J Ambient Intell Human Comput (2018) 9:85–94
DOI 10.1007/s12652-017-0528-y

ORIGINAL RESEARCH

Low overhead hybrid geographic-based routing algorithms with smart partial flooding for 3D ad hoc networks

Alaa E. Abdallah[1]

Received: 19 February 2017 / Accepted: 17 April 2017 / Published online: 6 July 2017

Abstract One known design for routing algorithms in mobile ad hoc networks is to use flooding. But, these algorithms are usually suffer from high overhead. Another common design is to use the nodes geographic locations to take routing choices. Current geographical routing algorithms usually address the routing environment in 2D space. However, in real life, nodes could be located in 3D space. To benefit from the advantages of both techniques we propose several 3D new routing algorithms that maximize the packets delivery rate and minimize the overhead. Our first set of algorithms (SPF: smart partial flooding) uses the nodes location to do the flooding in the direction of the destination over a sub-graph of the original dense graph. The second set (Progress–SPF) uses geographical routing to progress as much as possible to the destination, if its not possible, SPF is used over a sub-graph extracted locally. The 3rd set (Progress–SPF–Progress) used geographical routing to progress to the destination, if the progress is not possible, SPF is used over a sub-graph for one step only and then the algorithm goes back to the geographical routing. We evaluate our algorithms and compare them with current routing algorithms. The simulation results show a significant improvement in delivery rate up to 100% compared to 70% and a huge reduction in overall traffic around 60%.

Keywords Flooding-base routing · Geographic-based routing · Localized routing · 3D ad hoc networks · Unit ball graph

1 Introduction

Wireless networks became an essential part of todays communication infrastructure. Such networks are usually modeled by mobile ad hoc networks (MANETs). MANETs consist of a set of wireless mobile hosts that can communicate with no physical backbone infrastructure. Communication is based on radio propagation; two nodes can communicate directly if they are within the transmission range of each other (Cadger et al. 2013; Huang et al. 2016). To communicate with nodes outside the transmission range, multihop routing is used employing the nodes in between to forward packets. Since mobile ad hoc networks may change their topology often and because of the resource constraints, routing in such networks is difficult. In the last couple of decades, several routing algorithms have been suggested to address the multihop routing problem; each is based on different assumptions and theories. These routing algorithms can be classified to two main basic types (Mauve et al. 2001): flooding-based routing (Basagni et al. 1998; Ko and Vaidya 2000; Johnson and Malts 1996) and progress-based routing (Fevens et al. 2005, 2008; Kao et al. 2005; Karp and Kung 2000). In flooding-based routing, when a packet reaches a node, it forwards that packet to all its neighbors. This can usually find the shortest path between two nodes. However, it can be difficult for these routing algorithms to work with large MANETs because of the huge traffic and overhead that can be created around the network. Geocasting based Location-Aided Routing (LAR) (Ko and Vaidya 2000) is an example for this strategy.

In flooding-based routing the whole network contribute in choosing the rouging path. In contrast, progress-based routing algorithms (also referred as geographic-based routing) use local information about some nodes positions to take routing decisions. The node having a packet forwards

✉ Alaa E. Abdallah
aabdallah@hu.edu.jo

[1] Faculty of Information Technology, Hashemite University, Zarqa 13115, Jordan

it to one of its neighbors according to some heuristic, Greedy (Chvatal 1979; Abdallah et al. 2006b) and Compass (Kranakis et al. 1999) routing algorithms are considered as progress-based routing algorithms (Kranakis et al. 1999; Fevens et al. 2005). Progress-based routing algorithms eliminate the high overhead by flooding-based routing, making them simple, efficient, and scalable for big ad hoc and sensor networks. The main weakness of these algorithms is the non-guarantee packet delivery, they may fail when a packet reaches a node which does not have a neighbor that can make progress to the destination even though the source and the destination are connected in the network. In general, a geographic-based routing algorithm has the following assumptions:

- Each mobile host in MANET knows its geometric coordinates. This can be obtained through a GPS receiver, or other such mechanisms (Capkun et al. 2002; Kaplan 1996).
- Each mobile host has the means to determine the geometric coordinates of every other mobile host within its transmission range, this can be optioned using a periodical broadcasted hello messages where information about the mobile host like geographical coordinates are usually embedded in these messages.
- The source node knows the geometric location of the packet destination, which can be obtained by location services (Hightower 2001; Li et al. 2000), or by receiving the location from a previous packet from that node. The source node adds this location to the packet header. Once a node receives a packet, it can learn the location of the destination.
- There is no need for any node to store routing tables. Nodes only maintain the information about their neighbors for a fixed number of hops (usually one hop) away.

Current routing algorithms usually address the routing environment in 2D space (Cadger et al. 2013). However, in real life, nodes could be located in 3D space (Abdallah and Otoom 2016; Abdallah 2016). 3D ad hoc networks can be find in many situations (Wang et al. 2012; Sheltami and Menshawi 2016; Silva and Analide 2017), for example: (1) Ocean Sensor Networks: a set of sensors positioned at different locations and different depth, the sensors could be used to sense the pressure, oil leaks or temperature. If the sensors aware of their 3D positions and can communicate with each other, then they need routing algorithms designed for the communication. (2) Disaster Recovery Operations: consider a set of firefighters in a large high rise building, and they are located at different floors levels. The firefighters should have voice communication between each other to coordinate mission activities. Thus a 3D ad-hoc network with a reliable routing algorithms is needed. There are many other scenario for 3D ad hoc networks such that city landscape, and hilly terrain, airborne.

