Western | Graduate&PostdoctoralStudies

Western University
Scholarship@Western

Electronic Thesis and Dissertation Repository

6-8-2020 12:00 PM

Vehicle Networks: Statistical and Game Theoretic Approaches to Their Evaluation and Design

Gleb Dubosarskii, *The University of Western Ontario*

Supervisor: Dr. Serguei Primak, *The University of Western Ontario*
Co-Supervisor: Dr. Xianbin Wang, *The University of Western Ontario*
A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical and Computer Engineering
© Gleb Dubosarskii 2020

Follow this and additional works at: https://ir.lib.uwo.ca/etd

 Part of the Systems and Communications Commons

Abstract

Vehicle ad hoc networks (VANETs) have become a popular topic in modern research. The main advantages of these networks include: improved security, traffic optimization, and infotainment. However, deploying such networks in practice requires extensive infrastructure. To estimate the network load, one needs to have information about the network, such as the number of clusters, cluster size, etc. Since VANETs are formed by vehicles that rapidly change their location, the network topology is constantly changing, making its analysis by deterministic methods impossible. Therefore, in this dissertation, we use probability theory methods to obtain probability distributions of such fundamental network properties, such as the number of clusters, cluster size, and the number of disconnected vehicles in the case in which the vehicles are located on a highway. In previous articles, some of these characteristics are obtained only in terms of average values, while the total distributions remained unknown. The distribution of the largest cluster size is an important characteristic of the network. It is derived in the dissertation for the first time. We also study the distribution of the number of clusters and the size of the average cluster in the case of a 2D map with an almost arbitrary road topology. To the best of our knowledge, these results are the first for such a general map case.

Studying these properties raises a number of new questions about how these network properties change over time. We obtain distributions of the network characteristics, such as the duration of communication between vehicles, and the duration of cluster existence. We also derive the probability that a cluster exists between two time moments, as well as other network properties. The obtained distributions are new in the case of the Markov channel model. The results regarding the distribution of cluster lifetime and the probability of cluster existence between two fixed time moments are obtained in the literature for the first time.

This dissertation also addresses the security aspect of VANET. We consider single and multichannel anti-jamming games in the case in which two communicating vehicles are being pursued by the jammer, which tries to disrupt the communication. The optimal strategies of the vehicles and the jammer are described as the Nash equilibrium of this game. We prove

theorems that express Nash equilibrium through communication parameters. The considered model with quadratic power term is new as well as the results regarding the Nash equilibrium in the single and multichannel cases. We also first examine performance of such state-of-the-art machine learning algorithms as Dueling Q-learning and Double Q-learning, which by trial and error, successfully converge to the Nash equilibrium, deduced theoretically.

Lay Summary

The subject of this dissertation is vehicle networks. The wide deployment of the vehicle networks will happen in this decade, so this topic attracts increased scientific interest. It is assumed that the cars will share information about traffic accidents, weather conditions, their coordinates, speed, acceleration, etc. The vehicle networks will also provide Internet access, video and audio streaming, and other services. All this data will be used for traffic optimization, driving safety improvement, and passenger entertainment. However, the vehicle networks are unstable due to the high car speeds, so the vehicles are organized into small groups that share information. Such groups are called clusters and are the object of study in the first part of the dissertation. Statistical data related to such network characteristic as cluster size helps to estimate the network load. Network load prediction is important for installing antennas along the roads that support vehicle networks. We use the probability theory methods to study such characteristics as the number of clusters, cluster size, and the number of disconnected vehicles in the case in which the vehicles are located on a highway and 2D map.

The second part of the dissertation is devoted to the security aspect of the vehicle networks. We consider a game in which a jammer chases two vehicles in order to disrupt their communication. We consider both single-channel and multi-channel cases. In the multi-channel case, it is assumed that the vehicles change the communication channel in order to avoid channel attacks. We derive theorems that express the optimal vehicle and jammer strategies through the communication channel parameters. However, in practice, some of the communication parameters are unknown, which narrows the scope of applicability of the obtained theorems. Therefore, we also examine machine learning algorithms that by trial and error converge to the theoretically obtained optimal strategies.

CO-AUTHORSHIP

This thesis has been prepared in accordance with the regulations for a Monograph thesis as stipulated by the School of Graduate and Postdoctoral Studies of the University of Western Ontario. All analytical work and simulations were carried out by G. Dubosarskii under the close supervision of Dr. S. Primak. The results of this thesis are published in journals and conference proceedings co-authored by Dr. S. Primak and Dr. X. Wang. Dr. X. Wang helped to improve the final text of the articles and made several useful recommendations about their content.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my scientific supervisor Dr. S. Primak for guidance and encouragement, without his close supervision, this thesis would not have been possible. I would also like to thank my co-supervisor Dr. X. Wang for his attention to my work, improvements to my articles, and offer of useful suggestions.

Finally, I would like to thank my parents Alexander Dubosarskii and Olga Ancharova who have continued to believe in me all these years.

Contents

List of Figures

List of Tables

List of Symbols

p	probability of connection between two consecutive vehicles
G_T	transmit antenna gain
G_R	receive antenna gain
P_{tx}	transmit power
α	path loss exponent
K	constant associated with the path loss model
d	distance between cars
C	speed of light
W	thermal noise power
k	Boltzmann constant, $k = 1.38 \times 10^{-23} J/K$
f_c	carrier frequency
T_0	room temperature
B	transmission bandwidth
$f_\gamma(x)$	Signal-to-noise ratio probability density function
$f_d(x)$	intervehicle distance probability density function
$\binom{k}{s}$	binomial coefficient
$coeff_{x^n} f(x)$	coefficient of the term x^n in series $f(x)$
$clust(N)$	number of clusters of vehicle network N
$count(r; N)$	number of clusters of vehicle network N having size r
$Num(s, k)$	number of networks having k clusters, under condition that s of them have size r
C_E	Euler constant, $C_E = 0.577\ldots$

ClustNum	number of clusters in the network
ClustSize	size of the cluster
LargestClust	size of the largest cluster
IdleCars	number of disconnected vehicles
λ	the average SNR over the fading channel between two consecutive vehicles
A	signal amplitude
$\overline{A}$	amplitude threshold
p_G	probability that $A \geq \overline{A}$
p_B	probability that $A < \overline{A}$
f_D	Doppler shift
v	vehicle velocity
f_c	signal frequency
c	velocity of light
q	probability that channel between two consecutive vehicles goes from the state $A < \overline{A}$ to the state $A \geq \overline{A}$
q'	probability that the channel between two consecutive vehicles goes from the state $A \geq \overline{A}$ to the state $A < \overline{A}$
ρ	vehicle density
L	total length of roads
n	vehicle number
r	number of roads on the map
tr_j	number of lanes that pass through traffic light j, but do not end there
tr	sum of t_j taken over all traffic lights j
$AvClustS\,ize$	average network cluster size on 2D map

p'	probability of a successful connection between consecutive vehicles moving along the same road in the same direction and located on opposite sides of the intersection
p''	probability of a successful connection between consecutive vehicles moving along the same road in the same direction and approaching the same intersection
x_i	coordinate of i-th car
v_i	velocity of i-th car
a	maximum acceleration
δ	acceleration exponent
b	vehicle deceleration
s_0	minimum gap between vehicles
v_0	desired speed
T	time gap between consecutive vehicles
s_i	bumper-to-bumper distance between car i and next riding car
l_i	length of car i
Δv_i	speed difference between speed of car i and speed of car $i-1$
$sizex \times sizey$	size of the map (simulator in Chapter 4)
car_num	number of cars (simulator in Chapter 4)
$block\_num\_x$	number of blocks along the x axes (simulator in Chapter 4)
$block\_num\_y$	number of blocks along the y axes (simulator in Chapter 4)

min_interveh_dist	minimum acceptable distance between vehicles (simulator in Chapter 4)
car_num	number of cars (simulator in Chapter 4)
min_interveh_dist	minimum intervehicle distance (simulator in Chapter 4)
σ^2	noise power
h_{car}^2	vehicle channel power gain
h_J^2	jammer channel power gain
x	vehicle power in anti-jamming game
y	jammer power in anti-jamming game
y_k	jammer power in multichannel anti-jamming game transmitted through k-th channel
$SINR$	signal-to-noise-plus-interference ratio
m	channel number in multichannel anti-jamming game
$PmaxC$	maximum vehicle power
$PmaxJ$	maximum jammer power
u_{car}	vehicle utility function
u_J	jammer utility function
C_{car}	vehicle transmission cost
C_J	jammer transmission cost
γ	discount-rate in general reinforcement learning algorithm
α	learning rate
a	agent action
s	agent state
r_k	agent reward at step k
ε	probability of choosing a random action in Q-learning algorithm

ε_0 starting value of exponential ε decay

ε_∞ limiting value of exponential ε decay

Chapter 1

INTRODUCTION AND LITERATURE REVIEW

1.1   Vehicle networks and other advances in transport

Over the last decade, the main efforts related to transport development have been aimed at creating partially or even fully autonomous vehicles. Corporations such as Tesla, Waymo, Uber, Lyft, Nvidia and many others have developed and tested autonomous vehicle autopilots [1, 63]. The essence of an autopilot is that it analyzes the data obtained from cameras and various other sensors (such as LIDAR and radar) and decides on the further movement of the car. In this process, machine vision [67, 94, 106, 107] plays a fundamental role, which distinguishes people and various objects on the road, allowing the car to avoid any collision. However, despite the fact that the large-scale deployment of autonomous driving programs began more than ten years ago and significant progress having been made in this area, full vehicle autonomy has not yet been achieved.

Another direction in improving transportation quality is Vehicular Adhoc Networks (VANETs). VANETs consist of high-speed and high mobility vehicles, which form spontaneous networks in order to transmit and share information. Such networks are needed to improve transportation
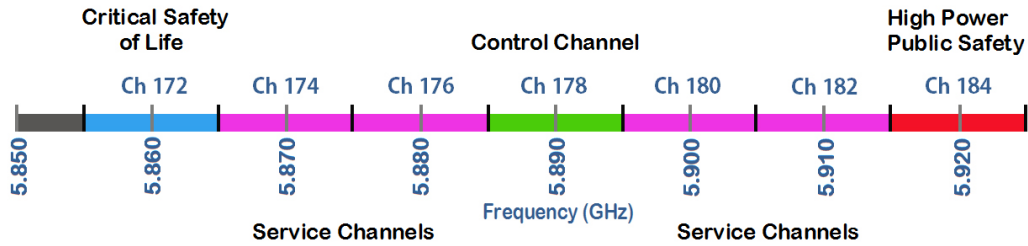
Figure 1.1: Layout of DRSC 5.9 GHz Frequency Band Spectrum.

safety, reduce the number of traffic accidents, improve traffic flow, reduce traffic congestion, and provide infotainment. To implement such networks, Dedicated Short Range Communication (DSRC) was developed. This technology provides reliable and high-speed communication between vehicles, as well as between vehicles and the surrounding infrastructure. For the implementation of Dedicated Short-Range Communications (DSRC) [22], the United States Federal Communication Commission allocated the 75 MHz spectrum in the 5.9 GHz band in 1999. DSRC spectrum is divided into seven 10 MHz channels, as shown in Figure 1.1. Among these channels, six are used to send service messages, and channel 178 is reserved for safety and control messages [44].

The main types of VANET communication are Vehicle-to-Vehicle (V2V) and Vehicle-to-RSU (V2R). The Vehicle-to-RSU communication scenario assumes vehicles communicate with Road Side Units (RSUs) located along roads. The distribution of packets in VANET occurs through multi-hop propagation, during which the vehicles send received messages between themselves until the messages reach the recipient. Another important application of VANET is platooning [42, 75, 96], which is that the vehicles moving in the same direction are organized in communicating groups. Platoons reduce intervehicle distance and provide data exchange with neighboring vehicles with little or no help from infrastructure.

The following types of messages are distributed in VANET [19, 26, 41]: safety messages, traffic management data, and infotainment data. Safety Messages are needed to reduce the number of accidents. Such messages include information about the location of the nearest cars, traffic lights, weather conditions, and information about traffic incidents, etc. In the event
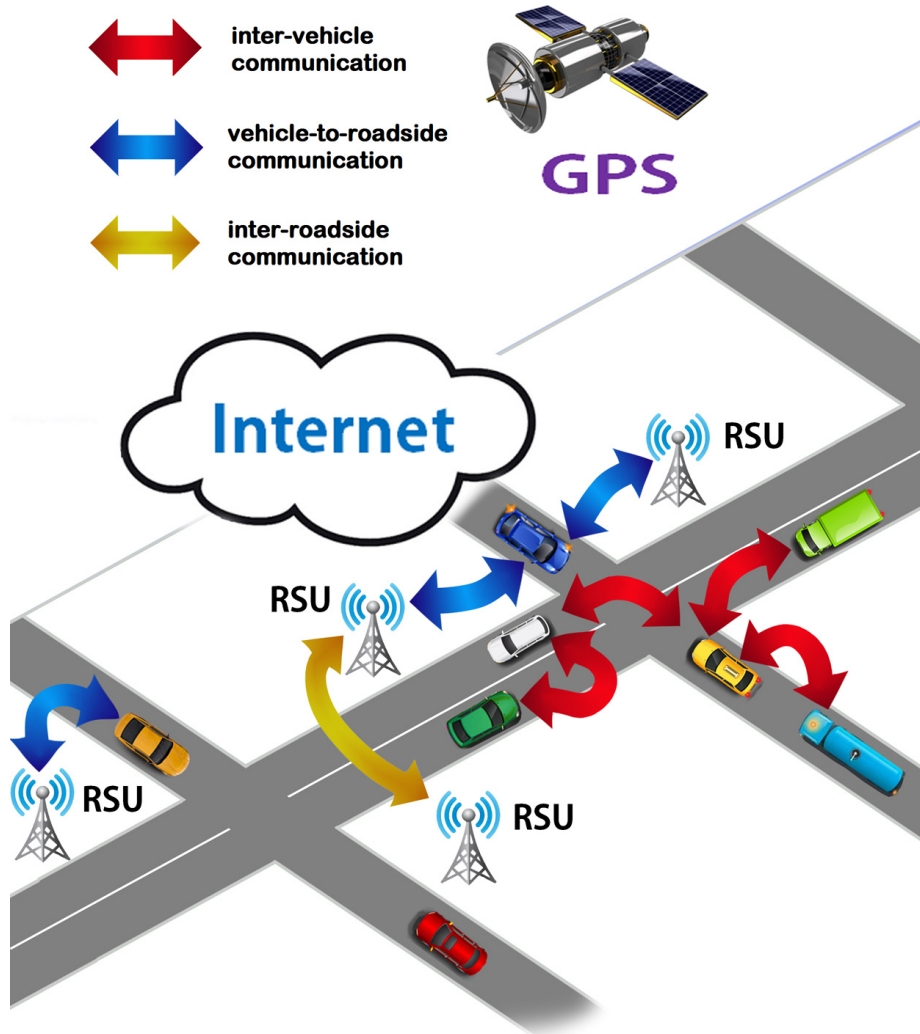
Figure 1.2: VANET communications.

of a malfunction, the vehicle transmits information about its coordinates, speed, and accelera-
tion in order to notify other network vehicles and prevent possible collisions. Such measures
are especially important in the case of limited visibility. Non-safety messages are divided into
traffic management and infotainment data. Traffic management messages are needed to opti-
mize traffic flow and reduce traffic congestion. Infotainment data includes information about
nearby restaurants, attractions, Internet access, games, music, and other media content.

1.2 Statistical analysis of VANET

This dissertation is devoted to the study of the statistical properties of VANET. Before considering articles on this topic, let us discuss the importance of statistical network analysis.

Statistical network data, such as the number of clusters and the cluster size can be useful in estimating the load on a network, which is an important information for infrastructure deployment. Another application of statistical information about cluster size is related to a scenario in which one of the cluster vehicles is infected with a virus. If one of the cars is infected, then by communicating with the cluster vehicles it eventually infects all other vehicles. Thus, statistical information about the size of the cluster allows for the assessment of damage to the network.

Figure 1.3: Clustering in VANET.

Analysis of the statistical properties of VANET is challenging, as the nodes of the network are highly mobile, and topology constantly changes. From the above, it follows that deterministic methods are not applicable for network analysis, therefore, analysis is carried out in terms of probability theory. However, the case of an arbitrary two-dimensional map is difficult to analyze due to a large number of variables, such as the routing of the cars, the states of traffic lights, the driving model on the road, so, in most articles, the authors limit themselves to the case of a highway.

Each article strikes a balance between the complexity of the model, and the depth of the results. As can be seen from the following literature review, some authors give preference to complex models, but derive only basic characteristics of the network, such as communication
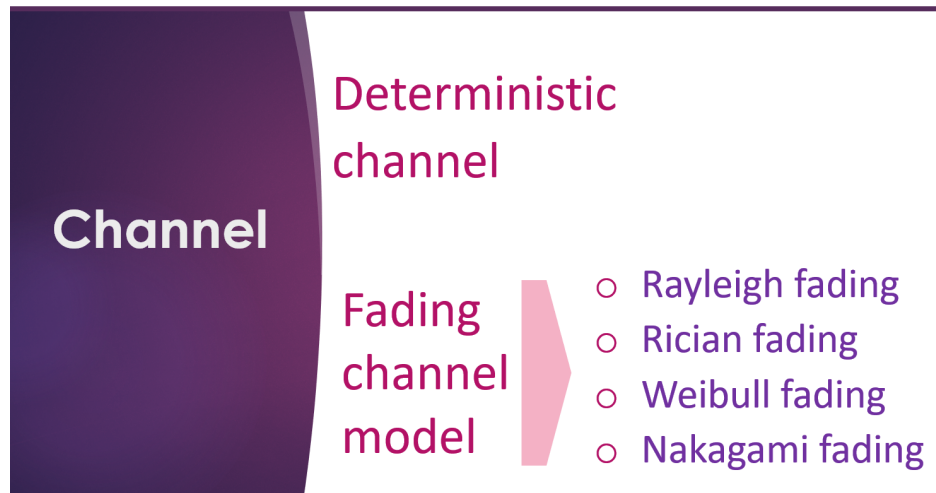
Figure 1.4: Main channel fading models.

probability, while other authors tend to simplify the model, but derive more nuanced characteristics such as the cluster number distribution, and cluster size.

There are a number of articles that consider the deterministic channel model, such as [47, 87, 108]. This model implies that as soon as the vehicles are located within a certain range, they can always establish a connection, otherwise they are disconnected. This channel model is a significant simplification compared to probabilistic fading channel models such as Rayleigh, Rician, and Weibull fading channels. Let us discuss these articles in more detail. In [87], under the assumption of a known distribution of distance between vehicles, the authors derive distribution of propagation distance from a fixed node, as well as its mean value and variance. The authors also derive analytical expression for the average number of vehicles involved in this information propagation process. In [47], network connectivity is studied in the case in which the vehicles are distributed between two consecutive RSUs. Assuming that the intervehicle distance has exponential distribution, the authors obtain the average value of the cluster size, and the probability that the vehicle network is fully connected. They introduce the concept of critical network size as the average network size necessary for vehicles to be connected and derive exact and asymptotic formulas for average critical network size. In [108] Laplace transform of the probability distribution of connectivity distance is derived, the explicit formula

for its expected value, as well as the expected value of the vehicle number in a platoon. In [98], under the assumption that intervehicle distance has exponential distribution, the probability of full network connectivity is obtained. The author defines outage connectivity probability as the probability that there is at least one disconnected node in the network, and outage transmission range as the range in which outage probability reaches a certain value. The approximated expressions for outage transmission range are derived and verified through simulations.

The following articles investigate connectivity in the case of a probabilistic fading channel, which is more relevant to practical needs, but at the same time, it introduces an additional level of complexity to the model. In [10, 17, 54], the authors consider several channel models such as Rayleigh, Rician, Weibull, and Nakagami fading channels. The authors did not delve into the more subtle properties of the network and derive only probabilities of connection between neighboring vehicles and the probability of full network connectivity for each of these models. Article [59] discusses a two-way street scenario. Under the assumption that the cars are moving in two opposite directions and the Nakagami fading channel model of vehicle communication, the probability of a successful multi-hop message transmission from one vehicle to another is deduced.

Several articles discuss the probabilistic model of communications with RSUs. These articles include the already discussed article [47], as well as article [60]. In [60], assuming a general fading channel and an exponential distribution of the distance between the vehicles, the authors deduce the probability of connection with the base station in the case of one-hop and two-hop communication.

Some articles discuss a more complex communication model, but this leads to the fact that such network parameters as the cluster size and the number of clusters cannot be derived due to the high complexity of the model. Such articles include [88], in which, along with the Rayleigh fading channel model, the authors suggest the Markov model of packet transmission, and the Physical Layer decoding failure model. The authors also assume that vehicles in clusters only communicate with the main vehicle and do not consider multi-hop propagation. Since the

model is complicated by numerous parameters, the authors derive only average packet loss probability and probability of connection.

We also note that in practice, not only the knowledge of the average cluster size is necessary, but also the distribution of the maximum cluster size. It characterizes the maximum load on the network in a certain part of the road. To the best of our knowledge, our results are the first in this field. It is also necessary to understand how the statistical characteristics change over time in terms of link duration, and average time of cluster existence etc. Unfortunately, relatively little attention has been devoted to these problems in the literature, and the question of the distribution of the lifetime of a cluster had not been studied before our results. Typically, the articles study the duration of vehicle communication in various scenarios. In [45], a case in which a signal is transmitted to a vehicle moving in the opposite direction and then sent back to the initial side is investigated. The probability of such a successful two-hop connection is derived under the assumption that the vehicles move at a constant speed. Article [104] focuses on the study of link duration in the specific case, in which the speed of the vehicles increases linearly, reaches a limit, and then remains constant. In [91], a link duration formula is derived under the assumption that the speeds of the vehicles are distributed according to the normal distribution with known mean value and variance. In [56], the authors derive the integral formula for link duration and use it to simulate message routing. It is worth mentioning that the statistical approach to estimating link duration is not the only possible methodology. For example, in [3], based on simulations, the authors convincingly prove that a neural network is able to predict link duration with a high degree of accuracy.

1.3 Anti-jamming game

Communication networks are vulnerable to various attacks by hostile entities which aim to disrupt ongoing communication, or steal private information being distributed in the network. Attacks on such devices include jamming attacks, eavesdropping attacks, and data falsification

attacks, etc. These attacks significantly degrade or even completely disrupt communications and reduce network security. Since this dissertation in Chapter 5 addresses an anti-jamming game, we concentrate on this subject in more detail. In this case, the considered network could include mobile phones, computers, and communicating vehicles. With the advent of smart homes, smart home devices and robots can be added to this list. In the case of VANET, jammer attacks are aimed at interrupting communication between cars, as well as between cars and Road Side Units (RSUs) located along roads. This scenario is more challenging because the car's connection is less stable due to the high speeds of the vehicles and constant changes in the network's topology.

Game theory has demonstrated its effectiveness in finding optimal strategies for network devices, allowing for improvements in the quality of communication at the time of an attack. It is especially effective against adaptive jammers, which adapt to the communication parameters in order to do maximum harm. In this setting, a jammer attack is considered as an antagonistic game between one or more cooperating devices and one or more jammers. In literature relating to the anti-jamming game, two main types of such games are recognized. The first kind of game involves devices and the jammer making moves at the same time. Thus, the jammer is not aware of the current device's communication state. In this situation, the optimal strategy of the device can be found as the Nash equilibrium of the game [8]. Another type of game is the so-called Stackelberg game [80]. In this game, it is assumed that the jammer is aware of the current device's state and adapts its strategy accordingly. In such a situation, the Nash equilibrium is no longer the optimal strategy, and the optimal strategy can be found as Stackelberg equilibrium.

Anti-jamming game considered in the game setting is an important area. Its main advantage is the presence of multi-parameter utility functions, which objectively measure performance of the devices. In addition, such games allow for multiagent setting in which several devices cooperate and find the optimal strategy for the whole group. The advantage of this approach is the ability to find the optimal strategy expressed through the network parameters.

Since the channel parameters such as channel gains and noise power are mostly unknown,
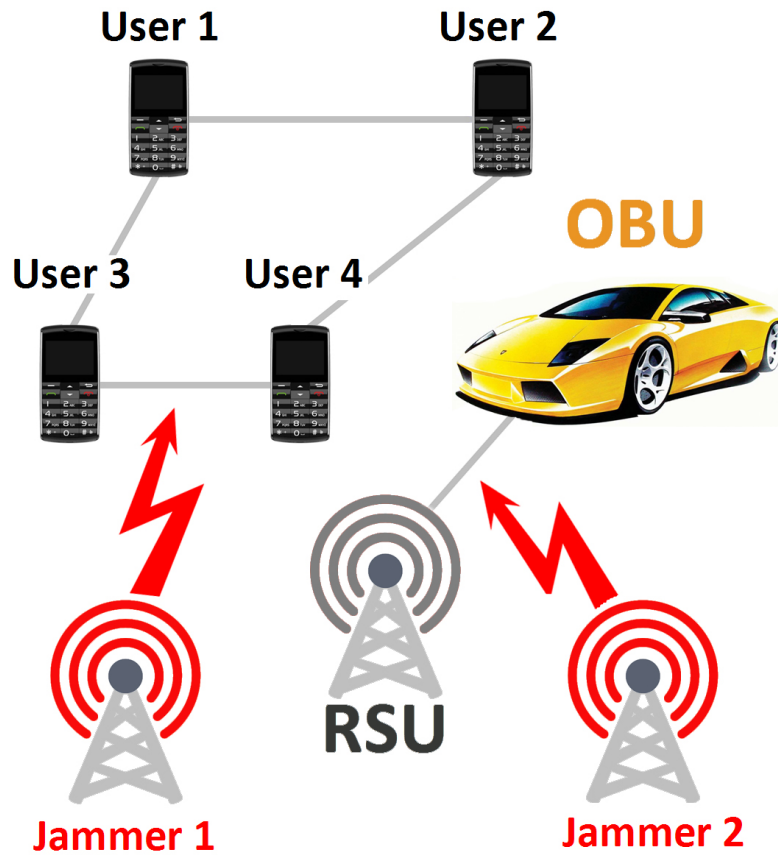
Figure 1.5: Network with two jammers.

as well as the geographical location of the jammer itself, it is almost impossible to consider anti-jamming game in an exact mathematical setting. Therefore, the use of machine learning methods to find the optimal communication strategy is widespread in literature [5, 9, 30, 34, 52, 53, 81, 101, 102, 105]. The essence of this method is that after sending the message, the sending device receives feedback information about the quality of the communication, such as SINR, and on the basis of this data corrects the transmission parameters. The most popular machine learning method for the anti-jamming game is Q-learning due to its fast convergence and easy implementation. Its modifications, such as Policy Hill Climbing (PCH) [14] algorithm is also popular due to its adaptability to changes in the environment.

Some articles do not use the channel model; this simplification leads to simpler but less realistic models. In [34], the authors consider a multi-channel game in which the system re-

ceives a fixed reward for successful message transmission, and the jammer receives a reward if it blocks the channel through which the message is being transmitted. The performances of such algorithms as Q-learning [55], Minimax-Q learning [50], Nash-Q [39], and Friend-or-foe Q-learning [51] are presented and compared.

Most articles consider the case of one or more jammers playing against several network devices. Article [101] discusses an anti-jamming game in the presence of several jammers. In addition, the authors suppose that the defending devices alter the transmission channels according to a predetermined pseudo-random sequence unknown by jammers. They can also move in space, choosing between several locations in order to minimize the negative effect of the jammers. Thus, in this game, the devices choose not only current transmission power, but also decide which location to move to. The authors use the Deep Q-learning algorithm to find the optimal device strategy and also include a theorem that formulates algorithm convergence conditions.

A relatively small number of articles are devoted to the VANET anti-jamming game. Article [30] considers the case in which the jammer is a vehicle approaching a platoon. The purpose of the jammer is to block the transmission of messages between two consecutive vehicles. Since it is a one channel game, the vehicles and the jammer can only change the power of the transmission adapting to the current situation and maximizing special utility functions. To protect against attack, the vehicles use the Q-learning algorithm and its modification called Dyna-Q (introduced in [81]). The presented graphs show that Dyna-Q converges faster to the optimal vehicle strategy than Q-learning. In [9], the anti-jamming game is considered on an open source 2D simulator JiST/SWANS. Along with the standard jammed channel change strategy, another strategy is also considered. According to this, at the time of attack detection, the network stops sending messages in the presence of the jammer. These strategies are compared using different metrics, such as, Packet Send Ratio.

Stackelberg game is discussed in the following articles. In [109], the authors prove theorems describing the optimal device and jammer strategies, and also explicitly express Stackel-

berg equilibrium in terms of the network parameters. In [100], a more realistic version of the Stackelberg game is considered, in which the jammer is informed about the device state with an error. The authors deduce the optimal strategy for both players, as well as the value of the Stackelberg equilibrium.

The games considered before are single agent games, since in them, each device tries to maximize its utility function without collaborating with other devices in the network. The case of multi-agent game is also considered in literature, in which devices act in a coordinated manner to develop a common strategy. In [5, 105], an anti-jamming game is considered with several transmitting devices. A feature of this game is that the devices both cooperate and compete for available communication channels. Since there are several devices, the state of the system is a vector consisting of the states of all devices, which leads to a state number explosion with an increase in the device number. Thus, this approach has limited application. The authors use the classic Q-learning algorithm to find the optimal strategy for all devices in the network. Article [110] describes a Stackelberg game with several cooperating devices and one jammer. The authors propose a cooperative anti-jamming algorithm, which demonstrates its superiority in comparison with random anti-jamming and selfish anti-jamming algorithms.

In the articles [52, 53, 102], a method for improving the reliability of VANET based on the use of drones is proposed. At the time of the attack, the drone duplicates the messages sent by VANET and thus increases the signal-to-interference-plus-noise ratio (SINR) and reduces bit error rate (BER). This measure could drastically improve the quality of communications, especially in the case where the channel is strongly attacked. It is assumed that the drone with at each iteration decides whether it replays the data or not (states 1 and 0). The authors find the conditions necessary for Nash equilibrium to be reached for states 1 and 0. They use a modification of the Q-learning algorithm called Policy Hill Climbing [14] which allows for the faster finding of the optimal strategy in comparison with the classical Q-learning algorithm.

1.4   Machine learning

In Chapter 5, dedicated to the anti-jamming game, we use machine learning methods, so we decide to include a summary of progress in this area (Sections 1.4, 1.5) in the Introduction. Machine learning is usually understood as the ability of a system to learn from its own experience without being explicitly programmed. Machine learning algorithms are trained on an extensive amount of data. Due to the fact that computers have become powerful enough to train complex models, as well as a large amount of accumulated data, machine learning has rapidly evolved and been applied to many areas. For instance, machine learning is used in search engines (e.g. Google), text translation from one language to another, recommendation systems (for example, YouTube, and Amazon etc.), also in spam filters, etc.

The advantage of machine learning is that it allows for the creation of intelligent systems, that without human intervention, can make complex decisions that are not pre-programmed. The widespread use of machine learning in various fields is also associated with the fact that it outperforms humans in classification and prediction tasks, which has found applications in engineering [15, 24, 64, 74], medicine [27, 61], banking [11, 16], and other fields.

Machine learning originated in the 1950s. One of the first machine learning publications was [70], in which Perceptron is introduced. Perceptron is an early prototype of modern neural networks for binary classification. Neural networks in the modern sense of the word are introduced in the famous article [71] by D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Despite the fact that there are many algorithms in this area, neural networks have shown their superiority with regards to a large amount of training data. This quality allows them to gradually replace other algorithms, becoming the main approach for solving machine learning problems. However, neural networks are not the only effective machine learning method. In the case where data is low-dimensional and the training set is sufficiently small, other methods, for example, Support Vector Machines (SVMs) [23] have proven their effectiveness in classification and regression problems [68].

The following important methods of machine learning have been developed in past decades.

- **Convolutional neural networks (CNNs)** [49] are widely applied to machine vision, image classification [20] and natural language processing [21, 43, 48]. The difference between CNN and traditional neural networks is the presence of special convolutional layers, pooling layers, fully connected layers, and normalization layers that allow for the learning of special features of images making it possible to significantly improve classification results in comparison with traditional neural networks.

- **Recurrent neural networks (RNNs) and Long Short Term Memory networks (LSTM is a special kind of RNN)** [38] have sequential data as an input. They can be perceived as neural networks with memory. Their output depends not only on the input data, but also on a special hidden state that stores information about previous network states. LSTMs are introduced to address the vanishing gradient problem. The reason behind it is that in some cases, the learning gradient is too small to train the network efficiently, which leads to a long training time. They have found applications in speech recognition [72], text translation [7], and text sentiment analysis [33].

- **Generative Adversarial Networks (GANs)**. GANs are one of the most important achievements of the last decade. They were introduced by Ian Goodfellow [31]. This architecture consists of two networks: teacher (discriminative network), and student (generative network). The result of the simultaneous training of these networks is a model that has found applications in the generation of new images [66], up-scaling low-resolution 2D textures [18], and reconstruction of 3D models from images [97].

- **Support-vector machines** SVMs [23, 82] are suitable for classification and linear regression. The basis of this method is the separation of sets using hyperplane, the position of which is iteratively recalculated. This method does not use neural networks, and is effective in high dimensional data spaces.

- **Random forest** Random forest [36, 37] is an ensemble of several decision trees built on random samples and based on a random set of features. It is well suited for classification

and regression problems, and is especially effective when the size of the training set is small enough. Each tree makes a prediction that classifies the input, then the most popular decision is considered as the prediction of the algorithm.

- **Linear regression** Linear regression [13, 79] is a method for predicting a numerical value by approximating a training set by a hyperplane. Unfortunately, the less the relationship between input and expected output resembles linear, the less efficient this algorithm becomes. However, for sparse training data, this algorithm can give good results.

- **k-Nearest Neighbors algorithm** [4, 46] This algorithm is the simplest on this list and its essence is quite natural. In order to classify the data, the algorithm considers k nearest neighbors to the sample. The class that occurs most often among these k samples is the answer. Despite its simplicity, this algorithm can be effective in cases where the size of the training set is sufficiently small.

1.5   Reinforcement learning

Since this thesis uses methods of reinforcement learning, we decide to make a detailed overview of this area. Reinforcement learning is a branch of machine learning that aims to find optimal actions for a single agent/group of agents to maximize a cumulative reward. These agents are trained by playing multiple games and learning from good and bad experiences without being explicitly programmed. The main feature of reinforcement learning is that agents can be in an environment that is not fully explored, so mathematical modeling and exact programming methods become very difficult or impossible to implement.

