GPSR-PPU: Greedy Perimeter Stateless Routing with Position Prediction and Uncertainty for FANETs

A. Rodrigues, A. B. Reis, S. Sargento
Instituto de Telecomunicações, Universidade de Aveiro, Aveiro, Portugal
Email: {andrenascimento, abreis, susana}@ua.pt

***Abstract*—Existing routing algorithms for ad-hoc networks are not tailored to the characteristics of Flying Ad-hoc Networks (FANETs), such as the integration with mission data, predictability and uncertainty. This paper proposes an algorithm designed specifically for FANETs, which works by adding position prediction and uncertainty to the well-known GPSR algorithm. This new algorithm improves the selection of the next-hop node in the highly-mobile and noisy FANET environments. A thorough set of simulations shows 33% improvements in packet loss, 12% in overhead, and 42% less jitter. Through an integration with mission planning software, empirical evaluations in a real-life firefighting scenario validate its behaviour and performance improvements.**

***Index Terms*—Flying Ad-hoc Networks, Position Prediction, Geographic Routing, Firefighting**

I. Introduction

Commonly known as aerial drones, Unmanned Aerial Vehicles (UAVs) have been through significant technological advancements in the last decade that contributed to their use in new areas: gather and analyze data for scientific purposes such as farming yield prediction [1], radiological mapping and monitoring of scenarios with nuclear threats [2], and building inspection through 3D image reconstruction.

On scientific areas, the advancements on drone mission planning platforms show that drones can co-operate and complete missions together; thus, the idea of assigning multiple missions to multiple drones to achieve organized swarms of drones make possible that these flying units interact with each other directly.

Swarms of flying drones can form wireless mesh networks between the drones, which, through multi-hop communications, can be made highly resilient, provide widespread coverage, and enable drones to talk with one another without any infrastructure in place. Multiple paths between any two drones may exist at once, and due to the drones' high mobility and terrain obstructions, those paths may change frequently.

In this work, we propose a routing protocol that is able to deal with FANETs, considering the drones' locations, their missions and unpredictability on the missions, such as obstacle avoidance, and change on route due to a sensed parameter. This approach, based on a location-based protocol, manages the FANET, reacting to the drones' positioning by changing routing paths according to the network's topology along space and time. Moreover, as previously referred, during their trajectories, drones may face some obstacles that forces them to cancel their assigned trajectory and bypass the obstacle by defining an alternative set of trajectories which lead to a new defined trajectory or destination.

The proposed approach has been evaluated in a simulation environment and in a real platform, by integrating it with a mission planner in a firefighting scenario. The obtained results show that position prediction with uncertainty increased Packet Delivery Ratio (PDR) by 33% on normal conditions where each UAV on the network reaches 100% of its assigned trajectory, and, on exceptional conditions, where UAVs reach at least 80% of the assigned trajectory, increased PDR by 15%.

This paper is organized as follows. Section II presents the related work about routing protocols in ad-hoc networks. Section III describes Greedy Perimeter Stateless Routing (GPSR) forwarding techniques. Section IV proposes the routing approach with position prediction and uncertainty. Section V presents the simulation results, while section VI describes the integration of the routing platform with an autonomous mission planner platform on a real-life experiment. Finally, concluding remarks are given in Section VII.

II. Related Work

GPSR [3] showed to be a prominent position-based algorithm on Mobile Ad Hoc Networks (MANETs) due to not picking, blindly, a path to a destination; thus, by having the knowledge of the neighbours' positions, this routing algorithm promises to better deal with more volatile environments where the real-life distance matters more than the latency. On [4], the selection methodology that opts for evaluating the neighbours through the *right hand rule* is augmented to include the *left hand rule*. This GPSR version is often referred to as Minimum Angle or Maxduration-Minangle Greedy Perimeter Stateless Routing (MM-GPSR).

GPSR can increase its performance by maximizing the amount of times the algorithm chooses for greedy mode, rather than opting for perimeter mode, when it has to proceed to an evaluation of the neighbour through *right/left-hand rule*. Authors on [5] proposed Path Aware Greedy Perimeter Stateless Routing (PA-GPSR), a GPSR version which includes a deny table to avoid routes that are not appropriate to a particular destination, and a recently sent table to handle and control the packet forwarding. On [6], authors propose another modification named GPSR with Lifetime (GPSR-L) which introduces the concept of lifetime which is calculated between a specific node and its surrounding neighbours; thus, a lifetime variable helps in determining the quality of neighbours

communication and with it GPSR-L improves PDR from 20% to 40%.

