PROTOCOL SOLUTIONS FOR ROUTING COMMUNICATIONS IN NETWORKS OF COOPERATIVE AERIAL DRONES

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Alicia Martin

In partial fulfilment

of the requirements for the degree in

TELECOMMUNICATIONS TECHNOLOGIES AND SERVICES ENGINEERING

Advisor: Novella Bartolini,

Xavier Hesselbach

Barcelona, June 2022

Abstract

Nowadays, Unmanned Aerial Vehicles (UAVs) have a wide range of commercial and civilian applications due to their ease of deployment, mobility, and reduced cost. The dynamic behaviour, high mobility and topology changes causes the UAVs links to be frequently disconnected and so the design of routing protocols is quite challenging.

The objective of this thesis is to study through simulations the routing protocol solution contributed by my advisor and the research group of the department, the Movement Assisted Delivery (MAD) protocol. First, we analyse a UAV network with homogenous packet deadline to study how MAD behaves for delay tolerant and non-delay tolerant scenarios separately. Then we focused on an urgent scenario, with heterogeneous packet deadlines, where based on the local findings, there is an area more tolerant to the delay than the other.

Revision history and approval record

Revision	**Date**	**Purpose**
0	03/06/2022	Document creation
1	06/06/2022	Document revision
2	10/06/2022	Document revision
3	13/06/2022	Document revision
4	17/06/2022	Document revision
5	20/06/2022	Document revision

DOCUMENT DISTRIBUTION LIST

Name	**e-mail**
Alicia Martin Faundez	
Novella Bartolini	
Xavier Hesselbach Serra	

Written by:		Reviewed and approved by:	
Date	20/06/2022	Date	
Name	Alicia Martin	Name	Xavier Hesselbach
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables:

1. Introduction

Unmanned Aerial Vehicles (UAVs) were initially designed for military applications but with the technological and research advances, it is now possible to employ small UAV networks at a low cost. Nowadays, UAVs have found a wide range of commercial and civilian applications due to their ease of deployment, mobility, and reduced cost. The dynamic behaviour, high mobility and topology changes causes the UAVs links to be frequently disconnected and so the design of routing protocols is quite challenging.

1.1. Statement of purpose

This project is carried out at the University of La Sapienza in Rome in the framework of the computer science department. It is based on previous contributions of the research group, concretely in the Movement Assisted Delivery (MAD) routing protocol.

The purpose of this project is to study protocols for routing communications in networks of cooperative aerial drones, focusing deeply on the MAD routing solution. The work analyses how MAD behaves in different scenarios and the advantages and disadvantages that has for different applications.

The main goals are:

1- To identify the design requirements of routing protocols for UAV networks.

2- To classify and study the already existing routing protocols for UAV networks.

3- To study the MAD routing protocol and analyse its performance in different scenarios.

1.2. Project requirements and specifications

Project requirements:

- Knowledge of Flying Ad-hoc Networks (FANETs), the requirements for the design of its routing protocols and the classification of the already existing routing protocols.
- Knowledge of MAD approach: its properties, the use of Reinforcement Learning (RL), the selection of the relay node….
- A computer to perform simulations of MAD for different scenarios.

Project specifications:

- Simulations of two different scenarios. First, we have simulated a UAV network with homogenous packet deadline to study delay tolerant applications and then, non-delay tolerant applications. The second scenario is focused as an emergency scenario where one part of the area is more tolerant to the delay than the other area. Therefore, the UAV network has heterogeneous packet deadlines, one for the delay tolerant area and another for the non-delay tolerant.
- Study of the following parameters: packet delivery ratio, average packet delay and average packet delay divided by packet deadline.

1.3. Methods and procedures

The project starts searching information and learning about ad-hoc wireless networks, specifically about FANETs and its properties and requirements. Then we focused on studying and classifying the already existing routing protocols for UAV networks.

To learn about MAD, we start from the following paper [1] published by my supervisor Novella Bartolini and the research group of the computer science department. In the paper it is exposed the MAD routing protocol and its properties. It explains how MAD uses a RL approach to make adaptive decisions, and it defines an adaptive relay selection scheme. Finally, through simulations, there is a comparison of MAD with benchmark solutions with no mobility control.

In this project, we want to analyse how the MAD routing protocol behaves for different scenarios to see how it performs in delay tolerant and non-delay tolerant scenarios. To do the simulations, we have used Python code previously developed by the research group of the department. I have worked based on this code and have added the modifications and code needed to simulate the desire scenarios, and obtain the wanted figures to be able to analyse some parameters.

1.4. Work plan

1.4.1. Work packages, Tasks and Milestones

The tasks to be carried out in the project are divided in three work packages: WP1 preparation phase, WP2 contribution phase and WP3 closing tasks.

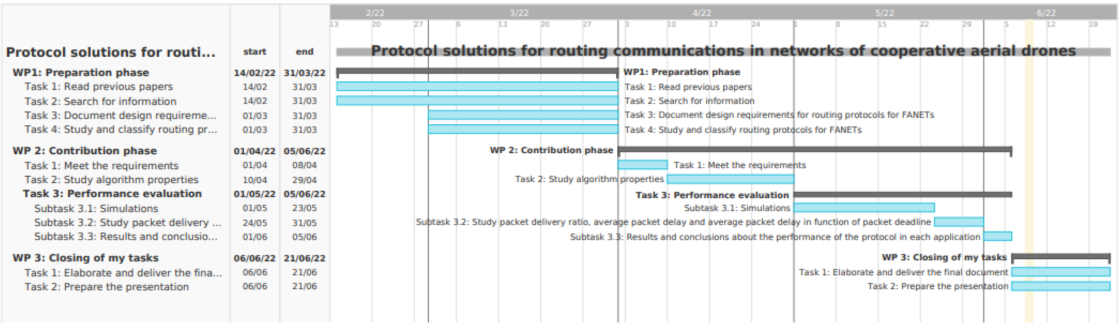
Project: Protocol solutions for routing communications in networks of cooperative aerial drones	WP ref: (**WP1**)	
Major constituent: **Preparation phase**	Sheet 1 of 1	
Short description: The aim is to learn about ad-hoc wireless networks, specifically Mobile Ad-Hoc Networks (MANETs), Vehicle Ad-Hoc Networks (VANETs) and FANETs. Also, document and analyse deeply the routing protocols for FANETs.	Planned start date: 14/02/22 Planned end date: 31/03/22 Start event: 14/02/22 End event: 31/03/22	
Internal task T1: Read previous papers, including those of the research group of the department Internal task T2: Search for information and learn about the topics Internal task T3: Document the design requirements for routing protocols for FANETs	Deliverables:	Dates:

Internal task T4: Study and classify the already existing routing protocols for FANETs		

Project: Protocol solutions for routing communications in networks of cooperative aerial drones	WP ref: (**WP2**)	
Major constituent: **Contribution phase**	Sheet 1 of 1	
Short description: Focus on studying the MAD routing protocol. Analyse its performance in different scenarios, specifically for delay critical and non-delay critical scenarios.	Planned start date: 01/04/22 Planned end date: 31/05/22 Start event: 01/04/22 End event: 05/06/22	
Internal task T1: Meet the requirements Internal task T2: Study algorithm properties Internal task T3: Performance evaluation Internal subtask T3.1: Simulations Internal subtask T3.2: Study packet delivery ratio, average packet delay and average packet delay divided by packet deadline for each scenario Internal subtask T3.3: Results and conclusions about the performance of the protocol in each scenario	Deliverables:	Dates:

Project: Protocol solutions for routing communications in networks of cooperative aerial drones	WP ref: (**WP3**)	
Major constituent: **Closing of my tasks**	Sheet 1 of 1	
Short description: Prepare the final document of the project based on all the documents done until then and elaborate the presentation for the defence of the thesis.	Planned start date: 31/05/22 Planned end date: 21/06/22 Start event: 06/06/2022 End event: 21/06/2022	
Internal task T1: Elaborate and deliver the final document Internal task T2: Prepare the presentation	Deliverables:	Dates:

1.4.2. Gantt diagram



Protocol solutions for routi...	start	end
WP1: Preparation phase	**14/02/22**	**31/03/22**
Task 1: Read previous papers	14/02	31/03
Task 2: Search for information	14/02	31/03
Task 3: Document design requireme...	01/03	31/03
Task 4: Study and classify routing pr...	01/03	31/03
WP 2: Contribution phase	**01/04/22**	**05/06/22**
Task 1: Meet the requirements	01/04	08/04
Task 2: Study algorithm properties	10/04	29/04
Task 3: Performance evaluation	**01/05/22**	**05/06/22**
Subtask 3.1: Simulations	01/05	23/05
Subtask 3.2: Study packet delivery ...	24/05	31/05
Subtask 3.3: Results and conclusio...	01/06	05/06
WP 3: Closing of my tasks	**06/06/22**	**21/06/22**
Task 1: Elaborate and deliver the fina...	06/06	21/06
Task 2: Prepare the presentation	06/06	21/06

1.5. Deviations from the initial plan

Initially, we were going to test the performance of MAD for a scenario with homogenous packet deadlines, to study delay tolerant and non-delay tolerant applications separately. Finally, we have also simulated an emergency scenario with heterogeneous packet deadlines. Based on the urgency of the local findings one part of the area is less tolerant to the delay than the other, and therefore, the requirements, like the packet deadlines, are different for each area.

