Investigation of Cartesian Routing for Unmanned Aerial Vehicle Networks

Karam mheidy Al-sofy$^{1,2}$ and Sahar Abdulaziz Al-Talib$^1$

$^1$Computer Engineering Dept., Electronics Engineering, Ninevah University, Mosul, Iraq

$^2$Electronic Computer Center, Mosul University, Mosul, Iraq

Corresponding author’s e-mail: eng.k.alsofy@uomosul.edu.iq

Abstract. Unmanned aerial vehicles (UAVs), commonly known as drones, are a pilotless aircraft that does not require any direct human intervention for flying. It can move autonomously based on its pre-programmed software, or can be remotely controlled. Besides its basic plane components, it also contains some computing devices and sensors for determining its position and for gathering information from the mission area.

Flying Ad hoc Networks (FANET) are comprised of autonomous flying vehicles. It is a special case of mobile ad hoc networks (MANETs) characterized by a high degree of mobility. One of the most difficult and complex challenges facing FANETs is the routing process. To achieve this goal the Cartesian Orientation Protocol is used. The Cartesian routing protocol exploits geographic information for UAVs rather than using the address to direct packets to its destination. In this work, the most prominent algorithms based on geographic location has been highlighted that have been adapted to work in a three-dimensional environment and implemented in a common scenario and evaluated through several measures such as Packet Delivery Ratio, Path Dilation and End-to-End Delay.

Keywords: UAVs, FANETs, Cartesian routing, GPS

1. Introduction

The advances in microelectronics have decreased the size and weight of wireless networks equipment's and sensor communication technologies, allowing the exploration of new ways to deploy wireless infrastructure. Recently there has been increasing interest in aerial communication networks, as shown by research and industry efforts. It has been possible to produce unmanned aerial vehicle (UAV) systems, commonly known as drones, which can fly autonomously or can be operated remotely without carrying any human personnel[1][2]. Although single-UAV systems have been in use for tens of years, instead of developing and operating one large UAV, using a group of small UAVs has many advantages. However, multi-UAV systems face communication problems. The progress of embedded systems and the advance of micro electromechanical systems, it has been possible to produce small or mini UAVs at a low cost. However, the capability of a single small UAV is limited. Collaboration and coordination of multiple UAVs can create a system that is beyond the capability of only one UAV.

The advantages of multi-UAV systems can be summarized in terms of Cost, Scalability, Survivability, and Speed-up Small radar cross-section) [2][3]. Depending on the application and goals, one needs to use an appropriate type of UAV that can meet various requirements imposed by the required Quality-of-Service (QoS), federal regulations and the nature of the environment. In fact, to properly use UAVs for any specific wireless networking application, many factors such as the UAVs’ capabilities and their flying altitudes must be taken into account [4]. In general, UAVs can be categorized, based on their altitudes, into High Altitude Platforms (HAPs) and Low Altitude Platform (LAPs). HAPs have altitudes above 17 km and are typically semi- stationary [5][6]. LAPs, on the other hand, can fly at altitudes of tens of meters up to a few kilometres, can quickly move, and are flexible [6].
2. Related works
Cartesian routing is a novel packet routing methodology in which a packet’s route is determined by the position of the router relative to that of the destination. Cartesian routing differs from existing provider-based unicast routing in that routing tables are unnecessary since communications are topologically dependent, thereby potentially reducing router and network overheads. For example, routing decisions, which can take O(log(n)) to O(n) time using routing tables is reduced to O(1) in Cartesian routing [7].

Larry Hughes, et al. [7] describes Cartesian routing for unicast communications within a local or metropolitan environment (e.g., limited areas, including buildings, suburbs, and small towns that exhibit an anthropogenic ordering) using two-dimensional address structures, such as latitude and longitude. A Cartesian network is constructed from two types of router: collector (for horizontal communications) and arterial (for both ‘horizontal’ and ‘vertical’ communications). Cartesian routing requires minimal state: typically, routers need only know their location and the reachability of arterial routers on their collector.

Brad karp and H.T. Kung, [8] presented Greedy Perimeter Stateless Routing (GPSR), a novel routing protocol for wireless datagram networks that uses positions of routers and packet’s destination to make packet forwarding decisions. GPSR makes greedy forwarding decisions using only information about a router’s immediate neighbours in the network topology. GPSR generates routing protocol traffic in a quantity independent of the length of the routes through the network, and therefore generates a constant, low volume of routing protocol messages as mobility increases, yet doesn’t suffer from decreased robustness in finding routes.

A.E. Abdallah, et al. [9] studied ad-hoc routing in 3D space which is more realistic than routing in 2D space. The authors extend AB randomized algorithm from 2D to 3D environment, improve a new version of project-face routing algorithms (CFace(i)) which decreases the path dilation to almost 50%, and proposed two new hybrid routing algorithms, AB3D-CFace(1)-AB3D and AB3D-CFace(3), which combine the efficiency of progress-based algorithms with high delivery rate of face routing. Their experiments show that these hybrid algorithms ended increase the delivery rate while keeping the average dilation if the route much smaller than face routing.

