Computer
Sciences
Department

Building Robust Wireless Mesh Networks Using Directional Antennas: How Many Radios are Enough?

Qunfeng Dong
Yigal Bijerano

Technical Report #1572

July 2007

UNIVERSITY OF
WISCONSIN
MADISON

Building Robust Wireless Mesh Networks Using Directional Antennas: How Many Radios are Enough?

Qunfeng Dong

Yigal Bejerano

Abstract— **Recently, wireless mesh technology has been used for military applications and fast recovery networks, referred to as *nomadic wireless mesh networks* (NWMNs). In such systems, wireless routers, termed nodes, are mounted on top of vehicles or vessels. The vessels may change their locations according to application needs and the nodes are required to establish a broadband and reliable wireless mesh network. For improving network performance, some vendors use directional antennas and the mesh topology comprises of point-to-point connections between adjacent nodes. Consequently, the number of point-to-point connections of a node is upper-bounded by the number of directional radios (and antennas) that it has, which is typically a small constant. This raises the need to build robust (i.e., two-node/edge-connected) mesh networks with bounded node degree, regardless of the node locations. This paper presents practical solutions with provable properties for constructing efficient and robust wireless mesh networks using directional antennas. First, we formulate the design problem to be theoretically equivalent to the construction of bounded degree two-connected mesh topologies. Then, we present simple schemes for constructing such solutions with small constant degree bounds. Finally, our extensive simulations show that our schemes find robust and efficient topologies for various settings with node degree bounded by 4, while preserving small hop-count distance between nodes and gateways.**

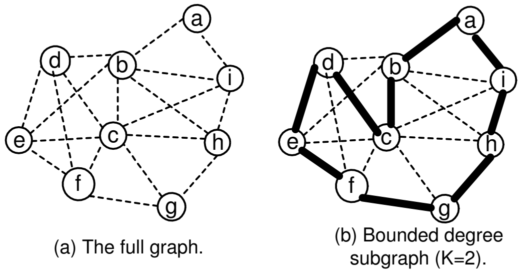
I. Introduction

Wireless mesh network has been widely recognized as an emerging technology for low-cast fast deployed communication networks. Nowadays, they are used for numerous applications such as wireless backhaul, public safety, and public Internet access [1], [2]. In such systems, some wireless routers, referred to as *nodes*, are *statically* deployed at different locations. Each node is typically equipped with multiple *mesh radios* to form a connected *mesh* [3], [4], [5]. Besides the mesh radios, each node may also have other wireless interfaces to form wireless local area networks (WLANs) for client access. Some of the wireless routers, termed *gateways*, are connected to the Internet through additional network interfaces. Thus, by routing packets within the mesh, wireless mesh networks can be used for both local communication and Internet access.

For improving network performance, several equipment vendors [3], [4], [5] connect mesh radios to directional antennas, referred to as *directional mesh radios*. Directional antennas have a number of technical advantages over conventional omni-directional antennas, including extended transmission range, low interference, low transmission power, and so on [6], which make them very attractive for *static* and *quasi-static* environments. For simplifying network deployment, some vendors [3], [4] utilize point-to-point paradigm. In particular, each directional mesh radio is paired with another directional mesh radio installed on another node within reach (i.e., *neighbor*) to form a point-to-point connection between them. For that, the two directional mesh radios must be properly oriented toward each other and assigned to the same wireless channel. Directional mesh radios on different links are assigned to orthogonal channels to avoid interference.

Recently, wireless mesh technologies have been used for fast deployment of disaster recovery networks and for military applications. In these applications, wireless mesh routers are mounted on top of vehicles or ships [7], which may change their location according to application needs. The nodes are required to identify their neighbors and establish point-to-point connections with some of them to form a robust connected network, regardless of node locations, referred to as *nomadic wireless mesh networks* (NWMN). NWMNs are required to provide broadband and reliable communication by using wireless mesh routers assuming quasi-static mobility patterns, i.e., a node may change its location but it tends to stay in the same place for a long duration. For such applications, directional mesh radios can be efficiently utilized to establish high capacity point-to-point connections, without suffering from the typical problems of dynamic directional antenna-based environments, such as deafness [7], [8].

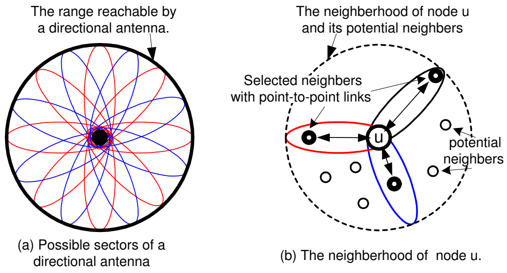
In this paper, we study the problem of building robust NWMN topologies, where the degree of any node does not exceed its number of available directional mesh radios. This study is inspired by a commercial project of our organization for designing a dynamic fault resilient wireless mesh networks for *naval applications*. In this project, we install a wireless mesh router (e.g., BelAir200 mesh nodes [9]) on each vessel, and the mesh routers are required to form a *robust wireless backbone* (i.e., *two-edge-connected* or *two-node-connected*) composed of point-to-point connections between adjacent nodes. Generally speaking a network is termed two-edge/node-connected if there are two edge/node disjoint paths between every pair of nodes. Thus, even if a link or node becomes temporarily unavailable because of ship movement or link failure, the entire network is still connected. These networks will be used for two main applications. The first is providing broadband and reliable communication for a fleet of ships at sea, where the flagship is used as the gateway. The second is connecting ducked ships in a harbor with the shore, where a few gateways are deployed on the shore side.



(a) The full graph.

(b) Bounded degree subgraph (K=2).

Fig. 1. An example of a bounded degree robust network.


The range reachable by a directional antenna.
The neighberhood of node u and its potential neighbers
Selected neighbers with point-to-point links
u
potential neighbers

(a) Possible sectors of a directional antenna

(b) The neighberhood of node u.

Fig. 2. An example of a node neighborhood and selected neighbors.

We assume all the routers are identical and hence have the same number of directional mesh radios, denoted by K, which we refer to as the *maximal node degree of the graph* or simply as the *graph degree*. In our study, we give as input a *full graph* comprised of all the *candidate links* between every pair of adjacent nodes that may establish a point-to-point communication link with an adequate channel quality. Since the maximal node degree of the full graph may be higher than K, our objective is to find a robust (i.e., 2-node/edge connected) sub-graph of the full graph with a maximal node degree at most K. For example, consider the network in Figure 1-(a), where the dashed lines denote the candidate links. Let us assume each node is equipped with $K = 2$ directional mesh radios and our objective is to build a strongly connected mesh topology. Although each node is adjacent to multiple potential neighbors, in the mesh topology we build (represented in Figure 1-(b)) every node establishes point-to-point connections only with two of its neighbors.

