EFFICIENT DATA DISSEMINATION IN WIRELESS AD HOC NETWORKS

Louisa Harutyunyan

A thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
Concordia University
Montréal, Québec, Canada

August 2014

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Miss Louisa Harutyunyan**

Entitled: **Efficient Data Dissemination in Wireless Ad Hoc Networks**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

______________________ Chair
Dr. Glenn Cowan

______________________ External Examiner
Dr. Michel Barbeau

______________________ Examiner
Dr. Chadi Assi

______________________ Examiner
Dr. Thomas Fevens

______________________ Examiner
Dr. Lata Narayanan

______________________ Supervisor
Dr. Jaroslav Opatrny

Approved ______________________
Chair of Department or Graduate Program Director

__________ 20 ______ ______________________

Amir Asif, Ph.D., Dean of Faculty of
Engineering and Computer Science

Abstract

Efficient Data Dissemination in Wireless Ad Hoc Networks

Louisa Harutyunyan, Ph.D.
Concordia University, 2014

In this thesis, we study the problem of efficient data dissemination in wireless sensor and mobile ad hoc networks. In wireless sensor networks we study two problems: (1) construction of virtual backbones and clustering hierarchies to achieve efficient routing, and (2) placement of multiple sinks, where each sensor is at a bounded distance to several sinks, to analyze and process data before sending it to a central unit. Often connected dominating sets have been used for such purposes. However, a connected dominating set is often vulnerable due to frequent node failures in wireless sensor networks. Hence, to provide a degree of fault-tolerance we consider in problem (1) a 2-connected (k, r)-dominating set, denoted $D_{(2,k,r)}$, to act as a virtual backbone or a clustering hierarchy, and in problem (2) a total (k, r)-dominating set to act as sinks in wireless sensor networks.

Ideally, the backbone or the number of sinks in the network should constitute the smallest percentage of nodes in the network. We model the wireless sensor network as a graph. The total (k, r)-dominating set and the 2-connected (k, r)-dominating set have not been studied in the literature. Thus, we propose two centralized approximation algorithms to construct a $D_{(2,k,r)}$ in unit disk graphs and in general graphs. We also derive upper bounds on the total (k, r)-domination number in graphs of girth at least $2k + 1$ as well as in random graphs with non-fixed probability p.

In mobile ad hoc networks we propose a hexagonal based beacon-less flooding algorithm, HBLF, to efficiently flood the network. We give sufficient condition that even in the presence of holes in the network, HBLF achieves full delivery. Lower and upper bounds are given on the number of forwarding nodes returned by HBLF in a network with or without holes. When there are no holes in the network, we show that the ratio of the shortest path returned by HBLF to the shortest path in the network is constant. We also present upper bounds on the broadcast time of HBLF in a network with or without holes.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor, Professor Jaroslav Opatrny, who has supported and guided me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way and to pursue my own interests. I would also like to thank Professor Michel Barbeau, Professor Chadi Assi, Professor Thomas Fevens, and Professor Lata Narayanan for being on my examining committee and for all their helpful and insightful comments and suggestions throughout this work.

In my daily work I have been blessed with friendly and cheerful group of fellow students. I thank them for all their support and friendship. Special thanks to my labmates Sirma Altay, Mohsen Eftekhari and Puspal Bhabak, who provided a much needed form of escape from my studies and helped me keep things in perspective.

Most importantly, I would like to thank my family for giving me their constant love and encouragement over the years. Without them I would not be here today. Thank you for your endless love, support and understanding not only throughout this work, but throughout my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless networks emerged into the computing industry in the 1970s and since then have been widely used in practical applications. There are two variations of wireless networks. The first is known as *infrastructure network*, a network with fixed and wired gateways. The bridges of an infrastructured network are known as *base stations*. Mobile units within the network connect to and communicate with the nearest base station within their communication radius. The mobile host may travel out of the range of one base station and into the range of another. In such a case a *handoff* occurs from the old base station to the new, and the mobile unit continues its communication throughout the network. Application of an infrastructured network includes office wireless local area networks (WLANs). [76]

The work considered in this thesis is for the second type of wireless networks, which are *infrastructure-less*. These networks are commonly known as *ad hoc networks*. *Ad hoc networks* do not rely on a preexisting infrastructure and have no fixed routers or base stations. All nodes function as routers that participate in routing to other nodes in the network. Some applications of ad hoc networks are emergency search-and-rescue operations, natural disasters and military conflicts. Wireless ad hoc networks can further be classified by their application into *wireless sensor networks (WSNs)* and *mobile ad hoc networks (MANETs)*. A *wireless sensor network* is a large number of sensors spatially distributed over a geographical region to cooperatively collect data for monitoring physical or environmental conditions. The data collected may be of physical nature such as light intensity, temperature, sound, or proximity to objects. A *mobile ad hoc network* is a collection of autonomous mobile hosts that communicate over a relatively bandwidth constrained wireless links. [76]

Sensors and mobile nodes in wireless ad hoc networks communicate among themselves using radio transceivers. Each node has a *transmission range*, the maximum distance it can transmit data. A single radio transmission of a node can be received by all of its neighbours within that range. Two nodes can communicate directly if they are within each others' transmission ranges. Nodes that are further away from each other may communicate by sending messages through intermediate nodes. [1, 76, 153]

There are several main differences between wireless sensor networks and mobile ad hoc networks. We give a brief overview of these differences below [76].

Equipment: The equipment used by wireless sensor networks are sensor nodes, which are typically small and simple. Hence, these devices do not need to be as powerful as the nodes in mobile ad hoc networks. Mobile ad hoc networks are associated with applications such as voice communication between two peers, which requires powerful equipment for multicasting in order to attain efficient group communication for both data and real time traffic. Therefore, the battery of terminals in mobile ad hoc networks are much larger than that of in wireless sensor networks. Also, it should be noted that mobile ad hoc networks mostly involve heterogeneous nodes (with different form, energy, transmission range and bandwidth factors) and heterogeneous traffic (voice, data and multimedia). Sensor nodes in wireless sensor networks most of the time are homogeneous since they are being deployed in large numbers. [76]

Application/Environment Specific: Wireless sensor networks are mostly used to interact in the environment and hence, their traffic characteristics are expected to be different from those in mobile ad hoc networks. In wireless sensor networks it is likely to exhibit very low data rates over a large timescale, but can have very bursty traffic when something happens in the network. At the application level, mobile ad hoc network users typically communicate and collaborate as teams. Therefore, mobile ad hoc networks are used to support more conventional applications such as web and voice communication, which do not have this diversity in traffic rate. [76]

Energy and Resource Scarceness: Both in wireless sensor networks and mobile ad hoc networks, energy is a scarce resource. Wireless sensor networks have a much higher requirements on the network lifetime since sensor nodes are simple devices compared to the nodes in mobile ad hoc networks. Recharging and replacing batteries of sensor nodes may not be feasible since sensors may be deployed in areas that are unreachable. For both types of networks memory is important for routing protocols. However, it is not largely available for wireless sensor networks as it is for mobile ad hoc networks. Hence, requiring scalable, resource-efficient solutions for wireless sensor networks is necessary. [76]

Mobility: Nodes in mobile ad hoc networks are mobile unlike in wireless sensor networks. Therefore, the network topology of mobile ad hoc networks may change rapidly and unpredictably over time affecting communication between nodes in the network. Wireless ad hoc networks are decentralized and all the network activity, such as discovering the network topology and delivering messages, must be executed by the nodes themselves. [76]

Communication in wireless ad hoc networks is affected by the network connectivity. A *connected* network is determined by the transmission range, the network density, and the physical location of each node. A wireless ad hoc network is *connected* if there is a path between any two nodes either directly or through other intermediate nodes. It is crucial for some applications that the network is not partitioned into disjoint connected components. A connected network facilitates the development of guidelines regarding the design and operation of wireless ad hoc networks, such as communication protocols and methods for data gathering. Sensor nodes and mobile nodes in wireless ad hoc networks may be constrained in processing ability, storage capacity and energy for communication. Over time, the network may become disconnected due to mobility of nodes, battery failures of nodes, or even due to software bugs. Hence, to communicate data within wireless ad hoc networks it is desirable to use as less of the network resources as possible. [1, 76, 153]

Routing and flooding in wireless ad hoc networks are important communication primitives. Flooding is the mechanism by which information needed by all nodes in the network is received at each node. Uncontrolled flooding, without any limitation on rebroadcasting at each node may result in an excess of redundancy, channel contention and collision. This phenomenon is called the *Broadcast Storm Problem* [113]. Hence, it is desirable to construct a flooding scheme with minimum overhead, bandwidth consumption and small number of forwarding nodes. Thus, to address redundancy, the decision whether to rebroadcast the message must be controlled at each node when receiving the message. Since the topology of a mobile ad hoc network changes frequently, communication in such networks is a challenging task.

One way to do routing, multi-hop communication, is to select some wireless nodes to form a *virtual backbone* of the network. Constructing virtual backbones reduces the route searching space. Nodes that are not on the backbone and wish to send a packet to another node in the network simply forward the packet to the nearest backbone node. The backbone nodes then are responsible for delivering the packet to the destination. Virtual backbones allow an increase in the number of nodes that can be inactive while still preserving the ability of the network to forward messages [11, 26, 42, 147, 150]. Hence, they play an important role in power management of wireless sensor networks by preserving energy among nodes and as a result increasing the network lifetime.

Efficient routing in wireless ad hoc networks can also be achieved through cluster-based hierarchical structures. *Clustering* builds a hierarchy among nodes [3]. Sub-structures collapsed in higher levels are called *clusters*. In each cluster at least one node may represent the cluster, and is usually called a *cluster-head*. Each cluster-head is responsible to maintain connectivity of all nodes within its cluster. Nodes in a cluster are either directly connected to the cluster-head or within a few hops of the cluster-head. Thus, different mechanisms can be used for intra-cluster routing (routing within a cluster) and inter-cluster routing (routing between clusters) [69, 100]. Clusters themselves can be grouped into super-clusters to built an m-level hierarchical clustering structure for $m \geq 1$ [86].

In this thesis it is of our interest to design algorithms for data communication in wireless sensor networks and in mobile ad hoc networks. In Sections 1.1 and 1.2 we present the problems considered in WSNs and in MANETs respectively. Section 1.3 outlines the contributions of this thesis.

1.1 WSNs - Network Model and Definitions

One way to do multi-hop communication in wireless sensor networks is via a virtual backbone or a cluster-based hierarchical structure, where designated sensors act as backbone nodes or cluster-heads. One of the two problems in WSNs that we consider in this thesis is to find a group of sensors to act as a backbone of the network or form a clustering hierarchy. Further details regarding this problem are discussed in Section 1.1.1.

In wireless sensor networks, some sensor nodes may be designated as *sink* nodes to which other sensors send their data. That is, sinks act as data collection points in wireless sensor networks. A wireless sensor network may have one or more sink nodes. In general sinks do much more computation and manipulate the collected data (e.g. aggregating similar data or filtering redundant information) and communicate it to a central unit for processing. A network with only one sink is prone to failure. Hence, we consider multiple sinks in wireless sensor networks. Thus, the second problem in WSNs that we consider is to determine an upper bound on the number of sinks that can be uniformly distributed in the network as data collection points to aggregate and remove redundant data before sending it to a central unit for processing. Further details regarding this problem are discussed in Section 1.1.2.

We represent a wireless sensor network by a graph $G = (V, E)$, where V is the vertex set consisting of sensor nodes and E is the edge set of communication links between sensor nodes. Throughout this thesis, we also use the notation $V(G)$ and $E(G)$ to denote the vertex set and the edge set of graph G respectively. We denote an edge between two vertices u and v as (u, v). For any $(u, v) \in E$, we say u and v are *adjacent*. The minimum number of transmissions required to send a message from a sensor node u to a sensor node v is the *distance* from u to v, denoted $d(u, v)$. To address the two problems in WSNs mentioned above formally, we first present some necessary definitions. All definitions are obtained from two books on domination in graphs by Haynes, Hedetniemi and Slater [62, 63].

An undirected graph G on at least $k + 1$ vertices is *k-vertex connected* or *k-connected* if every subgraph of G obtained by removing at most $k - 1$ vertices is connected.

The *open neighbourhood* $N(v)$ of the vertex v consists of the set of vertices adjacent to v, that is, $N(v) = \{w \in V | (v, w) \in E\}$. The *closed neighbourhood* of v is the set $N[v] = N(v) \cup \{v\}$.

The *open-k neighbourhood* of a vertex $v \in V$, denoted $N_k(v)$, is the set $N_k(v) = \{u | u \neq v \text{ and } d(u, v) \leq k\}$. The set $N_k[v] = N_k(v) \cup \{v\}$ is called the *closed k-neighbourhood* of v. Every vertex $w \in N_k[v]$ is said to be *k-adjacent* to v.

A set $S \subseteq V$ is a *dominating set* of G if every vertex $u \in V \setminus S$ is adjacent to a vertex $v \in S$. We say vertices of the dominating set S *dominate* the entire vertex set V, where each vertex $u \in S$ dominates its closed neighbourhood. A *minimum dominating set* of graph G is a dominating set of G such that its cardinality is the smallest among all dominating sets of G. A minimum dominating set is not necessarily unique for a given graph. The minimum cardinality of a dominating set in G is called the *domination number* of G, denoted $\gamma(G)$. A *connected dominating set (CDS)* S is a dominating set of G, whose induced subgraph is connected. There are different variations of the dominating set, some of which we define here. For all these variations we are interested in finding sets of minimum cardinality.

For a fixed positive integer k, a set $S \subseteq V$ is a *distance-k dominating set* of G if every vertex $u \in V \backslash S$ is within distance k of a vertex $v \in S$. The minimum cardinality of a distance-k dominating set in G is the *distance-k domination number* of G, denoted $\gamma_k(G)$. Note that a distance-1 dominating set is equivalent to a dominating set.

For a fixed positive integer r, a set $S \subseteq V$ is a r-dominating set of G if every vertex $u \in V \setminus S$ has at least r adjacent vertices in S. The minimum cardinality of a r-dominating set in G is the *r-domination number* of G, denoted $\gamma_{(\times r)}(G)$. Note that 1-dominating set is equivalent to a dominating set.

A variation of the r-dominating set is the r-tuple dominating set. For a fixed integer r, a set $S \subseteq V$ is a *r-tuple dominating set* of G if for every vertex $v \in V$, $|N[v] \cap S| \geq r$. The minimum cardinality of a r-tuple dominating set in G is the *r-tuple domination number* of G, denoted $\gamma_{\times r}(G)$.

A set $S \subseteq V$ is a *total r-dominating set* of G if for every vertex $v \in V$, $|N(v) \cap S| \geq r$. The minimum cardinality of a total r-dominating set in G is the *total r-domination number* of G, denoted $\gamma^t_{\times r}(G)$. Note that in a r-tuple dominating set, each vertex dominates its closed neighbourhood, while in a total r-dominating set, each vertex dominates its open neighbourhood. Note that a total 1-dominating set is a dominating set. However, the inverse is not necessarily true.

A (k, r)-dominating set is a combination of two previously defined problems, distance-k dominating set and r-dominating set for some positive integers k and r. For fixed positive integers k and r, a set $S \subseteq V$ is a *(k, r)-dominating set* of G if every vertex $v \in V \setminus S$ is within distance k of r vertices in S. The minimum cardinality of a (k, r)-dominating set in G is the *(k, r)-domination number* of G, denoted $\gamma_{(k,r)}(G)$. Note that when k and r are both 1, then $(1, 1)$-dominating set is simply a dominating set.

For fixed positive integers k and r, a set $S \subseteq V$ is a *total (k,r)-dominating set* of G if every vertex $v \in V$ is within distance k of r vertices in S. The minimum cardinality of a total (k,r)-dominating set in G is the *total (k,r)-domination number* of G, denoted $\gamma^t_{(k,r)}(G)$.

A related concept to a dominating set is an *independent set*. A set $S \subseteq V$ is an *independent set* of G if for any pair of vertices $u, v \in S$, $(u,v) \notin E$. A *maximal independent set* of G is an independent set $S \subseteq V$ such that S is not a subset of any other independent set of G. A *maximum independent set* of G is an independent set of G such that its cardinality is largest among all independent sets of G. Note that a maximal independent set of graph G is also a dominating set of G.

A generalization of the independent set is the *distance-k independent set*. For a positive integer k, a subset of vertices $S \subseteq V$ of a graph G is called a *distance-k independent set* if for any pair of vertices $u, v \in S$, $d(u,v) \geq k+1$. A set S is a *maximal distance-k independent set* of G, denoted MIS_k, if S is not a subset of any other distance-k independent set of G. Note that a maximal distance-k independent set is also a distance-k dominating set. However, when considering a distance-k dominating set, it is desirable to obtain a set of minimum cardinality, while when considering a distance-k independent set it is desirable to obtain a set of maximum cardinality. Hence, obtaining a maximal distance-k independent set of a graph G does guarantee a distance-k dominating set of G, but it may not be the best approximation for a distance-k dominating set in terms of its size.

1.1.1 Backbones and Clustering in WSNs

In the literature, connected dominating sets have been proposed to construct backbones in WSNs as well as maintaining cluster-based hierarchical structures, where the cluster-heads are the nodes in the connected dominating set. Efficient routing is achieved via the nodes on the connected dominating set, which are used to propagate the message from a source node to a destination node. However, a connected dominating set is often vulnerable due to frequent node and/or link failures, which are inherent to wireless sensor networks. Thus, in case of node failures, the backbone is disconnected and the network may cease to function properly. Also, in cluster-based hierarchies, instead of having one cluster-head representing a group of nodes, hierarchies could account for node failures by deploying multiple cluster-heads within each cluster. In such a case, nodes within the same cluster can have alternate access points when accessing nodes outside their own cluster, and adjacent clusters could be connected among each other through alternate paths. Thus, to construct a fault tolerant virtual backbone and a clustering hierarchy that functions after the failure of nodes and/or links is an important problem. At the same time, it is generally also considered important to keep the size of the backbone as small as possible to reduce energy consumption in the network. [126]

Instead of considering a connected dominating set for a backbone or a clustering hierarchy in WSNs, we consider a 2-connected (k, r)-dominating set, denoted $D_{(2,k,r)}$. By allowing for a distance-k dominating set, the distance parameter k allows increasing local availability by reducing the distance to the dominators. On the other hand, since every node not in the $D_{(2,k,r)}$ set is dominated by at least r dominators, we improve the robustness and fault-tolerance of the backbone. Finally, a 2-connected backbone is resilient to a single node or link failure. That is, if a node on the backbone fails, the backbone is still connected.

A network where all nodes have the same transmission range, can be modelled as a *unit disk graph*. A unit disk graph is a graph G, where there exists an edge between two nodes if their Euclidean distance is less than or equal to one unit. The complexity of 2-connected (k, r)-dominating set has not yet been studied. We propose a centralized algorithm to construct a 2-connected (k, r)-dominating set in unit disk graphs and in general graphs.

1.1.2 Multiple Sinks in WSNs

In sensor networks, communication is limited in energy and bandwidth and is non-trivial in terms of routing. Each data transmission by a sensor node consumes energy. In a sensor network with only one sink node, all sensors transmit their data to the sink. Sensors that are not direct neighbours of the sink send their data through other neighbouring nodes. Hence, energy is depleted at all of the intermediate nodes on the path to the sink. Sensors that are direct neighbours of the sink deplete their energy by forwarding data to the sink on behalf of other nodes. Thus, they are likely to run out of energy sooner than other nodes. [1]

A network with only one sink is prone to failure. In case of sink failure, the network ceases to function. To address the problem of sink failure, we consider multiple sinks in the network. However, sinks should not be clustered in one area of the network. Therefore, we limit the distance each piece of data travels to get to a sink, which results in significant savings in energy and hence, in an increase of the network lifetime. Sinks may also die due to attacks. By sneakily dismantling a few sinks, the functionality of the network is affected significantly. Since sinks are important and critical objects in the network, they need to be monitored as well. Therefore, this suggests the problem of finding a group of sinks such that every node is within distance k of r sinks, which may allow the network to continue to function even after some node or link failures. This problem is equivalent to finding a total (k, r)-dominating set for a graph $G = (V, E)$, where k and r are fixed positive integers. To save energy and network resource one would like the number of sinks in the network to be as small as possible and hence, deriving upper bounds on $\gamma^t_{(k,r)}(G)$ is an important problem.

In WSNs there is no pre-configured network infrastructure or centralized control and due to generally large number of sensors in such networks or unreachable terrain, arranging sensors manually is unrealistic. Consequently, sensors may be arranged in a stochastic manner. Hence, before the network is established, location of sensors and information of their neighbours are unknown, which introduces uncertainty and randomness into the network structure. Hence, the network can be modelled as a *random graph*, $G(n, p)$. A *random graph* $G(n, p)$ consists of n vertices with each of the potential $\binom{n}{2}$ edges being inserted independently with probability p. Thus, it is of interest to derive upper bounds on $\gamma^t_{(k,r)}(G(n, p))$ in random graphs as well as on $\gamma^t_{(k,r)}(G)$ in graphs with large girth.

1.2 MANETs - Network Model and Definitions

Communication in MANETs can be done through *topology-based* or *position-based* protocols. *Topology-based* protocols use the information about the links that exist in the network to perform packet forwarding. *Position-based* protocols eliminate some of the limitations of the topology-based protocols by using additional information on the position of nodes. Position-based protocols require that each node is aware of its physical position. Nodes may also be aware of the positions of their neighbouring nodes depending on the assumptions of the protocol used. Nodes can determine their positions via GPS or position service. At each node the decision to forward the packet is then based on the position of the forwarding node's neighbours and the information contained in the packet header. [64, 101]

In this research, we consider a position-based algorithm to flood a message throughout a MANET. Before defining our problem, we first distinguish between different kinds of position-based protocols. Position-based flooding algorithms proposed in the literature thus far can be classified into two categories: *beacon-based* and *beacon-less* algorithms. [143]

Beacon-based Protocols

Beacon-based algorithms make use of neighborhood tables obtained by beacon messages. Beacon messages are periodically broadcast by each node to account for topology changes in the network and/or node failures. That is, nodes in beacon-based protocols use beacon messages to find the positions of neighbouring nodes and use this location information if necessary. In such protocols data packets are forwarded via unicast to one or several known neighbours. An important issue in beacon-based algorithms is how to select a subset of neighbours of a forwarding node v, which will continue flooding the message throughout the network. Two strategies are used: *sender-based* and *receiver-based* [101]. In *sender-based* algorithms each forwarding node nominates a subset of its neighbours to be the next hop forwarding nodes. In *receiver-based* algorithms each node that receives a message makes its own decision whether it should forward the message or not based on the local information available to it. [101, 143]

Beacon-less Protocols

Beacon-less algorithms work without any beacon messages [64]. That is, there are no periodic messages sent to account for topology changes. Nodes that receive a data packet decide on their own whether they forward the data using geographical constraints and contention timers without any additional communication with neighbours. This is a preferable solution over beacon-based algorithms for several reasons: (1) periodically sent beacon messages cause communication overhead and are subject to collision, (2) unicasts may fail and due to node movements, nodes may not be reachable even though listed in neighbourhood tables. [143]

Nodes in a typical beacon-less protocol are not aware of the positions of their neighbours and a forwarding node v does not decide the next set of forwarders as may be done in beacon-based algorithms. The next set of forwarders in v's neighbourhood are decided locally by the neighbours of v themselves after receiving the packet. A disadvantage of beacon-less algorithms is when each node receives a message there is a delay before the message is rebroadcast due to contention timers. However, the advantage is that the next hop is determined without any additional communication via beacon messages. [64, 101, 143]

In this work, it is of our interest to efficiently flood a data packet throughout a network, where nodes do not use any beacon message to obtain topological updates of the network. We assume that each node in the network has the same transmission range and two nodes can communicate with each other if they are within each others' transmission range. Hence, the network can be represented as a unit disk graph. We propose a Hexagonal Beacon-Less Flooding algorithm, HBLF, in networks modelled as unit disk graphs, where each node dynamically determines whether to forward the message or not. It is of interest to limit the number of forwarding nodes to preserve the network resources, but at the same time it is desirable to have every node in the network receive the message. We present theoretical analysis, where we show that every node in the network receives the data packet as well as give lower and upper bounds on the number of forwarding nodes and an upper bound on the broadcast time of the algorithm.

1.3 Thesis Contributions and Outline

Chapter 2 presents a survey of several results in the literature concerning algorithmic solutions to dominating sets and its variations, upper bounds on the domination number and its variants, and flooding algorithms in MANETs.

In Chapter 3, we address the problem of finding a group of sensors in WSNs to act as a virtual backbone or a clustering hierarchy. This problem is equivalent to finding a 2-connected (k, r)-dominating set in graphs. We give two centralized algorithms to find a 2-connected (k, r)-dominating set in unit disk graphs and in general graphs. We show that our algorithm in unit disk graphs returns a size of 2-connected (k, r)-dominating set $2D\beta|OPT|$, where D is the diameter of the graph, β is $O(k)$ and OPT is the optimum solution to the 2-connected (k, r)-dominating set. In general graphs our proposed algorithm returns a solution of size $2D \ln \Delta_k |OPT|$, where Δ_k is the largest cardinality among all k-neighbourhoods in the graph.

Chapter 4 considers upper bounds on the minimum number of sinks necessary in a wireless sensor network such that every sensor is within distance k of r sinks. This is equivalent to giving upper bounds on the total (k, r)-domination number. Bounds on the total (k, r)-domination number in graphs have not been studied in the literature. Thus, we present an upper bound on $\gamma^t_{(k,r)}(G)$ in general graphs. We also give an upper bound on $\gamma^t_{(2,r)}(G(n, p))$ with $p \geq c\sqrt{\frac{\log n}{n}}$ with $c > 1$. This result is generalized to obtain an upper bound on $\gamma^t_{(k,r)}(G(n, p))$ with $p \geq k \sqrt[k]{\frac{\log n}{n^{k-1}}}$ for $k \geq 3$.

In Chapter 5, we propose a beacon-less flooding algorithm, HBLF, for MANETs. We also present theoretical analysis of the algorithm. We give a sufficient condition for HBLF to achieve full delivery even in the presence of holes in the network. Lower and upper bounds are given on the number of forwarding nodes returned by HBLF in a network with or without holes. When there are no holes in the network, we show that the ratio of the shortest path returned by HBLF to the shortest path in the network is constant. We also present upper bounds on the broadcast time of HBLF in a network with or without holes. Chapter 5 concludes with briefly discussing how HBLF may be used for routing purposes if the approximate area of a destination node is known.

Finally, in Chapter 6, we conclude this thesis and discuss possible future work.

Chapter 2

Related Work

In this chapter we present related work in the literature regarding algorithmic solutions to domination problems and its variants, upper bounds on the domination number and its variants (Section 2.1); and flooding algorithms in MANETs (Section 2.2).

2.1 Domination in Graphs

The study of dominating sets dates back to 1862 when de Jaenisch [73] studied the problem of determining the minimum number of queens, which are necessary to cover (or dominate) an $n \times n$ chess board. The mathematical study of dominating sets began around 1960. The concept of the domination number of a graph was defined by Berge in the book *Theory of Graphs and its Applications*, published in 1958, where he called the domination number as the *coefficient of external stability* [8]. The terms *dominating set* and *domination number* were used for the first time by Ore in the book *Theory of Graphs* published in 1962 [115]. Cockayne and Hedetniemi in 1977 published a survey of known results about dominating sets in graphs [34]. The decision problem for dominating sets can be stated as follows.

DOMINATING SET
INSTANCE: A graph G and a positive integer k
QUESTION: Does G have a dominating set of size less than or equal to k?

Garey and Johnson showed that DOMINATING SET is NP-complete for arbitrary graphs [54]. However, it is solvable in polynomial time in trees [33]. Garey and Johnson also show that the connected dominating set is NP-complete [54]. Clark et al. have shown that the dominating set and the connected dominating set are NP-complete in unit disk graphs [32]. The research area of dominating sets has vastly grown during the last few decades and several different variations of dominating sets are being considered today. Giving upper bounds on the domination number and developing heuristics that can give a bound on the size of dominating sets are important problems. We will discuss some of this work regarding the algorithmic problems of variations of dominating sets as well as upper bounds of domination numbers in graphs.

2.1.1 Algorithms for Variations of Dominating Sets

There are many papers in the literature for finding (connected/total) dominating sets in wireless sensor networks modelled as unit disk graphs as well as general graphs [30, 39, 41, 48, 55, 80, 81, 96, 97, 107, 114, 116, 123, 136, 139].

The idea of dominating each vertex in a graph multiple times originated with Fink and Jacobson [45]. It was shown by Jacobson and Peters that the problem of finding a minimum r-dominating set is NP-hard [72]. A variation of r-dominating set is the r-tuple dominating set, introduced by Harary and Haynes in [57], and the total (open) r-dominating set defined by Kulli [91].

An incremental algorithm constructing an r-dominating set in unit disk graphs is given in [36]. The algorithm iteratively constructs a monotone family of dominating sets $D_1 \subseteq D_2 \subseteq \cdots \subseteq D_r$ such that each D_i is an i-dominating set. For unit disk graphs, the size of each of the resulting i-dominating sets is at most six times the optimal solution.

Wang et al. gave centralized and distributed approximation algorithms to construct a total r-dominating set in unit disk graphs [141]. The centralized algorithm is an extension of the algorithm given by Marathe et al. in [107] for finding a total dominating set. Both the centralized and distributed algorithms give a 10-approximation for unit disk graphs.

Dai and Wu propose three localized algorithms to construct a r-connected r-dominating set [40]. For two positive integers m and r, an m-connected r-dominating set is a subset $S \subseteq V$ such that every vertex $u \in V \setminus S$ is adjacent to at least r

vertices in S and the subgraph induced S is m-connected. The two of the algorithms, r-gossip and colour based (r, r)-CDS, introduced by Dai and Wu are probabilistic. In the r-gossip algorithm, each vertex decides to be in the dominating set with a probability based on the network size, deploying area size, transmission range, and r. In the colour-based (r, r)-CDS algorithm, each vertex randomly selects one of the r colours such that the network is divided into r disjoint subsets based on the colours of vertices. For each subset of vertices, a connected dominating set is constructed and (r, r)-CDS is the union of the r connected dominating sets. The third algorithm, r-coverage algorithm, is deterministic, which only works in very dense networks and no upper bound on the size of the resulting dominating set is analyzed. Li et al. further extend this work to construct m-connected r-dominating sets for general m and r [98].

Wang et al. propose a centralized algorithm to construct a 2-connected dominating set as a virtual backbone in wireless networks [140]. The algorithm first constructs a connected dominating set and then computes all blocks and adds intermediate nodes to make all the backbone nodes in the same block. Thai et al. study the m-connected r-dominating set problem and propose two approximation algorithms for m-connected r-dominating sets and r-connected r-dominating sets [130].

Shang et al. gave a centralized algorithm for finding a connected r-dominating set in unit disk graphs [124]. They proved an approximation ratio of $\left(5 + \frac{5}{r}\right)$ for $r \leq 5$ and a 7-approximation for $r > 5$. They also propose an algorithm to construct a 2-connected r-dominating set with an approximation ratio of $\left(5 + \frac{25}{r}\right)$ for $2 \leq r \leq 5$ and an 11-approximation for $r > 5$. They present a third algorithm for m-connected r-dominating sets. The algorithm first constructs a r-connected r-dominating set and then for $3 \leq r \leq m$ sequentially constructs a maximal independent set to obtain an m-connected set.

Wu et al. give a centralized algorithm that constructs an m-connected r-dominating set [149]. In the first phase, a r-dominating set is constructed. In the second phase, this set is augmented to obtain an m-connected r-dominating set by adding enough number of connectors. Wu and Li further extend this algorithm to obtain an m-connected r-dominating set [148]. The construction of their algorithm is similar to that of Wu et al [149]. Li et al. propose centralized as well as distributed methods, deterministic and probabilistic, to construct an m-connected r-dominating set for general m and r [98].

In wireless sensor networks distance-k dominating sets have been used to implement cluster-based hierarchical structures to achieve efficient routing. The network is divided into several clusters, where each cluster contains a cluster-head responsible for maintaining the routing information. The distance-k dominating set, sometimes referred to as k-dominating set or a k-hop dominating set, was first introduced by Henning [65]. Distance-k dominating set and connected distance-k dominating set are proved in [4] and [111] to be NP-complete in unit disk graphs. Gao et al. give an approximation algorithm that computes a connected distance-k dominating set with size at most $O(k^3)$ [53].

Li and Zhang give two algorithms for minimum 2-connected distance-k dominating sets [95]. The first algorithm is based on a greedy heuristic in general graphs and uses the concept of *ear* decomposition of 2-connected graphs [95]. Given a graph G, let S be a subgraph of G. An *ear* of S in G is a non-trivial path in G whose ends lie in S, but the internal vertices do not. The second algorithm is only applicable to unit disk graphs. The algorithm first constructs a distance-k maximal independent set, MIS_k, and by iteratively adding vertices on the shortest path between the vertices in the MIS_k obtains a connected distance-k dominating set denoted D. In the next step, to make D 2-connected, the authors use the notion of *blocks* of a graph. A *block* of a graph G is a maximal connected subgraph B of G such that $B \setminus u$ is connected for any $u \in V(B)$. A *cut-vertex* of a connected graph G is a vertex u such that the graph $G \setminus u$ is disconnected. A *leaf-block* of G is a block of G which contains only one cut-vertex of G. Thus, in the last step of the algorithm to make D a 2-connected set, the block structure of D is computed. For every block of D the algorithm iteratively finds a path P of G, which connects a leaf block of D to another part of D. Vertices of P are added to D to obtain a 2-connected set. The approximation ratio obtained for this algorithm is $(2k+2\beta+1)(k+1)|OPT|-2(k+\beta)(k+1)-k$, where β is $O(k)$ and $|OPT|$ is the optimal solution to the 2-connected distance-k dominating set in unit disk graphs. Chan et al. extend this result to a distributed scenario in [23]. Other work in the literature for distance-k dominating sets applicable for wireless networks can be found in [28, 29, 52, 112].

A combination of distance and multiple domination gives rise to the (k, r)-dominating set problem. (k, r)-domination was first introduced by Joshi et al. as *r-neighbour k-domination* and proved it to be NP-complete on interval graphs [74, 75]. (k, r)-dominating sets also have been used for clustering techniques in wireless sensor networks. Spohn et al. construct a (k, r)-dominating set to address redundancy for bounded distance clusters in wireless networks [126]. They present centralized and distributed algorithms for arbitrary network topologies. The centralized algorithm is a greedy based approximation algorithm. The algorithm iteratively chooses a vertex u to be part of the (k, r)-dominating set that has the largest number of dominators needed to dominate $N_k(u)$. They give a $r \ln \Delta$-approximation ratio, where Δ is the largest cardinality among all distance-k neighbourhoods in the network.

Li et al. proposed two centralized approximation algorithms for minimum connected (k, r)-dominating set in unit disk graphs [93]. The first algorithm is in unit disk graphs, which yields an approximation ratio of $(2k+1)^3$ if $r \leq (2k+1)^3$ and $(2k+1)((2k+1)^2+1)$ for $r > (2k+1)^2$. Zhang et al. further improve this result and give an approximation algorithm with a performance ratio of $(k+1)\beta - k$ if $r \leq \beta$ and $(k+1)(\beta+1) - k$ if $r > \beta$, where β is at most $O(k)$ instead of $O(k^2)$ in the results of Li et al. [156]. The second algorithm by Li et al. is an extension of the centralized algorithm presented in [126]. In the first step, the algorithm constructs a (k, r)-dominating set S as in [126] and in the second step S is made connected by adding extra vertices on the shortest path between the vertices in S. Li et al. showed that in the first step of the algorithm, the approximation ratio in [126] can be improved to $\ln \Delta$ instead of $r \ln \Delta$, where Δ is the cardinality among all distance-k neighbourhoods in the network. Li et al. further show that their algorithm has an approximation ratio of $(2k+1)\ln \Delta$ for any undirected graph.

2.1.2 Upper Bounds on $\gamma^t_{(k,r)}$

In the literature, there are extensive number of works regarding bounds on the domination number and its variants in graphs. We present some the fundamental results here. We first present the upper bounds on the domination number and its variants in general graphs, then we present the known results in random graphs.

Given a graph $G = (V, E)$, let $n = |V|$, δ denote the minimum degree of G and let Δ denote the maximum degree of G.

Several upper bounds on the domination number in general graphs can be found in [9, 47, 117]. A fundamental result on the upper bound of the domination number was proved by many authors.

Theorem 2.1.1. *([5, 2, 106, 117]) For any graph G of minimum degree δ, $\gamma(G) \leq \frac{\ln(\delta+1)+1}{\delta+1}n$.*

This is an excellent upper bound when δ is large enough. For small values of δ better results can be found in [108, 115, 122].

Distance domination has been studied extensively by several authors [66, 94, 131, 56, 120, 132]. Meir and Moon gave an upper bound on $\gamma_k(G)$.

Theorem 2.1.2. *([109]) For any connected graph G of order n with $n \geq k+1$, $\gamma_k(G) \leq \lfloor \frac{n}{k+1} \rfloor$.*

Sridharan et al. considered the special case of $k = 2$, independently, and obtained $\gamma_2(G) \leq \left\lfloor \frac{n}{3} \right\rfloor$ for $n \geq 3$ [127]. Tian and Xu show that for a connected graph G $\gamma_k(G) \leq n\frac{\ln[m(\delta+1)+2-t]}{m(\delta+1)+2-t}$, where $m = \lceil \frac{k}{3} \rceil$ and $t = 3\lceil \frac{k}{3} \rceil - k$ [132]. Liu et al. in [104], give an upper bound on the 2-connected distance-k domination number, denoted $\gamma_k^2(G)$.

Theorem 2.1.3. *([104]) Let G be a 2-connected graph with order n and minimum degree δ. Then $\gamma_k^2(G) \leq (1+o_\delta(1))n\frac{\ln[m(\delta+1)+1-t]}{m(\delta+1)+1-t}$, where $m = \lceil \frac{k}{3} \rceil$, $t = 3\lceil \frac{k}{3} \rceil - k$ and $o_\delta(1)$ denotes a function in δ that tends to 0 as δ tends to ∞.*

The first upper bound for r-domination number is due to Cockayne et al. in [35], where they prove that with $\delta \geq r$, $\gamma_{(r)}(G) \leq \frac{r}{r+1}n$. This bound has been further improved in [18, 19]. The results in [19] improve the bound in [35] for larger values of δ, that is for $\delta > e^{r^2}$. Rautenbach and Volkmann extended the result in [19] for smaller values of δ in [121].

Theorem 2.1.4. *([121]) In a graph G of order n, where $\delta \geq 2r\ln(\delta+1) - 1$, then $\gamma_{(r)}(G) \leq \frac{n}{\delta+1}\left(r\ln(\delta+1) + \sum_{i=1}^{r-1}\frac{1}{i!(\delta+1)^{r-1-i}}\right)$ and $\gamma_{\times r}(G) \leq \frac{n}{\delta+1}\left(r\ln(\delta+1) + \sum_{i=1}^{r-1}\frac{r-i}{i!(\delta+1)^{r-1-i}}\right)$.*

In [51] Gagarin and Zverovich presented a generalized upper bound for the r-tuple domination number. Chang [24] further improved their result for any positive integer r and for any graph of n vertices with minimum degree δ, where $\gamma_{\times r}(G) \leq \dfrac{\ln(\delta - r + 2) + \ln \tilde{d}_{r-1} + 1}{\delta - r + 2} n$, where $\tilde{d}_m = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \binom{d_i + 1}{m}$ with d_i being the degree of the ith vertex of G.

Caro and Yuster in [19] show that for $\delta > e^{r^2}$, $\gamma^t_{\times r}(G) \leq \frac{\ln \delta}{\delta} n(1 + o_\delta(1))$. For large values of δ, this result implies an upper bound on the r-tuple and total r-domination numbers [19].