The advantage of reinforcement learning algorithms compared to conventional algorithms is that they do not need painstaking programming for behavior in every possible situation, but find optimal behavior using trial and error. In the real world, the number of states of the environment can be quite large or even infinite (as, for example, in the case of self-driving cars) that programing all of them is not possible and machine learning is the only possible

approach to find the optimal behavior. Another advantage of reinforcement learning over conventional algorithms is its adaptability to new changes in the environment. Reinforcement learning algorithms can be retrained and can adapt to new changes while classic algorithms require additional programming.
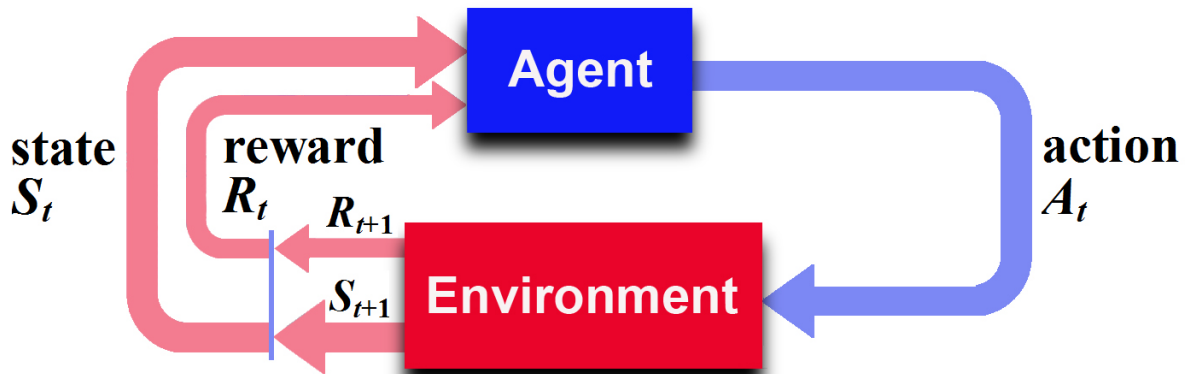
Figure 1.6: Reinforcement learning general scheme.

An important aspect of reinforcement learning is the balance between environmental exploration and reward accumulation [73, 83]. The naive approach here is greedy: each time the agent chooses the action with the highest reward, however, this approach can lead to a situation where the environment remains insufficiently explored and, therefore, the best strategies could not be discovered. Therefore, sufficient environmental exploration is an essential part of every reinforcement learning algorithm.

Let us formulate the problem of reinforcement learning using mathematical language. Agent strategy is denoted by π and referred to as a policy. Suppose that the agent finishes the game and after consecutive moves receives rewards $r_1, r_2, \ldots$ following the policy π. Then our goal is to maximize the following quantity:

$$\mathbf{E}\Big\{ \sum_k \gamma^k r_k \Big\}, \tag{1.1}$$

where $\mathbf{E}$ is a mathematical expectation and γ is called a discount-rate. Maximizing value (1.1) could be interpreted as maximizing the average cumulative reward obtained by following the

probabilistic policy π. The discount-rate γ is assumed to belong to the interval $[0, 1]$. It is included in (1.1), because we want to prioritize the rewards we get in the first steps over the rewards the agent gets closer to the end of the game. Most often, the parameter γ is selected from the range $[0.9, 0.999]$.

In 2013 an important article [57] was published, which laid the foundation for rapid progress in this area. The article proposes a method that allows for the use of neural networks instead of matrices in the classical Q-learning algorithm. A feature of Q-learning is that it is a general algorithm that can work for any mathematical model. In other words, receiving information about the current state of the system and action rewards, Q-learning can successfully find the optimal strategy for agent behavior in this environment. This strategy is optimal in the sense of maximization of (1.1). Initially, a Q-learning algorithm is proposed in the matrix case, where the value of $Q(s, a)$ determines the usefulness of the action of a from the state s. All that the algorithm should do is iteratively recalculate the Q matrix, taking into account the current awards. However, the number of system states can be very large, or even infinite, which makes the matrix approach impossible to implement. The article [57] proposes an approach based on neural networks, which are essentially trained to memorize the values of the Q matrix. Prior to this article, experts considered that it is impossible to use neural networks for reinforcement learning purposes. However, the crucial element proposed in the article called experience replay makes it possible. Experience replay is a periodical re-training of the network on a randomly selected set of past experiences.

Progress in Deep Q-learning has been continued by the following articles. One of the popular modifications of Q-learning is Double Q-learning introduced in [86]. In noisy environments, Q-learning may incorrectly evaluate the real value of the Q function, so an old copy of the network may be useful to address this issue. Thus, instead of using one neural network, two networks are used: a current and an old copy of the network. One network is used for value evaluation and another for the next action selection. Article [93] introduces another Q-learning modification called Dueling Q-learning. The essence of this modification is that another el-

ement called advantage is included in the architecture of the neural network measuring how good the reward for the action is in comparison with an average reward. The experiments have shown that Dueling Q-learning leads to a better policy evaluation and faster training.

Another family of reinforcement learning methods is called policy gradient methods. The difference between these methods compared to Q-learning is that the neural network predicts the probabilities of taking actions from a given state, and not their value. Usually, policy gradient methods have a slower convergence than Q-learning, but they are able to find more optimal strategies in complex games that may be missed by Q-learning due to local incorrect estimates of Q-matrix. In the classic Monte-Carlo policy gradient algorithm [40], action sequences are generated until the game is finished. After that, taking into account the received awards, the probabilities of the performed actions are reevaluated and this process is repeated many times. The disadvantage of the classical approach is that the new policy can differ significantly from the old one, especially in the initial stages of training, when the algorithm is just starting to explore the environment. To address this issue the Proximal Policy Optimization (PPO) algorithm is introduced [78]. A feature of this algorithm is that the objective function is changed and the ratio of the new probability and the old probability of action is included in it. If this ratio exceeds a certain threshold, it is truncated which allows for a stabilization of the training process. Another popular algorithm from this family is Trust Region Policy Optimization (TRPO) [77]. It also utilizes the idea of using the ratio of the new and old policy probabilities, but instead of clipping the reward function, it has trust region constraint which enforces the distance between the old and new policies. This algorithm shows approximately the same training results as PPO, but is more difficult to implement. The ideas of PPO and TRPO are further developed and refined in the following algorithms: Actor-Critic with Experience Replay (ACER) [92], Actor-Critic using Kronecker-Factored Trust Region (ACTKR) [99], Soft Actor-Critic (SAC) [35], etc.

1.6   Thesis Contributions

In Chapter 2, we consider a vehicle network on a single road and study its statistical properties. In this case, we derive such fundamental properties of the network as the distribution of the number of clusters, the size of randomly chosen cluster, and the largest cluster size, as well as the distribution of the number of disconnected vehicles. In previous articles, some of these characteristics are also investigated, however, only partial results are obtained in terms of average values, while the full distributions remain unknown. In Chapter 2, we conduct a versatile study and obtain analytical results, not only in terms of mean values and variances, but also in terms of complete distribution. We use the Rayleigh fading model to model the communication channel along with the classic signal propagation model. Simulations are carried out in the cases of urban and rural traffic with a different number of cars and vehicle density distributions. Obtained theoretical results coincide with the simulation results proving their correctness.

Chapter 3 is devoted to the investigation of the evolution of the vehicle network on a highway. Assuming that the state of the channel between two consecutive vehicles is determined by the Markov model with two states, we derive probability distributions of such fundamental evolution characteristics as the duration of the connection between consecutive vehicles, and the cluster lifetime duration. We also derive a formula expressing the probability that a cluster exists between two fixed time moments. Finally, in this chapter, we introduce the concept of ω-stability. This means that the vehicles may not be connected at certain time intervals, but they must at least be connected at each time interval $\omega \Delta t$ (Δt is a timestep between consecutive connections. See more details in Chapter 3). This communication performance is close to the actual operating conditions under which the connection between the cars can be interrupted, but we want to guarantee that it is reestablished again with a certain regularity. In Chapter 3 an algorithm is proposed for calculating the probability of a ω-stable connection between fixed periods of time. All the obtained results are confirmed by simulations done for realistic values of the network parameters.

In Chapter 4, we study the statistical properties of a network in the case in which the vehi-

cles are located on a 2D map with an almost arbitrary topology. To the best of our knowledge, this is the most general map topology considered in the literature, since the only considered scenarios are highways and intersections. In this case, obtaining accurate statistical results is difficult due to a large number of unknowns, such as car routing and traffic light states. Due to the large number of variables, this problem cannot be solved exactly, however, it is possible to obtain approximate distributions of the network properties. We derive the approximate distributions of such fundamental characteristics of the network as the distribution of the number of clusters, and the distribution of the average cluster size. The obtained results quite accurately approximate the simulation results for cases of urban and rural traffic.

Chapter 5 discusses the VANET anti-jamming game. We consider the following game formulation: we assume that the two vehicles are pursued by the jammer in order to disrupt their communication. We examine both single-channel and multi-channel cases. At each step, the vehicle decides on which channel it communicates and determines signal transmission power. The jammer at every step decides how to distribute its power along the channels in order to inflict maximum harm. The main results associated with finding the optimal vehicle strategy include theorems that explicitly contain formulas for calculating the Nash equilibrium. Nash equilibrium can be understood as the optimal player strategies, in this case, the players are the vehicle and the jammer. However, system parameter values such as the distance to the jammer and channel model are generally unknown. Therefore, we use machine learning algorithms to find the optimal vehicle strategy. We use such modern algorithms as Deep Q-learning, Dueling Q-learning, Double Dueling Q-learning and compare their performance with the popular in the literature modification of Q-learning called Policy Hill Climbing. The essence of these algorithms is that at each step, the vehicle receives information about the SINR of the channel; based on this information, the algorithm determines the power level of the next transmission. The simulation results show that the vehicles and the jammer find the optimal strategies derived theoretically in the cases of a constant distance between the vehicles and the distance that varies according to the Intelligent driver model.

It is worth noting that in the dissertation we use ideal models, however, the probability distributions of the intervehicle distance and the signal propagation in them can be replaced by real practical distributions. This could significantly improve prediction power of the models.

The dissertation results are published in journals and conference abstracts.

Chapter 2

Connectivity statistics of vehicle network

2.1 Introduction

This chapter focuses on the statistical properties of VANET in the case in which the vehicles move along the highway. Relatively many articles are devoted to this topic. Since more complex cases, such as a two-dimensional map are very difficult, or even impossible to analyze, the authors usually limit themselves to the case of one road. The complexity of this problem lies in the fact that cars are distributed on the road randomly, and the model of the communication channel is also probabilistic. Thus, in order to derive statistical distributions of such important network characteristics as the number of clusters, cluster size, etc., these two probabilistic models must be combined.

Let us discuss the previous results in more detail. We turn first to articles in which there is no channel model and it is assumed that vehicles within a certain range can always establish a connection, otherwise they are disconnected. In [87], assuming the known distribution of intervehicle distance, the authors derive how far the signal from one vehicle on the network can propagate. It is presumed that the signal is transmitted to the next vehicle in front as long as it is possible. The authors also obtain the explicit formula for the average value of the number of vehicles that are involved in this signal propagation. The study of such a property

as network full connectivity is popular in literature. This notion means that any two network vehicles can communicate with each other, possibly through other network vehicles (multi-hop propagation). One of the most profound articles in this area is [47], in which the authors study the statistical properties of a car network driving between two consecutive RSUs. The authors derive the cluster size distribution under the assumption of a known network vehicle number. The authors also obtain the probability that the network is fully connected and derive the average size of a fully connected network (they name it 'critical network size').

The following articles discuss the probabilistic channel model. Articles [10, 17] consider the Rayleigh, Rician, and Weibull fading channel models. The probabilities of successful communication between pairs of sequential vehicles, as well as the probability of full network connectivity are derived assuming known intervehicle distance distribution. In [88], the authors suggest that vehicles in the cluster do not communicate with each other, but, instead, all vehicles communicate with the main car. Under the presumption of the Rayleigh fading channel model, the Markov model of packet transmission, and the PHY decoding failure model, the average packet loss probability is derived. In [45, 104], the evolution of the network over time is considered. In [45], a case in which a signal is transmitted to vehicles moving in the opposite direction and then sent back to the initial side is investigated. The probability of such a successful two-hop connection is derived under the assumption that the vehicles move at a constant speed. Article [104] focuses on the study of link duration in the specific case, in which the speed of the vehicles grows linearly, reaches a limit, and then remains constant.

In this chapter, we consider the case of a vehicle network on a highway. Our goal is to give an extended description of the network, not only in terms of average values and variance, as is done in the previous articles (such as [47]), but to find the whole distribution of the number of clusters, size of randomly chosen cluster, and the number of disconnected vehicles (here we call them *idle* vehicles). We use a more realistic model of the communication channel (Rayleigh fading channel) than in [47, 87]. We advance further than the authors of [10, 17, 88] and find the distribution of the number of clusters and size of randomly chosen cluster. The

difference between this chapter and [45, 104] is that we do not consider the evolution of the network over time, but concentrate in detail on more nuanced connectivity characteristics at a fixed moment in time. Moreover, to the best of our knowledge, we are the first to investigate the distribution of the largest cluster size. We assume that every car communicates only with the nearest car in front and behind it. This assumption is made in the previous statistical research in this area as well, in order to make the model simple enough to analyze. Under the presumption of the Rayleigh fading model and known density function of intervehicle distance, we express the above mentioned distributions in terms of known parameters of the network. Our model can be applied to other scenarios if the probability of connection between every pair of consecutive vehicles has the same value p. We derive several interesting results describing network connectivity, namely, that the average size of randomly chosen cluster is a constant, approximately equaling $1/(1-p)$; the average number of clusters is proportional to the number of vehicles n equaling $1 + (n-1)(1-p)$; and in the average case there is a constant fraction $\approx 1/(1-p)^2$ of the *idle* vehicles. The average largest cluster of the network, however, is not a constant and grows as $\log_{1/p} n$. We confirm these and other theoretical results through simulations carried out in the cases of different car densities and number of cars. The averages and variances of the studied properties are summarized in the following tables (where $C_E = 0.577\ldots$ is Euler's constant):

Property	Average
Number of clusters	$1 + (n-1)(1-p)$
Size of clusters	$(1-p^n)/(1-p)$
Size of the largest cluster	$\approx \log_{1/p}\{(n-1)(1-p)\} + C_E/\ln(1/p) + 0.5$
Number of idle cars	$2(1-p) + (n-2)(1-p)^2$

Property	Variance
Number of clusters	$(n-1)p(1-p)$
Size of clusters	$(2p^{n+1}n - 2p^n n - p^{2n} - p^{n+1} + p^n + p)/(1-p)^2$
Size of the largest cluster	$\approx \pi^2/\ln^2(1/p) + \frac{1}{12}$
Number of idle cars	$-3np^4 + 10np^3 - 11np^2 + 4np + 8p^4 - 22p^3 + 18p^2 - 4p$

Table 2.1: Averages and variances obtained in Chapter 2

The studied statistical characteristics have several practical applications. Distribution of size of randomly chosen cluster and maximum cluster size distribution provide us with tools for network load prediction. Also, this statistical information can be used for security purposes. The size of randomly chosen cluster estimation, for example, allows for the prediction of the number of infected vehicles in the event of an attack on a cluster. Estimation of the number of disconnected vehicles makes it possible to evaluate the quality of connection, in order to make the percentage of disconnected vehicles acceptably low.

Chapter 2 is organized as follows. In Section 2.2 we describe the connectivity model and derive a formula for the probability of connection between two consecutive cars. Section 2.3 is devoted to describing the main results of this chapter. In 2.4 we derive all the probabilistic distributions mentioned above. Finally, in Section 2.5, the simulation results are presented and compared to the calculations done by the formulas from Section 2.4. Part of the content related to this chapter is located in Appendix A. This part is devoted to a quick introduction to the theory of generating functions needed in Section 2.4. The results of this chapter are published in [29].

2.2   Network model

We consider the network of randomly distributed cars on a highway and suppose that every vehicle of the network can establish a connection only with the closest front and back neighbours. We denote by d the distance between two consecutive vehicles. We assume that the distance between vehicles has random distribution with density function $f_d(x)$. Typically it is assumed that distance between cars has exponential distribution, normal distribution, gamma distribution or log-normal distribution. These distributions correspond to different flow conditions. For instance, exponential distribution corresponds to low traffic flow conditions, while high traffic flow conditions are described by normal distribution.

Let us suppose that there are n cars on a road and distribution of distance between cars has

a mean value M, therefore, the average distance between the first and the last car is $(n-1)M$, consequently, despite the fact that the road is long, it is improbable that the distance between first and last vehicle is very large.

We consider the same channel model as in [17] with the same parameters. We assume Rayleigh fading channel and that the average channel Signal-to-noise ratio (SNR) between two consecutive vehicles $\bar{\gamma}$ is calculated using the following formula:

$$\bar{\gamma} = \frac{P_{tx}K}{d^\alpha W}, \tag{2.1}$$

where P_{tx} is the transmit power, α is the path loss exponent and K is a constant associated with the path loss model. The parameter K is given by the formula

$$K = \frac{G_T G_R C^2}{(4\pi f_c)^2}, \tag{2.2}$$

where G_T and G_R are the transmit and receive antenna gains, C is the speed of light and f_c is the carrier frequency. We assume that the antennas are omni directional

$$G_T = G_R = 1 \tag{2.3}$$

and $f_c = 5.9 GHz$. The thermal noise power can be determined by the formula

$$W = kT_0 B, \tag{2.4}$$

where $k = 1.38 \times 10^{-23} J/K$ is the Boltzmann constant, T_0 ($T_0 = 300°K$) is the room temperature, and B ($B = 10$ MHz) is the transmission bandwidth.

In a real scenario, clusters are not necessarily formed by consecutive cars, however, this case is difficult for statistical analysis and is not considered in the literature. Therefore, in the dissertation we assume the following definition of a cluster.

Definition. A cluster is a group of sequential vehicles such that each previous vehicle communicates with the next. It is also assumed that this group cannot be expanded, therefore, communications with the vehicles before and after the cluster are not established.

The SNR density function $f_\gamma(x)$ of the Rayleigh channel is given by the formula

$$f_\gamma(x) = \frac{1}{\overline{\gamma}} e^{-x/\overline{\gamma}}. \tag{2.5}$$

We consider the Rayleigh channel model because it is traditionally used to model non-line-of-sight signal propagation. However, the results of this chapter can be generalized to other channel models by changing the density function (2.5). For example, for modeling line-of-sight signal propagation the Rice channel model is traditionally applied. It is assumed that there is a connection between consecutive cars if γ is greater than the threshold Ψ. Therefore,

$$P(\gamma > \Psi) = \int_\Psi^\infty f_\gamma(x)dx = e^{-\Psi/\overline{\gamma}}. \tag{2.6}$$

Let us denote the probability that two consecutive vehicles can establish a connection by p. By using (2.1), (2.6) and following the logic of [47] we can derive formula expressing value of the probability p through intervehicle density function $f_d(x)$:

$$p = \int_0^\infty P(\gamma(x) > \Psi)f_d(x)dx = \int_0^\infty exp\left(-\frac{\Psi x^\alpha W}{P_{tx}K}\right)f_d(x)dx. \tag{2.7}$$

Although we do not explicitly use such a characteristic as the minimum distance between vehicles in our model, it can be incorporated into the function $f_d(x)$ as follows. Let the minimum distance between cars be *MinDist*, then $f_d(x) = 0$ for $x < MinDist$.

The value of the parameter p lies within the interval $(0, 1)$ and plays a prominent role in the further investigations. It determines the quality of communication. A larger value of parameter p leads to a better quality of connection and consequently a bigger size of clusters.

2.3 Main results

Let us denote by *ClustNum* the number of clusters in the network. We derive the following formula for the probability $P(ClustNum = r)$ that the network has exactly r clusters

$$P(ClustNum = r) = \binom{n-1}{r-1} p^{n-r} (1-p)^{r-1}, \tag{2.8}$$

where parameter p is determined in (2.7).

The average cluster number $\mathbf{E}[ClustNum]$ and its variance $\mathbf{Var}[ClustNum]$ can be calculated using the following formulas:

$$\mathbf{E}[ClustNum] = 1 + (n-1)(1-p), \tag{2.9}$$

$$\mathbf{Var}[ClustNum] = (n-1)p(1-p). \tag{2.10}$$

Remark. We see from (2.9) that the average number of clusters in the system grows linearly with the number of vehicles with a growth factor $1 - p$.

We denote by $P(ClustSize = r)$ the probability that the size of randomly chosen cluster equals r. By size of a randomly chosen cluster we mean the following. Assuming the network has k clusters, we choose one of them with a probability of $1/k$. Then by size of randomly chosen cluster we mean the size of the cluster that is selected. The probability $P(ClustSize = r)$ is given by the following formula:

$$P(ClustSize = r) = \begin{cases} p^{r-1}(1-p), & \text{if } r < n, \\ p^{n-1}, & \text{if } r = n. \end{cases} \tag{2.11}$$

The average value and variance size of randomly chosen cluster are given by the formulas

$$\mathbf{E}[ClustSize] = \frac{1 - p^n}{1 - p}, \tag{2.12}$$

$$\mathbf{Var}[ClustSize] = \frac{2p^{n+1}n - 2p^n n - p^{2n} - p^{n+1} + p^n + p}{(1-p)^2}. \tag{2.13}$$

Remark. It follows from the formulas (2.12) and (2.13) that the average size of randomly chosen cluster approximately equals $1/(1-p)$ (constant!) and its variance approximately equals $p/(1-p)^2$, since $p^n \to 0, n \to \infty$.

In order to formulate the results about the distribution of the largest cluster we need the following function:

$$g(k) = \sum_{m=1}^{\lfloor n/(k+1)\rfloor} (-1)^m p^{mk}(1-p)^{m-1}\binom{n-1-mk}{m-1} +$$

$$+ \sum_{m=0}^{\lfloor (n-1)/(k+1)\rfloor} (-1)^m p^{mk}(1-p)^m\binom{n-1-mk}{m}, \tag{2.14}$$

where $\lfloor x \rfloor$ means rounding down. We denote the probability that the number of vehicles in the largest cluster equals r by $P(LargestClust = r)$. This probability is given by the following formula:

$$P(LargestClust = r) = \begin{cases} p^{n-1}, & \text{if } r = n, \\ g(r) - g(r-1), & \text{if } 1 < r < n, \\ (1-p)^{n-1}, & \text{if } r = 1. \end{cases} \tag{2.15}$$

However, the formula (2.15) is inconvenient for analysis, so we derive the following approximate formulas for average size of the largest cluster and its variance, which are simple and can be used for sufficiently large values of n:

$$\mathbf{E}[LargestClust] \approx \log_{1/p}\{(n-1)(1-p)\} + C_E/\ln(1/p) + 0.5, \tag{2.16}$$

$$\mathbf{Var}[LargestClust] \approx \pi^2/\ln^2(1/p) + \frac{1}{12}, \tag{2.17}$$

where $C_E = 0.577\ldots$ is Euler's constant.

Remark. From the previous remark we know that the average size of randomly chosen cluster is $\approx 1/(1-p)$, however, from (2.16) we conclude that the size of the largest cluster

grows logarithmically with the number of vehicles in the network.

We call a car *idle* if it cannot communicate with other vehicles in the network. Using the theory of generating functions, we derive the following formula for calculating the probability $P(IdleCars = r)$ that the number of *idle* vehicles in the network equals r:

$$P(IdleCars = r) = \frac{p^{n+1}}{(1-p)^2(n-r)!} \frac{d^{n-r}}{dx^{n-r}} \left\{ \frac{(1-x)^{r+1}}{((1-x)\frac{p}{1-p} - x^2)^{r+1}} \right\} \Big|_{x=0}, \qquad (2.18)$$

where $k!$ is a factorial and $\frac{d^k}{dx^k}f(x)\Big|_{x=0}$ is k-th derivative of function $f(x)$ calculated at the point $x = 0$. The formula (2.18) is quite complex, so we derived the following formulas for the average number of *idle* cars and their variance:

$$\mathbf{E}[IdleCars] = 2(1-p) + (n-2)(1-p)^2, \qquad (2.19)$$

$$\mathbf{Var}[IdleCars] = -3np^4 + 10np^3 - 11np^2 + 4np + 8p^4 - 22p^3 + 18p^2 - 4p. \qquad (2.20)$$

Remark. Formula (2.19) leads us to an important conclusion that in average there is a constant fraction $\approx (1-p)^2$ of *idle* vehicles in the network.

2.4 Mathematical derivations

2.4.1 Distribution of number of clusters

The main achievement of this chapter is that we drive not only the average values of various connectivity characteristics, as in the previous articles (see the Introduction), but also their probability distributions and variances. In this section we investigate such a significant characteristic of the network as distribution of number of clusters. In this section we establish the following formula for the probability of number of clusters *ClustNum* equaling r:

$$P(ClustNum = r) = \binom{n-1}{r-1} p^{n-r}(1-p)^{r-1}, \qquad (2.21)$$

where parameter p is determined in (2.7). In other words, number of clusters has binomial distribution.

To prove formula (2.21) we need to better understand the structure of the network. The idea behind the derivation is to consider connections between vehicles as Bernoulli trials. There are n vehicles in the network establishing no more than $n - 1$ connections between each other, thus, we can think about them as $n - 1$ independent trials, in which each of them is successful if consecutive vehicles can connect, and unsuccessful otherwise. Clusters are assumed to be formed by a series of several consecutive vehicles, therefore, if there are r clusters in the network, then exactly $r - 1$ vehicles cannot establish a connection. From the above we could conclude that the probability $P(ClustNum = r)$ equals the probability that in $n - 1$ trials exactly $r - 1$ are unsuccessful with the probability of success p. This probability is given by well-known formula for binomial distribution, which in our case is exactly formula (2.21).

By using known formulas for mean and variance of binomial distribution one can derive that the average number of clusters and their variance are given by the formulas

$$\mathbf{E}[ClustNum] = 1 + (n - 1)(1 - p), \tag{2.22}$$

$$\mathbf{Var}[ClustNum] = (n - 1)p(1 - p). \tag{2.23}$$

2.4.2   Distribution of size of randomly chosen cluster

In this subsection we concentrate on finding probability $P(ClustSize = r)$ that an size of randomly chosen cluster in the network equals r. We prove the following formula:

$$P(ClustSize = r) = \begin{cases} p^{r-1}(1 - p), & \text{if } r < n, \\ p^{n-1}, & \text{if } r = n. \end{cases} \tag{2.24}$$

Unfortunately, we do not know a simple proof. So, we use mathematical apparatus of generating functions, which allows for reformulating the problem in terms of series and use different

analytical methods to operate with them.

We denote number of clusters of the vehicle network N by $clust(N)$ and number of clusters having size r by $count(r; N)$. Let us establish the formula

$$P(ClustSize = r) = \sum_{k=1}^{n} \sum_{N:clust(N)=k} \frac{count(r; N)}{k} P(N), \tag{2.25}$$

where $P(N)$ is the probability of the network N and the second summation is carried out over all networks N with the restriction $clust(N) = k$. The network could have from 1 to n clusters, so k varies from 1 to n, and for each value of the parameter k, we consider all networks with k clusters. Finally, for each such case, we sum up the probabilities that a randomly selected cluster has size r. Since the network N contains $count(r; N)$ clusters of size r, then the probability of choosing a cluster of size r among k clusters is given as $\frac{count(r;N)}{k}$. We should multiply the last probability by $P(N)$ in order to calculate the probability that the cluster of size k is chosen in the network N.

We denote networks having k clusters, under condition that s of them have size r by $N_{s,k}$, therefore, $count(r; N_{s,k}) = s$. Since the network has k clusters, s varies from 1 to k. We rewrite the summation over all networks N as the summation over the networks $N_{s,k}$ as follows:

$$P(ClustSize = r) = \sum_{k=1}^{n} \sum_{s=1}^{k} \sum_{N_{s,k}} \frac{count(r; N_{s,k})}{k} P(N_{s,k}) = \sum_{k=1}^{n} \sum_{s=1}^{k} \sum_{N_{s,k}} \frac{s}{k} P(N_{s,k}). \tag{2.26}$$

Repeating the steps of derivation of (2.21) we conclude that

$$P(N_{s,k}) = (1 - p)^{k-1} p^{n-k}. \tag{2.27}$$

From (2.26) and (2.27) we derive that

$$P(ClustSize = r) = \sum_{k=1}^{n} \sum_{s=1}^{k} \sum_{N_{s,k}} \frac{s}{k} (1 - p)^{k-1} p^{n-k} = \sum_{k=1}^{n} (1 - p)^{k-1} p^{n-k} \sum_{s=1}^{k} \frac{sNum(s, k)}{k}. \tag{2.28}$$

Let us establish the formula

$$Num(s, k) = coeff_{x^n}\binom{k}{s}x^{rs}(x + x^2 + x^3 + \ldots - x^r)^{k-s}, \tag{2.29}$$

where $\binom{k}{s}$ is a binomial coefficient and $coeff_{x^n}P(x)$ is a coefficient of the term x^n in polynomial $P(x)$. The important step is to understand that the value of $Num(s, k)$ equals the number of representations of the integer n as a sum of k summands under condition that exactly s of them equal r. This is true, because the clusters are formed by group of consecutive vehicles. Therefore, we can use formula (A.4) in Appendix A identical to the formula (2.29) for calculation of $Num(s, k)$. From (2.29) we obtain

$$\sum_{s=1}^{k}\frac{sNum(s, k)}{k} = \frac{1}{k}coeff_{x^n}\sum_{s=1}^{k}\binom{k}{s}sx^{rs}(x + x^2 + x^3 + \ldots - x^r)^{k-s} =$$

$$= \frac{1}{k}coeff_{x^n}\sum_{s=1}^{k}\binom{k}{s}sx^{rs}\left(\frac{x}{1-x} - x^r\right)^{k-s}. \tag{2.30}$$

One can prove that the identity

$$\sum_{s=1}^{k}\binom{k}{s}sx^{rs}y^{k-s} = k(x^r + y)^{k-1}x^r \tag{2.31}$$

is satisfied. We substitute $y = \frac{x}{1-x} - x^r$ into (2.31) and use (2.30) to derive the following:

$$\sum_{s=1}^{k}\frac{sNum(s, k)}{k} = coeff_{x^n}\left(x^r + \frac{x}{1-x} - x^r\right)^{k-1}x^r = coeff_{x^n}\left(\frac{x}{1-x}\right)^{k-1}x^r. \tag{2.32}$$

Combining equations (2.28) and (2.32) we obtain

$$P(ClustSize = r) = \sum_{k=1}^{n}coeff_{x^n}\left(\frac{x}{1-x}\right)^{k-1}x^r(1-p)^{k-1}p^{n-k} =$$

$$p^{n-1}\sum_{k=1}^{n}coeff_{x^n}\left(\frac{x}{1-x}\right)^{k-1}x^r\left(\frac{1-p}{p}\right)^{k-1} = p^{n-1}\sum_{k=1}^{\infty}coeff_{x^{n-r}}\left(\frac{x}{(1-x)}\frac{1-p}{p}\right)^{k-1}. \tag{2.33}$$

The last equality in (2.33) is true because of the identity

$$p^{n-1} \sum_{k=n+1}^{\infty} coeff_{x^{n-r}} \left(\frac{x}{(1-x)} \frac{1-p}{p} \right)^{k-1} = 0.$$

The last identity is valid, because the series $\sum_{k=n+1}^{\infty} \left(\frac{x}{(1-x)} \frac{1-p}{p} \right)^{k-1}$ has nonzero coefficients only of $x^i, i \geq n$. Therefore, the coefficient of x^{n-r} equals zero. Transferring $coeff_{x^{n-r}}$ before the summation sign we deduce

$$P(ClustSize = r) = p^{n-1} coeff_{x^{n-r}} \sum_{k=1}^{\infty} \left(\frac{x}{(1-x)} \frac{1-p}{p} \right)^{k-1}. \tag{2.34}$$

Applying the formula of the sum of geometric progression to (2.34) we derive

$$P(ClustSize = r) = p^{n-1} coeff_{x^{n-r}} \frac{1}{1 - \frac{x}{(1-x)} \frac{1-p}{p}} = p^{n-1} coeff_{x^{n-r}} \frac{1-x}{1 - x(1 + \frac{1-p}{p})}. \tag{2.35}$$

By expanding fraction $\frac{1-x}{1-x(1+\frac{1-p}{p})}$ into power series we obtain

$$P(ClustSize = r) = p^{n-1} coeff_{x^{n-r}} (1-x) \sum_{k=0}^{\infty} x^k p^{-k} =$$

$$= \begin{cases} p^{n-1} \left(p^{-n+r} - p^{-n+r+1} \right), & \text{if } r < n, \\ p^{n-1}, & \text{if } r = n \end{cases} = \begin{cases} p^{r-1}(1-p), & \text{if } r < n, \\ p^{n-1}, & \text{if } r = n. \end{cases} \tag{2.36}$$

Formula (2.24) is proven.

Now we are ready to calculate the average value and the variance of the distribution (2.24). Firstly, we find the first two moments by the formulas

$$M_1 = np^{n-1} + (1-p) \sum_{\alpha=1}^{n-1} \alpha p^{\alpha-1}, \tag{2.37}$$

$$M_2 = n^2 p^{n-1} + (1-p) \sum_{\alpha=1}^{n-1} \alpha^2 p^{\alpha-1}. \tag{2.38}$$

By using the identities

$$\sum_{\alpha=1}^{n-1} \alpha p^{\alpha-1} = \frac{1 + ((n-1)p - n)p^{n-1}}{(1-p)^2}, \tag{2.39}$$

$$\sum_{\alpha=1}^{n-1} \alpha^2 p^{\alpha-1} = \frac{1 + p - ((n-1)^2 p^2 + (-2n^2 + 2n + 1)p + n^2)p^{n-1}}{(1-p)^3} \tag{2.40}$$

we derive

$$M_1 = \frac{1 - p^n}{1 - p}, \tag{2.41}$$

$$M_2 = \frac{2p^{n+1}n - 2p^n n - p^{n+1} - p^n + p + 1}{(1-p)^2}. \tag{2.42}$$

From formulas (2.41) and (2.42) we finally obtain

$$\mathbf{E}[ClustSize] = M_1 = \frac{1 - p^n}{1 - p}, \tag{2.43}$$

$$\mathbf{Var}[ClustSize] = M_2 - M_1^2 = \frac{2p^{n+1}n - 2p^n n - p^{2n} - p^{n+1} + p^n + p}{(1-p)^2}. \tag{2.44}$$

2.4.3   Distribution of the largest cluster size

In the previous section we explored the question about distribution of cluster size. It is established that the average cluster size is approximately $1/(1 - p)$, therefore, being a constant. However, it is natural to investigate this problem further and understand how the largest cluster size deviates from the average.

