UNIVERSITÀ DEGLI STUDI DI PADOVA
DEPARTMENT OF INFORMATION ENGINEERING
PH.D. IN INFORMATION ENGINEERING
CURRICULUM "SCIENCE AND INFORMATION TECHNOLOGY"
XXXV CYCLE

ANALYSIS OF SYSTEM WIDE OPTIMISATION SCHEMES BASED ON ANTICIPATORY TECHNIQUES FOR NEXT GENERATION NETWORKS

Coordinator
PROF. A. NEVIANI
UNIVERSITÀ DI PADOVA

Ph.D. Candidate
ANAY AJIT DESHPANDE

Supervisor
PROF. A. ZANELLA
UNIVERSITÀ DI PADOVA

Academic Year 2022/2023

To those
who believe in me.

Abstract

Future wireless networks would be responsible to integrate large number of devices with variety of Quality of Service (QoS)/Quality of Experience (QoE) requirements. The sheer volume of devices arises a need to anticipate the behaviour of these devices so as to design a system capable of satisfying the QoS/QoE requirements. Due to this reason, in the recent years, anticipatory networking has become one of the core research fields in wireless communication. Anticipatory Networking signifies the information that can be predicted, the techniques that can be used to predict such information and the optimization techniques that can potentially utilize such predictions to improve the performance of the wireless network. However, designing anticipatory networking systems becomes increasingly challenging when considering the variability in application scenarios envisioned for future wireless networks. So, analysis of anticipatory networking techniques taking into account the system wide performance for variety of wireless communication scenarios is very crucial. Hence, to obtain a complete view of anticipatory networking in wireless networks, we present a brief overview of the works partaken in the literature and then explain the multiple works undertaken pertaining to the design and analysis of anticipatory networking techniques from a system wide perspective at Physical, Network, Transport and Application Layers.

Sommario

Le reti wireless del futuro dovranno integrare un gran numero di dispositivi con diversi requisiti di qualità del servizio (QoS)/qualità dell'esperienza (QoE). L'enorme quantità di dispositivi fa sorgere la necessità di anticiparne il comportamento per progettare un sistema in grado di soddisfare i requisiti di QoS/QoE. Per questo motivo, negli ultimi anni, l'anticipazione delle reti è diventata uno dei principali campi di ricerca nel settore delle comunicazioni wireless. Per rete anticipatrice si intendono le informazioni che possono essere previste, le tecniche che possono essere utilizzate per prevedere tali informazioni e le tecniche di ottimizzazione che possono potenzialmente utilizzare tali previsioni per migliorare le prestazioni della rete wireless. Tuttavia, la progettazione di sistemi di rete anticipatori diventa sempre più impegnativa se si considera la variabilità degli scenari applicativi previsti per le future reti wireless. Pertanto, l'analisi delle tecniche di rete anticipatrici che tengono conto delle prestazioni dell'intero sistema per una varietà di scenari di comunicazione wireless è molto importante. Per questo motivo, al fine di ottenere una visione completa della rete anticipatrice nelle reti wireless, presentiamo una breve panoramica dei lavori presenti in letteratura e spieghiamo i molteplici lavori intrapresi per la progettazione e l'analisi delle tecniche di rete anticipatrici da una prospettiva di sistema a livello fisico, di rete, di trasporto e di applicazione.

Contents

List of Figures

List of Tables

Acronyms

ALQ Average Link Quality
ANN Artificial Neural Network
AoA Angle of Arrival
AODV Ad-Hoc On Demand Distance Vector Routing
AP Access Point
ARIMA Auto Regression Integrated Moving Average

BAR Beam Aware Stochastic Multihop Routing for FANETs
BBR bottleneck bandwidth and round-trip propagation time
BS Base Station

CF Collaborative Filtering
CL-IDA Cross-layer Interference and Delay Aware Metric
CNN Convolutional Neural Network
CoMP Coordinated Multipoint
CSI Channel State Information
CSMA Carrier Sense Multiple Access

D2D device-to-device
DBNG Deep Belief Network and Gaussian Model
DBR Distance-Based Routing
DNN Deep Neural Network
DPC Dirty Paper Coding
DRL Deep Reinforcement Learning

ESN Echo State Network
ETT Expected Transmission Time
ETX Expected Transmission Count

FANET Flying Ad-Hoc Network
FDD Frequency Division Duplexing
FIFO First In First Out
FNN Feed Forward Neural Network
FoV Field-of-View

GBDT Gradient Boosting Decision Tree
GBSM Geometry based Stochastic Model
GECR Genetic algorithm-based energy-efficient clustering and routing algorithm
GMM Gaussian Mixed Model
GPR Gaussian Process Regression
GPS Global Positioning System
GPU Graphics Processing Unit

HCRAN Heterogeneous Cloud Radio Access Network
HGMM Hidden Gaussian Markov Model
HMM Hidden Markov Model
HRS Hierarchical Rate Splitting

IoT Internet of Things
IOV Internet of Vehicles

JBR Joint Beamforming and Routing in FANETs
JND Just Noticeable Difference

JSDM Joint Spatial Division Multiplexing

KNN K-Nearest Neighbors

LIDAR Light Detection and Ranging
LoS Line of Sight
LSTM Long Short-Term Memory

MANET Mobile Ad-Hoc Network
MBF Matched Beamforming
MC Multi-Connectivity
MDP Markov Decision Process
MF Matrix Factorization
MIMO Multiple Input, Multiple Output
MIP Mixed Integer Programming
ML Machine Learning
MLR Maximum-local-rate
mmWave Milimeter Wave
MPGR Mobility based Geographic Routing Protocol
MPTCP Multipath TCP

NB Naive Bayes
NLoS Non-Line-of-Sight
NN Neural Network
NOMA Non-Orthogonal Multiple Access

OLSR Optimized Link State Routing
OS Operating System

PDF Probability Distribution Function
PDR Packet Delivery Ratio
PPP Poisson Point Process

QKP Quadratic Knapsack Problem
QoE Quality of Experience
QoS Quality of Service
QUIC-EST QUIC-Enabled Scheduling and Transmission

RBF-NN Radial Basis Function Neural Network
RL Reinforcement Learning
RLRP Radio Link Reliability Prediction
RNN Recurrent Neural Network
RS Rate Splitting
RSMA Rate Splitting Multiple Access
RSSI Received Signal Strength Indicator
RSU Road Side Unit
RTT Round Trip Time
RZF Regularized Zero Forcing

SBS Small Base Station
SC Superposition Coding
SCTP Stream Control Transmission Protocol
SDMA Spatial Division Multiple Access
SDN Software-Defined Networking
SDS-TWR Symmetric Double Sided Two Way Ranging
SIC Successive Interference Cancellation
SINR Signal to Interference plus Noise Ratio
SISO Single-Input Single-Output
SMS Short Message Service
SMURF Stochastic Multipath Routing for FANETs
SNR Signal-to-Noise ratio
SVD Singular Value Decomposition
SVM Support Vector Machine
SVR Support Vector Regression

TCP Transmission Control Protocol
TDMA Time Division Multiple Access
TDOA Time Difference of Arrival
TOA Time of Arrival
TPU Tensor Processing Unit

UAV Unmanned Aerial Vehicle
UDP User Datagram Protocol
ULA Uniform Linear Array
UPA Uniform Planar antenna Array
UWB Ultra Wide Band

V2X Vehicle-to-Everything
VANET Vehicular Ad-Hoc Network
VoI Value of Information

WLAN Wireless Local Area Network
WMN Wireless Mesh Network
WSN Wireless Sensor Network

1 Introduction

Design of next generation wireless networks needs to combine different communication technologies for a variety of applications thereby leading to a high number of devices needed to be integrated and handled efficiently. Hence, the need for network optimization in next generation wireless networks, such as Beyond 5G and 6G, has become very important to integrate a large number of devices with different application requirements in conjunction with advanced resource management algorithms to maintain an appreciable quality of service and quality of experience for all users. Satisfying these requirements with traditional optimization schemes would result in high computational costs to find optimal solutions for different types of applications. In order to handle the largely variable requirements, prediction and estimation of wireless network state parameters, such as channel state, user data requirement, user mobility, are necessary to determine optimal network configurations for a certain application at certain time for a certain user. Hence, to predict the network behaviour, the system must continuously track the network and device specific parameters, such as Channel State Information (CSI), user mobility, etc.

These network and device specific parameter predictions in wireless networks, however, are a huge challenge, due to the inherent randomness of the wireless systems. Additionally, the goal of network parameter prediction is to devise networking scenarios while accounting for the different moving parts of a wide and complex system and optimizing a particular function by exploiting information coming from other parts of the system.

This basic idea of system wide focus for optimization can be explored in a variety of different ways. Hence, in this thesis, we provide a brief overview of the works that have been explored in such directions for different wireless communication scenarios. We then highlight the possibility of improving the performance of a wireless communication system by exploiting exogenous information, such as the channel state information and geographical position of the wireless nodes as well as network specific information, such as available capacity. We consider such information to jointly optimize communication performance for novel spatial access schemes such as Rate Splitting, for dynamic networking scenarios such as Unmanned Aerial Vehicle (UAV) networks and novel communication

scenarios such as tactile internet and autonomous driving which have to account for available communication resources to satisfy the overall Quality of Service (QoS)/Quality of Experience (QoE) requirements of the system.
Hence, the goal of the thesis is to introduce the concept of anticipatory networking and the different facets such as anticipatory information, anticipatory techniques and anticipatory optimisation and then to provide a detailed description of works undertaken in anticipatory networking from the physical layer to the application layer. So, the rest of the thesis is structured as follows. In Chapter 2, we provide a detail overview of different works partaken in anticipatory networking. We present the different works that highlight anticipatory information obtainable in wireless networks, associated techniques that can be used to obtain such information and works that present the usage of anticipatory techniques at different layers. In Chapter 3, we presents a physical layer clustering mechanism for Joint Spatial Division Multiplexing (JSDM) in wireless networks. The clustering mechanism determines optimal clusters that maximises the spatial performance obtained for each user in the area while significantly reducing computational complexity with respect to the state of the art. Chapters 4, 5 and 6 highlights the different works undertaken for communication in Flying Ad-Hoc Networks (FANETs). FANETs refer to a network of UAVs that are deployed to accomplish a specific task or function. In these works, we focus on application of FANETs for tracking and monitoring in military and disaster related scenarios. In Chapter 4, we provide a background of FANETs and present a brief literature review regarding communication design for FANETs for tracking and monitoring applications and the related challenges in FANETs. In Chapter 5, we present works regarding route establishment, reliability and stability that take into account mobility prediction of the UAVs for tracking applications in FANETs. In Chapter 6, we extend the works in 5 by introducing beamforming design for routing in FANETs thereby providing a joint beamforming and routing design based on mobility prediction of UAVs. We highlight different works that design beamforming and routing based on mobility prediction of the UAVs to maximise the overall performance over the communication route. In Chapter 7, we shift the focus to a novel information scheduling problem that needs to take into account the predicted network capacity to maximise the QoE for future applications such as tactile internet and autonomous driving. Lastly, Chapter 8 provides an overall conclusion for the thesis.

Anticipatory Techniques for Wireless Network Optimization

2

Next generation wireless networks attempt to combine different types of devices with varying QoS/QoE requirements. To satisfy these requirements, there is a need to anticipate different types of information that can be gathered from the wireless network and define optimisation techniques to satisfy the application requirements. In this chapter, we present the contextual information obtainable in wireless networks and the anticipatory techniques that can harness such information to define optimisation techniques for various applications across the different layers.

2.1 Contextual Information for Wireless Networks

Wireless Networks incorporate a ton of contextual information which can be very useful for their optimization. User information such as position and mobility as well as Received Signal Strength Indicator (RSSI) and Channel State Information (CSI) provide essential information which can be used by the wireless systems to allocate resources and improve performance of the wireless network.

2.1.1 Geographical Context

Geographical context refers to the user information related to the physical environment. Prominently geographical context includes the position and mobility information regarding the users in the network, and also their trajectories. The difference between mobility and trajectory is that mobility refers to the prediction of the next position of the user, while trajectory refers to the prediction of the whole path of the mobile user. This information is essential while determining the configuration of the network and the requirements of the users so as to provide maximum QoS and QoE to the users.

2.1.1.1 Position Prediction

Position and mobility information is one of the most important contextual information to configure and manage a wireless system. Traditionally, the users periodically communicate their position information to the network infrastructure. But, with increase in the number of devices in the future wireless networks,

these periodic updates may become difficult and may lead to loss in performance for the users. Also, the performance of the system depends on the accuracy of the position updates as well. To accommodate the increase in the number of devices as well as the accuracy of the position updates, the research has focussed on predicting the future position of the users, and on developing mobility models so as to predict the movement of the users in the network.

To predict positions, multiple different methods have been proposed, from traditional algorithms such as Kalman filter based prediction schemes, to novel data driven models such as machine learning, deep learning and reinforcement learning based prediction schemes. In [1], a Deep Reinforcement Learning (DRL) algorithm for unsupervised location prediction is presented. The algorithm models the position of the Internet of Things (IoT) devices as a Markov Decision Process (MDP) and trains a DRL agent using RSSI gathered from the IoT devices to predict unknown position of the devices. In [2], the position of users is determined based on the phase, time of arrival, angle of arrival and signal strength of the mmWave signals. It also presents an idea of using a machine learning algorithm to learn the signal strength, time of arrival and angle of arrival values and predict the location of the users. Following that, in [3], a map based position estimation mechanism is presented based on the attributes similar to those used in the previous work, i.e., angle of arrival and time of arrival, but also adds a multipath component which exploits the multipath nature of mmWave signals in the environment. Based on these information, a map of the environment is created by the system and the location of the user is determined based on the values obtained for the considered attributes. In [4], the authors use a temporal convolutional networks to determine the position estimates based on mmWave beam tracking. The temporal convolutional networks takes into account the mmWave beam signals from the user over a period of time and provides position estimates based on the previous position and trajectory. In [5], each node calculates an area of coverage based on the probability of existence for the node to be in its coverage range. In [6], the authors created a dataset based on the RSSI and a random mobility model and trained a deep neural network to predict the future position of the users based on the RSSI. In [7], a dual Hidden Markov Model (HMM) is proposed. The first HMM predicts the position based on the WiFi RSSI and provides that as an input to the second HMM to determine the optimal Access Point (AP) based on the position estimate. Other approaches for the position and location prediction in different types of networks can be found in the surveys [8, 9].

2.1.1.2 Mobility and Trajectory Prediction

Mobility and Trajectory prediction enables the system to determine future positions of the user and optimize the system based on this information. To determine trajectory information about a user, the system has to take into account past positions over a period of time and predict the trajectory based on those positions. In [10], the position estimates are provided using Kalman filters to determine the position and trajectory of the vehicles in the network and find the optimal next hop for routing. In [11], a Gaussian Mixed Model (GMM) is proposed which takes into account the position information over time from Global Positioning System (GPS) coordinates and determines the trajectory. It also incorporates the noise in the position information by modeling the position with a Gaussian Distribution. A maximum likelihood algorithm is used to estimate the motion based on the position information and a Gaussian Process Regression (GPR) is used to predict the trajectory of the vehicle based on the trajectory data. In [12], two Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) and Echo State Network (ESN), are used to determine the trajectory for predicting resource allocation. In [13], user movement is predicted based on the GPS coordinates and timestamps collected from Twitter and training an ESN based on the position to determine the trajectory of the users. In [14], a LSTM is proposed to predict future position and using dead reckoning method to predict the trajectory based on the predicted position.

2.1.1.3 Uncertainties in Predictions

Prediction algorithms discussed previously are not always 100 % accurate, i.e., uncertainties exist in the prediction algorithms. These uncertainties are important to take into account, as large uncertainties in the predictions make them impossible to be used meaningfully for network optimizations. So it is important to take that into account, especially when prediction accuracy can be very crucial for optimizing the network. In [10], the location prediction is determined by Kalman filter. But Kalman filter predictions have uncertainty. To tackle that, a mean squared error is calculated to determine the quality of the prediction. In [15], the uncertainty is predicted using deep neural networks in wireless fingerprinting based on dead reckoning. In [16], the uncertainty in three dimensional localization, based on RSSI, Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AoA) and Symmetric Double Sided Two Way Ranging (SDS-TWR), is calculated based on sensitivity factor determined by partial differential equations. Based on the uncertainty calculated, the propagation of uncertainty in the position from each of the factors is determined.

2.1.2 Communication Link Context

Communication Link Context refers to the contextual information regarding the physical and abstract attributes of the wireless medium. Primarily, communication link refers to the channel over which the communication takes place, which is a physical attribute, as well as the quality of the channel or link, which is an abstract attribute based on the physical attribute. The link context information is crucial in wireless communication due to highly dynamic nature of wireless channels which leads to performance variations that are critical in a network containing a large number of devices.

2.1.2.1 Channel State Prediction

To determine the channel state, the system needs to gather continuous CSI updates from the user. But continuous updates increase the overhead especially for a network containing a large number of devices. In [17], the CSI is obtained for the uplink channel and then used to predict the state in the downlink channel. To this end, a deep neural network called SCNet is proposed which takes into account the uplink channel state for a Frequency Division Duplexing (FDD) Multiple Input, Multiple Output (MIMO) system and predicts the downlink channel state. In [18], an LSTM is proposed to predict the next CSI based on the previous CSI. The LSTM structure is trained on the dataset created by gathering CSI obtained from a Rayleigh fading channel and then used to predict the next channel states based on previous channel states in a rolling prediction fashion (previous predictions are used to determine the next predictions). In [19], a combination of Convolutional Neural Network (CNN) and LSTM called OCEAN is used to predict CSI. A combination of frequency band, location, time, temperature, humidity, and weather conditions is used to train the network offline (historical data) and online (current data) and provides an online prediction for the channel state. In [20], a combination of Feed Forward Neural Network (FNN) and Radial Basis Function Neural Network (RBF-NN) is used to predict the channel parameters. The network is trained using an input of Transmitter (Tx) and Receiver (Rx) coordinates, Tx-Rx distance and carrier frequency, which are determined using real time measurements and a Geometry based Stochastic Model (GBSM). In [21], an LSTM is used to predict the channel state. The training dataset for the LSTM model is gathered by using the IEEE802.11p in-phase and quadrature-phase signals to gather the channel information using algorithms such as down sampling, frame detection, symbol alignment, frequency offset correction and training sequence extraction. The LSTM model is trained

using the channel information obtained and then used to predict channel quality in vehicular networks.

2.1.2.2 Link State Prediction

Link State is an abstract attribute that is computed by Layers 2 and above to incorporate the quality of the physical channel in system design for different applications. Hence, link state prediction is crucial especially when determining the link quality to calculate metrics for data transmission so as to achieve an optimal network performance. In [22], a routing metric is proposed based on link quality estimation based on RSSI and Average Link Quality (ALQ), which is determined based on past measurements. Based on the link estimation, the next hop is chosen for routing in vehicular ad-hoc networks. In [23], the physical link is modelled based on the transmission energy consumption and a Mixed Integer Programming (MIP) optimization problem is presented which is solved by a deep neural network. In [24], a prediction algorithm is presented that anticipates link quality metric such as Expected Transmission Count (ETX) and Expected Transmission Time (ETT) to optimise the Optimized Link State Routing (OLSR) protocol. The link quality metric is predicted based on signal strength, which in turn is predicted based on linear regression. In [25], the Cross-layer Interference and Delay Aware Metric (CL-IDA) is estimated using multiple different machine learning techniques, such as multiple linear regression, support vector regression and Gaussian regression, and used as a metric for OLSR. In [26], a link quality state model is presented and based on the link quality model, a Kalman filter based link reliability prediction algorithm called Radio Link Reliability Prediction (RLRP) is presented which predicts the bounds of the reliability of existence of the link for the network.

2.1.3 Traffic Context

Traffic Context refers to the data traffic that is encountered in the wireless networks. Accurately predicting traffic demands in the network can be really crucial to satisfy the QoS and QoE requirements of the users, especially in a network containing a large number of devices.

2.1.3.1 Traffic Prediction

In [27], a traffic prediction algorithm called Coca-Predict based on traffic correlation and causality is presented. The algorithm uses correlation between traffic flows over time and over different cells in the area as well as traffic flows due to mobility and social events to predict the traffic at a particular place and time. The algorithm uses Auto Regression Integrated Moving Average (ARIMA), Deep Neural Network (DNN), Gradient Boosting Decision Tree (GBDT) and LightGBM

[28]. In [29], an exponential smoothing is used to predict the traffic generated by the nodes in each duty cycle in Wireless Sensor Network (WSN). In [30], a Deep Belief Network and Gaussian Model (DBNG) is used to predict the traffic. The DBNG is used to predict the network traffic, which is modelled based on the previous traffic information and fluctuations in a Wireless Mesh Network (WMN). In [31], a Spatial-Temporal Cross-domain Neural Network (STC-Net) is presented which is trained via transfer learning, by using traffic information from clusters in the cellular network. In [32], a combination of a CNN and a parameter estimator is used to predict city-wide traffic. The parameter estimator is based on the distance between two cells in the cellular network of the city. In [33], a Gaussian Process based predictor is presented, which takes live data from the radio access networks and predicts traffic for a load-aware management tasks.

2.1.3.2 Performance Analysis based on Traffic Analysis

Traffic prediction is not the only thing that can be considered to improve the system. Prediction of the performance for a certain type of traffic is also necessary. Performance prediction provides a good indication of the network configurations for a certain type of traffic. This helps in determining the network configurations that can be used in different scenarios. In [34], throughput is predicted based on the maximisation of the local and global optimization problems, which address the traffic local to the user and as a whole in the network respectively. The throughput prediction is used to devise a scheduling based on Carrier Sense Multiple Access (CSMA) threshold. In [35], a channel throughput predictor is devised, which takes into account the video streaming demand of the users in the cellular network and caches information at the nearest edge to obtain the required QoE. The predictor is devised as a optimization problem that takes into account channel utilization and buffer occupancy of the user to calculate the maximum throughput. In [36], a throughput predictor is designed based on device level data such as application and Operating System (OS) information and network level data such as channel quality and cell load in cellular network. The throughput predictor is designed using three different machine and deep learning techniques namely random forest, Support Vector Machine (SVM) and LSTM. In [37], a machine learning based prediction algorithm is presented to predict average throughput for video streaming applications. To predict the average throughput, the RSRP, RSRQ values for primary and neighbouring cell and SNR and CQI. In [38], the throughput prediction is designed for IEEE 802.11 networks. The predictor takes into account the relationship between two metrics which is devised from interference offered by multiple access points using directional antennas and omni-directional antennas.

2.1.4 Social Context

Social Context refers to the social information about the user in the wireless network. Social information contains the information regarding user interests and content preferences, mobility trends for the users and concentration of user-defined content in the network.

2.1.4.1 Social Context based caching

Caching refers to the temporary storage of content at the edge of the network so as to reduce load on the core networks for multiple redundant requests of the same content. Social context-based caching is necessary as it stores information locally, based on the user interests and other geographical and network specific aspects such as location and mobility of the users in the network. In [39], Proactive Caching-based Mobility Prediction (PCMP) is proposed, which predicts the nearest Road Side Unit (RSU) for caching information based on user content requests and mobility in Vehicular Ad-Hoc Networks (VANETs). The predictor is designed using an LSTM that takes into account the mobility information and predicts the next RSU and proactively caches information requested by the user. In [40], a logical topology is defined by combining mobile users based on their behaviours so as to create a mobile ad-hoc cloud network and optimize the network delay for computing operations. In [41], a Reinforcement Learning (RL) scheme is proposed to take into account the mobility in VANET. A deep Q algorithm is used to devise the optimal caching and computing allocation scheme based on the mobility of the users in the network. In [42], a social prediction algorithm using HMM based on user behaviour in social network is presented. Based on the user behaviour predictions, local cache is defined which stores the content based on the popularity predicted on the user behaviours.

2.1.4.2 Social Context based Mobility Prediction

Sometimes the system is not able to obtain mobility information from the users. To address such cases, a social context, such as previous interactions of the users with the system can be used to devise the mobility of the users. In [43], a Short Message Service (SMS) based mobility prediction scheme is defined by using SMS data to determine the location of the user and predict the mobility using two different algorithms, a Multi-Connectivity (MC) model and Naive Bayes (NB) model. In [39], as discussed earlier, an LSTM based mobility prediction scheme is proposed. The proposed Proactive Caching-based Mobility Prediction (PCMP) predicts the nearest RSU for caching information based on user content requests and mobility in VANETs. Additionally, in [44], a survey of different mobility prediction models based on social interaction in urban areas is presented.

2.1.4.3 Social Context based Access Control

Social context can be considered to devise efficient access control and resource allocation schemes. Information such as social mobility in the network can be used to determine demand at different times of the day and year, so as to efficiently optimize the system. In [41], as discussed previously, an RL scheme is proposed to take into account the mobility in VANET. A deep Q algorithm is used to devise the optimal caching and computing allocation schemes based on the mobility of the users in the network. In [45], an ARIMA model is used to take into account user behaviour in device-to-device (D2D) networks and predict an optimal resource allocation strategy. It also takes into account the social correlations between users so as to optimize the overall network utility in a social community, i.e., a pool of users who have similar interests and behaviours. Additionally, in [46], an overall survey of socially aware resource allocation scheme in D2D networks is presented.

2.2 Anticipatory Techniques

The different types of information discussed in the previous section leads to design of different types of anticipatory techniques for such information. This section will provide information and discuss the recent works on anticipatory networking, which apply different techniques to forecast further networking information based on contextual information obtained from network.

We have divided these techniques into three main categories and discussed them in details. These categories include: 1) *time series prediction*, that predict future values based on previously observed values; 2) *Probabilistic forecasting* methods, that make statements about the likelihood of the future events, based on available information; 3) *Data-driven* approaches, such as clustering, classification and regression that learn to make predictions by only relaying on data. In the following subsections, we discussed these methods and their applications in anticipatory networking.

2.2.1 Time Series Prediction

The complex varying nature of wireless channel has been a subject of study of researchers for many years. Knowing how the channel behave can provide plentiful competitive advantages in terms of radio resource management, allowing for instance network operators to expand services for more users with the same amount of network resources, while also maintaining strict QoS. Power control, resource blocks allocation, coding rate, transmit antennas, precoding codeword and constellation size are examples of transmission parameters that can be ad-

justed according to the prior knowledge of the channel condition, especially for applications that demand critical network requirements, such as Industry 4.0, tactile communication and IoT.

To achieve this potential, we need to transmit signals in a closed-loop manner, receiving the CSI as a feedback from the receiver. This information can be understood as a time series that represents an estimation of the channel condition, describing the combined effect of, for instance, scattering, fading, and power decay. Nevertheless, this estimation might not be always precise. Either in frequency-division duplex (FDD) or time-division duplex (TDD) systems, problems of estimation accuracy are still raised, especially in high mobility or high frequency scenarios. This can be a serious shortcoming, notably when it comes beyond 5G and 6G network requirements, which are expected to increasingly explore the use of millimetre waves under high speed scenarios, such as autonomous vehicles and aerial navigation systems.

As precisely summarized in [47] and [48], many authors have been extensively demonstrating the impact of CSI inaccuracy over a wide variety of adaptive transmissions techniques, such as MIMO systems [49, 50, 51], beam-forming [52, 53], antenna selection [54], mobility management [55, 56], precoding [57], interference coordination [58, 59] and relaying [60, 61].

In this regard, many authors have been proposing solutions to cope outdated CSI issues [48], which can be mostly categorized into three different classes:

1. sub-optimal methods [62], where imperfect CSI is assumed and only part of the full performance potential can be achieved;

2. passive methods [63], in which wireless resources (frequency, power, time, etc) are used to passively compensate the performance loss;

3. channel prediction methods [64], in which the main idea is trying to forecast the channel behavior in advance (future CSI values), without using any additional wireless resource.

In the past few years, prediction methods have been gaining increasingly more attention from researchers due to their impact on performance and efficient use of radio resources [47]. More in details we can distinguish the prediction methods according to two different models: autoregressive (AR) [65] and parametric (PR) [66]. Both represent classical approaches to establish a statistical modeling of the wireless channel. The PR models rely on estimating fading channel parameters (e.g., Doppler shift, delay spread, angles of arrival, etc), which is a process that can demand high computational complexity and be vulnerable to

changes at the scenario. The AR models, in turn, are based on an autoregressive process to extrapolate future CSI values by linearly combining past and current CSI measures. Besides its simplicity compared to the PR approach, the AR models are very susceptible to additive noise [67] and not really deployed in practice.

More recently, an alternative methodology using Machine Learning (ML) has been raising as a promising candidate to enhance prediction methods [47]. Its biggest advantage is that the data-driven nature of ML can replace statistical modeling of the channel, thus avoiding all the drawbacks involving parameters estimation and additive noise handling. In this regard, many authors have been exploring ML techniques in different prediction tasks. In [68] and [69], the authors applied LSTM networks to tackle classical channel estimation problems, including high speed scenarios. In [70], the authors propose a CNN, a technique typically used in computer vision tasks, to extract complex CSI features from the channel and predicting CSI aging. A solution for MIMO channel prediction problem was proposed by the authors in [71] and [72] using RNNs. The authors in [73] propose a novel CSI prediction scheme for improving the performance of massive MIMO, Non-Orthogonal Multiple Access (NOMA), Coordinated Multi-point (CoMP), and physical layer security by applying multi-hidden layer neural network. The work proposed by the authors in [74] provides a very insightful application of LSTM and CNN models for predicting the CSI values in downlink channel by using information from uplink channel. Likewise, the authors in [75] extrapolates the downlink CSI values from observed uplink CSI information, but also considering both Single-Input Single-Output (SISO) and MIMO scenarios. This new proposed scheme actually outperforms the classic Wiener filter-based approach.

2.2.2 Probabilistic Forecasting

Probabilistic forecasting methods employ available information of system to make statements about the likelihood of the future events. In the following subsections we introduce two well-known statistical methods for probabilistic forecasting: Markovian Models and Bayesian inference.

2.2.2.1 Markovian Models

A Markov model is a stochastic model used to represent random changing systems in which future states only depend on the current state. This section will provide information about MC, and its applications in anticipatory networking.

A (Discrete-time) MC is a sequence of random variables X_1, X_2, X_3, ..., where, probability of moving to the next state, is independent from the previous states and only depends on the current state (Markov property):

$$P(X_{n+1} = x|X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = P(X_{n+1} = x|X_n = x_n). \quad (2.1)$$

MCs can be represented by a state-flow diagram, i.e., a directed graph in which the nodes indicate the states and the edges specify the probability of moving from one state to another. Figure 2.1 shows a classical example of MC that models the "weather" process with three possible states: {*Sunny*, *Rainy*, *Snowy*}. As an example consider the observed state in the present time step is *Sunny*, then the probability that MC takes value *Sunny*, *Rainy*, *Snowy*, in the next time step, is 0.8, 0.19 and 0.01, respectively. The sum of all probability values of the outward edges from any state is equal to 1.

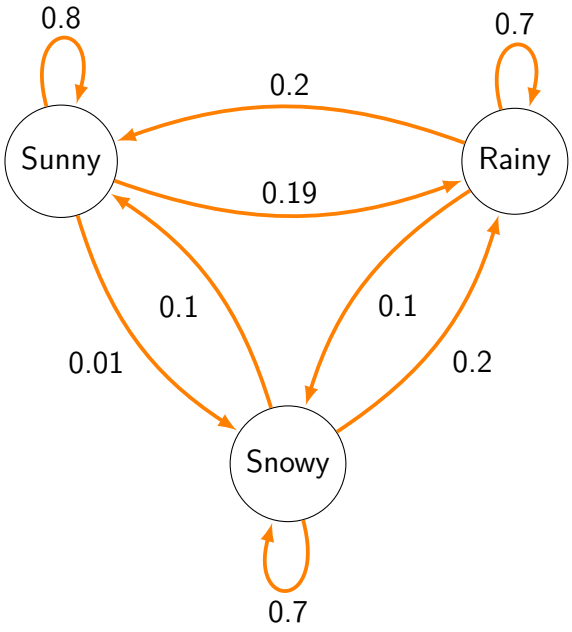


Figure 2.1: Example of state flow diagram of a MC.

Having the Markov transition probabilities, and the stationary distribution of the MC states could help to predict the behavior and the state of the system in future time steps. Markov modes have already been applied in wireless communication networks, mostly for user and vehicle positioning and mobility forecasting.
Several MC-based approaches have been proposed to predict the mobility and trajectory of locations of vehicles [76] and users [77, 78] in VANETs [76], Heterogeneous Cloud Radio Access Networks (HCRANs) [79], and WSNs [80], employing different models such as Hidden Markov Model (HMM) [81], second-

order Markov [82], and temporal Markov models [83]. The predicted mobility information can be used for different purposes like access point or service node selection [84, 85], routing in VANETs [86] or WSNs [80, 87] and traffic management [88]. To achieve more accurate results, some works have jointly applied Markov models with other techniques such as classification [89] and LSTM [90]. The Spatio-temporal correlation that existed in users' mobility patterns has also been investigated in some literature [91, 92, 93]. The method in [91] takes into account the non-Gaussian characteristics and Spatio-temporal correlation of real human mobility data to present a Markov model for the human mobility prediction.

The authors of [92] proposed a hybrid MC model that can adaptively apply first or second-order MC to predict the future location of users, based on the quality of the mobility traces. They further presented the Zone of Interest discovery scheme in urban areas, utilizing the Spatio-temporal analysis. A HMM-based mechanism, namely CityTracker, is presented in [93]. It predicts the individual's trajectory and analyzes the representative citywide crowd mobility. The proposed algorithm can achieve a representative crowd mobility visualization in the target area by integrating individual trajectories. Markov models have also been applied in prediction of other networking parameters like as signal fading [94], energy [95], node status in WSNs [96] and indoor positioning [45]. A distributed probabilistic approach based on Hidden Gaussian Markov Model (HGMM) presented in [97], to predict sensor failure in WSNs.

2.2.2.2 Bayesian Inference

Bayesian inference is a statistical method that applies the Bayes theorem to update the probability of a hypothesis given new information. In fact, the Bayesian inference derives the *posterior probability*, given the *prior probability* of hypothesis and a *likelihood function*, applying the Bayes' theorem:

$$p(\theta \mid y) = \frac{p(y \mid \theta) \cdot p(\theta)}{p(y)}, \tag{2.2}$$

where y and θ represent the data and the *hypothesis* whose probability may be affected by data. $p(\theta)$ indicates the prior probability (probability of θ before y data is observed). $p(\theta|y)$ is the posterior probability (probability of θ, given y). $p(y|\theta)$, called *likelihood*, represents the probability of y given θ.

Recently, Bayesian inferences have been applied to predict different networking parameters from mobility [98] to channel gain [99] and reliability [100]. The work in [98] employed Bayesian inference to devise a mobility prediction model for WSN. The authors of [101] presented a feature-based Bayesian method for

content requests and popularity prediction in edge-caching networks. The authors of [100] proposed a service prediction model and applied the Bayesian network method to learn and predict the reliability of the mobile wireless network. A channel gain prediction method for mobile users that exploits the Spatio-temporal correlation in a Bayesian framework is presented in [99]. Traffic prediction is another application that has been considered in some recent works. A prediction algorithm is proposed in [102] based on the Bayesian Spatio-temporal model to predict the spatial distribution of traffic in the cellular network at different moments. In [103], a wireless traffic prediction is presented based on Bayesian seasonal adjustment.

2.2.3 Clustering, classification, and Regression

Clustering, classification, and regression are three of the most frequently used techniques for extracting useful information and predicting incidents in different applications. In this subsection, we discuss a few selected methods in two main categories: similarity-based approaches and regression analysis.

2.2.3.1 Similarity-Based Approaches

The main goal in similarity-based approaches is to learn a similarity function between objects to reveal the similar latent structures in a dataset. They have a great variety of use cases in different applications, including ranking, recommendation systems, face verification, and speaker verification. These techniques have also been applied to anticipatory networking literature, which will be briefly discussed in the following, with reference to three main categories.

Collaborative Filtering (CF)

In collaborative filtering, which has been widely applied to recommendation systems, the underlying assumption is that if two people have the same opinion on an issue, they are more likely to have similar opinions on other issues. Based on this assumption, the CF techniques try to predict a user's opinion on an issue that is not rated by the user, given the preferences of other users.

Since there are a few comprehensive surveys on CF [104, 105, 106], we briefly introduce the concepts of CF, related to anticipatory networking in this section. In CF techniques, we usually have a matrix of size $n_u \times n_c$ in which n_u is the number of users, and n_c is the number of contents. The goal is to predict the missing values in this matrix.

There are two major categories of CF techniques: memory-based and model-based. In the memory-based methods, similar users are identified using similarity metrics like cosine similarity or Pearson correlation, and then the missing values are predicted using a weighted average over the ratings of the similar users on

the contents. Model-based approaches are another broad group of methods for CF in which machine learning algorithms including K-Nearest Neighbors (KNN), Singular Value Decomposition (SVD), Matrix Factorization (MF) and Artificial Neural Network (ANN) are used to estimate the ratings of the unrated contents in the matrix.