Theorem 2.1.5. *([19]) In a graph G of order n and minimum degree δ, if $\delta > e^{r^2}$ and $r \in \mathbb{N}$, then $\gamma_{\times r}(G) \leq \gamma^t_{\times r}(G) < \frac{\ln \delta + \sqrt{\ln \delta} + 2}{\delta} n$.*

Zhao et al. [157] study the total r-domination number in graphs, where they give an upper bound for $\gamma^t_{\times r}$.

Theorem 2.1.6. *([157]) In a graph G of order n and minimum degree $\delta \geq r$, where $r \in \mathbb{N}$, if $\frac{\delta}{\ln \delta} \geq 2r$, then $\gamma^t_{\times r}(G) \leq \frac{n}{\delta}\left(r \ln \delta + \sum_{i=0}^{r-1} \frac{r-i}{i!\delta^{r-1-i}}\right)$.*

There are some research in the literature that study upper bounds for the combination of distance and multiple domination. In [6] Bean et al. posed the following conjecture.

Conjecture 2.1.1. *([6]) Let G be a graph of order n and let δ_k denote the smallest cardinality among all k-neighbourhoods of G, where $\delta_k \geq k + r - 1$. Then for positive integers k and r $\gamma_{(k,r)}(G) \leq \frac{r}{r+k} n$.*

Fischermann and Volkmann confirmed that the conjecture is valid for all integers k and r, where r is a multiple of k [46]. In [87] Korneffel et al. show that $\gamma_{2,2}(G) \leq \frac{n(G)+1}{2}$.

During the last decade, bounds on the domination number and its variants have started to be studied in random graphs. Recall that a random graph $G(n, p)$ consists of n vertices with each of the potential $\binom{n}{2}$ edges being inserted independently with probability p. We say p is *non-fixed* if p is a function of n. Otherwise, we say p is *fixed.* We say that an event holds *asymptotically almost surely (a.a.s)* if the probability that it holds tends to 1 as n tends to infinity.

Dreyer [43] in his dissertation studied the question of domination in random graphs. Wieland and Godbole proved that $\gamma(G(n,p))$ has a two point concentration [142].

Theorem 2.1.7. *([142]) For $p \in (0,1)$ fixed, a.a.s $\gamma(G(n,p))$ equals $\lfloor \mathbb{L}n - \mathbb{L}\left((\mathbb{L}n)(\log n)\right) \rfloor + 1$ or $\lfloor \mathbb{L}n - \mathbb{L}\left((\mathbb{L}n)(\log n)\right) \rfloor + 2$, where $\mathbb{L}n = \log_{1/(1-p)} n$.*

Wang and Xiang [137] extend this result for 2-tuple domination number of $G(n,p)$.

Theorem 2.1.8. *([137]) For $p \in (0,1)$ fixed, a.a.s $\gamma_{\times 2}(G(n,p))$ equals $\left\lfloor \mathbb{L}n - \mathbb{L}(\log n) + \mathbb{L}\left(\frac{p}{1-p}\right) \right\rfloor + 1$ or $\left\lfloor \mathbb{L}n - \mathbb{L}(\log n) + \mathbb{L}\left(\frac{p}{1-p}\right) \right\rfloor + 2$, where $\mathbb{L}n = \log_{1/(1-p)} n$.*

Bonato and Wang [13] study the total domination number and the *independent domination number* in random graphs. For a graph $G = (V,E)$, a set $S \subseteq V$ is an *independent dominating set* of G if S is both an independent set and a dominating set of G. The *independent domination number* of G, denoted $\gamma_i(G)$, is the minimum order of an independent dominating set of G.

Theorem 2.1.9. *([13]) For $p \in (0,1)$ fixed, a.a.s $\gamma_t(G(n,p))$ equals $\lfloor \mathbb{L}n - \mathbb{L}((\mathbb{L}n)(\log n)) \rfloor + 1$ or $\lfloor \mathbb{L}n - \mathbb{L}((\mathbb{L}n)(\log n)) \rfloor + 2$, where $\mathbb{L}n = \log_{1/(1-p)} n$.*

Theorem 2.1.10. *([13]) For $p \in (0,1)$ fixed, a.a.s $\lfloor \mathbb{L}n - \mathbb{L}((\mathbb{L}n)(\log n)) \rfloor + 1 \leq \gamma_i(G(n,p)) \leq \lfloor \mathbb{L}n \rfloor$, where $\mathbb{L}n = \log_{1/(1-p)} n$.*

Wang further studied the independent domination number of random graphs [138].

Theorem 2.1.11. *([138]) Let $p \in (0,1)$ and $\epsilon \in \left(0, \frac{1}{2}\right)$ be two real numbers. Let $k = k(p,\epsilon) \geq 1$ be the smallest integer satisfying $(1-p)^k < \frac{1}{2} - \epsilon$. A.a.s. $\gamma(G(n,p)) \leq \gamma_i(G(n,p)) \leq \lfloor \mathbb{L}n - \mathbb{L}((\mathbb{L}n)(\log n)) \rfloor + k + 1$, where $\mathbb{L}n = \log_{1/(1-p)} n$.*

If $p > \frac{1}{2}$, then for $\epsilon \in \left(0, p - \frac{1}{2}\right) \subset \left(0, \frac{1}{2}\right)$, by Theorems 2.1.7 and 2.1.11, the following concentration result follows.

Corollary 2.1.1. *([138]) For $p \in \left(\frac{1}{2}, 1\right)$ fixed, a.a.s. $\gamma(G(n,p)) \leq \gamma_i(G(n,p)) \leq \lfloor \mathbb{L}n - \mathbb{L}((\mathbb{L}n)(\log n)) \rfloor + 2$, where $\mathbb{L}n = \log_{1/(1-p)} n$.*

To the best of our knowledge, there are no works in the literature that study the upper bounds on the total (k,r)-domination number in general graphs or in random graphs.

2.2 Flooding in MANETs

In this section we present some of the work in the literature regarding efficient flooding problems in MANETs. One simple solution to sending a message throughout a network and ensuring full coverage of the network is blind flooding, where each node forwards the message when it receives the message for the first time [67]. To alleviate inefficiencies of blind flooding other methods have been suggested in the literature.

We present some of these algorithms here. Section 2.2.1 presents beacon-based flooding algorithms in the literature, which assume that each node keeps location information of its 1-hop or 2-hop neighbours. Section 2.2.2 presents beacon-less flooding algorithms in the literature and Section 2.2.3 presents beacon-less routing algorithms in the literature.

2.2.1 Beacon-based Flooding Algorithms

In beacon-based algorithms each node keeps 1-hop or 2-hop information by exchanging HELLO messages. A major issue in such algorithms presented in the literature is the selection of subset of neighbours for forwarding the message.

Lim and Kim introduce two flooding algorithms in wireless ad hoc networks to reduce redundant transmissions [99]. They first introduce the *optimal flooding tree* problem in wireless ad-hoc networks. A *flooding tree* is a tree that covers all nodes in a graph. An *optimal flooding tree* is a flooding tree with minimum cost, where the cost of a flooding tree in a wireless ad hoc network is defined as the number of broadcasts to deliver a packet to all nodes. They show that the optimal flooding tree is similar to a minimum connected dominating set and prove its NP-completeness. Since finding an optimal flooding tree is difficult, Lim and Kim give two heuristics, *self-pruning* and *dominant pruning*, that obtain a flooding tree to flood a given network. Both methods reduce redundant broadcasts by using the neighbourhood information exchanged between mobile nodes. We give brief overview of both algorithms.

Self-pruning is a receiver-based algorithm that uses direct neighbourhood information where each node exchanges HELLO messages to obtain a list of its adjacent neighbours. The algorithm operates as follows. A node v that wishes to send a packet, attaches the list of nodes in $N(v)$ in the header of the packet and broadcasts it. A node u that receives the packet from v checks if $N(u) - N(v) - \{v\}$ is empty. If so, then u knows that all of its neighbours received the packet from v and thus, it stays silent. Otherwise, it forwards the packet, after attaching to its header $N(u)$.

Self-pruning uses only 1-hop neighbourhood information, while dominant-pruning uses 2-hop neighbourhood information, which can be obtained by exchanging the list of adjacent nodes with neighbours. Another difference is that dominant-pruning is a sender-based algorithm. That is, a forwarding node v decides the next set of forwarders from $N(v)$. The forwarding set chosen by a forwarding node v is as follows.

Let v be a forwarding node that has received the packet from a node w. Node v must decide on a forwarding list so that all nodes within 2-hops of v receive the packet, i.e. all nodes in $N_2(v) - N(w) - N(v)$ must receive the packet. In the algorithm proposed, Lim and Kim repeatedly select a vertex $u \in N(v)$, where the number of neighbouring nodes of u not covered yet is maximum. Their simulation results show that dominant pruning performs better than self-pruning due to extra neighbourhood knowledge.

A noteworthy result in the literature is the flooding algorithm proposed by Liu et al. [101]. Their algorithm requires each node to keep 1-hop neighbour information. They prove that their flooding scheme achieves full delivery as well as local optimality in terms of number of forwarding nodes. The network is assumed to be connected and modelled as a unit disk graph, where each node is assigned a distinct ID. Neighbourhood information is obtained via HELLO beacon messages periodically broadcast by each node.

The idea of the algorithm is as follows. Each time a node forwards a packet, it attaches to the header of the packet the list of the next forwarding nodes. A node upon receiving a packet, discards the packet if it has received it before. Otherwise, if it is in the list of forwarders it will compute the next set of forwarders and forward the packet. The set of next forwarding nodes of a node v is chosen from $N(v)$ so that $N_2(v)$ is completely covered. Liu et al. present an algorithm with time complexity of $O(n \log n)$ to find such a set, where n is the number of neighbours of a forwarding node v [101]. The idea of the algorithm is as follows. For each forwarding node v, initially each node in $N(v)$ is arbitrarily paired together to merge their coverage boundaries. Then the merged pair's boundary is further merged with another pair's boundary. This merge operation is repeated until eventually there is only one big merged boundary, which covers all nodes in $N_2(v)$. Note that, during the merging process boundaries of nodes that do not contribute additional coverage of nodes in $N_2(v)$, are not considered in the forwarding set.

An improvement on the algorithm by Liu et al. was proposed by Khabbazian et al. in networks modelled as unit disk graphs [84]. Khabbazian et al. propose a sender-based algorithm and a receiver-based algorithm, where they assume knowledge of 1-hop neighbourhood via HELLO messages periodically broadcast by each node. Both algorithms guarantee full delivery of the message. Simulation results show that both algorithms perform better than the algorithm proposed by Liu et al. in [101]. The sender-based algorithm even computes the next set of forwarders in $O(n)$ time, where n is the number of neighbours of a forwarding node v. This lowering of the time complexity to $O(n)$ compared to the $O(n \log n)$ proposed by Liu et al. is achieved at the cost of an increased end-to-end delay.

Yang et al. introduce a hybrid 1-hop neighbour information based flooding algorithm [151]. To integrate together the advantages of sender-based and receiver-based algorithms, their proposed algorithm consists of two phases: the sender-phase and the receiver-phase.

The sender-phase scheme allows a node to select a subset of its 1-hop neighbours to forward the flooding message. Given two adjacent nodes v and s, the *extended broadcasting area* of v with respect to s, denoted $EBA_s(v)$, is defined as part of v's coverage area not covered by s. Clearly, the size of $EBA_s(v)$ is proportional to the distance between v and s. If s is a forwarding node and selects v as part of the next set of forwarding nodes, then each node in $EBA_s(v)$ receives a new message, while the overlapping area between v and s receive a redundant message. Thus, to decrease the number of repeated receptions Yang et al. propose that each forwarding node s compute a convex-hull to find the smallest convex polygon containing all nodes in $N(s)$ and select this list of convex-hull nodes as the next set of forwarding nodes. Yang et al. use Chan's algorithm in [22] to compute a convex-hull in $O(n \log h)$ time, where n is the number of nodes in the network and h is the number of forwarding nodes (which is significantly lower than n in most cases).

The receiver-based scheme operates as follows. A node v that receives a packet from a node s, checks if it is in the set of forwarding nodes in the header of the packet. If so, then it forwards the packet. Otherwise, it checks if $EBA_s(v)$ is not completely covered by the forwarding nodes in the header of the packet and v has a 2-hop neighbour only covered by v, then v will forward the packet. Yang et al. show that the complexity of this procedure is at most $O(n)$. Through extensive simulations they show that their algorithm performs better than that of Liu et al. in [101] and the two algorithms of Khabbazian et al. in [84].

Another algorithm to note is proposed by Liu et al. in [103]. Their algorithm assumes knowledge of 1-hop neighbourhood and is called *vertex forwarding*. It operates as if there existed a hexagonal grid in the field of the network to guide the flooding procedure. The vertices in the hexagonal grid are the centres of each hexagon and the radius of each hexagon is equivalent to the transmission range of each node. A forwarding node v chooses the next set of forwarding nodes based on the location of its neighbours with respect to the vertices of the hexagonal grid. Nodes located nearest to the vertices of the hexagonal grid are chosen to be in the forwarding set to continue forwarding the message throughout the network.

There are several other algorithms in the literature that assume knowledge of 1-hop or 2-hop neighbourhood that are proposed in [17, 25, 38, 79, 85, 92, 105, 118, 125, 128, 144, 145, 146, 152, 154]. Hereafter, we present some of the beacon-less algorithms in the literature.

2.2.2 Beacon-less Flooding Algorithms

Tseng et al. propose probabilistic based schemes to reduce number of rebroadcasts [113, 133]. They propose four schemes: probabilistic, counter-based, distance-based, and location-based schemes. These four schemes differ in how a node estimates redundancy and how it accumulates knowledge to assist in making its decision. All schemes operate in a fully distributed manner. We present an overview of all the schemes in [133].

The *probabilistic scheme* operates as follows. A node that receives a message for the first time will rebroadcast it with probability P. Clearly, when $P = 1$, this scheme is equivalent to pure flooding. To differentiate the timing of rebroadcasting between nodes, a small random delay is added.

The *counter-based scheme* takes into account that a given node may repeatedly hear the same message multiple times before it starts transmitting the message. The idea of the counter-based scheme is that when a node v hears the same message multiple times, the expected additional area covered by v is reduced. Thus, a counter c is used to keep track of the number of times a node v receives the same message. A counter threshold C is chosen. Whenever $c \geq C$, the rebroadcast at v is inhibited.

The *distance-based scheme* makes use of the Euclidean distance between nodes to decide whether to drop a rebroadcast or not. Suppose a node v has heard the message from a node s. If the Euclidean distance between v and s is very small, then the additional area covered by v in the case of rebroadcasting the message is very little. Thus, a distance threshold D is chosen. If the distance between v and s is less than D, then the rebroadcast transmission of v is cancelled. Otherwise, v transmits the message.

Both the counter-based and distance-based schemes have no need of GPS since estimation of distances can be extracted from signal strength. The *location-based scheme*, however, assumes each node is aware of its exact geographical location. When a node rebroadcasts a packet it adds its own location in the header of the packet. If the receiving node based on location information covers an additional area greater than a given threshold, then it rebroadcasts the message.

Simulation results show that a simple counter-based scheme can eliminate many redundant rebroadcasts in dense neighbourhoods. The distance-based scheme has higher reachability, but among all location-based scheme performs best in sparse as well as dense neighbourhoods of a node.

These probabilistic schemes were further investigated in [67]. The results showed that in these schemes, a non-redundant transmission might be dropped out without being forwarded further. Thus, some nodes in the network may not receive the message. A critical question in these schemes is how to set the right threshold value in various network situations. To alleviate these concerns to some degree various other probabilistic schemes have been introduced in the literature that we briefly discuss below.

Cartigny et al. proposed an adaptive probabilistic scheme [20]. Each node determines to rebroadcast a packet with probability p and a fixed value k, where k is the efficiency parameter to achieve the reachability of the broadcast. The value of p is based on the local node density. However, a critical question in this work is how to optimally select k, since k is independent of the network topology.

Zhang et al. describe a dynamic probabilistic broadcast scheme, which is a combination of the probabilistic and counter based approaches [155]. The probability value of p for each node v is dynamically adjusted according to the number of times v has received the packet. Thus, in case of node movements the value of p changes. To control the effect of the packet counter at each node as a density estimate, two constants d and d_1 are used to increment and decrement the rebroadcast probability. However, a question in this work is how to determine optimal values of d and d_1.

Pleisch et al. propose an algorithm in [119] to address the problem of nodes failing to receive the message due to dropped packets in the probabilistic scheme. The algorithm they propose starts with the probabilistic scheme but compensates for dropped data packets by periodically broadcasting *compensation packets*. Every compensation packet encodes a set of packets that have been dropped by the sender. Thus, each node that does not rebroadcast a packet, adds the packet to the current compensation packet. When the number of packets in a compensation packet reaches a certain threshold, the compensation packet is broadcast. Simulation results show that their algorithm improves the node coverage by 20% than the pure probabilistic scheme.

Tseng et al. propose *adaptive counter-based scheme* and an *adaptive location-based scheme* in [134] to address the problem of constant threshold in the counter-based and location-based schemes. They extend the fixed threshold value C to a function $C(n)$, where n is the number of neighbours of a potentially forwarding node. However, they do not discuss how to determine the value of n.

Mohammed et al. in [110] propose an efficient counter-based scheme (ECS) that allows nodes to make localized rebroadcast decisions on whether or not to rebroadcast a message based on both a counter threshold and a forwarding probability value. A node v that receives a message for the first time initiates a counter c, which will record how many times v will receive the same message. After waiting for a random assessment delay, if c is greater than a defined threshold value, then v stays silent. Otherwise, v rebroadcasts the message with probability p. The simulation results show that an optimal value of p is 0.65. The simulations also show that ECS performs better in sparse and dense networks than the counter-based scheme, however, it still does not achieve full delivery of the message.

In contrast to a counter-based scheme, a *colour-based scheme* is introduced in [83]. The colour-based scheme uses η colours denoted $C_1, C_2, \cdots, C_\eta$. Each node that broadcasts the message select a colour which it writes to a *colour field* present in the broadcast message. A node v that receives a message will start a random timer, upon the expiration of which it will rebroadcast the message unless it has heard all η colours. Simulation results show that the color-based scheme and the counter-based scheme in general return similar results in terms of reachability. The color-based scheme outperforms the counter-based scheme for threshold value $2 \leq \eta \leq 3$.

Liu proposed distributed intelligent broadcasting protocol (DIBP) in [102]. Their algorithm is an extension of the counter-based scheme, where each node maintains an additional timer, called an *aging timer*. The purpose of the *aging timer* is to control the lifetime of the built broadcast topology. The initial aging timer is set by the originating node, denoted s, and is attached to the broadcast message. A forwarding node v updates the aging timer by reducing its value by the round-trip propagation time between node v and the node from which v has received the message. This new timer value is attached to the message and v forwards the message. Once, the aging timer times out, the built broadcast topology is terminated. However, a question not discussed in the paper, is how to initially determine the value of the aging timer.

2.2.3 Beacon-less Routing Algorithms

In this section we present an overview of some of the beacon-less routing algorithms in the literature. Noteworthy beacon-less routing algorithms in the literature are the Beacon-Less Routing Protocol (BLR) [64] and Blind Geographic Routing (BGR) [143]. We present an overview of them here.

The Beacon-Less routing Protocol (BLR) in [64] makes the following assumptions. Each node has a maximum transmission range r and hence, the wireless network is modelled as a unit disk graph. Each node is aware of their own position by means of GPS and the source node knows the ID and position of the destination node.

The BLR Protocol operates in the following manner. The source node before broadcasting the packet stores the position of the destination node, its own current position, and the forwarding area defined by geometric constraints in the header of the packet. As each node decides to forward the packet, it replaces the previous node's current position with its own before broadcasting it. A node v that receives a packet from a node u, based on the information in the header of the packet can derive if it is in u's forwarding area. If v is outside of u's forwarding area, then it drops the packet. Otherwise, v computes a Dynamic Forwarding Delay (DFD) based on its position relative to the position of u and the destination node. Thus, if v is located at the position closest to the destination, it will compute the shortest DFD and as a result will transmit the packet first. All other nodes in u's forwarding area will drop the packet upon hearing this transmission. Note that the forwarding area can be of any shape provided that all nodes in the forwarding area are within each others' transmission range. Thus, BLR takes care that only one node in the forwarding area will forward the packet. Node u also hears the transmission from v and concludes that the packet was received successfully and thus, acknowledgements can be avoided. The algorithm continues until the destination receives the packet. The destination node is the only node that sends an acknowledgement since it does not continue to forward the packet.

Witt and Turau introduce a beacon-less routing algorithm called Blind Geographic Routing (BGR) [143]. BGR is designed to support different delivery semantics. In the literature geographic routing algorithms assume that the location of the destination is known to the sender. This is the case if data packets are only routed to pre-defined locations. However, in some ad hoc networks it may be necessary to send packets to arbitrary locations. Thus, it is desired that the network protocol supports destination locations without the exact location information of the nodes in the vicinity of the network. A node that receives the message near the destination location has to decide if it is a suitable destination for that message.

Much like the Beacon-Less Routing Protocol described previously the forwarding node decides on a forwarding area and broadcasts the message. The width of the forwarding area is chosen such that nodes within the forwarding area can mutually communicate with each other. BGR does not assume a constant transmission range for all nodes. However, it uses a parameter r, the estimation of the transmission range, which is needed for constructing the forwarding areas. An accurate estimation of r results in a better performance of the algorithm.

The algorithm is as follows. The source node S stores the position of the destination and a unique packet ID in the packet header. The packet ID consists of the source node's destination and a time stamp. A forwarding node broadcasts the packet using a sector as the forwarding area. If nodes outside of the forwarding area hear the broadcast, they ignore it. If the sector is empty no node will respond. After the recovery timer of the forwarding node expires, the node will turn the forwarding area by 60° to the right or left randomly and broadcast the message again. Nodes in the forwarding area that receive the message will start a contention timer based on their distance to the destination. The node whose contention timer expires first will forward the message. Other nodes that hear the message will cancel their respective timers. Once nodes in the destination area receive the message they start destination timers. The timer of the nodes closest to the destination will expire first, it will broadcast a CANCEL packet and start a new, very short destination timer. Nodes that receive a CANCEL packet, cancel their timers and the algorithm is completed.

Among other several beacon-less geographic routing algorithms are Contention-Based Forwarding (CBF) [50], State-free Implicit Forwarding (SIF) [27], Geographic Random Forwarding (GeRaF) [158] and an enhanced version of GeRaF, ALBA-R [21]. All theses algorithms use different forwarding areas, timer functions, and recovery strategies. These protocols have some drawbacks. For example, CBF, SIF and GeRaF require a very high network density. They also produce additional communication overhead for selecting neighbours.

There are many other beacon-based routing protocols proposed in the current literature. Some of these results can be found in [14, 15, 16, 44, 49, 71, 77, 78, 88, 89, 90, 129].

Chapter 3

Algorithms for 2-connected (k, r)-dominating sets

In this chapter we consider the algorithmic problem of finding a 2-connected (k, r)-dominating set in graphs. Recall that given an undirected graph $G = (V, E)$, a 2-connected (k, r)-dominating set, denoted $D_{(2,k,r)}$, is a subset $S \subseteq V$ such that every vertex $u \in V \setminus S$ is within distance k of r vertices of S and the subgraph induced by S is 2-connected. Dominating sets are applicable in wireless sensor networks as virtual backbones or as cluster-based hierarchical structures to achieve efficient routing. In a 2-connected (k, r)-dominating set the distance parameter k allows local availability to dominators, every node not in the $D_{(2,k,r)}$ is dominated by at least r dominators improving the robustness and fault-tolerance of the $D_{(2,k,r)}$ set, and a 2-connected backbone is resilient to a single node or link failure in the network.

The complexity to find a $D_{(2,k,r)}$ in unit disk graphs as well as in general graphs is unknown. Hence, we propose two centralized approximation algorithms to construct a $D_{(2,k,r)}$. The first algorithm is considered in unit disk graphs, presented in Section 3.1. The second algorithm is considered in general graphs, presented in Section 3.2. For both algorithms we assume that for a given graph G for which we compute a $D_{(2,k,r)}$, G is 2-vertex connected and every vertex has at least r k-adjacent vertices. That is, a $D_{(2,k,r)}$ exists in G.

3.1 $D_{(2,k,r)}$ in Unit Disk Graphs

In this section we present our algorithm in Algorithm 2 to construct a 2-connected (k, r)-dominating set in unit disk graphs. We also present theoretic analysis of our algorithm. We give an approximation ratio of Algorithm 2 with respect to the optimal solution and show that if in a graph G a 2-connected (k, r)-dominating set exists, then Algorithm 2 returns such a set.

Algorithm 2 that constructs a $D_{(2,k,r)}$ in unit disk graphs first constructs a (k, r)-dominating set S, after which S is made into a 2-connected (k, r)-dominating set. The construction of a (k, r)-dominating set in Algorithm 2 is dependent on the construction of a distance-k maximal independent set, denoted MIS_k. The construction of an MIS_k is given in Algorithm 1 [156]. For every vertex $v \in V$ that is added to S_0, all vertices $u \in N_k[v]$ are not in S_0 (Algorithm 1, Steps $3 - 5$). This process is repeated until for every vertex $w_1 \notin S_0$ there is a vertex $w_2 \in S_0$ such that $w_1 \in N_k[w_2]$. Hence, the returned set S_0 by Algorithm 1 is an MIS_k.

Algorithm 1 Maximal Distance-k Independent Set

Input: $G = (V, E)$, $k \geq 1$.
Output: A maximal distance-k independent set S_0.

1: $S_0 = \emptyset$, $D = V$.
2: Choose an arbitrary vertex $v \in D$.
3: Add v to S_0 (i.e. $S_0 = S_0 \cup \{v\}$).
4: Delete v and $N_k(v)$ from D (i.e. $D = D \setminus N_k[v]$).
5: **while** $D \neq \emptyset$ **do**
6:     Choose the next vertex $v \in D$.
7:     Add v to S_0 (i.e. $S_0 = S_0 \cup \{v\}$).
8:     Delete v and $N_k(v)$ from D (i.e. $D = D \setminus N_k[v]$).
9: **end while**

In Algorithm 2, in round $i = 1$ an MIS_k set I_1 for graph G is constructed (Step 3). In the next round $i = 2$, another MIS_k set I_2 of G is constructed that does not use any of the same vertices in I_1 (Steps 2 -5). The sets I_1 and I_2 are disjoint MIS_k sets for G. Since a maximal distance-k independent set is a distance-k dominating set, this process is repeated r times to obtain a $D_{(0,k,r)}$ set $S = I_1 \cup I_2 \cup \cdots \cup I_r$ (Steps 2-6). Note that each vertex $v \notin S$ in each round i, where $1 \leq i \leq r$, is dominated by a vertex $u \in I_i$. Hence, each $v \notin S$ is dominated by r vertices in S. Each vertex

$v \in S$, where $v \in I_j$ for $1 \leq j \leq r$, in each round $i \neq j$, for $1 \leq i \leq r$, is dominated by a vertex $u \in I_i$. Hence, each $v \in S$ is dominated by $r-1$ vertices in S and satisfies r-tuple domination.

Once a $D_{(0,k,r)}$ set S is obtained, vertices are added to S such that the result is a 2-connected set. The general idea is as follows. We enumerate all vertices in S by $v_1, v_2, \cdots, v_l$ and find two vertex disjoint paths from v_i to v_{i+1}. A disjoint path can be found between two vertices by the $O(n^2)$ time algorithm for the disjoint path problem given in [82].

In the vertex-disjoint paths problem, for a given graph G and set of k pairs of vertices $(s_1, t_1), \cdots, (s_k, t_k)$ in G (which are sometimes called *terminals*), it must be decided whether or not G has k vertex-disjoint paths $P_1, \cdots, P_k$ in G such that P_i joins s_i and t_i for $i = 1, 2, \cdots k$ [82]. If a set of such paths exist, then the algorithm presented in [82] finds such paths in $O(n^2)$ time.

We apply the algorithm in [82] to the set of vertices obtained in S. The vertices in set S found in Step 6 of Algorithm 2 are enumerated as $v_1, \cdots, v_l$ (Step 8 of Algorithm 2). Denote the vertices in S as terminal vertices. There are l consecutive pairs of terminal vertices in S, namely $(v_1, v_2), (v_2, v_3), \cdots, (v_{l-1}, v_l), (v_l, v_1)$. Since between each pair we would like to find two disjoint paths, then we essentially have $2l$ pairs of terminal vertices, where we apply the disjoint path algorithm in [82] to find $2l$ disjoint paths.

We now show that for a 2-connected graph G, where a 2-connected (k, r)-dominating set exits, the set returned by Algorithm 2 is a $D_{(2,k,r)}$.

Theorem 3.1.1. *For a given graph G, if a 2-connected (k, r)-dominating set exists then the set of vertices H produced by Algorithm 2 is a 2-connected (k, r)-dominating set.*

Proof. Algorithm 2 constructs a $D_{(0,k,r)}$, where each round i produces a new MIS_k set I_i (Algorithm 2, Steps 1-6). An MIS_k is a distance-k dominating set. The set S produced by Algorithm 2 is a union of r distance-k dominating sets (Algorithm 2, Step 6). For each vertex $u \notin S$, u has at least r k-adjacent vertices in S, since in each round i an I_i is constructed, where there is at least one vertex in I_i that dominates u. Vertices selected to be in I_i during round i are not selected during the previous $i-1$ rounds since each new I_i is constructed from the vertex set $V \setminus (I_1 \cup I_2 \cup \cdots \cup I_{i-1})$. Each vertex $u \in S$, where u is selected to be in I_j in round j, is dominated by a

Algorithm 2 Centralized Algorithm for $D_{(2,k,r)}$

Input: $G = (V, E)$, $k \geq 1$, $r \geq 1$.
Output: A 2-connected (k, r)-dominating set H.

1: Let $S = \emptyset$ and $i = 1$.
2: **while** $i \leq r$ **do**
3:     Construct an MIS_k I_i for G using only the vertices in the set $V \setminus (I_1 \cup I_2 \cup \cdots \cup I_{i-1})$.
4:     $i = i + 1$.
5: **end while**
6: Let $S = I_1 \cup I_2 \cup \cdots \cup I_r$.
7: Let $H = \emptyset$.
8: Enumerate all the vertices in S by $v_1, v_2, \cdots, v_l$.
9: **for** $i \leftarrow 1$ to $l - 1$ **do**
10:     Find two disjoint paths P_i and Q_i from v_i to v_{i+1}.
11:     For all vertices u on the paths P_i and Q_i, where $u \notin S$, add u to H.
12: **end for**
13: Find two disjoint paths P_l and Q_l from v_l to v_1.
14: For all vertices u on the paths P_l and Q_l, where $u \notin S$, add u to H.
15: Output H, which is $H = S \cup \{\bigcup_{i=1}^{l} (P_i \cup Q_i)\}$.

vertex $v \in I_i$ in round $i \neq j$. Hence, for each $u \in S$, u has at least $(r-1)$ k-adjacent vertices in S. Therefore, Part 1 of Algorithm 2 produces a (k, r)-dominating set, i.e. $D_{(0,k,r)}$.

Step 8 of Algorithm 2 takes a $D_{(0,k,r)}$ set S and enumerates all the vertices in S by $v_1, v_2, \cdots, v_l$. Denote all vertices in S obtained by Algorithm 2 (Step 8) as *terminal vertices*. For each $1 \leq i \leq l$, let P_i and Q_i be two vertex disjoint paths from v_i to v_{i+1} (note that addition of indices is modulo l). We claim that $H = S \cup \left\{\bigcup_{i=1}^{l} (P_i \cup Q_i)\right\}$ obtained in Step 15 of Algorithm 2 induces a 2-connected subgraph of G. Let G_v be the graph induced by the vertex set $H - v$. We will show that G_v is connected for all $v \in H$.

We consider two cases. In case one we assume vertex v, which is removed from the set H, is a terminal vertex. In case two, we assume vertex v removed from H is not a terminal vertex, i.e. v was added to H in Step 11 of Algorithm 2.

Case 1: $v \in S$.

Without loss of generality, we may assume that $v = v_1$. We first show that there is a path from v_i to v_j in G_v for all $1 < i < j \leq l$. Note that since P_i and Q_i are disjoint, after the removal of vertex v at least one of them does not contain v. Similarly, for every $i < k < j$, since P_k and Q_k are disjoint, after the removal of vertex v one of the paths P_k and Q_k is still a path from v_k to v_{k+1} (see Figure 3.1.1). If P_k does not contain v, then let $R_k = P_k$, otherwise let $R_k = Q_k$. Clearly R_k does not contain the vertex v. Then the path $v_i R_i v_{i+1} R_{i+1} \cdots R_{j-1} v_j$ is a path from v_i to v_j that does not contain vertex v and is therefore, a path in G_v.

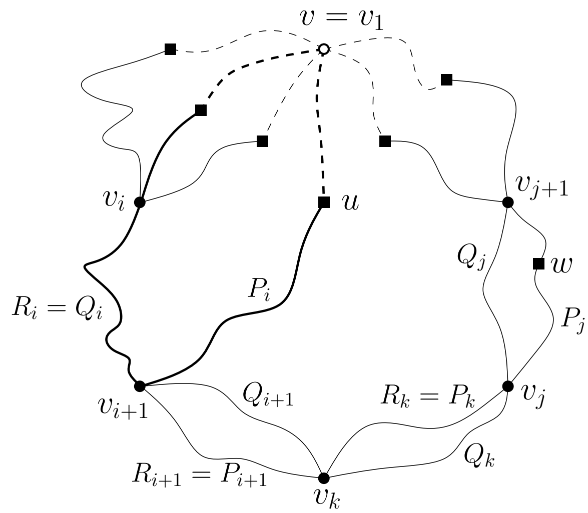


Figure 3.1.1: Removal of terminal vertex v. Circles indicate terminal vertices and squares indicate non-terminal vertices. Dashed lines indicate deleted edges that are incident to v.

Now, let u and w be arbitrary vertices in G_v. We will assume that u and w are non-terminal vertices in G_v (see Figure 3.1.1). In this subcase we show that there is a path from u to w. The case where u is a terminal vertex and w is a non-terminal vertex is similar. Assume without loss of generality that u is on the path P_i from v_i to v_{i+1} and w is on the path P_j from v_j to v_{j+1} for some $i < j$. Before the removal of vertex v, $P_i \cup Q_i$ is a cycle containing v_i and v_{i+1}. Thus, there must be a path from u to either v_i or v_{i+1} in G_v. Similarly, there is a path from w to v_j or to v_{j+1} in G_v. Note that if $i = 1$ or if $j = l$, then there is a path in G_v from u to v_2 and from w to v_l respectively. Since $v_i, v_{i+1}, v_j, v_{j+1}$ are all terminal vertices, there is always a path between any two of them by the previous subcase. Therefore, there is a path from u to w in G_v.

<u>Case 2: $v \notin S$</u>.

Next we consider G_v, where v is not a terminal vertex (see Figure 3.1.2). We first show that there is a path between two terminal vertices v_i and v_j for $1 \leq i < j \leq l$. For all $i \leq k < j$, we observe that at least one of P_k and Q_k does not contain v, since P_k and Q_k were chosen to be disjoint. Therefore, there exists a path from v_k to v_{k+1} in G_v for all k such that $i \leq k < j$. Concatenating these paths gives us a path from v_i to v_j in G_v.

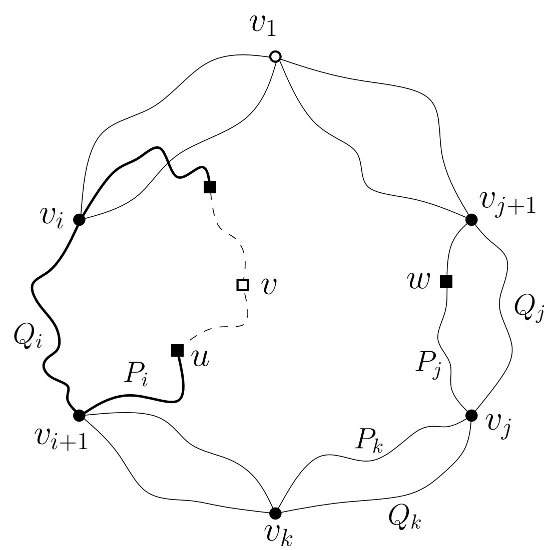


Figure 3.1.2: Removal of non-terminal vertex v. Circles indicate terminal vertices and squares indicate non-terminal vertices. Dashed lines indicate deleted edges that are incident to v.

Now let u and w be non-terminal vertices in G_v (see Figure 3.1.2). We will show that there is a path in G_v from u to w. The case where u is a terminal vertex and w a non-terminal vertex is similar. Assume u is on some path P_i and w is on some path P_j. Since $P_i \cup Q_i$ is a cycle containing u and v_i before the removal of vertex v, there is a path from u to v_{i+1} in G_v. Similarly, there is a path from w to v_j in G_v. Since v_{i+1} and v_j are both terminal vertices, there is a path between them in G_v by the previous subcase. Therefore, there is a path from u to w in G_v.

□

We now present our approximation ratio of Algorithm 2. Zhang et al. showed in [156] that in a unit disk graph, the number of independent vertices in the k-neighbourhood of a vertex is upper bounded by β as is given in Lemma 3.1.1. We use this result to give an approximation ratio of Algorithm 2.

Lemma 3.1.1. *[156] Let G be a unit disk graph and I be a distance-k independent set of G. Then for any vertex u in G, $N_k(u)$ contains at most β vertices from I, where*

$$\beta = \begin{cases} 5, & \text{if } k = 1, \\ 21, & \text{if } k = 2, \\ 5 + \frac{4k(k+1)}{\lceil \frac{1}{2} \lfloor \frac{k-1}{2} \rfloor \rceil} & \text{if } k \geq 3. \end{cases}$$

Theorem 3.1.2. *For a unit disk graph $G = (V, E)$, if a 2-connected (k, r)-dominating set exists, Algorithm 2 returns a 2-connected (k, r)-dominating set with approximation ratio of $2D\beta$, where D is the diameter of graph G and β is at most $O(k)$.*

Proof. From Steps 1-6 of Algorithm 2 where a $D_{(0,k,r)}$ is constructed, we know that for each vertex v at most β vertices k-adjacent to v are chosen to be in S in each round i of I_i (Lemma 3.1.1). Thus, for each vertex, there are at most βr k-adjacent vertices in S. For the optimum solution of $D_{(0,k,r)}$, for each vertex there are at least r k-adjacent vertices in S. Therefore, the number of vertices in S is at most β times of the optimum solution of $D_{(0,k,r)}$.

Vertices in S are denoted as terminal vertices and for two consecutive terminal vertices in an enumeration we find two vertex disjoint paths to connect them. The length of each path is at most size of the diameter of graph G. Hence, at most $2D|S|$ vertices are added to produce a 2-connected (k, r)-dominating set. Therefore, the number of vertices in H is at most $2D\beta$ times of the solution of $D_{(2,k,r)}$. □

Theorem 3.1.3. *Algorithm 2 computes a 2-connected (k, r)-dominating set for a unit disk graph in $O(n^3)$ time.*

Proof. Consider an arbitrary unit disk graph $G = (V, E)$, where $n = |V|$. Algorithm 1 returns a maximal distance-k independent set in $O(n^2)$ time. Steps 1-6 of Algorithm 2 takes at most rn^2 operations. To find two disjoint paths between any two vertices in graph G (Algorithm 2, Step 10) takes $O(n^2)$ time. Hence, for l pairs such vertices, the computation time of finding disjoint paths between l pairs of vertices is $lO(n^2)$. Since $l = |S|$ and $|S|$ can be at most be n, in the worst case it takes $O(n^3)$ time to find 2 disjoint paths between l pairs of terminal vertices (Algorithm 2, Steps 9-14). Hence, Algorithm 2 requires $O(n^3)$ computation time. □

3.2 $D_{(2,k,r)}$ in General Graphs

In this section we present a centralized algorithm to construct a 2-connected (k,r)-dominating set in general graphs. Our method of finding a $D_{(2,k,r)}$ in general graphs consists of two parts. In part one of our method, we use the algorithm introduced by Spohn et al. [126] to construct a (k,r)-dominating set S of graph G. In part two of our method, we use Steps 7-15 of Algorithm 2 and obtain a 2-connected (k,r)-dominating set H of G.