We also decided to do the simulations using a Brownian mobility model for the representation of UAVs movement, in addition to the random waypoint mobility model.

2. State of the art of the technology used or applied in this thesis:

2.1. Introduction to FANETs

UAVs, more known as drones, were initially designed for military missions. Among other features, they took advantage of the fact that no pilot is required on board, so it was useful on situations that required accessing harsh and inhabitable environments. Nowadays, thanks to the technological advances in embedded systems and wireless communications, and the trend for integration and miniaturization, it is possible to employ multiple small UAV systems at a low cost. Therefore, apart from the military domain, these vehicles are increasingly being used in civil and commercial applications due to their low maintenance cost, ease of deployment and high mobility**.**

UAVs can connect with each other forming a FANET that as an ad-hoc wireless network does not require a set infrastructure to work. This means that the communication between the UAV nodes is done with no need to access point (AP) or base station (BS) that provides access control.

FANETs enables multi-hop transmissions using the UAV as a router and using also ground control systems like a Ground Control Station (GCS) to provide the facilities for monitoring, control and operation. Only a subset of UAVs are required to communicate directly with the ground control systems, the others act as relays to accomplish data transmissions through multi-hop communications.

The nature of ad-hoc networks like also MANETs and VANETs is dynamic. In FANETs, the mobility features of UAVs are different from those in MANETs and VANETs because nodes have a higher speed variation and mobility degree. Other unique characteristics of FANETs are that nodes move in a 3D air space whereas in MANETs and VANETs, they move in a 2D space. In addition, in FANETs the number of nodes is relatively smaller due to the long distance separating the UAVs so we have sparse networks that require longer transmissions range. These unique features involve many challenges in the design of routing protocols for UAV networks. The majority of the approaches designed for FANETs, are based on already existing routing protocols designed for MANETs and VANETs, but adapted to fulfil the requirements of UAV networks.

2.2. Routing requirements for FANETs

Let see some design considerations to take into account to develop a routing protocol able to achieve low end-to-end delay, robust connectivity and high delivery ratio for FANETs.

1. Mobility. The rapid mobility of UAV nodes and variation in the distance between them, cause fluctuations in the wireless links between UAVs, which are even frequently disconnected. This force the network to re-organize implying in frequent topology changes. Therefore, the efficiency of routing techniques varies on the speed of the UAVs. The faster the UAVs move, the more likely the links between the nodes may be disconnected.

As the mobility of FANETs nodes is higher than in MANETs and VANETs, this feature becomes an important consideration to adapt, already existing protocols for these last two ad-hoc networks, for FANETs.

2. Latency. An important requirement of routing protocols for FANETs is to achieve a low end-to-end delay. Depending on the application of the UAV network, this feature takes more or less significance. Delay critical scenarios, like emergency monitoring, for example after an earthquake, require minimal latency as the information needs to be transmitted at very high rates to act as fast as possible. On the other hand, in non-delay critical scenarios, like griculture, for monitoring crops, there is a higher tolerance of this value as it is not essential to receive the data at high rates.

3. Frequent link disconnection: As previously said due to the dynamic rapid mobility of UAVs, the network density varies and that can lead to frequent disconnection of the network. Therefore, we have that in spare UAV networks the link disconnection is higher than in high-density networks where UAVs are easily connected and have less rates of broken communications and consequently less delay.

4. Scalability: The cooperation between UAVs increases the performance of the system, helping to execute missions that would otherwise be difficult or impossible to be executed by a single node. Cooperation addresses the collaborative behaviour of multiple nodes toward achieving a common objective. For the design of a routing protocol for FANETs, it is important to consider a good number of UAVs to perform together at a time without causing degradation to the system performance.

2.3. Routing protocols for FANETs

Let's classify some of the existing routing protocols for FANETs based on the routing strategies used, in four subsections: topology based routing, geographic routing, hybrid routing and bio-inspired routing.

1. **Topology based routing**: The routing information from the sender to the destination must be obtained according to the topological information of the nodes before data transmission begins. To define the node they exploit the IP addresses in the network. It does not support GPS.

    A. **Flat**: Uses a flat addressing strategy in which each UAV node taking part in the routing has an equal role.

        a) **Proactive Routing Protocols (PRP)**: Use routing tables to store all the routing in the communication network. These tables are updated and shared periodically among the nodes. When the topology changes, tables need to update what causes slow response and results in delay in the network. These protocols have high memory requirement, for the routing table, and very high bandwidth and energy consumption due to the high number of signalling for routing table maintenance and update. They have poor response to the fast topology changes characterize of FANETs.

Therefore, proactive schemes are suitable for applications using small number of UAVs with low mobility degree and that require quicker real-time communications. They have medium complexity.

b) **Reactive Routing Protocols (RRP):** Explore and maintain the routing path on demand, only when needed. When a pair of nodes are communicating the route is stored. These protocols adapt to topology changes and have less bandwidth and energy consumption due to the less number of signalling overhead for routing table creation and maintenance compared to proactive schemes. However, it may take a long time to find the route, resulting in high latency in the network.
Therefore, reactive schemes are suitable for applications using small to medium number of UAVs with high mobility degree and delay tolerant, to the latency introduced for new route discovery. They have average complexity

c) **Hybrid Routing Protocols:** Combine reactive and proactive routing mechanisms to try to minimize the overhead problem, since RRP needs more time to discover routes and PRP has a large overhead of control messages. These protocols have high memory requirement, medium bandwidth and energy consumption for control packets and a high transmission delay.
Therefore, hybrid schemes are suitable for applications using small to large number of UAVs that may have several sub network areas, where intra zone routing uses PRP and inter zone uses RRP. Also, applications with average mobility degree and delay tolerant. They have average complexity.

When the size of the networks increases, flat routing strategies produce an enormous volume of routing messages. In addition, the network can be made up of heterogeneous UAV nodes that have different sizes, power, computational and memory capabilities…which commonly produce a hierarchy in the network.

B. **Hierarchical (cluster-based) routing**: Reduce the volume of routing messages that spreads in the network, resolve node heterogeneity and improve the network scalability. These protocols reduce the memory use in large scale networks by using different routing strategies within a cluster and outside a cluster and reduce the bandwidth and energy consumption.
Therefore, hierarchical schemes are suitable for applications using large number of heterogeneous UAVs with average mobility degree and delay tolerant. They have high complexity.

2. **Geographic routing (Position based routing)**: Uses geographic position of the UAVs to discover the most efficient routing path to forward data packets. Now, there is no need for a routing table, it is enough with knowing the location of the neighbour nodes. Compared to topology based routing this scheme has less bandwidth consumption for control packet exchange, moderate network overhead and has low transmission delay. However, it uses GPS to find the geographical location of the nodes, which leads to hardware complexity.

A. **Non-delay tolerant networks (Non-DTN) geographic routing protocols**: Use greedy forwarding techniques where each node selects the next forwarder node closest to the destination among its neighbour nodes. This mechanism fails when a source node has no neighbour node close. Therefore, these protocols are suitable for large scale and dense networks.

B. **Delay tolerant networks (DTN) geographic routing protocols**: Use the store-carry-forward technique so when a node is trying to forward a data packet is unable to reach the other nodes it holds the message for a predefined time until it finds a potential forwarding node. However, this leads to high latencies and high bandwidth and energy consumption.
Therefore, these protocols are suitable for sparse networks with high mobility of the nodes and applications with packet delivery requirement and delay tolerant.

3. **Hybrid routing**: Combine the topology based and geographic routing mechanisms. During reactive link failure, it uses the geographic forwarding scheme to increase the packet delivery ratio. Is suitable for delay tolerant applications.

4. **Bio-inspired routing**: Is inspired from the natural phenomena of ant colonies, honey bees, and bird flocks. Is suitable for applications that use UAVs in a coordinated formation and swarm cooperation application architectures.

CRITERIA	PROACTIVE	REACTIVE	HYBRID	HIERARCHICAL (CLUSTER-BASED)
Memory requirement	Very high	High	High	Medium
Signalling overhead	Very high	High	Medium	High
Bandwidth and energy consumption	Very high	High	Medium	Medium
Latency	Very low	High	Medium to high	Medium
Network size	Small scale	Small to medium scale	Small to large scale	Small to large scale
Application scenarios	Real time applications	Post disaster operation, data collection in agriculture	Search and rescue operations	Battlefield applications, cooperative surveillance

Table 1. Comparison of topology-based routing protocols

CRITERIA	GEOGRAPHIC (Non-DTN)	GEOGRAPHIC (DTN)	HYBRID (topology and geographic)	BIO-INSPIRED
Memory requirement	Low	Low	High	Medium
Signalling overhead	Low	High	Medium	High
Bandwidth and energy consumption	Low	High	Medium	High
Latency	Low	High	Medium	Medium
Network size	Small to large scale	Small to large scale	Small to large scale	Small to large scale
Application scenarios	Monitoring, surveillance	Delay tolerant applications, sensor data collection	Network coverage	Battlefield applications

Table 2. Comparison of routing protocols approaches

2.4. MAD routing protocol

This routing protocol solution follows the basis of the geographic schemes supported by a reinforcement learning approach to guide device movements during their mission, based on delivery needs. Let's study how MAD works.

