國立交通大學

網路工程研究所

碩 士 論 文

以最小區域圓覆蓋集合為基礎用於異質無線隨意網路之廣播

Minimum Local Disk Cover Sets for Broadcasting in Heterogeneous Wireless Ad Hoc Networks

研 究 生：劉芳君

指導教授：易志偉　教授

中華民國 九十六 年六月

以最小區域圓覆蓋集合為基礎用於異質無線隨意網路之廣播
Minimum Local Disk Cover Sets for Broadcasting in Heterogeneous Wireless Ad Hoc Networks

研 究 生：劉芳君　　Student：Fang-Chun Liu

指導教授：易志偉　　Advisor：Chih-Wei Yi

國 立 交 通 大 學
網 路 工 程 研 究 所
碩 士 論 文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

以最小區域圓覆蓋集合爲基礎用於異質無線隨意網路之廣播

學生：劉芳君　　　指導教授：易志偉

國立交通大學網路工程研究所

摘要

在無線隨意網路中，廣播是其中一個最基礎的網路操作方法。他被廣泛使用在發現網路拓蹼、傳輸路徑與監視網路的完整性。而其中一個簡單的機制，稱爲氾濫式演算法，它的方法是當節點收到第一份訊息時，這個節點會傳送這個訊息給它所有的鄰居。儘管它很簡單，氾濫式演算法傾向去產生大量多餘的重傳。因此氾濫式演算法可能會引起廣播風暴問題，而且在電源和頻寬上也沒有效率。爲了減輕廣播風暴問題，當某個節點需要廣播時，這個節點只選擇部分的鄰居，稱爲前傳集合來幫它重送訊息，而不是所有鄰居都來傳送訊息。在前傳集合內的所有節點會覆蓋住所有 2-hop 鄰居，所以可以確定所有網路上的節點都會收到這廣播訊息。

在同質性網路中，根據 1-hop 鄰居的覆蓋範圍，可以計算出前傳集合。這個節點的前傳集合的覆蓋範圍跟所有鄰居的覆蓋範圍是一樣的。而被提出的演算法是區域性且分散式的演算法，而且具有最佳的時間複雜度。

在這篇論文中，我們提出一個機制以最小區域圓覆蓋集作爲前傳集合在異質性網路中(各個節點有不同的傳輸半徑)。對每個節點來說，如果它的鄰居的子集合有最少的個數而且它們的覆蓋範圍和所有鄰居的覆蓋範圍是一樣的，則這個子集合稱作最小區域圓覆蓋集合。首先我們證明找它的最小區域性的圓盤覆蓋集合與找它的輪廓線集合是相等的。我們提出一個各個擊破的演算法來局部地計算此輪廓線集合，而證明它有最佳的時間複雜度 *O(nlogn)*。

關鍵字：最小區域圓覆蓋集合、廣播、無線隨意網路、前傳集合

Minimum Local Disk Cover Sets for Broadcasting in Heterogeneous Wireless Ad Hoc Networks

Student: Fang-Chun Liu　　Advisor: Prof. Chih-Wei Yi

Department of Computer Science
National Chiao-Tung University

ABSTRACT

In wireless ad hoc networks, broadcasting is one of the fundamental networking operations. It is widely and frequently used to explore the network topology, find routing paths, and monitor network integrity. A simple broadcasting mechanism, known as flooding, is to let every node relay messages to all its 1-hop neighbors when it receives the messages the first time. Despite its simplicity, flooding tends to generate too many redundant retransmissions. It may cause the broadcast storm problem and is neither power nor bandwidth efficient. To relieve the problem, when a node does broadcasting, it selects a subset of neighbors called forwarding set to relay messages instead of all neighbors. Nodes in the forwarding set of a node would cover all its 2-hop neighbors, so it ensures that messages can reach all nodes in the network.

In homogeneous networks, it has been proposed computing the forwarding set based on the coverage area of 1-hop neighbors. The nodes in the forwarding set of a node can cover the same area as its all 1-hop neighbors. The proposed algorithm is localized, distributed, and with the optimal time complexity *O(nlogn).* In this paper, we propose to use the minimum local disk cover set as forwarding set in heterogeneous networks, where nodes may have different transmission radii. A minimum local disk cover set of a node is a subset of 1-hop neighbors and the number of set is smallest. The nodes in the minimum local disk cover set cover the same area of all 1-hop neighbors. We show that the minimum local disk cover set of a node is equivalent to its skyline set. We propose a divide-and-conquer algorithm with the optimal time complexity *O(nlogn)* to compute the skyline set locally.

Keywords: minimum local disk cover sets, broadcasting, wireless ad hoc networks, forwarding set

誌謝

進入研究所這兩年，從指導教授與系上老師們學到許多寶貴的經驗與知識，無論是做研究的方法與態度，還是做人處事的道理與如何與其他人相處，我都受益良多，對提攜我的老師與長輩們心懷感激。最後再次感謝所有指導、協助與鼓勵我的教授、同學與家人。

Contents

List of Figures

Chapter 1

Introduction

1.1 Wireless Ad Hoc Networks

Wireless ad hoc networks consist of a collection of wireless devices sharing a common channel without the need of centralized controls or fixed infrastructures. Each node is equipped with an omnidirectional antenna to access the common channel and usually applies the CSMA/CA(carrier sense multiple access with collision avoidance) [20] protocol. A communication session is built either through a single-hop radio transmission if the communication parties are close enough, or through relaying by intermediate nodes. Wireless ad hoc networks can be flexibly and quickly deployed for many applications such as personal area networks, smart home environment, environmental monitoring, battlefield surveillance, and emergency disaster relief.

Wireless ad hoc networks still have some challenges, including [18]

(1) Energy conservation: Because the devices in wireless ad hoc networks are typically battery equipped, one of the design goals is to use the limited energy as efficiently as possible.

(2)Resource constrained computation: The energy and network bandwidth are limited in wireless ad hoc network. Protocols must provide the good performance although the few available resource.

1.2 Broadcast Storm Problem

Broadcasting is one of the fundamental operations in wireless ad hoc networks. A source node sends messages to their 1-hop neighbors, and 1-hop neighbors relay messages to their neighbors, and so on so forth, until all nodes receive one copy of messages. It is widely and frequently used to explore the network topology, find routing paths, and monitor network integrity. Flooding, the most known broadcasting mechanism, is to let every node relay messages to all its 1-hop neighbors when it receives the messages the first time.

Despite its simplicity, flooding has a serious drawback, known as the broadcast storm problem [1]. First, because the radio propagation is omnidirectional and a physical location may be covered by the transmission ranges of several hosts, many rebroadcasts are considered to be redundant. Second, heavy contention could exist because rebroadcasting hosts are probably close to each other. Third, collisions are more likely to occur because the RTS/CTS dialogue is inapplicable and the timing of rebroadcasts is highly correlated. Since every node retransmits messages without considering whether their neighbors have received messages, unnecessary transmission occurs frequently. That results in shortening lifetime of battery-driven devices, and boosting network traffic and bandwidth contention. So, flooding is neither power-efficient nor bandwidth-efficient. Instead, many alternative broadcasting algorithms [1] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] have been discussed in literature.

1.3 Motivations and Research Methods

In most works, to relieve the broadcast storm problem, when a node receives a broadcast, it selects a subset of neighbors to relay messages instead of all neighbors. Such subset is referred as forwarding set [7] or multipoint relays (MPRs) [15]. Basically, each node needs to maintain a local network topology of its 1-hop or even 2-hop neighborhood. To ensure that a broadcast can reach all nodes in the network, nodes in the forwarding set of a node should cover all its 2-hop neighbors. At the same time, to better relieve the broadcast storm problem, the forwarding set should be kept as small as possible.

In [5], Sun et al. proposed constructing the forwarding set based on the coverage area of 1-hop neighbors. The idea is to ensure nodes in the forwarding set of a node to cover the same area as all its 1-hop neighbors. However, the algorithm works only when all nodes in the network have the same transmission radius. In this paper, we extend the work to heterogeneous networks in which nodes may have different transmission radii. The network topology is represented by disk graphs with bidirectional link. In other words, each node is associated with a transmission radius. A link exists between two nodes if the distance between them is not larger than the minimum radius of them. We propose to use the minimum local disk cover set as forwarding set. The minimum local disk cover set of a node is a subset of 1-hop neighbors, and its coverage is the same as the coverage of all 1-hop neighbors and cardinality is smallest. Thus, it would better alleviate the broadcast storm problem. We propose a divide-and-conquer algorithm to construct the minimum local disk cover set. Our algorithm has the advantage of needing 1-hop information and being an optimal solution with time complexity $O(nlogn)$.

The remaining of this paper is organized as follows. The related works are introduced in Chapter 2. In Chapter 3, network model is defined and the forwarding set problem is formulated as the minimum local disk cover set problem. Then the equivalence between minimum local disk cover sets and skyline sets is build. A divide-and-conquer algorithm is given to find skyline sets. In Chapter 4, the time complexity of proposed algorithm is derived. In Chapter 5, simulations are given to compare with other forwarding set algorithms and concludes this paper.

Chapter 2

Related Works

Many localized algorithms have been proposed to alleviate the broadcast storm problem in wireless ad hoc networks. Localized algorithm refers that the nodes build their view of the network topology by using local information only, that is, the information regarding up to h-neighbors in the graph, where h is a small constant (2-3 at most) [18]. We introduce some localized forwarding schemes, including the multipoint relays (MPRs) [15], selecting forwarding sets [6] and minimum disk cover sets [5] in the following sections.

2.1 Multipoint Relays

Qayyum et al. [15] introduced the scheme of multipoint relays (MPRs) to forward broadcasting messages in the wireless networks. The idea of multipoint relays is to reduce the number of redundant retransmissions while broadcasting messages. The mechanism reduces the forwarding set to a small set of neighbors instead of all neighbors like in flooding.