The algorithm introduced by Spohn et al. is a greedy based heuristic [126]. Before presenting an overview of the algorithm, we first introduce some needed notation. Consider a graph $G = (V, E)$ and let $S \subseteq V$ be a (k,r)-dominating set for G.

- Let Δ_k denote the largest cardinality among all k-neighbourhoods in G. i.e. $\Delta_k = max\left\{|N_k[u]| \,\middle|\, u \in V\right\}$.
- For every non r-dominated vertex u, let $f(u)$ be the number of vertices in S that are k-adjacent to u. Let $D(u)$ denote the number of vertices needed to dominate u. i.e. $D(u) = r - f(u)$.
- Let $T(u) = \sum_{v \in N_k[u]} D(v)$.

A (k,r)-dominating set S is constructed as follows. A vertex v with the maximum T value is repeatedly selected to be a dominator in S until every vertex is r-dominated. That is, T values of all vertices are updated until each vertex is either in the (k,r)-dominating set S or is dominated by a vertex in S (i.e. the D values of all vertices become 0) [126].

Li et al. show that this method returns a (k,r)-dominating set of size at most $\ln \Delta_k |OPT|$ for any undirected graph, where OPT is an optimum solution for the minimum (k,r)-dominating set [93]. As we use Steps 7-15 of Algorithm 2 on this set S, then by Theorem 3.1.1 the result returned by Algorithm 2 is a 2-connected (k,r)-dominating set.

Theorem 3.2.1. *For an undirected graph $G = (V, E)$, if a 2-connected (k,r)-dominating set exists, the set of vertices H produced by the above procedure returns a 2-connected (k,r)-dominating set and has size at most $2D \ln \Delta_k$ times the optimum solution of the minimum 2-connected (k,r)-dominating set, where D is the diameter of G and Δ_k is the largest cardinality among all k-neighbourhoods in G.*

Proof. The method introduced by Spohn et al. determines a (k, r)-dominating set of G [126] with $\ln \Delta_k$-approximation [93]. A minimum 2-connected (k, r)-dominating set will not be smaller than a minimum (k, r)-dominating set. Hence, by Theorem 3.1.1 and by finding two disjoint paths between any two consecutive terminal vertices in an enumeration defined in Step 8 of Algorithm 2, where each path is of length at most diameter D, will result in a 2-connected (k, r)-dominating set with $2D \ln \Delta_k$-approximation algorithm. □

Theorem 3.2.2. *Algorithm 2 computes a* 2*-connected* (k, r)*-dominating set for general graphs in* $O(n^3)$ *time.*

Proof. Consider an arbitrary graph $G = (V, E)$, where $n = |V|$. Spohn et al. show that for $\Delta_k \leq n$ (i.e. Δ_k increases as k approaches the network diameter, and is at most n when k is equal to the diameter), their algorithm that returns a (k, r)-dominating set runs in $O(n^3)$ time [126].

To find two disjoint paths between any two vertices in graph G takes $O(n^2)$ time. Hence, for l pairs such vertices, the computation time is in $lO(n^2)$ time. Since $l = |S|$ and $|S|$ can be at most be n, in the worst case to find two disjoint paths between l pairs of vertices takes $O(n^3)$ time (Algorithm 2, Steps 9-14). Hence, considering Part 1 and Part 2 together, we conclude that the method of constructing a 2-connected (k, r)-dominating set in general graphs requires $O(n^3)$ computation time. □

3.3 Summary

In this chapter we presented two centralized algorithms to find a 2-connected (k, r)-dominating set in unit disk graphs and in general graphs, where k and r are fixed positive integers. For both algorithms it was showed that if for a given graph G such a set exists then both algorithms returns such a set.

The first algorithm in unit disk graphs has an approximation ratio of $2D\beta$ of the optimal solution, where D is the diameter of the graph and β is $O(k)$. The complexity of the algorithm is $O(n^3)$, where $n = |V(G)|$. The algorithm in general graphs has an approximation ratio of $2D \ln \Delta_k$, where D is the diameter of the graph and Δ_k is the largest cardinality among all k-neighbourhoods in G. The complexity of this algorithm is $O(n^3)$, where $n = |V(G)|$.

Chapter 4

On the total (k, r)-domination number of the random graphs

In this chapter we consider upper bounds on the size of the total (k, r)-dominating set in graphs. We study this problem in connection to deriving upper bounds on the number of sinks in a wireless sensor network. Recall that a set $S \subseteq V$ is a *total* (k, r)*-dominating set* of a graph G, if every vertex $v \in V$ is within distance k of r vertices in S. The minimum cardinality of a total (k, r)-dominating set G is the *total* (k, r)*-domination number* of G, denoted $\gamma^t_{(k,r)}(G)$. In this chapter we consider upper bounds on the total (k, r)-domination number in general graphs and in random graphs. Recall that a *random graph* $G(n, p)$ consists of n vertices with each of the potential $\binom{n}{2}$ edges being inserted independently with probability p.

In Section 4.2 we derive an upper bound on $\gamma^t_{(k,r)}(G)$ in graphs with large girth. In Section 4.3 we derive upper bounds on total (k, r)-domination number of the random graphs, where we consider the cases $k = 2$, $k = 3$ and $k > 3$. Before we present our results we first present preliminary notations and definitions in Section 4.1.

4.1 Notation and Basic Facts

A *probability space* is a triple $\big(\Omega, \Sigma, P\big)$, where Ω is a finite set, Σ is $\mathscr{P}(\Omega)$, the set of all subsets of Ω, P is a non-negative measure on Σ and $P(\Omega) = 1$. Then P is determined by the function $\Omega \to [0, 1]$, $\omega \to P(\{\omega\})$, namely

$$P(A) = \sum_{\omega \in A} P(\{\omega\}), \text{ where } A \subset \Omega.$$

Thus, Ω is the *sample space* that represents all outcomes. Σ is a collection of subsets of Ω, called the *event space*. P is the probability function that assigns probabilities to the events in Σ. [12]

All the definitions that follow and in Section 4.1.1 are taken from [135].

For any event A in the probability space Ω, we write $\mathbb{P}[A]$ for the probability of A in the space.

For two disjoint subsets A and B of Ω, A and B are called *disjoint events*. For disjoint events A and B, $\mathbb{P}[A \cup B] = \mathbb{P}[A] + \mathbb{P}[B]$.

Two events A and B are *independent* if and only if $\mathbb{P}[A \cap B] = \mathbb{P}[A] \cdot \mathbb{P}[B]$.

If A, B are events in the probability space Ω and $A \subset B$ then $\mathbb{P}[A] \leq \mathbb{P}[B]$.

4.1.1 Linearity of Expectation

For a given (discrete) probability space Ω any mapping $X : \Omega \to \mathbb{Z}$ is called a *random variable*.

The *expected value* of a random variable X, denoted $\mathbb{E}[X]$, is defined as

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} X(\omega)\mathbb{P}[\omega] = \sum_{x \in \mathbb{Z}} x\mathbb{P}[X = x]$$

given that $\sum_{x \in \mathbb{Z}} |x|\mathbb{P}[X = x]$ converges.

An *indicator random variable* (or a *Bernoulli random variable*) is a random variable X_i such that

$$X_i = \begin{cases} 1 & \text{with some probability } p \\ 0 & \text{with probability } 1 - p. \end{cases}$$

The expected value of a sum of random variables is the sum of the expected values of the variables. More formally, *Linearity of Expectation* states the following.

For any random variables X and Y that may not necessarily be independent

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y].$$

For any random variable X and a constant $c \in \mathbb{R}$, $\mathbb{E}[cX] = c\mathbb{E}[X]$. Thus, for random variables X and Y and constants $c_1, c_2 \in \mathbb{R}$, $\mathbb{E}[c_1X + c_2Y] = c_1\mathbb{E}[X] + c_2\mathbb{E}[Y]$. Thus, expectation is a linear function and is applicable to more than two random variables.

For any random variables $X_1, \cdots, X_n$ and constants $c_1, \cdots, c_n \in \mathbb{R}$,

$$\mathbb{E}\left[\sum_{i=1}^{n} c_i X_i\right] = \sum_{i=1}^{n} c_i \mathbb{E}\left[X_i\right].$$

X is a *binomial random variable*, denoted $X \sim B(n,p)$, if over total n trials the probability each trial yields a success is p. Note that, the Bernoulli distribution is a special case of the Binomial distribution with $n = 1$.

If $X \sim B(n,p)$ is the random variable and $X = X_1 + \cdots + X_n$, where each X_i is an indicative random variable with probability p then

$$\begin{aligned}\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] &= \sum_{i=1}^{n} \mathbb{E}\left[X_i\right] \\ &= \sum_{i=1}^{n} (1 \cdot p + 0(1-p)) = \sum_{i=1}^{n} p \\ &= np\end{aligned}$$

4.2 Total (k,r)-domination number in graphs of large girth

In this section we derive an upper bound on the total (k,r)-domination number in graphs. For a given graph G, the girth of G is the length of the shortest cycle contained in G. Theorem 4.2.1 presents our result in graphs of girth at least $2k+1$. Although our result is not tight, we do obtain a bound with a relatively simple expression.

Theorem 4.2.1. *Consider a graph G, where $n = |V(G)|$. Let G be of minimum degree at least d, and of girth at least $2k+1$. Then for any positive integers k and r,* $\gamma^t_{(k,r)}(G) \leq \dfrac{2nr}{(d-1)^k} + nre^{-\frac{r}{4}}$.

Proof. Let us pick, randomly and independently, each vertex $v \in V(G)$ with probability p to be defined later. Let $S \subset V(G)$ be the set of vertices picked. S is a random set and is part of the total (k,r)-dominating set that we would like to obtain.

For every vertex $v \in V(G)$, let X_v denote the number of vertices in $N_k(v)$ that are also in S. Let Y be the set such that $Y = \{v \in V(G) | X_v \leq r - 1\}$. Note that S is a random set and $\mathbb{E}\big[|S|\big] = np$. We now estimate $\mathbb{P}\big[X_v < r\big]$.

For a given vertex $v \in Y$, let $m = |N_k(v)|$. Since by assumption graph G has girth at least $2k + 1$ then it follows that $m \geq (d-1)^k$. We show this by contradiction.

Assume that $m < (d-1)^k$. Then there exist vertices u_1, $u_2 \in N_k(v)$ such that there is a vertex $w \in N_k(v)$ and $w \in (N_k(u_1) \cap N_k(u_2))$. Vertex w is at most distance k from v. Thus, the distance from w to v through the path containing u_1 is at most k. Similarly, the distance from w to v through the path containing u_2 is also at most k. Thus, making a cycle of length at most $2k$, which is a contradiction. Therefore, by the assumption that G has girth at least $2k + 1$, it follows that $m \geq (d-1)^k$.

It can be seen that X_v is a $B(m, p)$ random variable. We use the well known Chernoff Bound [10, 37, 31, 7, 68, 2] to bound $\mathbb{P}\big[X_v < r\big]$.

The Chernoff Bound states: for any $a > 0$ and random variable X that has binomial distribution with probability p and mean pn,

$$\mathbb{P}\big[X - pn < -a\big] < e^{-a^2/2pn}. \tag{1}$$

We set $a = \epsilon pm$, where we let $\epsilon = 1 - \dfrac{r}{pm}$. Hence, $a = pm - r$, which results in $r = pm - a$. Then, by the Chernoff Bound given in Eq. 1 we have,

$$\begin{aligned}
\mathbb{P}\big[X_v < r\big] &= \mathbb{P}\big[X_v < pm - a\big] \\
&< e^{-\frac{a^2}{2pm}} = e^{-\frac{\epsilon^2 (pm)^2}{2pm}} = e^{-\frac{\epsilon^2 pm}{2}} \\
&\leq e^{-\frac{\epsilon^2 p(d-1)^k}{2}}.
\end{aligned} \tag{2}$$

Chernoff's bound holds whenever $\epsilon > 0$, that is when $1 - \dfrac{r}{pm} > 0$. Thus, it holds when $p > \dfrac{r}{(d-1)^k}$. By setting $p = \dfrac{2r}{(d-1)^k}$, from Eq. 2 we obtain

$$\mathbb{P}\big[X_v < r\big] < e^{-\epsilon^2 \left(\frac{2r}{(d-1)^k}\right) \frac{(d-1)^k}{2}} = e^{-\epsilon^2 r}.$$

For each vertex $v \in Y$, where $X_v \leq r-1$, we pick a set A_v of r vertices in $N_k(v)$ arbitrarily. For vertices v that satisfy $X_v \geq r$, $A_v = \emptyset$. Let $A = \bigcup_{v=1}^{n} A_v$. Clearly, $S \cup A$ is a total (k,r)-dominating set. We now estimate $\mathbb{E}\big[|A|\big]$.

By linearity of expectation, we obtain

$$\begin{aligned}\mathbb{E}\big[|A|\big] &= \mathbb{E}\left[\left|\bigcup_{v=1}^{n} A_v\right|\right] \leq \mathbb{E}\left[\sum_{v=1}^{n} |A_v|\right] \\ &= \sum_{v=1}^{n} \mathbb{E}\big[|A_v|\big] \leq nre^{-\epsilon^2 r}.\end{aligned}$$

Again by the linearity of expectation, we now estimate $\mathbb{E}\big[|S \cup A|\big]$.

$$\begin{aligned}\mathbb{E}\big[|S \cup A|\big] &= \mathbb{E}\big[|S|\big] + \mathbb{E}\big[|A|\big] \\ &\leq np + nre^{-\epsilon^2 r} \\ &= \frac{2nr}{(d-1)^k} + nre^{-\epsilon^2 r}.\end{aligned}$$

Therefore, we have shown that there exists a total (k,r)-dominating set in G, where

$$\begin{aligned}\gamma^t_{(k,r)}(G) &\leq \frac{2nr}{(d-1)^k} + nre^{-\epsilon^2 r} \\ &\leq \frac{2nr}{(d-1)^k} + nre^{-\frac{r}{4}}\end{aligned}$$

since $\epsilon > 1/2$. $\square$

4.3 Total (k,r)-domination number in random graphs

In this section we derive upper bounds for the total (k,r)-domination number in random graphs. Note that by definition the size of any total (k,r)-dominating set must be at least $r+1$. In Section 4.3.1 we give upper bounds on $\gamma^t_{(2,r)}(G(n,p))$ with $p \in (0,1)$ non-fixed. In Section 4.3.2 we present an upper bound on $\gamma^t_{(k,r)}(G(n,p))$ with $p \in (0,1)$ non-fixed. We first present some facts and definitions needed in Sections 4.3.1 and 4.3.2.

Recall that an event holds *asymptotically almost surely (a.a.s.)* if the probability it holds tends to 1 as n tends to infinity [2].

If X is a non-negative random variable with finite mean and $a > 0$, then

$$\mathbb{P}\big[X \geq a\big] \leq \frac{\mathbb{E}\big[X\big]}{a}.$$

This is known as Markov's Inequality [12].

4.3.1 Upper bounds on $\gamma^t_{(2,r)}(G(n,p))$

We first present an upper bound on $\gamma^t_{(2,r)}(G(n,p))$ in Theorem 4.3.2. It is well known that for fixed $p < 1$, the diameter of $G(n,p)$ is two (see Theorem 4.3.1).

Theorem 4.3.1. *[70] Let $p = c\sqrt{\frac{\ln n}{n}}$. For $c > \sqrt{2}$, $G(n,p)$ almost surely has diameter less than or equal to two.*

From Theorem 4.3.1, for $p \geq \sqrt{2}\sqrt{\frac{\ln n}{n}}$, a.a.s. $\gamma^t_{(2,r)}(G(n,p)) = r+1$ directly follows. For a weaker value of p, we can still have a.a.s. $\gamma^t_{(2,r)}(G(n,p)) = r+1$. In Theorem 4.3.1, we weaken the minimum value of p from $\sqrt{2}\sqrt{\frac{\ln n}{n}} = \sqrt{\frac{2}{\log e}}\sqrt{\frac{\log n}{n}}$ to $p \geq c\sqrt{\frac{\log n}{n}}$, for a fixed constant $c > 1$.

Theorem 4.3.2. *Let $c > 1$ be a fixed constant. Then for any positive integer r, in a random graph $G(n,p)$ with $p \geq c\sqrt{\dfrac{\log n}{n}}$, a.a.s. $\gamma^t_{(2,r)}(G(n,p)) = r+1$.*

Proof. Let $D \subseteq V(G(n,p))$ be a total $(2,r)$-dominating set and let the vertices in D be labelled as $v_1, v_2, \cdots, v_i, \cdots, v_{r+1}$, where $1 \leq i \leq r+1$. The probability that a vertex $u \in V(G(n,p))$ is not within distance-2 from a vertex $v_i \in D$ is given by

$$\mathbb{P}[v_i \notin N_2(u)] \leq (1-p^2)^{n-2}. \qquad (3)$$

Let X be a random variable that denotes the number of vertices $u \in V(G(n,p))$, where the number of 2-adjacent vertices of u in D is less than r. We would like to show that as n tends to infinity, the number of vertices in $V(G(n,p))$ with less than r dominators tends to 0. That is, $\mathbb{P}\big[X > 0\big] \to 0$ as $n \to \infty$.

A fixed vertex u is defined *bad*, if u in its 2-neighbourhood has less than r dominators in D. By linearity of expectation we have

$$\mathbb{E}[X] = n \cdot \mathbb{P}[\text{fixed } u \text{ is bad}]. \tag{4}$$

Let X_u be the random variable that denotes the number of non-dominators of u. We note that u itself may be an element of D. Then

$$\begin{aligned}\mathbb{E}[X_u] &\leq r\,(1-p^2)^{n-2} \\ &\leq r\,e^{-p^2(n-2)} \qquad \big(\text{by } 1-x \leq e^{-x}\big).\end{aligned}$$

By Markov's Inequality we have $\mathbb{P}[X_u > 0] \leq \mathbb{E}[X_u] \leq r\,e^{-p^2(n-2)}$. Thus,

$$\begin{aligned}\mathbb{P}[\text{fixed } u \text{ is bad}] &\leq \mathbb{P}[X_u > 0] \\ &\leq r\,e^{-p^2(n-2)}.\end{aligned} \tag{5}$$

By Eq. 4 and Eq. 5 we have $\mathbb{E}[X] \leq n\,r\,e^{-p^2(n-2)}$. By Markov's Inequality it follows,

$$\mathbb{P}[X > 0] \leq \mathbb{E}[X] \leq n\,r\,e^{-p^2(n-2)} \tag{6}$$

From Eq. 6, as $n \to \infty$ the value of p follows.

$$\begin{aligned}e^{p^2(n-2)} > r \cdot n \quad &\implies \quad e^{p^2(n-2)} > n \\ &\implies \quad p^2(n-2) > \log n \\ &\implies \quad p > \sqrt{\frac{\log n}{n-2}} \\ &\implies \quad p \geq \sqrt{\frac{\log n}{n}}.\end{aligned}$$

Let $p \geq c\sqrt{\dfrac{\log n}{n}}$, where $c > 1$ is a constant. We now determine the value of $e^{p^2(n-2)}$.

$$e^{p^2(n-2)} \geq \left(e^{\log n}\right)^{c^2\left(\frac{n-2}{n}\right)} = n^{c^2\left(1-\frac{2}{n}\right)} \tag{7}$$

From Eq. 6 and Eq. 7, we have

$$n\,r\,e^{-p^2(n-2)} \leq \frac{n\,r}{n^{c^2\left(1-\frac{2}{n}\right)}} = \frac{r}{n^{c^2\left(1-\frac{2}{n}\right)-1}}. \tag{8}$$

Since $c^2 > 1$ and as $n \to \infty$, $c^2\left(1 - \frac{2}{n}\right) > 1$ and hence, $c^2\left(1 - \frac{2}{n}\right) - 1 > 0$.

Thus, as $n \to \infty$, $\frac{r}{n^{c^2\left(1-\frac{2}{n}\right)-1}} \to 0$. Therefore, From Eq. 6 and Eq. 8 we have $\mathbb{P}[X > 0] \to 0$ as $n \to \infty$. □

Next we present an upper bound on $\gamma^t_{(k,r)}(G(n,p))$ in Theorem 4.3.5. We first present a needed result in Theorem 4.3.3 on random graphs of diameter greater than two.

Theorem 4.3.3. *[12] Let c be a positive constant, $d = d(n) \geq 2$ a natural number, and define $p = p(n, c, d)$, $0 < p < 1$, by*

$$p^d n^{d-1} = \log(n^2/c).$$

Suppose that $pn/(\log n)^3 \to \infty$. Then in $G(n,p)$ we have

$$\lim_{n\to\infty} \mathbb{P}(\textit{diam } G = d) = e^{-c/2} \quad \textit{and} \quad \lim_{n\to\infty} \mathbb{P}(\textit{diam } G = d+1) = 1 - e^{-c/2}.$$

For a random graph of diameter at most d, from Theorem 4.3.3, $p = \sqrt[d-1]{\frac{\log(n^2/c)}{n^{d-2}}}$. In Section 4.3.2, we weaken the value of p and give an upper bound on the total (k, r)-domination number in random graphs.

4.3.2 Upper bound on the $\gamma^t_{(k,r)}(G(n,p))$

In this section we present an upper bound on $\gamma^t_{(k,r)}(G(n,p))$ for fixed positive integers k and r. In Theorem 4.3.2, we proved that in a random graph $G(n,p)$ with $p \geq c\sqrt{\frac{\log n}{n}}$ and fixed constant $c > 1$, a.a.s. $\gamma^t_{(2,r)}(G(n,p)) = r + 1$. Note that in the proof of Theorem 4.3.2, to determine the probability that a vertex u is not within distance-2 from a dominator vertex v_i (given by Eq. 3) uses the fact that the connecting vertex w_i chosen to connect u to v_i (to obtain a path of length 2) cannot be chosen again to connect u and v_i via a different path (since the two paths would be the same). Hence, the probability of connecting u to v_i through different paths of length 2 are independent of each other.

The above holds true when we consider an upper bound for the total $(2, r)$-domination number and makes calculations relatively easy. However, once we generalize to give an upper bound for the total (k, r)-domination number, we cannot easily obtain this independence when considering paths of length greater than 2 from u to v_i. When considering paths of length k from u to v_i for the general case of total (k, r)-domination number it becomes more difficult to calculate the probability that there is a path of length k from u to v_i via $k - 1$ vertices. There may be two different paths P_1 and P_2 from u to v_i that may share some edges between any of the connecting $k - 1$ vertices and hence, are not independent anymore as they were in the case of total $(3, r)$-domination number (see Figure 4.3.1).

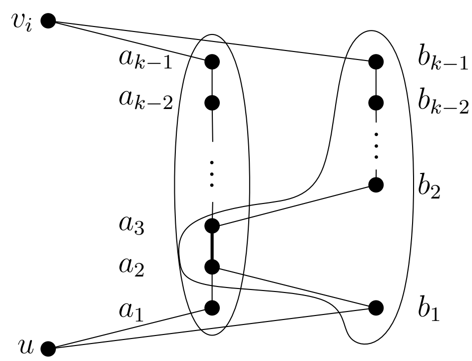


Figure 4.3.1: $P_1 = u\, a_1\, a_2\, a_3 \cdots a_{k-2}\, a_{k-1}\, v_i$ and $P_2 = u\, b_1\, a_2\, a_3\, b_2 \cdots b_{k-2}\, b_{k-1}\, v_i$ are two paths between u and v_i that share an edge, namely (a_2, a_3).

In Theorem 4.3.5, we generalize the upper bound on $\gamma^t_{(2,r)}(G(n, p))$ presented by Theorem 4.3.2 to give an upper bound on $\gamma^t_{(k,r)}(G(n, p))$. Note that from Theorem 4.3.3, the diameter of $G(n, p)$ is at most d for $p = \sqrt[d-1]{\frac{\log(n^2/c)}{n^{d-2}}}$. In Theorem 4.3.5, we weaken the value of p to $p' \geq d\sqrt[d]{\frac{\log n}{n^{d-1}}}$. Easy calculation shows that $d\sqrt[d]{\frac{\log n}{n^{d-1}}} < \sqrt[d-1]{\frac{\log(n^2/c)}{n^{d-2}}}$ for $d \leq (\log n)^{\epsilon}$ for a constant $\epsilon < \frac{1}{2}$. In particular $p' < p$ holds when d is constant. Our proof of Theorem 4.3.5 uses Janson's Inequality, which we present here first [2].

Let Ω be a finite universal set and let R be a random subset of Ω given by

$$\mathbb{P}[r \in R] = p_r, \tag{9}$$

these events mutually independent over $r \in \Omega$. Let $\{A_i\}_{i \in I}$ be subsets of Ω, where I

is a finite index set. Let B_i be the event that $A_i \subseteq R$. Let X_i be the indicator random variable for B_i and $X = \sum_{i \in I} X_i$ the number of $A_i \subseteq R$. Hence, $\mathbb{P}[X = 0] = \mathbb{P}\left[\bigcap_{i \in I} \overline{B_i}\right]$. For $i, j \in I$ we write $i \sim j$ if $i \neq j$ and $A_i \cap A_j \neq \emptyset$. Thus, define $\Delta = \sum_{i \sim j} \mathbb{P}[B_i \cap B_j]$. Set $\mu = \mathbb{E}[X] = \sum_{i \in I} \mathbb{P}[B_i]$. In Theorem 4.3.4 we state Janson's Inequality. [2]

Theorem 4.3.4. *[2] Let $\{B_i\}_{i \in I}$, Δ, μ be as above. Then* $\mathbb{P}\left[\bigcap_{i \in I} \overline{B_i}\right] \leq e^{-\mu + \Delta/2}$.

Theorem 4.3.5. *For any positive integers $k \geq 3$ and r, in a random graph $G(n,p)$ with $p \geq k \sqrt[k]{\dfrac{\log n}{n^{k-1}}}$, a.a.s.* $\gamma^t_{(k,r)}(G(n,p)) = r + 1$.

Proof. Let $D \subseteq V(G(n,p))$ be a total (k,r)-dominating set and let the vertices in D be labelled as $v_1, v_2, \cdots, v_i, \cdots, v_{r+1}$, where $1 \leq i \leq r+1$. The probability that a vertex $u \in V(G(n,p))$ is not within distance-k from a vertex $v_i \in D$ is denoted by $\mathbb{P}[v_i \notin N_k(u)]$.

Let X be a random variable that denotes the number of vertices $u \in V(G(n,p))$, where the number of k-adjacent vertices of u in D is less than r. We would like to show that the number of vertices in $V(G(n,p))$ with less than r dominators tends to 0. That is, $\mathbb{P}[X > 0] \to 0$ as $n \to \infty$.

We define a fixed vertex u as *bad*, if u in its k-neighborhood has less than r dominators in D. By linearity of expectation we have

$$\mathbb{E}[X] = n \cdot \mathbb{P}[\text{fixed } u \text{ is bad}]. \tag{10}$$

There are $n-2$ vertices aside from u and v_i to connect u to v_i via a path of length k. To connect u to v_i such that $d(u, v_i) = k$, additional $k-1$ connecting vertices are necessary to create a path of length k from u to v_i. There are $\binom{n-2}{k-1}$ possible ways to choose these $k-1$ vertices. Hence, we have $\binom{n-2}{k-1}$ such sets that consist of $k-1$ vertices. We denote these sets by $S_1, S_2, \cdots, S_{\binom{n-2}{k-1}}$.

We would like to show that $\mathbb{P}[v_i \notin N_k(u)] \to 0$ as $n \to \infty$. This is equivalent to showing that the probability one of S_i connects u to v_i via a path of length k tends to 1 as $n \to \infty$.

Let $S_i = \{a_{i_1}, a_{i_2}, \cdots, a_{i_{k-1}}\}$. For any pair u and v_i that are fixed, we number all other $n-2$ vertices and assume that all vertices in S_i are connected in ascending order of the vertex number. Note that some edges in S_i and S_j, where $i \neq j$ are the same. To calculate the probability of the appearance of the $k-2$ edges in each S_i we must consider the dependencies between any two sets S_i and S_j for $i \neq j$. To do this, we use Janson's inequality from Theorem 4.3.4.

Let R be the set $E(G(n,p))$ and let A_i be the set of edges such that $A_i = \{ua_{i_1}, a_{i_1}a_{i_2}, \cdots, a_{i_{k-2}}a_{i_{k-1}}, a_{i_{k-1}}v_i\}$. Let B_i be the event that $A_i \subseteq R$. So, $\mathbb{P}[A_i \in R] = \mathbb{P}[B_i]$. Let X_i be the indicator random variable for B_i and $X_B = \sum_{i=1}^{\binom{n-2}{k-1}} X_i$ be the number of $A_i \subseteq R$. Hence, $\mathbb{P}[X_B = 0] = \mathbb{P}\left[\bigcap_{i=1}^{\binom{n-2}{k-1}} \overline{B_i}\right]$. For $1 \leq i, j \leq \binom{n-2}{k-1}$ we write $i \sim j$ if $i \neq j$ and $A_i \cap A_j \neq \emptyset$. Δ is defined as $\sum_{i \sim j} \mathbb{P}[B_i \cap B_j]$. We would like to show that $\mathbb{P}[X_B = 0] \to 0$ as $n \to \infty$.

First we determine $\mu = \mathbb{E}[X_B] = \sum_{i=1}^{\binom{n-2}{k-1}} \mathbb{P}[B_i]$.

$$
\begin{aligned}
\mathbb{E}[X_B] &= \mathbb{E}\left[\sum_{i=1}^{\binom{n-2}{k-1}} X_i\right] = \sum_{i=1}^{\binom{n-2}{k-1}} \mathbb{E}[X_i] = \sum_{i=1}^{\binom{n-2}{k-1}} \mathbb{E}[B_i] \\
&= \binom{n-2}{k-1} p^k \geq \left(\frac{n-2}{k-1}\right)^{k-1} p^k \qquad \left(\text{by } \binom{n}{k} \geq \left(\frac{n}{k}\right)^k\right) \\
&\geq \frac{(n-2)^{k-1}}{(k-1)^{k-1}} \left(k \sqrt[k]{\frac{\log n}{n^{k-1}}}\right)^k \\
&= \frac{(n-2)^{k-1}}{(k-1)^{k-1}} k^k \frac{\log n}{n^{k-1}} = \left(\frac{k^k}{(k-1)^{k-1}}\right) \left(\frac{n-2}{n}\right)^{k-1} \log n \\
&= k \left(\frac{k}{k-1}\right)^{k-1} \left(1 - \frac{2}{n}\right)^{k-1} \log n \\
&\geq k \left(1 - \frac{2}{n}\right)^{k-1} \log n \\
&\geq 0.9k \log n
\end{aligned}
\tag{11}
$$

for n large enough. Thus, from Janson's Inequality let $\mu = 0.9k \log n$.

Now we determine Δ. Assume that the number of edges shared between any given A_i and A_j is given by t and hence, A_j shares at least t vertices with A_i. There are $\binom{n-2}{k-1}$ such A_i sets. We fix one such set A_i and determine the dependencies between A_i and all other sets A_j, where $j \neq i$. Thus, we have

$$\begin{aligned}
\Delta &= \sum_{i=1}^{\binom{n-2}{k-1}} \mathbb{P}[B_i \cap B_j] \\
&\leq \binom{n-2}{k-1} \sum_{\substack{i \text{ fixed} \\ j \sim i}}^{\binom{n-2}{k-1}} \mathbb{P}[B_j \cap B_i] \\
&\leq \binom{n-2}{k-1} \sum_{t=1}^{k-1} \binom{k}{t} \binom{n-2}{k-1-t} p^{2k-t}. \qquad (12)
\end{aligned}$$

In Equation 12, the probability that a fixed A_i intersects (i.e. shares) at t edges with a set A_j for $i \neq j$, is $p^k p^{k-t} = p^{2k-t}$. When calculating this probability we are interested in counting the number of edges t that are shared between A_i and A_j. That is, between which vertices t edges are shared is not of interest. Between any two vertices u and v_i there are k edges and hence, the number of ways to determine the t shared edges is $\binom{k}{t}$. For any A_j, the two vertices u and v_i are fixed. From the $k-1$ other vertices on the path from u to v_i, t are shared with A_i. Thus, to complete A_j that share t edges with A_i, there are $\binom{n-2}{k-1-t}$ possible ways to add the remaining vertices. Thus, for a given value t, $\binom{k}{t}\binom{n-2}{k-1-t}$ determine how many sets A_j share precisely t edges with A_i. Thus, from Equation 12 we have

$$\begin{aligned}
\Delta &\leq \binom{n-2}{k-1} \sum_{t=1}^{k-1} \binom{k}{t} \binom{n-2}{k-1-t} p^{2k-t} \\
&\leq \frac{n^{k-1}}{(k-1)!} 2^k \sum_{t=1}^{k-1} \binom{n-2}{k-1-t} p^{2k-t} && \left(\text{by } \binom{n}{k} \leq \frac{n^k}{k!} \text{ and } \binom{n}{k} \leq 2^n\right) \\
&\leq \frac{n^{k-1}}{(k-1)!} 2^k \sum_{t=1}^{k-1} \frac{(n-2)^{k-1-t}}{(k-1-t)!} p^{2k-t} && \left(\text{by } \binom{n}{k} \leq \frac{n^k}{k!}\right) \\
&\leq \frac{n^{k-1}}{(k-1)!} 2^k \sum_{t=1}^{k-1} n^{k-1-t} p^{2k-t} \qquad (13)
\end{aligned}$$

We now calculate $n^{k-1-t}p^{2k-t}$.

$$\begin{aligned}
n^{k-1-t}p^{2k-t} &= \frac{n^{k-t}}{n}p^k p^{k-t} = \frac{n^{k-t}p^{k-t}}{n}p^k \\
&= \frac{n^{k-t}p^{k-t}}{n}\left(k\sqrt[k]{\frac{\log n}{n^{k-1}}}\right)^k = \frac{n^{k-t}p^{k-t}}{n}\,k^k\,\frac{\log n}{n^{k-1}} \\
&= \frac{n^{k-t}p^{k-t}}{n^k}\,k^k\,\log n = n^{-t}p^{k-t}k^k\log n \\
&= (np)^{-t}\left(p^k k^k \log n\right) = \left(nk\sqrt[k]{\frac{\log n}{n^{k-1}}}\right)^{-t}\left(p^k k^k \log n\right) \\
&= \left(n^{1-(k-1)/k}\,k\,\sqrt[k]{\log n}\right)^{-t}\left(p^k k^k \log n\right) \\
&= \left(n^{1/k}\,k\,\sqrt[k]{\log n}\right)^{-t}\left(p^k k^k \log n\right) = \frac{\left(p^k k^k \log n\right)}{\left(kn^{1/k}\,\sqrt[k]{\log n}\right)^t} \\
&\leq \frac{\left(p^k k^k \log n\right)}{kn^{1/k}\,\sqrt[k]{\log n}} \qquad (14)
\end{aligned}$$

From Equations 13 and 14 we have

$$\begin{aligned}
\Delta &\leq \frac{n^{k-1}}{(k-1)!}2^k\sum_{t=1}^{k-1}n^{k-1-t}p^{2k-t} \\
&\leq \frac{n^{k-1}}{(k-1)!}2^k\sum_{t=1}^{k-1}\frac{p^k k^k \log n}{kn^{1/k}\sqrt[k]{\log n}} \\
&\leq \frac{n^{k-1}}{(k-1)!}2^k\,k\,\frac{p^k k^k \log n}{kn^{1/k}\sqrt[k]{\log n}} \\
&\leq 2^k\frac{n^{k-1}}{(k-1)!}\left(k\sqrt[k]{\frac{\log n}{n^{k-1}}}\right)^k\frac{k^k \log n}{n^{1/k}\sqrt[k]{\log n}} \\
&\leq \frac{2^k}{(k-1)!}\,k^{2k}\,\frac{n^{k-1}\log^2 n}{n^{k-1}n^{1/k}\sqrt[k]{\log n}} \\
&\leq O(k)\,\frac{\log^2 n}{n^{1/k}\sqrt[k]{\log n}} \\
&\leq O(k)\,\frac{\log^2 n}{n^{1/k}}
\end{aligned}$$

Thus, $\Delta \to 0$ as $n \to \infty$. Since $\Delta < \mu$ by Janson's Inequality we have

$$\mathbb{P}[X_B = 0] = \mathbb{P}\left[\bigcap_{i=1}^{\binom{n-2}{k-1}} \overline{B}_i\right]$$
$$\leq e^{-\mu/2} \leq e^{-\frac{0.9k\log n}{2}}$$
$$\leq e^{-\frac{9}{20}k\log n}$$

Thus, the probability that a vertex u is not within distance-k from a dominator vertex v_i is given by

$$\mathbb{P}\left[v_i \notin N_k(u)\right] \leq \mathbb{P}\left[\bigcap_{i=1}^{\binom{n-2}{k-1}} \overline{B}_i\right]$$
$$\leq e^{-\frac{9}{20}k\log n}. \tag{15}$$

Let X_u be the random variable that denotes the number of non-dominators of u. We note that u may be a dominating vertex. Then

$$\mathbb{E}[X_u] \quad \leq \quad r\, e^{-\frac{9}{20}k\log n}.$$

By Markovs's Inequality we have $\mathbb{P}[X_u > 0] \leq \mathbb{E}[X_u] \leq r\, e^{-\frac{9}{20}k\log n}$. Thus,

$$\mathbb{P}[\text{fixed } u \text{ is bad}] \leq \mathbb{P}[X_u > 0] \leq r\, e^{-\frac{9}{20}k\log n}. \tag{16}$$

By Eq. 10 and Eq. 16 we have $\mathbb{E}[X] \leq n\, r\, e^{-\frac{9}{20}k\log n}$ and by Markov's Inequality it follows,

$$\mathbb{P}[X > 0] \leq \mathbb{E}[X] \leq n\, r\, e^{-\frac{9}{20}k\log n}. \tag{17}$$

From Eq. 17, we determine the value of $e^{-\frac{9}{20}k\log n}$ to be

$$e^{-\frac{9}{20}k\log n} \geq \left(e^{\log n}\right)^{-\frac{9}{20}k} = n^{-\frac{9}{20}k}.$$

Thus, we have

$$nre^{-\frac{9}{20}k\log n} \leq \frac{nr}{n^{\frac{9}{20}k}} \leq \frac{r}{n^{\frac{9}{20}k-1}}.$$

For $k \geq 3$, $\dfrac{r}{n^{\frac{9}{20}k-1}} \to 0$ as $n \to \infty$.

Therefore, $\mathbb{P}[X > 0] \to 0$ as $n \to \infty$. □

4.4 Summary

In this chapter we considered upper bounds on the total (k, r)-domination number to bound the number of sinks in WSNs. Theorem 4.2.1 gives an upper bound on the total (k, r)-domination number in graphs of girth at least $2k + 1$. We show that in a graph G of girth at least $2k + 1$, $\gamma^t_{(k,r)}(G) \leq \frac{2nr}{(d-1)^k} + nre^{-\frac{r}{4}}$, where $n = |V(G)|$ and d is the minimum degree.

Theorem 4.3.2 gives an upper bound on the total $(2, r)$-domination number in random graphs. For a fixed constant $c > 1$ and any positive integer r, in a random graph $G(n, p)$ with $p \geq c\sqrt{\frac{\log n}{n}}$, a.a.s. $\gamma^t_{(2,r)}(G(n, p)) = r + 1$. Theorem 4.3.5 generalizes this result for positive integers $k \geq 3$ and r. That is, for any positive integers $k \geq 3$ and r, in a random graph $G(n, p)$ with $p \geq k\sqrt[k]{\frac{\log n}{n^{k-1}}}$, a.a.s. $\gamma^t_{(k,r)}(G(n, p)) = r + 1$.

