Spring 2004

A clustered back-bone for routing in ad-hoc networks

Delzad Kothawala
New Jersey Institute of Technology

ABSTRACT

A CLUSTERED BACK-BONE FOR ROUTING IN AD-HOC NETWORKS

by
Delzad Kothawala

In the recent years, a lot of research work has been undertaken in the area of ad-hoc networks due to the increasing potential of putting them to commercial use in various types of mobile computing devices. Topology control in ad-hoc networks is a widely researched topic; with a number of algorithms being proposed for the construction of a power-efficient topology that optimizes the battery usage of the mobile nodes.

This research proposes a novel technique of partitioning the ad-hoc network into virtually-disjoint clusters. The ultimate aim of forming a routing graph over which power-efficient routing can be implemented in a simple and effective manner is realized by partitioning the network into disjoint clusters and thereafter joining them through gateways to form a connected, planar back-bone which is also a t-spanner of the original Unit Disk Graph (UDG). Some of the previously proposed algorithms require the nodes to construct local variations of the Delaunay Triangulation and undertake several complicated steps for ensuring the planarity of the back-bone graph. The construction of the Delaunay Triangulation is very complex and time-consuming. This work achieves the objective of constructing a routing graph which is a planar spanner, without requiring the expensive construction of the Delaunay Triangulation, thus saving the node power, an important resource in the ad-hoc network. Moreover, the algorithm guarantees that the total number of messages required to be sent by each node is O(n). This makes the topology easily reconfigurable in case of node motion.

A CLUSTERED BACK-BONE FOR ROUTING IN AD-HOC NETWORKS

by
Delzad Kothawala

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

Department of Computer Science

May 2004

Blank Page

APPROVAL PAGE

A CLUSTERED BACK-BONE FOR ROUTING IN AD-HOC NETWORKS

Delzad Kothawala

Dr. Artur Czumaj, Thesis Advisor — Date
Associate Professor of Computer Science, NJIT

Dr. Wojciech Rytter, Committee Member — Date
Professor of Computer Science, NJIT

Dr. David Nassimi, Committee Member — Date
Associate Professor of Computer Science, NJIT

BIOGRAPHICAL SKETCH

Author: Delzad Kothawala

Degree: Master of Science

Date: May 2004

Undergraduate and Graduate Education:

- Master of Science in Computer Science,
  New Jersey Institute of Technology, Newark, 2004

- Bachelor of Engineering in Information Technology,
  G. H. Patel College of Engineering and Technology, Gujarat, India, 2002

Major: Computer Science

To my beloved family

ACKNOWLEDGEMENT

I would like to express my deepest appreciation and gratitude to Dr. Artur Czumaj, my thesis advisor, who not only served as a good mentor, guiding me all along but also made me believe that I could make it happen. He made me understand the importance of reasoning and proofs in research. I would also like to express my special thanks to Dr. Wojciech Rytter and Dr. David Nassimi for actively participating in the review committee.

Last but not the least; I would also like to express my sincere thanks to Ms. Hairong Zhao, a PhD student with the Computer Science Department at NJIT for her valuable guidance and insight.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of this dissertation is to undertake a detailed survey of the research problems currently being studied in the area of ad-hoc networks, particularly those related to the construction of a sparse power-efficient topology and routing in ad-hoc networks.

This research also proposes the distributed construction of a routing back-bone which is a connected, sparse, planar spanner of the original Unit Disk Graph (UDG). Several useful properties of the back-bone, like sparseness, planarity and the stretch-factor have been proved by detailed theoretical analysis and the simplicity and construction-efficiency of the algorithm, in terms of the computation and communication cost incurred by the nodes have been compared with some recently proposed back-bone constructions. The constructed topology can be effectively used by ad-hoc routing protocols like the GPSR (Greedy State Perimeter Routing) that require the underlying topology to be planar.

1.2 Background Information

An ad-hoc network consists of a set of mobile wireless nodes, not connected through any fixed infrastructure like a base-station as in cellular networks. The communication between any pair of nodes occurs by the transmission of radio waves. Since the transmission range of each node is limited and there is no fixed infrastructure, the communication between two nodes not within range of one another takes place by forwarding the information through intermediate nodes which act as routers. Hence, any node in an ad-hoc network can behave as a router. Ad-hoc networks are mainly used in the battle-field as the networks formed by the military personnel, sensor networks, etc. Gradually their application in commercial devices like PDA's and laptops is becoming increasingly popular.

Nodes in an ad-hoc network are mobile and battery-operated. This makes power a very valuable resource in the ad-hoc network. The goal of each node is to communicate with every other node of the network using the minimum possible power. Conservation of nodal power has a direct influence on the lifetime of the ad-hoc network. Network Lifetime is the time duration until the first node of an ad-hoc network becomes non-functional. Also, if each node communicates at its maximum power level, the nodal transmissions would interfere with one another and obstruct the communication between nodes. Therefore, it is important to either have an optimal transmission power assigned to the nodes of an ad-hoc network or construct a network topology which enables communication using the minimum power at each node.

Two kinds of problems are most widely studied in ad-hoc networks: *i)* Topology Control, which involves construction of a connected, power-efficient topology and *ii)* Routing which is concerned with proposing power-efficient ways of exchanging information in an ad-hoc network. The topology control problem is very closely related to Computational Geometry and several algorithms proposing the construction of various topologies have been suggested. Similarly, several routing algorithms using power as the metric have been proposed for power-efficient routing in ad-hoc networks. The rest of this section discusses these problems, particularly topology control in detail and also highlights the research work undertaken in these areas.

1.2.1 Topology control

1.2.1.1 Modeling Ad-Hoc Networks. An ad-hoc network can be modeled as a set of points in Euclidean space, where each point represents a node. Each node is characterized by its transmitting and computing power. The computing power is required for the internal processing by the node. The wireless medium is susceptible to path-loss, interference between transmissions, signal loss due to physical obstructions and noise. As a result, the reception power is smaller than the power with which the radio signal is transmitted. If P_r and P_t respectively denote the reception and transmission power-levels and if d is the distance between any two nodes,

$$P_r = O\ (P_t / d^{\alpha})\ [16]$$

where $2 \leq \alpha \leq 4$ and the hidden constant in the big Oh notation depends on the antenna gains and carrier frequency. [16]

While the model above is more suitable for modeling at the physical layer, at the network layer the ad-hoc networks are often modeled as graphs. The UDG (Unit Disk Graph) is very widely used to model the ad-hoc networks.

According to the UDG model, a graph G = (V, E) represents an ad-hoc network. Each vertex corresponds to a wireless node, whereas there exists an edge $e \in E$ between nodes u and v only if u and v can directly communicate with each other. In a Unit Disk Graph (UDG), each node has a transmission range limited to a disc of radius 1 unit centered at the node. Hence, each edge e Є E in a Unit Disk Graph has length less than or equal to 1 unit.

If each node transmits with its maximum power i.e. retains the entire transmission range consisting of the unit disk centered at it, the resulting Unit Disk Graph will be extremely dense, with each node having a very large number of neighbors. This would give rise to interference among node transmissions and an unnecessary wastage of nodal power, resulting in a reduction of the Network Lifetime.

