ROBUSTNESS IN AD HOC NETWORKS

by

YING LIU

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Wade Trappe

And approved by

____

____

____

____

New Brunswick, New Jersey

October, 2017

ABSTRACT OF THE DISSERTATION

Robustness in Ad Hoc Networks

By YING LIU

Dissertation Director:

Wade Trappe

Investigating the resilience of wireless network is critical since a network is susceptible to many types disturbances, ranging from natural (e.g. wind, building, hill, mobility) to adversarial (e.g. jamming, eavesdropping, denial of service). The notion of network resilience describes a network's ability to recover its structure should damages occur when one or more terminals are shut down or whenever sensitive information is compromised due to impersonating adversaries. This thesis addresses the following three problems related to the reduction of robustness in an ad hoc network: (1) The backtracking problem in overlay networking, which occurs during packet delivery. This condition forces a packet to traverse an extended distance due to the lack of network-layer information being fully shared with the overlay routing functions. With the presence of backtracking, the packet delivery error rate increases in relation with the traversal distance; (2) The black hole problem, which is characterized by the network being punctured, such as is caused by a jammer emitting format-compliant packets in an attempt to overflow a legitimate node's buffer. As a consequence, targeted nodes are segregated from the rest of network; (3) The reduced connectivity problem, which results from a repetitive jamming attack where the weakest link or node is selected by the jammer to be attacked.

In this thesis, the first problem is solved by redesigning an overlay routing protocol to be aware of the geographical location of cluster heads. The overlay hash table is re-organized in an order based on distance. The second problem is solved in two different ways: First, a method is presented that locates the jammer's approximate position. Reinforcement learning is applied at each node to learn the possible position of the jammer. The node's judgment is propagated by the routing protocol to interactively exchange the belief about the attacker's position. Second, a power control approach that is being used by wireless nodes is presented that adjusts the transmission power dynamically according to the Fiedler value, which is inferred from topology information carried inherently in the Optimized Link State Routing (OLSR) Protocol. By varying the radio range, packet flows can bypass the jamming area after topology transformation, making the network more resilient. The third problem is approached by developing a better mathematical understanding of connectivity when facing interference, which is achieved by analyzing the achievable throughput based on connectivity. A metric that measures the throughput and weighted throughput connectivity is introduced. Then, the connectivity issue under a jamming attack is solved by applying stochastic game theory to analyze the utility of connectivity and the interactive behaviors of both the jammer and the scanner. An optimal scanning strategy is designed to defend against repetitive jamming attacks involving different intents, and to strengthen the network connectivity at the same time.

Acknowledgements

I remember six years ago when I first stepped on the soil of United States, I was merely one of thousands of PhD students studying abroad who wanted to continue the pursuit of science. When I first decided to finish my doctoral studies in the U.S., my mind was still full of worries and unsettled feelings toward my uncertain future. Although the day of my Ph.D. graduation is near, the vivid memories of the discipline my instilled by parents before I left them still linger in my mind. During these many years of study, I realize not only that my academic skills have improved, but also my social skills through interaction with other scholars. These interactions have resulted in tremendous personal growth that has directly contributed to my excellence and academic success.

I understand that I would not have succeeded alone without assistance and support from many others. The people who I really need to give credit to are mentioned below:

First, I want to express my deep gratitude toward my academic advisor, Professor Wade Trappe, for his constant guidance, extraordinary patience, and unreserved support, which has led to my ultimate success. Our first exchange started back in 2009 even before I arrived in U.S. After he received my letter stating how I was fascinated in Network Security, he encouraged me by advising me to study at Rutgers University for my PhD I first met him was at the office of Graduate Director for ECE Department at Rutgers University when I requested a letter of recommendation from him to attend a CRA-W workshop. As Assistant Director of WINLAB, he thoroughly reviewed my case and invited me to join his research team. During the past 6 years of my PhD research, he has provided me financial support and academic insight associated with each research project carried out in the laboratory. His open-minded personality applied to realm of research allowed me to utilize my knowledge and background on networks for expansion into network-related applications with innovative ideas. Whenever I experienced obstacles or issues at home, or at the lab, he always gave a helping hand to guide me through without any complaint or impatience.

The second person on my list to thank is Professor Andrey Garnaev. Our first collaboration on research was in the fall of 2014. He became my research partner after Prof. Trappe suggested that I could improve my mathematical modeling skills. He is a very brilliant and intelligent mathematician with a specialty in game theory. With his assistance, I began my exposure to the field of game theory. He is also a patient and ingenious instructor who can magnify my small ideas and expand them into multiple possibilities. Through our interactions during research, I discovered many other interests of my own and I also improved my research discipline by implementing game theory concepts into network experiments. Without the opportunity to work with him and his expertise in game theory, my research would not have achieved the quality that it has achieved.

Next, I am very grateful to work in a competitive and friendly environment with other laboratory researchers and classmates. After countless days and nights spent at the research lab together, learning to cope with various challenges followed by finding solutions to problems is an unforgettable experience. I also want to thank all the mentors I met during my two internships, the people I viewed as competitors, and the people (including U.S. Custom and Border Agents who have repeatedly checked my immigration status before granting entry for me to resume my studies) or obstacles that might have impeded my progress toward success. Without their presence along every step of the way, I would not have grown or achieved as much of my true potential, nor would I have realized my strong desire to reach my ultimate goal of success.

Lastly, I want to thank the most important people of my life. They are my parents, who have given me unconditional love and support ever since the day I departed my hometown. As I recall, my father still had a set of dark hair before I left for studies in the U.S. Now, he ages with visible white hairs and this makes me feel regretful whenever I see my parents. This also has reminded me that they have to deal with many life-long issues themselves when I am gone. I want them to know that, wherever I am, I will always be there for them.

So, my PhD thesis is written in to the dedication of people I mentioned above. It is not the end of a wonderful journey. It is just the start of a new chapter.

“Everything should be made as simple as possible, but not simpler.”

- Albert Einstein

”The strongest people aren’t always the people who win, but the people who don’t give up when they lose.” **- Ashley Hodgeson**

Dedication

To my parents, Xinxin Liu and Guifang Xu, for their love, continuous support and encouragement.

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation

An ad hoc network is a wireless network that is composed of multiple devices interconnected to each other in a distributed fashion. These devices can either be a small mobile phone, or a larger computer with a wireless interface, or even a moving car and traffic lights in transportation system. From a network point of view, each device in the network is called a node, and functions like a router that can forward other nodes' packets or receive and send out its own packets [1]. Ad hoc networks have found application in a wide variety of scenarios, ranging from being deployed in battlefields to supporting communications as part of disaster and emergency recovery. In both cases, the primary advantage of an ad hoc network is that it does not need pre-built infrastructure to constitute the network and requires minimal configuration to route a packet. Ad hoc networks typically employ their own unique routing protocols, which can deliver packets in spite of the harsh environmental scenario, allowing for packets to be rerouted around local link failures that might arise because the communication was set up in an ad hoc fashion in the first place. The most commonly used routing protocols for ad hoc networks are Dynamic Source Routing (DSR) [2], Ad Hoc On-Demand Distance Vector (AODV) [3], Global State Routing (GSR) [4], On-Demand Packet Forwarding Scheme (ODPFS) [5].

Wireless ad hoc networks first appeared in 1970s sponsored by DARPA [6]. Ad hoc networks adopted packet switching technology with Aloha and CSMA channel access control at the link layer to provide collision free access in a distributed fashion. Starting in the 1980s, and continuing for the next few decades, the concept of the survivability of the ad hoc network was proposed and research into ad hoc networks mainly focused on network scalability, energy management and security. Moreover, many protocols were proposed to support communications in large-scale and hierarchical networks with the integration of clustering techniques. At that time ad hoc networks were mostly used in the military domain and for disaster recovery. However, more recently there has been a migration of these concepts to commercial use, particularly following the introduction of Bluetooth, IEEE 802.11 and HyperLAN [7]. Traditionally ad hoc networks were considered to be the extension to primary networks, such as Internet and cellular networks. They were deployed in infrastructure-less areas to reliably assist transmission of packets that were beyond the reach of primarily infrastructured network. Notably, this viewpoint has sustained considerable researchers' attention over the past decades [7]. With the growing popularity of designing a self-driving car, industry and government are funding research projects aimed to design an ecosystem to support vehicle-to-vehicle and vehicle-to-roadside infrastructure communications in an intelligent transportation system, which is essentially an ad hoc network. Thus, conducting researches in ad hoc networks becomes important, and it is the future in automation industry. Benefiting from the existence of artificial intelligence, many new applications require an ad hoc node to have the abilities of preprocessing the collected information, predicting the possible collisions, formulating topology to support sustainable connectivity and identifying a secure path to a destination with low latency. The information collected from each node is collaboratively mined in a local node which can possibly access history information to achieve a specific global wellness, such as secured communication, trusted network framework and energy efficiency.

There are many unsolved and open problems that exist in an ad hoc network such as network data exploration in distributed fashion, energy efficiency, resource allocation, mobility and security. They co-relate with each other with overlapped areas in certain ways. One challenging problem they all directly or indirectly face is to maintain reliable network whenever connectivity is under disruptions, and to react accordingly by applying robust strategies to assist the network to self-recover from disruption. The connectivity of an ad hoc network is fragile to natural and adversarial disturbances which is caused by the following distinct characteristics of an ad hoc network: 1) The transmission medium of an ad hoc network is wireless, making it open to the public and thus can be easily disturbed; 2) It has no pre-established facilities to support communications. Thus, its topology is facile; 3) Each node in a network can move in often a random direction or can change its course by following a particular path; 4) An ad hoc network is a resource-limited network, implying that computation of underlying algorithms should not consume too much energy or computation; 5) An ad hoc network transmits a packet in a multi-hop architecture through intermediate nodes which are unknown and untrustable. These intermediate nodes can selfishly drop packets due to lack of power or having malicious purpose to even imperceptibly read information content; 6) The membership of an ad hoc network is quite dynamic and thus adding and removing a member can cause security threats to the other members in the network.

1.2 Problem Description

Due to the previously illustrated characteristics, reliably transmitting a packet in the presence of such fragility is a significant concern for an ad hoc network. In an ad hoc network, reliably transmitting a packet includes two aspects:

1. Successfully transmitting a packet between neighbors. That is, transmitting a packet from one node to another node in its radio range which refers to a real physical link existing between two nodes, and relates to the physical parameters and collision avoidance algorithms executed in the physical and MAC layer. In this case to maintain its one-hop connectivity, a node can hear its neighbors' on-going transmission by simply performing the energy detection to avoid the packet collision;

2. Successfully transmitting a packet in a multi-hop fashion, which refers to an end-to-end delivery and corresponds to the concept of a path in the mechanisms of the routing layer and the layers above. The end-to-end connectivity of an ad hoc

network is sensitive to many interruptions that happen in each underlying layer due to the wireless medium and mobility of an ad hoc network, i.e. a natural noise or static buildings can break the connection of a link, thus, affect the total communication performance of an end-to-end transmission.

Therefore, network connectivity needs to be scrutinized carefully for ad hoc networks in order to ensure consistency of reliable performance in terms of both neighboring point-to-point and end-to-end transmission. The resilience of a network in a communications scenario involves examining the robustness of the network's connectivity as that network faces disturbances that might break some links or remove some nodes. As I described previously, these breaks can have impacts on the connectivity of both upper and lower layers in the protocol stack.

In this thesis, I have attempted to tackle the problem of robustness in an ad hoc network through several different, but related efforts. Specifically, the proposed research has involved: 1) Overlay design for network robustness with natural disturbance. I designed an overlay routing protocol that can reduce the amount of distances needed to travel for a data packet to decrease the ratio of losing a packet during transmission; 2) the localization of a jamming source in a distributed networks by an online learning methods which tolerates the data loss; 3) the adaptation of network topology to overcome a jamming attack in a distributed fashion, which integrates with the real network routing protocol to implement its defense strategy; and 4) a theoretical understanding of the physical connectivity by graph connectivity in a large scale network; 5) an understanding of the impact of a repeated jamming attack on an ad hoc network.

Strategy 1) deals with the enumerate 2. Strategy 2) deals with packet loss ratio at each local node which falls into the category of enumerate 1. Strategies 3) and 4) cope with connectivity problems of both enumerate 1 and 2 by optimally adding a bounch of directed connected links among neighboring nodes, thus improving the global connectivity performance.

In the discussion that follows, I briefly outline the underlying open problems in each of them.

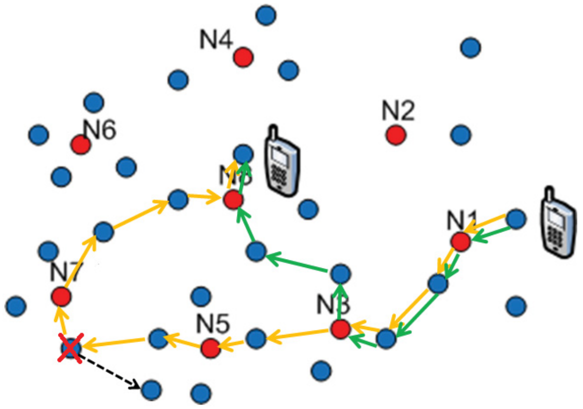


Figure 1.1: The impact of a node failure on the overlay routing and network connectivity

1.2.1 Overlay design for network robustness with natural disturbance

Routing a packet in an overlay network can face significant failure on performance degradation when the overlay routing does not take the underlying physical link status into consideration, such as is the case Chord [8]. Large-scale networks often use clustering techniques to simplify the management of routing and mitigate the impact of how control messages route through the network. An overlay routing protocol is often applied among clusters to logically route application data. However, the overlay protocol may direct packets to a cluster head where there is no connection between the previous and the next cluster head due to the movement of a gateway node or the breaking of connections between two clusters, as shown in Fig. 1.1. In Fig. 1.1, no route exists between cluster head five and cluster head seven because a node near cluster head seven has been turned off by some adversary attack, or moved to another location. However, this information is not passed to the above layer, and therefore, the overlay routing does not notice the change of the node status, and remains to route a packet to cluster head seven. The failure of packet transmission occurs quite naturally because of this scenario.

Another significant issue in overlay routing is the "back tracking" problem, because the overlay protocol does not consider the nodes' geographical location. Therefore, a packet may be delivered to a place that is at a far distance and comes back to the location near where it started. The "back tracking" causes the entire routing path to involve more links. Thus, it results in a larger failure rate in the connectivity compared to the path that is built only by the lower-layer (layer 3) “physical” routing. Moreover, more transmission time is required to finish routing.

1.2.2 Discover adversarial and design topology for robust networks

Beyond benign sources of failure, another significant concern are malicious sources of failure in an ad hoc network. A malicious attacker can utilize the impact of breaking links in the physical layer to purposely destroy the normal operation of multi-hop delivery. The most common attack to damage a link connection is jamming, where a malicious attacker proactively emits interfering power into the network in order to block the receiving operation of a terminal. This can be accomplished many different ways, such as: 1) either inject high interference power to channels on the fly. Such an attack has already been well investigated in [9] and has detrimental impact on the reliability of packet transmission. High power interference can be detected by a power detection device or algorithm; 2) or he can continuously transmit a stream of format-compliant packets without time gaps, with a low transmission power, which can not only cause interference to the legitimate packets, but also forces the receiver to remain in a receiving mode and thereby have no time to receive other’s packets. These malicious packets can also overflow the buffer in the MAC layer, while the victim node cannot send out its own packets due to the half-duplex transmission mode, and thus the victim node is effectively turned off by this MAC-layer attacker.

The MAC-layer jamming attack can cause a serious congestion around the attacking area because of the broken links on the paths. This congestion cannot recover by itself if an ad hoc network does not have sufficiently robust connectivity, and ultimately the congestion could spread out to an even wider area and eventually collapse the entire network. Moreover, a group of attackers can even collaborate to separate or partition the entire network into several isolated components if they can block the bottleneck links or nodes in the topology.

Therefore, designing a robust network topology with enough redundancy to defend against MAC-layer attacks is a key component for reliably transmitting packets in a network that is under attack. Several methods can improve the robustness of a topology, i.e. 1) Discover the position of a MAC-layer jammer or a group of jammers. After knowing the position of an attack, the network performance can be improved by dictating legitimate nodes to retreat from the jamming area or routing packets around it or even eliminate the jammer; 2) Purposely adapt the network topology so that one and several links can jump over the attack area, and make more routes available for the nodes that are not under attack.

The existence of a link in a topology of an ad hoc network refers to two nodes being in the radio range of each other, which is determined through the Shannon equation for capacity, which is formulated in terms of signal to interference and noise ratio (SINR) for the nodes by the following equation in the physical layer:

$$SINR_{i,j} = \frac{h_{i,j}P_{i,j}}{\sigma^2 + \sum_{k=1,k\neq i}^{n} N_k} \tag{1.1}$$

h is the **channel gain** that determines the values of amplitude and phase of a signal received at the terminals. Small channel gain can cause the amplitude of signal to be small so that the signal is discarded since the received Signal to Interference and Noise ratio (SINR) is below a threshold. As a result, the receiver cannot extract any useful information from the low SINR. Channel gain varies depends based on the environment, and many channel models have been developed in order to accurately describe the various channel conditions. For example, the free space channel model, Rayleigh fading, Ricean fading, Doppler shift model, etc [10–13]. It can be seen from these channel models that the channel gain is related to: 1) the signal wavelength; 2) the signal frequency; 3) the fading in the transmission path.

d refers to the **distance** between two nodes. It is common sense that the longer the distance, the more the loss will be. α is path loss exponent. α is generally larger than 2. For a free space model, it can be set to be 2. Considering the distance between two nodes can assist us in analyzing the network topology, mobility model and error occurred in decoding. Distance is a critical parameter in wireless networks. However, obtaining it is often difficult or costly since a node can only know its own position and predict the distances from itself to the neighbors in an ad hoc network. On the hand, GPS can localize the positions of entire nodes in a network. However, its measurement is often [14]. Therefore, knowing the positions of entire nodes in an ad hoc network is difficult in practice.

P is the **transmission power**. As we know, the larger the transmission power is, the more robustness for a link to any disruption it might face from noise. From equation 1.1, it can be seen that transmission power can counteract the effect of distance. However, from the whole network point of view, a larger transmission power can also cause more interference among nodes. Increasing one node's transmission power can cause interference to other nodes. For example, in equation 1.1, the term, $\sum_{k=1,k\neq i}^{n} N_k$, depicts the interferences caused by other users' signals. The larger the interferences are, the smaller SINR is. Two nodes in networks will be disconnected if SINR is smaller than a threshold. Therefore, balancing the transmission power and interferences in a network is important. The improvement of the total throughput of a network is uncertain if all nodes increases their transmission power. However, the total throughput of a network will decrease if the topology becomes more disconnected due to the lack of the connections among critical nodes. Therefore, the nodes' power can be heuristically adjusted to raise the connectivity of critical components. Many papers in the literature investigate methods of adjusting transmission power in order to increase the throughput of a whole network [15–19]. [16, 17] uses game theory to study the impact of a jammer or group of jammers on the total network throughput. Although the interference among users are not considered, the jammers' impact has the same influence as interference among users.

σ^2 and N_k refers to **noise and interference** which are quite frequent and substantial in wireless networks. Noise and interference can be categorized into natural and adversarial disturbances. The environmental noise and normal interferences among users belong to the natural disturbance. They always exist in networks. Even if one carefully designs the topology, the natural noise always exists. On the other hand, artificial noise is launched by a node for different purposes. For example, a jammer can purposely launch an interference signal at a specific node in order to shut down the target node. On the other hand, the legitimate user can emit an artificial noise to deteriorate eavesdropper's channel condition in order to protect the normal transmission.

1.2.3 Understanding connectivity with interferences

To understand the connectivity under presence of interferences is important for network deployment and protocol design. This is especially true for a node in an ad hoc network who has only limited power since transmitting a packet through stronger links can save network energy by lowering the number of retransmissions. However, natural noise and interference created by other nodes always exist in wireless communications which is one of the major reasons to cause disconnection from a link in physical layer. Thus, understanding its impact on connectivity of upper layer protocol stack is necessary and instructive for giving a global view of designing network architecture.

Moreover, due to multiple sources of interference in a network, the impact of interference on network connectivity becomes complicated, which makes it unintuitively by just considering the strength of one link. By common sense, increasing transmission power improves the strength of a link, thus improves connectivity. However, such increase of transmission power can also inadvertently break links of other nodes. Thus, connectivity and throughput of whole network become blurry due to the addition of one link which can simultaneously cut off one or several other links meantime. The connectivity becomes uncertain especially when several nodes simultaneously increase their transmission power. On the other hand, the position of a node in network is also critical to network connectivity. The low power node may become bottle neck of a connection. Thus, maintaining and improving its connection is important. Whereas, strong power node may be found in the low density area, and increasing its transmission power has no obvious impact on already fully connected network, which just wastes its energy.

The ultimate goal to increase connectivity is to improve the network throughput. Due to uncertainties of interference, the throughput may not be positively proportional to connectivity. Currently, there is no closed formulation to truly depict the impact of interference on network connectivity and, thus, throughput under this connectivity. Therefore, modeling this complicated relationship is essential. Several other open problems can utilize this formulation to design control strategies such as routing strategy to achieve maximum throughput connectivity, resource allocation among network nodes, and optimal network deployment. Later, Chapter 5 specifically discusses the mathematical solution for this kind of problems.

1.2.4 Connectivity with uncertainty of an adversarial attack

Connectivity under the jamming attack is uncertain due to the uncertainty of the attacker's behavior. Therefore, these uncertainties brings difficulties to improve network robustness. Only adjusting transmission power cannot achieve the maximum robustness of the connectivity since the interferences among legitimate nodes varies also with the behavior of an attacker. For example, a jammer can choose any node in a network to attack and can change its attack position in the next round. Thus, any adjustment which is optimal for the current situation can be sub-optimal in the future. In addition, the attacker can choose whether to continue his attack. If the attack ceases, continuing increasing transmission power is unnecessary.

Further, a jammer can also be intelligent, or random with simple purposes. An intelligent jammer can change his attacks according to different heuristics such as destroying the connectivity as much as possible or largely decreasing the network throughput or disclosing private information from legitimate users. Moreover, an intelligent jammer can alternate his heuristics for executing attacks according to rules which can benefit himself or just aim to confuse the legitimate user to protect being caught. Therefore, the legitimate users have no clue of the strategy used by a jammer at a specific time. They may only know the set of strategies which the jammer can adopt.

Thus, these uncertainties associated with an attack complicate the process of understanding network connectivity. Also the dynamics of a jammer's strategy with time becomes an interesting topic to design a defense strategy with both short and long time benefits. Therefore, understanding the connectivity with attack uncertainties can give a theoretical guidance for designing a secure and robust network.

1.3 Contributions of Thesis

This thesis addresses the robustness issues that appear in an ad hoc network and specifically focuses on the resilience of network connectivity. I approached the problems from three aspects:

1) Design a distance-aware overlay routing to reduce the probability of unreliable transmission caused by the “back tracking” path and the natural disturbances such as the variation of channel conditions and the movement of nodes. In this method, I designed a routing algorithm in a over-layer protocol composed by cluster heads. I made use of the nodes’ geographical location to guide the over-layer routing to prevent the unnecessary long path so that it indirectly increases the network connectivity since distance in physical layer is one of critical factors indicating whether a communication link can be established between two nodes.

2) Develop algorithms to overcome the problem of breaking links caused by adversarial disturbances such as a MAC-layer jamming. Although each algorithm cannot solve all the problems, it can address one side of the story. The methods I used are:

1. Find the location of disturbing source. I proposed a Q-learning method to localize a jammer in a distributed fashion, and I integrated it with the OLSR protocol in NS3 a popular network simulator. Simulation results proved that our algorithm was able to locate a MAC-layer jammer who produced congestions in network by maliciously turning off a node.

2. I proposed a Fiedler value power adjustment algorithm to improve the connectivity of overall network topology. Fiedler value is an indication of the connectivity of entire network topology. It is the second smallest eigenvalue of an adjacency matrix according to the graph theory [20, 21]. Deleting each node in the adjacency matrix and calculating the Fiedler value from the remaining matrix can imply the connectivity-weakest nodes in networks. The network connectivity can be strengthen by only increasing the transmission power of the node

who has the least Fiedler value. The adjacency matrix is timely updated by OLSR protocol through hello message.

3) Understand the connectivity of an ad hoc network mathematically with influence of both natural and malicious interference. I approach this problem in two ways.

1. Apply graph theory to analyze the variation of connectivity and throughput achieved through connectivity with the existence of interference. The closed-form equation to depict connectivity in terms of parameters in physical layer is provided. From the observation, throughput no longer continuously increases with the increase of transmission power with the presence of interference. Moreover, an optimal allocation of total transmission power to achieve maximum throughput connectivity is investigated. The result reveals the importance of collaboration among nodes to achieve maximum network throughput with the limited total transmission power.

2. Analyze the Nash equilibrium of connectivity when the network is suffering from a major damage by a malicious attacker. Then, understand the interaction between legitimate users and the connectivity jammer by game theory. Stochastic game is applied to model the interaction that affects dynamics in connectivity for the case that a network is under attack for an extensive amount of time. Under this game framework, a preventive strategy is provided.

1.4 Outline of Thesis

The structure of the thesis is shown in Fig.1.2 and it is presented as follows: Chapter 2 - I present the details of over-layer routing design to overcome the natural noise with the awareness of geographical distances among cluster heads. Chapter 3 - Q-learning is combined with OLSR to localize a MAC-layer jammer in an ad hoc network. Chapter 4 - an adaptive cross-layer power allocation is proposed to increase connectivity in routing layer of an ad hoc network in order to defend against a jamming attack. Chapter 5 - the concept of throughput connectivity is presented to theoretically understand

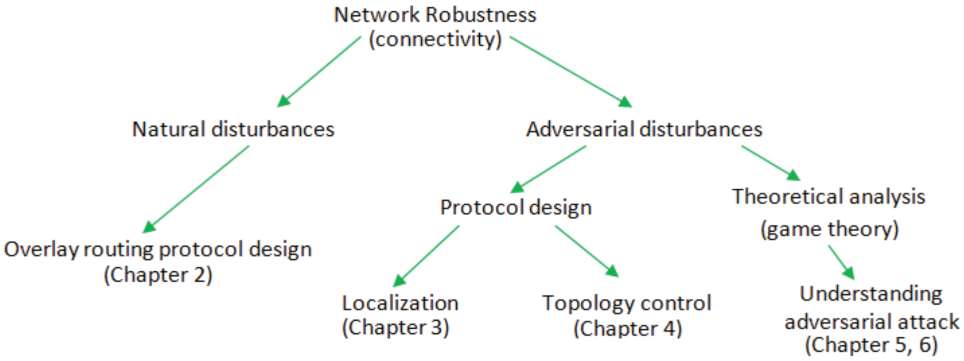


Figure 1.2: Overview of the thesis

the dynamics of connectivity with the existence of interference and the influence of parameters in physical layer. Chapter 6 - the connectivity jamming game is discussed, and a protective strategy is provided. Chapter 7 - an overview of possible methods to utilize artificial intelligence in the analysis to solve connectivity problem in an ad hoc network. Chapter 8 - conclusion made for possible future directions are discussed. Moreover, each chapter can be read independently without referencing to other chapters.

Chapter 2

Distance-aware Overlay Routing in Large Scale Ad Hoc Networks

2.1 Overview of the chapter

Overlay networks are a beneficial approach to designing robust and specialized networks on top of the generic IP architecture, and have been applied to the operation of mesh and mobile ad hoc networks. Unfortunately, when routing between entities in the overlay, inefficiencies are incurred due to potential “back tracking” that arises because of the discrepancies between the overlay and underlay topologies. In this chapter, I minimize the “back tracking” problem by applying physical contexts shared by the network layer with the overlay so as to efficiently guide application flow. I have devised an intelligent cluster head and path selection algorithm for our overlay routing and compared its performance with the popular Chord protocol and a baseline AODV routing protocol. Simulation results indicate that: 1) the integration between logical and physical routing gives a large improvement in the number of hops for each transmission path; and 2) the selection of a good cluster head has only a moderate increase in transmission time.

2.2 Introduction

Mesh and ad hoc networks are a means for mobile users to rapidly establish a network without a dedicated infrastructure. They have been proposed for a variety of scenarios, ranging from tactical networks to supporting urban infrastructure to sensor network deployments. Popular Internet applications, such as Skype and Youtube, will become increasingly prevalent on mesh/ad hoc networks. These popular applications require that communications experience limited delay. To support such applications in areas where there is no wide-area communication coverage (e.g. in tactical environments, or in remote areas removed from cellular coverage), it is necessary to employ peer-to-peer communication methods, such as mesh or mobile ad hoc networking (MANETs). This work aims to reduce the delay associated with packet delivery for peer-to-peer communications in large ad hoc networks, while maintaining reliable communication.

Unfortunately, the basic "physical" routing that is performed at the network layer does not scale well versus the number of nodes in an ad hoc network, and there are significant network-layer routing challenges that should be hidden from applications, e.g. in table-driven proactive ad hoc routing, the communication overhead scales as $O(n^2)$, where n is the number of nodes in the network. Similarly, for on-demand routing, although routing information is transmitted only when needed, these protocols flood the route discovery messages (RREQ) which introduces a heavy load. One approach to work around non-scalability in ad hoc networking involves hybrid routing, such as ZRP [22], which combines IARP proactive and IERP reactive routing. Another solution involves using a hierarchical network structure, where nodes are grouped into several clusters via clustering algorithms like k-Means clustering or BFR (Bradley-Fayyad-Reina) clustering [23]. The cluster head serves as a gateway and thus the clustering technique decreases the network flow between nodes. Routing among cluster heads is carried out at a higher-layer, via an overlay, leading to what will be referred to as "logical" routing in this chapter. Logical routing is more application-oriented and usually does not take into consideration the physical routing underneath.

This chapter explores the performance of hierarchical routing at the overlay level when used with the popular AODV routing protocol [24]. We adopted a cluster-based architecture and the applications we are concerned with are those involving latency constraints, such as peer-to-peer (P2P) VOIP applications between clients in an ad hoc network. Many of these P2P applications are built using an overlay protocol, such as Chord, which is suitable for P2P file sharing and web searching [25]– applications that do not demand strict time delivery. For applications like VOIP, Chord's performance cannot meet the required QoS delay constraints. This arises because Chord routing is unaware of the relative geographic position of two callers, and the effect is that standard Chord routing at the overlay leads to significant network-layer "back tracking". For example, packets that should be delivered to New York from New Jersey are first routed to California which is geographically far away and then forwarded back to NY state. Further, in ad hoc network settings, Chord-based routing does not take advantage of other cross-layer information available that might be desirable to use when forming routes, such as network topology, device power resources, or even link quality. Our contribution in this chapter is to address the "back tracking" problem by building a distance-aware overlay routing protocol. We use physical connectivity to guide our logical routing and implement a distributed distance table (DDT) in a manner similar to the distributed hash table (DHT) in Chord to find the shortest path among cluster heads, where the underlay routing protocol is AODV. In Section II, we give a brief introduction to Chord and AODV, which are closely related to our protocol. In Section III, we examine reasons why it is desirable to take the underlying geographic distance into consideration when routing. After the analysis, we describe our context-aware routing protocol in detail. In Section IV, simulation results are presented, and the chapter is concluded in Section V.

2.3 Related Material

In this section, we briefly summarize Chord and the AODV protocol, which serve as the motivation and basis for this work. Chord [8] is a P2P overlay routing protocol based on distributed hash tables (DHTs), and is generally considered one of the four DHT protocols, along with Tapestry [26], CAN [27] and Pastry [28]. In Chord, nodes in a network compose a logical ring, which can contain at most 2^m nodes. m is the length of node identifier. These nodes are uniformly positioned on the logical ring in a random order. Each node has its own unique IDs and "keys". The node ID is mapped to the DHT by the hash function: $n + 2^{k-1} \bmod 2^m$, $k \in [1, m]$, where n is the current node ID. When $k = 1$, we get the ID for the first successor of n, when $k = 2$, we get the second successor, and so on. This hash function makes adding and removing a node in DHT collision free. Each DHT table can contain up to m entries. Each node maintains a separate DHT table when nodes dynamically join and leave the chord circle. During the routing process, the source node first finds whether the target node ID is in its finger table. If not, it finds the closest predecessor to the target node in its DHT and then passes the request to that node. This process occurs recursively until the target node is reached.

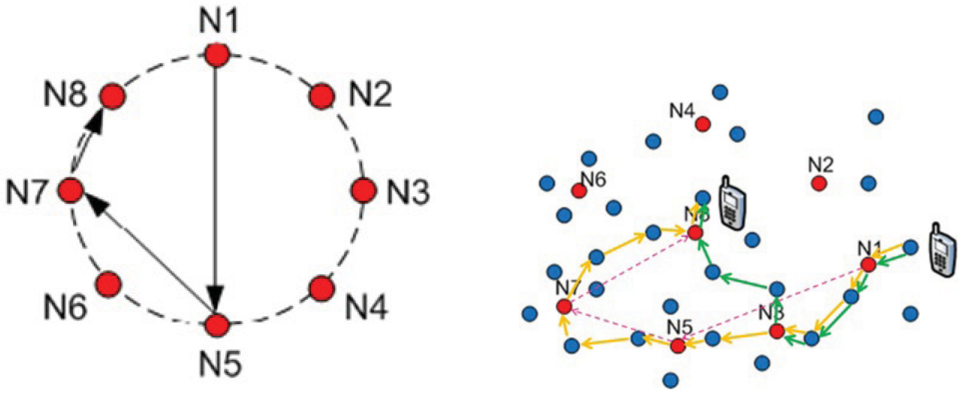
The ad hoc on demand distance vector (AODV) routing protocol [3] is one of the more popular mobile ad hoc network routing protocols, along with OLSR [29]. In AODV, when a connection request occurs, the source node broadcasts a route request (RREQ) message, which carries a source identifier, a destination identifier and a destination sequence number (DestSeqNum), to its neighbors. Then the neighbors flood the RREQ to other nodes for route discovery. Any intermediate node that receives the RREQ can either forward the request or send a route reply (RREP) to the source node if its routing table has a valid route entry to the destination. One of the primary advantages of AODV is that the route is built *on demand*, which significantly decreases the network load, while disadvantages for AODV include higher latency for finding routes and overhead associated with sharing stale route entries.