A. Wireless Mesh Networks With Directional Antennas

Directional antennas have gained a lot of attention in the last decade as a proved technology for improving the performance of wireless systems. Recently, directional antennas have also been used to improve the performance of wireless ad-hoc networks [7]. However, in dynamic multi-hop wireless environments, directional antennas raise several deployment difficulties that result from their asymmetric transmission characteristics. These difficulties include neighbor discovery, deafness, and new types of hidden node problems [7], [8], which have motivated the development of new MAC protocols [10], [11]. Other challenges, such as routing [12], [13], [14] and topology control for power efficiency [15], [17], [16] have been addressed at the networking layer. However, very little attention was given to topology control algorithms that construct strongly connected network topologies with bounded node degree.

B. Bounded Degree Sub-graphs

A similar problem to the one we address is finding minimum degree spanning subgraphs, i.e. trees or two-connected subgraphs, which is known to be NP-hard for general graphs [18]. Addressing this challenge, Fürer and Raghavachari introduce in [19] a near optimal solution that finds a spanning tree whose degree exceeds the optimal degree by at most one. The problem of finding minimum degree two-connected spanning subgraphs is more challenging. To this end, Klein *et al.* [20] provide a quasi-polynomial time approximation algorithm for finding two-edge-connected spanning subgraphs, whose degree in all cases is guaranteed to be at most $1 + \epsilon$ times the optimal degree Δ^*, plus an additive $O(\log_{1+\epsilon} n)$ for any $\epsilon > 0$. Thus, for large graphs the later cannot guarantee to find a two-connected subgraph with a small degree.

Unlike the minimum degree spanning subgraph problems that seek to minimize the degree of a given graph, our objective is to find a two-connected subgraph with bounded node degree regardless of the full graph topology. Obviously, in the case of arbitrary full graphs, the answer to this challenge may be as high as the number of the graph nodes minus two[1]. However, in wireless networks, a node typically may establish point-to-point connections only with nodes in its geographic vicinity. We assume the use of identical mesh nodes with the same sector size on all directions, as depicted by Figure 2-(a). From this implies that the potential neighbors of a node are located in a disk centered at the node location, referred to as the node *neighborhood*. For instance, Figure 2-(b) a node u with eight potential neighbor in its vicinity and it has established point-to-point connections with only three selected neighbors. Consequently, the full graph can be properly modeled as a unit disk graph (UDG) [21], *i.e.,* a candidate link exists between two nodes in the full graph if and only if they are within a given distance R from each other[2]. This is a commonly used model for wireless networks and, especially, it is a reasonable model for naval applications, which are the main interest of this study.

For UDGs, several algorithms for calculating bounded degree subgraphs have been designed. In [22], Wang and Li present an algorithm for localized construction of a bounded degree planner spanner. They proved a constant stretch factor and a bounded node degree, but the degree bound is very high (20). In [23], Li, Hou and Sha describe a minimum spanning tree based topology control algorithm that builds connected subgraphs with a bound of 6 on node degree. Later, in [24], Kumar, Gupta and Das present a topology control scheme that finds a small degree spanning tree by using the algorithm of

[1] For example, consider a graph comprised of two stars that share the end-nodes but the two hub nodes are not connected.

[2] In this study we are interested only in high capacity point-to-point connections and we assume that R is selected accordingly.

Fürer and Raghavachari in [19]. Their simulations show that typically 3 or 4 mesh radios are sufficient to construct such a tree. Recently, in [25], Wu *et al.* show that connected UDGs have spanning trees of degree 5 or less. To the best of our knowledge, this paper is the first to address the problems of finding bounded degree two-connected subgraphs of UDGs, which are known to be NP-hard [26].

C. Our results

This study has two fundamental objectives. The first is determining the minimal number of required directional mesh radios attached to each node to ensure two-connectivity, assuming the full graph is a UDG that satisfies the same two-connectivity requirement. To this end, we show that some instances require the graph degree to be at least 10 and 5 for ensuring two-edge-connectivity and two-node-connectivity, respectively.

Our second objective is finding simple algorithms for determining robust mesh topologies with guaranteed low bound on node degree. Besides connectivity requirements and degree bounds, we also consider other performance metrics such as throughput and end-to-end communication delay. For throughput maximization we assume that the full graph comprised only of candidate links with high capacity[3] and for reducing delays we seek solutions with small hop count distance between every node and its closest gateway.

This study introduces a simple two-phase scheme for building efficient two-connected subgraphs for any arbitrary full graph, instead of just a UDG. In the first phase, we build two-connected sparse subgraphs for any given full graph. If the given full graph is indeed a UDG, then small constant degree bounds can be derived for the subgraphs we build. More specifically, we describe two simple algorithms that for UDGs find subgraphs with maximal node degree of 10 and 6 that ensure two-edge-connectivity and two-node connectivity, respectively. These two-connectivity algorithms are the main algorithmic result of the paper. In the second phase, we perform a simple link augmentation process for reducing the hop-count distance of the nodes to the gateways, while preserving the subgraph degree bound and connectivity property. The scheme efficiently handles both single-gateway and multi-gateway scenarios as well as addressing candidate links with various capacities.

Finally, we evaluate the practical effectiveness of our proposed solutions through extensive simulations. In our simulations we consider two models of full graphs, one is the UDG model and the other is a *Quasi-UDG model*, which is an extension of the UDG model commonly considered as close enough to reality [27]. For both models, we evaluate both achieved bound on graph degree and the hop-count distance from the gateway(s). We observed impressive results for these two metrics for both graph models. Our simulation results demonstrate that using at most four directional mesh radios per node and less than 10% of nodes as gateways, we are able to build robust mesh topologies where each node is at most 3 hops from some gateway in the UDG model and at most 4 hops from some gateway in the Quasi-UDG model.

II. Preliminaries

A. The Network Model

In this work we consider a nomadic wireless mesh network (NWMN) comprised of wireless routers deployed on vessels or vehicles. Each wireless router is equipped with $K+1$ mesh radios that K of them are attached to directional antennas and are used for establishing high-capacity point-to-point data connections with adjacent nodes, while the last mesh radio is attached to an omni-directional antenna and is used only for management proposes, as proposed by BelAir Networks [3]. The vessels may change their locations for meeting application needs. Each time a vessel changes its location, its attached wireless router utilizes its omni-directional mesh radio to discover adjacent routers. Then, it sends the neighborhood information to decision points (one or several) that determine the network topology and update the nodes about the selected point-to-point connections. Evidently, the transmission range of a directional antenna is significantly longer than the one of the omni-directional antenna. However, since we interested only in establishing high capacity point-to-point connections, we assume that the potential neighbors of a node are confined in the transmission range of its onmi-directional antenna, while control traffic can be send over low capacity links.

