A MULTI-CHANNEL DEFENSE AGAINST COMMUNICATION DENIAL-OF-SERVICE ATTACKS IN WIRELESS NETWORKS

by

Ghada Matooq Alnifie
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computer Science

Committee:

	Dr. Robert Simon, Dissertation Director
	Dr. Michael Casey, Committee Member
Sanjeev Setia	Dr. Sanjeev Setia, Committee Member
	Dr. Elizabeth White, Committee Member
	Dr. Hassan Gomaa, Department Chair
	Dr. Lloyd J. Griffiths, Dean, The Volgenau School of Information Technology and Engineering

Date: 10.08.08

Fall Semester 2008
George Mason University
Fairfax, VA

A Multi-channel Defense Against Communication Denial-of-Service Attacks in Wireless Networks

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Ghada Matooq Alnifie
Master of Science
George Mason University,
Bachelor of Science
George Mason University,

Director: Dr. Robert Simon, Professor
Department of Computer Science

Fall Semester 2008
George Mason University
Fairfax, VA

Dedication

To my parents Matooq and Jawhara Alnifie.
To my parents in-law Mohammed and Mashael Alhaidar.
To my best friend and my husband Redha Alhaidar.
To my children Mohammed and Michelle, my true achievement in life.

Acknowledgments

I would like to thank the people who have contributed, in one way or another, to the work presented in this dissertation.

First of all and foremost, I would like to thank my advisor Prof. Dr Robert Simon. Dr Simon provided me with more than research advising, and I feel so fortunate to have the benefit of his support as a mentor. Dr Simon was tough on me to challenge me academically, only to be very gentle, supportive, and understanding when I needed it the most. His motivation to me prevailed through encouragement, support, and devotion. Dr Simon, you showed me the way and made it possible for me. Words can never express the appreciation I feel for all you have done for me and my research. I especially thank you for always believing in me and giving me great confidence.

Second, I would like to thank my committee members: Prof. Dr. Michael Casey, Prof. Dr Sanjeev Setia, and Prof. Dr. Elizabeth White. I especially feel deeply grateful to Dr. White whose contributions to this work exceeded the role of a committee member. Dr. White constantly shared her insight and wisdom. Through her continuous support, she kept me going when I was about to give up and then helped me find my way out when I felt so stuck to finish. I also feel indebted to Dr. Casey for his great support and encouragement, for all his valuable contributions, for providing lots of helpful feedback, and just for being a great person to talk to.

I would like to include a very special appreciation message to Ms. Margret Graham for taking the time to read my dissertation and providing editing feedback. Ms. Graham's help came out of her great generosity, or as she puts it "a mom helping out another mom", without expecting anything in return. I would like to acknowledge the great assistance I received over the years from the distinguished members of our research group: Muhammad Abdulla, Leijun Huang and Bo Zhang, for their contribution in any form towards the successful completion of my dissertation

Finally, my greatest gratitude to the Institute of Public Administration in Riyadh, Saudi Arabia, for supporting me financially and emotionally throughout my research years, for having the confidence in me to succeed, and for giving me a chance.

Table of Contents

List of Tables

List of Figures

Abstract

A MULTI-CHANNEL DEFENSE AGAINST COMMUNICATION DENIAL-OF-SERVICE ATTACKS IN WIRELESS NETWORKS

Ghada Matooq Alnifie, PhD

George Mason University, 2008

Dissertation Director: Dr. Robert Simon

In this dissertation, we focus on the security and reliability of wireless sensor networks (WSNs). We specifically address attacks which target the *availability* of the wireless communication medium. Due to their unattended functioning and over-the-air communication, WSNs are particularly vulnerable to link-level denial-of-service (DoS) attacks against the communication channel. Our main contribution is to design reactive solutions that can achieve a higher throughput out of an attacked region via the use of *parallel* communication paths.

Our approach uses a hybrid Medium Access Control layer consisting of both CSMA and time-slotted channel access mechanisms. We define and analyze a set of distributed transmission scheduling algorithms for use in the attacked region, ranging from management-free to precise coordinated scheduling. To implement our techniques, we develop a distributed data exfiltration protocol which has two variants, one to act as a response against single-hop attacks and another which provides resilience against attacks affecting multi-hop regions.

Extensive experimental evaluation shows that under many attack scenarios our approach achieves a data delivery rate comparable to pre-attack conditions and far lower latency than

other similar approaches. Furthermore, a simulation-based evaluation of the data exfiltration methods and protocols as a fault-tolerance mechanism in protecting a critical infrastructure has demonstrated that its incorporation in the application's design will lead to increased reliability and robustness in WSNs.

Chapter 1: Introduction

Providing security and trustworthiness in the wireless domain is an issue of critical importance. Many wireless security threats can be addressed by appropriately modifying traditional security services such as confidentiality, authentication, and integrity to appropriately incorporate them in the wireless domain [1–7]. However, a number of new security threats and challenges have emerged as a result of the shared nature of the medium, which did not exist in the wired domain. For instance, an interruption of communication can be easily launched by adversaries capturing the network's transmission channel, also known as *common channel*, shared by the different network devices or *nodes*. Such attacks present a significant challenge in the wireless environment and may have severe consequences since they can effectively lead to a complete block of the transmission and/or reception capability of a network node. Gligor, in [8], states the following: "*when authorized users are not provided a requested service within a defined maximum waiting time, we say that a denial-of-service (DoS) violation has occurred*" [8]. Therefore, when network nodes are prevented from accessing the channel to perform their transmissions, we say that a *communication denial-of-service* attack has commenced.

We note that current radio technology provides multiple channels available for communication [9]. In this dissertation, we investigate the role of multiple channel communication in providing sufficient numbers of concurrent transmission paths as a way of defeating such attacks. We contribute a novel design framework for the purpose of *data exfiltration*. The goal is to achieve intelligent flooding of sensor data in the direction of the network's operator when part of the network is prevented legitimate access to the network's common radio channel. This is achieved via the dynamic and efficient assigning of the set of extra channels to the different nodes undergoing the attack so that the maximum possible amount of data is exfiltrated. We focus on such attacks in the *wireless sensor networks* domain.

Compared to previous research, the proposed framework in this thesis is the first to exploit *concurrent* multi-channel communication as a defensive mechanism against attacks which target the wireless medium. Most proposed solutions in the literature use single channel communication for packet transmissions and receptions. Concurrent communication, done via the use of multiple channels simultaneously, provides significant benefits in terms of the amount of data being transmitted. The maximum throughput of using a single-channel system is bounded by the bandwidth of one channel [10], our main contribution in this dissertation is to design reactive solutions that can achieve a higher throughput out of an attacked region via the use of parallel communication paths.

1.1 Hypothesis

An effective, immediate and robust response to communication DoS attacks in WSNs is achieved via the **concurrent** *multi-channel exfiltration of sensor data.*

1.2 Problem

Wireless networks have gained huge popularity over the last decade and are being deployed in a variety of manners, ranging from wireless local area networks to mobile ad hoc networks. One promising type of is the wireless sensor networks (WSNs) [11]. WSNs are distributed systems consisting of tiny self-powered nodes equipped with radios, environmental sensors, memory and processing capabilities. Such networks can be rapidly deployed at low cost, thereby enabling large-scale, on-demand monitoring and tracking over a wide area. In addition, the sensor nodes can be placed in difficult to monitor events or positions such as during severe weather, wildfires, earthquakes, volcanic activities, the presence of chemical, biological or nuclear agents and in structural and habitat monitoring [12]. The sensors' task is to measure physical properties at different spatial and temporal positions. Data is periodically sent back to a *sink node* which monitors and actuates based on the received data. In-network processing on the collected data is also performed and results are periodically

communicated with one or more sink nodes and sent back to the *base station* [13]. The base station keeps track of received data and reports are generated and maintained for further analysis by the network operator. For many WSNs applications, continuous and timely data delivery is critical for the successful maneuver of the network.

However, the wireless nature of the medium on which sensor networks are built, coupled with their unattended deployment, makes them an easy target for adversaries to launch simple attacks against the radio channel which may alter data reception at the base station. An attacker can simply disable the medium access layer protocol or send a powerful signal to interfere with the physical layer decoding of messages thereby disrupting communication on the channel. If this persists, the resulted interruption in communication between the wireless devices will eventually lead to a form of denial-of-service (DoS) attack. Many WSNs operate and depend on the real-time monitoring of the different data and events. A delay in the reception of critical data may, in fact, alter the whole purpose of the sensor network.

Unattended sensors are vulnerable to radio interference attacks both at the physical and link layer [1]. Physical layer attacks involve producing sufficient levels of radio interference in order to prevent packets being sent or received. Link layer attacks produce the same effect by attacking the medium access control protocols. The purpose of such attacks can be to delay or prevent nodes from reporting their readings to the base station, to deplete vital network resources such as energy, or both [1]. Therefore, there is a clear need for defense mechanisms that can ensure timely data delivery in spite of ongoing attacks. We note that such attacks can be launched for no other reason than to provoke the targeted system to start rapidly depleting resources, for instance, in an attack leading to battery exhaustion due to excessive communication. A sensor node's lifetime is dependent on its battery's lifetime, however, there are many emerging wireless sensor network applications with different requirements and constraints. In a time-critical application, for instance, delivering data of each individual sensor node back to the sink with minimum delay becomes crucial and important. In such scenarios, the utility of the application in detecting and recovering from a communication DoS attack is far greater than maximizing system lifetime

by conserving energy. As part of this dissertation research, we consider situations when a WSN is deployed for the purpose of a critical infrastructure protection in which case the guarantee of continuous delivery of the data surpass the need for energy saving.

In summary, attacks that target the wireless medium are not addressed by conventional security measures found in the wired networks, furthermore, preventing them is impossible with the current radio technology [14]. Responding against such attacks seems to be the only way to defeat them, and this response can only be achieved through careful design of sensor network architecture and security protocols. Wood and Stankovic in [1] concluded that unless this issue is taken into account at design time, sensor networks will remain vulnerable to denial-of-service attacks on the wireless medium. We now recap the problem addressed in this thesis and introduce our terminology.

A Note on Terminology. Denial-of-service attacks in the traditional sense involve occupying the targeted resource and eventually preventing the targeted resource from providing its intended services to recipients. In the wireless domain, an adversary is empowered to launch similar types of denial-of-service attacks which target the wireless medium. For instance, an attacker can ignore the medium access protocol and continue transmitting its signal on the wireless channel resulting in repeated backoffs and collisions and eventually communication blocks. Such a behavior will occupy the wireless channel and hence other network nodes will be denied the legitimate services provided by the wireless channel. In this dissertation, we use the terms *communication DoS attacks* and *radio interference attacks* interchangeably to refer to types of attacks which target the *availability* of the wireless channel.

1.3 Approach

In this dissertation, we present a reactive multi-channel defense against communication DoS attacks. We mainly consider attackers in the sensor communication paradigm, and we specifically focus on device level attacks. We emphasize that there is a very broad range

of capabilities is available to the attacker, ranging from whether the interferer is incidental or intentional, powerful or resource-constrained, narrow band or broad band, or static or adaptive. We note that if the attacker is a high-powered, broadband source of interference (e.g. capable of occupying all channels simultaneously), then there is no hope of defending against it. Instead of considering powerful aggressive adversary models for which the only viable defense might be powerful physical layer techniques, in this dissertation, we consider a non-intentional or a resource-constrained attacker which blocks only a subset of the available channels at a time. This attacker model been studied numerously in the literature and was shown to be very effective in disrupting communication and affecting data delivery at the base station [1, 14–16].

The current radio technology on a wireless sensor device provides multiple channels available for use [9]. To provide resilience against the type of attacks outlined above, we propose a framework design and a methodology for reactive multi-channel data exfiltration. The motivation behind our work is the assumption that once an attack is detected the essential response is to rapidly *exfiltrate* sensor readings from the affected region to guarantee continuous data flow back to the sink. By using multiple channels, simultaneous communications can occur to improve effective network capacity and throughput [10]. Multi-channel communication allows multiple nodes in the same neighborhood to transmit concurrently on different channels, without interfering with one another. Therefore, the process of the actual exfiltration of the data can be achieved via exploiting this channel diversity to establish concurrent paths out of the attacked region. A major challenge in this research is in designing defensive multi-channel protocols that allow an ad hoc network (operating on a single common communication channel) to use the multiple channels provided at the physical layer simultaneously and in an efficient manner to increase effective exfiltration throughput.

1.3.1 Data Exfiltration

Exfiltration is military jargon for exiting an area, usually behind enemy lines or in enemy territory, to withdraw troops from a dangerous position. In computer communication networks, the term used is **data exfiltration**, which refers to the process of secretly getting data out of the network's boundaries resulting in unlawful exposure of classified and/or unclassified data. In this sense, the act of data exfiltration poses a threat to network security since it leads to undetected leaks in an enterprise's corporate data. An interesting definition found in [17] states that "*data exfiltration is the process of getting data without being noticed. This could be something as simple as walking away with the physical backup tapes to something as complex as using covert channels over the network.*" In this research, we investigate the use of such a threat to a network's private data, however, for a positive cause. We show that data exfiltration, which has been considered to be a vulnerability for a security breach, can be used as a reactive defense mechanism to confront DoS attacks resulting from intentional or unintentional transmission interference.

1.3.2 Reactive Defense Mechanism

A reactive defense mechanism is one which is activated by a victim host after an attack is detected. The reactive nature implies that no overhead is incurred when a network is not under attack. In addition, the defense approach is only performed by the victim node(s) in the part of the network undergoing an attack while the rest of the network remains uninterrupted. We present in this dissertation an architectural approach for rapid data exfiltration where the notion of data exfiltration is used as a *reactive defense mechanism*. The goal is to expand the technical capabilities in applications for providing real-time, robust data delivery even in situations where part of the network is undergoing a communication denial-of-service attack. Under our data exfiltration approach, suffering nodes will be able to create communication paths which avoid(s) the attacker so that appropriate information can be conveyed out of the attacked region. This response is appropriate for many types of sensor network applications, such as perimeter and infrastructure defense systems,

battlefield sensing systems, or homeland security systems. We note that our data exfiltration framework can also act as a fault tolerant mechanism for WSNs, and we evaluate the framework's effectiveness when used as a critical infrastructure protection tool.

The use of multiple channel data exfiltration as a reactive defense will achieve the following:

- The concurrent use of all available channels to effectively increase the sensor's data throughput coming out of an attacked region, thereby undoing the effect of the sustained DoS attack.
- The use of as many channels as possible so that the attacker(s) cannot effectively attack all the channels at once. An important observation is that an attack strategy that targets a static subset of channels is easily thwarted by finding an open channel and simply having the nodes communicate on that channel.
- Maximize the likelihood that each node will routinely have access to an un-attacked radio channel as either a sender or a receiver, which is also important for providing resilience against the case of an attacker targeting more than one channel.

This effective multi-channel communication can only be achieved by careful design of channel assignment scheduling. To the best of our knowledge, this thesis is the first to propose such an approach of using *concurrent* multi-channel communication as a defensive mechanism against communication DoS attacks.

1.4 Methodology

The methodology of this dissertation consists of four major phases: design, analysis, simulation, and application. Figure 1.1 depicts the methodology. Careful design and analysis leads to accurate simulation models which are then used to validate and revise our design approach. In addition, simulation is an essential tool in the case where the model is very complex with many variables and interacting components. The building of a simulation model allows for the replication of an actual system [18]. We then extend the resultant

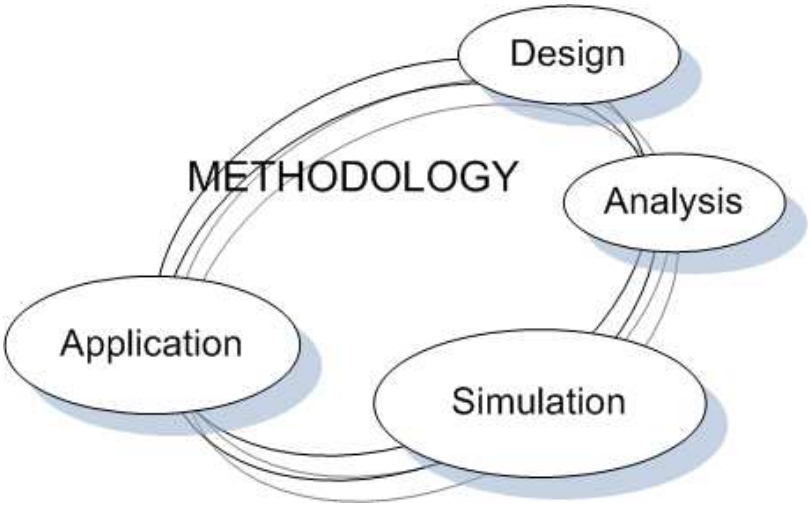


Figure 1.1: Methodology.

simulation model to test out our defense approach on more data intensive distribution networks. We exploit the applicability of our approach in more realistic scenarios and show its effectiveness in protecting a critical infrastructure.

An innovative simulation model is implemented in TOSSIM [19] to evaluate the reactive protocols through simulation experiments. TOSSIM is a discrete event simulator targeted at sensor networking research. It provides substantial support for simulation of protocols and describes systems that are assumed to change instantaneously in response to certain discrete occurrences. The data exfiltration protocols are built on top of the TinyOS [20] routing protocol to function as a reactive response to DoS radio interference attacks. Our approach requires that sensor radios change their channel selections. Thus, the radios employed must have a notion of a channel switch. Most sensor devices have a natural form of channelization that is accomplished by changing the carrier frequency. For our simulation purposes, we extend the default radio model in TOSSIM to support the multi-channel communication.

1.5 Roadmap

This thesis proceeds as follows. In chapter 2, we present background information and a survey about aspects of wireless networking we explore in this dissertation. We focus on

denial-of-service attacks resulting from a communication disruption in the WSNs domain, and we survey the different mitigation approaches.

In chapter 3, we present our data exfiltration approach. We describe a framework that aims at having minimal active security procedures at the link layer and only activate the security defense when an attack is actually present. We formally define the problem of concurrent channel assignments for the purpose of data exfiltration and we propose an efficient technique to address the problem's constraints and meet its requirements. The end result is to achieve the notion of *reactive multi-channel broadcasting*, which incorporates the solution to the radio interference problem into the network design.

In chapter 4, we propose RMBS (Reactive Multi-channel Broadcast Scheduling), which represents the first novel reactive mechanism in our exfiltration framework design. RMBS performs distributed channel assignments and aims at maximizing throughput and minimizing delay. It defines different modes of operation ranging from management-free scheduling to precise coordinated scheduling. The different modes are used to provide methods for scheduling concurrent transmissions where the trade-off is in protocol overhead versus exfiltration performance.

In chapter 5, we present MULEPRO (MULti-channel Exfiltration PROtocol), a fully distributed network based protocol designed to exfiltrate data rapidly from a region undergoing intentional or unintentional interference. MULEPRO performs RMBS scheduling at the link layer, when needed, to facilitate concurrent broadcasting of data.

In chapter 6, we discuss the evaluation aspects of our work, and we present performance analysis of the proposed schemes. We implement the different proposed approaches in the nesC programming language, on the TinyOS platform, and evaluate the relative performance of those schemes via extensive simulations and experiments done in TOSSIM. We use the synthetic network modeling software provided by TOSSIM to generate random networks. We then present simulation results to demonstrate the performance of the MULEPRO protocol under a variety of attack configurations, where we compare the MULEPRO's performance with other existing approaches.

In chapter 7, we present a practical application of our data exfiltration approach, we discuss the deployment of sensor technology on pre-existing networks, and we present the data exfiltration framework's role in protecting a critical infrastructure. The objective is to investigate the role of data exfiltration for the purpose of fault tolerance. In addition, to further investigate the applicability of the reactive protocols in realistic situations and to test the performance of the reactive protocols under real data-intensive network scenarios. We use benchmark water distribution networks along with their realistic data models generated by EPANET [21], a high fidelity data simulator for the water supply system. The findings lead to the conclusion that the data exfiltration framework presented in this dissertation is, in fact, capable of supporting many applications that the real world may require. More importantly, the incorporation of the data exfiltration framework in the application's design will lead to increased reliability and robustness in WSNs.

Finally, in chapter 8, we outline several directions for future work, addressing shortcomings that remain in our design framework as well as promising extensions. We then summarize the contributions made in this dissertation and offer our conclusions.

1.6 Contributions

The main contributions of this dissertation are summarized in table 1.1. The contribution items in the table are organized by the order in which each is introduced in this thesis. We also list the containing chapter for each.

Table 1.1: Summary of contributions.

Contribution	**Chapter**
A Framework and an Architecture for Concurrent Data Exfiltration - A Hybrid Medium Access Control - The Reactive Channel Assignment Problem (RCAP) Model - A Branch and Bound Solution to the RCAP Problem	3
Reactive Multi-channel Broadcast Scheduling - L-VC Mode: Collision-aware Broadcasting - MuVC Mode: Interference-aware Broadcasting - MuVC-S/R Mode: Coordinated Broadcasting	4
Distributed Multi-channel Data Exfiltration Protocols - MULEPRO-1: Single-hop Exfiltration - MULEPRO-k: Multi-hop Exfiltration	5
Simulation and Performance Analysis	6
Data Exfiltration for Fault-tolerance - Application to Critical Infrastructure Protection	7

Chapter 2: Background and Related Work

In this chapter, we provide the necessary background information on which the remainder of the dissertation is built. We also provide a brief discussion of related work in the area of wireless sensor network's vulnerability to communication denial-of-service attacks, mainly to properly explain the scope of our work.

2.1 Background

This section provides some general background about different topics relevant to the work presented in this dissertation.

2.1.1 Wireless Sensor Networks (WSNs)

Wireless networks generally fall into several classes based on their data rate and transmission range. The IEEE 802.11 group of standards specify the technologies for Wireless Local Area networks (WLANs) which focus on providing high data rate and relatively long range for the applications [22]. Wireless Personal Area Networks(WPANs) mainly target low data rate and short range applications. The WLAN's standard is specified by the IEEE 802.14 group [23], a specific example of which is wireless sensor networks. It targets low data rate, low power consumption, and low cost wireless networking and offers device level wireless connectivity.

A wireless sensor network is composed of a large number of sensor nodes that are densely deployed for the purpose of monitoring some environmental or physical conditions at different locations. The sensor nodes (also known as *motes*) are low cost, low power, tiny devices consisting of a power source, a memory unit, a processor, a radio and a sensing component.

The sensing component can measure light, acceleration, position, stress, pressure, humidity, sound, etc. Data samples gathered by the different sensors are passed onto the radio link for transmission from one sensor node to another until it reaches its sink node or base station destination. A sink node and/or a base station monitors the different sensors. The base station allows the aggregation of sensor network data onto a PC or other computer platform. The end result is an *ad hoc* network which serves a specific sensing application. In an ad hoc network, no infrastructure is required. Instead, the different nodes discover each other other by flooding the network with broadcast announcements. Data packets are forwarded from one node to another in a multi-hop fashion.

The applications of sensor networks range from military to home applications. Example applications include habitat monitoring, environment monitoring, health care, military applications, industrial applications, home automation and smart interactive places. Each area of a sensor network application comes with its own technical issues and challenges. For instance, the sensor nodes can be embedded into structures to give constant or periodic reports on structural integrity such as salt content levels in concrete. They can be used in agriculture to give a clear picture of the temperature, humidity, water level, etc. for a given location. They can also be used in traffic management and monitoring by placing these devices on major intersections and streets. In addition, sensors can be used in conjunction with power meters, water meters and other utility meters to transmit data automatically to a central node. These are few of the potential uses of the wireless sensing motes in the WSNs paradigm.

One of the commercially available sensor devices is the MICA mote, developed by UC Berkeley research group. MICA motes run the TinyOS operating system (section 2.1.4). A MICA mote has a size and shape which can range from 2.25 x 1.25 x 0.25 inches rectangular to 1 x 0.25 inches circular. It constitutes an Atmel ATmega 8 bit microcontroller 128L running at about 4 MHz and also contains a 28 kilobytes flash memory. It runs on two AA batteries and alternates between sleep mode and running mode in order to conserve energy consumption. With 8 milliamps power requirement when in operation and only 15 microamps in sleep mode, a MICA mote can run for two years [24].

Recent developments in sensor technology, as seen in Berkeleys MICA2 [25], MICAz [26], and TelosB motes [27] and the IEEE 802.15.4 Zigbee [28], have enabled support for single-transceiver, multi-channel communication [9]. A typical sensor device is usually equipped with a *single radio transceiver*, which can not perform simultaneous transmission and reception, but can work on different channels at different times.

2.1.2 Wireless Channel Access Schemes

A medium access control (MAC) protocol is a sublayer of the data link layer which acts as an interface between the logical link control (LLC) sublayer and the network's physical layer. It enables multiple users to share the finite amount of frequency resources and incorporates some channel access mechanism. For the case of ad hoc networks, channel access protocols are usually characterized as being deterministic or non-deterministic. The non-deterministic schemes allows the different nodes to *compete* for the channel, where a node is only allowed to transmit after sensing the channel clear for some period of time. Such non-deterministic schemes usually facilitate some collision avoidance mechanism to recover from possible collisions. One widely used protocol is Carrier Sense Multiple Access (CSMA). The use of non-deterministic approaches to channel access minimizes overhead. However, an increase in the congestion may drastically degrade throughput as a result of the rise in the probability of collisions [29].

