SECURITY ASSESSMENT OF DYNAMIC SPECTRUM ACCESS IN EMERGING WIRELESS NETWORKS

by

Moinul Hossain

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Electrical Engineering

Charlotte

2020

Approved by:

_____
Dr. Jiang (Linda) Xie

_____
Dr. Tao Han

_____
Dr. Weichao Wang

_____
Dr. Yu Wang

ABSTRACT

MOINUL HOSSAIN. Security Assessment of Dynamic Spectrum Access in Emerging Wireless Networks. (Under the direction of DR. JIANG (LINDA) XIE)

The proliferation of wireless technologies in a licensed manner has resulted in the scarcity of the radio spectrum. Therefore, spectrum availability has become a challenge for new and existing wireless technologies. Federal Communications Commission (FCC) has proposed a new spectrum sharing strategy, i.e., dynamic spectrum access (DSA), to opportunistically access the underutilized spectrum resources of licensed users and to fairly share the unlicensed spectrum with others. Cognitive Radio (CR) works as a promising technology to enable this opportunistic or dynamic spectrum access. Moreover, DSA and CR also work as fundamental building blocks of future spectrum coexistence, which aims to improve spectrum equity. This dissertation serves as a means to achieve secure and fair coexistence of different wireless technologies in the same spectrum—whether licensed or unlicensed.

In this research, a novel attack surface, off-sensing interval, is introduced. It highlights a novel room of vulnerabilities in state-of-the-art channel sensing processes. This vulnerability illustrates how an attacker can ingeniously avoid the sensing interval of the victim and can intelligently interfere with the transmission or reception of the victim to trick it into believing that it is interfering with a licensed user. Such a scenario pushes the victim to perform a spectrum handoff and influences its spectrum utilization. It is named an *off-sensing* attack. Furthermore, a cross-layer attack in CR-based wireless mesh networks is proposed, where the attacker deploys the off-sensing attack as an auxiliary attack to influence the traffic flow around the victim to divert traffic through a target node. It is named an *off-sensing and route manipulation* attack. Then, a strategy to thwart the off-sensing attack is proposed, where the defender hops through different channels in the spectrum band to confuse the

attacker. The interaction between attackers and a defender is modeled as a Markov decision process to assist the defender in making optimal decisions.

In addition, another attack surface is discovered in state-of-the-art rendezvous and spectrum handoff processes in infrastructure-less CR-based networks. As infrastructure-less networks operate without a central entity, the lack of vigilance in current spectrum handoff strategies engenders this vulnerability where a selfish SU can trigger an early handoff to reserve the best available channel sooner than benign SUs. This research helps to design secure spectrum handoff processes.

Finally, as the rapid commercialization of Internet-of-Things (IoT) is taking place, the density of spectrum hungry devices are increasing. This dense deployment of IoT devices—that may follow different wireless technologies—in the shared spectrum creates a new challenge to solve: secure coordination among co-located IoT devices from different IoT networks. This dissertation sheds light on this unique challenge and introduces a novel security vulnerability where an attacker can pose as a hidden terminal from a different network and compromise the victim device, namely *hidden terminal emulation* (HTE) attack. As the dense deployment of IoT devices will naturally aggravate such hidden terminal interference, it facilitates the HTE attacker with plausible deniability to interfere with its alleged hidden counterparts. This dissertation assesses this issue and proposes detection and defense measures, namely *Third Eye* and *Jump and Wobble*, receptively.

In summary, dynamic spectrum access promises to be one of the significant ideas to solve the spectrum scarcity problem. This research is essential to conduct an in-depth security assessment of spectrum sharing operations—which are unique to the dynamic spectrum access—before the commercialization of such technologies.

ACKNOWLEDGEMENTS

I would like to express my greatest and deepest gratitude to my advisor and mentor, Professor Jiang Xie, for her guidance and supervision throughout these years. The scientific methods and work ethic that I learned from her shaped me as a person and as a researcher. She has established a high standard of research for us and inspired us to pursue excellence in research. Though it was very challenging, it has been the best days of my life. I could not ask for a better mentor than her.

I would also like to thank my committee members: Dr. Tao Han, Dr. Weichao Wang, and Dr. Yu Wang for their generous time and insightful advice. In addition, I appreciate the GASP grant from UNC-Charlotte and Research Assistantships from the National Science Foundation (NSF) as the financial assistance for this dissertation.

I am grateful to my family, friends, and labmates. Their best wishes and consistent supports helped me to get through this long and difficult journey. Lastly, I share this achievement with my beloved wife and good friend, Nigar Sultana, who has always been there for me and walked every step with me during this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CCC Common Control Channel

CI Contention Interval

CRAHN Cognitive Radio-based Ad Hoc Network

CRN CR-based Network

CRWMN CR-based Wireless Mesh Network

CSMA Carrier Sensing Multiple Access

CTS Clear-to-send

DCF Distributed Coordination Function

DoS Denial-of-Service

DSA Dynamic Spectrum Access

FCC Federal Communications Commission

HMM Hidden Markov Model

HnS Hide and Seek

HTE Hidden Terminal Emulation

IoT Internet-of-Things

JnW Jumap and Wobble

MAC Media Access Control

MDP Markov Decision Process

OS Off-sensing

OS-DoS Off-sensing Denial-of-Service

OSRM Off-sensing and Route Manipulation

PDR Packet Drop Rate

PHY Physical

pmf Probability Mass Function

PU Primary User

PUE Primary User Emulation

RF Radio Frequency

RPU Routing-toward-primary-user

RSS Received Signal Strength

RTS Request-to-send

RTT Round-trip-time

SIC Successive Interference Cancellation

SSDF Spectrum Sensing Data Falsification

SU Secondary User

TCP Transmission Control Protocol

WLAN Wireless Local Area Network

WMN Wireless Mesh Network

WMR Wireless Mesh Router

CHAPTER 1: INTRODUCTION

1.1 Background on Dynamic Spectrum Access and Spectrum Coexistence

The number of network devices is expected to radically increase—with an estimate of above 50 billion connected devices by 2025—to support new services and applications. As we are moving towards a world where each thing could probably have some form of wireless connectivity, we are facing new challenges along the way. Moreover, these new services and applications demand higher data rates, with reduced latency and increased system capacity.

To conform to these rising demands, wireless networks must undergo suitable changes, and one of the biggest challenges in meeting these demands is limited spectrum resources because there is not enough room for growing wireless demands. The constrained amount of radio resource and the licensed way of utilizing this resource have made it a challenge to meet the ever-increasing demand for wireless services. On the other hand, the Federal Communications Commission (FCC) has concluded that the radio spectrum is not balanced in terms of resources and traffic-load; a significant portion of the radio spectrum remains underutilized, whereas a high volume of traffic appears in another portion. In light of such an inefficient utilization of precious radio resource, the FCC proposed a new spectrum sharing paradigm, where an unlicensed user (or secondary user (SU)) can opportunistically utilize a licensed channel when the licensed user (or primary user (PU)) is idle; this represents one of the aspects of opportunistic or dynamic spectrum access (DSA). One underlying idea that has emerged as a promising technology to realize this aspect is cognitive radio (CR). With the capability of sensing the frequency bands in a time-space varying spectrum environment and adjusting the operational parameters based on the sensing

outcome, CR technology allows an SU to exploit these underutilized licensed channels opportunistically, and an example of this mechanism is IEEE 802.22 (WRAN) where underutilized resources in the TV frequency spectrum is utilized by unlicensed users on a non-interference basis. However, the opportunistic use of licensed channels does not represent the full scope of DSA because it considers the presence of licensed and unlicensed users in only licensed spectrum. Another aspect of DSA is the coexistence of heterogeneous networks in the unlicensed spectrum, such as 2.4 GHz and 5 GHz bands. An example of this strategy is the spectrum coexistence between LTE and WiFi in the 5 GHz band. Therefore, the full scope of DSA represents spectrum sharing in both licensed and unlicensed spectrum.

Like traditional wireless networks, CR-based networks (CRNs) are prone to conventional network attacks [1] (e.g., jamming, packet drop, and eavesdropping). In addition, new genres of attacks have emerged in CRNs due to its unique way of operation, i.e., DSA [2, 3, 4]. The two most studied attacks specifically in CRNs that try to compromise the spectrum sensing process are primary user emulation (PUE) [5] and spectrum sensing data falsification (SSDF) [6]. Depending on the motive of the attacker, these attacks help to either maximize attacker's own channel utilization (i.e., selfish attacker) or to sabotage the network operation of the victim (i.e., malicious attacker). In PUE, an attacker masquerades as a PU during the sensing interval to trick the victim into avoiding the channel. In SSDF, an attacker shares false sensing information with victims to manipulate the consensus on channel availability in cooperative spectrum sensing techniques. Hence, both attacks have the same attack objective—to reduce the spectrum utilization of the victim.

Unfortunately, these two attacks do not represent the complete picture of security vulnerabilities in DSA, and security researchers are required to perform a thorough investigation of the vulnerabilities in the CR technology before deploying it commercially. Though aforementioned attacks are distinct in their strategies, they have

the same attack surface, i.e., the sensing process. Nonetheless, there could be other attack surfaces unique to the operational characteristics of the CR technology—or DSA—that require rigorous security assessments. Moreover, the deployment of heterogeneous wireless networks in the unlicensed spectrum may expose the spectrum sharing process to additional security threats.

This research discovers three novel attack surfaces that are vulnerable to intelligent attacks. Such attacks exploit the off-sensing interval of a victim, the heterogeneity among different coexistent networks, the shared nature of spectrum utilization, and the proximity to the victim device in a dense network scenario to interrupt the victim's communication. These attack surfaces are: (1) communication interval, (2) spectrum handoff, and (3) spectrum access. However, before explaining these new attack surfaces, the unique operational characteristics of DSA and spectrum coexistence that enable the creation of such novel attack surfaces require discussion. These unique operations are:

- Spectrum Sensing: SUs can sense the frequencies in a certain spectrum band and access an underutilized channel opportunistically without harmful interference to PUs. It is an important requirement of CR networks to sense the *spectrum holes*. Detecting PUs is the most efficient way to detect spectrum holes. Spectrum sensing techniques can be classified into three following categories. (1) Primary transmitter detection: CRs must have the capability to determine a signal from a primary transmitter [7]. (2) Cooperative spectrum sensing: multiple SUs share sensing information with each other and fuse this information for PU detection [8]. (3) Interference based detection: PU presence is inferred based on the interference experienced by SUs [9]. As SUs have to perform spectrum sensing and transmission, both in a half-duplex radio, the state-of-the-art works suggest that they must do it periodically. Figure 1.1 illustrates the sensing-transmission schedule of the MAC layer.

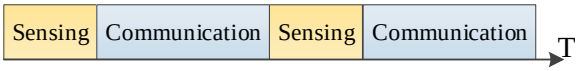


Figure 1.1: Periodic sensing and transmissions.

- Channel Rendezvous and Absence of Control Channel among Heterogeneous Networks: In prior research, most works either considered the availability of a dedicated common control channel (CCC) to exchange control information [10, 11, 12, 13] or did not consider at all [8, 14, 15]. In reality, due to the difference in spectrum usage by PUs (in both space and time), spectrum availability may differ depending on SU locations, and a single common channel is highly unlikely to be available. Hence, two SUs must find a common available channel between them to establish a connection. The state-of-the-art work usually proposes that two SUs hop onto different channels from one time slot to another (i.e., channel-hopping process) until they rendezvous on a common available channel [16, 17, 18, 19], and they can exchange control information afterward. However, the rendezvous process is only feasible when participating devices follow same network protocols, and there is no universal CCC among heterogeneous wireless networks. Therefore, managing spectrum coexistence among different networks is a crucial research issue.

- Spectrum Handoff: As the channel availability is random, SUs change their operating frequency based on its radio environment. CR technology aims to use the licensed spectrum in a dynamic manner by allowing SUs to operate in the best available frequency band, while maintaining seamless communication requirements during the transition to a better spectrum. This introduces a new type of handoff called spectrum handoff, which refers to the process that a SU switches to a new available channel to resume the transmission when the current channel is not available. Based on the moment when the spectrum handoff is carried out, two types of spectrum mobility are introduced: reactive

and proactive approaches. Though both approaches have different strengths, when ensuring permissible interference to PUs, the proactive approach works better than the reactive approach [20].

1.2 Problem Statement

1.2.1 Off-sensing Attack

The etiquette of spectrum sharing on a non-interference basis can turn this into a vulnerability. As discussed earlier, an attack that exploits this vulnerability is the PUE attack. Under this attack, a perpetrator transmits signals whose radio properties emulate the PUs, thereby causing unlicensed users to falsely detect the transmission as a benign PU. As a result, SUs abstain from transmitting on that channel. To defend such an attack, several solutions are proposed based on spectrum sensing approaches.

Under most existing spectrum sensing approaches, SUs periodically sense the spectrum for returning PUs. Therefore, sensing period has to be designed in such a way that the sensing interval coincides with the transmission of PU's. When a PU's transmission does not coincide with the sensing interval of a SU, the SU will fail to detect the PU and may interfere with the PU's transmission. Consequently, the throughput of both networks is impacted. This creates a new window of vulnerability (Figure 1.2) where attackers will interfere only when no neighboring SUs are sensing but transmitting. Using this approach, an attacker can corrupt the transmission of a victim SU. This will trick the SU into believing that it is interfering with a benign PU. Because FCC regulations require a SU to leave a channel within 2 seconds of the arrival of a PU [21], the SU will perform a spectrum handoff and hop to the next available channel. This research names it an off-sensing (OS) attack, meaning to interfere with the victim's transmission when it is not sensing and force it into believing that the victim is interfering with PUs.

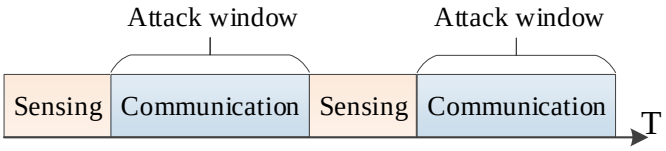


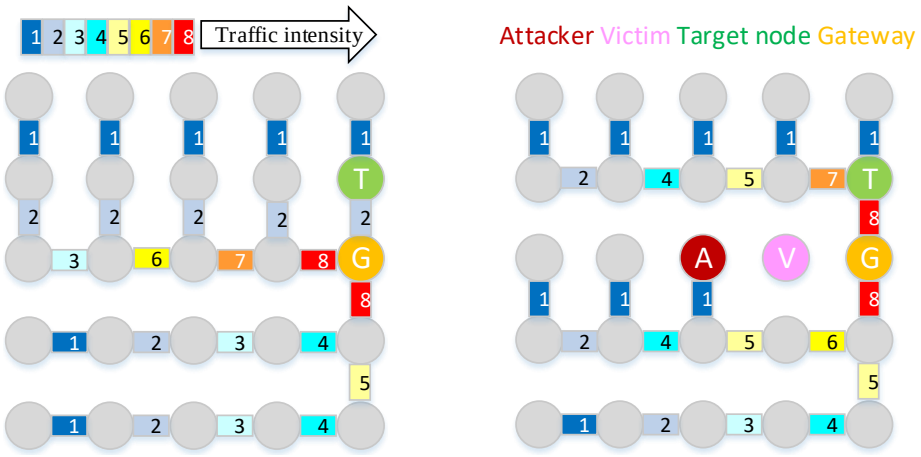
Figure 1.2: The attack window of OS attack.

1.2.2 Cross-layer Attack

One of the promising applications of the CR technology is wireless mesh networks (WMNs)[22, 23], because WMNs usually suffer from inter-flow interference [24] and insufficient channels to mitigate it, whereas the CR technology offers an intelligent solution to the interference problem in WMNs via accessing licensed bands in an opportunistic manner. However, since CRs adapt to the surrounding radio environment based on sensing the radio channels around them and collaborating with peer nodes, it is crucial that the belief of their surroundings is not compromised and diverted in the wrong direction by an attacker.

A CR-WMN consists of CR-enabled wireless mesh routers/access points (CR-WMRs or SUs interchangeably), mobile devices connected to the CR-WMRs, and a gateway which is connected to the Internet. Internet traffic between mobile devices and the gateway is carried by the CR-WMRs, and CR-WMRs can opportunistically access the spectrum when no PUs are using it. However, as discussed, the policy of accessing licensed channels on a non-interfering basis can make it a potential vulnerability; such vulnerabilities are utilized in OS, PUE, and SSDF attacks.

Furthermore, in CR-based networks, the cross-layer nature of some networking protocols may create a new degree of vulnerability because the coupling of multiple layers entails that the decisions made in one layer can be altered by changing the dynamics of other layers. This proposal proposes such an attack under which an attacker can manipulate the routing decisions in the network layer by employing the off-sensing attack as the front-end attack to change the channel availability in lower layers. As a result, the attacker can influence the traffic flow traversing around it and



(a) No attack. (b) OS-RM attack.

Figure 1.3: Traffic heat map.

direct them to a target node (i.e., route manipulation). In particular, the attacker will create a Denial-of-Service (DoS) situation for the victim SU node and divert the traffic flow which initially should go through the victim SU. This lower layer auxiliary attack is named as off-sensing DoS (OS-DoS) attack. With the careful selection of which neighboring SU to perform the OS-DoS attack on, the attacker can direct the diverted traffic flow to a designated target node. The proposal names this cross-layer attack as off-sensing and route manipulation (OS-RM) attack.

In Figure 1.3 (color and number coded), the changes in traffic flow due to the rebalancing effect caused by the OS-DoS attack on the victim node can be observed. Without attack, two neighboring CR-WMRs carry most of the traffic (Figure 1.3(a)) except the target node. However, after the OS-RM attack, it is observed that a portion of previous routes are disrupted (Figure 1.3(b)). As a result, traffic flows change directions and a few nodes who were carrying less traffic are exposed to higher traffic load now. Most significant change in traffic is observed in the target node.

1.2.3 Defense Against Off-sensing Attack

Prior work on OS attack considered two attack scenarios: the attacker always stays on a particular channel and attacks anyone who tries to access the channel (i.e., selfish

attacker), or the attacker knows the channel-hopping sequence of the victim SU and interferes with each transmission attempt of the victim to create a Denial-of-Service (DoS) situation (i.e., DoS attacker). In either case, the attacker plays a deterministic role from a victim's perspective in terms of the operating channel (i.e., the victim can infer the future attack channel). This deterministic hopping sequence of OS-attackers makes it difficult to fortify against traditional defense techniques [2, 4]. Similarly, the assumption that the attacker has the perfect knowledge of the victim's hopping sequence makes it a critical disadvantage for the victim and creates unrealistic scenarios (hopping sequence depends on each SU's surrounding environment, which varies in time and space). Therefore, in realistic conditions, OS-attackers require a random sequence.

Previous work on the defense and detection of the PUE and SSDF attack focused on the sensing interval and the cooperative nature of CRNs, respectively. However, these proposed methods cannot detect OS-attack due to different attack surfaces. Hence, the OS-defense requires focused efforts into the off-sensing interval to safeguard SUs.

As the channel-hopping process is random, SUs can follow any channel-hopping process to rendezvous with each other [16]. Moreover, the rendezvous channel (the channel where two SUs meet) and the transmission channel may differ [25]. Therefore, it is difficult for an attacker to find the operating channel of the victim to perpetrate an OS-attack without any predetermined knowledge. In addition, the OS-DoS attack requires successive detection of victim's operating channel, which is more challenging.

From the defense perspective, a straight-forward approach to identify an OS-attacker is to sense the channel when transmitting. However, hardware limitations (e.g., the transmission antenna would overwhelm the sensing antenna), design considerations (e.g., half-duplex radio), and a decrease in channel utilization (e.g., the victim SU could use an extra sensing time to utilize another white space) restrain this approach. Therefore, the defense and detection process of OS-attacks must adhere to

these constraints.

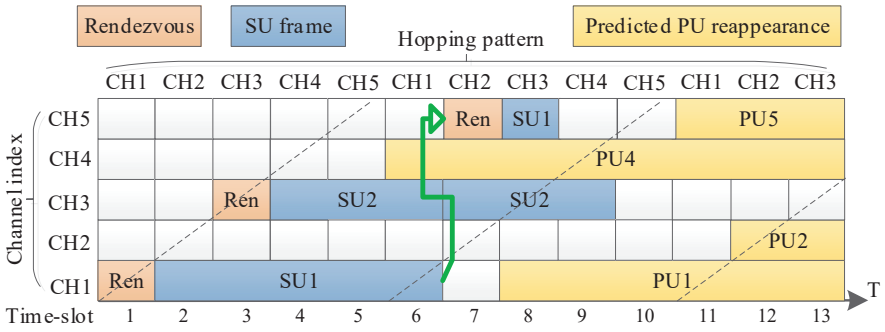
Moreover, most previous research on defense considered that attackers are always present and safeguard process(es) are deployed regardless of the presence of attackers. This assumption costs SUs networking, computational, and energy overhead. Therefore, in resource-constrained networks, the safeguarding process must be aware of the presence of attackers and deploy the safeguard process(es) only when under-attack. Additionally, it must provide the flexibility to trade-off between networking and security performance.

1.2.4 Covert Spectrum Handoff

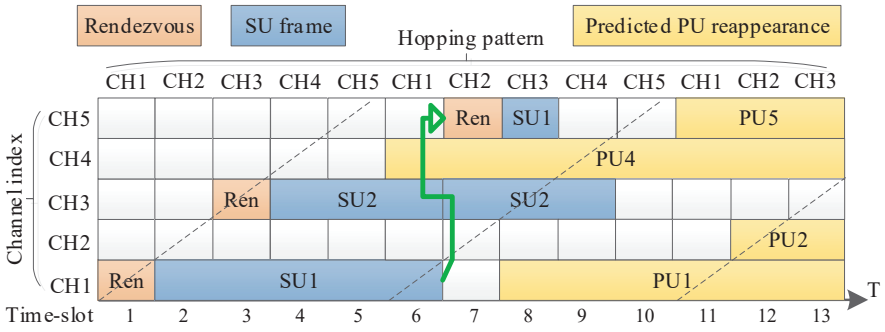
Currently, research on spectrum handoffs in CRNs falls into two approaches based on the moment when SUs initiate handoffs. In the reactive approach, SUs perform spectrum switching and radio frequency (RF) front end reconfiguration after detecting a PU reappearance. In the proactive approach, SUs predict the future channel activity and initiate spectrum switching and RF reconfiguration before a PU reappears on the current channel (based on observed channel usage statistics).

Moreover, most related works on spectrum handoffs and rendezvous processes had assumed identical channels in terms of service rate [20, 26, 14, 27, 17, 28, 29, 30] (i.e., all channels have equal bandwidth). In reality, the available channels are not always going to be identical, the diversity in service rate must be considered to manage handoff more efficiently (e.g., a faster target channel could compensate the handoff delay). Furthermore, in existing proactive handoff approaches, a handoff is triggered only when an SU finds the current channel unavailable for the next frame. Otherwise, it keeps transmitting on the current channel until all frames end.

The concepts of channel-hopping, rendezvous, spectrum handoff, non-identical channels, and handoff trigger time have mostly been studied in isolation. In reality, these functionalities must be considered together in a CRN and identify vulnerabilities before designing corresponding network protocols.



(a) SU1 acting benignly.



(b) SU1 acting selfishly.

Figure 1.4: The covert spectrum handoff in the proactive handoff process.

When all these functionalities are considered together, it engenders a novel vulnerability in the proactive spectrum handoff process where an attacker (or a selfish SU) can trigger an early handoff to reserve the best available channel sooner than benign SUs. An illustration of the vulnerability is provided in Figure 1.4. Here, a common hopping sequence-based rendezvous method is assumed and activity in five channels with non-identical service rates is shown. This illustration considers that each SU packet consists of two frames, a transmission attempt must be preceded by a rendezvous, and SUs must follow the hopping pattern to initiate a new packet transmission. The SU frame length in CH1, CH2, CH3, CH4, and CH5 is 5, 4, 3, 2, and 1 time slots long, respectively. Here, dotted lines represent the hopping pattern. In Figure 1.4(a), we can see SU1 switching from CH1 to CH5 (in slot-7) as it predicts an imminent reappearance of PU1 after transmitting the first frame. To select the target channel, SU1 finds the channel that is not occupied by any SU (e.g., CH3 occupied

by SU2), is least likely to affect by the returning PUs, and has a faster service rate. Hence, SU1 selects CH5 as the target channel. In contrast, a selfish SU can initiate a handoff promptly after the first rendezvous (Figure 1.4(b)) and reserve CH5 sooner (in slot-2). In doing this, the selfish SU is motivated to finish its transmission faster (4 time-slots faster in the example) rather than acting benignly. In this proposal, this selfish attack is named as the *covert spectrum handoff*, which represents performing spectrum handoff secretly to gain access to the best available channel sooner.

1.2.5 Hidden Terminal Emulation Attack

In a coexistence scenario, each heterogeneous network tries to share the same spectrum in a harmonious way among them, which assumes that participating networks are benign. Unfortunately, as the world moves to make spectrum coexistence a reality, we will face the inevitable risk that a participating network (or device) may utilize the coexistence for illicit and selfish purposes, and this participating network may potentially be malicious. Currently, very few existing works have addressed the security implications of spectrum coexistence [31, 32, 33, 34]. This dissertation has discovered a novel vulnerability where attackers can exploit a natural interference scenario to corrupt transmissions or receptions of particular victim IoT devices, i.e., *interference from hidden-terminal devices* of a *different* coexistent IoT network [35]. Fig. 1.5 provides an illustration of this vulnerability where networks A and B are two coexisting IoT networks sharing the same spectrum resource, presumably smart-home networks of two neighboring apartments. Here, nodes B2 and B4 are hidden terminals to nodes A1, A3, and A5, and vice versa. Note that *these two sets of nodes are from two different networks (or even follow different wireless technologies), and under the given scenario, each of these two sets has no idea about the transmissions of the other set because there is no resolution technique to solve the hidden-terminal problem among different networks/technologies.* Therefore, it is probable that, as these two sets of nodes are out of each other's radio range, they may utilize the same radio

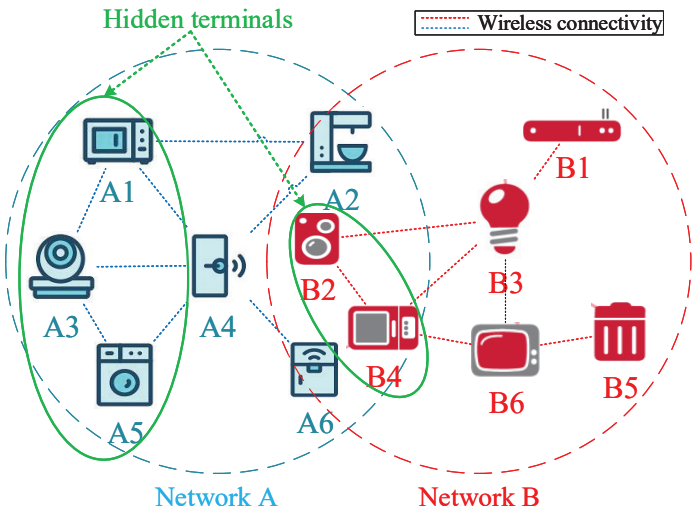


Figure 1.5: Hidden-terminal interference between two coexisting IoT networks.

channel and create interference at nodes that are exposed to both of these sets, i.e., A2 and A4. Hence, if a denial-of-service (DoS) attacker can emulate the transmission and physical characteristics of a hidden terminal, it can justifiably interfere with its hidden counterparts, this research calls it hidden terminal emulation (HTE) attack. Though successive interference cancellation (SIC) provides a solution to this interference problem [36, 37, 38, 39, 40, 41], an intelligent attacker with crafted interference signal will make SIC inoperative because the decodability of SIC depends on the received signal strength and the decoding threshold. Therefore, appropriate security considerations are required for the natural interference scenarios in coexistent IoT networks.

1.2.6 Detection of Hidden Terminal Emulation Attack

In coexistent IoT networks, it is impossible to differentiate between a benign hidden node and a reactive DoS attacker using existing techniques because hidden-terminal interference bears the signature of reactive attacks. Moreover, as hidden-terminal interference is never considered as a benign interference source, current DoS attack detection methods [42, 43, 44, 45, 46, 47, 48, 49, 50, 30] based on network performance measurements, e.g., packet delivery rate, received signal strength (RSS), channel busy

ratio, and the number of retransmission attempts, categorize hidden terminals as DoS attackers, which results in significant false positives. Therefore, new detection strategies are needed that consider hidden terminals as benign interference sources and that can recognize malicious interference from hidden terminals. This dissertation proposes a context-aware detention strategy, namely Third Eye.

1.2.7 Defense against Hidden Terminal Emulation Attack

Most existing counter-mechanism strategies to defend unwanted interference have the strong assumption that both the transmitting and receiving nodes can perceive the unwanted interference signal. This assumption does not hold here because hidden-terminal interference only affects the receiving node. Therefore, these counter-mechanism techniques are not directly applicable to avoid hidden-terminal interference, especially when it is intelligently engineered malicious hidden-terminal interference. In addition, in order to avoid unwanted interference (including the jamming attack), previous research in multi-channel wireless networks mostly adopt the channel-hopping (CH) strategy [51, 52, 53, 54, 55, 56, 57, 58, 59]. However, the enormity of densely deployed IoT networks that may coexist on the same channel, space, and time is not considered. The dense deployment of IoT devices will aggravate the hidden-terminal interference scenario; hence, it will create even more unwanted interference, which will have adverse effect on all the channels of the shared spectrum band. Therefore, new practical evasive strategies are desired to defend unwanted interference in multi-channel dense IoT networks. This dissertation proposes a Markov-based evasive strategy, namely Jump and Wobble.

1.3 Overview of the Proposed Research

Figure 1.6 shows an overview of the proposed research. It discusses three novel attack surfaces that are unique to the DSA and CR-based networks and proposes how these attack surfaces can be exploited.

At first, this dissertation introduces a new room of vulnerability in the conventional sensing approaches, where a perpetrator attacks only when no one is sensing the channel. This attack will decrease the channel utilization by victim SUs and could potentially create a Denial of Service (DoS) situation for victim SUs. This is named an off-sensing attack. This research also proposes an analytical model to analyze the impact of the off-sensing attack in a CRN. Numerical analysis and simulation results show that this attack possesses a serious threat to CRNs.

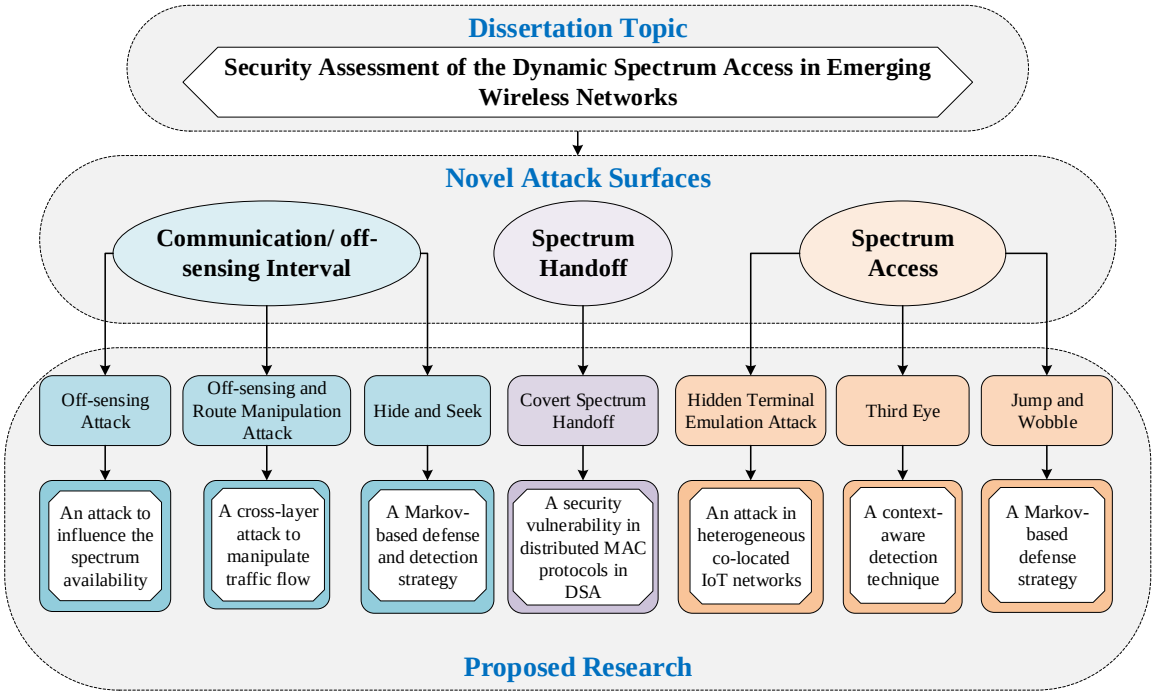


Figure 1.6: The overview of the proposed research.

Then, this dissertation introduces a new room of vulnerability in cross-layer routing protocols and demonstrates how a perpetrator can exploit this vulnerability to manipulate traffic flow around it. This research proposes the mentioned cross-layer attack in CR-based wireless mesh networks (CRWMNs), which is named an off-sensing and route manipulation (OSRM) attack. In this cross-layer assault, off-sensing attack is launched at the lower layers as the point of attack but the final intention is to manipulate traffic flow around the perpetrator. This dissertation also introduces a learning

strategy for a perpetrator, so that it can gather information from the collaboration with other network entities and capitalize this information into knowledge to accelerate its malice intentions. If conducted in a proper way, this attack would be far more detrimental than what we have experienced in the past and needs to be addressed before commercialization of CR-based network.

To realize the best strategy for off-sensing DoS attackers without having any prior knowledge, this research proposes a new random approach, the random-OS attack, which adapts to realistic scenarios and is difficult to detect using conventional techniques. Then, to counteract the off-sensing attack, a novel safeguard approach based on the Markov decision process to defend the proposed attack—namely hide and seek—is proposed. It also introduces an OS-attack detection strategy, which utilizes the sensing history to detect the presence of attackers without violating any policy or design constraints and without any networking overhead. This research advents a direction in designing safeguard strategies without amending the current FCC policies.

Afterwards, a vulnerability in the spectrum handoff process is introduced where spectrum handoff is an integral part of CR-based networks. It ensures the operational integrity of opportunistic spectrum access, the avoidance of harmful interference with licensed or primary users (PUs), and the delay requirement during a handoff. Understanding the significance of the spectrum handoff process, this research introduces a vulnerability and demonstrate how a selfish attacker can exploit this vulnerability to achieve personal gain. It is named as *covert spectrum handoff*. To the best of our knowledge, this is the first research to consider security aspects of spectrum handoffs and to introduce an attack in the proactive spectrum handoff process.

In security research, a detection or defense strategy is as strong as the attack model. Therefore, the instrumental step to devise an effective counter-mechanism technique is to devise a strong attack strategy. This research proposes an intelligent

attack model that exploits the natural hidden-terminal interference in coexistent IoT networks and that trade-offs between the attack performance (e.g., degrading victim's throughput) and the risk of exposure. In particular, the attack model lays out into two co-dependent multi-layer steps: (1) the reconnaissance and emulation phase that pertains to the smart-array antenna manipulation in the PHY layer and (2) the reactive interference phase that pertains to two different interference scenarios (based on different assumptions of the attacker's ability) in the MAC layer. This dissertation designs these steps by establishing analytical models, and the intelligent synthesis of these two steps will allow the attacker to impersonate a hidden-terminal.

Furthermore, we require a context-aware detection strategy to identify such attacks. Context-awareness helps an IoT device assess the in-hand information and deals with changes in the environment. To provide context-awareness, (1) this research proposes to design mathematical models that encompass the behavior of a benign hidden-terminal and a malicious hidden-terminal (or an HTE attacker). Based on this mathematical model and the observed behavior of a hidden-terminal, (2) this research proposes a signature-based detection and an anomaly-based detection; these context-aware detection strategies consider hidden terminals as both benign and malicious interference sources. The proposed detection strategy runs on the victim device and takes radio sensing and packet reception information into account to deduce the parameters of the introduced context-aware model.

Lastly, a defending IoT device must employ practical evasive strategies as a proactive measure to avoid HTE attacks. Therefore, this research proposes a safeguard approach to counteract the proposed HTE attack. In particular, this research introduces a Markov decision process (MDP) based safeguard strategy to thwart the HTE attack where a defender exploits the channel diversity in a multi-channel network by randomly hopping through different channels and exploits the proximity in dense IoT networks by diverting traffic through intermediate devices. Unlike the detection

strategy, the defense module runs on the transmitter of the affected link. As the transmitter has the primary ownership of a communication link, it is responsible for the successful reception of the packet. Thus, it will elude HTE attacks by taking the above mentioned proactive measures.

1.4 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, related work on the proposed research is introduced. In Chapter 3, the off-sensing attack is proposed and different strategies of an attacker is discussed.In Chapter 4, a cross-layer attack is proposed where off-sensing attack is launched in the lower layer. In Chapter 5, a counteract strategy is proposed to defend off-sensing attacks. In Chapter 6, a vulnerability in the spectrum handoff process is proposed. In Chapter 7, the hidden terminal emulation attack is proposed. The corresponding detection and defense strategies are proposed in Chapter 8 and Chapter 9, respectively. Following that, the publication and future work is listed in Chapter 10.

CHAPTER 2: RELATED WORK

This chapter discusses the related work in attacks and defenses in traditional wireless networks and more importantly in CR-based networks.

2.1 Existing Attacks in CR-based Networks

The presence of a PUE attacker can severely hamper the operations of a CRN. Therefore, new solutions to counteract such attacks are proposed. The idea of the PUE attack was first envisioned in [5], where the attacker aims to have a higher priority to access a vacant channel by imitating the signal characteristics of a PU. A defense strategy is proposed which utilizes the distance ratio and distance difference to detect the PUE attack. In [60], a location based transmitter verification scheme is proposed which verifies the transmission of a PU by estimating its location and transmission characteristics. In [61], an analytical model based on energy detection is proposed to calculate the probability of a successful PUE attack. In this approach, a lower bound on the probability of a successful PUE attack is obtained by using a Markov inequality approach. In [62], a received signal strength based defense approach is proposed, where SUs compare the received signal strength of a PU and an attacker. A cooperative spectrum sensing approach is proposed in [63], where the sensing information of different users is combined at a base station and the combined results are optimized to maximize the detection of PUE attacks. All the above mentioned counter-mechanism approaches consider that SUs would sense the signal of the attacker. However, an attacker can ingeniously avoid the sensing time of SUs and attack by interfering their transmission.

2.2 Existing Defenses in CR-based Networks

In [64], a cross-layer route manipulation attack is proposed where OS-DoS attack is utilized as a front-end attack to manipulate the traffic-flow in the network. A sweep jammer strategy is also proposed in [56] where jammers sweep through all channels to find the operating channel of any user. However, the proposed attack strategies are either ineffective in realistic scenarios or does not consider the DoS situation. Unlike previous research, this research devises a sophisticated attack strategy for OS-DoS attackers to adapt to realistic conditions.

A game theoretical approach has been proposed in [65, 66] to counteract PUE attacks by adopting a combination of extra-sensing and surveillance process. In [67], an MDP-based anti-jamming strategy is proposed to counteract jamming attacks in CRNs. A zero-sum Markov game is proposed in [55] and an optimal strategy to defend against reactive-sweep jammer is devised. In [56], an MDP-based strategy is proposed to thwart jamming attacks in multi-channel networks, where radios are equipped with in-band full-duplex capability. However, all these works neither consider an iterative attack model to prevent DoS attacks nor an attack detection model. In contrast, this proposal considers a more sophisticated attack model where the attacker can identify an individual victim's transmission and perpetrate a DoS attack on the victim, and our model can detect the presence of attackers.

2.3 Existing Cross-layer Attacks in CR-based Networks

In recent years, some cross-layer attacks have been proposed in the CR based networks. Cross-layer attacks have proven to be more detrimental than single-layer based attacks, due to their immunity to the single-layer based defense strategies. In [68], the coordination of two cross-layer attacks at the PHY layer and MAC layer is studied. The use of PUE attack as an auxiliary attack in order to degrade the throughput performance of TCP has been studied in [69]. In [70], the authors pro-

pose a MAC-TCP cross-layer attack where an attacker periodically preempts itself to use the shared channel and impacts the TCP performance by creating large variations in round-trip-time (RTT). Though the study of cross-layer attacks in terms of PHY-MAC-Transport layer has gained significant attention, very few efforts have been focused on security vulnerabilities in the network layer. A network layer attack in CR-based networks named routing-toward-primary-user (RPU) is proposed in [71], where a malicious node intentionally directs a large amount of traffic toward the PUs, aiming to cause interference to them. However, this is not a cross-layer attack and the perpetrator is an active participant in the attack, hence, less difficult to identify. In Hammer and Anvil attack [72], a jamming aided cross-layer attack is proposed in the multihop infrastructureless network. Nevertheless, a CR-based network is inherently immune to jamming attacks due to their ability to change operating channels dynamically.

Neither of the attacks mentioned above have considered an intelligent attacker who can gather information and learn about the whole network by leveraging the control information flowing in the collaborative CR-based network. With this knowledge, an attacker can conduct more sophisticated attacks with less risk of being flagged.

2.4 Existing Spectrum Handoff Techniques in CR-based Networks

In [20], a distributed proactive spectrum handoff process and a channel selection scheme are proposed. It considers most of the functionalities in the MAC layer. In [10], a Hidden Markov model-based prediction is used to provide a smart spectrum mobility scheme. It considers the idle duration of the channel and the reappearance probability of PUs on the channel to perform proactive handoffs. In [11], a voluntary spectrum handoff is proposed where SUs perform handoff voluntarily to reduce the handoff and channel selection delay based on probabilistic methods. In [14], a preemptive resume priority-based M/G/1 queuing model is proposed to minimize the total service time of SUs. Nonetheless, the queuing model is not distributed and

considers a central authority to maintain the queue. Moreover, the model does not consider a CCC and network coordination in the design. In [27], a distributed proactive spectrum handoff and channel selection method is proposed. It incorporates the channel rendezvous and the network coordination issue together in the spectrum handoff process. However, it does not consider collisions between SUs in multi-handoff scenarios.

Though all mentioned works contribute to the spectrum access in CRAHNs, no work has considered non-identical channels and the effect of non-identical channels in spectrum handoff decisions. In addition, security concerns of the proactive spectrum handoff process are overlooked.

2.5 Existing DoS Attacks in Wireless Networks

As discussed, an HTE attacker tries to emulate the radiation characteristics of a hidden terminal and creates a different physical scenario than the actual one. Thereby, a comparison is made to conventional location spoofing attacks in the localization paradigm, especially with received signal strength (RSS) based methods. In [73], it is experimentally shown that, by manipulating the RSS at the anchors, the localization method can be made futile. Directional antennas are exploited in [74], where the attackers have the ability to bias the location estimation to a direction of their choice. In [75], a mathematical analysis of beamforming-based perfect location spoofing against RSS-based localization techniques is proposed, where an attacker mimics the path-loss signature at the anchor nodes to manipulate the results of RSS-based localization algorithms. The vulnerability of WLAN-based Skyhook positioning system [76] is investigated in [77] where authors demonstrated the susceptibility of Skyhook against location spoofing attacks. However, location spoofing is more challenging in an exponentially denser network environment. Unlike previous works, this research addresses these challenges and formulate a mathematical model to test the feasibility of the HTE attack.

2.6 Existing Detection Strategies against DoS Attacks in Wireless Networks

In wireless networks, jamming is one of the well-researched attacks. The detection of traditional jamming attacks has been extensively studied in [78, 79, 80, 81, 82, 47, 83, 84, 85, 43, 86, 42]. In [78], the influence of different jamming strategies on the PDR and the RSS of network links is analyzed and a thresholding algorithm is proposed. In addition to the PDR and the RSS, the channel busy ratio and the number of retransmission attempts are employed in [80, 81], and a machine learning based technique is proposed to detect jamming attacks. Jamming attacks in time-critical networks are studied in [82], and numerical results on the impact of jamming on the network message invalidation ratio is presented. Moreover, in [87], an anomaly-based detection technique is proposed to detect anomalous behaviors of external neighboring nodes in dense IoT scenarios. An approach based on group testing to identify which node triggers the reactive attack is proposed in [47], in wireless sensor networks. In [48, 49, 50, 30], the impact of jamming attacks on the theoretical performance of IEEE 802.11 networks is presented and analyzed for different types of jamming strategies; these theoretical analyses are based on Bianchi's Markov chain model of 802.11 distributed coordination function (DCF) [88]. In CR-enabled networks, DoS attacks are studied in [89, 64, 90, 91, 29], where the attacker attacks in the off-sensing interval and creates an illusion of PU reappearance to force the victim out of its current operating channel. In [84], a mathematical model of an optimal jamming strategy is proposed, where an attacker can regulate its jamming probability to trade-off between the reward of jamming and the penalty of exposure.

Although there is no direct comparable work to compare HTE with (except an earlier work [87]), differences between existing work on jamming and our proposed research can be noted as follows. Interestingly, [84] considers the slotted Aloha protocol, which does not incorporate the carrier sensing multiple access (CSMA)—an essential tool in modern wireless networks; in contrast, our research is based on the

widely accepted CSMA approach. While the influence of *in network* hidden terminal interference is considered in [92], it did not explain how a reactive attacker can listen to the transmission of its hidden counterparts; in contrast, this research captures the impact of hidden terminal interference from *external networks*, based on the carrier sensing, and this research proposes how an HTE attacker can listen to its hidden counterparts via antenna manipulation. Moreover, though an anomaly-based detection technique is proposed in [87], it fails to efficiently identify HTE attacks when there are multiple anomalies in the network. In summary, compared to all these prior works, this research addresses the hidden terminal interference issue among different co-located networks/technologies, and an attack model is devised based on this. As prior work on attack detection mostly depends on the network performance and does not consider hidden terminals as benign interference sources (except [92, 87]), they may mis-categorize hidden terminals as reactive attackers. This research considers hidden terminals as benign interference sources, address the way attackers can inappropriately use it for malice intentions, and propose a signature-based context aware detection model to uniquely identify HTE attacks.

2.7 Existing Defense against DoS Attacks in Wireless Networks

The security research community has discovered numerous vulnerabilities and proposed their defenses in IoT [93, 94, 95, 96, 97]. In [98], a distributed DoS attack is studied where Mirai botnet was used to compromise 0.6 million IoT devices. Honeywell home controllers are shown to be vulnerable to a pair of bugs in their authentication system [99]. In a recent work [100], it is demonstrated that home assistant devices can be compromised by an attacker using inaudible voice commands. A large-scale coordinated attack on the power grid is shown in [101] where attackers can compromise high wattage devices to manipulate the load demand and create blackouts. Yet, these works focus on upper-layer vulnerabilities only. In contrast, this research studies the vulnerabilities caused by the changes in lower-layers in dense IoT networks

under shared spectrum operation. In addition, unlike [35, 87], this research proposes a constrained attack model and designs a safeguard strategy against the HTE attack.

Note that, in contrast to traditional jamming attacks, the HTE attack does not rely on a strong noise signal to corrupt the wireless reception of the victim. Instead, it exploits the proximity to the victim and utilizes regular data transmissions to corrupt the victim's reception.

