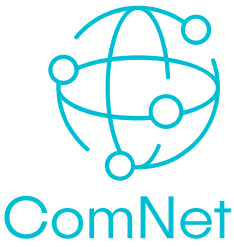


ComNets

TUHH
Institute of
Communication
Networks

# MASTER'S THESIS
# TERESA ALGARRA ULIERTE

—

Machine Learning-based Dynamic Spectrum Access for Aircraft-to-Aircraft Communication under Coexistence with Legacy Radio Systems

01.07.2022

tuhh.de

Machine Learning-based Dynamic Spectrum Access for Aircraft-to-Aircraft Communication under Coexistence with Legacy Radio Systems
*Dynamischer Kanalzugriff durch Maschinelles Lernen für Flugzeug-zu-Flugzeug Kommunikation unter Koexistenz mit klassischen Funksystemen*

Teresa Algarra Ulierte
Matriculation number: 55050
International M.Sc. in Information and Communication Systems

Hamburg University of Technology
Institute of Communication Networks
First examiner: Prof. Dr.-Ing. Timm-Giel
Second examiner: Prof. Dr. rer. nat. Volker Turau
Supervisor: Sebastian Lindner

Hamburg, 01.07.2022

# Declaration of Originality

I hereby declare that the work in this thesis was composed and originated by myself and has not been submitted for another degree or diploma at any university or other institute of tertiary education.

I certify that all information sources and literature used are indicated in the text and a list of references is given in the bibliography.

Hamburg, 01.07.2022

Teresa Algarra Ulierte

# Acknowledgements

I would like to thank my supervisor, Sebastian Lindner. Without his guidance and dedication this project would have never been possible. I would like to extend my thanks to Daniel Stolpmann, who has always made himself available for me whenever his help was needed. A special acknowledgement goes as well to Leonard Fisser, Konrad Fuger, Yevhenii Shudrenko and the rest of the members of the Institute for Communication Networks for their time and their invaluable feedback.

I am also extremely grateful for the help of Dr.-Ing. Koojana Kuladinithi, who has gone above and beyond to allow me to make the best out of this experience, and for Prof. Dr.-Ing. Andreas Timm-Giel, who made time for me even when no time was available.

Lastly, I'd like to show appreciation to my family and close friends, and more specifically to my parents, who have been an unconditional support in every step of the way.

# Abstract

Spectrum scarcity is viewed as one of the key obstacles in the area of wireless communications. The lack of available unlicensed resources is impairing the development of newer and more modern communications systems. This is the case of *L*-band Digital Aeronautical Communications System (LDACS), an innovative air communication system that aims to use the part of the frequency spectrum licensed by Distance Measuring Equipment (DME), a legacy radio navigation system. DME has a low channel utilization rate, leaving idle numerous resources that could be used by LDACS through the use of Dynamic Spectrum Access (DSA) in Cognitive Radio (CR). In order to avoid interferences and collisions while taking advantage of these idle resources, this thesis proposes a new LDACS Machine Learning (ML)-based Medium Access Control (MAC). It incorporates a Long Short Term Memory (LSTM) Recurrent Neural Network (RNN) in order to observe, learn, predict and avoid the DME licensed users. The results from this new MAC are analyzed and compared to two alternative approaches, showing the advantages of a ML-based approach.

# Contents

# 1 Introduction

Spectrum scarcity is viewed as one of the key obstacles in the area of wireless communication. This is not a new problem, and it has been studied since the 1960s [1]. However, during the past years a growing number of communication systems have been created, each requiring a portion of the frequency spectrum in which to transmit.

A more specific case of this problem is the air communication, that is, Air-to-Air (A2A) and Air-to-Ground (A2G), whose allocated part of the frequency spectrum is the aeronautical L-band (960 MHz - 1215 MHz). This part of the frequency spectrum is occupied by communication systems related to both civil and military aviation, such as DME, Tactical Air Navigation System (TACAN) or Automatic Dependent Surveillance-Broadcast (ADS-B). However, many of these systems are legacy systems, meaning that they have been operating for decades. Due to how the frequency spectrum was assigned and licensed when these legacy systems were created, they have a larger portion of the frequency spectrum under license than they actually need. In [2], it is shown that only 20% of these licensed channels have signals with more than 7dB over the environmental noise level. In other words, around 80% of the licensed spectrum is locally left unused by these legacy systems.

This means that designing a way of taking advantage of the remaining 80% of unused frequency spectrum would not only allow for other communication systems to transmit and receive, but it would also increase the efficiency as well as lowering the energetic and financial costs. Therefore, this project proposes the use of channel sensing and ML to learn and predict the frequency spectrum access behavior of these legacy systems in order for new communication system to be able to transmit during the time slots and frequency channels left underused and idle by the licensed legacy systems.

More specifically, this is done within the framework of two concrete systems: DME and LDACS. LDACS is part of Single European Sky ATM Research (SESAR), which is a European effort to accelerate the development of the "Digital European Sky" through research and innovation [3]. It aims to modernize the aviation infrastructure to allow for more air traffic volume at a higher speed, all while becoming more sustainable, efficient and cost-effective. However, one of LDACS's biggest obstacles is, as for all emerging wireless communication systems, the frequency spectrum scarcity. The aim of this thesis is to demonstrate that, using a ML- based MAC, LDACS would be able to use the same frequency channels as DME without colliding or causing interferences upon it.

# 2 Related Work

Coexistence between a licensed user and one or several unlicensed users in the same part of the frequency spectrum is not a new concept. This technique can be categorized into the field of CR and one example of its usage is Television White Space (TVWS) [4]. TVWSs are unused TV channels that were opened for unlicensed use in 2010 by the Federal Communications Commission (FCC). This opening made nearly 100 MHz available for unlicensed users. Nevertheless, transmissions must be scheduled several hours in advance in order to avoid collisions with the licensed user. A2A communications cannot be scheduled with such precision beforehand, they happen "on the spot". Moreover, the scenario of communicating aircraft does not have a static frequency spectrum, but rather a dynamic one, meaning that the communication process is affected by the varying distance between the different aircraft and between the aircraft and the ground stations, their speed and other interfering factors [5]. Therefore, the scenario described in this thesis also falls under the category of DSA, that is, an aspect of CR [6], [7]. Some examples of systems that rely on DSA are the Long Term Evolution Unlicensed (LTE-U) standard, which will allow for both LTE-U and Wi-Fi to coexist by accessing the unlicensed 5 GHz frequency band [8]; or many of the Internet of Things (IoT) deployments [9].

In order to avoid collisions and interferences, the observation and avoidance of the licensed user and of other unlicensed users accessing the same part of the frequency spectrum is a vital part of CR and DSA. One of the most popular solutions that is currently being researched is to **learn** how the other users behave, **predict** the future behavior and **avoid** the time slots and frequency channels in which it is predicted that a collision might happen. The learning and predicting part of this approach is achieved though ML. Some of these ML solutions use multiagent reinforcement learning such as [5], [7], [10], and other LSTM RNNs such as [11]–[13]. The one thing that all of these approaches have in common is that, for the learning algorithm to be able to identify a pattern, a pattern has to exist.

As an example, some protocols like Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) rely on random access in order to work and have therefore no clear pattern to be learned by a learning algorithm. These would not be a good learning target for the ML-based solutions presented above. However, many other systems show some kind of periodicity, whether native (from the protocol itself) or hidden (induced by the upper layers). For example, sensor networks that periodically offload data will present a periodic medium access behavior [14]. Similarly, IoT applications that have control loops or are in charge of monitoring will also show periodicity [9]. It is this sort of protocols that are most suitable to be observed and predicted. It is shown in [7] that this kind of

periodic behaviors can be learned using Deep Q learning, whereas [11], [12] reach the same conclusion using Feed-Forward Deep Learning and LSTM RNN, the same kind of Artificial Neural Network (ANN) that was used in this thesis.

What the learning algorithm predicts are not absolute values, but probabilities of the licensed user behaving in a certain way. As such, some of these probabilities can be expressed in terms of Markov Chain (MC) (see Figure 2.1). This is used in [7], [11], [15], and allows for a mathematical representation and analysis of the frequency channels as well as the possible transmitting options for the unlicensed user. This approach is not the natural one for the DME behavior since it is deterministic, and although possible, the Markov property would be violated.

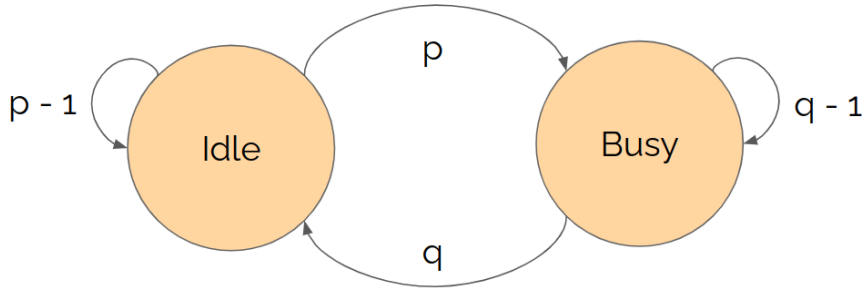



Figure 2.1: Markov Chain describing the state of a frequency channel.

Moreover, Game Theory (GT) can take advantage of this mathematical formulation and propose strategic solutions that can benefit all unlicensed users trying to access the licensed frequency channel while avoiding the main licensed user, as studied in [5], [16] and [15]. While GT is criticized in [17] for requiring too much information (and therefore overhead) to be shared among the different unlicensed users involved, this problem can be solved with an appropriate channel sensing technique, such as the ones proposed in [18].

# 3 Background

In order to further understand the methodology used in this thesis, a strong foundation must be established and each system must be clearly defined. Therefore, this section presents a detailed explanation of DME, LDACS, the coexistence between the two of them and, lastly, a description of the chosen learning algorithm.

## 3.1 DME

In this project, the DME communication system will be considered as the frequency spectrum license holder. Its name stands for Distance Measuring Equipment and is a radio navigation system. DME is considered a legacy system since it has been operative since the 1950s. It has a license over numerous channels across the aeronautical L-band. More specifically, it can operate in the range of 978 MHz - 1213 MHz [19]. However, and despite its large frequency spectrum range, it is a system that often leaves underutilized and idle channels for long periods of time, especially around low air traffic volume rural areas and over large water masses.

The distance is obtained by measuring the propagation delay between the aircraft and the ground station (see Figure 3.1) [20].

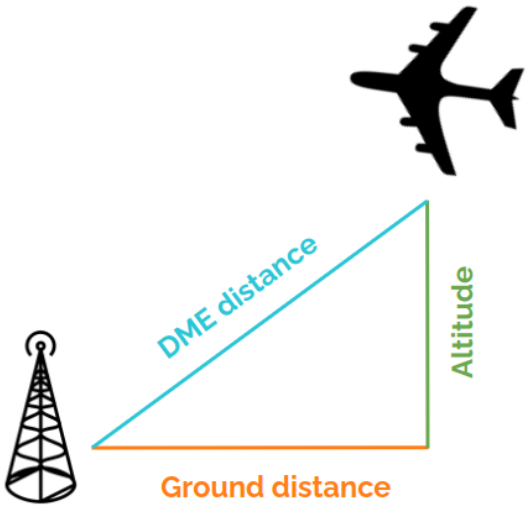



Figure 3.1: DME functioning.

For this, two signals travel from and to the DME user: one sent from the aircraft to the ground station, called request or interrogation, and another one sent from the ground station to the aircraft, called reply or response (see Figure 3.2). The time between the request transmission and the reply reception is used to calculate the distance from the aircraft to the ground station.

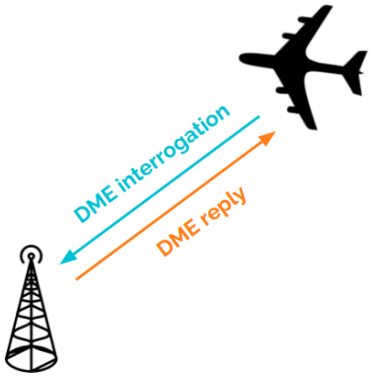



Figure 3.2: Interrogation and reply signals.

Both interrogation and reply are pulse pairs lasting 12 µs as seen on Figure 3.3. After the interrogation is sent, the aircraft will wait for an answer for (at least) 60 µs. During that time, it is expected that the ground station will reply after a certain delay and with a frequency offset of ± 63 MHz (see Figure 3.4). The reply delay depends on the "DME Channel", that is, the mode that is being used. There are four possible channels: X, Y, W and Z. The most common one and the only one that will be considered throughout this project is channel X, for which the reply delay is 50 µs, as shown in Figure 3.4. Channel Y is also commonly used but far less than channel X, and lastly channels W and Z are only utilized for microwave landing systems and are, therefore, rarely used. [21].

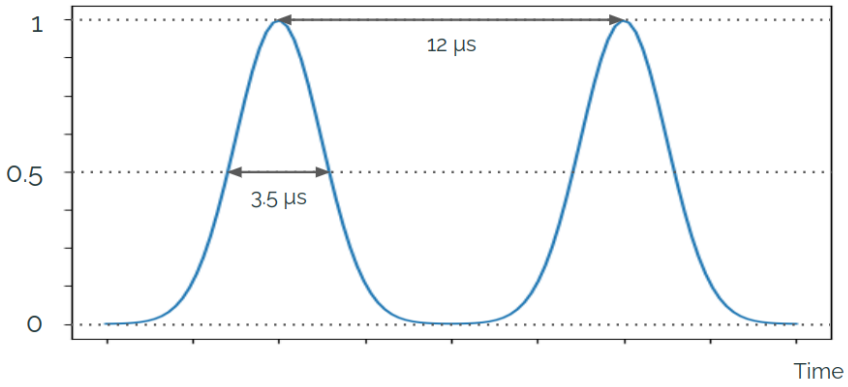



Figure 3.3: Interrogation and reply pulse pair.

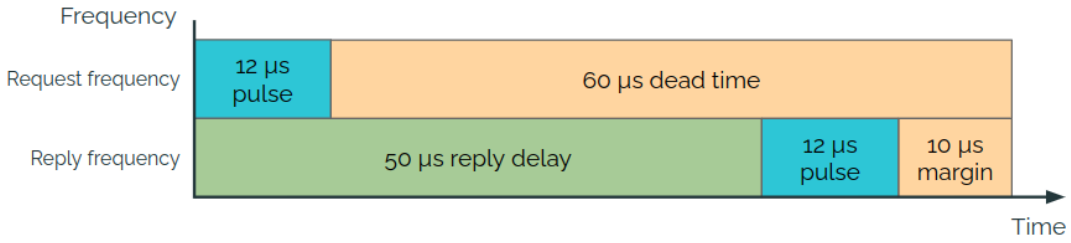



Figure 3.4: DME request-reply procedure.

DME does not send a single interrogation for positioning, but rather periodical interrogation pulses [22]. For this, there are two possibilities:

- **Search mode**: this mode is used when the aircraft has not yet located a responsive ground station. It has a high interrogation frequency, that is, it can send around 150 pulse pairs per second (ppps) in order to locate and establish which ground station is answering or if there is one at all. This mode is only used around 5% of the time by each user [21].
- **Track mode**: this mode is used once the aircraft has already located the ground station with which it will communicate. The frequency of interrogation pulses then decreases drastically to only 5-15 ppps. This mode is used during the remaining 95% of the time by each user [21].

Moreover, DME ground stations do not only send reply signals upon receiving a request, but they also send "squitter pulses", which are single pulses periodically sent with a frequency of up to 2700 pulses per second [20]. If a request is received, a pulse pair is sent instead of a single pulse, indicating to the aircraft that it was not a regular squitter pulse but a reply. However, there is no difference between the reply pulse pairs, so the aircraft cannot know which reply was meant for it and which was meant for another aircraft. If an aircraft sends a request, but it first detects and accepts a reply meant for a different aircraft, it would calculate an erroneous location for the ground station. Therefore, the location is only deemed accurate enough when multiple replies have been received with consistent delay, that is, a delay that matches the expected distance. For this, a maximum reply delay of 2500 µs is allowed. This number is the round trip for the maximum radius that DME can cover, which is 200 nautical miles [21].

## 3.2 LDACS

In this project, the LDACS communication system will be considered as the unlicensed user. Its name stands for L-band Digital Aeronautical Communication System, and it is a communication system designed for A2A and A2G communication. As stated above, it

is part of a European effort to digitalize and modernize the aviation infrastructure while creating more sustainable systems. Moreover, it is envisioned by European Organisation for the Safety of Air Navigation (EUROCONTROL) to be the next terrestrial data link system [23] due to its enhanced safety, security, resilience and cost efficiency [24]. At the time of writing, the LDACS A2A MAC has not been published, but early versions of the specification such as [25] have been made available to the author.

The main focus of this project is the A2A component of LDACS, which achieves its advantages thanks to an advanced two-part MAC. It comprises a Shared Channel (SH) used for broadcasting and control messages, and multiple Point-to-Point Channel (PP) which are used for unicast messages among two communicating aircraft (see Figure 3.5) [26].

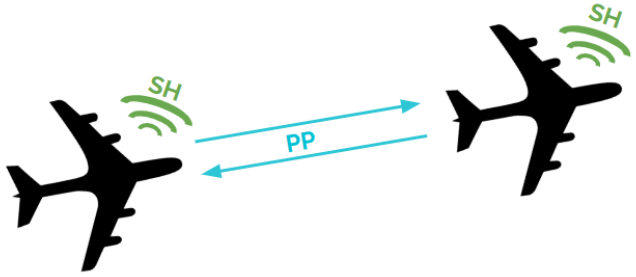



Figure 3.5: Two LDACS users communicating.

LDACS makes use of Orthogonal Frequency Division Multiplexing (OFDM) access as its digital modulation scheme [27]. This means that the frequency spectrum is divided into $n$ frequency channels of equal bandwidth, and the time is divided into time slots of equal duration, where the duration can be 6 ms, 12 ms or 24 ms. A graphical representation of this system can be seen in Figure 3.6.

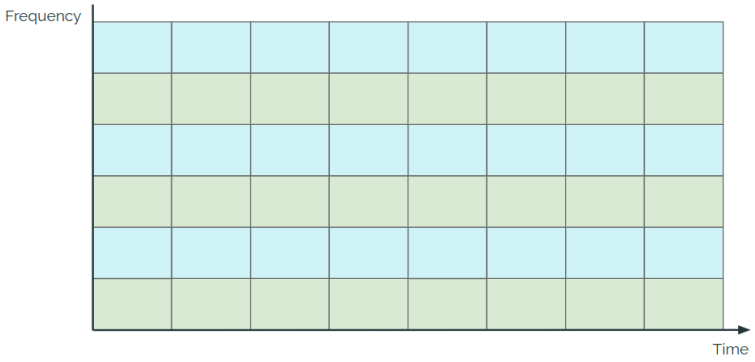



Figure 3.6: OFDM graphical representation.

Therefore, in terms of SH and PP, the LDACS frequency spectrum over time can be represented as seen in Figure 3.7.

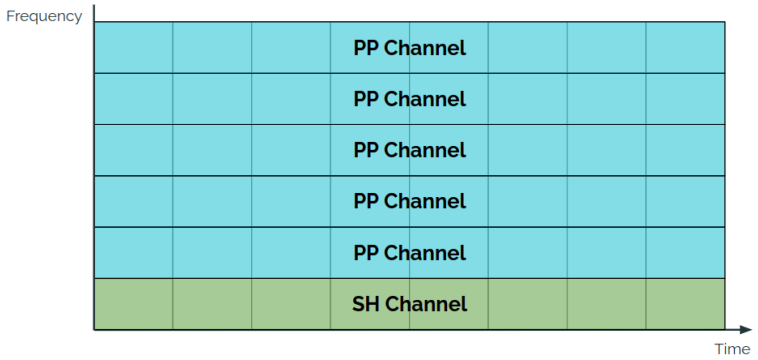



Figure 3.7: LDACS frequency spectrum.