Network caching is one of the critical concepts in ensuring QoE to the users. In [107], the authors have recently published a thorough survey on caching strategies and techniques on mobile edge, in which several techniques have used different types of collaborative filtering. Another comprehensive survey on caching-enabled networks from a popularity-based video caching perspective have also published [108]. In this survey, there are a few works on neural-network-based CF. Another work in [109] proposes a collaborative multicast beamforming approach for content delivery in cache-enabled networks in which the popularity of contents are estimated and, based on that, users are served using the cached content in Small Base Stations (SBSs). Similar to popularity-based content caching, several works have proposed approaches to use users' locations to provide them with the proper content [110].

Clustering

Clustering is a broad group of unsupervised machine learning techniques that aims to identify different subsets of objects in a dataset in the way that the objects inside a group are similar to each other.

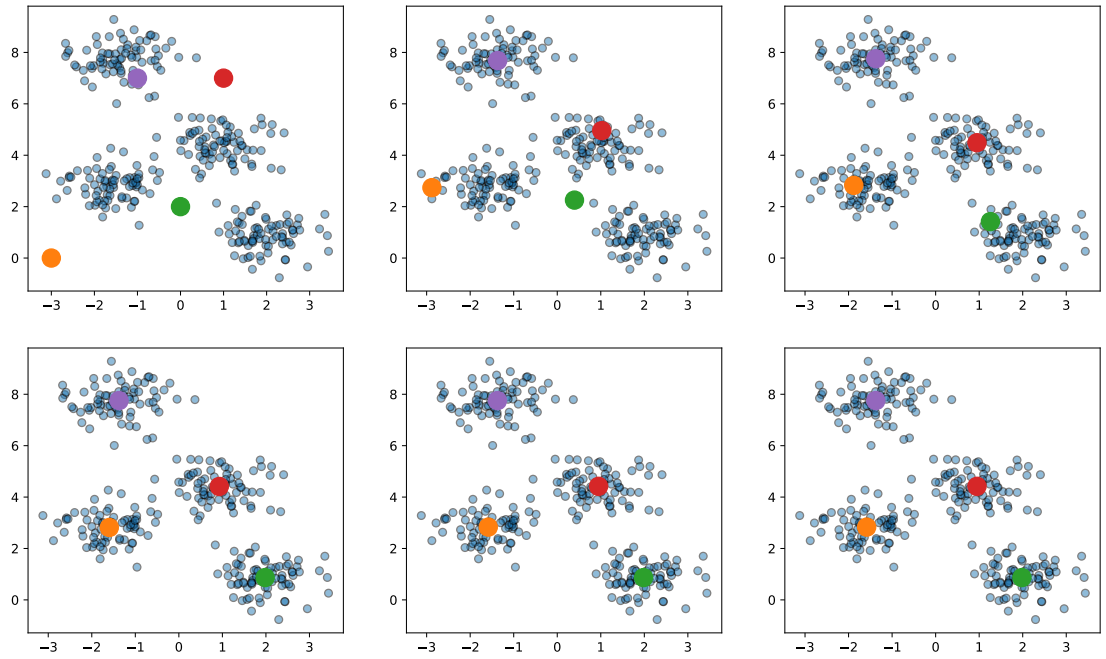


Figure 2.2: Example of k-means algorithm on a synthetic dataset after six steps with k equals to 4.

Here, we briefly introduce the k-means clustering algorithm as one of the most frequently used clustering algorithms in anticipatory networking. We are given a dataset $x^{(1)}, ..., x^{(m)}$ with m points. Each of these points has a feature vector $x^{(i)} \in \mathbb{R}^n$ of dimension n. The goal is to assign group labels $c^i = , 1, 2, ..., K$, to each point in the dataset in such a way that the points in the same group are close with respect to a distance metric defined over the feature space.
The k-means algorithm is described in Algo. 1, and a step by step visualization of the algorithm on a synthetic dataset with $k = 4$ is shown in Figure 2.2.

Algorithm 1 K-means Clustering Algorithm

Require: cluster centroids $\mu_1, \mu_2, ..., \mu_k \in \mathbb{R}^n$ intialized randomly

while the **centroids** are not stabilized **do**

for every i, set $c^i := \arg\max_j ||x^{(i)} - \mu_j||^2$

for each j, set $\mu_j := (1/||c^j||) \sum_{i=1}^{||c^j||} x^i$,

where $||c^j||$ represents the number of data points in the j^{th} cluster

end while

Different clustering algorithms, including k-means, have been applied to a broad range of cases in anticipatory networking. Optimizing the energy consumption in WSNs is one of the challenging problems for which a few clustering-based approaches have been proposed [111, 112].
Vehicular scenarios is another field in which clustering techniques play a significant role. To guarantee the stability of the self-organized communication structure in multimedia communication on the Internet of Vehicles (IOV), the authors in [113] propose a content-aware model. In another work in the vehicular domain, [114], the authors introduce a hierarchical clustering protocol to optimize the resource utilization in the network and increase the overall lifetime. Aligned with the previous work, MCA-V2I [115] is a multi-hub clustering mechanism aiming to decrease the number of control messages and increase the stability by using a master/slave paradigm in IOV.

Decision Tree-Based Classification

Decision tree learning approaches are simple yet powerful models in ML either for classification or for regression tasks. They have a tree-like structure consisting of decision nodes and leaves. Decision nodes direct the data toward a proper leave, and the leaves are the labels of the classification task.
This simple ML method has been applied to a vast number of problems, and anticipatory networking is not an exception. Decision tree-based methods play an

essential role in localization and positioning systems using Radio Frequency(RF) signals. The authors in [116] have studied different decision tree-based methods in indoor positioning systems. Another group of researchers has proposed a Non-Line-of-Sight (NLoS) location tracking system based on decision trees that use Ultra Wide Band (UWB) technology[117].
Decision trees have also been considered for device type classification in WSNs [118], multi-object detection and classification in outdoor scenarios [119], and relay selection for dual-hop wireless communications [120].

2.2.3.2 Regression Analysis

Regression analysis is a powerful statistical tool for examining the relationship between a few independent variables with a dependent variable of interest. In anticipatory networking, a great variety of papers have used regression analysis for different tasks. Hence, here, we introduce two of the most useful regression techniques and their use cases in anticipatory networking.

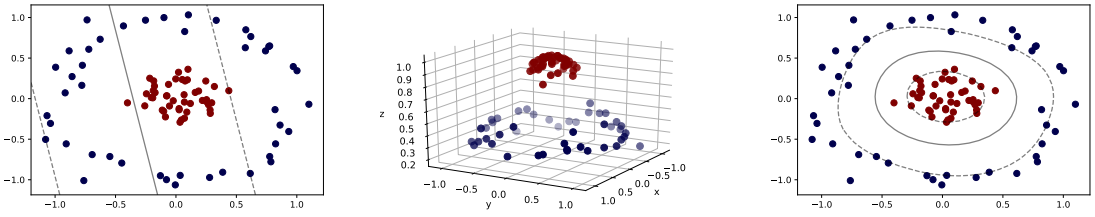
Support Vector Machine

SVM [121] is a discriminative classifier defined by a decision hyperplane separating instances of different classes with the maximum margin. There is a regression version of SVM named Support Vector Regression (SVR) sharing the same principles with SVM but for regression tasks. Here, we briefly introduce SVMs and overview recently published works in anticipatory networking using SVM and SVR.
We explain SVM in a binary classification scenario which can be extended to multiple classes classification tasks too. Imagine we have a training dataset $\{(x_i, y_i)|x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, 2, ..., M\}$, where x_i is the i^{th} training sample in an n dimensional space, y_i is the corresponding label for that sample, and M is the total number of training samples.
We assume that the samples are linearly separable, thus there is a hyperplane $w \cdot x - b = 0$ (where $W \cdot x$ is the inner product of W and x), which separates those two classes of samples. The optimization problem that yields decision hyperplane with the largest distance from the support vectors (the closest vectors to the decision hyperplane) is:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1, i = 1, 2, ..., M. \end{aligned} \tag{2.3}$$

This optimization problem is not able to find an optimal hyperplane for non-linear decision boundaries. To solve this problem, kernel methods have been proposed. In kernel-based SVM instead of using the original input attributes x, we can

transform the original input using a feature mapping function Φ and then apply the SVM. Since the Eq. (2.3) can be entirely written in terms of the inner product $x_i \cdot x_j$, we can simply replace $x_i \cdot x_j$ with $\Phi(x_i) \cdot \Phi(x_j)$. The linear model in the new space corresponds to a non-linear model in the original space.



(a) Linear kernel on a linearly non-separable synthetic dataset.

(b) The data after applying a Gaussian kernel to add a new dimension z.

(c) Classification with SVM with Gaussian kernel on a linearly non-separable synthetic dataset.

Figure 2.3: Examples of applying the SVM classifier with a linear and Gaussian kernels on a linearly non-separable synthetic dataset.

One of the most frequently used kernels in the literature is the Gaussian kernel: $\exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$. An example of applying a linear and Gaussian kernel-based SVM classifier on a non-linear synthetic data is shown in Figure 2.3. When we apply the Gaussian kernel, the data become linearly separable in a space with more dimensions which is equal to a non-linear decision boundary in the original space. In [122], a recent and comprehensive survey on SVMs and their applications is proposed. Additionally, SVM attempts to find a decision boundary for the given supervised set while SVR attempts to find a curve that fits the supervised set.

The authors in [123] proposed a machine learning-based context prediction system to boost vehicle-to-cloud communication. In this work, they have used SVR to predict the data rate by feeding three sets of features: mobility context, channel context, and application context to the model. Another group of researchers has applied a non-linear SVR to cancel the self-interfering signal in full-duplex communication systems [124]. They have used two separate SVR models for predicting the real and the imaginary parts of the interfering signal. Coverage area detection [125] and network quality prediction [126, 127, 128] have also benefited from SVRs for building a coverage map and predicting the data rate in vehicular scenarios to improve the resource efficiency and the network reliability, respectively.

Artificial Neural Networks (ANN)

ANNs are universal function approximators consisting of neurons grouped in multiple layers (input, hidden, and output layers). ANNs can be used for different tasks including clustering, classification and regression. Each neuron receives the input vector X from the previous layer, multiplies them by a vector of weights W, applies a summation over them, adds a bias term b, and finally uses an activation function (Sigmoid, Tanh, ReLU, ELU, etc.) g to introduce non-linearity:

$$\hat{y} = g(W^T \cdot X + b). \tag{2.4}$$

To train an ANN, the forward and a backward pass is used. In the forward pass, a chain rule is applied to determine the gradient and in the backward pass, a back propagation mechanism is used to adjust the weights and biases of the layers by using an optimizer such as Adam, RMSProb, Gradient Descent, Momentum, etc. Using this technique, we try to minimize the loss function to update learning parameters of the network. A loss function is dependent on the optimization problem to be solved and the dataset used for training the ANN. With the advance of computational resources like Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) and abundance of data, especially in networking, ANNs have increasingly gained attention in the literature.

CSI estimation is one of the key challenges in wireless communications. OCEAN[19] is an online neural network model that uses CNN and LSTM to predict CSI in 5G wireless communication systems. SCNet [17] is another ANN model that aims to predict the downlink CSI in Base Station (BS) of a FDD MIMO communication system using an autoencoder-like design which not only reduces the redunduncy of data but also makes it more robust to noise.

Additionally, Milimeter Wave (mmWave) communication is a relevant component of 5G and beyond. Beam selection and blockage prediction are two of the most important challenges in these systems. The authors in [129] have proposed a deep learning-based approach to tackle these challenges using ResNet18[130] architecture.

Context-aware wireless communication optimization is another direction to which a few ANN-based approaches have been applied. The authors in [131] have proposed a deep learning model using CNN to extract fine-grained features and the applied SVM classifier to classify the applications in a wireless network to improve QoE. Also, authors in [132] have used ANNs as well as other classifiers like SVMs for classifying the Wireless Local Area Network (WLAN) traffic to reduce the power consumption.

2.3 Anticipatory Optimization Techniques in Wireless Networks

Anticipatory techniques and information defined in the previous sections can hence be used for design of optimization techniques in different wireless networks which can been used to optimize the system for a variety of applications across different layers. In this section, we discuss the application of such optimization techniques based on anticipatory information from the wireless network.

2.3.1 Physical Layer

Physical layer functionalities such as beamforming design directly impacts the instantaneous performance of the wireless systems. In the current literature, we can find numerous optimization techniques applied at physical layer. In [133], the authors investigate the problem of how to maximize the sum rate of a 2-user uplink mm-wave-NOMA system, proposing a sub-optimal solution where the original problem is divided into two sub-problems: one is a power control and beam gain allocation problem, and the other is a beamforming problem under the constant-modulus (CM) constraint. The authors in [134] propose a centralized as well as a distributed power control algorithm, aiming to maximize the capacity of a D2D network. They consider a scenario of licensed and unlicensed spectrum, showing by simulations that the proposed approach could increase the throughput of the D2D networks compared with current state-of-the-art methodologies. In [135], the authors address the energy efficiency optimisation problem of a NOMA 5G wireless network. The proposed idea is based on improving energy efficiency of 5G terminals while satisfying the constraints on maximum transmit power budget, minimum data rate, and minimum harvested energy per terminal. The proposed scheme is compared with an exhaustive search method, showing convergence to a stable optimal value. The authors in [136] propose a new approach for multiple access in the fifth generation (5G) of cellular networks called power domain sparse code multiple access (PSMA). They compare the PSMA with other proposed NOMA strategies from the perspective of receiver complexity and system performance, showing that PSMA significantly outperforms other NOMA techniques while imposing a reasonable increase in complexity to the system, considering both aspects of transmitter and receiver. We can check in [137] a flexible mechanism for Discontinuous Reception (DRX) proposed for 5G networks, where the goal is to minimize the energy usage of user devices for applications of video streaming while preventing buffer underflows. The authors show that the proposed approach can provide an energy reduction usage by up to 60% compared to static DRX setups. The work in [138] proposes a novel channel state feedback scheme, where the minimum number of feedback

bits is calculated with respect to the channel reconstruction error. The authors compare the proposed approach with other traditional feedback schemes, showing analytically and numerically the advantages of the proposed solution, especially in conditions of low SNR. A pilot placement optimization problem is proposed by the authors in [139] for the radio access in 5G vehicle-to-everything communications to support Internet of Vehicles applications. The authors formulate the problem as a MDP, aiming to find enhanced pilot patterns for assisting OFDM technology. Simulation results show that the proposed scheme was able to effectively track fast time-varying vehicular channels. In [140], the authors provide an optimization algorithm for calculating the joint maximum-likelihood estimation of channel and clipping level at the receiver side of IoT-based OFDM networks. These network scenarios are typically characterized for having lots of low-cost low-power transmitters and more complex receiver nodes, such as a base station. Numerical evaluations show that the proposed estimator was capable to achieve almost the same performance of a perfect estimator, where both channel and clipping level are perfectly known.

2.3.2 MAC Layer

In MAC layer, a main focus has been on applying optimization techniques for radio resource allocation and error correction. In [141], the authors propose a novel scheduling framework that is able to select different scheduling policies according to the states of the scheduler by using RL principles. The main goal is minimizing packet delay and packet loss rates for attending the strict 5G network requirements. Simulation results show that the proposed scheme outperforms traditional packet schedulers. Likewise, the authors in [142] also propose a scheduling framework based on a RL strategy. Their idea is to optimize QoS provision management for heterogeneous traffic. The proposed scheme is capable of maximizing the average scheduling time when heterogeneous QoS requirements are met for diverse traffic classes, achieving a performance up to 50% higher than state-of-the-art solutions. The work proposed in [143] provides a solution for the ultra-reliable low-latency communications (URLLC) and enhanced mobile broadband (eMBB) coexistence on the same radio spectrum. The authors introduce a novel scheduler framework that aims at optimizing a cross-objective function, where the critical URLLC QoS is guaranteed while extracting the maximum possible eMBB ergodic capacity. Simulation results show that the proposed scheme guarantees instant scheduling for sporadic URLLC traffic, and with minimal impact on the overall ergodic capacity, overcoming state-of-the-art scheduling proposals from both industry and academia. In [144],

the authors optimize the hybrid automatic repeat request (HARQ) operation of 5G URLLC networks. The problem is formulated according to a non-convex and mixed integer programming, which is analytically intractable. The authors came up with a solution based on optimizing a repetition coding scheme, and showing, by simulation results, minimization o the required bandwidth to achieve URLLC traffic requirements. Likewise, the authors in [145] also propose a solution for optimizing HARQ operation in order to achieve URLLC requirements. The proposed scheme applies the queuing delay model based on the Pollaczek-Khinchine (P-K) formula, aiming to optimize bandwidth considering HARQ of 5G URLLC communication. Analytical and simulation results show that the proposed mechanism was able to reduce bandwidth requirements by up to 64.8% compared to traditional approaches. The work established in [146] utilizes Tennessee Eastman (TE) process model to create a novel concept for communication-edge-computing (CEC) loop, introducing an optimization problem for achieving the defined CEC efficiency with focus on industrial IoT applications. The authors derive a new uplink (UL)-based communication protocol, that was able to outperform typical HARQ performance in terms of latency, reliability, and bandwidth efficiency.

2.3.3 Network Layer

In Network Layer, the optimization techniques are mainly applied to routing problems. In [10], a location error resilient routing scheme is proposed for VANETs. It predicts the location of the vehicles in the network using Kalman filter and creates a routing directory to decide the next forwarding node in the network. The protocol is designed to optimize the selection of the next forwarding node to maximise the throughput and minimise the routing load in the network. In [147], a Genetic algorithm-based energy-efficient clustering and routing algorithm (GECR) is presented which takes into account the location of the nodes and devise routing and clustering strategies in WSN. The cluster head selection and next hop forwarding node is devised by modelling the optimization problem and solving it using a genetic algorithm. The algorithm aims to maximise the network time and minimise the energy consumption in the network. Similarly in [148], a multi objective pareto optimization approach is proposed to solve the problem of clustering and routing while taking into account energy efficiency, reliability and scalability. In [149], an optimization problem called minimum broadcast power is discussed and is solved using a hybrid particle swarm optimization algorithm (H-PSO). The H-PSO aims at minimising the transmission power so as to conserve energy in broadcast mode of transmission in wireless ad-hoc network. In [150], a non supervised deep learning based routing algorithm is proposed

to optimize traffic load in wireless Software-Defined Networkings (SDNs). The SDN controller trains CNN to take into account traffic information and network performance such as path delay and make routing decisions based on these experiences. In [151], a multihop routing scheme based on classification using machine learning techniques for Mobile Ad-Hoc Networks (MANETs) is presented. In this scheme, the algorithm takes into account the battery power utilization and internal storage of a node to determine the next hop and eventually the whole route. To do so, each neighbouring node is classified using three classification techniques namely Maximum-local-rate (MLR), SVM and KNN are used.

2.3.4 Application Layer

In Application Layer, the optimization techniques are generally applied to performance predictions over wireless network for applications. In [35], an adaptive HTTP video streaming approach is discussed which takes into account the performance variations due to the dynamic nature of wireless networks. The system predicts the channel throughput and maximises the average download segment level or video quality while taking into account the channel utilization and end user buffer. Similarly, in [37], a throughput prediction scheme is proposed which also takes into account channel information to optimize and improve the video streaming quality. In [152], a multi-stage ML is presented by combining an unsupervised feature extraction with a supervised classifier to extract the *quality-rate* characteristic of unknown videos that can be used in QoE-aware video admission controls and resource management. In [153], a flow rate allocation algorithm over Multipath TCP (MPTCP) is defined called Energy Distortion Aware MPTCP. The protocol aims to minimising the energy consumption while maximising video quality. The optimization problem takes into account the channel quality, delay, and video streaming requirement to provide an optimized flow allocation and energy consumption. In [154], a cross layer UDP based protocol which takes into account the wireless links to provide a guaranteed performance. It basically optimizes functions such as in order delivery, forward error correction and flow control so as to minimise delay and overhead in the wireless network.

2.4 Conclusion

In this chapter, we highlighted the different anticipatory information and techniques along with system-wide optimization schemes for different applications over the entire protocol stack technologies and considering different wireless network scenarios and applications. Primarily, the chapter shows works on anticipatory information that can be garnered from the wireless networks, the

anticipatory techniques that can be designed using the information available, and the optimization techniques that can be designed for applications at the Physical, Medium Access, Networking, and Application Layers, thereby giving a comprehensive overview of anticipatory networking that is applied in current wireless networks. Hence, the usage of anticipatory networking across all the layers provides a good motivation to devise optimisation techniques using different anticipatory techniques at all layers. So, in the following chapters, we present the works undertaken to design and analyse different communication systems using anticipatory networking in physical, network, transport and application layers.

Group Clustering Mechanism for Joint Spatial Division Multiplexing (JSDM)

3

3.1 Introduction

Multi-antenna radio technologies have shown to enhance spectral efficiency while ensuring connectivity to a large number of devices. Different encoding schemes such as Dirty Paper Coding (DPC) have been designed to achieve the multi-antenna channel capacity [155]. However, due to the high computational complexity as well as the need for precise CSI, there has been much focus of research on sub-optimal solutions which combine Superposition Coding (SC) and spatial processing such as NOMA[156]. Additionally, as these mechanisms tend to fully decode interference, the uncertainty over CSI directly affects interference cancellation among different users. Hence, the authors of [157] have recently proposed Rate Splitting Multiple Access (RSMA) as a non-orthogonal transmission scheme that partially decodes interference and partially treats it as noise thus further improving multiplexing gains. For 1-layer Rate Splitting (RS), the message intended to each user is divided into a common (s_c) and private (s_p) parts encoded separately. In order for this transmission scheme to work, it is necessary to ensure that every user perfectly decodes the common message. This is often tackled by allocating a larger fraction of the total power to the common message. In the presence of a large number of receivers, this condition limits the total rate by the minimal common rate achieved in the whole system[1]. Hence, in the presence of several users, the power assigned to each s_p is reduced, leading to a degradation in communication rate.

In these conditions, relying on multiple common streams (generalised rate splitting) leads to higher multiplexing gains, but at the cost of high complexity at the decoder caused by the several layers of Successive Interference Cancellation (SIC) [157]. To tackle the increasing complexity of generalised RS while having small loss in multiplexing, the authors in [158] consider a 2-layer Hierarchical Rate Splitting (HRS) transmission mechanism. In this scenario, users are considered to be divided into G groups and required to decode three messages: two common messages and a private message. One of the common messages (outer common -

[1]This happens regardless of the number of antennas at the transmitter. Instead, this is a consequence of power allocation to reduce interference among different users.

s_{oc}) is encoded using a codebook shared among all the user while the other one (inner common - $s_{ic,g}$) is encoded by a codebook share only among users in a specific group. But when the groups are orthogonal, i.e. the users are sufficiently separated spatially, optimal communication happens when inter-group and intra-group interference are reduced to a level that it can be completely distinguished from the intended signals.

But, to minimise the interference and maximise the rate using HRS, the Base Station (BS) is required to know what can be referred to as the optimal clustering scheme, i.e., the one that maximises the total communication rate. Unfortunately, finding this optimal clustering scheme is an NP hard problem which often requires an exhaustive search. Thus, it becomes extremely hard to come up with an optimisation mechanism that maximises the communication rate using HRS while also considering the clustering options as an optimisation variable. Hence, in this chapter, we briefly describe a predictive clustering technique that is capable of directly learning (or approximating) the optimal clustering option from the imperfect CSI, thereby, essentially predicting the optimal clustering to maximise the performance at each user.

3.2 System Model

Consider a downlink transmission scenario where N single-antenna user equipment (UEs) receive messages from a base station (BS) over a spatially correlated Rayleigh-fading channel. We further assume this BS to be equipped with an antenna with M isotropic antenna elements. Moreover, let these UEs be partitioned into $G \geq 1$ disjoint clusters. So, the signal $\mathbf{y} \in \mathbb{C}^N$ received by all the users is given by,

$$\mathbf{y} = \mathbf{H}^{\mathrm{H}}\mathbf{x} + \mathbf{n}, \tag{3.1}$$

where, $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]^{\mathrm{T}} \in \mathbb{C}^{M \times N}$ contains the stacked channels of all the $k = \{1, \ldots, N\}$ UEs, $\mathbf{n} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I}_N)$ is an additive white Gaussian noise vector and $\mathbf{x} \in \mathbb{C}^M$ is the combined signal given by,

$$\mathbf{x} = \sqrt{p_{oc}}\mathbf{w}_{oc}s_{oc} + \sum_{g=1}^{G} \mathbf{B}_g \left(\sqrt{p_{ic,g}}\mathbf{w}_{ic,g}s_{ic,g} + \sqrt{p_{gk}}\mathbf{W}_g\mathbf{s}_g \right), \tag{3.2}$$

where p_{oc}, $p_{ic,g}$ and p_{gk} are the power allocated to the outer common message $s_{oc} \in \mathbb{C}$, inner common messages $\mathbf{s}_{ic} \in \mathbb{C}^G$ and the private messages $\mathbf{s}_g \in \mathbb{C}^{N_g}$, respectively. $\mathbf{B}_g \in \mathbb{C}^{M \times b_g}$ is the group outer precoder designed from the gth group channel's second order statistics and dependent on the integer design parameters b_g rank of the channel covariance matrix. By knowing the UEs that belong to

the gth cluster, the matrix $\mathbf{H}_g = [\mathbf{h}_{g,1}, \ldots, \mathbf{h}_{g,N_g}]^{\mathrm{T}} \in \mathbb{C}^{M \times N_g}$ contains the stacked channels of all the N_g UEs that belong to the gth cluster. The downlink fading channel $\mathbf{h}_{g,k} \in \mathbb{C}^M$ associated to the kth user of the gth class can be factored out as,

$$\mathbf{h}_{g,k} = \mathbf{R}_g^{\frac{1}{2}} \mathbf{g} = \mathbf{U}_g \mathbf{\Lambda}_g^{\frac{1}{2}} \mathbf{g}_k, \tag{3.3}$$

where $\mathbf{R}_g \in \mathbb{C}^{M \times M}$ is the channel correlation matrix, $\mathbf{U}_g \in \mathbb{C}^{M \times M}$ a unitary matrix containing its eigenvectors, $\mathbf{\Lambda}_g \in \mathbb{C}^{M \times M}$ a diagonal matrix with its associated eigenvalues and $\mathbf{g}_k \in \mathbb{C}^M$ has Gaussian independent and identically distributed (i.i.d.) entries with zero mean and unit variance which describe the complex path gains.

In principle, the covariance matrices are directly dependent on the angular response of the channels [156]. Unfortunately, in a more realistic environment, due to limited feedback, the BS only observes an imperfect estimation of the channel. Following [159], we model this imperfection as the sum of a channel and a noise generated from the same subspace, given by,

$$\hat{\mathbf{h}}_{g,k} = \mathbf{U}_g \mathbf{\Lambda}_g^{\frac{1}{2}} \hat{\mathbf{g}}_k = \mathbf{U}_g \mathbf{\Lambda}_g^{\frac{1}{2}} \left(\sqrt{1 - \tau^2} \mathbf{g}_k + \tau \mathbf{z}_k \right), \tag{3.4}$$

where $\mathbf{z}_k$ has i.i.d entries and $\tau \in [0, 1]$ indicates the quality of the instantaneous channel. For instance, $\tau = 0$ leads to a perfect channel estimation, i.e., $\hat{\mathbf{h}}_{g,k} = \mathbf{R}_g^{\frac{1}{2}} \mathbf{g}_k$ while $\tau = 1$ leads to an uncorrelated channel in the subspace spanned by $\mathbf{U}_g$, i.e., $\hat{\mathbf{h}}_{g,k} = \mathbf{R}_g^{\frac{1}{2}} \mathbf{z}_k$ for uncorrelated $\mathbf{g}_k$ and $\mathbf{z}_k$.

3.2.1 HRS Transmission Mechanism

HRS transmission design is defined based on the combined transmission signal $\mathbf{x}$ from (3.2). To determine the transmission signal $\mathbf{x}$, we obtain the precoder $\mathbf{B}_g$ following [159, 158], so that the group effective channel $\tilde{\mathbf{H}}_g^{\mathrm{H}} = \hat{\mathbf{H}}_g^{\mathrm{H}} \mathbf{B}_g \in \mathbb{C}^{b_g \times N_g}$ represents the projection of $\hat{\mathbf{H}}_g$ onto the b_g–dimensional subspace orthogonal to the $r^* = \sum_{l \neq g}^{G} r_g$ singular vectors associated to the r_g largest singular values of each of the interference groups. In order to distinguish all the N_g users in the group we must have $N_g \leq b_g$, i.e., enough degrees of freedom in the b_g–dimensional subspace. Unfortunately, it is not possible to choose b_g and r_g indiscriminately large as one constrains the growth of the other. Specifically, as there exists at most M singular vectors at each group, we have that $N_g \leq b_g \leq M - r^*$. Consequently, a large number of groups leads to less freedom on the choice of both b_g and r_g.

Moreover, $\mathbf{w}_{oc}$, $\mathbf{w}_{ic,g}$ and $\mathbf{w}_{gk} = [\mathbf{W}_g]_k$ are the unit norm precoders associated to the instantaneous outer common, inner common and private messages, respec-

tively. We can design $\mathbf{W}_g = \xi_g \left(\tilde{\mathbf{H}}_g \tilde{\mathbf{H}}_g^{\mathrm{H}} + \varepsilon \mathbf{I}_{b_g}\right)^{-1} \tilde{\mathbf{H}}_g$, given a total transmission power P, as a Regularized Zero Forcing (RZF) precoder to allow distinguishing between the N_g users within the gth group by reducing the interference among the private messages in this group [159]. The parameter ξ_g is the power normalisation factor which normalizes $||\mathbf{W}_g||_2$ to the unit. Likewise, ε is also a normalisation parameter. Similarly, $\mathbf{w}_{ic,g} = \xi_{ic,g} \sum_{k=1}^{N_g} \mathbf{w}_{gk}$ is the equally weighted Matched Beamforming (MBF) built as a linear combination of the private precoders of the gth group where $\xi_{ic,g}$ is a normalisation parameter. Finally, the outer common precoder $\mathbf{w}_{oc} = \xi_{oc} \sum_{g=1}^{G} \sum_{k=1}^{N_g} \mathbf{B}_g \tilde{\mathbf{h}}_{gk}$ is also designed as a weighted MBF, but to handle inter-group power leakage where ξ_{oc} is a normalisation paramater. Notice that it is essential to reduce inter-group interference in order to guarantee communication. Specifically, when group leakage is completely nulled out, there is no need for $\mathbf{w}_{oc}$ and communication happens over G parallel 1-layer RS streams.

To allocate power among the different messages, we further design two parameters $\alpha, \beta \in (0, 1]$. The first one α represents the fraction of the total power P allocated to the outer common message. And the latter, the fraction of the remaining power allocated to the inner common message. Combining these, we have $p_{oc} = \alpha P$, $p_{ic,g} = \frac{(1-\alpha)\beta P}{G}$ and $p_{gk} = \frac{(1-\alpha)(1-\beta)P}{N_g}$. In this work we perform a brute force search to find the optimal α and β for every channel realisation.

As mentioned above, at the receiver side, the kth user associated to the gth group decodes its message in a 2-step successive interference cancellation fashion. In the first step, the user decodes the outer common message (s_{oc}) and removes it from the received signal. The group's inner common codeword is then decoded after applying SIC. After successfully decoding both common messages, each private message is extracted by considering all other private messages as interference. As a result, the Signal-to-Interference Plus-Noise Ratio (SINR) to each of these messages is written as,

$$\gamma_{gk}^{oc} = \frac{p_{oc}|\mathbf{h}_{gk}^{\mathrm{H}} \mathbf{w}_{oc}|^2}{1 + I_{gk}}, \tag{3.5}$$

$$\gamma_{gk}^{ic} = \frac{p_{ic}|\mathbf{h}_{gk}^{\mathrm{H}} \mathbf{w}_{ic,g}|^2}{1 + I_{gk} - p_{ic}|\mathbf{h}_{gk}^{\mathrm{H}} \mathbf{w}_{ic,g}|^2}, \tag{3.6}$$

$$\gamma_{gk}^{p} = \frac{p_{gk}|\mathbf{h}_{gk}^{\mathrm{H}} \mathbf{w}_{gk}|^2}{1 + I_{gk} - \left(p_{ic}|\mathbf{h}_{gk}^{\mathrm{H}} \mathbf{w}_{ic,g}|^2 + p_{gk}|\mathbf{h}_{gk}^{\mathrm{H}} \mathbf{w}_{gk}|^2\right)}, \tag{3.7}$$

where,

$$I_{gk} = \sum_{l=1}^{G} p_{ic,l} |\mathbf{h}_{gk}^{\mathrm{H}} \mathbf{B}_l \mathbf{w}_{ic,l}|^2 + \sum_{l=1}^{G} \sum_{k=1}^{N_g} p_{lk} |\mathbf{h}_{gk}^{\mathrm{H}} \mathbf{B}_l \mathbf{w}_{lk}|^2 \tag{3.8}$$

is the combination of all interference leaked from other users and groups. Finally, we can describe the achievable rate as the combination of the smallest achievable outer common rate among all users $R_{oc} = \min_{gk} \log_2(1 + \gamma_{gk}^{oc})$, the minimal inner common rate per group $R_{ic} = \sum_{g=1}^{G} \min_{k} \left(\log_2(1 + \gamma_{gk}^{ic}) \right)$ and the sum of the rate achievable at all private messages $R_p = \sum_{g=1}^{G} \sum_{k=1}^{N_g} \log_2(1 + \gamma_{gk}^{p})$. Then the total rate is the sum of these components, i.e, $R = R_{oc} + R_{ic} + R_p$.

3.3 User Clustering and Dataset Definition

As it becomes evident from the discussion above, and further supported in our results, choosing an appropriate clustering is crucial to take full advantage of two-tier precoding mechanisms, such as HRS [159, 158]. One can rely on extensive search in order to find the optimal clustering mechanism. However, this is an NP hard task as the number of ways that a set can be partitioned into nonempty sets is given by the Bell number which grows almost exponentially with N, i.e., the number of elements in the set. Moreover, in our scenario, many of these partitions lead to vanishing communication rates due to high interference. Therefore, in this work, we rely on (possible suboptimal) clustering options obtained from an agglomerative hierarchical clustering mechanism [160].

3.3.1 User Clustering

To devise the clustering mechanism, we define a bottom up approach where the objective is to combine clusters (groups of users in the wireless network) according to their similarity. Initially, each user is associated to a singleton cluster. At each step of the hierarchical clustering algorithm, the pair of users/clusters with highest similarity (according to a criterion discussed later) is then merged. As a result, after each merge we obtain a new clustering option and evaluate the rate achieved considering this new option. This process continues until we have evaluated all levels in the hierarchy. Notice that in this agglomerative mechanism there exist only $N + 1$ (total number of users plus one) possible clustering options, one for each level in the hierarchy. These, however, are often relevant clustering options as each cluster only contains elements that are particularly similar to each other. We consider the similarity measure between two channel matrices based on how close the principle angles of the subspaces

spanned by their column-spaces are. Specifically, for two clusters of size N_k and N_j, we take the projection-Frobenius (PF) similarity

$$s_{k,j} = \frac{\mathrm{tr}(\hat{\mathbf{P}}_k \hat{\mathbf{P}}_j)}{\min(N_k, N_j)}, \tag{3.9}$$

where $\hat{\mathbf{P}}_j$ is the projection matrix given by,

$$\hat{\mathbf{P}}_j = \hat{\mathbf{H}}_j (\hat{\mathbf{H}}_j^{\mathrm{H}} \hat{\mathbf{H}}_j)^{-1} \hat{\mathbf{H}}_j^{\mathrm{H}}, \tag{3.10}$$

which describes the first N_j left singular vectors of the kth group of channels. Moreover, to improve clustering results for $N_j \neq N_k$, we follow a statistical analysis of the quantity in (3.9) and further define the normalised similarity measure

$$\hat{s}_{k,j} = \frac{s_{k,j} - \eta_{k,j}}{\sigma_{k,j}} \tag{3.11}$$

based on its asymptotic mean $\eta_{k,j}$ and variances $\sigma^2_{k,j}$ defined as in [160]. However, this normalisation step is only possible for $M > N_j + N_k$, otherwise, we follow the projection-Frobenius similarity described in (3.9).

3.3.2 Dataset Definition

We design the dataset used for this work by devising channel matrices from (3.4) and clustering them according to the scheme described above. We consider four possible covariance matrices to which channels are randomly associated. Consequently, for different samples, we might obtain a different number of users associated to a specific covariance matrix. Notice that, this is not a cluster assignment, but merely a way to generate random channels. These covariance matrices are obtained by considering the azimuth angles $\theta_g = -\frac{\pi}{2} + \frac{\pi}{3}(g-1)$ and the constant angular spread $\Delta_g = \frac{\pi}{6}$. Moreover, we further assume the BS to be equipped with a Uniform Circular Array antenna.

