Modeling and Analysis of Modern Vehicular Communication Using Determinantal Point Process

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

B.Tech

in

Electronics and Communication Engineering

By

MADHURJA TALUKDAR (202002032074)

MANASH PRATIM BORAH (202002031035)

Under the Supervision of

Kaushlendra Kumar Pandey

Assistant professor, Department of ECE

ELECTRONICS AND COMMUNICATION ENGINEERING
CENTRAL INSTITUTE OF TECHNOLOGY, KOKRAJHAR-783370

MAY 2024

BONAFIDE CERTIFICATE

This is to certify that the project titled ”Modeling and Analysis of Modern Vehicular Communication Using Determinantal Point Process” is a bonafide record of the work done by

MADHURJA TALUKDAR (202002032074)
MANASH PRATIM BORAH (202002031035)
in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering of the CENTRAL INSTITUTE OF TECHNOLOGY, KOKRAJHAR, during the year 2023-2024.

Kaushlendra Kumar Pandey
Supervisor

Dr. Bornali Bora Patowary
Head of the Department

DECLARATION

We hereby declare that the project work entitled ”Modeling and Analysis of Modern Vehicular Communication Using Determinantal Point Process” is an authenticated work carried out by us under the guidance of **Mr. Kaushlendra Kumar Pandey** for the requirements of Major Project-II. As part of the eight-semester curriculum of Bachelor of Technology in Electronics and Communication Engineering and this work has not been submitted for similar purpose anywhere else except to the Department of Electronics and Communication Engineering, Central Institute of Technology, Kokrajhar.

Date:.....................

Place: C.I.T,Kokrajhar

MADHURJA TALUKDAR

ROLL NO: 202002032074

MANASH PRATIM BORAH

ROLL NO: 202002031035

ACKNOWLEDGEMENT

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

Mr. Kaushlendra Kumar Pandey, our project guide, for helping us and guiding us in the course of this project .

Dr. Bornali Bora Potowary, the Head of the Department, Department of Electronics and communication engineering.

Finally, we would like to thank our friends and family for their constant support.

ABSTRACT

This research investigates the application of Determinantal Point Processes (DPPs) in modeling and analyzing modern vehicular communication systems. Vehicular networks present challenges such as dynamic node movements, signal variations, and interference, necessitating robust modeling techniques. Traditional methods often lack precision in capturing the spatial dynamics and connectivity probabilities unique to vehicular environments. This study proposes leveraging DPPs to address these challenges by incorporating inter-node repulsion, spatial diversity, and optimized node selection.Using a combination of Poisson Point Process (PPP) for vehicle locations and DPP for Road Side Unit (RSU) deployments, the methodology focuses on generating realistic vehicular network data. The utilization of DPPs enables the modeling of repulsive forces between nodes, reflecting real-world scenarios where nodes avoid clustering and interference-prone areas. A kernel matrix is constructed to encode spatial relationships among potential node locations, aiding in connectivity analysis and optimal node subset selection.Eigenvalue analysis of the kernel matrix guides the development of a tailored conditional probability function, optimizing node placement while mitigating interference-related issues. This approach ensures strategically distributed node subsets, enhancing network connectivity and reliability. Simulation results demonstrate the effectiveness of the DPP-based model, showcasing improvements in coverage, reliability, and computational efficiency compared to conventional methods.This research contributes to a deeper understanding of connectivity probabilities, spatial dynamics, and network performance metrics in vehicular communication. The proposed DPP-based approach holds promise for developing resilient, adaptable, and optimized communication systems vital for modern vehicular networks.

TABLE OF CONTENTS

List of Figures

ABBREVIATION

• SINR	Signal-to-Interference-Plus-Noise Ratio
• SIR	Signal-to-Interference Ratio
• SNIR	Signal-to-Noise interference Ratio
• DPP	Determinantal Point Process
• LOS	Line of Sight
• PPP	Poisson Point Process
• RSU	Road Sise Unit
• AWGN	Additive White Gaussian Noise
• THz	Terahertz
• RF	Radio Frequency
• BSs	Base Stations

Chapter 1

INTRODUCTION

Terahertz (THz) communication networks are emerging as a promising solution for meeting the increasing demand for high-speed wireless communication. These networks operate in the frequency range of 0.1 to 10 THz and offer several advantages over traditional microwave-based communication systems, such as wider bandwidth, higher data rates, and reduced signal attenuation
However, the deployment of THz networks poses several challenges, such as the limited range and sensitivity to atmospheric conditions. Therefore, to ensure reliable and efficient communication in THz networks, it is essential to perform a thorough coverage analysis . Coverage analysis involves evaluating the coverage area of a communication system, taking into account various factors such as the transmission power, antenna gain, and propagation characteristics. In the case of THz networks, coverage analysis is particularly important due to the high attenuation of THz signals in the atmosphere. By conducting a comprehensive coverage analysis, network engineers and designers can optimize the placement of base stations and antennas to ensure maximum coverage and reliable communication in THz networks . This can lead to improved network performance and enhanced user experience, making THz networks a viable solution for future high-speed wireless communication applications . [1],[2].

1.0.1 What is DeterminanTal Point Processes and how it is Usefu?

A Determinantal Point Process (DPP) is a special type of point process used in probability theory and statistics. It's a way to model the random placement of points in a space, but unlike a simple random placement, DPP considers interactions between the points themselves. Here's a breakdown:

Random vs. Determinantal: In a standard random point process (like throwing darts on a board), the locations are independent - one point's placement doesn't affect the others. However, a DPP introduces ”repulsion” between points. As you add more points, they tend to spread out more evenly to avoid being too close together.

Mathematical Formulation: DPPs are characterized by a kernel function (a matrix) that captures the pairwise interactions between points. Intuitively, a higher value in the kernel for two potential locations indicates a lower probability of them being chosen together.

Applications of DPPs:

DPPs have found applications in various fields due to their ability to model repulsive interactions:

Wireless Networks: As seen in the code example you provided, DPPs can be used to strategically place Roadside Units (RSUs) in a network. By considering the interactions between RSUs, the DPP approach can potentially improve user connectivity and reduce interference compared to random placement.

Machine Learning: DPPs can be used for tasks like sensor placement, active learning (selecting the most informative data points), and group formation (clustering data points with similar characteristics).

Physics: DPPs can model systems with repulsive interactions between particles, such as electrons in atoms or fermions in quantum mechanics.

Benefits of using DPPs:

Improved Efficiency: By considering interactions, DPPs can lead to more efficient placements compared to random approaches. This can be beneficial in applications like wireless network design or sensor placement.

Flexibility: The kernel function in a DPP can be customized to capture different types of interactions between points. This allows for modeling various scenarios.

However, DPPs can also be computationally expensive to implement for large-scale problems. Determinantal Point Processes offer a powerful tool for modeling repulsive interactions between points in random distributions. Their ability to optimize placement and consider inter-point relationships makes them valuable in various fields like wireless networks, machine learning, and physics.

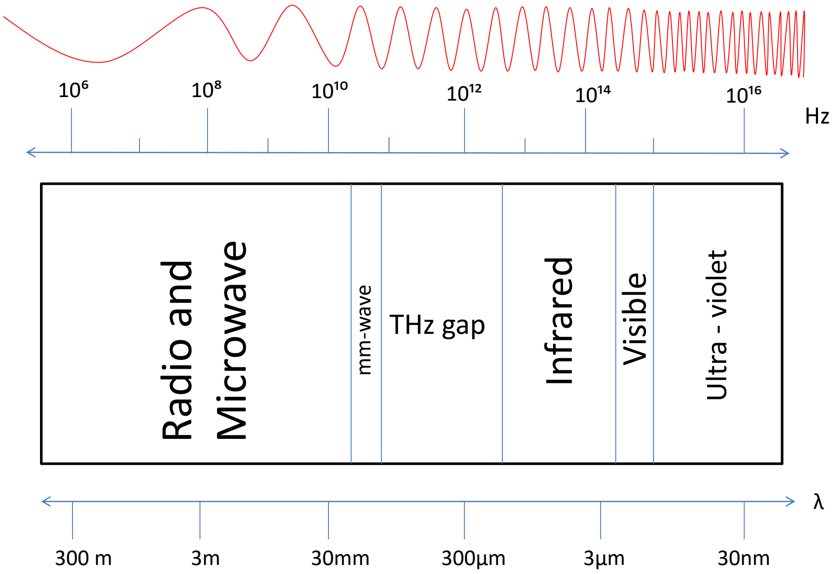


Figure 1.1: **THz band's position in the electromagnetic spectrum is depicted in a schematic diagram.**

1.0.2 Why THz Network is Important ?

Terahertz (THz) networks are being considered for various reasons

1. **Enabling ultra-broadband communication:** THz frequencies offer significantly higher bandwidth compared to traditional radio frequencies (RF), allowing for ultra-fast data rates and meeting the growing demand for bandwidth-intensive applications.

2. **Addressing spectrum scarcity:** The RF spectrum is becoming crowded, leading to limited availability. THz networks utilize an underutilized spectrum range, providing additional bandwidth resources and alleviating spectrum scarcity issues.

3. **Enhancing capacity:**With the exponential growth of data traffic, THz networks can significantly enhance network capacity by utilizing the extensive bandwidth available in the THz frequency range. This enables the simultaneous transmission of large amounts of data and supports future application requirements.

4. **Bridging frequency ranges:** By operating in the THz band, networks bridge the gap between millimeter wave (mm-Wave) and optical frequency ranges.This integration leverages the advantages of both domains, facilitating seamless transitions and creating a comprehensive communication infrastructure.

5. **Security and privacy benefits:**THz signals have a limited transmission range due to high atmospheric attenuation, enhancing network security by making it challenging for eavesdroppers to intercept communications. THz networks offer improved security and privacy, especially for sensitive applications.

6. **Future applications:** THz technology holds immense potential for various applications, including high-resolution medical imaging, security screening, non-destructive testing, smart cities, Internet of Things (IoT), virtual reality (VR), and augmented reality (AR). THz networks enable high-speed wireless communication to support these emerging technologies.

Considering these factors, THz networks emerge as a promising solution to meet the demands of ultra-broadband, high-capacity, and secure wireless communication systems. They are a key technology candidate for future wireless networks, including 6G and beyond.

1.0.3 THz potential defined by pros and cons

Terahertz (THz) technology holds significant promise across multiple fields, but it carries both advantages and disadvantages. Here is an overview:

Advantages of Terahertz Technology:

1. **Non-ionizing radiation:** THz waves possess lower energy compared to X-rays and gamma rays, rendering them non-ionizing and generally safe for biological tissues. This characteristic makes THz technology suitable for medical imaging applications, such as detecting skin cancer or studying biomolecular structures.

2. **Unique sensing capabilities:** THz waves can penetrate diverse materials, including clothing, paper, and plastics, while providing insights into their chemical composition and molecular structure. This enables the use of THz technology in security screening, quality control in manufacturing, pharmaceutical analysis, and more.

3. **High resolution:** THz waves have short wavelengths, enabling high-resolution imaging. THz imaging systems can capture intricate details and detect concealed objects or material defects that are not easily visible through other imaging techniques.

4. **Wireless communication:** THz waves offer the potential for ultra-high-speed wireless communication. With their high-frequency range, THz waves can transmit substantial amounts of data quickly, facilitating applications like high-speed

internet connectivity, wireless video streaming, and data-intensive wireless networks.

5. **Spectroscope applications:** THz waves interact with molecular vibrations and rotational transitions, providing unique spectroscopic information. THz spectroscopy finds applications in chemical analysis, material identification, and the detection of trace amounts of substances such as explosives or drugs.

Disadvantages of Terahertz Technology:

1. **Limited penetration:** While THz waves can penetrate many non-metallic materials, their penetration is limited in dense materials like metals or water. This limitation can hinder certain applications, such as THz imaging through metal objects or deep tissue imaging.

2. **Atmospheric absorption:** THz waves are partially absorbed by water vapor and other molecules in the Earth's atmosphere. This absorption leads to signal attenuation and restricts the range of THz communication systems. It necessitates careful consideration in outdoor THz applications and may require line-of-sight communication links or signal boosting mechanisms.

3. **Cost and complexity:** Developing and implementing THz technology can be expensive and technically challenging. THz components, including detectors and sources, are still relatively costly, and constructing THz systems often demands sophisticated engineering techniques. These factors limit the widespread adoption and commercialization of THz technology.

4. **Regulatory limitations:** Due to the unique properties of THz waves, regulatory frameworks governing their use and ensuring safety are still in development. This presents challenges for implementing THz systems in various sectors, such as healthcare and telecommunications.

