncoupling* of traffic protocol parameters from the actual traffic. On the contrary, the existing infrastructure can parameterize the traffic protocol with its dimensions, such as lane width, etc. This allows for an *infrastructure-agnostic* design of traffic protocols, which only need to be parameterized/configured for a physical dimension of a given intersections, eliminating the need for big infrastructural modifications. This concept of treating longer vehicles as exceptions is implemented by all traffic protocols proposed in this thesis — SV-LTR, PB-LTR and FleXS-TP.

Platooning for Well-Behaved Traffic

For well-behaved traffic, grouping vehicles performing the same maneuvers into tightly packed platoons allows for shorter inter-vehicle distances and reduces delays caused by transitions between different intersection maneuvers. To ensure all vehicles eventually cross the intersection, a maximum blocking time regularly enforces maneuver changes, guaranteeing fairness across all directions. Under well-behaved traffic, PB-LTR can further increase throughput.

Increasing Flexibility towards Traffic Composition

The proposed protocol FleXS-TP combines the high flexibility of asynchronous traffic protocols, which makes them more suitable for random traffic, and the high efficiency of synchronous traffic protocols under well-behaved traffic. This is achieved by synchronizing individual vehicles with respect to each other rather than at the intersection border through the use of individual blocking patterns. This way, groups of vehicle crossing simultaneously or *vehicle*

sets are not formed at, but more dynamically on the way to the intersection border. This allows for situational synchronicity, if the order of incoming vehicles allows for it.

By maintaining a synchronous strategy at heart, but scheduling vehicles on a vehicle-by-vehicle bases while still allowing for situational synchronicity, FleXS-TP can harness both the benefits of synchronous traffic protocols for well-behaved traffic, as well as the flexibility of asynchronous traffic protocols under randomized traffic, achieving substantial throughput in both settings.

Real-World Applicability

Further, in contrast to most protocols from the literature (synchronous and asynchronous alike), the proposed approaches consider that vehicles cross at two different speeds, depending on whether they drive through or perform a turn maneuver. This way, it benefits from a higher drive-through speed and slower (i.e., more comfortable) turn speeds. Other protocols either assume high-speed turns or require drive-through vehicles to considerably slow down for crossing.

Simulation Efforts

This thesis validates the proposed techniques using multiple simulation tools.

Considerations towards communication reliability were simulated using OMNeT++ [82], taking different vehicle speeds and payload lengths into account and achieving a resulting physical resolution (i.e., the distance a vehicle travels between two communication events) for these reliabilities.

All novel traffic protocols proposed in this thesis and their performance with regards to existing approaches were validated via extensive simulations of a realistic intersection based on SUMO [46], a well-accepted tool for traffic simulation, which can be interfaced with through a Python-based API called the Traffic Control Interface (TraCI).

1.3. Thesis Structure

The remainder of this thesis is structured as follows.

Chapter 2 is concerned with related work, discussing existing approaches from the literature. Specifically, both synchronous and asynchronous approaches are discussed in detail. Additionally, more adjacent but relevant existing approaches are discussed, covering platooning, machine learning and game theory approaches.

Chapter 3 introduces the fundamentals of what intelligent intersections and traffic protocols are, also covering models and assumptions used throughout this thesis. It details several mechanisms which are relevant for the remaining chapters such as i) sector, ii) the concept of overlength and iii) blocking patterns.

The individual contributions of this thesis are described in Chapters 4 to 7. More specifically, Chapter 4 discusses the notion of probabilistically modeling physical parameters.

Chapter 5 is concerned with space efficiency in traffic protocols and introduces SV-LTR, a space-efficient single-vehicle traffic protocol.

In Chapter 6, the previous protocol is then extended to a platooning-based approach called PB-LTR, focusing on efficiency and fairness under well-behaved traffic.

Then, Chapter 7 aims to combine the advantages of the previous contributions into a flexible synchronous traffic protocol called FleXS-TP, bridging the gap between synchronous and asynchronous traffic protocols.

Chapter 8 includes details for the different simulations used for validation and visualization, including configuration, scenarios and algorithms.

Chapter 9 evaluates the impact of each individual contribution, also discussing the performance of the introduced traffic protocols SV-LTR, PB-LTR and FleXS-TP in comparison to other approaches from the literature, against each other and with regards to conventional traffic lights.

Finally, Chapter 10 concludes this thesis by summarizing findings, advantages and disadvantages of the concepts introduced here. This chapter also provides an outlook toward possible future work and extensions.

Chapter 2.

Related Work

The need for fully automated road traffic to relieve congestion and prevent traffic jams has been acknowledged as early as 1997 [1]. Since then, many approaches were presented to solve typical problems or to improve overall effectiveness of roads, intersections, etc. In [63], for example, route guidance and driver information systems were proposed to manage and improve traffic flow. This also included several strategies for intersection control. However, the described method from [63] relies on traffic lights managing themselves with no connection to the vehicles and, hence, cannot dynamically adapt to changes in the traffic. To achieve this, traffic protocols are needed.

This chapter provides an overview of existing literature concerning traffic protocols. To this end, these existing approaches are grouped by how vehicles are scheduled — either synchronously, by enforcing a predetermined synchronized arrival pattern, or asynchronously, by scheduling vehicles as they arrive naturally. Note how this is not indicative of their implementation, which can be either centralized and decentralized/distributed.

Two specific traffic protocols, namely the *Ballroom Intersection Protocol (BRIP)* and the *Autonomous Intersection Management (AIM)*, and their respective iterations are discussed in detail as the most prominent representatives of each group.

Additionally, adjacent approaches for vehicle management are discussed, specifically regarding platooning, machine learning and game theory.

2.1. Asynchronous Arrival Pattern

One of the earliest traffic protocols/approaches for intelligent intersections was proposed with *Autonomous Intersection Management (AIM)* [25] [26] [27]. AIM is a query-based, centralized intersection manager using a multi-agent approach with a *First Come, First Served (FCFS)* reservation strategy to schedule vehicles through an intersection in an asynchronous fashion. To that goal AIM divides the intersection into a grid of $n \times n$ tiles, where n is termed *granularity* [27]. When approaching the intersection, a vehicle requests its intended trajectory, which translates into a sequence of tiles that need to be traversed (at specific points in time). This is then only granted by AIM, if the requested trajectory does not contain any tile, which has been already reserved for another vehicle (at the same point in time). To manage the reservations, AIM relies on *Vehicle-to-Infrastructure (V2I)* communication with a central manager, making it a centralized protocol.

Extensions to AIM were presented in [36] for multi-intersection settings, as well as for semi-autonomous vehicles in [10] and [78]. Further, in [24], robustness was increased by adding a vision-based monitoring system to counteract the issue of the central manager being a single point of failure. However, these extensions remain asynchronous in nature.

Under AIM, if a vehicle's trajectory is rejected, it has to stop and wait for its turn before being considered again. AIM is quite effective under low traffic, but it rather leads to a stop-and-go behavior in high-traffic situations due to the lack of enforced synchronicity and pre-planning of potential trajectories. The repeated re-requesting required by vehicles when their request is rejected further leads to a high communication overhead as well as a reduction in performance as vehicles have to slow down and sometimes come to a complete stop if they cannot reserve a spot. This effectively trades throughput for network overhead. While the timeout between requests can be reduced to increase performance, this will greatly increase network traffic, potentially compromising communication reliability.

In [3], a time-sensitive technique called Crossroads was presented to eliminate the effect of network and computational delay. This was later refined in [40], where a robust intersection management technique called RIM is proposed. RIM accounts for model mismatches and external disturbances and further considers the usage of different speeds for turn maneuvers, specifically by having vehicles travel with a high speed to the intersection and then cross with a slow, passenger-friendly speed. However, RIM's assumptions about vehicle size ($2\ m \times 6\ m$), lane width ($10\ m$) and maximum traffic flow (0.1) significantly limit its real-world applicability.

2.2. Synchronous Arrival Pattern

In a synchronous intersection protocol, vehicles are subjected to a strict spatio-temporal pattern, which is unique for each intersection. While these strategies assume homogeneous and fully automated traffic, they are very efficient in terms of throughput and reducing delay.

A popular such protocol is the *Ballroom Intersection Protocol (BRIP)* [11] [12], which synchronizes the arrival of vehicles into predetermined spatio-temporal patterns. To this end, several intersection layouts (i.e., different combinations of lanes and directions) are considered, which can be accessed by vehicles entering the intersection area such that they can adhere to the permitted time slots without relying on vehicular communications. Similar to AIM, the intersection is divided into square sectors, specifically with sides equal to $S_{BRIP} \geq L + W$, with L and W being vehicle length and width. With BRIP assuming homogeneous traffic, L and W inadvertently represent the dimensions of the longest possible vehicle allowed at the intersection, constraining vehicle size and intersection dimensions against each other. As a consequence, BRIP is not suitable fort most existing intersections, but rather requires infrastructural modifications.

More recently, BRIP was extended into two new protocols called CSIP [6] and DSIP [7]. CSIP (Configurable Synchronous Intersection Protocol) increases inter-vehicle distances to allow for better guarantees of collision avoidance when considering positioning errors by GPS, while DSIP (Distributed Synchronous Intersection Protocol) is a decentralized traffic protocol for mixed-traffic environments, i.e., when human participants are also present.

Throughput comparisons for CSIP [6] are made with regard to common signalized intersections. Here, they also list the assumption of same-size vehicles as one of the limitations of their work, which also affects BRIP. They note CSIP has reduced throughput compared to BRIP, but since it is focused around passenger comfort, that is to be expected and accepted.

DSIP [7] also compares throughput to standard traffic light intersections as well as STIP[12] and focuses on cooperative driving through V2V-communication, effectively implementing an asynchronous spatio-temporal protocol by simply scheduling to avoid collisions. It was further extended in [8] to consider cooperative perception for dynamic decision-making in mixed-traffic environments.

While these protocols clearly outperform conventional traffic lights, they also introduce additional limitations, effectively increasing inter-vehicle distances. This reduces their throughput when compared to BRIP. On the other hand, similar to BRIP, they are still based on stringent space requirements as stated before.

2.3. Adjacent Approaches

This section briefly discusses approaches that are not necessarily divisible into synchronous or asynchronous traffic protocols, but are still relevant to the contents of this thesis.

2.3.1. Platooning

The previously discussed approaches are basically intended for single-vehicle crossing. In [39], homogeneous vehicles going in the same direction may dynamically group into manageable clusters, which are then scheduled through the intersection in an asynchronous fashion. This allows for a reduction in travel and wait time as well as the overall communication overhead compared to single-vehicle crossing, similar to a platooning-based approach.

Further, in [50], vehicles are grouped into pre-determined zones, taking among others their length, width and arrival direction into account. These groups are then scheduled to cross together, but without regarding metrics such as wait time on conflicting fairness, interleaving vehicles from different arrival directions and the impact on throughput.

In [13], policies based on stop signs were introduced for autonomous intersection management and later refined into a platoon-based autonomous intersection management (PAIM) [14]. This is based on a time-triggered 4-phase plan (i.e., one phase for each direction). PAIM was shown to result in a reduction in travel delay and fuel consumption and to increase throughput with respect to conventional traffic lights. However, neither space requirements nor fairness-related metrics are discussed. In addition, since the 4 phases are fixed in time and order, it is difficult to adapt PAIM to scenarios of heterogeneous vehicles traveling in random directions.

2.3.2. Machine Learning

The recent rise of machine learning (ML) and large language models (LLMs) naturally leads to considerations of their application in the context of autonomous driving and intelligent intersection. For example, in [9], an end-to-end learning-based autonomous driving system named SuperDriver AI is presented, where Deep Neural Networks (DNNs) determine the driving maneuvers while ensuring road safety, which included a visual attention module to provide interpretability.

However, when considering the safety-critical nature of autonomous driving, the black-box character of current ML approaches makes guaranteeing safety or interacting with human participants almost impossible when machine learning is used as the driving mechanism of the traffic protocol. Further, especially in the context of system accidents [67] [68] (which are discussed in more detail in Chapter 4), the rising complexity and chaotic interaction between the layers of ML approaches runs the risk of escalating their occurrences, as warned frequently in recent times, e.g., in [17], [20] and [49]. This is exacerbated by the fact that ML strategies are based on training data, which rarely contains conditions leading to unexpected system accidents. As such, ML techniques are not explicitly considered as the sole controlling mechanism

for the concepts discussed in this thesis, although they have substantial potential when used in supporting systems such as cooperative perception [4] or path planning [89].

2.3.3. Game Theory

Especially in the context of mixed traffic settings with human participants, understanding the cooperation and competition behavior of traffic participants is of paramount importance [72]. Here, game theory approaches can be used due to their ability to model rationally selfish or cooperative behaviors, which can allow for the accommodation of human drivers in a mixed traffic system.

For example, in [44], a game-theoretic framework is used to propose a human-like payoff design methodology for automated decision making at unsignalized intersections for the interactions of two to four vehicles at a four-arm intersection. Similarly, [47] investigates a conflict resolution approach which fully considers personalized driving preferences to replicate human-like driving and execute personalized decisions by using personal payoff functions. In [81], level-k decision making is used to model heterogeneous and interactive traffic in a multi-intersection setting. Finally, [21] also proposes a conflict decision model using both fuzzy logic inference and game theory strategies under the Nash equilibrium (i.e., when no player can improve their expected outcome via changes of their own strategy in a static, noncooperative game) [42].

While these approaches are promising, especially in the context of mixed traffic where they can realistically model and respond to human driving behavior, their applicability to large scale situations and real-world scenarios is limited. This is also mentioned by several of the aforementioned publications, and as such they are not considered as the driving mechanisms behind traffic protocols for intelligent intersections in this thesis. However, these approaches can play a vital role in extending existing traffic protocols to various stages of mixed traffic by either simulating the sometimes unpredictable nature of human participants for existing traffic protocols, as noted in [81], or by introducing protocol-side strategy components to interact with human participants in a real-world mixed traffic scenario.

2.4. Summary

This section briefly summarizes the key related works discussed previously and presents a structured comparison in Table 2.1. The table highlights each reference with its presented approaches, the mechanisms or techniques used, as well as the key differences between the existing work and the approaches proposed in this thesis.

Ref.	**Approach**	**Mechanism**	**Contrast with proposed**
[10][24][25] [26][27][36] [78]	AIM, Semi-AIM and extensions	Asynchronous multi-agent FCFS, reservation-based	Communication overhead and reactive planning caused by asynchronicity are avoided in the proposed synchronous approaches.

Continued on next page

Ref.	**Approach**	**Mechanism**	**Contrast with Proposed**
[6][7][8] [11][12]	BRIP, CSIP, DSIP	Synchronized spatio-temporal traffic protocol	Escalating space requirements and reliance on well-behaved traffic. Both issues are addressed by the proposed approaches.
[40]	RIM	Robustness-centric mitigation of model mismatches and disturbances	RIM also uses varying speeds, but its assumptions about traffic flow, vehicle size, and lane width limit real-world applicability, unlike the proposed approaches.
[13][14] [39][50]	PAIM, spontaneous and fuzzy logic grouping	Grouping- and platooning-based approaches	Reliance on well-behaved traffic, similar to BRIP. Fairness and waiting time are not considered.
[4][9][89]	SuperDriver AI, etc.	ML-based approaches	The black-box nature limits guarantees, and emergent behavior increases the risk of system accidents. The proposed approaches address these issues.
[21][44] [72][81]	Risk field framework, prospect theory, Level-k reasoning.	Game theory approaches	Emergent behavior is inherent to game theory models, risking system accidents. Multi-agent games usually exhibit exponential scaling. The proposed approaches do not face these issues.

Table 2.1.: Comparison of existing literature with the proposed approaches.

Chapter 3.

Fundamentals of Intelligent Intersections and Traffic Protocols

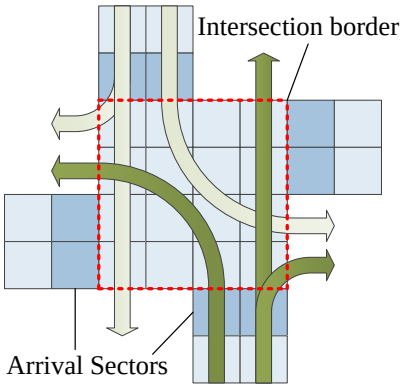


Figure 3.1.: Intersection layout, divided into square sectors.

This chapter covers the fundamentals of intelligent intersections and traffic protocols, as well as the specific approaches utilized in this thesis. With the traffic protocols at the core of this thesis being synchronous, particular attention is paid to underlying concepts of such protocols.

All presented concepts apply to both left-hand and right-hand traffic. However, for simplicity and consistency, right-hand traffic is assumed in all figures and explanations.

3.1. Models and Assumptions — Synchronization

The purpose of a traffic protocol is to schedule vehicles in a way that allows them to cross safely and efficiently, thereby reducing stops and ultimately increasing throughput. This is achieved by translating the vehicles' *arrival patterns* into a collision-free *crossing pattern*. In the specific case of synchronous traffic protocols, as discussed earlier, this is achieved by having predetermined patterns available at the intersection border into which vehicles are synchronized such that they can cross together in a synchronized group called *set*.

To ensure collision-free crossing, spatio-temporal planning of the intersection center is necessary as all potential collision points are clustered there, see Fig. 3.1. To this end, in this thesis and similar to other works from the literature, the intersection is divided into square same-size *sectors* in a $n \times n$ grid (i.e., with a *granularity* of n), as described next. In contrast to those works however, this thesis also utilizes rectangular sectors (instead of square sectors) to accommodate existing infrastructure, which is described later in Chapter 7 — for ease of exposition, this chapter is concerned with square sectors only.

3.1.1. Vehicle Lengths and Sectors

Synchronous traffic protocols in the literature differentiate in the way they select the size of a sector, as discussed in Chapter 2. Specifically, for BRIP [11], sectors are assumed to cover a lane and are therefore assumed to be square and large enough such that even the largest vehicle can fit in the sector with

$$S_{BRIP} > W_v + L_v, \tag{3.1}$$

with W_v and L_v being this vehicles' width and length respectively. While this has its benefits for homogeneous traffic with same-size vehicles, realistic traffic contains vehicles of varying lengths. Therefore, constraining the system with the largest possible, even if rare, vehicle in mind turns out to be ineffective for the shorter, more common, vehicles, as this leads to excessive empty spaces due to the connection of sector size and inter-vehicle spacing. To counteract this dependency, in this thesis, vehicles having to fit into a sector lengthwise is lifted as a requirement. Instead, the existing infrastructure (i.e., existing lanes in the intersection) determine the sector size and all vehicles exceeding a sector lengthwise are handled as *overlength* vehicles — they are the exception, not the norm.

This allows the lane width of the intersection to dictate the sector size, assuming that all lanes have identical widths, making the proposed approaches easier to deploy on existing infrastructure, as it does not impose any space restrictions [51]. On the contrary, traffic protocols are rather parameterized by the existing infrastructure.

Further, to allow describing distances smaller than S, σ is defined as a *sector fraction* of S, in particular, $\sigma = \frac{S}{k}$ for any positive integer k. For ease of exposition, σ is set to 1 m limiting S to multiples of 1 m, however, other values are also possible.

The distance from front bumper to front bumper of two consecutive vehicles on the same lane is denominated as *vehicle period* P and includes the length L of the leading vehicle. Therefore, the inter-vehicle separation is given by:

$$D = P - \lceil \frac{L}{\sigma} \rceil.$$

Assuming that vehicles always fit within one S, both vehicle period P and inter-vehicle separation D can be expressed as multiples of S to keep them independent of absolute values. Further, as a minimum, there has to be at least one empty sector S between two consecutive vehicles to guarantee a minimum inter-vehicle distance of S, even when vehicles exactly fit their sectors (i.e., $L = S$).

To coordinate arrival times, synchronous traffic protocols speed up/slow down vehicles (within the allowable speed limit and avoiding unnecessary stopping), which has an effect on the vehicle period. Generally, it is assumed that most vehicles fit entirely into one sector, i.e., $L_i \leq S$, with L_i denoting a vehicle i's length. This is mostly true for $S = 6$ m. Longer vehicles (e.g., trucks, buses, trams, etc.) occupy more than these initial sectors and are, thus, considered overlength vehicles as per the traffic protocol. Clearly, overlength vehicles impact the inter-vehicle separations to their follower and delay vehicles with conflicting trajectories, as described next.

3.1.2. Overlength penalty

Depending on the length L of a vehicle, this may not fit entirely into one sector. While most regular vehicles (i.e., passenger vehicles) have a length of $L \leq 6$ m, there also exist exceptionally

long vehicles (e.g., buses, trucks, etc.) [30] that exceed this length. In this thesis, these are referred to as *overlength vehicles*, requiring additional space denominated as the *overlength penalty* O.

$$O = \max\left(0, \left\lceil \frac{L-S}{\sigma} \right\rceil\right) = \max\left(0, \left\lceil \frac{L}{\sigma} \right\rceil - k\right). \tag{3.2}$$

Note that due to k's non-linearity and dependency on the other parameters, it is determined experimentally following [64]. Further details are provided in Section A.4.

While increasing the sector size S reduces the overlength penalty (if $S \geq L$, the overlength penalty is $O = 0$), this can lead to space-hungry intersections, requiring excessive lane widths. In this case, overlength vehicles are treated as the default case. Instead, the overlength penalty incorporated into the traffic protocol to account for the low probability of occurrence of overlength vehicles, as discussed later in Chapter 4 and Fig. 4.2. That is, the overlength penalty O needs to be added to the vehicle period P to ensure sufficient inter-vehicle separation D.

The specific impact of overlength vehicles on each traffic protocol is described later in their corresponding chapters, notably 4.1 and 7.2.4.

3.1.3. Cycles and the Two-Speed Scheme

In contrast to most approaches from the literature, this thesis employs a two-speed policy. Thereby, vehicles are assigned one of two speeds, namely a higher crossing V_{HI} and a reduced, passenger-friendly speed V_{LO} for the intersection crossing. Note that, whereas a vehicle can also drive through the intersection at V_{LO}, it cannot turn either right or left at a speed of V_{HI} to ensure passenger comfort[1]. In this thesis specifically and to simplify exposition, the speeds are setup such that $V_{HI} = 1.5V_{LO}$, however, other values are also possible.

To still allow for considerations independent of concrete speed values, the concept of *cycles* C is introduced to describe the time required to traverse the distance of S, i.e., one sector length, at a speed V_{LO}, i.e., $C = \frac{S}{V_{LO}}$. Correspondingly, a vehicle driving with the high speed V_{HI} covers $1.5S$ of distance during one cycle C, or requires $2/3 \cdot C$ to traverse a distance of $1S$.

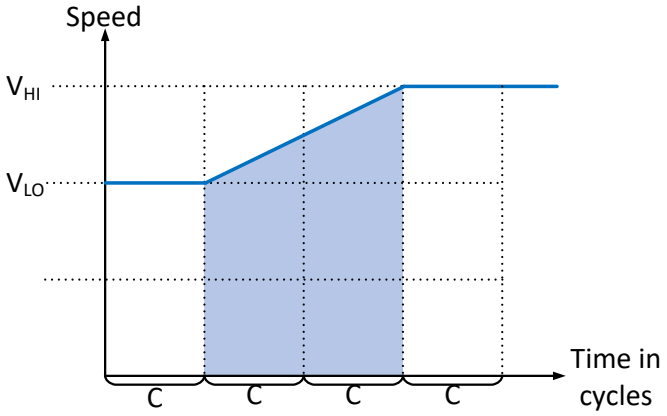


Figure 3.2.: Speed transitions over $2C$, with $S = V_{LO} \cdot C$.

[1] Note that this limits the turn maneuver only — vehicles making a left turn can drive with V_{HI} until the intersection border an then slow down to V_{LO} to perform the turn, as described next.

Transition between speeds (Fig. 3.2): Since the system allows for two different speeds (V_{LO} for turn and V_{HI} for drive-through maneuvers with a relation of $V_{HI} = 1.5V_{LO}$), transitions between them need to be defined in advance. Specifically, it is assumed that it takes any vehicle two cycles (i.e., $2C$) to transition from one speed to the other at a constant acceleration/deceleration — see Fig. 3.2.[2]

For a constant acceleration or deceleration a, the distance traveled during the time t of the transition can be described as:

$$\Delta d = V_{start} \cdot t + \frac{1}{2} \cdot at^2.$$

In the case of the previously described intelligent intersection, $V_{start} = V_{LO}$ and $t = 2C$. The acceleration/deceleration a from V_{LO} to $V_{HI} = 1.5V_{LO}$ or vice versa over $2C$ leads to an acceleration/deceleration of $a = \pm(\frac{V_{LO}}{2} \cdot \frac{1}{2C}) = \pm\frac{V_{LO}}{4C}$, resulting in the following:

$$\Delta d = V_{LO} \cdot 2C + \frac{1}{2} \cdot \frac{V_{LO}}{4C}(2C)^2,$$

where inserting $S = V_{LO} \cdot C$ leads to a traveled distance of $2.5S$ as per:

$$\Delta d = 2S + \frac{1}{2}S = 2.5S.$$

This is the same for both acceleration and deceleration — both transitions take a time of $2C$ and vehicles traverse a distance of $2.5S$ during that time. This can also be visualized in Fig. 3.2, where the shaded area underneath the speed/time graph is equal to $2.5(V_{LO} \cdot C) = 2.5S$, as each dotted square there equals $0.5S$.

For the specific values of $S = 6\ m$, $V_{LO} = 30\ km/h$ and $V_{HI} = 45\ km/h$, this leads to a braking deceleration and acceleration of roughly $2.9\ m/s^2$ or $0.295g$, with $g = 9.81\ m/s^2$ being the gravitational acceleration on Earth. This is well below the maximum of $7\ m/s^2$ possible on a normal dry asphalt road [84] and does not exceed the average maximum deceleration rates for combustion cars of $3.88\ m/s^2$ and $4.53\ m/s^2$ [18].

To summarize, vehicles making a turn will start decelerating $2.5S$ before the intersection, transitioning from V_{HI} to V_{LO} over a time of $2C$. They will then cross the intersection with V_{LO}. Once they have exited the intersection, they will accelerate again from V_{LO} to V_{HI} over a time of $2C$, again covering a distance of $2.5S$ in that process.

3.1.4. Synchronization Strategies

The most efficient crossing strategy in terms of throughput is vehicles crossing from all four directions simultaneously, denoted as *four-way synchronization*, leading to a set of four vehicles. However, depending on vehicle dimensions and drive direction, this is only possible under specific circumstances, otherwise leading to a less efficient (but more common) mode in which only vehicles from opposing lanes cross simultaneously, which is called *two-way synchronization*.

While the synchronization capabilities of each approach proposed in this work are described in their corresponding chapters, there are specific cases common to all these approaches, which are briefly described here.

[2]Note that, in principle, any other relation between V_{HI} and V_{LO} or time frame for the transition between speeds are also possible. However, the specific behavior of the transition between speeds will then differ from Fig. 3.2.

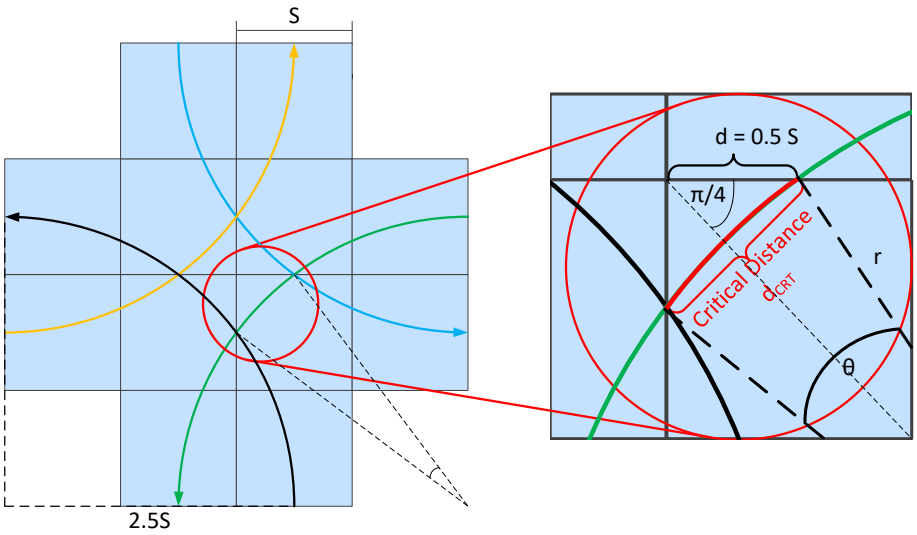


Figure 3.3.: Collision points and critical distance d_{CRT} for left turns.

Regarding left-turn synchronization: When a vehicle executes a left turn (see Fig. 3.1), their curved trajectory is approximated to be 1/4-th of a circumference with radius $r = 2.5S$, assuming that vehicles travel from the middle of their current lane to the middle of their target lane, see the left side of Fig. 3.3. Hence, the covered distance in full sectors can be calculated as follows:

$$d_{LT} = \frac{\pi}{4}(2 \cdot 2.5S) = \frac{5\pi}{4}S = 3.9270S. \tag{3.3}$$

This leads to left-turning vehicles exiting the intersection center slightly (0.073 cycles) before the 4-th cycle after they entered the intersection center. While this *desynchronizes* their trajectory, it does not compromise safety due to leaving the intersection center earlier. In addition, they still have a sufficient distance to preceding vehicles in the same direction ($> 1.9S$). For simplicity, it is therefore assumed that the distance covered by a left turn is $d_{LT} = 4S$.

Ideally, each set of left-turning vehicles consists of 4 vehicles in a four-way synchronization. However, since vehicles turning left on perpendicular lanes have overlapping trajectories, these approach up to a distance denoted as *critical distance* d_{CRT}:

$$d_{CRT} = r \cdot \ \theta = 2.5S \cdot \theta = 0.7095S, \tag{3.4}$$

where θ can be calculated through the Law of Sines, see Fig. 3.3. That is:

$$\frac{r}{sin(\frac{\pi}{4})} = \frac{d}{sin(\frac{\theta}{2})}.$$

Therefore, if any of the vehicles turning left has $L \geq 0.7S$ (i.e., is longer than 4.2 m assuming $S = 6\ m$), four-way synchronization is no longer possible. In that case, only opposing lanes can perform left-turn maneuvers simultaneously since their trajectories never coincide, leading to two-way synchronization.

Regarding vehicles driving through: Depending on intersection and vehicle dimensions, vehicles can drive through on all four directions simultaneously, provided they are able to vacate areas with conflicting trajectories before the vehicle on the perpendicular lane arrives, see Fig. 3.4.

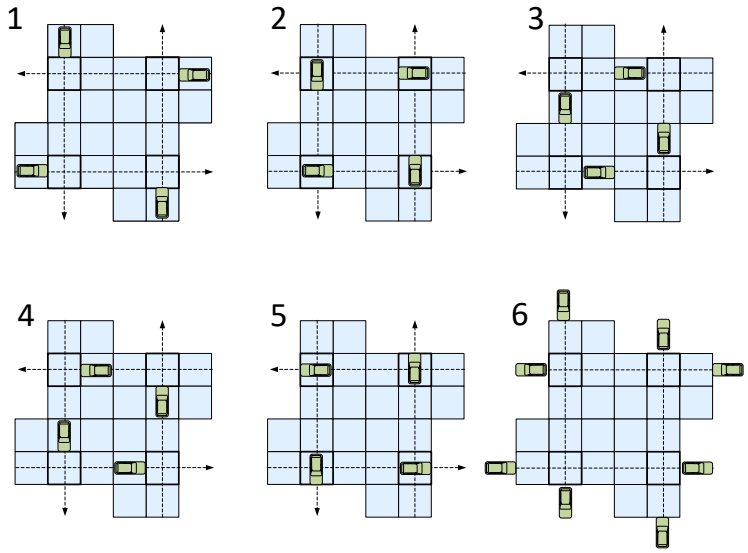


Figure 3.4.: Four-way synchronization of vehicles driving through.

Clearly, depending on the size of the intersection (and consequently its sectors) and the crossing vehicles, this maneuver might not be possible in all settings as the distances to both vehicles in the same set as well as vehicles in the following set can violate the inter-vehicle distance in the case of overlength vehicles. Similar to the left-turn maneuver, only vehicles on opposing lanes can then cross simultaneously, again enforcing a two-way synchronization.

Note that for both two-way and four-way synchronization, the vehicle length of the longest vehicle in the set determines whether the entire set incurs an overlength penalty due to the need for synchronicity to the next set.

Two-way and four-way synchronization, as the names suggest, attempt to always coordinate vehicles. While this allows for high throughput under well-behaved, homogeneous traffic, it might not always be beneficial for heterogeneous and less dense traffic, leading to larger delays or under-filled sets, as described next in Chapter 4. Further, Chapter 7 introduces a more flexible strategy to schedule vehicles utilizing *blocking patterns*, which retain the benefits for well-behaved systems while introducing more flexibility under randomized traffic, and which are the core mechanism of FleXS-TP, a protocol introduced in detail in the same chapter.

Chapter 4.