Concretely, we design 3 different configurations based on the choices of the number of antennas at the BS: 1) $N > M$, 2) $N = M$ and 3) $N < M$. Moreover, we evaluate these configurations for two different system loads, based on the number of users $N \in \{8, 12\}$. Specifically, we have $M \in \{6, 8, 12\}$ for $N = 8$ and $M \in \{6, 12, 16\}$ for $N = 12$. As a result, we have 6 different scenarios. For each these we generate $S = 10.000$ random samples, each sample containing both imperfect and perfect CSIT of equal size $N \times M$ and the clustering scheme that maximises the rate based on the hierarchical clustering mechanism. As a result of this randomness, for each scenario we obtain more than $G^* = 200$ possible

clustering options, thus, leading to very imbalanced datasets. To diminish this effect, for each scenario, we sub-sample the data such that only relevant classes are left, i.e., we discard classes that achieve less than 25% of the average rate of the scenario and have less than 50 samples. Moreover, to further balance the data, we crop the maximum number of samples in each class to be at most to 200. As a result, for each scenario, we still obtain an imbalanced dataset with approximately $G^* = 50$ classes, each containing at least 50 samples and at most 200 samples.

Finally, to compensate for this drop in the number of samples, we further augment the dataset of each configuration by randomly shuffling users that belong to the same cluster. This is a natural extension of this dataset as clustering should be indifferent to the ordering of the users.

3.4 Machine learning model and training

We solve the classification problem presented in the previous section by designing a shallow neural network. We used the Keras library, so we describe the layers with their notation [161]. For each scenario, we divide our dataset into training, validation and test sets in a proportion of 80/10/10. During the training procedure, we use the validation set to tune the corresponding hyper-parameters. Our model is defined as a shallow neural network following the parameters from Table 3.1. The output layer consists of G^* neurons with a *softmax* activation that correspond to each cluster where G^* is the total number of classes in the scenario. The softmax function in the output layer is used to obtain the probability of a user belonging to a specific cluster and it is given by,

$$\sigma(\mathbf{Z})_g = \frac{e^{z_g}}{\sum_{j=1}^{G} e^{z_j}}, \tag{3.12}$$

where $\mathbf{Z}$ is the input vector from the previous hidden layer, z_g the g-th element and the denominator sum is the normalisation factor to ensure the output is into the range of $[0, 1]$. Then, by selecting the maximum, we can obtain the highest probability that users are clustered in a particular way. For the training procedure, we use the Adam optimiser with a learning rate of 10^{-3}, we train for 50 epochs and use a batch size of 128 samples. For our multi-class classification task, we aim to minimise the categorical cross-entropy loss [162] given by,

$$\mathcal{L}(y_g, \hat{y}_g) = -\sum_{g=1}^{G^*} y_g \log \hat{y}_g, \tag{3.13}$$

Simulation Parameter	Simulation Value
Antenna Configuration	Uniform Circular Array
Angular Spread (Δ_g)	$\pi/6$
Number of Unique Distributions	4
Channel Quality (τ^2)	0.4
Dominant Eigenvectors ($b_g = r_g$)	$\lfloor M/G \rfloor$
Channel Quality (r_g)	0.4
Number Shuffling	10
Number of Neurons in NN	$\{256, 128\}$
NN Learning Rate	10^{-3}
NN Training Epochs	50
NN Training Batch Size	128
NN Input Layer Activation Function	ReLu Function
NN Hidden Layer Activation Function	ReLu Function
NN Output Layer Activation Function	Softmax Function
NN Loss Function	Categorical Cross-entropy Loss

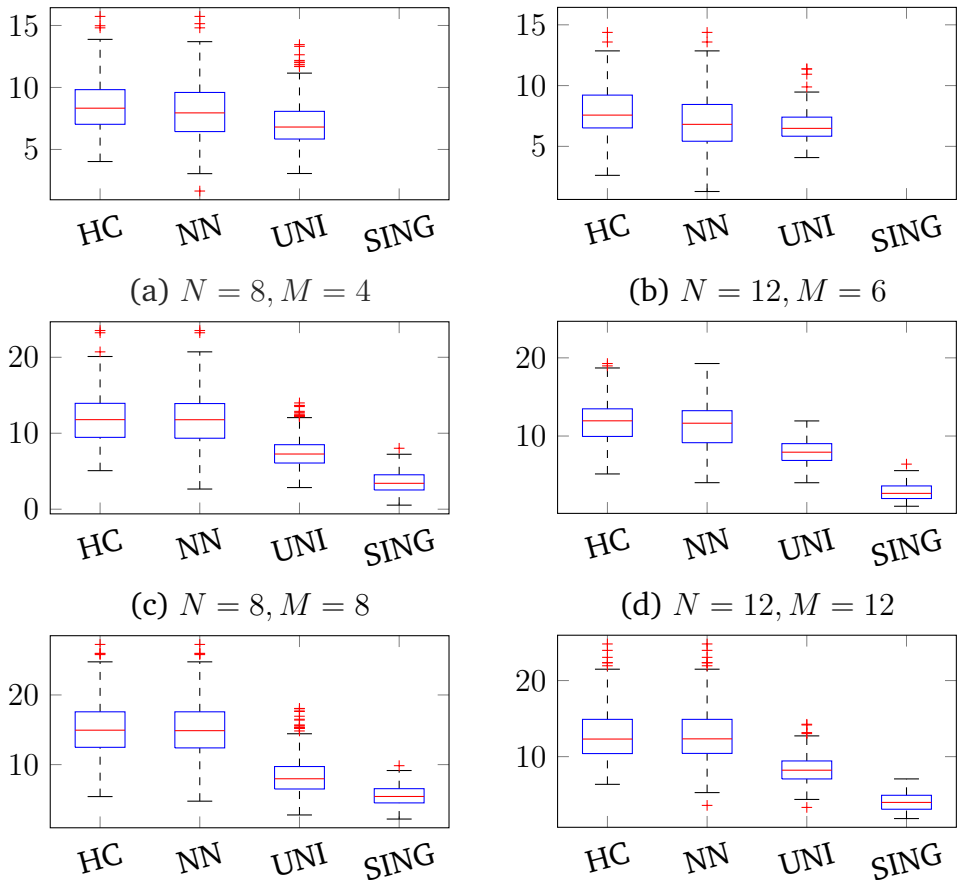
Table 3.1: Parameters of the Simulations

where y_g and $\hat{y}_g$ are the ground truth and NN score for each class. This loss is a very good measure of how distinguishable two discrete probability distributions are from each other. In this context, the vector $\hat{\mathbf{y}} = [\hat{y}_1, \ldots, \hat{y}_g] \in \mathbb{R}^{G^*}$ has entries which represent the probability that users are clustered in a specific manner and the sum of all entries is one. The accuracy of a model is often defined in terms of the entry with highest probability, this is often, called *top-1* accuracy. In our scenario, there exist several options which achieve the (close to) maximum rate. Therefore, it is also interesting to analyse the *top-k* accuracy of our model, i.e., if the desired clustering option is among the k most probable outputs.

Finally, we emphasise that we are applying a shallow neural network which contains only a small number of learnable parameters. This is designed as a consequence of our devised dataset. Recall that we have specifically defined it to be imbalanced and with a small number of samples to each class ($[50, 200]$). Nonetheless, as we present below, this network is capable of learning the relationship between the different channel matrices and directly output the desired clustering option that maximises transmission using HRS.

3.5 Performance Analysis

In this section, we evaluate the performance of the presented Neural Network (NN) method in comparison to RS under different scenario configurations. These numerical simulations are carried out in a MATLAB environment. The necessary configuration parameters are defined in Table 3.1.



(a) $N = 8, M = 4$ (b) $N = 12, M = 6$

(c) $N = 8, M = 8$ (d) $N = 12, M = 12$

(e) $N = 8, M = 12$ (f) $N = 12, M = 16$

Figure 3.1: Spectral efficiency (bps/Hz) achieved for clustering mechanisms using HRS.

In order to validate the learning of the NN, we compare the rate achieved using the NN predicted classes and different RS clustering options. To perform a complete evaluation, we determine the rate achieved by the following solutions,

- Hierarchical Clustering - Hierarchical Rate Splitting (HC): The users are clustered according to the clustering mechanism defined in Sec. 3.3, the group with higher communication performance is selected;
- Neural Network - Hierarchical Rate Splitting (NN): Proposed NN based clustering;
- Universal Cluster (UNI): All users are clustered into one single cluster;
- Singleton Cluster (SING): Each cluster contains only single user.

As mentioned above, we consider three scenarios to evaluate the clustering solutions

1. $M < N$,

2. $M = N$ and

3. $M > N$.

Hence, for $N = 8$, we determine the rate achieved for $M \in \{4, 8, 12\}$ and for $N = 12$, we determine the rate achieved for $M \in \{6, 12, 16\}$. Then, we compare the different clustering techniques mentioned before based on the rate achieved. Figure 3.1 shows the rate achieved for all four clustering techniques for the different values of M and N. Each box plot shows the rate obtained for different realisations of the channel. The median rate is presented by a horizontal line through box and the top and bottom of the box are the 75th and 25th percentile rate (i.e. rate achieved by 75% and 25% of the scenarios). Lastly, the extremities of the boxplot refer to the 1% and 99% and the red plus indicators in the boxplot denote the outlier rate values. Notice that the rate achieved by HC-HRS and NN-HRS is approximately similar while both clustering techniques outperform UNI and SING. This is due to the fact that with a noisy channel, it is really difficult to generate accurate precoders that can maximise the rate achieved and minimise the inter-group and intra-group interferences. Additionally, the NN-HRS only receives the instantaneous noisy channel as an input and determines its clustering solution while HC-HRS needs to iteratively determine the similarity between different channels making it considerably slower when compared to the NN solution. Moreover, for SING, the choice of parameters b_g and r_g seems to harm the performance. We recall that both parameters are integers thus are susceptible to the trade-off between M and G. For instance, for $G = N = 8$ and $M = 12$, there exist only one viable option of r_g, i.e., $r_g = \lfloor M/G \rfloor = 1$. Alternatively, we could select four ($\mathrm{mod}(M, G)$) groups to have $r_g = 2$, but this requires further processing on the choice of these groups. As a consequence, we obtain similar rates for $N = 8$ users served with $M = 8$ or $M = 12$. Similar consequences are obtained for $N = 12$. Moreover, for $G = N > M$, we have $r_g = \mathrm{mod}(M/N) = 0$ what makes impossible to derive meaningful precoders Figures 3.1(a)-(b). In contrast to that, the other three techniques, which consider clustering, do not suffer from this trade-off between G, r_g and M. Instead, even for $N > M$ we still achieve reasonable spectral efficiency.

Finally, we analyse the capability of the shallow NN to learn the grouping classification task as described above. To do so, we first analyse the accuracy of the network for class prediction. Recall that, here, a class represents a different clustering option. Table 3.2 presents, in percentage, the results obtained by training different NN according to the configuration parameters in Table 3.1 for different number of users (N) and antennas in the BS (M). The validation

N / M	Validation *(top-1)*	Test *(top-1)*	Test *(top-3)*	Test *(top-5)*	Test Relative Rate
8 / 4	65.38%	65.37%	85.22%	90.48%	94.12%
8 / 8	98.3%	92.0%	96.3%	97.7%	99.0%
8 / 12	96.9%	92.2%	97.0%	98.2%	99.5%
12 / 6	71.45%	35.6%	65.62%	77.75%	89.99%
12 / 12	98.7%	86.2%	96.8%	98.9%	93.5%
12 / 16	99.18%	95.62%	98.32%	93.32%	99.77%

Table 3.2: Summary of Results

column contains the final classification accuracy in the validation dataset and indicates some learning capability in untrained data. During our experiments we noticed that different points of the same dendrogram might result in similar communication rates, i.e., there might exist different clustering options which achieve the same rate. Therefore, for the test dataset, we show the *top-1*, *top-3* and *top-5* classification accuracy. Despite the fact that performance in *top-1* accuracy might be considered poor, the *top-5* results are, often, above 90%. Finally, the last row compares the communication rate decay (in %) if using the *top-1* option from the NN. Results show that, except in the cases where $N > M$, on average, the rate drops 2.5% which is an acceptable loss when compared to the complexity of the original problem. Moreover, we can infer from these results that the NN is capable of learning the maximum clustering option or clusters that approximate this option. In other words, it is capable to learn the relationship between different users directly from their channel matrices and cluster the users with a high degree of accuracy for most scenarios and finally achieve a rate comparable to more complicated similarity-based HC-HRS.

3.6 Conclusion

In this chapter, we have proposed NN based anticipatory optimization technique that learns and clusters users based on instantaneous noisy channel to maximise the rate achieved using Hierarchical Rate Splitting mechanism. The proposed technique is defined based on a shallow NN architecture thereby making it extremely quick to learn and cluster the users based on the instantaneous noisy channel. The proposed technique is able to achieve a rate comparable with current works while being less complex compared to other techniques. Furthermore, this also helps to investigate further complex NN structures such as Graph NN which can learn covariances between different users to define clustering.

FANETs- Introduction and Challenges

4

4.1 Introduction

Unmanned Aerial Vehicles (UAVs) are used in a wide range of applications, from tracking and monitoring animals in remote areas[163] to military applications[164]. In general, drones are used to search, identify and monitor interesting events over massive and/or inaccessible areas. In order to effectively accomplish the task, multiple UAVs are deployed in a certain area and are expected to coordinate actions in an autonomous fashion or execute direct instructions from a control center.

In many scenarios, the UAVs need to exchange a relatively large amount of data among themselves and/or with the control station to support a given service. For example, distributed area monitoring/patrolling applications may require the UAVs to stream high definition video or thermal camera recordings to the control station, which demands wideband communication technologies (e.g., mmWave) that typically have limited coverage range. Therefore, providing such services over wide areas may require multi-hop data connections, where the UAVs themselves can act as relays for other nodes in the network.

On the other hand, UAVs and the control station also need to exchange light control traffic, which usually has strict latency and reliability constraints, but low transmission speed requirements. This traffic can be carried by low-rate long-range communication technologies, such as LoRa, so as to realize direct links between the UAVs and the control center. For example, this control channel can be used by the UAVs to send periodic tracking updates to the control center, which can use these messages to track the UAVs position [165, 166, 167]. In these scenarios, UAVs and the control center can use different technologies to carry information and signaling traffic, physically separating the data and control planes. However, the randomness in the drones' movements makes the design of a multihop routing protocol for the data plane a challenging problem.

4.2 Literature Review

FANETs are similar to Mobile Ad-Hoc Networks (MANETs), but can have a higher node density and higher mobility. Hence, routing protocols for FANETs have to take into account this node density and mobility to devise routes. While

traditional routing protocols for MANETs, such as Ad-Hoc On Demand Distance Vector Routing (AODV)[168] and the OLSR [169], solve the issue by flooding, we should consider any information on the future position of nodes and the available tracking data so as to improve communication reliability. In this section, we provide a brief overview of the state of the art in the field, focusing on position- and link existence-based routing protocols.

4.2.1 Position-based routing protocols

Since UAV position tracking models are themselves a relatively new research topic [170], there have been few attempts to use this information for routing. Disseminating the trajectory information throughout the FANET in a timely manner is another important issue, as routing is typically distributed and updates need to be frequent.

Some of the protocols for FANETs in the literature are direct extensions of well-known MANET routing protocols: the On-Demand Routing With Boids of Reynolds Protocol (BR-AODV)[171] introduces an AODV-like protocol based on mobility management of UAVs. In this, a technique is presented to restrict and model position and mobility in order to prolong the existence of a link between UAVs when it is active, i.e., when data is being transferred on the link, thereby maintaining communication for a comparatively long time. The mobility restriction messages are incorporated in the HELLO message of the AODV protocol to minimize the signaling overhead. An extension of OLSR based on Prediction of Mobility and Delay (OLSR-PMD)[172] uses GPS position information to make mobility prediction, applying a Kalman filter on the position and velocity of nodes. Following that, Multi-Point Relays (MPRs) are chosen based on maximizing link lifetime and the number of next hop nodes, i.e., the reachability of the destination.

Other protocols are designed from scratch. In the Predictive Routing for Dynamic UAV Networks (PR-DUAV)[173], a trajectory prediction model is presented using Durbin's curve. The location and mobility of the neighboring nodes are predicted using Kalman filtering at each node, and the error occurring in the predictions is modeled as a standard normal distribution. Using the location estimates provided by the trajectory predictions, a shortest delay path is chosen for data transfer. Delay estimates are calculated for current locations of the node, while path selection for further data transfer is chosen based on location estimates of neighboring nodes. The Mobility Prediction-based Geographic Routing (MPGR) protocol [174] is a geographical routing protocol based on mobility prediction. The mobility prediction is made by modeling the future position of each UAV as a Gaussian distribution, which is consistent with Kalman-based tracking systems.

The distance between nodes is defined based on the location estimates obtained from the mobility predictions. Routing is then performed in a greedy way so as to maximise the one hop distance to the destination. If the greedy link fails, the routing protocol finds the next hop on the perimeter which is closest to the destination and thereby finds a route to the destination. A similar mobility prediction system is exploited by the Geographic Position Mobility-Oriented Routing Protocol (GPMOR)[175], which uses a Gaussian Markov Model to calculate the distance between nodes. Based on this distance calculations, a metric is defined which determines if the neighbor is in range of the destination. The path with lowest delay is chosen for routing. In GPSR-Adaptive Beacon and Position Prediction (GPSR-ABPP)[176], the protocol performs a linear regression analysis to predict the future position of each node. In order to reduce routing overhead, the location updates are transmitted dynamically based on the location predictions. This position information is then used for geographical routing.

4.2.2 Link Existence-Based Routing Protocols

Some protocols in the literature try to explicitly estimate the link existence probability, either enhancing the classic AODV and OLSR or designing new mechanisms.

OLSR With Expected Transmission Count (OLSR-ETX)[177] presents a improved Optimized Link State Routing[178] for ocean UAV networks. An additional metric called Expected Transmission Count, which calculates the link expiration time based on GPS positions of the nodes, is introduced in OLSR. In Link Stability Estimation-based Preemptive Routing protocol for Flying Ad Hoc Networks[179], a link stability estimation is calculated based on three parameters namely Link Quality, Safety Degree and Mobility. Link Quality is determined by successful packet transmission in any direction. Safety Degree is defined as the distance between the nodes based on GPS and AODV HELLO messages[168]. Mobility is defined as the instantaneous relative speed of the nodes with respect to each other. Using these parameters, routes are defined with a link stability threshold for each route so that the protocol switches proactively to other paths when one route goes below the particular link stability threshold. Predictive OLSR (P-OLSR)[180] also presents a similar routing protocol metric as Link Stability Estimation-based Preemptive Routing protocol for Flying Ad Hoc Networks[179]. The only difference is that the protocol metric is used in OLSR. Also, Improved OLSR (i-OLSR)[181] presents a routing metric similar to Link Stability Estimation-based Preemptive Routing protocol for Flying Ad Hoc Networks[179]. It also introduces a pursuit mobility model that calculates the position estimates based

on old position and node acceleration in a certain direction, scaled by a random offset vector.
In Robust and Reliable Predictive Routing (RRPR)[182], link connection time are estimated based on node velocity vector. Based on this link estimates, the routing path metric is defined as the combination of intermediate link connection times, hop count and risk, i.e. the residual energy and location of the node and the path with smallest routing path metric is chosen as the route. Directional communication can be performed using the position information of the nodes. This directional communication cone size depends on the distance between the two nodes. As the routing path is approaching its expiration time, the protocol proactively sets alternate routes up to maintain communication. The Optimized-Hybrid Wireless Mesh Protocol[183] is a hybrid mechanism for routing for mobile nodes where some nodes are assigned as Relay Nodes which are connected to control center. The routing between the relay nodes is proactively defined by the control center. The nodes other than the Relay Nodes, setup routes to the control center by reactively finding routes to the nearest relay nodes using Radiometric AODV (RM-AODV). In the Topology-Aware Routing Choosing Scheme (TARCS)[184], the topology of the whole network is maintained by exchanging information between neighboring nodes. The system then chooses between multiple routing protocols based on the node mobility and topology changes. The Geolocation-Based Multi-Hop Routing Protocol (GLMHRP) [185] is a location and velocity-based greedy routing protocol in which location predictions are based on periodic updates from neighboring nodes. Finally, a recent work [186] presents a bio-inspired routing protocol using bee optimization, in which the routing tasks are categorised for three tasks: namely, scouting, route determination, and resource foraging, i.e., data transfer. The protocol has a relatively small overhead compared to other standard FANET protocols.

4.3 Challenges

Hence, a FANET design differentiates from the more classical MANET, featuring different wireless communication technologies with almost orthogonal properties that have to be jointly managed in order to fully exploit the potential of the network. Using the SDN Paradigm, the control center, which can get the most complete view of the state of the network, can determine and propagate routes in a centralized fashion through the low data rate long range control channel [187]. Nonetheless, routing in FANETs is complex even if the control center knows each UAV's position at a given instant, as the dynamic, three-dimensional nature of a swarm makes maintaining stable routes a difficult problem. Most

routing protocols for FANETs as shown in the previous sections is an extension to the traditional MANET protocols such as AODV and OLSR. Moreover, the need for higher data rates in next generation wireless networks leads to incorporation of high bandwidth mmWave technologies, which comes at the cost of a significantly higher path loss, reducing the effective range of communication. To mitigate this issue, mmWave systems often use beamforming techniques that can direct the signal towards the receiver, wasting less power on other directions and reducing interference. However, accurate beamforming requires accurate knowledge of the transmitter's and receiver's positions, which can be very challenging in a FANET because the UAVs are moving, often at relatively high speed, and can only rely on imperfect sensors to measure their position. The usage of beamforming with imperfect information in conjunction with uncertainty in position introduces significant challenges such as if the distance between two UAVs is small, the effect of the positioning error on the beamforming angle is proportionally larger. Hence, the communication challenges introduced due to UAV mobility as well as usage of mmWave technologies provides a very interesting research problem.

Routing Techniques to Improve Route Stability and Robustness in FANETs

5

5.1 Introduction

Routing design in FANETs has had a considerable amount of work on position- and link existence-based routing; the literature review presented in the Chapter 4 does not cover the whole field, and for a more in-depth review of the literature on FANET routing protocols, please refer to the surveys [188, 189]. However, random position modeling has rarely been considered, as the GPS and mobility information was almost always considered correct by the algorithms. The routing protocol should deal with the inevitable estimation error and the relatively low frequency of position updates, or it will not be able to follow the fast dynamics of UAVs. Another factor that should be taken into account is that links involving a common node are not independent: OLSR-PMD [172] and MPGR [174] are the only protocols to jointly consider successive hops. Finally, RRPR [182] was the only protocol to consider additional backup routes to increase the reliability of data transfer. Though, RRPR only uses backup routes when the path expiration is near, i.e., when the weakest link between the intermediate nodes is about to expire. Hence, in this chapter, we propose a system which chooses multiple routes, avoiding single points of failure as much as possible, as this possibility has not yet been explored in FANETs [188].

5.2 Route Existence Design

We model a FANET as a time-varying graph $G(t) = (V, E(t))$, where V is the set of UAVs in the network and $E(t)$ is the set of existing links at time t. Each drone i is characterized by its position $\mathbf{x}_i(t) = (x_i(t), y_i(t), z_i(t))$ in the 3D space. We define the distance $d_{ij}(t) = ||\mathbf{x}_i(t) - \mathbf{x}_j(t)||_2$ as the Euclidean distance between the two drones. In the following, we consider a link $e_{ij}(t)$ as part of $E(t)$ if the distance between drones i and j is lower than the communication range D (which depends on the communication technology used): $E(t) = \{e_{ij}(t) : d_{ij}(t) \leq D\}$. This simple assumption is justified by the fact that the drones will be in line of sight of each other in most practical applications; however, the model can be extended to more complex scenarios.

In the following, we omit the time notation for readability; the operations described below need to be repeated at each time step, as the nodes in the network move and updated position information becomes available: as such, each routing decision is static, but routes are re-evaluated over time. We assume that the real position $\mathbf{x}_i$ of each UAV is not known by the control station, which keeps an estimate of its Probability Distribution Function (PDF) $p(\hat{\mathbf{x}}_i = \mathbf{x})$ instead. We can now define the link existence probability $P(e_{ij})$ as:

$$P(e_{ij}) = P_D(d_{ij} \leq D) = \int_{\mathcal{B}_D(0)} p(\mathbf{x}_i - \mathbf{x}_j = \mathbf{x}) d\mathbf{x}, \tag{5.1}$$

where $\mathcal{B}_D(\mathbf{x})$ is the sphere with radius D and center $\mathbf{x}$. Let $\mathbf{e}$ denote a path from a source (s) to a destination (d), and $\mathcal{E}_{sd}$ is the set all such routes. We then define the *optimal route* $\mathbf{e}^*$ from s to d as the vector of links that maximize the overall route existence probability:

$$\mathbf{e}^* = \arg\max_{\mathbf{e} \in \mathcal{E}_{sd}} P(\mathbf{e}), \tag{5.2}$$

If all links were independent as typically assumed in the literature, we would have $P(\mathbf{e}) = \prod_{e \in \mathbf{e}} P(e)$. Note that, loops are always avoided, as a route with a loop always has a lower or equal probability of existence than the same route without the loop. In this paper, furthermore, we model the joint existence probability of adjacent links, which slightly complicates the expression of $P(\mathbf{e})$, as explained later. Once we have found the optimal route $\mathbf{e}^*$, we can define its *optimal backup* as the route $\mathbf{b}(\mathbf{e}^*)$ that maximizes the success probability when the first route fails (an event denoted by $\bar{\mathbf{e}}^*$):

$$\mathbf{b}(\mathbf{e}^*) = \arg\max_{\mathbf{b} \in \mathcal{E}_{sd} | \bar{\mathbf{e}}^*} P(\mathbf{b} | \bar{\mathbf{e}}^*). \tag{5.3}$$

where $\mathcal{E}_{sd}|\bar{\mathbf{e}}^*$ indicates the set of a viable paths from source s to destination d, given that the primary path $\mathbf{e}^*$ is disrupted. We can generalize the notion of backup route to compute the optimal backup to a set of existing routes, considering the best route if the existing ones all fail. In the following subsections, we report the derivation of the route existence probability, along with the Stochastic Multipath Routing for FANETs (SMURF) algorithm to calculate the primary and backup routes.

5.2.1 Link existence probability

We now assume that the estimated position distribution for each node is a multivariate Gaussian distribution, $\hat{\mathbf{x}}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. This assumption is justified if the tracking system uses Kalman filtering, as is common in the literature [167]. We also assume that the positions of the UAVs are mutually independent. Note that, the covariance matrix $\boldsymbol{\Sigma}_i$ is not necessarily diagonal, as we expect a higher error in the direction of movement of UAVs. The PDF of the position for the node i is given by:

$$p_i(\mathbf{x}) = \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}_i|}} e^{\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_i)\right)}. \tag{5.4}$$

Hence, the link existence probability as expressed in (5.1) is given by:

$$P(e_{ij}) = \int_{\mathcal{B}_R(0)} \frac{e^{\left(-\frac{1}{2}(\mathbf{x}-(\boldsymbol{\mu}_{i-j}))^T \boldsymbol{\Sigma}_{i-j}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{i-j})\right)}}{2\pi\sqrt{|\boldsymbol{\Sigma}_{i-j}|}} d\mathbf{x}, \tag{5.5}$$

where $\boldsymbol{\mu}_{i-j} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_{i-j} = \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j$, as the difference of two independent multivariate Gaussian random variables is itself multivariate Gaussian with those parameters. This integral cannot be solved analytically, but it can be computed efficiently using numerical methods.

We now consider the existence probability $P(e_{ij}, e_{jk})$ of the two-hop path (e_{ij}, e_{jk}). The two links are correlated because they share the intermediate node j. Given that the positions $\mathbf{x}_i$ and $\mathbf{x}_k$ are mutually independent, the links' existence probabilities become independent when conditioned on $\mathbf{x}_j$. Hence, applying the total probability law, we get:

$$P(e_{ij}, e_{jk}) = \int_{\mathbb{R}^3} P(e_{ij}|\mathbf{x}_j = \mathbf{x}) P(e_{jk}|\mathbf{x}_j = \mathbf{x}) p_j(\mathbf{x}) d\mathbf{x}, \tag{5.6}$$

where $P(e_{ij}|\mathbf{x}_j = \mathbf{x})$ is given by:

$$P(e_{ij}|\mathbf{x}_j = \mathbf{x}) = \int_{\mathcal{B}_R(\mathbf{x})} p_i(\mathbf{y}) d\mathbf{y}. \tag{5.7}$$

All the computations above are so reduced to the calculation of multivariate Gaussian integrals, which can be performed efficiently with well-known numerical methods [190, 191]. Since the routing algorithm is executed by the control station, which should have sufficient computational power, there are no issues with the limited battery and computational capabilities of the UAVs.

5.2.2 Routing Metric Calculation

In order to compute the existence probability of a route, we need to consider all of its links jointly. In the following, we simplify the probability calculation by assuming that links that do not share nodes are independent, so that we can write:

$$P(\mathbf{e}) = P(e_{12})P(e_{23}|e_{12})\dots P(e_{n-1,n}|e_{n-2,n-1}). \tag{5.8}$$

This simplification is justified by the fact that each UAV's movement is assumed to be independent, so that it is reasonable to expect that the mutual dependence of links that do not share nodes is negligible. Considering more faithful approximations is possible, but left to future work. By considering only the dependence on the immediately previous link, we can efficiently build a spanning tree by using the negative logarithm of the link existence probability as a routing metric:

$$W(e_{jk}|e_{ij}) = -\log_{10}(P(e_{jk}|e_{ij})). \tag{5.9}$$

In this way, links with a higher existence probability are chosen by the routing algorithm.

5.2.3 Backup Routes Calculation

In a dynamic scenario, the primary route, which by definition is the route with highest probability of existence, can still fail. To mitigate the impact of the failure on communication and increase the reliability of the transmission, we devise a set of backup routes, which can be selected in case the primary one fails. The increase in reliability of the transmission is majorly dependent on the density of the FANET. If the FANET is dense enough, there will be multiple viable routes to the destination. So, in order to calculate the optimal backup, we consider single-link failures and define the conditional path existence probability, given the link between nodes i and j is down, as follows:

$$\mathbf{b}_{i,j}(\mathbf{e}^*) = \underset{\mathbf{b}\in\mathcal{E}|\bar{e}^*_{i,j}}{\arg\max}\, P(\mathbf{b}|\bar{e}^*_{i,j}). \tag{5.10}$$

where $\bar{e}^*_{i,j}$ denotes the event where the link between i^{th} and j^{th} nodes does not exist. Considering the event $\bar{e}^*_{i,j}$, we can compute the conditional joint position PDF of nodes i and j, as:

$$p((\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) = (\mathbf{x}, \mathbf{y})|\bar{e}_{i,j}) = \begin{cases} \frac{p_i(\hat{\mathbf{x}})p_j(\hat{\mathbf{y}})}{1-p(e_{i,j})}, & \hat{\mathbf{y}} \in \mathcal{B}_R(\hat{\mathbf{x}}); \\ 0, & \hat{\mathbf{y}} \notin \mathcal{B}_R(\hat{\mathbf{x}}); \end{cases} \tag{5.11}$$

where, $p(e_{i,j})$ is the link existence probability between nodes i and j. We can then adjust other links' existence probabilities with such a conditional PDF and rebuild the spanning tree to find the backup route. After computing the optimal backup $\mathbf{b}_i(\mathbf{e}^*)$ for each link failure, we compare them by considering the probability of the link failing. The optimal backup route is then given by:

$$\tilde{\mathbf{b}}(\mathbf{e}^*) = \underset{\mathbf{b}\in\{\mathbf{b}_1,\ldots,\mathbf{b}_{N(\mathbf{e}^*)-1}\}}{\arg\max} P(\mathbf{b}|\bar{e}^*_{i,j})(1-p(e_{i,j})), \tag{5.12}$$

where $N(\mathbf{e}^*)$ is the number of nodes in route $\mathbf{e}^*$. We compute successive backups by considering single broken links in the primary route to simplify the calculation, even though the result is slightly suboptimal. The calculation of the backup routes can be extended to longer time horizons in the same way we outlined for the primary route.

5.2.4 Route information propagation and data plane

In our model, routing calculations are performed by the central control station, which collects tracking information using LoRaWAN [167] and computes the routes. This centralized strategy provides two key benefits: first, the information collection and decision-making is in one place, so that the routing protocol inherently avoids loops and does not operate on contradictory information. Second, UAVs are spared from the computational load to perform numerical integration and calculate the route existence probability. The central node is not so constrained, and can even offload computation to the cloud.

The propagation of the routes to UAVs can be performed via SDN [192]: this paradigm involves a central controller gathering information and sending simple instructions to switch nodes, and it has already been proposed and tested in FANETs [193]. SDN also gives UAVs the possibility to send data over multiple interfaces, so the routing protocol could be implemented in an entirely transparent way, without requiring changes to the applications transmitting the data. Furthermore, SDN has been widely deployed in wired networks, and all major operating systems now support it by default.

The use of backup routes can increase the reliability of the transmission, but there is a trade-off: if packets are sent over multiple routes by multicast, this requires additional energy consumption and increases the load on the network. Choosing the optimal point in this trade-off is beyond the scope of this work, but the choice can be performed by tuning the number of backup routes that the nodes can use, which we choose as a system parameter in our analysis.

Parameter	Value
Number of Nodes	Poisson Point Process
PPP Density λ [UAVs/km^2]	150
Number of Scenarios	100
Number of Realizations per Scenario	100
Communication Range [m]	100

Table 5.1: Parameters of the Simulations

Naturally, this static procedure needs to be repeated over time as the UAVs move and new information about their position becomes available. The speed of the position information propagation should be large enough to limit the delay between the generation of the positioning update and the routing decision: route quality degrades if the controller uses outdated information, particularly if the UAVs are fast.

5.2.5 Simulation and Results

Simulation Analysis is performed in MATLAB environment. We define a Monte Carlo based simulation, where we generate 100 different network scenarios, which correspond to 100 possible combinations of different number and estimated positions of UAVs in the system. The routing algorithm computes the route for each scenario, and we consider 100 different realizations of the graph: each realization denotes a particular set of node positions, and can be considered as a different possible evolution of the network for the given tracking information scenario. Routing decisions taken at instant t are then evaluated over the next timestep $t + 1$. Due to this difference, not all network realizations are fully connected, and even perfect routing cannot achieve full reliability. The routing reliability in a given scenario is then measured as the fraction of realizations in which at least one of the chosen set of routes exists between node 1 as source and randomly chosen destination.

We consider a communication range $D = 100$ m, which is realistic for UAV line of sight mmWave propagation in urban scenarios [194]. For each scenario, the number and average positions (i.e. the means of multivariate Gaussian distribution that describe the positions across different realizations) of the nodes is generated by a spatial Poisson Point Process (PPP). The elements of the covariance matrices of the Gaussian distributions, instead, are drawn from a uniform distribution in [0,1], using rejection sampling to ensure the matrix is symmetric semidefinite positive. The covariance matrix is then multiplied by D^2. Table 5.1 shows the parameters used to generate the simulation environment. We also consider two other protocols for comparative analysis, namely,

- Mobility based Geographic Routing Protocol (MPGR) - Protocol from [174] that computes routes based on mobility of the UAVs;
- Distance - Distance based routing protocol that uses the tracking information obtained from the Kalman Filter to compute routes.