In this paper, we propose several 3D geographical routing algorithms that uses the advantage of the high delivery rate of the flooding-based algorithms and the low overhead of the progress-based routing algorithms. The first set of algorithms, (SPF: smart partial flooding), decreases the overhead of flooding-based routing algorithms by using local algorithms to extract connected sparse sub-graphs from the original dense graph. To prevent packets from going far from the destination, we have added more restriction on SPF by hybridized it with 3DLAR. The second set, (Progress–SPF) uses geographical routing to progress as much as possible to the destination, if the packet reaches a node where the progress is not possible, this node is used as a source and SPF starts. The third set, (Progress–SPF–Progress) uses geographical routing to progress to the destination, if the progress is not possible, a SPF is used over a sub-graph for one step only and then the algorithms go back to the geographical routing.

The rest of the paper is organized as follows. In Sect. 2, we define the network model and survey previous work. In Sect. 3 we give a detailed description of the new routing algorithms. We present simulation results to prove the much enhanced performance of the proposed methods in comparison with existing techniques in Sect. 4. Section 5 summarizes our results.

2 Preliminaries

2.1 Network model

Mobile ad hoc networks are usually modeled using geometric graphs that are embedded in a three-dimensional Euclidean space. These graphs vertices are points with coordinates and the edges are straight-line segments. The set of n wireless hosts is represented as a point set S in $\mathbb{R}^3$, each point possessing a geometric location. All network hosts have the same communication range of r, which represented as a sphere volume of radius r. We define the $dist(u, v)$ as the Euclidean distance between the points u and v, $dist(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}$. Two nodes are connected by an edge (undirected) if the Euclidean distance between them is at most r. The resulting graph is called a unit ball graph (UBG). For node u, we denote the set of its neighbors by $N(u)$. Define a sub-graph of G, $P(G)$, as a t-spanner of G if the length of the shortest path between any two nodes in $P(G)$ is not more than t times longer than the shortest path connecting them in G, where t is the stretch factor. The length of the path is the sum of the lengths of the edges along the path.

Many routing algorithms use a spanning sub-graph of the unit ball graph such that only the edges in the sub-graph are used for routing. Since the wireless hosts that we are modeling are commonly powered by a limited power supply like a battery as well as containing a limited amount of memory, may be mobile, and the topology of the whole network is usually not available and may be variable, localized algorithms are typically preferred. A distributed algorithm is called local if each node of the network only uses information obtained uniquely from the nodes located no more than a constant (independent of the size of the network) number of hops from it. Thus, during the algorithm, no node is ever aware of the existence of other nodes in the network further away than this constant number of hops. In the following we present some local distributed algorithm to extract a sub-graph from UBG.

- For a geometric graph G, 3D Gabriel Graph $3DGG(G)$ (Gabriel and Sokal 1969; Kao et al. 2005) is defined as follows: let $Sphere(u, v)$ be defined as the ball centered at the midpoint between the points u and v with a diameter $|uv|$. For any two adjacent nodes u and v in G, the edge (u, v) belongs to $3DGG(G)$ if, and only if, no other node $w \in G$ is located in the $Sphere(u, v)$, see Fig. 1a. Its proved that $3DGG(G)$ is connected and a $\left(4\pi\sqrt{2n-4/3}\right)$-spanner of G.
- For a geometric graph G, 3D Relative Neighborhood Graph $3DRNG\ (G)$ (Jaromczyk and Toussaint 1992; Toussaint 1980) is defined as follows: an edge (u, v) exists in $3DRNG(G)$ between the points u and v in G if no other point w in G is inside the lens formed by the intersection of the two spheres centered on u and v with radius equal to the transmission range of each node, see Fig. 1b. In other words, an edge $uv \in RNG(G)$ if, and only if, there is no node w such that $(u, w) \leq (u, v) and (v, w) \leq (u, v)$. It is proved that $RNG(G)$ is connected and a $(n-1)$-spanner of G.

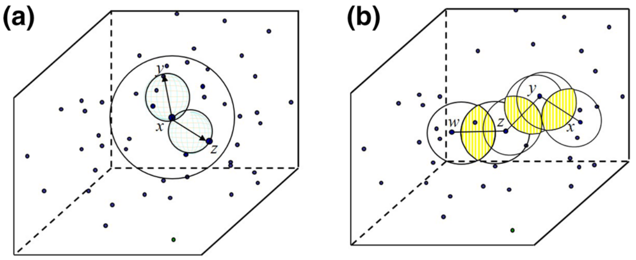

Fig. 1 **a** $3DGG(G)$: the edge xz considered in $3DGG(G)$ because the sphere contains xz does not have any other node, while the edge xy is not in $3DGG(G)$ because there are other nodes in the sphere xy. **b** $3DRNG\ (G)$: edges xy and yz are in $3DRNG\ (G)$ because their spheres intersection does not contains any other points. The edge wz is not in $3DRNG\ (G)$

2.2 Related routing algorithms

The routing task is to find a path from the source node s to the destination node d. Local information is used to determine how to route the packet. We are interested in the following performance measures for routing algorithms: *Delivery rate* the percentage of times that the algorithm succeeds in delivering its packet. *Path dilation* the average ratio of the length of the path returned by the algorithm to the length of the shortest path in the UBG. *Overhead* the average ratio of number of nodes participate in the routing process to the number of hops in the shortest path.