UAVs are deployed in the monitored area equipped with a localization module, like GPS, and separate transceivers for simultaneous transmission and reception activities. We assume they are aware of their current geographical destination and move towards it at a constant speed. They are also aware of the position of the sink (σ).
UAVs need to know their current neighbours, so each node has to exchange every δ_{hello} hello packets to communicate information necessary for routing, like node direction and speed. When a node *i* receives a hello$_{(u,*)}$ packet from node *u*, it stores the information in its neighbourhood map *Hu*.

When a UAV detects an ongoing event, it has to send the data to the sink. For this purpose, the source node searches for a suitable relay looking among its neighbours, which are approaching the sink. Then, the source selects the node with whom it can perform a two-way communication to send the data packet and received the acknowledge message (ACK), before the distance between them exceeds their transmission range and therefore they cannot communicate.
If this node exists, the source transmits the packet to the relay node and if the transmission is successful, this last one sends the ACK. Now, the relay node is responsible of delivering the packet to the sink and the source node can continue its monitoring task.

On the contrary, if the relay node is not available the source makes a new relay selection and transmission attempt after a time interval δ_k uniform for all nodes and that allows the node distinguish if the communication has failed.

To avoid many unsuccessful re-transmission attempts, MAD uses the reinforcement learning approach to decide whether the node should stay on the mission hoping to find a suitable relay, or to move toward the sink suspending the monitoring mission. The purpose of the policy is to find a proper trade-off between application availability and delivery guarantee.

2.4.1. Optimal relay node selection

To find the optimal relay node we must estimate the time needed by a packet to reach the relay.

The total expected time for a packet from source node *u* to be received at node *i* : $\Delta_{(u,i)} = (y - 1)\, \delta_k + \delta_{(u,i)}$

- $\delta_{(u,i)}$ → time for source *u* to send a packet to node *i* defined as the time between the hello$_{(i,*)}$ was generated on node *i* until node *u* received it.

- y → the number of trials meaning re-transmissions, source node *u* can try.
  Node *u* can estimate the transmission error probability towards node *i* defined as $e_{(u,i)} \in [0, 1]$, by considering the percentage of packets sent to *i* that had not been acknowledged by time *t*.
  We assume that a suitable relay node must show a successful delivery probability of at least $v_{(u,i)} \in [0, 1]$. Therefore, the probability of having at least one success out of *y* trials has to be ≥ to the hope *v* :

$$1- e_{(u,i)}(t)^{y} \geq v \quad \rightarrow \quad y \leq \frac{\log(1-v)}{\log_{e_{(u,i)}}(t)}$$

- δ_k → the time between two consecutive transmission attempts. $\delta_k > 2\delta_{(u,i)}$

- $(y - 1)\, \delta_k$ → the expected waiting time before a successful delivery.

Source node *u* selects its relay node on the basis of a score based mechanism. Firstly, it checks the following conditions for every node *i* for which it has received a hello packet.

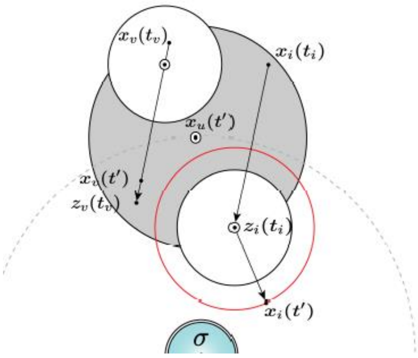


Figure 1. Optimal relay selection condition 1 and 2

<u>Condition 1</u>: The destination node *i* will stay in the communication range of source node *u* until it sends ack$_{(i,u)}$ back to *u* :

$$\|x_i(t+\Delta_{(u,i)}+\delta_{(i,u)})-x_u(t+\Delta_{(u,i)}+\delta_{(i,u)})\|+ \\ \dot{x}_i(t_i)\cdot((t-t_i)-\delta^i_z(t_i))\cdot\tau\leq\min(r_i,r_u)$$

The distance between the expected position of node *i* and souce node *u*, at the time of the ack packet reception, plus a pessimistic estimate in case node *i* has passed its advertised target position, to be greater than the mínimum communication radius.

<u>Condition 2</u>: Node *i* will be closer than source node *u* to the sink, at the moment of expected packet reception in node *i* :

$$\|x_i(t+\Delta_{(u,i)})-\sigma\|+\dot{x}_i(t_i)\cdot((t-t_i)-\delta^i_z(t_i))\cdot\tau< \\ \|x_u(t+\Delta_{(u,i)})-\sigma\|$$

The distance between the expected position of node *i* and source node *u*, at the time of the data packet reception, plus a pessimistic estimate in case node *i* has passed its advertised target position, to be less than the distance between the expected position of source node *u* and the sink.
Once node *i* receives the packet, it can still move in a direction that increases the distance of the packet from the sink before its delivery.

<u>Condition 3</u>: Fail safe mode is introduced optionally to prevent packet loss ensuring packet delivery, as it makes that nodes always opt for physical movement delivery whenever a packet residual lifetime is too close to the minimum time necessary to perform movement based delivery.

Node *u* assigns a score to every node that represents the quality of choosing it as a relay. The score is the distance of node *i* to the sink, at the time of the data packet reception:

$$\phi_{(u,i)}(t) \triangleq \|x_i(t + \Delta_{(u,i)}) - \sigma\|$$

The optimal relay of source *u* will be the one with lower score value chosen from the set of its candidate relays at time *t* :

$$relay_u(t) \triangleq \operatorname*{argmin}_{i \in \Gamma_u^*} \phi_{(u,i)}(t)$$

2.4.2. Reinforcement Learning

Through reinforcement learning, an agent gradually finds a decision policy on the basis of numerical reward signals it receives when performing allowed actions in the environment.

The goal of the agent is to learn a function mapping every state to the action, that allows him to maximize the cumulative future discounted reward.

MAD uses the Deep QNeural Network (DQN) trained to predict optimal action-values in a simulated environment fed with previous data. The node trains the weights of the action value function, the DQN, using the reward signal received for the action pursued during the previous *k* steps.

Action sapace: $\mathcal{A} \triangleq \{move, stay\}$

- As long as a node has packets to send, every *k* seconds it selects an action. assume that *k* is a multiple of δ_k. This to ensure that the pursued action will be evaluated for a number of transmission attempts.

State space: $\mathrm{s} \triangleq \langle s_{\mathrm{dist}}, s_{\mathrm{age}}, s_{\mathrm{dens}}, s_{\mathrm{meet}}, s_{\mathrm{mob}} \rangle \in \mathcal{S}$

- Each state feature is evaluated at the time they are queried, except for the mobility and local density, which consider an average of the most recent time window.
  - **Sink distance**: measures the distance of the node from the sink.
  - **Oldest packet age**: measures the age of the oldest packet in the buffer.
  - **Local density**: reflects the node density in the communication range of a node.
  - **Probability of encounter**: measures the probability of meeting at least one node in the path towards the next target of the node. The estimation of this probably in a tile is exchanged among the nodes and it is initialized with a high probability for all the tiles, to encourage exploration of the map in the initial steps of the simulation.
  - **Mobility**: measures the variability of the nodes. It is proportional to the rate of variation of the position of the neighbor nodes.

- All the domains of eact state feature are normalized in the range [0,1] to make them have equal relevance for the learning process.

Reward System:

- When a node needs to transmit packets, it determines its state and performs an action which results in a reward from the environment and a new landing state. The reward function is used by the agent to learn what it should do at best.

- The reward of action *a* while being in state *s* is the following:

$$r(\mathrm{s}, a) \triangleq \left(\frac{a\, s_{\text{dist}} + (1-a)\, s_{\text{meet}}}{s_{\text{age}}}\right)^{(1-2a)} \alpha^{a} \rho^{(1-a)}$$

- For a=0 (stay) → $\frac{\mathrm{Smeet}}{\mathrm{Sage}} \cdot \rho$
  Encourage mission exploration in case of lower urgency of packet delivery and high probability of meeting at least a relay while heading to the target.

- For a=1 (move) → $\frac{\mathrm{Sdist}}{\mathrm{Sage}} \cdot \alpha$
  Encouraging movement in case of high proximity to the sink and urgency of packet delivery.

- The parameters α, $\rho \in [0; 1]$ reflect the importance of mission availability and importance of delay respectively. Intermediate values are used to obtain the most suitable trade-off between system responsiveness and mission availability.

2.1. Delay critical and non-delay critical applications for FANETs

FANETs have several applications an each one requires specific routing design considerations. Looking into an important feature, which is the latency we can talk about delay critical and non-delay critical applications.

Critical delay applications have urgent communications and therefore tolerate a bounded delay. In these scenarios, we can find emergency applications like for example, disaster relief operations, which can require accessing extremely harsh environments and cover large geographical areas. They are typically constrained by the potential for human loss if not executed quickly. Therefore, these applications are non-delay tolerant, the data has to go from the source to the destination node as fast as possible to be able to make decisions and act quickly. For example, to search for survivors after an earthquake.

On the other hand, non-delay critical applications are delay tolerant, which means that having a low delay value is not essential. In these scenarios, we can find agriculture applications used for example, to monitor crops. The fact that they tolerate higher delays gives them advantages in other aspects. For example, they may have higher packet delivery ratio.