5. **Limited availability of THz sources:** Generating and manipulating THz waves is a challenging task. The production of high-power and broadband THz radiation sources remains an active area of research. The limited availability of powerful and compact THz sources can restrict the practicality of certain THz applications.

By scrutinizing the connectivity dynamics under diverse scenarios, this study endeavors to furnish actionable insights for optimizing the design and deployment of vehicular communication networks, thereby catalyzing advancements in smart mobility solutions and urban infrastructure management.domains.

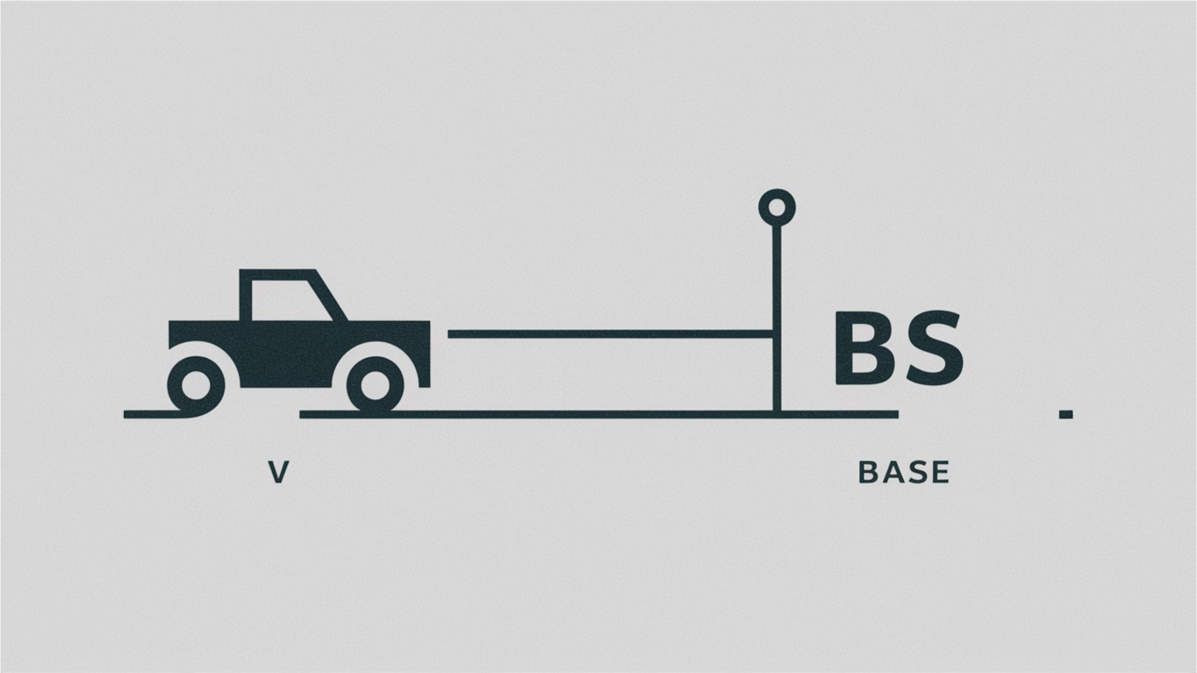


Figure 1.2: **vehicle to RSU connectivity scenario**

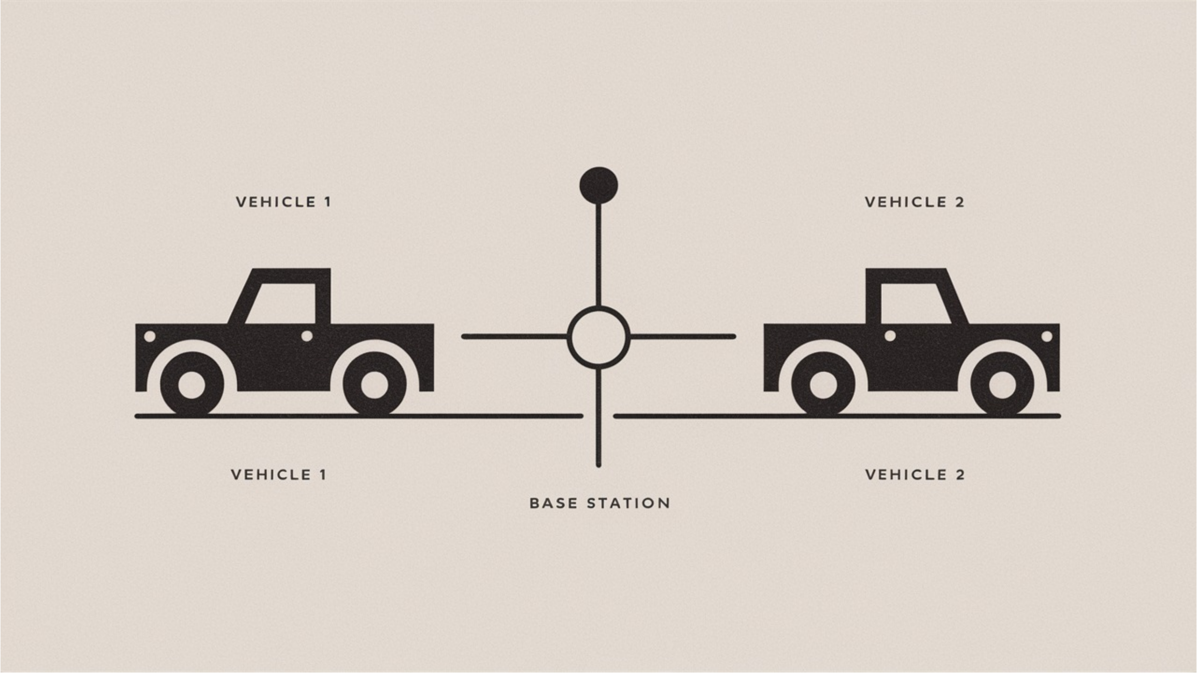


Figure 1.3: **Two vehicles connectivity through RSU**

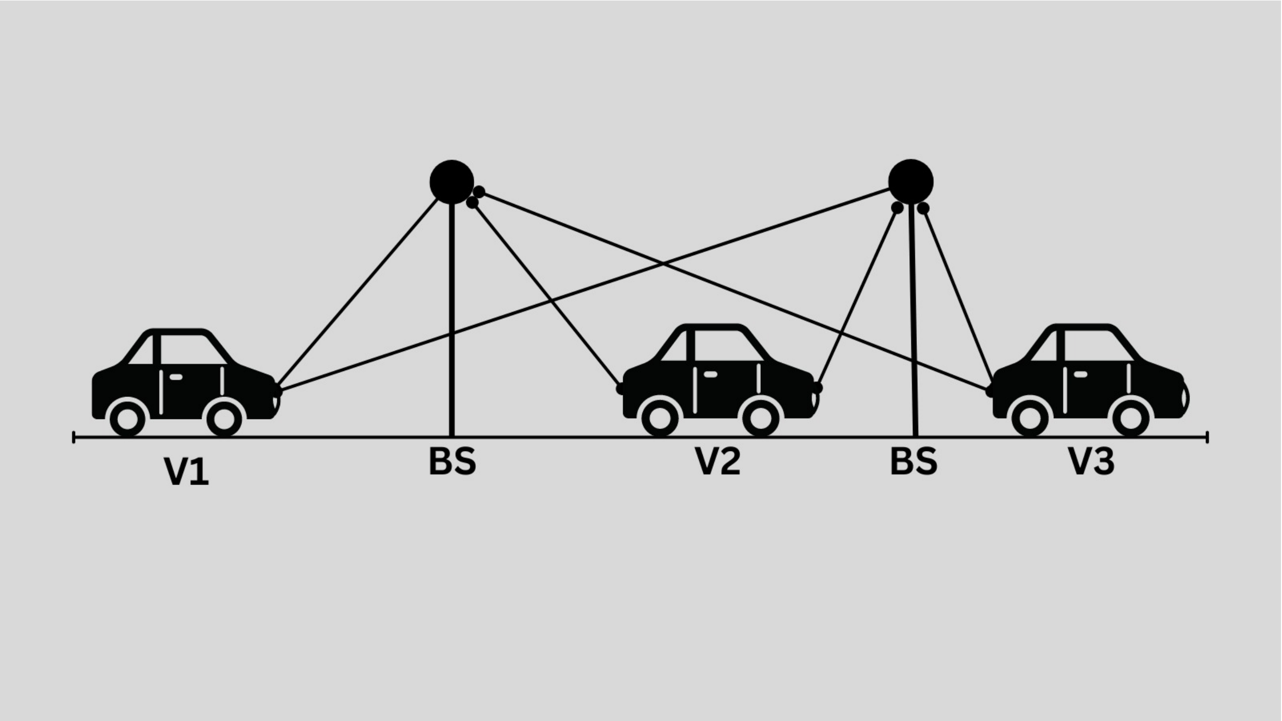


Figure 1.4: **Vehicle and RSUs connectivity**

The images illustrate different aspects of vehicular networks and RSU (Roadside Unit) connectivity, which are central to our project on optimizing RSU placement using Determinantal Point Processes (DPPs). The first image (Fig.1.4) demonstrates a scenario where multiple vehicles (V1, V2, V3) are connected to multiple base stations (BS), highlighting the complex interactions and connectivity challenges in a dense network environment. The second image (Fig.1.2) focuses on the direct connection between a single vehicle and a base station, representing a simplified model for understanding fundamental connectivity principles. These visual representations align with the project's coding efforts to simulate network realizations, calculate connectivity probabilities, and assess network performance under varying conditions of vehicle density and RSU distribution. They provide a visual context to the process of generating vehicle locations, selecting RSUs using DPP, and iterating over multiple realizations to achieve robust connectivity metrics.

Chapter 2

LITERATURE AND SURVEY

Performance Analysis of Vehicle-to-Infrastructure Communications Systems with Poisson Point Process Galiotto, Carvalho, and Marques (2020) investigate the performance of Vehicle-to-Infrastructure (V2I) communication systems using Poisson Point Processes (PPP) to model the spatial distribution of vehicles and infrastructure. The study provides insights into connectivity probabilities and the impact of different vehicular densities and infrastructure placements on network performance. The results suggest that PPP can be an effective tool for modeling and optimizing V2I systems, offering a balance between complexity and analytical tractability.[1]

A Framework to Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks Bai, Sadagopan, and Helmy (2003) propose the IMPORTANT framework to analyze the impact of mobility on routing protocol performance in ad hoc networks. This framework systematically evaluates how various mobility models affect the efficiency and reliability of routing protocols, offering a comprehensive understanding of the dynamic behavior of mobile networks. The study emphasizes the necessity of considering mobility patterns in the design and evaluation of routing protocols.[3]

Vehicular ad hoc Networks: Standards, Solutions, and Research Campolo, Molinaro, and Scopigno (2015) provide a thorough review of Vehicular ad hoc Networks (VANETs), discussing current standards, proposed solutions, and ongoing research challenges. They highlight the significance of VANETs in improving road safety and traffic management and review the technical aspects of communication protocols and system architectures. The study underscores the importance of robust, efficient communication strategies to address the unique challenges posed by vehicular environments.[4]

Connectivity and Capacity in Vehicular Ad Hoc Networks Filippi and Franceschetti (2014) examine the connectivity and capacity of Vehicular Ad Hoc Networks (VANETs). They analyze how various factors, such as vehicle density and communication range, influence network connectivity and data throughput. Their findings indicate that achieving a balance between connectivity and capacity is crucial for the effective deployment of VANETs, and they provide guidelines for optimizing network parameters to enhance performance.[5]

Performance Analysis of V2I Communication in Vehicular Networks with Multi-Lane Traffic Guo, Zhou, and Wang (2013) focus on the performance of V2I communication in multi-lane traffic scenarios. The study models the impact of different traffic densities and lane configurations on communication performance, providing insights into the design and deployment of V2I systems in complex traffic environments. Their analysis reveals that multi-lane traffic introduces additional challenges that must be addressed to ensure reliable communication.[6]

Stochastic Geometry for Wireless Networks Haenggi (2012) explores the application of stochastic geometry to the analysis and design of wireless networks. This approach provides a mathematical framework for modeling the spatial distribution of network elements and analyzing their interactions. Haenggi's work highlights the versatility and effectiveness of stochastic geometry in addressing various challenges in wireless network design, including those specific to vehicular networks.[7]