In order for a communication link to be established between two aircraft, a handshake must be executed in the SH through broadcast messages. This is accomplished by using randomized slotted ALOHA channel access with slot advertisement [28], [29]. The process starts by choosing a value $k$ when a broadcast message is to be sent. The slot in which the transmission will occur is randomly assigned out of the next $k$ time slots (see Figure 3.8). Should there be a transmission already planned in that time slot, both packets would collide. A large $k$ value would grant a lower collision probability, although a larger delay due to a greater number of time slots to be assigned. On the other hand, a smaller $k$ value would provide a higher collision probability in exchange for a smaller delay. The value $k$ should therefore be chosen accordingly to the requirements of the communication link.

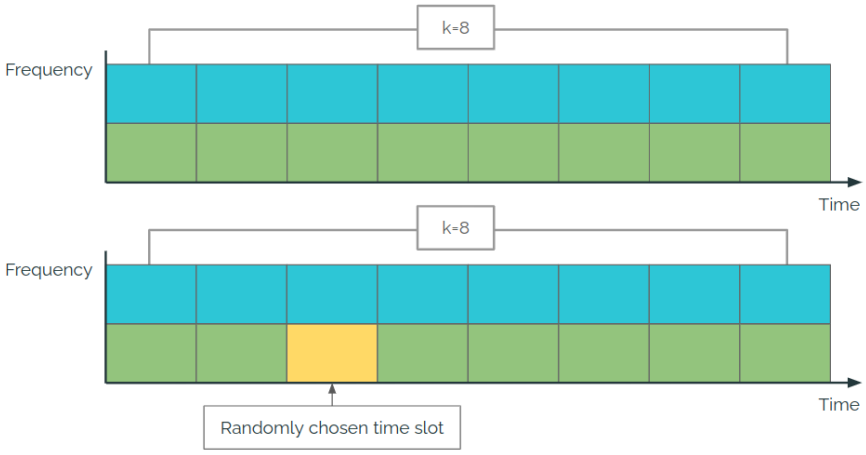



Figure 3.8: Randomized slotted ALOHA channel access.

The amount of data to be sent needs to be distributed among more than one time slot, so in order to avoid going through the chance of colliding again, the user simply announces in which time slot will the transmission be continued [28]. This is called "slot advertisement" (see Figure 3.9). All other users must avoid transmitting in this time slot, which guarantees interference-free communication for the transmitting user among the group of recipients only, meaning that the hidden node problem remains.

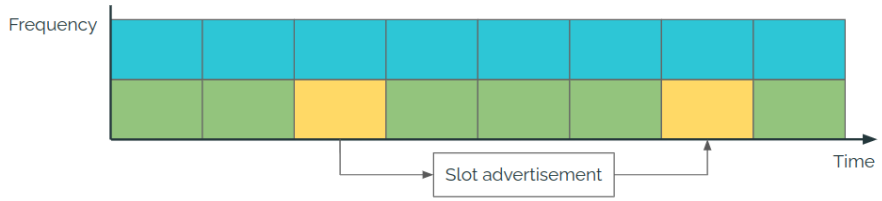



Figure 3.9: Slot advertisement.

For the actual handshake process, one aircraft will propose a set of time slots located in a PP to send unicast messages to another aircraft. Should the second aircraft find one of these time slots suitable, it will send a reply in the SH accepting the communication request (see Figures 3.10 and 3.11) [30].

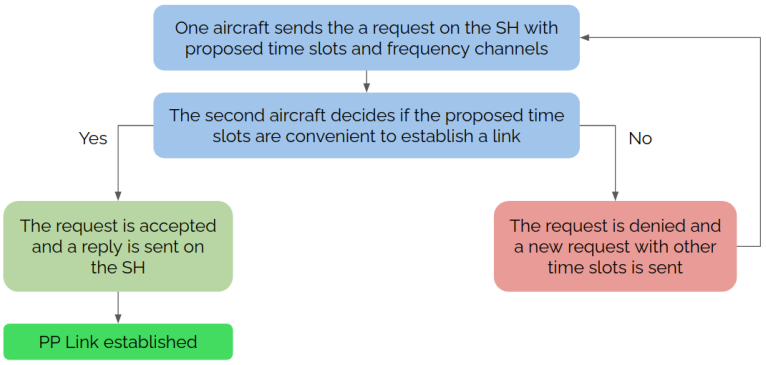



Figure 3.10: Handshake Workflow.

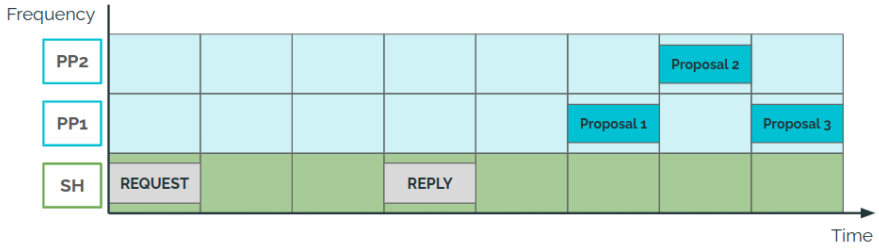



Figure 3.11: Handshake between two aircraft.

The communication in the PPs follows a reservation-based channel access. This means that, since both aircraft involved have broadcasted the time slots in which they will communicate in the SH, no other aircraft should interfere in that communication. The established communication link will be kept in the agreed upon time slots until the link times out, that is, after a fixed amount of time slots. At this point, the process for establishing a communication link will start once again with one of the aircraft sending a request in the SH (see Figure 3.12).

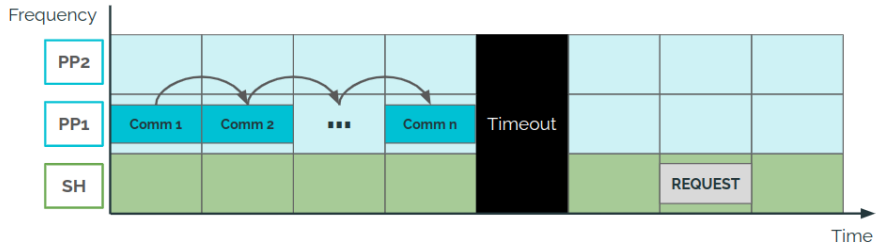



Figure 3.12: Established communication link in the PP.

## 3.3 Coexistence

LDACS aims to occupy the part of the aeronautical band that is licensed to DME (see Figure 3.13) [20]. This can cause interference-related problems as well as impair DME's transmissions. Being a radio navigation system, DME is a safety critical system and therefore cannot tolerate any kind of intrusion.

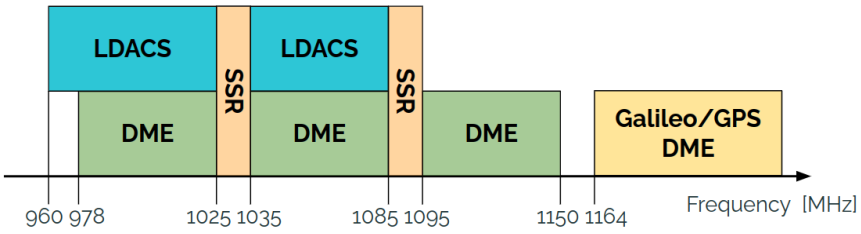
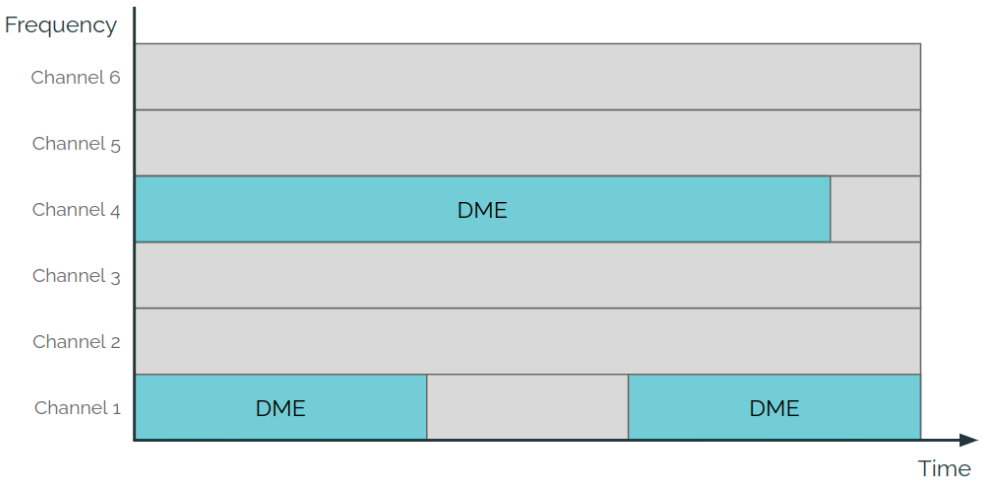
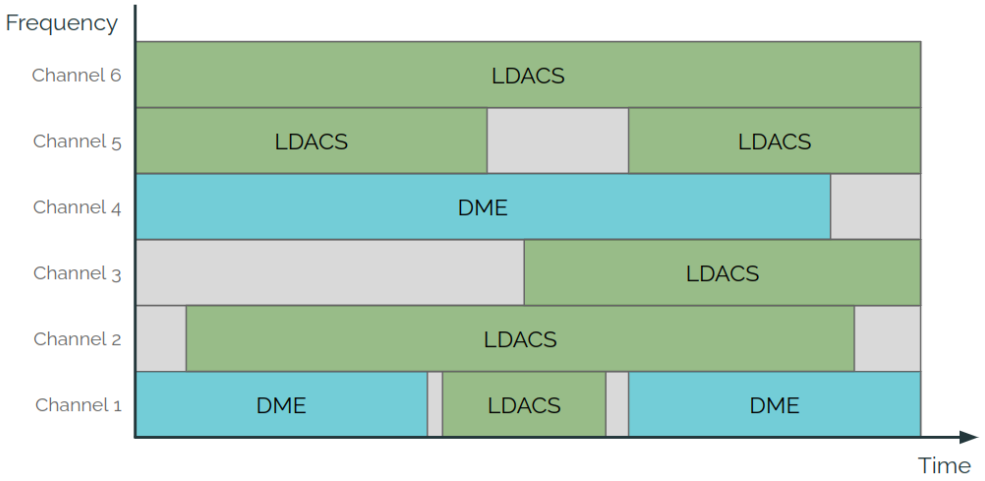



Figure 3.13: Frequency Spectrum Overview.

The coexistence solution that LDACS is aiming to achieve would be to take advantage of the empty slots that are left in DME's licensed spectrum. Given the low utilization rate of its licensed channels, the DME frequency spectrum usage could look as depicted in Figure 3.14a at a certain moment. Allowing LDACS to make use of the idle and underused channels would make the frequency spectrum usage look closer to the one depicted in Figure 3.14b.




(a) Exemplary underused frequency spectrum.




(b) Exemplary opportunistically accessed frequency spectrum.

Figure 3.14: Example of frequency spectrum usage.

This is what was defined in Chapter 2 as DSA and CR. In this case, DME is the licensed user, which means that the system must be respected at all costs, and LDACS is the unlicensed user, and must therefore adapt to the license holder.

In order to successfully take advantage of the underused and idle DME channels, a ML-based approach based on the literature will be presented.

## 3.4 Learning Algorithm

Given the periodic behavior of DME, the spectrum availability can be predicted based on channel sensing observations. This means that with the use of a learning algorithm, and given a set of observations containing the frequency spectrum access behavior of DME, the future behavior of DME can be predicted. These predictions can be used in order to access the idle and underused channels while avoiding collisions. It would work as depicted in Figure 3.15, and according to the following steps:

1. Observations are achieved through channel sensing.
2. These observations are fed to the learning algorithm.
3. The learning algorithm predicts the future behavior of DME.
4. The predicted behavior is later compared to the actual behavior of DME, and the accuracy is sent to the learning algorithm in order to better train itself.

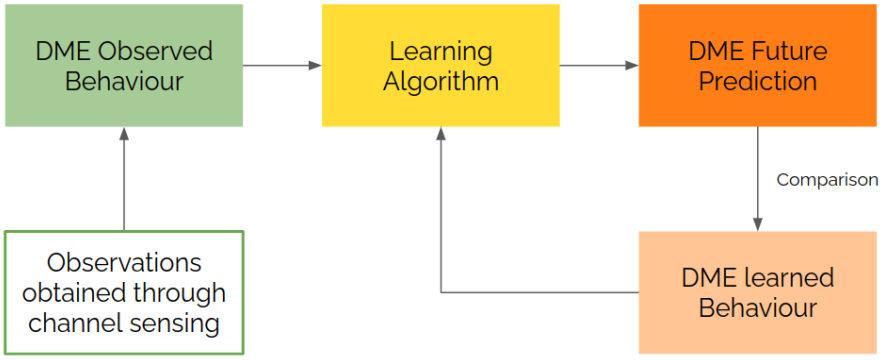



Figure 3.15: Learning Algorithm.

This learning algorithm is achieved through a LSTM RNN. RNNs are a type of ANN that are used to recognize patterns, classify, cluster and make predictions about data [31]. Moreover, they perform especially well when used on sequential data [32]. LSTMs are a certain type of RNNs that can train over long sequences of data and retain memory thanks to a memory cell [31]. The use of a LSTM RNN is proven to work very well for this scenario in [11] and [12].

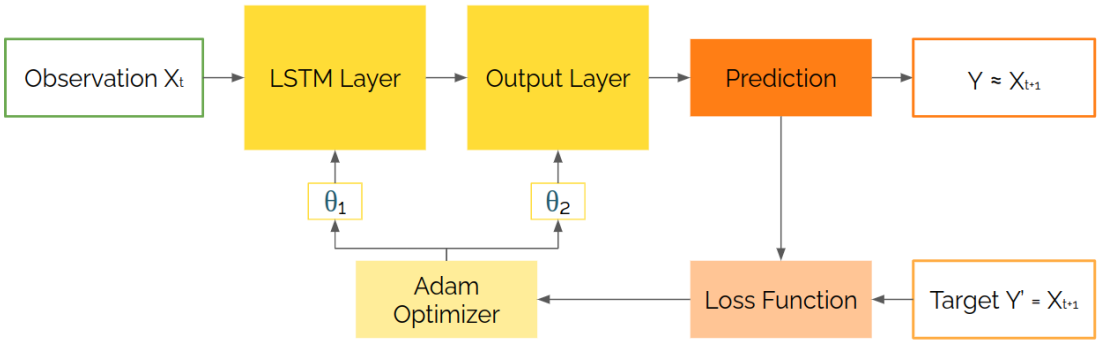



Figure 3.16: LSTM RNN Scheme.

For this project, it is assumed that the observations are achieved through channel sensing, and that all channels can be sensed at the same time [33]. In other words, full observability is assumed. It is acknowledged that this is an unrealistic assumption, and that more sophisticated learning algorithms may be required to solve the problem under partial observability. However, this thesis is concerned more with the integration of predictions into the LDACS A2A MAC, and so the assumption is made.

# 4 Model

It is clear that the frequency spectrum usage of DME leaves plenty of idle and underused channels that could potentially be used by LDACS. The questions that are left to answer are how successful this opportunistic medium access could be and how it would work.

## 4.1 Probability of Success

For this thesis, the probability of success is defined as the probability of LDACS successfully finding the necessary resources to establish a link. The question that needs to be answered is: what is the probability that LDACS finds idle resources to transmit in? In other words, what is the probability that no DME transmissions will happen during a full LDACS time slot? This issue can be simplified by treating it as a regular combinatorics problem, and exemplified using balls and boxes: each LDACS time slot is represented as a numbered box $B_m = 1, 2, ..., m$, and each DME transmission is represented as a numbered ball $B_n = 1, 2, ..., N$. All balls are indistinguishable from each other and have equal chances of arriving at any box (see Figure 4.1). Given this formulation, the aforementioned questions would turn into: what is the probability that one box will receive no balls? (see Figure 4.2).

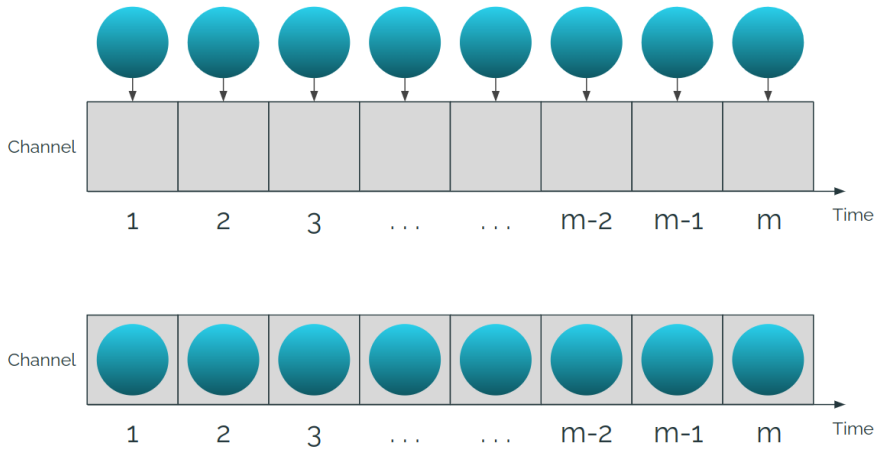



Figure 4.1: Exemplification of the probability of success.

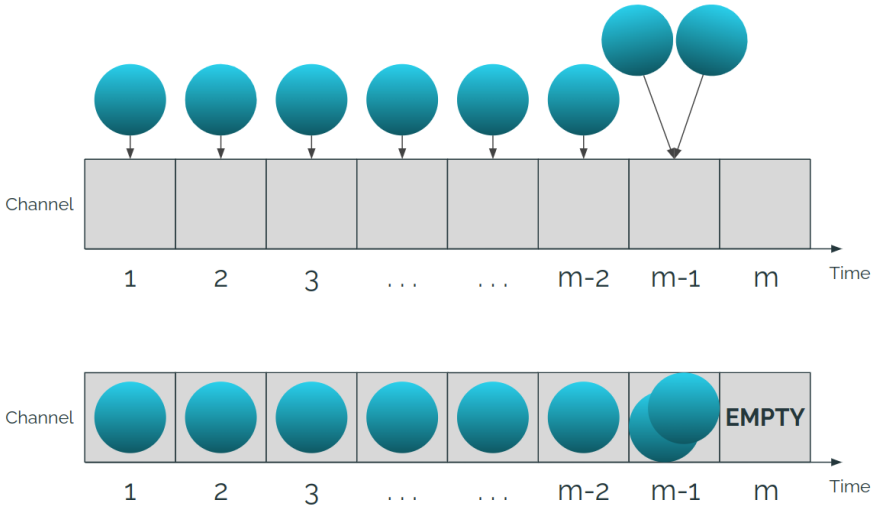



Figure 4.2: Exemplification of the probability of success questions.

Each ball has $m$ ways being distributed into the $m$ boxes. Therefore, $N$ balls have $m^N$ ways of being distributed into $m$ boxes. For one box to be empty, the $N$ balls would have to be distributed in $m-1$ boxes, having $(m-1)^N$ ways in which this can be done. Hence, the probability of this happening is as seen on Equation 4.1.

$$\left(\frac{m-1}{m}\right)^N \tag{4.1}$$

In order to make this formula a bit more specific, it will be given the values corresponding to 1 s of transmissions. Taking the LDACS time slot length as 12 ms, there are 8.3 time slots per second, which will be taken as just 8 time slots for this example. The result would then be as shown on Equation 4.2, and its plot can be seen in Figure 4.3.

$$\left(\frac{8-1}{8}\right)^N = \left(\frac{7}{8}\right)^N \tag{4.2}$$

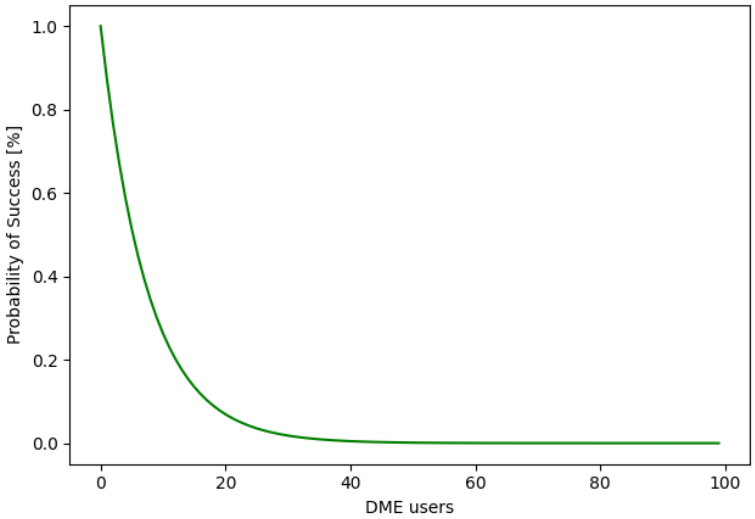



Figure 4.3: Probability curve for a 12 ms time slot length.