Chapter 5

Hexagonal Virtual Network based Beacon-less Flooding in MANETs

In this chapter, we consider the problem of efficiently flooding a data packet P in a wireless mobile ad hoc network. Flooding is an important primitive in MANETs. Due to mobile nodes and possible change of location information of nodes, it is of importance for a flooded packet P to be received by every node, but at the same time to limit the number of forwarding nodes. Thus, in this chapter we present a beacon-less flooding algorithm (HBLF), which is based on an overlayed hexagonal virtual network. An overlayed hexagonal virtual network allows us to depict a mobile network in a static manner. HBLF achieves full delivery even in the presence of holes in the network. We give further theoretic analysis of HBLF in regards to lower and upper bounds on the number of forwarders, dilation factor as well as the broadcast time of HBLF.

Before presenting the HBLF algorithm, we first present the network model in Section 5.1. The algorithm is presented in Section 5.2 and Section 5.3 presents the theoretic analysis.

5.1 Definitions and Network Model

We consider a wireless mobile ad hoc network modelled as a *unit disk graph*. Each node at a given time is aware of its position and the position of any node from which it receives a data packet, since this information can be stored in the header of the packet. In a similar manner each node is also aware of the position of the source node that generated the packet. However, nodes are mobile and over time the location information of nodes may change and as a result the topology of the network may change. Thus, to depict a mobile network in a static manner and to achieve small number of forwarders, we introduce a virtual layer of hexagon tiles over the network, where we limit at most one forwarding node per hexagon. Hence, over time with the possible change of the network topology the hexagonal virtual network stays the same.

At a given time each node belongs to a specific hexagon. In a hexagon, we assume that its left hand side boundary, starting from the top left apex of the hexagon up to the bottom right apex of the hexagon, belongs to the hexagon. Thus, only the top left apex and the two left lower apexes are considered to belong to the hexagon (see Figure 5.1.1).

Definition 5.1.1. *Two hexagons H_i and H_j are* adjacent *if H_i and H_j share one of their six sides.*

The size of each hexagon is chosen small enough so that when a node v in hexagon H_i forwards a message, all nodes in H_i and in hexagons adjacent to H_i will hear the message. The radius of each hexagon is denoted as r and the transmission range of each node as R. We let $R = 2\sqrt{7}r$ as shown in Figure 5.1.1. Given three hexagons in a row (see Figure 5.1.1), the longest distance spanning all three hexagons is $R = 2\sqrt{7}r$. Hence, all nodes within all three hexagons can communicate with each other.

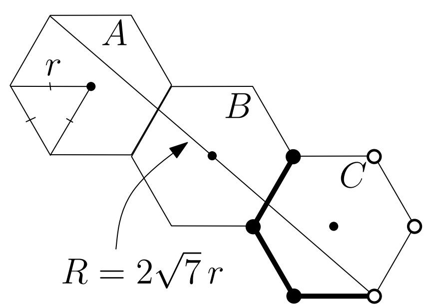


Figure 5.1.1: The transmission range of each node is R so that all nodes within hexagons A, B, and C can communicate with each other. Bold line illustrates the boundary that belongs to the hexagon C.

We assume one of the hexagons of the virtual layer is centred at $[0, 0]$ of the plane and all others are positioned accordingly. Note that a graph of hexagons is

3-colourable. Hence, we consider a virtual layer of hexagon tiles coloured one of three different colours: blue, yellow or pink. Given hexagons in a column, the colouring scheme from top to bottom is as follows. A blue hexagon always follows a pink hexagon and a yellow hexagon always follows a blue hexagon. By requiring that any two adjacent hexagons are of different colours, the above colouring scheme determines the colours of all other hexagons. It is easy to see that each node which knows its own position, the starting point of the hexagonal tiling and the colouring scheme of the virtual layer, it can calculate the position of its own hexagon and its colour in constant time.

Definition 5.1.2. *Given a hexagon* H_i*, the* inner neighbourhood *of* H_i*, denoted* $N_1(H_i)$*, consists of all six hexagons adjacent to* H_i*. The* outer neighbourhood *of* H_i*, denoted* $N_2(H_i)$*, consists of all hexagons adjacent to those in* $N_1(H_i)$*, but not in* $N_1(H_i)$*. A hexagon in* $N_1(H_i)$ *is referred to an* inner hexagon *and a hexagon in* $N_2(H_i)$ *is referred to an* outer hexagon *(see Figure 5.1.2).*

Definition 5.1.3. *The* coverage area *of a hexagon* H_i*, denoted* $C(H_i)$*, consists of* H_i *and all hexagons in* $N_1(H_i) \cup N_2(H_i)$ *(see Figure 5.1.2).*

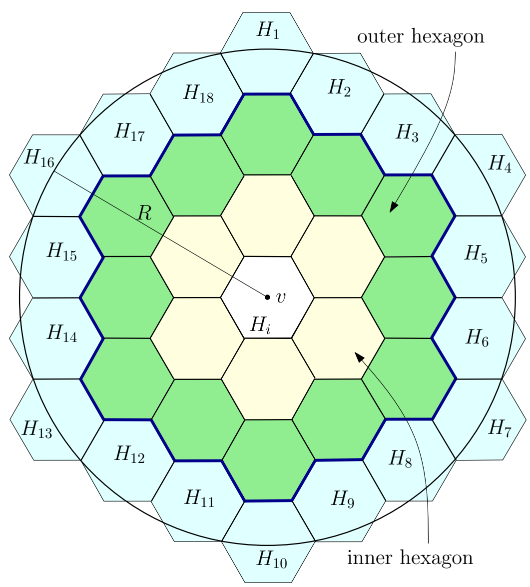


Figure 5.1.2: The circular disk of radius R is the transmission range of node v. Hexagons in $N_1(H_i)$ are light yellow and hexagons in $N_2(H_i)$ are shaded. $N_1(H_i) \cup N_2(H_i)$ make $C(H_i)$ shown by the bold boundary. Note that hexagons $H_1, \cdots, H_{18}$ that partially fall into the circular disk of v are not included in $C(H_i)$.

For easier readability, in the text that follows, a node u in a hexagon H_u is denoted as $u \in H_u$ and a node u in $C(H_u)$ is denoted as $u \in C(H_u)$. Similarly, any hexagon H_u in $C(H_v)$ is denoted as $H_u \subseteq C(H_v)$. In the text that follows a hexagon that contains a forwarding node is referred to a forwarding hexagon. Also, a node $u \in H_u$ that has received the packet P from a node $v \in H_v$ may be referred to as H_u has received P from H_v, or H_v covers H_u.

5.2 Beacon-less Flooding Algorithm

This section presents our beacon-less flooding algorithm HBLF based on a hexagonal virtual network in MANETs. In Section 5.2.1 we give an informal overview of the HBLF algorithm and present the algorithm in details in Algorithm 3. We first present some definitions that are needed.

Definition 5.2.1. *For any two hexagons H_u and H_v, if $H_u \subset C(H_v)$ or $H_v \subset C(H_u)$ then we say H_u and H_v are* neighbouring hexagons*.*

Definition 5.2.2. *If two hexagons H_i and H_j $(i \neq j)$ are of the same colour and H_i, H_j are neighbouring hexagons, then we say H_i and H_j are* separated by a single hop *and are called* bridged hexagons*, denoted (H_i, H_j).*

Definition 5.2.3. *Given a hexagon H_i, a* level 0 *of H_i, denoted $L_0(H_i)$, is the set of all hexagons adjacent and bridged with H_i.* Level k *of H_i, denoted $L_k(H_i)$, is the set of all hexagons that are bridged with the hexagons in $L_{k-1}(H_i)$, but not in $L_{k-1}(H_i)$ (see Figure 5.2.3).*

Definition 5.2.4. *On a given level i, the hexagons on the six corners of level i are called* corner hexagons*. The outer most hexagons of level i in between and of the same colour of corner hexagons are called* side hexagons *(see Figure 5.2.4).*

Definition 5.2.5. *From a given side hexagon H_s, the corresponding corner hexagon on the same side as H_s encountered first in a clockwise direction is called a* left corner hexagon *(LCH) and the corresponding corner hexagon on the same side as H_s encountered last is called a* right corner hexagon *(RCH) (see Figure 5.2.4).*

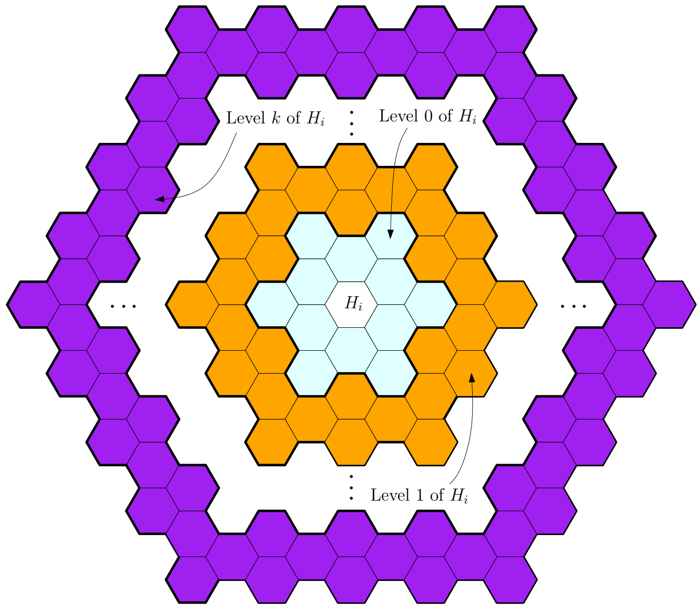


Figure 5.2.3: $L_0(H_i)$ is designated by hexagons in the lightest shade surrounding H_i. $L_1(H_i), \cdots, L_k(H_i)$ are designated by hexagons in different shades from light to dark.

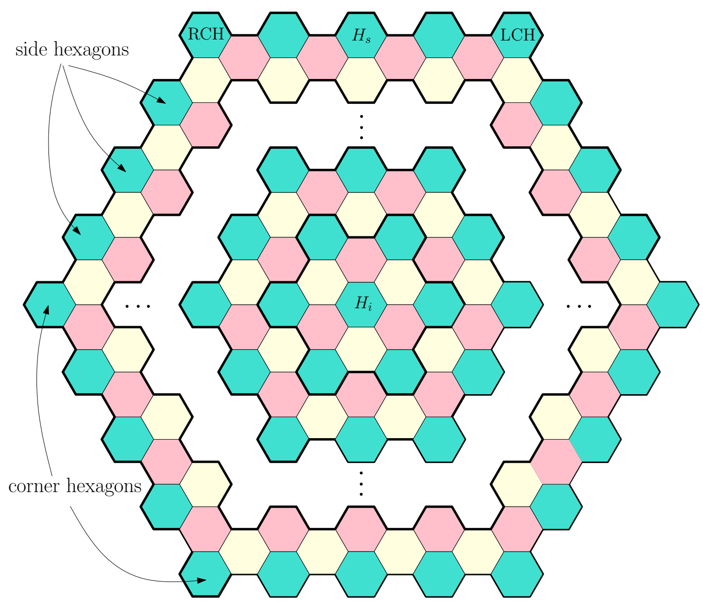


Figure 5.2.4: On a given level k of hexagon H_i, corner and side hexagons are of the same colour as H_i. LCH is the left corner hexagon of the side hexagon H_s and RCH is the right corner hexagon of side hexagon H_s.

Let h denote the number of side hexagons on a single side of a given level $i \geq 0$.

Definition 5.2.6. *A hexagon H_i among the h side hexagons is a* Type A hexagon, *if H_i is even number of hops from LCH when h is odd; or H_i is odd number of hops from LCH when h is even.*

Definition 5.2.7. *A hexagon H_i among the h side hexagons is a* Type B hexagon*, if H_i is odd number of hops from LCH when h is odd; or H_i is even number of hops from LCH when h is even.*

The reason for this multiple categorization is to reduce the number of forwarding nodes in the network. This is further discussed in details in Section 5.2.2.

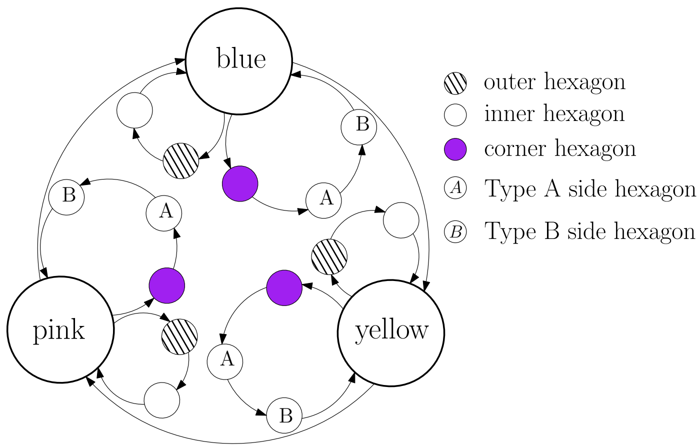


Figure 5.2.5: Cyclic order of colour priorities.

Let the source node be in hexagon H_o. The HBLF algorithm presented in Algorithm 3 makes use of the overlay network of hexagon tiles. That is, it uses an ordering of the colours, which depends on the colour of H_o. The priority of colours is cyclic, where the highest priority is always given to the colour of H_o. Without loss of generality, assume the colour of H_o is blue. Thus, the highest priority is given to hexagons of colour blue. The second priority we assume is given to hexagons of colour yellow and the third priority is given to hexagons of colour pink. Note that since H_o is of colour blue, it follows that on a given level i constructed around H_i the corner and side hexagons are of colour blue. Thus, blue hexagons are further prioritized into corner hexagons, Type A side hexagons and Type B side hexagons. Corner hexagons have higher priority than Types A and B side hexagons, and Type A side hexagons have higher priority than Type B side hexagons. Yellow and pink hexagons are further prioritized into outer and inner hexagons. Outer hexagons have higher priority than inner hexagons. As mentioned previously, the priority of colours is cyclic. That is, if the originator falls within a yellow hexagon, the first priority will be given to yellow hexagons, the second priority to pink hexagons and the third to blue hexagons and so on. The cyclic order of colours is demonstrated in Figure 5.2.5.

5.2.1 Overview of HBLF

Let S denote the source node in hexagon H_o, k denote the level number to be covered, and v_p denote the forwarding node that was an immediate predecessor to the current forwarding node. Let H_p denote the hexagon that contains the node v_p. Throughout this work the following notation is used.

- *ID(v)* = ID of a node v.
- *ID(P)* = ID of a data packet P.
- *location(H)* = location of a hexagon H defined by its centre.
- *colour(H)* = colour of a hexagon H.

The algorithm starts as follows. An originating node S in hexagon H_o sends a data packet P. The header of P contains the following information:

$$\begin{bmatrix} ID(P),\ ID(S),\ location(H_o),\ colour(H_o),\ level\ \mathrm{k} \\ ID(v_p),\ location(H_p),\ colour(H_p) \end{bmatrix}.$$

Initially S sends P with the following header

$$\begin{bmatrix} ID(P),\ ID(S),\ location(H_o),\ colour(H_o),\ k = 0,\ ID(v_p) = ID(S), \\ location(H_p) = location(H_o),\ colour(H_p) = colour(H_o) \end{bmatrix}.$$

The idea of the algorithm is to progress the forwarding of P in a controlled manner. Initially S sets $k = 0$ and broadcasts P, where all hexagons in $C(H_o)$ receive P, and from the value of k can determine the next level to be covered is level 1. Forwarding nodes from $C(H_o)$ set $k = 1$ in the header of P and forward P in the aim to cover level 1.

A node v in a hexagon H_v can deduce the following information.

1. v knows if a node $v' \in H_v$ has forwarded P, since the radius of any hexagon is smaller than the transmission range of any node.

2. v knows the colours of all hexagons in $C(H_v)$, since v knows the colour of H_v and the colouring scheme of the overlayed hexagonal virtual network.

3. v that receives P from a node $u \in H_u$, where $H_v \neq H_u$, can deduce the colours of all hexagons in $C(H_u)$, since v knows the colours of H_v, H_u and the colouring scheme of the overlayed hexagonal virtual network.

4. v that receives P knows which level H_v is on with respect to H_o since the location of H_o is stored in the header of P.

5. If H_v is of colour blue then v knows (a) the level which H_v is on with respect to H_o and hence, also knows h; (b) if H_v is a corner, Type A or Type B hexagon, and (c) the hop distance between H_v and LCH (if H_v is not LCH).

Thus, when a node $v \in H_v$ receives P from a node u, v will act differently based on the local information available to it. If $u \in H_v$, then v will ignore P. However, if $u \notin H_v$, then without loss of generality, let $u \in H_u$. Then, if $H_v \not\subset C(H_u)$, v will ignore P. Otherwise, where $H_v \subset C(H_u)$, v will start a contention timer $t_h + t_c$.

The t_h component is set to determine which hexagons amongst $C(H_u)$ contain a forwarding node. The t_c component is set to determine a forwarding node within a given hexagon H_i. When all nodes in a given hexagon H_i start respective contention timers $t_h + t_c$ upon receival of P from a node $v \in H_j$, for all $j \geq 0$ and $j \neq i$, the t_h component for all nodes $u \in H_i$ is the same. The value of t_h for all nodes $u \in H_i$ is determined by the colour of $H_i \subset C(H_j)$. As mentioned earlier, one colour has priority over another and the priorities of colours depend on the colour of H_o. The higher the priority of the colour of $H_i \subset C(H_j)$, the smaller will be the value of t_h. If H_i is blue and a corner hexagon, then the value of t_h will be smaller than if H_i was either a blue Type A/B hexagon or a yellow/pink hexagon. Similarly, if H_i is an outer yellow hexagon in $C(H_j)$, then the value of t_h for all nodes in H_i will be smaller than if H_i was an inner yellow hexagon or a pink hexagon in $C(H_j)$. The value of t_h for all nodes in a hexagon H_i can be set similarly, depending on the colour priority of H_i.

For a given hexagon $H_i \subset C(H_j)$ that has received a message from H_j for $i \neq j$, we define $t_h = \frac{m}{2} q$, where $q \in \mathbb{R}^+$ and

$$\begin{aligned}
m &= \tfrac{2}{3} && \text{if } H_i \text{ is a corner blue hexagon} \\
m &= \tfrac{4}{3} && \text{if } H_i \text{ is a Type A blue hexagon} \\
m &= 2 && \text{if } H_i \text{ is a Type B blue hexagon} \\
m &= 3 && \text{if } H_i \text{ is an outer yellow hexagon} \\
m &= 4 && \text{if } H_i \text{ is an inner yellow hexagon} \\
m &= 5 && \text{if } H_i \text{ is an outer pink hexagon} \\
m &= 6 && \text{if } H_i \text{ is an inner pink hexagon.}
\end{aligned}$$

The value of t_c can be chosen many ways. We choose the value of t_c of each node $v \in H_i$ to be determined by the distance between the centre of H_i and v, denoted as $d(v, C_i)$, where C_i is the centre of H_i. Define $t_c = \alpha_i \, d(v, C_i) + \dfrac{m}{2} q$, where α_i is a random number in the interval $[0, 1]$. The closer the distance between node $v \in H_i$ and C_i, the smaller will be the value of t_c for node v. Thus, from the definitions of t_h and t_c, $t_h + t_c = \alpha_i \, d(v, C_i) + m \, q$.

Upon the expiration of $t_h + t_c$, if H_v is a corner or a Type A blue hexagon, then v will forward P if it has not received P from any node $v' \in H_v$. Otherwise, if H_v is a blue hexagon of Type B, or a yellow/pink hexagon and v covers hexagons on level k not already covered by hexagons in $C(H_v)$, then v will forward P. There may arise a case when $v \in H_v$ receives P from $u \in H_u$ such that $H_v \subset C(H_u)$ and upon the expiration of $t_h + t_c$ v does not cover any additional hexagons on level k, but may cover hexagons in $C(H_v)$ not yet covered on levels greater than k or less than k. In this case, v will start a contention timer $t_h + t_c$ for the second time. Upon the expiration of this second timer $t_h + t_c$, if v still covers any hexagons in $C(H_v)$ not already covered, then v will forward P.

5.2.2 Motivation of Blue Hexagon Categorization

We assumed previously that the source node is in the hexagon H_o and H_o is of colour blue. The first colour priority is given to blue hexagons and hence, blue hexagons are further categorized into corner, Type A and Type B blue hexagons. From Lemma 5.3.1 in Section 5.3 it can be seen that if there is at least one node in every blue hexagon, then the corner and Type A blue hexagons cover the entire network and Type B blue hexagons do not need to forward P. If the categorization of blue hexagons

Algorithm 3 HBLF

Input: A network overlayed with hexagon tiles, source node $S \in H_o$, $location(H_o)$, $colour(H_o) = blue$.
Output: Every node in the network receives the message.

S sends P with header $\big[ID(P), ID(v_p) = ID(S), location(H_p) = location(H_o), colour(H_p) = colour(H_o), k = 0\big]$.
▷ ***Case 1:*** *Node $v \in H_v$ has received P from node $u \in H_v$ to cover level k.*
if v has a contention timer running for P **then**
    v cancels its timer and ignores P.
end if
▷ ***Case 2:*** *Node $v \in H_v$ has received P from node $u \in H_u$ to cover level k.*
if $H_v \not\subset C(H_u)$ **then**
    Node v ignores the data packet P.
else if v or any $v' \in H_v$ has already forwarded P **then**
    v ignores the data packet P.
else
    Node v starts a contention timer $t_h + t_c$.
    ▷ *Upon the expiration of $t_h + t_c$*
    if $colour(H_v)$ is blue **then**
        Blue_Fwd(v, P)
    end if
    if $colour(H_v)$ is yellow or pink **then**
        NonBlue_Fwd(v, P)
    end if
end if

was not present, then all blue hexagons will forward P and cover the entire network. Thus, the reason for this categorization is to reduce the number of forwarding nodes in the network. However, note that the disadvantage of this improvement in the number of forwarding nodes is that it increases the delay of the network. Thus, if we want improvement in the number of forwarding nodes, we use the categorization of blue hexagons into corner, Types A and B hexagons. Otherwise, to have improvement in the overall delay, we do not use any categorization of the blue hexagons at all and have all blue hexagons forward P.

Algorithm 4 BLUE_FWD(v, P)

Let $v \in H_v$
if v has not heard P from any other $w \in H_v$ **then**
  if $\big(H_v$ is a corner or Type A hexagon$\big)$ **or** $\big(H_v$ is a Type B hexagon and v covers additional hexagons on level k not yet covered$\big)$ **then**
    UPDATE(v, P)
    v forwards the data packet P
  else if H_v is a Type B hexagon and v covers additional hexagons in $C(H_v)$ not on level **then**
    v starts a contention timer $t_h + t_c$
    if upon the expiration of $t_h + t_c$ v has not heard P from any other $w \in H_v$ and v covers additional hexagons in $C(H_v)$ **then**
      UPDATE(v, P)
      v forwards the data packet P
    end if
  end if
end if

Algorithm 5 UPDATE(v, P)

v sets $k = k + 1$ in header of P
v sets $ID(v_p) = ID(v)$ in header of P
v sets $location(H_p) = location(H_v)$ in header of P
v sets $colour(H_p) = colour(H_v)$ in header of P

Algorithm 6 NONBLUE_FWD(v, P)

Let $v \in H_v$
if v has not heard P from any other $w \in H_v$ and v covers any additional hexagons on level k not yet covered **then**
  UPDATE(v, P)
  v forwards the data packet P
else if v covers additional hexagons in $C(H_v)$ not on level k **then**
  v starts a contention timer $t_h + t_c$.
  if upon the expiration of $t_h + t_c$ v has not heard P from any other $w \in H_v$ and v covers additional hexagons in $C(H_v)$ **then**
    UPDATE(v, P)
    v forwards the data packet P
  end if
end if

5.3 Theoretical Analysis

In this section we analyze the performance of HBLF. In a network data packets can collide with one another when being sent on a shared medium. In our analysis we consider a collision-free network, which is connected (i.e. there is a path from any node u to a node v). The next results that follow are concerning the full delivery of a data packet P in a network, where there are no holes as well as in networks that may contain holes. By *hole* we mean a region that does not contain any nodes, but may contain nodes around its perimeter. We also consider the number of rounds it takes for the algorithm to terminate. We define the term *round* as the time interval $(0, \alpha_i d(v, C_i) + 6q]$ for $q \in \mathbb{R}$, where this interval is bounded by the worst possible value of the contention timer $t_h + t_c$ described in the previous section. Section 5.3.1 presents a lower bound as well as upper bounds in networks with or without holes. The results in Section 5.3.2 are regarding the dilation factor of the shortest hexagonal path by HBLF, between the originator s and a node v, to the shortest path of the network between s and v. Section 5.3.2 also presents results on the broadcast time of HBLF.

Lemma 5.3.1. *Let the originator in the network be in a blue hexagon tile. If all blue hexagon tiles are not empty, then by Algorithm 3 every node in the network receives the message and Algorithm 3 terminates in at most $k + 2$ rounds, where k is the outermost level number with respect to the originating hexagon.*

Proof. The originator is in a blue hexagon H_o and hence, blue hexagons have highest priority. Each level is in the shape of a hexagon and all six sides are symmetric. By assumption there is at least one node in each blue hexagon and thus, it is enough to show that every other blue hexagon that forwards the message on level $i - 1$ cover all hexagons on level i.

The construction of each level is such that every yellow and pink hexagon on level i is within the coverage area of three blue hexagons on level $i - 1$. Also, every corner blue hexagon on level i is bridged with a corner blue hexagon on level $i - 1$, and every side blue hexagon on level i is bridged with two blue hexagons on level $i - 1$. Hence, by Algorithm 3 when all corner blue hexagons and every other side blue hexagon on level $i - 1$ forward the message, all hexagons on level i receive the message. Thus, blue hexagons on levels $(i - 1) \geq 0$ cover all hexagons on level i.

After Algorithm 3 terminates k is the outermost level number with respect to H_o. Note that when Algorithm 3 starts, the first level with respect to H_o is numbered 0. Thus, in the first round of HBLF, H_o forwards the message. In the second round, blue hexagons in $C(H_o)$, which are on level 0 still, forward the message. Thus, after Algorithm 3 terminates the length of the longest path returned by HBLF from H_o to the farthest blue hexagon on level k is $k + 1$ hops. Thus, Algorithm 3 terminates in at most $k + 2$ rounds. □

Lemma 5.3.2. *Let the originator in the network be in a blue hexagon tile. If all other blue hexagon tiles are empty and all yellow hexagon tiles are not empty, then by Algorithm 3 every node in the network receives the message and Algorithm 3 terminates in at most $k + 3$ rounds, where k is the outermost level number with respect to the originating hexagon.*

Proof. Let the originator be in the hexagon H_o. By assumption H_o is of colour blue and hence, blue hexagons have the highest priority. However, since all blue hexagons, except H_o, are empty the next highest priority is that of yellow hexagons. Each level is in the shape of a hexagon and all six sides are symmetric. By assumption there is at least one node in each yellow hexagon. Since, blue hexagons are empty and yellow hexagons have the highest priority, when their timers expire, they forward the message.

It is enough to show that all hexagons on level i are collectively covered by yellow hexagons on levels $i-1$ and i. Note that the construction of each level is such that a blue and pink hexagon on level i has at least 2 yellow hexagons in their coverage area on levels i and $i - 1$. Also note that for each yellow hexagon H_y on level $i - 1$, there is a blue and/or pink hexagon that is reachable only from H_y and its adjacent pink hexagons. Since yellow hexagons have higher priority, H_y will forward the message first before the pink hexagons' timers expire. This is the case for all other yellow hexagons. As a result all yellow hexagons will forward the message. Since all yellow hexagons are bridged together in the hexagonal graph and each yellow hexagon is adjacent to three pink and three blue hexagons due to the colouring scheme of the hexagonal graph, then when yellow hexagons on levels $i-1$ and i forward the message every pink and blue hexagon on level i will receive the message. Thus, every node in the network will receive the message.

After Algorithm 3 terminates k is the outermost level number with respect to

H_o. From Lemma 5.3.1 when the blue hexagons forward the message Algorithm 3 terminates in at most $k+2$ rounds. Since yellow hexagons on level i are adjacent to blue hexagons on level i, then there are yellow hexagons on level k that will forward the message in the $(k+2)$-th round. Note that the outermost hexagons on level k (i.e. hexagons adjacent to the boundary line of level k) are blue hexagons. The colouring scheme of the hexagonal graph and the construction of each level is such that in the three alternating sides of the total six sides of level k, yellow hexagons are adjacent to the boundary of level k; and on the remaining three alternating sides of the total six sides of level k, pink hexagons are adjacent to the boundary of level k. Hence, the pink hexagons adjacent to the boundary of level k potentially cover hexagons outside of the network region that could not be covered by their adjacent yellow hexagons on level k. Thus, the pink hexagons adjacent to the boundary of level k will forward the message after the yellow hexagons on level k have forwarded the message in the $(k+2)$-th round. Therefore, Algorithm 3 will terminate in at most $k+3$ rounds. □

Lemma 5.3.3. *Let the originator in the network be in a blue hexagon tile. If all other blue and yellow hexagons are empty and all pink hexagons are not empty, then by Algorithm 3 every node in the network receives the message and Algorithm 3 terminates in at most $k+2$ rounds, where k is the outermost level number with respect to the originating hexagon.*

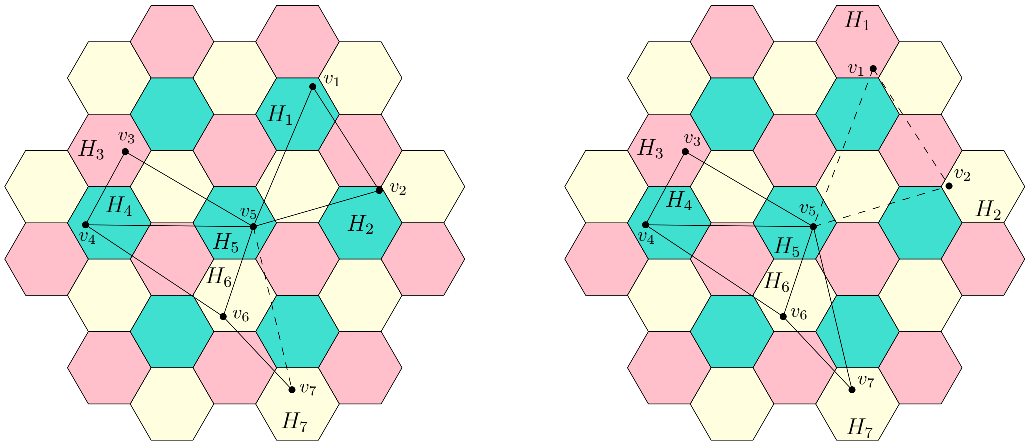
The proof of Lemma 5.3.3 is similar to that of Lemma 5.3.2.

We assume that the network is connected, i.e. there is a path from any node u to node v. To achieve full delivery by Algorithm 3 in presence of holes in the network that are larger than a hexagon, we must make a stronger assumption that the network is *hexagon connected* as defined below (see Figure 5.3.6).

Definition 5.3.1. *Let $P_H = H_1, H_2, \cdots, H_n$ be a sequence of hexagons such that each H_i for $1 \leq i \leq n$ is non-empty and $H_{i+1} \subset C(H_i)$ for $1 \leq i \leq n-1$, then we say that P_H is a* hexagonal path *from H_1 to H_n.*

Definition 5.3.2. *A network is* hexagon connected *if for any two nodes $u \in H_u$ and $v \in H_v$ there is a hexagonal path from H_u to H_v (see Figure 5.3.6).*

Theorem 5.3.1. *In a wireless mobile ad-hoc network, where the network is hexagon connected, Algorithm 3 terminates and achieves full delivery.*



(a) The network is connected as well as hexagon connected since between any two noes there exists a path and a hexagonal path.

(b) The network is connected since there is a path between any two nodes, but it is not hexagon connected since there is no path from v_1 to any other node.

Figure 5.3.6: Dashed lines represent edges that are in the original graph but not in the hexagonal graph. Solid lines represent edges that are both in the original graph and in the hexagonal graph.

Proof. Without loss of generality assume that the originator is in a blue hexagon tile. By the algorithm only one node from each hexagon forwards the message and each hexagon broadcasts the message at most once. Thus, broadcasting will eventually terminate.

By contradiction, assume after the termination of the algorithm there is at least one node v that has not received the message. Let v be in a hexagon denoted H_v. H_v can be of colour blue, yellow or pink. Let H_s be the hexagon that has started the broadcast. The network is hexagon connected and thus, there exists a hexagonal path from H_s to H_v. Clearly, there are two neighbouring hexagons H_A and H_B such that H_B has not received the message and H_A has received the message, but has not forwarded it. Since $H_A \subset C(H_B)$ has not forwarded the message, then H_A is either a blue side hexagon of Type B or of colour yellow or pink. H_A cannot be a corner blue hexagon or a side blue hexagon of Type A, since corner blue hexagons and side blue hexagons of Type A broadcast the message immediately upon the expiration of their timers.

H_A, upon receiving the message starts a contention timer $t_h + t_c$. From the packet header information and the network colouring scheme H_A can deduce if there are any hexagons in $C(H_A)$ that have not been covered. Upon the expiration of $t_h + t_c$, H_A may restart its contention timer if there are hexagons in $C(H_A)$ not covered. Since H_A has not broadcast the message after the expiration of any of its contention timers, then there must be hexagons $\{H_1, H_2, \cdots, H_j\} \subset (C(H_A) \cap C(H_B))$, for $j = 1, 2, \cdots, l$ and integer l, that have forwarded the message and H_A does not cover any more additional hexagons in $C(H_A)$. Thus, H_B must be reachable from $\{H_1, H_2, \cdots, H_j\}$, otherwise, H_A would broadcast the message. This contradicts our assumption that H_B has not received the message. □

5.3.1 Lower and Upper Bounds

We now study the number of forwarders necessary to flood the network by the HBLF algorithm. The analysis that follows determines lower and upper bounds on the number of forwarding nodes, denoted β. Since we consider an overlay network of hexagons, it is natural to give an upper bound on β on a hexagonal shape network. Thus, in all the results that follow we assume that the wireless ad hoc network, denoted as G, is of hexagonal shape and $\mathcal{H}(G)$ denotes the network G with the overlay hexagonal network.

Definition 5.3.3. *Consider a network G, where $\mathcal{H}(G)$ is the overlay hexagonal network of G. Let H_c be the central hexagon of $\mathcal{H}(G)$ and let H_b be the outermost corner hexagon of $\mathcal{H}(G)$. Let d denote the Euclidean distance between the centres of H_c and H_b. Then the* radius of $\mathcal{H}(G)$ *is given by $\frac{d}{3r}$, where r is the radius of each hexagon in $\mathcal{H}(G)$ and $3r$ is the Euclidean distance between the centres of two bridged hexagons.*

Theorem 5.3.2. *Let G denote the wireless mobile ad hoc network of hexagonal shape, where k is the radius of $\mathcal{H}(G)$. The number of forwarders β in G is at least $\beta \geq \left\lceil \frac{9\sqrt{3}\left(9k^2+3k+1\right)-168\pi}{28\left(2\pi+3\sqrt{3}\right)} \right\rceil + 1$.*

Proof. To determine a lower bound on the number of forwarders in G, we give an area argument. That is, dividing the area spanned by G by the area covered by a transmitting node, we will obtain the minimum number of forwarders needed to flood the network.

Each node has the same transmission range. Hence, the circular region that is covered independently by each node is the same and is given by $A_R = \pi R^2$. Let the originator be denoted as v. The area covered by v is $A_v = A_R = \pi R^2$. After the originator sends the data packet P, some or all neighbours of v must continue to forward P. At this step the best possible coverage by any neighbour of v will occur if it is on the boundary of the circular transmission disk of v (see Figure 5.3.7). Hence, any neighbour u of v on the boundary of the transmission disk of v will cover an area that is less than A_v, since the transmission disks of u and v intersect.

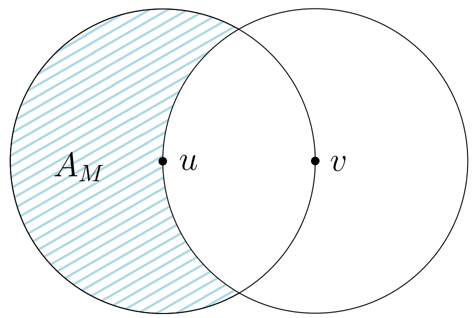


Figure 5.3.7: Node v is the originator. The most area that can be covered by a neighbouring node u of v is denoted as A_M.

The area covered by any neighbour u of v is in a moon shape and is denoted A_M (see Figure 5.3.7). The originator covers an area A_v and every other node covers at most an area A_M. Thus, knowing the area spanned by G, denoted A_G, a lower bound on the number of forwarders can be given by

$$\left\lceil \frac{(A_G - A_v)}{A_M} \right\rceil + 1. \tag{1}$$

We first calculate the area A_G, where the radius of $\mathcal{H}(G)$ is k. Since the radius of $\mathcal{H}(G)$ is k, then from the centre of $\mathcal{H}(G)$ the maximum level of $\mathcal{H}(G)$ is $k-1$. On a given level i, the number of blue hexagons on each of the six sides of $\mathcal{H}(G)$ is $i+2$. Hence, there are $6(i+2) - 6 = 6(i+1)$ blue hexagons on level i, since the corner hexagons are shared by consecutive sides. On each of the six sides of level i, there are $2i+1$ non-blue hexagons. Hence, on each level i of $\mathcal{H}(G)$, there are $6(2i+1)$ non-blue hexagons. Thus, on each level $i \geq 0$ there are $6(i+1) + 6(2i+1) = 18i + 12$ hexagons (excluding the originating hexagon). Since the maximum level from the centre of $\mathcal{H}(G)$ is $k-1$, then the number of hexagon tiles in $\mathcal{H}(G)$ is given by

$$\begin{aligned}
&1 + (18 \cdot 0 + 12) + (18 \cdot 1 + 12) + \cdots + (18i + 12) + \cdots + (18k + 12) \\
&= 1 + 12k + 18\Big[1 + 2 + \cdots + (k-1)\Big] \\
&= 1 + 12k + 18\frac{(k-1)(k)}{2} = 1 + 12k + 9k(k-1) \\
&= 9k^2 + 3k + 1. \qquad (2)
\end{aligned}$$

The area of each hexagon tile in the network with radius r is given by

$$A_{SH} = \frac{3\sqrt{3}}{2} r^2. \qquad (3)$$

Thus, from Equations 2, 3 and $r = \dfrac{R}{2\sqrt{7}}$ we have

$$\begin{aligned}
A_G &= \frac{3\sqrt{3}}{2} r^2 \Big[9k^2 + 3k + 1\Big] \\
&= \frac{3\sqrt{3}}{56} R^2 \Big[9k^2 + 3k + 1\Big]. \qquad (4)
\end{aligned}$$

To calculate A_M we use the areas shown in Figure 5.3.8. The area A_s denotes the area of the four segments made by the points xy, xw, yz and zw. The area A_T denotes the area of the two triangles made by the points xyz and xzw. The area A_c denotes the area of the sector made by the points xyw. Thus, the moon shape area is given by

$$A_M = \pi R^2 - (A_c + 2A_s). \qquad (5)$$

The area A_c, is given by $A_c = \frac{\theta}{2\pi}\pi R^2$, where $\theta = \angle yxw = 2\alpha$. Since, $\triangle xzw$ is an equilateral triangle, then $\alpha = \frac{\pi}{3}$. Thus,

$$A_c = \frac{2\pi}{3}\frac{1}{2\pi}\pi R^2 = \frac{\pi R^2}{3} \qquad (6)$$

The area $2A_s$, is given by

$$2A_s = A_c - 2A_T. \qquad (7)$$

The area $A_T = \frac{Rh}{2}$, where $h = R\sin\frac{\pi}{3} = \frac{\sqrt{3}}{2}R$. Thus,

$$2A_T = Rh = R\left(\frac{\sqrt{3}}{2}R\right) = \frac{\sqrt{3}}{2}R^2. \qquad (8)$$

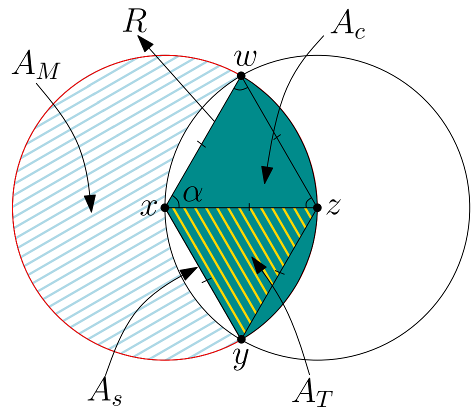


Figure 5.3.8: The area of intersection between the transmission disk of z and x is $A_c + 2A_s$.

