

A Position-based Deployment and Routing Approach for Directional Wireless Mesh Networks

Weisheng Si and Selvadurai Selvakennedy

School of Information Technologies, University of Sydney, Sydney, Australia
weisheng@it.usyd.edu.au and s.kennedy@usyd.edu.au

Abstract—Observing that simplicity implies efficiency and scalability, this paper proposes a position-based deployment and routing strategy, and then gives a concrete approach under this strategy, for the emerging wireless mesh networks with multiple radios and directional antennas. The main idea of this strategy is to deploy the mesh network in certain kind of geometric graph and then design a position-based routing protocol by exploiting the routing properties of this graph. The proposed concrete approach comprises two parts: (1) a topology generation algorithm based on Delaunay triangulations and (2) a routing protocol based on the greedy forwarding algorithm. Both parts have appealing properties for deployment or routing, with formal proofs provided when applicable. Our simulation results strongly support this proposed approach.

Keywords: wireless mesh networks; deployment; position-based routing; directional antennas; Delaunay triangulation

I. INTRODUCTION

Wireless mesh networks (WMNs) typically consist of a set of mesh routers that communicate with each other via wireless links and form a mesh topology [1-5]. The basic functionalities of these mesh routers are:

- Provide the backhaul connection for Wireless LANs (WLANs).
- Perform routing for the backhaul.

Based on their additional functionalities, these mesh routers can be classified into three categories:

- *gateways*: also interface with the Internet.
- *APs*: also serve as Access Points (APs) for WLANs.
- *pure mesh routers*: only have the above basic functionalities.

In this paper, we solely focus on the basic functionalities of these mesh routers, aiming to improve their performance in the backhaul. For brevity, we also refer to these mesh routers as “nodes” hereafter.

Also according to [1-5], the last three years saw the following two trends for WMNs:

- *Multiple radios and multiple channels*: each node is equipped with multiple radios, each of which uses a distinct non-overlapping channel. By enabling multiple channels to carry network traffic simultaneously, this trend essentially multiplies the available bandwidth for network nodes.
- *Directional antennas*: these antennas are used in the backhaul connections, enabling the nodes to communicate in a point-to-point fashion. Thus, the interference among links sharing identical channels is basically eliminated.

Up to now, not all WMN deployments have realized both trends, and only some of them did (e.g., [1, 3]). However, with the

decreasing hardware cost, we envision that the WMNs in future will realize both. Accordingly, in this paper, we consider such WMNs with multiple radios/channels and directional antennas, and refer to them as Directional WMNs (DWMNs). To further clarify the architecture of DWMNs, Figure 1 illustrates a DWMN example, where four non-overlapping channels are used, and each radio on the same node uses a distinct channel, and the backhaul links are point-to-point. As an example of DWMN’s advantages, the two thick links on channel 3 in Figure 1 will not interfere with each other due to the use of directional antennas.

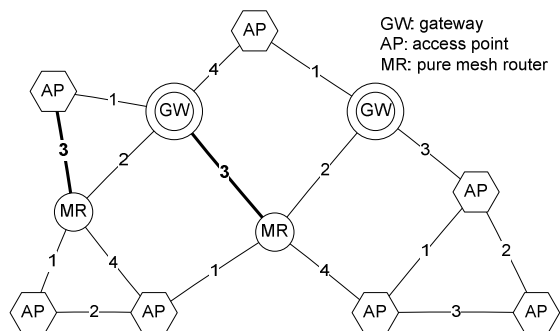


Figure 1. An example network of DWMNs

For the DWMNs, we propose a position-based joint deployment and routing strategy, which aims to perform whatever can be planned to simplify the operation of WMNs, so as to achieve efficiency and scalability. In retrospect, an important lesson from the haphazard-deployed Internet is the *Simplicity Principle* [11], which states that “complexity is the primary aspect which impedes efficient scaling, and is the primary driver of increases in both capital expenditure and operational expenditure for the network carriers”. Our basic idea for this strategy is to *deploy the DWMNs with certain kind of geometric graph as the network topology, and then design a position-based routing protocol by exploiting this graph’s routing properties*. The detailed justifications of this strategy are as follows:

- *Possibility*: the network topologies of many WMNs can be planned by the deployers, making this joint deployment and routing strategy possible.
- *Dynamics*: though the nodes in WMNs are stationary, there exist significant network dynamics such as link failure and congestion, making the static routing protocols unsuitable. However, for position-based routing protocols, such network dynamics can be easily considered.
- *Localness*: position-based routing protocols are *localized* [6] protocols in that the routing decision is solely based on a constant amount of information stored in the packets and the positions of the current forwarding node, its neighbors and

the destination. Thus, position-based routing protocols are highly efficient and scalable.

- **Ease of maintenance:** the localized property also significantly eases the maintenance of WMNs, since the reconfiguration to one part of the network does not need to be notified to the other parts.
- **Low overhead:** since the nodes in WMNs are configured with their neighbors' positions during the deployment, the periodical exchange of positions among them is obviated. In contrast, such overhead is considerable when the position-based routing protocols is used in *mobile* wireless networks [7, 8].
- **Practicality:** with the interference among backhaul links eliminated, the metrics related to positions become practical for making routing decisions. Otherwise, the interference-aware routing metrics (e.g., WCETT in [9] and MGF in [10]) seem more promising.