Additionally, we also considered the theoretical oracle routing protocol, which operates with perfect information about the UAVs' positions and, consequently, the existence of the routes.
To evaluate all the protocols, we compute Packet Delivery Ratio (PDR) by considering the existence of paths between all the nodes in primary and/or backup routes. If at least one of the selected routes exists for a particular realization of the distribution, we consider the packet to be delivered correctly, while when none of the selected paths exists, the packet delivery fails. Additionally, the packets are multicasted over the primary and backup routes so as to avoid additional energy consumption due to duplication of packets on links shared by such routes. Finally, we have assumed that there is no radio interference between the packets sent on multiple routes. We evaluate the delivery for every realization in every scenario and plot it as a boxplot. Figure 5.1 shows the PDR when incorporating multiple routes from source to destination, with a density $\lambda = 150$ UAVs/km^2.
The boxplot shows the empirical distribution of the PDR for the given scenario. The average PDR for SMURF is significantly higher even when considering a single path: while the distance-based protocol has an average PDR of 85% and MPGR is below 40%, SMURF can achieve 93%. MPGR performs even worse than the distance-based metric, as shown in the figure. Furthermore, the lower quartile edge of the boxplot shows that the distance-based protocol performs far worse in the worst cases, with 25% of possible network graphs having a PDR of 70% or lower, while single-path SMURF achieves an 88% PDR for the 25th percentile graph. This gain comes from the fact that SMURF computes the route existence probability accounting for the inter-dependency of adjacent links. The gains become even more significant when considering multiple routes: using 2 paths improves the average PDR to about 98%, and 3-path SMURF already has a performance very similar to the oracle routing protocol i.e. maximum achievable routing performance when all positions are accurately known by the control center. This justifies the need for backup routes, but also limits the algorithm to a small number of alternative paths, which helps reduce the overhead due to packet replication. We also tested the protocol performance for different scenarios, changing the rate parameter of the PPP to observe the effect of node density on the performance of the protocol. Figure 5.2 shows the average PDR obtained by

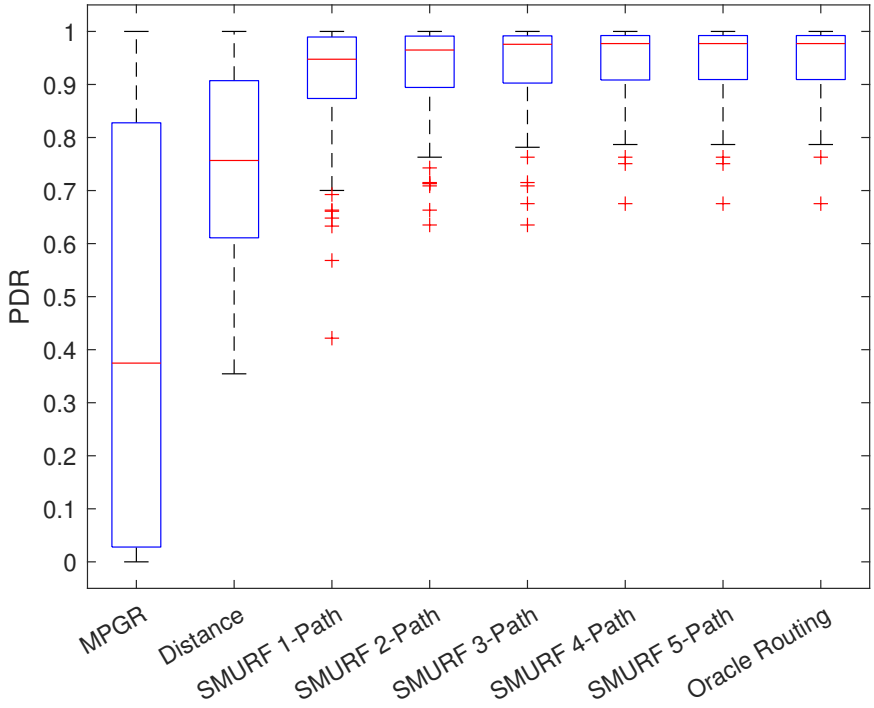


Figure 5.1: Comparison between MPGR, Distance Based Single Path Algorithm and SMURF with different number of paths

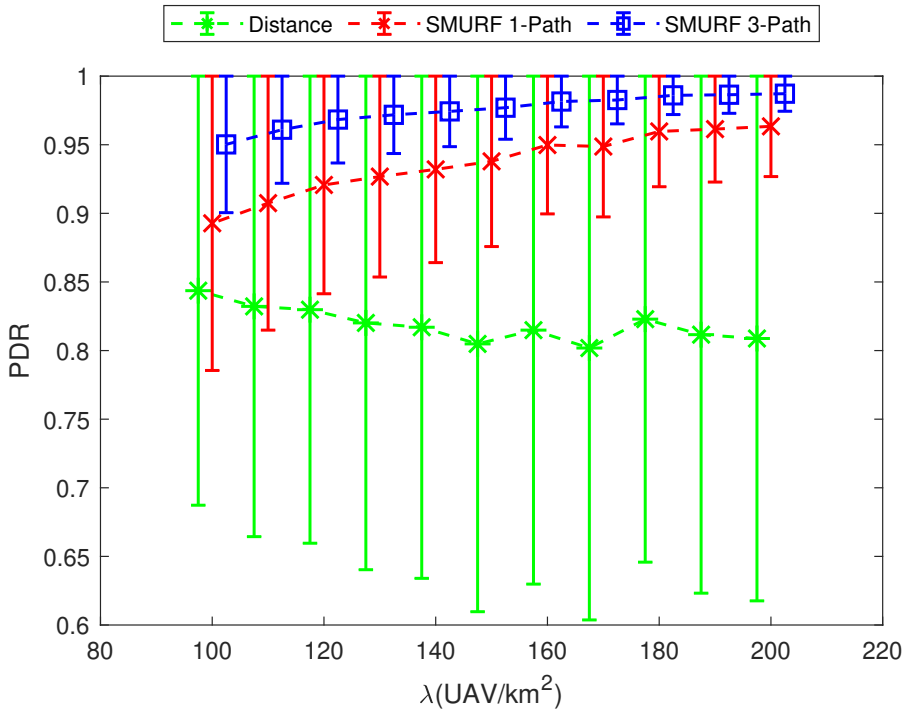


Figure 5.2: Average PDR as a function of the node density λ

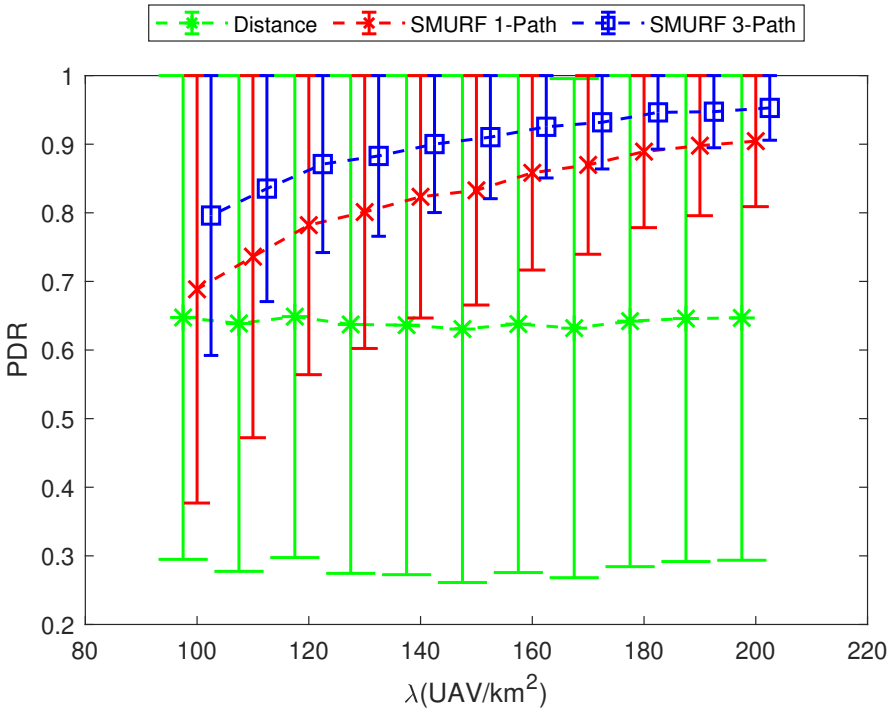


Figure 5.3: 25th Percentile PDR as a function of the node density λ

the distance-based single-path algorithm, single-path SMURF and 3-path SMURF, as a function of the parameter λ, which represents the node density.

As shown in Figure 5.2, SMURF's average PDR increases with increasing node density, and 3-path SMURF does significantly better than the single-path version. However, SMURF's PDR is already above 89% even when $\lambda = 100$ UAVs/km^2, and the gain from using backup routes increases as node density decreases: the chances of the primary route being interrupted are much higher, and the backups can help more. The distance-based algorithm performs far worse, both for very high and very low densities: if the density is high, it makes many more hops, increasing the chances that one link will not be available, while if the density is low, its less accurate characterization leads it to make more mistakes. The plot also shows the confidence interval for each algorithm: SMURF's performance is better than the distance-based routing algorithm's by more than 2 standard deviations for $\lambda \geq 120$ UAVs/km^2, so the gains are statistically significant. However, the confidence intervals for distance-based single-path routing and single-path SMURF have significant overlap for lower variables of λ, and the advantage of SMURF in this scenario is mainly its better predictability, i.e., having a more consistent PDR in different realizations. Figure 5.3 shows the PDR in the 25th percentile graph for different node densities. In this case, the trend is similar for

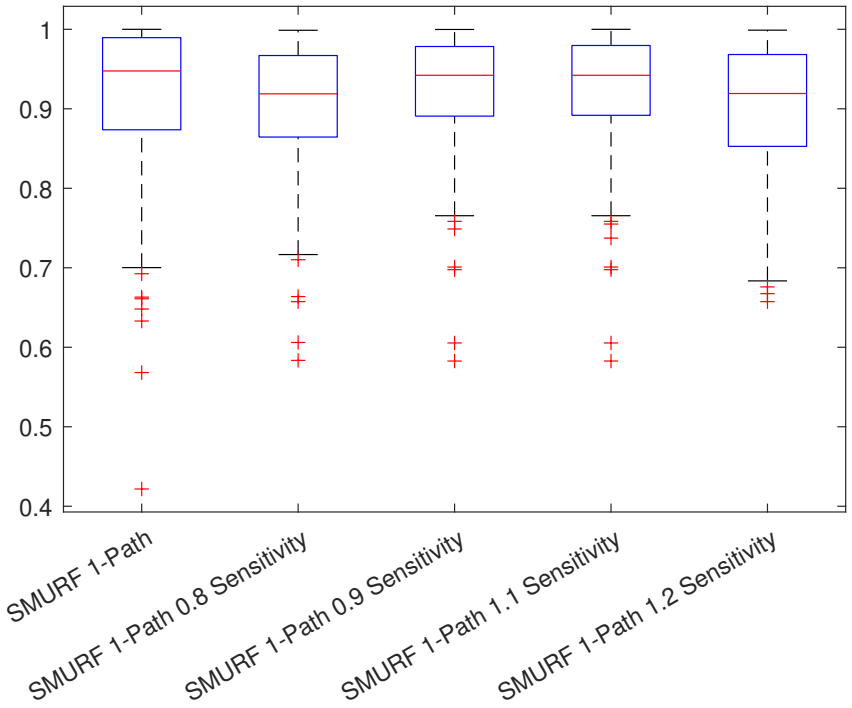


Figure 5.4: SMURF Sensitivity for -20%,-10%,10%,20% error on the estimated variances

all the considered algorithms: the dominating factor is the existence of at least one route, as we are considering worst-case scenarios.

As shown in the Figure 5.3, even in worst case scenarios with least density i.e. $\lambda = 100$ UAVs/km^2, SMURF outperforms the single path algorithm by almost 8% for single path and almost 15% for three paths. A common issue in Kalman-based systems is the estimation error, as the Gaussian predictions derived from the filter can over- or underestimate the variance. To understand the effect of such estimation errors, we performed a sensitivity analysis of the protocol for different over- and underestimations of the variance. Figure 5.4 shows the sensitivity analysis of SMURF when the protocol uses a variance shifted by -20%, -10%, 10%, and 20%. Performance losses are very small even in the case of significant variance of the estimation errors. Hence, SMURF is tolerant of such errors occurring in Kalman filter predictions.

5.3 Route Stability Design

The SMURF routing protocol discussed in the previous subsection is shown to be useful for single timestep routing decisions. But, it is crucial to evaluate the performance of the protocol over an entire flight duration of the UAVs. To do so, we extend the SMURF routing protocol defined by introducing a continuous

gradual mobility of the UAVs in the FANET. To introduce the effect of uncertainty due to UAV mobility, we consider the uncertainty in Kalman filter predictions defined in Sec. 5.2 as a function of time specifically dependent on the last tracking update received from the UAVs i.e. the uncertainty in the Kalman filter predictions is smallest in the next timestep after receiving the update and it increases based on the speed and estimated direction of travel till a new update is received. We reuse the definitions of position and link existence probabilities from Sec. 5.2 and compute them over each timestep t thereby obtaining the link existence probability over the entire mobility duration of the UAVs. For reader's ease, we denote the link existence probability over each timestep with subscript t notation.

5.3.1 Routing Metric Calculation

In order to compute the route existence probability for each timestep, we consider the same assumptions as before, i.e., links that do not share nodes are independent. For simplicity, we numbered the UAVs from source s to destination d in a sequential order. Therefore, for a path $\mathbf{e} = [e_{1,2}, e_{2,3}, \ldots, e_{n-2,n-1}, e_{n-1,n}]$, taking into account the link existence probability computed in Sec. 5.2 for timestep t, we can write:

$$P_t(\mathbf{e}) \simeq p_t(e_{1,2}) p_t(e_{2,3}|e_{1,2}) \ldots p_t(e_{n-1,n}|e_{n-2,n-1}). \tag{5.13}$$

Note that, the calculations above refer to a single timestep, but the Kalman filter-based location prediction can be extended to the next T timesteps. We can then extend the prediction to a vector:

$$\mathbf{p}_t^{(T)}(e_{i,j}) = \{p_t(e_{i,j}), p_{t+1}(e_{i,j}), \ldots, p_{t+T-1}(e_{i,j})\}. \tag{5.14}$$

In the same way, we can compute $\mathbf{P}_t^{(T)}(\mathbf{e})$ for any route as a vector of $P_t(\mathbf{e})$ for $[t, t+T-1]$ timesteps. Naturally, this is a slight simplification, as the error on the position of the nodes is not independent over time, but accumulates, so there is a positive correlation between the existence probability at one step and the next. However, in the interest of computational simplicity, we make this further simplification.

To compute the routing metric, we could consider any function of the vector $\mathbf{P}_t^{(T)}(\mathbf{e})$, but we take the average over the time horizon for simplicity's sake. Since the route existence probability at any given timestep is computed considering only the dependence on the immediately previous link, we can efficiently build a

spanning tree by using the negative logarithm of the link existence probability as a routing metric to minimize:

$$\mathrm{W}(e_{j,k}|e_{i,j}) = -\log\left(\frac{\sum_{\tau=t}^{t+T-1} p_\tau(e_{j,k}|e_{i,j})}{T}\right). \tag{5.15}$$

In this way, links with a higher existence probability over the whole timeframe are chosen by the routing algorithm. Additionally, the calculation of the backup routes computed in Sec. 5.2 can be extended to longer time horizons in the same way we outlined for the primary route.

5.3.2 Route information propagation and data plane

As discussed in Sec. 5.2, in our system model, routing computations are carried out by the central control station, which receives tracking information over long distance low bit rate channel such as LoRa [167] and defines the routes.

Also, depending on the frequency of the updates and the mobility patterns devised from the updates, the central controller can adapt the frequency of routing updates to the UAVs. However, the speed and frequency of the position information propagation and the routing updates should be sufficient to limit the delay between the generation of the positioning update and the routing decision as route quality can potentially degrade very fast if the controller uses outdated information, particularly if the UAVs move fast.

5.3.3 Simulation and Results

The simulation environment is set up in Python. The UAVs are autonomously moving in the area, i.e. they are not explicity controlled by the control center, which only tracks their position through the LoRa updates. UAV positions are determined based on the unscented Kalman filter predictions. Two additional protocols, namely, AODV and Distance-based SDN routing, are considered for the sake of comparison against SMURF. The AODV simulation is performed under the assumption that each UAV has perfect information regarding the positions of all other UAVs in the network at the time of network route computation. This assumption is not entirely realistic and leads to an upper bound to that a practical implementation of AODV would achieve in this scenario. In the evaluation, we consider two variants of position aware AODV protocol, namely,

- *AODV-L* - AODV with Long Route Timeout, which calculates the route every N timesteps (N is equivalent to the prediction window of SMURF);

Simulation Parameter	Simulation Value
Map size [m]	$200 \times 200 \times 20$
Density of UAVs [UAVs/km^3]	{20k-60k}
UAV motion model	CTRA 3D [167]
UAV tracking model	Unscented Kalman Filter
Transmission range (D) [m]	75
Prediction window [s]	{1,5,10}
Number of simulated networks	100

Table 5.2: Parameters of the Simulations

- *AODV-S* - AODV with Short Route Timeout, which calculates the route at each timestep. This improves the reliability but involves a large cost in terms of control traffic to update the routing tables.

We remark again that both versions of AODV are idealized, i.e., have information that UAVs in an actual network would not have, and provide an upper bound to the achievable performance.
In addition, we consider the distance-based SDN routing, henceforth referred to as Distance, which gets regular position updates from the UAVs and calculates the shortest routes based on the Kalman filter predictions of the UAV positions. We also consider a simple disk model for the communication, since the UAVs are in free space and the path loss is a direct function of the distance: if the distance between two UAVs is lower than the communication range, the link can be used. Additionally, we determine a general upper bound to the achievable performance, henceforth referred to as *Upper Bound*, computed via Monte Carlo simulation (the mobility of the UAVs makes it very difficult to compute the performance analytically). The upper bound is computed by considering perfect causal position information, so that the controller knows both where the UAVs are and where they will move in the future. The simulation parameters are given in Tab. 5.2.
To examine the performance of the protocols, we use two different metrics, Route Availability Ratio (RAR) and Route Establishment Overhead (REO):

$$\text{RAR} = \frac{\text{Route Availability Time}}{\text{Total Simulation Time}};$$
$$\text{REO} = \frac{\text{Number of Routing Packets}}{\text{Number of UAVs} \times \text{Total Simulation Time}}.$$

The RAR metric is used to determine the availability of at least one path between source and destination, while the REO gives the overall number of routing packets exchanged per timestep for any network size, and hence provides a measure of the protocol overhead.

The simulation results, hence, are divided in two subsections. In the first subsection, the protocols are evaluated using the defined metrics for different network densities and for different sizes of the prediction window, while in the second section, we focus only on the impact of backup routes on the RAR metric, as REO, which is SDN route control information sent over LORA is the same for single path and multipath transmissions.

5.3.3.1 Route Availability and Route Overhead Comparison

In this evaluation, we determine the performance achieved for all the protocols using the two metrics defined before. To do so, we set a route expiration timer of AODV-S to 1 second, ensuring that AODV-S will update the route at every second. Routing decisions, hence, are made very often for AODV-S, while we vary the prediction window of the other protocols. The route expiration time for AODV-L is set equal to the prediction window of SMURF and Distance. We recall that SMURF and Distance need to rely on location updates from the UAVs to determine routes and, between updates, they evolve the mobility model in open loop (i.e., blindly). AODV-L and AODV-S are also assumed to have perfect knowledge of the position and mobility of the UAVs in the current timestep, i.e., both the protocols exactly know which links are disconnected, which is an ideal assumption as, in reality, this information can only be known by the protocols when a transmission fails. To provide a fair comparison with the other algorithms, SMURF will only use the primary path. In these simulations, we also control the density of the nodes over the simulation space, which will determine the possible routes. Figure 5.5 provides the RAR of all different routes from one source for the density of 40000 UAVs/km^3. Note that in our scenario, such a density translates to approximately 10 nodes in the 200 m $\times$ 200 m $\times$ 20 m simulation area, representing a moderately dense network. As visible from Figure 5.5, the median RAR obtained by SMURF for prediction window of one and five seconds is significantly higher than the corresponding RAR obtained by AODV-L and AODV-S while its comparable to RAR obtained by Distance. For the 10 second prediction, instead, the median RAR of SMURF is lower than AODV-S, but still higher than the median RAR obtained by AODV-L and Distance. This shows that, as the information gets older, the routes become stale and need to be recalculated based on new location updates from the UAVs. SMURF is then very good at evaluating the routes, as long as the prediction window is relatively short and routes are updated frequently.

We can then look at the effect of the density of the nodes on the reliability of the three algorithms, as shown in Figures 5.6, 5.7 and 5.8. The three figures show

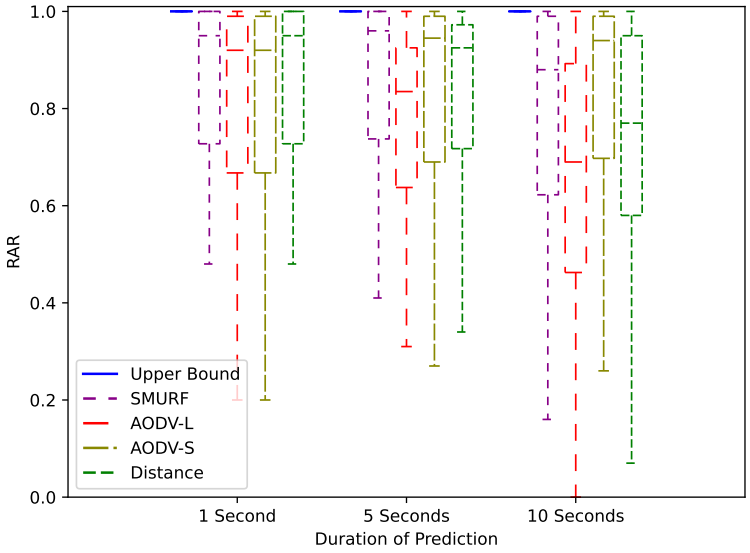


Figure 5.5: RAR comparison for different duration of predictions for a density of 40000 UAVs/km^3.

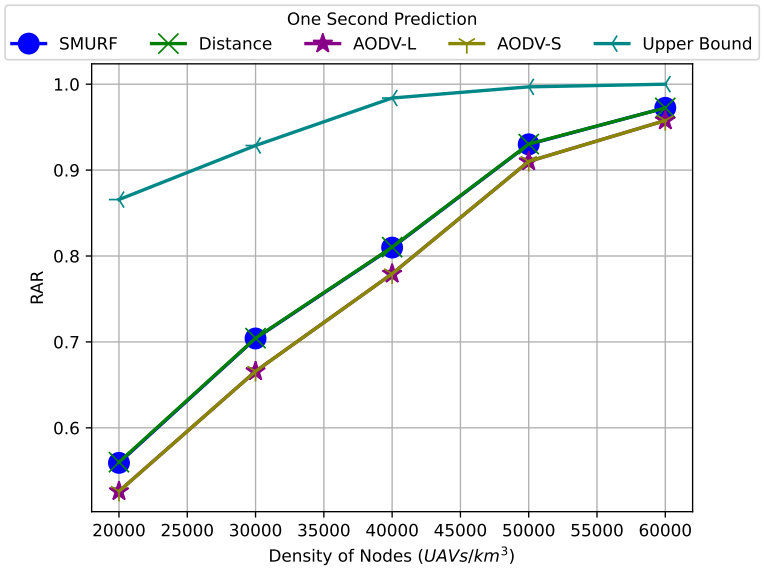


Figure 5.6: RAR as a function of the UAV density for different protocols for the prediction window of 1 s.

the mean RAR as a function of the network density for prediction windows of 1 s, 5 s, and 10 s.

As visible from Figure 5.6, the RAR for SMURF outperforms AODV-S and AODV-L and is equivalent to that of Distance. The reason the pure distance-based prediction performs as well as SMURF is due to the small uncertainty in the UAVs' future positions for the prediction window of 1s. This uncertainty in the position increases when the prediction window is increased to 5s. As visible from

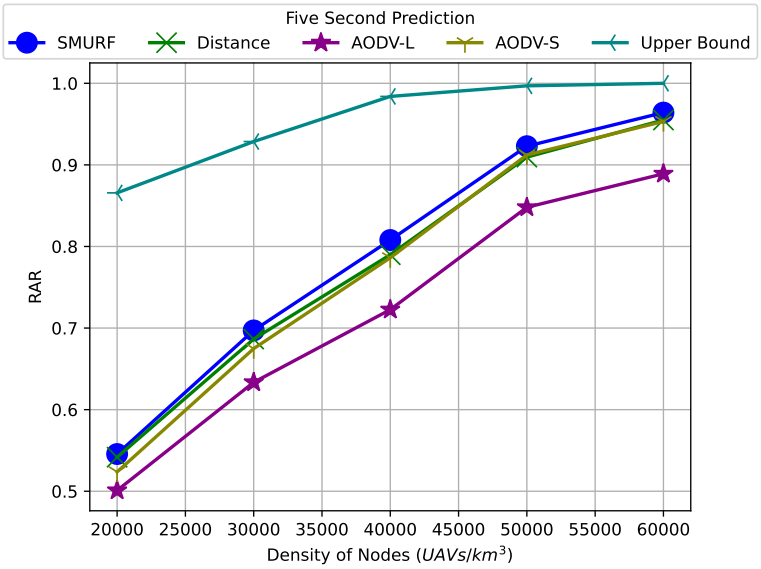


Figure 5.7: RAR as a function of the UAV density for different protocols for the prediction window of 5 s.

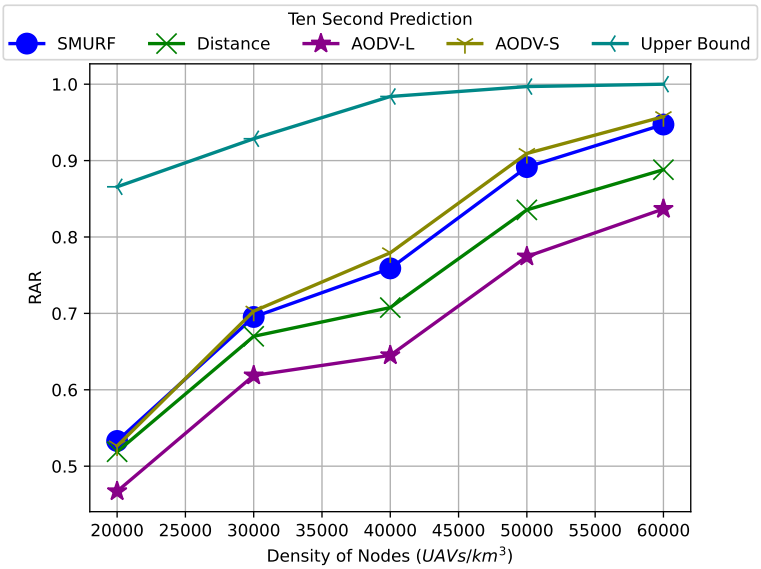


Figure 5.8: RAR as a function of the UAV density for different protocols for the prediction window of 10 s.

Figure 5.7, the RAR of SMURF outperforms all the other protocols at all densities even approaching perfect reliability at higher densities.

On the other hand, SMURF performs worse than AODV-S if the prediction window is 10s, as shown in Figure 5.8. The only case in which SMURF outperforms AODV-S is when the density is very high, reaching 60000 UAVs/km^3. The reason is that with higher densities of UAVs, the SMURF protocol can choose the lowest-uncertainty route, thereby increasing the overall RAR. Additionally, to determine

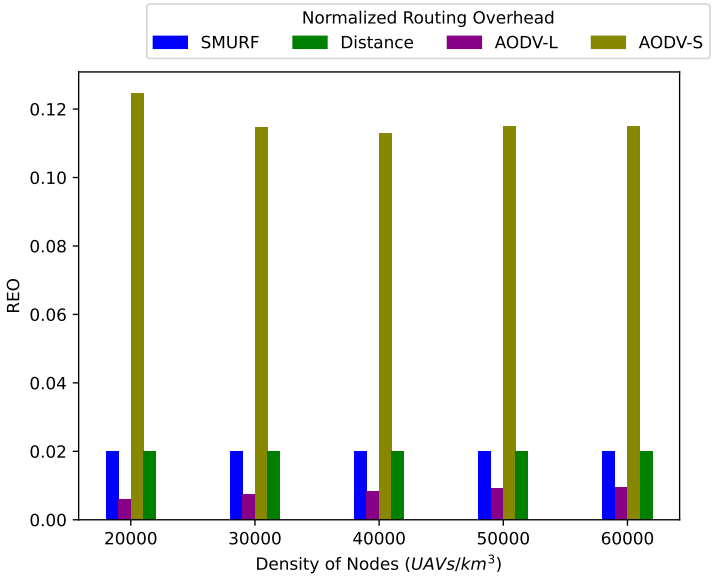


Figure 5.9: REO over different densities for the prediction window of 10 s.

the cost incurred while setting up these routes, we evaluate the metric of REO. Figure 5.9 shows the REO with respect to the different UAV densities.

As visible from Figure 5.9, the REOs for SMURF, Distance and AODV-L are much smaller than that of AODV-S. This is due to the fact that AODV-S has to send route requests and receive route replies at every time the route expires which in this case is every 1 s. This leads to high REO compared to SMURF, Distance and AODV-L. So, even if the RAR obtained by SMURF and AODV-S is similar for a prediction window of 10s, the REO for AODV-S is significantly larger compared to SMURF. Also, note that, AODV-S preemptively knows which links have failed and devises the trajectories which is an ideal case.

Additionally, the REO is constant across all densities for SMURF, Distance and AODV-S while it slightly increases for AODV-L with increasing densities. This shows that even though AODV-L has a smaller REO compared to SMURF, Distance and AODV-S, the RAR achieved is significantly lower for AODV-L. SMURF, on the other hand, has a slightly higher REO compared to AODV-L but provides a significant performance gain in RAR compared to that of AODV-L. Also, note that, AODV-L has preemptive knowledge regarding the links that have failed and devises routes accordingly which is an ideal case for AODV as the protocol can only detect link failures if a transmission fails. Hence, the simulation results show that SMURF can significantly improve the RAR while still maintaining a relatively low REO.

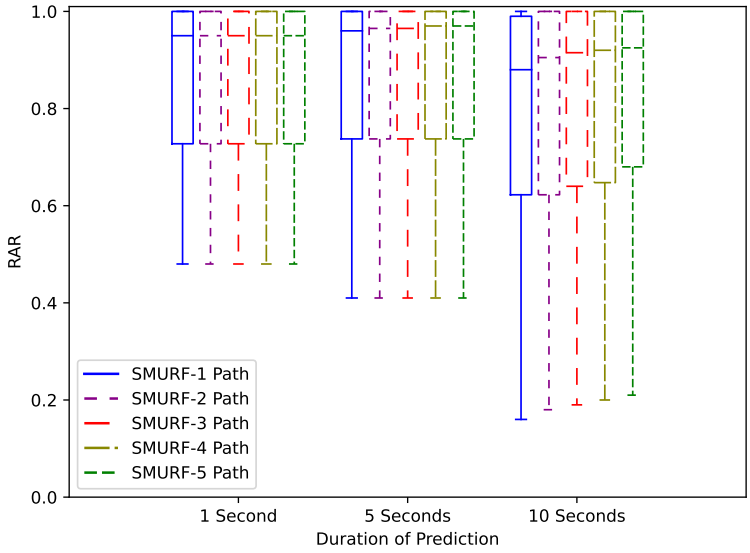


Figure 5.10: RAR for one path to five paths and for different duration of the prediction window and density of 40000 UAVs/km^3

5.3.3.2 Primary and Backup Path Availability Comparison

In this evaluation, we determined the effect of backup routes for the primary route defined by SMURF. Figure 5.10 shows the RAR for different durations of prediction, for a density of 40000 UAVs/km^3, i.e., approximately 10 nodes in the volume of 200 m $\times$ 200 m $\times$ 20 m which is a moderately dense network. For the prediction window of 1 s, the RAR is same from one to five paths, i.e., the backups do not provide any additional gain in terms of route availability. For the prediction window of 5 s, the backups provide a slight gain in mean RAR, i.e., approximately 2%, but in the worst 25^{th} percentiles, the backups do not provide any significant improvement. This, however, changes for a prediction window of 10 s. The backups improve the mean RAR and the worst 25^{th} percentiles by approximately 10%. This denotes that the impact of backup routes increases as the prediction window increases. However as the prediction window increases, the primary and backup routes become stale and thereby the RAR, even with backup routes, goes down. This indicates that the frequency of position updates from the UAVs to the central station can be reduced to a certain extent to update stale routes proactively and maintain similar performance while reducing the number of route computations at the central station as well as frequency of route updates from the central station to the UAVs.

5.4 Conclusion

In this chapter, we briefly describe the routing strategies based on position and mobility predictions that maximise route existence and stability in FANETs. We provide a statistical analysis of a FANET with tracking information, and derive the conditional existence probability for both single links and entire routes for a single timestep and multiple time prediction window. Also, we propose the SMURF protocol, a multipath routing protocol that computes a primary route (i.e., the route with the highest existence probability) and a series of backup routes that allow the transmission to succeed even if a link in the primary route is broken. Through simulations, we compute the system wide performance obtained for different performance indicators which show that SMURF can outperform baseline protocols for different time prediction windows.

Also, we also show that another interesting avenue of further research is to consider realistic propagation models and beamforming designs which can improve the system wide performance while taking into account physical channel attributes.

Joint optimization of beamforming and routing strategies in FANETs

6

6.1 Introduction

Routing, as shown in Chapter 5, is a complex problem in FANETs: even if the control center knows each UAV's position at a given instant, the dynamic, three-dimensional nature of a swarm makes maintaining stable routes a difficult problem. Moreover, usage of mmWave technologies for providing high data rates for UAV networks [195]. But, the usage of mmWave technologies, even with its high bandwidth availability, comes at the cost of a significantly higher path loss, reducing the effective range of communication. To mitigate this issue, mmWave systems often use beamforming techniques that can direct the signal towards the receiver, wasting less power on other directions and reducing interference. However, accurate beamforming requires an accurate knowledge of the transmitter's and receiver's positions, which is not always possible in a FANET as the UAVs are moving, often at relatively high speed, and can only rely on imperfect sensors to measure their position. Additionally, sharing positioning information requires some signaling [167], which can be performed over long-range and low-bitrate technologies.

The use of beamforming with imperfect information (caused due to errors in predictive tracking) introduces another challenge when making routing decisions, as UAVs at a shorter distance might have a higher probability of remaining in range, but also suffer more from beamforming errors: if the distance between two nodes is small, the effect of the positioning error on the beamforming angle is proportionally larger. So, in this chapter, we propose the works that jointly define routing and beamforming based on the position uncertainty.

6.2 Fixed Beamforming for Routing Design

We again describe the connectivity in a FANET as a time-varying graph $G = (K, E(t))$, where K represents the set of UAVs in the network and $E(t)$ the set of active links at time t. Each drone moves independently from the rest and has its attitude characterized by a 5-tuple: the coordinates in space $\mathbf{x}_k(t) =$

$(x_k(t), y_k(t), z_k(t))$ and the yaw and pitch angles. So, the quality of the link between a pair of drones (i, j) depends on their Euclidean distance [196]:

$$d_{ij}(t) = \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|_2.$$

Since UAVs fly high from the ground, the space between them is assumed to be free, i.e., without obstacles or reflections. We can then consider that there is predominantly Line of Sight (LoS) communication [197]. Thus, the pathloss between two UAVs is described by

$$P_L(d_{ij}) = \left(\frac{c}{4\pi f_0 d_{ij}}\right)^{\gamma}, \tag{6.1}$$

where f_0 is the carrier frequency, c is the speed of light, and γ is the path loss exponent. As a result, if two UAVs are nearby, then there exists a link between them. Otherwise, communication becomes impractical due to low SNR. More formally, we can describe the set of active links by

$$E(t) = \{e_{ij}(t) : d_{ij} < D\}, \tag{6.2}$$

where the choice of D often depends on the environment and the wireless technology used for communication [198]. Unfortunately, in more realistic scenarios, the real position of all drones is unknown, but the central controllers maintain estimators to predict the UAVs' positions based on current and previous location updates [167]. Hence, the position estimate is defined as:

$$\hat{\mathbf{x}}_k(t) = \mathbf{x}_k(t) + \mathbf{n}_k(t), \tag{6.3}$$

where the noise $\mathbf{n}_k(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_k)$ is associated to the kth user where $\mathbf{\Sigma}_k$ refers to the uncertainty covariance in the three coordinates. Hence, $\mathbf{\Sigma}_k$ is a symmetric matrix. Note that, even if $\mathbf{\Sigma}_k$ is a symmetric matrix, it is not an identity matrix, i.e. the error in estimate is not equal in all three coordinates. This is due to the fact that the error in estimate will be larger in the direction of movement for the UAV and smaller in other directions. In order to account for this uncertainty, one can also re-write (6.18) in terms of the probability that the two endpoints are within a sphere of radius D. As a result, the link existence probability can be defined as

$$\hat{E}(t) = \{(i, j) : i \neq j; P_{ij}(t)\} \tag{6.4}$$

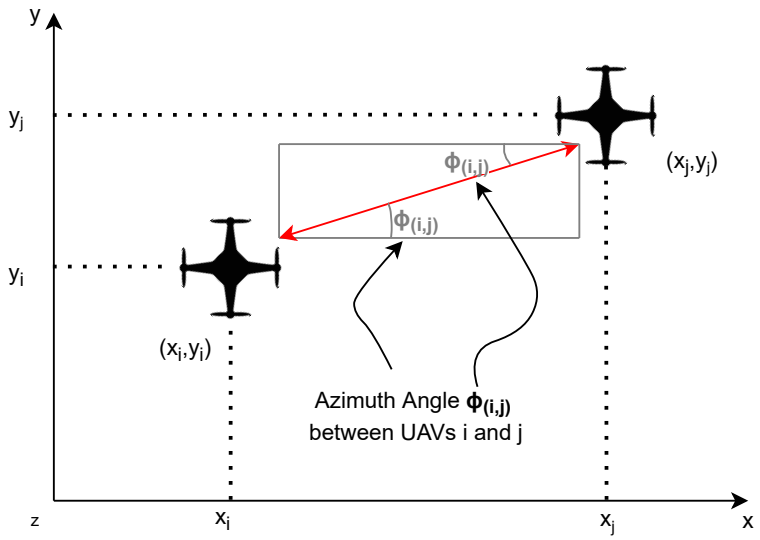


Figure 6.1: 2-D representation of the angular separation between UAVs i and j for perfect knowledge of position and attitude.

where the probability $P_{ij}(t)$ that the link between i and j is active at time t is given by:

$$P_{ij}(t) = \int_{\mathcal{B}_D(0)} \frac{e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\Delta}\hat{\mathbf{x}}_{ij}(t))^{\mathrm{T}}(\boldsymbol{\Sigma}_{ij}(t))^{-1}(\mathbf{x}-\boldsymbol{\Delta}\hat{\mathbf{x}}_{ij}(t))}}{2\pi|\boldsymbol{\Sigma}_{ij}(t)|^{\frac{1}{2}}}\, d\mathbf{x}. \tag{6.5}$$

Here, we have defined $\boldsymbol{\Delta}\hat{\mathbf{x}}_{ij}(t) = \hat{\mathbf{x}}_i(t) - \hat{\mathbf{x}}_j(t)$, $\boldsymbol{\Sigma}_{ij}(t) = \boldsymbol{\Sigma}_i(t) + \boldsymbol{\Sigma}_j(t)$ and $\mathcal{B}_D(0)$ as the sphere with radius D and center in the origin.