The work on [7] provides a performance comparison between GPSR and Zone Routing Protocol (ZRP) on Vehicular Ad Hoc Networks (VANETs). Authors conclude that ZRP has better performance when sending data packets in the sparse area but with low throughput, revealing to be sensitive to the number of nodes on the network. GPSR, although less reliable to transmit data packets, has the advantage with high throughput and low delay. Finally, [8] presents a GPSR enhancement that uses node movement information to increase the network's performance by estimating a node's future position.

III. GPSR Forwarding Techniques

GPSR follows a greedy methodology, opting for the closest nearby neighbour to the destination to forward the data (*greedy mode*); otherwise, it chooses the first one by the *right hand rule* (*perimeter mode*). This section describes these forwarding techniques.

A. Greedy Mode

The focus of GPSR is to maximize the number of times the algorithm chooses to data-forward through greedy mode instead of perimeter mode, due to its greedy methodology, illustrated in Figure 1. The algorithm forwards in *greedy mode* when there is a neighbour closer to the destination than itself.

B. Recovery/Perimeter Mode

When there is no other node closer to the destination than itself, and there is no direct connection from itself to the destination, then it is not possible to forward data in *greedy mode*. In this case, the algorithm proceeds to data-forward through *perimeter mode* in case there are neighbors around; otherwise, the data packets are discarded, meaning that the node is alone, as depicted in Figure 1. When entering the *perimeter mode*, an initial *void zone* is defined between the source and the destination, and the algorithm will keep forwarding in *perimeter mode* until it forwards to a node that is within the initial defined *void zone*; then, the *greedy mode* can again be chosen.

IV. Position Prediction

GPSR routing is developed based on a node's geographic location; thus, a position and a velocity vector are attached to every node. These vectors reveal the node's trajectory, and the time its journey takes to be accomplished, referred to as *time travel delay*.

Considering that each drone's flight pattern is linear and does not change during t, it is possible to predict the position of the drone. It is done by adding ($t\times$ velocity vector) to each axis of the position vector.

This version is built over MM-GPSR which, when forwarding in *recovery/perimeter mode*, evaluates both *left* and *right hand rule*, choosing one of them instead of duplicating data packets. It takes advantage of position prediction in both *greedy*

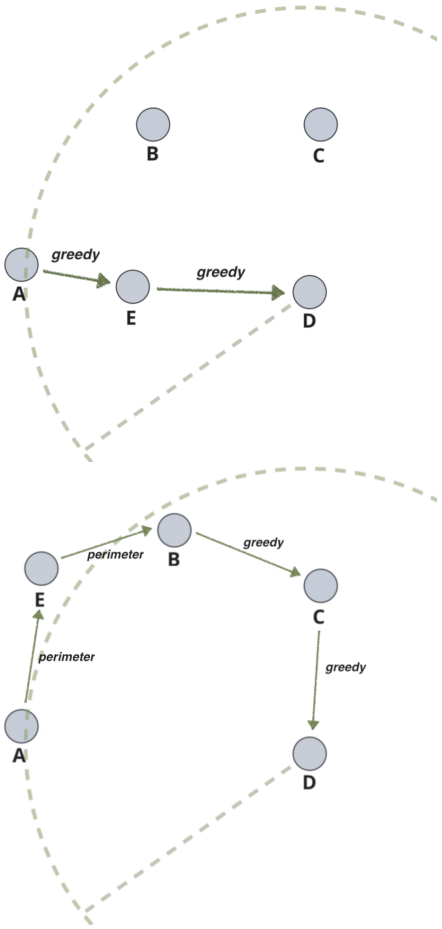

Figure 1. GPSR data-forwarding through *greedy mode* (top) and *perimeter mode* (bottom) between nodes A and D.

and *recovery/perimeter mode*, choosing what appears to be the best next-hop when evaluating a predicted scenario.