We denote the probability that the largest cluster of the network is formed by exactly r vehicles by $P(LargestClust = r)$. Let us introduce the function $g(k)$ by the formula

$$g(k) = \sum_{m=1}^{\lfloor n/(k+1)\rfloor} (-1)^m p^{mk}(1-p)^{m-1}\binom{n-1-mk}{m-1} + \sum_{m=0}^{\lfloor (n-1)/(k+1)\rfloor} (-1)^m p^{mk}(1-p)^m\binom{n-1-mk}{m},$$

$$\tag{2.45}$$

where $\lfloor x \rfloor$ means rounding down. We establish the following formula:

$$P(LargestClust = r) = \begin{cases} p^{n-1}, & \text{if } r = n, \\ g(r) - g(r-1), & \text{if } 1 < r < n, \\ (1-p)^{n-1}, & \text{if } r = 1. \end{cases} \qquad (2.46)$$

The cases $r = n$ and $r = 1$ are trivial, because under these assumptions all vehicles are connected or disconnected, respectively. Thus, we need to prove the formula (2.46) only in the case $1 < r < n$. We denote the length of the longest run in the sequence of the $n - 1$ Bernoulli trials by L_{n-1}. The following equation is proven in [62]:

$$P(L_{n-1} \leq r - 1) = g(r), \qquad (2.47)$$

where $g(k)$ is determined in (2.45). The number of vehicles in the network is n, consequently, there are $n - 1$ connections between them and we can consider them as a sequence of Bernoulli trials as before. The probability $P(LargestClust = r)$ equals the probability that the network has the longest run equaling $r - 1$ (number of connections is less than the number of cars by 1). From the above we derive

$$P(LargestClust = r) = P(L_{n-1} = r - 1) =$$

$$P(L_{n-1} \leq r - 1) - P(L_{n-1} \leq r - 2) = g(r) - g(r-1). \quad (2.48)$$

Formula (2.46) is proven.

Despite the fact that the formula (2.46) can be used for the calculation of the probabilities $P(LargestClust = r)$, it is overly complicated to predict behaviour of the network with growing number of vehicles. Fortunately, the asymptotic of the longest successful run with growing n is already found (see, for instance, [32, 76]), in which the expected value and variance of the longest run are derived for sufficiently large values of n. Taking into account that there are

$n - 1$ connections in our network, the fact that the number of connections is less that number of vehicles in a cluster by 1, and the formulas from above mentioned articles we conclude that the expected value and variance of the largest cluster of our network are

$$\mathbf{E}[LargestClust] \approx \log_{1/p}\{(n - 1)(1 - p)\} + C_E/\ln(1/p) + 0.5, \tag{2.49}$$

$$\mathbf{Var}[LargestClust] \approx \pi^2/\ln^2(1/p) + \frac{1}{12}, \tag{2.50}$$

where $C_E = 0.577\ldots$ is Euler's constant.

2.4.4   Distribution of idle car number

We call car *idle* if it fails to establish a connection with any of its neighbours. Unfortunately, there is no simple formula for calculating the probability $P(IdleCars = r)$ that the network has exactly r *idle* vehicles. We prove by using method of generating functions that the probability $P(IdleCars = r)$ can be calculated by the formula

$$P(IdleCars = r) = \frac{p^{n+1}}{(1 - p)^2(n - r)!} \frac{d^{n-r}}{dx^{n-r}}\left\{\frac{(1 - x)^{r+1}}{((1 - x)\frac{p}{1-p} - x^2)^{r+1}}\right\}\Big|_{x=0}, \tag{2.51}$$

where $k!$ is a factorial and $\frac{d^k}{dx^k}f(x)\Big|_{x=0}$ is k-th derivative of function $f(x)$ calculated at the point $x = 0$.

We denote by *IdleNum*(k, r) number of graphs having k clusters and r *idle* cars. By considering Problem 4 from Appendix A and (2.27) we derive

$$IdleNum(k, r) = coeff_{x^n}\binom{k}{r}x^r(x^2 + x^3 + \ldots)^{k-r}, \tag{2.52}$$

$$P(\text{Graph has } k \text{ clusters}) = (1 - p)^{k-1}p^{n-k}. \tag{2.53}$$

Combining formulas (2.52) and (2.53), we get

$$P(IdleCars = r) = \sum_{k=r}^{n} IdleNum(k, r)P(\text{Graph has } k \text{ clusters}) =$$

$$coeff_{x^n} \sum_{k=r}^{n} \binom{k}{r} x^r (x^2 + x^3 + \ldots)^{k-r} (1-p)^{k-1} p^{n-k} =$$

$$coeff_{x^n} \sum_{k=r}^{\infty} \binom{k}{r} x^r (x^2 + x^3 + \ldots)^{k-r} (1-p)^{k-1} p^{n-k}. \quad (2.54)$$

The last equality in (2.54) is correct because

$$coeff_{x^n} \sum_{k=n+1}^{\infty} \binom{k}{r} x^r (x^2 + x^3 + \ldots)^{k-r} (1-p)^{k-1} p^{n-k} = 0.$$

Equation (2.54) could be rewritten as follows:

$$P(IdleCars = r) = \frac{p^n}{1-p} coeff_{x^{n-r}} \left(\frac{1-x}{x^2}\right)^r \sum_{k=r}^{\infty} \binom{k}{r} \left(\frac{x^2}{1-x} \frac{1-p}{p}\right)^k. \quad (2.55)$$

We use the following known fact about the Taylor series. The coefficient $coeff_{x^i} f(x)$ in the series $f(x) = \sum_{i=0}^{\infty} a_i x^i$ can be calculated by the formula

$$coeff_{x^i} f(x) = a_i = \frac{1}{i!} \frac{d^i f}{dx^i}\Big|_{x=0}. \quad (2.56)$$

If we substitute $y = \frac{1-x}{x^2} \frac{p}{1-p}$ into the identity

$$\sum_{k=r}^{\infty} \binom{k}{r} y^k = \frac{1}{y} \frac{1}{\left(\frac{1}{y} - 1\right)^{r+1}},$$

we obtain

$$\sum_{k=r}^{\infty} \binom{k}{r} \left(\frac{1-x}{x^2} \frac{p}{1-p}\right)^k = \frac{x^2}{1-x} \frac{1-p}{p} \frac{1}{\left(\frac{x^2}{1-x} \frac{1-p}{p} - 1\right)^{r+1}}. \quad (2.57)$$

By substituting (2.57) into (2.55), reducing fractions to a common denominator and using

(2.56), we derive

$$P(IdleCars = r) = \frac{p^n}{1-p} coeff_{x^{n-r}} \Big(\frac{1-x}{x^2}\Big)^r \frac{1-x}{x^2} \frac{p}{1-p} \frac{1}{\Big(\frac{1-x}{x^2}\frac{p}{1-p} - 1\Big)^{r+1}} =$$

$$\frac{p^{n+1}}{(1-p)^2} coeff_{x^{n-r}} \Big(\frac{1-x}{x^2}\Big)^{r+1} \frac{1}{\Big(\frac{1-x}{x^2}\frac{p}{1-p} - 1\Big)^{r+1}} =$$

$$\frac{p^{n+1}}{(1-p)^2} coeff_{x^{n-r}} \frac{(1-x)^{r+1}}{\Big((1-x)\frac{p}{1-p} - x^2\Big)^{r+1}} =$$

$$\frac{p^{n+1}}{(1-p)^2(n-r)!} \frac{d^{n-r}}{dx^{n-r}} \Big\{ \frac{(1-x)^{r+1}}{((1-x)\frac{p}{1-p} - x^2)^{r+1}} \Big\}\Big|_{x=0}.$$

Formula (2.51) is proved.

Despite the fact that the formula of distribution of *idle* vehicles is quite complicated, we can derive simple expression for the expected number of *idle* vehicles. Let us denote the random variable that equals 1 it k-th vehicle is idle and 0 otherwise by ξ_k. Let us prove that the expected value of ξ_k can be determined by this formula

$$\mathbf{E}[\xi_k] = \begin{cases} (1-p)^2, & \text{if } 1 < k < n, \\ 1-p, & \text{if } k = 1 \text{ or } k = n. \end{cases} \tag{2.58}$$

It is enough to prove the formula (2.58) in the case $1 < k < n$, since derivations in other cases are similar. If $1 < k < n$, then the vehicle has exactly two neighbours. Since the probability of disconnection equals $(1-p)$, the probability of car being *idle* equals $(1-p)^2$. Finally,

$$\mathbf{E}[\xi_k] = 1 \times (1-p)^2 + 0 \times (1 - (1-p)^2) = (1-p)^2. \tag{2.59}$$

Formula (2.58) is proven. Therefore, the average number of *idle* cars is

$$\mathbf{E}[IdleCars] = \mathbf{E}\Big[\sum_{k=1}^{n} \xi_k \Big] = \sum_{k=1}^{n} \mathbf{E}[\xi_k] = 2(1-p) + \sum_{k=2}^{n-1}(1-p)^2 = 2(1-p) + (n-2)(1-p)^2. \tag{2.60}$$

Let us calculate the variance of distribution of number of *idle* vehicles. It is given as follows:

$$\mathbf{Var}[IdleCars] = \mathbf{E}\Big[\Big(\sum_{k=1}^{n}\xi_k\Big)^2\Big] - \mathbf{E}[IdleCars]^2 = \sum_{k=1}^{n}\mathbf{E}[\xi_k^2] + 2\sum_{k=1}^{n-1}\sum_{j=k+1}^{n}\mathbf{E}[\xi_k\xi_j] - \mathbf{E}[IdleCars]^2.$$

(2.61)

If two vehicles are not neighbours then the variables ξ_k and ξ_j are independent and, therefore, $\mathbf{E}[\xi_k\xi_j] = \mathbf{E}[\xi_k]\mathbf{E}[\xi_j]$, otherwise $\mathbf{E}[\xi_k\xi_j]$ can be derived by analogy with the derivation of (2.58). Thus, the formula

$$\mathbf{E}[\xi_k\xi_j] = \begin{cases} \mathbf{E}[\xi_k]\mathbf{E}[\xi_j], & \text{if } j > k+1, \\ (1-p)^3, & \text{if } j = k+1 \text{ and } 1 < k < n-1, \\ (1-p)^2, & \text{if } j = k+1, k = 1 \text{ or } k = n-1. \end{cases}$$

(2.62)

is valid for $j > k$. Using (2.62), we derive

$$\sum_{k=1}^{n-1}\sum_{j=k+1}^{n}\mathbf{E}[\xi_k\xi_j] = \sum_{k=1}^{n-1}\sum_{j=k+2}^{n}\mathbf{E}[\xi_k]\mathbf{E}[\xi_j] + \sum_{k=2}^{n-2}\mathbf{E}[\xi_k\xi_{k+1}] + \mathbf{E}[\xi_1\xi_2] + \mathbf{E}[\xi_{n-1}\xi_n]$$

$$= \sum_{k=1}^{n-1}\sum_{j=k+2}^{n}\mathbf{E}[\xi_k]\mathbf{E}[\xi_j] + (n-3)(1-p)^3 + 2(1-p)^2. \quad (2.63)$$

Consequently, by using (2.61) and (2.63) and taking into account that $\xi_k^2 = \xi_k$ we obtain

$$\mathbf{Var}[IdleCars] = \sum_{k=1}^{n}\mathbf{E}[\xi_k^2] + 2\sum_{k=1}^{n-1}\sum_{j=k+1}^{n}\mathbf{E}[\xi_k\xi_j] - \Big(\mathbf{E}\Big[\sum_{k=1}^{n}\xi_k\Big]\Big)^2 =$$

$$\sum_{k=1}^{n}\mathbf{E}[\xi_k^2] + 2\sum_{k=1}^{n-1}\sum_{j=k+2}^{n}\mathbf{E}[\xi_k]\mathbf{E}[\xi_j]$$

$$+ 2(n-3)(1-p)^3 + 4(1-p)^2 - \sum_{k=1}^{n}\mathbf{E}[\xi_k]^2 - 2\sum_{k=1}^{n-1}\sum_{j=k+1}^{n}\mathbf{E}[\xi_k]\mathbf{E}[\xi_j] =$$

$$\sum_{k=1}^{n}\mathbf{E}[\xi_k] - \sum_{k=1}^{n}\mathbf{E}[\xi_k]^2 - 2\sum_{k=1}^{n-1}\mathbf{E}[\xi_k]\mathbf{E}[\xi_{k+1}] + 2(n-3)(1-p)^3 + 4(1-p)^2. \quad (2.64)$$

We derive the explicit formulas for every sum in (2.64). Equality (2.60) gives us

$$\sum_{k=1}^{n} \mathbf{E}[\xi_k] = 2(1-p) + (n-2)(1-p)^2. \tag{2.65}$$

By analogy with derivation of (2.60) we can deduce that

$$\sum_{k=1}^{n} \mathbf{E}[\xi_k]^2 = 2(1-p)^2 + (n-2)(1-p)^4. \tag{2.66}$$

The last sum in (2.64) can be derived using (2.58) as follows:

$$\sum_{k=1}^{n-1} \mathbf{E}[\xi_k]\mathbf{E}[\xi_{k+1}] = \sum_{k=2}^{n-2} \mathbf{E}[\xi_k]\mathbf{E}[\xi_{k+1}] + \mathbf{E}[\xi_1]\mathbf{E}[\xi_2]+$$

$$\mathbf{E}[\xi_{n-1}]\mathbf{E}[\xi_n] = (n-3)(1-p)^4 + 2(1-p)^3. \tag{2.67}$$

Combining formulas (2.64)–(2.67) we obtain

$$\mathbf{Var}[IdleCars] = 2(1-p)+(n-2)(1-p)^2-2(1-p)^2-(n-2)(1-p)^4-2(n-3)(1-p)^4-4(1-p)^3$$

$$+ 2(n-3)(1-p)^3 + 4(1-p)^2 = -3np^4 + 10np^3 - 11np^2 + 4np + 8p^4 - 22p^3 + 18p^2 - 4p. \tag{2.68}$$

2.5　Simulations

We use the model described by the equations (2.1)–(2.7) and make simulations for $n = 10$ and $n = 20$ vehicles on the road assuming exponential distribution of the intervehicle distance with probability density function $\rho e^{-\rho x}$, where ρ is a vehicle density. We consider two scenarios. In the first case, the density is low $\rho = 0.01$ (10 cars per kilometer) which is the case for rural traffic, in the second case $\rho = 0.05$ (50 cars per kilometer), which corresponds to urban traffic. We use the following values of the parameters taken from [17] (see their descriptions in Section 2.2):

$$G_T = G_R = 1, f_c = 5.9 \text{ GHz}, \alpha = 2.5,$$

$$T_0 = 300° \text{ K}, B = 10 \text{ MHz}, \Psi = 10 \text{ dB}.$$

We use the transmit power value $P_{tx} = 4$ dBm. This value corresponds to the message trans-mission over short distances up to 99 meters [69].

Graphs of all network properties are plotted in the cases (a) $n = 10$, $\rho = 0.01$, (b) $n = 10, \rho = 0.05$, (c) $n = 20, \rho = 0.01$, and (d) $n = 20, \rho = 0.05$. In the case $\rho = 0.01$, the probability of connection between two consecutive cars is $p = 0.5576$, in the case $\rho = 0.01$ the probability of connection is $p = 0.9525$ (it does not depend on the number of cars within the model). To verify the obtained formulas we generate sequence of vehicles randomly 100000 times and then calculate network connectivity distributions from these sample values. The graphics of the simulations are identical to the results obtained by the formulas, which confirms their correctness.

In Figure 2.1 the graphics of distribution (2.21) of number of clusters are presented as well as simulation results for scenarios (a), (b), (c), and (d). As one can see from Figure 2.1 in cases (a) and (c), the connection is less stable ($p = 0.5576$), therefore, the network on average has sufficiently large number of clusters 4.9812 and 9.4048, respectively. In cases (b) and (d), the connection occurs with a much higher probability $p = 0.9525$, therefore, in these cases the network tends to be fully connected, which is expressed in a high probability of having 1–3 clusters. In these cases average sizes of randomly chosen cluster are significantly lower: 1.4278 and 1.9031, respectively. The fact that the distributions in Figure 2.1 are close to normal is not surprising, since, as well known, the binomial distribution in the case $n \to \infty$ tends to normal.

In Figure 2.2 the graphs of the distribution of size of randomly chosen cluster are presented. The results are obtained by the formula (2.24) and compared to the simulation results. The same logic as in the previous paragraph leads us to the conclusion that the size of randomly chosen cluster in cases (a) and (c) should be smaller than in cases (b) and (d). This hypothesis is confirmed, because the average sizes of randomly chosen cluster in these cases are 2.2541

(a) $n = 10, \rho = 0.01$ (b) $n = 10, \rho = 0.05$

(c) $n = 20, \rho = 0.01$ (d) $n = 20, \rho = 0.05$
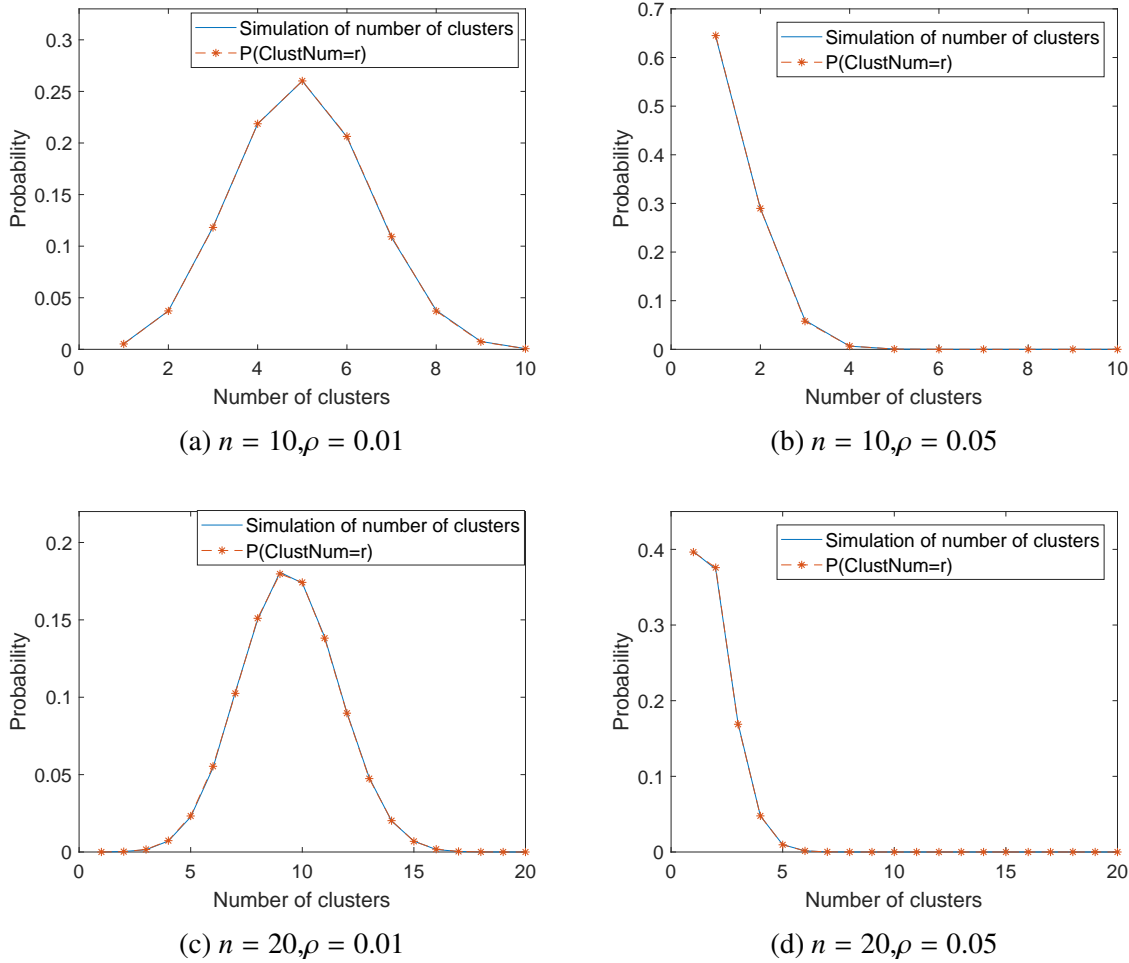
Figure 2.1: Probability distribution of number of clusters in the network for different n and ρ.

and 2.2606, respectively, versus 8.1109 and 13.0948 in cases (b) and (d). The decrease in graphs in cases (a) and (c) is explained by the fact that in these cases the connection is less stable, therefore, the probability of a cluster having a size i decreases with increasing of the parameter i. Graphs (b) and (d) have a pronounced peak, since in the case of stable connection, the network tends to be fully connected, therefore, a cluster having size n has a maximum probability.

The graphs of the largest cluster size distribution obtained by the formula (2.46) and simulation results are presented in Figure 2.3. By analogy with the explanation of the behavior of cluster size distribution graphs, the behavior of the maximum cluster size distribution can be explained. The average values in cases (a), (b), (c), and (d) equal 4.0425, 8.9048, 5.2253, and

(a) $n = 10, \rho = 0.01$ (b) $n = 10, \rho = 0.05$

(c) $n = 20, \rho = 0.01$ (d) $n = 20, \rho = 0.05$
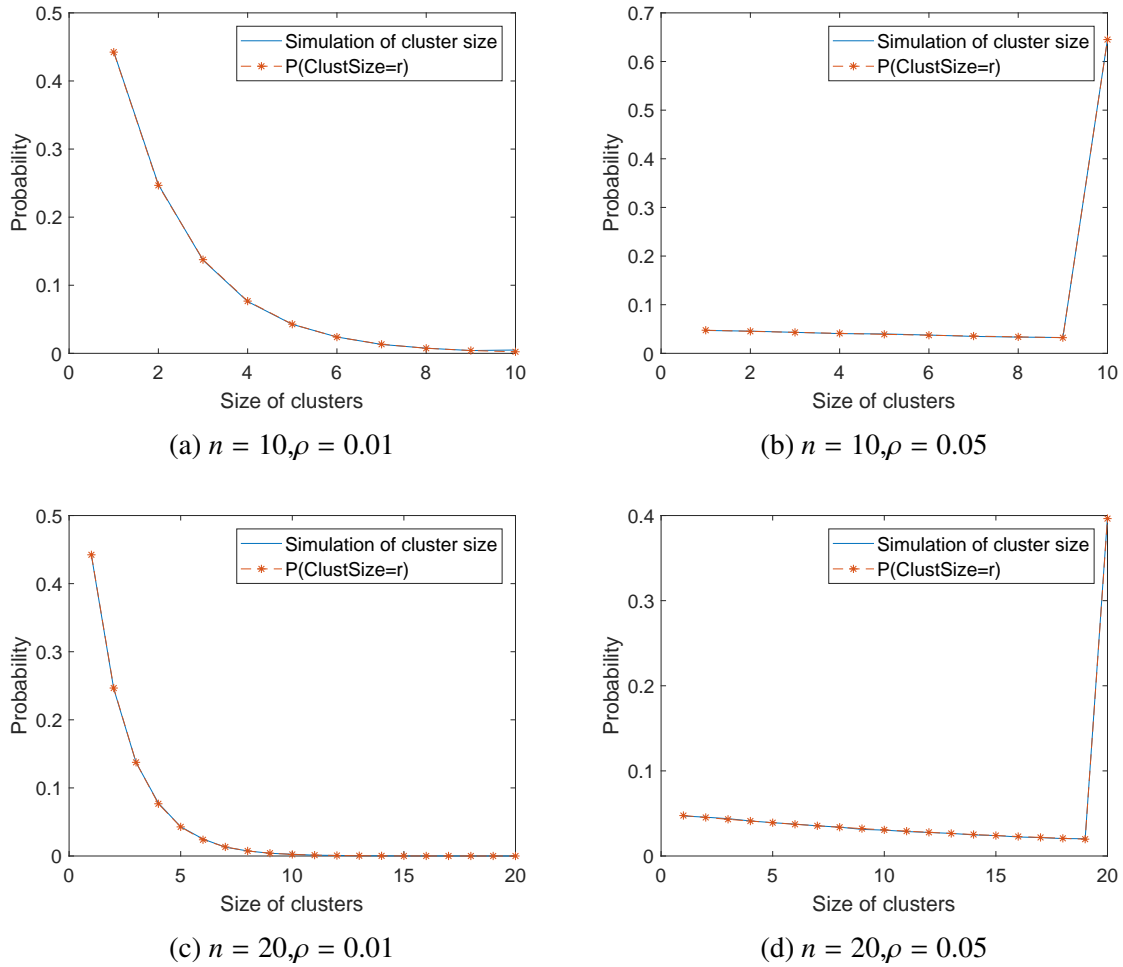
Figure 2.2: Probability distribution of size of randomly chosen cluster for different n and ρ.

16.0203, respectively.

Finally, the graphs of the distribution of the number of *idle* vehicles are depicted in Figure 2.4. The simulation results are compared with the probabilities calculated by the formula (2.51). One can explain the graphs of distribution of *idle* cars by the fact that in the case of a more stable connection (graphs (b) and (d)), the probability that *idle* cars are not present in the network is close to 1, and then quickly decreases with increasing their number. On graphs (a) and (c), on the contrary, the probability of the network having several *idle* vehicles is quite high due to the less stable connection. This can also be explained in terms of average numbers of *idle* vehicle in cases (a), (b), (c), and (d) equaling 2.4501, 0.1131, 4.4069, and 0.1357 respectively.

(a) $n = 10, \rho = 0.01$

(b) $n = 10, \rho = 0.05$

(c) $n = 20, \rho = 0.01$

(d) $n = 20, \rho = 0.05$
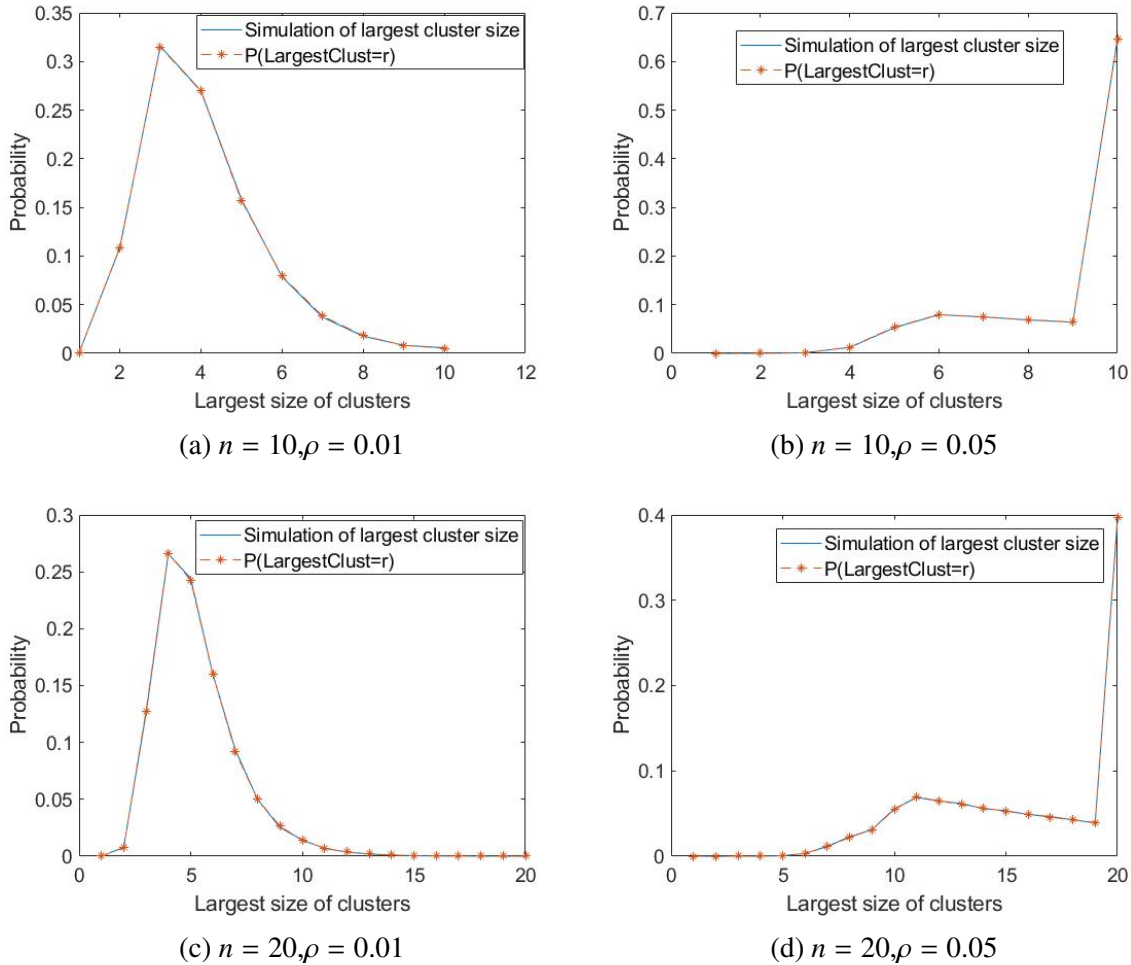
Figure 2.3: Probability distribution of size of the largest cluster for different n and ρ.

2.6   Chapter summary

In this chapter, we study the statistical properties of VANET in the case in which the vehicles are located on a highway. We assume the known probabilistic distribution of distance between vehicles and the Rayleigh channel model. Under these assumptions, we derive explicit formulas for the probability distribution of the number of clusters, cluster size, and the number of disconnected vehicles (here we call them *idle* cars). For the first time in the literature, we obtain formulas for the largest cluster size distribution. We perform simulations in the case of urban and rural traffic, and thus confirm the correctness of the obtained analytical results.
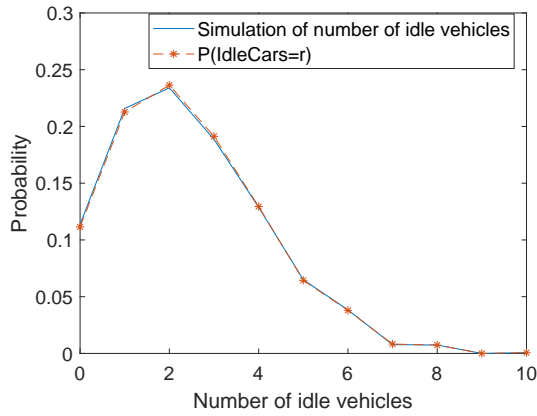
(a) $n = 10, \rho = 0.01$ (b) $n = 10, \rho = 0.05$
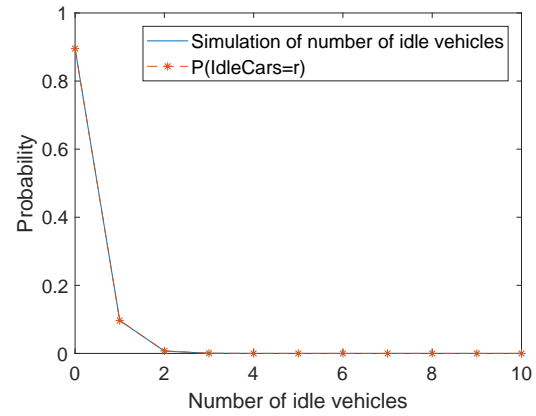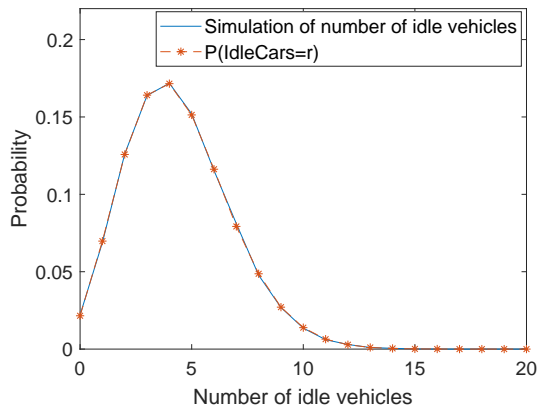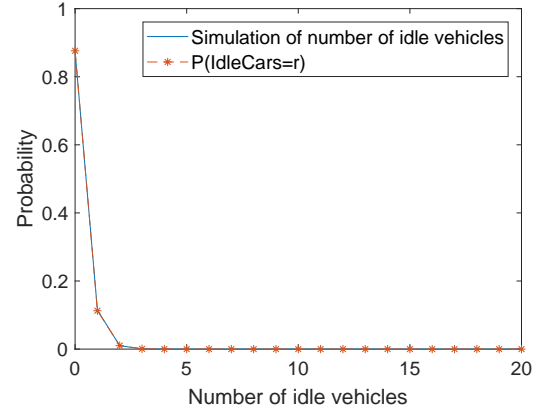
(c) $n = 20, \rho = 0.01$ (d) $n = 20, \rho = 0.05$

Figure 2.4: Probability distribution of the number of *idle* vehicles for different n and ρ.

Chapter 3

Evolution of Vehicle Network

3.1 Introduction

In Chapter 2, we consider the statistical properties of the network in the case in which the vehicles are located on a highway. However, these characteristics are limited by the current moment, while in practice it is also important to know how the behavior of the network changes over time. In the literature, this analysis is usually done in terms of link duration or probability of connection in the case of multi-hop signal propagation.

In [45], the authors study link duration in the case of two-way traffic. They suggest that the vehicle establishes communication with another vehicle traveling in the opposite direction, which then transmits a message to the recipient located in the same lane as the sender. The authors deduce the probability of such a two-hop message transmission under the assumption that the speeds of the vehicles are constant. In [91], the authors consider a more general case in which vehicle speeds are distributed according to a normal distribution and derive explicit formulas for link duration. In [56], in order to model signal propagation, the authors derive a formula for the probability of a connection between nearby cars. The peculiarity of this article is that the authors assume that the recipient of the message does not have to be on the same road, they take into account the geometric location of the sender, recipient and next

46

node when deriving the integral formula for the distribution density of the signal transmission time. In [104], the average link duration is calculated in the case of one-way traffic under the assumption that speed increases linearly until it reaches the speed limit and then remains constant.