It can be seen that there is a relatively high chance of finding available resources for up to 20 DME users. Moreover, the time slot duration can be reduced to 6ms in areas with high air traffic volume, resulting in 16 time slots per second instead of 8. Therefore, the result would be as shown on Equation 4.3, and its plot can be seen in Figure 4.4.

$$\left(\frac{16-1}{16}\right)^N = \left(\frac{15}{16}\right)^N \tag{4.3}$$

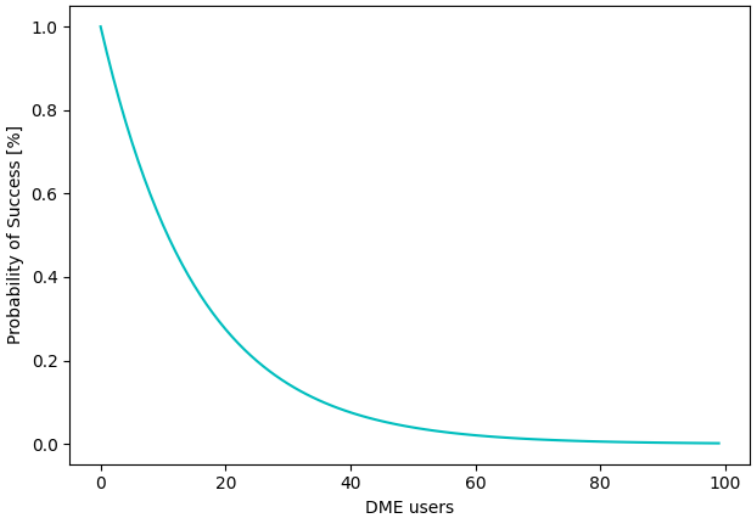



Figure 4.4: Probability curve for a 6ms time slot length.

This time slot duration yields a higher probability of finding available resources for a higher number of DME users, being consequently better suited for busier areas with higher air traffic volume and more transmitting DME users.

Each DME ground station can service up to 100 users [21], which is the reason why the x-axis in both Figures 4.3 and 4.4 shows up to that number. However, this is rarely the case as mentioned in Chapter 3. On a regular basis, the frequency spectrum of which DME is the license holder is underused, and it is only filled and taken advantage of in certain areas with very high air traffic volume, such as the European airports in Dublin or London [20]. This opportunistic medium access system is not expected to work around said areas, but rather take advantage of the available resources in lower air traffic areas with lower channel utilization rate.

## 4.2 Augmented LDACS MAC

Knowing that there are available resources left by DME, and that LDACS has chances of being able to access them, the LDACS MAC needs to be modified and adapted in order to take this knowledge into consideration. This modification is referred to as **Augmented LDACS MAC**.

### 4.2.1 Learning Algorithm

As mentioned in Chapter 3, the behavior of DME is periodic and can be predicted using ML. More specifically, the literature suggests the use of LSTM RNNs. The predictions of the future medium access behavior of DME can be used by LDACS to identify upcoming idle time slots where a transmission can be established (see Figure 4.5). This information is collected and processed independently by all the aircraft involved in the communication process. Consequently, the handshake must involve these steps.

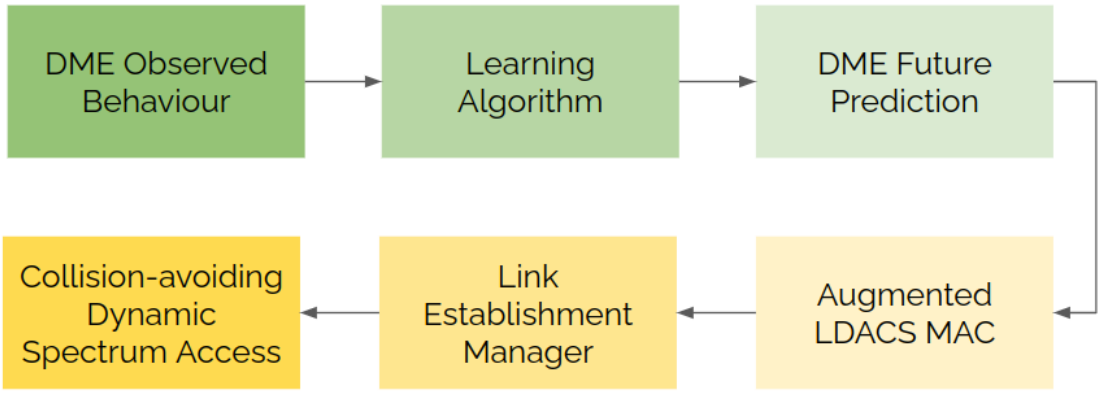
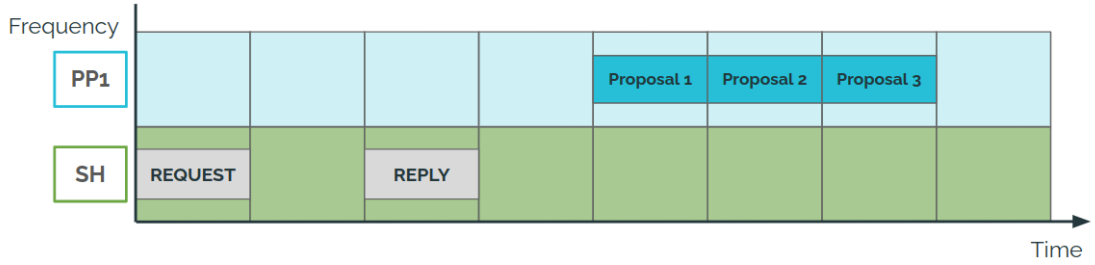



Figure 4.5: Augmented LDACS MAC overview.

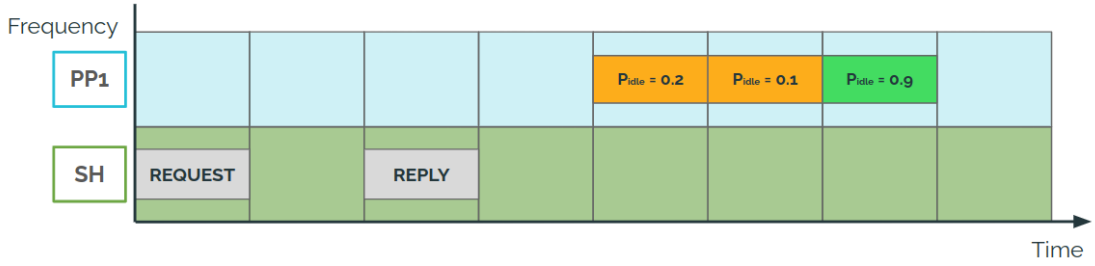
### 4.2.2 Handshake

The first and most important changes happen in the handshake. In the non-augmented MAC, the handshake was as depicted in Figure 3.10. In the Augmented MAC, the handshake has new information to take into account, that is, the predicted behavior of DME. It is locally predicted by all aircraft involved in the communication process. Thus, it does not only check if the time slots are available for LDACS, but it also checks if there is a high probability of them not being used by DME.

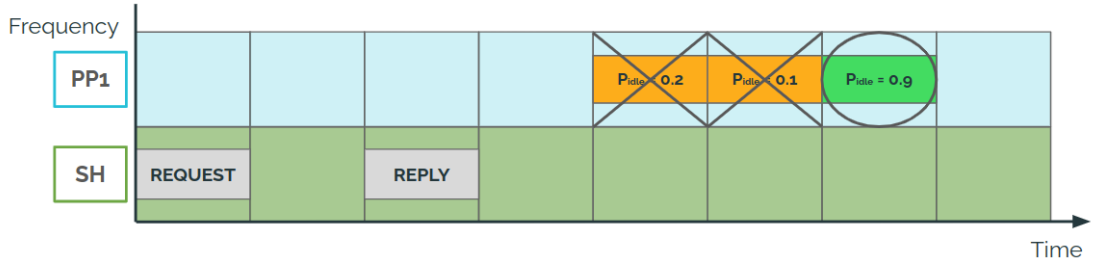
The augmented handshake starts like the old one, with time slots being proposed for the link establishment (see Figure 4.6a). The next step is to check the probabilities of these time slots being idle based on the predicted future DME behavior (see Figure 4.6b). Lastly, the chosen time slots are those that are both available for LDACS and have a high probability of being idle from DME access (see Figure 4.6c).




(a) Time slots proposal.




(b) Checking the probability of the time slots being idle.

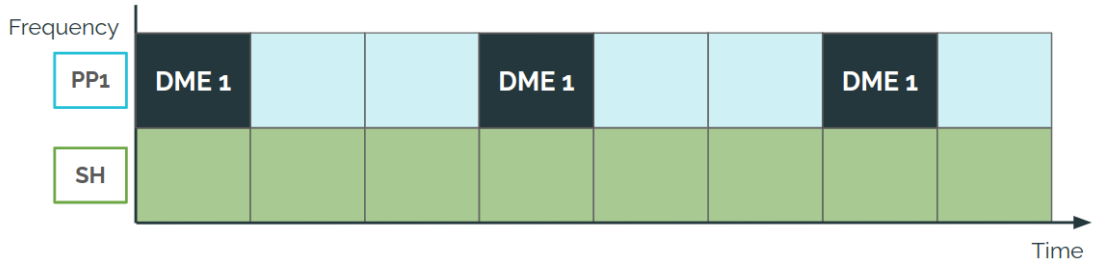



(c) Chosen time slots.

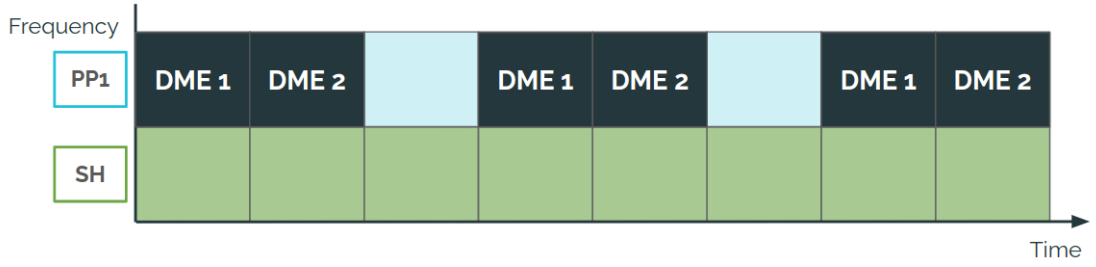
Figure 4.6: Example of the handshake process.

Nevertheless, both communicating aircraft must be involved in the decision process when it comes to evaluating the probability of these time slots being idle. This is done in order to avoid the "hidden node" problem. This problem arises when one aircraft does not detect a transmitting DME user in proximity and is therefore not able to take it into account when predicting the DME medium access behavior in the future, causing a collision. A more graphic example can be seen in Figure 4.7.

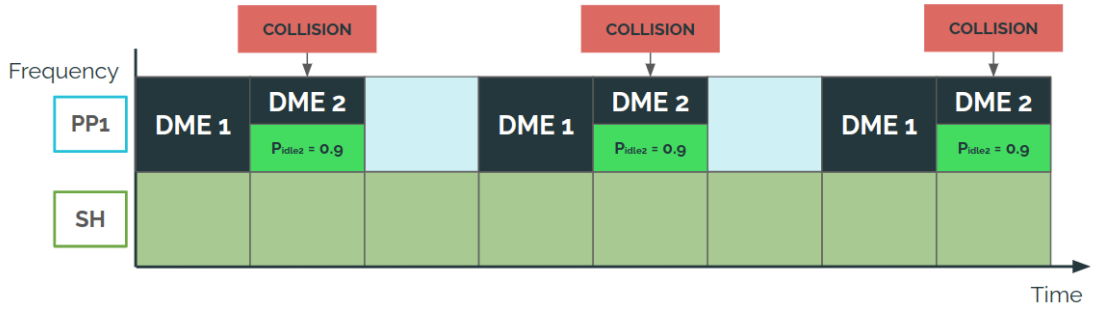
One LDACS user detects one DME user, but fails to detect a second DME user in proximity (see Figure 4.7a). Another LDACS user can detect both DME users due to its positioning (see Figure 4.7b). If only the first LDACS user was involved in the decision process, a wrong prediction would have been used and LDACS would have collided upon the "hidden" DME user (see Figure 4.7c).




(a) One LDACS user detects one DME user.




(b) Another LDACS user detects two DME users.




(c) The problem in the detection of DME users causes a collision.

Figure 4.7: Hidden node problem.

This is the reason why both LDACS users share their predictions along with the proposed time slots (see Figure 4.8).

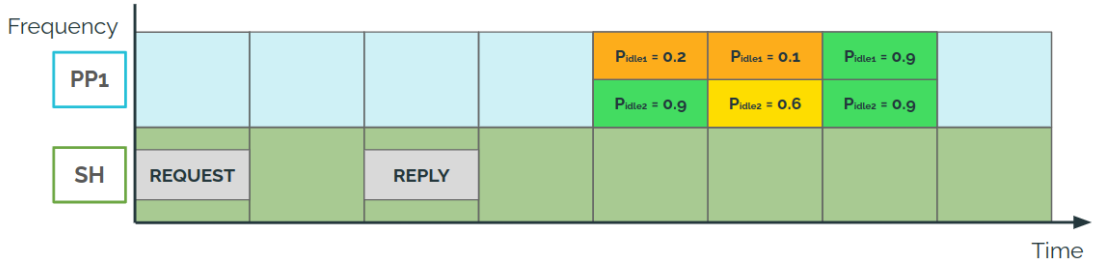



Figure 4.8: Sharing probabilities of being idle for every proposed time slot.

When comparing the probabilities, the lowest one is chosen in order to avoid false positives. This probability must be above a certain threshold $T_t$ that should be determined depending on the situation.

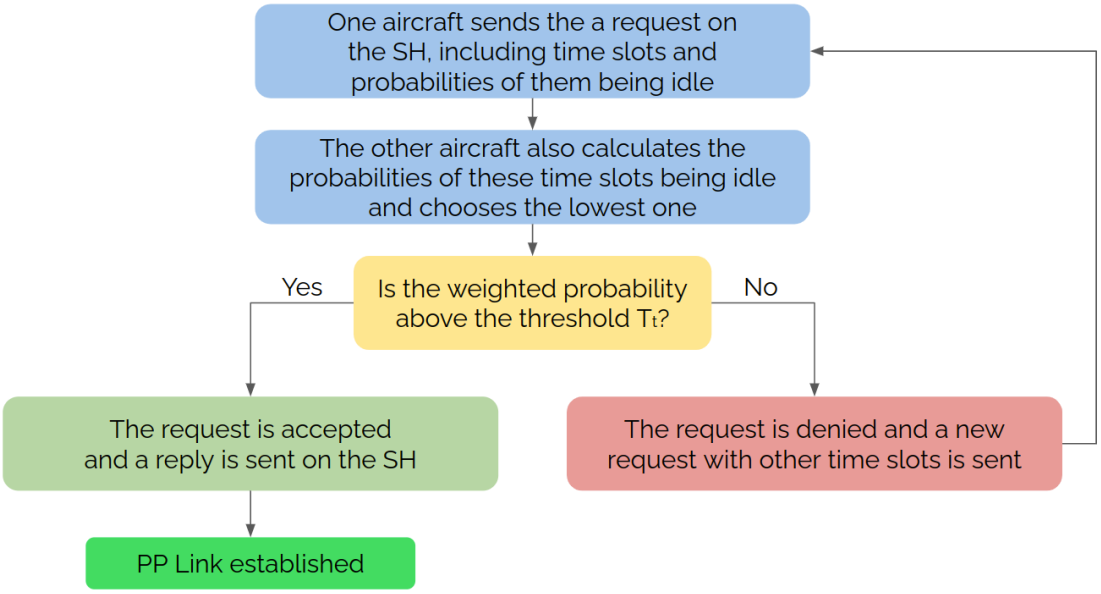



Figure 4.9: Handshake Workflow.

### 4.2.3 Dynamic Threshold

The question of where to set the threshold is not an easy one to answer. Authors have proposed the use of an adaptive threshold in [34]. This means that the threshold is constantly being updated in order to take into account the target collision rate and the actual collision rate. Should the actual collision rate be higher than the target collision

rate, the probability threshold must be stricter. Should the actual collision rate be much lower than the target collision rate, the probability threshold can be relaxed in order to avoid false negatives and not waste resources.

However, this kind of adaptive threshold needs the actual collision rate as an input. Due to the LDACS receiver being blind during an ongoing transmission, it would be impossible to detect the collisions upon DME, making the necessary information for the adaptive threshold unavailable. Hence, a more detailed study of possible fixed or adaptive thresholds must be performed in the future.

# 5 Implementation

This chapter describes the implementation steps that have been followed throughout the thesis. It consists of three main parts: communication simulations (DME implementation and LDACS adaptation), the learning algorithm and the TCP connection between both (see Figure 5.1).

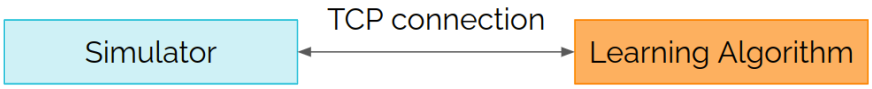



Figure 5.1: Implementation description.

The simulations were built over an existing project, the Inter Aircraft Network [25]. This project is financed by the *Bundesministerium für Wirtschaft und Klimaschutz*, and it comprises the description and simulation of LDACS. The protocol stack can be seen in Figure 5.2, and it includes:

- Fragmentation, Concatenation and Reassembly;
- Automatic Repeat Request;
- Medium Access Control (MAC).

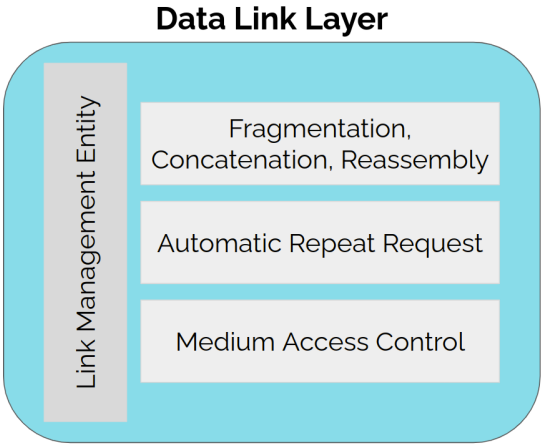



Figure 5.2: Data Link Layer Protocol Stack.

This thesis is solely focused on the MAC, and hence is where all the changes and adaptations have taken place.

The simulations of LDACS are done with the OMNeT++ simulator, along with the INET library. OMNeT++ is an object-oriented discrete event network simulation framework used to the modeling, validation and evaluation of wired and wireless communication systems as well as other areas such as queuing networks, distributed hardware systems et cetera [35]. INET is a model library for OMNeT++ that provides protocols, agents and other models in order to simulate communication networks. For this thesis, the OMNeT++ version used was v5.6.2, and the INET version used was v4.2.5.

## 5.1 DME Implementation

The DME communication system and its behavior was coded as a new configurable module integrated into the LDACS modules from [25]. It consists of two main parts: one for the aircraft and one for the ground stations. These mimic the real-life behavior of DME as described in the literature [20]–[22] (see Figure 5.3).