Mobile nodes on FANETs move quickly, and their movement is not constant due to real-life factors that may influence and change the initial traced route. For example, when a route is initially defined the node, during its travel, may have to change its trajectory due to the existence of trees, buildings, or other obstructions that make it impossible to follow its initial route. The node then has to define an alternate route to reach the initially-defined destination. In multi-drone missions, the contact time when two nodes coexist is short, and this led to the analysis of how much time it can be predicted that a node is in contact with another one. The distance between a node A to node N with positions P_A and P_N, is given by Equation 1, and the relative speed between two nodes is given by Equation 2.

$$\mathrm{D_{A,N}} = \sqrt{(\mathrm{P_N} - \mathrm{P_A})_x^2 + (\mathrm{P_N} - \mathrm{P_A})_y^2} \tag{1}$$

$$\mathrm{Vrel_{A,N}} = \begin{cases} \sqrt{(\mathrm{V_A} - \mathrm{V_N})_x^2 + (\mathrm{V_A} - \mathrm{V_N})_y^2} & \text{same dir.} \\ \sqrt{(\mathrm{V_A} + \mathrm{V_N})_x^2 + (\mathrm{V_A} + \mathrm{V_N})_y^2} & \text{otherwise} \end{cases} \tag{2}$$

Equation 3 determines the amount of time that node A distances, in seconds, to a node N, named as *Lifetime*. The *Lifetime* variable purpose is inspired in [6], where a similar ideology is followed to opt for a better forwarding in high mobility environments.

$$\text{Lifetime}_{\text{A,N}} = \frac{\text{D}_{\text{A,N}}}{\text{Vrel}_{\text{A,N}}} \tag{3}$$

The maximum useful time that is used to predict a neighbour N position is equal to the maximum time that this neighbour N is on that given trajectory, which is its *time travel delay*. However, the time travel delay cannot be used directly as a prediction time for each individual node, because then it would not matter how fast does a node reach a point, only the final reached position would be used for the distance comparison with the destination node, which is misleading.

We can, therefore, infer that the prediction factor, used to predict the position of a neighbour N, is given by the *lifetime* between a source node and the neighbour node per *time travel delay (tr)* of the neighbour node times the neighbour's uncertainty percentage (u), shown in Equation 4, which is the fraction of time both nodes can keep a communication while being on the neighbour's trajectory.

$$\text{PredFactor}_{\text{N}} = \frac{\text{Lifetime}_{\text{A,N}}}{\text{tr}_N \times \text{u}_N} \tag{4}$$

When the uncertainty value is 100%, this means that the current node will accomplish 100% of its assigned trajectory. If it is measured as 50%, it is expected that the node will accomplish only 50% of its traced trajectory, on average. This value is obtained by averaging the success rates of previous trajectories.

V. Experimental Results

In this section we compare the performance of GPSR-PPU with the GPSR base version and MM-GPSR. The experiment is carried out in NS-3.25 simulator under Ubuntu Server 16.04.06. In a simulation, the throughput, packet delivery ratio, communication latency, jitter and routing overhead are used as metrics for the analysis of the results. The simulation parameters are depicted in Table I where 50 nodes are placed in a rectangular area that can be 150 to 900 meters wide.

There are two simulation scenarios: in the first one, nodes achieve 100% of its assigned trajectory; in the second scenario, nodes achieve at least 80% of the assigned trajectory, to evaluate the impact of the *uncertainty* regression variable. In this case, the remaining 20% of the trajectory are decided by a random variable.

1) Complete Trajectory: The Packet Delivery Ratio (PDR) of GPSR, MM-GPSR and GPSR with position prediction is shown in Figure 2. The PDR of GPSR-PPU sees an absolute increase of up to +15.46%, a relative improvement of **33%** when compared to base GPSR, since the mobility prediction provides more information on the destination and the path to the destination. The results of jitter (in Table III) and routing overhead (in Table IV) with mobility prediction show absolute improvements of 1.23 seconds and 8.33%, respectively, which are relative improvements of **42%** and **12%** against base GPSR. Again, the prediction of mobility through the information on the missions of the network of drones is able to improve the stability of the network, while decreasing the control packets in the routing approach.

