Marie Skłodowska Curie Action

WINDMILL

Machine Learning for Wireless Communications

H2020-MSCA-ITN-ETN
Grant Agreement Number: 813999

WP5–Data-driven optimisation schemes for radio access management

D5.1–Initial report on ML for radio resource management and initial datasets

Contractual Delivery Date:	Month 20 (August 2020)
Actual Delivery Date:	August 17, 2020
Responsible Beneficiary:	NBLF
Contributing Beneficiaries:	AAU, CTTC, NBLF, UNIPD, WSE
Dissemination Level:	Public
Version:	Draft

PROPRIETARY RIGHTS STATEMENT
This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813999.

PROPRIETARY RIGHTS STATEMENT
This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813999.

Document Information

Document ID:	WP5/D5.1
Version Date:	August 17, 2020
Total Number of Pages:	51
Abstract:	The term Radio Resource Management (RRM) describes the mechanisms used by wireless networks to conduct various operations. Today, those mechanisms are mostly the product of human minds, who have designed them and optimized them over time. This document introduces some well-known as well as other novel challenges in RRM and proposes some innovative approaches to solve them using machine learning. The problems covered by this report include random access, mobility, routing, scheduling, and scalability. Reinforcement-Learning approaches dominate most of the solutions proposed here. Finally, a framework for benchmarking solutions to wireless problems is also discussed.
Keywords:	Machine Learning, Reinforcement Learning, Random Access, Mobility, Routing, Scheduling, Drones, 802.11ad, Lorawan

Authors

Full name	**Beneficiary / Organisation**	**e-mail**	**Role**
Muhammad Jadoon	CTTC	muhammad.jadoon@cttc.es	Contributor
Chien-Cheng Wu	AAU	ccw@es.aau.dk	Contributor
Anay Desphande	UNIPD	anayajit.deshpande@unipd.it	Contributor
Salman Mohebi Ganjabadi	UNIPD	salman.mohebiganjabadi@unipd.it	Contributor
Ayesha Sahar	WSE	asahar@worldsensing.com	Contributor
Alvaro Valcarce	NBLF	alvaro.valcarce_rial@nokia-bell-labs.com	Deliverable co-ordinator

Table of Contents

List of Figures

List of Acronyms and Abbreviations

A-BFT Association-BeamForming Training

ADDTS Add Traffic Stream

ANN Artificial Neural Network

AP Access Point

AR Augmented Reality

ATI Announcement Transmission Interval

BHI Beacon Header Interval

BI Beacon Interval

BTI Beacon Transmission Interval

CBAP Contention-Based Access Period

EDCA Enhanced Distributed Channel Access

CPU Central Processing Unit

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

DTI Data Transmission Interval

DMG Directional Multi-Gigabit

eNB eNodeB

FANET Flying Ad-Hoc Network

gNB gNodeB

LoRaWAN Long Range Wide Area Network

MAC Medium Access Control

MCS Modulation and Coding Scheme

ML Machine Learning

mmW Millimeter Wave

mMTC Massive Machine Type Communications

MRO Mobility Robustness Optimisation

MU-MIMO Multi-User Multiple Input, Multiple Output

ns-3 Network Simulator 3

OFDM Orthogonal Frequency Division Multiplexing

PBSS Personal Basic Service Set

PCP Personal Basic Service Set (PBSS) Central Point

PCP/AP PBSS Central Point (PCP)/Access Point (AP)

PDF Probability Density Function

PDU Protocol Data Unit

PHY Physical Layer

QoS Quality of Service

RL Reinforcement Learning

RLC Radio Link Control

RLF Radio Link Failure

RRC Radio Resource Control

RRM Radio Resource Management

SDN Software Defined Network

SON Self-Organising Networks

SP Service Period

SSW Sector Sweep

STA Station

TDMA Time Division Multiple Access

TSPEC Traffic Specification

UAV Unmanned Aerial Vehicle

UE User Equipment

VBR Variable Bit Rate

VR Virtual Reality

WSN Wireless Sensor Network

1. Introduction

Wireless networks are possible thanks to the computing capabilities of modern hardware. Complex calculations are performed on every decision taken while serving User Equipments (UEs). From measurements filtering, to handover target evaluation, passing through signaling and Protocol Data Unit (PDU) construction. Every procedure executed by the network entails a detailed evaluation of hundreds of inputs and a forecast about the effects of future reconfiguration decisions. These evaluations are computationally expensive and essential to guarantee the connectivity and Quality of Service (QoS) of the served users.
Traditionally, the optimization of these procedures has been done manually by humans with years of experience tuning parameters for wireless networks. However, the evolution of mobile telephony and its ever growing complexity has rendered such an approach unscalable. Despite the progress made by computing and wireless networks in the last two decades, the panacea promised by Self-Organising Networks (SON) is not as automated as expected. Most SON features have configuration parameters of their own that also require tuning, and their gains are hard to evaluate without temporarily impacting network performance. Consequently, the search for more reliable techniques to automate network procedures continues.
The past decade has witnessed a rise in Machine Learning (ML) techniques, which capitalise on an abundance of data to reach ever growing performance highs. By virtue of their real-time operating nature, wireless networks have the privilege of generating thousands, if not millions, of data samples per second. They are therefore an ideal source of data for ML algorithms.
The lower layers of the protocol stack (i.e., the Physical Layer (PHY) and the Medium Access Control (MAC) layer) have the most demanding real-time operating points (see fig. 1.1). This has been exploited in works like [1], which applied ML to the PHY. Higher layers tend to operate at a slower pace and therefore produce less data per unit of time. Convergence may hence take longer when training algorithms at Radio Link Control (RLC) or Radio Resource Control (RRC) layers. Nevertheless, some pioneering work in the application of ML to higher layers was also carried out early (see [2]).

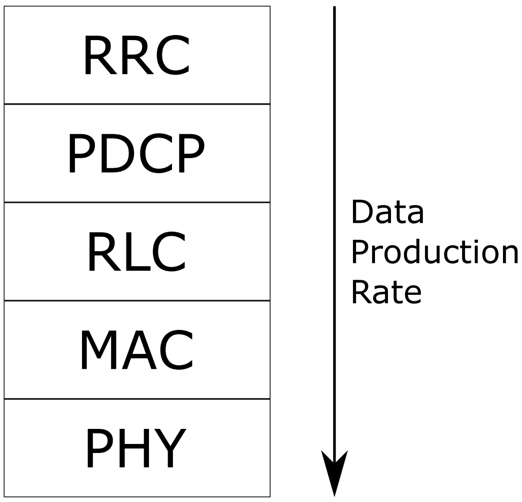


Figure 1.1: Lower layers generate more data samples per unit of time.

1.1. RRM

In modern wireless networks, RRM is the collection of procedures needed to control the radio connections between terminals and their serving base station. This differs from the responsibilities of other layers, which address issues such as access to the wireless medium, encoding, modulation and etc. RRM procedures are usually implemented at Layer 3 and include features such as:

- Broadcasting of system information
- Connection establishment and termination
- Paging
- Connection configuration and reconfiguration
- Security activation
- Mobility
- Channel measurements

Most wireless technologies standardize the messages and steps required to execute each of these procedures. Nevertheless, the standards always allow room for flexibility when it comes to using them. For example, the operators or users may be allowed to choose from a range of values for configuring a feature's parameters. This configuration has traditionally been done manually.

For illustration purposes, consider a basic mobility procedure in 5G networks: the handover. Handovers between two cells begin when a UE sends a Measurement Report about the radio conditions of a neighbour cell. Measurement Reports are triggered when a radio measurement from a neighbour cell crosses a pre-configured threshold. The type of the magnitude (e.g., power or signal quality) and the value of the threshold must be carefully selected by the network operator in order to maximize the handover success rate. The selection of these thresholds for each of the thousands of cells within a network is hence a major responsibility of network operators. Poorly configured handover thresholds will lead to mobility problems affecting millions of connections. It is thus no wonder that autonomous solutions to select these values quickly and optimally are desired by operators.

Mobility Robustness Optimisation (MRO) is a distributed SON feature designed to address this challenge. It enables neighbour cells to exchange information about Radio Link Failures (RLFs) and diagnose specific mobility problems (e.g. handover too early, too late or to the wrong cell). However, MRO relies on heavy signaling between neighbouring eNodeBs (eNBs) and/or gNodeBs (gNBs). In a way, it creates a new challenge while trying to solve another one. For this reason, this project is pursuing the research line described in chapter 3 to address this and other remaining mobility-related challenges.

Medium access is another fundamental procedure in wireless networks. Approaches to this problem can be classified in two categories: Coordinated and un-coordinated access. Traditional random access algorithms, such as ALOHA, have worked well for most networks. However, the advent of Massive Machine Type Communications (mMTC) has uncovered

scalability constraints. Chapter 2 describes these and lays the ML-based research line to address them.
Traffic routing is a well-known network process, with IP routing being the best known protocol. However, traditional routing algorithms are not necessarily applicable in new networks with radically novel topologies and behaviors. Such is the case of Flying Ad-Hoc Networks (FANETs), where novel challenges emerge due to the need for multi-hop connections. Within this context, chapter 4 describes a novel routing protocol, which is to be used as a baseline for future ML-based solutions.
The next RRM function covered by this report is the scheduling of radio resources in 802.11ad networks. Contrary to the random access approach described above, centralized scheduling is one possible algorithm in the coordinated medium access category. In the context of 802.11ad networks, chapter 5 describes the challenge of scheduling contention-free transmissions and how to simulate them. The chapter concludes with the definition of the research plan for the rest of the project.
The management of network resources becomes even more relevant in battery-powered devices. This is specially true in Wireless Sensor Networks (WSNs), where sensors typically have limited processing power. Furthermore, if the network size is large (i.e. made of a high number of sensors), managing the data from all these sources also becomes hard. Chapter 6 discusses these problems in the context of Long Range Wide Area Networks (LoRaWANs) and it presents the main challenges that ML-based solutions will encounter in this domain.

1.2. Objectives

The objective of this Work Package is to investigate whether problems in RRM can be solved using ML techniques. These include supervised, unsupervised and semi-supervised learning algorithms based on datasets collected during the lifetime of the project. In addition, the application of online learning techniques such as Reinforcement Learning (RL) is also investigated for problems that can be formulated as sequential decision making tasks. Due to the dynamic and interactive nature of RL challenges, these will be addressed by means of models and simulators, rather than through static datasets.
RRM problems are commonly addressed in the upper layers of the protocol stack, where numerous procedures and messages have been standardised. This means that, when compared with the lower layers, data is only available in limited amounts. A main challenge of ML-based RRM solutions will thus be the ability to converge and provide gains quickly on limited data.
The RRM challenges addressed in this report are the following:

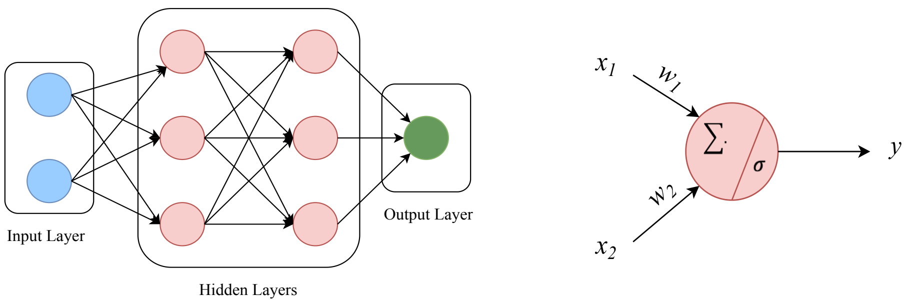
- Random Access in mMTC
- Mobility Management in 5G
- Routing in drone networks
- Scheduling in IEEE 802.11ad
- LoRaWAN network management

1.3. Deep Learning

Deep learning (DL) is a broad category of machine learning (ML) that incorporates Artificial Neural Networks (ANNs) for learning. Recent advances in DL [3] have enabled remarkable progress in the fields of computer vision, natural language processing, speech recognition, and autonomous industry to name a few. Given raw input, DL algorithms use multiple layers of neurons to extract the higher level features from the input data to learn any task (e.g., classification or regression). DL models usually require a huge amount of data samples to learn from, in order to make effective decisions. This allows the model to generalize to inputs that it hasn't observed before.