To actually validate this strategy, we present a concrete approach that consists of the following two parts dealing with deployment and routing respectively:

- *The PDT Generation Algorithm:* we use Delaunay triangulations (shortened as DT, see Section 3 for its definition and routing properties) as the basis to generate the network topologies. Specifically, this algorithm produces first refined and then pruned DTs (called PDTs by us). The PDTs have the advantages of (1) being practical for the deployment of DWMNs and (2) retaining good routing properties of the complete DTs.
- *The Backward-Enabled Greedy Forwarding protocol (BE-GF):* this protocol extends the well-known greedy forwarding algorithm [7] by enabling the delivery of a packet farther from the destination at certain hops without causing loops. BE-GF has the advantages of (1) being *localized* as defined previously and (2) considering the network dynamics such as link failure and congestion.

To our best knowledge, this is the first position-based approach that combines deployment and routing for WMNs. We show by extensive simulations that

- The PDT Generation algorithm can generate topologies that not only enable BE-GF to find paths very close to the actual shortest paths, but also reduce the installation cost for deployers.
- The BE-GF protocol can scale to a large number of nodes. With our implementation under ns-2, BE-GF can run on at least 1000-node networks and still ensure high throughput and low packet drop ratio for the upper layer applications.

The rest of this paper is organized as follows. Section 2 reviews the related work and distinguishes our work from others. Section 3 introduces the theoretical background for our approach. Section 4 presents the PDT generation algorithm and its evaluation. Section 5 presents the BE-GF protocol and its simulation results under ns-2. Finally, section 6 concludes this paper and points out the future work.

II. RELATED WORK

As our work covers two areas, node deployment and position-based routing, the related work in these two areas is discussed respectively below. Generally speaking, there is no

other work using the same idea as ours, and our work has its uniqueness in both areas.

A. Node deployment

The node deployment issue exists in the context of WMNs, wireless sensor networks (WSNs), and cellular mobile networks. Since WMNs just emerged in the recent years and the deployment issue has not received enough attention from the academic community, we only discovered one study [12] that investigates the impact of several factors such as number of radios, backhaul connectivity, and network topologies on the deployment of WMNs. As a simulated study on network performance, this work is based on extensive Monte Carlo simulations, which reveal several practical guidelines on the WMN deployment.

For WSNs, the study on deployment mainly considers the factors of area coverage, connectivity, and installation cost [13, 14]. It is worth noting that in WSNs, connectivity is generally optimized instead of routing, since the WSNs are more concerned with the capability of gathering data or detecting events than the efficiency of routing data. Currently, most research efforts in WSNs use a Poisson or grid deployment strategy.

For cellular mobile networks, the study on the deployment of base stations (BS) is called *cellular BS planning*, which is a major research topic in this area. For cellular BS planning, the main factors to consider include area coverage, traffic distribution, signal quality, and installation cost [15, 16]. Note that routing is not considered here, since routing is basically done at the mobile switching centers instead of BSs in cellular networks. The current BS planning approaches generally use the hexagonal deployment strategy.

In brief, the uniqueness of our approach for WMN deployment lies in the following:

- The main factors considered are routing and installation cost, since we believe that routing is the foremost functionality provided by the WMNs, while installation cost is a primary concern of a deployer.
- A novel kind of geometric graph, PDT, is proposed as the network topology.

B. Position-based routing

According to two surveys [17, 18], there is a vast literature in exploiting various geometric graphs to facilitate routing in wireless networks. These graphs mainly include the *relative neighborhood graph* (RNG), *Gabriel graph* (GG), *Yao graph* (YG), and *Delaunay triangulation*. Despite using different kinds of graphs, most of the existing position-based approaches have the following common characteristics:

- The wireless network environment is modeled by the *unit disk graph* (UDG), in which all network nodes use omnidirectional antennas with an identical transmission range and two nodes have a link between them if their distance is no more than the transmission range. Thus, there exists significant interference among links that are near each other, which makes the routing protocols solely based on position information not appealing.
- The network topology needs to be maintained by exchanging control messages among the nodes, which adds significant overhead to the network.

- In forwarding a packet to the destination, if the forwarding node does not have a neighbor closer to the destination than itself, *face routing* [6] is used to guarantee the delivery. Though face routing can guarantee the delivery in static networks, it cannot do this in dynamic networks where link failures make the forwarding loops possible.

Compared with these previous approaches, the uniqueness of our routing approach lies in the following:

- The UDG model is replaced with the point-to-point link model due to the use of directional antennas, thus the interference from nearby links is avoided.
- With the network topology planned in advance, the overhead of maintaining network topology is obviated.
- When the forwarding node does not have a neighbor closer to the destination, a simple technique of routing backward under certain conditions is used to increase the delivery ratio. This technique has the advantage that no forwarding loops can occur under network dynamics.

III. THEORETICAL BACKGROUND

In this section, the concepts and properties of greedy forwarding (GF) and Delaunay triangulation (DT) — the basis to understand our approach — are introduced.

GF is a position-based routing algorithm that finds a path from a source node s to a destination node d in the following method [7]: at each node (say u) along the path to d , u chooses the neighbor v that has the smallest $d(v, d)$ as the next hop, where $d(x, y)$ denotes the direct Euclidean distance between node x and y ; and ties are broken arbitrarily. GF is characterized by simplicity, since it is not only *localized*, but also makes the routing decision simply by one search of the neighbor list.