CHAPTER 3: PROPOSED OFF-SENSING ATTACK IN CR NETWORKS

This chapter proposes a new attack in CR-based networks that exploits the off-sensing interval in periodic channel-hopping processes to influence the spectrum availability of victims.

3.1 Network Coordination Scheme

In the absence of a CCC, rendezvous is a pre-requisite step to establish a successful handshake between two SUs. After the rendezvous, the two SUs can initiate data transmissions. In [17, 18, 19], a successful rendezvous is achieved when a request-to-send and clear-to-send (RTS/CTS) exchange on a common available channel of two SUs is completed. Throughout this chapter, the term "rendezvous" and "RTS/CTS exchange" is used interchangeably.

This chapter considers the common frequency-hopping strategy as the network coordination scheme [102]. Figure 3.1 illustrates the operation of a conventional common frequency-hopping approach, where the system is time slotted and SUs communicate with each other in a synchronized manner. When there is no packet to transmit, a SU keeps hopping through all the channels in the band (i.e., the hopping pattern periodically goes through 1, 2, $\cdots$, M, where M is the total number of channels).

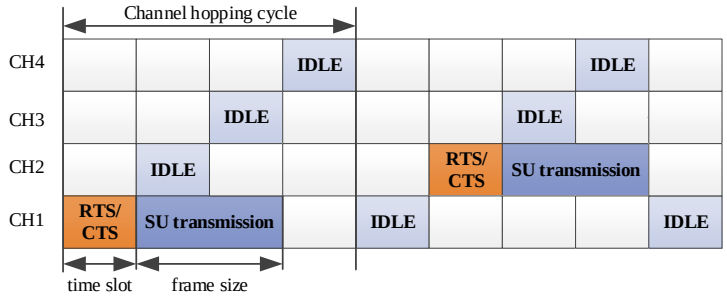


Figure 3.1: An example of conventional network coordination scheme.

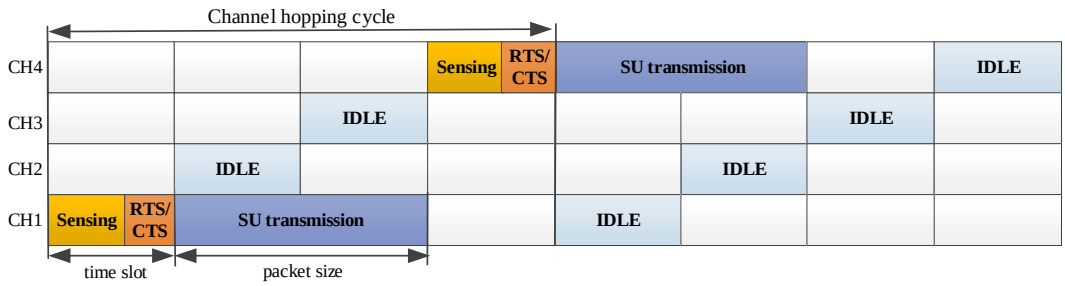


Figure 3.2: The proposed network coordination scheme.

Whenever it has a packet to send, it sends an RTS message in each time slot and waits for the CTS from the intended destination SU. If an RTS/CTS exchange fails in a time slot, the SU sender continues this process in the next time slot. After a successful RTS/CTS exchange between a SU sender and receiver, they stop channel-hopping and start data transmissions on the same channel. Meanwhile, other non-transmitting SUs continue the channel-hopping. After finishing the transmission, both SUs continue the channel-hopping by following the common hopping sequence. In [26], a time slot is defined as the transmission time of an RTS and a CTS. This research considers that every transmission of SUs and collision with PUs only happens at the beginning of a time slot, and ends at the end of a time slot. Throughout this chapter, it is considered that a SU can always detect a PU (i.e., no mis-detection) and does not trigger false-alarms.

In an effort to rendezvous, RTS/CTS packets may collide with an ongoing PU transmission and cause interference to PUs. Therefore, we require a new channel access mechanism where the rendezvous process is preceded by channel sensing. This research proposes a modified network coordination scheme where any transmission attempt is preceded by channel sensing. In Figure 3.2, each time slot is considered as the time to perform sensing and the transmission of an RTS/CTS pair. Throughout this chapter, it is considered that N number of SUs form a cognitive radio ad-hoc network and try to opportunistically access M identical licensed channels. To better illustrate the activity of a SU under the proposed network coordination scheme, Figure

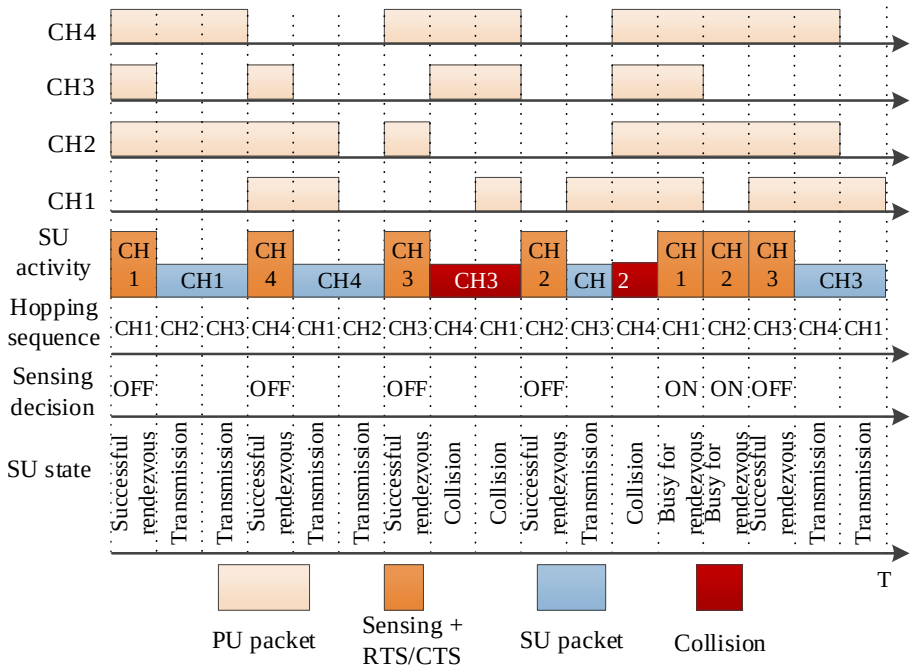


Figure 3.3: An illustration of the proposed network coordination scheme with an ON/OFF PU model.

3.3 provides an example. In the example, the SU always has a packet to send.

3.2 Proposed Analytical Model

A multi-dimensional discrete-time Markov model to analyze the network performance in the absence and presence of our proposed *off-sensing* attack is proposed. For simplicity, this research ignores the propagation delay, processing time, and collision between SUs in the model. It also assumes that the receiver is always available to receive a packet, that is, the destination of a packet is always available.

From the above descriptions of spectrum sensing and rendezvous under our proposed network co-ordination scheme, each SU has the following different states:

- *Idle*: when a SU has no packet to transmit.
- *Busy for rendezvous*: when a SU has a data packet to transmit but cannot find an available channel to rendezvous with its destination. Until the SU achieves a successful rendezvous, the packet will be kept in the buffer of the SU.

- *Successful rendezvous*: when a SU has a data packet to transmit and achieves a successful rendezvous with the destination on a common available channel.
- *Transmission*: when the transmission of a SU does not collide with a PU's transmission in a time slot.
- *Collision*: when the transmission of a SU collides with a PU's transmission in a time slot.

An SU sender learns whether a transmission is successful or not at the end of the packet transmission (by receiving an acknowledgment from the destination SU). After an unsuccessful attempt of transmission due to collision, the SU will try to retransmit. It keeps retransmitting until the maximum number of retransmission attempts exceeds and then it drops the packet.

Based on the above assumptions, the state of a SU at time slot t is defined as $[A_t(t), A_c(t), A_a(t), A_b(t)]$. Descriptions of the state variables are:

$A_t(t)$- denotes the number of time slots including the current time slot that have been used for successful transmission. This value goes from 0 to h.

$A_c(t)$- denotes the number of time slots including the current time slot that have encountered collisions with a PU. This value goes from 0 to h.

$A_a(t)$- represents the number of transmission attempts for the current packet. This value goes from 0 to m.

$A_b(t)$- represents the deferral in data transmission due to unsuccessful rendezvous. This value goes from 0 to 1. "1" means that there is a packet in the buffer and waiting for a successful rendezvous, and "0" means that a successful rendezvous has happened. In *Transmission* and *Collision* states, this value always remains 0.

The notations used in our proposed Markov model are listed in Table I.

In Figure 3.4, ($A_t(t)$=0, $A_c(t)$=0, $A_a(t)$=0, $A_b(t)$=0) represents the *Idle* state of a SU. Similarly, ($A_t(t) \in [1, h]$, $A_c(t)$=0, $A_a(t) \in [1, m]$, $A_b(t)$=0) represents the *Transmis-*

Table 3.1: Notations Used in the Markov Model

p	Probability that a channel is available in a time slot
p_{at}	Probability that a channel is available in a time slot under attack (under no attack, p_{at}=p)
s	Probability that a SU packet arrives in a time slot
h	The length of a SU data packet in terms of time slots
m	Maximum number of transmission attempts before dropping a packet

sion states, ($A_t(t)\in[0, h-1]$, $A_c(t)\in[1, h]$, $A_a(t)\in[1, m]$, $A_b(t)$=0) represents the *Collision* states, ($A_t(t)$=0, $A_c(t)$=0, $A_a(t)\in[1, m]$, $A_b(t)$=1) represents the *Busy* states for unsuccessful rendezvous, and ($A_t(t)$=0, $A_c(t)$=0, $A_a(t)\in[1, m]$, $A_b(t)$=0) represents the *Successful rendezvous* states.

3.3 Steady-State Probabilities

To derive the steady-state probabilities of each state, first the single-step transition probabilities are calculated. Here, (i, j, k, l) and ($A_t(t)$=i, $A_c(t)$=j, $A_a(t)$=k, $A_b(t)$=l) is used interchangeably, where $i\in[0, h]$, $j\in[0, h]$, $k\in[0, m]$, $l\in[0, 1]$. The single step transition probability from time slot t to $t+1$ represented as $P(i_{t+1}, j_{t+1}, k_{t+1}, l_{t+1}|i_t, j_t, k_t, l_t)$. The single-step transition probabilities are given as follows. Here $i'\in[1, h]$, $i''\in[0, h-1]$, $j'\in[2, h]$, and $k'\in[1, m]$.

$$
\begin{aligned}
P(0,0,0,0|0,0,0,0) &= 1-s \\
P(0,0,1,1|0,0,0,0) &= s(1-p) \\
P(0,0,1,0|0,0,0,0) &= sp \\
P(0,0,k',0|0,0,k',1) &= p \\
P(0,0,k',1|0,0,k',1) &= 1-p \\
P(i',0,k',0|i'-1,0,k',0) &= p_{at} \\
P(i'',1,k',0|i'',0,k',0) &= 1-p_{at} \\
P(i',j',k',0|i',j'-1,k',0) &= 1 \\
P(0,0,k',1|i'',h-i'',k'-1,0) &= 1-p \\
P(0,0,k',0|i'',h-i'',k'-1,0) &= p \\
P(0,0,0,0|h,0,k',0) &= 1-s \\
P(0,0,1,1|h,0,k',0) &= s(1-p) \\
P(0,0,1,0|h,0,k',0) &= sp \\
P(0,0,0,0|i',h-i',m,0) &= 1
\end{aligned}
\tag{3.1}
$$

Now, the steady-state probabilities of each state can be derived.

Though sensing can guide a SU to make a decision on the channel availability, it cannot guarantee collision-free transmissions between SUs and returning PUs on that channel. Therefore, the probability of channel availability in a time slot is always less than one (i.e., $0 \leq p < 1$). It implies that the steady-state probability of the *Collision* state is non-zero. In the following, the steady-state probabilities are derived in terms of the first *Rendezvous* state $P(0,0,1,0)$.

First, *Successful rendezvous* states,

$$
P(0,0,k,0) = (1-p_{at}^{h})^{k-1} P(0,0,1,0), \tag{3.2}
$$

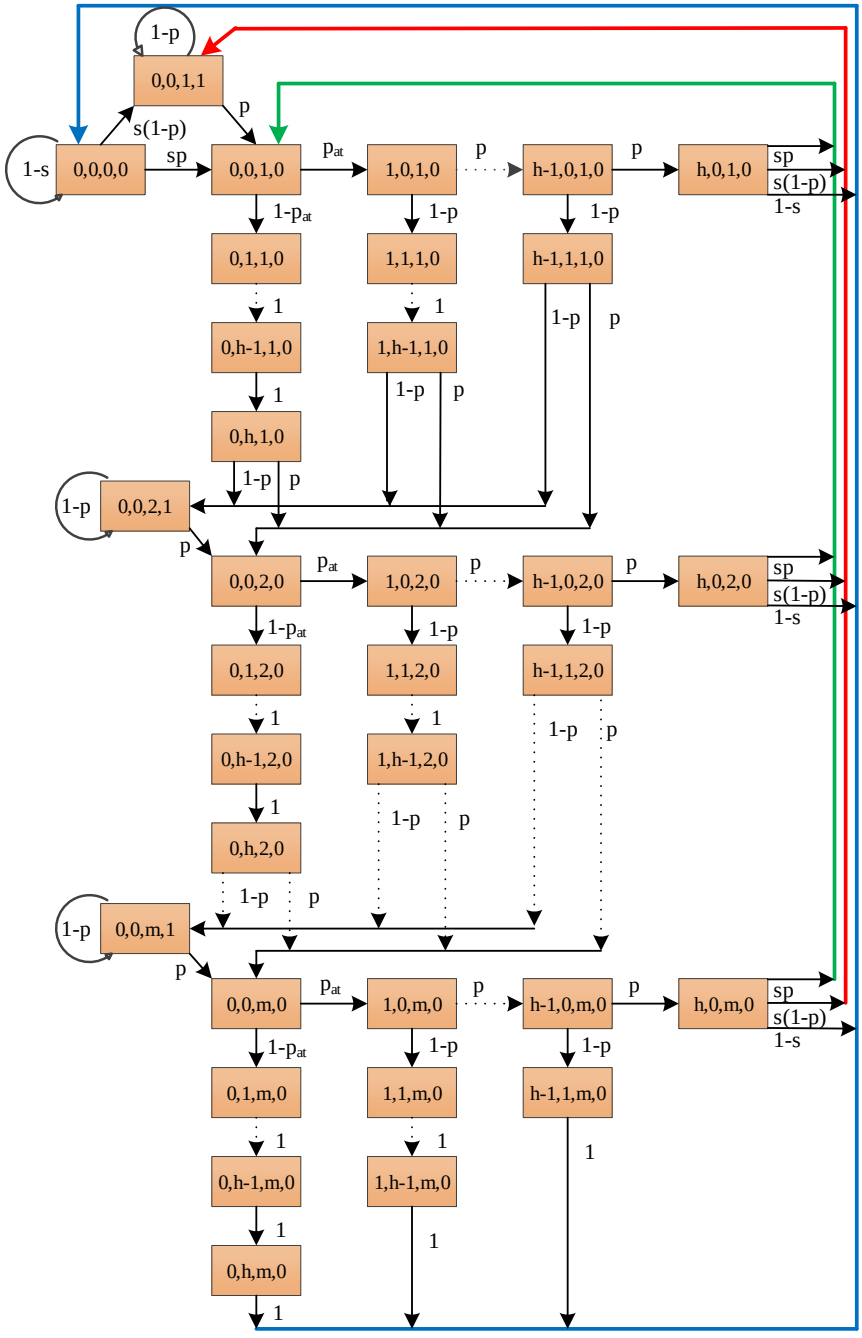


Figure 3.4: The proposed multi-dimensional Markov model.

where $k > 1$.

Second, *Busy* states for rendezvous,

$$P(0,0,k,1) = \frac{1-p}{p}(1-p_{at}^{h})^{k-1}P(0,0,1,0), \tag{3.3}$$

where $k > 1$.

Third, *Transmitting* states,

$$P(i,0,k,0) = p_{at}^{i}(1-p_{at}^{h})^{k-1}P(0,0,1,0), \tag{3.4}$$

where $1 \leq i \leq h, 1 \leq k \leq m$.

Fourth, *Collision* states,

$$P(i,j,k,0) = p_{at}^{i}(1-p_{at})(1-p_{at}^{h})^{k-1}P(0,0,1,0), \tag{3.5}$$

where $0 \leq i \leq h-1$, $1 \leq j \leq h$, $1 \leq k \leq m$, and $i+j \leq h$.

Lastly, *Ideal* state,

$$\begin{aligned} &P(0,0,0,0) = \\ &\frac{1}{s}\left[(1-s)\{1-(1-p_{at}^{h})^{m}\} + (1-p_{at}^{h})^{m}\right] P(0,0,1,0). \end{aligned} \tag{3.6}$$

Since $\sum_{i=0}^{h}\sum_{j=0}^{h}\sum_{k=1}^{m}\sum_{l=0}^{1} P(i,j,k,l) = 1$, the steady-state probabilities of every state in the proposed Markov model can be deduced. Here, the summation of the steady-state probabilities of all the *Transmission* states represents the normalized throughput of a SU:

$$\text{Normalized Throughput} = \sum_{k=1}^{m}\sum_{i=1}^{h} P(i,0,k,0). \tag{3.7}$$

In this chapter, a homogeneous traffic characteristic for each PU channel is considered. Without loss of generality, each PU is designated to a unique channel and

model the activity of each PU as an ON/OFF process [103, 104, 105, 106]. SUs can only access the channel if the PU is in the OFF state while sensing. It assumes a buffer for each PU where it can hold at most one packet while transmitting and it keeps the packet in the buffer until the packet is successfully transmitted. Hence, the OFF period of a PU follows the geometric distribution, under which the probability mass function (pmf) can be expressed as

$$\Pr(N_{OFF} = n) = p^n(1-p), \tag{3.8}$$

where N_{OFF} is the number of time slots of an OFF period and p is the probability that a channel is available in a time slot.

3.4 Adversary Model

Conventional PUE attacks are usually carried out by two types of users: selfish user, whose purpose is to maximize its own spectrum availability and throughput, and malicious user, who has an ill-intention of sabotaging the network operation and making it unavailable for the victim nodes. In our proposed *off-sensing* attack, the perpetrator's goals remain the same. However, the approach a perpetrator takes to achieve these goals is different. It does not emulate PU signals, rather transmits concurrently with the victim SU, to create interference and mislead the victim's belief. To perform this attack, perpetrators must have the knowledge of the channel-hopping sequence and the sensing schedule of the victim SUs. As the perpetrator is a legitimate SU inside the network, it has the knowledge of the channel-hopping sequence.

In [107], a window-based transport protocol for CR ad-hoc networks is proposed to perform end-to-end packet delivery. In this protocol, SUs exchange their physical and MAC layer information which includes the sensing schedule. In addition, in IEEE 802.22, the quiet period is used to perform reliable sensing. During a quiet period, all the nodes sense the spectrum and do not transmit. In both cases, a perpetrator can

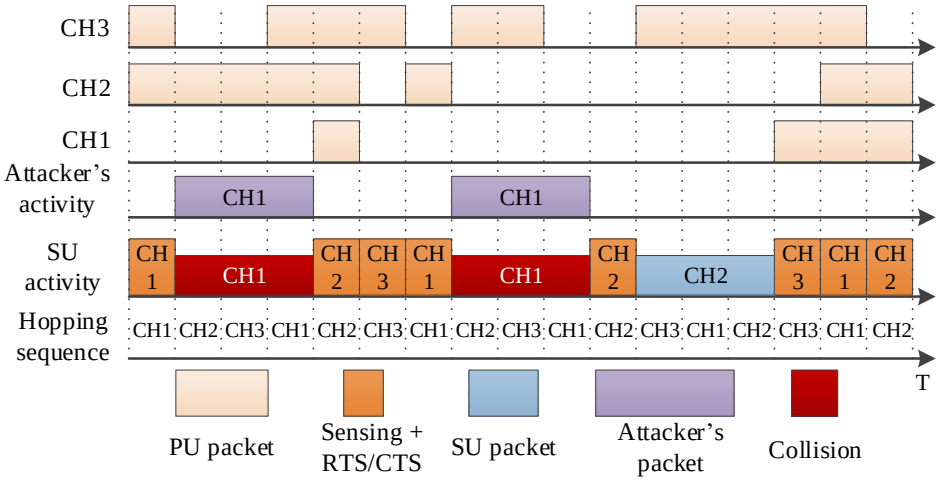


Figure 3.5: An illustration of the proposed attack in scenario-1.

easily learn the sensing schedule of all the nodes and conduct attacks in off-sensing intervals. As the victim cannot detect the transmission, it will think it is interfering with PU's transmission. Thus, according to the FCC regulation, the victim SU will leave the channel within 2 seconds [21] and perform a spectrum handoff.

Now, some probable scenarios of these attacks are proposed:

Scenario-1: In this scenario (Figure 3.5), a perpetrator tunes to a certain channel of interest and keeps using it by avoiding the sensing intervals of neighboring SUs. When a benign SU has a packet to send, it exchanges RTS/CTS packets with its destination SU. Because RTS/CTS can be heard by the perpetrator, whenever it overhears any RTS/CTS exchange, it performs attacks and keeps doing it on that particular channel. The goal of this attack is to selfishly use the channel for its own advantage. In this scenario, there is no particular victim SU. Whoever tries to access the channel becomes a victim.

Scenario-2: In this scenario (Figure 3.6), a perpetrator attacks an individual SU to sabotage its network operation by creating a Denial-of-Service (DoS) situation. As a perpetrator has the knowledge of the channel-hopping pattern and sensing schedule, it can easily infer every transmission attempt of the victim SU from overhearing RTS/CTS messages. After an unsuccessful attempt of transmission, the victim

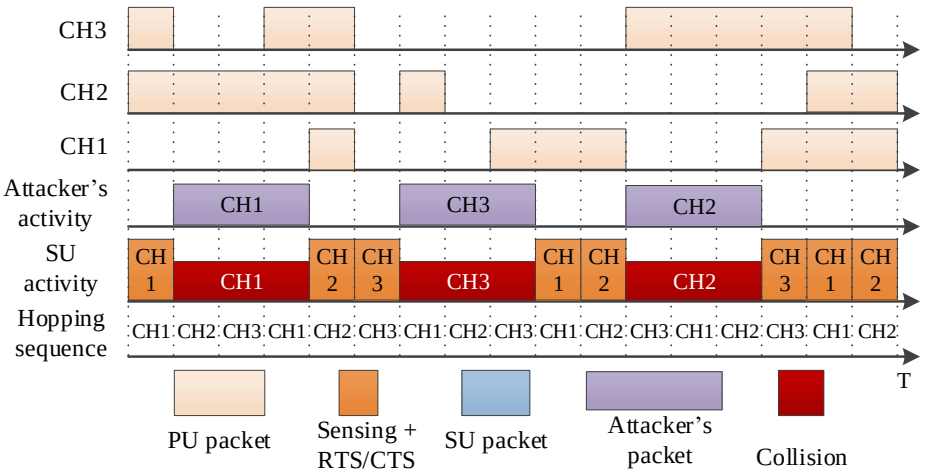


Figure 3.6: An illustration of the proposed attack in scenario-2.

will hop to a new channel (a new channel will be selected according to the hopping sequence) and try to retransmit the packet. However, as the perpetrator has the knowledge of the channel-hopping sequence, it can figure out the new channel and attack again. The perpetrator will keep doing it and create a DoS situation for the victim SU. As a result, the victim SU's operations will stop.

Scenario-3: In this scenario, multiple malicious nodes collude to sabotage the network operation. Each malicious node tunes to a designated channel and attacks whenever any benign SU tries to access the channel. When the malicious nodes attack on every channel in the band, they can create a network-wide DoS within their attack range.

The disruptive nature of this attack can engender a fatal impact where, impact on the physical layer (i.e, DoS) can also affect the functionality of upper layers.

3.5 Performance Evaluation

In this section, the proposed *off-sensing* attack is evaluated under different scenarios by numerical analysis, and confirm our analysis by conducting simulations. Parameters used in our simulation are listed in Table II. Each PU is assigned to a designated channel. Four scenarios are considered in this simulation. *No attack*, rep-

resents normal network condition under no *off-sensing* attack. *Attack-10%*, *Attack-30%*, and *Attack-50%*, represent the scenario where 10%, 30% and 50% of the total channels are under attack, respectively.

Table 3.2: Simulation Parameters: Off-sensing Attack

Simulation area	500x500
Simulation time	10000 time slots
SU sensing range	50
The number of PUs	10
The number of SUs	400
Channel data rate	2 Mbps
The size of (RTS+CTS)	160 + 112 bits (802.11b/g)
Sensing duration	1 ms (802.22)
Maximum no. of transmission attempts	3
SU traffic	Saturated (i.e., s=1)
SU packet size	1 time slot (284 byte)
Number of channels	10

3.5.1 Throughput Performance

To understand the impact of the attack on throughput performance, we calculate the average normalized throughput of all SUs. It is shown in Fig. 3.7(a), the simulation results match well with the numerical results obtained through our proposed analytical model with the average difference only 3%, 4%, 7%, and 7% for *no-attack*, attack-10%, attack-30%, and attack-50% case, respectively. We can also observe the throughput performance under different percentage of attacked channels. As the percentage of attacked channels (or attackers) increases, throughput decreases significantly.

3.5.2 Collision Probability and Packet Drop Rate

Based on the proposed Markov model, the collision probability between SUs and PUs is the summation of all the steady-state probabilities of the *Collision* states. From Fig. 3.7(b), it is observed that the collision probability decreases as the channel availability increases. In the absence of a PU, the collision probability goes to zero in the *no attack* scenario, whereas collision still persists under attacks. This influx of

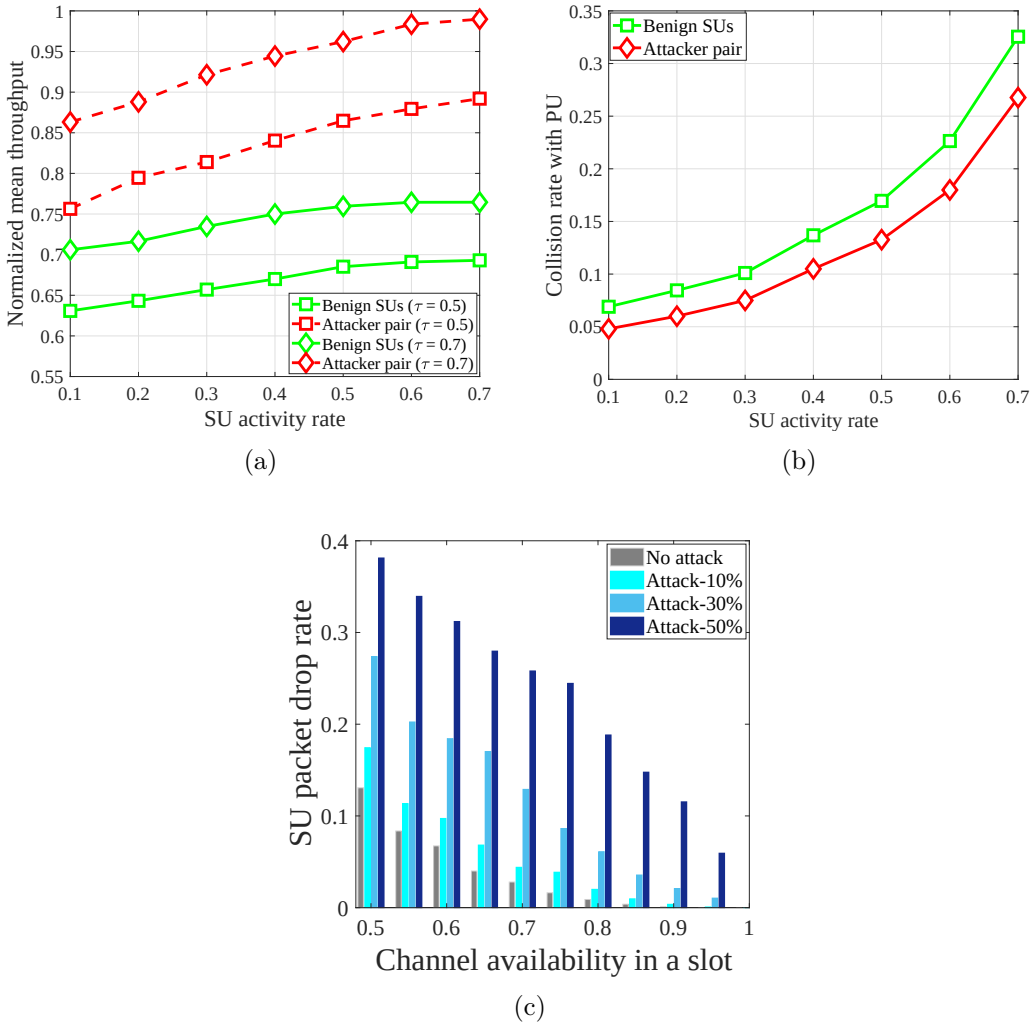


Figure 3.7: Impact of off-sensing attack on (a) normalized throughput, (c) SU and PU collision probability, and (c) packet drop rate.

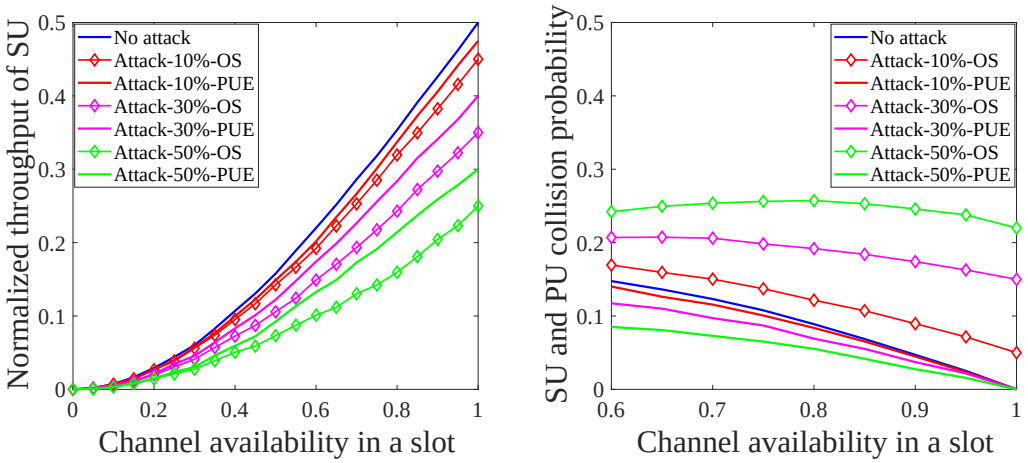


Figure 3.8: Performance comparison between off-sensing and PUE attack (a) normalized throughput and (b) SU and PU collision probability.

collisions are generated from the collision between victim SUs and attackers.

With increasing collisions, packets start to drop more. The trend of packet drop is shown in Fig. 3.7(c). We can observe that the impact of this attack is significant, even when the channel availability is high.

3.5.3 Performance Comparison between Off-sensing and PUE Attack

We compare the performance between off-sensing attack and PUE attack under similar boundary conditions. In off-sensing attack, nodes will waste more time on collisions due to the collisions between victims and attackers. Whereas in PUE attack, nodes will not experience such extra collisions. In Fig. 3.8(a), we can observe the difference in throughput between both attacks.

As SUs will encounter extra collisions in off-sensing attack, the collision probability increases (Fig. 3.8(b)). Moreover, it is interesting to see that the collisions in PUE attack reduces more than under *no attack* scenario. As PUE victims have to defer their transmission attempts more often than the *no attack* case, they have less room for transmission and hence less collisions.

CHAPTER 4: PROPOSED OFF-SENSING AND ROUTE MANIPULATION ATTACK IN CR-BASED WIRELESS MESH NETWORKS

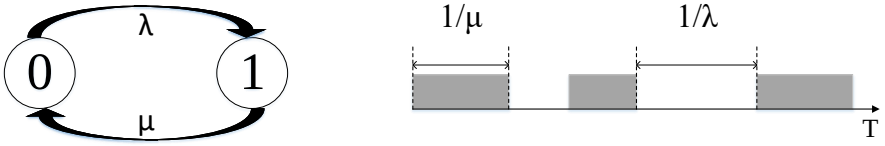
This chapter illustrates a cross-layer attack model where the off-sensing attack is exploited as a front-end attack to manipulate the traffic flow in CR-based wireless mesh networks.

4.1 System Model

In this section, we provide an outline of the assumptions made for the basic functionalities of the PHY, MAC, and network layers in our considered CR-WMNs.

4.1.1 Primary User and Secondary User Model

This chapter considers totally M homogeneous channels each with a fixed bandwidth for the PUs and SUs in the network, and N CR-WMRs trying to opportunistically access the channels. Each PU randomly selects a channel to access. An SU is allowed to access a channel when it senses no PU is using it. During the transmission, if an SU senses the channel busy, it stops transmitting on that channel and performs a spectrum handoff. Each SU is equipped with only one radio for spectrum sensing, control information exchange, and data transmission. Each PU alternates between the ON and OFF state according to a continuous-time Markov process. In Figure 4.1, let λ denote the transition rate from the OFF to ON state, and let μ denote the transition rate from the ON to OFF state. Thereby, the mean sojourn time in the ON and OFF state is $1/\mu$ and $1/\lambda$, respectively, and both follow the exponential distribution.



(a) Transition rate of the Markov chain.

(b) PU activity in the time line.

Figure 4.1: PU activity model.

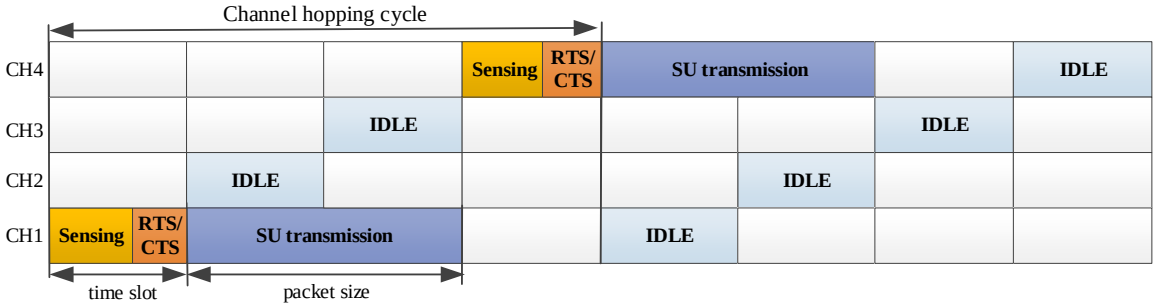


Figure 4.2: Network coordination scheme

4.1.2 Network Coordination Scheme

Rendezvous is a pre-requisite step before two SUs can communicate and exchange control information with each other in the absence of a dedicated CCC. A successful rendezvous happens when both transmitting and receiving SUs are on the same channel and have completed a successful handshake between them, e.g., a Request-to-Send/Clear-to-Send (RTS/CTS) exchange.

The common frequency-hopping as the network coordination scheme [26, 20] is considered, which means that the channel-hopping pattern is the same for all SUs. Figure 4.2 illustrates the operation of the common frequency-hopping-based network coordination. We consider a time-slotted system. Each time slot consists of a sensing interval (sensing) and a contention interval (CI) with the transmission of an RTS/CTS pair. When there is no packet in the buffer of an SU, it keeps hopping through the channels from one time slot to another based on the predetermined common channel-hopping pattern.

The MAC model is adopted from [28, 108] for network coordination. Whenever

a SU has a packet to send, it first senses the channel. If the channel is idle, the SU chooses a random number between 0 and $CW - 1$ (in terms of mini-slots) as its backoff time to avoid contention on the channel. If it hears no RTS before the backoff time runs out, it sends an RTS on the channel. Otherwise, it saves the remaining time in the backoff timer and will try to resend the RTS in the next time slot. After sending an RTS, the source SU waits for the CTS from the intended SU receiver. If the RTS sender fails to receive a CTS, it means the RTS/CTS exchange has failed in this slot and the source SU will continue the same process in the next time slot. After a successful RTS/CTS exchange, both SUs stop channel-hopping and start the data transmission on the same channel. After a successful transmission, both SUs start channel-hopping again by following the common hopping sequence. Meanwhile, all other SUs keep hopping through the channels. To better illustrate the activity of a SU under the coordination scheme,an example in Figure 4.3 is provided. In this example, the SU always has a packet in its buffer, wins contentions and a SU packet length is two time slot long.

4.1.3 Routing Scheme

Many routing protocols have been proposed for CR-based networks[109, 110, 111, 112]. In all these papers, spectrum availability has been given the highest weight for routing decisions. Therefore, it is clear that CR-based routing protocols consider spectrum availability as a significant cost metric.

This chapter does not focuses into proposing a new routing protocol. Instead, it adopts a link-state based routing protocol with channel availability as the only cost metric for routing decisions. The goal is to show the impact of the proposed attack on routing performance. In our CR-WMN, CR-WMRs calculate their link cost periodically with a period of 'Δ' and broadcast it. We also define an activity threshold τ (in Δ interval) above which a PU will be considered busy and hence the channel is not available. Along with cost, nodes also share their available channel

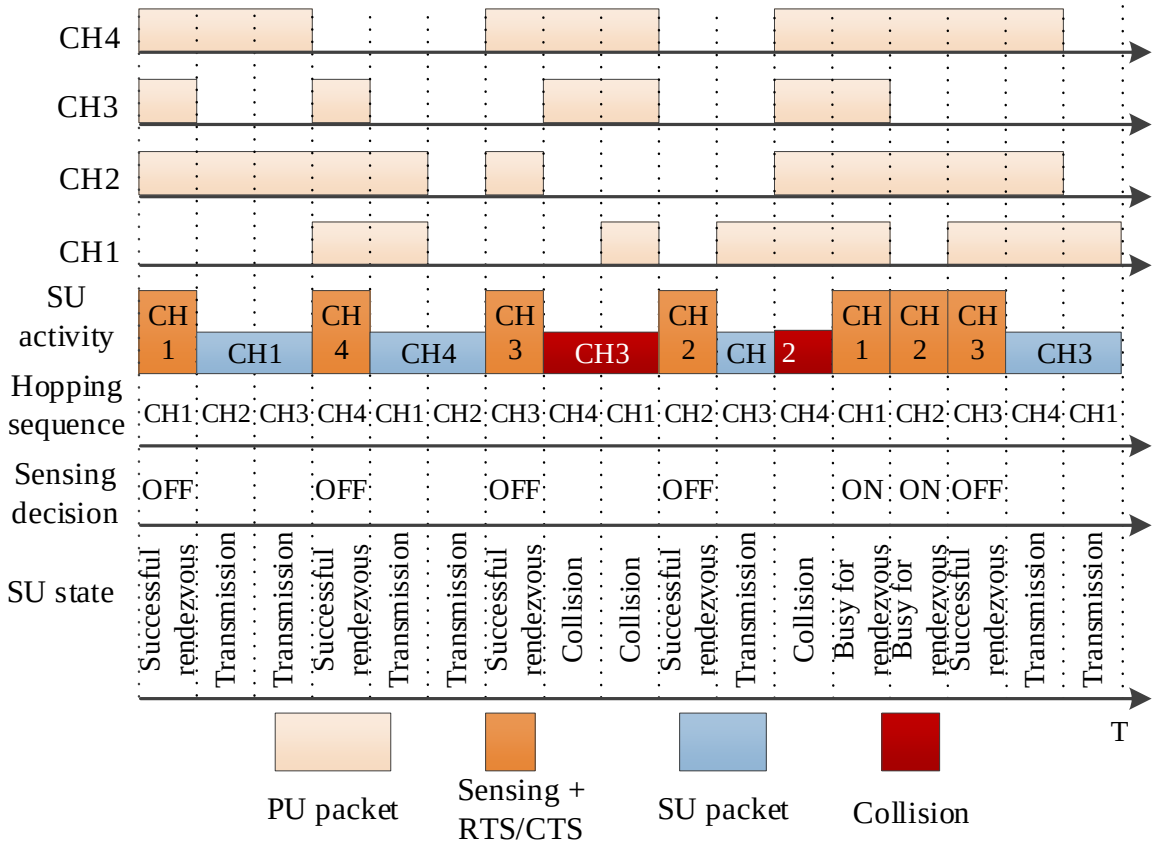


Figure 4.3: An illustration of the network coordination with an ON/OFF PU model.

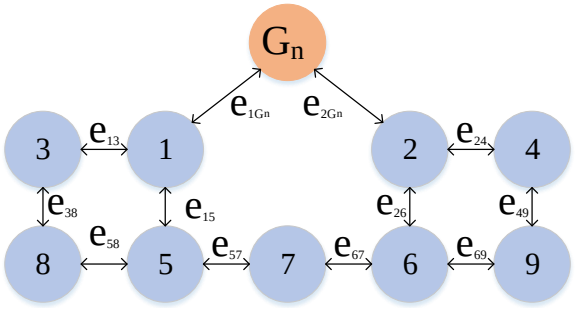


Figure 4.4: An illustration of a network graph.

list (ACL). For the calculation of the shortest path from a CR-WMR to the gateway, we consider the CR-WMN as an undirected graph $G = \{V, E\}$, called a connectivity graph. Each node $i \in V = \{1, \cdots, N\}$ represents a CR-WMR, which is characterized by a circular transmission range and an interfering range. Each edge E represents the connectivity between neighboring CR-WMRs and the edge cost is characterized by the spectrum availability. Figure 4.4 illustrates a network graph with 9 nodes and a gateway (G_n). Link cost between node i and j is defined as e_{ij}.

4.2 Proposed Off-Sensing and Route Manipulation (OS-RM) Attack Model

In reality, it is very unlikely for one to take control of a significant portion of the CR-WMRs in a CR-WMN without being flagged. However, under our proposed attack model, without even compromising a significant amount of routers, the perpetrator can still have the control over a significant portion of traffic flow around him. This can be done by exploiting and taking advantage of the many cross-layer routing protocols in CR enabled networks, where affecting lower layers can result in influencing decisions in the network layer.

The configuration of the proposed HMM-based system for the OS-RM attack is shown in Figure 4.5. Time is slotted into a duration of routing updates Δ. Therefore, we consider a discrete-time model, where the time variable takes values in {0, 1, ..., T}. The attacker has a separate HMM block for each channel. The input to the system at time t consists of the routing updates received from the neighboring nodes.

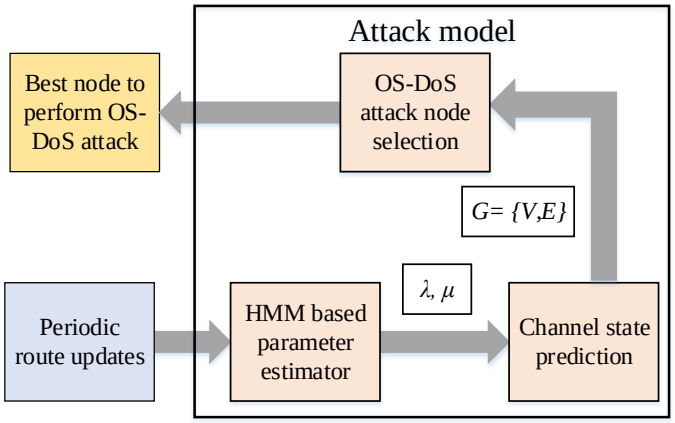


Figure 4.5: Proposed attack model.

The attacker model consists of three components: OS-DoS attack node selection, channel state prediction, and HMM-based channel parameter estimator. The OS-DoS attack node selector chooses the best node as the victim node based on the updated network graph $G = \{V, E\}$. The output of the system consists of the best neighbor to perform the OS-DoS attack, in order to divert traffic flow through the target node. The attacker updates the network graph G depending on the adjacency list (i.e., neighboring list of the SUs) and prediction of the future state of the channels, and the HMM-based channel parameter estimator facilitates to estimate the channel activity based on the routing updates.

We consider the frequency of routing updates comparable to the frequency of channel status change. Also, due to computational and physical efforts by the attacker, we consider a constant delay between when the routing update arrives and the attacker conducts an OS-RM attack without learning. We will see that this delay degrades the attack performance and hence it indicates the importance of predicting network conditions beforehand to counteract the effect of the delay.

4.2.1 OS-DoS Node Selection

The victim of the OS-DoS attack will be disconnected from the network (or has a very high cost to use it) and traffic flows that have been going through it, will switch

to the next best available route. The performance of the OS-RM attack depends on the right neighbor node to perform the OS-DoS attack on. Depending on the predicted network graph, the attacker finds the neighboring node whose traffic flow is most likely to traverse through the target node, if attacked. Here, the attacker's goal is to choose a neighbor in such a way that the rebound effect will divert most traffic flows to the target node.

The attacker will use a shortest-path algorithm (i.e., Dijkstra's algorithm) to figure out the best route for each node in the network to reach the gateway. At every step, the attacker first calculates the number of routers choosing the target router as a forwarder, under no attack (i.e., successor routers, π_{max}). Then, it finds the best neighbor router to perform the OS-DoS attack which will maximize its objective. It does it by measuring what would happen if it attacks a neighbor. If there is no neighbor that offers $\pi > \pi_{max}$, it will not conduct the OS-DoS attack and wait for the next update to come. Here, π is the number of successor nodes, under attack. Algorithm 1 and 2 show the pseudocode for calculating the successor CR-WMRs of the target CR-WMR and OS-DoS node selection, respectively. Next, we will discuss how an attacker can update the network graph G. Here, the target node and the gateway node are denoted as T_n and G_n, respectively.

Algorithm 1 Calculating the number of nodes that has the target node in their forwarding set to the gateway

Input: G, T_n, G_n

Result: T_n's successor node quantity ϕ_{T_n}

Function : Compute_Successors (G, T_n, G_n)

$\phi_{T_n} = 0$;

for $i = 1 : N$ **do**

Use Dijkstra's algorithm to calculate the shortest path to the gateway,

P_i =$\{i, \cdots$ forwarding nodes $\cdots, G_n\}$

if $T_n \in P_i$ **then**

$\phi_{T_n} = \phi_{T_n} + 1$;

end

end

return ϕ_{T_n};

Algorithm 2 Selecting the best node to perform OS-DoS attack

Input: G, T_n, G_n

Result: *OS-DoS node*

π_{max}= Compute_Successors(G, T_n, G_n);

OS-DoS node = *empty*;

for $i = 1$ *: all the neighbors* **do**

  Detach the neighbor i from G ▷ i= neighbor index

  Update network graph, $G' = \{V', E'\}$ ▷ $i \notin V', (\cdot, i) \notin E'$

  π= ComputeSuccessors(G', T_n, G_n);

  if $\pi > \pi_{max}$ **then**

    $\pi_{max} = \pi$;

    OS-DoS node $= i$;

  end

end

if *OS-DoS node* $\neq$ *empty* **then**

  Perform OS-RM attack

else

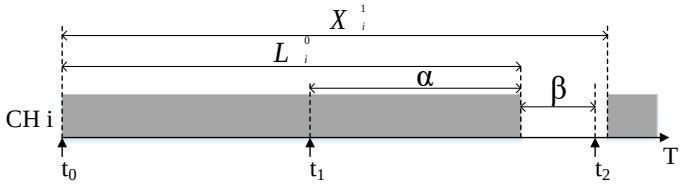
  Wait for the next update

end

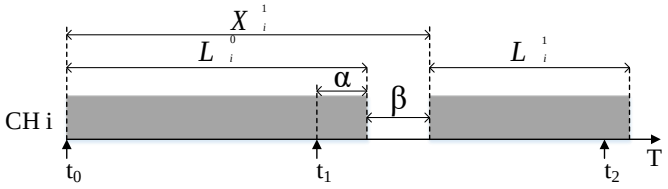
4.2.2 Channel State Prediction

Channel state predictor assists in updating the network graph G in each period, based on routing updates. In this section, we propose the prediction model to forecast future channel activity to update the network graph.