ANNs are inspired by the information processing systems of a biological brain. A deep neural network is a network of artificial neurons connected via weighted connection among layers. A basic deep network consists of an input layer, an output layer and *hidden layers* between the input and the output layers. Except for the input and output layers, the number of hidden layers and the number of neurons in each hidden layer are configurable depending on the problem. The overall number of layers defines the **depth** of the network, hence the term "deep" in deep learning. The dimensionality of each hidden layer (number of neurons in each layer) determines the **width** of a neural network. A fully connected deep neural network with two inputs, two hidden layers (each having three neurons), and one output is shown in Fig. 1.2a. As mentioned before, each unit in a hidden layer is an artificial neuron, and each neuron receives inputs from other neurons and outputs its own value using some non-linear activation. A simple neuron that acts as a binary classifier is known as *perceptron*.

To generate the desired output, supervised learning (for labeled data) or unsupervised learning (for unlabeled data) algorithms are used.



(a) A neural network

(b) A neuron

Figure 1.2: An example of a fee forward ANN with an input layer, 2 hidden layers and an output layer along with the functionality neuron with two inputs and an output, and activation function

1.3.1. How does a neuron work?

An artificial neuron is a function f of the input $x = (x_1, x_2, \ldots, x_d)$ weighted by edges' weights $w = (w_1, w_2, \ldots, w_d)$, along with bias b and an associated activation function σ. The first computation at the neuron is as follows:

$$z = \sum_i w_i x_i + b, \tag{1.1}$$

or for neuron Fig. 1.2b, $z = w_1 x_1 + w_2 x_2 + b$, where b is a bias parameter. The above calculation is then passed through an activation function σ to produce the output $y = \sigma(z)$. When stacked in multiple layers, neural networks are capable of approximating any continuous function on a compact domain. This universal approximation property lies at the core of their success in the context of deep learning.

1.3.2. Activation and Loss Function

There are several activation functions typically used in the literature; however, the application of a particular activation function only depends on the properties of the problem. For example, a step function could be used if only binary output is desired. A few popular activation functions are the following:

- Linear or Identity function [1]
$$\sigma(x) = x \tag{1.2}$$
- The sigmoid function
$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{1.3}$$
- The Rectified Linear Unit (ReLU) activation function
$$\sigma(x) = \max(0, x) \tag{1.4}$$

A DL algorithm takes input data x and uses this input to estimate the output y that is used for the prediction. Usually, a large amount of data samples (x_i, y_i) are required for a neural network to make predictions for an unseen input. For the learning and training process, weight parameters w and the bias parameters b are tuned iteratively to minimize the difference between the predicted values and the true values for a particular training examples in the data. The difference between the predicted and expected values shows how good the prediction was, and it is determined by a loss function. As an example, mean squared error (MSE) loss function is calculated as

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{1.5}$$

where $\hat{y}$ is the prediction and y is the expected output, and N is the number of training examples in the data. One of the widely used techniques to tune the parameters and minimize the loss is the *backpropagation* (BP) algorithm that usually employs gradient descent or variants such as stochastic gradient descent (SGD) to reduce the loss by updating weights. Since then, many variants of gradient descent have been proposed as optimizers. *RMSProp* and *Adam* optimizers are among the ones that are widely used.

[1]Modern neural networks use non-linear activation functions. Linear activation functions are usually used at the output layer only.

1.3.3. Recent Advances in Deep Learning

The ANN shown in Fig. 1.2a is a fully connected feed forward network. However, these DL networks do not scale well for image classification tasks. Convolutional neural networks (CNNs) [4] revolutionized the image processing and computer vision fields. CNNs allow the extraction of features from images that are usually required relatively little pre-processing of images The convolution operation is used at the layers to extract the features and then pooling layer is used to reduce the dimensionality of the data.
Furthermore, to deal with sequential data such as time series, recurrent neural networks (RNNs) became popular in speech recognition and translations. The popularity of RNNs is due to a special type of neural network known as long short-term memory (LSTM) [5]. LSTMs contain memory to maintain dependencies of hidden layer entries on the previous entries in time.
One of the distinctive deep learning techniques are generative adversarial networks (GANs) [6]. These networks are designed to *generate* plausible new data samples, that are consistent with the training data. A GAN consists of a generative network and a discriminative (adversarial) network, both of which are trained iteratively against each other: the former generates new samples, while the latter learns to distinguish between true samples (from the original training set) and generated samples.
Other advances in deep learning techniques include autoencoders and deep belief networks (DBNs) [7]. The autoencoder based models are used to learn the useful representations within data in order to develop a compressed representation of the input.

1.4. Reinforcement Learning

In reinforcement learning (RL), an agent interacts with an environment, takes actions and receives feedback from the environment in form of reward (or penalty). The agent learns by the *trial and error* method. At every time step, the agent observes the state of the environment, chooses an action from a set of possible actions and receives the reward for choosing that action. An RL environment is usually modelled as a Markov decision process (MDP), with the environment being quantized in states and an agent ought to take actions in order to maximize the total reward in the long-term. One of the challenges that arise in RL is to strike the right balance between *exploration* and *exploitation*: the agent has to *exploit* rewarding actions, but it also has to invest (time) resources to *explore* better strategies.

1.4.1. Markov Decision Process (MDP)

An MDP is a discrete-time stochastic control process. As described in [8], an agent interacts with the environment in a sequence of discrete time steps $t = 0, 1, 2, \dots, T$. MDPs follow the Markov property, which means that the environment's response in state S_{t+1} at time $t + 1$ only depends on the present state S_t, and not on the past states. A typical MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, \mathcal{R}, Q)$ where

- $\mathcal{S}$ is a finite set of possible states
- $\mathcal{A}$ is a finite set of possible actions

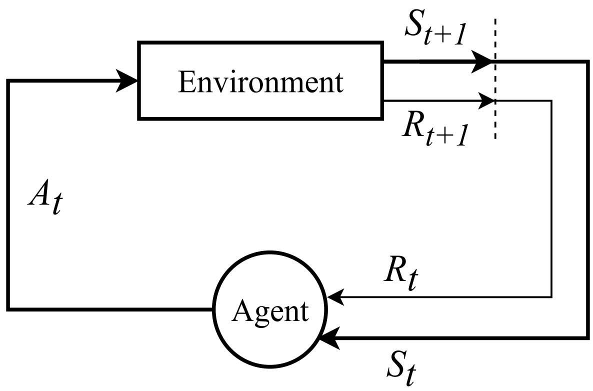


Figure 1.3: A general agent-environment interaction and reward process in the RL

- $\mathcal{P}$ is the transition probability matrix where $\mathcal{P}_{ss'} = P[S_{t+1} = s' \mid S_t = s, A_t = a]$
- $\gamma \in [0, 1]$ is a discount factor, and
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function
- $Q(S_t, A_t)$, a value function which is updated in each iteration, whenever an agent selects an action, and observes reward and move to a new state after time t.

The transition model returns the probability of going from state S_t to the next state S_{t+1} when action A_t is chosen. The reward is calculated when the environment makes a transition from state S_t to S_{t+1}. Agent-environment interaction is depicted in Fig. 1.3. The core problem of MDPs is to find a *policy* π_t that maps observations to a probability distribution over the action space, from which the next action is drawn.
At each time step t, the reward is a real value received by an agent in state s and it is immediate reward. Let's say that agent receives sequence of rewards after time step t denoted as $R_{t+1}, R_{t+2}, \dots$, the return will be the discounted sum of the rewards:

$$U_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots + \gamma^2 R_{t+3} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \tag{1.6}$$

where parameter γ is a *discount rate* or *discount factor*, with range $0 \leq \gamma \leq 1$. The parameter γ makes an agent shortsighted or farsighted. If $\gamma = 0$, only the immediate reward is considered, ignoring the total reward that could be obtained in the long term. We are usually concerned with the expected reward for each state-action pair. The goal of an agent in RL-based problems is to learn a policy that maximizes the long term reward. How good is the expected return in a given state or how good is it for a given action in a given state, is estimated by value functions. There are two type of value functions: *state-value function* and *action-value function* that are used to derive the *bellman equation* that expresses the recursive relationship between the value of the state and the value of its successor states as in [9]. The *bellman equations* for state-value function and action-value function when policy π is being followed, can respectively be written as

$$v_\pi(s) = \sum_a \pi(a|s)\Big(r + \gamma \sum_{s'} \mathcal{P}_{ss'} v_\pi(s')\Big), \tag{1.7}$$

and

$$q_\pi(s,a) = r(s,a) + \gamma \sum_{s'} \mathcal{P}_{ss'} \sum_{a'} \pi(a'|s') q_\pi(s',a') \quad (1.8)$$

The *optimal value functions* v_* and q_* allow us to choose optimal actions without having a prior knowledge of the environment. This is why RL is suitable in dynamic environments that are subject to change randomly. And this is the reason we have chosen RL to formulate our problem of random access in wireless networks because future wireless environments will need dynamic resource allocation.

1.4.2. Q-Learning

Perhaps one of the most important RL algorithms is Q-Learning (QL) developed by Watkins [10]. It follows an *off-policy* learning method, which means estimating the Q-function and updating the policy separately. An agent explores to generate the behavior using *behavior policy*, while it learns a deterministic policy known as *target policy* that is independent of the policy followed for the action selection.
In QL, the action-value function is updated as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - A(S_t, A_t) \big]. \quad (1.9)$$

One of the reasons to use *off-policy* methods such as QL is to deal better with *exploration* and *exploitation* trade-off. This trade-off is one of the challenges in RL. An agent has to *explore* the environment for better actions than already learned ones, and it also has to *exploit* what it already has learned in terms of rewards. One simple but important policy is the ϵ-*greedy policy*, which in state S_t assigns the action

$$A_t \leftarrow \begin{cases} \text{a random action} & \text{with probability } \epsilon \\ \arg\max_a Q(S_t, a) & \text{with probability } 1-\epsilon. \end{cases} \quad (1.10)$$

The exploration–exploitation trade-off can be tuned via the parameter ϵ [9]. At early learning stages, it is advisable to use higher values of ϵ in favor of exploration, and then gradually decrease the ϵ to approach 0.

1.5. Deep Reinforcement Learning

Deep reinforcement learning (DRL) combines the capabilities of deep neural network and RL [11]. This combination has allowed to tackle challenging tasks for which no tractable algorithm was previously known. One landmark in the progress of DRL is when Google's AlphaGo program defeated a human champion in Go [12].
Since classical RL algorithms such as QL rely on a tabular representation of actions and states, they are only tractable for small action–state spaces. To cope with large state spaces, DRL leverages the universal approximation property of neural networks to approximate the value functions in RL (or Q-values in the case of QL). In DRL, an agent learns the optimal policy. If a state is given to a deep neural network, it can approximate the best action for that particular state. This is shown in Fig. 1.4.

A DRL algorithm that uses QL and neural networks is known as deep Q-network (DQN). DQN has become a popular tool to solve a wide range of RL problems and has matched or exceeded human-level performance [12] on some RL problems, such as playing Atari games. The DQN proposed in [12] used convolutional layered neural network (although any kind of layers such as LSTM or dense can be used). The past experiences of the agent are stored in a memory and are randomly sampled to replay these experiences. This technique is known as *experience replay*.