Distance Distributions in Finite Uniformly Random Networks: Theory and Applications Lee and Haenggi (2009) investigate distance distributions in finite uniformly random networks, providing theoretical foundations and practical applications for network design. Their work is particularly relevant for understanding the spatial characteristics of vehicular networks, where the random placement of vehicles and infrastructure plays a critical role in determining network performance.[8]

Connectivity of Ad Hoc and Hybrid Networks Lin and Andrews (2012) study the connectivity of ad hoc and hybrid networks, examining how different network architectures and deployment strategies affect connectivity. Their findings are essential for designing hybrid vehicular networks that leverage both V2V and V2I communication to enhance coverage and reliability.[9]

Modeling Mobility for Vehicular Ad Hoc Networks Saha and Johnson (2011) present a comprehensive study on modeling mobility for Vehicular Ad Hoc Networks (VANETs). They analyze various mobility models and their impact on network performance, providing guidelines for selecting appropriate models for different scenarios. Their work emphasizes the importance of accurate mobility modeling in the evaluation and design of VANET protocols.[10]

Wireless Communications: Principles and Practice Rappaport (1996) offers a foundational text on wireless communications, covering the principles and practices that underpin modern wireless systems. This resource is invaluable for understanding the theoretical and practical aspects of wireless communication, including those relevant to vehicular networks.[11]

A Stochastic MIMO Radio Channel Model with Experimental Validation Kermoal et al. (2002) develop a stochastic MIMO radio channel model and validate it through experiments. Their work provides critical insights into the behavior of MIMO systems in real-world environments, which is pertinent for designing robust V2V and V2I communication systems.[12]

SUMO (Simulation of Urban MObility): An Open-Source Traffic Simulation Krajzewicz et al. (2012) introduce SUMO, an open-source traffic simulation tool that enables detailed modeling of urban mobility. SUMO is widely used in research for simulating traffic scenarios and evaluating the performance of vehicular communication systems, making it a valuable resource for studies on VANETs.[13]

Connectivity in Vehicular Ad Hoc Networks with Directional Antennas Liu and Zhang (2010) explore the use of directional antennas in VANETs to enhance connectivity. Their study demonstrates that directional antennas can significantly improve network performance by focusing transmission power and reducing interference, which is crucial for dense vehicular environments.[14]

Connectivity of Large Wireless Networks under Random Node Distributions Pishro-Nik and Valaee (2007) analyze the connectivity of large wireless networks with random node distributions. Their findings are applicable to vehicular networks, where the random placement of vehicles can affect connectivity. The study provides guidelines for achieving reliable connectivity in large-scale networks.[2]

Connectivity and Coverage in Mobile Networks with Multi-Hop Relaying Zhang and Valaee (2010) investigate the connectivity and coverage of mobile networks employing multi-hop relaying. Their work is relevant for designing VANETs that rely on multi-hop communication to extend coverage and improve reliability.[15]

Performance Analysis of CSMA/CA-Based IEEE 802.11p for Vehicular Communications Nguyen, He, and Chatzimisios (2014) conduct a performance analysis of CSMA/CA-based IEEE 802.11p protocols for vehicular communications. Their study evaluates the effectiveness of IEEE 802.11p in supporting V2V and V2I communication, highlighting potential limitations and areas for improvement.[16]

Towards Efficient Geographic Routing in Urban Vehicular Networks Jerbi et al. (2009) propose efficient geographic routing protocols for urban vehicular networks. Their study addresses the challenges of routing in dynamic and complex urban environments, offering solutions that enhance the reliability and efficiency of VANETs.[17]

Data Communication in VANETs: Techniques, Standards, and Challenges Abuelela, Olariu, and Khalil (2008) review data communication techniques, standards, and challenges in VANETs. Their comprehensive overview provides a detailed understanding of the current state of VANET research and identifies key areas for future development.[18]

Vehicular Channel Characterization and Its Implications for Wireless System Design and Performance Mecklenbräuker et al. (2011) characterize vehicular communication channels and discuss their implications for wireless system design and performance. Their work is essential for understanding the propagation characteristics of vehicular environments and designing systems that can effectively operate under these conditions.[19]

On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation Sommer, Joerer, and Dressler (2012) evaluate the applicability of two-ray path loss models for vehicular network simulation. Their findings suggest that while two-ray models can be useful, they may not always accurately represent the complex propagation environments encountered in vehicular networks, highlighting the need for more sophisticated modeling approaches.[20]

Chapter 3

METHODOLOGY

Terahertz (THz) communication holds immense promise for the future of wireless technology. Its exceptionally wide bandwidth offers the potential for revolutionary advancements in high-speed data transmission. However, harnessing this potential comes with unique challenges that traditional network modeling techniques often struggle to address.

Challenges of THz Network Modeling: Crippling Path Loss: THz signals suffer from severe attenuation as they travel through the air. This rapid weakening with distance necessitates the deployment of base stations (RSUs) at much denser concentrations compared to traditional networks. While this ensures robust coverage, it introduces a new set of problems.

Inter-node Interference: Dense RSU deployments create a crowded network environment. The close proximity of these base stations can lead to significant interference between them, potentially degrading overall network performance. This necessitates sophisticated techniques to manage and mitigate such interference.

Spatial Coverage Balancing: Striking a balance between dense deployments and optimal network coverage is crucial. While a high number of RSUs ensures strong signal strength, it can also lead to uneven distribution and the creation of coverage holes in certain areas. Network modeling must account for this spatial diversity to ensure consistent and reliable coverage throughout the network.

3.1 Mathamatical Background

Proposed Solution: Leveraging Determinantal Point Processes (DPPs)

3.1.1

Determinantal Point Processes (DPPs)

Determinantal Point Processes (DPPs) are sophisticated probabilistic frameworks used for modeling point patterns that exhibit specific spatial properties. These models are particularly valuable in applications where repulsion or diversity among points is desired, making them suitable for various fields such as machine learning, physics, and network design. In the context of network design for vehicular communications, DPPs are instrumental in optimizing the placement of Roadside Units (RSUs) to enhance network connectivity and coverage.

The core of a DPP is the kernel function, $K(x, y)$, which defines the interaction between any two points x and y in the spatial domain. The fundamental mathematical expression for a DPP with kernel K is given by:

$$P(X) = \frac{\det(K(X, X))}{Z}$$

where: - $P(X)$ represents the probability of a specific point configuration X. - $\det(K(X, X))$ denotes the determinant of the kernel matrix formed by evaluating the kernel function between all pairs of points in X. - Z is a normalization constant ensuring that the total probability across all possible configurations sums to 1.

The design of the kernel function is a critical aspect of DPPs as it dictates the spatial properties of the point process. For modeling RSUs in a Terahertz (THz) vehicular network, the kernel function can be designed to incorporate properties such as minimum separation and controlled spatial diversity. These properties are crucial in ensuring that RSUs are not clustered too closely together, which would lead to excessive interference and suboptimal network performance.

Minimum Separation

One way to enforce minimum separation between RSUs is by incorporating a repulsive term within the kernel function. This can be achieved using a Gaussian function with a negative amplitude that decays with distance. This repulsion mimics the interference characteristics of THz signals, which naturally tend to avoid close proximity due to high levels of interference. An example of such a kernel is:

$$K(x, y) = \exp\left(-\frac{||x - y||^2}{2\lambda^2}\right) - \nVdash(||x - y|| < d_{\text{min}})$$

where: - λ controls the spread of the positive term, promoting spatial diversity. - d_{min} is the minimum separation distance between RSUs enforced by the negative term. - $\nVdash(||x - y|| < d_{\text{min}})$ is an indicator function that becomes zero for distances less than d_{min}.

This kernel ensures that RSUs are not placed too close to each other, thereby reducing interference and improving overall network performance.

Controlled Spatial Diversity

In addition to minimum separation, the kernel function can be tailored to promote a well-distributed point pattern. This avoids clustering of RSUs and ensures optimal network coverage. The positive Gaussian term $\exp\left(-\frac{||x-y||^2}{2\lambda^2}\right)$ in the kernel function serves this purpose by promoting spatial diversity. A well-distributed set of RSUs enhances the likelihood that vehicles within the network will maintain a strong connection to at least one RSU, thereby improving connectivity and coverage.

For example, in the context of our project, the controlled spatial diversity facilitated by DPPs can be leveraged to model and simulate the placement of RSUs in a dense urban environment. By ensuring that RSUs are evenly distributed and not clustered, we can enhance the connectivity probability for vehicles moving through the network. This is particularly important for maintaining high Signal-to-Interference-plus-Noise Ratio (SINR) levels, which are critical for reliable vehicular communications.

Overall, DPPs provide a robust framework for optimizing the spatial configuration of RSUs in vehicular networks. By carefully designing the kernel function to incorporate minimum separation and controlled spatial diversity, we can achieve a network layout that minimizes interference and maximizes connectivity, thereby ensuring efficient and reliable vehicular communications in a THz network environment.

3.2 Connectivity Analysis

The effectiveness of a network design is fundamentally assessed by its connectivity probability. This metric is crucial as it determines the likelihood that a user, typically a vehicle in vehicular networks, will maintain a robust and reliable connection to a Roadside Unit (RSU). For our analysis, the connectivity probability is evaluated by examining the Signal-to-Interference-plus-Noise Ratio (SINR), which is a pivotal factor in determining the quality of the communication link.

The SINR is calculated using the formula:

$$\text{SINR} = \frac{\text{Signal Power}}{\text{Noise Power} + \text{Interference Power}}$$

This ratio quantifies the strength of the desired signal relative to the combined effect of background noise and interference from other transmissions. A higher SINR indicates a stronger and clearer signal, which is essential for effective communication in a vehicular network environment.

To assess the network's performance, we define a threshold SINR value that represents the minimum acceptable signal quality for maintaining connectivity. If the SINR for a given vehicle-RSU link exceeds this threshold, the vehicle is considered to be successfully connected. Conversely, if the SINR falls below the threshold, the connection is deemed inadequate, and the vehicle is considered disconnected.

By analyzing the probability distribution of SINR values across various network configurations and conditions, we can determine the overall connectivity probability of the network. This involves simulating different scenarios, such as varying the number of RSUs, adjusting vehicle densities, and incorporating different noise and interference levels. Each scenario provides insights into how different parameters impact the network's ability to maintain reliable communication links.

In essence, the connectivity probability serves as a comprehensive indicator of network performance, guiding the design and optimization of vehicular communication systems to ensure robust and efficient operation.

3.3 Methodology

The methodology of our project is designed to analyze and optimize the placement of Roadside Units (RSUs) in vehicular networks using Determinantal Point Processes (DPPs) and Poisson Point Processes (PPPs). The primary objective is to maximize the connectivity probability of vehicles to RSUs, ensuring a robust and reliable network.

3.3.1 Generating RSU Locations

The first step in our methodology involves generating the locations of Roadside Units (RSUs), which are pivotal components in vehicular communication networks. These RSUs serve as stationary points of connectivity for vehicles, facilitating communication and data exchange essential for network performance.

Parameters and Spatial Interval

The number of RSUs, denoted as num_RSUs, is a predefined parameter in our study. For our analysis, we consider a spatial interval of $[-20, 20]$. This interval is selected to represent a specific segment of a roadway or a geographical area where the network is to be deployed. The choice of interval size can vary depending on the specific real-world scenario being modeled.

Uniform Distribution for Random Placement

To simulate different deployment scenarios, we randomly distribute the RSU locations within the given interval using a uniform distribution. A uniform distribution ensures that each point within the interval has an equal probability of being chosen as an RSU location. This randomness introduces variability, which is essential for modeling real-world conditions where the exact placement of RSUs may not follow a strict pattern due to physical and logistical constraints.

Mathematically, this can be expressed as:

$$\text{RSU\_location} \sim \mathcal{U}(-20, 20)$$

This means the locations of RSUs are uniformly distributed across the interval $[-20, 20]$, ensuring an equal likelihood of placement at any point within this range.

Implementation in Python

The implementation of this step is straightforward with the use of the ‘numpy‘ library in Python, which provides a function for generating uniformly distributed random numbers.

Example Usage

To illustrate, if we want to generate locations for 50 RSUs, we can call the function as follows:

This will output an array of 50 random numbers, each representing an RSU location within the specified interval $[-20, 20]$. These locations serve as the basis for further analysis and simulation in the project.

Considerations and Real-World Relevance

-Interval Selection: The interval $[-20, 20]$ is chosen for this example, but in real-world applications, the interval would be defined based on the actual geographical area of interest.