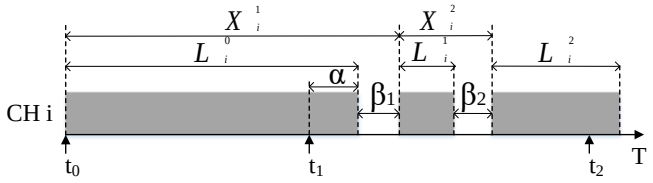
By utilizing the periodic routing update, an attacker can make predictions of the channel availability before the next route update arrives. Based on the prediction results, an attacker decides whether to change the link costs or not. We propose two criteria for determining whether the channel should be considered busy or idle: 1) the predicted probability that the channel is busy or idle and 2) the expected length



(a) No PU packet arrives between t_1 and t_2



(b) Only one PU packet arrives between t_1 and t_2



(c) Two PU packets arrive between t_1 and t_2

Figure 4.6: The PU activity on channel i; $N_i(t_1) = 1$.

of the activity or inactivity.

In Figure 4.6, t_0 represents the last moment PU becomes active, t_1 represents the last moment route update arrives, and t_2 represents the expected moment of the next route update. Figure 4.6 shows the PU traffic activity on channel i, where X_i^k represents the inter-arrival time of the kth packet. We denote $Y(t_2)$ as the number of PU packets that arrive between t_1 and t_2 and $N_i(t_2)$ as the status of the channel at time t_2, which is a binary variable between 0 and 1 representing the idle and busy state, respectively.

In the following, we calculate the probability that the channel state is active upon the next route update. All the figures are normalized to routing update length Δ. As shown in Figure 4.6(a), where $N_i(t_1) = 1$, the probability that the next channel state will be active and no PU packet arrives between t_1 and t_2 is

$$
\begin{aligned}
&Pr\{N_i(t_2) = 1, Y(t_2) = 0\} \\
&= Pr\{X_i^1 > t_2 - t_0\} Pr\{\alpha > \tau\} \\
&= Pr\{X_i^1 > t_2 - t_0\} Pr\{L_i^0 - (t_1 - t_0) > \tau\},
\end{aligned} \tag{4.1}
$$

where $L_i(k)$ denotes the length of the kth new PU packet in channel i and τ represents the activity threshold of PU. $X_i(k)$ and $L_i(k)$ depend on the channel parameters λ_i and μ_i.

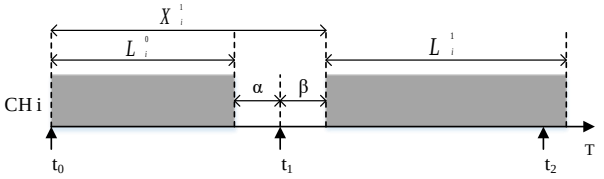
As shown in Figure 4.6(b), the probability that the channel state will be active and only one PU packet arrives between t_1 and t_2 is

$$
\begin{aligned}
&Pr\{N_i(t_2) = 1, Y(t_2) = 1\} \\
&= Pr\{\beta < 1 - \tau\} Pr\{\alpha + L_i^1 > \tau\} \\
&= Pr\{X_i^1 - L_i^0 < 1 - \tau\} Pr\{\alpha + L_i^1 > \tau\}.
\end{aligned} \tag{4.2}
$$

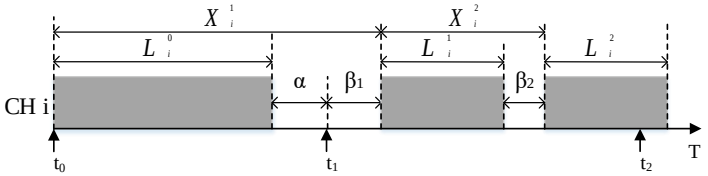
Similarly, in Figure 4.6(c), the probability that channel i is active and two packets come between t_1 and t_2 is,

$$
\begin{aligned}
&Pr\{N_i(t_2) = 1, Y(t_2) = 2\} \\
&= Pr\{\beta_1 + \beta_2 < 1 - \tau\} Pr\{\alpha + (L_i^1 + L_i^2) > \tau\} \\
&= Pr\{X_i^1 + X_i^2 - (L_i^0 + L_i^1) < 1 - \tau\} \\
&\quad Pr\{\alpha + L_i^1 + L_i^2 > \tau\}.
\end{aligned} \tag{4.3}
$$

Assume that U is the maximum number of PU packets that could come between t_1 and t_2. Hence, the probability of having the channel active and arriving h ($h \in [1, U]$)



(a) Only one PU packet arrives between t_1 and t_2



(b) Two PU packets arrive between t_1 and t_2

Figure 4.7: The PU activity on channel i; $N_i(t_1) = 0$.

PU packets is

$$\begin{aligned} & Pr\{N_i(t_2) = 1, Y(t_2) = h\} \\ & = Pr\{\sum_{k=1}^{h} \beta_k < 1 - \tau\} Pr\{\alpha + \sum_{k=1}^{h} L_i^k > \tau\} \\ & = Pr\{\sum_{k=1}^{h} X_i^k - \sum_{k=0}^{h-1} L_i^k < 1 - \tau\} Pr\{\alpha + \sum_{k=1}^{h} L_i^k > \tau\}. \end{aligned} \tag{4.4}$$

Therefore, the probability that channel i is active at time t_2 can be obtained by,

$$\begin{aligned} & Pr\{N_i(t_2) = 1 | N_i(t_1) = 1\} \\ & = Pr\{X_i^1 > t_2 - t_0\} Pr\{L_i^0 - (t_1 - t_0) > \tau\} \\ & + \sum_{h=1}^{U} \left[Pr\{\sum_{k=1}^{h} X_i^k - \sum_{k=0}^{h-1} L_i^k < 1 - \tau\} Pr\{\alpha + \sum_{k=1}^{h} L_i^k > \tau\} \right]. \end{aligned} \tag{4.5}$$

Likewise, in Figure 4.7(a), where $N_i(t_1) = 0$, the probability that next channel status will be active and one PU packet arrives between t_1 and t_2 is

$$\begin{aligned} & Pr\{N_i(t_2) = 1, Y(t_2) = 1\} \\ & = Pr\{\beta < 1 - \tau\} Pr\{L_i^1 > \tau\} \\ & = Pr\{X_i^1 - L_i^0 - \alpha < 1 - \tau\} Pr\{L_i^1 > \tau\}. \end{aligned} \tag{4.6}$$

Similarly, in Figure 4.7(b), the probability that channel i is active and two packets come between t_1 and t_2 is,

$$\begin{aligned} & Pr\{N_i(t_2) = 1, Y(t_2) = 2\} \\ & = Pr\{\beta_1 + \beta_2 < 1 - \tau\} Pr\{(L_i^1 + L_i^2) > \tau\} \\ & = Pr\{X_i^1 + X_i^2 - (L_i^0 + L_i^1) - \alpha < 1 - \tau\} \\ & \quad Pr\{L_i^1 + L_i^2 > \tau\}. \end{aligned} \tag{4.7}$$

Therefore the probability that channel i is active at time t_2 can be obtained by,

$$\begin{aligned} & Pr\{N_i(t_2) = 1 | N_i(t_1) = 0\} \\ & = Pr\{X_i^1 - L_i^0 - \alpha < 1 - \tau\} Pr\{L_i^1 > \tau\} \\ & + \sum_{h=1}^{U} \left[Pr\{\sum_{k=1}^{h} X_i^k - \sum_{k=0}^{h-1} L_i^k - \alpha < 1 - \tau\} Pr\{\sum_{k=1}^{h} L_i^k > \tau\} \right]. \end{aligned} \tag{4.8}$$

Thus, if the channel statistics (e.g., λ and μ) are known, the predicted probabilities can be calculated. Therefore, based on the prediction, the policy that we consider the channel as active, when

$$Pr\{N_i(t_2) = 1\} > \Gamma, \tag{4.9}$$

where Γ is the threshold above which the channel is considered active by the predictor model. After making channel decisions, the attacker will calculate the corresponding link costs.

However, learning the channel statistics requires significant efforts and hence, we design and propose a HMM based technique to estimate the channel parameters λ

and μ.

4.2.3 HMM based Parameter Estimator

A slotted discrete-time model is used for the channel activity. The decision on whether a channel is busy or not is made based on the channel activity during the last period. If the channel activity exceeds the given threshold τ, then it is assumed to be in the ON state or otherwise OFF.

We first present the structure of the HMM and then we give a brief introduction of the forward-backward procedure in Baum-Welch (BW) algorithm[113]. Finally, by analyzing the estimated parameters, we calculate the channel parameters.

Hidden Markov Model: A Hidden Markov process is a Markov process consisting of two states, where X is the hidden process that is never observable and Z is the observation process that can be seen by the observers (i.e., the OS-RM attacker). X_t and Z_t denote the hidden state and observation state at time t, respectively. The hidden process follows a Markov process with a finite number of states and the observable process is another probabilistic function which generates *symbols* based on the hidden states. The set of symbols comes from a defined *alphabet* A. In our case, $A = \{0, 1\}$ (i.e., 0 = OFF and 1 = ON).

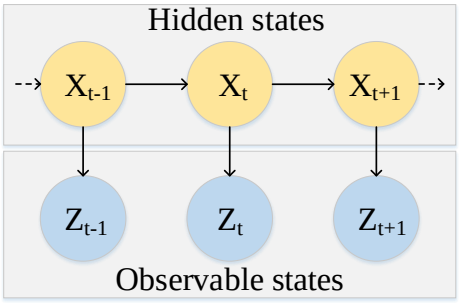


Figure 4.8: The hidden Markov model.

The general concept of an HMM is illustrated in Figure 4.8. A system of discrete time is changing randomly from one state to another, within a finite state space S. In our case, the finite space $S = \{0, 1\}$. The evolution of the hidden sequence

$X_1, X_2, ..., X_T$ is hidden, which represents PU states. However, it can be expressed by a sequence of observed symbols from the alphabet A (i.e., $Z_t \in A$), which represents routing updates. In order to model the HMM, it is necessary to define the parameters first:

- Number of hidden states, $s = 2$
- Number of symbols, $a = 2$
- Initial state distribution, $\pi = \{\pi_i\}$, where $i = 0, \cdots, s-1$
- One-step state transition probabilities, $P = p_{ij}$, where $i, j = 0, \cdots, s-1$
- Symbol emission probability, $B = b_j(k)$, where $j = 0, \cdots, s-1$ and $k = 0, \cdots, a-1$

Therefore, the one-step state transition probability is

$$\begin{aligned} &Pr(X_t = j | X_{t-1} = i, X_{t-2} = i_{t-2}, \cdots, X_2 = i_2, X_1 = i_1) \\ &\quad = Pr(X_t = j | X_{t-1} = i) \\ &\quad = p_{ij}, \end{aligned} \tag{4.10}$$

where, $i_1, i_2, .., i_{t-2}, i, j \in \{0, 1\}$ and $t \geq 2$. And the emission probability is

$$b_j(k) = Pr(Z_t = k | X_t = j). \tag{4.11}$$

The BW algorithm is an iterative approach to estimate the HMM parameters $\eta = [\pi, P, B]$ such that the $Pr(Z|\eta)$ is maximized. To estimate the parameters, we define the following parameters:

- Forward probability, $\alpha_t(i) = Pr(Z_1, Z_2, \cdots, Z_t, X_t = S_i | \eta)$, for $S_i \in \{0, 1\}$
- Backward probability, $\beta_t(i) = Pr(Z_{t+1}, Z_{t+2}, \cdots, Z_{T-1}, Z_T, X_t = S_i | \eta)$, for $S_i \in \{0, 1\}$

- Estimate of state transitions, $\gamma_t(i,j) = Pr(X_t = S_i, X_{t+1} = S_j|Z,\eta)$, for $S_i, S_j \in \{0,1\}$. It represents the probability of being in state S_i at instant t and in state S_j at instant $t+1$, given the observation sequence Z and the model parameters $\eta = [\pi, P, B]$

- Estimate of the state at each observation, $\delta_t(i) = Pr(X_t = S_i|Z,\eta)$, for $S_i \in \{0,1\}$. It represents the probability of being in state S_i at instant t, given the observation sequence Z and the model parameters $\eta = [\pi, P, B]$

The estimation variables for the HMM parameters are expressed in terms of $\gamma_t(i,j)$ and $\delta_t(i)$:

$$p_{ij} = \frac{\sum_{t=1}^{t=T-1} \gamma_t(i,j)}{\sum_{t=1}^{t=T-1} \delta_t(i)}. \tag{4.12}$$

$$b_j(k) = \frac{\sum_{t=1,Z_t=k}^{t=T} \delta_t(j)}{\sum_{t=1}^{t=T} \delta_t(j)}. \tag{4.13}$$

$$\pi_i = \delta_1(i). \tag{4.14}$$

In (12) the numerator represents the expected number of transitions from state S_i to state S_j over the interval $T-1$, while the denominator represents the expected number of times a transition happens from state S_i. The numerator in (13) represents the expected number of transitions from state S_j at which symbol k is observed. In (12)-(14), $\gamma_t(i,j)$ and $\delta_t(i)$ are calculated as follows:

$$\gamma_t(i,j) = \frac{\alpha_t(i) p_{ij} b_j(Z_{t+1}) \beta_{t+1}(j)}{Pr(Z|\eta)}. \tag{4.15}$$

$$\delta_t(i) = \sum_{all\ S_j \in \{0,1\}} \gamma_t(i,j). \tag{4.16}$$

The forward and backward probabilities in the above equations are calculated recursively as follows:

Initialization:

$$\alpha_1(i) = \pi_i b_i(1), \quad 0 \le i \le s-1. \tag{4.17}$$

$$\beta_t(i) = 1, \;\; 0 \leq i \leq s-1. \tag{4.18}$$

Recursion:

$$\alpha_{t+1}(j) = \left[\sum_{i=0}^{s-1} \alpha_t(i) p_{ij}\right] \; b_j(Z_{t+1}). \tag{4.19}$$

$$\beta_t(i) = \sum_{j=0}^{s-1} p_{ij} b_j(Z_{t+1}) \beta_{t+1}(j). \tag{4.20}$$

The recursion process terminates when $Pr(Z|\eta)$ maximizes, which is the probability of observing the sequence Z given the parameter $\eta = [\pi, P, B]$.

$$Pr(Z|\eta) = \sum_{i=0}^{s-1} \prod_{t=1}^{T} \alpha_t(i). \tag{4.21}$$

Analysis of PU Activity: In this section, we need to extract the PU activity from the estimated HMM parameters $\eta = [\pi, P, B]$. To do this, we first introduce a new set of PU parameters, $\theta = [\lambda, \mu]$, where λ means the traffic arrival rate and μ means the traffic departure rate. From our network model, the length of the ON and OFF state are exponentially distributed. In [114], a useful method to compute the state transition rate matrix from the state transition probability matrix is provided. We denote the transition rate matrix as Q and

$$Q = \begin{pmatrix} -\lambda & \lambda \\ \mu & -\mu \end{pmatrix}. \tag{4.22}$$

As described in η, P is the one-step state transition probability matrix. We know that $P = \exp(Q\Delta)$ and $Q = \log(P)/\Delta$, where Δ is the route update period. However, the computational procedure is cumbersome and $\log(\cdot)$ has a limitation when P has a non-positive eigenvalue. Therefore, we adopt the mapping approach introduced in [114], which provides an easier computational approach and provides enough degree of accuracy. If the two-dimensional transition rate matrix is the form shown in (22),

then the transition probability matrix is:

$$P = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} = \begin{pmatrix} \exp^{-\lambda\Delta} & 1 - \exp^{-\lambda\Delta} \\ 1 - \exp^{-\mu\Delta} & \exp^{-\mu\Delta} \end{pmatrix}. \tag{4.23}$$

In (23), the relation between P and Q unfolds the relationship between η and θ.

4.3 Performance Evaluation

We evaluate the impact of the OS-RM attack by conducting simulations in Matlab. We consider a grid size distribution of 25 CR enabled nodes, with 24 being CR-WMRs and a gateway (Fig. 4.9). The attacker and the target node are colored with red and green color, respectively. The gateway has three neighboring CR-WMRs via which other routers can communicate with the gateway. In reality, traffic is not uniformly distributed among these three CR-WMRs due to their different spectrum availability. We consider a uniform distribution of PUs in the network. Parameters of our simulations are listed in Table I.

Table 4.1: Simulation Parameters: OS-RM Attack

Simulation area	1000x1000
Simulation time	50 seconds
Training time	25 seconds
SU sensing range	200
The number of PUs	10
The number of SUs	25
Bandwidth	2 Mbps
The size of (RTS+CTS)	160 + 112 bits (802.11b/g)
Sensing duration	1 ms (802.22)
SU traffic	$\rho = \lambda_s/\mu_s = 0.05 \sim 0.25$
SU packet size	750 bytes
Number of channels	10

4.3.1 HMM Estimation

The performance of the OS-RM attack relies significantly on how accurately HMM-based estimators can estimate the parameters of PUs in the network. Furthermore,

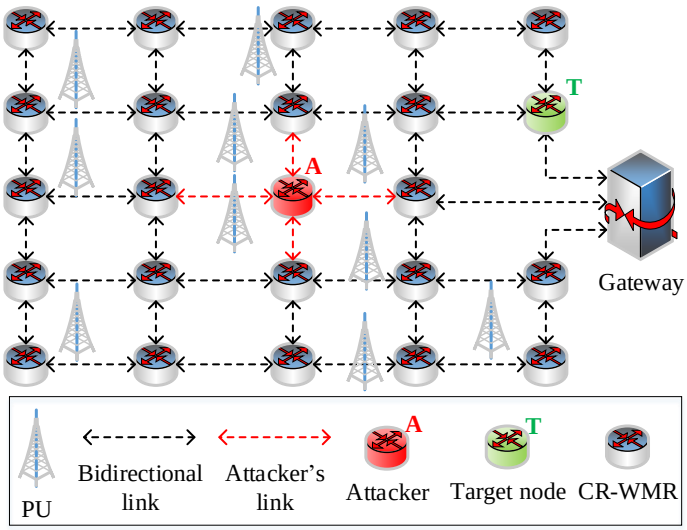


Figure 4.9: Simulation scenario.

the length of a training sample is instrumental to the learning performance. In Fig. 4.10, we can observe the trend of estimation error over the time for packet arrival rate (λ) and service rate (μ). Estimation errors reduce to below 4% when the estimator is trained to 50 seconds.

In our simulations, we train the HMM estimator with 25 seconds of data and observe the impact of the attack for the next 25 seconds without changing the PU activity rate. Nevertheless, in reality, the PU activity rate is not going to be constant all the time and the HMM estimator should reestimate to track changes. The optimal training time length based on the traffic change rate is out of this research's scope. In the future, we plan to propose a strategy for the attacker in a time-varying PU network.

4.3.2 Impact on Traffic Flow

In Fig. 4.11 (color and number coded), we observe changes in traffic flow due to the rebalancing effect caused by the OS-DoS attack on the victim node. Without attack, two neighboring CR-WMRs carry most of the traffic (Fig. 4.11(a)) except the target node. However, with the OS-RM attack, we can see that a portion of previous routes are disrupted (Fig. 4.11(b)). As a result, traffic flows change directions and a

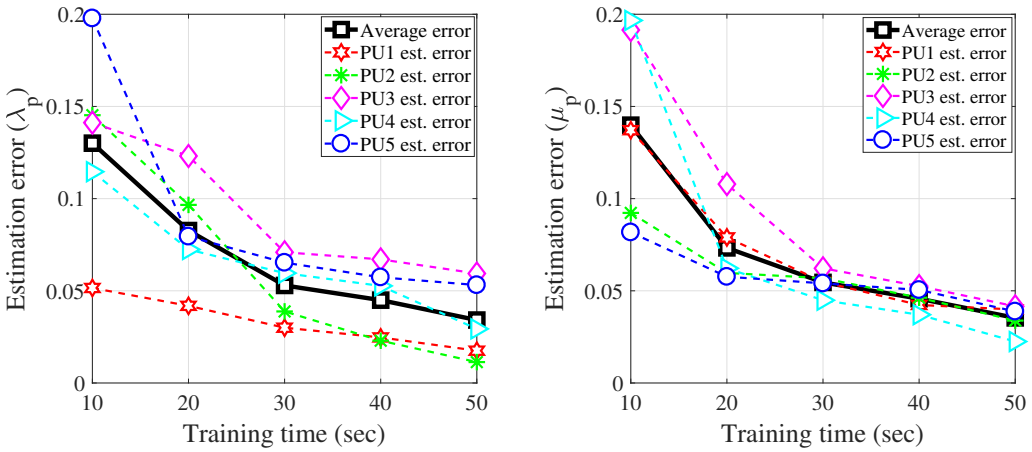
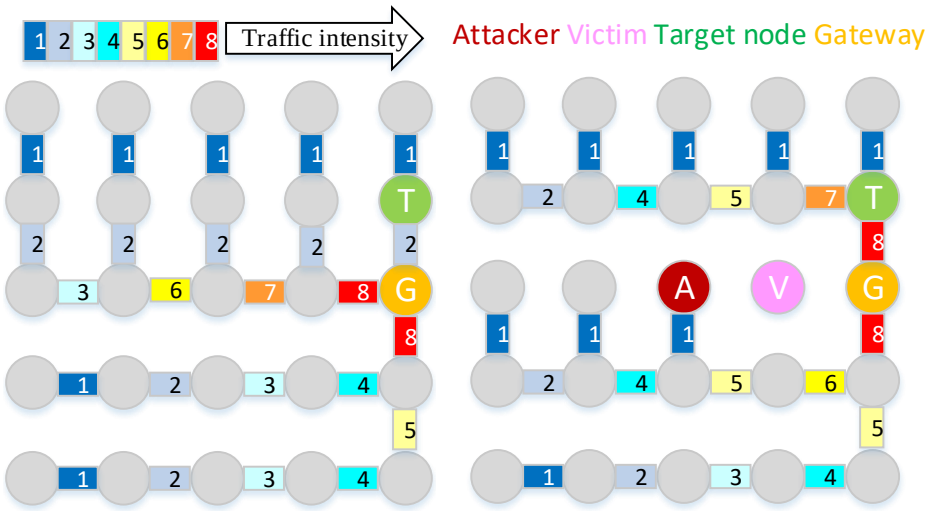


Figure 4.10: HMM estimation performance.

few nodes who were carrying less traffic are exposed to higher traffic load now. Most significant change in traffic is observed in the target node. This strategy works as the driving force to maneuver traffic to any node an attacker wants. Though we discussed only about diverting traffic towards a particular node, the same kind of strategy can be employed to divert traffic from one.

4.3.3 Impact on Network Performance

We compare the impact of lower-layer attacks, e.g., conventional jamming, random jamming, OS-RM attack without learning, and OS-RM attack with learning, used as an auxiliary attack in an effort to manipulate routes. In Fig. 4.12(a)-(d), we compare the impact of these front-end attacks with an increasing SU activity. From Fig. 4.12(a), we can observe the increased number of traffic flows going through the target node. Though the jamming attack can also influence traffic flows, it is less significant as compared to the OS-RM attack. In the jamming attack, all the nodes within the radio range of the jammer get affected, hence, the traffic flows disperse in the whole network. Moreover, it is inefficient to use the jamming strategy due to the high energy required by the jammer. Furthermore, as the attacker is an authorized network entity and has the similar power requirement as other entities, it is unrealistic



(a) (b)

Figure 4.11: Traffic heat map: (a) no attack and (b) OS-RM attack.

to perform jamming. However, unlike the jamming attack, an OS-DoS attack can be performed on an individual node of choice. Thus, we can observe more than 50% increase in traffic flows to the target node.

In Fig. 4.12(b)-(d), we can observe the change in key performance metrics of the flows going through the target node (i.e., throughput, delay, and packet drop). If the perpetrator's objective is to increase congestion at the target node, then from Fig. 4.12(b)-(c), it is quite evidential that this attack reduces throughput and increases delay experienced by the flows going through the target node. The effect of delay stems from the queuing delay in intermediate nodes. In addition, a virtual blackhole creates in the network as more packets are being dropped. The increase in packet drop stems from the packet drop in intermediate nodes due to the timeout and blocking of new sessions. From Fig. 4.12, we can observe the performance improvement by implementing learning strategy of the attacker when $\Gamma \geq 0.6$.

4.3.4 Influence on Traffic vs. Distance

We also observe that the attacker is more influential when it is situated higher up in the routing tree (gateway is the root of the tree). In another word, the attacker

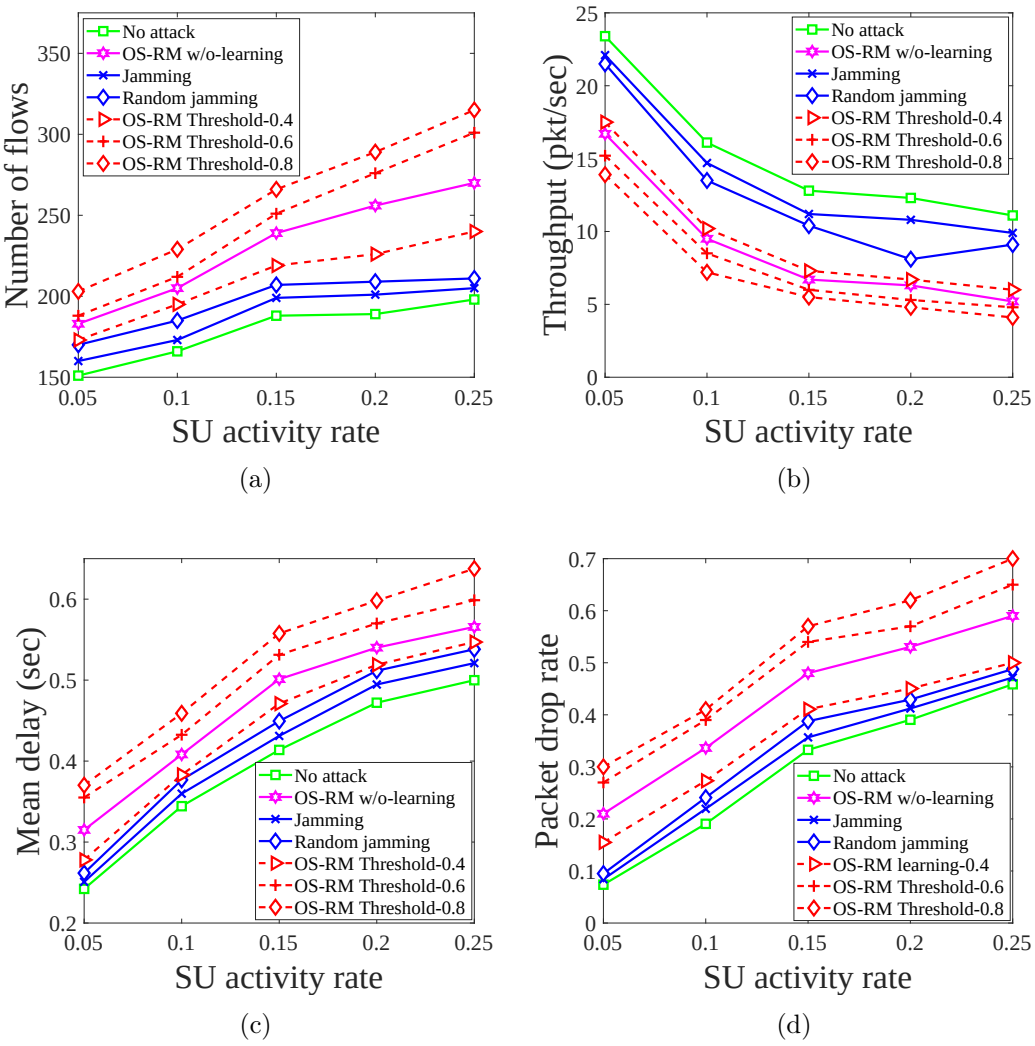


Figure 4.12: Impact of lower-layer attacks on route manipulation: (a) number of traffic flows, (b) throughput, (c) mean dealy, and (d) packet drop rate.

is more influential when more number of traffic flows go around it. In Fig. 4.13, we can observe that the number of traffic flows actually increases when the distance between the attacker and the target node changes from 1-hop to 2-hop, which is counterintuitive to what we just mentioned. However, when the attacker is a direct neighbor to the target node, it cannot perform the OS-DoS attack on the target node. Therefore, the attacker has one less neighbor to maneuver the neighbor's traffic flows and hence the decrease in the number of flows. Therefore, we can deduce that the attacker is more potent when it is 2-hop away from the target node.

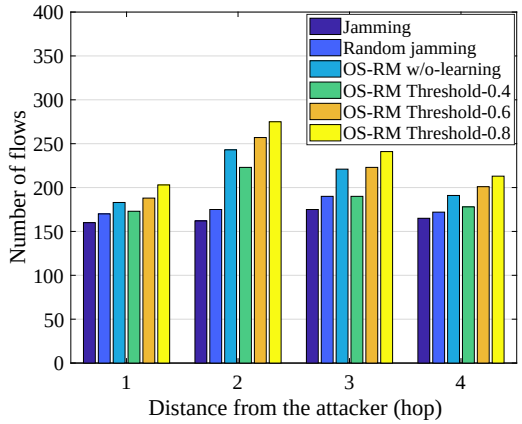


Figure 4.13: Impact on traffic flows vs. distance between the attacker and target node.

Depending on the end objective of the attacker, the impact of the OS-RM attack can affect other network layers also. In our proposed attack, the target node could be actually a pre-compromised node to perform wormhole attacks, black-hole attacks or perhaps a benign node to create network congestion. From the above observations, one could imagine the atrocities an attacker can perpetuate if it achieves a significant amount of control over the traffic flow.

CHAPTER 5: PROPOSED DEFENSE AGAINST OFF-SENSING ATTACKS IN CR NETWORKS

In this chapter, a safeguard approach based on the Markov decision process is proposed to counteract the off-sensing attack.

5.1 System Model

We consider two SUs that are trying to communicate between themselves in the presence of OS-attackers. These two SUs could be network entities of either an infrastructure-based network (i.e., one SU is a CR access point that opportunistically accesses the licensed spectrum, and the other is a CR user communicating with other network users through the access point) or an ad-hoc network. They are located within the interference region of OS-attackers, and OS-attackers are authorized and authenticated entities in the network.

5.1.1 Network Model

PU and SU Model: We consider the presence of M homogeneous channels (and M PUs), each with a fixed bandwidth. Time is divided into equal slots. Transmissions are packet based for both PUs and SUs, and a packet starts at the beginning of a mini-slot and finishes at the end of a mini-slot. A mini-slot is the time to perform a fast-sensing [115] and to exchange a request to send/clear to send (RTS/CTS) pair, and a slot is a multiple of mini-slots. Each PU randomly selects a channel to access and alternates between the ON and OFF states, according to an ON-OFF model. Let α and β denote the transition probabilities from the ON to OFF state and from the OFF to ON state, respectively. We consider a saturated SU traffic scenario, which means SUs always have a packet in their buffer to transmit. Hence, an SU

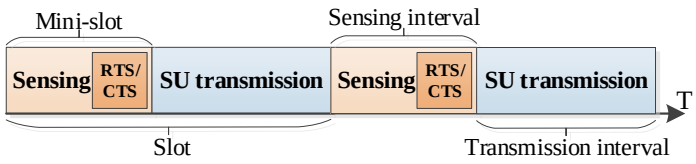


Figure 5.1: The SU schedule.

continuously transmits on a channel until it finds the current channel busy during a sensing interval or experiences a transmission failure (e.g., if an ACK is not received from the other SU). Transmission failures can result from two reasons: collision with a reappeared PU and interference from an OS-attacker. However, SUs are unable to determine the exact reason of transmission failures due to their inability to sense the channel during transmission or reception.

SU Access Protocol: Each transmission attempt of an SU must be preceded by a sensing interval. As shown in Figure 5.1, SUs periodically run between sensing and transmission intervals. An SU is allowed to access a channel when it finds the sensing result suitable to transmit (e.g., senses that no PU is present). After sensing the channel available, two SUs exchange RTS/CTS messages to reserve the channel. Each SU is equipped with one half-duplex radio for spectrum sensing, control information exchange, and data transmission. With one radio, an SU can sense the channel only before initiating the transmission (i.e., in the sensing interval).

5.1.2 Network Coordination Scheme

In this chapter, we assume that a common control channel (CCC) is unavailable and two SUs must find a common available channel between them to initiate a data transmission. Rendezvous technique works as the process to find a common available channel, where two SUs follow a channel-hopping process to meet and exchange control information on a common available channel. A significant amount of research has been conducted on rendezvous techniques. However, the choice of a specific scheme does not impact the performance of our proposed attack and defense mechanism, as

long as attackers have no prior knowledge of the victim's hopping sequence. Thereby, we assume that benign SUs have successfully performed rendezvous with each other, using any existing rendezvous scheme, and they share a time-seeded pseudo-random channel-hopping sequence for future communications.

5.1.3 OS-DoS Attack

An OS-attacker detects the transmission of a particular victim SU from the RTS/CTS message that precedes each transmission attempt. Figure 5.2 provides an illustration of the OS-DoS attack under a periodic channel-hopping process.

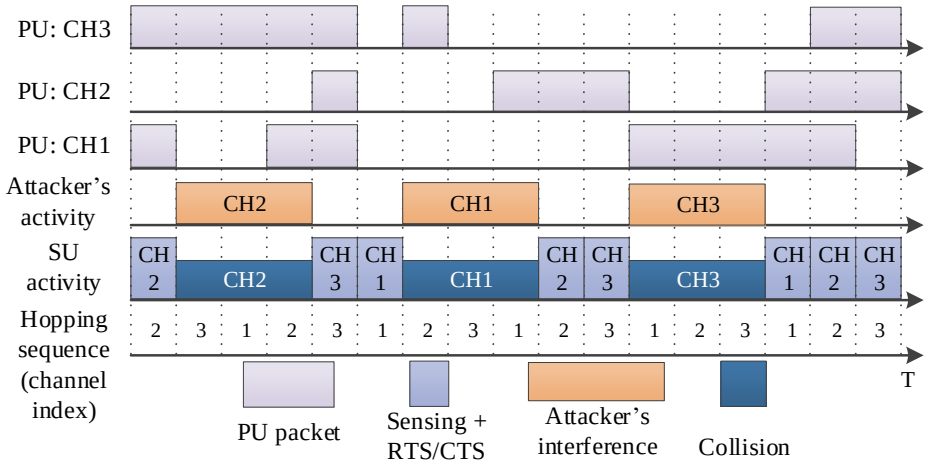


Figure 5.2: OS-DoS attack under periodic channel-hopping process.

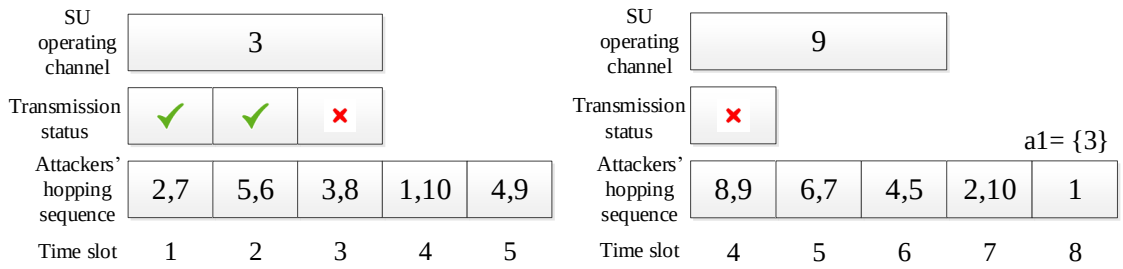
In Figure 5.2, the OS-attacker knows the channel-hopping sequence of the victim SU and interferes with each transmission originating from and to the victim (by overhearing RTS/CTS messages). Here, the attacker interferes the whole packet time to make sure that the victim cannot decode the packet and tries to create a DoS situation for the victim SU by causing consecutive successful collisions. However, in reality, it is likely that the attacker does not have any knowledge of the victim's hopping sequence, and it requires shrewder efforts from the attacker to perpetrate successive transmission failures. Next, we propose a novel strategy for an attacker to perpetrate the OS-DoS attack, *without any knowledge of the victim's hopping sequence and operating channel.*

5.2 Proposed Random-OS Attack Model

In our proposed OS-DoS attack, the short-term goal is to cause successive transmission failures, and the long-term goal is to reach the maximum limit of transmission attempts to force the victim to drop the current packet. In Figure 5.2, if the maximum transmission attempt is 3, then the SU packet would be dropped. However, the assumption that attackers know the channel-hopping sequence of the victim is unrealistic and so is the strategy of an attacker to interfere with each transmission of the victim (due to the deterministic hopping sequence of the attacker); the victim can infer the attacker's activity and detect the attacker with a longer fine-sensing (explained in Section 5.3). Therefore, we propose a random strategy for OS-DoS attackers, where attackers have no prior knowledge of the victim's channel-hopping sequence, and they hop to different channels in each slot to detect the victim and to perpetrate the OS-DoS attack.

Basic Principles: We assume the presence of m OS-attackers ($m<M$) with the same hardware configuration as benign SUs. We consider that these OS-attackers coordinate among themselves using an out-of-band secure channel (i.e., a secure control channel for attackers only), and they attack non-overlapping channels to increase their chance to detect the operating channel of the victim sooner. Attackers detect a transmission of a particular victim by listening to the RTS/CTS message. After the detection, they perform OS-attack in the transmission interval.

Short-term Strategy: With the help of coordination, attackers visit m different channels during each slot. Here, attackers randomly generate a channel-hopping sequence after each successful attack (i.e., transmission failure) and hop through the sequence periodically until they find the operating channel of the victim SU. This strategy of channel-hopping helps attackers to put an upper bound on how long (i.e., the channel residence time) a victim SU can continuously use a channel. The upper bound will be discussed later in this section. Figure 5.3(a) shows an illustration of



(a) Before OS-attack. (b) After OS-attack.

Figure 5.3: First phase of the random-OS attack.

the attack sequence with $M = 10$ and $m = 2$. It shows the hopping sequence of attackers before a successful OS-attack. Here, the operating channel of the victim SU is channel-3 and, in slot-3, attackers detect and perpetrate the attack. a_i represents the channels where attackers have conducted OS-attack and i represents the number of successive attacks (or transmission failures).

In the OS-attack, a victim cannot determine the exact reason of the transmission failure. Thereby, the victim will randomly hop to a new channel (believing that it has interfered with a reappeared PU), will try to stay on that channel as long as plausible, and will not hop back to the previously attacked channels (i.e., a_i) until it achieves a successful packet transmission. Hence, it is inefficient for attackers to revisit the previously attacked channels for a particular packet. After each successful perpetration of the attack (or transmission failure), attackers randomize their hopping sequence, excluding a_i. Therefore, after i successive transmission failures, attackers have $M - i$ channels to randomize. Figure 5.3(b) illustrates a new hopping sequence of attackers.

Long-term Strategy: As OS-DoS attack considers that the victim must experience G consecutive transmission failures ($G<M$) before discarding the current packet, attackers stay persistent to increase their chance of successful attack after each successive OS-attacks. Hence, they keep excluding channels that they have already attacked earlier, for the current packet. Figure 5.4 shows an illustration of a scenario, where

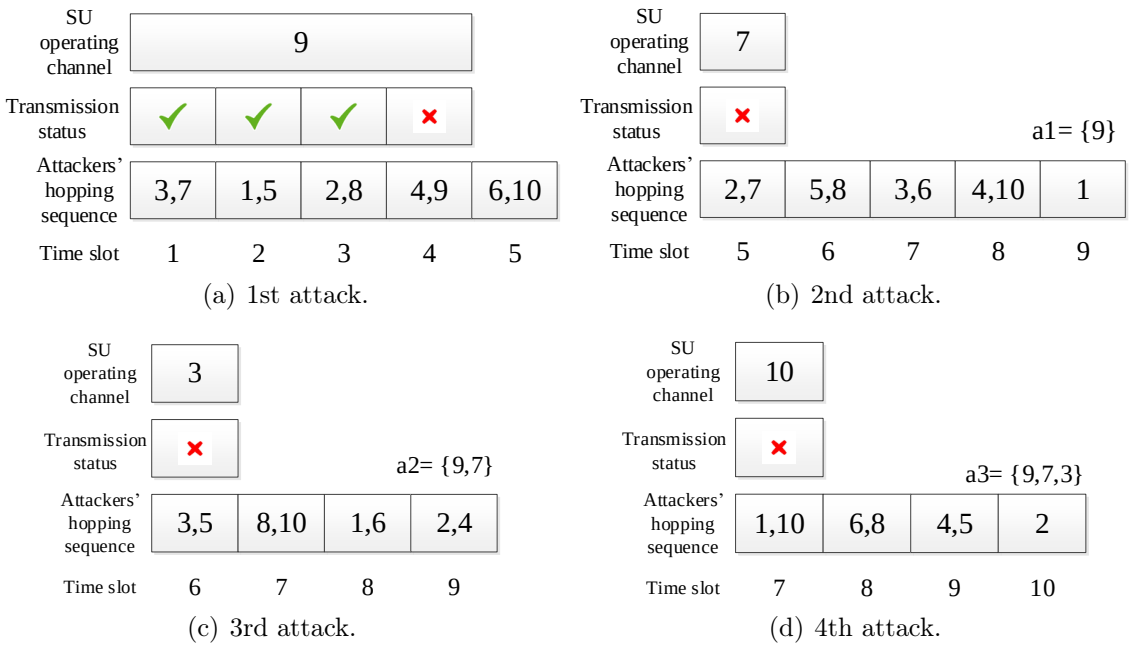


Figure 5.4: A scenario of successful OS-DoS attack with $G = 4$.

$G = 4$, and attackers are successful to drop the packet with successive attacks.

After i_{th} successful attack, if attackers are not successful in the subsequent slot, they consider that the victim had a successful transmission. Hence, they will randomize their hopping sequence (i.e., nullify a_i), excluding the channels they have visited in the current slot (since currently visited channels are free, there is no need to visit again in this period), and begin a new period (one period $= \lceil M/m \rceil$ slots).

If attackers cannot detect the operating channel and one period has finished, they will revisit the channels following the same sequence. Given M channels and m OS-attackers, if the victim SU stays on the same channel, the operating channel of the victim will be detected within $\lceil M/m \rceil$ slots. Thereby, the maximum number of consecutive successful transmissions an SU can have in a channel is $K = \lceil M/m \rceil - 1$. This is the upper-bound that has been discussed earlier in this section.

Summary: The proposed OS-DoS attack strategy introduces uncertainties in actions of attackers; hence, we name it random-OS attack. Unlike the deterministic approach in Figure 5.2, the proposed strategy introduces a random hopping sequence

Figure 5.5: The extra-sensing interval.

for attackers. Due to this randomness, it is not guaranteed that the victim can detect an attacker's interference by a single fine-sensing [116], rather it may require multiple attempts to detect an attacker. Therefore, the victim SU must use the fine-sensing interval (explained in the next section) wisely to maximize the chance of detection.

5.3 Proposed Safeguard Approach: Hide and Seek

In this section, we propose a solution to the random-OS attack problem by modeling it as an MDP-based game with three actions: *stay*, *hop*, and *extra-sense*. Besides *stay* and *hop*, we propose an action *extra-sense* to increase the diversity of defense (Figure 5.5). In *extra-sense*, instead of transmitting in the transmission interval, an SU tries to detect OS-attackers by fine-sensing the channel which we call the extra-sensing interval. With fine-sensing, an SU can differentiate between the transmission of a PU and an attacker. Now, with these available actions, the MDP deduces an optimal policy, which provides the optimal action to take at each state that maximizes the reward of playing this MDP-based game. In this section, we model the attack and defense problem as an MDP, and we develop a single agent (i.e., a victim SU) MDP-based defense method to counteract the random-OS attack.

5.3.1 Formation of the MDP

We assume that the channel-hopping sequence of the victim SU is unknown to the attacker; however, the attacker can iteratively sweep through the available channels and detect the presence of the victim SU. As we consider the presence of multiple (i.e., m) OS-attackers and coordination among themselves, they will not hop to the same channel together. Instead, they will hop to m different channels to determine the operating channel of the victim SU faster. The SU will decide its action at the

end of each time slot, based on the observation of the current state. The SU receives an immediate reward $U(n)$ in the n_{th} time slot,

$$\begin{aligned} U(n) = & R.\mathbf{1}(Successful\ transmission) \\ & - L.\mathbf{1}(Transmission\ failure) - C.\mathbf{1}(Hopping\ cost) \\ & - B.\mathbf{1}(Busy\ channel) \\ & - F.\mathbf{1}(Penalty\ for\ policy\ violation) \\ & - Q.\mathbf{1}(Packet\ drop) + E.\mathbf{1}(Attacker\ detection), \end{aligned} \tag{5.1}$$

where $\mathbf{1}(\cdot)$ is an indicator function of the event in brackets.

As the employed strategy impacts the current state and also the future states, the expected reward of this game is,

$$\overline{U} = \sum_{n=1}^{\infty} \delta^{n-1} U(n), \tag{5.2}$$

where δ represents the discount factor ($0 < \delta \leq 1$). It measures the significance of the future reward values.

5.3.2 Markov Model

This subsection demonstrates the proposed MDP model and defines state space, action space, rewards, and transition probabilities. We assume that attackers sweep through all channels periodically; hence, the probability of an operating channel being detected depends on the channels that have been visited earlier in the sequence. This consideration helps us to conform the requirement of a Markov process (i.e., a future state of the Markov process depends only on the current state).

Markov States: The state denotes the status of an SU at the end of a time slot. Here, the proposed Markov model (Figure 5.6) has six kinds of states:

P : The SU senses that the channel is occupied by a PU.

T_i : The SU hopped onto a new channel and had i consecutive successful transmissions ($1 \leq i \leq K$).

D_j : The SU had j consecutive transmission failures in j different channels ($1 \leq j \leq G$).

ES_0 : The SU employed the action *extra-sense* and found the channel is free (i.e., no PU or OS attacker).

ES_1 : The SU employed the action *extra-sense* and found the channel is reoccupied by a PU.

ES_a : The SU employed the action *extra-sense* and detected an OS-attacker successfully.

We represent the whole state space as $\mathbf{X} \triangleq \{P, T_1, T_2, \cdots, D_1, D_2, \cdots, ES_0, ES_1, ES_a\}$.

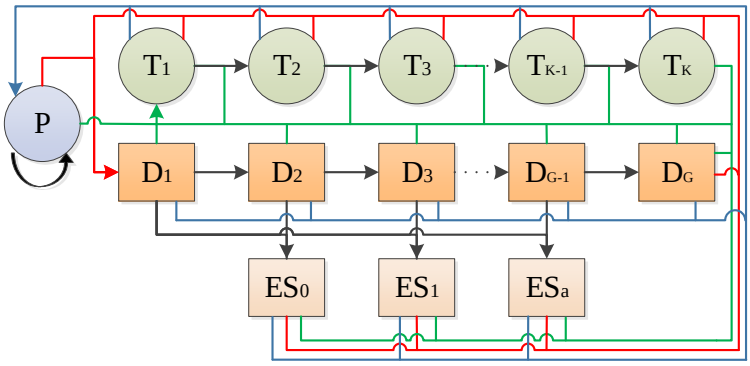


Figure 5.6: The proposed MDP.

