

# On Greedy Routing in Dynamic UAV Networks

Mehrdad Khaleedi, Arnau Rovira-Sugranes, Fatemeh Afghah, and Abolfazl Razi  
School of Informatics, Computing and Cyber Systems, Northern Arizona University

**Abstract**—Unmanned aerial vehicles (UAVs), commonly known as drones, are becoming increasingly popular for various applications. Freely flying drones create highly dynamic environments, where conventional routing algorithms which rely on stationary network contact graphs fail to perform efficiently. Also, link establishment through exploring optimal paths using hello messages (as is used in AODV algorithm) deems extremely inefficient and costly for rapidly changing network topologies.

In this paper, we present a distance-based greedy routing algorithm for UAV networks solely based on UAVs' local observations of their surrounding subnetwork. Thereby, neither a central decision maker nor a time consuming route setup and maintenance mechanism is required. To evaluate the proposed method, we derive an analytical bound for the expected number of hops that a packet traverses. Also, we find the expected end-to-end distance traveled by each packet as well as the probability of successful delivery. The simulation results verify the accuracy of the developed analytical expressions and show considerable improvement compared to centralized shortest path routing algorithms.

## I. INTRODUCTION

Unmanned Aerial Vehicles (UAV) have recently attracted significant interest in many civilian, commercial and military applications. The popularity of UAVs emanate from their low cost, rapid deployment, and ability to fly above obstacles. It is anticipated that the global market revenue of UAVs reach \$11.2 billion by 2020 [1]. Some applications require deployment of a large number of UAVs to complete designated tasks. A network of autonomous UAVs form a flying Ad-hoc network (FANET), which provides more sensing and actuation capabilities. However, it also poses many challenges in designing networking protocols.

One of the main challenges in FANETs is the high degree of mobility which causes frequent changes in network topology. As such, conventional communication protocols face a considerable performance degradation. In routing, for instance, link breakage and frequent topology changes can lead to packet loss, excessive re-transmissions and eventually increased delay.

For environments with low degrees of mobility or with geographically confined movements, several routing protocols have been proposed recently. For instance, in vehicular Ad-Hoc networks (VANETs), a routing protocol based on the Dijkstra's algorithm [2] and an optimized multicast routing protocol [3] have been developed. In [4], a stable routing protocol is presented which finds the most stable path by considering velocity, direction and the link expiration time using fuzzy logic. In addition, learning-based routing protocols (e.g. Q-routing) have been proposed to learn the link states based on the current transmission experience [5]. Such algorithms rely on the assumption of low speed vehicles moving in a confined two dimensional space with limited and predefined movement patterns dictated by obstacles and roads. These assumptions, however, are not realistic in FANETs with UAVs moving freely in space with potentially high speeds.

For dynamic UAV networks, however, there have been few efforts on routing algorithm design. For instance, the authors

in [6], presented the RARP protocol which utilizes GPS information to estimate the duration of a path. This algorithm is based on AODV [7] and requires a route setup phase before transmission. To avoid repeated route setups the algorithm assumes that nodes keep their current movement patterns for some period of time which is a typical assumption in mobile Ad-Hoc networks (MANETs). This assumption is not valid for network of autonomous UAVs. This protocol is also evaluated using a mobility model primarily developed for low-speed terrestrial nodes, hence not suited for freely flying drones. Another recently proposed routing protocol for UAV networks is the Predictive-OLSR [8] which utilizes GPS coordinates to estimate the quality of a link. However, they adopt a restrictive assumption that only the source node is mobile.

In this paper, we present a distance greedy routing algorithm for dynamic UAV networks. This algorithm relies on local forwarding decisions and does not require a route setup phase. This low-complexity algorithm imposes no additional signaling overhead to the system, hence well suited to dynamic UAV networks. In order to analyze the performance of the greedy routing, we derive analytical lower and upper bounds for the expected number of hops that a packet traverses. Using this result, we find the expected end-to-end distance that a packets travels using the greedy algorithm, which can be a ground to optimize distance-based performance metrics such as end-to-end delay and transmission power consumption. We also derive the probability of successful delivery which is the probability that all intermediate nodes along the path can forward the packet to a node closer to the destination. Finally, simulation results verify the accuracy of the derived analytical results and also indicate superior performance of the greedy routing compared to the conventional shortest path routing based on Dijkstra's algorithm [9].