Qayyum et al. proposed a scheme to select these MPRs in a wireless environment. The heuristic is adapted from the Chvătal's greedy algorithm [3] for Set Cover, and gives an approximation ratio of $O(\log \Delta)$, where Δ is the maximum number of 1-hop neighbors. The heuristic iteratively chooses a 1-hop neighbor covering the maximum number of 2-hop neighbors not yet covered as multipoint relays, and completes when all 2-hop neighbors have been covered. The information requiring for the multipoint relays is the 1-hop neighbors and 2-hop neighbors' information of each node. Multipoint relays scheme works in a distributed manner. Each node calculates its own MPRs, which is completely independent of other nodes' selection of their MPRs. In [2], multipoint relays are used to transmit control messages from a mesh point into the network.

2.2 Selecting Forwarding Sets

Călinescu et al. [6] solved the Minimum Forwarding Set problem for the geometric aspect. The problem is formulated as that given a source node and the sets of 1-hop and 2-hop neighbors. The idea is to select a minimum subset of 1-hop neighbors which dominates all 2-hop neighbors. The Minimum Forwarding Set problem is s special case of the Unit-Disk Cover problem known as NP-hard. The complexity of the Minimum Forwarding Set problem is not known.

Călinescu et al. proposed several heuristics with constant approximation ratios based on partition the plane into quadrant and established basic geometric properties of the 1-hop and 2-hop neighboring sets in wireless networks. A 2-approximation $O(nlogn)$ time algorithm is proposed, when all 2-hop neighbors are in the same quadrant with respect to the source node. First, the algorithm computes the skyline and numbers the skyline disks in counter-clockwise order. Then, each 2-hop neighbors are constructed the intervals covered by the skyline disks. Finally, using a simple greedy algorithm picks the disks.

The work needs to collect 1-hop and 2-hop information. The algorithms work only when the network model is homogeneous.

2.3 Minimum Disk Cover Sets

Sun et al. [5] show that the concept of the disk cover set. Given the set of disks S, the problem of disk cover set is to find a minimum subset of S, say $S\prime$, such that the union of the disks in $S\prime$ is equal to the union of the disks in S. All nodes in wireless ad hoc networks have the same transmission range in two-dimensional. Constructing the minimum disk cover set is based on the coverage area of 1-hop neighbors. The nodes in minimum disk cover set cover the same area as all one-hop neighbors. A node can simply request the neighbors in the minimum disk cover set to rebroadcast messages.

Sun et al. proved that finding the minimum disk cover set for disks is equivalent to finding the arcs of disks that make up the boundary. Then, a divide-and-conquer algorithm is proposed to construct the minimum disk cover set efficiently in homogeneous networks. The time complexity of construction the minimum disk cover set is $O(nlogn)$, where n is the number of 1-hop neighbors. Any algorithm that solves the minimum disk cover set problem needs at least $O(nlogn)$ time in the worst case. Therefore, based on 1-hop information, the algorithm is optimal in the aspect of time complexity.

Chapter 3

Minimum Local Disk Cover Sets

3.1 Network Model

To alleviate the broadcast storm problem, the size of the forwarding set needs to be reduced. However, at the same time, we need to make sure that messages may reach all nodes in the network. The selection of the forwarding set at each node should guarantee that the message will reach all its 2-hop neighbors.

We assume that wireless nodes are distributed in a two-dimensional plane R^2. In what follows, $\|x\|$ is the Euclidean norm of a point $x \in R^2$, and $|A|$ is shorthand for the cardinality of a countable set A. For any two points x, $y \in R^2$, $\|x - y\|$ is the Euclidean distance between x and y; $\overline{xy}$ denotes a line segment between x and y; and $\overrightarrow{xy}$ denotes a ray (or called a half line) from x to y.

The topology of a heterogeneous ad hoc network is modeled by disk graphs with bidirectional links. Each node u_i is associated with a transmission radius r_i. See Figure 3.1(a). A link exists between two nodes if the distance between them is not larger than their transmission radii. Two nodes u_i, u_j are said to be neighbors if and only if the Euclidean distance $\|u_i - u_j\| <= min(r_i, r_j)$.See

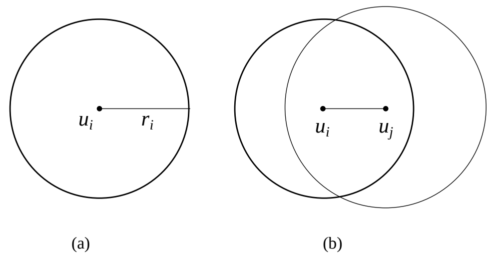


Figure 3.1: The configuration of a node u_i.

Figure 3.1(b). For a node u_i, its coverage is modeled as a disk with center u_i and radius r_i, which is denoted as $B(u_i, r_i)$. A node u_j is said to be covered by u_i if $u_j \in B(u_i, r_i)$. The topological boundary of a set $S \subset R^2$ is denoted by ∂S, and thus, $\partial B(u_i, r_i)$ is the circle centered at u_i and with radius r_i.

We assume each node knows the information of their 1-hop neighbors, especially locations and transmission radii. Due to the bidirectional link model, for any two nodes u, v, if u is the one-hop neighbor of v, then it implies v is the one-hop neighbor of u. So, the intersection of coverage of 1-hop neighbors is not empty, i.e. $\bigcap_{u_i \in S} B(u_i, r_i) \neq \varnothing$, where S is 1-hop neighbors.

3.2 Problem Description

We propose a strategy for each node to choose its forwarding set locally and statelessly based on the coverage of 1-hop neighbors. To make sure all 2-hop neighbors may receive messages, each node selects its forwarding set that the coverage of the forwarding set is the same as the coverage of all 1-hop neighbors.

We define the disk cover set of a set of nodes V as $S \subseteq V$ such that $\bigcup_{u_i \in S} B(u_i, r_i) = \bigcup_{u_i \in V} B(u_i, r_i)$. For a set of nodes V , if there exists a node in V such that all other nodes in V are neighbors to that node, then V is called the local set and the corresponding disk set, $\{B(u_1, r_1), ..., B(u_n, r_n)\}$, is called the local disk set. For a local set, its cover set is called a local disk cover set. Obviously, the closed neighbor set of a relay node n_0 is a local set.

To alleviate the broadcast storm problem and ensure the reachability of broadcast messages, we propose to use the minimum local disk cover set (MLDCS) as the forwarding set. The problem of the minimum local disk cover set is formally stated as follows.

Minimum Local Disk Cover Set (MLDCS) Problems

Input: A local disk set $\{B(u_0, r_0), B(u_1, r_1), ..., B(u_n, r_n)\}$
such that for all $1 \le i \le n$, $\|u_0 - u_i\| \le \min(r_0, r_i)$,
i.e. $u_i \in B(u_0, r_0)$ and $u_0 \in B(u_i, r_i)$.
Let $V = \{u_0, u_{1,}..., u_n\}$ be the set of disk centers.

Output: A subset S of V such that $\bigcup_{u_i \in S} B(u_i, r_i) = \bigcup_{u_i \in V} B(u_i, r_i)$.

Measure: $|S|$ is minimal.

Figure 3.2 illustrates the minimum local disk cover set. Assume source node is **o**. Node **o** has five one-hop neighbors $u_1, u_2, ...u_5$ and five two-hop neighbors $u_6, u_{7,}..., u_{10}$. We find the union of $B(u_1, r_1), B(u_2, r_2), B(u_4, r_4), B(u_5, r_5) = (\bigcup_{i=0}^{5} B(u_i, r_i))$. $B(u_3, r_3)$ is covered by $(\bigcup_{i=0}^{5} B(u_i, r_i))$, so $B(u_3, r_3)$ isn't in the minimum local disk cover set. The collection of these disks is minimum local disk cover set. The coverage of MLDCS is equal to the coverage of all one-hop neighbors, and the cardinality of MLDCS is 4.

We assume that each node learns the locations and radii of its neighbors through the exchanges of beacons. In addition, we define the *skyline* for a disk set as the boundary of the union of disks in the set. Hence, the skyline of the local disk set $\{B(u_0, r_0), B(u_1, r_1), \ldots, B(u_n, r_n)\}$ is $\partial(\bigcup_{i=0}^{n} B(u_i, r_i))$. Obviously, a skyline is composed of arcs of disks. The collection of origins of disks that contribute arcs to a skyline is called the *skyline set*. In the next

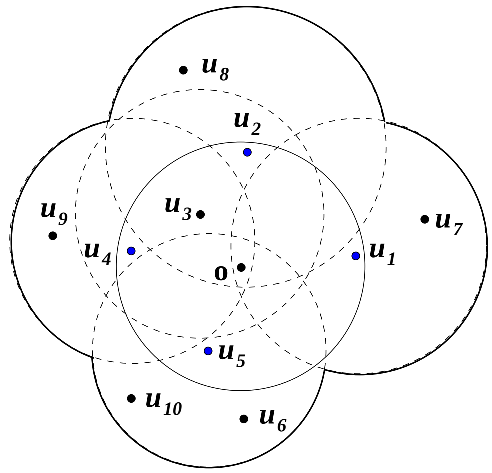


Figure 3.2: The source node **o** relays the message to MLDCS.

section, we will show that the MLDCS of a local set is the skyline set of the corresponding local disk set, and thus, we can solve the MLDCS problem for a given local disk set by computing the corresponding skyline.

3.3 The Geometry of Skyline Sets

In this section, we give properties of skylines and build the relation between the MLDCS for a given local set and the skyline set for the corresponding local disk set. We then propose a divide-and-conquer algorithm to compute the skyline set.