2.4 Context Aware Routing

2.4.1 Why distance aware

Chord does not consider the physical distance between nodes, and rather places all nodes randomly on the Chord ring. Therefore, when performing routing using Chord, packets can be directed to an area that is far from the ideal source-destination path, which results in additional transmissions and extra latency shown in Figure 2.1.

Another weakness that results in Chord having a higher failure probability is that Chord routing does not consider the existence of degraded links in the underlay. The lack of a connection to actual network link performance can lead to large failure rates.



(a)

(b)

Figure 2.1: (a) Chord overlay routing; (b) the physical route that a packet traverse.

We now examine the failure probability of delivering a packet using Chord. We first assume every cluster head ID is on the Chord ring so as to exclude from our analysis the failure probability of being unable to find a node ID on the Chord ring. Therefore, the failure probability of one logical link on the ring is solely caused by the failure probability of underlay's broken links. Assume r is the expected number of physical links that each logical link contains and q is the success rate of each physical link that every logical link contains. Then the probability for one logical link to fail is equivalent to anyone of the composed physical links failing. The expected probability for one logical link to fail is

$$E(p) = (1 - q^r) \tag{2.1}$$

While the probability a node cannot find a destination node on the logical ring is $p^{\log N}$. N is the number of nodes on the Chord ring. Then the final probability for a node to be unable to deliver a packet to the destination is

$$Q = (1 - q^r)^{\log N} \tag{2.2}$$

The successful transmission probability q can be represented by the distance between nodes. For simplicity of calculation, we use the bit error rate (BER) for BPSK as a surrogate for packet error rate $1-q$. To recall, the BER for BPSK is

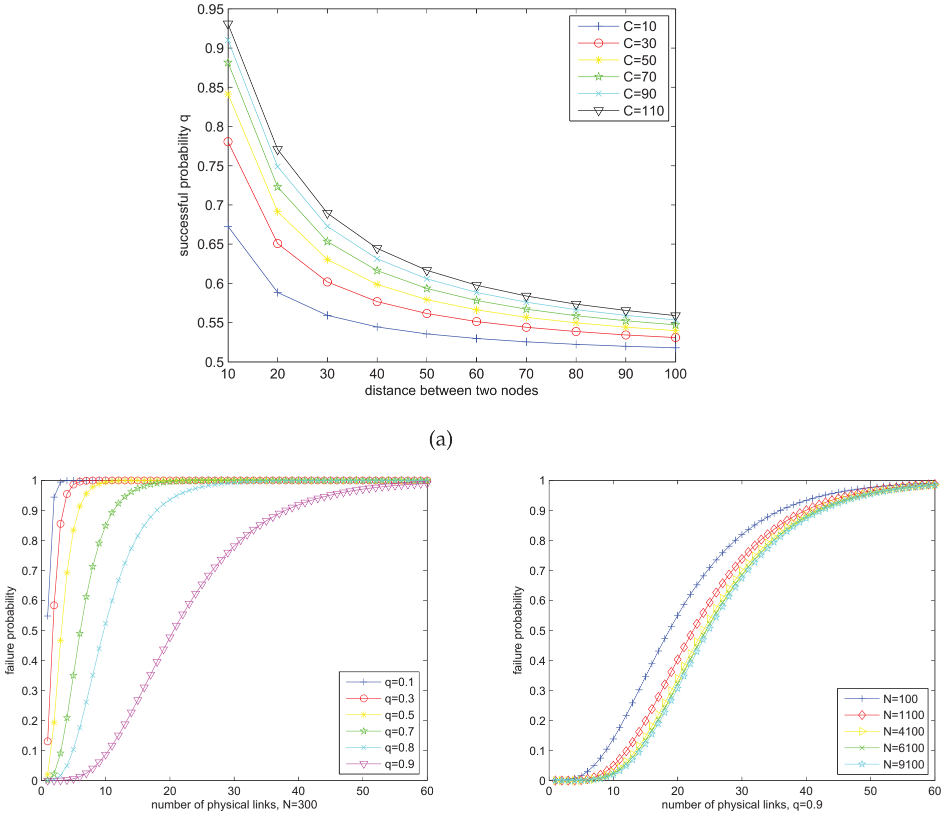
$$BER = \frac{1}{2}erfc(\sqrt{\frac{E_b}{N_0}}) \tag{2.3}$$

where $\frac{E_b}{N_0} = SINR\frac{B}{R_b}$ and B is the bandwidth of the signal, and R_b is the transmission bitrate. The BER decreases as $\frac{E_b}{N_0}$ increases, and thus, with B, R_b fixed, $\frac{E_b}{N_0}$ is proportional to $SINR$. In our work, we have used the Friis equation to arrive at the SINR estimate for a pair of nodes with a separation d, which yields the successful probability q for one physical link as:

$$q = 1 - BER = 1 - \frac{1}{2}erfc(\sqrt{\frac{C}{d^2}}) \tag{2.4}$$

Here, $C = \frac{P_t G_t G_r \left(\frac{\lambda}{4\pi}\right)^2 \frac{B}{R_b}}{N_0}$ and incorporates transmit power P_t, and transmitter and receiver gains G_t and G_r. Using these equations, we illustrate the relationship between geographical distance and physical link success in Figure 2.2.

Figure 2.2 (a), shows that the success probability of one physical link decreases when the distance between a pair of nodes increases, and thus distance affects the design of the overlay routing architecture. In Figure 2.2 (b), the failure probability of a physical path built by Chord increases substantially when the expected number of physical links composing one logical link increases. It is also shown that the high successful probability of one physical link could allow one logical link to accommodate more physical links. For example, to maintain a failure probability below 0.2, for the number of nodes $N = 300$, it is better to maintain the number of physical links between each cluster head below 14 as q is below 0.9.Although the failure probability of Chord, $O(\frac{1}{N})$, decreased with N, N has little effect on the failure probability of a physical path because there is also a relationship involving q and r. For example, to maintain the failure probability below 0.2, for N up to 9000 with success rate $q = 0.9$, the expected number of physical links should be around 15 hops.



(a)

(b)

Figure 2.2: (a) The probability that a physical link experiences failure increases versus increased node separation distance d. (b) The physical path success is related to the probability of a physical layer link being successful q, as well as to the number of physical links, but shows limited dependence on the number of nodes in the network.

2.4.2 Overlay Topology Structure

Since we intend for our overlay routing algorithm to work on large-scale networks, we need to employ hierarchies in the overlay topology. A depiction of the network topologies in this work is presented in Figure 2.3. For a given ad hoc network, in the overlay nodes are grouped into clusters. For cluster formation, we group nodes according to their mutual distance from each other. Each cluster has its own cluster head which is chosen by a cluster head selection algorithm. One possible cluster head selection algorithm is to choose the node with the largest remaining power energy in the cluster as the cluster head, or choose a node as the cluster head who could lead to the minimum total energy consumption [30] [31]. We note the algorithm presented

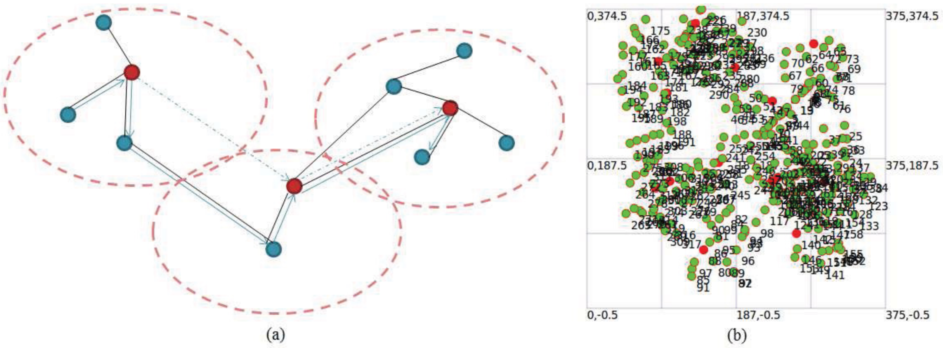


Figure 2.3: Overlay and underlay topology: (a) the solid line represents the physical underlay links between nodes, while the dashed line represents the overlay logical link between cluster heads; (b) is a topology used in our simulations, the red circles were cluster heads, the green circles were ordinary nodes.

in [31] also considered balancing the traffic loads for the cluster head in order to reduce the total amount of energy consumed by the whole cluster. As another example, the overlay could simply choose a cluster head which is closest to being the geographical center of the cluster. In our design, we wanted the overlay routing to minimize the path length of the underlay network being traversed, and thus an ideal cluster head was the one that had the minimum number of hops to every other node in a cluster. The objective function in our case is:

$$CH = argmin_{i\epsilon S} \sum_{j\epsilon S \setminus i} hop(i,j) \tag{2.5}$$

where S is the collective set containing all the node IDs in the cluster. $hop(i,j)$ is the number of hops from node i to node j. $S \setminus i$ refers to a set which does not contain i.

2.4.3 Overlay Routing with AODV: DDT-AODV

Modifying Chord so that it becomes distance-aware is not a simple task. For a new overlay routing protocol to be distance aware, we cannot simply change the Chord DHT table to contain a list of the k of nearest cluster heads without also needing to change the hash function associated with finding closest preceding node. Because the closest preceding function searches the nodes in reverse order and found a node which was farthest to the source. In order to reduce the back and forth, or inefficient paths as mentioned previously, we need to introduce a new process. The new process is to determine the precedence of a node based on its distance from the starting point. The precedence ranks its corresponding nodes in the order based on their relative distance to the starting point, with the shortest distance node on top of the list. The optimal path is the node selected based on its shortest distance from the starting point where the route request was first initiated. Therefore, the starting point for calculating the distance changes every time a route request appears. This alteration could result in a huge computational overhead.

In our algorithm, each cluster head stores the k nearest cluster heads and builds an adjacent link list among these cluster heads. An adjacency link list corresponds to one topological graph among the cluster heads. The overlay routing found the shortest path in this topological graph. We note that this topological graph could either be directed or undirected. All that matters are the entries of the DDT table. We could also set the graph to be completely connected or partially connected or sparsely connected. This could be done by tuning the length, k, of the DDT table. Another advantage of tuning the length k is that it also let us make a tradeoff between the number of hops of the logical path, which is related to packet delivery time, and the complexity of finding a shortest path.

In our routing scheme, DDT-AODV, we use AODV for network layer routing. When an application sends a data packet to a node, the cluster head updates geographic positions of other cluster headers through the RREQ and RREP of AODV, which is easily modified to convey location information (e.g. GPS). Each cluster head maintains a distributed distance table (DDT) containing up to k nearest cluster heads. This was done using the RREQ and RREP message of AODV. We modified the RREQ message to carry the position of nodes it encounters as it traverses the network. The cluster head also contains other cluster heads' distance tables. The update of another cluster head's DDT table was also conveyed through the RREQ and RREP of AODV. A second approach is for each cluster head to send their distance tables to a back-up server. The back-up server then takes care of updating the DDT table of the cluster heads and the calculation of the logical path between source and destination heads according to the new DDT. The updating of DDT need only happen when there's a routing request. After figuring out the logical path, the back-up server sends this logical path to the corresponding cluster heads on the logical path.

We adopted the first method since we wanted to avoid having a single point of failure in the network. Our main goal was to compare the length of the path after it was built and the failure probability caused by ignoring the underneath physical link status. When there is a search request, each cluster head updates their distance tables. Only the newly updated entry of the routing table needs to be passed. Then the cluster head calculates the shortest path between the source and destination cluster head and hands over the logical path to the underlay (AODV) for actual network layer routing. The header of the AODV data packet contains the logical path for routing purposes when transmitting. The source node first wraps the data packet and tunnels it to the next cluster head on the logical path. Then the received cluster head unwraps the packet and sets the next cluster head on the logical path as destination and tunnels it out. This process run continuously until data packets reach the destination. AODV finds the real physical path to connect each cluster head on the logical path. Since we assumed each cluster moved very slowly, the re-building of a path did not occur frequently. Figure 2.4 shows the whole routing mechanism across overlay routing and AODV. Finally, we note that other metrics, such as channel bandwidth, data rate, and node energy could be used to rank the entries of DDT table.

2.5 Simulation Results

Our simulation results were obtained using the popular network simulator, NS-3. We used networks consisting of 320 nodes. These nodes were placed in a square area with length 500 units. Several topologies were produced, each composed of 16 clusters and each cluster contained 20 nodes. We required that, within each cluster, the average hops between node and cluster head should be larger than 4, yet no larger than 15. We also set the radio range for each node to be 20 units so that two nodes were not connected when the distance between them was more than 20 units. Under these

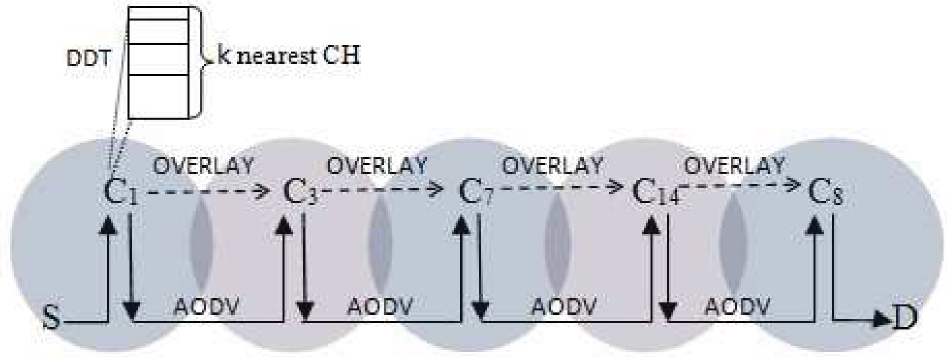


Figure 2.4: In context-aware overlay routing each cluster head stores a distributed distance table that contains k nearest cluster heads. Overlay logical routing is taken between cluster heads. AODV routing was used to connect cluster heads on the logical path. The colored circles represent cluster regions, which may overlap.

requirements, we produced 15 topologies. We compared our context-aware protocol with Chord using these 15 topologies. Since our goal was to prove the hop efficiency of the routing protocol, we fixed the cluster heads for each topology at the start of the simulation and kept the cluster heads fixed during the simulation. The ideal cluster head has the minimum average number of hops to every other node within the cluster since the upper layer application, such as VOIP, require low latency delivery of voice packets. We also wanted to see the effect of a “bad cluster head” on increasing routing hops, so “bad“ cluster heads were randomly chosen within a cluster.

For the network layer, AODV was implemented. The length of the DDT table was set to be 5. We compared our DDT context-aware routing with AODV routing without clustering and Chord-AODV with min-hop and a random cluster head (CH). Each simulation was run 100 times. Figure 2.5 shows the average number of hops a packet went through for each protocol. The simulation demonstrated that our distance-aware DDT routing gave much fewer average hops than Chord for cases where Chord-based routing used mini-hop and random cluster head selection. Even with a bad cluster head, DDT-AODV performed almost same as Chord with good cluster head. Compared to Chord, DDT-AODV with minimum hop cluster head selection had performance closest to the ideal of using the baseline AODV. To present clearly, Figure 2.6 shows the additional cost of DDT-AODV and Chord-AODV compared to baseline

AODV, which is calculated as:

$$\frac{average\_hop - baseline\_AODV\_hop}{baseline\_AODV\_hop} \tag{2.6}$$

Figure 2.6 implies that DDT-AODV with minimum hop cluster head results in the least additional hops when compared with Chord-AODV. Although Chord with mini-hop cluster head selection sometimes performed better than DDT-AODV, its performance was unstable and inconsistent across topologies. Chord had a larger variance compared to DDT-AODV with good cluster head selection. This big variance was caused by Chord protocol positioned node randomly on the Chord ring without the consideration of the geographic position of nodes. Therefore, some extra hops occurred in the network path. This extra number of hops had a large variance because of the underlying randomness. This large variance reveals the general behavior associated with “back tracking” in Chord-selected paths. We also conclude that choosing a good cluster head according to application requirements affects the performance of routing protocols. A good cluster head was one which improved the QoS of the applications running on top of it. For voice applications, the simulation results indicated that cluster head which considered actual physical distance was a wise choice. Although DDT-AODV does not perform the best for all cases, its average behavior is better than Chord-based methods and this translates into it having desirable delay properties, making it suitable for multimedia applications.

Figure 2.7 shows the path failure probability for different routing schemes for the 15 topologies. In Figure 2.7, DDT-AODV with minimum hop cluster head had a low failure probability compared to Chord. Its failure probability was always below 0.2, showing consistency across topologies. However, Chord failure probability was frequently large. The reason for Chord’s large variance and inconsistency was the “back tracking" caused by nodes randomly positioned on Chord ring without considering the underlying broken links. The logical path it chose could not successfully deliver the packet to the destination when the AODV underlay could not find any network

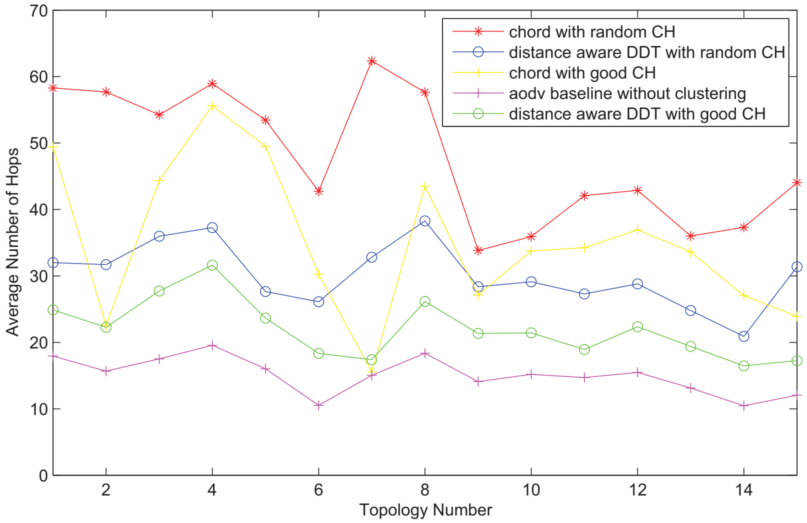


Figure 2.5: Average hops of DDT-AODV, Chord-AODV and baseline AODV for 15 topologies

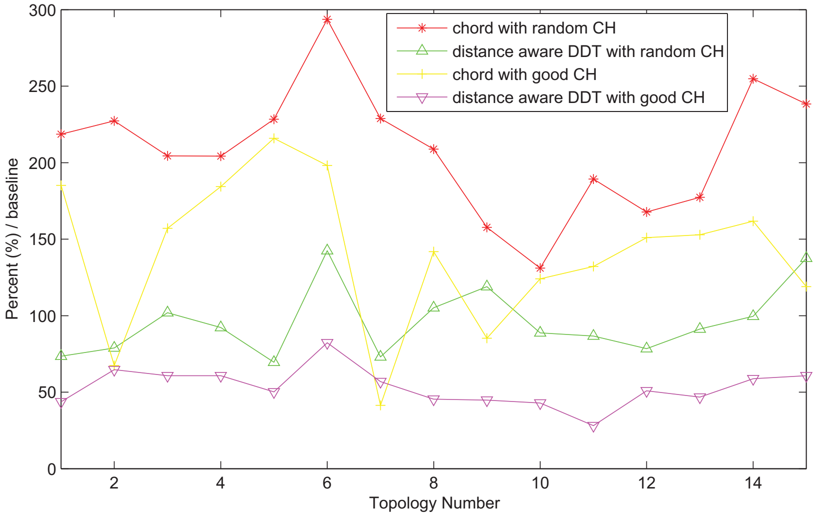


Figure 2.6: Additional cost of DDT-AODV and Chord-AODV for 15 topologies

path connected to the destination. The longer path, as expected, has an increased failure probability since the failure probability of the whole path depends on the number of physical links.

When comparing the effects of cluster heads, choosing a good cluster heads was more important for DDT-AODV than Chord. The failure probability of DDT-AODV with minimum hop cluster heads was always under that of DDT-AODV with random head selection, while Chord's performance was almost the same regardless of cluster

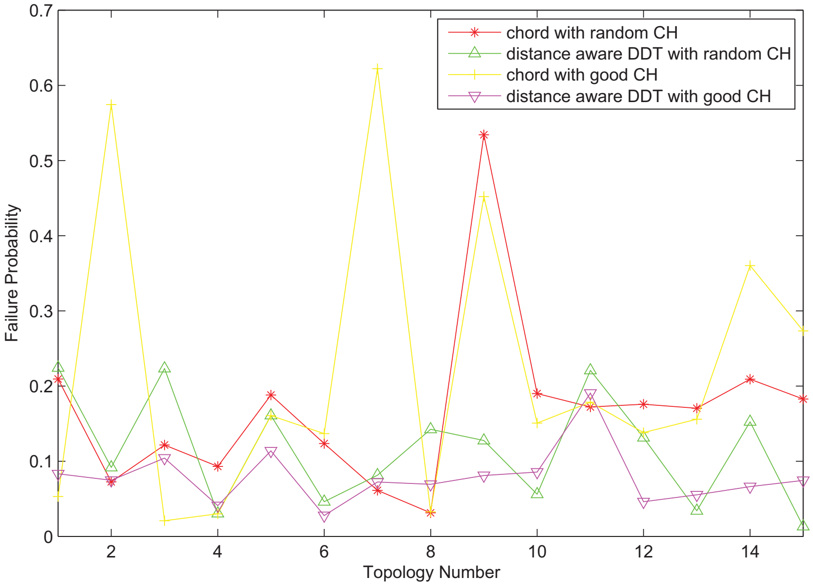


Figure 2.7: Failure probability of DDT-AODV and Chord-AODV for 15 topologies

head selection.

2.6 Conclusion

We analyzed the weakness of Chord-based overlay routing in ad hoc networks. We pointed out that Chord routing usually experiences a “back tracking” problem since it randomly distributes nodes on the Chord ring without consideration of the physical/network position of nodes. We proposed a distance-aware overlay routing scheme that uses a lower layer AODV routing protocol to guide network-level packet delivery. Our protocol, DDT-AODV, involves cluster heads updating each other with their physical position and ranking the closest cluster heads in their DDT table. When there is a route request, logical routing is first calculated among cluster heads then passed to the lower tier nodes. Lower tier nodes use AODV to find the network path between themselves and next hop cluster heads. NS-3 simulations showed that: 1) DDT-AODV out-performed Chord, both with mini-hop and random cluster heads, in terms of additional cost; 2) DDT-AODV with good cluster head selection had dramatically reduced failure probability. DDT-AODV always showed consistency in performance across different topologies: the differences in terms of average hops for 15 topologies were small compared to that of Chord-AODV.

Beyond the proposed distance-aware overlay routing protocol to counteract the natural distances caused by the environmental changes and node's mobility, the following chapters consider different approaches to cope with the adversarial attack, namely, the physical jamming attack. I will provide defending strategies both in terms of theoretical and practical views. Firstly, the next chapter demonstrates how a Q-learning method can repeatedly learn the jammer's location in a totally distributed manner through the guidance of networks' performance metrics.

Chapter 3

Jammer Forensics: Localization in Peer to Peer Networks

3.1 Overview of the chapter

Jamming attacks are a class of network denial of service attacks that can easily be carried out in wireless networks. In order to be able to repair a network in the presence of such attacks, it is desirable to identify the location of jammed nodes and the congested area that is affected by the jammer. In this chapter, I propose the design of a Q-learning based attack-localization algorithm that is integrated with the OLSR routing protocol. Our Q-learning attack-localization algorithm is distributed, asynchronous and can identify the location of the jammer in run-time as the attack takes place. I examine the performance of our approach using NS3 network simulations under two different network topologies, and for both naive and intelligent attack scenarios.

3.2 Introduction

Wireless communications are easily be subjected to interference attacks that reduce the effectiveness of the network and its protocols to reliably deliver data from a source to a destination. Interference attacks can be performed in many different ways, ranging from emitting a high energy interference signal, which degrades physical layer capabilities to decode modulated information, to the intentional use of MAC-layer congestion that prevents nodes from transmitting or receiving [9]. One of the challenges that must be addressed to ensure that networks can survive in the presence of malicious interference or congestion is to locate the jamming source [32, 33]. Once the location of the adversary has been determined, then remedies may be applied to ensure the network can operate reliably. For example, the location of an adversary may be fed

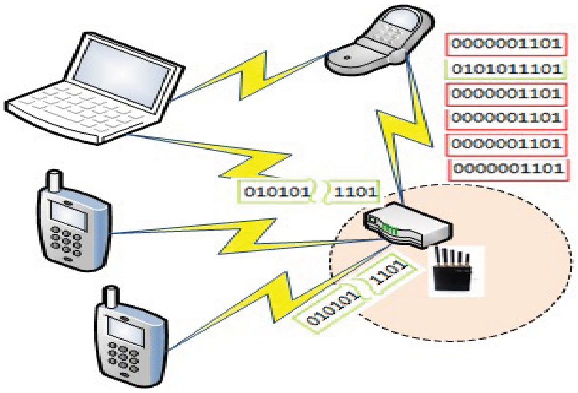


Figure 3.1: Congestion-style MAC layer jamming attack

to network layer (e.g. routing) functions and used to alter routes in the network, or the location of an adversary may be used to adjust power or channel allocations for network nodes near the adversary.

Localizing a moderately high-power interference source is a problem that has been relatively well-studied [34–39]. It must be realized, however, that an attacker need not apply large amounts of power to adversely impact the network's performance. In fact, a small amount of transmit power being employed by a congestion-style interference source, which continually emits format-compliant packets, can be quite effective at shutting down the MAC-layer functionality of neighboring nodes in a network shown in Figure 3.1. The implication of this observation is that simple schemes, such as those that employ received signal strength (RSS), have limited ability to locate an attacker. For such attacks, it is necessary to analyze the collection of network statistics *across* the entire network in order to locate the adversary. These network statistics are dynamic *signals* associated with the network's graph that provide forensic evidence regarding the adversary's presence and must be cleverly leveraged in order to locate the adversary.

In this chapter, our goal is to localize the legitimate network node(s) that are closest to the source of congestion, i.e. the attacker. Our approach operates in a distributed fashion, and involves an online learning algorithm that has been tailored to locating a congestion-style adversary in run-time. Specifically, we have modified the well-known Q-learning algorithm, and the advantages of our method are: (1) the algorithm is distributed (Q-learning runs locally at each network node, each node only needs to know network statistics for itself and its one-hop neighbors); and (2) the algorithm is asynchronous and can converge even if some data available is missing or old (as long as the obsolete information eventually vanishes with time).

The chapter is organized as follows: In Section 2, we provide an overview of our problem and the network scenario. In Section 3, we briefly overview Q-learning and then describe how Q-learning was integrated with a routing protocol to support network forensics, including detection and isolation of the jamming attack. We next analyze the performance of our jamming localization algorithm in Section 4 through simulations involving different attacker models. In Section 5 we summarize related work, and conclude the chapter in Section 6.

3.3 Related Work

To find the location of an attacker, or the node under attack, is important and can serve as the basis for repairing the network [40–42]. Previous work on finding the area that is jammed [43–45] can be classified into three categories. The first category involves measuring the received signal strength and using propagation modeling to estimate distance. For example, [46] utilized the variation of the hearing range of a node under attack and its affected neighbors to formulate equations from which the jammer can be localized. The second category obtain the jammer's location by utilizing information about the geometric location of affected nodes. The typical examples for this category are the conventional Centroid Location (CL) [47], Weighted Centroid Localization (WCL) [48] and its improved versions, Virtual Force Iterative Localization (VFIL) [33] and Double Circle Localization (DCL) [44]. Unfortunately, both the first and second categories require the exact locations of neighbor nodes. The third category, however, uses the *relative* position of nodes in the network, and one example of this category is [49], which uses gradient descent to find the node with minimum Packet Delivery Ratio (PDR). One unfortunate behavior of such an approach is that it may be trapped in local minima during the searching process. The approach presented in this chapter belongs to the third category. Compared to [49], our scheme is distributed and can be executed in run-time during an attack. Our algorithm can adapt to environmental changes because nodes with historically large PLR can be revisited when their PLR becomes small. In particular, our Q-learning approach bases its decision on the expected PLR and sum of its discounted histories instead of instantaneous values, thus the optimal decision is not easily subjected to the bursty nature of network statistics, and consequently exhibits algorithmic stability.

3.4 Problem Overview

In this section, we describe the basic problem that we are considering by introducing the network scenario and the adversary's characteristics. We consider a network that consists of nodes that communicate via wireless communications with each other. Depending on parameters, such as transmission power and modulation format, each node might only be able to observe communications from a limited amount of the full set of nodes. Our network, thus, will be arranged as a graph where the vertices correspond to wireless nodes, and the edges correspond to wireless links connecting nodes that are within radio range of each other. In our study, we have used two different motivating topologies: a regular grid deployment, and a deployment where the nodes are randomly placed. We illustrate these two topologies in Figure 4.6a.

Complementing the topological configuration of the network, it is necessary to employ an appropriate networking protocol that manages the routing of communication between nodes in the network. In our study, we have chosen to use the OLSR routing protocol [29] at the network layer, while we employed an 802.11 MAC with RTS/CTS turned off.

Our objective is to develop a network forensics tool that locates of a "jamming" node that is attacking the network. Specifically, our attacker is a congestion-style jammer that *continuously* broadcasts format-compliant packets, which results in the loss of legitimate traffic (packets from other nodes). In particular, for victim nodes (i.e.

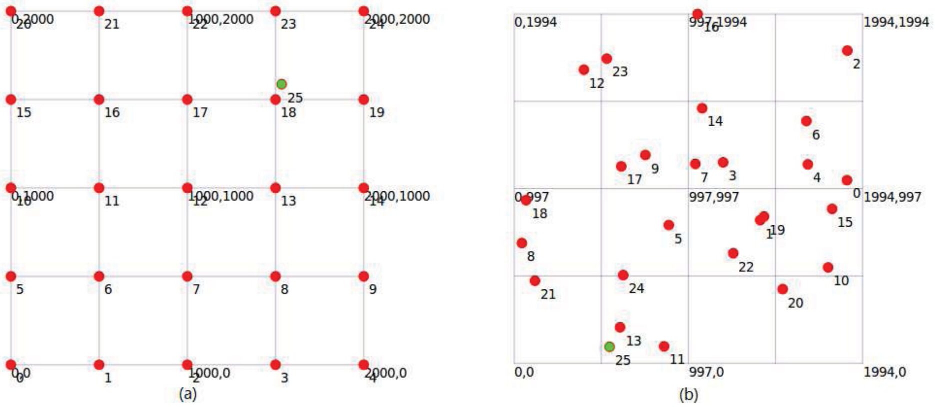


Figure 3.2: Grid and random topologies, red nodes are legitimate members. Green is the attacker, which broadcasts valid packets to surroundings.

nodes near the jammer), the attack has two immediate consequences: (1) the drop of legitimate packets at the physical and/or MAC layer of a legitimate node; (2) the increased proportion of total control messages to data packets in the network as the routing protocol must send out more control messages to maintain its connectivity. In this study, we assume that the malicious attacker is targeting a specific legitimate node, and hence that node will experience more degradation than other nodes. In this study, we will also consider two variations of this attacker: a naive jammer who simply injects blocking packets, and an intelligent jammer who has infiltrated the network and sends blocking packets as well as false information into the attack localization algorithm. For the intelligent jammer, the attacker is an insider to the network and can announce format-compliant messages that will be interpreted as if he is a legitimate member of the network.

Detecting and isolating the attacker will require forensics upon signals being created at each node that are associated with network traffic statistics. We have adopted the use of Packet Loss Rate (PLR), which captures the proportion of packets at a receiving node to total packets arriving at a receiving node. The calculation of PLR is based on all packets transmitted by the network, including control and data packets. PLR is a readily available network statistic, but we note that other network statistics, such as delay, may also be appropriate.

3.5 Attacker Localization in a Network via Q-learning

3.5.1 Q-learning Preliminaries

Our approach to localizing an adversary in a network uses a modified version of Q-learning. The standard discounted Q-learning was proposed by Watkins in [50], and an asynchronous and distributed version presented in [51]. In Q-learning, the agent learns an optimal policy from past experience by minimizing or maximizing the expected total discounted reward. The agent first randomly chooses an available action from its action set, then it obtains an immediate reward or penalty. This reward/penalty value will factor into calculating the new Q value for the current state and action pair. The optimal action for the current state is collected by finding the action which achieves the minimum/maximum of Q value among all the state-action pairs. We use the Q learning formulas given in [51]:

1. Calculating Q value (policy evaluation)

$$\widetilde{Q}_{t+1}(s,a) = (1-\alpha_{t+1})Q_t(s,a) + \alpha_{t+1}[R(s,a,s') + \gamma_{t+1}V_t(s')] \tag{3.1}$$

if $\widetilde{Q}_{t+1}(s,a) \leq V_t(s)$

$$Q_{t+1}(s,a) = \widetilde{Q}_{t+1}(s,a) \tag{3.2}$$

else

$$Q_{t+1}(s,a) = \alpha_{t+1}\widetilde{Q}_{t+1}(s,a) + (1-\alpha_{t+1})V_t(s) \tag{3.3}$$

2. Calculating V value (policy improvement)

$$V_{t+1}(s) = \min_a Q_{t+1}(s,a) \tag{3.4}$$

$$a_{optimal} = arg\min_a Q_{t+1}(s,a) \tag{3.5}$$

In our scheme, each wireless node will run its own Q-learning algorithm, and the Q-values at a particular node will correspond to the packet loss rates associated with the links between that node and each of its neighbors. The internal state within each node's Q-learning algorithm will be an assessment as to *which of its neighbors that node believes to be in the direction of the attacker*. With each node in the network running its own Q-learning algorithm, the collection of assessments can be thought of as a collection of fingers pointing in the direction of the adversary, and thus the final stage of the attacker localization algorithm is to collectively examine these assessments to infer where the adversary is located. Before proceeding to the specifics of our proposed Q-learning algorithm, we note that one of the advantages for using the Q-learning approach is that the Q-value within the algorithm is a discounted, expected value of the PLR, and hence naturally incorporates data smoothing to mitigate bursty PLR fluctuations that naturally arise in the operation of a network.