2) Incomplete Trajectory: We now consider the second scenario, where nodes achieve at least 80% of its assigned trajectory in their mission. PDR and delay results are presented in Figures 3 and 4. Relevant results are highlighted in Table V, VI, and VII. From the obtained results, the impact of the uncertainty regression variable is evaluated. We can see that,

Table I
Simulation Parameters

Nodes	50
Packet Size	1024 bytes
Max. Packet Count	200
Simulation Time	20 seconds
Total of Simulations Run	150
Mobility Model	RandomWaypointMobilityModel
Speed	RandomVariable[5.0-10.0]
Position Allocator	RandomRectanglePositionAllocator
Rectangle Width & Weight	150, 300, 450, 600, 750 and 900
Propagation Delay Model	ConstantSpeedPropagationDelay
Propagation Loss Model	RangePropagationLossModel
Data Type	CBR
Transport Protocol	UDP
Channel Data Rate	6 Mbps OFDM

Table II
PDR (%)
GPSR with position prediction vs base GPSR

	150	**300**	**450**	**600**	**750**	**900**
GPSR	97.12	80.99	74.68	63.47	47.32	34.4
GPSR-PPU	98.26	87.75	83.21	76.63	62.75	43.25
Δ_{abs}(%)	↑ 1.14	↑ 6.76	↑ 8.53	↑ 13.16	↑ 15.46	↑ 8.9
Δ_{rel}(%)	↑ 1.2	↑ 8.3	↑ 11.4	↑ 20.7	↑ **32.6**	↑ 25.7

Table III
Jitter (sec)
GPSR with position prediction vs base GPSR

	150	**300**	**450**	**600**	**750**	**900**
GPSR	0.19	1.72	2.63	2.90	3.44	3.94
GPSR-PPU	0.17	1	1.6	1.97	2.31	2.71
Δ_{abs}(s)	↓ 0.02	↓ 0.72	↓ 1.03	↓ 0.93	↓ 1.13	↓ 1.23
Δ_{rel}(%)	↓ 10.5	↓ **41.9**	↓ 39.2	↓ 32.1	↓ 32.8	↓ 31.2

Table IV
Routing overhead results (%)
GPSR with position prediction vs base GPSR

	150	**300**	**450**	**600**	**750**	**900**
GPSR	38.83	43.36	51.46	63.01	73.91	86.26
GPSR-PPU	35	41.7	45.22	55.62	65.58	84.08
Δ_{abs}(%)	↓ 3.83	↓ 1.66	↓ 6.24	↓ 7.39	↓ 8.33	↓ 2.18
Δ_{rel}(%)	↓ 9.9	↓ 3.8	↓ **12.1**	↓ 11.7	↓ 11.3	↓ 2.5

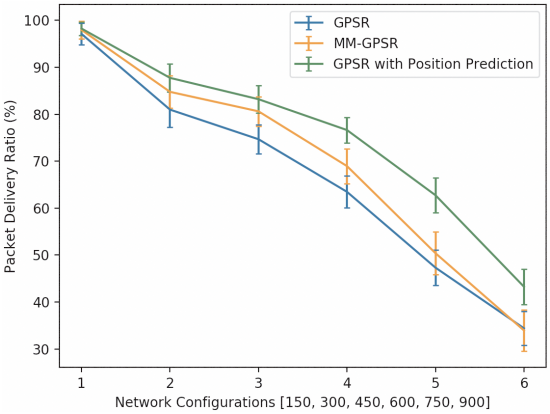

Figure 2. Packet Delivery Ratio when drones always achieve 100% of its assigned trajectories.

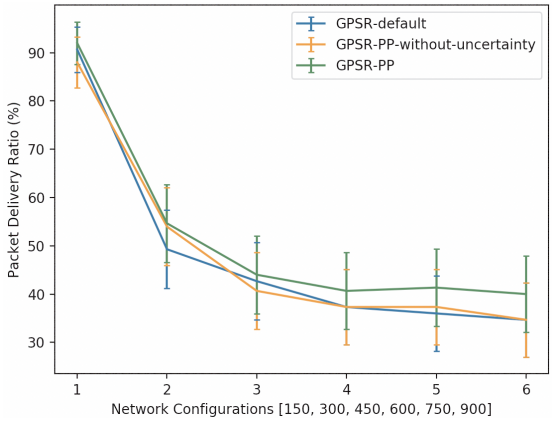

Figure 3. PDR when drones achieve at least 80% of its assigned trajectories.