1.2.1.2 Useful Properties of the Topology. It is clear from the above discussion that if each node retains all its neighbors, it won't help form a power-efficient topology. As a result, most of the work undertaken in topology control is concentrated towards the formation of a sub-graph of the original UDG, at the same time ensuring that the formation of the sub-graph does not give rise to a disconnected network, thereby hindering communication between nodes. Besides, several algorithms have also been proposed to construct a topology having some other useful properties such as a bounded maximum and average node-degree, constant-bounded maximum and average transmission power, constant stretch factor, planarity, constant number of messages required to be sent by each node, etc. The importance of each of these properties is listed below:

1) ***Total Messages***: In ad-hoc networks, lesser message-passing between nodes for topology construction will result in lesser consumption of their battery-power and hence, a longer Network Lifetime. Moreover, the algorithm should be able to reconfigure the network topology quickly in case of node motion, which again requires lesser rounds of information exchange among the nodes. The goal of a topology control algorithm should be to construct a topology which requires at the most linear (O (n)) total messages sent by each node.

2) ***Average Node Degree***: The node degree is a measure of the number of neighbors with which each node will be interacting for sending/receiving messages. Hence, a

smaller average node-degree will imply lesser contention and lesser interference among nodal transmissions, thereby increasing the throughput of the network.

3) ***Maximum Node Degree***: A larger node degree at a node will cause a greater usage of power at the node and also lead to more interference. Topology control algorithms aim to produce a sub-graph with a constant-bounded maximum and average node degree.

4) ***Average and Maximum Node Power***: The maximum power used at each node is proportional to the longest edge incident on the node. Hence, a smaller node power will obviously save power and contribute towards increasing Network Lifetime. At the same time it is important not to lose network connectivity in an effort to reduce the average and max node powers because this can lead to partitioning of the network and hinder communication between every pair of nodes in the network.

5) ***Stretch Factor*:** Let G (V, E') be the sub-graph of the UDG (V, E). For any two arbitrary nodes, u and v, the maximum ratio of the length of the shortest path u...v in G to the length of the shortest path u...v in the original UDG is called the length-stretch factor of the graph G. When the length of the path is measured in terms of number of hops, it is called the Topological Stretch Factor. Topology control algorithms strive to construct a network topology with a constant stretch factor which will enable power-efficient routing to occur across the network. Such graphs are called t-spanners.

In an ad-hoc network, the topology construction needs to take place in a distributed manner by all the nodes forming the network. Each node needs to select a subset of neighbors from among all the nodes within its transmission range to form a connected topology, preferably satisfying the above mentioned properties. The topologies proposed by several previous algorithms vary in their degrees of simplicity, quality of the constructed topology and the ease with which they can be constructed in a distributed manner. In addition to these properties, certain routing algorithms which guarantee a high packet delivery success rate require the underlying topology to be planar. The GPSR (Greedy Perimeter Stateless Routing) algorithm is one such example. Besides the underlying topology being a planar spanner, it is also important that the total communication cost for the topology construction be as small as possible, since this will have a direct impact on the battery power consumption of the nodes and hence, on the network lifetime.

1.2.1.3 Commonly Used Geometric Structures. Most of the research work undertaken so far relies on certain basic geometric structures in one way or another for the topology-construction. These structures are either used directly or are combined with other structures with useful properties to obtain a hybrid structure satisfying as many of the above discussed properties as possible. One such set of graphs commonly used are the proximity graphs, wherein two nodes u and v are said to be in proximity of each other and are connected by an edge if they satisfy some geometric property. Some of the most widely used structures for topology control are described below:

1) ***Relative Neighborhood Graph (RNG):*** Defined as a sub-graph RNG = (V, E') of the UDG = (V, E), any two nodes u and v in a relative neighborhood graph are connected by an edge uv if and only if uv $\leq$ 1 and the lune formed by the unit discs centered at u and v contains no other node w ($w \neq u$ and $w \neq v$). In the equation form:

$$\forall\, w \neq u,v : d(u,v) \leq \max\,[d(u,w), d(v,w)] \quad [12]$$

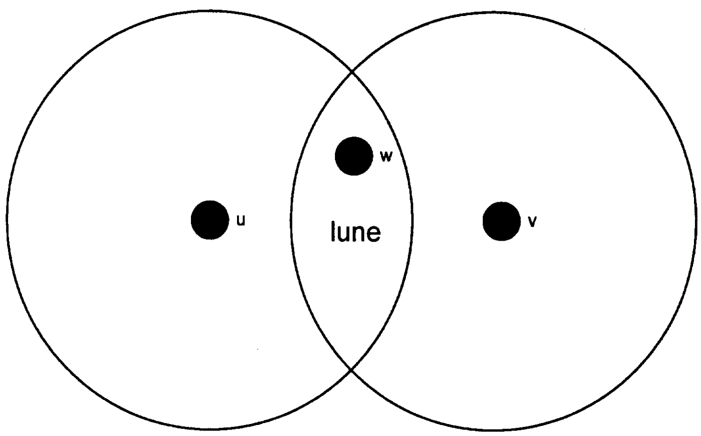

Figure 1.1 The RNG graph. For edge (u,v) to be included, the lune must contain no witness w.

2) ***Gabriel Graph (GG):*** Similar to the RNG, any two nodes u and v are connected by an edge uv in the GG if and only if uv $\leq$ 1 and the disc with uv as the diameter does not contain any other node w ($w \neq v$ and $w \neq u$). In the equation form:

$$\forall\, w \neq u,v : d^2(u,v) < d^2(u,w) + d^2(w,v) \qquad [12]$$

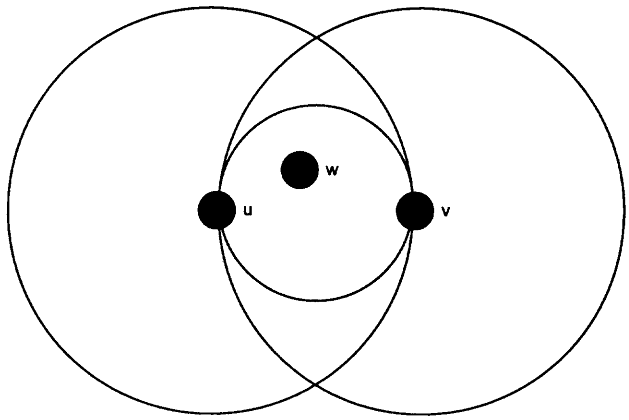

Figure 1.2 The GG graph. For edge (u,v) to be included, the circle with uv as the diameter must contain no witness w.

3) ***Delaunay Triangulation (DT):*** For a set of points in the plane, the Voronoi diagram partitions the plane into convex polygonal faces such that all points inside a face are closest to only one site. The Delaunay Triangulation is the dual graph of the Voronoi diagram, obtained by connecting the sites whose faces are adjacent in the Voronoi diagram.

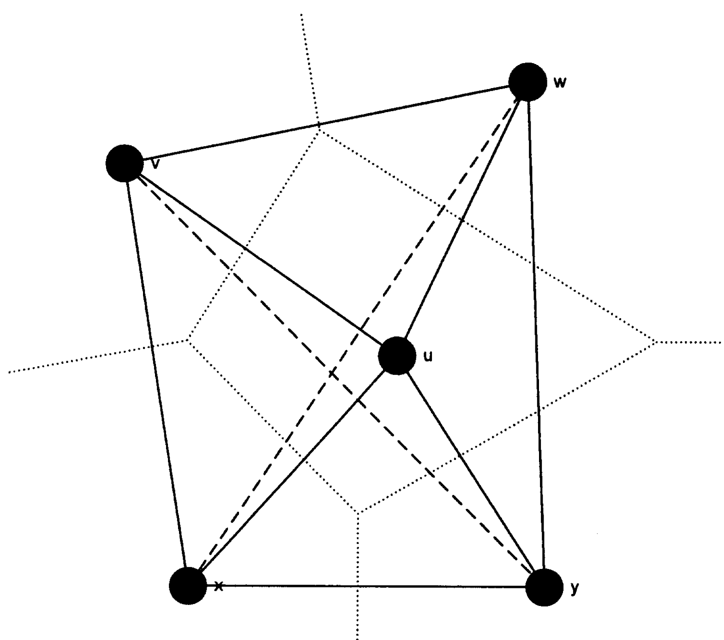

Figure 1.3 Voronoi Diagram and Delaunay Triangulation of a set of points.