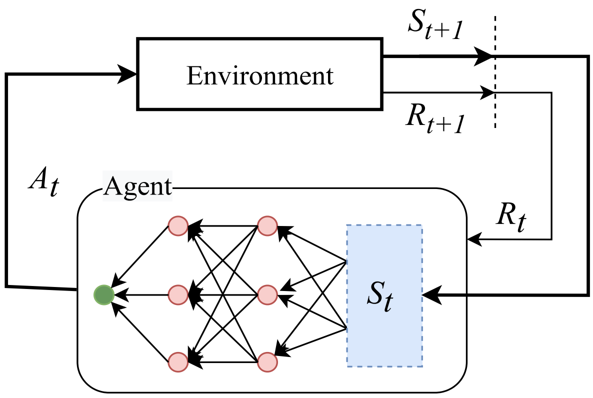


Figure 1.4: Deep Reinforcement Learning Environment

Moreover, a separate target network is used to minimize the loss between actual Q-values and the target Q-values. The target network calculates the target values as follow

$$Y(s, a, r, s') = r + \gamma \max_{a'} Q(s', a'; \theta^{-}), \tag{1.11}$$

and hence, the loss function is calculated as

$$\mathcal{L}(\theta) = \Big(Y(s, a, r, s') - Q(s, a; \theta) \Big)^2, \tag{1.12}$$

where the value $Q(s, a; \theta)$ are the Q-values of actual network and $Q(s', a'; \theta^{-})$ represent the Q-values of the target network. The parameters θ and θ^{-} are updated by stochastic gradient descent by minimizing $\mathcal{L}(\theta)$.
Further improvements on DQN have been made to provide training stability with Double DQN [13], in which action selection and action-value estimation are decoupled and calculated separately by two different DQNs. Furthermore, dueling DQN [14] uses two separate estimators: one for the state value function (scalar) and the other for state-independent advantage function for each action. The advantage function measures how much better an action is compared to other actions in a given state. Both of them are then combined for the final Q-values.

2. Machine Learning for Massive Random Access in mMTC

Recent advances in ML have encouraged the wireless research community to explore how to incorporate ML algorithms in wireless networks. The upcoming fifth generation (5G) network will support dense deployment of the Internet of Things (IoT), and verticals such as autonomous cars, e-health, and virtual reality (VR) applications. In order to meet complex, dynamic and diverse requirements, artificial intelligence (AI) and ML tools have already been regarded as a paradigm shift towards future wireless generations. For dynamic resource allocation in wireless systems, ML algorithms such as DL and RL have shown a great potential in recent years. However, there is still a long way for the ML tools to go in wireless communications to overcome several design and performance challenges [15].
Multiple access in a wireless communication system refers to the process by which multiple users coordinate their access to the medium. The fundamental problems are: how the physical resources should be divided; and which user should access which resource, and when. In legacy communications networks, users go through the process of establishing the connection with access points (AP) or base station (BS) in order to access the medium. Multiple access can be categorized into *coordinated* multiple access and *uncoordinated* multiple access. In coordinated multiple access, the central unit (BS or AP) assigns certain resources prior to transmission. This is known as *grant-based* multiple access. Typical coordinated multiple access schemes are time division multiple access (TDMA), frequency division multiple access (FDMA), code division multiple access (CDMA), orthogonal frequency division multiple access (OFDMA) and non-orthogonal multiple access (NOMA). These techniques have been incorporated in wireless cellular network generations. For instance, in first generation (1G) analog systems, FDMA was a popular access technique. In 2G and 3G, TDMA and CDMA were incorporated, respectively. In 4G, OFDMA is being used as a multiple access technique. For 5G, NOMA, along with OFDMA, is under consideration. In uncoordinated multiple access, users access the medium randomly without coordination by the central unit. The transmission is opportunistic. Users access the medium in a *grant-free* manner, in that devices transmit without prior permission from the central unit. Classical ALOHA [16] and its variants fall into this category.

2.1. Random Access Protocols

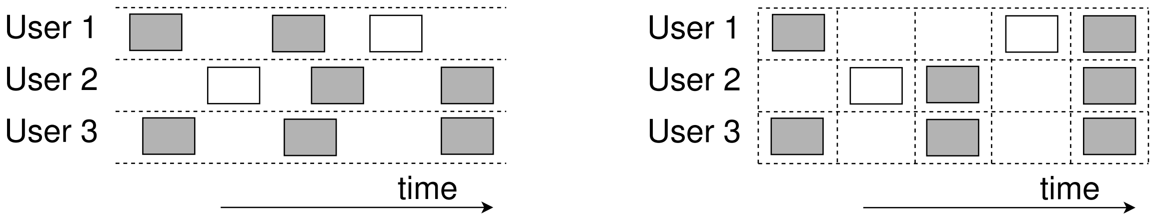
In this section, we will briefly discuss classic RA protocols such as ALOHA [16] and its slotted version called *Slotted* ALOHA [17], and recent advances on RA protocols aiming to increase the throughput of grant-free access employing advanced signal processing techniques.

2.1.1. ALOHA

The first ALOHA protocol was introduced in 1970 as the implementation of random multiple access in a wireless medium. The simultaneous transmissions from users in the ALOHA protocol may result in collisions (either complete or partial), in which case the collided packets cannot be successfully decoded. The packet transmission is only successful if there are no concurrent transmissions. We illustrate the ALOHA transmission protocol in Fig. 2.1a. Usually, the packet arrivals are modelled by a Poisson distribution. Let λ be the average number of packet arrivals within a time interval equal to the packet duration. The throughput of ALOHA in terms of channel utilization, namely the average number of packets recovered by the receiver over a reference time interval as in [17], can be written as

$$G = \lambda e^{-2\lambda} \tag{2.1}$$

The maximum throughput of ALOHA is $\eta = 1/2e = 0.184$, which is approximately 18.4% of successful packets. One shortcoming of ALOHA is that packets are discarded even when there is only a slight overlap between packets.



(a) Pure ALOHA (b) Slotted ALOHA

Figure 2.1: Classic Random Access Protocols

In Slotted ALOHA, packets are forced to fall within a certain time interval at the receiver. The transmissions are time synchronized, but still uncoordinated. The transmissions are divided into discrete time slots and each user can start its transmission only at the beginning of a time slot. Slotted ALOHA transmissions are depicted in Fig. 2.1b. Collisions are reduced in this way and the overall throughput is increased. The throughput of slotted ALOHA case as in [17] then becomes,

$$L = \lambda e^{-\lambda} \tag{2.2}$$

packets per time time slot. The maximum throughput in the slotted case, as can be seen, is $\eta = 1/e$, which means that only 36.8% transmissions are successful. In this way, forcing and synchronizing the users to transmit in time slots, we double the channel utilization in the slotted ALOHA as opposed to the pure ALOHA.
Several other variations of ALOHA such Carrier Sense Multiple Access (CSMA), have been widely adopted multiple access techniques for digital communication. In CSMA, empty channels are sensed before transmitting the packet. The CSMA collision detection (CSMA/CD) scheme became very popular in Ethernet switches, while CSMA collision avoidance (CSMA/CA) is a popular multiple access technique in IEEE 802.11 networks. However, these techniques required some level of signaling for detection and sensing.

2.1.2. Modern Random Access Protocols

Recent advances in signal processing and receiver design techniques led the quest for researchers to develop sophisticated RA algorithms for higher throughput and reliability which can compete with their coordinated counterparts. This paved the way for the birth of *modern RA* techniques and their applications in IoT as well as 5G and beyond. The key idea of modern RA is to embrace the interference constructively at the receiver and use SIC to recover the transmitted packets. This idea was first applied and presented in contention resolution diversity slotted ALOHA (CRDSA) [18] and it was proposed in the context of satellite communications. The idea is to send multiple copies of the same packet in different time

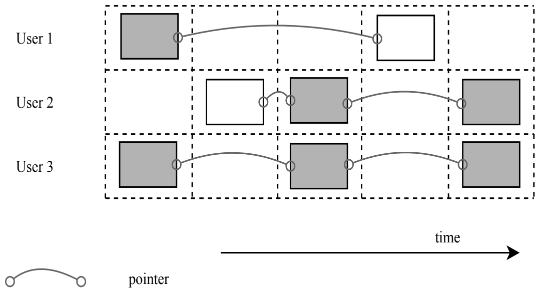


Figure 2.2: Irregular repetition coded slotted ALOHA

slots for diversity. The receiver uses SIC to recover replicas of collided packets from already successfully recovered packets. The recovery of packets happens in an iterative fashion in such way that already recovered packets in the previous time slot propel the new iteration until all the packets are recovered. This phenomenon is shown in Fig.2.2. The real potential of this idea has only been unleashed when PHY layer coding techniques are applied to the transmitted packets in [19] and [20], and the technique is dubbed as coded slotted ALOHA (CSA). The idea presented in these works is to allow active terminals to protect their packets with coding against erasures, in multiple predetermined time slots like in slotted ALOHA. The receiver employs multipacket reception techniques such as SIC to recover packets as discussed above. This technique is extended to use irregular repetition slotted ALOHA (IRSA) [20], [21], where coded packets employ repetition codes. In IRSA, repetition rate of each packet is variable and carefully designed. CSA [19] is a more general framework that allows to use any forward error correction coded for encoding of packets. The *inner code* is used to encode the information for the error correction, while the *outer code* is used to encode the packet for user detection. These techniques have shown a remarkable improvement over classical RA (pure ALOHA and variants), more than double their throughput [22]. In this manuscript, we refer to these coded random access based protocols as *modern RA* protocols.

2.2. Multiple Access Challenges in mMTC

Massive machine-type communication (mMTC) paradigm is one of the key enablers of 5G and beyond communication technologies [23]. We have experienced IoT, but mMTC contains internet of everything (IoE), which basically is network on humans, machines, vehicles and their inter connections. Different verticals will be served and supported in 5G such as e-health, smart industry, precision agriculture, and connected cars etc., as shown in Fig. 2.3. If on one hand, mMTC brings business opportunities, while on the other hand it poses challenges for engineers and researchers. In mMTC, the number of active devices is expected to increase several orders of magnitude (1 million devices per km^2 [24]), scheduled transmissions will suffer a significant signalling overhead, thereby affecting the overall performance of the network. RA techniques suite better for such technologies where transmissions are not coordinated or scheduled. However, RA protocols for mMTC are required to be more efficient and scalable to support massive access and the classical and modern RA protocols are not sufficient to support massive connectivity.

A mMTC system is a network of devices like the IoT network but on a massive scale. The de-

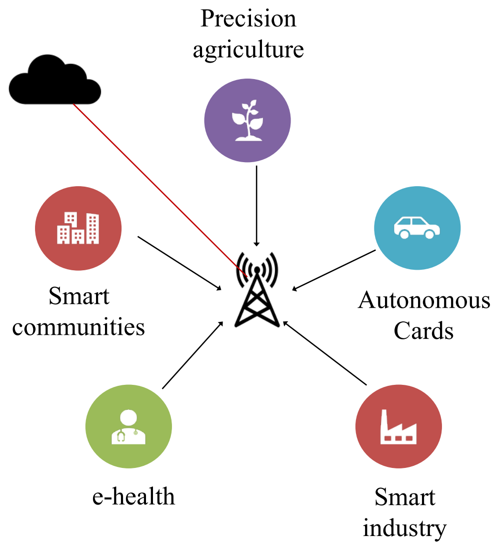


Figure 2.3: A general setting of massive machine-type communication

vices will have different requirements as we mentioned, in terms of quality of service (QoS), latency and energy or spectral efficiency. Moreover, the devices will have large spatial densities and sporadic transmissions with short packet lengths. The peculiarities of mMTC settings makes it different and challenging compared to the traditional cellular networks because of the following reasons:

- When the devices are sporadic and the transmissions of massive number of devices are unpredictable, it makes it difficult to design efficient resource allocation algorithms. Pre-allocation of resources like in conventional cellular communication for human-type communication becomes inefficient in this scenario [25]
- The conventional contention-based access techniques are infeasible when massive numbers of devices will access the medium. It will create a huge signaling overhead and messaging overhead for resource allocation. This signaling overhead will highly affect the battery power of devices. Therefore, is unreasonable to assume that receiver knows perfect channel state information (CSI) and assigning pilots to each device for channel estimation is a difficult task [26].
- The above reasons suggest the use of uncoordinated RA procedure for massive access. However, modern RA protocols' performance is still a question for massive number of devices. mMTC devices are low complexity and low power devices that needs to preserve their energy. Coded RA could pose challenges because of the fact that it will need large number of packet retransmissions when massive IoT devices will access the medium at peak traffic hours, thereby reducing the energy efficiency both at the transmitter and the receiver. [27].