## II. SYSTEM MODEL

We consider a FANET with  $N$  UAV nodes  $(n_1, n_2, \dots, n_N)$ , distributed uniformly in a  $L \times L$  rectangular area. Let  $R$  denote the radius of the circular communication range of a UAV node, then the set of neighbors for node  $n_i$  is defined as:

$$S_i = \left\{ n_j : d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R \right\}$$

where  $d_{ij}$  denotes the Euclidean distance between nodes  $n_i$  and  $n_j$ . Also, we use the popular mobility model for UAV nodes which integrates linear and circular motions [10]. Transition between two mobility modes occur based on an underlying Markov process with adjustable transition probabilities. The mobility parameters include speed and direction of linear motion, and initial phase, angular speed, and radius of circular motion. Speed and radii are drawn from exponential distributions and the rest of parameters are uniformly distributed. The parameters are initialized when transition occurs between the two states and remain constant until the next transition.

UAV nodes do not keep track of entire network topology, but each node needs to know the location of its neighbors.

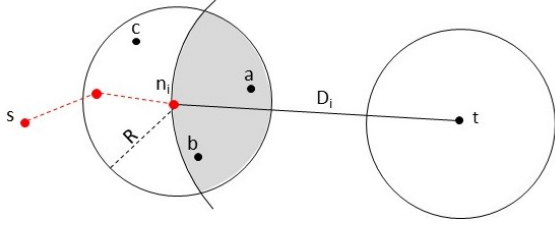


Fig. 1. The intermediate node,  $n_i$ , forwards the packet to a neighbor which is closer to the destination,  $t$ , than its other neighbors ( $n_{i+1} = a$  here). The shaded area called *progress area* which is the valid locations for neighbors that have lower distances to the destination than the current distance,  $D_i$ .

We assume the anticipated locations of neighbors can be predicted by a node, either through a model-based motion trajectory prediction method [11] or exploiting online path-planning information by UAVs [12]. Also, since the algorithm works based on the remaining distance to the destination, the source node needs to know the location of the destination. This information can be embedded into the packet to inform the intermediate nodes about the destination's location.

### III. THE DISTANCE GREEDY ROUTING ALGORITHM

Consider a source node  $s$  who wants to send a packet to a destination node  $t$  which is located at distance  $D$ . The distance greedy algorithm works based on a simple forwarding rule. At each step,  $i$ , the packet is passed by the current node  $n_i$  to a neighbor node  $n_j$  which is closest to the destination, i.e.  $n_{i+1} = \operatorname{argmin}_{n_j \in S_i} d_{j,t}$ . The packet is passed only if the next

node makes a progress, (i.e. if the next node is closer to the destination than the current node:  $d_{i+1,t} < d_{i,t}$ ). If such a neighbor does not exist the current session fails and the packet journey is re-initiated. This algorithm continues until the packet is delivered to the destination. This constraint is required to ensure a loop-free path towards the destination. In this way, we guarantee a progress at each step provided that there is at least one neighbor in the *progress area*.

Fig. 1, represents one iteration of this algorithm. The shaded area represent the valid locations for the next node. This *progress area* is the intersection of two circles centered at  $n_i$  and  $t$  with radii  $R$  and  $D_i$ , respectively. Here,  $D_i$  is the remaining distance to the destination, once the packet reaches node  $n_i$ . The algorithm chooses a node which makes the highest progress towards the destination (here  $n_{i+1} = a$ ).

### IV. ANALYSIS OF THE GREEDY ROUTING ALGORITHM

In this section, we evaluate the performance of the greedy routing algorithm through several steps. We first bound the expected number of hops a packet travels from source to destination. Next, using the results we find the expected end-to-end distance traversed by the greedy algorithm. Finally, we find the probability of successful delivery in terms of transmission range, number of nodes and the size of the area through finding the probability of having at least one node in the *progress area* throughout the selected path.

#### A. Analysis of The Number of Hops

In order to bound the expected number of hops for sending a packet from source to destination, we first find the probability distribution of the progress made at each hop.