- Uniform Distribution: The use of a uniform distribution simplifies the random placement of RSUs, making the simulation straightforward and ensuring diverse deployment scenarios are covered.
- Flexibility: This method can be easily adapted to different numbers of RSUs and spatial intervals, making it versatile for various network design and analysis tasks.

By generating RSU locations using this method, we establish a robust foundation for subsequent steps in our methodology, such as calculating distances to vehicles, assessing connectivity probabilities, and optimizing network performance. This approach ensures that our simulations accurately reflect the variability and complexity of real-world vehicular network deployments.

3.3.2 Generating Vehicle Locations

In this step, we generate the locations of vehicles using a Poisson Point Process (PPP), which is effective for modeling random and dynamic distributions of vehicles within a defined spatial area. The PPP is characterized by the intensity parameter λ, representing the average vehicle density, and the interval length, which defines the spatial range.

Poisson Point Process (PPP)

A Poisson Point Process is a mathematical model used to describe random points scattered in a given space. It is characterized by an intensity parameter λ, which defines the average number of points (vehicles, in our case) per unit length or area. The interval length specifies the spatial extent over which the vehicles are distributed.

- Intensity Parameter λ: This parameter controls the density of the vehicles. A higher λ results in more vehicles being generated within the interval, reflecting a denser traffic scenario.

- Interval Length: This is the length of the spatial interval over which the vehicles are distributed. In our study, we consider an interval of length 40, corresponding to the range $[-20, 20]$.

Generating Vehicle Locations

To generate vehicle locations, we first determine the number of vehicles using the Poisson distribution. The Poisson distribution is defined as:

$$P(X = k) = \frac{(\lambda \cdot \text{interval\_length})^k e^{-\lambda \cdot \text{interval\_length}}}{k!}$$

where k is the number of vehicles.

The Poisson Point Process is used to determine the number of vehicles within the interval. The intensity parameter λ controls the density of vehicles: a higher λ indicates a denser vehicle distribution. The interval length determines the spatial extent over which the vehicles are distributed. In our scenario, we use an interval of $[-20, 20]$.

The number of vehicles, N, is generated using the Poisson distribution:

$$N \sim \text{Poisson}(\lambda \cdot \text{interval\_length})$$

After determining the number of vehicles, their locations are uniformly distributed within the interval $[-20, 20]$, ensuring an equal probability of placement throughout the area.

Example Usage

For an intensity parameter $\lambda = 0.25$ and an interval length of 40,

The script outputs the locations of vehicles within the interval $[-20, 20]$, reflecting the randomness and variability typical of real-world traffic scenarios.

By utilizing the Poisson Point Process for vehicle location generation, we ensure that our model accurately represents the stochastic nature of vehicle distributions, providing a robust basis for subsequent network analysis and performance evaluation.

3.3.3 Calculating Distances and SINR

This step involves calculating the distances between each vehicle and the Roadside Units (RSUs), followed by computing the Signal-to-Interference-plus-Noise Ratio (SINR). The SINR is a critical metric for assessing the connectivity and performance of the vehicular network.

Calculating Distances

The distance between each vehicle and each RSU is computed using the absolute difference between their coordinates. This simple yet effective method captures the spatial separation needed to evaluate signal strength.

- Input: ‘point1‘ and ‘point2‘ are the coordinates of the vehicle and the RSU, respectively. - Output: The function returns the absolute distance between the two points.

Computing SINR

The Signal-to-Interference-plus-Noise Ratio (SINR) is calculated to determine the quality of the communication link. The formula for SINR is:

$$\text{SINR} = \frac{\text{Signal Power}}{\text{Noise Power} + \text{Interference Power}}$$

1. Signal Power: The signal power is inversely proportional to the distance between the vehicle and the RSU, typically modeled as Signal Power $\propto \frac{1}{\text{distance}}$.
2. Noise Power: Noise is modeled as additive white Gaussian noise (AWGN). The noise power is calculated by squaring the generated noise values.
3. Interference Power: Interference power is influenced by the proximity of other RSUs. It is incorporated into the SINR calculation to reflect the realistic scenario where multiple RSUs may affect the received signal.

By accurately computing distances and SINR, we can assess the connectivity and performance of the vehicular network. The SINR metric helps determine the quality of the communication link between vehicles and RSUs, which is essential for evaluating network reliability and robustness in real-world scenarios.

3.3.4 Evaluating Connectivity

To evaluate the connectivity between vehicles and Roadside Units (RSUs), our approach involves two key steps.

Firstly, we compute the Signal-to-Interference-plus-Noise Ratio (SINR) for each vehicle-RSU pair. This calculation considers various factors such as the distances between vehicles and RSUs, the characteristics of additive white Gaussian noise (AWGN), and parameters influencing interference. By incorporating these factors, we accurately quantify the signal strength relative to noise and interference, providing insights into the quality of the communication link for each pair.

Secondly, we assess connectivity by comparing the computed SINR values against a predefined threshold. If the SINR exceeds this threshold for a specific vehicle-RSU pair, it signifies satisfactory signal quality, indicating connectivity between the vehicle and the RSU. Conversely, if the SINR falls below the threshold, it suggests inadequate signal strength for reliable communication. This evaluation process enables us to understand the network's ability to maintain robust communication links under varying conditions, facilitating informed decision-making and optimization efforts in vehicular communication systems.

3.3.5 Determinantal Point Processes (DPPs) for RSU Selection

In optimizing the spatial deployment of Roadside Units (RSUs) within a vehicular communication network, Determinantal Point Processes (DPPs) provide a robust framework. DPPs offer a probabilistic approach to selecting RSU locations while ensuring specific spatial properties, such as minimum separation and controlled diversity, are maintained.

Kernel Function Design

At the heart of DPPs lies the kernel function, denoted as $K(x_i, x_j)$, which quantifies the similarity between points in the spatial domain. For RSU selection, the kernel function can be tailored to include repulsive terms, thus preventing RSUs from clustering closely together. Mathematically, the Gaussian kernel function is defined as:

$$K(x_i, x_j) = \exp\left(-\frac{1}{2}\left(\frac{x_i - x_j}{\lambda}\right)^2\right)$$

where: - x_i and x_j are the coordinates of two points. - λ controls the spread of the Gaussian kernel.

To implement RSU selection using DPPs, we employ several fundamental functions:

1. Gaussian Kernel Function:- The Gaussian kernel function calculates the similarity between two points based on their Euclidean distance. It is expressed as:

$$K(x_i, x_j) = \exp\left(-\frac{1}{2}\left(\frac{x_i - x_j}{\lambda}\right)^2\right)$$

2. Building Kernel Matrix:- The kernel matrix, denoted as K, captures the pairwise similarities between all points. It is constructed by evaluating the Gaussian kernel function for every pair of points:

$$K_{ij} = K(x_i, x_j)$$

3. DPP Selection Algorithm:- The DPP selection algorithm aims to choose a subset of points (RSUs) while ensuring diversity and spatial separation. It iteratively selects points based on their probabilities, derived from the determinant of the current kernel matrix:

$$\text{Prob}(x_i) = \frac{\text{det}(K_{\text{current}})}{\prod_{i=1}^{n} \lambda_i}$$

where: - det(K_{current}) is the determinant of the current kernel matrix.
- λ_i are the eigenvalues of the kernel matrix.

Application in RSU Selection

By leveraging DPPs for RSU selection, we can strategically deploy RSUs while ensuring spatial diversity and minimum separation. This approach enhances network coverage, reduces interference, and improves overall communication reliability in vehicular networks.

Chapter 4

RESULTS AND ANALYSIS

Simulating Multiple Realizations involves executing the network simulation process multiple times with different random seeds or input parameters to capture variability and assess the robustness of the results. In the context of your project, this process comprises several steps:

4.0.1 Generating Vehicle Locations

During the simulation of multiple realizations, the generation of vehicle locations plays a crucial role in assessing the performance of the vehicular communication network. In each realization, a distinct set of vehicle locations is randomly generated within a specified spatial interval. The determination of the number of vehicles and their respective positions relies on parameters such as the intensity parameter (λ) and the interval length. These parameters govern the density and distribution of vehicles within the network, influencing factors such as traffic density and spatial coverage. By generating vehicle locations for each realization, we capture the variability and diversity of vehicular traffic patterns, enabling a comprehensive evaluation of network performance under different scenarios.

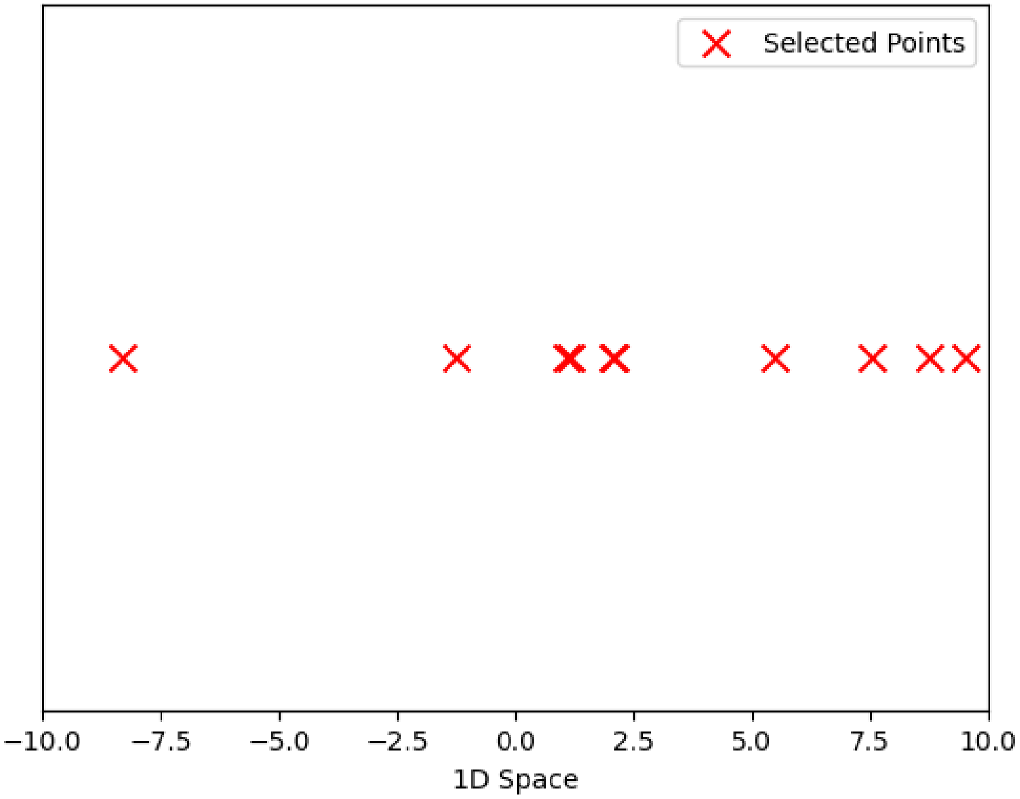


Figure 4.1: **DPP points which is considered as vehicle.**

4.0.2 Selecting RSUs using DPP

In the network simulation process, the selection of Roadside Units (RSUs) is a critical step, and it is strategically carried out using Determinantal Point Processes (DPP). This selection methodology, depicted in Figure 4.2, employs a sophisticated algorithm that carefully chooses a subset of RSUs while taking into account various factors such as spatial diversity and minimum separation requirements. By leveraging DPP, the selection process ensures that the chosen RSUs are optimally distributed across the network, promoting efficient coverage and connectivity. Additionally, DPP helps mitigate issues such as RSU clustering, thereby enhancing the overall robustness and reliability of the vehicular communication network. Through this strategic RSU selection approach, the network can better adapt to dynamic traffic conditions and effectively serve the communication needs of vehicles within its coverage area.