It is evident from the reasons above that there is need to design and develop dynamic random access techniques in order to handle massive access. It is called as *massive random access* in the literature [27], [26]. Moreover, for heterogeneous IoT traffic generated by mMTC devices, specific traffic characteristics and contextual information needs to be taken into account for designing massive random access protocols.

As we have discussed above, ML techniques especially RL and DL have already shown their potential in wireless networks. These techniques make sense in complex networks such as mMTC, which is dynamic due to its nature and where the requirements are diverse and varying. Although state-of-the-art ML-based approaches, that we will present in the next section, have potential in dynamic channel access decisions based on their local observations and without centralized controller, their performance for massive number of devices is yet to be explore. One fundamental question that arises is that if these techniques are scalable to massive network. If not, what are the fundamental drawbacks in terms of algorithm design and complexity that needs to be explored for the scalability as well as their performance in mMTC. Therefore, in this project, our goal is to design ML-based RA techniques that could scale well for mMTC systems, without compromising the performance of the network.

2.3. Machine Learning for Random Access

With the advent of powerful processing and huge storage capabilities of hardware, machine learning (ML) tools have become popular to solve challenging tasks and optimal decision making in dynamic environment. Since the number of configuration parameters will be large, especially in mMTC networks, conventional methods and well-known mathematical tools in wireless communications may not scale well to provide us efficient solutions. The dynamics of future networks as for automated processes where human intervention is least possible or where it is not practical.

One promising direction to achieve this goal is through ML tools. ML has been known for a while to make intelligent decisions in complex environments. ML algorithms are used to learn about the environment to make optimal decisions using already collected observations (data) such as DL, or by observing and interacting with the environment like trial and error such as RL. Future networks, e.g., 6G and beyond are expected to integrate ML tools throughout the network, and ML tools are particularly useful for dynamic channel allocation in networks where the number of users is large and the physical resources are limited [28], [29]. In the next section, we will provide an overview of existing RA techniques that use ML for spectrum access. We will start the state-of-the-art with mentioning few modern RA protocols that have been recently proposed for massive connectivity, followed then by ML-based techniques in the literature.

2.3.1. State-of-the-art

Modern RA schemes [18]– [20] started a new direction of research for uncoordinated multiple access issues in future wireless networks. Interestingly, besides more than double throughput than slotted ALOHA, modern RA protocols can serve a larger number of users than ALOHA-based techniques and therefore, they have attracted much attention for IoT scenarios in future wireless generations [30]. As reported in [27], modern RA techniques, such as coded RA, could pose challenges in terms of complexity when the number of users grows large. For this purpose, there has been a number of works on efficient code design for coded RA, e.g., in [31], a scheme is proposed for asynchronous RA where authors showed the performance of their code design for RA with and without the abstraction of PHY layer.

In [32], authors proposed a low-complexity code design for K-*fold* slotted ALOHA scheme for multiuser detection in mMTC. Like coded slotted ALOHA, an outer code is designed for user detection in such a way that if K or less number of users transmit in a certain slot at the same time, the receiver can decode the corresponding packets successfully from their linear combination, whereas no packet can be decoded when more than K users transmit. It employs the concept of compute-and-forward based grant-free access. These techniques have their advantages in terms of throughput; however, they still incur computational complexity when the number of users is massive. Furthermore, how to design optimal solutions for channel assignment in such complex systems is still unclear. This motivates us to explore and leverage ML algorithms for channel access problem.

RL has become a popular choice for resource management in wireless networks. Its ability to learn through interaction with dynamic environments and to adapt accordingly has made it an effective tool to address medium access problems (both with and without coordination). Since for ALOHA-like random access, users are not scheduled and there is no communication between users, the state of the environment is not fully observable, as each user only knows its own state. Such problems are usually modelled as *partially observable* Markov decision process (POMDPs), in contrast to classic MDPs, where the environment is fully observable. In wireless networks, nodes such as BSs or APs are represented by agents that learn to schedule resources, while user devices may be represented by agents that compete for the channel access. Several channel access frameworks have been proposed in the literature in which agents are able to learn and make decisions without requiring any prior knowledge, in an independent fashion, using local observations (e.g., ACK signals). First applications of RL to multiple access in wireless communication date back to 2010, when authors in [33] proposed a multi-agent Q-learning based ALOHA-like technique for cognitive radios (CR), whereas RL-based single-user and multiuser cases are compared in [34] and the effects of different RL hyperparameters on learning are analyzed. Although QL works well for small state and action spaces as in [33], it becomes problematic in terms of complexity and convergence as the state space grows. In order to overcome this problem, DQN [35] has become a very popular technique that combines QL with neural networks for Q-value approximation and to find the optimal policy. Using DQN for spectrum access in [36], a dynamic multiple channel access technique is proposed in which the user (agent) observes good or bad states of the channel. The channel is considered *good* if the transmission is successful and a reward +1 is obtained, while the reward -1 is obtained if the transmission is failure and the channel is considered *bad*. The past actions and channel-state observations are used to train the DQN in order to find a policy that selects the best channel. However, this work only considers the single-user case, which is not practical in the context of wireless networks. For the multiuser case, on the other hand, the problem becomes fundamentally different.

A deep *actor-critic*(A2C) algorithm is proposed for multi channel access in [37], in which both single-user and multiuser cases are presented. The A2C algorithm uses two deep networks: an Actor and a Critic network. The Actor determines the action when the state is given, while Critic evaluates the value of the state. In [38], a multiagent deep learning approach has been considered for resource allocation in LTE networks. Agents are LTE BSs that aim at predicting the future states of the system to proactively assign resources. In this work, authors have used LSTM with the REINFORCE algorithm [39] to predict the future states. The REINFORCE algorithm is a policy gradient method, unlike QL, it evaluates the policy instead of Q-values. For the multiagent case where each user device is acting as an agent, a distributed random spectrum access technique has been presented in [40]. Offline centralized training is performed for each agent to learn independently through their local observations. In this paper, authors have presented two scenarios: a) *competitive approach* in which each user aims at maximizing its own network utility, e.g., achievable rate and b) *cooperative approach* in which each user maximizes one and the same system wide global network utility. Dueling DQN with LSTM is used to model to track the history of previous states. In dueling DQN, before evaluating the Q-values, the network is divided into two layer streams: an advantage steam that evaluates how good the selected action is; and a state-value stream that evaluates state-value function. Both steams are combined at the end for the resulting Q-values. Furthermore, a recent work [41], studies joint channel access and channel aggregation in the CRs taking channel correlation into account. Each user only observes a segment of frequency band in the multiple channels, and the vacant channels are aggregated for transmission in that segment.

In [42], authors proposed a Q-learning-based decentralized algorithm under the POMDP framework to optimize the degree distribution (number of packet replicas) of IRSA in IoT systems. In [43], LSTM-based NOMA technique has been proposed for random access. The data in this work is generated through simulations for training the network. Furthermore, another recent work [44], employs Q-learning for channel allocation in NOMA-based RA for MTC network with the consideration path loss and fading effects. Multi-agent deep deterministic policy gradient (MADDPD) method is used in [45] to exploit spatial and temporal correlation of devices and events in IoT network for random access. However, this work does not consider the affect of these correlations on network utilization. Moreover, they have evaluated the performance for very few number of users; therefore, it would be interesting to see if MADDPG performs well for large number of users. In [46], a DQN-based joint power control and channel access scheme has been proposed, while DRL and graph convolutional networks are used in [47] for channel allocation in IEEE 802.11 wireless local networks.

An interesting area where ML solutions have been employed to handle massive access management is to consider heterogeneous network requirements of devices such as latency and QoS. Coexistence of TDMA and ALOHA has been considered and optimized using DRL for proportional fairness in [48]. In [49], DRL is used for ultra reliable and low latency communication (URLLC) traffic where BS were used as agents to allocate physical resources to users. In this work, authors used transfer learning and cooperative learning techniques to enhance overall network's performance.

Moreover, predictive resource allocation by using a technique called as *fast uplink grant* is emerging as a promising solution for resource management in mMTC. Uplink resources can proactively be allocated to users by BS. This has been realized by using ML techniques for prediction [50] and assigning resources prior to transmission without RA channels being scheduled.

3. Mobility Management in 5G

3.1. Background

The telecommunication and networking systems have evolved tremendously over the last ten years to satisfy the increasing demand for low latency and high reliability. The most prominent design in the next-generation telecommunication networks is the Heterogeneous Network (HetNet) architecture [51]. HetNet is a multi-tier cellular wireless network that provides ubiquitous coverage to indoor and outdoor user equipments (UEs). In HetNets, a massive number of small-cells are deployed underlying some macrocells that meet the requirements of next-generation of cellular service categories, such as the Ultra-reliable Low-latency Communications (URLLC), Internet of Things (IoT), and extreme Mobile BroadBand (eMBB). On the other hand, Machine Learning (ML) is a significant technology that can be used to improve user mobility prediction and handover (HO) decisions, fostering seamless services' execution in HetNets.

3.2. Mobility Management

One of the main goals of mobile cellular systems is to provide UE a seamless wireless connection from one base station (a serving cell) to another base station (a target cell). More specifically, the UEs can always attach to the telecommunication network, and when attached, maintain the connection, such that the data transfer will not be delayed or lost.
The purpose of mobility management for the connected mode UEs is to maintain the wireless connections between UEs and telecommunication networks. Mobility management consists of two procedures which are cell selection and HO procedures. But only the HO procedure governs the active data traffics. Moreover, the HO implementation can be separated into two different categories. The first category is the soft HO (SHO). SHO or connect-before-break HO executes the HO procedure where the incoming radio links are added to an UE while the existing radio links are still connected to the UE. On the other hand, the second category is the hard HO (HHO). HHO or break-before-connect HO executes the HO procedure where the incoming radio links are added to an UE after the existing radio links are disconnected from the UE. Both SHO and HHO can be triggered by UEs or base stations.
From Fig. 1, in general, UEs will measure the signal strength or signal quality over the radio channels from all the available base stations in around of the UEs. If a predefined condition is fulfilled in the measurements result, the UEs send a measurement report (MR) to the serving base station (S-BS). Once the MR is correctly received at the S-BS, the HO evaluation starts between the target base station (T-BS) and the S-BS. During the HO evaluation, a HO request is sent to the T-BS. Upon successful admission, a HO command is transmitted from the S-BS to the UE. Once the HO command is successfully received, the UE executes the HO procedures. At that time, the UE connects to the target cell. If performing HHO, the UE will disconnect it from the serving cell as well. But SHO will still keep the connection between the UE and the serving cell. A HO confirmation message is transmitted by the UE once it can receive broadcast information from the T-BS. Before the HO completion, the user data gateway (UDG) switeches the DL data path from S-BS to

the T-BS. And therefore the T-BS starts receiving packets from the UDG. Finally, the T-BS transmits a HO complete message to the S-BS to inform the success of the HO. After that, the S-BS releases any allocated radio resources to that UE.

There are many HO algorithms defined by 3GPP [52]. The HO algorithms are designed and optimized based on the performance index such as: minimizing the total number of HOs, reducing the ping-pong effect, and minimizing the HO decision time. Since the network architecture becomes more complex than before and more performance index are claimed, optimizing the HO procedure to achieve the performance goal is considered as an important issue in the telecommunication network.