3.5.2 Q-learning Integrated with Routing Protocol

Our approach to identifying the jammer location integrates Q-learning with the underlying routing protocol. Routing protocols, such as OLSR, include unused message fields in broadcast control messages that may be utilized to convey additional information between nodes. Specifically, in order to support the dynamic operation of Q-learning at each node, we need the following information to be exchanged: a node's ID; the time slot for a Q value; the time slot for a V value; $V(self\_ID)$; PLR; and the optimal action (i.e. a decision as to which neighbor is in the direction of the jammer).

Using the distributed Q-learning framework of [51], we integrated Q-learning related messages into the routing protocol by attaching them to `hello` messages. These messages are propagated to all one hop neighbors. When its neighbor receives those messages, it detaches the Q-learning messages and stores them in a corresponding V table in reverse time order for the calculation of future Q values. For each time slot, the calculation function of Q learning is called, which searches the old Q value and V table to decide the optimal action (i.e. decision) for next time slot. Pseudocode for the procedure running on each node is as follows:

The receiving node stores the above information into V and PLR lists. The new Q value was calculated iteratively by equations (3.1), (3.2) and (3.3) until it converged

Algorithm 1 function Q-learning

```
Initialize $Q_0$, $V_0$
Get neighbors N from routing table
Calculate new $\gamma_t$, $\alpha_t$
/*policy evaluation*/
for all N in time slot t do
    search $Q_{t-1}(s, a)$ in $selfQNTable$
    repeat
        search available $V$ in $neighborVCTable$
    until find one
    calculate $Q_t(s, a)$
    $selfQNTable.push\_back(Q_t(s, a))$
    if $Q_t(s, a) < minQ$ then
        $minQ = Q_t(s, a)$
    end if
    recalculate cost (packet loss ratio)
end for
/*policy improvement*/
if every k time slots then
    $V_t = minQ$
    $a_{t,optimal} = argmin_a Q$
end if
```

in each node. Based on our simulation experience, this process takes $O(N)$ rounds, where N is the total number of nodes in the networks, and we are currently working on a proof for this conjecture. The asynchronous feature of the form of Q-learning we employ is well-suited for congestion cases as it allows the old neighbor V value in equation (3.1) to remain (though obsolete) in case there is a loss of information during transmission. Obsolete information can fade out with time due to the discount factor γ. As long as the attacked node can receive and send a little information out, other legitimate nodes can utilize this to find the node nearest to the jammer.

Algorithm 2 function Sendpacket

```
if $selfQNTable$ is not empty then
    send $\{selfV_t, PLR_t, ID, time\,stamp\}$
    send $a_{t,optimal}$
end if
```

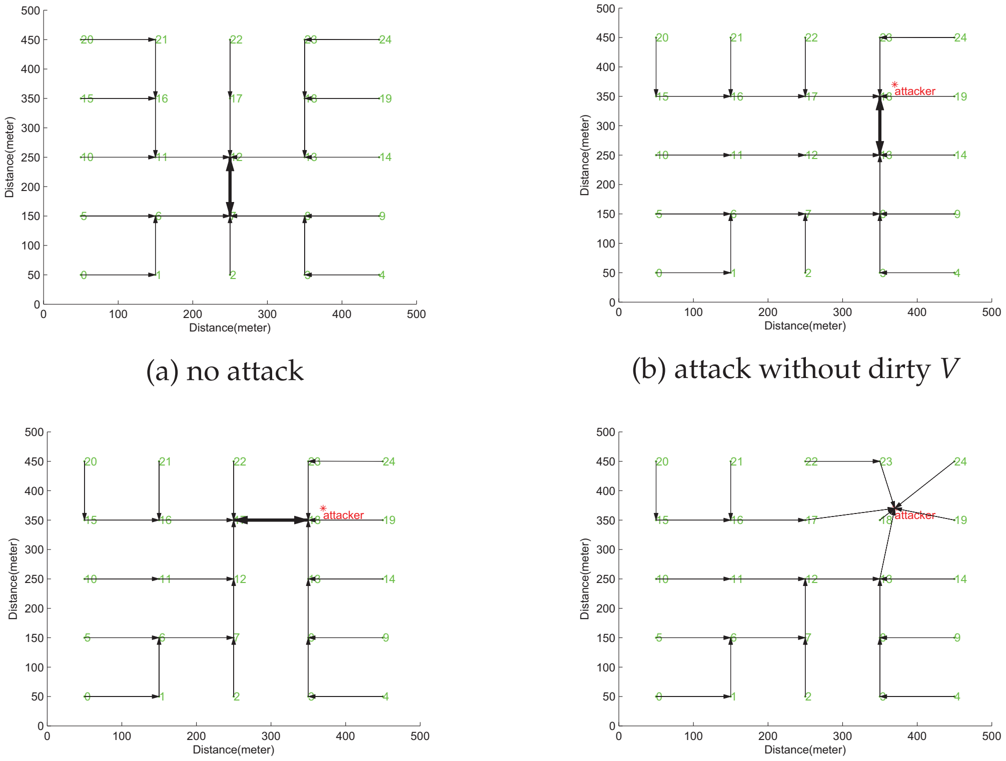
Algorithm 3 function Receivepacket

```
$neighborVCTable.push\_back(V_t, PLR_t)$
**if** neighbour's $a_{t,optimal}$ is self id **and** self's $a_{t,optimal}$ is neighbor id **then**
  **if** self's $V_t <$ neighbor's $V_t$ **then**
    mark self as the target node
  **end if**
**end if**
```

3.6 Simulation

We evaluated our algorithm using the NS3 simulator with two 25-node networks as presented in Fig. 4.6a. We employed the OLSR routing protocol with an 802.11 MAC where RTS/CTS was turned off in MAC layer. In our simulations, the nodes were static. The duration for each simulation was 100 time units. In all cases, the attacker begins to broadcast packets starting at time 20 and continuing until the end of the simulation. The attacker's traffic pattern was created to follow a Poisson distribution with inter-arrival time, 0.09 time units, and the length of the attacker's blocking packet was 95 bytes. In each time slot, the PLR was estimated and used to calculate the internal rewards in our Q-function, while the actual Q-values were updated every 2 time units and were estimated in a distributed fashion by each node. Every 6 time units, the V values were updated in a distributed manner. The learning rate was set to be less than 1 and decreased with time. We conducted simulations under different scenarios: (1) there was no attack; (2) there was a naive attacker; (3) there was an intelligent attacker that introduced large V and Q values; and (4) there was an intelligent attacker who introduced small V and Q values. These four scenarios were examined both in the grid and random topologies.

Fig. 3.3 provides the final optimal policies for each node in the grid network under the four scenarios mentioned above. For each node in the figure, there is an arrow that points to the neighbor that node *believes* is in the direction of the attacker. We see that in all cases our Q-learning jammer-locator algorithm was able to find the attacker, or the closest node to the attacker. One interesting phenomena that we observed is that the distribution of V-values exhibits increased variance when the network is under attack, which intuitively occurs because the attack disrupts the behavioral balance associated



(a) no attack

(b) attack without dirty V

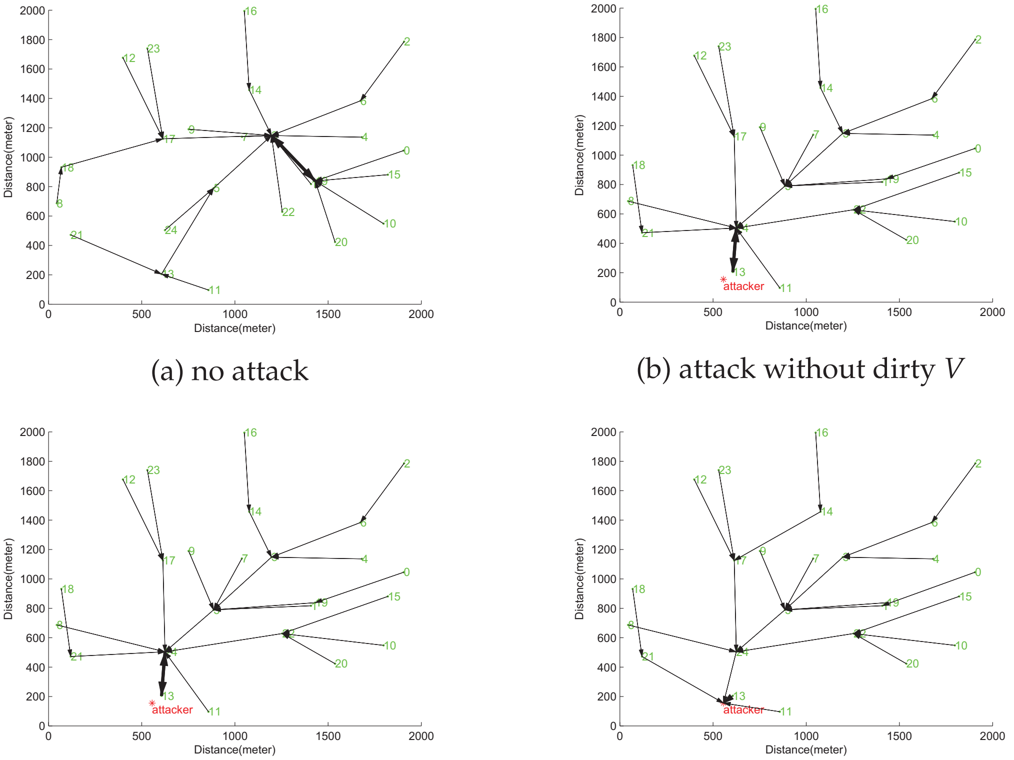
(c) attack with large dirty V

(d) attack with small dirty V

Figure 3.3: Jamming localization in the grid topology. Bold double-arrow link indicates network belief that the jammer is between the arrow endpoint nodes.

with nodes in the network. We also examined the case of the random topology. The results in Fig. 3.4 exhibit similar behavior to the case of the grid topology.

Lastly, we note, as with any such detection scheme, our approach will experience difficulty in differentiating between benign causes of congestion (such as the convergence of many high-traffic flows) and the congestion introduced by an intentional jammer. In particular, it is possible to construct high-traffic flow cases converging at a specific node that will lead to significant packet loss, in which case our algorithm identifies the network node closest to the worst region of benign congestion. Further, the ability of our scheme to identify and locate a jammer is tied to the inter-arrival time between the jammer's emitted packets, and a jammer can avoid detection by increasing its inter-arrival time. Nonetheless, in our experiments, we have witnessed that even so, our Q-learning approach locates the *region* around the jammer. One approach we are exploring integrating into our algorithm is a traffic-control method whereby nodes



(a) no attack

(b) attack without dirty V

(c) attack with large dirty V

(d) attack with small dirty V

Figure 3.4: Jamming localization in the random topology.

in the network adjust their outgoing traffic rate when the network suspects there is a jammer present. Such adjustment would reduce the likelihood of congestion being falsely declared as jamming by our forensics algorithms.

3.7 Conclusion

In this chapter, I examined the problem of locating the source of a jamming or congestion attack against a wireless network. I proposed integrating reinforcement learning (specifically, the popular Q-learning algorithm), with a network routing protocol to arrive at a distributed forensics algorithm that processes signals associated with network statistics to infer the location of a jamming event. The network signals I utilized were local estimates of packet loss rates as made by each node in a network. Our algorithm was validated through NS3 simulations under two different network topologies, and for both naive and intelligent attack scenarios. In particular, I found that our algorithm could reliably identify the location of an intelligent insider-attacker who both emits blocking packets and introduces false Q-learning information into the network. As part of our ongoing work, I intend to explore adaptively tuning our attack localization algorithm according to dynamic conditions within the network, such as node mobility (both the network and attacker's mobility) as well as under varying network traffic conditions.

In the next chapter, a cross-layer power control is proposed to defend against the jamming attack by heuristically adding connections in the peer to peer network to reduce the power consumption. The Fiedler value derived from the Laplacian matrix is utilized to rate the connectivity of network topology. In order to obtain the Laplacian matrix by each node, a light weighted approach is implemented by integrating with the OLSR routing messages, which can carry the information from two or more hops away, to transmit the real-time network information.

Chapter 4

Topology Adaptation for Robust Ad Hoc Networks

4.1 Overview of the chapter

Many cyber physical networks will involve ad hoc deployments utilizing peer-to-peer communications. Examples include transportation systems where a group of moving cars communicate in order to avoid collisions, teams of robotic agents that work together in support of disaster recovery, and sensor networks deployed for health-care monitoring, monitoring the operation of a factory plant or to coordinate and actuate mechanisms for energy conservation in a building. These networks may face a variety of threats that puncture their connectivity and, should their performance degrade, the result could be catastrophic. Consider, for example, a vehicular ad hoc network where communication assists collision avoidance. In such a case, degradation could lead to vehicle accidents. Therefore, in order to overcome network performance degradations and the puncture of a network (such as, blackhole or jamming) which is under attack, we propose an algorithm called the Fielder Value Power Adjustment Topology Adaption (FVPATA). FVPATA aims to dynamically adapt an ad hoc network's topology, even if the attacker varies its location and in the case of an interference-style attack by increasing the interference power. The algorithm utilizes the formulation from the graph theory which works with the Fiedler value to guide each node in wireless ad hoc network utilizing power adjustments to enhance the network's overall robustness. The advantage of the proposed mechanism, is that it is a light-weight approach which is totally distributed, based on topology updates inherent in the Optimized Link State Routing (OLSR) protocol and, hence, it is unnecessary to introduce additional messages. Additionally, an algorithm was developed to resolve problems involving asymmetric links that arise in ad hoc networks by eliminating unnecessary energy consumption of Fiedler nodes. Simulation results using NS3 show that the proposed mechanism successfully decreases the average amount of hops used by 50% and the delay of flows when nodes are migrating at a modest rate below 60 m/min.

4.2 Introduction

Many cyberphysical systems will be deployed using ad hoc wireless technologies, involving autonomous entities such as robots maneuvering in an environment. Such wireless networks can be easily subjected to a variety of attacks primarily because the transmission medium is an open one, allowing for observation and introduction of interference or false messages. These problems are particularly pernicious in the case of ad hoc networks where nodes in the network communicate with each other in a dynamic and opportunistic manner. In ad hoc cyberphysical networks, a malicious attacker can simply employ interferences to cause legitimate nodes around him unable to communicate with the neighboring nodes. Alternatively, he could also introduce attacks that would puncture the network by dropping packets or locally disrupting the routing procedure. In either case, a region of the network becomes unusable and the performance of the network significantly degrades around the areas near the attack.

There are many other complementary tools that can be used to cope with such attacks directed against ad hoc networks. These countermeasures are grouped into four main categories: (1) carefully design routing protocol to re-route packets around the attack area [52–55] and those attack areas can be discovered by machine learning methods [56–58]; (2)Implement multi-path plus tunneling to add redundancy to the current route [59–63]; (3) adjust the location of network nodes [40,64,65]; and (4) apply robust and redundant coding [66–70].

In this chapter, we examine a complementary approach to coping with jamming attacks in a distributed fashion in an ad hoc network. Our approach aims to aid the participants in an ad hoc network to avoid holes punctured in the network connectivity by an attacker through network control algorithms, and to strengthen the reliability of communication should the attacker shut down one or more of the legitimate nodes. In order to accomplish this, we propose an algorithm that aims to control the network topology so it can minimize network degradation in the instances of an attack. Our proposed FVPATA algorithm is integrated with the popular OLSR routing protocol for wireless ad hoc networks and it uses the concept of “Algebraic Connectivity” of a network’s topology, as characterized by the Fiedler value [71–73], to identify connectivity-sensitive nodes in the ad hoc network. These nodes then adjust their transmission power to enhance the network’s robustness. The advantages of FVPATA are: (1) it only requires an adjustment to the power employed by a small set of carefully chosen Fiedler nodes. Thus, this method conserves energy for the whole network rather than increasing the power for all nodes. Additionally, increasing the power of every node can introduce undesirable interference, which often results in a decrease of the network throughput; (2) it is a distributed algorithm involving only local actions to affect the entire operation of the network. Each node uses a unique network topology shared by hello and TC messages of the OLSR protocol, and it tailors the topology through its own local actions.

This chapter is organized as follows: In Section II, we provide the background and mathematical foundation for our algorithm. In Section III, we describe the mechanism of our Fiedler value power adjustment algorithm in details and demonstrate how the algorithm can be integrated with the OLSR protocol through pseudo codes. Indeed, our approach can universally be integrated with any state-sharing ad hoc network routing algorithms [2–5]. In Section IV, we analyze the performance of our FVPATA through simulations involving scenarios of different attacks, followed by our conclusion in Section V.

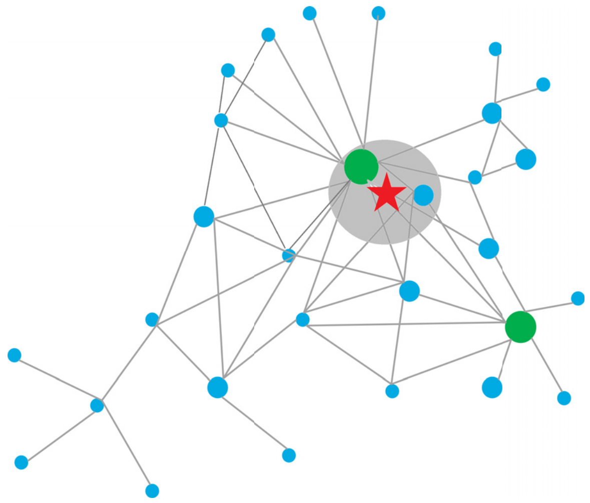


Figure 4.1: A possible scenario of attacks in an ad hoc network. The red star represents the attacker placed in an area where many routes and flows must transit through. The shaded region indicates a region that is "punctured" by the attacker to effectively isolate many nodes which causes serious structural damage to the network's topology.

4.3 Background and theoretical foundation

4.3.1 Attack model

As a starting point for our discussion, we shall consider a very simple attack model where an attacker is positioned near the center of a network since nodes in the central area are surrounded by densely populated neighboring nodes and could potentially become a bottleneck in traffic flows. As an example, we illustrate a network with an attack in Figure 4.1. In this figure, a single attacker is located near an area in which many routes intersect, and the attacker can potentially cause serious structural damage to the network's topology by attacking one or more nodes nearby. Our approach works well with multiple attackers as our approach is distributed and the power adjustment is done according to local views of topology. However, for the sake of our discussion, we consider the case of only one MAC-layer attacker causing harm to several neighboring nodes simultaneously.

An attacker's goal is to shut down the maximum number of nodes with a minimum amount of effort. Therefore, the attacker's objective is to place himself in an area with heavy network traffic. In our example, the MAC-layer attacker continuously injects format-compliant packets to legitimate nodes without time gaps in between the packets. As a result, nodes under attack become unable to communicate properly (e.g. access the channel and complete packet transmission and reception successfully) and essentially become shut-out from the network's operation. Throughout our discussion, we will refer to an ad hoc network that uses the OLSR protocol [29]. The reason why OLSR is used is because it can support our algorithm easily since: (1) its TC message can deliver link connectivity status from three hops away and it can assist nodes in gaining complete knowledge of the network connectivity; (2) it reduces the need for extra messages when updating topology information; and (3) it is amenable for executing a distributed algorithm on each node in any ad hoc network. We note that our approach can apply equally well to other routing algorithms which have similar state-sharing features [74]. We can integrate our algorithm with them in two ways:1) utilize the periodic hello messages to carry the extra topology information which is from three or more hops away, such as the hello messages in Dynamic Source Routing (DSR) [2] and Ad Hoc On-Demand Distance Vector (AODV) [3]; 2) adopt the self-contained topology update mechanism in routing protocols, for example, Global State Routing (GSR) [4] consults the vectors of link states exchanged with routing information to obtain the global knowledge of the network topology, and the On-Demand Packet Forwarding Scheme (ODPFS) [5] constructs a virtual backbone among nodes. During the construction, the global topology information is propagated.

4.3.2 Motivating foundation

The main purpose of FVPATA is to increase the network's robustness while minimizing the energy needed to confront an attack. The network's robustness/connectivity is closely related to the node's degree in a network graph [75]. The average node degree in a random network (when being deployed according to a spatial homogeneous Poisson process) increases with the node's radio range, as illustrated in Figure 4.2. Moreover, Figure 4.2 also indicates that the spread or variation of the average degree first rises and then decreases at a larger radio range. Figure 4.3 implies that increasing the

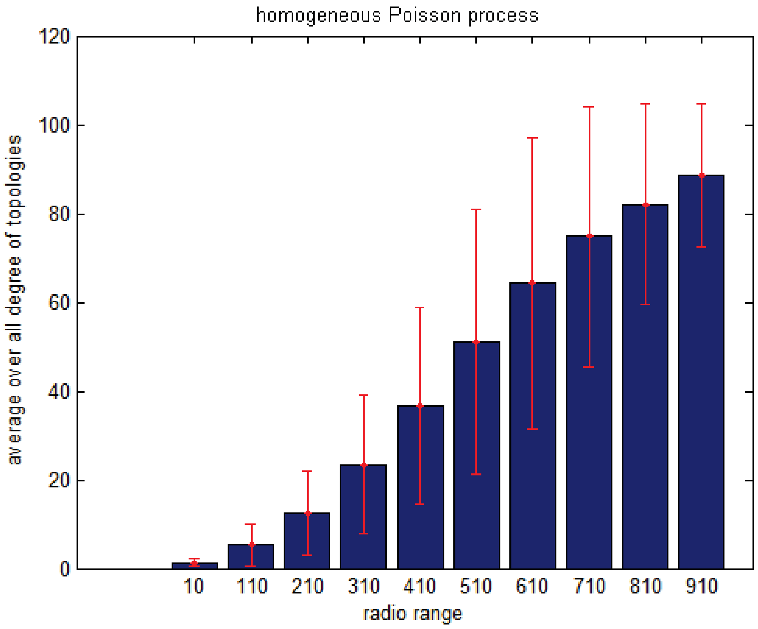


Figure 4.2: The variation of node degree as function of radio range

radio range can cause the number of mutually reachable source and destination pairs to grow, which directly corresponds to the graph's connectivity. However, after reaching a certain threshold no obvious growth occurs. This characteristic illustrates that it is not necessary to increase radio range indefinitely in order to strengthen the network connectivity. On the other hand, increasing transmission power without restraint can inadvertently generate unnecessary radio interference in the ad hoc network.

4.3.3 Fiedler value and graph-theoretic connectivity

Now we briefly introduce the graph theory and lemmas that will be applied in our algorithm, followed by a description on the working mechanism behind FVPATA.

Network connectivity can be depicted by its algebraic connectivity, also called the Fiedler value. It is the second smallest eigenvalue of the Laplacian matrix, $L(V, E)$, of a network's topological graph, $G(V, E)$ where V is a vertex set and E is the edge set connecting two vertices in graph. The Fiedler value is always non-negative, and its amplitude is proportional to the graph connectivity. It is zero if and only if the graph is disconnected. The number of zero eigenvalues in the eigenvalue set of L equals to the number of connected components in a graph. According to [71], the Fiedler

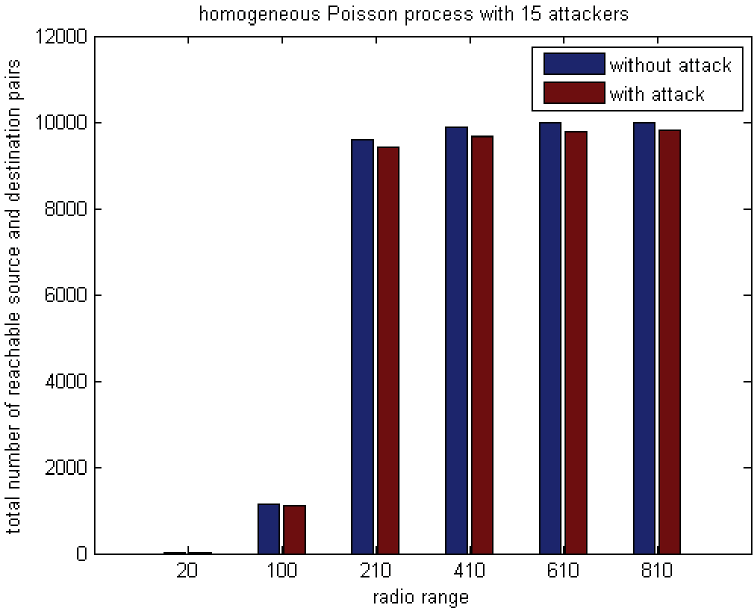


Figure 4.3: The variation of network connectivity as function of radio range

value represented by λ_1, of a graph, G, can be obtained by the following eigenvalue optimization problem.

$$\begin{aligned} &\lambda_1 = \min y^T L(V,E) y \\ &st.\ y^T y = 1 \ \ and \ \ y^T \mathbf{1} = 0 \end{aligned} \tag{4.1}$$

where y is a vector which does not equal to $\mathbf{1}$.

The Laplacian matrix of a given graph is defined as follows: Given a graph $G(V,E)$ without self cycles and multiple links between two nodes, the Laplacian matrix L is calculated by

$$L(V,E) = D(V,E) - A(V,E) \tag{4.2}$$

where $D(V,E)$ is a diagonal matrix whose diagonal entry contains the degrees for each node. $A(V,E)$ is the adjacency matrix with each entry being a value of zero or one when nodes are connected to each other. In addition, its diagonal is zero since $G(V,E)$ has no self cycles. According to equation (6.5), the Laplacian matrix has the following properties [20,76]:

- *Lemma 1*: L is symmetric matrix. Its (i,j)th and (j,i)th entries are same and its diagonal entries contain each node's total degree.

- *Lemma 2*: All its eigenvalues are real since L is symmetric.
- *Lemma 3*: L is a positive semi-definite matrix. Thus, it has no negative eigenvalues. Its first smallest eigenvalue is always 0 since the sum of each row or column is zeros. By sorting the eigenvalues, we obtain: $\lambda_0 = 0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{n-1}$
- *Lemma 4*: The number of zeros in eigenvalue indicates the number of disconnected components in the graph. If the graph is strongly connected, then the second smallest eigenvalueλ_1, which is also the Fiedler value is always larger than zero.
- *Lemma 5*: If the attacker kills the links in between nodes or when the network's links are broken because of natural distances, the Fiedler value $\lambda_1(V, E_1) \leq \lambda_1(V, E)$ where $E_1 \subseteq E$
- *Lemma 6*: Fiedler value's upper bound is limited to the minimum degree of nodes and the total number of nodes exist in networks. The upper bound approaches to the minimum value of degrees in nodes when the network is large. The exact relationship between them is given by [20]

$$\lambda_1(V, E) \leq \frac{|V|}{|V| - 1} \min_v d_v \tag{4.3}$$

Lemma 5 informs us that the Fiedler value can become larger when adding edges to a graph. Thus, a network becomes more robust as the Fiedler value increases, which implies stronger connectivity.

Our objective is to identify the weakest point in the network connection and heuristically improve the network by increasing the degree or number of neighbors associated with that node. Particularly, we are interested in what happens when we remove a node from a network's graph, and hence we will introduce a modified notion of the Fiedler value, which corresponds to the impact associated with removing all of a node's links (i.e. connections to other nodes in the network). Specifically, we define a node's Fiedler value as:

Definition: For a graph $G(V, E)$, the node Fiedler value associated with node j corresponds to the Fiedler value $\lambda_1(V, E_1)$ where E_1 corresponds to a revised set of edges for G where all edges containing node j have been removed from E.

With this definition in mind, we can re-examine the connectivity of the topologies that remain on a case-by-case basis after removing each node, and discover the nodes in the network whose deletion would have the most harmful impact on the network's algebraic connectivity. We propose a heuristic for improving the network's condition whereby we attach more links to the nodes with low nodal Fiedler value.

Lemma 6 informs us that increasing the degrees for all nodes is ineffective because the upper bound of Fiedler value is constrained by the minimum value of the degree of the nodes. Conceptually, we only need to select a few nodes to add links to, and this will be reflected in the FVPATA algorithm by having each node examine whether it is in the set of m nodes with the lowest Fiedler value.

4.4 OLSR-based topology adaptation algorithms

In this section, we use the Fiedler value's properties previously described to guide an online cross-layer power adjustment scheme that enhances the network's robustness when facing an attack. The idea is to select a node that is least-suitably connected to the network. By increasing the power of transmission on those nodes, we can strengthen network's capability by being able to reach more nodes and thus improve the network's overall connectivity.

4.4.1 Choosing the node

As the amplitude of the Fiedler value represents the network's connectivity, we choose several nodes in accordance with their Fiedler values after determining their associated adjacency matrices with those nodes removed. Removing a node from the adjacency matrix means deletion of the corresponding i-th row and column.

Each node first builds an adjacency matrix for the topology it obtained in an online manner from OLSR's TC and hello messages, which are sent periodically by OLSR protocol. In the following section, the procedure for obtaining topology from the OLSR protocol will be discussed. This topology is updated every T time-units where T is a free parameter that can be adjusted. Upon obtaining the adjacency matrix, a node calculates a list of Fiedler Values from the remaining adjacency matrices through removing each node. A node with the smallest Fiedler value indicates that deletion of that particular node will cause maximum damage to the network.

Moreover, nodes with the least number of links often correspond to being located in a less densely populated area, or in an area without many surrounding neighbors. They could also be located in an environment where the condition of the local channel is poor with large levels of local noises making them likely to be Fiedler nodes. In these situations, increasing the nodes' power might be inefficient due to a significant amount of energy being needed to reach other nodes or to overcome the channel conditions. However, to give them the opportunity to connect to a larger network, our algorithm is iterative in the sense that it continuously chooses the weakest nodes from the resulting network topology. Each node in the minimum Fiedler value set will choose to increase its power with probability p. We choose p according to the binomial distribution, in terms of n_1, where n_1 is the number of nodes with the least number of neighbors and N is the network size. Hence, p equals to $1 - \left(1 - \frac{1}{N}\right)^{k(N-n_1)}$ and k is a parameter that manages the tradeoff between adding power and redundancy while k can vary for each node.

Upon obtaining the self-evaluated Fiedler node id, a node ascertains whether this node id is identical to its own. If so, it starts to increase transmission power until reaching the degree or power limit. Otherwise, it recalculates the Fiedler value using the remaining adjacency matrices. The remaining adjacency matrices are obtained by deleting the row and the column of each corresponding Fiedler node id that was obtained from the previous round. This process iterates until reaching a maximum number of iterations, or a node becoming a Fiedler node, whichever happens first.

4.4.2 Getting the updated topology

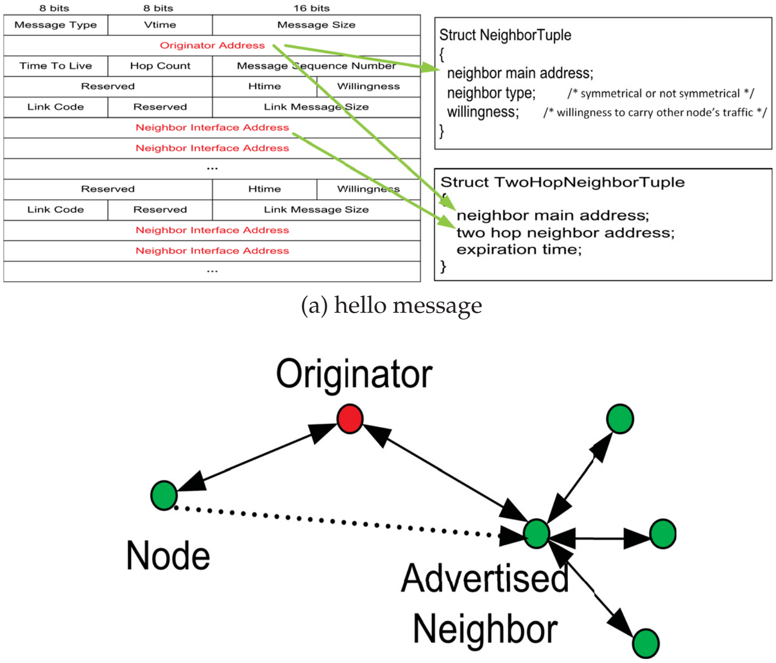
The adjacency matrix for the network's topology is critical when calculating the Fiedler value. We obtain it from the TC and hello messages of the OLSR protocol.

Hello messages and obtaining one and two-hop neighbors

Hello messages in OLSR protocol provide both one and two hops of neighboring link status information. Messages not received by directly connected neighbors are discarded. Figure4.4a shows the hello message format and contains the link status information from the network topology. The link code in the hello message identifies both the link and the neighbor type between the originator and its following list of neighbor interfaces. When receiving hello messages, the originator's main address is stored into the neighbor's main address in the neighbor tuple as shown in Figure 4.4a. The originator is the node's one hop neighbor if the main address of "Neighbor Interface Address" field is the address of the node itself. The rest of the main address of the "Neighbor Interface Address", whose neighbor type is symmetric specified by the link code, corresponds to the node's two-hop neighbors that are intermediately connected by the originator. This two-hop neighbor's main address is stored in the two-hop neighbor's tuple, shown in Figure 4.4a. Figure4.4b illustrates the mapping from the protocol field to the network topology.

TC messages for obtaining more than two-hop neighbor link status

Since a node only receives one hop neighbors' hello messages, we are unable to obtain link statuses on nodes that are more than two hops away from hello messages. Therefore, we had to adopt TC messages to solve this issue. When receiving TC messages, a node first verifies whether the sender of TC messages is from the set of trusted one-hop neighbors. If not, it discards the message. If yes, it updates the TC tuple shown in Figure 4.5 if and only if the Advertised Neighbor's Sequence Number(ANSN) is larger than the previous one stored. Else, it adds a new TC tuple if there is no record found. The "originator" field in the OLSR message is then copied to the main address and the



8 bits	8 bits	16 bits	
Message Type	Vtime	Message Size	
Originator Address			
Time To Live	Hop Count	Message Sequence Number	
Reserved		Htime	Willingness
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			
...			
Reserved		Htime	Willingness
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			
...			