Roland Flury and Roger Wattenhofer, [10] explained that a cubic routing stretch constitutes a lower bound for any local memory less routing algorithm, and propose and analyze several randomized geographic routing algorithms which work well for 3D network topologies. For unit ball graphs, the authors present a technique to locally capture the surface of holes in the network, which leads to 3D routing algorithms similar to the greedy-face-greedy approach for 2D networks.

Cong Liu and Jie Wu, [11] tackled the problem of efficient geometric routing in 3D networks. The authors proposed routing on hulls, a 3D analogue to face routing, and presented the first 3D partial unit Delaunay triangulation (PUDT) algorithm to divide the entire network space into a number of closed subspaces.

Luiz Filipe et al., [12] investigated the performance of greedy forwarding geographic routing in the Internet of Drones, deriving analytical results and demonstrated mathematically and experimentally that increasing the network density or the radio communication range will increase the probability Pn of a node having a neighbor closer to the destination and therefore, the end-to-end packet delivery ratio (PDR).

3. Flying Ad hoc Networks (FANET)
FANETs are a special case of mobile ad hoc networks (MANETs) characterized by a high degree of mobility. In a FANET, the topology of the network can change more frequently than in a typical MANET or VANET [2][13]. As an improved mobile ad hoc network, FANETs use the aircraft Unmanned Ariel Vehicle (UAV) as a node for transmitting, receiving, forwarding or sharing information over the air wireless communication. A network can be set at any time and any place without any fixed facilities and realize a multi-aircraft system. The FANETs have similar features as traditional MANETs, such as no center, multi-hop, self-organization, and self-healing, but are given
specific design goals and network characteristics [14][15]. According to this definition, single UAV systems cannot form a FANET, which is valid only for multi UAV systems. On the other hand, not all multi UAV systems form a FANET. The UAV communication must be realized by the help of an ad hoc network between UAVs. Therefore, if the communication between UAVs fully relies on UAV-to-infrastructure links, it cannot be classified as a FANET [2].

3.1. Applications of FANET
The usage of FANETs has attracted more attention and importance recently because they are not only resilient against isolated attacks or node failure, but they are also relatively economical. As FANET nodes operate in the sky, line-of-sight communications is possible most of the time. Besides, its easy installation, versatility, flexibility, adaptability and rapidly deployment, FANETs are becoming a promising solution for various commercial, military and civilian applications [16], such as disaster inspection, search and rescue operations, border surveillance, forest fire detection, relaying networks, wind estimation, civil security, agricultural purposes and traffic monitoring.

3.2. Challenges of FANET
FANET is somewhat different from traditional MANETs and VANETs; but the fundamental idea is the same: having mobile nodes and networking in an ad-hoc manner. Hence, in a FANET, some challenges are valid as in a VANET while facing additional challenges. Although, many researches have been trying to increase the efficiency of networks with flying nodes, there are still many unsolved problems, which should be explored in future works such as (National Regulations, Routing, Path Planning, Quality of Service (QoS), Integration with a Global Information Grid (GIG), Coordination of UAVs and manned aircrafts, Coordination of UAVs and manned aircrafts, UAV mobility and placement and Standardize FANETs [2][3][15]. Routing is the selected subject to be investigated in this work.

4. Position-Based Routing
Position-based routing protocols have been proposed to assume knowledge of the geographical position information of UAVs to supports efficient routing. In this type of protocols, they assume that the source UAV knows about the physical position of the communicating UAVs and sends message to the destination UAVs without route discovery. Generally, to obtain location information, UAVs use a location service such as the Global Positioning System (GPS), or other types of location services. To acquire the position of the neighbours, UAVs make use of a beaconing mechanism in which each UAV sends a beacon to its neighbour UAVs, containing its position. This routing is perfect for highly dynamic UAV communication networks. Position-based routing can be classified into two categories: single-path-based, and multi-path-based. Both single- and multi-path-based routing protocols are categorized further into heterogeneous networks, delay tolerant networks (DTNs), and non-delay-tolerant networks (Non-DTNs). In general, Position-based protocols have the following characteristics [17][18][19][20]:

- Each node can determine its position (longitude, latitude and altitude), the position of its neighbors (usually 1 hop);
- The destinations location information is assumed known a-priori (e.g., acquired through external means);
- Nodes store the information about their neighbors (e.g. position, speed, direction) in a neighbor table;
- The next-hop decision can be made based on the location of the current node (the node holding the packet), its neighboring nodes and the destination node.

4.1. Challenges and Issues of position based routing
Since the main concern in this work is the position based routing (Cartesian routing), so it is necessary to explore the challenges and issues related to this type. The various types of data routing for the UAVs networks have been discussed, and advance knowledge about the mechanism of action and the characteristics of each type was formed. The following points summarize the challenges facing this type of routing [21]:

- Routing stretch is the ratio of measured path length and shortest path length from source to destination in the given network. There are two types of routing stretch (hop stretch and path stretch). Hop stretch is the ratio of measured hop distance to the shortest hop distance between source and destination. Path stretch is the ratio of the measured Euclidean distance to the shortest Euclidean distance between the source and destination nodes. For a good routing scheme, the routing stretch should be near to 1.