Many state-of-the-art approaches improve the HO procedure with different HO algorithms, which can be categorized into four groups based on their mechanisms and parameters. The first category comprises the mobility pattern-based algorithms, e.g., [53], which use a historical information table for the network and the UEs to predict target cells. In the next category are the algorithms optimized by the movement direction and location. The location parameters that depend on Global Position System (GPS) were used in [54] to determine the positions of neighbor eNBs and UE for predicting the new UE position, such that UEs select the target cell based on its direction. In the third are self-optimizing algorithms, e.g., [55] [56], which decide on the HO through tuning the HO parameter values or adding new parameters to improve HO performance. The last category consists of multi-hop cellular algorithms, e.g., [57] [58], which use an advanced HO decision controlled in a relay node or in a base station.

3.3. Problem Statement

Besides the enormous advantages of HetNets, the frequent Handover (HO) due to the deployment of the ultra-dense network is one of the most critical challenges in the development of HetNets. That will lead to increasing the ping-pong effect and Radio Link Failure (RLF). As a result, the system performance of a HetNet degrades severely. In order to improve the performance of the HetNet system, a state-of-the-art decision-making ML-based algorithm is required to establish the HO accurately and efficiently. Moreover, the inclusion of ML can be a driving source to minimize the increasing effects of frequent HOs in the ultra-dense HetNet system of future generation wireless networks.

3.4. Research Method

The study will be implemented by a discrete-time simulator e.g. NS-3. From the simulator, the theoretical simulation will be conducted via three major components. The first is HetNets environment which implements a macro-cell and micro-cell co-existing network with the Rayleigh channel model [59]. The next is the mobility model. The UEs moving behaviors e.g. Random Waypoint model, will be implemented in the simulation. The third is a ML framework. We aim to use the Reinforcement Learning (RL) framework [60] in the proposed algorithm for our simulation. Then simulation results will be analyzed and compared with the state-of-the-art HO approach. In addition, we will also seek the other ML frameworks for performance comparison.

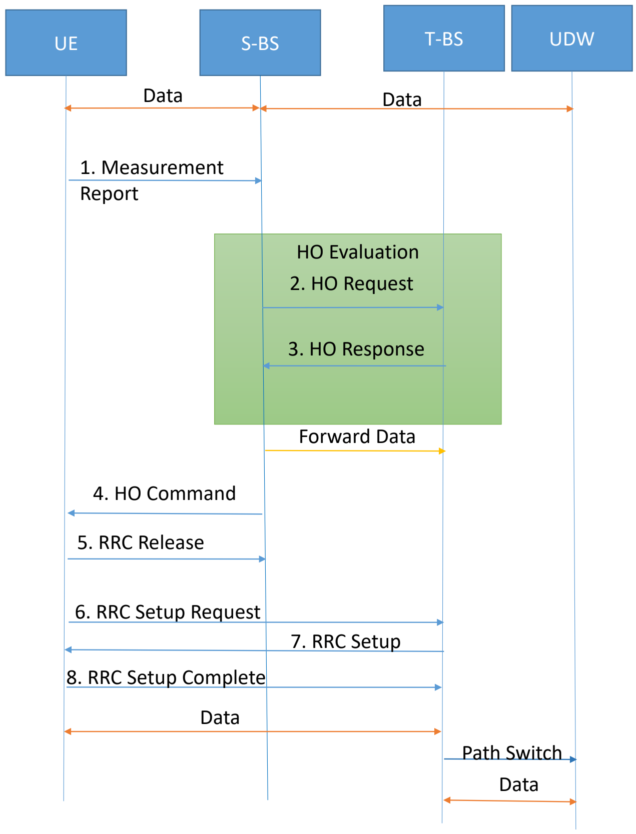


Fig. 1 Handover Procedure

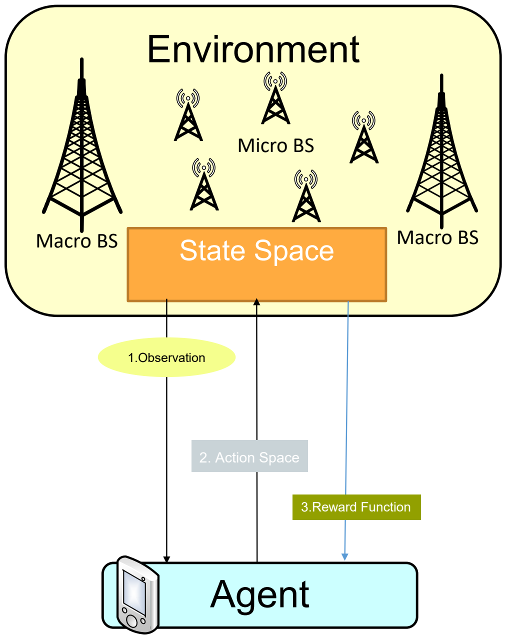


Fig.2 RL Simulation Environment

3.5. Expected Research Result

An efficient handover (HO) algorithm based on ML will be developed. The HO algorithm will also generate the HO parameters for HO decision. The developed algorithm will be investigated and validated by using a simulator tool. The proposed algorithms will be investigated and compared to other state of the art algorithms in HetNet with the consideration of dual connectivity.

4. Routing in Drone Networks

4.1. Introduction

Drones or Unmanned Aerial Vehicles (UAVs) are used in a wide range of applications, from tracking and monitoring animals in remote areas [61] to military applications [62]. 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 [63–65]. 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.

In this introductory work, we designed a Stochastic Multipath UAV Routing for FANETs (SMURF) protocol. It is a centralized multipath routing protocol for FANETs, which exploits the tracking information available at the control center to estimate the reliability of routes and select the set of routes that guarantee the overall highest reliability. The routing tables can be computed by the control center and propagated to UAVs by using the Software Defined Network (SDN) [66] paradigm.

4.2. SMURF Routing Protocol

A FANET is modelled 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 R (which depends on the communication technology used): $E(t) = \{e_{ij}(t) : d_{ij}(t) \leq R\}$. 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 Density Function (PDF) $p(\hat{\mathbf{x}}_i = \mathbf{x})$ instead. We can now define the link existence probability $P(e_{ij})$ as:

$$P(e_{ij}) = P_R(d_{ij} \leq R) = \int_{\mathcal{B}_R(0)} p(\mathbf{x}_i - \mathbf{x}_j = \mathbf{x}) d\mathbf{x}, \tag{4.1}$$

where $\mathcal{B}_R(\mathbf{x})$ is the sphere with radius R 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:

$$\mathbf{e}^* = \arg\max_{\mathbf{e} \in \mathcal{E}_{sd}} P(\mathbf{e}), \tag{4.2}$$

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. 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 $\bar{\mathbf{e}}^*$):

$$\mathbf{b}(\mathbf{e}^*) = \arg\max_{\mathbf{b} \in \mathcal{E}_{sd} | \bar{\mathbf{e}}^*} P(\mathbf{b} | \bar{\mathbf{e}}^*). \tag{4.3}$$

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 SMURF algorithm to calculate the primary and backup routes.

4.2.1. Link existence probability

We 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 [65]. 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:

$$p_i(\mathbf{x}) = \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}_i|}} e^{\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_i)\right)}. \tag{4.4}$$

Hence, the link existence probability as expressed in (4.1) is given by:

$$P(e_{ij}) = \int_{\mathcal{B}_R(0)} \frac{e^{\left(-\frac{1}{2}(\mathbf{x}-(\boldsymbol{\mu}_{i-j}))^T \boldsymbol{\Sigma}_{i-j}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{i-j})\right)}}{2\pi\sqrt{|\boldsymbol{\Sigma}_{i-j}|}} d\mathbf{x}, \tag{4.5}$$

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:

$$P(e_{ij}, e_{jk}) = \int_{\mathbb{R}^3} P(e_{ij}|\mathbf{x}_j = \mathbf{x}) P(e_{jk}|\mathbf{x}_j = \mathbf{x}) p_j(\mathbf{x}) d\mathbf{x}, \tag{4.6}$$

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

$$P(e_{ij}|\mathbf{x}_j = \mathbf{x}) = \int_{\mathcal{B}_R(\mathbf{x})} p_i(\mathbf{y}) d\mathbf{y}. \tag{4.7}$$

All the computations above are so reduced to the calculation of multivariate Gaussian integrals, which can be performed efficiently with well-known numerical methods [67, 68]. 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.

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:

$$P(\mathbf{e}) = P(e_{12})P(e_{23}|e_{12}) \ldots P(e_{n-1,n}|e_{n-2,n-1}). \tag{4.8}$$

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. 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:

$$W(e_{jk}|e_{ij}) = -\log_{10}(P(e_{jk}|e_{ij})). \tag{4.9}$$

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

4.2.2. Backup Routes Calculation

By definition, the primary route is the one with the highest probability of existence, but it might still fail in a dynamic scenario. For this reason, we consider a set of backup routes, which can be selected in case the primary one fails. This can significantly increase the reliability of the transmission if the UAV swarm is dense enough, as there will be multiple viable routes to the destination. In order to calculate the optimal backup, we consider single-link failures and define the conditional path existence probability, given the link is down, as follows:

$$\mathbf{b}_i(\mathbf{e}^*) = \underset{\mathbf{b}\in\mathcal{E}_{sd}|\bar{\mathbf{e}}^*}{\arg\max}\, P(\mathbf{b}|\bar{e}_i^*). \tag{4.10}$$

If the i-th link in the primary route does not exist, we can compute the conditional joint position PDF of nodes i and $i+1$, as:

$$p((\mathbf{x}_i, \mathbf{x}_{i+1}) = (\mathbf{x}, \mathbf{y})|\bar{e}_{i,i+1}) = \begin{cases} \frac{p_i(\mathbf{x})p_{i+1}(\mathbf{y})}{1-P(e_{i,i+1})}, & \mathbf{y} \in \mathcal{B}_R(\mathbf{x}); \\ 0, & \mathbf{y} \notin \mathcal{B}_R(\mathbf{x}). \end{cases} \tag{4.11}$$

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:

$$\tilde{\mathbf{b}}(\mathbf{e}^*) = \underset{\mathbf{b}_1,\ldots,\mathbf{b}_{N(\mathbf{e}^*)-1}}{\arg\max}\, P(\mathbf{b}|\bar{e}_i^*)(1 - P(e_i)), \tag{4.12}$$

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.

4.2.3. Route information propagation and data plane

In our model, routing calculations are performed by the central control station, which collects tracking information using LoRaWAN [65] 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 [69]: 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 [70]. 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.

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.

4.3. Further Steps

In the introductory work, we proposed a statistical analysis of a FANET with tracking information, and derive the existence probability for both single links and entire routes. We then design SMURF, a multipath routing protocol that computes a primary route (i.e., the route with the highest existence probability), and a series of backups that allow the transmission to succeed even if a link in the primary route is broken. Using this work, a dataset for routes in different FANET scenarios can be generated which can then further be applied for further work.

The work, hence, is currently being extended by considering the problem of routing over a longer time horizon, choosing backups that can maintain the connection over time. To exploit this correlations over longer time horizons, the research is being focused to employ machine learning techniques with the possibility of learning how to route in a map with several blockers and realistic propagation.

