

Improving Message Delivery in UAV-based Delay Tolerant Networks

Cintia Barroca, António Grilo and Paulo Rogério Pereira
INESC-ID / Instituto Superior Tecnico - Universidade de Lisboa,
Lisboa, Portugal

Email: cintia.barroca@tecnico.ulisboa.pt, antonio.grilo@inesc-id.pt, prbp@inesc.pt

Abstract—Delay Tolerant Networks (DTNs) are sparse networks where complete direct end-to-end paths between source and destination can seldom be established. Routing mechanisms in DTN rely on the nodes' mobility to connect disconnected nodes, carrying messages around the network to overcome path disconnection. The proactive DTN approach consists of introducing dedicated nodes whose only purpose is to establish communication between ordinary nodes, and relieve them from energy-consuming work, such as message routing and forwarding. This paper addresses the use of Unmanned Aerial Vehicles (UAVs) as DTN relays, introducing a proactive scheme called Deadline Triggered Pigeon with Travelling Salesman Problem with Deadlines (DTP-TSP-D). We envision a DTN where ground nodes can only communicate by flying UAVs with the capacity of carrying messages from one location to another. Each UAV either belongs to one node or to a cluster of nodes, and its role is to hover over their home-ground node (or ferrying around their home cluster) until they are triggered to deliver messages directed to other ground nodes. The triggering criterium is based on the deadlines of the messages present in the UAVs buffer, evaluating its ability to deliver all of them in time. It uses a developed TSP Genetic Algorithm to compute the route that achieves the most (timely) deliveries. The performance of DTP-TSP-D has been compared to dedicated node protocols found in the literature, namely SIRA and MRT-Grid. The performance metrics used were delivery ratio and average delay. The results show that DTP-TSP-D achieves higher delivery rates than its competitors, while keeping a consistent average delay.

I. INTRODUCTION

With their maneuverability and increasing affordability, Unmanned Aerial Vehicles (UAVs) have many potential applications in wireless communication systems, like providing cost effective wireless connectivity for devices without infrastructure coverage due to e.g., severe shadowing by urban or mountainous terrain, or damage to the communication infrastructure caused by natural disasters [1]. This paper addresses the use of UAVs as traffic relays in the context of Delay Tolerant Networks (DTNs).

Delay Tolerant Networks (DTNs) [2] are sparse networks where complete end-to-end paths between source and destination do not always exist. Routing mechanisms in DTN usually rely on the nodes mobility to bridge gaps in space and time, carrying messages around the network to overcome path disconnection using a store-carry-and-forward scheme. A message originated from the source is first forwarded to an intermediate node that is supposed to be (or get) closer to the

destination. Then the receiver stores the message locally and carries it for a while until a next contact is available.

Most solutions for routing in DTN are reactive approaches, where nodes rely on their inherent movements to disseminate data when they encounter each other, and their mobility is not controlled. Due to unpredictable mobility, this random behavior leads to low delivery rates and large delays. On the other hand, a proactive approach is to introduce extra nodes (messengers), which actively move around the network and regularly visit ordinary nodes, creating chances to re-connect disconnected nodes.

There are several reasons to choose such a proactive approach with extra auxiliary mobile nodes. Firstly, it can serve a variety of DTNs, especially in some challenging situations where it is impossible to deliver messages based on the movements of ordinary nodes only, like in disaster recovery, where mobile nodes (helicopters, UAVs, or personnel) equipped with communication devices capable of storing a large number of messages, can be commanded to follow a trajectory that interconnects disconnected user partitions. Secondly, these messengers can be controlled to provide predictable end-to-end service qualities, such as message delivery delay. Thirdly, these messengers are dedicated to message transmission tasks, which can relieve ordinary nodes from message routing and forwarding.

The early work starts with using a single messenger as a ferry, for forwarding messages in DTNs [3]. However, these single ferry approaches are not scalable in terms of traffic load, network size or geographic coverage, leading to solutions based on multiple ferries [4], [5], [6], [7]. The most basic extension of the simple ferry scheme is to have multiple ferries treated as identical and not require them to cooperate with each other [4]. Zhao et al [5] began to investigate the interaction between ferries and studied whether the multiple ferries should take the same (single) route or multiple routes, proposing four algorithms. Zhang et al [6], explored the design of traveling routes for multiple ferries, proposing three new schemes which vary on the node allocation manner. Later, Guo et al [8][9] proposed an alternative usage of auxiliary nodes. Instead of having *shared messengers*, they propose an approach in which all the ordinary nodes own at least one *dedicated messenger*, called pigeon due to its similarity to the ancient messaging system using homing pigeons.