Deterministic access schemes set up *schedules* for individual nodes or links, such that the different nodes' transmissions (or link assignments) are conflict-free in the code, time, frequency or space divisions of the channel. A widely adopted approach is the time division multiple access scheme (TDMA) which provides for efficient scheduling of time-slots to result in collision-free transmitting. The schedules for conflict-free channel access can be established based on the topology of the network, or it can be topology independent [29]. Topology-dependent schemes may require prohibitive overhead associated with constant updates due to changes in topology. Topology-independent schemes are proposed to minimize such overhead, however, they do require that maximum number of nodes and the

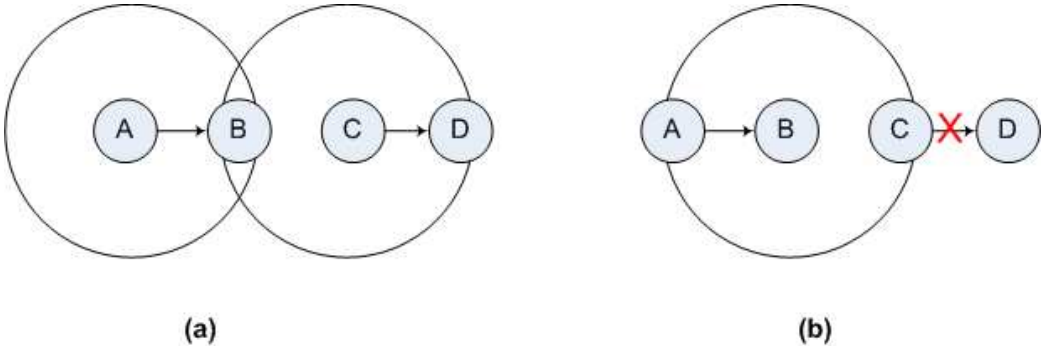


Figure 2.1: The hidden terminal and exposed terminal problems.

maximum degree of nodes in the network to be known beforehand to guarantee successful transmission of each data packet to all one-hop neighbors [29].

In general, the medium access control protocols in ad hoc networks have to combat two problems, namely,"hidden terminal problem" and "exposed terminal problem". A *hidden node* is one that is within the range of the intended destination but out of range of the sender. An *exposed node* is one that is within the range of the sender but out of range of the destination.

The two problems are illustrated in figure 2.1. In the figure, node A is transmitting to node B in part (a). Node C wanting to transmit to node D, senses the channel as clear since node A is *hidden*. If C start transmitting, a collision will occur at node B as a result of the hidden terminal problem. In part (b) of the figure, node B's transmission to node A prevents node C from transmitting to node D even though its transmission will not collide at A or D. C is said to be an *exposed* node in this case.

2.1.3 Sensor Technology for Pre-exiting infrastructure

The application scenario presented in this dissertation is based on the use of sensor technology for pre-existing infrastructure, this subsection provides some background about this topic. As sensor network technology emerges from research laboratories, many attempts are already progressing toward using the sensor technology and WSNs paradigm to automate many of the monitoring tasks which are essential to different practices. Example applications include pollution monitoring [30], petrochemical industry [31], medical assistance

[32], emergency response systems [33] and water distribution networks [34]. In many of these suggested systems, wireless sensors co-exist with the already installed infrastructure for the purpose of augmenting data collection and real-time response. Due to the intelligent distributed capabilities and the ability to operate under severe conditions, WSNs permit data gathering and computation to be deeply embedded in the physical environment. Vital events and data are automatically collected and fully integrated for real-time monitoring and long-term observation purposes. Therefore, WSNs provide a great tool for monitoring critical pre-existing infrastructure. The research community is recognizing the potential role of WSNs in the field of critical infrastructure protection [31, 35–39].

For instance, a wireless sensor network to report pollution and traffic is suggested and being implemented for the city of Cambridge, MA [30]. Researchers at Harvard University and BBN Technologies have designed a wireless network capable of reporting real-time sensor data across the entire city. To conduct such an experiment, wireless sensor nodes were mounted onto telephone poles. Readings from the different sensors allowed the researchers to see the specific locations and times of day when pollution peaks. "The plan is to install 100 general-purpose nodes onto the street lights of Cambridge, drawing power from the city's infrastructure" [30]. Furthermore, to solve the battery lifetime constraint present at a wireless node, more sensor nodes were further mounted on the municipal street lamps. In this way, a sensor node can simply draw power from city electricity. Such an approach opens up a new range of uses for the sensors performing long-term experiments [40].

2.1.4 TinyOS

The simulation work we conduct in this dissertation is done in the TinyOS [20] environment using the TOSSIM simulator [19]. TinyOS is a lightweight, open source operating system specifically designed for embedded sensor networks. Its efficient framework provides for modularity and a resource-constrained, event-driven concurrency model. The modularity feature is achieved via the separation of construction from composition, which allows the operating system to adapt to hardware diversity while at the same time providing for the reuse

of common software abstractions. TinyOS programs are built out of *components* and each of the component's services is specified by an *interface*. Components are statically wired together during run time through their interface, which increases the run-time efficiency of a TinyOS program. The concurrency model is necessary for the concurrency-intensive operation of a sensor network. Two types of concurrency are present in a TinyOS program, *tasks* and *hardware event handlers*. Hardware events are interrupts, caused by a timer, sensor, or communication device. Tasks are a form of deferred procedure call that allows a hardware event or task to postpone processing [41]. The TinyOS distribution contains a simulation tool called TOSSIM, which simulates the event-driven architecture of a sensor network application as a series of discrete events.

TOSSIM is a discrete event simulator targeted at sensor networking research. It provides substantial support for simulation of protocols and describes systems that are assumed to change instantaneously in response to certain discrete occurrences. TOSSIM scales to thousands of nodes and compiles directly from TinyOS code. It provides an efficient simulation tool for developers to test their algorithms and implementations. In this dissertation, we build an innovative simulation model and implement it in TOSSIM in order to evaluate the performance of the designed protocols. The major advantage of implementing in TOSSIM is that the written code for simulation can be used to run on real hardware with little change [42].

2.1.5 EPANET

The application scenario presented in this dissertation makes use of the EPANET [21] simulator for the generation of a physically-based hydraulic model to be used in a data-driven simulation of the exfiltration algorithms. EPANET is a windows-based simulator which models the hydraulic and water quality behavior of water distribution piping systems. EPANET is developed and maintained by the Environmental Protection Agency's [43] Water Supply and Water Resources Division, as a public domain software that may be freely copied and distributed. Given a specification of a water distribution network topology

as an input, an EPANET program performs extended-period simulation of the hydraulic and water quality behavior within pressurized pipe networks, to allow for the monitoring of the pipe network. Pipe networks consist of pipes, nodes (pipe junctions), pumps, valves, and storage tanks or reservoirs. EPANET tracks the flow of water in each pipe, the pressure at each node, the height of the water in each tank, and the concentration of a chemical species throughout the network during a simulation period. In this dissertation, we use the EPANET simulator for building a physically-based data model. We then use the resultant hydraulic model as an input component to our TOSSIM simulation model. The result is an application of our data exfiltration framework which uses realistic benchmark water distribution networks to apply our defense approach for the purpose of critical infrastructure protection.

2.2 Related Work

This section provides previous work which is directly related to our research. We first survey the use of multi-channel communication in the WSNs domain. We also survey the use of Latin square for scheduling concurrent transmissions in multi-channel wireless networks. We then focus on the specific issue of communication DoS attacks, and we investigate current solution approaches. We end this section with an evaluation of the related work.

2.2.1 Multi-channel Communication

The use of multi-channel communication presents an attractive approach to the design of wireless networks [44–48]. Concurrent transmissions performed on the different channels are guaranteed not to collide and hence the system's throughput is increased if multi-channel communication is facilitated. The use of multi-channel communication at the MAC layer also leads to better energy savings. One example of a multi-channel MAC-layer protocol is MMAC [49], proposed for the 802.11 environment. Other work proposed in the literature, on exploiting channel diversity, is based on devices with multiple radio interfaces [50, 51] or with interfaces which can listen simultaneously on different channels [52, 53]. These

protocols require multiple radio transceivers at each node and hence are not applicable in the single-transceiver environment of a WSN. A single-transceiver radio requires that a node can only perform at one channel at a time, in either listening or transmitting mode.

Some papers have evaluated the use of multiple channels communication in the specific domain of WSNs. Multi-channel MAC protocols have been proposed for WSNs to achieve higher throughput and energy savings [54–56]. Simulation results show that they improve performance in WSNs compared with single channel protocols. The work in [57] presents a protocol for load balancing across channels for throughput maximization. Furthermore, multiple channel communication is proposed to achieve more efficient data dissemination in WSNs [58, 59] and to also increase the security of WSNs [60]. The challenge in all such multiple channel protocols is assigning the channels and/or time-slots across multiple nodes to avoid collisions and congestion. In a recent paper, the authors in [61] studied how to efficiently use multi-channel communication to improve performance in WSNs.

In this dissertation, we present a multi-channel defense which exploits multi-channel communication for the purpose of maximizing throughput coming out of an attacked region. Furthermore, we make use of Latin square for performing scheduling concurrent transmissions on the multiple channels as will be explained next.

2.2.2 The Use of Latin Squares in Transmission Scheduling

The study of Latin squares provides an environment rich in important results, in unsolved problems and practical applications. The results of such fields are algebra, finite geometrics, coding theory combinatorial design theory and statistics [62]. A Latin square of order p is a $p \times p$ matrix with entries from a set of p distinct symbols such that each row and each column contains every symbol exactly *once*. The unique pattern of each symbol and the inherent fairness attribute in a Latin square assignment present a very attractive feature for scheduling problems in general. The use of this type of scheduling at the MAC layer, in an ad hoc wireless network, will eliminate the need for information exchange between the participating nodes [63].

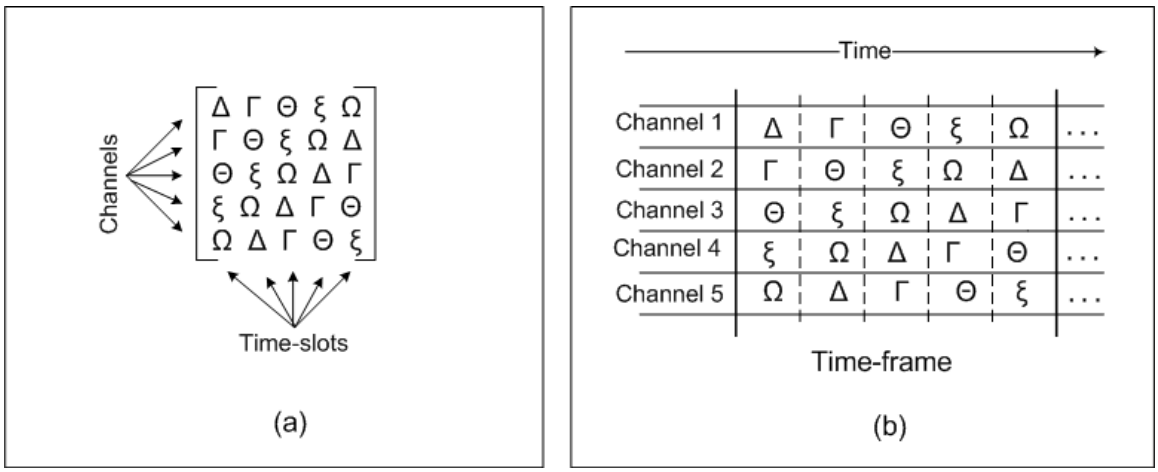


Figure 2.2: Using a Latin square for transmission scheduling. **a)** The Latin square used for the multi-channel scheduling. **b)** Scheduling *concurrent* transmissions based on square symbols on the multiple channels.

When Latin squares are used for transmission scheduling in ad hoc network, they provide a useful tool for setting up schedules for individual nodes to allow for deterministic access. Latin square based scheduling works by assigning nodes to square symbols and then using the pattern of the symbol for the purpose of schedule creation [63]. The uniqueness feature of square symbol assignments during each time-slot allows for simple generation of the concurrent transmission assignments on the different channels. Another nice feature of using Latin square assignments for schedule creation is that transmission schedules remain unchanged. Therefore, the overall network performance is not seriously degraded by the dynamic changes of the network topology [64]. Figure 2.2 shows an example.

Several scheduling approaches based on Latin square assignments have been proposed in the literature [64–67]. Pottie and Calderbank in [65] used Latin square based scheduling in order to improve reuse of time/frequency slots in a cellular radio system. Ju and Li in [64] presented a topology-transparent TDMA scheduling based on Latin squares in multi-channel mobile network. Bao in [66] proposed a medium access scheme based on Latin squares, from which deterministic time-division channel access schedules are generated for nodes in ad hoc networks. The scheduling approach in [66] is applicable in both traditionally studied time division multiple access (TDMA) scheme, and random access

scheme by generating deterministic schedules according to the Latin squares. In [68], Cai, et al. proposed a topology-transparent broadcast scheduling for single channel networks. In the specific domain of sensor networks, Simon and Farrugia proposed in [67] a topology transparent technique for automatic time-slotted link management. In [69], El Moutia, et al. proposed a space division multiple access (SDMA) scheme for sensor networks based on Latin squares which guaranties a collision-free medium access to the different sensor nodes.

In this dissertation, we make use of Latin square scheduling at the link layer to provide the concurrent channel assignments amongst the attacked nodes when implementing our reactive defense.

2.2.3 Radio Interference Attacks in WSNs

In conventional radio technology, two parties communicate on the same (fixed) frequency. Therefore, the sender and the receiver would be using one single common channel for their communication. However, since radio is built on top of an open medium, two communicating parties are vulnerable to interception of their communication or even disruption. Interception is the unauthorized monitoring of radio traffic, whereas disruption is the deliberate interference with the radio signal. In the late 1970s, the US military put forth a new definition document for a system known as SINCGARS (SINgle Channel Ground and Airborne Radio Subsystem) [70]. These radios were designed to be reliable, secure and easily maintained to handle both voice and data. And to provide immunity to eavesdropping and interference attacks, *frequency hopping* was found to be the only effective counter measure to both forms of attacks [12, 70]. Using the frequency hopping method, both the sender and the receiver would change frequency many times per second during their conversation while maintaining perfect synchronization throughout. The hopping sequence used by both parties follows a pseudo-random pattern, which usually has an extremely long repeat time. This makes it impossible for an eavesdropper to be able to follow the hopping sequence, and therefore network communication would be impossible to intercept or disrupt [12,70]. Only users who have programmed their radios with the same frequency, sideband, and hopping

code can communicate, providing complete resilience to intercepting and interference.

The DARPA Packet Radio Network was one of the first ad hoc multi-hop wireless networks [71]. The communication medium in such networks is also shared and hence protecting against interception and/or interference is necessary. Spreading techniques, such as the classic frequency hopping approaches used in the military systems, are also used in this type of communication networks in order to provide resilience to interference [71].

In sensor networks, radio interference affect on communication disruption was identified as a type of denial-of-service attack (DoS) in [1]. The authors provided a taxonomy of DoS attacks for sensor networks from the physical up to the transport layer. Radio interference DoS attacks were recognized to be initiated at the physical layer or the link layer. The authors concluded with the fact that the use of spreading techniques (such as the frequency hopping approach used in military communication) would require much more power and design complexity than what can be offered by a low-cost, low-power sensor device. They also concluded that the best defense against such radio interference DoS attacks would be in the appropriate *detection* and *mitigation*, and we survey both next.

2.2.3.1 Detection

The issue of radio interference DoS attack detection in WSNs was studied in [14–16, 72]. In [73], Wood, et al proposed a heuristic approach for attack detection based on the wireless channel's availability. If the channel availability fell beyond some predefined threshold, an attack is declared. The check can be performed at either the physical layer or the link layer. However, this study posed the issue of attack detection in the loose context of the utility of the communication channel, and presented several factors that might affect the channel's utility. In their approach, each node is assumed to have a detection-module that periodically returns a JAMMED or UNJAMMED message. The message output by the detection module is then broadcast locally, assuming that this special type of message is always guaranteed access to the channel.

Xu, et al. presented several attacker models and explored the need for more advanced

form of detection algorithms to identify DoS attack resulting from radio interference. In [14], Xu, et al. presented two strategies that may be employed by wireless devices to evade a MAC/PHY-layer jamming-style wireless denial-of-service attack. The first strategy, channel surfing, is a form of spectral evasion that involves legitimate wireless devices changing the channel that they are operating on. The second strategy, spatial retreats, is a form of spatial evasion whereby legitimate mobile devices move away from the locality of the DoS emitter. They also note that the fact that no single measurement is sufficient for reliably classifying the presence of an attack is an important observation and necessitates the development of enhanced detection schemes.

In a later work, [16], Xu, et al. examined the feasibility and effectiveness of radio interference DoS attacks and the detection schemes of such attacks. They defined a comprehensive model of attack profiles capable of disrupting communication in some area of the wireless network. They also showed that the *packet delivery ratio (PDR)* measurement is the only single metric that can detect all types of attackers. More specifically, it is shown in [16] that even at high congestion scenarios, the PDR is around 78%. On the other hand, when there is a radio interference attack, PDR drops almost to 0. So, a simple threshold can distinguish a normal congested state of the network from an attack sitiuation on the communication channel. Unfortunately, there are sitiuations when the PDR measurements may lead to a false alarm. A PDR measurement cannot distinguish between an attack scenario and a network failure scenario, like battery failure on a node.

Additional attack strategies were studied by Law et al. in [15]. The authors described attacks against specific MAC protocols that attempt to conserve the energy resources of the attacker, in an effort to complement the attack profiles stated in [14]. The efficiency of the proposed attack methods was quantified in terms of the amount of resources needed to conduct an attack.

2.2.3.2 Mitigation

Radio interference DoS attacks are not addressable through conventional security mechanisms and preventing them is difficult with the current radio technology. Protecting against such attacks seems to be the only way to defeat such attacks. This is done through careful design of sensor network architecture and security protocols. Unless this issue is taken into account at design time, sensor networks will remain vulnerable to such denial-of-service attacks [1]. For instance, protecting against interference (whether intentional or not) is considered an important consideration when designing media access control protocols in WSNs. Some MAC protocols, such as B-MAC [74], use adaptive thresholding for carrier sensing to identify the best parameters for wireless sensor networks application at compile time or run time and thus provide some level of immunity against certain types of constant jamming attacks. The Jammed-Area Mapping (JAM) scheme in [73] identifies regions of attacked sensors to be avoided by routing protocols. Attacked nodes turn themselves into sleep mode to outlast the attacker. However, an intelligent attacker can detect the communication silence and adjust their power consumption accordingly. RID [75] is a real-time adaptive protocol designed to detect interference in radio communications. The concept of this protocol is that a sender sends a high-powered signal followed by a normal-powered signal. The receiver can then infer the interference level at the sender. All of these mentioned protocols make use of interference detection results in order to avoid an attack and hence do not in general provide resistance or mitigation solutions.

In [76], wormhole-based anti-jamming techniques are analyzed. The proposed approaches try to ensure the successful delivery of at least one alarm message by passing attack notification messages out of a jammed region through the creation of wormhole links between sensors, one of which resides out of the attack area. Links are created through frequency hopping over a channel set either in a predetermined or in an ad hoc fashion. A similar approach is presented in [77] which creates a timing channel that serves as the basis for a new link layer overlay, that can allow communication of important messages in spite of the presence of broadband interference. Our work differs in that the goal is to

allow attacked nodes to communicate *while* a communication DoS attack is occurring and to provide continuous data delivery at the base station.

Khattab, et al. in [78] investigate mitigation solutions to communication DoS attacks in the multi-radio context and study optimal interaction between channel hopping and data redundancy enabled by the availability of multiple radios. In [79] they generalize the software-based channel hopping defense into multi-radio wireless networks and compared two defense strategies, namely proactive and reactive. The focus in this dissertation is on the single-transceiver radio networks.

Related to our research is the work presented in Xu, et al [14]. The authors propose channel surfing as an adaptive form of frequency hopping which is performed at the data link layer as opposed to the physical layer. In this type of surfing, instead of continuously hopping from one frequency to another, a node only switches to a different channel when it discovers that the current one is being attacked. Upon detection of the attack, two nodes that want to communicate start to change channels according to a pseudo-random sequence, in an attempt to thwart the attacker by communicating on an available (un-attacked) channel. A shared secret between the two parties is used to generate the pseudo-randomly sequence in order to make scanning more difficult for the attacker. This approach however assumes only a single channel is attacked at a time and does not hold against multiple-channel attacks. In a recent work [80], the authors further explored 2 different approaches to the basic channel surfing strategy: coordinated channel switching, where the entire sensor network adjusts its channel; and spectral multiplexing, where nodes in an attacked region switch channels while nodes on the boundary of an attacked region act as radio relays between different spectral zones. DEEJAM is proposed in [81] as a MAC-layer protocol for defeating mote-class attackers with IEEE 802.15.4-based hardware. The authors defined four attack types and proposed defenses for each of them. However, the DEEJAM protocol provides mitigation solutions against a stealthy attacker targeting a specific MAC protocol. In addition, it assumes that only a single channel is attacked at a time, and so, it does not hold against multiple attackers or even a single attacker targeting multiple channels. In addition, the

protocol results in a very large overhead since the network needs to apply all the four suggested mitigation solutions at the same time. Our work in this dissertation provides mitigation solutions that go beyond an attacker targeting a single channel at a time and provides solutions against multi-channel attackers.

2.2.4 Evaluation of Related Work

Current sensor technology provides multiple channels available for communication. We design reactive defense mechanisms against attacks that target the availability of the radio channel to achieve a higher throughput out of an affected region via the use of parallel communication paths. While many protocols have proposed multi-channel communication as a way to overcome an attack on the wireless medium, none of them make *concurrent* use of the available extra channels as a way of increasing the sensor data throughput coming out of an attacked region. Until the time of this writing, the described research in this thesis is the first to adopt such an approach. We also note that such attacks can be launched for no other reason than to provoke the targeted system to start to rapidly deplete resources, for instance, an attack leading to battery exhaustion due to excessive communication. However, for many applications the utility to the application in detecting and recovering from an attack, which affects the availability of the communication medium, is far greater then maximizing system lifetime by conserving energy.

Chapter 3: Data Exfiltration Framework and Architecture

This chapter presents the framework and architecture of the multi-channel data exfiltration approach. We describe a light-weighted, reactive framework that aims at having minimal active security procedures during normal operations and is only triggered when an attack is detected. We start, in section 3.1, by providing a high-level overview of data exfiltration as a reactive response to a communication DoS attack. We then present the data exfiltration framework architecture, in section 3.2. We explain the framework's role in attack mitigation, identify important design principles and present the architectural abstraction models assumed in our work.

In section 3.3, we present a high-level overview of the general TDMA broadcast scheduling, which establishes the underlying transmission scheduling mechanism of our data exfiltration approach. We formally define the problem of concurrent channel assignments for the purpose of data exfiltration, and we propose an efficient technique to address the problem's constraints and requirements. The end result is to achieve the notion of *reactive multi-channel broadcasting*, which incorporates the solution to the radio interference problem into the network design.

The theoretical investigation presented in this chapter brings forward important design principles, insofar as a part of the thesis is devoted to the design of sound defense protocols based on these principles. We conclude this chapter in section 3.4, where we introduce some important definitions and preliminary background useful in the development of the reactive scheduling approach presented in the next chapter.

3.1 Reactive Data Exfiltration Overview

A high-level visualization of the multi-channel data exfiltration approach, as a reactive defense against a communication DoS attack, is presented in figure 3.1. During the *normal*

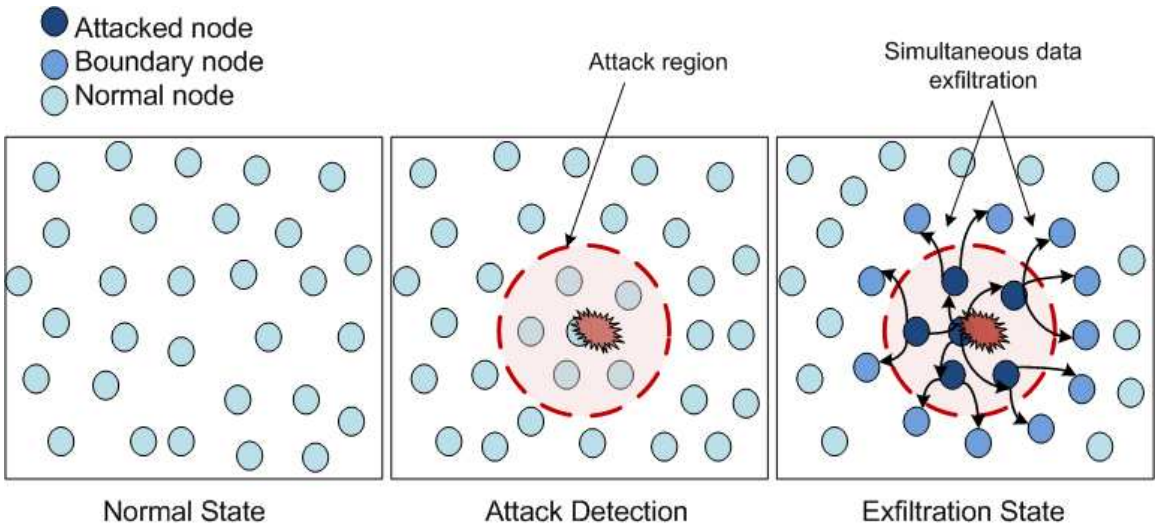


Figure 3.1: Thwarting a communication DoS attack via simultaneous exfiltration of data packets on all available channels.

state, network nodes will be performing some sensing task and generating data samples. Reports of the gathered data are periodically sent back to a sink node or a base station, in a multi-hop fashion. Communication in this state is performed on the network's transmission channel, known as the common channel.

When a communication DoS attack commences, as shown in the attack scenario in the figure, communication will be disrupted in some part of the network. Nodes that lie within the transmission range of the attacker will be denied access to the common channel. These nodes, which we call *attacked* nodes, are presumably capable of detecting the presence of the attacker by applying some type of attack detection mechanism(s).

To overcome the affect of the attack, nodes inside of the attack region will immediately switch into an *exfiltration* state. The communication paradigm in this state is concurrent in the sense that attacked nodes will be transmitting simultaneously during *each* time-slot on *all* available channels. Nodes which lie just outside of the attack region, which we call *boundary* nodes, will also detect the presence of the attack. These boundary nodes must also participate in the exfiltration process by receiving exfiltrated data packets and forwarding them further in the direction of the base station. Hence, boundary nodes will be switching channels to receive packets coming from the inside of the attack region and back to the

common channel.

Attacked nodes will be transmitting all the time, while boundary nodes communicate with both attacked nodes and other regular network nodes. For commodity level sensor radios, the challenge is to schedule when nodes are sending and when nodes are receiving while, at the same time, attempting to utilize all available channels, so that maximum throughput out of the attacked region is achieved. To achieve this, attacked nodes and their boundary neighbors will be implementing deterministic scheduling supported by a *hybrid* MAC. A time-slotting mechanism is performed at the time of network deployment to allow for the TDMA implementation.