- Local minimum is a problem of greedy approach where a node doesn’t have any next neighbor nearer to the destination. In this case, the greedy approach cannot guarantee the delivery of the message even if another alternate path is available.

- Localization is an ability to find the position information of the node. Finding the accurate location of wireless devices is a crucial requirement for location-aware protocols. Hence, it is an essential requirement for geographical routing protocols. One alternative is to use GPS but it is costlier and inefficient.

- Mobility states the movement prediction of mobile nodes in the network. Many researchers have proposed different mobility models to predict mobility, but predicting accurate mobility is an hard problem. For this reason, it is the most challenging task for aerial networks.

- Energy Efficiency: The nodes in wireless networks are usually powered and have a limited power supply. On some networks, recharging the node may not be feasible or cost-effective. Thus, the design of the energy-efficient routing algorithm is an important research issue especially when proposed for 3D networks.

- Load balancing is part of energy efficiency. It ensures that every node in the network must have an equivalent load in order to improve the life of the network. Hence, all nodes should die about the same time.

- Security in ad hoc wireless networks, each node acts as a potential router, and therefore, each node is subject to security attacks through routing protocols. Typical routing attacks are black hole attacks, hole attacks, basin attacks, defective routing information attacks, selective forwarding attacks, flood welcome attacks, etc. An alternative solution to such attacks is to manage trust as data should only be directed to trusted nodes.

### 4.2. Common techniques for position-based routing

Having explained previously the main problems and challenges facing the geographic routing process of 3D networks, we will try to present in this section the most important and most common algorithms used in such scenarios in order for us to choose the appropriate protocol for each application.

#### 4.2.1. Greedy Forward

Greedy is a simple progress-based forwarding strategy. The purpose of Greedy Forwarding is to minimize the number of hops in which a data packet can make during its transition to the target destination. The principle is to select the geographically closest node to the target destination as a relay node and so on until the packet reaches its destination (Figure 1). If there is no neighbour node closer to the destination (i.e., there is a void), the algorithm fails and the node storing the packet is called local minimum [21][22][23].

#### 4.2.2. Compass

The Compass (or Directional, DIR) algorithm uses the direction of nodes to select the best forwarding node. It uses the location information of nd to calculate its direction. Then, it forwards the packet to
the neighbour node that minimizes the angle between nc and nd. In (Figure 1) node nc selects node n4 as the next node, since the angle $\angle n4ncnd$ is the smallest among all [24].

### 4.2.3. Most forward (MFR)

It is very similar to Greedy, but, in this case, nc forwards the packet to the neighbour node (nu) whose projection on the line (nc nd) is closer to nd. If the packet reaches a local minimum (there is no neighbour projection that makes progress from nc towards nd), the algorithm fails. In most cases MFR chooses the same path as Greedy. In (Figure 1) nc selects n7 as the next node, since the latter has the smallest projected distance to nd on the line (nc nd) [25].

### 4.2.4. Ellipsoid

In the Ellipsoid algorithm, nc forwards the packet to the neighbor node nu that minimizes the sum of the distance from nc to nu and the distance from nu and nd. Unfortunately, if the packet reaches a local minimum, the algorithm fails. According to (Figure 1), the node nc will select n4 as the next node to handle the packet [26].

![Figure 1. Illustration of several next node chosen by nc using the progress-based forwarding strategies [20].](image)

### 4.2.5. Parametric AB3D (PAB3D)

PAB3D is a randomized algorithm that tries to solve the local minimum problem described previously, by randomly choosing the next node from a subset of the current neighbouring nodes to make progress toward a destination. Therefore, the idea of randomized algorithms is presented in order to avoid the emergence of loops by exploiting a random walk approach. PAB3D fails, when the number of hops in the path computed so far exceeds a threshold value, here called Time To Live Random (TTLR). In 3D environments, the authors proposed new algorithm called Above/Below 3D (AB3D) and, instead of a line, they use a plane to divide the 3D space in two regions as shown in (Figure 2) [27][28].

In general, a parametric algorithm called parametric AB3D can be defined that has four attributes [20][27][28]:

- M is the number of possible candidate neighbors to choose from NE(nc).
- R is the name of the progress-based strategy used to choose the M candidates, and it is one of Random, Compass, Greedy or Most Forward, as in AB3D.
S is used to represent the probability weighting when randomly choosing, and it is one among Uniform, Distance, Angle, and Projected Distance. Probability weightings are defined as in AB3D.

P is a Boolean flag indicating whether to define the candidates over and below the plane PL1 (or even over and below the plane PL2, if $M = 5$), or select candidates without considering the planes.

**Figure 2.** In AB3D, plane PL1 passes through ns, nd and n1, plane PL2 is orthogonal to PL1. Both planes contain the line (nsnd) [20].