3. Methodology and Results

To study the performance of the MAD routing protocol, we have simulated two different scenarios based on Python code previously develop by the research group of the department. In the first one, we have simulated a UAV network with homogenous packet deadline, meaning that all packets are generated with the same TTL value. The second one is an emergency scenario with heterogeneous packet deadlines. Based on the urgency of the local findings, one part of the area is less tolerant to the delay than the other, and therefore, packets generated on the first half of the area will have a packet deadline whereas the ones generated on the other part will have another.

The duration of each simulation is of 1 hour long missions and we have executed 4 runs for each simulation using 4 seeds to initialize the internal pseudo-random number generator. With a fixed seed we make sure that the same series of calls to ‘RandomState’ methods will always produce the same results.

For each simulated scenario, we have studied the following parameters: average packet delay, packet delay divided by packet deadline and packet delivery ratio, all in function of UAV speed. Using the configuration of the first scenario, we have also studied the packet delivery ratio in function of packet deadline. Finally, on the second scenario we have also analysed in which part of the area, the delay or non-delay tolerant, where the delivered packets generated.

We have considered interesting to analyse the previous parameters using two different mobility models to study how the different mobility patterns affect them. Therefore, to represent the movement of UAVs, we have used the random waypoint mobility model (RWPMM) and the Brownian mobility model (BMM).

On the following paper [2] it is stated that “there exists a smooth tradeoff between the delay and capacity in ad-hoc networks under the RWPMM, whereas there is virtually no tradeoff under the BMM”. The authors believe that this difference is due to the different mobility patterns. In RWPMM, “nodes move “purposefully””, each one chooses a random end position within the area of simulation and moves towards it along a straight-line path. When the node reaches the destination, it then moves to another randomly end position. Therefore, nodes can cover large distances in relatively short time. On the contrary, in BMM, “nodes always wander around like “drunkards””, they constantly undergo small random movements around a local neighbourhood for a large duration of time. Authors expose that “it is intuitive to believe that reducing the mobility delay under the Brownian motion model would be more difficult”.

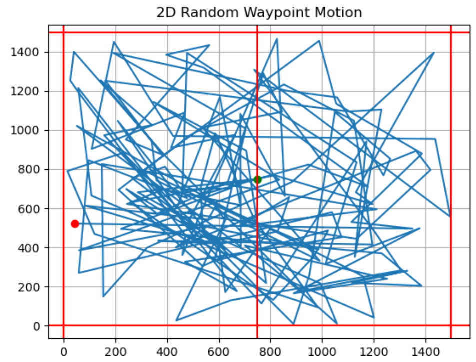


Figure 2. 2D Random waypoint motion starting on coordinates (750,750)

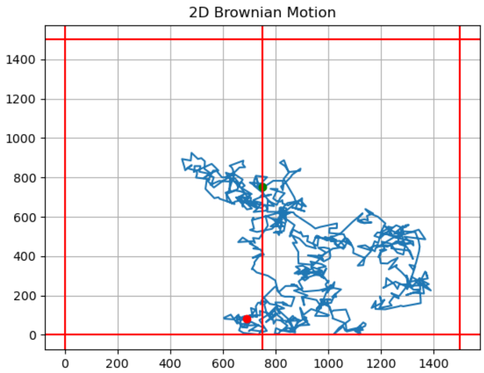


Figure 3. 2D Brownian motion starting on coordinates (750,750)

3.1. First simulated scenario

To study the performance of MAD in a UAV network with homogenous packet deadline, with Python we have simulated a scenario of 1.500 meters square, a monitoring area of 2,25 km^2. We have considered 4 runs of 1 hour long missions.

We have used a data collection sink on coordinates (750,0) meters and 15 UAVs moving inside the boundaries at 8 m/s and following first, a random way-point mobility model and then, a Brownian mobility model. The first six UAVs start their mission on coordinates (750,750) meters and the last eight on coordinates (750,1500) meters.

Following the settings on [1], nodes have a transmission range of 200 meters and unlimited buffer size and energy availability. We have set an event generation rate of 7 events per minute, modelling random events occurring throughout the area of interest with a uniform packet deadline. We consider a retransmission waiting time δ_k = 1,5 seconds, making decisions every k = 15 seconds with a required delivery probability v = 0,95 and a hello packet rate of 0,75 packets/second.

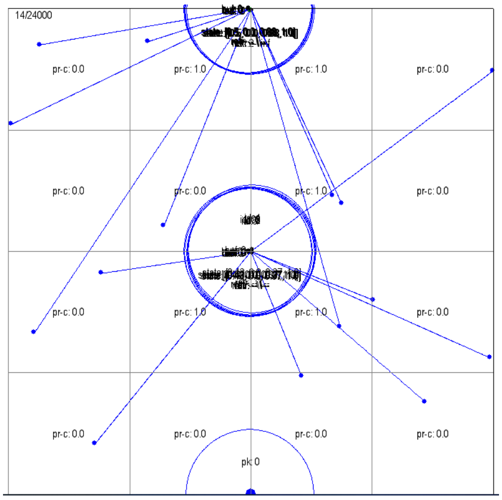


Figure 4. First simulated scenario with RWPMM

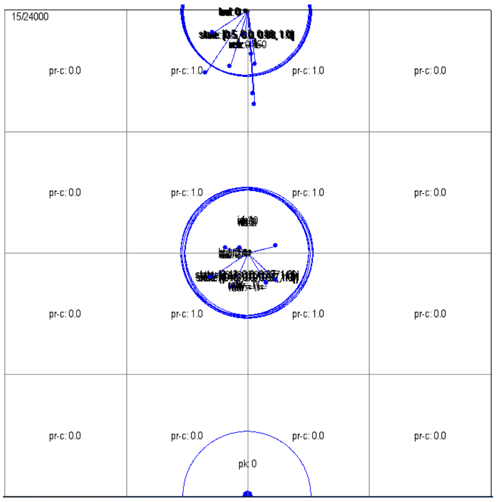


Figure 5. First simulated scenario with BMM

In Figure 4 and Figure 5, we observed the area at the initial steps of the simulation. We can see the position of the sink and the initial positions of the UAVs, drawn with their transmission range and with the vector to their next position.

3.1.1. Results

3.1.1.1. Packet delivery ratio in function of packet deadline

In the next figures, we study the packet delivery ratio of the UAV network in function of the packet deadline (TTL). We execute a simulation for each TTL value 30; 60; 150; 300; 600; 900 seconds.

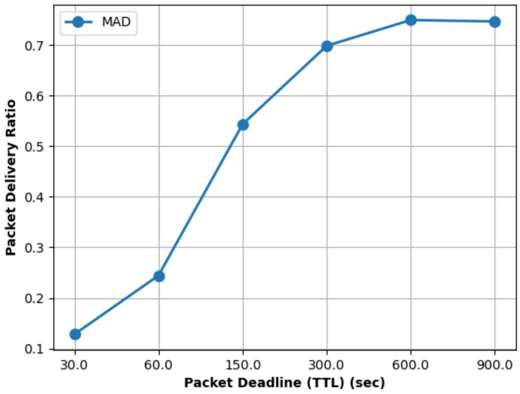


Figure 6. Packet delivery ratio in function of packet deadline with RWPMM

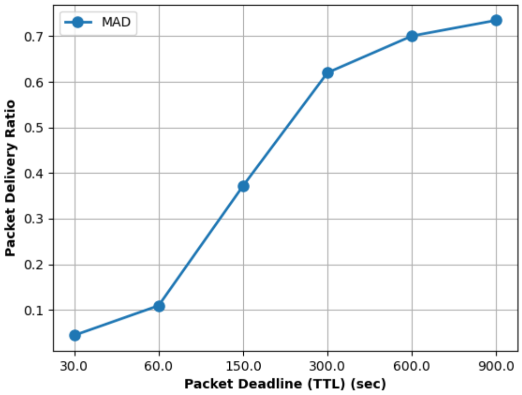


Figure 7. Packet delivery ratio in function of packet deadline with BMM

In both figures, we see that when the packet deadline is low, the algorithm shows a very low delivery ratio. In Figure 6, for a TTL value of 60 seconds the packet delivery ratio is about 0,24 and in Figure 7 is approximately 0,11 which means that many of the generated packets are being lost. As the packets have a short time period of life, the UAVs are not able to carry all of them to the depot on time, so the majority will expire. When the TTL is 150 the packet delivery ratio is low, 0,54 for Figure 6 and 0,37 for Figure 7, which means we are losing for Figure 6 half and for Figure 7 more than half of the packets generated in the UAV network.

If we calculate the time require for a packet generated at the farthest point from the sink, like the one located in the position (0,1500) meters, to be delivered in the depot which is located in the position (750,0) meters, we see that moving at 8 m/s a drone will require 210 seconds. So clearly, there are a few devices with respect to the size of the area, and it is not enough to ensure the delivery ratio.

On the contrary, we see that when we increase the value of the packet deadline, the performance of MAD improves. In Figure 6, we obtained that for a TTL value of 300 seconds, 5 minutes, the packet delivery ratio is about 0,69 and in Figure 7 is approximately 0,62, which means that many of the packets generated are being delivered to the sink. For higher values of the TTL like 900 seconds, 15 minutes, we see that in Figure 6 the packet delivery ratio is 0,75 and in Figure 7, is 0,74 which is a high ratio.