The following geometric lemma and corollary are used to build the relation between MLDCS for a local set $V = \{u_0, u_1, \ldots, u_n\}$ and the skyline set for the corresponding local disk set $\{B(u_0, r_0), B(u_1, r_1), \ldots, B(u_n, r_n)\}$. Without loss of generality, in the following discussion we assume $n_0 = \mathbf{o}$.

Lemma 1 *For any point $a \in \partial B(u_i, r_i)$, the line segment $\overline{\mathbf{o}a} \subset B(u_i, r_i)$.*

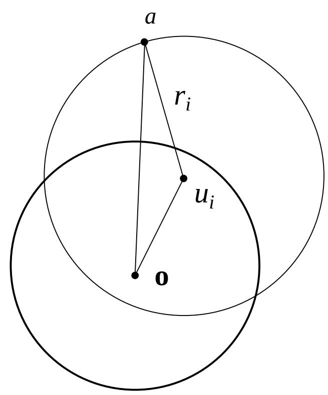


Figure 3.3: $\overline{\mathbf{o}a}$ is contained in $B(u_i, r_i)$.

Proof. Note that $\mathbf{o} \in B(u_i, r_i)$ since $\|\mathbf{o} - u_i\| \leq r_i$. Since $B(u_i, r_i)$ is convex and $\mathbf{o}, a \in B(u_i, r_i)$, the line segment $\overline{\mathbf{o}a} \subset B(u_i, r_i)$. See Figure 3.3. ■

Then, we have the following corollary.

Corollary 2 *For any ray from* **o***, it intersects the skyline at exactly one point.*

Proof. Obviously, any ray from **o** intersects the skyline. We only need to show the uniqueness of the intersection point. We can prove this by contradiction. Assume there exists a ray that intersects the skyline at two points a and b. Without loss of generality, we also assume a is farther from **o** than b. Since a is in the skyline, a is in $\partial B(u_i, r_i)$ for some i. According to Lemma 1, we have $\overline{\mathbf{o}a} \subset B(u_i, r_i)$. This implies that b is inside of $B(u_i, r_i)$, and b can't be in the skyline. It is a contradiction, and thus the corollary is proved. ■

According to the corollary, we can know that a skyline is composed of a sequence of arcs surrounding the origin from 0 to 2π. An arc can be represented by four parameters $(\alpha_i, u_j, r_{u_j}, \alpha_{i+1})$. Here u_j and r_j respectively are the center and radius of the disk contributing the arc, and α_i and α_{i+1} with $\alpha_i < \alpha_{i+1}$ are two angles corresponding to two endpoints of the arc measured at **o** in counterclockwise direction from the x-axis. See Figure 3.4. Note that the

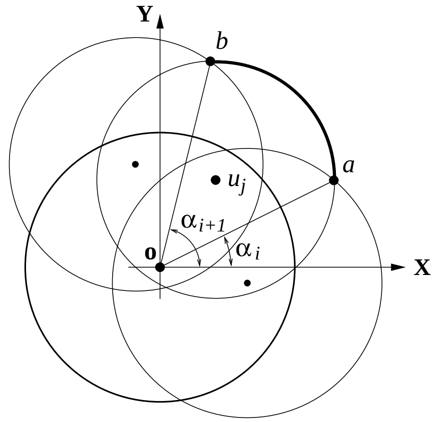


Figure 3.4: An arc ab is represented by 4 parameters $\left(\alpha_i, u_j, r_{u_j}, \alpha_{i+1}\right)$.

reference point to measure α_i and α_{i+1} is $\mathbf{o}$, not u_i. For convenience, if an arc crosses the x-axis in the positive direction then it is splitting by the x-axis into two arcs, i.e. an arc $\left(\alpha_i, u_j, r_{u_j}, \alpha_{i+1}\right)$ with $\alpha_i < 360^{\circ}$ and $\alpha_{i+1} > 360^{\circ}$, then we split this arc into 2 arcs $(\alpha_i, u_j, r_{u_j}, 360^{\circ})$ and $\left(0^{\circ}, u_j, r_{u_j}, \alpha_{i+1}\right)$. Without loss of generality, we could assume that there are no arcs exceeding 360°. Thus, a skyline consisting n arcs can be represented as $(\alpha_0, u_{s_0}, r_{s_0}, \alpha_1, u_{s_1}, r_{s_1}, \alpha_2, ..., \alpha_n)$, where $0 = \alpha_0 < \alpha_1 < ... < \alpha_n = 2\pi$ and for any $0 \leq i \leq n-1$, $(\alpha_i, u_{s_i}, r_{s_i}, \alpha_{i+1})$ is an arc in the skyline.

Now, we give the following theorem that builds the relation between a skyline set and the corresponding minimum local disk cover set.

Theorem 3 *For a given local set* $V = \{u_0, u_{1,} ..., u_n\}$*, its minimum local disk cover set is the skyline set for the corresponding local disk set* $\{B(u_0, r_0), B(u_1, r_1), ..., B(u_n, r_n)\}$.

Proof. To prove that the skyline set is the minimum local disk cover set of a local set, we first prove that a skyline set is a local disk cover set. Assume the skyline is composed of arcs $a_1b_1, a_2b_2, \ldots, a_kb_k$ belonging to $B(u_{i_1}, r_{i_1}), B(u_{i_2}, r_{i_2}), \ldots, B(u_{i_k}, r_{i_k})$, respectively. Let $\sphericalangle \mathbf{o}a_jb_j$ denote the

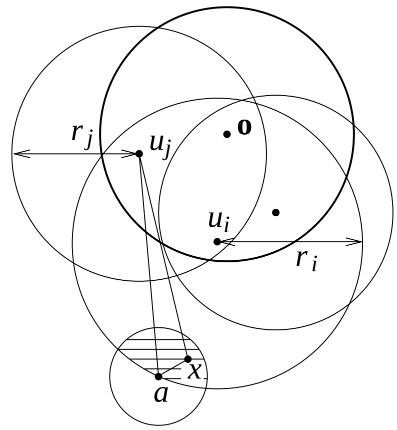


Figure 3.5: $B(a, r) \cap B(u_i, r_i)$ is exclusively covered by $B(u_i, r_i)$.

sector area surrounded by line segments $\overline{\mathbf{o}a_j}$, $\overline{\mathbf{o}b_j}$, and arc a_jb_j. Each sector has no intersection with others. The covered area $\bigcup_{i=0}^{n} B(u_i, r_i)$ is equal to the union of sectors $\bigcup_{j=1}^{k} \sphericalangle \mathbf{o}a_jb_j$. According to Lemma 1, for each skyline arc a_jb_j, we have $\sphericalangle \mathbf{o}a_jb_j \subseteq B\left(u_{i_j}, r_{i_j}\right)$. Thus, $\bigcup_{i=0}^{n} B(u_i, r_i) \subseteq \bigcup_{j=1}^{k} B\left(u_{i_j}, r_{i_j}\right)$. This means $\{B(u_{i_1}, r_{i_1}), B(u_{i_2}, r_{i_2}), \ldots, B(u_{i_k}, r_{i_k})\}$ is a disk cover set.

Next, we prove that if $B(u_i, r_i)$ is in the skyline set, $B(u_i, r_i)$ must be in any disk cover set. To prove this, we are going to show there exists some area belonging to $B(u_i, r_i)$ but not belonging to any other disk. Assume a is a point in the internal of the skyline arc contributed by $B(u_i, r_i)$. For any $B(u_i, r_i)$, $j \neq i$, we have the distance between u_j and a is larger than r_j, i.e. $\|u_j - a\| > r_j$. Let $r = \frac{1}{2}\left(\min_{j \neq i} \|u_j - a\| - r_j\right)$. We may draw a disk which is with center located at a and whose radius is r. For any point $x \in B(a, r)$ and $j \neq i$, see Figure 3.5.

$$
\begin{aligned}
\|u_j - x\| &\geq \|u_j - a\| - \|x - a\| \\
&\geq \|u_j - a\| - \frac{1}{2}\left(\min_{j \neq i} \|u_j - a\| - r_j\right) \\
&> \|u_j - a\| - (\|u_j - a\| - r_j) \\
&= r_j.
\end{aligned}
$$

Then we have $\|u_j - x\| \geq r_j$. Thus, for any $j \neq i$, $B(a, r) \cap B(u_j, r_j) = \varnothing$. This implies that $B(a, r) \cap B(u_i, r_i)$ belongs to $B(u_i, r_i)$ but does not belong to any other disk, thus $B(a, r) \cap B(u_i, r_i)$ is exclusively covered by $B(u_i, r_i)$. So the theorem is proved. ■

3.4 A Divide-and-Conquer Algorithm

According to Theorem 3, computing the MLDCS of a local set is the same as finding the skyline set of the corresponding local disk set. In this subsection, a divide-and-conquer algorithm is proposed to find the skyline set.