### 4.2.6. Greedy-Random-Greedy (GRG)

It belongs to the progress/randomization-based class and uses Greedy as the primary stage and a randomized algorithm as a recovery strategy. Referring to PAB3D as the randomized algorithm, obtaining Greedy-PAB3D-Greedy (G-PAB3D-G), its general algorithm starts with a greedy approach until it finds a local minimum nc. At this point, G-PAB3D-G stores the distance $\text{dist}(\text{nc}; \text{nd})$, and switches to the random phase, as recovery strategy, where nc randomly selects one nu of its neighbouring nodes, using the steps defined in PAB3D. If $\text{dist}(\text{nu}; \text{nd}) < \text{dist}(\text{nc}; \text{nd})$, the algorithm resumes the greedy forwarding, otherwise it continues with PAB3D [10].

### 4.2.7. Projective Face

The face routing algorithms are based on planar graph traversal (Gabriel graph or relative neighbourhood graph. The algorithm traverses through the boundary of the face using left-hand rule (or right-hand rule). After traversing the whole face, it finds the nearest node to the destination. After that, it proceeds by traversing the next face closer to the destination. This process will continue until reaching the destination [21][29][30].

The face routing is the first geographical routing algorithm that guarantees delivery of the message, if at least one path is possible. But, the planar graph traversal is not suitable for 3D networks. So, a proposed solution is to project the nodes onto a plane, as seen in (Figure 3), to perform the face algorithm. The first extension of face-based strategy in 3D space uses two orthogonal planes intersecting at the line connecting source and destination. Since the delivery rate is not guaranteed, the algorithm makes use of a local threshold value, Time To Live Face (TTLF), in order to terminate the algorithm in case it does not reach the destination within TTLF hops to avoid stuck in a loop. More precisely, the TTLF counter is started twice, once for the first plane and then for the orthogonal plane, obtaining a global threshold value, $\text{TT L} = 2 \times \text{TT LF1}$ [21][29][30].
4.2.8. Coordinate Face(3)
Coordinate Face(3) or called CFace(3), uses another set of projection planes, which is composed by the planes xy, xz and yz for the projection of the nodes. With CFace(3) all the nodes are projected on the first xy plane (node:z = 0) and then the face routing is started on the projected graph. If the packet does not arrive at the destination (TTLF has expired), the original coordinates of all nodes are projected on the xz plane (node:y = 0) and face routing is performed again. If the packet does not reach the destination, the original coordinates of all nodes are projected on the yz plane (node:x = 0) and face routing is performed again. If the packet does not arrive even through this last plane, the algorithm fails. Note that, in this algorithm, TTL is at most 3×TTLF [9].

4.2.9. Adaptive Least-Square Projective Face (ALSP)
ALSP Face is the development or improvement of the projection face algorithm, which is achieved through three heuristics. First, heuristic Least-Squares Projection (LSP) Plane. Second, ALSP Face chooses the third point adopting a mathematical optimization technique (first heuristic) for finding the best fitting plane to the set of neighbour nodes. The second heuristic Adaptive Behaviour Scale (ABS) plane, defines a parameter called (ABS) that is used to determine when recalculate the LSP plane, in order to ensure that the plan is always appropriate for nc. The third heuristic Multi-Projection-Plane Strategy, uses a set of projection planes arranged in a fixed order around an axis. The algorithm switches between these planes, following the order, to disrupt any looping that may occur during routing. The performing of face routing on the additional projection plane, significantly increases the delivery rate. Therefore, the third heuristic tries to increase the number of projection planes [31].

4.2.10. Greedy-Face-Greedy (GFG)
The Greedy-Face-Greedy algorithm, also referred to as Greedy Perimeter Stateless Routing (GPSR) for 2D networks, uses a combination of greedy and face methods. With GFG, a flag is stored in each data packet. This flag can be set into greedy-mode and face-mode (or face-mode), indicating whether the packet is forwarded with either a Greedy or a Face approach [8][30].

The algorithm starts from ns with Greedy, setting the packet into greedy-mode and forwarding it. Each node that receives a greedy-mode packet searches among its neighbours the node that is closest to nd. If this node exists, the packet is forwarded to it; otherwise it is marked to face-mode. (Figure 4) shows an example where the packet starts from n1 and travels n2 and n3 in greedy mode, stopping at n3, that is the local minimum. Then, from n3, the face-mode is started, and forwards the packet on progressively closer faces of the planar graph, each of which is crossed by the segment [n3nd]. So, the packet reaches n4, n5 and n6 in face-mode. n6 is closer to nd than n3, so the packet can be returned to...
greedy-mode, and reaches nd. In a 3D context, GFG uses ALSP Face as face-mode, which offers the best performance in terms of delivery rate [8][30].

Figure 4. Performance of GFG on a 2D graph. Solid arrows represent greedy-mode forwarding, dashed arrows represent face-mode forwarding [20].