Probabilistic Reliability Modeling

As a subdomain of autonomous driving, intelligent intersections naturally are safety-critical systems — the consequences of failure are considerable. However, no single safety layer is infallible and every system can experience unavoidable **system accidents** (or „normal accidents“) [67] [68] — inevitable and unpredictable interactions between a system's components that will occur despite best efforts to avoid them. Notable examples of in recent times are the nuclear accident on Three Mile Island [66] [69] or the accident of Air France Flight 447 [15].

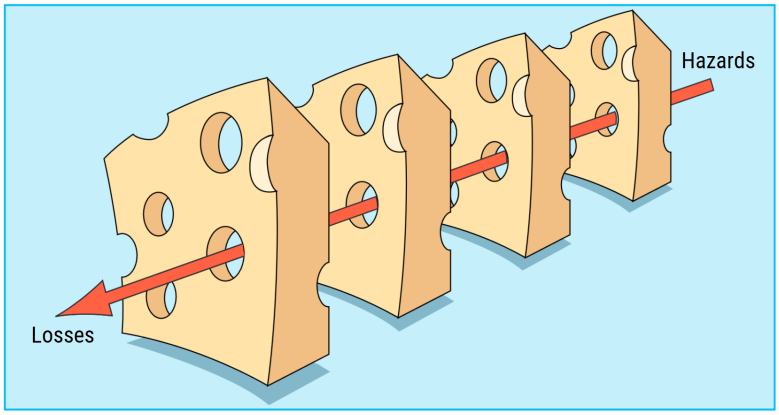


Figure 4.1.: The Swiss Cheese Model [73] [74] visualizes system-level causes of system accidents: the vulnerabilities (holes) of each layer have to line up.

Such accidents are frequently described using the *Swiss Cheese Model* [73] [74], see Fig. 4.1 — all individual holes had to line up for the accident to take place. Naturally, this is counteracted through redundancy by means of multiple safety layers, i.e., a system failure can only occur when *all* layers fail simultaneously. Assuming that each layer has a very small probability of failure ($< 1\%$), adding another safety layer reduces the probability of a total system failure by order of two magnitudes.

Further, due to the open-ended nature of intelligent intersections, i.e., vehicles can enter and leave at arbitrary points in time, it becomes difficult and inefficient to perform analyses based on deterministic methods, where all guarantees have to be towards the worst possible case, no matter how unlikely it is to occur.

For this reason, this chapter proposes a probabilistic approach for such a system through the example of communication reliability, which is a critical component. Specifically, physical parameters influencing communication reliability are handled not through a single, deterministic worst case, but with a stepwise worst case, each having its own occurrence probability. This allows for a probabilistic modeling of communication reliability, reducing the stringency of a

single such safety layer to massively improve efficiency, while adding additional safeguards to not reduce, but rather increase overall system reliability [55][56]. This further has implications towards protocol space efficiency, which is detailed later in Chapter 5.

The primary risk of unreliable communication at the intersections stems from interference/-packet loss through collision, caused by excess numbers of vehicles at the intersection. The worst-case, maximum vehicle count at the intersection can be considered through the physical factors influencing it — namely vehicle length, traffic density and traffic direction. This section considers these factors and their impact on the worst-case vehicle count by focusing on the a **single-lane, two-way synchronized** setting, however, the results generalize to other cases as well.

For ease of exposition, the impact of variable vehicle length *alone* is described first, i.e., under well-behaved, homogeneous traffic. This is then extended by the consideration of traffic density and direction.

4.1. Vehicle Length Distribution

As briefly discussed in Section 3.1.2, assuming homogeneous traffic also assumes a uniform vehicle length L. If vehicles are expected to fit into a sector S (see again Fig. 5.1), extending this to heterogeneous traffic while considering communication reliability leads to an impasse regarding the systems worst case:

- If vehicles are expected to fit into S, the largest possible vehicle at the intersection determines S, increasing the default inter-vehicle distance.
- When considering communication reliability, having more vehicles in the system decreases communication reliability — which is achieved by having a large number of smaller vehicles with short inter-vehicle distances.

This is resolved by removing the restriction that each vehicle has to fit lengthwise into a single sector, as discussed in Section 3.1.2. Specifically, the sector size is adjusted to be able to accommodate the most common vehicle length.

To highlight the impact of that changes, this section first introduces the standard deterministic approach for worst-case vehicle count estimation. Then, a probabilistic modification is presented.

4.1.1. Deterministic Approach

The deterministic approach to estimate the worst case vehicle count in terms of communication reliability, i.e., the maximum number of vehicles in the intersection, leads to the assumption of *homogeneous*, continuous traffic of only overly short vehicles, i.e., only short motorbikes driving through, even though that occurrence probability is minuscule. Clearly, this is an overly pessimistic assumption which can be alleviated later through probabilistic estimates.

The deterministic maximum vehicle count considers how many vehicles with $L = L_{min}$ physically fit into the intersection range R when considering a given inter-vehicle separation D. In this thesis, it is assumed that $D = S$, which leads to a single-lane deterministic worst-case of:

$$n_{max}^{hg} = \left\lceil \frac{R}{S + L_{min}} \right\rceil . \tag{4.1}$$

Note that this also assumes fully homogeneous traffic where all vehicles have a length of $L = L_{min}$ and perform the same maneuver — once vehicles of different lengths and directions are introduced, this is no longer possible due to desynchronization and conflicts between different lanes.[1]

Including vehicles of different lengths can be achieved via the assumptions introduced in Section 3.1.2 of vehicles fitting into a sector S, with vehicles exceeding the sector size ($L > S$) being handled as overlength vehicles. In this case, provided that $L_{min} < S$, the deterministic maximum vehicle count from Eq. 4.1 is modified to:

$$n_{max} = \left\lceil \frac{R}{2S} \right\rceil . \tag{4.2}$$

This deterministic vehicle count is safe, but overly pessimistic, which results in an inefficient design of the communication strategy. To overcome this drawback, probabilistic estimation is deployed as discussed next.

4.1.2. Probabilistic Approach

Although the previously established deterministic worst-case vehicle number of n_{max} can realistically occur, its occurrence is extremely unlikely. In other words, it is extremely unlikely that *exclusively* vehicles with $L \approx 1.8\ m$ (i.e., motorbikes) are in the intersection's surroundings and consistently crossing without conflicts (i.e., only driving through).

To obtain a more realistic estimation, the probability of occurrence of specific vehicle types can be used to achieve probability distribution of the number of vehicles at the intersection.

To this end, publicly available statistical data of vehicles' sales on the European market from 2005 to 2015 [30] is combined with sales of exceptionally short cars [29], motorbikes [71] and trucks [30] to achieve a weighted distribution of 258 different vehicle lengths in Europe as displayed by green dots in Fig. 4.2.

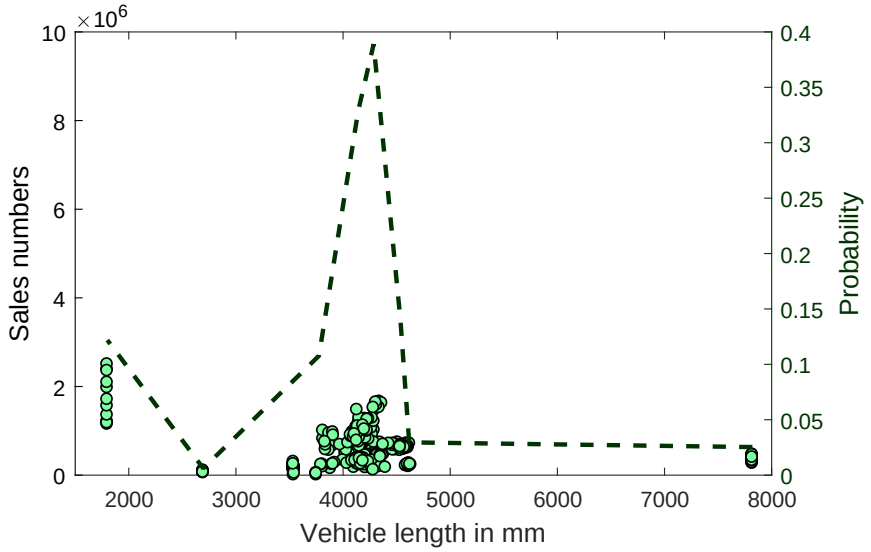


Figure 4.2.: Probability distribution of vehicle lengths.

[1] For completeness, considerations of such a fully homogeneous system are included in Section A.4.

Considering all permutations of up to n vehicles within the intersection — i.e., the worst-case number of vehicles via the deterministic approach — amounts to n^{258} possibilities. Clearly, it is not feasible to use exhaustive methods for this. As a result, to make this tractable, vehicles are clustered according to similarities. This way, $x = 8$ weighted categories were obtained, whose resulting probability distribution is displayed by the dashed line in Fig. 4.2.

Further, having vehicles cross together in synchronized sets requires these sets to be parameterized by the longest vehicle in the set, i.e., the longest vehicle in a set determines the separation to the next vehicle on *all affected lanes*.

This allows a description of the number of sets on a given lane with a *multinomial distribution* [61], which has the following probability mass function $f(\cdot)$:

$$f(k_1, ..., k_x, p_1, ..., p_x) = \frac{n!}{k_1!...k_x!}(p_1^{k1} \times ... \times p_x^{k_x}). \tag{4.3}$$

Here, x is the number of different categories of vehicles, $n = \sum_{i=1}^{x} k_i$ is the number of vehicles on a lane and p_i is the corresponding probability that the given vehicle is of type i with $1 \leq i \leq x$. Eq. (4.3) gives the probability of having n sets on a lane, whereby k_1 are of type 1, k_2 are of type 2, and so on. Note that the order of vehicles on the lane plays no role.

Considering the upper bound n_{max}^{hg} for fully homogeneous traffic going only one way from Eq. 4.2, the corresponding minimum bound n_{min}^{hg} can be determined by assuming only the longest vehicles, i.e., trucks with L_{max}, are present:

$$n_{min}^{hg} = \left\lceil \frac{R}{S + L_{max}} \right\rceil \tag{4.4}$$

As before, modifying this to account for is done via the assumption of vehicles fitting into a sector S, with vehicles exceeding the sector size ($L > S$) being handled as overlength vehicles. Assuming $L_{max} > S$, the incurred overlength penalty modifies Eq. (4.4) to:

$$n = n_{min} = \left\lceil \frac{R}{3S} \right\rceil.$$

With both bounds known, the different combinations of k_1 to k_x can be considered for $n \in [n_{min}, n_{max}]$, with $1 \leq x \leq 8$. Now, not every combination of n vehicles is *valid*. If the combined lengths and inter-vehicle separations exceed R, the combination is not valid.

Similarly, if the combined length is below R, but leaves room for *additional vehicles*, the combination is not valid either as this does not correspond to the worst-case number of vehicles, i.e., the maximum possible number of vehicles that fit into R.

As a result, a valid combination for a set count n is a set $(k_1, ..., k_x)$, where the following condition holds for any $1 \leq i \leq x$:

$$R - (3S) < 2n \cdot 2S + k_1 \cdot L_1 + ... + k_x \cdot L_x < R. \tag{4.5}$$

Accounting for all cases of $n \in [n_{min}, n_{max}]$ leads to the probabilities of worst-case set counts on a single lane and at the intersection. Here, the worst-case number of vehicles at the intersection equals twice the lane's worst case set number, i.e., $c_{max} = 2 \cdot n_{max}$, as this example employs two-way synchronization.

Different vehicle lengths impact this probabilistic vehicle count due to the previously introduced concept of overlength, since vehicles exceeding the sector size according to Eq. (3.2) to prevent collisions with subsequent vehicles.

These concepts are now modified by also considering different traffic densities and directions.

4.2. Traffic Density and Direction

Deterministic worst-case conditions assume consistent, sufficiently high traffic to have vehicles crossing constantly, interweaving in two-way synchronization without empty spaces. It further assumes that vehicles arrive in a well-behaved fashion, i.e., with the exact intended trajectories necessary to fully fill the intersection.

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, the predetermined crossing patterns of synchronous traffic protocols can have unoccupied slots. This is also the case if the vehicles present have different intended trajectories than provided by the current crossing pattern.

As previously mentioned, for ease of exposition, this section is concerned with a one-lane, two-way synchronized setting. Hence, depending on traffic, it is possible to have only one single vehicle crossing at a time, as well as having no vehicle crossing at all if no vehicles can safely reach the next feasible slots. These are denoted as *half set* (only one of two potential vehicles) and *empty set* (no vehicles) respectively.

If only one vehicle crosses, i.e., a half set, there are no overlapping trajectories. In this case, the crossing is executed as efficiently as the drive direction allows.

As for empty sets, a fixed time buffer of $g = 2C$ is assumed, which is equal to the cycle cost of the shortest possible maneuver (vehicles turning right) in the intersection. Note that having a minimum cycle cost for empty sets plays a significant role for the subsequent analysis, as it avoids having excessive numbers of very small empty sets.

To determine the occurrence probabilities of full, half and empty sets, recall that the intersection example in this section processes vehicles in sets of two. That is, whenever a new vehicle reaches the arrival zone (see Fig. 3.1), 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. Now, if this incoming vehicle arrives too late, it cannot safely catch up to the next slot, and a half set is formed with only the vehicle on the opposing lane and the new vehicle is scheduled at the next opportunity. Should that vehicle be 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) delay the vehicle $2C$. This procedure is repeated, i.e., empty sets will take place delaying the new vehicle by a multiple of $2C$, until this can reach the intersection at the scheduled time.

4.2.1. Probabilities

To obtain the probability of a full, a half or an empty set, the relation between the *arrival rate* and the *exit rate* is used. The arrival rate specifies how many vehicles actually arrive at the intersection per time unit, whereas the exit rate specifies the maximum number of vehicles that 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 influences how many vehicles can cross during a specified amount of time. As a variable of traffic density, the *traffic ratio* r is described by arrival and exit rate:

$$r = \frac{r_{arr}}{r_{exit}}. \tag{4.6}$$

If $r = 1$, the intersection is completely filled. Similarly, if $r = 0$, the intersection is empty.[2] Therefore, the probability of a vehicle being absent can be modeled with:

$$p_A = 1 - r. \tag{4.7}$$

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

$$p_E = p_A^2 = (1 - r)^2 . \tag{4.8}$$

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

$$p_H = p_A \cdot (1 - p_A) + (1 - p_A) \cdot p_A = 2 \cdot r - 2 \cdot r^2. \tag{4.9}$$

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

$$p_F = (1 - p_A) \cdot (1 - p_A) = r^2. \tag{4.10}$$

These probabilities are displayed in Fig. 4.3 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 $p_E + p_H + p_F = 1$.

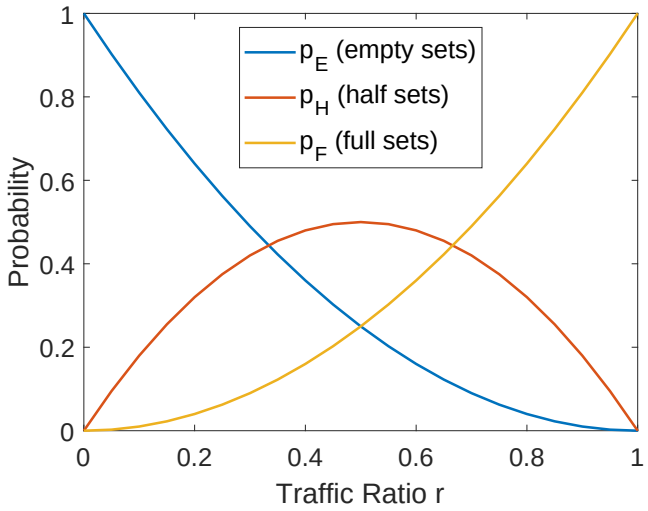


Figure 4.3.: Probabilities for half, empty and full sets in relation to traffic ratio r.

4.2.2. Deriving Vehicle Count

To estimate the number of vehicles at a intersection, all combinations of different vehicle sets within the intersection's range R have to be considered. To this end, each combination of vehicles crossing is assigned a 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. For the example in this section, this results in Table 4.1, R, T and L mark the directions for

[2]Note that, as this traffic ratio r is modeled uniformly across all lanes of the intersection. This simplifies the analysis but may yield conservative (worst-case) absence probabilities when applied to individual lanes or turning movements, particularly under unbalanced traffic distributions.

right turn, driving **t**hrough and **l**eft turn respectively, while A marks a vehicle being **a**bsent, leading to an *empty* slot on one of the lanes, i.e., a *half-pair*. These values are also provided standard vehicles on the left and for overlength vehicles on the right of each cell.

Standard/OL	**West**			
East	**Right turn**	**Through**	**Left turn**	**Absent**
Right turn	2/3	3/4	4/5	2/3
Through	3/4	3/5	5/6	3/4
Left turn	4/5	5/6	4/5	4/5
Absent	2/3	3/4	4/5	2/2

Table 4.1.: Cycle cost g for turn combinations without OL/with OL.

As briefly mentioned in Section 4.2, two empty slots lead to an empty set, which has a fixed cost of $g = 2C$.

These distances can then 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. 3.1.

As before, the upper and lower bounds for the number of vehicle sets within the intersection's range are considered, now taking different directions into account. For the upper bound n_{max}, only sets with the shortest possible cycle cost of $g = 2$ (i.e., right turns without overlength vehicles or empty sets only) are assumed to be present. This leads to the following deterministic worst case:

$$n_{max} = \left\lceil \frac{R}{\min\limits_{\forall i}(g_i) \cdot S} \right\rceil = \left\lceil \frac{R}{2S} \right\rceil, \tag{4.11}$$

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 Table 4.1 there are $y = 5$ different cycle costs (2 to 6). Note that this leads to the same deterministic worst case as before in Eq. (4.2)

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

$$n_{min} = \left\lceil \frac{R}{\max\limits_{\forall i}(g_i) \cdot S} \right\rceil = \left\lceil \frac{R}{6S} \right\rceil, \tag{4.12}$$

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

As each set can have 0, 1 or 2 vehicles, the lower bound on the number of vehicles c_{min} is zero, i.e., there are only empty sets and the upper bound c_{max} equals deterministic worst case, i.e., when the intersection is as tightly packed with vehicles as physically possible. This leads to:

$$c_{min} = 0, \tag{4.13}$$

$$c_{max} = 2 \cdot n_{max}. \tag{4.14}$$

When generating different sequences of vehicle sets, it must be guaranteed that these fit into the intersection's range R according to the used traffic protocol. Following [54], a *valid* sequence must meet the following condition:

$$R - 2S <= \sum_{\forall g_i}^{\max_{\forall i}(g_i)} k_{g_i} \cdot g_i \cdot S < R, \tag{4.15}$$

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 to occur, the probabilities leading to that case must be considered, specifically the probability of having overlength vehicles p_{OL}, the probability of having a certain *combination of drive direction* 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 leads to the probability p_{set} of a given set to occur by:

$$p_l^{set} = p_D \cdot \{p_F \vee p_H \vee p_E\} \cdot \{p_{OL} \vee (1 - p_{OL})\} \tag{4.16}$$

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 current set 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, probabilities of all sets that comprise the sequence are multiplied:

$$p_{seq} = \prod_{\forall l} p_l^{set}. \tag{4.17}$$

As before, this leads to a weighted probabilistic worst-case vehicle count p_c, i.e., several gradually worsening cases with increasingly small probabilities.

Details regarding the implementation of this vehicle count estimation algorithm and its performance are discussed in detail in Section 8.2.

4.3. Communication Protocol/Scheme

When it comes to intelligent intersections of any kind, vehicles have to either communicate with each other (*V2V* communication) or with existing infrastructure (*V2I* communication). To facilitate this, a communication scheme has to be defined.

This section presents the communication scheme used for the approaches presented in this thesis, which is also evaluated later in Chapter 9. All vehicles within a certain distance from the intersection periodically transmit their current coordinates and speed to the RSU. The RSU then processes this data and replies with updated speed values, which vehicles have to adopt. The periodicity of the transmission is determined by the speed of vehicles and the required physical resolution, i.e., the distance a car can travel before it must receive an update. For example, if the speed is 50 km/h and an update must be received every 1 m traveled, the transmission interval is equal to $\frac{1\ m}{50\ km/h} = 72\ ms$.

To lower complexity, it is assumed that vehicles and the RSU are within range of each other and, hence, can communicate directly in a single-hop fashion. This avoids delays over multiple

hops and can be easily achieved with off-the-shelf hardware, as shown in [88], where tests showed possible ranges of ≥ 300 m using 5.9 GHz vehicle-to-vehicle modems. It is further assumed that there is no interference with neighboring systems, which is typically achieved by using multiple (different) radio channels.

Similar to other vehicular ad-hoc network (VANET) scenarios, a peculiarity of an intelligent intersection is that communication between vehicles and RSU is only established for a short time. Using classic synchronous communication methods for data exchange such as TDMA results in considerable overhead, since time slots must be re-assigned continuously. To reduce this complexity, a hybrid approach as depicted in Fig. 4.4 is used instead.

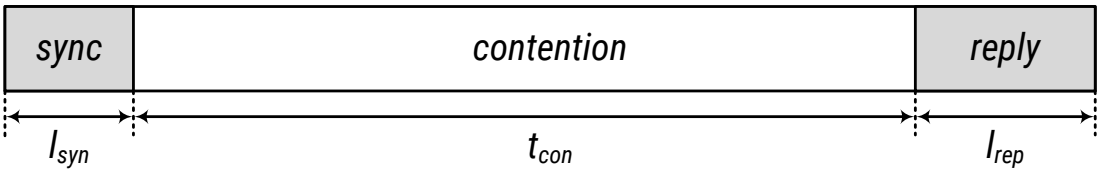


Figure 4.4.: Transmission cycle: sync, contention phase and reply [54], own depiction.

Here, the RSU periodically transmits a sync beacon, which indicates the start of a new communication cycle and informs newly arrived vehicles about the intersection. This is followed by a contention phase, in which vehicles transmit a request packet to the RSU using a random transmission pattern, as explained later. At the end of the cycle, i.e., after all request messages have been received, the RSU calculates a new speed value for each vehicle and transmits these together in one reply messages.

This hybrid, RSU-initiated topology has the advantage that there are only collisions, where these cannot be prevented, i.e., among request messages in the contention phase. The RSU messages, i.e., sync and reply, on the other hand, are not disrupted, which increases the overall reliability of the network. In the next section, the collisions within the contention phase are analyzed in more detail.

4.3.1. Medium access control

As mentioned before, synchronous communication protocols would incur in high overhead due to the high mobility within the network. To solve this, instead a hybrid frame structure is used and implemented into an asynchronous protocol in the contention phase to effectively convey data. To this end, any asynchronous medium access control (MAC) protocol could be used, for example Carrier Sense Multiple Access (CSMA). However, in this thesis, the probabilistic approach presented in [64] is used, where vehicles transmit requests as a sequence of redundant packets with random back-off times in-between transmissions. This scheme has the advantage that it allows calculating worst-case transmission reliabilities, which facilitates a connection to the fail-safe mechanisms discussed later.

In addition, it does not rely on sending acknowledgments — these are also not needed, since feedback is obtained from the reply message — which further reduces overhead and generated traffic and, therefore, improves the overall transmission reliability.

Using the MAC protocol from [64], vehicles now transmit their request message k times within the contention phase of length t_{con}. Each of these transmission requires certain amount of time depending on the number of bits to be transmitted and the transmission rate. This

time is denoted as packet length, whereas the length of the request message is denoted as l_{req}. The time in-between two consecutive request messages is randomly selected within a lower boundary t_{min} and an upper boundary t_{max}:

$$t_{max} = \frac{t_{con} - l_{req}}{k},$$
$$t_{min} = \frac{t_{max}}{2}.$$

To evenly spread messages across the whole contention phase t_{con} and, therefore, achieve a lower collision rate, t_{max} is set to the highest possible value. That is, the k-th packet of a vehicle must start transmitting at latest at $t_{con} - l_{req}$ time to finish transmission before the end of the contention phase. Similarly, t_{min} is chosen as small as possible to achieve a greater interval $t_{max} - t_{min}$ and, hence, increase the number of possible choices for back-off times, which is again beneficial for reliability. However, t_{min} is lower bounded to $\frac{t_{max}}{2}$, since smaller values would allow more than one request message within $t_{max} - t_{min}$, for example, if a vehicle randomly selects t_{min} as a back-off multiple times. As shown in [64], this reduces the possible reliability and is therefore not desirable.

The probability that one packet of a given vehicle is interfered results from the ratio between $2(2n-1)l_{req}$, i.e., the fraction of the interval $[t_{min}, t_{max}]$ that is potentially being used by other vehicles, and $t_{max} - t_{min}$, i.e., the total length of this interval. The term $2(2n-1)l_{req}$ results from the fact that potentially $2n-1$ other vehicles are sending packets of length l_{req} and that any overlapping between two packets is considered to yield packet loss, which explains the factor 2. Since this probability is independent of the individual transmission, a binomial distribution can be used allowing to calculate the resulting worst-case reliability of the network:

$$p = 1 - \left(\frac{2(2n-1)l_{req}}{t_{max} - t_{min}} \right)^k, \tag{4.18}$$

where again $2n$ is the number of vehicles at the intersection, t_{min} and t_{max} are back-off interval boundaries, l_{req} is the length of a request message and k is the number of (redundant) transmissions.[3]

Eq. (4.18) assumes a deterministic value of n. To extend this for different possible values of n (such as probabilistic estimates of the worst-case number of vehicles as later discussed in Section 9.1.2), (4.18) must be extended to:

$$\bar{p} = \sum_{n=n_{min}}^{n_{max}} p_n \left(1 - \left(\frac{2(2n-1)l_{req}}{t_{max} - t_{min}} \right)^k \right), \tag{4.19}$$

where n corresponds to the worst case vehicle count with $n_{min} \leq n \leq n_{max}$ and p_n is the corresponding probability of n.

4.3.2. Physical layer

Using the physical layer (PHY) and protocol packet data unit (PPDU) header of the IEEE 802.11p standard [38], each transmission consists of a preamble field, a signal field and variable length data field — see Fig. 4.5.

[3] Note that [64] does not provide a formula to calculate k, but this is determined experimentally. This is briefly covered in Section A.4.

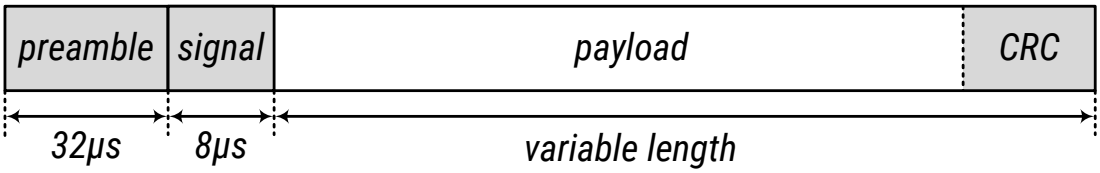


Figure 4.5.: Protocol packet data unit (PPDU) header from IEEE 802.11p, own depiction.

The preamble consists of a series of predefined symbols that allow the demodulator to lock onto the carrier frequency and decode the data. The signal field defines the data field length and the transmission rate, while the payload of the message is contained in the data field. In order to securely detect any corrupted transmission, a 2 bytes cyclic redundancy check (CRC) is included into the data field.

The IEEE 802.11p PHY can support data rates between 3 and 27 Mbps, when using a channel bandwidth of 10 MHz [38]. However, since the data rate is unknown prior to reception of the signal field, the preamble and signal field are transmitted at the lowest rate of 3 Mbps for improved robustness. This results in a duration of 32 μs and 8 μs respectively. For the data field, 6 Mbps is selected, which is a good compromise of robustness and transmission speed. The total length of any packet can hence be calculated:

$$l_{packet} = 32\ \mu s + 8\ \mu s + \frac{payload + 2\ bytes}{6\ Mbps}.$$

Regarding the payload, each request message contains the location and speed of a vehicle as well as an identifier, which is needed associate the speed values within the reply message to the corresponding vehicles. Assuming 8 bytes for the GPS position, 1 byte for speed and 17 bytes for the ID[4], the length of a request message is 78 μs. Similarly, the sync message holds 10 bytes of information about the intersection, e.g., number of lanes, etc., and, hence, has a length of $l_{syn} = 56\ \mu s$. The reply message, on the other hand, is of variable length, depending on how many request messages were (successfully) received. For each received message, it replies with 17 bytes ID and 1 byte speed.

Suitability of IEEE 802.11p: IEEE 802.11p-based protocols have a number of drawbacks that make them unsuitable for applications with real-time requirements and high data traffic [28]. First, they are based on carrier-sense multiple access (CSMA), which performs poorly at high traffic loads due to large overhead and an unsuited back-off scheme [31][65]. Second, since transmission time is divided in control channel (CCH) and service channel (SCH) intervals, channel efficiency is poor and cannot exceed 50 % [16]. Third, transmission intervals are only 100 ms [16], which severely lowers the intended physical resolution of 1 m as discussed above (recall that vehicles have to send x messages within $t_{con} \approx 72\ ms$).

Suitability of 5G: 5G is gaining attention due to its high performance, i.e., low latency and high transmission speeds [23]. In contrast to 802.11p, 5G uses a centralized, non-orthogonal multiple access (NOMA) approach where data transfers are parallelized and controlled by a base station [75]. This leads to better performance especially at high density traffic [23].

[4] In this case, the vehicle identification number (VIN) is used, as this does not only contain a unique production number, but also useful information about the vehicle itself, such as weight, length, etc.

However, 5G requires expensive infrastructure, which is partly not available in most countries and cannot be just deployed for an intersection. In contrast to this, 802.11p is much easier to deploy in one isolated location and currently associated with less costs than 5G.

4.4. Fallback-Mechanisms

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 in the case of an impending partial or full system fault.

In the case of communication reliability, using probabilistic estimates leaves a residual risk that the maximum vehicle count estimation is exceeded, which compromises the communication scheme. In that case, packets from or to the RSU do not arrive at the same rate anymore (see Section 4.3), which clearly compromises safety. Although the residual risk of the probabilistic estimates in this section is negligible (as discussed later in Section 9.1.1), this probability describes the chance of any combination of vehicles to exceed this vehicle number — when some vehicles have already entered the intersection, this becomes an issue of conditional dependence, making an exceedance not a sudden event, but rather a gradual increase, with predictable steps leading to it. A specific example for this gradual prediction is discussed in Section 9.1.3.

Once such an exceedance is impending, there are different possible strategies to reduce the communication risk incurred. Some potential examples are listed below. This is further discussed in Section 10.2.3.

System-wide measures: For example, the system can mandate a gradual cruise speed reduction once the number of vehicles at the intersection starts converging towards the maximum. As the physical resolution of the communication scheme (i.e., the updates must be received every 1 m traveled, as discussed in Section 4.3) depends on the vehicles' speeds, reducing the speed even by 10% from $V_H I = 45\ km/h$ to 40.5 km/h increases the maximum transmission interval (i.e., the contention phase t_{con}) from 80 ms to 89 ms — potentially allowing the system to accommodate a higher number of vehicles without compromising communication.

In the case of immediate exceedance, e.g., by unexpected occurrences such as a tandem motorcycle motorcade entering the intersection, the system can mandate a system-wide response such as switching to manual mode (with traffic light behaving in a conventional manner) or even mandating all vehicles to stop at the intersection border to safely restart the crossing processes with some delay.

System-wide measures however are not a meaningful way to respond to a complete loss of communication such as a RSU failure, for which vehicle-side measures are required.

Vehicle-based emergency system: In aviation, the Traffic Collision Avoidance System (TCAS) [35] [83] is a system constantly monitoring the airspace around aircraft to issue resolution advisories in case of a violation of minimum inter-aircraft distance. These resolution advisories supersede all other instructions in terms of priority and are coordinated between the participating aircraft, with TCAS being installed in every aircraft above 5.7 t or above 19 passengers.

As most modern vehicles are already equipped with several safety mechanisms such as collision avoidance systems (CAS), forward collision warning systems (FCW) or autonomous

emergency braking (AEB), similar systems with gradually increasing severity can be implemented.

In the most extreme case of a total loss of communication (e.g., in the case of a complete RSU failure, not just a missed communication package), a similar approach can be used to detect impending risks between nearby vehicles and give individual resolution advisories for vehicles or even lanes with overlapping trajectories. This clearly is advantageous to simply issuing an intersection-wide stop command or an immediate switch to manual mode, as such sudden system-wide changes have a high risk for emergent and unexpected issues, provided that these resolution advisories are coordinated with all vehicles in the immediate area. This however leads to a substantial communication overhead, which the aviation sector solves with dedicated communication frequencies for these systems.

In the case of an exceedance of maximum vehicle count, which does not immediately guarantee a communication failure, but simply a higher risk of packet loss, executing an extreme, system-wide response can potentially cause more risk than the initial packet loss risk, making rather smaller measures more appropriate[5].