In this study we focus on the design of topology selection algorithms executed by decision points. To this end, we denote by $F(V, E_F)$ the *full graph* that represents all the possible point-to-point connections, where the set of nodes V denotes the wireless routers and the set of edges E_F specifies all the possible point-to-point connections. Recall that in the full graph $F(V, E_F)$, the degree of a node may be more than K.

B. Problem Statement

In this work, we would like to find a subgraph $G(V, E)$ of $F(V, E_F)$, such that $E \subseteq E_F$, the degree of any node in G is at most K and G still satisfies some connectivity requirements. In particular, we consider *two-edge-connectivity* that guarantees two edge-disjoint paths between every pair of nodes, and *two-node-connectivity* that provides two node-disjoint paths between every pair of nodes. Our objectives are formally defined as follows.

Definition 1: (Bounded Degree Two-Edge-Connectivity) Given a two-edge-connected full graph $F(V, E_F)$, find a minimal integer K_{2EC} and a polynomial-time algorithm that finds a two-edge-connected subgraph $G(V, E)$ with maximal node degree no larger than K_{2EC}.

Definition 2: (Bounded Degree Two-Node-Connectivity) Given a two-node-connected full graph $F(V, E_F)$, find a minimal integer K_{2NC} and a polynomial-time algorithm that finds a two-node-connected subgraph $G(V, E)$ with maximal node degree no larger than K_{2NC}.

[3] Obviously, the performance of wireless links is time varying. Thus, we consider only candidates links that provide high capacity in the long run.

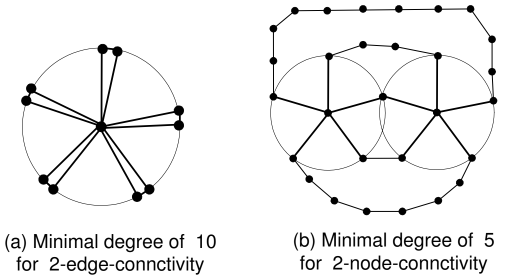


Fig. 3. Lower bounds on K_{2EC} and K_{2NC} in UDGs.

Obviously, for some full graph topologies there are subgraphs with degree as low as 2 that preserve the connectivity requirements, i.e., when the full graph contains a Hamiltonian cycle [18]. However, our objective is to find two minimal constants K_{2EC} and K_{2NC} that ensure the existence of twoedge/node connected subgraphs with degree bounds of K_{2EC} and K_{2NC}, respectively, *regardless of the full graph topology*. As explained in Sub-section I-B, in the case of arbitrary full graph, the required number of directional antennas may be as high as the number of the graph nodes minus 2. Thus, *just for the analysis of the degree bound*, we assume the use of identical mesh nodes with the same transmission range R on all directions. Thus, we model the full graph as a *unit disk graph* (UDG) as defined in [21]. More specifically, the nodes are considered as points in the plane and two nodes are adjacent if and only if the (Euclidean) distance between them is at most R, where R is taken as the unit distance. Recall that our algorithms guarantee that the calculated subgraph are two-connected for any arbitrary full-graph that satisfies the connectivity requirements.

Beside topology control, appropriate channel allocation that eliminates interferences is another critical requirement for improving the network performance. In this study we assume enough channels such that adjacent point-to-point connections can easily associated with non-interfering channels, *e.g.*, Blair [3] solution is based on IEEE 802.11-a technology with at least 12 non-interfering channels. The problem of appropriate channel allocation when the number of available channels is small is beyond the scope of this paper.

C. Properties of Unit Disk Graphs

We now turn to present lower bounds for K_{2EC} and K_{2NC}. These upper bounds are based on the following fundamental property of UDGs, proved in [28].

Property 1: Consider any node u of any given UDG and let $H(V, E)$ be the subgraph induced by u, all its neighbors and the edges between them. Then, the maximum independent set of H has at most 5 nodes, *i.e.,* any subset of V of size 6 or more contains neighboring nodes.

By using Property 1, we illustrate in Figure 3 that the lower bounds on K_{2EC} and K_{2NC} in UDGs are 10 and 5, respectively. Figure 3-(a) demonstrates that at least degree 10 is needed for preserving two-edge-connectivity of some UDGs. In this example, 5 node pairs are evenly placed along the fringe of the unit disk centered at the central node. Thus, every node is adjacent to its peer node and the central node but not adjacent to any other node. To provide two-edge-connectivity, the central node has to be connected to the other 10 nodes, resulting in a degree of 10 on the central node. Figure 3-(b) shows that at least degree 5 is required to maintain two-node-connectivity of the considered graph. Again, the 5 nodes along the fringe of each unit disk are even distributed. Two nodes are connected if and only if their Euclidean distance is at most 1. The central nodes of the unit disks both have degree 5. In this example any removing of a link results with a subgraph that is not two-node connected.

III. Topology Control Algorithms

A. Scheme Overview

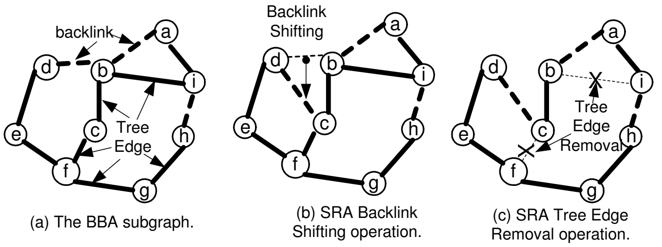
In this section, we present efficient two-phase scheme for building bounded degree two-connected subgraphs. At the first phase, the scheme builds two-connected sparse subgraphs. In Section III-B, we present a simple Backlink-Based Algorithm (BBA for simplicity) for building a subgraph G that preserves the two-connectivity of the full graph (either two-edge-connectivity or two-node-connectivity). First the algorithm calculates a Depth-First-Search (DFS) tree which has a maximal node degree of at most 6, as prove below. Then, for ensuring two connectivity, the scheme calculates a subgraph G that contains the links of the DFS tree and non-nested backlinks. If the full graph is a UDG, the maximal node degree of the subgraph G we build is at most 10. As we have discussed in Section II, this is the minimal node degree that is required to maintain the two-edge-connectivity of some UDG topologies. Then, in Section III-C, for two-node-connected full graphs, we present a backlink Shifting and tree edge Removal Algorithm (SRA for simplicity) for post-processing the two-node-connected subgraph calculated by BBA. The resulting subgraph G is still two-node-connected, and if the full graph is a UDG, the maximal node degree of the resulting subgraph G is at most 6. At the second phase, described in Section III-D, the scheme augments the calculated subgraph with additional links for reducing the hop-count distance of the nodes to the gateways, while preserving the degree bound and connectivity property. Then, in Section III-E we extend the scheme to handle networks with multiple gateways. Finally, in Section III-F we consider the case where the links have different capacity and we briefly describe some techniques for improving the quality of the calculated subgraph. Due to space constraint only sketches of the proofs are provided.