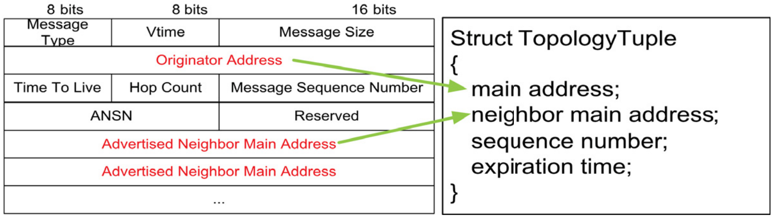
```
Struct NeighborTuple
{
 neighbor main address;
 neighbor type;      /* symmetrical or not symmetrical */
 willingness;      /* willingness to carry other node's traffic */
}
```

```
Struct TwoHopNeighborTuple
{
  neighbor main address;
  two hop neighbor address;
  expiration time;
}
```

(a) hello message

(b) two hop neighbors

Figure 4.4: Hello message and data structure storing one or two-hop neighbors



8 bits	8 bits	16 bits
Message Type	Vtime	Message Size
Originator Address		
Time To Live	Hop Count	Message Sequence Number
ANSN		Reserved
Advertised Neighbor Main Address		
Advertised Neighbor Main Address		
...		

```
Struct TopologyTuple
{
  main address;
  neighbor main address;
  sequence number;
  expiration time;
}
```

Figure 4.5: TC message and data structure of storing neighbors

Advertised Neighbor's Main Address is also copied to the neighbor's main address shown in Figure 4.5.

Integrating Fiedler value into the OLSR protocol

Therefore, the OLSR protocol gives the opportunity for a node to construct self-evaluated adjacency matrices to support the ability for each node to calculate Fiedler values that represent the network connectivity. Whenever a hello or TC message is received, the OLSR protocol processes the messages and stores the link connections in each corresponding tuples. At each time interval, a node re-computes the adjacency matrices according to their neighboring table. If nodes have symmetric links between them,

Algorithm 4 : increase the transmission power of Fiedler nodes until each of them reaches the degree limit

```
Initialization  degree_limit = $D$ and power_limit = $P$
Every time slot:
Get $adjacency\_matrix$ from the OLSR protocol by hello and TC messages
Get $n_1$ from $adjacency\_matrix$ and set $explore$ to be zero
while $\left[1 - \left(1 - \frac{1}{N}\right)^{explore \cdot (N - n_1)}\right] < threshold$ do
  for i from 0 to $num\_nodes - 1$ do
    remove node i from $adjacency\_matrix$
    calculate the second smallest eigenvalue
    fiedler_list.push_back(the second smallest eigenvalue)
  end for
  /* get the connectivity-weakest node */
  Sort(fielder_list)
  /* index 0 refers to the smallest fielder value */
  $node\_to\_adjust$ = fielder_list(0);
  if $node\_to\_adjust$ = self id then
    if self degree$<$ $D$ and self power $< P$ then
      increase power of $node\_to\_adjust$
      break
    end if
    $adjacency\_matrix.remove(node\_to\_adjust)$
    $explore = explore + 1$
  end if
end while
```

then the corresponding entry of the adjacency matrice is set to be one, otherwise, it is zero. The diagonals of the adjacency matrice are set to be as zeros since a node has no self cycle. However, a node is able to set the exploration time for itself. One exploration corresponds to removing a connectivity-weakest node from the adjacency matrix. This exploration process iterates until the connectivity-weakest node becomes the node itself or when the total exploration time is reached. As soon as a node realizes it is the connectivity-weakest node, it begins to increase its transmission power if its degree is below the total degree limit and its transmission power is lower than the total power limit. The node stops increasing its transmission power within a certain time. This process is online and distributed and the pseudo code is given in Algorithm 1.

4.4.3 Symmetric and asymmetric links

In FVPATA, we only consider symmetric links to meet the requirements of the OLSR routing table so that we can guarantee packets are successfully delivered. However, asymmetric links may occur. For example, one Fiedler node may raise its transmission power in order to get connect with another node. However, that node might remain at the same transmission power if that node's total degree limit has already been reached or that node is not a Fiedler node. In this case, the Fiedler node continues to increase its transmission power and wastes the energy oblivious to the other node having no willingness/incentive to cooperate. For instance, in Figure4.4b, the advertised neighbor already has four neighbors. If the degree limit is three, it has no incentive to increase its power to connect with the node. Thus, an asymmetric link may exist between a Fiedler node and a non-Fiedler node, as shown in the dotted line in Figure4.4b. Further, the Fiedler node will iteratively increase its power until its total power limit is reached.

To solve the problem caused by asymmetric links, we require each non-Fiedler node to verify whether the Fiedler node has enough neighbors before increasing its power. After n time intervals, a non-Fiedler node verifies whether the Fiedler node is its an asymmetric neighbor. If yes, then it examines whether the Fiedler node is a two-hop neighbor of itself and whether the number of Fiedler node's neighbors remains under its degree limit. If the answer to both conditions is yes, this non-Fiedler node will increase its power if the power limit for itself has not been reached. This process iterates until the Fiedler node becomes its symmetric neighbors or when the number of Fiedler node's neighbors have reached the degree limit, whichever comes first. The link code in hello message can indicate whether the link is asymmetric or symmetric with the symmetric links being its neighbors. Algorithm 2 gives the pseudo code when dealing with asymmetric issues.

Our power adjustment algorithm has weaknesses and does not solve all the problems. Firstly, since the beacons in the OLSR protocol may update the topology too fast,

Algorithm 5 : solve asymmetric links of Fiedler node

```
After $n$ time slots:
if node is not Fiedler node but the obtained Fiedler node is its asymmetric neighbor then
  if Fiedler node is node's two hop neighbors then
    calculate the number of Fiedler node's neighbors from self-evaluated adjacency_matrix
    if Fiedler node degree $< D$ and self power $< P$ then
      increase non-Fiedler node power
    end if
  end if
end if
```

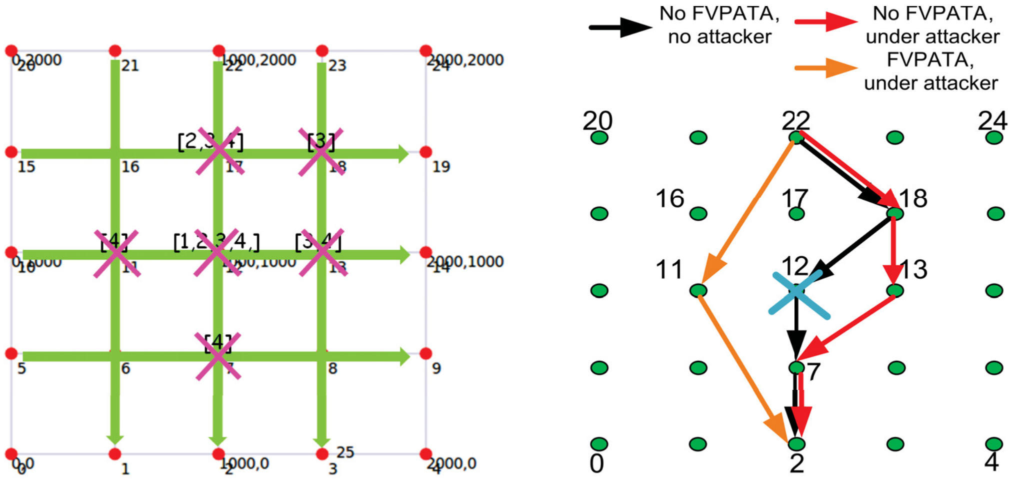
it may cause an overshoot of some nodes' degrees and may result in larger interference due to too much transmission power. Secondly, the algorithm does not totally solve the asymmetric link problem although we designed the algorithm to allow non-Fiedler nodes to respond to the connection request of Fiedler nodes. However, for a node which is in a low density area, even if some nodes reply to its request, the final degree still cannot meet the degree requirement and that node continues to increase transmission power until exhausting all the energy.

4.5 Simulation

4.5.1 Simulation results

Simulations were performed using the NS3 network simulator. The topology used in the simulations is shown in Figure 4.6a. There are a total of 25 nodes positioned in a grid with an interval of 500 meters in between nodes both horizontally, and vertically. Exploring the case of a grid topology gives us a clear view on the operation of our schemes. In the study, we introduced a total of six flows running through the networks simultaneously, shown as green lines in the Figure4.6a. As we can see, those flows are close to each other and this creates interferences among themselves.

Beyond the existence of interferences among flows, we introduced an attacker who can simultaneously shut down several nodes near its location. We also assumed the attacker's power would grow gradually so that he can affect one node to five nodes. We



(a) topology (b) true path

Figure 4.6: Topology, flow pattern, attacker position and true paths. a) numbers in parentheses refers to different jamming locations. "1" denotes a MAC-layer attacker is at node 12. "2" denotes a MAC-layer attacker is between node 12 and 17. "3" depicts a MAC-layer attacker located at the intersection of diagonal lines of node 12, 13, 17 and 18. "4" refers to a MAC-layer attacker located at node 12 with a radio range of 500 meters; b) changes of true paths with one node failure with or without FVPATA

examined the effectiveness of FVPATA in terms of average hop, delays, and throughputs under different attacking scenarios. The results show that FVPATA has provided a significant improvement in performance when the nodes are under these different kinds of attacks.

We placed the attacker at four different locations, corresponding to four scenarios where they are all approximately centered around the populated area as shown in Figure4.6a. The purpose behind this attack was to simulate the attacker's attempts to reach out to as many flows as possible. In the first scenario, we assumed the attacker had a relatively low power level based on its radius of impact; hence, he could only attack one node in the network. Therefore, we chose Node 12, which is the center of the whole topology. For the second attacking scenario, the MAC-layer attacker raised its transmission power with an effective attacking range of 250 meters. We placed the attacker in between Node 17 and Node 12 to block the source transmission as much as possible. For the third scenario, we increased the attacker's radio range to be $\sqrt{2} \cdot 250$ meters and position it at the intersection of two diagonal lines of a rectangle composed by Node 12,17,13 and 18. In the last scenario, we set the attacker's radio range to 500 meters and put it in the center of the flows again. Under this condition, the attacker could affect Node 12, 11, 17, 13 and 7. Under all four scenarios, the attacker's interference range increases so as the attacker's ability to affect other nodes.

The simulation runs contained 1000 time-units and our application data started at 100 time-units after the routing tables were established and continuously fed into the network without gaps in time. The attacker started the attack at 100 time units while the OLSR protocol was running continuously to maintain the routing information. Hello messages for the OLSR protocol were sent every 2 time units and TC messages were sent at every 5 time-units when RTS/CTS was turned off. FVPATA started when it detects a 80.5775% packet loss ratio and explored at most five times in each time interval.

Figure 4.6b shows the changes of the route from source Node 22 to destination Node 2 after suffering from an attack with the application of FVPATA. Here we consider scenario 1 (where only Node 12 is the node under attack) as the simplest case for an easier understanding on the concept. In Figure 4.6b the black route is the general case under the absence of an attacker and FVPATA. The route walks through Node 12, travels through Node 18, and finally reaches Node 2 which is used as a baseline for comparison. The red line is a route depicting the aftermath of shutting down Node 12 by an attacker without application of FVPATA. As the figure suggests, the red route skips Node 12 and the OLSR protocol finds an alternative route running through Node 13. Although the OLSR protocol has self-recovery capabilities, the disabled Node 12 causes traffic congestion around the affected nodes which resulted in more network-layer interference. This condition becomes even worse when more than one node breaks down. Therefore, we cannot solely rely on the self-recovery mechanism of the OLSR protocol. The brown route represents the condition when FVPATA is applied after nodes detected an abnormal packet loss. The figure shows that Node 2 increased its transmission power since it is the Fiedler node. Moreover, to obtain a degree of 6, a non Fiedler node (Node 11) also increases its transmission power and as a result, Node 2 could directly reach Node 11. Thus, the final route contained only two hops

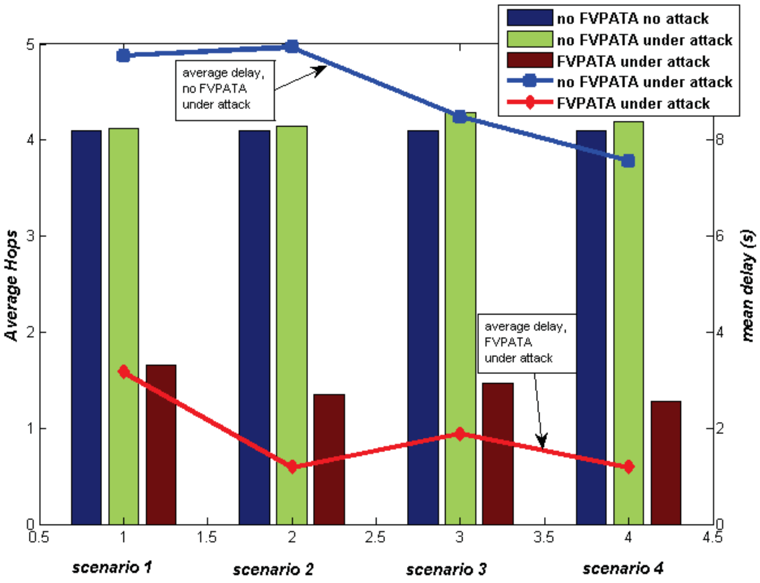


Figure 4.7: Average hops and mean delay of all flows

and it reduced the switching time spent by Node 6 and 16. FVPATA has effectively diminished the total transmission time by cutting down the number of hops needed for the flows. Figure 4.7 also demonstrates that FVPATA actually lowers the delay of the whole path.

While examining Figure 4.7, it is clear that it illustrates the decrease in the average number of hops and the mean delays among six total flows after the application of FVPATA while the network is under attack. The improvement is apparent and despite more and more nodes being shut down (Node 7, 11, 12, 13 and 17) in Scenario 4, its average number of hops and mean delay are the lowest. The possible explanation for this is that nodes around the attack areas begin to have reduced connections to the networks. Therefore, with large probability, removing them can cause network to be disconnected. For example, in our simulation topology shown in Figure 4.6a, removing Node 2 and 5 can cause a separation of networks into two parts. FVPATA chooses those nodes near the border and increases their transmission power and weaves a connection between them. Therefore, if more and more nodes are disconnected, nodes near them (may also be two hops away) are more likely to be selected by FVPATA. Based on this and FVPATA's distributed structure, FVPATA should work well in situations where multiple attackers exit. In the simulations, we actually saw fewer numbers

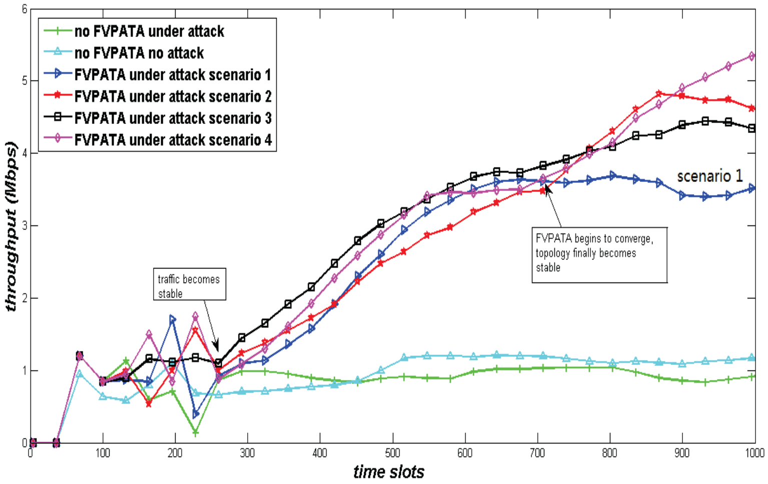


Figure 4.8: Throughput in each time slot for Scenario 1

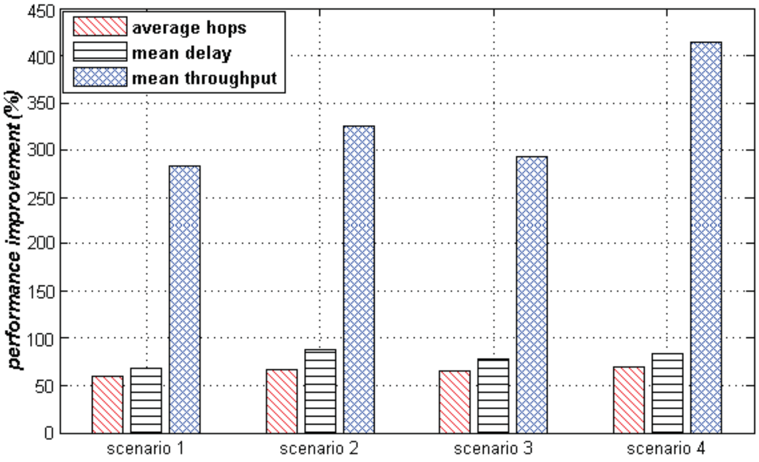


Figure 4.9: Performance improvement in terms of average hop, delay and throughput

of hops when more flow interference exists.

Figure 4.8 represents the throughput we collected at each time interval for Scenario 1. We applied a sliding window with a width of 60 time-units to smooth data across time intervals since the network data collected experienced random variations. Simulation results indicated FVPATA increased the network throughputs compared to non-FVPATA employed network. Moreover, FVPATA converged after approximately 700 time-units and stabilized thereafter. However, the convergence time depends on the number of Fiedler nodes participating in the power adaptation. A much more disconnected network requires a longer stabilization time.

We also computed the improvement on performance associated with the application of FVPATA under attack in Figure 4.9. Comparing to the cases without power adjustments, FVPATA reduces the number average hops by at least 50% and cuts the mean delay by at least 60%. To calculate mean throughputs, we collected data from 700 and 1000 time-units during the periods where the FVPATA algorithm converges and the network throughputs stabilize. The calculated mean of the associated sample data is then compared with the case without power adjustments. We discovered there is an improvement by more than 2.5 times in terms of throughputs.

4.5.2 Simulation results involving mobility

Besides static networks, we also considered FVPATA's performance when facing the mobility of nodes, as might occur in a cyberphysical application involving robotic agents. Each node traveled within a square area of 2000 meters by 2000 meters, randomly changing direction every two seconds. We set each node as having the same rate and we increased their rates at succeeding rounds of simulation. In the simulations, we assumed a MAC-layer attacker tracked a node as a target and never changed its target throughout the course of attacks. The MAC-layer attacker could only track one node at a time since legitimate nodes moved constantly, and randomly, from one place to another. Therefore, it resembles the effects in Scenario 1 we studied earlier, but in a dynamic sense. We collected the average numbers of hops, mean delays, and throughputs data under moving rates of 1m/s, 10m/s, 20m/s and 30m/s. Each simulation ran for 1000 time-units, which is the same as static network.

Figure 4.10 shows that FVPATA decreases the average number of hops and mean delays when nodes are moving at moderate speeds. It also implies that sometimes increasing speed can help latency. A possible explanation for this, is that some pairs of source and destination nodes could approach each other, closer and closer, to shorten the distance and time during the packet transmission. Figure 4.11 presents the percentage improvement on the performance of FVPATA in average number of hops, mean delays and throughputs. We can see that the improvements of mean delay and

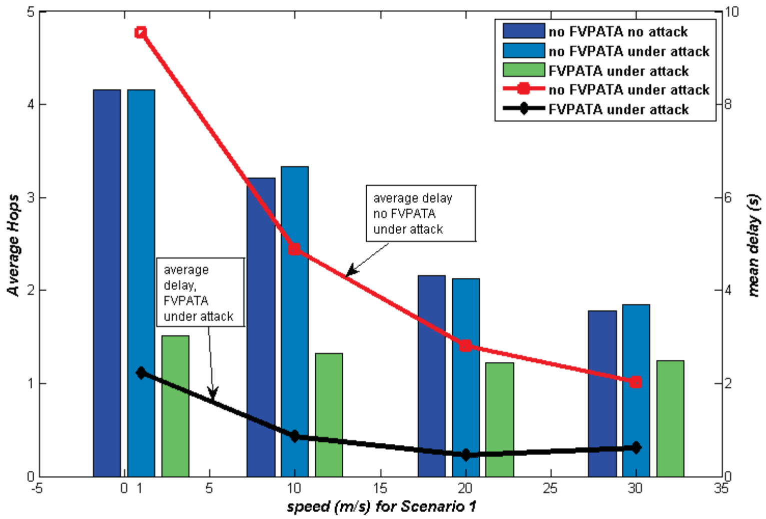


Figure 4.10: FVPATA decreases the average number of hops and mean delays under moving nodes

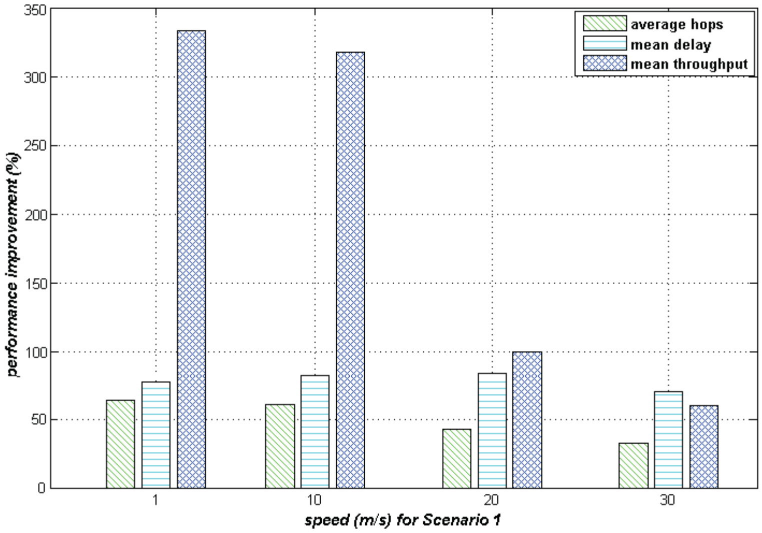


Figure 4.11: FVPATA improves their average number of hops, mean delays and throughputs

throughputs are above 50% when the moving speed is no more than 30m/s. The percentage improvement on average hops and throughputs decreases when the speed of nodes increase, especially the throughputs, it drops rapidly under a relative high speed. To put the speed of 30m/s into perspective, it is equivalent to a car traveling at speed of 67 mph on the road. Therefore, the conceptual application of FVPATA in daily life may be viewed as rather practical.

4.5.3 Parameters that affect performance

The magnitude of FVPATA's performance improvement depends on many factors: (1) the beacons in OLSR which affect the converging and stabilizing time of FVPATA; (2) the size of the steps in power increase. Smaller steps lower the converging time while bigger steps often overshoot the node degree, and waste energy; (3) the limits on total number of degrees. A larger number of degrees results in a higher throughput with more energy consumption; (4) the position of the attacker and the pattern of network flows. Figure 4.12 considers the position of attackers simulated by matlab and the results indicate that the attacker's position can also effect network connectivity. Network performance deteriorates with the number of affected nodes if the affected nodes are selected by the indication of their Fiedler values. However, FVPATA works well under the condition where multiple affected nodes since the nodes near broken nodes have more opportunity to be selected as Fiedler nodes and thus, increase their power; (5) RTS/CTS, this is a solution to the problems of hidden terminal and the reduction of flow interference. Since it affects the traffic pattern, it also influences the network throughputs; (6) the depth of exploring process in Algorithm 1. A more in-depth exploration of network connectivity indicates a lesser number of hops in a route, which translates to a much stronger and more robust network. On the other hand, this also causes an overshoot in the node power.

4.6 Conclusion

In order to achieve topology adaptation for the resilient communication operation of cyberphysical networks deployed using ad hoc networking technologies, we proposed an algorithm called FVPATA to overcome attacks that puncture a hole in the ad hoc network. The objective of FVPATA is to use the Fiedler value, which approximates the robustness of a graph, to guide the nodes in the cyberphysical network to discover weak vertices in the underlying network topology. Each node verifies and promotes itself to be a frail node if it is the Fiedler node. Fiedler nodes increase their transmission power after finding they are weak nodes. FVPATA is totally distributed since each

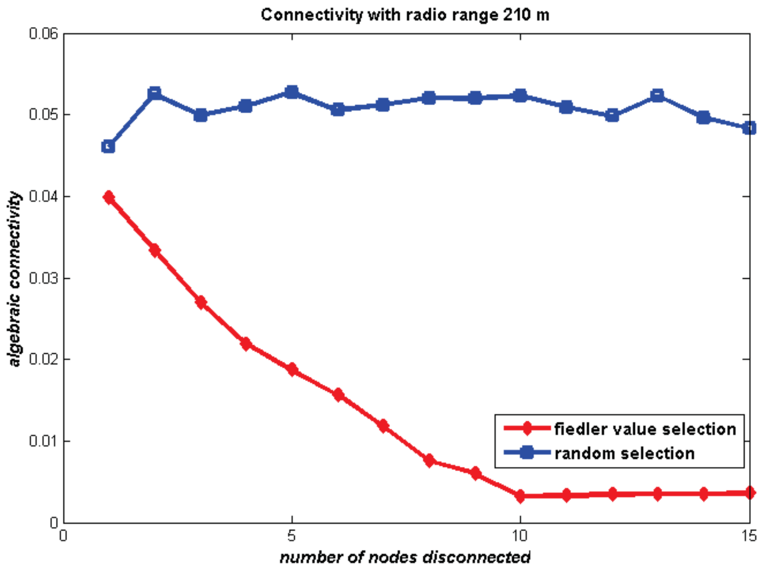


Figure 4.12: The relationship between attackers' positions and graph connectivity in a one-hundred-node network

node can obtain self-evaluated topology information through OLSR routing messages, which requires no extra communication messages. After FVPATA converges, the robustness of the network locally around the Fiedler node is enhanced. Moreover, we proposed a method to solve the problems associated with asymmetric links during the process of increasing power so that nodes will not raise their power indefinitely. The final state for the network is that Fiedler nodes are connected to each other, accompanied by some non-Fiedler nodes participating as bridges among them. Those Fiedler nodes and special non-Fiedler nodes compose a backbone for the network. This structure significantly reduces the number of hops along a route and lowers the latency, yielding a higher network efficiency since only a few nodes increased their power.

The current and previous chapters aim to improve the network connectivity by designing the real network protocols, such as designing overlay routing to reduce the redundancy in the real physical route, localizing the jamming source as function of packet loss ratio and extracting OLSR topology information to dynamically adjust transmission power. In the next chapter, I will list out our future works. I are trying to utilize game theory to understand the jammer's behavior and its impact on network connectivity. I tries to figure out best possible control strategies for network users to increase the network reliability.

Chapter 5

Measuring Throughput Network Connectivity in Ad Hoc Networks

5.1 Overview of the chapter

This chapter focuses on the challenge of maintaining reliable connectivity in an ad hoc network, where interference is possible. To cope with such interference, this work introduces throughput connectivity and weighted throughput connectivity. Throughput connectivity reflects the possibility of establishing communication between nodes for given a signal power level, while weighted throughput connectivity associates the throughput as a weight in the associated network graph. Throughput connectivity is less sensitive to network's parameters than the one based on weighted throughput connectivity. It makes maintaining throughput connectivity protocol less resource consuming (say, by sending less frequently channel state information (CSI)). Whereas, weighted throughput protocol is more efficient in power allocation due to employing a continuous scale in Laplacian matrix. To illustrate these notions, an adaptive transmission protocol that re-allocates transmission power between nodes is considered as an application. It was modeled by a maxmin problem, and solved by Semi-Definite Programming.

5.2 Introduction

In order for networks to be reliable, they must maintain their underlying connectivity, and resist to adversarial attack. An important characteristic of network connectivity is algebraic connectivity, as characterized by the network's Fiedler value, which is the second smallest eigenvalue of the graph's Laplacian. This measures how well-connected the graph is, and has been used to optimize a network's design, and I now survey a few such works. A greedy heuristic algorithm was presented in [77], which adds edges (from a set of candidate edges) to a graph to maximize its algebraic connectivity. A distributed algorithm for the estimation and control of the connectivity of ad hoc networks for random topologies was suggested in [78], while a steepest-descent algorithm was proposed for control of the algebraic connectivity in [79]. The problem of improving network connectivity by adding a set of relays to increase number of links between network's nodes was considered in [80]. Its simplified version was reduced to a semi-definite programming optimization problem. In [81] a genetic algorithm and swarm algorithm were applied for finding the best positions of adding nodes to a network to meet trade off between deployment cost and network's connectivity. A decentralized algorithm to increase the connectivity of a multi-agent system was suggested in [82]. In [83], a problem of finding the best vertex positional configuration to maximize Fiedler value of a weighted graph was studied. Finally note that besides algebraic connectivity the other type of connectivity (such as global message connectivity, worst-case connectivity, network bisection connectivity, and *k*-connectivity, see [84]) are used in networks depending on characteristics to be maintained and methods used,

.

I note that in all of these papers the possibility of establishing a new communication link in a network did not depend on signal interference. Interference, however, can lead to a significant impact since signals sent to establish new communication links also serve as a noise for all the other links and their signals, thereby reducing the network's capacity for maintaining existing communication links. To deal with this problem, in this chapter, two types of connectivity are introduced. First is throughput connectivity, which reflects the possibility of establishing communication between nodes for a given power level. Second is weighted throughput connectivity, which associates with each link a weight corresponding to that link's throughput. To illustrate these notions, two approaches to maximizing connectivity were considered: (a) an adaptive transmission protocol that re-allocates transmission power between nodes, and (b) detecting and eliminating a malicious threat to maintain accumulated connectivity over time slots.

The first problem is modeled by a maxmin problem, and is solved by a generic method. The second problem is modeled by a stochastic game and solved explicitly. Example applications of stochastic games in modeling network security can be found in [85–88] and [89]. I also note that there is quite an extensive literature on detecting an intruder's signal or its source (see, for example, books [90–92], papers on the detection of unknown signals [56,93–95] and on game-theoretic modeling of spectrum scanning [96,97]).

The chapter is organized as follows. In Section 5.3, the new notions for a network's connectivity are defined. In Section 5.4, the problem of designing an optimal transmission protocol to maximize a network's connectivity is considered. In Section 5.5, an optimal scanning protocol to maintain a network's connectivity is explored.

5.3 Throughput network connectivity based on graph theory

I model a wireless network consisting of n nodes in radio range of each other. I denote a node by $v_i = (x_{1i}, x_{2i}), i \in [1, n]$, where (x_{1i}, x_{2i}) is the coordinate for node v_i. Let $V = \{v_i, i \in [1, n]\}$ be the set of all nodes. I assume that, when each node communicates, it emits the same power in all directions. Of course, due to fading gains, pathloss and mutual interference of the signals, not every signal can reach each receiver. Let $\boldsymbol{P} = (P_1, \dots, P_n)$ be the transmission power allocation where signal P_i is the signal power sent by node i to every other node. Interference between signals could take place, and its effect depends on the distance between the receiver and the sender. Namely, the throughput of received signal by node j is

$$T_{ij,\epsilon}(\boldsymbol{P}) = \begin{cases} 0, & \mathrm{SINR}_{ij}(\boldsymbol{P}) < \epsilon, \\ \ln(1 + \mathrm{SINR}_{ij}(\boldsymbol{P})), & \mathrm{SINR}_{ij}(\boldsymbol{P}) \geq \epsilon, \end{cases}$$

where $\epsilon \geq 0$ is a threshold value for SINR, and $\mathrm{SINR}_{ij}(\boldsymbol{P}) = (h_i P_i / d_{ij}^2)/(\sigma^2 + \sum_{k \neq i, k \neq j} h_k P_k / d_{kj}^2)$ with σ^2 is the background noise, h_i is the fading channel gain, and $d_{ij} = \sqrt{(x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2}$

is the distance between node v_i and node v_j.

To define a communication network's topology beyond the nodes, *links* (edges) between nodes have to be established. Note that due to its communication background this topology has to depend on communication type maintained by the network. In this work, I consider *symmetric* communication, i.e., two nodes (say, node i and node j) are considered to be linked if and only if $T_{ij,\epsilon}(\boldsymbol{P})$ and $T_{ji,\epsilon}(\boldsymbol{P})$ are positive. A link means a possibility to maintain communication. Since communication is symmetric, link is undirected. Denote the link between node v_i and node v_j by e_{ij}. Let $E(\boldsymbol{P})$ be the set of all links. It is clear that the graph $\Gamma(\boldsymbol{P}) = (V, E(\boldsymbol{P}))$ is simple, i.e., there is no self loop for each node and there are not multiple links connecting two nodes.

The graph $\Gamma(\boldsymbol{P})$, associated with a network, can be represented by the Laplacian matrix as

$$L_{ij}(\Gamma(\boldsymbol{P})) = \begin{cases} -1, & i \neq j,\ v_i \text{ and } v_j \text{ are linked}, \\ 0, & i \neq j,\ v_i \text{ and } v_j \text{ are not linked}, \\ -\sum\limits_{k=1,k\neq i}^{n} L_{ik}, & i = j, \end{cases}$$

where $L_{ii}(\Gamma(\boldsymbol{P}))$ equals the number of nodes connected with node v_i. Also, it is possible to consider a weighted network by assigning throughput as weight for each link, in which case the weighted network can be represented by a Laplacian matrix as

$$L_{ij}(\Gamma(\boldsymbol{P})) = \begin{cases} -w_{ij}, & i \neq j,\ v_i \text{ and } v_j \text{ are linked}, \\ 0, & i \neq j,\ v_i \text{ and } v_j \text{ are not linked}, \\ -\sum\limits_{k=1,k\neq i}^{n} L_{ik}, & i = j, \end{cases}$$

where $w_{ij} = T_{ij,\epsilon}(\boldsymbol{P}) + T_{ji,\epsilon}(\boldsymbol{P})$ is total throughput of symmetric communication between node v_i and node v_j, and $L_{ii}(\Gamma(\boldsymbol{P}))$ is the total throughput of symmetric communication between node v_i and others nodes.