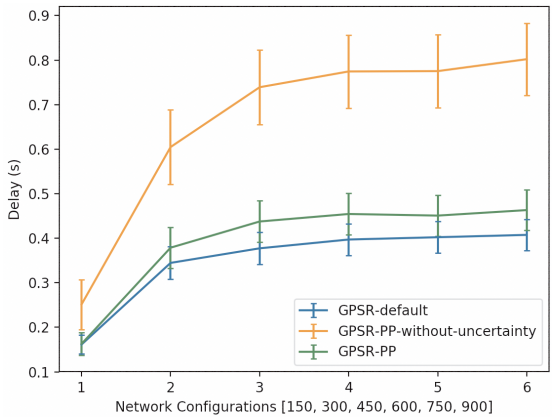

Figure 4. End-to-End Delay when drones achieve at least 80% of its assigned trajectories.

introducing uncertainty to the GPSR-PPU, helps to reduce the average delay by 0.34s (absolute value), increasing the average PDR by 5.35% (absolute value), achieving +73% and +15% (relative value) respectively. This approach learns the previous trajectories and how far they are from the previous assigned missions, being able to understand when the current mission is changed and updating the routing accordingly.

VI. Firefighting Integrated Experiment

The routing platform has been developed in a real environment and has been integrated with a mission planner that assigns missions to multiple UAVs. The integration of both platforms resulted in a real-life experiment, providing drone-to-drone communication through a wireless mesh.

The scenario developed is the one to support firefighters in the mission to extinguish a fire. In this scenario, UAVs distribute themselves to provide the maximum wireless coverage to the firefighters, which are moving near the fire hose. UAVs gather sensing data from the firefighters sensors and position, and temperature and other gas data information from the air near the firefighters, and follow firefighters on duty.

Firefighter movement, temperature, and the fire location are simulated by a developed Firefighter Sensors Simulator module and Fire Simulator module, respectively, which was built to validate its functionalities before testing on a real fire.

In the initial experiment phase, UAVs on the multi-hop stage are distanced from each other, resorting to multi-hop to communicate; then, in this case, drone 2 acts as a bridge for communications between drone 1 and drone 3, as previously described. During the time, UAVs tend to become closer to each other, so multi-hop is not needed anymore. This scenario and the drones' positions in the several phases are depicted in Figure 5.

The evolution of this experiment over time is illustrated in

Table V
PDR (%), GPSR-PPU without and with uncertainty, compared with base GPSR

	150	**300**	**450**	**600**	**750**	**900**
W/o	-2.65	+4.69	-2.01	+0.01	+1.35	+0.01
With	+1.33	+5.37	+1.33	+3.35	+5.35	+5.35
Δ_{abs}(%)	↑ 3.98	↑ 0.68	↑ 3.34	↑ 3.34	↑ 4	↑ **5.34**

Table VI
Delay (sec), GPSR-PPU without and with uncertainty, related with base GPSR

	150	**300**	**450**	**600**	**750**	**900**
W/o	+0.09	+0.26	+0.36	+0.38	+0.37	+0.40
With	0	+0.03	+0.06	+0.06	+0.05	+0.06
Δ_{abs}(s)	↓ 0.09	↓ 0.23	↓ 0.30	↓ 0.32	↓ 0.32	↓ **0.34**

Table VII
Jitter (sec), GPSR-PPU without and with uncertainty, related with base GPSR

	150	**300**	**450**	**600**	**750**	**900**
W/o	+0.008	+0.002	+0.003	+0.004	+0.003	+0.004
With	+0.003	+0.020	+0.240	+0.026	+0.024	+0.025
Δ_{abs}(s)	↓ 0.005	↓ 0.018	↓ **0.237**	↓ 0.022	↓ 0.021	↓ 0.021

Figure 5. Multi-hop scenario (left) and non-multi-hop scenario (right).

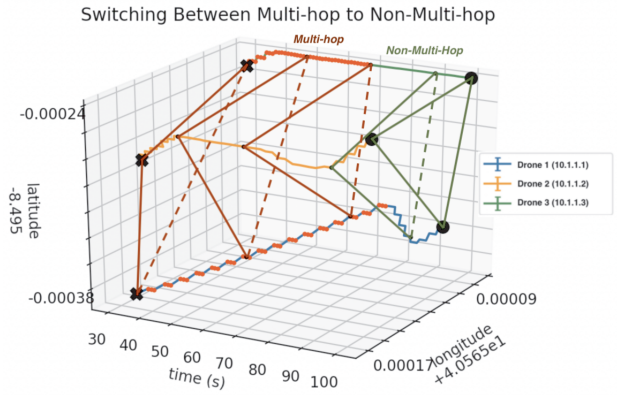

Figure 6. Drones trajectories between Multi-Hop zone and Non-Multi-Hop zone, marking active routes as red on traced paths.