This paper proposes a hybrid approach between shared and

dedicated messengers. A new scheme for DTN was developed, based on proactive schemes like Message Ferrying and Homing Pigeon, named Deadline Triggered Pigeon with Travelling Salesman Problem Deadlines (DTP-TSP-D). In this approach, the auxiliary nodes act as a shared messenger by ferrying over a cluster of nodes, collecting and delivering messages. However, it can also act as a dedicated messenger (pigeon), whenever it is triggered to do so, being detached to deliver messages outside its assigned cluster. As far as the authors know, this is the first time that a hybrid ferry-pigeon scheme is proposed, and also the first time that a proactive DTN scheme takes message deadlines into account for delivery route planning. Comparative performance evaluation is performed in a DTN scenario where ground nodes are far apart and the only means of communication is by flying UAVs with the capacity of carrying messages, from one location to another.

The rest of this paper is organized as follows: Section II presents a review of the related work. Section III presents the implementation and analysis of the algorithms present in the literature that will serve as term of comparison for the new approach. Section IV presents the implementation of DTP-TSP-D. Section V presents the simulation results to evaluate all the algorithms, followed by conclusions in Section VI.

II. RELATED WORK

This section presents the relevant related work, which comprises Delay Tolerant Networks (DTN) and the Traveling Salesman Problem (TSP).

A. Routing in DTN

Over the years, many DTN routing schemes have been proposed and a simple way to classify them can be by whether or not they use dedicated auxiliary nodes. In the category of reactive schemes, which do not single out special nodes for delivery purpose, message delivery only relies on the inherent movement of nodes. Epidemic routing [10] is one of the earliest schemes. It uses a flooding technique to spread messages throughout the network. This technique is guaranteed to find the optimal path for message delivery with the shortest delay, since it explores all available communication paths to deliver messages. However, flooding the network results in serious congestion. Later came more efficient alternatives, such as Spray and Wait [11], which limits the number of copies of each messages, or PROPHET [12], which uses past encounters to evaluate the probability of nodes meeting each other again. Other dissemination protocols are described in [13].

In the second category, proactive schemes introduce extra auxiliary nodes (called messengers), dedicated to collect and deliver messages. The typical proactive scheme is Message Ferrying [3], where the messengers are called ferries. The latter move according to fixed paths to connect disconnected nodes. When using multiple ferries, several approach approaches are possible: Zhao et al. [5] proposes four different algorithms, namely Single Route Algorithm (SIRA) which is simply having all available ferries follow one route, Multiple

Routing Algorithm (MURA) which extends its routes whenever it needs to reach a new node, Node Relaying Algorithm (NRA) which uses nodes as relays between ferry routes and Ferry Relaying Algorithm (FRA) where the ferries program their routes to encounter each other and exchange messages.

Later, Zhang et al. [6], concentrated their work on the multi-ferry route design, especially how nodes are allocated to multiple ferries and how multiple ferry routes are connected, coming up with two new multiple route (MRT) algorithms.

Guo et al. [8] took a different approach, using the extra auxiliary nodes as dedicated messengers. In their Homing Pigeon Based DTN routing (HoP-DTN), each node in the network owns a dedicated messenger called pigeon due to its similarity to the ancient messaging system. Messages are generated at the source (home), and when a certain number of messages are ready to be delivered, the corresponding pigeon computes a TSP solution to evaluate the best path to go through all the destinations.

Later, in [9], the same authors evaluate the extension of their previous model using multiple pigeons per node.

None of the mentioned approaches uses deadlines as a factor for maximizing deliveries or minimizing delays.

B. Traveling Salesman Problem

The TSP consists of an hypothetical salesman looking for the best path, the one minimizing the total length of the trip, to visit a set of cities, starting from a given one, the hometown, stopping only once at each city, and ending up at the initial starting location. It is defined as a permutation problem with the objective of finding the path with the shortest length (or the minimum cost). It can be modeled as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the sum of the lengths of the edges it comprises. Figure 1 illustrates the TSP with an example.

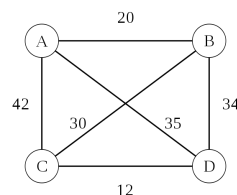


Figure 1. Undirected graph example.