3.1.1 Hybrid MAC Scheduling

The underlying MAC paradigm in WSNs is based on CSMA (Carrier Sense Multiple Access) scheme where network nodes perform a carrier sensing step before transmitting and only transmit when the channel is sensed clear. CSMA is a good option for a sensor network's MAC layer; however, for the purpose of performing concurrent data exfiltration, the use of CSMA will fail since the increased contention will only lead to high collisions and repeated backoffs. In our data exfiltration framework, the MAC layer behaves in a CSMA mode during the normal state. When a communication DoS attack commences, the MAC layer switches into a deterministic TDMA (Time Division Multiple Access) mode to facilitate concurrent transmissions on the extra channels. Figure 3.2 shows the design of the data link layer in the exfiltration framework, which facilitates a hybrid MAC.

The concurrent channel assignment schedules need to be generated automatically once an attack is detected, independent of the attack topology. The distributed schedule creation is facilitated by *RMBS* scheduling, as shown in the figure. RMBS performs on top of TDMA for the purpose of schedule generation. The main objective of the RMBS approach is to act as a deterministic time division channel access to guarantee collision-free, concurrent transmissions out of an attacked region.

Therefore, in the data exfiltration framework, network nodes work on top of a CSMA layer and use TDMA schedules on the extra channels when needed. Attacked nodes will

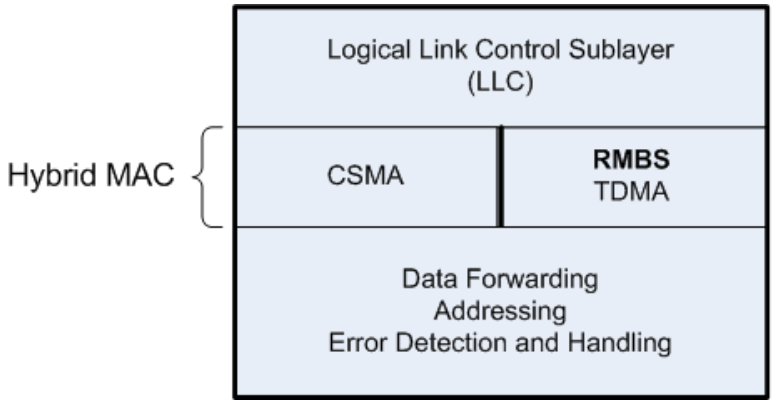


Figure 3.2: Data link layer in the data exfiltration framework.

be operating in a TDMA mode throughout the attack period. A boundary node, which is capable of communicating with both the attack region and other normal network nodes, needs to maintain connectivity to the network. Hence, boundary nodes will be switching back and forth between CSMA and TDMA modes at the MAC layer. This dual mode at the link layer which mixes CSMA and TDMA implies that careful scheduling and coordination is needed to guarantee successful transmissions.

3.2 Data Exfiltration Framework Architecture

The data exfiltration framework is reactive and is only triggered when an attack is present. The reactive nature implies that no extra overhead is incurred when the network is not under attack. Furthermore, the framework's functions extend from the application layer down to the data link layer to achieve the precise multiple channel communication. The result is an adaptive cross-layer design which will be explained next.

3.2.1 Architectural Diagram

Figure 3.3 shows the architectural diagram of the data exfiltration framework. It is important to note that the data exfiltration framework is presented as a defense mechanism and, therefore, only acts when the network's common communication channel is no longer

available. Additionally, the data exfiltration framework operates only at a subset of the network nodes. Therefore, in the exfiltration architecture, a network node is assumed to be operating in one of two internal states: *normal state* and *exfiltration state*, as depicted by the two dotted boxes in figure 3.3. It is also possible for a node (like in the case of a boundary node) to be working in a "meta state", i.e., combination of the two states.

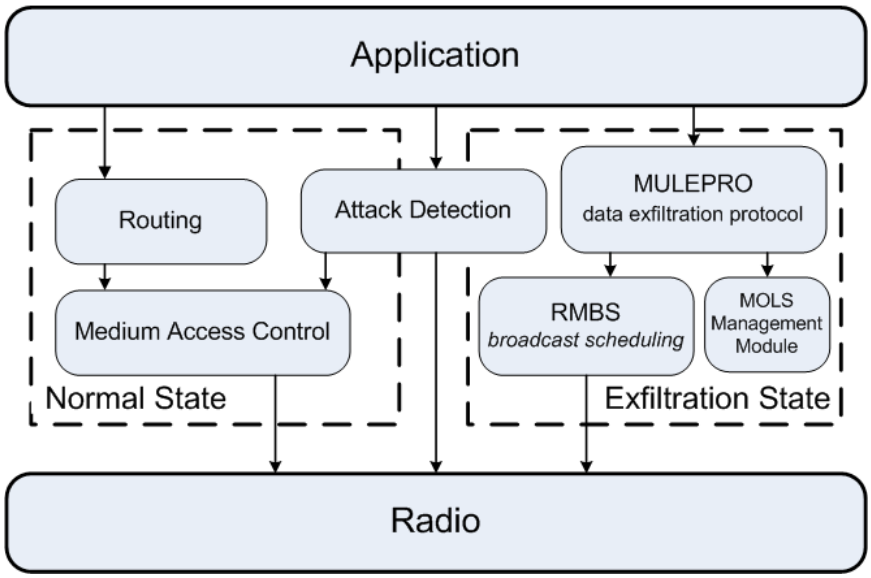


Figure 3.3: Architectural diagram of the data exfiltration framework. The arrows indicate a "uses" relation.

Initially, network nodes are operating in the normal state, represented by the left dotted box in the diagram. Under normal operation, each node performs its usual application specific tasks such as sensing the environment, generating data reports, relaying multi-hop packets and managing sleep cycles. The application layer in this state performs on top of the network's MAC layer which controls access to the communication channel.

Each node can independently determine that it is being subjected to a communication DoS attack through the services provided by the *attack detection* module. The attacker model (presented in section 3.2.2) defines an attack as an adversary controlling the communication channel either at the link or physical layer. This can be achieved by ignoring

proper MAC layer etiquette or by employing a radio emitter as a jammer that will interfere with successful physical layer decoding of legitimate packets. The differences between these two attacks suggest that we may perform detection either at the MAC layer or at the physical layer [14]. Therefore, the attack detection module shown in the diagram has a "uses" association with both the MAC layer and the radio component to allow for the direct monitoring of both. We note the separation of the attack detection module from the data exfiltration part of the system, as shown in the diagram. Such a separation in the design is necessary since the framework presented in this thesis is intended for attack mitigation and resilience which occurs after an attack has already been detected.

Based on receiving positive attack results, suffering nodes will immediately switch into the exfiltration state, represented by the right dotted box in the diagram. Once in this state, nodes will resort to the extra channels available on their radios to start the exfiltration task of their data packets. A switch of mode is also performed at the link layer to transition from the underlying CSMA into a multi-channel TDMA access method. The method we use for schedule generation on the multiple channels, during TDMA mode, is called *RMBS*. RMBS implements a form of TDMA broadcast scheduling and, therefore, acts at the link layer during the exfiltration state providing the higher layers access to the different channels.

The attack detection module is also capable of identifying nodes that are neighbors of an attacked node, but not actually part of the attacked region (i.e. boundary nodes). Boundary nodes will be switching their channel back and forth between the network's common channel (in a regular normal mode) and the other extra channels to listen to data coming from attacked neighbors (by coordinating their reception schedules accordingly).

Nodes operating in the exfiltration state will also have to decide what data to exfiltrate. Although this is somewhat application specific, we expect that the data will directly relate to the sensing task at hand. The data can come in the form of individual node reports or aggregated data, requiring in-network processing. Such techniques are evaluated by the *MULEPRO* protocol.

MULEPRO is an application level data exfiltration protocol functioning at the top level

of the exfiltration stack, as depicted in the diagram. It constitutes a fully distributed network based protocol and is designed to rapidly and concurrently exfiltrate sensor data. MULEPRO performs time-slotted transmissions to guarantee collision free broadcasting in the face of unavailability of service in the common communication channel. We note here that in the architectural diagram both MULEPRO and RMBS are closely coupled. This is due to the fact that the MULEPRO protocol is actually implementing RMBS scheduling as the underlying channel access mechanism during exfiltration state. Both RMBS scheduling and the MULEPRO protocol are designed and evaluated in this thesis and presented in chapters 4 and 5, respectively. Performance analysis results are presented in chapter 6.

The MULEPRO protocol extends its scheduling scheme to perform further *coordinated* scheduling via the aid of the *mutually orthogonal Latin squares (MOLS)* management module, as will be explained in the MULEPRO chapter. In the diagram, a "uses" relation exists between MULEPRO and the MOLS management module representing the fact that MULEPRO is using the coordination schedules created by the MOLS management module.

Other network nodes which are not participating in the exfiltration process will continue operating in the normal state and will be using the common channel to perform communication. When the attack stops, the affected nodes and their boundary neighbors will recover and go back to normal operation state.

Next, we highlight the design principles which are necessary for developing our data exfiltration architecture.

3.2.2 Design Principles

Part of this thesis is devoted to the design of sound defense protocols based on well-defined principles. We define the following design principles.

3.2.2.1 Adaptive cross-layer protocol design

It is important that the protocols at each layer not be developed in isolation, but rather within an integrated and hierarchal framework to take advantage of the interdependencies

between them. In such a cross-layer design, the link layer, which is capable of measuring link connectivity quickly and accurately, can trigger a higher level of the network stack to respond when a communication DoS attack is present. The application layer will need to adapt to the underlying network conditions by switching into the exfiltration state to deliver the highest possible amount of exfiltrated data packets. The application layer will be implementing the concurrent exfiltration of sensor data on the system's extra channels for the duration of the attack. When the attack ends, the application layer, triggered by the negative attack results from the link layer, will go back to its normal state of operation.

3.2.2.2 On-Demand Channel Allocation

Efficient channel allocation is at the heart of the design of an efficient data exfiltration system. The finite number of channels should be efficiently allocated to maximize exfiltration throughput and avoid co-channel interference. We define this design principle to guarantee automatic generation of the concurrent channel assignment schedules once an attack is detected, independent of the attack topology. Furthermore, appropriate coordination between transmission and reception tasks of senders and receivers needs to be incorporated in the channel allocation process.

3.2.2.3 Attack Detection and Mitigation: Separation of Duties

Defenses against radio interference attacks consist of first identifying that such an attack is underway then implementing mitigation solutions. The protocols in this thesis address attack mitigation and resilience which occurs after an attack has been detected. We state this design principle to separate attack detection from the actual working of the data exfiltration framework. Such a separation in the design is represented by the inclusion of the *attack detection module* as a separate entity in the exfiltration system's architecture, in figure 3.3.

The principles identified above are intended to be incorporated in the design of our reactive schemes presented in the followings chapters. Next, we define the architectural assumptions and abstraction models we consider in this research.

3.2.3 Architectural Assumptions

The network is assumed to operate in a multi-hop fashion by routing packets over multiple wireless hops. There are several kinds of multi-hop networks depending on the application for which the network is designed. Examples include ad hoc networks, mesh networks, and sensor networks. Our concentration is on the WSNs environment. We therefore assume a static network that operates on a single common channel for regular communication and the multi-channel capability is utilized only when the reactive defense is triggered. With this background, we now outline the basic abstraction models that we consider throughout this thesis.

3.2.3.1 Network and Interference Model

A wireless sensor network can be modeled as an undirected graph G = {N,E}. N is the set of network nodes, each mounted with an omni-directional radio transceiver and assigned a unique ID number. $E \subseteq N \times N$ is the set of links between nodes. A link $(m, n) \in E$ is assumed to exist only if m and n are within transmission range of each other. All network nodes have the same transmission range equal to the network's radio range, R_{tx}. Furthermore, all links are symmetric, hence $(n, m) \in E$ must also hold in this case. A link existing between two nodes also indicates that the two nodes are *one-hop neighbors*. Two distinct nodes having a common one-hop neighbor are called *two-hop neighbors*. As for transmission interference, we assume that each node n_k has an interference range such that any node n_j will be interfered with by the signal from n_k if $|| n_k - n_j || <= R_{tx}$, where the norm used is the Euclidean norm. In other words, a transmission from node n_j is viewed successful if $|| n_k - n_j || > R_{tx}$ for every node n_k transmitting during the same time-slot using the same channel.

The radio on each node is equipped with a single radio *half-duplex transceiver* that facilitates multiple communication channels. A half-duplex transceiver can only transmit or listen on one channel at a time, in either, but not both, send or receive mode. The practical impact of this hardware feature is that multichannel protocols must be carefully

designed to efficiently coordinate a node's *sending* and *receiving* assignments. We use Γ to represent the number of extra channels (besides the common channel). A node performs a channel switch by tuning its radio to the appropriate frequency range. We assume that the channel switching overhead is negligible since rapid channel switching will become feasible with the availability of improved hardware [82].

All communication channels of the system are assumed to be error-free and reception failure is due only to packet collisions with no overlap between the different channels. Time synchronization is assumed and is required for the successful implementation of the time-slotted scheduling. Any one of a number of proposed energy-aware time synchronization protocols can be used, including [83–85]. Each time axis of each channel is divided into transmission *time-slots* and is synchronous with other channels. When performing the reactive broadcasting, the time aligned on all the channels are grouped into a *multi-channel time-frame*.

Finally, network connectivity is assumed to always be maintained. In other words, at least one channel will be available during each time-slot, and an attack will never result in a partitioned network.

3.2.3.2 Attacker Model

An attacker(s) is capable of attacking up to γ channels such that $\gamma \leq \Gamma$, where $\Gamma + 1$ is the number of available channels. This also means that an adversary is assumed to disrupt some, but not all, communication channels. We note that an attack strategy that attacks a static subset of γ channels is easily thwarted by finding an open channel and simply having the nodes communicate on that channel.

An adversary node can be a node with similar transmission power to other network nodes, or it can be a more powerful node consisting of a more powerful transmission capability and/or multiple transmitters. Therefore, a single attacker is capable of affecting a multi-hop region and may be attacking multiple channels at the same time. In addition, we also expect that attackers may employ a channel changing strategy. In other words, the attackers will either try to find the channels the network nodes are currently using and

attack those channels, or simply attack channels according to a pseudo-random sequence. However, we do not assume a fast-hopping attacker capable of destroying a whole packet by simply flipping one or a few bits of the packet. This is a reasonable assumption since a resistance to such an attacker can be easily achieved by encoding packets using appropriate error-correcting codes as explained in [86].

We assume that the attackers can launch several types of radio interference attacks against the network. In [67], Xu, et al. describe four attacker profiles. A constant jammer continuously emits a radio signal in effect continuously sending out random bits to the channel. A deceptive jammer continuously injects entire regular packets to the channel without any gaps. A slight variant on the first two is a random jammer, which alternates between sleeping and jamming, for purposes of energy conservation. The fourth profile is a reactive jammer, which starts jamming when it senses activity on the channel. Law, et al. [87] describe attacks against specific MAC protocols that attempt to conserve the energy resources of the attacker. Our work assumes that any of these attacks are possible.

We assume that when an adversary disrupts a channel using any one of the above attacks, then, for the most part, that channel cannot be used by the system. Finally, we assume that the sensor node security system has not been compromised, and that existing link and network level security mechanisms for confidentiality and authentication are in place.

3.2.3.3 Attack Detection Model

For the purpose of attack detection, we assume that a network node is capable of detecting a communication a DoS attack either at the physical layer or the link layer. We, therefore, assume that the attack detection module is somehow incorporated in the network stack, and its functionality is available to serve the higher application layer. Current work in attack identification consists of algorithms for distinguishing between normal and abnormal noise in a channel using various types of thresholding approaches, correlating expected and actual packet delivery ratios, or detecting MAC protocol misbehavior [65,67,87]. We list here some

of the proposed techniques from the literature, and we state their effectiveness in identifying the different jammers introduced in the attacker model in the previous subsection.

- **Signal Strength Measurement.**

  In this type of checking, the average power of the received signal is used. This can be done either at the MAC or physical layer by simply comparing the signal strength with some predefined threshold value. An enhanced mechanism may adapt the threshold value based on the noise level on the channel. However, this simple statistical metric may fail to distinguish between normal network traffic (as in the case of congestion) and attack scenarios. Xu, et al. [16, 72] concluded that, even if a threshold value was chosen correctly, this technique would only be capable of detecting the constant and deceptive attackers and fails in identifying the random and reactive attackers.

- **Carrier Sensing Time Measurement.**

  Carrier sensing time is another statistical tool which can be used to detect an attack on the communication channel. The carrier sensing time's distribution is usually known for a specific network under the normal conditions. Such a statistical threshold can be determined either theoretically or empirically. Just as in the previous metric, the effectiveness of the carrier sensing metric is also limited to detecting constant and deceptive attackers only.

- **Packet Delivery Ratio (PDR) Measurement.**

  This metric monitors the ratio of successfully received packets by the total number of received packets. Thus, it allows for the detecting of an attack at the receiver side, by using CRC check results of all incoming packets. It was shown that even during the worst network congestion, the PDR measurement would not drop beyond a 78% value. However, when a radio interference attack is performed it drops to almost a 0. Furthermore, Xu, et al. in [16, 72] concluded that this single metric is capable of detecting all the differnt types of attacker models. However, PDR measurement also fails in distinguishing between attack scenarios and network failures scenarios (like a

battery failure) that can disrupt the communication between two nodes.

- **Consistency Checks.**

  This technique was proposed by the authors in [16] to overcome the shortcoming of PDR. In other words, this techniques was also proposed to allow for the detection of all types of radio interference attacks and overcome the problem of distinguishing between networks dynamics and interference attacks. We refer the reader to [16] for more information about this metric.

As stated earlier, the focus in our work is on mitigating the effect of an attack. Therefore, we assume that nodes possess the necessary functionality to detect that an attack is underway by using one or more of the preceding techniques.

This section presented the conceptual architecture of the data exfiltration framework, highlighted some design issues, and defined the underlying abstraction models. The problem of concurrent data exfiltration is abstracted in this thesis as a TDMA broadcast scheduling problem. The following section presents a brief overview of the general problem of TDMA broadcast scheduling to establish the underlying mechanism of our approach.

3.3 Data Exfiltration as a TDMA Broadcast Scheduling Problem

A TDMA broadcast scheduling algorithm is defined as an algorithm whose task is to produce and/or maintain an infinite transmission schedule such that all transmissions are guaranteed collision-free [88]. The main task in designing a TDMA schedule is to allocate transmission *time-slots* to the different network nodes, depending on the topology and the node packet generation rates.

Most broadcast scheduling algorithms operate by producing a finite length schedule, also known as a transmission *time-frame*. Each node is assigned to exactly one time-slot for its transmission during a time-frame. The scheduling task also aims at maximizing the number of interference-free transmissions assigned per a time-slot. A proper schedule not

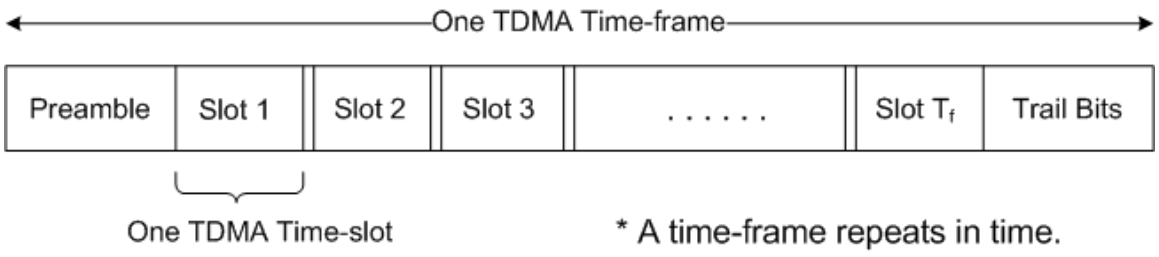


Figure 3.4: General time structure in TDMA systems.

only avoids collisions by silencing interfering nodes in each time-slot, but also minimizes the number of time-slots and hence the latency. Once a proper time-frame is created, most broadcast scheduling algorithms operate by infinitely repeating that schedule [88]. Figure 3.4 shows the general frame and time-slot structure in TDMA systems. Time on the transmission channel is divided into T_f time-slots. Hence, T_f represents the length of a transmission time-frame in time-slots.

The traditional transmission scheduling algorithms focus on single channel systems and attempt to optimize the time-slot allocation on one channel only. To extend this type of broadcast scheduling to the multi-channel scenario, a more sophisticated slot allocation is needed to allow for simultaneous assignments on all channels. In this case, the produced schedule (or time-frame) will be multi-dimensional [88]. We adopt this multi-channel TDMA broadcast scheduling approach when defining our concurrent data exfiltration problem; however, the problem's structure is altered to meet the single-transceiver special requirement (schedule creation is even more challenging for the case of the concurrent data exfiltration due to the single-transceiver environment). A formal model of the channel assignment problem for the purpose of concurrent data exfiltration is presented next. We note here that distributed scheduling access in ad hoc wireless networks, in general, is NP-hard. Heuristic methods are used for finding good feasible solutions [89].

3.3.1 Formulation of the RCAP problem

The Channel Assignment Problem (CAP) in radio networks is formally defined in the literature and is known to be NP-complete [90]. The main objective of the CAP problem is to result in an efficient assignment of the different nodes to the different radio channels such that the resultant solution satisfies the given constraints. Several heuristic approaches have been used to solve various channel assignment problems. The work in [91] proposed a mathematical programming approach in which the CAP was formulated as an integer linear programming (ILP) problem. The authors also presented a formal solution to the problem based on the *branch and bound* method.

We present here an extension to the approach found in [91] to apply the CAP problem's approach to the specific case of assigning channels to nodes inside of an attacked region for the purpose of reactive exfiltration. We use the term RCAP (Reactive CAP) to refer to the resultant formal model. Insights from the formal modeling will be used in designing more robust exfiltration solutions.

The problem's main parameters are:

1. **The Compatibility Matrix, C.**

   In general, the topology of an ad hoc network with N nodes can be described by an $N \times N$ symmetric binary matrix which we call *connectivity matrix CN* and is defined as:

$$\mathrm{CN}_{ij} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are one-hop neighbors} \\ 0, & \text{otherwise} \end{cases}$$

   This CN matrix also defines the *compatibility matrix C* which is defined as the $N \times N$ binary matrix that keeps track of all possible interference between nodes (refer back to section 3.2.3.1 for the network's interference model). The C matrix is symmetric since we assume symmetric links. The values in this matrix are considered constant with time since we assume a static network model. Furthermore, it defines the input

parameter to our scheduling algorithm.

2. **Exfiltration Matrix, EX.**

   The broadcast scheduling algorithm will be generating a multi-channel, conflict-free, and constrains-satisfied TDMA frame to be repeated over time. This TDMA frame is represented by the *exfiltration matrix EX* in our problem formulation, and is modeled as the $J \times \Gamma$ binary matrix (J attacked nodes and Γ extra channels) defined as the following:

$$Ex_{ij}^{t} = \begin{cases} 1, & \text{if channel } i \text{ is assigned to node } j \\ & \text{during the current time-slot } t \\ 0, & \text{otherwise} \end{cases}$$

   Therefore, a matrix element, $\mathbf{Ex}^{t}{}_{ij}$ decides the transmission of node j on channel i during time-slot t. This matrix contains the concurrent channel assignments and is dynamically created when a node switches into exfiltration mode.

3. **The Channel Assignment Vector, CA.**

   The channel assignment for a node during exfiltration phase is described by a T_f-element vector, called *Channel Assignment vector CA* and is defined as:

$$CA = [ca_1, ca_2, ca_3, ..., ca_{T_f}]$$

   The elements of this vector have a value assigned in the range [1..Γ]. Each element, $CA(\mathrm{j}) = ca_j$, in CA represents the channel number on which a node is assigned to during the jth time-slot. The values in this vector are directly derived from the **EX** matrix.

Thus, in the resultant formulation, matrix C represents the input parameter and matrix EX is the output. The objective from the solution technique would be the Exfiltration Matrix EX which *maximizes* the number of channels being used simultaneously, subject to the following constraints and requirements:

- The interference *constraint* ensures that any simultaneous transmissions, on some channel, should not interfere. This can be guaranteed by preventing one-hop neighbors from transmitting simultaneously on the same channel and can be written as follows:

$$Ex^t{}_{ij} + Ex^t{}_{i'j'} <= 1, \quad \forall \ \ (i,j),(i',j') \in I \tag{3.1}$$

where I represents the set of incompatible transmission assignments:

$$I = \{(i,j),(i',j') \mid \ \ c_{jj'} \ != \ 0, \ \ (i,j) \ != \ (i',j')\}$$

where $c_{jj'}$ is the *C(j,j')* element of the Compatibility matrix C.

- The single interface *constraint* ensures that during a time-slot a node is assigned to either transmit or receive on *one* single channel only. This can be written as follows:

$$Ex^t{}_{ij} + Ex^t{}_{i'j} <= 1, \ \ \forall \ \ i' \ \ in \ \ [1-\Gamma], \ \ i \ != \ \ i' \tag{3.2}$$

- It should be ensured that all nodes are allocated at least one time-slot in every time-frame, which we define here as a minimum-rate *requirement*. This can be written as follows:

$$\sum_{t=1}^{T_f} Ex_{i_j} >= 1 \quad \forall \ \ i,j \tag{3.3}$$

The resultant formulation for the channel assignment problem for the purpose of concurrent data exfiltration is summarized next.

Problem(**RCAP**):

Objective function:

$$maximize \quad \sum_{t=1}^{T_f} \sum_{j=1}^{J} \sum_{i=1}^{\Gamma} Ex_{ij}^{t} \tag{3.4}$$

subject to:

1. the **channel-interference** *constraint*, given in 3.1
2. the **single-interface** *constraint*, given in 3.2
3. the **minimum-rate** *requirement* given in 3.3

3.3.2 Formal Solution Strategy

The authors in [91] proposed a branch and bound solution to the CAP problem and showed its effectiveness in reducing the complexity of the assignment problem. In the above formulation, all the Ex_{ij} variables are binary, which means that the RCAP model can also be represented as an integer linear programming problem that can be solved using a branch and bound method. We follow the same approach presented in [91], and we present a formal representation of our solution technique next.