B. The BBA Algorithm

We start with the following key property of *Depth-First-Search* (DFS) trees that is central to the design of our algorithms.

Lemma 1: If two nodes u and v are adjacent in the full graph, then in any DFS tree of the full graph, either u is v's ancestor or vice versa.

Proof: Without loss of generality, assume that during an execution of the DFS algorithm, u is visited before v is visited.



(a) The BBA subgraph.

(b) SRA Backlink Shifting operation.

(c) SRA Tree Edge Removal operation.

Fig. 4. A subgraph calculated by the BBA and SRA algorithms.

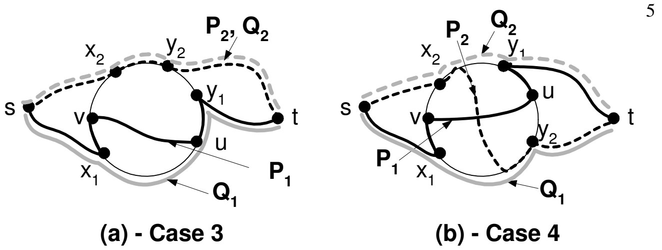


Fig. 5. Figures for the proof of Lemma 2.

It u is the root, our conclusion trivially follows. If u is not the root, it is clear from the definition of DFS that when the DFS execution backtracks from u to its parent, all nodes adjacent to u (including v) must have been visited by the DFS execution. Thus from the definition of DFS follows that v resides in the sub-tree rooted at u. In other words, u is v's ancestor.

From Lemma 1 and Property 1 follows that,

Corollary 1: The children of each node form an independent set, which contains at most 5 nodes in UDGs. Therefore, the degree of any DFS tree is at most 6.

We now provide two definitions that are essential for our algorithms.

Definition 3 (Backlink): Let T be a DFS tree of the full graph $F(V, E_F)$. We refer to each edge $(u, v) \in E_F$ such that $(u, v) \notin T$ as a backlink of T. By Lemma 1, u is either an ancestor or a descendant of v in T. We refer to u as a backlink neighbor of v, and vice versa.

Definition 4 (Nested Backlink): Consider a backlink (u, v) of T, where u is an ancestor of v. We refer to (u, v) as a nested backlink if there is another backlink (u', v') of T, where u' is an ancestor of v', such that (1) Either $u = u'$ or u' is an ancestor of u in T. (2) Either $v = v'$ or v' is a descendant of v in T.

The BBA algorithm is a simple iterative algorithm that starts with an initial subgraph G that is identical to the full graph. The BBA algorithm first calculates a DFS tree T of the full graph, typically, rooted by a gateway node. Then iteratively removes a nested backlink of T from the residual graph G until G does not contain any nested backlink of T.

Figure 4-(a) illustrates as an example a subgraph calculated by the BBA algorithm for the full graph depicted in Figure 1-(a). In our examples (Figure 1) the DFS-tree is rooted at node a, the tree links are denoted by solid bold lines, while backlinks are denoted by dashed bold lines. For instance, at this example the links (c, e), (c, d), (b, f), (d, f) and (e, b) are nested links by the link (d, b) and have been removed from the calculated subgraph.

Below, we prove that for a UDG the BBA algorithm finds two-node/edge connected subgraph with node degree at most 10. The proof comprised of the following two steps: First, in Theorem 1 we prove that it is sufficient to keep only the non-nested backlinks to preserve the two-connectivity property of the graph. Then, we prove the degree bound. Theorem 2 proves that the set of backlink neighbors of any node v and its parent yield an independent set with at most 5 nodes, by using Property 1. In addition, from Corollary 1 follows that a node may have at most 5 children, thus a node degree is bounded by 10.

Connectivity Analysis

We start with the following key observation.

Lemma 2: Consider a two-node-connected (two-edge- connected) graph $H(V, E)$ and an edge $e = (u, v) \in E$. If H contains a cycle C that contains both nodes v and u without the edge e, then the subgraph $H'(V, E - \{e\})$ is also two-node-connected (two-edge-connected).

Proof: We prove this lemma just for two-node-connected graphs, the proof of two-edge-connectivity is very similar. Assuming that H is two-node-connected, we just need to prove that H' also contains two node-disjoint paths between every pair of nodes. Consider any node pair $s, t \in V$ and we denote by P_1 and P_2 the two node-disjoint paths between s and t in H.

Case 1: If neither P_1 nor P_2 contains e, then both of them are present in H' and we are done.

In the following let us assume that e is included in P_1.

Case 2: If P_2 does not contain any node in C then there is an alternative path to P_1 that contains few links of C.

In the following we assume that P_2 also contains some node(s) in C and we calculate two new node-disjoint paths Q_1 and Q_2 between s and t in H'. Let us denote by x_i and y_i the two nodes in both P_i, $i \in \{1, 2\}$, and C that are the closest to s and t in P_i, respectively. Moreover, let us denote by $P_i^{w,z}$ the segment of P_i between the nodes w and z.

case 3: Assume that C contains two node-disjoint segments C_1 and C_2 that connect x_1, y_1 and x_2, y_2, respectively. Thus, two node-disjoint paths between s and t in H' can be defined as $Q_1 = P_1^{s,x_1}, C_1, P_1^{y_1,t}$ and $Q_2 = P_2^{s,x_2}, C_2, P_2^{y_2,t}$, as illustrated in Figure 5-(a).

case 4: Assume that C does not contain two node-disjoint segments between x_1, y_1 and x_2, y_2, respectively. Thus, in this case C must contain two node-disjoint segments, C_1' and C_2' that connect x_1, y_2 and x_2, y_1, respectively. In this case, two node-disjoint paths between s and t in H' are $Q_1 = P_1^{s,x_1}, C_1', P_2^{y_2,t}$ and $Q_2 = P_2^{s,x_2}, C_2', P_1^{y_1,t}$, as illustrated in Figure 5-(b). This completes our proof.