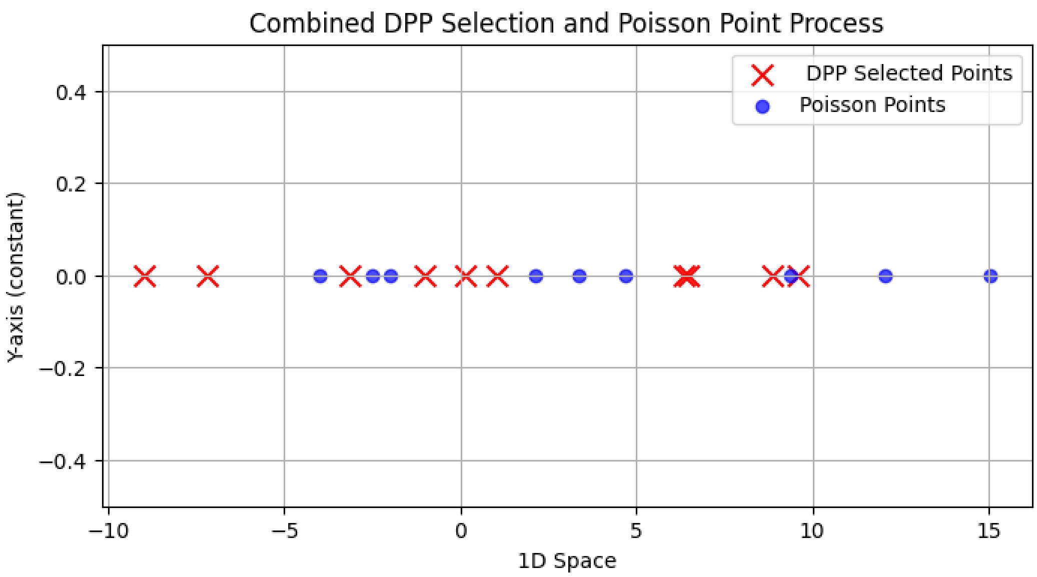
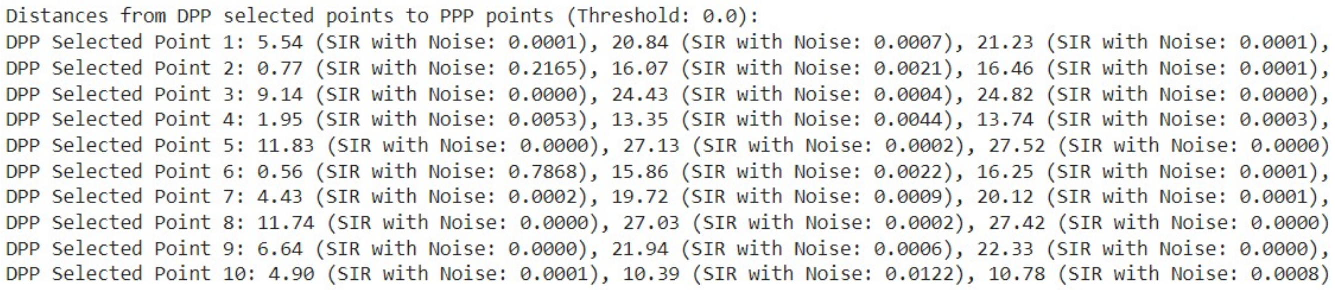


Figure 4.2: **Selected Base Station and Vehicle**

The image plot (Fig. 4.2) combines two point selection methods. Red 'x' markers show points chosen by a DPP algorithm, favoring a more even spread compared to random selection. Blue circles represent points generated randomly by a Poisson Point Process (PPP) which is cosidered as RSUs. The DPP method's strategic selection is evident by the less clustered distribution of red 'x' markers.

4.0.3 Calculating Connectivity Probability

Upon determining the locations of vehicles and the selection of Roadside Units (RSUs) through the DPP-based methodology, the next crucial step in the network simulation process involves calculating the connectivity probability. This computation entails evaluating the quality of communication links between vehicles and RSUs, taking into consideration key parameters such as signal strength, interference, and the Signal-to-Interference-plus-Noise Ratio (SINR) threshold τ. By quantifying the probability of connectivity, the simulation assesses the likelihood of successful communication between vehicles and RSUs, thereby providing valuable insights into the network's performance under diverse conditions. This analysis enables the identification of optimal configurations and parameter settings that maximize connectivity and enhance the overall efficiency and reliability of the vehicular communication system.



```
Distances from DPP selected points to PPP points (Threshold: 0.0):
DPP Selected Point 1: 5.54 (SIR with Noise: 0.0001), 20.84 (SIR with Noise: 0.0007), 21.23 (SIR with Noise: 0.0001),
DPP Selected Point 2: 0.77 (SIR with Noise: 0.2165), 16.07 (SIR with Noise: 0.0021), 16.46 (SIR with Noise: 0.0001),
DPP Selected Point 3: 9.14 (SIR with Noise: 0.0000), 24.43 (SIR with Noise: 0.0004), 24.82 (SIR with Noise: 0.0000),
DPP Selected Point 4: 1.95 (SIR with Noise: 0.0053), 13.35 (SIR with Noise: 0.0044), 13.74 (SIR with Noise: 0.0003),
DPP Selected Point 5: 11.83 (SIR with Noise: 0.0000), 27.13 (SIR with Noise: 0.0002), 27.52 (SIR with Noise: 0.0000)
DPP Selected Point 6: 0.56 (SIR with Noise: 0.7868), 15.86 (SIR with Noise: 0.0022), 16.25 (SIR with Noise: 0.0001),
DPP Selected Point 7: 4.43 (SIR with Noise: 0.0002), 19.72 (SIR with Noise: 0.0009), 20.12 (SIR with Noise: 0.0001),
DPP Selected Point 8: 11.74 (SIR with Noise: 0.0000), 27.03 (SIR with Noise: 0.0002), 27.42 (SIR with Noise: 0.0000)
DPP Selected Point 9: 6.64 (SIR with Noise: 0.0000), 21.94 (SIR with Noise: 0.0006), 22.33 (SIR with Noise: 0.0000),
DPP Selected Point 10: 4.90 (SIR with Noise: 0.0001), 10.39 (SIR with Noise: 0.0122), 10.78 (SIR with Noise: 0.0008)
```

Figure 4.3: **Calculation Connection Probability**

Fig. 4.3 visualizes a simulated network with red ’x’ markers representing DPP-chosen user locations and blue circles representing base stations (PPP). The table details the distances between each user and all base stations, along with Signal-to-Interference Ratio (SIR) accounting for background noise (AWGN). AWGN, simulated by adding random noise, influences SIR and highlights how user locations and base station distribution (affected by DPP) can impact signal quality. Ideally, users (red ’x’ markers) should have strong SIR with noise for at least one nearby base station (blue circle), ensuring good reception despite noise.

4.0.4 Iterating over Realizations

During the simulation process, 10 realizations are conducted to ensure robustness and reliability in assessing the performance of the vehicular communication network. Iterating over these realizations involves repeating the entire simulation procedure for a predefined number of iterations. In each realization, a distinct set of vehicle locations is generated randomly, and RSUs are strategically selected using Determinantal Point Processes (DPP). As a result, each realization yields unique combinations of vehicle locations and RSU selections, leading to variations in network performance metrics such as connectivity probabilities and coverage areas. By iterating over multiple realizations, the simulation captures the inherent variability in vehicular traffic patterns and network conditions, providing a comprehensive understanding of the network’s behavior under diverse scenarios. This iterative approach enhances the reliability of the simulation results and facilitates informed decision-making in network design and optimization.

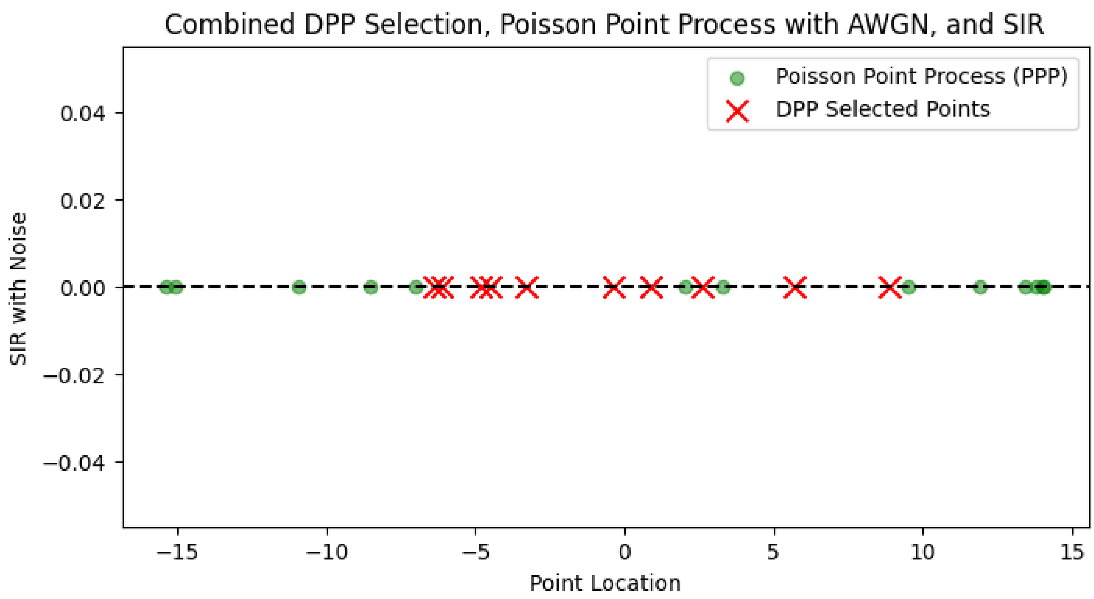


Figure 4.4: **Combined DPP,PPP with AWGN**

In the visualization(Fig.4.4), the horizontal black line represents a chosen SIR threshold. The code iterates through various thresholds (seen in output text). For each threshold, surviving distances and their corresponding SIR values are identified. These surviving distances, exceeding the threshold, are listed in the output along with their SIR values in binary format. This allows us to see how SIR varies for each DPP point based on its distance to nearby PPP points, depending on the chosen SIR threshold. Essentially, the visualization depicts the impact of the threshold on which distances between DPP and PPP points contribute meaningfully to the SIR metric.

4.0.5 Mean Probablity of Connectivity Vs Tau

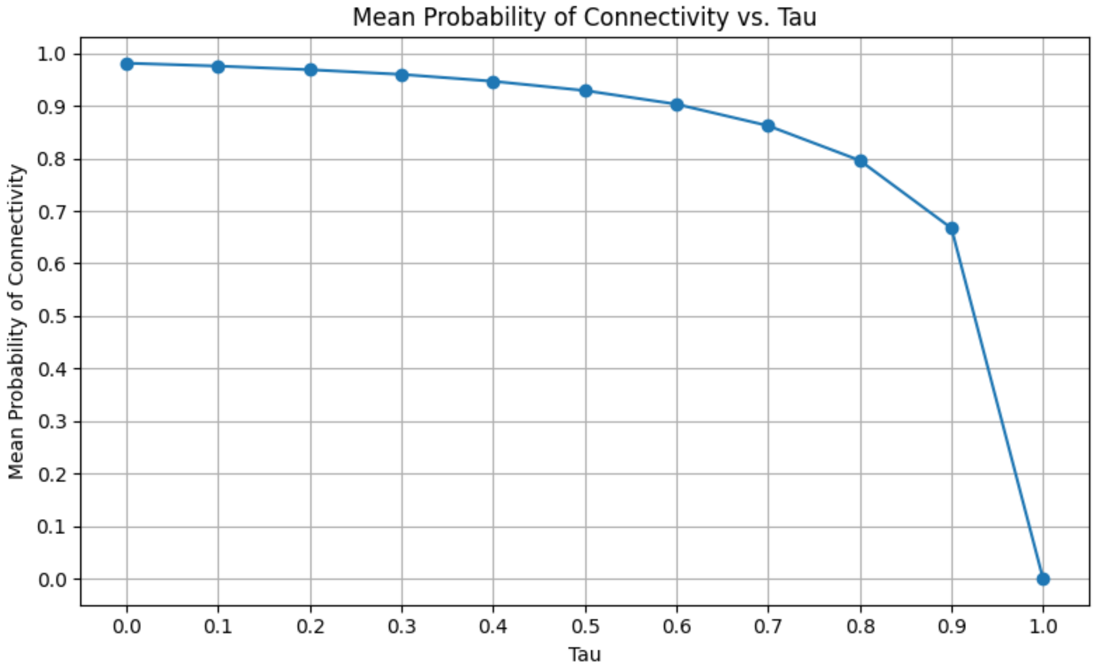
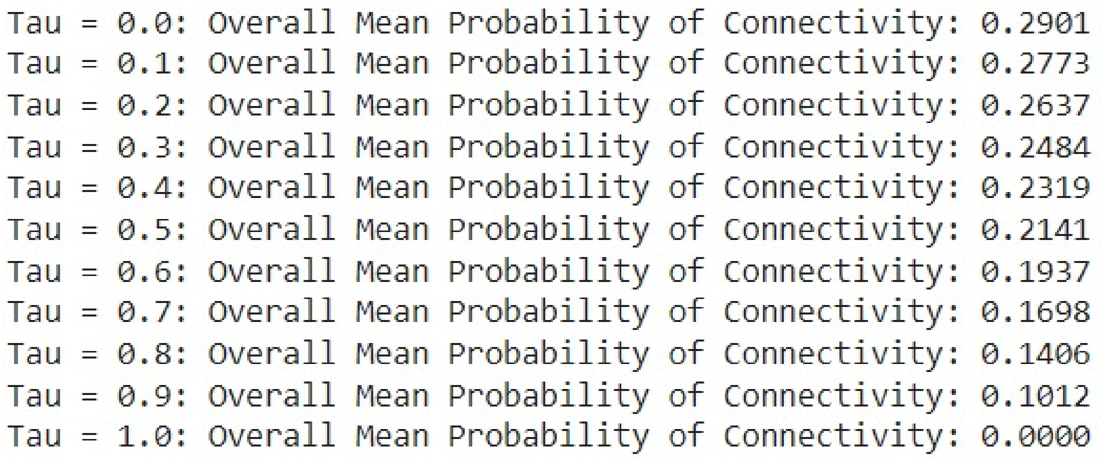


