ABSTRACT

Title of dissertation: COVERAGE & ROUTING IN DYNAMIC NETWORKS

Ladan Rabieekenari,
Doctor of Philosophy, 2016

Dissertation directed by: Professor John S. Baras
Department of Electrical & Computer Engineering,
Institute for Systems Research

Dynamic networks have become ubiquitous in the current technological framework. Such networks have widespread applications in commercial, public safety and military domains. Systems utilizing these networks are deployed in scenarios influencing critical aspects of human lives, e.g. connecting first responders to command center in disasters, wildlife monitoring, vehicular communication, and health care systems. In this dissertation, we explore two significant aspects of dynamic networks.

In the first part of the dissertation, we study coverage problem in dynamic networks such as public safety networks. Networking infrastructure can partially (or sometimes fully) breakdown during a catastrophe. At the same time, unusual peaks in traffic load could lead to much higher blocking probability or service interruptions for critical communication. Lack of adequate communication among emergency responders or public safety personnel could put many lives at risks. One possible solution to deal with such scenarios is through the use of mobile/portable

infrastructures, commonly referred to as Cells on Wheels (COW) or Cells on Light Trucks (COLT). These mobile cells can effectively complement the existing undamaged infrastructure or enable a temporary emergency network by themselves. Given the limited capacity of each cell, variable and spatially non-uniform traffic across the disaster area can make a big impact on the network performance. Not only judicious deployment of the cells can help to meet the coverage and capacity demands across the area, but also intelligent relocation strategies can optimally match the network resources to potentially changing traffic demands. Assuming that each cell can autonomously change its location, in this dissertation, we investigate such opportunities and constraints. We propose strategies for autonomous relocation of the mobile resource to adapt network coverage and increase the supported user traffic. We demonstrate the performance improvement for several scenarios via simulations using our algorithms.

In practical scenarios, typically there are some areas in the field where mobile base stations cannot move into. Structural obstacles, areas with outstanding water or other hazardous materials, or surfaces with debris are examples of prohibited areas that mobile cells are expected to avoid. Such prohibited areas introduce additional constraints on designing intelligent relocation strategies. We propose a decentralized relocation algorithm that enables mobile cells to adapt their positions in response to potentially changing traffic patterns in a field with such prohibited areas.

In the second part of dissertation, we study routing problem in dynamic networks. Routing is critical when there is no direct link connecting source to its destination. Performance of this algorithm is critical in many different applica-

tions. Two important metrics in routing are delay and throughput. We propose a throughput-optimal routing and scheduling algorithm that improves delay performance by accounting for the network topology. First, we propose algorithm for the fixed topology scenario. We improve delay performance by solving an optimization problem which aims to send packets mostly to greedy neighbors, subject to throughput-optimality constraints. Next, we consider the network with dynamic topology, where routers or links may be added or removed during time. We propose variations of the proposed algorithm for networks with dynamic topology. We identify key design parameters and illustrate the performance of our schemes through simulations.

COVERAGE AND ROUTING IN DYNAMIC NETWORKS

by

LADAN RABIEEKENARI

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Doctor Kamran Sayrafian
Professor Gang Qu
Professor Prakash Narayan
Professor S. Raghu Raghavan, Dean's Representative

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible. First and foremost, I would like to express my deepest gratitude to my advisor and mentor Professor John S. Baras, for his patience, motivation and immense knowledge. His guidance and encouragement helped me in all aspects of my academic life. His wide ranging expertise, vision and incredible energy have been, and will always be an inspiration. He provided a research environment in which I was encouraged to get familiar with different fields and broaden my knowledge. He taught me how to think and connect different domains to each other. This has provided me with a broader view when dealing with research problems.

I owe a debt of gratitude to my mentor Dr. Kamran Sayrafian for defining new research direction and for the many valuable discussions that we had. His mentorship has been truly indispensable. His experience, deep knowledge and insightful comments have guided me along this path tremendously.

I would also like to thank Professor Gang Qu, Professor Prakash Narayan and Professor S. Raghu Raghavan for serving on my dissertation committee. Their feedback during various stages has been instrumental in the improvement and completion of this work. I am greatly thankful to Mrs. Kim Edwards, for her generous and timely assistance with the administrative aspects of my work. She kept all these administrative details to an absolute minimum.

I would like to thank my friends and colleagues in HyNet and SEIL labs for giving me valuable feedbacks. I am specially indebted to my friend and labmate

Shalabh Jain, who has spent countless hours on discussing the problems, giving me insight and kindly passing the experiences he had gained during his graduate research life. His critical feedbacks on many of the projects was very helpful during the last three years. I would also like to thank Tuan Ta for his valuable feedbacks on my research problems and the times he spent for discussion. I would like to thank my office-mates Peixin Gao, Evripidis Paraskevas, Ren Mao, Wentao Luan, Christoforos Somarakis, Dipankar Maity, Yuchen Zhou who made the office an enjoyable place to work. I also like to thank my fellow PhD comrades Gavin Weng, Xiangyang Liu, Baobing Wang, Iakovos Katsipis, Doohyun Sung, Eirini Tsiropoulou, Chrysa Papagianni, Aneesh Raghavan for their valuable feedbacks on my research problems.

My experience at UMD would be incomplete without the wonderful friends I have had the pleasure of sharing my time with. I will forever cherish the fun times spent here with my friends Niloofar, Parastoo, Setareh, Mahshid, Ali, Pouya, Sheikh, Alborz, Melika, Roshanak, Mohammad, Alireza, Farhad, Ramin, Nima, Farid, Negar, Sahar, Azadeh, Naeem, etc. Thank you for sharing many wonderful moments with me. Life would not have been as colorful without your company.

This dissertation research is based upon work supported by the National Science Foundation (NSF), the Air Force Office of Scientific Research (AFOSR) and the National Security Agency (NSA).

None of this would have been possible without the strong foundation of my family. Their unconditional love gave me strength to overcome the many obstacles and challenges.

Ladan Rabieekenari

College Park, MD

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

The past few decades have witnessed an unprecedented shift in the paradigm of digital communication. The notion of communication, from transfer of voice data between humans over fixed networks, has evolved to transmission of large volume of heterogeneous data over mobile network between ‘things’. This change can be attributed to significant progress in fundamental communication technology that enables the advanced protocols and networks, coupled with advances in device technology that enable the connectivity to these networks.

Complex wireless networks are ubiquitous in the world today in commercial, civil and military domains. Examples of this range from the traditional cellular networks to sensor networks for monitoring civil infrastructure or human health, or ad-hoc networks between vehicles on the road, soldiers in a battlefield or emergency responders in a disaster area.

It is clear that the application of these networks is no longer restricted to augmenting the human life for comfort. Instead, they form an essential component of several critical aspects of our lives, ranging from basic health to public safety.

Thus, ensuring availability of these networks and guaranteeing their performance is important to ensure the quality of our lives. Improving these aspects of networks broadly encapsulates the direction of this dissertation.

1.1 Network Challenges

The goal of communication networks is to provide connectivity between the end points for successful transmission of data from a source within the network to a destination. The heterogeneity of the applications of these networks combined with variation in the connecting devices leads to a broad range of architectures and constraints. For example, cellular networks utilize fixed infrastructure to provide connectivity between mobile users which are within the coverage area.

Sensor networks on the other hand utilize the sensor nodes as both infrastructure and sources to produce an aggregate of the state information at a designated sink node. Similarly, ad-hoc networks consisting of mobile nodes, for example vehicular networks or army platoons, create dynamic infrastructure using the nodes and use it to exchange data between mobile sources and destinations. Public safety networks utilize temporary dynamic infrastructure provide connectivity between mobile first responders.

One of the challenges to reliable connectivity, and hence data transmission, is the inherent dynamicity in the network architectures. Dynamicity can be either due to infrastructure dynamics or input dynamics. Infrastructure dynamics refers to the scenarios where the topology of the network changes over time due to changes in the infrastructure or the end points. In these networks nodes may arrive (or depart) and edges may form (or break) arbitrarily. An example of this is networks of platoons in the battle field.

Input dynamics refers to changes in traffic source of the network due to both temporal and geographic events. This includes networks that contain traffic triggered by events, for example sensor networks or surveillance networks or networks in disaster areas. One of the most well-known examples of networks that encapsulates all forms of dynamicity is the public internet, where the infrastructure, traffic, applications and geographical spread are all varying.

Since their initial deployment, significant research effort has been directed towards improving the performance these networks. In this dissertation, we consider two different aspects of networks which have dynamicity either in traffic source or topology. Firstly, we consider the issue of coverage, wherein the goal is to vary the network topology to optimally accommodate time varying traffic distribution. Secondly, we consider the fundamental problem of routing packets from the source to the destination in a multi-hop network, where the topology may be arbitrarily varying over time.

1.2 Supported Traffic and Coverage Concerns in Public Safety Networks

During emergencies, the importance of public safety network becomes clear. Natural disasters such as severe storms, earthquakes, and tornados can destroy base station sites and network connections. Additionally, during these times of crisis, the volume of cell users typically increases, which can overload the service network. Obviously, the combination of these two phenomena makes the situation worse [1].

Since September 11, 2001, and Hurricane Katrina, the Federal Communications Commission (FCC) has taken important steps to improve availability of 911 services and public safety networks in case of emergency. For example the FCCs Public Safety and Homeland Security Bureau (PSHSB) worked on several fronts to improve communications during disasters, including streamlining collection of outage information during times of crisis through the Disaster Information Reporting System, helping ensure that communications workers receive essential personnel credentials during emergencies, working with other federal agencies to improve interoperability among first responders [2].

New traffic hot-spots that typically involve vital life-saving information are a major challenge for the communication network covering the disaster area. The exact locations and magnitudes of these traffic hot-spots within a disaster area are usually not known apriori. As the size of these possible emergency incidents are unpredictable, the capacity requirements to meet the resulting highly variable excess traffic is nearly impossible.

There is a need for a solution which can be quickly deployed and be built spontaneously as devices get connected to each other. The proposed solution should adapt the network to the changes in the traffic in order to provide service to the emergency responders. Two important metrics in these networks are supported traffic and covered area. The main goal of public safety networks is providing service to the users. As a result, we are interested in large supported traffic. At the same time, having larger covered area can increase the supported traffic. We are interested to have a network which covers a large area, so new users joining the network are under network coverage.

1.3 Routing Concerns in Multi-hop Networks

A multi-hop network is a network of computers and devices (nodes) which are connected by communication links. In these networks, many pairs of nodes cannot communicate directly, and must forward data to each other via one or more cooperating intermediate nodes. In multi-hop networks, if two nodes are not directly connected by a communication link, they need to forward packets to each other through intermediate nodes. A source node transmits a packet to a neighboring node with which it can communicate directly. The neighboring node in turn transmits the packet to one of its neighbors, and so on until the packet is transmitted to its ultimate destination. Each link that a packet is sent over is referred to as a hop; the set of links that a packet travels over from the source to the destination is called a route or path. Routes are discovered by running a distributed routing protocol on the network. Figure 1.1 shows an example of a multi-hop network.

In general, there will be many potential routes between each pair of nodes in the network. Since these route use different set of links, they will have different performances. Routing algorithm is supposed to choose a path to forward the packets. Two important metrics in routing algorithm are delay and throughput. There is an

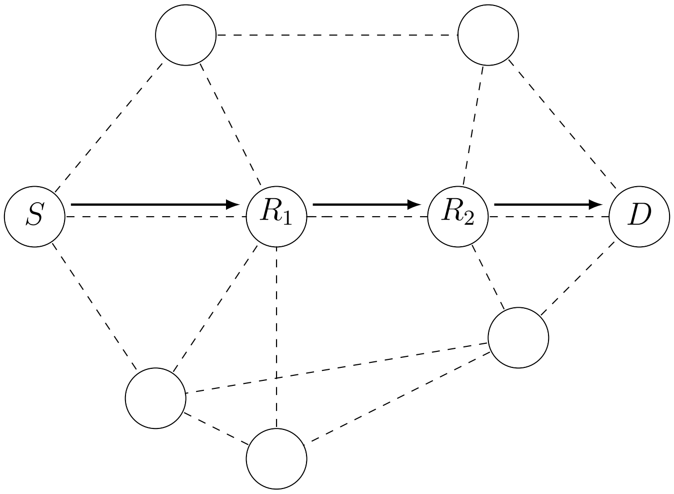


Figure 1.1: A multi-hop network. Node S sends data to node D via cooperating nodes R_1 and R_2

increasing demand for high throughput and low delay scheduling and routing algorithms in both wireless and wired networks. High throughput is critical to respond to increasing demand of different applications. Besides that, delay is very important in real-time applications such as VoIP.

Tassiulas and Ephremides in [3] have proposed a new scheduling and routing algorithm called back-pressure and proved its throughput-optimality. The algorithm can be applied to multi-hop networks, including sensor networks, mobile ad hoc networks (MANETS). A routing/scheduling algorithm is throughput-optimal if it can stabilize any traffic that can be stabilized by any other routing/scheduling algorithm. The back-pressure algorithm [3], is a congestion based routing and scheduling protocol which sends packet along the links with higher queue differential backlog. It has been proven in [3] that back-pressure algorithm is throughput-optimal.

Capacity region of the network is defined as the set of all end-to-end traffic load matrices that can be stably supported under some network control policy. By stability we mean the time average queue length of all queues in the network doesn't go to infinity. A network policy is called throughput-optimal if its capacity region is the same as network capacity region. Tassiulas and Ephremides have proved in [3] that the back-pressure algorithm is throughput optimal for the capacity region of the network denoted as Λ_G. Λ_G is defined as the set of all input rate matrices (λ_i^d) with $\lambda_i^d \neq 0$ if $i \neq d$ and (i, d) is a source destination pair, such that there exists a rate matrix $[\mu_{ij}]$ satisfying the following constraints:

- Efficiency constraints: $\mu_{ij}^d \geq 0, \mu_{ii}^d = 0, \mu_{dj}^d = 0, \sum_d \mu_{ij}^d \leq \mu_{ij}, \forall i, d, j.$
- Flow constraints:$\lambda_i^d + \sum_l \mu_{li}^d \leq \sum_l \mu_{il}^d, \forall i, d : i \neq d.$

As we know the queue length in the destination is 0. In the back-pressure we send the packet to nodes which have smaller queue length. So, in heavy traffic the traffic forces the packets toward the destination. However, in light loads there is not enough packets in the network to push the traffic toward the destination. So it explores all feasible paths between each source and destination. This extensive exploration leads to network stability. However, in light or moderate traffic the back-pressure may lead to unnecessarily long paths and routing-loops. As a result, the back-pressure algorithm has poor end-to-end delay performance.

Greedy routing is another routing algorithm in which each base station sends its packets to a neighbor which is closer to the destination in comparison to itself. Since the path results in a decreasing distance to the destination, these paths are loop free. Loop free paths results in better delay performance in comparison to back-pressure algorithm. However, in this method there is no consideration of throughput.

We are interested in a routing algorithm which has good delay performance like greedy routing and is throughput-optimal like back-pressure.

1.4 Contributions of Dissertation

In public safety problem, we propose a solution through the use of mobile/portable infrastructures commonly referred to as Cells on Wheels (COW) or Cells on Light Trucks (COLT) besides the existing infrastructure. These mobile cells can effectively complement the existing undamaged infrastructure or enable a temporary emergency network by themselves. We introduce a new angle to address user coverage concerns in public safety networks. Instead of deploying COWs on pre-specified locations in the target field, we propose distributed adaptive deployment of COWs. Given the limited capacity of each cell, variable and spatially non-uniform traffic across the disaster area can make a big impact on the network performance. Not only judicious deployment of the cells can help to meet the coverage and capacity demands across the area, but also intelligent relocation strategies can optimally match the network resources to potentially changing traffic demands. Assuming that each cell can autonomously change its location, we formulate the novel relocation optimization problem for maximization of total covered area subject to capacity constraints of base stations. Capacity constraint ensures any user within coverage range of any base station is supported. Maximizing coverage area will result in covering new users. Then, we propose an adaptive self-deployment algorithm to solve the formulated problem. Our simulations show an average improvement of approximately 35% in the supported traffic compared to static uniform deployment.

In real scenarios, these mobile base stations cannot freely relocate to all points within the target field. Structural obstacles, areas with outstanding water or other hazardous materials, or surface with debris are examples of prohibited areas where mobile cells are expected to avoid. Such prohibited areas introduce additional constraints on designing an intelligent relocation strategy. We propose a decentralized relocation algorithm that enables mobile cells to adapt their positions in response to potentially changing traffic patterns in a field with prohibited areas. As shown by our simulations, this autonomous network of mobile cells offers considerable improvement in terms of the supported traffic.

Next, we propose a throughput-optimal routing and scheduling algorithm that improves delay performance by greedy embedding of the network in hyperbolic space. We improve delay performance by solving an optimization problem, which aims to send packets mostly to greedy neighbors, subject to throughput-optimality constraints. The algorithm that solves this optimization problem has a design parameter M. We study the effect of M on delay performance analytically. We validate our theoretical results via simulations and demonstrate that the proposed algorithm improves the delay performance. In real scenarios, nodes may join and leave the network. For instance, consider a sensor network in which sensors are not moving, topology of the network changes when sensors run out of battery or when a new sensor joins the network. We propose an adaptation of the algorithm that has good delay and throughput performance when topology of the network changes slowly.

1.5 Organization of the Dissertation

This dissertation is organized into three primary parts (three chapters). The first two chapters address the coverage problem. The last chapter addresses the routing problem.

In Chapter 2, we formulate the problem of adapting coverage subject to increasing supported traffic in public safety networks when base stations can freely relocate to all points within the target field. We propose a decentralized relocation algorithm that adapts the network coverage in order to increase the supported users traffic. Our simulations show an average improvement of approximately 35% in the supported traffic compared to static uniform deployment.

In Chapter 3, we formulate the problem of adapting coverage subject to increasing supported traffic in public safety networks when these mobile cells might not be able to freely relocate to all points within the target field. We propose a decentralized relocation algorithm that enables mobile cells to adapt their positions in response to potentially changing traffic patterns in a field with prohibited areas. As shown by our simulations, this autonomous network of mobile cells others considerable improvement in terms of the supported traffic.

In Chapter 4, we study routing problem in multi-hop network. We propose a solution to improve delay performance in the network while maintaining queue stability. The algorithm that solves this optimization problem has a design parameter M. We study the effect of M on delay performance analytically. We validate our theoretical results via simulations and demonstrate that the proposed algorithm improves the delay performance. Then, we consider a network in which the topology of the network may change slowly. We propose an adaptation of the proposed algorithm to adapt to these changes. We validate our theoretical results via simulations and demonstrate that the proposed algorithm improves the delay performance.

In Chapter 5, we conclude the dissertation and discuss some future directions.

CHAPTER 2

Autonomous Relocation Strategies for Cells on Wheels in Public Safety Networks

2.1 Overview

Emergency scenarios such as natural or man-made disasters are typically characterized by unusual peaks in traffic demand caused both by people in the disaster area as well as the first responders and public safety personnel [1, 4]. Such traffic hot-spots that typically involve vital life-saving information are a major challenge for the communication network covering the disaster area. The exact locations and magnitudes of these traffic hot-spots within a disaster area are usually unknown apriori. As the size of these possible emergency incidents are unpredictable, estimating the capacity requirements to meet the resulting variable excess traffic is nearly impossible.

Designing the communication network for peak traffic is clearly inefficient and prohibitively expensive due to the large peak-to-average traffic ratio [1, 4]. In addition, an over-provisioned network infrastructure itself may be subject to break-downs or target of attacks; and therefore, not able to address the peak communication needs during emergencies. A reasonable solution to this problem is using a set of mobile base stations that can be quickly deployed to service the excess traffic during the disaster recovery.

A portable cell site - a cell on light truck (COLT) or a cell on wheels (COW) - can be used to augment the remaining communication infrastructure and keep first responders connected to their command centers. Cellular antennas that are attached to a pneumatic mast on a COLT or COW provide additional mobile connection points. By properly deploying these mobile connection points, we can create a temporary network to support first responders need and manage critical public safety communication throughout the disaster area. Such mobile networks that can be easily deployed, configured and adapted, offer an ideal solution to any disaster response effort. These networks would allow public safety personnel and agencies to maintain communication connectivity throughout their operation.

In this chapter, we propose an adaptive self-deployment algorithm where base stations use to autonomously relocate and maximize network coverage subject to their capacity limits. Our algorithm is a sub-optimal solution to a stochastic optimization problem that aims to maximize network coverage subject to capacity constraints. We assume that each base station has access to information about the location of its neighboring base stations and their capacity demand.

2.1.1 Related Work

The base station deployment or positioning problem has been well studied in the past. Simulated annealing [5], genetic approaches [6], greedy algorithms [7], linear programming [8] and evolutionary algorithms [9] are among different approaches used to solve this problem. It has been shown that the identification of the globally optimum base stations deployment in a network is far too complex to be solved computationally (i.e. an NP-hard problem) [10]. In practice, most of the system parameters required to find such an optimal solution is unknown. In addition, the optimal positions could change due to the variations in traffic demand or its spatial distribution. As a result, it is natural to assume that adaptive positioning of mobile base stations based on instantaneous traffic distribution is also an NP-hard problem.

There have been serveral studies regarding providing network coverage in disasters. One possible solution is roaming the public saftey traffic on other commercial networks [11–14]. This solution has remedies such as security and different network requirements between public safety and commercial network users which needs to be implemented by commercial network providers [15]. Besides that, the network infrastructure is still subject to break-downs or target of attacks. In [16–18] the authors proposed using a portable self-configurable cellular system to assist with damaged or destroyed network infrastructure in emergencies or other natural disasters. In [19,20], the authors proposed using UAVs to provide connectivity. However, the deployment phases in all proposed approaches were not considered to be autonomous or adaptive. As a result, when the spatial distribution of traffic changes, the network may fail to adequately meet the traffic demand at various locations on the field. In [21, 22], the authors have also proposed an adaptive relocation algorithm to meet the capacity requirements of the traffic; but, their proposed algorithm is centralized and requires knowledge of traffic distribution across the field. The autonomous adaptive relocation problem in which each mobile node has local information about the location of traffic sources in its coverage area has been considered in [23,24]. There, the authors proposed a distributed algorithm for adaptive relocation of wireless access points in order to minimize the total transmission power. However, minimizing total transmission power does not guarantee increasing the total coverage area or total supported traffic. In addition, in their algorithm each node requires information about the location of traffic sources within its coverage area at each step in order to calculate its new location. This will incur a large overhead for their proposed algorithm. To the best of our knowledge, there is no distributed algorithm that aims to maximize network coverage subject to capacity constraints.

Autonomous relocation algorithms have been extensively studied in mobile sensor network to improve the total area coverage [25–30]. However, no consideration has been given to scenarios with non-uniform spatial traffic distribution. In all these proposed approaches, each node constructs its Voronoi polygon in each iteration and chooses a point inside its Voronoi polygon as its new target point. The main idea behind these algorithms is to move each sensor iteratively to a location that improves its local coverage. Once a new location for a sensor is computed, the corresponding local coverage area of the sensor (in the previously constructed Voronoi diagram) is compared to the preceding local coverage area. If the new local coverage area is larger than the preceding one, the sensor moves to the new location; otherwise, it remains in the current position. If the local coverage area by each sensor in an iteration does not exceed a certain threshold, the algorithm is terminated (to ensure a finite number of iterations). These strategies are either based on distance from vertex or distance from the edge of the Voronoi polygon. As illustrated in [25], in some scenarios vertex based algorithms have better performance and vice versa. As a result, the VEDGE algorithm which considers distance from both vertex and edge has the best coverage performance. VEDGE algorithm is a combination of the Minimax-vertex algorithm and Maxmin-edge algorithm. Two candidate points are calculated for each sensor based on these two methods, and the one which provides better coverage is selected as the candidate location for that sensor.

Algorithm 1 represents Maxmin-edge strategy. The idea behind this strategy is that when the sensors are evenly distributed, none of them should be too close to any of its Voronoi edges. The candidate location of a sensor under the Maxmin-edge strategy is a point inside the corresponding Voronoi polygon whose distance from the nearest Voronoi edge is the largest. This point is denoted as Maxmin-edge centroid. As showed in [25], the Maxmin-edge centroid is the center of the largest incircle or excircle of one of the triangles created by three (extended) edges of the polygon. Hence, one can develop an algorithm of complexity $O(e_i^4)$ (which is typically not too high, as noted earlier) to find the Maxmin-edge centroid of a Voronoi polygon

Algorithm 2 represents Minmax-vertex strategy [27]. The idea behind this strategy is that when the sensors are evenly distributed, none of them should be too

Algorithm 1 Algorithm: Finding the Maxmin-edge centroid of the i-th Voronoi polygon

```
for $f = 1, 2, ..., e_i - 2$ do
    for $g = f + 1, f + 2, ..., e_i - 1$ do
        for $h = g + 1, g + 2, ..., e_i$ do
            ▷ calculate $\omega^{f,g,h}$
            if $\omega^{f,g,h}$ is inside the polygon then
                record it
            end if
        end for
    end for
end for
The center of the largest circle is the Maxmin-edge centroid of the polygon
```

close to any of its Voronoi vertices. The candidate location of a sensor under the Minmax-vertex strategy is a point inside the corresponding Voronoi polygon whose distance from the furthest Voronoi vertex is the smallest. This point is denoted as Minmax-vertex centroid. As showed in [27], the Minmax-vertex centroid is the center of smallest enclosing circle of the set of vertices. To find this circle, we only need to find all the circumcircles of any two and any three Voronoi vertices. Among those circles, the one with the minimum radius covering all the vertices is the Minimax circle. The center of this circle is the Minimax point.