In this subsection we review some representative geographical routing algorithms that are closely related to the new proposed algorithms: (1) 3DGreedy routing (Abdallah et al. 2006a): a node forwards the packet to its neighbor that minimizes the remaining distance to the destination. This is repeated until the destination node is reached. In many cases this routing algorithm may fail to deliver the packet to the destination when a local minimum node is reached (a node that has no neighbor closer to the destination). (2) 3DCompass routing (Abdallah et al. 2006b): a node forwards the packet to its neighbor that minimizes the angle formed between the current node, next node, and destination. This is repeated until the destination node is reached. As in 3DGreedy routing, this algorithm may suffer from the local minimum when the nodes enter an infinite loop of sending to each other without making progress to the destination. (3) 3DLAR (Ko and Vaidya 2000; Abdallah et al. 2006a): this algorithm also uses the position information of nodes to restrict the flooding process during the route discovery phase. The node holding the packet forwards it to all neighbors that are located in a rectangular box as follows: (a) with the available information of the destination node d, the source node computes the expected zone for the destination, its a sphere centered at d with radius equal to $v \times (t_1 - t_0)$, where t_1 is the current time, t_0 is the last time stamp known about the destination position and v is the maximum speed known about the destination node. (b) The rectangular box longest diagonal starts at the source node and ends at the surface of the expected zone. Because 3DLAR relies on flooding, the delivery rate usually is very high, but it suffers from the huge overhead and traffic in the network.

Algorithm 1: Directional-SPF over *3DGG(G)* graph

```
// Consider $s, d, c$ are the source, the destination node, and the current node
respectively
// Execution starts when a source node needs to route a packet to some other node.
begin
    $S$ calculates the 3DLAR rectangular box parameters and append them to the
    packets
    $c$ applies $3DGG(G)$ graph algorithm on its neighbors
    Let $Q$ equal the set of neighbors around $c$ from $3DGG(G)$ graph
    for $i \leftarrow 1$ to number of nodes in $Q$ do
        if $i$ inside 3DLAR rectangular box then
            send the packet through edge $\overrightarrow{ci}$
            if $i = d$ then                              /* $d$ has been reached */
                do nothing
            else if $i$ has seen the packet before then
                do nothing
            else                                  /* Recursively call Directional-SPF */
                $c \leftarrow i$
                call Algorithm 1
            end
        end
    end
end
```

3 Proposed routing algorithms

Although 3DGreedy And 3DCompass algorithms delivery rate is relatively low, they usually find a path very close to the shortest path. Thus, the over-head and traffic are very low. On the other hand, flooding-based routing algorithm (3DLAR) can almost provide a nearly guaranteed delivery rate but with huge traffic because there are many nodes receive and forward packets without being part of the path from the source to the destination.

In what follows, we describe several new 3D routing algorithms that attempt to decrease the overhead and increase the delivery rate over regular flooding algorithms and progress-based routing algorithms:

1. Smart partial flooding (SPF): when a node receives a packet, it extracts locally the neighbors that belongs to 3*DGG*(*G*) and forward that packet to all extracted neighbors. This is repeated until the destination node is reached. To stop looping, when a node sees the same packet for the second time, it just discards that packet.
2. Directional-SPF: this algorithm adds more restriction on SPF flooding process, directional-SPF uses a rectangular box similar to the box used in 3DLAR to prevent packets from going too far and stay in the destination direction. The routing process works as follows, the current node having a packet extracts the set of neighbors in 3*DGG*(*G*) and forward its packets to each neighbor if its location is inside the 3DLAR box. See Algorithm 1 and Fig. 2.
3. Progress–SPF: this algorithm is a general approach that hybridize progress-based routing algorithms with SPF routing algorithms to benefit from both algorithms. The algorithm starts by using 3DGreedy algorithm as much as possible, if the packet reaches a local minimum node, this node adds a special character to the packet to tell nodes in the path to start extracting locally the 3*DGG*(*G*) graph and to send the packet to all neighbors in 3*DGG*(*G*), see Fig. 3. Progress–SPF algorithm definitely guarantees the delivery of packets,

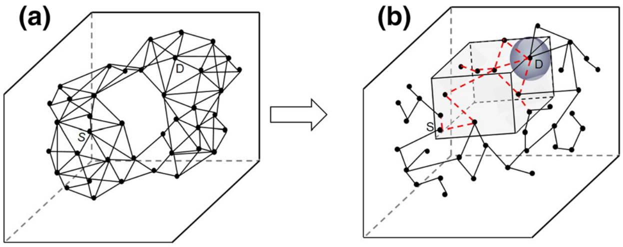

Fig. 2 **a** Node *S* has a set of packets and wants to send them to node *D*, the background network is the *UBG*. **b** Packets are sent only to the nodes that belongs to *3DGG*(*G*) and inside the directional box