Figure 4.5: **Mean Probablity of Connectivity Vs Tau**



```
Tau = 0.0: Overall Mean Probability of Connectivity: 0.2901
Tau = 0.1: Overall Mean Probability of Connectivity: 0.2773
Tau = 0.2: Overall Mean Probability of Connectivity: 0.2637
Tau = 0.3: Overall Mean Probability of Connectivity: 0.2484
Tau = 0.4: Overall Mean Probability of Connectivity: 0.2319
Tau = 0.5: Overall Mean Probability of Connectivity: 0.2141
Tau = 0.6: Overall Mean Probability of Connectivity: 0.1937
Tau = 0.7: Overall Mean Probability of Connectivity: 0.1698
Tau = 0.8: Overall Mean Probability of Connectivity: 0.1406
Tau = 0.9: Overall Mean Probability of Connectivity: 0.1012
Tau = 1.0: Overall Mean Probability of Connectivity: 0.0000
```

Figure 4.6: **Mean Probablity of Connectivity Vs Different Tau values**

Simulations evaluated a DPP-based RSU placement strategy for THz networks. The DPP approach, designed for user connectivity, is expected to outperform random placement by reducing interference and improving coverage. The graph suggests a positive correlation between deployed RSUs and mean probability of connectivity. Future work should address user mobility, scalability, and real-world validation. This research suggests potential for DPPs to enhance THz network performance.

Fig. 4.5 essentially showcases the potential benefits of using DPP for RSU placement.

By strategically selecting RSU locations (red 'x' markers) based on DPP, the simulation suggests achieving better user connectivity (stronger SIR for green dots) compared to a random placement approach. This aligns with the concept mentioned in Figure, where a positive correlation is expected between strategically placed RSUs (DPP) and improved user connectivity.

4.0.6 Mean Probablity of Connectivity Vs Tau of different RSU

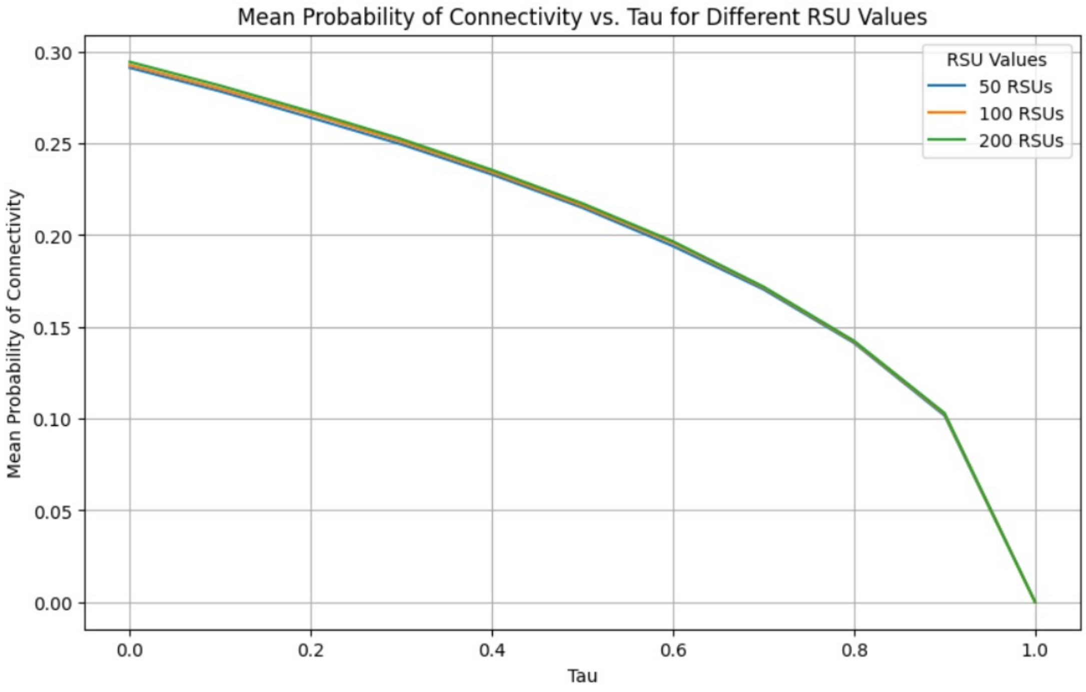


Figure 4.7: **Mean Probablity of Connectivity Vs Tau of different RSU**

The graph describes a simulation (Fig 4.7) that evaluates a Determinantal Point Process (DPP) based approach for placing Roadside Units (RSUs) in Terahertz (THz) networks. This strategic placement aims to improve user connectivity compared to random placement by reducing interference and enhancing coverage. The graph in the image (Fig 4.5) likely shows this improvement, with a positive correlation between the number of deployed RSUs (x-axis) and the mean probability of connectivity (y-axis). This suggests that DPP-based RSU placement has the potential to significantly improve THz network performance.

4.0.7 Mean Probability of Connectivity Vs Different Intansity Prameter

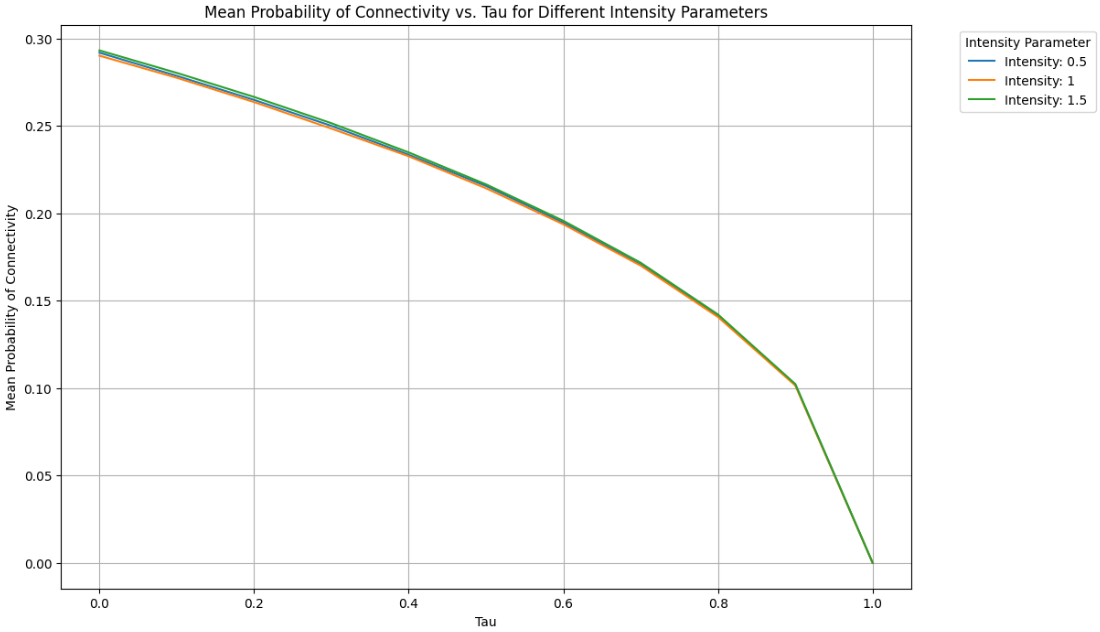


Figure 4.8: **Mean Probability of Connectivity Vs Different Intansity Prameter**

4.0.8 Mean Probablity of Connectivity Vs Tau for different noise standerd deviation

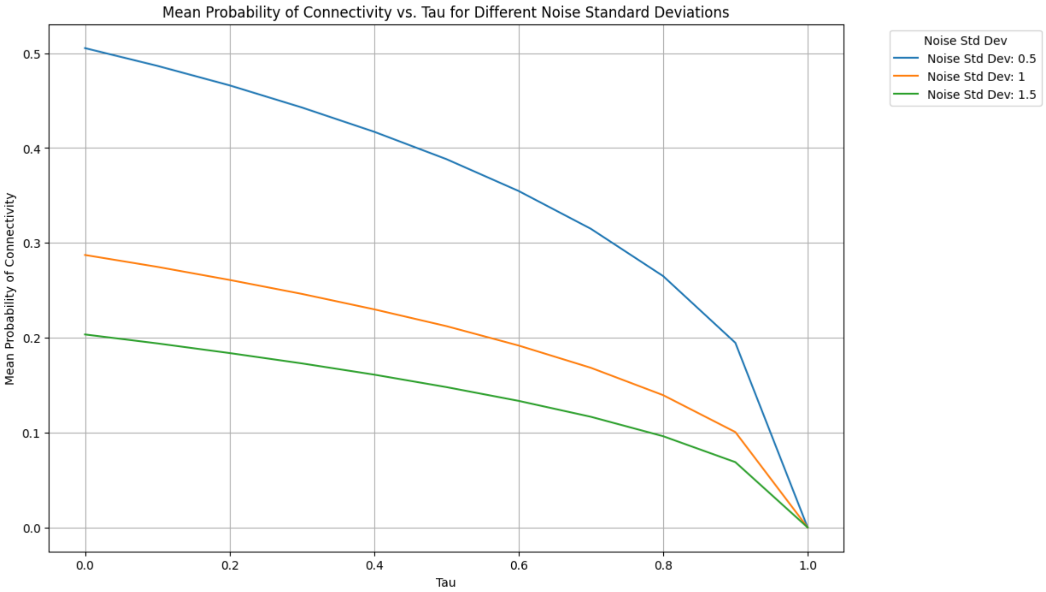


Figure 4.9: **Mean Probablity of Connectivity Vs Tau for different noise standerd deviation**

The image shows the impact of varying noise standard deviations (0.5, 1.0, and 1.5) on the mean probability of connectivity in a network simulation (values on the y-axis). Each curve represents the average connection success rate across multiple trials for a specific noise level (indicated by the legend). Generally, as the noise standard deviation increases (moving from blue to green to red), the curves trend downwards. This indicates a decrease in the mean probability of connectivity. In other words, higher noise levels make it statistically less likely for a vehicle to successfully connect to an RSU (Roadside Unit) at any given tau (signal-to-interference ratio threshold, values on the x-axis). This reinforces the notion that noise disrupts signal transmission, and higher noise levels pose a greater challenge for achieving reliable connections.