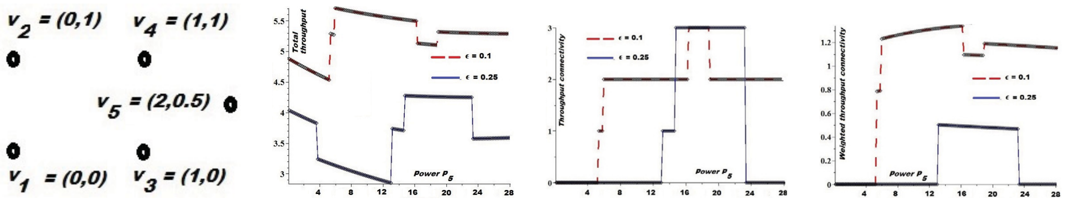
Since $L(\Gamma(\boldsymbol{P}))$ is positive semi-definite and symmetric, its eigenvalues are all nonnegative. By ordering the eigenvalues in an increasing way, I have: $0 = \lambda_1(\Gamma(\boldsymbol{P})) \leq \lambda_2(\Gamma(\boldsymbol{P})) \leq \ldots \leq \lambda_n(\Gamma(\boldsymbol{P}))$. The eigenvector corresponding to the first eigenvalue is always $\boldsymbol{e}^T = (1, \dots, 1)$. The second eigenvalue $\lambda_2(\Gamma(\boldsymbol{P}))$ is the algebraic connectivity of the system, and is an indicator of how connected the graph is, and is also called the Fiedler value. To emphasize that I consider connectivity based on the fact that there is bi-directional throughput (above a threshold value) for a link, I will use the term *throughput connectivity* and *throughput Fiedler value*. For a fixed transmission protocol involving a power assignment $\boldsymbol{P}$, the throughput Fiedler value can be found as solution of the following optimization problem

$$\lambda_2(\Gamma(\boldsymbol{P})) = \min_{\boldsymbol{y}^T\boldsymbol{y}=1, \boldsymbol{e}^T\boldsymbol{y}=0} \boldsymbol{y}^T L(\Gamma(\boldsymbol{P}))\boldsymbol{y}.$$

Let us illustrate the behavior of throughput connectivity by the following example. Let the network consist of five nodes $(0,0)$, $(1,0)$, $(0,1)$, $(1,1)$ and $(2,0.5)$ (Figure 5.1(a)), and $h = 1$, $\sigma^2 = 2$, $\epsilon = 0.1, 0.25$ and $\boldsymbol{P} = (10, 20, 15, 25, 10)$ and P_5 varies from 0.2 to 40. Of course, increasing ϵ yields a decrease in total throughput (Figure 5.1(b)). Throughput connectivity is piece-wise constant versus varying of the power (in our case, P_5, see, Figure 5.1(c)), while weighted throughput connectivity is piece-wise continuous on P_5 (Figure 5.1(d)). Thus, weighted throughput connectivity is more sensitive than throughput connectivity to a variation of the power. In this example, I can observe that there is a continuum where throughput connectivity obtains its maximum, and the value of this maximum is not too sensitive to the threshold ϵ (in the considered example they coincide for $\epsilon = 0.1$ and $\epsilon = 0.25$, and are equal to 3). Also, I can observe that there is a reduction of the set where the throughput connectivity obtains its maximum on reducing the threshold ϵ, but there is no simple monotonic dependence between throughput connectivity and threshold ϵ, For weighted throughput connectivity, such dependence could be observed, as well as the fact that it obtains its maximum for a unique P_5.

5.4 Case study: optimal transmission protocol

The network provider might improve the network's connectivity by varying transmission power vector. Let Π be the set of feasible transmission protocols. For example,



(a) (b) (c) (d)

Figure 5.1: (a) Nodes of the network,(b) Total throughput, (c) Throughput connectivity and (d) Weighted throughput connectivity as functions on P_5.

it could be $\Pi(\overline{P}) = \{\boldsymbol{P} \geq 0 : \sum_{j=1}^{n} P_i = \overline{P}\}$, where $\overline{P}$ is the total power allowed by the network's provider among the nodes. Then, the problem of optimal transmission power assignment is given as the following maxmin problem:

$$\lambda_2(\Gamma(\boldsymbol{P})) = \max_{\boldsymbol{P}\in\Pi(\overline{P})} \min_{\boldsymbol{y}^T\boldsymbol{y}=1,\boldsymbol{e}^T\boldsymbol{y}=0} \boldsymbol{y}^T L(\Gamma(\boldsymbol{P}))\boldsymbol{y}. \tag{5.1}$$

This maximization problem of the second smallest eigenvalue of the Laplacian matrix on its inner parameters is equivalent to the following optimization problem (see, [98]):

$$\begin{aligned} &\max_{\boldsymbol{P},z} z, \\ &\text{subject to} \\ &L(\Gamma(\boldsymbol{P})) - z\boldsymbol{I} \succ \boldsymbol{0}, \quad \boldsymbol{P} \in \Pi(\overline{P}) \text{ and } z > 0, \end{aligned} \tag{5.2}$$

where $\boldsymbol{I}$ is the $n \times n$ identity matrix, and "$\succ$" represents positive definiteness. By definition, Laplacian matrix $L(\Gamma(\boldsymbol{P}))$ is symmetric. Thus, $L(\Gamma(\boldsymbol{P})) - z\boldsymbol{I}$ is also symmetric. Therefore, (5.2) belongs to Semi-Definite Programming (SDP) problems [99]. It can be solved by SDP optimization tools, such as SDPT3 [100, 101], SDPA-M [80, 102] and CSDP [103].

Figure 5.2(a) illustrates dependence of throughput connectivity and weighted throughput connectivity versus total power $\overline{P}$ with $\epsilon = 0.1$. It is interesting that these two forms of connectivity are non-decreasing due to the cooperative re-allocation of transmission power between the nodes. Meanwhile, as it was shown in Figure 5.1, selfishly increasing transmission power of just one node could lead to decreasing the network's connectivity. Of course, cooperative throughput is larger than selfish throughput.

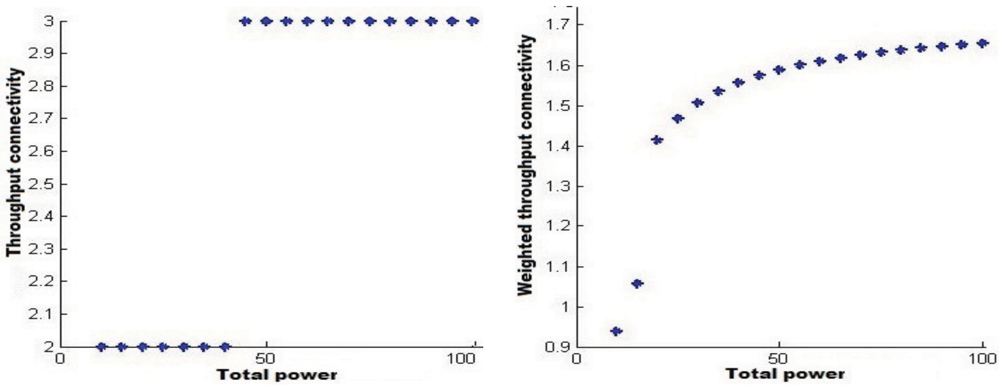


Figure 5.2: Throughput connectivity (left) and weighted throughput connectivity (right) as functions on $\overline{P}$.

5.5 Case study: Optimal scanning protocol

This section considers a problem where an adversary wants to damage connectivity of a network Γ by attacking its nodes, while an IDS (Intrusion Detection System), scanning nodes, intends to detect the adversary to stop his malicious activity. Assume that all the actions (scanning by the IDS and attacking by the adversary) are performed in discrete time slots $1, 2, ..., \infty$. At each time slot, the adversary can choose a node to attack, and the IDS can choose a node to scan. If node i is attacked, then connectivity of the remaining undamaged network $\Gamma_i = \Gamma \backslash \{v_i\}$ is C_i. If the rivals choose different nodes then the IDS gets connectivity for an un-jammed network as an instantaneous payoff, and the game moves to the next time slot and is played recursively with discount factor δ. If the rivals choose the same node, then with probability $1 - \gamma$ the adversary is detected and eliminated from the network. Then, the network keeps on working, and the IDS gets as instantaneous payoff the discounted connectivity C_0 of the whole network. With probability γ the adversary is not detected, the game moves to the next time slot and is played recursively with discount factor δ. This game can be considered as a two-state (1 and 2) stochastic game $G = (G^1, G^2)$. State 1 represents the malicious state in which the network is vulnerable to an attack by the adversary, while state 2 represents the state in which the adversary is detected and is not a threat to the network anymore. Stochastic game $G = (G^1, G^2)$ can be described in matrix form as follows:

$$G^1: \begin{array}{c|cccc} & 1 & 2 & ... & n \\ \hline 1 & C_1|(\gamma, 1-\gamma) & C_2|(1,0) & ... & C_n|(1,0) \\ 2 & C_1|(1,0) & C_2|(\gamma, 1-\gamma) & ... & C_n|(1,0) \\ ... & ... & ... & ... & \\ n & C_1|(1,0) & C_2|(1,0) & ... & C_n|(\gamma, 1-\gamma) \end{array},$$

$$G^2: \begin{array}{c|c} & 1 \\ \hline 1 & C_0|(0,1) \end{array},$$

In state 1, matrix notation is used such that each entry corresponds to a pair of nodes (i, j) chosen by the IDS and the adversary. The value in the left part of each entry is the instantaneous payoff (un-jammed connectivity) to the IDS in this zero-sum stochastic game, while the right part gives the probability distribution over the future states. Thus, if $i \neq j$ then the instantaneous payoff to the IDS is C_j, and the next state is state 1. If $i = j$ then the instantaneous payoff to the IDS is C_i, and the next state is state 1 with probability γ, and it is state 2 with probability $1 - \gamma$. Note that the payoff at the next epoch is discounted with discount rate δ.

In state 2, the rivals are passive, since the adversary is detected and cannot attack the network anymore. The game cannot leave this safe state. At each time slot the IDS obtains the discounted payoff C_0, which is the connectivity of un-jammed network. Thus, the total accumulated discounted payoff in state 2 is equal to $(1 + \delta + \delta^2 + ...)C_0 = C_0/(1 - \delta)$. Thus, the game G is equivalent just to the game G^1 with a single state. The game G^1 has a solution in (mixed) stationary strategies, i.e., the strategies that are independent of history and current time slot. A (mixed) stationary strategy to the IDS is a probability vector $\boldsymbol{p}^T = (p_1, p_2, ..., p_n)$, where p_i is the probability to scan node i and $\boldsymbol{e}^T\boldsymbol{p} = 1$. A (mixed) stationary strategy to the jammer is a probability vector $\boldsymbol{q}^T = (q_1, q_2, \dots, q_n)$, where q_i is the probability to jam node i, and $\boldsymbol{e}^T\boldsymbol{q} = 1$. Solution of the game G^1 is given as a solution to the Shapley (-Bellmann) equation game [104]:

$$\begin{aligned}\mathrm{val}(G^1) &= \max_{\boldsymbol{p}\geq 0, \boldsymbol{e}^T\boldsymbol{p}=1} \min_{\boldsymbol{q}\geq 0, \boldsymbol{e}^T\boldsymbol{q}=1} \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}(\mathrm{val}(G^1))p_i q_j, \\ &= \min_{\boldsymbol{q}\geq 0, \boldsymbol{e}^T\boldsymbol{q}=1} \max_{\boldsymbol{p}\geq 0, \boldsymbol{e}^T\boldsymbol{p}=1} \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}(\mathrm{val}(G^1))p_i q_j,\end{aligned}$$

where

$$A_{ij}(G^1) = \begin{cases} C_j + (1-\gamma)C_0/(1-\delta) + \gamma\delta G^1, & i = j, \\ C_j + \delta G^1, & i \neq j, \end{cases}$$

and $V := \mathrm{val}(G^1)$ is the value of the game, i.e., the equilibrium total accumulated payoff to the IDS. This game G^1 is a stochastic discounted game [104], and so, it has the unique solution in stationary strategies. To find the equilibrium strategies explicitly without loss of generality, it can be assumed that the nodes are arranged in non-increasing order by C_i, i.e., $C_1 \leq C_2 \leq \ldots \leq C_n$. Also, connectivity of an unjammed network is considered larger than connectivity of a damaged network, i.e., $C_0 \geq \max_{1\leq i\leq n} C_i$.

Theorem 1. *The game has a unique equilibrium in stationary strategies. The value of the game and stationary equilibrium strategies are given as follows:*

$$V = \frac{\delta(1-\gamma)C_0/(1-\delta) + \sum_{i=1}^{k} C_i}{k(1-\delta) + \delta(1-\gamma)},$$

$$p_i = \begin{cases} \dfrac{(1-\delta)V - C_i}{\delta(1-\gamma)(C_0/(1-\delta) - V)}, & i \leq k, \\ 0, & i > k, \end{cases}$$

$$q_i = \begin{cases} 1/k, & i \leq k, \\ 0, & i > k, \end{cases}$$

where $k \in \{1, \ldots, n\}$ is an integer given by

$$\varphi_k \leq C_0 < \varphi_{k+1}, \tag{5.3}$$

and φ_i is such that

$$\varphi_i = \frac{(1-\delta)\sum_{j=1}^{i}(C_i - C_j) + \delta(1-\gamma)C_i}{\delta(1-\gamma)}, \quad i \in \{1, \ldots, n\}$$

and $\varphi_{n+1} = \infty$. Since φ_i is increasing and $\varphi_1 = C_1 < C_0$, the k is uniquely defined by (5.3).

5.6 Conclusions

In this chapter, the concept of throughput connectivity and weighted throughput connectivity was introduced to to describe the reliability of a network's communication in the presence of signal interference due to an adversary. In particular, I have shown a difference between selfish and cooperative power allocation, namely, selfishly increasing power by a node might reduce network connectivity, while cooperative allocation improves the connectivity.

This chapter proposed the idea of throughput network connectivity based on the graph theory. Besides commonly used connectivity metrics such as packet reception rate and packet delivery ratio, it can be applied as the connectivity metric, to evaluate the global connection of the network which can reflect the impact of physical parameters such as transmission power and node distances. In the next chapter,game theory is used based on this throughput connectivity to understand the dynamics of connectivity under repeating jamming attack.

Chapter 6

Throughput Connectivity Jamming Game in Ad Hoc Networks

6.1 Overview of the chapter

Due to the open access nature of wireless communications, wireless networks can suffer from malicious activity, such as jamming attacks, aimed at undermining the network's ability to sustain communication links and acceptable throughput. One important consideration when designing networks is to appropriately tune the network topology and its connectivity so as to support the communication needs of those participating in the network. This chapter examines the problem of interference attacks that are intended to harm connectivity and throughput, and illustrates the method of mapping network performance parameters into the metric of topographic connectivity. Specifically, this chapter arrives at anti-jamming strategies aimed at coping with interference attacks through a unified stochastic game. In such a framework, an entity trying to protect a network faces a dilemma: 1) the underlying motivations for the adversary can be quite varied, which depends largely on the network's characteristics such as power and distance; 2) the metrics for such an attack can be incomparable (e.g. network connectivity and total throughput). To deal with the problem of such incomparable metrics, this chapter proposes using the attack's expected duration as a unifying metric to compare distinct attack metrics since a longer-duration of unsuccessful attack assumes a higher cost. Based on this common metric, a mechanism of maxmin selection for an attack prevention strategy is suggested.

6.2 Introduction

The connection properties of a wireless network can degrade easily with adverse environments, such as a tall building that obstruct signals or strong noise that interferes with normal communications. Links breaking is a common phenomena in wireless communications. However, if a malicious jammer purposely breaks a link and separates a node from a network, this harmful behavior can seriously interrupt the normal operation of the network, especially if the node happens to be the hub of several routes. Therefore, investigating the impact of the removal of critical nodes and analyzing the jammer's strategy in choosing a node for an attack is essential to maintain network connectivity in adversarial settings.

Many techniques have been presented to detect the intrusive behavior of an attacker. In [105], a survey of intrusion detection techniques is given. In [106], several host-based and network-based intrusion detection systems (IDS) are surveyed as well as their characteristics are described. In [96], a Bayesian approach was used to detect an intruder in a spectrum band while taking into account whether the intruder sneaks for file-downloading or streaming video. In [97], a Bayesian learning mechanism is used to design a scanning strategy if there is incomplete knowledge whether the intruder is present. In [95], fictitious play from game theory is adopted to classify the type of a jammer based on the historical belief in the throughput under attack uncertainty. In [107] and [108], data mining techniques to recognize anomalies as well as known intrusions are presented. In [49], a lightweight and generic localization algorithm is developed for finding the location of a jamming device after detecting its malicious activity. In [56], an algorithm of localization in peer to peer networks based on Q-learning approach is suggested. However, none of these papers considered the intruder's impact on network connectivity nor mechanisms that can maintain connectivity in the presence of such an adversary.

In this paper, different anti-jamming strategies versus jamming attacks aimed at harming different network characteristics, like connectivity or throughput, are investigated in a uniform framework. In such a situation, an entity intending to protect a network faces a problem that while an adversary might apply a fixed set of jamming tools, the underlying intent or strategy behind an attack can be quite varied, depending largely upon the network's characteristics. In particular, the metrics associated with such an attack can be incomparable (e.g., network's connectivity and total throughput). To deal with the incomparable metrics problem, this paper makes the following contributions:

1. A general stochastic game involving the protection of a network, where a jammer might sense nodes' scanning and switch to a hiding (i.e. silent) mode, is suggested. In the considered model, the meaning of the instantaneous costs depends on the type of jamming attack strategy being applied. For the attack aimed at harming network connectivity, the instantaneous costs for the jammer are described by the algebraic connectivity or Fiedler value of the network [109, 110]. For the attack aimed at harming throughput, the costs are the network's throughput. It is important to note that since the network protector aims to maximize cost of an adversary's attack, this game is fundamentally about the prevention of an attack rather than about the network's protection.

2. Since the longer an attack is unsuccessful leads to higher cost, we propose a unifying metrics that makes it possible to compare such attacks whose metrics of success (e.g. harming connectivity or throughput) would otherwise be incompatible. In particular, we propose the use of the attack's expected duration and, based on this common metric, we arrive at a mechanism of maxmin selection for nodes' scanning strategy is suggested.

The paper is organized as follows. In Section 6.3, related works are discussed. In Section III, jamming attacks aimed at undermining network connectivity and network throughput are presented, as well as a preliminary overview of algebraic connectivity. In Section IV, the problem of preventing an attack against a network is formulated as a stochastic game, and it is solved explicitly. In Section V, a tool for comparing defenses against jamming attacks aimed to harm different network characteristics is developed. In Section VI, results of numerical evaluation for the optimal solutions and their dependence on network characteristics are supplied. Finally, in Section 6.8, conclusions are given.

6.3 Related works

Studies that explore the connectivity of networks and their topological properties can be found in the literature and a sample includes [75, 111, 112]. The mostly widely adopted approach for summarizing a network's topological connectivity involves the calculation or prediction of node degree from statistical results. Network connectivity data have been collected from a variety of real networks, such as social networks [113] and citation networks [114], and is complemented by mathematical models, such as scale-free networks, where the node degree follows a probability distribution that decays in a power-law, or a Poisson random networks, whose nodal probability distribution follows a Poisson distribution.

An important research area that applies to all of the network models mentioned above involves studying network connectivity under malicious attacks. While these attacks can happen at each network layer, most research about network connectivity traditionally focuses on designing secure routing protocols by which packets can route around a black hole or wormhole in networks [115–117]. Those routing protocols usually aim to find the most efficient and free path in a topological graph after an attack happens. On the other hand, the impact of a broken single link or removal of a node in a path, and the resulting diffusion of attack damage across the broader network context has been studied much less, particularly when the connectivity issues appear at the physical layer.

Ensuring the robustness of the physical layer typically involves examining links in isolation (e.g. robust error coding), and notably separate from the broader network context. The robustness of *networks* at the physical layer should examine the network's performance after one node/link, or even several nodes/links, are degraded or removed at the physical layer. For example, an attacker can randomly delete several nodes or strategically delete nodes according to his purposes though targeted interference, aimed at greedily removing nodes with higher degree first or deleting nodes in high density areas in order to exacerbate the damage. To the best of our knowledge, most prior research into the resilience of networks are statistical and they fail to consider the interaction between legitimate nodes and the attacker and, moreover, tend to consider that the jammer behavior is random in how it eliminates nodes, without a deeper strategy behind how to maximize its attack effectiveness.

Game theory is a natural tool to investigate and rationalize about a jammer's behavior. Game theory investigates the interactions between players to arrive at equilibrium strategies for both sides [118]. In [119], a survey of works that applied game theory to deal with network security at each layer is given. Physical layer security can be described by game theory both in the form of a Nash game and a Stackelberg game. Game theory papers at the physical layer security often model the rational behaviors of a jammer, or an eavesdropper, or cooperative behavior between them, to solve the problem of allocating transmission power or increasing transmission rate. Typically, the utility function being employed is Shannon capacity, SINR (signal to interference and noise ratio), information entropy or bit error equations. There are a limited set of works dealing with maintaining the connectivity of the network topology. In [120], a problem of minimizing the probability that the spanning tree disrupted by an adversary attack was studied. In [121], to identify key players engaged in attacking a network, the Shapley value was applied. In [122], a problem with two types (good and bad) of users was studied by a repeated game, where good users were willing to trade energy for connectivity depending on neighbors' behaviours, while bad users try to destroy connectivity and lure the good users to waste energy. In this paper we consider the game where users' throughput and network connectivity are combined in a unified framework.

6.4 Formulation of the problem

As a motivating scenario, we consider a zone that involves n nodes (users) allocated at points (y_{1i}, y_{2i}), $i \in [1, n]$ and operating in a P2P full duplex communication fashion, which allows nodes to communicate in both directions. Let $e_{i,j}$ be *a duplex communication link* (a channel) for communication between node i and j. A possible connection between any two nodes is defined by the channel's condition, the mutual distance between nodes, receiver threshold, transmission power and transmission protocols, i.e., the collision avoidance protocol in MAC layer. Let $\boldsymbol{h} = \{h_{i,j}\}_{i,j=1}^{n}$ be an $n \times n$ matrix describing the communication capabilities between nodes (fading channel gains). Generally, this matrix might be non-symmetrical, i.e., the component h_{ij}, mapping communication from node i to node j, might be different from the component h_{ji}, mapping communication from node j to node i. Some components of matrix $\boldsymbol{h}$ might be zero, reflecting the fact that either communication between these nodes is not allowed, or this node has no intent to communicate with another. For a particular, $h_{ii} = 0$ for any i, as there is no need for a node to communicate with itself. Therefore, based on these conditions, the complete possible communication topology is already determined. Let $P_{i,j}$, $i, j \in \{1, \dots, n\}$ be transmission protocol between nodes, i.e., $P_{i,j}$ represents transmission power from sender i to receiver j on channel $e_{i,j}$. Then, by applying a Shannon-type formulation for channel capacity, the throughput for communication node i to node j is $T_{i,j} = \ln\left(1 + h_{i,j}P_{i,j}/(\sigma^2 + \sum_{k=1,k\neq j}^{n} h_{k,j}P_{k,j})\right)$, where σ^2 is the background noise.

In the zone, besides of the legitimate nodes, an adversary jammer equipped with limited power is present to harm communication. Its location is given by the coordinates $\boldsymbol{x} = (x_1, x_2)$. The effect of jamming (jammed throughput) for communication node i to node j is $\ln\left(1 + h_{i,j}P_{i,j}/(\sigma^2 + \sum_{k=1,k\neq i}^{n} h_{k,j}P_{k,j} + g_j J/d_j^2)\right)$ which depends on the distance $d_j = \sqrt{(x_1 - y_{j1})^2 + (x_2 - y_{j2})^2}$ between the jammer and the receiver, the fading channel gain g_j, as well as the jamming power J being applied. Since the jammer has a power limitation, he cannot effectively impact the communication of nodes located far away. However, if the attacker is allocated close by a particular node, then the jammer can effectively jam that node from all incoming traffic received. Due to all incoming messages for the jammed node are blocked, the acknowledge messages(ACK) corresponding to his request to establish communication with other nodes are also blocked [123]. Thus, the jammed node cannot recognize its neighboring nodes, and hence it cannot communicate with them. So, when the jammer is attacking a node, it can disrupt bi-lateral communications (incoming and outcoming) for that node. In the meantime, the RTS/CTS (Request to Send/Clear to Send) problem can also make the receiver detecting the existence of a hidden terminal and ceasing the transmission to the target. Ultimately the jammer achieves his goal by blocking the whole receiving and sending functions of the target. To describe the effects of blocking such bilateral communications, we assume the jammer's ability to block the communications is equivalent to its ability to disrupt the communication links in bilaterally.

Note there are lots of types of jamming attacks. A reader can find comprehensive surveys of such threats in [124]. In this research, we introduce a new type of attacks which targets the network's connectivity, and compare it to the jamming attack which targets the network's throughput.

6.4.1 Cost of breaking connectivity attack

In this section, we describe breaking connectivity attack and its cost. This is a jamming attack targeting to break duplex communication links between nodes. In order to break a link from sender to receiver in its physical layer, the received SINR must be smaller than the threshold ω. Let the threshold be the same for all the nodes. Then we can express the condition of a broken link from node i to node j by

$$\frac{h_{i,j}P_{i,j}}{\sigma^2 + \sum_{k=1,k\neq i}^{n} h_{k,j}P_{k,j} + g_j J/d_j^2} < \omega. \tag{6.1}$$

Thus, to break communication from node i to node j, the following induced jamming power has to be applied

$$\left\lfloor \frac{h_{i,j}P_{i,j}}{\omega} - \sigma^2 - \sum_{k=1,k\neq i}^{n} h_{k,j}P_{k,j} \right\rfloor_{+} \leq \frac{g_j J}{d_j^2}, \tag{6.2}$$

where $\lfloor \xi \rfloor_+ = \max\{\xi, 0\}$. Due to block ACK to break all the bilateral links on node j from other nodes, the following induced jamming power has to be applied

$$\max_{i, i \neq j} \left\lfloor \frac{h_{i,j} P_{i,j}}{\omega} - \sigma^2 - \sum_{k=1, k \neq i}^{n} h_{k,j} P_{k,j} \right\rfloor_+ \leq \frac{g_j J}{d_j^2}. \tag{6.3}$$

This condition can be achieved by having sufficient closely positioned adversary to node j in spite of its limited jamming resource. Such adversary's strategy allows the elimination of a selected node from networkâĂŹs communication to cause network disruption in terms of connectivity. To deal with the remaining network connectivity, a concept of Fiedler value [125] developed in spectral graph theory can be applied.

Fiedler value is the second smallest eigenvalue of the Laplacian matrix, $L(V, E)$, of a network's topological graph, $\Gamma(V, E)$ where V is a vertex set and E is the edge set connecting two vertices in the graph. The Fiedler value is always non-negative, and its amplitude is proportional to the graph connectivity. It is zero if and only if the graph is disconnected. The number of zero eigenvalues in the eigenvalue set of L equals to the number of connected components in a graph. According to [71], the Fiedler value represented by λ_1, of a graph, Γ, can be obtained by the following eigenvalue optimization problem.

$$\begin{aligned} &\lambda_1 = \min \boldsymbol{y}^T L(V, E) \boldsymbol{y} \\ &st.\ \boldsymbol{y}^T \boldsymbol{y} = 1 \ and \ \boldsymbol{y}^T \boldsymbol{e} = 0 \end{aligned} \tag{6.4}$$

where $\boldsymbol{y}$ is a vector which does not equal to $\boldsymbol{e}$, with $\boldsymbol{e}^T = (1, 1, ..., 1)$, and M^T is a transpose to matrix M.

The Laplacian matrix of a given graph is defined as follows: Given a graph $\Gamma(V, E)$ without self cycles and multiple links between two nodes, the Laplacian matrix L is calculated by

$$L(V, E) = D(V, E) - A(V, E) \tag{6.5}$$

where $D(V, E)$ is a diagonal matrix whose diagonal entry contains the degrees for each node. $A(V, E)$ is the adjacency matrix with each entry being a value of zero or one when nodes are connected to each other. In addition, its diagonal is zero since

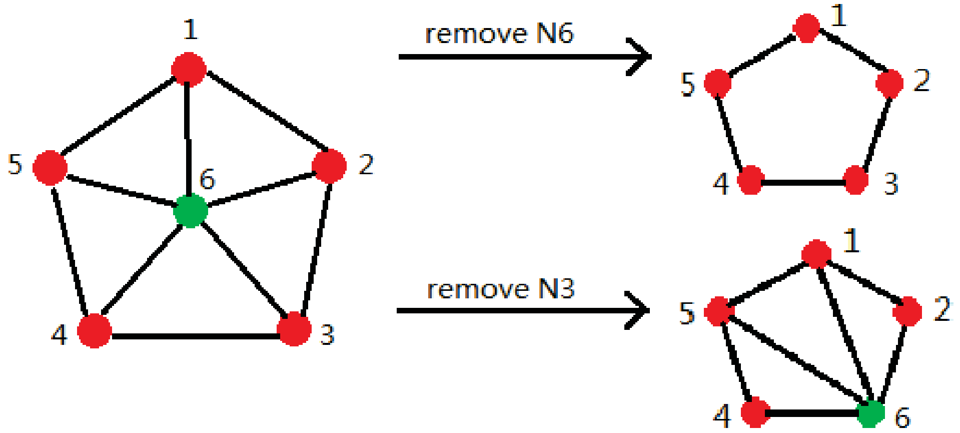


Figure 6.1: Fiedler Values of different remaining graph are comparable when the number of nodes is the same

$\Gamma(V, E)$ has no self cycles.

In a network topology graph, V is the set of users, and E is the set of links which support duplex communications, we assume a link exists if and only if bilateral communication between two users is possible.

Since the amplitude of Fiedler value is proportional to the connectivity of networks which is known as the algebraic connectivity, we adopt the algebraic connectivity as our connectivity metric. The smaller the Fiedler value is, the larger the negative impact is onto the networks. Assume λ_i is Fiedler value of a graph $\Gamma\backslash\{i\}$ by removing all the incident edges attached to node i in the graph Γ. Here we consider a jammer can only turn off one node. These Fiedler values,$\{\lambda_i(\Gamma\backslash\{i\})|i = 1, 2, \ldots, n\}$, on remaining graphs obtained by removing a different node from the same graph, Γ, are comparable in terms of graph connectivity although they have different connections on the same number of vertexes. $\lambda_i(\Gamma\backslash\{i\})$ does not relate to the position of node in the graph and nodes' labels. Figure 6.1 shows an example that the connectivity in different graphs obtained by removing different nodes from the same graph is comparable as long as the number of nodes in remaining graph is the same.

If the jammer aims to reduce network connectivity, Fiedler Value $\{\lambda_i\}$ can be considered as the cost of such adversary's attack. Namely, the jammer's cost of attack to disrupt connectivity is

$$\bar{\lambda}_i = \lambda_i(\Gamma\backslash\{i\}). \tag{6.6}$$

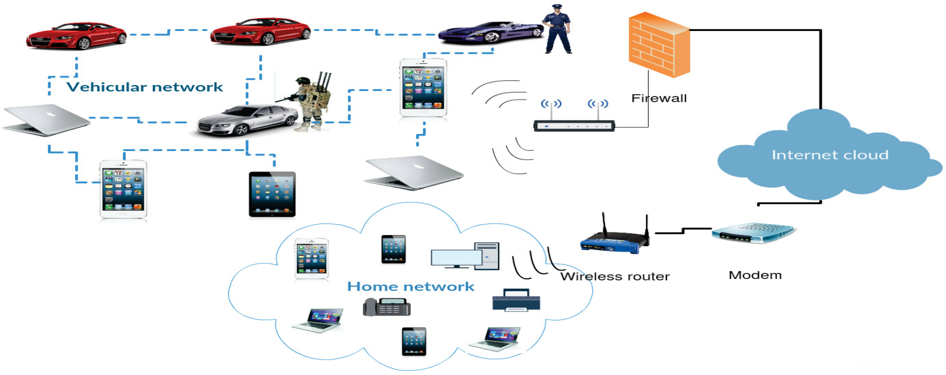


Figure 6.2: A connectivity attack in an ad hoc network

6.4.2 Cost of jamming throughput attack

If the adversary targets to harm network's throughput, then the total number of throughput for the unaffected network can be considered as a cost of such attack. If the adversary has a selected node i for low-power jamming attack, and because this attack also blocks ACK, the total throughput for the rest of the network, or the cost of the throughput jamming attack is given as follows:

$$\bar{\lambda}_i = \sum_{l,j=1,j\neq i,l\neq i}^{n} \ln\left(1+\frac{h_{l,j}P_{l,j}}{\sigma^2+\sum_{k=1,k\neq j}^{n} h_{k,j}P_{k,j}}\right). \tag{6.7}$$

6.5 A stochastic game of intrusion prevention

We consider, as a motivating application, the problem of mitigating an attack directed against an ad hoc network, as depicted in Figure 6.2. In this scenario, a jammer aims to hurt network by choosing a node to direct interference against, while the network itself aims to reduce the harm this attack has on the network by scanning to detect the harm and ultimately force the adversary into more costly option for conducting its attack.

The category of jammer's attack is fixed in the entire intrusion, which might either be a category of jamming throughput or disrupting connectivity attack. The jammer senses a node which could mostly jeopardize network connectivity through blocking its communication. In some probability, if the victim node determined by a jammer is also simultaneously scanned by the scanner, since the scanner is present (a jammer can observe presence of authority by only watching or executing some detection techniques which is why he has no intention to perform jamming attack due to the fear of being caught), the jammer switches to the hiding (silent) mode, and if he is not caught, he can continue his attack. However, if the node he chooses is not scanned, the jammer performs an attack. Let C_h be the cost of hiding mode for the jammer corresponding to an applied category of the attack. Let α be the probability to be detected in the hiding mode, and $1-\alpha$ be the probability not to be detected. Thus, the instantaneous cost to the jammer combines the expected hiding cost and cost of network penetration in future if the jammer is not caught. Please note our method can also be applied to hierarchical networks (say, Wifi networks) by assigning more weights to critical nodes, such as access points and cluster heads.