The basic concept of the branch and bound method is to partition the given problem into a number of intermediate smaller problems. In the case of ILP problems, a relaxed version of the initial problem is solved in an initial step to provide a *bound* to the resultant subproblems, which will be solved in the second step. Each subproblem will then be characterized and solved by the inclusion of one or more of the ILP model's constraint(s), and this provides the basis of the *branching* step. The decomposition process is repeated at each level of the generated subproblems, creating a search solution tree. The process continues until each unexamined subproblem is decomposed, solved, or shown not to lead to an optimal solution to the original problem. Ideally, the procedure stops when all nodes of the search tree are either pruned or solved.

Figure 3.5 presents an example of the channel assignment problem under the RCAP

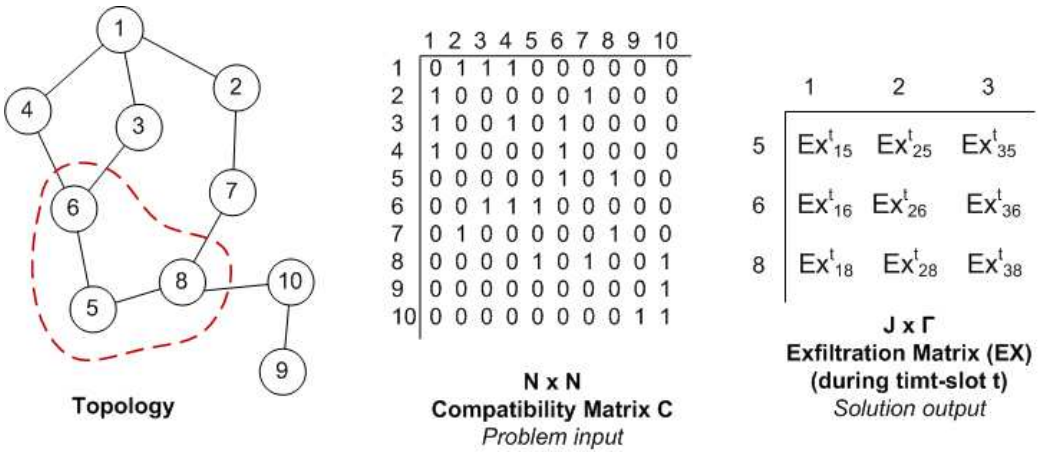


Figure 3.5: The channel assignment problem under the RCAP model. The instance of **EX** shown here includes the multi-channel assignments during a *single* time-slot.

model. Nodes 5,6 and 8's transmissions on the common channel are altered by the presence of an attacker, and, therefore, nodes 5,6 and 8 will need to perform their transmissions on the extra channels. The two parameters identified in the RCAP problem, namely, the compatibility matrix **C** as an input and the exfiltration matrix **EX** as an output, are also shown in the diagram. In the shown scenario, the number of extra system channels Γ is equal to 3. c_{56}=1, c_{58}=1, and c_{68}=0 where $c_{jj'}$ is the C(j,j') element of the compatibility matrix C. Equations (3.1) and (3.2) will result in the following constraints:

1) Channel-interference Constraints:

$$Ex^t_{15} + Ex^t_{16} \leq 1$$
$$Ex^t_{15} + Ex^t_{18} \leq 1$$
$$Ex^t_{25} + Ex^t_{26} \leq 1$$
$$Ex^t_{25} + Ex^t_{28} \leq 1$$
$$Ex^t_{35} + Ex^t_{36} \leq 1$$
$$Ex^t_{35} + Ex^t_{38} \leq 1$$

2) Single-interface Constraints:

$$Ex^t_{15} + Ex^t_{25} + Ex^t_{35} \leq 1$$
$$Ex^t_{16} + Ex^t_{26} + Ex^t_{36} \leq 1$$
$$Ex^t_{18} + Ex^t_{28} + Ex^t_{38} \leq 1$$

In addition, by enforcing the minimum rate requirement, the logical operator in the second set of equations will be even more restrictive.

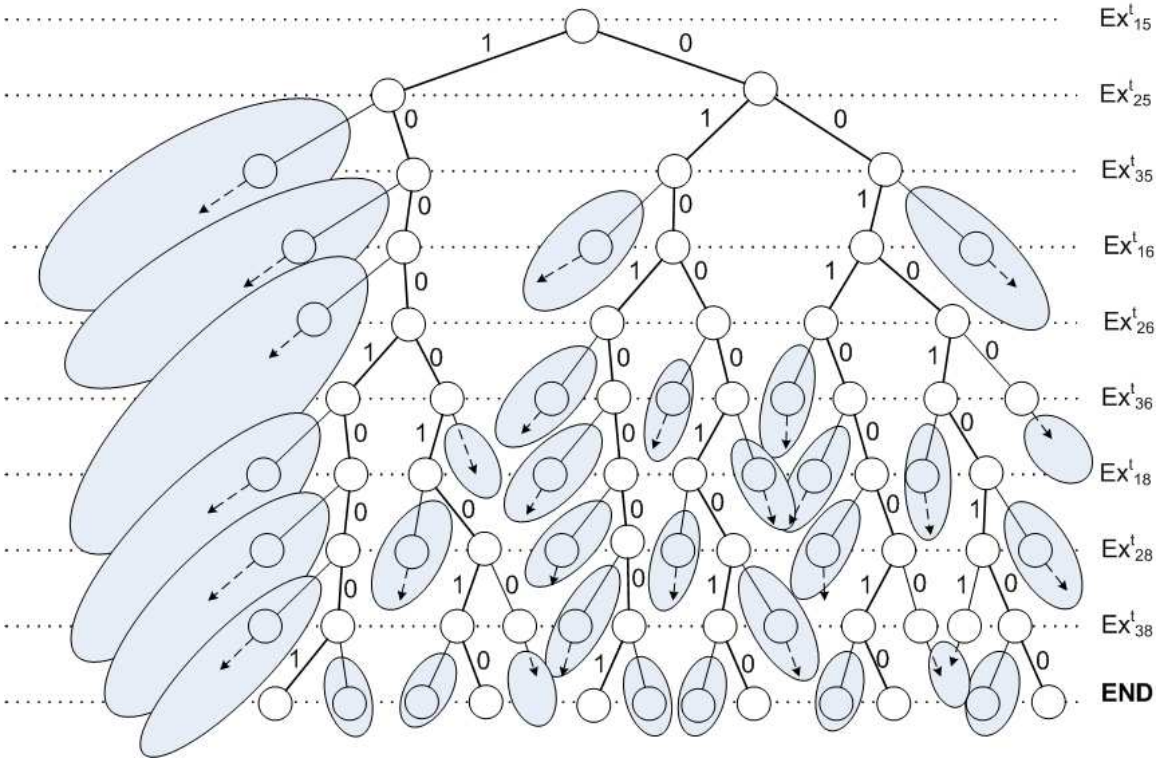


Figure 3.6: A branch and bound search solution tree. The shaded areas in the figure include the nodes that will not be considered because of the constraints and the enforcement of the minimum rate requirement.

A branch and bound solution starts by assuming that all the Ex_{ij} are continuous to get a lower bound (since maximization) for the value of the objective function. The branching procedure can then work by fixing the possible values of the binary variables as zero or one. In each level the value of one variable is fixed and the generated subproblems are solved. Ineffective branches, which lead to violations of the generated set of constraints (incompatible assignments), are instantly pruned. The branching procedure and the effect of the compatibility constraints given by 3.1 and 3.2 are shown in figure 3.6. For this example, the order of the branching variables is taken to be:

$$Ex_{15} \Rightarrow level1, Ex_{25} \Rightarrow level2, Ex_{35} \Rightarrow level3,$$
$$Ex_{16} \Rightarrow level4, Ex_{26} \Rightarrow level5, Ex_{36} \Rightarrow level6,$$
$$Ex_{18} \Rightarrow level7, Ex_{28} \Rightarrow level8, Ex_{38} \Rightarrow level9.$$

Such a branch and bound solution to the CAP problem was shown to be very effective in reducing the complexity of the assignment problem [91]. The pruning of the shaded

branches will lead to a significant reduction in the size of the problem space. In the shown example, the total number of tree nodes up to the level 4 is 28; only 14 of those needed to be examined by the branching procedure. To reach a final solution (9 levels), the total number of tree nodes is 1023; only 50 of those needed to be examined by the branching procedure. The use of the solution bounds at each level will further simplify the search tree. The authors in [91] concluded that the exploitation of the problem's special structure can improve the computational efficiency of the branch and bound method. The authors also stated that an effective heuristic method for improving the solution performance would be to assign a branching priority to as many binary variables (Ex_{ij} in this case) as possible and that the branching priorities should be assigned based on the knowledge of the problems special structure [92].

3.4 Heuristic Broadcast Scheduling based on Latin Squares

The proposed revised branch and bound algorithm in our work is based on the use of Latin squares assignments as a heuristic method for the branching procedure by which a priori precedence can be assigned to the Ex_{ij} branching variable. As we explained in section 2.2.2, When Latin squares are used for scheduling, the rows of the Latin square correspond to channels and the columns correspond to time-slots. Each node generates its unique transmission schedule based on a symbol's pattern in a common Latin square shared by the different competing nodes. Figure 3.7 shows an example.

The appearance of a square symbol in the current time-slot for channel i will assign a transmission priority for all nodes assigned that symbol to perform their transmissions on that channel during the current slot. In other words, the appearance of a symbol in the current transmission slot will assign a value 1 to the $Ex_i^t_j$ binary variable and a value 0 otherwise. This will result in further pruning of unnecessary tree branches and a much smaller search space, as shown in figure 3.8. In the shown example problem, only 10 nodes needed to be examined by the branching procedure before reaching the solution level, as opposed to 28 nodes when pure branch and bound was used.

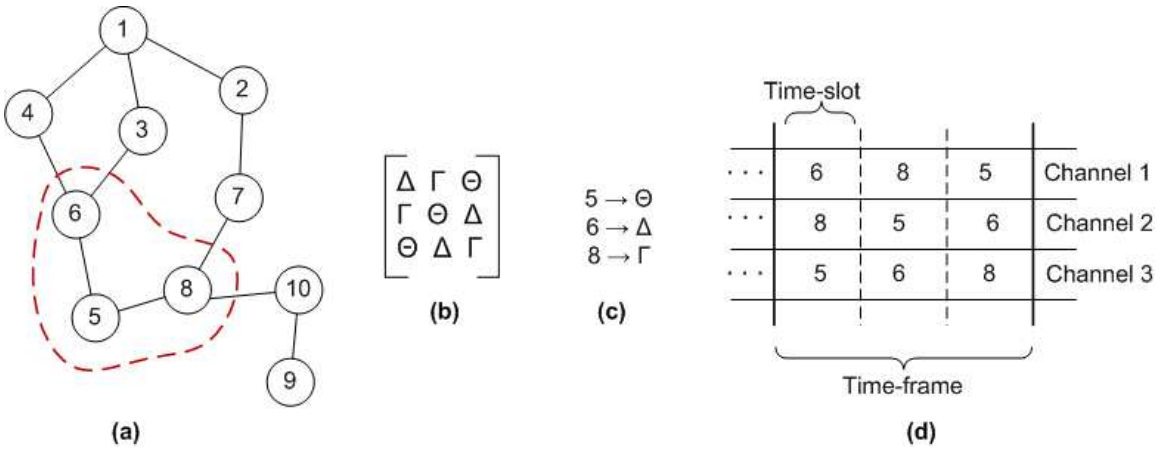


Figure 3.7: Heuristic broadcast scheduling based on Latin squares. **a)** Nodes 5,6, and 8 are performing multi-channel communication. **b)** The Latin square used for scheduling. **c)** Mapping nodes into square symbols. **d)** Scheduling *concurrent* transmissions based on square symbols on the multiple channels.

Furthermore, by the properties of a Latin square, both the channel-interference and the single-interface constraints are preserved. In fact, the use of Latin square assignments reduces the number of constraints, which works more efficiently when handling ILP problems with many constraints and less variables [93]. According to [93], the most important factor affecting computations in ILP is the number of integer variables and the feasible range in which they apply. They conclude that a reduction in the number of constraints will lead to a reduction in the feasible range.

We emphasize the use of the Latin square assignments here as a heuristic approach to enhance the performance of the branch and bound solution. However, its effectiveness as a formal solution is not evaluated as part of this dissertation work since our interest is more about the protocol engineering aspect. In the next chapter, we transition toward the design and development of what we call *reactive multi-channel broadcast scheduling (RMBS)* for the purpose of immediate data exfiltration. The RMBS approach implements a *distributed mapping function* which assigns nodes to transmission time-slots on the extra channels based on their interference information. Latin square assignments constitute the mapping function such that the different nodes are assigned transmission time-slots on the different channels by picking a square symbol and following its pattern. We conclude this

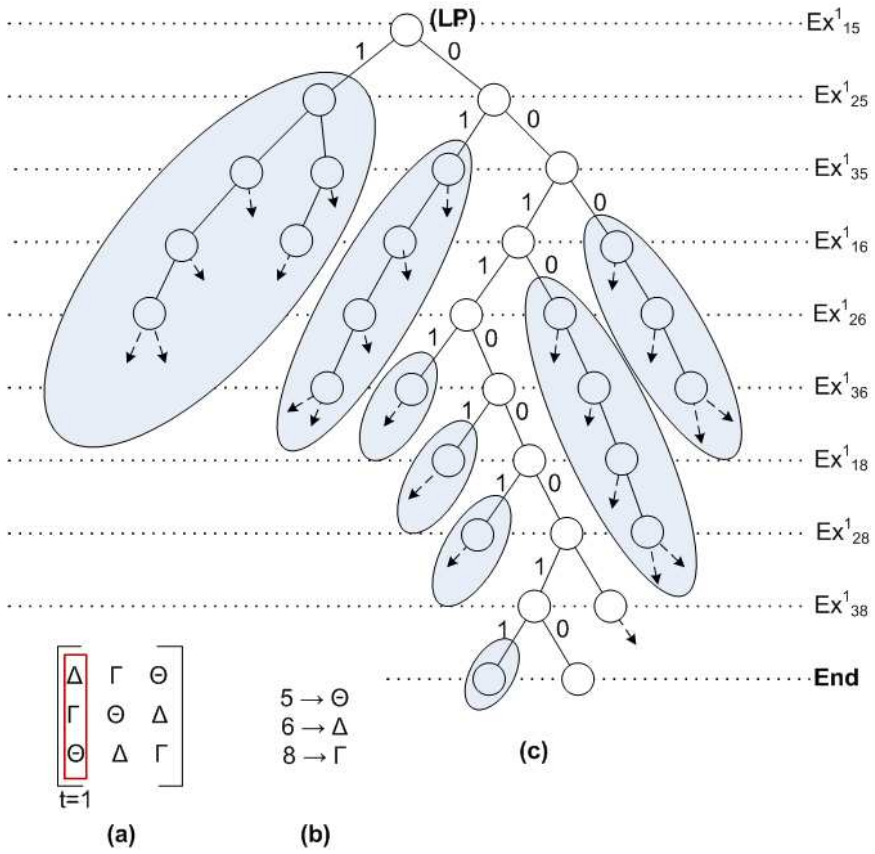


Figure 3.8: A branch and bound search solution tree based on Latin square assignments. a) Latin square used in scheduling. b) mapping of nodes and square symbols. c) the example shows the channel assignment during the *first* time-slot of the time-frame, represented by the first column of the Latin square.

chapter by providing some important definitions and some preliminary background useful in the development of our scheduling algorithms in the next chapter.

3.4.1 Latin Squares

Definition. *A $p \times q$ rectangular array formed by the symbols $1, 2, ..., k$, where $k \geq p$ and $k \geq q$ is called a* **Latin rectangle** *if every symbol from the symbol set appears at most once in each column and once in each row [63].*

Definition. *A* **Latin square** *A of* **order** *p is a $p \times p$ matrix with entries from a set of p distinct symbols such that each row and column contains every element exactly once. The* **square symbol** *in the ith row and the jth column is written as $(a_{i,j})$.*

The square matrices A and B shown below are examples of Latin squares of order 5.

$$A = \begin{bmatrix} 2\ 3\ 4\ 0\ 1 \\ 3\ 4\ 0\ 1\ 2 \\ 4\ 0\ 1\ 2\ 3 \\ 0\ 1\ 2\ 3\ 4 \\ 1\ 2\ 3\ 4\ 0 \end{bmatrix} \qquad B = \begin{bmatrix} 3\ 0\ 2\ 4\ 1 \\ 4\ 1\ 3\ 0\ 2 \\ 0\ 2\ 4\ 1\ 3 \\ 1\ 3\ 0\ 2\ 4 \\ 2\ 4\ 1\ 3\ 0 \end{bmatrix}$$

Lemma 1. *If two nodes are assigned two distinct symbols in a common Latin square, these two nodes will be collision-free with each other all the time.*

The proof of lemma 1 can be found in [64].

An important parameter in such scheduling is the p value which represents the order (or dimension) of the square. P network nodes share the same Latin square, and the time-slot assignment pattern of each of these nodes is represented by one of the distinct symbols in the Latin square. To allow the transmission assignments of more than p nodes, we can assign nodes to use a different Latin square given that the two squares are *mutually orthogonal* with respect to each other. We now introduce *mutually orthogonal Latin square (MOLS)* and explain their role in multi-channel schedule creation, as presented in [63].

3.4.2 Mutually Orthogonal Latin Squares (MOLS)

Definition. *Two distinct $p \times p$ Latin squares A and B, where $(a_{i,j})$ and $(b_{i,j}) \in \{1,2,...,p\}$, are said to be* **orthogonal** *if the p^2 ordered pairs $\langle..,(a_{i,j},b_{i,j}),..\rangle$ are all different.*

For example, combining the two Latin squares, square A and square B (shown above), will result in the following square matrix:

$$AB = \begin{bmatrix} (2,3)\ (3,0)\ (4,2)\ (0,4)\ (1,1) \\ (3,4)\ (4,1)\ (0,3)\ (1,0)\ (2,2) \\ (4,0)\ (0,2)\ (1,4)\ (2,1)\ (3,3) \\ (0,1)\ (1,3)\ (2,0)\ (3,2)(4,4) \\ (1,2)\ (2,4)\ (3,1)\ (4,3)\ (0,0) \end{bmatrix}$$

This resultant matrix introduces a very important property: the two original squares (A and B) are now said to be *orthogonal* since all pairs which appear in the AB matrix are

unique. Orthogonality of Latin squares corresponds to there being exactly one time/channel collision for every pair of nodes assigned to symbols from the two squares.

Lemma 2. *If two nodes are assigned two symbols from two different orthogonal Latin squares, then there is at most one collision for these two nodes in every time frame.*

The proof of lemma 2 is given in [64].

In general, if σ is a set of distinct Latin squares of order p, the squares in σ are said to form a *mutually orthogonal family of Latin squares (MOLS)* if every pair in the set σ is orthogonal. In addition, a family of MOLS of order p is said to be *complete* if it consists of $(p-1)$ squares [63].

Lemma 3. *If there is an orthogonal family of σ Latin squares of order p, then $\sigma \leq p-1$.*

Definition. *An orthogonal family of σ Latin squares of order p is called* **complete** *if σ is equal to p-1.*

Lemma 4. *If $p > 1$ and $p = i^m$, where i is a prime number and m is a positive integer, then there is a complete orthogonal family of Latin squares of order p.*

Proof. The proofs of lemmas 3 and 4 can be found in [63]. □

To summarize, when MOLS assignments are used for scheduling, if two nodes obtain different symbols from the same square, their transmission slots will never collide at any time. If two nodes obtain symbols from different squares, their transmission slots will have at most one collision during each frame [63].

Chapter 4: Reactive Multi-Channel Broadcast Scheduling

The previous chapter presented a high level overview of the data exfiltration framework and presented the RCAP problem model. In this chapter, we describe *reactive multi-channel broadcast scheduling (RMBS)*, a novel mechanism to provide transmission survivability under the adversarial model presented in this dissertation (subsection 3.2.3.3). We define different modes of operations for RMBS to incorporate several implementation options ranging from management-free scheduling to precise coordinated scheduling. In its most basic form, RMBS assumes no knowledge about the network topology and ,therefore, will be implementing topology-transparent broadcasting based on Latin squares. For topology-aware broadcasting, we adopt a combined approach in which we optimize the Latin square scheduling by incorporating local neighborhood information in its design.

We start in section 4.1 by providing a high level overview of the RMBS scheduling approach, and we outline its different modes of operation. We then present the implementation details of each of the four different modes in sections 4.2 through 4.5. We conclude by discussing the anticipated advantages from the RMBS scheduling approach.

4.1 RMBS Overview

RMBS provides a time-slotted channel access mechanism for nodes inside of an attacked region to perform concurrent transmissions on the extra channels. It uses Latin square assignments to produce *reactive* transmission schedules. The properties of Latin square assignments present attractive features when applying it to scheduling problems, as discussed in section 2.1.3. RMBS defines four modes of operations to provide support for several implementation options ranging from management-free scheduling to precise coordinated scheduling. The basic assumption in all of the different modes is that the network's degree

$\triangle$ and the number of the extra channels Γ are all assumed to be known beforehand. Below are some basic definitions.

Definition. *The number of neighbors of a vertex v is called the* **degree** *of v. The* **maximum degree** *vertex in a graph G defines the graph degree $\triangle(G) = \triangle$.*

Assumption. *The network is communicating on a single* **common channel** *and each node has a* **half-duplex single-transceiver** *radio that facilitates Γ extra channels, in addition to the common channel.*

Based on these two parameters ($\triangle$ and Γ), the RMBS scheduling algorithm will guarantee a pre-determined minimum throughput for each of the attacked nodes, independent of the underlying topology of the attacked region. Furthermore, the RMBS scheduling algorithm defines different modes that tradeoff protocol overhead with the exfiltration performance. We now provide an overview of the different modes of operations.

In its most basic form (mode 0), RMBS provides a topology-transparent broadcasting. The basic idea in this mode is to assign network nodes to different square/symbol combinations from a set of mutually orthogonal Latin squares to produce conflict reduced transmission schedules. The major advantage of this approach is the reduction or elimination of control message overhead for schedule creation and management. However, such an approach is probabilistic in the sense that it does not guarantee collision-freedom. If two one-hop neighbors were randomly assigned the same square/symbol combination, then their concurrent transmissions will be colliding all the time. In order to address this possibility, we design the second mode of operation, which we call **L-VC**.

L-VC mode combines **L**atin square scheduling with a **V**ertex-**C**oloring algorithm to achieve a more robust exfiltration design. The vertex-coloring step captures the direct interference information between the network nodes and guarantees a unique vertex-color assignment for all interfering nodes. Each node can then use its assigned vertex-color directly as an entry in the Latin square (rather than random assignment of square symbols as in the topology-transparent mode), resulting in a much more efficient mapping between nodes and square symbols. This combined approach can now guarantee no collisions will

occur between any two neighboring nodes transmitting concurrently because the two nodes will always be assigned to different channels. Another advantage in this approach is that the scheduling complexity is greatly reduced since the scheduling algorithm is now generating schedules for k vertex-colors rather than N network nodes. We will show in the L-VC mode section that this reduction in the problem's space will allow for the use of only one *single* Latin square for schedule creation. The use of a single square for schedule provides guarantees *conflict-free* scheduling, as will be explained later in this chapter.

Although mode L-VC guarantees collision freedom between neighboring nodes, it does not prevent collisions from occurring at the sender which makes this mode vulnerable to the hidden terminal problem (section 2.1.2). To provide RMBS robustness against the hidden terminal problem, we consider the 2-hop distance when capturing interference. As a result, the number of interfering nodes will increase dramatically and, therefore, a single common Latin square approach is no longer possible. Instead, we use **Mu**tually orthogonal Latin squares combining their assignments with a distance-2 **V**ertex-**C**oloring step. Such a combination constitutes the **MuVC** mode.

Finally, in some attack scenarios, some attacked nodes must also cooperate in receiving data from inside the attacked region and broadcasting it in the direction of a boundary node. Such cooperating nodes must therefore coordinate their listening duties along with their exfiltration tasks. For this purpose, we design the fourth mode of operation (**MuVC-S/R**) which performs *coordinated* scheduling, as opposed to just broadcast scheduling. To do this, we redesign the underlying physical TDMA time-frame structure to incorporate the coordinated assignments. We introduce the notion of an **exfiltration time-frame** in which a logical TDMA frame structure is constructed where we further divide each TDMA time-slot into two logical mini-slots. The number of mini-slots is chosen to allow for both transmission and reception.

4.1.1 Initialization and Reaction

During network setup phase, RMBS runs the following operations in sequence: neighbor discovery, distributed vertex-coloring (if needed), Latin square construction and global time

synchronization. These operations run during the setup phase and need not run again unless a significant change in the network topology (such as physical relocation of sensors) occurs. The idea is that the initial upfront costs for running these operations are compensated by the immediate reaction when an attack commences. The details of steps 1-3 stated above will be explained when we go through the specifics of each of the different modes.

RMBS can be combined with any typical routing protocol. When no attack is occurring, broadcast and multicast packets are transmitted using the network's common channel. The reaction step is only triggered when an attack is detected and is only implemented by the subset of the network nodes which are affected. Therefore, we divide the actual implementation of RMBS scheduling into two parts (the different parameters are briefly defined in table 4.1):

Initialization Step	1) Perform distributed vertex-coloring using interference information of the compatibility matrix **C** (if needed). 2) Based on the value of ($\triangle$ or k), compute a value for T_f. 3) Based on the values of (N or k), T_f and Γ, construct the Latin square family.
Reaction Step	1) Create the $J \times \Gamma$ exfiltration matrix **EX** based on the Latin square assignments. 2) Use the channel assignment vector $\mathbf{CA}(j)$ to transmit accordingly on the multiple channels, while in exfiltration state.

The specifics of the different steps depend on the specific mode of operation. Again, different modes will be presented which tradeoff protocol overhead with the exfiltration performance. Note how in the mode 0, for instance, the first step (1) of initialization is not performed. This is because this mode is providing topology-transparent broadcasting and hence the vertex-coloring algorithm is not implemented. The protocol overhead is minimized in this case.