Algorithm 2 Finding the Minmax-vertex centroid of the i-th Voronoi polygon

```
for $u = 1, 2, ..., n-2$ do
    for $v = u+1, u+2, ..., n-1$ do
        for $w = v+1, v+2, ..., n$ do
            ▷ calculate $C(V_u, V_v, V_w)$
            if $V$ is inside $C(V_u, V_v, V_w) \quad \forall V \in \mathcal{V}_p$ then
                record it
            end if
        end for
    end for
end for
for $u = 1, 2, ..., n-2$ do
    for $v = u+1, u+2, ..., n-1$ do
        ▷ calculate $C(V_u, V_v)$
        if $V$ is inside $C(V_u, V_v) \quad \forall V \in \mathcal{V}_p$ then
            record it
        end if
    end for
end for
choose the one with minimum radius
```

2.1.2 Summary of Contributions

Our contributions in this chapter can be summarized as follows

- We formulate the novel relocation optimization problem for maximization of total covered area subject to capacity constraints of base stations.
- We propose an adaptive self-deployment algorithm in which base stations are capable of autonomous relocation to increase total covered area while base stations are not overloaded. Each base station uses information about location of its neighboring base stations and their capacity demand.
- We validate the proposed method via simulations for different scenarios and study the sensitivity of the solution to system parameters.

2.1.3 Outline of Chapter

The rest of this chapter is organized as follows. System description and assumptions are provided in Section 2.2. In Section 2.2.1 we derive a coverage model for each base station, which is used for problem formulation. In Section 2.3 the problem formulation is provided. A distributed adaptive relocation algorithm that simultaneously maximizes coverage and supported traffic is presented in Section 2.4. In Section 2.5, we analyze the efficiency of the proposed algorithm through extensive simulations.

2.2 System Model

Consider a set of mobile nodes (i.e. base stations) denoted by $S = \{s_1, s_2, ..., s_N\}$. We assume that these mobile nodes can wirelessly communicate with each other. Let $Q \subset \mathbb{R}^2$ represent the total geographical area (i.e. target field) which we are interested to cover. Let $P_0 = \{p_{0,1}, p_{0,2}, p_{0,3}, ..., p_{0,N}\}$ denote the initial position of these base stations where $p_{0,i} \in Q, \ \forall i \in \{1, 2, .., N\}$.

Each user in Q connects to the base station with the strongest reference signal which is greater than some specified threshold (i.e. receiver sensitivity denoted by η_r). For simplicity, we assume a flat terrain propagation field with shadow fading which has the same distribution over the region. We assume all base stations are using equal power for transmission.

We also assume that there is an interference-coordination mechanism among adjacent base stations; therefore, interference is negligible. For example, Inter-Cell Interference Cancelation algorithms (ICIC) such as dynamic frequency reuse schemes can be used to mitigate inter-cell interference. There is also non-inter-cell coordinated schemes in which each base station uses orthogonal channel [31]. We are also assuming that the spatial distribution of traffic sources in the target field is non-uniform, and slowly variable.

2.2.1 Coverage Model

We define coverage area of a base station as the geographical region where the average received signal strength is greater than or equal to η_r. This corresponds to

50% coverage probability at cell-edge when shadow fading has log-normal distribution. In order to increase reliability of connection in coverage area, we can consider a fade margin η_F which increases coverage probability at cell edge.

Our propagation and channel loss assumptions imply that there exists a D_{max} such that the average received signal strength is greater than $\eta = \eta_F + \eta_r$ for all points at distance less than or equal to D_{max} of each base station. In order to formalize the average total covered area over region Q, we define Voronoi region $V_i = V(p_i)$ as follows:

$$V_i = \big\{q \in Q \mid \mathbb{E}[P_{rx}(p_i, q)] \geq \mathbb{E}[P_{rx}(p_j, q)], \quad \forall j \in \{1, ..., N\} - \{i\}\big\} \tag{2.1}$$

where $P_{rx}(p_i, q)$ is the received signal strength of base station i at point q.

Since all base stations are transmitting using equal power, Voronoi region $V_i = V(p_i)$ will be the set of all points $q \in Q$ such that $\mathbb{E}[L(p_i, q)] \leq \mathbb{E}[L(p_j, q)]$ where $L(p_i, q)$ is the channel loss between point q and p_i in dB. Thus, $\mathbb{E}[L(p_i, q)] = \mathbb{E}[L_s] + \mathbb{E}[L_p(p_i, q)]$ holds where L_s and L_p represent shadow fading and pathloss respectively. As a result, V_i will be the set of all points $q \in Q$ such that $\mathbb{E}[L_p(p_i, q)] \leq \mathbb{E}[L_p(p_j, q)]$. Due to the flat terrain assumption, this is equivalent to $dist(q, p_i) \leq dist(q, p_j)$. As a result, the Voronoi region $V_i = V(p_i)$ is the set of all points $q \in Q$ such that $dist(q, p_i) \leq dist(q, p_j)$ for all $i \neq j, \ i \in S$.

2.3 Problem Formulation

Based on the defined coverage model, if coverage area of a base station does not include a point within its Voronoi region, that point cannot be in coverage area of any other base station. We define the coverage metric as follows:

$$O(p_1, ..., p_N) = \int_Q \max_{i \in \{1,2,...,N\}} f(dist(q, p_i)) d_q = \sum_{i=1}^{N} \int_{V_i} f(dist(q, p_i)) d_q \tag{2.2}$$

Where $f(x)$ is equal to 1 if $x \leq D_{max}$ otherwise $f(x) = 0$.

Variable spatial traffic implies that, the amount of traffic in the coverage area of each base station or equivalently its traffic load is also changing. This could lead to situations where one or more base stations are located in hot-spots; and therefore, cannot meet the traffic demand within their coverage (i.e. become overloaded). If we assume that the total traffic demand throughout the target field is less than the total network capacity (i.e. capacity of a base station multiplied by the number of base stations), then it is imaginable that the overload scenarios faced by few base stations can be overcome by judicious relocation of all base stations in the network. However, in order to enhance the performance, such relocations should increase the total traffic served by the network, with the ultimate objective of meeting the total target area traffic demand.

In this chapter, we propose a strategy where base stations adaptively and autonomously adjust their positions in order to maximize the supported traffic and eliminate the base station overload situations in hot-spot zones. Given the aforementioned traffic constraint, our proposed relocation algorithm also tries to maximize the network coverage area at the same time.

Let P_n denote the locations of base stations at iteration n, we are interested to find a distributed algorithm in which P_n converges to P^* for a given traffic distribution and such that:

$$P^* = \arg\max \quad \sum_{i=1}^{N} \int_{V_i} f(dist(q, p_i)) d_q \tag{2.3}$$

$$s.t. \quad \widehat{\rho_i} < 1 \qquad \forall i \in \{1, ..., N\}$$

Where $\widehat{\rho_i}$ denotes the estimated average capacity demand of base station i which is the sum of the required resources of all users u connected to cell i by a connection function which gives the serving cell i to user u.

$$\rho_i = \sum_{u \in U_i} \rho_{i,u} \tag{2.4}$$

$$\rho_{i,u} = \frac{s_{i,u}}{s} \tag{2.5}$$

$$s_{i,u} = \left\lceil \frac{\sigma_u}{e_{i,u}} \right\rceil \tag{2.6}$$

where U_i denotes the set of users which are supposed to connect to cell i. $s_{i,u}$ denotes the number of resources used by node u. s denotes the total number of available resources at each base station. σ_u represents the required bit rate of user u in order to transmit data. $e_{i,u}$ is the bandwidth efficiency of user u. The $\lceil x \rceil$ represents the minimum integer larger than x. We are interested to propose a relocation algorithm that can achieve autonomous adaptive base station deployment subject to the capacity constraints.

2.4 Autonomous Relocation of Cell on Wheels

Our proposed approach, described in Algorithm 3, makes use of the location and capacity demand of the each base station and the base stations in its neighborhood [32]. It applies an adaptation of simulation optimization algorithm presented in [33], which makes use of feasible direction method to carry out constrained optimization. The basic strategy of the algorithm in [33] is to generate a sequence of feasible and improving solutions. If the magnitude of the constraint function is less than a lower threshold, it means the constraint is well satisfied, then the variables change in order to improve the objective function. If the constraint function is not satisfied and it is greater than an upper threshold, the variables change in order to satisfy the constraint.

Intuitively, Algorithm 3 aims to maximize network coverage while ensuring that base stations are not over utilized. Each base station tries to increase its local coverage, when the capacity constraints of itself and its neighbors are satisfied. We refer to this phase as coverage improvement phase. On the other hand, if the capacity constraint of a base station is not satisfied (i.e. overload situation), it makes a request for help by sending a signal to the neighboring base stations and asking them to get closer. We refer to this phase as load balancing phase. In this phase, the neighboring base stations can relocate closer to the overloaded base stations if they have available capacity. As a result of their moves, some traffic in the coverage area of the overloaded base station have the opportunity to be offloaded onto the neighboring base stations. The sequence of these relocations are expected to improve the overall traffic support throughout the target area. As several neighboring base stations could be in similar situations with varying degrees of excess traffic, the algorithm uses the concept of a virtual force to determine the final direction where an under-loaded base station should move along.

The virtual force exerted by s_j on s_i is denoted as $\overrightarrow{F}_{ji}$, with the direction from s_i to s_j. The final aggregate virtual force on each base station is the vector summation of the virtual forces from all Voronoi neighbors. These virtual forces will result in base stations moving toward areas with heavy traffic demands i.e. traffic hot-spots. If the aggregate virtual force to a node is zero and the node is not over utilized, it will move in a direction that increases its local coverage. If the aggregate virtual force is not equal to zero, then the moving direction of that node will be the same as the direction of the virtual force vector. The computational complexity of the proposed algorithm in load-balancing phase for base station s_i is $O(e_{s_i})$ where e_{s_i} represents the number of Voronoi edges/neighbors of base station s_i. Since typically a node does not have too many Voronoi neighbors, the computational complexity of such an algorithm is not expected to be high.

Authors in [27] have proposed an autonomous and distributed relocation algorithm to improve coverage in mobile sensor networks. The algorithm iteratively updates the location of each node in a way that improves its local coverage in the previously constructed Voronoi polygon. The rationale behind the algorithm is that when the mobile nodes are evenly distributed, none of them should be too far from any of the Voronoi vertices. So a point inside a Voronoi polygon that has the shortest distance from the furthest Voronoi vertex is selected as the candidate destination point to relocate. The following theorem proves that there is no degradation in the local coverage of each base station at each coverage improvement round. Therefore, we can employ the algorithm in [27], in order to calculate the relocation direction of each base station in the coverage improvement phase.

Algorithm 3 Autonomous adaptive deployment algorithm

```
Each base station $s_i$ broadcasts its location $p_{i,t}$ at time $t$ and its capacity demand $\rho_{s_i}$ to its Voronoi neighbors $\mathcal{N}(s_i)$ and then constructs its Voronoi polygon based on the similar information it receives from other base stations
for each $s_j \in \mathcal{N}(s_i)$ do
    $\overrightarrow{u}_{ji} = \frac{p_{j,t} - p_{i,t}}{\|p_{j,t} - p_{i,t}\|}$
    $\overrightarrow{F}_{ji} = \max\{\rho_{s_j} - 1, 0\} \overrightarrow{u}_{ji}$
    $\overrightarrow{F}_i = \overrightarrow{F}_i + \overrightarrow{F}_{ji}$
end for
▷ Each node $s_i \in S$ calculates its new location as follows:
if $\overrightarrow{F_i} \neq \overrightarrow{0}$ and $\rho_{s_i} \leq 1$ then
    ▷ Load-balancing phase
    $< \overrightarrow{D}_i >=< \overrightarrow{F}_i >= \frac{\overrightarrow{F}_i}{|\overrightarrow{F_i}|}$ which is normalized vector $\overrightarrow{F_i}$ to unit length
else if $\overrightarrow{F_i} = \overrightarrow{0}$ and $\rho_{s_i} \leq 1$ then
    ▷ Coverage improvement phase
    if moving to $c_i$ improves local coverage then
        $< \overrightarrow{D}_i >= \frac{c_i - p_{i,t}}{\|c_i - p_{i,t}\|}$
    end if
end if
$p_{t+1,i} = \Pi_i(p_{t,i} + a_{t,i} < \overrightarrow{D}_i >)$
```

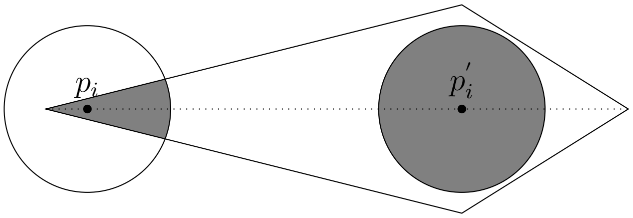


Figure 2.1: Example of a configuration for Theorem 1

Theorem 1. *Consider the set $S = \{s_1, s_2, ..., s_N\}$ of nodes described in the previous section, and let p_i denote the location of node s_i. Let V_i and $\mu(.)$ denote the Voronoi region of node s_i and the area function respectively. $C(p_i, r)$ represents a circle with radius r centered at p_i. If $\exists p_i^{'} \in V_i \quad s.t. \quad \mu(V_i \cap C(p_i^{'}, r)) > \mu(V_i \cap C(p_i, r))$, then*

$$\mu(V_i \cap C(q, r)) \geq \mu(V_i \cap C(p_i, r)) \quad \forall q \in L_{p_i, p_i^{'}}$$

Where $L_{p_i, p_i^{'}}$ denotes the line which endpoints are p_i and $p_i^{'}$. As Figure 2.1 shows, if the local coverage of node s_i at point $p_i^{'}$ is greater than its initial local coverage (which is at point p_i), then its local coverage at any point on the line segment which endpoints are p_i and $p_i^{'}$ is greater than or equal to the initial coverage.

Proof. To prove that the area function does not have any true local minima as the circle moves from p_i toward $p_i^{'}$, we assume the opposite i.e. there exists a local minima. Then, we show that this leads to a contradiction.

Without loss of generality, we assume that the circle is moving on the x axis. Let C_t denote a circle with radius r and a center that is located at $(t, 0)$. It is also important to note, since Voronoi diagrams are intersection of half spaces, they are convex polygons.

Let $W_t = C_t \cap V_i$ denote the intersection of the circle and the convex polygon.

$\mu(W_t)$ is the area of W_t. We assume $\mu(W_t)$ is having a true local minima at $t = t_0$, so there exists t_1 and t_2 such that $\mu(W_{t_1}), \mu(W_{t_2}) \geq \mu(W_{t_0})$ and at least one of the inequalities is strict. For any set of t_0, t_1 and t_2 such that $t_1 < t_0 < t_2$, there exists a $0 < \lambda < 1$ such that $t_0 = \lambda t_1 + (1 - \lambda)t_2$. Denote $\widetilde{C} = \{(x, y, z)|(x - z)^2 + y^2 \leq r^2, (x, y, z) \in \mathbb{R}^3\}$, $\widetilde{V_i} = V_i \times \mathbb{R}$ and $\widetilde{W} = \widetilde{V_i} \cap \widetilde{C}$. $\widetilde{C}$ and $\widetilde{V_i}$ are convex regions in $\mathbb{R}^3$, so $\widetilde{W}$ is also convex. Let G_t denote the plane at $z = t$, so

$$\widetilde{W} \cap G_t = W_t \times \{t\}$$

Consider the convex hull of points in $\widetilde{W} \cap (G_{t_1} \cup G_{t_2})$ in $\widetilde{W}$. The convex hull of a set of points is the smallest convex region containing all points. Due to convexity of $\widetilde{W}$, we conclude the convex hull is a subset of $\widetilde{W}$. Due to convex hull definition, the convex hull is the set of points in X for any $0 < \lambda < 1$. Where X is defined as follows:

$$X = \{\lambda p_1 + (1 - \lambda)p_2 | p_1 \in \widetilde{W} \cap G_{t_1}, p_2 \in \widetilde{W} \cap G_{t_2}\} =$$

$$\{\lambda p_1 + (1 - \lambda)p_2 | p_1 \in W_{t_1} \times \{t_1\}, p_2 \in W_{t_2} \times \{t_2\}\} =$$

$$\{\lambda p_1 + (1 - \lambda)p_2 | p_1 \in W_{t_1}, p_2 \in W_{t_2}\} \times \{\lambda t_1 + (1 - \lambda)t_2\}$$

$$= \{\lambda p_1 + (1 - \lambda)p_2 | p_1 \in W_{t_1}, p_2 \in W_{t_2}\} \times \{t_0\}$$

So we conclude $X \subset G_{t_0}$ and as mentioned earlier $X \subset \widetilde{W}$. So $X \subset (\widetilde{W} \cap G_{t_0}) = (W_{t_0} \times \{t_0\})$. So we conclude

$$\lambda W_{t_1} + (1 - \lambda)W_{t_2} \subset W_{t_0} \tag{2.7}$$

As a result, the following holds

$$\sqrt{\mu(W_{t_0})} \geq \sqrt{\mu(\lambda W_{t_1}) + \mu((1-\lambda)W_{t_1})}$$

$$\geq^* \sqrt{\mu(\lambda W_{t_1})} + \sqrt{\mu((1-\lambda)W_{t_2})}$$

$$= \lambda\sqrt{\mu(W_{t_1})} + (1-\lambda)\sqrt{\mu(W_{t_2})}$$

$$>^{**} \lambda\sqrt{\mu(W_{t_0})} + (1-\lambda)\sqrt{\mu(W_{t_0})}$$

Inequality "*" is valid due to Brunn-Minkowski theorem [34]. Since at least one of $\mu(W_{t_2})$ or $\mu(W_{t_1})$ is strictly greater than $\mu(W_{t_0})$ inequality "**" holds. This leads to a contradiction $\sqrt{\mu(W_{t_0})} > \sqrt{\mu(W_{t_0})}$. Therefore, we can conclude that there is no local minima along the path. □

Based on Theorem 1, if moving base station i to $p_i^{'}$ improves its local coverage, then the local coverage of base station i increases or remains the same if it moves to any point on the line segment between p_i and $p_i^{'}$. Therefore, to guarantee there is no local coverage degradation in the coverage improvement phase, we define the moving direction to be the unit vector connecting current location to the destination point obtained by the algorithm in [27]. The computational complexity in this phase is equal to complexity of the algorithm in [27]. As a result, the complexity is equal to $O(m_{s_i}^4)$, where m_{s_i} represent number of vertices in Voronoi polygon of base station s_i, which is typically not too high.

In reality due to limited transmission power between base stations, a base station may not be able to communicate to all its neighbors. Thus, the set of neighbors is reduced to $\mathcal{L}_i$. Each base station i constructs its local Voronoi LV_i as follows:

$$LV_i = \{q \in Q \quad | \quad dist(q, p_i) \leq dist(q, p_j) \quad \forall j \in \mathcal{L}_i\} \tag{2.8}$$

Consequently, some of the edges of the resultant polygon may be different from those of the exact Voronoi polygon. As a result, the polygons constructed in this case do not necessarily partition the field in the sense that some of them may overlap with each other. This can have a negative impact on the detection of coverage holes. Furthermore, the overlap of the polygons can lead to collision between base stations.

The following Theorem proves we can guarantee coverage improvement in exact Voronoi polygon by limiting the moving distance of each base station in each round.

Theorem 2. *Let R_{com} denote the communication distance between any pair of base stations. V_i represents the exact Voronoi polygon of node i. LV_i denotes the polygon calculated by base station i based on the location information it receives from its neighbors. $C(p_i, r)$ denotes a circle centered at p_i with radius r. Assuming $R_{com} > 2 * R_{cov}$, if base station i relocates to $p_i^{'}$ within circle $C(p_i, \frac{R_{com}}{2} - R_{cov})$ then the local covered area within LV_i is equal to local covered area within V_i.*

Proof. Since $\frac{R_{com}}{2} > R_{cov}$, all points out of circle $C(p_i, \frac{R_{com}}{2})$ are not initially covered by base station i. So

$$C(p_i, R_{cov}) = C(p_i, R_{cov}) \cap C(p_i, \frac{R_{com}}{2}) \tag{2.9}$$

Since $p_i^{'} \in C(p_i, \frac{R_{com}}{2} - R_{cov})$, even after relocating to $p_i^{'}$ none of the points that are out of circle $C(p_i, \frac{R_{com}}{2})$ would be covered by base station i.

$$C(p_i^{'}, R_{cov}) = C(p_i^{'}, R_{cov}) \cap C(p_i, \frac{R_{com}}{2}) \tag{2.10}$$

Then, we will prove:

$$C(p_i, \frac{R_{com}}{2}) \cap LV_i = C(p_i, \frac{R_{com}}{2}) \cap V_i \tag{2.11}$$

It can be concluded from 2.8,

$$V_i \subset LV_i$$

so

$$C(p_i, \frac{R_{com}}{2}) \cap V_i \subset C(p_i, \frac{R_{com}}{2}) \cap LV_i$$

Now we have to prove:

$$C(p_i, \frac{R_{com}}{2}) \cap LV_i \subset C(p_i, \frac{R_{com}}{2}) \cap V_i$$

In order to prove it, we use contradiction. We assume $\exists q \in Q \quad s.t. \quad q \in C(p_i, \frac{R_{com}}{2}) \cap LV_i$ while $q \notin C(p_i, \frac{R_{com}}{2}) \cap V_i$, this means

$$\exists j \text{ s.t. } dist(q, p_i) > dis(q, p_j) \tag{2.12}$$

Based on definition of LV_i, $j \notin \mathcal{L}_i$, so

$$dist(p_i, p_j) > R_{com} \tag{2.13}$$

On the other hand $q \in C(p_i, \frac{R_{com}}{2})$ concludes, $dist(q, p_i) < \frac{R_{com}}{2}$. By applying 2.12, $dist(q, p_j) < \frac{R_{com}}{2}$ can be concluded.

We can rewrite:

$$dist(p_i, p_j) \leq dist(p_i, q) + dist(p_j, q) \leq R_{com} \tag{2.14}$$

where the first inequalities follows from the triangle inequality. Inequality 2.14 contradicts to 2.13. So we conclude 2.11 holds. By applying 2.11 to 2.9 and 2.10, we have:

$$\begin{aligned} &C(p_i, R_{cov}) \cap LV_i = C(p_i, R_{cov}) \cap LV_i \cap C(p_i, \frac{R_{com}}{2}) = \\ &C(p_i, R_{cov}) \cap V_i \cap C(p_i, \frac{R_{com}}{2}) = C(p_i, R_{cov}) \cap V_i \end{aligned} \tag{2.15}$$

and

$$\begin{aligned} &C(p_i^{'}, R_{cov}) \cap LV_i = C(p_i^{'}, R_{cov}) \cap LV_i \cap C(p_i, \frac{R_{com}}{2}) = \\ &C(p_i^{'}, R_{cov}) \cap V_i \cap C(p_i, \frac{R_{com}}{2}) = C(p_i^{'}, R_{cov}) \cap V_i \end{aligned} \tag{2.16}$$

As a result, initial coverage and final coverage within both V_i and LV_i is the same.

□

In Algorithm 3, base station i obtains the coordinate which increases the local coverage within LV_i by using algorithms presented in Section 2.1.1. Then, base station i relocates by size $a_{t,i}$ in the calculated direction. Based on Theorem 1, the new local covered area within LV_i is equal to or greater than the initial local covered area within LV_i. Based on Theorem 2, if $a_{t,i} \leq \frac{R_{com}}{2} - R_{cov}$, the local covered area within V_i has also improved. In reality due to existence of shadow fading, base stations cannot determine exact communication distance R_{com}. Assuming normal shadow fading $SNR_{p_i,p_j} = P_i - L_p(p_i, p_j) + N + \epsilon$, where ϵ is a Gaussian random variable with standard deviation σ on the link connecting BS i to BS j. We define the probability of correct decoding p_c, as the probability that the received SNR is greater than threshold η_r. Thus, for the direct transmission from the BS i to BS j,

$$\begin{aligned} p_c =& \Pr(SNR_{p_i,p_j} > \eta_r) = \Pr(P_i - L_p(p_i,p_j) + N + \epsilon > \eta_r) \\ &= \Pr(\epsilon > \eta_r - N - P_i + L_p(p_i,p_j)) = Q(\frac{\eta_r - N - P_i + L_p(p_i,p_j)}{\sigma}) \end{aligned} \tag{2.17}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{2\pi} e^{-\frac{x^2}{2}}$. We calculate R_{com} for a specific p_c. A larger p_c ensures a larger connection probability at distance R_{com}.