From Eqs. 6, 8 and 7 we have

$$\begin{aligned} 2A_s &= A_c - 2A_T \\ &= \frac{\pi R^2}{3} - \frac{\sqrt{3}}{2}R^2 = R^2\left(\frac{\pi}{3} - \frac{\sqrt{3}}{2}\right) \\ &= R^2\left(\frac{2\pi - 3\sqrt{3}}{6}\right) \end{aligned} \tag{9}$$

From Eqs. 6, 9 and 5 we obtain the moon shape area to be

$$\begin{aligned} A_M &= \pi R^2 - (A_c + 2A_s) \\ &= \pi R^2 - \frac{\pi R^2}{3} - R^2\left(\frac{\pi}{3} - \frac{\sqrt{3}}{2}\right) \\ &= \frac{2\pi R^2}{3} - \frac{\pi R^2}{3} + \frac{\sqrt{3}R^2}{2} \\ &= \frac{\pi}{3}R^2 + \frac{\sqrt{3}}{2}R^2 = R^2\left(\frac{\pi}{3} + \frac{\sqrt{3}}{2}\right) \\ &= R^2\left(\frac{2\pi + 3\sqrt{3}}{6}\right) \end{aligned} \tag{10}$$

From Equations 1, 4 and 10, we have

$$\begin{aligned}
\beta &\geq \left\lceil \frac{A_G - A_v}{A_M} \right\rceil + 1 \\
&= \left\lceil \left(\frac{3\sqrt{3}}{56} R^2 \left(9k^2 + 3k + 1\right) - \pi R^2 \right) \frac{6}{R^2 \left(2\pi + 3\sqrt{3}\right)} \right\rceil + 1 \\
&= \left\lceil \frac{6}{\left(2\pi + 3\sqrt{3}\right)} \left(\frac{3\sqrt{3}}{56} \left(9k^2 + 3k + 1\right) - \pi \right) \right\rceil + 1 \\
&= \left\lceil 6 \left(\frac{3\sqrt{3}\left(9k^2 + 3k + 1\right) - 56\pi}{56\left(2\pi + 3\sqrt{3}\right)} \right) \right\rceil + 1 \\
&= \left\lceil \frac{9\sqrt{3}\left(9k^2 + 3k + 1\right) - 168\pi}{28\left(2\pi + 3\sqrt{3}\right)} \right\rceil + 1.
\end{aligned}$$

□

Lemma 5.3.4. *Let G denote the wireless mobile ad hoc network of hexagonal shape and $\mathcal{H}(G)$ is centred at a blue hexagon. Let k be the radius of $\mathcal{H}(G)$. If the originator is in a blue hexagon and all blue hexagons are not empty, then the number of forwarders β in G is at most $\beta \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$.*

Proof. Since the central hexagon of $\mathcal{H}(G)$ is blue and G is of hexagonal shape, then the outermost hexagons of $\mathcal{H}(G)$ must be blue to keep the symmetry of the hexagonal shape of G. Determining β in G is equivalent to determining the number of blue hexagons in $\mathcal{H}(G)$ that contain forwarding nodes. Since not all blue hexagons forward the packet P (i.e. Type B blue hexagons stay silent), we cannot simply calculate the number of blue hexagons defined by the region $\mathcal{H}(G)$.

Let H_o denote the originating hexagon and hence, it is of colour blue. Note that H_o can be anywhere in $\mathcal{H}(G)$. Let H_c denote the central hexagon of $\mathcal{H}(G)$. Since by HBLF every other blue hexagon forwards P, depending on where the algorithm starts in $\mathcal{H}(G)$, the number of forwarders may vary. Thus, we must consider the shift of H_o from H_c.

There are two possible shifts: a shift on one of the six axis, denoted s_a and an angular vertical shift, denoted s_v. Figure 5.3.9 depicts the two possible shifts of H_o from H_c in a hexagonal shape network G. Let the total shift be denoted by $s = s_a + s_v$. By the HBLF algorithm, the construction of each level is centred at H_o. Thus, to calculate the number of forwarding hexagons in $\mathcal{H}(G)$, we must calculate the most number of forwarding hexagons in each shifted sector centred at H_o (see Figure 5.3.9).

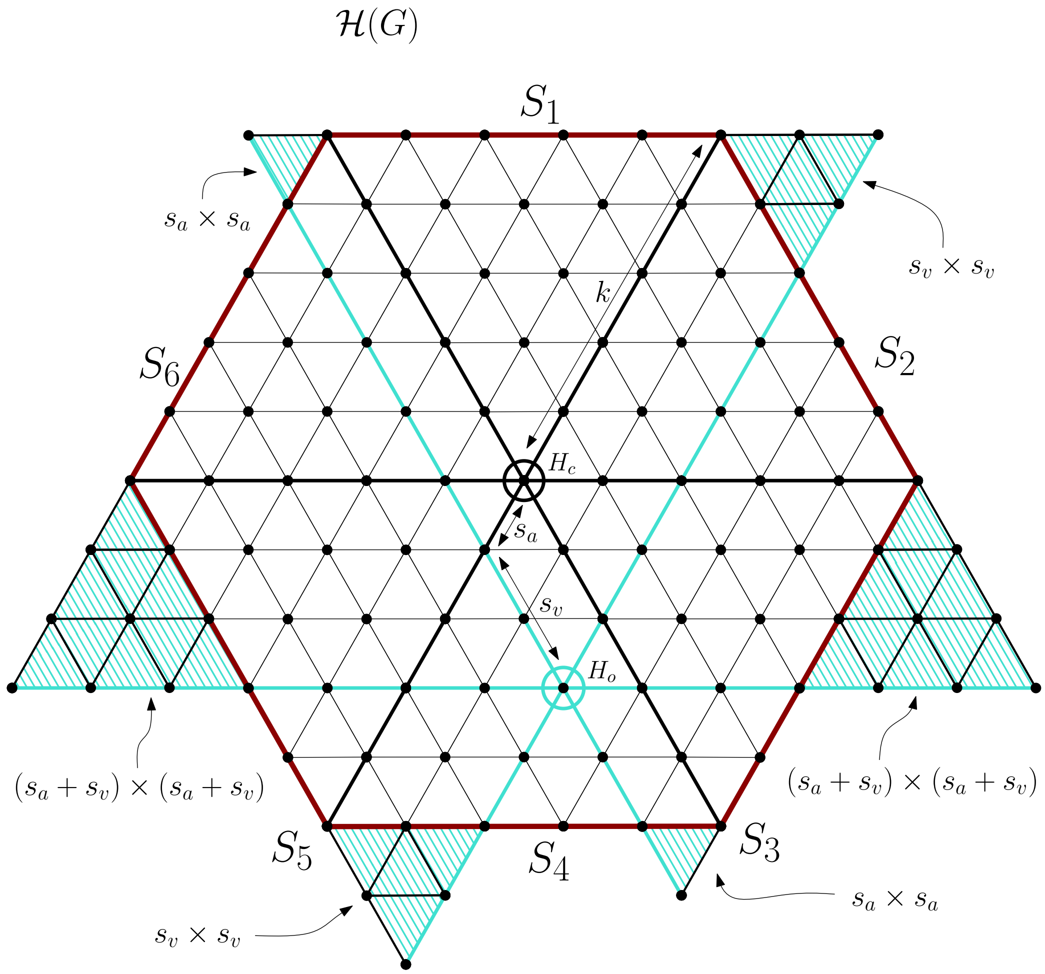


Figure 5.3.9: Each node represents a blue hexagon in $\mathcal{H}(G)$. The 6 black bold lines are the six axis of $\mathcal{H}(G)$ centred at H_c. The 6 blue bold lines are the six shifted axis centred at H_o.

A shifted sector can cover an area that is outside of the boundary region of $\mathcal{H}(G)$. For example, sector S_1 in Figure 5.3.9 contains a triangular region with dimensions $s_a \times s_a$ on the left and a triangular region with dimensions $s_v \times s_v$ on the right that are not in $\mathcal{H}(G)$. Thus, when calculating number of forwarding nodes in S_1, the number of forwarding hexagons that fall within $s_a \times s_a$ and $s_v \times s_v$ must be subtracted. However, note that those hexagons that are on the base of $s_a \times s_a$ and on the base of $s_v \times s_v$ share a boundary with $\mathcal{H}(G)$, and hence, may be forwarding hexagons. Thus, these hexagons should not be subtracted.

The dimensions of each of the shifted sectors and the additional triangular regions not in $\mathcal{H}(G)$ (labelled A.T.R.) are as follows.

$S1$: $(k - s_a + 2s_a + s_v) \times (k - s_a + 2s_a + s_v) \equiv (k + s_a + s_v) \times (k + s_a + s_v)$
A.T.R.: $s_a \times s_a$; $s_v \times s_v$

$S2$: $(k - s_v + s_a + s_v) \times (k - s_v + s_a + s_v) \equiv (k + s_a) \times (k + s_a)$
A.T.R.: $(s_a + s_v) \times (s_a + s_v)$

$S3$: $(k - s_a - s_v + s_a) \times (k - s_a - s_v + s_a) \equiv (k - s_v) \times (k - s_v)$
A.T.R.: $s_a \times s_a$

$S4$: $(k - s_a - s_v) \times (k - s_a - s_v)$

$S5$: $(k - s_a - s_v + s_v) \times (k - s_a - s_v + s_v) \equiv (k - s_a) \times (k - s_a)$
A.T.R.: $s_v \times s_v$

$S6$: $(k - s_a + s_a + s_v) \times (k - s_a + s_a + s_v) \equiv (k + s_v) \times (k + s_v)$
A.T.R.: $(s_a + s_v) \times (s_a + s_v)$

On each level $i = k-1$, there are $i+2$ blue hexagons and hence, by HBLF $\left\lfloor \frac{i+2}{2} \right\rfloor + 1$ forwarding blue hexagons. Thus, the number of forwarding hexagons in a sector of dimension $(k+s) \times (k+s)$ excluding the originator (i.e. the hexagon on the apex of the sector) is given by

$$T_S = \left(\left\lfloor \frac{0+2}{2} \right\rfloor + 1 \right) + \left(\left\lfloor \frac{1+2}{2} \right\rfloor + 1 \right) + \cdots + \left(\left\lfloor \frac{k+s-1+2}{2} \right\rfloor + 1 \right)$$
$$= k + s + 1 + \left(\left\lfloor \frac{2}{2} \right\rfloor + \left\lfloor \frac{3}{2} \right\rfloor + \cdots + \left\lfloor \frac{k+s+1}{2} \right\rfloor \right)$$

<u>If $k+s$ is even</u>

$$T_S = (k+s+1) + 1 + 1 + 2 + 2 + \cdots + \frac{k+s}{2} + \frac{k+s}{2}$$
$$= (k+s+1) + 2\left(1 + 2 + \cdots + \frac{k+s}{2}\right) = (k+s+1) + 2\frac{(k+s)}{2}\left(\frac{k+s}{2} + 1\right)\frac{1}{2}$$
$$= \frac{1}{4}k^2 + \frac{1}{2}ks + \frac{3}{2}k + \frac{1}{4}s^2 + \frac{3}{2}s + 1$$

If $k+s$ is odd

$$
\begin{aligned}
T_S &= (k+s+1)+1+1+2+2+\cdots+\frac{k+s-1}{2}+\frac{k+s-1}{2}+\frac{k+s+1}{2} \\
&= (k+s+1)+2\left(1+2+\cdots+\frac{k+s-1}{2}\right)+\frac{k+s+1}{2} \\
&= (k+s+1)+2\frac{(k+s-1)}{2}\left(\frac{k+s-1}{2}+1\right)\frac{1}{2}+\frac{k+s+1}{2} \\
&= (k+s+1)+\frac{(k+s-1)}{2}\frac{(k+s+1)}{2}+\frac{(k+s+1)}{2} \\
&= (k+s+1)+\frac{(k+s+1)}{2}\frac{(k+s+1)}{2} = (k+s+1)+\frac{1}{4}(k+s+1)^2 \\
&= \frac{1}{4}k^2+\frac{1}{2}ks+\frac{3}{2}k+\frac{1}{4}s^2+\frac{3}{2}s+\frac{5}{4}
\end{aligned}
$$

In a $s \times s$ sector there are $(s+1)+s+\cdots+1 = \frac{(s+1)(s+2)}{2}$ hexagons. The $s+1$ hexagons on the base of the $s \times s$ sector that share a boundary with $\mathcal{H}(G)$, in the worst case will forward P. Thus, the number of hexagons not in $\mathcal{H}(G)$ defined by the $s \times s$ sector is given by $T_D = \frac{(s+1)(s+2)}{2} - (s+1) = \frac{1}{2}s^2 + \frac{1}{2}s$. Thus, the number of forwarding hexagons in a sector with dimensions $(k+s) \times (k+s)$ that are also in $\mathcal{H}(G)$, excluding H_o, is given by $T_{F_j} \leq T_{S_j} - \frac{1}{2}s^2 - \frac{1}{2}s + (s+1) = T_{S_j} - \frac{1}{2}s^2 + \frac{1}{2}s + 1$. Therefore, the number of forwarding hexagons in $\mathcal{H}(G)$ is the sum of all the T_{F_j} sums of each of the six shifted sectors. Note that this sum counts the hexagons that fall on the axis twice, hence, it must be subtracted. Now, we present this calculation. There are two cases to consider: (a) $k+s$ is even and (b) $k+s$ is odd.

(a) If $k+s$ is even $\implies T_S = \frac{1}{4}k^2 + \frac{1}{2}ks + \frac{3}{2}k + \frac{1}{4}s^2 + \frac{3}{2}s + 1$.

$S1 : (k+s_a+s_v) \times (k+s_a+s_v)$; A.T.R.: $s_a \times s_a$; $s_v \times s_v$.

$$
\begin{aligned}
T_{F_1} \leq{} & \frac{1}{4}k^2+\frac{1}{2}k(s_a+s_v)+\frac{3}{2}k+\frac{1}{4}(s_a+s_v)^2+\frac{3}{2}(s_a+s_v)+1-\frac{1}{2}s_a^2+\frac{1}{2}s_a+1 \\
& -\frac{1}{2}s_v^2+\frac{1}{2}s_v+1 \\
={} & \frac{1}{4}k^2+\frac{1}{2}ks_a+\frac{1}{2}ks_v+\frac{3}{2}k-\frac{1}{4}s_a^2-\frac{1}{4}s_v^2+\frac{1}{2}s_as_v2s_a+2s_v+3
\end{aligned}
$$

$S2 : (k+s_a) \times (k+s_a)$; A.T.R.: $(s_a+s_v) \times (s_a+s_v)$.

$$
\begin{aligned}
T_{F_2} &\leq \frac{1}{4}k^2+\frac{1}{2}ks_a+\frac{3}{2}k+\frac{1}{4}s_a^2+\frac{3}{2}s_a+1-\frac{1}{2}(s_a+s_v)^2+\frac{1}{2}(s_a+s_v)+1 \\
&= \frac{1}{4}k^2+\frac{1}{2}ks_a+\frac{3}{2}k-\frac{1}{4}s_a^2-\frac{1}{2}s_v^2-s_as_v+2s_a+\frac{1}{2}s_v+2
\end{aligned}
$$

$S3:(k-s_v)\times(k-s_v)$; A.T.R.: $s_a\times s_a$.

$$\begin{aligned}T_{F_3} &\le \frac{1}{4}k^2-\frac{1}{2}ks_v+\frac{3}{2}k+\frac{1}{4}s_v^2-\frac{3}{2}s_v+1-\frac{1}{2}s_a^2+\frac{1}{2}s_a+1\\ &=\frac{1}{4}k^2-\frac{1}{2}ks_v+\frac{3}{2}k-\frac{1}{2}s_a^2+\frac{1}{4}s_v^2-\frac{3}{2}s_v+\frac{1}{2}s_a+2\end{aligned}$$

$S4:(k-s_a-s_v)\times(k-s_a-s_v)$.

$$\begin{aligned}T_{F_4} &\le \frac{1}{4}k^2+\frac{1}{2}k(-s_a-s_v)+\frac{3}{2}k+\frac{1}{4}(-s_a-s_v)^2+\frac{3}{2}(-s_a-s_v)+1\\ &=\frac{1}{4}k^2-\frac{1}{2}ks_a-\frac{1}{2}ks_v+\frac{3}{2}k+\frac{1}{4}s_a^2+\frac{1}{4}s_v^2+\frac{1}{2}s_as_v-\frac{3}{2}s_a-\frac{3}{2}s_v+1\end{aligned}$$

$S5:(k-s_a)\times(k-s_a)$; A.T.R.: $s_v\times s_v$.

$$\begin{aligned}T_{F_5} &\le \frac{1}{4}k^2-\frac{1}{2}ks_a+\frac{3}{2}k+\frac{1}{4}s_a^2-\frac{3}{2}s_a+1-\frac{1}{2}s_v^2+\frac{1}{2}s_v+1\\ &=\frac{1}{4}k^2-\frac{1}{2}ks_a+\frac{3}{2}k+\frac{1}{4}s_a^2-\frac{1}{2}s_v^2-\frac{3}{2}s_a+\frac{1}{2}s_v+2\end{aligned}$$

$S6:(k+s_v)\times(k+s_v)$; A.T.R.: $(s_a+s_v)\times(s_a+s_v)$.

$$\begin{aligned}T_{F_6} &\le \frac{1}{4}k^2+\frac{1}{2}ks_v+\frac{3}{2}k+\frac{1}{4}s_v^2+\frac{3}{2}s_v+1-\frac{1}{2}(s_a+s_v)^2+\frac{1}{2}(s_a+s_v)+1\\ &=\frac{1}{4}k^2+\frac{1}{2}ks_v+\frac{3}{2}k-\frac{1}{4}s_v^2-\frac{1}{2}s_a^2-s_as_v+\frac{1}{2}s_a+2s_v+2\end{aligned}$$

The number of nodes on the six shifted axis, excluding the originator, is given by $(k+s_v)+(k+s_a)+(k-s_v)+(k-s_a-s_v)+(k-s_a-s_v)+(k-s_a)=6k-2s_a-2s_v$. Thus, the number of forwarding nodes in $\mathcal{H}(G)$ is given by

$$\begin{aligned}T_F &\le T_{F_1}+T_{F_2}+T_{F_3}+T_{F_4}+T_{F_5}+T_{F_6}-6k+2s_a+2s_v\\ &=\frac{3}{2}k^2+3k-s_as_v-s_a^2+4s_a-s_v^2+4s_v+13\\ &=\left[\frac{3}{2}k^2+3k+13-s_as_v\right]+\left[-s_a^2+4s_a\right]+\left[-s_v^2+4s_v\right],\end{aligned}$$

where $0\le s_a\le k$; $0\le s_v<k$; $0\le s_a+s_v\le k$; and if $s_a=0 \implies s_v=0$. Note that $\frac{3}{2}k^2+3k+13-s_as_v$ is maximized when $s_as_v=0$. The maximum of the parabolic function $-s_a^2+4s_a$ occurs when $s_a=2$ and hence, $-s_a^2+4s_a=4$. Similarly, the maximum of the parabolic function $-s_v^2+4s_v$ occurs when $s_v=2$ and hence, $-s_v^2+4s_v=4$. Thus, the maximum possible value of T_F is given by $T_F\le\frac{3}{2}k^2+3k+13+4+4=\frac{3}{2}k^2+3k+21$ for $k\ge4$.

Now, consider when $k \leq 3$. Note that $k + s_a + s_v$ must be even. Thus, for $k = 1$, $k = 2$, and $k = 3$ the shifts s_a and s_v take on the following values and the answer for T_F follows.

$k = 1$**:** $s_a = 1$ and $s_v = 0 \implies T_F \leq \frac{3}{2}k^2 + 3k + 16 \leq \frac{3}{2}k^2 + 3k + 21$

$k = 2$**:** (a) $s_a = 0$ and $s_v = 0 \implies T_F \leq \frac{3}{2}k^2 + 3k + 13 \leq \frac{3}{2}k^2 + 3k + 21$
(b) $s_a = 1$ and $s_v = 1 \implies T_F \leq \frac{3}{2}k^2 + 3k + 18 \leq \frac{3}{2}k^2 + 3k + 21$
(c) $s_a = 2$ and $s_v = 0 \implies T_F \leq \frac{3}{2}k^2 + 3k + 17 \leq \frac{3}{2}k^2 + 3k + 21$

$k = 3$**:** (a) $s_a = 1$ and $s_v = 0 \implies T_F \leq \frac{3}{2}k^2 + 3k + 16 \leq \frac{3}{2}k^2 + 3k + 21$
(b) $s_a = 1$ and $s_v = 2 \implies T_F \leq \frac{3}{2}k^2 + 3k + 18 \leq \frac{3}{2}k^2 + 3k + 21$
(c) $s_a = 2$ and $s_v = 1 \implies T_F \leq \frac{3}{2}k^2 + 3k + 18 \leq \frac{3}{2}k^2 + 3k + 21$
(d) $s_a = 3$ and $s_v = 0 \implies T_F \leq \frac{3}{2}k^2 + 3k + 16 \leq \frac{3}{2}k^2 + 3k + 21$

Thus, for all $k \geq 1$, when $k + s_a + s_v$ is even $\beta \leq \frac{3}{2}k^2 + 3k + 21$.

(b) If $k + s$ is odd $\implies T_S = \frac{1}{4}k^2 + \frac{1}{2}ks + \frac{3}{2}k + \frac{1}{4}s^2 + \frac{3}{2}s + \frac{5}{4}$.

$S1 : (k + s_a + s_v) \times (k + s_a + s_v)$; A.T.R.: $s_a \times s_a$; $s_v \times s_v$.

$$\begin{aligned} T_{F_1} &\leq \frac{1}{4}k^2 + \frac{1}{2}k(s_a + s_v) + \frac{3}{2}k + \frac{1}{4}(s_a + s_v)^2 + \frac{3}{2}(s_a + s_v) + \frac{5}{4} \\ &\quad - \frac{1}{2}s_a^2 + \frac{1}{2}s_a + 1 - \frac{1}{2}s_v^2 + \frac{1}{2}s_v + 1 \\ &= \frac{1}{4}k^2 + \frac{1}{2}ks_a + \frac{1}{2}ks_v + \frac{3}{2}k - \frac{1}{4}s_a^2 - \frac{1}{4}s_v^2 + \frac{1}{2}s_a s_v + 2s_a + 2s_v + \frac{13}{4} \end{aligned}$$

$S2 : (k + s_a) \times (k + s_a)$; A.T.R.: $(s_a + s_v) \times (s_a + s_v)$.

$$\begin{aligned} T_{F_2} &\leq \frac{1}{4}k^2 + \frac{1}{2}ks_a + \frac{3}{2}k + \frac{1}{4}s_a^2 + \frac{3}{2}s_a + \frac{5}{4} - \frac{1}{2}(s_a + s_v)^2 + \frac{1}{2}(s_a + s_v) + 1 \\ &= \frac{1}{4}k^2 + \frac{1}{2}ks_a + \frac{3}{2}k - \frac{1}{4}s_a^2 - \frac{1}{2}s_v^2 - s_a s_v + 2s_a + \frac{1}{2}s_v + \frac{9}{4} \end{aligned}$$

$S3 : (k - s_v) \times (k - s_v)$; A.T.R.: $s_a \times s_a$.

$$\begin{aligned} T_{F_3} &\leq \frac{1}{4}k^2 - \frac{1}{2}ks_v + \frac{3}{2}k + \frac{1}{4}s_v^2 - \frac{3}{2}s_v + \frac{5}{4} - \frac{1}{2}s_a^2 + \frac{1}{2}s_a + 1 \\ &= \frac{1}{4}k^2 - \frac{1}{2}ks_v + \frac{3}{2}k + \frac{1}{4}s_v^2 - \frac{1}{2}s_a^2 - \frac{3}{2}s_v + \frac{1}{2}s_a + \frac{9}{4} \end{aligned}$$

$\underline{S4 : (k - s_a - s_v) \times (k - s_a - s_v).}$

$$
\begin{aligned}
T_{F_4} &\le \frac{1}{4}k^2 + \frac{1}{2}k(-s_a - s_v) + \frac{3}{2}k + \frac{1}{4}(-s_a - s_v)^2 + \frac{3}{2}(-s_a - s_v) + \frac{5}{4} \\
&= \frac{1}{4}k^2 - \frac{1}{2}ks_a - \frac{1}{2}ks_v + \frac{3}{2}k + \frac{1}{4}s_a^2 + \frac{1}{2}s_a s_v + \frac{1}{4}s_v^2 - \frac{3}{2}s_a - \frac{3}{2}s_v + \frac{5}{4}
\end{aligned}
$$

$\underline{S5 : (k - s_a) \times (k - s_a);\ \text{A.T.R.: } s_v \times s_v.}$

$$
\begin{aligned}
T_{F_5} &\le \frac{1}{4}k^2 - \frac{1}{2}ks_a + \frac{3}{2}k + \frac{1}{4}s_a^2 - \frac{3}{2}s_a + \frac{5}{4} - \frac{1}{2}s_v^2 + \frac{1}{2}s_v + 1 \\
&= \frac{1}{4}k^2 - \frac{1}{2}ks_a + \frac{3}{2}k + \frac{1}{4}s_a^2 - \frac{1}{2}s_v^2 - \frac{3}{2}s_a + \frac{1}{2}s_v + \frac{9}{4}
\end{aligned}
$$

$\underline{S6 : (k + s_v) \times (k + s_v);\ \text{A.T.R.: } (s_a + s_v) \times (s_a + s_v).}$

$$
\begin{aligned}
T_{F_6} &\le \frac{1}{4}k^2 + \frac{1}{2}ks_v + \frac{3}{2}k + \frac{1}{4}s_v^2 + \frac{3}{2}s_v + \frac{5}{4} - \frac{1}{2}(s_a + s_v)^2 + \frac{1}{2}(s_a + s_v) + 1 \\
&= \frac{1}{4}k^2 + \frac{1}{2}ks_v + \frac{3}{2}k - \frac{1}{4}s_v^2 - \frac{1}{2}s_a^2 - s_a s_v + 2s_v + \frac{1}{2}s_a + \frac{9}{4}
\end{aligned}
$$

The number of nodes on the six shifted axis, excluding the originator, is given by $(k+s_v)+(k+s_a)+(k-s_v)+(k-s_a-s_v)+(k-s_a-s_v)+(k-s_a) = 6k-2s_a-2s_v$. Thus, the number of forwarding nodes in $\mathcal{H}(G)$ is given by

$$
\begin{aligned}
T_F &\le T_{F_1} + T_{F_2} + T_{F_3} + T_{F_4} + T_{F_5} + T_{F_6} - 6k + 2s_a + 2s_v \\
&= \frac{3}{2}k^2 + 3k - s_a s_v - s_a^2 + 4s_a - s_v^2 + 4s_v + \frac{29}{2} \\
&= \left[\frac{3}{2}k^2 + 3k + \frac{29}{2} - s_a s_v\right] + \left[-s_a^2 + 4s_a\right] + \left[-s_v^2 + 4s_v\right],
\end{aligned}
$$

where $0 \le s_a \le k$; $0 \le s_v < k$; $0 \le s_a + s_v \le k$; and if $s_a = 0 \implies s_v = 0$. Note that $\frac{3}{2}k^2 + 3k + \frac{29}{2} - s_a s_v$ is maximized when $s_a s_v = 0$. The maximum of the parabolic function $-s_a^2 + 4s_a$ occurs when $s_a = 2$ and hence, $-s_a^2 + 4s_a = 4$. Similarly, the maximum of the parabolic function $-s_v^2 + 4s_v$ occurs when $s_v = 2$ and hence, $-s_v^2 + 4s_v = 4$. Thus, the maximum possible value of T_F is given by $T_F \le \frac{3}{2}k^2 + 3k + \frac{29}{2} + 4 + 4 = \frac{3}{2}k^2 + 3k + \frac{45}{2}$ for $k \ge 4$.

Now, consider when $k \le 3$. Note that $k + s_a + s_v$ must be odd. Thus, for $k = 1$, $k = 2$, and $k = 3$ the shifts s_a and s_v take on the following values and the answer for T_F follows.

$k = 1$**:** $s_a = 1$ and $s_v = 0 \implies T_F \le \frac{3}{2}k^2 + 3k + \frac{29}{2} \le \frac{3}{2}k^2 + 3k + \frac{45}{2}$

$k = 2$**:** $s_a = 1$ and $s_v = 0 \implies T_F \leq \frac{3}{2}k^2 + 3k + \frac{35}{2} \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$

$k = 3$**:** (a) $s_a = 1$ and $s_v = 0 \implies T_F \leq \frac{3}{2}k^2 + 3k + \frac{29}{2} \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$
(b) $s_a = 1$ and $s_v = 1 \implies T_F \leq \frac{3}{2}k^2 + 3k + \frac{39}{2} \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$
(c) $s_a = 2$ and $s_v = 0 \implies T_F \leq \frac{3}{2}k^2 + 3k + \frac{37}{2} \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$

Thus, for all $k \geq 1$, when $k + s_a + s_v$ is odd $\beta \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$. Therefore, for all $k \geq 1$, $\beta \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$.

□

Lemma 5.3.5. *Let G denote the wireless mobile ad hoc network of hexagonal shape and $\mathcal{H}(G)$ is centred at a non-blue hexagon. Let the radius of $\mathcal{H}(G)$ be k. If the originator is in a blue hexagon and all blue hexagons are not empty, then the number of forwarders β in G is at most $\beta \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$.*

Proof. Let $\mathcal{H}(G_b)$ denote the hexagonal network with radius k centred at a blue hexagon. The difference between $\mathcal{H}(G)$ and $\mathcal{H}(G_b)$ is a vertical shift of one hexagon (see Figure 5.3.10). Thus, the three of the six sides of the outermost hexagons of $\mathcal{H}(G_b)$ are not in $\mathcal{H}(G)$. Thus, it follows that the number of forwarders in $\mathcal{H}(G)$ is at most $\beta \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$. □

From Lemmas 5.3.4 and 5.3.5, Theorem 5.3.3 follows.

Theorem 5.3.3. *Let G denote the wireless mobile ad hoc network of hexagonal shape, where the radius of $\mathcal{H}(G)$ is k. If the originator is in a blue hexagon and all blue hexagons are not empty, then the number of forwarders β is at most $\beta \leq \frac{3}{2}k^2+3k+\frac{45}{2}$.*

The upper bound on the number of forwarders $\beta \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$ in Theorem 5.3.3 considers the case when blue hexagons are further prioritized into corner, Type A and Type B hexagons. As we noted earlier in Section 5.2.2 this incurs additional delay. To reduce the overall delay, this categorization can be removed and as a result all blue hexagons will forward P. Thus, Theorem 5.3.4 presents this result where blue hexagons are not categorized at all.

Theorem 5.3.4. *Let G denote the wireless mobile ad hoc network of hexagonal shape, where the radius of $\mathcal{H}(G)$ is k. Let the originator be in a blue hexagon, where blue hexagons are not categorized. If all blue hexagons are not empty then the number of forwarders β is at most $\beta \leq 3k^2 + 3k + 1$.*

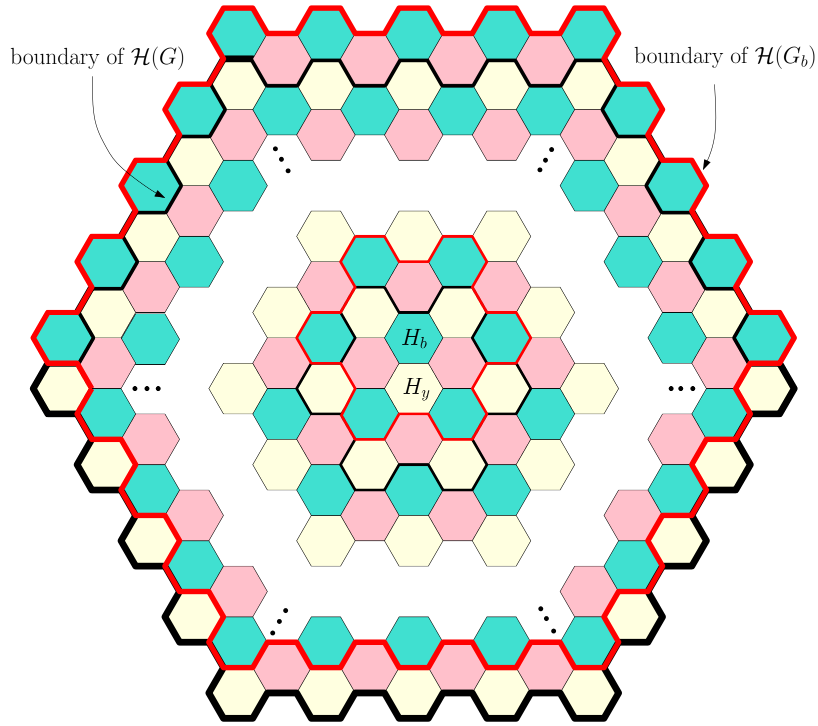


Figure 5.3.10: Black bold line represents the outside boundary of $\mathcal{H}(G)$, which is centred at H_y. The red bold line represents the outside boundary of $\mathcal{H}(G_b)$ centred at H_b and is of the same radius as $\mathcal{H}(G)$. Note that the blue hexagons are entirely engulfed in $\mathcal{H}(G_b)$.

Proof. Since blue hexagons are not categorized, then all blue hexagons will forward P. Thus, to obtain an upper bound on β we must only count the number of blue hexagons in $\mathcal{H}(G)$. Since the radius of $\mathcal{H}(G)$ is k, then from the centre of $\mathcal{H}(G)$ the maximum level of $\mathcal{H}(G)$ is $k-1$. The central hexagon of $\mathcal{H}(G)$ is either a blue hexagon or a non-blue hexagon. From Figure 5.3.10 and Lemmas 5.3.4, 5.3.5 it can be seen that the number of blue hexagons is more when the central hexagon in $\mathcal{H}(G)$ is of colour blue. Thus, without loss of generality let the central hexagon of $\mathcal{H}(G)$ be blue.

On a given level i, the number of blue hexagons on each of the six sides of $\mathcal{H}(G)$ is $i+2$. Hence, there are $6(i+2)-6=6(i+1)$ blue hexagons on level i, since the corner hexagons are shared by consecutive sides. Since the maximum level from the centre of $\mathcal{H}(G)$ is $k-1$, then the number of blue hexagon tiles in $\mathcal{H}(G)$ including the central hexagon is given by

$$\begin{aligned}
&1 + 6(0+1) + 6(1+1) + 6(2+1) + \cdots + 6(k-1+1) \\
&= 1 + 6\Big[1 + 2 + 3 \cdots + k\Big] \\
&= 1 + 6\frac{k(k+1)}{2} = 3k^2 + 3k + 1
\end{aligned}$$

Thus, the number of forwarders β is at most $\beta \leq 3k^2 + 3k + 1$. □

For $k \leq 3$ the result in Theorem 5.3.4 is better than that of in Theorem 5.3.3. However, for $k > 3$ the number of forwarders is much less when blue hexagons are categorized; and as k becomes large enough, the number of forwarders when blue hexagons are not categorized become almost 2 times more than if categorization of hexagons was present.

We now determine the upper bound on the number of forwarding nodes in presence of voids in the network. First we present some observations and definitions.

Definition 5.3.4. *Two hexagons H_i and H_{i+1} are called* aligned hexagons *if H_i and H_{i+1} are of the same colour and the Euclidean distance between their centres is $3\sqrt{3}r$, where r is the radius of a hexagonal tile (see Figure 5.3.11).*

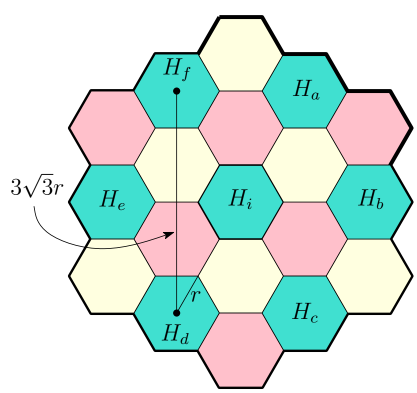


Figure 5.3.11: The pairs of aligned hexagons in $C(H_i)$ are (H_a, H_e), (H_a, H_c), (H_b, H_f), (H_b, H_d), (H_c, H_e), (H_d, H_f).

Consider an empty blue hexagon denoted H_e. Let H_i and H_{i+1} be two bridged blue hexagons in $C(H_e)$. We note the following observations on hexagons forwarding P to cover $C(H_e)$.

Observation 5.3.1. *Let H_a denote an outer yellow or pink hexagon in $C(H_e)$. If H_a is adjacent to both H_i, H_{i+1} (i.e. H_i and H_{i+1} are bridged hexagons) and H_i, H_{i+1} have forwarded P, then H_a will stay silent (see Figure 5.3.12).*

Observation 5.3.2. *A non-blue hexagon in $C(H_e)$ that has received P from two aligned blue hexagons, H_i and H_{i+1}, will stay silent (see Figure 5.3.13).*

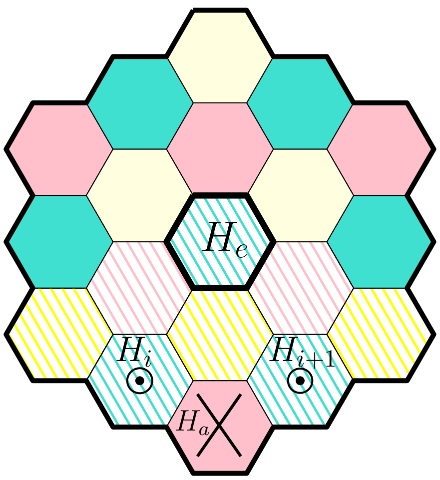


Figure 5.3.12: H_e is empty. H_i and H_{i+1} have forwarded P. Patterned region denote $C(H_e) \cap C(H_a)$, which is covered by H_i and H_{i+1}. Thus, H_a that has received P from two bridged blue hexagons H_i and H_{i+1} stays silent.

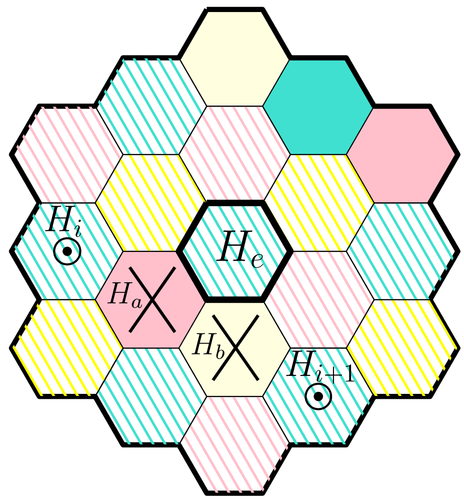


Figure 5.3.13: H_e is empty. H_i, H_{i+1} have forwarded P. Patterned region denote $[C(H_e) \cap C(H_a)] \cup [C(H_e) \cap C(H_a)]$, which is covered by H_i and H_{i+1}. Thus, H_a and H_b that have received P from two aligned hexagons H_i, H_{i+1} stay silent.

Definition 5.3.5. *Let H_i and H_j be blue hexagons and let $H_j \subset C(H_i)$. The hexagons in $[C(H_i) \cap C(H_j)]$ are categorized as* left neighbourhood of H_j*, denoted $LN(H_j)$, and* right neighbourhood of H_j*, denoted $RN(H_j)$. Hexagons in $[C(H_i) \cap C(H_j)]$ encountered in a clockwise direction from the centre of H_j to the centre of H_i fall in $LN(H_j)$. All others in $[C(H_i) \cap C(H_j)]$ fall in $RN(H_j)$ (see Figure 5.3.14).*

Theorem 5.3.5. *Let G denote the wireless mobile ad hoc network of hexagonal shape, where the radius of $\mathcal{H}(G)$ is k. If the originator is in a blue hexagon, then the number of forwarders β is at most $\beta \leq 5\left(\frac{3}{2}k^2 + 3k + \frac{45}{2}\right)$.*

Proof. For each empty Type A and corner hexagon, denoted H_e, we must determine at most how many extra hexagons forward the packet P to cover $C(H_e)$.