In the algorithm, the disk set is divided into 2 subsets of disks $DS1$ and $DS2$ recursively. If the disk set is small enough (i.e. the size of disk set is 1), then the *skyline* returns the skyline of the disk. The skyline of each subset is discovered by recursive techniques, and then two skylines are merged to find the skyline of all disks. Eventually, *Skyline* returns a new skyline list. Without loss of generality, we may assume that the position of n_0 (i.e., $\mathbf{o}$) and the value r_0 are stored in the algorithms.

procedure $Skyline\left(DS = \{(u_1, r_1), ..., (u_n, r_n)\}\right)$
// (u_i, r_i) represents the center and radius of a disk.
begin
if $|DS| = 1$ **return** the skyline of $\{B(u_1, r_1)\}$
else // if $|DS| > 1$
    begin
    $DS1 = \left\{(u_1, r_{u_1}), ..., \left(u_{\lfloor \frac{n}{2} \rfloor}, r_{u_{\lfloor \frac{n}{2} \rfloor}}\right)\right\}$
    $DS2 = \left\{\left(u_{\lfloor \frac{n}{2} \rfloor + 1}, r_{u_{\lfloor \frac{n}{2} \rfloor + 1}}\right), .., (u_n, r_{u_n})\right\}$
    $Skyline1 = Skyline(DS1)$
    $Skyline2 = Skyline(DS2)$
    return $Merge(Skyline1, Skyline2)$
    end
end

procedure $Merge(SL1, SL2)$
// $SL1$ and $SL2$ are skylines.
begin
Step 1: Refine $SL1$ and $SL2$ to align arcs in skylines.
        Then, we may assume $SL1 = (\alpha_0, u_1, r_{u_1}, \alpha_1, ..., \alpha_m)$
        and $SL2 = (\alpha_0, v_1, r_{v_1}, \alpha_1, ..., \alpha_m)$
Step 2: For each $0 \leq i \leq m$, decide new skyline arcs from
        $(\alpha_i, u_i, r_{u_i}, \alpha_{i+1})$ and $(\alpha_i, v_i, r_{v_i}, \alpha_{i+1})$
        with the same angle span.
Step 3: Combine neighboring skyline arcs from the same disk.
Return the new skyline.
end

Skyline is a classical divide-and-conquer procedure, and most works are done in the procedure *Merge*. There are three steps in *Merge*.

First, two skylines are aligned by splitting arcs such that two skylines have the same angle sequences. For example, assume $SL1 = \left(\beta_0, u'_0, r'_{u_0}, \beta_1, u'_1, r'_{u_1}, \beta_2, ..., \beta_k\right)$ and $SL2 = \left(\gamma_0, v'_0, r'_{v_0}, \gamma_1, v'_1, r'_{v_1}, \gamma_2, ..., \gamma_l\right)$ are two skylines. Let $(\alpha_0, \alpha_1, \cdots, \alpha_m)$ be a monotonic sequence of angles such that $\{\alpha_0, \alpha_1, \cdots, \alpha_m\} = \{\beta_0, \beta_1, ..., \beta_k\} \cup \{\gamma_0, \gamma_1, ..., \gamma_l\}$. Then, $SL1$ and $SL2$ are refined according to angles $\alpha_0, \alpha_1, \cdots, \alpha_m$. After refinement, both lists should have the same angles sequence and the same number of arcs, and we may assume $SL1 = (\alpha_0, u_1, r_{u_1}, \alpha_1, ..., \alpha_m)$ and $SL2 = (\alpha_0, v_1, r_{v_1}, \alpha_1, ..., \alpha_m)$.

For instance, there are two skyline lists,

$$\begin{cases} SL1 = \left(0^\circ, u'_0, r'_{u_0}, 40^\circ, u'_1, r'_{u_1}, 100^\circ, u'_2, r'_{u_2}, 150^\circ\right) \\ SL2 = \left(0^\circ, v'_0, r'_{v_0}, 60^\circ, v'_1, r'_{v_1}, 120^\circ\right) \end{cases}$$

We split the skyline lists into smaller arcs. After the step 1 in procedure $Merge$, they will be refined to

$$\begin{cases} SL1 = \left(0^\circ, u'_0, r'_{u_0}, 40^\circ, u'_1, r'_{u_1}, 60^\circ, u'_1, r'_{u_1}, 100^\circ, u'_2, r'_{u_2}, 120^\circ, u'_2, r'_{u_2}, 150^\circ\right) \\ SL2 = \left(0^\circ, v'_0, r'_{v_0}, 40^\circ, v'_0, r'_{v_0}, 60^\circ, v'_1, r'_{v_1}, 100^\circ, v'_1, r'_{v_1}, 120^\circ, v'_1, r'_{v_1}, 150^\circ\right) \end{cases}$$

In the second step, for each $0 \leq i \leq m$, new skyline arcs are decided from $(\alpha_i, u_i, r_{u_i}, \alpha_{i+1})$ and $(\alpha_i, v_i, r_{v_i}, \alpha_{i+1})$. Given two arcs (α, u, r_u, β) and (α, v, r_v, β) with the same angle span, we have following three cases to decide the new skyline arc.

Case 1: $arc(\alpha, u, r_u, \beta)$ and $arc(\alpha, v, r_v, \beta)$ have no intersection. One arc is closer to **o** than the other, and the arc closer to **o** will be removed from the skyline. The outer arc remains on the skyline. For instance, in Figure 3.6(a), arc ab and cd have no intersection point. Arc ab are the new skyline arc of arcs ab and cd.

Case 2: $arc(\alpha, u, r_u, \beta)$ and $arc(\alpha, v, r_v, \beta)$ intersect at one point e. Let γ be the angle corresponding to the intersection point. Applying the principle

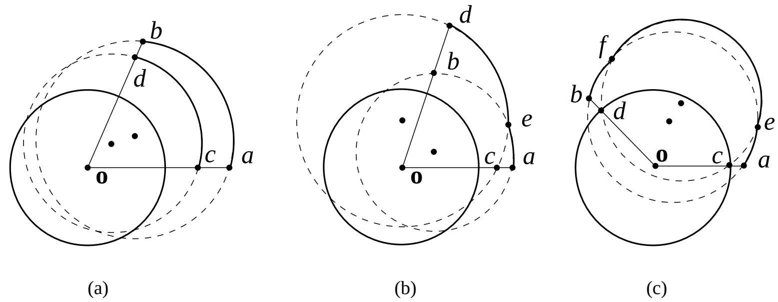


Figure 3.6: Merge two arcs with the same corresponding angles.

used in **Case 1**, new skyline arcs can be decided from $arc(\alpha, u, r_u, \gamma)$ and $arc(\alpha, v, r_v, \gamma)$; and $arc(\gamma, u, r_u, \beta)$ and $arc(\gamma, v, r_v, \beta)$. For instance, in Figure 3.6(b), e is the intersection point of arcs ab and cd. Arcs ae and ed are the new skyline arcs of arcs ab and cd.

Case 3: $arc(\alpha, u, r_u, \beta)$ and $arc(\alpha, v, r_v, \beta)$ intersect at two points e, f. Let γ_1 and $\gamma_2 (\gamma_1 < \gamma_2)$ be the angles corresponding to the intersection points. Applying the principle used in **Case 1**, new skyline arcs can be decided from $arc(\alpha, u, r_u, \gamma_1)$ and $arc(\alpha, v, r_v, \gamma_1)$; $arc(\gamma_1, u, r_u, \gamma_2)$ and $arc(\gamma_1, v, r_v, \gamma_2)$; and $arc(\gamma_2, u, r_u, \beta)$ and $arc(\gamma_2, v, r_v, \beta)$. For instance, in Figure 3.6(c), e, f is the intersection points of arcs ab and cd. Arcs ae, ef and fb are the new skyline arcs of arcs ab and cd.

In the first and second steps, one arc may be split into several pieces. So, in the last step, before returning the new skyline, we try to combine neighboring skyline arcs if they are from the same disk. This step could reduce the overhead in splitting skyline lists, if there are many small arcs in the refined skyline lists.

Chapter 4

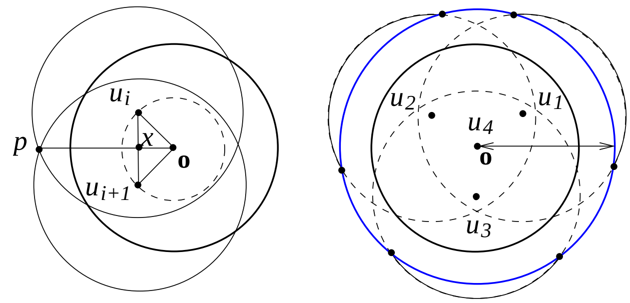
Time Complexity Analysis

In this chapter, we show that the time complexity of the proposed algorithm is $O(n \ln n)$. In [5], it has been shown that the time complexity of the algorithm that computes the minimum local disk cover set for homogeneous networks (i.e., all nodes have the same radius) is bounded by $O(n \ln n)$. Since homogeneous networks are special cases of heterogeneous networks, the time complexity for the algorithm that computes the minimum local disk cover set for heterogeneous networks is at least $O(n \ln n)$. Hence, the proposed algorithm is with the optimal time complexity.

Skyline is a divide-and-conquer algorithm. The time complexity can be formulated by a recursive equation

$$\begin{cases} T(n) = O(1) \text{ if } n = 1, \\ T(n) = 2T\left(\frac{n}{2}\right) + f(n) \text{ otherwise.} \end{cases}$$

Here $T(n)$ is the time complexity of *Skyline*. $f(n)$ is the time complexity of $Merge$. $f(n)$ is linear with respect to the number of arcs. If we may have $f(n) = O(n)$, according to the master theorem [22], then we have $T(n) = O(n \log n)$. In Lemma 8, we will show that the number of arcs in a skyline is upper bounded by $2n$.



(a) (b)

Figure 4.1: The configuration $B(u_4, r_4)$ contributes 3 arcs in the skyline.

First we present the geometric arguments and related lemmas to support the proof in the following section. Then we give the time complexity of the algorithm.