If n is the number of cities to be visited, then the number of possible routes is $(n - 1)!$. Following this basic formulation, an exponential relationship exists between the number of cities and possible routes. For instance if there are 4 cities, there are 6 possible routes, for 6 cities 120, for 10 cities 362880, and so on. As the number of cities increases, the amount of input data rises and the problem increases in complexity. The required computational time makes brute force impractical for all but a small number of cities.

We can immediately make a parallel between this and a DTN scenario, where the UAV is the salesman and the nodes

to be visited are the cities. Every time a UAV has to visit several nodes it needs to know which is the best route that optimizes message delivery. Thus, the UAV has to make use of algorithms to compute this route. For the computation to be quicker, it can use heuristics to approximate the exact solution. Although they can sometimes not be the best solution, they approximate the optimal result, and their execution time is significantly lower than classical approaches.

Matai et al. [14] present several solutions for the TSP problem. One common metaheuristic used to solve the TSP is the Genetic Algorithm (GA), which simulates the processes of biological evolution, natural selection and survival of the fittest in living organisms. In nature, individuals compete for the resources of the environment, and they also compete in selecting mates for reproduction. Individuals who are fitter in terms of their genetic traits survive to breed and produce offspring. The offspring carry their parents basic genetic material, alongside small mutations that introduce diversity in the new generation. Over many generations, this favorable genetic material evolves and propagates to an increasing number of individuals. The combination of good characteristics from different ancestors can sometimes produce super fit offspring which out-perform their parents. In this way, populations evolve to become better suited to their environment.

The GA is started with a set of solutions (usually denoted chromosomes), which form a population. Solutions from the original population are taken and used to form a new population. This is motivated by the hope that the new generation will be better than the old one in its characteristics. Solutions, which are designated to form new solutions (offspring), are selected according to their fitness attributes; the more suitable they are, the higher is their probability to replicate. This action is repeated until certain conditions are satisfied, e.g., until reaching a maximum number of iterations, or until the population fitness gradient is lower than a predefined threshold.

III. ANALYSIS AND IMPLEMENTATION OF EXISTING ALGORITHMS

From the literature review, we decided to implement and use SIRA and MRT-Grid as benchmarks to evaluate the proposed DTP-TSP-D in ferry scenarios. This section will present each of them in more detail and discuss the respective advantages and shortcomings. These protocols were implemented in MATLAB [15].

A. SIRA

Although this algorithm was developed over a decade ago by Zhao [3], it is still relevant as a basic message ferrying scheme, being the preferred comparison model for every new scheme in this area. As discussed before, SIRA is the algorithm in which all ferries follow the same route, being equally spaced. In SIRA there is no ferry assignment to ground nodes, which is the same as saying that all ground nodes are assigned to all ferries that share the responsibility of transporting messages between them.

To compute the optimized route that minimizes the traveled distance, a GA that solves the TSP problem was used [16]. First, the ground node positions are generated at random, the only restriction being ground nodes far enough so that there is no possible interaction, creating a realistic DTN scenario. Then, the genetic algorithm function operates on those ground node positions, retrieving the best route possible for all ferries to follow. After all UAVs are set to have the same constant speed, they start their movement and detect when they are near a ground node. When the UAV enters the transmission range, it first delivers the messages that have this ground node as destination and then collects as many messages as its buffer can handle at that time. After concluding this process, it continues in its route until reaching the next node, where it will repeat the process. This process goes on until the simulation time limit is reached. Fig. 2 depicts an example scenario with ferries going through a single ferry route.

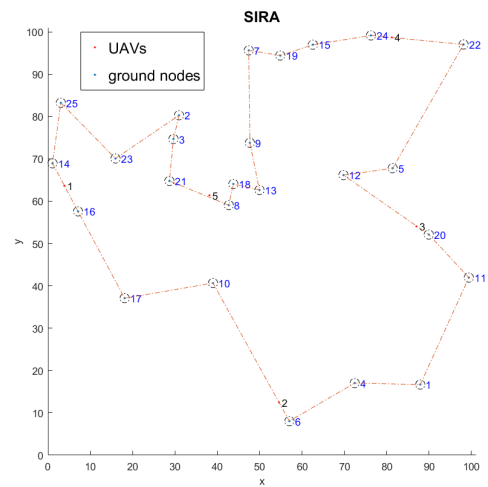


Figure 2. SIRA illustration.