The steps, as stated above, provide a very high-level overview of the RMBS scheduling approach. We discuss the details of both steps with respect to each of the different modes of operations in the upcoming sections of this chapter.

Table 4.1: RMBS design parameters.

Parameter	Description	Reference
N	number of nodes in the network	subsection 3.2.3.1
k	number of produced vertex-colors	subsection 4.3.1
$\triangle$	maximum nodal degree of the network	section 4.1
Γ	number of extra communication channels available	section 4.1
T_f	time-frame length in transmission time-slots, during exfiltration state.	section 3.3
C	the N×N matrix which keeps track of all primary interference between the different network nodes. This matrix can be set to also capture secondary interference, depending on the RMBS mode.	subsection 3.3.1
EX	the J×Γ matrix contains the concurrent channel assignments in an exfiltration state.	subsection 3.3.1
CA(j)	the T_f vector whose values are in the range [1..Γ] and hence maintains the channel assignments for node j during a time-frame.	subsection 3.3.1

4.1.2 A Running Example

Figure 4.1 defines a simple attack scenario as a running example used which will be used in this chapter when discussing the different RMBS modes. The figure shows a 15 nodes network topology with the base station located at node A. The input parameters, shown on the diagram, include the number of extra channels (Γ) as 5 and the maximum nodal degree of the network ($\triangle$) which is equal to 4. In the shaded region, an adversary is capturing the common channel for some period, leading to a communication DoS attack. Three nodes, E, G and F, are the attacked nodes and five nodes, C, D, H, L and K, are boundary nodes. As a result, nodes E, G and F will switch their transmission mode at the MAC layer from the non-deterministic CSMA to the deterministic TDMA scheduling and will remain in the TDMA mode throughout the attack period. Boundary nodes will be working in a dual mode, continuing to use CSMA to communicate with the rest of the network on the common channel while switching into TDMA mode when listening to transmissions on the extra channels coming from inside of the attack region.

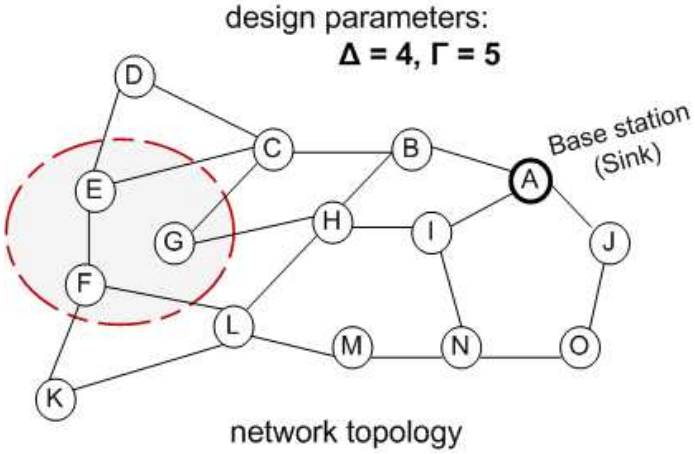


Figure 4.1: The chapter's running example. The shaded region shows the nodes undergoing a communication DoS attack.

Next, we explain the working details of RMBS under the different modes.

4.2 Mode 0: Topology Transparent Broadcasting

In this mode, nodes perform their data exfiltration without any assumptions about the underlying topology and no guarantee on the collision-freedom of the concurrent transmissions. MOLS assignments (section 3.4.2) are used for the schedule creation. We base our approach in this mode on the work presented in [63].

4.2.1 Initialization

During the initialization phase of RMBS under this mode, nodes generate the $p \times p$ MOLS family based on the given parameters $\triangle$ and N. Every node will share a Latin square for schedule creation. To guarantee a unique assignment for each of the N network nodes, the following must hold:

$$p \times \sigma \geq N \tag{4.1}$$

Where σ defines the size of the square family and p defines its dimension. We use lemma 4 to determine the value of σ then choose the minimum p value that would satisfy equation 4.1. In the running example, the number of network nodes is equal to 15, we therefore set

p to equal 5 which gives a σ value of 4. Based on the derived p value, the MOLS family can then be generated using any one of the many algorithms proposed in the literature [64–66]. The orthogonal square family generation step is shown in figure 4.2(a). The time-frame length (T_f) is also determined during the initialization step and is set to p to guarantee minimum throughput for each node [63].

Square/symbol assignment is also performed during the initialization phase. Since this mode is topology transparent, we expect the assignment of square symbols between the different network nodes to be done arbitrarily where each node is randomly assigned to a square and a square symbol. The authors in [67] have defined three different assignment methods, which are:

- **STATIC** Nodes are assigned to symbols and squares at deployment time such that each node has its unique assignment.
- **GREEDY** The base station initiates a broadcast protocol in which, after a passive learning period, each node ultimately broadcasts its symbol and square assignment. Nodes choose an unused symbol in the most used square based on the broadcasts they receive from their neighbors during the learning period.
- **RANDOM** The base station initiates a broadcast protocol in which, after a passive learning period, each node ultimately broadcasts its symbol and square assignment. Nodes choose an unused symbol in a random square based on the broadcasts they receive from their neighbors during the learning period.

We note how the STATIC method completely eliminates a configuration cost, while the other two methods require a neighbor exchange, although of a low cost. Once a sensor assigns itself a symbol-square combination it never needs to change it. Figure 4.2(b) shows the square/symbol assignment for the network nodes during the initialization step of the network setup phase. We focus on nodes E, G and F since they will be implementing the reaction step in our attack example.

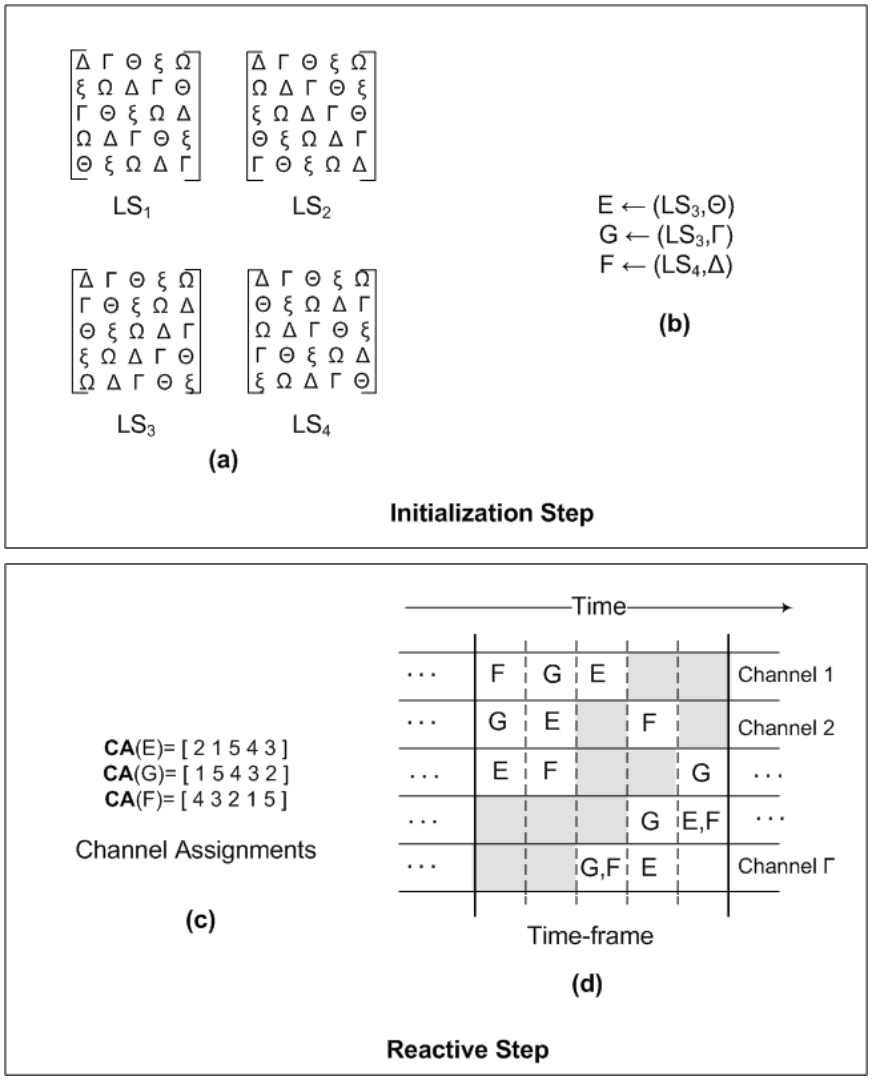


Figure 4.2: Running example under topology-transparent mode (mode 0).

4.2.2 Reaction

When a node switches to exfiltration state due to an attack, it immediately starts exfiltrating its data on the different channels using its symbol's pattern as it appears in the Latin square assigned to it. Going back to the RCAP formal model (section 3.3.1), a node j in exfiltration state will use its assigned transmission square to construct its channel assignment vector CA(j). The channel assignment vector captures the result of performing Latin square assignments and represents the output of the RCAP modeling approach. Figure 4.2(c) shows the channel assignment vectors for each of the attacked nodes during exfiltration

time using the chapter's running example (Figure 4.1). The node then exfiltrates its data during the different time-slots on the different channels assigned by its **CA** vector.

In the case when the number of available channels Γ is less than p, a transmitting node can trim its square into a $\Gamma \times p$ rectangle and use the resultant Latin rectangular. Figure 4.2(d) shows the multi-channel transmission time-frame for the running example. As shown, all three attacked nodes are transmitting concurrently on the different channels based on the pattern of their assigned symbol in their assigned transmission square. We note that concurrent transmissions during the third and fifth time-slots will result in collisions since the two transmitting nodes are neighboring nodes. This is because this mode provides no guarantee about preventing transmission collisions between neighboring nodes. However, the work in [63, 67] shows that it is possible to maintain a minimum throughput for each node by using simple dimensioning heuristics.

The boundary nodes also need to switch their transmission channels to receive the exfiltrated packets. In this mode, coordination between attacked nodes and their boundary neighbors may not occur since each node may or may not remember the square-symbol assignments of its neighbors, depending on the protocol's implementation of the square/symbol assignment method (static, greedy or random). Therefore, a boundary node may need to perform some type of channel hopping in order to learn the transmission pattern of its attacked neighbor. Furthermore, a boundary node needs to maintain its connection to the rest of the network on the common channel and only need to perform a channel switch during certain time-slots (after learning the transmission pattern of its attacked neighbor).

This mode of operation is presented here as a base case. Its performance is not evaluated as part of the concurrent data exfiltration framework presented in this dissertation. We note that this mode's performance is easily exceeded using the approach which combines vertex-coloring with Latin square assignments, which we explain next.

4.3 L-VC Mode: Collision Aware Broadcasting

L-VC mode combines **L**atin square assignments with **V**ertex-**C**oloring. This second mode

of operation focuses on the process of exfiltrating data packets such that all concurrent transmissions are *guaranteed not to collide between any two neighboring nodes.* Each attacked node will maintain its own local time frame that fits its local neighborhood size and avoids any conflicts with its 1-hop neighbors. To achieve this, we combine a vertex-coloring algorithm with the Latin square assignments such that a node is assigned to a square symbol based on its vertex-color. We start by stating some definitions to provide the needed background.

4.3.1 Preliminaries

Regarding the transmission time-slots as vertex-colors and allowing only a node whose vertex-color is matching the color assigned to the current transmission-slot is an approach widely used in transmission scheduling. We adopt this approach when designing our reactive defense, and we assign a time-slot for each vertex-color during a time-frame. We further provide the following definitions about vertex-coloring, found in [94].

Definition. *A* **vertex k-coloring** *of a graph G is an assignment $f : V_G \leftarrow C$ from its vertex-set V_G onto the set C=1,...,k, or onto another set of cardinality k, whose elements are called colors. For any k, such an assignment is called a* **vertex-coloring**.

Definition. *A* **color class** *is a vertex-coloring of a graph G is a subset of V_G containing all the vertices of a given color.*

Definition. *A* **proper vertex coloring** *of a graph is a vertex-coloring such that the endpoints of each edge are assigned to two different colors.*

Definition. *A graph is said to be* **vertex k-colorable** *if it has a proper vertex k-coloring.*

Definition. *An* **optimal vertex coloring** *of a graph is a vertex-coloring which results in the minimum number of colors.*

Lemma 5. *The vertices of any graph G with N nodes and maximum node degree $\triangle$ can be colored using at most $\triangle + 1$ colors.*

With this background, we now explain the RMBS steps during the L-VC mode.

4.3.2 Initialization

In the L-VC mode, we perform a *distributed* vertex-coloring of all network nodes during network initialization to capture direct interference between the different nodes. Such interference information is represented by the compatibility matrix **C** which is regarded as an input parameter to the RCAP model. The calculation of the **C** matrix is therefore assumed as a prerequisite step to the implementation of vertex-coloring step. According to the interference model of the network, two nodes' transmissions are considered interfering when the two nodes' broadcast areas overlap (i.e, one-hop neighbors) which would result in a collision if they broadcast simultaneously on the same channel.

Definition. *Two neighboring network nodes i and j separated by a 1-hop distance, are said to have a* **primary conflict** *since their simultaneous transmissions on the same channel will always collide. Two nodes are* **conflict-free** *if they do not suffer from a primary conflict.*

We use the running example to present the actual working of the L-VC mode in figure 4.3. The top of the figure shows the network topology with numbers in parenthesis to indicate the vertex-colors assigned by the vertex-coloring algorithm. We note here that the network example displays a *proper vertex coloring* of the different network nodes where the network is shown to be *vertex 3-colorable*. Furthermore, since the coloring algorithm has resulted in the minimum number of colors, vertex-coloring in this case is said to be *optimal*. Such an optimality is desirable in our approach since each vertex-color corresponds to a time-slot assignment during the reaction phase. This leads to a critical system decision in regards to the value of T_f.

In general, the minimization of a TDMA time-frame is important since the length of the scheduling time-frame has its own impact on system performance [88]. Specifically, a longer frame results in longer waiting time between subsequent transmissions and hence will have its effect on *delay*. In our schedule creation approach, vertex-colors are translated into transmission time-slots, and we set the time frame length to equal to k, which represents the total number of produced vertex-colors in the system. Therefore, the time-frame length decision in this case is dependent on the vertex-coloring step. A coloring algorithm which

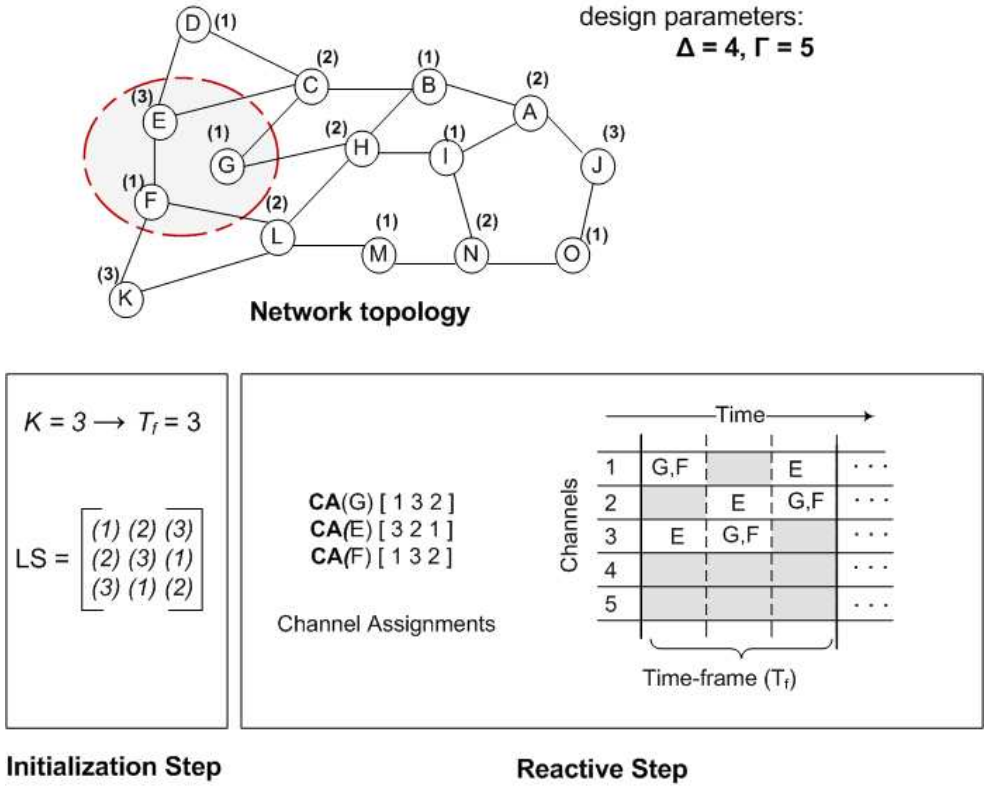


Figure 4.3: Running example under L-VC mode.

produces the minimum number of vertex-colors will lead to a schedule with a minimum number of time-slots per a time-frame. In our example scenario (figure 4.3), the total number of produced colors k is 3 and therefore T_f is also set to 3 in this case.

The square generation in this mode of operation is performed as follows: set the square dimension p to be equal to the time-frame length T_f. If the number of channels Γ is less than this value of p, then the $p \times p$ Latin square is trimmed into an $\Gamma \times p$ Latin rectangular. In this sense, each Latin square or Latin rectangle can be used to construct the transmission schedules for p different vertex-colors. Since p is set to equal the time-frame length which incorporates all possible vertex-colors, all network nodes can share a *single* Latin square or rectangle for all their transmissions. Therefore, the use of vertex-colors as square symbols leads to a significant reduction of the scheduling complexity. As shown in the figure, a *single* common square is now sufficient to be shared by all the N nodes for their schedule creation during attack time. We use the word common here to enforce the use of the *same* square

amongst the different nodes for schedule creation and we state the following theorem.

Theorem 1. *For a given k of vertex-colors in a network with N nodes,* **conflict-free** *transmissions are guaranteed if all network nodes use their vertex-color as a square symbol in a* **common** *$k \times k$ Latin square shared by all nodes.*

Proof. Given that the Latin square dimension's incorporates all the different vertex-colors in the system, all nodes can share the *same* $k \times k$ square to perform their transmissions since each node can guarantee a transmission slot assignment when it uses its vertex-color as a square symbol. Furthermore, from the definition of a Latin square (definition 3.4.1) we know that every symbol appears only once in each row and once in each column. Therefore, any two time-slot assignments chosen from the same Latin square will not have any overlap in their patterns and therefore their transmissions will never collide, as formally stated in lemma 1. □

For the case when the k value is not known beforehand (since our approach assumes only local information), the time frame length T_f can then be set to equal to $\triangle + 1$, which is equal to the maximum number of colors possible as stated in lemma 5. This may not be the best choice since it considers the upper bound value and the actual number of colors may be less, leading to unused slots. However, we believe that this is a reasonable design choice since the main task in designing a TDMA schedule is to allocate time-slots depending on the topology and the node packet generation rates. We make use of the *empty* slots to further optimize the performance, as will be explained in the **MuVC** section.

4.3.3 Reaction

In the exfiltration state, a node j will use the common $k \times k$ Latin square (or $k \times \Gamma$ triangle) to construct its own channel assignment vector CA(j). The node then exfiltrates the data during the different time-slots on the different channels using its **CA** vector. A guarantee on the minimum throughput for each participating node is provided by the next theorem.

Theorem 2. *For a given k number of vertex-colors in a network with N nodes where each node has a transmission schedule corresponding to a symbol pattern in a $k \times k$ Latin square, each node will be assigned a guaranteed minimum of transmission slots equal to min(Γ,k) in each multi-channel time-frame. Further, since all transmissions are guaranteed conflict-free, the* **guaranteed minimum throughput** *for each node is equal to min(Γ,k).*

Proof. The dimension of the Latin square, used for transmission scheduling in this case, is set to equal k, which corresponds to the number of colors in the system. Hence, each color class will be guaranteed a transmission during *each* time-slot, provided enough channels are available. Thus, in the case when $\Gamma >= k$, each color class will be assigned to one of the different channels in each time-slot and, hence, a color class will be guaranteed a total of k transmissions during a time-frame. In the case when there are not enough channels to be assigned for each color class during a transmission time-slot ($\Gamma <= k$), the $k \times k$ Latin square will be trimmed into a $\Gamma \times k$ Latin triangle and, hence, a color class may or may not be assigned to a channel during a time-slot. In this case, the total number of square assignments per a time-frame is reduced from $k \times k$ assignments to $\Gamma \times k$. Each color class can then guarantee $(\Gamma \times k)/T_f$, i.e, Γ assignments during a time frame. Since all transmissions are guaranteed conflict-free, the number of assigned transmissions also decides throughput in this case, which is equal to the minimum of k,Γ. □

As shown in figure 4.3, the L-VC provides for an efficient technique for schedule generation based on the interference information of the node. In the depicted example scenario, the results obtained when the 3 attacked nodes implement the reactive step of the RMBS model are shown in the lower box. Nodes E, F and G will start their exfiltration process by first calculating the channel assignment vector **CA** which is then used for performing the actual transmissions during a transmission time-frame, as shown on the diagram. Their boundary neighbors will also be able to figure out the transmission pattern of their attacked neighbor (by calculating the channel assignment vector of the attacked neighbor n, **CA**(n). Hence, the boundary neighbor can simply switch their channel at the appropriate time to receive the exfiltrated data packets.

4.3.4 Channel Allocation Rules

We summarize our discussion about the RMBS scheduling under the L-VC mode as the following: during exfiltration mode, all transmission slots on all the channels are treated as a multi-dimensional time-frame. A *common* Latin square provides the mapping function between the different nodes and the different time-slots on the multiple channels based on their direct interference information. A node's interference information is captured by its vertex-color. We now present the channel allocation principles for the L-VC mode.

L-VC Distributed Scheduling Rule. For a given k of vertex-colors in a network with N nodes, we set the length of the transmission time-frame T_f to k to guarantee the minimum-rate requirement for each node. Let J be a subset of network nodes undergoing a communication DoS attack. A node j will be using its assigned vertex-color as an entry into a *common* $k \times k$ square (or $\Gamma \times k$ rectangle) to decide its transmissions on the multiple channels during the different time-slots of the time-frame. All assigned transmissions are guaranteed to be conflict-free by theorem 1. Furthermore, from Theorem 2, each node will have a guaranteed minimum throughput equal to min(k,Γ).

The above rule assumes knowledge about the *actual* number of resultant vertex-colors k. Setting the time-frame to equal to the total number of colors will result in best performance since the time-frame length is minimized in this case. It is possible for the sink node to initiate a broadcast protocol in which, after a passive learning period, each sensor ultimately broadcasts its vertex-color assignment. However, this may incur an increased overhead. Therefore, we define the following rule for the case when the k value is not global.

L-VC Centralized Scheduling Rule. For a given network with N nodes and maximal nodal degree $\triangle$, we set the length of the transmission time-frame T_f to $\triangle + 1$ in order to guarantee minimum-rate requirement for each node. Let J be a subset of network nodes undergoing a communication DoS attack. A node j will be using its assigned vertex-color as an entry into a *common* $(\triangle + 1) \times (\triangle + 1)$ square (or $\Gamma \times (\triangle + 1)$ rectangle) to decide its

transmissions on the multiple channels during the different slots. All assigned transmissions are guaranteed to be conflict-free by theorem 1. Furthermore, from Theorem 2, each node will have a guaranteed minimum throughput equal to min($\triangle + 1$,Γ)

4.4 MuVC Mode: Interference Aware Broadcasting

In this section, we present the third mode which provides RMBS with resilience against the hidden terminal problem. **MuVC** combines **Mu**tually orthogonal Latin square assignments with **V**ertex-**C**oloring.

4.4.1 Motivation

Theorem 1 states that L-VC does prevent against collisions occurring from a primary conflict between two nodes. However, due to the broadcast nature of radio signals, a node's interference range can be larger than its transmission range, i.e., it can interfere with another node's *reception* even if the other sender is not within its transmission range. Nodes within the transmission range of a receiver are usually called hidden nodes. When the receiver is receiving a packet, if a hidden node also tries to start a transmission concurrently, collisions will happen at the receiver. To design more efficient broadcasting techniques, both the collision and the interference among multiple transmissions must be addressed. Therefore, in this mode, we extend RMBS scheduling to include some collision avoidance mechanism, in order to provide resilience against the hidden terminal problem. Figure 4.4 shows the hidden-terminal collision vulnerability present in the L-VC mode.

In wireless networks, there are three basic radio range classifications which are essential in understanding the hidden terminal issue [95]. The carrier sensing range R_{cs} is the range within which a transmitter detects the carrier as busy if there is a transmission. This detection is usually determined by the antenna sensitivity. The transmission range R_{tx} represents the range within which a packet is successfully received if there is no interference from other radios. This transmission range is usually determined by transmission power and radio propagation properties. The interference range R_i is the range outside the carrier

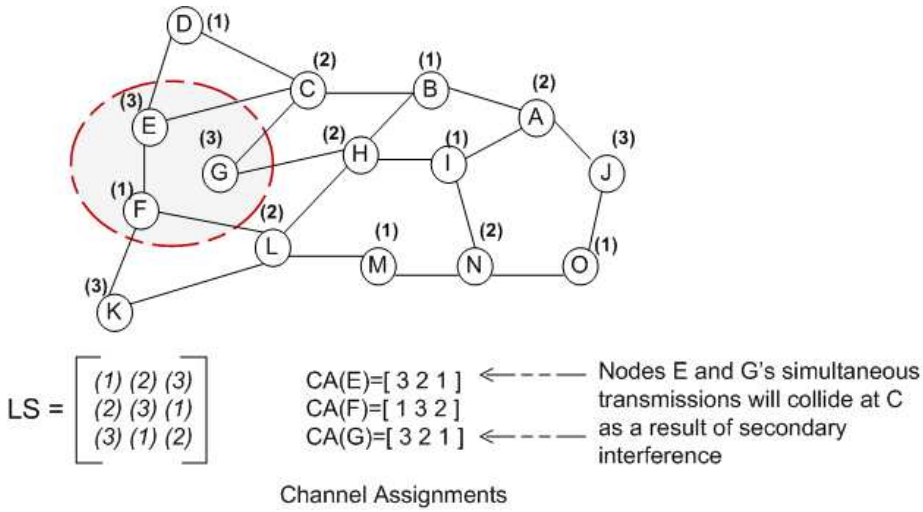


Figure 4.4: Hidden terminal problem in L-VC mode. Two simultaneous transmissions from non-interfering nodes may collide at the receiver.

sensing range, and within which the radio signal appears as noise at the receivers.