5. Scheduling in IEEE 802.11ad

5.1. Introduction

Wi-Fi is a well-known technology that nowadays can be found everywhere, from the houses, offices to the public institutions, and even public transportation. The IEEE 802.11ad standard [71] introduced the Millimeter Wave (mmW) band communications into Wi-Fi, as an answer to increasing demands for higher data-rates. The standard enables multi-gigabit short-range communications over the 60 Ghz band and offers data-rates up to 6.75 Gbps. Its successor, IEEE 802.11ay, will be standardized by the end of 2020 [72] and can provide even higher data-rates by employing new technologies such as Multi-User Multiple Input, Multiple Output (MU-MIMO), channel bonding, and using higher-order modulations. This high data-rate can pave the ways for applications such as wireless office docking, 8K Ultra High Definition video transfer, wireless Augmented Reality (AR) and Virtual Reality (VR), mobile front-hauling and offloading, etc. [73].
On the downside, transmission over the high mmW frequencies suffers from the increased propagation loss, and deeper diffraction shadows, as well as the higher penetration and reflection losses that make communication more difficult and less stable. To alleviate this problem, the Stations (STAs) can steer their beams toward the specific direction and transmits over the directional links. Transmission directionality will provide a very high potential for spatial reuse, where multiple concurrent transmissions can coincide in the same area as long as they do not interfere with each other. It also creates deafness issues for Stations (STAs) and worsen the hidden node problems and makes the mobility more complex to handle. The standard introduced a hybrid Medium Access Control (MAC), where the Stations (STAs) can transmit during the reserved contention-free access periods (in a Time Division Multiple Access (TDMA) manner) or over the contention-based access periods (using the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism). The proposed MAC has brought enough flexibilities to the system to support the coexistence traffics with very different Quality of Service (QoS) requirements.
In this chapter, we introduce some of the resource allocation and scheduling problems and challenges in the IEEE 802.11ad scenarios. We also discuss some pre-existing works and propose some research directions. The rest of the chapter is organized as follows. In 5.2, describes the main characteristics of IEEE 802.11ad. The scheduling and resource allocation in IEEE 802.11ad and the literature review present in 5.3. The 5.4 provides some research plan, and finally 5.5 concludes the chapter.

5.2. IEEE 802.11ad Overview

In this section, we briefly describe the hybrid-MAC introduced by the standard [71] and refer the interested readers to [74] for more details.
In IEEE 802.11ad, to simplify beam management procedure, all the Stations (STAs), including Personal Basic Service Set (PBSS) Central Point (PCP)/Access Point (AP) (PCP/AP), divided the surrounding area into sectors shown in fig. 5.1.
Also, the time is divided into Beacon Intervals (BIs) of about 100 ms, as shown in fig. 5.2.

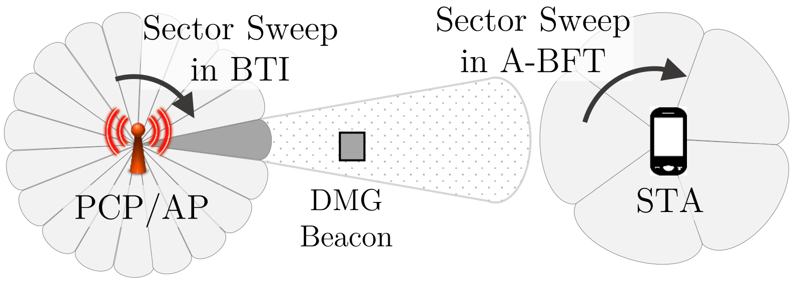


Figure 5.1: Graphical representation of sector structure in IEEE 802.11ad.

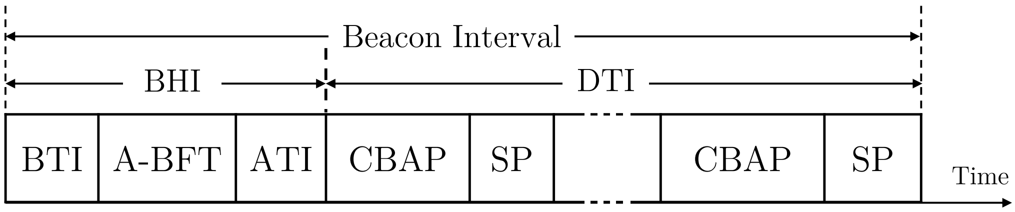


Figure 5.2: Representation of a Beacon Interval.

Each BI is further divided into Beacon Header Interval (BHI) and Data Transmission Interval (DTI) that briefly described in the next sections.

5.2.1. Beacon Header Interval

The BHI is used for network configuration and management operations such as network announcement, beamforming training, and scheduling and can be consist of three optional subintervals: Beacon Transmission Interval (BTI), Association-BeamForming Training (A-BFT), and Announcement Transmission Interval (ATI).
The BTI is used by PCP/AP to send Directional Multi-Gigabit (DMG) Beacon frames over different sectors to announce the network, provide information about the BI structure, and initiate the beamforming training procedure with other Stations (STAs). Next, the Stations (STAs) can complete the beamforming training procedure in A-BFT, by sending Sector Sweep (SSW) frames over their different sectors. Finally, further network management can be done during ATI.

5.2.2. Data Transmission Interval

The DTI is mostly used for the data transmission, but it also can be used for beam refinement procedure to improve the quality of directional links. The DTI is consist of Contention-Based Access Period (CBAP) and Service Period (SP), which can appear in any combination and order.
Transmission in CBAP follows the rules of Enhanced Distributed Channel Access (EDCA) in which Stations (STAs) compete to each other before being able to transmit their data.
Conversely, Service Periods (SPs) are the scheduled access periods that dedicated to the pairs of Stations (STAs) to transmit over a contention-free channel to guarantee QoS.

5.3. Scheduling in IEEE 802.11ad

The IEEE 802.11ad provides a very flexible radio resource allocation and scheduling mechanisms. Here we describe one of the possible scenarios in a simple form, presented in fig. 5.3.

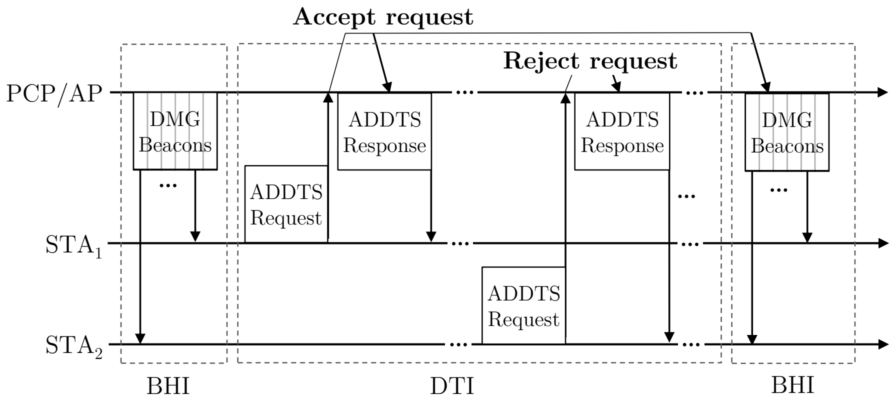


Figure 5.3: Representation of ADDTS scheduling in IEEE 802.11ad.

First, during the BHI, scheduling information for the next BI is announced by the DMG Beacon sent by PCP/AP Later, in DTI, the Stations (STAs) can set up an allocation by sending Add Traffic Stream (ADDTS) Request to the PCP/AP. An ADDTS Request consist of a DMG Traffic Specification (TSPEC) element that defines the STA's desired allocation by parameters such as allocation period, and the minimum and maximum allocation duration. The PCP/AP either accept or reject the request based on its admission policy and notifies the STA by sending an addts Response. The accepted requests are scheduled and announced within the next BHI. By knowing the scheduling information, the transmitter and receiver can steer their antenna toward the right direction before starting the transmission, while the other Stations (STAs) can switch to power-saving mode. The resource allocation problem is clearly, a trade-off between QoS traffic, which needs resources to fulfill the minimum requirements imposed by the application, and elastic traffic, which still needs resources even though with less stringent requirements.
A mathematical model for preliminary allocation of SP for Variable Bit Rate (VBR) flows derived in [75], to determine TSPEC parameters to minimizing the amount of allocated time while meeting applications QoS requirements. Other works consider different aspects of the DTI. For example, [76] derives the theoretical maximum throughput for Contention-Based Access Periods (CBAPs) when two-level MAC frame aggregation is used. Beamforming is also considered in [77], which proposes a joint optimization of beam width selection and scheduling to maximize the effective network throughput, while other works, though not specifically concerning IEEE 802.11ad, deal with transmission scheduling for mmW communications [78].

5.4. Future Research

In this section, we highlight some possible research directions. First, we describe the available tools for the research, and then we propose a possible research plan.

5.4.1. Available Research Tools

The commercial devices that support the IEEE 802.11ad standard are currently available. However, in most of the case, they are costly, and they do not provide full access to low-level functionalities to implement and compare different resource allocation methods. Instead, simulators give a more flexible, timely, cost-effective, and arguably less realistic approach to analyzing such a system.
The Network Simulator 3 (ns-3) module for IEEE 802.11ad standard [79], is publicly available, and in its last release, it supports more accurate and realistic channel models based on ray-tracing [80].
Resource allocation and scheduling algorithms, historically, are based on heuristics. However, in recent years, following the concepts of *self-driving networks* [81], tremendous research effort aimed to design Machine Learning (ML) based approaches, where the algorithms learn from real on-line data instead of manually-designed methods. OpenAI Gym is a well-known Reinforcement Learning (RL) toolkit that has already been integrated by most ML frameworks. Given their potential in many fields of communications, OpenAI Gym has also been integrated into ns-3 [82] with the name of *ns3-gym*.

5.4.2. Research Plan

From one perspective, the resource allocation in IEEE 802.11ad can be divided into two sub-problems: In the STA, translating the information given by the application to the desired allocation, described by DMG TSPEC elements. Besides, in the PCP/AP side, efficiently scheduling the DTI by considering the availability of resources and different networking metrics (e.g., delay, jitter, throughput, fairness).
Regarding the former, the applications may have very dynamic traffic patterns in terms of inter-packet arrival time (e.g., frame-rate drop in video applications) and the packet size (e.g., when compression is considered). Meantime, the transmission conditions may change very rapidly due to environmental changes, blockage, and mobility. In this situation, the RL agent can learn the patterns and update TSPEC element in a proactive manner. On the other side, the PCP/AP should apply effective scheduling methods, taking into account different factors such as resource availability, network metrics, and possible evaluation of the Modulation and Coding Schemes (MCSs) since the packet transmission time mostly depends on it. An RL agent can jointly adapt the MCS and perform scheduling to optimize the network performance by observing the evolution of both channel statistics and network traffic.
A significant development effort needs to be taken in creating a proper simulation environment, with particular attention to the computational complexity where, e.g., 10 s of simulation for high data-rate traffic may currently take around an hour in ns-3. This makes the design, evaluation, and optimization of scheduling protocol a lengthy process, and even impossible when it comes to RL that needs to iterate over thousands of episodes to learn the dynamics of the environment. One approach to solve the slow simulation problem is to pre-train the agent on a fast and simple simulator to learn some basic and general rules and later train it on a realistic simulator. To further decrease the simulation time, it is also possible to create a database of simulation results and use them to passively train the agents [83] before fine-tuning their behavior on the real simulators. Transfer learning can also be considered as another approach to speed up the convergence of the RL algorithm to effective policies in different scenarios.

5.5. Conclusions

In this chapter, we briefly described the main characteristics of IEEE 802.11ad by focusing on the MAC layer and introduced some of the possible scheduling mechanisms provided by the standard. We also discussed some resource allocation and scheduling challenges in the standard and gave possible research directions to replaced the complicated heuristics with the ML and RL based methods.

6. Lorawan Network Management

Recently, wireless sensor networks (WSNs) have become one of the most practical research paradigms, providing services in numerous fields ranging from construction to manufacturing, monitoring to smart solutions and from normal cities to the mountain areas. WSN is a wireless network that consists of base stations and numbers of nodes (wireless sensors).These networks are used to monitor physical or environmental conditions like sound, pressure, temperature and efficiently transfer data through the network to a main location. This coherent interaction among diverse objects such as sensors, nodes, gateways and internet, makes WSN a revolutionary prototype that can provide ubiquitous and real time applications. LoRa (long range) is one of the most promising wide area communication technologies being used frequently in IoT.