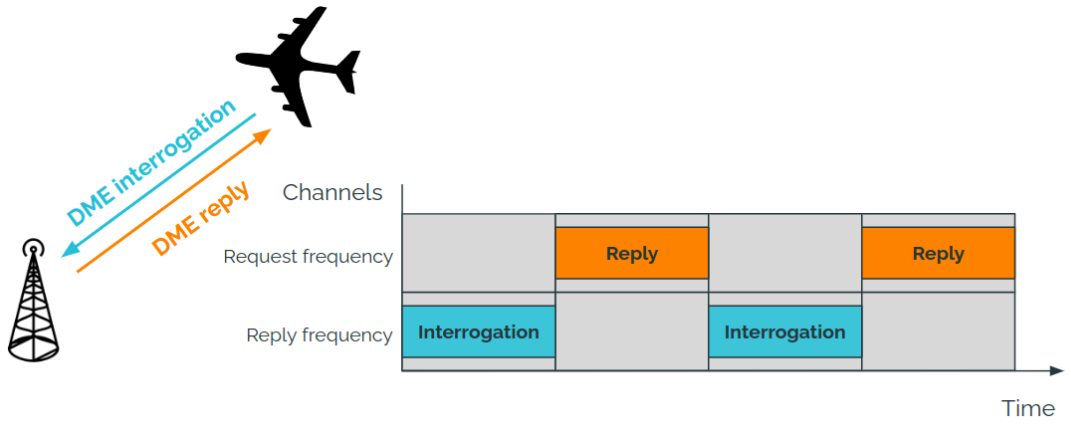



Figure 5.3: The module mimics the real-life DME behavior.

The module functions according to the following steps, and as seen in Figure 5.4:

1. The aircraft schedules a request to be sent.
2. The request is sent and, after a certain transmission delay, it is received by the ground station. There, it is processed in order to differentiate DME requests from interfering communication systems, i.e., LDACS packets.
3. Once the packet has been processed, the ground station schedules a reply to be sent after the stipulated message delay (50 µs).

4. The reply is sent and, after a certain transmission delay, it is received by the aircraft where it is processed in order to differentiate DME replies from interfering communication systems, i.e., LDACS packets.

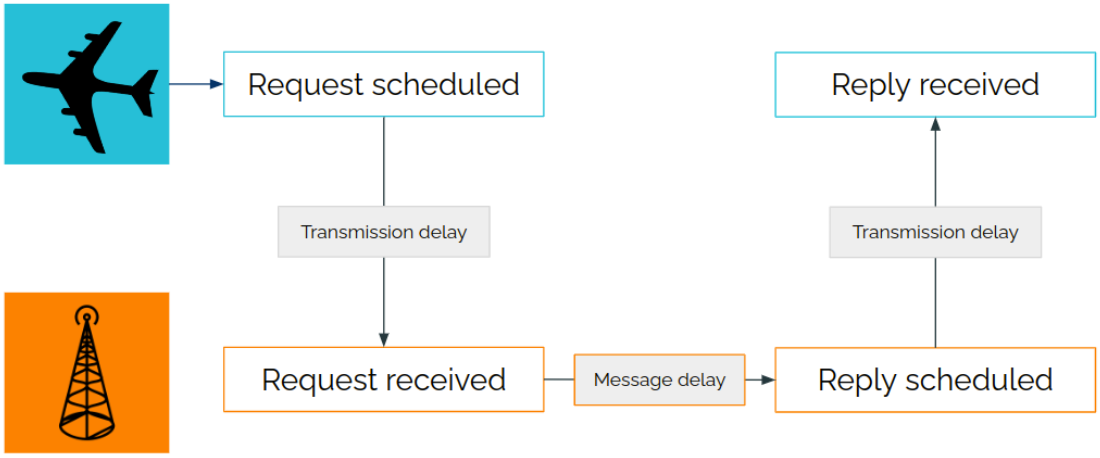



Figure 5.4: Overview of the DME OMNeT++ implementation.

Should any LDACS transmission be present during a DME transmission, this DME signal will not be received, it will be lost. This represents how destructive LDACS interferences are on DME.

## 5.2 LDACS MAC Adaptation

The second part of the OMNeT++ simulations was to modify the provided LDACS MAC, and adapt it to the changes described in Chapter 4.

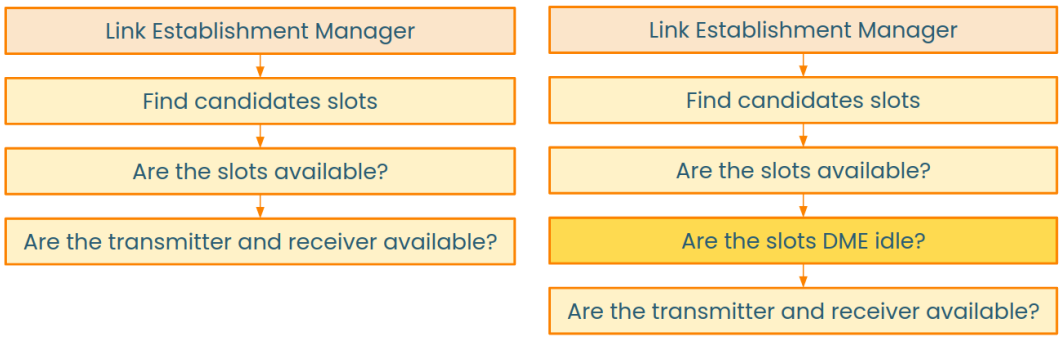
Firstly, the channel sensing was added. Once every time slot, LDACS senses all channels and collects information about the DME behavior. This information is put into a vector and sent over to the learning algorithm, where it is processed. Then, the predictions for all the sensed frequency channels for the next $\tau$ time slots are returned. These predictions are sent to the Link Establishment Manager to be used.

Previously, the Link Establishment Manager acted according to the following steps in order to choose the link establishment time slots, as in Figure 5.5a:

1. Finding candidate slots, that is, possible time slots in which to transmit.
2. Checking these time slots for availability, in other words, checking if those time slots are idle or if they are already allocated for some other transmission.
3. Lastly, checking that both the transmitter and the receiver are available for their respective parts of the communication link.

4. If all these checks are positive, the earliest available time slots are sent back to the Link Establishment Manager in order for the communication link to be established.

For the augmented process, an extra step was added in order to ensure that the DME predicted behavior is taken into account, as seen in Figure 5.5b.




(a) Previous time slot selection process. (b) Augmented time slot selection process.

Figure 5.5: Time slot selection process comparison.

Now that DME is taken into account, the LDACS MAC needs to be able to adapt to the DME predicted patterns, and therefore to be more flexible than it was before. This has to do with how the time slots are chosen. This process has three main parameters, which can also be seen in Figure 5.6:

- **Start offset** or **start time slot**: it is the first time slot of the transmission.
- **Burst length**: it is the number of transmissions per burst.
- **Burst offset**: it is the number of idle time slots left between each burst.

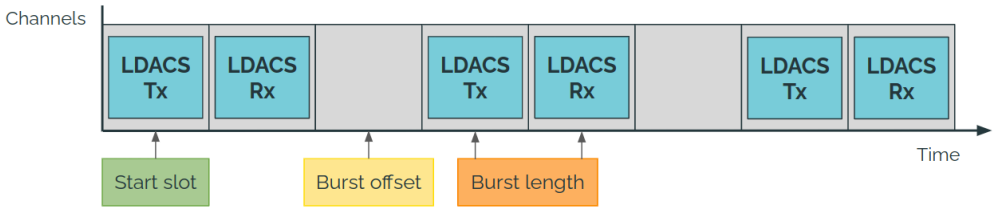



Figure 5.6: Time slot selection parameters.

The burst offset is determined by the number of neighbors, meaning that the more LDACS users there are, the longer the burst offset is. This is done to avoid interferences among each other. As seen in Figure 5.7, a scenario with only two LDACS users would have a burst offset of 0 while a scenario with three LDACS users would have a burst offset of 1.

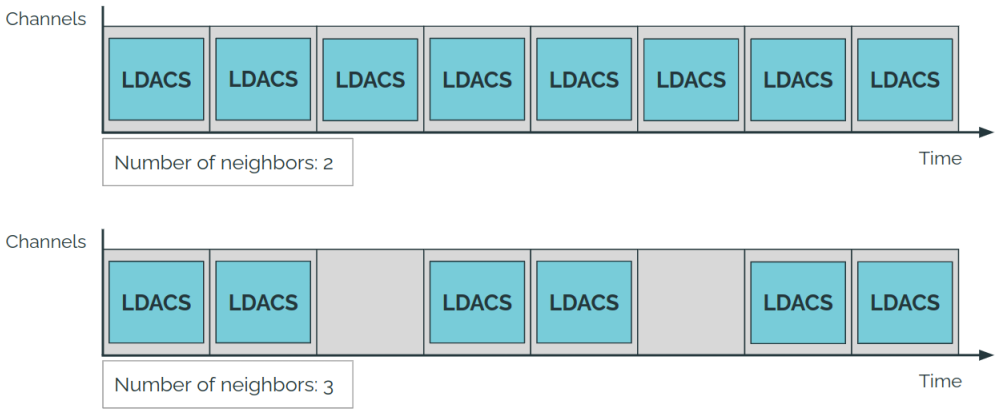
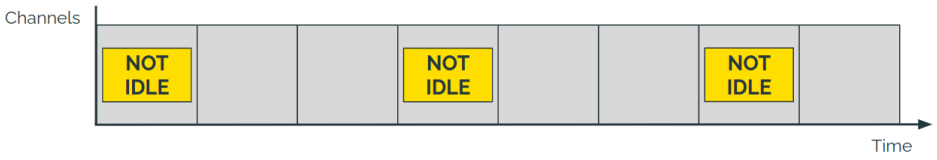
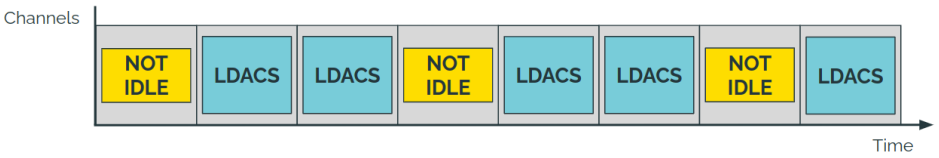



Figure 5.7: Burst offset with respect to the number of LDACS users.

The burst length is fixed and always lasts 2 time slots, one for transmitting and one for receiving. The start offset is the parameter that changes when looking for available time slots to transmit in. An example can be seen in Figure 5.8, where the starting time slot must be the second one instead of the first one in order to avoid a time slot that is not available for transmission.




(a) The first time slot is not available for transmission.

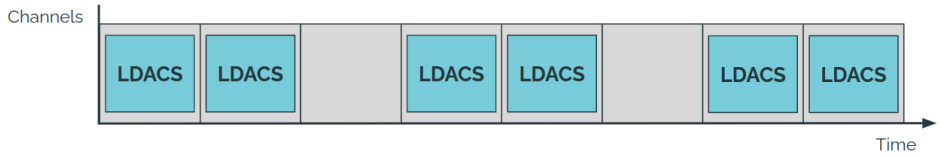



(b) The starting time slot is the second one instead of the first one.

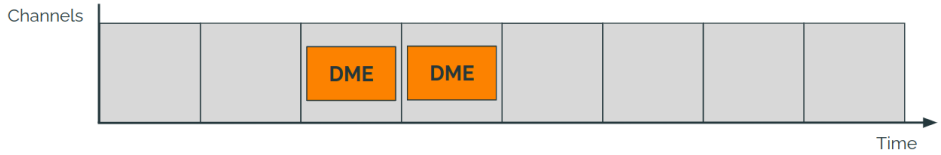
Figure 5.8: Start offset example.

The non-augmented LDACS MAC would only change the start offset when looking for available time slots. This approach proved to be problematic for some DME patterns. An example can be seen in Figures 5.9 and 5.10. It depicts an LDACS slot selection with a burst offset of 1 (see Figure 5.9a), and a predicted DME behavior occupying two consecutive time slots (see Figure 5.9b). The initial LDACS pattern would provoke a collision at the beginning of the second LDACS burst (see Figure 5.10a), whereas changing

the start offset would provoke a collision at the end of the first LDACS burst (see Figure 5.10b). However, by increasing the burst offset by 1 time slot, the DME transmissions can be circumvented, making use of all the available resources while avoiding collisions at the same time (see Figure 5.10c).

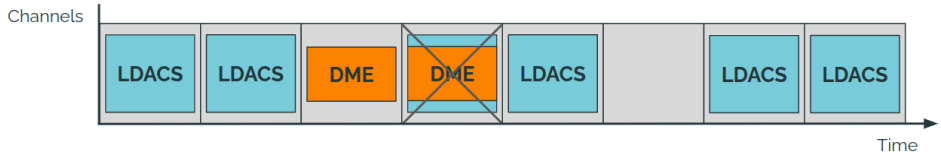



(a) LDACS initial time slot selection.

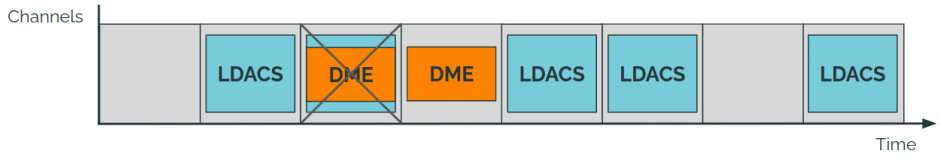



(b) DME behavior prediction.

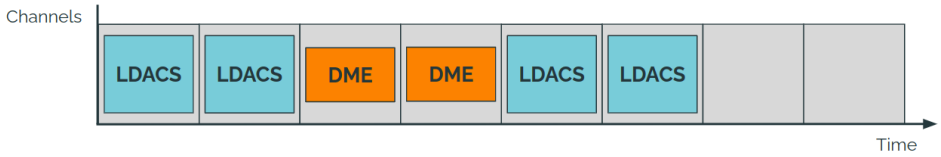
Figure 5.9: Example of initial LDACS time slot selection and DME behavior prediction.




(a) The initial LDACS time slot selection would cause a collision.



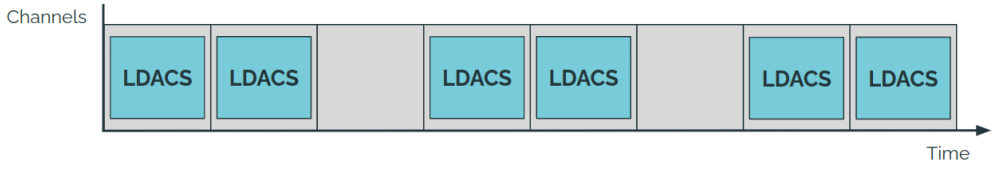

(b) Changing the start offset does not fix the collision.



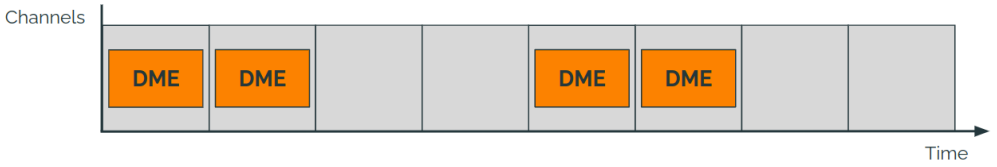

(c) Changing the burst offset fixes the collision.

Figure 5.10: Example of collision avoidance.

The augmented LDACS MAC can now change both the start and the burst offset to adapt to more complicated patterns. Another example can be seen in Figures 5.11 and 5.12. It depicts an LDACS slot selection with a burst offset of 1 (see Figure 5.11a), and a predicted DME behavior occupying two consecutive time slots twice (see Figure 5.11b), so now the DME behavior would collide with both the first and the second LDACS burst (see Figure 5.12a). In order to solve this scenario, both parameters need to be changed, that is, the start offset needs to be increased to 2 time slots in order to avoid the first DME cluster, and the burst offset must also be increased by 1 time slot in order to avoid the second DME cluster (see Figure 5.12b).

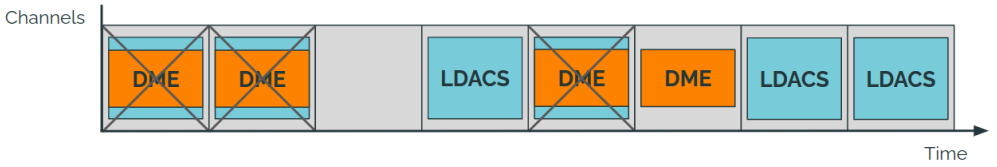



(a) Initial LDACS time slot selection.

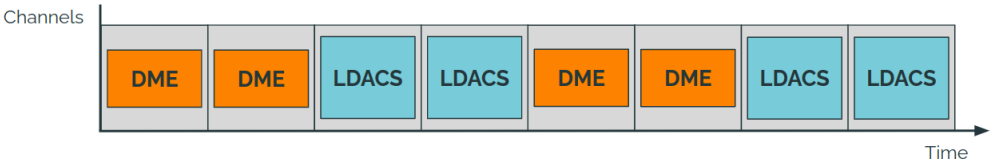



(b) DME behavior prediction.

Figure 5.11: Example of initial LDACS time slot selection and DME behavior prediction.




(a) The initial LDACS time slot selection would cause collisions.




(b) Changing both the start and the burst offset fixes the collisions.

Figure 5.12: Example of collision avoidance.

Thanks to these changes, the augmented LDACS MAC is now able to take into account the predictions given by the learning algorithm.

## 5.3 Learning Algorithm Implementation

The augmented LDACS MAC is now equipped to provide observations and to deal with predictions. The processing of the observations must be done by the learning algorithm, where the observations containing the DME behavior are processed in order to output predictions about the future DME behavior (see Figure 5.13). It is implemented in Python v.3.8.10 using Tensorflow v.2.8.0 and Keras v.2.8.0. Tensorflow is a ML software library that allows to train and build models [36], [37]. Keras is a deep learning API written in Python which runs on top of TensorFlow [38], [39]. It provides the simplicity and abstraction needed for the code to be comfortably implemented in Tensorflow. Additionally, Unittest is used as the testing framework for the evaluation of the project [40].

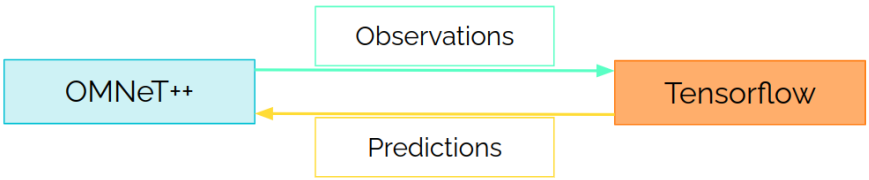



Figure 5.13: OMNeT++ - Tensorflow connection.

More specifically, the observations are sent as a column vector with as many rows as frequency channels are being sensed. This vector is processed by the learning algorithm which outputs a matrix with the future predictions. The size is as many rows as frequency channels were being sensed, and as many columns as time slots in the future are being predicted (see Figure 5.14).

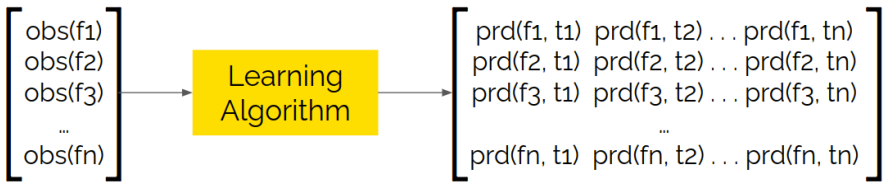



Figure 5.14: Observation vector ("obs") and prediction matrix ("prd") on frequency channels $f_i$ and time slots $t_j$.

An example can be seen in Figure 5.15. The observations can either be a $-1$ if the frequency channel was occupied, or a $+1$ if the frequency channel was idle. The predictions

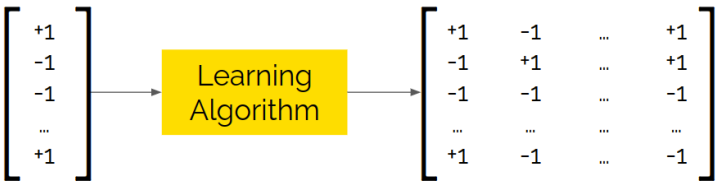



Figure 5.15: Example of observation vector and prediction matrix.

can be any value between $-1$ and $+1$ since they represent a probability. However, in this example only $-1$ and $+1$ are assigned for simplicity.