A point in favor of this algorithm is the reduced computation power necessary in each UAV, since the route is only defined in the beginning. All the UAVs have to do is to exchange messages and to follow a pre-computed path. The big disadvantage that comes to mind looking at this algorithm is the constraint of following a single route. Several times, the destination node could be very close but, since the UAV has to strictly follow the route, the delay is much longer than hypothetically going straight to it. And, in case of having deadlines involved, the messages might not even get there in time. For example, imagine a message having ground node 10 as source, and ground node 8 as destination - see Fig. 2. This is where the idea of clustering emerges from, trying to deal with nodes in smaller clusters instead of one singular vast cluster.

B. MRT-Grid

To assign ground nodes to clusters, MRT-Grid uses a 2D-Tree partition algorithm, which acts by alternatively cutting horizontally and vertically in balanced parts, which is achieved by always cutting through the horizontal and vertical medians.

In order to make any ground nodes reachable from any other ground nodes, the ferry routes must be interconnected in some way. Two ferry routes are connected if they share one or more ground nodes. To choose the nodes that will act as relays, the first step is finding the closest pair of nodes between two adjacent partitions. Then, a TSP GA evaluates the two possible extensions of ferry routes, and chooses the one with minimum round-trip distance. Fig. 3 shows the resulting relay nodes and routes.

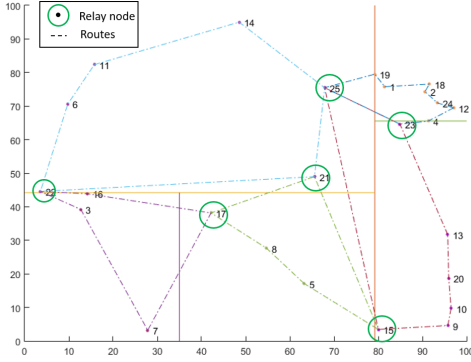


Figure 3. Relay nodes and ferry routes obtained for MRT-Grid.

Zhang et al. [6] do not specify the routing algorithm used in this scheme, explicitly assuming that ferries have routing knowledge beforehand. We decided to use directed graphs and shortest paths, using a source routing header, with the full ground node route sequence that the message should go through, appended to each message.

After having this pre-setting done, the execution can finally start. Each UAV will start its journey at one of the ground nodes in its assigned cluster. Having all the same constant speed, each UAV will follow its own ferry route. Whenever a UAV is near a ground node, it first delivers the messages intended for that destination (if it exists) and then proceeds to collect messages from it. In case of the ground node being a designated relay node, the UAV will transfer all the messages that have the next hop out of its own cluster, and collect all those that have been left there with next hop belonging to its cluster.

The bright side of this algorithm is the fact of using smaller clusters that allow UAVs to travel smaller distances, minimizing the transport delay, making easier the message transmission to nodes in the same cluster. However, when a network becomes too partitioned, the delay of relaying messages from clusters that are far apart increases, since they have to be relayed several times.

IV. DEADLINE TRIGGERED PIGEON WITH TSP DEADLINE (DTP-TSP-D)

After discussing all the advantages and shortcomings of each algorithm presented, we are now in position to present a scheme that uses those results to improve network performance. The model proposed here derives from both the MRT-Grid and the HoP DTN. In the new scheme, some new

configurations to help achieve better results were added, like a TSP genetic algorithm that takes deadlines into account, and smart UAVs that will not load messages unable to reach their destination in time. The proposed approach is flexible, meaning it can act like a pigeon or ferry depending on the ratio of UAVs available and the ground nodes in the deployed area.

First of all, a k-means algorithm [17] for clustering the ground nodes and assigning UAVs was used, creating as many clusters as there are UAVs available.

The UAVs act as ferries if their cluster has more than one ground node. If a cluster has more than three ground nodes, the TSP genetic algorithm presented earlier kicks in to obtain the best route i.e. the one that represents less traveled distance for the UAV to act as a ferry. Once the pre-setting is done, the simulation can finally start. Fig. 4 presents the decision diagram by which each UAV goes through during the simulation.

Ferry mode is the case when the UAV is going through ground nodes in its own cluster following a ferry route. When it enters pigeon mode, it acts as described in HoP DTN algorithm, with the improvement of collecting messages in each destination it goes through, addressed to nodes in its own cluster.