DT is widely used as network topologies by position-based routing protocols. It is defined as a triangulation graph that no fourth node lies inside the circumcircle of any triangle in this graph [19]. A DT has the following appealing properties for position-based routing:

- For a triangulation graph, let n denotes the number of nodes, e the number of edges, k the number of convex hull edges, we have $e = 3n - 3 - k$. This implies that in a DT, the average node degree is slightly less than 6 and hence bounded [19].
- In a DT, for any node u to any destination d , there always exists a neighbor v of u satisfying that $d(v, d) < d(u, d)$. Hereafter, we call this property *backward-free*. Since a DT is backward-free, GF can always find a path between any two nodes [20]. Besides, we also say that u routes a packet *backward*, if u routes this packet to its neighbor v with $d(v, d) \geq d(u, d)$.
- In a DT, the length of the shortest path between any two nodes u and v is guaranteed to be less than $c \cdot d(u, v)$, where c is proved a constant [21].

IV. THE PDT GENERATION ALGORITHM

A. Problem formulation

To formulate our topology generation problem, the following assumptions are made on the DWMNs.

- The positions of the AP nodes, largely decided by the sites of hotspots instead of a deployer, are essentially given. And a deployer can decide: (1) the positions of gateways and pure

mesh routers and (2) the existence of links between any pair of nodes (including all three categories).

- Due to the imperfection of directional antennas, though radios on the same node use distinct channels, the angle between any two links incident on a node should be no less than a threshold value θ_{min} , so as to fully avoid the inter-channel interference. Hereafter, an angle less than θ_{min} is referred to as a *bad angle*.
- Due to the regulation on transmitting power, the length of a point-to-point link cannot exceed a threshold value l_{max} . Hereafter, a link longer than l_{max} is referred to as a *bad link*.

Since a DT has the appealing routing properties mentioned in Section 3, it is straight-forward to use the DT obtained by directly triangulating the given AP nodes as the network topology (denoted by DT_{AP} hereafter). However, there usually exist bad angles and links in DT_{AP} , making it impractical for deployment. Figure 2 illustrates a DT_{AP} of 39 given AP nodes, where $\angle 1 - \angle 3$ are examples of bad angles, and l_1, l_2 are examples of bad links. To remove these bad angles and links, a proper way for deployers is to add pure mesh routers into the topology to change the triangulation. Thus, we formulate the following topology generation problem:

Given a set of AP nodes and the thresholds θ_{min} and l_{max} , find a graph T as the network topology by adding pure mesh routers into the DT_{AP} and recalculating the triangulation, such that:

1. T has no bad angles or bad links.
2. T is backward-free.
3. The number of pure mesh routers added is as small as possible.

Note that in this formulation, objective 2 is to guarantee the support to GF algorithm, and objective 3 is to reduce the installation cost for a deployer. Though adding more pure mesh routers can make the network more robust, this issue is not addressed in this paper.

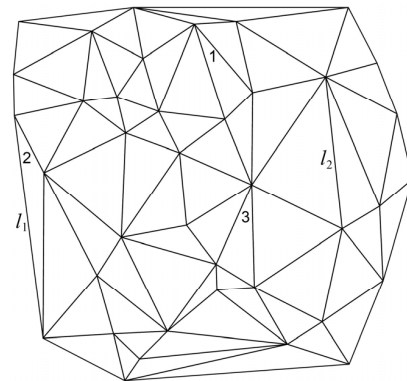


Figure 2. A DT_{AP} of given 39 AP nodes

B. Algorithm description

To solve the problem formulated above, we propose the PDT generation algorithm, which has the following three stages:

- 1) DT construction: construct the DT_{AP} of the given AP nodes.
- 2) DT refinement: add pure mesh routers to remove bad angles and links in the interior of the DT_{AP} , producing a refined DT (denoted DT_R hereafter) that has bad angles and links only near the boundary.

- 3) DT pruning: remove the remaining bad angles and links in DT_R by eliminating certain related edges, while guaranteeing that the resulting graph (called PDT) is still backward-free.

Since there is extensive research on DT construction and refinement by adding new nodes in the literature of computational geometry, stages 1 and 2 are accomplished using the existing state-of-the-art algorithms, which are implemented in the open-source software Triangle [22].

For stage 1, we use the well-known divide-and-conquer algorithm [19] to construct the DT_{AP} . For stage 2, the refinement algorithm implemented in Triangle is a hybrid of three algorithms [23-25] ever proposed in the literature, and it can refine an input DT into another DT that satisfies certain specified criteria. Software Triangle calls a triangle that does not satisfy the specified criteria a *bad triangle*, and its basic refinement operation is to add a new node at either the circumcenter [23] or the off-center [25] of each bad triangle to split it, until there are no bad triangles.

In using Triangle to do the refinement, we use the criteria of $\theta_{min} = 30^\circ$ and $l_{max} = 500m$. The 30° is used because Triangle is proved to terminate when $\theta_{min} \leq 30^\circ$ [23, 25], otherwise it may infinitely split the triangles. Actually, it can usually terminate at a larger θ_{min} in practice [25]. For l_{max} , Triangle can terminate at any value.