Actions: Here, we have three actions available at each state:

stay (s): The SU remains on the current channel in the next time slot and initiates a transmission.

hop (h): The SU hops to a new channel in the next time slot and initiates a transmission.

extra-sense (es): The SU hops to a new channel in the next time slot and fine-senses the channel for interference.

We represent the whole action space as $\mathbf{A} \triangleq \{s, h, es\}$.

Rewards: Let $U(S, a, S')$ represent the reward when an SU takes action $a \in \mathbf{A}$ in state $S \in \mathbf{X}$ and enters into state $S' \in \mathbf{X}$. Now using (5.1), we define rewards:

$$U(S,a,S') =$$

$$\begin{cases} R, & \text{if } \{S,a,S'\} = \{T_i, s, T_{i+1}\}, i = 1, \cdots, K-1 \\ R-C, & \text{if } \{S,a,S'\} = \{\mathbf{X}, h, T_1\} \\ -L, & \text{if } \{S,a,S'\} = \{T_i, s, D_1\}, i = 1, \cdots, K-1 \\ -L-C, & \text{if } \{S,a,S'\} = \{\mathbf{X}, h, D_j\}, j = 1, \cdots, G-1 \\ -Q-C, & \text{if } \{S,a,S'\} = \{D_{G-1}, h, D_G\} \\ -B, & \text{if } \{S,a,S'\} = \{T_i, s, P\}, i = 1, \cdots, K-1 \\ -B-C, & \text{if } \{S,a,S'\} = \{\mathbf{X}, h, P\} \\ -F, & \text{if } \{S,a,S'\} = \{Z, s, \mathbf{X}\}, Z \in \{D, P\} \\ -Q, & \text{if } \{S,a,S'\} = \{\mathbf{X}, es, Z\}, Z \in \{ES_0, ES_1\} \\ E-Q, & \text{if } \{S,a,S'\} = \{\mathbf{X}, es, ES_a\}. \end{cases} \tag{5.3}$$

Transition Probabilities: As m attackers are going through their attack channel sequence, at state T_i, only $\max(M - im, 0)$ channels have yet to be visited by attackers, and another m channels will be visited in the subsequent slot. Therefore, the probability of an OS-attack (with action *stay*) in absence of a PU on the channel,

$$Pr_{at|s} = \begin{cases} \dfrac{m}{M - im}, & \text{if } i < K \\ 1, & \text{otherwise.} \end{cases} \tag{5.4}$$

The transition probabilities from state T_i with action *stay* is,

$$\begin{aligned} Pr(T_{i+1}|T_i, s) &= (1-\beta)^{l+1}(1 - Pr_{at|s}), \\ Pr(D_1|T_i, s) &= (1-\beta)\{1 - (1-\beta)^l\} \\ &\quad + (1-\beta)^{l+1} Pr_{at|s}, \\ Pr(P|T_i, s) &= \beta, \end{aligned} \tag{5.5}$$

where an SU packet is l mini-slots long, and each SU packet is preceded by 1 mini-slot long sensing interval. Note that the action *stay* is a violation of hard-coded network policy in state P and D_j and subject to penalty (i.e., $-F$).

When there are plenty of channels in the network, the time interval of visiting back to a channel is long; hence, we can approximate the probability of finding the channel busy with action *hop* as the steady-state probability,

$$Pr(P|S,h) = \frac{\beta}{\alpha+\beta} = \rho, \quad S \in \mathbf{X}. \tag{5.6}$$

Now, the SU takes action *hop* and selects a new channel randomly from $M-1$ channels (the SU does not hop to the same channel it found busy in the current slot) from the current state P and hands off to that channel. Provided that the new channel is available, the probability of an OS-attack is,

$$Pr_{at|h,P} = \frac{1}{M} \cdot \frac{m-1}{M-1} + \frac{M-1}{M} \cdot \frac{m}{M-1}. \tag{5.7}$$

The transition probabilities from state P with action *hop* is,

$$\begin{aligned} Pr(T_1|P,h) =& (1-\rho)(1-\beta)^l(1-Pr_{at|h,P}), \\ Pr(D_1|P,h) =& (1-\rho)\{1-(1-\beta)^l\} \\ &+ (1-\rho)(1-\beta)^l Pr_{at|h,P}. \end{aligned} \tag{5.8}$$

When an SU takes action *hop* from state T_i, it randomly selects a channel from $M-1$ channels (excluding the current one). The probability that attackers will attack the new channel in the next slot depends on two cases:

- The new channel is already visited by attackers: The new channel is one of the im channels visited by attackers.

- The new channel is not visited by attackers: The new channel is among one of the $M - im - 1$ channels that are not visited by attackers, and it will not be visited by attackers in the next slot.

Therefore, the probability of OS-attack,

$$Pr_{at|h,T} = 1 - \left(\frac{mi}{M-1} + \frac{M - im - 1}{M-1}(1 - Pr_{at|s}) \right). \tag{5.9}$$

The transition probabilities from state T_i with action *hop* is,

$$\begin{aligned} Pr(T_1|T_i, h) =& (1-\rho)(1-\beta)^l(1 - Pr_{at|h,T}), \\ Pr(D_1|T_i, h) =& (1-\rho)\{1-(1-\beta)^l\} \\ &+ (1-\rho)(1-\beta)^l Pr_{at|h,T}. \end{aligned} \tag{5.10}$$

When an SU takes action *hop* from state D_j, it randomly selects a channel from $M-j$ channels. As the SU has experienced transmission failures j times in j different channels, it does not visit back to these channels until it successfully transmits the current packet. Since attackers also randomize their attack sequence, excluding these j channels, the probability that attackers will attack the new channel in the next slot is uniformly distributed over $M - j$ channels. Therefore, the probability of an OS-attack is,

$$Pr_{at|h,D} = \frac{m}{M-j}. \tag{5.11}$$

The transition probabilities from state D_j with action *hop* is,

$$\begin{aligned} Pr(T_1|D_j, h) =& (1-\rho)(1-\beta)^l(1 - P_{at|h,D}), \\ Pr(D_{j+1}|D_j, h) =& (1-\rho)\{1-(1-\beta)^l\} \\ &+ (1-\rho)(1-\beta)^l Pr_{at|h,D}. \end{aligned} \tag{5.12}$$

The transition probabilities from state D_j with action es is,

$$\begin{aligned} Pr(ES_0|D_j, es) &= (1-\rho)(1-\beta)^l(1-Pr_{at|h,D}), \\ Pr(ES_1|D_j, es) &= (1-\rho)\{1-(1-\beta)^l\}, \\ Pr(ES_a|D_j, es) &= (1-\rho)(1-\beta)^l Pr_{at|h,D}, \\ Pr(P|D_j, es) &= \rho. \end{aligned} \tag{5.13}$$

Here, the more successive attacks attackers can perpetrate, the higher the chance of successful attack in the next slot, i.e.,

$$Pr(T_1|D_j, h) > Pr(T_1|D_{j+1}, h). \tag{5.14}$$

5.3.3 Optimal Defense Strategy

An MDP consists of four components: a finite set of states, a finite set of actions, transition probabilities, and immediate rewards. We have modeled the defense problem as an MDP. Now, we can find the optimal defense strategy by solving it.

For an MDP, a *policy* is defined as the action to take in each state, i.e., $\pi : S_n \rightarrow a_n$. In other words, a policy maps each state $S \in \mathbf{X}$ to an action $a \in \mathbf{A}$ and is represented by $\pi(S)$. Among all possible policies, the optimal policy returns the maximum expected total discounted payoffs. The value of a state S is defined as the highest expected payoff, starting from the state S and represented as,

$$V^*(s) = \max_{\pi} E\Big[\sum_{n=1}^{\infty} \delta^{n-1} U(n)\Big| S = s\Big]. \tag{5.15}$$

Here, the optimal policy $\pi^*(S)$ returns the maximum expected payoff. One important point is that, after making a move from the current state, the remaining part of an optimal policy should still be optimal. Therefore, the first move should maximize the immediate payoff and the future expected payoff, which are conditioned on the

current action. This is called Bellman equation [117],

$$\begin{aligned} Q(S,a) &= \sum_{S'} Pr(S'|S,a)\Big(U(S,a,S') + \delta V^*(S')\Big), \\ V^*(S) &= \max_{Q} Q(S,a), \\ \pi^*(S) &= \text{ argmax } Q(S,a). \end{aligned} \tag{5.16}$$

Now, we can use the value iteration method to derive the optimal defense strategy and show that the solution has a structure mentioned in Proposition 1.

Proposition 1: The optimal policy can be represented by two critical states $k^* \in \{1, 2, \cdots, K\}$ and $g^* \in \{1, 2, \cdots, G\}$, i.e.,

$$\begin{aligned} \pi^*(T_i) &= \begin{cases} s, & \text{if } T_i < T_{k^*} \\ h, & \text{otherwise,} \end{cases} \\ \pi^*(D_j) &= \begin{cases} h, & \text{if } D_j < D_{g^*} \\ es, & \text{otherwise.} \end{cases} \end{aligned} \tag{5.17}$$

Proof: From (5.4) and (5.5), the probability of a successful transmission with action *stay* (i.e., $Pr(T_{i+1}|T_i, s)$) decreases over i. Therefore, from the definition of $Q(S,a)$ from (5.16), $Q(T_i, s) - Q(T_{i-1}, s) < 0$. Now, (5.9) indicates that the probability of a successful transmission with action *hop* (i.e., $Pr(T_1|T_i, h)$) increases over i. Therefore, $Q(T_i, h) - Q(T_{i-1}, h) > 0$. Now, the optimal action at state T_i is *stay* if $Q(T_i, s) \geq Q(T_i, h)$, or *hop* if $Q(T_i, h) \geq Q(T_i, s)$. Since $Q(T_i, s)$ is decreasing and $Q(T_i, h)$ is increasing, there exists a k^*, where $Q(T_{k^*-1}, s) \geq Q(T_{k^*-1}, h)$ and $Q(T_{k^*}, h) > Q(T_{k^*}, s)$, and $k^* \in \{1, 2, \cdots, K\}$. This concludes the first part of the proof.

Similarly, from (5.11)-(5.14), we can show that $Q(D_j, h) < Q(D_{j-1}, h)$ and $Q(D_j, es) > Q(D_{j-1}, es)$. Therefore, there exists a g^*, where $Q(D_{g^*-1}, h) \geq Q(D_{g^*-1}, es)$ and $Q(D_{g^*}, es) > Q(D_{g^*}, h)$, and $g^* \in \{1, 2, \cdots, G\}$. This concludes the second part of the proof.

Summary: An SU's strategy to use an underutilized channel *as long as plausible* and the *iterative process* of random-OS attack facilitates the design of the attack and defense problem as an MDP. The proposed defense can be summarized in two aspects: (1) an SU keeps utilizing an underutilized channel for k^* time slots and then hops to another channel, and (2) after g^* successive transmission failures, an SU takes the action *extra-sense*. In this chapter, we consider that the strategy of attackers remains unchanged, and the strategy of attackers can be learned over time. Nevertheless, an attack and defense problem is comparable to an arms race: the attacker and defender will change their strategies to outsmart each other.

5.4 Proposed Attack Inference Model

In this section, we propose an attack inference model to detect the presence of attackers. The proposed model has two features: 1) it utilizes the in-hand sensing history of the victim; hence, no networking overhead occurs to estimate PU parameters, and 2) it does not violate any policy and hardware constraints; hence, no policy change and extra hardware required. Depending on the parameters of the model, it helps the safeguard process to detect the presence of attackers.

In reality, it is impossible for a victim to know the exact network parameters (i.e., α, β, m) to devise the MDP. Therefore, an SU must learn the MDP over time. A *model-based* learning technique requires the Markov process to exhibit constant parameters over time, and it has a limitation in scalability; hence, a *model-free* learning is best suitable for this scenario. We employ the Q-learning technique that works as a model-free off-policy method, learns the game without the need of transition probabilities, and fits well with sudden changes in MDP parameters. Figure 5.7 shows the framework of the proposed attack inference model and Q-learning.

5.4.1 Q-learning

The Q-learning tries to approximate the unknown transition probability by the empirical distribution of states that have been experienced over time. It iteratively

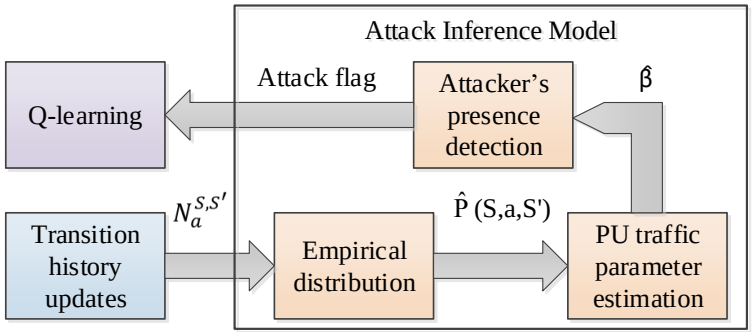


Figure 5.7: The Q-learning and attack inference model.

calculates and updates Q-value based on the state-action tuple (S, a, S').

$$\begin{aligned} Q_n(S,a) =& Q_{n-1}(S,a) + \gamma\big[\{R(S,a,S') + \delta V_n(S')\} \\ & - Q_{n-1}(S,a)\big], \\ V_n(S) =& \max_Q Q_n(S,a), \end{aligned} \tag{5.18}$$

where γ is the learning rate and δ is the discount factor.

In Q-learning, there is no fixed policy while learning the MDP and agents take random actions (with probability ϵ) to discover the MDP. However, the randomness decreases over time (i.e., $\epsilon \to 0$) and defenders are more likely to take actions with highest Q-values. After Q-values converge, the learning process ends. The optimal policy after the learning period is,

$$\pi^*(S) = \text{argmax}\, Q_n(S,a), \quad a \in \mathbf{A}, S \in \mathbf{X}. \tag{5.19}$$

In quest of learning the optimal policy, the defender makes mistakes and takes random decisions to explore the MDP. Hence, Q-learning engenders a cost in performance, and it is represented by *regret*, which quantifies the difference between the expected rewards (while learning) and the optimal rewards. Therefore, the more the defender learns, the fewer mistakes it makes (i.e., regret is a decreasing function of time).

Hence, to minimize the learning cost, the attack inference model reinitializes the

learning process (i.e., reinitialize ϵ) when the model detects the presence of OS-DoS attackers.

5.4.2 Attacker's Presence Detection

In this approach, benign SUs initiate their operation with three policies: 1) stay on the current channel until a transmission failure (i.e., $\pi(T) = s$) occurs, 2) hop to another channel after sensing the channel busy in the sensing interval (i.e., $\pi(P) = h$), and 3) hop to another channel after a transmission failure (i.e., $\pi(D) = h$). Without detecting the presence of OS-attackers, Q-learning does not employ the action es.

With recorded historic states and actions, SUs are able to compute the occurrences of transitions given any action. For example, the notation $N_a^{S,S'}$ represents the total number of transitions from state S to S', taking action a.

We define $T_p \triangleq \max\{T : N_s^{T_i,T_i+1} = 0\}$ (e.g., under-attack, $T_p = K$). From (5.5), we can understand that the absence of attack (i.e., $Pr_{at|s} = 0$) will result in an empirical probability $\widehat{Pr}(D_1|T_i, s) = \dfrac{N_s^{T_i,D_1}}{N_s^{T_i,D_1} + N_s^{T_i,P} + N_s^{T_i,T_i+1}}$ that is close to the probability of transmission failure by PUs only,

$$Pr(D_1|T_i, s, Pr_{at|s} = 0) = (1 - \widehat{\beta})\{1 - (1 - \widehat{\beta})^l\}, \tag{5.20}$$

where $\widehat{\beta}$ represents the PU traffic parameter from empirical observations, which will be explained later in this section.

Now, with the presence of attackers (i.e., $Pr_{at|s} > 0$), $\widehat{Pr}(D_1|T_i, s) > Pr(D_1|T_i, s)$. We represent this by,

$$X_i^n = \frac{\widehat{Pr}_n(D_1|T_i, s) - Pr_n(D_1|T_i, s; Pr_{at|s} = 0)}{Pr_n(D_1|T_i, s; Pr_{at|s} = 0)}, \tag{5.21}$$

where $\widehat{Pr}_n$ and Pr_n represent empirical probabilities after n time slots (i.e., $\widehat{Pr}_n$ and Pr_n are running parameters).

SUs track these values of X_i over time. From (5.4) and (5.5), we can observe that

$Pr_{at|s}$ increases with the residence time of SUs on a channel. Therefore, to deduce the presence of attackers, X_i values should conform to the requirement below,

$$X_1^n < X_2^n < \cdots < X_{p-1}^n < X_p^n. \tag{5.22}$$

This inequality characterizes the primary condition to detect the random-OS attack. It differentiates the random-OS attack from the naive attack where m attackers randomly choose m channels in each slot with equal probabilities (i.e., m/M), and it does not consider which channels have been detected in the past. Therefore, X_i^n will not meet the requirement in (5.22), instead, the values of X_i^n will lie within a close approximation,

$$X_1^n = X_2^n = \cdots = X_{p-1}^n = X_p^n \approx c, \tag{5.23}$$

where c is a constant.

Since each channel has an equal probability of encountering attack in the naive approach, hopping strategy cannot reduce the risk of attacks. Moreover, the hopping cost makes it a futile effort to avoid the attack by hopping from one channel to another. Hence, SUs stay on the same channel until they sense the PU reappearance or experience a transmission failure.

Next, we consider a safety margin τ to finally trigger the presence of attackers in the network. Besides a safety margin, τ also works as a trade-off parameter between performance and security. We compare the value of X_1^n to τ to decide the presence of attackers. Since the state T_1 is visited more frequently than other T states, we make an educated choice of comparing the safety margin with X_1^n. Therefore, the second requirement is,

$$X_1^n > \tau. \tag{5.24}$$

We can further control it by starting a counter when (5.22) and (5.24) are met, then triggering the attack flag once these requirements are consistently met for a certain time.

5.4.3 PU Traffic Parameter Estimation

We define $\mathbf{S} \triangleq \{T_1, T_2, \cdots, T_p - 1\}$ and $\mathbf{H} \triangleq \{P, D, T_p\}$. Now, given the state transition history $N_a^{S,S'}$ over time, we can deduce the empirical value of the PU traffic parameter,

$$\widehat{\beta} = \frac{\sum_{T \in \mathbf{S}} N_s^{T,P}}{\sum_{T \in \mathbf{S}} \left(N_s^{T,P} + N_s^{T,D} + N_s^{T,T+1} \right)}. \tag{5.25}$$

The empirical value of $\widehat{\beta}$ remains unaffected by the presence of attackers. It depends on the results from the sensing interval, and OS-attackers remain inactive during this interval. Therefore, (5.25) provides a close estimation of the actual PU parameter to decide the presence of attackers in the network.

Summary: Unlike previous research, we consider the absence and the presence of attackers. When attackers initiate an OS-DoS attack, the proposed attack inference model detects the attack and reinitializes the Q-learning process to minimize the regret (i.e., learning cost) and to take appropriate action.

5.5 Performance Evaluation

In this section, we present simulation results to evaluate the performance of our proposed research. Here, we consider that the victim SU detects an attacker, but does not oust it from the network; the appropriate attack response (e.g., network isolation, bandwidth limitation, and network elimination) is an open research issue. Unless otherwise stated, the simulation parameters are:

The presented simulation results are the average of 100 independent trials.

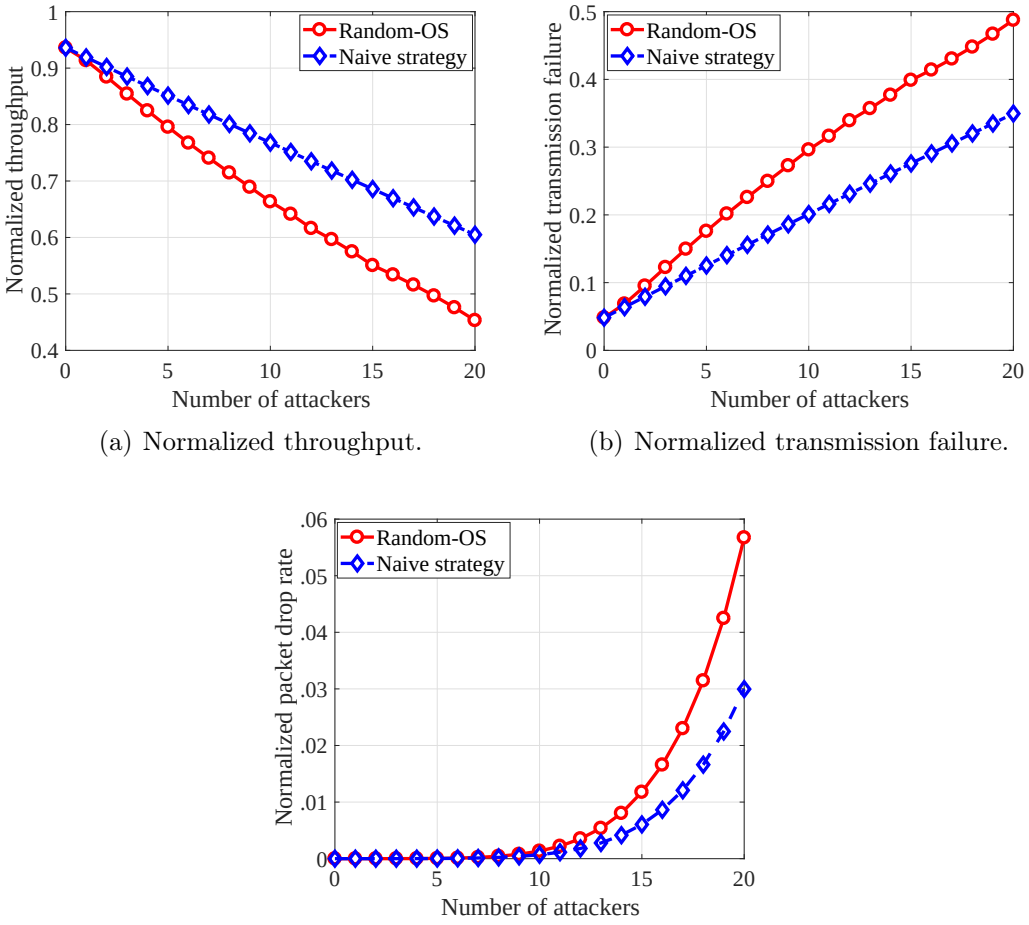
Table 5.1: Simulation Parameters: Hide and Seek

Parameter	**Value**
Communication gain, R	5
Cost of transmission failure, L	5
Hopping cost, C	1
Cost of busy channel, B	1
Penalty for policy violation, F	50
Maximum transmission attempt, G	7
Cost of packet drop, Q	$G \cdot L$
Reward for detecting an attacker, E	20
SU packet length l	5
Discount factor, δ	0.95
Learning rate, γ	$1/\sqrt{\text{number of time slots}}$
PU parameters	$\beta = 0.01$, $\rho = 0.1$
Number of channels, M	60

5.5.1 Random-OS Attack

In this research, we consider that attackers do not have any predetermined knowledge of the victim's channel-hopping sequence and operating channels. Therefore, we discard the comparison with conventional OS-DoS attack where the victim experiences null throughput regardless of the number of attackers (i.e., unrealistic scenario). Fig. 5.8 demonstrates the performance of the random-OS strategy in contrast to the naive approach, where attackers do not consider the knowledge of which channels have been visited in the past, instead randomly select channels at each time slot.

In Fig. 5.8(a), the normalized throughput is shown, where victims experience less throughput in the random-OS attack due to the iterative process and the re-randomization technique of random-OS. Likewise, victims of the random-OS attack suffer more transmission failures (Fig. 5.8(b)) and higher rate of packet drop (Fig. 5.8(c)). However, transmission failures are not enough to cause significant packet drop or DoS attack unless attackers can perpetrate it consecutively. This reflects in Fig. 5.8(c) where the packet drop rate follows a different trend than the rate of transmission failure; the packet drop rate starts to increase exponentially after



(a) Normalized throughput.

(b) Normalized transmission failure.

(c) Normalized packet drop rate.

Figure 5.8: Performance of the random-OS attack.

$m = 10$. Therefore, in this scenario, more than 10 attackers are required to cause significant damage to the victim.

5.5.2 Critical States

We demonstrate the critical states k^* and g^* of the optimal policy (Fig. 5.9) derived from the value iteration of the MDP, with the change in the number of attackers (m), the cost of transmission failure (L), the reward of attacker detection (E), the cost of channel-hopping (C), and the number of operating channels (M).

Effect of m: In Fig. 5.9(a)-(h), both k^* and g^* decrease with the increase in the number of attackers. As m increases, attackers can visit more channels in each time

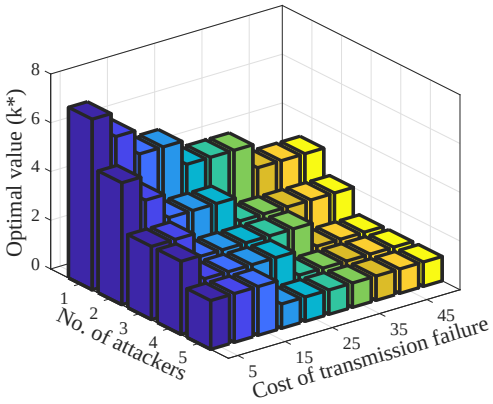
slot; hence, K starts to decrease, and SUs have less channels to hop on after each transmission failure. Therefore, the channel residence time decreases and SUs have to hop more frequently to avoid the attack.

Effect of L: In Fig. 5.9(a) and Fig. 5.9(e), as the cost of transmission failure L increases, SUs tend to hop more to avoid imminent transmission failures, thus k^* decreases. However, g^* demonstrates relatively less sensitivity towards changes in L due to the significantly high cost of Q. In transmission failure states, choosing action es over h means that the defender has to compromise its packet transmission regardless of the outcome of the action es; hence, the defender is reluctant to take action es.

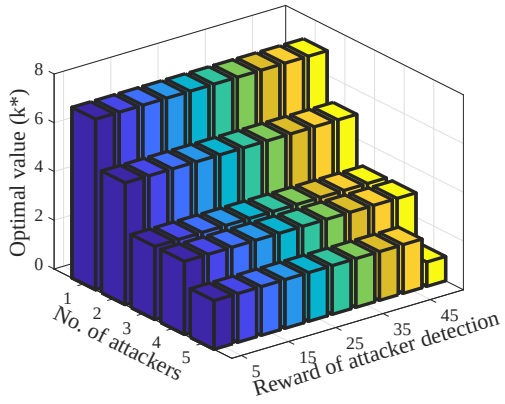
Effect of E: In Fig. 5.9(b), k^* remains almost insensitive to the change in the reward of attacker detection E. Because E largely dictates the action es only, *stay* and *hop* from transmission states remain out of its influence. For the similar reason, in Fig. 5.9(f), g^* illustrates linear sensitivity to the change in E. Therefore, as the reward for detecting an attacker increases, SUs become more motivated to take the action es instead of *hop*, to detect attackers. The parameter E works as a trade-off parameter between the networking performance and the security performance. Lower and higher values of E mean that victims have more tendency toward avoiding and victims have more tendency toward detecting OS-attackers, respectively.

Effect of C: As discussed in Section 5.3, channel-hopping engenders insignificant cost in terms of channel throughput; we quantify this cost by C. In Fig. 5.9(c), we can observe that k^* increases with C. As C increases, defenders become reluctant to take action *hop* and stays in a channel longer. Therefore, the cost of hopping significantly impacts the proposed defense strategy because defenders become limited in their capability to utilize the channel diversity a multi-channel network has to offer. However, unlike k^*, g^*—though exhibits very low sensitivity—decreases with C (Fig. 5.9(g)).

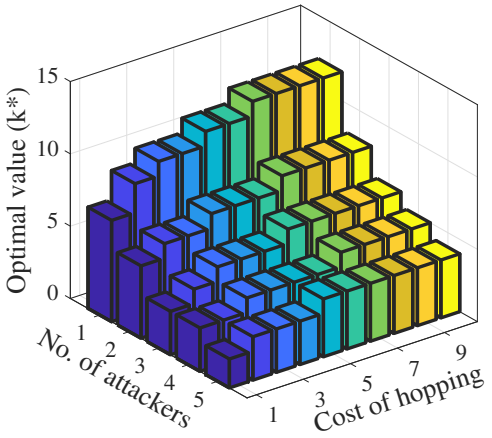
Effect of M: As the number of channels M increases, the maximum channel residence time K increases. Therefore, attackers have more channels to sweep through and defenders have more time to stay on a channel. In Fig. 5.9(d), we can observe that k^* increases linearly with the increase of M. Similarly, as M increases, defenders experience more incentive to hop through different channels than to detect attackers. As a result, g^* increases with M.



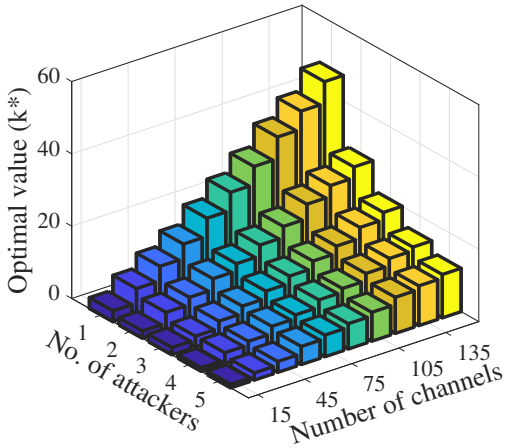
(a) Optimal value k^*; m vs. L.



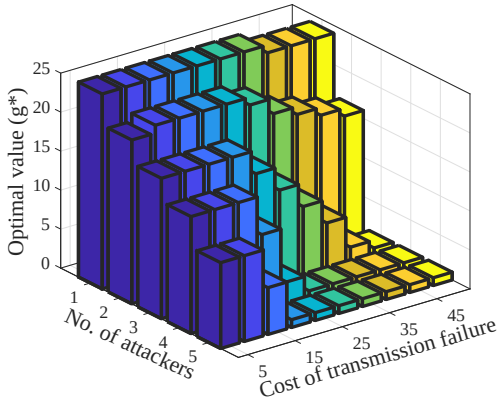
(b) Optimal value k^*; m vs. E.



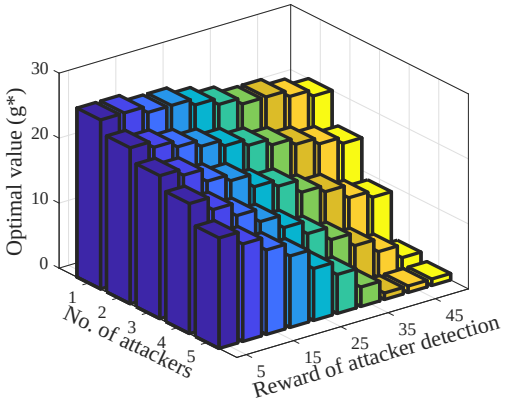
(c) Optimal value k^*; m vs. C.



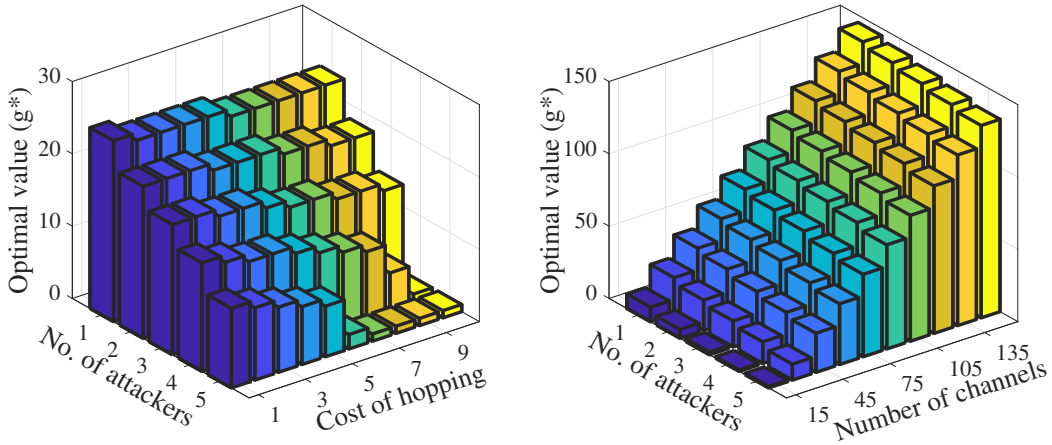
(d) Optimal value k^*; m vs. M.



(e) Optimal value g^*; m vs. L.



(f) Optimal value g^*; m vs. E.



(g) Optimal value g^*; m vs. C.

(h) Optimal value g^*; m vs. M.

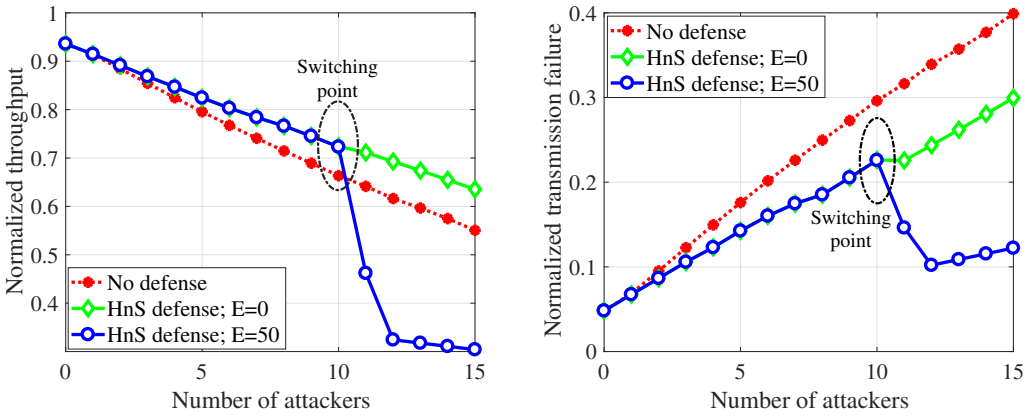
Figure 5.9: The sensitivity of optimal values to the changes in L, E, C, and M.

5.5.3 Hide and Seek

Fig. 5.10(a) compares the performance of our proposed hide and seek strategy with three scenarios: no defense, hide and seek with no reward ($E = 0$), and hide and seek with a high reward ($E = 50$). It illustrates that both $E = 0$ and $E = 50$ follow the same line until the number of attackers surpasses $m = 10$ (when $E = 50$); the throughput drops below the no defense line afterwards. We call this moment the *switching point* after which the victim prefers to detect attackers (using the action es) rather than avoiding them (using the action hop); hence, the throughput drops. As E increases, the victim becomes more motivated to detect attackers and the switching point moves to the left. As discussed earlier, E works as a tuning parameter between the networking and security performance. Likewise, in Fig. 5.10(b), we can observe that the transmission failure decreases after the switching point. However, after $m = 12$, it starts to increase again due to the increasing number of attackers.

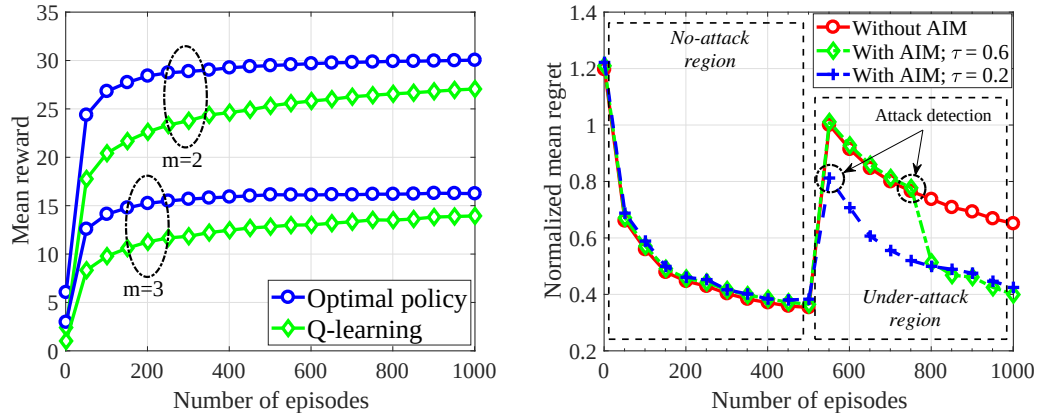
5.5.4 Q-Learning and Attack Inference Model

We evaluate the performance of Q-learning (Fig. 5.11 (a)) by showing the difference in mean reward after each episode between an SU that knows the optimal values and



(a) Normalized throughput. (b) Normalized transmission failure.

Figure 5.10: Performance of hide and seek.



(a) Q-learning. (b) Attack inference model.

Figure 5.11: Performance of Q-learning and attack inference model.

an SU that learns the MDP over time via Q-learning. Here, we can observe that in both cases (i.e., $m = 2$ and $m = 3$), the reward converges to the optimal reward. However, with $m = 3$, the agent converges more quickly due to the fewer amount of states.

In Fig. 5.11(b), the performance of our proposed attack inference model is shown with different values of the threshold τ. We change the scenario from $m = 0$, $M = 10$ to $m = 2$, $M = 10$ at *epsiode* $= 501$. As the MDP progresses, an SU takes fewer random actions (i.e., ϵ decreases); hence, it takes more time to track the changes without the assistance from the attack inference model. The proposed model assists the Q-learning to detect changes in the MDP and re-initializes the parameter ϵ to minimize the regret based on the threshold τ. With $\tau = 0.2$ and $\tau = 0.6$, the attack inference model detects the presence of the attacker on *epsiode* $= 549$ and *epsiode* $= 753$, respectively. Hence, a lower value of τ assists the SU to track the changes sooner and yields in less regret.

CHAPTER 6: PROPOSED COVERT SPECTRUM HANDOFF ATTACK IN CR NETWORKS

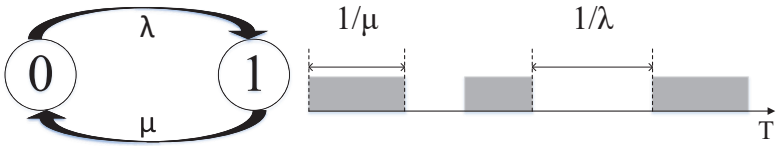
In this chapter, a vulnerability in proactive spectrum handoff processes is discussed and an attack model is proposed.

6.1 System Model

PU and SU Model: We consider that all PUs are under the sensing range of SUs and that PUs do not interfere with each other's transmission. Here, M channels (i.e., M PUs) have different service rates, and a PU randomly selects a channel to access. N SUs can opportunistically access these M channels. An SU can access a channel when it senses no PU is using it. In addition, an SU can detect a collision with a PU only after the SU finishes the frame transmission (e.g., if an ACK is not received). After detecting a collision, the SU stops transmitting on the current channel and initiates a spectrum handoff. Each SU is equipped with one radio for spectrum sensing and one radio for control information exchange and data transmission. The sensing radio has two key functions: 1) observe the channel usage characteristics and store the channel statistics to predict future channel activity and 2) confirm that the newly selected channel is idle for the transmission of SU.

Each PU alternates between the ON and OFF state according to a continuous-time Markov process. In Figure 6.1, let λ and μ denote the transition rate from the OFF to ON state and from the ON to OFF state, respectively. Thereby, the mean sojourn time in the ON and OFF states is $1/\mu$ and $1/\lambda$, respectively, and both follow the exponential distribution.

Network Coordination Scheme: Rendezvous is a prerequisite in establishing



(a) Transition rate. (b) PU activity in the time line.

Figure 6.1: PU activity model.

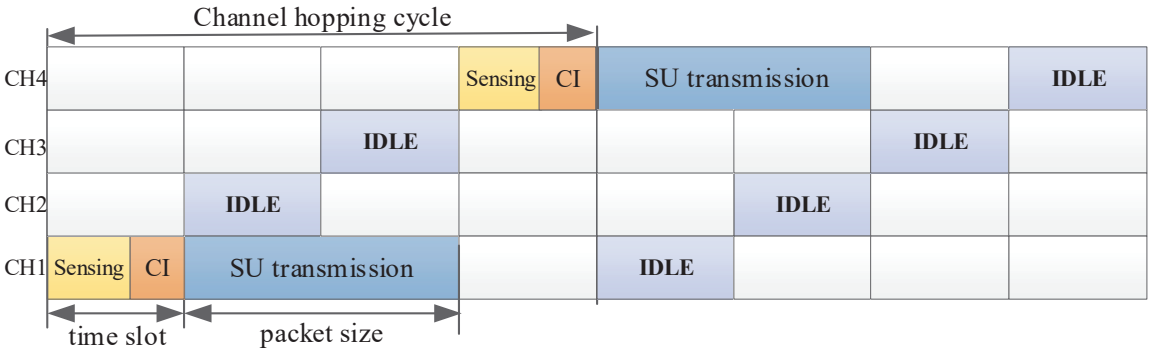
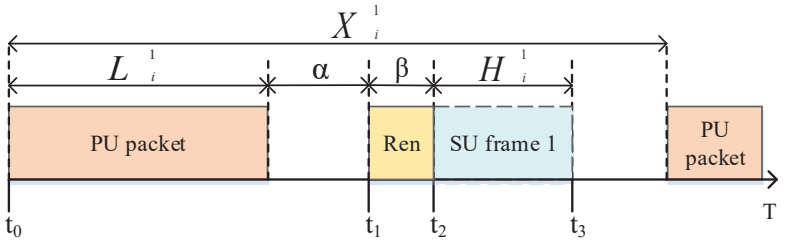


Figure 6.2: Network coordination scheme.

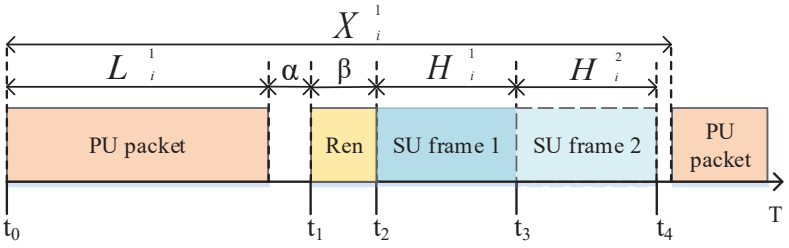
a connection between two SUs in the absence of a dedicated CCC. A successful rendezvous happens when both transmitting and receiving SUs are on the same channel and have completed a successful handshake between them, e.g., a Request-to-Send/Clear-to-Send (RTS/CTS) exchange.

We consider the common channel-hopping as the network coordination scheme [27], which means that the hopping pattern is the same for all SUs. Figure 6.2 illustrates the operation of the common frequency-hopping network coordination. We consider a time-slotted system. Each time slot consists of a sensing interval (sensing) and a contention interval (CI) with the transmission of an RTS/CTS pair. When there is no packet in the buffer of an SU, it keeps hopping through the channels from one time slot to another, based on the predetermined common channel-hopping pattern.

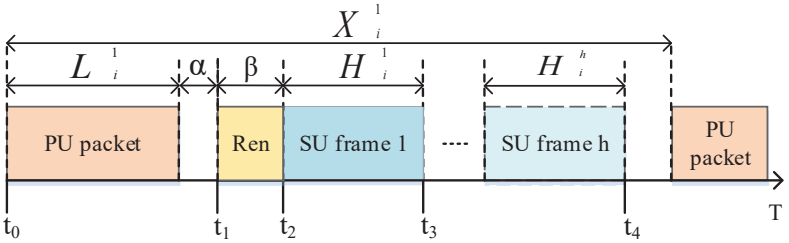
Then, we adopt the MAC model from [17] for network coordination. Whenever an SU has a packet to send, it first senses the channel. If the channel is idle, the SU chooses a random number (in terms of mini-slots) as its backoff time to avoid contention.



(a) SU sending the 1st frame.



(b) SU sending the 2nd frame.



(c) SU sending the hth frame.

Figure 6.3: PU and SU activity on channel i.

Proactive Spectrum Handoff Model: Proactive spectrum handoff helps to decrease the unwanted interference between PUs and SUs. In this section, we briefly discuss the proactive model we use in our simulations.

We consider that each SU calculates the likelihood of PU reappearance after performing a successful rendezvous. Using the sensed channel statistics, an SU can predict the channel availability before the transmission of the current frame ends. Based on the prediction, an SU decides whether to transmit on the current channel, switch to another channel, or pause the on-going transmission and remain on the current channel. We set a threshold (τ) for PU reappearance, above which an SU will not initiate the transmission. Figure 6.3 shows the PU and SU traffic activity on channel i, where X_i^k represents the inter-arrival time of the k^{th} PU packet on channel

i. L_i^k and H_i^k denote the length of the k^{th} PU and k^{th} SU packet on channel i, respectively. Here, t_0 represents the last sensed arrival of a PU packet and $N_i(t)$ denotes the status of PU reappearance within time t. $N_i(t)$ is a binary random variable with values 0 and 1, representing no PU reappearance and PU reappearance, respectively. As shown in Figure 6.3(a), the probability that channel i will be idle till the first frame ends (t_3) is given by,

$$Pr(N_i(t_3) = 0) = Pr(X_i^1 > L_i^1 + \alpha + \beta + H_i^1). \tag{6.1}$$

where β and α represent the time to successfully perform a rendezvous (i.e., 1 time slot), and the time between when the PU packet finishes the transmission and the rendezvous starts, respectively. In Figure 6.3(b), the probability that channel i will be idle till the second frame ends (t_4) is given by,

$$Pr(N_i(t_4) = 0) = Pr(X_i^1 > L_i^1 + \alpha + \beta + H_i^1 + H_i^2). \tag{6.2}$$

Therefore, the probability that an SU successfully transmits a packet on channel i, consisting of h frames, (Figure 6.3c) is,

$$Pr(N_i(t_4) = 0) = Pr(X_i^1 > L_i^1 + \alpha + \beta + \sum_{l=1}^{h} H_i^l). \tag{6.3}$$

Hence, based on the above predictions, the probability that an SU will handoff to a new channel is,

$$Pr(N_i(t) = 1) = 1 - Pr(N_i(t) = 0) > \tau. \tag{6.4}$$

Here, we consider the same threshold to make the decision on whether to switch from the current channel and to select a target channel. Every transmission on a new channel must be preceded by a sensing and contention attempt (i.e., rendezvous). In addition, the highest priority to access channels is given to handoff SUs to maintain

low handoff delays.