Most of the time, a node's carrier sensing range R_{cs} is at least twice as big as its transmission range R_{tx} [95]. Therefore, node activation scheduling usually requires all neighbors of a node within two hops be silent when the node transmits [95]. In developing a collision avoidance scheme for this mode, we adopt the two-hop distance when considering *secondary* interference.

Definition. *Two neighboring network nodes i and j separated by a 2-hop distance are said to have a* **secondary conflict** *since their simultaneous transmissions on the same channel might lead to a collision at the receiver.*

Definition. Collision-free *scheduling is one which protects against both primary and secondary conflicts.*

Therefore, in order to guarantee *collision-free* transmissions for the different node, secondary interference must also be considered. We enforce the following design decision to act as a collision avoidance scheme: *active nodes within a two-hop range of each other transmit on different orthogonal channels.* To achieve this, we extend the interference capturing step implemented by the vertex-coloring algorithm to a more restrictive vertex-coloring known as

distance-2 coloring. Thus, two nodes cannot be assigned the same vertex-color if one node is within the transmission range of the other or the two nodes have a common neighbor.

4.4.2 Prelimineries

We start by stating some definitions to provide the needed background about distance-2 vertex-coloring, found in [96].

Definition. *The* **distance-2 neighbors** *of a node include all of its one-hop neighbors and the one-hop neighbors of its one-hop neighbors.*

Definition. *The purpose of* **distance-2 coloring** *of a graph G=(V,E) is to produce an assignment of colors C:V* $\rightarrow$ *1,2,.. such that no two nodes are assigned the same color if they are distance-2 neighbors.*

Fact 1. *Every proper distance2 vertex-coloring of G requires at least* $\triangle + 1$ *colors, and is upper bounded by* $o(\triangle^2)$.

Next, we explain the working details of mode MuVC.

4.4.3 Initialization

To count for the possibility of secondary conflict between the different nodes, we redefine the compatibility matrix **C** of the RCAP problem so that the 2-hop distance is taken into consideration. The corresponding compatibility matrix C can be directly obtained from the connectivity matrix CN (section 3.3.1), and is defined as:

$$\mathrm{C}_{ij} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are one-hop or two-hop neighbors} \\ 0, & \text{otherwise} \end{cases}$$

We implement distance-2 vertex-coloring as the following: if G denotes the graph modeling the network, then the distance-2 coloring problem on G is equivalent to the standard minimum vertex-coloring problem on G^2, where G^2 has the same vertex set as G and there

is an edge between two vertices of G^2, if and only if, there is a path of length at most 2 between the vertices in G [96].

By implementing the distance-2 variant of the distributed vertex-coloring algorithm implemented in the RBMS approach, we now enforce a mapping between nodes and transmission slots to also consider secondary conflict as well. As a result, simultaneous transmissions under this mode of operation will be guaranteed *collision-free*.

Since each color class corresponds to a channel assignment in the transmission time-frame, it is important to produce a distance-2 coloring that uses the minimum number of colors in order to result in a minimal time-frame length (T_f). In the L-VC approach, the time-frame is constructed to guarantee transmissions for all the the different nodes within a neighborhood on the different channels. Therefore, its length T_f is decided by either the *actual* total number of vertex-colors k or the *upper-bound* value of the expected number of vertex-colors ($\triangle + 1$). However, such an approach will not be effective in the MuVC mode. $\triangle + 1$, which was an upper bound in the distance-1 vertex-coloring case, becomes a lower bound in the distance-2 coloring case as stated in fact 1. The number of vertex-colors in the distance-2 coloring case is upper-bounded by $\triangle^2$. Such a large choice for the frame-length will lead to degradation in performance since many slots may go unused and hence wasted.

We make the choice for the value of T_f in the MuVC mode to be independent from the actual k value or the upper-bound value on the possible number of vertex-colors. A more effective approach would be to minimize the length of the time-frame, by using the lower-bound value on the possible number of vertex-colors, and then use a technique that would still incorporate all the possible different vertex-colors into this minimal frame length. We fix the time frame length T_f to equal the lower bound expected by the distance-2 vertex-coloring algorithm, $\triangle + 1$. We also adopt this same value as the order of the transmission common Latin square. If the number of channels Γ is less than the order of the square, the square is then trimmed into a $\Gamma \times (\triangle + 1)$ Latin rectangle and the resulted rectangle is used for the creation of concurrent exfiltration assignments.

We use the running example to present the actual working of the MuVC mode in figure

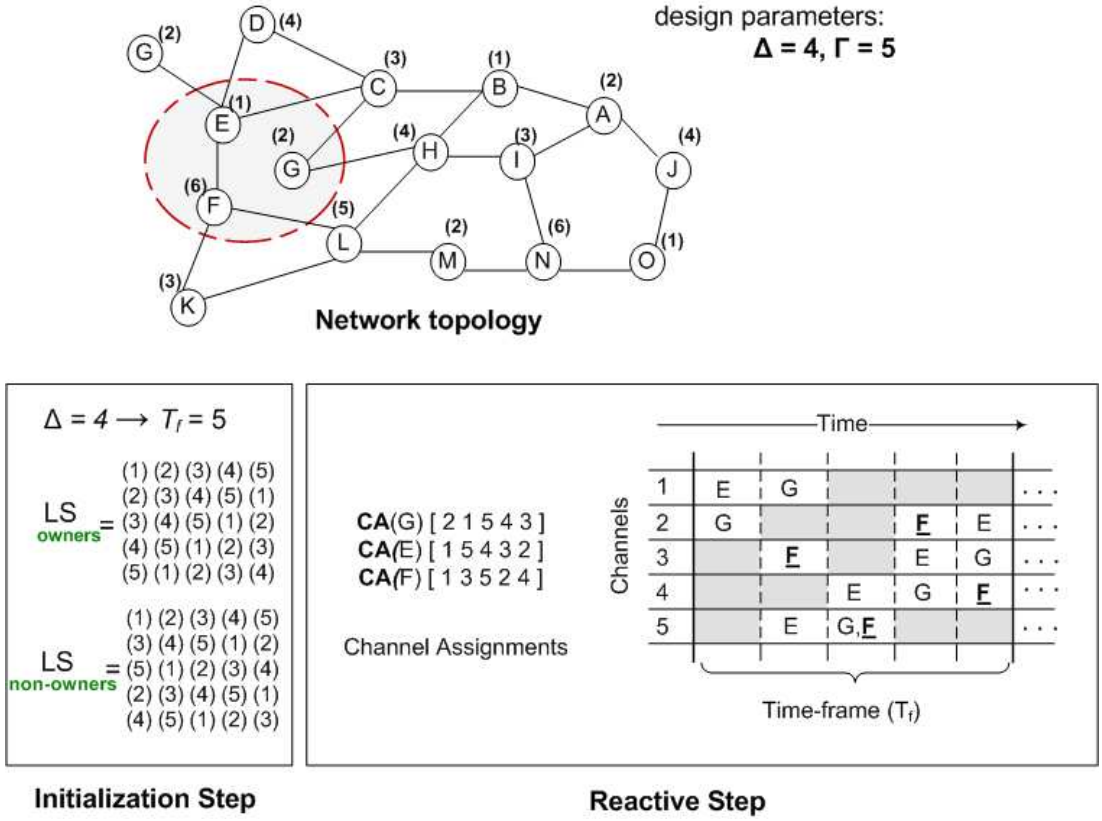


Figure 4.5: Running example under MuVC for a sparse network. Node F is a non-owner node and hence can only share the time-frame.

4.5. The result of performing the distance-2 vertex-coloring step is also shown on the topology depicted in the figure. The time-frame length T_f is set to 5 since the maximal nodal degree $\triangle$ is equal to 4. For the square generation, we use *two* orthogonal squares of order 5. We justify the use of two squares when we explain schedule creation in the reaction phase.

4.4.4 Reaction

In the MuVC mode, the dimension of the transmission Latin square does not guarantee a transmission slot for all the possible vertex-colors. We call nodes whose original color is the same as their square symbols *owners* of any transmission slot which has their vertex-color assigned for transmission. A node whose assigned vertex-color exceeds the dimension of the transmission Latin square will be a *non-owner* node and will be using *modular arithmetic*

to assign itself a square symbol in the range [0..$\triangle$+1]. Furthermore, to incorporate all the extra vertex-colors, we define a different transmission square (from the same orthogonal family). The use of a different transmission square is necessary to minimize the possibility of collisions since it is stated in lemma 2 that if two nodes obtain the same symbol from two different orthogonal squares, their transmission slots will have at most one conflict during each frame. Therefore, we define two transmission squares from the same orthogonal family, one to be a common square for the owner nodes and another to be common for the non-owners. We then distinguish the performance MuVC based on the density of the network since the length of scheduling is directly related to the density of the network [88]. We use the parameter k, which represents the actual number of vertex-colors in the system, to decide the density of the network as follows:

Definition. *For a given k of vertex-colors in a network with N nodes, the network is said to exhibit a* **sparse** *density if* $k \leq 2(\triangle + 1)$*, a* **dense** *density if* $2(\triangle + 1) \leq k \leq 4(\triangle + 1)$*, and a* **very dense** *density if the value of k exceeds* $4(\triangle + 1)$*.*

We then construct the scheduling scheme as follows.

1) If the network's density is *sparse*, we use a single-time frame to minimize scheduling latency. We incorporate all the different transmissions into this single frame in the following way: during the slots where owners have data to transmit, RMBS eliminates the chance of collision since *only* owners are given direct transmission right, but when a slot is not in use by its owners, non-owners can steal the slot. In other words, non-owner nodes are given a lower priority for performing their transmissions. In the example scenario, nodes E and G will generate their channel assignment vectors by using their assigned vertex-colors as a symbol in the owners' square, and node F will use the value (6 mod 5) as its square symbol and hence is considered a non-owner node. Since the square symbols for both F and E are identical, their concurrent transmissions will be colliding at all times if they use the same common Latin square, therefore, node F will be using the non-owners square. Such an approach will lead to *collision-free* scheduling, however, it will penalize nodes that are assigned larger vertex-colors.

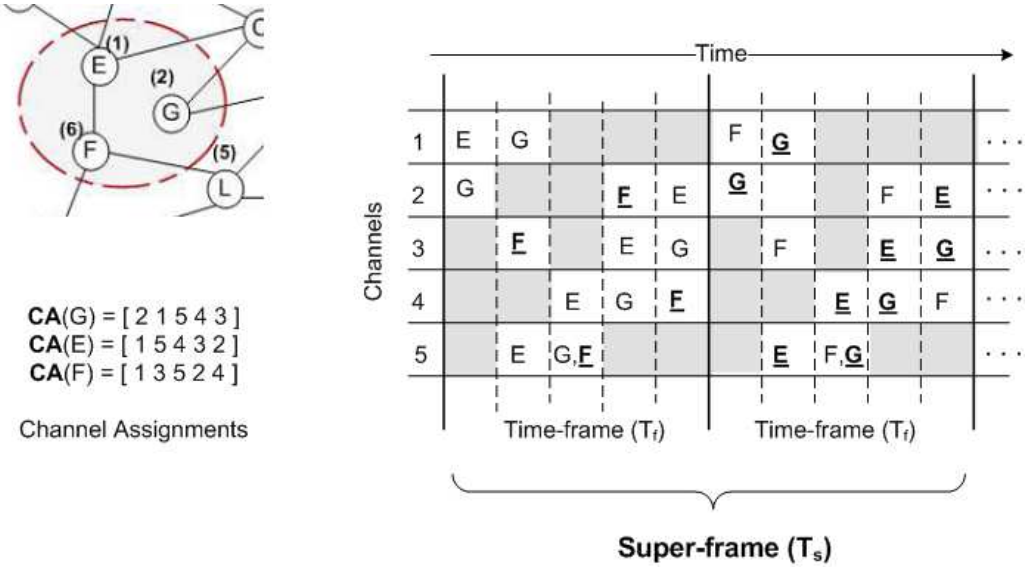


Figure 4.6: Running example under MuVC for a dense network. A *super-frame* will guarantee collision free transmissions by using the *owners* and *non-owners* squares consecutively. Node F is "sharing" the slots in the first time-frame, nodes E and G "share" the second time-frame.

A different approach would be to allow both owners and non-owners to transmit simultaneously during a slot. In this case, since a slot is shared by an owner node and a non-owner node, the collision possibility is at most one (lemma 2). We call such an approach *collision-reduced* scheduling.

2) In the case of *dense* networks, an increased time-frame length is necessary since a larger number of non-owners is expected. For that, we introduce the notion of a *super-frame* T_s (figure 4.6) which consists of two consecutive time-frames facilitated as follows: in the first time-frame, owners will have the explicit transmission right and non-owners can only transmit if a slot is unused. In the second time-frame, non-owners will have the transmission priority. Such an approach will lead to *collision-free* scheduling (as will be formally stated in theorem 3) and guarantees transmission fairness between owners and non-owners. Again, we can allow both owners and non-owners to transmit concurrently in which case the resulted scheduling will be *collision-reduced*.

Theorem 3. *For a given k*2 *of vertex-colors in a network with N nodes,* **collision-free** *transmissions are guaranteed if all network nodes use their* **distance-2** *vertex-color as a*

square symbol in a **common** $k2 \times k2$ *Latin square shared by all nodes.*

Proof. Given that the Latin square dimension's incorporates all the different vertex-colors in the system, all nodes can share the same $k2 \times k2$ square to perform their transmissions since each node can guarantee a transmission slot assignment when it uses its vertex-color as a square symbol. Furthermore, from the definition of a Latin square, we know that every symbol appears only once in each row and once in each column. Therefore, any two time-slot assignments chosen from the same Latin square will not have any overlap in their patterns. In addition, since distance-2 vertex-coloring takes into account secondary interference, we conclude that the resulted scheduling is collision-free. □

3) For the case when the network is *very dense*, we use a construction algorithm for finding the optimal p value and time-frame length T_f to implement MOLS scheduling that would lead to best throughput. We base our construction on the work presented in [64]. The authors in [64] present a topology-transparent algorithm based on Latin squares and MOLS assignments for the case of ad hoc networks in general. The difference between their algorithm and ours is that in our work the network nodes are topology-aware, since we incorporate a vertex-coloring algorithm. As a result, the scheduling complexity is greatly reduced in our approach since we only consider vertex-colors as opposed to their approach which considers all the N nodes in the network. In other words, their construction algorithm performs a square/symbol assignment for each node, while in our approach the square/symbol assignment is performed for each vertex-color. We use their approach to derive the best value for the square family order p that would lead to best throughput when *vertex-colors are used as square-symbols.*

4.4.4.1 Optimal Algorithm Construction

The different nodes will be using different orthogonal Latin squares when performing their transmissions concurrently. If $p > \Gamma$, a Latin square is trimmed into a $\Gamma \times p$ rectangle. Furthermore, we assume that the size of the orthogonal family is equal to σ. To ensure

a transmission slot for all nodes during a transmission time-frame, the number of available square symbols must be enough to allow $\triangle^2$ (which represents the upper-bound on the number of possible vertex-colors) assignments, hence, the following equation must be satisfied:

$$p \times \sigma \geq \triangle^2 \tag{4.2}$$

Therefore, the p value is directly dependent on the number of vertex-colors (represented by $\triangle^2$) in the system, and since we can only assume knowledge about the input parameter $\triangle$, we used the upper-bound on the number of possible vertex-colors. However, replacing the $\triangle^2$ by the *actual* resultant number of vertex-colors, in the 4.2 equation, will lead to better performance in regards to *delay*. We note here the major reduction in the problem size from the topology-transparent mode when we compare equation 4.2 with equation 4.1.

The dimension of the Latin square used for transmission scheduling is set to equal p, and each p color classes share a $p \times p$ square to generate their schedules. In the case when $\Gamma >= p$, each color class will be assigned to one of the different channels in each time-slot and hence will be guaranteed a total of p transmissions during a time-frame. In the case when there are not enough channels to be assigned for each color class during a transmission time-slot (i.e. $\Gamma < p$), the $p \times p$ Latin square will be trimmed into a $\Gamma \times p$ Latin rectangular and hence a color class may or may not be assigned to a channel during a time-slot. In this case, the total number of square assignments per a time frame is reduced from $p \times p$ assignments to $\Gamma \times p$. Each color class can then guarantee $\Gamma \times p/p$, i.e, Γ assignments during a time frame. We now provide the following definition.

Definition. *The total number of* **successful transmissions** θ *is equal to the guaranteed transmission assignments per time-frame less the number of possible collisions:*

$$\theta = min(\Gamma, p) - number\ of\ collisions.$$

Our construction objective is to find a p value that will maximize throughput for the case when network nodes are using their distance-2 vertex-color as square symbols. We know that every node will have a *non-zero* throughput when the following inequality holds:

$$p \geq \theta > 0 \tag{4.3}$$

To analyze the number of possible collisions during a time-frame, we present the following theorem.

Theorem 4. *In a given network where nodes use their distance-2 vertex-color to construct their transmission schedule based on a symbol pattern in one of the Latin squares from an order p orthogonal Latin square family. When a node is surrounded by $\triangle$ other nodes, it suffers from at most $\triangle$ collisions. In addition, the minimum number of collision chances that it can have is equal to max($\triangle + 1 - p$, 0).*

Proof. From Lemma 2 and following the reasoning in [63], we know that each neighbor of node j can result in at most one collision for node j. Therefore, the worst case scenario would be if node j is connected to all $\triangle$ possible neighbors and if they all chose a different square for their transmissions. In this case, node j's transmissions will all result in collisions and hence the maximum number of collision chances would be equal to $\triangle$. To calculate the minimum number of collision chances, we have to count the maximum number of nodes that can determine their transmission schedules from a common Latin square, which is p, and assume that for these p radio units no collision will occur among them according to lemma 1. Furthermore, we also know (from lemma 2) that there is only a singe chance of a collision between any two nodes belonging to two different squares. Therefore, for each of the remaining nodes $\triangle$ we can assume that a collision may or may not occur. Therefore, the minimum number of collisions possible would be max($\triangle - (p-1)$,0)=max($\triangle + 1 - p$,0). $\square$

Based on Theorem 4, we derive the upper and lower bounds on θ to be:

$$\theta_{max} = min(\Gamma, p) - max(\triangle + 1 - p, 0) \tag{4.4}$$

$$\theta_{min} = min(\Gamma, p) - \triangle \tag{4.5}$$

The above equations can be used further to derive bounds on throughput.

Definition. *The* **guaranteed throughput** Ω_{min} *is defined as the ratio of number of guaranteed successful transmissions during a time-frame divided by the length of the time-frame:*

$$\Omega_{min} = \theta_{min}/T_f \tag{4.6}$$

Definition. *The* **maximum throughput** Ω_{max} *is defined as the ratio of the maximum possible number of successful transmissions in a time-frame divided by the time-frame length:*

$$\Omega_{max} = \theta_{max}/T_f \tag{4.7}$$

Theorem 5. *For a given* $\triangle$ *and* Γ*, the guaranteed minimum throughput* Ω_{min} *and the maximum possible throughput* Ω_{max} *are equal to:*

$$\Omega_{min} = \frac{min(\Gamma, p) - \triangle}{p} \tag{4.8}$$

$$\Omega_{max} = \frac{min(\Gamma, p) - max(\triangle + 1 - p, 0)}{p} \tag{4.9}$$

Proof. For the first part of the theorem (Ω_{min}), use equation 4.6 and substitute the θ_{min} in the nominator by equation 4.5 and substitute the T_f value in the denominator with p since in Latin square scheduling the time-frame length is always set to the order of the Latin square family [64]. For the second part of the theorem (Ω_{max}), use equation 4.7 and substitute the θ_{max} in the nominator with equation 4.4 and substitute the T_f value in the denominator with p. □

Next, we introduce the following theorem which formally states the upper and lower bounds on throughput. We note here that our proof in this theorem is achieved by demonstrating the applicability of theorem 5 in [63] into our approach.

Theorem 6. *For a given* $\triangle$ *and* Γ*, the maximal nonzero upper and lower bounds of throughput* Ω *are:*

$$1 \quad \geq \Omega \geq 1 - \frac{\triangle}{\Gamma}, \qquad \text{if } p \leq \Gamma$$

$$\frac{\Gamma}{max(\lfloor \frac{\triangle^2}{\sigma} \rfloor, \Gamma)} \geq \Omega \geq \quad \frac{\Gamma - \triangle}{max(\lfloor \frac{\triangle^2}{\sigma} \rfloor, \Gamma)}, \quad \text{if } p > \Gamma$$

Proof. To incorporate all the possible vertex-color values to transmit during a time-frame and from equation 4.2, we know that the following must hold:

$$p \geq \lfloor \triangle^2/\sigma \rfloor \tag{4.10}$$

1) Case when $p \leq \Gamma$: We use theorem 5 to calculate the throughput bounds in this case as:

$$\Omega_{min} = 1 - \frac{\triangle}{p} \tag{4.11}$$

$$\Omega_{max} = 1 - \frac{max(\triangle + 1 - p, 0)}{p} \tag{4.12}$$

Since we are only considering the case of $p \leq \Gamma$ in this first part of the proof, and by using equation 4.10 we therefore bound the value of p as:

$$\lfloor \frac{\triangle^2}{\sigma} \rfloor \leq p \leq \Gamma \tag{4.13}$$

From equations 4.11 and 4.12, it is apparent that the use of larger squares (larger p value) will increase throughput since the number of concurrent assignments per a time-frame is increased. Thus, we now calculate the upper and lower bounds on the throughput as the following:

$$\Omega_{min} = 1 - \frac{\triangle}{\Gamma} \tag{4.14}$$

$$\Omega_{max} = 1 \tag{4.15}$$

2) Case when $p > \Gamma$: In this case, the value of p is defined as:

$$p > max(\lfloor \frac{\triangle^2}{\sigma} \rfloor, \Gamma) \tag{4.16}$$

We calculate the upper and lower bounds on throughput using theorem 5 as follows

$$\Omega_{min} = \frac{\Gamma - \triangle}{p} \tag{4.17}$$

$$\Omega_{max} = \frac{\Gamma - max(\triangle + 1 - p, 0)}{p} \tag{4.18}$$

From the above equation, we can see that the throughput bounds are now decreasing with an increased value of p, i.e. when there are not enough available channels to incorporate all the concurrent transmissions assigned. This is because the p value is in fact deciding the number of concurrent transmissions per a time-slot. We now calculate the upper and lower bounds on the throughput as the following:

$$\Omega_{min} = \frac{\Gamma - \triangle}{max(\lfloor \frac{\triangle^2}{\sigma} \rfloor, \Gamma)}$$

$$\Omega_{max} = \frac{\Gamma - max(\triangle + 1 - max(\lfloor \frac{\triangle^2}{\sigma} \rfloor, \Gamma), 0)}{max(\lfloor \frac{\triangle^2}{\sigma} \rfloor, \Gamma)} = \frac{\Gamma}{max(\lfloor \frac{\triangle^2}{\sigma} \rfloor, \Gamma)}$$

□

Based on the above, we can now construct an optimal scheduling algorithm for RMBS scheduling for the case of a *very dense* network, under this mode.

4.4.4.2 MuVC Optimal Algorithm.

- To incorporate the transmissions of all possible vertex-colors, we set the optimal p value according to the following rules:
  - When $T_f \leq \Gamma$, use $p = \lfloor \frac{\triangle^2}{\sigma} \rfloor$
  - When $T_f > \Gamma$, use $p = \max(\lfloor \frac{\triangle^2}{\sigma} \rfloor, \Gamma)$
- Use p to construct an orthogonal family of $p \times p$ Latin squares.
- Each node j uses its vertex-color (C_j) as its square symbol and assigns itself to Latin square number $_j$ $/p\rfloor$, from the generated Latin square family.
- During exfiltration mode, each node constructs its transmission schedule based on its vertex-color and its transmission Latin square and performs its transmissions on the multiple channels accordingly.

4.4.5 Channel Allocation Rules

We summarize our discussion about the RMBS scheduling under the MuVC mode as the following: during exfiltration state, a common MOLS family provides the mapping function between the different nodes and the different time-slots on the multiple channels based on their interference information which is captured by a distance-2 vertex-coloring step. For the case of a sparse network, a single time-frame is sufficient to incorporate the transmissions of the different nodes. For dense network, a super time-frame (consisting of two consecutive time-frames) can be used. Furthermore, for both cases, the transmissions can be performed collision-free or collision-reduced, as explained in the channel allocation rules presented next. Furthermore, to incorporate *all* possible $\triangle^2$ vertex-color values, the optimal construction presented can be used. We now present the channel allocation rules for the MuVC mode.

1) MuVC for Sparse Networks:

Collision-free Broadcasting. Let J be a subset of network nodes undergoing a communication DoS attack. During exfiltration state, each node will be setting its time-frame length T_f to be equal to $(\triangle + 1)$. A node j_1 whose assigned vertex-color value c_1 is in the range [0 - $\triangle + 1$] will be an *owner node* and hence will be using the common $(\triangle + 1) \times (\triangle + 1)$ **owner's square** to create its transmission schedule and perform its direct transmissions. A node j_2 whose assigned vertex-color value c_2 exceeds the value $(\triangle + 1)$ will be a *non-owner* node and hence will be using the value (c_2 **mod** $\triangle + 1$) as its square symbol in the common $(\triangle + 1) \times (\triangle + 1)$ **non-owner's square** to create its transmission schedule and perform its transmissions on the transmission slots which are unused by their owners. All concurrent transmissions are guaranteed collision-free in this case (theorem 3).

Collision-reduced Broadcasting. Let J be a subset of network nodes undergoing a communication DoS attack. During exfiltration state, each node will be setting its time-frame length T_f to be equal to $(\triangle + 1)$. A node j_1 whose assigned vertex-color value c_1 is in the

range [0 - $\triangle+1$] will be an *owner node* and hence will be using the common $(\triangle+1)\times(\triangle+1)$ **owner's square** to create its transmission schedule and perform its direct transmissions. A node j_2 whose assigned vertex-color value c_2 exceeds the value $(\triangle+1)$ will be a *non-owner* node and hence will be using the value (c_2 **mod** $\triangle+1$) as its square symbol in the common $(\triangle+1)\times(\triangle+1)$ **non-owner's square** to create its transmission schedule and perform its transmissions obliviously. A minimum throughput is guaranteed for each node based on theorem 6.