After base station i calculates its moving direction at step t, it moves by $a_{t,i}$ meters toward the calculated direction. $a_{t,i}$ denotes step size sequence for iterative updates of base station i's location. $a_{t,i} = A_i g(step(t,i))$, where A_i is the scaling factor and $g(step(t,i))$ is the decaying factor which gradually decreases from 1 to 0. $step(t,i)$ is initially set to 1 and each time base station i moves, it is incremented by 1. Choice of $a_{t,i}$ can affect the speed of convergence of the algorithm. In order to adjust $a_{t,i}$ to achieve proper convergence speed, we propose to use the following procedure:

- If over the last M relocations of base station i, the moving direction remains the same, then let $A_i = 2a_{t-1,i}$ and set $step(t,i) = 1$.
- If over the last M relocations, the new location of base station i falls out of its corresponding Voronoi polygon , then let $A_i = \frac{a_{t-1,i}}{2}$ and set $step(t,i) = 1$.

In the above procedure, the value of M is also important. Small M could result into incorrect updates due to small amount of information, while a large choice of

M increases the convergence time due to the slow update frequency of $a_{t,i}$. If over the last M relocations, the total relocated distance by base station i is too small or too large, this means $a_{t,i}$ has not been properly chosen for the current network status. This could be either due to the size the area or rapid change in the traffic pattern.

$\Pi_i(.)$ in Algorithm 3, represents the projection function. If s is in load-balancing phase, and if $p_{t,i} + a_{t,i} < \overrightarrow{D}_i >$ falls out of the Voronoi polygon of base station i, then $\Pi_i(p_{t,i} + a_{t,i} < \overrightarrow{D}_i >)$ will be projected in the polygon. Otherwise, if $p_{t,i} + a_{t,i} < \overrightarrow{D}_i >$ falls out of the line segment whose endpoints are c_i and $p_{t,i}$, $\Pi_i(p_{t,i} + a_{t,i} < \overrightarrow{D}_i >)$ will be projected onto c_i.

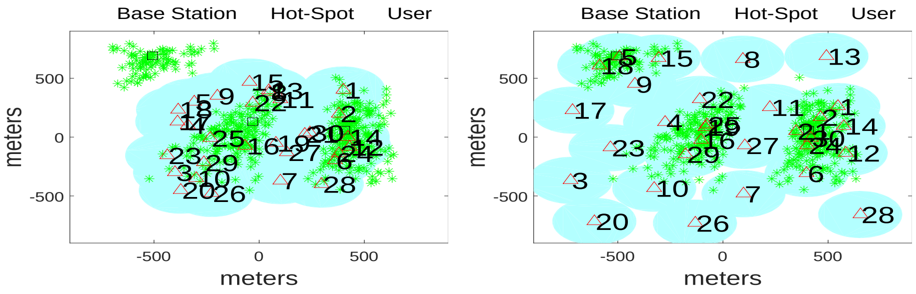
To conserve energy and decrease unnecessary nodes relocation in the network while providing acceptable service quality, we also propose a stopping criterion. If the base station is in coverage enhancement phase and point c_i (in Algorithm 3) cannot improve the coverage by a threshold ϵ_c, it will not move any further. If the base station is in load-balancing phase and the magnitude of $\overrightarrow{F}$ is less than ϵ_{lb}, it will again not move. We can achieve a trade-off between total node relocation and performance by changing ϵ_{lb} and ϵ_{cov}. Larger ϵ_{lb} and ϵ_{cov} will decrease the relocation which is at the cost of worse performance. Due to the nature of our algorithm, there may be situations where a particular base station needs to relocate to opposing directions in order to satisfy the load-balancing and coverage improvements phase (i.e. oscillations). However, such oscillatory behaviors can be avoided by setting up a proper stopping criterion for base stations relocation.

Remark: The problem investigated in this paper is a non convex optimization problem with unknown information about location of users within the field. Thus, the proposed algorithm will not necessarily result in the optimal solution.

2.5 Simulations and Results

Consider a target area of size $1800m \times 1800m$. This target area size is comparable with case 3 of Scenario III cited in the FCC report on the Public Safety Nationwide Interoperable Broadband Network [1]. Several mobile base stations that are connected to a wireless backhaul network are expected to provide communication services to users in this area. It is assumed that each base station has 50 resource blocks of 180KHz in size. It is also assumed that the carrier frequency is 700MHz, channel bandwidth is 10MHz, and transmission power of each base station is equal to 16.39dBm/resource block. The receiver's sensitivity is considered to be -90dBm. Each base station has limited power in communicating with other base stations. It is assumed transmission power of each base station for communication between each other is equal to 33dBm.

We assume that traffic hot-spots are distributed with Poisson point process (PPP), and users (i.e. traffic sources) are generated based on the model in [35]. In this model, first a random location is assigned to each user. Then, each user u is moved toward its closest traffic hot-spot HS_u by a factor of $\beta \in [0, 1]$. So, the user's new location u^{new} is calculated as $u^{new} = \beta HS_u + (1 - \beta)u$. β has a Gaussian distribution with mean $\mu_\beta \in [0, 1]$ and variance $\sigma_\beta = \frac{0.5-|\mu_\beta-0.5|}{3}$. A large μ_β will result in users being closer to hot-spots, while small μ_β will lead to



(a)

(b)

Figure 2.2: First scenario: (a) Initial locations of base stations; (b) Final locations of base stations after execution of Algorithm 3

a uniform distribution of traffic. Each user is generating traffic with the rate of 64kbps, 128kbps or 256kbps based on a uniform distribution.

The path-loss at distance d of base station is modeled as $40\log(d)+30\log(f)+49$ where d is in km and f is in MHz. In addition, shadow fading with a standard deviation of 5dB is also considered. A spatially correlated shadow fading environment with correlation function $r(x) = e^{-\frac{x}{50}}$ was generated as described in [24]. Using the path-loss model and receiver sensitivity, R_{cov} is calculated to be 200m. Mobile base stations employ our proposed algorithm to autonomously relocate and provide better support of traffic within the target area. Node relocation, control signaling exchange and all other updates are carried out using a 60s simulation time-step. We set $a_{i,t} = \frac{200}{step(i,t)}$, a decreasing function of $step(i,t)$ and slowly converging to 0. We set $\epsilon_{lb} = 0.04$ and $\epsilon_{cov} = 1m^2$. To the best of our knowledge, there are no dynamic deployment methodologies that aim to maximize network coverage subject to capacity constraints without any information on spatial traffic distribution across

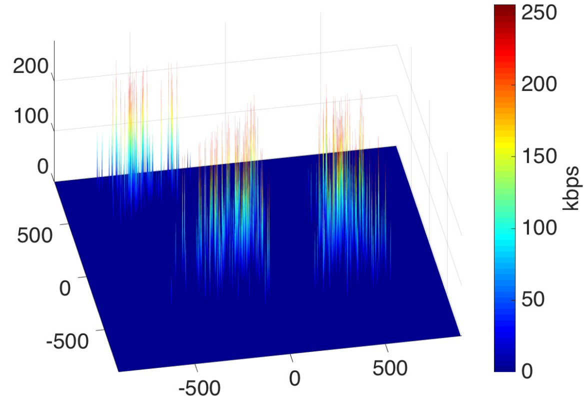


Figure 2.3: Distribution of traffic demand in the first scenario

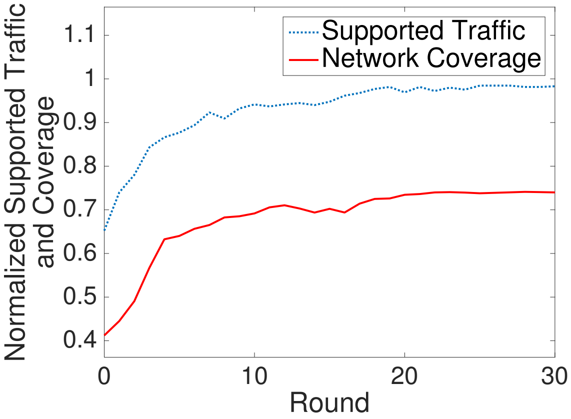
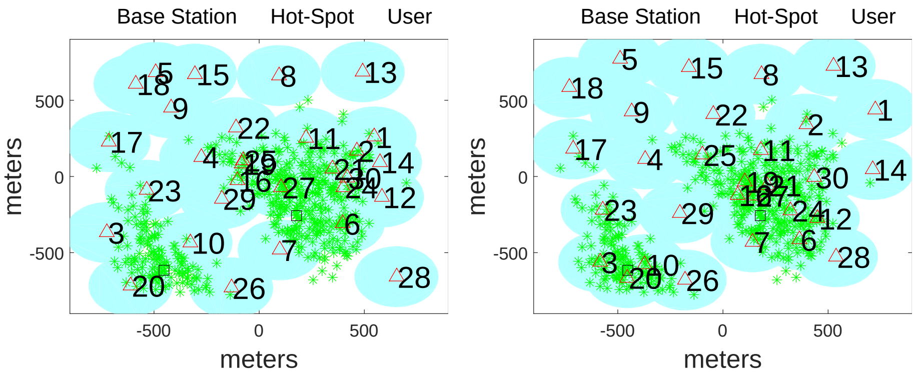


Figure 2.4: Network coverage and supported traffic during execution of Algorithm 3 (first scenario)

the target field. As such, we have shown the performance of our proposed algorithm with respect to static deployment scenarios.

First, we consider the capacity and coverage performance of the network considering an initial random deployment of mobile base stations at the center of a $400m \times 400m$ target field. For example, Figure 2.2(a) shows the initial positions of the base stations (marked by red triangles) along with initial user distribution (marked by green asterisk). Figure 2.3 displays the traffic demand and its spatial distribution over the region. In our simulations, μ_β is equal to 0.63. Given this initial deployment, base station 27 encounters high traffic demand beyond its capacity limit. With the execution of our proposed relocation algorithm, base stations that have available capacity relocate closer to traffic hot-spots. When the capacities of base stations meet the traffic demand within their coverage area, they will continue relocating to expand network coverage within the target field. In this way, traffic hot-spots that were originally outside the coverage area of the initial deployment will get an opportunity to be discovered. The above process continues until all base stations can meet their respective traffic demands and maximum network coverage is achieved. Figure 2.2(b) shows the final base station positions after 35 time-steps. Figure 2.4 shows how network coverage and the total supported user traffic evolve during the execution of our proposed algorithm. As observed, Algorithm 3 results in increasing the supported user traffic from 50% to 98% as well as improving the network coverage from 31% to 71%.

Next, after the base stations converge to their final positions in Figure 2.2(b), we consider a scenario where traffic hot-spots locations change. This is shown in



(a)

(b)

Figure 2.5: Second scenario: (a) Initial locations of base stations; (b) Final locations of base station after execution of Algorithm 3

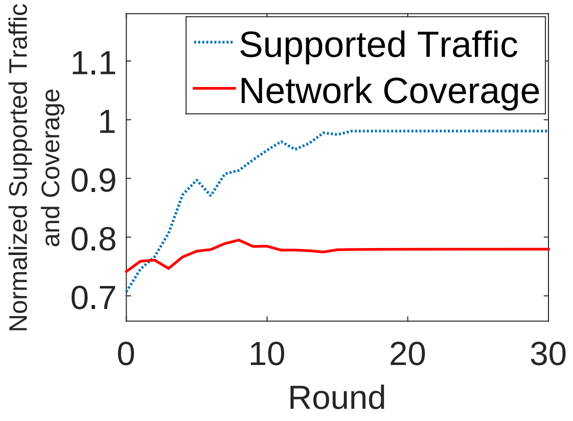


Figure 2.6: Network coverage and supported traffic during execution of Algorithm 3 (second scenario)

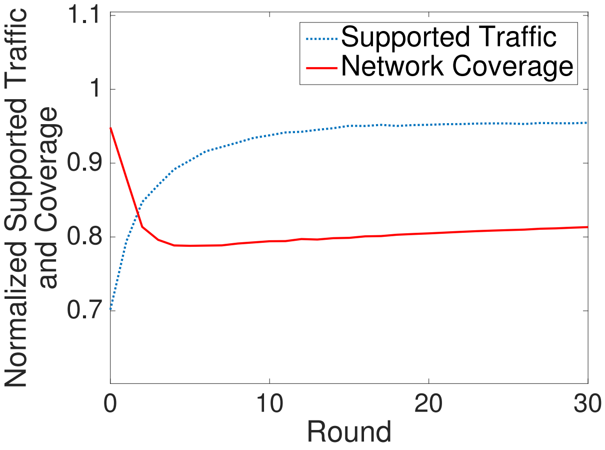


Figure 2.7: Average network coverage and supported traffic during execution of Algorithm 3 (assuming a uniform initial deployment)

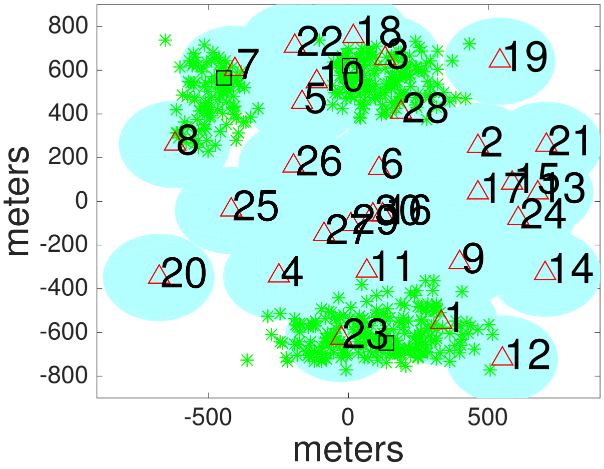
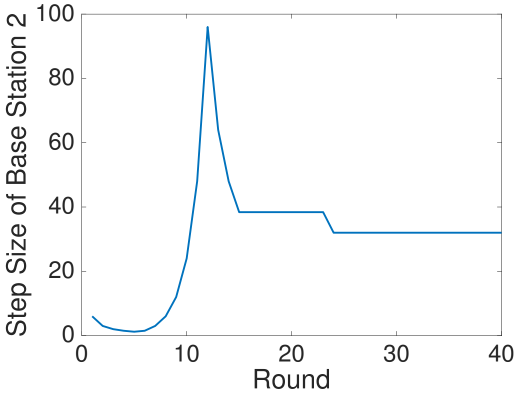
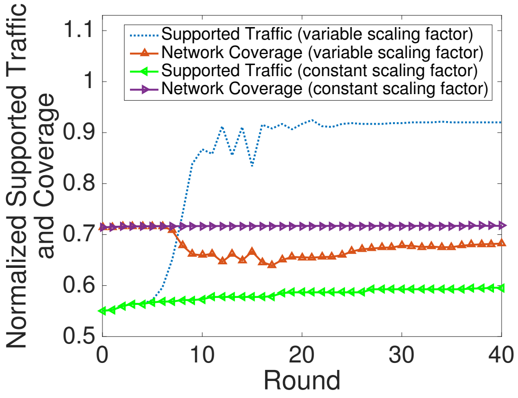


Figure 2.8: Initial location of base stations

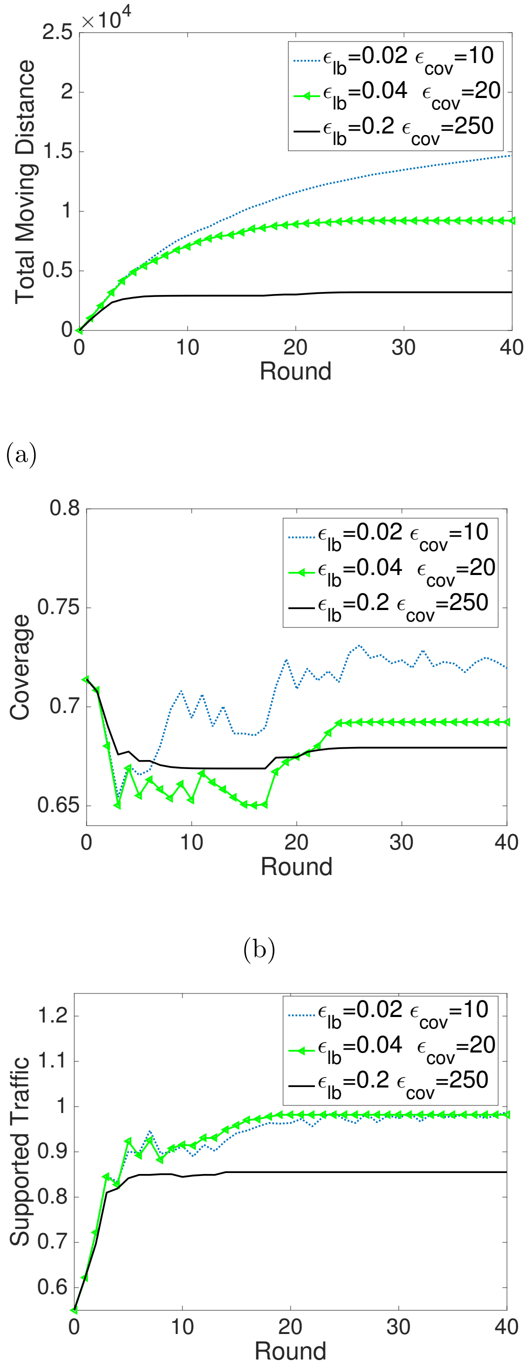


(a)



(b)

Figure 2.9: (a) Step size variations of base station 2 during execution of Algorithm 3 when the step size changes adaptively; (b) Network coverage and supported traffic during execution of Algorithm 3 for both variable and fixed scale factor



(a)

(b)

(c)

Figure 2.10: (a) Total moving distance of base stations up to each round during execution of Algorithm 3 for two different set of stopping criterias; (b) Network coverage during execution of Algorithm 3; (c) supported Traffic during execution of Algorithm 3

Figure 2.5(a). With this new traffic distribution, base stations 1, 3, 7, 23 and 28 will face high traffic demands above its capacity limit. Again, using Algorithm 3, the base stations will autonomously relocate to new positions in order to adapt to new traffic hot-spots and accommodate the corresponding demand. Final base station positions are shown in Figure 2.5(b).

Figure 2.6 depicts changes in the area coverage and supported traffic during the execution of our algorithm. As observed, the supported user traffic increases from 55% to 98%. The trade-off in supporting almost all traffic demand in this scenario is the decrease in the overall network coverage. However, the relocation algorithm will still achieve maximum possible coverage given the spatial distribution of traffic and capacity limits of the base stations.

Next, we investigate the performance of our proposed algorithm by averaging over 100 different scenarios assuming a uniform initial deployment and random spatial traffic demands (i.e. μ_{beta}, number of hot-spots and their location). The results are shown in Figure 2.7. With an initial uniform deployment of base stations, occurrences of traffic hot-spots will cause several base stations to face traffic above their capacity limits. These situations result in a low average supported traffic of only 70%. Using Algorithm 3, the base stations will adaptively relocate to meet non-uniformities in the traffic demand; and therefore, the average supported traffic in the network will increase to 96%.

Next, we study how adaptation of step-size affects the performance of the proposed algorithm in the network. Consider a scenario with initial distribution as depicted in Figure 2.8. We set $a_{i,t} = \frac{6}{step(i,t)}$ which is small in comparison to dimensions of the Voronoi polygons. We assume there is no stopping constraint. M is set to 4, which means each base station keeps track of its relocations during the last 4 rounds. Figure 2.9(a) depicts changes in $a_{2,t}$ during time. As Figure 2.9(a) shows, $a_{2,t}$ is initially equal to 6. Starting from round 2, base station 2 keeps moving in the same direction which means the step-size is small. As a result from round 7, we observe a sudden increase in $a_{2,t}$. In order to measure effectiveness of the proposed adaptation algorithm, we compare coverage and supported traffic when scale factor adaptation is deployed with when scale factor is fixed. As comparison between Figure 2.9(a) and Figure 2.9(b) shows, after there is a sudden improvement in step size, the coverage and supported traffic increase faster which is because the initial step size was too small. As Figure 2.9(b) shows, adaptive scaling factor improves network performance faster than the fixed one. This is due to the fact that 6 meters is too small in comparison to the size of Voronoi polygons, which means it takes a large number of rounds for each base station to reach the desired point.

Next, we study the effect of ϵ_{lb} and ϵ_{cov} on the performance of our proposed algorithm. Assume initial distribution as depicted in Figure 2.8. Figures 2.10(b)-2.10(c) depict how network coverage and supported traffic change for varying values of ϵ_{cov} and ϵ_{lb}. As these figures show, smaller ϵ_{cov} and ϵ_{lb} results in better performance. However, as Figure 2.10(a) illustrates, the total moving distance increases as the threshold decreases. This is because larger threshold will require larger performance improvement to relocate each base station. As a result, the base stations will not move if they cannot improve the performance by the specified threshold. This limitation will degrade the performance while it decreases the total moving

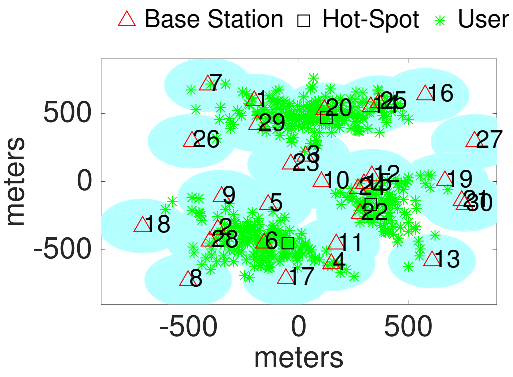


Figure 2.11: Initial locations of base stations

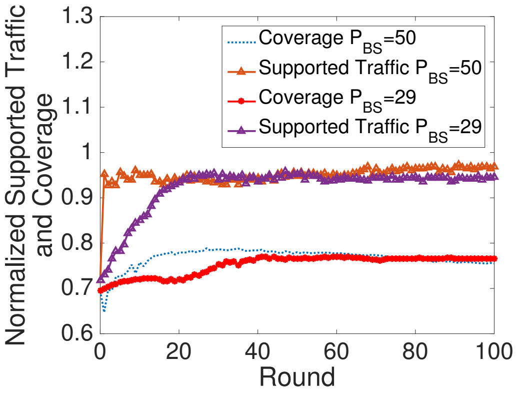


Figure 2.12: Supported traffic and network coverage for varying values of P_{BS} in scenario depicted in Figure 2.11

distance.

Finally, we study effect of limited communication range between base stations on the performance of the proposed algorithm. We compare the performance by considering two different scenarios to study the effect of limited communication range on coverage improvement and load balancing phases. First, we consider the capacity and coverage performance of the network considering an initial random deployment of mobile base stations and user distribution as shown in Figure 2.11. In this scenario, Voronoi neighbors of overloaded base stations are within commu-

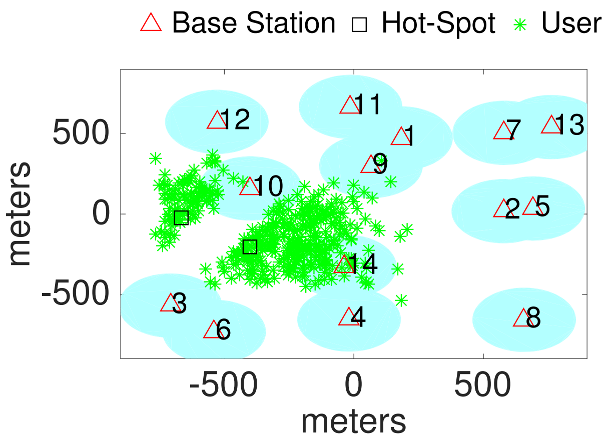


Figure 2.13: Initial locations of base stations

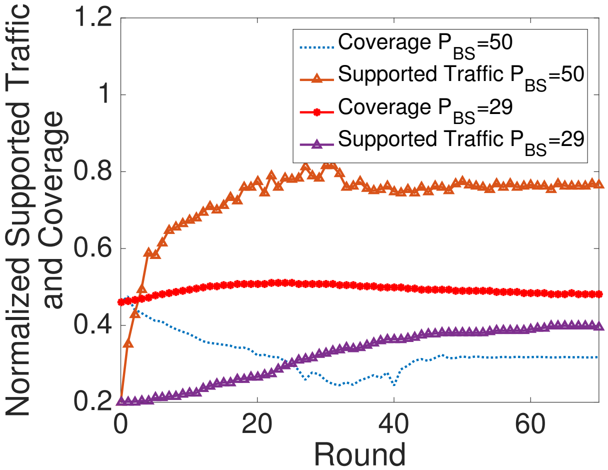


Figure 2.14: Supported traffic and network coverage for varying values of P_{BS} in scenario depicted in Figure 2.13

nication distance of each other, so load balancing phase is not affected by power significantly. As Figure 2.12 shows, coverage performance is not affected by the transmission power between base stations. However, smaller communication range results in slower improvement. This is because, based on Theorem 2, the maximum step size for small communication ranges is smaller than large communication ranges.

Next, we consider the capacity and coverage performance of the network considering an initial random deployment of mobile base stations and user distribution as shown in Figure 2.13. In this scenario there are not enough number of base stations in the network and at most they can cover 54% of the area. So when a node gets overloaded, it does not necessarily have a neighbor which is within its communication distance. In Algorithm 3, if the neighbor nodes of overloaded base stations cannot detect it, they will not relocate to offload the traffic from overloaded neighbor. As a result, as shown in Figure 2.14, in this scenario larger power results in better supported traffic.

2.6 Discussion

Cells on Wheels are a cost effective solution to complement a public safety network during emergencies. The variable nature of the spatial distribution of traffic throughout the target field along with the large peak-to-average traffic ratio necessitates judicious and adaptive deployment of the cells. Assuming, autonomous mobile base stations, we have proposed a distributed relocation algorithm that aims to adaptively enhance network coverage subject to base stations capacity limit. The execution of the algorithm will effectively adapt the overall network coverage in order to maximize the supported user traffic. Simulations show that substantial gain in performance can be achieved under typical usage scenarios.