To introduce beamforming into the system, we consider each UAV to be equipped with an Uniform Planar antenna Array (UPA) of dimension $M = M_{\mathrm{H}} \times M_{\mathrm{V}}$, where antennas are spaced d_{H} (horizontally) and d_{V} (vertically) wavelengths from one another [199]. This allows each drone to communicate with other devices in the same altitude, as well as, with drones at different heights. As discussed above, due to the nature of the open environment, we assume that there is predominantly LoS communication. In this scenario, the UAVs can also apply beamforming to improve the Signal to Interference plus Noise Ratio (SINR) by increasing antenna gain. To determine the steering vector for beamforming, we need to determine the angular separation between the corresponding UAVs. Figure 6.1 shows the 2-D representation of the angular separation, i.e. azimuth angle, between the UAVs i and j for perfect knowledge of position and attitude. Since $\hat{\mathbf{x}}_i(t)$ and $\hat{\mathbf{x}}_j(t)$ are imperfect estimates, the angle between the estimated position $\hat{\mathbf{x}}_j(t)$ of UAV j with respect to $\hat{\mathbf{x}}_i(t)$ of UAV i is itself a random variable. In the following, we omit the time index t for simplicity. Hence, the steering

vector is defined based on the estimated azimuth $\Delta\hat{\phi}_{ij}$ and elevation $\Delta\hat{\theta}_{ij}$ angles between the jth and ith UAVs, respectively, described by,

$$\Delta\hat{\phi}_{ij} = \pi\mathbb{1}(\hat{x}_j - \hat{x}_i) + \arctan\left(\frac{\hat{y}_j - \hat{y}_i}{\hat{x}_j - \hat{x}_i}\right) - \hat{\phi}_i; \tag{6.6}$$

$$\Delta\hat{\theta}_{ij} = \arctan\left(\frac{\hat{z}_j - \hat{z}_i}{\sqrt{(\hat{x}_j - \hat{x}_i)^2 + (\hat{y}_j - \hat{y}_i)^2}}\right) - \hat{\theta}_i, \tag{6.7}$$

where $\mathbb{1}(x)$ is the step function, equal to 1 if $x > 0$ and 0 otherwise and $\hat{\phi}_i$ and $\hat{\theta}_i$ refers to the estimated yaw and pitch angles of the UAV. We can then compute the beamforming gain due to uncertainty in position information as a function of $\Delta\hat{\phi}_{ij}$ and $\Delta\hat{\theta}_{ij}$:

$$g(\Delta\hat{\phi}_{ij}, \Delta\hat{\theta}_{ij}) = \frac{1}{M_{\mathrm{H}} M_{\mathrm{V}}} \frac{1}{\mathbf{\Delta x}_{ij}} \mathrm{h}_{Rx} \mathrm{h}_{Tx}, \tag{6.8}$$

where,

$$\mathrm{h}_{Tx} = |\mathbf{a}(\Delta\hat{\phi}_{ij}, \Delta\hat{\theta}_{ij})^{\mathrm{H}} \mathbf{w}_{Tx}|, \tag{6.9}$$

$$\mathrm{h}_{Rx} = |\mathbf{a}(\pi + \Delta\hat{\phi}_{ij}, \pi + \Delta\hat{\theta}_{ij})^{\mathrm{H}} \mathbf{w}_{Rx}|, \tag{6.10}$$

where $\mathbf{w}_{Tx}/\mathbf{w}_{Rx} \in \mathbb{C}^{M\times 1}$ denotes the beamforming vector used for transmission/reception and

$$\mathbf{a}(\phi, \theta) = \left[e^{j\kappa^{\mathrm{T}}(\phi,\theta)\mathbf{u}_1}, \ldots, e^{j\kappa^{\mathrm{T}}(\phi,\theta)\mathbf{u}_M}\right]^{\mathrm{T}}, \tag{6.11}$$

is the steering vector matrix associated to the azimuth (ϕ) and elevation (θ) angles. Also, note that, the azimuth for the j^{th} UAV with respect to i^{th} UAV is opposite of the angles of i^{th} UAV with respect to j^{th} UAV, i.e., these angles are supplementary due to the central controller devising the directions the UAVs have to orient themselves for transmission/reception. The same analogy is valid for the elevation angles between these UAVs. For simplicity, we have also defined $\kappa(\alpha, \theta)$ as the wave vector for a planar wave impinging with angles ϕ and θ, λ is the wavelength and $\mathbf{u}_m$ is the 3D spatial location of the mth element of the antenna array. Specifically, for UPA, $\mathbf{u}_m = [0,\ i(m)d_H\lambda,\ j(m)d_V\lambda]^{\mathrm{T}}$ where we also consider the auxiliary functions $i(l) = \mathrm{mod}(l-1, M_{\mathrm{H}})$ and $j(l) = \lfloor (l-1)/M_{\mathrm{V}} \rfloor$.

We can then compute the expected received power $P_{ij}^{(r)}$ over link (i, j) as:

$$P_{ij}^{(r)} = \frac{P_{\mathrm{Tx}} |g(\Delta\hat{\phi}_{ij}, \Delta\hat{\theta}_{ij})|^2}{P_L(d_{ij})}. \tag{6.12}$$

The received power only depends on the transmission power P_{Tx}, the beamforming gain for the transmission and reception, and the distance d_{ij}. Note that, the central controller devises the transmit and receive beamforming vector $\mathbf{w}_{Rx}$ based on the estimated positions of the UAVs and thereby determines the expected received power at the receiver. By using the Shannon rate formula, we can then get the expected capacity $C_{ij}(t)$ under interference

$$C_{ij}(t) = B \log_2 \left(1 + \frac{P_{ij}^{(r)}(t)}{\sum\limits_{\ell \neq i} P_{\ell j}^{(r)}(t) + N_0 B} \right), \tag{6.13}$$

where N_0 is the noise power spectral density and B the bandwidth of the channel. Naturally, the distribution of this capacity is extremely complex, as it is a highly nonlinear function of the estimated positions and attitudes in the swarm. As such, it is extremely hard to estimate directly, but we can use a Monte Carlo sampling method to draw the real state of the swarm from the belief distribution, which can approximate the real distribution when given enough samples.

6.2.1 Analog Beamforming Design

The design of the beamforming vectors $\mathbf{w}_{Tx}$ and $\mathbf{w}_{Rx}$ directly influences the behaviour of the power response $P_{ij}^{(r)}$. For a fixed transmitter to simultaneously communicate with multiple receivers, it is essential to design a beamforming that directs the emitted signal towards these receivers while keeping the interference from other receivers as low as possible. For instance, when the CSI of all receivers is perfectly known at the transmitter, one can design $\mathbf{w}_{Tx}$ and $\mathbf{w}_{Rx}$ as a zero-forcing beamformers [199]. This type of beamforming design is primarily applicable for traditional fixed-transmitter communication systems. Our scenario, however, differs in at least three aspects from this traditional scheme. Firstly, each UAV is only interested in communicating with a single receiver at a particular time based on the route devised by the routing protocol. Secondly, the control station does not have the current CSI for each of the UAVs. In fact, keeping track of all CSIs among the different UAVs in this scenario is a hard task due to high mobility of the UAVs. Finally and the most important characteristic of the scenario is the mobility of the transmitter. In our scenario, UAVs are mobile transmitters which can easily be rotated towards a desired direction. Thus, in our devised scenario, alignment happens by rotating pairs of UAVs towards each other.

Moreover, due to the characteristics mentioned above, designing $\mathbf{w}_{Tx}$ and $\mathbf{w}_{Rx}$ for our communication scheme boils down to choosing a narrow or wide beam.

Notice that, in the former, a narrow beam potentially reduces the amount of interference leaked towards other UAVs, but requires knowledge over the true position $\mathbf{x}_k$. In contrast, designing a wide beam can compensate for the uncertainty over the desired location while potentially increasing interference among different UAVs. We simulate this behaviour by turning on/off the last rows or columns of the UPA of each UAV. To simulate this idea, let us consider the logical matrix $\mathbf{W}$ with entries $\{0, 1\}$ and dimension $M_H \times M_V$. We can design the beamforming vector $\mathbf{w}$ by stacking the columns of matrix $\mathbf{W}$, i.e.,

$$\mathbf{w} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_V \end{bmatrix} \tag{6.14}$$

where $\mathbf{W}_k$ denotes the kth column of the logical matrix.

Then choosing a wide beam translates into setting the last rows/columns of the matrix $\mathbf{W}$ to zero and the remaining entries to one. For instance, to have a omnidirectional transmission, it is sufficient to have a single active antenna [200]. Opposite to that, setting all entries of $\mathbf{W}$ to one results in the narrow beam pattern. Using this idea we derive wider and narrower beams during our simulations. Moreover, we assume every transmitter to use the same beamformer over the entire route.

Finally, we also assume that there is no interference between the routes, that have common UAVs as their intermediate relays, by defining a simplified Time Division Multiple Access (TDMA) mechanism which divides the total time of communication equally among all different routes for the common UAV. This TDMA mechanism is defined by the control station, which knows all the concurrent routes in the network at a particular time.

6.2.2 Position Uncertainty Based Beamformed Routing

Using the beamforming vectors defined in the previous section, we define the Beam Aware Stochastic Multihop Routing for FANETs (BAR) protocol. In this protocol, the edge weight $\hat{e}_{ij} \in \hat{E}(t)$ represents the average expected capacity of the link between UAVs i and j, denoted by $C_{ij}(t)$. Hence, we design this routing problem as a standard maximum capacity route problem [201] over the graph $\hat{G} = (K, \hat{E}(t))$, which can be solved by determining the maximum spanning tree using the link capacities, weighted by the source load, as a weighting metric for the graph edges. So, for a given beamformer, we apply a maximum spanning

Simulation Parameter	Simulation Value
Map Size [m]	$200 \times 200 \times 10$
Density of UAVs [UAVs/km^3]	{25k-75k}
UAV Position Model	Unscented Kalman Filter
Maximum Transmission Distance [m]	100
MIMO Antenna	Uniform Planar Array (UPA)
Antenna Configurations for the UAVs	$\{1, 4, 8, 16, 32, 64\}$
Transmission Power [W]	1
Number of Simulated Networks	240
Bandwidth (MHz)	100

Table 6.1: Parameters of the Simulations

tree on the network graph G to determine the route with maximum achievable capacity.

Hence, using (6.13), we devise the routes that maximize the minimum expected capacity. We assume that each UAV i is occupied in transmitting or receiving cross-traffic for a fraction ρ_i of the time i.e. $\rho_i = [0, 1]$, which must be subtracted to determine the available capacity of the link for UAV i. As transmission is not full-duplex, we also assume that each node can spend at most half the time transmitting. So, the expected capacity of each link can be approximated by Monte Carlo sampling, and we can build a graph, knowing that the capacity of a route $\mathbf{r}$ is given by:

$$C(\mathbf{r}) = \min_{(i,j)\in\mathbf{r}} \frac{(1-\rho_i)C_{i,j}}{2}. \tag{6.15}$$

The entire routing algorithm is devised at the central controller which tracks the positions of all the UAVs [167]. This, however, limits the beamforming adaptability over a single route, i.e., all the UAVs in the route follow the same beamforming pattern. In this work, the central controller devises the beamforming patterns for all the UAVs in the entire route based on the estimated position of the receiver UAV with respect to transmitter UAV, i.e., the central controllers designs the beamsteering vector shown in (6.11). In principle, it might seem sub-optimal to fix the beamforming pattern and only design the beamsteering vector (i.e., deploy analog beamforming), but to devise beamformed routes, the central controller needs to have a perfect instantaneous knowledge of the channel (i.e., deploy digital beamforming), which is difficult in the current scenario.

6.2.3 Simulation Results

In order to numerically evaluate the effects of position uncertainty and beamforming in routing protocols, we deploy a Monte Carlo simulation in MATLAB where

the protocols are evaluated over randomly generated networks. The simulation parameters for the system are described in Table 6.1. The protocols evaluated in the Monte Carlo simulation are:

1. Distance-Based Routing (DBR): A purely distance-based protocol, which does not consider positioning uncertainty;

2. SMURF: our scheme from [196], limited to a single path, which considers positioning uncertainty but does not include beamforming in the probability calculation;

3. BAR: the proposed scheme, which takes into account the position uncertainty and beamforming.

These protocols are evaluated for both the ideal and tracked position. We indicate the former by attaching "-I" to the end of the respective protocol name, and the latter by attaching "-T'. Moreover, the tracked position information is obtained from the output of the Kalman filter at the central controller. Hence, the performance of the protocols evaluated in this scenario is the performance achieved based on the available tracked information of the UAVs. By taking into account the true position of the UAVs, we can determine the ideal performance of the protocol, which represents an upper bound to the practically achievable performance. Additionally, each UAV is equipped with the same UPA antenna configuration and is able to communicate with other drones using the beamforming design defined in the previous section. To accurately determine the performance achieved for all the protocols, we carry out simulations with different antenna configurations and different densities. Figure 6.2 compares the average throughput of all the protocols applied in a network with density of 50000 UAVs/km^3 (which corresponds to about 20 drones in network) and for the antenna configuration of $M_H = M_V = 4$, i.e., $M = 16$ elements in the antenna. Firstly, it is easy to see that the most important factor is the availability of the true position: the real protocols, which operate on uncertain information, have a throughput that is lower by about 10% than the one obtained by their respective ideal versions. However, BAR outperforms both SMURF and DBR for both true and tracked positions. This indicates that the routes chosen by BAR are able to provide median as well as the worst case throughput (i.e., 25^{th} percentile) higher than SMURF and DBR for both tracked and true scenarios. Hence, BAR is able to outperform the other two protocols by about 5% even with a static beamforming scheme. A similar behavior is observed when comparing all the protocols considering different UAV densities. Considering the same antenna configuration as above, Figure 6.3 illustrates this

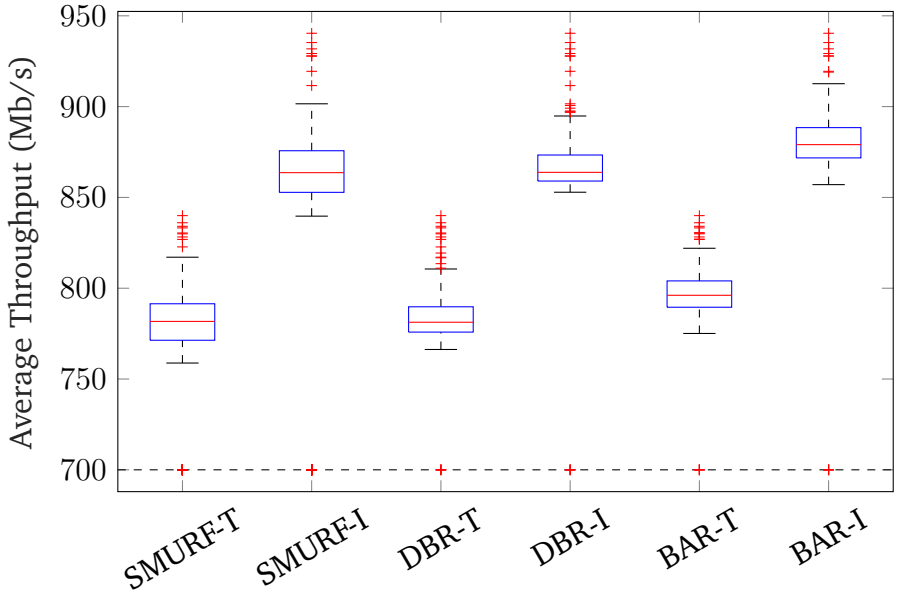


Figure 6.2: Boxplot of the average throughput obtained for all the protocols in different experiments with a network density of 50000 UAVs/km^3 and $M = 16$ antenna elements.

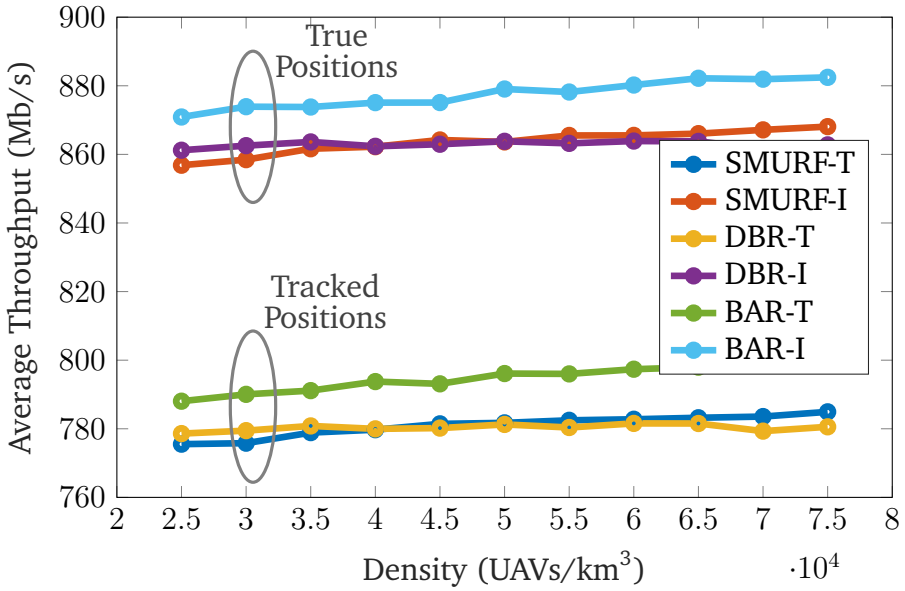


Figure 6.3: Average throughput obtained by the protocols for different UAV densities with $M = 16$ antenna elements.

density comparison. Both SMURF and the enhanced BAR version can exploit high-density network to find better routes, i.e., routes with high link existence probability (for SMURF) and high minimum expected capacity (for BAR). On the other hand, DBR's throughput does not increase with increasing density, as the protocol only takes into account the distance between the tracked positions, without considering the uncertainty: consequently, it will not choose safer routes,

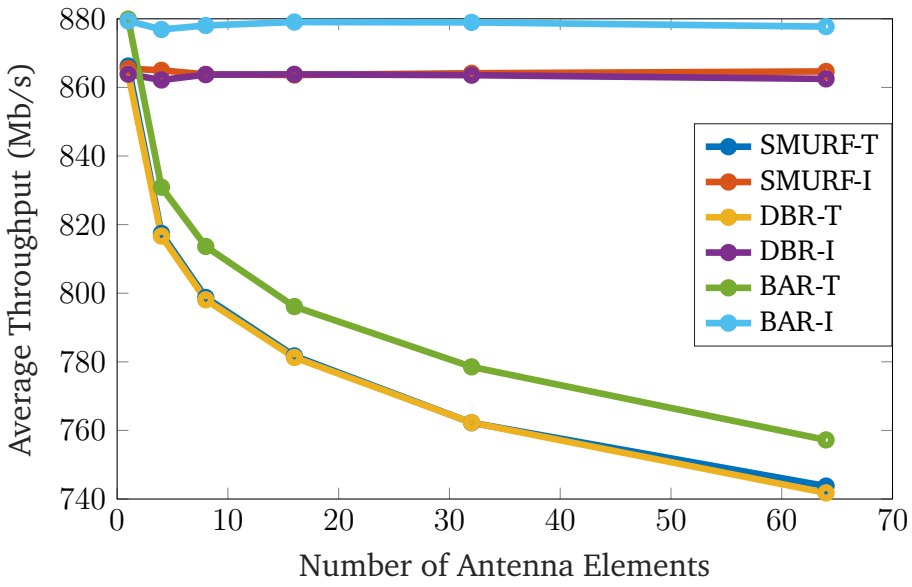


Figure 6.4: Average throughput obtained by the protocols for different UAVs antenna configurations for a network density of 50000 UAVs/km^3.

which are available if the density of the network increases. It is also evident from the figure that BAR outperforms both SMURF and DBR for all densities and for both the tracked and true position information, thanks to its joint consideration of the position uncertainty and beamforming pattern. Additionally, considering the beamforming design can allow the system to potentially reduce interference to other established routes, as well as allowing for more efficient power allocation for the transmission towards the receiver UAV. Another interesting evaluation is the impact of different antenna configurations, i.e., different beamforming patterns, assuming a fixed power allocation for the antenna. Figure 6.4 shows the performance achieved by all the protocols for different antenna configurations in a network with a density of 50000 UAVs/km^3. The performance of the protocols when taking into account true positions of the UAVs in the network is similar for all different antenna configurations. This shows that beamforming does not impact the performance when the true positions are known by the protocols, as the transmitter UAV is able to beamform the signal in the correct direction. However, when considering the tracked position information, beamforming becomes a problem: as the number of elements in the antenna increases, the transmitted beam narrows, leading to a stronger impact of the position uncertainty, and consequently, to a lower throughput. Note that, we consider analog beamforming design i.e. the controller can only devise steering vector for the UAVs antennas. The loss in performance is due to the fact that the central controller does not know the true position of the UAVs and it devises the beamsteering vector $a_l(\phi, \theta)$

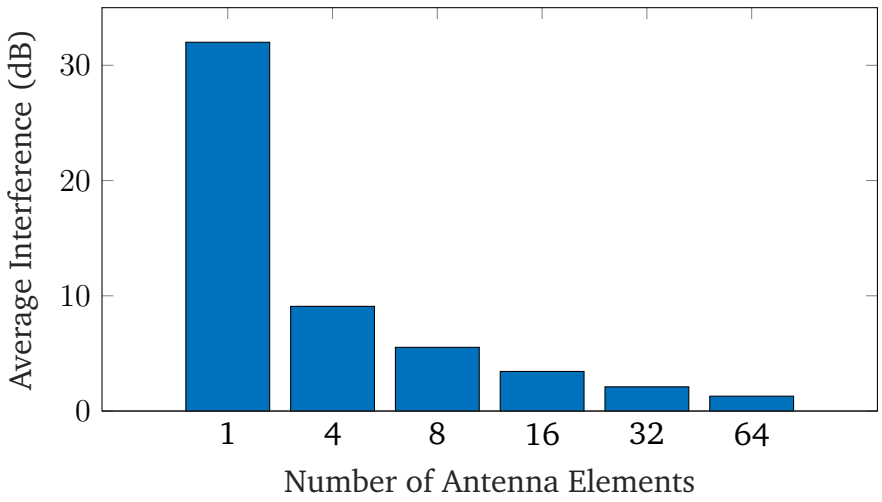


Figure 6.5: Average interference incurred for different UAVs antenna configurations.

based on the expected position of the UAVs: the narrower the beam gets, the larger the impact of pointing errors becomes. When the number of elements is 1, the performance of all the protocols is similar for tracked and true position information, since the antenna is omnidirectional. Additionally, BAR outperforms SMURF and DBR both when considering perfect information for the protocols and in the more realistic setting for all the antenna configurations. This highlights the importance of incorporating beamforming information to determine routes in FANETs. While the average throughput for the considered route is higher when choosing a single antenna element (i.e., an omnidirectional antenna), the downside of this configuration is the high interference to neighboring UAVs, as shown in Figure 6.5: an omnidirectional transmission will have an increased impact on other transmissions. When the network is interference-limited, the use of narrow beamforming design to reduce the neighbourhood interference is beneficial: the lower received power due to pointing errors is compensated by the lower interference. On the other hand, when the network is noise-limited, the use of wide beamforming is beneficial, as it can reduce the impact of UAV positioning uncertainty. Our results also show that the need for adaptive beamforming for each UAV is crucial, as having the same beamforming design for the entire network can reduce the overall performance achieved by the network.

6.3 Adaptive Beamforming for Routing Design

The use of fixed beamforming invariably hinders the performance as shown in Sec. 6.2 when considering imperfect information for routing decisions. For example, if UAVs employ narrow beamforming at a shorter distance, they might have a higher probability to suffer more from beamforming errors: the Signal-to-

Noise ratio (SNR) drop due to small positioning error will be very high. Hence, the effect of the positioning error on the beamforming angle is proportionally larger. In order to compensate, we define the beamformers in this section in an adaptive way, i.e., considering the estimation error in the angular deviation perceived by the transmitter UAV with respect to receiver UAV.

6.3.1 System Model

We again describe the connectivity in a FANET as a time-varying graph $G = (K, E(t))$, where K represents the set of UAVs in the network and $E(t)$ the set of active links at time t. Each drone moves independently from the rest and has its attitude characterized by a 5-tuple: the coordinates in space $\mathbf{x}_k(t) = (x_k(t), y_k(t), z_k(t))$ and the yaw and pitch angles. So, the quality of the link between a pair of drones (i, j) depends on their Euclidean distance [196]:

$$d_{ij}(t) = \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|_2. \tag{6.16}$$

Since UAVs fly high from the ground, the space between them is assumed to be free, i.e., without obstacles or reflections. For simplicity, we consider $z_k(t)$ to be constant for all UAVs, i.e. the height of all the UAVs is constant. We can then consider that there is predominantly LoS communication [197]. Thus, the pathloss between two UAVs is described by

$$P_L(d_{ij}) = \left(\frac{c}{4\pi f_0 d_{ij}}\right)^{\gamma}, \tag{6.17}$$

where f_0 is the carrier frequency, c is the speed of light, and γ is the path loss exponent. As a result, if two UAVs are nearby, then there exists a link between them. Otherwise, communication becomes impractical due to low SNR. More formally, we can describe the set of active links by

$$E(t) = \{e_{ij}(t) : d_{ij} < D\}, \tag{6.18}$$

where the choice of D often depends on the environment and the wireless technology used for communication [198].

As mentioned in the previous section, in more realistic scenarios, the real position of all drones is unknown, but the central controllers maintain estimators to predict the UAVs' positions based on current and previous location updates [167] and are described by

$$\hat{\mathbf{x}}_k(t) = \mathbf{x}_k(t) + \mathbf{n}_k(t), \tag{6.19}$$

where the noise $\mathbf{n}_k(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_k)$ is associated to the kth UAV and $\mathbf{\Sigma}_k \in \mathbb{R}^4$ is the covariance matrix that relates the uncertainties in both x and y coordinates. Note that, $\mathbf{\Sigma}_k$ is not an identity matrix, i.e. the error in estimate is often not equal nor proportional in x and y coordinates. This is due to the fact that the error in estimate will be larger in the direction of movement of the UAV and smaller in other directions. Hence the uncertainty in position of kth UAV can be denoted as,

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{\Sigma}_k), \tag{6.20}$$

and the uncertainty in deviation of kth UAV with respect to lth UAV is denoted by,

$$\hat{\mathbf{x}}_{kl} = \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_l = \begin{bmatrix} \hat{x}_{kl} \\ \hat{y}_{kl} \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}_{kl}, \mathbf{\Sigma}_{kl}), \tag{6.21}$$

where x_{kl} represents entries in x-axis and y_{kl} in the y-axis. For simplicity, we have also defined

$$\boldsymbol{\mu}_{kl} = \boldsymbol{\mu}_k - \boldsymbol{\mu}_l = \begin{bmatrix} \mu_{x_{kl}} \\ \mu_{y_{kl}} \end{bmatrix} \tag{6.22}$$

and

$$\mathbf{\Sigma}_{kl} = \mathbf{\Sigma}_k + \mathbf{\Sigma}_l = \begin{bmatrix} \sigma^2_{x_{kl}} & 2\rho_{kl}\sigma_{x_{kl}}\sigma_{y_{kl}} \\ 2\rho_{kl}\sigma_{x_{kl}}\sigma_{y_{kl}} & \sigma^2_{y_{kl}} \end{bmatrix} \tag{6.23}$$

for a given scale correlation parameter $\rho_{kl} \in [0, 1]$. For instance, if $\rho_{kl} = 0$, the $\hat{x}_{kl}$ and $\hat{y}_{kl}$ are uncorrelated. To determine the beam direction and width, we compute the angular deviation between $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_l$ by considering the random variable $\hat{\mathbf{x}}_{kl}$. We first note that, the random variable $\hat{\mathbf{x}}_{kl}$ can also be represented in the complex space (using Euler's formula) by $\hat{z}_{kl} = \hat{r}_{kl} e^{j\hat{\theta}_{kl}}$, where $\hat{r}_{kl}$ and $\hat{\theta}_{kl}$ are the envelope and phase of the complex random variable $\hat{z}_{kl}$. Then from [202], we obtain that the joint PDF of $\hat{r}_{kl}$ and $\hat{\theta}_{kl}$ is given by

$$\begin{aligned} \mathrm{P}(\hat{r}_{kl}, \hat{\theta}_{kl}) = & \frac{1}{2\pi\sigma_{x_{kl}}\sigma_{y_{kl}}\sqrt{1-\rho^2_{kl}}} \\ & \times \exp\left[-\left(\frac{\mu^2_{x_{kl}}}{2\sigma^2_{x_{kl}}(1-\rho^2_{kl})} + \frac{\mu^2_{y_{kl}}}{2\sigma^2_{y_{kl}}(1-\rho^2_{kl})} - \frac{\rho_{kl}\mu_{x_{kl}}\mu_{y_{kl}}}{\sigma_{x_{kl}}\sigma_{y_{kl}}(1-\rho^2_{kl})}\right)\right] \\ & \times \hat{r}_{kl} \ \exp\left[-\hat{r}^2_{kl}\mathcal{A}(\hat{\theta}_{kl}) + \hat{r}_{kl}\mathcal{B}(\hat{\theta}_{kl})\right], \end{aligned} \tag{6.24}$$

where,

$$\mathcal{A}(\hat{\theta}_{kl}) = \left(\frac{\cos^2 \hat{\theta}_{kl}}{2\sigma_{x_{kl}}^2(1-\rho_{kl}^2)} + \frac{\sin^2 \hat{\theta}_{y_{kl}}}{2\sigma_{y_{kl}}^2(1-\rho_{kl}^2)} - \frac{\rho_{kl} \sin \hat{\theta}_{kl} \cos \hat{\theta}_{kl}}{\sigma_{x_{kl}}\sigma_{y_{kl}}(1-\rho_{kl}^2)} \right), \tag{6.25}$$

$$\mathcal{B}(\hat{\theta}_{kl}) = \left\{ \frac{1}{\sigma_{x_{kl}}(1-\rho_{kl}^2)} \left[\frac{\mu_{x_{kl}}}{\sigma_{x_{kl}}} - \frac{\rho_{kl}\mu_{y_{kl}}}{\sigma_{y_{kl}}} \right] \cos \hat{\theta}_{kl} + \frac{1}{\sigma_{y_{kl}}(1-\rho_{kl}^2)} \left[\frac{\mu_{y_{kl}}}{\sigma_{y_{kl}}} - \frac{\rho_{kl}\mu_{x_{kl}}}{\sigma_{x_{kl}}} \right] \sin \hat{\theta}_{kl} \right\}. \tag{6.26}$$

Additionally, the PDF of angular deviation $\hat{\theta}_{kl}$ can be given as,

$$\begin{aligned} \mathrm{P}(\hat{\theta}_{kl}) = & \frac{1}{2\pi\sigma_{x_{kl}}\sigma_{y_{kl}}\sqrt{1-\rho_{kl}^2}} \\ & \times \exp\left[-\left(\frac{\mu_{x_{kl}}^2}{2\sigma_{x_{kl}}^2(1-\rho_{kl}^2)} + \frac{\mu_{y_{kl}}^2}{2\sigma_{y_{kl}}^2(1-\rho_{kl}^2)} - \frac{\rho_{kl}\mu_{x_{kl}}\mu_{y_{kl}}}{\sigma_{x_{kl}}\sigma_{y_{kl}}(1-\rho_{kl}^2)} \right) \right] \\ & \times \frac{1}{2\mathcal{A}(\hat{\theta})} \left\{ 1 + \frac{\sqrt{\pi}\mathcal{B}(\hat{\theta})}{2\sqrt{\mathcal{A}(\hat{\theta})}} \exp\left[\frac{\mathcal{B}^2(\hat{\theta})}{4\mathcal{A}(\hat{\theta})} \right] \mathrm{erfc}\left(-\frac{\mathcal{B}(\hat{\theta})}{2\sqrt{\mathcal{A}(\hat{\theta})}} \right) \right\}, \end{aligned} \tag{6.27}$$

where $\mathrm{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty \exp(-x^2)\, dx$.

6.3.2 Adaptive Beamforming Design: Accounting for Uncertainty

To define an adaptive beamforming design, we consider the angle estimates $\hat{\theta}_{kl}$ from the kth source to the lth receiver. Moreover, we assume all UAVs to be equipped with two Uniform Linear Array (ULA) with M elements elements each. These antennas are placed parallel to each other at opposite sides of the UAV and, depending on the angular deviation between the transmitter and receiver UAVs, the controller instructs the UAV to activate one antenna or another. The goal of having two parallel antennas is to avoid the rotation of the transmitter/receiver UAV towards another receiver/transmitter as assumed in previous section. Moreover, at no time, one UAV has the pair of antennas simultaneously activated for a single task (i.e., transmit or receive). Hence, each antenna is independent of each other and the the relationship between the M elements of each of these antennas is fully described by estimated angular deviation $\hat{\theta}_{kl}$. Specifically, at each antenna this relationship is determined by the covariance matrix

$$\mathbf{R}_{kl} = \int_{-\pi}^{\pi} \mathbf{a}(\hat{\theta}_{kl})\mathbf{a}(\hat{\theta}_{kl})^{\mathrm{H}} \mathrm{P}(\hat{\theta}_{kl}) d\hat{\theta}_{kl}, \tag{6.28}$$

where $\mathbf{a}(\hat{\theta}_{kl})$ is the steering vector in the estimated angle $\hat{\theta}_{kl}$ and is given by

$$\mathbf{a}(\hat{\theta}_{kl}) = \left[e^{-j\pi \sin(\hat{\theta}_{kl})}, e^{-j2\pi \sin(\hat{\theta}_{kl})} \ldots, e^{-j(M-1)\pi \sin(\hat{\theta}_{kl})}\right]^{\mathrm{T}}. \tag{6.29}$$

To account for the uncertainty over $\hat{\theta}_{kl}$, we are interested in maximizing the power throughout a desired angular range δ_{kl} that ideally encloses this uncertainty. We do so by reducing the energy transmitted outside this desired angular space, i.e., by designing the beamformer

$$\hat{\mathbf{w}}_{kl}(\delta_{kl}) = \mathbf{w}(\hat{\theta}_{kl}, \delta_{kl}) = \frac{\tilde{\mathbf{R}}_{kl}^{-1}\mathbf{a}(\hat{\theta}_{kl})}{\mathbf{a}(\hat{\theta}_{kl})^{\mathrm{H}}\tilde{\mathbf{R}}_{kl}^{-1}\mathbf{a}(\hat{\theta}_{kl})}, \tag{6.30}$$

where $\tilde{\mathbf{R}}_{kl}$ is the complementary covariance matrix which is given by,

$$\tilde{\mathbf{R}}_{kl} = \int_{-\pi}^{\hat{\theta}_{kl}-\delta_{kl}} \mathbf{a}(\theta_{kl})\mathbf{a}(\theta_{kl})^{\mathrm{H}}\mathrm{P}(\theta_{kl})d\theta_{kl} + \int_{\hat{\theta}_{kl}+\delta_{kl}}^{\pi} \mathbf{a}(\theta_{kl})\mathbf{a}(\theta_{kl})^{\mathrm{H}}\mathrm{P}(\theta_{kl})d\theta_{kl}. \tag{6.31}$$

Finally, we obtain the gain at the angular deviation θ_{kl} using the designed beamformer $\hat{\mathbf{w}}_{kl}(\delta)$ given by

$$g_{kl}(\delta) = g(\theta_{kl}, \hat{\theta}_{kl}, \delta) = |\mathbf{a}(\theta_{kl})^{\mathrm{H}}\hat{\mathbf{w}}_{kl}(\delta)|^2 \tag{6.32}$$

where δ is a pre-defined beamwidth. For simplicity we also denote $\hat{g}_{kl}(\delta) = g(\hat{\theta}_{kl}, \hat{\theta}_{kl}, \delta)$ as the gain obtained in the noise direction.