In this chapter, we also derive the probability distribution of link duration, as well as its average value and variance. In order to make the model more realistic, we consider the Wang-Moayeri model of signal transmission. It is convenient for us to consider the state of the network with a small time step Δt. At each step, the probability of a successful connection is calculated by the probability of a connection in the previous step according to Markov chain model. The parameters of this Markov model are explicitly expressed through the macro parameters of the network in [65, 89]. Moreover, we study such advanced properties of the network as the lifetime of the cluster, as well as the probability of the existence of a cluster between two fixed time moments. We obtain results regarding the lifetime of the cluster, both in terms of distribution and in terms of average value and variance. To the best of our knowledge, these results are the first in the literature devoted to cluster connectivity evolution over time.

Also, we consider such a stability characteristic of the connection between two consecutive cars as ω-stable connection. This type of connection guarantees that the time between consecutive connections does not exceed $\omega \Delta t$ (Δt is a timestep). In other words, this weakened condition means that at some moments of time the vehicles may fail to establish a connection, but they reconnect at least once at each time interval that has length $\omega \Delta t$. This type of communication is closer to the actual operating conditions, where connection may disappear for short periods of time, but it is important to ensure that it is regularly reestablished. We derive recurrent equations for calculating the probabilities and verify the results using simulations.

Chapter 3 is organized as follows. Section 3.2 describes the parameters and the structure of a considered network model. Section 3.3 investigates the statistics of node-to-node connectivity between cars assuming a fading channel between the nodes. A brief summary of the results is given in 3.4, and the detailed mathematical derivations are presented in Section 3.5.

The derived expressions are verified through a number of simulations in Sections 3.6, 3.7. The results of this chapter are published in [28].

3.2   Network model

It is convenient to investigate the connectivity properties of the system not at every moment, but only on a discrete uniform time mesh. It means that the difference between two consecutive moments of time has a fixed duration Δt. Let t be the current moment of time. The Markov model determines the probability of an event that at the moment of time $t + \Delta t$, the consecutive vehicles could establish a connection based on the communication state at the time moment t.

We consider the following probabilistic model that describes the evolution of the network over time. It is assumed that there are two states of connection *Good* and *Bad*. In the *Good* state, neighboring cars can establish a connection, while the *Bad* state corresponds to the case where neighboring cars cannot connect with each other. We suppose that initially the system is in the equilibrium state; the initial probabilities of connection are calculated in Section 3.5. We consider the system evolution process with timestep Δt (introduced at the end of the previous section). Let q' be the probability that connected neighbouring cars cannot establish a connection at the next moment of time, in other words, q' is the probability that at the next moment of time the system goes from the *Good* state to the *Bad* state. By analogy, let q be the probability that at the next moment the system goes from the *Bad* state to the *Good* state. Therefore, connectivity between two consecutive cars can be described by two-state Markov Model depicted below (letters **G** and **B** denote the *Good* and the *Bad* states, respectively). The explicit values of the parameters q and q' are given in the next section.

3.3   Probabilistic connectivity model

Everywhere in the article it is assumed that all n vehicles move along one-way road with the same constant speed v. We also assume that the communication state of every pair of cars is
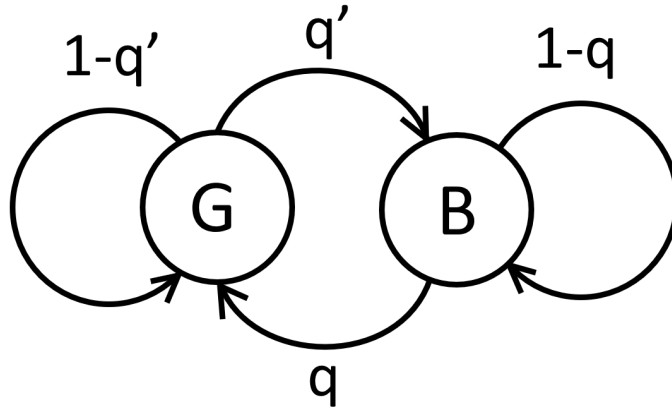
Figure 3.1: Markov diagram of the connectivity process with the *Good* and the *Bad* states.

independent of the communication states of other cars. We use the Wang-Moayeri channel model [65] and the same cluster definition as given in Section 2.2. We assume a Rayleigh fading channel between every pair of cars. Consequently, the amplitude of the received signal is exponentially distributed with pdf

$$p(x) = \frac{1}{\lambda} e^{-x/\lambda},$$ (3.1)

where λ represents the average SNR (Signal-to-noise ratio) over the fading channel. According to this model, we determine state of the system by comparison signal amplitude A and threshold $\overline{A}$. If $A \geq \overline{A}$ then we assume, that the system is in the *Good* state, otherwise it is in the *Bad* state. We denote by p_B the probability of the *Bad* case, therefore,

$$p_B = \int_0^{\overline{A}} p(x)dx = \int_0^{\overline{A}} \frac{1}{\lambda} e^{-x/\lambda}dx = 1 - e^{-\overline{A}/\lambda}.$$ (3.2)

The probability of the *Good* state is given by the formula

$$p_G = 1 - p_B = e^{-\overline{A}/\lambda}.$$ (3.3)

Assuming the Clark model, the following formula for the level crossing rate is obtained in [65]:

$$LCR(x) = \sqrt{\frac{2\pi x}{\lambda}} f_D e^{-x/\lambda}, \tag{3.4}$$

where f_D is Doppler shift. Also, the following two explicit formulas for parameters q and q' (they are introduced in the previous section) are derived

$$q = \frac{LCR(\overline{A})}{Rp_B} = \frac{\sqrt{\frac{2\pi\overline{A}}{\lambda}} f_D e^{-\overline{A}/\lambda}}{R(1 - e^{-\overline{A}/\lambda})}, \tag{3.5}$$

$$q' = \frac{LCR(\overline{A})}{Rp_G} = \frac{\sqrt{\frac{2\pi\overline{A}}{\lambda}} f_D}{R}. \tag{3.6}$$

where R is the symbol rate. Taking into account (3.5), (3.6), and the following formula for the maximum Doppler shift [1]:

$$f_D = \frac{v f_c}{c}, \tag{3.7}$$

where f_c and c are transmitted frequency and velocity of light respectively, we obtain

$$q = \frac{LCR(\overline{A})}{Rp_B} = \frac{\sqrt{\frac{2\pi\overline{A}}{\lambda}} v f_c e^{-\overline{A}/\lambda}}{Rc(1 - e^{-\overline{A}/\lambda})}, \tag{3.8}$$

$$q' = \frac{LCR(\overline{A})}{Rp_G} = \frac{\sqrt{\frac{2\pi\overline{A}}{\lambda}} v f_c}{Rc}. \tag{3.9}$$

The timestep Δt (see description of timestep in the Section 3.1) satisfies

$$\Delta t \leq \frac{1}{2f_D}. \tag{3.10}$$

---

[1]Maximum Doppler shift is consistent with geometry of the problem, since vehicles follow the signal propagation path.

3.4 Results

Let $p_G(m)$ be the probability that at the moment $m\Delta t$ two fixed consecutive cars can establish a connection. We assume that the system is initially in the equilibrium state (for more details see Section 3.5). In the above mentioned section, the probability $p_G(m)$ of a successful connection between two consecutive cars at the moment of time $m\Delta t$ satisfies the following formula:

$$p_G(m) = \frac{q}{q + q'}, \tag{3.11}$$

where parameters q and q' are calculated by the formulas (3.8) and (3.9). Thus, probability $p_G(m)$ does not depend on m, and, therefore, we denote it by p_G.

Let symbol $P_{twocars}(m)$ stand for the probability that connection between two cars once established has duration $m\Delta t$. The probability $P_{twocars}(m)$ is given by the formula

$$P_{twocars}(m) = (1 - q')^{m-1}q'. \tag{3.12}$$

The average link duration $\overline{T}_{twocars}$ and its variance $\sigma^2_{twocars}$ of the distribution (3.12) are determined by the formulas

$$\overline{T}_{twocars} = \frac{\Delta t}{q'}, \tag{3.13}$$

$$\sigma^2_{twocars} = \frac{(1 - q')\Delta t^2}{q'^2}. \tag{3.14}$$

In our model, each vehicle connects only to the closest forward and backward cars. Therefore, in the framework of our model, clusters are formed by several consecutive cars. In Figure 3.2 below, five cars are depicted that form two clusters. The first cluster is formed by cars 1, 2, 3 and the second cluster is composed of cars 4 and 5. Significantly, cars 1 and 3 cannot communicate directly, and the only communication way for them is through car 2, while vehicles 3 and 4 are disconnected. We prove that the connectivity probability is given by the

Figure 3.2: Network of 5 cars that form two clusters.

formula $q'/(q + q')$. Therefore, connectivity state between two consecutive vehicles depends on the parameters listed in Section 3.3. It is worth mentioning that this probability is close to 1 (stable connection) if $q' \approx 1$ and $q \approx 0$.

We derive the probability $P_{clust}(m)$ that the cluster formed by the cars $k, k+1, \ldots, k+s$ once being formed exists exactly time $m\Delta t$. Let us introduce parameter γ by the formula

$$\gamma = \begin{cases} 2, & \text{if } 1 < k \text{ and } k + s < n, \\ 1, & \text{if } k = 1 \text{ or } k + s = n, \text{ but not both,} \\ 0, & \text{if } k = 1 \text{ and } k + s = n. \end{cases} \tag{3.15}$$

The probability $P_{clust}(m)$ satisfies the formula

$$P_{clust}(m) = (1 - q')^{s(m-1)}(1 - q)^{\gamma(m-1)}(1 - (1 - q')^s(1 - q)^\gamma). \tag{3.16}$$

The average existence cluster lifetime $\overline{T}_{clust}$ and variance σ^2_{clust} of the distribution (3.16) can be obtained by the formulas

$$\overline{T}_{clust} = \frac{\Delta t}{1 - (1 - q')^s(1 - q)^\gamma}, \tag{3.17}$$

$$\sigma^2_{clust} = \frac{(1 - q)^\gamma(1 - q')^s \Delta t^2}{(1 - (1 - q)^\gamma(1 - q')^s)^2}, \tag{3.18}$$

where γ is determined in (3.15).

We study not only the duration of a cluster lifetime, but also the probability that cluster does exist between two given moments of time. More precisely, we derive the following formula for the probability $P_{clust}(m, l)$ that the cluster that consists of cars $k, k + 1, \ldots, k + s$ is formed at the moment of time $m\Delta t$, exists until $l\Delta t$ $(l \geq m)$, and does not exist at the moment $(l + 1)\Delta t$ as follows:

$$P_{clust}(m, l) = \left\{ p_G^s (1 - p_G)^\gamma - (1 - q')^s p_G^s (1 - p_G)^\gamma (1 - q)^\gamma \right\}$$
$$\times (1 - q')^{s(l-m)} (1 - q)^{\gamma(l-m)} \left\{ 1 - (1 - q')^s (1 - q)^\gamma \right\}, \quad (3.19)$$

where p_G and γ are given by the formulas (3.11) and (3.15).

Finally, we analyse the property of ω-stability of a connection that is defined as the ability of two vehicles to establish connection within every time interval $\omega\Delta t$. More formally, we assume that connection is ω-stable between moments $m\Delta t$ and $l\Delta t$, if there is at least one successful connection not later that the moment $(m + \omega)\Delta t$, time difference between every two consecutive connections does not exceed $\omega\Delta t$ and the last connection is established at the moment $(l - \omega)\Delta t$ or later. At the end of Section 3.5, we derive the algorithm for finding the probability that the connection between two cars is ω-stable on a pre-selected interval. The algorithm has linear complexity and recurrently calculates the probability under the assumption of known parameters q and q'.

3.5 Mathematical derivations

3.5.1 Probability of two cars being connected at the moment $m\Delta t$

Let $p_G(m)$ be the probability that there is a connection between predetermined consecutive cars at the moment of time $m\Delta t$ and by $p_B(m)$ the probability that there is no connection at $m\Delta t$. Let us find the limiting distributions $p_G(\infty) = \lim_{m \to \infty} p_G(m)$ and $p_B(\infty) = \lim_{m \to \infty} p_B(m)$.

According to the Markov model (see Figure 3.1), we have

$$p_G(m) = p_G(m-1)(1-p) + p_B(m-1)q, \tag{3.20}$$

$$p_G(m) + p_B(m) = 1. \tag{3.21}$$

When $m \to \infty$, the equations (3.20) and (3.21) take the following form:

$$p_G(\infty) = p_G(\infty)(1-q') + p_B(\infty)q, \tag{3.22}$$

$$p_G(\infty) + p_B(\infty) = 1. \tag{3.23}$$

Solving system of linear equations (3.22) and (3.23), we obtain

$$p_G(\infty) = \frac{q}{q+q'}, \tag{3.24}$$

$$p_B(\infty) = \frac{q'}{q+q'}. \tag{3.25}$$

We assume that at the moment when we observe the system, it has reached its limiting distribution, therefore, the probabilities at every moment $m\Delta t$ equal the limiting probabilities

$$p_G(m) = \frac{q}{q+q'}, \tag{3.26}$$

$$p_B(m) = \frac{q'}{q+q'}. \tag{3.27}$$

Thus, we denote them p_G and p_B.

3.5.2   Distribution of link duration

Let us consider two consecutive cars. We find the probability $P_{twocars}(m)$ of the event that the connection lifetime between these cars is exactly $m\Delta t$. In other words, if the connection

between cars is firstly established at the moment of time Δt, we calculate the probability that the cars keep the connection up to the time instant $m\Delta t$, and at the time $(m+1)\Delta t$ the connection cannot be established. This probability is given by the formula

$$P_{twocars}(m) = (1 - q')^{m-1}q', \tag{3.28}$$

since at the time Δt the system is in the *Good* state, and the probability that it remains in the *Good* state at the moments $2\Delta t, 3\Delta t, \ldots m\Delta t$ is $(1 - q')^{m-1}$ (see Markov diagram in Figure 1), and the probability that the system changes the state from *Good* to *Bad* at the time instant $(m + 1)\Delta t$ is q'.

Using relation (3.28), we can calculate the average link duration $\overline{T}_{twocars}$ and variance $\sigma^2_{twocars}$ of the distribution by the formulas

$$\overline{T}_{twocars} = \sum_{l=1}^{\infty} l\Delta t P_{twocars}(l) = \sum_{l=1}^{\infty} l\Delta t (1 - q')^{l-1}q' = \frac{\Delta t}{q'}, \tag{3.29}$$

$$\sigma^2_{twocars} = \sum_{l=1}^{\infty} (l\Delta t)^2 P_{twocars}(l) - \overline{T}^2_{twocars} = \sum_{l=1}^{\infty} l^2 \Delta t^2 (1 - q')^{l-1}q' - \frac{\Delta t^2}{q'^2} =$$

$$\left(\frac{2 - q'}{q'^2} - \frac{1}{q'^2} \right)\Delta t^2 = \frac{(1 - q')\Delta t^2}{q'^2}. \tag{3.30}$$

3.5.3   Distribution of a cluster lifetime

In this section we find the probability $P_{clust}(m)$ that the existence duration of a cluster composed by cars $k, k + 1, \ldots, k + s$ equals exactly $m\Delta t$. We could assume that the cluster exists since the moment Δt until the moment $m\Delta t$ and does not exist at the moment $(m + 1)\Delta t$. The probability $P_{clust}(m)$ satisfies the formula

$$P_{clust}(m) = (1 - q')^{s(m-1)}(1 - q)^{\gamma(m-1)}(1 - (1 - q')^{s}(1 - q)^{\gamma}), \tag{3.31}$$

where γ is defined by (3.15).

Let us derive the equation (3.31). We consider only the case where $k > 1$ and $k + s < n$, because in other cases derivations are similar. The probability $P_{clust}(m)$ can be calculated by the formula

$$P_{clust}(m) = P_1 P_2, \tag{3.32}$$

where P_1 is a probability that the cluster exists at the moments $2\Delta t, 3\Delta t, \ldots m\Delta t$ under the condition that it exists at the moment Δt, and P_2 is a probability that the cluster does not exist at the moment $(m + 1)\Delta t$ under the assumption that it exists at the moment $m\Delta t$. First, we calculate the probability P_1. The fact that the cluster $k, k + 1, \ldots, k + s$ exists at the moment Δt simply means that at this moment there is a connection between every pair of cars $(k, k + 1)$, $(k + 1, k + 2)$, ..., $(k + s - 1, k + s)$ and there is no connection between pairs of cars $(k, k - 1)$ and $(k + s, k + s + 1)$. The probability that all pairs of cars $(k, k + 1)$, $(k + 1, k + 2)$, ..., $(k + s - 1, k + s)$ preserve a connection at the moment $2\Delta t$ is $(1 - p)^s$ (see Markov diagram on Figure 1). The probability that the pairs of cars $(k, k - 1)$ and $(k + s, k + s + 1)$ continue to be disconnected at the moment $2\Delta t$ is $(1 - q)^2$. Thus, the probability that the cluster exists at the moment $2\Delta t$ is $(1 - q')^s (1 - q)^2$. Continuing these derivations until the moment $m\Delta t$, we find the probability P_1 given as follows:

$$P_1 = (1 - q')^{s(m-1)} (1 - q)^{2(m-1)}, \tag{3.33}$$

since there are $m - 1$ transitions between steps Δt and $m\Delta t$ and at each step we need to multiply the answer by the transitional probability $(1 - q')^s (1 - q)^2$. Finally, we derive the probability P_2. The cluster should cease to exist at the moment $(m + 1)\Delta t$. The probability that the cluster continues to exist at the next moment is $(1 - q')^s (1 - q)^2$, therefore, the probability that it does not exist is

$$P_2 = 1 - (1 - q')^s (1 - q)^2. \tag{3.34}$$

By substituting formulas (3.33) and (3.34) into (3.32) we derive

$$P_{clust}(m) = (1 - q')^{s(m-1)}(1 - q)^{\gamma(m-1)}(1 - (1 - q')^s(1 - q)^\gamma).$$

Equation (3.31) is proven.

Using standard formula for mathematical expectation, we obtain that the average time of cluster existence $\overline{T}_{clust}$ is determined by the following formulas:

$$\overline{T}_{clust} = \sum_{l=1}^{\infty} l\Delta t P_{clust}(l) =$$

$$\sum_{l=1}^{\infty} l\Delta t (1 - q')^{s(l-1)}(1 - q)^{\gamma(l-1)}(1 - (1 - q')^s(1 - q)^\gamma)$$

$$= \frac{\Delta t}{1 - (1 - q')^s(1 - q)^\gamma}. \quad (3.35)$$

Similarly, we can obtain the following formula for variance of cluster existence σ^2_{clust} :

$$\sigma^2_{clust} = \sum_{l=1}^{\infty} (l\Delta t)^2 P_{clust}(l) - \overline{T}^2_{clust}$$

$$= \frac{(1 + (1 - q')^s(1 - q)^\gamma)\Delta t^2}{(1 - (1 - q')^s(1 - q)^\gamma)^2} - \left(\frac{\Delta t}{1 - (1 - q')^s(1 - q)^\gamma}\right)^2$$

$$= \frac{(1 - q)^\gamma(1 - q')^s\Delta t^2}{(1 - (1 - q)^\gamma(1 - q')^s)^2}. \quad (3.36)$$

3.5.4 Probability of cluster existence between fixed moments of time

In the section, we derive the probability of a cluster existence between two particular moments of time. We consider only the case where $k > 1$ and $k + s < n$, because in other cases derivations are similar. However, before solving the problem, we find the probability $P_{clustbeg}(m)$ that at the time $m\Delta t$ the cluster that consists of cars $k, k + 1, \ldots, k + s$, is formed, and it does not exist at the moment $(m - 1)\Delta t$. Let us denote the probability of the event that at the moment $m\Delta t$ there is a cluster consisting of cars $k, k + 1, \ldots, k + s$ by P_1 and the probability that the cluster

$k, k + 1, \ldots, k + s$ exists at the moments $(m - 1)\Delta t$ and $m\Delta t$ by P_2. Therefore, the probability $P_{clustbeg}(m)$ is given as follows:

$$P_{clustbeg}(m) = P_1 - P_2. \tag{3.37}$$

Next, we calculate each of the probabilities P_1 and P_2 separately. Firstly, we establish that

$$P_1 = p_G^s(1 - p_G)^2. \tag{3.38}$$

Indeed, if the cluster $k, k + 1, \ldots, k + s$ exists at the moment $m\Delta t$ then the pairs of cars $(k, k+1)$, $(k + 1, k + 2), \ldots, (k + s - 1, k + s)$ are connected (each with the probability p_G). Thus, the probability of this event is p_G^s, since there are s such pairs. Additionally, there is no connection between the pairs of cars $(k, k - 1)$ and $(k + s, k + s + 1)$ (each of disconnections occurs with the probability $1 - p_G$). Multiplying the aforementioned probabilities, we get the relation (3.38). Regarding the probability P_2, we prove the formula

$$P_2 = (1 - q')^s p_G^s(1 - p_G)^2(1 - q)^2. \tag{3.39}$$

The probability P_2 can be represented as follows:

$$P_2 = P_3 P_4, \tag{3.40}$$

where P_3 is a probability that the cluster $k, k + 1, \ldots, k + s$ exists at the moment $(m - 1)\Delta t$ and P_4 is a probability that it continues to exist at the moment of time $m\Delta t$ under the assumption that it exists at the moment $(m - 1)\Delta t$. Repeating derivations for the probability P_1 in the case of the time instant $(m - 1)\Delta t$, we have

$$P_3 = P_1 = p_G^s(1 - p_G)^2. \tag{3.41}$$

Next, we prove the relation

$$P_4 = (1 - q')^s (1 - q)^2.$$
(3.42)

Suppose, the cluster exists at the moment of time $(m - 1)\Delta t$. The probability that a connection between any pair of the consecutive cars of the cluster is preserved at the moment $m\Delta t$ equals $1 - q'$. Consequently, the probability that the connection between all s pairs of cars is preserved equals $(1 - q')^s$. The pairs of cars $(k-1, k)$ and $(k+s, k+s+1)$ cannot establish a connection at the moment $(m - 1)\Delta t$, the probability that these pairs remain disconnected is $(1 - q)^2$. Multiplying two probabilities $(1 - q')^s$ and $(1 - q)^2$, we derive the formula (3.42).

From (3.38)–(3.42), we conclude that

$$P_{clustbeg}(m) = p_G^s (1 - p_G)^2 - (1 - q')^s p_G^s (1 - p_G)^2 (1 - q)^2.$$
(3.43)

Using the obtained probabilities, we can find the probability $P_{clust}(m, l)$ that the cluster $k, k + 1, \ldots k + s$ exists between moments of time $m\Delta t$ and $l\Delta t$ ($l \geq m$). In other words, it is the probability that the cluster exists from the time instant $m\Delta t$ to $l\Delta t$ ($l \geq m$), and does not exist at the moments $(m - 1)\Delta t$ and $(l + 1)\Delta t$. The probability $P_{clust}(m, l)$ satisfies the formula

$$P_{clust}(m, l) = (p_G^s (1 - p_G)^2 - (1 - q')^s p_G^s (1 - p_G)^2 (1 - q)^2)$$
$$\times (1 - q')^{s(l-m)} (1 - q)^{2(l-m)} (1 - (1 - q')^s (1 - q)^2). \quad (3.44)$$

The probability $P_{clust}(m, l)$ can be decomposed in the product of three terms as follows:

$$P_{clust}(m, l) = P_{clustbeg}(m) P_5 P_6,$$
(3.45)

where P_5 is a probability that the cluster exists until the moment $l\Delta t$, and P_6 is a probability that the cluster ceases to exist at the moment $(l + 1)\Delta t$ assuming that it exists at the moment

$l\Delta t$. First, we consider the probability P_5 and prove the formula

$$P_5 = (1 - q')^{s(l-m)}(1 - q)^{2(l-m)}. \tag{3.46}$$

By analogy with derivations of (3.33), we deduce that the probability, that the connection established between the pairs of cars $(k, k + 1)$, $(k + 1, k + 2), \ldots, (k + s - 1, k + s)$ at the moment of time $m\Delta t$ is preserved at the time instances $(m + 1)\Delta t, \ldots, l\Delta t$, equals $(1 - q')^{s(l-m)}$. Following the same logic of (3.33), we derive that the probability, that the pairs of cars $(k-1, k)$ and $(k + s, k + s + 1)$ do not communicate at the moments of time $(m + 1)\Delta t, \ldots, l\Delta t$ under the condition that the connection is not established at the moment $(m - 1)\Delta t$ equals $(1 - q)^{2(l-m)}$. The probability P_5 can be calculated as a product of these probabilities and, therefore, (3.46) is proven. By repeating steps of the proof of the equation (3.34), we conclude that

$$P_6 = 1 - (1 - q')^s(1 - q)^2. \tag{3.47}$$

From the formulas (3.43), (3.45)–(3.47), we derive (3.44).

3.5.5 ω-stable connection

Definition   A connection between moments of time m and l ($m \leq l$) is ω-stable if the time difference between every two consecutive connections does not exceed time $\omega\Delta t$. Additionally we assume that there exists at least one successful connection established not later than $(m + \omega)\Delta t$, and the last connection is established at the moment of time $(l - \omega)\Delta t$ or later.

It does not make sense to consider ω-stable connection if $l < m + \omega$, because in this case every connection is ω-stable. Therefore, we assume that $l \geq m + \omega$. Also, we suppose that ω is an integer number and $\omega \geq 2$.

In the section, we find the probability $P_\omega(m, l)$ that the connection between two consecutive cars is ω-stable between the moments $m\Delta t$ and $l\Delta t$. Let us introduce the function $h(a, b)$ equaling the probability that at the moment of time $a\Delta t$, the last connection is established at the time

instant $b\Delta t$, and the function $g(a)$ equaling the probability that there is no connection at the time interval $[0, a\Delta t]$. To derive the function $g(a)$ explicitly, we should multiply the probability $(1 - p_G)$, that at the moment $m\Delta t$ there is no connection, by the probability $(1 - q)^{a-m}$ that at the moments $(m + 1)\Delta, \ldots a\Delta$ there is no connection as well. Therefore, the function $g(a)$ has the form

$$g(a) = \begin{cases} (1 - p_G)(1 - q)^{a-m}, & \text{if } a < m + \omega \\ 0, & \text{if } a \geq m + \omega. \end{cases} \tag{3.48}$$

The following recurrences take place:

$$h(a, b) = 0, b < a - \omega, \tag{3.49}$$

$$h(a, b) = h(a - 1, b)(1 - q), a - \omega \leq b < a - 1, \tag{3.50}$$

$$h(a, a - 1) = h(a - 1, a - 1)q', \tag{3.51}$$

$$h(a, a) = h(a - 1, a - 1)(1 - q') + \sum_{r=a-\omega}^{a-2} h(a - 1, r)q + g(a - 1)q. \tag{3.52}$$

The formula (3.49) holds, since for $b < a - \omega$, the probability $h(a, b)$ equals zero due to the fact that the time difference between the last connection at the moment of time $b\Delta t$ and the next connection exceeds $\omega\Delta t$. The formula (3.50) holds, because if $b < a - 1$ then at the moment $(a - 1)\Delta t$ the system is in the *Bad* state, and the probability that it remains in this state is $1 - q$. The formula (3.51) is derived by applying the same logic. Finally, in (3.52), the recursive formula for the probability that the connection is established at the time instant $a\Delta t$, is obtained. Due to the ω-stability of the connection, the previous time of the connection lies between $(a-\omega)\Delta t$ and $(a-1)\Delta t$. The summand $h(a-1, a-1)(1-q')$ occurs in the formula (3.52), since it is the probability that at the moment of time $(a - 1)\Delta t$ the system is in the *Good* state and remains at this state up to the moment of time $a\Delta t$. The sum $\sum_{r=a-\omega}^{a-2} h(a - 1, r)q$ contains probabilities $h(a - 1, r)$ that the time instant $r\Delta t$ is the last moment of time when the system is in the *Good* state. It means that the system is in the *Bad* state at the moment $(a - 1)\Delta t$ and goes

to the *Good* state at the moment $a\Delta t$. It explains the multiplication of the terms $h(a-1, l)$ by q.

The probability $P_\omega(m, l)$ can be calculated by the formula

$$P_\omega(m, l) = \sum_{r=l-\omega}^{l} h(l, r), \tag{3.53}$$

because the last moment of time before $l\Delta t$ when the system is in the *Good* state, can be only $l\Delta t, (l-1)\Delta t, \ldots, (l-\omega)\Delta t$.

The following formulas hold:

$$h(m, m) = p_G, \tag{3.54}$$

$$h(m+1, m+1) = p_G, \tag{3.55}$$

where p_G is determined by the formula (3.24). The equality (3.54) takes place, since the probability that connection is established at the moment $m\Delta$ is exactly p_G, the formula (3.55) is obtained similarly, taking into account that $\omega \geq 2$. Under condition of $a > b$, application of the relations (3.50) and (3.51) provides the expression

$$h(a, b) = h(a-1, b)(1-q) = h(a-2, b)(1-q)^2 = \ldots$$
$$= h(b+1, b)(1-q)^{a-b-1} = h(b, b)(1-q)^{a-b-1}q'. \tag{3.56}$$

From (3.52) and (3.56), we obtain

$$h(a, a) = h(a-1, a-1)(1-q') + \sum_{r=a-\omega}^{a-2} h(a-1, r)q + g(a-1)q$$
$$= h(a-1, a-1)(1-q') + \sum_{r=a-\omega}^{a-2} h(r, r)(1-q)^{a-r-2}q'q + g(a-1)q. \tag{3.57}$$

By substituting $a-1$ instead of a into (3.57), we derive

$$h(a-1, a-1) = h(a-2, a-2)(1-q') + \sum_{r=a-\omega-1}^{a-3} h(r, r)(1-q)^{a-r-3}q'q + g(a-2)q. \tag{3.58}$$

Multiplying (3.58) by $(1 - q)$, we get the following equality:

$$h(a - 1, a - 1)(1 - q) = h(a - 2, a - 2)(1 - q')(1 - q)$$

$$+ \sum_{r=a-\omega-1}^{a-3} h(r, r)(1 - q)^{a-r-2} q' q + g(a - 2)q(1 - q). \quad (3.59)$$

Subtraction (3.59) from (3.57) gives us

$$h(a, a) - h(a - 1, a - 1)(1 - q) = h(a - 1, a - 1)(1 - q')$$

$$- h(a - 2, a - 2)(1 - q')(1 - q) + h(a - 2, a - 2)q' q$$

$$- h(a - \omega - 1, a - \omega - 1)(1 - q)^{\omega-1} q' q + g(a - 1)q - g(a - 2)q(1 - q). \quad (3.60)$$

After simplifying (3.60), we finally derive

$$h(a, a) = h(a - 1, a - 1)(2 - q' - q)$$

$$- h(a - 2, a - 2)(1 - q' - q) - h(a - \omega - 1, a - \omega - 1)(1 - q)^{\omega-1} q' q$$

$$+ g(a - 1)q - g(a - 2)q(1 - q). \quad (3.61)$$

Let us introduce the function $f(a)$ by the formula

$$f(a) = h(a, a). \quad (3.62)$$

Expressions (3.61) and (3.62) provide the relation

$$f(a) = f(a - 1)(2 - q' - q) - f(a - 2)(1 - q' - q)$$

$$- f(a - \omega - 1)(1 - q)^{\omega-1} q' q + g(a - 1)q - g(a - 2)q(1 - q). \quad (3.63)$$

Our next goal is to express the probability $P_\omega(m, l)$ in terms of the function $f(a)$. Substitu-

---

Function for calculating the probability $P_\omega(m, l)$

1: **function** PROBABILITY OF ω-STABLE CONNECTION(m, l, ω, p, q)
2:     $f = ARRAY[1..l + 1]$;
3:     $g = ARRAY[1..l + 1]$;
4:     $f = \mathbf{0}$;
5:     $f(m) = \frac{q}{q'+q}$;
6:     $f(m + 1) = \frac{q}{q'+q}$;
7:     $g = \mathbf{0}$;
8:     **for** $a = m$ to $m + \omega - 1$ **do**
9:         $g(a) = \frac{q'}{q'+q}(1 - q)^{a-m}$;
10:     **end for**
11:     **for** $a = m + 2$ to $l + 1$ **do**
12:         $f(a) = f(a - 1)(2 - q' - q) - f(a - 2)(1 - q' - q) + g(a - 1)q - g(a - 2)q(1 - q)$;
13:         **if** $a - \omega - 1 \geq m$ **then**
14:             $f(a) = f(a) - f(a - \omega - 1)(1 - q)^{\omega-1}q'q$;
15:         **end if**
16:     **end for**
17:     **return** $\frac{1}{q}\{f(l + 1) - f(l)(1 - q' - q) + f(l - \omega)(1 - q)^{\omega-1}q'q - g(l)q\}$
18: **end function**

---

tion $a = l + 1$ into (3.52) gives us

$$h(l + 1, l + 1) = h(l, l)(1 - q') + q \sum_{r=l+1-\omega}^{l-1} h(l, r) + g(l)q. \tag{3.64}$$

Subtracting (3.53) multiplied by q from (3.64) produces the following equality:

$$h(l + 1, l + 1) - qP_\omega(m, l) = h(l, l)(1 - q') - h(l, l)q - h(l, l - \omega)q + g(l)q. \tag{3.65}$$

Applying (3.56) to (3.65), we get

$$h(l + 1, l + 1) - qP_\omega(m, l) = h(l, l)(1 - q' - q) - h(l - \omega, l - \omega)(1 - q)^{\omega-1}q'q + g(l)q. \tag{3.66}$$

Therefore, taking into account (3.62), we derive

$$P_\omega(m, l) = \frac{1}{q}\{f(l + 1) - f(l)(1 - q' - q) + f(l - \omega)(1 - q)^{\omega-1}q'q - g(l)q\}. \tag{3.67}$$

From (3.54) and (3.55), we obtain

$$f(m) = p_G, \ f(m + 1) = p_G. \tag{3.68}$$

Combining previous formulas we derive the algorithm above for calculating the probability $P_\omega(m, l)$. In lines 2, 3, 4 and 7 two arrays f and g are declared and initialized to zero. Lines 5, 6 and 9 are obtained by the formulas (3.68) and (3.48), respectively. Lines 12–14 are derived by (3.63), the return value in the line 17 is obtained by the formula (3.67).

3.6 Simulations

In order to verify the theoretical development we conduct a number of numerical simulations with the following parameters [89]:

$$R = 10^5 \text{symbol/sec}, \overline{A}/\lambda = 0.1, fc = 3.9\text{GHz}^2$$

The speed of cars v takes the following values:

$$v = 30, 60, 90 \text{ km/h}.$$

The Doppler frequency shift is calculated by the formula

$$f_D = \frac{v f_c}{c}.$$

---

[2]Modern standarts would use additional frequency 5.9 GHz, but the calculation methodology would remain the same.