Lorawan with limited battery, Central Processing Units (CPUs), and memory resources is a widely implemented technology for early warning detection systems. The main advantage of Lorawan is their ability to be deployed in areas that are difficult to access by humans. In such areas, regular maintenance may be impossible; therefore, Lorawan devices must utilize their limited resources to operate for as long as possible, but longer operations require maintenance.

A plethora of significant applications can be provided by WSN such as in construction WSN can collect the data needed by the civil engineers to centrally monitor their assets, analyze risks, and preplan their responses. Thus, this provides efficient geotechnical and structural monitoring. Similarly, remote monitoring of soil parameters such as humidity, temperature, etc. in real time to improve crop production, environmental monitoring, monitoring of green energy systems, water monitoring, waste management, road & traffic management, disaster mitigation, etc [84]. Other than this, Semeteh has made a system of sensors for remote monitoring of patients in clinics. The benefit is that the patients do not need to wear any sensors. Another great example of WSN application is the traffic monitoring solutions. Apart from these, there exist a wide range of applications, however, there are limitations in the resources that must be taken care of. These limitations are mostly due to the constraints of an individual node or the gateway such as minimal power and processing capacities.

Resource management in Lorawan has been a centre of interest for researchers in both academia and industry. The need to develop more robust methods has always triggered new challenges in the field. These challenges include the heterogeneity of the devices, size, cost and battery usage. A significant amount of work has been done at the research level but surprisingly, a little of them is used in a real WSN. At the system level, more work has been put into practical level because of its promising applications. System level management consists of data administration, routing solutions, energy efficient solutions, devices' diversity management, memory and network management etc. A latest direction in resource management is the inclusion of clouds, fogs and the servers from which WSN can borrow the resources whenever it requires. Thus, despite the limitations, all this work has made it possible to use the data for correlation and comparison for various applications.

6.1. Resources of Lorawan

Lorawan is an advanced form of wireless networking where several sensors are attached with the network to collect a diverse variety of data. However, the network nodes are not abundant in resources like memory, power and processing capability. This is the tradeoff to the phenomenal characteristics of WSN. The nodes are mostly placed in challenging environment so they are supposed to be small, simple to install and should be capable to ensure harsh environments. Hence, it becomes impossible to instill high memory units, very large batteries and highly capable processing unit in the hardware.
From the memory perspective, Nowadays, some latest WSNs are using clouds and fog technology. Similarly, in such WSNs, clouds can be used for performing the tasks that require high end processing such as data analysing and graph visualization. In addition to this, some WSNs are also using solar rechargeable batteries. This solves the problem of battery related issues but not feasible in all applications for instance, the nodes do not get the sunlight when used in tunnel monitoring or underground scenarios/applications. Therefore, it is inevitable to manage the limited resources efficiently.

6.1.1. Constraints of Lorawan

- Storage capacity is only a few hundred kilobytes.
- Limited processing power upto 8MHz
- Works in short communication range which consumes a lot of energy
- Requires minimal energy constraint protocols
- Batteries have limited life time
- Passive devices provide little energy

6.2. Non Machine Learning based solutions

Resource management in WSNs has been the focus of interest for many years now because of the huge applications it provides [85]. Studies have been conducted to carefully manage the resources at the operating system level [86]. This provides the solution for the limited resources in WSNs and addresses the scalability issues by leveraging the cloud storage and computation services. Researchers have also suggested adding cloud services into the wireless sensor networks and showed that this results in a more practical approach, called Sensor Clouds [87]. Sensor cloud infrastructure is designed in a way to handle numerous physical sensors with different owners and paradigms. Hence, through this sensor cloud various sensors groups can share the physical sensors. These sensors produce data in massive amount. As the memory of the sensors is limited, researchers have also encouraged the idea of employing virtual memory into the sensor networks [88].
WSNs have been integrated with FaTVM named virtual memory for carrying out data intensive tasks [89]. FaTVM makes sensor nodes to perform complex computations without using much energy with extensive RAMs. FaTVM has the secondary storage with NAND flashes

and also has the functionality for reducing overhead of the virtual memory . Another idea is to rent the resources from a server that has enough resources and can solve the limited resources problems in WSNs [90].

6.3. Machine Learning based solutions

The possibility of adding cloud storage or server is available, still it is critical to utilize the deployed resources of sensors as efficiently as possible. This is compulsory for the reliable communication in WSNs. There are many associated challenges in this such as figuring out the deficiency of resources, redundancies, controlling the scheduling in an area. Once these challenges are overcome, the network can provide high efficiency localized solution for the maximum-minmum problem. The resource constrained nodes have to operate in dynamic environment. Hence, the management of the resources should be well optimized for adaptive and automatic tasks.
Machine learning has high hopes to perform excellently for different kinds of resource management in networks. Researchers have started implementing machine learning and also deep learning algorithms for this problem such as recently Independent Reinforcement Learning (DIRL) which has been proposed for the optimization of task management [91]. This algorithm gradually learns using the local data present at the nodes with reinforcement learning. DIRL focuses on optimizing the overall system performance and manages a wide range of parameters such as energy consumption, network lifetime and many more.
State of the art machine learning techniques have also been proposed focusing on individual resource management areas. The study has proposed Support Vector Machine (SVM) for energy efficiency in networks [92]. This approach is tested based on the user satisfaction level as most of the times comes at the cost of compromising on user satisfaction. Their results show a tradeoff between energy efficiency and user satisfaction level. Literature review shows that enough studies have proven the effectiveness of machine learning in this area [93].
Due to challenges in data accumulation in WSNs, reinforcement learning has been used extensively in the field. Researchers have used reinforcement learning for energy efficiency [94]. They used the reinforcement learning based algorithm to effectively allocate power to the software update services. Another study also used reinforcement learning for improving routing efficiency and reduces the interferences in the network. Through this algorithm, transmission opportunities through the network also maximize [95].

6.3.1. Challenges for lorawan network managment using machine learning

One of the biggest challenges to use machine learning for resource management is the data collection. It is mandatory to store the massive amount of data from all nodes and gateways in a server so that the data can be analyzed. Lots of information may be lost in this process. Some scenarios of WSNs are particularly difficult to be done with machine learning such as sleeping cell detection [96]. Such nodes lack the essential data required for the timely detection. As a result, applying supervised learning techniques becomes impossible and hence the optimization goals suffer.
Second challenge is the choice of the appropriate machine learning algorithms for the tasks. These algorithms can be trained online or offline [97]. Offline training can be suitable for the

tasks where instant response is not required. On the contrary, online training is necessary to adapt to the dynamic environment instantaneously, for instance, when the tasks are mainly dependent on the time. However, using online training with long time period and highly dependant on the time results in difficulty to produce enough accurate predictions.
In summary, the abundant literature and practical results show that inclusion of machine learning is a feasible idea. However, it still has many technological complexities. Real time intelligence is required particularly for decentralized systems. Data collection and selection of appropriate training algorithms are essential for meeting the complex demands of upcoming latest WSNs.

7. Benchmarking solutions to RRM problems

As a rapidly growing field, numerous ML solutions to communications problems are published every day. It is therefore difficult to keep up with all the proposals, and much less compare them objectively to judge which ones are novel or promising. A major obstacle in this task is the lack of standard problem definitions and implementations against which to benchmark. Most publications in this arena include a definition of the problem jointly with a solution proposal. Although this approach satisfies self-containment requirements, it also facilitates meddling in the problem definition, often inadvertently by the authors themselves. Instead, blinded studies, where the team proposing a solution differs from the one defining the problem seem more reliable and less prone to biases.

7.1. Wireless Suite

The Windmill project is currently exploring approaches to address the problem described above. One of such approaches is *Wireless Suite*, which is a collection of problems in wireless telecommunications, implemented as a Python package. Furthermore, we have open sourced it and made it available online on Github at

https://github.com/nokia/wireless-suite

Wireless Suite is born with the ambition of becoming the reference library against which to evaluate innovations in wireless telecommunications. The library is still in its early days and it only contains one problem definition: the *TimeFreqResourceAllocation-v0* environment. By open sourcing it, we expect it to grow over time as the community adds newer and more challenging problems. Similarly, we expect researchers in the field of ML for communications to evaluate their solutions against the problems provided in Wireless Suite.

The *TimeFreqResourceAllocation-v0* environment simulates an Orthogonal Frequency Division Multiplexing (OFDM) resource allocation task, where a limited number of frequency resources are to be allocated to a large number of User Equipments (UEs) over time. An agent interacting with this environment plays the role of the MAC scheduler. Each time step, the agent must allocate one frequency resource to one of a large number of UEs. The agent gets rewarded for these resource allocation decisions. The reward increases with the number of UEs, whose traffic requirements are satisfied. The traffic requirements for each UE are expressed in terms of their Guaranteed Bit Rate (if any) and their Packet Delay Budget (PDP). Researchers are then invited to develop new agents that interact with this environment and take effective resource allocation decisions.

It is worth noting that *Wireless Suite* is only one of a myriad of possible tools that could be deployed to address Radio Resource Management (RRM) problem benchmarking. We therefore expect additional frameworks and techniques to be developed as the project evolves.

8. References

[1] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[2] B. Bojović, E. Meshkova, N. Baldo, J. Riihijärvi, and M. Petrova, "Machine learning-based dynamic frequency and bandwidth allocation in self-organized LTE dense small cell deployments," *Eurasip Journal on Wireless Communications and Networking*, vol. 2016, no. 1, 2016.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[4] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," 2015.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[7] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning I: Introduction*. Cambridge, MA, USA: A Bradford Book, 1998.

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction. (Second Edition)*. Cambridge, MA, USA: A Bradford Book, 2018.

[10] C. J. C. H. Watkins and P. Dayan, "Q-learning," in *Machine Learning*, pp. 279–292, 1992.

[11] Y. Li, "Deep reinforcement learning: An overview," *ArXiv*, 2017.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, 2015.

[13] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," 2015.

[14] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," 2015.

[15] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, “Ten challenges in advancing machine learning technologies toward 6G,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 96–103, 2020.

[16] N. Abramson, “THE ALOHA SYSTEM: Another alternative for computer communications,” in *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, AFIPS ’70 (Fall), (New York, NY, USA), p. 281–285, Association for Computing Machinery, 1970.

[17] N. Abramson, “The throughput of packet broadcasting channels,” *IEEE Transactions on Communications*, vol. 25, no. 1, pp. 117–128, 1977.

[18] E. Casini, R. De Gaudenzi, and O. Del Rio Herrero, “Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1408–1419, 2007.

[19] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, “Coded random access: applying codes on graphs to design random access protocols,” *IEEE Communications Magazine*, vol. 53, no. 6, pp. 144–150, 2015.

[20] G. Liva, “Graph-based analysis and optimization of contention resolution diversity slotted ALOHA,” *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 477–487, 2011.

[21] E. Paolini, “Finite length analysis of irregular repetition slotted aloha (IRSA) access protocols,” in *2015 IEEE International Conference on Communication Workshop (ICCW)*, pp. 2115–2120, 2015.

[22] M. Berioli, G. Cocco, G. Liva, and A. Munari, “Modern random access protocols,” *Foundations and Trends® in Networking*, vol. 10, no. 4, pp. 317–446, 2016.

[23] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[24] I. T. U. (ITU), *Guidelines for evaluation of radio interface technologies for IMT-2020*. ITU-R Report M.2412-0, 2017. `https://www.itu.int/pub/R-REP-M.2412-2017`.

[25] A. Laya, L. Alonso, and J. Alonso-Zarate, “Is the random access channel of LTE and LTE-A suitable for M2M communications? a survey of alternatives,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 4–16, 2014.

[26] Y. Wu, X. Gao, S. Zhou, W. Yang, Y. Polyanskiy, and G. Caire, “Massive access for future wireless communication systems,” *IEEE Wireless Communications*, pp. 1–9, 2020.

[27] Y. Polyanskiy, “A perspective on massive random-access,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 2523–2527, 2017.