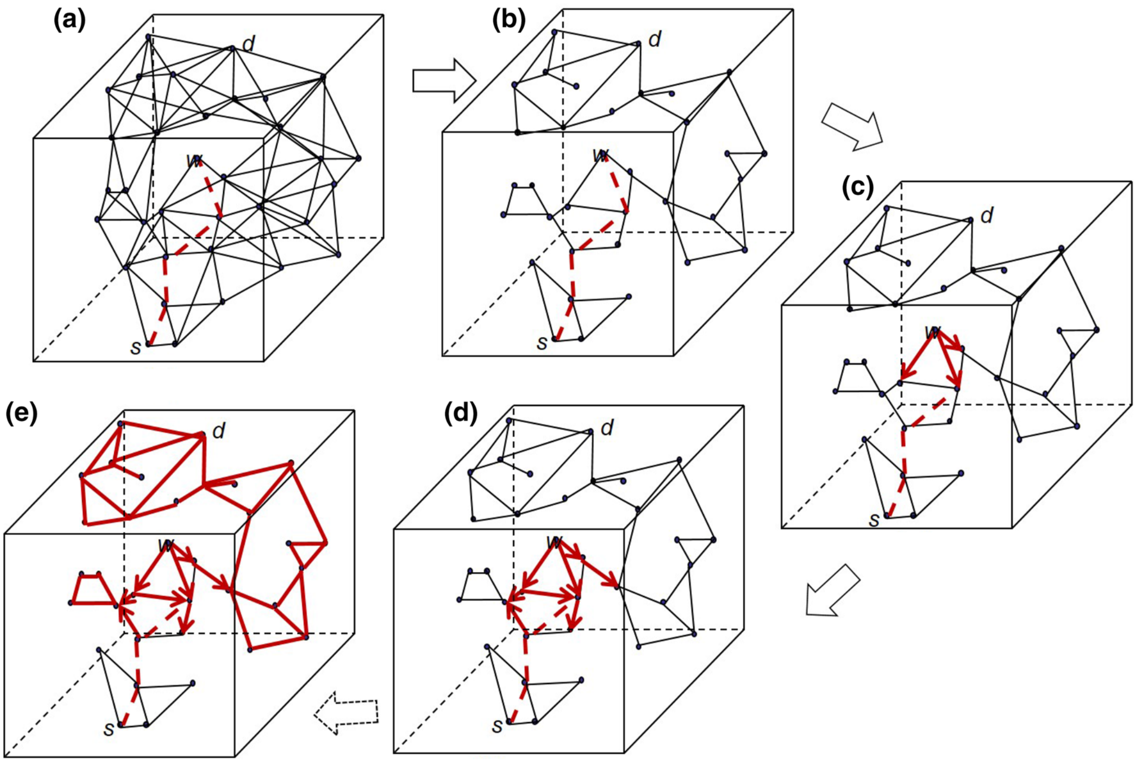

Fig. 3 Progress–SPF. **a** The algorithm uses 3DGreedy forwarding as much as possible. **b** At the local minimum, each node starts generating $3DGG(G)$ graph locally. **c**–**e** Flooding is done over the sub-graph