The predictions are done with an ANN as the one described in Chapter 3, that is, a LSTM RNN. It consists of an input layer with 16 neurons, an output layer with as many neurons as frequency channels, and a hidden middle layer (see Figure 5.16). In this hidden layer there is an activation function $f(\cdot)$, which is a non-linear differentiable transformation. This makes sure that the LSTM RNN is able to learn both linear and non-linear patterns.

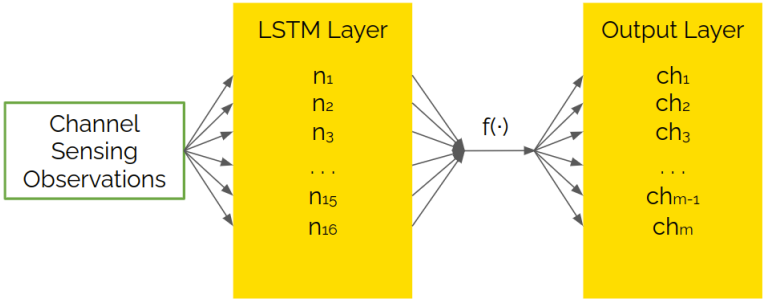



Figure 5.16: LSTM Layers.

The observation vector is fed into the LSTM layer, and the result of this is passed through the activation function and then fed to the output layer. The output from this last layer is then compared to the actual DME behavior in order to obtain the loss function. This is then passed through the Adam Optimizer, which is a state-of-the-art optimization algorithm for stochastic gradient descent for training [41], back into the different layers (see Figure 5.17).

This process provides the first column of the prediction matrix, that is, the prediction for the next time slot. However, in order to provide the LDACS MAC with a prediction matrix that goes $\tau$ time slots into the future, further inference over a certain planning horizon is needed. Therefore, the first prediction column is fed to the LSTM RNN again in order to obtain the second prediction column, and the second one is fed to obtain the third, and so on recursively until the desired number of columns in the matrix has been achieved (see Figure 5.18).

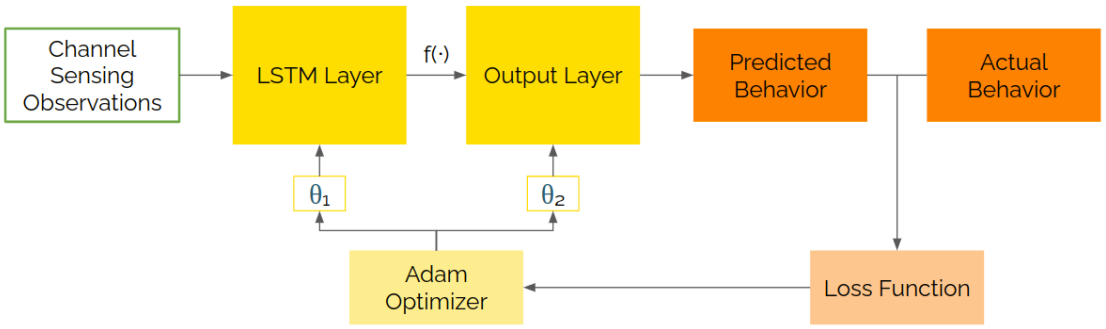



Figure 5.17: LSTM architecture.

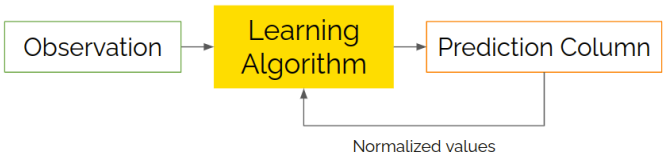



Figure 5.18: Inference.

Before being re-fed into the learning algorithm, the prediction columns are normalized, that is, the values are set to $\pm 1$. If the prediction is a positive value, it is set to $+1$. On the other hand, if it is 0 or a negative value, it is set to $-1$. This is done for the learning algorithm to treat the predictions like observations again, and as seen in 5.15, the observations can only be $+1$ or $-1$.

All of these inference results are put together into what has been referred to as the prediction matrix, which is then passed back to the OMNeT++ simulator for the Link Establishment Manager to work with it.

## 5.4 TCP Connection

The last step of the implementation was to establish the TCP connection between the OMNeT++ simulations and the TensorFlow learning algorithm. This was accomplished using Veins-Gym, a project that exports simulations as AI Gyms [42]. An AI Gym allows for the model to be trained outside the actual simulations. In the case of this project, it allows for the DME behavior observations to be sent from OMNeT++ to the learning algorithm, and for the predictions to be sent back to OMNeT++ in order for them to be used in the LDACS MAC (see Figure 5.19).

Veins-Gym is a piece of research software developed by the Telecommunication Networks Group of TU Berlin, and it was initially designed for vehicular networks. Its code has therefore been adapted and re-engineered to serve the purposes of this project.

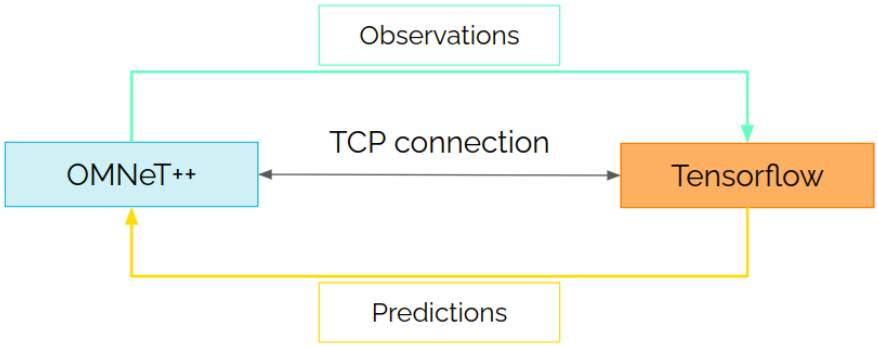



Figure 5.19: TCP Connection.

The other TCP connection alternative would have been to compile the RNN into C++. However, it was decided not to pursue this concept on the basis that the C++ interface for Tensorflow is very limited and unflexible. This would have coerced the expected extents of this project and would have yielded more limited results than the ones hereby presented.

## 5.5 Exemplary time step

In order to summarize this chapter, an exemplary time step will be followed. All the values and behaviors in this section are hypothetical, and are only meant to show the functioning of the whole system.

DME starts by transmitting. In the example in Figure 5.20, it occupies two frequency channels at different time slots.

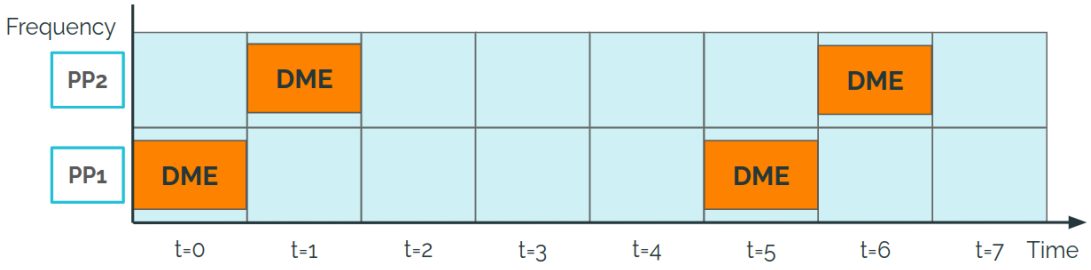



Figure 5.20: Example: DME transmission.

When LDACS performs the channel sensing, it will sense the two frequency channels every time slot, so the observations will look like $\begin{bmatrix} PP_2 \\ PP_1 \end{bmatrix}$. At $t = 0$ the observation vector

will be $\begin{bmatrix}0\\1\end{bmatrix}$, at $t = 1$ the observation vector will be $\begin{bmatrix}1\\0\end{bmatrix}$, $t = 2$ to $t = 4$ will have a $\begin{bmatrix}0\\0\end{bmatrix}$ observation vector and then the pattern repeats.

This information is passed each time slot from OMNeT++ to Tensorflow through the TCP connection (see Figure 5.21).


Figure 5.21: Example: TCP Connection for observations.

When it reaches TensorFlow, the model is trained on the observation vector, and it outputs a prediction for the upcoming time slot. This first prediction is the input for the inference over the planning horizon part of the code, which outputs the prediction for the second upcoming time slots. Using this second prediction, the prediction for the third upcoming time slot is calculated, and so on until the prediction matrix is big enough. Once the model is fully trained, for an observation vector of $\begin{bmatrix}0\\1\end{bmatrix}$ the prediction matrix will be:

$$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots\end{bmatrix}$$

This prediction matrix is sent back from Tensorflow to OMNeT++ through the TCP connection (see Figure 5.22), where the Link Establishment Manager will use it to establish an LDACS PP link.

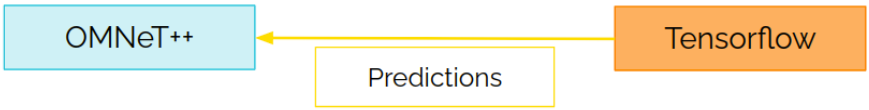
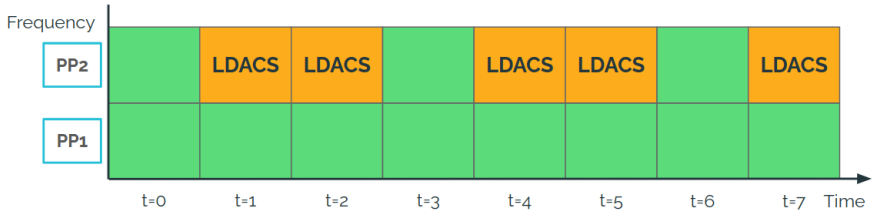
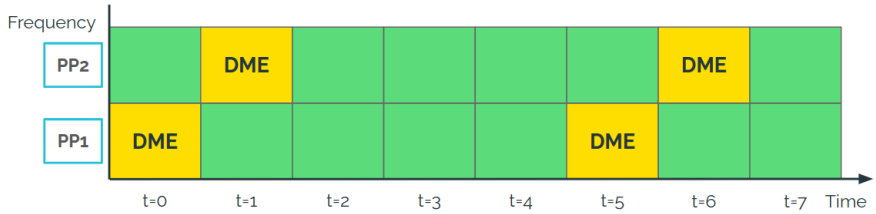



Figure 5.22: Example: TCP Connection for predictions.

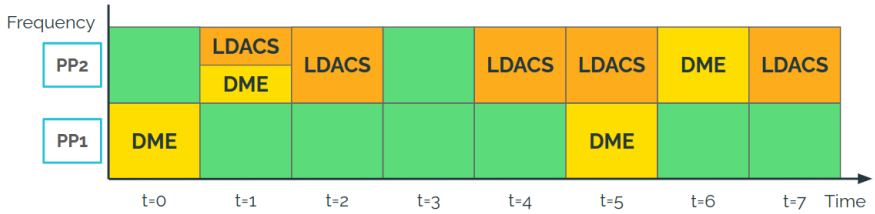
The Link Establishment Manager chooses a set of candidate slots to transmit in and checks the LDACS availability during these slots (see Figure 5.23a), the DME availability (see Figure 5.23b) and only chooses the ones that are available for both (see Figure 5.23c).




(a) LDACS available slots (in green) and occupied slots (in orange).




(b) DME available slots (in green) and occupied slots (in yellow).




(c) Collective available slots (in green) and occupied slots (in orange and yellow).

Figure 5.23: Available and occupied slots.

It then goes through all the combinations of start time slots and burst offsets until it finds one that fits the LDACS MAC requirements and uses only the DME available time slots in one frequency channel only. In this case, that combination is in $PP_1$ with start time slot $t = 1$ and with a burst offset of 3 time slots (see Figure 5.24).

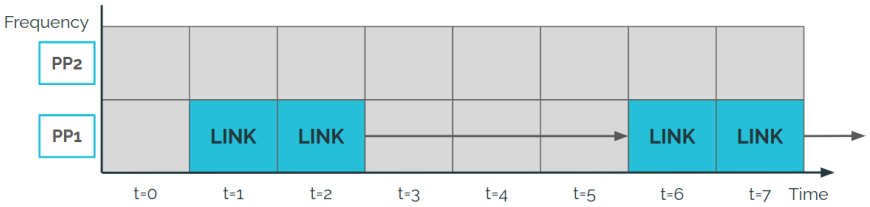



Figure 5.24: Example: chosen slots for LDACS transmission (in blue).

This pattern for LDACS is kept until the timeout is reached. Once this happens, the whole process is repeated from the beginning.

# 6 Analysis

Following the implementation, the results were observed and analyzed. They are discussed in this chapter.

## 6.1 Non-augmented LDACS MAC

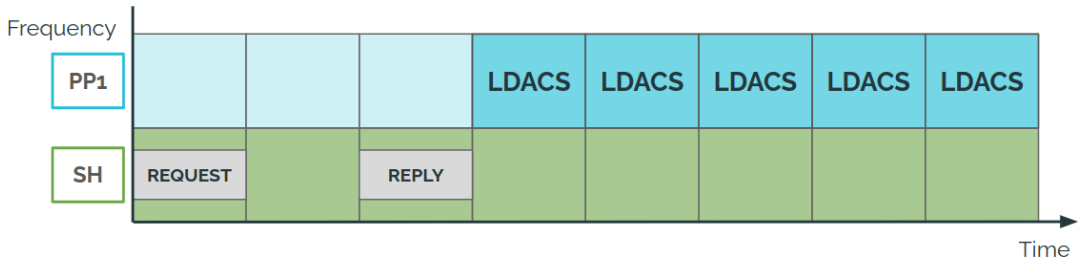
The non-augmented LDACS MAC is the previous design of the LDACS MAC and cannot take into account the behavior of DME when transmitting. Therefore, there are two ways in which it can behave:

### 6.1.1 Native Approach

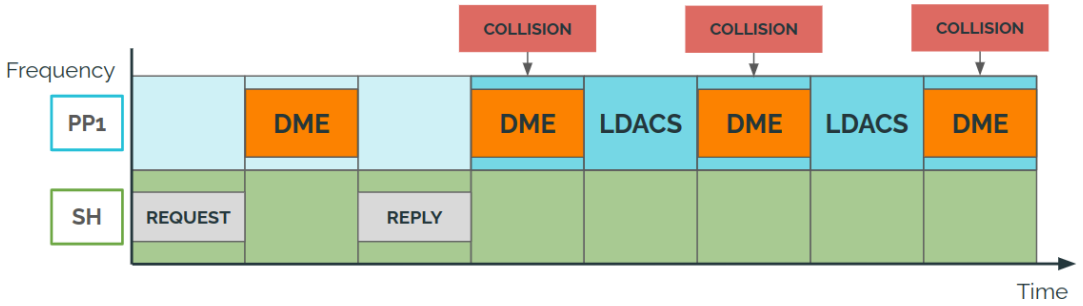
The most basic way in which LDACS can behave is the "native approach", that is, it has no information about the DME behavior, and it does not try to get any. This is of course a very disruptive approach, and it causes a lot of collisions.

The metric that was chosen to evaluate the simulations was not the absolute number of collisions, but the **collision rate** on DME. This is calculated dividing the number of received DME replies by the number of sent DME requests, that is, checking how many packets are lost due to collisions with LDACS and calculating the rate at which this happens. For this approach, the number of collisions depends on the burst offset. As seen in Figure 5.7, the number of time slots between bursts depends on the number of LDACS neighbors. Therefore, for the case of 2 LDACS users transmitting, the burst offset would be zero (see Figure 6.1a) and the collision rate on DME during the LDACS PP transmission would be of 100% (see Figure 6.1b).

However, and as seen in Figure 6.1b, the PP transmission does not occupy 100% of the time. In order to see how much it actually occupies, one whole link establishment plus transmission will be taken into account, which lasts about 90 time slots. Out of these, 14 are used for the link establishment (assuming that resources are found without a problem) and the remaining 86 are used for the PP transmission. Therefore, 100% collision rate would only happen for 86% of the time, resulting in an analytical result of 86% collision rate on DME overall. The simulated results reflect these numbers, as can be seen in Figure 6.2.




(a) Burst offset of 0 time slots.




(b) Collision rate on DME.

Figure 6.1: Native approach, 2 LDACS users.

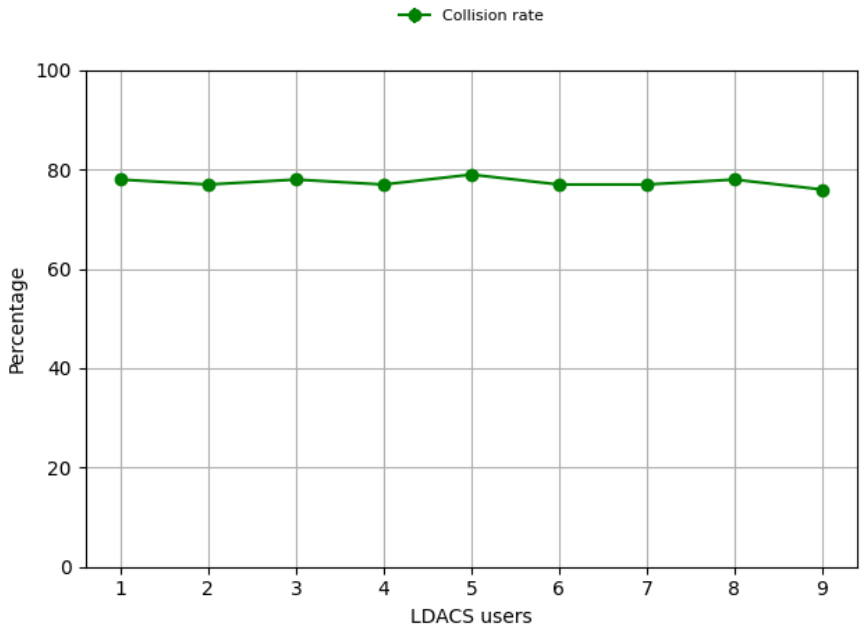
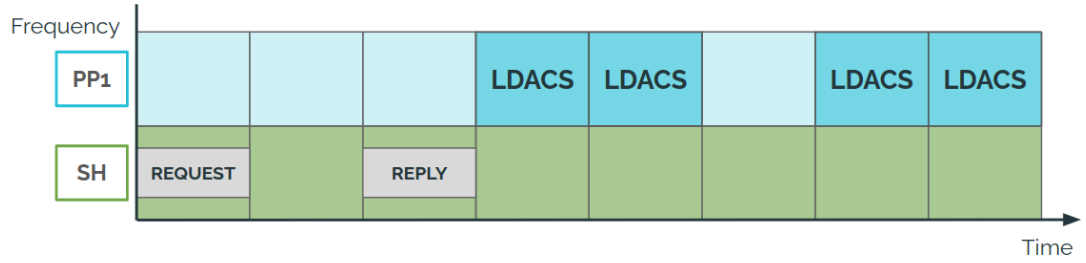


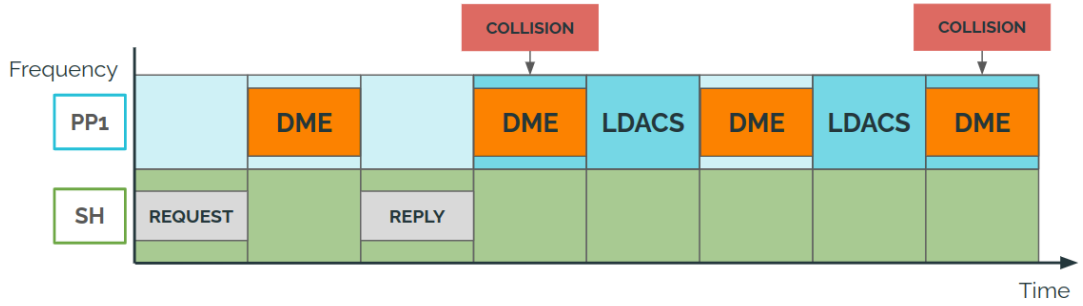

Figure 6.2: Simulated collision rate on DME for 2 LDACS users.