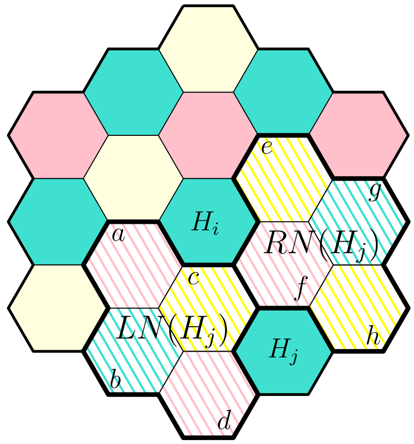


Figure 5.3.14: Hexagons a, b, c, d are in $LN(H_j)$ and hexagons e, f, g, h are in $RN(H_j)$.

In the proof that follows we only consider the cases, where the extra forwarding hexagons are in $C(H_e)$. Note that, there may be a hexagon $H_a \not\subset C(H_e)$ that forwards P and all hexagons in $C(H_a) \cap C(H_e)$ receive P. If H_a is a corner or Type A blue hexagon, then it is not considered as an extra forwarding hexagon to compensate for the empty hexagon H_e. If H_a is a Type B blue hexagon or a yellow or pink hexagon, there must be an empty Type A or corner hexagon in $C(H_a)$ that is empty. Thus, H_a is already counted as an extra forwarding hexagon once. Therefore, it is sufficient to consider the number of extra forwarding hexagons in $C(H_e)$.

We now determine the extra number of forwarding hexagons in $C(H_e)$. Before the details of the proof, we first present a sketch of the proof in several steps that follows.

There are 6 main cases to consider (i.e. any combination of the six blue hexagons in $C(H_e)$ may be empty). These 6 main cases are based on combinations of 4 basic cases. There may be several Type A and corner hexagons in $C(H_e)$ that may forward P or may be empty. Due to each level construction the location orientation of Type A and corner hexagons with respect to Type B hexagons in $C(H_e)$ (denoted *AC-orientation* henceforth) are specific. There are 4 such AC-orientations possible in $C(H_e)$, for any Type A or corner hexagon H_e. Hence, once the number of forwarding hexagons are determined in each of the 6 main cases, the results must be mapped to each AC-orientation of $C(H_e)$ to determine the largest number of forwarding hexagons for one empty Type A or corner hexagon. Thus, the steps of the proof are as follows:

1. 4 base cases in $C(H_e)$.

2. 6 main cases, where any combination of blue hexagons in $C(H_e)$ may be empty (based on combinations of the 4 base cases in (1)).

3. Map results from (2) to each of the 4 AC-orientations of $C(H_e)$.

Base Case 1 (BC1):

Let H_i be an empty blue hexagon in $C(H_e)$ and let all other five blue hexagons forward P as shown in Figure 5.3.15. Crossed pink and yellow hexagons will stay silent by Observations 1 and 2. Thus, instead of H_i forwarding P, a or b will forward P, depending whose timer expires first.

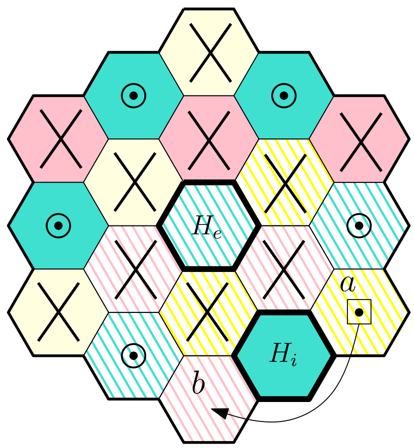


Figure 5.3.15: Bold outlined hexagons denote empty hexagons. Crossed hexagons denote silent hexagons by Observations 1 and 2. Circles/squares centred in hexagons denote forwarding hexagons. Patterned region is $C(H_e) \cap C(H_i)$. a will forward P instead of H_i.

Base Case 2 (BC2):

Let H_i and H_{i+1} be empty blue hexagons in $C(H_e)$ and let all other four blue hexagons forward P as shown in Figure 5.3.16. Crossed pink and yellow hexagons stay silent by Observations 1 and 2. We determine the largest number of extra hexagons to forward P to cover $[C(H_e) \cap C(H_i)] \cup [C(H_e) \cap C(H_{i+1})]$ in two steps: 1) largest number of hexagons needed to forward P to cover $C(H_e) \cap C(H_i)$ and 2) laregest number of hexagons needed to forward P to cover $C(H_e) \cap C(H_{i+1})$.

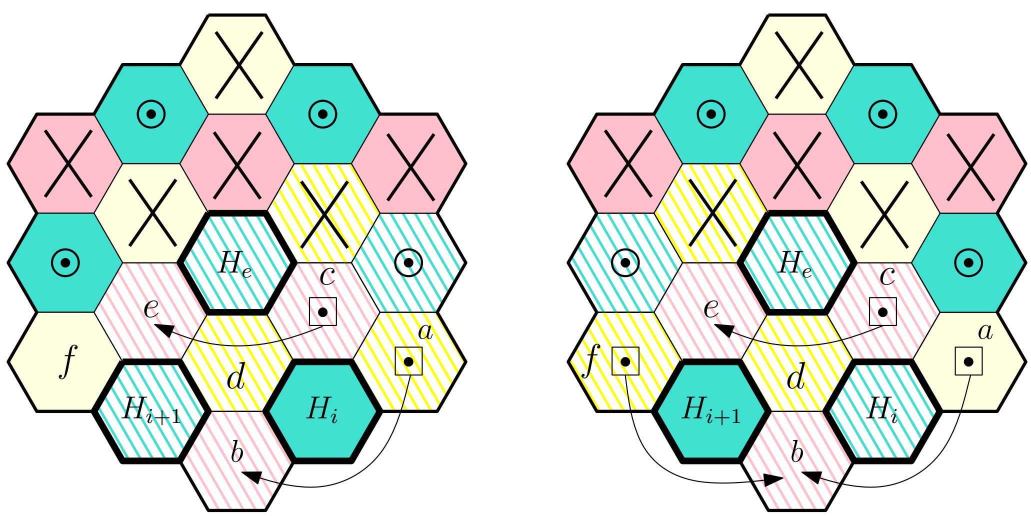
Step1: Determine which of a, b, c, d, e in $C(H_e) \cap C(H_i)$ forward P (Figure 5.3.16(a)).

If d or b forward P, then a, c, e hearing P from d or b will stay silent since d or b

alone cover the non-crossed hexagons (a, c, e, H_{i+1}) in $C(H_e) \cap C(H_i)$. Hence, in this case at most 1 hexagon is needed to forward P to compensate for the empty hexagon H_i. However, if d and b are silent/empty, then a will forward P to cover b, and c will forward P to cover e and H_{i+1} or e will forward P to cover c and H_{i+1}. Thus, at most 2 hexagons will forward P to compensate for the empty hexagon H_i.

Step2: Determine which of b, c, d, e, f in $C(H_e) \cap C(H_{i+1})$ forward P (Figure 5.3.16(b)).
In a similar argument as in *Step1*, if d or b forward P, then c, e, f will stay silent. Thus, in this instant at most 1 hexagon will forward P instead of the empty H_{i+1}. However, if d and b are silent then, f will forward P to cover b, and e will forward P to cover c and H_i or c will forward P to cover e and H_i. Hence, at most 2 hexagons will forward P to compensate for the empty hexagon H_{i+1}. However, note that one of these two hexagons (e or c) is the same as in *Step1*.

Thus, for BC2 at most 3 hexagons forward P to compensate for the empty hexagons H_i and H_{i+1} in $C(H_e)$.



(a) *Step1 of BC2:* the hexagons a and c forward P to cover $C(H_e) \cap C(H_i)$.

(b) *Step2 of BC2:* the hexagons c and f forward P to cover $C(H_e) \cap C(H_{i+1})$.

Figure 5.3.16: Bold outlined hexagons denote empty hexagons. Crossed hexagons denote silent hexagons by Observations 1 and 2. Circles/squares centred in hexagons denote forwarding hexagons. The hexagons a, c, f collectively cover the region $[C(H_e) \cap C(H_i)] \cup [C(H_e) \cap C(H_{i+1})]$.

Base Case 3 (BC3):

Let H_i and H_{i+2} be empty blue hexagons in $C(H_e)$ and let all other four blue hexagons forward P as shown in Figure 5.3.17. Crossed pink and yellow hexagons stay silent by Observations 1 and 2. We determine the largest number of extra hexagons to forward P to cover $[C(H_e) \cap C(H_i)] \cup [C(H_e) \cap C(H_{i+2})]$ in two steps: 1) largest number of hexagons needed to forward P to cover $C(H_e) \cap C(H_i)$ and 2) largest number of hexagons needed to forward P to cover $C(H_e) \cap C(H_{i+2})$.

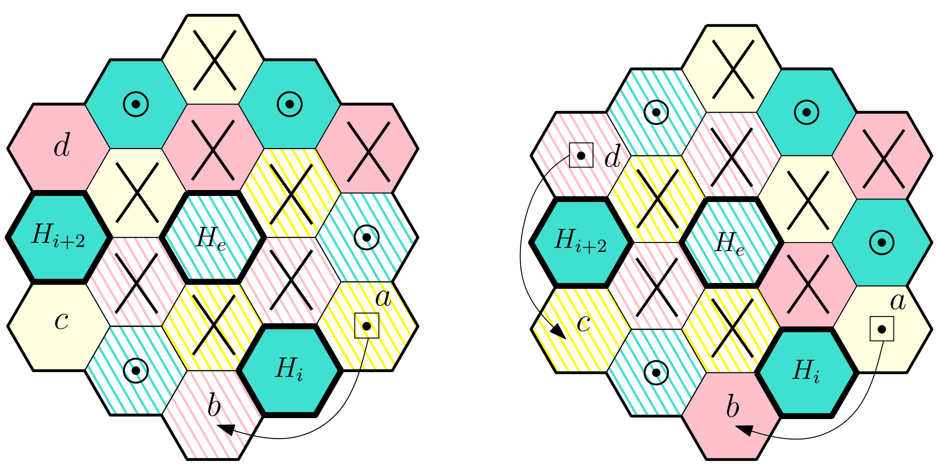
Step1 (Figure 5.3.17(a))

In $C(H_e) \cap C(H_i)$, a and b are the only non-crossed hexagons that may forward P. a will forward P to cover b or vice versa, depending whose timer expires first. Thus, at most 1 hexagon will forward P to compensate for the empty hexagon H_i.

Step2 (Figure 5.3.17(b))

In $C(H_e) \cap C(H_{i+2})$, c and d are the only non-crossed hexagons that may forward P. c will forward P to cover d or vice versa, depending whose timer expires first. Hence, at most 1 hexagon will forward P to compensate for the empty hexagon H_{i+2}.

Thus, for BC3 at most 2 hexagons forward P to compensate for the empty hexagons H_i and H_{i+2} in $C(H_e)$.



(a) *Step1 of BC3:* the hexagon a forwards P to cover $C(H_e) \cap C(H_i)$.

(b) *Step2 of BC3:* the hexagon d forwards P to cover $C(H_e) \cap C(H_{i+2})$.

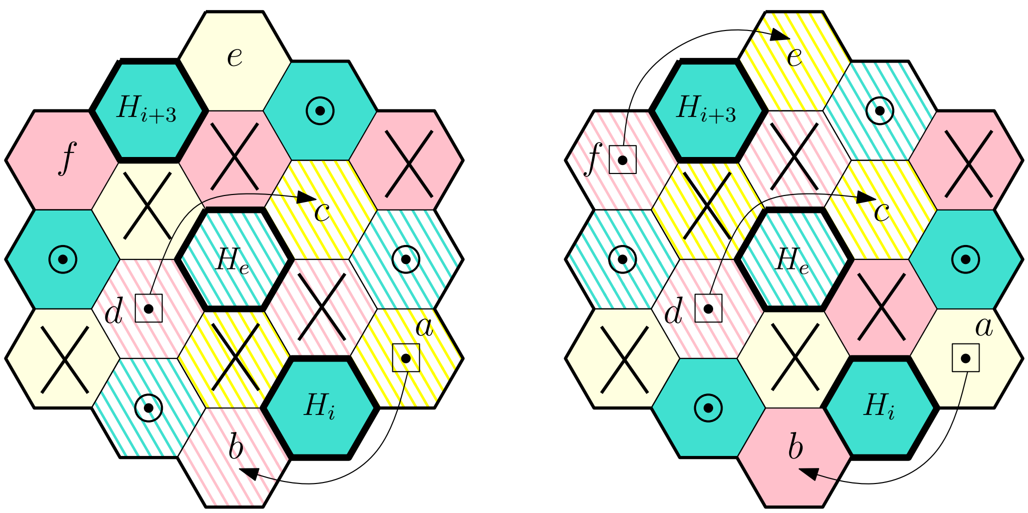
Figure 5.3.17: Bold outlined hexagons denote empty hexagons. Crossed hexagons denote silent hexagons by Observations 1 and 2. Circles/squares centred in hexagons denote forwarding hexagons. The hexagons a, d collectively cover the region $[C(H_e) \cap C(H_i)] \cup [C(H_e) \cap C(H_{i+2})]$.

Base Case 4 (BC4):

Let H_i and H_{i+3} be empty blue hexagons in $C(H_e)$ and let all other four blue hexagons forward P as shown in Figure 5.3.18. Crossed pink and yellow hexagons stay silent by Observations 1 and 2. We determine the largest number of extra hexagons to forward P to cover $[C(H_e) \cap C(H_i)] \cup [C(H_e) \cap C(H_{i+3})]$ in two steps: 1) largest number of hexagons needed to forward P to cover $C(H_e) \cap C(H_i)$ and 2) largest number of hexagons needed to forward P to cover $C(H_e) \cap C(H_{i+3})$.

Step1 (Figure 5.3.18(a))

In $C(H_e) \cap C(H_i)$, a, b, c, d are the only non-crossed hexagons that may forward P. a will forward P to cover b or vice versa, depending whose timer expires first. Similarly, c will forward P to cover d or vice versa, depending whose timer expires first. Thus, at most 2 hexagons will forward P to compensate for the empty hexagon H_i.



(a) *Step1 of BC4:* the hexagons a, d forward P to cover $C(H_e) \cap C(H_i)$.

(b) *Step2 of BC4:* the hexagons d, f forward P to cover $C(H_e) \cap C(H_{i+3})$

Figure 5.3.18: Bold outlined hexagons denote empty hexagons. Crossed hexagons denote silent hexagons by Observations 1 and 2. Circles/squares centred in hexagons denote forwarding hexagons. The hexagons a, d, f collectively cover the region $[C(H_e) \cap C(H_i)] \cup [C(H_e) \cap C(H_{i+3})]$.

Step2 (Figure 5.3.18(b))

In $C(H_e) \cap C(H_{i+3})$, c, d, e, f are the only non-crossed hexagons that may forward P. c will forward P to cover d or vice versa, depending whose timer expires first. Similarly,

e will forward P to cover f or vice versa, depending whose timer expires first. Hence, at most 2 hexagons will forward P to compensate for the empty hexagonH_{i+3}. Note that, either of c or d forwarding P is also considered in *Step1*. Thus, for BC4 at most 3 hexagons forward P to compensate for the empty hexagons H_i and H_{i+3} in $C(H_e)$.

We now consider the 6 main cases, where any combination of the blue hexagons in $C(H_e)$ may be empty (step 2 of proof sketch). Without loss of generality let the 6 blue hexagons in $C(H_e)$ be denoted as H_1, H_2, H_3, H_4, H_5, H_6 as shown in Figure 5.3.19.

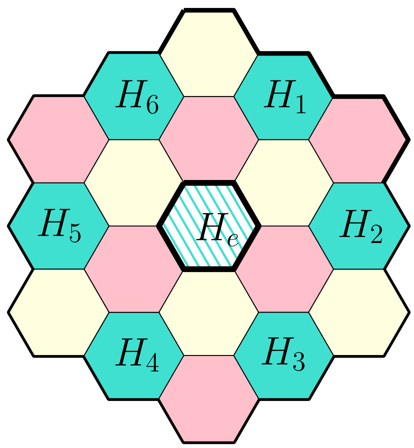


Figure 5.3.19: $C(H_e)$, where H_e is a Type A or corner blue hexagon that is empty.

Case 1: 1 blue hexagon in $C(H_e)$ is empty

Without loss of generality, assume the empty blue hexagon in $C(H_e)$ is H_3. The result of this case is exactly BC1. Hence, the number of yellow and/or pink hexagons that forward P to compensate for the empty H_3 is at most one. Thus, in $C(H_e)$, where one blue hexagon is empty 5 blue hexagons and 1 yellow or pink hexagon forward P.

Case 2: 2 blue hexagon in $C(H_e)$ is empty

(a) Let the two empty blue hexagons be bridged together.

Without loss of generality (*wlog*), assume the empty blue hexagons in $C(H_e)$ are H_3, H_4. The result of this case is exactly as in BC2. Hence, the number of yellow and/or pink hexagons that forward P to compensate for the empty H_3 and H_4 are at most three. Thus, in $C(H_e)$, where H_3 and H_4 are empty, 4 blue hexagons and 3 yellow and/or pink hexagons forward P.

(b) Let the two empty blue hexagons be separated by 2 hops.

Wlog, assume the empty blue hexagons in $C(H_e)$ are H_3, H_5. The result of this case is exactly as in BC3. Hence, the number of yellow and/or pink hexagons that forward P to compensate for the empty H_3 and H_5 are at most two. Thus, in $C(H_e)$, where H_3 and H_5 are empty, 4 blue hexagons and 2 yellow and/or pink pink hexagons forward P.

(c) Let the two empty blue hexagons be separated by 3 hops.

Wlog, assume the empty blue hexagons in $C(H_e)$ are H_3, H_6. The result of this case is exactly as in BC4. Hence, the number of yellow and/or pink hexagons that forward P to compensate for the empty H_3 and H_6 are at most three. Thus, in $C(H_e)$, where H_3 and H_6 are empty, 4 blue hexagons and 3 yellow and/or pink hexagons forward P.

Case 3: 3 blue hexagons in $C(H_e)$ are empty

(a) Let the three empty blue hexagons be separated by 2 hops from each other.

Wlog, assume the empty blue hexagons in $C(H_e)$ are H_3, H_5 and H_1. The result of this case is the combination of BC3 applied three times, once on each of the following pairs: (i) H_3 and H_5; (ii) H_5 and H_1; (iii) H_1 and H_3.

(i) From BC3 at most 2 yellow and/or pink hexagons forward P to compensate for the empty hexagons H_3 and H_5. Denote these two hexagons as a and b. From BC3 one of a, b must be in $C(H_e) \cap C(H_3)$ and the other in $C(H_e) \cap C(H_5)$. Thus, wlog let $a \subset [C(H_e) \cap C(H_3)]$ and $b \subset [C(H_e) \cap C(H_5)]$.

(ii) From BC3 at most 2 yellow and/or pink hexagons forward P to compensate for the empty hexagons H_5 and H_1. From BC3 one of these two hexagons must be in $C(H_e) \cap C(H_5)$ and the other in $C(H_e) \cap C(H_1)$. From (i), the forwarding yellow or pink hexagon in $C(H_e) \cap C(H_5)$ is denoted as b. Denote the forwarding yellow or pink hexagon in $C(H_e) \cap C(H_1)$ as c.

(iii) From BC3 at most 2 yellow and/or pink hexagons forward P to compensate for the empty hexagons H_1 and H_3. From BC3 one of these two hexagons must be in $C(H_e) \cap C(H_1)$ and the other in $C(H_e) \cap C(H_3)$. From (ii), the forwarding yellow or pink hexagon in $C(H_e) \cap C(H_1)$ is denoted as c and from (i) the forwarding yellow or pink hexagon in $C(H_e) \cap C(H_3)$ as a.

Hence, in this instance the yellow and/or pink hexagons a, b, c forward P instead of the empty hexagons H_3, H_5, H_1. Thus, in this case 3 blue hexagons and 3 yellow and/or pink hexagons in $C(H_e)$ forward P.

(b) Let two of the three empty blue hexagons be bridged together and the third blue hexagon be 2 hops from the bridged pair.

Wlog, assume the empty blue hexagons in $C(H_e)$ are H_3, H_4 and H_6. The result of this case is the combination of BC4 applied on the pair (i) H_6 and H_3, BC2 applied on the pair (ii) H_3 and H_4, and BC3 applied on the pair (iii) H_4 and H_6.

(i) From BC4 at most 3 yellow and/or pink hexagons forward P to compensate for the empty hexagons H_6 and H_3. Denote these three hexagons as a, b and c. One of these three hexagons is in $C(H_e) \cap C(H_6)$, one in $C(H_e) \cap C(H_3)$ and the third in $[C(H_e) \cap C(H_6)] \cup [C(H_e) \cap C(H_3)]$. Wlog, let $c \subset C(H_e) \cap C(H_6)$, $a \subset C(H_e) \cap C(H_3)$ and $b \subset [C(H_e) \cap C(H_6)] \cup [C(H_e) \cap C(H_3)]$.
(ii) From BC2 at most 3 yellow and/or pink hexagons forward P to compensate for the empty hexagons H_3 and H_4. Two of these three hexagons are either in $C(H_e) \cap C(H_3)$ or in $C(H_e) \cap C(H_4)$, and the third either in $C(H_e) \cap C(H_4)$ or in $C(H_e) \cap C(H_3)$ respectively. Wlog let two of these three hexagons be in $C(H_e) \cap C(H_3)$ and the third in $C(H_e) \cap C(H_4)$. From (i) we know a, $b \subset C(H_e) \cap C(H_3)$. Denote the third forwarding yellow or pink hexagon in $C(H_e) \cap C(H_4)$ as d.
(iii) From BC3 at most 2 yellow and/or pink hexagons forward P to compensate for the empty hexagons H_4 and H_6. One of these two hexagons is in $C(H_e) \cap C(H_4)$ and the other in $C(H_e) \cap C(H_6)$. From (ii) the forwarding yellow or pink hexagon in $C(H_e) \cap C(H_4)$ is denoted as d, and from (i) the forwarding yellow or pink hexagon in $C(H_e) \cap C(H_6)$ is denoted as c.

Hence, in this instance the yellow and/or pink hexagons a, b, c, d forward P instead of the empty hexagons H_3, H_4, H_6. Thus, in this case 3 blue hexagons and 4 yellow and/or pink hexagons in $C(H_e)$ forward P.

(c) Let the three empty blue hexagons consist of 2 bridged pairs.

Wlog, assume the empty blue hexagons in $C(H_e)$ are H_3, H_4 and H_5. The result of this case is the combination of BC2 applied twice, once on each of the following pairs: (i) H_3 and H_4; (ii) H_4 and H_5 (see Figure 5.3.20).

(i) From BC2 at most 3 yellow and/or pink hexagons forward p to compensate for the empty hexagons H_3 and H_4. Denote these 3 hexagons as a, b, c. From BC2 one of a, b, c is in $C(H_e) \cap C(H_3)$, one in $C(H_e) \cap C(H_4)$ and one in $[C(H_e) \cap C(H_3)] \cup [C(H_e) \cap C(H_4)]$. Wlog, let $a \subset C(H_e) \cap C(H_3)$, $c \subset C(H_e) \cap C(H_4)$, and $b \subset [C(H_e) \cap C(H_3)] \cup [C(H_e) \cap C(H_4)]$.
(ii) From BC2 at most 3 hexagons forward P to compensate for empty hexagons H_4 and H_5. The three hexagons must be in $[C(H_e) \cap C(H_4)] \cup [C(H_e) \cap C(H_5)]$. From ($i$) a, $c \subset C(H_e) \cap C(H_4)$. Hence, the third forwarding yellow or pink hexagon, denoted as d, must be in $C(H_e) \cap C(H_5)$.

Hence, in this instance the yellow and/or pink hexagons a, b, c, d forward P instead of the empty hexagons H_3, H_4, H_5. Thus, in this case at most 3 blue hexagons and 4 yellow and/or pink hexagons in $C(H_e)$ forward P.

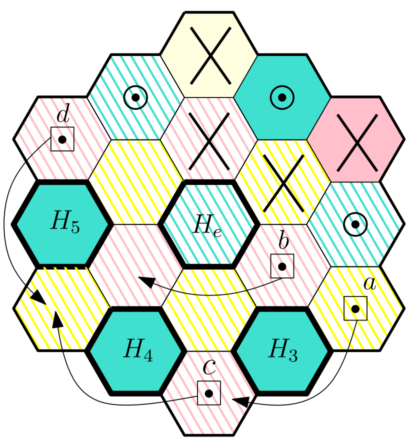


Figure 5.3.20: Bold outlined hexagons denote empty hexagons. Tiled hexagons denote the hexagons in the region $[C(H_e) \cap C(H_3)] \cup [C(H_e) \cap C(H_4)] \cup [C(H_e) \cap C(H_5)]$. Crossed hexgons denote silent hexagons by Observations 1 and 2. Circles/squares centred in hexagons denote forwarding hexagons. The hexagons a, b, c, d collectively cover the region $[C(H_e) \cap C(H_3)] \cup [C(H_e) \cap C(H_4)] \cup [C(H_e) \cap C(H_5)]$

Case 4: 4 blue hexagons in $C(H_e)$ are empty

(a) Let the four empty blue hexagons consist of 2 bridged pairs separated by 2 hops.

Wlog, assume the empty blue hexagons in $C(H_e)$ are H_3, H_4, H_6 and H_1 (see Figure 5.3.21). The result of this case is BC2 applied together with BC4 twice, once on each of the following pairs: (i) H_3, H_4 with H_6; (ii) H_6, H_1 with H_4.

(i) From BC2 at most 3 hexagons will forward P to compensate for the empty hexagons H_3 and H_4. Denote these hexagons as a, b, c. From BC2, note that two of a, b, c must be outer hexagons and one an inner hexagon in $C(H_e)$. Wlog let a and b be outer hexagons in $C(H_e)$ and hence, let $a \subset C(H_e) \cap C(H_3)$, $b \subset C(H_e) \cap C(H_4)$. Wlog let $c \subset C(H_e) \cap C(H_3)$. Note that c may also be in $C(H_e) \cap C(H_4)$.

From BC4 at most 3 hexagons will forward P to compensate for the empty hexagons H_3 and H_6. One of these three hexagons must be in $C(H_e) \cap C(H_3)$, one in $C(H_e) \cap C(H_6)$ and the third may be in $C(H_e) \cap C(H_3)$ and/or $C(H_e) \cap C(H_6)$. Denote the forwarding yellow or pink hexagon in $C(H_e) \cap C(H_6)$ as d. Note a, c have forwarded P and that a, $c \subset C(H_e) \cap C(H_3)$. Since c is an inner hexagon in $C(H_e)$, no other inner hexagon in $C(H_e) \cap C(H_6)$ will forward P to cover $C(H_e) \cap C(H_6)$ since they have heard P from c and d.

(ii) From BC4 at most 3 hexagons will forward P to compensate for the empty hexagons H_4 and H_1. From BC4 one of these three hexagons must be in $C(H_e) \cap C(H_4)$, one in $C(H_e) \cap C(H_1)$ and one in $C(H_e) \cap C(H_4)$ and/or $C(H_e) \cap C(H_1)$. From (i) $b \subset C(H_e) \cap C(H_4)$. Let the forwarding yellow or pink hexagon in $C(H_e) \cap C(H_1)$ be denoted as e. From (i) c is an inner hexagon in $C(H_e)$ and $c \subset C(H_e) \cap C(H_3)$. Hence, $c \subset C(H_e) \cap C(H_4)$ and/or $c \subset C(H_e) \cap C(H_1)$.

From BC2 at most 3 hexagons will forward P to compensate for the empty hexagons H_6 and H_1. One of these three hexagons must be in $C(H_e) \cap C(H_6)$, one in $C(H_e) \cap C(H_1)$ and one in $C(H_e) \cap C(H_6)$ and/or $C(H_e) \cap C(H_1)$. From (i) $d \subset C(H_e) \cap C(H_6)$. From ($ii$) the hexagon e forwards P in $C(H_e) \cap C(H_1)$. From (i) c is an inner hexagon and $c \subset C(H_e) \cap C(H_3)$. Hence, $c \subset C(H_e) \cap C(H_6)$ and/or $c \subset C(H_e) \cap C(H_1)$.

Hence, in this instance the yellow and/or pink hexagons a, b, c, d, e forward P instead of the empty hexagons H_3, H_4, H_6 and H_1. Thus, in this case at most 2 blue hexagons and 5 yellow and/or pink hexagons in $C(H_e)$ forward P.

(b) Let three of the four empty blue hexagons consist of 2 bridged pairs and the fourth empty blue hexagon is separated from the 2 bridged pairs by 2 hops.

Wlog, assume the empty blue hexagons in $C(H_e)$ are H_3, H_4, H_5 and H_1 (see Figure 5.3.22). The result of this case is the combination of (i) the results of *Case* 3(c), where the empty hexagons are H_3, H_4, H_5 and (ii) BC4 applied on the pair H_1 and H_4.

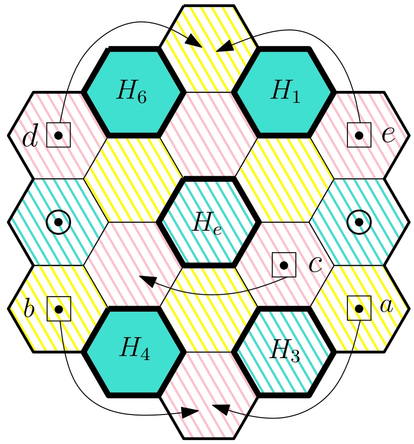


Figure 5.3.21: Bold outlined hexagons denote empty hexagons. Tiled hexagons denote the hexagons in the region $[C(H_e) \cap C(H_3)] \cup [C(H_e) \cap C(H_4)] \cup [C(H_e) \cap C(H_6)] \cup [C(H_e) \cap C(H_1)]$. Circles/squares centred in hexagons denote forwarding hexagons. The hexagons a, b, c, d, e collectively cover the region $[C(H_e) \cap C(H_3)] \cup [C(H_e) \cap C(H_4)] \cup [C(H_e) \cap C(H_6)] \cup [C(H_e) \cap C(H_1)]$.

(i) From *Case* $3(c)$ at most 4 yellow and/or pink hexagons forward P to compensate for the empty hexagons H_3, H_4, H_5. Denote these hexagons as a, b, c, d.
(ii) From BC4 at most 3 hexagons forward P to compensate for the empty hexagons H_1 and H_4. Note that by *Case* $3(c)$ two of a, b, c, d must be in $C(H_e) \cap C(H_4)$. Thus, the third hexagon in $C(H_3) \cap C(H_1)$ to forward P by BC4 is denoted as e.

Hence, in this instance the yellow and/or pink hexagons a, b, c, d, e forward P to compensate for the empty hexagons H_3, H_4, H_5 and H_1. Thus, in this case at most 2 blue hexagons and 5 yellow and/or pink hexagons in $C(H_e)$ forward P.

(c) Let the four empty blue hexagons consist of 3 bridged pairs.

Wlog, assume the empty blue hexagons in $C(H_e)$ are H_3, H_4, H_5 and H_6. The result of this case is similar to that of *Case* 6. Thus, in this case at most 2 blue hexagons and 5 yellow and/or pink hexagons in $C(H_e)$ forward P.

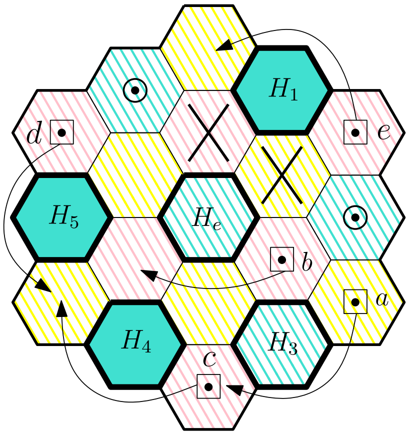


Figure 5.3.22: Bold outlined hexagons denote empty hexagons. Tiled hexagons denote the hexagons in the region $[C(H_e) \cap C(H_3)] \cup [C(H_e) \cap C(H_4)] \cup [C(H_e) \cap C(H_5)] \cup [C(H_e) \cap C(H_1)]$. Crossed hexgons denote silent hexagons by Observation 2. Circles/squares centred in hexagons denote forwarding hexagons. The hexagons a, b, c, d, e collectively cover the region $[C(H_e) \cap C(H_3)] \cup [C(H_e) \cap C(H_4)] \cup [C(H_e) \cap C(H_5)] \cup [C(H_e) \cap C(H_1)]$.

Case 5: 5 blue hexagons in $C(H_e)$ are empty

Wlg, assume the empty blue hexagons in $C(H_e)$ are H_3, H_4, H_5, H_6 and H_1.

The result of this case is the result of *Case* 6. Thus, in this case at most 1 blue hexagon and 5 yellow and/or pink hexagons in $C(H_e)$ forward P.

Case 6: 6 blue hexagons in $C(H_e)$ are empty

The empty blue hexagons in $C(H_e)$ are H_1, H_2, H_3, H_4, H_5 and H_6 (see Figure 5.3.23).

The result of this is the combinations of BC2 applied on each of the following pairs: (i) H_3 and H_4; (ii) H_4 and H_5; (iii) H_5 and H_6; (iv) H_6 and H_1; (v) H_1 and H_2; (vi) H_2 and H_3.

By considering BC2 and BC4 at the same time, for a blue hexagon $H_i \subset C(H_e)$, the number of non-blue hexagons in $LN(H_i) \cup RN(H_i)$ are at most 3 as was seen in *Case 4(a)*. To maximize the number of non-blue forwarders, two of these 3 hexagons must be outer hexagons and as a result the third is an inner hexagon in $C(H_e)$.

(i) Considering the empty pair H_3 and H_4.

BC2 Step 1 (covering $LN(H_3) \cup RN(H_3)$)

At most 3 non-blue hexagons forward P to cover $LN(H_3) \cup RN(H_3)$. Two of these three hexagons are outer hexagons and the third is an inner hexagon in $C(H_e)$. Denote the outer hexagons as a, b and the inner hexagon as c. By BC2 and BC4 the hexagons a, b cannot both be in $LN(H_3)$ or $RN(H_3)$. Hence, wlog let $a \subset RN(H_3)$ and $b \subset LN(H_3)$. The hexagon c is either in $LN(H_3)$ or in $RN(H_3)$. Wlog, let $c \subset RN(H_3)$.

BC2 Step 2 (covering $LN(H_4) \cup RN(H_4)$)

From BC2 Step 1, $b \subset LN(H_3) \implies b \subset RN(H_4)$. Non-blue hexagons in $RN(H_4)$ will stay silent since they have heard P from c, a, b and do not cover any additional hexagons in $LN(H_4) \cup RN(H_4)$. Since c is an inner hexagon it covers all inner hexagons in $C(H_e)$ and hence, inner hexagons will not forward P to cover inner hexagons. Since a is an outer hexagon in $RN(H_3)$ it covers the outer hexagon, namely b, in $LN(H_3)$. Since b is an outer hexagon in $RN(H_4)$ it will cover the outer hexagon in $LN(H_4)$. Thus, out of the three non-blue hexagons in $LN(H_4)$, at most one will forward P to completely cover $LN(H_4)$. Denote this hexagon as d.

(ii) Considering the empty pair H_4 and H_5.

BC2 Step 1 (covering $LN(H_4) \cup RN(H_4)$)

The result is of that in BC2 Step 2 of (i).

BC2 Step 2 (covering $LN(H_5) \cup RN(H_5)$)

Since $d \subset LN(H_4) \implies d \subset RN(H_5)$ or $d \subset LN(H_5)$. Wlog let $d \subset RN(H_5)$. The non-blue hexagons in $LN(H_5) \cup RN(H_5)$ will stay silent since they heard P from b, c, d collectively, which cover $LN(H_5) \cup RN(H_5)$ completely.

(iii) Considering the empty pair H_5 and H_6.

BC2 Step 1 (covering $LN(H_5) \cup RN(H_5)$)

The result is of that in BC2 Step 2 of (ii).

BC2 Step 2 (covering $LN(H_6) \cup RN(H_6)$)

From (i) $c \subset RN(H_3) \implies c \subset LN(H_6)$, otherwise there would be less than three forwarding hexagons in (i). The inner hexagons of $C(H_e)$ in $RN(H_6)$ will stay silent since they heard P from c and d, which collectively cover $LN(H_6) \cup RN(H_6)$. The outer hexagon in $RN(H_6)$ may forward P to cover the two non-blue hexagons in $LN(H_6)$ since it would not have heard P from c. As a result the two non-blue hexagons in $LN(H_6)$ will stay silent. If, however, the outer hexagon in $RN(H_6)$ stays silent, then one of the non-blue hexagons in $LN(H_6)$ will forward P since none of them would have heard P from d. Denote this additional forwarding hexagon in $LN(H_6) \cup RN(H_6)$ as e.

(*iv*) Considering the empty pair H_6 and H_1.

BC2 Step 1 (covering $LN(H_6) \cup RN(H_6)$)

The result is of that in BC2 Step 2 of (*iii*).

BC2 Step 2 (covering $LN(H_1) \cup RN(H_1)$)

Since $c \subset RN(H_3) \implies c \subset LN(H_1)$ (H_3 and H_1 are aligned hexagons). The three non-blue hexagons in $RN(H_1)$ will stay silent since they heard P from c and e, which collectively cover $LN(H_1) \cup RN(H_1)$. The two non-blue hexagons in $LN(H_1)$ that have not forwarded P will also stay silent since they heard P from c and a and do not cover any additional hexagons in $LN(H_1) \cup RN(H_1)$.

(*v*) Considering the empty pair H_1 and H_2.

BC2 Step 1 (covering $LN(H_1) \cup RN(H_1)$)

The result is of that in BC2 Step 2 of (*iv*).

BC2 Step 2 (covering $LN(H_2) \cup RN(H_2)$)

Since $c \subset RN(H_3)$ and is an inner hexagon $\implies c \subset RN(H_2)$. Otherwise there would be less than three forwarding hexagons in (*i*). Since $a \subset RN(H_3)$ and is an outer hexagon $\implies a \subset LN(H_2)$. All other non-blue hexagons in $LN(H_2) \cup RN(H_2)$ will stay silent since they heard P from a and c, which collectively cover $LN(H_2) \cup RN(H_2)$.

(*vi*) Considering the empty pair H_2 and H_3.

BC2 Step 1 (covering $LN(H_2) \cup RN(H_2)$)

The result is of that in BC2 Step 2 of (*v*).

BC2 Step 2 (covering $LN(H_3) \cup RN(H_3)$)

The result of this case is BC2 Step 1 of (*i*).

Hence, the non-blue forwarding hexagons are a, b, c, d, and e in $C(H_e)$. Thus, there are no blue forwarding hexagons and 5 yellow and/or pink forwarding hexagons in $C(H_e)$.

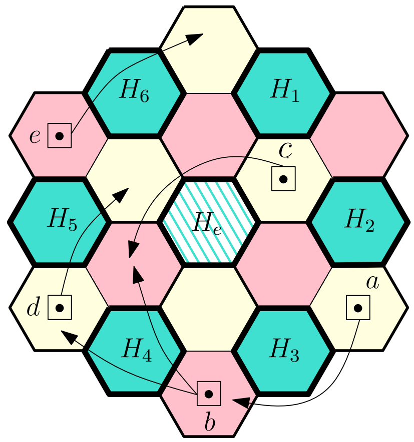
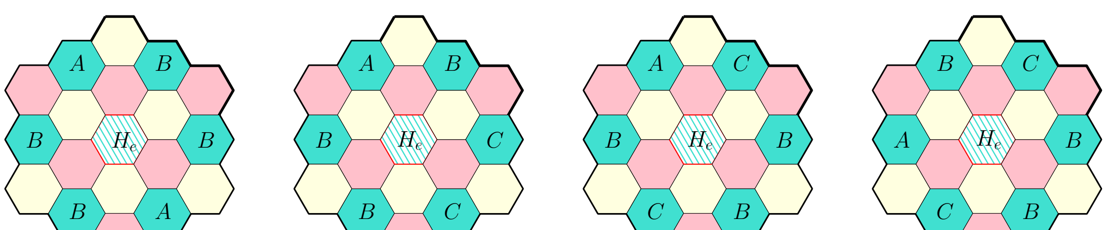


Figure 5.3.23: Bold outlined hexagons denote empty hexagons. Squares centred in hexagons denote forwarding hexagons. The hexagons a, b, c, d and e collectively cover $C(H_e)$.