In this chapter, it was assumed that the base stations can move toward any direction without any restrictions. If there are obstacles or prohibited areas within the target field, the relocation algorithm has to consider additional constraints. In Chapter 3, we address this problem. Besides that, it was assumed all base stations are mobile. However, as we discuss in future work, there may be scenarios in which the overall network consists of a combination of mobile and static base stations. We had also assumed all base stations are transmitting with the same fixed power. We discuss in future work how having power as another variable will affect the formulation. Another direction which is discussed in future work is keeping connectivity in the network as the base stations relocate.

CHAPTER 3

Autonomous Relocation Strategies for Cells on Wheels in Environments with Prohibited Areas

3.1 Overview

As discussed in Chapter 2, networking infrastructure can partially (or sometimes fully) breakdown during a disaster. A mobile communication infrastructure composed of Cells on Wheels was proposed as a viable solution to complement or replace the existing static infrastructure. Two important metrics for any cellular network is supported traffic and area coverage. We are interested in a deployment which has large coverage and large supported traffic, however, when we have limited number of resources it may not be possible to have both full coverage and full supported traffic. In Chapter 2, we proposed an algorithm which maximizes covered area subject to capacity constraints in the network. However, we considered the scenario in which base stations can move to any point within the field. In practice, these mobile cells might not be able to freely relocate to all points within the target field. Structural obstacles, areas with outstanding water or other hazardous materials, or surfaces with debris are examples of prohibited areas where mobile cells are expected to avoid. Such prohibited areas introduce additional constraints on designing an intelligent relocation strategy.

3.1.1 Related Work

A stated in 2.1.1, autonomous relocation algorithms have been extensively studied in mobile sensor network to improve the total area coverage. However, no consideration of variable non-uniform traffic have been taken into account. In [36], the authors proposed a strategy to maximize the sensing coverage in mobile sensor network. The proposed algorithm is an iterative algorithm. In each iteration, each sensor moves to a new location inside its Voronoi polygon which increases local coverage. To this end, a gradient-based nonlinear optimization approach is utilized. In each round of the coverage algorithm, every sensor moves to an optimum target point within its Voronoi polygon in order to maximize the local coverage within the polygon. The problem is efficiently solved using some nonlinear programming techniques.

Consider a region defined by k relations $h_j(p_{i,t}, q) \leq 0$ for $j = 1, ..., k$. By concatenation of the boundary functions $h_j(p_{i,t}, q)$ as $h(p_{i,t}, q) = [h_1(p_{i,t}, q), ..., h_k(p_{i,t}, q)]^T$, this region can be represented by $h(p_{i,t}, q) \leq 0$. Denote the boundary of $\mu(p_{i,t})$ by $\partial\mu(p_{i,t})$. The boundary of the region corresponds to the equalities in the above formulation. Note that this boundary has k segments, where each segment can be expressed as:

$$\partial_j \mu(p_{i,t}) = \{q \in \mu(p_{i,t}) : h_j(p_{i,t}, q) = 0\} \tag{3.1}$$

Consider the following integral function over the region:

$$F(p_{i,t}) = \int_{\mu(p_i,t)} z(p_{i,t}, q)dq \tag{3.2}$$

where $z(.,.)$ is a given function. The gradient of $F(p_{i,t})$ with respect to $p_{i,t}$ can be computed as [36]:

$$\begin{aligned} \nabla_{p_{i,t}} F(p_{i,t}) &= \int_{\mu(p_{i,t})} \nabla_{p_{i,t}} z(p_{i,t}, q)dq \\ -\sum_{j=1}^{k} \int_{\partial_j \mu(p_{i,t})} &\frac{z(p_{i,t}, q)}{\|\nabla_q h_j(p_{i,t}, q)\|} \nabla_{p_{i,t}} h_j(p_{i,t}, q)dq \end{aligned} \tag{3.3}$$

At each iteration base station i moves to location $p'_{i,t}$ which maximizes $F(x)$. Each base station uses an iterative algorithm to find the optimal position within its Voronoi polygon. At each iteration, each base station calculates the gradient of the objective function. Then, by using a line search algorithm it moves in the gradient direction. This calculation stops when the difference between the coverage between two consecutive round is less than a pre-specified threshold. The local coverage area of each base station can be formulated as

$$J(p_{i,t}) = \int_{\mu(p_{i,t})} dq \tag{3.4}$$

Here $z(p_{i,t}, q) = 1$. Note that a polygon with m edges is represented as $Hq - K \leq 0$, where $H_{m\times 2}$ and $K_{m\times 1}$ are matrices with real entries. The sensing disk centered at $p_{i,t}$ can be also represented as $\|q - p_{i,t}\|^2 - R_S^2 \leq 0$. $Hq - K \leq 0 \quad \cap \quad \|q - p_{i,t}\|^2 - R_S^2 \leq 0$ represents the intersection of the polygon and disk.

Gradient of 3.4 can be calculated by 3.3. Since $z(p_{i,t}, q) = 1$, the first term in 3.3 vanishes. Also $Hq - K \leq 0$ does not depend on $p_{i,t}$. As a result the only term that is not equal to zero belongs to $h_{m+1}(p_{i,t}, q) = \|q - p_{i,t}\|^2 - R_S^2$ which represents the sensing disk.

$$\nabla_{p_{i,t}} h_{m+1}(p_{i,t}, q) = -2(q - p_{i,t}) \tag{3.5}$$

$$\nabla_q h_{m+1}(p_{i,t}, q) = 2(q - p_{i,t}) \tag{3.6}$$

Therefore, the gradient of $J(p_{i,t})$ with respect to the position of the sensor is given by:

$$\nabla_{p_{i,t}} J(p_{i,t}) = \int_{\partial\mu(p_{i,t})} \frac{q - p_{i,t}}{\|q - p_{i,t}\|} dq \tag{3.7}$$

where $\partial\mu(p_{i,t})$ is the portion of the perimeter of the sensing disk which is inside V_i.

Given the direction $\nabla_{p_{i,t}} J(p_{i,t})$, a line search procedure determines the optimal step size for maximizing the objective function in that direction. This process will be done in several iterations to get close to the optimal solution.

3.1.2 Summary of Contributions

Our contributions in this chapter can be summarized as follows:

- We formulate the novel relocation optimization problem for maximization of total covered area subject to capacity constraints of base stations in existence of prohibited areas.
- We propose an adaptive self-deployment algorithm in which base stations are capable of autonomous relocation to increase total covered area while base stations are not overloaded. Each base station uses information about location of its neighboring base stations and their capacity demand.

- We validate the proposed method via simulations for different scenarios and study the sensitivity of the solution to system parameters. .
- We compare the proposed method with the method in Chapter 2.

3.1.3 Outline of Chapter

The rest of this chapter is organized as follows. System description and assumptions are provided in Section 3.2. In Section 3.3, problem formulation is provided. In Section 3.4, a distributed adaptive relocation algorithm that simultaneously maximizes coverage and supported traffic is presented. In Section 3.5, we analyze the efficiency of the proposed algorithm through extensive simulations.

3.2 System Model

We consider same assumptions and coverage model as in section 2.2. We also assume the base stations cannot move into specific areas in the field. Let $Q_F \subset Q$ represent the total geographical area (i.e. target field) in which the base station can move into.

3.3 Problem Statement

We need a strategy where mobile base stations adaptively and autonomously adjust their positions in order to increase the supported traffic by eliminating the base station overload situations in traffic hot-spot zones. Given the aforementioned traffic constraint, our proposed relocation algorithm also tries to maximize the network coverage area at the same time. Let P_n denote the locations of base stations at iteration n, we are interested to find a distributed algorithm in which P_n converges to P^* for a given traffic distribution and such that:

$$P^* = \underset{p_1,...,p_M \in Q_F}{\arg\max} \quad \sum_{i=1}^{N} \int_{V_i} f(dist(q, p_i))dq \tag{3.8}$$

$$s.t. \quad \widehat{\rho_i} \leq 1 \qquad \forall i \in \{1, ..., N\}$$

Where $\widehat{\rho_i}$ denotes the estimated average capacity demand of base station i which is the sum of the required resources of all users u connected to cell i by a connection function which gives the serving cell i to user u.

$$\rho_i = \sum_{u \in U_i} \rho_{i,u}, \quad \rho_{i,u} = \frac{s_{i,u}}{s}, \quad s_{i,u} = \left\lceil \frac{\sigma_u}{e_{i,u}} \right\rceil$$

where U_i denotes the set of users which are supposed to connect to cell i. $s_{i,u}$ denotes the number of resources used by node u. s denotes the total number of available resources at each base station. σ_u represents the required bit rate of user u in order to transmit data. $e_{i,u}$ is the bandwidth efficiency of user u. The $\lceil x \rceil$ represents the minimum integer larger than x. We are interested to propose a relocation algorithm that can achieve autonomous adaptive base station deployment subject to the capacity constraints.

3.4 Autonomous Relocation Strategy in Existence of Prohibited Areas

Our proposed approach is an iterative algorithm; where in each iteration every base station first broadcasts its location along with the capacity demand from users in its coverage area to other neighboring base stations [37]. Each base station then uses this information to calculate its new location. The new location is calculated through an adaptation of simulation optimization algorithm presented in [33]. The basic strategy of the algorithm in [33] is to generate a sequence of feasible and improving solutions. If the constraint is well satisfied, then the variables change in the direction which improves the objective function. If the constraint function is not satisfied, the variables change in a direction which satisfies the constraint.

Intuitively, the proposed algorithm aims to maximize network coverage while ensuring that base stations can meet their corresponding traffic demand. Each base station tries to increase its local coverage, when the capacity constraints of itself and its neighbors are satisfied. We refer to this phase as coverage improvement phase. On the other hand, if the capacity constraint of a base station is not satisfied (i.e. overload situation), it makes a request for help by sending a signal to the neighboring base stations. We refer to this phase as load balancing phase.

In the load balancing phase, a base station moves in the direction which would result in the fastest offloading of traffic from the overloaded Voronoi neighbors. Since we do not have any information about traffic distribution in the area, we assume traffic sources of each base station is uniformly distributed within its coverage area. Let $\varphi(q)$ denote the estimated traffic density at point q for overloaded areas, we set $\varphi(q)$ for the rest of area to 0. The amount of offloaded traffic from overloaded neighbor base stations when base station i is at $p_{i,t}$ is as follows:

$$G(p_{i,t}) = \int_{\mu(p_{i,t})} \varphi(q) dq \tag{3.9}$$

Here $z(p_{i,t}, q) = \varphi(q)$ which is not a function of $p_{i,t}$. So first term in equation 3.3 will be equal to zero. Based on the definition of $\varphi(q)$, $\varphi(q)$ is nonzero only on the Voronoi edges which are besides overloaded neighbor base stations. As a result by using 3.3, we have:

$$\nabla_{p_{i,t}} G(p_{i,t}) = \\ -\sum_{j=1}^{|\mathcal{O}_i|} \int_{\partial_j \mu(p_{i,t})} \frac{\varphi(q)}{\|\nabla_q h_j(p_{i,t}, q)\|} \nabla_{p_{i,t}} h_j(p_{i,t}, q) dq \tag{3.10}$$

where $\partial_j \mu(p_{i,t})$ is the part of Voronoi polygon V_i which is mutual between base station i and o_j. $h_j(p_{i,t}, q)$ represents the equation of $\partial_j \mu(p_{i,t})$ and it is equal to $(p_{o_j,t} - p_{i,t})^T(q - \frac{p_{i,t}+p_{o_j,t}}{2})$. The gradients can be computed as follows:

$$\frac{\partial h_j(p_{i,t}, q)}{\partial p_{i,t}} = p_{i,t} - q \tag{3.11}$$

$$\frac{\partial h_j(p_{i,t}, q)}{\partial q} = p_{o_j,t} - p_{i,t} \tag{3.12}$$

Then, we get the following:

$$\nabla_{p_{i,t}} G(p_{i,t}) = \sum_{j=1}^{|\mathcal{O}_i|} \int_{\partial_j \mu(p_{i,t})} \varphi(q) \frac{q - p_{i,t}}{\|p_{o_j,t} - p_{i,t}\|} dq \tag{3.13}$$

$\nabla_{p_{i,t}} G(p_{i,t})$ is the direction which offloads the traffic from the overloaded neighbor base stations fastest. Figure 3.1 provides a geometrical interpretation of equation 3.13. In this figure, $p_{k,t}$ is the only overloaded neighbor of $p_{i,t}$. Same weight is assigned to all the points within the gray area. As Figure 3.1 shows, the gradient of the $G(p_{i,t})$ is toward $p_{k,t}$.

In the coverage improvement phase, each base station moves in the direction which increases local coverage fastest. The local coverage area of each base station

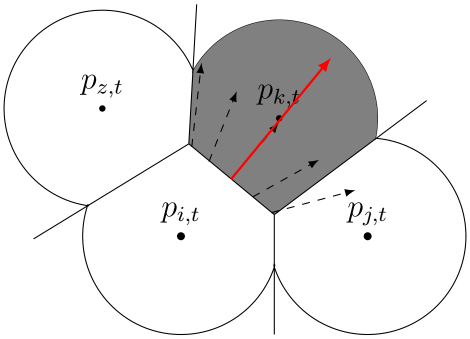


Figure 3.1: Geometrical interpretation of the gradient function with respect to $p_{i,t}$ for load balancing phase when base station located at $p_{k,t}$ is overloaded

can be formulated as

$$J(p_{i,t}) = \int_{\mu(p_{i,t})} dq \tag{3.14}$$

Each base station can calculate gradient of its local coverage function as [36]

$$\nabla_{p_{i,t}} J(p_{i,t}) = \int_{\partial\mu(p_{i,t})} \frac{q - p_{i,t}}{\|q - p_{i,t}\|} dq \tag{3.15}$$

where $\partial\mu(p_{i,t})$ is the portion of the perimeter of the sensing disk which is inside V_i. Figure 3.2 provides a geometrical interpretation of equation 3.15. It is desired in this figure to maximize the gray region by properly relocating the base station i inside the polygon. As Figure 3.2 shows, the gradient of the local coverage is toward right.

It is important to note, if base station i is on the border of prohibited region and prohibited area blocks the most desirable movement path, the gradient is not a valid moving direction. In this case, base station moves in a valid direction which has the largest positive directional derivative. Directional derivative of objective

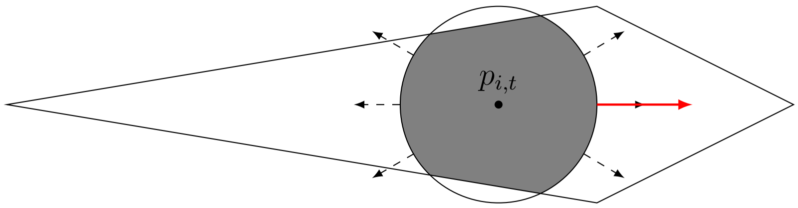


Figure 3.2: Geometrical interpretation of the gradient function with respect to $p_{i,t}$ for coverage improvement phase

function F with respect to direction u can be calculated as follows:

$$D_{\overrightarrow{u}} F(p_{i,t}) = \nabla_{p_{i,t}} F(p_{i,t}).\overrightarrow{u} = \|\nabla_{p_{i,t}} F(p_{i,t})\| \|\overrightarrow{u}\| \cos(\theta)$$

where θ is the angle between $\nabla_{p_{i,t}} F(p_{i,t})$ and $\overrightarrow{u}$. As a result, among all valid moving directions, the one with the smallest θ (and positive $\cos(\theta)$), is increasing the objective function fastest.

After base station i calculates its moving direction at step t, it moves by $a_{t,i}$ meters toward the calculated direction. $a_{t,i}$ denotes step size sequence for iterative updates of base station i's location. $a_{t,i} = A_i g(\mathit{step}(t,i))$, where A_i is the scaling factor and $g(\mathit{step}(t,i))$ is the decaying factor which gradually decreases from 1 to 0. $\mathit{step}(t,i)$ is initially set to 1 and each time base station i moves, it is incremented by 1. Choice of $a_{t,i}$ can affect the speed of convergence of the algorithm. In order to adjust $a_{t,i}$ to achieve proper convergence speed, we propose to use the following procedure:

- If over the last M relocations of base station i, the moving direction remains the same, then let $A_i = 2a_{t-1,i}$ and set $\mathit{step}(t,i) = 1$.

- If over the last M relocations, the new location of base station i falls out of its corresponding Voronoi polygon, then let $A_i = \frac{a_{t-1,i}}{2}$ and set $step(t, i) = 1$.

In the above procedure, the value of M is also important. Small M could result into incorrect updates due to small amount of information, while a large choice of M increases the convergence time due to the slow update frequency of $a_{t,i}$. If over the last M relocations, the total relocated distance by base station i is too small or too large, this means $a_{t,i}$ has not been properly chosen for the current network status. This could be either due to the size the area or rapid change in the traffic pattern.

$\Pi_i(.)$ in Algorithm 4, represents the projection function. If $p_{t,i} + a_{t,i} < \overrightarrow{D}_i >$ falls out of the Voronoi polygon of base station i, then $\Pi_i(p_{t,i} + a_{t,i} < \overrightarrow{D}_i >)$ will be projected in the polygon. Besides that, in both phases, when base station reaches the boundary of a prohibited area, it stops.

To conserve energy and decrease unnecessary nodes relocation in the network while providing an acceptable service quality, we also propose a stopping criterion. If the base station is in the coverage enhancement phase and the magnitude of coverage hole is less than ϵ_{cov}, it will not move any further. If the base station is in the load-balancing phase and the total amount of overloaded traffic within its neighbors is less than ϵ_{lb}, it will not move any further. We can achieve a trade-off between stopping time and performance by changing ϵ_{lb} and ϵ_{cov}. Larger ϵ_{lb} and ϵ_{cov} will decrease the stopping time which is at the cost of worse performance.

Remark: The problem investigated in this paper is a non convex optimization problem with unknown information about location of users within the field. Thus,

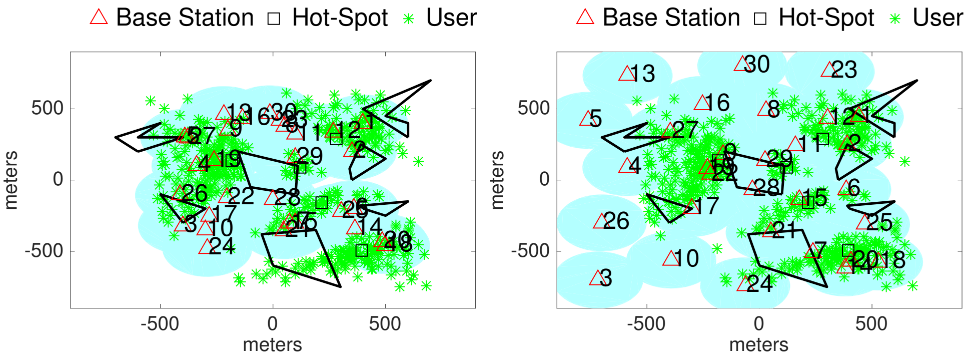
Algorithm 4 Autonomous adaptive deployment algorithm

```
▷ Each base station $s_i$ broadcasts its location $p_{i,t}$ at time t and its capacity demand $\rho_{s_i}$ to its neighbors and then constructs its Voronoi polygon based on the similar information it receives from other base stations
▷ Each node $s_i \in S$ calculates its new location as follows:
Calculate $\nabla_{p_{i,t}} G(p_{i,t})$ by equation 3.13
if $\nabla_{p_{i,t}} G(p_{i,t}) \neq \overrightarrow{0}$ and $\rho_{s_i} \leq 1$ then
    ▷ Choosing a valid moving direction which offloads traffic from overloaded neighbors fastest
    $< \overrightarrow{D}_i >= \arg\max_{\overrightarrow{D}_i : \|\overrightarrow{D}_i\|=1} \overrightarrow{D}_i . \nabla_{p_{i,t}} G(p_{i,t})$
else if $\rho_{s_i} \leq 1$ then
    ▷ Choosing a valid moving direction which increases local coverage fastest
    Calculate $\nabla_{p_{i,t}} J(p_{i,t})$ by equation 3.15
    $< \overrightarrow{D}_i >= \arg\max_{\overrightarrow{D}_i : \|\overrightarrow{D}_i\|=1} \overrightarrow{D}_i . \nabla_{p_{i,t}} J(p_{i,t})$
end if
$p_{t+1,i} = \Pi_i(p_{t,i} + a_{t,i} < \overrightarrow{D}_i >)$
```

the proposed algorithm will not necessarily result in the optimal solution.

3.5 Simulation and Results

Consider a target area of size $1800m \times 1800m$. This target area size is comparable with case 3 of Scenario III cited in the FCC report on the Public Safety Nationwide Interoperable Broadband Network [1]. Several mobile base stations that



(a)

(b)

Figure 3.3: First scenario: (a) Initial locations of base stations; (b) Final locations of base station after execution of Algorithm 4

are connected to a wireless backhaul network are expected to provide communication services to users in this area. It is assumed that each base station has 50 resource blocks of 180KHz in size. It is also assumed that the carrier frequency is 700MHz, channel bandwidth is 10MHz, and transmission power of each base station is equal to 16.39dBm/resource block. The receiver's sensitivity is considered to be -90dBm. Each base station has limited power in communicating with other base stations. It is assumed transmission power of each base station for communication between each other is equal to 36dBm.

We assume that traffic hot-spots are distributed with Poisson point process (PPP), and users (i.e. traffic sources) are generated based on the model in [35]. In this model, first a random location is assigned to each user. Then, each user u is moved toward its closest traffic hot-spot HS_u by a factor of $\beta \in [0,1]$. So, the user's new location u^{new} is calculated as $u^{new} = \beta HS_u + (1-\beta)u$. β has a Gaussian distribution with mean $\mu_\beta \in [0,1]$ and variance $\sigma_\beta = \frac{0.5-|\mu_\beta-0.5|}{3}$. A large

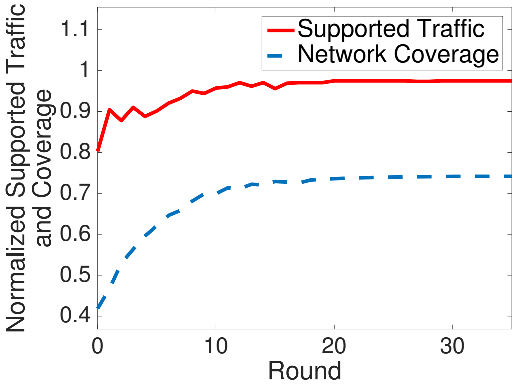


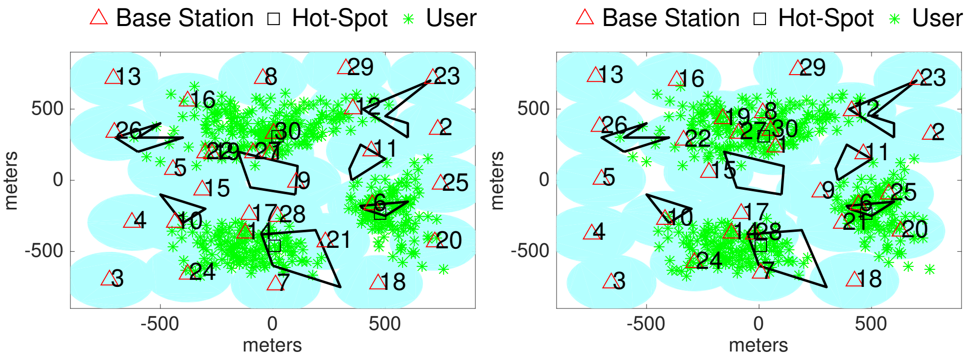
Figure 3.4: Network coverage and supported Traffic during execution of Algorithm 4 (first scenario)

μ_β will result in users being closer to traffic hot-spots, while small μ_β will lead to a uniform distribution of traffic. Each user is generating traffic with the rate of 64kbps, 128kbps or 256kbps based on a uniform distribution.

The path-loss at distance d of a base station is modeled as $40\log(d)+30\log(f)+49$ where d is in km and f is in MHz. In addition, shadow fading with a standard deviation of 5dB is also considered. A spatially correlated shadow fading environment with correlation function $r(x) = e^{-\frac{x}{50}}$ was generated as described in [24]. Using the path-loss model and receiver sensitivity, R_{cov} is calculated to be 200m. Mobile base stations employ our proposed algorithm to autonomously relocate and provide better support of traffic within the target area. Node relocation, control signaling exchange and all other updates are carried out using a 60s simulation time-step. It is assumed that each base station can relocate up to a maximum of 60m during a time-step. We set $a_{i,t} = \frac{200}{step(i,t)}$, a decreasing function of $step(n,t)$ and slowly converging to 0. We set $\epsilon_{lb} = 2$ and $\epsilon_{cov} = 3m^2$.

To the best of our knowledge, there are no dynamic deployment methodologies that aim to maximize network coverage subject to capacity constraints without any information on spatial traffic distribution across the target field. As such, we have shown the performance of our proposed algorithm with respect to static deployment scenarios.