For the speeds $v = 30, 60, 90$ km/h Doppler frequency f_D equals $108, 217, 325$ Hz, respectively.

The timestep Δt between two consecutive transmissions should satisfy

$$\Delta t \leq \frac{1}{2f_D}.$$

We assume that Δt is given by the formula

$$\Delta t = \frac{1}{10f_D}.$$

For speeds $v = 30, 60, 90$ km/h, values of the parameter Δt are $9 \times 10^{-4}, 4.6 \times 10^{-4}, 3 \times 10^{-4}$

seconds, respectively. Under these assumptions from (3.6) and (3.5), we get the following

values of parameters q and q' for speeds $v = 30, 60, 90$ km/h:

$$q = 8.1 \times 10^{-3}, 1.6 \times 10^{-2}, 2.5 \times 10^{-2},$$

$$q' = 8.5 \times 10^{-4}, 1.7 \times 10^{-3}, 2.5 \times 10^{-3}.$$

We assume that the network consists of $n = 10$ vehicles.

We use logarithmic scale for all graphs below. We consider the probabilities of the event

that the system remains connected throughout the time interval $[0, t]$. By this, we mean that

the connection is established at each moment of time on a discrete time grid during the interval

$[0, t]$. If we increase the value of the parameter t, then we increase the number of time moments

when the vehicles should remain connected, so the probability of this event decreases. In other

words, the considered probabilities are monotonically decreasing functions of the parameter

t. All graphs are represented as straight lines, since the probabilities exponentially decreasing

with time, which in logarithmic scale, is represented as linear decreasing straight lines. We

compare the results obtained by the formulas and the results of numerical simulations. The

simulation results obtained by generating evolution of the vehicle network on the road many

times, calculating number of the cases where required characteristics of the network occur and

dividing this number by the number of trials. The values of parameters q' and q are too small (about 10^{-3}–10^{-4}) that results in negligible probabilities computed by the formulas (3.12), (3.16) and (3.19). It makes impossible to achieve an appropriate simulation precision for the reasonable time. Therefore, we do not depict the corresponding simulation results in Figures 3.3, 3.4 and 3.5.

Figure 3.3 presents graphs of the distribution (3.12) of the link duration between two consecutive cars for $v = 30, 60, 90$ km/h.
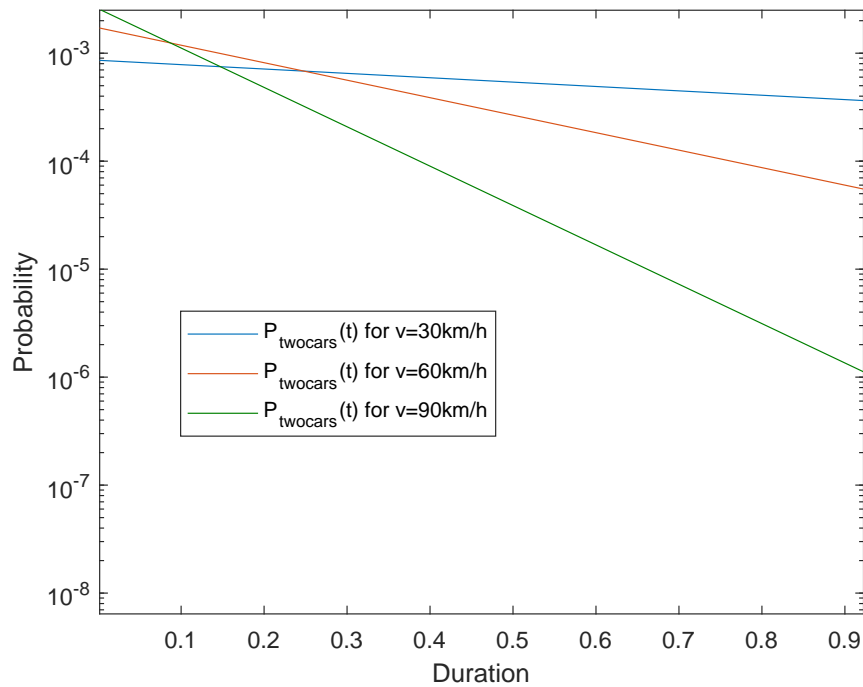
Figure 3.3: Graphs of the distribution (3.12) of the link duration between two consecutive cars for $v = 30, 60, 90$ km/h.

Figure 3.4 shows the distribution (3.16) of the lifetime of the cluster that consists of cars 2, 3 and 4 for $v = 30, 60, 90$ km/h.

To illustrate decreasing of the probability (3.19) of the cluster existence between fixed moments of time $m\Delta t$ and $l\Delta t$, we fix the value of the parameter $m = 2$ and vary the value of $l = m, m + 1, \ldots$. We assume that the cluster consists of cars with numbers 2, 3 and 4. The graphs of the obtained functions of parameter l are shown in Figure 3.5.

Figure 3.4: Graphs of the probability $P_{clust}(m)$ given by the formula (3.16) for $v = 30, 60, 90$ km/h.

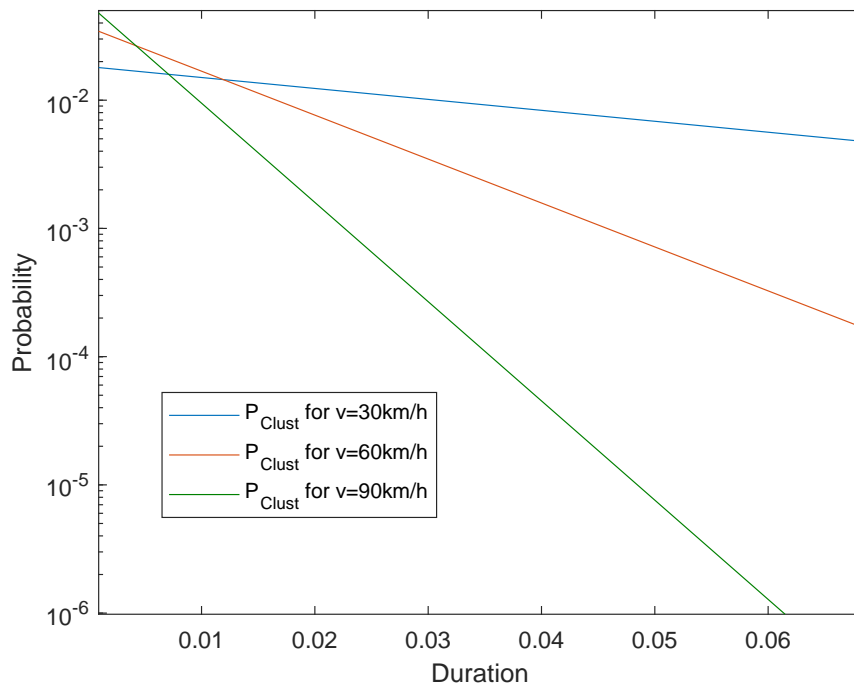Figure 3.6 demonstrates both the simulation results and the probabilities computed by the algorithm from Section 3.5.5. We fix an initial moment of time $m\Delta t$ and assume that $m = 2$, and vary $l = m, m + 1, \ldots$. Simulations results are obtained after 50000 generations of the car evolution process.

3.7   Simulations for large values of parameters q and q'

In the section, we perform simulations that confirm the correctness of the formulas (3.12), (3.16) and (3.19) for increased values of parameters q' and q, which have the order of 10^{-2}. The results are represented in Figures 3.7, 3.8, and 3.9. The graphs perfectly match each other, which proves the correctness of the above mentioned formulas.

Figure 3.7 presents two graphs of the numerical simulation and probability of the link duration $P_{twocars}$ given by the formula (3.12) for $q = 0.02$, $q' = 0.02$, $\Delta t = 0.01$ and 10^5 iterations.
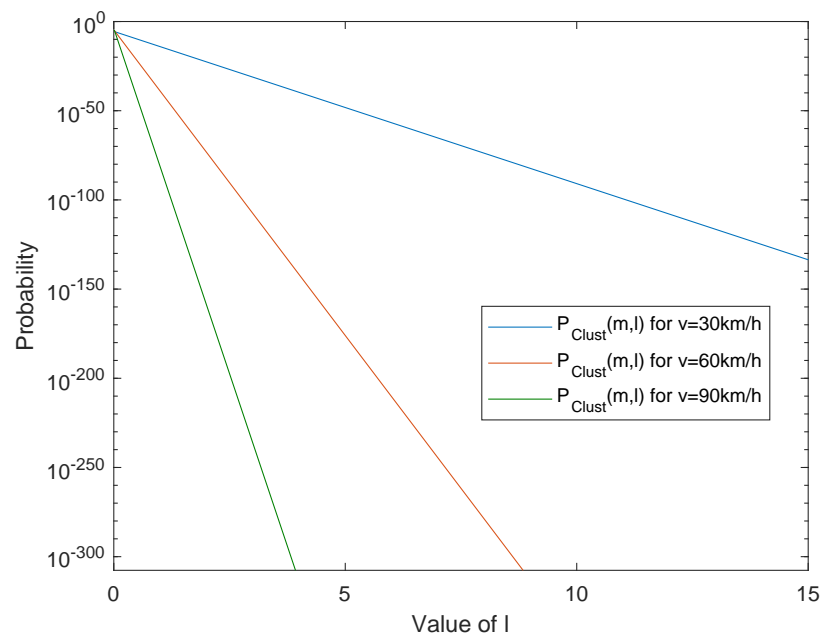
Figure 3.5: Graphs of the probability $P_{clust}(r,t)$ of the cluster existence between times given by the formula (3.19).
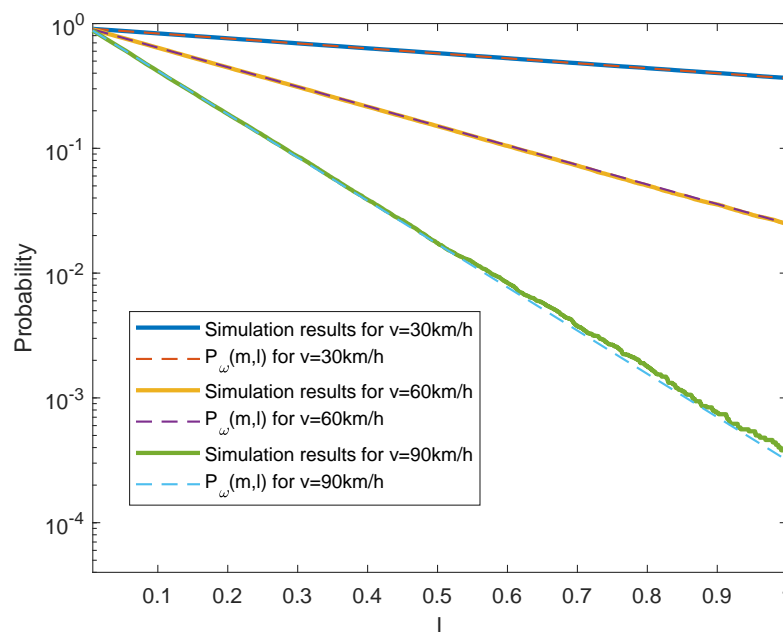
Figure 3.6: Graphs of numerical simulation of the probability of 3-stable connection between times $m = 2$ and l with variable l, and the probability $P_\omega(m,l)$ returned by algorithm from Section 3.5.5.
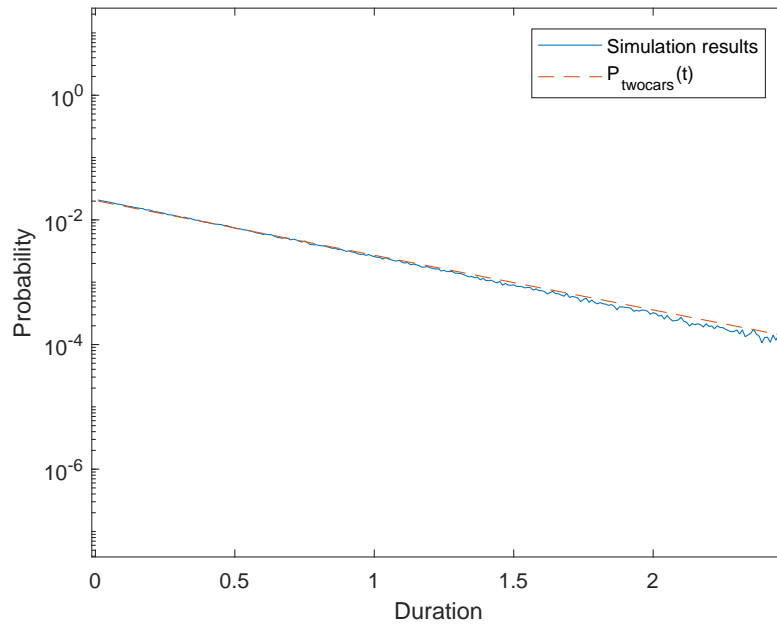
Figure 3.7: Graphs of the numerical simulation of the distribution of the connection lifetime duration between two consecutive cars and distribution (3.12).
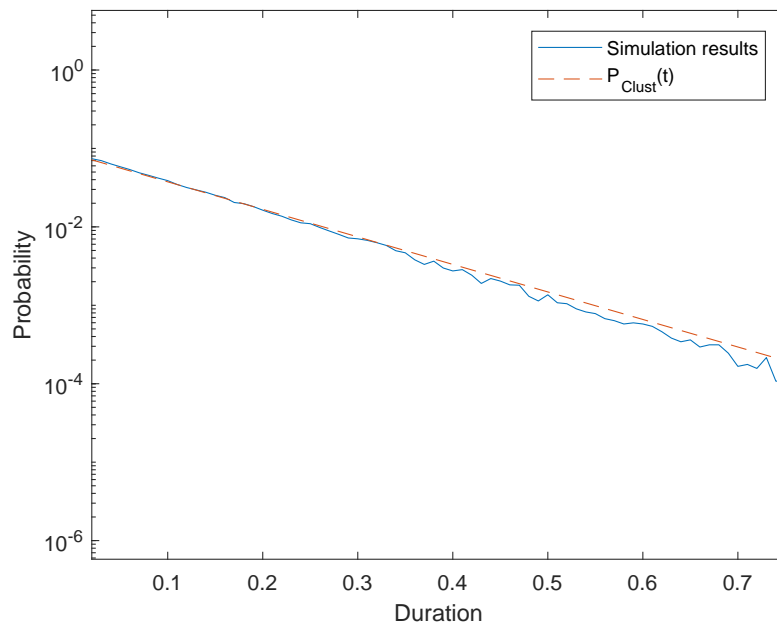
Figure 3.8: Graphs of the numerical simulations of the cluster lifetime and the probability $P_{clust}(m)$.
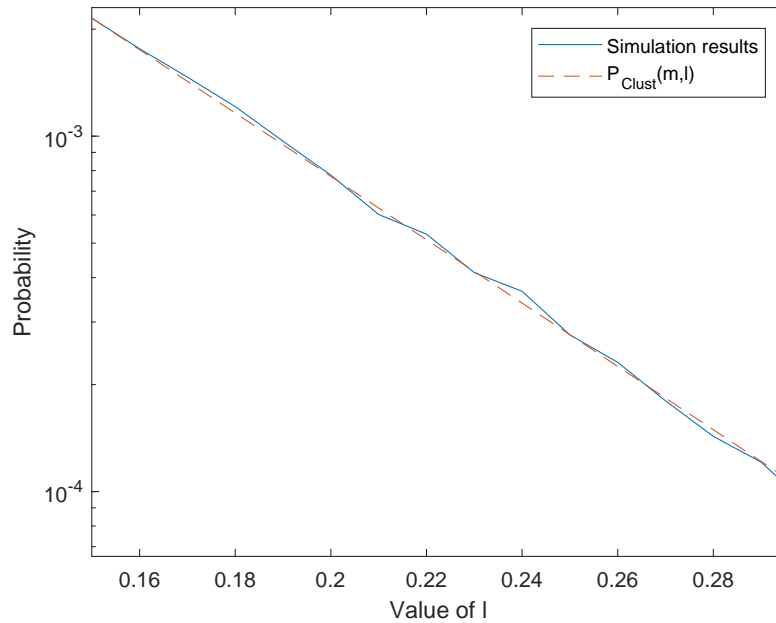
Figure 3.9: Graphs of the numerical simulation of the probability of the cluster existence between times $15\Delta t$ and $15\Delta t$, $16\Delta t \ldots 30\Delta t$ and $P_{clust}(m, l)$.

In Figure 3.8, the simulation results of the cluster lifetime duration are presented for the cluster that consists of cars 2, 3 and 4, and also, the graph of the predicted distribution $P_{clust}(m)$ for $q = 0.02$, $q' = 0.02$, $\Delta t = 0.01$ and 10^5 iterations.

We assume value of $m = 15$ to be constant and change the value of the parameter $l = 15, 16, \ldots 30$. Figure 3.9 depicts the graph of the cluster existence probability between times $m\Delta t$ and $l\Delta t$ (3.19) for $q = 0.05$, $q' = 0.05$, $\Delta t = 0.01$ and 10^6 iterations.

3.8   Chapter summary

In this chapter, we consider the evolution of the vehicle network on a highway and assume that the connection between each pair of consecutive cars can be established with a certain probability given by the Wang-Moayeri model. We derive the probability distributions that describe the evolution of such characteristics of the network as a distribution of the link duration and cluster lifetime. We also derive the probability that a cluster exists between fixed time

moments. Such a connection characteristic between two vehicles as a ω-stable connection is considered. This type of connection is introduced to simulate the real process of communication, in which communication can be lost for a while, but must be reestablished with some regularity. We propose a linear complexity algorithm for calculating the probability that the connection is ω-stable over a given time interval. All the derivations are performed under the assumption that the connectivity model is described by the two state Markov chain model and the parameters of the model are explicitly expressed through the parameters of the network.

Chapter 4

VANET statistics on a 2D map

4.1 Introduction

Previous chapters are devoted to the case in which the vehicles are located on a highway. In the literature, this case is popular due to the fact that the consideration of statistical properties in this case can be done using analytical derivations. The articles [10,17,47,88], which are already discussed in the Introduction and Introduction to Chapter 2, study such statistical properties of the network as the average cluster size, the probability of a successful connection, and the probability of full connectivity, etc. Our Chapter 2 derivations also add probability distributions of cluster size, largest cluster size, and the number of disconnected vehicles (we call them *idle* cars) to the list of studied properties. In [45, 56, 91, 104], the authors analyze network evolution in terms of link duration. In Chapter 3, in addition to link duration distribution, distributions of the cluster lifetime, the probability of the existence of a cluster between two fixed time moments, and the probability of ω-stable connection (connection that can be lost, but reestablished with certain regularity) are obtained and verified by simulations.

To the best of our knowledge, the case of a 2D map was still unexplored until the results presented in this chapter. The reason for this is that in this case, the model must take into account the work of traffic lights, the routes of cars on the map, as well as the traffic model.

These factors cannot be put into a precise analytical model, therefore, we consider here a simplified model that gives quite accurate approximations to the simulation results.

To perform the simulations, we develop a special vehicle traffic simulator on a 2D map with the realistic communication model from Chapter 2. In this Chapter, we derive such fundamental network distributions as cluster number distribution and average cluster size distribution. The obtained analytical formulas give quite accurate approximations of the simulation results in the case of urban and rural traffic.

Chapter 4 is organized as follows. In Section 4.2, we describe vehicle motion and communication models. In Section 4.3, we derive the probability distributions of the number of clusters and the average cluster size. In Section 4.4, we describe the Intelligent driver model, which is further used in the simulator. Section 4.5 is devoted to a discussion of the main simulator principles. In Section 4.6, we compare simulation results and theoretical results done in the cases of urban and rural traffic.

4.2   Road topology and communication model

Our results are derived under the following assumptions about the road topology, traffic model, and communication models:

1. The map consists of a connected system of several roads and intersections.

2. The roads are straight and long enough.

3. At each intersection, several roads with different numbers of lanes and at different angles converge. Throughout the road, the number of lanes is assumed to be the same.

4. All vehicles move according to the same motion model, and the probability distribution of intervehicle distance is the same for each road.

5. The vehicles communicate with each other only if they are moving along the same lane in the same direction.

6. There is at least one car on each lane of every road.

Let us clarify these conditions. Assumption 2 is necessary so we can use methods similar to the methods in Chapter 2. In Chapter 2, the road is assumed to have infinite length, so here we presume that the roads are long enough. Assumption 3 means that we consider complex road topologies. Examples of such topologies are shown in Figure 4.1. It is worth mentioning that in Figure 4.1(a) the roads are not perpendicular and in Figure 4.1(b) two roads with a different number of lanes converge. We also note that this model can also be used in the case in which some of the requirements are not satisfied, however, its accuracy would be lower. Assumption 6 is quite realistic since the roads are long enough.

We use the same cluster definition as given in Section 2.2. We assume that the car density on the map equals ρ and the total length of roads is L, then the number of cars on the map n equals $\lfloor \rho L \rfloor$, where $\lfloor x \rfloor$ is integer part of the number x. There may be several traffic lights on each road. We also need the value tr_j, which characterizes the topology of road junctions. Namely, tr_j is defined as follows:

$$tr_j = \text{number of lanes that pass through traffic light } j, \text{ but do not end there.} \quad (4.1)$$

For example, in Figure 4.1(a) tr_j equals 2, because two-lane road A continues after the intersection, and road B does not. Thus, road B is not taken into account. In Figure 4.1(b) tr_j equals 6, since road A has 2 lanes and road B has 4 lanes. Then value of tr is calculated by the formula

$$tr = \sum_j tr_j,$$

where the summation over all traffic lights j is implied. By r we denote the total number of lines on the whole map. For instance, there are 4 lines in Figure 4.1(a) and there are 6 lines in Figure 4.1(b).

We use the same communication model as in Chapter 2 taken from [17]. Let us denote SNR of the channel between vehicles by γ. According to this model the average SNR $\bar{\gamma}$ can be
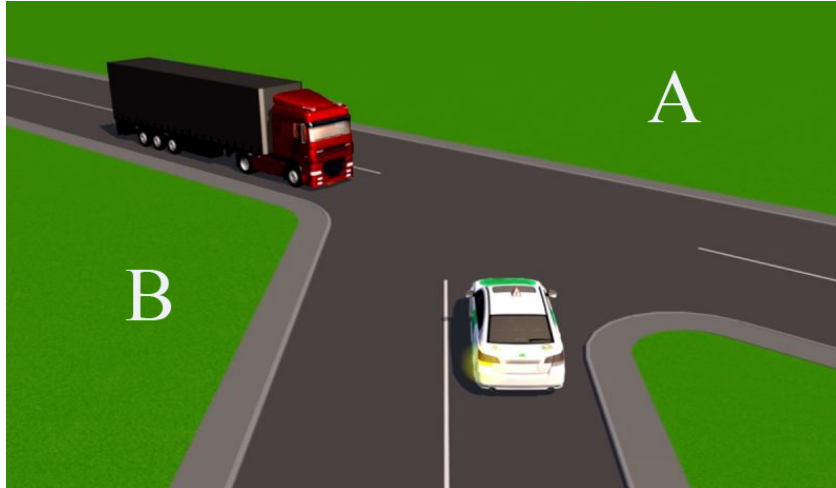
(a) Example 1
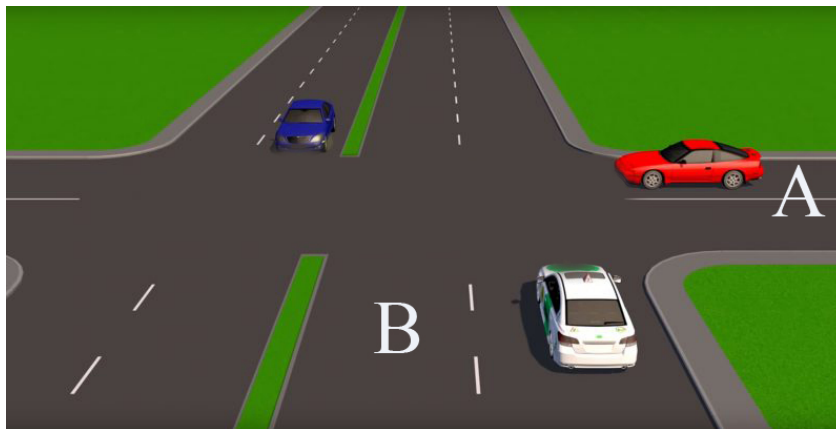
(b) Example 2

Figure 4.1: Crossroads examples

calculated by the following formula:

$$\overline{\gamma} = \frac{P_{tx}K}{d^{\alpha}W},$$ (4.2)

where P_{tx} is the transmit power, K is a constant associated with the path loss model, d is distance between cars, and α is the path loss exponent (for more details see Section 2.2).

The SNR density function $f_{\gamma}(x)$ is given by the formula

$$f_{\gamma}(x) = \frac{1}{\overline{\gamma}}e^{-x/\overline{\gamma}}.$$

It is assumed that there is a connection between the cars if γ is greater than the threshold Ψ.

We denote the distance between the vehicle at the intersection to the next car moving along the same line by d' and the distance between vehicles moving along the same line towards the same intersection by d'' (see Figure 4.2 for better understanding). We assume that the probability distributions of d' and d'' are known and described by the probability density functions $f_{d'}$ and $f_{d''}$. It is important to understand that the distance distribution of d'' is not a free movement distance distribution, since the movement of cars is hindered by other vehicles and the traffic lights. Then the probabilities p' and p'' (by analogy with (2.7)) are given by the formulas

$$p' = \int_0^\infty \exp\left(-\frac{\Psi x^\alpha W}{P_{tx} K}\right) f_{d'}(x)dx, \tag{4.3}$$

$$p'' = \int_0^\infty \exp\left(-\frac{\Psi x^\alpha W}{P_{tx} K}\right) f_{d''}(x)dx. \tag{4.4}$$
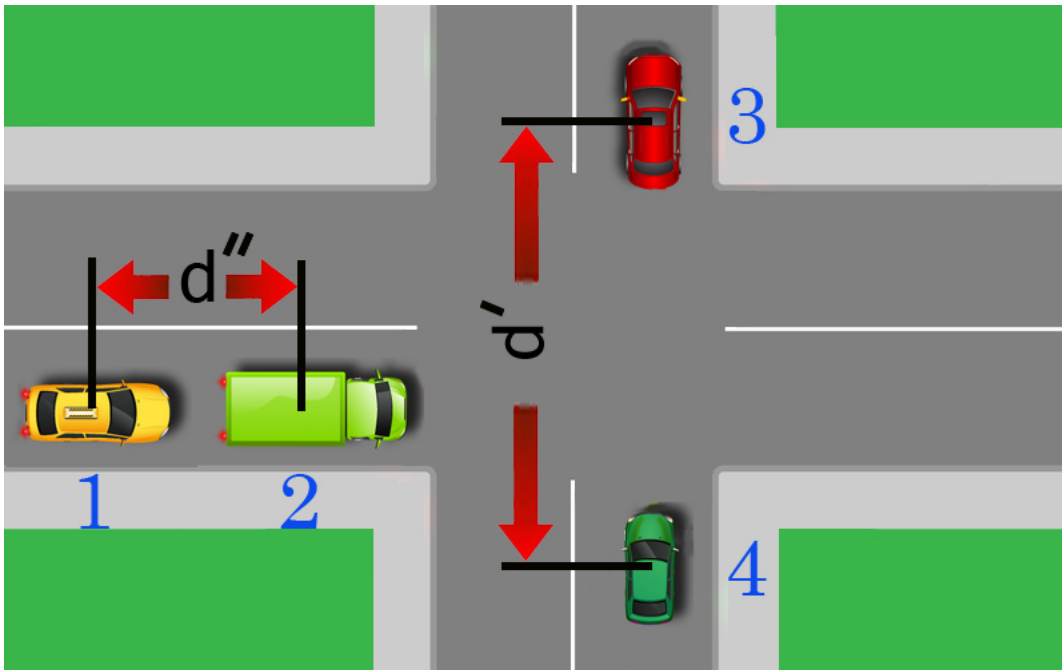

Figure 4.2: Illustration of how distances d' and d'' are measured.

4.3 Mathematical derivations

4.3.1 Cluster number distribution

In this section, we derive the cluster number distribution. We denote by *ClustNum* the number

of clusters in the network. Let us denote by ξ_1 the number of consecutive cars located in the

same line and not separated by intersection that cannot establish a connection with the front car

(as cars 1 and 2 in Figure 4.2). By ξ_2 we denote the number of cars that cannot communicate

with the car in front and located on the opposite sides of the intersection (as cars 3 and 4 in

Figure 4.2). Let us derive the following formula:

$$ClustNum = r + \xi_1 + \xi_2. \tag{4.5}$$

Since cars located on different lines cannot communicate with each other and there are r lines,

the network has at least r clusters. Each unestablished connection adds 1 to the number of

clusters. In total, there are $\xi_1 + \xi_2$ unestablished connections in the network, so the number of

clusters in the network is $r + \xi_1 + \xi_2$. Formula (4.5) is proven.

Let us note that the values ξ_1 and ξ_2 are distributed according to the binomial distribution.

We carry out the proof for ξ_1, the proof for ξ_2 is carried out similarly. We establish the following

formula:

$$P(\xi_1 = i) = \binom{tr}{i}(1 - p')^i p''^{tr-i}. \tag{4.6}$$

In total there are tr cars approaching intersections. The probability that two consecutive vehi-

cles cannot establish a connection equals $(1 - p')$. Therefore, the probability that i of them can-

not establish a connection equals $(1 - p')^i$. The probability that the remaining $tr - i$ vehicles can

establish a connection equals p''^{tr-i}. In addition, there are $\binom{tr}{i}$ ways to select i unsuccessful con-

nections among all tr possible connections. Thus, the resulting probability is $\binom{tr}{i}(1 - p')^i p''^{tr-i}$.

Formula (4.6) is proved. It can be similarly established that

$$P(\xi_2 = i) = \binom{n-r-tr}{i}(1 - p'')^i p''^{n-tr-r-i} \tag{4.7}$$

Now we are ready to obtain the probability $P(ClustNum = i)$ that the network has exactly i clusters. If $ClustNum = i$ and $\xi_1 = j$, then it follows from (4.5) that $\xi_2 = i - r - j$. Thus, the probability $P(ClustNum = i)$ can be calculated using the following formula:

$$P(ClustNum = i) = \sum_j P(\xi_1 = j)P(\xi_2 = i - r - j). \tag{4.8}$$

Substituting (4.6) and (4.7) into (4.8), we get

$$P(ClustNum = i) = \sum_{j=\max(0,i-n+tr)}^{\min(i-r,tr)} \binom{tr}{j}(1 - p')^j p'^{tr-j}\binom{n-r-tr}{i-r-j}(1 - p'')^{i-r-j} p''^{n-tr-i+j}. \tag{4.9}$$

The summation limits in the formula (4.9) are obtained from the conditions of non-negativity of powers of p', $1-p'$, p'' and $1-p''$ (from which we conclude $0 \le j \le tr$ and $i-n+tr \le j \le i-r$).

Now we can derive the average value and variance of the distribution of the number of clusters. Since ξ_1 and ξ_2 have a binomial distribution, we can use the following well-known formulas for their mean values and variances:

$$\mathbf{E}\xi_1 = (1 - p')tr \tag{4.10}$$

$$\mathbf{Var}\xi_1 = p'(1 - p')tr \tag{4.11}$$

$$\mathbf{E}\xi_2 = (1 - p'')(n - r - tr) \tag{4.12}$$

$$\mathbf{Var}\xi_2 = p''(1 - p'')(n - r - tr) \tag{4.13}$$

Since ξ_1 and ξ_2 are independent random variables and (4.5), taking into account (4.10) – (4.13), we obtain the following formulas:

$$\mathbf{E}[ClustNum] = r + (1 - p')tr + (1 - p'')(n - r - tr), \tag{4.14}$$

$$\mathbf{Var}[ClustNum] = p'(1 - p')tr + p''(1 - p'')(n - r - tr). \qquad (4.15)$$

Let us note that the distribution of the number of clusters can be accurately approximated by the normal distribution. From the formulas (4.5), (4.6) and (4.7) it follows that the distribution of the number of clusters is the sum of the constant and two binomial distributions. Since it is well-known that binomial distributions could be approximated by normal distribution, and the sum of two normal distributions is normal, we derive that the distribution of the number of clusters can be approximated by a normal distribution with the mathematical expectation and variance given by the formulas (4.14) and (4.15). In the Simulations section, we confirm that the distribution (4.9) can be accurately approximated to normal distribution for various parameter values.

Formula (4.9) can be made more precise with additional statistical traffic information. At low car densities, such as $\rho = 0.01$, the assumption that at every intersection there is a pair of cars located in the same line and on the opposite sides of the intersection is too strong, which leads to large deviations from the simulation results. Let us assume a known distribution of the number of pairs of such vehicles. We denote the number of pairs of such vehicles by μ (their number cannot exceed tr) and the probability that their number equals t through $P(\mu = t)$. For a better understanding, see Figure 4.2, in this case $\mu = 1$, since cars 3 and 4 are located on opposite sides of the intersection, and cars 1 and 2 are not. To use this new information, it is enough in the formula (4.9) to replace tr with t, multiply the probability by $P(\mu = t)$, and sum over t. The new formula is presented below.

$$P(ClustNum = i) =$$

$$\sum_{t=0}^{tr} P(\mu = t) \sum_{j=\max(0,i-n+tr)}^{\min(i-r,t)} \binom{t}{j}(1 - p')^j p'^{t-j} \binom{n-r-t}{i-r-j}(1 - p'')^{i-r-j} p''^{n-t-i+j}. \quad (4.16)$$

Using (4.14) and (4.15) we can derive formulas for the expectation and variance of the new

distribution.

$$\mathbf{E}[ClustNum] = r + \sum_{t=0}^{tr} P(\mu = t)\{(1 - p')t + (1 - p'')(n - r - t)\} =$$

$$r + \mathbf{E}[\mu](1 - p') + (1 - p'')(n - r) - (1 - p'')\mathbf{E}[\mu] =$$

$$r + \mathbf{E}[\mu](p'' - p') + (1 - p'')(n - r), \quad (4.17)$$

$$\mathbf{Var}[ClustNum] = \sum_{t=0}^{tr} P(NumType2 = t)\{p'(1 - p')t + p''(1 - p'')(n - r - t)\} =$$

$$\mathbf{E}[\mu]p'(1 - p') + p''(1 - p'')(n - r) - p''(1 - p'')\mathbf{E}[NumType2] =$$

$$\mathbf{E}[\mu](p'(1 - p') - p''(1 - p'')) + p''(1 - p'')(n - r). \quad (4.18)$$

4.3.2   Cluster size distribution

In this section, we derive formulas for cluster size distribution. We characterize the cluster size
distribution in different terms than it is done in Chapter 2. Namely, we consider the network at
some point in time and calculate the average cluster size $AvClustSize$. The goal of this section
is to obtain the probability that $AvClustSize$ lies between two fixed integer values. Thus, we
deduce the probability $P(i \leq AvClustSize < i+1)$, where i is a positive integer. Let us establish
the formula

$$P(i \leq AvClustSize < i + 1) = \sum_{\max(r-1,\frac{n}{i+1})<k\leq\frac{n}{i}} P(ClustNum = k), \quad (4.19)$$

where $P(ClustNum = k)$ is given by the formula (4.9). To establish (4.19), we note that if the
network has exactly k clusters, then their average size is n/k. Thus, in order for the average
cluster size to be in the range from i to $i + 1$, it is sufficient that k satisfies the inequalities
$i \leq n/k < i + 1$, from which we get $n/(i + 1) < k \leq n/i$. In addition, k must be no less
than r and no more than the total number of cars n, which gives the limits of summation in

the formula (4.19). To obtain the formula (4.19) it is enough to notice that the probability

$P(i \leq AvClustSize < i + 1)$ equals the sum of the probabilities $P(ClustNum = k)$ within the

given limits. The formula (4.19) is proved.