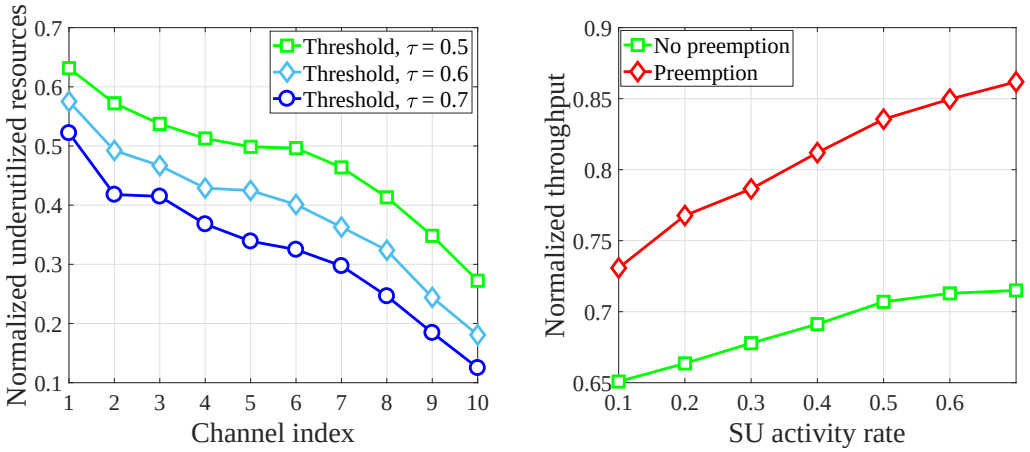
6.2 Covert Spectrum Handoff

Although the distributed nature of proactive spectrum handoff processes provide such protocols with significant performance gain in terms of avoiding collisions with PUs, it also exposes such approaches to new security vulnerabilities (e.g., covert spectrum handoff). In this section, we first identify the motivating reasons to exploit this vulnerability and then discuss the strategy of an attacker. We consider that a selfish SU is compromised, is authorized to use the secondary network, and has similar hardware configurations as benign SUs. To exploit this vulnerability, both transmitter and receiver SU must act selfishly. Throughout this chapter, we will use the term *attacker* and *selfish SU* interchangeably.

6.2.1 Vulnerability Analysis

Here, we shed light on the reasons behind this vulnerability and how a selfish SU can remain undetected in current proactive approaches.

Underutilized Radio Resources: Previous works on rendezvous ([17, 28, 19]) focus only on achieving guaranteed and fast rendezvous. However, the radio resource utilization is not considered as a performance metric. SUs' waste radio resources in the rendezvous process until they successfully handshake with each other. Figure 6.4(a) shows the amount of wasted radio resources in the common-hopping sequence-based rendezvous system in saturated SU traffic (i.e., SUs always have a packet to send). We consider non-identical service rates for each channel ranging from 1 to 10 Mbps for channel-1 to channel-10, respectively (i.e., channel-1 offers the lowest service rate and channel-10 offers the highest). Here, we show the normalized wasted radio resources of each channel and vary the value of threshold (τ) to observe the channel wastage trend. It clearly exhibits the ramifications of using the periodic hopping sequence approach, even with a higher threshold and saturated SU activity.



(a) Underutilized resources. (b) Conventional vs. prompt.

Figure 6.4: The motivating reasons behind the attack.

Less Prompt in Handoff Initiation: As we discussed in the introduction, most proactive handoff processes trigger handoff only when the next frame is likely to collide with a reappeared PU on the channel. As they consider identical channels and emphasize on reducing the delay constituting from handoff operations, prior handoff processes are inherently reluctant to handoffs. However, if we consider non-identical channels, it is likely to manage a trade-off between the delay constituting from switching to a faster channel and the service rate of that channel. In this process, an SU can initiate the handoff process instantly after the rendezvous (if a faster channel is available to off-set the handoff delay), and we call it preemptive proactive handoff process. In Figure 6.4(b), we can observe a significant increase in the normalized throughput between the conventional and preemptive proactive handoff process. Here, the preemptive process considers non-identical channels, likelihood of PU reappearance, and channel bandwidth as handoff criteria. Therefore, this finding indicates that the preemptive trigger offers a sizable performance gain for an attacker if it exploits this vulnerability.

Absence of a Central Entity: The absence of a central entity in CRAHNs makes it difficult to detect an attacker with selfish intentions. Current research on the

detection and defense of deviant behaviors in distributed networks are based on long-term monitoring of neighboring nodes and exchanging this monitored information with each other to make a consensus [118, 119, 62]. However, this is difficult to perform in a network without a dedicated CCC, especially when the attacker and defenders are not on the same channel. Here, the attacker avoids detection by covertly utilizing channels that are not currently used by any SUs.

These three aspects of distributed CRNs can motivate an SU to deviate from established protocol and to act selfishly.

6.2.2 Attacker Model

The attacker acts benignly during the hopping process to avoid suspicion. It starts to exploit the vulnerability only after performing a successful rendezvous (Figure 1.4(b)). According to the common-hopping process, an SU pair stays on the rendezvous channel and initiates a transmission, and other SUs hop to the next channel in the sequence. Prior defense techniques against selfish SUs work only if they would stay on the same channel. Therefore, the integral part of remaining undetected is to handoff to a channel that is not used by any other SU (e.g., the subsequent channel in the hopping sequence).

However, after a successful rendezvous, the attacker pair tries to search for a better channel to switch. The strategy of the proposed attack model to exploit the vulnerability in proactive spectrum handoff processes is given in Algorithm 3. In Algorithm 1, we include the additional strategy of the selfish attacker only, and the general processes are not included.

Algorithm 3 Attacker's Activity

Input: hopping sequence S, time t, PU reappearance threshold τ

Result: *decision*

$rendezvous\_status := unsuccessful$;

while $rendezvous\_status = unsuccessful$ **do**

  $i := (t-1)\%length(S)+1$; ▷ follow channel-hopping

  $currnet\_channel := S(i)$;

  if *current_channel=free & contention_status=win* **then**

    $rendezvous\_status := successful$;

    $r_{ch} := current\_channel$;

  else

    $t := t+1$; ▷ proceed to the next time slot

  end

end

$C :=$ Compute Target Channel(S,t,r_{ch});

decision := Handoff Preemption (C,r_{ch},τ);

Preceded by a successful rendezvous, the attacker pair tries to find a suitable target channel. Algorithm 4 shows the pseudocode of the channel selection process. It first sorts all the channels according to the prediction of PU reappearance and service rate in each channel, then starts checking them one by one to select the most suitable target channel.

Algorithm 4 Computing the Target Channel for Selfish SU

Input: hopping sequence S, time t, rendezvous channel r_{ch}

Result: target channel C

Function : Compute Target Channel(S,t,r_{ch})

$CH := sort\, channels\, according\, to\, the\, likelihood\, of\, PU\, reappearance\, and\, service\, rate$;

$j := 1$;

while $CH(j) = busy \| CH(j) = next\, channel\, in\, sequence$ **do**
| $j := j + 1$;

end

return $C := CH(j)$;

The criteria for selecting the most suitable channel is described in the steps below.

Less Likely to Be Affected by Reappeared PUs: By utilizing the in-hand resources of proactive handoff, the attacker pair can calculate the probability of PU reappearance in each channel. Then, they will try to reserve the channel that offers the least likelihood to be affected by a returning PU.

Faster Service Rate: An attacker's motive is to finish packet transmission sooner and to maximize its own channel utilization. After performing a successful rendezvous on channel i, the target channel j needs to maintain the inequality condition,

$$\epsilon_{ij} + L_j < L_i, \tag{6.5}$$

where L_i and L_j represent the packet length of the attacker in channel i and j, respectively, and ϵ_{ij} represents the delay of performing a handoff from channel i to channel j.

Not Being Used by Other SUs: The attacker pair will avoid channels that are already being used by other SUs. However, such avoidance can ensure the availability of a corresponding channel only in the current time slot and there is a probability that another SU might handoff to the same channel in the next time slot; hence, they need to contend to reserve the channel. Moreover, the attacker will not handoff to

the channel that comes next in the hopping sequence. In Figure 1.4(b), the selfish SU would not handoff to CH2 from CH1 to avoid suspicion.

Finally, the handover decision of the attacker pair depends on the target channel. If they are currently operating on the best available channel, then they do not perform a handoff. Algorithm 5 shows the pseudocode of the handoff preemption decision process. The attacker pair will handoff to a channel only if the channel satisfies the earlier mentioned criteria. Otherwise, the attacker pair will stay on the current channel.

Algorithm 5 Handoff Preemption Decision

Input: target channel C, rendezvous channel r_{ch}, threshold τ

Result: *decision*

Function : Handoff Preemption (C,r_{ch},τ)

if C=r_{ch} **then**
| $handoff\ \ preemption := 0$; ▷ no preemption

else
| $handoff\ \ preemption := 1$; ▷ activate preemption

end

if *handoff preemption=0* **then**

if *prediction*(r_{ch}) $\leq \tau$ **then**
| $decision := begin\, transmission$;

else
| $decision := stay\, idle\, and\, wait\, for\, a\, better\, channel$;

end

else

if *prediction*(C) $\leq \tau$ **then**
| $decision := handoff\, to\, C$;

else
| $decision := stay\, idle\, and\, wait\, for\, a\, better\, channel$;

end

end

return *decision*

In our model, we consider that the attacker pair initiates a preemptive handoff only after rendezvous, and they refrain from searching for better channels for each successive frames. Otherwise, attackers are likely to lose their opportunity to transmit, to become trapped in a loop of handoffs, and to increase handoff delay significantly.

6.3 Performance Analysis

In this section, we evaluate the impact of the proposed covert spectrum handoff by conducting extensive simulations. The parameters used to obtain the simulation

results are listed in Table I. The arrival of PU and SU packets follow the Poisson process, and the length of PU and SU packets are exponentially distributed and fixed, respectively. As the attacker pair initiates its malicious act only after the rendezvous and does not continue it for each subsequent transmitting frame, we consider a packet as 1 frame length long to analyze the performance. In the simulation, we consider one pair of attacker if not stated otherwise.

Table 6.1: Simulation Parameters: Covert Spectrum Handoff

Simulation area	500x500
Simulation time	50 sec
The number of PUs	10
The number of SUs	50
Number of channels	10
Channel data rate	1-10 Mbps
The size of (RTS+CTS)	160 + 112 bits (802.11b/g)
PU ON time	0.5 (uniform for all PUs)
SU traffic rate	0.1-0.7 (uniform for all SUs)
Length of a time slot	1.5ms
Frame length	1-10 time slot long

Increased Average Throughput: One important metric to evaluate the performance of this attack is throughput. In Section III, we discussed the difference in throughput between the conventional and the preemptive handoff initiation. However, earlier we considered that all SUs follow the preemptive handoff initiation process. In this section, we consider that only the attacker pair follows the preemptive handoff initiation process, and benign SUs follow the conventional process.

Fig. 6.5 illustrates a throughput gain $(19 - 30\%)$ by the attacker pair compared to benign SUs. As the attacker pair preempts handoff process and reserves a channel with faster service rate earlier, it experiences significantly higher throughput. Moreover, benign SUs experience less room to utilize faster channels as the attacker pair utilizes faster channels more often. Therefore, we can observe an increase in attackers performance from Fig. 6.4(b) to Fig. 6.5.

Higher Channel Utilization of Faster Channels: As discussed, the attacker pair

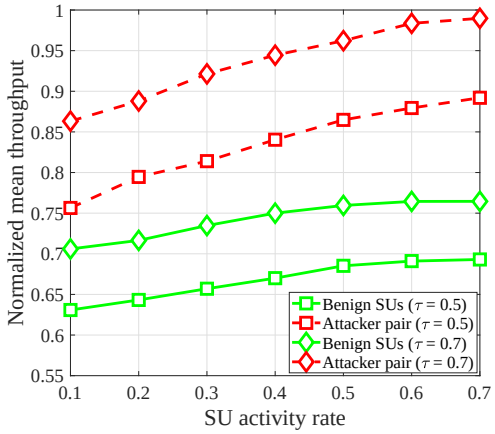
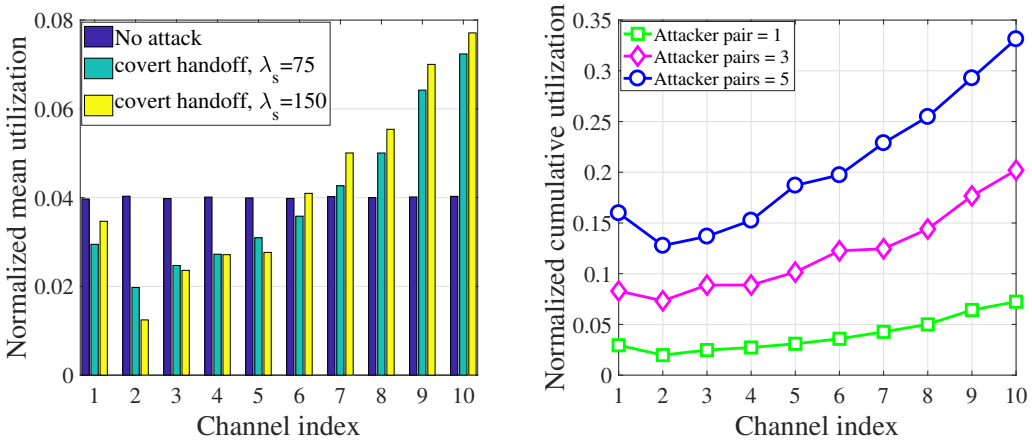


Figure 6.5: Normalized average throughput of benign SUs vs. the attacker pair.



(a) With vs. without attack.

(b) Increasing attacker pair.

Figure 6.6: Normalized average channel utilization of benign SUs and attackers.

always tries to reserve the best available channel by initiating the handoff process earlier (i.e., preemption). In Fig. 6.6(a), we can observe the channel utilization by the the attacker pair in no-attack and attack scenarios. In the benign scenario, they use all the channels uniformly, and this uniformity represents the fairness in the system. However, after they become selfish (i.e., preemption in handoff), the utilization of faster channels by the attacker pair increases. In addition, we observe an increase in the utilization, as we increase the traffic rate of selfish SUs.

In Fig. 6.6(b), the impact of increasing the number of attackers on the channel utilization is shown. As the number of attackers increases, they occupy the faster channels more, and it increases the cumulative channel utilization of attackers. If we consider 5 pairs of attackers among 50 SUs, then it shows the utilization of channel-10, approximately 33% (i.e., 20% nodes are utilizing 33% radio resource).

Higher Collision Avoidance: In the process of increasing the throughput, the attackers are inherently avoiding collisions with reappeared PUs. As the covert spectrum handoff (or preemptive handoff) happens only to a faster channel that offers less probability of PU reappearance, attackers increase their throughput and ensure less collisions from PUs. In Fig. 6.7, the collision rate with PUs are shown and we can observe a reduction in the collision rate between PUs and SUs.

Handoff Delay: We observe a reduction in the average handoff delay of the attacker pair compared to benign SUs. Though it might seem that attackers perform more handoffs, their propensity toward faster channels with lower PU reappearance probability ensures that they experience fewer handoffs later in the transmission time. In Fig. 6.8(a), the normalized average delay of the benign and selfish SUs are demonstrated. Here, channel index represents the channels that handoff initiated from, not the target channel. Therefore, it indicates that more handoff takes place in slower channels, which is expected.

Moreover, in Fig. 6.8(b), as the number of attackers increases, they occupy faster

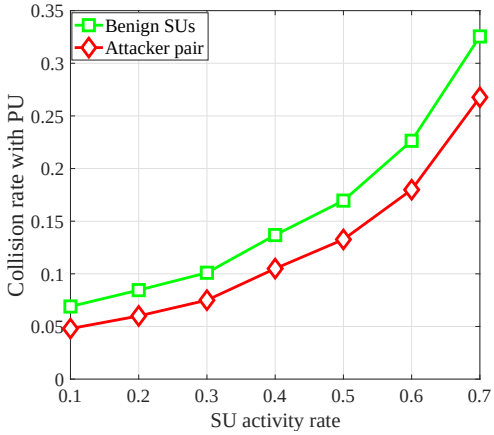
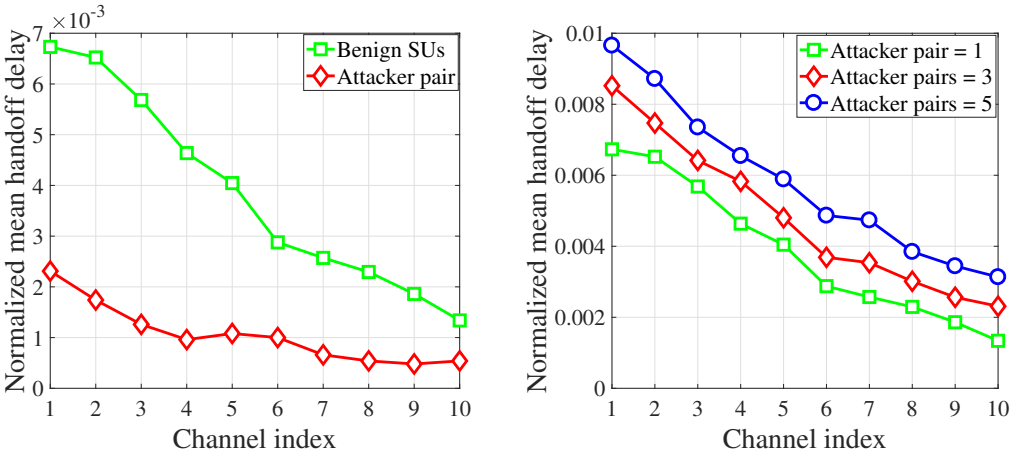


Figure 6.7: Normalized average collision rate of benign SUs vs. attackers.



(a) Benign SUs vs. selfish SUs.

(b) Delay of benign SUs.

Figure 6.8: Normalized average handoff delay of benign SUs and selfish SUs.

channels more. Therefore, benign SUs are deprived from utilizing faster channels, and sometimes they are forced to stop transmissions due to the unavailability of a channel. Moreover, as handoff SUs are given higher priority to access a channel, benign SUs lose contention to attackers; hence, benign SUs waste more time in the handoff process to transmit each packet.

CHAPTER 7: PROPOSED HIDDEN TERMINAL EMULATION ATTACK

This chapter proposes a novel attack in spectrum coexistence between heterogeneous networks.

7.1 What is Hidden Terminal

In a wireless network, hidden terminals refer to the wireless devices that are out of each other's radio range but have a common exposed neighbor that is inside both of their radio range. Fig. 7.1 provides an illustration of a hidden terminal scenario where A and C are both hidden from each other, and they create interference at the exposed node B when they try to transmit at the same time in the same channel. It is a natural phenomenon, and the dense deployment of IoT devices will drastically increase the occurrence of such scenarios. To manage this problem, request-to-send/clear-to-send (RTS/CTS) handshaking is employed in IEEE 802.11.

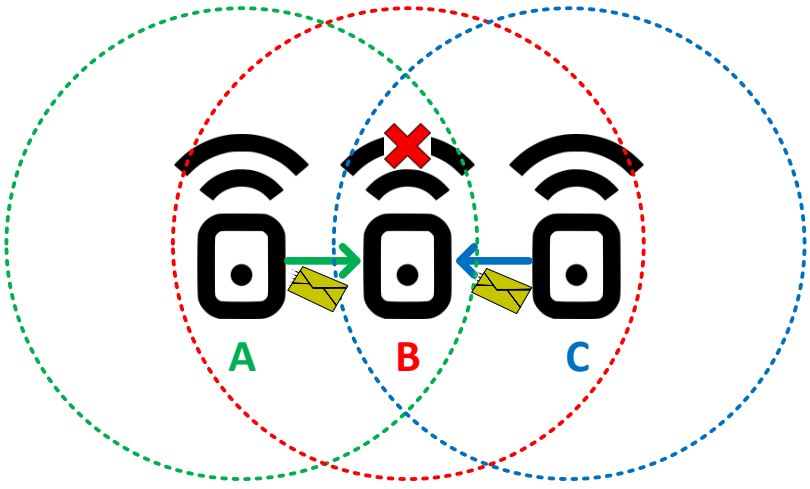


Figure 7.1: Hidden terminal interference between wireless nodes.

7.2 Proposed Random-HTE Attack: The Reconnaissance and Emulation Phase

In this phase, the goal of an HTE attacker is to successfully emulate the radiation characteristics of a benign hidden-terminal, which indicates that the attacker must spoof a location from where it can behave like a hidden terminal to the intended transmitters, however, still can listen to their transmissions. However, realizing this phase is not possible with conventional omni-directional antennas because the path-loss vector is similar at each direction. Therefore, we propose to use *smart antenna's beamforming capability* to mimic the signal characteristics of the spoofed location. To achieve this, the attacker first obtains the geometric locations of the IoT devices by wardriving [120] and other off-the-shelf techniques, such as the angle-of-arrival or the distance to the transmitter. Then, it deduces an optimal antenna configuration that enables the emulation of the intended hidden-terminal location.

In reality, IoT devices are unlikely to have sophisticated tools to analyze RSS readings received from IoT devices of external networks. Therefore, the attacker is not required to mimic the exact RSS signature, rather it focuses on maintaining an average signal strength equal to or above R_{th} (i.e., the receiver sensitivity threshold) at the exposed node(s) and an average signal strength lower than S_{th} (i.e., the carrier sensing threshold) at the hidden nodes. Now, the question is, *whether such antenna configuration is feasible that can facilitate the hidden-terminal emulation from the attacker's current physical location.* In the following, the quest for this answer begins.

In reality, IoT nodes are unlikely to have sophisticated tools to analyze RSS readings that are received from different IoT nodes of different networks, which makes low-powered, computationally limited IoT nodes more vulnerable to the HTE attack. In this case, the attacker is not required to mimic the RSS signature, rather it focuses on maintaining a signal level above the receiver sensitivity threshold. The remainder of this research considers that the HTE attacker is equipped with a circular smart antenna array that consists of I_{ele} isotropic elements put on a circle with a radius r,

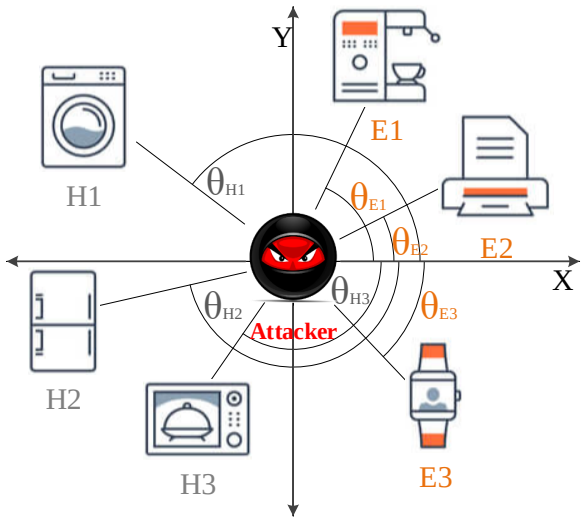


Figure 7.2: Feasibility test of the HTE attack.

and the i^{th} antenna element is placed with a phase angle ϕ_i. The beamforming-pattern for the circular smart antenna is characterized by,

$$G(\theta) = \sum_{i=1}^{I_{ele}} w_i \exp\left[j\frac{2\pi}{\lambda} r\cos(\theta - \phi_i)\right], \tag{7.1}$$

where λ is the signal wavelength, θ represents the direction to the respective IoT node, and $\mathbf{w} = [w_1, w_2, \cdots, w_{I_{ele}}]$ is the complex weight vector that can be tuned to change the radiation pattern. A circular array antenna can produce flexible asymmetric radiation patterns and can deflect a beam through 2π. Nonetheless, our analysis is not limited to circular antenna array; a different antenna model with a different geometric form can be incorporated by replacing its corresponding beamforming pattern equation in (7.1).

However, designing such an attack model requires realistic constraints to consider, such as smart antenna design and relative distances to each IoT node. Therefore, it is probable that not all locations are feasible to perpetrate this attack.

7.2.1 Problem Overview

To understand how we analyze the condition that an attacker at a certain location can launch the HTE attack, let us look at the illustration in Fig. 7.2 where an attacker is trying to reveal its transmission to nodes $E1$, $E2$, and $E3$ and to hide its transmission from nodes $H1$, $H2$, and $H3$. To reduce the radio coverage at unwanted directions and to steer the radio transmission to intended directions, we use log-distance path-loss model to infer the mean RSSs at given distances. According to the log-distance path-loss model, the mean path-loss at distance d is,

$$PL_d(dB) = 10\alpha \log_{10} d + PL_{d_0}(dB), \tag{7.2}$$

where PL_{d_0} is the path-loss at the reference distance $d_0 = 1m$ and α is the path-loss exponent. Moreover, the path-loss at distance d can be expressed as,

$$PL_d(dB) = P_0(dBm) - P_{R_d}(dBm), \tag{7.3}$$

where P_0 is the required transmission power to keep a good connection with the receiver if omni-directional antenna is used and P_{R_d} is the received signal strength at distance d. Therefore, combining (7.2) and (7.3), we have,

$$P_{R_d} = \frac{P_0}{PL_0 d^{\alpha}}. \tag{7.4}$$

For a smart antenna with a steering capacity, the transmission power in direction θ is represented by $P_0|G(\theta)|^2$ instead of P_0. So we rewrite (7.4) as,

$$P_{R_d}(\theta) = \frac{P_0|G(\theta)|^2}{PL_0 d^{\alpha}}, \tag{7.5}$$

where $P_{R_d}(\theta)$ represents the received signal strength at distance d along the direction θ. Now, for e exposed nodes and h hidden nodes,

$$P_{R_{d_i}}(\theta_{E_i}) = \frac{P_0|G(\theta_{E_i})|^2}{PL_0 (d_{E_i})^{\alpha}}, \tag{7.6}$$

$$P_{R_{d_j}}(\theta_{H_j}) = \frac{P_0|G(\theta_{H_j})|^2}{PL_0(d_{H_j})^\alpha}, \tag{7.7}$$

where $i = 1, 2, \cdots, e$ and $j = 1, 2, \cdots, h$.

From (7.1), the beamforming directional gain in the direction of the i_{th} node can be written as,

$$|G(\theta_{E_i})|^2 = |\mathbf{w}\mathbf{c_i}|^2, \tag{7.8}$$

where

$$\mathbf{c}_i = \begin{bmatrix} \exp[j\frac{2\pi}{\lambda}r\cos(\theta_{E_i} - \phi_1)] \\ \exp[j\frac{2\pi}{\lambda}r\cos(\theta_{E_i} - \phi_2)] \\ \cdot \\ \cdot \\ \cdot \\ \exp[j\frac{2\pi}{\lambda}r\cos(\theta_{E_i} - \phi_{I_{ele}})] \end{bmatrix}. \tag{7.9}$$

Letting,

$$\begin{aligned} \mathbf{h}_i &= \big[\frac{P_0}{PL_0(d_{E_i})^\alpha}\big]^{\frac{1}{2}}\mathbf{c}_i, \quad i = 1, 2, \cdots, e \\ \mathbf{g}_j &= \big[\frac{P_0}{PL_0(d_{H_j})^\alpha}\big]^{\frac{1}{2}}\mathbf{c}_j, \quad j = 1, 2, \cdots, h, \end{aligned} \tag{7.10}$$

the feasibility of the HTE attack can be modeled as,

$$\begin{aligned} &\text{find any} && \mathbf{w} \\ &\text{subject to} && |\mathbf{w}^H\mathbf{h}_i|^2 \geq R_{th}, \;\; i = 1, \cdots, e \\ & && |\mathbf{w}^H\mathbf{g}_j|^2 < S_{th}, \;\; j = 1, \cdots, h \end{aligned} \tag{7.11}$$

where R_{th} and S_{th} represent the receiver sensitivity and carrier sensing threshold, respectively. It can be seen that the above problem belongs to the class of quadratically constrained quadratic programming (QCQP) problems. The constraints are concave homogeneous quadratic constraints. The problem contains a special case of

the problem considered in [121]; hence, it is NP-hard.

7.2.2 Solving HTE Problem

As the feasibility problem of HTE defined in (7.11) is an NP-hard problem, it is not possible to analyze the properties of HTE by directly solving it. Therefore, we first formulate the derivation of a relaxed problem, which will incorporate a solution that provides an upper bound for the feasibility answers to the HTE problem; that is, if the relaxed problem is infeasible, (7.11) is definitely infeasible. Afterward, we provide a randomization technique, which in most of the cases finds a feasible solution through a local search around the point generated by the relaxed problem. This randomization algorithm essentially serves with a lower bound on the HTE problem (7.11); that is, if the randomization algorithm can find a feasible solution, (7.11) is certainly feasible.

Relaxation: To deduce the relaxed problem, first, we include an objective function to the problem (7.11); therefore, when multiple solutions exist, the one with the minimum objective value is returned. We reformulate the HTE feasibility problem to minimizing the transmission power problem,

$$\begin{aligned} \text{minimize}_{\mathbf{w}} \quad & ||\mathbf{w}||_2^2 \\ \text{subject to} \quad & |\mathbf{w}^H \mathbf{h}_i|^2 \geq R_{th}, \quad i = 1, \cdots, m \\ & |\mathbf{w}^H \mathbf{g}_j|^2 < S_{th}, \quad j = 1, \cdots, n \end{aligned} \tag{7.12}$$

where $||\cdot||_2$ stands for the Euclidean norm of a vector.

Now using the fact that $\mathbf{h}^H \mathbf{w}\mathbf{w}^H \mathbf{h} = \text{trace}(\mathbf{h}^H \mathbf{w}\mathbf{w}^H \mathbf{h})$ where trace$(\cdot)$ represents the trace of a matrix, (7.11) can recast as,

$$\begin{aligned} \text{minimize}_{\mathbf{X}} \quad & \text{trace}(\mathbf{X}) \\ \text{subject to} \quad & \text{trace}(\mathbf{XQ}) \geq R_{th}, \;\; i = 1, \cdots, m \\ & \text{trace}(\mathbf{XQ}) < S_{th}, \;\; j = 1, \cdots, n \\ & \mathbf{X} \succeq 0, \\ & \text{rank}(\mathbf{X}) = 1, \end{aligned} \tag{7.13}$$

where $\mathbf{X} = \mathbf{w}\mathbf{w}^H$, $\mathbf{Q} = \mathbf{h}\mathbf{h}^H$, rank$(\cdot)$ denotes the rank of a matrix, and $\mathbf{X} \succeq 0$ means that $\mathbf{X}$ is a Hermitian positive semidefinite matrix.

Note that since (7.11) is an NP-hard problem, so is (7.13). Therefore, in the following, a heuristic solution is utilized to analyze a relaxed version of (7.13). The relaxation is based on the observation that (7.13) is almost similar to a semidefinite programming problem except for the last constraint; that is, rank$(\mathbf{X}) = 1$, which is non-convex. As a semidefinite problem can be solved in polynomial time, we relax (7.13) by discarding the rank constraint and deduce an SDR problem,

$$\begin{aligned} \text{minimize}_{\mathbf{X}} \quad & \text{trace}(\mathbf{X}) \\ \text{subject to} \quad & \text{trace}(\mathbf{XQ}) \geq R_{th}, \;\; i = 1, \cdots, m \\ & \text{trace}(\mathbf{XQ}) < S_{th}, \;\; j = 1, \cdots, n \\ & \mathbf{X} \succeq 0 \end{aligned} \tag{7.14}$$

The optimal solution $\mathbf{X}_{opt}$ of the SDR problem provides a lower bound for the objective value of (7.12). If the problem does not yield in a solution, (7.11) is infeasible. This is because the feasible region of the actual problem (7.11) is actually a subset of the feasible region of the relaxed problem (7.14). However, the solution $\mathbf{X}_{opt}$ of the SDR problem does not necessarily solve the NP-hard problem. Nonetheless, the rank relaxation of a general QCQP problem results in the Lagrange bi-dual problem, which is the closest convex problem to the original NP-hard problem. Therefore, though $\mathbf{X}_{opt}$ may not be the optimal solution for the HTE problem, it conforms to

other constraints in (7.13), which means that it could be close to the feasible region of the original HTE problem. Based on this observation, we employ a local-search based randomization algorithm to search for a feasible solution to (7.11).

Randomization Algorithm: If the solution $\mathbf{X}_{opt}$ is rank-one, $\mathbf{w}$ can be deduced by finding the principal eigenvector corresponding to only the non-zero eigenvalue. However, as the SDR relaxes the rank-one constraint, $\mathbf{X}_{opt}$ may not be rank-one in reality. Similar to [121], once the SDR problem is solved, a randomized technique can be used to obtain an approximate solution to the original HTE feasibility problem. Numerous randomization techniques have been proposed so far, and we modify the one proposed in [121]. The general idea of this method is to create a set of candidate vectors $\{\tilde{\mathbf{w}}_{can,i}\}_{i=1}^{L}$ (L = number of randomizations) using $\mathbf{X}_{opt}$ and choose the optimal solution from these candidate vectors. In our application, first, to deduce the candidate vectors, the eigencomposition of $\mathbf{X}_{opt}$ is expressed in the form,

$$\mathbf{X}_{opt} = \mathbf{A}\mathbf{V}\mathbf{A}^{H}, \tag{7.15}$$

and the candidate beamforming vector in the form,

$$\tilde{\mathbf{w}}_{can,i} = \mathbf{A}\mathbf{V}^{1/2}\lambda_l, \tag{7.16}$$

is selected as a candidate vector, where $\mathbf{A}$ is a unitary matrix of eigenvectors, $\mathbf{V}$ is a diagonal matrix of eigenvalues, and λ_l is the random vector that consists of uniformly distributed independent random variables on the unit circle in the complex plane. It helps us to ensure that $\tilde{\mathbf{w}}_{can,i}\tilde{\mathbf{w}}_{can,i}^{H} = \mathbf{A}\mathbf{V}^{1/2}\lambda_l\mathbf{A}^{H}(\mathbf{V}^{1/2})^{H}{\lambda_l}^{H} = \text{trace}(\mathbf{V}\lambda_l{\lambda_l}^{H}) = \text{trace}(\mathbf{V}) = \text{trace}(\mathbf{X}_{opt})$. If any constraint in (7.12) is not met by $\tilde{\mathbf{w}}_{can,i}$, a new randomization round begins. If multiple feasible candidates are found, the one with the smallest norm is selected.

Summary: We formulate a numerical method to test the feasibility of the emulation phase. (7.11) belongs to the class of QCQP problems. It contains a special case of

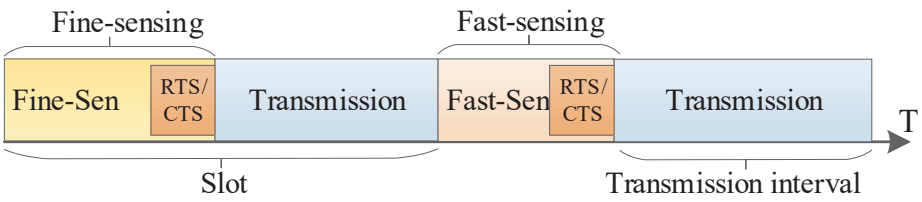


Figure 7.3: The channel access schedule.

the problem considered in [121]; hence, it is NP-hard, and it is not possible to analyze the properties by directly solving it. Therefore, we first formulate the derivation of a relaxed problem which will provide an upper bound for the feasibility answers to the emulation problem; that is, if the relaxed problem is infeasible, (7.11) is definitely infeasible. Afterward, we use a randomization technique which finds a feasible solution through a local search around the point generated by the relaxed problem. This randomization algorithm essentially serves as a lower bound on the HTE problem (7.11); that is, if the randomization algorithm can find a feasible solution, (7.11) is certainly feasible.

7.2.3 Performance Analysis of the Reconnaissance and Emulation Phase

In this section, we simulate the SDR problem and the randomization algorithm described in Section 7.2 to analyze the feasibility of HTE attack under different scenarios.

7.2.3.1 Simulation Setup

In the simulation, this research considers a possible beamforming aiming error ($\gamma_\theta = 1^o$) when the attacker directs its beam towards a certain direction. Hence, $G(\theta)$ is replaced by $G(\theta \pm \gamma_\theta)$. First, we analyze the feasibility of HTE attacks with the fixed location of the victim or exposed nodes and under randomly generated locations of the hidden nodes. We consider a 20×20 2-D space. The path-loss exponent $\alpha = 3.5$, victim's true location is $(0, 0)$ (with two victims $(0, 5)$ and $(0, -5)$), the required transmit power when omnidirectional antenna is used $P_0 = 10$ dBm, receiving antenna sensitivity $R_{th} = -70$ dBm, carrier sensing threshold $S_{th} = -100$ dBm, and the path loss at $d_o = 1m$ is $P_d = 30$ dB.

IoT Node Model: We consider that the victim IoT node has n neighbors in its radio range, and they use omni-directional antennas for communications. Every benign IoT node is equipped with one radio for spectrum sensing and one radio for control information exchange and data transmission.

Channel Access: In shared spectrum operations, each transmission attempt of an IoT node must be preceded by a sensing interval. As shown in Fig. 7.3, IoT nodes employ longer fine-sensing to sense the current channel before initiating a transmission, and they continue to sense the channel—using shorter fast-sensing—during the transmission to negate the collision with co-located IoT nodes. An IoT node is allowed to access a channel when it finds the channel available. After accessing the channel, two IoT nodes exchange RTS/CTS messages to reserve the channel.

Though the scope of this research is to illustrate the PHY-layer constraints and configurations of the attacker (i.e., the reconnaissance and emulation phase), we incorporate MAC-layer information to help readers grasp a more comprehensive overview of the HTE attack.

7.2.3.2 Success Rate of HTE

We use the Monte Carlo simulation to estimate the success rate of HTE attacks—in terms of successfully impersonating as a hidden terminal—with different combinations of the number of antenna elements (N_{ele}) and the number of hidden nodes (n). In each simulation, the location of each node is randomly generated, except the victim node (i.e., $(0, 0)$).

For each combination, totally 1000 trial runs are launched, and the values in Table 7.1 represent the average of these trials. The table contains the number of times where the SDR problem finds a solution (A_{SDR}), the number of times where the randomization algorithm finds a feasible solution (A_{local}), and how tightly these two results are bounded (A_{local}/A_{SDR}). As the SDR solution provides an upper bound and the local-search provides a lower bound on the original HTE feasibility problem,

the number of times that the original problem has feasible solutions lie between A_{SDR} and A_{local}.

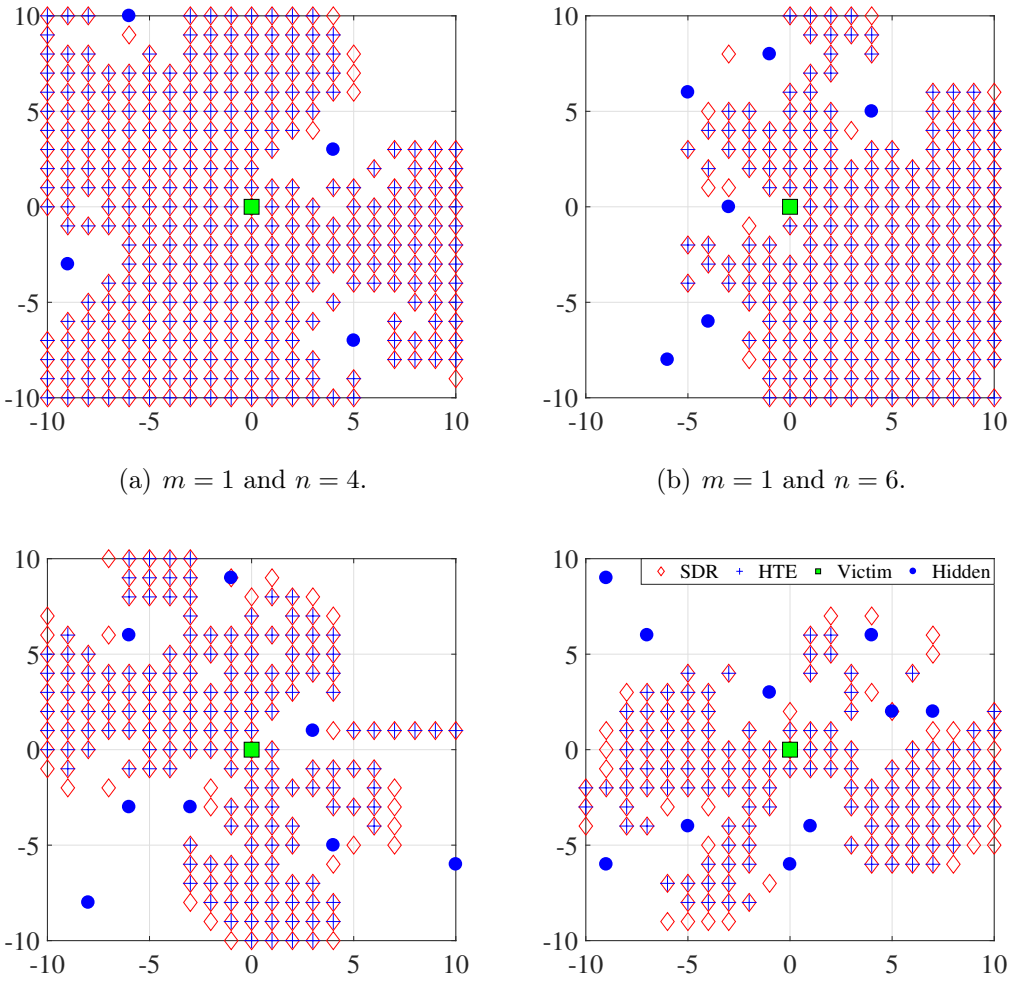
Table 7.1: Successful Cases: Emulation of Hidden Terminal

n	$N_{ele} = 4$	$N_{ele} = 6$	$N_{ele} = 8$	$N_{ele} = 10$
4	84/80 95%	141/137 97%	154/149 97%	310/303 98%
5	71/65 92%	112/107 96%	121/113 93%	289/271 94%
6	62/53 85%	98/92 94%	117/102 87%	258/230 89%
8	2/0	14/12 86%	73/60 82%	217/186 86%
10	0/0	5/3 60%	29/22 77%	91/67 74%

Table 7.1 demonstrates two important trends. First, both A_{SDR} and A_{local} increases as we increase the number of antenna elements (N_{ele}). It happens because a smart array antenna with more antenna elements offers more flexibility in tuning the radiation pattern; hence, it makes an attacker more capable to perpetrate HTE attacks. In addition, mathematically, more antenna elements means that $\mathbf{w}$ is more tunable and hence larger degree of freedom in solving the problem. Second, both A_{SDR} and A_{local} decrease as n increases. Intuitively, we can understand that adding more hidden nodes represents adding more constraints to the original problem; hence, reducing the feasible space. We can also observe the feasibility of HTE attack with a comparatively lower number of antenna elements than the number of hidden nodes. In the simulation, we observe only two cases where HTE is not feasible, i.e., $n = \{8, 10\}$ and $N_{ele} = 4$. It signifies the weakness of dense IoT deployment against the HTE attack.

7.2.3.3 Impact of Exposed and Hidden Node Density

In this part, we investigate how the feasibility of HTE is impacted by the number of exposed or victim nodes (m) and hidden nodes (n), more importantly, how the relative positions and angles of all nodes impact the feasibility problem. In the simulation,



(a) $m = 1$ and $n = 4$.

(b) $m = 1$ and $n = 6$.

(c) $m = 1$ and $n = 8$.

(d) $m = 1$ and $n = 10$.

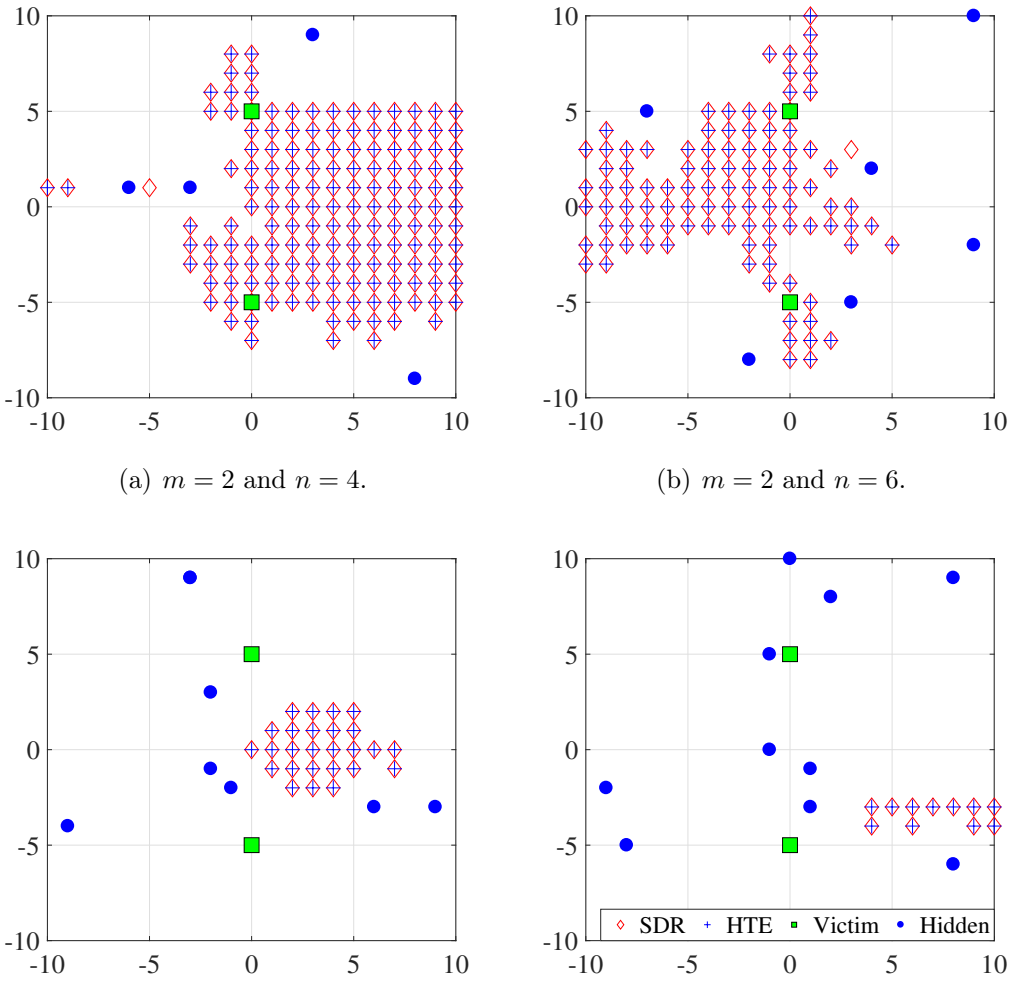
Figure 7.4: The geometric statistics of HTE feasibility problem.

all nodes are fixed, and we vary the true location of the attacker along a square grid in the simulated 2-D space to identify the location where the attacker can find a feasible solution and can launch HTE attacks. A group of simulations are shown in Fig. 7.4. The parameters used in generating the figure are $N_{ele} = 10$, $m = 1$, and $n = \{4, 6, 8, 10\}$. In Fig. 7.4 and 7.5, the locations marked by green filled squares, blue filled circles, red unfilled diamonds, and blue pluses represent the location of the victim(s) (m), hidden nodes (n), SDR feasible points, and HTE feasible points, respectively.

Hidden Node Density: In the figure, most of the locations where SDR is feasible, are also marked by blue pluses; it means that solutions to the HTE feasibility problem are tightly bounded by solutions to the SDR and the randomization algorithm. By comparing the figures in Fig. 7.4, we can observe that as the number of hidden nodes in the attacker's transmission range increases, the number of locations where HTE is feasible decreases. It indicates that the higher density of IoT nodes is less susceptible to HTE attacks. Thereby it provides an important understanding of secure IoT deployments.