First, we consider the capacity and coverage performance of the network considering an initial random deployment of mobile base stations at the center of a $850m \times 850m$ target field. For example, Figure 3.3(a) shows the initial positions of the base stations (marked by red triangles) along with initial user distribution (marked by green asterisk). Prohibited areas are shown by black polygons. In our simulations, μ_{β} is equal to 0.6. Given this initial deployment, base stations 1, 18 and 19 encounter high traffic demands beyond their capacity limits. With the execution of our proposed relocation algorithm, base stations that have available capacity relocate closer to traffic hot-spots. When the capacities of base stations meet the traffic demand within their coverage area, they will continue relocating to expand network coverage within the target field. In this way, traffic hot-spots that were originally outside the coverage area of the initial deployment will get an opportunity to be discovered. The above process continues until all base stations can meet their respective traffic demands and maximum network coverage is achieved. Figure 3.3(b) shows the final base station positions after 30 time-steps. Figure 3.4 shows how network coverage and the total supported user traffic evolve during the execution of our proposed algorithm. As observed, Algorithm 4 results in increasing the supported user traffic from 80% to 97% as well as improving the network coverage from 41% to 74%.



(a)

(b)

Figure 3.5: Second scenario: (a) Initial locations of base stations; (b) Final locations of base station after execution of Algorithm 4

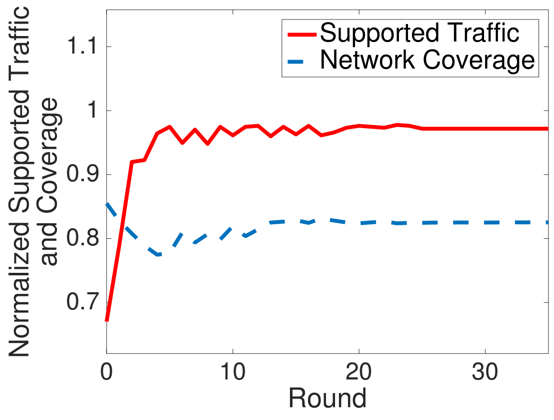


Figure 3.6: Network coverage and supported Traffic during execution of Algorithm 4 (second scenario)

Next, after the base stations converge to their final positions in Figure 3.3(b), we consider a scenario where traffic hot-spots locations change. This is shown in Figure 3.5(a). With this new traffic distribution, base stations 2, 15, 21 and 22 will face high traffic demands above their capacity limits. Again, using Algorithm 1, the base stations will autonomously relocate to new positions in order to adapt to new traffic hot-spots and accommodate the corresponding demand. Final base station positions are shown in Figure 3.5(b). Figure 3.6 depicts changes in the area coverage and supported traffic during the execution of our algorithm. As observed, the supported user traffic increases from 80% to 98% as well as improving the network coverage from 76% to 81%.

Next, we investigate the performance of our proposed algorithm by averaging over 100 different scenarios assuming a uniform initial deployment and random spatial traffic demands (i.e. μ_{beta}, number of hot-spots and their location). The results are shown in Figure 3.7. With an initial uniform deployment of base stations, occurrences of traffic hot-spots will cause several base stations to face traffic above their capacity limits. These situations result in a low average supported traffic of only 67%. Using Algorithm 4, the base stations will adaptively relocate to meet non-uniformities in the traffic demand; and therefore, the average supported traffic in the network will increase to 97%.

Next, we study how adaptation of step-size affects the coverage and supported traffic in the network. Consider a scenario with initial distribution as depicted in Figure 3.8. We set $a_{i,t} = \frac{5}{step(i,t)}$ which is small in comparison to dimensions of the Voronoi polygons. M is set to 4, which means each base station keeps track of its

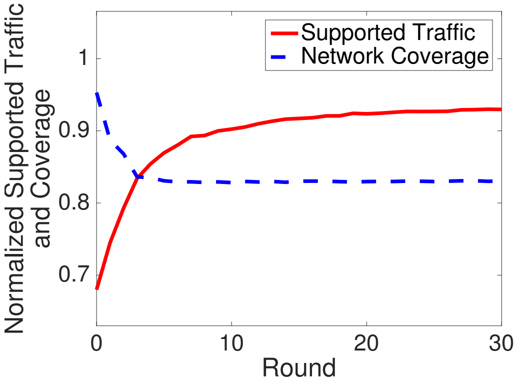


Figure 3.7: Average network coverage and supported traffic during execution of Algorithm 4 (assuming a uniform initial deployment)

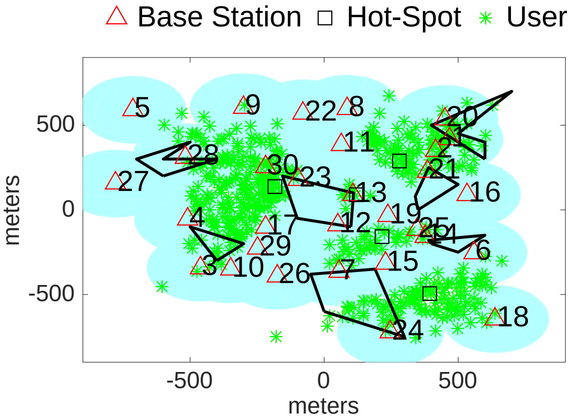
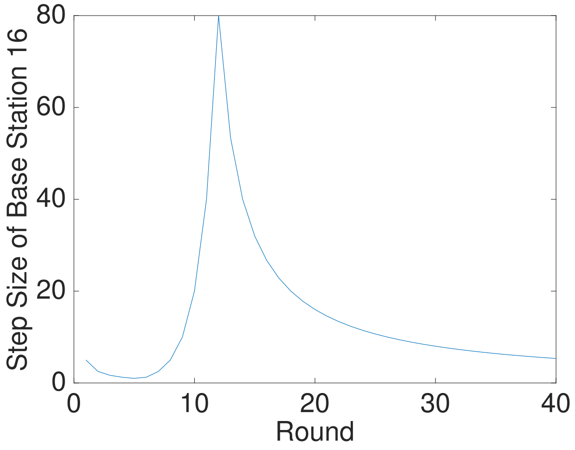
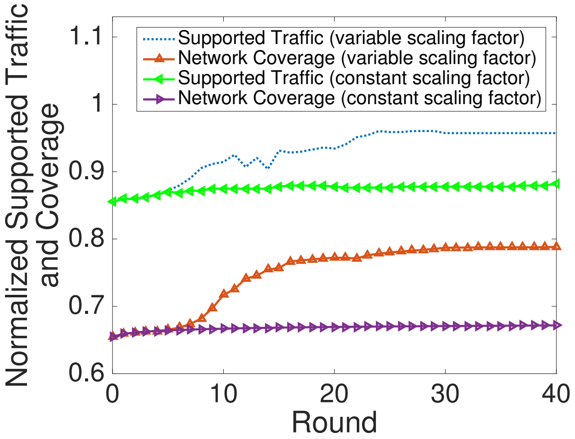


Figure 3.8: Initial location of base stations



(a)



(b)

Figure 3.9: (a) Step size variations of base station 16 during execution of Algorithm 4 with adaptive scale factor; (b) Network coverage and supported traffic during execution of Algorithm 4 for both adaptive and fixed scale factor

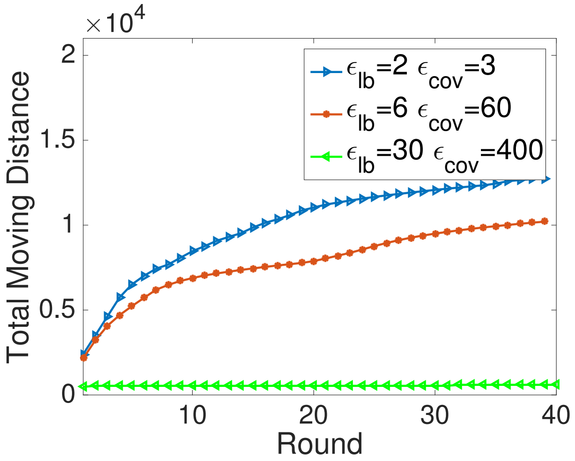
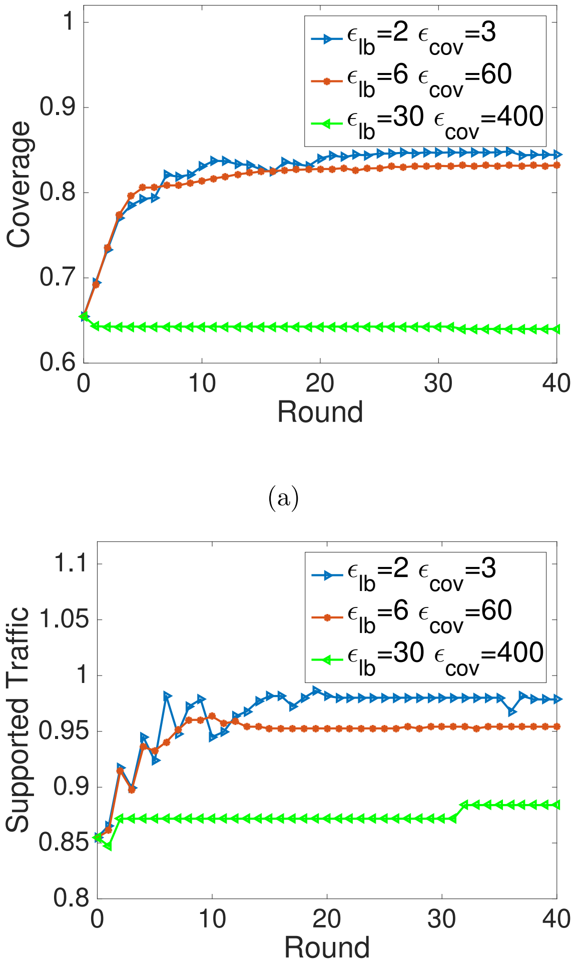


Figure 3.10: total moving distance of base stations up to each round during execution of Algorithm 4 for two different set of stopping criterias

relocations during the last 4 rounds. Figure 3.9(a) depicts changes in $a_{16,t}$ during time. As Figure 3.9(a) shows, $a_{16,t}$ is initially equal to 4. Starting from round 3, base station 16 keeps moving in the same direction which means the step-size is small. As a result, from round 7, we observe a sudden increase in $a_{16,t}$. In order to measure effectiveness of the proposed adaptation algorithm, we compare coverage and supported traffic when scale factor adaptation is deployed with when scale factor is fixed. As comparison between Figure 3.9(a) and Figure 3.9(b) shows, after there is a sudden improvement in step size, the coverage and supported traffic increase faster which is because the initial step size was too small. As Figure 3.9(b) shows, adaptive scaling factor results in faster improvement. As observed, there is only a slight improvement in coverage and supported traffic with fixed scale factor. This is due to the fact that 5 meters is too small in comparison to the size of Voronoi polygons, which means it takes a large number of rounds for each base station to reach the desired point.



(a)

(b)

Figure 3.11: (a) Network coverage during execution of Algorithm 4; (b) supported Traffic during execution of Algorithm 4

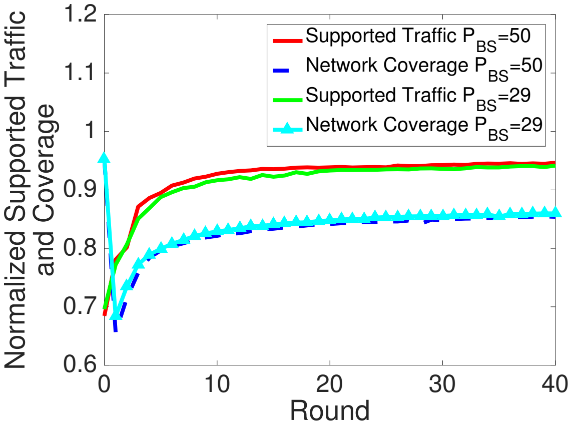


Figure 3.12: Network coverage and supported Traffic during execution of Algorithm 4 for two different communication ranges

Next, we study the effect of ϵ_{lb} and ϵ_{cov} on the performance of our proposed algorithm. Assume initial distribution as depicted in Figure 3.8. Figures 3.11(a)-3.11(b) depict how network coverage and supported traffic change for varying values of ϵ_{cov} and ϵ_{lb}. As these figures show, smaller ϵ_{cov} and ϵ_{lb} results in better performance. However, as Figure 3.10 illustrates, the total moving distance increases as the threshold decreases. This is because larger threshold will require larger performance improvement to relocate each base station. As a result, the base stations will not move if they cannot improve the performance by the specified threshold. This limitation will degrade the performance while it decreases the total moving distance.

Finally, we study effect of limited communication range between base stations on the performance of the proposed algorithm. In Algorithm 4, base station i just uses information of the neighbor base stations which are within distance $2R_{cov}$ from itself to choose the moving direction. It is reasonable to assume transmission power between base stations is such that it guarantees communication between base

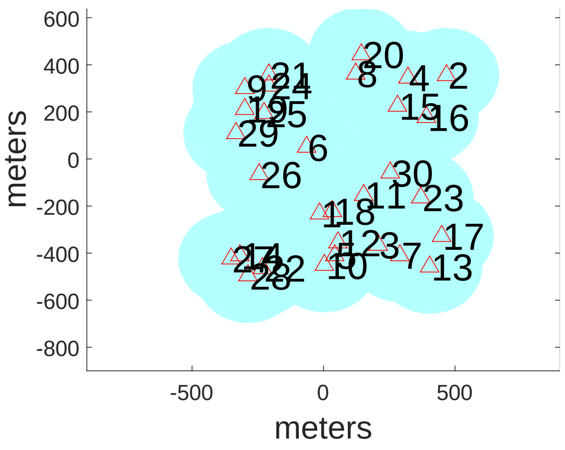


Figure 3.13: Initial locations of base stations

stations that are within distance $2R_{cov}$. Hence, irrespective of the power, base stations choose the same moving direction when Algorithm 4 is used. However, the maximum moving distance depends on the Voronoi region which is affected by the power. We study the effect of communication range between base stations on the performance of Algorithm 4 by averaging over 100 different scenarios assuming a uniform initial deployment and random μ_{beta}. We consider the same target field as in 3.5(a). As Figure 3.12 illustrates, in both scenarios the network achieves the same performance on average.

3.6 Comparison between Algorithms 3 and 4

In Chapter 2, we proposed a relocation algorithm (Algorithm 3) to increase coverage and supported traffic in public safety networks. In this chapter, it is assumed base stations can relocate to any point within the target field. However, in many practical scenarios, there are some areas in the field where mobile base stations cannot move into. In Chapter 3, we proposed another algorithm (Algorithm

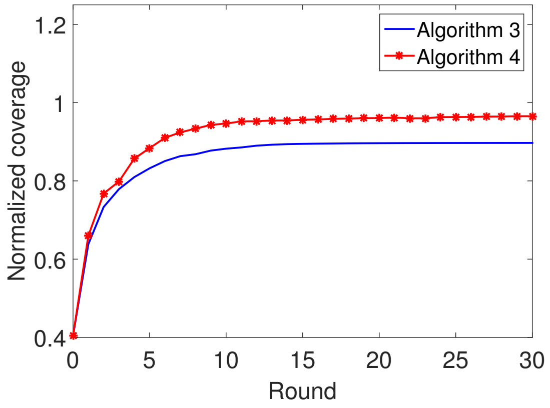


Figure 3.14: Supported traffic and network coverage during execution of Algorithms 3 and 4

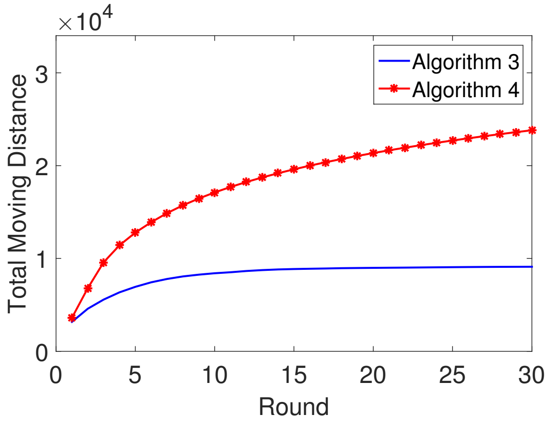


Figure 3.15: Total moving distance during execution of Algorithms 3 and 4

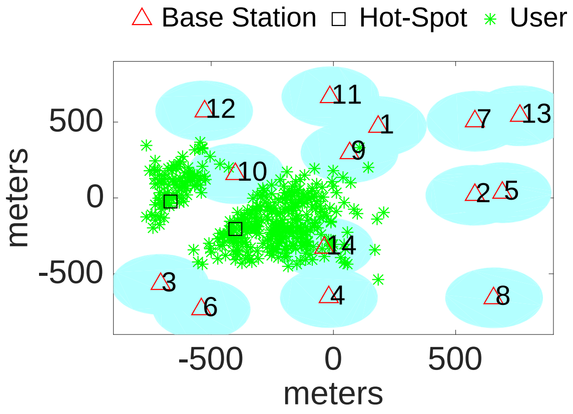


Figure 3.16: Initial locations of base stations

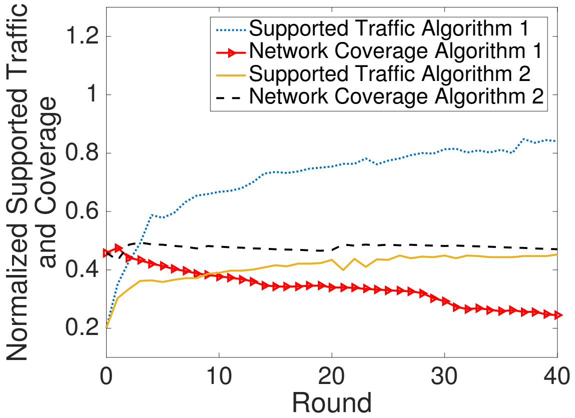


Figure 3.17: Supported traffic and network coverage during execution of Algorithms 3 and 4

4) which enables mobile cells to adapt their positions in response to potentially changing traffic patterns in a field with prohibited areas. It is important to notice, the proposed algorithm can also be used in scenarios with no prohibited area. In this section, we compare the performance of these two algorithms when there is no prohibited area within the target field by considering two different scenarios to compare the coverage improvement and load balancing phases of the two proposed algorithms.

First, we consider the coverage performance of the network considering an initial random deployment of mobile base stations with no user as shown in Figure 3.13. Given that there is no traffic in the area, we can compare the performance of the proposed coverage algorithms. As Figure 3.15 shows, total moving distance in Algorithm 4 is much larger than Algorithm 3. In coverage improvement phase of Algorithm 3, coverage improvement is guaranteed in every iteration which prevents unnecessary movements by the base station. However, in Algorithm 4, since the chosen moving direction is calculated through gradient, it guarantees coverage improvement for very short moving distance. After moving by $a_{i,t}$ in the calculated direction, the new location does not necessarily result in better local coverage. We can guarantee coverage improvement in every iteration by using line search back tracking method, but it increases complexity of the algorithm dramatically. These unnecessary movements increase the total moving distance in Algorithm 4 dramatically. Figure 3.4 shows how network coverage and the total supported user traffic evolve during the execution of these proposed algorithm. As observed, Algorithm 4 results in better coverage performance in comparison to Algorithm 3. This is due to the fact, in Algorithm 4, base stations improve coverage by using gradient function. In Figure 3.4, both Algorithms achieve almost the same supported traffic. This is because in this scenario there are enough number of base stations in the network such that each base station has at least one neighbor which is within distance $2R_{cov}$. As a result, in case of traffic overload, there is at least one neighbor which gets notified and relocates to offload the traffic from over loaded base station.

Total moving distance in Algorithm 4 can be reduced by choosing an step size which guarantees improvement in local covered area. This step size can be chosen by deploying backtracking line search; however, this increasing complexity of the algorithm dramatically.

Next, we consider a scenario as shown in Figure 3.16. In this scenario there are not enough number of base stations in the network and at most they can cover 54% of the area. So when a node gets overloaded, it does not necessarily have a neighbor which is within distance $2R_{cov}$. As a result, as shown in Figure 3.17, in these scenarios Algorithm 3 achieves better performance in comparison to Algorithm 4. In Algorithm 4, since node 14 does not have enough of number of base neighbor base stations within distance $2R_{cov}$.

3.7 Discussion

In this chapter, we considered providing a communication network which can be quickly deployed in the area. The proposed method in previous chapter does not consider disability of base stations to move to all locations within the target field due to the existence of various obstacles or other prohibited areas. Assuming, autonomous mobile base stations, we have proposed a distributed relocation algorithm that effectively adapts the overall network coverage in order to maximize the supported user traffic. Our proposed algorithm iteratively determines the best relocation direction for mobile cells while avoiding any prohibited area. Simulations show that substantial gain in performance can be achieved under typical usage scenarios.

In this chapter, it was assumed all base stations are mobile. However, as we discuss in future work, there may be scenarios in which the overall network consists of a combination of mobile and static base stations. We had also assumed all base stations are transmitting with the same fixed power. We discuss in future work how having power as another variable will affect the formulation. Another direction which is discussed in future work is keeping connectivity in the network as the base stations relocate.

CHAPTER 4

Routing in Multi-hop Networks

4.1 Overview

In recent decades, there has been huge improvement in multi-hop networks which led to new networks such as sensor networks and mobile ad-hoc networks. In these networks, designing a high performance and efficient routing algorithm is of great importance. Some of the metrics critical to the performance of these networks are throughput and delay. There is an increasing demand for high throughput and low delay scheduling and routing algorithm in multi-hop networks. High throughput is critical to respond to increasing demand of different applications. Besides that, for a large class of applications such as video or voice over IP, embedded network control and for system design; metrics like delay are of prime importance. In this context, good routing and scheduling algorithms are needed to dynamically allocate resources to maximize the network throughput region while resulting in small delay. These networks can be either static or dynamic. Dynamicity in the network can be either due to changes in network's load or the topology of the network. We are interested in a routing algorithm which can adapt the routing paths as the network changes while providing large throughput and small delay.

4.1.1 Related Work

Tassiulas and Ephremides in [3] proposed the back-pressure algorithm for scheduling and routing in multi-hop networks, and proved its throughput-optimality. A routing/scheduling algorithm is throughput-optimal in the sense of [3], if it can stabilize any traffic that can be stabilized by any other routing/scheduling algorithm. The back-pressure algorithm [3], is a congestion based routing and scheduling protocol which sends packet along the links with higher queue differential backlog. It has been proven in [3] that back-pressure algorithm is throughput-optimal. It is shown that the back-pressure algorithm can be combined with congestion control to fairly allocate resources among competing users in a network [38], [39], [40], [41]; thus, providing a complete resource allocation solution from the transport layer to the MAC layer. The back-pressure policy gives higher priority to links with higher differential backlog. The algorithm is as follows:

1. **Queue backlog Differential**: For each link (i, j) calculate $P_{ij}^d = q_i^d - q_j^d$.

2. **Destination selection**: At time t, node i chooses the maximum queue differential backlog to its neighbors.

$$d_{ij}^* = \arg\max_{d \in \mathcal{D}} P_{ij}^d$$

$$P_{ij}^* = \max\{P_{ij}^{d_{ij}^*}, 0\}$$

3. **Scheduling**: Γ is the set of all possible link schedules. The back-pressure

algorithm chooses the link scheduling:

$$\mu^* = \arg\max_{\mu \in \Gamma} \sum_{(i,j)\in\mathcal{L}} \mu_{ij} P^*_{ij}$$

4. **Packet forwarding**: transmit μ^*_{ij} packets of queue $q_{ij}^{d^*_{ij}}$, if there is less than μ^*_{ij} packets in $q_i^{d^*_{ij}}$ transfer all the packets in the queue.

While the ideas behind scheduling using the weights suggested in that paper have been successful in practice in base stations and routers, the adaptive routing algorithm is rarely used. The main reason for this is that the routing algorithm can lead to poor delay performance. In the back-pressure, the packets are sent to the neighbor nodes which have smaller queue length. Since the queue length at destination is 0, in heavy loads the traffic forces the packets toward the destination. However, in light loads there is not enough packets in the network to push the traffic toward the destination. This may lead to unnecessarily long paths and routing-loops. As a result, this approach has poor delay performance especially in light loads because it explores all feasible paths between each source and destination without considering delay metric. This extensive exploration leads to network stability. However, in light or moderate traffic the back-pressure may lead to unnecessarily long paths and routing-loops. As a result, the back-pressure algorithm has poor end-to-end delay performance. In [42] and [43], Gupta et al. analyzed the delay performance of back-pressure with a fixed route between each source destination pair. They provided a lower bound on the delay metric; however, this is only applicable when back-pressure is applied just as the scheduling policy.

There have been several studies on delay improvement in back-pressure. In [44], Ji et al. used the age of head-of-line packets instead of queue length as link weights, which is throughput-optimal for fixed routing. This solution solves last packet problem, whereby packets belonging to some flows may be excessively delayed due to lack of subsequent packet arrivals. In [45], Ying et al. proposed an algorithm that adaptively selects a set of optimal routes between each source and destination, but increases the computational complexity and the number of queues per node considerably. Both [46] and [47] use shadow queues to improve delay performance and decrease the number of queues in the network. Both algorithms are throughput-optimal scheduling for fixed routing. Some attempts have been done to adopt the original framework for handling finite buffer sizes [48]. There have been some efforts to reduce the number of queues per desination [49]. There have been also some works which study the delay performance under heavy trailed traffic [50]- [51].