4) ***Yao Graph (YG_k):*** For the construction of the Yao graph, each node divides the area around itself into k-equal sized cones and connects itself with the nearest neighbor in each cone. The Yao graph is also called θ-graph.

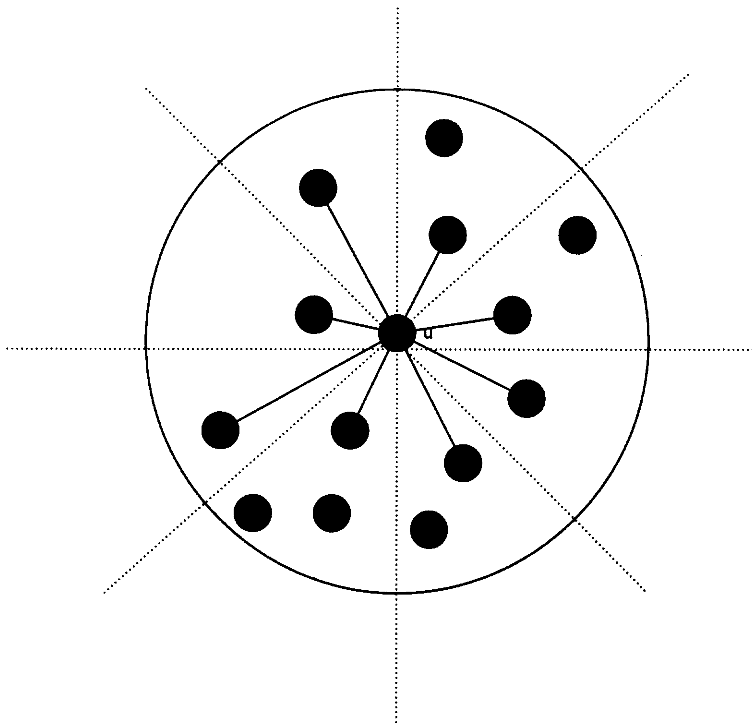

Figure 1.4 The Yao graph (YG_k) of a set of points.

All the four structures above can be categorized as Flat Structures, the other category being that of Hierarchical Structures. With respect to the above listed properties desirable in an ad-hoc network topology, the structures described above can be compared as shown in the table below:

Table 1.1 Comparison of widely used geometric structures

Topology	Length-Stretch Factor	Max-Degree	Planar
RNG	$\Theta(n)$	n-1	Yes
GG	$\Theta(\sqrt{n})$	n-1	Yes
YG_k	$1/(1-2\sin(\pi/k))$	k(out-degree), n-1(in-degree)	No
DT	$(1+\sqrt{5})\Pi/2$		Yes

Besides, it is also true that RNG, GG and DT are all sparse i.e the total number of edges is O (n) and:

$$RNG \subseteq GG \subseteq DT \qquad [2, 4, 5, 11]$$

As shown in the table above, the RNG and the GG, though planar, are not good spanners. Thus, two points directly connected by an edge in the UDG may end up being several hops apart in the RNG and GG. The Yao graph is a good spanner, but not planar. The Delaunay Triangulation is the only structure above which has both a constant stretch factor and is also planar. However, the main drawback of a DT is that it cannot be constructed efficiently in a localized manner. Some edges of the DT can be longer than the range of the node.

Most of the efforts in topology control are concentrated towards the formation of a planar sub-graph of the original UDG, which is also a t-spanner. Hence, much of the research work done in this area either uses the flat structures described above or a combination of these forming a hierarchical structure, often called a back-bone, to realize the goal of constructing a planar spanner as the underlying topology. While topology control mainly involves formation of a sub-graph of the UDG, routing is concerned with the implementation of power-efficient ways of forwarding information among nodes. The next section undertakes a comprehensive review of some of the research work done in the area of topology control.

1.2.2 Routing

Power-efficient routing in ad-hoc networks is also a very widely researched topic. Several routing algorithms have been proposed to enable communication between each pair of nodes in the network such that the total power used is minimized. The routing algorithms proposed for ad-hoc networks can be classified as flat, hierarchical or geographical [16]. While this research is primarily concerned with proposing a power-efficient topology, [12], [21], [22] deal with the routing problem by proposing newer power-efficient routing techniques.

CHAPTER 2

TOPOLOGY CONTROL: A SURVEY OF RELATED WORK

R. Rajaraman [16] conducted a detailed survey on two major aspects of ad-hoc networks: topology control and routing. It contained an excellent description of the entire topology control process right from modeling the network at the physical and network layers to describing the various techniques recently proposed towards realizing the objective of topology control. The paper clearly explains how the problem of topology control has been formulated as one in computational geometry by various researchers and also explains the desirable properties that an efficient topology control algorithm must possess. The ultimate goal of any topology control algorithm is to construct a topology which can enable power-efficient routing to be implemented in the ad-hoc network.

Karp and Kung [12] proposed the use of two planar sub-graphs: GG and the RNG as the underlying topology for routing using the GPSR (Greedy Perimeter Stateless Routing), which combines greedy forwarding on the full network graph with perimeter forwarding on the planarized network graph where greedy forwarding is not possible. The effectiveness of their technique in terms of packet delivery success rate, path length and routing protocol overhead is shown using network simulations.

Recently, Wattenhoffer et al. [1,3] proposed a cone-based topology control algorithm, very similar to the Yao graph. They claimed that the power efficiency of the routes obtained using their topology can be made arbitrarily close to optimal by a careful choice of parameters. As mentioned above, in spite of having a constant-bounded stretch factor, the Yao graph cannot guarantee planarity and constant-bounded node degree that could lead to interferences among node transmissions and improper use of the spatial bandwidth.

In [6] and [7], X.Y. Li et al. proposed various combinations of the GG, RNG and the YG_k and also proved some important points about the length and power-stretch factors of the resulting constructions. The First Yao then Gabriel graph, First Gabriel then Yao graph and the Yao plus Reverse Yao graph are some examples of the structures proposed. They also proposed the Yao and Sink topology, which has a constant-bounded node degree and is also a length-spanner, but not a hop-spanner. Also, it is a bit difficult to construct it in a distributed manner. Moreover, all the Yao-graph based topologies, though good spanners are non-planar.

Since no flat-structure topology exhibits all the desired properties of being a planar bounded-degree spanner, several hierarchical constructions have been proposed. The main idea in such algorithms revolves around the construction of a planar, spanner back-bone over which routing occurs. The back-bone consists of a subset of the entire vertex-set, with each node either being called a cluster-head or a gateway. Initially, a set of nodes called cluster-heads are selected on the basis of some property and thereafter gateways are appointed by these cluster-heads to form a connected back-bone.