Guard Against HTE Attacks: This analysis is insightful to trace the physical location of HTE attackers. If we can determine the presence of the HTE attacker (using a different method), this analysis has the potential to help us narrow down possible hiding locations of the HTE attacker, as HTE can be launched from only certain places. Furthermore, this analysis is also helpful for finding the weaknesses in critical IoT infrastructure in the shared spectrum operation; therefore, it can help design a robust IoT network.

Attack Efficiency vs Risk of Exposure: Intuitively, an attacker must utilize its resources to maximize its attack objective, i.e., attacking more IoT nodes. However, it must also take into account the risk of detection. From Fig. 7.4 and Fig. 7.5, we can observe that, as we increase the number of victim nodes or exposed nodes



(a) $m = 2$ and $n = 4$.

(b) $m = 2$ and $n = 6$.

(c) $m = 2$ and $n = 8$.

(d) $m = 2$ and $n = 10$.

Figure 7.5: Attack efficiency vs risk of detection.

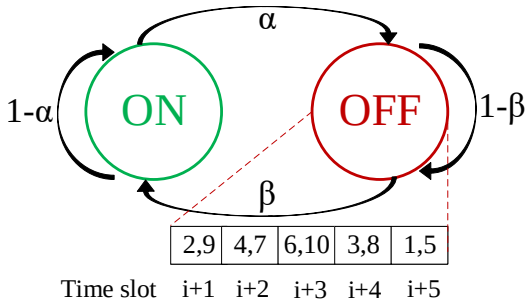


Figure 7.6: Activity of HTE attacker.

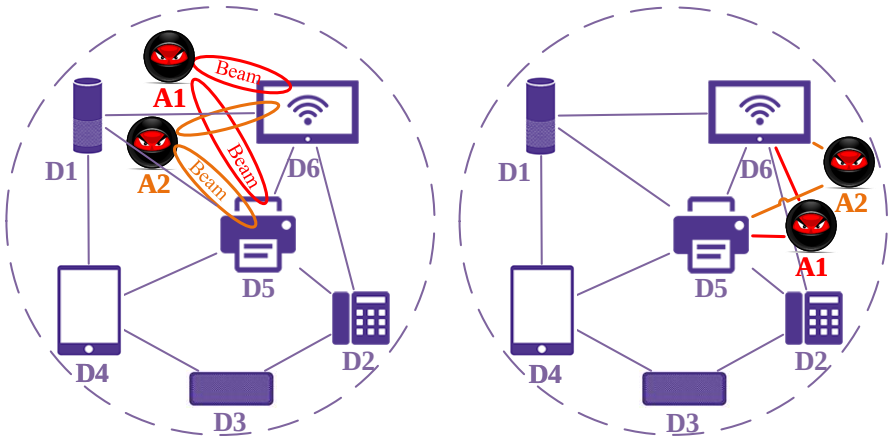
from $m = 1$ to $m = 2$, the feasible space to launch the attack decreases. Thereby, it also increases the risk of exposing the attacker's location. Hence, considering this observation, an attacker must trade-off between the reward of attack and the cost of exposure.

7.3 Proposed Random-HTE Attack: The Interference Phase

An omniscient HTE attacker with unrestricted resources can find the operating channel of the victim instantly and can degrade the SINR well enough to make it infeasible for communication. However, in reality, an attacker has realistic constraints and restricted knowledge of the victim. In this section, we propose a random sweeping strategy for an HTE attacker, where the attacker randomly fluctuates between acting benignly (by generating legitimate communication packets between nodes in its own network) and maliciously. The attacker has limited sensing capability, and the channel-hopping sequence of the victim is unknown to it, but it can hop through different channels at each time slot to detect the operating channel of the victim. The interference phase pans out in two parts: 1) plausible deniability and 2) detect and interfere.

7.3.1 Plausible Deniability

Unlike a conventional jammer, an HTE attacker acts as a legitimate network device that performs regular communications with devices in its own network (e.g., A2 in Fig. 7.7); this, along with the impersonation of a hidden terminal, provides the attacker



(a) The real scenario. (b) The emulated scenario.

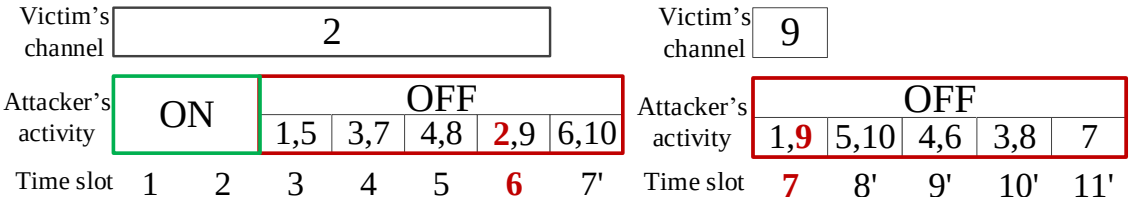
Figure 7.7: Illustration of hidden terminal emulation attack.

an alibi to reactively interfere with its hidden counterparts. This behavior effectively helps the attacker to avoid state-of-the-art jamming detection systems [87]. Thereby, an attacker randomly generates (i.e., OFF to ON state) and terminates packets (i.e., ON to OFF state) at each time slot with a probability β and α, respectively. Here, β and α are attack parameters and they impact the dynamics of the defense problem. In Section 7.3.3, we will illustrate the influence of these parameters on the optimal policy.

7.3.2 Detect and Interfere

In its OFF period, the attacker sweeps through different channels to detect the operating channel of the victim. Assume that the attacker has finished its current packet at i^{th} time slot (Fig. 7.6), thereby it will start the channel sweeping process from $(i+1)^{th}$ time slot. As the attacker plans to execute a DoS attack, the attacker tries to cause successive transmission failures and to force the victim to drop the current packet by reaching the maximum transmission failures.

Attacker's Constraints: We assume that the attacker can only sense n channels ($n < N$) at each slot and sniffs for RTS/CTS messages; it detects the transmission of a particular victim by listening to RTS/CTS messages. After the detection, the attacker interferes with the reception of the victim. However, the attacker has limited



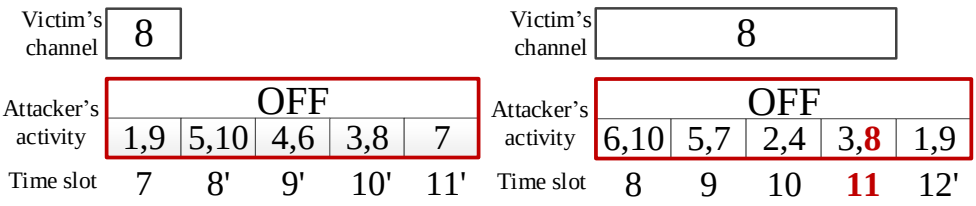
(a) Before an HTE attack. (b) After an HTE attack.

Figure 7.8: Randomization after each successful attack.

interference power that it can use in each channel; if the attacker fails to corrupt the packet (with a probability $1-\nu$) in the first attempt, it will divert all its interference power to the target channel in the next time slot to corrupt it. The attacker will be successful in the second attempt because of the heightened interference power.

Attacker's Strategy: Here, the attacker randomly generates a channel-hopping sequence after each successful attack (i.e., successful transmission failure) and hops through the sequence periodically until it detects the operating channel of the victim. This hopping strategy fosters the attacker to put an upper bound on how long (i.e., the channel residence time) the victim can continuously utilize a channel when the attacker is in the OFF state. Given N channels, if the victim stays on the same channel, it will be detected within $\lceil N/n \rceil$ slots. Therefore, the maximum residence time in a channel is $K = \lceil N/n \rceil - 1$.

Fig. 7.8(a) shows an illustration of the attack sequence with $N = 10$ and $n = 2$, where the attacker initiates malicious actions from slot-3. Here, the victim operates in channel-2; at slot-6, the attacker detects it and perpetrates the attack. After a packet drop, the defender hops to channel-9, and, at the same time, the attacker re-randomizes its attack sequence discarding the earlier attack channel (i.e., channel-2). This strategy helps the attacker to detect the victim faster (due to omission of earlier attack channels) after every attack. In Fig. 7.8(b), we can see that the attacker attacking again at the subsequent slot (i.e., slot-7).



(a) Randomization after an attack. (b) Re-randomization.

Figure 7.9: An unsuccessful attack preceded by a successful one.

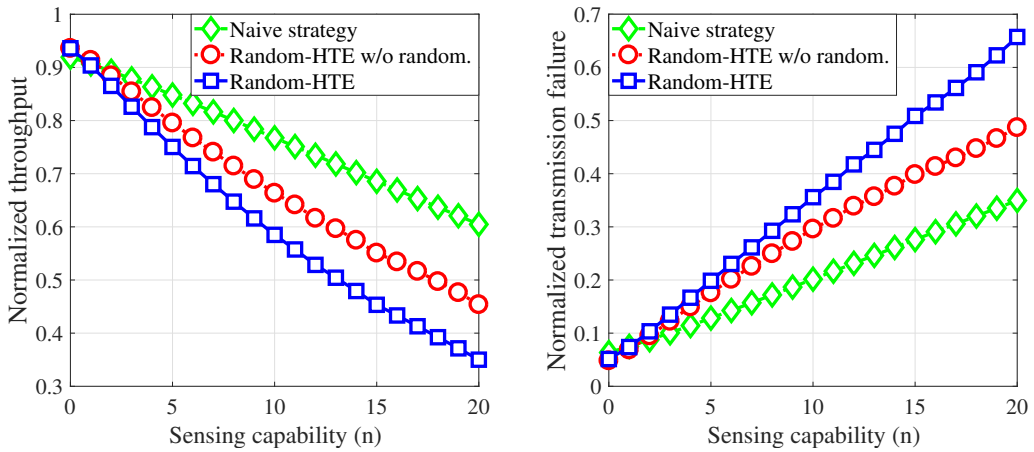
If the attacker cannot detect the victim in the subsequent slot, it will re-randomize its attack sequence, without altering the channels it visited in the current slot; otherwise, the defender can learn the deterministic part of the random attack sequence of the attacker, i.e., omission of earlier attacked channels. Fig. 7.9(a) illustrates an alternative scenario if the victim had chosen channel-8 instead of channel-9 in Fig. 7.8(b), and Fig. 7.9(b) illustrates the re-randomized sequence.

Summary: The proposed attack strategy introduces uncertainties in actions of the attacker; hence, we name it random-HTE attack. Though the attack behavior and the attack sequence are random, random-HTE is strategically designed to detect the victim fast. The attack model unfolds in four steps: 1) alternate between ON and OFF states, 2) hop through the attack sequence until the victim is detected, 3) randomize the sequence after each attack, and 4) re-randomize when an unsuccessful attack attempt preceded by a successful one.

7.3.3 Performance Analysis of the Interference Phase

First, we present the attack performance by the normalized throughput and transmission failure of the victim, then we study the proposed attack in different scenarios.

Random-HTE Attack: Fig. 7.10 demonstrates the performance of the random-HTE strategy in comparison to the naive-random approach, where the attacker randomly selects n channels at each slot (i.e., n/N), and it does not consider the channels

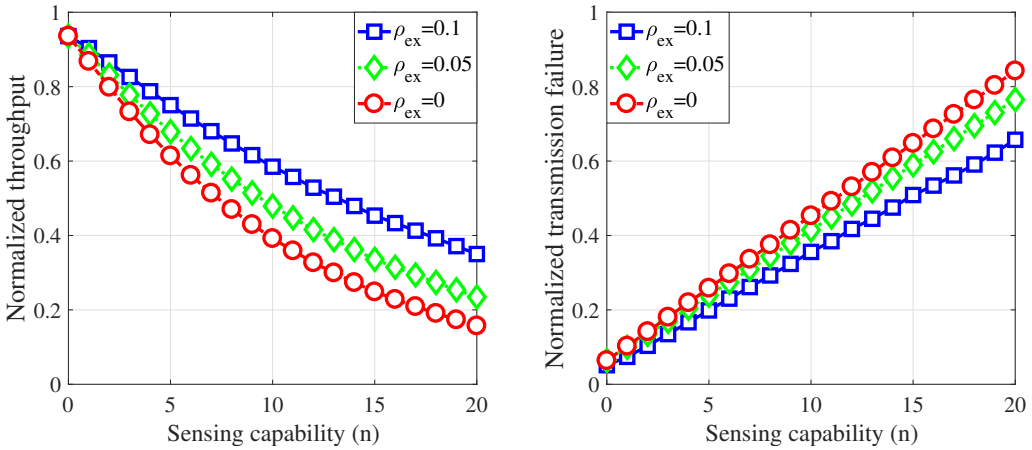


(a) Victim's throughput. (b) Victim's transmission failure.

Figure 7.10: Performance of random-HTE attack.

that have been visited in the past. In addition, we compare it to the random-HTE without re-randomization approach, where the attacker does not re-randomize after each unsuccessful attempt followed by a successful one, and the defender exploits this deterministic trait. In Fig. 7.10(a), the victim experiences the least throughput in random-HTE attack due to the iterative process and re-randomization of random-HTE. Similarly, in Fig. 7.10(b), the victim of random-HTE attack endures most transmission failures.

Effect of ρ_{ex}: The attacker randomly fluctuates between benign and malicious behaviors to reduce the risk of detection. Therefore, it denies opportunities to attack when it is behaving benignly. The benign behavior is represented by ρ_{ex}, which denotes the amount of time the attacker acts benignly, i.e., the attacker (A1) exchanges regular data packets with A2. From Fig. 7.11 we can observe that the attacker's performance degrades with the increase in its benign behavior, i.e., ρ_{ex}.



(a) Victim's throughput. (b) Victim's transmission failure.

Figure 7.11: Performance of random-HTE attack with variable ρ_{ex}.

CHAPTER 8: PROPOSED CONTEXT-AWARE DETECTION AGAINST HIDDEN TERMINAL EMULATION ATTACK: THIRD EYE

This chapter introduces a context-aware Markov-based detection strategy against the HTE attack.

8.1 Proposed Mathematical Modeling of Hidden Terminals

The reception behavior of the defending (or victim) IoT device is considered as an ON-OFF process: $(X(t); t \geq 0)$ with state space $\{0, 1\}$, where 0 and 1 correspond to the idle and the receiving state, respectively. Let A4 denote the IoT device that is evaluating abnormal interference, hereafter referred as the *node under test* (NUT), and the hidden terminal from the external network (i.e., HTE-1) is named as the *external node* (EX).

8.1.1 Proposed Markov Model

In this subsection, we formulate different components necessary to capture the benign behavior of a co-located hidden terminal of an external network, using a five-state Markov process that captures the key aspects of the interaction among PUs, NUT, and EX.

Markov States: We define $X(t)$, $E(t)$, and $Y(t)$ as the state of the NUT, the EX, and the PU in the current channel at time slot t, respectively. Note that $E(t))$ and $Y(t)$ are ON-OFF processes with state space $\{0, 1\}$, where 0 and 1 correspond to the idle and the transmitting state, respectively. The interaction between $X(t)$, $E(t)$, and $Y(t)$ is captured in a five-state discrete-time Markov model, which is represented in Table 8.1.

Table 8.1: State Description of the Proposed Contextual Model

$Z(t)$	$Y(t)$	$X(t)$	$E(t)$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	X	X

The Markov state $Z(t) \equiv \{Y(t), X(t), E(t)\}$ denotes the state of the proposed contextual model in NUT's current operating channel at the end of a time slot. The brief descriptions of the states are:

0: The current channel is free (i.e., PU is idle), the NUT is idle (i.e., not receiving), and the EX is either idle or transmitting on another channel.

1: The current channel is free, the NUT is idle, and the EX is transmitting.

2: The current channel is free, the NUT is receiving, and the EX is either idle or transmitting on another channel.

3: The current channel is free, the NUT is receiving, and the EX is transmitting. *This state represents the collision or interference.*

4: The current channel is busy (i.e., PU is active).

The state transition diagram of the proposed Markov model is shown in Fig. 8.1, which depicts the interaction between the PU, the NUT, and the EX. Transitions between non-neighboring states are presented by dashed arrows.

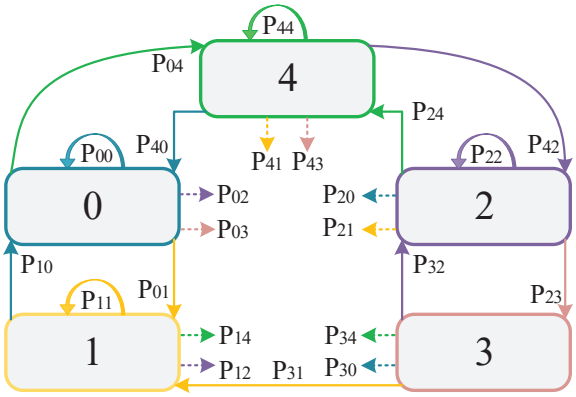


Figure 8.1: The proposed Markov model.

8.1.1.1 Transition Probabilities

We consider that each neighbor of the NUT has a packet arrival rate λ that is destined for the NUT and $\lambda_{in} = (k-1)\lambda$. We capture the effect of hidden terminals by the parameter $k_h \in \{0, \cdots, k-1\}$, which represents the number of internal nodes that are hidden terminals to the EX. In addition, we define the parameter $\alpha \equiv k_h/(k-1)$ as the fraction of internal IoT devices that are hidden to the EX. We assume that each IoT device broadcasts its identity periodically, and IoT devices sniff the wireless medium to discover the presence of other IoT devices—from external networks—within their radio range. In Fig. 7.7, though A1, A3, and A5 cannot listen to the transmission of the node HTE-1 (or EX), A2, A4, and A6 can listen to its transmission. Hence, each device maintains a list of external nodes that are hidden to them (by exchanging information within internal IoT devices), and it helps them to deduce the value of α. Table 8.2 summarizes the notations used in the proposed Markov model.

To derive steady-state probabilities, we first deduce the single-step transition probabilities. We use P_{ij} to denote $\Pr(Z(t+1) = j | Z(t) = i)$, i.e., the probability of transitioning to state j at the next slot from the current state i. We capture the feature of the random channel-hopping process in our model, where an IoT device can start a new transmission only when there is a channel available. In the following

Table 8.2: Notations Used in the Markov Model

Symbol	**Definition**
P_{λ_p}	Pr{a PU packet arrival in a slot}
P_{μ_p}	Pr{a PU packet ending in a slot}
$P_{\lambda_{in}}$	Pr{an internal packet arrival in a slot for the NUT}
$P_{\mu_{in}}$	Pr{an internal packet ending in a slot}
$P_{\lambda_{ex}}$	Pr{an external packet arrival in a slot}
$P_{\mu_{ex}}$	Pr{an external packet ending in a slot}

discussion, we use the terms *states* in the proposed Markov model and the *status* of the NUT in a time slot interchangeably.

The transition from state '**0**' depends on PU activities (P_{λ_p} and Π_b), internal traffic parameter ($P_{\lambda_{in}}$), external traffic parameter ($P_{\lambda_{ex}}$), and collision probability (P^b_{col}). Now, transitions from the idle state (i.e., $Z(t) = 0$):

$$P_{00} = \sum_{b=0}^{M-2} \Pi_b(1 - P_{\lambda_p})(1 - P_{\lambda_{in}})(1 - P^b_{col}) + \Pi_{M-1}(1 - P_{\lambda_p})(1 - P_{\lambda_{in}})(1 - P_{\lambda_{ex}}), \tag{8.1}$$

$$P_{01} = \sum_{b=0}^{M-2} \Pi_b(1 - P_{\lambda_p})(1 - P_{\lambda_{in}})P^b_{col} + \Pi_{M-1}(1 - P_{\lambda_p})(1 - P_{\lambda_{in}})P_{\lambda_{ex}}, \tag{8.2}$$

$$P_{02} = \sum_{b=0}^{M-2} \Pi_b(1 - P_{\lambda_p})P_{\lambda_{in}}(1 - P^b_{col}) + \Pi_{M-1}(1 - P_{\lambda_p})P_{\lambda_{in}}(1 - P_{\lambda_{ex}}), \tag{8.3}$$

$$P_{03} = \sum_{b=0}^{M-2} \Pi_b(1 - P_{\lambda_p})P_{\lambda_{in}}P^b_{col} + \Pi_{M-1}(1 - P_{\lambda_p})P_{\lambda_{in}}P_{\lambda_{ex}}, \tag{8.4}$$

$$P_{04} = \sum_{b=0}^{M-1} \Pi_b P_{\lambda_p}, \tag{8.5}$$

where Π_b = the steady-state probability that exactly b channels are busy by PUs, $P^b_{col} = (1 - \rho_{ex})P_{\lambda_{ex}}P^b_{match}$, ρ_{ex} = the steady-state probability that the EX is active,

and $P^b_{match} = 1/(M-b)$.

Similarly, the transition from state '**1**' depends on PU activities (P_{λ_p} and Π_b), internal traffic parameter ($P_{\lambda_{in}}$), and external traffic parameter ($P_{\mu_{ex}}$). Now, transitions from the EX active state (i.e., $Z(t) = 1$):

$$\begin{aligned} P_{10} = &\sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})(1 - P_{\lambda_{in}}) \\ &+ \Pi_{M-1}(1 - P_{\lambda_p})(1 - P_{\lambda_{in}}) P_{\mu_{ex}}, \end{aligned} \tag{8.6}$$

$$P_{11} = \Pi_{M-1}(1 - P_{\lambda_p})(1 - P_{\mu_{ex}}), \tag{8.7}$$

$$\begin{aligned} P_{12} = &\sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p}) P_{\lambda_{in}} \\ &+ \Pi_{M-1}(1 - P_{\lambda_p}) P_{\lambda_{in}} P_{\mu_{ex}}, \end{aligned} \tag{8.8}$$

$$P_{14} = \sum_{b=0}^{M-1} \Pi_b P_{\lambda_p}. \tag{8.9}$$

Now, the transition from state '**2**' depends on PU activities (P_{λ_p} and Π_b), internal traffic parameter ($P_{\mu_{in}}$), external traffic parameter ($P_{\lambda_{ex}}$), and collision probability (P^b_{col}). However, the collision probability (P^b_{col}) changes in this scenario because the NUT is already transmitting and a collision can only happen from hidden terminals. Therefore, the model must account the hidden terminal factor (α). Now, transitions from the NUT's receiving state (i.e., $Z(t) = 2$):

$$\begin{aligned} P_{20} = &\sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p}) P_{\mu_{in}} (1 - P^b_{col}) \\ &+ \Pi_{M-1}(1 - P_{\lambda_p}) P_{\mu_{in}} (1 - P_{\lambda_{ex}}), \end{aligned} \tag{8.10}$$

$$\begin{aligned} P_{21} = &\sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p}) P_{\mu_{in}} (1 - \rho_{ex}) P_{\lambda_{ex}} P^b_{match} \\ &+ \Pi_{M-1}(1 - P_{\lambda_p}) P_{\mu_{in}} P_{\lambda_{ex}}, \end{aligned} \tag{8.11}$$

$$P_{22} = \sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})(1 - P_{\mu_{in}})(1 - P_{col}^b) + \Pi_{M-1}(1 - P_{\lambda_p})(1 - P_{\mu_{in}})(1 - \alpha P_{\lambda_{ex}}), \tag{8.12}$$

$$P_{23} = \sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})(1 - P_{\mu_{in}}) P_{col}^b + \Pi_{M-1}(1 - P_{\lambda_p})(1 - P_{\mu_{in}})\alpha P_{\lambda_{ex}}, \tag{8.13}$$

$$P_{24} = \sum_{b=0}^{M-1} \Pi_b P_{\lambda_p}, \tag{8.14}$$

where $P_{col}^b = (1 - \rho_{ex})\alpha P_{\lambda_{ex}} P_{match}^b$ and $\alpha = k_h/(k-1)$.

Now, the NUT immediately tries to avoid a collision after detecting it, and the transition from state '**3**' depends on PU activities (P_{λ_p} and Π_b), internal traffic parameter ($P_{\lambda_{in}}$), and external traffic parameter ($P_{\mu_{ex}}$). Hence, transitions from the collision state (i.e., $Z(t) = 3$):

$$P_{30} = \sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})(1 - P_{\lambda_{in}}) + \Pi_{M-1}(1 - P_{\lambda_p}) P_{\mu_{ex}}, \tag{8.15}$$

$$P_{31} = \Pi_{M-1}(1 - P_{\lambda_p})(1 - P_{\mu_{ex}}), \tag{8.16}$$

$$P_{32} = \sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p}) P_{\lambda_{in}}, \tag{8.17}$$

$$P_{34} = \sum_{b=0}^{M-1} \Pi_b P_{\lambda_p}. \tag{8.18}$$

After experiencing the channel busy by a PU, the NUT hops to another available channel. The transition from state '**4**' depends on PU activities (P_{λ_p} and Π_b), internal traffic parameter ($P_{\lambda_{in}}$), external traffic parameter ($P_{\mu_{ex}}$), and collision probability (P_{col}^b). Now, transitions from the channel busy state (i.e., $Z(t) = 4$):

$$P_{40} = \sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})(1 - P_{\lambda_{in}})(1 - P_{col}^b) + \Pi_{M-1}(1 - P_{\lambda_p})(1 - P_{\lambda_{in}})P_{free} + \Pi_M P_{\mu_p}, \tag{8.19}$$

$$P_{41} = \sum_{b=0}^{M-1} \Pi_b (1 - P_{\lambda_p})(1 - P_{\lambda_{in}})P_{col}^b + \Pi_{M-1}(1 - P_{\lambda_p})\rho_{ex}(1 - P_{\mu_{ex}}), \tag{8.20}$$

$$P_{42} = \sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})P_{\lambda_{in}}(1 - P_{col}^b) + \Pi_{M-1}(1 - P_{\lambda_p})P_{\lambda_{in}}P_{free}, \tag{8.21}$$

$$P_{43} = \sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})P_{\lambda_{in}}P_{col}^b + \Pi_{M-1}(1 - P_{\lambda_p})P_{\lambda_{in}}(1 - \rho_{ex})P_{\lambda_{ex}}, \tag{8.22}$$

$$P_{44} = \sum_{b=0}^{M-1} \Pi_b P_{\lambda_p} + \Pi_M (1 - P_{\mu_p}), \tag{8.23}$$

where $P_{free} = \rho_{ex}P_{\mu_{ex}} + (1 - \rho_{ex})(1 - P_{\lambda_{ex}})$ and $P_{col}^b = (1 - \rho_{ex})P_{\lambda_{ex}}P_{match}^b$.

Note that all transition probabilities except the ones from the channel busy state (i.e., $Z(t) = 4$) are conditioned on the fact that at least one channel is available. Therefore, we must transform (1)-(18) and $P_{ij} \leftarrow P_{ij}/(1 - \Pi_M)$, where $i \in \{0, 1, 2, 3\}$ and $j \in \{0, 1, 2, 3, 4\}$.

8.1.1.2 Calculation of $\mathbf{\Pi_b}$

Π_b represents the probability that, at a given time, b PUs are active, where $b \in \{0, \cdots, M\}$. Here, we consider that PU traffic is homogeneous on each channel, the buffer in each PU can store at most one packet at a time, and a packet is kept in the buffer until it is transmitted successfully; hence, the PU traffic follows the $\mathbf{M/M/1/1}$ queuing model.

Let us consider that $\mathbb{A}(t) = b$ represents the number of active PUs at time slot t.

The process $\{\mathbb{A}(t), t = 0, 1, \cdots\}$ forms a Markov chain whose state transition diagram is given in Fig. 8.2. To characterize the behavior of PU channels, we define $\mathbb{F}_f^\gamma$ as the event that f PUs will finish their transmission in the next slot, given that γ PUs are transmitting. In addition, we define $\mathbb{S}_s^\beta$ as the event that s PUs will start new transmissions in the next slot, given that β PUs are idle. Hence, the probabilities of events $\mathbb{F}_f^\gamma$ and $\mathbb{S}_s^\beta$ are:

$$\mathbb{F}_f^\gamma = \binom{\gamma}{f} P_{\mu_p}^f (1 - P_{\mu_p})^{\gamma - f}, \tag{8.24}$$

$$\mathbb{S}_s^\beta = \binom{\beta}{s} P_{\lambda_p}^s (1 - P_{\lambda_p})^{\beta - s}. \tag{8.25}$$

Therefore, the state transition probability from state $\{\mathbb{A}(t) = i\}$ to state $\{\mathbb{A}(t+1) = j\}$ can be written as:

$$P_{i,j} = \begin{cases} \sum_{f=0}^{i} \Pr(\mathbb{F}_f^i) \Pr(\mathbb{S}_{j-i+f}^{M-i+f}), & \text{for } j \geq i \\ \sum_{f=i-j}^{i} \Pr(\mathbb{F}_f^i) \Pr(\mathbb{S}_{j-i+f}^{M-i+f}), & \text{for } j < i. \end{cases} \tag{8.26}$$

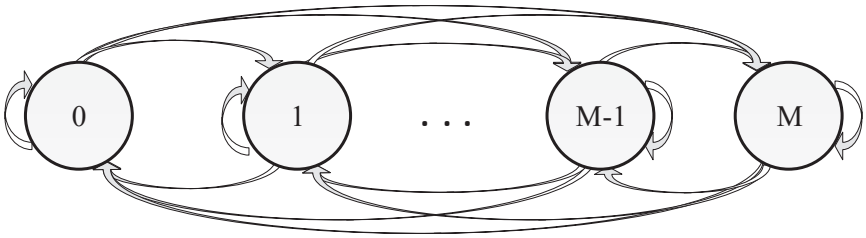


Figure 8.2: The transition diagram of the number of busy channels in a time slot.

Hence, we deduce the steady-state probability of the number of active PUs (or busy channels) in a time slot, denoted as $\mathbf{\Pi} = \left[\Pi_0, \Pi_1, \cdots, \Pi_M\right]$, where Π_b denotes the steady-state probability that b channels are busy in a time slot.

8.1.1.3 Calculation of $\rho_{\mathbf{ex}}$

As mentioned, ρ_{ex} represents the steady-state probability of an external node in the active state. To calculate this, we design a separate Markov model without

the influence of internal nodes. Hence, the model characterizes only the interaction between PUs and the EX. The state transition diagram is given in Fig. 8.3; the state **PU** represents the ON (i.e., 1) or OFF (i.e., 0) state of a PU on the current channel, and the **ON** and the **OFF** states represent the activity of EX on its current operating channel. The corresponding steady-state probabilities are given:

$$\Pi_{off} = \frac{(1-P_{\lambda_p})\{1-(1-P_{\lambda_p})(1-P_{\mu_{ex}})-uP_{\lambda_{ex}}P_{\lambda_p}\}}{(1-P_{\lambda_p})(P_{\lambda_{ex}}+P_{\mu_{ex}})+P_{\lambda_p}}, \tag{8.27}$$

$$\Pi_{on} = \frac{\{1-P_{\lambda_p}(1-u)\}P_{\lambda_{ex}}\Pi_{off}}{1-(1-P_{\lambda_p})(1-P_{\mu_{ex}})-uP_{\lambda_{ex}}P_{\lambda_p}}, \tag{8.28}$$

$$\Pi_{pu} = \frac{P_{\lambda_p}(\Pi_{off}+\Pi_{on})}{1-P_{\lambda_p}}, \tag{8.29}$$

where $\Pi_{off}+\Pi_{on}+\Pi_{pu}=1$, $u=\sum_{b=0}^{M-1}\Pi_b$, and $\rho_{ex}=\Pi_{on}$.

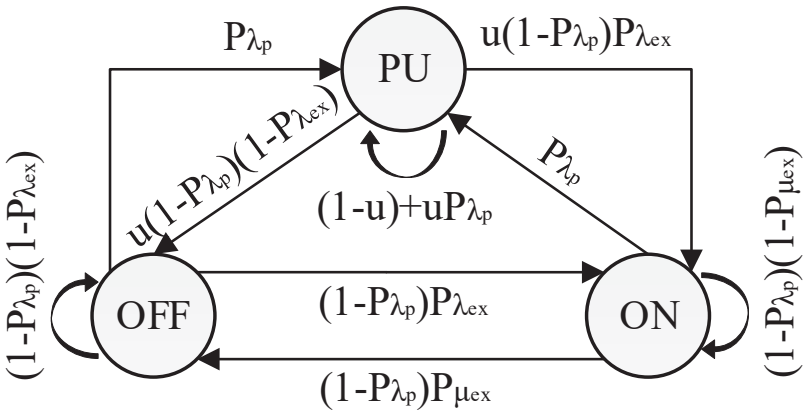


Figure 8.3: The transition diagram of activities of the EX.

8.1.2 Proposed Parameter Estimation of Priority and External Users

In the earlier subsection, we formulated the proposed contextual model using traffic characteristics of all entities in the network. Though the NUT knows its own traffic parameters, traffic parameters of other entities are unknown to it. In this subsection, we propose a Hidden Markov Model (HMM) based parameter estimation technique to extract the required parameters (i.e., $\lambda_{ex}, \mu_{ex}, \lambda_p,$ and μ_p) from the interaction (i.e., statistics from the wide-band sensing) with other entities. In the following, we first present the structure of the HMM, then we give a brief introduction of the forward-

backward procedure in the Baum-Welch (BW) algorithm[113]. Finally, by analyzing the algorithm, we estimate the required parameters.

Hidden Markov Model: A hidden Markov process is a Markov process consisting of two different processes, where $\mathbb{X}$ is the hidden process that is never observable and $\mathbb{Z}$ is the observable process that is perceivable to the agent (i.e., the NUT). X_t and Z_t denote the hidden state and observation state at time t, respectively. Here, the hidden process follows a Markov process with a finite number of states and the observable process is a probabilistic function that generates *symbols* based on the hidden states. The set of symbols comes from a defined *alphabet* $\mathbb{A}$. In our case, $\mathbb{A} = \{0, 1\}$ (i.e., 0 = OFF and 1 = ON).

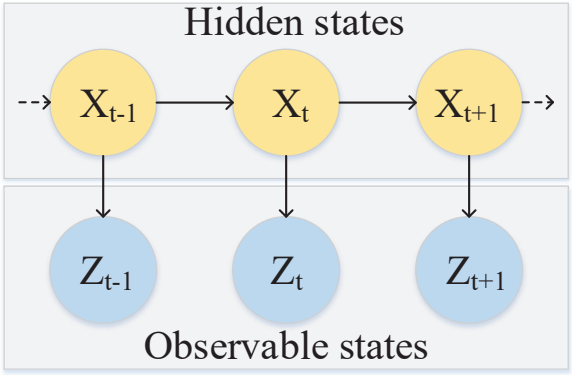


Figure 8.4: The hidden Markov model.

The general concept of an HMM is illustrated in Fig. 8.4. A system of discrete time changes randomly from one state to another, within a finite state space $\mathbb{S}$. In our case, the finite space $\mathbb{S} = \{0, 1\}$. The evolution of the hidden sequence $X_1, X_2, \cdots, X_T$ is unknown, which represents PU or EX states. However, it can be expressed by a sequence of observed symbols from the alphabet $\mathbb{A}$ (i.e., $Z_t \in \mathbb{A}$), which represents the sensing decision on PU or EX activity. However, the sensing result is mixed with measurement errors and differs from the actual states of the PU or EX. To model the HMM, let us define the parameters first:

- Number of hidden states, $s = 2$.
- Number of symbols, $a = 2$.

- Initial state distribution, $\prod^{\mathbf{f}} = \{\pi_i^{\mathbf{f}}\}$, where $i = 0, \cdots, s-1$ and $\mathbf{f}$ ={PU, EX}.
- One-step state transition probabilities, $\mathbb{P}^{\mathbf{f}} = p_{ij}^{\mathbf{f}}$, where $i, j = 0, \cdots, s-1$.
- Symbol emission probability, $\mathbb{B}^{\mathbf{f}} = b_j^{\mathbf{f}}(k)$, where $j = 0, \cdots, s-1$ and $k = 0, \cdots, a-1$.

The one-step state transition probability is:

$$\begin{aligned} &\Pr(X_t^{\mathbf{f}} = j | X_{t-1}^{\mathbf{f}} = i, X_{t-2} = i_{t-2}, \cdots, X_2^{\mathbf{f}} = i_2, X_1^{\mathbf{f}} = i_1) \\ &\quad = Pr^{\mathbf{f}}(X_t^{\mathbf{f}} = j | X_{t-1}^{\mathbf{f}} = i) \\ &\quad = p_{ij}^{\mathbf{f}}, \end{aligned} \tag{8.30}$$

where, $i_1, i_2, \cdots, i_{t-2}, i_{t-1}, i, j \in \{0, 1\}$ and $t > 1$. Therefore, the joint distribution of $X_1^{\mathbf{f}}, X_2^{\mathbf{f}}, \cdots, X_t^{\mathbf{f}}$ is expressed as:

$$\Pr(X_1^{\mathbf{f}} = i_1, X_2^{\mathbf{f}} = i_2, ..., X_t^{\mathbf{f}} = i_t) = \pi_{i_1}^{\mathbf{f}} P_{i_1 i_2}^{\mathbf{f}} \cdots P_{i_{t-1} i_t}^{\mathbf{f}}. \tag{8.31}$$

The emission probability, which represents the probability of observing $Z_t^{\mathbf{f}} = k$ when $X_t^{\mathbf{f}} = j$, i.e., $\mathbb{B}^{\mathbf{f}} = b_j^{\mathbf{f}}(k), j = 0, \cdots, s-1$ and $k = 0, \cdots, a-1$. Therefore,

$$b_j^{\mathbf{f}}(k) = \Pr(Z_t^{\mathbf{f}} = k | X_t^{\mathbf{f}} = j). \tag{8.32}$$

Now, as the sensing process is mixed with measurement errors, the sensing mechanism may experience misdetection and false-alarms. The probability of inferring a PU (or EX) idle while it is actually active is called the probability of misdetection. Similarly, the probability of inferring a PU (or EX) active while it is actually idle is called the probability of false-alarm. These are mathematically expressed as:

$$\begin{aligned}
\Pr(Z_t^{\mathbf{f}} = 0 | X_t^{\mathbf{f}} = 0) &= b_0^{\mathbf{f}}(0), \\
\Pr(Z_t^{\mathbf{f}} = 1 | X_t^{\mathbf{f}} = 0) &= b_0^{\mathbf{f}}(1), \\
\Pr(Z_t^{\mathbf{f}} = 0 | X_t^{\mathbf{f}} = 1) &= b_1^{\mathbf{f}}(0), \\
\Pr(Z_t^{\mathbf{f}} = 1 | X_t^{\mathbf{f}} = 1) &= b_1^{\mathbf{f}}(1).
\end{aligned} \tag{8.33}$$

The BW algorithm proposes an iterative approach to estimate the HMM parameters $\eta^{\mathbf{f}} = \left[\prod^{\mathbf{f}}, \mathbb{P}^{\mathbf{f}}, \mathbb{B}^{\mathbf{f}}\right]$, such that the $\Pr(Z^{\mathbf{f}}|\eta^{\mathbf{f}})$ is maximized. For simplicity, we discard the notation $\mathbf{f}$ from the following calculations. Now, to estimate the parameters, we define the following:

- Forward probability, $\alpha_t(i) = \Pr(Z_1, Z_2, \cdots, Z_t, X_t = S_i|\eta)$, for $i \in \{0, 1\}$
- Backward probability, $\beta_t(i) = Pr(Z_{t+1}, Z_{t+2}, \cdots, Z_{T-1}, Z_T, X_t = S_i|\eta)$, for $i \in \{0, 1\}$
- State transition estimation, $\gamma_t(i, j) = Pr(X_t = i, X_{t+1} = j|\mathbb{Z}, \eta)$, for $i, j \in \{0, 1\}$. It represents the probability of being in state S_i at instant t and in state S_j at instant $t+1$, given the observation sequence $\mathbb{Z}$ and the model parameters $\eta = [\pi, P, B]$
- Estimate of the state at each observation, $\delta_t(i) = Pr(X_t = i|\mathbb{Z}, \eta)$, for $i \in \{0, 1\}$. It represents the probability of being in state S_i at instant t, given the observation sequence $\mathbb{Z}$ and the model parameters $\eta = \left[\prod, \mathbb{P}, \mathbb{B}\right]$

The estimation variables for the HMM parameters are expressed in terms of $\gamma_t(i, j)$ and $\delta_t(i)$:

$$p_{ij} = \frac{\sum_{t=1}^{t=T-1} \gamma_t(i, j)}{\sum_{t=1}^{t=T-1} \delta_t(i)}, \tag{8.34}$$

$$b_j(k) = \frac{\sum_{t=1, Z_t=k}^{t=T} \delta_t(j)}{\sum_{t=1}^{t=T} \delta_t(j)}, \tag{8.35}$$

$$\pi_i = \delta_1(i). \tag{8.36}$$

In (8.34), the numerator represents the expected number of transitions from state i to state j over the interval $T-1$, while the denominator represents the expected number of times a transition happens from state i. The numerator in (8.35) represents the expected number of transitions from state j at which symbol k is observed. In (8.34)-(8.36), $\gamma_t(i,j)$ and $\delta_t(i)$ are calculated as follows:

$$\gamma_t(i,j) = \frac{\alpha_t(i)p_{ij}b_j(Z_{t+1})\beta_{t+1}(j)}{Pr(Z|\eta)}. \tag{8.37}$$

$$\delta_t(i) = \sum_{all\ S_j \in \{0,1\}} \gamma_t(i,j). \tag{8.38}$$

The forward and backward probabilities in the above equations are calculated recursively as follows:

Initialization:

$$\alpha_1(i) = \pi_i b_i(1), \;\; 0 \leq i \leq s-1. \tag{8.39}$$

$$\beta_t(i) = 1, \;\; 0 \leq i \leq s-1. \tag{8.40}$$

Recursion:

$$\alpha_{t+1}(j) = \left[\sum_{i=0}^{s-1} \alpha_t(i)p_{ij}\right] \; b_j(Z_{t+1}). \tag{8.41}$$

$$\beta_t(i) = \sum_{j=0}^{s-1} p_{ij}b_j(Z_{t+1})\beta_{t+1}(j). \tag{8.42}$$

The recursion process terminates when $Pr(\mathbb{Z}|\eta)$ maximizes, which is the probability of observing the sequence $\mathbb{Z}$ given the parameter $\eta = \left[\prod, \mathbb{P}, \mathbb{B}\right]$:

$$Pr(\mathbb{Z}|\eta) = \sum_{i=0}^{s-1} \prod_{t=1}^{T} \alpha_t(i). \tag{8.43}$$

Extraction of Traffic Parameters: Here, we extract traffic parameters of the PU and EX from the estimated HMM parameters $\eta^{\mathbf{f}} = \left[\prod^{\mathbf{f}}, \mathbb{P}^{\mathbf{f}}, \mathbb{B}^{\mathbf{f}}\right]$, such that the $\Pr(Z^{\mathbf{f}}|\eta^{\mathbf{f}})$ is maximized. To do this, let us recall the parameters, $\theta_{\mathbf{f}} = [\lambda_{\mathbf{f}}, \mu_{\mathbf{f}}]$, where λ means the traffic arrival rate, μ means the packet service rate, and $\mathbf{f}$ ={PU, EX}. From the network model, the length of the ON and OFF state are exponentially distributed. In [114], a useful method to compute the state transition rate matrix from the state transition probability matrix is provided. We denote the transition rate matrix as $Q_{\mathbf{f}}$ and

$$Q_{\mathbf{f}} = \begin{pmatrix} -\lambda_{\mathbf{f}} & \lambda_{\mathbf{f}} \\ \mu_{\mathbf{f}} & -\mu_{\mathbf{f}} \end{pmatrix}. \tag{8.44}$$

As described in η, $\mathbb{P}$ is the one-step state transition probability matrix. We know that $\mathbb{P} = \exp(Q\Delta)$ and $Q = \log(\mathbb{P})/\Delta$, where Δ is the sensing period. However, the computational procedure is cumbersome and $\log(\cdot)$ has a limitation when $\mathbb{P}$ has a non-positive eigenvalue. Therefore, we adopt the mapping approach introduced in [114], which provides an easier computational approach and provides a sufficient degree of accuracy. If the two-dimensional transition rate matrix is the form shown in (8.44), then the transition probability matrix is:

$$\mathbb{P} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} = \begin{pmatrix} \exp^{-\lambda\Delta} & 1 - \exp^{-\lambda\Delta} \\ 1 - \exp^{-\mu\Delta} & \exp^{-\mu\Delta} \end{pmatrix}. \tag{8.45}$$

In (8.45), we can calculate Q from $\mathbb{P}$ inversely. In other words, the relation between $\mathbb{P}$ and Q unfolds the relationship between η and θ.

8.1.3 Summary

In this section, we explained the mathematical structure to formulate the building blocks of the proposed context-aware detection strategy. The calculations in this section helps to identify the accepted behavior of a benign hidden terminal of an external network. Though many parameters to deduce the context-aware model are unknown, we proposed an HMM-based estimation strategy to estimate the required parameters. Again, we utilized only the in-hand sensing statistics to compute the estimation without any hardware and networking overhead. The required probabilities can be expressed in terms of the estimated parameters (i.e., $\widehat{\lambda_{\mathbf{f}}}$ and $\widehat{\mu_{\mathbf{f}}}$) as follows:

$$P_{\lambda_{\mathbf{f}}} = 1 - \exp^{-\widehat{\lambda_{\mathbf{f}}} * \mathbf{t}}, \tag{8.46}$$

$$P_{\mu_{\mathbf{f}}} = 1 - \exp^{-\widehat{\mu_{\mathbf{f}}} * \mathbf{t}}. \tag{8.47}$$

Here, $\mathbf{t}$ represents the length of a time slot. Next, we will discuss the strengths and weaknesses of different attack strategies against the proposed context-aware detection model.

8.2 Proposed Reactive Interference Models

As discussed earlier in this research, a strong detection strategy requires a strong attack model. In this section, we discuss different attack models and their efficacy against the proposed context-aware detection strategy.

8.2.1 Attack Models

Though an aggressive attack strategy that constantly interferes with the reception of the victim results in better attack performance, it deviates significantly from benign behaviors and to an reactive attacker. A context-aware detection strategy, which regularly monitors external nodes, can identify this malicious interference; hence, an attacker must haggle between the attack objective and the risk of exposure. In the following, we discuss three different attack strategies.

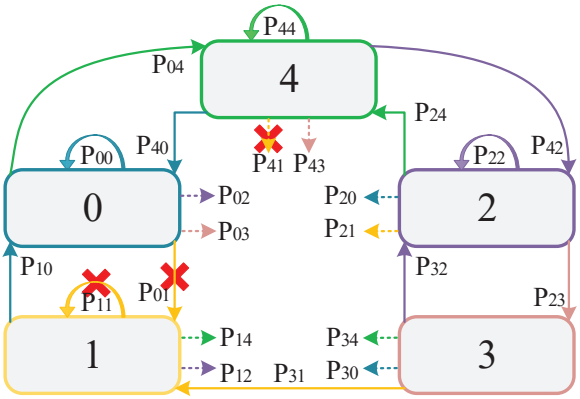


Figure 8.5: Markov chain between a naive attacker and the NUT.