When removing a bad triangle near the boundary of a DT, if the circumcenter or off-center of this bad triangle lies outside the boundary of this DT, Triangle will not add a new node at the circumcenter or off-center. Instead, several new nodes will be added on the boundary of this DT to remove this bad triangle. This phenomenon is illustrated in Figure 3, where the DT_{AP} in Figure 2 is the input, and bold points depict those new nodes added on the boundary. It can be seen that the number of such new nodes is large and these new nodes are not in the critical positions for relaying network traffic.

Based on this observation, we use the options provided in Triangle to prohibit it from adding new nodes on the boundary when performing the refinement. Instead, we propose a DT pruning algorithm as stage 3 to remove those bad triangles left in stage 2 by removing some edges. Thus, considerable cost for installing new nodes can be saved for a deployer. Besides, we also want this pruning algorithm to retain the backward-free property of complete DTs. To illustrate this pruning idea, Figure 4 (using the DT_{AP} in Figure 2 as the input) shows the actual DT_R obtained by stage 2 with some bad triangles left near the boundary (e.g., $\Delta A - \Delta E$), and Figure 5 shows the PDT obtained by our pruning algorithm. For both figures, the bold points depict the pure mesh routers added within the boundary of the DT_{AP} during the refinement.

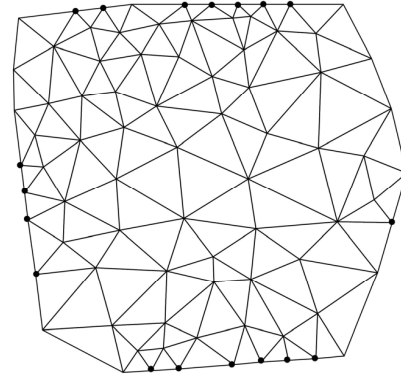


Figure 3. The DT_R with new nodes added on the boundary

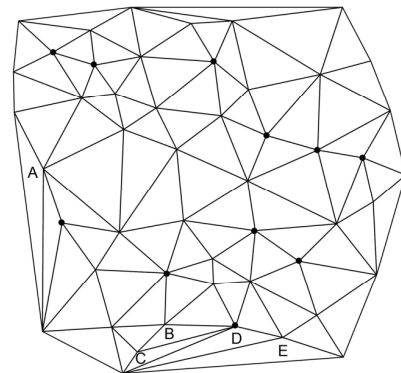


Figure 4. The DT_R without new nodes added on the boundary

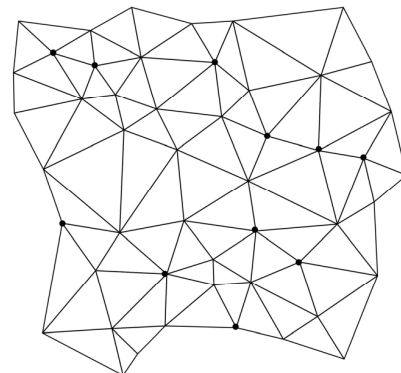


Figure 5. The PDT obtained by our pruning algorithm

According to [23, 25], a bad triangle left in stage 2 satisfies the following two properties: (1) it is a right or obtuse triangle with its circumcenter or off-center outside the boundary of the DT_R and (2) it either has a vertex on the boundary of DT_R (e.g., ΔA in Figure 4), or is recursively adjacent to another bad triangle that has a vertex on the boundary of DT_R (e.g., ΔB in Figure 4). Based on these two properties, the basic idea of our pruning algorithm is as follows: repeatedly traverse around the boundary of DT_R to remove those bad triangles that have a vertex on the boundary, until no such triangles are found; in removing a bad triangle, delete the edge that opposes the right or obtuse angle of this bad triangle. The details of our pruning algorithm are given in Figure 6, and some comments are also included for explanation.

```

1. // Initialization
   graph  $T = DT_{\mathbb{R}}$ ;
    $v$  = an arbitrary node on the boundary of  $T$ ;
2. // Traversing each node on the boundary of  $T$  clockwise
   do {
     // examining triangles at node  $v$ 
     for (each triangle with  $v$  as a vertex) {
       // to keep the triangulation structure during the
       // examination, only mark for deletion here
       if (it is a bad triangle)
         mark the edge opposite to its right or obtuse
         angle, the longest edge in this triangle, for deletion;
     }
      $T = T -$  those marked edges;
     // advance  $v$  in clockwise order
      $v$  = next node on the boundary of  $T$ ;
   }
3. Output  $T$  as the final PDT;

```

Figure 6. The DT pruning algorithm

To justify the backward-free property of PDTs, we prove the following theorem. Due to the page limit, the proof is omitted below, but can be found in our technical report [26].

Theorem 1: *The PDTs obtained by our proposed topology generation algorithm are backward-free.*

As a final note, it is empirically shown that Triangle adds very limited number of new nodes during the refinement [25], but no theoretical results on the minimum number of nodes required are known to date [27]. Therefore, we use the refinement algorithm implemented in Triangle as a practical means to achieve the objective 3 in our problem formulation, and no optimal guarantee is provided right now. Our evaluations presented next also show that Triangle indeed adds only a small proportion of new nodes in our settings.

C. Algorithm evaluation

This subsection first describes the experiment setup, and then gives the major experimental results on the following three metrics: the ratio of added pure mesh routers, the path efficiency of hops, and the path efficiency of Euclidean distance. The first metric reflects the installation cost for deploying the PDTs, and the last two metrics reflect the path quality of PDTs with GF as the routing algorithm. For other experimental results, please refer to our technical report [26].