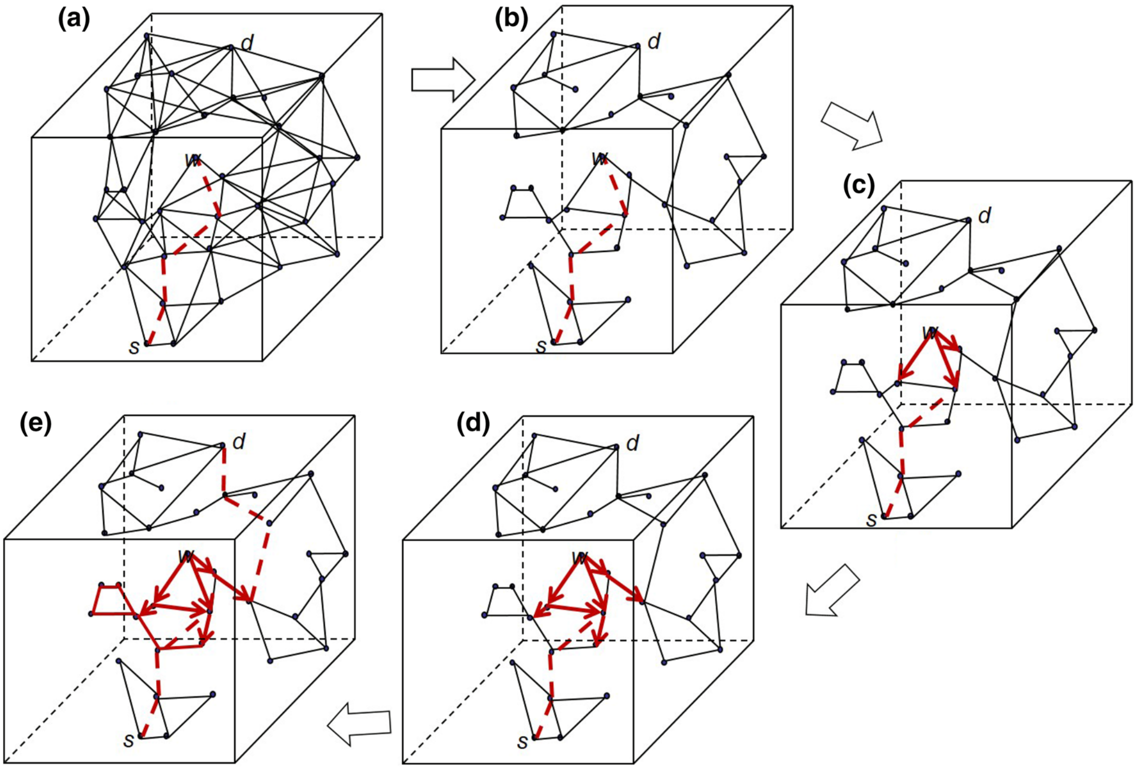

Fig. 4 Progress–SPF–Progress. **a** The algorithm uses 3DGreedy forwarding as much as possible. **b** At the local minimum, each node starts generating $3DGG(G)$ graph locally. **c** SPF is done for a couple of steps until we reach a node closer to the destination. **d**, **e** 3DGreedy is used again

and the traffic has dropped dramatically compared to regular flooding-based routing

4. Progress-directional-SPF: this is another hybrid algorithm that combine progress-based routing with directional-SPF. The algorithm tries to progress using 3DGreedy algorithm as much as possible. If there is no delivery due to the local minimum, the algorithm switches to directional-SPF. See Algorithm 2 for details.
5. Progress–SPF–Progress: although Progress–SPF and Progress–directional-SPF, have excellent deliver rate, but the traffic and overhead still can be improved. Our fifth hybrid algorithm is (Progress–SPF–Progress), starts by 3DGreedy algorithm as much as possible, if the packet reaches a local minimum node (u). It runs the $3DGG(G)$ and forward the packet to all neighbors of u that are belong to $3DGG(G)$, see Fig. 4. The algorithm goes back to 3DGreedy routing right after that. See Algorithm 3 for more details.

4 Performance evaluation of routing algorithms

In the simulation experiments, a (100, 150, 200, 250, 300) nodes are randomly positioned in a cube of side length equal to 200m. The transmission range is fixed on 25m. The reason behind these numbers is to satisfy the simulation environment suggested in (Kuhn et al. 2003). According to their study, the number of mobile nodes per unit disk graph should be around 5, which would match to graphs with average node degrees of around 4. Through our study, graphs with $n = 150$ would most closely match the required node degrees. If the resulted network graph is not connected, we calculate the largest connected component (LCC), and then we use it to perform the studied routing algorithms. The source and the destination are both chosen randomly from the LCC. To estimate the performance, we execute each algorithm on 100 random graphs and the percentage of successful delivers determined. To compute the average deliver rate, this process

Algorithm 2: Progress-Directional-SPF

```
// Consider $s, d, c$ are the source, the destination node, and the current node
respectively
// Execution starts when a source node needs to route a packet to some other node.
begin
    $c \leftarrow s$                    /* At the beginning source is the current node  */
    repeat
        Let $L(c)$ is the set of neighbors nodes around $c$
        for each node $x$ in $L(c)$ do
            find the Euclidean distance for each edge $dist(\overrightarrow{xd})$
        let the neighbor $m$ has the shortest distance out of neighbors $dist(\overrightarrow{md})$
        if $dist(\overrightarrow{md}) \leq dist(\overrightarrow{cd})$ then              /* Can progress to $d$ */
            $c$ forward the packet to the node $m$
            $c \leftarrow m$
        else                                          /* Local minimum phenomena */
            break
    until $c = d$
    if $c = d$ then                   /* algorithm succeed to deliver the packet */
        return
    else                                      /* Continue with partial flooding */
        $c$ calculates the 3DLAR rectangular box parameters and append them to
        the packets
        $c$ applies $3DGG(G)$ graph algorithm on its neighbors
        Let $Q$ equal the set of neighbors around $c$ from $3DGG(G)$ graph
        for $i \leftarrow 1$ to number of nodes in $Q$ do
            if $i$ inside 3DLAR rectangular box then
                send the packet through edge $\overrightarrow{ci}$
                if $i = d$ then                           /* $d$ has been reached */
                    do nothing
                else if $i$ has seen the packet before then
                    do nothing
                else                        /* Recursively call Directional-SPF */
                    $c \leftarrow i$
                    call Algorithm 1
```

is repeated 100 times and the average is taken. Moreover, out of the 10,000 runs used to compute the average packet delivery rate, the overhead which measured as the number of packets created and exchanged in the network by a routing algorithm is computed. The path dilation also computed on the same run.

4.1 Performance of the new algorithms

We present a comparison between the new proposed routing algorithms and other related algorithms in terms of delivery rate in Table 1. It can be seen that 3DLAR and all new routing algorithms have 100% delivery rate. We can explain the result of each algorithm as follows: 3DGeedy and 3DCompass routing have relatively low delivery rate because of the local minimum problem. As the number of nodes increases, the devilry rate reaches 100%. This is because dense graphs have many more paths to the destination and the local minimum nodes in the way almost disappear. 3DLAR uses flooding, thus, packets reach the destination eventually. If the correct path is outside the rectangular box, 3DLAR increases the size of the expected range and try again until the packet reaches the destination. SPF and Directional-SPF are flooding algorithms over connected sub-graph of UBG so delivery is guaranteed as well. Because Progress–SPF and Progress-directional–SPF use 3DGreedy, then around 70% of the graphs success in packet delivery. For the failed cases, Progress–SPF, uses flooding at the local minimum point to guarantee packets delivery. Progress–SPF–Progress also uses 3DGreedy, to overcome local minimum it uses flooding until it reaches a node closer to the destination, then it goes back to 3DGreedy, thus it has 100% delivery rate.