Here, other already existing systems such as the distance assistant of adaptive cruise control (ACC) can maintain normal operation in the case of a temporary communication loss. Generally, however, due to the usually tight and interwoven nature of synchronous traffic protocols, individual vehicles performing uncoordinated braking rather increases the risk of collisions than reducing it due to delay propagation, desynchronizing the affected lane from the remaining ones. This is further exacerbated by the fact that, in theory, once a vehicle has gotten its intended trajectory cleared after entering the intersection range, it does not receive revised instructions under normal conditions. Naturally, when vehicles deviate from their intended trajectories, such altered commands might become necessary to provide conflict resolutions to other vehicles now in conflict with the diverging vehicle.

In addition, vehicles should also implement a fail-safe behavior independent of the RSU such that, if communication goes down, 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 brake to prevent crashes. However, again, since these mechanisms disrupt normal operation jeopardizing utility, the system should be designed not to overuse them.

To summarize, it is clear that an intelligent intersection should be designed properly such that fail-safe mechanisms are not overused, but remain in place for rare exceptions. This way, probabilistic estimates can be used to guarantee safety during normal operation while, at the same time, reducing overdesign.

[5]This is briefly discussed in the context of system accidents in Section 10.2.2

Chapter 5.

Space Efficiency

If square sectors are used, the length of a sector on the western lane is inadvertently tied to the width of a sector of the crossing northern lane, as they intersect in the center — see again Fig. 3.1.

For a traffic protocol to synchronize vehicles on the different lanes such that they can cross the intersection in all possible directions without needing to stop, a safe inter-vehicle separation needs to be guaranteed on each lane. This safety distance depends on vehicles' dimensions, in particular, on the lengths of vehicles on the conflicting lanes. For example, the separation between two vehicles traveling from east to west should be sufficiently large for a vehicle to cross from north to south in between them.

Existing protocols, such as BRIP [11], account for this by considering the worst case, i.e., assuming that the largest possible vehicles are always present at the intersection. Combining this with the requirements of vehicles to always fit into one sector, see Eq. (3.1) and Fig. 5.1, leads to significant space requirements towards its sector size, which either forces the intersection to be built from scratch or imposes severe restrictions on the maximum length of a vehicle allowed at the intersection.

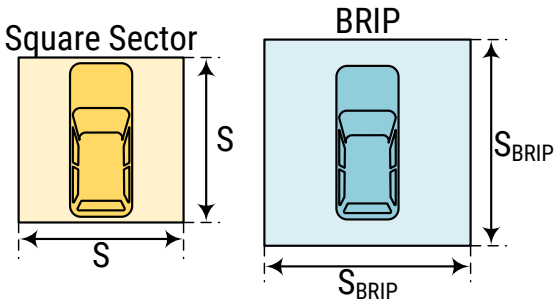


Figure 5.1.: Comparison of regular square sector and BRIP sector for the same vehicle.

To overcome this issue, this thesis proposes abandoning the requirement of vehicles always having to fit into a sector lengthwise. Aside from its implications regarding deterministic pessimism, as discussed previous in Chapter 4, this relaxes the space requirements imposed by a traffic protocol on the infrastructure, i.e., on the lane width. In that case, as proposed in this work, overlength vehicles need to be handled as exceptions instead of the norm. Having a fixed sector size, determined by the existing lane width, makes the proposed approach independent of the length of vehicles crossing the intersection (within the limits of them having to be able to cross safely). This allows for a more space-efficient traffic protocol that can be easily deployed on already existing infrastructure.

This chapter details the specific implications of these changes through a synchronous, space-efficient traffic protocol called Single-Vehicle Left, Through, Right (SV-LTR) and its differences from the BRIP variants, which are introduced next.

5.1. Ballroom Intersection Protocol (BRIP)

As mentioned in Chapter 2, BRIP is a synchronous, tightly interwoven spatio-temporal traffic protocol [11], which introduces several different types depending on the existing layout, as depicted in Fig. 5.2:

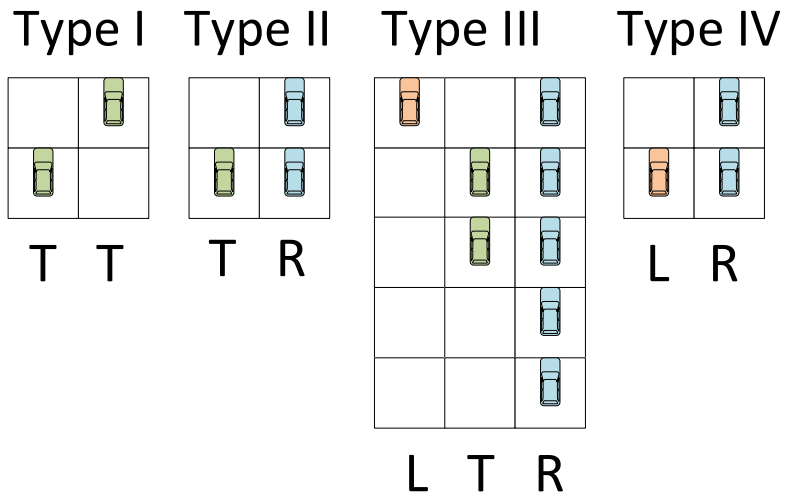


Figure 5.2.: Arrival patterns of BRIP Types [11].

As mentioned before with Eq. (3.1), a core assumptions of BRIP is that the sector size has to be at least $S_{BRIP} \geq L + W$, where L and W are the length and width of the largest vehicle crossing the intersection. Having a two-lane layout (such as BRIP Types I, II or IV) yield an intersection side length of

$$Size_{BRIP} = 4S_{BRIP} = 4(L + W) \tag{5.1}$$

leading to a space requirement of

$$Area_{BRIP} = 16S_{BRIP}^2 = 16(L^2 + 2WL + W^2) \tag{5.2}$$

Clearly, the longer vehicles in the intersection are, the less existing intersection can comply with BRIP's space requirements. In addition, since the definition of a *cycle* is bound to the sector length, a *BRIP cycle* can be considerably larger than the previously discussed cycles of the proposed protocol, as visualized previously in Fig. 5.1, increasing spatial requirements even more compared to the proposed approaches.

5.2. Single-Vehicle Left, Through, Right (SV-LTR) Protocol

Due to the fixed sector sizes achieved by uncoupling sector size from vehicle length, the proposed approaches have a fixed intersection width and area requirement that does not increase when longer vehicles are permitted for the intersection. Similar to BRIP above, the intersection width is determined by the chosen sector size S. For a two-lane layout, this amounts to:

$$Size_{SVLTR} = 4S. \tag{5.3}$$

Consequently, the space required for the intersection also remains static with:

$$Area_{SVLTR} = 16S^2. \tag{5.4}$$

This section presents a synchronous space-efficient traffic protocol labeled Left, Through, Right (LTR) [56]. As this protocol will later be extended to a platooning-based version, it is denoted as single-vehicle LTR (SV-LTR) in the remainder of this thesis.

As described before, SV-LTR assumes a fixed sector size S, see Fig. 5.1, with vehicles exceeding this sector size being handled as overlength vehicles following Section 3.1.2, also utilizing four-way synchronization whenever possible.

5.2.1. Drive through only/drive through and right turns

Due to the synchronous nature of the protocol, all vehicles arrive at the border of the intersection simultaneously. As previously discussed, vehicles cover a distance equal to S in one cycle when traveling at V_{LO}. However, since vehicles driving through travel at $V_{HI} = 1.5V_{LO}$, they traverse a distance of $1.5S$ per cycle. Therefore, the crossing process can be divided into 4 different steps (one per cycle), as depicted in Fig. 5.3, yielding a vehicle period P of $6S$ (i.e., 4 cycles at V_{HI}).[1] A shorter vehicle period can potentially lead to accidents with vehicles crossing perpendicularly, while a longer vehicle period decreases throughput.

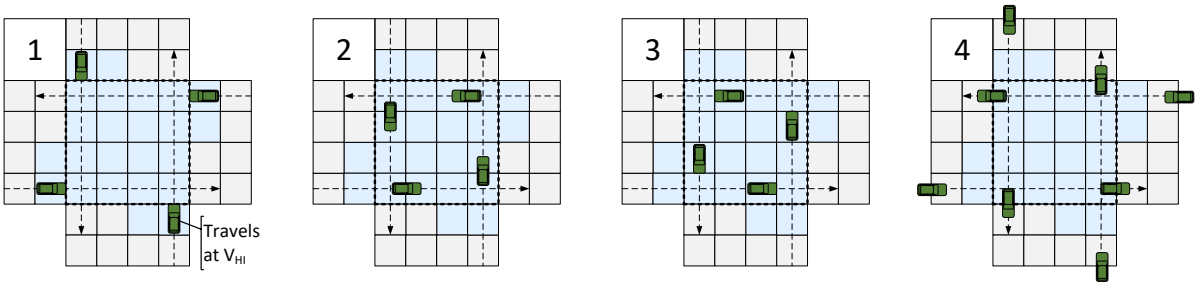


Figure 5.3.: Driving through in a *four-way synchronization*.

Note that it is possible that any of the vehicles in Fig. 5.3 turns right instead of driving through. Such a vehicle arrives at the intersection at the same time with the others (see step 1 in Fig. 5.4), but it turns right at V_{LO}[2] traversing an arc-like trajectory $d_{RT} = \frac{\pi}{4}S \approx 0.79S$ (i.e., 1/4-th of a circumference with radius $r = S/2$, assuming the vehicle travels from the middle of

[1] While a vehicle period $P = 5S$ would also be possible, P is selected with $P = 6S$ to preserve safety, since this maneuver is later combined with other maneuvers.

[2] Note that a vehicle turning right at the intersection may approach the intersection at V_{HI}, however, it has to slow down to V_{LO} for the turn maneuver.

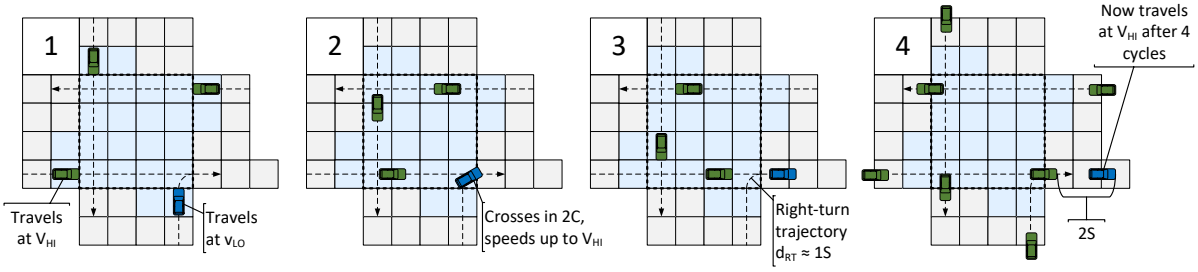


Figure 5.4.: Driving through with one vehicle turning right.

its current lane to the middle of its target lane. At V_{LO}, this maneuver takes around $0.79C$ to complete, however, which is approximated to one full cycle for simplicity. The vehicle then transitions to V_{HI}, which takes a time $2C$ and a distance of $2.5S$ (as discussed in Section 3.1.3), leading to vehicle period $P = 2S$ to its following vehicle on the same lane, see step 4 in Fig. 5.4.

Note that any single vehicle in the drive-through setting can be substituted with a vehicle turning right without increasing the vehicle period. Moreover, if all vehicles turn right, one can reduce the vehicle period even further, as described later when discussing right and left turns in Fig 5.6.

Finally, when adding another through/turn-right lane to this setting, vehicles on the outermost such lane will have to arrive at the intersection border with a delay of one cycle (with respect to vehicles on the innermost lane). Otherwise, these will collide with vehicles turning right from the innermost through/turn-right lane. In general, there will have to be an additional one-cycle delay per through/turn-right lane added to the setting.

5.2.2. Drive through and left turns

Having a set of vehicles driving through requires 4 cycles, as described above. Once these vehicles have cleared a sufficiently large portion of the intersection center such that collisions can be avoided, left turn maneuvers can start as depicted in Fig. 5.5.

As discussed in Section 3.1.4, left turns describe a curved path taking $\approx 4C$, see step 8 in Fig. 5.5.

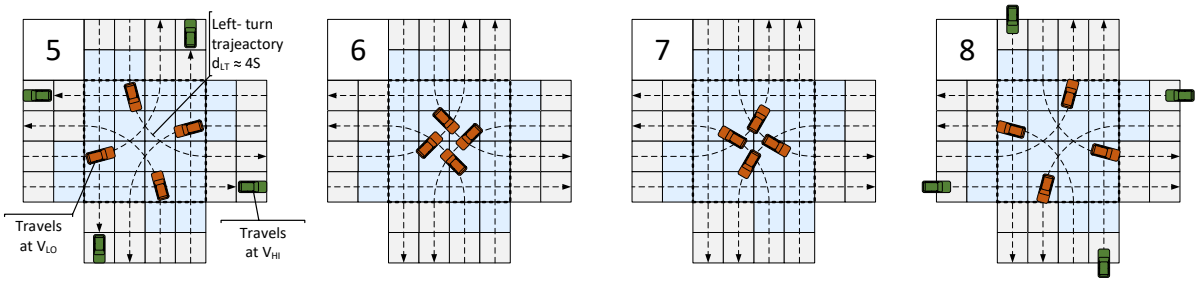


Figure 5.5.: Combined drive-through and left-turn maneuvers in a *four-way synchronization*.

Ideally, each set of left-turning vehicles consists of 4 vehicles in a four-way synchronization, provided none of the vehicles turning left has $L \geq 0.7S$ (i.e., is longer than 4.2 m assuming $S = 6$ m), as discussed in Section 3.1.4 regarding synchronization strategies. If one or more vehicles exceed this length, a four-way synchronization would lead to collisions. In that case, only opposing lanes can perform left-turn maneuvers simultaneously since their trajectories

never coincide, leading to a two-way synchronization.

In summary, interleaving vehicles driving through and turning left allows for a total throughput of 6 to 8 vehicles per 8 cycles. That is, 4 vehicles drive through in the first 4 cycles (step 1 to 4 as per Fig. 5.3), whereas either 2 or 4 vehicles turn left over a time of 5 cycles (step 4 to 8 as per Fig. 5.5).

5.2.3. Right and left turns

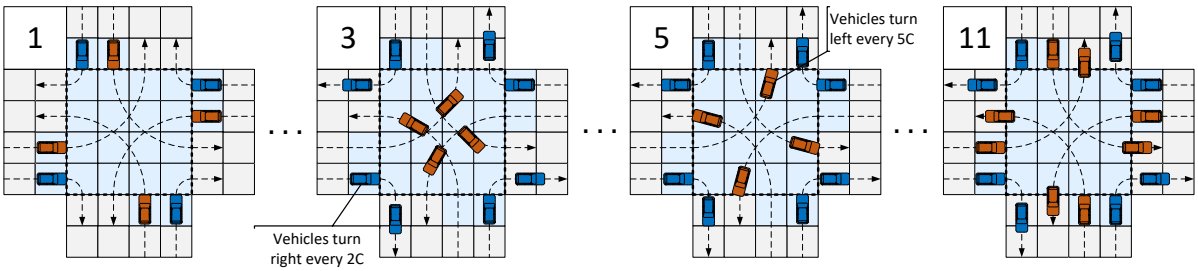


Figure 5.6.: Right and left turns in a *four-way synchronization*.

As shown in Fig. 5.6, since there are no conflicting trajectories, right and left turns do not interfere with each other. Similar to the previous case, vehicles turning left can cross every 5 cycles in either sets of 4, provided that all vehicle are shorter than $0.7S$, or sets of 2 otherwise. In addition, right turns allow for 4 vehicles to cross every 2 cycles and, hence, totaling a throughput of 24 to 28 vehicles per 10 cycles, consisting of 20 vehicles turning right and 4 to 8 vehicles turning left.

Since trajectories do not interfere with each other, this combination results in the highest possible throughput. Vehicles making right turns can cross in sets of 4 every 2 cycles. Also, vehicles turning left can cross in sets of 2 to 4 every 5 cycles. This results in 24 to 28 vehicles every 10 cycles, depending on whether two-way or a four-way synchronization needs to be used for left turns.

5.2.4. Considering overlength vehicles

All previous considerations assume that vehicles always fit entirely into one sector. Extending this to vehicles longer than S is achieved by introducing an overlength penalty which adds an appropriate number of sector fractions σ to the vehicle period P following Eq. (3.2).

As an example for sake of exposition, a vehicle with $L = 7.5\ m$ in an intersection with $S = 5\ m$ and $\sigma = 1\ m$ (i.e., $\sigma = \frac{S}{5} = 0.2S$) leads to $O = 3\sigma$. In a drive-through-only setting, this increases the set's vehicle period from $P = 4S$ to $P = 4.6S$ (i.e., 4 full cycles and 3 additional σ, with $\sigma = 0.2S$), reducing throughput from 4 vehicles per 4 cycles to 4 vehicles per 4.6 cycles.

Other crossing patterns are affected a similar way, i.e., the vehicle throughput is reduced by some amount that depends on the overlength penalty.

In particular, note that overlength vehicles unavoidably lead to a two-way synchronization in left-turn settings (i.e., they always exceed the critical distance as per Eq. (3.4)) and may further enforce a two-way synchronization in drive-through-only settings when $O \geq 10\sigma$, i.e., $2S$ with the values of S and σ assumed before.

To summarize, whereas BRIP's sector size increases depending on the vehicles present at the intersection, the removal of the restriction of each vehicle having to fit into a single sector lengthwise discussed in Section 4.1 completely uncouples vehicle length and sector size, leading to a static space requirement. A detailed evaluation and comparison is included in Section 9.3.2.

5.2.5. Algorithm Performance

To handle all vehicles in the intersection following SV-LTR as described in Section 5.2, the RSU periodically runs the Algorithm 1 presented below.

This algorithm i) monitors planned vehicles for deviations from their intended trajectory to guarantee safety and ii) assigns unplanned (i.e., newly arrived) vehicles to either an existing set or iii) creates a new set that accommodates an unplanned vehicle if there is no match.

To this end, the RSU loops over all vehicles in the intersection's region of influence, including vehicles that are not yet assigned to a vehicle set (i.e., that have arrived after the last execution of the algorithm). It then request updates from all vehicles in the system (line 3), e.g., position, speed, etc.

If a planned vehicle (already part of a set) deviates from its intended trajectory, the RSU triggers a fallback mechanism which brings the intersection to a fail-safe state to avoid collisions, e.g., all red lights (line 7).

For unplanned vehicles, the RSU checks all existing vehicle sets for empty slots that are compatible with the vehicle, i.e., the set has a slot the vehicle can realistically reach and that matches the vehicle's direction (line 18). If compatible, the vehicle is assigned to the corresponding set (line 20). If no sets are compatible, a new set is created for the vehicle (line 30).

Complexity: Since the algorithm checks all existing vehicle sets (lines 12 to 27) for all unplanned vehicles, it has a linear complexity of $\mathcal{O}(n)$, where n is the number of vehicles at the intersection. This is because there can only be one vehicle per set and the number of unplanned (i.e., newly arrived) vehicles is upper-bounded by the number of lanes, e.g., 8 for a two-lane intersection (2×4, i.e., two lanes in every direction).

Algorithm 1: SV-LTR control loop for all vehicles

```
1  foreach (v in Vehicles) /* Cycle through all vehicles */
2  do
3      Update(v); /* Get current vehicle information for v */
4      if (v.Planned == true) then
5          if (⌈v.CurrentPosition - ExpectedPosition(v)⌉ > threshold) /* Deviation? */
6           then
7              Fallback(); /* If deviation exceeds threshold, fallback mechanism */
8          else
9              continue; /* Check next vehicle */
10         end
11     else
12         foreach (s in Sets) /* Vehicle sets are sorted, from crossing time from
               earliest to latest */
13          do
14             if (s.Full == true) /* Set is full set, i.e., no empty slots */
15              then
16                 continue; /* Check next set */
17             else
18                 if (IsCompatible(s, v) /* Direction of v matches the direction of an
                       empty slot in s */
19                  then
20                     Assign(s, v); /* Assign v to s) */
21                     v.Planned=true; /* Flag vehicle as planned  */
22                     break; /* Step out of inner foreach-loop, go to next vehicle */
23                 else
24                     continue; /* No match, go to next set */
25                 end
26             end
27         end
28         if (v.Planned==false) /* v does not match any existing set */
29          then
30             s_new = CreateNewSet(v); /* Create new set that fits the trajectory of
                   vehicle v */
31             Assign(s_new, v); /* Assign v to new set s_new */
32             v.Planned = true; /* Flag vehicle as planned */
33             Sets.Append(s_new); /* Add s_new to list of sets */
34         else
35             continue; /* Vehicle is assigned - go to next vehicle. */
36         end
37     end
38 end
```

Chapter 6.

Platooning and Fairness

When again considering the cycle costs in Table 4.1, it becomes apparent that maximum throughput can be achieved by simply having more efficient maneuvers (i.e., right turns or vehicles driving through) take place without interruption. However, this will unfairly delay vehicles going in different directions for as long as these maneuvers take place, making this strategy unsuitable for real-world scenarios.

Still, having vehicles repeatedly perform the same maneuver allows for a tighter scheduling at the intersection, which evidences that transitions between maneuvers (i.e., ensuring all collision areas are vacated) are the primary cause of inefficiency. Therefore, this section is concerned with platooning vehicles with the same trajectory to increase efficiency, while still considering fairness.

Platooning, i.e., grouping several vehicles to perform coordinated actions, is a common strategy to maximize efficiency of CAVs. This allows for short inter-vehicle distances, which would not be possible otherwise [43]. Applying this concept to the previously introduced SV-LTR protocol, previously introduced in Chapter 5, leads to an approach called platooning-based LTR (PB-LTR) [52], which is described next.

6.1. Platooning-Based Left, Through, Right (PB-LTR) Protocol

Grouping vehicles with the same trajectory allows for a reduction in vehicle period in the direction of crossing, increasing the waiting time of vehicles with conflicting trajectories as a trade-off. Provided this incurred delay does not exceed a threshold denoted as the metric for fairness, this can improve overall throughput of the intersection. This chapter presents a platooning-based variant of the space-efficient SV-LTR called PB-LTR. The specific platoon crossing regimes and their transitions are discussed next.

6.1.1. Drive-through platoon crossing

If vehicles fit in one sector of length S, as shown in Fig. 5.3, a throughput of 4 vehicles per 4 cycles can be reached under single-vehicle crossing. However, having platoons (i.e., chains of vehicles) that are almost always longer than $2S$ enforces a two-way synchronization, where only vehicles on opposing (and not on perpendicular) lanes are allowed to cross at a time.

On the other hand, platoon crossing allows multiple vehicles to cross (in the same direction) in the least possible time as depicted in Fig. 6.1. The intelligent intersection then switches to allow vehicles to cross in a different direction.[1]

[1] Clearly, this is similar to conventional traffic lights that change after some time. However, the proposed platoon crossing allows for much shorter inter-vehicle separations achieving a considerably higher efficiency.

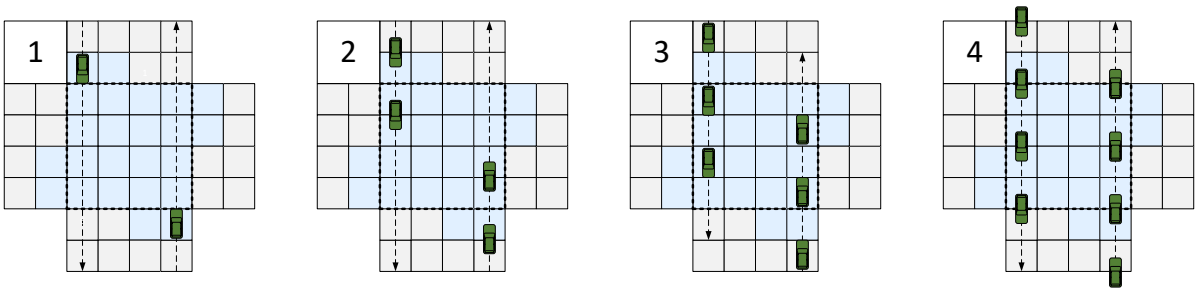


Figure 6.1.: Drive-through platoon crossing initialization.

There are three different cases leading to a state transition from a drive-through regime: approaching vehicles that turn left either in the same or perpendicular direction or vehicles driving through in the perpendicular direction. The different possible transitions are displayed in Fig. 6.2. Due to the increased speed $V_{HI} = 1.5V_{LO}$, transitioning between perpendicular drive-through regimes requires a non-integer amount of cycles $((4 + 1/3)C)$ for the next set of vehicles to reach the intersection center.

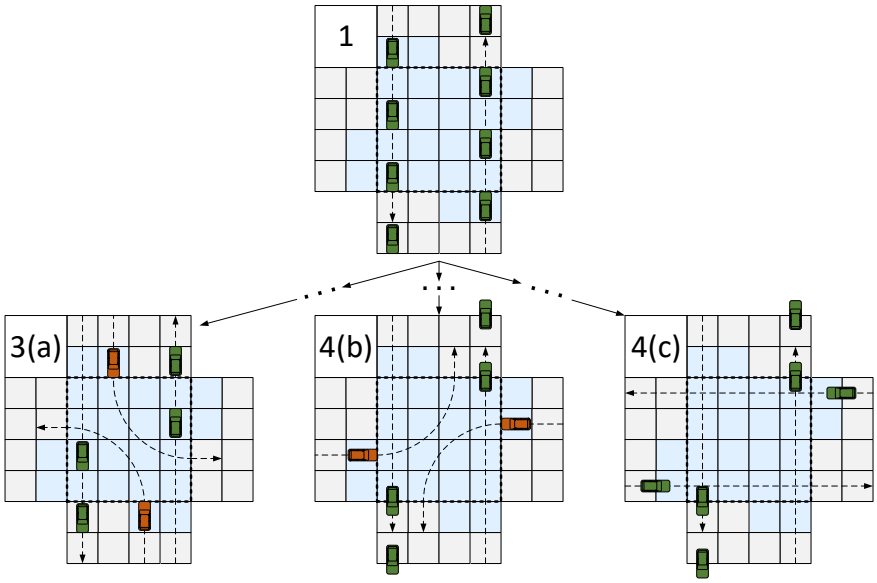


Figure 6.2.: Transition from drive-through platoon crossing to (a) a left-turn regime from the same direction (b) a left-turn regime from the perpendicular direction (c) a drive-through regime from the perpendicular direction.

6.1.2. Left-turn platoon crossing

As discussed above, a throughput of 4 vehicles per 5 cycles can be achieved for single vehicles turning left, if vehicles are shorter than $0.7S$. However, again, having platoons that are typically longer than $2S$ enforce a two-way synchronization, i.e., only vehicles on opposing lanes can cross at a time. On the other hand, this allows us to increase the throughput to 2 vehicles per 3 cycles. Necessary transitions to a left-turn platoon crossing regime are shown in Fig. 6.3. After an initialization phase of 5 cycles, 2 vehicles cross the intersection every 2 cycles with a vehicle period $P = 2S$.

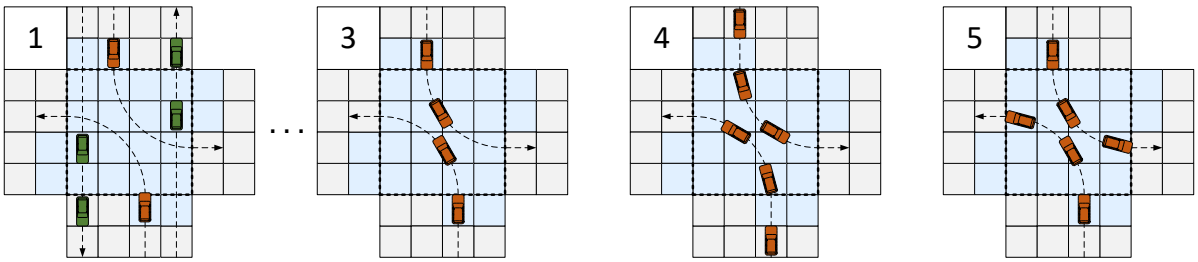


Figure 6.3.: Left-turn platoon crossing.

Similar to before, there are three different situations forcing a state transition from a left-turn regime: approaching vehicles that turn left in the perpendicular direction or vehicles driving through in either the same or the perpendicular direction, see Fig. 6.4. Again, step 1 in this figure corresponds to step 5 in Fig. 6.3, step 5c corresponds to step 1c in Fig. 6.3 and step 6a/b in this figure corresponds to step 1 in Fig. 6.1 respectively.

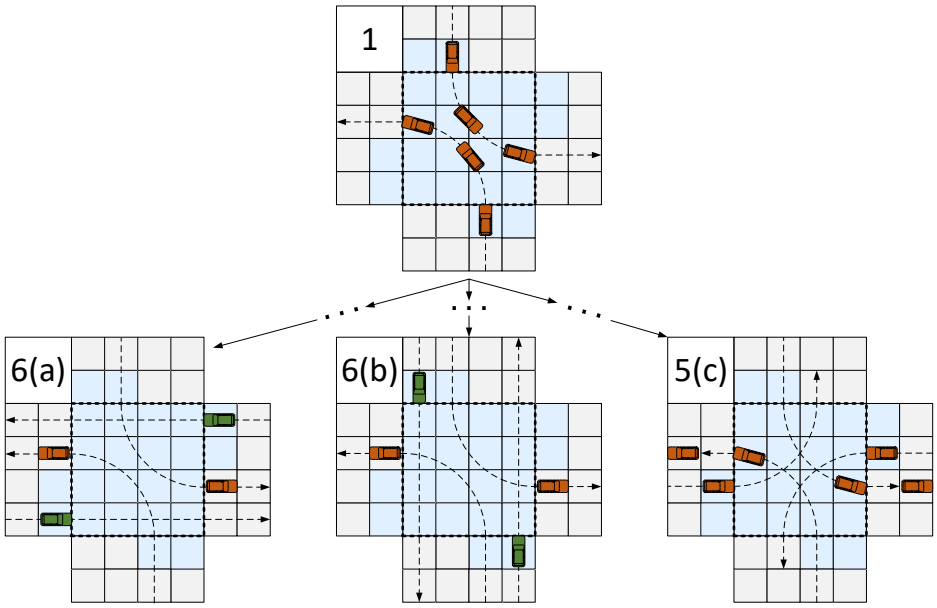


Figure 6.4.: Transition from a left-turn platoon crossing to (a/b) a drive-through regime from either direction or (c) a left-turn regime from the perpendicular direction.

6.1.3. Considering right turns

Since vehicles turning right do not have overlapping trajectories with vehicles turning left, they can coexist under both single-vehicle and platoon crossing. On the other hand, right-turn and drive-through trajectories overlap, however, it is possible to interleave them under single-vehicle crossing using a vehicle period of $P \geq 6S$ (see Fig. 5.3) to avoid collisions. Since drive-through platooning schemes use a vehicle period of $P = 2S$, interleaving drive-through and right-turn maneuvers is not possible unless an outer dedicated lane is available for right

turns. However, in that case, they are completely independent of the rest of the intersection, as also further discussed in Section 9.3 and Figure 9.9.

6.1.4. Maximum blocking time

To ensure *fairness* by avoiding long waiting times, the intersection switches from one platoon regime to another based a *maximum blocking threshold*, denoted by B_{max} and expressed in cycles. That is, having $B_{max} = 25$ indicates that the current platoon regime (e.g., on the southbound/northbound lanes) will run for 25 cycles before a transition is performed (at the next appropriate point in time) to allow the vehicles on the currently inactive lanes (in this example, the eastbound/westbound) to cross, provided any such vehicles are present.

Finally, all considerations regarding the handling of overlength vehicles and the overlength penalty discussed in Section 3.1.2 also apply to the platoon crossing regimes introduced before.

Chapter 7.

Flexible Synchronous Traffic Protocol (FleXS-TP)

Previous synchronous approaches such as BRIP [11][5][6] and both variants of LTR[54][51][52] synchronize vehicles into sets at the intersection border. This is very effective in a well-behaved setting in which vehicles arrive with a given order that either guarantees no conflicts (e.g., all vehicles turn right) or which allows solving conflicts (e.g., all vehicles drive through). When faced with random traffic however, these protocols require either large-scale rearranging, which is both complicated and costly, or must leave a substantial number of sets that are not completely filled, negatively impacting throughput.

In contrast to these traditional synchronous traffic protocols, this thesis also proposes a more flexible approach which schedules individual vehicles at the intersection instead of forming (synchronous) vehicle sets, but retains the rougher granularity of synchronous approaches to ensure for situational synchronicity. This leads to a lane-based *First Come, First Served* (FCFS) scheme, such that once a vehicle has been scheduled for crossing, all subsequently scheduled vehicles have to consider its trajectory to avoid conflicts. That is, an individual vehicle impacts all subsequent vehicles and some of them — having conflicting trajectories — might need to be delayed accordingly, i.e., they are blocked for some time, as signaled by individual *blocking patterns*.

This allows vehicles to be synchronized with each other rather than at the intersection border, in a proposed, space-efficient synchronous traffic protocol called *Flexible Synchronous Traffic Protocol* (FleXS-TP) [53]. These blocking patterns and the resulting traffic protocol is described next.