For a variable number of DME users between 1 and 9, the collision rate stays stable around 80%. It is slightly under the analytical expectations, but this is due to the fact that the link establishment process is not always accomplished in one try. Even without taking DME into account it can last longer than the assumed 14 time slots due to LDACS itself, therefore taking a longer percentage of the link establishment time without causing collisions.

For a growing number of LDACS users, the collision rate decreases. This is due to how the burst offset is calculated (see Figure 5.7). For more than 2 LDACS users, the burst offset would increase (see Figure 6.3a), decreasing thus the changes of colliding with a DME transmission (see Figure 6.3b).




(a) Burst offset of 1 time slots.

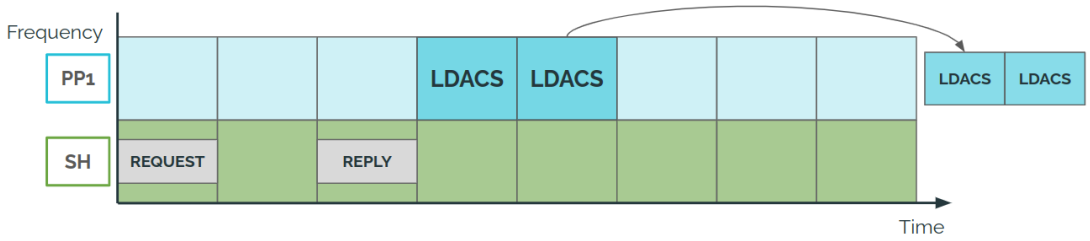



(b) Collision rate on DME.

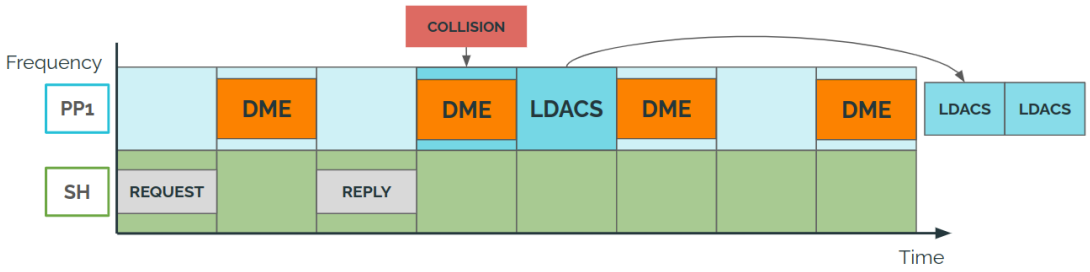
Figure 6.3: Native approach, LDACS users $> 2$.

The more LDACS users there are, the larger the burst offset is (see Figure 6.4a), and the more the chances of collision with DME decrease (see Figure 6.4b).

The different burst offsets and consequent collision probabilities are reflected in the simulated results as shown in Figure 6.5.




(a) Burst offset of over 3 time slots.




(b) Collision rate on DME.

Figure 6.4: Native approach, LDACS users $\gg 2$.

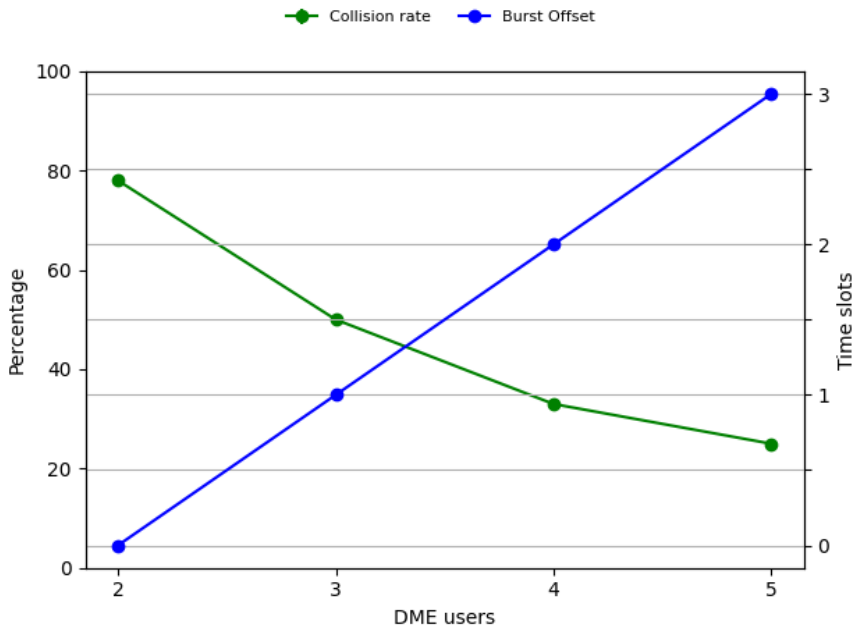


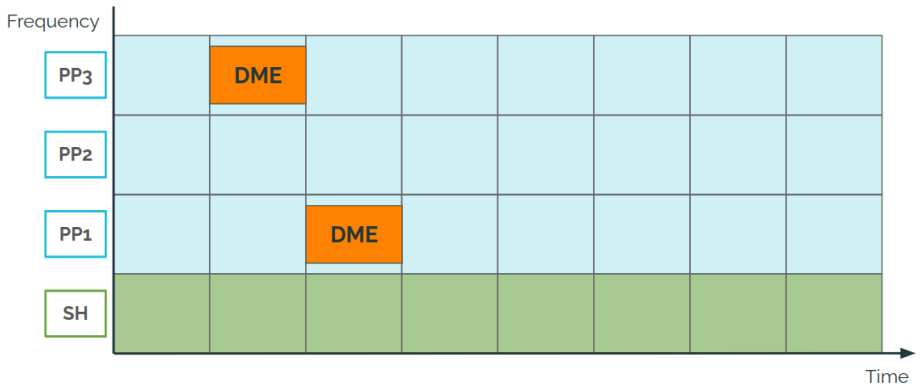

Figure 6.5: Collision rate on DME and burst offset for several LDACS users.

The collision rate on DME starts at around 80% for 2 LDACS users and decreases down to around 30% for 5 LDACS users while the burst offset starts at 0 time slots for 2 LDACS users and goes up to 3 time slots for 5 LDACS users, as expected.

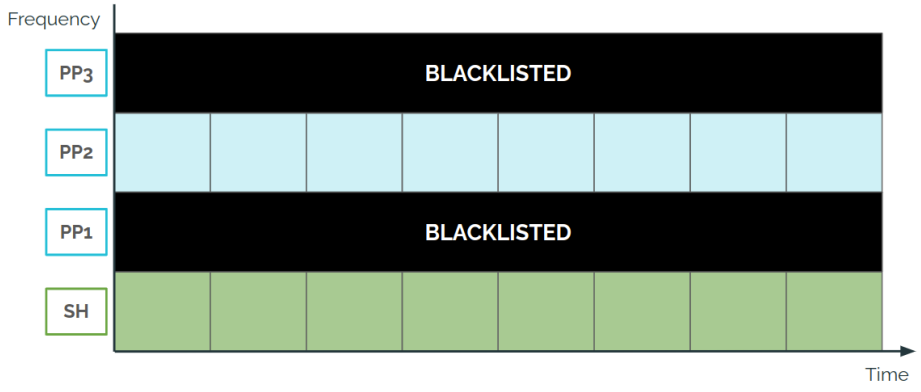
In conclusion, the collision rate is variable between 20% and 80% for 2 to 5 LDACS users, but even at its lowest these numbers are extremely high, and even more taking into account that DME is a safety critical system and should under no circumstances be interfered with. A different approach is required.

### 6.1.2 Blacklisting Approach

The next approach and the one that is currently used by the LDACS MAC is channel blacklisting. This means that if DME is known to use one channel (see Figure 6.6a), LDACS will blacklist this channel entirely, and it will not be used under any circumstances (see Figure 6.6b).




(a) DME-used channels.

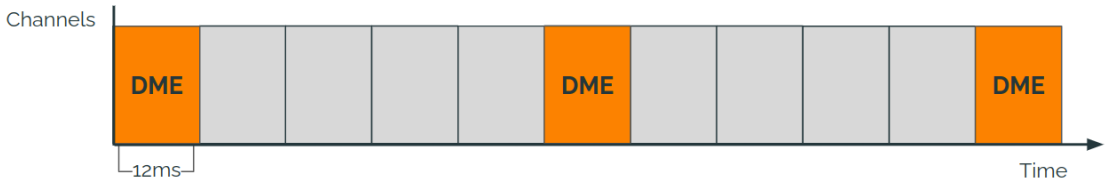



(b) Blacklisted channels.

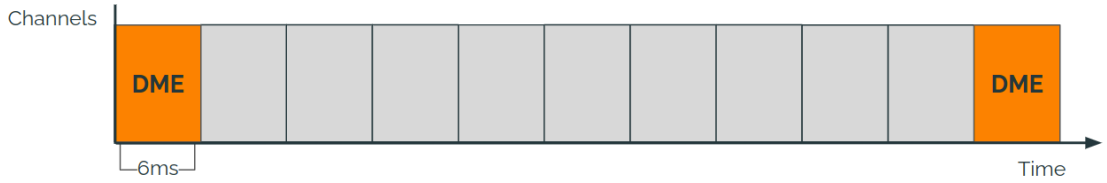
Figure 6.6: Blacklisting Approach.

This approach is a very safe one and yields no collisions at all. However, it is problematic in the sense that it wastes entire frequency channels that could be taken advantage of otherwise.

DME transmits 5 to 15 ppps on track mode, which is every 60 ms to 200 ms [21]. Looking firstly at the worst case scenario, where DME transmits the most, there is a DME packet every 60 ms. Assuming 12 ms-long LDACS time slots, there is one DME transmission every 5 time slots, which yields a utilization of 20% (see Figure 6.7a). Assuming 6 ms-long LDACS time slots, there is one DME transmission every 10 time slots, which yields a utilization of 10% (see Figure 6.7b).




(a) 12 ms LDACS time slots.

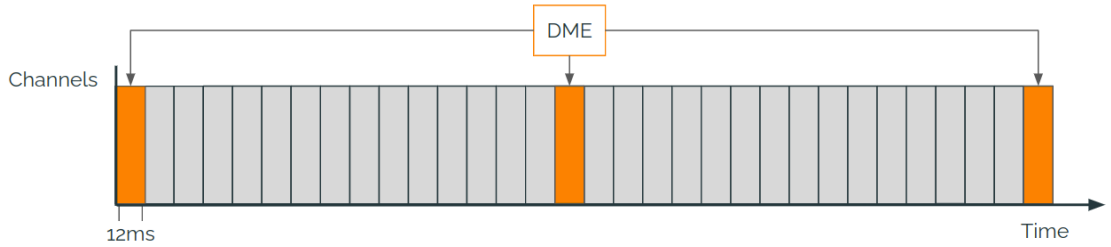



(b) 6 ms LDACS time slots.

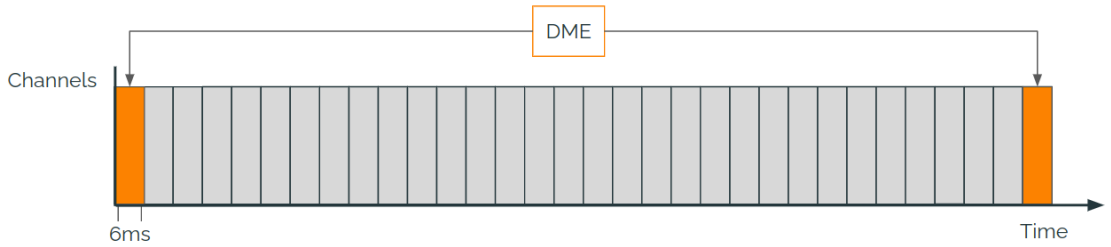
Figure 6.7: Worst case scenario DME channel utilization.

When looking at the best case scenario, where DME transmits the least, there is a DME packet every 200 ms. Assuming 12 ms-long LDACS time slots, there is one DME transmission approximately every 16 time slots, which yields a utilization of 6.25% (see Figure 6.8a). Assuming 6 ms-long LDACS time slots, there is one DME transmission approximately every 32 time slots, which yields a utilization of 3.125% (see Figure 6.8b).

The presented percentages are only for one DME user. More users will likely increase the DME channel utilization. That is not guaranteed though, as several DME users might be transmitting within the same LDACS time slot. As seen in Section 3, the length of a DME request is around 20 µs for one pulse pair. The waiting time and the reply are not counted towards this since they happen on another frequency channel (differing 63 MHz from the request channel). These 20 µs represent 0.33% of a 6 ms LDACS time slot or 0.166% of a 12 ms LDACS time slot (see Figure 6.9). These values mean that one 6 ms LDACS time slot could host up to 300 DME users, and one 12 ms LDACS time slot up to 600 DME users.




(a) 12 ms LDACS time slots.




(b) 6 ms LDACS time slots.

Figure 6.8: Best case scenario DME channel utilization.

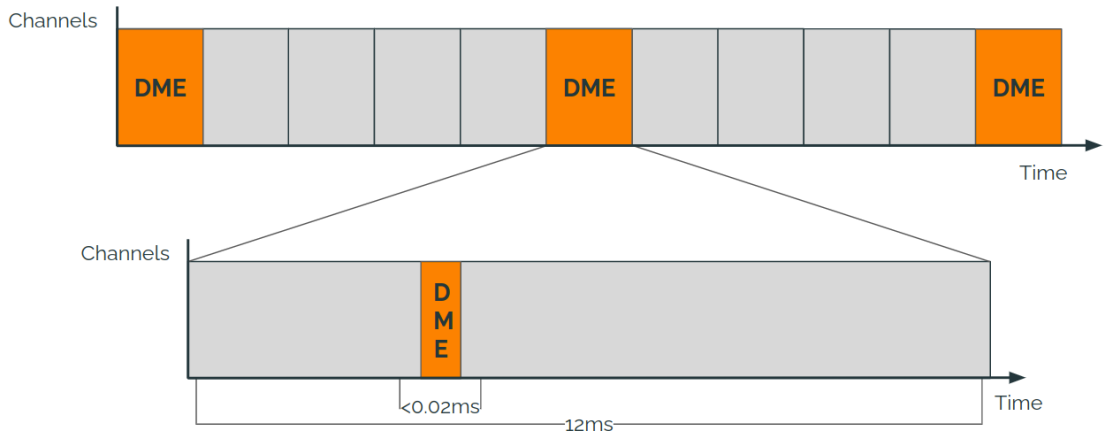
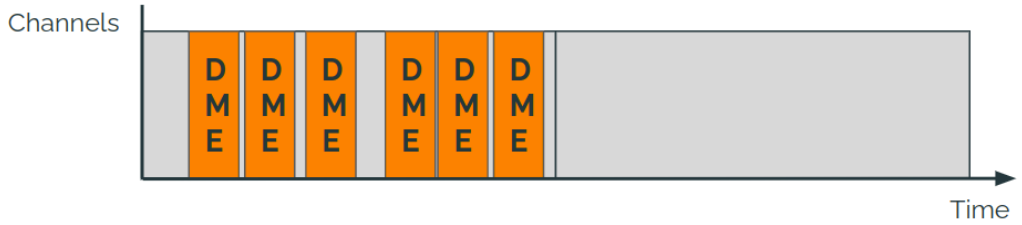



Figure 6.9: DME time slot utilization.

Depending on how the DME users are transmitting, there is the possibility that 2 DME users will transmit in two separate LDACS time slots (see Figure 6.10a), but also the possibility that many DME users will transmit in the same LDACS time slot (see Figure 6.10b).


(a) Two DME users on two LDACS time slots.




(b) Several DME users on one LDACS time slot.

Figure 6.10: DME users with respect to LDACS time slots.

The probability of this happening has already been reviewed in Section 4.1, but this comes to show that even for a larger number of DME users, the channel utilization rate in terms of LDACS time slots could still be as low as the percentages shown above, and that an increased number of DME users does not necessarily mean a decrease in the number of available time slots.

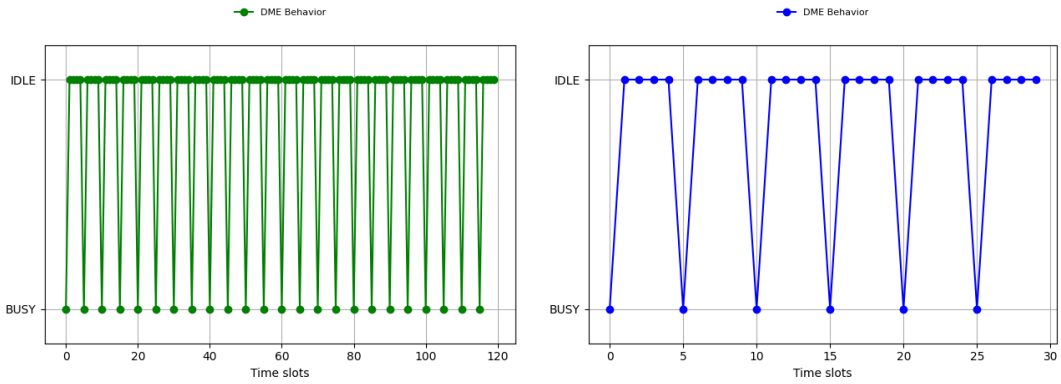
Moreover, not only are the utilization rates low, but this approach also blacklists channels that *could* be used by DME without checking if they are used at all. Therefore, channels with 100% availability in all their time slots could be automatically ignored, hence wasting even more resources.

In conclusion, if the DME channel utilization (for one user) ranges between 3.125% and 20%, LDACS could take advantage of 80% to 96.875% of the DME licensed channels. Consequently, it can be concluded that the "Blacklisting Approach", while safe, is extremely wasteful and wastes up to almost 97% of the available time slots in the worst case scenario.

## 6.2 Augmented LDACS MAC

The inadequacy of both aforementioned approaches results in the need for the augmented LDACS MAC. The results presented here are the results from the implementation described in Chapter 5.

The first step is for the RNN to learn the DME behavior. The initial simulations used one transmitting DME user every 60 ms, so every fifth time slot (see Figure 6.11).




(a) DME Behavior over 120 time slots. (b) DME Behavior over 30 time slots.

Figure 6.11: Simulations of the DME Behavior.

The code was successfully able to learn the pattern as shown in Figure 6.12.

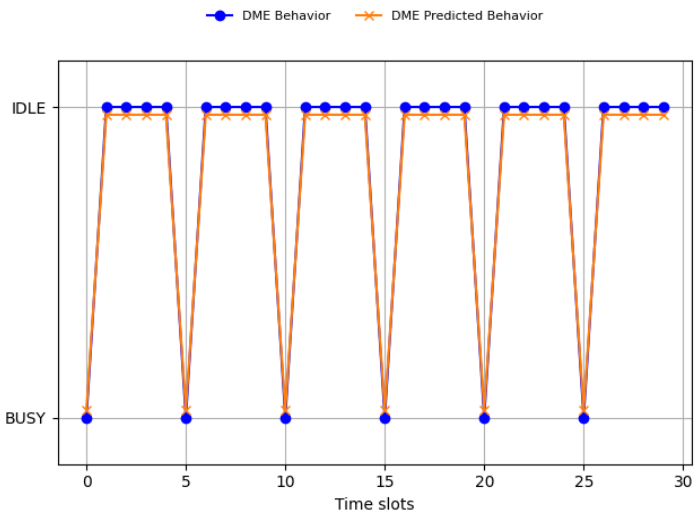



Figure 6.12: Predictions accuracy.