Let  $A_{D_i}(R, D_i)$  denote the area of the shaded region in Fig. 1 which is the intersection of two circles with centers at  $n_i$ ,  $t$  and with radii  $R$  and  $D_i$ , respectively. Also, let  $X_i$  be a random variable representing the remaining distance to the destination at the  $i^{\text{th}}$  hop. In fact,  $X_i$  represents the distance from destination  $t$  to its closest node in the shaded area. The probability of  $X_i$  being at least  $x$  equals the probability that there are no nodes in the area  $A_{D_i}(R, x)$  which is the overlap of two circles with centers at  $n_i$  and  $t$ , which are at distance  $D_i$  of each other, and with radii  $R$  and  $x$ , respectively. Since, nodes are distributed uniformly, the number of nodes in any region with area  $A$  follows a Binomial distribution with  $N$  trials and success probability of  $\frac{A}{L^2}$ . Thus, we find

$$P[X_i \geq x] = \left(1 - \frac{A_{D_i}(R, x)}{L^2}\right)^N, \quad (1)$$

where  $D_i - R \leq x < D_i$  and we can use geometric analysis to find the area  $A_{D_i}(R, x)$  as

$$\begin{aligned} A_{D_i}(R, x) &= R^2 \cos^{-1} \left( \frac{D_i^2 + R^2 - x^2}{2D_i R} \right) \\ &\quad + x^2 \cos^{-1} \left( \frac{D_i^2 + x^2 - R^2}{2D_i x} \right) \\ &\quad - \frac{1}{2} \sqrt{(R - D_i + x)(D_i - R + x)(D_i + R - x)(D_i + R + x)}. \end{aligned} \quad (2)$$

Now, we can find the probability distribution of the progress made at each hop. Let  $Y_i = D_i - X_i$  denote the progress the  $i^{\text{th}}$  hop, we have:

$$P[Y_i \leq y] = \begin{cases} 0 & y < 0 \\ \left(1 - \frac{A_{D_i}(R, D_i - y)}{L^2}\right)^N & 0 \leq y \leq R \\ 1 & y > R \end{cases} \quad (3)$$

The probability density function (PDF) of  $Y_i$  can be computed by taking the derivative of its distribution function in (3). Note that there is a discontinuity point at  $Y_i = 0$ , therefore we can write the PDF as:

$$f_{Y_i}(y) = P[Y_i \neq 0] f_{Y_i}^c(y|Y_i \neq 0) + P[Y_i = 0] \delta(y), \quad (4)$$

where  $f_{Y_i}^c(y|Y_i \neq 0)$  denotes the continuous part conditioned on progress, which is the derivative of the distribution function for  $Y_i$  between 0 and  $R$ . Knowing the PDF, we can find the expected progress at the  $i^{\text{th}}$  hop as follows:

$$\mathbb{E}[Y_i] = \int_0^R y f_{Y_i}(y|Y_i \neq 0) dy.$$

Using integration by parts, we have:

$$\begin{aligned} \mathbb{E}[Y_i] &= y \left(1 - \frac{A_{D_i}(R, D_i - y)}{L^2}\right)^N \Big|_0^R \\ &\quad - \int_0^R \left(1 - \frac{A_{D_i}(R, D_i - y)}{L^2}\right)^N dy \\ &= R - \int_0^R \left(1 - \frac{A_{D_i}(R, D_i - y)}{L^2}\right)^N dy, \end{aligned} \quad (5)$$

where the first term equals  $R$  since  $A_{D_i}(R, D_i - R) = 0$  as there is no intersection between two circles with centers at  $n_i$ ,  $t$ , which are at distance  $D_i$  of each other, and with radii  $R$  and  $D_i - R$ , respectively.

Now that we have the average progress at each hop, we find the number of hops that a packet traverses to reach a destination located at distance  $D$  of the source node. The number of hops is of this form;  $n = m + 1$  where  $m$  is the number of hops needed to reach the communication range of the destination. That means, the first  $m$  hops takes the packet to the destination's communication range where there is only one hop left to the destination node. We have:

$$\sum_{i=1}^{m-1} Y_i < D - R \leq \sum_{i=1}^m Y_i. \quad (6)$$

It should be noted that  $m$  is a stopping time step with respect to the sequence  $Y_i$ . That means, at time  $m$  we have enough information to stop and we do not need any future information to decide. For a special case of stopping times when the sequence of random variables are independent and identically distributed (i.i.d.), we can utilize the Wald's equation [13] to find the sum of random variables up to time  $m$ .