4.2.11. PAB3D-CFace(1)-PAB3D
This algorithm, initially conceived as AB3D-CFace(1)-AB3D, starts with PAB3D. Once the local threshold TTLR is reached and the algorithm reaches a local minimum, it switches to CFace(1). CFace(1) traverses one projective plane, which is chosen randomly from the xy, yz, or xz planes, starting from the node in which the algorithm is switched. At this point, TTLF is initialized to 0 and CFace(1) restarts. If the destination is not reached during this phase and TTLF is passed (a looping occurs), the algorithm goes back to PAB3D and TTLR restarts at 0 [9].

4.2.12. PAB3D-CFace(3)
This algorithm starts as PAB3D-CFace(1)-PAB3D, but instead of going back to PAB3D if the phase in a projective plane fails, PAB3D-CFace(3) tries the other two projective planes, defined as in CFace(3). This algorithm starts with the PAB3D stage and if the destination is not reached and the TTLR is passed, the algorithm switches to CFace(3) using the first xy plane. Again, if a loop occurred (TTLF is reached), it switches to yz plane, and finally the same process is repeated for xz plane [9].

4.2.13. LAR 3D
Is the extension of LAR in 3D space, (ns) computes the expected zone for (nd), which is a sphere around nd of radius equal to vmax (t1− t0) where t1 is the current time, t0 is the time stamp of the position information that ns has about nd, and vmax is the maximum speed of the node in the network. This zone is used to define the 3D flooding area, which is the minimum size rectangular box with ball (ns; r) in one corner and the expected zone in the opposite corner. In our case, since we consider a static scenarios (vmax = 0), the expected zone is ball (nd; r) [20][32].

4.2.14. PAB3D-LAR
PAB3D-LAR is a hybrid variant combining PAB3D with LAR. This algorithm tries to reduce the high path dilation of LAR. All the combination in PAB3D-LAR uses the same partitions as in PAB3D. The difference is that, while PAB3D selects only one of the M candidates chosen from the neighbourhood,
PAB3D-LAR sends the packet to all those selected candidates which are within the rectangular box defined as in LAR [9][32].

5. Implementation and Simulation Parameters

5.1. Network simulator (NS2)
Network Simulator (Version 2), widely known as NS2, is simply an event-driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours. Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community. NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism of the simulation, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL as shown in (Figure 5) [33].

![Figure 5. Basic architecture of NS2](image)

5.2. Simulation parameters
The simulation scenarios proposed for the protocol in this work consist of set of nodes distributed randomly within a space of a cube in 3D space. The value of length, width, and height is 1000 meters for each of them. The transmission range of the node is set to 200 meters. To evaluate the protocol, the nodes were supposed to be fixed in order to obtain the results and also to simplify the simulation process. A summary of simulation parameters is shown in Table 1.

| NS-2 version | Simulation time | simulation area | MAC type | Transmission range | Traffic type | Data packet size | Queue type | Number of nodes | Node mobility |
|--------------|----------------|----------------|----------|--------------------|--------------|-----------------|------------|----------------|--------------|
| 2.35         | 600 seconds    | 1000 x 1000 x 1000 (in meters) | IEEE 802.11g, FreeSpace model | 200 meters | CBR | 512 bytes | Drop Tail | 50, 100, 150, then 200 | Static |

It is worth noting that more than 500 runs were performed for each comparison and parameter. In each run, a different network topology has been prepared in which the nodes are randomly distributed
where the position of each node is different from its previous location. This is done to ensure the accuracy of the evaluation of the performance of each algorithm in delivering data packets from the source to the target.

5.3. Network Simulation Topology
In this scenario, coordinates of 50 nodes have been generated and distributed in 3D space within the implementation space. A UDP type of connection has been established between (node-0) as the source node that sends (1200) packets in a time period of (600) seconds towards the destination (node-1) as shown in (Figure 6). Fourteen routing algorithms were implemented each at a time are listed in Table 2 taking into consideration the setting of parameters for each of the Cartesian routing algorithms.

![Figure 6. Network topology of 50-nodes distributed randomly in space.](image)

**Table 2.** Cartesian routing algorithms

| Greedy | Compass | Most Forward | Ellipsoid | PAB3D | Projective Face | CFace | ALSP | GFG | P-C(1)-P | P-C(3) | G-PAB3D-G | LAR | PAB3D-LAR |
|--------|---------|--------------|-----------|-------|-----------------|-------|------|-----|----------|--------|------------|-----|-----------|
| 1      | 2       | 3            | 4         | 5     | 6               | 7     | 8    | 9   | 10       | 11     | 12         | 13  | 14        |

The results are aggregated in a trace file for analysis and performance evaluation. (Figure 7) shows snapshot portion of the scenario implementing the 50-node network topology. The figure reveals important details such as: wireless network options, node configuration, protocol parameters and connection options between the source and the destination.