How well the pattern was learned was not only proven with Figure 6.12, but also calculated using the accuracy: the predictions at time $t$ are compared with the channel sensing observations from $t+1$ which yields the accuracy at time $t+1$, the predictions at time $t+1$ are compared with the channel sensing observations from $t+2$ which yields the accuracy at time $t+2$, and so on for as long as the simulation lasts. These values are not only used in order to check that the project is succeeding, but for the learning algorithm to train itself and know which predictions are correct and which ones are incorrect. As can be seen in Figure 6.13, the accuracy goes to 1 after less than 100 time slots, i.e. 1.2 s. Once the accuracy reaches 1, the pattern is learned and there are no more incorrect predictions. The other parameter that can be seen in Figure 6.13 is the loss, which is the input to the Adam Optimizer described in Section 5.3.

Once that it is clear that the learning algorithm is learning the pattern without a problem, this information was input to the adapted LDACS MAC.

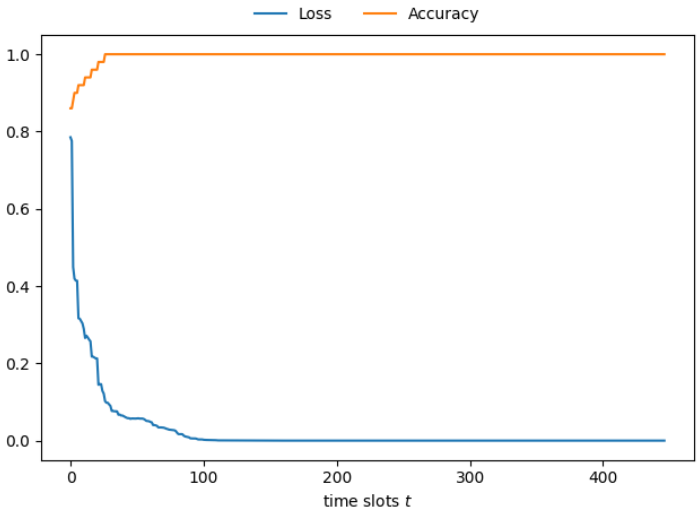



Figure 6.13: Accuracy and losses.

The first simulation that will be presented evaluates the capacity of LDACS to determine which frequency channels are occupied by DME and which ones are not. It was done using 10 frequency channels, out of which 2 were used by DME users and 8 were idle. The LDACS users were set to start checking the DME-occupied channels first, but were expected to quickly learn which channels were used and which were never used. This was achieved, and the links were successfully established without causing interferences.

The first set of results can be seen in Figure 6.14, where the collision rate is stable around 1.25%. This slight error is due to the first 50-100 time slots, in which the model is still being trained and, therefore, mistakes are still being made and some collisions occur.

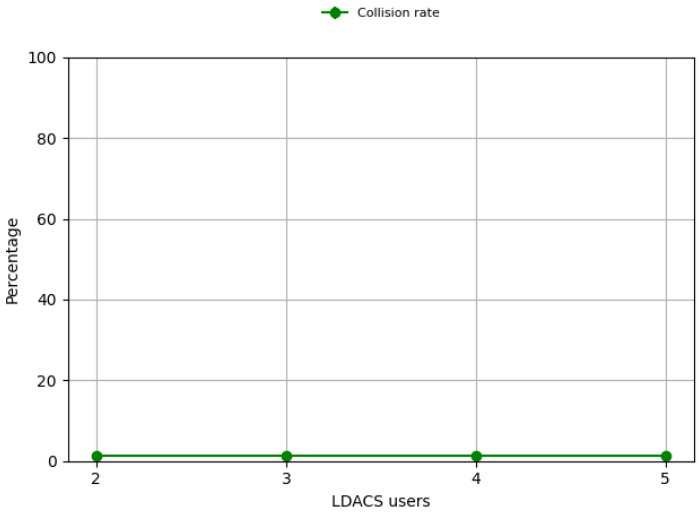



Figure 6.14: Collision rate on DME (untrained).

The second set of results can be seen in Figure 6.15. These results were achieved by training the model on the DME behavior first without LDACS communications and adding the LDACS communications once the accuracy had reached 1. This is why the collision rate drops to 0%: the LDACS users already know which channels to avoid and which ones are available.

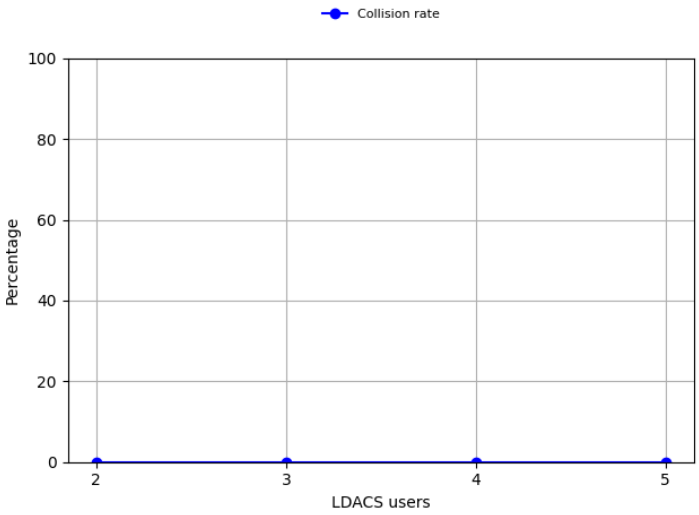



Figure 6.15: Collision rate on DME (trained).

These results can be seen in comparison with the previous collision rate in Figure 6.16. In blue is the data for the non-augmented MAC, in green the data for the untrained augmented LDACS MAC and in red the data for the trained augmented LDACS MAC. The biggest remark is that the collision rate for 2 LDACS users goes from 80% down to 1.25% for the untrained model, that is, a difference of 78.75%; and down to 0% for the trained model. For 3 LDACS users the difference is 48.75% and 50% respectively, for 4 LDACS users 31.75% and 33% respectively, and lastly for 5 LDACS users 23.75% and 25% respectively.

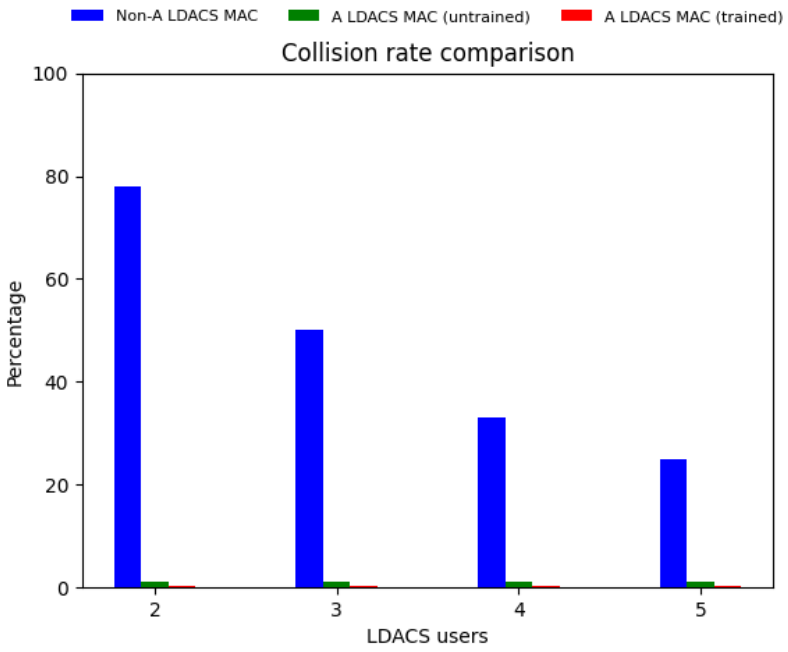



Figure 6.16: Collision rate comparison.

The second simulation to be evaluated is the critical scenario. Here, LDACS has only one available frequency channel and there are already DME users transmitting in said channel. The LDACS user were expected to learn how to go around the DME user without causing collisions by modifying their burst offset and start time slot. The results can be seen in Figure 6.17.

The collision rate is not 0%, but rather 8%. Moreover, there are certain simulations in which the accuracy of the learning algorithm decreases, as seen in Figure 6.18.

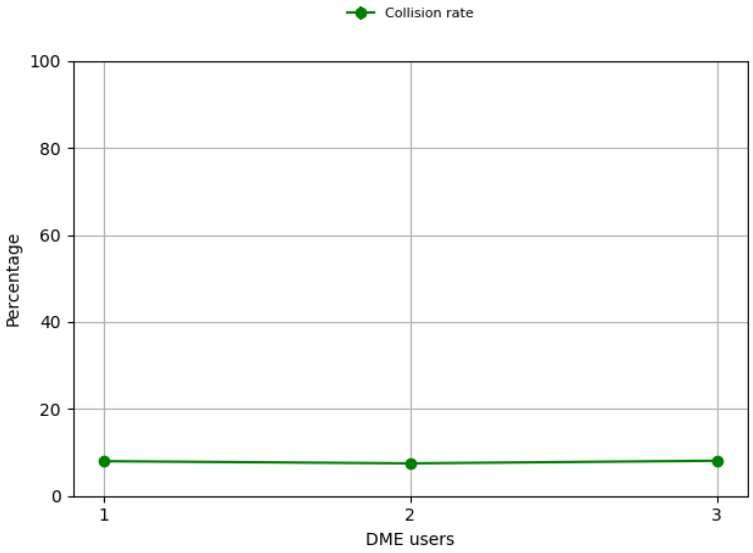



Figure 6.17: Collision rate.

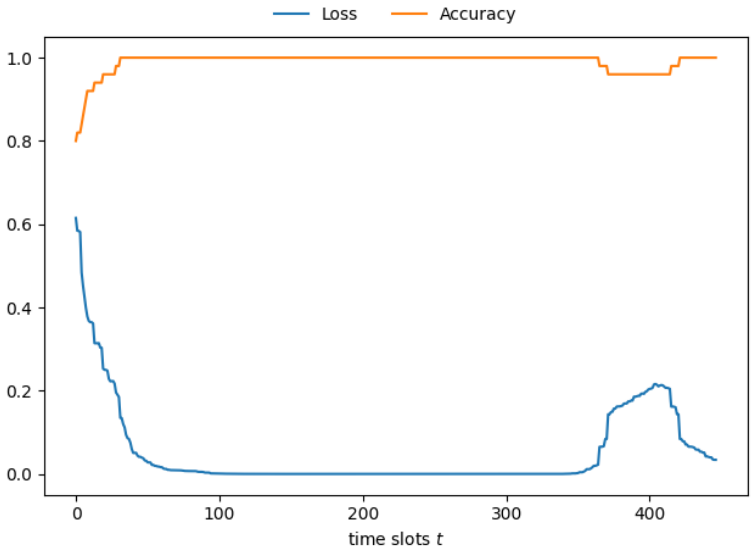
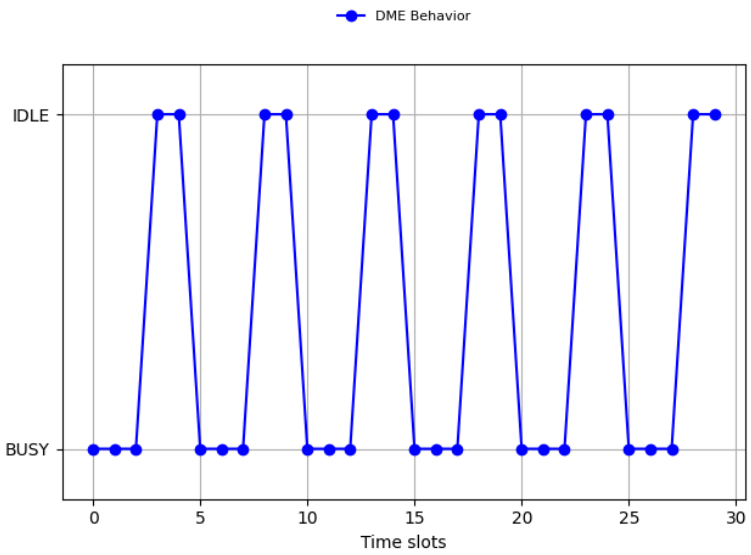



Figure 6.18: Accuracy decreasing.

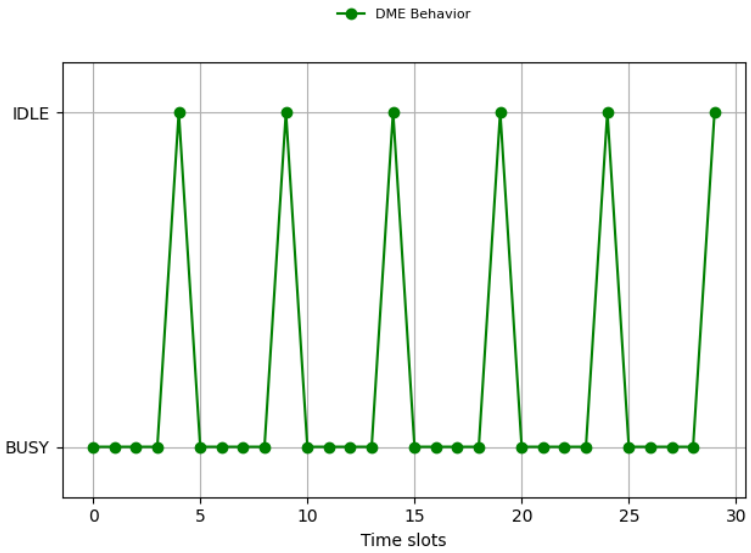
This drop in the accuracy is only present in some iterations of the simulations, not in all of them. The reason for this is not perfectly clear, but some theories showcase that it might be due to rounding errors that shift the pattern within the LSTM RNN, some internal offset happening, or perhaps some configuration of the learning algorithm is not

the ideal one, such as the learning rate or the optimizer used. All in all, the learning algorithm does not perform as flawlessly as expected.

Additionally, the more DME users that were added, the less resources LDACS could take advantage of. This can be seen in Figure 6.19.




(a) DME Behavior with 3 users (edge case).




(b) DME Behavior with 4 users (insufficient idle resources).

Figure 6.19: DME Behavior with 3 and 4 users.

Given that the burst length for LDACS is always 2 time slots, the case with 3 DME users (see Figure 6.19a) is an edge case since it has only the necessary resources for LDACS to be able to transmit. The case with 4 DME users (see Figure 6.19b) has only one available time slot out of each five, making it impossible for LDACS to establish a link. This is due to the fact that DME users transmit in consecutive LDACS slots.

This lack of idle resources can be seen in the number of established LDACS links. Figure 6.20 represents how many LDACS links are established in the span of 3 s of simulation time. It can be seen that it is possible to achieve 2 links in 3 s of simulation with 1, 2 and 3 DME users. From 4 DME users on, no links can be established due to the lack of resources.

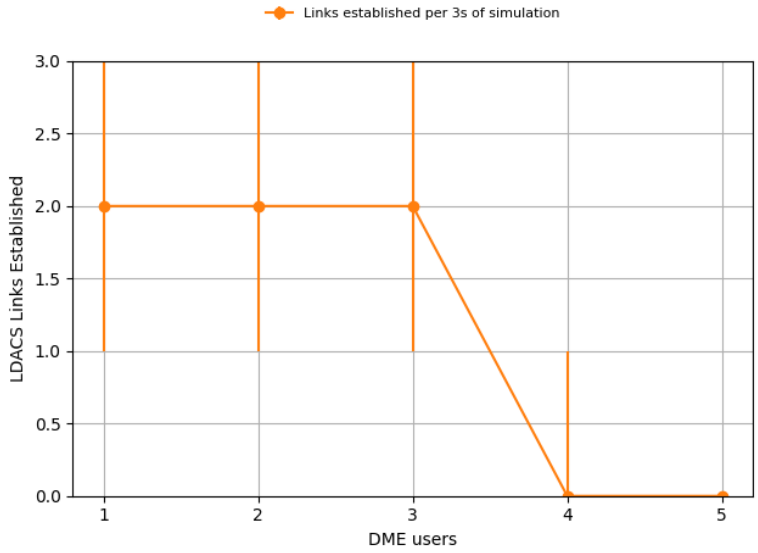



Figure 6.20: LDACS Links Established per 3 s of simulation.

Lastly, in previous simulations a growing number of DME users presented a constant collision rate of 80%. The comparison between these results and the results from the current LDACS MAC can be seen in Figure 6.21. The collision rate goes from 80% to 8%, resulting in a decrease of 72%.

These simulations are of course extreme cases in which the DME users were coded to occupy as many resources as possible in order to study the behavior of LDACS. They are therefore portrayed as the worst case scenario rather than as a realistic situation. As stated in Section 4.1, having $n$ DME users does not mean that $n$ time slots will be occupied, and there are in fact high chances that several DME users will be transmitting.

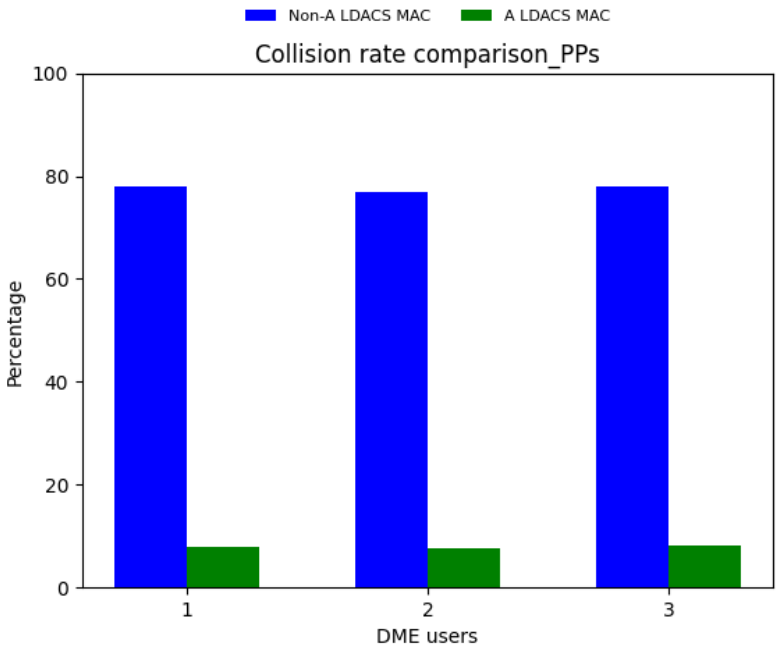



Figure 6.21: Collision rate comparison.

In conclusion, the performed simulations show that the new augmented LDACS MAC can:

- Learn the DME behavior.
- Identify idle frequency channels and avoid DME-occupied channels.
- When left no other choice, transmit in a DME-occupied channel making use of the idle time slots left by DME and avoiding most of the collisions that would take place otherwise.
- Keep establishing links even when several DME users are using the frequency channel (until the point where not enough idle slots are available).

# 7 Conclusion

Spectrum Scarcity is an obstacle in the way of newer and more advanced emerging communication systems. Thanks to CR and DSA, parts of the frequency spectrum that are licensed by legacy systems with low utilization rate can be used by unlicensed users while avoiding collisions and interferences with the main user. This thesis demonstrates that using ML for this purpose proves to be an efficient and successful method for finding available resources to transmit in.

This thesis was based on the project described in [25], which describes and simulates LDACS and its behavior. It was build using the OMNeT++ simulator. The DME system description was taken from the literature [20]–[22] and implemented in OMNeT++ as a new configurable module integrated in the LDACS modules from [25]. This implementation simulates how the aircraft send requests to the ground stations, how the ground stations answer in a different frequency channel after a fixed delay and how, upon detecting a collision with LDACS, the packet is lost.

This implementation served to simulate the interactions between LDACS and DME, which were measured in terms of the collision rate, that is, how many of the packets sent by DME end up colliding with an LDACS transmission. The first case to be analyzed was when LDACS "ignored" the existence of DME and just transmitted without taking it into account. The results from this approach were that the collision rate caused upon DME by 2 transmitting LDACS users was up to 80%, matching the analytical expectations. Due to how LDACS behaves when a larger number of LDACS users are in proximity, this 80% collision rate lowers down to around 30% with 5 transmitting LDACS users. Nevertheless, and taking into account that DME is a safety critical system, neither of these percentages are acceptable. The second case to be analyzed was the blacklisting approach, that is, LDACS does not transmit in any frequency channel that might be locally assigned to DME. This approach is very safe and yields no collisions, but it fails to take advantage to underused DME frequency channels where the utilization rate is as low as 3%, and even misses entire frequency channels where no DME users are transmitting at all, that is, where the DME utilization rate is of 0%.