Stai et al. in [52] assigned virtual coordinates in hyperbolic space to each node such that there is a greedy path in hyperbolic space between each pair of nodes. They applied back-pressure scheduling over a fixed set of greedy paths, and called it Greedy back-pressure (GBP). In greedy back-pressure, described in Algorithm 5, delay performance in light loads is improved by restricting the packets to be sent along specific loop-free paths. However, it is at the cost of decreasing the capacity region, which is the main characteristic of the back-pressure algorithm. Besides that, as our results show, the delay of greedy back-pressure (as implemented in [52]) in heavy loads may be larger than traditional back-pressure. This is due to the fact that in the greedy back-pressure of [52] they restrict packets to be sent along specific paths without considering the arrival rate information, which may lead to congestion.

4.1.2 Summary of Contributions

Our contributions in this chapter can be summarized as follows:

- We formulate the problem by maximizing the total amount of packets that are routed through loop-free paths subject to throughput-optimality constraints.
- We propose an algorithm which solves the optimization problem with same complexity as back-pressure.
- We propose an adaptation of the algorithm to handle failures in the network.
- We validate our theoretical results via simulations and demonstrate that the proposed algorithm improves the delay performance while keeping the throughput-optimality feature.

4.1.3 Outline of Chapter

The rest of this chapter is organized as follows. In Section 4.2, we summarize the properties of hyperbolic embedding and greedy routing. In Section 4.3, we describe our system and state our assumptions. In Section 4.4, we study the problem when the topology of the network is not changing. In Subsection 4.4.1, we formalize our optimization problem and obtain a greedy-aided back-pressure algorithm. We study the influence of the design parameter on performance of the proposed al-

Algorithm 5 Greedy back-pressure (GBP)

```
▷ Each node $i$ maintains a separate queue for each destination $d$
for each directed link $(i, j)$ do
    for each destination d do
        ▷ if node $j$ is a greedy neighbor of node $i$
        if $dist_H(i, d) > dist_H(j, d)$ then
            $P_{ij}^d(t) \leftarrow q_i^d(t) - q_j^d(t)$
        end if
    end for
    ▷ Each link is assigned a weight $P_{ij}$
    $P_{ij}(t) \leftarrow \max\{\max_d P_{ij}^d(t), 0\};$
    ▷ The destination which achieves the maximum in previous line
    $d^*(i, j, t) \leftarrow \arg\max_d P_{ij}^d(t);$
end for
▷ Scheduling and routing rule: Choose the rate matrix through the maximization:
$[\mu_{ij}(t)] \leftarrow \arg\max_{\mu' \in \Gamma} \sum_{(i,j)} \mu'_{ij} P_{ij}(t)$
for each directed link $(i, j)$ do
    if $\mu_{ij}(t) > 0$ then
        the link $(i, j)$ serves $d^*(i, j, t)$ with $\mu_{ij}^{d^*}(t) = \mu_{ij}(t)$
    end if
end for
```

gorithm. In Subsection 4.4.2, we describe our simulation settings and compare the proposed algorithm with the traditional back-pressure and the greedy back-pressure. In Section 4.5, we assume after some time there may be a small change in topology of the network. Then, we propose an adaptation of the algorithm in these networks. In Section 4.5.3, we compare the proposed algorithm with traditional back-pressure and the algorithm proposed in 4.4.1. In Section 4.6, we discuss complexity and distributivity of the proposed algorithm.

4.2 Hyperbolic Embedding and Greedy Geographical Paths

There are several different models for constructing hyperbolic geometry. One of the standard models is the Poincaré disc model. In this model, hyperbolic space $\mathbb{D}$ is represented by a set of points $(x, y) \in \mathbb{R}^2$ such that $x^2+y^2 \leq 1$, which represents a unit disc. We refer to points in the hyperbolic plane using complex coordinates, such that (x, y) is represented by the complex number $z = x + yi$.

The hyperbolic plane has a boundary circle denoted by $\partial\mathbb{D}$, which is $x^2+y^2 = 1$, and represents the infinity. Points on $\partial\mathbb{D}$ are at infinite distance from any point inside the circle. If u and v are two points in the unit disc, the distance between these two points in the Poincaré disk model is :

$$\cosh\ d_{\mathbb{D}}(u, v) = \frac{2|u - v|^2}{(1 - |u|^2)(1 - |v|^2)} + 1 \tag{4.1}$$

In hyperbolic geometry, the path that realizes the hyperbolic distance between two points (i.e. the shortest path) is called geodesic. In the Poincaré Disk model, geodesics between two points are represented by arcs of Euclidean circle in $\mathbb{D}$ that are perpendicular to $\partial\mathbb{D}$. Two distinct points on $\partial\mathbb{D}$ thus determine a hyperbolic line in $\mathbb{D}$. It is easy to show that given two points $a = e^{i\alpha}$ and $b = e^{i\beta}$ on $\partial\mathbb{D}$, the center of the Euclidean circle in C containing the hyperbolic line whose endpoints at infinity are a and b, and the corresponding radius are given by

$$c = \frac{1}{m^*} \quad R^2 = \frac{1}{|m|^2} - 1 \tag{4.2}$$

where $m = \frac{a+b}{2}$ is the midpoint of the Euclidean chord joining a and b, and m^* is the complex conjugate of m.

Definition 1. An embedding of a graph G in $\mathbb{H}^d$ is a mapping $C(G) : V \to \mathbb{H}^d$ that assigns to each vertex $v \in V$, a virtual coordinate $C(v)$.

In greedy geographical routing, nodes forward the packets based on the coordinates of the destination and coordinates of their neighbors. Each node sends the packet to the destination by forwarding the packet to any neighbor which is closer to the destination than the node itself. If we use Euclidean coordinates of the physical location of nodes, packets may get stuck in local minima of the distance-to-destination function.

Greedy embedding is a graph embedding that makes simple greedy geometric packet forwarding successful for every source-destination pair. In [53], the authors showed every connected graph has a greedy embedding in hyperbolic space. In [54], the authors proposed a distributed algorithm that assigns a virtual coordinate in hyperbolic plane to each node in the network, such that there exists a greedy geographical path for each pair of source and destination with respect to these virtual coordinates.

To embed the actual graph G in the hyperbolic plane, first an arbitrary spanning tree T of G is chosen. If T admits a greedy embedding in the hyperbolic space then G also admits the greedy embedding. So, the algorithm embeds the spanning tree T in the hyperbolic plane such that edges of T provide a greedy path between each pair of source and destination. As a result, in this embedding each node has at least one greedy neighbor to the destination, which is one of its children or its parent.

The idea behind the embedding algorithm is based on the following Lemma:

Lemma 3. *(Greedy Embedding): For a graph G with embedding $C(G)$, let T be a spanning tree of G. For each edge $e \in T$, let $b(e)$ be the perpendicular bisector of the embedded edge $C(e)$. Then a sufficient condition for C to be a greedy embedding of G is that for each $e \in T$, $b(e)$ intersects no embedded edges of T other than $C(e)$.*

Figure 4.1 shows an embedding of a spanning tree. The dashed lines represent the bisectors. Coordinate of root node is randomly chosen. Coordinate of its children are chosen by calculating projection of the root node with respect to the bisectors. Having a spanning tree T, the embedding algorithm is as follows:

1. Initialize by assigning to the root node r of the tree: (i) a virtual coordinate $C(r)$ in the hyperbolic plane (ii) the angles $\alpha_r = \pi$ and $\beta_r = 2\pi$ corresponding to the ideal points $a_r = e^{i\alpha_r}$ and $b_r = e^{i\beta_r}$

2. For each node $n \in G$:

    - Its parent p_n: (i) sends $C(p_n)$, $\alpha_n = \alpha_{p_n}$ and $\beta_n = \frac{\alpha_{p_n}+\beta_{p_n}}{2}$ to n; and (ii) updates $\alpha_{p_n} := \beta_n$.

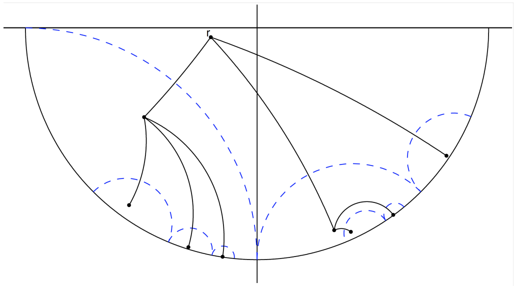


Figure 4.1: Example of a greedy embedding of an irregular spanning tree in the Poincaré disk model

- Node n: (i) calculates c and R according to 4.2 with $a_n = e^{i\alpha_n}$ and $b_n = e^{i\beta_n}$ and its own coordinate

$$C(n) = \frac{R^2}{(C(p_n))^* - c^*} + c \tag{4.3}$$

and (ii) updates $\alpha_n := \frac{\alpha_n+\beta_n}{2}$.

All steps of the algorithm are suitable for distributed and asynchronous computation. Communication takes place only between a node joining the embedded graph and its parent node, which is elected from the immediate topographic neighborhood in the graph. This embedding assigns coordinates to each node such that it guarantees paths on the spanning tree are greedy for any pair of source and destination. It is important to note that there may be other greedy paths besides the paths on the tree.

4.3 System Model

Consider a network represented by a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of directed links. Nodes are transmitters/receivers and links represent the channel between two nodes if they can directly communicate with each other. $\mu_{m,n}$ is the maximum transmission rate supported on a directed link from node m to node n. We assume that time is slotted, with a typical time slot denoted by t. We denote $\mu_{ij}^d(t)$ as communication traffic in link (i, j) for destination d at time t. Denote by $A_i^d(t)$ the amount of new exogenous data that arrives at node i on slot t that must eventually be delivered to node d. We assume each $A_i^d(t)$ satisfies the Strong Law of Large Numbers (SLLN). That is with probability 1 we have:

$$\lim_{t\to\infty} \frac{\sum_{\tau=0}^{t-1} A_i^d(\tau)}{t} = \lambda_i^d.$$

We assume $\lambda_i^d = 0$ if $i = d$. $q_i^d(t)$ denotes the queue length of a FIFO queue at node i for destination d. A scheduling policy is a set of links that are active at the same time. A scheduling policy is called feasible if activated links do not interfere with each other. We call Γ the set of all feasible schedules. Also we use the notation $\mathcal{N}(i)$ to denote the one-hop neighbors of node i.

The capacity region of the network is defined as the set of all end-to-end traffic load matrices that can be stably supported under some network control policy. By stability we mean the time average queue length of all queues in the network doesn't go to infinity. A network policy is called throughput-optimal if its capacity region is the same as the network capacity region. In [3], the authors proved that the back-pressure algorithm is throughput-optimal for the capacity region of the network denoted as Λ_G. Λ_G is the set of all input rate matrices (λ_i^d) such that there exists a rate matrix $[\mu_{ij}]$ satisfying the following constraints:

- Efficiency constraints: $\mu_{ij}^d \geq 0,\ \mu_{ii}^d = 0,$

$\mu_{dj}^d = 0,\ \sum_d \mu_{ij}^d \leq \mu_{ij},\ \forall i, d, j.$

- Flow constraints: $\lambda_i^d + \sum_l \mu_{li}^d \leq \sum_l \mu_{il}^d,\ \forall i, d : i \neq d.$

4.4 Network with Static Topology

In this section, we develop our algorithm for networks with fixed topology.

4.4.1 Greedy-aided Back-pressures

We are interested in a routing algorithm which minimizes average delay in the network subject to the throughput-optimality constraint. Assume we have hyperbolic coordinates of nodes in our network obtained through a distributed hyperbolic greedy embedding algorithm by choosing a random spanning tree. We assume our network topology does not change frequently such that at each time slot the virtual coordinates result in greedy paths between each pair of nodes.

In greedy routing, each node knows the virtual coordinates of itself, its neighbors and destination. In this routing, packets are forwarded to a neighbor which is closer to the destination than the node itself, so the distance to the destination is decreasing. Decreasing distance to the destination ensures one node cannot be passed twice, so the path is loop free. As a result, if packets are sent through greedy links, they get to the destination through loop free paths. Thus, sending packets through greedy links results in low delay when the network is not congested. However, restricting packets to be sent over a fixed set of paths may result in large delay and unstable queues in heavy loads.

In order to ensure throughput-optimality of the algorithm, the packets should not be restricted to go through a set of pre-specified paths for all set of packet arrivals. In order to keep the throughput-optimality feature while providing good delay performance, we introduce a penalty function which is the total amount of resources used over non-greedy links [55]. We are interested to find the routes for flows such that time average expected penalty is minimized subject to throughput-optimality constraints. Thus, we formulate the following optimization problem:

$$\min_{\mu_{i,j}^d(t)} \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i,j,d\in\ \mathrm{P}} \mathbb{E}\{\mu_{ij}^d(t)\} \tag{4.4}$$

$$s.t. \quad \{\mu_{nj}(t)\}_{(n,j)\in L} \in \Lambda_G,$$

where P denotes the set of (i,j,d) such that node j is not a greedy neighbor of node i for destination d.

This optimization problem minimizes the total amount of resources used by non-greedy links subject to throughput-optimality constraints. The solution to this problem will route packets through greedy paths unless greedy paths will lead to instability of queues.

Theorem 4. *The scheduling and routing algorithm described in Algorithm 6 asymptotically solves the described optimization problem.*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i,j,d} \mathbb{E}[\mu'^{d}_{i,j}(t)] = c^{opt} + O(\frac{1}{M}) \tag{4.5}$$

where c^{opt} is the infimum time average cost achievable by any policy that meets the required constraints and $\mathbb{E}\left[\mu'^{d}_{i,j}(t)\right]$ is the average link rate when using Algorithm 6 with fixed parameter M.

Furthermore, the queues are stable and the expected value of the queue length is bounded as follows:

$$\sum_{i,d} \mathbb{E}[q_i^d(\infty)] = O(M(c^{opt} - c_{min})) \tag{4.6}$$

where $\mathbb{E}[q_i^d(\infty)]$ *is the queue length as* $t \to \infty$ *and* c_{min} *represents a lower bound of the cost function.*

Proof. We denote by $Q(t) = (q_i^d(t))$, the matrix of queues in the network. We define the Lyapunov function $L(Q(t)) = \sum_{i,d} q_i^d(t)^2$. Based on Chapter 4 of [56], we need to design a controller that, at every time slot t, observes the $Q(t)$ values and subject to the known $Q(t)$ greedily minimizes the drift-plus-penalty expression which is as follows:

$$\mathbb{E}\{L(Q(t+1)) - L(Q(t)) \mid Q(t)\} + M\mathbb{E}\{\sum_{i,j,d\in\ \mathrm{P}} \mu_{ij}^d(t) \mid Q(t)\} \tag{4.7}$$

where $M > 0$ is a control parameter that affects performance-delay trade-off.

Intuitively, minimizing $\mathbb{E}\{L(Q(t+1)) - L(Q(t)) \mid Q(t)\}$ alone would tend to push the network to a lower congestion state; however, it may result in large penalty. Thus, we minimize a weighted drift-plus-penalty, where M represents how much we emphasize penalty minimization.

In order to minimize the drift-plus-penalty expression, we define two indicator functions:

$I_1(i,j,d) = \{dist_H(i,d) > dist_H(j,d) \wedge (j \in \mathcal{N}(i))\}$ which means link (i,j) is a greedy path for destination d.

$I_2(i,j,d) = \{dist_H(i,d) < dist_H(j,d) \wedge (j \in \mathcal{N}(i))\}$ which means link (i,j) is not a greedy path for destination d.

We observe that $I_1 \cap I_2 = \varnothing$. So we have:

$$\sum_{i,j,d} \mu_{ij}^d = \sum_{(i,j,d)|I_1} \mu_{ij}^d + \sum_{(i,j,d)|I_2} \mu_{ij}^d \tag{4.8}$$

The queue dynamics are:

$$q_i^d(t+1) = \max\left\{q_i^d(t) - \sum_j \mu_{ij}^d(t), 0\right\} + \sum_j \mu_{ji}^d(t) + A_i^d(t) \tag{4.9}$$

Based on Lemma 4.3 [57], if V, U, μ, A are all non-negative numbers and $V \leq \max[U - \mu, 0] + A$ then the following holds:

$$V^2 \leq U^2 + \mu^2 + A^2 - 2U(\mu - A) \tag{4.10}$$

We derive an upper-bound of the drift-plus-penalty expression as follows:

$$\begin{aligned}
&\mathbb{E}(L(Q(t+1)) - L(Q(t))|Q(t)) + M \sum_{(i,j,d)|I_2} \mathbb{E}[\mu_{ij}^d(t)|Q(t)] \\
&= \mathbb{E}\left[\sum_{i,d} q_i^d(t+1)^2 - \sum_{i,d} q_i^d(t)^2 \middle| Q(t)\right] + M \sum_{(i,j,d)|I_2} \mathbb{E}[\mu_{ij}^d(t)|Q(t)] \\
&\leq^{(4.9),(4.10)} \sum_{i,d} \left\{q_i^d(t)^2 + \mathbb{E}\big[(\sum_j \mu_{ij}^d(t))^2 + (\sum_j \mu_{ji}^d(t) + A_i^d(t))^2 |Q(t)\big]\right\} \\
&- \sum_{i,d} \left\{2q_i^d(t)\mathbb{E}\big[(\sum_j \mu_{ij}^d(t) - \sum_j \mu_{ji}^d(t) - A_i^d(t))|Q(t)\big]\right\} - \sum_{i,d} q_i^d(t)^2 \\
&+ M \sum_{(i,j,d)|I_2} \mathbb{E}[\mu_{ij}^d(t)|Q(t)] \leq B + 2\sum_{i,d} q_i^d(t)\lambda_i^d + M \sum_{(i,j,d)|I_2} \mathbb{E}[\mu_{ij}^d(t)|Q(t)] \\
&- 2\sum_{i,d} q_i^d(t)\mathbb{E}\left[\left(\sum_j \mu_{ij}^d(t) - \sum_j \mu_{ji}^d(t)\right)\middle| Q(t)\right]
\end{aligned} \tag{4.11}$$

where B is an upper-bound of

$$\sum_{i,d}\mathbb{E}\Big[\Big(\sum_{j}\mu_{ij}^{d}(t)\Big)^2+\Big(\sum_{j}\mu_{ji}^{d}(t)+A_i^d(t)\Big)^2\Big|Q(t)\Big]$$

As a result by applying (4.8) in (4.11) we have

$$\begin{aligned}&\mathbb{E}(L(Q(t+1))-L(Q(t))|Q(t))+M\sum_{(i,j,d)|I_2}\mathbb{E}[\mu_{ij}^{d}(t)|Q(t)]\leq B+2\sum_{i,d}q_i^d(t)\lambda_i^d\\&-2\sum_{i,d}\sum_{(i,j)|I_1}\Big\{\Big(q_i^d(t)-q_j^d(t)\Big)\mathbb{E}\Big[\mu_{ij}^{d}(t)\Big|Q(t)\Big]\Big\}-2\sum_{i,d}\sum_{(i,j)|I_2}\Big\{\Big(q_i^d(t)-q_j^d(t)\Big)\\&\mathbb{E}\Big[\mu_{ij}^{d}(t)\Big|Q(t)\Big]\Big\}+M\sum_{(i,j,d)|I_2}\mathbb{E}[\mu_{ij}^{d}(t)|Q(t)]=B+2\sum_{i,d}q_i^d(t)\lambda_i^d-2\sum_{i}\Big\{\sum_{j,d|I_2}\\&(q_i^d(t)-q_j^d(t)-\frac{M}{2})\mathbb{E}[\mu_{ij}^{d}(t)|Q(t)]+\sum_{j,d|I_1}(q_i^d(t)-q_j^d(t))\mathbb{E}[\mu_{ij}^{d}(t)|Q(t)]\Big\}\end{aligned}\tag{4.12}$$

Every time slot, the control decision variables are chosen to minimize the right hand side of the above inequality which results in Algorithm 6 (substitute $\frac{M}{2}$ with M). □

In Algorithm 6, $M \geq 0$ is a design parameter. The algorithm prioritizes routing packets along greedy paths over non-greedy ones in order to improve delay performance while achieving throughput-optimality. With this change, one node can send packets to the non-greedy neighbor if and only if the queue differential backlog between the node and the non-greedy neighbor exceeds M. If we choose $M = 0$, GA-BP would be the same as the traditional BP. Since hyperbolic embedding in [54] guarantees existence of a greedy path to the destination, the routing algorithm proposed in Algorithm 6 ensures packets would be routed to the destination even when $M = \infty$.

Algorithm 6 Greedy-aided back-pressure (GA-BP)

```
▷ Each node $i$ maintains a separate queue for each destination $d$
for each directed link $(i, j)$ do
    for each destination d do
        ▷ if node $j$ is a greedy neighbor of node $i$
        if $dist_H(i, d) > dist_H(j, d)$ then
            $P_{ij}^d(t) \leftarrow q_i^d(t) - q_j^d(t)$
        else
            $P_{ij}^d(t) \leftarrow q_i^d(t) - q_j^d(t) - M$
        end if
    end for
    ▷ Each link is assigned a weight $P_{ij}$
    $P_{ij}(t) \leftarrow \max\{\max_d P_{ij}^d(t), 0\};$
    ▷ The destination which achieves the maximum in previous line
    $d^*(i, j, t) \leftarrow \arg\max_d P_{ij}^d(t);$
end for
▷ Scheduling and routing rule: Choose the rate matrix through the maximization:
$[\mu_{ij}(t)] \leftarrow \arg\max_{\mu' \in \Gamma} \sum_{(i,j)} \mu'_{ij} P_{ij}(t)$
for each directed link $(i, j)$ do
    if $\mu_{ij}(t) > 0$ then
        the link $(i, j)$ serves $d^*(i, j, t)$ with $\mu_{ij}^{d^*}(t) = \mu_{ij}(t)$
    end if
end for
```

As shown in Theorem 4, the performance of the algorithm which solves the optimization problem with fixed parameter M is within $O(\frac{1}{M})$ of the optimal solution. However, the total queue length in the network increases linearly with M (or, equivalently, delay by Little's law). Very large M results in larger congestion, while too small M results in being far from the optimal solution. So we are interested in an M which is neither too large nor too small. The intuition is that greedy paths don't always provide best set of paths. By choosing a proper M we will prevent large delay in light loads by pushing packets to go through loop-free paths while as congestion in the network increases the greedy paths may not guarantee stability of queues. Then, the packets would be sent through non-greedy paths besides greedy ones.

4.4.2 Simulations and Results

In this section, we evaluate the delay and throughput performance of greedy-aided back-pressure (GA-BP) and compare it to BP and GBP via simulations. We consider two networks of wireless nodes distributed over a region and a wireline network which represents the GMPLS network of North America [58]. We assume a one hop interference model between the links for wireless networks. We assume a Poisson arrival process with mean λ for each flow in the network. In the simulations, we observed the performance of the algorithms under different traffic loads and network topology. For each, the simulation is executed for 50000 iterations.

We consider three scenarios with varying network topology depicted in Fig-

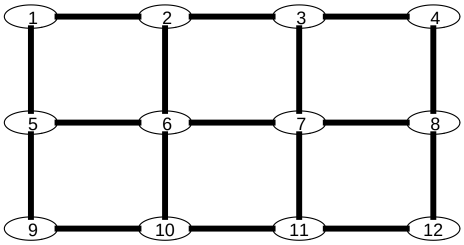


Figure 4.2: Grid network topology

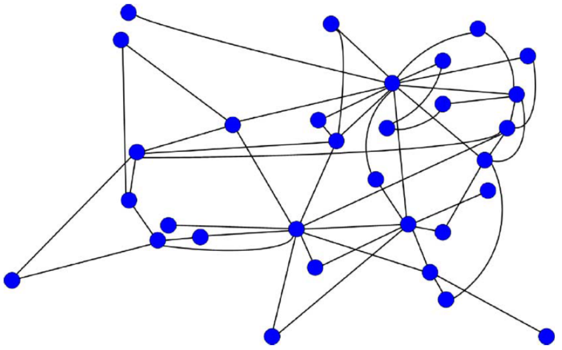


Figure 4.3: Sprint GMPLS network topology of North America

ures 4.2, 4.3 and 4.4. The scenarios are selected to contrast extreme performance scenarios in the current algorithms. In the first and second scenarios, GBP has poor performance in heavy loads and it achieves only about 50% of the network capacity. In the third scenario, GBP almost achieves the capacity region of the network. We study the performance of GA-BP in all scenarios and compare it with the performance of GBP and BP.

The optimal throughput region is defined as the set of arrival rates in which queue length and thus delay remains finite. We can consider the traffic load under

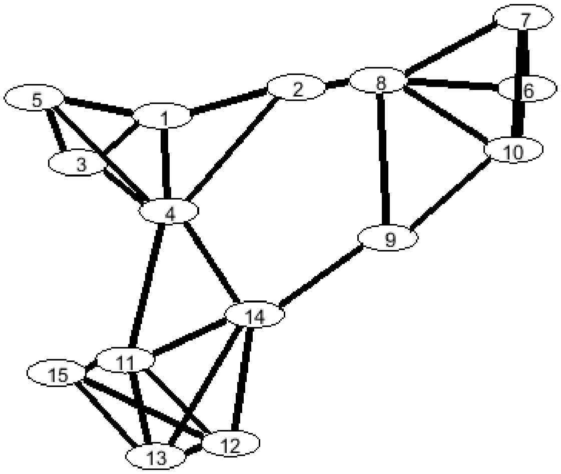


Figure 4.4: Random network topology

Flow ID	**(Source,Destination)**
1	(3,11)
2	(5,12)
3	(2,9)
4	(1,8)

Table 4.1: Set of arrival rates for scenario 1

which the queue length and thus delay increases rapidly as the boundary of the optimal throughput region.