The average value of the random variable $AvClustSize$ is given by the formula

$$\mathbf{E}[ClustSize] = \sum_{k=r}^{n} \frac{n}{k} P(ClustNum = k). \tag{4.20}$$

The formula (4.20) is valid, since in total the average cluster size n/k is multiplied by the

probability $P(ClustNum = k)$ that the network has exactly k clusters. The following formula

for variance can be similarly derived:

$$\mathbf{Var}[ClustSize] = \sum_{k=r}^{n} \left(\frac{n}{k}\right)^2 P(ClustNum = k) - \mathbf{E}[ClustSize]^2. \tag{4.21}$$

4.4   Intelligent driver model

We use the Intelligent driver model [84] to generate car traffic. According to it, the movement

of vehicle i on a single road is described by the following differential equations:

$$\dot{x}_i = v_i, \tag{4.22}$$

$$\dot{v}_i = a \left(1 - \left(\frac{v_i}{v_0}\right)^\delta - \left(\frac{s^*(v_i, \Delta v_i)}{s_i}\right)^2 \right), \tag{4.23}$$

$$s^*(v_i, \Delta v_i) = s_0 + v_i T + \frac{v_i \Delta v_i}{2 \sqrt{ab}}, \tag{4.24}$$

where x_i and v_i and are coordinate and velocity of i-th car, a is a maximum acceleration, δ is

acceleration exponent, b is a vehicle deceleration, s_0 is a minimum gap between vehicles, v_0 is

a desired speed, T is a time gap between consecutive vehicles, s_i is bumper-to-bumper distance

between car i and next riding car $i - 1$ expressed through the length l_{i-1} of the car $i - 1$ as

follows:

$$s_i = x_{i-1} - x_i - l_{i-1},$$

and Δv_i is speed difference between speed of car i and speed of car $i - 1$

$$\Delta v_i = v_i - v_{i-1}.$$

Typical values of the parameters v_0, T, s_0, δ, a, b are summarised in the table below taken from [85].

Desired speed v_0	54 km/h
Time gap T	1.0 s
Minimum gap s_0	2m
Acceleration exponent δ	4
Acceleration a	1.0m/s^2
Comfortable deceleration b	1.5m/s^2

Unfortunately, the system of equations (4.22) and (4.23) does not allow finding an explicit solution. Therefore, we use the Euler method [6] in order to find approximation to the solution. The essence of this method is that instead of solving the equation on the entire numerical axis, it is solved on a discrete grid with a fixed step. At each step, we numerically solve the differential equation system (4.22), (4.23) for each of the vehicles and update information about their current coordinate and velocity with time step 0.05 second.

4.5 Simulator description

The simulator is developed in Python using set of its modules called PyGame. The simulator map is a several rectangular blocks separated by two-way roads. We consider simulations on large maps with hundreds of vehicles, but the details on them appear quite small. Therefore, we illustrate the main simulator principles on a reduced map depicted in Figure 4.3. The simulator

Figure 4.3: Simulator map.

has the following customizable parameters:

$$sizex, sizey, car\_num, road\_width, block\_num\_x, block\_num\_y,$$

$$min\_interveh\_dist, car\_num, \quad (4.25)$$

where $sizex \times sizey$ is the size of the map, car_num is the number of cars, $block\_num\_x$ and $block\_num\_y$ are the numbers of blocks along the x and y axes, respectively, $min\_interveh\_dist$ is a minimum acceptable distance between vehicles, and car_num is the number of cars. In section Simulations, we specify these and other parameter values. Cars on the map are shown as circles of different colors, their goals are shown in circles of the same color, but with a larger radius. At each intersection there are traffic lights that cyclically change their state according to the diagram in Figure 4.4.

At the beginning of the simulation, cars are distributed on the map randomly. The only
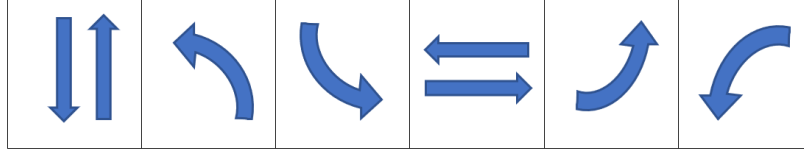
Figure 4.4: Traffic lights states.

requirement is that the distance between any two pairs of cars, as well as between the car and any of the intersections should be more than *min_interveh_dist*. Also at the beginning of the game, the initial coordinates and the destinations of each vehicle are randomly generated. The number of cars on the map is defined as the vehicle density ρ multiplied by the total road length L and rounded down. The vehicles every Δt seconds dynamically change their coordinates according to the Intelligent driver model.

We assume that each vehicle moves towards its goal along the shortest path that is effectively found using the Dijkstra algorithm. This algorithm allows for finding the shortest path in a graph with non-negative weights from a fixed vertex to all other vertices in time $O(ve)$, where v and e are the number of vertices and edges of the graph, respectively. However, this algorithm is not directly applicable to this case, since the next direction of the vehicle depends on its previous state. For example, if the vehicle previously turns at an intersection, then it cannot start turning again. Thus, to correctly calculate the distance, we need to take into account the previous state of the vehicles. This can be implemented using a graph, where each turning point corresponds not to one, but four vertices of the graph corresponding to four possible actions taken before: turning right/left, moving along the road, and continuing moving in the same direction at the intersection.

4.6 Simulation results

We perform simulations for the following parameter values [17]:

$$G_T = G_R = 1, f_c = 5.9 \text{ GHz}, \alpha = 2.5,$$

$$T_0 = 300° \text{ K}, B = 10 \text{ MHz}, \Psi = 10 \text{ dB}.$$

and the motion parameters taken from Section 4.4. We present simulation results in two cases: urban and rural traffic. In the case of rural traffic, we reduce the value of α to 2. The graphs for the vehicle densities $\rho = 0.01, 0.03, 0.05$ (10, 30 and 50 vehicles per kilometer, respectively) are presented below. In the case of urban traffic we assume that the sizes of the blocks coincide with the sizes of the Calgary blocks (100×170 m) depicted in Figure 4.5, and in the rural case the block sizes are the same as in Woodstock, Canada in Figure 4.6 (100×140 m).
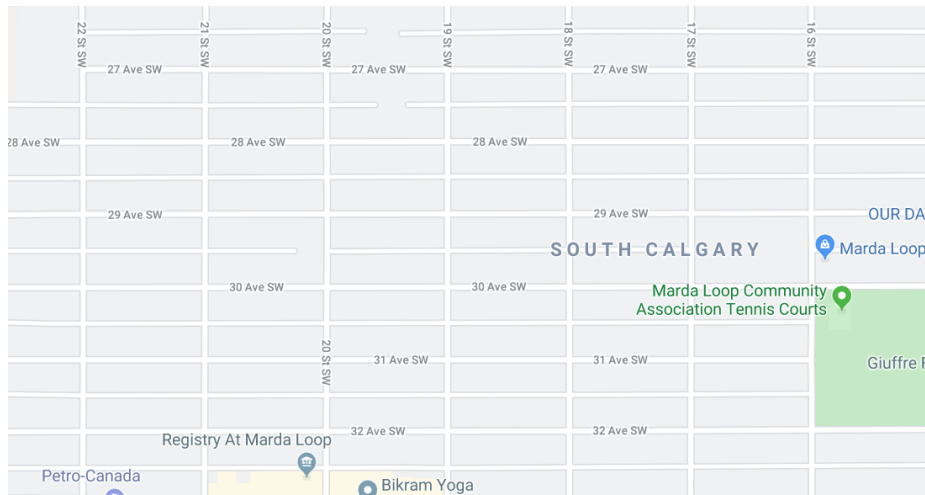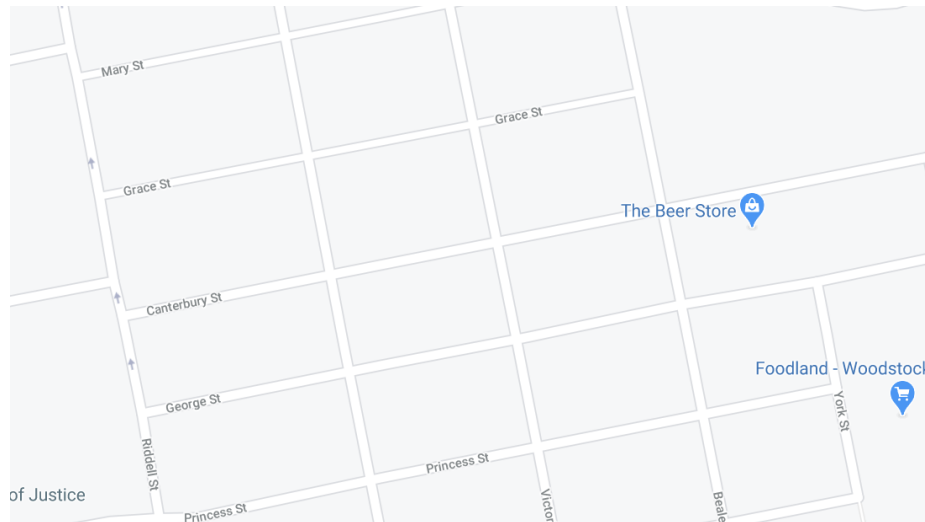

Figure 4.5: Map of Calgary.

Figure 4.6: Map of Woodstock.

We carry out simulations on a map with 16 blocks arranged in a grid 4×4. In this case $tr = 36$ and $r = 16$. Figures 4.7, 4.9 show the simulation results for various vehicle densities.

At the densities $\rho = 0.01, 0.03, 0.05$ the number of cars equals 88, 265, 442 (Calgary) and 78, 236, 394 (Woodstock). The graphs also show approximations of distributions using normal distributions. The advantage of this approach is the simplicity of the formulas and, as the results of the simulations show, a relatively small deviation from the approximate distributions.

We numerically calculate the distribution of the distances d' and d'' (see Figure 4.2). Using the formulas (4.3), (4.4) and numerical integration, we calculate the values of p' and p''. At densities $\rho = 0.01, 0.03, 0.05$ these values are $p' = 0.427, 0.606, 0.656$, $p'' = 0.9974, 0.998$, 0.9986 (Calgary) and $p' = 0.86, 0.93, 0.94$, $p'' = 0.9992, 0.9993, 0.9995$ (Woodstock), respectively.

Figures 4.7 and 4.8 illustrate the distribution of the cluster number in the cases of urban and rural traffic. As it can be seen the distributions are close to normal distributions. It is worth mentioning that the cluster number distribution in Chapter 2 is also close to normal. At a low density of $\rho = 0.01$, the distribution (4.4) gives a rather poor approximation to the simulation results, so we also include the distribution (4.16) graph. In addition, Figures 4.7 and 4.8 show approximations of the distributions (4.4) and (4.3) with normal distributions with mathematic expectation and variance given by the formulas (4.14), (4.15) and (4.17), (4.18). On the graphs, these distributions are indicated as first and second normal distributions. As one can see from (4.14), the average value of the number of clusters grows linearly with increasing vehicle density. This trend can be observed in Figures 4.7 and 4.8, in which the graphs shift to the right with increasing density.

Graphs of the cluster size distribution are presented in Figures 4.9 and 4.10. Since we use a different cluster distribution definition than in Chapter 2, distribution graphs have a different shape. Namely, we consider the distribution of the average value of the cluster size, rather than a size of a randomly selected cluster as in Chapter 2. As one can see, the simulation results are quite well approximated by theoretical results for different vehicle density values.

The theoretical results do not coincide with the results of simulations as in Chapter 2, but they give a fairly good approximation. The reasons that the graphs deviate from the simulation

results include the following:

1. At low density $\rho = 0.01$ the vehicles tend to accumulate close to traffic lights, which greatly distorts the traffic model.

2. At higher densities $\rho = 0.03, 0.05$, at some moments, traffic lights are not able to cope with the flow of cars, which leads to traffic jams and deviations from the theoretical model.

3. The simulator is implemented in such a way that the cars communicate with each other only if they are on the same road. Therefore, during the turn the vehicles cease to communicate with other cars until they turn into another road.

4.7   Chapter summary

This chapter explores the statistical properties of VANET on a 2D map. Traditionally, statistical properties are studied in the case of a highway or an intersection, since a more complex topology introduces additional unknowns into the model such as vehicle routing and traffic light states. This makes the model virtually impossible for accurate analysis. Nevertheless, in this chapter we obtain approximate distributions of the number of clusters and cluster size, which approximate the simulation data quite accurately. To simulate the traffic and communication model, we develop a special simulator with the ability to collect statistical data on the network. As for the communication model, we implement the communication model from Chapter 2, and as the motion model, we use the widely accepted Intelligent Driver model. We compare the simulation results for urban and rural traffic with theoretical predictions. The graphs of theoretical distributions quite accurately approximate the simulation results done for different vehicle densities on the map in the cases of urban and rural traffic.

(a) $\rho = 0.01$

(b) $\rho = 0.03$

(c) $\rho = 0.05$

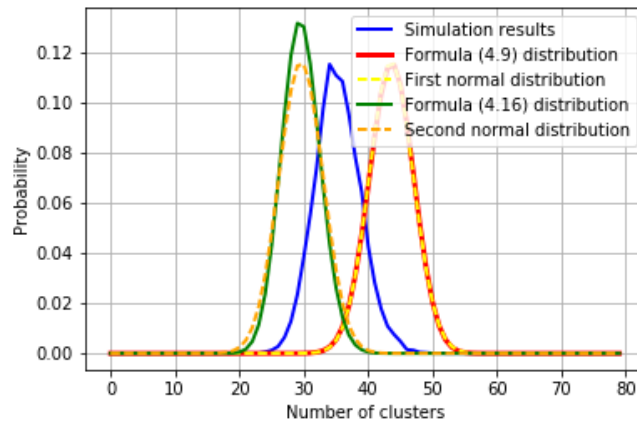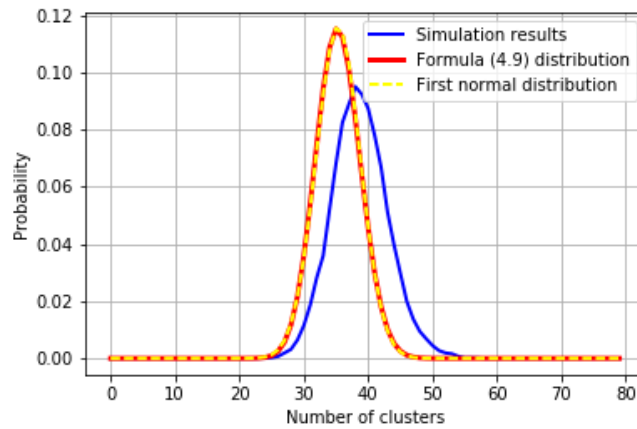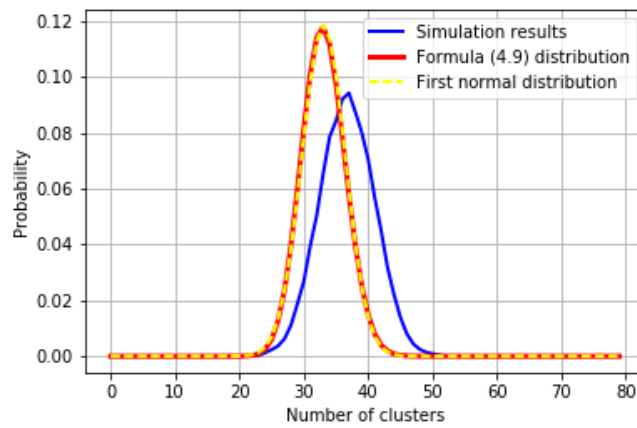Figure 4.7: Cluster number distribution (urban traffic).
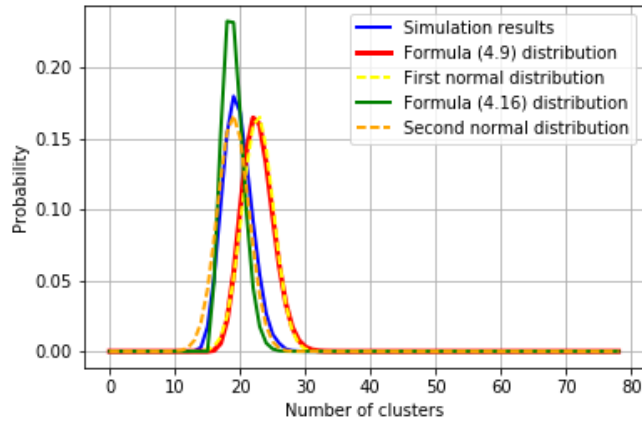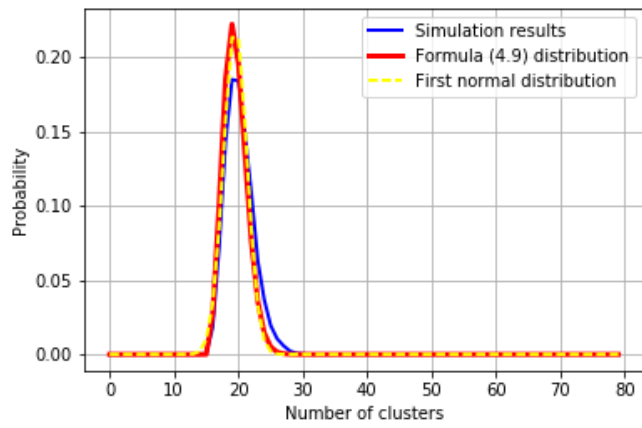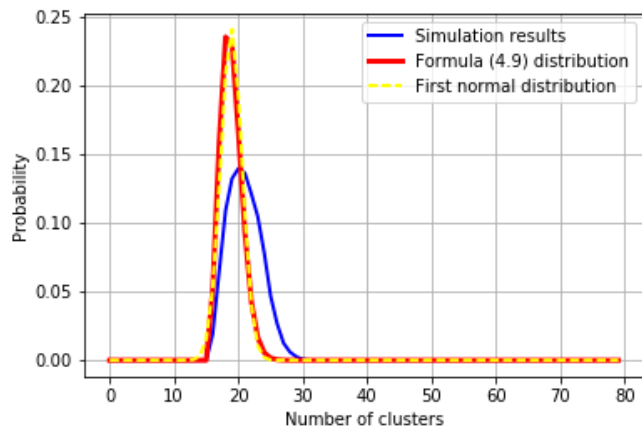
(a) $\rho = 0.01$

(b) $\rho = 0.03$

(c) $\rho = 0.05$

Figure 4.8: Cluster number distribution (rural traffic).
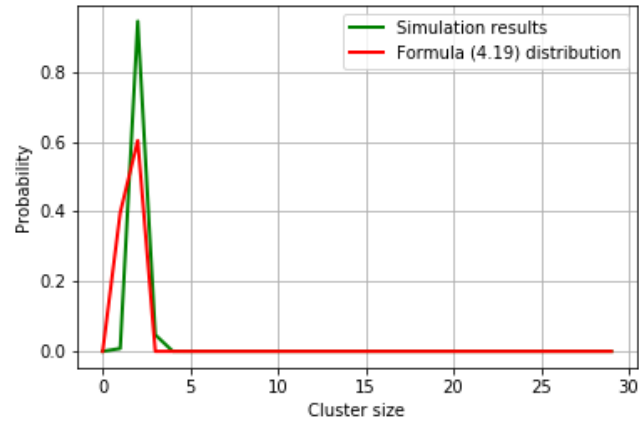
(a) $\rho = 0.01$
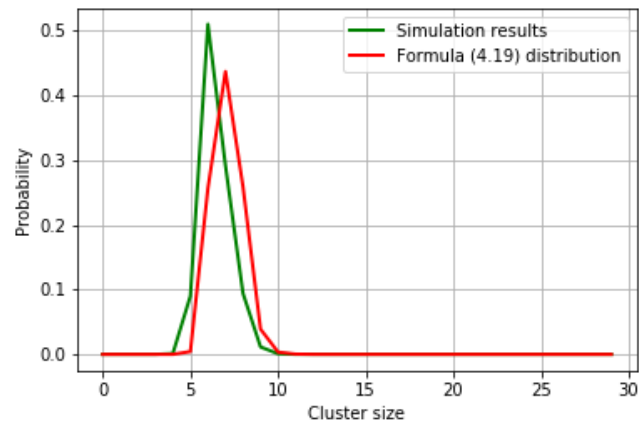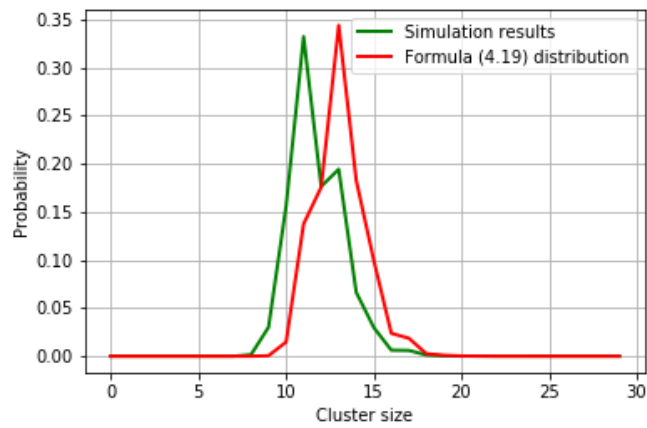
(b) $\rho = 0.03$

(c) $\rho = 0.05$

Figure 4.9: Cluster size distribution (urban traffic).
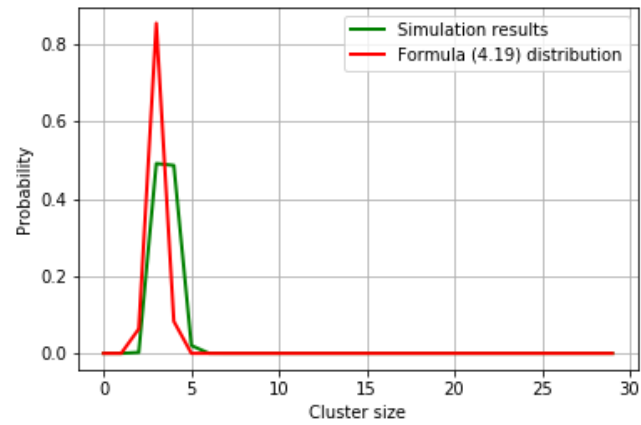
(a) $\rho = 0.01$
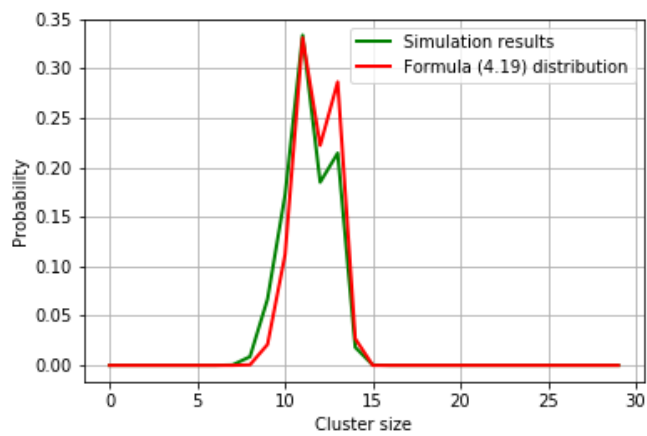
(b) $\rho = 0.03$
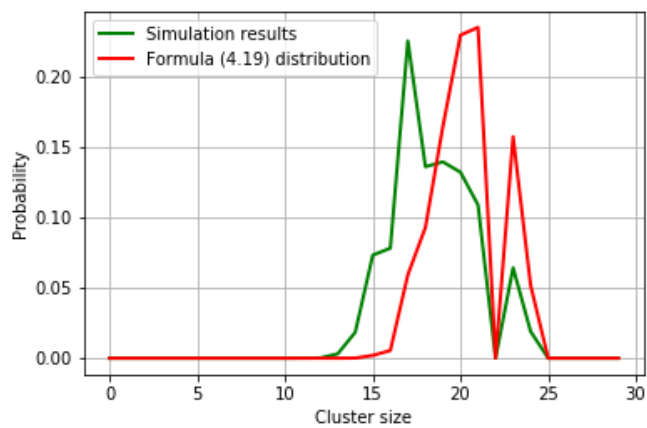
(c) $\rho = 0.05$

Figure 4.10: Cluster size distribution (rural traffic).

Chapter 5

Jamming and anti-jamming strategies of mobile vehicles

5.1 Introduction

One or more hostile devices may exist in a cognitive radio network, the purpose of which is to break or completely disrupt the communication between devices. One of the most common types of such attacks is the so-called Denial of Service (DoS) Attack [12, 25, 90, 103, 111]. The essence of this attack is that the attacker tries to disrupt the network by sending numerous fake or repeated signals. To combat jammer attacks, network devices can change transmission channels, increase transmission power, or even change their location to avoid being in close proximity to the jammer. Typically, the literature uses a description of an anti-jamming game in terms of game theory [58]. This theory is a powerful tool for finding the optimal device strategy. Within the framework of this theory, it is assumed that one or more cooperating network devices maximizes its utility function, while each of the jammers maximizes its own. However, the game theory approach cannot always be used explicitly in practice, since the network parameters are usually unknown, therefore, over the past decade, finding the optimal solution implicitly using machine learning algorithms [2] has become popular in research.

These algorithms, through trial and error, constantly improve their strategy. One of the most popular algorithms is Q-learning [55] and its modifications due to their fast convergence and simplicity of implementation. We discuss below several articles that use game theory and machine learning algorithms for finding optimal strategies in the anti-jamming game in various settings.

In [30], the authors examine a case in which the jammer attempts to disrupt the communication between two consecutive vehicles in a platoon. Since data is transmitted over one channel, both the jammer and the vehicles adapt their transmission power in order to maximize utility functions. The authors use a modification of the Q-learning algorithm called Dyna-Q and compare the learning outcomes with the classic Q-learning algorithm. The results convincingly show that Dyna-Q converges to the same strategy, but has a higher convergence rate. The articles [52, 53, 102] discuss the VANET anti-jamming game with drones. The essence of the game is that in the case of a jammer attack, the drone replays the data, sent by VANET nodes in order to increase SINR and reduce BER. Thus, at every time moment, the drone decides whether to send data or not. The authors deduce the Nash equilibrium [8] and compare it with results iteratively found by Q-learning and its effective modification called Policy Hill Climbing.

The case of a cooperative game between devices is also popular in the literature. The difference between this case and single agent games is that instead of having independent utility functions, devices have a common utility function that evaluates the network state as a whole. Such games are more difficult to consider, since in this case there is a large number of system states that grow exponentially with the network device number growth. In [5, 105], the authors examine an anti-jamming game with several transmitting cooperating devices. Q-learning algorithm for finding a common optimal strategy in this formulation shows its advantage over the non-cognitive sub-band selection policy. In [110], a game with cooperating devices and one jammer is considered. The authors propose an iterative algorithm for finding a cooperative strategy and compare the results with random anti-jamming and selfish anti-jamming algo-

rithms. The simulation results show that the iterative algorithm achieves higher throughput and better performance than non-cooperative algorithms.

In this chapter, we consider the VANET anti-jamming game, however, instead of the static model as in the previous articles, we use our simulator as described in Chapter 4. All simulations are carried out in the case of straight road. We assume that the two communicating vehicles are pursued by the jammer interrupting their ongoing communication. We consider two cases: single-channel and multi-channel game. In the multi-channel case, it is assumed that the vehicles change channels according to a predetermined pseudo-random sequence. In this situation, we presume that the jammer shares its power between channels because it cannot predict the next state of the network in advance. To confirm the simulation results we formulate and prove theorems that describe the Nash equilibrium of the game, which can be interpreted as the optimal strategy for the vehicle and the jammer. At first we suggest that power included in the vehicle and jammer utility functions is linear, but from the Theorem it follows that the optimal strategy of the vehicles is to transmit with maximum power. Therefore, we change the classic formula of the utility function in order to find a non-trivial vehicle strategy. To do this, we consider the quadratic power function in both vehicle and jammer utility functions. Such a power function is closer to practical implementation, since transmitting on higher power levels requires a greater expenditure of system resources than at low ones. Under this assumption, we formulate and prove the Nash equilibrium theorems in both single-channel and multi-channel cases. Next, we examine several machine learning algorithms such as Policy Hill Climbing, Deep Q-learning, Quelling Q-learning, and Quelling Deep Q-learning. All the algorithms successfully converge to the theoretically derived Nash equilibrium.

Chapter 5 is organized as follows. In Section 5.2, the anti-jamming game is described in terms of game theory. In Sections 5.3, 5.4 and 5.5, the necessary and sufficient conditions of the Nash equilibrium are established in the single-channel and multi-channel cases. Section 5.6 describes machine learning algorithms that are used in Section 5.7 in order to find the optimal vehicle and jammer strategies.

5.2   Game description

In this section we consider the anti-jamming game on a single road. We assume that two communicating cars are chased by the jammer. Signal-to-interference-plus-noise ratio (SINR) of the vehicle is given by the formula

$$SINR = \frac{h_{car}^2 x}{\sigma^2 + h_J^2 y},$$

where σ^2 is a noise power, h_{car}^2 and h_J^2 are vehicle and jammer channel power gains, x and y are powers of the car and the jammer, respectively.

Figure 5.1: Illustration of anti-jamming game.

We assume that the utility functions of the vehicle u_{car} and the jammer u_J are calculated by the formulas taken from [101]

$$u_{car} = \frac{h_{car}^2 x}{\sigma^2 + h_J^2 y} - C_{car} x, \tag{5.1}$$

$$u_J = -\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y} + C_{car} x - C_J y, \tag{5.2}$$

where C_{car} and C_J are positive transmission costs of the vehicle and the jammer. The goal of each player is to maximize their utility functions. In the future, we consider other utility

functions with quadratic power function, but in the beginning we focus on this simplest model.

We assume that after the signal is transmitted, the vehicle receives back the signal-to-interference-plus-noise ratio (SINR) value and, based on this information, makes a decision on increasing/decreasing the transmission power. By *PmaxC* and *PmaxJ* we denote maximum vehicle and jammer transmission power, respectively.

5.3 Nash equilibrium in the case of the linear cost function

By definition, Nash equilibrium is a strategy (x^*, y^*), which satisfies the following two inequalities:

$$u_{car}(x^*, y^*) \geq u_{car}(x, y^*),$$ (5.3)

$$u_J(x^*, y^*) \geq u_J(x^*, y).$$ (5.4)

We first consider the case when the system parameters satisfy the inequality

$$C_{car} \geq \frac{h_{car}^2}{\sigma^2}.$$

In this case, the point $(0, 0)$ is a Nash equilibrium, because

$$u_{car}(x, 0) = x\left(\frac{h_{car}^2}{\sigma^2} - C_{car}\right) \leq 0 = u_{car}(0, 0),$$

$$u_J(0, y) = -C_J y \leq 0 = u_{car}(0, 0).$$

Thus, in this case, the optimal behavior of the vehicle and the jammer is to transmit zero power due to the high transmission cost C_{car}. To ensure that this unrealistic case does not take place, we assume that

$$C_{car} < \frac{h_{car}^2}{\sigma^2}.$$ (5.5)

Theorem. If the inequality (5.5) holds, then the Nash equilibrium is reached at the point (x^*, y^*) that can be expressed in terms of the parameter

$$\hat{y} = \frac{-\sigma^2}{h_J^2} + \frac{h_{car}}{h_J} \sqrt{\frac{PmaxJ}{C_J}}$$

as follows:

$$x^* = PmaxC,$$

$$y^* = \begin{cases} \hat{y}, & \text{if } 0 \leq \hat{y} \leq PmaxJ, \\ 0, & \text{if } \hat{y} < 0, \\ PmaxJ, & \text{if } \hat{y} > PmaxJ. \end{cases} \tag{5.6}$$

Proof. The function u_{car} is linear with respect to x, therefore, if we fix value of y, it reaches a maximum at the ends of the segment $[0, PmaxC]$. Let us assume that the maximum is reached at the point $x^* = 0$. Since the function $u_J(x^*, y) = u_J(0, y) = -C_J y$ reaches its maximum at the point $y = 0$ we conclude that $y^* = 0$. However, it is impossible for the point $(0, 0)$ to be a Nash equilibrium, since if $y^* = 0$ and the inequality (5.5) holds, then the function $u_{car}(x, 0)$ reaches its maximum at $x^* = Pmax$, and not at $x^* = 0$. Thus, $x^* = PmaxC$.

Let us derive value of y^*. The derivative of utility function u_{car} is given as follows:

$$\frac{d}{dy} u_J = \frac{d}{dy} \left(-\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y} + C_{car} x - C_J y \right) = \frac{h_{car}^2 h_J^2 x}{(\sigma^2 + h_J^2 y)^2} - C_J. \tag{5.7}$$

Solving the equation $\frac{d}{dy} u_J = 0$, we find its root $\hat{y}$

$$\hat{y} = \frac{-\sigma^2}{h_J^2} + \frac{h_{car}}{h_J} \sqrt{\frac{x}{C_J}}. \tag{5.8}$$

Since $\frac{d^2}{dy^2} u_J < 0$, the function u_J is convex downward and its maximum value can be reached at the point $\hat{y}$ or at the ends of the segment 0 and $PmaxJ$. Taking into account that the function u_J increases with $y < \hat{y}$ and decreases with $y > \hat{y}$, we conclude that the maximum point y^* can

be calculated using the formula (5.6).