Critical delay applications tolerate a bounded delay, which is reflected in the TTL value. This one has to be low, like 300 seconds [1]. We have observed that for this packet deadline, the packet delivery ratio is low and the maximum value obtained is 0,69 for Figure 6 and 0,62 for Figure 7. On the other hand, for non-critical delay scenarios where the tolerance of the packet deadline is bigger like 900 seconds or more, we have that the packet delivery ratio is higher between 0,75 for Figure 6 and 0,74 for Figure 7.

We appreciate that with RWPMM we obtain better packet delivery ratio values than with BMM, but this behaviour is softened for high speeds.

To sum up, when the packet deadline is higher packets stay more time on the network, so UAVs have more time to bring most of the packets generated to the depot obtaining higher values of the packet delivery ratio. On the other hand, when the packet deadline is lower, packets have a short lifetime so UAVs are not able to deliver many of the packets generated, as most of them will expire before the delivery.

3.1.1.2. Average packet delay

For the following figures, we study the average packet delay of the UAV network, meaning the average time from when a package is generated until it is delivered to the depot, in function of the drone speed. We use the previous configuration, but now fixing the value of the TTL and varying the drone speed.

Following the experimental settings in [1], our speed range goes from 3 to 20 m/s which is 10,8 to 72 km/h. The average speed of a UAV can span from almost 30 to 460 km/h in a three-dimensional environment [3]. Therefore, we will look closely at the behaviour of our mid-range speeds around 10 m/s, which are 36 km/h respectively.

We execute a simulation for each drone speed value 3 ; 5 ; 8 ; 10 ; 15 ; 20 m/s for a TTL value of 900 seconds to study non-critical delay scenarios focusing on agriculture applications, and then with a TTL of 300 seconds to analyse critical delay scenarios with urgent communications [1], focusing on emergency applications.

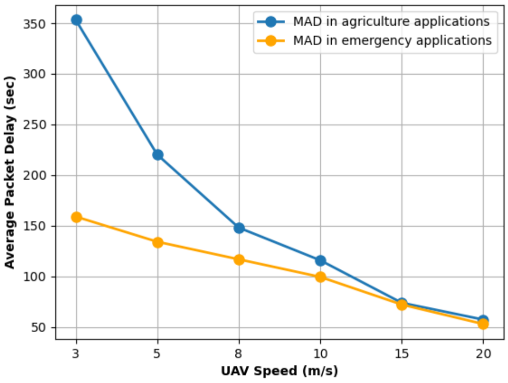


Figure 8. Average packet delay in function of UAV speed with RWPMM

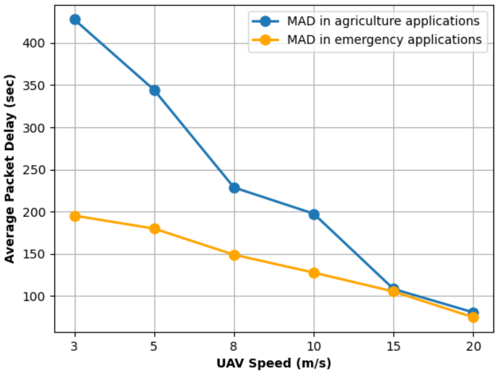


Figure 9. Average packet delay in function of UAV speed with BMM

As we obtain in both figures, it is clear that the average packet delay for both applications improves when the speed of the drones increases, as it takes less time for the moving nodes with onboard packets to bring them to the sink.

On agriculture applications when drones move at low speeds like 3 m/s the delay in Figure 8 is 353 seconds and in Figure 9 is 427 seconds. As the speed increases the delay decreases and for 10 m/s we obtain a better delay of 115 seconds in Figure 8 and of 197 seconds in Figure 9. For high-speed values like 20 m/s, the delay in Figure 8 is 57 seconds and in Figure 9 is 80 seconds, which are low values. We appreciate that with RWPMM we obtain better values than with BMM.

Agriculture applications are delay tolerant, meaning that they do not require low delays. Based on both figures, we can say that for this application it is not essential to use drones moving at high speeds, as we do not need to deliver packets as quickly.

On the other hand, for emergency applications, we obtain that for low speeds like 3 m/s the delay in Figure 8 is 158 seconds and in Figure 9 is 195 seconds, which are quite high values. As the speed increases the delay decreases and for 10 m/s we obtain a much better delay of 99 seconds in Figure 8 and of 127 seconds in Figure 9. For high-speed values like 20 m/s, the delay in Figure 8 is 52 seconds and in Figure 9 is 74 seconds, which are very low values. Again, we appreciate that with RWPMM we obtain better values than with BMM.

Comparing the results, the agriculture application shows a higher average packet delay with respect to the emergency application. When drones move at 3 m/s the delay in the agriculture scenario for Figure 8 and 9 is more than 2 times the value in the emergency scenario which represents an increase of the delay 123,42% and 118,97% respectively. This behaviour is softened as the speed increases. For high-speeds, the difference of delay between both applications starts to be quite insignificant. For example at 20 m/s the difference is very low just 5 seconds in Figure 8 and 6 seconds in Figure 9, which represents an increase of the delay of just 9,62% and 8% respectively.

3.1.1.3. Average packet delay divided by the packet deadline

In the next figures, we study the average packet delay divided by the packet deadline in function of the drone speed. For the plotting, we use the simulations obtained in the last experiment.

We know that for a homogenous TTL when the average packet delay is low we will obtain a good relation between average packet delay and packet deadline, meaning that drones are carrying the packets to the sink quickly regarding the packets’ lifetime. On the other hand, when the average packet delay is high it means drones are taking a long time to deliver the packets regarding their lifetime and those may even be about to expire, this is for very high values close to 1.

As we are dividing the average packet delay for a constant in each application, we expect to find in the applications of Figure 10 and Figure 11 the same shapes of the plot as in Figure 8 and Figure 9.

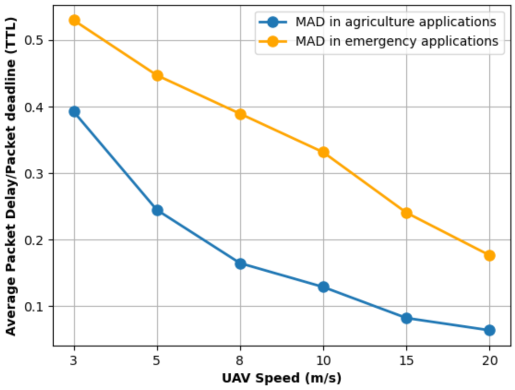


Figure 10. Average packet delay divided by the packet deadline in function of UAV speed with RWPMM

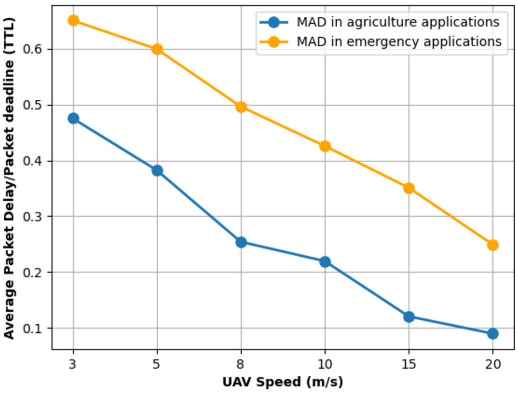


Figure 11. Average packet delay divided by the packet deadline in function of UAV speed with BMM

On agriculture applications, we used a uniform TTL value of 900 seconds. We see that as the speed of the UAVs increases, the relation between average packet delay and packet deadline decreases. When drones move faster like at 20 m/s this relation is in both figures less than 0,1 which is a really great value. For low speeds like 3 m/s this relation is in Figure 10 about 0,39 and in Figure 11 about 0,47. This are higher values especially with BMM and it means that packets stay on the UAV network for quite a long time regarding their lifetime.

On emergency applications, we used a uniform TTL value of 300 seconds. We can appreciate that the relation between average packet delay and packet deadline also decreases as the speed increases. For low speeds like 3 m/s this relation is in Figure 10, about 0,52 and in Figure 11, about 0,65 which is a high value because it means packets stay in the network more than half of their lifetime before they are delivered to the sink. When drones move faster like at 20 m/s this relation decreases to 0,17 for Figure 10 and to 0,24 for Figure 11 which are better values. Therefore, even though the delay of the packets is not too high, as we have analysed in Figure 8 and Figure 9, the fact that the packet deadline is very low causes the relation between average packet delay and packet deadline to be quite high especially for low speeds.

Comparing the results, in both pictures the agriculture application shows lower values of this relation with respect to the emergency application, and as we have explained before the lower the relation the better. When drones move at 3 m/s the relation in the emergency scenario increases 33,33% with respect to the agriculture scenario for Figure 10 and 38,3% for Figure 11. As the speed increases this behaviour is accentuated and at 20 m/s the increase is of 183,33% for Figure 10 and of 200% for Figure 11. Also, we appreciate that with RWPMM we obtain better values than with BMM.

3.1.1.4. Packet delivery ratio in function of UAV speed

In the next figures, we study the packet delivery ratio of the UAV network in function of the drone speed. For the plotting, we use the simulations obtained in 3.1.1.2.