2) MuVC for Dense Networks

Collision-free Broadcasting: Let J be a subset of network nodes undergoing a communication DoS attack. During exfiltration state, each node j will be setting its transmission **super-frame** length EX_f to be equal to $2(\triangle+1)$, and will assign itself a square symbol by using modular arithmetic to round its vertex-color in the range [1 - $(\triangle+1)$]. A node j_1 whose assigned square-symbol is in the range [0 - $\triangle+1$] will be an owner node and hence will be using the $(\triangle+1)\times(\triangle+1)$ owner's square to create its transmission schedule and perform its transmissions *during the first time-frame* of the super-frame. A node j_2 whose assigned square-symbol exceeds the value $\triangle+1$ will be a non-owner node and hence will be using the non-owners common square to create its transmission schedule and perform its transmissions during the *second time-frame* of the super-frame. All transmissions are guaranteed collision-free based on theorem 3. If a slot is being unused during the first time-frame, a non-owner may use that slot. If a slot is being unused in the second time-frame, an owner node can use it.

Collision-reduced Broadcasting: Let J be a subset of network nodes undergoing a communication DoS attack. During exfiltration state, each node j will be setting its transmission **super-frame** length EX_f to be equal to $2(\triangle+1)$, and will assign itself a square symbol by using modular arithmetic to round its vertex-color in the range [1 - $2(\triangle+1)$]. A node j_1 whose assigned square-symbol is in the range [0 - $(\triangle+1)$] will be using the $(\triangle+1)\times(\triangle+1)$ owner's square to perform its transmissions during the *first* time-frame of the super-frame.

A node j_2 whose assigned square-symbol is in the range $[\triangle + 1..2\triangle + 1]$ will also be using the owners square to perform its transmissions during the *second* part of the super-frame. A node whose assigned square symbol exceeds the value $2(\triangle + 1)$ will be using the non-owners common square to create its transmission schedule and perform its transmissions during either the first or second time-frame of the super-frame. All nodes are guaranteed minimum throughput provided by theorem 6.

4.5 MuVC-S/R Mode: Coordinated Broadcasting

In this section, we develop a *coordinated* scheduling strategy and incorporate the strategy in the RMBS design. When the attack affects a multi-hop area, in order to result in a comprehensive exfiltration task, some attacked nodes must also cooperate in receiving data from the inside of the attacked region and exfiltrating it further in the direction of a boundary node. Such cooperating nodes must therefore coordinate their listening duties along with their exfiltration tasks. For this purpose, we design this fourth mode of operation which performs coordinated scheduling, as opposed to just broadcast scheduling.

4.5.1 Motivation

RBMS is based on a fixed channel allocation approach. The mapping procedure in all previous modes maps a single sender with multiple receivers. For the purpose of *coordinating* senders and receivers, a channel assignment is defined as the mapping of a single sender and a single receiver to a single communication channel at any single time-slot.

For the purpose of channel allocation in this case, we resort again to MOLS scheduling to achieve precise channel assignment between *each* sender and receiver. A mapping between senders and receivers can then be represented by *ordered* pairs. Figure 4.7 shows an example. In the figure, two squares are used by senders S_{Owners} and $S_{Non-owners}$ and one square is assigned for the receivers $S_{Receiver}$. Note how the orthogonality of the squares guarantees exactly one time/channel assignment for every *pair* of nodes in the two squares, as shown by the S_{OR} and S_{NR} matrices. For instance, an owner node assigned to symbol b can transmit

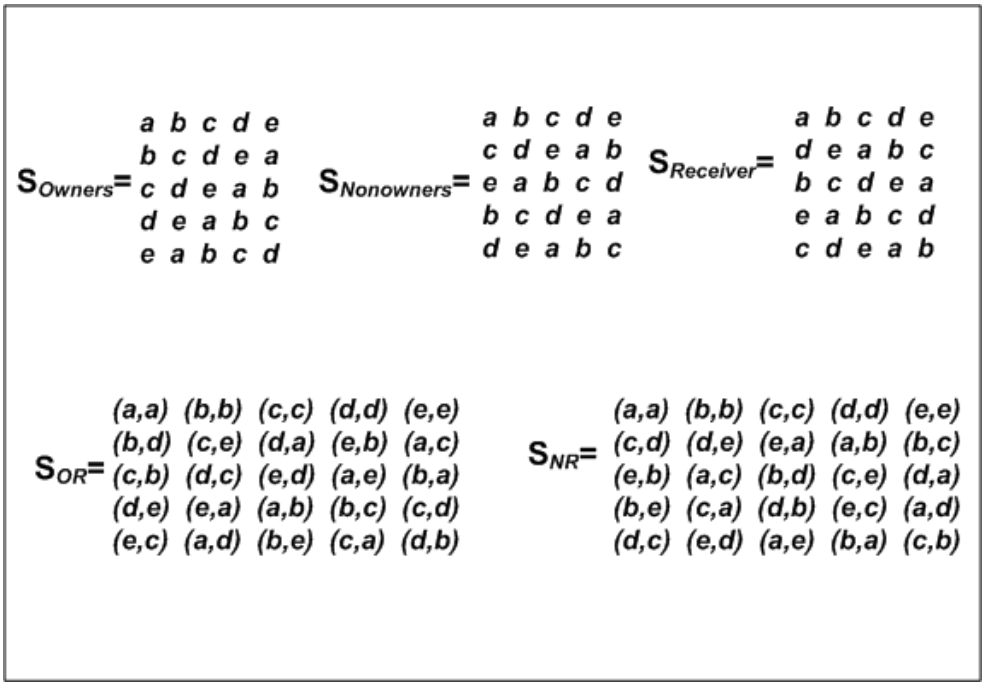


Figure 4.7: Coordinating senders and receivers using MOLS assignments.

to its neighbor with an assigned symbol d during the first time-slot of the time-frame on channel 2. This is done by combining the owner's square with the receiver's square and then finding the position of the *(b,d)* pair. A non-owner node will follow the same procedure, however, using the non-owners square.

This idea of assigning senders to receivers is very critical to designing effective exfiltration protocols against multi-hop attacks. Precise alignment of the sending and receiving tasks needs to be incorporated in the design of the transmission super-frame to guarantee accurate mapping and successful transmissions. We introduce next the structure of the **exfiltration time-frame** of the system which is especially designed to incorporate the coordinated assignments.

4.5.2 Exfiltration Time-frame ($\mathbf{exT}_f$)

For the purpose of coordinating senders and receivers, we use an extension to the RMBS scheduling where we redesign the underlying physical TDMA time-frame structure shown in Fig 3.4. In the extended design, a logical TDMA frame structure is constructed where we

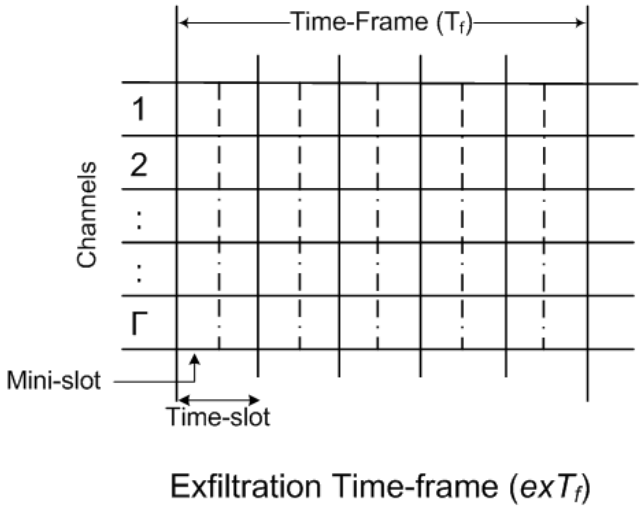


Figure 4.8: Exfiltration Time-frame $(\mathbf{exT}_f)$: a *logical* TDMA frame structure where we further divide each TDMA time-slot into two logical mini-slots.

further divide each TDMA time-slot into two logical mini-slots. The number of mini-slots is chosen to allow for both transmission and reception functionalities between nodes inside of the attacked region and their boundary neighbors during any single time-slot. We now describe how to implement such a logical design.

The logical TDMA frames are assumed to be occurring during the same physical TDMA frames of the RMBS, wherein each of the physical time-slots is further divided into two parts during physical time-slots and are temporally spaced apart by some predefined guard time. The guard time is needed to prevent the different transmissions from overlapping and to accommodate possible timing skew among nodes since different nodes have different propagation delays (due to the difference in distance between sender/receiver pairs). Therefore, the guard time is set to equal to the maximum difference in the round-trip radio propagation delay between any two nodes. The coordinated mini-slot assignments are performed concurrently on all the available channels as depicted in figure 4.8, which describes the concurrent channel assignments during a logical frame. In the diagram, a total of Γ channels are being assigned Γ concurrent transmissions and each time-slot is further divided to incorporate both the sending and receiving assignments.

Such a time-slot division is effective for the case of a boundary node listening to transmissions coming from inside of the attacked region in the first half of a time-slot, and then using the second part for forwarding the exfiltrated data packet further in the direction of the base station. It also provides a solution for attacks which may extend beyond the single-hop radius, in which case an attacked node will cooperate in receiving data packets coming from the inside of an attacked region, as will be explained in the MULEPRO protocol implementation in the next chapter.

4.5.3 Channel Allocation Rules for Receivers

From the MOLS family, RMBS assigns one square for owners, another square for non-owners, and a third square for the receivers, as depicted in figure 4.9. In the figure, Squares S_{Owners} and $S_{Non-owners}$ are used by the senders and square $S_{Receiver}$ is used by a receiving node. All three squares belong to the same family and hence are orthogonal with respect to each other. The orthogonality property of two squares guarantees a unique pair assignment of symbols from the two squares. Matrix S_{OR} includes unique pair assignments for the symbols from S_{Owners} and $S_{Receiver}$. This provides for a unique channel/time-slot assignment for each sender (using the S_{Owners}) and each receiver. Thus, a receiving node will be able to generate its *listening* schedule by combing its attacked neighbor's sending square (can be owner or non-owner) with the receiver's square.

We note here that this provides a solution for attacks which extend to multi-hop regions. A cooperating attacked node will be a sender but will also manage to listen to transmissions coming from the inside of the attack region. In addition, it provides a solution to a boundary neighbor trying to listen to transmissions coming from its attacked neighbors. In this way, a boundary node will be functioning in a normal mode and will *only* perform a channel switch during the slots to which it expects its attacked neighbor to be performing its transmissions. The assignments shown in figure 4.9 now assigns each transmitting node with a receiving neighbor. The figure shows the structure of the **exfiltration time-frame** exT_f under MuVC mode, for the sparse network. In the case of a sparse network, one

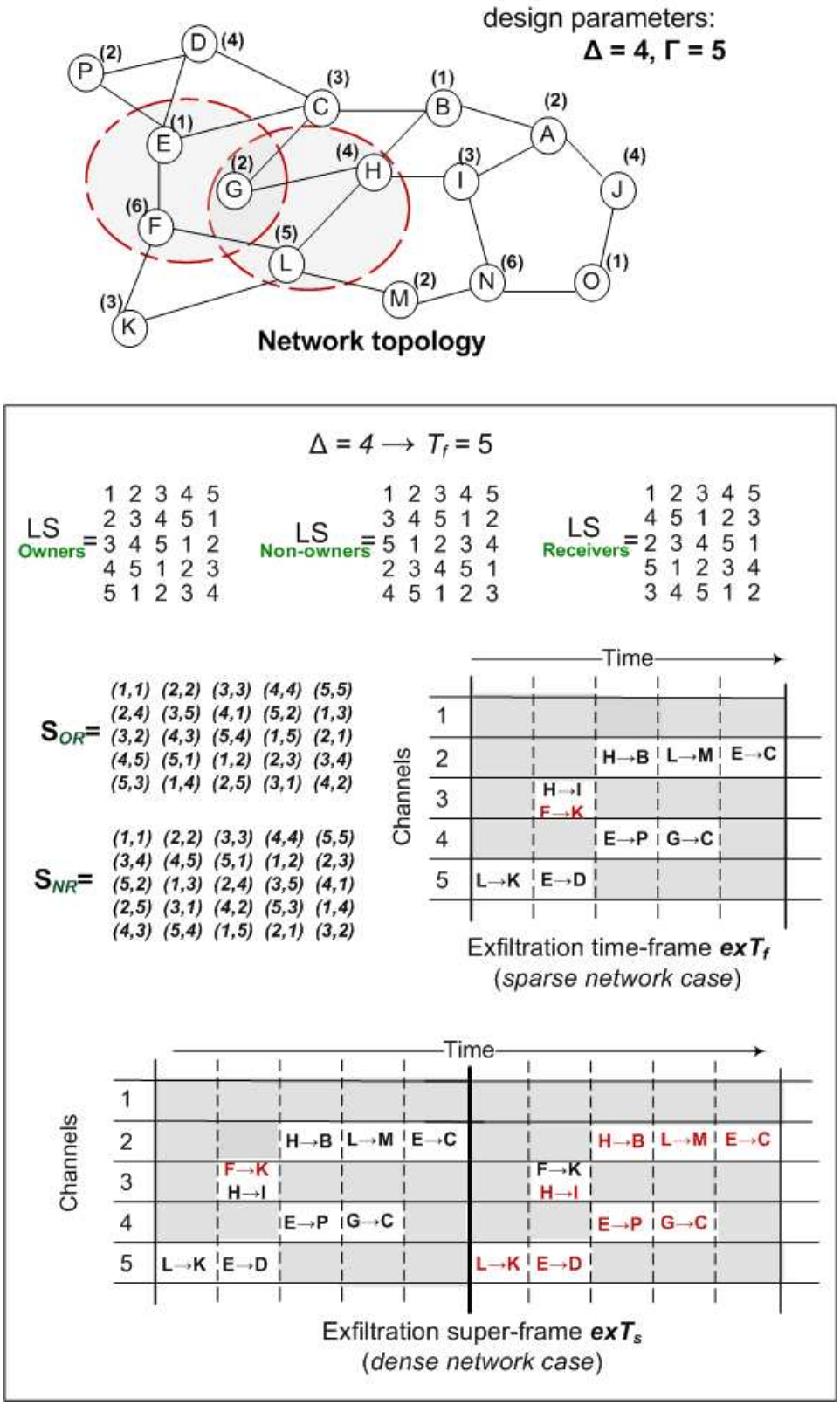


Figure 4.9: Running example under MuVC-S/R mode of operations.

square is used and hence the exfiltration time-frame in this case consists of a single time-frame. In the case of dense networks, an exfiltration transmission period consists of two consecutive time-frames, i.e. a super-frame, resulting in an **exfiltration super-frame** exT_s structure, as depicted in the figure 4.9. Coordinating listeners and receivers can also be combined with the L-VC mode. We now define the channel allocation rule under this mode.

MuVC-S/R Rule.

Let J be a subset of network nodes undergoing a communication DoS attack. During exfiltration state, each node j will assign itself a square symbol by using modular arithmetic to round its vertex-color in the range [1 - ($\triangle$+1)]. A node j_1 whose assigned square-symbol is in the range [0 - $\triangle + 1$] will be an *owner node* and hence will be using the $(\triangle + 1) \times (\triangle + 1)$ **owner's square** to create its transmission schedule, a node j_2 whose assigned square-symbol exceeds the value $\triangle + 1$ will be a *non-owner* node and hence will be using the common **non-owners square** to create its transmission schedule. Furthermore, nodes can also coordinate their listening tasks by using their assigned symbols in the $(\triangle + 1) \times (\triangle + 1)$ **receivers's square** in the following manner: if the sender is an owner node, combine the owner's square with the receivers square and find the unique assignment pair which combines the sender's symbol with the receiver's symbol (in the correct order) and listen on that assigned channel during the corresponding time-slot.

4.6 Advantages of RMBS Scheduling

The following are the advantages expected from the RMBS scheduling when reacting to a communication DoS attack.

Guaranteed Minimum Throughput.

The use of MOLS in transmission scheduling cannot guarantee complete collision freedom; however, provides a guaranteed minimum throughput for each node [63]. In our RMBS approach, we have extended this minimum guarantee on throughput and shown that, through proper choice of system parameters, collision-free scheduling is possible.

No Channel Negotiation

Channel negotiation for a data channel when packets are available to transmit can lead to significant overhead in terms of both latency and network traffic. In the RMBS approach, the need for such a negotiation is eliminated since the assignments are made based on some shared pattern amongst nodes. This leads to the elimination of such overhead in the exfiltration protocols which implements RMBS at the link layer.

No Channel Scanning

In some multi-channel schemes, transmitting nodes are required to do full channel scanning to select the best channel. The overhead of channel scanning can be high especially when many network nodes are trying to transmit at the same time. Other than in the topology transparent mode (where boundary nodes may need to perform some channel scanning to learn the transmission pattern of their attacked neighbors), RMBS approach does not need channel scanning since nodes are assigned to channels based on their vertex-colors which leads to a more robust multi-channel implementation.

No Communication Overhead

The attack mitigation affect in the case of RMBS is localized since only the part of the network undergoing a DoS attack (and its boundary region) will be performing the reactive exfiltration of the data while the rest of the network would not experience any interruption. In addition, the use of Latin square based scheduling within the attacked region minimizes overhead since the scheduling information is available to a node without any interprocessor communication and thus no communication overhead is incurred.

Chapter 5: MULEPRO: the MULti-channel Exfiltration PROtocol

This chapter presents MULEPRO, the MULti-channel Exfiltration PROtocol. MULEPRO is a fully distributed, network based protocol designed to meet the objectives and requirements set by the data exfiltration framework presented in this thesis. The design of the MULEPRO protocol is meant to constitute the upper layer of the exfiltration framework. Therefore, the MULEPRO protocol performs RMBS scheduling at the data link layer and can operate under any of the different modes of operation of RMBS. We present in this chapter two protocol variants: MULEPRO-1, which act as response against single-hop attacks, and MULEPRO-k, which provides resilience against attacks which may effect multi-hop areas. We have experimentally evaluated both protocols in a variety of network topologies and attack scenarios. Our results show that, for most types of commonly considered attack models, MULEPRO can exfiltrate a higher percentage of data as compared to other reaction strategies. Performance analysis is presented in the next chapter.

We start in Section 5.1 by stating our design objectives where we formulate the requirements under which the presented protocol is designed. In section 5.2, we define the different exfiltration algorithms used by the MULEPRO protocol and explore possible future optimizations. Section 5.3 discusses the MULEPRO's software structure and component-oriented implementation under the TinyOS environment.

5.1 Design Overview

The main objective in this chapter is to design an application level data exfiltration protocol. The basic idea is to perform rapid data exfiltration when a communication DoS attack is present. As shown in the architectural diagram of the data exfiltration framework (figure

3.3), MULEPRO functions on top of RMBS at the link layer. The incorporation of RMBS scheduling allows each node to quickly generate its own time/channel hopping pattern without the need for information exchange between neighboring nodes and with minimal overhead. As was shown in the previous chapter, RMBS guarantees that, during any time-slot, only non-interfering nodes will be allowed to transmit on the same channel. Interfering neighbors are assigned to different channels to maximize concurrent transmissions out of the attacked region. This also generates multiple exfiltration paths out of the attack region to overcome the affect of the resource-constrained attacker model (capable of blocking a subset of the channels during any time-slot).

5.1.1 Objectives, Performance Metrics and Requirements

The key design goals of MULEPRO are: 1) to achieve a high exfiltration throughput coming out of an attacked region, 2) to reduce time needed before the attacked nodes regain their connectivity with the rest of the network, and 3) to minimize communication overhead. Therefore, we define the following three performance metrics to be used when evaluating the performance with respect to the design objectives identified above:

- **Message Delivery Ratio (MDR)**: the ratio of the number of delivered messages at the base station to the number of messages sent by all of the network's nodes.
- **Reaction Time**: the time taken from when suffering nodes detect the presence of an attack till they regain their connectivity to the network.
- **Overhead**: the protocol's communication overhead counted as the number of control packets.

These metrics reflect the efficiency of the data exfiltration protocol design and the underlying reactive broadcast scheduling scheme. A good exfiltration protocol design should achieve high message delivery ratio, low overhead and low reaction time.

With this background, the following defines the set of **requirements** the protocols will be designed to meet:

1. **Flexibility to operate with different number of available channels.** The number of channels may be different depending on the network's underlying hardware. The flexibility to operate with different number of channels is, therefore, a key requirement.

2. **Maintain the network's transmission rate requirement.** During the exfiltration process, attacked nodes continuously transmit their data packets to boundary nodes during each time-slot. An important protocol requirement is that the data rate coming out of the attacked region must not be less than the network's expected transmission rate. This may result in multiple copies of the same data packet reaching the neighbors. A receiving neighbor can simply drop a packet if it notices that this packet is a duplicate.

3. **Support for offline scheduling.** When the network's common channel becomes unavailable due to an attack, communication between nodes will be disturbed and attacked nodes may become completely blocked. Motivated by this constraint, the exfiltration protocols need to be carefully designed such that not to require any control message exchange once an attack has occurred. Attacked nodes should be able to communicate amongst them selves (on other channels) collision-free based on prior knowledge of the network topology and without the need for any further exchange of any control packets. Furthermore, since the scheduling algorithm assumes knowledge of the total number of packets and the inter-arrival times of these packets, scheduling can be successfully performed offline.

4. **Maintain Network Connectivity.** When the network is under attack, the exfiltration protocols must maintain connectivity between attack region and the rest of the network. Regular network nodes will continue using the common channel for their communication. In the absence of knowledge of the attack extent, it is prudent for boundary nodes to remain connected with both the attack region (on the extra channels) and the rest of the network (on the common channel). Hence, boundary

nodes will be switching their channels back and forth, between the network's common channel and other channels, to listen to data coming from their attacked neighbors.

In addition to these requirements, the primary goal of the protocols in this thesis is to improve network resilience to communication DoS attacks. Other performance metrics, such as energy savings, are therefore not explicitly considered. The state transition diagram of the exfiltration protocol is presented next.

5.1.2 State Transition Diagram

Figure 5.1 shows the state transition diagram of the MULEPRO data exfiltration protocol. Initially, network nodes start in the network's *set up* phase which runs only once and does not run again unless a significant change in the network topology occurs (since we assume a static network model). Nodes start out in the *discovery* state since neighbor discovery is one of the first steps in the initialization of a network of randomly distributed nodes. This starts with a probe of neighboring nodes and perform neighborhood observation and discovery for some epoch of time. From the perspective of the individual nodes, this is the process of determining the number and identity of network nodes with which direct communication is feasible given the networkś transmission power and link performance. At the end of the epoch, each node obtains a list of its one-hop neighbors and their neighbors. Therefore, a node keeps track of all neighbors up to two-hops away. In addition, a global time-slotting mechanism is performed to allow for the needed time synchronization during TDMA mode. A node then switches to the *coloring* state in which it performs a distributed vertex-coloring algorithm. The type of vertex-coloring algorithm can be distance-1 or distance-2 depending on the underlying RMBS mode. Once the coloring algorithm diverges, the network setup phase is completed and nodes are ready to start their *normal state* operations.

Normal operation includes 3 different states, namely, active, sleeping, and send/receive, as shown in the diagram. When a node is in the *active* state, the node will be performing its sensing task and generating data samples. In addition, a node continues performing neighborhood observation via the continuous sending/receiving of `HELLO` beacons (the `HELLO`

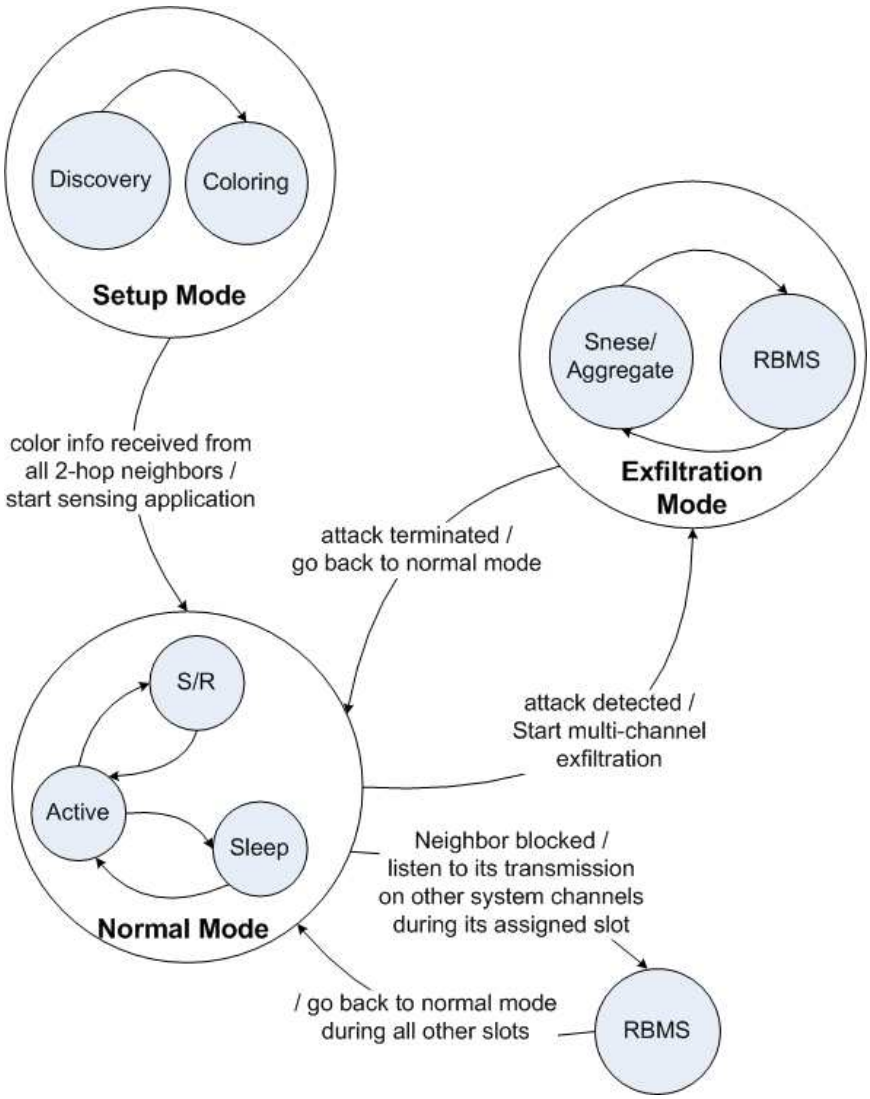


Figure 5.1: State transition diagram for the data exfiltration framework.

beacons are necessary for the *attack detection module* to be able to monitor 1-hop neighbors). When a node has data to forward back to the base station, it switches to the *send/receive* state and perform the packet transmission. Nodes also perform multi-hop relay of neighbors packets while in this state, and manage their sleep cycles by switching to the *sleeping* state accordingly.