5.4 Nash equilibrium in the case of the quadratic cost function

In the previous section, we examine the case in which the power increases linearly in car and jammer utility functions. This assumption leads to a situation in which the optimal strategy for the car (see Theorem) is to transmit a maximum power. In this section, we change the utility function, which leads to a more realistic model. Namely, we assume that the power term in the utility function does not grow linearly, but quadratically. The quadratic power term is more suitable, since in this case the high transmission power leads to the high transmission cost. The utility functions for car and jammer are now described by the following equations:

$$u_{car} = \frac{h_{car}^2 x}{\sigma^2 + h_J^2 y} - C_{car} x^2, \tag{5.9}$$

$$u_J = -\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y} + C_{car} x^2 - C_J y^2. \tag{5.10}$$

We assume that the vehicle and jammer powers x and y satisfy the following requirements:

$$0 < x \le PmaxC, 0 < y \le PmaxJ. \tag{5.11}$$

Theorem 2. In the game with utility functions given by (5.9) and (5.10) in the area (5.11) the point (x^*, y^*) is Nash equilibrium if and only if one of the following conditions is satisfied.

1. $0 < x^* < PmaxC$, $0 < y^* < PmaxJ$, y^* is a solution of the equation

$$y^*(\sigma^2 + h_J^2 y^*)^3 = \frac{h_{car}^4 h_J^2}{4C_{car}C_J}, \tag{5.12}$$

and x^* is expressed in terms of y^* as follows:

$$x^* = \frac{h_{car}^2}{2C_{car}(\sigma^2 + h_J^2 y^*)}.$$

(5.13)

2. $x^* = PmaxC$, $0 < y < PmaxJ$, y^* is a solution of the equation

$$y^*(\sigma^2 + h_J^2 y^*)^2 = \frac{h_{car}^2 h_J^2 PmaxC}{2C_J}$$

(5.14)

and

$$\frac{h_{car}^2}{\sigma^2 + h_J^2 y^*} - 2C_{car} PmaxC \geq 0.$$

(5.15)

3. $0 < x^* < PmaxC$, $y^* = PmaxJ$, and

$$x^* = \frac{h_{car}^2}{2C_{car}(\sigma^2 + h_J^2 PmaxJ)},$$

(5.16)

$$\frac{h_{car}^2 h_J^2 x^*}{(\sigma^2 + h_J^2 PmaxJ)^2} - 2C_J PmaxJ \geq 0.$$

(5.17)

4. $x^* = PmaxC$, $y^* = PmaxJ$, and

$$\frac{h_{car}^2}{\sigma^2 + h_J^2 PmaxJ} - 2C_{car} PmaxC \geq 0,$$

(5.18)

$$\frac{h_{car}^2 h_J^2 PmaxC}{(\sigma^2 + h_J^2 PmaxJ)^2} - 2C_J PmaxJ \geq 0.$$

(5.19)

Proof. The derivative of the utility functions u_{car} and u_J are given as follows:

$$\frac{d}{dx} u_{car} = \frac{h_{car}^2}{\sigma^2 + h_J^2 y} - 2C_{car} x,$$

(5.20)

$$\frac{d}{dy} u_J = \frac{d}{dy}\left(-\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y} + C_{car} x^2 - C_J y^2 \right) = \frac{h_{car}^2 h_J^2 x}{(\sigma^2 + h_J^2 y)^2} - 2C_J y.$$

(5.21)

Let us calculate the derivative $\frac{d^2}{dy^2}u_J(x, y)$ taking into account (5.21).

$$\frac{d^2u_J(x, y)}{dy^2} = -\frac{2h_{car}^2 h_J^4 x}{(\sigma^2 + h_J^2 y)^3} - 2C_J. \tag{5.22}$$

Since all the terms in (5.22) are negative, we conclude that

$$\frac{d^2}{dy^2}u_J(x, y) < 0. \tag{5.23}$$

By analogy, one can derive that

$$\frac{d^2}{dx^2}u_{car}(x, y) < 0. \tag{5.24}$$

Case 1. Let us consider the case in which Nash equilibrium is reached at the point (x^*, y^*), located inside the area $0 < x < PmaxC$, $0 < y < PmaxJ$. From (5.23) and (5.24) one can conclude that the functions $u_{car}(x, y)$ and $u_J(x, y)$ are convex upward as the functions of x and y, respectively. Therefore, the conditions

$$\frac{d}{dx}u_{car}(x^*, y^*) = 0, \tag{5.25}$$

$$\frac{d}{dy}u_J(x^*, y^*) = 0 \tag{5.26}$$

are equivalent to the conditions (5.3), (5.4). Equations (5.25) and (5.26) can be rewritten as follows:

$$\sigma^2 + h_J^2 y^* = \frac{h_{car}^2}{2C_{car}x^*}, \tag{5.27}$$

$$(\sigma^2 + h_J^2 y^*)^2 = \frac{h_{car}^2 h_J^2 x^*}{2C_J y^*}. \tag{5.28}$$

Multiplying the equalities (5.27) and (5.28) we can derive the following equation of the fourth degree with respect to y^*:

$$y^*(\sigma^2 + h_J^2 y^*)^3 = \frac{h_{car}^4 h_J^2}{4C_{car}C_J}.$$

Let us note that the right-hand side is an increasing function; therefore, it has at most one root in the region $0 < y < PmaxJ$. The value of x^* can be obtained from (5.27) as follows:

$$x^* = \frac{h_{car}^2}{2C_{car}(\sigma^2 + h_J^2 y^*)}.$$

We consider cases in which the Nash equilibrium lies on the border of the region. Since $x > 0$ and $y > 0$, only cases $(x = PmaxC, 0 < y < PmaxJ)$, $(0 < x < PmaxC, y = PmaxJ)$, and $(y = PmaxJ$ and $y = PmaxJ)$ are possible. Below we consider them all.

Case 2. In the case $x^* = PmaxC$, $0 < y < PmaxJ$ let us establish that the conditions $\frac{d}{dy}u_J(PmaxC, y^*) = 0$ and $\frac{d}{dx}u_{car}(PmaxC, y^*) \geq 0$ are necessary and sufficient for point (x^*, y^*) to be Nash equilibrium.

We prove that fulfillment of the condition $\frac{d}{dx}u_{car}(PmaxC, y^*) \geq 0$ ensures that the condition (5.3) is satisfied. From (5.23) we conclude that the derivative $\frac{d}{dx}u_{car}(x, y^*)$ decreases. If $\frac{d}{dx}u_{car}(x, y^*)$ is non-negative at the point $x = PmaxC$, then it is non-negative over the entire interval $0 < x \leq PmaxC$. Thus, in this interval, the function $u_{car}(x, y^*)$ is non-decreasing and condition (5.3) is satisfied.

Let us establish that $\frac{d}{dy}u_J(PmaxC, y^*) = 0$ guarantees that (5.4) holds. Since conditions $\frac{d^2}{dy^2}u_J(x, y) < 0$ and $\frac{d}{dy}u_J(PmaxC, y^*) = 0$ hold we derive that y^* is a maximum of the function $u_J(PmaxC, y)$. Therefore, (5.4) is satisfied.

Carrying out the reverse reasoning, we can verify that the conditions $\frac{d}{dy}u_J(PmaxC, y^*) = 0$ and $\frac{d}{dx}u_{car}(PmaxC, y^*) \geq 0$ are also sufficient to satisfy (5.3) and (5.4). These conditions can be rewritten as follows:

$$y^*(\sigma^2 + h_J^2 y^*)^2 = \frac{h_{car}^2 h_J^2 PmaxC}{2C_J},$$

$$\frac{d}{dx}u_{car}(PmaxC, y^*) = \frac{h_{car}^2}{\sigma^2 + h_J^2 y^*} - 2C_{car}PmaxC \geq 0.$$

Case 3. Case $0 < x^* < PmaxC$, $y^* = PmaxJ$ can be analyzed similarly to case 2. By analogy with case 2, one can prove that is this case conditions $\frac{d}{dx}u_{car}(x^*, PmaxJ) = 0$ and

$\frac{d}{dy}u_J(x^*, PmaxJ) \geq 0$ must be satisfied, therefore,

$$x^* = \frac{h_{car}^2}{2C_{car}(\sigma^2 + h_J^2 PmaxJ)},$$

$$\frac{d}{dy}u_J(x^*, PmaxJ) = \frac{h_{car}^2 h_J^2 x^*}{(\sigma^2 + h_J^2 PmaxJ)^2} - 2C_J PmaxJ \geq 0.$$

Case 4. In the case $x^* = PmaxC, y^* = PmaxJ$ (by analogy with case 2) it could be proven that in order for the equalities (5.3) and (5.4) to be satisfied, it is necessary and sufficient that $\frac{d}{dx}u_{car}(PmaxC, PmaxJ) \geq 0$ and $\frac{d}{dy}u_J(PmaxC, PmaxJ) \geq 0$ are satisfied. Thus, the inequalities

$$\frac{d}{dx}u_{car}(PmaxC, PmaxJ) = \frac{h_{car}^2}{\sigma^2 + h_J^2 PmaxJ} - 2C_{car}PmaxC \geq 0,$$

$$\frac{d}{dy}u_J(PmaxC, PmaxJ) = \frac{h_{car}^2 h_J^2 PmaxC}{(\sigma^2 + h_J^2 PmaxJ)^2} - 2C_J PmaxJ \geq 0$$

hold.

Remark. Equations (5.12) and (5.14) have increasing right side, therefore, they could be efficiently solved by using binary search on the interval $[0, PmaxJ]$.

5.5 Nash equilibrium in the case of multi-channel game

In the case of a multi-channel game with m channels, we assume that the vehicle selects the next channel according to a predetermined pseudorandom sequence, and all channels are chosen equally probable. We assume that the jammer divides its power between m channels. Thus, the jammer state is the vector $(y_1, y_2, \ldots, y_m)$, where y_i is the power transmitted through the channel i. Since the vehicle chooses a certain channel with probability $1/m$ and the vehicle utility function on this channel is given by the formula (5.9), we conclude that the average value of the received reward is calculated by the formula

$$u_{car} = \sum_{k=1}^{m} \frac{1}{m}\left(\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y_k} - C_{car}x^2\right). \tag{5.29}$$

We assume that the jammer utility function is given by the following formula:

$$u_J = -u_{car} - C_J \sum_{k=1}^{m} y_k^2 = \sum_{k=1}^{m} \frac{1}{m}\left(-\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y_k} + C_{car}x^2\right) - C_J \sum_{k=1}^{m} y_k^2. \tag{5.30}$$

The sum of the transmitted jammer powers y_k over all k channels must not exceed the maximum jammer power *PmaxJ* and should be positive

$$\sum_k y_k \leq PmaxJ, \tag{5.31}$$

$$y_k > 0. \tag{5.32}$$

In addition, we assume that the car power x lies in the following range:

$$0 < x \leq PmaxC. \tag{5.33}$$

The point $(x^*, y_1^*, y_2^*, \ldots, y_m^*)$ is called Nash equilibrium if the following inequalities:

$$u_{car}(x^*, y_1^*, y_2^*, \ldots, y_m^*) \geq u_{car}(x, y_1^*, y_2^*, \ldots, y_m^*), \tag{5.34}$$

$$u_J(x^*, y_1^*, y_2^*, \ldots, y_m^*) \geq u_J(x^*, y_1, y_2, \ldots, y_m) \tag{5.35}$$

hold.

Theorem 3. Nash equilibrium $(x^*, y_1^*, y_2^*, \ldots, y_m^*)$ exists in the game with cost functions (5.29), (5.30) in the region given by the inequalities (5.31), (5.32), (5.33) if and only if

$$y_1^* = y_2^* = \ldots = y_m^* = y^* \tag{5.36}$$

and one of the following conditions is satisfied.

1. $0 < x^* < PmaxC$, $y^* < \frac{PmaxJ}{m}$. In this case, y^* can be found from the equation

$$y^*(\sigma^2 + h_J^2 y^*)^3 = \frac{h_{car}^4 h_J^2}{4mC_{car}C_J}.$$

(5.37)

The value of x^* is expressed in terms of y^* according to the formula

$$x^* = \frac{h_{car}^2}{2C_{car}(\sigma^2 + h_J^2 y^*)}.$$

(5.38)

2. $x^* = PmaxC$, $y^* < \frac{PmaxJ}{m}$. In this case y^* is a solution to the equation

$$y^*(\sigma^2 + h_J^2 y^*)^2 = \frac{h_{car}^2 h_J^2 PmaxC}{2mC_J}$$

(5.39)

and the inequality

$$\frac{h_{car}^2}{\sigma^2 + h_J^2 y^*} - 2C_{car}PmaxC \geq 0$$

(5.40)

holds.

3. $0 < x^* < PmaxC$, $y^* = \frac{PmaxJ}{m}$. The value x^* is given by the formula

$$x^* = \frac{h_{car}^2}{2C_{car}(\sigma^2 + h_J^2 \frac{PmaxJ}{m})},$$

(5.41)

and the following inequality:

$$\frac{h_{car}^2 h_J^2 x^*}{(\sigma^2 + h_J^2 \frac{PmaxJ}{m})^2} - 2C_J PmaxJ \geq 0$$

(5.42)

is satisfied.

4. $x^* = PmaxC$, $y^* = \frac{PmaxJ}{m}$. In this case, the following conditions must be satisfied:

$$\frac{h_{car}^2}{\sigma^2 + h_J^2 \frac{PmaxJ}{m}} - 2C_{car}PmaxC \geq 0,$$

(5.43)

$$\frac{h_{car}^2 h_J^2 PmaxC}{(\sigma^2 + h_J^2 \frac{PmaxJ}{m})^2} - 2C_J PmaxJ \geq 0. \tag{5.44}$$

Proof. The proof of this theorem consists of two parts. We first establish the equality $y_1^* = y_2^* = \ldots = y_m^*$. Denoting these variables by y^*, we reduce the problem to finding Nash equilibrium in the two-dimensional case. This problem is similar to the problem considered in the previous section, and it can be solved in the same way.

First, we obtain the derivatives $\frac{d}{dx} u_{car}$ and $\frac{d^2}{dx^2} u_{car}$

$$\frac{d}{dx} u_{car} = \sum_{k=1}^{m} \frac{d}{dx} \frac{1}{m} \left(\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y_k} - C_{car} x^2 \right) = \sum_{k=1}^{m} \frac{1}{m} \left(\frac{h_{car}^2}{\sigma^2 + h_J^2 y_k} - 2C_{car} x \right), \tag{5.45}$$

$$\frac{d^2}{dx^2} u_{car} = -2C_{car} < 0. \tag{5.46}$$

Let us consider the case $0 < x^* < PmaxC$ first. From (5.46) we derive that the function $u_{car}(x, y_1^*, y_2^*, \ldots, y_m^*)$ is convex upward with respect to the variable x. Inequality (5.34) means that the point x^* is a maximum of the function $u_{car}(x, y_1^*, y_2^*, \ldots, y_m^*)$. Therefore, to fulfill the inequality (5.34) it is necessary and sufficient that $\frac{d}{dx} u_{car}(x^*, y_1^*, y_2^*, \ldots, y_m^*) = 0$. Solving this equation, we find that

$$x^* = \frac{1}{2mC_{car}} \sum_{k=1}^{m} \frac{h_{car}^2}{\sigma^2 + h_J^2 y_k^*} \tag{5.47}$$

Let us note that (5.35) is equivalent to the point $(y_1^*, y_2^*, \ldots, y_m^*)$ being the maximum of the function $u_J(x^*, y_1, y_2, \ldots, y_m)$. To find this maximum with restrictions (5.31), (5.32), and (5.47) we consider the Lagrangian

$$L(x, y) = u_J - \lambda \left(\sum_k y_k - PmaxJ \right) - \mu \left(\sum_{k=1}^{m} \frac{h_{car}^2}{\sigma^2 + h_J^2 y_k} - 2mC_{car} x \right),$$

where $\lambda \geq 0$. The derivative $\frac{d}{dy_l}L$ is calculated by the formula

$$\frac{d}{dy_l}L = \frac{d}{dy_l}u_J - \lambda + \mu\frac{h_{car}^2 h_J^2}{\left(\sigma^2 + h_J^2 y_l\right)^2} =$$

$$\sum_{k=1}^{m}\frac{1}{m}\frac{d}{dy_l}\left(-\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y_k} + C_{car}x^2\right) - 2C_J y_l - \lambda + \mu\frac{h_{car}^2 h_J^2}{\left(\sigma^2 + h_J^2 y_l\right)^2} =$$

$$\frac{h_{car}^2 h_J^2 x}{m(\sigma^2 + h_J^2 y_l)^2} - 2C_J y_l - \lambda + \mu\frac{h_{car}^2 h_J^2}{\left(\sigma^2 + h_J^2 y_l\right)^2}. \quad (5.48)$$

Solving the equation $\frac{d}{dy_l}L(x^*, y_1^*, y_2^*, \ldots, y_m^*) = 0$, we obtain

$$h_{car}^2 h_J^2 x^* + \mu m h_{car}^2 h_J^2 = m(2C_J y_l^* + \lambda)(\sigma^2 + h_J^2 y_l^*)^2. \quad (5.49)$$

Let us note that the equation (5.49) with respect to y_l has no more than one root, since the left side is a constant, and the right side is a monotonically increasing function (because $\lambda \geq 0$). Let us denote this root by y^*. Since the equation (5.49) is satisfied for every l, we conclude that

$$y_1^* = y_2^* = \ldots = y_m^* = y^*. \quad (5.50)$$

The case $x^* = PmaxC$ can be analyzed in a similar way and also leads to the formula (5.50). Thus, regardless of where x^* is located on the interval $(0, PmaxC]$, the equalities (5.50) are fulfilled.

Since (5.50) holds, it is convenient for us to consider the functions $\tilde{u}_{car}(x, y)$ and $\tilde{u}_J(x, y)$ given as follows:

$$\tilde{u}_{car}(x, y) = u_{car}(x, y, y, \ldots, y) = \frac{h_{car}^2 x}{\sigma^2 + h_J^2 y} - C_{car}x^2, \quad (5.51)$$

$$\tilde{u}_J(x, y) = u_J(x, y, y, \ldots, y) = -\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y} + C_{car}x^2 - mC_J y^2. \quad (5.52)$$

Let us establish that the function $u_J(x^*, y_1, y_2, \ldots, y_m)$ is a convex upwards function with

regards to the variables $y_1, y_2, \ldots, y_m$. We rewrite the function $u_J(x^*, y_1, y_2, \ldots, y_m)$ in the following form:

$$u_J = \sum_{k=1}^{m} \frac{1}{m} \left(-\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y_k} + C_{car} x^2 - mC_J y_k^2 \right).$$

By proving that the second derivative is negative, it can be established that each term $-\frac{h_{car}^2 x}{\sigma^2 + h_J^2 y_k} +$ $C_{car} x^2 - mC_J y_k^2$ is a convex upward function of the argument y_k. Therefore, each term is a convex upward function of the arguments $y_1, y_2, \ldots, y_m$. Thus, the function $u_J(x^*, y_1, y_2, \ldots, y_m)$ is a convex upward function of the arguments $y_1, y_2, \ldots, y_m$, since it equals to the sum of convex upward functions. Similarly, we can establish that the function $u_J(x, y_1^*, y_2^*, \ldots, y_m^*)$ is a convex upward function of the argument x.

From the convexity of the functions discussed in the previous paragraph we can conclude that the fulfillment of inequalities (5.34), (5.35) is equivalent to the fulfillment of the following inequalities:

$$\tilde{u}_{car}(x^*, y^*) \geq \tilde{u}_{car}(x, y^*), \tag{5.53}$$

$$\tilde{u}_J(x^*, y^*) \geq \tilde{u}_J(x^*, y). \tag{5.54}$$

Thus, we can reduce the multidimensional problem of finding Nash equilibrium to the two-dimensional case. From (5.31) and (5.50) we derive

$$0 < y^* \leq \frac{PmaxJ}{m}.$$

Therefore, there are 4 different cases we need to consider: $(0 < x^* < PmaxC, 0 < y^* < \frac{PmaxJ}{m})$, $(x^* = PmaxC, 0 < y^* < \frac{PmaxJ}{m})$, $(0 < x^* < PmaxC, y^* = \frac{PmaxJ}{m})$, and $(x^* = PmaxC, y^* = \frac{PmaxJ}{m})$.

Case 1. $0 < x^* < PmaxC, 0 < y^* < \frac{PmaxJ}{m}$. From the equations (5.47), and (5.50) we deduce that

$$x^* = \frac{h_{car}^2}{2C_{car}(\sigma^2 + h_J^2 y^*)}. \tag{5.55}$$

Let us notice that (5.54) is equivalent to the point y^* being the maximum of the function

$\tilde{u}_J(x^*, y)$. Taking into account that $\tilde{u}_J$ is convex upward, we conclude that $\frac{d}{dy}\tilde{u}_J(x^*, y^*) = 0$. The derivative $\frac{d}{dy}\tilde{u}_J(x, y)$ is given by the following formula:

$$\frac{d}{dy}\tilde{u}_J(x, y) = \frac{h_{car}^2 h_J^2 x}{(\sigma^2 + h_J^2 y)^2} - 2mC_J y. \tag{5.56}$$

Substituting (5.55) into (5.56) and equating $\frac{d}{dy}\tilde{u}_J(x^*, y^*)$ to zero we obtain

$$\frac{d}{dy}\tilde{u}_J(x^*, y^*) = \frac{h_{car}^4 h_J^2}{2C_{car}(\sigma^2 + h_J^2 y^*)^3} - 2mC_J y^* = 0,$$

$$y^*(\sigma^2 + h_J^2 y^*)^3 = \frac{h_{car}^4 h_J^2}{4mC_{car}C_J}.$$

Case 2. $x^* = PmaxC$, $0 < y^* < \frac{PmaxJ}{m}$. To find the maximum of the function $\tilde{u}_J$, as in the previous case, we calculate the derivative $\frac{d}{dy}\tilde{u}_J(PmaxC, y)$ and equate it to zero. We derive the following equality:

$$0 = \frac{d}{dy}\tilde{u}_J(PmaxC, y^*) = \frac{h_{car}^2 h_J^2 PmaxC}{(\sigma^2 + h_J^2 y^*)^2} - 2C_J m y^*.$$

From the previous equation we deduce

$$y^*(\sigma^2 + h_J^2 y^*)^2 = \frac{h_{car}^2 h_J^2 PmaxC}{2mC_J}. \tag{5.57}$$

By analogy with the consideration of case 2 in Theorem 2, we conclude that the conditions (5.57) and $\frac{d}{dx}\tilde{u}_{car}(PmaxC, y^*) \geq 0$ are necessary and sufficient for point (x^*, y^*) being the Nash equilibrium. We obtain

$$\frac{d}{dx}\tilde{u}_{car}(PmaxC, y^*) = \frac{h_{car}^2}{\sigma^2 + h_J^2 y^*} - 2C_{car}PmaxC \geq 0.$$

Case 3. $0 < x^* < PmaxC$, $y^* = \frac{PmaxJ}{m}$. By analogy with case 2, we conclude that in order to find x^* it is enough to equate the derivative $\frac{d}{dx}\tilde{u}_{car}(x^*, \frac{PmaxJ}{m})$ to zero. Therefore, we obtain the

following equation:

$$0 = \frac{d}{dx}\tilde{u}_{car}\left(x^*, \frac{PmaxJ}{m}\right) = \frac{h_{car}^2}{\sigma^2 + h_J^2 \frac{PmaxJ}{m}} - 2C_{car}x^*$$

From the last equation we conclude that

$$x^* = \frac{h_{car}^2}{2C_{car}(\sigma^2 + h_J^2 \frac{PmaxJ}{m})}.$$

By analogy with case 2 of Theorem 2, we conclude that it is necessary and sufficient for the point $(x^*, \frac{PmaxJ}{m})$ to be Nash equilibrium if the following inequality:

$$\frac{d}{dy}\tilde{u}_J\left(x^*, \frac{PmaxJ}{m}\right) = \frac{h_{car}^2 h_J^2 x^*}{(\sigma^2 + h_J^2 \frac{PmaxJ}{m})^2} - 2C_J PmaxJ \geq 0$$

is satisfied.

Case 4. $x^* = PmaxC$, $y^* = \frac{PmaxJ}{m}$. As in case 4 of Theorem 2, we conclude that the conditions $\frac{d}{dx}\tilde{u}_{car}\left(PmaxC, \frac{PmaxJ}{m}\right) \geq 0$ and $\frac{d}{dy}\tilde{u}_J\left(PmaxC, \frac{PmaxJ}{m}\right) \geq 0$ are necessary and sufficient. From the last two inequalities we deduce

$$\frac{d}{dx}\tilde{u}_{car}\left(PmaxC, \frac{PmaxJ}{m}\right) = \frac{h_{car}^2}{\sigma^2 + h_J^2 \frac{PmaxJ}{m}} - 2C_{car}PmaxC \geq 0,$$

$$\frac{d}{dy}\tilde{u}_J\left(PmaxC, \frac{PmaxJ}{m}\right) = \frac{h_{car}^2 h_J^2 PmaxC}{(\sigma^2 + h_J^2 \frac{PmaxJ}{m})^2} - 2C_J PmaxJ \geq 0.$$

Remark. Equalities (5.37) and (5.39) are equations for the variable y^*, which can be solved numerically using binary search, since their left side is an increasing function, and the right side is a constant.

5.6 Algorithms

In this chapter, we derive Nash equilibrium in the case of multi-channel and single-channel games. However, communication parameters such as channel gains are generally unknown, which makes their use in practice not always possible. Therefore, in practice, it is of interest to use machine learning algorithms, which by trial and error find the optimal strategy. In this chapter, we compare the performance of Policy hill-climbing [14] algorithm and several state-of-the-art modifications of the classic Q-learning algorithm [55]. All these algorithms are general and can be applied to turn-based games. The essence of the game is that at each step of k each agent is in some state s_k and at each turn performs the action a_k for which he receives a reward r_k. The goal of the agent is to maximize cumulative reward

$$\mathbf{E}\Big\{ \sum_k \gamma^k r_k \Big\}, \tag{5.58}$$

where $\mathbf{E}$ is a mathematical expectation and γ is called a discount-rate. Maximizing value (5.58) could be interpreted as maximizing the average cumulative reward obtained by following the certain probabilistic policy. The discount rate $\gamma, (0 < \gamma < 1)$ determines how important the future rewards are to the agent. If γ is close to 1, this means the high importance of the rewards, otherwise they are less important and the agent focuses more on the current state.

The essence of classic Q-learning algorithm is in recalculating the Q-matrix in which a action utility made from the state s is stored. With a probability ε, the agent selects an action randomly, in other cases it acts greedily and selects an action with a maximum Q-value. Often the parameter ε decreases to zero with an increasing number of iterations, since it is believed that the environment becomes explored quite well over time and does not require future exploration. In this chapter, we use exponential ε decay with a starting value of ε_0, a limiting value of ε_∞, and *decay_rate* given by the formula

$$\varepsilon = \varepsilon_\infty + (\varepsilon_0 - \varepsilon_\infty)e^{-iteration\_number/decay\_rate}, \tag{5.59}$$

where *iteration_number* is the number of the game iterations until the current moment. At each step, the recalculation of Q-matrix is performed according to the formula

$$Q^{new}[s_k, a_k] = Q^{old}[s_k, a_k] + \alpha(r_k + \gamma \max_a Q^{old}[s_{k+1}, a] - Q^{old}[s_k, a_k]). \qquad (5.60)$$

One of the main parameters of this algorithm is learning rate α which determines how significant the impact of new experience on Q-values would be.

Policy hill-climbing algorithm (PHC) is a modification of Q-learning. Its main difference from Q-learning is the choice of action. Q-learning is based on the greedy choice of the action a from the state s with the highest value $Q[s, a]$ and exploration of the environment with the probability ε. The PHC algorithm selects each action with a certain probability, which is updated each time taking into account the received awards.

Modifications of the classical Q-learning algorithm [55] discussed below are based on neural networks. A feature of Deep Q-learning [57] is that instead of memorizing values in Q-matrix (the size of which could be very large), the algorithm trains the neural network to store Q-values. To do this, it uses a special method of training, called experience replay. The essence of this method is that previous experiences are stored in a buffer of constant length *replay_mem_size* and are repeatedly used to train a neural network. After each iteration of the algorithm, *batch_size* of the previous experiences are randomly extracted from the buffer and used to retrain the network.

Another modification of Q-learning is called Double Q-learning [86]. Its creation is caused by the fact that values of Q-matrix can be locally overestimated in comparison with the real values. The essence of this modification is that instead of using one network, two are used. One is the current version of the network, and the other is an old copy saved a few steps back. An old copy of the network is updated every *update_target_frequency* iterations. One network version is used for value evaluation and another for the next action selection.

A modification called Dueling Q-learning [93] improves convergence and stabilizes the

training process by introducing a new element called advantage. The essence of advantage is that it is used to compare the Q-value of the current action and the average Q-value, so the algorithm tries to encourage more promising actions. We also implement Double Dueling Q-learning, which is a combination of the ideas of Double Q-learning and Dueling Q-learning.

Anti-jamming game algorithm

 1: **while** (Game is not terminated) **do**
 2:     Recalculate jammer power distribution
 3:     Recalculate state of the system using Intelligent Driver model
 4:     Choose new vehicle transmission channel according to pseudorandom sequence
 5:     Retrieve and discretize $SINR\_old$ from memory obtained from the previous iteration
 6:     Calculate ε using exponential decay rule (5.59)
 7:     $new\_action = Learning\_Algorithm(SINR\_old, \varepsilon)$
 8:     Calculate and discretize $SINR\_new$ after action new_action
 9:     Add to memory $(SINR\_old, new\_action, SINR\_new, reward)$
10:     Retrain algorithm
11:     Save current state of the system
12: **end while**

The anti-jamming game algorithm is described in pseudocode. We first consider the case of Q-learning modifications based on neural networks such as Deep Q-learning, Dueling Q-learning, and Dueling Double Q-learning, and then describe what should be changed in this pseudo-code if PHC algorithm is used. We discretize the range [0, *PmaxJ*] into *power_level_num* levels. In line 2, by exhaustive search over all power distributions the jammer finds an optimal distribution of power between *m* channels. Taking into account the previous vehicle state in line 3 the vehicle locations are updated using the Intelligent driver model from Section 4.4. In line 4, the transmitting vehicle changes the channel according to the pseudorandom sequence, which is assumed to be unknown to the jammer. To discretize the SINR in lines 5 and 8, the *disc_step* step size is used. Line 6 evaluates the value of ε which is responsible for the amount of exploration in Q-learning algorithms. Regardless of which modification of the Q-learning algorithm we use lines 7-10 look the same. In line 7, the algorithm predicts the next action. To do this, it returns a value of 2, 1 or 0, meaning that the vehicle must transmit signal on a higher power level, stay at the current level, or go to a level lower, respectively. It must be ensured

that the level does not go beyond the permissible power limits. In line 9 the new system state is added to memory as an array of four values ($SINR\_old, new\_action, SINR\_new, reward$). Line 10 calls a function that retrieves the *batch_size* of previous experiences, updates the state estimate using the formula (5.60), and trains the neural network to remember the updated values. In the case of the PHC algorithm, line 6 is not needed, since this algorithm does not have the parameter ε, in line 10, the called algorithm additionally recalculates the probabilities with which actions would be selected in the future.

5.7 Simulations

We consider the case of adaptive jammer, which is the most dangerous for the network. Based on the state of communication at the previous moment, the jammer finds the optimal transmit power by considering all the possible options. We perform simulations in the single-channel and multi-channel cases and compare the performance of PHC, Deep Q-learning and its recent modifications Dueling Q-learning and Dueling Double Q-learning. We assume the following parameter values (see their description in the previous section):

$$\gamma = 0.7, replay\_mem\_size = 50, batch\_size = 32,$$

$$update\_target\_frequency = 20, \varepsilon_0 = 1,$$

$$\varepsilon_\infty = 0.01, decay\_rate = 100, disc\_step = 0.05.$$

The vehicle motion parameters are taken from Section 4.4. In all algorithms based on Deep Q-learning, we use the following neural network architecture. Since we want to speed up the learning process, it has only one hidden layer of size 64; the number of inputs equals 1, the number of outputs equals 3. We assume that the output size equals 3, since the network returns 3 values corresponding to the values of transmission at the next power level (which is 1 higher), the transmission at the current level, and transmission at the previous power level

(which is 1 lower), respectively. In the case of Dueling Q-learning, advantage and value layers are added to this architecture. We use a fairly low value of $\gamma = 0.7$, because the system is constantly changing and we want the network to concentrate more on current rewards than on the future rewards. Since the system is changing rapidly, we assume a low value for the *update_target_frequency* in Dueling Double Q-learning, so that the system can quickly adapt to new experiences. For the same reason, we assume a low value of *replay_mem_size* in all versions of Deep Q-learning.

5.7.1 Single-channel game with quadratic power function

In this section, we discuss simulations of a single-channel anti-jamming game. We assume that the vehicles are located on the same road, with the jammer chasing two communicating cars. Figures 5.2a – 5.3b shows graphs of vehicle rewards and SINR in the case in which the distance between all consecutive network vehicles remains constant and equals 6.4 meters. Figures 5.2a and 5.2b shows graphs in the case when the initial coordinates of the vehicle equal 9.6 and 16 meters, and the jammer initial coordinate is 0.8 meters. In simulations corresponding to Figures 5.3a and 5.3b it is assumed that the vehicles are moving according to the Intelligent Driver model described in Section 4.4 with the parameter values given in this section. Due to the fact that the car in front is moving according to the free road model (since there are no other vehicles in front of it), the distance between it and the transmitting vehicle increases over time, resulting in worse communication quality. This explains decreasing of the graphs in Figures 5.3a and 5.3b.

The red color in all figures indicates the Nash equilibrium rewards and SINR calculated according to the formulas from the theorems proven in this chapter. As can be seen from Figures 5.2a – 5.3b, the graphs converge to theoretical predictions, confirming their correctness. We assume that the learning rate α is 0.05 in the case of PHC and 0.01 for the rest of the algorithms. We increase the learning rate, because the otherwise PHC would converge to Nash Equilibrium too slowly.