8.2.1.1 Naive Reactive Attacker

The interaction between the NUT and the EX is modeled as the Markov chain illustrated in Fig. 8.5, when the EX is a naive reactive attacker. The behavioral difference between a benign hidden terminal and a naive reactive attacker is that a benign hidden terminal transmits irrespective of the transmission from its hidden counterparts, whereas a naive reactive attacker transmits *only* when it senses transmissions from its hidden counterparts on the wireless channel (i.e., P_{23}=1). Thereby, the transition rates of the corresponding discrete time Markov chain (DTMC) from states 0, 1, and 4 to state 1 is zero. Now, if we observe Fig. 8.1 and Fig. 8.5, the state transition structures are distinct. It means that the Neyman-Pearson test of differentiating these two Markov chains is degenerate, i.e., it becomes a singular detection problem [122], meaning that the test results in an arbitrarily small error [123].

8.2.1.2 Naive Random Attacker

The only difference between a naive random attacker and a naive reactive attacker is that the naive random attacker does not interfere with each reception of the victim, i.e., $P_{23} \neq 1$. Instead, it randomly chooses its attack window to interfere. Nonetheless, both of these attack models follow the similar state transition structures (i.e., P_{01}=P_{11}=P_{41}=0) and yield a singular detection problem. Therefore, though this attack strategy introduces randomness in its behavior, it still remains ineffective against

the context-aware detection model.

8.2.1.3 Intelligent HTE Attacker

To avoid the singular detection problem, we propose a more advanced random reactive attacker, called the intelligent HTE attacker, that better disguises its malicious behavior by mimicking characteristics of benign hidden terminals. In this attack model, the HTE attacker generates regular data packets and communicates with its neighbor (i.e., the passive attacker, HTE-2) regardless of the state of the PU and the victim, meaning that P_{01}=P_{11}=$P_{41} \neq 0$. This attack model increses the detection difficulty since the incorporation of random behaviors makes the HTE attacker similar to a benign hidden terminal. Therefore, an attacker acts benignly by performing regular communications with its neighbor, and—if in its idle period (when the attacker is not transmitting to its neighbor) it finds the victim is receiving—it interferes with the reception of the victim (i.e., interference rate=1). We can make the strategy more random by changing the interference rate; nonetheless, without loss of generality, in this research, we assume interference rate=1. Hence, unlike the benign model, the transition probability from state 2 is:

$$P_{20} = \sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p}) P_{\mu_{in}} \rho_{ex}, \tag{8.48}$$

$$\begin{aligned} P_{21} = &\sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p}) P_{\mu_{in}} (1 - \rho_{ex}) \\ &+ \Pi_{M-1} (1 - P_{\lambda_p}) P_{\mu_{in}}, \end{aligned} \tag{8.49}$$

$$\begin{aligned} P_{22} = &\sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})(1 - P_{\mu_{in}})(1 - P_{col}^b) \\ &+ \Pi_{M-1} (1 - P_{\lambda_p})(1 - P_{\mu_{in}})(1 - \alpha), \end{aligned} \tag{8.50}$$

$$\begin{aligned} \mathrm{P}_{23} = &\sum_{b=0}^{M-2} \Pi_b (1 - P_{\lambda_p})(1 - P_{\mu_{in}}) P_{col}^b \\ &+ \Pi_{M-1} (1 - P_{\lambda_p})(1 - P_{\mu_{in}}) \alpha, \end{aligned} \tag{8.51}$$

$$P_{24} = \sum_{b=0}^{M-1} \Pi_b P_{\lambda_p}, \tag{8.52}$$

where $P_{col}^{b} = (1 - \rho_{ex})\alpha$ and $\alpha = k_h/(k-1)$.

8.2.2 Summary

Three different attack traits, including naive, naive-random, and intelligent, are discussed. Though a naive behavior yields a better attack performance, it increases the risk of exposure because of its distinct state transition structures. In contrast, the proposed intelligent HTE attack model that closely imitates a benign hidden terminal offers a different attack detection challenge. In Section 8.4, we will illustrate the detection performance of our proposed context-aware detection strategy against these attack models. Next, we formulate the detection challenge as a binary hypothesis test to differentiate an observed behavior between benign and malicious.

8.3 Proposed Detection of the Hidden Terminal Emulation Attack

The proposed detection approach is comprised of two steps: *i*) designing a contextual model to characterize the behavior of benign hidden terminals and *ii*) formulating the detection problem as a binary hypothesis testing problem to identify whether a sequence of observed behaviors is likely to be produced from the established benign model or attack model. In Section 8.1, we comprehensively illustrated the first step, and now, we shed light on the second one.

8.3.1 Binary Hypothesis Test

The NUT monitors activities on all channels and collects transmission patterns of all wireless nodes in its surroundings over a time window of $\mathbf{d} = \mathbf{w}/\mathbf{t}$ equal-length slots, where $\mathbf{w}$ is the observation time length and $\mathbf{t}$ is the length of a time slot. To test whether or not the NUT is experiencing HTE attacks, we collect the sequence of observations of the NUT's status $\mathbf{z_d} \equiv \{Z(t)\}_{t=1}^{\mathbf{d}+1}$, called a *sample path* of the discrete time Markov chain that is generated by the influence of either a benign hidden terminal or by an HTE attacker. Now, let us denote transition probability

matrices that characterizes a benign hidden terminal as $\mathbf{P^0}$, that characterizes an HTE attacker as $\mathbf{P}^A$, and that is generated from the observations as $\mathbf{P}$. Thus, a binary hypothesis testing problem can be formed:

$$\mathrm{H}_0 : \mathbf{P} = \mathbf{P^0} \;\; \mathrm{H}_A : \mathbf{P} = \mathbf{P}^{\mathrm{A}}. \tag{8.53}$$

Though most binary hypothesis testing problems require supervised learning, the proposed detection model does not require supervised training as we have formulated closed-form expressions to characterize benign and malicious behaviors. It is reasonable to assume that the initial-state probability distribution is similar to the steady-state probabilities of the states. However, as indicated in [82], the initial distribution has an effect on the detection threshold, which decreases to 0 in $\mathbf{d}$ as $1/\mathbf{d}$. Hence, it is insignificant when $\mathbf{d}$ is large.

Let us define the number of transitions from state i to state j of $\mathbf{z_d}$ as $N_{ij} = \sum_{t=1}^{\mathbf{d}} \mathbf{1}_{\{z_t=i, z_{t+1}=j\}}$, where z_t denotes the t-th element of the sequence $\mathbf{z_d}$ and $i, j \in \{0, 1, 2, 3, 4\}$. Now, the counts $N_i \equiv \sum_{j=0}^{4} N_{i,j} = \sum_{t=1}^{d} \mathbf{1}_{\{z_t=i\}}$. The log-likelihood of $\mathbf{z_d}$ under hypothesis $\mathrm{H_b}$ is (where $\mathrm{b} \in \{0, A\}$):

$$\begin{aligned} \log \Pr(\mathbf{z_d}|\mathrm{H_b}) &= \log \Pi^{\mathrm{b}}_{z_1} \prod_{t=1}^{\mathbf{d}} \mathrm{P}^{\mathrm{b}}_{z_t, z_{t+1}} \\ &= \log \Pi^{\mathrm{b}}_{z_1} + \sum_{i=0}^{4} \sum_{j=0}^{4} N_{ij} \log \mathrm{P}^{\mathrm{b}}_{i,j}. \end{aligned} \tag{8.54}$$

Therefore, the log-likelihood ratio between H_A and H_0 is:

$$\log \frac{\Pr(\mathbf{z_d}|\mathrm{H_A})}{\Pr(\mathbf{z_d}|\mathrm{H}_0)} = \log \frac{\Pi^{\mathrm{A}}_{z_1}}{\Pi^{0}_{z_1}} + \sum_{i=0}^{4} \sum_{j=0}^{4} N_{ij} \log \frac{\mathrm{P}^{\mathrm{A}}_{i,j}}{\mathrm{P}^{0}_{i,j}}. \tag{8.55}$$

The log-likelihood ratio test with threshold τ:

$$\log \frac{\Pi^{\mathrm{A}}_{z_1}}{\Pi^{0}_{z_1}} + \sum_{i=0}^{4} \sum_{j=0}^{4} N_{ij} \log \frac{\mathrm{P}^{\mathrm{A}}_{i,j}}{\mathrm{P}^{0}_{i,j}} \underset{\mathrm{H}_0}{\overset{\mathrm{H}_A}{\gtrless}} \tau. \tag{8.56}$$

We can further fine-tune the threshold by dynamically adjusting it to compensate for the observation window size $\mathbf{d}$:

$$\begin{aligned} &\sum_{i=0}^{4}\sum_{j=0}^{4} N_{ij} \log \frac{\mathrm{P}_{i,j}^{\mathrm{A}}}{\mathrm{P}_{i,j}^{0}} \underset{\mathrm{H}_0}{\overset{\mathrm{H}_A}{\gtrless}} \tau(\mathbf{d}) - \log \frac{\Pi_{z_1}^{\mathrm{A}}}{\Pi_{z_1}^{0}}, \\ &\sum_{i=0}^{4}\sum_{j=0}^{4} \frac{N_{ij}}{\mathbf{d}} \log \frac{\mathrm{P}_{i,j}^{\mathrm{A}}}{\mathrm{P}_{i,j}^{0}} \underset{\mathrm{H}_0}{\overset{\mathrm{H}_A}{\gtrless}} \tau' \equiv \frac{\tau(\mathbf{d}) - \log \frac{\Pi_{z_1}^{\mathrm{A}}}{\Pi_{z_1}^{0}}}{\mathbf{d}}. \end{aligned} \tag{8.57}$$

Here, $\tau(\mathbf{d})$ varies with the observation window size $\mathbf{d}$ to balance the trade-off between the false alarm rate and mis-detection rate. The educated approach is $\tau(\mathbf{d}) = \tau_0\mathbf{d}$ for which $\tau' \approx \tau_0$ as $\mathbf{d}$ increases. The test statistics of the log-likelihood ratio test is:

$$\mathbf{Z} \equiv \sum_{i=0}^{4}\sum_{j=0}^{4} \frac{N_{ij}}{\mathbf{d}} \log \frac{\mathrm{P}_{i,j}^{\mathrm{A}}}{\mathrm{P}_{i,j}^{0}}, \tag{8.58}$$

where $\mathbf{d}, \mathrm{P}_{i,j}^{0}$, and $\mathrm{P}_{i,j}^{\mathrm{A}}$ are constants, and to derive the distribution of $\mathbf{Z}$ under $\mathrm{H_b}$, we must know the distribution of N_{ij}. According to [124], N_{ij} are asymptotically Gaussian distributed; hence, as a linear combination of N_{ij}, the test statistic $\mathbf{Z}$ is also asymptotically Gaussian.

8.3.2 Summary

The detection model captures the interference pattern an IoT device experiences under the influence of a hidden terminal and flags the HTE attack when observations deviate towards the established attack model. The proposed Markov model accumulates all required information into five states, and the binary hypothesis test verifies how well the observed sequence fits with the established benign or malicious behavior model. Our proposed detection technique requires only the carrier sensing information, which is readily available for channel access purposes.

8.4 Performance Analysis of the Third Eye

In this section, we present numerical and simulation results to evaluate the performance of our proposed research. Our research employs a five-state Markov model that is a tractable model, and it can capture the key characteristics of the network transmission patterns. Here, we consider that all CR-enabled IoT devices physically reside within the proximity of each other and share the same ACL at a given time. The simulation parameters are listed in Table 8.3.

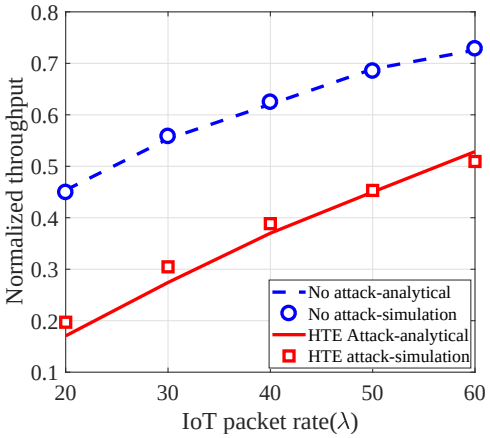
Table 8.3: Simulation Parameters: Third-eye

Parameter	**Value**
Simulation time	100 seconds
SU sensing range	50
The number of channels (or PUs)	10
PU traffic rate (in pkts/sec)	$\lambda_p = 50$; $\mu_p = 100$
Bandwidth	2 Mbps
The size of (RTS+CTS)	160 + 112 bits (802.11b/g)
Fast and fine sensing duration	1 ms (802.22) and 2 ms
IoT traffic rate (in pkts/sec)	$\lambda = 20, 30, 40, 50, 60$; $\mu = 100$
SU packet size	1024 bytes
Hidden terminal factor, α	5/7

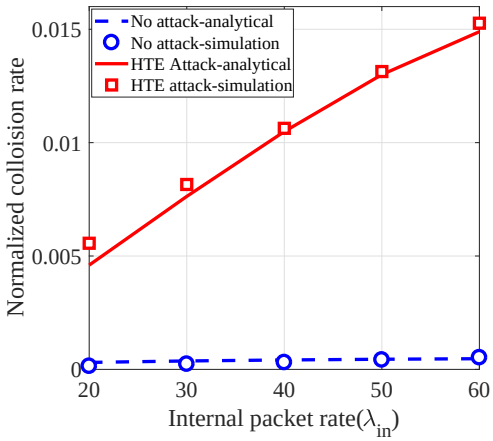
During the simulation, we assume that the NUT is able to capture the transmission pattern of all adjacent IoT devices, and it knows the number of IoT devices in its vicinity (via wireless sniffing). The objective of the NUT (which is receiving) is to determine if the observed interference maintains the pattern set by the mathematical model. During the HMM training phase, IoT devices may estimate the traffic parameters in a long enough training time, so that the estimated values are close to the true values. In contrast, the attacker tries to maintain a stable data packet rate to avoid suspicious behaviors and attacks in its inactive intervals.

8.4.1 Hidden Terminal Emulation Attack

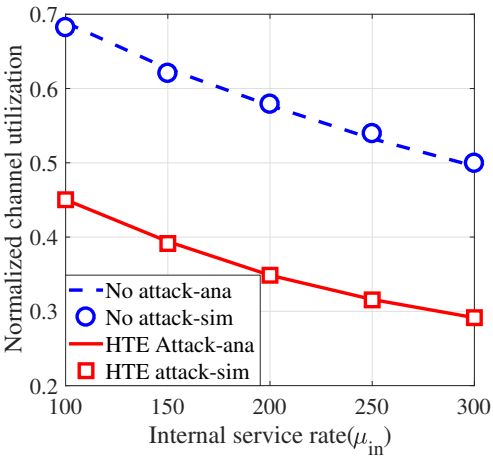
This subsection shows the impact of the proposed HTE attack on the network performance of the NUT.



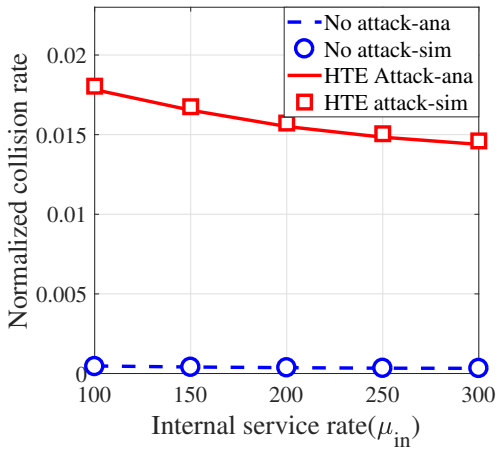
(a) Impact of λ_{in} on throughput.



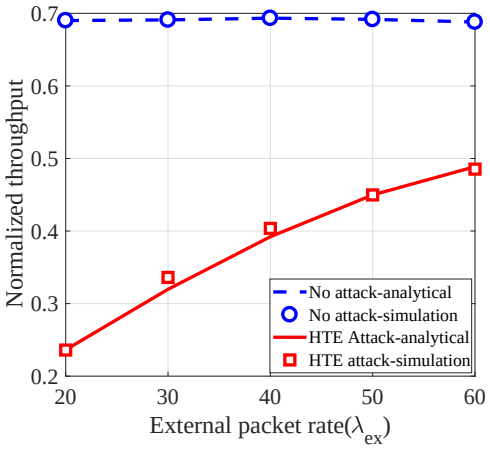
(b) Impact of λ_{in} on collisions.



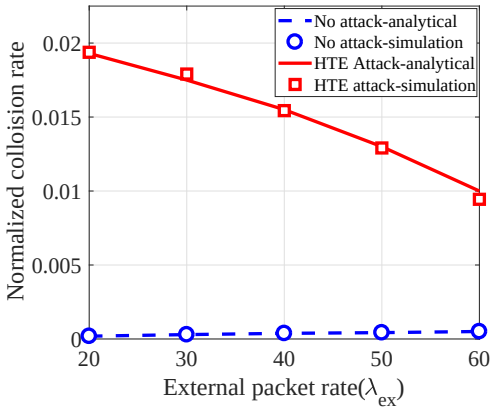
(c) Impact of μ_{in} on channel utilization.



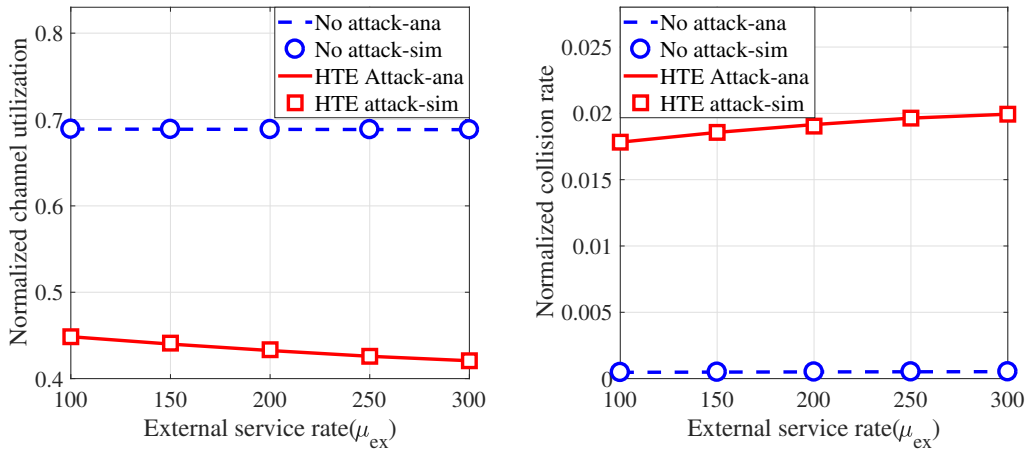
(d) Impact of μ_{in} on collisions.



(e) Impact of λ_{ex} on throughput.



(f) Impact of λ_{ex} on collisions.



(g) Impact of μ_{ex} on channel utilization.

(h) Impact of μ_{ex} on collisions.

Figure 8.6: The impact of different traffic parameters (λ_{in}, μ_{in}, λ_{ex}, and μ_{ex}) on NUT's throughput, channel utilization, and collision.

8.4.1.1 Impact of λ_{in} on the HTE Attack

A higher rate of incoming traffic (i.e., $\lambda_{in} = (k-1)\lambda$) to the NUT increases the opportunity for the attacker to interfere with the NUT's reception. As the attacker tries to interfere each time it is inactive and the victim is receiving, in Fig. 8.6(a), we can observe that the effect of the attack increases with the increase of the incoming traffic rate. However, the effect is not clearly perceivable from this figure because the mean time (or the steady-state probability) in collision state (i.e., state '**3**' in Table 8.1) is insignificant as compared to the normalized throughput.

Fig. 8.6(b) helps to grasp a better picture where the collision rate experienced by the NUT increases rapidly with the increase of the internal traffic rate. As we consider that the NUT can perceive collisions and discard packets instantly, it minimizes the total amount of time the NUT stays at the collision state. Nonetheless, these incidents engender in packet drops, stifle the throughput, and increase the collision rate.

8.4.1.2 Impact of μ_{in} on the HTE Attack

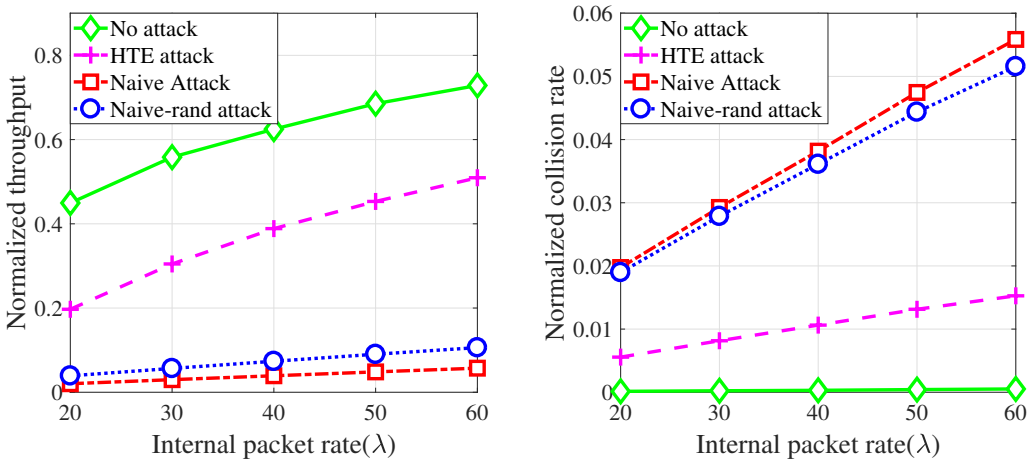
A higher service rate represents faster throughput and shorter packet length for a given data. Therefore, we use a different performance indicator than normalized throughput to illustrate the impact of μ_{in}, i.e., normalized channel utilization. Channel utilization represents the portion of time the NUT utilized the network successfully for communication purposes. Intuitively, we can understand that as we increase the service rate of each packet, the channel utilization decreases. Fig. 8.6(c) provides the corresponding impact of internal packet service rate on the channel utilization. Likewise, the collision rate decreases because attackers have less time to perpetrate the attack. Fig. 8.6(d) shows the change in collision rate with the increase of packet service rate.

8.4.1.3 Impact of λ_{ex} on the HTE Attack

Note that the attacker can only interfere if it is inactive during the transmission of its hidden counterparts; otherwise, it must continue and finish its own data packet transmission. As the traffic rate of the EX rises, the time it stays in the active state also increases (i.e., ρ_{ex}). Hence, the room for interference decreases. Therefore, to augment the impact of the attack, the attacker must decrease its packet arrival rate. In Fig. 8.6(e)-(f), we can observe that under no attack (i.e., when the EX is benign), the EX's traffic has an insignificant effect on the throughput and the collision of the NUT. However, under attack, it illustrates sensitivity to the change in λ_{ex}. Besides attack performance, λ_{ex} also influence the detection accuracy. Later, we will discuss the effect of λ_{ex} on the detection performance.

8.4.1.4 Impact of μ_{ex} on the HTE Attack

Similar to μ_{in}, we consider normalized channel utilization as a performance metric instead of normalized throughput. As we increase the service rate of the attacker (i.e., μ_{ex}), it shortens the amount of time the attacker remains busy with benign



(a) Impact on throughput.

(b) Impact on collisions.

Figure 8.7: Different attack performance.

actions and provides the attacker with more opportunities to perpetrate the attack. As a result, the normalized channel utilization of the NUT decreases (Fig. 8.6(g)). Similarly, the normalized collision rate increases with the increase in μ_{ex} (Fig. 8.6(h)).

8.4.1.5 Different Attack Models

As discussed in Section 8.2, different attack models have their own advantages and disadvantages. In Fig. 8.7(a)-(b), the normalized throughput and collision rate of the NUT are shown for the naive, naive-random, and proposed HTE attack. It is evident that the naive and naive-random attack results in superior attack performance than the proposed HTE attack. Nonetheless, they suffer from singular detection problem and have a negligible immunity against the proposed context-aware detection technique, even with a small observation time.

8.4.2 PU and EX Parameter Estimation

The performance of the proposed *Third Eye* depends on how accurately HMM-based estimators can estimate the required traffic parameters of PUs and EX in victim's sensing range. In addition, the length of a training sample is instrumental to the learning performance. In Fig. 8.8, we can observe the trend of estimation error

for PU packet arrival rate (λ_p) and service rate (μ_p) (showed for 5 PUs). Estimation errors reduce to below 4% when the estimator is trained to 50 seconds.

In this research, we train the HMM estimator with 25 seconds of data and observe the impact of the attack detection for the next 75 seconds without changing the PU or EX activity rate. Nevertheless, in reality, the PU and EX activity rate is not going to be constant all the time and the HMM estimator must re-estimate to track changes.

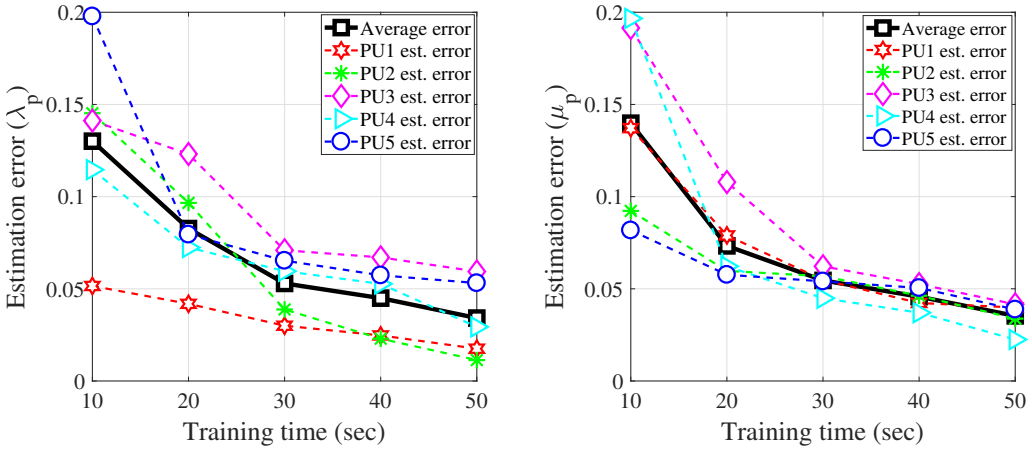
8.4.3 Attack Detection

The proposed mathematical model can effectively distinguish the activity of an attacker through carrier sensing and detect the interference created by HTE attackers. This subsection analyzes the performance of the detection model.

8.4.3.1 ROC Curve

To illustrate the effectiveness of our proposed detection strategy, we compare it with the jamming detection approach that considers the RSS and BER as the primary metrics of jamming detection [42]; here, we name it as the naive method. In this approach, the intuition is that when there is a bit error whereas the RSS value is high, this indicates jamming attack. In addition, we compare the performance to an earlier work [87], which detects anomalies in hidden nodes' behavior. We point out that, to the best of our knowledge, there is not yet a signature-based detection method for the proposed HTE attack to compare with. Our effort is to compare the ability of attack activity detection, with the naive method [42], the anomaly-based method [87], and the proposed *Third Eye*.

Fig. 8.9(a) illustrates the receiver operating characteristic (ROC) curve that represents the efficiency of detection by plotting the true positive rate (i.e., the probability of detection) versus the false positive rate (i.e., the probability of false alarm). Comparing these four ROC curves, we find that the proposed context-aware detection strategy results in a large area under the curve (AUC). Thus, it achieves significantly



(a) Estimation error of λ_p. (b) Estimation error of μ_p.

Figure 8.8: HMM estimation performance.

more reliable detection results. In the case of false negatives, the attacker conducts a very low level of interference, which the detector identifies as statistically insignificant to match with the behavior of an attacker.

Though the anomaly-based detection technique provides almost similar results—if not better—it fails to uniquely identify an HTE attacker because it does not consider exclusive characteristics of an HTE attacker in its detection approach; it only performs well when the goal is to detect anomalous behavior. Conversely, the naive method has a much smaller AUC and suffers extensively from poor false positive rate. As the naive approach does not consider that an interference source could be benign, it detects the interference from co-located benign neighboring nodes as malicious interference; hence, it exhibits poor performance.

8.4.3.2 Impact of Observation Window Size on the Detection

The observation window size plays an instrumental role in the effectiveness of HTE attack detection. Fig. 8.9(b) represents the ROC curves with respect to $\mathbf{d} = 200, 1000, 5000,$ and 10000. We can observe that the detection performance declines as $\mathbf{d}$ decreases; with $\mathbf{d} = 200$, it performs very close to the random detection approach. A larger observation window size provides the NUT with better abilities to

see through the randomness in an attacker's behavior and to differentiate an attacker from a benign one. Therefore, a larger observation window size is required to extract better performance from the detection strategy. As different window sizes offer different performance, a proper choice of **d** depends upon the cost and time-criticalness of the application.

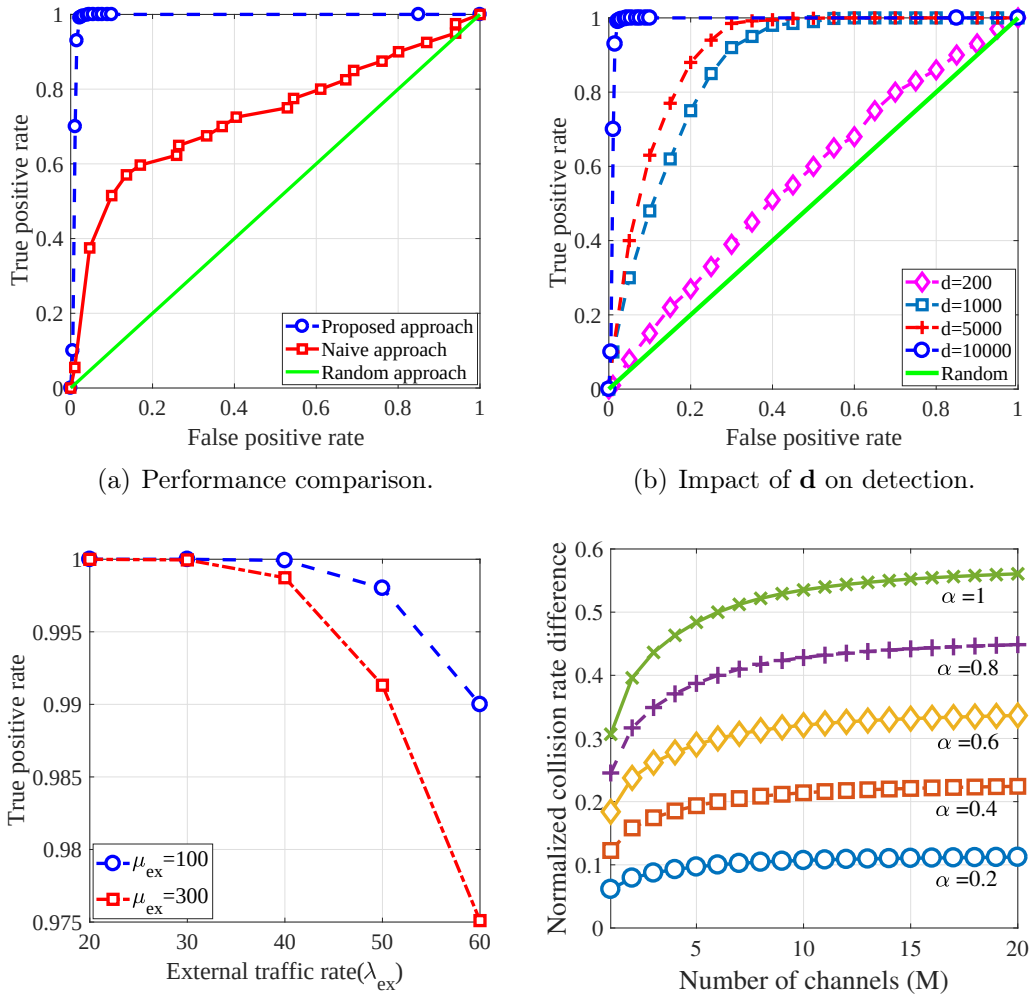
8.4.3.3 Impact of λ_{ex} and μ_{ex} on the Detection

The traffic parameters of the attacker impact the performance of the proposed detection model. In Fig. 8.9(c), we represent the true positive rate vs. λ_{ex} (with a fixed false positive rate, 0.05) to illustrate the relationship between them. We can observe from the figure that the true positive rate decreases with the increase in λ_{ex}. Though a lower λ_{ex} facilitates heightened attack performance (Figs 8.6(c)-(d)), it also increases the probability of detection. Moreover, μ_{ex} also impacts the true positive rate. Hence, these findings create a practical design challenge for an attacker who wants to maximize the attack efficiency and remain undetected at the same time.

8.4.3.4 Impact of M and α on the Detection

The proposed signature-based detection model weighs in different network parameters to model a benign hidden terminal and a malicious one. Among them, the number of channels (M) and the hidden terminal factor (α) play pivotal roles. Intuitively, as the number of channels increases, the probability of collision decreases because co-existing IoT nodes have more channels to utilize. However, the number of channels does not make significant difference in collision rate after it passes a certain threshold, such as $M = 5$ in Fig. 8.9(d) where the normalized collision rate difference represents the difference between state transition probabilities P_{23} of benign and malicious hidden terminals. Here P_{23} represents the probability of experiencing interference from hidden terminals while the NUT is receiving.

However, as α increases, the difference increases significantly. Though higher values



(a) Performance comparison.

(b) Impact of **d** on detection.

(c) Impact of λ_{ex} and μ_{ex} on detection.

(d) Impact of M and α.

Figure 8.9: The HTE attack detection.

of α offers more performance increase for the attacker, it also exposes the attacker to higher risks of detection. Thereby an attacker remains constrained in its attack performance to avoid detection.

8.4.4 Qualitative Comparison with the Literature

The proposed signature-based detection strategy depends on learning the context of each transmission from its neighboring nodes; therefore, it requires different set of information than traditional strategies and, in some cases, may incur additional computational and memory resources. In this subsection, we shed light on these from a qualitative perspective.

8.4.4.1 Detection Parameters

Unlike general network performance indicators—such as packet delivery ratio, signal strength, bad packet ratio, throughput, delay—the proposed context-based detection strategy relies on the traffic characteristics of neighboring nodes from external networks (i.e., λ_{ex} and μ_{ex}) and the network topology (i.e., α).

Traditional detection strategies try to determine whether the NUT is under attack, and they do not consider the source of interference (or jamming). In contrast, our proposed strategy tries to determine whether the NUT is under attack based on the source of interference. Besides identifying the attack, this approach provides the ability to identify the attacker; this allows us to build a context-based detection model.

8.4.4.2 Computational and Memory Cost

The computation tasks are divided into two stages: i) offline phase: the NUT captures behaviors of a benign and a malicious node using the proposed Markov model and, afterward, the Markov model produces closed-form expressions to feed into the detector module. Note that, learning finishes in this phase and no further learning is required in the online phase and ii) online phase: the NUT keeps track of PUs' traffic parameters (λ_p and μ_p), EX's traffic parameters (λ_{ex} and μ_{ex}), and network topology (α), which are available from the sensing process. The computation steps are constant for each external node and increase linearly with the number of neighboring external nodes. Therefore, the computational cost, though higher than some traditional techniques, is tractable to support dense networks. However, unlike most traditional jamming detection strategies, this strategy incurs memory cost to maintain the tracking of the required parameters.

CHAPTER 9: PROPOSED DEFENSE AGAINST HIDE AND SEEK ATTACK: JUMP AND WOBBLE

This chapter proposes a safeguard approach to counteract the random-HTE attack by modeling the interaction between the attacker and the defender as an MDP-based game with three available actions: *stay*, $handoff$, and *route*. Besides *stay* and $handoff$, it utilizes the routing diversity in dense IoT networks to increase the heterogeneity of defense. In *route*, instead of transmitting the packet directly to the intended device, an IoT device utilizes intermediate devices to forward the packet to that receiver. The *route* action is based on the constraint that it is highly unlikely for an HTE attacker to remain hidden to the victim and impersonate an exposed terminal to all neighboring nodes of the victim (because of the proximity of IoT devices in dense scenarios) at the same time. In the following subsections, a single agent (i.e., a defender/victim) MDP-based defense method to avoid the random-HTE attack is modeled.

9.1 Formation of the MDP

As discussed, the attacker has a limited sensing capability, and the channel-hopping sequence of the defender is unknown to the attacker. Therefore, the attacker iteratively sweeps through the available channels—following a random attack sequence—to detect the operating channel of the victim. Meanwhile, the defender takes an action at the end of each time slot, based on the perceived current state. The defender receives an immediate reward $U(t)$ in the t_{th} time slot,

$$\begin{aligned} U(t) = & R_1.\mathbb{1}(Direct\ successful\ transmission) \\ & - F.\mathbb{1}(Transmission\ failure) - C.\mathbb{1}(Handoff\ cost) \\ & - P.\mathbb{1}(Penalty\ for\ policy\ violation) \\ & - Q.\mathbb{1}(Packet\ drop) + R_2.\mathbb{1}(Routing\ reward), \end{aligned} \tag{9.1}$$

where $\mathbb{1}(\cdot)$ is an indicator function of the event in brackets.

In MDP, the employed policy impacts the current state and also the future states; therefore, the expected discounted reward of this game with infinite horizon is,

$$\overline{U} = \sum_{t=1}^{\infty} \delta^{t-1} U(t), \tag{9.2}$$

where δ represents the discount factor $(0 < \delta \leq 1)$. It signifies the importance of the future reward values.

9.1.1 Markov Model

This subsection enumerates the proposed MDP model and defines state space, action space, state transition probabilities, and rewards. As discussed, the attacker randomly jumps between ON and OFF states and performs sweeping through the channels only when it is in the OFF state; hence, this particular part of the attacker's strategy is Markovian. In addition, the probability of detecting the operating channel of the victim (in the OFF period) depends on the channels that have been visited earlier in the sequence. These two strategies together help to conform the essential requirement of the Markov process, i.e., the future state depends only on the current state.

9.1.2 Markov States

The state represents the status of the defender at the end of a time slot, which is deduced from the embedded SINR and RSS information of ACK and NACK messages. a state is defined based on the state variables $[\mathrm{ACK_t}, \mathrm{IF_t}, \mathrm{CS_t}]$, and their descriptions are,

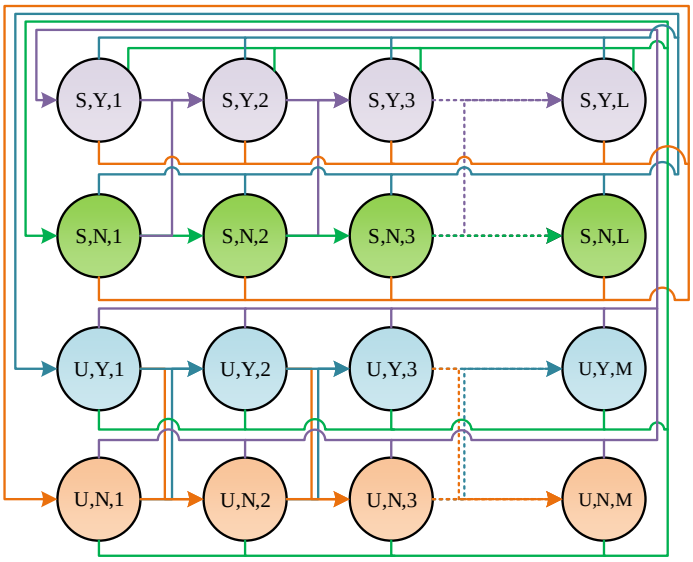


Figure 9.1: The proposed Markov model.

ACK_t: denotes whether an ACK message (ACK_t = S) or a NACK is received (ACK_t = U) in time slot t.

IF_t: denotes whether the transmitted packet experienced interference (IF_t = Y) or not (IF_t = N) in time slot t.

CS_t: denotes the consecutive successful or failed transmission attempts, where CS_t $\in \{\mathbb{Z} > 0\}$.

The states represent a combination of these state variables. Here, the proposed MDP (Fig. 9.1) has four kinds of states:

S, Y, i : The defender hand-offs to a new channel and had i consecutive successful transmissions, despite experiencing co-channel interference in the current slot.

S, N, i : The defender hand-offs to a new channel and had i consecutive successful transmissions without any interference.

U, Y, j : The defender experienced j consecutive transmission failures, the current one due to co-channel interference.

U, N, j : The defender experienced j consecutive transmission failures, the current one due to channel fading.

For notational convenience, the commas in-between is discarded and the whole state

space is represented as $\mathbb{X} \triangleq \{SY_1, \cdots, SN_1, \cdots, UY_1, \cdots, UN_1, \cdots\}$. As a design consideration, $1 \leq i \leq L$ is assumed, where after L consecutive successful transmissions the defender will take action $handoff$, and $1 \leq j \leq M$, where M denotes the maximum transmission attempts after which the packet will drop.

9.1.3 Actions

Here, three actions are available at each state,

stay (s): The defender remains on the current channel in the next time slot and initiates a transmission.

$handoff$ (h): The defender randomly hands-off to a new channel in the next time slot and initiates a transmission.

route (r): The defender randomly hands-off to a new channel and forwards the packet to an intermediate node.

The whole action space is represented as $\mathbb{A} \triangleq \{s, h, r\}$.

9.1.4 Transition Probabilities

As the attacker sweeps through its attack channel sequence, at state SN_i, only $\max(N - i \cdot n, 0)$ channels have yet to be visited by the attacker, and another n channels will be visited in the subsequent slot. Therefore, the probability of detecting the victim's transmission—with action *stay*—without experiencing channel fading is,

$$\Pr_{i,i+1}^{det|s} = \begin{cases} \dfrac{n}{N - i \cdot n}, & \text{if } i < K \\ 1, & \text{otherwise,} \end{cases} \tag{9.3}$$

where it is considered that the attacker is in its OFF period and actively sweeping through the channels. However, the attacker may also reside in the ON period, and the victim may not experience malicious interference in the current cycle (i.e., successful transmissions for L slots). The transition probabilities from state SN_i with action *stay* is,

$$
\begin{aligned}
&\Pr(SN_{i+1}|SN_i, s) = (1 - \mathrm{P_{fad}})(1 - \mathrm{Pr}_{i,i+1}^{att|s|SN}),\\
&\Pr(SY_{i+1}|SN_i, s) = (1 - \mathrm{P_{fad}})\mathrm{Pr}_{i,i+1}^{att|s|SN}(1 - \nu),\\
&\Pr(UN_1|SN_i, s) = \mathrm{P_{fad}},\\
&\Pr(UY_1|SN_i, s) = (1 - \mathrm{P_{fad}})\mathrm{Pr}_{i,i+1}^{att|s|SN}\nu,
\end{aligned}
\tag{9.4}
$$

where $\mathrm{Pr}_{i,i+1}^{att|s|SN}$ represents the probability of experiencing malicious interference from the attacker in the $(i+1)^{th}$ slot, where $1 \leq i \leq L-1$. $\mathrm{Pr}_{i,i+1}^{att|s|SN}$ depends on two factors: 1) the current traffic state (i.e., ON or OFF) of the attacker and 2) its duration in the OFF state (i.e., the number of channels it has swept through). It is represented as,

$$
\mathrm{Pr}_{i,i+1}^{att|s|SN} =
\begin{cases}
(1 - \rho_{ex})(1 - \beta)^i \mathrm{Pr}_{i,i+1}^{det|s} + \\
\rho_{ex}\alpha \sum_{j=1}^{i}(1 - \beta)^{j-1}(1 - \alpha)^{i-j}\mathrm{Pr}_{j-1,j}^{det|s}, & \text{if } i < K \\
(1 - \rho_{ex})(1 - \beta)^K \mathrm{Pr}_{K,K+1}^{det|s} + \\
\rho_{ex}\alpha \sum_{j=1}^{K}(1 - \beta)^{j-1}(1 - \alpha)^{i-j}\mathrm{Pr}_{j-1,j}^{det|s}, & \text{otherwise,}
\end{cases}
\tag{9.5}
$$

where $\rho_{ex} = \dfrac{\beta}{\alpha + \beta}$.

If the first attempt of the attacker is not successful and the defender stays on the current channel, the attacker employs maximum interference power in the next slot. Therefore,

$$
\begin{aligned}
&\Pr(UY_1|SY_i, s) = 1 - \mathrm{P_{fad}},\\
&\Pr(UN_1|SY_i, s) = \mathrm{P_{fad}}.
\end{aligned}
\tag{9.6}
$$

Now, the state transition probabilities from channel fading states (i.e., UN_j) with action *stay* is,

$$\begin{aligned}
&\Pr(SN_1|UN_j, s) = (1 - \mathrm{P_{fad}})(1 - \Pr_{UN,1}^{att|s|UN}),\\
&\Pr(SY_1|UN_j, s) = (1 - \mathrm{P_{fad}})\Pr_{UN,1}^{att|s|SN}(1 - \nu),\\
&\Pr(UN_{j+1}|UN_j, s) = \mathrm{P_{fad}},\\
&\Pr(UY_{j+1}|UN_j, s) = (1 - \mathrm{P_{fad}})\Pr_{UN,1}^{att|s|SN}\nu,
\end{aligned} \tag{9.7}$$

where $1 \leq j \leq M - 1$. Once the defender experiences a packet drop because of channel fading, the sweeping approach of the attacker resets (from defender's point of view). Therefore, the detection probability depends on the state of the attacker and $\Pr_{UN,1}^{att|s|UN} = (1 - \rho_{ex})\frac{n}{N}$.

Now, similar to SY_i states, the transition probabilities from state UY_j with action *stay* is,

$$\begin{aligned}
&\Pr(UY_{j+1}|UY_j, s) = 1 - \mathrm{P_{fad}},\\
&\Pr(UN_{j+1}|UY_j, s) = \mathrm{P_{fad}},
\end{aligned} \tag{9.8}$$

To avoid detection, a defender exploits the diversity in multi-channel network by taking the action $handoff$. When a defender takes action $handoff$ from states SN_i, it randomly selects a channel from the remaining $N - 1$ channels (discarding the current one). Therefore, the probability that the new channel is detected by the attacker depends on three factors:

- The current traffic state of the attacker: whether the attacker is in the ON or OFF state, and how far in the past the last state transition has occurred.
- The new channel was visited earlier in the attack sequence: whether the new channel is one of the $i \cdot n$ channels visited by the attacker.
- The new channel was not visited earlier in the attack sequence: whether the new channel is among one of the $N - i \cdot n - 1$ channels, which were not visited by the attacker, and it will not be visited in the next slot.