In the third step of the proof we map the results of each of the 6 cases to the 4 AC-orientations of $C(H_e)$. The 4 AC-orientations of $C(H_e)$ are when (1) H_e is Type A hexagon on an odd level, (2) H_e is a Type A hexagon on an even level, (3) H_e is a corner hexagon on an odd level, and (4) H_e is a corner hexagon on an even level (see Figure 5.3.24). Note that Type A and corner hexagons have the same function, that is they always forward P. Thus, we see that Figures 5.3.24(b), 5.3.24(c), 5.3.24(d) are isomorphic. Therefore, we can only consider 2 mappings of AC-orientations of $C(H_e)$.



(a) H_e is a Type A hexagon on an odd level.

(b) H_e is a Type A hexagon on an even level.

(c) H_e is a corner hexagon on an odd level.

(d) H_e is a corner hexagon on an even level.

Figure 5.3.24: The 4 AC-orientations of $C(H_e)$, where H_e is an empty Type A or corner hexagon.

From each of the 6 cases it can be determined how many blue and how many non-blue hexagons forward P. Among the blue hexagons empty in $C(H_e)$, aside from H_e, the empty blue hexagons may be of Type A, corner or Type B. Thus, we must consider all cases and map them to all 4 AC-orientations to determine the largest number of non-blue and Type B hexagons in $C(H_e)$ that forward P for one empty Type A or corner hexagon. Note that if H_e is the originating hexagon H_o, then all blue hexagons in $C(H_o)$ are corner hexagons. Hence, in this case there will be no Type B blue hexagons that may forward P, but only non-blue hexagons. Thus, the number of forwarding hexagons in this case cannot be more than that of the results of any of the six cases. We present the summary of these results in a table for each case. Bold entries in the Tables 5.1-5.6 designate the largest number of forwarding hexagons for a corner or Type A hexagon.

Case 1: 1 blue hexagon is empty

Number of empty Type A and Corner hexagons in $C(H_e)$	**Number of Type B and non-blue forwarding hexagons in $C(H_e)$**
1	3 or **4**
2	4 or 5

Table 5.1: Results of *Case* 1.

Thus, in this case for one empty Type A or corner hexagon at most 4 additional hexagons forward P.

Case 2: 2 **blue hexagon are empty**

Case 2	**Number of empty Type A and Corner hexagons in** $C(H_e)$	**Number of Type B and non-blue forwarding hexagons in** $C(H_e)$
(a)	**1** 2 3	**4** 5 6
(b)	1 **1** or 2 2 or 3	3 **4** 5
(c)	1 **1** or 2 3	4 **5** 6 or 7

Table 5.2: Results of *Case* 2.

Thus, in this case for one empty Type A or corner hexagon at most 5 additional hexagons forward P.

Case 3: 3 **blue hexagon are empty**

Case 3	**Number of empty Type A and Corner hexagons in** $C(H_e)$	**Number of Type B and non-blue forwarding hexagons in** $C(H_e)$
(a)	**2** 3	4 or **5** 5
(b)	**1** 2 3	4 or **5** 5 or 6 6 or 7
(c)	**2** 3	5 or **6** 6

Table 5.3: Results of *Case* 3.

Thus, in this case for one empty Type A or corner hexagon at most 5 additional hexagons forward P.

Case 4: 4 **blue hexagon are empty**

Case 4	**Number of empty Type A and Corner hexagons in** $C(H_e)$	**Number of Type B and non-blue forwarding hexagons in** $C(H_e)$
(a)	**1** or 2 3 4	**5** 6 or 7 7
(b)	**2** 3 4	5 or **6** 6 7
(c)	**2** 3 4	5 or **6** 6 or 7 7

Table 5.4: Results of *Case* 4.

Thus, in this case for one empty Type A or corner hexagon at most 5 additional hexagons forward P.

Case 5: 5 **blue hexagon are empty**

Number of empty Type A and Corner hexagons in $C(H_e)$	**Number of Type B and non-blue forwarding hexagons in** $C(H_e)$
2	**5**
3	5 or 6
4	6

Table 5.5: Results of *Case* 5.

Thus, in this case for one empty Type A or corner hexagon at most 5 additional hexagons forward P.

Case 6: 6 **blue hexagon are empty**

Number of empty Type A and Corner hexagons in $C(H_e)$	**Number of Type B and non-blue forwarding hexagons in $C(H_e)$**
3 or 4	**5**

Table 5.6: Results of *Case* 6.

Thus, in this case for one empty Type A or corner hexagon at most 5 additional hexagons forward P.

From Theorem 5.3.3, the number of forwarding hexagons in $\mathcal{H}(G)$ if there are no voids is at most $\frac{3}{2}k^2+3k+\frac{45}{2}$. From the results of Tables 5.1-5.6 it can be seen that the worst ratio of the number of empty Type A and corner hexagons to the number Type B and non-blue forwarding hexagons is 1 : 5. Thus, for one empty Type A or corner hexagon at most 5 additional hexagons forward P. Therefore, $\beta \leq 5\left(\frac{3}{2}k^2+3k+\frac{45}{2}\right)$. □

The number of forwarding nodes β, when there are no holes in the network and blue hexagons are categorized is at most $\beta \leq \frac{3}{2}k^2+3k+\frac{45}{2}$ (given in Theorem 5.3.3). The number of forwarders in a network that may contain holes is at most $\beta \leq 5\left(\frac{3}{2}k^2+3k+\frac{45}{2}\right)$ (given in Theorem 5.3.5). The lower bound given in Theorem 5.3.2 is not sharp. The number of forwarding nodes β is at least

$$\begin{aligned}
\beta &\geq \left\lceil \frac{9\sqrt{3}\left(9k^2+3k+1\right)-168\pi}{28\left(2\pi+3\sqrt{3}\right)} \right\rceil + 1 \\
&\geq \frac{9\sqrt{3}\left(9k^2+3k+1\right)-168\pi}{28\left(2\pi+3\sqrt{3}\right)} + 1 \\
&\geq 0.436k^2+0.145k-0.594 \\
&\geq \frac{3}{7}k^2+\frac{1}{7}k-\frac{3}{5}
\end{aligned}$$

Thus, the upper bound with no holes in the network and where blue hexagons are categorized (Theorem 5.3.3) approximately is $\frac{7}{2}$ times the lower bound. The upper bound, where there may be holes in the network (Theorem 5.3.5) is approximately $5\left(\frac{7}{2}\right)$ of the lower bound in Theorem 5.3.2.

5.3.2 Dilation Factor and Broadcast Time

In this section, we consider the length of the shortest hexagonal path obtained by HBLF from the originator s to a node v. We also consider the *dilation factor* in networks that do not contain holes. That is, blue hexagons are not empty. *Dilation factor* is defined as the ratio of the shortest hexagonal path from s to v by HBLF to the shortest path in the network from s to v. In the analysis that follows we also consider the broadcast time of HBLF, that is the number of time units it takes for HBLF to complete. Let $B_t(G)$ denote the broadcast time of HBLF in the network G. Now we present our result on the shortest hexagonal path from the originator s to a node v by HBLF. We only know the Euclidean distance between the two nodes s and v, from which an upper bound on the number of hops from s to v returned by HBLF must be determined.

Lemma 5.3.6. *Let G denote the wireless mobile ad hoc network of hexagonal shape, where blue hexagons in $\mathcal{H}(G)$ are not empty. Let s be the originator in hexagon H_s centred at c_s. The length of the shortest hexagonal path from s to a node $v \in V(G)$ in hexagon H_v centred at c_v is at most $l_{HBLF} \leq \left\lceil \frac{2}{\sqrt{3}} \frac{d_E(c_s,c_v)}{3r} \right\rceil$, where $d_E(c_s, c_v)$ is the Euclidean distance between c_s, c_v and r is the radius of each hexagon tile in $\mathcal{H}(G)$.*

Proof. To determine the shortest hexagonal path from s to v is equivalent to determining the shortest hexagonal path from H_s to H_v. There are two cases to consider: (1) H_v is a corner blue hexagon and (2) H_v is a Type A/B or a yellow/pink hexagon.
Case 1: H_v is a corner hexagon

Since H_s is the originating hexagon it is on level 0. Let H_v be on level $i \geq 0$. The shortest hexagonal path by HBLF from H_s to H_v is via the corner hexagons on levels $0, 1, 2, \cdots, i-2, i-1$. Thus, the length of the shortest hexagonal path is $\frac{d_E(c_s,c_v)}{3r}$, where $3r$ is the Euclidean distance between the centres of two corner (i.e. bridged) hexagons.

Case 2: H_v is a Type A/B or a yellow/pink hexagon

If H_v is a Type A hexagon, let H_v be on level $i \geq 2$ (there are no Type A hexagons on levels 0 and 1). Each forwarding blue hexagon (i.e. Type A or corner hexagons) on a given level i is bridged with a forwarding blue hexagon on level $i-1$. This is due to the way categorization of blue hexagons are done.

If H_v is a Type B hexagon, let H_v be on level $i \geq 1$ (there are no Type B hexagons on level 0). Each Type B hexagon on level i is bridged with a Type A or a corner hexagon on level $i-1$.

If H_v is a yellow or pink hexagon, let H_v be on level $i \geq 0$. Each yellow or pink hexagon on level i is in the coverage area of at least one Type A and/or corner hexagon on level $i-1$.

Since H_v on level i is in the coverage area of a Type A and/or corner hexagon on level $i-1$, then each transmitting hexagon on the shortest hexagonal path P_{HBLF} from H_s on level 0 to H_v on level i is on a different level than any of the other hexagons on P_{HBLF}. Thus, the length of the shortest hexagonal path is the same as the length of the shortest hexagonal path from H_s to a corner hexagon H_c on level i centred at c_c. We now determine the length of the shortest hexagonal path from H_s to H_v in terms of the Euclidean distance $d_E(c_s, c_v)$.

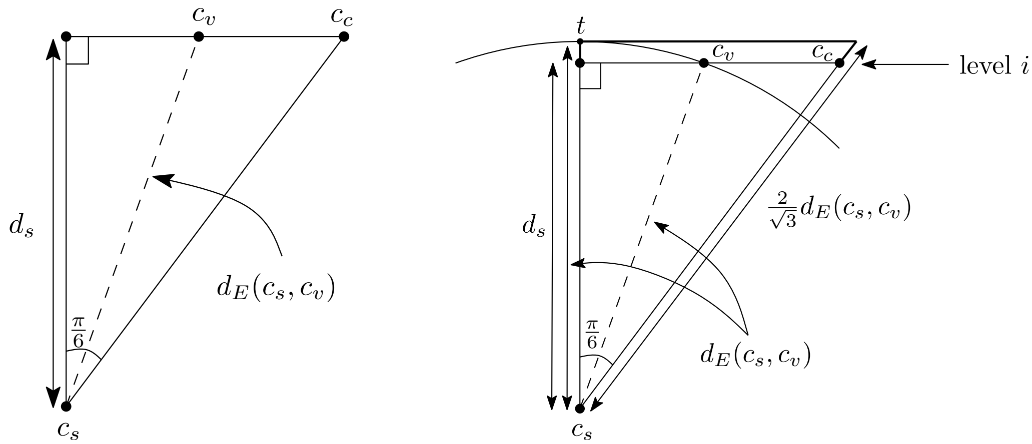
The length of the shortest hexagonal path from H_s to H_c on level i is given by $\frac{d_E(c_s,c_v)}{3r} = i+1$ (*Case 1*). For a non-corner blue hexagon on level i $d_E(c_s, c_v) < d_E(c_s, c_c)$ and hence, $\left\lceil \frac{d_E(c_s,c_v)}{3r} \right\rceil$ may not necessarily give a correct upper bound on the length of the shortest hexagonal path (i.e. the path length may be less than $i+1$ depending on the value of $d_E(c_s, c_v)$) as it did in *Case 1*. Thus, we determine an upper bound on the length of the shortest hexagonal path from s to v returned by HBLF.

The angle made by two consecutive corner hexagons on level i with H_s on level 0 is $\frac{\pi}{3}$. The shortest Euclidean distance possible from c_s to c_v, denoted d_s, is when the line $\overline{c_s c_v}$ is perpendicular to the line $\overline{c_v c_c}$. Thus, the angle made by $\angle c_v c_s c_c = \frac{\pi}{6}$ (see Figure 5.3.25(a)). Thus, $d_s \leq d_E(c_s, c_v) \leq d_E(c_s, c_c)$. We take a tangent from point t (see Figure 5.3.25(b)) to the circle the radius of which is $d_E(c_s, c_v)$. As can be seen from Figure 5.3.25(b) the distance $\frac{2}{\sqrt{3}} d_E(c_s, c_v)$ is beyond level i. That is, $\frac{2}{\sqrt{3}} d_E(c_s, c_v) \geq d_E(c_s, c_c)$ and hence, $\left\lceil \frac{2}{\sqrt{3}} \frac{d_E(c_s,c_v)}{3r} \right\rceil \geq i+1$. Thus, the length of the shortest hexagonal path from s to v is at most $\left\lceil \frac{2}{\sqrt{3}} \frac{d_E(c_s,c_v)}{3r} \right\rceil$.

By combining *Case 1* and *Case 2* we have

$$\frac{2}{\sqrt{3}} \frac{d_E(c_s, c_v)}{3r} > \frac{d_E(c_s, c_v)}{3r} \implies \left\lceil \frac{2}{\sqrt{3}} \frac{d_E(c_s, c_v)}{3r} \right\rceil \geq \frac{d_E(c_s, c_v)}{3r}$$

Thus, the shortest hexagonal path from the originator s in hexagon H_s centred at c_s to a node v in hexagon H_v centred at c_v is at most $l_{HBLF} \leq \left\lceil \frac{2}{\sqrt{3}} \frac{d_E(c_s,c_v)}{3r} \right\rceil$. □



(a) $d_s \leq d_E(c_s, c_v) \leq d_E(c_s, c_c)$.

(b) The number of hops from c_s to c_v is bounded by $\left\lceil \frac{2}{\sqrt{3}} \frac{d_E(c_s,c_v)}{3r} \right\rceil$.

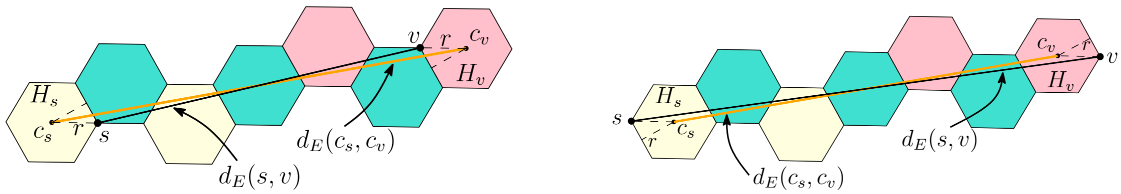
Figure 5.3.25: c_s is the central point of H_s that contains node s. c_c is the central point of H_c. c_v is the central point of H_v that contains v. d_s is the shortest possible Euclidean distance from c_s to c_v.

Theorem 5.3.6. *Let G denote the wireless mobile ad hoc network of hexagonal shape, where blue hexagons in $\mathcal{H}(G)$ are not empty. Let s be the originator. Let l_{min} be the shortest path from s to any node $v \in V(G)$ and l_{HBLF} be the shortest hexagonal path from s to v. Then, $\frac{l_{HBLF}}{l_{min}} \leq \frac{4(\sqrt{7}+2)}{3\sqrt{3}} + 2$.*

Proof. The shortest path between the originator s and any node v is given by $\left\lceil \frac{d_E(s,v)}{2\sqrt{7}r} \right\rceil$, where the transmission range of each node is $R = 2\sqrt{7}r$ and r is the radius of each hexagon tile in $\mathcal{H}(G)$. Let s be in the hexagon H_s centred at c_s and v be in the hexagon H_v centred at c_v. Then by Lemma 5.3.6, the length of the shortest hexagonal path from s to v is at most $\left\lceil \frac{2}{\sqrt{3}} \frac{d_E(c_s,c_v)}{3r} \right\rceil$, where again r is the radius of hexagon tiles in $\mathcal{H}(G)$. Thus,

$$\frac{l_{HBLF}}{l_{min}} \leq \frac{\left\lceil \frac{2}{\sqrt{3}} \frac{d_E(c_s,c_v)}{3r} \right\rceil}{\left\lceil \frac{d_E(s,v)}{2\sqrt{7}r} \right\rceil} \leq \frac{\left(\frac{2}{\sqrt{3}} \frac{d_E(c_s,c_v)}{3r} + 1 \right)}{\left(\frac{d_E(s,v)}{2\sqrt{7}r} \right)} = \left(\frac{2d_E(c_s, c_v) + 3\sqrt{3}r}{3\sqrt{3}r} \right) \left(\frac{2\sqrt{7}r}{d_E(s, v)} \right)$$

For nodes s and v, the range of the Euclidean distance $d_E(s, v)$ in terms of $d_E(c_s, c_v)$ is given by $d_E(c_s, c_v) - 2r \leq d_E(s, v) \leq d_E(c_s, c_v) + 2r$ (see Figure 5.3.26). Since $d_E(c_s, c_v) \leq d_E(s, v) + 2r$, then



(a) $d_E(s, v) \geq d_E(c_s, c_v) - 2r$ (b) $d_E(s, v) \leq d_E(c_s, c_v) + 2r$

Figure 5.3.26: $d_E(c_s, c_v) - 2r \leq d_E(s, v) \leq d_E(c_s, c_v) + 2r$

$$\begin{aligned}\frac{l_{HBLF}}{l_{min}} &\leq \left(\frac{2\left[d_E(s,v)+2r\right]+3\sqrt{3}r}{3\sqrt{3}r}\right)\left(\frac{2\sqrt{7}r}{d_E(s,v)}\right) \\ &= \frac{2\sqrt{7}\left(2\, d_E(s,v)+(4+3\sqrt{3})r\right)}{3\sqrt{3}\, d_E(s,v)} \\ &= \frac{2\sqrt{7}}{3\sqrt{3}}\left(2+\frac{(4+3\sqrt{3})r}{d_E(s,v)}\right)\end{aligned}$$

If $H_v \subset C(H_s)$, then $l_{HBLF} = l_{min}$ and thus, $l_{HBLF}/l_{min} = 1$. Otherwise, for $H_v \not\subset C(H_s)$, the shortest Euclidean distance between s and v is $d_E(s, v) > \sqrt{7}r$, which is the distance from the top left apex of H_s to the bottom right apex of H_v or the opposite (i.e from bottom left to top right), or from the top right apex of H_s to the bottom left apex of H_v or the opposite (i.e. from bottom right to top left) (see Figure 5.3.27).

Thus,

$$\begin{aligned}\frac{l_{HBLF}}{l_{min}} &\leq \frac{2\sqrt{7}}{3\sqrt{3}}\left(2+\frac{(4+3\sqrt{3})r}{\sqrt{7}r}\right) \\ &= \frac{2}{3\sqrt{3}}\left(2\sqrt{7}+(4+3\sqrt{3})\right) \\ &= \frac{4(\sqrt{7}+2)}{3\sqrt{3}}+2\end{aligned}$$

□

Note that, when there may be holes in the network, the dilation factor is not constant as is in the case when there are no holes in the network. The reason for this is that, for example, the shortest path between the originator s and a node v may

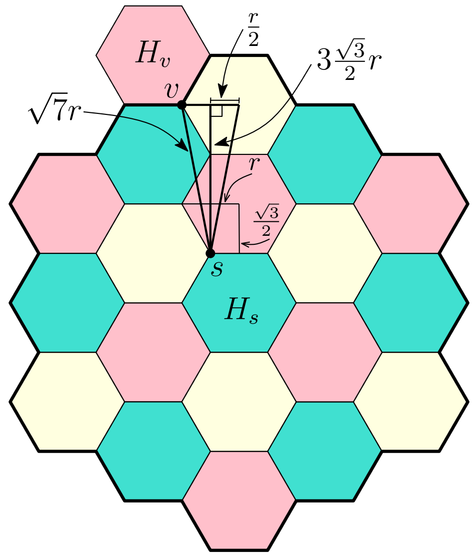


Figure 5.3.27: For $H_v \not\subset C(H_s)$, the shortest Euclidean distance between $s \in H_s$ and $v \in H_v$ is $d_E(s, v) > \sqrt{7}r$.

be 1 hop, while HBLF may return a much longer hexagonal path traversing over the entire network.This is due to the definition of the coverage area of a given hexagon. The shortest hexagonal path returned by HBLF in such a case depends on the radius of the hexagonal network, denoted as k. As will be seen in the results that follow on the broadcast time, the length of the longest shortest hexagonal path returned by HBLF is quadratic based on k.

Lemma 5.3.7. *Let G denote the wireless mobile ad hoc network of hexagonal shape, where the central hexagon of $\mathcal{H}(G)$ is blue. Let the originator be in a blue hexagon and let k denote the radius of $\mathcal{H}(G)$. If all yellow and pink hexagons are empty then $B_t(G) \leq \frac{3}{2}k^2 + 3k + 4$.*

Proof. Let H_o be the originating hexagon and denote the hexagon farthest from H_o as H_f. To determine $B_t(G)$, we must determine the length of the longest hexagonal path from H_o to H_f, denoted l_{max}, since $B_t(G) = l_{max} + 1$.

Let P_{max} denote the longest hexagonal path from H_o to H_f. Note that all hexagons in P_{max} are of colour blue since all yellow and pink hexagons are assumed to be empty. Since P_{max} is the longest hexagonal path from H_o to H_f, for any two subsequent segments S_1 and S_2 of P_{max}, S_1 and S_2 must be separated by enough empty blue hexagons so that from any hexagon H_i on S_1 there will not be a shorter path than P_{max} to any hexagon H_j on S_2. To obtain the largest number of forwarders, the number of blue hexagons necessary to be empty to keep the separation between any two segments of P_{max} is exactly one. That is, for any hexagon H_a on a segment S_i and a hexagon H_b on segment S_{i+1}, H_a and H_b are separated by two hops. Using this, we now determine the largest number of forwarding hexagons in each of the six sectors centred at H_o from which we will determine the largest number of forwarding nodes in G to determine l_{max}.

The originating hexagon H_o can be anywhere in $\mathcal{H}(G)$. Let H_c denote the central hexagon of $\mathcal{H}(G)$. Depending from which hexagon the HBLF algorithm starts, the number forwarders may vary. Thus, we must consider the shift from H_o to H_c as was done in Lemma 5.3.4.

There are two possible shifts: a shift on one of the six axis, denoted s_a and an angular vertical shift, denoted s_v. Figure 5.3.9 depicts the two possible shifts of H_o from H_c in a hexagonal shape network G. Let the total shift be denoted by $s = s_a + s_v$. By the HBLF algorithm, the construction of each level is centred at H_o. The dimensions of each of the shifted sectors and the additional triangular regions not in $\mathcal{H}(G)$ (labelled A.T.R.) are as follows.

$S1$**:** $(k - s_a + 2s_a + s_v) \times (k - s_a + 2s_a + s_v) \equiv (k + s_a + s_v) \times (k + s_a + s_v)$
A.T.R.: $s_a \times s_a$; $s_v \times s_v$

$S2$**:** $(k - s_v + s_a + s_v) \times (k - s_v + s_a + s_v) \equiv (k + s_a) \times (k + s_a)$
A.T.R.: $(s_a + s_v) \times (s_a + s_v)$

$S3$**:** $(k - s_a - s_v + s_a) \times (k - s_a - s_v + s_a) \equiv (k - s_v) \times (k - s_v)$
A.T.R.: $s_a \times s_a$

$S4$**:** $(k - s_a - s_v) \times (k - s_a - s_v)$

$S5$**:** $(k - s_a - s_v + s_v) \times (k - s_a - s_v + s_v) \equiv (k - s_a) \times (k - s_a)$
A.T.R.: $s_v \times s_v$

$S6$**:** $(k - s_a + s_a + s_v) \times (k - s_a + s_a + s_v) \equiv (k + s_v) \times (k + s_v)$
A.T.R.: $(s_a + s_v) \times (s_a + s_v)$

Thus, for each $(k + s) \times (k + s)$ sector we calculate the largest possible number of forwarding hexagons. On each level $i = k + s - 1$ (centred at H_o), there are $i + 2$ blue hexagons. To obtain the largest number of forwarding hexagons, where the hop distance between two blue hexagons on two different segments of P_{max} is at least 2, occurs when we consider the sum of hexagons on levels $(k+s-1)$, $(k+s-3)$, $(k+s-5)$, $\cdots, 2$ (if $k+s$ is odd) or 1 (if $k+s$ is even). Thus, we consider the two cases: $k+s$ is even and $k+s$ is odd.

If $(k+s)$ is even:

The largest possible number of forwarders in a sector of dimension $(k+s)\times(k+s)$ including H_o is

$$\begin{aligned} T_S &= 1+3+5+\cdots+(k+s-1)+(k+s+1) \\ &= \left(\left\lceil\frac{k+s+1}{2}\right\rceil\right)^2 = \left(\frac{k+s}{2}+1\right)^2 \\ &= \frac{(k+s)^2}{4}+2\frac{(k+s)}{2}+1 \\ &= \frac{1}{4}k^2+\frac{1}{2}ks+k+\frac{1}{4}s^2+s+1 \end{aligned}$$

If $(k+s)$ is odd:

The largest possible number of forwarders in a sector of dimension $(k+s)\times(k+s)$ excluding H_o is

$$\begin{aligned} T_S &= 2+4+6+\cdots+(k+s-1)+(k+s+1) \\ &= \left(\frac{k+s+1}{2}\right)\left(\frac{k+s+1}{2}+1\right) = \frac{(k+s+1)(k+s+3)}{4} \\ &= \frac{1}{4}k^2+\frac{1}{2}ks+k+\frac{1}{4}s^2+s+\frac{3}{4} \end{aligned}$$

An A.T.R. defined by a sector $s\times s$, have a base of $s+1$ hexagons that share a boundary with $\mathcal{H}(G)$. Note that forwarding hexagons in T_S also include hexagons in A.T.R. defined by $s\times s$ sectors. Thus, we calculate the number of hexagons in $s\times s$ sector that are also considered in T_S and subtract it from T_S. This calculation is similar to that of T_S.

If s is even, then the number of forwarding hexagons in $s\times s$ sector is given by $\frac{1}{4}s^2+s+1$. This sum includes the $\left(\frac{s}{2}+1\right)$ forwarding hexagons (including the apex hexagon of $s\times s$ sector) on the boundary of $\mathcal{H}(G)$. Thus, the number of forwarding hexagons in $s\times s$ sector that are not in $\mathcal{H}(G)$ is given by $T_D=\frac{1}{4}s^2+s+1-\frac{s}{2}-1=\frac{1}{4}s^2+\frac{1}{2}s$. Thus, the number of forwarding hexagons in a sector of dimensions $(k+s)\times(k+s)$ and in $\mathcal{H}(G)$ is at most $T_{F_j}\le T_{S_j}+\sum\limits_{\forall s\times s}-\frac{1}{4}s^2-\frac{1}{2}s$.

If s is odd, then the number of forwarding hexagons in $s \times s$ sector is given by $\frac{1}{4}s^2 + s + \frac{3}{4}$. This sum includes the $\lceil \frac{s}{2} \rceil$ forwarding hexagons on the boundary of $\mathcal{H}(G)$. Thus, the number of forwarding hexagons in $s \times s$ sector that are not in $\mathcal{H}(G)$ is given by $T_D = \frac{1}{4}s^2 + s + \frac{3}{4} - \lceil \frac{s}{2} \rceil = \frac{1}{4}s^2 + s + \frac{3}{4} - \left(\frac{s-1}{2} + 1\right) = \frac{1}{4}s^2 + \frac{1}{2}s + \frac{1}{4}$. Thus, the number of forwarding hexagons in a sector of dimensions $(k+s) \times (k+s)$ and in $\mathcal{H}(G)$ is given by $T_{F_j} \leq T_{S_j} + \sum_{\forall s \times s} -\frac{1}{4}s^2 - \frac{1}{2}s - \frac{1}{4}$.

Therefore, the number of forwarding hexagons in $\mathcal{H}(G)$ that constitute to P_{max} is the sum of all the T_{F_j} sums of each of the six shifted sectors. Now we present this calculation. There are two cases to consider: (a) if $(k+s)$ is even and (b) if $(k+s)$ is odd.

(a) if $(k+s)$ is even $\implies T_S = \frac{1}{4}k^2 + \frac{1}{2}ks + k + \frac{1}{4}s^2 + s + 1$.

$S1 : (k + s_a + s_v) \times (k + s_a + s_v)$; A.T.R.: $s_a \times s_a$; $s_v \times s_v$.

$$
\begin{aligned}
T_{F_1} \leq\ & \frac{1}{4}k^2 + \frac{1}{2}k(s_a + s_v) + k + \frac{1}{4}(s_a + s_v)^2 + (s_a + s_v) + 1 - \frac{1}{4}s_a^2 - \frac{1}{2}s_a \\
& - \frac{1}{4}s_v^2 - \frac{1}{2}s_v \\
=\ & \frac{1}{4}k^2 + \frac{1}{2}ks_a + \frac{1}{2}ks_v + k + \frac{1}{2}s_a s_v + \frac{1}{2}s_a + \frac{1}{2}s_v + 1
\end{aligned}
$$

$S2 : (k + s_a) \times (k + s_a)$; A.T.R.: $(s_a + s_v) \times (s_a + s_v)$.

$$
\begin{aligned}
T_{F_2} &\leq \frac{1}{4}k^2 + \frac{1}{2}ks_a + k + \frac{1}{4}s_a^2 + s_a + 1 - \frac{1}{4}(s_a + s_v)^2 - \frac{1}{2}(s_a + s_v) \\
&= \frac{1}{4}k^2 + \frac{1}{2}ks_a + k - \frac{1}{4}s_v^2 - \frac{1}{2}s_a s_v + \frac{1}{2}s_a - \frac{1}{2}s_v + 1
\end{aligned}
$$

$S3 : (k - s_v) \times (k - s_v)$; A.T.R.: $s_a \times s_a$.

$$
\begin{aligned}
T_{F_3} &\leq \frac{1}{4}k^2 - \frac{1}{2}ks_v + k + \frac{1}{4}s_v^2 - s_v + 1 - \frac{1}{4}s_a^2 - \frac{1}{2}s_a \\
&= \frac{1}{4}k^2 - \frac{1}{2}ks_v + k - \frac{1}{4}s_a^2 + \frac{1}{4}s_v^2 - \frac{1}{2}s_a - s_v + 1
\end{aligned}
$$

$S4 : (k - s_a - s_v) \times (k - s_a - s_v)$.

$$
\begin{aligned}
T_{F_4} &\leq \frac{1}{4}k^2 + \frac{1}{2}k(-s_a - s_v) + k + \frac{1}{4}(-s_a - s_v)^2 + (-s_a - s_v) + 1 \\
&= \frac{1}{4}k^2 - \frac{1}{2}ks_a - \frac{1}{2}ks_v + k + \frac{1}{4}s_a^2 + \frac{1}{4}s_v^2 + \frac{1}{2}s_a s_v - s_a - s_v + 1
\end{aligned}
$$

$\underline{S5 : (k - s_a) \times (k - s_a);\ \text{A.T.R.: } s_v \times s_v.}$

$$T_{F_5} \leq \frac{1}{4}k^2 - \frac{1}{2}ks_a + k + \frac{1}{4}s_a^2 - \frac{1}{4}s_v^2 - s_a - \frac{1}{2}s_v + 1$$

$\underline{S6 : (k + s_v) \times (k + s_v);\ \text{A.T.R.: } (s_a + s_v) \times (s_a + s_v).}$

$$\begin{aligned} T_{F_6} &\leq \frac{1}{4}k^2 + \frac{1}{2}ks_v + k + \frac{1}{4}s_v^2 + s_v + 1 - \frac{1}{4}(s_a + s_v)^2 - \frac{1}{2}(s_a + s_v) \\ &= \frac{1}{4}k^2 + \frac{1}{2}ks_v + k - \frac{1}{4}s_a^2 - \frac{1}{2}s_a s_v - \frac{1}{2}s_a + \frac{1}{2}s_v + 1 \end{aligned}$$

Thus, the largest number of forwarding hexagons in $\mathcal{H}(G)$ centred at H_c, where the originating hexagon in H_o, is $\sum_{j=1}^{6} T_{F_j}$. Note that the result of each T_{F_j} includes H_o. That is, in $\sum_{j=1}^{6} T_{F_j}$, H_o is counted 6 times. Also note that in $\sum_{j=1}^{6} T_{F_j}$, the hexagons that fall on the 6 shifted axis are counted twice. The number of all hexagons on the 6 shifted axis is given by $(k+s_v)+(k+s_a)+(k-s_v)+(k-s_a-s_v)+(k-s_a-s_v)+(k-s_a) = 6k - 2s_a - 2s_v$, excluding H_o. Thus, the number of forwarding hexagons on the 6 shifted axis is given by $\frac{6k-2s_a-2s_v}{2} = 3k - s_a - s_v$. Therefore,

$$\begin{aligned} T_F &\leq \sum_{j=1}^{6} T_{F_j} - 5 - 3k + s_a + s_v \\ &= \frac{3}{2}k^2 + 6k - 2s_a - 2s_v + 6 - 5 - 3k + s_a + s_v \\ &= \frac{3}{2}k^2 + 3k - s_a - s_v + 1 \end{aligned}$$

Henc, $l_{max} + 1 \leq \frac{3}{2}k^2 + 3k - s_a - s_v + 1$, which will be maximized if $s_a = 0$ and $s_v = 0$. Thus, $l_{max} + 1 \leq \frac{3}{2}k^2 + 3k + 1$.

$\underline{(b) \text{ if } (k + s) \text{ is odd}} \implies T_S = \frac{1}{4}k^2 + \frac{1}{2}ks + k + \frac{1}{4}s^2 + s + \frac{3}{4}$.

Note that the difference between the T_S functions when $(k+s)$ is even and $(k+s)$ is odd is $-\frac{1}{4}$. Similarly, the difference between the T_D functions when s is even and s is odd is $-\frac{1}{4}$. Let $T_{F_{j(odd)}}$ denote the corresponding T_{F_j} for $1 \leq j \leq 6$ when $k + s$ is odd. Thus, for

$S1$**:** $T_{F_{1(odd)}} \leq T_{F_1} - \frac{1}{4} - \frac{1}{4} - \frac{1}{4}$

$S2$**:** $T_{F_{2(odd)}} \leq T_{F_2} - \frac{1}{4} - \frac{1}{4}$

$S3$: $T_{F_{3(odd)}} \leq T_{F_2} - \frac{1}{4} - \frac{1}{4}$

$S4$: $T_{F_{4(odd)}} \leq T_{F_2} - \frac{1}{4}$

$S5$: $T_{F_{5(odd)}} \leq T_{F_2} - \frac{1}{4} - \frac{1}{4}$

$S6$: $T_{F_{6(odd)}} \leq T_{F_2} - \frac{1}{4} - \frac{1}{4}$.

Thus,

$$\begin{aligned}\sum_{j=1}^{6} T_{F_{j(odd)}} &\leq \sum_{j=1}^{6} T_{F_j} + 12\left(-\frac{1}{4}\right)\\ &= \frac{3}{2}k^2 + 6k - 2s_a - 2s_v + 6 - 3\\ &= \frac{3}{2}k^2 + 6k - 2s_a - 2s_v + 3\end{aligned}$$

Note that the originating hexagon H_o is not included in the count of $\sum_{j=1}^{6} T_{F_{j(odd)}}$. Also note that the hexagons on the 6 shifted axis are counted twice in $\sum_{j=1}^{6} T_{F_{j(odd)}}$. The number of forwarding hexagons on the 6 shifted axis as was previously determined (in the case of even $k+s$) is given by $3k - s_a - s_v$. Therefore,

$$\begin{aligned}T_F &\leq T_{F_{j(odd)}} + 1 - 3k + s_a + s_v\\ &= \frac{3}{2}k^2 + 6k - 2s_a - 2s_v + 3 + 1 - 3k + s_a + s_v\\ &= \frac{3}{2}k^2 + 3k - s_a - s_v + 4\end{aligned}$$

Hence, $l_{max} + 1 \leq \frac{3}{2}k^2 + 3k - s_a - s_v + 4$, which is maximized when $s_a = 0$ and $s_v = 0$. Thus, $l_{max} + 1 \leq \frac{3}{2}k^2 + 3k + 4$, when $k+s$ is odd.

Therefore, from cases (a) and (b) we have $B_t(G) \leq \frac{3}{2}k^2 + 3k + 4$.

□

Lemma 5.3.8. *Let G denote the wireless mobile ad hoc network of hexagonal shape, where the central hexagon of $\mathcal{H}(G)$ is non-blue. Let the originator be in a blue hexagon and let k be the radius of $\mathcal{H}(G)$. If all yellow and pink hexagons are empty then $B_t(G) \leq \frac{3}{2}k^2 + 3k + 4$.*

Proof. Let $\mathcal{H}(G_b)$ denote the overlay hexagon network centred at a blue hexagon and is of radius k. The difference between $\mathcal{H}(G)$ and $\mathcal{H}(G_b)$ is a vertical shift of one hexagon. That is, $\mathcal{H}(G)$ includes in its region all blue hexagons of $\mathcal{H}(G_b)$ except three consecutive outermost sides of blue hexagons of $\mathcal{H}(G_b)$ that are excluded due to the shift (see Figure 5.3.10). From Lemma 5.3.7, $B_t(G_b) \leq \frac{3}{2}k^2 + 3k + 4$. Since all yellow and pink hexagons are assumed to be empty, then it follows that $B_t(G) \leq B_t(G_b) \leq \frac{3}{2}k^2 + 3k + 4$. □

Theorem 5.3.7. *Let G denote the wireless mobile ad hoc network of hexagonal shape. Let the originator be in a blue hexagon and let k be the radius of $\mathcal{H}(G)$. For any combination of empty hexagons of any colour in G, $B_t(G) \leq 3k^2 + 6k + 8$.*

Proof. Let the originating hexagon be denoted as H_o and the hexagon farthest from H_o as H_f. Let the longest hexagonal path from H_o to H_f in $\mathcal{H}(G)$ be denoted as P_{max}. Let $\mathcal{H}(G')$ denote the same overlay network of hexagons as $\mathcal{H}(G)$, where all yellow and pink hexagons are empty. To ensure the longest hexagonal path via the blue hexagons from H_o to H_f in $\mathcal{H}(G')$ some of the blue hexagons must be empty. Let the longest hexagonal path from H_o to H_f in $\mathcal{H}(G')$ be denoted as P'_{max}. Now, we show that the length of P_{max}, denoted l_{max}, in $\mathcal{H}(G)$ is at most two times the length of P'_{max}, denoted l'_{max}, in $\mathcal{H}(G')$.

Let S_i and S_{i+1} be two subsequent segments on P_{max} for $i \geq 1$. From Lemmas 5.3.7 and 5.3.8 we see that in order to maximize l'_{max}, the width between S_i and S_{i+1} must be large enough to fit an empty blue hexagon (i.e. a hexagon that bridges a hexagon on S_i to a hexagon on S_{i+1}) and yellow and pink hexagons must be empty. Note that, however, in $\mathcal{H}(G)$ there may be yellow, pink and/or blue hexagons in between S_i and S_{i+1} that are not empty. There may also be a blue hexagon on S_i that is empty in $\mathcal{H}(G)$. Thus, we must consider these cases and the effects of it on l_{max} in comparison to l'_{max}. There are two cases that we must consider: (a) yellow, pink and blue hexagons within the width between S_i and S_{i+1} may not be empty in $\mathcal{H}(G)$ (as was considered in Lemmas 5.3.7 and 5.3.8), and (b) for any hexagon on S_i that is empty, we must determine at most how many extra yellow, pink and/or blue hexagons within the width between S_i and S_{i+1} forward the message and not interfere with hexagons on S_{i+1} (i.e. create a shorter path from S_i to S_{i+1}).