In the first scenario, shown in Figure 4.2, the network has a grid structure with 12 nodes and 17 links. 4 flows are created in the network as shown in Table 4.1. The arrival rate of each flow ranges from 0.05 to 1.3. Each link can transmit three packets during a time slot. We find the hyperbolic embedding of the graph for a minimum spanning tree (by assigning weight equal to 1 to each link) with 1

Flow ID	**(Source,Destination)**
1	(1,23)
2	(4,23)
3	(15,27)
4	(19,11)

Table 4.2: Set of arrival rates for scenario 2

Flow ID	**(Source,Destination)**
1	(1,14)
2	(5,15)
3	(12,3)
4	(7,13)

Table 4.3: Set of arrival rates for scenario 3

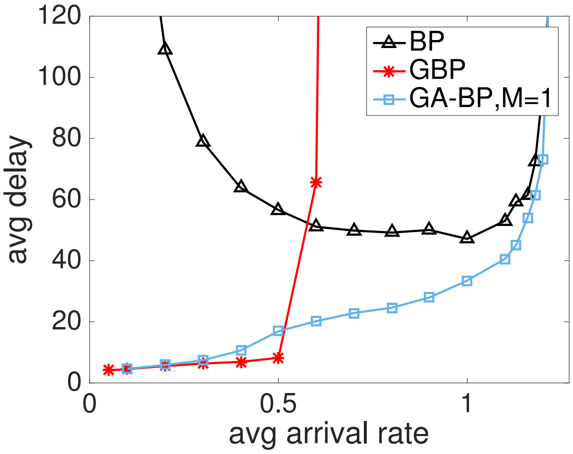
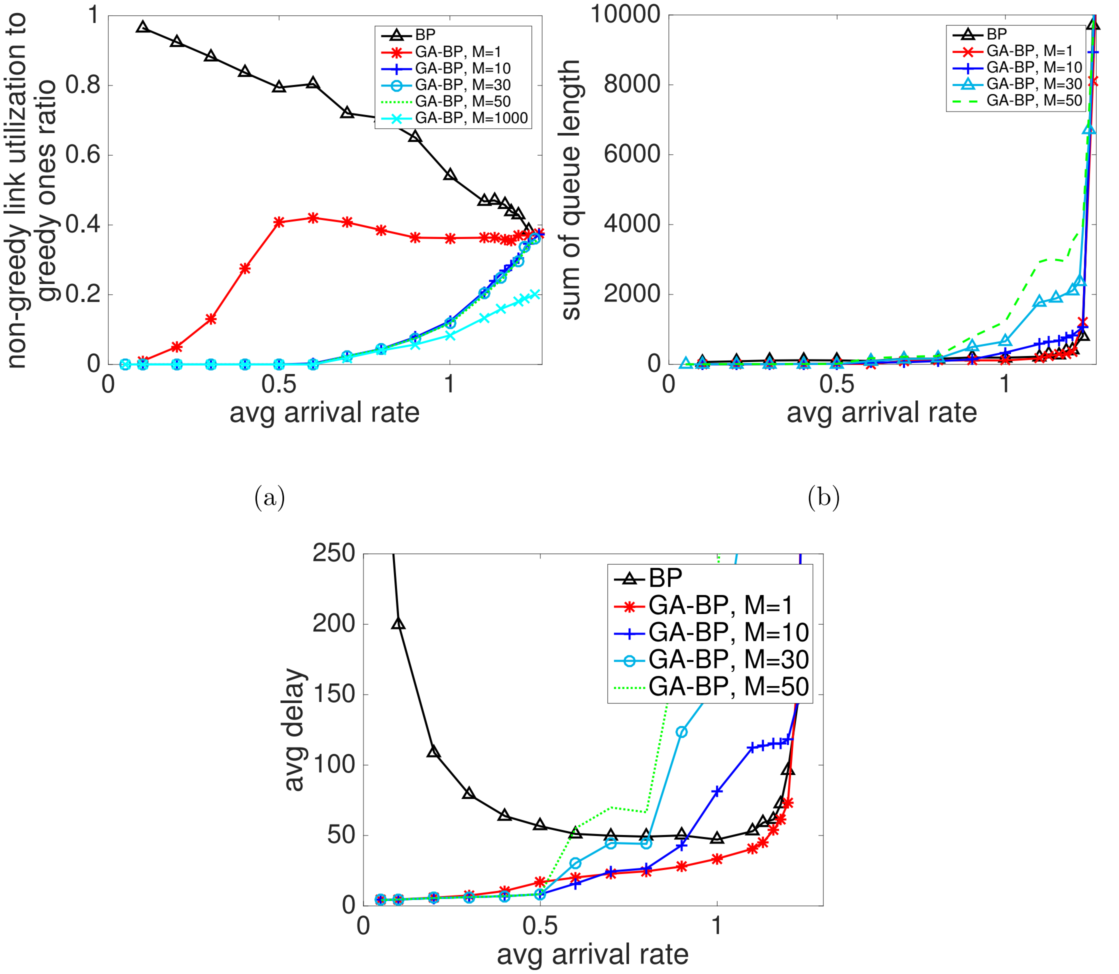


Figure 4.5: Average delay vs. average arrival rate in scenario 1



(a)

(b)

(c)

Figure 4.6: Performance parameters in scenario 1 for varying M: (a) ratio of total packets routed over non-greedy links to the packets routed over greedy ones; (b) sum of queue length vs. average arrival rate; (c) average delay vs. average arrival rate

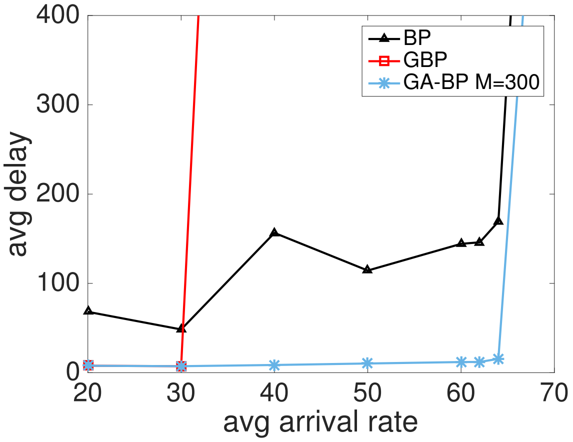
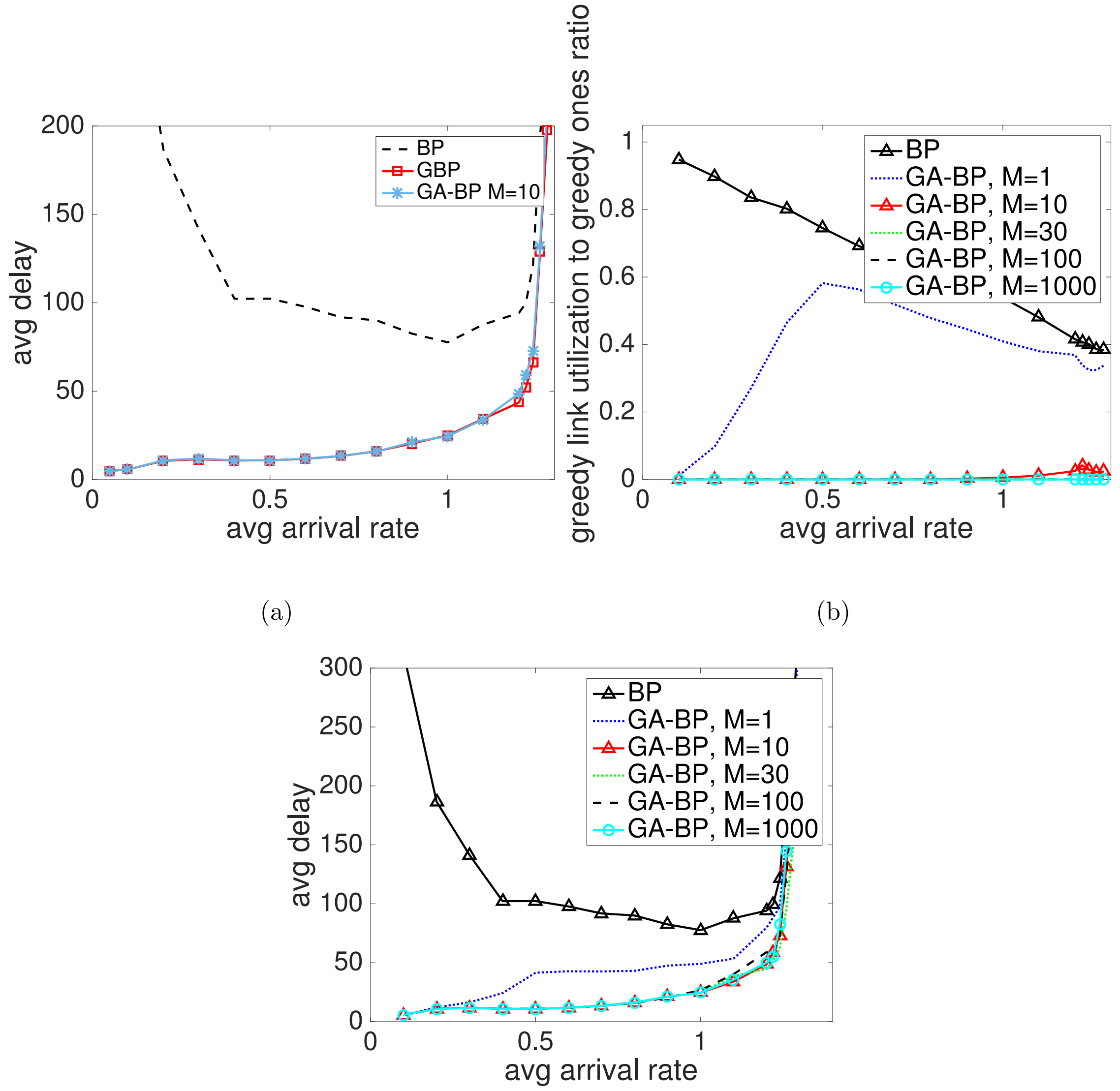


Figure 4.7: Average delay vs. average arrival rate in scenario 2

as the root node.

Figure 4.5 shows delay as a function of the arrival rate for the three algorithms BP, GBP and GA-BP with $M = 1$. It can be seen GA-BP and BP achieve the same capacity region boundary which supports our theoretical results on throughput-optimality. Moreover, GA-BP achieves better delay performance compared to BP and GBP. This is because under the back-pressure algorithm, packets are sent over routing loops and unnecessarily long paths when there is not enough congestion in the network. This leads to poor delay performance especially in light and moderate traffic. By using GA-BP, packets are routed through loop free paths of the greedy embedding in light loads. However, as the gradients build up toward the destination, packets are also forwarded through non-greedy paths which decreases the congestion in the network. So GA-BP improves delay by routing packets through shorter paths, while it also exploits long paths in the heavy traffic regime.

We vary our control parameter M to study its impact on the performance of GA-BP. In Figure 4.6(a), we illustrate the impact of M on the ratio of total



(a)

(b)

(c)

Figure 4.8: Performance parameters in scenario 3: (a) average delay vs. average arrival rate; (b) ratio of total packets routed over non-greedy links to the packets routed over greedy ones; (c) average delay vs. average arrival rate varying M.

packets routed over non-greedy links to the packets routed over greedy ones. It can be seen that as M increases the ratio of the packets routed over non-greedy links over greedy ones decreases. In Theorem 4, we have proved that the average packets sent through non-greedy links are asymptotically minimized when $M \rightarrow \infty$. Our simulation results are consistent with the theorem.

In light traffic, for the case of $M = 0$ (traditional back-pressure), the ratio is large. This is because in very light loads, there is not sufficient traffic in the network. So it takes very long time to build up gradient toward the destination. As a result, the packets choose their next hop randomly. As the arrival rate increases, the gradients towards the destinations build up faster. Thus, the packets traverse mostly through short loop free paths. Since greedy paths also contain a set of short loop free paths, when $M = 0$, as arrival rate increases, the ratio decreases. As shown in Figure 4.6(a), the ratio of GA-BP became closer to that of the back-pressure algorithm. This is because in a heavy traffic regime, GA-BP also exploits non-greedy paths to maintain stability. As M increases, higher priority is assigned to send packets through greedy links, so as expected the ratio decreases.

Next, we study the effect of M on the sum of queue length in the network. In Figure 4.6(b), we illustrate congestion in the network when using GA-BP for various values of M. As stated in the Theorem 4, the upper-bound on sum of instantaneous queue length in the network increases as M increases. Our simulation results are consistent with the theorem.

Next, we study the effect of M on delay performance. Figure 4.6(c) depicts the delay performance for varying values of M. The delays for different values of M

(except 0) are almost the same in the light traffic region. However, in moderate to heavy traffic, small M leads to better delay performance. This is because as stated in the Theorem 4, the sum of queue lengths in the network is bounded by a term proportional to M. So for large M, total congestion in the network increases which results in larger delay. On the other hand, for very small M ($M = 0$) the ratio of the packets sent through non-greedy links increases which causes loops and long paths in light loads. As M increases, the nodes prefer to send the packets to greedy neighbors. Thus for heavy load scenario, this significantly increases congestion along those paths, leading to an increase in delay. It can be observed that while setting $M = 50$ results in an improvement in delay for light load, it has the opposite effect for heavy load. This clearly highlights the influence of congestion along greedy paths as we increase the load.

In the second scenario, we consider a wireline network shown in Figure 4.3, which represents the GMPLS network topology of North America [58]. This network has 31 nodes and 52 links. We assume each link can transmit 64 packets during a time slot. We construct the hyperbolic embedding of the graph for one arbitrary spanning tree.

In Figure 4.7, we compare the delay performance of GA-BP when $M = 300$ with GBP and BP. It can be seen GA-BP and BP achieve the same capacity region boundary which supports our theoretical results on throughput-optimality. Moreover, GA-BP achieves better delay performance compared to BP and GBP. This is because under the back-pressure algorithm, packets are sent over routing loops and unnecessarily long paths when there is not enough congestion in the network. This leads to poor delay performance especially in light and moderate traffic. By using GA-BP, packets are routed through loop free paths of the greedy embedding in light loads. However, as the gradients build up toward the destination, packets are also forwarded through non-greedy paths which decreases the congestion in the network. So GA-BP improves delay by routing packets through shorter paths, while it also exploits long paths in the heavy traffic regime.

As stated earlier, we are interested to use long routes when the short routes are congested. We can detect congestion over greedy paths and start sending packets through non-greedy links by choosing a proper M. Increasing M will delay sending packets over non-greedy routes. Proper choice of M depends on queue differential backlog over greedy links when the packets are sent just through greedy paths and the network is stable. If the maximum achievable average queue length of a network is larger in comparison to another network, based on the characteristics of back-pressure, average queue differential backlog in this network is larger. As a result, a larger M should be chosen in order to prevent packets to be sent along non-greedy links before the greedy ones become heavily loaded. If the maximum achievable average queue length is large, small M results in sending packets through non-greedy links before congestion happens. In this scenario, the average length of active queues is much larger than the average queue length when $\lambda = 0.5$ in scenario 1. As a result, proper M in scenario 2 is larger than the one in scenario 1.

For the third scenario, as illustrated in Figure 4.4, we consider 15 nodes with 58 links. 4 flows are created in the network as shown in Table 4.2. The arrival rate of each flow ranges from 0.05 to 1.3. We assume each link can transmit five packets during a time-slot. We construct the hyperbolic embedding of the graph for one arbitrary spanning tree.

In Figure 4.8(a), we compare the delay performance of GA-BP when $M = 10$ with GBP and BP. It can be seen that the throughput-optimal region of GBP is the same as BP. And GA-BP and GBP perform considerably better than BP. Based on this figure, we conclude the greedy paths are performing well for this set of sources and destinations in the network. As expected in this scenario, proper M is larger than scenario 1. The reason is that the average queue length when λ is close to the capacity region, using the GBP algorithm is larger than the corresponding number in scenario 1.

In Figure 4.8(b), we illustrate the impact of M on the ratio of total packets routed over non-greedy links to the packets routed over greedy ones. It can be seen that as M increases, the ratio of the packets routed over non-greedy links over greedy ones decreases. Since routing over the greedy paths in this scenario achieves the capacity region of the network, we can achieve throughput-optimality without sending packets through non-greedy links which results in zero penalty. Based on Theorem 4, since the zero penalty is achievable, the upper bound of average sum of queue length would not depend on M. As a result, as M increases the upper bound of sum of queues remains the same. On the other hand, as M increases packets are sent through loop free greedy paths which improves delay performance.

Simulation results in this section show that the GA-BP algorithm has better performance and it achieves the capacity region of the network. In GA-BP, M is a critical parameter that should be selected carefully. The selected M should neither be too large nor too small compared to the scale of the queue length.

4.5 Network with Dynamic Topology

As stated earlier, in multi-hop networks, designing a high performance and efficient routing algorithm is of great importance. Some of the metrics critical to performance of these networks are throughput and delay. In Section 4.4, we considered networks in which the topolgy remains the same. In this section, we consider dynamic networks. In these networks, the topology of the network (and the capacity region) changes due to the addition and deletion of nodes. Networks with variable topology are everywhere. They are in the internet where links and routers are continuously added and removed, in local-area networks where users and hence traffic are dynamic, also in mobile adhoc wireless networks in which nodes are moving and environment condition changes. Dynamicity can be either continuous or transient. In continuous ones, changes are constantly occurring and the system has to constantly adapt to them. In transient dynamic network, which is considered in this section, changes occur for a short period, after which the system is static for an extended time period. For instance, consider a sensor network in which sensors are not moving, topology of the network changes when sensors run out of battery or when a new sensor joins the network.

An important issue in dynamic networks is the design and analysis of routing schemes. In these kind of networks, routing paths should be chosen according to real-time network topology changes. In this chapter, we are interested to propose an adaptation of the Algorithm 6 to adapt to minor changes in the network topology.

4.5.1 Related Work

The back-pressure algorithm proposed in [3], is an algorithm for dynamically routing traffic over a multi-hop network by using congestion gradients. The algorithm can be applied to wireless communication networks, including sensor networks, mobile ad hoc networks (MANETS), and heterogeneous networks with wireless and wireline components. Back-pressure is adaptive to dynamic topology, when either the links are not interfering with each other or when a distributed scheduling algorithm is used. However, as stated in the previous section, this algorithm may introduce large delay due to sending packets through routing loops.

In [54], the authors proposed an embedding and routing scheme for arbitrary network connectivity graphs, based on greedy routing and utilizing virtual node coordinates. The authors suggested to use adapted routing algorithm in case a packet gets stuck in a local minimum due to changes in the topology of the network. The resulting routing is referred as Gravity-Pressure routing. This technique tries to forward the packets via neighbors which decrease the distance toward the destination as much as possible. To achieve this end, they considered two modes for a packet called gravity mode and pressure mode. They added a flag bit to the header of the packet which determines mode of the packet. When a packet is generated, it is in gravity mode. When it reaches local minima, it changes its mode to pressure mode and saves its distance to the destination denoted by d_v. The packet's mode changes to gravity mode when it reaches a node which distance is less than d_v. In order to avoid the packet from getting stuck into a loop, each packet has to maintain a path trace from the moment a local minimum is detected to prevent going through the same path again. However, Gravity-Pressure has very poor delay performance.

In [52], the authors proposed an algorithm to address local minima as a result of node removal in dynamic networks. However, the proposed algorithm does not guarantee delivery of packets to the destination in connected networks. In [52], when the packet reaches a node which does not have any greedy neighbor (i.e. the neighbor which lies on the greedy path to the destination), it can be sent to any of its neighbors until it gets to a node which has a greedy neighbor toward the destination. This method may results in sending packet over a loop and periodically returning to the same local minima point.

4.5.2 Dynamic Greedy-aided Back-pressure

Based on the embedding method provided in [54], new coordinates will be assigned to nodes joining the network such that the greedy property is guaranteed. However, when a node leaves, the greedy property can be locally destroyed and the greedy routing must be adapted in order to avoid possible local minima, without re-embedding the whole network. If node i is the only greedy neighbor of node j toward destination d, node j will not have any greedy path toward destination d after i's removal. In dynamic greedy-aided back-pressure, when node j loses all its greedy neighbors toward destination d, it notifies its neighbors that destination d is not reachable through j by following greedy path. Each node j keeps track of its greedy neighbors which cannot reach destination by following greedy neighbors. G_j includes all (k, d) such that $dist(k, d) < dist(j, d)$ and d is not greedily reachable through k.

We modify the objective function in static scenario for dynamic networks:

$$\begin{aligned} \min \quad & \sum_{i,j,d} \mu_{ij}^{d}(1 - I(i,j,d))R_{i,d} \\ s.t \quad & \text{throughput-optimality} \end{aligned} \tag{4.13}$$

Where $I(i, j, d) = 1_{\{dist_H(i,d)>dist_H(j,d) \wedge (j \in \mathcal{N}(i))) \wedge (j,d) \notin G_i\}}$ which means d is greedily reachable through neighbor j. $R_{i,d}$ represents reachability of node d through node i. $R_{i,d} = 0$ if node d is reachable through greedy neighbors of node i, otherwise it is equal to 1.

This optimization problem aims to minimize sending packets to non-greedy links, if the initiator of the link has greedy path to the destination.

Theorem 5. *The scheduling and routing algorithm described in Algorithm 7 asymptotically solves the described optimization problem.*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i,j,d} \mathbb{E}[\mu'^{d}_{i,j}(t)(1 - I(i,j,d))R_{i,d}] = c^{opt} + O(\frac{1}{M}) \tag{4.14}$$

where c^{opt} is the infimum time average cost achievable by any policy that meets the required constraints and $\mathbb{E}\left[\mu'^{d}_{i,j}(t)\right]$ is the average link rate when using Algorithm 7 with fixed parameter M.

Furthermore, the queues are stable and the expected value of the queue length is *bounded as follows:*

$$\sum_{i,d} \mathbb{E}(q_i^d[\infty]) = O(M(c^{opt} - c_{min})) \tag{4.15}$$

where $\mathbb{E}[q_i^d(\infty)]$ *is the queue length as* $t \to \infty$ *and* c_{min} *represents a lower bound of the cost function.*

Proof. The proof follows exactly the same procedure as Theorem 4. □

The following theorem proves throughput-optimality of DGA-BP assuming a controller adapts the arrival rates to lie inside the capacity region. This means the algorithm can ensure throughput-optimality in dynamic networks.

Theorem 6. *Let* T_k *be a time interval during which the number of nodes* $N(T_k)$ *stays constant and* $I(i,j,d)$ $\forall i,j,d$ *does not change. In addition, assume* $\lambda_i^d(t)$ *lies within capacity region* $\Lambda_G(T_k)$*. Then, the* $q_i^d(t)$ *queues are strongly stable.*

Proof. We denote $I(i,j,d)$ and $R_{i,d}$ in interval T_k by $I_{T_k}(i,j,d)$ and $R_{T_k}(i,d)$. We define four indicator functions:

$I_{1_{T_k}}(i,j,d) = \{I_{T_k}(i,j,d) = 1\}$, which means based on current information, node i assumes node j is a greedy neighbor with a greedy path toward destination d. $I_{2_{T_k}}(i,j,d) = \{I_{T_k}(i,j,d) = 0 \ \wedge \ R_{T_k}(i,d) = 1\}$, which means based on current information, node i assumes it has at least one greedy path toward destination d; however, node j is either not a greedy neighbor or it does not have a greedy path toward destination. $I_{3_{T_k}}(i,j,d) = \{R_{T_k}(j,d) = 0\}$, which means based on current information, node j assumes it does not have any greedy path toward destination d. $I_{4_{T_k}}(i,j,d) = \{I_{T_k}(j,i,d) = 1\}$, $I_{5_{T_k}}(i,j,d) = \{I_{T_k}(j,i,d) = 0 \ \wedge \ R_{T_k}(j,d) = 1\}$,

$I_{6_{T_k}}(i,j,d) = \{R_{T_k}(j,d) = 0\}$.