As the name of the algorithm suggests, instead of capacity triggering as in the original HoP DTN algorithm, we use deadline triggering to convert the ferry into a pigeon. Meaning, the algorithm takes into account the deadlines of all messages in the buffer to compute the best route to deliver them, and checks if it can still deliver all of them if it goes to the next ground on its ferry route. In the negative case, the UAV computes the route starting from its current position, and transforms into a pigeon to deliver the messages through the new computed route.

To achieve this triggering, we had to modify the genetic algorithm used to solve the TSP. The best route is no longer the one that achieves minimum distance but the one which maximizes the number of timely delivered messages, in the minimum travelling distance.

From the code in [16], the fitness function was modified to accomplish what was desired. Our simulation experiments have shown that the modified version achieves better results for the purpose intended. Even though it does not take the smallest route, it is able to make more timely deliveries.

The TSP GA Deadlines algorithm is used as a trigger. Whenever a UAV reaches a ground in its ferry route, it will collect messages, and right after, if it did not fill up its buffer, it will run the TSP GA Deadlines algorithm having the next ground position as the starting point and subtracting the time it takes to reach it. After finding how many timely deliveries it is able to achieve, if this number is less than the total number of messages in the buffer, then it will start pigeon mode and compute again the best route using TSP GA Deadline with its current position.

Summing up, the main concept behind our model is to use the deadlines as a constraint to start the pigeon mode delivery

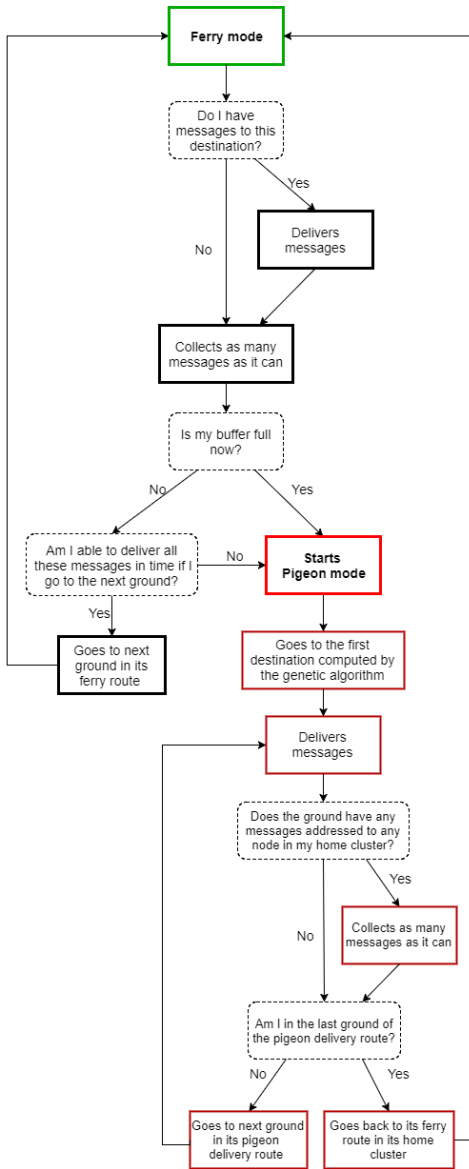


Figure 4. UAVs decision diagram.

when the UAV is working initially as a ferry. Compared to ferry models, the difference lies in the clustering, and in reaching nodes out of the cluster. Compared to the pigeon model, the main difference is this deadline constraint that makes the UAV start the delivery earlier than having its buffer full. In both scenarios, we seek to maximize the number of deliveries, using the modified genetic algorithm. Comparative performance evaluation is done in Section V.

V. PERFORMANCE EVALUATION

All the models were evaluated using simulations results obtained with MATLAB.

A. Simulation Settings

In our simulations, 25 nodes are randomly distributed in the deployment area. Ground nodes are stationary, and no direct

communications are feasible between any pair of grounds. All messages have to be delivered by the UAVs. The generated messages are buffered in a First Come First Serve (FCFS) queue in the ground node, waiting for a UAV to enter its transmission range. The following assumptions were made:

- 1) The messages are generated at each node according to a Poisson process with a mean rate λ .
- 2) Each message picks one of the other ground nodes as its destination with an equal probability.
- 3) The buffer capacity of a ground node is unlimited.
- 4) All messages have the same expiration time (deadline).
- 5) The UAVs move around at a constant speed.
- 6) The ground node positions are known beforehand.

Table I lists the simulation parameters and their value.