7.1. Blocking Chart and Blocking Patterns

For ease of exposition, to visualize these blocking patterns at the intersection caused by individual vehicles, a *blocking chart* is used, which acts as a visualization of intersection lane and sector blocking across time and specifically describes at which times the arrival sectors of the intersection are blocked for incoming vehicles, taking the ensuing trajectory into account.

An example for a blocking chart can be seen in Fig. 7.1. From left to right, this chart displays all four cardinal directions, i.e., north (N), east (E), south (S), and west (W). Each cardinal direction features two lanes each, i.e., left turn (L) and through/right turn (T/R), which corresponds to the intersections of Fig. 3.1. From top to bottom, the chart displays a timeline in cycles C. The first row in the blocking chart, i.e., at time $1C$, corresponds to the ego vehicle being in its corresponding arrival sector displayed in Fig. 3.1.

Different blocking types are marked through both color and alphanumerical code for the four different types of blocking that can occur, as detailed in the legend in Fig. 7.1.

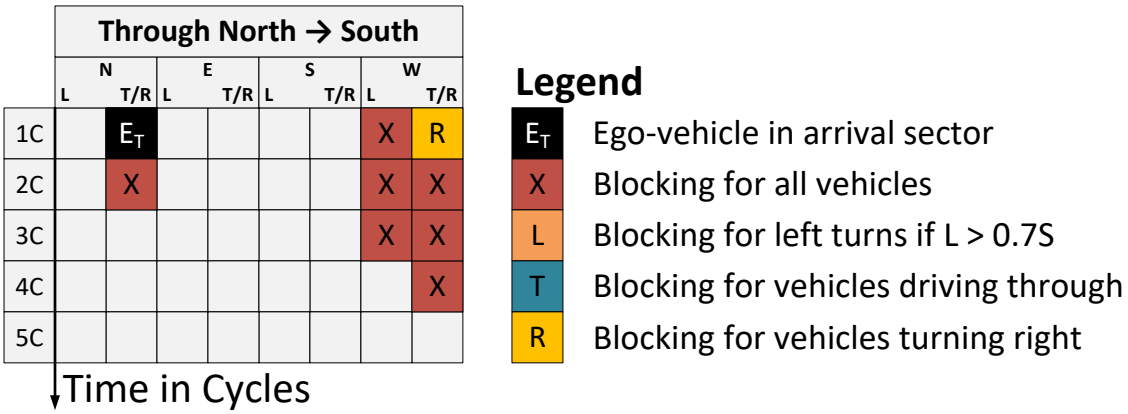


Figure 7.1.: Blocking chart example.

Red blocks with label „X“ indicate a general blocking for all types of vehicles. Orange blocking labeled „L“ is a conditional blocking for left turns if the ego-vehicle does not fulfill the condition for four-way synchronization by exceeding $L_V > d_{CRT}$ from Eq. (3.4). Finally, due to the shared right/through lane, blue blocks labeled „T“ indicate a blocking for vehicles driving through only, i.e., these can be ignored for vehicles making a right turn, and similarly yellow blocking labeled „R“ indicates a blocking for right turns only, but not for vehicles driving through.

To safely schedule a vehicle on the blocking chart, it must be guaranteed that:

1. The vehicle's blocks for ego-vehicle (marked in black with label „E“) do not overlap with *any* already existing blocking.
2. The blocking induced by the ego-vehicle on other lanes does not overlap with already scheduled vehicles' ego blocks.

Otherwise, the vehicle cannot be safely scheduled at that position on the chart. Note that a vehicle can have more than one block of ego-vehicle for overlength vehicles, as discussed later in detail.

In the example presented in Fig. 7.1, which is also discussed in detail in Section 7.2.1, a southbound vehicle reaches the arrival zone at $1C$ on the north T/R lane, which blocks vehicles from the western lanes.

In general, each individual maneuver has its own blocking pattern, which impacts subsequent traffic on different lanes depending on the intended direction. As vehicles are scheduled at the intersection, the proposed flexible synchronous traffic protocol superpose these blocking patterns without any overlap to guarantee safety.

With the help of these blocking charts, already scheduled maneuvers and their resulting impact on subsequent vehicles can be visualized. When a new vehicle arrives, the intersection's collective blocking pattern reliably shows the next possible time at which the vehicle can cross safely, i.e., a point in time at which the new vehicle's blocking pattern can be inserted without conflicts with previously scheduled vehicles. Note that some blocking segments by the new vehicle can overlap with blocking segments by another previously schedule vehicle, i.e., both vehicles block the same segments at the same time. However, clearly, the new vehicle cannot block already scheduled vehicles from before and must be delayed until these conditions are met.

Next, the blocking patterns for each of the different maneuvers (driving through, turning left and right) are described individually and the impact of vehicle length, specifically for overlength vehicles, is discussed briefly. This is done first for square sectors in Section 7.2 and expanded afterwards for variable sector lengths in Section 7.3.

7.2. Square Sectors

As introduced in Section 3.1.1 and shown in Fig. 3.1, it is assumed that sectors are square with a sector length S, which can accommodate most vehicles' lengths, i.e., most vehicles fit into a square sector S, i.e., $S > L_v$. This leads to an intersection center granularity of 4×4.

This section covers the application of the blocking pattern described in Section 7.1 to these conditions.

7.2.1. Drive Through

A vehicle driving through the intersection in any cardinal direction (i.e., N, E, S, or W) has conflicting trajectories with a few other maneuvers that may follow. In the following, directions are referred to as *clockwise*, *opposing* and *counterclockwise* ensure an analysis independent of the cardinal direction of the ego vehicle.

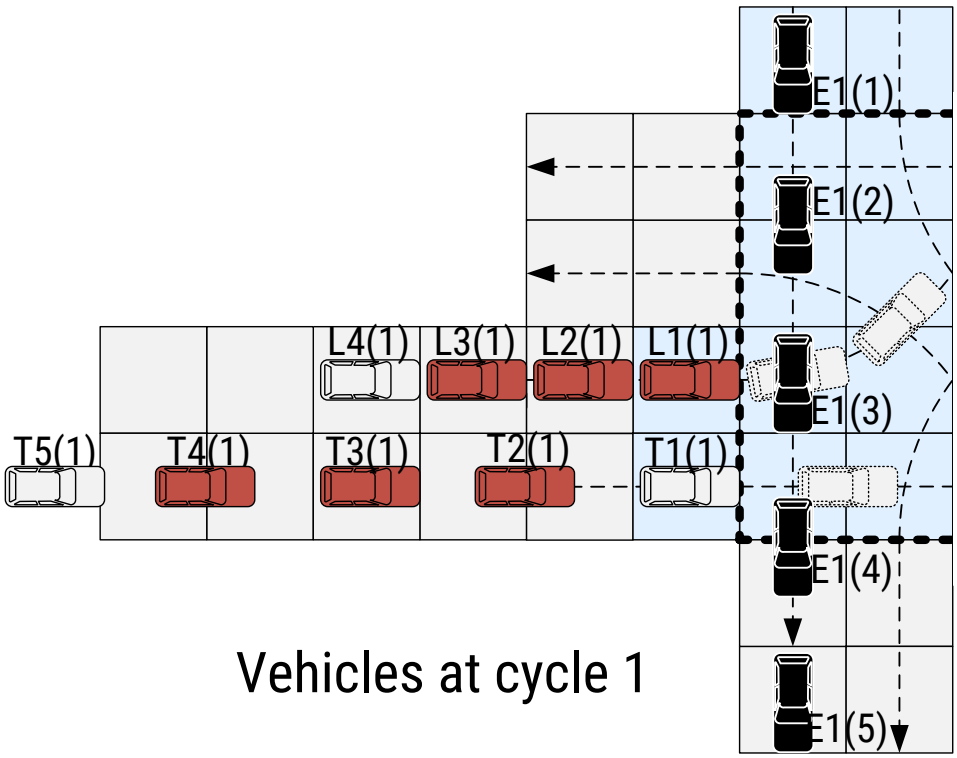


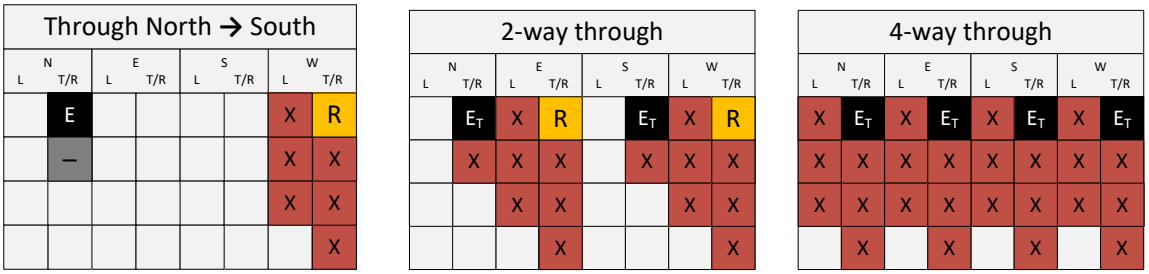
Figure 7.2.: Potential conflicts for vehicles driving through.

Revisiting the example in Fig. 7.1, a southbound vehicle driving through (on the north T/R lane) has potential conflicts with vehicles on both western (i.e., counterclockwise) lanes.

On the other hand, the opposing lanes are unaffected, i.e., a southbound vehicle driving through does not affect a northbound vehicle also driving through.

The time-dependent conflicts for the case from Fig. 7.1 are displayed in Fig. 7.2. Here, vehicles are named according to their direction, arrival cycle and current cycle. For example, $T5(1)$ denotes a vehicle driving through, which will arrive at the intersection border in cycle 5, but is displayed in cycle 1. L marks left-turning vehicles and E marks the black-colored ego vehicle, where the subscript is omitted for simplicity. Vehicles colored in red have conflicts with the ego vehicle, whereas vehicles in white are free of conflict.

As can be seen, a vehicle driving through would have collisions with vehicles driving through and arriving in cycles 2 to 4 on the counterclockwise direction, whereas vehicles driving through and arriving in cycle 1 or earlier, as well as in cycle 5 or later are free of collisions. Similarly, left turning vehicles from the counterclockwise direction may not arrive in cycles 1, 2 or 3. Vehicles arriving before or after these specific cycles have no conflict. This leads to the blocking chart displayed in Fig. 7.3.



(a) Blocking pattern of a southbound vehicle driving through.

(b) Collective blocking pattern for combinations of vehicles driving through in both two-way and four-way synchronization.

Figure 7.3.: Blocking patterns for vehicles driving through.

Here, note that there is a conditional blocking for right-turning vehicles on the counterclockwise lane, as the reduced crossing caused by the lower turn speed would reduce the inter-vehicle distance below the minimum.

The individual blocking behavior from Fig. 7.3a can be extended for two or four vehicles in Figs. 7.3b, which allows the proposed FleXS-TP to combine multiple vehicles driving through to re-create maneuvers such as two-way or four-way through driving vehicle sets, as they are considered in traditional, i.e., fully synchronized, traffic protocols such as BRIP [11] and SV-LTR [51].

Further, to avoid desynchronizing sets of vehicles, especially in the case of overlength, FleXS-TP enforces situational synchronicity in the case of vehicles driving through. This is done with a counter overseeing the previous 4 scheduled vehicles, which will trigger a rescheduling if all four vehicles are driving through on different lanes, as described in Alg. 2.

Algorithm 2: FleXS-TP enforcing situational synchronicity for vehicles driving through.

```
   Result: Enforced synchronicity for vehicles driving through
   /* On new vehicle schedule                                          */
 1 if (Veh_route == through) and (origin still available) then
 2 |   if (through_count==0) then
 3 |   |   snapshot scheduling state;
 4 |   |   mark vehicle origin as unavailable
 5 |   |   continue; /* Next vehicle                                   */
 6 |   end
 7 |   if (through_count>3) then
 8 |   |   rollback last 4 vehicles
 9 |   |   determine shared starting point
10 |   |   schedule vehicles together
11 |   |   through_count = 0; continue; /* Reset counter              */
12 |   else
13 |   |   through_count ++;
14 |   |   mark vehicle origin as unavailable
15 |   end
16 else
17 |   through_count = 0;
18 |   continue; /* Next vehicle                                       */
19 end
```

7.2.2. Left Turns

The time-dependent conflicts of a vehicle turning left from north to east, as displayed in Fig. 7.4, affect all 3 remaining directions. Similar to before, $T_2(1)$ displays a vehicle driving through which will reach the intersection border in cycle 2 and is displayed in cycle 1, whereas $L5(1)$ denotes a vehicle turning left, displayed in cycle 1, which will arrive at the intersection border in cycle 5. Vehicles colored red, yellow or blue have conflicts with the ego vehicle, but white vehicles are free of conflict.

Specifically, left turns from the counterclockwise direction may not arrive in the same cycle as the ego vehicle, unless they are sufficiently short to allow for four-way synchronization, i.e., $L_i < 0.7S$ [51], as described in Section 3.1.4. This conditional blocking is marked in yellow with the label „L“. The same applies to vehicles making a left turn from the clockwise direction when arriving in cycle 1. Vehicles arriving in cycles 2 and 3, however, are blocked.

Vehicles driving through from the clockwise direction are only blocked when arriving in the same cycle as the ego vehicle. However, since their lane is shared between vehicles driving through and this conflict only applies to vehicles driving through and not those turning right, this is specifically marked in blue with the label „T“. That is, only vehicles driving through are blocked, but not those turning right. Similarly, vehicles driving through from the opposing direction have conflicts with the ego vehicle when arriving in cycles 2 to 4.

Again, the sum of these individual effects by a vehicle making a left turn leads to its individual and combined blocking pattern as shown in Fig. 7.5a.

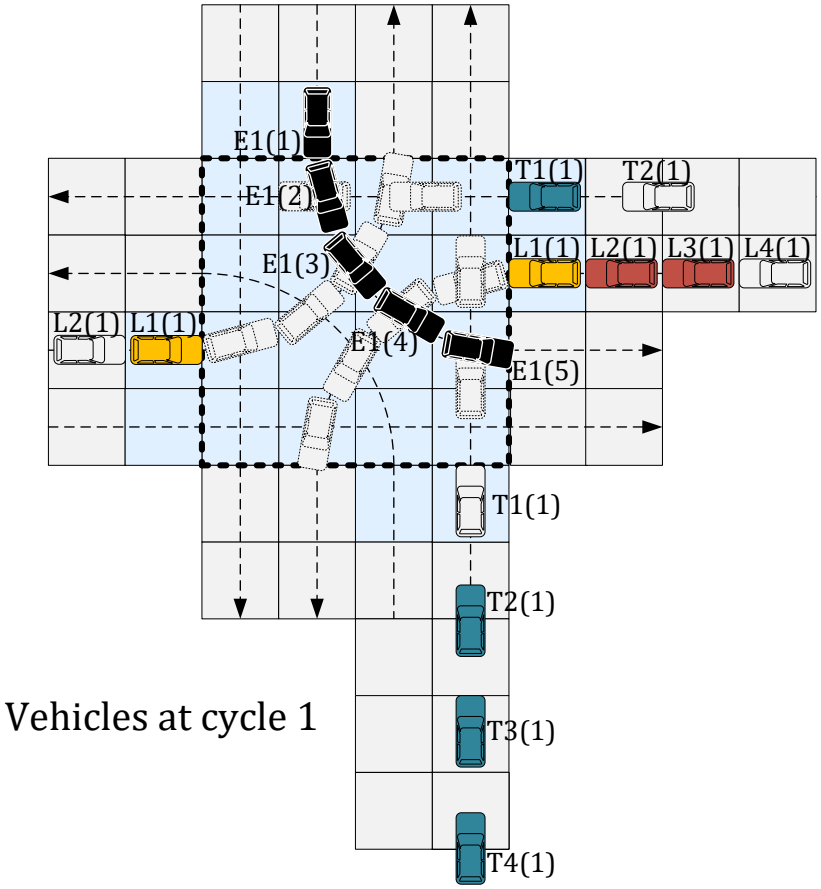
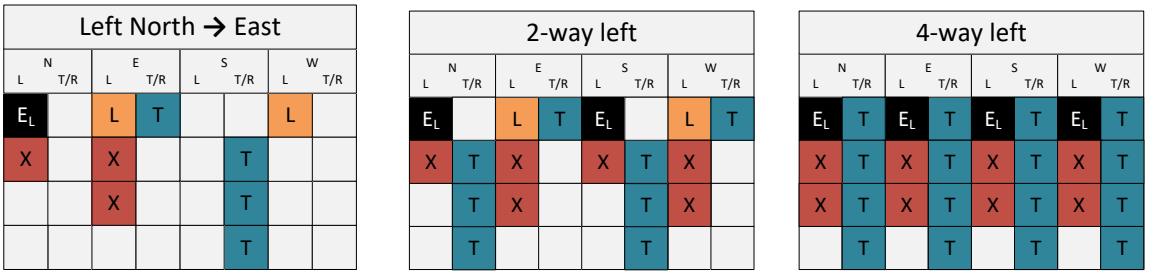


Figure 7.4.: Potential conflicts for vehicles turning left.

As described before, the orange squares highlight a blocking, only if one or more of the vehicles involved do not fulfill the four-way condition of $L_i < 0.7S$ [51]. The blue squares signal a blocking for vehicles driving through only, i.e., right turns are not affected.

Similar to before, this allows the proposed FleXS-TP to re-create the fully synchronized sets



(a) Blocking pattern of an east-bound vehicle turning left.

(b) Collective blocking pattern for vehicles turning left (two-way and four-way synchronization).

Figure 7.5.: Blocking patterns for left turns.

of two-way and four-way synchronization as described in traditional STPs such as BRIP [11] and SV-LTR [51], as shown in Fig. 7.5b. Note again that a four-way synchronization is only possible, if all vehicles are shorter than $L_i < 0.7S$, i.e., vehicles can be scheduled on orange squares.

7.2.3. Right Turns

Vehicles making a right turn only conflict with vehicles driving through on the clockwise lane. However, similar to before, for a conflict to occur, those vehicles would have to arrive before the ego vehicle. Therefore, their personal blocking pattern would prevent the ego-vehicle in the first place. This allows for very simple individual and collective blocking patterns for right turns as shown in Fig. 7.6.

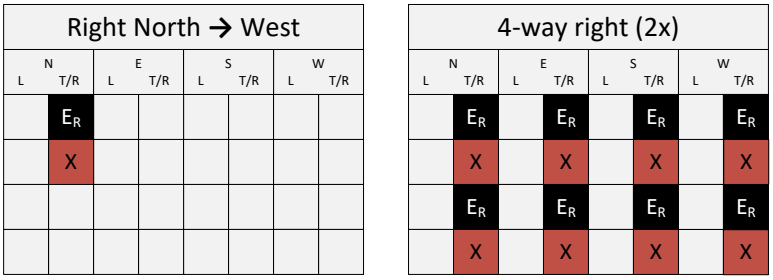


Figure 7.6.: Blocking pattern of a single right turn as well as collective blocking pattern for two sets of four-way synchronized right turns.

7.2.4. Overlength Vehicles

All previously derived blocking patterns rely on the assumption that vehicles fit into one sector, i.e., $L_i \leq S$ for all i. While this should be the case for the majority of vehicles, exceptionally larges vehicle such as buses and trucks can have lengths of up to 10 meters. Since these vehicles are the exception and not the norm, handling them as a special case is beneficial compared to handling them as the default case.

For ease of exposition, this section only covers blocking pattern modifications for vehicles with a length excess of $1S$, i.e., for vehicles with $2S > L_i > S$. However, a greater length excess can be incorporated in the same fashion. Now, the direct impact of overlength vehicles as considered above is that they occupy the arrival zone and the intersection center for an additional cycle. This adds additional *blocking* on all conflicting lanes. For the previously described individual blocking patterns, the impact of one sector of overlength penalty can be seen in Fig. 7.7.

For right turns, it is worth noting that a length excess would reduce the inter-vehicle distance to a vehicle driving through on the clockwise lane to below a critical threshold, which is reason why that lane has to be delayed by $1C$ for overlength ego vehicles making right turns to protect vehicles on the clockwise lane.

As previously mentioned, this version of FleXS-TP and all its blocking charts assume square sectors. The next section extends this to variable sector length, using rectangular sectors.

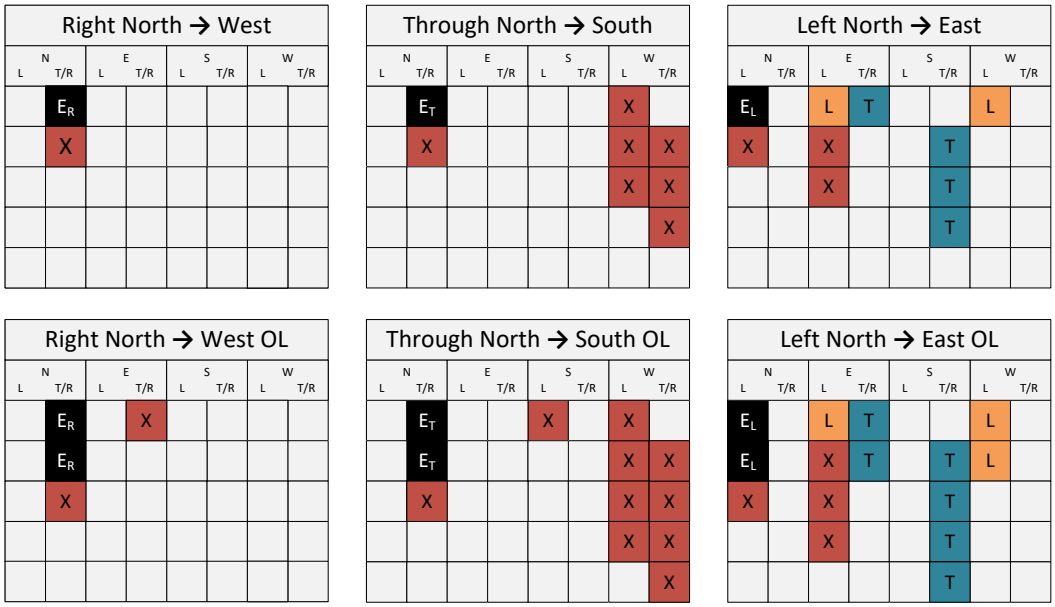
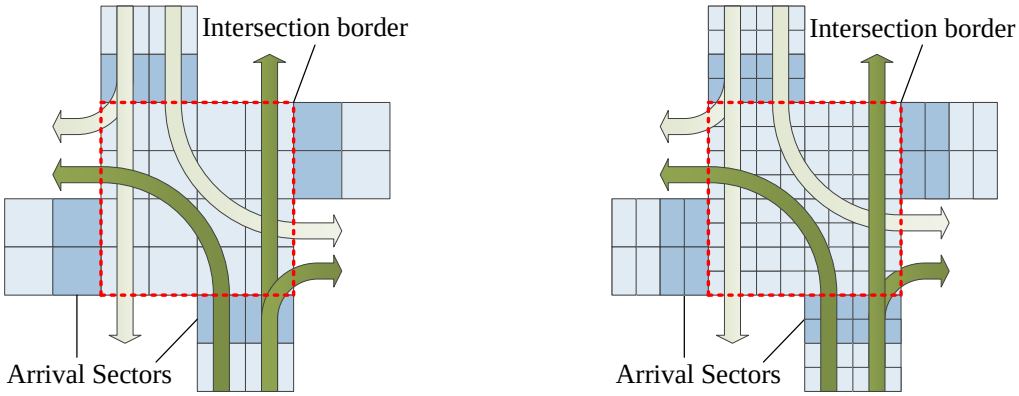


Figure 7.7.: Blocking patterns for overlength vehicles, including right turns (left side), driving through (center) and left turns (right side).

7.3. Variable Sector Length

To extend the previously described version of FleXS-TP for variable sector length, this section assumes that sectors that are only half as long as they are wide, see Fig. 7.8. This leads to an effective intersection center granularity of 8×8.



(a) Square sectors.

(b) Variable sectors.

Figure 7.8.: Intersection layout comparison.

This modifies the sector composition previously shown in Fig. 5.1 by adding variable sectors with $S_W = S$ and $S_L = S/2$, as visualized below in Fig. 7.9.

Considering again the blocking chart example from Fig. 7.1, these modifications lead to the blocking chart for the same maneuver when considering variable sectors displayed in Fig. 7.10.

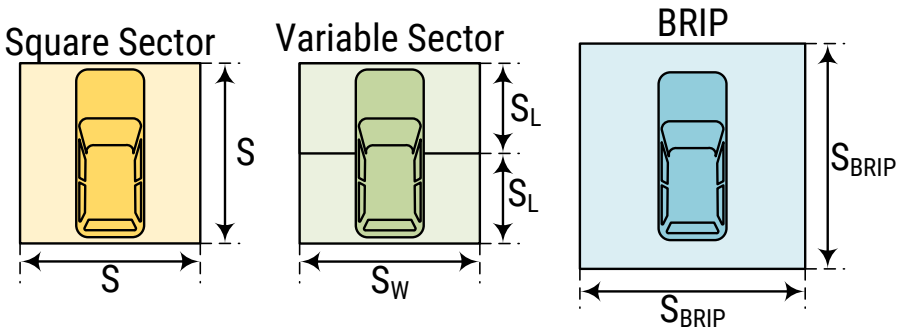


Figure 7.9.: Comparison of square sector, variable sector and BRIP sector for the same vehicle.

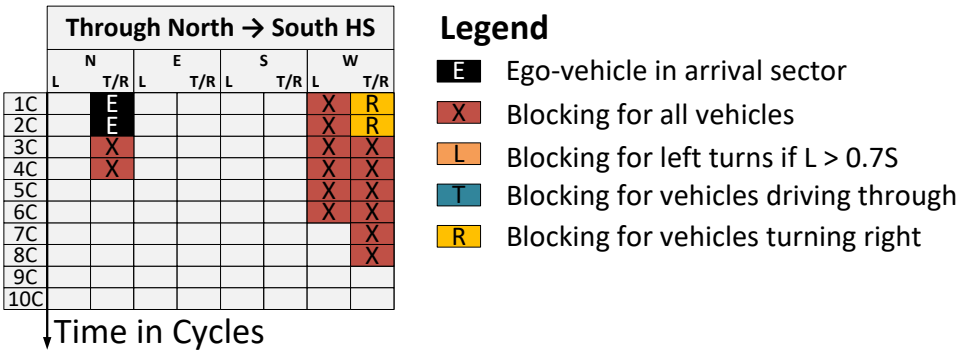


Figure 7.10.: Legend from Fig. 7.1 modified for variable sector lengths.

In this case, due to the reduced medial dimension S_L of the sector, the length of a cycle is also reduced by half since $C_{Half} = S/V_{LO}$, essentially leaving this specific maneuver functionally identical between the two sector compositions. However, the increase in granularity allows for more efficient handling of overlength and increase inter-vehicle distances in general, as they are dependent on the sector size. All patterns including their overlength variants can be seen next in Fig. 7.11.

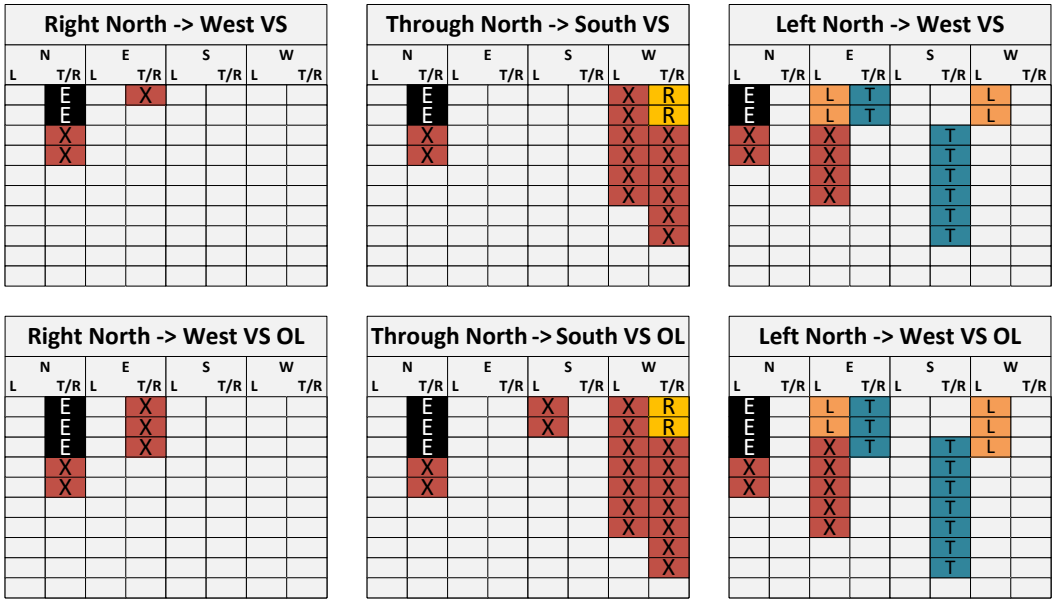


Figure 7.11.: All maneuvers and their overlength variants modified variable sector lengths.

Chapter 8.

Simulations and Algorithms

Several approaches and concepts of this thesis have been implemented or visualized as simulations in SUMo/Python and C#. This chapter covers these individual implementations in detail.

8.1. Simulation of Urban Mobility (SUMo)

Simulation of Urban Mobility (SUMo) [41] [46][1] is a well-accepted open source tool for traffic simulation which can simulate realistic intersections. It can be interfaced from Python via the Traffic Control Interface (TraCI)[2].

To run a simulation in SUMo, the following components in Extensible Markup Language (XML) format are required: i) a network file containing the road layout (in this case, following Fig. 3.1), ii) a route file, containing possible routes and vehicles and iii) a configuration file, linking these files to the execution of SUMo.

Network File: The network file (or *net file*) is created and edited in a visual network editor included in the SUMo distribution named *netedit*[3], see below in Fig. 8.1.

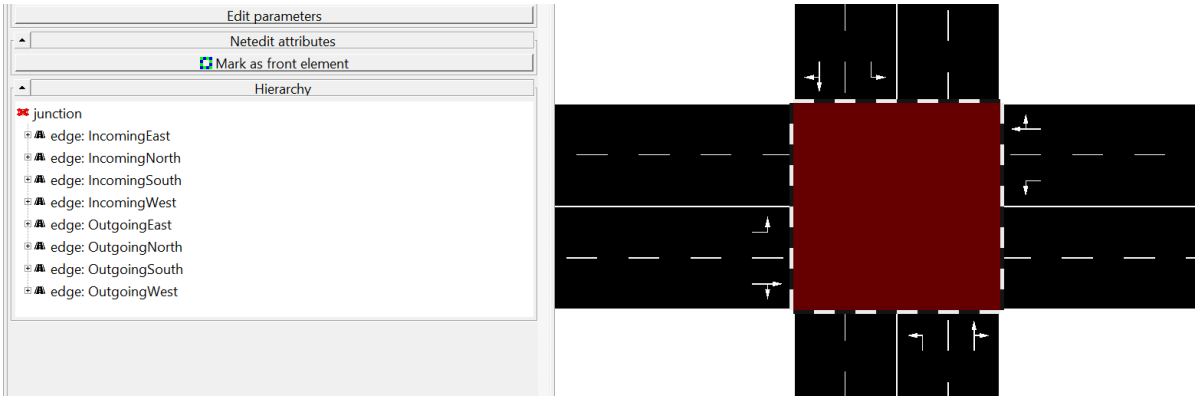


Figure 8.1.: netedit view of network file and edge definitions for layout from Fig. 3.1.

Each of the edges shown above is further divided into lane 0 (left turn lane) and lane 1 (right/through lane). Then, each junction (such as the center) acts as a connector between incoming and outgoing edges, such that different connections are created, see Fig 8.2.

[1] `https://sumo.dlr.de/docs/`
[2] `https://sumo.dlr.de/docs/TraCI/Interfacing_TraCI_from_Python.html`
[3] `https://sumo.dlr.de/docs/Netedit/index.html`

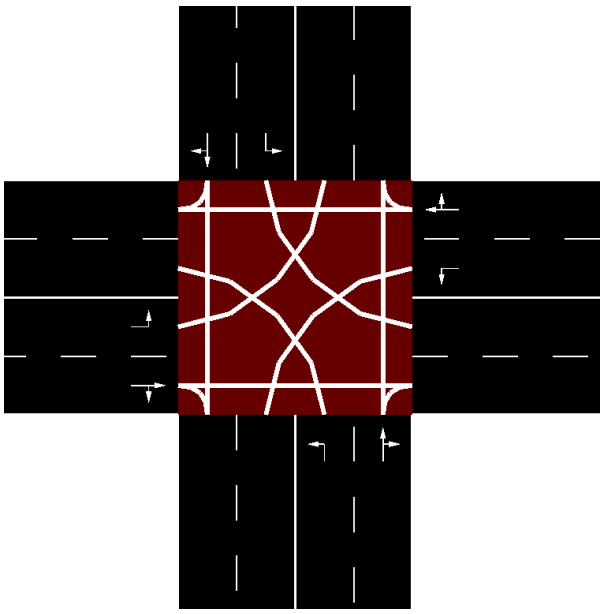


Figure 8.2.: netedit view of connections in the central junction.