Algorithm 3: Progress-SPF-Progress

```
// Consider $s, d, c$ are the source, the destination node, and the current node
respectively
// Execution starts when a source node needs to route a packet to some other node.
begin
    $c \leftarrow s$                    /* At the beginning source is the current node */

    repeat
        Let $L(c)$ is the set of neighbors nodes around $c$
        for each node $x$ in $L(c)$ do
            find the euclidean distance for each edge $dist(\overrightarrow{xd})$

        let the neighbor $m$ has the shortest distance out of neighbors $dist(\overrightarrow{md})$
        if $dist(\overrightarrow{md}) \leq dist(\overrightarrow{cd})$ then            /* Can progress to $d$ */
            $c$ forward the packet to the node $m$
            $c \leftarrow m$
        else                    /* No more progress to $d$ (Local minimum) */
            break
    until $c = d$
    if $c = d$ then                  /* algorithm succeed to deliver the packet */
        return
    else                                          /* Flooding is needed */
        $c$ applies $3DGG(G)$ graph algorithm on its neighbors
        Let $Q$ equal the set of neighbors around $c$ from $3DGG(G)$ graph
        for $i \leftarrow 1$ to number of neighbors in $Q$ do
            send the packet through edge $\overrightarrow{ci}$

        if $i = d$ then                              /* $d$ has been reached */
            do nothing
        else if $i$ has seen the packet before then
            do nothing
        else                                         /* Go back to 3DGreedy */
            $c \leftarrow i$
            call Algorithm 3
```

Table 1 Average delivery rate

Algorithms	n = 100	n = 150	n = 200	n = 250	n = 300
3DGreedy	60	71	89	100	100
3DCompass	59	70	90	99	100
3DLAR	100	100	100	100	100
SPF	100	100	100	100	100
Directional-SPF	100	100	100	100	100
Progress–SPF	100	100	100	100	100
Progress–directional-SPF	100	100	100	100	100
Progress–SPF–Progress	100	100	100	100	100

Table 2 Average path dilation

Algorithms	n = 100	n = 150	n = 200	n = 250	n = 300
3DGreedy	1.33	1.27	1.19	1.12	1.06
3DCompass	1.35	1.26	1.16	1.08	1.02
3DLAR	1.12	1.03	1	1	1
SPF	1.28	1.22	1.13	1.08	1.06
Directional-SPF	1.42	1.37	1.28	1.23	1.12
Progress–SPF	1.27	1.22	1.08	1.12	1.06
Progress-directional-SPF	1.32	1.25	1.18	1.12	1.06
Progress–SPF–Progress	1.26	1.19	1.12	1.12	1.06

Table 3 Average overhead

Algorithms	n = 100	n = 150	n = 200	n = 250	n = 300
3DGreedy	1.33	1.27	1.19	1.12	1.06
3DCompass	1.35	1.26	1.16	1.08	1.02
3DLAR	3.5	5.25	9.5	12.55	15.33
SPF	2.3	3.25	5.33	7.88	9.65
Directional-SPF	2.1	2.81	3.65	4.78	6.5
Progress–SPF	1.88	1.85	1.55	1.12	1.06
Progress–directional-SPF	1.56	1.62	1.43	1.12	1.06
Progress–SPF–Progress	1.35	1.28	1.25	1.12	1.06

In Table 2, we show the path dilation of all studied algorithms. It is immediately evident that 3DLAR has the lowest path dilation, because it uses flooding and the shortest path will be chosen most of the time (if the shortest bath is inside the rectangular box). SPF definitely chooses the shortest bath in the extracted sub-graph, but it does not have to be the same path as the one in UBG thats why it comes second after 3DLAR. Directional-SPF has increases the path dilation a little over SPF, because the box added more restriction on the path. 3DGreedy and 3DCompass algorithm also have very low path dilation because they try to progress as much as possible in every step and this is the definition of the shortest path. Progress–SPF, and Progress–SPF–Progress are both a combination of both 3DGreedy and Flooding, thus they also have very low path dilation. In these two hybrid algorithms as the number of nodes increases 3DGreedy always succeed in the delivery, thats why the path dilation equal to 3DGreedy.

Table 3 shows the overall overhead of all algorithms. The results can be explained as follows: 3DGreedy and 3DCompass have almost no overhead because there is no flooding. The nodes that see the packets are actually part of the path between the source and the destination. 3DLAR, uses flooding all the time, which means so many nodes receive and forward each packet (all nodes in the flooding box). Thus the overhead is the highest between all studied algorithms. Even SPF uses flooding but it decreases 3DLARs overhead by about 40%, this is because SPF works over sub-graph which means less number of edges, thus less number of nodes see the packets. Still this algorithm has the second highest overhead. Directional-SPF decreases the overhead even more than SPF because it adds more restriction on the packet propagation using the flooding box. Progress–SPF decreases the high overhead of SPF and directional-SPF, because at least in 70% of the graphs have no flooding (3DGreedy success in delivery). For the other 30% flooding is used after progressing to the destination as much as possible. Progress–SPF–Progress has the lowest overhead between all new algorithms, because it uses flooding for a couple of steps only and it goes back to 3DGreedy (it benefits from the low overhead of 3DGreedy algorithm more than Progress–SPF and Progress–directional-SPF).