6.3.3 Routing Design

In general, depending on the application, FANETs might need to fulfill different requirements. For instance, applications that generate data bursts such as emergency information often require the highest communication rate possible for quick data transfer. Opposite to that, video monitoring systems often requires a minimal and constant rate over the entire route. To account for these different needs in FANETs we devise two routing protocols with different objectives, namely,

- **Expected Capacity Maximisation:** the goal is to select sequential links (between UAVs) so that they maximise the minimum expected capacity over the entire route, i.e. it finds routes which maximises the bottleneck capacity. This approach is potentially useful for our first example scenario, where the communication over the route necessitates higher data rates over a short period of time.

- **Minimal Capacity Guarantee:** the routing protocol maximises the probability of achieving a minimum capacity. This approach is potentially useful when the communication over the route is constant, such as video stream, over a period of time thereby needing a minimum capacity for data transfer.

In the following we describe two routing protocols that satisfy each of these objectives while taking into account the position uncertainty of the UAVs in a fleet.

6.3.3.1 Expected Capacity Maximisation (Joint Beamforming and Routing in FANETs (JBR)-E)

Let us start by noticing that the expected capacity in the link e_{kl} between the kth and lth UAV pair can be computed by the double integral

$$\hat{C}_{kl} = \int_0^{\infty} \int_{-\pi}^{\pi} \mathrm{P}(r_{kl}, \theta_{kl}) C(r_{kl}, \theta_{kl}) d\theta_{kl} dr_{kl}, \tag{6.33}$$

where we have used the $(k, l)-$link capacity definition

$$C(r_{kl}, \theta_{kl}) = B \log_2 \left(1 + \frac{p_{kl}^{\mathrm{tx}} \, g_{kl}(\delta)}{P_L(r_{kl}) \sigma^2} \right), \tag{6.34}$$

for a fixed transmission power (p_{kl}^{tx}) and noise power (σ^2). Likewise, the expected capacity throughout a route $\mathcal{E}_{sd}$ is determined by its bottleneck link and is obtained by

$$\bar{C}_{sd} = \min_{e_{kl} \in \mathcal{E}_{sd}} \hat{C}_{kl}. \tag{6.35}$$

In the context of FANETs, the minimal capacity in a route translates in the link with the highest positional uncertainties for transmitter and receiver UAVs and largest distance between them. Moreover, due to the nature FANETs, there might exist several paths/routes that lead from a source s to a destination d. As described above, we are interested in the route $\mathcal{E}_{sd}^{\mathrm{E}}$ that maximise the minimal expected capacity over the entire route. To do so, we take advantage of modeling the connectivity of the network as graph with each node representing an UAV and edges $\hat{C}_{kl}, \forall k, l$. Then, it becomes intuitive to find the path with maximum expected capacity $\mathcal{E}_{sd}^{\mathrm{E}}$ using the maximum spanning tree [203] over the defined graph, which is given by,

$$\mathcal{E}_{sd}^{\mathrm{E}} = \underset{\mathcal{E}_{sd} \in \Gamma_{sd}}{\arg\max} \, \bar{C}_{sd}, \tag{6.36}$$

where Γ_{sd} represents all the possible routes from source s to destination d.

6.3.3.2 Minimal Capacity Guarantee (JBR-P)

In the second routing design, we take into account the probability that a link has expected capacity higher than a certain minimum value $C_{\min}$. This probability can be computed by

$$\mathrm{P}(\hat{C}_{kl} \geq C_{\min}) = \int_0^{\infty} \int_{-\pi}^{\pi} \mathrm{P}(r_{kl}, \theta_{kl})\, \mathcal{I}\left(C(r_{kl}, \theta_{kl}) \geq C_{\min}\right)\, d\theta_{kl}\, dr_{kl} \tag{6.37}$$

where, $\mathcal{I}$ is an indicator function. Our objective is to find the route $\mathcal{E}_{sd}^{\mathrm{P}}$ that maximize this probability for every link in the route, i.e., we are interested in the route, that maximizes the joint probability,

$$\mathcal{E}_{sd}^{\mathrm{P}} = \underset{\mathcal{E}_{sd} \in \Gamma_{sd}}{\arg\max} \prod_{e_{kl} \in \mathcal{E}_{sd}} \mathrm{P}(\hat{C}_{kl} \geq C_{\min}), \tag{6.38}$$

where Γ_{sd} represents all the possible routes from source s to destination d. Depending on the length of the route, this product might cause numerical instability. Hence, we rely on the monotonous characteristic of log functions and re-write the above maximization problem as a minimization problem, given by

$$\mathcal{E}_{sd}^{P} = \underset{\mathcal{E}_{sd} \in \Gamma_{sd}}{\arg\min} \sum_{e_{kl} \in \mathcal{E}_{sd}} -\log_{10} \mathrm{P}(\hat{C}_{kl} \geq C_{\min}). \tag{6.39}$$

Finally, we note that, ideally, $\mathrm{P}(\tilde{C}_{23} \geq C_{\min})$ should be dependent on $\mathrm{P}(\tilde{C}_{12} \geq C_{\min})$. Unfortunately, in this case, the computational complexity increases exponentially as the number of hops increase rendering the problem complex to solve. So, in this scenario, we relax the problem an consider $\mathrm{P}(\tilde{C}_{kl} \geq C_{\min})$ to be independent for each (k, l) UAV link.

6.3.3.3 Quantization for Optimal Beamwidth

Up until now, we have formalised the our two proposed routing protocols (JBR-E and JBR-P) as continuous optimizations based on maximising expected capacity (6.33) and maximising probability of the guaranteed capacity (6.37). Unfortunately, solving such optimizations turn out to be computationally complex due to the continuity over the beamwidth parameter δ. Hence, instead of trying to consider δ as an optimisation parameter, we quantize it and define a set of L equally spaced beamwidths $\Delta_L = \{\delta_1, \ldots, \delta_L | \delta_k = \frac{k\pi}{2L}\}$. To further ease the computational costs, we consider a similar quantization for the angle-of-departure $\Omega_K = \{\theta_1, \ldots, \theta_K | \theta_k = \frac{k\pi}{2K}\}$. Now, in this new quantization scenario, depending on the protocol, the controller has to chose an optimal beamwidth $\delta_{kl}^* \in \Delta_L$ and a direction $\theta_{kl}^* \in \Omega_K$ for each $(k, l)-$link. One of the advantages of considering this quantized scenario is that it allows us to perform an exhaustive search whose

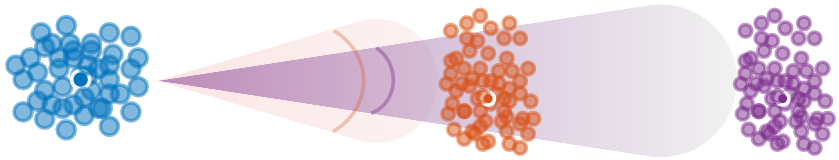


Figure 6.6: Choice of Beamwidth for identical UAV position uncertainties for variable distance

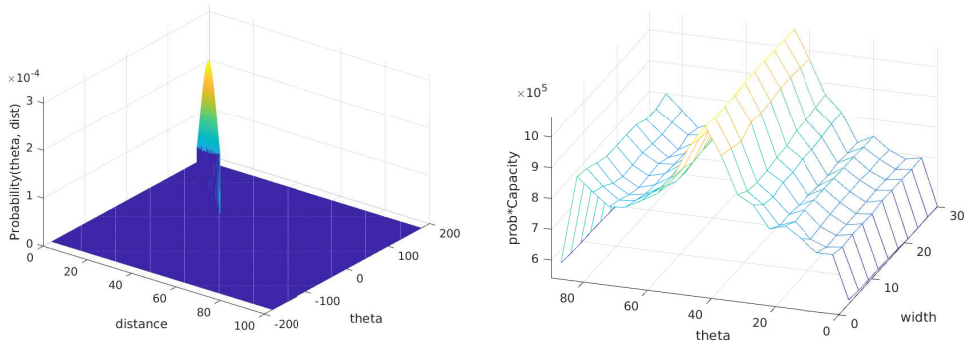


Figure 6.7: (a): Probability of UAV position. (b) Link expected capacity as in (6.33).

size can be controlled based on the choice of L and K. Figure 6.6 shows the quantized beamwidth change with respect to different distances for a constant position uncertainty and Figure 6.7 provides a visualization of $P(r_{kl}, \theta_{kl})$ and its corresponding $\hat{C}_{kl}$ for a specific beamforming codebook. Here, a beamforming codebook is built of several possible pointing directions (x-axis) and beam widths options (y-axis). In the z-axis we have the expected capacity as defined in (6.33). In this case, our search space contains $K = 36$ possible pointing directions equally spaced 10° apart and $L = 10$ possible beam widths.

6.3.4 Results

In order to numerically evaluate the effects of position uncertainty and beamforming in routing protocols, we deploy a Monte Carlo simulation in MATLAB where the protocols are evaluated over randomly generated networks. The simulation parameters for the system are described in Table 6.2. To compare the efficacy of our proposed two approaches, JBR-E and JBR-P, we take into account the three other protocols: Distance-Based Routing (DBR), Stochastic Multipath Routing for FANETs (SMURF), Beam Aware Stochastic Multihop Routing for FANETs (BAR) that are defined in the previous sections. To ensure fairness in computing the capacity for the DBR and SMURF as they inherently do not include the beamforming design, we compute the beamformers for DBR and SMURF which maximise the capacity for the individual links in the route. With respect to capacity computation in BAR, we compute the capacities by considering a fixed

Simulation Parameter	Simulation Value
Map Size [m]	100×100
Density of UAVs [UAVs/km^2]	$\{500 - 2500\}$
UAV Position Model	Unscented Kalman Filter
Maximum Transmission Distance [m]	100
Operating Frequency [GHz]	26
Path loss exponent	2
MIMO Antenna	Uniform Linear Array (ULA)
Antenna Configurations for the UAVs	$\{2, 4, 8, 16, 32\}$
Transmission Power [W]	1
Number of Simulated Networks	100
Bandwidth (MHz)	100
Minimum Capacity ($C_{\min}$) (Gbps)	3

Table 6.2: Parameters of the Simulations

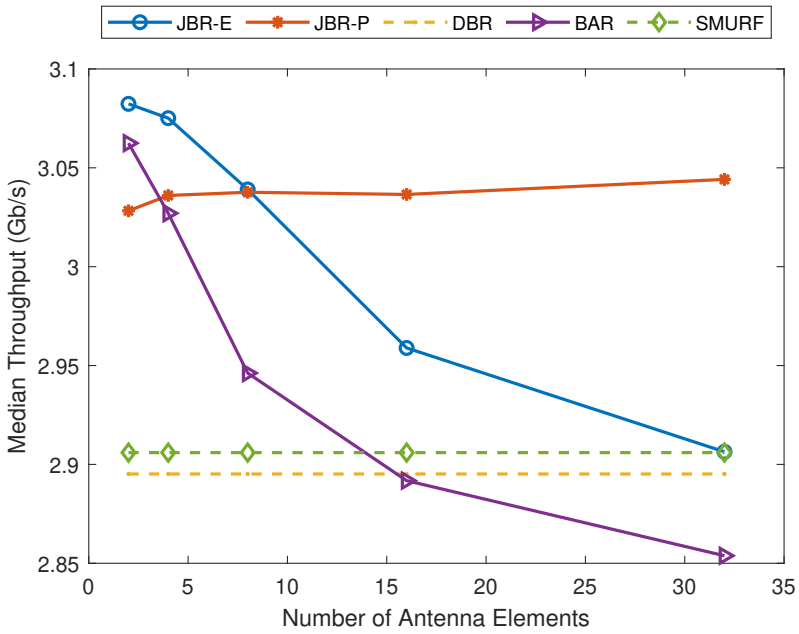


Figure 6.8: Median Throughput Performance for different number of antenna elements at a density of 1500 UAVs/km^2

beamforming design as defined in the previous section. To evaluate the protocols, we determine the performance obtained, i.e., throughput achieved, for different densities and different number of antenna elements.

Figure 6.8 shows the performance obtained for all the protocols for UAV density of 1500 UAVs/km^2 (note that, 1500 UAVs/km^2 for the map size considered in the simulation is around 15 UAVs in the area). As visible from the figure, the performance obtained for JBR-E and BAR are dependent on the number of antenna elements in the UAVs. This dependency is due to fact that both the

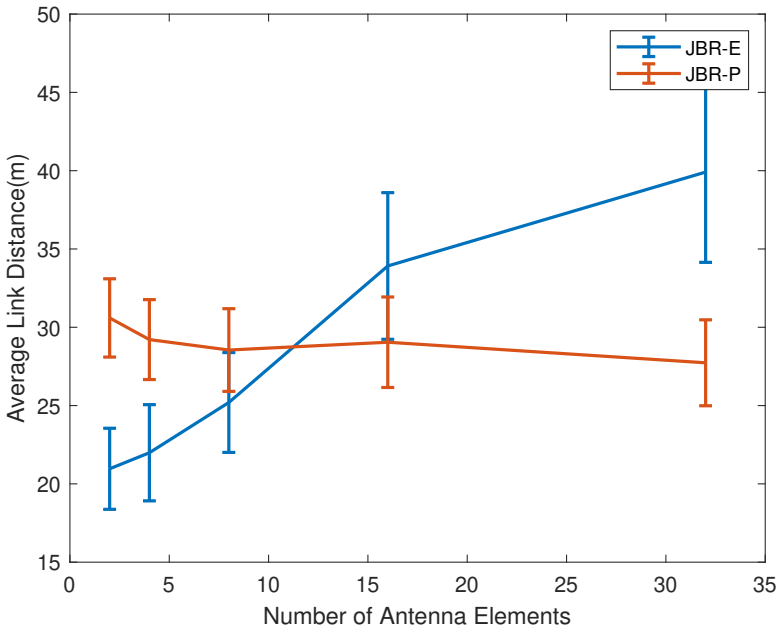


Figure 6.9: Average Link Distance for JBR-E and JBR-P for different number of antenna elements at a density of 1500 UAVs/km^2

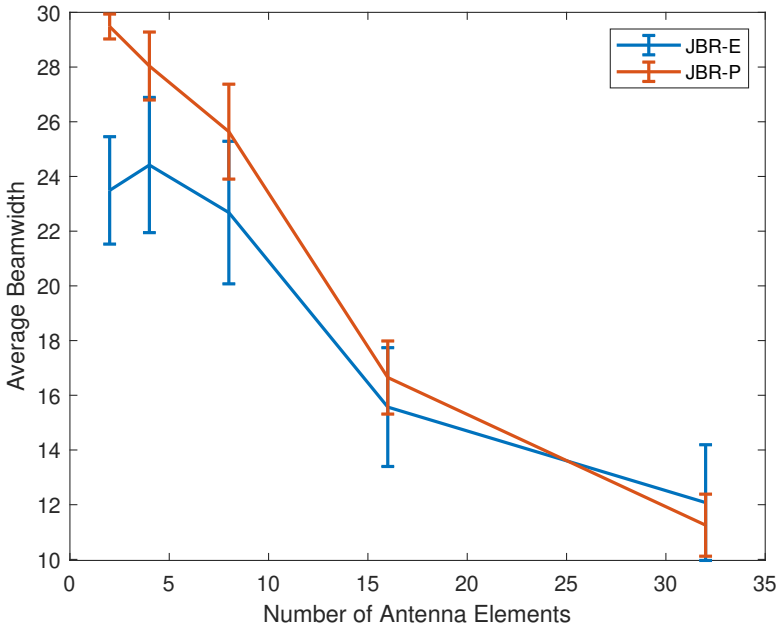
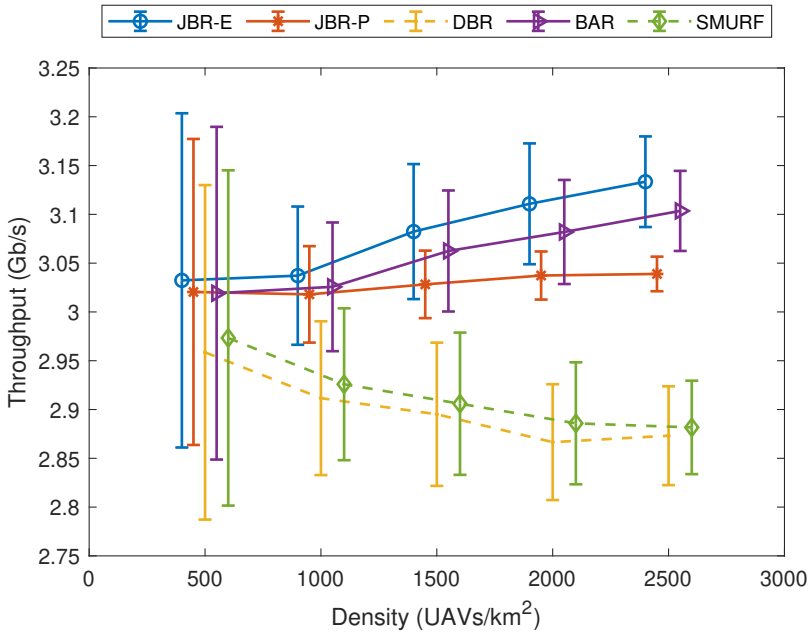


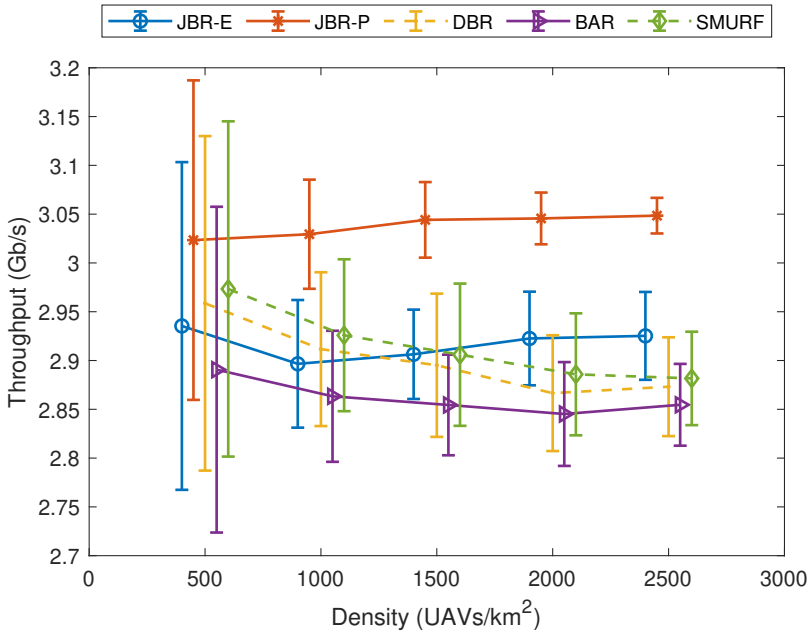
Figure 6.10: Average Beamwidth for JBR-E and JBR-P for different number of antenna elements at a density of 1500 UAVs/km^2

protocols try to maximize the minimum capacity achieved over the entire route. To do so, the protocols choose beamwidth as narrow as possible while taking into account next hop distance and uncertainty in the position. This causes the protocols to choose narrow beamwidth when available to be design i.e. for larger

number of antenna elements in the UAVs. Specifically, JBR-E tries to maximise the minimum expected capacity and thereby chooses narrow beamwidth with larger next hop distances. But this also increase the chance that the beam might not be correctly oriented due to the uncertainty in the position of the receiver UAV thereby reducing the capacity achieved at the receiver UAV. This chance is significantly reduced in JBR-P as the protocol only tries to satisfy a particular minimum capacity $C_{\min}$ instead maximizing the minimum capacity achieved. This allows the protocol to choose wider beamwidths and shorter next hops even when antenna allows for narrow beams as long as the expected capacity is higher than $C_{\min}$. Hence, the performance achieved by JBR-P is almost constant for different number of antenna elements. To clearly visualize the difference between JBR-E and JBR-P, we plot the average link distance and the average beamwidth v/s number of antenna elements used by JBR-E and JBR-P in Figures 6.9 and 6.10. As visible from the figures, JBR-P chooses links with similar average distances and varies the beamwidths to achieve the minimum capacity while JBR-E chooses shorter links when only wider beamwidths can be designed but chooses longer links when narrow beamwidths are available. This hurts the performance of JBR-E as longer links with narrow beamforming are more susceptible to sharp drop in performance due to uncertainties in position. Additionally, as visible from the Figure 6.8, for DBR and SMURF which are distance based protocols dependent on density of the UAVs, the performance remains constant as the density is constant thereby showing that beamforming designs do not show any improvement when used in conjunction with distance based protocols. This is interesting when considering different densities. Figures 6.11a, and 6.11b show the median performance obtained for all the protocols with different densities for 2 and 32 antenna elements. The figures also show the median absolute deviation in the performance obtained, thereby, denoting the variability in the performance obtained for all protocols. For 2 antenna elements, JBR-E and BAR outperforms JBR-P as JBR-E maximises the minimum capacity while considering long range, narrow beamforming designs. Also, JBR-E outperforms all other protocols due to the fact that JBR-E chooses short links when wide beamforming design is only available. Additionally, the performance improves when density is increased for both protocols. This is due to the fact that higher densities provides more relay UAVs thereby increasing the chance of achieving the expected capacity. But, even if the performance obtained for JBR-P is lower than JBR-E and BAR, the variance in the performance for JBR-P is smaller than JBR-E and BAR. Also, increasing the density reduces the variance in performance for all the protocols. This is actually interesting when considering the performance achieved for DBR and SMURF. For



(a) 2 antenna elements



(b) 32 antenna elements

Figure 6.11: Median Throughput Performance for different densities of UAVs

DBR and SMURF, the variance decreases as expected but the median performance also decreases. This is due to the fact that the median performance is dependent on the beamforming design which has to be designed with respect to uncertainty to reduce the chance of beam missing the receiver UAV. This is also be seen in the performance obtained for 32 antenna elements in Figure 6.11b, with the difference being, the median performance achieved for JBR-E and BAR reduces

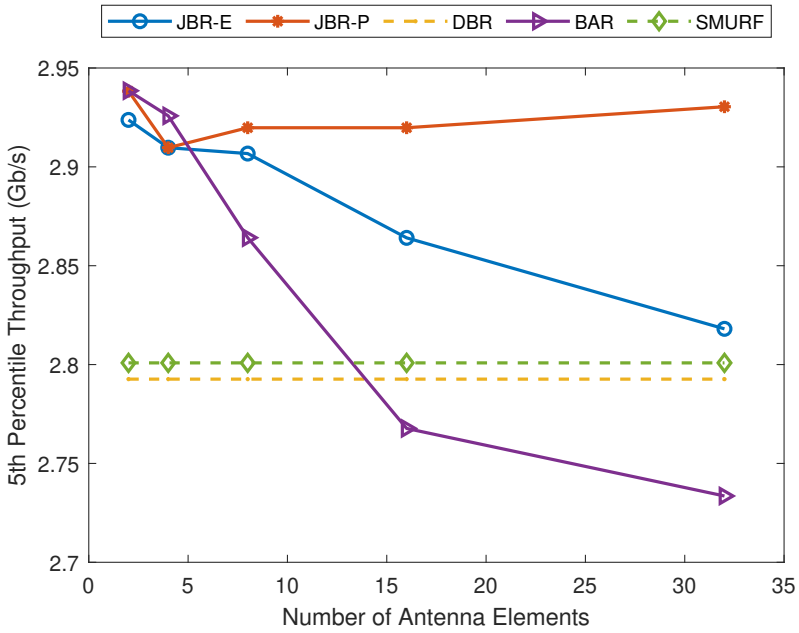
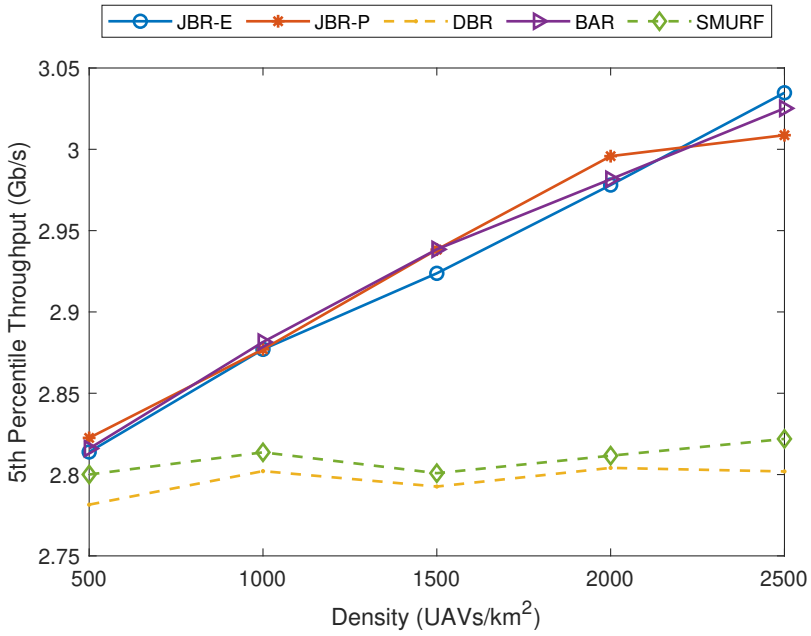


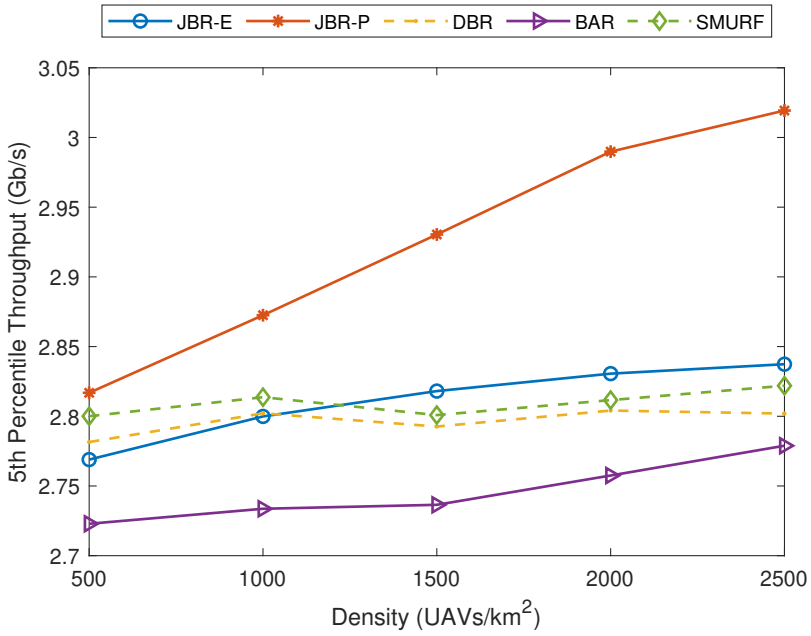
Figure 6.12: 5th Percentile Throughput Performance for different number of antenna elements at a density of 1500 UAVs/km^2

as number of antenna elements is increased for all densities concurrent with performance achieved in Figure 6.8. There is another interesting observation for performance obtained for JBR-E in contrast with BAR for 32 antenna elements. The performance remains almost constant for JBR-E while it decreases for BAR. This is due to the fact that JBR-E devises beamforming for each individual link while BAR has a fixed beamforming design over the entire route. Also, note that, the median performance of JBR-P remains almost constant for different densities and different number of antenna elements. This shows that JBR-P is more resilient to different network configurations and antenna design compared to median performance obtained for JBR-E which is dependent on the density of the UAVs and number of antenna elements.

Also, it is necessary to evaluate the protocols in worst case networking scenarios so as to determine the minimum performance obtained in any networking scenario. So, to evaluate the protocols in the worst case scenarios, we take into account the 5th percentile performance obtained by the protocols. Figure 6.12 show the 5th percentile performance for all the protocols for UAV density of 1500 UAVs/km^2. In coherence with the median throughput performance, the performance achieved by JBR-E and BAR is almost similar. But the most significant factor is the performance obtained by JBR-P in worst case scenarios is much higher than both JBR-E and BAR. This shows that JBR-P is applicable in



(a) 2 antenna elements



(b) 32 antenna elements

Figure 6.13: 5th Percentile Throughput Performance for different densities of UAVs

worst case scenarios i.e. scenarios where the uncertainty is higher thereby having higher chances of failing to achieve the maximum expected capacity. This also shows that JBR-P is much more robust with respect to scenario definition and can still provide an acceptable performance in worst case scenarios. This can be attested even further by incorporating variations in UAV density.

Figures 6.13a and 6.13b show the 5th percentile performance for all the protocols with varying UAV densities and 2 and 32 antenna elements respectively. In all the figures, one common trend that is visible is the increase in 5th percentile performance for increasing UAV densities. This is due to the fact that in worst case scenarios, all the protocols are aided by higher densities and thereby more routing options. But the critical part is, for UAVs with 2 antenna elements, the performance for JBR-E, JBR-P and BAR is almost equal. This signifies that in worst case scenarios, lack of narrow beamforming design due to small number of antennas causes the performance to be almost similar. The difference between JBR-P and JBR-E becomes very significant when the antenna elements are increased to 32 as visible from Figure 6.13b. In this case, JBR-P outperforms all the protocols including JBR-E. This shows that JBR-P is applicable to boost the worst case performance of the system at any density for higher number of elements.
So, finally, both JBR-E and JBR-P can be used for different application scenarios. For applications that need quick data transfer for data bursts, i.e., where the maximum possible capacity needs to be achieved using UAVs with small antenna design in sparse density FANET, JBR-E can be useful, while, for applications that have constant bit rate data stream, i.e., where capacity should be consistent over the entire route irrespective of UAVs' antenna design and the FANET density, then JBR-P is much more applicable. JBR-P is also applicable in scenarios where acceptable worst case performance is necessary.

6.4 Conclusion

In this chapter, we briefly describe the works related to joint optimization of routing and beamforming based on position predictions in FANETs. We provide a statistical analysis of a FANET with tracked position information and beamforming design, and derive the minimum expected capacity for both single links and entire routes. We then present the BAR and JBR protocols that computes the route with to either maximize the expected capacity or maximise the probability to achieve a minimum capacity. Through simulations, we show that the protocols outperform the baselines for different application requirements in realistic network conditions.
Also, an interesting avenue of further research is to define beamforming optimization scenarios for multiple different data-rate and delay requirements while taking into account the tracking information available for each UAV in the networks, thereby, jointly optimizing communication for multiple different data streams and the routing process based on the UAVs' tracked positions and the estimation uncertainty.

7 QUIC Scheduling and Transmission Scheme to Maximize VoI with Correlated Data Flows

7.1 Introduction

Next generation networks are expected to support new, challenging interactive applications that, besides the freedom of movement given by wireless connectivity, generally require the timely and synchronized delivery of a multitude of sensor data and commands to guarantee interactivity and control effectiveness [204].

For example, haptic communication allows users to interact with remote environments using haptic sensors and actuators that exchange kinesthetic and tactile information. In the case of closed-loop bilateral teleoperation systems, kinesthetic data is time-sensitive. Although stability control mechanisms can be employed to compensate for end-to-end delays that can perturb the stability of such systems, this approach may deteriorate the *transparency* of the service, i.e., the feeling of interactivity and, hence, the quality of telepresence [205]. A more transparent way to decrease the end-to-end delay, instead, consists in reducing the sensor data to be transmitted according to human perception models, but at the cost of a less accurate reconstruction of the signal at the receiver [206]. Somehow similarly, connected vehicles can exchange data generated by on-board sensors via Vehicle-to-Everything (V2X) communications, in order to collaboratively build a richer context awareness and coordinate driving decisions. However, disseminating the sensors' observations is expected to increase data traffic in vehicular networks by multiple orders of magnitude, thus potentially leading to congestion, so that proper data transmission schemes are needed [207]. Other applications that generate streams of correlated data are, e.g., remote control of swarms of mobile robots, tele-monitoring of industrial processes, immersive interactive virtual reality.

In order to operate effectively, these applications need flow control strategies to avoid delays and packet losses caused by congestion, as well as recovery mechanisms to protect particularly valuable data. The most common transmission protocols, namely, the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), offer complementary services that, however, are not adequate

for the purpose. By using TCP, applications can delegate congestion control and packet retransmissions to the transport layer, which provides a simple and well-tested interface with standardized behavior. However, most congestion control mechanisms can create significant latency issues. tThe TCP in-order delivery constraint can cause the head-of-line blocking problem when all the data streams are multiplexed into the same TCP connection. Indeed, if a packet from any of the sensors is lost, successive packets from all other sensors are buffered at the receiver and released to the application only after that the previously lost packet is successfully recovered. Conversely, UDP offers full flexibility to the applications, but leaves the burden of managing congestion and retransmissions to them.
In order to overcome the issues of these protocols, in this chapter, we propose QUIC-Enabled Scheduling and Transmission (QUIC-EST). QUIC is a recently developed transport protocol that allows data to be sent in parallel and logically independent *streams*, thus avoiding the head-of-line blocking problem among different streams. This reduces unnecessary delays in the reception of the data, particularly when the number of independent data flows is large. QUIC-EST combines the features of QUIC with a multi-stream scheduling scheme that biases data transmissions as a function of the Value of Information (VoI) provided by the application. Here, the VoI is considered as a scalar value quantifying the utility of the data for the receiver [208]. The VoI takes into consideration the potential correlation of the information flows in time and across different sensors, as well as their intrinsic value. To better illustrate the proposed methodology, we apply it to two relevant use cases, namely autonomous driving and haptic communication, and we show that our approach guarantees better utility compared to traditional transport schemes.

7.2 Adapting QUIC for Time-Sensitive Multi-Sensory Applications

The QUIC protocol [209] was designed by Google to solve some of the latency issues that TCP typically causes with Internet traffic. Indeed, TCP offers a single in-order byte-streaming service, leaving the task of separating application-level objects to the application itself. To guarantee in-order delivery, TCP blocks the delivery to the upper layer of any data that has been received out of order, even if logically independent of the lost/delayed packet. QUIC addresses this head-of-line blocking problem by adopting a solution previously implemented by the older Stream Control Transmission Protocol (SCTP) [210], i.e., defining separate streams of data within the same connection. Each stream is treated by the protocol as a logically separate data flow with in-order reliable delivery, independent of the other streams. Figure 7.1 shows an example of how QUIC

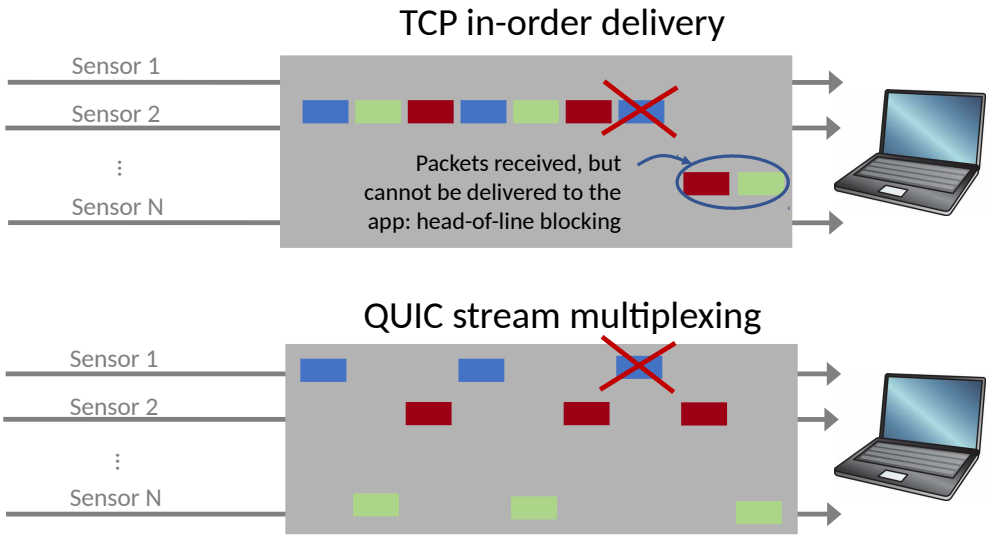


Figure 7.1: The head-of-line blocking problem and the stream-based solution.

handles multiple streams: while the loss of the blue packet also blocks the orange and green packets in TCP, the logical separation between the streams allows QUIC to deliver the data.