These connections across the central junction are then used in the route file to define the different trajectories for the vehicles.

Route File: The route file fulfills 3 functions: i) Defining vehicle types, ii) defining routes and iii) defining vehicular traffic.

Vehicle type definitions contain, among other parameters, basic vehicle information such as acceleration/deceleration behavior, length and maximum speed. For example, to include a vehicle type called "scarVHI" with a maximum acceleration and deceleration of $a = \pm 5\ m/s^2$, a length of $L = 5\ m$ and a maximum speed of $v_{max} = 12.5\ m/s = 45\ km/h$ in the shape of a van, the following line is included in the route file:

```
<vType id="scarVHI" accel="5" decel="5"  length="5.0"
       maxSpeed="12.5" guiShape="passenger/van"/>
```

To define the possible routes a vehicle can take, the existing edges show in Fig. 8.1 are connected in this route file. For example, having vehicles drive through from north to south is enabled by creating the following route:

```
<route id="n2s" edges="IncomingNorth OutgoingSouth"/>
```

Finally, the route file can contain specific traffic for the simulation, as vehicles can be inserted into the simulation by using the previously defined vehicle types, routes and departure information such as departure time and speed. To have a vehicle of type "scarVHI" enter the simulation with the intention to go from north to south on the rightmost lane 20 seconds into the simulation, the following command is used:

```
<vehicle id="V1_n2s_L1" departLane="1" type="scarVHI"
         route="n2s" depart="20" departSpeed="12.5"/>
```

Configuration File: The SUMo configuration file (*.sumocfg*) connects the previous files to a running simulation and also specifies other constraints such as timing. To connect the previously discussed files to a simulation, the input-section of the configuration file simply references their local path.

```
<input>
    <net-file value="networkfile.net.xml"/>
    <route-files value="routefile.rou.xml"/>
</input>
```

This configuration file also defines the time resolution, which is $\Delta t = 0.05\ s$ for all simulations. It can then be started to execute a basic simulation in the defined setting. This can be used to implement the approaches proposed and discussed in this thesis by i) generating a parameterized, randomized traffic set and ii) executing the protocol logic on this traffic set to achieve efficient and collision-free crossing. These two aspects of the simulations are covered next.

8.1.1. Generation of Well-Behaved and Randomized Traffic

There are two types of traffic generation for simulations in this thesis — *well-behaved* and *randomized*, 1000 vehicles at a time.

Well-behaved traffic arrives exactly as required from the traffic protocol, such that all potential slots are filled and the maximum throughput at the intersection can be simulated. Due to the diminished role of right turns, well behaved traffic is generated in one of two forms — either only vehicles driving through (effectively replicating BRIP Type I) or only vehicles making left turns. Well-behaved traffic acts as an upper bound of traffic compliance, representing a best-case scenario of vehicles arriving exactly as required by the protocol to achieve seamless, maximum throughput. While their application in real-world settings is limited due to the heterogeneous nature of realistic traffic, having a protocol perform well under both well-behaved and randomized traffic is a key goal of this thesis.

As for randomized traffic, this is generated with the following constraints:

- Uniformly randomized vehicle type out of 5 different types (see Table 8.1).
- Uniformly randomized route out of 12 (4 origins with 3 destinations each).
- Uniformly randomized spacing on the same lane of 2 C to 7 C.[4]

The 5 different vehicle types used in the simulations are displayed in Table 8.1. Further, the different routes represent the 12 different directions a vehicle can go, starting from one of the four origin points and going to one of three possible destinations (U-turns are not considered). Their explicit route file definitions can be seen in Listing A.1.

Once all 1000 vehicles are generated, their information is saved to the route file such that they are automatically included in the next run of the simulation. This route file can then be used for the different traffic protocols to show their different performance on the same traffic set to ensure comparability.

[4] As the vehicle generation requires a entering time *depart*, vehicles are spaced in time. This can be translated to distance as $1\ C = S/V_{LO}$.

Name	Model	Length
scar	van	3 m
realcar	default	5 m
ecar	wagon	5 m
bcar	bus	5 m
tcar	truck	8 m

Table 8.1.: Different vehicles types used in the simulations.

Vehicle Insertion Offset: Vehicles are inserted at their departure time with their full length already on the road, i.e., if a vehicle with $L = 5\ m$ is inserted at a distance of 200 m from the intersection border, its distance to the intersection border after insertion will be $200\ m - 5\ m = 195\ m$. This offset has to be considered when determining a vehicle's arrival time at the intersection border.

This randomized traffic set is then saved as „unsorted“ and acts as the input for all other protocols in the simulation. The source code for the generation of randomized traffic and the unsorted route file can be seen in Listing A.2. This unsorted route file is further simulated directly in the simulation type called "without contention", as described next.

8.1.2. Simulation of Unsorted Traffic without Contention

When investigating different traffic protocols with regards to throughput, the resulting values must be put into context by what effectively is the maximum possible throughput. For any given randomized traffic set, this is described by the traffic running *without contention*, i.e., the vehicle travel through the intersection at the maximum speed with no regard for collisions or safety. While this is not a practical simulation that is compatible with any real-world setting, it provides the absolute upper bound for the execution time of this specific traffic set and, as such, for the throughput of the protocol — it is physically impossible to have vehicles traverse the intersection faster (provided they adhere to the speed limits).

However, due to the collisions this simulation will inadvertently contain, several SUMo-based safeguards have been deactivated such that the vehicles continue traversing the intersection with the maximum allowed speed in all cases. This will also be the case for the simulations of FleXS-TP and BRIP, as described later.

These safeguards are deactivated separately for each possible vehicle type in the route file and are subsequently documented in Table 8.2.

Parameter	Set to	Description
minGap	1	Empty space after leader
InsertionChecks	none	Safety checks during vehicle insertion
collisionMinGapFactor	0	Min. fraction of minGap required to leader
SpeedDev	0	Deviation from speed
Sigma	0	Driver imperfection
SpeedMode	32	All safety checks off

Table 8.2.: SUMo safeguards disabled to simulate unsorted traffic without contention.

8.1.3. Conventional Traffic Lights

To simulate the randomly generated traffic with conventional traffic lights, a traffic light system is added to the intersection center.

For so called „4-arm intersections“, traffic light programs in SUMo are generated with 4 individual phases with a fixed cycle time of 90 *s*. The 4 phases are: i) a straight phase, ii) a left-turning phase (only if there is a dedicated left-turn lane), iii) a straight phase for the direction orthogonal to the first one and iv) a left-turning phase for the direction orthogonal to the first one (only if there is a dedicated left-turn lane).[5]

This leads to the following description inside the network file:

```
<tlLogic id="Center" type="static" programID="0" offset="0">
   <phase duration="33" state="GGgrrrGGgrrr"/>
   <phase duration="3"  state="yygrrryygrrr"/>
   <phase duration="6"  state="rrGrrrrrGrrr"/>
   <phase duration="3"  state="rryrrrrryrrr"/>
   <phase duration="33" state="rrrGGgrrrGGg"/>
   <phase duration="3"  state="rrryygrrryyg"/>
   <phase duration="6"  state="rrrrrGrrrrrG"/>
   <phase duration="3"  state="rrrrryrrrrry"/>
</tlLogic>
```

This is also displayed as a timeline within netedit, as shown in Fig. 8.3.

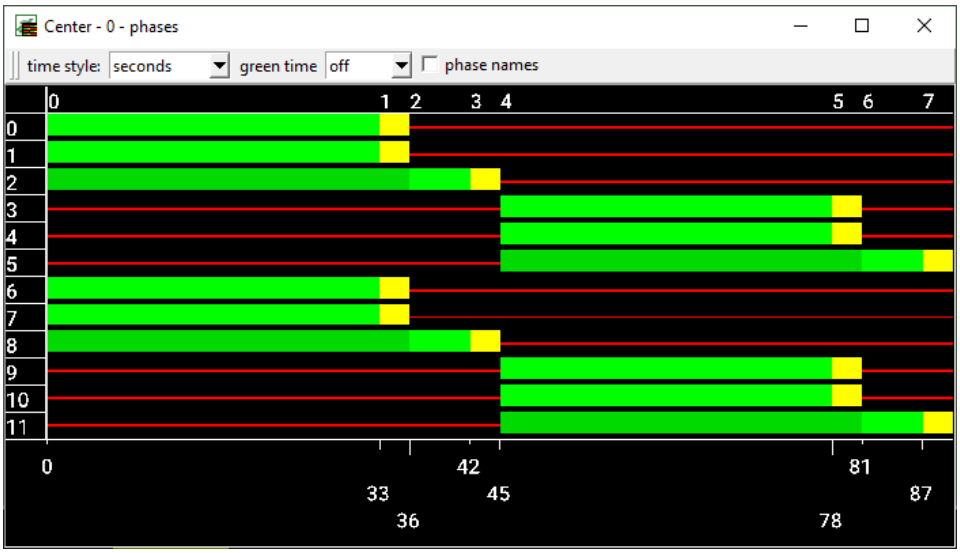


Figure 8.3.: Traffic light phases visualized in netedit.

Here, „G“ (bright green) is priority green, „g“ (darker green) is non-priority green, „y“ is yellow and „r“ is red. Further, the states are ordered lane by lane from right to left, i.e., the order of states is: 0 — north right, 1 — north through, 2 — north left, 3 — east right and so on.

Once this traffic light is added to the intersection center via netedit, the unsorted route file can again be executed, although with the safety mechanisms of Table 8.2 re-enabled.

[5] https://sumo.dlr.de/docs/Simulation/Traffic_Lights.html

8.1.4. Simulation of BRIP

In this thesis, BRIP type I is replicated through simulation to show the general behavior of BRIP by generating well-behaved traffic exactly matching the protocol. The different BRIP types are shown in Fig. 5.2.

This simulation has two variants, with $L = 5\ m$ and $L = 10\ m$. Here, only a single vehicle type is used to simulate the homogeneous traffic assumed by BRIP. These vehicles are then inserted on all lanes with a distance of S_{BRIP}. The source code for the route file generation for BRIP Type I can be seen in Listing A.3.

Due to the BRIP-typical, extremely tight safety distances between vehicles, it is also required to deactivate the SUMo-based safeguards described before in Table 8.2 for the simulation to run as intended.

With these safeguards disabled, BRIP Type I can be simulated in SUMo, see Fig. 8.4.

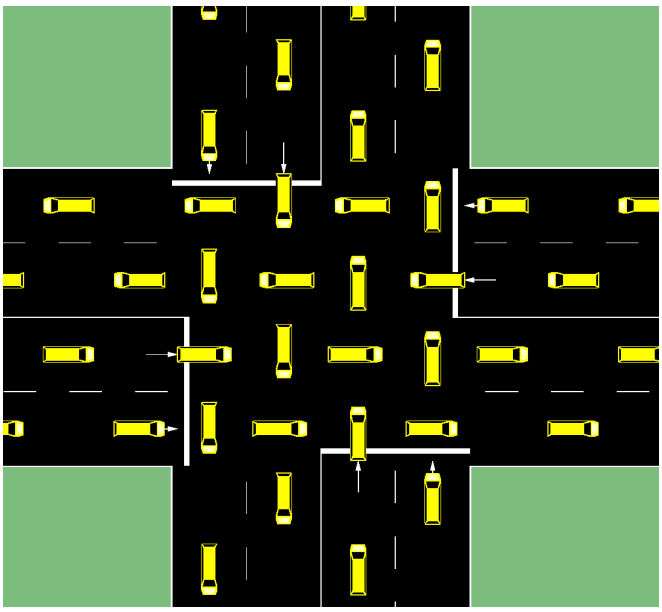


Figure 8.4.: Simulation of BRIP Type I.

8.1.5. Simulation of SV-LTR

Similar to BRIP, the behavior of SV-LTR is replicated through simulation. However, whereas BRIP requires well-behaved traffic exactly generated to match the protocol, SV-LTR can also accommodate randomized traffic. As such, this simulation receives the randomized traffic generated in Section 8.1.1 as input and then delays vehicles accordingly such that they arrive at the intersection precisely when they are scheduled to to cross safely without stopping, by sorting them into sets matching the rigid, pre-determined patterns characteristic for synchronous traffic protocols. The step-by-step process is documented in Algorithm 3.

Once this list of sets is known, the time difference between each vehicles intended arrival time and the actual planned execution time of its vehicle set can be determined — effectively representing the delay this specific vehicle experiences by the traffic protocol.

Algorithm 3: SV-LTR generating a set list from unsorted traffic.

```
Result: Set List generated from unsorted traffic
for i = 0 to NumberOfVehicles do
    if (setCount < 1) /* Manually add first set                          */
     then
        Create(NewSet(Veh[i])); SetList.append(NewSet); setCount++;
    else
        /* Does the vehicle match that set?                              */
        if Compatible(Veh[i], SetList[SetCount-1]); /* Check previous set */
         then
            Set[SetCount-1].append(Veh[i]); /* Assign to set             */
        else
            if setCount >= 2 /* Look back 2 sets, if that many exist     */
             then
                /* Does the vehicle match that set?                      */
                if Compatible(Veh[i], SetList[SetCount-2]); then
                    Set[SetCount-2].append(Veh[i]); /* Assign to set     */
                else
                    /* No match - create new set                         */
                    Create(NewSet(Veh[i])); SetList.append(NewSet); setCount++;
                end
            else
                /* No match - create new set                             */
                Create(NewSet(Veh[i])); SetList.append(NewSet); setCount++;
            end
        end
    end
end
```

8.1.6. Simulation of FleXS-TP

FleXS-TP follows the a similar strategy as SV-LTR when it comes to the simulation replicating its behavior. It also uses the randomized traffic generated in Section 8.1.1 as input and using this information to assign vehicles into the blocking chart. As before, the time difference between the originally intended arrival time and the planned crossing time following the blocking chart can be used to describe the delay incurred by the traffic protocol. Additional, as before with BRIP, due to the extremely short inter-vehicle distances, deactivating the SUMo-based safeguards described before in Table 8.2 is required as any kind of emergency braking will lead to desynchronization.

As before, the step-by-step process is documented in Algorithm 4. Note that this behavior is identical for variable sectors as described in Section 7.3, as these differences are introduced through the different blocking patterns assigned to the individual maneuvers.

For well-behaved traffic, i.e., traffic arriving as described in Section 8.1.1 with vehicles either only driving through or making left turns, without overlength vehicles, the behavior of FleXS-TP exactly mirrors that of SV-LTR. This is further discussed in detail in Section 9.4.

Algorithm 4: FleXS-TP generating a filled blocking chart from unsorted traffic.

```
   Result: Filled blocking chart generated from unsorted traffic
 1 BlockingChart.Init(); /* Set all to 0 / empty                              */
 2 for i = 0 to NumberOfVehicles do
 3 |   lane = getLane(Veh[i]); /* Corresponds to row in blocking chart        */
 4 |   index = BlockingChart.NextFreeTimeOnLane(lane); /* Next free spot      */
 5 |   route = Veh[i].route; /* Maneuver - left, through, right               */
 6 |   BP = getBlockingPattern(lane, route);/* Blocking pattern for maneuver   */
 7 |   iterator = 0;
 8 |   while true do
 9 |   |   time = index + iterator; /* Iterate along the chart               */
10 |   |   if (BlockingChart.Compatible(BP, time) /* Maneuver compatible?     */
11 |   |    then
12 |   |   |   BlockingChart.Add(Veh[i], BP, time); /* Add to chart          */
13 |   |   |   break; /* Break loop, next vehicle                            */
14 |   |   end
15 |   |   iterator++; /* Check next slot                                     */
16 |   end
17 end
```

8.2. Performance of Vehicle Count Estimation Algorithm

Algorithm 5 details the individual steps of the vehicle count estimation described in Chapter 4, which is implemented in C#.

As described previously, a valid sequence at the intersection consists of n_{min} to n_{max} sets which fulfill (4.15). Therefore, after initializing probability values (see line 1 to 3), all possible sequences out of $n \in [n_{min}, n_{max}]$ sets (see lines 4 to 5) are determined, where each of the n sets performs one of the k driving combinations shown in Table 9.2 (requiring a given number of cycles and, hence a given inter-vehicle separation).

Each of these sequences is then tested for validity, following (4.15), 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 the next sequence is considered (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 9.3.

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 9.2), 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 intersection itself, having a more complex setting can greatly increase the number of combinations k and therefore the runtime of the algorithm. Further, the performance also greatly depends on the sector size S due to its impact in the calculation of n_{min} and n_{max}

Algorithm 5: Calculating vehicle count estimates

Result: Vehicle estimates c with corresponding probabilities p_c as in Table 9.3

```
for c = 0 to c_max do
    p_c(c) = 0
end
for n = n_min to n_max do
    forall different sequences with n sets do
        if sequence is valid as per (4.15) then
            get sequence vehicle count c_seq;
            get sequence probability p_seq as per (4.17);
            p_c(c_seq) = p_c(c_seq) + p_seq;
        else
            continue;
        end
    end
end
```

in (4.12) and (4.11) — having a reduced n_{max} also reduces the runtime substantially. On the 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 for $S = 5\ m$ ($n_{max} = 15$) to below 100 ms for $S = 8\ m$ ($n_{max} = 10$), see Table 8.3. Changing other parameters such as turn direction, vehicle absence, etc. had no impact on the runtime of the algorithm.

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.

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

Table 8.3.: Estimation algorithm runtime in relation to sector size S.

Chapter 9.

Evaluation

This chapter presents a comprehensive evaluation of the techniques for the design of intelligent intersections proposed in Chapters 4 to 7.

9.1. Probabilistic Vehicle Count Estimation and Communication Reliability

This section provides results and evaluations for the probabilistic vehicle count estimations and their impact on communication reliability for the one-lane, two-way synchronization intersection proposed in Chapter 4 when considering variable vehicle lengths, traffic density and traffic direction.[1]

9.1.1. Probabilistic Vehicle Count Estimation

In Section 4.1, Table 4.1 detailed the cycle cost g for different turn combinations, with and without overlength.

The first step to obtain a probabilistic vehicle count estimation is to determine the occurrence probability each of the different turn combinations. To this end, the following three primary factors are required: i) vehicle length distribution, ii) traffic directions and iii) traffic density, which are obtained next. Note that the values in this chapter are examples chosen for illustrative purposes only — the subsequent analysis remains valid for other values as well.

The vehicle **length distribution** is obtained as detailed in Section 4.1 via statistical sales data in Europe [29] [30] [71] and described by the probability mass function $f(\cdot)$ from Eq. (4.3) and is displayed in Fig. 4.2. As previously described, these values were clustered into 8 categories, which are displayed below in Table 9.1

Length [m]	1.800	2.695	3.775	4.141	4.285	4.536	4.617	7.820
Probability	0.122	0.005	0.092	0.280	0.332	0.119	0.025	0.025

Table 9.1.: Vehicle lengths and their occurrence probabilities derived from the data of Fig. 4.2.

Determining the probabilities of different **traffic directions** can be achieved via prediction algorithms such general traffic forecasting [85] [86] or be derived from statistical data, e.g., by observing the traffic flow for a sufficiently long period of time [80]. In this chapter, these

[1] For the sake of completeness, a short evaluation for the impact of only variable vehicle length as discussed in Section 4.1 can be found in Section A.4.

probabilities were set to $p_L = 0.3$, $p_R = 0.1$ and $p_T = 0.6$ for right turns, left turns and driving through respectively.

The overall probability of a vehicle combination can be calculated by multiplying the individual turn probabilities, assuming the vehicles operate independently. For clarity, each combination is labeled according to the movement direction of both vehicles — Left (L), Through (T), or Right (R). For example, TT represents both vehicles proceeding straight through the intersection, while TL indicates one vehicle going through and the other turning left.

As an illustration, the probability of the TT combination (both vehicles going straight) is calculated as $p_D = p_T \cdot p_T = 0.36$. In cases involving mirrored combinations — such as LT and TL, or RL and LR — both permutations must be accounted for. For example, the probability of either **TL** or **LT** occurring is $p_D = p_T \cdot p_R + p_R \cdot p_T = 0.12$.

This allows for an estimation of the likeliness with which a certain cycle cost will occur and thus, allows for conclusions regarding the maximum number of vehicles.

Next, this is further extended through the concept of **traffic density** via the probability p_A of a vehicle to be *absent*, which in turn depends on the traffic ratio r, see Eqs. (4.6) and (4.7). With r given, the probabilities for empty, half and full sets can be determined via Eqs. (4.8), (4.9) and (4.10) respectively. As described in Section 4.3, if both vehicles of a set are absent, an *empty* set with a cycle cost of $g = 2$ is executed.

While the traffic ratio r, representing intersection saturation, ranges from 0 to 1, assuming a high value (e.g., $r = 0.9$) enables evaluation under near-saturated conditions while retaining a nonzero probability of vehicle absence. Accordingly, pA is set to 0.1 for this analysis.

Now that all three primary factors have been obtained, probabilities can be assigned to the different combinations in Table 4.1. Specifically, for $p_A = 0.1$, $p_L = 0.3$, $p_T = 0.6$, $p_R = 0.1$, $p_{OL} = 0.025$ and $R = 150\ m$, the cumulative probabilities for different cycle costs are displayed in Table 9.2.

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 9.2.: Exemplary cycle cost probabilities.

Assuming an intersection range of $R = 150\ m$ and a sector size of $S = 5\ m$, the minimum and maximum number of sets following Eqs. (4.13) and (4.14) at the intersection ranges between $n_{min} = 5$ and $n_{max} = 15$. 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 following Eq. 4.15 with $n \in [5, 15]$, the different vehicle counts c with their corresponding probabilities p_c can be estimated, as shown in table 9.3.

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

Table 9.3.: Resulting maximum number of vehicles with their corresponding probabilities p_c for $p_A = 0.1$, $p_L = 0.3$, $p_T = 0.6$, $p_R = 0.1$, $R = 150\ m$ and $S = 5\ m$.

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 number of vehicles c at the intersection is less than 4.12×10^{-7}.

Next, the impact of drive direction probabilities, absent vehicles and sector size (i.e., overlength penalty) on the probabilistic estimate of c is evaluated. If not specified otherwise, parameters are set to $p_A = 0.1$, $p_L = 0.3$, $p_T = 0.6$, $p_R = 0.1$, $R = 150\ m$ and $S = 5\ m$.

Regarding Drive direction: As discussed before and shown in Table 9.2, different combinations of drive directions have different cycle costs and, therefore, affect the estimated maximum number of vehicles within the intersection. 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.

If $p_T = 0.6$ remains constant, but the remaining $p_L + p_R = 0.4$ is shifted between left and right turns, the discussed behavior can be observed in Fig. 9.1.

That is, the more right turns there are, the higher the chance of having high vehicle counts. Additionally, 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 when $p_A = 0$, (i.e., having less than 12 vehicles only possible if vehicles are absent). 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 (as indicated by the purple line in Fig. 9.1). 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 9.3.

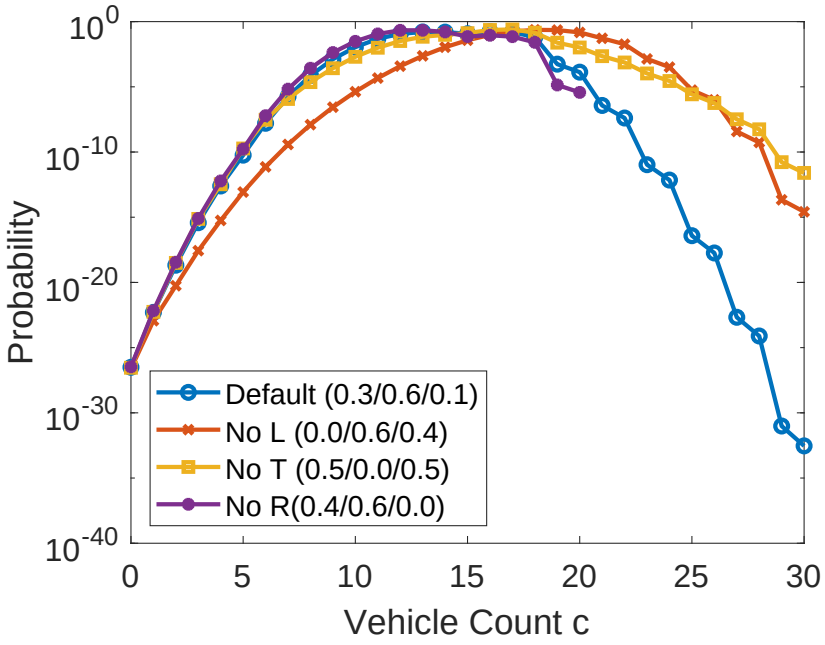


Figure 9.1.: Impact of drive direction (brackets denote $(p_L/p_T/p_R)$).

Vehicle density: The impact of variable traffic density, i.e., different probabilities p_A of vehicles being absent, is depicted in Fig. 9.2. 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.

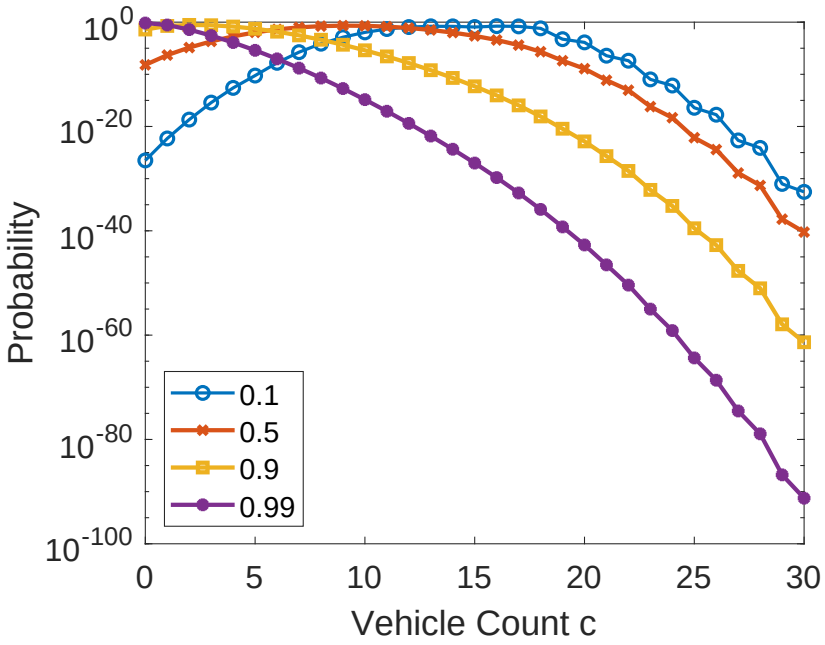


Figure 9.2.: Impact of vehicles being absent p_A.

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.

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, all cycle costs are given in multiples of S and therefore increase for larger S. Following Eq. (4.11), this also reduces the maximum vehicle count as the number of sets is inversely proportions to the sector size S. This is clearly visible in Fig. 9.3, where higher vehicle counts c are not possible for settings with higher sector size S — their graphs end abruptly.

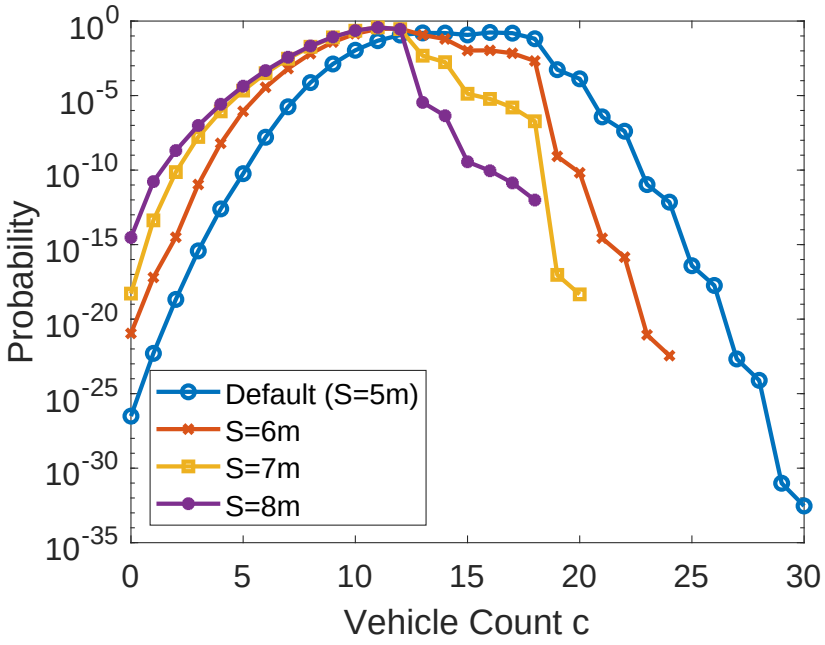


Figure 9.3.: Impact of sector size S.

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., $S = 5\ m$ to $S = 6\ m$.

9.1.2. Impact on Communication Reliability

This section examines the impact of probabilistic vehicle estimates on the communication reliability of the communication scheme introduced in Section 4.3. To this end, simulations based on the OMNeT++ simulation framework [82] were performed, which allowed the recording of statistical data for a very large number of transmissions — for each of the presented curves in this section, at least 100,000 communication cycles were simulated. Channel models and parameters were taken from [88] and while it was assumed that there was no external interference present, this can be easily added as described in [65]. Note that $c_{max} = 30$ was selected for the deterministic approach, since this represents the worst-case number of vehicles at the intersection as per Eq. (4.14). For the proposed estimate, $c_{prop} = 20$ was selected in the fol-

lowing experiments, which is a reasonable safe value, i.e., the chance of having a higher c is very small with $\approx 4.12 \times 10^{-7}$.

To analyze communication reliability, the weighted probability p_c of different vehicle counts c can now be incorporated into the communication reliability discussed in Eq. (4.19) in Section 4.3.1, which yields the following:

$$\bar{p} = \sum_{n=n_{min}}^{n_{max}} p_n \left(1 - \left(\frac{2(n-1)l_{req}}{t_{max} - t_{min}}\right)^x\right).$$

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 9.3. Further, c_{min} and c_{max} are the lower and upper bound of the vehicle estimate as per Eqs. (4.13) and (4.14). Note that for the evaluation presented in this section, it is assumed that $l_{req} = 80$ μs, which corresponds to 30 bytes payload at 6 Mbit/s — see also [54]. Further, in order to optimize reliability, x is set to $x = 3$ as per [54] 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. 4.4 [65].

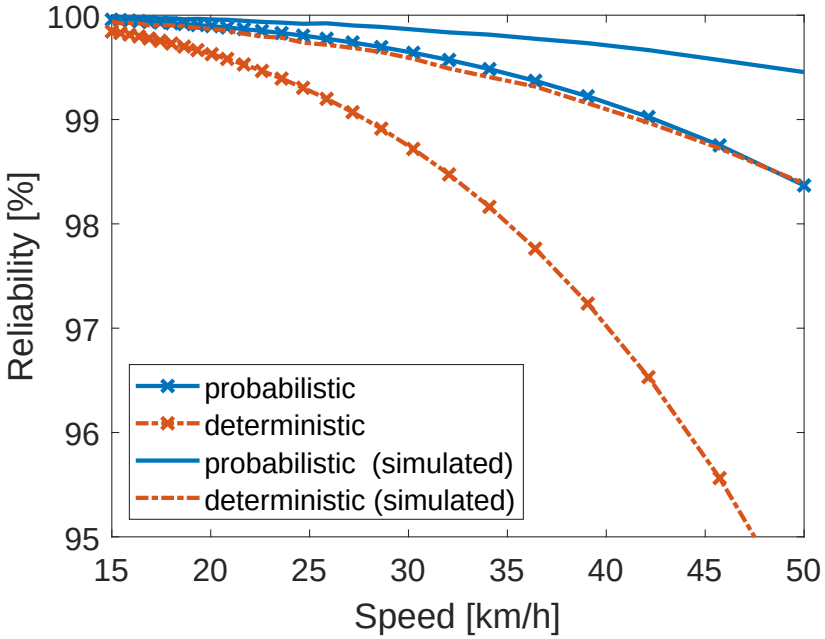


Figure 9.4.: Calculated communication reliability for varying vehicle speed.

Communication Reliability vs. Speed: Fig. 9.4 shows the calculated (as per Eq. (4.19)) and the simulated (average) transmission reliability in relation to the vehicle's cruise speed at the intersection. 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. 9.4.