1) Experiment setup

The given AP nodes are assumed to be randomly distributed in a square area with a constant density, and this density is set to 10 nodes/km². For instance, if the number of nodes is 20, the side length of the square area is approximately 1414m. Also, since two AP nodes can not be too close in reality, a minimum distance of 50m between any two AP nodes is assumed. Since a DT exists for any set of points on a plane [19], our PDT generation algorithm works for any kind of AP distributions. We only use the random distribution here for conducting experiments.

Our PDT generation algorithm is implemented by modifying the aforementioned software Triangle. With this implementation, experiments are conducted on network sizes of 50, 100, 200, 400, 700, and 1000 given AP nodes respectively. For each network size, 200 random topologies are generated and the average result of them is calculated.

2) The ratio of added pure mesh routers

The ratio of added pure mesh routers for a topology equals the number of added pure mesh routers divided by the total number of nodes after the topology generation. Figure 7 plots the average ratio of 200 topologies for each experimented network size. From this figure, we see that (1) this ratio increases with the growth of the network size, while the increasing speed slows down gradually; and (2) this ratio is less than 0.5 for all plotted network sizes. Therefore, it can be inferred that the installation cost of pure mesh routers is below half of the installation cost for all nodes and tends to increase linearly in relation to the network size.

3) The path efficiency of hops

Given a routing algorithm, the path efficiency of hops for a pair of source and destination (s, d) is defined as the ratio of the minimum number of hops between s and d versus the number of hops in the path found by this algorithm from s to d . The path efficiency of hops for a topology is defined as the average path efficiency for all (s, d) pairs in this topology by this algorithm. Similarly, given a routing algorithm, the path efficiency of Euclidean distance for a (s, d) pair or for a topology can be defined in terms of Euclidean distance instead of hops.

These two efficiency concepts of hops and Euclidean distance are first introduced in [28], and their basic meaning is that the closer the efficiency is to 1, the better path quality the network topologies can provide for a given routing algorithm. As to be described later, since our BE-GF protocol reduces to the GF algorithm if no network dynamics exist, we use GF here to evaluate the path quality of PDTs. Henceforth, GF will no longer be mentioned explicitly when discussing the path efficiencies.

Figure 8 plots the average path efficiency of hops of 200 topologies for each experimented network size. To make a comparison between the topologies before and after the refinement and pruning, we plot the efficiencies for DT_{AP} and PDT respectively. From this figure, we see that (1) the path efficiencies for both DT_{AP} and PDT decrease with the growth of network size, because the differences between the paths found by GF and the actual shortest paths tend to become larger when the number of hops in the network paths increase; (2) PDT has significant improvement on this path efficiency over DT_{AP}; and (3) for PDTs, this efficiency remains above 0.9 for all plotted network sizes. *Therefore, it can be concluded that the PDTs provide very high path quality in terms of number of hops.*

4) The path efficiency of Euclidean distance

The definition is given in the previous subsection. Figure 9 plots the average path efficiency of Euclidean distance of 200 topologies for each experimented network size. For the same purpose as in Figure 8, the efficiencies for DT_{AP} and PDT are also plotted respectively. From this figure, we see that (1) the path efficiency of Euclidean distance is generally higher than its counterpart of hops, since the distance is an accurate real number while the number of hops is simply a rough integer; (2) for this efficiency of Euclidean distance, PDT also has significant improvement over DT_{AP}; and (3) for PDTs, this efficiency remains above 0.95 for all plotted network sizes. *Therefore, it can be concluded that the PDTs also provide very high path quality in terms of Euclidean distance.*

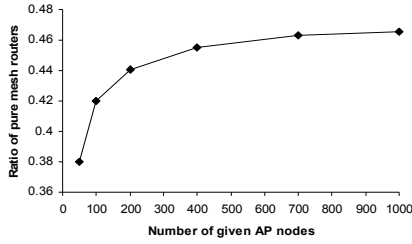


Figure 7. The ratio of added pure mesh routers

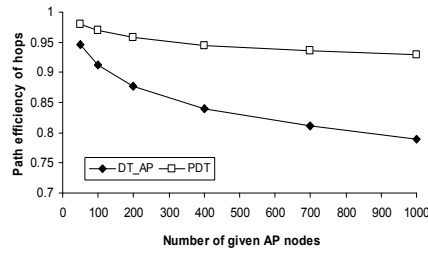


Figure 8. The path efficiencies of hops for DT_{AP} and PDT

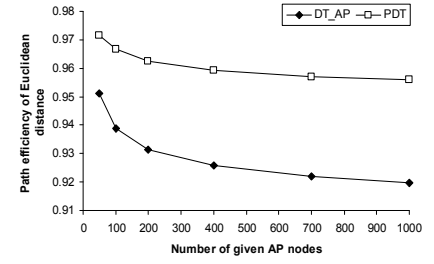


Figure 9. The path efficiencies of Euclidean distance for DT_{AP} and PDT

V. THE BE-GF PROTOCOL

A. Protocol overview

Given PDTs as network topologies, BE-GF is designed with the following main ideas.