As mentioned above, the idea behind hierarchical structures is that the back-bone becomes the routing graph. For communication to occur between any pair of nodes u and v, u first forwards the packet to its cluster-head and from there, the packet then makes its way among cluster-heads and gateways till it reaches the cluster-head of the destination node v. The cluster-head of the cluster to which v belongs finally delivers the packet to node v. Hence, the algorithms proposing the construction of such structures strive to ensure that the constructed back-bone as opposed to the entire constructed graph possesses as many of above described properties as possible.

In [4], Guibas et al. proposed the construction of a routing back-bone which is a connected, planar spanner. They call it the RDG (Restricted Delaunay Graph). The algorithm initially involves selection of a sub-set of nodes as cluster-heads and gateways which constitute the routing graph. Each cluster consists of a cluster-head and all the nodes that elected it. The clusters are allowed to overlap. Gateways are chosen according to the clustering algorithm in [9]. Secondly, a planar RDG is formed over the selected cluster-heads and gateways as the node set. They prove that the RDG is a Euclidean and a Topological spanner with the spanning ratio being approximately 5.08. They claim that the RDG outperforms the GG and the RNG in terms of routing performance using the GPSR algorithm as proposed in [12], since the RDG, though denser than the GG and RNG, is still sparse i.e. has O (n) edges and is also a planar spanner.

In [5], X.Y Li et al. claimed that the approximation constant achieved by Gao et al. was too big to have any practical meaning and that the construction of the RDG was not at all computation or communication efficient, that the communication cost can be as high as O (n^2). They proposed the construction of another hierarchical structure which was guaranteed to be a planar, bounded-degree spanner. Initially, a Connected Dominating Set (CDS) is calculated over the node set. This is followed by the selection of gateways. Thereafter to ensure the planar spanner property, the Localized Delaunay Triangulation (LDel) is constructed over the set of cluster-heads and gateways. They claim that the construction of the back-bone requires a total of O (n) messages i.e. the communication cost is linear. However, the LDel is not guaranteed to be planar and hence, extra code is required to be run at each node to ensure planarity. This makes the construction complex and time consuming.

Most of the hierarchical topology control algorithms aimed at constructing a planar, spanner back-bone, thus rely on the use of the Delaunay Triangulation due to its proven planar and spanner properties. Researchers have tried different ways of constructing the DT in a distributed manner in order to achieve this goal.

Another topic which has been widely researched in ad-hoc networks is that of Power Control. It deals with the optimum assignment of power levels to all the nodes in the ad-hoc network, just enough to keep the network connected and enable communication between each pair of nodes in the network. In formal words the problem can be described as: Assignment of transmission power levels to each node such that the wireless network is connected with the optimization criteria being minimizing the maximum or total transmission power assigned.

A transmission power assignment on the vertices in V is a function f from V to the set of real numbers. The communication graph G_f, associated with a transmission power assignment f, is a directed graph with V as its vertices and has a directed edge v_iv_j if and only if $\|v_i v_j\|^{\alpha} + c \leq f(v_i)$. A transmission power assignment f is said to be complete if the communication graph G_f is strongly connected. Here c is the fixed overhead incurred at each node in receiving and processing a signal. The maximum-cost of a transmission power assignment f is defined as $\max_{v_i \in V} f(v_i)$, while the total cost is defined as $\Sigma_{v_i \in V} f(v_i)$. The min-max assignment problem is then to find a complete transmission power assignment whose maximum cost is the least among all complete assignments. The min-total assignment problem is to find a complete transmission power assignment f whose total cost is the least among all complete assignments. [13], [15], [17] and [18] study the power assignment problem.

This work is closely related to [4] and [5] with respect to the idea of constructing a virtual back-bone graph consisting of cluster-heads and gateways, which is a connected, planar spanner. It aims to achieve the same in a more simplistic and less computation and communication-intensive manner. Since the nodes in an ad-hoc network are mobile, it is important for any topology control algorithm to be simple and time-efficient so that it can easily adapt to the changes in node positions and reconstruct the topology faster. While [4] and [5] rely on the Delaunay Triangulation, which is constructed in a localized manner, it is true that the construction of a DT is complex and time consuming [7]. This paper investigates the issue of retaining the desired planar spanner properties in the back-bone without requiring the complex and communication-intensive construction of the Delaunay Triangulation. The rest of the paper is organized as follows: The next chapter begins with a description of the model used by the algorithm followed by a detailed description of the algorithm itself. Finally, Chapter 4 will discuss and prove some important properties of the construction.

CHAPTER 3

CONSTRUCTION OF THE BACK-BONE

3.1 The Network Model

This paper models the ad-hoc network as a Unit Disk Graph UDG = (V, E) where V is the set of vertices with each vertex representing a wireless node and E is the set of edges where an edge uv $\in$ E if and only if $||uv|| \leq 1$ i.e. u is visible to v and vice versa. Thus, the transmission range of each wireless node is assumed to be limited to a circle of radius 1 centered at that node. Also, assume that each node is characterized by a unique ID.

Let N (u) denote the set of nodes visible to u, including u itself, which effectively includes all the nodes lying within the circle of radius 1 unit, centered at the node u. If $|V|=n$, then it is possible that there might be Θ (n^2) edges in the UDG. For any two nodes u and v, let Π_{UDG} (u, v) denote the shortest path from u to v in the UDG. Similarly, for any sub-graph G of the UDG, let Π_G (u,v) denote the shortest path from u to v in G. Then graph G is called a t-spanner of the UDG if $||\Pi_G (u,v)|| \leq t \, ||\Pi_{UDG} (u,v)||$ i.e. if the length of the shortest path in G is only a constant (t) times the length of the shortest path in the UDG.

The stretch factor measures the quality of the routing paths produced by the graph. One of the major goals of this paper is to construct a routing graph of the UDG with a constant stretch factor. Moreover, the algorithm ensures that the routing graph is also planar, so that it can be effectively used by routing algorithms such as the GPSR (Greedy Perimeter Stateless Routing), which guarantee high packet delivery success rates. The algorithm presented in the next section consists of two phases. The aim of the first phase is to partition the network into virtually-disjoint clusters. It begins by selecting a subset of the set of vertices V as the cluster-heads of the network. The algorithm guarantees that no two cluster-heads lie within range of each other ensuring that each node is covered by exactly one cluster-head to form disjoint clusters.

In the second phase, the virtually-disjoint clusters are connected by the selection of gateway nodes, at the same time maintaining the planarity of the topology. The goal here is to avoid the necessity of a separate step for computing the gateways and then having another step that takes care of planarizing the network. This would help reduce the message complexity of the algorithm and reconfigure the topology efficiently in case of node motion. The routing graph obtained at the end of the second has the following properties:

- It is planar
- It has a constant topological stretch factor and is therefore, a topological spanner
- It can be efficiently computed in a distributed setting

Even though this paper constructs the topology assuming the nodes to be static, the issue of maintaining the topology in case of node mobility can be taken up as a part of future work.

3.2 The Algorithm

3.2.1 Phase 1: Selection of cluster-heads and formation of clusters

Step 1: Each node compares its ID with that of its neighbors and if it is greater than the IDs of all its neighbors, marks itself as a cluster-head. If its ID is not greater than all the nodes in its neighborhood, it nominates the node with the highest ID in its neighborhood as the cluster-head, by sending an 'I nominate you' message to the node.