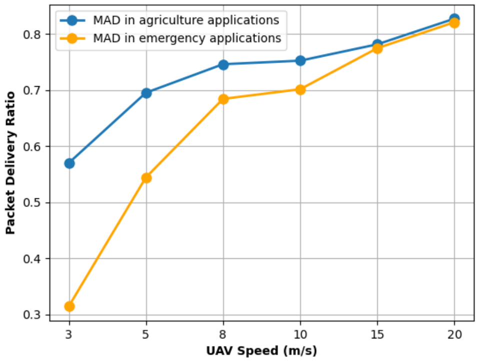


Figure 12. Packet delivery ratio in function of UAV speed with RWPMM

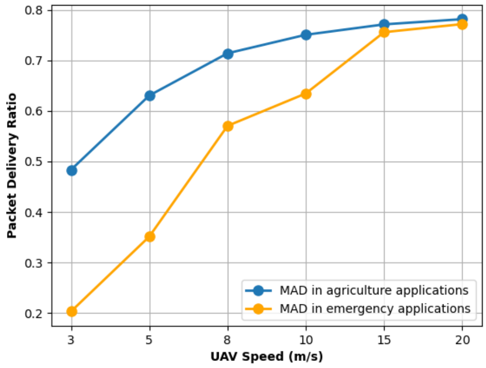


Figure 13. Packet delivery ratio in function of UAV speed with BMM

As we obtain in both figures, it is clear that the packet delivery ratio for both applications improves when the speed of drones increases, as it take less time to bring the packets to the sink.

On agriculture applications when drones move at low speeds like 3 m/s this ratio in Figure 12 is 0,56 and in Figure 13 is 0,48 which means that almost half of the packets generated are not being delivered. As the speed increases the delivery ratio increases and for 10 m/s we obtain a great value of 0,75 in Figure 12 and of 0,75 in Figure 13. For high-speed values like 20 m/s, the ratio in Figure 12 is 0,82 and in Figure 13 is 0,78, which are very high values.

On the other hand, for emergency applications, we obtain that for low speeds like 3 m/s the packet delivery ratio in Figure 12 is 0,31 and in Figure 13 is 0,2, which are quite low values. As the speed increases the ratio increases and for 10 m/s we obtain a better value of 0,7 in Figure 12 and of 0,63 in Figure 13. For high-speeds like 20 m/s, the ratio is very high in Figure 12 of 0,82 and in Figure 13 of 0,77.

Comparing the results, the agriculture application shows a higher packet delivery ratio with respect to the emergency application. When drones move at 3 m/s the ratio in the agriculture scenario for Figure 12 is almost 2 times the value in the emergency scenario and in Figure 13 more than 2 times which represents an increase of 80,65% and 140% respectively. This behaviour is softened as the speed increases. For high-speeds, the difference of ratio between both applications is quite insignificant. Also, we appreciate that with RWPMM we obtain better values than with BMM.

3.2. Second simulated scenario

To study the performance of MAD in a UAV network with heterogeneous packet deadline, with Python we have simulated an urgency scenario of an area of 2,25 km^2, a 1.500 meters square, that has different requirements depending on the local findings. This scenario could be monitoring a forest where a fire is taking place, this would be the non-delay tolerant area, and the part of the forest that has still not been reached by the fire, but that has risk of being caught by it, would be the delay tolerant area.

The square is divided in two rectangles of 750 meters wide and 1500 meters high each. The first rectangle represents the delay tolerant area where packets are generated with a packet deadline of 900 seconds, whereas the second one represents the non-delay tolerant area, where the fire is taking place, and that has a lower packet deadline, specifically 300 seconds.

We have considered 4 runs of 1 hour long missions.

We have added the corresponding modifications so that MAD gives priority of delivery to the packets with minimum residual lifetime. Therefore, it looks for the most urgent packet in the buffer, which is the one whose age respect its deadline is maximum. If this relation value is near 1 this means the packet is close to its expiration time.

We have used a data collection sink on coordinates (750,0) meters and 15 UAVs moving inside the boundaries at 8 m/s and following first, a random way-point and then, a Brownian mobility model. The first six UAVs start their mission on coordinates (750,750) meters and the last eight on coordinates (750,1500) meters.

Following the settings on [1], nodes have a transmission range of 200 meters and unlimited buffer size and energy availability. We have set an event generation rate of 7 events per minute, modelling random events occurring throughout the area of interest. We consider a retransmission waiting time δ_k = 1,5 seconds, making decisions every k = 15 seconds with a required delivery probability v = 0,95 and a hello packet rate of 0,75 packets/second.

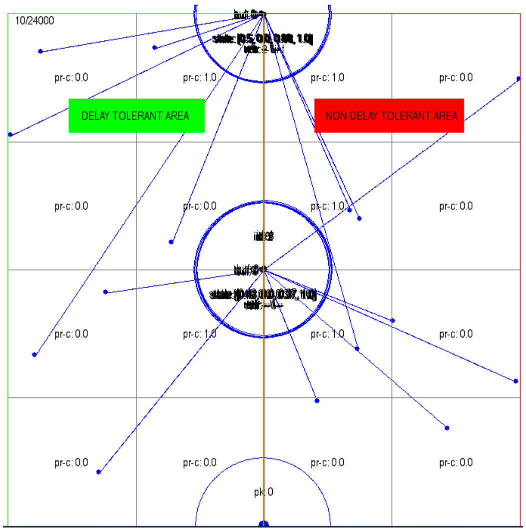


Figure 14. Second simulated scenario with RWPMM

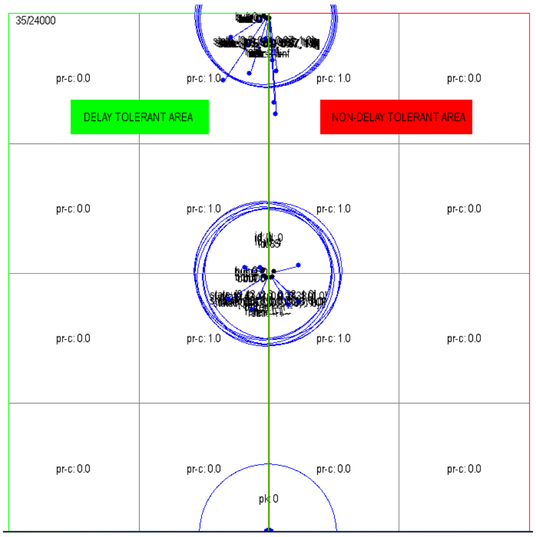


Figure 15. Second simulated scenario with BMM

In Figure 14 and Figure 15, we observed the area at the initial steps of the simulation. We can see the position of the sink and the initial positions of the UAVs, drawn with their transmission range and with the vector to their next position.

3.2.1. Results

3.2.1.1. Average packet delay

For the following figures, we study the average packet delay of the UAV network in function of the UAV speed. We execute a simulation for each drone speed value 3; 5; 8; 10; 15; 20 m/s.

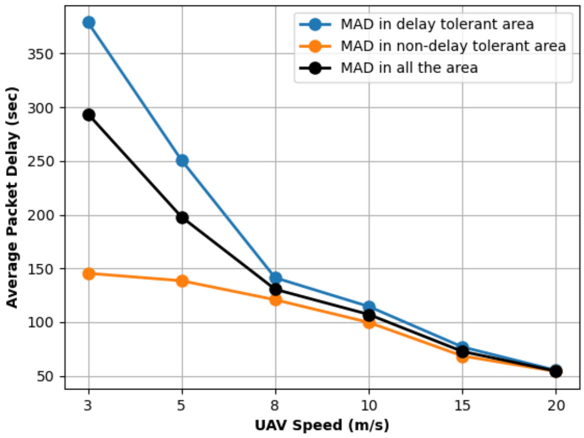


Figure 16. Average packet delay in function of UAV speed with RWPMM

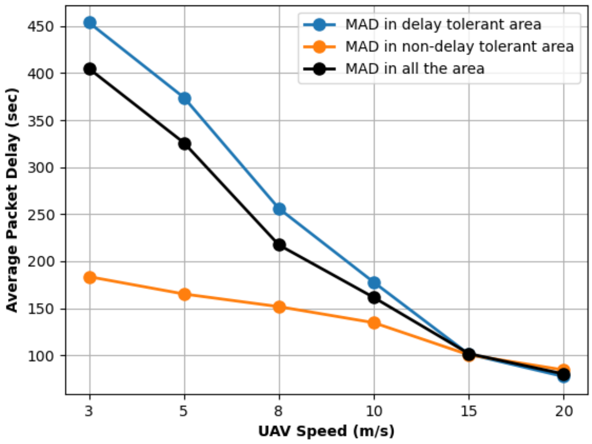


Figure 17. Average packet delay in function of UAV speed with BMM

In Figure 16, we see that when drones move at 3 m/s the delay for all the area is 293 seconds, while in Figure 17 is 404 seconds. If we look closely at each area, we have that for this same speed, the packets generated on the delay tolerant area have a delay of 378 seconds in Figure 16 and 453 seconds in Figure 17. On the other hand, non-delay tolerant packets, generated on the other area, have a delay of 145 seconds in Figure 16 and 183 seconds in Figure 17. Therefore, the packets generated in the non-delay tolerant area, are delivered to the sink faster than the packets generated in the delay tolerant area. We observed that the delays are higher when using the BMM.

As the speed increases, the average packet delay of the entire network decreases. This behaviour is also fulfilled by each subarea separately.