- It is based on the GF algorithm. When no network dynamics exist, it reduces to GF and fully exploits the backward-free property of PDTs.
- It extends GF by considering two kinds of network dynamics: *link failure* and *link congestion*. The former can be promptly notified by the link-layer functionalities, and the latter can be obtained by examining the occupancy of link packet buffers. If the packet buffer for a link is full, the BE-GF simply does not send packets to this link. Thus, both kinds of network dynamics can be monitored locally by nodes, such that no exchanges of network-layer control packets are involved.
- With the consideration of network dynamics, the underlying network topology may not be backward-free. Accordingly, BE-GF supports routing a packet backward to increase its delivery ratio. However, this may result in loops. To prevent loops, BE-GF stipulates that a node u can route a packet with destination d backward, only when $d(u, d) < d(b, d)$, where b is the node that routes this packet backward last time (if exists); otherwise, this packet is dropped. Thus, a packet only needs to remember one node (i.e., b) in its header while traversing the network.

Compared with face routing, the BE-GF protocol has the following advantages and disadvantages:

- **Advantages:** First, in case of network dynamics, no forwarding loops can occur, while face routing can incur loops [6]. Second, the amount of information stored in the packet header is significantly less than that of face routing; for instance, in GPSR [7], a well-known face routing protocol, five new fields are added into the packet header.
- **Disadvantages:** Since packets are dropped under the aforementioned condition $d(u, d) \geq d(b, d)$, they may not be delivered to the destination when there exists a path. Face routing performs better in this aspect, because it guarantees to find a path if the network is connected and static. However, with the PDTs being backward-free and having an average node degree of approximately six, the packets drops in BE-GF is not significant, which is verified in our simulations.

B. Protocol description

Required by BE-GF, at most three fields are added to a packet's network-layer header. Their meanings are as follows:

- **Dst Coords:** the (x, y) coordinates of the destination node.

- **Backward Flag (BF):** this flag is used to indicate whether a packet has ever been routed backward. That is, whenever a packet is routed backward by a node, its BF is set to 1 by this node and will not change later.
- **Backward Coords:** This field is present only when the BF is 1. When a node sets the BF of a packet to 1, it also copies its (x, y) coordinates into this field. If a packet is routed backward by one node and later by another node, the later node's coordinates will overwrite the previous node's coordinates.

Exploiting these three fields in packets as well as the neighbor lists at nodes, the BE-GF protocol is detailed in Figure 10, where the node running BE-GF is denoted u , the packet being processed P , P 's destination node d , and the node remembered by Backward Coords (if present in P) b . For explanation, several comments are also included in Figure 10.

```

1. // search the neighbor list
   u looks for a neighbor v with the shortest d(v, d) that satisfies:
   (1) link uv is up;
   (2) packet buffer of link uv is not full;
       // avoid sending P back to b
   (3) v is not b (if Backward Coords exists in P);
2. // all neighbors are not available
   if (v is not found) {
       u drops P;
       return;
   }
3. // v is closer to d than u
   if (d(v, d) < d(u, d)) {
       u forwards P to v;
       return;
   }
4. // otherwise, check whether to route this packet backward
   if (BF in P == 0 || d(v, d) < d(b, d)) {
       u puts 1 into BF of P;
       u puts its coordinates into Backward Coords of P;
       u forwards P to v;
   }
   else {
       // BF in P == 1 and d(v, d) >= d(b, d)
       u drops P;
   }

```

Figure 10. The BE-GF protocol

We highlight here that BE-GF drops a packet P in the following two cases:

- 1) At step 2 of the protocol, when no neighbors of u are available for forwarding P . This case occurs when the network is in a very poor condition — all links adjacent to u either fail or are congested.
- 2) At step 4, when neighbor v is available for backward routing, but $d(v, d) \geq d(b, d)$. As to be detailed in the proof to

Theorem 2, this case is enforced to prevent P from entering a loop.

We next prove the following two theorems regarding the loop-freeness and complexity of BE-GF respectively. Easy for reading, these two proofs also help to understand our protocol.

Theorem 2: *For any network topology, BE-GF is loop-free in that it either delivers a packet (denoted by P) to the destination (denoted by d) or drops P .*

Proof: The main idea is to show that P under BE-GF has the general trend to move closer to d , so no loops can occur.

Specifically, the proof is done in two cases. On one hand, suppose P is never routed backward toward d . In this case, P gets closer to d in each hop, so P arrives at a new node each time. Since there are limited number of nodes in the network, P is certain to reach d . On the other hand, suppose P is ever routed backward. Since BE-GF stipulates that P can only be routed backward at node u if u has a shorter distance to d than the node that routes P backward last time, it follows that u is a different node from all other nodes that have ever routed P backward. Because there are limited number of nodes in the network, P can only be routed backward by limited times, which contradicts a loop. Therefore, P surely reaches d or is dropped. Q.E.D.

As a worthy note, this proof reveals that the loop-free property of BE-GF does not rely on the backward-free property of the network topologies. Hence, the purpose for PDTs to be backward-free is not to guarantee the loop-freeness of BE-GF, but to reduce the packet drop ratio of BE-GF. As to be shown in our evaluations, the packet drops happen very rarely with PDTs as the network topology.

Theorem 3: *For any network topology with constant maximum node degree, BE-GF runs with $O(1)$ complexity in both time and space at a node.*

Proof: For the time complexity, the only costly computation in BE-GF is searching the neighbor list. Given the maximum node degree is constant, this search can be completed in $O(1)$ time.

For the space complexity, a node needs to store its neighbor list in the memory, which costs $O(1)$ space due to the constant maximum node degree. Also, a packet, which also costs a node's memory, needs to carry at most three fields in its header for BE-GF. Combining these two space costs, BE-GF consumes $O(1)$ space at a node. Q.E.D.