The other node, on receiving the 'I nominate you' message checks whether it has previously marked itself as a "cluster-head". If not, it marks itself as "cluster-head-elect". Thus, at the end of this step we have a set of nodes, called the "potential-cluster-head" set, with every node in the set marked either as a cluster-head or cluster-head-elect. For any two arbitrarily chosen nodes in this set, it is possible that they lie within range of each other. However, since the aim of this phase is to come up with clusters with a single cluster-head in each cluster, a way has to be found to eliminate such a situation.

Consider u and v to be a pair of nodes belonging to the potential-cluster-head set such that they are within range of each other. Then, there are three possibilities:

1) Both u and v are cluster-heads
2) u is a cluster-head and v is a cluster-head-elect or vice versa
3) Both u and v are cluster-head-elects

But the following lemma proves that the first condition can never occur:

Lemma 1: Two nodes selected as "cluster-heads" cannot be within range of one another.

Proof: Assume for the sake of contradiction that both u and v lie within range of each other and that both have selected themselves as "cluster-heads" in Step 1 above.

Consider node u first. u is a cluster-head and v is one of its neighbors. Hence, according to Step 1 of the algorithm above, u would have marked itself as a cluster-head only if ID (u) > ID (v)........ (1)

The same argument holds true for v too, which implies that ID (v) > ID (u)..... (2)

But (1) and (2) cannot both be true. Hence, the lemma follows.

Thus, the remaining steps of the algorithm should come up with a way to deal with situations 2) and 3) described in Step 1 above. This is explained in Step 2, which comprises of Step 2(a) and Step 2(b):

Step 2: Cluster-head-elect Elimination

Step 2(a): Each node selected as cluster-head-elect in Step 1 determines if it is within range of a node selected as a cluster-head in Step 1. If so, it deselects itself and also informs its neighbors about it no longer being cluster-head-elect. Thus, at the end of this step the potential-cluster-head set gets updated with some cluster-head-elect nodes deselecting themselves and hence, being eliminated from the set.

Note: At the end of Step 2(a), each node has updated information about whether its neighbors are cluster-heads or cluster-head-elects.

Step 2(b): Each node marked as a cluster-head-elect in the updated potential-cluster-head set (it was updated at the end of Step 2(a)), now determines if it is within range of another node, also marked as a cluster-head-elect. If it does, it compares its ID with the ID of that node and if its ID is less than the ID of the other node, it deselects itself and also informs all its neighbors about it no longer being cluster-head-elect. Thus, at the end of this step the potential-cluster-head set again gets updated with some more cluster-head-elect nodes deselecting themselves.

Lemma 2: At the end of Step 2 of the Phase 1, no two nodes marked as cluster-heads or cluster-head-elects lie within range of one another.

Proof: We already proved in Lemma 1 that two nodes marked as cluster-heads cannot be within range of one another.

However, for any two arbitrary nodes u and v belonging to the potential cluster-head set and within range of each other, two possibilities still existed at the end of Step 1, which were not dealt with there:

1) u is a cluster-head and v is a cluster-head-elect or vice versa
2) Both u and v are cluster-head-elects

The Step 2(a) guarantees that each cluster-head-elect would deselect itself, if it is within range of a cluster-head. Hence, condition 1) will not occur after Step 2.

Similarly, Step 2(b) guarantees that each cluster-head-elect would deselect itself if it is within range of another cluster-head-elect with a larger ID and for any two cluster-head-elects u and v within range of one another, either ID(u) > ID(v) or ID(v) > ID(u) since each node has a unique ID.

Hence, in Step 2(b) either u or v must deselect itself. Thus, condition 2) won't occur at the end of Step 2 too, and hence, the lemma follows.

As described above, at the end of Step 1, every node either nominates a node which would be its cluster-head-elect or selects itself as a cluster-head. Hence, we can guarantee that at the end of Step 1 each node was covered by at least one cluster-head or a cluster-head elect. However, since in Steps 2(a) and 2(b) certain cluster-head-elect nodes deselected themselves and hence, that guarantee no longer holds. To take care of this situation, the Step 3 is performed as shown below:

Step 3: Each remaining node in the potential cluster-head set at the end of steps 2(a) and 2(b) sends a message "I am a cluster-head" to all its neighbors, lying within the unit circle centered at itself. As a result, each node that receives this message knows that it is covered by at least one cluster-head. The nodes that do not receive this message realize that they are not covered by any cluster-head or remaining cluster-head-elects and hence, mark themselves as forced-cluster-heads. Thus, at the end of Step 3 the guarantee that each node is either covered by at least one cluster-head and if not, is marked as a forced cluster-head holds. In order to reduce the number of forced cluster-heads, Step 4 is performed:

Step 4: Each node that selected itself as a forced cluster-head, determines if its ID is greater than the IDs of all other forced cluster-heads in its neighborhood and in that case sends a message "I am the forced cluster-head" to all the nodes in its neighborhood. All the nodes on receiving this message, deselect themselves in case they had marked themselves as forced-cluster-heads in Step 3 and form a cluster with the sending node becoming the cluster-head of the cluster. Note that the other nodes that might receive the "I am the forced cluster-head message" and which have not been marked as forced-cluster-heads in Step 3 are already covered by some cluster-head or cluster-head-elect and hence, wont form a cluster with the forced-cluster-heads. A cluster formed by a forced- cluster-head thus, can only contain other forced-cluster-heads with lower IDs. The remaining forced-cluster-heads form a cluster by themselves. Note that the forced-cluster-heads do not participate in cluster-formation in Step 5.

The nodes marked as cluster-head-elects also mark themselves as cluster-heads now and participate in cluster-formation along with the other cluster-heads which were selected in Step 1, as shown by Step 5 below:

Step 5: Formation of Virtually-Disjoint Clusters

The aim of this step is to partition the network into virtually-disjoint clusters. At the end of Step 4, the network is composed of the clusters formed by forced-cluster-heads, the nodes marked as cluster-heads (which now include cluster-heads and cluster-head-elects as mentioned in Step 4) and the other non-cluster-head nodes, each of which is covered by at least one cluster-head. Each node decides in this step which cluster-head to be with (including the forced-cluster-heads) while forming a cluster. Having selected one cluster-head, the node ignores any communication from other cluster-heads it might be within range of. This results in a virtual-partitioning of the network. Again, note that forced-cluster-heads do not participate in this step and have been already grouped into clusters.

3.2.2 The Collinear Problem

While each node which is within range of two or more cluster-heads can easily select one of them on the basis of some property, while ignoring communication from the others to form virtually-disjoint clusters across the network, there is one situation which can hinder the virtually-disjoint cluster formation:

Consider any two arbitrary nodes u and v which are within range of a cluster-head CH1. Also assume that the nodes CH1, u, v are collinear, with u lying between v and CH1. i.e. CH1-u-v as shown in the figure below:

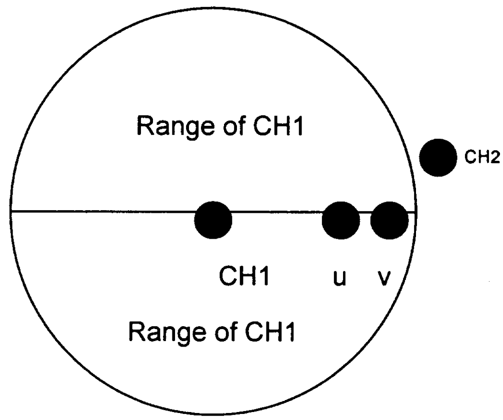

Figure 3.1 The collinear problem

Assume that the node u is also within range of another cluster-head CH2 not collinear with CH1, u and v, as shown in the Figure 3.1 above. If the node u, due to some reason chooses to go with the cluster-head CH2, while the node v chooses to remain with CH1, it would be impossible to form disjoint clusters.