4.1 Geometric Property of Disks

We would like to show that the number of arcs in a skyline can not be more than $2n$. This can be proved that if we can show that adding a disk into a disk set increases the number of skyline arcs by at most 2. However, this is not true in the general case. For a positive integer $k \geq 3$, let $u_i = \left(\frac{1}{2}\cos\frac{2\pi i}{k}, \frac{1}{2}\sin\frac{2\pi i}{k}\right)$ for any $i = 0, 1, \cdots, k-1$, the center of these disks averagely located on $\partial B\left(\mathbf{o}, \frac{1}{2}\right)$. In Figure 4.1(a), p is an intersection point of $\partial B\left(u_i, 1\right)$ and $\partial B\left(u_{i+1}, 1\right)$ outside $B\left(\mathbf{o}, 1\right)$, and x is the intersection point of $\overline{u_i u_{i+1}}$ and $\overline{op}$. We have $\|u_i - x\| = \frac{1}{2}\sin\frac{\pi}{k}$, $\|x - p\| = \sqrt{1 - (\frac{1}{2}\sin\frac{\pi}{k})^2}$, and $\|\mathbf{o} - p\| = \frac{1}{2}\cos\frac{\pi}{k} + \sqrt{1 - (\frac{1}{2}\sin\frac{\pi}{k})^2}$. Let $r \in \left(\|\mathbf{o} - p\|, 1 + \frac{1}{2}\right)$ be a constant. If we add $B\left(\mathbf{o}, r\right)$ to the disk set $\{B\left(\mathbf{o}, 1\right), B(u_1, r_2), B(u_2, r_2), ..., B(u_k, r_k)\}$, then $B\left(\mathbf{o}, r\right)$ contributes k disjoint arcs to the skyline and doesn't fully cover any previous skyline arc. Thus, adding $B\left(\mathbf{o}, r\right)$ increases the number of skyline arcs by k. Figure 4.1(b) illustrates the configuration, for $k = 3$, $B(u_4, r_4)$ contributes 3 arcs in the skyline, where $u_4 = \mathbf{o}$ and $r_4 \in \left(\left\| \frac{1}{2} \cos \frac{\pi}{3} + \sqrt{1 - (\frac{1}{2} \sin \frac{\pi}{3})^2} \right\|, 1 + \frac{1}{2} \right)$.

However, if the disk is added into the disk set follows the decreasing order of radius, the claim will be true. Since the order to add disks into the disk set doesn't change the final skyline, we change the arrangement of disks added. Therefore, in what follows, we assume disks are added according to their radii in the decreasing order of radius.

In the following, we introduce some lemmas about the geometry of disks.

Lemma 4 *In the skyline of $B_1, B_2, \ldots, B_n$, if B_n contributes at least 3 arcs, then we can pick B_i, B_j, B_k from $B_1, B_2, \ldots, B_{n-1}$ such that B_n contributes 3 arcs in the skyline of B_i, B_j, B_k, B_n.*

Proof. First, we claim that if B_n contributes at least 3 arcs in the skyline of B_i, B_j, B_k, B_n, then B_n contributes exactly 3 arcs in the skyline. Assume there are more than 3 arcs contributed by B_n. Since each arc is with 2 endpoints, there are at least 8 intersection points in ∂B_n. But $\partial B_i, \partial B_j, \partial B_k$ have at most 6 intersection points with ∂B_n. This is a contradiction. Thus, our claim is true. So, to prove the lemma, it is enough to show that B_n contributes at least 3 arcs in the skyline of B_i, B_j, B_k, B_n.

Choose 3 arcs contributed by B_n in the skyline of $B_1, B_2, \ldots, B_n$, and consider those disks whose boundaries intersect with ∂B_n at endpoints of the 3 arcs. These 6 endpoints are from intersection points of B_n with another 3, 4, 5, or 6 disks since the boundary of each disk can contribute either 1 or 2 intersection points with ∂B_n in the skyline. If there are 3 disks, the lemma is proved. So, we consider the case that there are more than 3 disks. By counting the endpoints, if there are more than 3 disks, at lease one intersects with ∂B_n exactly at 1 point in the skyline. If we remove a disk which intersects ∂B_n at exact 1 point in the skyline, the number of arcs contributed by B_n

does not decrease. Thus, in the skyline of remaining disks, B_n contributes at least 3 arcs, but the number of disks decrease by 1. Figure 4.2(a) illustrates the configuration in which the skyline contains 4 disks B_1, B_2, B_3, B_4 and B_n contributes 3 arcs. The arcs of B_n are presented in dotted line. We remove the

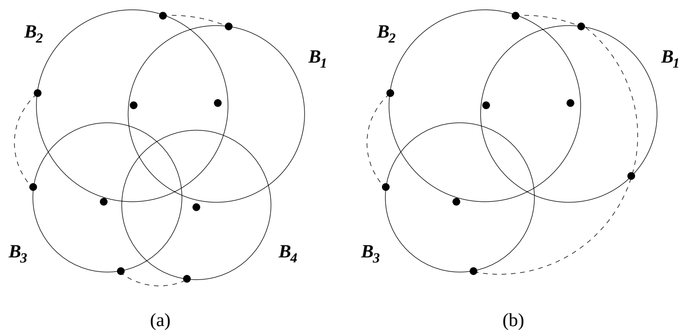


Figure 4.2: Proof of Lemma 4.

disk B_4 which intersects ∂B_n at exact 1 point in the skyline, and the number of arcs contributed by B_n doesn't decrease, see Figure 4.2(b). Then, we repeat the process until there remains 3 disks, and then the lemma is proved. ■

Lemma 5 *Assume two circle ∂B_1 and ∂B_2 have two intersection points a and d. Let ac' (and ab', respectively) be a diameters of B_1 (and B_2, respectively), and c (and b, respectively) is a point in arc $c'd$ ($b'd$ respectively). See Figure 4.3. If the angle $\angle cab$ is obtuse, we have*

$$\|b - c\| > 2 \min (r_1, r_2) .$$

Proof. First, we explore an extreme case in which ∂B_1 and ∂B_2 are tangent, i.e. c', a, b' are in a line and a, d are overlapping and $\measuredangle cab$ is $\frac{\pi}{2}$. See Figure 4.4. Since $\overline{ac'}$ ($\overline{ab'}$ respectively) is the diameter of B_1 (B_2 respectively), $\measuredangle c'ca$ ($\measuredangle b'ba$ respectively) is right angle. $\Delta acc'$ and $\Delta abb'$ are similar. If $r_1 \leq r_2$,

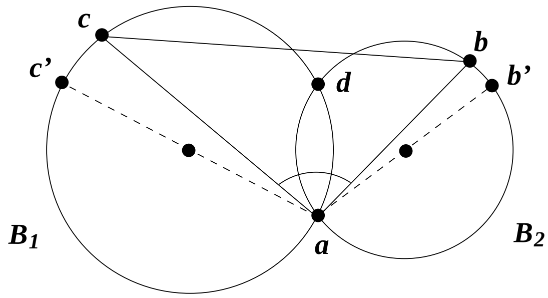


Figure 4.3: The structure in Lemma 5.

since $\measuredangle c'ca$ and $\measuredangle b'ba$ are also right angles, we have

$$\begin{aligned}
\|b-c\|^2 &> \|a-c\|^2 + \|a-b\|^2 \\
&= \|a-c\|^2 + \left(\frac{r_2}{r_1}\|c'-c\|\right)^2 \\
&= \|a-c\|^2 + \|c'-c\|^2 + \left(\left(\frac{r_2}{r_1}\right)^2 - 1\right)\|c'-c\|^2 \\
&= (2r_1)^2 + \left(\left(\frac{r_2}{r_1}\right)^2 - 1\right)\|c'-c\|^2 \geq (2r_1)^2 = (2\min(r_1,r_2))^2 .
\end{aligned}$$

Similarly, if $r_2 \leq r_1$, we also have

$$\begin{aligned}
\|b-c\|^2 &> \|a-c\|^2 + \|a-b\|^2 \\
&= \|a-b\|^2 + \left(\frac{r_1}{r_2}\|b'-b\|\right)^2 \\
&= \|a-b\|^2 + \|b'-b\|^2 + \left(\left(\frac{r_1}{r_2}\right)^2 - 1\right)\|b'-b\|^2 \\
&= (2r_2)^2 + \left(\left(\frac{r_1}{r_2}\right)^2 - 1\right)\|b'-b\|^2 \geq (2r_2)^2 = (2\min(r_1,r_2))^2 .
\end{aligned}$$

Thus $\|b-c\| > 2\min(r_1,r_2)$. The lemma is correct for this extreme case.

The inequality can be extended for general cases by the following simple observation. Rotate B_1 and/or B_2 by a to let $\measuredangle c'ab'$ become smaller but don't let ac' across ac and ab' across ab. Let c'' denote the intersection of the ray ac and ∂B_1 and b'' denote the intersection of the ray ab and ∂B_2.

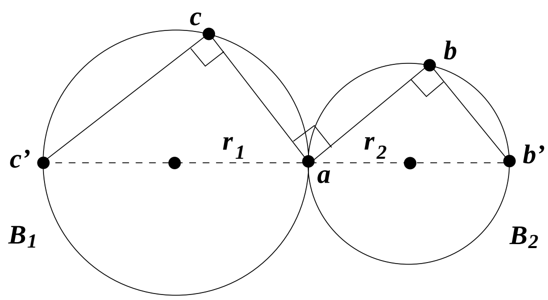


Figure 4.4: A extreme case when ∂B_1 and ∂B_2 are tangent.

See Figure 4.5. We have $\|a-b''\| \geq \|a-b\|$ and $\|a-c''\| \geq \|a-c\|$. Thus, $\|b''-c''\| \geq \|b-c\|$. So, the proof is complete.

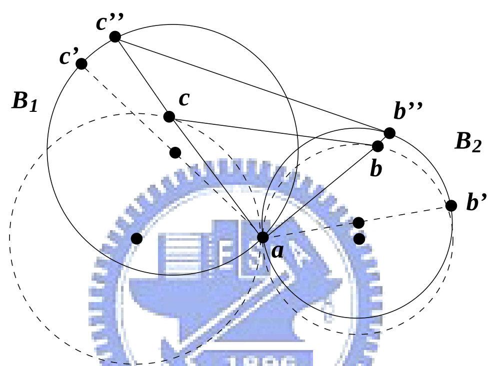


Figure 4.5: Rotate B_1 and B_2 by a, and then $\|b''-c''\| \geq \|b-c\|$.