Theorem 1: If the full graph $F(V, E_F)$ is two-node-connected (two-edge-connected), BBA calculates a subgraph $G(V, E)$ that is also two-node-connected (two-edge-connected).

Proof: This is a direct result from Lemma 2. A backlink (u, v) is removed only if it is nested by other Backlink, denoted by (x, y). Consider the cycle C that contains the backlink (x, y) and all the DFS tree links between nodes x and y, also contains the nodes u and v. By using Lemma 2, the backlink (u, v) can be removed without violating the connectivity requirement and this completes our proof.

Degree Bound Analysis

To derive an upper bound on the maximal node degree in G, recall that the children of any node from an independent set (i.e., Corollary 1). Furthermore, since G contains no nested backlink, we show that the parent and backlink neighbors of any node also form an independent set in G. By using Property 1 it can be proved that,

Lemma 3: If the full graph is a UDG, each node is incident to at most 5 backlinks in G.

Theorem 2: If the full graph is a UDG, the maximal node degree of G is at most 10.

Proof: If v is the root, by Corollary 1 we know that it has at most 5 children in T. From Lemma 3 results that v has degree 10 or less in G.

If v is not the root, given Lemma 3, v has a degree larger than 10 in G only if v has 5 children as well as 5 backlink neighbors in G. To prove by contradiction, let us assume that is the case and that the parent of v is u. At least one of v's backlink neighbors, say b_i, is adjacent to u. (1) If b_i is a descendant of v, the backlink (b_i, v) is nested given the existence of (b_i, u), which contradicts the fact that (b_i, v) is a non-nested backlink. (2) If b_i is an ancestor of v, it follows that there must be some child w of v that is adjacent to b_i. Therefore, (b_i, v) is nested given the existence of (b_i, w), which also contradicts the fact that (b_i, v) is a non-nested backlink.

C. The SRA Algorithm

Compared with the BBA algorithm, the SRA algorithm further reduces node degree by making the children and backlink neighbors of each node form an independent set in the resulting subgraph. In particular, the SRA algorithm consists of three rounds. In the first round it calculates a DFS tree and obtains a two-node-connected subgraph $G'(V, E)$, using the BBA algorithm. In the second round, the algorithm performs a top-down visit of the DFS tree nodes and performs *Backlink Shifting* operation on some nodes. In the third round, the algorithm performs another visit of the DFS tree nodes and performs *Tree Edge Removal* operation on some nodes. These operations reduce the degree of nodes in G'. Throughout the entire degree reduction process in the second and third round, the SRA algorithm always preserves the following two Reservations.

I. G' does not contain any nested backlink.

II. G' is a two-node-connected subgraph that connects all the nodes in V.

Essentially, during the top-down round (i.e., second round) and the bottom-up round (i.e., third round) we reduce the degree of a node v if its combined set (of children and backlink neighbors) in the subgraph G' is not an independent set. As result from Lemma 9 below, if G' is two-node connected then the degree of the root node is at most 6. Thus, in the following we perform degree reduction operations only for non-root nodes and we denote by $\mathcal{T}_x$ the subtree of the DFS-tree rooted by node x.

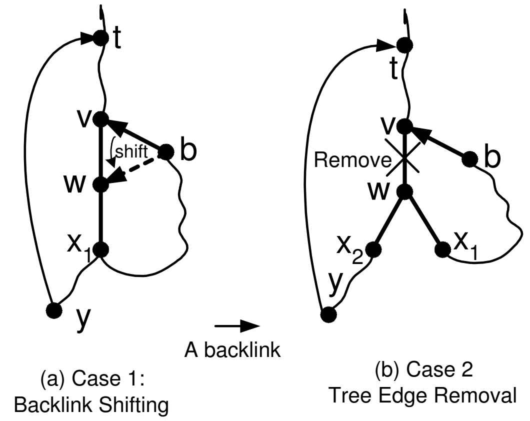


Fig. 6. The two cases for reducing node degree.

Consider a non-root node v with degree higher than 6. As we have discussed, the children of v form an independent set and the backlink neighbors of v form an independent set as well. Thus, it has to be the case that v has a child w and a backlink neighbor b in G' such that b is adjacent to w in G'. Since b is adjacent to w, by Lemma 1, b must be a descendant of both w and v in T. Let us assume that b resides in the subtree $\mathcal{T}_{x_1}$ rooted at some child x_1 of w in T. We distinguish between two cases.

case 1 (Backlink Shifting): Lets assume that G' contains at least one backlink between some node y in $\mathcal{T}_{x_1}$ and some ancestor t of v, as illustrated in Figure 6-(a). Thus, the nodes v, w and x_1 are included in two cycles. The first cycle contains the nodes $\{v, w, x_1, y, t\}$, while the second cycle contains the nodes $\{v, w, x_1, b\}$. Consequently, the nodes included in these cycles induce a two node connected component. In this case, *we reduce the degree of node v during the top-down round by replacing (v, b) with (w, b)*, referred to as *Backlink Shifting* operation. After this operation node v is included just in the first cycle but nodes w and x_1 are still included in both cycles. Thus, after the backlink shifting operation the nodes in the two cycles still induce a two node connected component. As we prove in Lemma 4 below, this is sufficient to preserve the two-node-connectivity of G'. Finally, if (w, b) is nested, we remove it to preserve Reservation I.

Case 2 (Tree Edge Removal): Unlike case 1, lets assume that G' does not contain any backlink between a node in $\mathcal{T}_{x_1}$ and any ancestor of v. Since, G' is two-node connected, node v is not a cut node. Thus, node w must have another child x_2 such that G' contains at least one backlink between some node y in $\mathcal{T}_{x_2}$ and some ancestor t of v, as illustrated in Figure 6-(b). Recall that G' contains two cycles. The first cycle C_1 contains the nodes $\{v, w, x_1, b\}$, while the second cycle C_2 contains the nodes $\{t, v, w, x_2, y\}$. The two cycles share only the nodes v, w and the edge between them. Consequently, G' contains a "big" cycle C that comprises all the edges of C_1 and C_2 beside the edge (v, w). *We reduce the degree of node v during the bottom-up round by removing* (v, w), referred to as *Tree Edge Removal* operation. This operation preserves the cycle C and thus all the nodes in the cycles C_1 and C_2 still included in a two-node connected component. As we prove in Lemma 7 below, this is sufficient to preserve the two-node-connectivity of G'.