The aim of this step is to produce disjoint clusters which may not be symmetric in shape, however, it must ensure that for a case like u and v shown in the figure above, there should never occur a situation wherein v chooses to go with CH1 and u which is on the line-segment vCH1 chooses to go with another cluster-head CH2.

Hence, in Step 5, each node that is covered by at least one cluster-head, decides to go with the cluster-head which is closer to itself. Each node is assumed to have the capability to calculate its distance from another node on the basis of the signal strength. Also, in case of a tie, i.e when a node is equidistant from two cluster-heads it chooses to go with the one having a larger ID. The following theorem and lemma prove that these two decisions will not give rise to the collinear problem and hence, produce disjoint clusters. Note that, it is not required to take into account the situation where CH2 is collinear with u, v and CH1 because in that case no matter where CH2 lies, the collinear problem will not occur.

Theorem 1: The collinear problem cannot occur when a node decides to go with the cluster-head closest to itself.

Proof: For the sake of contradiction, assume that the collinear problem can arise even if each non-cluster-head node decides to go with the cluster-head which is closest to itself. As shown in Figure 3.1 above, let CH1, u, v be collinear, with both u and v lying within range of the cluster-head CH1 and u lying between CH1 and v i.e. CH1-u-v.

Also assume that there exists another cluster-head CH2 such that:

1) uCH2<uCH1<1, therefore, u chooses to go with CH2
2) vCH1<vCH2, therefore, v chooses to go with CH1

Since v is within CH1's range, vCH1$\leq$ 1. Also since CH1-u-v, uCH1<vCH1

Hence, if uv = Δ,

uCH1 = vCH1 - Δ.. (3)

Now, according to our assumption 1), uCH2<uCH1.

Therefore, let uCH2 = uCH1 - Δ_2 = vCH1 - Δ - Δ_2.......................... .(4) [By (3)]

Then, according to the triangular inequality, in the triangle uvCH2,

vCH2 < uv + uCH2

Hence, vCH2 < Δ + (vCH1- Δ - Δ_2) = vCH1 - Δ_2 [By (4)]

which implies that vCH2<vCH1.

This contradicts the assumption 2) that vCH2>vCH1, and hence, v chooses vCH1. Thus, it can be said that the assumption was incorrect and the collinear problem cannot occur if each node chooses to go with the cluster-head closest to itself.

Lemma 3: In case of a tie, which is when a node is equidistant from two cluster-heads and within range of both, if the node chooses to go with a cluster-head with a higher ID, the collinear problem cannot occur.

Proof: Consider the situation wherein as shown in the figure 3.1 above the node u is equidistant from the cluster-heads CH1 and CH2 and as in the figure:

CH1, u and v are collinear with u being between CH1 and v.

Again, as in Theorem 1 above, assume that:

1) uCH1 = uCH2, i.e. node u is equidistant from CH1 and CH2, but since ID(CH2) > ID(CH1), u chooses CH2
2) vCH1 < vCH2 and hence, v chooses CH1

Again, since u, v, CH1 are collinear and also since u lies between v and CH1, assuming that uv = Δ, uCH1 = vCH1 - Δ.......................... (3)

Since uCH1=uCH2, uCH2 = vCH1 - Δ................ (4)

Again, by the triangular inequality, in the triangle uvCH2,

vCH2 < uv + uCH2 = Δ + (vCH1 - Δ) = vCH1 [By (3)]

Thus, vCH2<vCH1 which again contradicts the assumption (2) that vCH1 < vCH2 and therefore, v chooses CH1.

Thus, according to Theorem 1 and Lemma 3, whenever there is a situation wherein two nodes u and v within range of a cluster-head CH1 are collinear such that u is between CH1 and v, and if u is within range of another cluster-head CH2, the collinear problem will not occur if u chooses to go with the cluster-head that is closest to itself or if it goes with the cluster-head with a higher ID, in case it is equidistant from both CH1 and CH2.

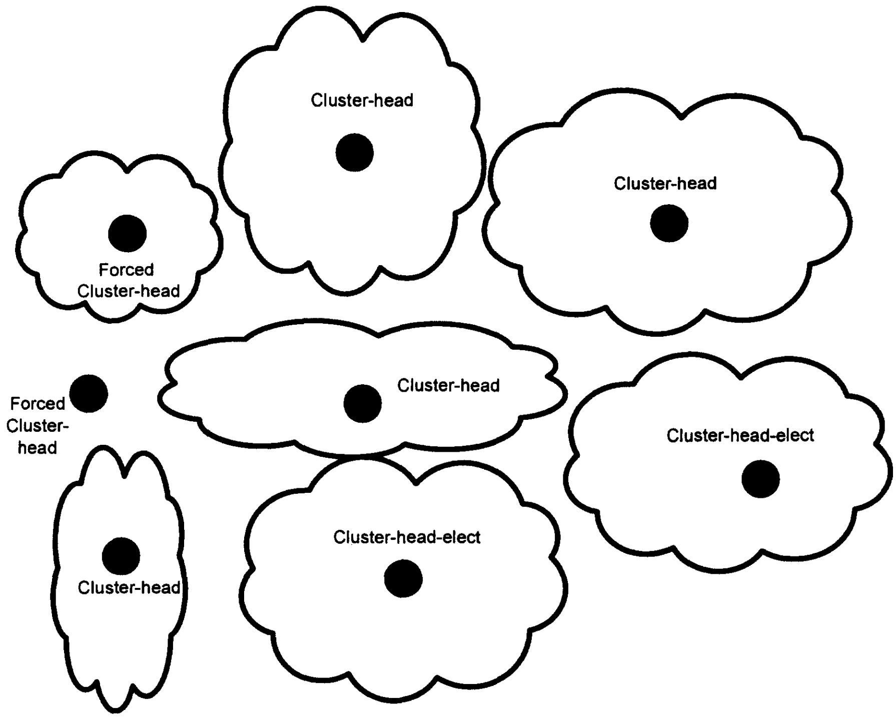

Figure 3.2 The virtually-disjoint clusters

Therefore, as shown in the Figure 3.2 above, it can be said that at the end of Phase 1, the network is partitioned into clusters that are virtually-disjoint with each node being covered by exactly one cluster-head. The next phase deals with joining the clusters by means of gateways, giving rise to a connected, planar back-bone which is also a t-spanner of the original UDG.

3.2.3 Phase 2: Joining the Clusters to form the Back-bone

As mentioned before, the goal of this phase is to construct a back-bone by joining the virtually-disjoint clusters formed during Phase 1 through the selection of gateways. It is important for this back-bone formed by cluster-heads and gateways to be a planar graph, since routing will be implemented over the back-bone and several routing algorithms such as the GPSR which guarantee high packet-delivery rates require the underlying topology to be planar. Moreover, it is also desirable for the back-bone to be a t-spanner of the original Unit Disk Graph, which will guarantee power-efficient routing across the network, thereby increasing the Network Lifetime.

In this phase of the algorithm, the objective of forming a planar spanner back-bone for routing is realized through the use of the Gabriel Graph property. Two nodes u and v are said to form a Gabriel edge if there is no other node w in the circle with the Euclidean distance uv as the diameter. Hence, the goal of this phase is to select gateway nodes for each of the clusters produced at the end of Phase 1 and connect them by means of Gabriel edges which can guarantee the planarity of the back-bone. This phase comprises of two steps which realize the objective of forming a planar spanner back-bone:

Step 1: Gateway Selection

In this step each node, including the cluster-heads checks to see if any of its neighbors i.e. nodes lying within the disc of radius 1 centered at it belong to clusters other than the one it belongs to. If it has neighbors belonging to a different cluster, it marks itself as a gateway and connects with each of those neighbors through an edge.