Hence, assuming the attacker is in the OFF state, the probability of detection from

state SN_i with action $handoff$ is,

$$\Pr_{i,1}^{det|h|SN} = \frac{N - i \cdot n - 1}{N - 1} \Pr_{i,i+1}^{det|s}. \tag{9.9}$$

Now, after the current traffic state of the attacker is incorporated, the probability of experiencing malicious interference from state SN_i with action $handoff$ is,

$\Pr_{i,1}^{att|h|SN} =$

$$\begin{cases} (1-\rho_{ex})(1-\beta)^i \Pr_{i,1}^{det|h|SN} + \rho_{ex}\alpha(1-\alpha)^{i-1}\dfrac{n}{N} \\ +\rho_{ex}\alpha \sum_{j=2}^{i}(1-\beta)^{j-1}(1-\alpha)^{i-j}\Pr_{j-1,1}^{det|h|SN}, & \text{if } i < K \\ (1-\rho_{ex})(1-\beta)^K \Pr_{i,1}^{det|h|SN} + \rho_{ex}\alpha(1-\alpha)^{K-1}\dfrac{n}{N} \\ +\rho_{ex}\alpha \sum_{j=2}^{K}(1-\beta)^{j-1}(1-\alpha)^{i-j}\Pr_{j-1,1}^{det|h|SN}, & \text{otherwise.} \end{cases} \tag{9.10}$$

The transition probabilities from state SN_i with $handoff$ is,

$$\begin{aligned} &\Pr(SN_1|SN_i,h) = (1-\mathrm{P_{fad}})(1-\Pr_{i,1}^{att|h|SN}), \\ &\Pr(SY_1|SN_i,h) = (1-\mathrm{P_{fad}})\Pr_{i,1}^{att|h|SN}(1-\nu), \\ &\Pr(UN_1|SN_i,h) = \mathrm{P_{fad}}, \\ &\Pr(UY_1|SN_i,h) = (1-\mathrm{P_{fad}})\Pr_{i,1}^{att|h|SN}\nu, \end{aligned} \tag{9.11}$$

The transition probabilities from state SY_i with $handoff$ is,

$$\begin{aligned} &\Pr(SN_1|SY_i,h) = (1-\mathrm{P_{fad}})(1-\Pr_{i,1}^{att|h|SY}), \\ &\Pr(SY_1|SY_i,h) = (1-\mathrm{P_{fad}})\Pr_{i,1}^{att|h|SY}(1-\beta_1), \\ &\Pr(UN_1|SY_i,h) = \mathrm{P_{fad}}, \\ &\Pr(UY_1|SY_i,h) = (1-\mathrm{P_{fad}})\Pr_{i,1}^{att|h|SY}\beta_1, \end{aligned} \tag{9.12}$$

where like $\Pr_{i,1}^{att|h|SN}$, $\Pr_{i,1}^{att|h|SY}$ depends on the same three factors; however, unlike the former, the attacker is in the OFF period and has detected the operating channel in the current slot. Note that the attacker does not randomize its attack sequence unless the attack is successful; hence, the attacker keeps hopping through the same attack sequence. So the probability of experiencing malicious interference from state

SA_i with action *hop* is,

$$\Pr_{i,1}^{att|h|SY} =$$

$$\begin{cases} (1-\beta)\Pr_{i,1}^{det|h|SA}, & \text{if } i=1 \\ (1-\rho_{ex})(1-\beta)^i \Pr_{i,1}^{det|h|SY} + & \\ \rho_{ex}\alpha \sum_{j=1}^{i}(1-\beta)^j(1-\alpha)^{i-j-1}\Pr_{j,1}^{det|h|SY}, & \text{if } 1<i\leq K \\ (1-\rho_{ex})(1-\beta)^{K+1}\dfrac{n}{N} + & \\ \rho_{ex}\alpha \sum_{j=1}^{K}(1-\beta)^j(1-\alpha)^{K-j}\Pr_{j,1}^{det|h|SY}, & \text{otherwise,} \end{cases} \tag{9.13}$$

where unlike $\Pr_{i,1}^{det|h|SN}$, $\Pr_{i,1}^{det|h|SY}$ has $N - i \cdot n$ unvisited channels. Therefore, the probability of detection from state SY_i with action $handoff$ is,

$$\Pr_{i,1}^{det|h|SY} = \frac{N - i \cdot n}{N-1}\Pr_{i,i+1}^{det|s}. \tag{9.14}$$

When the defender takes action $handoff$ from state UN_j and selects a channel randomly from $N-1$ channels, the probability of experiencing malicious interference is,

$$\Pr^{att|h|UN} = \frac{n(n-1)(N-1)}{N(N-1)}, \tag{9.15}$$

where it depends on the scenario whether the attacker visits the same channel in the next slot that the defender visited earlier and experienced channel fading. Now, the transition probabilities from state UN_j with action $handoff$ is,

$$\begin{aligned} &\Pr(SN_1|UN_j,h) = (1-\mathrm{P_{fad}})(1-\Pr^{att|h|UN}), \\ &\Pr(SY_1|UN_j,h) = (1-\mathrm{P_{fad}})\Pr^{att|h|UN}(1-\nu), \\ &\Pr(UN_{j+1}|UN_j,h) = \mathrm{P_{fad}}, \\ &\Pr(UY_{j+1}|UN_j,h) = (1-\mathrm{P_{fad}})\Pr^{att|h|UN}\nu. \end{aligned} \tag{9.16}$$

While performing $handoff$ in state UY_j, the defender randomly selects a channel from $N-j$ channels. Since the attacker also discards j channels from its attack sequence, the probability of detection increases with j. The transition probabilities from state UY_j with action $handoff$ is,

$$\begin{aligned}
&\Pr(SN_1|UY_j,h) = (1-\mathrm{P_{fad}})(1-\mathrm{Pr}_j^{att|h|UY}),\\
&\Pr(SY_1|UY_j,h) = (1-\mathrm{P_{fad}})\mathrm{Pr}_j^{att|h|UY}(1-\nu),\\
&\Pr(UN_{j+1}|UY_j,h) = \mathrm{P_{fad}},\\
&\Pr(UY_{j+1}|UY_j,h) = (1-\mathrm{P_{fad}})\mathrm{Pr}_j^{att|h|UY}\nu,
\end{aligned} \tag{9.17}$$

where $\mathrm{Pr}_j^{att|h|UY} = \dfrac{n}{N-j}$.

Similar to action $handoff$, action $route$ hands-off to another channel, but routes the packet through a forwarding node. Therefore, in the case of action $route$, $\mathrm{Pr}^{att|r|X} = \mathrm{Pr}^{att|h|X}.\mathrm{P}^{\mathrm{det}}_{\mathrm{route}}$, where $\mathrm{P}^{\mathrm{det}}_{\mathrm{route}}$ depends on the topology of the network and the attacker's configuration. Hence, $\mathrm{Pr}^{att|r|X}$ is replaced in (9.11), (9.12), (9.16), and (9.17) to deduce the transition probabilities from corresponding states with action $route$.

9.1.5 Rewards

Let $U(S,a,S')$ represent the reward when an IoT node takes action $a \in \mathbb{A}$ in state $S \in \mathbb{X}$ and enters into state $S' \in \mathbb{X}$. Now using (9.1), rewards are defined as:

$$U(S,a,S') =$$

$$\begin{cases} R_1, & \text{if } \{S,a,S'\} = \{Z,s,Z'\}, Z \in \{SN_i, SY_i\}, \\ & Z' \in \{SN_{i+1}, SY_{i+1}\}, i = 1, \cdots, L-1 \\ -F, & \text{if } \{S,a,S'\} = \{Z,s,Z'\}, Z \in \{SN_i, SY_i\}, \\ & Z' \in \{UN_1, UY_1\}, i = 1, \cdots, L-1 \\ -P, & \text{if } \{S,a,S'\} = \{Z,s,\mathbb{X}\}, Z \in \{SN_L, SY_L\} \\ R_1 - C, & \text{if } \{S,a,S'\} = \{\mathbb{X},h,Z\}, Z \in \{SN_1, SY_1\} \\ -F - C, & \text{if } \{S,a,S'\} = \{\mathbb{X},h,Z'\}, Z' \in \{UN_j, UY_j\}, \\ & j = 1, \cdots, M-1 \\ -Q - C, & \text{if } \{S,a,S'\} = \{Z,h,Z'\}, Z \in \{UN_{M-1}, UY_{M-1}\} \\ & Z' \in \{UN_M, UY_M\} \\ R_2 - C, & \text{if } \{S,a,S'\} = \{\mathbb{X},r,Z\}, Z \in \{SN_1, SY_1\} \\ -F - C, & \text{if } \{S,a,S'\} = \{\mathbb{X},r,Z'\}, Z' \in \{UN_j, UY_j\}, \\ & j = 1, \cdots, M-1 \\ -Q - C, & \text{if } \{S,a,S'\} = \{Z,r,Z'\}, Z \in \{UN_{M-1}, UY_{M-1}\} \\ & Z' \in \{UN_M, UY_M\}, \end{cases} \tag{9.18}$$

where $R2 < R1$ because of the routing delay.

9.2 Optimal Policy

The required components of an MDP is deduced: a finite set of states, a finite set of actions, transition probabilities, and immediate rewards. Now, the defense problem is modeled as an MDP and find the optimal policy by solving it.

In MDP, a *policy* is defined as the action to take in each state, i.e., $\pi : S_n \to a_n$. In other words, a policy maps each state $S \in \mathbb{X}$ to an action $a \in \mathbb{A}$ and is represented by $\pi(S)$. Among all possible policies, the optimal policy returns the maximum expected total discounted payoffs. The value of a state S is defined as the highest expected

payoff, starting from the state S and represented as,

$$V^*(s) = \max_{\pi} E\Big[\sum_{t=1}^{\infty} \delta^{t-1} U(t) \Big| S = s\Big]. \tag{9.19}$$

Here, the optimal policy $\pi^*(S)$ returns the maximum expected payoff. However, after moving from the current state, the remaining part of an optimal policy should still be optimal. Therefore, the first move must maximize the immediate payoff and the future expected payoff, which are conditioned on the current action. This is called the Bellman equation [117],

$$\begin{aligned} Q(S,a) &= \sum_{S'} Pr(S'|S,a)\Big(U(S,a,S') + \delta V^*(S')\Big), \\ V^*(S) &= \max_{Q} Q(S,a), \\ \pi^*(S) &= \text{ argmax } Q(S,a). \end{aligned} \tag{9.20}$$

Now, the value iteration method can be used to derive the optimal defense strategy and show that the solution has a structure mentioned in Proposition 1.

Proposition 1: The optimal policy can be represented by two critical states $l^* \in \{1, 2, \cdots, L\}$ and $m^* \in \{1, 2, \cdots, M\}$,

$$\begin{aligned} \pi^*(SN_i) &= \begin{cases} s, & \text{if } SN_i < SN_{l^*} \\ h, & \text{otherwise,} \end{cases} \\ \pi^*(UY_j) &= \begin{cases} h, & \text{if } UY_j < UY_{m^*} \\ r, & \text{otherwise.} \end{cases} \end{aligned} \tag{9.21}$$

Proof: From (9.3) and (9.4), the probability of a successful transmission with action *stay* (i.e., $\Pr(SN_{i+1}|SN_i, s)$) decreases over i. Therefore, from the definition of $Q(S,a)$ from (9.20), $Q(SN_i, s) - Q(SN_{i-1}, s) < 0$. Now, (9.9) indicates that the probability of a successful transmission with action $handoff$ (i.e., $\Pr(SN_1|SN_i, h)$) increases over i. Therefore, $Q(SN_i, h) - Q(SN_{i-1}, h) > 0$. Now, the optimal action

at state SN_i is *stay* if $Q(SN_i, s) \geq Q(SN_i, h)$, or $handoff$ if $Q(SN_i, h) \geq Q(SN_i, s)$. Since $Q(SN_i, s)$ is decreasing and $Q(SN_i, h)$ is increasing, there exists a l^*, where $Q(SN_{l^*-1}, s) \geq Q(SN_{l^*-1}, h)$ and $Q(SN_{l^*}, h) > Q(SN_{l^*}, s)$, and $l^* \in \{1, 2, \cdots, L\}$.

Similarly, from (9.10)-(9.18), it can be shown that $Q(UY_j, h) < Q(UY_{j-1}, h)$ and $Q(UY_j, r) > Q(UY_{j-1}, r)$. Therefore, there exists a m^*, where $Q(UY_{m^*-1}, h) \geq Q(UY_{m^*-1}, r)$ and $Q(UY_{m^*}, r) > Q(UY_{m^*}, h)$, and $m^* \in \{1, 2, \cdots, M\}$.

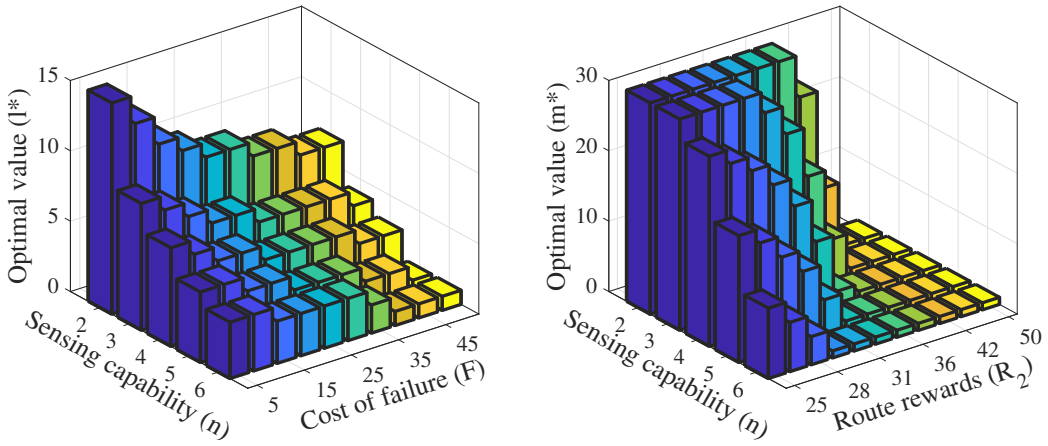
A defender's strategy to use a channel *as long as plausible* and an attacker's *random and iterative strategy* facilitates the design of the attack and defense problem as an MDP. The proposed defense can be summarized in two aspects: 1) a defender keeps utilizing a channel for l^* time-slots, then hands-off to another channel and 2) after m^* successive transmission failures, the defender takes the action *route* to exploit the proximity in dense IoT networks. A defender always takes action *stay* in UN states because $\mathrm{P}_{\mathrm{fad}}$ is constant, and in SY states, it takes action $handoff$. In reality, it is impossible for a defender to know the exact transition probabilities to devise the MDP; hence, it must learn the MDP over time. The Q-learning technique is employed that works as a model-free off-policy method to learn the MDP.

9.3 Performance Evaluation

In this section, the findings are presented to evaluate the performance of the proposed research.

9.3.1 Simulation Setup

Here, the simulation parameters are: communication gain $R_1 = 5$, cost of transmission failure $F = 5$, handoff cost $C = 1$, penalty for policy violation $P = 50$, maximum residence time $L = 30$, maximum transmission attempts $M = 30$, cost of packet drop $Q = M \cdot F$, communication gain for routing $R_2 = 4$, discount factor $\delta = 0.95$, and channel parameters are $\alpha = 0.09$, $\beta = 0.01$, and $N = 60$.



(a) Optimal value l^*; n vs. F.

(b) Optimal value m^*; n vs. R_2.

Figure 9.2: The sensitivity of optimal values to the changes in n, F, and R_2.

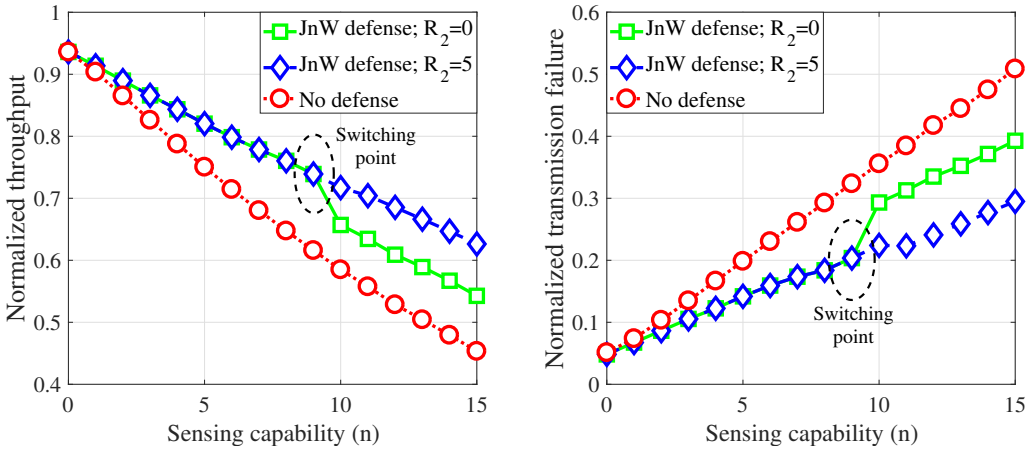
9.3.2 Jump and Wobble

We demonstrate the critical states l^* and m^* (Fig. 9.2) derived from the value iteration of the MDP, with the change in the attacker's sensing capability (n), the cost of transmission failure (F), and the communication gain with routing (R_2).

9.3.2.1 Critical States

In Fig. 9.2(a), l^* decreases with the increase in n. As n increases, K starts to decrease, and IoT nodes have less channels to handoff; hence, IoT devices have to handoff more frequently to avoid the attack. Moreover, as the cost of transmission failure F increases, IoT nodes handoff more to avoid transmission failures (i.e., l^* decreases).

Likewise, in Fig. 9.2(b), m^* maintains a downward trend with the increase in n. However, R_2 largely dictates the action $handoff$ here, and as the reward for routing increases, IoT nodes become more motivated to route the packets through intermediate nodes to avoid interference.



(a) Victim's throughput. (b) Victim's transmission failure.

Figure 9.3: Performance of Jump and Wobble.

9.3.2.2 Routing Gain R_2

Fig. 9.3 compares the performance of this proposed strategy in three scenarios: no defense, jump and wobble with $R_2 = 0$, and jump and wobble with $R_2 = 5$. It illustrates that both $R_2 = 0$ and $R_2 = 5$ follow the same trend until the attacker's sensing capability surpasses $n = 9$, yet the throughput ($R_2 = 0$ line) stays above the no defense line. We denote this moment the *switching point* after which the defender prefers to route data packets (using the action *route*). As R_2 decreases, the victim becomes less motivated to route data packets and the switching point moves further to the left. Likewise, in Fig. 9.3(b), we can observe that the transmission failure increases after the switching point. Therefore, R_2 serves as a tuning parameter between actions $handoff$ and *route*.

CHAPTER 10: CONCLUSION

10.1 Completed Work

This dissertation advocates the shared approach toward spectrum utilization, systematically assesses the security vulnerabilities in such shared spectrum scenarios, and introduces novel counter-mechanism strategies.

First, this research discussed a credible threat in CRNs. Unlike traditional PUE attacks, the proposed *off-sensing* attack exploits the vulnerabilities in the periodic sensing approach. It affects the channel availability of SUs and causes significant reduction in throughput. Though the attack under the common-hopping sequence is illustrated, it can also be implemented in any hopping sequence. Furthermore, it explained a few scenarios of the proposed attack and the knowledge required by a perpetrator to conduct such attack. Then, it explained the goals of such attack and what strategy a selfish/malicious node should take to achieve these goals. Afterwards, it analyzed the attack using a discrete-time Markov model and validated the numerical results through simulations. Lastly, a comparison in attack performance is made between *off-sensing* and PUE attacks. From the comparison, it can be reasoned that the *off-sensing* attack is more detrimental than the PUE attack. While this analysis and observations reveal a new kind of threat to CRNs, they also provide insights on how to design more robust sensing approaches.

Second, a new strategy, random-OS, to perpetrate OS-DoS attack without any predetermined knowledge of the victim's channel-hopping sequence is proposed. Afterwards, an MDP-based safeguard approach is proposed, hide and seek, to avoid and detect the proposed attack. The MDP-game showed that by hopping to random channels, an SU can avoid OS-DoS attack, and when it becomes necessary (based

on rewards) to detect interference it employs an extra-sensing interval to detect an attacker. Here, the victim SU learns the optimal policy by using Q-learning. Lastly, this research proposed an attack inference model to detect the presence of attackers and to reinitialize the learning process to incur less regret. Furthermore, Numerical investigations and simulation results showed that the random-OS outperforms the naive approach and the hide and seek improves the network throughput without ousting the attackers. A thorough search on previous work shows that, this is the first research to introduce a new avenue in designing defensive measures of OS-attack without changing the FCC policy.

Third, a cross-layer route manipulation attack in CR-WMNs, namely OS-RM attack, is proposed. In this attack, a discussion is made on how the off-sensing attack can be weaponized as an aid to influence routing decisions in the network layer. Here, the perpetrator as an intelligent entity and it estimates necessary network information through learning. Furthermore, this research illustrated a general model of the attack and analyzed through extensive simulations on how to coordinate the OS-RM attack in order to achieve the best-attacking result. The analysis and observations not only sheds light on a new kind of threats to the CR-based network, but also provide some insightful findings on how to design cross-layer protocols.

Fourth, this dissertation introduced a novel attack, which exploits a vulnerability in existing spectrum handoff processes. Here, attackers maximize their personal gain by preempting the channel switching process. As they strategically avoid channels where benign SUs are trying to rendezvous and transmit, attackers remain undetected. This research made an strategy to exploit this vulnerability and analyzed the impact of this attack through simulations. While the impacts of such attack is discussed, it also identified the reasons behind this vulnerability. A thorough search on relevant literature concluded that, this is the first research to introduce a vulnerability in the spectrum handoff process in CRNs.

Fifth, this dissertation discussed a vulnerability that the dense IoT deployment will likely bring, i.e., interference from impersonating hidden terminals of external IoT networks, and it illustrated how an HTE attacker can exploit this vulnerability by manipulating its antenna radiation pattern. This research is among the first to foresee this vulnerability of IoT deployment, study it, and, a thorough search on relevant literature yielded it the first to propose an attack feasibility study based on array antenna synthesis. This research utilized the SDR technique and a randomization algorithm to efficiently solve the HTE feasibility problem. Simulation results indicate that the proposed method provides a strong approximation to the HTE feasibility problem. In addition, the observation from the simulation results provides an attacker's conundrum to trade-off between the attack efficiency (i.e., attacking more victims) and the risk of exposure. Lastly, the analysis and observation provide insightful guidance to narrow down the probable locations of an HTE attacker.

Sixth, this research captured the effect of external hidden terminals through a Markov model and detected the aberrant behaviors of HTE attacks. The numerical and simulation results showed the superior performance of the proposed detection model as compared to the naive jamming detection approach.

In the end, this dissertation proposed a new strategy, random-HTE strategy, to perpetrate HTE attacks without any predetermined knowledge of the victim's operating channel. Afterward, it proposed an MDP-based safeguard approach, jump and wobble, to avoid the proposed attack. The results showed that by randomly changing the operating channel, a defender can avoid the attack, and when it becomes necessary, it can route packets through intermediate devices. Furthermore, numerical investigations and simulation results showed that the random-HTE outperforms the naive approach, and the jump and wobble improves the network throughput by avoiding the attacker. A thorough search on relevant literature showed that, this is the first research to introduce a constrained attack model of HTE and to designe

defensive measures against HTE-attacks.

10.2 Future Work

My Ph.D. research opens up many theoretical and practical research possibilities in security assessments of network infrastructures, in spectrum management, and in other related areas in wireless communications. I would like to direct—but not limit my future efforts to—the following research topics:

• Security in Heterogeneous IoT Systems

IoT is a ubiquitous technology that can intricately integrate devices (or things) that surround us, and these devices communicate within themselves by forming a closed connected network to intelligently solve real-life problems. Such a broad scope requires an enormous amount of IoT deployments at our homes, offices, transportation systems, healthcare, and industries. Therefore, in reality, it will be composed of components designed by different manufacturers along with different wireless technologies. This heterogeneity will create unparalleled security loopholes in IoT infrastructures, especially from the perspective of spectrum utilization with a critical question: *how can different wireless technologies securely coexist in the same spectrum?* My current research trend is based on this fundamental question, which specifically addresses security vulnerabilities and their counter mechanisms in spectrum coexistence, spectrum access, and mobility within heterogeneous IoT networks.

• Security in Machine Learning based Systems

Applications of machine learning (ML) based systems are becoming mainstream in smart grid, healthcare, security, finance, and numerous mission critical systems; as a result, the security risk of ML-based systems is emerging as a grave concern. ML-based applications evolve through multiple stages, such as data collection, data preparation, data labeling, model training, testing, and deployment. An attacker with malicious intention can impact the reliability and dependability of a ML-based system by exploiting vulnerabilities at any of these important stages. I believe that my

approaches in assessing security threats can be extended to assess ML-driven infrastructures. I am particularly interested in threat identification, adversarial modeling, and alleviation of threats in ML-driven decision processes in critical wireless network infrastructures.

10.3 Published and Submitted Works

The following list is a summary of my publications and submitted works.

1. Moinul Hossain and Jiang Xie, *"Third Eye: Context-aware Detection for Hidden Terminal Emulation Attacks in Cognitive Radio-enabled IoT Networks,"* IEEE Transactions on Cognitive Communications and Networking, vol. 6, no. 1, pp. 214-228, 2020.

2. Moinul Hossain and Jiang Xie, *"Hide and Seek: A Defense Against Off-sensing Attack in Cognitive Radio Networks,"* to appear in IEEE Transactions on Network Science and Engineering, 2020.

3. Moinul Hossain and Jiang Xie, *"Modeling of Off-sensing Attacks in Cognitive Radio Networks,"* submitted to IEEE Transactions on Networking, Apr 2020.

4. Moinul Hossain and Jiang Xie, *"Off-sensing and Route Manipulation Attack: A Cross-layer Attack in Cognitive Radio based Wireless Mesh Networks,"* submitted to IEEE Transactions on Wireless Communications, Jan 2020.

5. Moinul Hossain and Jiang Xie, *"Covert Spectrum Handoff: A Threat Against Future Spectrum Coexistence,"* in preparation to submit for journal publication.

6. Moinul Hossain and Jiang Xie, *"Jump and Wobble: A Defense Against Hidden Terminal Emulation Attack in Dense IoT Networks,"* submitted to IEEE Global Telecommunications Conference (GLOBECOM), 2020.

7. Moinul Hossain and Jiang Xie, *"Hidden Terminal Emulation: An Attack in*

Dense IoT Networks in the Shared Spectrum Operation," Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), 2019, pp. 1–6.

8. Moinul Hossain and Jiang Xie, *"Detection of Hidden Terminal Emulation Attacks in Cognitive Radio-enabled IoT Networks,"* Proceedings of IEEE International Conference on Communications (ICC), 2019, pp. 1-6.

9. Moinul Hossain and Jiang Xie, *"Hide and Seek: A Defense Against Off-sensing Attack in Cognitive Radio Networks,"* Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2019, pp. 613-621.

10. Moinul Hossain and Jiang Xie, *"Covert Spectrum Handoff: A Vulnerability in the Spectrum Handoff Process in Cognitive Radio Networks,"* Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), 2018, pp. 1-6.

11. Moinul Hossain and Jiang Xie, *"Off-sensing and Route Manipulation Attack: A Cross-layer Attack in Cognitive Radio based Wireless Mesh Networks,"* Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2018, pp. 1376-1384.

12. Moinul Hossain and Jiang Xie, *"Impact of Off-sensing Attacks in Cognitive Radio Networks,"* Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), 2017, pp. 1-6.

REFERENCES

[1] D. Hlavacek and J. M. Chang, "A layered approach to cognitive radio network security: A survey," *Elsevier Computer Networks*, vol. 75, pp. 414–436, 2014.

[2] R. K. Sharma and D. B. Rawat, "Advances on security threats and countermeasures for cognitive radio networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1023–1043, 2015.

[3] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: Spectrum misuse detection in dynamic spectrum access systems," *IEEE Transactions on Mobile Computing*, 2018.

[4] A. G. Fragkiadakis *et al.*, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 428–445, 2013.

[5] R. Chen and J.-M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in *Proc. IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, pp. 110–119, 2006.

[6] Y. Song and J. Xie, "Finding out the liars: Fighting against false channel information exchange attacks in cognitive radio ad hoc networks," in *IEEE GLOBECOM*, pp. 2095–2100, 2012.

[7] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proc. IEEE Conference on Signals, Systems and Computers, 2004.*, vol. 1, pp. 772–776, 2004.

[8] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. IEEE ICC*, vol. 4, pp. 1658–1663, 2006.

[9] FCC. (2003, December), "Et docket no 03-222 notice of proposed rule making and order."

[10] I. A. Akbar and W. H. Tranter, "Dynamic spectrum allocation in cognitive radio using hidden markov models: Poisson distributed case," in *Proc. IEEE SoutheastCon*, pp. 196–201, 2007.

[11] S.-U. Yoon and E. Ekici, "Voluntary spectrum handoff: a novel approach to spectrum management in CRNs," in *Proc. IEEE ICC*, pp. 1–5, 2010.

[12] Y. Zhang, "Spectrum handoff in cognitive radio networks: Opportunistic and negotiated situations," in *Proc. IEEE ICC*, pp. 1–6, 2009.

[13] S. Geirhofer, J. Z. Sun, L. Tong, and B. M. Sadler, "Cognitive frequency hopping based on interference prediction: Theory and experimental results," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 49–61, 2009.

[14] C.-W. Wang and L.-C. Wang, "Modeling and analysis for proactive-decision spectrum handoff in cognitive radio networks," in *Proc. IEEE ICC*, pp. 1–6, 2009.

[15] L. Yang, L. Cao, and H. Zheng, "Proactive channel access in dynamic spectrum networks," *Physical Communication*, vol. 1, no. 2, pp. 103–111, 2008.

[16] X. Liu and J. Xie, "A practical self-adaptive rendezvous protocol in cognitive radio ad hoc networks," in *Proc. IEEE INFOCOM*, pp. 2085–2093, 2014.

[17] X. Liu and J. Xie, "Contention window-based deadlock-free MAC for blind rendezvous in cognitive radio ad hoc networks," in *Proc. IEEE GLOBECOM*, pp. 1–6, 2015.

[18] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Enhanced jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1742–1745, 2013.

[19] X. Liu and J. Xie, "Subset: A joint design of channel selection and channel hopping for fast blind rendezvous in cognitive radio ad hoc networks," in *Proc. IEEE SECON*, pp. 426–434, 2015.

[20] Y. Song and J. Xie, "ProSpect: A proactive spectrum handoff framework for cognitive radio ad hoc networks without common control channel," *IEEE Transactions on Mobile Computing*, vol. 11, no. 7, pp. 1127–1139, 2012.

[21] T. Bansal, B. Chen, and P. Sinha, "Fastprobe: Malicious user detection in cognitive radio networks through active transmissions," in *Proc. IEEE INFOCOM*, pp. 2517–2525, 2014.

[22] N. Bouabdallah, B. Ishibashi, and R. Boutaba, "Performance of cognitive radio-based wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 122–135, 2011.

[23] W. Zhao and J. Xie, "Imex: intergateway cross-layer handoffs in internet-based infrastructure wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1585–1600, 2012.

[24] A. P. Subramanian, M. M. Buddhikot, *et al.*, "Interference aware routing in multi-radio wireless mesh networks," in *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, pp. 55–63, IEEE, 2006.

[25] X. Liu and J. Xie, "A 2D heterogeneous rendezvous protocol for multi-wideband cognitive radio networks," in *Proc. IEEE INFOCOM*, pp. 1–9, 2017.

[26] Y. Song and J. Xie, "Performance analysis of spectrum handoff for cognitive radio ad hoc networks without common control channel under homogeneous primary traffic," in *Proc. IEEE INFOCOM*, pp. 3011–3019, 2011.

[27] Y. Song and J. Xie, "Common hopping based proactive spectrum handoff in cognitive radio ad hoc networks," in *Proc. IEEE GLOBECOM*, pp. 1–5, 2010.

[28] X. Liu and J. Xie, "A slot-asynchronous MAC protocol design for blind rendezvous in cognitive radio networks," in *Proc. IEEE GLOBECOM*, pp. 4641–4646, 2014.

[29] J. Xie, "User independent paging scheme for Mobile IP," *Wireless Networks*, vol. 12, no. 2, pp. 145–158, 2006.

[30] J. McNair, T. Tugcu, W. Wang, and J. L. Xie, "A survey of cross-layer performance enhancements for Mobile IP networks," *Computer Networks*, vol. 49, no. 2, pp. 119–146, 2005.

[31] Y. E. Sagduyu, Y. Shi, T. Erpek, W. Headley, B. Flowers, G. Stantchev, and Z. Lu, "When wireless security meets machine learning: Motivation, challenges, and research directions," *arXiv preprint arXiv:2001.08883*, 2020.

[32] Z. Zhou, X. Chen, Y. Zhang, and S. Mumtaz, "Blockchain-empowered secure spectrum sharing for 5G heterogeneous networks," *IEEE Network*, vol. 34, no. 1, pp. 24–31, 2020.

[33] R. Ali, B. Kim, S. Kim, H. Kim, and F. Ishmanov, "ReLBT: A reinforcement learning-enabled listen before talk mechanism for LTE-LAA and Wi-Fi coexistence in IoT," *Computer Communications*, vol. 150, pp. 498–505, 2020.

[34] A. Garnaev and W. Trappe, "One-time spectrum coexistence in dynamic spectrum access when the secondary user may be malicious," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1064–1075, 2015.

[35] M. Hossain and J. Xie, "Hidden terminal emulation: An attack in dense IoT networks in the shared spectrum operation," in *Proc. IEEE GLOBECOM*, pp. 1–6, 2019.

[36] I. A. Mahady, E. Bedeer, S. Ikki, and H. Yanikomeroglu, "Sum-rate maximization of NOMA systems under imperfect successive interference cancellation," *IEEE Communications Letters*, vol. 23, no. 3, pp. 474–477, 2019.

[37] T. Liu, J. Tong, Q. Guo, J. Xi, Y. Yu, and Z. Xiao, "Energy efficiency of Massive MIMO systems with low-resolution ADCs and successive interference cancellation," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 3987–4002, 2019.

[38] Z. Guo, M. Li, and Y. Xiao, "Enhancing LAA/Wi-Fi coexistence via concurrent transmissions and interference cancellation," in *Proc. IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–10, 2019.

[39] Y. Yan, P. Yang, X.-Y. Li, Y. Zhang, J. Lu, L. You, J. Wang, J. Han, and Y. Xiong, "Wizbee: Wise Zigbee coexistence via interference cancellation with single antenna," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2590–2603, 2014.

[40] R. Kassab, O. Simeone, and P. Popovski, "Coexistence of URLLC and eMBB services in the C-RAN uplink: an information-theoretic study," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2018.

[41] M. Wildemeersch, T. Q. Quek, M. Kountouris, A. Rabbachin, and C. H. Slump, "Successive interference cancellation in heterogeneous networks," *IEEE Transactions on Communications*, vol. 62, no. 12, pp. 4440–4453, 2014.

[42] M. Strasser, B. Danev, and S. Čapkun, "Detection of reactive jamming in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 2, pp. 1–16, 2010.

[43] M. Spuhler, D. Giustiniano, V. Lenders, M. Wilhelm, and J. B. Schmitt, "Detection of reactive jamming in dsss-based wireless communications," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1593–1603, 2014.

[44] K. Thiha, B.-H. Soong, V. Vaiyapuri, and S. Nadarajan, "A new method of defeating reactive jamming: Hardware design approach," in *Proc. IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 184–188, 2019.

[45] S. Xu, W. Xu, C. Pan, and M. Elkashlan, "Detection of jamming attack in non-coherent Massive SIMO systems," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2387–2399, 2019.

[46] F. Fang, Y. Li, Y. Niu, Y. Wang, and C. Han, "Research on attacks detection in CSMA wireless networks," in *Proc. IEEE Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, 2019.

[47] I. Shin, Y. Shen, Y. Xuan, M. T. Thai, and T. Znati, "Reactive jamming attacks in multi-radio wireless sensor networks: an efficient mitigating measure by identifying trigger nodes," in *Proc. 2nd ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing*, pp. 87–96, 2009.

[48] I. Harjula, J. Pinola, and J. Prokkola, "Performance of IEEE 802.11 based WLAN devices under various jamming signals," in *Proc. IEEE MILCOM*, pp. 2129–2135, 2011.

[49] A. Benslimane, M. Bouhorma, *et al.*, "Analysis of jamming effects on IEEE 802.11 wireless networks," in *Proc. IEEE ICC*, pp. 1–5, 2011.

[50] E. Bayraktaroglu *et al.*, "Performance of IEEE 802.11 under jamming," *Mobile Networks and Applications*, vol. 18, no. 5, pp. 678–696, 2013.

[51] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," in *Proc. IEEE INFOCOM*, pp. 2526–2530, 2007.

[52] G.-Y. Chang, S.-Y. Wang, and Y.-X. Liu, "A jamming-resistant channel hopping scheme for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6712–6725, 2017.

[53] E. V. Belmega and A. Chorti, "Protecting secret key generation systems against jamming: Energy harvesting and channel hopping approaches," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2611–2626, 2017.

[54] L. Xiao, H. Dai, and P. Ning, "Jamming-resistant collaborative broadcast using uncoordinated frequency hopping," *IEEE transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 297–309, 2011.

[55] M. K. Hanawal, M. Abdel-Rahman, and M. Krunz, "Game theoretic anti-jamming dynamic frequency hopping and rate adaptation in wireless systems," in *Proc. IEEE WiOpt*, pp. 247–254, 2014.

[56] M. K. Hanawal, D. N. Nguyen, and M. Krunz, "Jamming attack on in-band full-duplex communications: Detection and countermeasures," in *Proc. IEEE INFOCOM*, pp. 1–9, 2016.

[57] M. K. Hanawal, M. J. Abdel-Rahman, and M. Krunz, "Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2247–2259, 2015.

[58] J.-F. Huang, G.-Y. Chang, and J.-X. Huang, "Anti-jamming rendezvous scheme for cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 648–661, 2016.

[59] M. J. Abdel-Rahman and M. Krunz, "Game-theoretic quorum-based frequency hopping for anti-jamming rendezvous in DSA networks," in *Proc. IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pp. 248–258, 2014.

[60] R. Chen, J.-M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 25–37, 2008.

[61] S. Anand, Z. Jin, and K. Subbalakshmi, "An analytical model for primary user emulation attacks in cognitive radio networks," in *Proc. IEEE DySPAN*, pp. 1–6, 2008.

[62] Z. Yuan, D. Niyato, H. Li, J. B. Song, and Z. Han, "Defeating primary user emulation attacks using belief propagation in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 10, pp. 1850–1860, 2012.

[63] C. Chen, H. Cheng, and Y.-D. Yao, "Cooperative spectrum sensing in cognitive radio networks in the presence of the primary user emulation attack," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2135–2141, 2011.

[64] M. Hossain and J. Xie, "Off-sensing and route manipulation attack: A cross-layer attack in cognitive radio based wireless mesh networks," in *Proc. IEEE INFOCOM*, pp. 1376–1384, 2018.

[65] N. Nguyen-Thanh *et al.*, "Surveillance strategies against primary user emulation attack in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 9, pp. 4981–4993, 2015.

[66] D.-T. Ta *et al.*, "Strategic surveillance against primary user emulation attacks in cognitive radio networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 582–596, 2018.

[67] Y. Wu, B. Wang, K. R. Liu, and T. C. Clancy, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 4–15, 2012.

[68] W. Wang, Y. Sun, H. Li, and Z. Han, "Cross-layer attack and defense in cognitive radio networks," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–6, IEEE, 2010.

[69] J. Hernandez-Serrano, O. León, and M. Soriano, "Modeling the lion attack in cognitive radio networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, p. 2, 2011.

[70] D. Nagireddygari and J. P. Thomas, "Mac-tcp cross-layer attack and its defense in cognitive radio networks," in *Proceedings of the 10th ACM symposium on QoS and security for wireless and mobile networks*, pp. 71–78, ACM, 2014.

[71] Z. Yuan, Z. Han, Y. L. Sun, H. Li, and J. B. Song, "Routing-toward-primary-user attack and belief propagation-based defense in cognitive radio networks," *IEEE Transactions on mobile computing*, vol. 12, no. 9, pp. 1750–1760, 2013.

[72] L. Zhang and T. Melodia, "Hammer and anvil: The threat of a cross-layer jamming-aided data control attack in multihop wireless networks," in *Communications and Network Security (CNS), 2015 IEEE Conference on*, pp. 361–369, IEEE, 2015.

[73] Y. Chen, W. Trappe, and R. P. Martin, "Attack detection in wireless localization," in *Proc. IEEE INFOCOM*, pp. 1964–1972, 2007.

[74] K. Bauer *et al.*, "The directional attack on wireless localization: how to spoof your location with a tin can," in *Proc. IEEE GLOBECOM*, pp. 4125–4130, 2009.

[75] T. Wang and Y. Yang, "Analysis on perfect location spoofing attacks using beamforming," in *Proc. IEEE INFOCOM*, pp. 2778–2786, 2013.

[76] Inc. Skyhook, "http://www.skyhookwireless.com," Accessed April 2019.

[77] N. O. Tippenhauer *et al.*, "iPhone and iPod location spoofing: Attacks on public WLAN-based positioning systems," *Technical Report/ETH Zürich, Department of Computer Science*, vol. 599, 2012.

[78] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. ACM MobiHoc*, pp. 46–57, 2005.

[79] G. Thamilarasu, S. Mishra, and R. Sridhar, "A cross-layer approach to detect jamming attacks in wireless ad hoc networks," in *Proc. IEEE MILCOM*, pp. 1–7, 2006.

[80] O. Puñal, I. Aktaş, C.-J. Schnelke, G. Abidin, K. Wehrle, and J. Gross, "Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation," in *Proc. IEEE WoWMoM*, pp. 1–10, 2014.

[81] A. Marttinen, A. M. Wyglinski, and R. Jäntti, "Statistics-based jamming detection algorithm for jamming attacks against tactical MANETs," in *Proc. IEEE MILCOM*, pp. 501–506, 2014.

[82] Z. Lu, W. Wang, and C. Wang, "Modeling, evaluation and detection of jamming attacks in time-critical wireless applications," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1746–1759, 2014.

[83] A. Hamieh and J. Ben-Othman, "Detection of jamming attacks in wireless ad hoc networks using error distribution," in *Proc. IEEE International Conference on Communications*, pp. 1–6, 2009.

[84] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks," in *Proc. IEEE INFOCOM*, pp. 1307–1315, 2007.

[85] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Communications surveys & tutorials*, vol. 13, no. 2, pp. 245–257, 2011.

[86] D. Ciuonzo, A. Aubry, and V. Carotenuto, "Rician MIMO channel-and jamming-aware decision fusion," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 3866–3880, 2017.

[87] M. Hossain and J. Xie, "Detection of hidden terminal emulation attacks in cognitive radio-enabled IoT networks," in *Proc. IEEE ICC*, pp. 1–6, 2019.

[88] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[89] M. Hossain and J. Xie, "Impact of off-sensing attacks in cognitive radio networks," in *Proc. IEEE GLOBECOM*, pp. 1–6, 2017.

[90] M. Hossain and J. Xie, "Covert spectrum handoff: An attack in spectrum handoff processes in cognitive radio networks," in *Proc. IEEE GLOBECOM*, pp. 1–6, 2018.

[91] M. Hossain and J. Xie, "Hide and seek: A defense against off-sensing attack in cognitive radio networks," in *Proc. IEEE INFOCOM*, 2019.

[92] N. An and S. Weber, "Efficiency and detectability of random reactive jamming in wireless networks," in *Proc. IEEE SECON*, pp. 1–9, 2018.

[93] T. Denning *et al.*, "Computer security and the modern home," *Communications of the ACM*, vol. 56, no. 1, pp. 94–103, 2013.

[94] A. Mosenia and N. K. Jha, "A comprehensive study of security of Internet-of-Things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586–602, 2016.

[95] V. Sachidananda *et al.*, "Poster: Towards exposing Internet of Things: A roadmap," in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pp. 1820–1822, 2016.

[96] A. K. Simpson, F. Roesner, and T. Kohno, "Securing vulnerable home IoT devices with an in-hub security manager," in *Proc. IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 551–556, 2017.

[97] T. Yu *et al.*, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things," in *Proc. ACM Workshop on Hot Topics in Networks*, pp. 1–7, 2015.

[98] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *Proc. USENIX Security Symposium*, pp. 1093–1110, 2017.

[99] Pair of Bugs Open Honeywell Home Controllers up to Easy Hacks, "https://threatpost.com/pair-of-bugs-open-honeywell-home-controllers-up-to-easy-hacks/113965/," Accessed July 2019.

[100] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pp. 103–117, 2017.

[101] S. Soltan, P. Mittal, and H. V. Poor, "BlackIoT: IoT botnet of high wattage devices can disrupt the power grid," in *Proc. USENIX Security Symposium*, pp. 15–32, 2018.

[102] Y. Song and J. Xie, "Proactive spectrum handoff in cognitive radio ad hoc networks based on common hopping coordination," in *Proc. IEEE INFOCOM Workshops*, pp. 1–2, 2010.

[103] S. Wang, J. Zhang, and L. Tong, "Delay analysis for cognitive radio networks with random access: A fluid queue view," in *Proc. IEEE INFOCOM*, pp. 1–9, 2010.

[104] H. Su and X. Zhang, "Cross-layer based opportunistic MAC protocols for QoS provisionings over cognitive radio wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 118–129, 2008.

[105] J. Xie and I. Howitt, "Multi-domain WLAN load balancing in WLAN/WPAN interference environments," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4884–4894, 2009.

[106] U. Narayanan and J. Xie, "Signaling cost analysis of handoffs in a mixed IPv4/IPv6 mobile environment," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1792–1796, 2007.

[107] K. R. Chowdhury, M. Di Felice, and I. F. Akyildiz, "TP-CRAHN: A transport protocol for cognitive radio ad-hoc networks," in *Proc. IEEE INFOCOM*, pp. 2482–2490, 2009.

[108] X. Liu and J. Xie, "A self-adaptive optimal fragmentation protocol for multi-channel cognitive radio ad hoc networks," in *Global Communications Conference (GLOBECOM), 2016 IEEE*, pp. 1–6, IEEE, 2016.

[109] X. Huang, D. Lu, P. Li, and Y. Fang, "Coolest path: Spectrum mobility aware routing metrics in cognitive ad hoc networks," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pp. 182–191, IEEE, 2011.

[110] I. Pefkianakis, S. H. Wong, and S. Lu, "Samer: Spectrum aware mesh routing in cognitive radio networks," in *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pp. 1–5, IEEE, 2008.

[111] K. R. Chowdhury and I. F. Akyildiz, "Crp: A routing protocol for cognitive radio ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 794–804, 2011.

[112] M. Youssef, M. Ibrahim, M. A. Latif, L. Chen, and A. V. Vasilakos, "Routing metrics of cognitive radio networks: A survey.," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 92–109, 2014.

[113] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[114] K. W. Choi and E. Hossain, "Opportunistic access to spectrum holes between packet bursts: A learning-based approach," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2497–2509, 2011.

[115] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with MAC-layer sensing in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 533–545, 2008.

[116] H. Kim and K. G. Shin, "In-band spectrum sensing in cognitive radio networks: energy detection or feature detection?," in *Proc. ACM MobiCom*, pp. 14–25, 2008.

[117] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[118] V. C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 1, pp. 1–24, 2008.

[119] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in *Proc. IEEE International Symposium on Integrated Network Management (IM)*, pp. 606–611, 2015.

[120] D. Han *et al.*, "Access point localization using local signal strength gradient," in *Proc. International Conference on Passive and Active Network Measurement*, pp. 99–108, 2009.

[121] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Transactions on Signal Processing*, vol. 54, no. 6-1, pp. 2239–2251, 2006.

[122] B. C. Levy, *Principles of signal detection and parameter estimation*. Springer Science & Business Media, 2008.

[123] S. Marano, V. Matta, and L. Tong, "Distributed inference in the presence of byzantine sensors," in *Proc. IEEE Conference on Signals, Systems and Computers*, pp. 281–284, 2006.

[124] M. S. Bartlett, "The frequency goodness of fit test for probability chains," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 47, pp. 86–95, 1951.