Note that this theorem certainly holds for PDTs, where the average node degree is less than 6 due to the facts presented in Section 3, and the maximum node degree is 12 due to $\theta_{min} = 30^\circ$.

C. Protocol evaluation

We implement the BE-GF protocol in ns-2 [29], into which two parts of codes are added: a routing module that realizes the steps in Figure 10 and a link object that simulates the point-to-point wireless links. To simulate these point-to-point links, the WirelessChannel object in ns-2 is modified, so that it only delivers the wireless signal to the two neighboring nodes in a point-to-point link. Besides, to ease our implementation and comparison with other's work, the IEEE 802.11 is kept as the link and physical layer protocols, though new protocols in these two layers are desired for these point-to-point wireless links.

Using the network topologies produced by our PDT generation algorithm, we conduct experiments on BE-GF with network sizes of 50, 100, 200, 400, 700, and 1000 given AP

nodes respectively. To evaluate the performance of BE-GF upon link failures, the Exponential Model included in ns-2 is used to generate link dynamics. In this model, the up/down time for a link is exponentially distributed with the mean *up-interval* and *down-interval* respectively, and both intervals are configurable. Using this model, we conduct experiments on the following four scenarios for each network size:

- Static: no link failures
- Dyna1: for each link, up-interval = 10s, down-interval = 2s
- Dyna2: for each link, up-interval = 10s, down-interval = 4s
- Dyna3: for each link, up-interval = 10s, down-interval = 6s

For each scenario of a given network size, 50 experiments with different random topologies are conducted and the average result of them is calculated. In each experiment, every node is found a peer randomly, and then for each pair of peering nodes, two ftp flows in opposite directions are generated. That is, if there are n nodes in the network, then $2n$ ftp flows are generated. This is to test the performance of BE-GF in a heavily-utilized network, as TCP tries not to waste the network bandwidth.

Next, major evaluation results on the following three metrics for BE-GF are presented: network throughput, packet drop ratio, and backward ratio. Besides these metrics' own meanings, their changing patterns with network sizes also reflect the scalability of BE-GF. For other experimental results, please refer to our technical report [26].

1) Network throughput

Network throughput is defined as the total throughput of all communicating node pairs in the network. Figure 11 plots our experimental results on it, mainly showing that (1) for all four scenarios, the network throughput increases steadily at least up to 1000 nodes, implying BE-GF is highly scalable; and (2) the network throughput drops as the network becomes more dynamic, implying that more link failures incur more packet drops, which have a significant impact on the TCP performance.

2) Packet drop ratio

Packet drop ratio is defined as the total number of data packets dropped by all nodes during routing divided by total number of data packets generated by all nodes. Figure 12 plots our experimental results on it, showing that (1) for the scenarios Static and Dyna1, the packet drop ratios are extremely low (less than 0.01) and (2) for the scenarios Dyna2 and Dyna3, the packet drop ratios become significantly larger, but are still below 0.1. Besides, according to the evaluations in our technical report, the packet drops at step 2 of BE-GF are negligible compared with those at step 4. Since the packet drops at step 4 is enforced to prevent loops by BE-GF, Figure 12 also reflects that the loop-free property of BE-GF is achieved at a very small price of packet drops with PDTs as the network topology.

3) Backward ratio

Backward ratio is defined as the total number of backward routing decisions made by all nodes divided by total number of routing decisions made by all nodes. Figure 13 plots our experimental results on it, mainly showing that: (1) For all the four scenarios, the backward ratios are very low (less than 0.08), reflecting that BE-GF seldom routes a packet backward when running on top of PDTs. This is because, according to the facts presented in Section 3, the average node degree in PDTs is slightly less than 6, so a forwarding node has a large chance to find a neighbor closer to the destination.

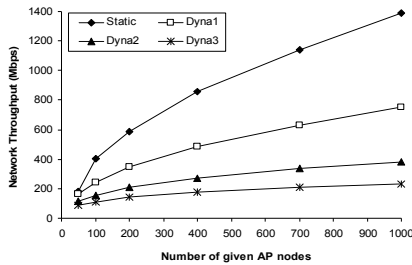


Figure 11. Network throughput

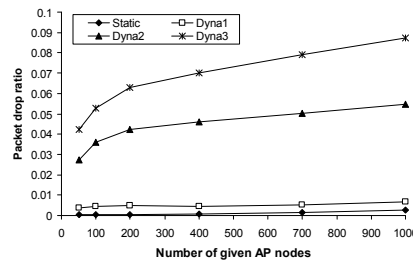


Figure 12. Packet drop ratio

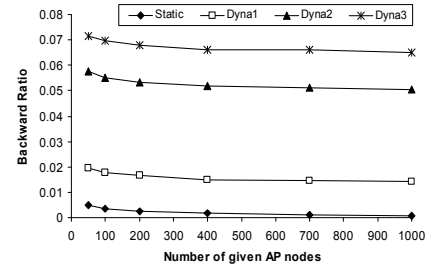


Figure 13. Backward ratio

(2) For all the four scenarios, the backward ratios drop slowly with the growth of network size. This is because a routing decision is made for a packet at each node it traverses, so the total number of routing decisions increases faster than the total number of backward routing decisions.