Figure 6 and Figure 8. On the first scenario, it can be seen that, initially, the triangle made by UAVs is larger; according to UAVs movements, the triangle is becoming smaller, meaning that nodes are closer to each other. On the second scenario, the opposite happens. In both scenarios it is possible to identify different time zones where UAVs adopted multi-hop or direct connections between them. The FANET throughput and its transition between the multi-hop zone to the non-multi-hop zone is depicted in Figure 7, and the opposite, between the non-multi-hop zone to the multi-hop zone, is illustrated in Figure 9.

A. Multi-hop Zone

On the multi-hop zone, UAVs moved to follow the firefighters as expected. The triangle formation got wider, covering both teams and the ground station, Drone 1 loses the direct connection with Drone 3, resorting to multi-hop via Drone 2.

Drone 1 is directly connected to Drone 2 and establishes a route to Drone 3 through Drone 2. Drone 2 acts as a bridge, assisting the multi-hop between Drone 1 and Drone 3. Drone 3 establishes a route to Drone 1 through Drone 2, which is its only direct neighbour and the closest to Drone 1. While in multi-hop, the throughput measured by drones 1 and 2 has slight variations, on the transition phase it contains some peaks and decreases at drone 1, since this is the one transiting from multi-hop to single-hop. An existing RSSI monitor process, which analyzes the RSSI levels of direct connections, is registering and smoothing the RSSI levels

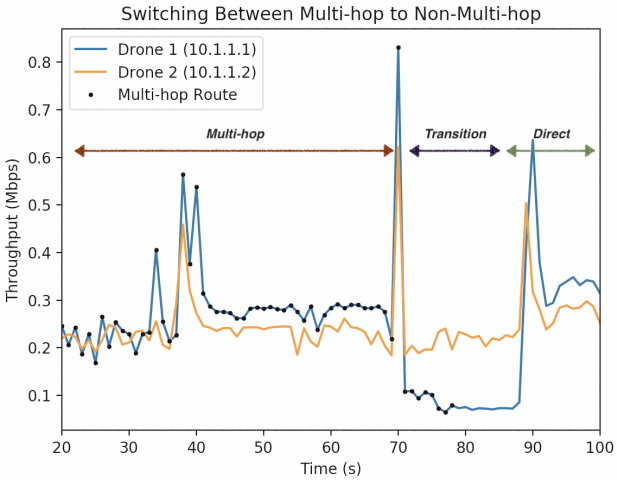

Figure 7. Throughput behavior while nodes switch between Multi-hop and Non-Multi-hop.

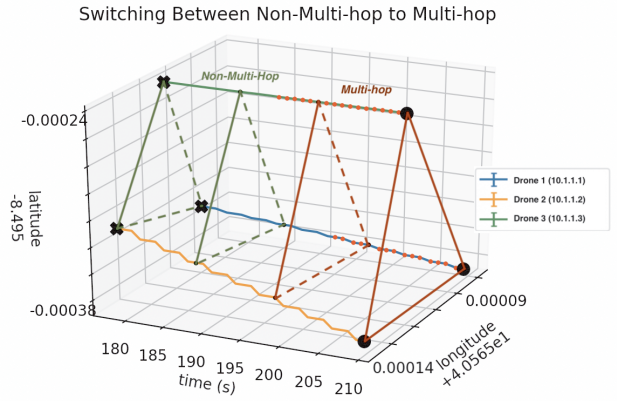

Figure 8. Drones trajectories between Non-Multi-Hop zone and Multi-Hop zone, marking active routes as red on traced paths.