Therefore, we propose the strategies to prevent such attacks in addition to the design of a defense network. Assuming the instantaneous payoff to the legitimate authority equals the instantaneous cost to the jammer. This recursively played zero-sum game G_γ can be considered as a single state stochastic game ([126]), which we are going to solve by stationary strategies (i.e., the strategies which do not depend on history and time slot), and it is given as follows:

$$G_\gamma = \begin{array}{c} \\ 1 \\ 2 \\ ... \\ n \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & ... & n \end{array} \\ \begin{pmatrix} C_h+\gamma G_\gamma & \bar{\lambda}_2 & ... & \bar{\lambda}_n \\ \bar{\lambda}_1 & C_h+\gamma G_\gamma & ... & \bar{\lambda}_n \\ ... & ... & ... & ... \\ \bar{\lambda}_1 & \bar{\lambda}_2 & ... & C_h+\gamma G_\gamma \end{pmatrix} \end{array}, \tag{6.8}$$

where rows correspond to the authority's strategies, i.e., chosen nodes to scan, and columns correspond to the jammer's strategies, i.e., chosen nodes to attack.

Let us describe in details the components of this matrix. Assume the authority has chosen strategy i and the jammer has chosen strategy j. If $i \neq j$, node j is jammed successfully, the jammer suffers the instantaneous cost $\bar{\lambda}_j$, and the game is over. If $i = j$ then the jammer switches to the hiding mode paying instantaneous cost C_h. With probability α, the jammer will be detected, and the game is over. However, if the jammer is not detected, with probability $1-\delta$, he stops the attempts and exits the game. The game is over. Whereas, with probability δ the jammer keeps playing the game recursively. Therefore, the instantaneous reward for authority is $\alpha C_h + (1-\alpha)\,[C_h + \delta \cdot \mathrm{val}(G_\gamma)]$. Then, the conditional probability to keep on the jamming attacks is $\gamma = (1-\alpha)\delta$, and with this probability the game G is played recursively with the expected instantaneous jammer's costs accumulated as $C_h + \gamma G_\gamma$. Since $\gamma < 1$, it can be considered as a discount factor and is the condition that guarantees the convergence of the solution. Here employing stochastic game tools is quite natural, since the authority and the jammer have opposing objectives, and it is uncertain how persistent the jammer can manage to perform its malicious attack before it is detected. The applications of stochastic games in modeling network security can be found in [85–88] and [89]. Finally, note that the game (6.8) can be used to model different types of attacks by assigning appropriate content of its parameters. Accordingly, the variable, $\bar{\lambda}_i$, can correspond to either the network's connectivity in a connectivity disruption attack, or the network's throughput in a throughput disruption attack.

Game G_γ has a solution in (mixed) stationary strategies, i.e., the strategies that are independent of history and current time slot. A (mixed) stationary strategy to the authority is a probability vector $\boldsymbol{p}^T = (p_1, p_2, ..., p_n)$, where p_i is the probability to scan node i and $\boldsymbol{e}^T\boldsymbol{p} = 1$. A (mixed) stationary strategy to the jammer is a probability vector $\boldsymbol{q}^T = (q_1, q_2, \dots, q_n)$, where q_i is the probability to jam node i, and $\boldsymbol{e}^T\boldsymbol{q} = 1$. Solution of the game G_γ is given as a solution to the Shapley (-Bellmann) equation game [126]:

$$\begin{aligned}\mathrm{val}(G_\gamma) &= \max_{\boldsymbol{p}\geq 0, \boldsymbol{e}^T\boldsymbol{p}=1} \; \min_{\boldsymbol{q}\geq 0, \boldsymbol{e}^T\boldsymbol{q}=1} \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}(\mathrm{val}(G_\gamma))p_i q_j, \\ &= \min_{\boldsymbol{q}\geq 0, \boldsymbol{e}^T\boldsymbol{q}=1} \; \max_{\boldsymbol{p}\geq 0, \boldsymbol{e}^T\boldsymbol{p}=1} \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}(\mathrm{val}(G_\gamma))p_i q_j,\end{aligned} \tag{6.9}$$

$$A_{ij}(x) = \begin{cases} C_h + \gamma x, & i = j, \\ \bar{\lambda}_j, & i \neq j, \end{cases} \tag{6.10}$$

and $V_\gamma := \mathrm{val}(G_\gamma)$ is the value of the game, i.e., the optimal accumulated cost to the jammer.

Without loss of generality we can assume that all the nodes have different jamming costs, i.e., $\bar{\lambda}_i \neq \bar{\lambda}_j$ for $i \neq j$. Also, let all the nodes are indexed in ascending order by $\bar{\lambda}_i$, i.e.,

$$\bar{\lambda}_1 < \bar{\lambda}_2 < \ldots < \bar{\lambda}_n. \tag{6.11}$$

Despite the fact that the stochastic game considered has $n \times n$ instantaneous payoff matrix, we can obtain the solution explicitly from the following theorem given below:

Theorem 2. *The stochastic game G_γ has an equilibrium in stationary strategies $(\boldsymbol{p}, \boldsymbol{q})$ and the value V_γ given as follows:*

(a) Let

$$C_h/(1-\gamma) < \bar{\lambda}_1. \tag{6.12}$$

Then

$$\begin{aligned} V_\gamma &= \bar{\lambda}_1, \\ p_i &\begin{cases} = 0, & i = 1, \\ \geq \dfrac{\bar{\lambda}_i - \bar{\lambda}_1}{\bar{\lambda}_i - C_h - \gamma\bar{\lambda}_1}, & i \geq 2, \end{cases} \\ q_i &= \begin{cases} 1, & i = 1, \\ 0, & i \geq 2. \end{cases} \end{aligned} \tag{6.13}$$

(b) Let

$$\bar{\lambda}_1 \leq C_h/(1-\gamma) < \lambda_2. \tag{6.14}$$

Then

$$V_\gamma = C_h/(1-\gamma),$$
$$p_i(x) = \begin{cases} 1, & i = 1, \\ 0, & i \geq 2, \end{cases} \tag{6.15}$$
$$q_i(x) = \begin{cases} 1, & i = 1, \\ 0, & i \geq 2. \end{cases}$$

(c) Let

$$\bar{\lambda}_k < C_h/(1-\gamma) \leq \bar{\lambda}_{k+1} \tag{6.16}$$

with $\lambda_{n+1} = \infty$*, and* $m \in [1, k]$ *be such that*

$$\varphi_m^{k+1} \leq 1 < \varphi_{m+1}^{k+1}, \tag{6.17}$$

with

$$\varphi_s^{k+1} = \sum_{i=1}^{s} \frac{\bar{\lambda}_s - \bar{\lambda}_i}{C_h + \gamma\bar{\lambda}_s - \bar{\lambda}_i} \text{ for } s \leq k \tag{6.18}$$

and $\varphi_{k+1}^{k+1} = \infty$*. Note that, by (6.16),* φ_s^{k+1} *is increasing from zero for* $s = 1$ *to infinity for* $s = k+1$*. Thus, m is uniquely defined by (6.17).*

Then,

$$p_i = \begin{cases} \dfrac{V_\gamma - \bar{\lambda}_i}{C_h + \gamma V_\gamma - \bar{\lambda}_i}, & i \leq m, \\ 0, & i > m, \end{cases}$$
$$q_i = \begin{cases} \dfrac{1/(C_h + \gamma V_\gamma - \bar{\lambda}_i)}{\sum_{j=1}^{m} 1/(C_h + \gamma V_\gamma - \bar{\lambda}_j)}, & i \leq m, \\ 0, & i > m, \end{cases} \tag{6.19}$$

and V_γ is an unique root of the equation

$$F_m(V_\gamma) := \sum_{i=1}^{m} \frac{V_\gamma - \bar{\lambda}_i}{C_h + \gamma V_\gamma - \bar{\lambda}_i} = 1. \tag{6.20}$$

Proof: First note that V_γ, $\boldsymbol{p}$ and $\boldsymbol{q}$ is a solution of Shapley equation (6.9) if and only if

$$V_\gamma = \nu, \tag{6.21}$$

$$\begin{aligned} &\max\ \nu, \\ &\sum_{i=1}^{n} A_{ij}(V_\gamma) p_i \geq \nu,\ i \in \{1, \dots, n\}, \\ &\boldsymbol{p} \text{ is probability vector,} \end{aligned} \tag{6.22}$$

$$\begin{aligned} &\min\ \nu, \\ &\sum_{j=1}^{n} A_{ij}(V_\gamma) q_i \leq \nu,\ j \in \{1, \dots, n\}, \\ &\boldsymbol{q} \text{ is probability vector.} \end{aligned} \tag{6.23}$$

Taking into account (6.10) and the fact that $\boldsymbol{p}$ and $\boldsymbol{q}$ are probability vectors yield that these LP problems (6.22) and (6.23) are equivalent to

$$\begin{aligned} &\max\ \nu, \\ &(C_h + \gamma V_\gamma - \bar{\lambda}_i) p_i + \bar{\lambda}_i \geq \nu,\ i \in \{1, \dots, n\}, \\ &\boldsymbol{p} \text{ is probability vector,} \end{aligned} \tag{6.24}$$

$$\begin{aligned}&\min\ \nu,\\ &(C_h+\gamma V_\gamma-\bar{\lambda}_i)q_i+\sum_{j=1}^{n}\bar{\lambda}_j q_j\le \nu,\ j\in\{1,\dots,n\},\\ &\boldsymbol{q}\text{ is probability vector}\end{aligned}\tag{6.25}$$

Then, (6.21), (6.24) and (6.25) imply that V_γ, $\boldsymbol{p}$ and $\boldsymbol{q}$ is a solution of Shapley equation (6.9) if and only it the following conditions hold:

$$(C+\gamma V_\gamma-\bar{\lambda}_i)q_i+\sum_{j=1}^{n}\bar{\lambda}_j q_j\begin{cases}=V_\gamma, & p_i>0,\\ \le V_\gamma, & p_i=0,\end{cases}\tag{6.26}$$

$$(C+\gamma V_\gamma-\bar{\lambda}_i)p_i+\bar{\lambda}_i\begin{cases}=V_\gamma, & q_i>0,\\ \ge V_\gamma, & q_i=0.\end{cases}\tag{6.27}$$

Let (6.12) hold. Then, by (6.11), (6.26) and (6.27), there is no i such that $p_i>0$ and $q_i>0$. Also, $q_1=1$ and $p_1=0$. Substituting them into (6.26) and (6.27) implies (a).

Let (6.12) do not hold. Then, by (6.11), (6.26) and (6.27), there is a m such that $p_i>0$ and $q_i>0$.

$$p_i\begin{cases}>0, & i\le m,\\ =0, & i>m\end{cases}\text{ and } q_i\begin{cases}>0, & i\le m,\\ =0, & i>m.\end{cases}\tag{6.28}$$

Let $m=1$. Then, by (6.11), (6.26) and (6.27), the condition (6.14) has to hold, and (b) follows.

Let (6.16) hold. Note that

$$\max\left\{\frac{\bar{\lambda}_i-C_h}{\gamma},\bar{\lambda}_i\right\}=\begin{cases}\dfrac{\bar{\lambda}_i-C_h}{\gamma}, & \bar{\lambda}_i\ge\dfrac{C_h}{1-\gamma},\\ \bar{\lambda}_i, & \bar{\lambda}_i\le\dfrac{C_h}{1-\gamma}.\end{cases}\tag{6.29}$$

Since $m>1$, by (6.11), (6.26), (6.27) and (6.28) $\boldsymbol{p}$ and $\boldsymbol{q}$ have to have the form given by (6.19). Then, since sums of the components of vector $\boldsymbol{p}$ equals 1, V has to be given as a root of the equation (6.20). It is only left to show that this equation has a unique root. Let $m < k$. Then, by (6.11) and (6.29), F_m is increasing in $[\bar{\lambda}_m, \bar{\lambda}_{m+1}]$ such that, by (6.17), $F_m(\bar{\lambda}_m) = \varphi_m^{k+1} \leq 1 < \varphi_{m+1}^{k+1} = F_m(\bar{\lambda}_{m+1})$. Thus, V is uniquely defined. Let $m = k$. Then, by (6.11) and (6.29), F_k is increasing in $[\bar{\lambda}_k, (\bar{\lambda}_k - C)/\gamma]$, and $F_k((\bar{\lambda}_k - C)/\gamma) > 1$, and (c) follows.

Theorem 2 allows to observe some interesting properties of the solution.

If hiding cost C_h is too big, namely, $C_h \geq \bar{\lambda}_n$, then all the nodes will be under attack, and thus, have to be scanned, i.e., $p_i > 0$ and $q_i > 0$ for any i, and the value of the game is the unique root of the equation $F_n(V_\gamma) = 1$. Also, the value V_γ of the game is increasing on C_h and γ where including $\gamma = 1$.

Since the game G_0 is one time slot game, it is just a matrix game. Its solution is given in the following theorem in the closed form.

Theorem 3. *One time slot matrix game G_0, which is the limit of the stochastic game G_γ for γ tends to zero, has value $V_0 = V(C_h)$ and the equilibrium strategies $\boldsymbol{p}$ and $\boldsymbol{q}$ given as follows:*

(a) Let

$$C_h < \bar{\lambda}_1. \tag{6.30}$$

Then

$$\begin{aligned} V(C_h) &= \bar{\lambda}_1, \\ p_i &\begin{cases} = 0, & i = 1, \\ \geq \dfrac{\bar{\lambda}_i - \bar{\lambda}_1}{\bar{\lambda}_i - C_h}, & i \geq 2, \end{cases} \\ q_i(x) &= \begin{cases} 1, & i = 1, \\ 0, & i \geq 2. \end{cases} \end{aligned} \tag{6.31}$$

(b) Let

$$\bar{\lambda}_1 \leq C_h < \bar{\lambda}_2. \tag{6.32}$$

Then

$$V(C_h) = C_h,$$
$$p_i = \begin{cases} 1, & i = 1, \\ 0, & i \geq 2, \end{cases} \tag{6.33}$$
$$q_i = \begin{cases} 1, & i = 1, \\ 0, & i \geq 2. \end{cases}$$

(c) Let

$$\bar{\lambda}_k < C_h \leq \bar{\lambda}_{k+1} \tag{6.34}$$

and m be given by (6.17). Then,

$$V(C_h) = \frac{1 + \sum_{j=1}^{m} \bar{\lambda}_j / (C_h - \bar{\lambda}_j)}{\sum_{j=1}^{m} 1/(C_h - \bar{\lambda}_j)},$$
$$p_i = \begin{cases} \dfrac{V(C_h) - \bar{\lambda}_i}{C_h - \bar{\lambda}_i}, & i \leq m, \\ 0, & i > m, \end{cases} \tag{6.35}$$
$$q_i = \begin{cases} \dfrac{1/(C_h - \bar{\lambda}_i)}{\sum_{j=1}^{m} 1/(C_h - \bar{\lambda}_j)}, & i \leq m, \\ 0, & i > m. \end{cases}$$

Theorem 3 also allows to suggest two procedures to find the value of the stochastic game.

Theorem 4. *The value of the stochastic game G_γ is given as follows*

$$V_\gamma = \frac{x - C_h}{\gamma},$$

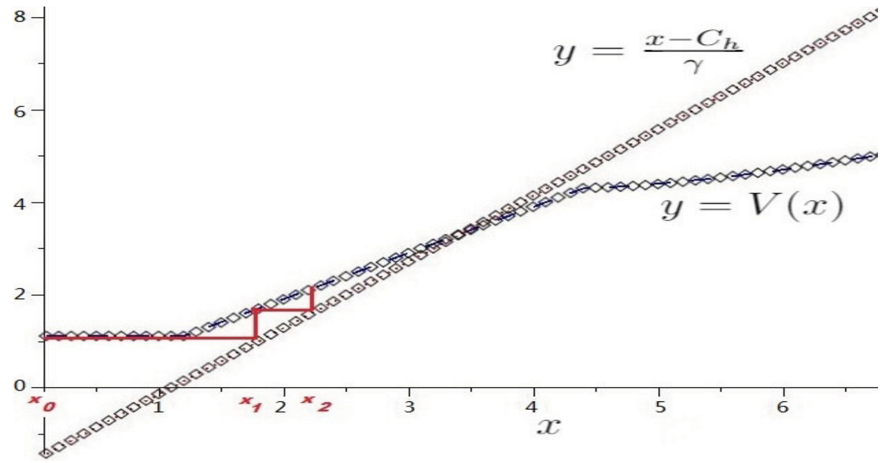


Figure 6.3: Convergence of iterative procedure for $n = 5$, $\bar{\lambda} = (1.11, 4.31, 6.12, 8.31, 9.11)$, $C_h = 1$ and $\gamma = 0.7$.

where $x \geq C_h$ is a unique root of the equation

$$\frac{x - C_h}{\gamma} = V(x). \tag{6.36}$$

The unique root of (6.36) can be found by

(a) *iterative procedure $x_0 = C_h$, $x_{i+1} = \gamma V(x_i) + C_h$, $i = 0, 1, \ldots$ until $|x_{i+1} - x_i| \leq \epsilon$ with ϵ is tolerance.*

(b) *bisection method since $(x - C_h)/\gamma - V(x) = -C_h/\gamma - v(0) < 0$ for $x = 0$ and $(x - C_h)/\gamma - v(x) > 0$ for enough large x.*

Figure 6.3 illustrates convergence of iterative procedure to the equilibrium point.

6.6 Maxmin selection of scanning strategy

In reality, the jammer can deteriorate network performance in many aspects such as reducing either its connectivity or throughput or secrecy communication. Motivated by these different categories of malicious activity, the jammer could vary the corresponding optimal strategies. However, the authority might have no knowledge of the jammer's motivation for an attack, and so about the strategy employed. The authority might only know the set of all possible motivations and the corresponding optimal strategies used by the jammer.

Under this situation, the need for comparing these strategies arises since they aim to achieve different metrics. Say, connectivity is a metric for the strategies which aim to jeopardize network connectivity, whereas, throughput is a metric for the strategies which aim to harm the throughput. However, in spite of differences in metrics, the ultimate goal of a jammer is to speed up the process of completing the attack since long time commitment involves more cost. Thus, the expected time of successful attack can be considered as a common metric for all the categories of malicious activity, where the authority wants to maximize this metric while the jammer aims to minimize it. If we assume the rival chooses a specific category and he follows the category over time before completing the attack, then the expected jamming time, T, before a successful attack appears, can be represented as following:

$$T(\boldsymbol{p}, \boldsymbol{q}) = \sum_{t=1}^{\infty} t \left[\left(\sum_{i=1}^{n} \gamma p_i q_i \right)^{t-1} \left(1 - \sum_{i=1}^{n} \gamma p_i q_i \right) \right] \tag{6.37}$$

$$= \frac{1}{1 - \gamma \boldsymbol{p}^T \boldsymbol{q}}. \tag{6.38}$$

Where $\boldsymbol{q}$ is a probability vector which represents a category of strategies employed by the jammer. $\boldsymbol{p}$ is a probability vector which represents a category of strategy applied by the authority to scan the attack. Thus, these strategies depend on the category of malicious activity chosen by the jammer.

Although suggested approach might be applied to any category of an attack, to get insight of the problem, we focus only on two which are also the most important metrics for network performance, namely, network's connectivity and network's throughput. We denote these metrics (connectivity, throughput) by the symbols, "c" and "t". The optimal strategy pair, $(\boldsymbol{p}_c, \boldsymbol{q}_c)$ for dealing with attack aiming to destroy connectivity, was found in the previous section. The optimal strategies $(\boldsymbol{p}_t, \boldsymbol{q}_t)$ for dealing with attack aiming to harm throughput, can be found by substituting connectivity cost $\bar{\lambda}_i$ with total remaining throughput expressed in equation (6.7) into matrix (6.8).

The authority wants to maximize the jammer's attacking time in order to force the jammer to make his attack more expensive. Whereas, the jammer wants to minimize it. The authority does not know what category of the attack the jammer intends to follow. The jammer does not know versus what category of the attack the authority intends to build up his defense. Thus, the rival faces with a dilemma of choosing the proper strategies. This dilemma can be described by the following zero-sum 2×2 matrix game

$$D = \begin{array}{c c} & \begin{array}{cc} \quad c & \quad\quad t \end{array} \\ \begin{array}{c} c \\ t \end{array} & \begin{pmatrix} T(\boldsymbol{p}_c, \boldsymbol{q}_c) & T(\boldsymbol{p}_c, \boldsymbol{q}_t) \\ T(\boldsymbol{p}_t, \boldsymbol{q}_c) & T(\boldsymbol{p}_t, \boldsymbol{q}_t) \end{pmatrix} \end{array},$$

where rows correspond to the authority's strategies, i.e., choosing attack's category to response, and columns correspond to the jammer's strategies, i.e., choosing attack's category.

This matrix game has an equilibrium (see [127]) either in pure strategies, that is, when the rival selects a specific one, or in mixed strategies, when the rival randomizes his selection. Since the game is zero-sum, then the authority's equilibrium strategy is also his maxmin strategy, i.e., it is the best response strategy for the most dangerous adversary's attack. This result is given in the following two propositions.

Proposition 1. *The game has an equilibrium in (pure) strategies if and only if the following conditions hold:*

1. *If $\boldsymbol{p}_t^T \boldsymbol{q}_c \leq \boldsymbol{p}_c^T \boldsymbol{q}_c \leq \boldsymbol{p}_c^T \boldsymbol{q}_t$ then (c, c) is an equilibrium,*

2. *If $\boldsymbol{p}_c^T \boldsymbol{q}_c \leq p_t^T q_c \leq \boldsymbol{p}_t^T \boldsymbol{q}_t$ then (t, c) is an equilibrium,*

3. *If $\boldsymbol{p}_t^T \boldsymbol{q}_t \leq \boldsymbol{p}_c^T \boldsymbol{q}_t \leq \boldsymbol{p}_c^T \boldsymbol{q}_c$ then (c, t) is an equilibrium,*

4. *If $\boldsymbol{p}_c^T \boldsymbol{q}_t \leq \boldsymbol{p}_t^T \boldsymbol{q}_t \leq \boldsymbol{p}_t^T \boldsymbol{q}_c$ then (t, t) is an equilibrium.*

Proposition 2. *If there is no equilibrium in pure strategies, the rival applies the randomized strategies. Namely, with probability, x_c (x_t), the authority should defend against "c" ("t") attack's category, and with probability, y_c (y_t), the jammer applies strategy corresponding to* *"c" ("t") attack's category, where*

$$\begin{aligned}
x_c &= \frac{T(\boldsymbol{p}_t, \boldsymbol{q}_t) - T(\boldsymbol{p}_t, \boldsymbol{q}_c)}{T(\boldsymbol{p}_c, \boldsymbol{q}_c) + T(\boldsymbol{p}_t, \boldsymbol{q}_t) - T(\boldsymbol{p}_c, \boldsymbol{q}_t) - T(\boldsymbol{p}_t, \boldsymbol{q}_c)}, \\
x_t &= 1 - x_c, \\
y_c &= \frac{T(\boldsymbol{p}_t, \boldsymbol{q}_t) - T(\boldsymbol{p}_c, \boldsymbol{q}_t)}{T(\boldsymbol{p}_c, \boldsymbol{q}_c) + T(\boldsymbol{p}_t, \boldsymbol{q}_t) - T(\boldsymbol{p}_c, \boldsymbol{q}_t) - T(\boldsymbol{p}_t, \boldsymbol{q}_c)}, \\
y_t &= 1 - y_c.
\end{aligned} \tag{6.39}$$

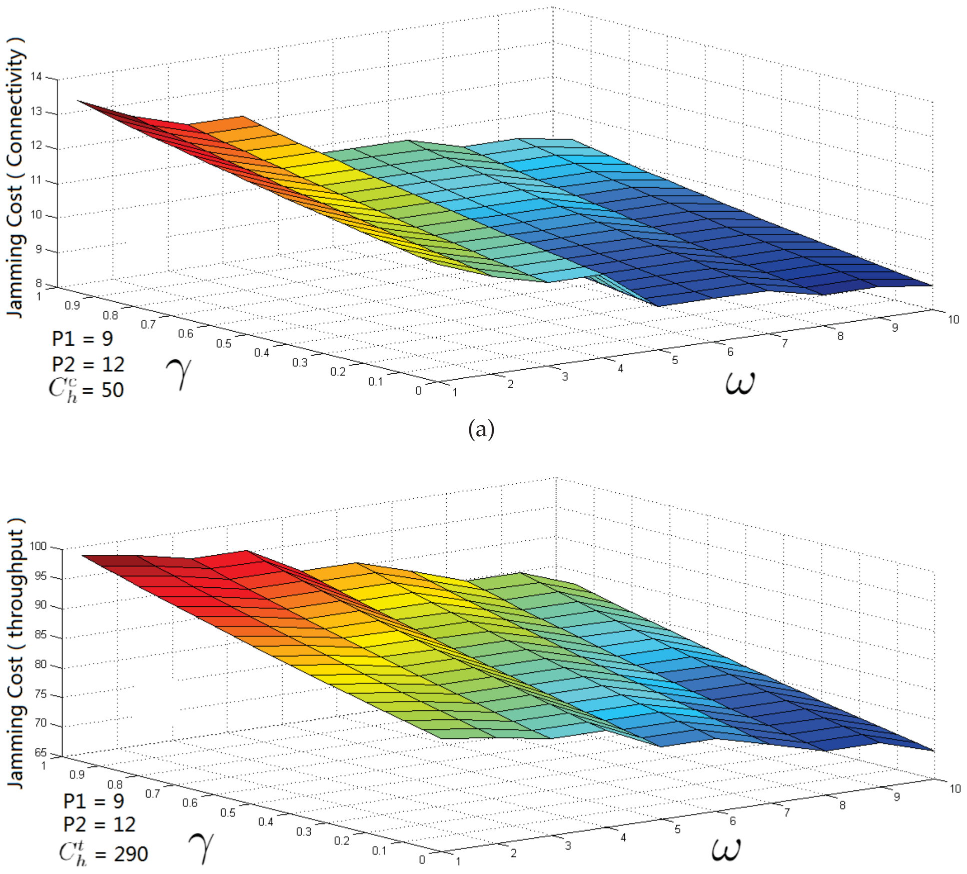
6.7 Simulation

In this section, numerical results are given to illustrate the impact of network parameters, such as transmission power of nodes and SINR's threshold, on maintaining the communication links. In simulation setting, the network consists of six nodes, i.e., $n = 6$ with background noise σ^2 equals to one. The authority scans the network to prevent malicious activity, and he does not participate in packet transmissions. The channel gain matrix, $\boldsymbol{h}$, are randomly generated and given as follows:

$$\boldsymbol{h} = \begin{bmatrix}
0 & 0.3128 & 1.1790 & 1.6488 & 1.6335 & 0.8458 \\
0.3128 & 0 & 0.4524 & 1.9653 & 0.5215 & 0.1885 \\
1.1790 & 0.4524 & 0 & 1.4605 & 1.1887 & 1.1970 \\
1.6488 & 1.9653 & 1.4605 & 0 & 0.0450 & 0.9418 \\
1.6355 & 0.5215 & 1.1887 & 0.0450 & 0 & 1.3919 \\
0.8458 & 0.1885 & 1.1970 & 0.9418 & 1.3919 & 0
\end{bmatrix}.$$

For each node as a transmission protocol, we consider the uniform power allocation that transmits the same power signals to its neighbors, but each node, of course, can have different total power levels. Here we consider that transmission powers ($P1$ and $P2$) of node 1 and 2 vary from zero to twenty. Transmission power of node 3, $P3$, is eleven. $P4$ is ten. $P5$ is nine. $P6$ is eight. Note that, the protocol of uniformly allocating transmission power is proved to be optimal for independent and identically distributed Gaussian channels [128,129]. Thus, given $\boldsymbol{h}$, P, σ^2 and receiving threshold, they already define a network's topology.

Figure 6.4 illustrates the accumulated connectivity and throughput costs, i.e., the value V_γ of the game G_γ, as function of receiver's SINR threshold, ω, and probability



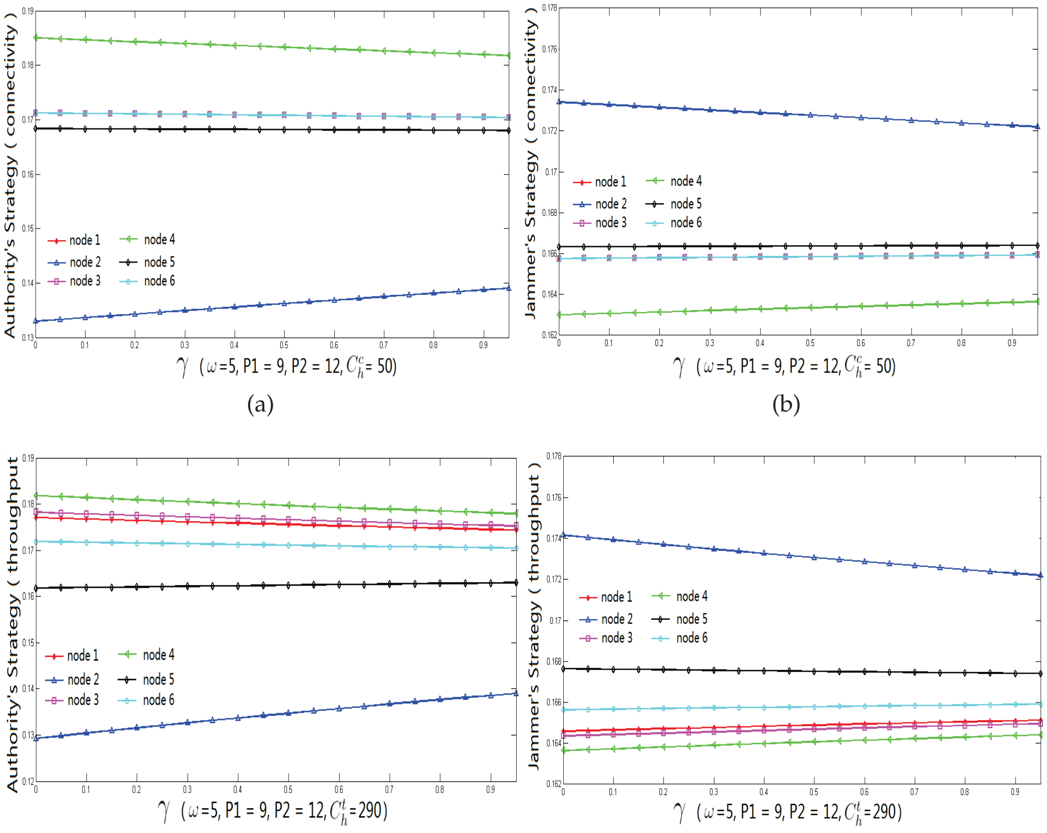
(a)

(b)

Figure 6.4: Stochastic game: accumulated connectivity (up) and throughput (down) costs as function of receiver's SINR threshold, ω, and probability of re-playing the game, γ.

of re-playing the game, γ. It shows that the accumulated jamming cost decreases with increasing threshold since the larger value of threshold assumes smaller effort to break communication links, and so smaller efforts to harm networks' connectivity. Also, increasing probability of re-playing the game, γ, yields in raising the value of the game, and so the jamming cost since longer duration of the game calls for the growth on the accumulated hiding cost. The smallest jamming cost is achieved when $\gamma = 0$, which is when the jammer can manage to perform only one time shot attack.

Figure 6.5 illustrates the authority's and the jammer's strategies for connectivity and throughput jamming game as function of probability for the game are continuous. It shows how the jammer tries to avoid being detected by the authority, in order to perform a successful attack.



(a)

(b)

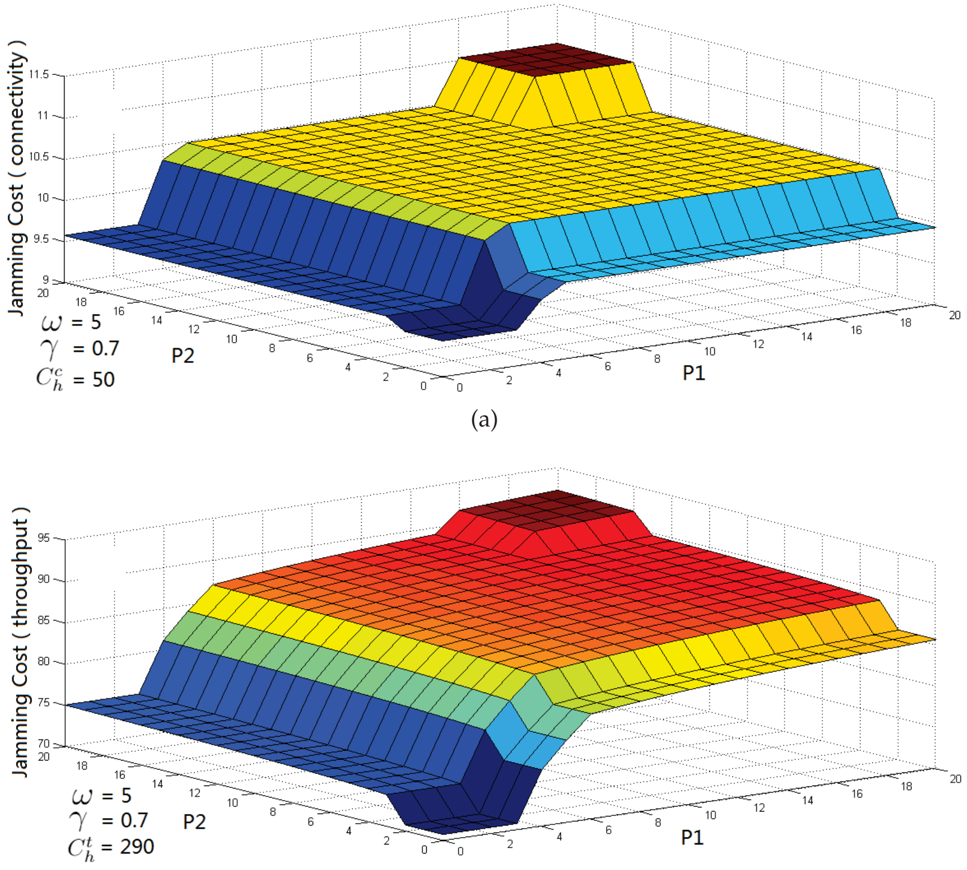
(c)

(d)

Figure 6.5: Stochastic game: (a) the authority's strategy for connectivity game, (b) the jammer's strategy for connectivity game, (c) the authority strategy for throughput jamming game and (d) the jammer's strategy for throughput jamming game as functions of probability for the game to be continued, γ.