*Lemma 1 (Wald's Equation [13]):* If  $\tau$  is a stopping time with respect to an i.i.d. sequence  $\{X_i : i \geq 1\}$ , and if  $\mathbb{E}[\tau] < \infty$  and  $\mathbb{E}[X] < \infty$ , then

$$\mathbb{E}\left[\sum_{i=1}^{\tau} X_i\right] = \mathbb{E}[\tau] \mathbb{E}[X].$$

In our case, however, the sequence  $Y_i$  is not i.i.d., therefore, we cannot directly use the Wald's equation. For this reason, we first find the number of hops using some i.i.d. random variables  $Z_i$ . Next, we replace  $Z_i$  with i.i.d. random variables that upper bound and lower bound  $Y_i$ . Thereby, we conclude about the bounds on the number of hops a packet travels. Using Lemma 1 for i.i.d. random variables  $Z_i$ , we have

$$\mathbb{E}\left[\sum_{i=1}^m Z_i\right] = \mathbb{E}[m] \mathbb{E}[Z]. \quad (7)$$

from the inequality in (6), we have:

$$\mathbb{E}\left[\sum_{i=1}^m Z_i\right] \geq D - R. \quad (8)$$

Combining (7) and (8) we get:

$$\mathbb{E}[m] \geq \frac{D - R}{\mathbb{E}[Z]}. \quad (9)$$

To find an upper bound for  $\mathbb{E}[m]$  we use the left inequality in (6) and the fact that the progress at each hop is at most  $R$ ,

$$\sum_{i=1}^m Z_i \leq \sum_{i=1}^{m-1} Z_i + R < D.$$

taking expectation we have

$$\mathbb{E}\left[\sum_{i=1}^m Z_i\right] < D.$$

using (7), we get:

$$\mathbb{E}[m] < \frac{D}{\mathbb{E}[Z]} \quad (10)$$

Now, utilizing (10) and (9) the expected number of hops  $\mathbb{E}[n] = \mathbb{E}[m] + 1$  can be bounded as follows

$$\frac{D - R}{\mathbb{E}[Z]} + 1 \leq \mathbb{E}[n] < \frac{D}{\mathbb{E}[Z]} + 1. \quad (11)$$

As mentioned earlier the random variables  $Y_i$  are not i.i.d. and we need to bound them using i.i.d. random variables to be able to use the result in (11).

For this purpose, let us define the progress at each hop as a function of the remaining distance to the destination, as  $Y(D)$ . It is worth noting that the  $Y(D)$  is a non-decreasing function of the remaining distance  $D$ . More precisely, for  $D \geq D'$  we have  $P[Y(D) > y] \geq P[Y(D') > y]$ . Intuitively, if there is more distance to the destination it is more likely that we make more progress than the case of having less distance to the destination. This can be shown formally using (3), as follows

$$P[Y(D) > y] = 1 - \left(1 - \frac{A_D(R, D - y)}{L^2}\right)^N \geq \quad (12)$$

$$1 - \left(1 - \frac{A_{D'}(R, D' - y)}{L^2}\right)^N = P[Y(D') > y],$$

where the inequality follows from the fact that if  $D \geq D'$  we have  $A_D(R, D - y) \geq A_{D'}(R, D' - y)$ .

Now, observe that at any of the  $m$  hops, the distance between the intermediate node and the destination is between  $R$  and  $D$ . Using (12), we get

$$P[Y(R) > y] \leq P[Y_i > y] \leq P[Y(D) > y]. \quad (13)$$

for  $i = 1, \dots, m$

Thus, in (11) we can replace  $Z_i$  with i.i.d. random variables  $Y(D)$  to find a lower bound for the number of hops. Similarly, we can use i.i.d. random variables  $Y(R)$  to find an upper bound.

$$(1_{\{D \geq R\}}) \cdot \frac{D - R}{\mathbb{E}[Y(D)]} + 1 \leq \mathbb{E}[n] < \frac{D}{\mathbb{E}[Y(R)]} + 1, \quad (14)$$

where  $\mathbb{E}[Y(D)]$  can be computed using (5) and  $1_{\{D \geq R\}}$  is an indicator to account for the case where source and destination are immediate neighbors.

## B. Analysis of The End-to-End Delay

In this section, we analyze the total distance a packet travels from source to destination using the distance greedy routing. Considering the delay is proportional to the distance, this will give us the end-to-end delay metric.