Figures 4-(b) and 4-(c) demonstrate executions of Backlink Shifting and Tree-Edge Removal operations, respectively, on the graph G' depicted in Figure 4-(a). Even, in this simple example Backlink Shifting operations reduced the subgraph degree from 4 to 3 (the degree of node b), while Tree-Edge Removal operations further reduce the subgraph degree from 3 to 2 (nodes b, c, f and i). Consequently the scheme produced the optimal solution depicted in Figure 1-(b).

Connectivity Analysis

Lemma 4: The backlink shifting operation preserves the two-node-connectivity of G'.

Proof: In the resulting subgraph G', a loop C containing v and b is composed of the following segments: (1) the tree path between v and t, (2) the backlink (t, y), (3) the tree path between y and b, (4) the backlink (b, w) and (5) the tree edge (w, v). Since C does not contain the removed backlink (v, b), from Lemma 2 it follows that G' remains two-node-connected after the backlink shifting operation.

We now turn to prove that the tree edge removal operation also preserves the two-node-connectivity of G'. Our proof is based on the same argument provided above, thus to meet space limitation we provide some auxiliary claims without proofs.

Lemma 5: After removing (v, w), G' contains a path P_1 from x_1 to w and a path P_2 from x_1 to v such that P_1 and P_2 are node-disjoint and contain only nodes in $\mathcal{T}_{x_1}$ (except v and w).

Lemma 6: After removing (v, w), G' contains a path P_0 between w and some ancestor t of v such that P_0 does not contain v and any node in $\mathcal{T}_{x_1}$.

Lemma 7: The tree edge removal operation preserves the two-node-connectivity of G'.

Proof: In the resulting subgraph G', a loop C containing v and w is composed of the following segments: the tree path between v and t, P_0, P_1 and P_2. Since C does not contain the removed backlink (v, w), from Lemma 2 it follows that G' remains two-node-connected after the tree edge removal operation.

Lemma 8: The resulting graph is two-node-connected.

Proof: The initial graph is two-node-connected. From Lemmas 4 and 7 follow that the graph remains two-node-connected also after each backlink shifting and tree edge removal operation.

Degree Bound Analysis

Lemma 9: If a UDG is two-node-connected, then the root of any DFS tree has only one child.

Proof: To prove by contradiction, assume that the root has two children u and v. By Lemma 1, there is no edge crossing $\mathcal{T}_u$ and $\mathcal{T}_v$. Therefore, every path between u and v must traverse the root. This contradicts the fact that the unit disk graph is two-node-connected.

Theorem 3: The final sub-graph is two-node-connected and its maximal node degree is 6 or less.

Proof: By Lemma 9, the root has at most one child and hence at most degree 6 in G', since each node has at most 5 backlink neighbors. For any non-root node v, note that after the degree reduction process, its children and backlink neighbors cannot be adjacent to each other. Therefore, v has at most 5 children or backlink neighbors in total. It follows that v has at most degree 6 in G'.

D. Link Augmentation

In previous sections, we have presented simple DFS-based algorithms for building a bounded degree subgraph G that satisfies two-edge-connectivity and two-node-connectivity. While the DFS-based approach possesses some fundamental properties that allow us to achieve the proved degree bounds, it tends to build deep trees where some nodes are many hops away from the root (gateway). Besides degree bound and two-connectivity, end-to-end delay and loss rate are also important performance metrics of wireless communication networks. To improve these performance metrics, it is often preferable to use short paths for communication. In this paper, we also evaluate the quality of a calculated mesh topology (i.e., subgraph) G in terms of the average length of the shortest paths (in hops) between the gateway and individual nodes. For each node v, we use l_v to denote the length of the shortest path between v and the gateway in the calculated subgraph G. Our objective is thus to minimize $L_G = \sum_{v \in V} l_v$.

Without exceeding the achieved degree bound, we can still decrease L_G by augmenting the calculated subgraph G with additional links. In a greedy manner, we iteratively add additional "shortcut" links to G to minimize L_G in the augmented subgraph. In each iteration, we augment G with one additional link such that the maximal node degree in G is not increased while L_G is maximally decreased. For simplicity, we refer to this optimization as *augmentation*. Augmentation can be applied on any subgraph G we build.

E. Multiple Gateways

In such case, we would like construct a subgraph G that ensures two-edge/node disjoint paths from every non-gateway node to two distinct gateways. To this end, we augment the full graph with a "super gateway" node, which is adjacent to all gateways but not adjacent to any other node. If the augmented full graph is two-edge/node-connected, we can apply BBA and SRA as usual to build a two-edge/node-connected subgraph of the augmented full graph. In the subgraph, each non-gateway node v has two edge-disjoint (node-disjoint) paths P_1 and P_2 to the super gateway. Since the super gateway is only adjacent to gateways, its predecessors on P_1 and P_2 are both gateways, denoted by g_1 and g_2. Thus, removing the super gateway from

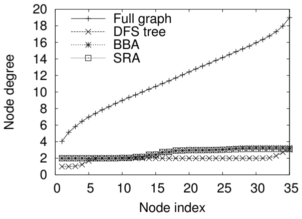


Fig. 7. Maximum node degree (UDG).

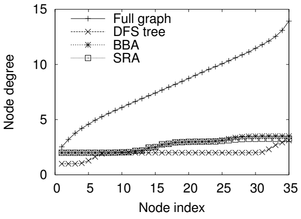


Fig. 8. Maximum node degree (Quasi-UDG).

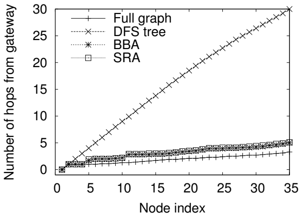


Fig. 9. Distance to the gateway (UDG).

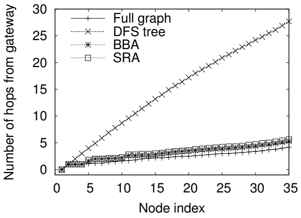


Fig. 10. Distance to the gateway (Quasi-UDG).

P_1 and P_2 gives us two Edge/node-disjoint paths from v to g_1 and g_2. As to node degree, it is not hard to verify that after removing the super gateway, the degree bound analysis of BBA and SRA apply as well. Therefore, the degree bounds of UDGs still hold.

F. Links with Various Capacities

We now turn to the cases where the candidate links have different capacities. In such case, obviously, we prefer the links with the highest capacity over the other. However, the connectivity constraints may enforce us to select also candidate links lower capacity. Yet, the proposed scheme provides us several degrees of freedom when selecting the links and we briefly describe few or them.

The initial full graph: we select an initial full graph that contains only highest-capacity candidate links required for satisfying the required connectivity and other requirements such as hop-count distance. For instance, the following process maximizes the minimal link capacity used by the selected subgraph G. We start with an empty full graph F We and we sort the links in non-increasing order according to their capacity. Then we add links to the full graph according to their order until the required connectivity constraints is satisfy.