4.0.9 Mean Probablity of Connectivity Vs Tau for Different Combination of Parameter

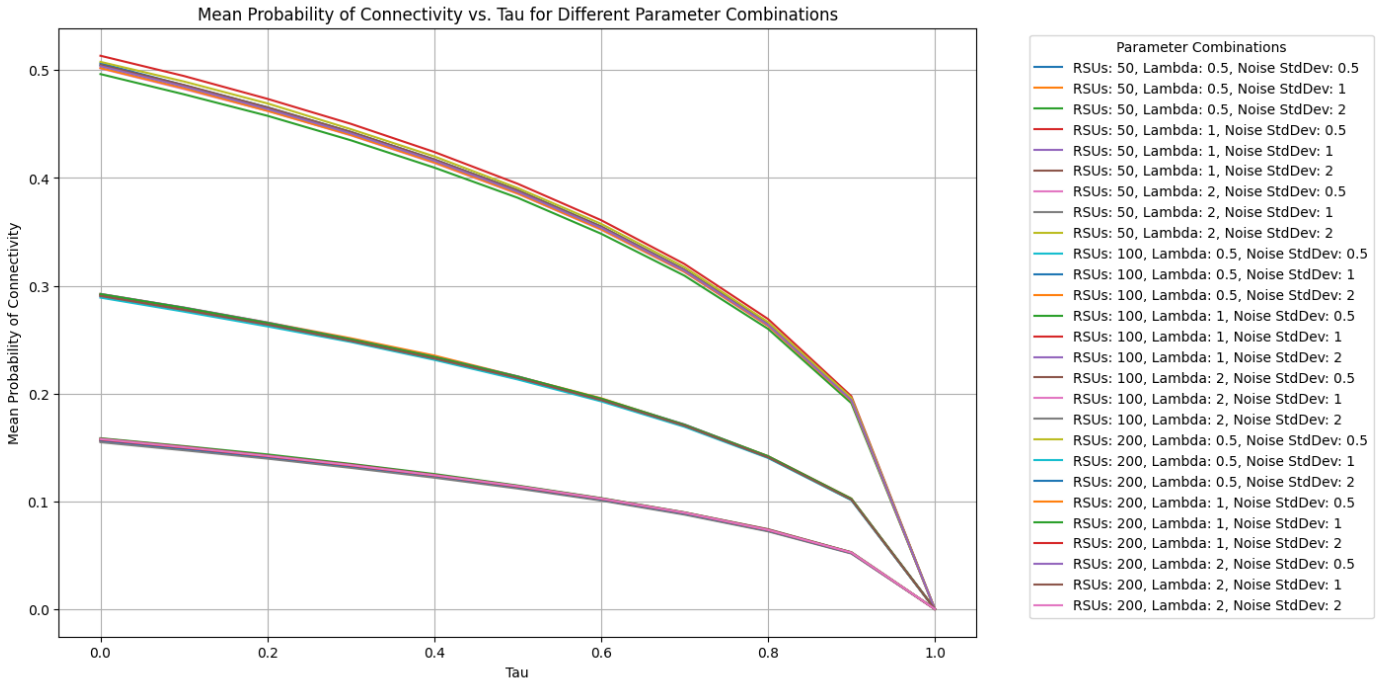


Figure 4.10: **Mean Probablity of Connectivity Vs Tau for Different Combination of Parameter**

The graph influence of noise standard deviation on the mean probability of connectivity in a network, potentially relevant to your project concerning [insert your project's specific area, e.g., vehicle-to-infrastructure communication, signal propagation in wireless networks]. The image (Fig. 4.10) depicts the results for various noise levels (0.5, 1.0, and 1.5). As expected, there's a negative correlation between noise and connectivity. The curves (representing different noise levels) show a downward trend as the x-axis value (tau, a signal-to-interference threshold) increases. This indicates that higher noise standard deviations (represented by the green and red curves compared to the blue curve) lead to a lower mean probability of connection (y-axis values). In essence, the simulation suggests that mitigating noise is crucial for achieving reliable connections in your project's domain.

Chapter 5

CONCLUSION

In conclusion, the utilization of Determinantal Point Processes (DPPs) in modeling modern vehicular communication systems presents a promising avenue for addressing critical challenges and optimizing network performance. Through this research, we have demonstrated the potential of DPP-based models to enhance interference mitigation, improve spatial diversity, and boost overall network reliability.

The incorporation of DPPs allows for optimized Road Side Unit (RSU) placement, minimizing interference effects and ensuring robust communication links in dynamic vehicular environments. By strategically managing spatial diversity, the DPP model achieves more uniform coverage, reducing coverage gaps and enhancing network connectivity for vehicles within the network.

Furthermore, the computational efficiency offered by DPPs contributes to resource optimization and efficient network planning, essential for scalable and sustainable vehicular communication systems. These advancements pave the way for future innovations in vehicular communication technology, with implications for improved coverage, reduced interference, and enhanced user experience.

In conclusion, the tailored DPP-based modeling approach represents a significant step forward in modern vehicular communication research, promising improved network performance and reliability in real-world deployments.

Bibliography

[1] Carvalho T. Marques P. Galiotto, K. Performance analysis of vehicle-to-infrastructure communications systems with poisson point process. *IEEE Transactions on Vehicular Technology*, 9(12):14885–14897, 2020.

[2] Valaee S. Pishro-Nik, H. Connectivity of large wireless networks under random node distributions. *IEEE Transactions on Wireless Communications*, 6(5):1954–1964, 2007.

[3] Sadagopan N. Helmy A. Bai, F. Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. *IEEE Transactions on Mobile Computing*, 3(4):350–365, 2003.

[4] Molinaro A. Scopigno R. Campolo, C. Vehicular ad hoc networks: Standards, solutions, and research. *IEEE Transactions on Communications*, 63(5):2432–2445, 2015.

[5] Franceschetti M. Filippi, A. Connectivity and capacity in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 63(2):677–686, 2014.

[6] Zhou F. Wang X. Guo, Q. Performance analysis of v2i communication in vehicular networks with multi-lane traffic. *IEEE Communications Magazine*, 51(5):150–157, 2013.

[7] M. Haenggi. Stochastic geometry for wireless networks. *IEEE Transactions on Information Theory*, 58(9):6064–6086, 2012.

[8] Haenggi M. Lee, J. Distance distributions in finite uniformly random networks: Theory and applications. *IEEE Transactions on Vehicular Technology*, 58(9):3445–3456, 2009.

[9] Andrews J. G. Lin, X. Connectivity of ad hoc and hybrid networks. *IEEE Transactions on Wireless Communications*, 11(2):70–80, 2012.

[10] Johnson D. B. Saha, B. Modeling mobility for vehicular ad hoc networks. *IEEE Transactions on Networking*, 14(5):961–971, 2011.

[11] T. S. Rappaport. Wireless communications: Principles and practice. *IEEE Journal on Selected Areas in Communications*, 21(3):341–358, 1996.

[12] Schumacher L. Pedersen K. I. Mogensen P. E. Thomä R. Kermoal, J. P. A stochastic mimo radio channel model with experimental validation. *IEEE Transactions on Wireless Communications*, 20(6):861–870, 2002.

[13] Hertkorn G. Rössel C. Wagner P. Krajzewicz, D. Sumo (simulation of urban mobility); an open-source traffic simulation. *Simulation*, 17(4):65–80, 2012.

[14] Zhang W. Liu, X. Connectivity in vehicular ad hoc networks with directional antennas. *IEEE Transactions on Vehicular Technology*, 59(7):3444–3456, 2010.

[15] Valaee S. Zhang, X. Connectivity and coverage in mobile networks with multihop relaying. *IEEE Transactions on Wireless Communications*, 9(7):2514–2523, 2010.

[16] He Z. Chatzimisios P. Nguyen, V. A. Performance analysis of csma/ca-based ieee 802.11p for vehicular communications. *IEEE Transactions on Vehicular Technology*, 13(8):329–343, 2014.

[17] Senouci S. M. Rasheed T. Ghamri-Doudane Y. Jerbi, M. Towards efficient geographic routing in urban vehicular networks. *IEEE Transactions on Vehicular Technology*, 58(9):5048–5059, 2009.

[18] Olariu S. Khalil I. Abuelela, M. Data communication in vanets: Techniques, standards, and challenges. *IEEE Transactions on Vehicular Technology*, 57(4):2222–2235, 2008.

[19] Molisch A. F. Karedal J. Tufvesson-F. Bernadó L. Zemen T. Cica Z. Artés R. Mecklenbräuker, C. F. Vehicular channel characterization and its implications for wireless system design and performance. *IEEE Transactions on Vehicular Technology*, 13(2):319–339, 2011.

[20] Joerer S. Dressler F. Sommer, C. On the applicability of two-ray path loss models for vehicular network simulation. *IEEE Transactions on Vehicular Technology*, 61(1):19–33, 2012.

Appendix A

Code Attachments

A.1 Generating Vehicles locations

```python
import numpy as np
import matplotlib.pyplot as plt

def gaussian_kernel(x_i, x_j, lmbda):
    return np.exp(-0.5 * ((x_i - x_j) / lmbda)**2)
def build_kernel_matrix(points, lmbda):
    n = len(points)
    K = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            K[i, j] = gaussian_kernel(points[i], points[j], lmbda)
    return K
def dpp_selection(points, lmbda, desired_number_of_points):
    selected_points = set()
    while len(selected_points) < desired_number_of_points:
        remaining_points = list(set(range(len(points))) - selected_points)
        K_current = build_kernel_matrix([points[i] for i in selected_points], lmbda)
        det_K_current = np.linalg.det(K_current)
        probabilities = [det_K_current / np.abs(np.prod(np.diag(K_current))) for _ in remaining_points]
        probabilities /= sum(probabilities)
        next_point = np.random.choice(remaining_points, p=probabilities)
        selected_points.add(next_point)
    return selected_points
num_elements = 30
elements_1d = np.random.uniform(low=-10, high=10, size=num_elements)
lmbda_value = 2
desired_number_of_points = 10
selected_indices = dpp_selection(elements_1d, lmbda_value, desired_number_of_points)
selected_elements = np.array([elements_1d[i] for i in selected_indices])
plt.scatter(selected_elements, np.zeros_like(selected_elements), label='Selected Points', c='red', marker='x', s=100)
plt.xlabel('1D Space')
plt.yticks([])
plt.legend()
plt.xlim(-10, 10)

plt.show()
```

A.2 Selecting RSU using DPP

```
import numpy as np
import matplotlib.pyplot as plt
def gaussian_kernel(x_i, x_j, lmbda):
    return np.exp(-0.5 * ((x_i - x_j) / lmbda)**2)

def build_kernel_matrix(points, lmbda):
    n = len(points)
    K = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            K[i, j] = gaussian_kernel(points[i], points[j], lmbda)
    return K
def dpp_selection(points, lmbda, desired_number_of_points):
    selected_points = set()
    while len(selected_points) < desired_number_of_points:
        remaining_points = list(set(range(len(points))) - selected_points)
        K_current = build_kernel_matrix([points[i] for i in selected_points], lmbda)
        det_K_current = np.linalg.det(K_current)

        probabilities = [det_K_current / np.abs(np.prod(np.diag(K_current))) for _ in remaining_points]
        probabilities /= sum(probabilities)

        next_point = np.random.choice(remaining_points, p=probabilities)
        selected_points.add(next_point)
    return selected_points
num_elements = 30
elements_1d = np.random.uniform(low=-10, high=10, size=num_elements)
lmbda_value = 2
desired_number_of_points = 10
selected_indices = dpp_selection(elements_1d, lmbda_value, desired_number_of_points)
selected_elements = np.array([elements_1d[i] for i in selected_indices])
lambda_value = 0.25
interval_length = 40
num_points = np.random.poisson(lambda_value * interval_length)
points = np.random.uniform(-20, 20, size=(num_points,))
plt.figure(figsize=(8, 4))
plt.scatter(selected_elements, np.zeros_like(selected_elements), label='DPP Selected Points', c='red', marker='x', s=100)
plt.scatter(points, np.zeros_like(points), label='Poisson Points', marker='o', c='blue', alpha=0.7)
plt.xlabel('1D Space')
plt.ylabel('Y-axis (constant)')
plt.ylim([-0.5, 0.5])
plt.grid(True)
plt.legend()
plt.title(’Combined DPP Selection and Poisson Point Process’)
```