Looking at the whole network, we obtain that for 10 m/s the delay in Figure 16 is 107 seconds and in Figure 17 a value higher of 161 seconds. Focusing again in each area, we see that now the difference of delays between them is very small, about 15 seconds for Figure 16, and 40 seconds for Figure 17.

For high-speed values like 20 m/s, the delay in Figure 16 is around 54 seconds and in Figure 17 about 80 seconds, which are low values especially with RWPMM. The difference of delays between each area is minimal.

As obtained, it is clear that the average packet delay improves as the speed of UAVs increases and that for low speeds the non-delay tolerant area shows a better performance regarding the delay than the delay tolerant. We also see that the scenario using the Random waypoint mobility model shows better delay values in respect to the scenario with the Brownian approach.

3.2.1.2. Average packet delay divided by the packet deadline

In the next figures, we study the average packet delay divided by the packet deadline in function of the UAV speed.

We have considered all the packets delivered to the depot and for each one we have divided its delay, the time between the instant it is generated until it is delivered to the depot, by their deadline, which depends on the part of the area where the packet was generated. In case it is the first half of the area, the delay tolerant, the TTL is 900 seconds, whereas if it was generated in the second half, the non-delay tolerant area, the TTL is 300 seconds.

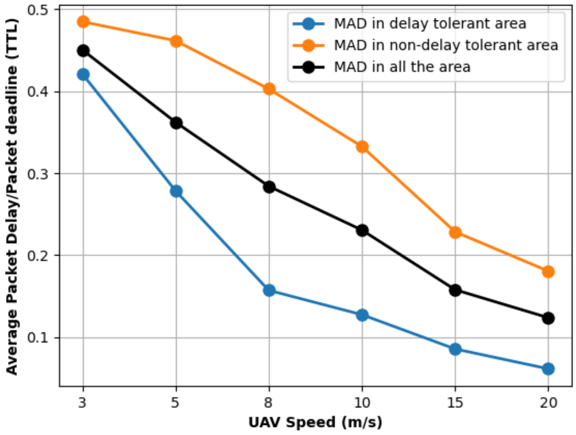


Figure 18. Average packet delay divided by the packet deadline in function of UAV speed with RWPMM

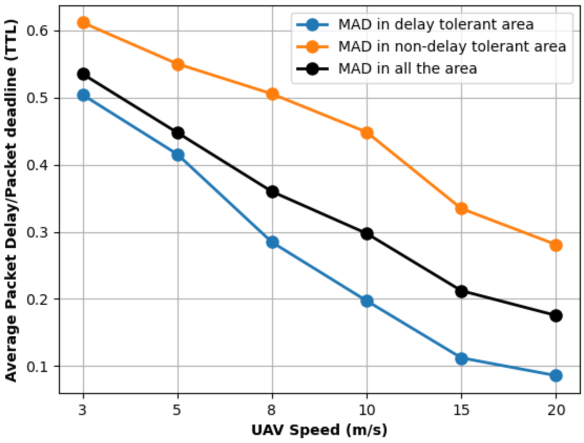


Figure 19. Average packet delay divided by the packet deadline in function of UAV speed with BMM

Now, unlike the first simulated scenario, we do not expect the relation between average packet delay and packet deadline for all the area to have the same shape as the average packet delay for all the area, as now the packet deadline is not uniform, so the denominator is not a constant.

In both figures, we see that as the speed of the UAVs increases, the relation between average packet delay and packet deadline decreases. When drones move faster like at 20 m/s this relation is about 0,12 in Figure 18 and 0,17 in Figure 19 which are low values.

If we look closely at each area, we have that for all the speeds, the packets generated on the delay tolerant area have the worst relation between average packet delay and packet deadline than the non-delay tolerant area.

For low speeds like 3 m/s this relation is 0,45 in Figure 18 and 0,53 in Figure 19 which is higher when using the RWPMM.

As obtained, it is clear that the relation between average packet delay and packet deadline improves when the speed of the drones increases and that the non-delay tolerant area shows a better performance than the delay tolerant. We also see that the scenario using the RWPMM model shows better values in respect to the scenario with the BMM approach.

3.2.1.3. Packet delivery ratio

In the next figures, we study the packet delivery ratio in function of the UAV speed. For the plotting, we use the simulations obtained in 3.1.1.1.

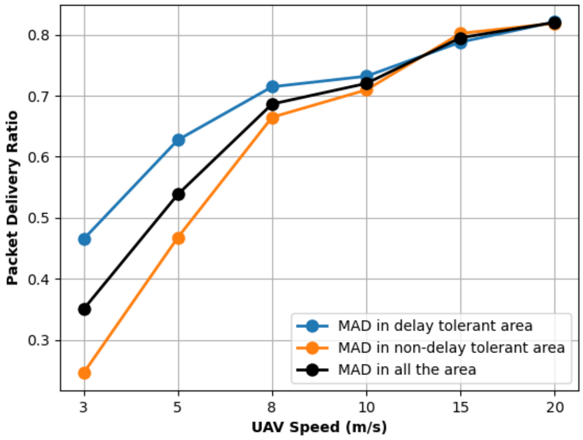


Figure 20. Packet delivery ratio in function of UAV speed with RWPMM

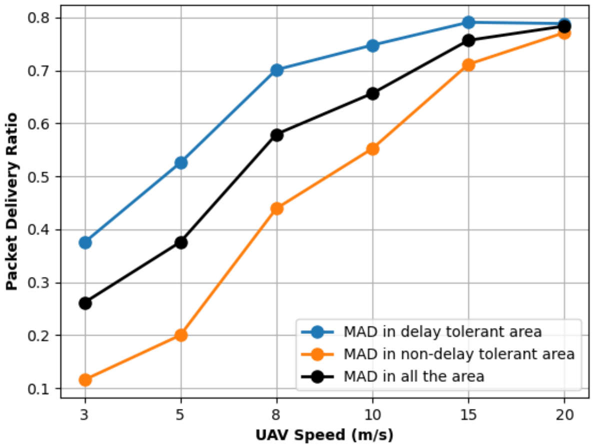


Figure 21. Packet delivery ratio in function of UAV speed with BMM

In both figures, we see that when the speed is low, the algorithm shows a low packet delivery ratio. In Figure 20, for 3 m/s this ratio is about 0,35 and in Figure 21 is approximately 0,26 which means that many packets are being lost. If we look closely at each area, we obtain that the delay tolerant area shows a better packet delivery ratio than the non-delay tolerant area. The first half area has a ratio of 0,46 in Figure 20 and 0,37 in Figure 21, and the second half of 0,24 for Figure 20 and 0,11 in Figure 21.

On the contrary, we see that when we increase the speed, the performance of MAD improves. In Figure 20, we obtained that for a speed value of 20 m/s the packet delivery ratio is about 0,82 and in Figure 21 is approximately 0,78, which means that many of the packets generated are being delivered to the sink. For high-speed values the ratio for each area is almost the same for Figure 20 while Figure 21, keeps up with the behaviour previously mentioned in which the delay tolerant area shows a better delivery ratio than the non-delay tolerant area, but more slightly.

As obtained, it is clear that the packet delivery ratio improves when the speed of the drones increases and that the non-delay tolerant area shows a better performance than the delay tolerant except for high speeds with RWPMM where they show almost identical ratios.

3.2.1.4. Packets delivered to the sink

For the following figures, we analyse the packets that are being delivered to the sink in function of the UAV speed. The aim is to know how many packets generated on each area are finally delivered to the sink.

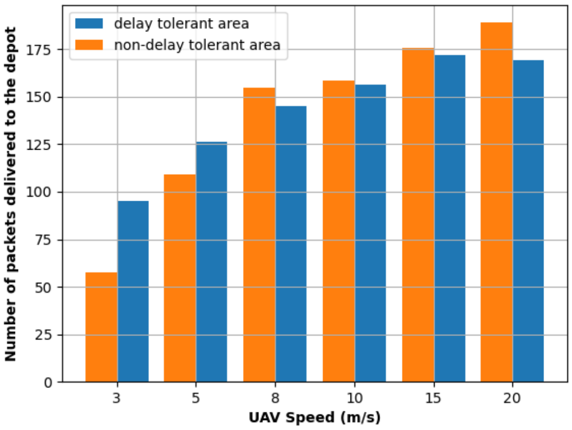


Figure 22. Number of packets delivered to the depot in function of UAV speed with RWPMM

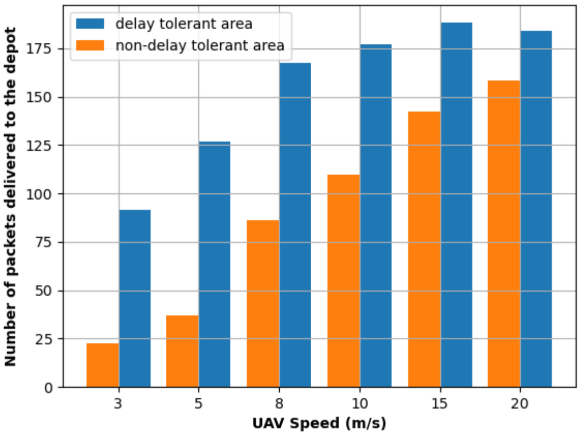


Figure 23. Number of packets delivered to the depot in function of UAV speed with BMM

In both pictures, we see that as the speed increases, it also increases the delivered packets generated on the non-delay tolerant area. In Figure 22, we obtain that for low speeds between 3 and 5 m/s this type of packets are lower regarding the number of packets generated on the non-delay area, but for speeds from 8 m/s the opposite happens. At 3 m/s we see that the non-delay tolerant packets are 57 and at 20 m/s this value increase until 189. In Figure 23, we also see that when drones move faster, the number of non-delay tolerant packets brought to the depot increases. In this case, though, the number of non-delay tolerant packets delivered are much less regarding the delay tolerant packets. At 3 m/s we see that the non-delay tolerant packets are 23 and at 20 m/s this value increase until 158.