The need for an augmented LDACS MAC is clear from the analysis of both approaches. Therefore, a detailed augmented LDACS MAC was designed and presented. It observes the DME medium access behavior through channel sensing and uses these observations to predict the future behavior of DME and avoid it. These predictions are done by a learning algorithm like the one described in [11], [12] and [34]. It was implemented including not only the regular RNN predictions but also inferring over a certain time horizon in order to constantly provide the LDACS MAC with as many future predictions

as necessary. This was done using Tensorflow, a ML software library that allows to build and train models. In order to take into account these predictions, the code from [25] was changed and adapted. Not only were the predictions taken into account to avoid collisions, but the MAC was made more flexible in order to be able to adapt to the DME predicted behavior and make opportunistic use of idle slots that would have never been used otherwise. Lastly, a TCP connection was established between OMNeT++ (containing the LDACS MAC and the DME modules) and Tensorflow (containing the learning algorithm).

The results form this implementation were quite favorable. It first showed that the new augmented LDACS MAC can identify which channels are being used and which not and switch in order to avoid DME altogether, resulting in a 0% collision rate. Most importantly, when encountered with a "critical scenario", that is, only one available channel in which DME is already transmitting, it proved to be able to adapt to the predicted DME behavior and transmit with a lower collision rate than the native approach, namely only 8% collision rate. This rate could be maintained until the DME pattern exceeded the proposed augmented LDACS MAC's communication pattern. For this reason, the next chapter will outline possible future extensions to improve upon the predictive MAC that has been investigated in this thesis.

## 7.1 Future Work

As successful as these results may be, there is plenty of work to be done in the future in order to make the LDACS MAC as efficient as possible and for it to be able to take advantage of all the idle available resources left by DME.

### 7.1.1 Learning Algorithm

As seen in Section 6.2, the collision rate for the worst case scenario remains at 8%, and the accuracy exhibits irregularities during certain simulations. Some reasons why this might happen are rounding errors within the learning algorithm, internal offsets or even a wrong LSTM parameter configuration. Future work should investigate different configurations for the LSTM RNN as well as more robust algorithms that might fit the objectives better.

### 7.1.2 Handshake Implementation

The augmented MAC model presented in Chapter 4 included both communicating aircraft sharing their local predictions and the probabilities of the candidate time slots of being idle. Due to the time constraints of this project, this part was not implemented in the current Augmented LDACS MAC.

Moreover, the inability to find an appropriate fixed probability acceptance threshold and the unrealistic technical requirements of a dynamic adaptive threshold were also presented. Therefore, there is the need to set a limit or threshold (whether fixed or dynamic) to discern which probabilities of being idle can be accepted and which cannot due to them being too low and not offering enough security for the transmission.

Future work should then include the implementation of the comparison in predictions between the communicating aircraft, as well as finding an appropriate and technically possible probability acceptance threshold.

### 7.1.3 Game Theory

Leaving behind the augmented LDACS MAC proposed in Chapter 4, the design itself could be improved. As seen in Figure 6.19b, there are patterns that can be simply not used opportunistically. This is due to two reasons:

- The link is established in one frequency channel and cannot hop to another one.
- The burst length is always two consecutive time slots.

These characteristics (as seen in Figure 7.1) limit the range of possible resources that can be taken advantage of, especially in highly congested environments with many DME users transmitting.

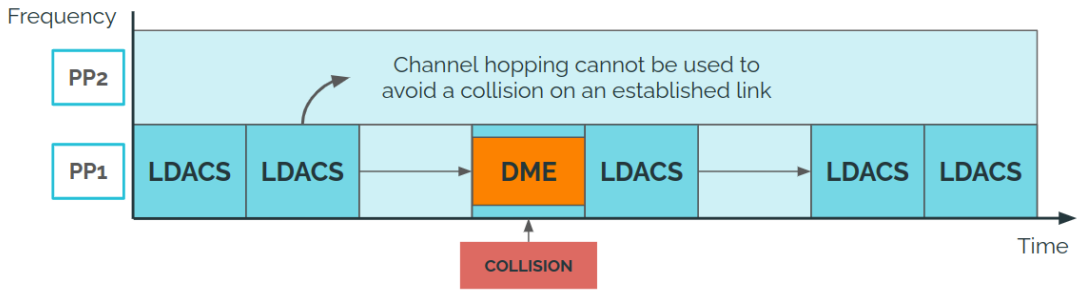
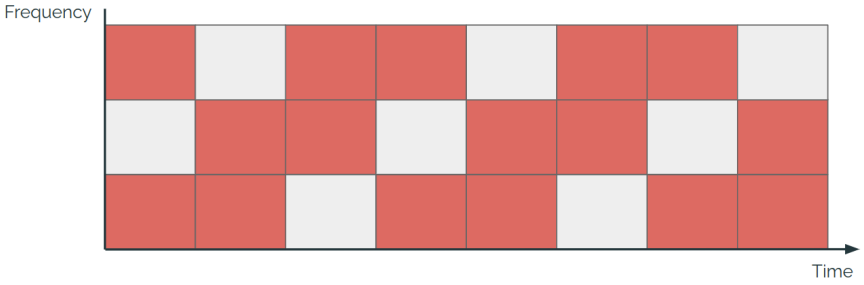
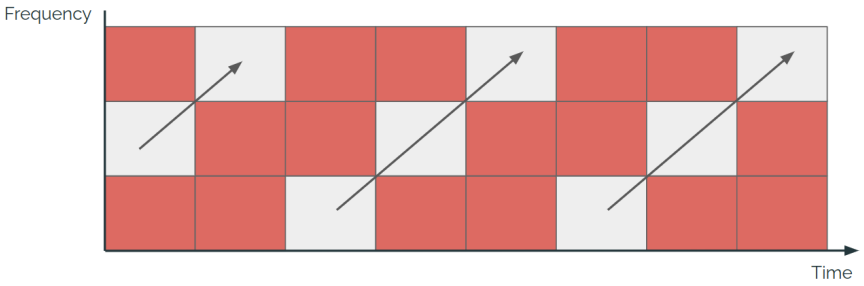



Figure 7.1: LDACS MAC Limitations.

Techniques like GT could be used to alleviate this problem [15]. GT formulates the problem of fitting the LDACS transmission to the DME behavior as a kind of optimization problem called a Spectrum Mobility Game (SMG) [16]. As its name indicates, the LDACS MAC now has to move through the available spectrum in order to find the best possible combination of frequency channels and time slots in which to transmit. An example of this can be seen in Figure 7.2. Figure 7.2a shows a frequency spectrum sensing in which the gray slots are idle, and the red ones are busy. Using GT this game can be solved as seen in Figure 7.2b, where the unlicensed user has found a way of transmitting while avoiding collisions.




(a) Pattern.

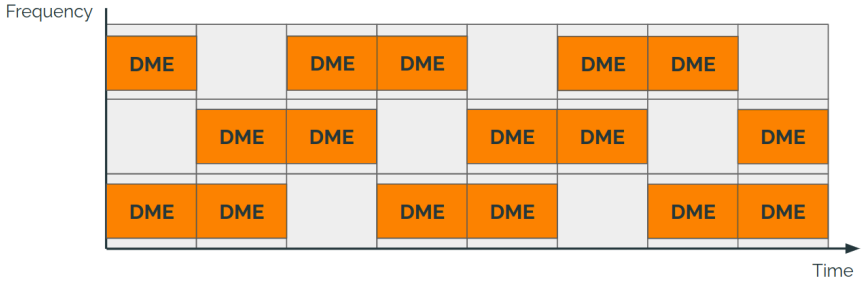



(b) Opportunistic Access.

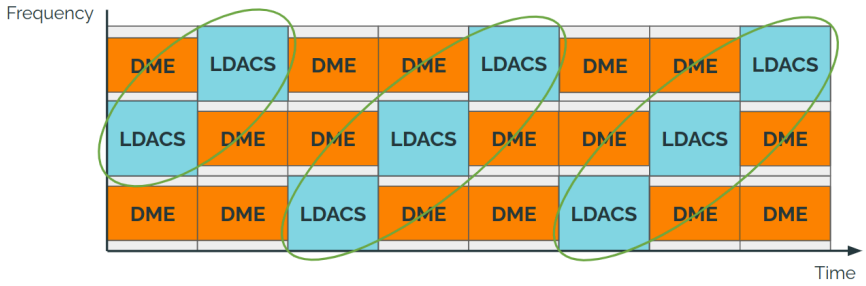
Figure 7.2: SMG example.

Similarly, this can be applied to LDACS MAC. Figure 7.3a shows the same pattern as 7.2a, one that could never be opportunistically accessed by the current LDACS MAC due to the required hops between frequency channels and the inability to have two consecutive LDACS time slots assigned to a single burst. However, Figure 7.3b shows how this SMG would have been solved using GT, the same way as Figure 7.2b was solved. LDACS would be able to transmit in all the idle slots without colliding with DME.

Moreover, SMGs are proven to have at least one Nash equilibrium point. A Nash equilibrium point means that a strategy is reached for all players of the game in which none can improve their individual payoff by changing strategies unilaterally. In other words, since the DME behavior is fixed (only depending on the frequency with which pulse pairs are sent), reaching a Nash equilibrium would mean that LDACS finds an optimal way of transmitting around DME without causing collisions. This could drastically improve LDACS's chances of being able to opportunistically transmit around congested areas or in locations with high air traffic volume. Therefore, future work should include implementing GT and SMG into the LDACS MAC.




(a) Pattern.




(b) Opportunistic Access.

Figure 7.3: DME as a SMG.

# List of Figures

# Acronyms

**A2A** Air-to-Air. 1, 2, 6, 7, 13

**A2G** Air-to-Ground. 1, 6

**ANN** Artificial Neural Network. 3, 12, 30

**CR** Cognitive Radio. 2, 11, 51

**CSMA/CA** Carrier Sense Multiple Access/Collision Avoidance. 2

**DME** Distance Measuring Equipment. 1, 3–6, 10–12, 14, 16–19, 21–46, 48–57

**DSA** Dynamic Spectrum Access. 2, 11, 51

**EUROCONTROL** European Organisation for the Safety of Air Navigation. 7

**FCC** Federal Communications Commission. 2

**GT** Game Theory. 3, 53, 54

**IoT** Internet of Things. 2

**LDACS** *L*-band Digital Aeronautical Communications System. 1, 4, 6–8, 10, 11, 13–15, 17–46, 48–54, 56, 57

**LSTM** Long Short Term Memory. 2, 3, 12, 13, 17, 30, 47, 52, 56

**LTE-U** Long Term Evolution Unlicensed. 2

**MAC** Medium Access Control. 1, 7, 13, 17, 18, 22–26, 28–31, 34, 35, 39, 43, 44, 46, 49–54, 56, 57

**MC** Markov Chain. 3

**ML** Machine Learning. 1, 2, 11, 17, 29, 51, 52

**OFDM** Orthogonal Frequency Division Multiplexing. 7, 56

**PP** Point-to-Point Channel. 7–10, 56

**RNN** Recurrent Neural Network. 2, 3, 12, 13, 17, 30, 32, 43, 47, 51, 52, 56

**SESAR** Single European Sky ATM Research. 1

**SH** Shared Channel. 7–10

**SMG** Spectrum Mobility Game. 53–55, 57

**TVWS** Television White Space. 2

# Bibliography


[1] William H Melody. 'Radio spectrum allocation: role of the market'. In: vol. 70. 2. JSTOR, 1980, pp. 393–397.

[2] Hamed Mosavat-Jahromi, Yue Li, Lin Cai et al. 'Prediction and Modeling of Spectrum Occupancy for Dynamic Spectrum Access Systems'. In: vol. 7. 3. 2021, pp. 715–728. DOI: 10.1109/TCCN.2020.3048105.

[3] *SESAR*. https://www.sesarju.eu/. [Online; accessed 30-May-2022].

[4] Santosh Nagaraj and Faris Rassam. 'Cognitive Radio in TV White Space'. In: *2014 National Wireless Research Collaboration Symposium*. 2014, pp. 84–89. DOI: 10.1109/NWRCS.2014.20.

[5] Junhong Chen, Zhan Gao and Yuhua Xu. 'Opportunistic spectrum access with limited feedback in unknown dynamic environment: a multiagent learning approach'. In: *The 2014 5th International Conference on Game Theory for Networks*. November 2014, pp. 1–6. DOI: 10.1109/GAMENETS.2014.7043715.

[6] Hamed Mosavat-Jahromi, Yue Li, Lin Cai et al. 'Prediction and modeling of spectrum occupancy for dynamic spectrum access systems'. In: *IEEE Transactions on Cognitive Communications and Networking*. Vol. 7. Issue: 3. IEEE, 2021, pp. 715–728.

[7] Shangxing Wang, Hanpeng Liu, Pedro Henrique Gomes et al. 'Deep reinforcement learning for dynamic multichannel access in wireless networks'. In: vol. 4. 2. IEEE, 2018, pp. 257–265.

[8] Adam Dziedzic, Vanlin Sathya, Muhammad Iqbal Rochman et al. 'Machine Learning Enabled Spectrum Sharing in Dense LTE-U/Wi-Fi Coexistence Scenarios'. In: vol. 1. 2020, pp. 173–189. DOI: 10.1109/OJVT.2020.2981519.

[9] Jonas Beuchert, Friedrich Solowjow, Sebastian Trimpe et al. 'Overcoming Bandwidth Limitations in Wireless Sensor Networks by Exploitation of Cyclic Signal Patterns: An Event-triggered Learning Approach'. In: 2020, p. 260. DOI: 10.3390/s20010260.

[10] Husheng 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)*. 2010. DOI: 10.1109/SDR.2010.5507919.

[11] Sebastian Lindner, Leonard Fisser and Andreas Timm-Giel. 'Coexistence of Shared-Spectrum Radio Systems through Medium Access Pattern Learning using Artificial Neural Networks'. In: 2020.

[12] Sebastian Lindner, Daniel Stolpmann and Andreas Timm-Giel. 'Time-and Frequency-Domain Dynamic Spectrum Access: Learning Cyclic Medium Access Patterns in Partially Observable Environments'. In: vol. 80. 2021.

[13] Lixing Yu, Qianlong Wang, Yifan Guo et al. 'Spectrum availability prediction in cognitive aerospace communications: A deep learning perspective'. In: *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*. IEEE. 2017, pp. 1–4.

[14] Saurabh Kumar and Hyungwon Kim. 'Energy Efficient Scheduling in Wireless Sensor Networks for Periodic Data Gathering'. In: 2019, pp. 11410–11426. DOI: 10.1109/ACCESS.2019.2891944.

[15] Teresa Algarra Ulierte. 'Predictive medium access for shared-spectrum radio systems using Game Theory'. In: 2021.

[16] R. Southwell, J. Huang and X. Liu. 'Spectrum mobility games'. In: *2012 Proceedings IEEE INFOCOM*. 2012, pp. 37–45. DOI: 10.1109/INFCOM.2012.6195776.

[17] Sunil Jacob, Varun G. Menon, Saira Joseph et al. 'A Novel Spectrum Sharing Scheme Using Dynamic Long Short-Term Memory With CP-OFDMA in 5G Networks'. In: vol. 6. 3. 2020, pp. 926–934. DOI: 10.1109/TCCN.2020.2970697.

[18] Xiaoge Wu, Lin Zhang and Zhiqiang Wu. 'Quantized Soft-Decision-Based Compressive Reporting Design for Underlay/Overlay Cooperative Cognitive Radio Networks'. In: vol. 6. 3. 2020, pp. 1044–1055. DOI: 10.1109/TCCN.2020.2988479.

[19] Alexander Steingass, Thanawat Thiasiriphet and Jaron Samson. 'Modeling distance measurement equipment (DME) signals interfering an airborne GNSS receiver'. In: vol. 65. 2. Wiley Online Library, 2018, pp. 221–230.

[20] Ulrich Epple, Felix Hoffmann and Michael Schnell. 'Modeling DME interference impact on LDACS1'. In: *2012 Integrated Communications, Navigation and Surveillance Conference*. IEEE. 2012, G7–1.

[21] Sherman C Lo and Per Enge. 'Assessing the capability of distance measuring equipment (DME) to support future air traffic capacity'. In: vol. 59. 4. 2012, pp. 249–261.

[22] Taher Jalloul, Wessam Ajib, Omar A. Yeste-Ojeda et al. 'DME/DME navigation using a single low-cost SDR and sequential operation'. In: *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*. 2014, pp. 3C2-1-3C2–9. DOI: 10.1109/DASC.2014.6979451.

[23] *EUROCONTROL*. https://www.eurocontrol.int/system/l-band-digital-aeronautical-communication-system/. [Online; accessed 14-May-2022].

[24] *LDACS*. https://www.ldacs.com//. [Online; accessed 14-May-2022].

[25] Andreas Timm-Giel, Sebastian Lindner, Musab Ahmed et al. 'MCSOTDMA Specification Appendix Deliverable AP 3.1'. In: *Inter Aircraft Network*. 2021.

[26] Thomas Ewert, Nils Mäurer and Thomas Gräupl. 'Group Key Distribution Procedures For The L-Band Digital Aeronautical Communications System (LDACS)'. In: *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE. 2021, pp. 1–10.

[27] Michael Schnell, Ulrich Epple, Dmitriy Shutin et al. 'LDACS: future aeronautical communications for air-traffic management'. In: vol. 52. 5. 2014, pp. 104–110. DOI: 10.1109/MCOM.2014.6815900.

[28] Miguel A. Bellido-Manganell and Michael Schnell. 'Towards Modern Air-to-Air Communications: the LDACS A2A Mode'. In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. 2019, pp. 1–10. DOI: 10.1109/DASC43569.2019.9081678.

[29] Michael Zaisberger and Holger Arthaber. 'Random Access Performance Evaluation and Improvements of the LDACS'. In: *Proceedings of the 4th International Conference on Telecommunications and Communication Engineering*. Ed. by Maode Ma. Singapore: Springer Singapore, 2022, pp. 26–34.

[30] Miguel A. Bellido-Manganell. 'Design Approach of a Future Air-to-Air Data Link'. In: *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. 2018, pp. 1–9. DOI: 10.1109/DASC.2018.8569594.

[31] Navin Kumar Manaswi, Navin Kumar Manaswi and Suresh John. 'Deep Learning with Applications Using Python'. In: Springer, 2018, pp. 115–126.

[32] Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep learning*. MIT press, 2016.

[33] Yiyang Pei, Ying-Chang Liang, Kah Chan Teh et al. 'Sensing-throughput trade off for cognitive radio networks: A multiple-channel scenario'. In: *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*. 2009, pp. 1257–1261. DOI: 10.1109/PIMRC.2009.5450216.

[34] Leonard Fisser, Sebastian Lindner and Andreas Timm-Giel. 'Predictive scheduling and opportunistic medium access for shared-spectrum radio systems in aeronautical communication'. In: *Deutscher Luft-und Raumfahrtkongress 2020*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2020.

[35] *OMNeT++*. https://doc.omnetpp.org/omnetpp/manual/. [Online; accessed 12-June-2022].

[36] *Tensorflow*. https://www.tensorflow.org/about. [Online; accessed 12-June-2022].

[37] Peter Goldsborough. 'A tour of Tensorflow'. In: 2016.

[38] *Keras*. https://keras.io/about/. [Online; accessed 12-June-2022].

[39] Nikhil Ketkar. *Introduction to Keras*. Springer, 2017, pp. 97–111.

[40] *Unittest*. https://docs.python.org/3/library/unittest.html. [Online; accessed 12-June-2022].

[41] Zijun Zhang. 'Improved Adam optimizer for Deep Neural Networks'. In: *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE. 2018, pp. 1–2.

[42] Max Schettler, Dominik S. Buse, Anatolij Zubow et al. 'How to Train your ITS? Integrating Machine Learning with Vehicular Network Simulation'. In: *12th IEEE Vehicular Networking Conference (VNC 2020)*. Virtual Conference: IEEE, December 2020. DOI: 10.1109/VNC51378.2020.9318324.