Consider the forwarding scenario at the  $i^{th}$  hop, depicted in Fig. 2, where node  $b$  has been chosen by the intermediate node  $a$  and we make a progress of  $Y_i$  towards the destination. We want to find the distance traveled at the  $i^{th}$  hop (i.e. the length of the line  $ab$  in Fig. 2), which we denote it by  $W_i$ . Given the progress  $Y_i$ , we know that node  $b$  should be on the arc  $cc'$ . Considering the fact that nodes are uniformly distributed, node  $b$  can be anywhere on the arc  $cc'$  with equal probability. Also, the distance of any node on the arc  $cc'$  from the transmitting

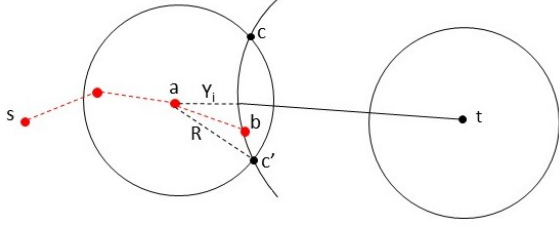


Fig. 2. The length of line  $\bar{ab}$  can range between  $Y_i$ , the progress made at the  $i^{th}$  hop, and the transmission range  $R$ .

node  $a$  ranges between  $Y_i$  and  $R$ . Therefore, we can roughly estimate that the distance from node  $a$  to node  $b$  is uniformly distributed between  $[Y_i, R]$ . That is  $W_i \sim U[Y_i, R]$  for the  $i^{th}$  hop.

Now, we can find the expected end-to-end distance traveled by a packet as

$$\mathbb{E} \left[ \sum_{i=1}^n W_i \right] = \sum_{i=1}^n \mathbb{E}[W_i] = \sum_{i=1}^n \frac{Y_i + R}{2} = \frac{1}{2}(D + \mathbb{E}[n]R), \quad (15)$$

where  $n$  is the number of hops whose expectation is characterized in (14). Also, we have used the fact that the progresses at each hop sum up to the distance between source and destination,  $\sum_{i=1}^n Y_i = D$ .

### C. Analysis of Network Density for Successful Delivery

In the greedy routing algorithm, we assume that there is always a neighbor to forward the packet towards the destination and thereby we ignore the possibility of the packet reaching an isolated node (more precisely, a node without any progress-making neighbors). In fact, if the network is dense enough or if the transmission range of nodes are large enough, such a situation can be avoided.

In this section, we find the probability of successful delivery<sup>1</sup>. First, we find the probability of a node being isolated which equals the probability of having no node in its *progress area*. To simplify the analysis, we estimate the *progress area* of a node by the half of its transmission region which faces the destination. Then, we can write the probability of a node's isolation,  $P_{iso}$ , as

$$P_{iso} \approx \left(1 - \frac{\pi R^2}{L^2}\right)^{N-1} \quad (16)$$

we can now solve (16) for  $R$  and find the minimum transmission range such that the probability of node isolation is less than  $\epsilon$

$$R > \sqrt{\frac{2}{\pi}} \sqrt{L^2 + L^2(\epsilon^{\frac{1}{N-1}})} \quad (17)$$

The probability of success equals the probability of no node along the path being isolated which is  $(1 - P_{iso})^n$ , where  $n$  denotes the number of hops that is bounded in expectation by (14)

$$P_{success} \approx \left(1 - \left(1 - \frac{\pi R^2}{2L^2}\right)^{N-1}\right)^n \quad (18)$$

<sup>1</sup>It is worth noting that by successful delivery we mean the packet travels from source to destination without facing an isolated node.

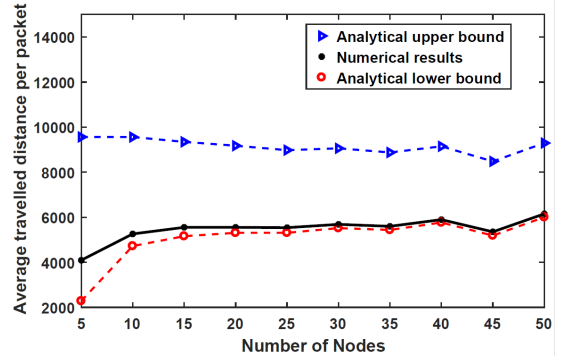


Fig. 3. Total travel distance from source to destination per packet vs the number of nodes in the network ( $N$ ): simulation results are compared against analytical lower and upper bounds.