Figure 6.6 illustrates the relations between the jamming cost and transmission power. These relations are piece-wise continuous with jumps happen in between. Increasing transmission power causes adding new communication links into the network topology. This yields into increasing Fiedler value and throughput by jumps. Thus, it produces the leap in the value of the game. Since the interference is not considered in the simulation for showing an obvious tendency without fluctuation, the value of connectivity game is piece-wise constant on transmission power while the value of throughput jamming game is piece-wise continuous.

Figure 6.7a illustrates the probability that the authority intends to deal with an attack aiming to disrupt connectivity as function of transmission power of node 2. It shows that bigger probability γ assumes smaller transmission power in order to switch



(a)

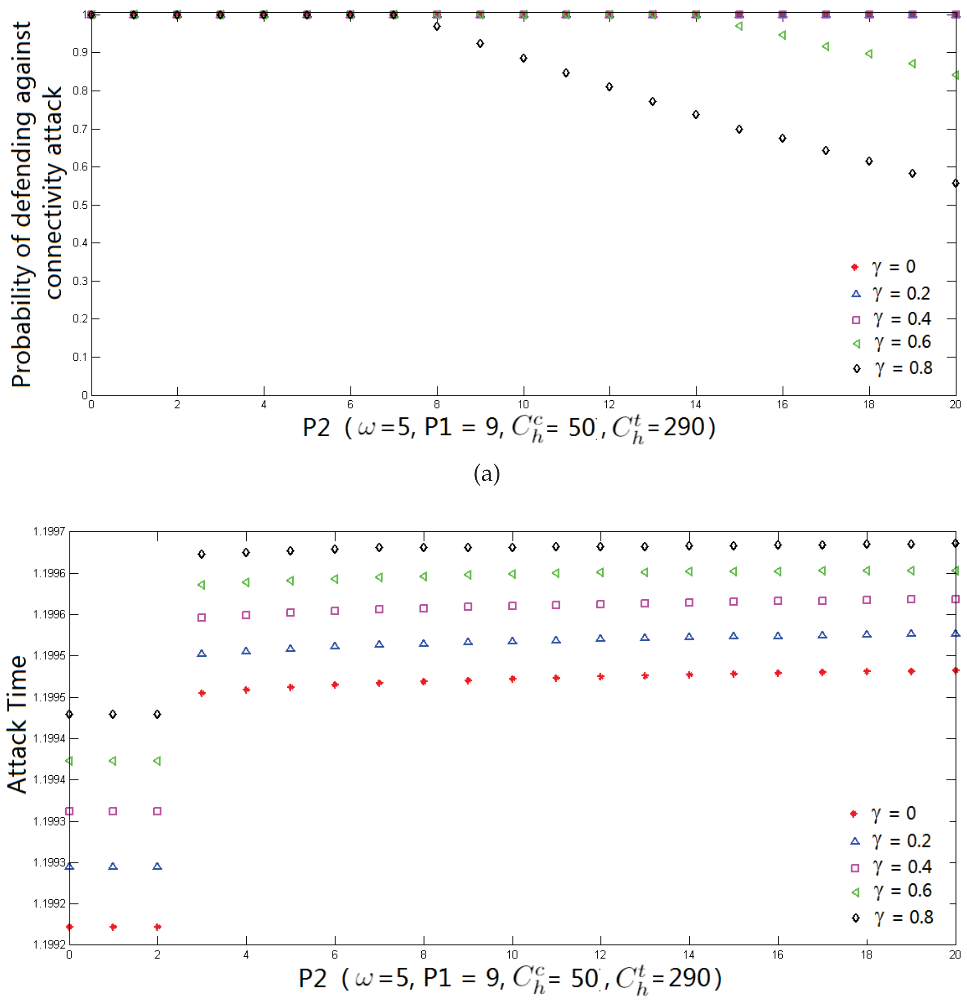
(b)

Figure 6.6: Stochastic game: accumulated connectivity (up) and throughput (down) costs as functions of transmission powers of node 1 and node 2.

to mixed strategy. Also, the authority's strategy on maintaining connectivity is non-increasing on the transmission power. Figure 6.7b illustrates the expected duration of the game as function on transmission power of node 2. On the same reason as one for the value of the game, namely, changing in the network's topology due to adding new links, the expected duration of the game are piece-wise continuous function on the transmission power.

6.8 Conclusions

In this chapter, costs for jamming attack on connectivity and throughput are introduced. Then, a uniform stochastic game with network scanning to prevent jamming attack is suggested and solved explicitly. Due to different aims, the jammer might



(a)

(b)

Figure 6.7: Maxmin selection of scanning strategy: (a) the probability that the authority intends to deal with attack aiming to disrupt connectivity, and (b) the expected durations of the game as function on transmission power of node 2.

choose different categories of attacks, for example, connectivity and throughput. Comparing such incomparable attacks is a challenge. To deal with this issue, an approach is suggested to compare them by duration of attack instead of the damage they bring to. Game theoretical model for such comparison is suggested, and optimal strategies are proposed. Results for numerical evaluation of the optimal solutions and their dependence on network's characteristics are supplied. A goal of our future work is to investigate more sophisticate behaviour of the adversary where he can switch between different categories of malicious activities based on the archived result of attack, and to incorporate mechanism of learning in the authority strategy based on the accumulated results of scanning.

Game theory has provided a solution to make best decision when topology changes dynamically. It has also shown its effectiveness on describing these interactions between agent and environment outside. This effectiveness guides us a possible direction to study the network connectivity and robustness by artificial intelligence which currently gains its fast and big improvement on machine learning in both industry and academy. Thus, the next chapter investigates the relationship between network connectivity and artificial intelligence, and illustrates the possible directions of applying it to improving the connectivity. Some of my works on these two areas are presented. Moreover, the network referred in the next chapter is expended to any type of network that is connected through wireless medium such as sensor networks, Wifi, vehicular networks, and even the MIMO networks. The aim is to show the advancement of this area and confirm my possible future research.

Chapter 7

Improving Connectivity by Artificial Intelligence

Maintaining the connectivity of an ad hoc network through the use of artificial intelligence is a relative new topic. It requires combining several research areas such as distributed control theory, designing network protocol, measuring topology connectivity, learning connectivity information in big data, and making optimal decision in face of uncertain forces and disturbances. However, the awkward situation arises when engineering researchers who have the knowledge of network protocol are not familiar with the operation mechanism of artificial intelligence which belongs to science discipline. Due to this unfamiliarity, most artificial tools used in network area do not yield promising results. Thus, they delay the application of artificial intelligence to network area. Whereas, scientists focus their efforts on algorithm itself, and have no interests or knowledges to apply their algorithm to complicated engineering tasks, especially to the network area which requires in-depth professional training. Thus, there is a disconnect between network engineering and artificial intelligence. Although many researches and companies have paid attention to this gap, the application of artificial intelligence only regards network or cloud as a database and transmission tool or a black box through which to obtain data from other applications. As a result, network to these applications is transparent, and the design process no longer needs to consider the network protocol and architecture. Thus, there is no real need of application of artificial intelligence to network design.

Indeed, an ad hoc network itself contains sizable data which scales with the number of nodes. These data exist in every level of network protocol stack, and it describes the wellness of network operation. Data science as a part of Artificial intelligence are in favor of these data. They are good resources for artificial intelligence to diagnose the operation of network, and to tailor its control strategy or transmission protocol to adapt to changing environment to maintain network connectivity. For example, data from routing layer can be utilized to recognize topology of ad hoc network which dynamically changes in every second. The topology information can assist in directing node mobility or placing a new node to enhance the bottleneck of connectivity in an ad hoc network. Another example is the data from application layer can locate a part of an ad hoc network which needs transmitting large bundles of data or where a network component is congested, and alleviate traffic congestion by either redistributing data copy, or adding several intermediate nodes.

Besides topology and application data produced in each level of ad hoc networks, the control policy in reference to these data through inter-communicating information by protocols are also important, but yet being investigated sufficiently. These control strategies should combine both classical control mechanism and adaptive ability to well adjust to environment changes while maintaining the stability and reliability requirements to achieve certain functions. It should have the function of predicting a potential disorder or disastrous network components so that a preventive measure can be executed in advance. It can also intelligently monitor threats, then take corresponding cognitive decision without human intervention. Moreover, it should be able to self-recovery should a failure occurs. Under normal operation, in a power limited ad hoc networks, the evolution of relationship to compete and to collaborate among nodes should adapt environmental constraints so that every node can survive longer to achieve its goals. However, only a few of present literatures compares the control policy designed by artificial intelligence with conventional network control protocol. Thus, the current improvement of artificial intelligence on control is not obvious. Also the traditional control parameter is either pre-configured, or manually set during policy making operation.

The artificial intelligence area contains both data mining, and an optimal decision making part, and they complement each other. Currently, most researches in industry and academy involve executing data mining part, this is true when deep learning has successfully proven its superior advantage over human's mind after winning the go match in 2016. However, optimal decision making based on learned pattern, or even in the process of learning is seldomly considered in the literature. Thus, learning while deciding strategies is critical research area, and will promote the progress of industry.

I am always trying to put myself in the way of applying artificial intelligence to different network scenarios. However, it is just in the primary stage. Chapter 3 uses reinforcement learning technique in artificial intelligence to learn the possible location of a jammer based on packet loss data. Chapter 6 utilizes the game theory to make optimal strategy to maintain network connectivity under repeated jamming attacks. Another work [95] of mine combines both online data learning and optimal decision making in game theory to learn different types of a jamming attack. My work during internship in Mitsubishi Electric Research Laboratories is using the reinforcement learning to find optimal channel allocation to avoid interferences in heterogeneous networks. Another research conducted during internship at Fujitsu Laboratories of America involves with the adaption of multi agent distributed reinforcement leaning to transportation system to control traffic signals in a vehicular network.

There are various methods to maintain connectivity of an ad hoc network in the levels of both direct-connected points and multi-hop path. Normally artificial intelligence method is used to find an optimal and congestion-less route to reliably deliver a packet, such as Q-learning, swarm intelligence, genetic algorithm [130] and fuzzy logics [131]. In the section below, I listed other possible tools to achieve different connectivity objectives besides routing.

7.1 Multi-agent distributed reinforcement learning

Reinforcement learning well suits the requirement of decision making under dynamic characteristics of an ad hoc network described in chapter 1. The mobility of nodes and wireless transmission medium cause uncertainty of decision making. The strategy that currently works may no longer work in the next minute. A device has no specific rules to follow. Thus, a node equipped with reinforcement learning ability must make decisions by learning other behaviors of nodes and the change of natural environment. The optimal decision is reinforced during the learning process. Reinforcement learning is usually called Markov process. A node takes an action, then it observes the impact of actions on environment by receiving a reward from it. A correct action gives a positive reward while a wrong action gives a negative reward. The less benefit action gives a smaller value of reward. The ultimate goal is to achieve a cumulated maximum of rewards in long run despite of taking single action that may not be optimal in one instance.

The distributed characteristic of an ad hoc network requires reinforcement learning running in each node in a distributed fashion. This phenomenon has multiple reinforcement learning that is operating simultaneously. It is called multi-agent reinforcement learning. The node in multi-agent reinforcement is an agent that can both learn the environment and make an optimal decision. An ad hoc network can be viewed as a network composed of multiple agents described above.

The cooperation among multi-agents is as important as the cooperation among its ad hoc nodes. Thus, designing a multi-agent reinforcement learning must consider other agents' policies in order to achieve a common goal or to agree on a consistent group behavior. If an agent only learns the environment without considering other nodes' strategies, it may choose a selfish, and even disastrous policy for the network. Although in the short run, it gets a huge reward. However, since it cannot survive without others' assistance, its temporary reward does not go far in the long run.

Integrating collaboration in the process of making a policy in reinforcement learning can form in possible two ways:

- One is to design a cooperative reward function in Q learning so that increasing rewards of others results in increasing the benefit of itself or take a group's common profit into consideration by contributing positively proportional benefits to the agent itself. As a consequence, an agent tends to select an action that can increase the benefits of others' and the group.

- The second way is to consider other nodes' strategy as a state in the Markov process. The transition probability leads to bad states of other agents is small.

7.1.1 Case study: adaptive distributed network control

Vehicular network is a classical representation of a distributed network which is composed of vehicles, road-side infrastructure, and traffic lights, etc.. Traffic lights control mainly deals with vehicular traffic. The performance of control influences the efficiency or connectivity of whole transportation system. More efficient control system causes less traffic jam, reduces wait time at junctions while drivers uses less gas and generates less pollution to the environment. A congestion on the road can be seen as a broken link in a graph topology.

These performance metrics can be used as an instantaneous cost in reinforcement learning to guide the traffic light control. To improve system efficiency, it calculates optimal green time by learning traffic data observed at junctions. To achieve global optimality, a traffic light should consider the traffic conditions of neighboring junctions so that it restricts the number of cars entering congestion junctions. To achieve this, an interaction between neighboring nodes must be executed. Thus, the neighboring traffic is taken into consideration based on junction's local cost, and can be expressed as follows:

$$C^t_{i,d}(a^t_{i,d}, a^t_{j,d}, S^t_{i,d}, S^t_{j,d}, W^t_{i,d}) = \frac{1}{|N_i|}(w^t_{1,d} \sum_{j \epsilon N_i} q^t_{ji,d} + w^t_{2,d} \frac{1}{|N_j|} \sum_{j \epsilon N_i} \sum_{k \epsilon N_j} q^t_{kj,d} + \frac{w^t_{3,d}}{2} \sum_{j \epsilon N_i} (m^t_{ji,d,L} + m^t_{ji,d,R})) \tag{7.1}$$

Were $\sum_{j \epsilon N_i} q^t_{ji,d}$ corresponds to queue length of incoming traffic. $\sum_{j \epsilon N_i} \sum_{k \epsilon N_j} q^t_{kj,d}$ corresponds to all vehicular queue lengths waiting at the neighboring junctions, and this item is always non-negative. Thus, if the junction releases many cars to neighboring junctions, the cost function (7.1) will increase. $\sum_{k \epsilon N_j} q^t_{kj,d} + \frac{w^t_{3,d}}{2} \sum_{j \epsilon N_i} (m^t_{ji,d,L} + m^t_{ji,d,R}))$ is pedestrians' waiting queue at the junction.

The multi-agent reinforcement learning is trying to minimize the total queue lengths at each intersection in the network. Therefore, if the traffic light releases too many vehicles to neighboring junctions, it will cause neighboring congestion. As a result, its own cost will increase. Then, less likely, this action should be chosen in the future.

This multi-agent reinforcement learning is implemented in SUMO, a transportation simulator, that can model the urban intermodal traffic systems with small granularity designed by Institute of Transportation Systems in Germany. In Fig.7.1, the

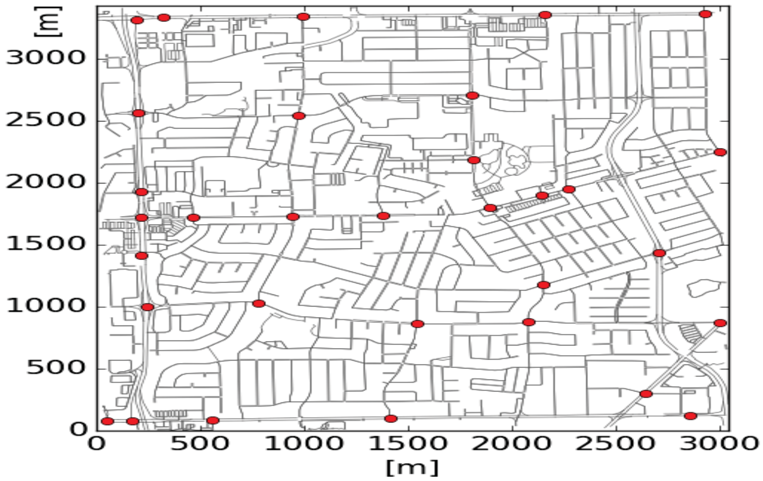


Figure 7.1: Intersections that implement multi-agent reinforcement learning

	vehicular queue length	pedestrian queue length	vehicular wait time(s)	vehicular travel time(s)	pedestrian wait time(s)	pedestrian travel time(s)
multi-agent	256202	22844	85.3054	327.0856	28.4818	1852.5433
4 actions (fixed)	902186	50501	205.4251	480.4376	64.4948	1874.2610
4 actions (sensor)	812445	28091	201.3827	468.8852	36.4066	1851.1138
6 actions (sensor)	667710	33943	136.3300	385.5124	43.0872	1858.4526
8 actions (sensor)	605530	28177	136.7952	390.1535	35.4368	1852.6488

Table 7.1: Multi-agent reinforcement learning in traffic light control system

number of reinforcement learning agents is 32. Each agent is implemented at a junction. Multi-agent reinforcement learning is compared with four traffic light control strategies implemented in real world: 1) fixed timing control where the time interval of green light, and the order of red, green, yellow lights are fixed; 2) sensor triggering cases where a green light changes to red when a sensor detects no traffic in the direction of green lights. The order of red, green, yellow lights follows three state transition diagrams currently implemented at junctions. Each diagram has different number of actions. So I divide it into three cases: a) four-action sensor triggering case; b) six-action sensor triggering case; c) eight-action senor triggering case. Table 7.1 shows the overall results that multi-agent approaches performs the best.

7.2 Topology analysis with machine learning

Studying the topology of an ad hoc network is important, but also a complicated task since the topology is changing with unclear rules. Thus, studying the dynamic process of growing network is also complex, and no single enclosed formula can express the regulation of network growth. Thus, to simplify the analysis of topology, especially for a large scale network, the failure of direct connection between two points has limited influence to whole network's function, thus is usually ignored.

Clustering algorithm is normally applied to analyze the backbone structure of large scale networks. Network nodes with similar features and characteristics are grouped together, and perform the same local function. A center of a cluster is considered as a gateway or representative of nodes inside a cluster to exchange information with other cluster centers. These cluster centers are considered as the critical nodes or bottlenecks to network connectivity or control topology.

The conventional clustering algorithm measures the euclidean distances between data points, and iteratively updates the cluster center if a new datum is added into a cluster. The euclidean distances is a mathematical term. Network can map any physical performance metrics to the distances, such as connectivity, QoS, security, congestion status and power consumption. When fetching the data, the application in upper layer does not need to know exactly every node connection in the network. By performing clustering, it identifies which cluster center has the most data that can be consulted to.

Neural networks can be used to calculate the weights incident to the connection of two neurons. This weight can either be zero, or one to indicate whether to trigger a neuron. A neuron in neural network can be considered as a node in an ad hoc network. The values of weights result in different network structures. Every structure corresponds to a specific function. Thus, by changing the neuron network topology, various network function is achieved.

Evolutionary algorithm can be used to study the survival of network nodes when several parts of network connectivity are destroyed by one or multiple malicious nodes

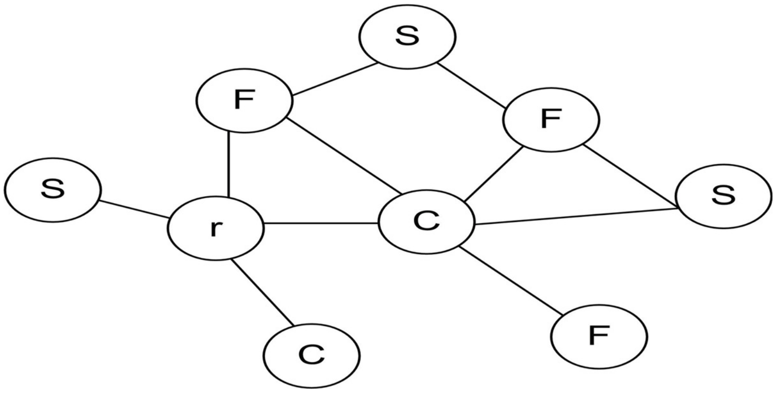


Figure 7.2: connectivity evolution in different types of node

such as malicious nodes which isolate legitimate nodes from a network, or a few nodes which broadcast and propagate wrong routing information to interrupt the normal operation (which we call the Byzantine attack). The node that broadcasts wrong information is allocated with a low reputation. Packets are routed through these low reputation nodes with small probability. This probability is positively proportional to the reputation.

For example, culture evolution can be used to model the cooperation of node population on a commonly agreed culture, and study the impact of the culture evolution through cooperation on other network population/clusters. Coevolution can be used to design a survival strategy for a node's population in a network to adapt to changes occurred in other population through cooperation and competition. Fig.7.2 illustrates a problem how network connectivity survives longer with different types of nodes involved in a network, and is able to achieve a common goal. *"C"* represents a cooperative node. *"S"* represents a selfish node. *"r"* represents a random node which can either be a cooperative node or to transform itself to a selfish node. The rule of transformation can either be specified, or unspecified. *"F"* is a follower. It follows other nodes with his belief.

7.3 Learning dynamic data of a network by game theory

Fictitious play, repeated and stochastic game are three categories in game theory that mostly relate to artificial intelligence by learning the opponents' data with time axis. A portfolio of belief on opponents' strategies is established and updated in the feedback of learning process. It is used as a future reference to behave rationally in future. The ultimate mixed Nash Equilibrium is calculated in the sense of time average. Fictitious play can solve transformed repeated and stochastic games. It converges in a time-average sense if the game is a zero sum game or all players have the same pay-off function which complies with the definition of an ad hoc network where every node performs identical work.

7.3.1 Case study: pilot transmission strategy

Fictitious play can be used to control transmission strategy to increase network connectivity. The following research is one project that uses fictitious play to design an optimal pilot transmission strategy when natural noise exists. This pilot transmission game is formulated in a stochastic game matrix and described in details as follows.

If terminals send only data signals in good channel condition, the uplink throughput BS can obtain is $C_{\bar{p}}^{G}$, which is the instantaneous payoff in current time interval. The transmission is re-played with a discounted factor, γ_G, for good channel condition. If data signals are sent in bad channel condition without pilots, then their transmission incurs severe interference, which results in the throughput, $C_{\bar{p}}^{B}$, to decreases. The BS tends to request more frequently for the terminals to re-send correct messages. Thus, the re-transmission frequencies, or discounted factor, γ_B, is greater than γ_G. If terminals add pilots to data transmission in good channel condition, the instantaneous uplink throughput is C_{p}^{G}, and new data are sent in next coherent time with discounted factor γ_G. If terminals send pilots with data in bad channel conditions, the transmission system gets two payoffs: one is the throughput with pilots under large interference. The other is the gain of using pilots in bad channel which encourages terminals to use pilot to cancel severe interference. As previously discussed, the game is re-played with discounted factor γ_B.

The above pilot transmission protocol is formulated as a stochastic game in the following format:

$$\Gamma: \begin{array}{c|cc} & G & B \\ \hline \bar{p} & C^G_{\bar{p}} + \gamma_G \Gamma & C^B_{\bar{p}} + \gamma_B \Gamma \\ p & C^G_p + \gamma_G \Gamma & C^B_p + \eta(C^B_p - C^B_{\bar{p}}) + \gamma_B \Gamma \end{array}, \tag{7.2}$$

Where G and B represent a good and bad channel condition, respectively p and $\bar{p}$ represent to transmit or not to transmit pilots. The theoretical result is shown in appendix 7.2.

A fictitious play combined with online Weighted Multiplicative Weights method is used to learn the pilot transmission strategy. A-optimality is used to judge the channel condition as below:

$$A_1 = tr\left\{(\mathbf{H}^H\mathbf{H})^{-1}\right\} \tag{7.3}$$

Algorithm 6 provides the pseudo code of learning the pilot transmission strategy. Where D is decoding matrix. H is the channel matrix. Γ is the game value. μ is the learning rate.

Proposition 7.3.1. *: The average regret made by the learning Algorithm 6, for pilot transmission, is bounded by*

$$-\frac{1}{2} \leq \frac{1}{T}\sum_{t=1}^{T}\frac{\mathbf{p}^t\mathbf{e}^t}{m^t} - \frac{1}{T}\sum_{t=1}^{T}\frac{e_i^t}{m^t} \leq 2\sqrt{\frac{ln2}{T}}$$

with $\xi = \min\left\{\sqrt{\frac{ln2}{T}}, \frac{1}{2}\right\}$*, where* $\mathbf{e} = [e_1^t\ e_2^t]$*,* $e_i^t = c_i^t - \tilde{c}^t$*, and* $m^t = \max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t$*.*

Algorithm 6 is validated in fixed instantaneous rewards. For every time interval, we sampled the channel gain matrix, H, from Gaussian process, $\mathcal{CN}(\mathbf{0}, \mathbf{I})$ to decide the probability for good and bad channels. The game matrix for our validation is $\Gamma = \begin{pmatrix} 4 + r_G\Gamma & 1 + r_B\Gamma \\ 3 + r_G\Gamma & 2 + r_B\Gamma \end{pmatrix}$. r_G is set to be 0.5. r_B is set to be 0.7. We compare the learned game results which vary with good channel probability that is 1, 0.7, 0.5, 0.3, 0. Fig. 7.3 demonstrates the learned game value under stationary channel conditions. It reveals that the spectrum efficiency stabilizes after 150 iterations. According to the theoretical solution of game 7.2, Nash equilibrium happens as the probability of good

Algorithm 6 Learning pilot transmission strategy

```
  Initialize: a stochastic process,$X_h^t$, for channel model and threshold, $r_h$, for judging channel conditions (good or bad), and fix learning rates $\zeta \le \frac{1}{2}$, $\mu < 1$. Set the first decoding matrix $D^0$ to be a unit matrix. Assume $w_i^1 = 1$, for all $i = 1, 2$. Set $n_G^0 = 0$, $n_B^0 = 0$, and game value, $val^1(\Gamma) = 0$
  for t = 1, 2, ... , T do
    1. Choose decision, $l$, from strategy set, $L = \{1, 2\}$, according to probability $\mathbf{p}^t = \left[p_1^t,\ p_2^t\right]$, proportional to the weights $w_i^t$ where $p_1^t = \frac{w_1^t}{w_1^t + w_2^t}$, $p_2^t = 1 - p_1^t$
    2. Generate $M$ by $K$ samples from $X_h$ and compose $\mathbf{H}^t$ by these samples
    3. Update decoding matrix
        If $l = 1$, $D^t = D^{t-1}$
        If $l = 2$, $D^t = H^t$ for $MRC$ decoder
    4. Calculate $A_i^t$ criterion according to (7.3)
    5. Set the channel condition indicator:
        $x^t = \begin{cases} 1 & if A_i^t \le r_h,\ good\ \mathbf{H}^t \\ 0 & if A_i^t > r_h,\ bad\ \mathbf{H}^t \end{cases}$
        if $x^t = 1$,  $n_G^t = n_G^{t-1} + 1$
        if $x^t = 0$,  $n_B^t = n_B^{t-1} + 1$
    then the empirical distribution of $\mathbf{q}^t = \left[q_G^t,\ q_B^t\right]$ is derived by $q_G^t = \frac{n_G^t}{n_G^t + n_B^t}$, $q_B^t = 1 - q_G^t$
    6. Compute the theoretical value of spectrum efficiency
        $\tilde{v}_1^t = \Gamma_{11}(val^t(\Gamma))q_G^t + \Gamma_{12}(val^t(\Gamma))q_B^t$
        $\tilde{v}_2^t = \Gamma_{21}(val^t(\Gamma))q_G^t + \Gamma_{22}(val^t(\Gamma))q_B^t$
        $\tilde{c}^t = \max\left\{\tilde{v}_1^t, \tilde{v}_2^t\right\}$
    7. Observe the spectrum efficiency, $\mathbf{c}^t = \left[c_1^t,\ c_2^t\right]$, under decision $l$. For decisions that are not $l$, set their payoffs to be zeros.
        $c_l^t = \Gamma_{l1}(val^t(\Gamma))x^t + \Gamma_{l2}(val^t(\Gamma))(1 - x^t)$
        $c_{i\epsilon L\backslash l}^t = 0$
    8. Update weights
        $w_i^{t+1} = w_i^t \left[1 - \zeta \frac{c_i^t - \tilde{c}^t}{\max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t}\right]$
    9. Update the game value
        $val^{t+1}(\Gamma) = (1 - \mu)val^t(\Gamma) + \mu \max\left\{c_1^t, c_2^t\right\}$
  end for
```

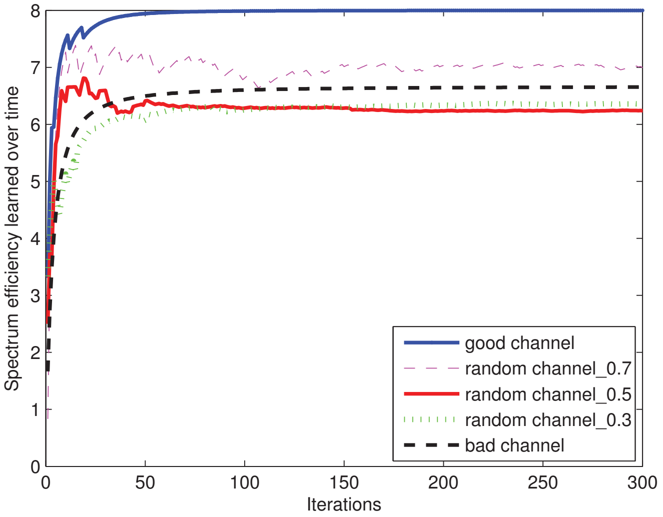


Figure 7.3: Learning game value under stationary channel

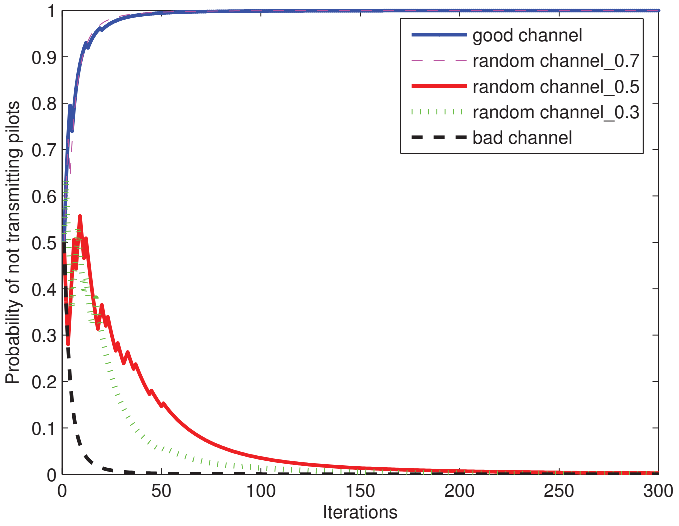


Figure 7.4: Learning pilot strategies under stationary channel

channel has a value of 0.5. Therefore the red line converges at a low value. Fig. 7.4 shows the corresponding learned probability of not transmitting pilots.

To illustrate the correctness of learning Algorithm 6, we compared the theoretical game value under fixed channel conditions, where the probability of good channel is fixed in simulation. Fig. 7.5 displays the comparison. If we denote $T^G(q)$ as theoretical values obtained for spectrum efficiency, we found $T^G(1) > T^G(0.7) > T^G(0) > T^G(0.3) > T^G(0.5)$ which complies with the learned results shown in Fig. 7.3.

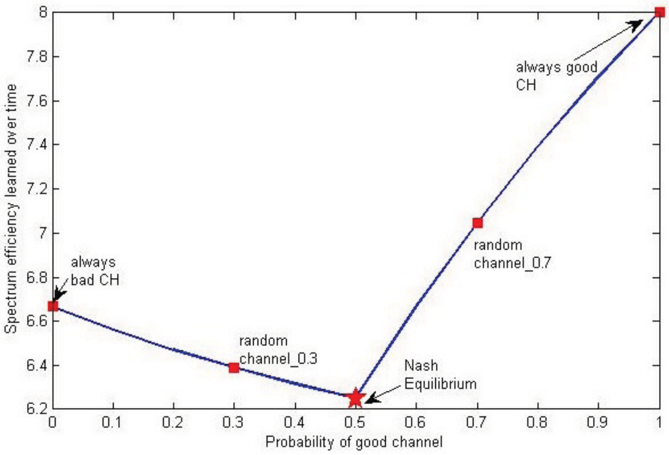


Figure 7.5: Comparing the theoretic game value with fixed channel probability

The inherent characteristics of a network require application of artificial intelligence to maintain large-scale network connectivity since its topology is dynamic which makes the interactions among components in a network more complicated. Thus, no explicit rules can model the network's dynamic process on connectivity. Artificial intelligence as a data learning tool can automatically provide rational decisions. Thus, it caters the idea of converting a complex problem into a general solution. Thus, it is very promising area for network studies.

Chapter 8

Conclusion

8.1 Summary

An ad hoc network is susceptible to disturbance, and even destruction, since it is wirelessly connected by low power devices. Thus, enabling its robustness is a very important issue. Robustness can be achieved by designing a reliable network transmission protocol, deploying nodes in optimal positions, collaborating nodes to achieve a common goal of reliability through interaction, etc.. Indeed, each of these categories is an open problem, and can be solved by many tools. This thesis provides possible solutions in the categories listed below:

1. Design transmission protocol to cope with natural noise. An overlay routing protocol is designed to comply with the underlying physical routing to solve the "backtracking problem" by using geographical location aware cluster heads. Thus, the receiving error rate is decreased by reducing the length of transmission path and the number of re-forwarding messages at each node;

2. Design transmission protocol to cope with attacks by adversaries. 1) Localize jammer's position by learning the network performance data in a distributed way. This algorithm has light-weighted characteristics and does not involve additional control messages. After pin pointing the location, a defense strategy can be deployed to the exact spot. 2) Adaptive cross-layer power allocation to achieve robust network connectivity. It is a light-weighted algorithm which physical layer recognizes the network topology by the OLSR routing;

3. Understand the connectivity and achievable throughput while under interference. The concept of throughput connectivity is proposed to guide the allocation of physical resources to achieve network robustness;
4. Understand the dynamic change of connectivity with repetitive jamming attack by game theory. The jammer causes the worst form of damage to network connectivity in an ad hoc network. A defense strategy is proposed to limit the attacker's destruction;
5. Illustrate possible tools with application of artificial intelligence to cope with network connectivity attacks. The way of integrating these tools is introduced with examples given.