At the end of this step, it is guaranteed that we have a connected graph if the original UDG was connected. However, the resulting graph at the end of this step is not planar, and the edges between gateways might cross each other. Therefore, an additional step is necessary to guarantee the planar property.

Step 2: Planarizing the back-bone

Though the Step 1 guarantees the formation of a connected back-bone, it cannot guarantee planarity and as discussed before, it is essential for the underlying graph to be planar to be useful for routing by algorithms like the GPSR (Greedy Perimeter Stateless Routing). This step takes care of planarizing the back-bone by eliminating the edges joining the gateway nodes if they do not satisfy the Gabriel edge property.

In this step, each node that had marked itself as a gateway checks each of the edges incident on it to see if it satisfies the Gabriel property. Assume that the nodes u, v and w were selected as gateways in the Step 1 above and are connected to each other i.e. each node connects to the other two. However, in this step, the nodes u and v discover that the edge uv is not a Gabriel edge since w lies inside the disc with diameter uv. Hence, the edge uv is removed. Note that the node w has to be a gateway node in order to account for the removal of the edge uv.

While Step 1 guaranteed connectivity of the back-bone, Step 2 constructs a planar back-bone. Moreover, the connectivity of the topology is also retained since the formation of a Gabriel graph by the elimination of edges does not disconnect the underlying graph if it was originally connected [12]. Thus, at the end of Step 2 a planar connected back-bone is constructed.

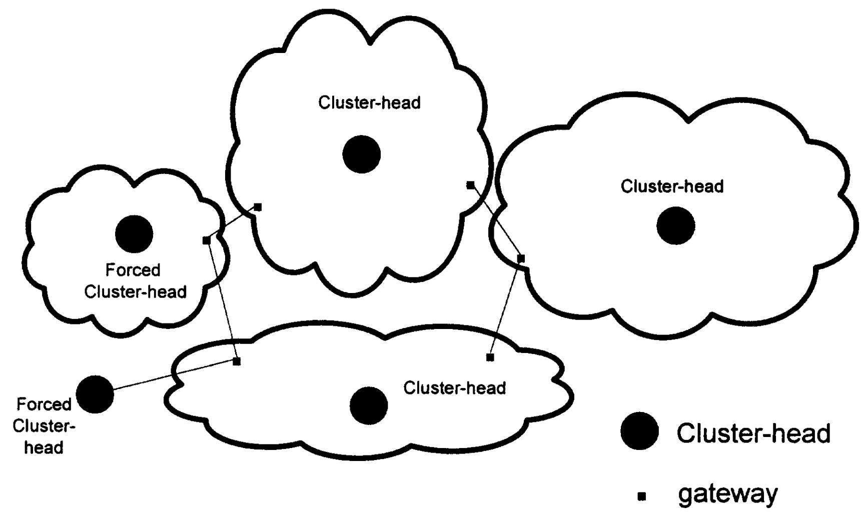

Figure 3.3 The final back-bone

The Figure 3.3 above gives an idea of how the back-bone produced at the end of the Phase 2 of the algorithm looks. As shown in the figure, the network gets partitioned into virtually-disjoint clusters with each cluster having a cluster-head which could have been marked as a cluster-head, cluster-head-elect or a forced-cluster-head during the Phase 1.

During the Phase 2, the clusters are joined by means of gateways to produce a planar spanner back-bone over which power-efficient routing can be implemented.

CHAPTER 4

PROPERTIES OF THE BACK-BONE

Lemma 4: At the end of Phase 2, the number of cluster-heads and gateways in any unit disk in the plane is O (1) in expectation.

Proof: This lemma is based on a corollary in [4]. Since the construction of the algorithm described above is similar to the back-bone constructed in [4], the lemma holds true here too. A detailed proof can be found in [19]. Also since the set of cluster-heads and gateways obtained in the algorithm above is a subset of the set derived in [4], the lemma holds.

In the algorithm proposed in [4], each node gets an opportunity to nominate a node in its neighborhood as its cluster-head and the nodes thus marked as cluster-heads, form the final set of cluster-heads. On the other hand, as shown in Phase 1 of the algorithm above, the nodes undergo an elimination round in Step 2 which guarantees that no two nodes marked as cluster-heads or cluster-head-elects are within range of each other. Thus, the set of cluster-heads formed by this algorithm is a sub-set of the one produced in [4].

Although the nodes marked as forced-cluster-heads may lie within range of other chosen cluster-heads, they can be avoided from consideration here since they are already grouped into clusters or constitute a cluster by themselves and do not participate in the cluster-formation process in the Step 5 of Phase 1.

Similarly, the set of gateways produced by the Phase 2 in the algorithm above is a subset of the set produced in [4]. This is because in [4], any two nodes within range of one another become gateways and connect with each other, whereas in the algorithm above, as shown in the Step 2 of the second phase, gateway nodes are retained only if they form GG edges with the gateways of other clusters. Hence, the lemma holds.

4.1 Planarity of the Back-bone

The routing back-bone constructed by the algorithm above is planar since the edges joining any two gateways in the graph are GG edges and the Gabriel graph is a planar graph. [4, 5, 12]

4.2 The t-spanner Property

Lemma 5: The back-bone constructed by the algorithm above is a topological spanner graph with a constant stretch factor. That is, for any two nodes u and v (cluster-heads/gateways) in the back-bone graph G_{bb}, $\delta_{bb}(u, v) \leq C_1 \cdot \delta_{UDG}(u,v)$ for some constant $C1 > 0$ where δ (u,v) denotes the shortest distance between u and v.

Proof: Consider any two nodes u and v in the back-bone graph G_{bb}. Since the back-bone graph is formed over the node set consisting of only cluster-heads and gateways, each of the nodes u and v is either a cluster-head or a gateway. According to the Lemma 4 above, the back-bone graph has constant density i.e. in any unit circle of the back-bone graph, there are O(1) cluster-heads and gateways in expectation.

Hence, for any two nodes u and v of the back-bone that were connected in the UDG and were selected as gateways/cluster-heads but could not connect in the back-bone due to a node w that prevented the formation of a GG edge uv, Lemma 4 implies that there are always a constant number of such nodes w.

As a result, the topological distance between the nodes u and v in the back-bone is always a constant times the shortest distance between them in the UDG, which means that $\delta_{bb}(u,v) \leq C_1 \cdot \delta_{UDG}(u,v)$.

4.3 The Routing Graph

While G_{bb} denotes the back-bone constructed by the algorithm described above and consists of only the cluster-heads and gateways, the entire topology comprises of the cluster-heads, gateways and the remaining nodes which were not selected as either cluster-heads or gateways. This graph is called the routing graph R.

The transmission of a packet from a source u to the destination v occurs as follows in the graph R: The node u first forwards the packet to its cluster-head, from here on the packet makes its way over the back-bone i.e. over cluster-heads and gateways, till it reaches the cluster-head of the cluster to which node v belongs. The cluster-head then forwards the packet to v.

Lemma 6: Graph R is also a topological spanner graph with a constant stretch factor.