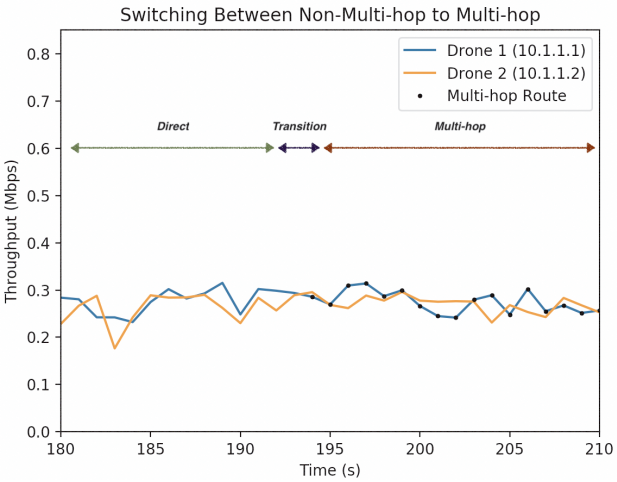

Figure 9. Throughput behavior while nodes switch between Non-Multi-hop to Multi-hop.

of a possible direct connection from where the multi-hop connection may transition to.

B. Non-Multi-hop Zone

On the non-multi-hop zone, UAVs do not need to multi-hop to reach any nearby UAV; thus, there are no established multi-hop routes, each UAV has the knowledge of all existing neighbours on the network. While in the non-multi-hop zone, the throughput measured by the drones reveals to be stable during the single-hop stage. Throughout the transitioning to multi-hop and afterwards, the throughput shows an unremarkable behavior due to the actions of the RSSI monitor, which triggers a transition to a multi-hop route when the signal strength becomes weak.

VII. Conclusion

Improvements on the GPSR routing algorithm led to the development of position prediction, where a prediction time value estimates the future location of a node. A routing platform was developed to interconnect all the existing drones on a network and provide routing paths to each other. For node discoverability, we developed a three-way handshake service that establishes neighbour relationships and synchronizes their neighbor entries with the latest data.

The proposed algorithm improves the PDR by 33%, lowers overhead by 12% and mean jitter by 42% when compared to baseline GPSR (relative values). In scenarios with unpredictability where UAVs reach at least 80% of their expected trajectory, the algorithm improves PDR by 15% and lowers the delay by 73% when compared to GPSR-PP without uncertainty (relative values). The real-life experiments showed that the improved GPSR algorithm with position prediction is successful at ensuring routing capabilities on a FANET.

Acknowledgments

This work is funded by FCT/MCTES through national funds under the projects FRIENDS (PTDC/EEI-ROB/28799/2017), S2MovingCity (CMUP-ERI/TIC/0010/2014), and when applicable co-funded EU funds under the project UIDB/50008/2020-UIDP/50008/2020, as well as by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Lisbon (POR LISBOA 2020) and the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020 framework under Project 5G (POCI-01-0247-FEDER-024539).

References

[1] X. Zhou *et al.*, "Predicting grain yield in rice using multi-temporal vegetation indices from uav-based multispectral and digital imagery", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130, pp. 246–255, 2017.

[2] *Friends: Fleet of drones for radiological inspection, communication and rescue*. [Online]. Available: https://www.ipfn.tecnico.ulisboa.pt/FRIENDS/ (visited on 08/10/2019).

[3] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks", in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, 2000, pp. 243–254.

[4] L. Lai, Q. Wang, and Q. Wang, "Research on one kind of improved gpsr algorithm", in *2012 International Conference on Computer Science and Electronics Engineering*, IEEE, vol. 3, 2012, pp. 715–718.

[5] A. Silva, N. Reza, and A. Oliveira, "Improvement and performance evaluation of gpsr-based routing techniques for vehicular ad hoc networks", *IEEE Access*, vol. 7, pp. 21 722–21 733, 2019.

[6] S. A. Rao, M. Pai, M. Boussedjra, and J. Mouzna, "Gpsr-l: Greedy perimeter stateless routing with lifetime for vanets", in *2008 8th International Conference on ITS Telecommunications*, IEEE, 2008, pp. 299–304.

[7] Setiabudi *et al.*, "Performance comparison of gpsr and zrp routing protocols in vanet environment", in *2016 IEEE Region 10 Symposium (TENSYMP)*, IEEE, 2016, pp. 42–47.

[8] Z. Houssaini *et al.*, "Improvement of gpsr protocol by using future position estimation of participating nodes in vehicular ad-hoc networks", in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, IEEE, 2016, pp. 87–94.