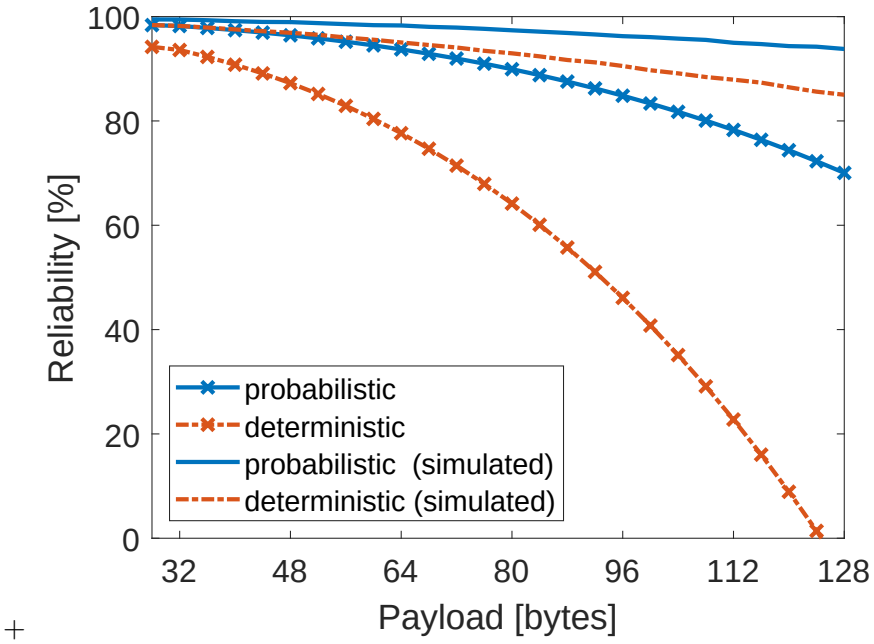


Figure 9.5.: Simulated communication reliability for varying payload length.

Communication Reliability vs. Payload: Fig. 9.5, shows 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.

Physical Resolution: Fig. 9.6 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. 9.6 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 50 km/h case, reliability is increased from 89 % to $\approx$ 99 % with the probabilistic 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 % [65].

Further, note that a lower speed improves resolution and/or reliability, since the vehicles have again more time to transmit data — see again Fig. 9.4. For example, reducing the speed from 50 km/h to 30 km/h and assuming 99 % reliability, the resolution improves from 2.2 m to 1.8 m. Further, it can be seen in Fig. 9.6 that the probabilistic technique at 50 km/h still has a slightly better (lower) resolution than the deterministic one at 30 km/h.

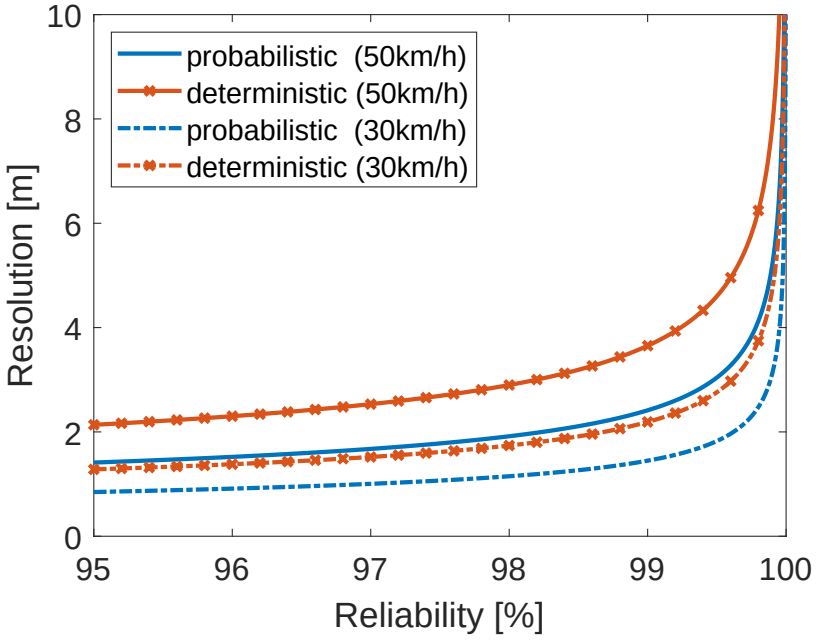


Figure 9.6.: Physical resolution for varying communication reliability.

In summary, it can be seen that probabilistic estimates for the maximum number of vehicles can greatly reduce pessimism compared to deterministic approaches. In particular, a much higher communication reliability with regards to packet loss can be achieved. 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 $c = 20$). These results indicate that it is more likely that the communication fails rather than c is exceeded in this example.

9.1.3. Fallback Mechanism — Examples

With the deterministic worst-case vehicle count being $c = 30$ and a chose probabilistic worst-case vehicle count of $c = 20$ with an exceedance probability of $\approx 4.12 \times 10^{-7}$, some specific considerations can be made by the RSU during runtime.

With the deterministic maximum vehicle count only occurring when only overly short vehicles are present at the intersection, the RSU can keep track of the length of vehicles already at the crossroad to provide a band of possible vehicle counts in the upcoming cycles. For example, already having a single overlength vehicle with $L > S$ in each lane of the intersection reduces the maximum worst-case vehicle count on that lane by 1 and the total at the intersection by 4 until these vehicles have traversed the intersection center due to the overlength penalty affecting the vehicle periods. This way, unless completely unexpected events (such as a tandem motorcycle motorcade) occur, a maximum vehicle count exceedance at the intersection will not come as a surprise to the RSU. It will rather be a gradual and transparent process, allowing the RSU to start implementing system-wide measures as discussed in Section 4.4 at vehicle counts close to the maximum, for example for $c = 16$ as the maximum number of vehicles that can join at roughly the same time is 1 per lane, so 4 in total.

9.2. Space Efficiency

This section is concerned with the space-efficient traffic protocol SV-LTR introduced in Chapter 5 and its size and area requirements compared to BRIP [11]. However, note that all traffic protocols proposed in this thesis, (SV-LTR, PB-LTR and FleXS-TP) lead to the same space requirements on the intersection. As such, the evaluation in this section applies to all these protocols.

While for BRIP, the longest possible vehicle at the intersection determines the sector size $S_{BRIP} \geq L + W$, the sector size S for SV-LTR is independent of vehicle length but rather determined by the width of the lane, i.e., by the given characteristics of the infrastructure. Therefore, this evaluation includes three different variants of SV-LTR: LTR_3 with $S = 3\ m$, LTR_5 with $S = 5\ m$ and LTR_7 with $S = 7\ m$.

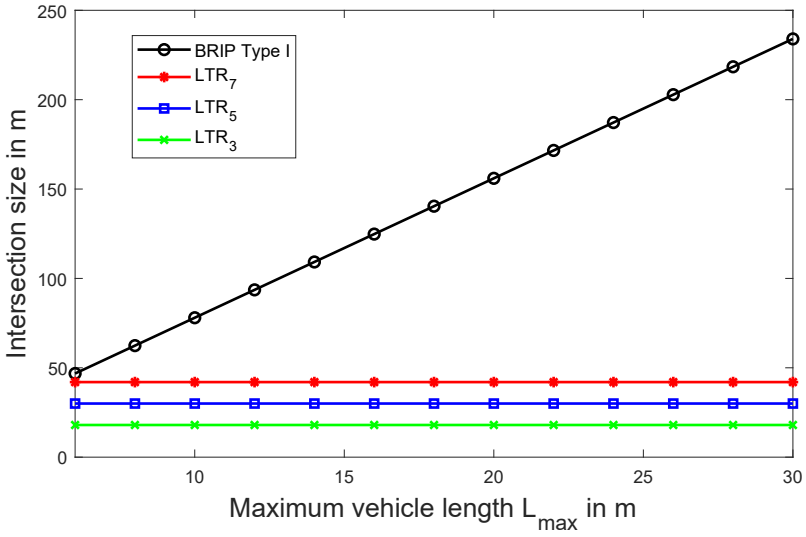


Figure 9.7.: Intersection size versus vehicle length.

Figs. 9.7 and 9.8 depict a comparison of the space and area requirements on the intersection infrastructure by BRIP and the SV-LTR-variants discussed above. As expected, BRIP's space requirements rapidly increase with the length of the longest vehicle allowed at the intersection. For example, for vehicles with $L_{max} = 15\ m$ to be allowed at the intersection, BRIP requires the intersection to provide a length of 80 m, which translates in 6400 m^2. Clearly, the longer vehicles are, the less existing intersections comply with BRIP's space requirements.

In contrast to this, SV-LTR does not have any inherent space requirements, but rather is parameterized to current lane width in an existing intersection. For example, LTR_5 requires the intersection width to be 20 m (i.e., 400 m^2) independent of vehicles' lengths — provided the vehicles can still maneuver (i.e., not get stuck when turning).

It should be noted that the values in Figs. 9.7 and 9.8 are representing BRIP Type I as the default layout from Fig. 3.1 is a two-lane intersection. To extend this to a three-lane layout such as BRIP Type III, the additional lane increases the size and area requirements from Eqs. (5.3) and (5.4), with $Size = 4S \rightarrow 6S$ and $Area = 16S^2 \rightarrow 36S^2$.

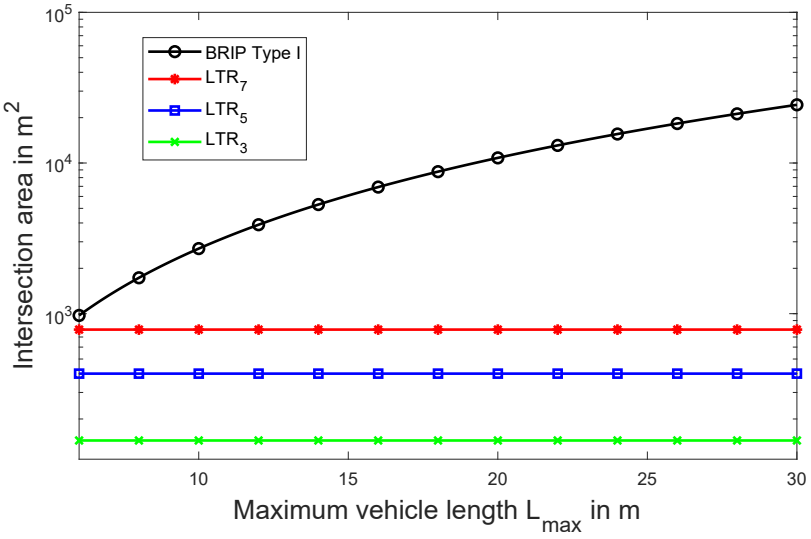


Figure 9.8.: Intersection area (logarithmic) versus vehicle length.

Clearly, uncoupling vehicle length and sector size greatly reduces the size and area requirements for the intersection, which in turn increases the range of intersections a traffic protocol can potentially be applied to.

9.3. Throughput Part I: LTR vs. BRIP

This section evaluates the different versions of LTR with regards to throughput. Specifically, it contains the following individual comparisons:

- BRIP Types I and III [11] vs. SV-LTR [51] from Section 5.2
- BRIP Types I and III [11] vs. PB-LTR [52] from Section 6.1

9.3.1. SV-LTR vs. BRIP

As discussed in Chapter 2, while BRIP has been extended into CSIP [6] and DSIP [7], the additional constraints these protocols introduce reduce their throughput when compared to BRIP. On the other hand, they retain the same space requirements as BRIP. In this section, for ease of exposition, all comparisons and evaluation are only made towards BRIP based on best-case and worst-case performance, as all comparisons are also valid for CSIP and DSIP.

As previously discussed in Section 3.1.1, one of the core assumptions of BRIP is that the sector length has to be at least $L+W$, where L and W are the length and width. To this end, the width of a vehicle has been set to be 30 % of its length in this analysis, but no more than 2.55 m (i.e., all vehicles longer than $L = 8.5$ m are considered to be as wide as $W = 2.55$ m), which is the maximum authorized width for vehicles in the European Union [22].

Furthermore, BRIP synchronizes vehicles in all direction with the same constant speed. For settings with both vehicles driving through and vehicles turning right/left, this requires the chosen speed to be low enough to allow for safety and comfort, even for the vehicles driving through. This affects all BRIP types except for Type I.

While BRIP was conceived for same-size vehicles only, the possibility considering overlength vehicles was contemplated by changing arrival patterns and allowing for more empty arrival slots [11]. However, this implies nontrivial modifications, certainly diminishing throughput, and remains unsolved.

To perform a comparison, in this section, *well-behaved* traffic is assumed, i.e., uninterrupted vehicle flow following the patterns by the corresponding protocol. Further, vehicles are assumed to have a constant *baseline speed* of $V_{LO} = V_{BRIP} = 30\ km/h$. This way, throughput can be expressed concretely with vehicles per minute (instead of vehicles per cycle). This also leads to $V_{HI} = 1.5V_{LO} = 45\ km/h$ for vehicles driving through under SV-LTR. Note that while higher baseline speeds may yield greater throughput, the following comparison and conclusions hold regardless of the specific speed chosen.

Concerning right turns: BRIP Types II, III and IV have a dedicated right-turn lane. All three types allow for one vehicle per BRIP cycle and direction to turn right. However, the remaining lanes in these three cases are completely independent of right turns, as depicted in Fig. 9.9 for the case of Type III, where left turns and drive-through maneuvers are confined to a a two-lane space as marked by the red square, with right turns being inconsequential for the aforementioned maneuvers. As a result, right turns are analyzed separate from the other the lanes. First, a detailed comparison of SV-LTR with BRIP Type I and Type III without right turns is performed. Then, the effect of a dedicated right-turn lane on the different protocols is analyzed separately. Note that Type III is a generalization of Type II and Type IV. So Type II and IV are not explicitly included in this comparison.

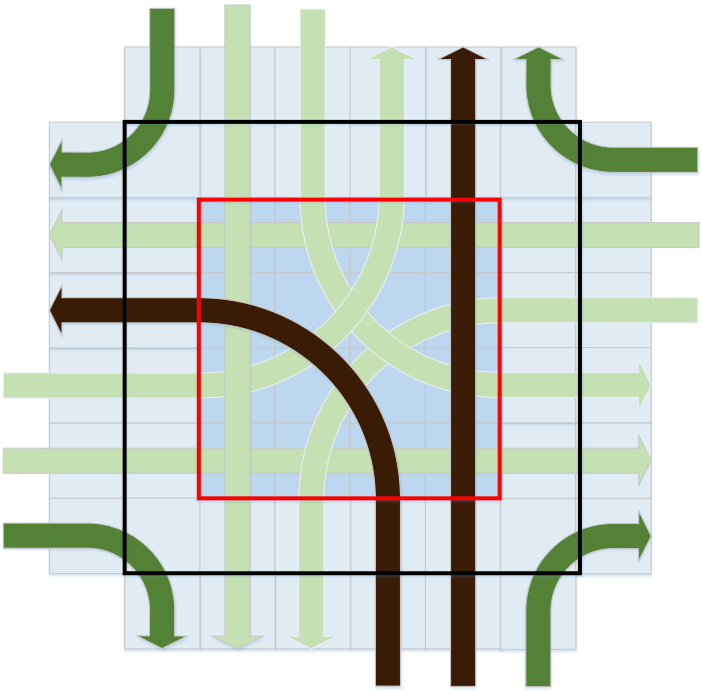


Figure 9.9.: Layout for BRIP Type III.

BRIP Type I: Note that, since BRIP Type I only consists of vehicles driving through, both BRIP and the two LTR-versions utilize the higher speed $V_{BRIP} = V_{HI} = 1.5V_{LO}$ for the entire maneuver. Fig. 9.10 shows the comparison of throughput versus maximum vehicle length for BRIP Type I against SV-LTR with $V_{Hi} = 45\ km/h$ and the variants LTR_3, LTR_5 and LTR_7 for sector sizes 3 m, 5 m and 7 m respectively. For settings consisting of only short vehicles, BRIP Type I substantially outperforms SV-LTR in all its variants in terms of throughput. With longer vehicles being allowed at the intersection, this disparity grows smaller until SV-LTR starts outperforming BRIP.

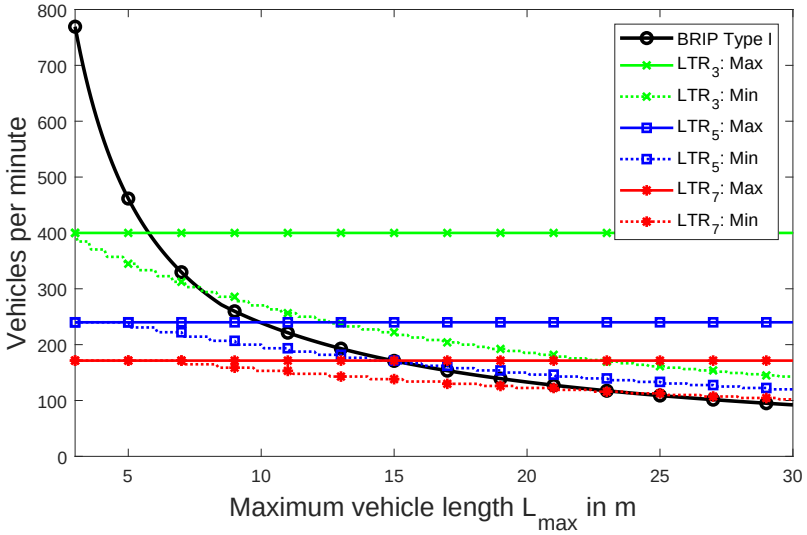


Figure 9.10.: BRIP Type I vs. SV-LTR — Throughput.

For a given maximum vehicle length L_{max}, all LTR-variants lead to a maximum and a minimum throughput, depending on the frequency with which these vehicles actually appear at the intersection. If these long vehicles are possible, but never cross the intersection, the best/maximum possible throughput (i.e., the best case) for SV-LTR is achieved. The worst case takes place if only these long vehicles cross the intersection, resulting in the worst/minimum possible throughput (i.e., the worst case) for SV-LTR. In all figures, these are marked with „$LTR_S : Max$“ and „$LTR_S : Min$“ respectively.

On average, the expected throughput is between these maximum and minimum curves of the corresponding SV-LTR variant. For example, LTR_5 provides a throughput between 170 and 240 vehicles per minute for vehicles as long as $L_{max} = 15\ m$. BRIP Type I allows for around 167 vehicles per minute in this case (independent of how often long vehicles cross). That is, when considering vehicles as long as $L_{max} = 15\ m$, LTR_5 roughly breaks even with BRIP in the worst case, but it allows for up to 40 % more throughput in the best case. In particular, LTR_5 starts outperforming BRIP Type I for $L_{max}^{BC} > 7.8\ m$ in the best case and $L_{max}^{WC} > 14.5\ m$ in the worst case. From now on, these bounds by be denoted by $L_{max} > [L_{max}^{BC}, L_{max}^{WC}]$, i.e., $L_{max} > [9.95\ m, 14.95\ m]$ for the current setting. These values are visualized at the end of this section.

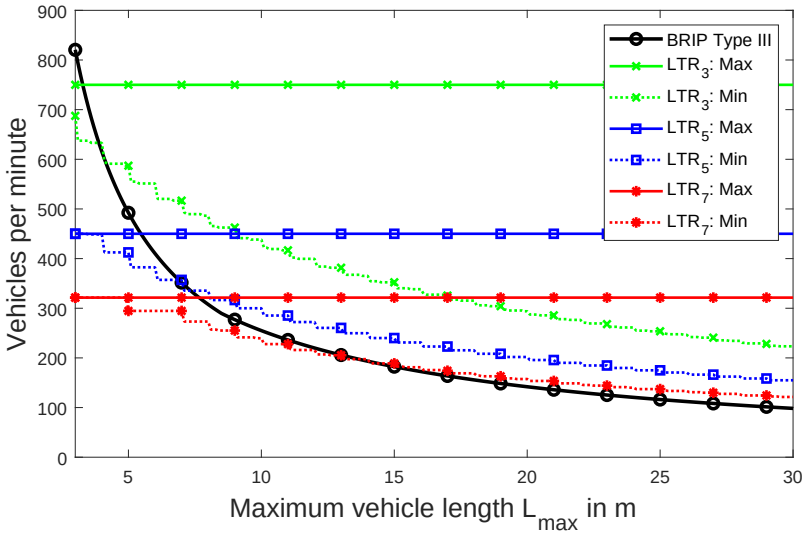


Figure 9.11.: BRIP Type III vs. SV-LTR — Throughput (with right turns).

BRIP Type III: Since Type III combines turn and drive-through maneuvers, BRIP is bounded by the turn speed V_{LO}, since, in contrast to the approaches proposed in this thesis, it does not allow for a two-speed regime.

Independent of whether right turns are considered or not (in Fig. 9.11 and Fig. 9.12 respectively), a similar behavior as in the previous comparison against BRIP Type I can be observed. That is, for settings with only short vehicles, BRIP Type III outperforms LTR in all its variants. However, SV-LTR starts outperforming BRIP Type III as longer vehicles start being possible/allowed at the intersection.

For instance, when disregarding right turns in Fig. 9.11, LTR_5 leads to a throughput between roughly 90 and 150 vehicles per minute when considering that vehicles of up to $L_{max} = 15\ m$ are allowed at the intersection. Similar to before, the minimum throughput occurs when only

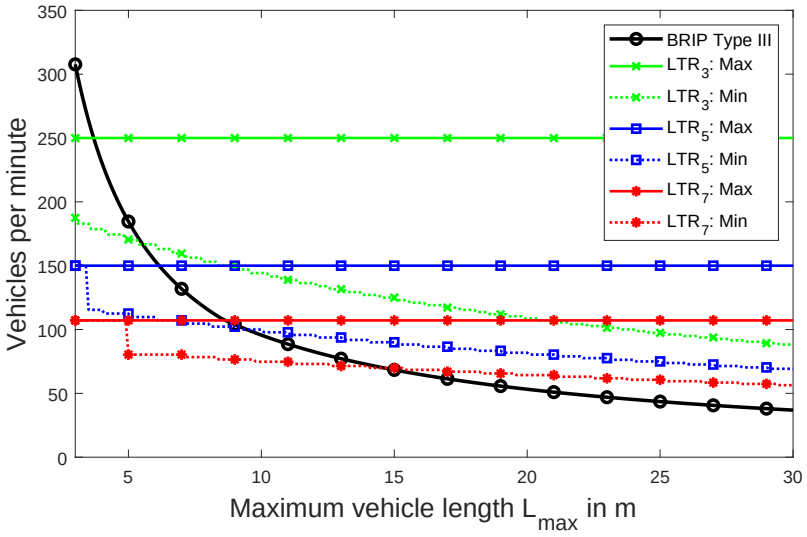


Figure 9.12.: BRIP Type III vs. SV-LTR — Throughput (no right turns).

15 m vehicles cross the intersection, whereas the maximum throughput results when 15 m vehicles are possible, but seldom present at the intersection. In contrast to this, BRIP Type III reaches a performance of around 67 vehicles per minute if vehicles as long as 15 m can cross, independent of whether these vehicles are present or not at the intersection. Specifically, LTR_5 starts outperforming BRIP Type III in this case for $L_{max} > [6.2\ m, 9.5\ m]$, i.e., for maximum vehicle lengths greater than 6.2 m in the best case and for vehicle lengths greater than 9.5 m in the worst case.

Taking right turns into consideration substantially increases throughput numbers due to the constant uninterrupted flow of right turns on a dedicated lane. Here, as shown in Fig. 9.12, LTR_5 leads to a throughput between 240 and 450 vehicles per minute for the same maximum vehicle length $L_{max} = 15\ m$, whereas BRIP reaches a throughput of roughly 178. When considering right turns, LTR_5 starts outperforming BRIP Type III for $L_{max} > [5.48\ m, 6.9\ m]$.

Note that for $L_{max} = 5\ m$, BRIP outperforms SV-LTR, since BRIP always allows for a four-way synchronization (i.e., vehicles crossing in all four directions at the same time) for left turns. More specifically, fixing the vehicle length to 5 m changes BRIP's sector size to something more than 5 m, i.e., length plus width of the vehicle, which increases the critical margin for left turns discussed in Section 3.1.4. BRIP simply assumes that the intersections are always able to accommodate its necessary sector size.

In contrast to this, SV-LTR's sectors do not depend on vehicle length and are rather determined by the (existing) lane width. As previously mentioned, the considered lane widths are 3 m, 5 m and 7 m, which results in the SV-LTR variants LTR_3, LTR_5 and LTR_7.

For smaller sector sizes, SV-LTR does not always allow for a four-way synchronization, due to the lack of space in smaller intersections (with narrower lanes). For example, SV-LTR does not allow for a four-way synchronization for left turns with vehicles longer than $\approx 0.7S$ (as described in Section 3.1.4) and, hence, has a worse performance than BRIP.

9.3.2. PB-LTR vs. BRIP

This section considers the platoon crossing variant PB-LTR proposed in Section 6.1. For ease of exposition, in the following analysis, PB-LTRs behavior for $S = 5\ m$ for different maximum blocking thresholds B_{max} (i.e., the maximum blocking time in cycles C before a switch must be enacted as described in Section 6.1.4) is shown.

BRIP Type I: A comparison of BRIP Type I with the $PB - LTR_5$ can be seen in Fig. 9.13. Here, $S = 5\ m$ and solid lines represent the best-case throughput, while dotted lines represent the worst-case throughput.

Just as before, both the best case (i.e., the longest possible vehicles are rarely present) and the worst case (all vehicles are as long as the longest possible vehicle) are considered. As noted previously, SV-LTR outperforms BRIP in terms of throughput under single-vehicle crossing for $L_{max} > [9.95\ m, 14.95\ m]$. Under platoon crossing with $B_{max} = 24$, these values improve to $L_{max} > [6.16\ m, 6.67\ m]$. Similarly, PB-LTR outperforms BRIP under platoon crossing for $B_{max} = 48$ and $L_{max} > [5.59\ m, 5.85\ m]$, whereas $L_{max} > [5.35\ m, 5.47\ m]$ results for $B_{max} = 96$. That is, the greater B_{max}, the smaller the gap between best and worst case.

Now, for $L_{max} = 15\ m$, BRIP allows for 167 vehicles per minute in a Type I setting. Here, PB-LTR with $B_{max} = 48$ can reach a throughput of 282 to 412 vehicles per minute, i.e., 65 % to 145 % higher throughput than BRIP. Since B_{max} is the maximum number of cycles vehicles on conflicting lanes have to wait to cross, $B_{max} = 48$ results in a waiting time $t_{wait} = 48\frac{S}{V_{LO}} = 28.8\ s$ for $S = 5\ m$ and $V_{LO} = 30\ km/h$.

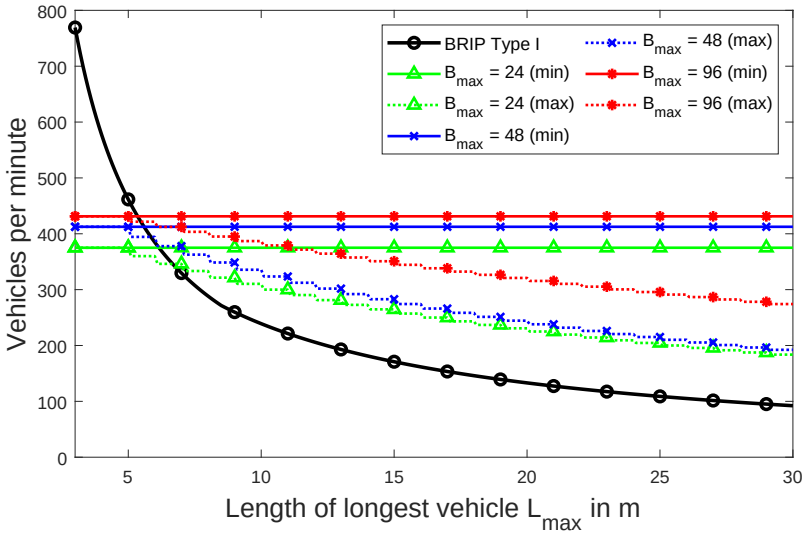


Figure 9.13.: BRIP Type I vs. PB-LTR — Throughput.

BRIP Type III: In Fig. 9.14, disregarding right turns, the bound at which PB-LTR outperforms BRIP are $L_{max} > [7.4\ m, 15.0\ m]$ for $B_{max} = 24$, $L_{max} > [5.9\ m, 7.75\ m]$ for $B_{max} = 48$ and $L_{max} > [5.37\ m, 5.85\ m]$ for $B_{max} = 96$. When comparing to Fig. 9.11, PB-LTR can be seen to outperform SV-LTR for a sufficiently high B_{max}. When considering vehicles that are as long as 15 m, PB-LTR allows for between 76 to 156 vehicles per minute for $B_{max} = 48$ (which corresponds to a waiting time of 28.8 s on conflicting lanes). For the same setting, BRIP allows only 67 vehicles per minute.

The same holds true when considering right turns, see Fig. 9.15. In particular, the bounds of outperforming BRIP are now at $L_{max} > [5.8\ m, 6.9\ m]$ for $B_{max} = 24$, $L_{max} > [5.4\ m, 5.94\ m]$ for $B_{max} = 48$, and $L_{max} > [5.22\ m, 5.72\ m]$ for $B_{max} = 96$. For vehicles of at most 15 m,

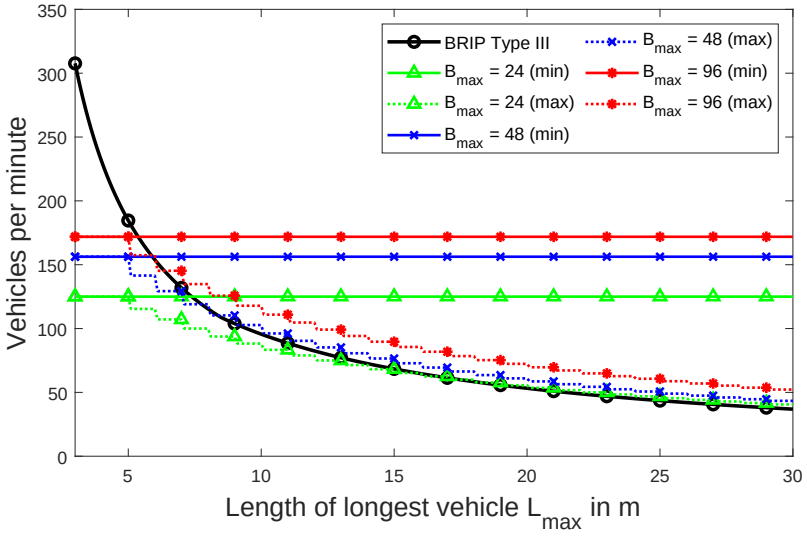


Figure 9.14.: BRIP Type III vs. PB-LTR – Throughput (no right turns).

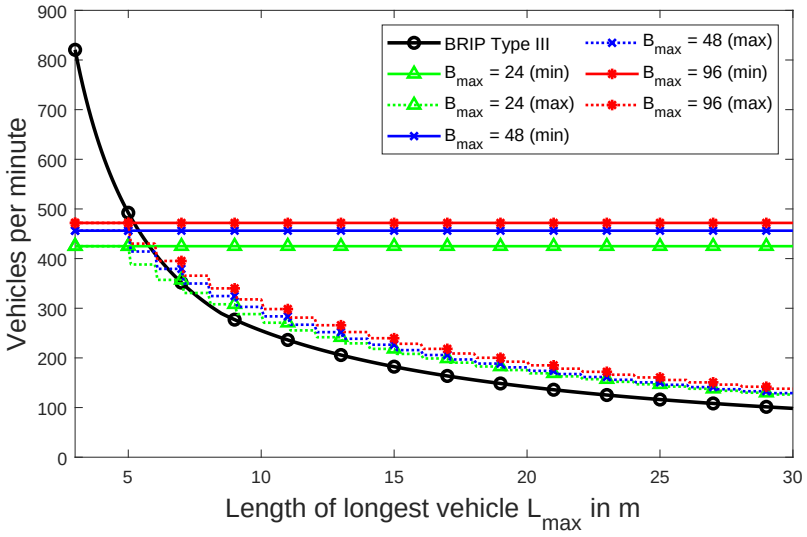


Figure 9.15.: BRIP Type III vs. PB-LTR — Throughput (with right turns).

PB-LTR can achieve between 226 and 456 vehicles per minute for $B_{max} = 48$ (i.e., a waiting time of no more than 28.8 s on conflicting lanes), as compared to 178 vehicles per minute by BRIP, i.e., allowing for an increase in throughput of roughly 25 % to 150 %.

As mentioned before, for $L_{max} = 5\ m$, BRIP outperforms PB-LTR for most cases due to allowing for a four-way synchronization at all times.

While vehicle periods (i.e., bumper-to-bumper distances) between the two approaches are similar (e.g., $2S_{LTR}$ and $2S_{BRIP}$ for driving through on one lane), this consistent four-way synchronization gives BRIP an initial advantage in terms of throughput. However, with increasing vehicle lengths, S_{BRIP} also increases while S_{LTR} remains constant, which shifts the comparison in favor of PB-LTR.