To sum up, we see that although in both figures the number of non-delay tolerant packets delivered to the sink increase as the UAV speed increases, when using the BMM these values are still lower in respect to the delay tolerant packets. For low speeds, the UAV network is not able to guarantee a good packet delivery ratio for the non-delay tolerant area, as also observed in Figure 20 and Figure 21. Therefore, this scenario works well for high speeds where it is guaranteed a quite balanced delivery between packets from both areas.

4. Budget

On this section, we present the budget of the project divided in personal, direct, indirect and auxiliary costs.

Regarding to the personal costs, we have a junior engineer working for the entire duration of the project, about 4 months working 20 hours a week. The employee earns 12€/hour so its total salary is 5.107,2€ as its present here:

Position	Wages/ month	Social Security (33%)	Total cost/month	Duration	Total cost
Junior engineer	960€	316,8€	1.276,8€	4 months	5.107,2€
TOTAL					**5.107,2€**

The direct costs refer to the materials and tools used to develop the project. We are using a laptop and a free license for educational use of PyCharm.

Material list	Cost/unit	Number of units	Amortization time	Amortization	Total cost
Laptop	780€	1	5 years	13€/month	52€
PyCharm software	0€	1	-	-	0€
TOTAL					**52€**

The indirect costs refer to the general costs needed to carry out the project like the cost of the local, the supply of water and electricity...

	Cost/month	Total cost
Local rent	460€	1.840€
Water	15€	60€
Electricity	70€	280€
Internet	30€	120€
Cleaning service	160€	640€

	Cost/month	Total cost
TOTAL		**2.940€**

The auxiliary costs are contemplated for unexpected incidents. It is a 10% of the sum of personal, direct and indirect costs.

Type of cost	Value
Personal	5.107,2€
Direct	52€
Indirect	2.940€
Auxiliary	890,92€
TOTAL	**8.990,12€**

Finally, we have that to develop this project which has a duration of 4 months the budget necessary is 8.990,12€.

5. Conclusions:

After performing the described simulations, we can conclude that for the first simulated scenario where the UAV network had homogenous packet deadline:

- The packet delivery ratio increases as the packet deadline increases. We saw that non-critical delay applications that have higher TTL have a better ratio than critical delay applications, which have lower TTL. Using RWPMM the ratio goes from 0,13 for a TTL of 30 seconds to 0,75 for a TTL of 900. Using BMM the ratio goes from 0,04 for a TTL of 30 seconds to 0,74 for a TTL of 900.Therefore, with RWPMM we obtained better values than with BMM, especially for low speeds.
- Studying the emergency and agriculture applications, we saw that:
    - The average packet delay decreases as the UAV speed increases. The emergency scenario showed a better performance for this parameter than the agriculture scenario. Using RWPMM for agriculture this parameter goes from 353 seconds for 3 m/s to 57 for 20 m/s, and for emergency applications from 158 to 52 seconds. Using BMM for agriculture this parameter goes from 427 seconds for 3 m/s to 80 for 20 m/s, and for emergency applications from 195 to 74 seconds. Therefore, with RWPMM we obtained better values than with BMM, especially for low speeds.
    - The average packet delay divided by the packet deadline decreases as the UAV speed increases. Now, the emergency scenario showed a worse performance than the agriculture scenario. Using RWPMM for emergency this parameter goes from 0,52 for 3 m/s to 0,17 for 20 m/s, and for agriculture applications from 0,39 to 0,07. Using BMM for emergency this parameter goes from 0,65 for 3 m/s to 0,24 for 20 m/s, and for agriculture from 0,47 to 0,09. Therefore, with RWPMM we obtained better values than with BMM.
    - The packet delivery ratio increases as the UAV speed increases. Emergency applications also showed a worse performance for this ratio than agriculture applications except for high speeds. Using RWPMM for emergency this parameter goes from 0,31 for 3 m/s to 0,82 for 20 m/s, and for agriculture applications from 0,56 to 0,82. Using BMM for emergency this parameter goes from 0,2 for 3 m/s to 0,77 for 20 m/s, and for agriculture from 0,48 to 0,78. Therefore, with RWPMM we obtained better values than with BMM.
    - We conclude that for low speeds the advantage of having higher delays like the agriculture applications regarding the emergency applications, is that we are getting more packets meaning higher packet delivery ratio.

For the second simulated scenario where the UAV network had heterogeneous packet deadlines:

- The average packet delay decreases as the UAV speed increases. The non-delay tolerant area showed a better performance for this parameter than the delay tolerant area, especially for low speeds. Using RWPMM this parameter for all the area goes from 293 seconds for 3 m/s to 54 for 20 m/ whereas using

BMM, it goes from 404 to 80 seconds. Therefore, with RWPMM we obtained better values than with BMM.

- The average packet delay divided by the packet deadline decreases as the UAV speed increases. The non-delay tolerant area now showed a worse performance than the delay tolerant area. Using RWPMM this parameter for all the area goes from 0,45 for 3 m/s to 0,12 for 20 m/s whereas using BMM it goes from 0,53 to 0,17. Therefore, with RWPMM we obtained better values than with BMM.
- The packet delivery ratio increases as the UAV speed increases. The non-delay tolerant area also showed a worse performance for this ratio than the delay tolerant area, especially for low speeds. Using RWPMM this parameter for all the area goes from 0,35 for 3 m/s to 0,82 for 20 m/s whereas using BMM it goes from 0,26 to 0,78. Therefore, with RWPMM we obtained better values than with BMM.
- The number of non-delay tolerant packets delivered to the sink increase as the UAV speed increases. With BMM, these values are much lower in respect to the delay tolerant packets, especially for low speeds. For 3 m/s there are 23 packets generated in the non-delay tolerant area in contrast to the 92 of the other area. At 20 m/s the value increase until 158 but is still low respect the 185 packets of the delay tolerant area. In contrast, with RWPMM, non-delay tolerant packets at high speeds are more than the delay tolerant packets. At 20 m/s we have 189 non-delay tolerant packets and 170 of the other area.

Bibliography:

[1] N. Bartolini, A. Coletta, A. Gennaro, G. Maselli and M. Prata, "MAD for FANETs: Movement Assisted Delivery for Flying Ad-hoc Networks," 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 2021, pp. 315-325, doi: 10.1109/ICDCS51616.2021.00038.

[2] Xiaojun Lin, G. Sharma, R. R. Mazumdar and N. B. Shroff, "Degenerate delay-capacity tradeoffs in ad-hoc networks with Brownian mobility," in IEEE Transactions on Information Theory, vol. 52, no. 6, pp. 2777-2784, June 2006, doi: 10.1109/TIT.2006.874544.

[3] G. Gankhuyag, A. P. Shrestha and S. Yoo, "Robust and Reliable Predictive Routing Strategy for Flying Ad-Hoc Networks," in IEEE Access, vol. 5, pp. 643-654, 2017, doi: 10.1109/ACCESS.2017.2647817.

[4] Bujari A, Calafate CT, Cano J-C, Manzoni P, Palazzi CE, Ronzani D. Flying ad-hoc network application scenarios and mobility models. International Journal of Distributed Sensor Networks. October 2017. doi:10.1177/1550147717738192

[5] M. Y. Arafat and S. Moh, "Routing Protocols for Unmanned Aerial Vehicle Networks: A Survey," in IEEE Access, vol. 7, pp. 99694-99720, 2019, doi: 10.1109/ACCESS.2019.2930813.

[6] D. Shumeye Lakew, U. Sa'ad, N. -N. Dao, W. Na and S. Cho, "Routing in Flying Ad Hoc Networks: A Comprehensive Survey," in IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1071-1120, Secondquarter 2020, doi: 10.1109/COMST.2020.2982452.

[7] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque and M. S. Hossain, "Routing in Flying Ad Hoc Networks: Survey, Constraints, and Future Challenge Perspectives," in IEEE Access, vol. 7, pp. 81057-81105, 2019, doi: 10.1109/ACCESS.2019.2923840.

Glossary

UAV: Unmanned Aerial Vehicle

MAD: Movement Assisted Delivery

FANET: Flying Ad-hoc Network

RL: Reinforcement Learning

MANET: Mobile Ad-Hoc Network

VANET: Vehicle Ad-Hoc Networks

AP: Access Point

BS: Base Station

GCS: Ground Control Station

PRP: Proactive Routing Protocol

RRP: Reactive Routing Protocol

Non-DTN: Non-delay tolerant network

DTN: Delay tolerant network

ACK: Acknowledge message

DQN: Deep QNeural Network

TTL: Time To Live

RWPMM: Random Waypoint Mobility Model

BMM: Brownian Mobility Model