The performance evaluation considers two of the most common performance metrics in protocol evaluation: Delivery Ratio and Average Delay. For each set of parameter values, 10 different ground node deployment locations were tested, each run three times with different seeds. The 95% confidence intervals were calculated, which are included in the graphics. To further analyze the behavior of the algorithms, the following independent variables were used: buffer size (UAV Capacity), deadline and message generation rate. Several setups were used, having the buffer size vary from 10 up to 150 messages, with increments of 10 messages, and the deadlines and generation rate (λ) are combined into nine different scenarios. Table II will be helpful to understand the result figures presented later.

When comparing DTP-TSP-D with other ferry models, we varied the UAV to ground ratio, using 25 ground nodes together with 5 or 15 UAVs.

Fig. 5 presents the results for 5 UAVs in every scenario presented in Table II.

Overall, our DTP-TSP-D obtains better results in terms of delivery ratio, with the corresponding curve being always higher than the others with the improvement ranging between 2.6 to 163.6 times better.

With lower deadlines, [Scenario (a), (b) and (c)] the delivery ratio is practically constant with the increasing UAV capacity.

Table I
SIMULATION PARAMETERS

Parameters	Values
Simulation duration	1000 units of time
Total number of ground nodes	25
Deployment area	100 x 100
UAV moving speed	1 unit of distance per unit of time

Table II
COMBINATIONS OF DEADLINE AND GENERATION RATE CORRESPONDING TO EACH SIMULATION SCENARIO.

		Generation Rate (λ)		
		0.5	1	1.5
Deadline	50	(a)	(b)	(c)
	100	(d)	(e)	(f)
	150	(g)	(h)	(i)

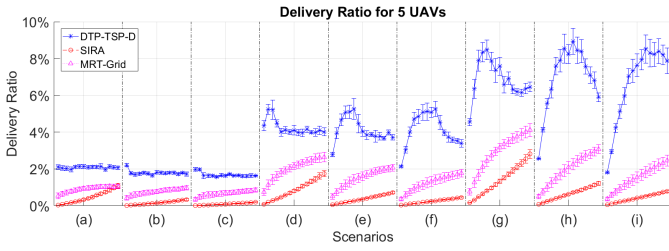


Figure 5. Delivery Ratio Results for 5 UAVs as a function of UAV buffer capacity.

However, for higher deadlines, we can see a different behavior, with the delivery ratio growing initially with the buffer capacity, reaching a peak and then decreasing a little. This behavior can be explained by a first phase where the lack of space in the buffer has a greater influence, having the pigeon mode triggered more frequently due to reaching full capacity. The highest value must correspond to an equilibrium between capacity triggered and deadline triggered. From that point on, the delivery ratio slightly decreases, meaning that the lack of capacity triggering results in lower delivery ratios. This is an expected result, since the UAV starts less trips in pigeon mode, thus not delivering so many messages in its way, and not bringing back messages to its cluster.

The behavior of SIRA and MRT-Grid is similar in all scenarios against the increase of buffer size, with the delivery ratio increasing as the buffer capacity increases. We can also observe that these algorithms are better suited for small generation rate, having a worse delivery ratio when the generation rate increases, meaning that they are unable to drain all the traffic in the network, as the number of generated messages per unit of time increases. Overall, MRT-Grid is better than SIRA, proving that small clusters instead of one single route improve the delivery ratio, and thus the network performance. With only one route, for small deadlines, SIRA's valid transmissions are those made for nearby grounds next on the route, which are the only ones able to comply with deadlines. On the other hand, MRT-Grid intra-cluster transmissions are easily reachable, which can explain the difference obtained in delivery ratio. Fig. 6 presents the average delay for each model in the nine scenarios listed in Table 2, with 5 UAVs.

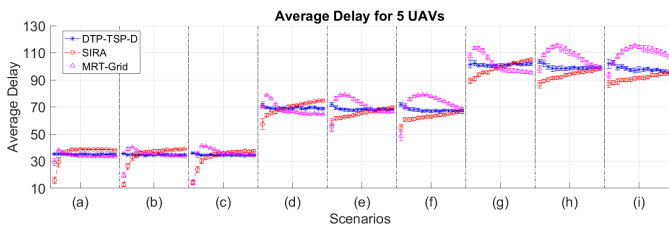


Figure 6. Average Delay Results for 5 UAVs, as a function of UAV buffer capacity.