The DFS tree: The scheme may start with any DFS tree. Clearly, a DFS tree that contains high-capacity links will yield a subgraph with higher capacity.

Link augmentation: The link augmentation enables us to strike a balance between various objectives, for instance the subgraph capacity versus the average hope count distance to the nearest gateway.

IV. Evaluation

In this section we evaluate the performance of the BBA and SRA algorithms. We consider two metrics. The first is the obtained *degree bound*, while the second is the *hop-count distance* of the nodes from the gateway(s). Since, this is the first study that calculates two-node connected subgraphs with guaranteed small upper bound on the node degrees, we use the following benchmarks to evaluate our results. We compare our calculated subgraphs with the DFS-trees of the full graphs. Since, DFS-trees efficiently find low-degree connected subgraph (but *not* two connected subgraphs), we use them as a benchmark for evaluating the degree bound. Then, we compare the hop-count distances achieved by our algorithms with the hop-count distances of the full graphs. Since our algorithms calculate subgraph of the full graphs, the hop-count distances of the full graphs are the optimal hop-count distances that can be achieved by our algorithms. In our simulations we consider both UDG model and a more generalized *Quasi-UDG model*, which is commonly considered as close enough to reality while being concise enough [27]. In the Quasi-UDG model, nodes are also placed in the plane. Two nodes $u, v \in V$ are not adjacent if their Euclidean distance $d_{uv} > 1$ and are adjacent if $d_{uv} \leq r$, where $r \in [0, 1]$ is a constant of the Quasi-UDG model. In the undecided case where $r < d_{uv} \leq 1$, we define the probability of u and v being adjacent as $1 - \frac{d_{uv} - r}{1 - r}$. The UDG model is a special case of the Quasi-UDG model, where $r = 1$.

We conduct extensive simulations on thousands of randomly generated trial networks with different topologies and various number of nodes. Due to space limitations, we present only typical results that have been obtained for networks with 35 nodes. We simulated one thousand (valid) randomly generated trial networks. Each trial network is obtained by randomly deploying 35 identical mesh Nodes, each one with transmission range of 1, within a 2.5×2.5 square area and it was accepted as a valid one only if it satisfied the requested two-connectivity. Then, one of the nodes is randomly chosen as the gateway. In the quasi-UDG model, r is chosen to be 0.6. Through simulation and analysis, we find this network setting both challenging and practically interesting. We found that the node density was sufficient to ensure the required two-connectivity of the full graphs. Yet, the node density is sparse enough, thus finding low-degree two-connected subgraph is non-trivial. For example, in the extreme case where the mesh nodes form a clique, any DFS tree is a chain and the two-connected mesh topology is ring, where the maximal node degree is only 2.

A. Effectiveness of the proposed algorithms

Figure 7-10 present the average node degree and hop-count distance to the gateway in the full graphs, the DFS trees and the subgraphs calculated by the BBA and SRA algorithms. We report for each node v its degree and hop-count distance to the gateway, denoted by l_v. For ease of comparison, for each trial network we sort the nodes in non-decreasing order of their degree and distance to the gateway. We then report the average of the sorted node degree vectors and distance vectors of the one thousand (valid) trial networks.

From these figures, we can see that DFS trees tend to be very deep and hence many nodes are many hops away from the gateway in DFS trees, which is consistent with our discussion in Section III-D. As additional backlinks are included by BBA and SRA to form a two-connected subgraph[4] the distance between the gateway and individual nodes is very close to their distance in full graphs and is significantly smaller then their hop-count distances along the DFS-trees (by up to a factor of $5 \sim 6$ on average). However, the node degree has been significantly reduced (by up to a factor of $5 \sim 6$ as well), compared with full graphs.

As we have discussed in Section III, SRA is designed to further reduce the maximum node degree of the subgraph built by BBA. Now we demonstrate the advantages of the SRA algorithm by presenting the distribution of maximal node degree (i.e., the subgraph degree) in the two-connected subgraphs over the one thousand trial networks. The distributions observed in the UDG model and the Quasi-UDG model are presented in Figure 11 and Figure 12, respectively. As we can see, the maximal node degree almost never exceeds 4. Moreover, we can clearly observe that SRA tends to build subgraphs of lower maximum degrees than the subgraphs built by BBA.

Instead of average distance, in some cases it is also important to bound the maximal distance between the gateway and individual nodes. Taking 4 as the degree bound, we augment the calculated subgraphs with additional links such that the maximal node degree will not exceed 4 (due to the augmentation). We present the distribution of maximal distance between the gateway and individual nodes in the augmented subgraphs over the one thousand trial networks. The distributions observed in the UDG model and the Quasi-UDG model are presented in Figure 13 and Figure 14, respectively. As we can see, the maximal distance between the gateway and individual nodes almost never exceeds 5 hops in the UDG model and never exceeds 6 hops in the Quasi-UDG model.

B. Multi-gateway

Our evaluation so far has been based on the conservative assumption that the mesh network contains only one gateway. Although having more gateways in the mesh network will not affect node degrees, it clearly helps reduce the distance from each node to the closest gateway. Using 4 as the degree bound, we also conduct simulations with three gateways and present the results in Figure 15 and Figure 16. With three randomly chosen gateways, the maximal distance between individual nodes and the closest gateway almost never exceeds 3 hops in the UDG model and almost never exceeds 4 hops in the Quasi-UDG model.

V. Conclusion

In this paper, we study the problem of building robust topologies for nomadic wireless mesh networks, where the degree of any node does not exceed its number of available directional mesh radios. Our scheme is shown to build efficient mesh topologies while satisfying practically interesting connectivity requirements and degree bounds. Currently, the number of directional mesh radios installed on each node is chosen in a somewhat ad hoc manner. We believe the results reported in this paper provide a guideline on the number of directional mesh radios that should be installed on each node for building resilient and efficient mesh topologies.

[4] Especially after augmentation, without increasing maximum node degree.

References

[1] R. Bruno, M. Conti, and E. Gregori, Mesh Networks: Commodity Multihop Ad Hoc Networks. *IEEE Com. Magazine*, Vol. 43, No. 3, pp 123–131, March 2005.

[2] I. F. Akyildiz, X. Wang and W. Wang, A survey on wireless mesh networks, *IEEE Com. Magazine*, Vol. 43, No. 9, pp S23–S30, September 2005.

[3] “BelAir Networks,” http://www.belairnetworks.com/.

[4] “Mesh Dynamics,” http://www.meshdynamics.com/.