Now, the number of nodes has been increased to 100, 150, and then 200. These nodes were randomly distributed within the same workspace. For each set of nodes the same steps explained before have been repeated for each Cartesian routing algorithm. Finally, the implementation results were gathered for all scenarios in trace files and analyzed using (Awk file) for the purpose of performance evaluation.
5.4. Evaluation performance

In this paper, the performance of Cartesian routing has been examined and evaluated for UAV networks implementing many algorithms. To achieve this, two metrics have been taken into account: packet delivery ratio and end-to-end delay. The metrics of interest introduced to evaluate the performance of the Cartesian routing protocol are described in this work and are defined as follow:

Packet Delivery Ratio (PDR) is the ratio of the number of packets received by the destination(s) to the number of packets sent by the source(s). While, End-to-End Delay is the average time taken from sending a packet from a source node until it reaches its destination node. It includes all possible delays generated by queuing at the interface queue, retransmission delays, propagation delay, transfer times of data and packets buffering during route discovery process.

For more clarification, the results have been plotted as shown in (Figure 8). It is noticed that there is a common behaviour response of all algorithms at the rate of packet delivery. As the network size increases, the delivery rate decreases until it reaches (150-node), then the rate starts to increase again when the network size become (200-node). This is clear in both Greedy and Ellipsoid algorithms, where it is less than 50%. In Compass and Most forward algorithms the rate is less than 60%. This indicates that in (150-node) scenario there is a high probability of falling into the local minimum. In spite of this, it is clear that the lower path dilation is close to (1 or 2). As path dilation indicates the number of hops the packet takes to reaches destination, as shown in (Figure 9). Good results of the delivery rate appear in PAB3D and G-PAB3D-G algorithms that reaches to above 90% especially in the case of (50-nodes). A slight increase in path dilation is negligible if we compare it with the previous case in terms of delivery ratio. This can be justified as the small number of hops in the G-PAB3D-G algorithm due to hybridization with the greedy algorithm.

In general, the best results have been obtained in the delivery rate in (Projective Face, CFace (3), and ALSP Face) algorithms, because these algorithms are more capable of finding the destination node than others. More precisely, in CFace(3) algorithm, which consists of (xy, xz, and yz) planes for
projection of the nodes, there will be three attempts to find the destination instead of two as is the case with the Projective Face algorithm, so the delivery rate is slightly higher. But, these good results in the delivery rate are offset by results that may be the worst in the path dilation shown in (Figure 9). For GFG algorithm, it is better than ALSP algorithm in terms of delivery rate as well as the path dilation due to the mechanism it follows in its work. Initially, the packets are delivered to the destination via greedy algorithm. If it stuck a local minimal, it turns to Face algorithm to deliver the packet, but on a smaller search scope, after that it returns to the Greedy algorithm. Finally, if we compare (LAR and LARPAB3D) with other algorithms, we notice that they do not give very high results in packet delivery rate. However, little progress has been noticed for LAR over PAB3DLAR in the delivery rate. But in the path dilation, PAB3DLAR needs half the number of hops that LAR needs to deliver the packet.

Figure 8. Packet delivery ratio of all algorithms, for different network sizes, and with TT LR = N, TT LF = 2N and TT L = 6N.

Figure 9. Path Dilation of all algorithms, for different network sizes [20]
In order to explore the comparison of Cartesian routing algorithms, another metric has been selected that is the end-to-end delay. This delay is calculated from the packets leave the source node till they reach the destination node. Now, the results have been plotted as shown in (Figure 10) after implementing all algorithms. It is noticed at first glance that the time it takes for the packet to reach from the source to the destination does not exceed (2 msec) in the scenario of (50-nodes) and in all algorithms except for the algorithms (Projective Face, CFace (3), and ALSP Face). Then it reaches (2.5 msec) in the scenario of (100-nodes) and remains approximately the same time or with a slight difference that is negligible in scenarios (150-nodes and 200-nodes).

![End-to-End Delay of all algorithms, for different network sizes](image)

**Figure 10.** End-to-End Delay of all algorithms, for different network sizes

In the algorithms (Projective Face, CFace (3), and ALSP Face) the packet arrival time from the source to the destination is (2.5 msec) in the (50-nodes) scenario and reaches twice as much in the (100-nodes) scenario. It continues to increase in (150-nodes) scenario which is the worst case while fall below this limit in the (200-nodes) scenario. The reason for the rise in the arrival time of the packet to the destination in (Projective Face and CFace (3)) algorithms is the large number of hops that the package passes through in order to progress towards the target. This is clearly shown in (Figure 10), where these two algorithms use threshold value (TTLF) to prevent loop occurrence in the network which in turn is used twice in Projective Face algorithm, once for the first plane and the second for the vertical plane. The total threshold value is (TTL=2 * TTLF). While, it is used 3 times in CFace (3) algorithm, once for each plane (xy, xz and yz) so the threshold value will be (TTL= 3 * TTLF). As for the ALSP algorithm, the packet arrival time at the target is the highest among its peers and is considered the worst. The delay in this ALSP algorithm is due to its use of a set of projection planes and switches between them to prevent a loop in the network and accompanies each transition process. Setting the threshold value (TTLF) for each plane leads to an increase in the number of hops that the packet takes to reach its destination.