■

Lemma 6 *Given an acute triangle, for each edge of the triangle, draw a circle with the edge as a chord whose center is outside the triangle and radius is equal to the circumradius. Then, three circles intersect at the orthocenter.*

Proof. Let Δabc be an acute triangle, C_1 be the circumcircle of Δabc, and C_2 (respectively, C_3 and C_4) be a circle with the circumradius of Δabc as its radius, and edge $\overline{ab}$ (respectively, $\overline{bc}$ and $\overline{ac}$) as a chord, and its center outside Δabc. See Figure 4.6(a). To prove this lemma, it is enough to show that C_2,

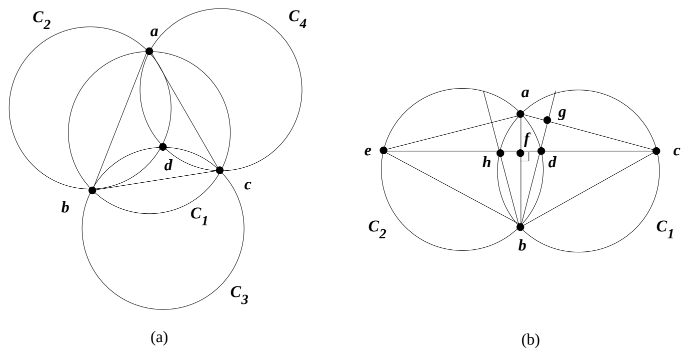


Figure 4.6: The configuration of Lemma 6.

C_3, C_4 contain the orthocenter of Δabc. Figure 4.6(b) illustrates the relation between C_1 and C_2, and we want to prove C_2 contains the orthocenter.

Let f denote the perpendicular foot on $\overline{ab}$. Let h denote the intersection point of C_1 and line cf, and e(respectively, d) denote the intersection points of C_2 and line cf outside (respectively, inside) Δabc. Let g be the intersection point of lines bd and ac. If we can show g is the perpendicular foot on $\overline{ac}$, then d is the orthocenter. Since C_1 and C_2 have the same radius, and $\overline{ab}$ and $\overline{ce}$ are perpendicular, $acbe$ is a rhombus. In the rhombus, $\angle aef = \angle acf$. Since $\angle aed$ and $\angle abd$ correspond to the same arc ad and they are inscribed angles on C_2, $\angle aed = \angle abd$. Thus, $\angle abd = \angle acf$. In Δdbf and Δdcg, $\angle dbf = \angle dcg$ and $\angle bdf = \angle cdg$, so $\angle bfd = \angle cgd$. Since $\angle bfd = 90°$, $\angle cgd = 90°$. $\overline{bg} \perp \overline{ac}$ and $\overline{cf} \perp \overline{ab}$, therefore d is the orthocenter of Δabc.

Similarly, we can prove C_3, C_4 also contain the orthocenter. So, the lemma is proved. ■

Then, we have the following corollary.

Corollary 7 *Given an acute or right triangle, for each edge of the triangle, draw a circle with the edge as a chord whose center is outside the triangle and radius is larger than the circumradius. Then, three circles have no intersection.*

We apply simple counting techniques. Since each arc has two endpoints, each endpoint is an intersection point of two disks, and two disks have at most 2 intersection points.

Lemma 8 *The number of arcs in a skyline of n disks is upper bounded by $2n$.*

Proof. We prove this by mathematical induction on n. Without loss of generality, we may assume all n nodes have arcs in the skyline and $B_1, B_2, \cdots, B_n$ have been sorted according to their radii in decreasing ordering.

If $n = 1$, there is only one disk, and thus, the skyline consists of one arc. If $n = 2$, two disks intersect at 2 points. There are at most 2 arcs in the skyline. See Figure 4.7(a). If $n = 3$, the topology can be categorized into 2 configurations, like Figure 4.7(b) and Figure 4.7(c). Figure 4.7(b) illustrates one configuration in which each disk contains one intersection point of the other two disks, and the skyline is composed of 3 arcs. Figure 4.7(c) illustrates the other configuration in which one disk contains two intersection points of the other two disks. In addition, 3 disks is allowed to have a common intersection point like Figure 4.8. The skyline is composed of 3 or 4 arcs. No matter how, the number of arcs is no more than $2n$.

Now, assume that as $n = k$, the skyline contains at most $2k$ arcs. If we can show that after a disk B_{k+1} is added into the set, the number of arcs in the skyline increases at most by two, then the new skyline contains no more than $2(k+1)$ arcs, and the proof is complete. Without loss of generality, we may assume B_{k+1} is the disk with the smallest radius among $B_0, B_1, B_2, \ldots, B_{k+1}$ since it doesn't change the skyline. We denote the number of arcs in the skyline of $B_1, B_2, \ldots, B_k$ as $Sky(B_1, B_2, \ldots, B_k)$. Now, we are going to prove this by contradiction. Assume B_{k+1} can contribute at least 3 arcs. According to Lemma 4, without loss of generality, we may assume that B_{k+1} contributes 3 arcs in the skyline of B_1, B_2, B_3, B_{k+1}. According to the two possible configurations of B_1, B_2, B_3 like Figure 4.7(b) or Figure 4.7(c), we discuss the problem in the two cases.

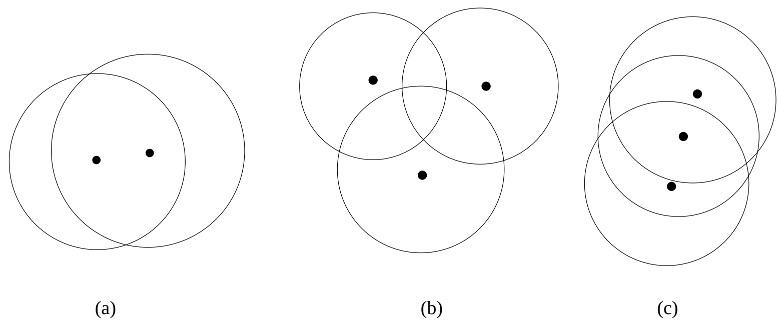


Figure 4.7: $n \leq 3$, the skyline contains $2n$ arcs at most.

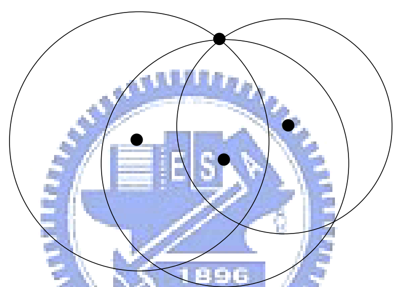


Figure 4.8: 3 disks have a common intersection point.

First, we consider the configuration like Figure 4.7(b). Let a be the intersection point of B_1 and B_2 not in B_3; b be the intersection point of B_1 and B_3 not in B_2; and c be the intersecting point of B_2 and B_3 not in B_1. In order to contribute 3 arcs, B_{k+1} must intersect with 3 disks and contain a, b, c. Now, the problem is discussed by following two cases: (1) Δabc is an acute or right triangle; and (2) Δabc is an obtuse triangle.

Case 1: Δabc is an acute or right triangle. Let r_c be the circumradius of Δabc. Since Δabc is an acute or right triangle and B_{k+1} contains a, b, c, we have r_{k+1} is larger than r_c. In addition, since B_{k+1} is the smallest among $B_0, B_1, B_2, \ldots, B_{k+1}$. So, we have $r_c < r_{k+1} \leq r_1, r_2, r_3$. But according to

Corollary 7, if r_1, r_2, r_3 are larger than r_c, B_1, B_2, B_3 have no intersection. This is a contradict to the fact that the intersection of B_1, B_2, B_3 is not empty.

Case 2: Δabc is an obtuse triangle. It doesn't affect the correctness of following argument if we assume $\angle cab$ is obtuse. Since B_{k+1} contribute three arcs in the skyline of B_1, B_2, B_3, B_{k+1} and $r_{k+1} \leq r_1, r_2, r_3$, degree of each arc in the skyline of B_1, B_2, B_3 must be larger than π. If ac' is a diameter of B_2 and ab' is a diameter of B_1, c' and b' are in the skyline of B_1, B_2, B_3. c is in the arc ac' intersected by arc ab' and not contained in B_1 and b is in the arc ab' intersected by arc ac' and not contained in B_2. According to Lemma 5, if $\angle cab$ is obtuse, we have $\|b - c\| > 2\min(r_1, r_2)$. On the other hand, since B_{k+1} contains Δabc, we have $r_{k+1} \geq \frac{1}{2}\|b - c\|$, and $2r_{k+1} \geq \|b - c\| > 2\min(r_1, r_2)$. Thus, we have a contradiction.

Next, we consider the configuration like Figure 4.7(c). Without affecting the correctness of following argument, we assume B_3 is the one containing two intersection point of the boundary of other two disk. Let b, e denote intersection points of B_1 and B_3, and c, f denote intersection points of B_2 and B_3. To contribute three arcs to the skyline of B_1, B_2, B_3, B_{k+1}, B_{k+1} should cover at least 3 intersection points. According to the number of covered intersection points, there are two cases.

Case 3: If B_{k+1} cover exactly 3 intersection points, like Figure 4.9(a), then B_{k+1} must have 3 intersection points with B_3. This is not possible to happen, since 2 disks have at most 2 intersection points.