Proof:

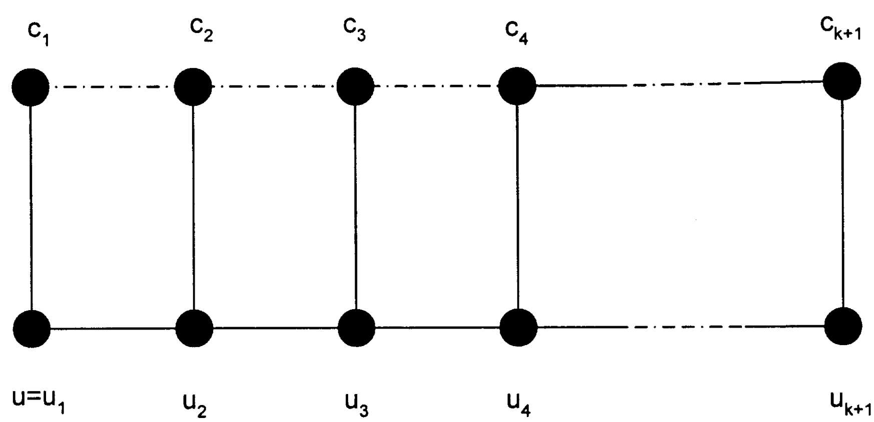

Figure 4.1 The routing graph

Suppose that the shortest topological path in the UDG between nodes u and v is P: $u_1 = u$, u_2, $u_3 \ldots\ldots u_{k+1} = v$. Suppose that in the back-bone graph G_{bb}, the cluster-head of u_i is c_i. As proved in Lemma 5, $\delta_{bb}(c_i,c_{i+1}) \leq C_1 \cdot \delta_{UDG}(c_i,c_{i+1})$ for some constant $C_1 > 0$. Then for the path P' between u and v in R, which is the union of $\delta_{bb}(c_1,c_{k+1})$ and the edges u_1c_1 and $u_{k+1}c_{k+1}$, $\delta_R(u,v) \leq 2 + \sum_{i=1}^{k} C_1 \cdot \delta_{UDG}(c_i,c_{i+1})$.

Also, for nodes c_i and c_{i+1}, $\delta_{UDG}(c_i,c_{i+1}) \leq 3$ since c_i and c_{i+1} if not neighbors, are connected through the nodes u_i and u_{i+1}. Thus, $\delta_R(u,v) \leq 2 + 3 \cdot k \cdot C_1$ which implies that R is a topological spanner graph.

4.4 Conclusion

Topology control is a very important aspect in ad-hoc networks and since nodes are battery operated, power is a very important resource. Besides, topology control also has a direct impact on the quality and power-efficiency of routes over which nodes transmit information to other nodes in the network. Since the nodes in an ad-hoc network are mobile, it is important for the underlying topology to be simple and easily reconfigurable in case of node motion.

This paper proposed a novel technique for partitioning the ad-hoc network into virtually-disjoint clusters such that each node is covered by exactly one cluster and there are is no overlapping of clusters. This is probably the first technique that proposes the formation of disjoint clusters. The paper also proposed a powerful heuristic for joining the clusters through gateways in such a way that the routing back-bone obtained is a planar spanner of the original Unit Disk Graph. Several arguments were also provided to support the claim.

While most of the algorithms recently proposed [4,5] for the construction of a back-bone rely on the expensive and complex construction of the Delaunay Triangulation for achieving the planar and spanner properties in the topology, this paper proposed a unique distributed technique to construct a planar back-bone which is also a t-spanner of the original UDG, without requiring the complex construction of the Delaunay Triangulation, while retaining the important planar and spanner properties through a simple and efficient construction. Moreover, it can be easily shown that the total number of messages required by each node for the topology construction will be linear i.e. O(n). This simplicity and ease of construction are important characteristics for reconfiguring the topology in case of node motion. The constructed topology can be efficiently used by routing algorithms like the GPSR which guarantee a high packet delivery success rate.

REFERENCES

1. R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks", *Proc. IEEE Infocom,* 2001.

2. X. Y. Li, G. Calinescu, P. Wan, "Distributed Construction of a Planar Spanner and Routing for Ad-hoc Wireless Networks", *Proc. 21st Ann. Joint Conf. IEEE Computer and Comm. Soc. (INFOCOM),* vol. 3, 2002.

3. L. Li, J. Y. Halpern, P. Bahl, Y. Wang and R. Wattenhofer, "Analysis of a Cone-based Distributed Topology Control Algorithm for Wireless Multi-hop Networks", *Proc. ACM Principles of Distributed Computing Conference (PODC'01),* Aug. 2001.

4. J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Geometric Spanner for Routing in Mobile Networks," *Proc. Second ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 01),* 2001.

5. K. Alzoubi, X.Y. Li, Y. Wang, P. Wan and O. Frieder, "Geometric Spanners for Wireless Ad Hoc Networks", *IEEE Transactions on Parallel and Distributed Systems*, May 2003.

6. X.Y. Li, P. J. Wan, Y. Wang, and O. Frieder, "Sparse Power Efficient Topology for Wireless Networks", *Proc. IEEE Hawaii Int'l Conf. System Sciences (HICSS),* 2002.

7. X.Y. Li, P. J. Wan, and Y. Wang, "Power Efficient and Sparse Spanner for Wireless Ad Hoc Networks," *Proc. IEEE Int'l Conf. Computer Comm. And Networks (ICCCN),* 2001.

8. X.Y. Li, Y. Wang, P. Wan, W. Song and O. Frieder, "Localized Low-Weight Graph and Its Applications in Wireless Ad Hoc Networks".

9. J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Discrete Mobile Centers," *Proc. 17^{th} ACM Symp. Computational Geometry (SCG 01),* 2001.

10. Toussaint, G, "The Relative Neighborhood Graph of a finite planar set," *Pattern Recognition* 12, 4 (1980), 261-268.

11. P. Bose, L. Devroye, W. Evans, and D. Krikpatrick, " On the spanning ration of Gabriel graphs and β-skeletons," *submitted to SIAM Journal on Discrete Mathematics,*2001.

12. B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* 2000.

13. S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," *European Wireless Conference,* 2002.

14. V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Trans. On Communications,* vol. 34, no. 1, pp. 38-44, January 1986.

15. R. Ramanathan and R. Rosales-Hain, "Topology control of multi-hop wireless networks using transmit power adjustment," *Proc. IEEE Infocom 2000,* March 2000.

16. R. Rajaraman, "Topology control and routing in ad hoc networks: a survey," *SIGACT News,* 2002.

17. L.M. Kirousis, E. Kranakis, D. Krizanc, A. Pelc., "Power consumption in packet radio networks," *(STACS '97).*

18. A.E.F. Clementi, P. Penna, and R. Silvestri, "The power range assignment problem in radio networks on the plane," *XVII Symposium on Theoretical Aspects of Computer Science (STACS'00), LNCS (1770)*:651-660, 2000.

19. J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Discrete Mobile Centers," *Proc. 17^{th} ACM Symp. Computational Geometry (SoCG 01),* pp. 188-196, 2001.

20. E.L. Lloyd, R. Liu, M.V. Marathe, R. Ramanathan, and S.S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks," *In Proc. Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc'2002)*

21. D. Johnson and D. Maltz, "Dynamic Source Routing in ad-hoc wireless networks" In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153-181. Kluwer Academic Publishers, 1996.

22. C. Perkins, Ad-hoc on-demand distance vector (AODV) routing, Internet draft, draft-ietf-manet-aodv-04.txt, October, 1999.