### 6. Conclusions and Future work

In this study, the most prominent position-based routing algorithms implemented in three-dimensional networks were discussed, and a comprehensive description of the mechanism of work and what restrictions would cause obstruction was provided. Progress-based forwarding algorithms have demonstrated their efficiency in dense networks, but this performance is declining in dispersed networks due to falling into the local minimum. While face-based forwarding algorithms proved to be efficient in all networks, but at the expense of high path dilation and delay. PAB3D-CFace (1) - PAB3D and PAB3D-CFace (3) are considered the ideal choice for real-time applications because of their efficiency and performance. Finally, the study of energy in UAV networks and the impact of
mobility on the performance of position-based routing protocols are among the topics to be addressed in the future.

References

[1] S. Chandrasekharan et al., “Designing and implementing future aerial communication networks,” IEEE Commun. Mag., vol. 54, no. 5, pp. 26–34, 2016, doi: 10.1109/MCOM.2016.7470932.

[2] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, “Flying Ad-Hoc Networks (FANETs): A survey,” Ad Hoc Networks, vol. 11, no. 3, pp. 1254–1270, 2013, doi: 10.1016/j.adhoc.2012.12.004.

[3] O. K. Sahingoz, “Networking models in flying ad-hoc networks (FANETs): Concepts and challenges,” J. Intell. Robot. Syst. Theory Appl., vol. 74, no. 1–2, pp. 513–527, 2014, doi: 10.1007/s10846-013-9959-7.

[4] M. Mozaﬀari, W. Saad, M. Bennis, Y. H. Nam, and M. Debbah, “A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems,” IEEE Commun. Surv. Tutorials, vol. 21, no. 3, pp. 2334–2360, 2019, doi: 10.1109/COMST.2019.2902862.

[5] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” IEEE Commun. Mag., vol. 54, no. 5, pp. 36–42, 2016, doi: 10.1109/MCOM.2016.7470933.

[6] A. F. Molisch et al., “Modeling Air-to-Ground Path Loss for Low Altitude Platforms in Urban Environments Modeling Air-to-Ground Path Loss for Low Altitude Platforms in Urban Environments,” IEEE Trans. Veh. Technol., vol. 66, no. 1, pp. 632–636, 2016, doi: 10.1109/TVT.2015.2414819.

[7] L. Hughes, O. Banyasad, and E. Hughes, “Cartesian routing,” Comput. Networks, vol. 34, no. 3, pp. 455–466, 2000, doi: 10.1016/S1389-1286(00)00130-4.

[8] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for wireless networks,” Proc. Annu. Int. Conf. Mob. Comput. Networking, MOBICOM, pp. 243–254, 2000.

[9] A. E. Abdallah, T. Fevens, and J. Opatrny, “Randomized 3D position-based routing algorithms for ad-hoc networks,” 2006 3rd Annu. Int. Conf. Mob. Ubiquitous Syst. Netw. Serv. MobiQuitous, 2006, doi: 10.1109/MOBIQ.2006.340457.

[10] R. Flury and R. Wattenhofer, “Randomized 3D geographic routing,” Proc. - IEEE INFOCOM, pp. 1508–1516, 2008, doi: 10.1109/INFOCOM.2007.135.

[11] C. Liu and J. Wu, “Efficient geometric routing in three dimensional ad hoc networks,” Proc. - IEEE INFOCOM, pp. 2751–2755, 2009, doi: 10.1109/INFOCOM.2009.5062225.

[12] L. F. M. Vieira and A. V. dos S. Cunha, “Performance of Greedy Forwarding in Geographic Routing for the Internet of Drones,” Internet Technol. Lett., vol. 1, no. 5, pp. e47, 2018, doi: 10.1002/itl2.47.

[13] S. Rosati, K. Kruzelecki, G. Heitz, D. Floreano, and B. Rimoldi, “Dynamic Routing for Flying Ad Hoc Networks,” IEEE Trans. Veh. Technol., vol. 65, no. 3, pp. 1690–1700, 2016, doi: 10.1109/TVT.2015.2414819.

[14] H. Yang and Z. Liu, “An optimization routing protocol for FANETs,” Eurasip J. Wirel. Commun. Netw., vol. 2019, no. 1, 2019, doi: 10.1186/s13638-019-1442-0.

[15] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, “Routing in flying Ad Hoc networks: Survey, constraints, and future challenge perspectives,” IEEE Access, vol. 7, pp. 81057–81105, 2019, doi: 10.1109/ACCESS.2019.2923840.

[16] G. Gankhuyag, A. P. Shrestha, and S. J. Yoo, “Robust and Reliable Predictive Routing Strategy for Flying Ad-Hoc Networks,” IEEE Access, vol. 5, no. c, pp. 1–11, 2017, doi: 10.1109/ACCESS.2017.2647817.

[17] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, “Routing in flying Ad Hoc networks: Survey, constraints, and future challenge perspectives,” IEEE Access, vol. 7, pp. 81057–81105, 2019, doi: 10.1109/ACCESS.2019.2923840.
[18] M. Y. Arafat and S. Moh, “A Survey on Cluster-Based Routing Protocols for Unmanned Aerial Vehicle Networks,” *IEEE Access*, vol. 7, pp. 498–516, 2019, doi: 10.1109/ACCESS.2018.2885539.