In Fig. 5, we show the effect of the background subgraph on most of the new proposed routing algorithm. We can see in Fig. 5a that using $3DRNG(G)$ gives less overhead than using $3DGG(G)$, this is due to the fact that $3DRNG(G)$ has less number of edges than $3DGG(G)$. SPF uses flooding and less number of edges means less packets in the network and less overhead. Figure 5b has similar results because directional-SPF also based on flooding. Figure 5c, d show the results on the hybrid algorithms, Progress–SPF and Progress–SPF–Progress. $3DGG(G)$ gives better results because $3DRNG(G)$ has less number of edges, then there is more chances for the local minimum problem. Thus hybrid algorithms will switch to flooding more and the overhead increases.

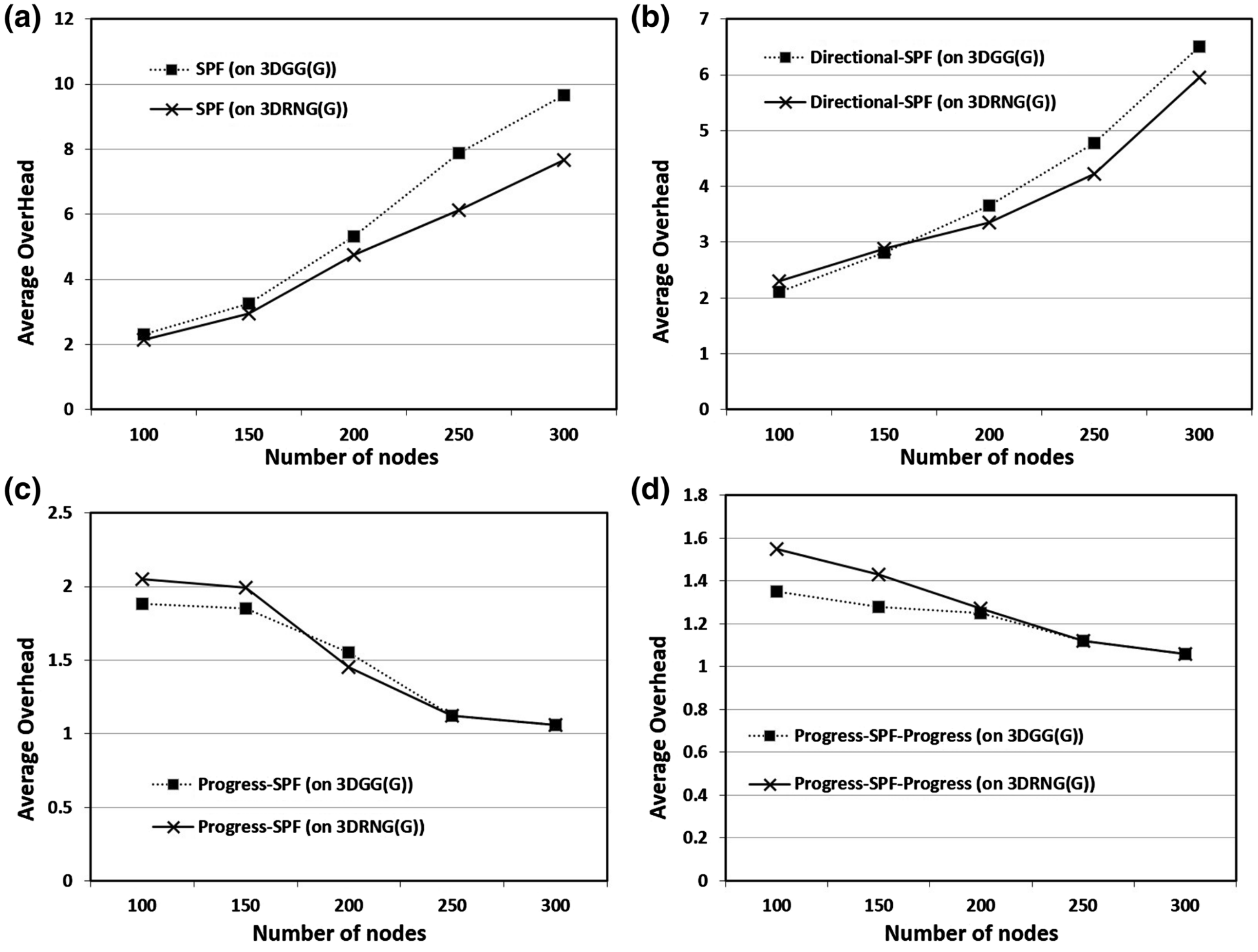

Fig. 5 Effect of the background sub-graph on the new proposed routing algorithm

5 Conclusions

In this paper, we present several new 3D routing algorithms for mobile ad hoc networks. The new algorithms are based on idea of benefiting from the low overhead of progress-based routing and the guaranteed delivery of flooding-based routing. All proposed algorithms performed over a sub-graph of UBG to decrease the overhead if flooding is needed. In the new hybrid algorithms we try to advance as much as possible using progress-based routing and when they fail; we use the new proposed smart partial flooding algorithms to guarantee the delivery. The simulation results show an excellent improvement in terms of delivery rate and low overhead. For future work we are planning to study the effect of other network layers on the new proposed algorithms.