Note that if we replace i and j with each other, $I_{1_{T_k}}, I_{2_{T_k}}, I_{3_{T_k}}$ change to $I_{4_{T_k}} I_{5_{T_k}}, I_{6_{T_k}}$ respectively. We define $L(t) = \sum_{i=1}^{N_{T_k}} \sum_{d=1}^{V_{T_k}} q_i^d(t)^2$ as the Lyapunov function for $t \in T_k$. We can prove throughput-optimality by mimicing Theorem 2 in [52] as follows. First we need to prove the Lyapunov drift takes the form

$$\mathbb{E}[L(q(t+1)) - L(q(t))|q(t)] \leq B_{T_k} - \sum_{i=1,d=1}^{N_{T_k}} q_i^d(t)\epsilon \tag{4.16}$$

This shows that for each interval T_k with constant number of nodes $N(T_k)$, if the arrival rates are adapted through a controller to lie inside the capacity region of this interval $\Lambda_G(T_k)$, the sum of queues is bounded. Then, we check the Lyapunov drift at the transitions between two intervals with different number of nodes or different $I(i,j,d)$. Then, we will have

$$\exists B_{max}\ \forall t \quad \text{s.t.} \quad \mathbb{E}[L(q(t+1)) - L(q(t))|q(t)] \leq B_{max} - \sum_{i,d} q_i^d(t)\epsilon \tag{4.17}$$

Then, we take expectation of 4.17 with respect to $Q(t)$ and take summation over t. By using telescopic sum, we can prove stability of queues. □

4.5.3 Simulation and Results

In this section, we evaluate the delay and throughput performance of dynamic greedy-aided back-pressure (DGA-BP) and compare it to BP and GA-BP via simulations. We consider GMPLS network which was also studied in Section 4.4.2. We assume a Poisson arrival process with mean λ for each flow in the network. In the simulations, we observed the performance of the algorithms. For each, the simulation is executed for 50000 iterations.

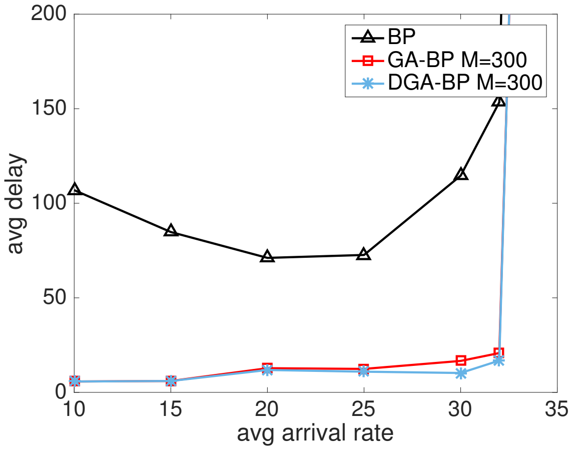


Figure 4.9: Average delay vs. average arrival rate after node 7's removal

The optimal throughput region is defined as the set of arrival rates in which queue length and thus delay remains finite. We can consider the traffic load under which the queue length and thus delay increases rapidly as the boundary of the optimal throughput region.

Figure 4.3 represents the topology of the network. This network has 31 nodes and 52 links. We assume each link can transmit 64 packets during a time slot. 4 flows are created in the network as shown in Table 4.4. The arrival rate of each flow ranges from 20 to 67. Each link can transmit 64 packets during a time slot. We find the hyperbolic embedding of the graph for a minimum spanning tree (by assigning weight equal to 1 to each link) with 1 as the root node.

After some time, node 7 fails. The proposed algorithm is deployed to route the packets in the network. Figure 4.9 shows delay as a function of the arrival rate for the three algorithms BP, GA-BP and DGA-BP with $M = 300$. After node 7's failure, there still exists greedy paths between each pair of nodes and there is no local minima on the paths connecting sources to destinations. As a result, both

Flow ID	**(Source,Destination)**
1	(1,31)
2	(4,31)
3	(15,27)
4	(19,11)

Table 4.4: Set of flows in dynamic scenario

Flow ID	**(Source,Destination)**
1	(1,31)
2	(4,31)
3	(19,11)

Table 4.5: Set of flows after node 15's removal

GA-BP and DGA-BP achieve good delay performance.

After some time node 15 fails as well. Due to failure, there won't be any greedy path between nodes 1 and 31, 4 and 31. Besides that, since node 15 is removed from the network, it does not generate any packet. Table 4.5 shows the updated set of flows in the network. Figure 4.10 depicts the delay as a function of the arrival rate for the three algorithms BP, GA-BP and DGA-BP with $M = 300$. As stated earlier, DGA-BP, GA-BP and BP achieve the same capacity region boundary which supports our theoretical results on throughput-optimality. As expected, DGA-BP achieves better performance in comparison to GA-BP and BP. In GA-BP packets are mostly forced to go through greedy paths which may result in local minima. The packets generated by nodes 1 and 4 get stuck at local minima. When they reach local minima, they cannot be forwarded to any neighbor until the queue differential backlog reaches M. In light loads, it takes long time to acheive a queue differential backlog greater than M. As a result, this will increase delay in light loads significantly. However, in DGA-BP packet is not forced to go through paths which lead to local minima. This will improve the delay performance of DGA-BP in comparison to GA-BP significantly.

4.6 Complexity and Distributivity of Greedy-aided back-pressure

In this section, we compare the computational complexity and distributivity of the proposed algorithm to the traditional back-pressure algorithm. Intuitively, it may be observed that the complexity of our method is similar to the traditional

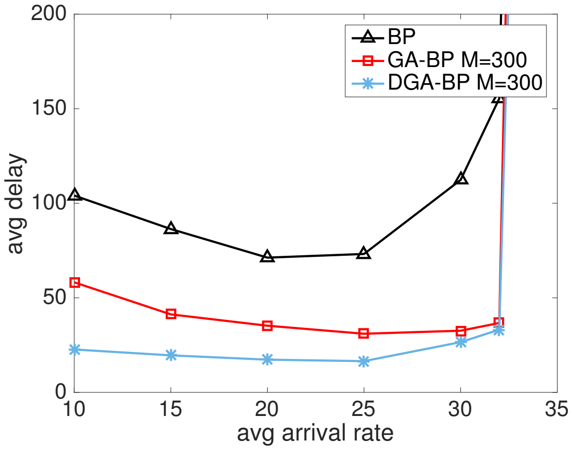


Figure 4.10: Average delay vs. average arrival rate after node 15's removal

back-pressure. In the first stage of our method, we embed the graph in hyperbolic space by assigning a hyperbolic coordinate to each node. This involves the overhead of computing a spanning tree in the graph. However, since this step is only required once in the beginning, the overhead may be negligible over the lifetime of the network. In both algorithms, each node computes and utilizes queue differential backlogs (with a bias factor for our algorithm) to make routing decisions. Thus, the recurring steps in each algorithm have the same complexity. Addition of new nodes will not affect complexity of the algorithm. However, in case of failure there is a signal overhead to let the neighbors of node i know if node i does not have any greedy neighbor with reachable path toward destination d.

Even though the traditional back-pressure is a centralized algorithm, several studies on suitable distributed scheduling algorithms that achieve throughput-optimality have been proposed in the literature, e.g. [59]- [60]. We may modify the weights assigned to links in these scheduling algorithms by utilizing the metrics from GA-BP. Thus, these algorithms can be utilized to substitute the scheduling in GA-BP, enabling the execution of our algorithm in a distributed manner while satisfying throughput-optimality. Since the hyperbolic coordinates are calculated in a distributed manner, as demonstrated in [54], a fully distributed implementation of our algorithm is possible.

4.7 Discussion

In this paper, we have proposed a greedy-aided back-pressure algorithm to improve the delay performance, while maintaining the throughput-optimality property of the traditional back-pressure algorithm. We analyzed the proposed algorithm analytically and via simulations. We demonstrated the improvement in delay performance of our algorithm over traditional routing schemes. Our algorithm provides the network designer a control parameter M to tune the delay-performance of the network. As we discuss in future work, an algorithm which can adaptively find the proper parameter in the network is a future direction. Further, our algorithm is robust to addition of new nodes. Due to the incremental property of hyperbolic embedding, there is no need to re-embed the network. However, for real network deployments, link and node failures may occur. We proposed an adaptation of the algorithm to adapt to changes in the network. However, the proposed solution is not proper for highly dynamic networks. As discussed in future work, finding an algorithm that can adapt to frequent changes in the network is a future direction.

Algorithm 7 Dynamic greedy-aided back-pressure (DGA-BP)

```
▷ Each node $i$ maintains a separate queue for each destination $d$
for each directed link $(i, j)$ do
    for each destination $d$ do
        ▷ if node $d$ is greedily reachable through node $i$
        if $R_{i,d} = 1$ then
            ▷ if node $j$ is a greedy neighbor with a greedy path toward destination $d$
            if $I(i, j, d) = 1$ then
                $P_{ij}^{d}(t) \leftarrow q_{i}^{d}(t) - q_{j}^{d}(t)$
            else
                $P_{ij}^{d}(t) \leftarrow q_{i}^{d}(t) - q_{j}^{d}(t) - M$
            end if
        else
            $P_{ij}^{d}(t) \leftarrow q_{i}^{d}(t) - q_{j}^{d}(t)$
        end if
    end for
    ▷ Each link is assigned a weight $P_{ij}$
    $P_{ij}(t) \leftarrow \max\{\max_{d} P_{ij}^{d}(t), 0\};$
    ▷ The destination which achieves the maximum in previous line
    $d^{*}(i, j, t) \leftarrow \arg\max_{d} P_{ij}^{d}(t);$
end for
```

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

In this dissertation, we discussed two different problems in dynamic networks. In first problem, we discussed coverage problem in public safety networks. Public safety organizations increasingly rely on wireless technology for their mission critical communication during emergencies and disaster response operations. In such scenarios, a communication network could face much higher traffic demands compared to its normal operation. Given the limited capacity of base stations in the network, such peak traffic scenarios could lead to high blocking probability or equivalently service interruptions during critical communications. At the same time, networking infrastructure can partially (or sometimes fully) breakdown during a disaster. A mobile communication infrastructure composed of Cells on Wheels can be a viable solution to complement or replace the existing static infrastructure. A proper deployment of these mobile cells can enhance the network coverage or accommodate excess traffic in areas with high concentration of users such as first responders. In addition, an intelligent relocation strategy can be used to efficiently adapt the cell locations to match variations in the spatial distribution of the traffic.

In the second part of dissertation, we considered routing problem in multi-hop networks. We proposed greedy-aided back-pressure algorithm to improve the delay performance, while maintaining the throughput-optimality property of the traditional back-pressure algorithm. Our algorithm introduces a control parameter M by which we tune the delay performance of the network. We analyzed the performance of the proposed algorithm analytically and via simulations. We demonstrated the improvement in delay performance of our algorithm over traditional routing schemes. Further, our algorithm is robust to addition of new nodes. Due to the incremental property of hyperbolic embedding, there is no need to re-embed the network. However, for real network deployments, link and node failures may occur frequently. This may lead to loss of greedy paths, causing the packets to get stuck in local minima. This requires an adaptation in the greedy route selection. We addressed this problem through exchanging additional information between nodes. However, the proposed algorithm does not work in networks with frequent failures.

In the remainder of this chapter, we provide intuitive connections and results that may be explored for future research in these directions.

5.2 Autonomous Relocation Strategies for Cells on Wheels

In Chapters 2 and 3, we emphasized on the need for a network which can be quickly deployed and adapt itself to changes in traffic distribution. We proposed distributed relocation algorithms that effectively adapt the overall network coverage in order to increase the supported user traffic. Our proposed algorithms iteratively determine the best relocation direction for mobile cells. Simulations show that substantial gain in performance can be achieved under typical usage scenarios. Here, we suggest some future directions in the autonomous relocation problem.

5.2.1 Adaptive Power Allocation

One of the assumptions for our proposed algorithm is all base stations are having the same coverage range. As discussed in Section 2.2.1, this assumption results in approximating the total covered area by summing the local covered areas in each Voronoi polygon. By considering power as another variable, we will be able to increase the total covered area and supported traffic. By such a solution, if traffic load in an area is light, the power of the base station increases to increase the covered area and provide service to more number of users; however, base stations decrease transmission power in areas with dense traffic. In order to formulate the total covered area in case of non-identical communication range, we can use multiply weighted Voronoi Diagrams [27].

5.2.2 Mix of Stationary and Mobile Base Stations

Natural disasters such as tornado and earthquake may not break down the network infrastructure. However, it could result in sudden increase in traffic load and large peak to average traffic ratio. These unusual peaks in traffic load could lead to much higher blocking probability or service interruptions for critical communication.

One possible solution to tackle this problem is using Cell on Wheels. By deploying Cell on Wheels in the areas where the base stations are overloaded, we can offload the traffic from overloaded base stations. There is a need for an autonomous relocation algorithm for COWs underlay LTE network in order to improve the supported traffic.

5.2.3 Connectivity

As discussed in Chapters 2 and 3, base stations have limited communication range between each other. As stated earlier, one possible way to create backhaul network for such networks is by creating an adhoc network which consists of mobile base stations. These mobile base stations will deploy routing algorithms to forward traffic between each other. This network necessitate connectivity between base stations in order to guarantee the collected data can reach the destination. Adaptation of the proposed solution to guarantee connectivity in the network remains an unsolved issue that is critical for future systems.

5.3 Routing

In Chapter 4, we proposed a routing algorithm based on back-pressure and greedy routing which improves delay performance while maintaining throughput optimality feature. Our proposed algorithm combines back-pressure and greedy routing in an innovative method such that it restricts route selection to greedy paths as long as queues are stable and it reverts to non-greedy paths in other scenarios. The proposed algorithm does not increase the complexity of the routing algorithm with respect to back-pressure routing. Then, we proposed an adaptation of the proposed algorithm to handle infrequent failures of nodes in the network.

5.3.1 Self-tuning Algorithm for M

In Chapter 4, our algorithm provided the network designer a control parameter M to tune the delay-performance of the network. The magnitude of optimal M may vary depending on the network topology and arrivals. Self adaptation of M remains an unsolved issue that is critical for future network.

5.3.2 Extracting Path Length Information from Hyperbolic Coordinates

As stated in Chapter 4, we are interested in a routing algorithm which sends packet through shorter paths in light load. As congestion in the network increases the packets are sent through non-greedy paths. Hyperbolic embedding usually provides a set of greedy paths for each pair of source and destination. Extracting information regarding length of the paths from the hyperbolic coordinates of neighbors can help in prioritizing greedy paths with respect to each other.

5.3.3 Frequent Node Failures

The proposed algorithm results in large overhead in case of frequent node failures in the network. An adaptation of our routing algorithm which can be deployed in highly dynamic scenarios remains an unsolved issue that is critical for future network.

Bibliography

[1] Technical report, Federal Communications Commission, 2008. `https://transition.fcc.gov/pshs/docs/releases/DOC-298799A1.pdf`.

[2] Technical report, Federal Communications Commission, 2015. `http://transition.fcc.gov/cgb/consumerfacts/emergencies.pdf`.

[3] L. Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transaction on Automatic Control*, 37(12):1936–1948, Dec 1992.

[4] Technical report, Federal Communications Commission, 2008. `http://www.fcc.gov/general/emergency-communications-during-minneapolis-bridge-disaster`.

[5] S. Hurley. Planning effective cellular mobile radio networks. *IEEE Transactions on Vehicular Technology*, 51(2):243–253, 2002.

[6] Jin Kyu Han, Byoung Seong Park, Yong Seok Choi, and Han Kyu Park. Genetic approach with a new representation for base station placement in mobile communications. In *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, volume 4, pages 2703–2707 vol.4, 2001.

[7] Ranveer Chandra, Lili Qiu, Kamal Jain, and Mohammad Mahdian. Optimizing the placement of integration points in multi-hop wireless networks. In *IEEE International Conference on Network Protocols (ICNP)*. IEEE Communications Society, October 2004.

[8] Rudolf Mathar and Thomas Niessen. Optimum positioning of base stations for cellular radio networks. *Wirel. Netw.*, 6(6):421–428, 2000.

[9] N. Weicker, G. Szabo, K. Weicker, and P. Widmayer. Evolutionary multi-objective optimization for base station transmitter placement with frequency assignment. *Trans. Evol. Comp*, 7(2):189–203, April 2003.

[10] Rudolf Mathar and Thomas Niessen. Optimum positioning of base stations for cellular radio networks. *Wirel. Netw.*, 6(6):421–428, December 2000.

[11] Ryan Hallahan and Jon M Peha. Enabling public safety priority use of commercial wireless networks. *Homeland Security Affairs*, 9:13, 2013.

[12] Jon M Peha. A public-private approach to public safety communications. *Issues in Science and Technology*, 29(4):37–42, 2013.

[13] Ryan Hallahan and Jon M Peha. The business case of a network that serves both public safety and commercial subscribers. *Telecommunications Policy*, 35(3):250–268, 2011.

[14] G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, and N. Brahmi. Device-to-device communications for national security and public safety. *IEEE Access*, 2:1510–1520, 2014.

[15] Technical report, Motorola Solutions, Inc., 2012. `http://www.motorolasolutions.com/content/dam/msi/docs/business/services/_document/static_files/public_safety_data_communications_survey.pdf`.

[16] Maissa Boujelben, Sonia Ben Rejeb, and Sami Tabbane. Interference coordination schemes for wireless mobile advanced systems: A survey. *CoRR*, abs/1403.3818, 2014.

[17] K. T. Morrison. Rapidly recovering from the catastrophic loss of a major telecommunications office. *IEEE Communications Magazine*, 49(1):28–35, January 2011.

[18] Xu Chen, Dongning Guo, and John Grosspietsch. The public safety broadband network: A novel architecture with mobile base stations. *CoRR*, abs/1303.4439, 2013.

[19] A. Merwaday and I. Guvenc. UAV assisted heterogeneous networks for public safety communications. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE*, pages 329–334, March 2015.

[20] Aerial base stations with opportunistic links for unexpected and temporary events. Technical report. `http://www.absoluteproject.eu`.

[21] S. Bhattarai, S. Wei, S. Rook, W. Yu, D. Griffith, and N. Golmie. Optimizing the location deployment of dynamic mobile base stations. In *Computing, Networking and Communications (ICNC), 2015 International Conference on*, pages 579–583, Feb 2015.

[22] Russell Halper and S Raghavan. The mobile facility routing problem. *Transportation Science*, 45(3):413–434, 2011.

[23] Holger Claussen. Autonomous self-deployment of wireless access networks in an airport environment. In *Proceedings of the Second International IFIP Conference on Autonomic Communication*, WAC'05, pages 86–98, Berlin, Heidelberg, 2006. Springer-Verlag.

[24] Holger Claussen. Efficient modelling of channel maps with correlated shadow fading in mobile radio systems. In *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005.*, volume 1, pages 512–516, 2005.

[25] Hamid Mahboubi, Kaveh Moezzi, Amir G. Aghdam, Kamran Sayrafian-Pour, and Vladimir Marbukh. Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors. *IEEE Transaction of Industrial Informatics*, 10(1):163–174, 2014.

[26] G. Wang, G. Cao, and T. F. La Porta. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, June 2006.

[27] H. Mahboubi, K. Moezzi, A. G. Aghdam, and K. Sayrafian-Pour. Distributed deployment algorithms for efficient coverage in a network of mobile sensors with nonidentical sensing capabilities. *IEEE Transactions on Vehicular Technology*, 63(8):3998–4016, Oct 2014.

[28] H. Mahboubi, A. G. Aghdam, and K. Sayrafian-Pour. Toward autonomous mobile sensor networks technology. *IEEE Transactions on Industrial Informatics*, 12(2):576–586, April 2016.

[29] H. Mahboubi, J. Habibi, A. G. Aghdam, and K. Sayrafian-Pour. Distributed coverage optimization in a network of static and mobile sensors. In *2013 American Control Conference*, pages 6877–6881, June 2013.

[30] Walid Masoudimansour, Hamid Mahboubi, Amir G. Aghdam, and Kamran Sayrafian-Pour. Maximum lifetime strategy for target monitoring in a mobile sensor network with obstacles. In *Proceedings of the 51th IEEE Conference on Decision and Control, CDC 2012, December 10-13, 2012, Maui, HI, USA*, pages 1404–1410, 2012.

[31] Maissa Boujelben, Sonia Benrejeb, and Sami Tabbane. Interference coordination schemes for wireless mobile advanced systems: a survey. *arXiv preprint arXiv:1403.3818*, 2014.

[32] Ladan Rabieekenari, Kamran Sayrafian, and John S. Baras. Autonomous relocation strategies for cells on wheels in public safety networks. In *To appear in Consumer Communications and Networking Conference, 2017. CCNC 2017.*

[33] Sridhar Bashyam and Michael C Fu. Optimization of (s, S) inventory systems with random lead times and a service level constraint. *Management Science*, 44(12-part-2):S243–S256, 1998.

[34] Rolf Schneider. *Convex bodies: the Brunn–Minkowski theory*. Number 151. Cambridge University Press, 2013.

[35] Meisam Mirahsan, Rainer Schoenen, and Halim Yanikomeroglu. Hethetnets: Heterogeneous traffic distribution in heterogeneous wireless cellular networks. *CoRR*, abs/1505.00076, 2015.

[36] J. Habibi, H. Mahboubi, and A. G. Aghdam. A gradient-based coverage optimization strategy for mobile sensor networks. *IEEE Transactions on Control of Network Systems*, PP(99):1–1, 2016.

[37] Ladan Rabieekenari, Kamran Sayrafian, and John S. Baras. Autonomous relocation strategies for cells on wheels in environments with prohibited areas. In *under preparation*.

[38] Xiaojun Lin and N. B. Shroff. Joint rate control and scheduling in multihop wireless networks. In *43rd IEEE Conference on Decision and Control CDC.*, volume 2, pages 1484–1489 Vol.2, Dec 2004.

[39] Alexander L. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing System Theory Appl.*, 50(4):401–457, August 2005.

[40] Alexander L. Stolyar. Greedy primal-dual algorithm for dynamic resource allocation in complex networks. *Queueing Syst. Theory Appl.*, 54(3):203–220, November 2006.

[41] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1794–1803 vol. 3, March 2005.

[42] G. R. Gupta and N. B. Shroff. Delay analysis and optimality of scheduling policies for multihop wireless networks. *IEEE/ACM Transactions on Networking*, 19(1):129–141, Feb 2011.

[43] G. R. Gupta and N. B. Shroff. Delay analysis for wireless networks with single hop traffic and general interference constraints. *IEEE/ACM Transactions on Networking*, 18(2):393–405, April 2010.

[44] Bo Ji, Changhee Joo, and N.B. Shroff. Delay-based back-pressure scheduling in multi-hop wireless networks. In *Proceedings of IEEE INFOCOM*, pages 2579–2587, April 2011.

[45] Lei Ying, S. Shakkottai, A. Reddy, and Shihuan Liu. On combining shortest-path and back-pressure routing over multihop wireless networks. *IEEE/ACM Transactions on Networking*, 19(3):841–854, June 2011.

[46] L.X. Bui, R. Srikant, and A. Stolyar. A novel architecture for reduction of delay and queueing structure complexity in the back-pressure algorithm. *IEEE/ACM Transactions on Networking*, 19(6):1597–1609, Dec 2011.

[47] Bo Ji, Changhee Joo, and N.B. Shroff. Throughput-optimal scheduling in multi-hop wireless networks without per-flow information. *IEEE/ACM Transactions on Networking*, 21(2):634–647, April 2013.

[48] D. Xue, R. Murawski, and E. Ekici. Distributed utility-optimal scheduling with finite buffers. In *10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 278–285, May 2012.

[49] L. Ying, R. Srikant, D. Towsley, and S. Liu. Cluster-based back-pressure routing algorithm. *IEEE/ACM Transactions on Networking*, 19(6):1773–1786, Dec 2011.

[50] M. G. Markakis, E. Modiano, and J. N. Tsitsiklis. Delay stability of back-pressure policies in the presence of heavy-tailed traffic. In *Information Theory and Applications Workshop (ITA), 2014*, pages 1–10, Feb 2014.

[51] M. G. Markakis, E. Modiano, and J. N. Tsitsiklis. Delay analysis of the max-weight policy under heavy-tailed traffic via fluid approximations. In *51st Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2013*, pages 436–444, Oct 2013.

[52] Eleni Stai, John S. Baras, and Symeon Papavassiliou. Throughput-delay trade-off in wireless multi-hop networks via greedy hyperbolic embedding. In *Proceedings of Intl. Symposium on Mathematical Theory of Networks and Systems*, MTNS '12, pages 1–8, 2012.

[53] R. Kleinberg. Geographic routing using hyperbolic space. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 1902–1909, May 2007.

[54] A. Cvetkovski and M. Crovella. Hyperbolic embedding and routing for dynamic graphs. In *Proceedings of IEEE INFOCOM*, pages 1647–1655, Apr 2009.

[55] Ladan Rabieekenari and John S. Baras. Combining greedy hyperbolic routing with back-pressure scheduling for better network performance. In *Proceedings of Intl. Symposium on Mathematical Theory of Networks and Systems*, MTNS '16, pages 698–705, 2016.

[56] Michael J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems.* Morgan and Claypool Publishers, 2010.

[57] Leonidas Georgiadis, Michael J, and Ros Tassiulas. Resource allocation and cross-layer control in wireless networks. In *Foundations and Trends in Networking*, pages 1–149, 2006.

[58] Sprint ip network performance. `https://www.sprint.net/performance/`, 2011.

[59] Atilla Eryilmaz, Asuman Ozdaglar, and Eytan Modiano. Polynomial complexity algorithms for full utilization of multi-hop wireless networks. In *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications.*, pages 499–507. IEEE, 2007.

[60] Eytan Modiano, Devavrat Shah, and Gil Zussman. Maximizing throughput in wireless networks via gossiping. *SIGMETRICS Perform. Eval. Rev.*, 34(1):27–38, June 2006.