For any two hexagons a and b let $h(a, b)$ denote the length of the shortest hexagonal path from a to b.

Case (a): hexagons between S_i and S_{i+1} may not be empty in $\mathcal{H}(G)$

For this case we determine if yellow, pink and/or blue hexagons within the width between S_i and S_{i+1} are not empty, then how does l_{max} compare to l'_{max}. As shown in Figure 5.3.28, let (a, b) be part of the segment S_i and (d, e) be part of the segment S_{i+1}. The shortest possible hexagonal path from S_i to S_{i+1} is through the blue hexagon c in 2 hops as shown in Figure 5.3.28.

From Figure 5.3.28 the following results can be seen.

If the blue hexagons B_1 and B_2 are not empty in $\mathcal{H}(G)$, then $h(a, d) = 2$ and $h(a, e) = 2$ on P_{max} compared to $h(a, d) = 4$ and $h(a, e) = 5$ on P'_{max} in $\mathcal{H}(G')$. Thus, $l_{max} \leq l'_{max}$. That is, to maximize l_{max} B_1, B_2 must be empty.

If any of Y_1, Y_2, Y_3 are not empty in $\mathcal{H}(G)$, then $h(b, e) = 2$ and $h(a, e) = 2$ on P_{max} compared to $h(b, e) = 3$ and $h(a, e) = 4$ on P'_{max} in $\mathcal{H}(G')$. Thus, $l_{max} \leq l'_{max}$. That is, to maximize l_{max} Y_1, Y_2, Y_3 must be empty.

If any of P_1, P_2, P_3 are not empty in $\mathcal{H}(G)$, then $h(b, e) = 2$ and $h(a, e) = 2$ on P_{max} compared to $h(b, e) = 3$ and $h(a, e) = 4$ on P'_{max}. Thus, $l_{max} \leq l'_{max}$. That is, to maximize l_{max} P_1, P_2, P_3 must be empty.

If F_{Y1} is not empty and F_{P1} is empty in $\mathcal{H}(G)$, then $h(a, c) = 2$ on P_{max} through the hexagon F_{Y1}. On P'_{max} $h(a, c) = 2$ as well through the hexagon b. Thus, $l_{max} = l'_{max}$. However, if F_{Y1} and F_{P1} are both non-empty in $\mathcal{H}(G)$, then $h(a, e) = 3$ on P_{max} compared to $h(a, e) = 4$ on P'_{max}. Thus, in this case $l_{max} \leq l'_{max}$. That is, to maximize l_{max} one of F_{Y1} and F_{P1} must be empty.

Similarly if both F_{Y2} and F_{P2} are non-empty in $\mathcal{H}(G)$, then $h(a, e) = 3$ on P_{max} compared to $h(a, e) = 4$ on P'_{max}. Thus, $l_{max} \leq l'_{max}$. That is, to maximize l_{max} one of F_{Y2} and F_{P2} must be empty.

The hexagons F_1, F_2, F_3, F_4 ensure a hexagonal path from b to d, and as can be seen from Figure 5.3.28 the shortest path from b to d is through c. Thus, it is not necessary for any of F_1, F_2, F_3, F_4 to be empty since they do not have an affect of l_{max}.

Therefore, in this case for any yellow, pink and/or blue hexagon between S_i and S_{i+1} that is not empty in $\mathcal{H}(G)$ l_{max} is bounded by l'_{max}.

Case (b): Any hexagon on S_i may be empty

From Case (a) it can be seen that blue hexagons between S_i and S_{i+1} must be

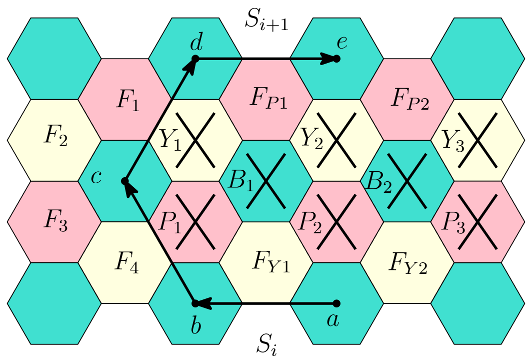


Figure 5.3.28: (a, b) is part of the segment S_i. (d, e) is part of the segment S_{i+1}. Arrowed lines denote the longest hexagonal path from S_i to S_{i+1}. Crossed hexagons are empty hexagons to ensure the longest hexagonal path from S_i to S_{i+1}.

empty to maximize l_{max}. From Case (a), it can also be seen that the only case where yellow and pink hexagons may be non-empty and not shorten the path P'_{max} from S_i to S_{i+1} are when one of F_{Y1}, F_{P1} is non-empty; one of F_{Y2}, F_{P2} is non-empty; and F_1, F_2, F_3, F_4 are non-empty. As can be seen from Figure 5.3.29, for every empty hexagon on S_i on P'_{max} that is empty in $\mathcal{H}(G)$ at most 2 non-blue hexagons on P_{max} between S_i and S_{i+1} can forward the message. Thus, $l_{max} \leq 2\, l'_{max}$.

From Cases (a) and (b), the length of the longest hexagonal path in $\mathcal{H}(G)$ in terms of the length of the longest hexagonal path in $\mathcal{H}(G')$ is at most $l_{max} \leq 2\, l'_{max}$. The broadcast time of G is equal to $l_{max} + 1$. From Lemmas 5.3.7 and 5.3.8, $B_t(G') = l'_{max} + 1 \leq \frac{3}{2}k^2 + 3k + 4$. Since $l_{max} \leq 2\, l'_{max}$, then $B_t(G) \leq 2B_t(G') = 3k^2 + 6k + 8$.

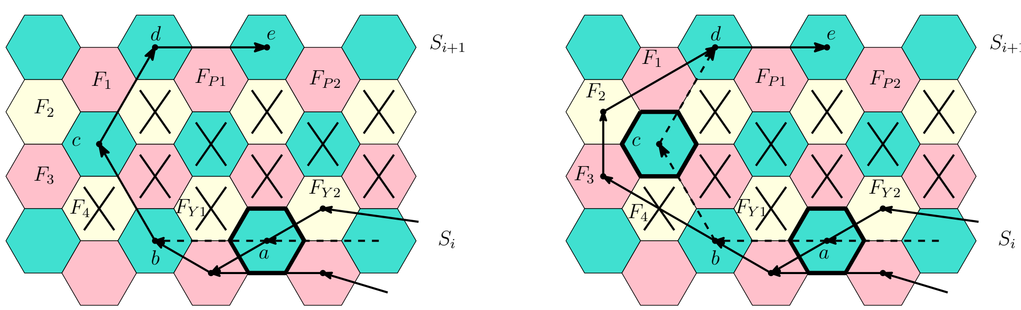


Figure 5.3.29: (a, b) is part of the segment S_i. (d, e) is part of the segment S_{i+1}. Arrowed lines denote the longest hexagonal path from S_i to S_{i+1}. Crossed hexagons are silent hexagons to ensure the longest hexagonal path from S_i to S_{i+1}. Dashed arrowed lines denote edges part of the longest hexagonal path from S_i to S_{i+1} if there were no empty hexagons on P'_{max}.

□

5.4 Beacon-Less Routing using HBLF

In a network, there may arise the case during the lifetime of the network that a packet is unicast from the source node to a specific destination node v_d. Thus, with unknown location information of v_d, the entire network must be flooded in order for v_d to receive the packet. However, if it is known which area of the network v_d is located in, then only that area can be flooded with the packet and save some of the network resources.

It is desirable that the shape of this area that contains v_d is large enough so that even in case of node movement, v_d does not move out of this area. At the same time it is desirable that this area be much smaller than the network size. Thus, we consider the area that contains v_d by a $\frac{\pi}{3}$ sector, where the apex of the sector is the source node s. Thus, to send a a packet specifically intended for v_d, with the knowledge of the $\frac{\pi}{3}$ sector that contains v_d, only the $\frac{\pi}{3}$ sector is flooded instead of the entire network.

Hexagons completely outside of the sector region do not participate in the flooding scheme, while those completely within the sector do. Hexagons that fall on the borders of the sector and may only be partly inside the sector are considered part of the sector and thus, participate in forwarding the message. Thus, HBLF can be used to route data needed in a specific area of the network defined as a $\frac{\pi}{3}$ sector. Note that, the theoretic results obtained in the previous sections are applicable to a $\frac{\pi}{3}$ sector, where the results in a $\frac{\pi}{3}$ sector are roughly $\frac{1}{6}$ times those of for the entire network.

5.5 Summary

In this chapter, we gave an efficient beacon-less algorithm, HBLF, to flood a mobile ad hoc network. HBLF uses a virtual layer of hexagon tiles to achieve delivery of the message. Under the assumption that the network is hexagon connected we show that every node in the network receives the message even in the presence of voids. We present lower and upper bounds on the number of forwarding nodes, denoted β, in a hexagonal shape networks of radius k. That is, $\beta \geq \left\lceil \frac{9\sqrt{3}\left(9k^2+3k+1\right)-168\pi}{28\left(2\pi+3\sqrt{3}\right)} \right\rceil + 1$ (Theorem 5.3.2). When there are no voids in the network, $\beta \leq \frac{3}{2}k^2+3k+\frac{45}{2}$ (Theorem

5.3.3). In presence of voids, we showed that $\beta \leq 5\left(\frac{3}{2}k^2 + 3k + \frac{45}{2}\right)$ (Theorem 5.3.5). Thus, when there are no voids in the network, the upper bound is approximately $\frac{7}{2}$ times the lower bound. The upper bound, where there may be voids in the network, is approximately $5\left(\frac{7}{2}\right)$ times of the lower bound.

We compare the length of the shortest hexagonal path returned by HBLF, denoted l_{HBLF}, to the length of the shortest path in the network, denoted l_{min}, between a source node s and any node v in a hexagonal shape network. If there are no voids in the network then we showed that $\frac{l_{HBLF}}{l_{min}} \leq \frac{4(\sqrt{7}+2)}{3\sqrt{3}} + 2$ (Theorem 5.3.6) is constant. Note that, however, if there are voids in the network then $\frac{l_{HBLF}}{l_{min}}$ is not constant, but rather quadratic in terms of k, the radius of the hexagonal shape network. This can be seen from the upper bound on the broadcast time, which is at most $3k^2 + 6k + 8$ (Theorem 5.3.7).

Chapter 6

Conclusion and Future Work

In this thesis we studied how to efficiently disseminate data in wireless ad hoc network.

In Chapter 3 two centralized algorithms are given to construct a 2-connected (k, r)-dominating set as a base for hierarchical clustering or virtual backbone in order to efficiently route data in wireless sensor networks. The network is modelled as a graph. The first algorithm is for unit disk graphs, which returns a 2-connected (k, r)-dominating set of size at most $2D\beta|OPT|$, where D is the diameter of the graph, β is order of $O(k)$ and OPT is the optimum solution to the 2-connected (k, r)-dominating set. The second algorithm is in general graphs, which returns a 2-connected (k, r)-dominating set of size at most $2D \ln \Delta_k |OPT|$, where Δ_k is the largest cardinality among all k-neighbourhoods in the graph. The work in Chapter 3 appears in [60].

Chapter 4 considers the problem of multiple sink placement in wireless sensor networks such that every sink is within distance k from r sinks. The network is modelled as a graph and hence, this problem is equivalent to total (k, r)-dominating set in graphs. Chapter 4 considers the problem of deriving upper bounds on the total (k, r)-domination number in general graphs of girth at least $2k + 1$ and in random graphs.

For fixed positive integers k and r, in a graph of girth at least $2k+1$ and minimum degree d we showed $\gamma^t_{(k,r)}(G) \leq \dfrac{2nr}{(d-1)^k} + nre^{-\frac{r}{4}}$. This result appears in [59]. The results in random graphs are as follows. For $k = 2$ and non-fixed $p \geq c\sqrt{\dfrac{\log n}{n}}$, where $c > 1$ is a fixed constant, a.a.s $\gamma^t_{(2,r)}(G(n, p)) = r+1$. Upper bounds on $\gamma^t_{(k,r)}(G(n, p))$, where $k = 2$ and $k = 3$ appear in [58]. These results are further generalized for $k \geq 3$

and non-fixed $p \geq k\sqrt[k]{\dfrac{\log n}{n^{k-1}}}$, a.a.s. $\gamma^t_{(k,r)}(G(n,p)) = r+1$. This result appears in [59].

Chapter 5 considers the problem of constructing a beacon-less flooding algorithm, HBLF, to efficiently flood a message in MANETs. HBLF operates based on a virtual hexagonal graph layered over the network, which is modelled as a unit disk graph. HBLF uses the notion that a hexagonal graph is 3-colourable based on which it orders forwarding nodes. In the theoretic analysis, we showed that for a hexagon connected graph HBLF achieves full delivery even in the presence of holes in the network. Lower and upper bounds on the number of forwarding nodes are presented. For a hexagonal shape network of radius k, where blue hexagons are not categorized and are not empty, the number of forwarding nodes determined by HBLF is at most $\beta \leq 3k^2 + 3k + 1$. The lower bound on the number of forwarding nodes is at least $\beta \geq \left\lceil \frac{9\sqrt{3}\left(9k^2+3k+1\right)-168\pi}{28\left(2\pi+3\sqrt{3}\right)} \right\rceil + 1$. These results appear in [61]. In this thesis, we further showed that in presence of blue hexagon categorization and no holes in the network, the number of forwarding nodes is at most $\beta \leq \frac{3}{2}k^2 + 3k + \frac{45}{2}$. In presence of holes in the network $\beta \leq 5\left(\frac{3}{2}k^2 + 3k + \frac{45}{2}\right)$. Note that the upper bound is not dependent on the number of nodes in the network, but rather on the size of the network area. Thus, HBLF is most efficient in dense networks. The theoretic analysis also present a constant dilation factor of $\frac{4(\sqrt{7}+2)}{3\sqrt{3}}$ in networks that do not contain holes. If there are holes present in the network, then the dilation factor is quadratic in the network radius k. This is seen in the upper bound of the broadcast time, $B_t(G)$, where $B_t(G) \leq 3k^2 + 6k + 8$. The results of Chapter 5 are in preparation to be submitted to a journal.

6.1 Future Work

The two algorithms presented in Chapter 3 are centralized algorithms for wireless sensor networks. The approximation ratios given are dependent on the diameter of the graph. Thus, it is desirable to design approximation algorithms, which allows us to decrease this ratio. Also, wireless sensor networks do not have centralized control. Thus, it is of interest to extend these algorithms to distributed scenarios, where nodes make a forwarding decision based on local information. In such a case it is also of interest to give approximation ratios.

In Chapter 4, the upper bound on total (k, r)-domination number in graphs of girth at least $2k + 1$ is a relatively simple expression, but it is not a tight bound. Thus, it is desirable to tighten this bound as well as to weaken the assumption on the girth of the graph and consider the total (k, r)-domination number in general graphs of not necessarily large girth.

The HBLF algorithm in Chapter 5 considers a hexagon connected network in order to achieve full delivery in presence of holes in the network. Thus, it is desirable to relax the assumption that the network is hexagon connected as well as determine any relationship between graph connectivity and hexagon connectivity. The underlying network model considered is a unit disk graph. However, nodes in the network may have different transmission ranges. Thus, it is of interest to construct a beacon-less flooding algorithm in MANETs with heterogeneous transmission ranges.

Bibliography

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[2] N. Alon and J.H. Spencer. *The Probabilistic Method*. John Wiley & Sons, Inc., 1992.

[3] E. Alotaibi and B. Mukherjee. A survey on routing algorithms for wireless ad-hoc and mesh networks. *Computer Networks*, 56(2):940–965, 2012.

[4] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *Proceedings of the 21th IEEE International Conference on Computer Communications (INFOCOM'02)*, pages 32–41, 2002.

[5] V.I. Arnautov. Estimation of the exterior stability number of a graph by means of the minimal degree of the vertices (Russian). *Prikl. Mat. i Programmirovanie*, 11:3–8, 1974.

[6] T.J. Bean, M.A. Henning, and H.C. Swart. On the integrity of distance domination in graphs. *Australasian Journal of Combinatorics*, 10:29–43, 1994.

[7] G. Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57:33–45, 1962.

[8] C. Berge. *Theory of Graphs and its Applications*. Methuen, London, 1962.

[9] C. Berge. *Graphs and hypergraphs*. North Holland, Amsterdam, 1973.

[10] S. Bernstein. On modification of Chebyshev's inequality and of the error formula of Laplace (russian). *Section Mathématique des Annales Scientifique des Institutions Savantes de l'Ukraine*, 1:38–49, 1924.

[11] J. Blum, M. Ding, A. Thaeler, and X. Cheng. Connected dominating set in sensor networks and manets. In *Handbook of Combinatorial Optimization*, pages 329–369. Springer US, 2005.

[12] B. Bollobás. *Random Graphs*. Cambridge University Press, 2001.

[13] A. Bonato and C. Wang. A note on domination parameters in random graphs. *Discussiones Mathematicae Graph Theory*, 28:335–343, 2008.

[14] P. Bose and P. Morin. Online routing in triangulations. In *Proceedings of 10th International Symposium on Algorithms and Computation (ISAAC'99)*, pages 113–122, 1999.

[15] P. Bose, P. Morin, A. Brodnik, S. Carlsson, E.D. Demaine, R. Fleischer, J.I. Munro, and A. Lopez-Ortiz. Online routing in convex subdivisions. In *Proceedings of 11th International Symposium on Algorithms and Computation (ISAAC'00)*, pages 47–59, 2000.

[16] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd Internation Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 48–55, 1999.

[17] Y. Cai, K.A. Hua, and A. Phillips. Leveraging 1-hop neighbourhood knowledge for efficient flooding in wireless ad hoc networks. In *Proceedings of the 24th IEEE International Performance Computing and Communications Conference (IPCCC)*, pages 347–354, April 2005.

[18] Y. Caro and Y. Roditty. A note on the k-domination number of a graph. *Internation Journal of Mathematics and Mathematical Sciences*, 13(1):205–206, 1990.

[19] Y. Caro and R. Yuster. Dominating a family of graphs with small connected subgraphs. *Combinatorics, Probability and Computing*, 9(4):309–313, 2000.

[20] J. Cartigny and D. Simplot. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In *Telecommunication Systems*, pages 189–204, 2003.

[21] P. Casari, M. Nati, C. Petrioli, and M. Zorzi. Efficient non-planar routing around dead ends in sparse topologies using random forwarding. In *Proceedings of IEEE International Conference on Communications (ICC'07)*, pages 3122–3129, 2007.

[22] T.M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete and Computational Geometry*, 16(4):361–368, 1996.

[23] Z. Chan, Y. Ling, and Z. Yiqing. Constructing r-hop 2-connected dominating sets for fault-tolerant backbone in wireless sensor networks. In *Proceedings of the 8th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'12)*, pages 1–4, 2012.

[24] G.J. Chang. The upper bound on k-tuple domination numbers of graphs. *European Journal of Combinatorics*, 29:1333–1336, 2008.

[25] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of MOBICOM'01*, pages 85–96, 2001.

[26] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002.

[27] D. Chen, J. Deng, and P. Varshney. A state-free data delivery protocol for multihop wireless sensor networks. In *Proceedings of IEEE Wireless Communications Networking Conference (WCNC'05)*, pages 1818–1823, 2005.

[28] G. Chen, F. Nocetti, J. Gonzalez, and I. Stojmenovic. Connectivity based k-hop clustering in wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), IEEE Computer Society*, volume 7, pages 2450–2459, 2002.

[29] X. Cheng, M. Ding, H. Du, and X. Jia. Virtual backbone construction in multihop ad hoc wireless networks. *Wireless Communications and Mobile Computing*, 6(2):183–190, 2006.

[30] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.

[31] H. Chernoff. A measure of asymptotic efficiency of tests of a hypothesis based on the sum of ovservations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[32] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1–3):165–177, 1990.

[33] E. Cockayne, S. Goodman, and S. Hedetniemi. A linear algorithm for the domination number of a tree. *Information Processing Letters*, 4(2):41–44, 1975.

[34] E. Cockayne and S. Hedetniemi. Towards a theory of domination in graphs. *Networks*, 7:247–261, 1977.

[35] E. J. Cockayne, B. Gamble, and B. Shepherd. An upper bound for the k-domination number of a graph. *Journal of Graph Theory*, 9(4):533–534, 1985.

[36] M. Couture, M. Barbeau, P. Bose, and E. Kranakis. Incremental construction of k-dominating sets in wireless sensor networks. In *Principles of Distributed Systems*, volume 4305 of *Lecture Notes in Computer Science*, pages 202–214. Springer Berlin/Heidelberg, 2006.

[37] C.C. Craig. On the Tchebychef inequality of Bernstein. *Annals of Mathematical Statistics*, 4:94–102, 1933.

[38] G. Călinescu, I. Măndoiu, P.-J. Wan, and A. Zelikovsky. Selecting forwarding neighbors in wireless ad hoc networks. In *Proceedings of the 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM'01)*, pages 34–43, 2001.

[39] J. Czyzowicz, S. Dobrev, T. Fevens, H. Gonzalez-Aguilar, E. Kranakis, J. Opatrny, and J. Urrutia. Local algorithms for dominating and connected dominating sets of unit disk graphs with location aware nodes. In *LATIN 2008: Theoretical Informatics*, volume 4957 of *Lecture Notes in Computer Science*, pages 158–169. Springer Berlin/Heidelberg, 2008.

[40] F. Dai and J. Wu. On constructing k-connected k-dominating set in wireless networks. *Parallel and Distributed Computing*, 66(7):947–958, 2006.

[41] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Proceedings of IEEE International Conference on Communications (ICC97)*, volume 1, pages 376–380, 1997.

[42] M. Ding, X. Cheng, and G. Xue. Aggregation tree construction in sensor networks. In *IEEE Conference on Vehicular Technology*, pages 2168–2172, 2003.

[43] P.A. Dreyer. *Applications and variations of domination in graphs*. PhD thesis, Department of Mathematics (Rutgers University), 2000.

[44] Q. Fang, J. Gao, L.J. Guibas, V. Silva, and L. Zhang. Glider: Gradient landmark-based distributed routing for sensor networks. In *Proceedings of the 24th Conference of the IEEE Communication Society (INFOCOM'05)*, pages 339–350, 2005.

[45] J. F. Fink and M. S. Jacobson. n-domination in graphs. *Graph Theory with Applications to Algorithms and Computer Science*, pages 283–300, 1985.

[46] M. Fischermann and L. Volkmann. A remark on a conjecture for the (k, p)-domination number. *Utilitas Mathematica*, 67:223–227, 2005.

[47] P. Flach and L. Volkmann. Estimations for the domination number of a graph. *Discrete Mathematics*, 80(2):145–151, 1990.

[48] S. Funke, A. Kesselman, U. Meyer, and M. Segal. A simple improved distributed algorithm for minimum cds in unit disk graphs. *ACM Transaction on Sensor Networks (TOSN)*, 2(3):444–453, 2006.

[49] S. Funke and N. Milosavljevic. Guaranteed-delivery geographic routing under uncertain node locations. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'07)*, pages 1244–1252, 2007.

[50] H. Füßler, J. Widmer, M. Mauve, and H. Hartenstein. A novel forwarding paradigm for position-based routing (with implicit addressing). *IEEE Computer Communications Workshop (CCW'03)*, pages 194–200, 2003.

[51] A. Gagarin and V. Zverovich. A gerneralised upper bound for the k-tuple domination number. *Discrete Mathematics*, 308:880–885, 2008.

[52] X. Gao, W. Wang, Z. Zhang, S. Zhu, and W. Wu. A ptas for minimum d-hop connected dominating set in growth-bounded graphs. *Optimization Letters*, 4(3):321–333, 2010.

[53] X.F. Gao, X.Y. Li, X.F. Zhang, and W.L. Wu. A constant-factor approximation for d-hop connected dominating sets in unit disk graphs. *International Journal of Sensor Networks*, 12(3):125–136, 2012.

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

[55] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20:374–387, 1998.

[56] A. Hansberg, D. Meierling, and L. Volkmann. Distance domination and distance irredundance in graphs. *The Electronic Journal of Combinatorics*, 14:#R35, 2007.

[57] F. Harary and T. W. Haynes. Double domination in graphs. *Ars Combinatoria*, 55:201–213, 2000.

[58] L. Harutyunyan. On placement of multiple sinks in wireless sensor networks. In *Proceedings of the 15th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'14)*. 4 pp, 2014.

[59] L. Harutyunyan. Upper bounds on the total (k, r)-domination number of the random graphs. *Submitted to Discrete Applied Mathematics*, 2014.

[60] L. Harutyunyan and L. Narayanan. Minimum 2-connected distance-k p-dominating set in wireless sensor networks. In *Proceedings of the 8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'12)*, pages 1–6, 2012.

[61] L. Harutyunyan and O. Opatrny. Efficient beacon-less broadcasting in MANETs. In *Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications (AINA'14)*, pages 172–179, 2014.

[62] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Domination in Graphs, Advanced Topics*. Marcel Dekker, Inc., New York, 1998.

[63] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamental of Domination in Graphs*. Marcel Dekker, Inc., New York, 1998.

[64] M. Heissenbüttel, Braun T., Bernoulli T., and M. Wälchli. BLR: beacon-less routing algorithm for mobile ad-hoc networks. *Computer Communications Journal*, 27(11):1076–1086, 2004.

[65] M. A. Henning. Distance domination in graphs. In T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, editors, *Domination in graphs, advanced topics*, pages 321–349. 1998.

[66] M.A. Henning, O.R. Oellermann, and H.C. Swart. The diversity of domination. *Discrete Mathematics*, 161:161–173, 1996.

[67] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath. Flooding for reliable multicast in multi-hop ad hoc networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M'99)*, pages 64–71, 1999.

[68] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[69] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16(4):11–21, 2002.

[70] John Hopcroft and Ravi Kannan. Mathematics for modern computing. `http://www.cs.cornell.edu/jeh/NOSOLUTIONS90413.pdf`.

[71] T.-C. Hou and V.O.K. Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications*, 34(1):38–44, 1986.

[72] M.S. Jacobson and K. Peters. Complexity questions for n-domination and related parameters. *Congressus Numerantium*, 68:7–22, 1989.

[73] C. F. De Jaenisch. *Applications de l'Analyse mathematique an Jen des Echecs.* Petrograd, 1862.

[74] D. Joshi, S. Radhakrishnan, and N. Chandrasekharan. The k-neighbor, r-domination problems on interval graphs. *European Journal of Operational Research*, 79(2):352–368, 1994.

[75] D. Joshi, S. Radhakrishnan, and C. Narayanan. A fast algorithm for generalized network location problems. In *Proceedings of the ACM/SIGAPP Symposium on Applied Computing*, pages 701–708, 1993.

[76] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks.* John Wiley and Sons Ltd., 2007.

[77] B. Karp and H.T. Kung. Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 243–254, 2000.

[78] B.N. Karp. *Geographic Routing for Wireless Networks.* PhD thesis, Harvard University, 2000.

[79] N. Karthikeyan, V. Palanisamy, and K. Duraiswamy. Performance comparison of broadcasting methods in mobile ad hoc network. *International Journal of Future Generation Communication and Networking*, 2(2):47–58, 2009.

[80] H. Kassaei, M. Mehran, L. Narayanan, and J. Opatrny. Local algorithms for backbone formation in ad hoc networks. In *ACM Proceedings of PE-WASUN Conference*, pages 49–57, 2009.

[81] H. Kassaei, M. Mehran, L. Narayanan, and J. Opatrny. Efficient algorithms for connected dominating sets in ad hoc networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'10)*, pages 1–6, 2010.

[82] K. Kawarabayashi, Y. Kobayashi, and B. Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory Series B*, 102(2):424–435, 2012.

[83] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi. Color-based broadcasting for ad hoc networks. In *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–10, 2006.

[84] M. Khabbazian and V. Bhargava. Efficient broadcasting in mobile ad-hoc networks. *IEEE Transactions on Mobile Computing*, 8(2):231–245, 2009.

[85] D. Kim, C.-K. Toh, J.-C. Cano, and P. Manzoni. A bounding algorithm for the broadcast storm problem in mobile ad hoc networks. In *IEEE Conference on Wireless Communications and Networking (WCNC'03)*, pages 1131–1136, 2003.

[86] L. Kleinrock and F. Kamoun. Optimal clustering structures for hierarchical topological design of large computer networks. *Networks*, 10(3):221–248, 1980.

[87] T. Korneffel, D. Meierling, and L. Volkmann. A remark on the $(2,2)$-domination number. *Discussiones Mathematicae Graph Theory*, 28(2):361–366, 2008.

[88] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, pages 51–54, 1999.

[89] F. Kuhn, R. Wattenhofer, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In *Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing*, pages 63–72, 2003.

[90] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'03)*, pages 267–278, 2003.

[91] V. R. Kulli. On n-total domination number in graphs. *Graph Theory, Combinatorics, Algorithms, and Applications*, pages 319–324, 1991.

[92] S. Lee and C. Lee. Broadcasting in mobile ad hoc networks. In Xin Wang, editor, *Mobile Ad-Hoc Networks: Protocol Design*, pages 579–594. InTech, 2011.

[93] D. Li, L. Liu, and H. Yang. Minimum connected r-hop k-dominating set in wireless sensor networks. *Discrete Mathematics, Algorithms and Applications*, 1(1):45–57, 2009.

[94] S. Li. On connected k-domination numbers of graphs. *Discrete Mathematics*, 274:303–310, 2004.

[95] X. Li and Z. Zhang. Two algorithms for minimum 2-connected r-hop dominating set. *Information Processing Letters*, 110(22):986–991, 2010.

[96] Y. Li, D. Kim, F. Zou, and D.-Z. Du. Constructing connected dominating sets with bounded diameters in wireless networks. *WASA*, pages 89–94, 2007.

[97] Y. Li, M. Thai, F. Wang, C.-W. Yi, P.-J. Wan, and D.-Z. Du. On greedy construction of connected dominating sets in wireless networks. *Wireless Communications and Mobile Computing*, 5(8):927–932, 2005.

[98] Y. Li, Y. Wu, C. Ai, and R. Beyah. On the construction of k-connected m-dominating sets in wireless networks. *Journal of Combinatorial Optimization*, 23(1):118–139, 2012.

[99] H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications*, 24:353–363, 2001.

[100] C. Liu and J. Kaiser. A survey of mobile ad-hoc network routing protocols. Technical Report 2003-08, University of Ulm, 2005.

[101] H. Liu, P. Wan, X. Jia, X. Liu, and F. Yao. Efficient flooding scheme based on 1-hop information in ad hoc networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications.*, pages 1–12, 2006.

[102] H.-I. Liu. A distributed intelligent broadcasting protocol for mobile ad hoc networks. *Mobile Networks and Applications*, 14(5):638–648, 2009.

[103] X. Liu, X. Jia, H. Liu, and L. Feng. A location aided flooding protocol for wireless ad hoc networks. In *Mobile Ad-hoc and Sensor Networks*, volume 4864 of

Lecture Notes in Computer Science, pages 302–313. Springer Berlin/Heidelberg, 2007.

[104] Z.L. Liu, F. Tian, and J.M. Xu. Probabilistic analysis of upper bound for 2-connected distance k-dominating set in graphs. *Theoretical Computer Science*, 410:3804–3813, 2009.

[105] W. Lou and J. Wu. Double-covered broadcast (dcb): A simple reliable broadcast algorithm in manets. In *Proceedings of the 23th IEEE International Conference on Computer Communications (INFOCOM'04)*, pages 2084–2095, 2004.

[106] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4):383–390, 1975.

[107] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi, and D.J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(2):59–68, 1995.

[108] W. McCuaig and B. Shepherd. Domination in graphs with minimum degree two. *Journal of Graph Theory*, 13(6):749–762, 1989.

[109] A. Meir and J. W. Moon. Relations between packing and covering numbers of a tree. *Pacific Journal of Mathematics*, 61(1):225–233, 1975.

[110] A. Mohammed, M. Ould-Khaoua, and L. Mackenzie. An efficient counter-based broadcast scheme for mobile ad hoc networks. In *Proceedings of the 4th European Performance Engineering Conference on Formal Methods and Stochastic Models for Performance Evaluation (EPEW'07)*, pages 275–283, 2007.

[111] T. N. Nguyen and D. T. Huynh. Connected d-hop dominating sets in mobile ad hoc networks. *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 1(4):1–8, 2006.

[112] T.N. Nguyen, D.T. Huynh, and J.A. Bolla. Minimum power minimum d-hop dominating sets in wireless sensor networks. In *Wireless Algorithms, Systems, and Applications*, volume 5258 of *Lecture Notes in Computer Science*, pages 176–187. Springer Berlin Heidelberg, 2008.

[113] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in mobile ad hoc network. *International Conference on Mobile Computing and Networking (MobiCom)*, pages 151–162, 1999.

[114] T. Nieberg and J. Hurink. A ptas for the minimum dominating set problem in unit disk graphs. In *Approximation and Online Algorithms*, volume 3879 of *Lecture Notes in Computer Science*, pages 296–306. Springer Berlin/Heidelberg, 2006.

[115] O. Ore. In *Theory of Graphs*, volume 38 of *American Mathematical Society Colloquium Publications*. (American Mathematical Society, Providence, RI), 1962.

[116] A. K. Parekh. Analysis of a greedy heuristic for finding small dominating sets in graphs. *Information Processing Letters*, 39(5):237–240, 1991.

[117] C. Payan. Sur le nobre d'absorption d'un graphe simple (French). *Cahiers du Centre d'Études de Recherche Oprationnelle*, 17(2–4):307–317, 1975.

[118] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st Annual International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'00)*, pages 129–130, 2000.

[119] S. Pleisch, M. Balakrishnan, K. Birman, and R. van Renesse. MISTRAL: Efficient flooding in mobile ad-hoc networks. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*, pages 1–12, 2006.

[120] D. Rautenbach and L. Volkmann. On $\alpha_r\gamma_s(k)$-perfect graph. *Discrete Mathematics*, 270:241–250, 2003.

[121] D. Rautenbach and L. Volkmann. New bounds on the k-domination number and the k-tuple domination number. *Applied Mathematics Letters*, 20(1):98–102, 2007.

[122] B. Reed. Paths, stars and the number three. *Combinatorics, Probability and Computing*, 5:277–295, 1996.

[123] L.A. Sanchis. Experimental analysis of heuristic algorithms for the dominating set problem. *Algorithmica*, 33(1):3–18, 2002.

[124] W. Shang, F. Yao, P. Wan, and X. Hu. Algorithms for minimum m-connected k-dominating set problem. In *Combinatorial Optimization and Applications*, volume 4616 of *Lecture Notes in Computer Science*, pages 182–190. Springer Berlin/Heidelberg, 2007.

[125] H. Sharma and S. Verma. A survey on data flooding detection in mobile ad-hoc network (MANET). *International Journal of Advanced and Innovative Research*, 2(1):217–222, 2013.

[126] M.A. Spohn and J.J. Garcia-Luna-Aceves. Bounded distance multi-clusterhead formation in wireless ad hoc networks. *Ad Hoc Networks*, 5(4):504–530, 2007.

[127] N. Sridharan, V.S.A. Subramanian, and M.D. Elias. Bounds on the distance two-domination number of a graph. *Graphs and Combinatorics*, 18(3):667–675, 2002.

[128] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elim-iniation based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002.

[129] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[130] M. Thai, N. Zhang, R. Tiwari, and X. Xu. On approximation algorithms of k-connected m-dominating sets in disk graphs. *Theoretical Computer Science*, 385(1–3):49–59, 2007.

[131] F. Tian and J.M. Xu. On distance connected domination numbers of graphs. *Ars Combinatoria*, 84:357–367, 2007.

[132] F. Tian and J.M. Xu. A note on distance domination numbers of graphs. *Australasian Journal of Combinatorics*, 43:181–190, 2009.

[133] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in mobile ad hoc network. *Wireless Networks*, 8:153–167, 2002.

[134] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *IEEE Transactions on Computers*, 52(5):545–557, 2003.

[135] D.D Wackerly, W. Mendenhall III, and R.L. Scheaffer. *Mathematical Statistics with Applications*. Duxbury Advanced Series, sixth edition edition, 2002.

[136] P.-J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hod networks. *Mobile Networks and Applications*, 9(2):141–149, 2004.

[137] B. Wang and K.-N. Xiang. On k-tuple domination of random graphs. *Applied Mathematics Letters*, 22:1513–1517, 2009.

[138] C. Wang. The independent domination number of random graphs. *Utilitas Mathematica*, 82:161–166, 2010.

[139] D. Wang, Q. Zhang, and J. Liu. The self-protection problem in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 3(4), 2007. Article No. 20.

[140] F. Wang, M. Thai, and D.-Z. Du. On the construction of 2-connected virtual backbone in wireless networks. *IEEE Transactions on Wireless Communications*, 8(3):1230–1237, 2009.

[141] Y. Wang, X.-Y. Li, and Q. Zhang. Efficient algorithms for p-self-protection problem in static wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(10):1426–1438, 2008.

[142] B. Wieland and A. Godbole. On the domination number of a random graph. *The Electronic Journal of Combinatorics*, 8:#R37, 2001.

[143] M. Witt and V. Turau. Robust and low-communication geographic routing for wireless ad hoc networks. *Wireless Communications and Mobile Computing*, 10(4):486–510, 2010.

[144] J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. In *Proceedings of the 22th IEEE International Conference on Computer Communications (INFOCOM'03)*, pages 2240–2250, 2003.

[145] J. Wu and F. Dai. Efficient broadcasting with guaranteed coverage in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(3):259–270, 2005.

[146] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM)*, pages 7–14, 1999.

[147] J. Wu, B. Wu, and I. Stojmenovic. Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets. *Wireless Communications and Mobile Computing*, 3(4):425–438, 2003.

[148] Y. Wu and Y. Li. Construction algorithms for k-connected m-dominating sets in wireless sensor networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 83–90, 2008.

[149] Y. Wu, F. Wang, M.T. Thai, and Y. Li. Constructing k-connected m-dominating sets in wireless sensor networks. *Military Communications Conference*, October 2007.

[150] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom'01)*, pages 70–84, 2001.

[151] Shun-Ren Yang, Chun-Wei Chiu, and Wei-Torng Yen. A novel convex hull-based flooding scheme using 1-hop neighbor information for mobile ad hoc networks. *Wireless Networks*, 17(7):1715–1729, 2011.

[152] M. Yassein, Q. Abuein, and D. Alzoubi. Dynamic probabilistic flooding in DSR routing algorithm for wireless network. *Journal of Emerging Technologies in Web Intelligence*, 4(4):352–361, 2012.

[153] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[154] H. Zeng, M. Li, H. Liu, and X. Jia. Efficient flooding in mobile ad hoc networks. In Jason N. Turner and Clive S. Boyer, editors, *Ad Hoc Networks: New Research*. Nova Science Publishers, 2008.

[155] Q. Zhang and D.P. Agrawal. Dynamic probabilistic broadcasting in MANETs. *Journal of Parallel and Distributed Computing*, 65(2):220–233, 2005.

[156] Z. Zhang, Q. Liu, and D. Li. Two algorithms for connected r-hop k-dominating set. *Discrete Mathematics, Algorithms and Applications*, 1(4):485–498, 2009.

[157] W. Zhao, H. Wang, and G. Xu. Total k-domination in graphs. *International Journal of Pure and Applied Mathematics*, 35:235–242, 2007.

[158] M. Zorzi. A new contention-based mac protocol for geographic forwarding in ad hoc and sensor networks. In *Proceedings of IEEE International Conference on Communications (ICC'04)*, volume 6, pages 3481–3485, 2004.