## V. SIMULATION RESULTS

To test simulations results and prove the efficacy of the analysis work, random networks are generated using uniform distributions for the initial real positions in a  $L \times L$  grid. We use the mobility model explained in Section II to generate motion trajectories for  $N$  nodes. We use the actual positions for all nodes when quantifying the performance metric, but use the predicted positions when finding the optimal path. The predicted locations are the actual locations mixed with Normally distributed prediction noise of variance  $\sigma_N^2 = 10$ .

We use dynamic contact graph by making connections between nodes with pairwise distances below  $R$ . The rest of simulation parameters include number of nodes:  $N = 10$ , the grid size:  $L = 10$  km, communication range:  $R = 5$  km, average node velocities:  $\bar{v} = 50$  m/sec, and average waiting time:  $\bar{w} = 20$ , unless specified otherwise. Also the transition probability between the circular and linear motions is 20%. Finally, we note that for all figures, we take average over 100 runs of the algorithm with different initializations.

We first, verify the accuracy of the derived upper and lower bounds for two important performance metrics, namely the end-to-end delay per packet and the probability of success.

In Fig.3, we present the expected end-to-end distance per packet vs  $N$ . The upper and lower bounds are presented based on 15, where the bounds on the expected number of hops,  $\mathbb{E}[n]$  is obtained from (14). We note that the lower bound is tighter, which is due to the tightness of the lower bound in (14). The fluctuation in the results is due to the average distance between the source and destination ( $D$ ), which is a probabilistic value.

Another important performance indicator of the proposed algorithm is the probability of success, which means possibility of progress at all intermediate nodes (having at least one node in the current node's *progress area*), as characterized in (18) based on the average number of hops per packet in (14). Fig 4 suggests that as we increase the number of nodes, the network density increases and therefore the probability of getting stuck in an intermediate node with empty *progress area* diminishes. Similar to Fig. 3, the obtained lower bound is more accurate.

Now, we compare the performance of the proposed greedy algorithm with the conventional Dijkstra's shortest path algorithm in Fig. 5. We also evaluate the proposed algorithm with and without including predictive location information under different average node velocities. The results show that

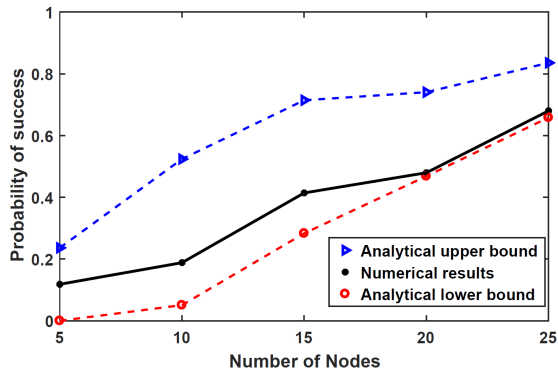


Fig. 4. Probability of delivery success versus the number of nodes: comparison between the simulation results and analytical lower and upper bounds

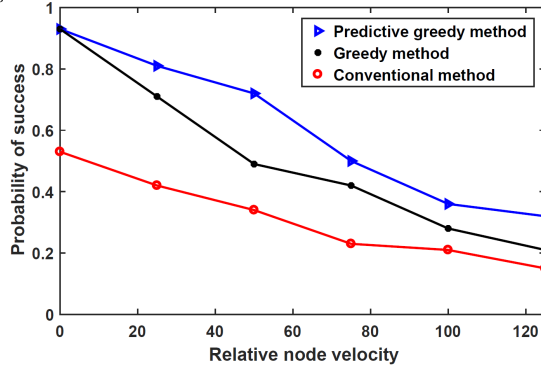


Fig. 5. Probability of delivery success for predictive greedy algorithm, static greedy algorithm and conventional Dijkstra's algorithm.

the probability of delivery success for the greedy method is higher than that of the conventional shortest path algorithm consistently for all average node velocities. Also, the predictive greedy method outperforms the static greedy algorithm, which shows including predictive location information decreases the probability of selecting nodes with empty *progress area*, as was expected. Finally, when the network is more dynamic, more nodes are subject to getting out of the communication ranges of their neighbors, and hence the probability of success declines.