Here, the behavior is quite diverse. There is no algorithm that is always better than the others. However, the average delay achieved by DTP-TSP-D is more constant than its

competitors. Overall, SIRA is able to achieve smaller average delay, which was predictable due to the fact that its valid transmissions are only those with destinations close enough and next in the route. Thus, the deadlines impact both delivery ratio and average delay. If there was no time limit, SIRA would be able to deliver more messages, but would also attain worse average delay. On the other hand, MRT-Grid achieves overall greater average delays than the others, which may result from the amount of time lost at the relaying nodes, waiting for a UAV to take the messages to the next hop. Since DTP-TSP-D is deadline triggered, it was expected that the average delays would be closer to the deadlines. However, since every time the UAV is in pigeon mode it retrieves messages to its home cluster, those messages are delivered with smaller delay than those taken from the home cluster to the outside. Therefore, the average delay is quite smaller than the deadline.

Fig. 7 presents the Delivery Ratio results for 15 UAVs.

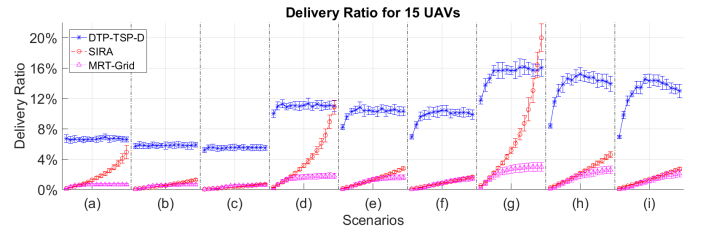


Figure 7. Delivery Ratio Results for 15 UAVs, as a function of UAV buffer capacity.

DTP-TSP-D continues to perform better than the others in terms of delivery ratio. Although it appears that, with a higher number of UAVs, SIRA achieves comparable performance for a small generation rate [Scenarios (a), (d), (g)]. looking at Fig. 5 and 7, we can see that DTP-TSP-D improves in terms of delivery ratio when more UAVs are employed. The behavior in each scenario mostly remains the same, achieving on average 2.12 times more timely deliveries. SIRA also improves when more UAVs are available, which is due to having ground nodes visited more often. On the contrary, more UAVs does not affect MRT-Grid in a positive way. The delivery ratio obtained with more UAVs is frequently smaller than the one obtained with fewer UAVs. The reason behind this result is the fact that the network with 15 UAVs becomes too partitioned, having messages from ground nodes that are far apart, going through a high number of relay nodes to get to their destination.

Fig. 8 presents the Delivery Ratio results for 15 UAVs.

As seen for 5 UAVs, DTP-TSP-D average delay seem to remain essentially constant in each scenario, and not be affected by the generation rate. However, its average delay is frequently the highest of the three algorithms. Contrary to what was observed with 5 UAVs, MRT-Grid with 15 UAVs often achieves lower average delay than its competitors.

Looking at Fig. 6 and 8 we can conclude that:

- DTP-TSP-D average delay is not significantly affected by using more UAVs in the network.
- More UAVs seem to affect SIRA in a positive way, since globally it achieves lower average delay with 15 UAVs

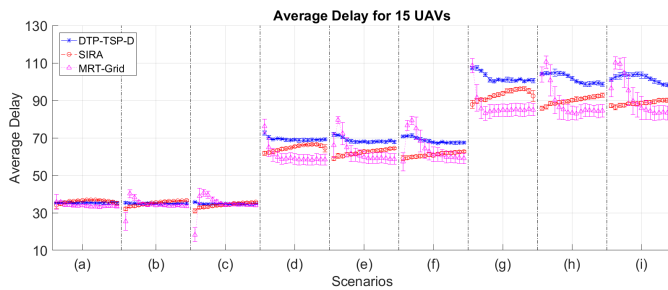


Figure 8. Average Delay Results for 15 UAVs, as a function of UAV buffer capacity.

than with 5 UAVs, which is due to UAVs passing more frequently, meaning the messages will wait less time in the source than with 5 UAVs.

- MRT-Grid also presents lower average delay with the use of more UAVs. However, these results do not mean that having more UAVs is better for this algorithm, since the reason the average delay decreases is the fact that the range of valid transmissions also decreases (the number of messages arriving within bounds are only those intended for close ground nodes that need fewer hops).