Case 4: If B_{k+1} cover exactly 4 intersection points, like Fig. 4.9(b), let e, f denote the two intersection points covered by the same arc of B_{k+1} and a is a intersection point of B_1 and B_2 that is closer to the arc covering e, f. Since B_{k+1} is smaller than B_1, B_2, B_3, arcs bc of B_3 outside B_1, B_2, arc be of B_1 outside B_3, and arc cf of B_2 outside B_3 are larger than π. So, the angle $\measuredangle bac > \pi/2$, the diameter of B_1 with one endpoint at a is outside of B_2, and

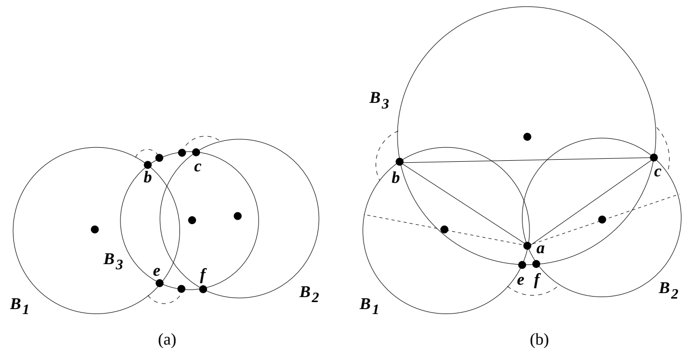


Figure 4.9: In the configuration, B_{k+1} doesn't contribute 3 arcs.

the diameter of B_2 with one endpoint at a is outside of B_1. According to Lemma 5, just like **Case 2**, we have $2r_{k+1} > \|b - c\| > 2\min(r_1, r_2)$. This is a contradiction.

According to previous discussion, B_{k+1} can't contribute 3 arcs to the skyline of $B_1, B_2, \cdots, B_{k+1}$, and therefore, the number of arcs in the skyline $B_1, B_2, \cdots, B_{k+1}$ is at most $2(k+1)$. By mathematical induction, we conclude that the number of arcs in the skyline of n disks is upper bounded by $2n$. ■

Now we show that our algorithm has time complexity $O(n \log n)$.

4.2 Time Complexity of Algorithm

Theorem 9 *The time complexity of Skyline is $O(n \log n)$, where n is the number of disks.*

Proof. The running time $T(n)$, of *Skyline* has the recursive equation

$$\begin{cases} T(n) = O(1) & \text{if } n = 1, \\ T(n) = 2T\left(\frac{n}{2}\right) + f(n) & \text{otherwise.} \end{cases}$$

If $n = 1$, we have $T(n) = O(1)$. If $n \geq 2$, *Skyline* executes *Skyline* twice but with half problem size and then executes *Merge* once. According to Lemma 8, the time complexity of *Merge* is $O(n)$. Hence, according the master theorem [22], $T(n) = O(n)$. The proposed algorithm has the time complexity $O(n \log n)$. ■

Chapter 5

Simulations and Conclusions

5.1 Simulations

Simulations are done to compare the performance of forwarding set algorithms, including the blind flooding, skyline algorithm, selecting forwarding set [6], greedy algorithm, and optimal algorithm. In the simulation, nodes are deployed over a 12.5×12.5 square. A source node u is placed at the center of the deployment region. If n is the average number of neighbors, $\frac{12.5^2}{\pi r^2} n$ nodes are generated with uniform distribution over the deployment region, where r is the transmission radius of a node. Two types of networks, homogeneous networks and heterogeneous networks, are considered. In homogeneous networks, all nodes are with the same transmission radius 1. In heterogeneous networks, every node may have different transmission radius that is generated by a function which produces randomly a real number between 1 and 2, including the source node. Bidirectional links are considered. 200 random point sets are generated in the simulation. For each random point set, the forwarding sets of node u are calculated by each algorithm. For each algorithm, we calculate the average size of forwarding set and the distribution of the size of forwarding sets.

Since it is still open that whether the minimum forwarding set problem of disk graphs is NP-Complete or not, we use brute force algorithm to find the minimum forwarding set. In the blind flooding, all 1-hop neighbors relay messages, so the number of forwarding nodes is equal to the number of 1-hop neighbors. In the skyline algorithm, each node chooses a subset of one-hop neighbors that contribute arcs to the skyline as its forwarding set. The selecting forwarding set algorithm introduced in [6] for homogeneous networks is with constant approximation ratios. In the greedy algorithm is that each node iteratively chooses a 1-hop neighbor covering the maximum number of 2-hop neighbors not yet covered, and completes when all 2-hop neighbors have been covered.

Excepting the skyline algorithm that only needs 1-hop information, the optimal algorithm, greedy algorithm, and selecting forwarding set algorithm [6], need one-hop and two-hop information to calculate the forwarding set. To obtain the information of 1-hop neighbors, each node periodically sent HELLO messages to report its current status. To obtain information of 2-hop information, the HELLO messages should contain the sender's 1-hop neighbor list are sent periodically.

5.1.1 Homogeneous Networks

In homogeneous networks, all nodes have the same transmission radii 1. We run simulation for the blind flooding, skyline algorithm, selecting forwarding set algorithm [6], greedy algorithm, and optimal algorithm. The average number of forwarding nodes of five algorithms is illustrated in Figure 5.1. The x-axis denotes the average number of 1-hop neighbors of node u. The y-axis denotes the average number of forward nodes. The five curves in the figure form top to down are corresponding to the blind flooding, skyline algorithm, selecting forwarding set [6], greedy algorithm and optimal algorithm, respectively.

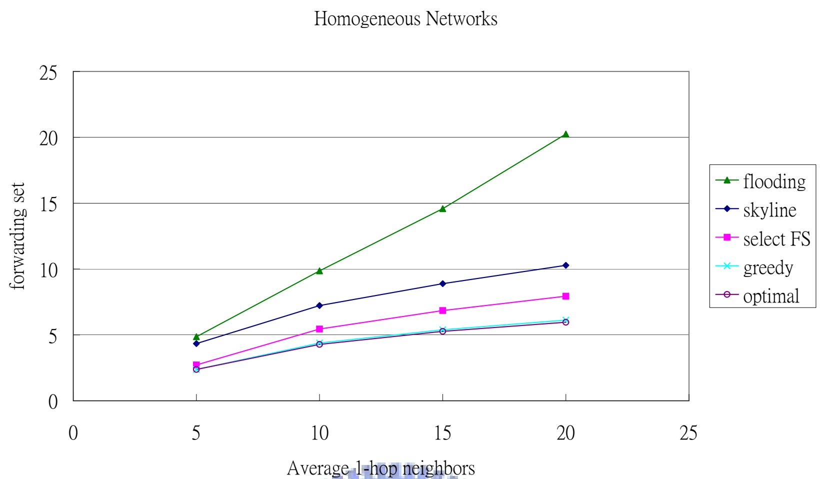


Figure 5.1: The algorithms are compared in homogeneous networks.

The selecting forwarding set algorithm and greedy algorithm usually generate smaller forwarding sets than the skyline algorithm. But instead of 2-hop information, the skyline algorithm needs only 1-hop information. So, the selecting forwarding set algorithm and greedy algorithm demand more resource than the skyline algorithm. In addition, if nodes have mobility, more efforts are needed to maintain 2-hop information then the nodes cost a lot of space and time in collecting two-hop information. Thus, the skyline algorithm is more easily to implement than the other two algorithms in wireless ad hoc networks.

Figure 5.2 and Figure 5.3 illustrate the distribution of the number of forwarding nodes. Figure 5.2 is of the network in which nodes have 10 1-hop neighbors in average, and Figure 5.3 is of the network in which nodes have 20 1-hop neighbors in average. The x-axis is corresponding to the number of forwarding nodes, and y-axis is corresponding to the number of random point sets.

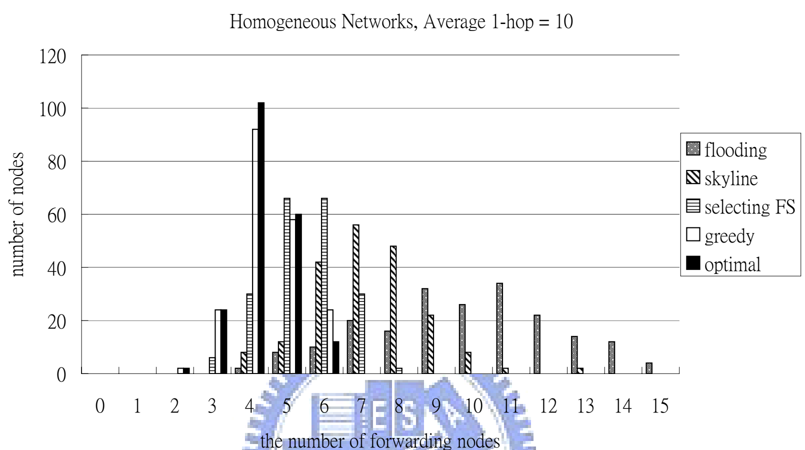


Figure 5.2: The distribution of forward nodes with 1-hop neighbors = 10.

If an algorithm is with better performance, in most cases, it generates forwarding sets with small size. So, the distribution is located at the left hand side in the figures. Without too much surprise, the greedy algorithm and selecting forwarding set algorithm generate smaller forwarding sets than the skyline algorithm. Note that in Figure 5.3, since the main part of forwarding nodes is distributed below 25, for simplicity, we show the main part instead of all, and there are some nodes with forwarding set larger than 25 nodes generated by the blind flooding algorithm.

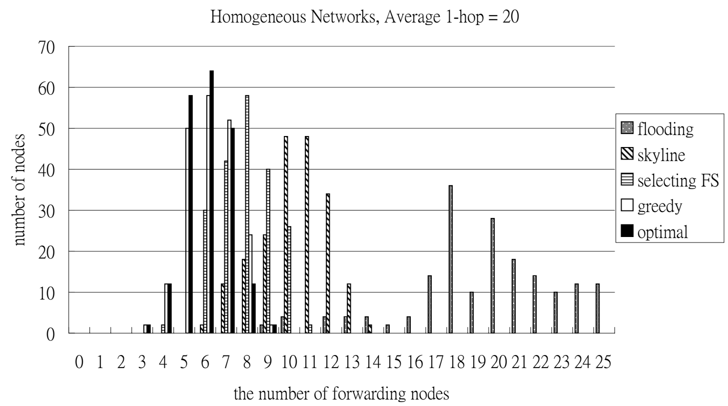


Figure 5.3: The distribution of forward nodes with 1-hop neighbors = 20.