9.3.3. Summary: LTR-versions vs. BRIP

The previously mentioned bounds of maximum vehicle length at which the different LTR-versions start outperforming BRIP denoted $L_{max} > [L^{BC}_{max}, L^{WC}_{max}]$ are subsequently displayed in Tables 9.4 and 9.5 and visualized for $L \in [0\ m, 30\ m]$ in Fig. 9.16 for BRIP Type I and Fig. 9.17 for BRIP Type III with and without right turns. In these figures, depending on the maximum vehicle length allowed at the intersection, the protocol achieving the highest throughput is displayed, with BRIP, marked with a gray bar, being the clear winner for only short vehicles due to the tightly interwoven nature of the protocol. The yellow bar between the LTR-versions and BRIP denotes that this vehicle length is between L^{BC}_{max} and L^{WC}_{max}, i.e., depending on how often vehicles of this length actually occur, either protocol might perform better. The green bar denotes vehicles lengths at which the corresponding LTR-version clearly outperforms BRIP, even in the worst case. Further, the different strategies are denoted as follows: SV-LTR is denoted with the superscript „SV“ and includes its sector size in the subscript, i.e., LTR^{SV}_5 describes single vehicle LTR with $S = 5\ m$. Similarly, PB-LTR includes the value for the maximum blocking B_{max} in the subscript, together with the identifier „PB“ in the superscript, i.e., PB-LTR with $B_{max} = 48$ is described via LTR^{PB}_{48}.

BRIP Type I		
Approach	L_{max}^{BC}	L_{max}^{WC}
LTR_3^{SV}	5.80 m	7.80 m
LTR_5^{SV}	9.95 m	14.95 m
LTR_7^{SV}	14.95 m	23.30 m
LTR_{24}^{PB}	6.16 m	6.67 m
LTR_{48}^{PB}	5.59 m	5.85 m
LTR_{96}^{PB}	5.35 m	5.47 m

Table 9.4.: Bounds of outperforming BRIP Type I $L_{max} > [L_{max}^{BC}, L_{max}^{WC}]$ for the different proposed approaches.

BRIP Type III — with right turns		
Approach	L_{max}^{BC}	L_{max}^{WC}
LTR_3^{SV}	3.30 m	3.90 m
LTR_5^{SV}	5.48 m	6.90 m
LTR_7^{SV}	7.66 m	13.65 m
LTR_{24}^{PB}	5.80 m	6.90 m
LTR_{48}^{PB}	5.40 m	5.94 m
LTR_{96}^{PB}	5.22 m	5.72 m

BRIP Type III — no right turns		
Approach	L_{max}^{BC}	L_{max}^{WC}
LTR_3^{SV}	3.70 m	5.50 m
LTR_5^{SV}	6.20 m	9.50 m
LTR_7^{SV}	8.70 m	14.60 m
LTR_{24}^{PB}	7.40 m	15.00 m
LTR_{48}^{PB}	5.90 m	7.75 m
LTR_{96}^{PB}	5.37 m	5.85 m

Table 9.5.: Bounds of outperforming BRIP Type III $L_{max} > [L_{max}^{BC}, L_{max}^{WC}]$ for the different proposed approaches.

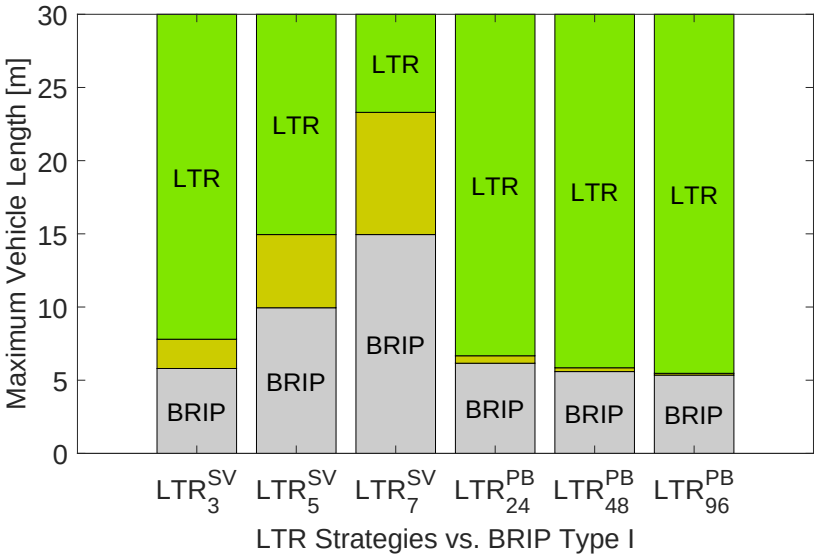
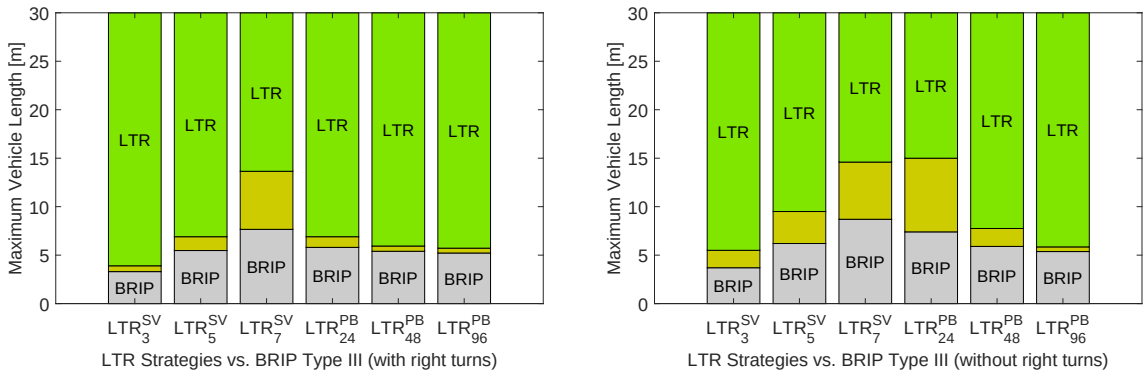


Figure 9.16.: Performance bounds for LTR-versions vs. BRIP Type I.



(a) Including right turns.

(b) Without right turns.

Figure 9.17.: Performance bounds for LTR vs. BRIP Type III.

To summarize, BRIP provides very high throughput numbers and reasonable space requirements on the intersection for short (e.g., less than 5 m) and homogeneous vehicles. However, with increasing vehicle lengths, BRIP's space requirements increase drastically, which makes it rather unsuitable for most existing intersection.

In contrast to this, the proposed approaches SV-LTR and PB-LTR require substantially less space both for single-vehicle and platoon crossing alike. Moreover, when heterogeneous vehicles are allowed at the intersection, SV-LTR's throughput starts outperforming BRIP. The greater the difference in length between the longest and the standard vehicle at the intersection, the higher SV-LTR's throughput will be with respect to BRIP.

Further, for homogeneous traffic, the platooning-based version PB-LTR also outperforms BRIP when larger vehicles are present at the intersection. Note however that the PB-LTR heavily relies on traffic arriving exactly as required to fill the platoons. Having randomized traffic will lead to significantly more frequent transitions between individual platoons which in turn will greatly reduce throughput. Still, the assumption of well-behaved traffic — and its associated drawback — is shared by both PB-LTR and BRIP. The next section evaluates FleXS-TP, which aims to mitigate this limitation while preserving the benefits under well-behaved conditions.

9.4. Throughput Part II: SV-LTR vs. FleXS-TP

This section, the proposed flexible synchronous traffic protocol named FleXS-TP introduced in Chapter 7 is evaluated regarding throughout via simulation. Specifically, results for both a well-behaved system, where vehicles arrive as anticipated by the protocol, as well as for a system with randomly generated vehicles are presented. To this end, FleXS-TP is compared to SV-LTR as well as a conventional traffic light system. Since SV-LTR has been shown to be more suitable under realistic conditions (i.e., in existing infrastructure and with overlength vehicles that arrive from time to time) in the previous Section, the following analysis is restricted to this protocol alone and no explicit comparison to PB-LTR is made.

9.4.1. Well-Behaved Traffic

First, a well-behaved system is considered, where vehicles arrive exactly in the order the corresponding protocol expects them to. More specifically, two well-behaved systems are considered: one exclusively with vehicles turning left and one exclusively with vehicles driving through. For these settings, the following approaches are compared: i) As before, vehicles driving through without contention (i.e., without stopping and disregarding safety entirely) as an upper bound for performance ii) a regular traffic light as described in Section 8.1.3, iii) SV-LTR, iv) FleXS-TP for both variable and square sectors.

Since a well-behaved system is entirely deterministic, a parameter of occurrence rate for overlength vehicles has been added, i.e., each vehicle in the system has a chance of being generated as an overlength vehicle with $2S > L_i > S$. The results for a left turn only system can be seen in Fig. 9.18 when four-way synchronization is not possible (i.e., $L_{nonOL} > 0.7S$) and in Fig. 9.19 when four-way synchronization is possible for non-overlength vehicles (i.e., $L_{nonOL} \leq 0.7S$). The results for the well-behaved system of vehicles driving through can be seen in Fig. 9.20.

Left turns only: To harness the performance of SV-LTR under fully synchronous settings, FleXS-TP is designed to correspond exactly to that behavior in the case of a well-behaved system with homogeneous traffic. This can also be seen here where both approaches take the same amount of time when no overlength vehicles are present. Further, as the key mechanic enabling throughput for the proposed protocol is four-way synchronization, when all vehicles exceed the condition for four-way synchronization and only two-way synchronization is possible, throughput suffers greatly overall, as can be seen in Fig. 9.18.

When four-way synchronization is possible, as presented in Fig. 9.19, the proposed protocols again share the same performance for a setting without overlength vehicles. With increasing numbers of overlength vehicles however, the throughput is reduced again as four-way synchronization becomes increasingly less likely, with FleXS-TP remaining close to the simple traffic light since it effectively enforces a form of two-way synchronization. However, as the traffic

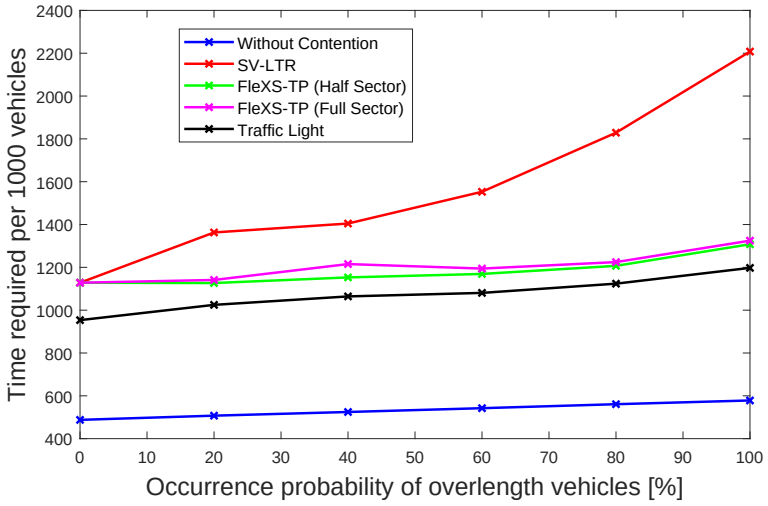


Figure 9.18.: Time required to have 1000 vehicles traverse for well-behaved traffic turning left **without** four-way synchronization.

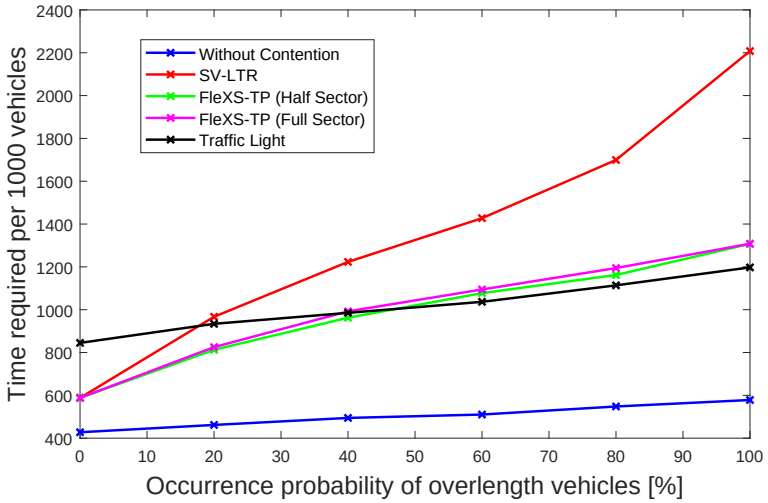


Figure 9.19.: Time required to have 1000 vehicles traverse for well-behaved traffic turning left **with** four-way synchronization

light simulation is fed the traffic set without contention as described in Section 8.1.2, it utilizes a minGap of 1, allowing vehicles to queue very closely at traffic lights, making it very effective when two-way synchronization is the only possible strategy.

Driving through only: As before, since FleXS-TP exactly represents SV-LTR in the case of a well-behaved system, both approaches are identical in the case of no overlength vehicles, see Fig. 9.20. Further, as both protocols enforce synchronicity for sets of four vehicles driving through, their overall performance is almost identical. As before, the traffic light enforces a complete two-way synchronization which leads to a substantial decrease in throughput compared to both proposed protocols.

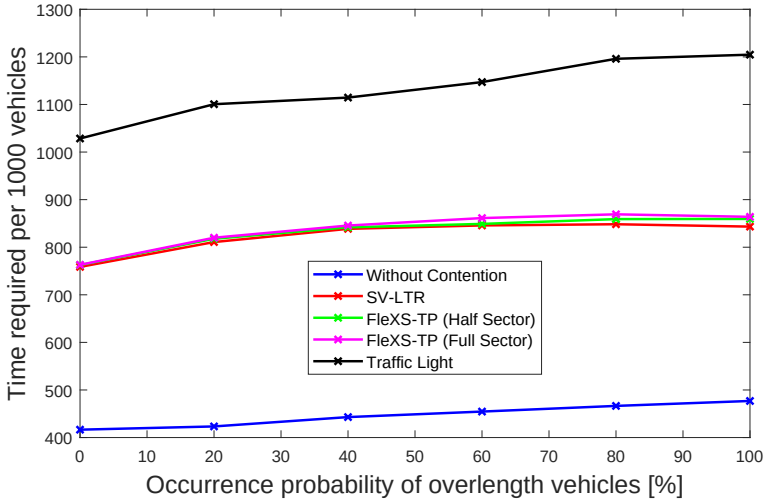


Figure 9.20.: Time required to have 1000 vehicles traverse for well-behaved traffic driving through.

9.4.2. Randomly Generated Traffic

For a more realistic system with randomly generated traffic, as described in Section 8, the time required for 1000 vehicles to traverse in relation to the sector size can be seen in Fig. 9.21.

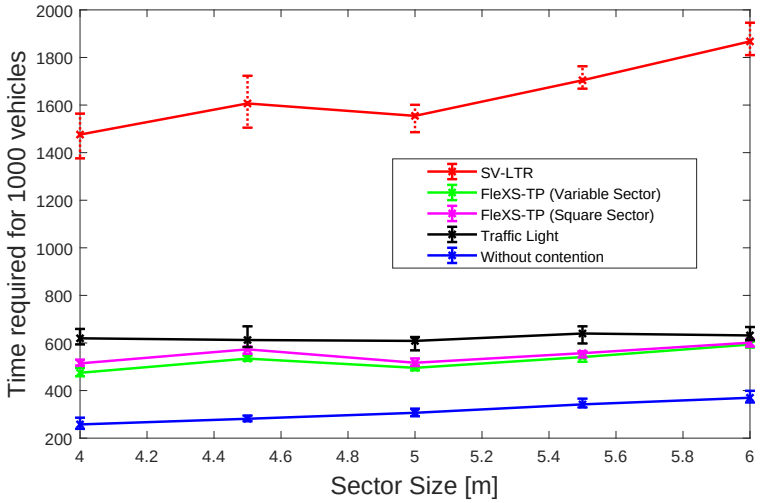
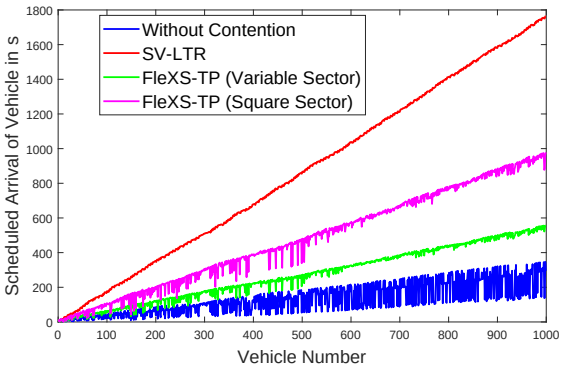
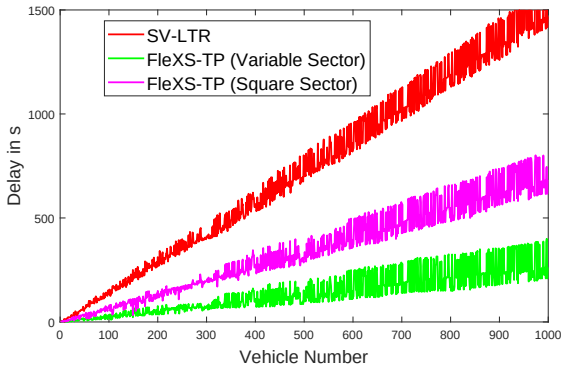


Figure 9.21.: Time required to have 1000 vehicles traverse for randomly generated traffic.

Here, substantially less deviation between generated traffic sets can be observed for FleXS-TP, whereas both SV-LTR and the traffic-light based approach differ by up to ≈ 10 %. Additionally, the rigid nature of the fully synchronous SV-LTR proves to be detrimental for unpredictable traffic due to a large number of empty set that this leads to. In contrast, the traffic-light based approach performs quite well for more realistic traffic. Specifically, for a sector size of $S = 4.5\ m$, its performance rivals that of FleXS-TP. This is due to sector size directly influencing cycles and as such, inter-vehicle distances. In addition, for this sector size, 80 % of traffic is considered overlength traffic. Once the sector size becomes greater than $S = 5\ m$, the overwhelming majority of vehicles fit into a sector and the rate of overlength vehicles drops to 20 %.



(a) Intended arrival time at the intersection.



(b) Delay incurred by the over the idealized case of no contention.

Figure 9.22.: Intended arrival and delay incurred by the different protocols.

Since vehicles in the randomly generated set are spaced by a short, also randomly distributed distance to the previous vehicle on their lane, individual vehicle delays were considered in Figs. 9.22a and 9.22b. Fig. 9.22a compares vehicle arrival times at the time of creation and after scheduling, as determined by the two approaches. Here, SV-LTR has effectively no variance due to the synchronized nature of the protocol. In contrast, FleXS-TP is more flexible to react to different generated sets which leads to slightly different delays between the vehicles. Fig. 9.22b shows the difference between contention-free crossing times and those under each respective approach, i.e., the effective delay caused by the scheduling of the approaches, which validates these observations. Note that any delay also inadvertently delays subsequent vehicles on the same lane, which increases total delay over time.

To summarize, while FleXS-TP matches the performance of the fully synchronous SV-LTR under well-behaved traffic, it significantly improves throughput and reduces delays under more realistic, randomized conditions. These results confirm that FleXS-TP effectively achieves its goal of broadening the protocol's applicability to a wider range of traffic patterns while also easing integration with existing infrastructure.

Chapter 10.

Discussion and Conclusion

This final chapter presents a summary of all findings and further includes a critical reflection on the methods of this thesis. Finally, an outlook towards further works in this field is given.

10.1. Summary of Findings

A key element of the motivation for this work is to maximize throughput in all types of traffic, both well-behaved and random, by harnessing the benefits of both synchronous and asynchronous approaches. With the ever-growing number of autonomous participants in day-to-day traffic, confining strategies for intelligent intersections to a specific type of traffic greatly restricts their applicability in the real world. This applies to both existing traffic and infrastructure alike.

To alleviate this situation, this thesis proposed several techniques with the goal of bridging the gaps between synchronous and asynchronous traffic protocols and making them applicable to a wider range of settings. Specifically, this thesis proposed a probabilistic modeling of communication reliability to reduce deterministic pessimism, as well as three traffic protocols: i) SV-LTR to increase space efficiency of traffic protocols, ii) PB-LTR to apply the concept of platooning to intelligent intersections and iii) FleXS-TP to increase flexibility towards traffic composition.

The proposed techniques were evaluated and validated by extensive simulations using OMNeT++ [82] for communication reliability and SUMo [46] for traffic protocols. The findings/-contributions by this thesis are stated next in more detail.

10.1.1. Chapter 4 — Reducing Deterministic Pessimism

Due to the safety-critical nature of intelligent intersections, 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.

Instead of constraining the entire traffic protocol by the longest possible vehicle at the intersection regardless of its actual occurrence rate, using a realistic, probabilistic vehicle length distribution allows for a reduction in deterministic pessimism and the following overdesign by using stepwise, probabilistic worst cases.

10.1.2. Chapter 5 — Infrastructure-Agnostic Design

Treating extraordinarily large vehicles as exception instead of the norm and including the overlength condition as a measure to handle vehicles exceeding a singular sector leads to an uncoupling of the traffic protocols sector size S from the actual traffic, as introduced in the space-efficient traffic protocol SV-LTR and also used in PB-LTR and FleXS-TP. Instead, the existing infrastructure now provides the sector size S, as it is inadvertently tied to the lane width. This allows for infrastructure-agnostic design, where the traffic protocol can be applied to any given infrastructure with the existing geometry parameterizing the protocol, alleviating the need for great modifications of existing intersections, while providing competitive or even greater throughput.

10.1.3. Chapter 6 — Platooning for Well-Behaved Traffic

The traffic protocol PB-LTR extends SV-LTR's single-vehicle maneuvers to platoons consecutively performing the same maneuver, enabling shorter inter-vehicle distances and reducing downtime from maneuver transitions. To ensure all vehicles eventually cross the intersection, a maximum blocking time regularly enforces maneuver changes, guaranteeing fairness across all directions. Under well-behaved traffic, PB-LTR can further increase throughput.

10.1.4. Chapter 7 — Increasing Flexibility towards Traffic Composition

By maintaining a synchronous strategy at heart, but scheduling vehicles on a vehicle-by-vehicle basis while still allowing for situational synchronicity, FleXS-TP can harness both the benefits of synchronous traffic protocols for well-behaved traffic, as well as the flexibility of asynchronous traffic protocols under randomized traffic, achieving substantial throughput in both settings.

10.2. Reflecting Design Choices/Limitations

This section briefly reflects on some choices made throughout the thesis to address potential drawbacks and/or to provide justifications.

10.2.1. Vehicle Reordering

None of the protocols discussed in this thesis (BRIP, SV-LTR, PB-LTR, FleXS-TP) utilize active vehicle reordering, but instead handle vehicles in a First Come, First Served (FCFS) fashion. That is due to several reasons: i) vehicle reordering demands considerable computational and operational cost, ii) it introduces additional collision risks beyond the intersection center, and iii) reordering inherently introduces persistent complexity, as conveying and following detailed instructions remains challenging, especially if human drivers are to be included eventually (as discussed in Section 10.3).

Recent publications model vehicle reordering as a computationally expensive multiple-integer quadratic programming formulation, which is unsuitable for real-time applications [32].

Further, to allow for large-scale vehicle reordering, the intersections communication range must be increased substantially, which exacerbates the issue of communication overhead.

Due to the required lane changes, vehicle reordering introduces potential collision points outside the intersection, necessitating extra safeguards and fallback mechanisms, which further complicate implementation.

As such, to be well-equipped for any type of realistic, randomized and unpredictable traffic without having to rely on vehicle reordering, none of the proposed approaches utilize active vehicle reordering.

10.2.2. Coping with System Accidents

A key observation made by Perrow [67] [69] when discussing system accidents (as introduced in Chapter 4) was the role additional safety layers played in their course of events. As these type of incidents usually contain multiple unforeseen and effectively unpredictable failures, adding safety layers can have equally unforeseen, detrimental interactions with other safety mechanisms due to the rising complexity — paradoxically reducing overall system safety under the specific circumstances leading to the accident.

While, system accidents are eventually unavoidable, taking potential interactions between safety layers into account can still be a way to reduce their overall risk of occurrence, e.g., by performing *fault tree analysis* [76] for additional safety layers.

One example of such a potentially unwanted interaction is the consideration of fallback mechanisms in the case of a gradual exceedance of the probabilistic worst-case vehicle count — bluntly switching to a conventional traffic light during active operations brings its own risks that might even exceed the risk incurred by the system overshooting its maximum vehicle count, as this reflects a pessimistic analysis where packet loss from interference might occur, though it is not guaranteed. Here, it is necessary to consciously weigh the implications of each individual safety measure on other parts of the system.

10.2.3. System-Level Safety

This section briefly discusses potential measures to provide system-level safeguards to mitigate potential failures.

Redundancy

As with all centralized traffic protocols such as AIM [26], SV-LTR, PB-LTR and FleXS-TP, the RSU being the central arbiter of the traffic protocol establishes it as a singular point of failure. Here, again taking inspiration from other safety-critical systems in aviation such as the air data inertial reference unit (ADIRU) [79], a form of Triple Modular Redundancy (TMR) [48] can be utilized, where several RSUs can work in parallel following a majority ruling. While this is not infallible in itself, it can reduce the impact of a singular RSU failure and reduce the overall chance of a full system failure.

Fallback Mechanisms

Further, as discussed in Section 4.4, having vehicle-side fallback strategies such as adaptive cruise control or a form of automated collision resolution process can reduce the risk incurred by a RSU-side outage. As an example following Section 4.4, it is possible to utilize the concept of the aviation-based Traffic Collision Avoidance System (TCAS) [35] [83] to introduce a potential safety layer by including a permanent fallback watcher to the system that continuously observes the RSU and issues a system-wide, highest-priority warning state, such as a reduction in speed, should the RSU display faulty behavior (such as missed communication cycles).

These systems can even be extended by a collision prediction mechanism using machine learning [45] or game theory [47], as briefly discussed in Sections 2.3.2 and 2.3.3, potentially adding extra safety layers by using these promising technologies as supporting systems. However, as discussed in Section 10.2.2, the risk of such extra safety layers exacerbating the occurrence of system accidents is to be considered with utmost care.

Still, a key advantage of the presented centralized approaches is that vehicles receive their trajectory immediately upon entering the intersection range and generally do not receive deviating information unless a vehicle diverges from or disregards these instructions. As such, already scheduled vehicles can cross as commanded even in case of a system outage. Here, only newly incoming vehicles will not receive trajectory information at all and instead have to react by utilizing conventional methods such as traffic lights or come to a stop until the system is restored. This is a key advantage of using centralized systems, as detailed next.

10.2.4. Centralized vs. Distributed Protocols

All protocols proposed in this thesis are centralized synchronous traffic protocols, using a centralized RSU. While this makes the RSU a central point of failure, as discussed previously in Section 10.2.3, it allows for maximum predictability of the system, as all participants and their position, speed and intended trajectory are known to this centralized unit. Essentially, this allows for vehicles to be communicated their intended trajectory and as such, crossing time, as soon as they enter the intersection range, effectively reducing uncertainty and clearly communicating protocol behavior several seconds ($\geq 15\ s$) in advance.

While decentralized, i.e., V2V-based approaches, certainly have an easier time incorporating human drivers in a mixed-traffic environment [4], their large individual number of active participants can lead to substantial emergent behavior [60], greatly reducing the predictability of the system. Further, while a centralized RSU has a stationary position, the mobile nature of vehicles makes their perception and communication subject to occlusion issues, such as the *hidden terminal problem* [33].

10.3. Outlook

Intelligent intersections are a subdomain of autonomous driving and, as such, will continue to receive substantial research attention in the immediate future as the field evolves further. From the concepts discussed in this thesis, several directions of extension or improvement are possible. A, by no means exhaustive, list of potential further works is presented in this section.

Multi-intersection settings: A potential area of extension is to consider multi-intersection settings for the protocols proposed in this thesis. While dynamic traffic assignment (DTA) [2] [59] is widely used to adjust traditional control features such as dictating right of way and conventional traffic lights between multiple intersections, only few works such as [36] explicitly cover the implications for intelligent intersections when considering more than a single intersection. Further, existing such works are often confronted with Braess' paradox [19], where adding lanes to a traffic system unexpectedly reduces overall traffic flow.

Mixed traffic: While autonomous driving systems are expected to become more prevalent in the future, it is questionable whether it will ever be possible to truly replace *all* human-driven vehicles with autonomous vehicles.

Therefore, existing and future approaches have to be able to accommodate mixed traffic where human drivers are present. This introduces several additional factors and constraints, some of which are outlined below.

To include **human drivers**, additional sensors to perceive them and methods of human-machine-interaction to communicate with them or their vehicles are required. This leads to an overall performance reduction, as is the case for CSIP [6], where small losses in throughput compared to BRIP are incurred to make the system bearable to human participants who would otherwise be scared off by minuscule inter-vehicle distances of BRIP. This is also the case for FleXS-TP, as shown by the need to deactivate several safeguards concerning minimum distance and emergency braking as described in Section 8.1.6 and Table 8.2.

After increasing inter-vehicle distances in a similar fashion to account for human behavior and comfort, a possibility to allow human drivers to participate in the intersection is to clearly and intuitively communicate them their intended speed, which they simply have to obey to cross the intersection safely. This can be done by utilizing a heads-up display (HUD), which is projected into the field of view of the driver. Note that having this information displayed on a HUD is of utmost importance for vehicular traffic as a driving strategy focused on actively observing instruments shown on the dashboard is unfeasible and unsafe.

Especially at intersections in urban areas, **pedestrians** are a common part of existing intersections and their crossing and general behavior will have to be considered for future intelligent intersections. However, due to the naturally inconsistent flow and walking speeds of pedestrians, fixed traffic-light style schemes might not be universally efficient. Here, a query-based system, such as presented in [62], which utilizes a maximum pedestrian waiting time similar to the fairness-based measure of PB-LTR described in Chapter 6, can balance vehicular and pedestrian efficiency. The authors also discuss the challenges with connecting and detecting pedestrians, as not everyone might be willing to connect with a smartphone or sensor, requiring additional sensors or peripherals such as request buttons.

Due to these ongoing challenges in pedestrian-system communication and the increased number of collision points and the elevated vulnerability of pedestrians in vehicular environments, this remains a challenging and complex extension.

Including **cyclists** into autonomous driving systems combines the previously discussed challenges of both human drivers and pedestrians. Similar to human drivers, they share the road with autonomous vehicles (either on the same lane or on dedicated bike lanes), but conventional vehicular sensors such as radar, lidar or cameras might not correctly detect them as participants of traffic [70] — an issue that is shared by motorcyclists as well [77]. At the same time, cyclists and pedestrians also face the elevated vulnerability and challenges regarding system communication due to the lack of impact-absorbing material, sophisticated sensors and safety systems as well as visual signal devices such as blinkers — although there are efforts to increase cyclists visibility such as adding visual alerts to the cyclists and nearby vehicles [58].

To summarize, cyclists add a vast amount of new possible interactions to autonomous driving systems and might require substantial changes to both infrastructure and policy to facilitate a safe incorporation [37].

Bibliography

[1] G. Abu-Lebdeh and R. Benekohal. Development of Traffic Control and Queue Management Procedures for Oversaturated Arterials. *Transportation Research Record: Journal of the Transportation Research Board*, 1603:119–127, 1997.

[2] R. Aghamohammadi and J. A. Laval. Dynamic traffic assignment using the macroscopic fundamental diagram: A review of vehicular and pedestrian flow models. *Transportation Research Part B: Methodological*, 137:99–118, 2020.

[3] E. Andert, M. Khayatian, and A. Shrivastava. Crossroads: Time-sensitive autonomous intersection management technique. In *Proceedings of the 54th Annual Design Automation Conference 2017*, pages 1–6, 2017.

[4] S. Aoki, T. Higuchi, and O. Altintas. Cooperative perception with deep reinforcement learning for connected vehicles. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 328–334. IEEE, 2020.

[5] S. Aoki and R. Rajkumar. Dynamic intersections and self-driving vehicles. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 320–330. IEEE, 2018.

[6] S. Aoki and R. Rajkumar. CSIP: A synchronous protocol for automated vehicles at road intersections. *ACM Transactions on Cyber-Physical Systems*, 3(3):1–25, 2019.

[7] S. Aoki and R. Rajkumar. V2v-based synchronous intersection protocols for mixed traffic of human-driven and self-driving vehicles. In *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 1–11. IEEE, 2019.

[8] S. Aoki and R. Rajkumar. Safe intersection management with cooperative perception for mixed traffic of human-driven and autonomous vehicles. *IEEE Open Journal of Vehicular Technology*, 3:251–265, 2022.

[9] S. Aoki, I. Yamamoto, D. Shiotsuka, Y. Inoue, K. Tokuhiro, and K. Miwa. SuperDriverAI: Towards Design and Implementation for End-to-End Learning-Based Autonomous Driving. In *2023 IEEE Vehicular Networking Conference (VNC)*, pages 195–198. IEEE, 2023.

[10] T.-C. Au, S. Zhang, and P. Stone. Semi-autonomous intersection management. In *AAMAS*, pages 1451–1452, 2014.

[11] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige. Ballroom Intersection Protocol: Synchronous Autonomous Driving at Intersections. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2015 IEEE 21st International Conference on*. IEEE, 2015.