[19] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, “Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols,” 2017 1st Int. Conf. Latest Trends Electr. Eng. Comput. Technol. INTELLECT 2017, vol. 2018-Janua, pp. 1–9, 2018, doi: 10.1109/INTELLECT.2017.8277614.

[20] A. Bujari, C. E. Palazzi, and D. Ronzani, “A Comparison of Stateless Position-based Packet Routing Algorithms for FANETs,” *IEEE Trans. Mob. Comput.*, vol. 17, no. 11, pp. 2468–2482, 2018, doi: 10.1109/TMC.2018.2811490.

[21] N. K. Gupta, R. S. Yadav, and R. K. Nagaria, “3D geographical routing protocols in wireless ad hoc and sensor networks: an overview,” *Wirel. Networks*, vol. 0123456789, 2019, doi: 10.1007/s11276-019-01983-y.

[22] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, and M. B. Yagoubi, “A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs),” *Veh. Commun.*, vol. 10, no. October, pp. 29–56, 2017, doi: 10.1016/j.vehcom.2017.10.003

[23] T. Narten, “Routing and Addressing Problem Statement,” no. draft-narten-radir-problem-statement-02, 2008.

[24] E. Kranakis, H. Singh, and J. Urrutia, “Proc. in 11 Th Canadian Conference on Computational Geometry,” *Compass routing Geom. networks*, pp. 1–4, 1999.

[25] H. Takagi and L. Kleinrock, “Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals,” *IEEE Trans. Commun.*, vol. 32, no. 3, pp. 246–257, 1984, doi: 10.1109/TCOM.1984.1096061.

[26] K. Yamazaki and K. Sezaki, “A Proposal of Geographical Routing Protocols for Location-Aware Services,” *Electron. Commun. Japan, Part I Commun. (English Transl. Denshi Tsushin Gakkai Ronbunshi)*, vol. 87, no. 4, pp. 26–34, 2004, doi: 10.1002/ecja.10138.

[27] P. Bose and P. Morin, “Online routing in triangulations?,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1741, pp. 88–91, 2005, doi: 10.1007/3-540-46632-0_12.

[28] T. Fevens, I. T. Haque, and L. Narayanan, “Randomized routing algorithms in mobile ad hoc networks,” *IFIP Adv. Inf. Commun. Technol.*, vol. 162, pp. 347–357, 2005.

[29] G. Kao, T. Fevens, and J. Opatrný, “Position-based routing on 3-D geometric graphs in mobile Ad Hoc networks,” *Proc. 17th Can. Conf. Comput. Geom. CCCG 2005*, pp. 88–91, 2005.

[30] P. Bose, P. Morin, I. Stojmenovi, and J. Urrutia, “Routing with Guaranteed Delivery in MANET 2001,” vol. 0, pp. 1–9, 2001.

[31] G. Kao, T. Fevens, and J. Opatrný, “3-D localized position-based routing with nearly certain delivery in mobile ad hoc networks,” *2007 2nd Int. Symp. Wirel. Pervasive Comput.*, pp. 344–349, 2007, doi: 10.1109/ISWPC.2007.342627.

[32] A. E. Abdallah, “Hybrid Position-Based 3 D Routing Algorithms,” no. May, pp. 227–230, 2006.

[33] A. K. Nayak, S. C. Rai, and R. Mall, *Introduction to NS2*. 2016

**BIOGRAPHIES OF AUTHORS**

Karam mheidy Al-Sofy
Received the B.S. degree in Computer and Information Engineering from Mosul University, Mosul, Iraq in 2006. He is currently working toward the Master’s degree Computer and Information Engineering, Electronics Engineering, Ninevah University, Mosul, Iraq. His current research interests include Cartesian routing and UAV Networks. He can be contacted at eng.k.alsofy@uomosul.edu.iq
Sahar Abdul Aziz Al-Talib received her BSc. in Electrical and Communication Engineering from University of Mosul in 1976, her MSc, and PhD in Computer and Communications Engineering from University Putra Malaysia (UPM), in 2003 and 2006 respectively. She served at MIMOS Berhad, Technology Park Malaysia, for about 4 years as Staff Researcher where she enrolled in infrastructure mesh network project (P000091) adopting IEEE 802.16/e/d wireless technology. She became a lecturer at the Department of Computer and Information Engineering, Electronics Engineering, University of Mosul, Iraq in 2011. She was the main inventor of 10 patents and published more than 20 papers in local, regional, and international journals and conferences. She certified CISCO-CCNA, WiMAX RF Network Engineer, WiMAX CORE Network Engineer, IBM “Basic Hardware Training Course”- UK, Gii-Honeywell Bull “Software Training Course on Time Sharing System and Communications Concept”- France, Six-Sigma white belt, Harvard Blended Leadership Program by Harvard Business School Publishing – USA, Protocol Design and Development for Mesh Networks by ComNets – Germany. She can be contacted at sahar.altalib@uoninevah.edu.iq