Lastly, in Fig. 6 we present the average power consumption per packet to complete the path for the proposed predictive greedy algorithm and the standard Dijkstra's algorithm without including predictive information in order to show the practical utility of the proposed method. Since the power consumption is proportional to the sum of all link distances squared, considering predictive information provides a significant gain for our suboptimal algorithm. This gain is higher for more dynamic networks, since the inclusion of predictive locations is more beneficial for higher average node velocities.

## VI. CONCLUSION

In this paper, we studied the routing problem in dynamic UAV networks. Prior approaches fail to perform well in such dynamic environments due to the requirement of maintaining information about the network topology or using frequent route-establishment phases. We studied an agile distance-greedy routing algorithm which is low-complexity and the

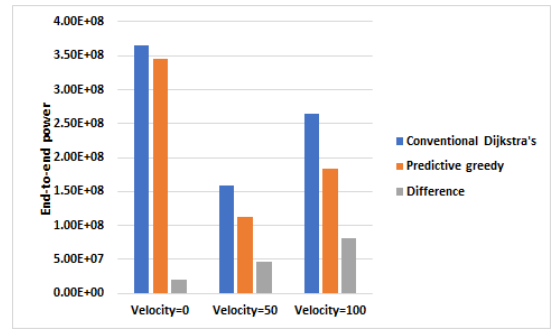


Fig. 6. End-to-end power comparison between the conventional Dijkstra's algorithm and the predictive greedy algorithm.

intermediate nodes take decisions solely based on the predicted locations of their neighbors. This algorithm is fully distributed and incorporates predicted locations into the algorithm, hence outperforms the centralized shortest algorithm. We characterized the number of hops, the probability of success delivery and the expected distance a packet travels based on system parameters. We plan to characterize the impact of the prediction uncertainty on the optimality of the selected path as an extension of this work.

## REFERENCES

- [1] "Gartner says almost 3 million personal and commercial drones will be shipped in 2017," Feb 2017. [Online]. Available: <http://www.gartner.com/newsroom/id/3602317>
- [2] J. d. Zhang, Y. j. Feng, F. f. Shi, G. Wang, B. Ma, R. s. Li, and X. y. Jia, "Vehicle routing in urban areas based on the oil consumption weight-dijkstra algorithm," *IET Intelligent Transport Systems*, vol. 10, no. 7, pp. 495–502, 2016.
- [3] W. Farooq, M. A. Khan, and S. Rehman, "Amvr: A multicast routing protocol for autonomous military vehicles communication in vanet," in *2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Jan 2017, pp. 699–706.
- [4] H. C. Premkumar, V. R. Budyal, and M. S. Kakkasageri, "Cognitive agent based stable routing protocol for vehicle-to-vehicle communication," in *2016 IEEE Annual India Conference (INDICON)*, Dec 2016, pp. 1–5.
- [5] S. P. Choi and D.-Y. Yeung, "Predictive q-routing: A memory-based reinforcement learning approach to adaptive traffic control," in *Advances in Neural Information Processing Systems*, 1996, pp. 945–951.
- [6] G. Gankhuyag, A. P. Shrestha, and S. J. Yoo, "Robust and reliable predictive routing strategy for flying ad-hoc networks," *IEEE Access*, vol. 5, pp. 643–654, 2017.
- [7] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*. IEEE, 1999, pp. 90–100.
- [8] S. Rosati, K. Kruelecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic routing for flying ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1690–1700, March 2016.
- [9] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
- [10] O. Bouachir, A. Abrassart, F. Garcia, and N. Larrieu, "A mobility model for uav ad hoc network," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2014, pp. 383–388.
- [11] G. Aoude, J. Joseph, N. Roy, and J. How, "Mobile agent trajectory prediction using bayesian nonparametric reachability trees," *Proc. of AIAA Infotech@ Aerospace*, pp. 1587–1593, 2011.
- [12] G. D. Goez, R. A. V. Velez, and J. S. B. Valencia, "Uav route planning optimization using pso implemented on microcontrollers," *IEEE Latin America Transactions*, vol. 14, no. 4, pp. 1705–1710, April 2016.
- [13] A. Wald, "Sequential tests of statistical hypotheses," *Ann. Math. Statist.*, vol. 16, no. 2, pp. 117–186, 06 1945.