5.1.2 Heterogeneous Networks

Since the selecting forwarding set algorithm [6] doesn't work for heterogeneous networks, we don't consider it in the simulation for heterogeneous networks. We run simulation for the blind flooding algorithm, skyline algorithm, greedy algorithm, and optimal algorithm. In heterogeneous networks, the transmission range of each node is randomly assigned between 1 and 2.

Figure 5.4 illustrated the average number of forward nodes. The x-axis denotes the average number of 1-hop neighbors of node u. The y-axis denotes the average number of forward nodes. In the figure, there are four curves, form top to down, corresponding to the blind flooding algorithm, skyline algorithm, greedy algorithm, and optimal algorithm.

Because we use the bidirectional link, the number of 1-hop of source node u may less than the average 1-hop neighbors.

Figure 5.5 illustrate the distribution of the number of forwarding nodes in

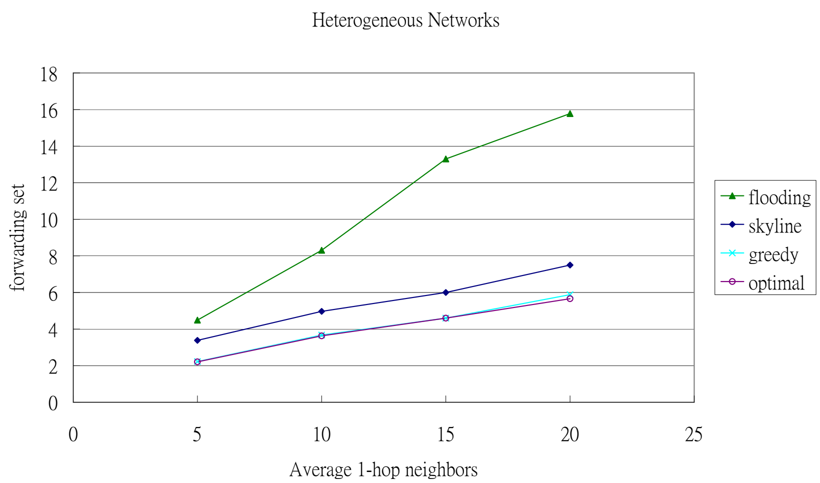


Figure 5.4: The algorithms are compared in heterogeneous networks.

networks in which nodes with 10 1-hop neighbors in average. The x-axis is corresponding to the number of forwarding nodes, and y-axis is corresponding to the number of random point sets.

Last, we want to point out a drawback of the skyline algorithm that is due to the directional links and bidirectional links. In Figure 5.6, node u has three 1-hop neighbors u_1, u_2, u_3 and u_4, u_5 are 2-hop neighbors. u_4 is a neighbor of u_1, and u_5 is a neighbor of u_2. The transmission range of u_3 can cover u_4 and u_5, but the transmission range of u_4 or u_5 can not cover u_3. So, u_4 and u_5 are not neighbors of u_3. The optimal forwarding set under the bidirectional link model is $\{u_1, u_2\}$, but the skyline set is $\{u_3\}$. Since the skyline algorithm utilizes only 1-hop information, it can't know the information about 2-hop neighbors. We leave this problem as our one of our future works.

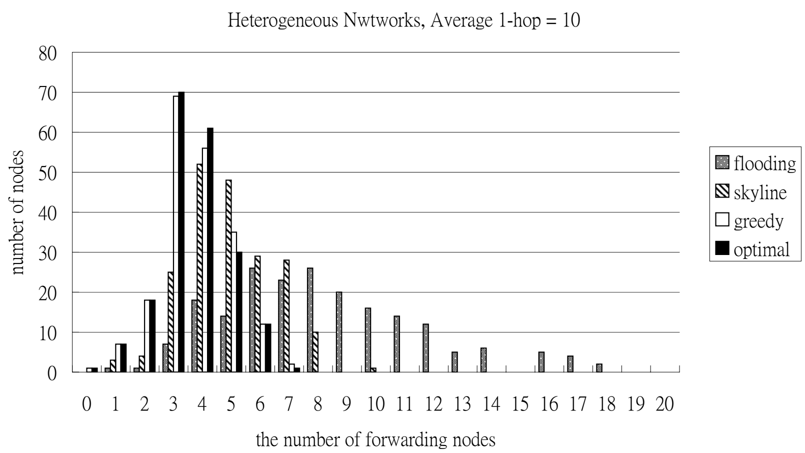


Figure 5.5: The distribution of forward nodes with 1-hop neighbors = 10.

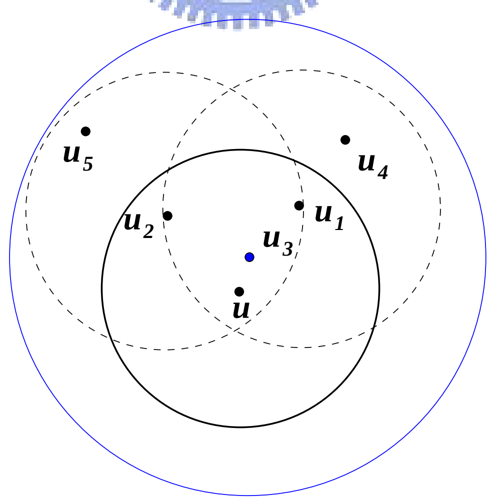


Figure 5.6: The special case in heterogeneous networks.

5.2 Conclusions

To relieve the broadcast storm problem in wireless ad hoc network, we suggest using the minimum local disk cover set as forwarding nodes for broadcasting instead of all 1-hop neighbors. In this work, we have established the equivalence of the MLDCS and the skyline set. We propose a divide-and-conquer algorithm to find the skyline set with the optimal time complexity $O(nlogn)$. In heterogeneous wireless networks, the skyline set that is based on 1-hop information, can't guarantee the coverage of 2-hop neighbors under bidirectional links. This drawback will be studied in our future works.

Bibliography

[1] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen and J.-P. Sheu, "The Broadcast Storm problem in a mobile ad hoc network," in: Proc. of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking(1999) pp. 151–162

[2] Aoki, et al. "802.11 TGs Simple Efficient Extensible Mesh (SEE-Mesh) Proposal," doc.:IEEE 802.11-05/0562r0(2005) pp. 65–68

[3] V. Chvătal, "A greedy heuristic for the set-covering problem," Mathematicsof Operation Research 4(3) (1979) 233–235.

[4] M. Sun and T. H. Lai, "Computing Optimal Local Cover Set for Broadcast in Ad Hoc Networks," Proc. IEEE ICC pp. 3291-3295, Apr. 2002.

[5] M.-T. Sun, X. Ma, C.-W. Yi, C.-K. Yang, and T. H. Lai, "Minimum Disc Cover Set Construction in Mobile Ad Hoc Networks," in the 2005 International Conference on Computer Networks and Mobile Computing (ICCNMC'05), Zhangjiajie, China, 2-4 August 2005.

[6] G. Călinescu, I. Mandoiu, P.J. Wan, and A. Zelikovsky, "Selecting Forwarding Neighbors in Wireless Ad Hoc Network," MONET 9(2): 101-111 (2004)

[7] P. Sinha, R. Sivakumar and B. Vaduvur, "Enhancing ad hoc routing with dynamic virtual infrastructures," in: Proc. of IEEE INFOCOM (2001) pp. 1763–1772.

[8] K.M. Alzoubi, P.J. Wan, and O. Frieder, "New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks," Proc. Hawaii Int'l Conf. System Sciences '35, 2002.

[9] W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks," IEEE Trans. Mobile Computing, vol. 1, no. 2, pp. 111-123, Apr.-June 2002.

[10] J. Wu and F. Dai, "Broadcasting in Ad Hoc Networks Based on Self-Pruning," Proc. IEEE Infocom, Mar.-Apr. 2003.

[11] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In Proc. DIAL M, pages 7–14, August 1999. Seattle, USA.

[12] H. Lim and C. Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks," Proc. ACM MSWiM, Aug. 2000.

[13] I. Stojmenovic, S. Seddigh, and J. Zunic, "Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks," IEEE Trans. Parallel and Distributed Systems, vol. 13, no. 1, pp. 14-25, Jan. 2002.

[14] J. Sucec and I. Marsic, "An Efficient Distributed Network-Wide Broadcast Algorithm for Mobile Ad Hoc Networks," CAIP Technical Report 248, Rutgers Univ., Sept. 2000.

[15] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Message in Mobile Wireless Networks," Proc. HICSS, vol. 9, p. 298, Jan. 2002.

[16] F. Dai and J. Wu, "Efficient Broadcasting in Ad Hoc Wireless Networks Using Directional Antennas," IEEE Trans on Parallel and Distributed Systems, VOL. 17, NO. 4, APRIL 2006

[17] B.M.E. Moret and H.D. Shapiro, "Algorithms from P to NP, Vol. I: Design and Efficiency" (Benjamin/Cummings, 1991).

[18] P. Santi, " Topology Control in Wireless Ad Hoc and Sensor Networks" (Wiley, 2005).

[19] B.N. Clark, C.J. Colbourn and D.S. Johnson, Unit disk graphs, Discrete Mathematics 86 (1990) 165–177.

[20] LAN MAN Standards Committee of the IEEE Computer Socity, editor. IEEE Std 802.11-1997, Wireless LAN Medium Access Control MAC and Physical Layer (PHY specifications. IEEE, Nov. 1997).

[21] E. Horowitz, S. Sahni, and S. Rajasekaran, " Computer algorithms " (Computer Science Press, 1997).

[22] T.H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, " Introduction to Algorithms " (The MIT Press, 2nd edition, 2001)