8.2 Future direction

Most artificial intelligence scientists focus on improving the accuracy of machine learning results, mainly on algorithm designs. For example, they are trying to improve every aspect of deep learning by various methods such as replacing the 2nd generation of neural network to the 3rd generation one in order to reduce energy consumption caused by heavy computation without undermining its accuracy. However, the emphasis on well integrating artificial intelligence into specific engineering problems with classical expert knowledge of this area is less investigated due to various reasons, and they are discussed at the beginning of the chapter 7.

Applying artificial intelligence for the analysis of connectivity for network robustness presents a promising picture, and leads me a direction for my future research. Therefore, my research project evolves around in four aspects mentioned below:

1. Study the different corresponding optimal network architecture for achieving distinct global network functions. The possible tools to achieve this goal are:
   (a) data mining which explores the route of information flow, and builds the application on QoS required network architecture, then designs a network topology to support demand and supply requirements;

   (b) use artificial neural network for modeling and mining the network architecture by calculating and adjusting the weights of each link to horizontally and vertically explore and achieve possible complex network function. Network architecture is regarded as a black box composed of several connected basic entities. It can achieve basic functions, such as sigmoid function to achieve judgment ability and wavelets function to describe the significance of one component;

   (c) Applying a combination of machine learning with information theory and communication formulas to assist fitting data to proper communication model. By doing so, a network architecture complies more with communication mechanism.

2. Study the interaction among network entities, such as evolution of network connectivity through cooperation and competition of its members. Thus, possible questions involved are as follows: How one component or a cluster's culture or changes can impact rest of network? How to control impacts such as restricting a bad impact within a small area and promoting a good impact to broader scope? How to achieve consensus by appropriately leading and guiding interactions? How to investigate the impact of network architecture on the way and style of interaction? Many of these questions can be solved by the application of artificial intelligence, such as using evolutionary theory, belief propagation through Bayesian networks, fuzzy control and game theory;

3. Combine classical control theory with artificial intelligence to achieve network robustness by the design of a real-time adaptive feedback control system. Classical control theory has proven its effectiveness in the industry for the past few centuries. However, it has limitations on coping with uncertain and unknown changes. Artificial intelligence which bases its control to analyze real world rewards can be integrated to compensate weakness and to broaden range of system robustness. Possible tools available to achieve the effective combination is

through reinforcement learning, deep reinforcement learning, double reinforcement learning, inverse reinforcement learning, and multi-agent reinforcement learning, etc..

4. Apply artificial intelligence to network intrusion detection system (IDS). Designing a reliable IDS with high accuracy is one primary objective for network security. One of the challenges is to minimize false alarm rate in the face of continuously varying attack patterns in large quantities of data. Besides using excellent data classification tools, such as deep learning, making a correct control decision is also important. The conventional tool of decision making is to use logic control, that is, if the pattern belongs to the attack type in the data base, it will be segregated. Then, with many uncertain types of attacks not found in database, one possible tool is to combine fuzzy logic control with reinforcement learning to learn the attack uncertainty and timely adapt the control strategy. Another possible tool is to use game theory to model the attack behavior and to predict the possible attack targets with inverse reinforcement learning to learn an attacker's rewards.

There are many artificial intelligence tools available through the ideas of data mining and intelligent decision making. Not every techniques are covered in the chapter 7. A combination of possibly one, or several of them could solve most real-world problems. To stimulate the best potentials of artificial intelligence, knowing the engineering working mechanism along with well integrating it with artificial intelligence is critical for achieving satisfying control results.

References

[1] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *Communications Magazine, IEEE*, vol. 46, no. 6, pp. 164–171, 2008.

[2] D. B. J. D. A. Maltz and J. Broch, "Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks," *Computer Science Department Carnegie Mellon University Pittsburgh, PA*, pp. 15 213–3891, 2001.

[3] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*. IEEE, 1999, pp. 90–100.

[4] T.-W. Chen and M. Gerla, "Global state routing: A new routing scheme for ad-hoc wireless networks," in *Proc. IEEE International Conference on Communications (ICC)*, vol. 1. IEEE, 1998, pp. 171–175.

[5] J. N. Al-Karaki and A. E. Kamal, "Efficient virtual-backbone routing in mobile ad hoc networks," *Computer Networks*, vol. 52, no. 2, pp. 327–350, 2008.

[6] "Wireless ad hoc network," http://en.wikipedia.org/wiki/Wireless_ad_hoc_network, wikipedia.

[7] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad hoc networks*, vol. 1, no. 1, pp. 13–64, 2003.

[8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, 2001.

[9] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2005, pp. 46–57.

[10] H. S. Wang and N. Moayeri, "Finite-state markov channel-a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, 1995.

[11] J.-P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, and F. Frederiksen, "A stochastic mimo radio channel model with experimental validation," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1211–1226, 2002.

[12] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky, and W. Papke, "The land mobile satellite communication channel-recording, statistics, and channel model," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 2, pp. 375–386, 1991.

[13] A. F. Molisch, K. Balakrishnan, D. Cassioli, C.-C. Chong, S. Emami, A. Fort, J. Karedal, J. Kunisch, H. Schantz, U. Schuster *et al.*, "Ieee 802.15. 4a channel model-final report," *IEEE P802*, vol. 15, no. 04, p. 0662, 2004.

[14] "Gps accuracy," http://www.gps.gov/systems/gps/performance/accuracy/.

[15] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabaey, "Distributed algorithms for transmission power control in wireless sensor networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 1. IEEE, 2003, pp. 558–563.

[16] A. Garnaev, Y. Liu, and W. Trappe, "An anti-jamming strategy for a coalition versus a team of jammers in a p2p network," *Ad Hoc Networks*, 2014, submitted.

[17] ——, "A low power jamming attack against a p2p wireless network," *IEEE Communication Letters*, 2014, submitted.

[18] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "Conserving transmission power in wireless ad hoc networks," in *Network Protocols, 2001. Ninth International Conference on*. IEEE, 2001, pp. 24–34.

[19] M. Krunz, A. Muqattash, and S.-J. Lee, "Transmission power control in wireless ad hoc networks: challenges, solutions and open issues," *IEEE Network*, vol. 18, no. 5, pp. 8–14, 2004.

[20] C. Pandana and K. R. Liu, "Robust connectivity-aware energy-efficient routing for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3904–3916, 2008.

[21] X.-D. Zhang, "The laplacian eigenvalues of graphs: a survey," *arXiv preprint arXiv:1111.2897*, 2011.

[22] Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (zrp) for ad hoc networks," *draft-ietf-manet-zone-zrp-04. txt*, 2002.

[23] U. M. Fayyad, C. Reina, and P. S. Bradley, "Initialization of iterative refinement clustering algorithms." in *Proc. the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998, pp. 194–198.

[24] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *Personal Communications, IEEE*, vol. 6, no. 2, pp. 46–55, 1999.

[25] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer, "Minerva: Collaborative p2p search," in *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 1263–1266.

[26] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 1, pp. 41–53, 2004.

[27] S. Ratnasamy12, P. Francis, M. Handley, R. Karp12, and S. Shenker, "A scalable content-addressable network," *Proc. the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2001.

[28] A. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,” in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*. Springer, 2001, pp. 329–350.

[29] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot *et al.*, “Optimized link state routing protocol (olsr),” *RFC Editor*, 2003.

[30] M. C. M. Thein and T. Thein, “An energy efficient cluster-head selection for wireless sensor networks,” in *Intelligent systems, modelling and simulation (ISMS), 2010 international conference on*. IEEE, 2010, pp. 287–291.

[31] A. Thonklin and W. Suntiamorntut, “Load balanced and energy efficient cluster head election in wireless sensor networks,” in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference on*. IEEE, 2011, pp. 421–424.

[32] A. Wood, J. A. Stankovic, and S. H. Son, “Jam: A jammed-area mapping service for sensor networks,” in *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*. IEEE, 2003, pp. 286–297.

[33] H. Liu, X. Wenyuan, Y. Chen, and Z. Liu, “Localizing jammers in wireless networks,” in *Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2009, pp. 1–6.

[34] G. Sun and J. van de Beek, “Simple distributed interference source localization for radio environment mapping,” in *Wireless Days (WD), 2010 IFIP*. IEEE, 2010, pp. 1–5.

[35] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Strom, “Cooperative received signal strength-based sensor localization with unknown transmit powers,” *Signal Processing, IEEE Transactions on*, vol. 61, no. 6, pp. 1389–1403, 2013.

[36] L. Lin, H.-C. So, and Y. Chan, “Received signal strength based positioning for multiple nodes in wireless sensor networks,” *Digital Signal Processing*, vol. 25, pp. 41–50, 2014.

[37] H. Lohrasbipeydeh, A. Gulliver, and H. Amindavar, “Blind received signal strength difference based source localization with system parameter errors,” *IEEE Transactions on Signal Processing*, 2014.

[38] R. W. Ouyang, A.-S. Wong, and C.-T. Lea, “Received signal strength-based wireless localization via semidefinite programming: noncooperative and cooperative schemes,” *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 3, pp. 1307–1318, 2010.

[39] P. Moravek, D. Komosny, M. Simek, D. Girbau, and A. Lazaro, “Energy analysis of received signal strength localization in wireless sensor networks,” *Radioengineering*, vol. 10, no. 4, pp. 937–945, 2011.

[40] W. Xu, K. Ma, W. Trappe, and Y. Zhang, “Jamming sensor networks: attack and defense strategies,” *IEEE Network*, vol. 20, no. 3, pp. 41–47, 2006.

[41] W. Xu, T. Wood, W. Trappe, and Y. Zhang, “Channel surfing and spatial retreats: defenses against wireless denial of service,” in *Proceedings of the 3rd ACM workshop on Wireless security*. ACM, 2004, pp. 80–89.

[42] G. Noubir, “On connectivity in ad hoc networks under jamming using directional antennas and mobility,” in *Wired/Wireless Internet Communications*. Springer, 2004, pp. 186–200.

[43] Y. S. Kim, F. Mokaya, E. Chen, and P. Tague, “All your jammers belong to usâĂŤlocalization of wireless sensors under jamming attack,” in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 949–954.

[44] T. Cheng, P. Li, and S. Zhu, “An algorithm for jammer localization in wireless sensor networks,” in *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*. IEEE, 2012, pp. 724–731.

[45] D. Liu, P. Ning, A. Liu, C. Wang, and W. K. Du, “Attack-resistant location estimation in wireless sensor networks,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 11, no. 4, p. 22, 2008.

[46] Z. Liu, H. Liu, W. Xu, and Y. Chen, “Exploiting jamming-caused neighbor changes for jammer localization,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 547–555, 2012.

[47] N. Bulusu, J. Heidemann, and D. Estrin, “Gps-less low-cost outdoor localization for very small devices,” *Personal Communications, IEEE*, vol. 7, no. 5, pp. 28–34, 2000.

[48] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, “Weighted centroid localization in zigbee-based sensor networks,” in *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*. IEEE, 2007, pp. 1–6.

[49] K. Pelechrinis, I. Koutsopoulos, I. Broustis, and S. V. Krishnamurthy, “Lightweight jammer localization in wireless networks: System design and implementation,” in *Proc. IEEE Global Telecommunications Conference (GLOBECOM ’09)*, 2009, pp. 1–6.

[50] C. J. C. H. Watkins, “Learning from delayed rewards.” Ph.D. dissertation, University of Cambridge, 1989.

[51] D. P. Bertsekas and H. Yu, “Distributed asynchronous policy iteration in dynamic programming,” in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 1368–1375.

[52] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, “Speed: A stateless protocol for real-time communication in sensor networks,” in *Proc. IEEE the 23rd International Conference on Distributed Computing Systems*, 2003, pp. 46–55.

[53] W. Su and M. Gerla, “Ipv6 flow handoff in ad hoc wireless networks using mobility prediction,” in *Proc. IEEE Global Telecommunications Conference (GLOBECOM’99)*, vol. 1, 1999, pp. 271–275.

[54] C.-K. Toh, “Associativity-based routing for ad hoc mobile networks,” *Wireless Personal Communications*, vol. 4, no. 2, pp. 103–139, 1997.

[55] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *Proc. the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 255–265.

[56] Y. Liu and W. Trappe, “Jammer forensics: Localization in peer to peer networks based on q-learning,” in *Proc. 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 1737–1741.

[57] G. G. Helmer, J. S. Wong, V. Honavar, and L. Miller, “Intelligent agents for intrusion detection,” in *Proc. IEEE Information Technology Conference*, 1998, pp. 121–124.

[58] Y.-a. Huang and W. Lee, “A cooperative intrusion detection system for ad hoc networks,” in *Proc. the 1st ACM workshop on Security of ad hoc and sensor networks*, 2003, pp. 135–147.

[59] L. Zhou and Z. J. Haas, “Securing ad hoc networks,” *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[60] M. K. Marina and S. R. Das, “On-demand multipath distance vector routing in ad hoc networks,” in *Proc. IEEE the Ninth International Conference on Network Protocols*, 2001, pp. 14–23.

[61] M. Al-Shurman, S.-M. Yoo, and S. Park, “Black hole attack in mobile ad hoc networks,” in *Proceedings of the 42nd annual Southeast regional conference*. ACM, 2004, pp. 96–97.

[62] P. Tague, S. Nabar, J. A. Ritcey, and R. Poovendran, “Jamming-aware traffic allocation for multiple-path routing using portfolio selection,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 1, pp. 184–194, 2011.

[63] P. Tague, S. Nabar, J. A. Ritcey, D. Slater, and R. Poovendran, “Throughput optimization for multipath unicast routing under probabilistic jamming,” in *Proc. IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC ’08)*, 2008, pp. 1–5.

[64] K. Ma, Y. Zhang, and W. Trappe, “Mobile network management and robust spatial retreats via network dynamics,” in *Proc. IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 2005, pp. 8–pp.

[65] K. Panyim and P. Krishnamurthy, “A hybrid key predistribution scheme for sensor networks employing spatial retreats to cope with jamming attacks,” *Mobile Networks and Applications*, vol. 17, no. 3, pp. 327–341, 2012.

[66] G. Yue, “Antijamming coding techniques,” *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 35–45, 2008.

[67] G. Yue and X. Wang, “Anti-jamming coding techniques with application to cognitive radio,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5996–6007, 2009.

[68] X. Xu, B. Zheng, J. Zhang, and J. Yan, "An efficient anti-jamming piecewise coding in cognitive ofdm," in *Proc.IEEE the 12th IEEE International Conference on Communication Technology (ICCT)*. IEEE, 2010, pp. 1–4.

[69] K. Cheun and W. E. Stark, "Performance of robust metrics with convolutional coding and diversity in fhss systems under partial-band noise jamming," *IEEE Transactions on Communications*, vol. 41, no. 1, pp. 200–209, 1993.

[70] O. Sidek and A. Yahya, "Reed solomon coding for frequency hopping spread spectrum in jamming environment," *American Journal of Applied Sciences*, vol. 5, no. 10, p. 1281, 2008.

[71] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[72] A. Jamakovic and S. Uhlig, "On the relationship between the algebraic connectivity and graph's robustness to node and link failures," in *Proc. IEEE the 3rd EuroNGI Conference on Next Generation Internet Networks*, 2007, pp. 96–102.

[73] N. M. M. de Abreu, "Old and new results on algebraic connectivity of graphs," *Linear algebra and its applications*, vol. 423, no. 1, pp. 53–73, 2007.

[74] S. Jain and S. Sahu, "Topology vs position based routing protocols in mobile ad hoc networks: A survey," in *International Journal of Engineering Research and Technology*, vol. 1, no. 3 (May-2012). ESRSA Publications, 2012.

[75] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[76] B. Mohar and Y. Alavi, "The laplacian spectrum of graphs," *Graph theory, combinatorics, and applications*, vol. 2, pp. 871–898, 1991.

[77] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *Proc. IEEE the 45th Conference on Decision and Control (CDC)*, 2006, pp. 6605–6611.

[78] P. Di Lorenzo and S. Barbarossa, "Distributed estimation and control of algebraic connectivity over random graphs," *IEEE Transactions on Signal Processing*, vol. 62, no. 21, pp. 5615–5628, 2014.

[79] F. Morbidi, "On the control of the algebraic connectivity and clustering of a mobile robotic network," in *Proc.European Control Conference (ECC)*, 2013, pp. 2801–2806.

[80] A.S.. Ibrahim, K.G.. Seddik, and K.J.R.. Liu, "Improving connectivity via relays deployment in wireless sensor networks," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, 2007, pp. 1159–1163.

[81] M. Romoozi and H. Babaei, "Improvement of connectivity in mobile ad-hoc networks by adding static nodes based on a realistic mobility model," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 4, pp. 76–83, 2011.

[82] M.C.. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2009, pp. 3628–3633.

[83] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," in *Proc. American Control Conference (ACC)*, 2005, pp. 99–103.

[84] Z. Han, A.L.. Swindlehurst, and K.J.R.. Liu, "Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3533–3546, 2009.

[85] K. C. Nguyen, T. Alpcan, and T. Başar, "Stochastic games for security in networks with interdependent nodes," in *Proc. IEEE International Conference on Game Theory for Networks (GameNets' 09)*, 2009, pp. 697–703.

[86] B. Wang, Y. Wu, K.J.R.. Liu, and T.C.. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 877–889, 2011.

[87] A. Garnaev and W. Trappe, "Stationary equilibrium strategies for bandwidth scanning," in *Multiple Access Communcations*, ser. LNCS, M. Jonsson and et al, Eds., vol. 8310. Springer, 2013, pp. 168–183.

[88] ——, "Anti-jamming strategies: a stochastic game approach," in *Mobile Networks and Management*, ser. LNICST, R. Aguero and et al, Eds., vol. 141. Springer, 2015, pp. 230–243.

[89] G. Calinescu, S. Kapoor, K. Qiao, and J. Shin, "Stochastic strategic routing reduces attack effects," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2011, pp. 1–5.

[90] C. Comaniciu, N.B.. Mandayam, and H.V.. Poor, *Wireless networks multiuser detection in cross-layer design*. New York: Springer, 2005.

[91] S. Verdu, *Multiuser detectiony*. New York: University Press, 1998.

[92] H.L.V.. Trees, *Detection, estimation, and modulation theory*. New York: Wiley, 2001.

[93] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, pp. 523–531, 1967.

[94] F.F.. Digham, M.S.. Alouini, and M.K.. Simon, "On the energy detection of unknown signals over fading channels," in *Proc. IEEE International Conference on Communications (ICC)*, 2003, pp. 3575–3579.

[95] A. Garnaev, Y. Liu, and W. Trappe, "Anti-jamming strategy versus a low-power jamming attack when intelligence of adversary's attack type is unknown," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 1, pp. 49–56, March 2016.

[96] A. Garnaev, W. Trappe, and C.-T.. Kung, "Dependence of optimal monitoring strategy on the application to be protected," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 1054–1059.

[97] A. Garnaev and W. Trappe, "Bandwidth scanning involving a Bayesian approach to adapting the belief of an adversary's presence," in *Proc. IEEE Conference on Communications and Network Security (CNS)*, 2014, pp. 35–43.

[98] A.M.. Blanco and J.A.. Bandoni, “Eigenvalue and singular value optimization,” *Mecanica Computational*, 2003.

[99] L. Vandenberghe and S. Boyd, “Semidefinite programming,” *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[100] R.H.. Tutuncu, K.C.. Toh, and M.J.. Todd, “Solving semidefinite-quadratic-linear programs using SDPT3,” *Mathematical programming*, vol. 95, no. 2, pp. 189–217, 2003.

[101] M.J.. Todd, K.C.. Toh, and R.H.. Tutuncu, “A MATLAB software package for semidefinite programming,” *Technical report. School of OR and IE, Cornell University. Ithacat NY*, 1996.

[102] K. Fujisawa, Y. Futakata, M. Kojima, S. Matsuyama, S. Nakamura, K. Nakata, and M. Yamashita, “SDPA-M (semidefinite programming algorithm in matlab) user’s manual-version 6.2,” *Research Reports on Mathematical and Computing Sciences, Series B: Operation Res., Dep. Math. and Computing Sci., Tokyo Institute of Technol., Japan*, vol. 10, 2000.

[103] B. Borchers, “CSDP, AC library for semidefinite programming,” *Optimization methods and Software*, vol. 11, no. 1-4, pp. 613–623, 1999.

[104] G. Owen, *Game Theory*. Academic Press, 1982.

[105] T. F. Lunt, “A survey of intrusion detection techniques,” *Computers & Security*, vol. 12, no. 4, pp. 405–418, 1993.

[106] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, “Network intrusion detection,” *IEEE Network*, vol. 8, no. 3, pp. 26–41, 1994.

[107] W. Lee and S. J. Stolfo, “Data mining approaches for intrusion detection,” in *Proc. the 7th Conference on USENIX Security Symposium*, vol. 7. USENIX Association, 1998, pp. 6–6.

[108] W. Lee, S. J. Stolfo, and K. W. Mok, “A data mining framework for building intrusion detection models,” in *Proc. the 1999 IEEE Symposium on Security and Privacy*, 1999, pp. 120–132.

[109] Y. Liu, A. Garnaev, and W. Trappe, “Maintaining throughput network connectivity in ad hoc networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[110] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[111] R. Albert, H. Jeong, and A.-L. Barabási, “Error and attack tolerance of complex networks,” *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[112] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[113] C. Haythornthwaite, “Social networks and internet connectivity effects,” *Information, Community & Society*, vol. 8, no. 2, pp. 125–147, 2005.

[114] N. P. Hummon and P. Dereian, “Connectivity in a citation network: The development of dna theory,” *Social Networks*, vol. 11, no. 1, pp. 39–63, 1989.

[115] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, “Wireless sensor network security: A survey,” *Security in distributed, grid, mobile, and pervasive computing*, vol. 1, p. 367, 2007.

[116] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures,” *Ad hoc networks*, vol. 1, no. 2, pp. 293–315, 2003.

[117] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, “Security in mobile ad hoc networks: challenges and solutions,” *IEEE Wireless Communications*, vol. 11, no. 1, pp. 38–47, 2004.

[118] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, “A survey of game theory as applied to network security,” in *Proc.IEEE the 43rd Hawaii International Conference on System Sciences (HICSS)*, 2010, pp. 1–10.

[119] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, “Game theory meets network security and privacy,” *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 25, 2013.

[120] A. Gueye, J. C. Walrand, and V. Anantharam, “Design of network topology in an adversarial environment,” in *Decision and Game Theory for Security*. Springer, 2010, pp. 1–20.

[121] R. Lindelauf and I. Blankers, “Key player identification: A note on weighted connectivity games and the shapley value,” in *Proc. IEEE International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2010, pp. 356–359.

[122] G. Theodorakopoulos and J.S. Baras, “A game for ad hoc network connectivity in the presence of malicious users,” in *Proc. IEEE Global Telecommunications Conference(GLOBECOM ’06)*, 2006, pp. 1–5.

[123] Z. Zhang, J. Wu, J. Deng, and M. Qiu, “Jamming ack attack to wireless networks and a mitigation approach,” in *Proc. IEEE Global Telecommunications Conference (GLOBECOM ’08)*, 2008, pp. 1–5.

[124] A. Attar, H. Tang, A.V. Vasilakos, F.R. Yu, and V.C.M. Leung, “A survey of security challenges in cognitive radio networks: Solutions and future research directions,” *Proceedings of the IEEE*, vol. 100, pp. 3172–3186, 2012.

[125] B. Mohar, “Laplace eigenvalues of graphsâĂŤa survey,” *Discrete mathematics*, vol. 109, no. 1, pp. 171–183, 1992.

[126] A. Neyman and S. Sorin, *Stochastic games and applications*. Springer Science & Business Media, 2003, vol. 570.

[127] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press Cambridge, 2007, vol. 1.

[128] I. E. Telatar *et al.*, “Capacity of multi-antenna gaussian channels,” *European transactions on telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[129] W. Rhee and J.M. Cioffi, “Ergodic capacity of multi-antenna gaussian multiple-access channels,” in *Proc. IEEE Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 1, 2001, pp. 507–512.

[130] P. Karthikeyan and S. Baskar, “Genetic algorithm with ensemble of immigrant strategies for multicast routing in ad hoc networks,” *Soft Computing*, vol. 19, no. 2, pp. 489–498, 2015.

[131] A. Baskaran, N. Chaudhari, R. D. Caytiles, and N. C. S. Iyengar, “Fuzzy optimized and bee inspired routing protocol for improved qos in mobile ad hoc networks,” *International Journal of Control and Automation*, vol. 9, no. 8, pp. 391–402, 2016.

[132] S. Arora, E. Hazan, and S. Kale, “The multiplicative weights update method: a meta-algorithm and applications.” *Theory of Computing*, vol. 8, no. 1, pp. 121–164, 2012.

Appendix A

A.1 Solution for game matrix 7.2

Since the game Γ is discounted stochastic game, it has an equilibrium. Introduce the following auxiliary notations: $\Gamma_{1,1} = C^G_{\bar{p}}, \Gamma_{1,2} = C^B_{\bar{p}}, \Gamma_{2,1} = C^G_{p}, \Gamma_{2,2} = C^B_{p} + \eta(C^B_{\bar{p}} - C^B_{p})$, $\gamma_1 = \gamma_G$ and $\gamma_2 = \gamma_B$. None that $\gamma_2 > \gamma_1$ due to $\gamma_B > \gamma_G$.

Note that $\mathrm{val}(V)$, $\mathbf{p}$ and $\mathbf{q}$ is a solution of maxmin and minmax Shapley equations if and only if

$$\mathrm{val}(V) = \nu. \tag{A.1}$$

$$\begin{aligned}
&\max\ \nu,\\
&L_1(\nu, \mathbf{p}) := (\Gamma_{11} + \gamma_1 \nu) p_1 + (\Gamma_{21} + \gamma_1 \nu) p_2 \geq \nu,\\
&L_2(\nu, \mathbf{p}) := (\Gamma_{12} + \gamma_2 \nu) p_1 + (\Gamma_{22} + \gamma_2 \nu) p_2 \geq \nu,\\
&p_1 + p_2 = 1,\ p_1 \geq 0,\ p_2 \geq 0,
\end{aligned} \tag{A.2}$$

$$\begin{aligned}
&\min\ \nu,\\
&l_1(\nu, \mathbf{q}) := (\Gamma_{11} + \gamma_1 \nu) q_1 + (\Gamma_{12} + \gamma_2 \nu) q_2 \leq \nu,\\
&l_2(\nu, \mathbf{q}) := (\Gamma_{21} + \gamma_1 \nu) q_1 + (\Gamma_{22} + \gamma_2 \nu) q_2 \leq \nu,\\
&q_1 + q_2 = 1,\ q_1 \geq 0,\ q_2 \geq 0.
\end{aligned} \tag{A.3}$$

The equilibrium can be either in pure strategies or in mixed (randomized) strategies. First let us consider the cases when the equilibrium is in pure strategies. By

(A.1)-(A.3), $\mathbf{p} = (1,0)$ and $\mathbf{q} = (1,0)$ if and only if the following conditions hold

$$\Gamma_{12} + \gamma_2 \text{val}(\Gamma) \geq \Gamma_{11} + \gamma_1 \text{val}(\Gamma) \geq \Gamma_{21} + \gamma_1 \text{val}(\Gamma), \tag{A.4}$$

$$\text{val}(\Gamma) = \Gamma_{11} + \gamma_1 \text{val}(\Gamma). \tag{A.5}$$

By (A.5), we get $\text{val}(\Gamma)$ equals to $\frac{\Gamma_{11}}{1-\gamma_1}$. Since $1 > \gamma_2 > \gamma_1$, (A.4) is equivalent to

$$\Gamma_{11} \geq \Gamma_{21} \text{ and } \text{val}(\Gamma) \geq \frac{\Gamma_{12} - \Gamma_{11}}{\gamma_1 - \gamma_2}. \tag{A.6}$$

Substituting $\text{val}(\Gamma)$ into the last condition (A.6) implies

$$\Gamma_{21} \leq \Gamma_{11} \leq \frac{1-\gamma_1}{1-\gamma_2} \Gamma_{12},$$

and (a) follows. The cases (b)-(d) can be considered similarly.

If conditions of (a)-(d) do not hold then the equilibrium has to be in mixed strategies, i.e., $p_i > 0$, $q_i > 0$ for $i = 1, 2$. Then, since $L_i(\nu, \mathbf{p})$ is linear on $\mathbf{p}$ and $l_i(\nu, \mathbf{q})$ is linear on $\mathbf{q}$ for fixed ν, to get the Nash equilibrium, the constraints in (A.2) and (A.3) have to hold equalities [104]. Namely,

$$L_i(\nu, \mathbf{p}) = \nu \text{ and } l_i(\nu, \mathbf{q}) = \nu \text{ for } i = 1, 2. \tag{A.7}$$

Taking into account that $\mathbf{p}$ and $\mathbf{q}$ are probability vectors, solving (A.7) implies (e), and the result follows.

A.2 Proof of Proposition 7.3.1

Proof. According to [132], let us assume $\Phi^t = \sum_i w_i^t$, then

$$\Phi^{t+1} = \sum_i w_i^t \left(1 - \zeta \frac{c_i^t - \tilde{c}^t}{\max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t} \right)$$

Since $p_i^t = w_i^t / \Phi^t$, we get

$$\Phi^{t+1} = \Phi^t - \zeta \Phi^t \sum_i p_i^t \frac{c_i^t - \tilde{c}^t}{\max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t}$$

Assume $f(e_i^t) = \frac{c_i^t - \tilde{c}^t}{\max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t}$. Because $1 - x \leq e^{-x}$ when $|x| < 1$, we obtain the following upper bound for Φ^{t+1},

$$\Phi^{t+1} \leq \Phi^t exp(-\zeta \sum_i p_i^t f(e_i^t))$$

Then, we derive

$$\Phi^{T+1} \leq \Phi^1 exp(-\frac{\zeta}{\lambda} \sum_{t=1}^{T} \sum_i p_i^t f(e_i^t)) \tag{A.8}$$

where $\Phi^1 = 2$. Then, let us obtain the lower bound of Φ^{T+1} since

$$\Phi^{T+1} \geq w_i^{T+1} = \prod_{t \leq T} (1 - \zeta \frac{c_i^t - \tilde{c}^t}{\max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t})$$

Since $c_i^t \geq 0$, $\tilde{c}^t \geq 0$ and $c_i^t - \tilde{c}^t \leq \max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t$, we obtain $\frac{c_i^t - \tilde{c}^t}{\max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t}$ is within interval $[-1,\ 1]$. By using the facts that $(1-\zeta)^x \leq (1 - \zeta x)$ when $x \epsilon\, [0, 1]$ and $(1+\zeta)^{-x} \leq (1 - \zeta x)$ when $x \epsilon\, [-1, 0]$, we obtain

$$\Phi^{T+1} \geq (1-\zeta)^{\sum_{f \geq 0} f(e_i^t)} (1+\zeta)^{-\sum_{f<0} f(e_i^t)} \tag{A.9}$$

Putting (A.8) and (A.9) together, take the logarithms and re-arrange them, we obtain,

$$\begin{aligned} \sum_{t=1}^{T} \sum_i p_i^t f(e_i^t) \leq \frac{1}{\zeta} ln 2 + \frac{1}{\zeta} \sum_{f \geq 0} f(e_i^t) ln(\frac{1}{1-\zeta}) \\ + \frac{1}{\zeta} \sum_{f<0} f(e_i^t) ln(1+\zeta) \end{aligned}$$

Since $ln(\frac{1}{1-\zeta}) \leq \zeta + \zeta^2$ and $ln(1+\zeta) \geq \zeta - \zeta^2$ both when $\zeta \leq \frac{1}{2}$, from the above equation we get:

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i} p_i^t f(e_i^t) &\leq \frac{1}{\zeta} ln2 + \frac{1}{\zeta}\sum_{f\geq 0} f(e_i^t)(\zeta+\zeta^2) \\ &\quad + \frac{1}{\zeta}\sum_{f<0} f(e_i^t)(\zeta-\zeta^2) \\ &= \frac{1}{\zeta} ln2 + \sum_{t=1}^{T} f(e_i^t) + \zeta\sum_{t=1}^{T}\left|f(e_i^t)\right|\end{aligned}$$

Thus, if we assume $m^t = \max\left\{c_1^t, c_2^t\right\} + \tilde{c}^t$, we obtain,

$$\begin{aligned}\sum_{t=1}^{T}\frac{\mathbf{p}^t\mathbf{e}^t}{m^t} &\leq \frac{1}{\zeta} ln2 + \sum_{t=1}^{T}\frac{e_i^t}{m^t} + \zeta\sum_{t=1}^{T}\frac{\left|e_i^t\right|}{m^t} \\ &\leq \frac{1}{\zeta} ln2 + \sum_{t=1}^{T}\frac{e_i^t}{m^t} + \zeta T \qquad \text{(A.10)}\end{aligned}$$

Assume $\xi = \min\left\{\sqrt{\frac{ln2}{T}}, \frac{1}{2}\right\}$. Since $\xi = \sqrt{\frac{ln2}{T}}$ when $T > 2$, (A.10) becomes

$$\sum_{t=1}^{T}\frac{\mathbf{p}^t\mathbf{e}^t}{m^t} \leq \sum_{t=1}^{T}\frac{e_i^t}{m^t} + 2\sqrt{Tln2}$$

The right hand-side is proved.

We prove the left hand-side. From the definition of $\mathbf{p}^t$, we obtain:

$$\sum_{t=1}^{T}\frac{\mathbf{p}^t\mathbf{e}^t}{m^t} - \sum_{t=1}^{T}\frac{e_i^t}{m^t} = \sum_{t=1}^{T}\frac{p_1e_1^t + p_2e_2^t - e_i^t}{m^t}$$

Since the expert makes the same decision during the whole process, without loss of generality, we assume the expert always chooses the 2nd decision. Therefore, we get

$$\sum_{t=1}^{T}\frac{\mathbf{p}^t\mathbf{e}^t}{m^t} - \sum_{t=1}^{T}\frac{e_i^t}{m^t} = \sum_{t=1}^{T} p_1^t\frac{e_1^t - e_2^t}{m^t} \geq -\frac{1}{2}\sum_{t=1}^{T} p_1^t \geq -\frac{T}{2}$$

since $\frac{e_1^t - e_2^t}{m^t} \geq -\frac{1}{2}$ according to the definition of e_i^t and m^t. The proposition 7.3.1 gets proved. □