[12] R. Azimi, G. Bhatia, R. R. Rajkumar, and P. Mudalige. STIP: Spatio-temporal intersection protocols for autonomous vehicles. In *2014 ACM/IEEE international conference on cyber-physical systems (ICCPS)*, pages 1–12. IEEE, 2014.

[13] M. Bashiri and C. H. Fleming. A Platoon-Based Intersection Management System for Autonomous Vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017.

[14] M. Bashiri, H. Jafarzadeh, and C. H. Fleming. PAIM: Platoon-Based Autonomous Intersection Management. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018.

[15] K. Batz, S. Masse, W. Taylor, M. Soja, and T. Todd-Stone. Organizational Accidents and the Swiss Cheese Model: Learning from Air France flight 447. 2021.

[16] 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.

[17] F. Bianchi, A. C. Curry, and D. Hovy. Artificial Intelligence Accidents Waiting to Happen? *Journal of Artificial Intelligence Research*, 76:193–199, 2023.

[18] P. S. Bokare and A. K. Maurya. Acceleration-deceleration behaviour of various vehicle types. *Transportation research procedia*, 25:4733–4749, 2017.

[19] D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.

[20] A. Chan. Loss of control: "normal accidents" and ai systems. In *ICLR-21 Workshop on Responsible AI*, 2021.

[21] X. Chen, Y. Sun, Y. Ou, X. Zheng, Z. Wang, and M. Li. A conflict decision model based on game theory for intelligent vehicles at urban unsignalized intersections. *IEEE Access*, 8:189546–189555, 2020.

[22] Council Directive 96/53/EC of 25 July 1996 laying down for certain road vehicles circulating within the Community the maximum authorized dimensions in national and international traffic and the maximum authorized weights in international traffic. `http://data.europa.eu/eli/dir/1996/53/2019-08-14`, September 1996.

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

[24] R. Dedinsky, M. Khayatian, M. Mehrabian, and A. Shrivastava. A dependable detection mechanism for intersection management of connected autonomous vehicles (interactive presentation). In *Workshop on Autonomous Systems Design (ASD 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[25] K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *Autonomous Agents and Multiagent Systems, International Joint Conference on*, volume 3, pages 530–537. Citeseer, 2004.

[26] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 31:591–656, 2008.

[27] K. M. Dresner and P. Stone. Sharing the road: Autonomous vehicles meet human drivers. In *Ijcai*, volume 7, pages 1263–1268, 2007.

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

[29] European Car Sales Data. `http://carsalesbase.com/european-car-sales-data/`, June 2017.

[30] European Vehicle Market Statistics - Pocketbook 2016/2017. `http://www.theicct.org/sites/default/files/publications/ICCT_Pocketbook_2016.pdf`, June 2017.

[31] 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.

[32] M. Faris, M. Zanon, and P. Falcone. An Optimization-Based Dynamic Reordering Heuristic for Coordination of Vehicles in Mixed Traffic Intersections. *IEEE Transactions on Control Systems Technology*, 2024.

[33] C. L. Fullmer and J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. *ACM SIGCOMM Computer Communication Review*, 27(4):39–49, 1997.

[34] A. Goel, S. Ray, and N. Chandra. Intelligent Traffic Light System to Prioritized Emergency Purpose Vehicles Based on Wireless Sensor Network. *International Journal of Computer Applications*, 40(12):36–39, 2012.

[35] W. H. Harman. TCAS- A system for preventing midair collisions. *The Lincoln Laboratory Journal*, 2(3):437–457, 1989.

[36] M. Hausknecht, T.-C. Au, and P. Stone. Autonomous intersection management: Multi-intersection optimization. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4581–4586. IEEE, 2011.

[37] M. Hou, K. Mahadevan, S. Somanath, E. Sharlin, and L. Oehlberg. Autonomous vehicle-cyclist interaction: Peril and promise. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[38] IEEE Standard 802.11p. `https://www.ietf.org/mail-archive/web/its/current/pdfqf992dHy9x.pdf`, June 2010.

[39] Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth. Platoon-Based Multi-Agent Intersection Management for Connected Vehicle. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013.

[40] M. Khayatian, M. Mehrabian, and A. Shrivastava. RIM: Robust intersection management for connected autonomous vehicles. In *2018 IEEE Real-Time Systems Symposium (RTSS)*, pages 35–44. IEEE, 2018.

[41] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner. SUMO (Simulation of Urban MObility) - an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, pages 183–187, 2002.

[42] D. M. Kreps. Nash equilibrium. In *Game theory*, pages 167–177. Springer, 1989.

[43] D. Krishna Murthy and A. Masrur. Controlled Intra-Platoon Collisions for Emergency Braking in Close-Distance Driving Arrangements. In *Proceedings of the 24th Euromicro Conference on Digital Systems Design (DSD)*, Hsinchu (Taiwan), 2021.

[44] D. Li, G. Liu, and B. Xiao. Human-like driving decision at unsignalized intersections based on game theory. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 237(1):159–173, 2023.

[45] D.-J. Lin, M.-Y. Chen, H.-S. Chiang, and P. K. Sharma. Intelligent traffic accident prediction model for Internet of Vehicles with deep learning approach. *IEEE transactions on intelligent transportation systems*, 23(3):2340–2349, 2021.

[46] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner. Microscopic Traffic Simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582, 2018.

[47] X. Lu, H. Zhao, C. Li, B. Gao, and H. Chen. A game-theoretic approach on conflict resolution of autonomous vehicles at unsignalized intersections. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):12535–12548, 2023.

[48] R. E. Lyons and W. Vanderkulk. The use of triple-modular redundancy to improve computer reliability. *IBM journal of research and development*, 6(2):200–209, 1962.

[49] M. M. Maas. Regulating for'Normal AI Accidents' Operational Lessons for the Responsible Governance of Artificial Intelligence Deployment. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 223–228, 2018.

[50] H. V. Maddiboyina and V. S. Ponnapalli. Mamdani Fuzzy-Based Vehicular Grouping at the Intersection of Roads for Smart Transportation System. *International Journal of Intelligent Engineering Informatics*, 7(2-3), 2019.

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

[52] D. Markert and A. Masrur. A Two-Speed Synchronous Traffic Protocol for Intelligent Intersections: From Single-Vehicle to Platoon Crossing. *ACM Transactions on Cyber-Physical Systems*, 7(2):1–21, 2023.

[53] D. Markert and A. Masrur. A Flexible Synchronous Traffic Protocol for Future Intelligent Intersections. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2024.

[54] D. Markert, P. Parsch, and A. Masrur. Using Probabilistic Estimates to Guarantee Reliability in Crossroad VANETs. In *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. ACM, 2017.

[55] D. Markert, P. Parsch, and A. Masrur. Analyzing the Impact of Probabilistic Estimates on Communication Reliability at Intelligent Crossroads. In *In Proceedings of the 22nd Euromicro Conference on Digital Systems Design (DSD)*, volume 22, pages 206–2013, 2019.

[56] D. Markert, P. Parsch, and A. Masrur. Impact of Probabilistic Vehicle Estimates on Communication Reliability at Intelligent Crossroads. *Microprocessors and Microsystems*, 78, 2020.

[57] F. McLeod and N. Hounsell. Bus Priority at Traffic Signals — Evaluating Strategy Options. *Journal of Public Transportation*, 6(3):1, 2003.

[58] D. Mejia, S. Gomez, and F. Martinez. A low-cost wearable autonomous system for the protection of bicycle users. *International Journal of Advanced Computer Science and Applications*, 14(1), 2023.

[59] D. K. Merchant and G. L. Nemhauser. A model and an algorithm for the dynamic traffic assignment problems. *Transportation science*, 12(3):183–199, 1978.

[60] P. Morignot, O. Shagdar, and F. Nashashibi. Emergent behaviors and traffic density among heuristically-driven intelligent vehicles using V2V communication. In *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1017–1022. IEEE, 2014.

[61] J. E. Mosimann. On the Compound Multinomial Distribution, the Multivariate β-Distribution, and Correlations Among Proportions. *Biometrika*, 49:65–82, 1962.

[62] T. Niels, N. Mitrovic, K. Bogenberger, A. Stevanovic, and R. L. Bertini. Smart Intersection Management for Connected and Automated Vehicles and Pedestrians. In *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 1–10, 2019.

[63] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of Road Traffic Control Strategies. *Proceedings of the IEEE*, 91:2043–2067, 2003.

[64] P. Parsch and A. Masrur. A Reliability-Aware Medium Access Control for Unidirectional Time-Constrained WSNs. In *Proceedings of the International Conference on Real Time and Networks Systems (RTNS)*, 2015.

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

[66] C. Perrow. Normal accident at three mile island. *Society*, 18(5):17–26, 1981.

[67] C. Perrow. Normal accidents: Living with high-risk technologies. 1984.

[68] C. Perrow. *Normal accidents: Living with high risk technologies - Updated edition*. Princeton University Press, 2011.

[69] C. Perrow. Three Mile Island: a normal accident. In *The International Yearbook of Organization Studies 1981 (RLE: Organizations)*, pages 1–25. Routledge, 2013.

[70] S. Pettigrew, J. D. Nelson, and R. Norman. Autonomous vehicles and cycling: Policy implications and management issues. *Transportation research interdisciplinary perspectives*, 7:100188, 2020.

[71] Powered Two Wheeler Registrations in EU and EFTA Countries. `http://www.acem.eu/images/publiq/2015/2014-registrations-statistics.pdf`, June 2017.

[72] Z. Qin, A. Ji, Z. Sun, G. Wu, P. Hao, and X. Liao. Game theoretic application to intersection management: A literature review. *IEEE Transactions on Intelligent Vehicles*, 2024.

[73] J. Reason. Human error: models and management. *Bmj*, 320(7237):768–770, 2000.

[74] J. Reason, E. Hollnagel, and J. Paries. Revisiting the Swiss cheese model of accidents. *Journal of Clinical Engineering*, 27(4):110–115, 2006.

[75] S. Redana, Ö. Bulakci, A. Zafeiropoulos, A. Gavras, A. Tzanakaki, A. Albanese, A. Kousaridas, A. Weit, B. Sayadi, B. T. Jou, et al. 5G PPP Architecture Working Group: View on 5G Architecture. Version 3.0, 2019.

[76] E. Ruijters and M. Stoelinga. Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. *Computer science review*, 15:29–62, 2015.

[77] Z. B. Shabestari, A. Hosseininaveh, and F. Remondino. Motorcycle detection and collision warning using monocular images from a vehicle. *Remote Sensing*, 15(23):5548, 2023.

[78] G. Sharon and P. Stone. A protocol for mixed autonomous and human-operated vehicles at intersections. In *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*, pages 151–167. Springer, 2017.

[79] M. L. Sheffels. A fault-tolerant air data/inertial reference unit. *IEEE Aerospace and Electronic Systems Magazine*, 8(3):48–52, 1993.

[80] M. S. Shirazi, B. T. Morris, and S. Zhang. Intersection analysis using computer vision techniques with SUMO. *Intelligent Transportation Infrastructure*, 2:liad003, 2023.

[81] R. Tian, N. Li, I. Kolmanovsky, Y. Yildiz, and A. R. Girard. Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2211–2226, 2020.

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

[83] T. Williamson and N. A. Spencer. Development and operation of the traffic alert and collision avoidance system (TCAS). *Proceedings of the IEEE*, 77(11):1735–1744, 1989.

[84] L. Yang, X. Zhang, J. Gong, and J. Liu. The research of car-following model based on real-time maximum deceleration. *Mathematical Problems in Engineering*, 2015(1):642021, 2015.

[85] 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.

[86] 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.

[87] W. Yue, C. Li, Y. Chen, P. Duan, and G. Mao. What is the root cause of congestion in urban traffic networks: Road infrastructure or signal control? *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8662–8679, 2021.

[88] R. Zerod. 5.9 GHz V2X Modem Performance Challenges With Vehicle Integration. In *Proceedings of the ITS World Congress*, 2014.

[89] J. Zhao and A. F. Burke. Deep Learning for Planning and Control in Autonomous Vehicles. In D. M. Reyhanoglu, editor, *Modeling and Control of Autonomous Systems*, chapter 0. IntechOpen, Rijeka, 2025.

[90] M. Zweck and M. Schuch. Traffic light assistant: Applying cooperative ITS in European cities and vehicles. In *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 509–513. IEEE, 2013.

Appendix A.

Code Samples and Extra Material

A.1. Example Route File

```
<routes>
  <vType id="scarVHI" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="3" departSpeed="desired" maxSpeed="12.5"
       guiShape="passenger/van"/>
  <vType id="realcarVHI" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="5.0" departSpeed="desired" maxSpeed="12.5"/>
  <vType id="ecarVHI" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="5.0" departSpeed="desired" maxSpeed="12.5"
       guishape="passenger/wagon"/>
  <vType id="bcarVHI" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="5.0" departSpeed="desired" maxSpeed="12.5"
       guiShape="bus"/>
  <vType id="tcarVHI" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="8.0" departSpeed="desired" maxSpeed="12.5"
       guiShape="truck"/>
  <vType id="scarVLO" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="3" departSpeed="desired" maxSpeed="8.3"
       guiShape="passenger/van"/>
  <vType id="realcarVLO" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="5.0" departSpeed="desired" maxSpeed="8.3"/>
  <vType id="ecarVLO" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="5.0" departSpeed="desired" maxSpeed="8.3"
       guishape="passenger/wagon"/>
  <vType id="bcarVLO" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="5.0" departSpeed="desired" maxSpeed="8.3"
       guiShape="bus"/>
  <vType id="tcarVLO" minGap = "0" accel="5.5" speedDev="0" decel="5.5"
       sigma="0.0" length="8.0" departSpeed="desired" maxSpeed="8.3"
       guiShape="truck"/>
  <route id="NL" edges="IncomingNorth OutgoingEast" />
  <route id="NT" edges="IncomingNorth OutgoingSouth"/>
  <route id="NR" edges="IncomingNorth OutgoingWest" />
  <route id="EL" edges="IncomingEast  OutgoingSouth"/>
  <route id="ET" edges="IncomingEast  OutgoingWest" />
  <route id="ER" edges="IncomingEast  OutgoingNorth"/>
  <route id="SL" edges="IncomingSouth OutgoingWest" />
  <route id="ST" edges="IncomingSouth OutgoingNorth"/>
  <route id="SR" edges="IncomingSouth OutgoingEast" />
  <route id="WL" edges="IncomingWest  OutgoingNorth"/>
  <route id="WT" edges="IncomingWest  OutgoingEast" />
  <route id="WR" edges="IncomingWest  OutgoingSouth"/>
</routes>
```

Listing A.1: Example Route File — Note that some values have been truncated for readability

A.2. Randomized Traffic Route File Generation

```
def generate_routefile_randomized(self):
        cycle = self.sector/self.VLO    #length of a cycle
        NumberOfVehicles = 1000         #1000 vehicles
        #maximum random spacing (2 = heavy traffic, 5 = medium traffic)
        maximum_spacing = 5
        count = math.ceil(self.R/self.sector -1) #sectors in range R
        #initialize lane-based memory of where vehicles already are (in cycles)
        Blocked = [count,count,count,count,count,count,count,count]
        #Must be sorted in order of departure time
        vehiclelines = []                   #vehicle strings
        departindex = []                    #departure times (as sorting index)
        for i in range(NumberOfVehicles):   #for each vehicle
            vehType = randrange(5)          #randomize type
            route = randrange(12)           #randomize route
            spacing = randrange(maximum_spacing)   #random spacing on same lane

            Block = Blocked[lane_of[route]] + (2 + spacing)   #2 cycles + spacing
            Blocked[lane_of[route]] = Block #block specific lane for new gen.

            speedType = speed[route] #speed of chosen route (1=HI, 0=LO)
            Speed = self.VHI * speedType + self.VLO * (1 - speedType)

            arrival=newBlock*cycle    #arrival time at intersection border
            depart = arrival - count*self.sector/speed #Conside distance

            #Consider vehicle insertion offset
            length = self.vehTypeLengths[vehType]
            departTimeLengthOffset = depart + length/Speed

            routeName = TypeStrings[route]
            #Generate the vehicle line string for the route file
            vehiclelines.append(f'<vehicle id="Veh{i}_{routeName}_ML{length}"
            departLane="{lane}" type="{self.vehTypeStrings[vehType]}
            {SpeedStrings[speedType]}unsorted" route="{routeName}"
            depart="{departTimeLengthOffset}" departSpeed="max" />')
            #assign departure time for sorting
            departindex.append(departTimeLengthOffset)

            self.vehicleList.append(vehicle_info(i,route,vehType,depart,
            arrival,vehType))                           #Document vehicles
            self.unsortedTime.append(round(arrival, 4)) #For sorting

            filename = os.path.join(self.dirname, "data\\cross.rou.unsorted.xml")
            #generating the route file from the unsorted data (accidents!)
            with open(filename, "w") as routes:
                header = LTR.genRouteFileHeader(self)  #Generate the header
                print(f"{header}", file=routes)
                #Sort the routefile lines by departure times
                sortedvehiclelines = [vehiclelines for _, vehiclelines in
                sorted(zip(departindex,vehiclelines))]
                for el in sortedvehiclelines:
                    print(el, file=routes)             #Add vehicle lines
                print("</routes>", file=routes)        #Conclude routefile
#end of generate_routefile_randomized
```

Listing A.2: Routefile generation for randomized traffic

A.3. BRIP Type I Route File Generation

```
def generate_routefile_BRIPT1(self):
        #Generation steps - 250x4 = 1000 vehicles
        NumberOfVehicles = 250
        #Must be sorted in order of departure time
        vehiclelines = []               #vehicle strings
        departindex = []                #departure times (as sorting index)
        #Init
        Speed = self.VHI
        cycle = self.sector/self.VHI    #length of a cycle (only VHI)
        [...]
        counter = 0                     #switches between right and left lane

        for i in range(NumberOfVehicles):  #for each set
            lane = counter
            counter = (counter ^ 1)     #XOR to switch between 0 and 1

            time += cycle                   #increment time
            departTime = time - timeToCenter #Consider distance from insertion

            for n in range (4):                         #all 4 directions
                #n: 1 2 3 4 = north east south west, lane 0 = right, 1 = left
                route = self.RouteStrings[2 * n - lane]

                #Consider vehicle insertion offset
                departTimeLengthOffset = 1 + self.vehTypeLength/Speed

                #Generate the vehicle line string for the route file later
                vehiclelines.append(f'<vehicle id="Set{setNr}_{route}_T{lane}"
                departLane="{lane}" type="car" route="{route}" depart=
                "{departTimeLengthOffset}" departSpeed="{speed}"/>')
                #assign departure time for sorting
                departindex.append(departTimeLengthOffset)
                vehNr += 1                                  #another vehicle
            #end of for n in range (4)
            setNr += 1                                      #another set
        #end of for i in range(NumberOfVehicles)

        #grab routefile path
        filename = os.path.join(self.dirname, "data\\cross.rou.xml")

        #generating the route file from the data
        with open(filename, "w") as routes:
            header = LTR.genRouteFileHeader(self)     #Generate the header
            print(f"{header}", file=routes)
            #Sort the routefile lines by departure times
            sortedvehiclelines = [vehiclelines for _, vehiclelines in
            sorted(zip(departindex,vehiclelines))]
            for el in sortedvehiclelines:
                print(el, file=routes)                #Add vehicle lines
            print("</routes>", file=routes)           #Conclude routefile
#end of generate_routefile_BRIPT1
```

Listing A.3: Routefile generation for well-behaved traffic for BRIP Type I

A.4. Variable Vehicle Length Alone

For the sake of completeness, this section presents a brief evaluation of a fully homogeneous system from [54] with only very short vehicles driving through, as mentioned in Section 4.1. However, these considerations were revised and improved in [56], as documented in Section 9.1.

This system is assumed to have fully homogeneous traffic with length L only driving through in two-way synchronization with an inter-vehicle separation $D = S$, $R = 200\ m$, $S = 4\ m$, $L_{min} = 1.8\ m$ and $L_{max} = 7.8\ m$.

This leads to a deterministic single-lane set count following Eq. (4.1) of $n^{hg}_{max} = 21$. Similarly, the probabilistic minimum bound from Eq. (4.4) amounts to $n^{hg}_{min} = 13$. For two fully filled opposing lanes, this leads to the following possible vehicle counts in the entire intersection range:

$$[2 \cdot n^{hg}_{min}, 2 \cdot n^{hg}_{max}] = [26, 42]$$

Utilizing the probabilities for different vehicle lengths as shown in Fig. 4.2 in the multinomial distribution from Eq. (4.3) and considering the condition for valid combinations from Eq. (4.5) leads the probabilistic worst-case vehicle counts displayed in Table A.1 below:

One lane (n)	**Intersection ($2n$)**	**Probability**
21	42	≈0
20	40	1.1880E-33
19	38	5.6016E-22
18	36	1.6768E-12
17	34	5.7923E-05
16	32	0.59656894
15	30	0.40271516
14	28	6.5791E-04
13	26	6.6878E-08

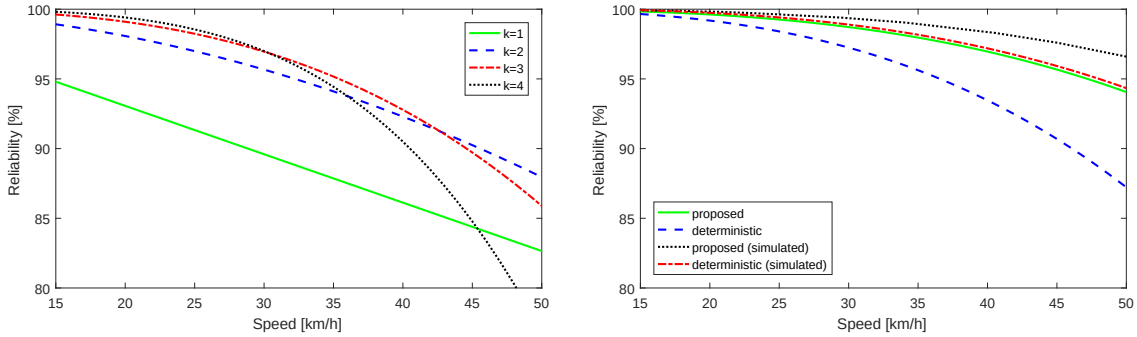
Table A.1.: Resulting worst-case numbers of vehicles with their corresponding probabilities for a fully homogeneous system.

Accumulating these probabilities shows that by accounting for the variability in length alone, the maximum number of vehicles at the intersection is ≤ 32 in 99.9942 % of all cases, or ≤ 34 in $(1 - 1.68 \times 10^{-10})$ % of all cases, compared to the absolute, deterministic worst-case vehicle count of 42.

This was further simulated as described in Section 9.1.2 via an OMNeT++ simulation framework [82], which allowed the recording of statistical data for at least 100,000 communication cycles for each graph. Channel models and parameters were taken from [88] and while it was assumed that there was no external interference present, this can be easily added as described in [65].

This allows for comparisons with the more straightforward deterministic worst case detailed in Section 4.1.1. The results broadly match those in Section 9.1.2, so they are only briefly summarized here.

Determining k: The underlying MAC layer is based on transmitting k request messages within the contention phase. Due to k's non-linearity and dependency on the other parameters, it is determined experimentally following [64], see Fig. A.1a. To achieve good performance over a high range of speeds, $k = 3$ is selected for the subsequent experiments.

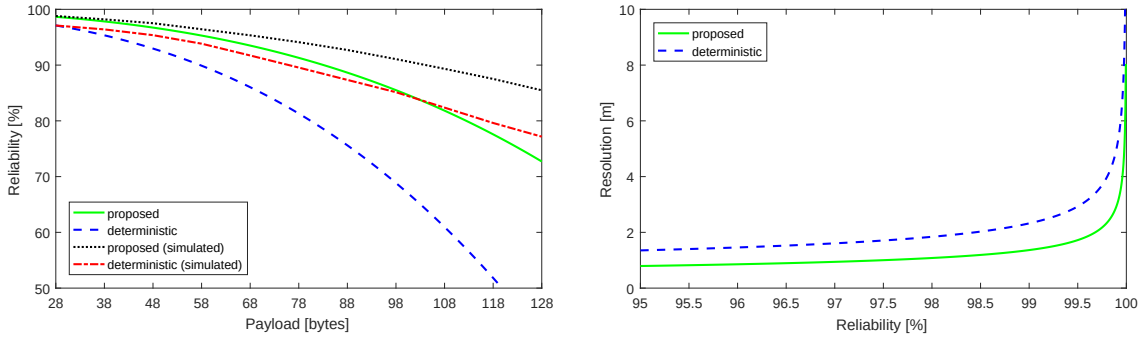


(a) Relation between the speed of a vehicle and the resulting worst-case transmission reliability for different k.

(b) Relation between the speed of a vehicle and its general transmission reliability.

Figure A.1.: Relation between the speed and transmission reliability.

Reliability vs speed: Following Eq. (4.19), preciser estimates for the vehicle number n lead to higher network reliability, both for simulated and non-simulated *proposed* and *deterministic* approaches. Note that the simulated curves show the average reliability, which is typically higher than the computed worst-case reliability as per (4.18) and (4.19).



(a) Relation between the payload size of a request message and the transmission reliability.

(b) Relation between transmission reliability and the required physical resolution, i.e, the distance a vehicle travels at 50 km/h to achieve this reliability.

Figure A.2.: Impact of payload size on reliability and resulting physical resolution.

Reliability vs payload: As expected, an increasing payload size reduces the possible reliability, see Fig. A.2a. That is due to longer messages being more vulnerable to interference by another packet. This effect is slightly non-linear due to channel saturation effects. Again, the *proposed* curves show a considerable improvement over the *deterministic* one.

Reliability vs physical resolution: When a vehicle travels faster through the intersection, it has less time to transmit its data within a given distance, e.g, 1 m. Fig. A.2b shows that physical resolution increases only very slowly for rising reliability until starting to strongly increase from a reliability of 99 % onward and getting unbounded. That is, reliabilities close to 100 % are hard to achieve and, therefore, come at high costs when using asynchronous protocols.

Summary of Findings

Confining strategies for intelligent intersections to a specific type of traffic or infrastructure greatly restricts their applicability in the real world. To alleviate this, this thesis proposes several techniques with the goal of bridging these gaps and making these protocols applicable to a wider range of settings. Specifically, this thesis proposed a probabilistic modeling of communication reliability to reduce deterministic pessimism, as well as three traffic protocols: i) SV-LTR to increase space efficiency, ii) PB-LTR to apply the concept of platooning to intelligent intersections and iii) FleXS-TP to increase flexibility towards traffic composition.

Reducing Deterministic Pessimism

Ensuring communication reliability between vehicles and the corresponding road side unit (RSU) requires knowledge of the maximum number of vehicles in the system to assess interference. By using a realistic, probabilistic vehicle length distribution instead of assuming longest possible vehicle allows for a reduction in deterministic pessimism and the following overdesign by using stepwise, probabilistic worst cases.

Infrastructure-Agnostic Design

Treating extraordinarily large vehicles as exception instead of the norm leads to an uncoupling of the traffic protocols sector size S from the actual traffic, as introduced in the space-efficient traffic protocol SV-LTR and also used later in PB-LTR and FleXS-TP. Instead, the existing infrastructure now provides the sector size S, allowing for infrastructure-agnostic design where the traffic protocol can be applied to any given infrastructure with the existing geometry parameterizing the protocol, alleviating the need for great modifications of existing intersections, while providing competitive or even greater throughput.

Platooning for Well-Behaved Traffic

The traffic protocol PB-LTR extends SV-LTR's single-vehicle maneuvers to platoons consecutively performing the same maneuver, enabling shorter inter-vehicle distances and reducing downtime from maneuver transitions. To ensure all vehicles eventually cross the intersection, a maximum blocking time regularly enforces maneuver changes, guaranteeing fairness across all directions. Under well-behaved traffic, PB-LTR can further increase throughput.

Increasing Flexibility towards Traffic Composition

By maintaining a synchronous strategy at heart, but scheduling vehicles on a vehicle-by-vehicle basis while still allowing for situational synchronicity, the traffic protocol FleXS-TP can harness both the benefits of synchronous traffic protocols for well-behaved traffic, as well as the flexibility of asynchronous traffic protocols under randomized traffic, achieving substantial throughput in both settings — combining the best of asynchronous and synchronous strategies.

Daniel Markert — Curriculum Vitae

Contact Information

Dept. of Computer Science
Technische Universität Chemnitz
Straße der Nationen 62
09111 Chemnitz, Germany

Office: +49 371 531 34946
Mobile: +49 172 836 4598
Email: d.markert@outlook.de

Research Interest

My research interests include open-ended cyber-physical systems with a focus on strategies for intelligent intersections, especially synchronous traffic protocols and their applicability to existing settings and infrastructures. Recent research efforts also include WiFi-sensing and its use to predict driver characteristics inside vehicles using machine learning.

Education

2017 - 2025 PhD, Computer Science - TU Chemnitz, Germany
Field: Cyber-Physical Systems and Intelligent Intersections

2014 - 2017 M.Sc., Computer Science - TU Chemnitz, Germany
Field: Automotive Software Engineering

Teaching

- Computer Organisation (German)
- Computer Architecture (German)
- Hardware Development with VHDL (English)

Employment History

2017 - 2025 Scientific Researcher - Computer Architectures and Systems - TU Chemnitz, Germany
Scope: Embedded Systems, Cyber-Physical Systems and Intelligent Intersections, WiFi-Sensing

Language Skills

- Native: German
- Fluent: English

List of own publications

The following publications were made in the context of this work.

Journal publications

- Daniel Markert, Philip Parsch and Alejandro Masrur, *Impact of Probabilistic Vehicle Estimates on Communication Reliability at Intelligent Crossroads*, In Microprocessors and Microsystems: Embedded Hardware Design (MICPRO). vol. 78, Elsevier, 2020

- Daniel Markert and Alejandro Masrur, *A Two-Speed Synchronous Traffic Protocol for Intelligent Crossroads: from Single-Vehicle to Platoon Crossing*, In Transactions on Cyber-Physical Systems. vol 7(2), pp. 1-21. ACM, 2023

Peer-reviewed conferences (acceptance ratio < 30 %)

- Daniel Markert, Philip Parsch and Alejandro Masrur, *Using Probabilistic Estimates to Guarantee Reliability in Crossroad VANETs*, In Proceedings of the 6th ACM Symposium on Developement and Analysis of Intelligent Vehicular Networks and Applications (DIVANet). pp. 135-142, ACM, 2017, **Best Poster Award**

- Daniel Markert, Philip Parsch and Alejandro Masrur, *Analyzing the Impact of Probabilistic Estimates on Communication Reliability at Intelligent Crossroads*, In Proceedings of the 22nd Euromicro Conference on Digital Systems Design (DSD). vol. 22, pp. 206-213, 2019

- Daniel Markert and Alejandro Masrur, *Space-Efficient Traffic Protocols for Intelligent Crossroads*, In 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 1099-1104, IEEE, 2019

- Bijin Muthiyackal Abraham, Christian Drescher, Daniel Markert, Ana Cecilia Perez Grassi and Alejandro Masrur, *Characterizing Road Maps for Vehicle Endurance Testing with Machine Learning*, In 35th IEEE Intelligent Vehicles Symposium (IV). 2024

- Daniel Markert and Alejandro Masrur, *A Flexible Synchronous Traffic Protocol for Future Intelligent Intersections*, In 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2024

- Mohamed Abouelnaga, Mathias Haberjahn, Daniel Markert, and Alejandro Masrur, *Hardware-Compatible Deep Reinforcement Learning-Based Lateral Trajectory Controller*, In 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2024

- Daniel Markert, Estelle Wünsche, Hussein Kerdi, Erik Leo Hass, Ian Robert Blake, Yannic Warias, Ana Perez Grassi and Alejandro Masrur, *WiSe4Car: A Dataset for WiFi Sensing in Automotive Vehicles*, In IEEE International Conference on Omni-Layer Intelligent Systems (COINS), ***accepted for publication*** , 2025

Erklärungen

☒ Ich versichere, dass die vorgelegte Arbeit weder im Inland noch im Ausland in gleicher oder in ähnlicher Form einer anderen Prüfungsbehörde zum Zwecke einer Promotion oder eines anderen Prüfungsverfahren vorgelegt wurde und auch noch nicht veröffentlicht wurde.

☐ Es fand ein früheres Promotionsverfahren statt. nein

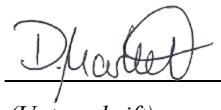
(bei ja) Thema:

Bescheid: *Zeit:*

Hochschule:

☒ Ich versichere, dass die vorliegende Arbeit ohne unzulässige Hilfe und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken in der Arbeit als solche kenntlich gemacht sind.

☒ Ich versichere, dass weitere Personen bei der geistigen Herstellung der vorliegenden Arbeit nicht beteiligt waren, insbesondere auch nicht die Hilfe eines Promotionsberaters in Anspruch genommen wurde, und dass Dritte vom Bewerber weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.



(Unterschrift)

20.06.2025

(Datum)