[28] S. Ali, W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wietfeld, K. Mei, H. Shiri, H.-J. Zepernick, T. M. C. Chu, I. Ahmad, J. Huusko, J. Suutala, S. Bhadauria, V. Bhatia, R. Mitra, S. Amuru, R. Abbas, B. Shao, M. Capobianco, G. Yu, M. Claes,

T. Karvonen, M. Chen, M. Girnyk, and H. Malik, "6G white paper on machine learning in wireless communication networks," 2020.

[29] D. C. Nguyen, P. Cheng, M. Ding, D. Lopez-Perez, P. N. Pathirana, J. Li, A. Seneviratne, Y. Li, and H. V. Poor, "Wireless AI: Enabling an AI-governed data life cycle," 2020.

[30] F. Clazzer, A. Munari, G. Liva, F. Lazaro, C. Stefanovic, and P. Popovski, "From 5g to 6g: Has the time for modern random access come?," 2019.

[31] F. Clazzer, B. Matuz, S. Jayasooriya, M. Shirvanimoghaddam, and S. J. Johnson, "Protograph LDPC code design for asynchronous random access," *Algorithms*, vol. 12, no. 8, p. 170, 2019.

[32] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access gaussian channel," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 2528–2532, 2017.

[33] H. Li, "Multi-agent Q-learning for competitive spectrum access in cognitive radio systems," in *2010 Fifth IEEE Workshop on Networking Technologies for Software Defined Radio Networks (SDR)*, pp. 1–6, 2010.

[34] K. A. Yau, P. Komisarczuk, and D. T. Paul, "Enhancing network performance in distributed cognitive radio networks using single-agent and multi-agent reinforcement learning," in *IEEE Local Computer Network Conference*, pp. 152–159, 2010.

[35] Y. Li, "Deep reinforcement learning: An overview," 2017.

[36] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, 2018.

[37] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar, "A deep actor-critic reinforcement learning framework for dynamic multichannel access," 2019.

[38] U. Challita, L. Dong, and W. Saad, "Proactive resource management for LTE in unlicensed spectrum: A deep learning perspective," *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4674–4689, 2018.

[39] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, p. 229–256, May 1992.

[40] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2019.

[41] Y. Li, W. Zhang, C. Wang, J. Sun, and Y. Liu, "Deep reinforcement learning for dynamic spectrum sensing and aggregation in multi-channel wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 464–475, 2020.

[42] E. Nisioti and N. Thomos, "Fast Q-learning for improved finite length performance of irregular repetition slotted ALOHA," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 844–857, 2020.

[43] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8440–8450, 2018.

[44] M. V. da Silva, R. D. Souza, H. Alves, and T. Abrão, "A NOMA-based Q-learning random access method for machine type communications," *IEEE Wireless Communications Letters*, pp. 1–1, 2020.

[45] R. Kassab, A. Destounis, D. Tsilimantos, and M. Debbah, "Multi-agent deep stochastic policy gradient for event based dynamic spectrum access," 2020.

[46] Z. Lu and M. C. Gursoy, "Dynamic channel access and power control via deep reinforcement learning," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1–5, 2019.

[47] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura, "Deep reinforcement learning-based channel allocation for wireless LANs with graph convolutional networks," *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, Sep 2019.

[48] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, 2019.

[49] H. Yang, Z. Xiong, J. Zhao, D. Niyato, C. Yuen, and R. Deng, "Deep reinforcement learning based massive access management for ultra-reliable low-latency communications," 2020.

[50] S. Ali, N. Rajatheva, and W. Saad, "Fast uplink grant for machine type communications: Challenges and opportunities," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 97–103, 2019.

[51] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. T. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: the dominant theme for wireless evolution into 5g," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, 2014.

[52] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," Technical Specification (TS) 36.331, 3rd Generation Partnership Project (3GPP), 04 2017. Version 14.2.2.

[53] Y. Watanabe, Y. Matsunaga, K. Kobayashi, H. Sugahara, and K. Hamabe, "Dynamic neighbor cell list management for handover optimization in lte," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pp. 1–5, 2011.

[54] P. Lei, T. Shen, W.-C. Peng, and I. Su, "Exploring spatial-temporal trajectory model for location prediction," in *Proceedings - 2011 12th IEEE International Conference on Mobile Data Management, MDM 2011*, Proceedings - IEEE International Conference on Mobile Data Management, pp. 58–67, Nov. 2011. null ; Conference date: 06-06-2011 Through 09-06-2011.

[55] F. Yang, H. deng, F. Jiang, and X. Deng, "Handover optimization algorithm in lte high-speed railway environment," *Wireless Personal Communications*, vol. 84, 06 2015.

[56] I. Balan, T. Jansen, B. Sas, I. Moerman, and T. Kürner, "Enhanced weighted performance based handover optimization in lte," in *2011 Future Network Mobile Summit*, pp. 1–8, 2011.

[57] J. Chen, C. Yang, and Y. Mai, "A novel smart forwarding scheme in lte-advanced networks," *China Communications*, vol. 12, no. 3, pp. 120–131, 2015.

[58] H. Tu, J. Lin, T. Chang, and K. Feng, "Prediction-based handover schemes for relay-enhanced lte-a systems," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2879–2884, 2012.

[59] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception," TS TS36.104, 3rd Generation Partnership Project (3GPP), Sept. 2019. 15.8.0.

[60] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, second ed., 2018.

[61] 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, p. 97, Jan 2016.

[62] M. A. Ma'Sum, M. K. Arrofi, G. Jati, F. Arifin, M. N. Kurniawan, P. Mursanto, and W. Jatmiko, "Simulation of intelligent unmanned aerial vehicle (UAV) for military surveillance," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 161–166, IEEE, Mar 2013.

[63] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Transactions on Vehicular Technology*, vol. 62, pp. 3359–3370, Mar 2013.

[64] 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)*, pp. 1–8, IEEE, May 2014.

[65] 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*, Jul 2020.

[66] Z. Zhao, P. Cumino, A. Souza, D. Rosário, T. Braun, E. Cerqueira, and M. Gerla, "Software-defined unmanned aerial vehicles networking for video dissemination services," *Ad Hoc Networks*, vol. 83, pp. 68–77, Feb 2019.

[67] Z. Drezner, "Computation of the multivariate normal integral," *ACM Transactions on Mathematical Software (TOMS)*, vol. 18, pp. 470–480, Dec 1992.

[68] P. N. Somerville, "Numerical computation of multivariate normal and multivariate-t probabilities over convex regions," *Journal of Computational and Graphical Statistics*, vol. 7, pp. 529–544, Dec 1998.

[69] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, pp. 1123–1152, Nov 2015.

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

[71] "IEEE standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," Dec. 2016.

[72] "Status of project IEEE 802.11ay." Available: http://www.ieee802.org/11/Reports/tgay_update.htm.

[73] Task Group ay, "IEEE 802.11 TGay use cases," Aug. 2017. doc.: IEEE 802.11-2015/0625r7.

[74] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: directional 60 GHz communication for multi-gigabit-per-second Wi-Fi," *IEEE Communications Magazine*, vol. 52, pp. 132–141, Dec. 2014.

[75] E. Khorov, A. Ivanov, A. Lyakhov, and V. Zankin, "Mathematical model for scheduling in IEEE 802.11ad networks," in *IFIP Wireless and Mobile Networking Conference (WMNC)*, pp. 153–160, July 2016.

[76] M. U. Rajan and A. Babu, "Theoretical maximum throughput of IEEE 802.11 ad millimeter wave wireless LAN in the contention based access period: with two level aggregation," in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2531–2536, IEEE, Mar. 2017.

[77] H. Shokri-Ghadikolaei, L. Gkatzikis, and C. Fischione, "Beam-searching and transmission scheduling in millimeter wave communications," in *IEEE International Conference on Communications (ICC)*, pp. 1292–1297, IEEE, June 2015.

[78] W. Feng, Y. Wang, D. Lin, N. Ge, J. Lu, and S. Li, "When mmWave communications meet network densification: a scalable interference coordination perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 1459–1471, July 2017.

[79] H. Assasa and J. Widmer, "Implementation and evaluation of a WLAN IEEE 802.11ad model in ns-3," in *Workshop on Ns-3 (WNS3)*, pp. 57—64, June 2016.

[80] H. Assasa, J. Widmer, T. Ropitault, and N. Golmie, "Enhancing the ns-3 IEEE 802.11ad model fidelity: beam codebooks, multi-antenna beamforming training, and quasi-deterministic mmwave channel," in *Workshop on Ns-3 (WNS3)*, pp. 33–40, June 2019.

[81] N. Feamster and J. Rexford, “Why (and how) networks should run themselves,” in *Applied Networking Research Workshop (ANRW)*, pp. 20–20, ACM, July 2018.

[82] P. Gawłowicz and A. Zubow, “Ns-3 meets OpenAI gym: The playground for machine learning in networking research,” in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 113–120, Nov. 2019.

[83] R. Munos, T. Stepleton, A. Harutyunyan, and M. G. Bellemare, “Safe and efficient off-policy reinforcement learning,” in *International Conference on Neural Information Processing Systems (NIPS)*, pp. 1054–1062, Dec. 2016.

[84] P. P. Ray, “A survey on internet of things architectures,” *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.

[85] P. Dutta and A. Dunkels, “Operating systems and network protocols for wireless sensor networks,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 68–84, 2012.

[86] W.-H. Cho, J. Kim, and O. Song, “An efficient resource management protocol for handling small resource in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 5, p. 324632, 2013.

[87] S. K. Dash, J. P. Sahoo, S. Mohapatra, and S. P. Pati, “Sensor-cloud: assimilation of wireless sensor network and the cloud,” in *International Conference on Computer Science and Information Technology*, pp. 455–464, Springer, 2012.

[88] M. Yuriyama and T. Kushida, “Sensor-cloud infrastructure-physical sensor management with virtualized sensors on cloud computing,” in *2010 13th International Conference on Network-Based Information Systems*, pp. 1–8, IEEE, 2010.

[89] N. Lin, Y. Dong, and D. Lu, “Fast transparent virtual memory for complex data processing in sensor networks.,” *SENSORNETS*, vol. 24, p. 34, 2012.

[90] E. Callaway, P. Gorday, L. Hester, J. A. Gutierrez, M. Naeve, B. Heile, and V. Bahl, “Home networking with ieee 802.15. 4: a developing standard for low-rate wireless personal area networks,” *IEEE Communications magazine*, vol. 40, no. 8, pp. 70–77, 2002.

[91] A. Arya, A. Malik, and R. Garg, “Reinforcement learning based routing protocols in wsns: a survey,” *Int. J. Comput. Sci. Eng. Technol*, vol. 4, pp. 1401–1404, 2013.

[92] L. Xu and A. Nallanathan, “Energy-efficient chance-constrained resource allocation for multicast cognitive ofdm network,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1298–1306, 2016.

[93] M. Ali Imran, A. Flávia dos Reis, G. Brante, P. Valente Klaine, and R. Demo Souza, “Machine learning in energy efficiency optimization,” *Machine Learning for Future Wireless Communications*, pp. 105–117, 2020.

[94] X. Chen, Z. Zhao, and H. Zhang, "Stochastic power adaptation with multiagent reinforcement learning for cognitive wireless mesh networks," *IEEE transactions on mobile computing*, vol. 12, no. 11, pp. 2155–2166, 2012.

[95] H. A. Al-Rawi, K.-L. A. Yau, H. Mohamad, N. Ramli, and W. Hashim, "Effects of network characteristics on learning mechanism for routing in cognitive radio ad hoc networks," in *2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP)*, pp. 748–753, IEEE, 2014.

[96] S. Fortes, R. Barco, and A. Aguilar-Garcia, "Location-based distributed sleeping cell detection and root cause analysis for 5g ultra-dense networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 149, 2016.

[97] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2392–2431, 2017.

www.windmill-itn.eu