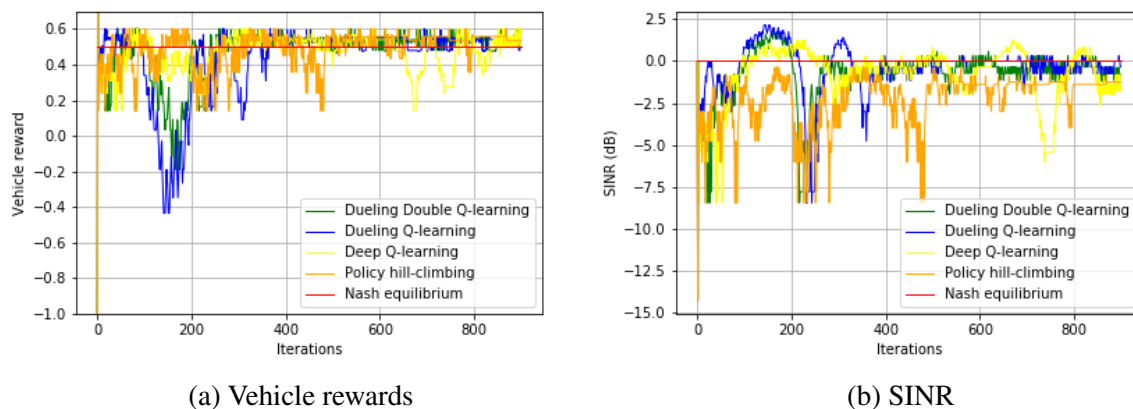(a) Vehicle rewards   (b) SINR

Figure 5.2: Single-channel game with quadratic power function and constant intervehicle distance.

Figures 5.2a and 5.2b show that Dueling Q-learning and Dueling Double Q-learning are the most stable among all considered algorithms, while Deep Q-learning and PHC have significant deviations from Nash equilibrium. Figures 5.3a and 5.3b show that all the algorithms adapt quite well to changes in the network state. However, the PHC is closest to Nash equilibrium throughout all iterations, while the rest of the algorithms deviates significantly from the optimal curve. This is due to the fact that Q-learning algorithms use experience replay which allows for constantly retraining experience from the past. Even taking into account the fact that we chose the small buffer size $replay\_mem\_size = 50$ (usually such a buffer has a size of the order of 10000), it can be seen from the simulations that such algorithms adapt to a change in environment with a noticeable delay. Thus, if the state of the network changes rapidly, PHC is the best among considered algorithms.

5.7.2 Multi-channel game with quadratic power function

Figures 5.4a – 5.5b show graphs in the case of a multi-channel game with $m = 3$ channels. We assume that in this case the vehicle changes the transmission channels according to a pre-determined pseudo-random sequence and each of the $m = 3$ channels in it is chosen with the same probability. Since the jammer must divide its power between channels in a multi-channel

case, the rewards of the vehicle in this case are higher than in a single-channel case. To speed up the convergence rate, we increase the learning rete to 0.3 in the case of PHC and to 0.05 for other algorithms. Figures 5.4a and 5.4b show that all algorithms perform quite well in the case in which the distances between cars are constant, especially PHC and Double Dueling Q-learning, however Dueling Q-learning has a big deviation from Nash equilibrium at iterations 900–1000. In the dynamic case (Intelligent driver model) presented in Figures 5.5a and 5.5b it can be seen that the PHC adapts to a change in environment faster and deviates less from Nash equilibrium.

According to Theorem 3, the optimal jammer strategy is to transmit the same power across all channels. The optimality of this strategy is confirmed by the simulations in Figures 5.6a and 5.6b in the case of PHC algorithm and constant or variable intervehicle distance. In these simulations, the jammer iterates over all combinations of possible power distributions and selects the one that maximizes its utility function. The graphs confirm that transmitting the same power across all channels is the optimal jammer strategy among all possible.

(a) Vehicle rewards   (b) SINR
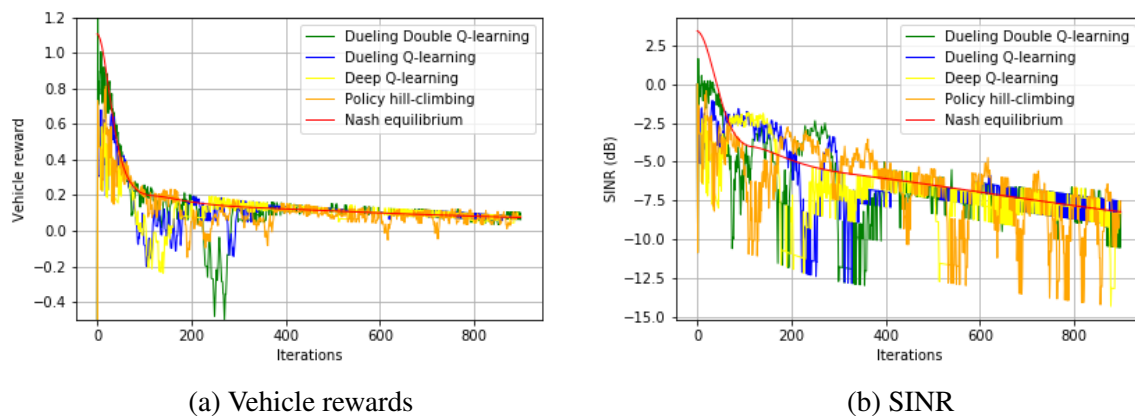
Figure 5.3: Single-channel game with quadratic power function and variable intervehicle distance.

(a) Vehicle rewards

(b) SINR
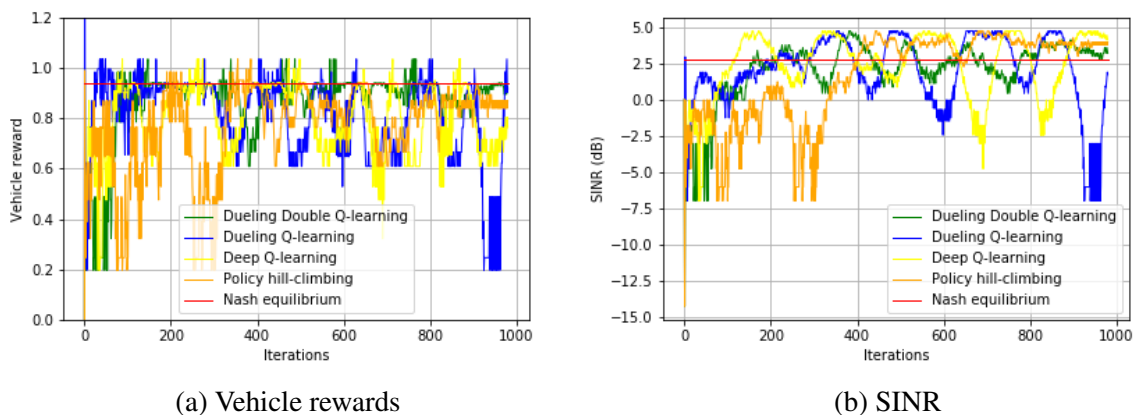
Figure 5.4: Multi-channel channel game with quadratic power function and constant intervehicle distance.

(a) Vehicle rewards

(b) SINR
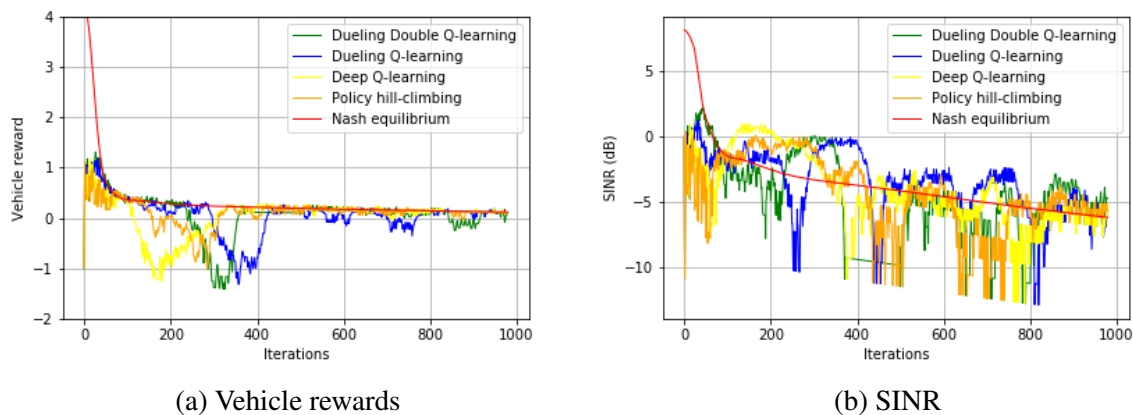
Figure 5.5: Multi-channel game with quadratic power function and variable intervehicle distance.

5.8 Chapter summary

This chapter considers the single-channel and multi-channel VANET anti-jamming game. This game is an antagonistic game between the jammer and a pair of communicating vehicles. At the beginning of the chapter, we assume that the power term of the vehicle and the jammer utility functions is linear. In this case, we prove that the optimal strategy of the vehicle is signal transmission at the maximum power (see Theorem). We change utility functions by replacing the term linearly dependent on power with a quadratic one. This case is closer to the actual operational conditions, since the cost of the transmission at higher levels should

(a) Constant intervehicle distance   (b) Variable intervehicle distance
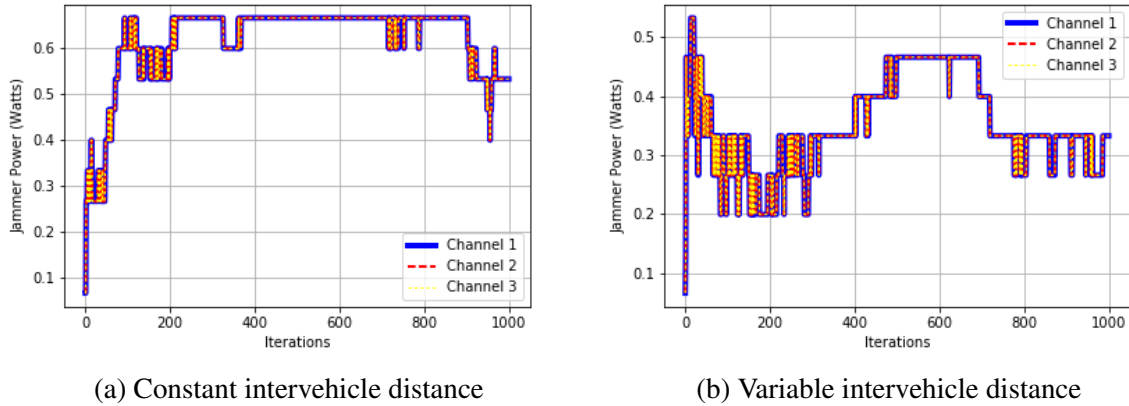
Figure 5.6: Jammer power distribution among 3 channels game with quadratic power function in the cases of constant (left graph) and variable (right graph) intervehicle distance.

be higher than at low levels. We examine a single-channel and multi-channel game under this condition. In this case the optimal strategy is not trivial as before. We express the Nash equilibrium of the system through communication parameters (Theorems 2 and 3). We also consider modern machine learning algorithms such as Deep Q-learning, Dueling Q-learning, Double Dueling Q-learning, and Policy Hill Climbing, and compare their performance. All the algorithms successfully converge to the theoretically deduced Nash equilibrium, however, Policy Hill Climbing shows better adaptability in the case of a rapidly changing system state.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1   Summary of the results

The statistical information about VANET is practically important and allows, for example, the estimation of the network load. In Chapter 2, we present a versatile investigation of VANET connectivity statistics in the case where vehicles are distributed on the road and express results in terms of parameters of known probability distributions of intervehicle distance and connectivity model. We derive distributions of number of clusters, cluster size, largest cluster size and disconnected vehicle number as well as expected value and variance of all of these characteristics. The results are confirmed by the simulations made for urban and rural traffic scenarios.

Another important aspect of VANET is how network connectivity characteristics change over time. In Chapter 3, we consider evolution of the vehicle network on a highway assuming that the connection between pairs of consecutive cars can be established with a certain probability. We derive the probability distributions describing evolution of such characteristics of the network as distribution of link duration between every pair of cars, time of cluster existence,

and probability of cluster existence between two fixed moments of time. We also examine such a connection characteristic as ω-stable connection. This type of connection means that the connection between the vehicles may be lost, but must be restored within each specific time interval. This communication performance is closer to the actual operating conditions under which the connection between the cars can be interrupted, but we want to guarantee that it is reestablished again with a certain regularity. We derive an algorithm for calculating the probability that a connection on a given interval is ω-stable. All these derivations are done in the frame of the Wang-Moayery model that describes communication channel using the two state Markov chain model.

Chapter 4 is devoted to the study of the statistical properties of VANET on a two-dimensional map with an almost arbitrary topology. To the best of our knowledge, this is the first study on such a general scenario. Prior to this dissertation, only cases of highways and intersections have been considered. However, an accurate analysis of statistical properties on a 2D map is not possible due to the large number of unknowns such as car routes, traffic lights, and traffic model, etc. Therefore, we derive approximate distributions of such characteristics as the distribution of the number of clusters and the size of the average cluster. In order to verify the obtained results, we develop a simulator that simulates both realistic traffic and communication models. The obtained theoretical results quite accurately approximate the simulation results in the case of urban and rural traffic.

Chapter 5 discusses a VANET anti-jamming game between the jammer and two connected vehicles. In terms of game theory, such antagonistic games are formulated as games in which each player maximizes their utility functions. Finding the optimal strategy in this setting can be interpreted as finding the Nash equilibrium of this game. We derive Nash equilibrium in cases of single-channel and multi-channel games expressed in terms of system parameters. In practice, however, some of the transmission parameters are unknown, therefore, machine learning has become popular in the literature, which by trial and error, finds the optimal strategy. We compare the performance of such machine learning algorithms as Policy Hill Climbing, Deep

Q-learning, Dueling Q-learning, and Dueling Double Q-learning. All the indicated algorithms by trial and error are able to converge to theoretically derived optimal strategies, both in the case of constant, and in case of variable distances between vehicles.

The results of this dissertation are published in journals and conference abstracts.

6.2   Suggestions for future research

Chapter 2 explores the statistical properties of VANET on a highway. These studies can be continued in the case of multi-lane traffic. The statistical properties of VANET in the case of multi-lane traffic have not been considered in the literature because of its complexity. However, even if accurate results are difficult to obtain in this case, there is probably a potential to obtain good approximations of these distributions.

The results of Chapter 3 regarding the evolution of the network are deduced under the assumption of a constant speed of movement being the same for all vehicles. The case in which this proposition is omitted or at least weakened is worth consideration. This case is much more difficult to study, since in it, the parameters of the Markov model begin to depend on vehicle motion parameters. Perhaps this case is too complicated for analysis, so all the results of Chapter 3 cannot be generalized to it, but at least studying the duration of the connection between the pair of vehicles is definitely possible.

Chapter 4 shows approximate statistical distributions of the number of clusters and the average cluster size. Further research in this direction can be aimed at making them more accurate. In the derivation of these formulas, we try to minimize information about the traffic model and traffic light states and incorporate it into the model through the distribution of the distance between cars. Perhaps, expanding this information could improve the predictive accuracy of this model.

The results of Chapter 5 regarding an anti-jamming game are deduced when the couple of vehicles establishes a connection, and the jammer tries to disrupt it. However, the case in which

several vehicles cooperate in order to maximize the overall utility function is also important for the practice. In other words, it is also important to consider a cooperative anti-jamming game and obtain analytical results regarding the optimal strategy of the vehicles. It is also important to compare the performance of modern multi-agent machine learning algorithms applied to this game in order to find a stable algorithm with a high convergence rate.

Bibliography

[1] K. Ahiska, M. K. Ozgoren, and M. K. Leblebicioglu. Autopilot design for vehicle cornering through icy roads. *IEEE Transactions on Vehicular Technology*, 67(3):1867–1880, 2018.

[2] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[3] Nizar Alsharif, Khalid Aldubaikhy, and Xuemin Sherman Shen. Link duration estimation using neural networks based mobility prediction in vehicular networks. In *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4. IEEE, 2016.

[4] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[5] Mohamed A Aref and Sudharman K Jayaweera. A novel cognitive anti-jamming stochastic game. In *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*, pages 1–4. IEEE, 2017.

[6] Uri M Ascher and Linda R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam, 1998.

[7] Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. 2013.

[8] Robert Aumann and Adam Brandenburger. Epistemic conditions for nash equilibrium. *Econometrica: Journal of the Econometric Society*, pages 1161–1180, 1995.

[9] Ikechukwu K Azogu, Michael T Ferreira, Jonathan A Larcom, and Hong Liu. A new anti-jamming strategy for vanet metrics-directed security defense. In *2013 IEEE Globecom Workshops (GC Wkshps)*, pages 1344–1349. IEEE, 2013.

[10] A.V. Babu and V.K. Muhammed Ajeer. Analytical model for connectivity of vehicular ad hoc networks in the presence of channel randomness. *Int. J. Commun. Syst.*, 26(7):927–946, 2013.

[11] Vahid Behbood, Jie Lu, and Guangquan Zhang. Fuzzy refinement domain adaptation for long term prediction in banking ecosystem. *IEEE Transactions on Industrial Informatics*, 10(2):1637–1646, 2013.

[12] Abderrahim Benslimane and Huong Nguyen-Minh. Jamming attack model and detection method for beacons under multichannel operation in vehicular networks. *IEEE Transactions on Vehicular Technology*, 66(7):6475–6488, 2016.

[13] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[14] Michael Bowling and Manuela Veloso. Rational and convergent learning in stochastic games. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pages 1021—1026. Morgan Kaufmann Publishers Inc., 2001.

[15] Widodo Budiharto, Alexander AS Gunawan, Jarot S Suroso, Andry Chowanda, Aurello Patrik, and Gaudi Utama. Fast object detection for quadcopter drone using deep learning. In *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, pages 192–195. IEEE, 2018.

[16] Mahil Carr, Vadlamani Ravi, G Sridharan Reddy, and D Veranna. Machine learning techniques applied to profile mobile banking users in india. *International Journal of Information Systems in the Service Sector (IJISSS)*, 5(1):82–92, 2013.

[17] Neelakantan Pattathil Chandrasekharamenon and Babu AnchareV. Connectivity analysis of one-dimensional vehicular ad hoc networks in fading channels. *EURASIP Journal on Wireless Communications and Networking*, 2012(1), Jan 2012.

[18] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2492–2501, 2018.

[19] Ho Ting Cheng, Hangguan Shan, and Weihua Zhuang. Infotainment and road safety service support in vehicular networking: From a communication perspective. *Mechanical systems and signal processing*, 25(6):2020–2038, 2011.

[20] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *IN COMPUTER VISION AND PATTERN RECOGNITION*, pages 3642–3649, 2012.

[21] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM.

[22] Federal Communications Commission et al. Amendment of the commission's rules regarding dedicated short-range communication service in the 5.850-5.925 ghz band. *FCC, Washington, DC, USA, Tech. Rep. FCC*, pages 02–302, 2002.

[23] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[24] Mike Daily, Swarup Medasani, Reinhold Behringer, and Mohan Trivedi. Self-driving cars. *Computer*, 50(12):18–23, 2017.

[25] Johannes Dams, Martin Hoefer, and Thomas Kesselheim. Jamming-resistant learning in wireless networks. *IEEE/ACM Transactions on Networking*, 24(5):2809–2818, 2015.

[26] Mario De Felice, Ian Victor Calcagni, Francesca Pesci, Francesca Cuomo, and Andrea Baiocchi. Self-healing infotainment and safety application for vanet dissemination. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 2495–2500. IEEE, 2015.

[27] Rahul C Deo. Machine learning in medicine. *Circulation*, 132(20):1920–1930, 2015.

[28] G. Dubosarskii, S. L. Primak, and X. Wang. Evolution of vehicle network on a highway. volume 68, pages 9088–9097, 2019.

[29] Gleb Dubosarskii, Serguei Primak, and Xianbin Wang. Stochastic geometry of network of randomly distributed moving vehicles on a highway. *International Journal of Communication Systems*, 33(5):e4277, 2020.

[30] Yexian Fan, Xingyu Xiao, and Wei Feng. An anti-jamming game in vanet platoon with reinforcement learning. In *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pages 1–2. IEEE, 2018.

[31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[32] Louis Gordon, Mark F. Schilling, and Michael S. Waterman. An extreme value theory for long head runs. *Probab. Theory Relat. Fields*, 72(2):279–287, Jun 1986.

[33] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS*, pages 5–6, 2005.

[34] Youngjune Gwon, Siamak Dastangoo, Carl Fossa, and HT Kung. Competing mobile network game: Embracing antijamming and jamming strategies with reinforcement learning. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 28–36. IEEE, 2013.

[35] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[36] Tin Kam Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA, 1995.

[37] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, August 1998.

[38] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[39] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

[40] Andrew Ilyas, Logan Engstrom, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Are deep policy gradient algorithms truly policy gradient algorithms? *arXiv preprint arXiv:1811.02553*, 2018.

[41] C. Jayapal and S. S. Roy. Road traffic congestion management using vanet. In *2016 International Conference on Advances in Human Machine Interaction (HMI)*, pages 1–7, 2016.

[42] D. Jia, K. Lu, and J. Wang. A disturbance-adaptive design for vanet-enabled vehicle platoon. *IEEE Transactions on Vehicular Technology*, 63(2):527–539, 2014.

[43] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[44] John B Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[45] A. Kesting, M. Treiber, and D. Helbing. Connectivity statistics of store-and-forward intervehicle communication. *IEEE Trans. Intell. Transp. Syst.*, 11(1):172–181, 2010.

[46] Laszlo Kozma. k nearest neighbors algorithm (knn). *Helsinki University of Technology*, 2008.

[47] S. Kwon, Y. Kim, and N. B. Shroff. Analysis of connectivity and capacity in 1-d vehicle-to-vehicle networks. *IEEE Trans. Wireless Commun.*, 15(12):8182–8194, 2016.

[48] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[49] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[50] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[51] Michael L Littman. Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, pages 322–328, 2001.

[52] Xiaozhen Lu, Dongjin Xu, Liang Xiao, Lei Wang, and Weihua Zhuang. Anti-jamming communication game for uav-aided vanets. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[53] Xiaozhen Lu, Dongjin Xu, Liang Xiao, Lei Wang, and Weihua Zhuang. Anti-jamming communication game for uav-aided vanets. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[54] Ninsi Mary Mathew and PC Neelakantan. Analyzing the network connectivity probability of a linear vanet in nakagami fading channels. In *International Conference on Distributed Computing and Networking*, pages 505–511. Springer, 2014.

[55] Francisco S Melo. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pages 1–4, 2001.

[56] Raghda Nazar Minihi and Haider M AlSabbagh. Analysis of path duration in vanets using b-mfr forwarding method.

[57] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

[58] Roger B Myerson. *Game theory*. Harvard university press, 2013.

[59] PC Neelakantan and AV Babu. Network connectivity probability of linear vehicular ad hoc networks on two-way street. *Communications and Network*, 4(04):332, 2012.

[60] Seh Chun Ng, Wuxiong Zhang, Yu Zhang, Yang Yang, and Guoqiang Mao. Analysis of access and connectivity probabilities in vehicular relay networks. *IEEE Journal on Selected Areas in Communications*, 29(1):140–150, 2010.

[61] Ziad Obermeyer and Ezekiel J Emanuel. Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216, 2016.

[62] Erol A. Pekoz and Sheldon M. Ross. A simple derivation of exact reliability formulas for linear and circular consecutive-k-of-n: F systems. *Journal of Applied Probability*, 32(2):554–557, 1995.

[63] Z. Peng, Y. Tian, D. Wang, and L. Liu. Autopilot design for a robotic unmanned surface vehicle. In *2015 34th Chinese Control Conference (CCC)*, pages 6116–6120, 2015.

[64] Jan Peters. Machine learning for motor skills in robotics. *KI-Künstliche Intelligenz*, 2008(4):41–43, 2008.

[65] Serguei Primak, Valeri Kontorovich, and Vladimir Lyandres. *Stochastic methods and their applications to communications: stochastic differential equations approach*. John Wiley & Sons, 2005.

[66] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[67] B. Ranft and C. Stiller. The role of machine vision for intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):8–19, 2016.

[68] Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, 2004.

[69] F. P. Rezha, T. S. Siadari, and Soo Young Shin. Adaptive transmission power in cluster-based routing vanet. In *2012 18th Asia-Pacific Conference on Communications (APCC)*, pages 539–543. Jeju Island, Korea, Oct 2012.

[70] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[71] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[72] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

[73] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 3(3):210–229, July 1959.

[74] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[75] A. Sarker, C. Qiu, and H. Shen. Quick and autonomous platoon maintenance in vehicle dynamics for distributed vehicle platoon networks. In *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 203–208, 2017.

[76] Mark F. Schilling. The longest run of heads. *College Math. J.*, 21(3):196–207, 1990.

[77] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[78] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[79] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 329. John Wiley & Sons, 2012.

[80] Marwaan Simaan and Jose B Cruz. On the stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11(5):533–555, 1973.

[81] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

[82] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[83] Gerald Tesauro. Temporal difference learning and td-gammon. *Commun. ACM*, 38(3):58–68, March 1995.

[84] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

[85] Martin Treiber and Arne Kesting. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[86] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[87] Bruce (Xiubin) Wang, Teresa M. Adams, Wenlong Jin, and Qiang Meng. The process of information propagation in a traffic stream with a general vehicle headway: A revisit. *Transp. Research Emerg. Technol. C*, 18(3):367 – 375, 2010. 11th IFAC Symposium: The Role of Control.

[88] H. Wang, R. P. Liu, W. Ni, W. Chen, and I. B. Collings. Vanet modeling and clustering design under practical traffic, channel and mobility conditions. *IEEE Trans. Wireless Commun.*, 63(3):870–881, 2015.

[89] Hong Shen Wang and Nader Moayeri. Finite-state markov channel-a useful model for radio communication channels. *IEEE transactions on vehicular technology*, 44(1):163–171, 1995.

[90] Qiwei Wang, Thinh Nguyen, Khanh Pham, and Hyuck Kwon. Mitigating jamming attack: A game-theoretic perspective. *IEEE Transactions on Vehicular Technology*, 67(7):6063–6074, 2018.

[91] Xiufeng Wang, Chunmeng Wang, Gang Cui, and Qing Yang. Practical link duration prediction model in vehicular ad hoc networks. *International Journal of Distributed Sensor Networks*, 11(3):216934, 2015.

[92] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

[93] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

[94] Z. Wen and C. Miao. Vehicle flow detection based on machine vision. In *2010 Second WRI Global Congress on Intelligent Systems*, volume 3, pages 70–72, 2010.

[95] H. S. Wilf. *Generatingfunctionology*. CRC Press, 3rd edition, 2005.

[96] D. Wu, J. Wu, and R. Wang. An energy-efficient and trust-based formation algorithm for cooperative vehicle platooning. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 702–707, 2019.

[97] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[98] Jingxian Wu. Connectivity analysis of a mobile vehicular ad hoc network with dynamic node population. In *2008 IEEE Globecom Workshops*, pages 1–8. IEEE, 2008.

[99] Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, pages 5279–5288, 2017.

[100] Liang Xiao, Tianhua Chen, Jinliang Liu, and Huaiyu Dai. Anti-jamming transmission stackelberg game with observation errors. *IEEE communications letters*, 19(6):949–952, 2015.

[101] Liang Xiao, Donghua Jiang, Dongjin Xu, Hongzi Zhu, Yanyong Zhang, and H Vincent Poor. Two-dimensional antijamming mobile communication based on reinforcement learning. *IEEE Transactions on Vehicular Technology*, 67(10):9499–9512, 2018.

[102] Liang Xiao, Xiaozhen Lu, Dongjin Xu, Yuliang Tang, Lei Wang, and Weihua Zhuang. Uav relay in vanets against smart jamming with reinforcement learning. *IEEE Transactions on Vehicular Technology*, 67(5):4087–4097, 2018.

[103] Liang Xiao, Caixia Xie, Minghui Min, and Weihua Zhuang. User-centric view of unmanned aerial vehicle transmission against smart attacks. *IEEE Transactions on Vehicular Technology*, 67(4):3420–3430, 2017.

[104] G. Yan and S. Olariu. A probabilistic analysis of link duration in vehicular ad hoc networks. *IEEE Trans. Intell. Transp. Syst.*, 12(4):1227–1236, 2011.

[105] Fuqiang Yao and Luliang Jia. A collaborative multi-agent reinforcement learning antijamming algorithm in wireless networks. *IEEE Wireless Communications Letters*, 2019.

[106] Yong Jiang, Jie Cao, Ruihua Liu, and Yaling Du. An algorithm for automatic guided vehicle based on machine vision and dr. In *2006 6th World Congress on Intelligent Control and Automation*, volume 2, pages 8582–8586, 2006.

[107] Yongjin Zhang, Wei Li, Xiao Wang, Jianhui Zhao, and Yi Yuan. General design of vehicle safety assistant system based on machine vision and electronic control steering.

In *2010 International Conference On Computer Design and Applications*, volume 3, pages V3–20–V3–23, 2010.

[108] Saleh Yousefi, Eitan Altman, Rachid El-Azouzi, and Mahmood Fathy. Analytical model for connectivity in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(6):3341–3356, 2008.

[109] Long Yu, Yusheng Li, Chen Pan, and Luliang Jia. Anti-jamming power control game for data packets transmission. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 1255–1259. IEEE, 2017.

[110] Yuli Zhang, Yuhua Xu, Yitao Xu, Yang Yang, Yunpeng Luo, Qihui Wu, and Xin Liu. A multi-leader one-follower stackelberg game approach for cooperative anti-jamming: No pains, no gains. *IEEE Communications Letters*, 22(8):1680–1683, 2018.

[111] Fengchao Zhu, Feifei Gao, Minli Yao, and Hongxing Zou. Joint information-and jamming-beamforming for physical layer security with full duplex base station. *IEEE Transactions on Signal Processing*, 62(24):6391–6401, 2014.

Appendix A

Method of generating functions

Here we give a brief exposition of mathematical method of generating functions. This elegant and effective method is used in Chapter 2 to obtain distributions of cluster size and number of disconnected vehicles. The essence of the method is that it treats infinite sequence of numbers a_k as the coefficients of a power series $\sum_{k=0}^{\infty} a_k x^k$. We use this method to establish a relation between the number of different representations of integer number as a sum of integer numbers with some restrictions. We touch only one aspect of this theory, for other applications we recommend to read the book [95]. Here we formulate several problems in ascending order of complexity. We need the results of the problems 3 and 4, but in order to obtain them we solve problems 1 and 2.

Problem 1. *What is the number of representations of positive integer number n as a sum of k positive integer summands?*

We denote coefficient of x^n in power series $F(x)$ by $coeff_{x^n} F(x)$. It is stated that the following coefficient:

$$coeff_{x^n}\left(x + x^2 + x^3 + \ldots\right)^k \tag{A.1}$$

equals the number of representations of n as k summands. Let us prove it. We can rewrite the

137

$\text{sum}\left(x + x^2 + x^3 + \ldots\right)^k$ as

$$\sum_{\alpha_1}\sum_{\alpha_2}\ldots\sum_{\alpha_k} x^{\alpha_1} x^{\alpha_2} \ldots x^{\alpha_k} = \sum_{\alpha_1}\sum_{\alpha_2}\ldots\sum_{\alpha_k} x^{\alpha_1+\alpha_2+\ldots+\alpha_k} = \sum_{l=1}^{\infty} x^l \sum_{\alpha_1+\alpha_2+\ldots+\alpha_k=l} 1. \qquad (A.2)$$

Therefore, the coefficient (A.1) indeed equals the number of representations of n as a sum of k summands $\alpha_1, \alpha_2, \ldots, \alpha_k$.

Problem 2. *What is the number of representations of positive integer number n as a sum of k positive integer summands not equaling r?*

The following formula gives the solution to this problem:

$$coeff_{x^n}\left(x + x^2 + x^3 + \ldots - x^r\right)^k. \qquad (A.3)$$

The only difference between formulas (A.1) and (A.3) is that we subtract x^r because integer summands do not equal r. It can be proven similar to the proof of formula (A.1).

Problem 3. *What is the number of representations of positive integer number n as a sum of k positive integer summands with the following restriction: among them exactly s numbers equal α?*

The answer is given by the following equality:

$$coeff_{x^n}\left\{\binom{k}{s} x^{\alpha s}\left(x + x^2 + x^3 + \ldots - x^\alpha\right)^{k-s}\right\}, \qquad (A.4)$$

because we can choose s summands equaling α in $\binom{k}{s}$ ways and other $k - s$ summands should not be equal α (see Problem 2).

Problem 4. *What is the number of representations of a positive integer number n as a sum of k positive integer summands assuming that s of them equal 1?*

By substituting $\alpha = 1$ into answer of Problem 3 we get the following result:

$$coeff_{x^n}\binom{k}{s} x^s (x^2 + x^3 + x^4 + \ldots)^{k-s}. \qquad (A.5)$$

Curriculum Vitae

Name:	Gleb Dubosarskii
Post-Secondary Education and Degrees:	Ural Federal University Yekaterinburg, Russia 2005–2008 B.Sc.
	Ural Federal University Yekaterinburg, Russia 2008–2011 M.Sc.
	The Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences Yekaterinburg, Russia 2011–2014 Ph.D. (Candidate of Phys. and Math. Sciences)
	University of Western Ontario London, Ontario, Canada 2016 – 2020 Ph.D.
Honours and Awards:	Academician A.F. Sidorov Prize given for outstanding work in the field of differential equations, 2015
Related Work Experience:	Graduate Teaching Assistant The University of Western Ontario 2016–2020
	Graduate Research Assistant The University of Western Ontario 2016–2020
	Research Assistant The Krasovskii Institute of Mathematics and Mechanics, Yekaterinburg, Russia 2011-2014
	Researcher The Krasovskii Institute of Mathematics and Mechanics, Yekaterinburg, Russia 2014-2016

Publications:

1. G. Dubosarskii, S. Primak and X. Wang, "Evolution of Vehicle Network on a Highway", in IEEE Transactions on Vehicular Technology, vol. 68, no. 9, pp. 9088–9097, 2019.

2. G. Dubosarskii, S. Primak and X. Wang, "Stochastic geometry of network of randomly distributed moving vehicles on a highway", International Journal of Communication Systems, pp. e4277, 2019.

Peer-reviewed conference proceedings:

1. G. Dubosarskii, S. Primak, X. Wang, "On the higher order statistics of car clustering in vehicle communications networks on a road", *in Proc. 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017.

2. G. Dubosarskii, S. Primak, X. Wang, "Connectivity and Clustering in a Network of Randomly Distributed Vehicles on a Highway", *in Proc. IEEE Int. Vehicular Technology Conference: VTC2017-Fall*, 2017.

3. G. Dubosarskii, S. Primak, X. Wang, "Multichannel Power Allocation Game against Jammer with Changing Strategy", *in Proc. 2018 IEEE Global Communications Conference (GLOBECOM)*, 2018.