VI. CONCLUSION

Observing that simplicity brings the efficient and scalable operation of networks, this paper newly proposed a position-based deployment and routing strategy, and then gave a concrete approach under this strategy for the DWMNs. Our experimental results validated this concrete approach.

It is worth mentioning that the proposed new strategy opens a novel ground of research. Under this strategy, many novel problems related to joint deployment and routing can be explored. Moreover, the backward-free property is newly introduced for geometric graphs. Since the possession of this property can significantly improve the performance of position-based routing protocols, investigating the sufficient and necessary conditions for a geometric graph to be backward-free becomes an interesting open problem. While our proposed PDT is only an example of the backward-free graphs, other kinds of backward-free graphs can be further explored to suit the different requirements of network deployment and routing protocols.

For future work, we note that the BE-GF protocol is proposed only as a preliminary protocol that can deal with network dynamics while causing no loops. Our continuing work includes: (1) comparing BE-GF against existing face routing protocols with various traffic scenarios and (2) designing a routing protocol with less packet drop ratio for DWMNs, especially when the network suffers from frequent link failures.

REFERENCE

- [1] BelAirNetworks, "BelAir300 Converged Multi-service Wireless Node," <http://www.belairnetworks.com>, Data Sheet 2007.
- [2] MeshDynamics, "Technology - Performance Analysis," <http://www.meshdynamics.com>, Online 2006.
- [3] Nortel, "Wireless Mesh Network Basics," <http://www.nortel.com>, Standard Document Release 3.0 2007.
- [4] StrixSystems, "Solving the wireless mesh multi-hop dilemma," <http://www.strixsystems.com>, White Paper 2006.
- [5] TroposNetworks, "Tropos Networks MetroMesh Architecture," <http://www.tropos.com>, White Paper 2005.
- [6] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," in *ACM MobiCom*, 2006, pp. 390-401.
- [7] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks" in *ACM MobiCom*, 2000, pp. 243-254.
- [8] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanners for routing in mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 174-185, 2005.
- [9] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *ACM Mobicom*, 2004, pp. 114 - 128.
- [10] P. Kyasanur and N. H. Vaidya, "Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 10, 2006.
- [11] R. Bush and D. Meyer, "Some Internet Architectural Guidelines and Philosophy," in *RFC 3439, Informational*, 2002.
- [12] J. Robinson and E. W. Knightly, "A Performance Study of Deployment Factors in Wireless Mesh Networks," in *IEEE Infocom*, 2007, pp. 2054 - 2062.
- [13] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. Shen, "Topology control for wireless sensor networks," in *ACM MobiCom*, San Diego, CA, USA, 2003, pp. 286-299.
- [14] C.-F. Huang and Y.-C. Tseng, "The Coverage Problem in a Wireless Sensor Network," *Mobile Networks and Applications*, vol. 10, pp. 519-528, 2005.
- [15] E. Amaldi, A. Capone, and F. Malucelli, "Planning UMTS Base Station Location: Optimization Models with Power Control and Algorithms," *IEEE Transactions on Wireless Communications*, vol. 2, pp. 939-952, 2003.
- [16] R. Mathar and T. Niessen, "Optimum positioning of base stations for cellular radio networks," *Wireless Networks*, vol. 6, pp. 421-428, 2000.
- [17] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, pp. 30-39, 2001.
- [18] D. Chen and P. K. Varshney, "A Survey Of Void Handling Techniques For Geographic Routing In Wireless Networks," *IEEE Communications Surveys & Tutorials*, vol. 9, pp. 50-67, 2007.
- [19] M. d. Berg, M. v. Kreveld, M. Overmars, and O. Schwarzkopf, *Computational geometry: algorithms and applications*, 2nd ed. New York: Springer, 2000.
- [20] P. Bose and P. Morin, "Online Routing in Triangulations," in *International Symposium on Algorithms and Computation*, 1999, pp. 113-122.
- [21] J. M. Keil and C. A. Gutwin, "Classes of graphs which approximate the complete euclidean graph," *Discrete and Computational Geometry*, vol. 7, pp. 13-28, 1992.
- [22] J. R. Shewchuk, "Triangle version 1.6," <http://www.cs.cmu.edu/~quake/triangle.html>, 2005.
- [23] P. Chew, "Guaranteed Quality Mesh Generation for Curved Surfaces," in *The Annual Symposium on Computational Geometry*, 1993, pp. 274-280.
- [24] J. Ruppert, "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation," *Journal of Algorithms*, vol. 18, pp. 548-585, 1995.
- [25] A. Ungor, "Off-centers: A new type of Steiner points for computing size-optimal guaranteed-quality Delaunay triangulations," in *Proceedings of LATIN*, 2004, pp. 152-161.
- [26] W. Si and S. Selvakennedy, "A Joint Deployment and Routing Strategy for Directional Wireless Mesh Networks," School of IT, U. of Sydney, Technical Report, TR620, <http://www.it.usyd.edu.au/~weisheng/papers/tr620.pdf> 2008.
- [27] H. Erten and A. Ungor, "Triangulations with Locally Optimal Steiner Points," in *Eurographics Symposium on Geometry Processing*, 2007, pp. 1-10.
- [28] P. Cucka, N. S. Netanyahu, and A. Rosenfeld, "Learning in navigation: Goal finding in graphs," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, pp. 429-446, 1996.
- [29] NS-2, "Network Simulator 2," <http://www.isi.edu/nsnam/ns>, 2008.