QUIC was designed for Web traffic consisting of a potentially large number of logically independent objects to be delivered with the lowest possible latency. However, its features are also suitable to support interactive multi-sensory applications that need to transmit data from multiple sensors, potentially with low delay, to preserve the user's QoE. Nonetheless, unlike Web traffic, sensing data traffic is typically redundant, so that applications do not usually require to receive all the data. This makes the head-of-line blocking issue even more pressing, since the undelivered data might not even be necessary for successful operation. We hence propose the QUIC-EST scheme as a way of adapting QUIC to the multi-sensory application requirements.

In QUIC-EST, each sensory reading can be considered as a separate object. As sensors produce several readings per second, we propose to use not just a different stream for each sensor, but a *different stream for each object*. Whenever all packets sent into a stream are acknowledged, the stream can be reused for a new object. On the contrary, if a stream gets blocked by a packet loss and the data become stale, the sender will transmit a `RESET_STREAM` control frame (which is not bound by in-order delivery constraints) to tell the receiver to discard any existing out-of-order data received from that stream and start again, as suggested in [211].

7.3 Value of Information-Based Scheduling

While the use of streams allows QUIC to transmit data from different sensors independently, the capacity of the connection might not be sufficient to deliver the data from all sensors within the required time. In this case, the choice of which sensor data should be transmitted and in which order becomes a central problem. Since the QUIC protocol does not specify any scheduler, we propose to implement a priority-based mechanism.

We then define a scheduling algorithm that aims at maximizing the *effective VoI* at the receiver, while avoiding congestion in the connection. To this end, the algorithm needs to be fed with four types of information, namely: *(i)* the (estimated) available capacity of the connection, *(ii)* the size of the data, *(iii)* the intrinsic VoI of the data, and *(iv)* the correlation between the data generated by different sensors (which impacts the effective VoI of the transmitted data). We assume that these input variables are passed to the scheduler using specific interfaces, whose definition is out of the scope of this work. In the following, we provide a more formal description of the variables, and describe their meaning and use.

Let N be the number of objects generated in a batch by an application. Hence, the scheduler is provided with the following inputs.

- The available capacity C along the path, defined as the product of the bottleneck capacity and the minimum Round Trip Time (RTT). These values are estimated directly by the recent bottleneck bandwidth and round-trip propagation time (BBR) congestion control algorithm [212], and can be obtained indirectly when using other latency-aware mechanisms such as the classic Vegas algorithm (which, however, tends to underestimate the capacity in volatile scenarios). In our implementation, we consider the estimate provided by BBR, but note that QUIC natively supports both Vegas and BBR.

- The *size vector* $\mathbf{s} \in \mathbb{N}^N$, which contains the sizes of the objects, in bits. This information is used to check that the amount of data scheduled for transmission does not exceed the connection capacity C, to avoid congestion.

- The *weight vector* $\mathbf{v} \in [0;1]^N$, which contains the *intrinsic value* of the objects, i.e., the VoI when considering only that source. The intrinsic VoI can depend on a number of factors, such as the position of the sensor (e.g., front sensors in an autonomous vehicle are generally more informative than side sensors for driving decisions, or the finger sensors in a haptic

application are more informative than wrist sensors), and the current state of the process (e.g., the presence of an object in a camera's Field-of-View (FoV), or the detection of a sudden gesture in haptic applications). The intrinsic VoI can also depend on the *time correlation* of the sampling process. If the process is slow-varying, consecutive readings from the same sensor can be highly correlated and, hence, easily predictable by the receiver. Although the relation between the time since the last update from a sensor and the correlation with the new reading is highly application-dependent, it is often assumed to follow an exponential decrease [208]. Some control applications have inbuilt compensation mechanisms for delay, which do not require new measurements until a certain time has passed, so the correlation for these cases can be modeled as a step function. A sigmoid function can then be used to model an imperfect compensation mechanism with a gentler degradation curve. Given the specificities of the different applications, we assume that the intrinsic VoI is determined by the application itself, and passed to the scheduling algorithm in the form of the weight vector. In the next section we will provide examples of how these values can be computed in the two considered use cases.

- The *cross-sensor correlation matrix* $\mathbf{W} \in [0;1]^{N \times N}$, which contains the correlation between objects. Indeed, if the application relies on multiple sensors, there is often a significant amount of redundancy in their information. For example, multiple cameras might have partially overlapping FoVs, or scalar sensors might be measuring correlated quantities. Therefore, the intrinsic VoI of some data may need to be discounted to account for the cross-sensor correlation, because the effective VoI of two correlated measurements can be lower than the sum of the VoIs of the two individual measurements.

The scheduling problem consists in selecting the sensor measurements that provide the maximum VoI at the receiver, while respecting the capacity constraint, i.e., having a total size that is lower than the bandwidth-delay product of the connection. Computing the VoI for all possible scheduling patterns is a combinatorial problem, which soom becomes unfeasible. However, if we limit the analysis to couples of objects, i.e., we do not consider the effects of triplets of correlated objects, this is an instance of the Quadratic Knapsack Problem (QKP) [213], which is NP-hard, but for which there are efficient heuristic solutions.
Figure 7.2 shows the basic structure of the proposed scheduling framework: multiple sensors write data with a given VoI to a QUIC socket, and the application supplies the cross-sensor correlation matrix $\mathbf{W}$. The available capacity is read

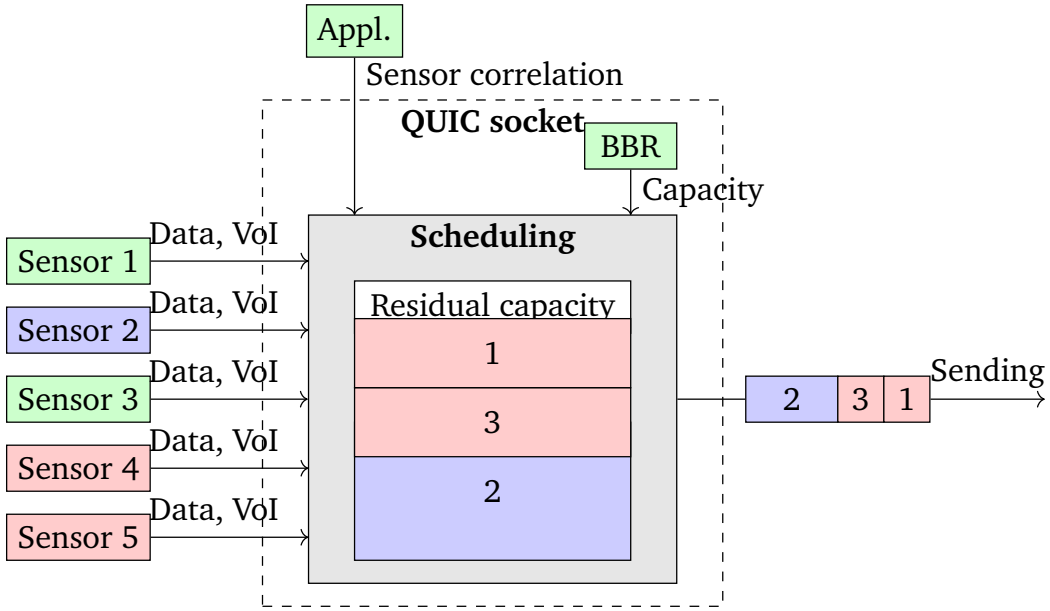


Figure 7.2: Basic components of the framework and main data exchanges. In the figure, the data from sensors 1 and 5 is discarded, while the data from sensors 2, 3, and 4 is sent in that order.

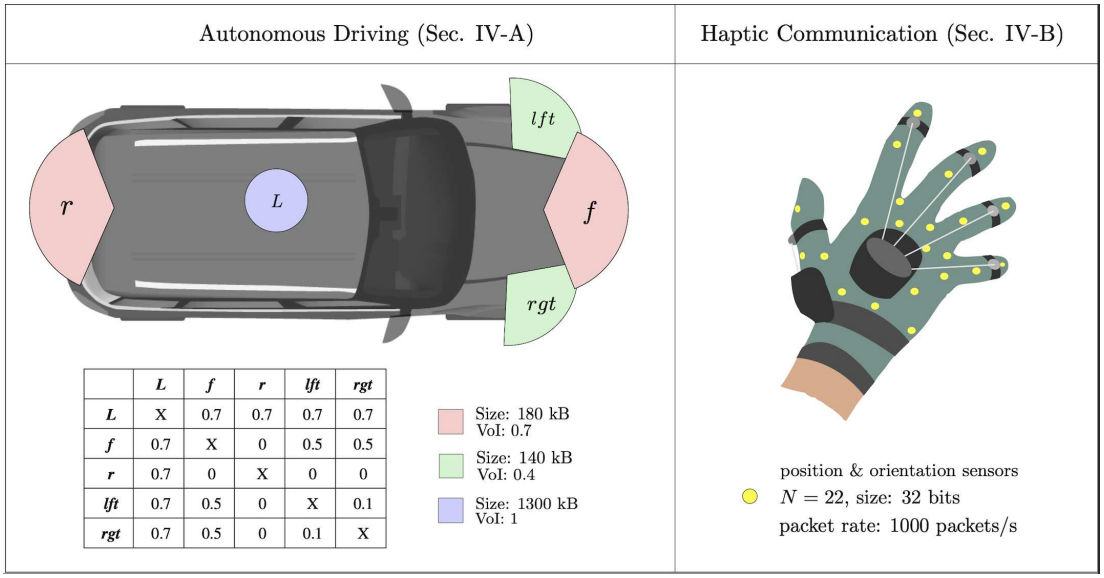
from the BBR estimate, and the scheduler finds the optimal set of objects that can be delivered before the next sensor update, sending them through the connection as fast as congestion control allows. If the connection is lossy or time-varying, scheduling decisions can be revised based on what was already sent, recomputing the solution to the problem.

To the best of our knowledge, transport layer scheduling of multi-sensory data is an open research problem, which requires the study of the application and sensor features and the estimation of end-to-end capacity dynamics. QUIC-EST forgets the problem by considering correlated measurements in time and across multiple sensors and using congestion control to estimate the path capacity. The scheduling framework is relatively simple, but it can support a wide range of applications, guaranteeing reliability and maximizing the delivered VoI.

7.4 Use-case Scenarios for QUIC-EST

The methodology we propose can be applied to any type of application that generates correlated data streams, whose relative importance can be represented in the form of a VoI matrix. In the following, we give two examples of such applications, namely, autonomous driving and haptic communication.

In our scheduler, data transmission is discriminated based on the VoI provided by the application layer, which depends on the intrinsic characteristics of the different sensors and on the time correlation of consecutive observations. We



	L	f	r	lft	rgt
L	X	0.7	0.7	0.7	0.7
f	0.7	X	0	0.5	0.5
r	0.7	0	X	0	0
lft	0.7	0.5	0	X	0.1
rgt	0.7	0.5	0	0.1	X

Figure 7.3: Scheduling input parameters for the autonomous driving (left) and haptic communication (right) scenarios.

remark that determining the VoI is not the main focus of this contribution. For the sake of completeness, however, we consider a simple and intuitive definition of the VoI, based on expert knowledge about the relative importance of the different information sources and of their temporal obsolescence. The same empirical approach is used to determine the cross-sensor correlation matrix, which evaluates the degree of correlation among the different sensors. Clearly, more sophisticated strategies may yield different values for the VoI and the correlation matrix that, once fed into QUIC-EST, may result in different transmission strategies. Nonetheless, the values considered in this study are reasonable and apt to illustrate the rationale of the proposed scheme.

7.4.1 QUIC-EST for Autonomous Driving

Size vector. The size vector depends on the type of automotive sensor that is considered, the rate at which observations are made, and their resolution. For example, Light Detection and Ranging (LIDAR) sensors can generate data flows with a rate from about 50 kbps to 30 Mbps, depending on the sensor resolution. Similarly, the data rate for camera images ranges from 10 Mbps to 500 Mbps, depending on the image resolution [214], even though compression can reduce the image size by several orders of magnitude. In this work, we consider $N = 5$ sensors: two cameras on the vehicles' top left (lft) and top right (rgt) corners, one on the front (f) and one on the rear side (r), and one LIDAR on the rooftop of the car (L). The sizes of the sensor observations are calculated based on the nuScenes dataset [215], which contains a full autonomous vehicle sensor suite,

assuming a 1 Byte pixel encoding and JPG compression for the camera images: we consider a size for the front/rear cameras of 180 KB, for the lateral cameras of 140 KB, and for the LIDAR of 1300 KB, as depicted in Figure 7.3 (left).

Intrinsic VoI. We reasonably expect that the LIDAR would be more valuable compared to automotive cameras because it can provide a three-dimensional (rather than bi-dimensional) representation of the environment, and can work efficiently in different weather/time conditions. Also, we assume that the importance of the images taken by the cameras depends on the characteristics of the environment in which the vehicles move (e.g., in the highway scenario lateral cameras will likely make background observations with little marginal information, while frontal/rear cameras might provide more valuable information). Based on these assumptions, we empirically define the correlation vector $\mathbf{v} \in [0, 1]^N$ as shown in Figure 7.3 (left). Moreover, following the method suggested in [208], we account for the temporal obsolescence of the information by means of an exponential function that depends on the relative age of information, i.e., the time between the generation and reception of the information, with a temporal decay parameter that is proportional to the delay sensitivity of the observation, i.e., the temporal horizon over which that piece of information is considered valuable.

Cross-sensor correlation. We assume that the correlation between images taken by different cameras is proportional to the overlapping of their FoVs. Therefore, the rear camera's images are uncorrelated with those of any other camera. The images taken by lateral cameras are slightly cross-correlated, while higher correlation can be assumed between the frontal and lateral cameras. On the other hand, the LIDAR sensor operates through 360-degree rotations and its observations can be highly correlated/redundant with those of the cameras. The correlation matrix is hence structured as displayed in Figure 7.3 (left).

7.4.2 QUIC-EST for Haptic Communication

Size vector. In this scenario, the size vector should depend on the number of sensors and actuators integrated on the haptic devices. The CyberGrasp [216] device combines a haptic glove that can sense orientation and movement of the hand and an exoskeleton with five kinesthetic actuators for providing force feedback to the user. Since each haptic glove has 22 movement sensors, considering two hands we have in total $N = 44$ sources of sensor data. Each sensor transmits one floating-point value (i.e., typically 32 bits using the IEEE 754 standard) with a 1 kHz sampling rate, resulting in a 1.4 Mbps total data rate, as represented in Figure 7.3 (right).

Intrinsic VoI. In order to determine the VoI of each data sample generated by the haptic device's sensors, we rely on the psycho-physical aspects of human perception. More specifically, we can use Weber's law of Just Noticeable Difference (JND), as in the deadband transmission algorithm in [217], which can be applied in position, velocity and force data. The VoI is then given by the difference between the last transmitted sample from that sensor and the current value, which can be easily computed by the sending application and given to the scheduler. Sensors have the same inherent VoI, but the actual value of the information depends on how novel it is with respect to the one currently available to the receiver. This definition implicitly includes the time correlation between samples, as the difference between consecutive samples will usually be small, but then grow with time consistently with the age of the data.

Cross-sensor correlation. In the haptic communication case, the flexibility of a robotic hand makes the relation between different sensors strongly dependent on their position. If the hand is grasping an object, the correlation between sensors will be different from when it is at rest. Consequently, we cannot give a constant cross-sensor correlation matrix based on the sensors' positions, like we did in the vehicular case. Ideally, the application should be able to compute the instantaneous correlation between sensors in real time and pass the correlation vector to the scheduler. As a simpler (and likely suboptimal) alternative, here we consider the measurements to be independent.

7.5 Performance Evaluation

In this section, we present a performance evaluation of QUIC-EST, comparing it with other scheduling algorithms in the two scenarios presented in Sec. 7.4, with extremely different features. While realistic, the assumptions about the two scenarios are arbitrary, and their purpose is to illustrate the methodology from a qualitative perspective, rather than giving a complete quantitative assessment of the scheme. The autonomous vehicle in the first scenario transmits only 10 frames per second but with a maximum rate of 155.2 Mbps; on the other hand, haptic communication has a maximum rate of just 1.4 Mbps, but its sampling frequency is 100 times higher, i.e., 1 kHz. Furthermore, while the haptic communication scenario has 44 different sensors that need to be scheduled, the autonomous driving scenario only has 5.

In both scenarios, we study the average VoI as a function of the available (constant) connection capacity. We consider three other schedulers:

- *First In First Out (FIFO)*. This is the default QUIC scheduler, which transmits pieces of data in the same order they were received from the application. It limits transmission to the achievable send rate, discarding any objects that would exceed the connection's capacity. We consider this as a baseline, as its behavior is similar to TCP, without the head-of-line blocking.

- *VoI-based*. This scheduler considers the VoI as the decision factor for transmitting objects that fit the transmission capacity. It is an instance of the classic knapsack problem, as it does not consider cross-sensor correlation or even temporal correlation between values, but only the intrinsic value of each sensor.

- *Cross-sensor VoI*. This scheduler considers cross-sensor correlation, but neglects the temporal correlation. It is equivalent to the optimal scheduler if subsequent measurements from the same sensor are independent, and to the VoI-based scheduler if the measurements are independent between different sensors as well.

- *QUIC-EST*. The scheduler considers VoI as well as time and cross-sensor correlation. The scheduling is obtained by using existing solvers for the QKP. This scheduler gives the best performance if we consider the full application model.

The performance analysis is based on MATLAB simulations, and the code has been made publicly available.[1]
In Figure 7.4 we report the normalized VoI achieved by the different schedulers when varying the connection capacities in the autonomous driving scenario. The normalized VoI value is defined as the ratio between the average VoI achieved by each scheduler for a certain channel capacity over the VoI obtained with infinite capacity (which is the same for all the considered schedulers). As expected, the performance grows with the channel capacity for all schedulers, but cross-sensor VoI and QUIC-EST, which account for the cross-sensor correlation, exhibit a clear advantage over the others. As Figure 7.5 shows, this stark difference is due to the frequency at which the schedulers pick LIDAR frames, which are large and highly correlated with data from the cameras. When the channel capacity is limited, the schedulers that consider cross-correlation among the sensor data flows will limit the number of transmitted LIDAR frames, which are highly correlated, thus leaving more space from camera frames that, with the considered setting, have a higher joint value.

[1] `https://github.com/Anay191/Scheduling_Policies_QUIC`

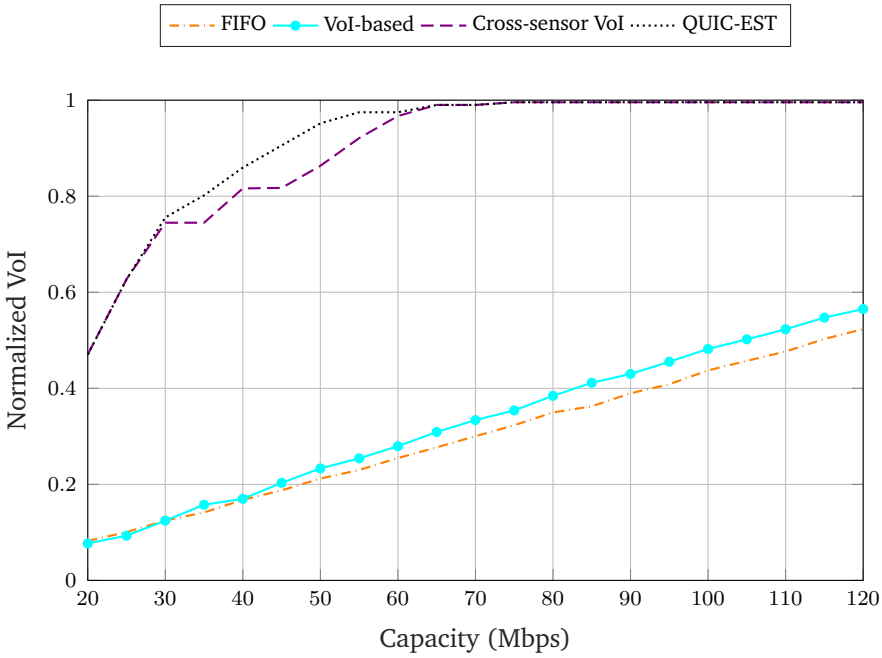


Figure 7.4: Normalized VoI for the different schedulers when varying the connection capacity, in the autonomous driving scenario.

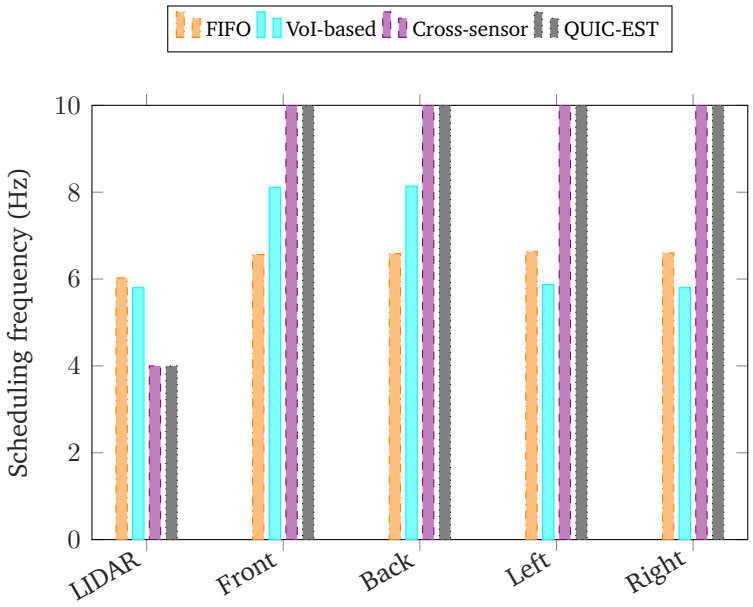


Figure 7.5: Average update frequency for the different schedulers in the autonomous driving scenario, with $C = 100$ Mbps.

For the haptic communication scenario, we consider the VoI as a logistic function of the difference between the current sample and the last transmitted one. We used realistic haptic traffic model parameters from [218] and cautiously selected a JND value of 5% of the dynamic range of the sensors. Accordingly, we simulate each sensor as an independent Gauss-Markov process, setting $\sigma = 2.15\%$ of

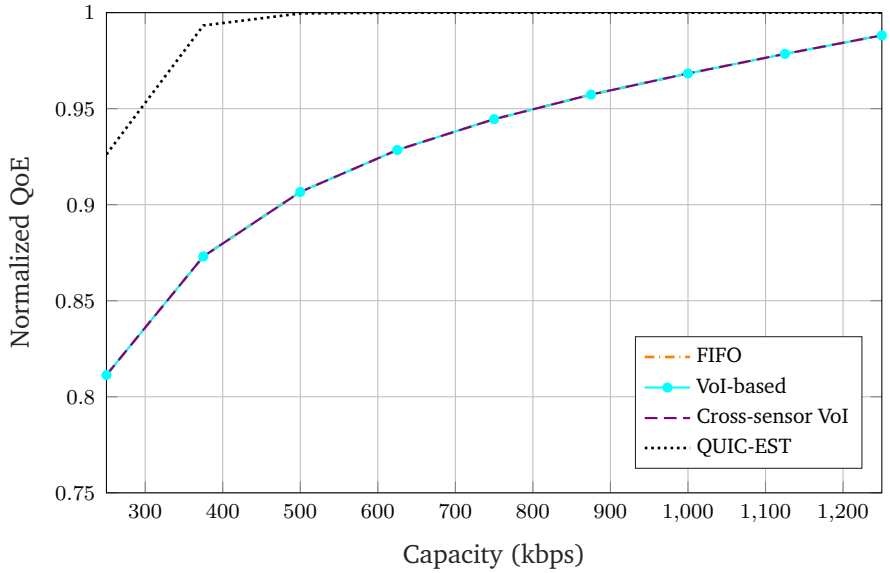


Figure 7.6: Comparison between schedulers in the haptic communication scenario in terms of the normalized QoE as a function of capacity.

the dynamic range to fit the empirical model from the paper. The VoI is then given by a logistic function with center $x_0 = 1.65\sigma$ and sharpness $k = 10$. These values ensure that all sensor measurements that differ for more than the JND are prioritized, while the remaining data are sent only in case of residual capacity. As mentioned, in the haptic communication scenario we neglect the cross-sensor correlation, and all sensors have the same intrinsic VoI, so that the FIFO, VoI-based, and cross-sensor VoI schedulers are all equivalent.

Figure 7.6 shows the normalized QoE, defined as the overall fraction of time sensors are under the JND threshold, when varying the channel capacity. We can observe that the QUIC-EST scheme can achieve almost perfect QoE even at less than a third of the capacity needed to send all packets. In this case, the time correlation is critical: the schedulers that do not use a JND-based value, indeed, achieve a lower performance. To be noted that the availability of cross-sensor correlation estimates could further improve the QUIC-EST performance, decreasing the amount of transmission resources needed to support the application.

7.6 Conclusions

In this chapter, we have presented QUIC-EST, a flexible transmission scheme obtained by combining the emerging QUIC protocol and a VoI-based scheduling strategy that consider capacity predictions and defines scheduling to maximise VoI at the receiver for multi-sensory applications with time-sensitive data. We showed that this scheme can be adapted to widely different applications with

good results, using autonomous driving and haptic communication as our Future Internet use cases.

At the moment, the design of general transport frameworks for multiple parallel streams of correlated data is still an open research area, whose importance is rising with the popularity of this kind of applications. The combination of such new transport protocols with network slicing techniques is another research avenue of great interest, given the potential to provide applications with reliability guarantees, which can be fundamental for safety-critical services.

Conclusions

8

Next generation networks have shown the necessity to analyse different types of information and design systems based on anticipating user and network parameters that can be exploited to optimize the performance of the wireless networks. On that note, this thesis provided a brief overview in design and analysis of system-wide optimization schemes based on anticipatory techniques for various different networking scenarios from MIMO cellular networks to FANETs and considering different future wireless network scenarios and applications such as Tactile Internet. Primarily, the thesis showed works on the Physical, Networking, Transport and Application Layers and hence gives a comprehensive understanding of system-wide optimization schemes that can be applied for different networking scenarios.

For physical layer optimization schemes, the thesis provided a brief definition of a novel Spatial Division Multiple Access (SDMA) scheme called Rate Splitting [219], specifically Hierarchical Rate Splitting (HRS) which needs to cluster the users together to maximise the rate achieved for each user while minimising the interference between the users. It highlights the predictive clustering scheme which use NN to cluster users based on CSI which reducing the complexity compared to hierarchical clustering mechanisms. Hence, the clustering schemes defines fast user clusters such that the rate achieved is maximised for each user.

Additionally, for network layer optimisation schemes, the thesis analyses the next generation networks such as FANETs which have been considered critical in 6G networks especially for providing connectivity in disaster/remote areas [220]. It highlights the routing design with respect to mobility of the UAVs while taking into account high capacity mmWave wireless links. First, it provides a brief analysis of route stability with respect uncertainties in position of the UAVs due to high mobility and then provides a brief analysis of routing scenario for mmWave beamforming designs. Hence, the thesis provides a brief analysis for system wide routing schemes for FANETs.

Also, for transport and application layer optimisation schemes, the thesis considers the next generation network scenario of tactile internet for haptic communication and autonomous driving [221]. The thesis highlights a Value of Information (VoI) based scheduling mechanism based on QUIC. It highlights

the work that takes capacity predictions for the wireless network and provides scheduling design that maximises VoI at the receiver end for haptic communication and autonomous driving scenarios. Hence, the thesis provides a brief analysis of application level scheduling schemes that maximises the QoE for the users.

Lastly, the thesis also enhances the fact that the *divide-et-impera* approach, which consists in splitting a highly complex problem in a number of more manageable sub-problems that are then solved independently, may actually prevent the exploitation of some synergies among different parts of the system or hide interactions among the parts resulting in suboptimal system performance. On the other hand, optimizing a complex system by accounting for all possible aspects is an extremely challenging problem, with obvious scaling issues. The thesis shows an intermediate approach, which consists in focusing on specific aspects of the system, to limit the complexity, but without losing a more general vision of the system, in order to exploit synergies and avoid pitfalls. How to express this basic principle in practice, however, depends strongly on the characteristics of the problem addressed and the scenario considered, so it is virtually impossible to identify a well-defined methodology for pursuing system-level optimization beyond the works highlighted above and exemplified in the studies reported in this thesis.

Finally, moving forward, the concept of anticipatory networking has provided an interesting avenue of research in design of system wide optimization techniques. As highlighted in the thesis, it is shown to be useful in multiple different facets in a wireless communication system and, thereby, proving to be interesting research field for system design in next generation wireless networks.

List of Publications

Publications

- Deshpande, A.A., Chiariotti, F. and Zanella, A., 2020, June. SMURF: Reliable multipath routing in flying ad-hoc networks. In 2020 Mediterranean Communication and Computer Networking Conference (MedComNet) (pp. 1-8). IEEE

- Paro, U., Chiariotti, F., Deshpande, A.A., Polese, M., Zanella, A. and Zorzi, M., 2020, November. Extending the ns-3 QUIC Module. In Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 19-26).

- Chiariotti, F., Deshpande, A.A., Giordani, M., Antonakoglou, K., Mahmoodi, T. and Zanella, A., 2021. QUIC-EST: a QUIC-enabled scheduling and transmission scheme to maximize VoI with correlated data flows. IEEE Communications Magazine, 59(4), pp.30-36.

- Deshpande, A.A., Zanella, A., Pereira, R., Pastore, A., Mestre, X. and Chiariotti, F., 2022, May. Beam Aware Stochastic Multihop Routing for Flying Ad-hoc Networks. In 2022 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1065-1070). IEEE.

- Deshpande, A.A., Vaca-Rubio, C.J., Mohebi, S., Salami, D., De Carvalho, E., Popovski, P., Sigg, S., Zorzi, M. and Zanella, A., 2022, June. Energy-Efficient Design for RIS-assisted UAV communications in beyond-5G Networks. In 2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet) (pp. 158-165). IEEE.

- Pereira, R., Deshpande, A.A., Vaca-Rubio, C.J., Mestre, X., Zanella, A., Gregoratti, D., de Carvalho, E. and Popovski, P., 2022. User Clustering for Rate Splitting using Machine Learning. Accepted in EUSIPCO 2022 (2022).

Pending Publications

- Deshpande, A.A., Chiariotti, F. and Zanella, A., 2022, July. SMURF: A Reliable Multipath Routing Scheme for Flying Ad-Hoc Networks. Submitted to IEEE Internet of Things Journal.
- Deshpande, A.A., Zanella, A., Pereira, R., Mestre, X. and Chiariotti, F., 2022, September. Adaptive Beamforming based Stochastic Multihop Routing for Flying Ad-hoc Networks. Targetting IEEE Transactions for Wireless Communications.

Other Research Outputs

As part of H2020 Marie Sklodowska-Curie ITN WindMill Project (Grant Number: 813999), we produced the following research deliverables,

- D6.3 - Final report on system wide cognitive optimisation schemes
- D5.3 - Final report on machine learning techniques for radio resource management
- D3.3 - Machine learning for wireless communication
- D6.2 - Hierarchical and distributed learning architecture and multi-objective optimisation strategies
- D5.2 - Intermediate report on machine learning techniques for radio resource management
- D6.1 - State of the art on context acquisition & anticipatory optimisation techniques for network optimisation
- D5.1 - Initial report on ML for radio resource management and initial datasets

Acknowledgements

This thesis represents the final destination of a long journey, but also a monumental proof of the path I have taken since I left my home and my family. Therefore, I take this opportunity to express my sincere gratitude to the following people and organizations, who supported me during the writing of the thesis and the research behind it.

My supervisor, Prof. Andrea Zanella, for his enthusiasm, trust, useful feedbacks and guidance, and for mentoring and supporting me throughout my doctoral program. The whole SIGNET and WINDMILL research group, students and professors for the time spent together.

The ISPIC research team of the Centre Tecnològic Telecomunicacions de Catalunya (CTTC), especially Dr. Adriano Pastore and Dr. Xavier Mestre, with whom I collaborated to explore Joint Beamforming and Routing Designs for FANETs. Also, thank you to everyone else at CTTC who assimilated me quickly in their hangouts and coffee breaks.

My colleagues in the WINDMILL Project, Salman, Roberto, Cristian, Dariush, Pedro, Shirin, Matteo, Jadoon, Mateus, Parham, Sobhi, Davit, Stanley and Sergey with whom, I instantly connected during the first interactions way back in 2019 in Aalborg and who helped me through the tough times during Covid, as well as all the collaborations, technical and non-technical, during the entire PhD.

My collegues at DEI, University of Padova who helped me adjust myself in a new environment very quickly and giving valuable advise along the way: Martina, Davide, MarcoC, Giovanni, Jacopo, FedeM, Paolo, MatteoD, MatteoP, Silvia, Francesco, Francesca, Tommaso, Elvina, MarcoG, Michele. Lastly, a special mention to Federico Chiariotti, whose helped me understand the intricacies of research during my entire PhD and with whom I shared one of my favourite conference trips in Seoul, South Korea.

Lastly, I also am very grateful of my friends, Srivats, Aditya, Moon, Suman and Madhuri, who I had not been able to meet in person for a quite awhile but they have made sure to keep us connected online everyday during the Covid lockdowns and further even with timezone differences.

Finally, the nest I left early on to carry on this journey. I thank my mum Seema and my dad Ajit for being virtually next to me all the time, my brother Soham, who would engage with me on football to relax my mind and my whole family and relatives.

Bibliography

[1] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang, and N. El-Sheimy, "Deep reinforcement learning (drl): Another perspective for unsupervised wireless localization," *ieee internet of things journal*, vol. 7, no. 7, pp. 6279–6287, 2019.

[2] O. Kanhere and T. S. Rappaport, "Position locationing for millimeter wave systems," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 206–212.

[3] O. Kanhere, S. Ju, Y. Xing, and T. S. Rappaport, "Map-assisted millimeter wave localization for accurate position location," in *2019 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2019, pp. 1–6.

[4] J. Gante, G. Falcão, and L. Sousa, "Deep Learning Architectures for Accurate Millimeter Wave Positioning in 5G," *Neural Processing Letters*, vol. 51, no. 1, pp. 487–514, 2020.

[5] U Palani, K. Suresh, and A. Nachiappan, "Mobility prediction in mobile ad hoc networks using eye of coverage approach," *Cluster Computing*, vol. 22, no. 6, pp. 14 991–14 998, 2019.

[6] Y. Yayeh, H.-p. Lin, G. Berie, A. B. Adege, L. Yen, and S.-S. Jeng, "Mobility prediction in mobile ad-hoc network using deep learning," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, IEEE, 2018, pp. 1203–1206.

[7] K.-L. Yap, Y.-W. Chong, and W. Liu, "Enhanced handover mechanism using mobility prediction in wireless networks," *PloS one*, vol. 15, no. 1, e0227982, 2020.

[8] L. N. Balico, A. A. Loureiro, E. F. Nakamura, R. S. Barreto, R. W. Pazzi, and H. A. Oliveira, "Localization prediction in vehicular ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2784–2803, 2018.

[9] H. Zhang and L. Dai, "Mobility prediction: A survey on state-of-the-art schemes and future applications," *IEEE Access*, vol. 7, pp. 802–822, 2018.

[10] R. Kasana, S. Kumar, O. Kaiwartya, W. Yan, Y. Cao, and A. H. Abdullah, "Location error resilient geographical routing for vehicular ad-hoc networks," *IET Intelligent Transport Systems*, vol. 11, no. 8, pp. 450–458, 2017.

[11] L.-l. Wang, Z.-g. Chen, and J. Wu, "Vehicle trajectory prediction algorithm in vehicular network," *Wireless Networks*, vol. 25, no. 4, pp. 2143–2156, 2019.

[12] W. Zhang, Y. Liu, T. Liu, and C. Yang, "Trajectory prediction with recurrent neural networks for predictive resource allocation," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*, IEEE, 2018, pp. 634–639.

[13] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7957–7969, 2019.

[14] Q. Liu, G. Chuai, J. Wang, and J. Pan, "Proactive Mobility Management With Trajectory Prediction Based on Virtual Cells in Ultra-Dense Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8832–8842, 2020.

[15] Y. Li *et al.*, "Wireless fingerprinting uncertainty prediction based on machine learning," *Sensors*, vol. 19, no. 2, p. 324, 2019.

[16] X. Yan, Q. Luo, P. Zhou, and J. Liu, "An Uncertainty Propagation Mechanism Analysis Method for Three-dimensional Quadrilateral Localization," in *2019 Prognostics and System Health Management Conference (PHM-Paris)*, IEEE, 2019, pp. 39–44.

[17] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1994–1998, 2019.

[18] G. Liu, Y. Xu, Z. He, Y. Rao, J. Xia, and L. Fan, "Deep learning-based channel prediction for edge computing networks toward intelligent connected vehicles," *IEEE Access*, vol. 7, pp. 114 487–114 495, 2019.

[19] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5g wireless communications: A deep learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 227–236, 2018.