References

Abdallah AE (2016) Smart partial flooding routing algorithms for 3D ad hoc networks. In: Proceedings of the the 11th international conference on future networks and communications, Montreal-Canada, pp 1–8

Abdallah A, Fevens T, Opatrny J (2006a) Hybrid position-based 3d routing algorithms with partial flooding. In: Proceedings of the Canadian conference on electrical and computer engineering, Ottawa, pp 1135–1138

Abdallah A, Fevens T, Opatrny J (2006b) Randomized 3-d position-based routing algorithm for ad-hoc networks. In: Proceedings of the 3rd annual international conference on mobile and ubiquitous systems: networks and services (MOBIQUITOUS), San Jose, pp 1–8

Abdallah AE, Abdallah E, Bsoul B, Otoom AF (2016) Randomized geographic-based routing with nearly guaranteed delivery for three-dimensional ad hoc network. Int J Distrib Sens Netw 12(10):1–11

Basagni S, Chlamtac I, Syrotiuk V, Woodward B (1998) A distance routing effect algorithm for mobility (DREAM). In: Proceedings of the 4th annual ACM/IEEE international conference on mobile computing and networking (MOBICOM), Dallas, pp 76–84

Cadger F, Curran K, Santos J, Moffett S (2013) A survey of geographical routing in wireless ad-hoc networks. IEEE Commun Surv Tutor 15(2):621–653

Capkun S, Hamdi M, Hubaux J-P (2002) GPS-free positioning in mobile ad-hoc networks. Clust Comput J 5(2):118–124

Chvatal V (1979) A greedy heuristic for the set covering problem. Math Oper Res 4:233–235

Fevens T, Abdallah A, Bennani B (2005) Randomized AB-face-AB routing algorithms in mobile ad hoc network. In: 4th international conference on ad-hoc networks and wireless, volume 3738 of LNCS, Mexico, pp 43–56

Fevens T, Liu S, Abdallah A (2008) Hybrid position-based routing algorithms for 3-d mobile ad hoc networks. In: Proceedings of the 4th international conference on mobile ad-hoc and sensor networks (MSN), China, pp 177–186

Gabriel K, Sokal R (1969) A new statistical approach to geographic variation analysis. Syst Zool 18(3):259–278

Hightower JGB (2001) Location systems for ubiquitous computing. Computer 34(8):57–66

Huang H, Yin H, Luo Y, Zhang X, Min G, Fan Q (2016) Three-dimensional geographic routing in wireless mobile ad hoc and sensor networks. IEEE Netw 30(2):82–90

Jaromczyk J, Toussaint G (1992) Relative neighborhood graphs and their relatives. Proc IEEE 80(9):1502–1517

Johnson D, Malts D (1996) Dynamic source routing in ad hoc wireless networks. In: Imielinski T, Korth H (eds) Mobile computing, vol 5. Kluwer Academic Publishers, Dordrecht, pp 153–181

Kao G, Fevens T, Opatrny J (2005) Position-based routing on 3D geometric graphs in mobile ad hoc networks. In: Proceedings of the 17th Canadian conference on computational geometry (CCCG'05), Windsor, pp 88–91

Kaplan E (1996) Understanding GPS. Artech House, Norwood

Karp B, Kung H (2000) GPSR: greedy perimeter stateless routing for wireless net-works. In: Proceedings of the 6th ACM/IEEE conference on mobile computing and networking (Mobicom 2000), Boston, pp 243–254

Ko Y, Vaidya N (2000) Location-aided routing (LAR) in mobile ad hoc networks. ACM Balt Wirel Netw 6(4):307–321

Kranakis E, Singh H, Urrutia J (1999) Compass routing on geometric networks. In: Proceedings of the 11th Canadian conference on computational geometry (CCCG '99), Vancouver, pp 51–54

Kuhn F, Wattenhofer R, Zollinger A (2003) Ad-hoc networks beyond unit disk graphs. In: Proceedings of the 2003 joint workshop on the foundation of mobile computing (DIALM-POMC), San Diego, pp 69–78

Li J, Jannotti J, De Couto D, Karger D, Morris R (2000) A scalable location service for geographic ad-hoc routing. In: Proceedings of the 6th ACM international conference on mobile computing and networking (MobiCom), Boston, pp 120–130

Mauve M, Widmer J, Hartenstein H (2001) A survey of position-based routing in mobile ad-hoc networks. IEEE Netw Mag 15(6):30–39

Silva F, Analide C (2017) Ubiquitous driving and community knowledge. J Ambient Intell Humaniz Comput 8(2):157–166

Sheltami TR, Shehryar Khan EMS, Menshawi MK (2016) Continuous objects detection and tracking in wireless sensor networks. J Ambient Intell Humaniz Comput 7(4):489–508

Toussaint G (1980) The relative neighbourhood graph of a finite planar set. Pattern Recogn 12(4):261–268

Wang Y, Liu Y, Guo Z (2012) Three-dimensional ocean sensor networks: a survey. J Ocean Univ China 11(4):436–450