[5] “Tropos Networks,” http://www.tropos.com/.

[6] A. Alexiou and M. Haardt. Smart Antenna Technologies for Future Wireless Systems: Trends and Challenges. *IEEE Com. Magazine*, Vol. 42, No. 9, pp 90–97, September 2004.

[7] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit. Ad Hoc Networking With Directional Antennas: A Complete System Solution. *IEEE JSAC*, Vol. 23, No. 3, March 2005.

[8] G. Li, L. Yang, W. S. Conner and B. Sadeghi. Opportunities and Challenges in Mesh Networks Using Directional Antennas. In *WiMesh*, 2005.

[9] “BelAir200,” http://www.belairnetworks.com/products/ba200.cfm.

[10] R. R. Choudhury and N. H. Vaidya. Deafness: A MAC Problem in Ad Hoc Networks when using Directional Antennas. In *ICNP*, 2004.

[11] K. H. Grace, J. A. Stine and R. C. Durst: An Approach for Modestly Directional Communications in Mobile Ad Hoc Networks. *Telecommunication Systems* 28(3-4): pp 281–296 (2005).

[12] R. R. Choudhury and N. H. Vaidya. Performance of Ad Hoc Routing using Directional Antennas. *Journal of Ad Hoc Networks*, November 2004, Elsevier Publishers.

[13] S. Bandyopadhyay, D. Saha, S. Roy, T. Ueda and S. Tanaka. A Network-Aware MAC and Routing Protocol for Effective Load Balancing in Ad Hoc Wireless Networks with Directional Antenna. In *ACM Mobihoc*, 2003.

[14] Y. Li, H. Man, J. Yu and Y. Yao. Multipath routing in ad hoc networks using directional antennas. Advances in Wired and Wireless Communication, *2004 IEEE/Sarnoff Symposium*, April 2004, pp 119–122.

[15] Z. Huang and C.-C. Shen, Multibeam Antenna-Based Topology Control with Directional Power Intensity for Ad Hoc Networks. *IEEE TMC*, Vol. 5, No. 5, May 2006.

[16] Ning Li and Jennifer C. Hou, FLSS: a fault-tolerant topology control algorithm for wireless networks, in *ACM MobiCom*, 2004.

[17] F. Dai, Q. Dai and J. Wu. Power efficient routing trees for ad hoc wireless networks using directional antenna *Journal of Ad Hoc Networks 3*, 2005, pp 621–628.

[18] M. R. Garey and D. S. Johnson, *Computers and Intractability : A Guide to the Theory of NP-completeness*. Freeman, 1979.

[19] M. Fürer and B. Raghavachari, “Approximating the minimum-degree Steiner tree to within one of optimal,” *Journal of Algorithms*, vol. 17, pp. 409–423, 1994.

[20] P. N. Klein, R. Krishnan, B. Raghavachari, and R. Ravi, “Approximation algorithms for finding low-degree subgraphs,” *Networks*, vol. 44, no. 3, pp. 203–215, 2004.

[21] B. N. Clark, C. J. Colbourn, and D. S. Johnson, “Unit disk graphs,” *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.

[22] Y. Wang and X.-Y. Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. In *DIALM-POMC*, 2003. ACM Press.

[23] N. Li, J. C. Hou, and L. Sha. Design and analysis of a MST-based distributed topology control algorithm for wireless ad-hoc networks. In *IEEE INFOCOM*, 2003.

[24] U. Kumar, H. Gupta and S. R. Das. A Topology Control Approach to Using Directional Antennas in Wireless Mesh Networks. In *IEEE ICC*, 2006.

[25] W. Wu, H. Du, X. Jia, Y. Li, and S. C.-H. Huang, “Minimum connected dominating sets and maximal independent sets in unit disk graphs,” *Theoretical Computer Science*, vol. 352, no. 1-3, pp. 1–7, 2006.

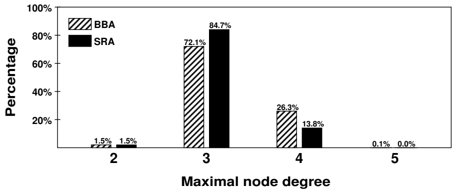


Fig. 11. Maximal node degree (UDG).

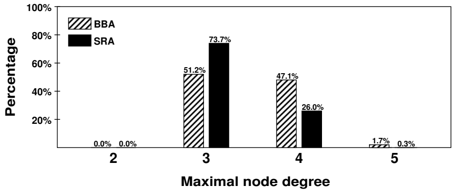


Fig. 12. Maximal node degree (Quasi-UDG).

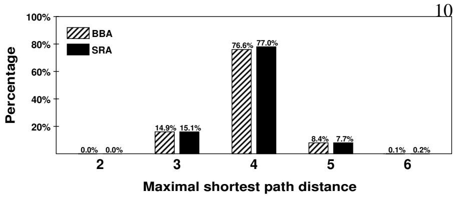


Fig. 13. Maximal distance to the gateway (UDG).

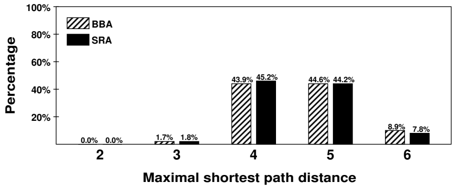


Fig. 14. Maximal distance to the gateway (Quasi-UDG).

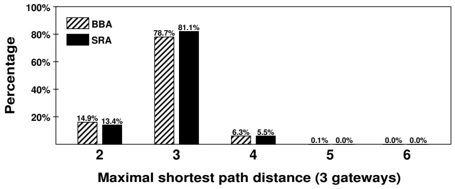


Fig. 15. Maximal distance to the gateways (UDG), using 3 gateways.

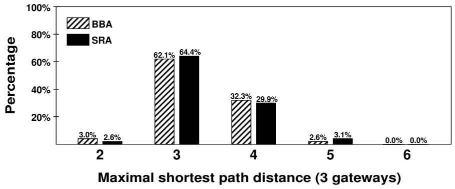


Fig. 16. Maximal distance to the gateways (Quasi-UDG), using 3 gateways.

[26] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter, “Hamilton paths in grid graphs,” *SIAM Journal on Computing*, vol. 11, no. 4, pp. 676–686, November 1982.

[27] F. Kuhn, R. Wattenhofer, and A. Zollinger, “Ad-hoc networks beyond unit disk graphs,” In *DIALM-POMC*, 2003.

[28] S. Banerjee and S. Khuller, “A Clustering Scheme for Hierarchical Control in Wireless Networks,” In *IEEE INFOCOM*, 2001.