[20] J. Huang *et al.*, "A big data enabled channel model for 5g wireless communication systems," *IEEE Transactions on Big Data*, vol. 6, no. 2, pp. 211–222, 2018.

[21] J. Joo, M. C. Park, D. S. Han, and V. Pejovic, "Deep learning-based channel prediction in realistic vehicular communications," *IEEE Access*, vol. 7, pp. 27 846–27 858, 2019.

[22] K. N. Qureshi, F. Bashir, and A. H. Abdullah, "Distance and signal quality aware next hop selection routing protocol for vehicular ad hoc networks," *Neural Computing and Applications*, vol. 32, no. 7, pp. 2351–2364, 2020.

[23] A. Akbas, H. U. Yildiz, A. M. Ozbayoglu, and B. Tavli, "Neural network based instant parameter prediction for wireless sensor network optimization models," *Wireless Networks*, vol. 25, no. 6, pp. 3405–3418, 2019.

[24] S. Naimi, A. Busson, V. Vèque, and R. Bouallegue, "Metric anticipation to manage mobility in mobile mesh and ad hoc wireless networks," *Annals of Telecommunications*, vol. 73, no. 11-12, pp. 787–802, 2018.

[25] M. Naravani, D. Narayan, S. Shinde, and M. M. Mulla, "A Cross-Layer Routing Metric with Link Prediction in Wireless Mesh Networks," *Procedia Computer Science*, vol. 171, pp. 2215–2224, 2020.

[26] W. Sun, Q. Li, J. Wang, L. Chen, D. Mu, and X. Yuan, "A radio link reliability prediction model for wireless sensor networks," *International Journal of Sensor Networks*, vol. 27, no. 4, pp. 215–226, 2018.

[27] J. Wen, M. Sheng, J. Li, and K. Huang, "Assisting Intelligent Wireless Networks with Traffic Prediction: Exploring and Exploiting Predictive Causality in wireless traffic," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 26–31, 2020.

[28] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[29] H. Luo, M. He, Z. Ruan, and F. Chen, "A duty-cycle mac algorithm with traffic prediction for wireless sensor networks," in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, IEEE, 2017, pp. 16–19.

[30] L. Nie, D. Jiang, S. Yu, and H. Song, "Network traffic prediction based on deep belief network in wireless mesh backbone networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2017, pp. 1–5.

[31] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1389–1401, 2019.

[32] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1656–1659, 2018.

[33] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, "Wireless traffic prediction with scalable Gaussian process: Framework, algorithms, and verification," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1291–1306, 2019.

[34] A. Adel Aly, H. M ELAttar, H. ElBadawy, and W. Abbas, "Aggregated Throughput Prediction for Collated Massive Machine-Type Communications in 5G Wireless Networks," *Sensors*, vol. 19, no. 17, p. 3651, 2019.

[35] L. Tao, Y. Gong, S. Jin, and J. Zhao, "Energy-efficient HTTP Adaptive Streaming with Anticipated Channel Throughput Prediction in Wireless Networks," in *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, 2017, pp. 253–258.

[36] D. Raca *et al.*, "On Leveraging Machine and Deep Learning for Throughput Prediction in Cellular Networks: Design, Performance, and challenges," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 11–17, 2020.

[37] D. Raca *et al.*, "Empowering video players in cellular: throughput prediction from radio network measurements," in *Proceedings of the 10th ACM Multimedia Systems Conference*, 2019, pp. 201–212.

[38] S. Kandasamy, R. Morla, P. Ramos, and M. Ricardo, "Predicting throughput in IEEE 802.11 based wireless networks using directional antenna," *Wireless Networks*, vol. 25, no. 4, pp. 1567–1584, 2019.

[39] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, and F. Naït-Abdesselam, "An optimized proactive caching scheme based on mobility prediction for vehicular networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2018, pp. 1–6.

[40] C. Li, Z. Liye, T. Hengliang, and L. Youlong, "Mobile user behavior based topology formation and optimization in ad hoc mobile cloud," *Journal of Systems and Software*, vol. 148, pp. 132–147, 2019.

[41] R. Q. Hu *et al.*, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 190–10 203, 2018.

[42] E. J. Kitindi and G. Rehman, "User Aware Edge Caching in 5G Wireless Networks," *IJCSNS*, vol. 18, no. 1, p. 25, 2018.

[43] R. Fernandes and R. D. GL, "A new approach to predict user mobility using semantic analysis and machine learning," *Journal of medical systems*, vol. 41, no. 12, p. 188, 2017.

[44] J. Wang, X. Kong, F. Xia, and L. Sun, "Urban human mobility: Data-driven modeling and prediction," *ACM SIGKDD Explorations Newsletter*, vol. 21, no. 1, pp. 1–19, 2019.

[45] Z. Deng, W. Liu, X. Yan, W. Wang, *et al.*, "Multisimi-Markov: An Improved Markov Position Prediction Method," in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, IEEE, 2018, pp. 1–7.

[46] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, "Social-aware resource allocation and optimization for D2D communication," *IEEE wireless communications*, vol. 24, no. 3, pp. 122–129, 2017.

[47] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118 112–118 124, 2019.

[48] W. Jiang and H. D. Schotten, "Deep Learning for Fading Channel Prediction," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 320–332, 2020.

[49] M. Li, M. Lin, W.-P. Zhu, Y. Huang, A. Nallanathan, and Q. Yu, "Performance analysis of MIMO MRC systems with feedback delay and channel estimation error," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 707–717, 2015.

[50] Q. Wang, L. J. Greenstein, L. J. Cimini, D. S. Chan, and A. Hedayat, "Multi-user and single-user throughputs for downlink MIMO channels with outdated channel state information," *IEEE Wireless Communications Letters*, vol. 3, no. 3, pp. 321–324, 2014.

[51] K. T. Truong and R. W. Heath, "Effects of channel aging in massive MIMO systems," *Journal of Communications and Networks*, vol. 15, no. 4, pp. 338–351, 2013.

[52] J.-B. Kim, J.-W. Choi, and J. M. Cioffi, "Cooperative distributed beamforming with outdated CSI and channel estimation errors," *IEEE Transactions on Communications*, vol. 62, no. 12, pp. 4269–4280, 2014.

[53] Y. Isukapalli, R. Annavajjala, and B. D. Rao, "Performance analysis of transmit beamforming for MISO systems with imperfect feedback," *IEEE Transactions on Communications*, vol. 57, no. 1, pp. 222–231, 2009.

[54] X. Yu, W. Xu, S.-H. Leung, and J. Wang, "Unified performance analysis of transmit antenna selection with OSTBC and imperfect CSI over Nakagamim fading channels," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 494–508, 2017.

[55] Y. Teng, M. Liu, and M. Song, "Effect of outdated CSI on handover decisions in dense networks," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2238–2241, 2017.

[56] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li, "Spectrum and power allocation for vehicular communications with delayed CSI feedback," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 458–461, 2017.

[57] J. Zheng and B. D. Rao, "Capacity analysis of MIMO systems using limited feedback transmit precoding schemes," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 2886–2901, 2008.

[58] P. Aquilina and T. Ratnarajah, "Performance analysis of IA techniques in the MIMO IBC with imperfect CSI," *IEEE Transactions on Communications*, vol. 63, no. 4, pp. 1259–1270, 2015.

[59] X. Chen and C. Yuen, "On interference alignment with imperfect CSI: Characterizations of outage probability, ergodic rate and SER," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 47–58, 2015.

[60] J. L. Vicario, A. Bel, J. A. Lopez-Salcedo, and G. Seco, "Opportunistic relay selection with outdated CSI: outage probability and diversity analysis," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 2872–2876, 2009.

[61] M. Chen, T. C.-K. Liu, and X. Dong, "Opportunistic multiple relay selection with outdated channel state information," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1333–1345, 2012.

[62] D. J. Love, R. W. Heath, V. K. Lau, D. Gesbert, B. D. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE Journal on selected areas in Communications*, vol. 26, no. 8, pp. 1341–1365, 2008.

[63] W. Jiang, T. Kaiser, and A. H. Vinck, "A robust opportunistic relaying strategy for co-operative wireless communications," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2642–2655, 2015.

[64] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proceedings of the IEEE*, vol. 95, no. 12, pp. 2299–2313, 2007.

[65] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1650–1662, 2005.

[66] R. O. Adeogun, P. D. Teal, and P. A. Dmochowski, "Extrapolation of MIMO mobile-to-mobile wireless channels using parametric-model-based prediction," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4487–4498, 2014.

[67] W. Jiang and H. D. Schotten, "A comparison of wireless channel predictors: Artificial intelligence versus kalman filter," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, 2019, pp. 1–6.

[68] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[69] Y. Liao, Y. Hua, X. Dai, H. Yao, and X. Yang, "ChanEstNet: A deep learning based channel estimation for high-speed scenarios," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, 2019, pp. 1–6.

[70] J. Yuan, H. Q. Ngo, and M. Matthaiou, "Machine learning-based channel estimation in massive MIMO with channel aging," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, IEEE, 2019, pp. 1–5.

[71] C. Potter, K. Kosbar, and A. Panagos, "MIMO channel prediction using recurrent neural networks," 2008.

[72] G Routray and P Kanungo, "Rayleigh fading MIMO channel prediction using RNN with genetic algorithm," in *International Conference on Computational Intelligence and Information Technology*, Springer, 2011, pp. 21–29.

[73] R.-F. Liao, H. Wen, J. Wu, H. Song, F. Pan, and L. Dong, "The Rayleigh fading channel prediction via deep learning," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[74] J. Wang, Y. Ding, S. Bian, Y. Peng, M. Liu, and G. Gui, "UL-CSI data driven deep learning for predicting DL-CSI in cellular FDD systems," *IEEE Access*, vol. 7, pp. 96 105–96 112, 2019.

[75] M. Arnold, S. Dörner, S. Cammerer, S. Yan, J. Hoydis, and S. t. Brink, "Enabling FDD massive MIMO through deep learning-based channel prediction," *arXiv preprint arXiv:1901.03664*, 2019.

[76] D. Papakostas and D. Katsaros, "A Rich-Dictionary Markov Predictor for Vehicular Trajectory Forecasting," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, 2018, pp. 556–563.

[77] L. Menz, R. Herberth, C. Luo, F. Gauterin, A. Gerlicher, and Q. Wang, "An improved method for mobility prediction using a Markov Model and density estimation," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2018, pp. 1–6.

[78] J. Guo, L. Liu, S. Zhang, and J. Zhu, "Mobility Prediction with Missing Locations Based on Modified Markov Model for Wireless Users," in *2019 IEEE International Congress on Big Data (BigDataCongress)*, IEEE, 2019, pp. 132–138.

[79] H. Park and Y. Lim, "A Markov-based prediction algorithm for user mobility at heterogeneous cloud radio access network," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, IEEE, 2019, pp. 1–5.

[80] G. A. Montoya and Y. Donoso, "A prediction algorithm based on Markov chains for finding the minimum cost path in a mobile wireless sensor network," in *2018 7th International Conference on Computers Communications and Control (ICCCC)*, IEEE, 2018, pp. 169–175.

[81] A. T. Akabane, R. W. Pazzi, E. R. Madeira, and L. A. Villas, "Modeling and prediction of vehicle routes based on hidden Markov Model," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, IEEE, 2017, pp. 1–5.

[82] D. Zhao, Y. Gao, Z. Zhang, Y. Zhang, and T. Luo, "Prediction of Vehicle Motion Based on Markov Model," in *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, IEEE, 2017, pp. 205–209.

[83] F. R. de Araújo, D. L. Rosário, K. Machado, E. C. Cerqueira, and L. Villas, "TEMMUS: A Mobility Predictor based on Temporal Markov Model with User Similarity," in *Anais Principais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, SBC, 2019, pp. 594–607.

[84] K.-L. Yap and Y.-W. Chong, "Optimized access point selection with mobility prediction using hidden Markov Model for wireless network," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, IEEE, 2017, pp. 38–42.

[85] J. Yan, C. Chen, and D. Zhang, "Node selection method based on Markov location prediction in vehicle networking," in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, IEEE, 2018, pp. 754–758.

[86] L. Yao, J. Wang, X. Wang, A. Chen, and Y. Wang, "V2X routing in a VANET based on the Hidden Markov Model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 889–899, 2017.

[87] S Nagadivya and R Manoharan, "Energy Efficient Markov Prediction Based Opportunistic Routing (Eempor) For Wireless Sensor Networks," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, IEEE, 2019, pp. 1–6.

[88] H. Liu, L. T. Yang, J. Chen, M. Ye, J. Ding, and L. Kuang, "Multivariate multi-order Markov multi-modal prediction with its applications in network traffic management," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 828–841, 2019.

[89] Y. Li, L. Lei, and M. Yan, "Mobile user Location Prediction Based on user Classification and Markov Model," in *2019 International Joint Conference on Information, Media and Engineering (IJCIME)*, IEEE, 2019, pp. 440–444.

[90] P. Wang, H. Wang, H. Zhang, F. Lu, and S. Wu, "A hybrid Markov and LSTM model for indoor location prediction," *IEEE Access*, vol. 7, pp. 185 928–185 940, 2019.

[91] Y. Qiao, Z. Si, Y. Zhang, F. B. Abdesslem, X. Zhang, and J. Yang, "A hybrid Markov-based model for human mobility prediction," *Neurocomputing*, vol. 278, pp. 99–109, 2018.

[92] M. Karimzadeh, Z. Zhao, F. Gerber, and T. Braun, "Pedestrians complex behavior understanding and prediction with hybrid Markov chain," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, 2018, pp. 200–207.

[93] S.-H. Fang, L. Lin, Y.-T. Yang, X. Yu, and Z. Xu, "CityTracker: Citywide Individual and Crowd Trajectory Analysis Using Hidden Markov Model," *IEEE Sensors Journal*, vol. 19, no. 17, pp. 7693–7701, 2019.

[94] S. KORDNOORİ, H. MOSTAFAEİ, and M. BEHZADI, "An Application of a Semi Markov Model in Predicting the Fading Occurrences in Wireless Communication Channels," *Gazi University Journal of Science*, vol. 33, no. 2, pp. 566–577, 2020.

[95] P. Theerthagiri, "CoFEE: Context-aware futuristic energy estimation model for sensor nodes using Markov Model and autoregression," *International Journal of Communication Systems*, e4248, 2019.

[96] P. Chaturvedi and A. Daniel, "Hidden Markov Model based node status prediction technique for target coverage in wireless sensor networks," in *2017 International Conference on Intelligent Communication and Computational Techniques (ICCT)*, IEEE, 2017, pp. 223–227.

[97] M. Saihi, A. Zouinkhi, B. Boussaid, M. N. Abdelkarim, and G. Andrieux, "Hidden Gaussian Markov Model for distributed fault detection in wireless sensor networks," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 6, pp. 1788–1798, 2018.

[98] F. Somaa, C. Adjih, I. El Korbi, and L. A. Saidane, "A Bayesian model for mobility prediction in wireless sensor networks," in *2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*, IEEE, 2016, pp. 1–7.

[99] Q. Liao, S. Valentin, and S. Stańczak, "Channel gain prediction in wireless networks based on spatial-temporal correlation," in *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, IEEE, 2015, pp. 400–404.

[100] Y. Hu, Q. Meng, and J. Zhu, "Bayesian Network Based Reliability Prediction in Mobile Wireless Networks," in *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*, IEEE, 2019, pp. 299–303.

[101] S. Mehrizi, A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "A feature-based Bayesian method for content popularity prediction in edge-caching networks," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2019, pp. 1–6.

[102] Z. Zhang, F. Liu, Z. Zeng, and W. Zhao, "A traffic prediction algorithm based on Bayesian spatio-temporal model in cellular network," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*, IEEE, 2017, pp. 43–48.

[103] X. Wang and Q. Zhou, "Wireless Traffic Prediction Based on Bayesian Seasonal Adjustment Algorithm," in *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, IEEE, 2020, pp. 431–435.

[104] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[105] R. Chen, Q. Hua, Y. Chang, B. Wang, L. Zhang, and X. Kong, "A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on social networks," *IEEE Access*, vol. 6, pp. 64 301–64 320, 2018.

[106] H. Ambulgekar, M. K. Pathak, and M. Kokare, "A survey on collaborative filtering: tasks, approaches and applications," in *Proceedings of International Ethical Hacking Conference 2018*, Springer, 2019, pp. 289–300.

[107] J. Yao, T. Han, and N. Ansari, "On Mobile Edge Caching," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2525–2553, 2019.

[108] H. S. Goian, O. Y. Al-Jarrah, S. Muhaidat, Y. Al-Hammadi, P. Yoo, and M. Dianati, "Popularity-based video caching techniques for cache-enabled networks: a survey," *IEEE Access*, vol. 7, pp. 27 699–27 719, 2019.

[109] H. T. Nguyen, H. D. Tuan, T. Q. Duong, H. V. Poor, and W. Hwang, "Collaborative Multicast Beamforming for Content Delivery by Cache-Enabled Ultra Dense Networks," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3396–3406, 2019.

[110] Y. Liu, A. Liu, N. N. Xiong, T. Wang, and W. Gui, "Content Propagation for Content-Centric Networking Systems From Location-Based Social Networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 10, pp. 1946–1960, 2019.

[111] S. Su and S. Zhao, "An optimal clustering mechanism based on Fuzzy-C means for wireless sensor networks," *Sustainable Computing: Informatics and Systems*, vol. 18, pp. 127–134, 2018.

[112] Y. Padmanaban and M. Muthukumarasamy, "Energy-efficient clustering algorithm for structured wireless sensor networks," *IET Networks*, vol. 7, no. 4, pp. 265–272, 2018.

[113] K. Lin, F. Xia, and G. Fortino, "Data-driven clustering for multimedia communication in Internet of vehicles," *Future Generation Computer Systems*, vol. 94, pp. 610–619, 2019.

[114] A. K. Dutta, M. Elhoseny, V. Dahiya, and K Shankar, "An efficient hierarchical clustering protocol for multihop Internet of vehicles communication," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, e3690, 2020.

[115] O. Senouci, Z. Aliouat, and S. Harous, "MCA-V2I: A Multi-hop Clustering Approach over Vehicle-to-Internet communication for improving VANETs performances," *Future Generation Computer Systems*, vol. 96, pp. 309–323, 2019.

[116] L. Chanama and O. Wongwirat, "A comparison of decision tree based techniques for indoor positioning system," in *2018 International Conference on Information Networking (ICOIN)*, 2018, pp. 732–737.

[117] A. Musa, G. D. Nugraha, H. Han, D. Choi, S. Seo, and J. Kim, "A decision tree-based nlos detection method for the uwb indoor location tracking accuracy improvement," *International Journal of Communication Systems*, vol. 32, no. 13, e3997, 2019.

[118] M. N. Yuldashev, A. I. Vlasov, and A. N. Novikov, "Energy-efficient algorithm for classification of states of wireless sensor network using machine learning methods," *Journal of Physics: Conference Series*, vol. 1015, p. 032 153, May 2018.

[119] A. Ahmed, A. Jalal, and K. Kim, "Region and Decision Tree-Based Segmentations for Multi-Objects Detection and Classification in Outdoor Scenes," in *2019 International Conference on Frontiers of Information Technology (FIT)*, 2019, pp. 209–2095.

[120] X. Wang, "Decision-Tree-Based Relay Selection in Dualhop Wireless Communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6212–6216, 2019.

[121] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[122] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[123] B. Sliwa, R. Falkenberg, T. Liebig, N. Piatkowski, and C. Wietfeld, "Boosting vehicle-to-cloud communication by machine learning-enabled context prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[124] M. Emara, K. Roth, L. G. Baltar, M. Faerber, and J. Nossek, "Nonlinear digital self-interference cancellation with reduced complexity for full duplex systems," in *WSA 2017; 21th International ITG Workshop on Smart Antennas*, VDE, 2017, pp. 1–6.

[125] H. Wang, Y. Zhou, and W. Sha, "Research on wireless coverage area detection technology for 5G mobile communication networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, 2017.

[126] B. Sliwa and C. Wietfeld, "Empirical analysis of client-based network quality prediction in vehicular multi-MNO networks," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, IEEE, 2019, pp. 1–7.

[127] B. Sliwa, R. Falkenberg, T. Liebig, J. Pillmann, and C. Wietfeld, "Machine learning based context-predictive car-to-cloud communication using multi-layer connectivity maps for upcoming 5g networks," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, IEEE, 2018, pp. 1–7.

[128] B. Sliwa, T. Liebig, R. Falkenberg, J. Pillmann, and C. Wietfeld, "Efficient machine-type communication using multi-metric context-awareness for cars used as mobile sensors in upcoming 5g networks," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, IEEE, 2018, pp. 1–6.

[129] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, "Millimeter Wave Base Stations with Cameras: Vision-Aided Beam and Blockage Prediction," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, IEEE, 2020, pp. 1–5.

[130] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[131] J. Ren and Z. Wang, "A novel deep learning method for application identification in wireless network," *China Communications*, vol. 15, no. 10, pp. 73–83, 2018.

[132] A. Saeed and M. Kolberg, "Towards optimizing WLANs power saving: Novel context-aware network traffic classification based on a machine learning approach," *IEEE Access*, vol. 7, pp. 3122–3135, 2018.

[133] L. Zhu, J. Zhang, Z. Xiao, X. Cao, D. O. Wu, and X.-G. Xia, "Joint power control and beamforming for uplink non-orthogonal multiple access in 5G millimeter-wave communications," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6177–6189, 2018.

[134] M. Sun, X. Xu, X. Tao, P. Zhang, and V. C. M. Leung, "NOMA-Based D2D-Enabled Traffic Offloading for 5G and Beyond Networks Employing Licensed and Unlicensed Access," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4109–4124, 2020.

[135] J. Tang *et al.*, "Energy efficiency optimization for NOMA with SWIPT," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 452–466, 2019.

[136] M. Moltafet, N. Mokari, M. R. Javan, H. Saeedi, and H. Pishro-Nik, "A new multiple access technique for 5G: Power domain sparse code multiple access (PSMA)," *IEEE Access*, vol. 6, pp. 747–759, 2017.

[137] F. Moradi, E. Fitzgerald, M. Pióro, and B. Landfeldt, "Flexible DRX Optimization for LTE and 5G," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 607–621, 2019.

[138] P. Huang and Y. Pi, "A novel MIMO channel state feedback scheme and overhead calculation," *IEEE Transactions on Communications*, vol. 66, no. 10, pp. 4550–4562, 2018.

[139] Y. Yang, S. Dang, Y. He, and M. Guizani, "Markov decision-based pilot optimization for 5G V2X vehicular communications," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1090–1103, 2018.

[140] E. Olfat and M. Bengtsson, "Joint channel and clipping level estimation for OFDM in IoT-based networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 18, pp. 4902–4911, 2017.

[141] I.-S. Comşa *et al.*, "Towards 5G: A reinforcement learning-based scheduling solution for data traffic management," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1661–1675, 2018.

[142] I.-S. Comşa, R. Trestian, G.-M. Muntean, and G. Ghinea, "5MART: A 5G sMART scheduling framework for optimizing QoS through reinforcement learning," *IEEE Transactions on Network and Service Management*, 2019.

[143] A. A. Esswie and K. I. Pedersen, "Opportunistic spatial preemptive scheduling for URLLC and eMBB coexistence in multi-user 5G networks," *Ieee Access*, vol. 6, pp. 38 451–38 463, 2018.

[144] A. Anand and G. de Veciana, "Resource allocation and HARQ optimization for URLLC traffic in 5G wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2411–2421, 2018.

[145] H. Jang, J. Kim, W. Yoo, and J.-M. Chung, "URLLC Mode Optimal Resource Allocation to Support HARQ in 5G Wireless Networks," *IEEE Access*, vol. 8, pp. 126 797–126 804, 2020.

[146] P. Hu and J. Zhang, "5G-Enabled Fault Detection and Diagnostics: How Do We Achieve Efficiency?" *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3267–3281, 2020.

[147] T. Wang, G. Zhang, X. Yang, and A. Vajdi, "Genetic algorithm for energy-efficient clustering and routing in wireless sensor networks," *Journal of Systems and Software*, vol. 146, pp. 196–214, 2018.

[148] R. Elhabyan, W. Shi, and M. St-Hilaire, "A Pareto optimization-based approach to clustering and routing in Wireless Sensor Networks," *Journal of Network and Computer Applications*, vol. 114, pp. 57–69, 2018.

[149] A. Moradi, "Broadcast Routing in Wireless Ad-Hoc Networks: A Particle Swarm optimization Approach," *Caspian Journal of Mathematical Sciences (CJMS)*, vol. 7, no. 1, pp. 46–67, 2018.

[150] B. Mao *et al.*, "A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks," *IEEE Wireless Communications*, vol. 25, no. 4, pp. 74–81, 2018.

[151] S. Chaudhary and R. Johari, "ORuML: Optimized Routing in wireless networks using Machine Learning," *International Journal of Communication Systems*, vol. 33, no. 11, e4394, 2020.

[152] M. D. F. De Grazia, D. Zucchetto, A. Testolin, A. Zanella, M. Zorzi, and M. Zorzi, "QoE multi-stage machine learning for dynamic video streaming," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 1, pp. 146–161, 2017.

[153] J. Wu, B. Cheng, M. Wang, and J. Chen, "Quality-aware energy optimization in wireless video communication with multipath TCP," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2701–2718, 2017.

[154] J. Heuschkel, E. Fleckstein, M. Ofenloch, and M. Muhlhauser, "UDP++: Cross-layer optimizable transport protocol for managed wireless networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2019, pp. 1–6.

[155] W. Yu and J. M. Cioffi, "Sum capacity of Gaussian vector broadcast channels," *IEEE Transactions on information theory*, vol. 50, no. 9, pp. 1875–1892, 2004.

[156] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[157] Y. Mao, B. Clerckx, and V. O. Li, "Rate-splitting multiple access for downlink communication systems: bridging, generalizing, and outperforming SDMA and NOMA," *EURASIP journal on wireless communications and networking*, vol. 2018, no. 1, pp. 1–54, 2018.

[158] M. Dai, B. Clerckx, D. Gesbert, and G. Caire, "A Rate Splitting Strategy for Massive MIMO With Imperfect CSIT," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4611–4624, 2016.

[159] A. Adhikary, J. Nam, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexing—The large-scale array regime," *IEEE transactions on information theory*, vol. 59, no. 10, pp. 6441–6463, 2013.

[160] R. Pereira, X. Mestre, and D. Gregoratti, "Subspace Based Hierarchical Channel Clustering in Massive MIMO," in *IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2021.

[161] F. Chollet *et al.*, *Keras*, `https://keras.io`, 2015.

[162] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[163] L. F. Gonzalez, G. A. Montes, E. Puig, S. Johnson, K. Mengersen, and K. J. Gaston, "Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation," *Sensors*, vol. 16, no. 1, p. 97, 2016.

[164] M. A. Ma'Sum *et al.*, "Simulation of intelligent unmanned aerial vehicle (UAV) for military surveillance," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, IEEE, 2013, pp. 161–166.

[165] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3359–3370, 2013.

[166] J.-D. M. M. Biomo, T. Kunz, and M. St-Hilaire, "An enhanced Gauss-Markov mobility model for simulations of unmanned aerial ad hoc networks," in *7th IFIP Wireless and Mobile Networking Conference (WMNC)*, IEEE, 2014, pp. 1–8.

[167] F. Mason, F. Chiariotti, M. Capuzzo, D. Magrin, A. Zanella, and M. Zorzi, "Combining LoRaWAN and a new 3D motion model for remote UAV tracking," in *IEEE INFOCOM Wireless Sensor, Robot and UAV Networks (WiSARN) Workshop*, 2020.

[168] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *2nd Workshop on Mobile Computing Systems and Applications (WMCSAA)*, IEEE, 1999, pp. 90–100.

[169] T. Clausen *et al.*, "Optimized link state routing protocol (OLSR)," *RFC 3626*, 2003.

[170] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 6753–6760.

[171] N. E. H. Bahloul, S. Boudjit, M. Abdennebi, and D. E. Boubiche, "A flocking-based on demand routing protocol for unmanned aerial vehicles," *Journal of Computer Science and Technology*, vol. 33, no. 2, pp. 263–276, 2018.

[172] M. Song, J. Liu, and S. Yang, "A mobility prediction and delay prediction routing protocol for UAV networks," in *10th International Conference on Wireless Communications and Signal Processing (WCSP)*, IEEE, 2018, pp. 1–6.

[173] A. Rovira-Sugranes and A. Razi, "Predictive routing for dynamic UAV networks," in *2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, IEEE, 2017, pp. 43–47.

[174] L. Lin, Q. Sun, S. Wang, and F. Yang, "A geographic mobility prediction routing protocol for ad hoc UAV network," in *Global Communications Conference Workshops (Globecom Workshops)*, IEEE, 2012, pp. 1597–1602.

[175] L. Lin, Q. Sun, J. Li, and F. Yang, "A novel geographic position mobility oriented routing strategy for UAVs," *Journal of Computational Information Systems*, vol. 8, no. 2, pp. 709–716, 2012.

[176] X. Li and J. Huang, "Abpp: An adaptive beacon scheme for geographic routing in FANET," in *18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, IEEE, 2017, pp. 293–299.

[177] P. Xie, "An enhanced OLSR routing protocol based on node link expiration time and residual energy in ocean FANETs," in *24th Asia-Pacific Conference on Communications (APCC)*, IEEE, 2018, pp. 598–603.

[178] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *nternational Multi Topic Conference (INMIC)*, IEEE, 2001, pp. 62–68.

[179] X. Li and J. Yan, "LEPR: Link stability estimation-based preemptive routing protocol for flying ad hoc networks," in *International Symposium on Computers and Communications (ISCC)*, IEEE, 2017, pp. 1079–1084.

[180] S. Rosati, K. Krużelecki, L. Traynard, and B. R. Mobile, "Speed-aware routing for UAV ad-hoc networks," in *Global Communications Conference Workshops (Globecom Workshops)*, IEEE, 2013, pp. 1367–1373.

[181] S. N. Pari and D Gangadaran, "A reliable prognostic communication routing for flying ad hoc networks," in *2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, 2018, pp. 33–38.

[182] G. Gankhuyag, A. P. Shrestha, and S.-J. Yoo, "Robust and reliable predictive routing strategy for flying ad-hoc networks," *IEEE Access*, vol. 5, pp. 643–654, 2017.

[183] C. J. Katila, A. Di Gianni, C. Buratti, and R. Verdone, "Routing protocols for video surveillance drones in IEEE 802.11s wireless mesh networks," in *European Conference on Networks and Communications (EuCNC)*, IEEE, 2017, pp. 1–5.

[184] J. Hong and D. Zhang, "TARCS: A topology change aware-based routing protocol choosing scheme of FANETs," *Electronics*, vol. 8, no. 3, p. 274, 2019.

[185] S.-C. Choi, H. R. Hussen, J.-H. Park, and J. Kim, "Geolocation-based routing protocol for flying ad hoc networks (FANETs)," in *10th International Conference on Ubiquitous and Future Networks (ICUFN)*, IEEE, 2018, pp. 50–52.

[186] A. V. Leonov, "Application of bee colony algorithm for FANET routing," in *17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, IEEE, 2016, pp. 124–132.

[187] Z. Zhao *et al.*, "Software-defined unmanned aerial vehicles networking for video dissemination services," *Ad Hoc Networks*, vol. 83, pp. 68–77, 2019.

[188] M. F. Khan, K.-L. A. Yau, R. M. Noor, and M. A. Imran, "Routing schemes in FANETs: A survey," *Sensors*, vol. 20, no. 1, p. 38, 2020.

[189] D. S. Lakew, U. Sa'ad, N.-N. Dao, W. Na, and S. Cho, "Routing in flying ad hoc networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1071–1120, 2020.

[190] Z. Drezner, "Computation of the multivariate normal integral," *ACM Transactions on Mathematical Software (TOMS)*, vol. 18, no. 4, pp. 470–480, 1992.

[191] P. N. Somerville, "Numerical computation of multivariate normal and multivariate-t probabilities over convex regions," *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 529–544, 1998.

[192] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[193] B. Barritt, T. Kichkaylo, K. Mandke, A. Zalcman, and V. Lin, "Operating a UAV mesh & internet backhaul network using temporospatial SDN," in *Aerospace Conference*, IEEE, 2017, pp. 1–7.

[194] I. Rodriguez *et al.*, "Analysis of 38 GHz mmWave propagation characteristics of urban scenarios," in *21th European Wireless Conference*, VDE, 2015, pp. 1–8.

[195] Z. Xiao *et al.*, "A survey on millimeter-wave beamforming enabled uav communications and networking," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 557–610, 2021.

[196] A. A. Deshpande, F. Chiariotti, and A. Zanella, "SMURF: Reliable Multipath Routing in Flying Ad-Hoc Networks," in *2020 Mediterranean Communication and Computer Networking Conference (MedComNet)*, IEEE, 2020, pp. 1–8.

[197] A. Al-Hourani, "On the probability of Line-of-Sight in urban environments," *IEEE Wireless Communications Letters*, vol. 9, no. 8, pp. 1178–1181, 2020.

[198] F. Noor, M. A. Khan, A. Al-Zahrani, I. Ullah, and K. A. Al-Dhlan, "A Review on Communications Perspective of Flying Ad-Hoc Networks: Key Enabling Wireless Technologies, Applications, Challenges and Open Research Topics," *Drones*, vol. 4, no. 4, p. 65, 2020.

[199] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017.

[200] D. Qiao, H. Qian, and G. Y. Li, "Broadbeam for massive MIMO systems," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2365–2374, 2016.

[201] T. Hu, "The maximum capacity route problem," *Operations Research*, vol. 9, no. 6, pp. 898–900, Dec. 1961.

[202] V. A. Aalo, G. P. Efthymoglou, and C. Chayawan, "On the envelope and phase distributions for correlated gaussian quadratures," *IEEE Communications letters*, vol. 11, no. 12, pp. 985–987, 2007.

[203] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[204] J. G. Andrews *et al.*, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[205] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, Oct. 1993.

[206] K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler, "Toward haptic communications over the 5G Tactile Internet," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3034–3059, Jun. 2018.

[207] T. Higuchi, M. Giordani, A. Zanella, M. Zorzi, and O. Altintas, "Value-anticipating V2V communications for cooperative perception," in *IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019, pp. 1947–1952.

[208] M. Giordani, A. Zanella, T. Higuchi, O. Altintas, and M. Zorzi, "Investigating Value of Information in Future Vehicular Communications," *IEEE Connected and Automated Vehicles Symposium (CAVS)*, 2019.

[209] A. Langley, A. Riddoch, A. Wilk, *et al.*, "The QUIC transport protocol: Design and internet-scale deployment," in *Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, Aug. 2017.

[210] R. Seggelmann, M. Tüxen, and E. P. Rathgeb, "Stream scheduling considerations for SCTP," in *18th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, IEEE, Sep. 2010, pp. 412–416.

[211] H. Shi, Y. Cui, F. Qian, and Y. Hu, "DTP: Deadline-aware transport protocol," in *3rd Asia-Pacific Workshop on Networking*, Aug. 2019.

[212] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *ACM Queue*, vol. 14, no. 5, pp. 20–53, Oct. 2016.

[213] D. Pisinger, "The quadratic knapsack problem—a survey," *Discrete applied mathematics*, vol. 155, no. 5, pp. 623–648, Mar. 2007.

[214] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Millimeter wave vehicular communications: A survey," *Foundations and Trends in Networking*, vol. 10, no. 1, pp. 1–118, Jun. 2016.

[215] H. Caesar *et al.*, "nuScenes: A multimodal dataset for autonomous driving," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE/CVF, Jun. 2020, pp. 11 621–11 631.

[216] M. Aiple and A. Schiele, "Pushing the limits of the CyberGrasp™ for haptic rendering," in *International Conference on Robotics and Automation*, IEEE, May 2013, pp. 3541–3546.

[217] P. Hinterseer, E. Steinbach, S. Hirche, and M. Buss, "A novel, psychophysically motivated transmission approach for haptic data streams in telepresence and teleaction systems," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, vol. 2, Mar. 2005, pp. 1097–1100.

[218] M. Abu-Tair and A. Marshall, "An empirical model for multi-contact point haptic network traffic," in *2nd International Conference on Immersive Telecommunications*, May 2009.

[219] Y. Mao, O. Dizdar, B. Clerckx, R. Schober, P. Popovski, and H. V. Poor, "Rate-splitting multiple access: Fundamentals, survey, and future research trends," *IEEE Communications Surveys & Tutorials*, 2022.

[220] M. M. Azari *et al.*, "Evolution of non-terrestrial networks from 5g to 6g: A survey," *IEEE Communications Surveys & Tutorials*, 2022.

[221] N. Promwongsa *et al.*, "A comprehensive survey of the tactile internet: State-of-the-art and research directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 472–523, 2020.