VI. CONCLUSIONS AND FUTURE WORK

This paper proposes a new dedicated node-based protocol, assuming a DTN where ground nodes are far apart and the only mean of communication is by flying UAVs from one location to another. In the Deadline Triggered Pigeon with Travelling Salesman Problem with Deadlines (DTP-TSP-D), when there are fewer UAVs available than ground nodes in the network, the UAVs may act as ferries, each being assigned to a cluster of ground nodes. The UAV will hover over the ground nodes in its cluster by following a ferry route obtained beforehand by a basic TSP GA. The UAV will follow this route, until it is triggered to become a pigeon, whereupon it delivers the messages targeted to ground nodes outside its cluster.

Simulations scenarios were created to evaluate the performance of DTP-TSP-D against the SIRA and MRT-Grid ferry schemes. The results demonstrate that the DTP-TSP-D outperforms the other schemes in terms of delivery ratio, although it does not always present the best average delay. Besides, by using different UAV to ground ratios, first 5 UAVs to 25 ground nodes and later 15 UAVs to 25 ground nodes, we could observe that more UAVs means more deliveries, with the delivery ratio increasing on average approximately twice with the use of 15 UAVs versus using 5 UAVs. We also realized that adding more UAVs does not significantly affect the average delay.

Future work includes making more realistic simulation taking into account processing delay, physical transmission delay and power consumption. Another interesting study would be applying this algorithm when partitioned clusters are mobile and dynamic, which may entail the development of ground node trajectory learning algorithms and collaboration between

UAVs (e.g., by establishing intersecting routes or meeting points).

VII. ACKNOWLEDGEMENTS

This work was supported by national funds through Fundao para a Ciênci e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

REFERENCES

- [1] Y. Zeng et al. "Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges", in *IEEE Communications Magazine*, vol.54, n.5, pp. 3-42, May 2016.
- [2] K. Fall, "A delay-tolerant network architecture for challenged internets" in *Proceedings of the 2003 Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03)*. ACM, NY, USA, pp.27-34.
- [3] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks", in *Proceedings of the 5th ACM MobiHoc*, Tokyo, Japan, May 2004, pp. 187-198.
- [4] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks", in *Proceedings of IEEE INFOCOM*, pp. 398-408, FL, USA, March 2005.
- [5] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network", in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, FL, USA, March 2005.
- [6] Z. Zhang, Z. Fei, "Route Design for Multiple Ferries in Delay Tolerant Networks" in *Wireless Communications and Networking Conference (WCNC)*, Kowloon, China, March 2007.
- [7] L. Xue, J. Liu and J. Peng, "An Adaptive Message Ferry Routing Algorithm for Delay Tolerant Networks", in *ICCT- International Conference in Communication Technology*, Chengdu, China, November 2013.
- [8] H. Guo, J. Li and Y. Qian, "HoP-DTN: Modeling and Evaluation of Homing-Pigeon-Based Delay Tolerant Networks" in *IEEE Transactions on Vehicular Technology*, vol.59, n.2, pp. 857-868, February 2010
- [9] H. Guo et al., "A practical routing strategy in Delay Tolerant Networks using multiple pigeons", in *MILCOM*, San Diego, CA, USA, November 2008.
- [10] A. Vahdat, and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks", *Technical Report CS-200006*, Duke University, NC, April 2000.
- [11] T. Spyropoulos et al., "Spray and wait: an efficient routing scheme for intermittently connected mobile networks", in *Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networks (WDTN)*, Philadelphia, USA, August 2005.
- [12] A. Lindgren, A. Doria, O. Schelen, "Probabilistic routing in intermittently connected networks", in *ACM SIGMOBILE Mobile Computing and Communications Review*, vol.7, no.3, pp. 19-20, New York, USA, July 2003.
- [13] Sobin CC, V. Raychoudhury, G.Marfia, A. Singla, "A survey of routing and data dissemination in Delay Tolerant Networks", in *JNCA*, V.67, Supplement C, pp. 128 - 146, London, UK, May 2016.
- [14] Rajesh Matai, Surya Singh and Murari Lal Mittal, "Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches, Traveling Salesman Problem, Theory and Applications", Prof. Donald Davendra (Ed.), InTech, December 2010.
- [15] MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.
- [16] J. Kirk, Traveling Salesman Problem - Genetic Algorithm File Exchange MATLAB Central. [online] Available at: <https://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm> [Accessed 29 August 2017].
- [17] Mathworks.com. (2017). k-means clustering - MATLAB kmeans - MathWorks United Kingdom. [online] Available at: <https://www.mathworks.com/help/stats/kmeans.html> [Accessed 2 Sep. 2017].