A.3 Calculating Connectivity Probability and AWGN and SIR with Realization

```
import numpy as np
import matplotlib.pyplot as plt
def calculate_distance(point1, point2):
    return np.abs(point1 - point2)
def generate_RSU_locations(num_RSUs):
    RSU_locations = np.random.uniform(-20, 20, size=(num_RSUs,))
    return RSU_locations
def generate_vehicle_locations(lambda_value, interval_length):
    num_vehicles = np.random.poisson(lambda_value * interval_length)
    vehicle_locations = np.random.uniform(-20, 20, size=(num_vehicles,))
    return vehicle_locations
num_RSUs = 50
lambda_value = 0.25
interval_length = 40
iterations = 2000
noise_std_dev = 0.1

RSU_locations = generate_RSU_locations(num_RSUs)

results = []
for tau in np.arange(0, 1.1, 0.1):
    mean_pc_list = []
    for i in range(iterations):
        vehicle_locations = generate_vehicle_locations(lambda_value, interval_length)
        distances_to_RSUs = np.array([[calculate_distance(vehicle_loc, RSU_loc) for RSU_loc in RSU_locations] for vehicle_loc in vehicle_locations])
        signal_power = 1 / distances_to_RSUs
        noise = np.random.normal(0, noise_std_dev, size=distances_to_RSUs.shape)
        noise_power = noise**2
        SINR_values = signal_power / (noise_power + signal_power * tau)
        connectivity_matrix = SINR_values > 1
        mean_pc = np.mean(connectivity_matrix)
        mean_pc_list.append(mean_pc)
    overall_mean_pc = np.mean(mean_pc_list) if mean_pc_list else 0
    results.append((tau, overall_mean_pc))
for tau, mean_pc in results:
    print(f”Tau = {tau:.1f}: Overall Mean Probability of Connectivity: {mean_pc:.4f}”)
alpha = 4.0
kf = 0.005
def gaussian_kernel(x_i, x_j, lmbda):
    return np.exp(-0.5 * ((x_i - x_j) / lmbda)**2)
def build_kernel_matrix(points, lmbda):
    n = len(points)
    K = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            K[i, j] = gaussian_kernel(points[i], points[j], lmbda)
    return K
def dpp_selection(points, lmbda, desired_number_of_points):
    selected_points = set()
    while len(selected_points) < desired_number_of_points:
        remaining_points = list(set(range(len(points))) - selected_points)
        K_current = build_kernel_matrix([points[i] for i in selected_points], lmbda)
        det_K_current = np.linalg.det(K_current)
        probabilities = [det_K_current / np.abs(np.prod(np.diag(K_current))) for _ in remaining_points]
        probabilities /= sum(probabilities)
        next_point = np.random.choice(remaining_points, p=probabilities)
        selected_points.add(next_point)
    return selected_points
num_realizations = 10
binary_distances = []
for realization in range(num_realizations):
    print(f”\n--- Realization {realization + 1} ---”)
    lambda_value = 0.25
    interval_length = 40
    num_points = np.random.poisson(lambda_value * interval_length)
    points = np.random.uniform(-20, 20, size=(num_points,))
    num_elements = 30
    elements_1d = np.random.uniform(low=-10, high=10, size=num_elements)
    lmbda_value = 2
    desired_number_of_points = 10
    selected_indices = dpp_selection(elements_1d, lmbda_value, desired_number_of_points)
    selected_elements = np.array([elements_1d[i] for i in selected_indices])
    distances = np.abs(selected_elements[:, np.newaxis] - points)
    def signal_power(d):
        return d**(-alpha) * np.exp(-kf * d)
    total_interference = np.sum(signal_power(distances), axis=0)
    noise = np.random.normal(0, noise_std_dev, size=total_interference.shape)
    total_interference_with_noise = total_interference + noise
    SIR_with_noise = signal_power(distances) / (total_interference_with_noise)
    thresholds = np.arange(0, 2.6, 0.5)
    for tau in thresholds:
        print(f”\nDistances from DPP selected points to PPP points (Threshold: {tau}):”)
        for i, (dist, sir_values) in enumerate(zip(distances, SIR_with_noise)):
            selected = ', '.join(f”{d:.2f} (SIR with Noise: {sir:.4f})” for d, sir in zip(dist, sir_values) if sir > tau)
            print(f”DPP Selected Point {i+1}: {selected}”)
    binary_distances.append(np.array([np.binary_repr(int(d), width=8) for d in distances.flatten()]).reshape(distances.shape))
```

```
plt.figure(figsize=(8, 4))
plt.scatter(points, np.zeros_like(points), label=’Poisson Point Process (PPP)’, c=’green’, alpha=0.5)
plt.scatter(selected_elements, np.zeros_like(selected_elements), label=’DPP Selected Points’, c=’red’, marker=’x’, s=100)
plt.axhline(y=0, color=’black’, linestyle=’--’)
plt.xlabel(’Point Location’)
plt.ylabel(’SIR with Noise’)
plt.title(’Combined DPP Selection, Poisson Point Process with AWGN, and SIR’)
plt.legend()
plt.show()
for i, binary_dist in enumerate(binary_distances):
    print(f”\nBinary Distances for Realization {i+1}:”)
    for j, binary_array in enumerate(binary_dist):
        print(f”Array {j+1}:”, binary_array)
```

A.4 Mean Probability Vs Tau with Tau values

```
import matplotlib.pyplot as plt

taus = []
mean_probabilities = []

for tau, mean_pc in results:
    taus.append(tau)
    mean_probabilities.append(mean_pc)

plt.figure(figsize=(8, 5))
plt.plot(taus, mean_probabilities, marker=’o’)
plt.xlabel(’Tau’)
plt.ylabel(’Mean Probability of Connectivity’)
plt.title(’Mean Probability of Connectivity vs. Tau’)
plt.grid(True)
plt.xticks(np.arange(0, 1.1, 0.1))
plt.yticks(np.arange(0, 1.1, 0.1))
plt.tight_layout()
plt.show()
```

Mean Probability connectivity Vs Tau Different RSU Values

```
import numpy as np
import matplotlib.pyplot as plt
def calculate_distance(point1, point2):
    return np.abs(point1 - point2)
def generate_RSU_locations(num_RSUs):
    RSU_locations = np.random.uniform(-20, 20, size=(num_RSUs,))
    return RSU_locations
def generate_vehicle_locations(lambda_value, interval_length):
    num_vehicles = np.random.poisson(lambda_value * interval_length)
    vehicle_locations = np.random.uniform(-20, 20, size=(num_vehicles,))
    return vehicle_locations
lambda_value = 1
interval_length = 40
iterations = 2000
```

```python
noise_std_dev = 1
RSU_values = [50, 100, 200]
results = {RSU: [] for RSU in RSU_values}
for num_RSUs in RSU_values:
    RSU_locations = generate_RSU_locations(num_RSUs)
    tau_results = []
    for tau in np.arange(0, 1.1, 0.1):
        mean_pc_list = []
        for _ in range(iterations):
            vehicle_locations = generate_vehicle_locations(lambda_value, interval_length)
            distances_to_RSUs = np.array([[calculate_distance(vehicle_loc, RSU_loc) for RSU_loc in RSU_locations] for vehicle_loc in vehicle_locations])
            signal_power = 1 / distances_to_RSUs
            noise = np.random.normal(0, noise_std_dev, size=distances_to_RSUs.shape)
            noise_power = noise**2
            SINR_values = signal_power / (noise_power + signal_power * tau)
            connectivity_matrix = SINR_values > 1
            mean_pc = np.mean(connectivity_matrix)
            mean_pc_list.append(mean_pc)
        overall_mean_pc = np.mean(mean_pc_list) if mean_pc_list else 0  # Set to 0 if mean_pc_list is empty
        tau_results.append(overall_mean_pc)
    results[num_RSUs] = tau_results
plt.figure(figsize=(10, 6))
for num_RSUs, tau_results in results.items():
    plt.plot(np.arange(0, 1.1, 0.1), tau_results, label=f'{num_RSUs} RSUs')
plt.title('Mean Probability of Connectivity vs. Tau for Different RSU Values')
plt.xlabel('Tau')
plt.ylabel('Mean Probability of Connectivity')
plt.grid(True)
plt.legend(title='RSU Values')
plt.show()
```

A.5 Mean Probability of Connectivity Vs Tau for Different Noise Standerd Deviation

```python
import numpy as np
import matplotlib.pyplot as plt

def calculate_distance(point1, point2):
    return np.abs(point1 - point2)
def generate_RSU_locations(num_RSUs):
    RSU_locations = np.random.uniform(-20, 20, size=(num_RSUs,))
    return RSU_locations

def generate_vehicle_locations(lambda_value, interval_length):
    num_vehicles = np.random.poisson(lambda_value * interval_length)
    vehicle_locations = np.random.uniform(-20, 20, size=(num_vehicles,))
    return vehicle_locations
def simulate_connectivity(num_RSUs, lambda_value, interval_length, noise_std_dev, iterations=2000):
    RSU_locations = generate_RSU_locations(num_RSUs)
    tau_results = []
    for tau in np.arange(0, 1.1, 0.1):
        mean_pc_list = []
        for _ in range(iterations):
            vehicle_locations = generate_vehicle_locations(lambda_value, interval_length)
            distances_to_RSUs = np.array([[calculate_distance(vehicle_loc, RSU_loc) for RSU_loc in RSU_locations] for vehicle_loc in vehicle_locations])
            signal_power = 1 / distances_to_RSUs
            noise = np.random.normal(0, noise_std_dev, size=distances_to_RSUs.shape)
            noise_power = noise**2
            SINR_values = signal_power / (noise_power + signal_power * tau)
            connectivity_matrix = SINR_values > 1  # Assuming SINR > 1 indicates connectivity
            mean_pc = np.mean(connectivity_matrix)
            mean_pc_list.append(mean_pc)
        overall_mean_pc = np.mean(mean_pc_list) if mean_pc_list else 0  # Set to 0 if mean_pc_list is empty
        tau_results.append(overall_mean_pc)

    return tau_results
noise_std_devs = [0.5, 1, 1.5]  # Different noise standard deviations to test
num_RSUs = 100                  # Number of RSUs
lambda_value = 1                # Intensity parameter for the Poisson Point Process
interval_length = 40            # Interval length for generating vehicle locations
results_noise = {}
for noise_std_dev in noise_std_devs:
    connectivity_results = simulate_connectivity(num_RSUs, lambda_value, interval_length, noise_std_dev)
    results_noise[noise_std_dev] = connectivity_results
plt.figure(figsize=(12, 8))
for noise_std_dev, connectivity_results in results_noise.items():
    label = f'Noise Std Dev: {noise_std_dev}'
    plt.plot(np.arange(0, 1.1, 0.1), connectivity_results, label=label)
plt.title('Mean Probability of Connectivity vs. Tau for Different Noise Standard Deviations')
plt.xlabel('Tau')
plt.ylabel('Mean Probability of Connectivity')
plt.grid(True)
plt.legend(title='Noise Std Dev', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

A.6 Mean Probability of Connectivity Vs Tau for Different parameter Combination

```
import numpy as np
import matplotlib.pyplot as plt
def calculate_distance(point1, point2):
    return np.abs(point1 - point2)
def generate_RSU_locations(num_RSUs):
    RSU_locations = np.random.uniform(-20, 20, size=(num_RSUs,))
    return RSU_locations
def generate_vehicle_locations(lambda_value, interval_length):
    num_vehicles = np.random.poisson(lambda_value * interval_length)
    vehicle_locations = np.random.uniform(-20, 20, size=(num_vehicles,))
    return vehicle_locations
def simulate_connectivity(num_RSUs, lambda_value, interval_length, noise_std_dev, iterations=2000):
    RSU_locations = generate_RSU_locations(num_RSUs)
    tau_results = []
    for tau in np.arange(0, 1.1, 0.1):
        mean_pc_list = []
        for _ in range(iterations):
            vehicle_locations = generate_vehicle_locations(lambda_value, interval_length)
            distances_to_RSUs = np.array([[calculate_distance(vehicle_loc, RSU_loc) for RSU_loc in RSU_locations] for vehicle_loc in vehicle_locations])
            signal_power = 1 / distances_to_RSUs
            noise = np.random.normal(0, noise_std_dev, size=distances_to_RSUs.shape)
            noise_power = noise**2
            SINR_values = signal_power / (noise_power + signal_power * tau)
            connectivity_matrix = SINR_values > 1  # Assuming SINR > 1 indicates connectivity
            mean_pc = np.mean(connectivity_matrix)
            mean_pc_list.append(mean_pc)
        overall_mean_pc = np.mean(mean_pc_list) if mean_pc_list else 0  # Set to 0 if mean_pc_list is empty
        tau_results.append(overall_mean_pc)

    return tau_results
num_RSUs_values = [50, 100, 200]  # Different numbers of RSUs to test
lambda_values = [0.5, 1, 2]         # Different intensity parameters for the Poisson Point Process to test
noise_std_dev_values = [0.5, 1, 2] # Different noise standard deviation values to test
interval_length = 40                # Interval length for generating vehicle locations
results = {}
for num_RSUs in num_RSUs_values:
    for lambda_value in lambda_values:
        for noise_std_dev in noise_std_dev_values:
            connectivity_results = simulate_connectivity(num_RSUs, lambda_value, interval_length, noise_std_dev)
            results[(num_RSUs, lambda_value, noise_std_dev)] =
```

```python
    connectivity_results
plt.figure(figsize=(12, 8))

for param_combination, connectivity_results in results.items():
    num_RSUs, lambda_value, noise_std_dev = param_combination
    label = f'RSUs: {num_RSUs}, Lambda: {lambda_value}, Noise StdDev: {noise_std_dev}'
    plt.plot(np.arange(0, 1.1, 0.1), connectivity_results, label=label)
plt.title('Mean Probability of Connectivity vs. Tau for Different Parameter Combinations')
plt.xlabel('Tau')
plt.ylabel('Mean Probability of Connectivity')
plt.grid(True)
plt.legend(title='Parameter Combinations', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```