Another state of operation is the *exfiltration state*, and this is where the data exfiltration framework is executed. The switch in state from normal to exfiltration is performed when the *attack detection module* returns positive results and hence the exfiltration state

is *reactive*. Nodes will be exfiltrating their data concurrently on the extra channels via the facilitation of of RMBS scheduling at the link layer. We further distinguish three types of nodes in this state: an **inner node** whose all one hop neighbors are also attacked , an **outer node** which has at least one boundary neighbor, and a **boundary node** which lies within a 1-hop distance from the attacked region. Such nodes will be performing their exfiltration tasks as follows: an inner node will be transmitting all the time, an outer node will be transmitting to boundary neighbors and also listening to transmissions coming from inner neighbors, and a boundary node will only be a receiver during a *soft* exfiltration state. This soft state is to allow the boundary node to be working in a dual mode at the link layer by switching between CSMA and TDMA as needed. Attacked nodes will be working in an RMBS TDMA mode at the link layer for the duration of the attack. Table 5.1 provides more details about the anticipated system states and their communication tasks.

5.2 Exfiltration Algorithms

MULEPRO is designed as a general-purpose protocol with a dedicated architecture for data exfiltration. It acts as a sublayer at the application layer level and supports a dual mode at the link layer. Therefore, it switches between CSMA and RMBS as needed. In addition, MULEPRO defines *two* common Latin squares, shared by all nodes, such that one square (SSquare) is used by the sending nodes and the other square (RSquare) is used by the receivers. Furthermore, it is capable of supporting different modes of RMBS scheduling. In its most basic form, MULEPRO performs *coordinated* scheduling.

To further enhance its performance beyond RMBS, MULEPRO performs a schedule rotation mechanism such that new square assignments are routinely created. The use of different common squares in subsequent time-frames will provide for a **transmission rotation mechanism** of the different node/channel assignments during the different time-frames, as opposed to repeating the same pattern when only a single common square is used. In addition, nodes will be granted access to the different channels and this will increase the

Table 5.1: System's states.

State	Communication Task
`sleeping`	-
`discovery`	1) after booting up, listen for neighbors' broadcasts. 2) after σ beacons, broadcast initial coloring info. 3) when all 2-hop neighbors are discovered and all their initial color info is received, switch to `coloring state`.
`coloring`	1) perform the distributed vertex-coloring algorithm. 2) when color values from all 2-hop neighbors are received, switch to `normal state`.
`normal`	1) transmit sampled data every λ_r seconds. 2) send/receive `HELLO` beacons from 1-hop neighbors. 3) relay multi-hop packets. 4) if the *attack detection module* returns positive results, switch to `exfiltration state`.
`exfiltration`	- if **attacked node**: 1) for the period of $\sigma/2$ time frames after attack detection, do: - transmit `ALERT` signal as a local broadcast on the channel number equal to my color. - during other time-slots, listen on the different channels to identify the extent of the attack. - if 1-hop all neighbors are also attacked, transmit `BLOCKED` signal as a local broadcast. 2) if no boundary neighbor, switch state to `exfiltration-1`. 3) if some neighbors are not transmitting on the extra channels (i.e. boundary neighbors), switch state to `exfiltration-2`.
	- if **boundary node**: switch to state `exfiltration-3`.
`exfiltration-1 (inner node)`	1) start the exfiltration process by transmitting data packets all the time, on the different channels, according to Latin square assignments 2) when the *attack detection module* returns negative results, go back to `normal state`.
`exfiltration-2 (outer node)`	1) start the exfiltration process by transmitting/receiving self data and data coming from inner neighbors according to the created schedule which is based on the MOLS assignments 2) when the *attack detection module* returns negative results, go back to `normal state`.
`exfiltration-3 (boundary node)`	1) if one of my 1-hop neighbors is attacked, start receiving its data on the different channels assigned for its transmission by the MOLS schedule. 2) perform `normal state` operations during other times, by switching back to the common channel. If after σ beacons no data is received, go back to `normal state`. 3) when the *attack detection module* returns negative results, go back to `normal state`.

probability of a node transmitting on an *un-attacked* channel, in case the attacker is targeting multiple system channels. In addition, such an approach will also increase channel access fairness when the number of channels is less than the neighborhood size.

Hence, at the end of a transmission time-frame, nodes regenerate their schedules using a *different* common square from the same family according to some shared sequence. A secret pseudo-random generator is used by the different nodes so that the square generation step will result in the same common Latin square in each time-frame, where a key shared by all nodes is used. Such a rotation mechanism will increases MULEPRO's resilience against multiple attackers and will also lead to increased security against an attacker who may be hopping from one channel to another trying to find the channels which are being used by the exfiltration protocols and then targeting those channels.

MULEPRO adopts a static approach when deciding a time-frame length which fits the local neighborhood of a node, and also when constructing the MOLS family. A more robust implementation would separate the MOLS family generation and handling from the actual working of the MULEPRO, as was shown in the architectural diagram in figure 3.3. In the diagram, a separate module (called the *MOLS management module*) is included such that MOLS construction and handling is optimized.

MOLS Management Module. Separating the task of the MOLS creation and maintenance from the actual operation of the data exfiltration protocol leads to a more robust exfiltration architecture since it allows for dynamic MOLS scheduling. Two critical issues have been identified for applying MOLS scheduling in ad hoc networks: (1) the determination of the order of the MOLS family and (2) the distribution of the Latin square symbols to the different network nodes [66]. In MULEPRO, the second issue is automatically resolved since vertex-colors are directly mapped into square symbols. The determination of the order of the MOLS family is directly dependent on the number of resulted colors of the system. The use of the suggested MOLS management protocol will provide MULEPRO with the robustness needed so that the determination of the order of the MOLS family

(represented by the p value) is optimized. Based on the derived p value, the MOLS family can then be generated using any one of the many algorithms proposed in the literature [64–66]. In addition, channel access fairness becomes an issue here since some vertex-colors may have better chances to access a channel in the case where a $p \times p$ square is trimmed into a $\Gamma \times P$ rectangular. To maximize channel access fairness and efficiency, the minimum Latin square order should be chosen depending on the actual topology, and such a decision is therefore considered a dynamic attribute to be determined when a node switches its state to exfiltration operations.

Protocol Variants. In this dissertation, we evaluate the performance of two different variants of the MULEPRO protocol: MULEPRO-1 and MULEPRO-k. MULEPRO-1 is intended as a defense against attacks which affect a single-hop area and executes RMBS scheduling under an interference-aware L-VC mode (section 4.3). The vertex-coloring step in L-VC mode considers direct interference only. MULEPRO-1 considers secondary interference by performing distance-2 vertex-coloring which provides it with resilience to the hidden terminal problem. MULEPRO-k provides a defense against attacks which extend beyond the one-hop radius by working under the MuVC-S/R mode at the RMBS layer. Therefore, it implements the notion of a *logical* exfiltration time-frame (exT_f) as presented in section 4.5.2.

5.2.1 MULEPRO-1

Since the purpose of this protocol is to react against single-hop attacks, the participating nodes can be defined as either attacked nodes or boundary nodes. Attacked nodes will be transmitting their data packet on the channel assigned to their their vertex-color during each time-slot, with no listening tasks. They continuously transmit their data packet to their neighboring nodes; this may result in multiple copies of the same data packet reaching the neighbors. A receiving neighbor can simply drop a packet if it notices that this packet is a duplicate. To decide whether or not an exfiltrated packet is a duplicate, a node checks

an associated sequence number. The schedule generation algorithm implemented by an attacked node is given in algorithm 1.

Algorithm 1 Pseudo-code for an *inner* node's exfiltration algorithm.

```
1. while (JAMMED(COMMON_CHANNEL)==true) do
2.   SSquare,RSquare ⇐ new matrix assignments from
3.   the MOLS family, using a pseudo-random sequence
4.   for t=0 to (T_f - 1) do
5.      for c=1 to Γ do
6.         if (SSquare[c][t]==MY_COLOR) then
7.            { -switch to channel c
8.              -broadcast pkt^t_self }
9.      END for c
10.     perform sensing task
11.  END for t
12. END while
```

In each time-slot, a total of Γ simultaneous transmissions are assigned (Γ is the number of available channels), as indicated by the inner for loop (lines 5-9). An attacked node in this case is using the sender square $SSquare$ to create its transmission schedule (line 6) and hence will be a sender at all times (to achieve maximum exfiltration of its data). Note that in the case of multiple channels being attacked, it is possible for a node to be assigned to transmit on an attacked channel. Therefore, the total number of successful transmissions per slot is equal to $\Gamma - \gamma$, where γ is the number of attacked channels. This means that some of the assigned transmissions in line 8 may result in collisions and never reach their destinations. This is not a limitation, since attacked nodes will be transmitting their data (pkt^t_{self}) multiple times. At the end of each round, a new assignment is chosen from the MOLS family for the senders' square and another for the receivers' square which would change the channel assignments for each node.

Nodes that lie just outside the attack region are considered *boundary* nodes and must also participate in the exfiltration protocol. A node that believes it may be a boundary node (by missing `HELLO` beacons from one or more of its neighbors) checks to see if an attack is, in fact, present by testing to see if any communication is going on other channels. If the test returns positive, the node declares it self as a boundary node of the attacked region

and immediately start receiving the data packets coming out of the attacked region. The schedule generation algorithm implemented at a boundary node is give in algorithm 2.

Algorithm 2 Pseudo-code for a *boundary* node's exfiltration algorithm.

```
1. J_NODES⇐ all 1-hop neighbors which are jammed
2. while (DoS_DETECTED (J_NODES ) != null) do
3.   SSquare,RSquare ⇐ new matrix assignments from
4.   the MOLS family, using a pseudo-random sequence
5.   for t=0 to (T_f-1) do
6.     for c=1 to Γ do
7.       if (SSquare[c][t]==j_COLOR &&
8.       RSquare[c][t]==MY_COLOR) and j ∈J_NODESthen
9.         {- switch to channel c
10.         - listen for data packet pkt^t_j from j
11.         - switch channel back to COMMON_CHANNEL
12.           to transmit the exfiltrated packet
13.           pkt^t_j back to the BS
14.         - break from for c }
15.    END for c
16.    perform sensing task
17.    if pkt^t_self is ready then fwd it to BS
18.  END for t
19.END while
```

Note how a boundary node maintains its connectivity to the network throughout the attack period by continuing the use of the common channel as its main channel of communication. At the same, it performs a channel switch only during the one time-slot which has its square symbol as the receiver and the attacked neighbor node's symbol as the sender, as indicated by the if statement in line 7 (the orthogonality of the two squares will guarantee that there is only one single channel/time assignment for the 2 nodes). After a short period of time of not receiving any exfiltrated data, the boundary node will realize that it was not chosen to participate in the exfiltration process and hence will go back to normal state.

5.2.2 MULEPRO-k

In the case of an attack which affects a multi-hop region, *outer* nodes inside of the attacked region will also need to cooperate in exfiltrating packets of the *inner* nodes, in addition to their own (fig 5.2). In order to save resources and energy, data must be aggregated to

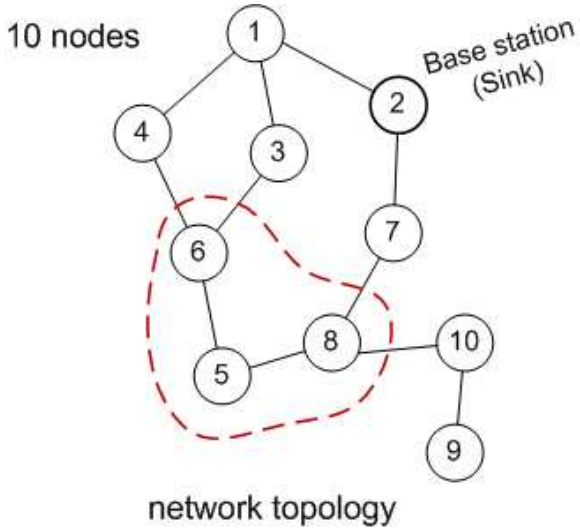


Figure 5.2: Multi-hop attack scenario: an adversary affecting a 2-hop area. Node 5 in this case is an inner node, nodes 6 and 8 are outer nodes, and nodes 4,3,7,10 are the boundary nodes.

avoid overwhelming amounts of traffic in the network. Therefore, in-network data aggregation is performed on the received packets to minimize transmission overhead and possible redundancy. To learn about the extent of the attack, nodes inside the attacked region will participate in an initial step where they will be transmitting a special `ALERT` signal on the channel number corresponding to their own color value and during the time-slot which is assigned to their vertex-color. A node also needs to scan through the different channels and listen for other attacked neighbors, when not transmitting. When a node notices that all of its 1-hop neighbors are also affected, it will declare its self as an *inner* node and will transmit a special `BLOCKED` signal. Nodes which have at least one non-blocked neighbor will declare them selves as being *outer* nodes.

A node which happens to be an inner node will be transmitting its data all the time, while an outer node will be switching back and forth between sending and receiving. Therefore, attacked nodes which lie in the outer region will need to perform multiple functions; they will need to switch back and forth between exfiltrating data in the direction of the outside region and also listen for data coming from the inside of the attacked area. This means that an attacked node which happens to be an outer node needs to decide on when to be a sender and when to be a receiver. A simple approach is to be a receiver on the slots which

have an inner blocked neighbor assigned as a sender and the outer node as a receiver to receive inner data, and then to be a sender during all other times. However, it is possible to have a transmitting blocked neighbor node and a listening boundary node during the same slot (on different channels). In this case, the node must decide whether to be a sender or a receiver during this time-slot. MULEPRO-k uses a simple heuristic to make this decision. The scheduling algorithm used by an inner node and a boundary node are the same as in the MULEPRO-1 case, while an outer attacked node will be implementing algorithm 3.

Algorithm 3 Pseudo-code for an *outer* node's exfiltration algorithm.

```
1.  J_NODES ⇐ all 1-hop neighbors which are also jammed
2.  while (DoS_DETECTED(J_NODES) != null) do
3.    SS,RS ⇐ new matrix assignments from the MOLS
4.    family, using a pseudo-random sequence
5.    for t=0 to (T_f-1) do
6.       minislot=0
7.       for c=1 to Γ do
8.         if (((SS[c][t]==j_COLOR)&&(RS[c][t]==MY_COLOR)
10.             && (j ∈ J_NODES) && (minislot==0)) then
11.                {- switch to channel c
12.                 - receive data packet pkt_j^t from j
13.                   and add it to send_QUEUE
14.                 - minislot++ }
15.        else if ((SS[c][t]==MY_COLOR) && (minislot==1)) then
16.                {- switch to channel c
17.                 - transmit the next data packet
18.                   from send_QUEUE }
19.        else if (minislot > 1) then break from for c
20.        minislot++
21.      END for c
22.      perform sensing task
23.      if pkt_self^t is ready then add it to send_QUEUE
24.   END for t
25. END while
```

Note the use of the MOLS (in line 8) where an outer attacked node wanting to listen to its inner blocked neighbor uses that neighbor's symbol in the sender square $SSquare$ and its own color in the receiver square $RSquare$, while in line 14, it decides its transmission using its own color as a symbol in the SSquare. Also note that a time-slot is further divided into 2 mini-slots, one for listening to data coming from inside of the attack region, and another for the exfiltration of the received data (plus self data) to the outside of the region (lines

7-20). This division of a time-slot during exfiltration mode provides for the implementation of the *logical* exfiltration time-frame introduced in section 4.5.2. It also guarantees that an outer node will be dividing its time equally between receiving data from the inside of the attacked region and exfiltrating data to the outside region.

The fully distributed schedule generation algorithms presented above are implemented by the different nodes for the duration of the attack. These exfiltration algorithms define the operation of the three different participating node types, namely, an inner attacked node, an outer attacked node, and a boundary node. However, it is possible, for instance, for a node to be an inner node (with no boundary neighbors) and at the same time may need to cooperate in further receiving packets from another inner node. In other words, outer nodes may consist of multiple layers before reaching a boundary. We note that the algorithms presented above are also applicable in such a case, with minor modifications. Figure 5.3 shows an example in which node B (an inner node) is also participating in the receiving of its inner most neighbor's data (node K) and forwarding it along with its own data in the direction of the outside region.

5.2.3 Experimental Results: Dynamic vs Static MOLS

MULEPRO uses a static approach when deciding the order of the MOLS family in which it sets the value of p to be equal to the the frame length (T_f)ex and also when constructing the MOLS family. The rational behind such a choice for the p value is to guarantee the minimum-rate requirement for each node since T_f incorporates all the possible vertex-colors. However, MULEPRO assumes Γ channels are available and it does not consider the number of *actual* available channels (i.e. *un-attacked* channels). Therefore its schedule generation technique remains oblivious to the number of attacked channels γ. To study the effect of γ on the exfiltration throughput, we performed an experiment in where we varied the number of attacked channels γ with time and used the same constant p value.

Figure 5.4 shows the overall result for the message delivery ratio metric at the base station for a 50-node network in three different experiment, where Γ is 3,5 and 7. In each

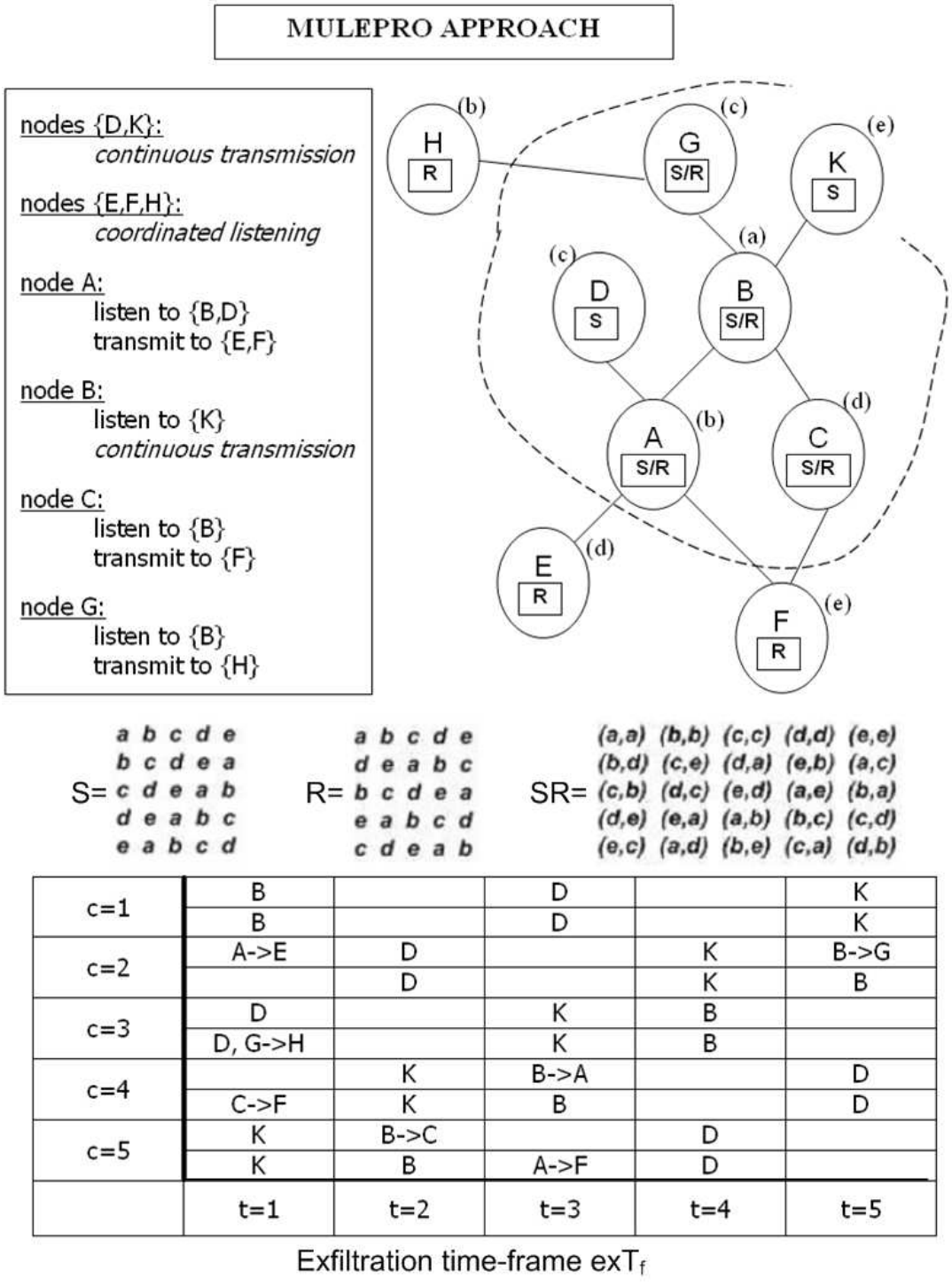


Figure 5.3: Multi-hop exfiltration under the MULEPOR protocol. Note how MULEPRO's implementation includes only two Latin squares, one for the senders and another for the receivers.

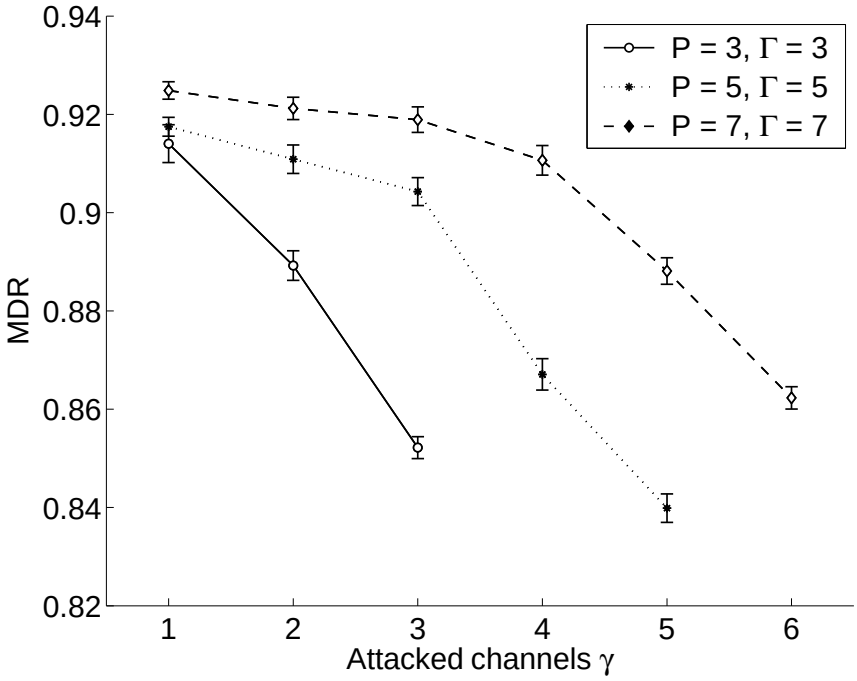


Figure 5.4: Exfiltration throughput is effected by the p/Γ^* ratio, where Γ^* represents the *actual* number of available channels.

experiments, the attacker targets a total of γ channels from the set of $\Gamma + 1$ total system channels. The result in figure 5.4 shows that the exfiltration throughput is directly affected by the Γ^*/p ratio, where Γ^* represents the number of available $un - attacked$ channels and is equal to $(\Gamma + 1) - \gamma$. Such findings also conform with the optimal MOLS construction (presented in the previous chapter), which shows that the throughput bounds are decreasing when the p values increases (equation 5.1). We can rewrite the throughput bounds in this case as:

$$\Omega_{min} = \frac{\Gamma^* - \triangle}{p} \tag{5.1}$$

$$\Omega_{max} = \frac{\Gamma^* - max(\triangle + 1 - p, 0)}{p} \tag{5.2}$$

This concludes that to optimize the exfiltration throughput, the MOLS construction needs to be done dynamically during attack time so that the Γ^* value is considered, as opposed to a static assignment done during the network setup phase. Such an optimization

function needs to be separated from the actual working of MULEPRO so that the MOLS family is also created dynamically, depending on the number the available, un-attacked channels (Γ^*). We leave this as a future extension to further optimize the performance of MULEPRO.

5.3 TinyOS Components and Software Architecture

In this section, we present the software structure of the MULEPRO protocol using the component-oriented of the TinyOS environment. Experimental evaluations under the TOSSIM simulator are presented in the next chapter.

TinyOS is an operating system designed to run on distributed embedded WSNs. Programming in TinyOS requires the use a special language known as *network embedded systems C (nscC)*. NesC is a C-based programming language which provides an event-based execution model to support the event-driven concurrency model of the TinyOS environment. It also constitute a special type of programming known as *component-oriented programming* [97].

A nesC program consists of several "components" where default and user's components are defined and included as-needed in the programs. A Component *provides* services and *uses* other component's services. Components in the nesC programming language have similarities to objects: they encapsulate state and interact through well-defined interfaces [98]. They also have significant differences: the set of components and their interactions are fixed at compile-time (to promote reliability and efficiency), rather than at run-time, as object-oriented references and instantiation do [98].

A significant challenge in TinyOS development is the creation of flexible, reusable components. In the MULEPRO TinyOS architecture, we focus on modular design and reusable components. Figure 5.5 shows the component diagram of the MULEPRO protocol which reflects its software implementation architecture. The main idea of a component diagram is to show the different components in the system and the interactions between them. In the diagram, a components is represented as a rectangular where two different types are

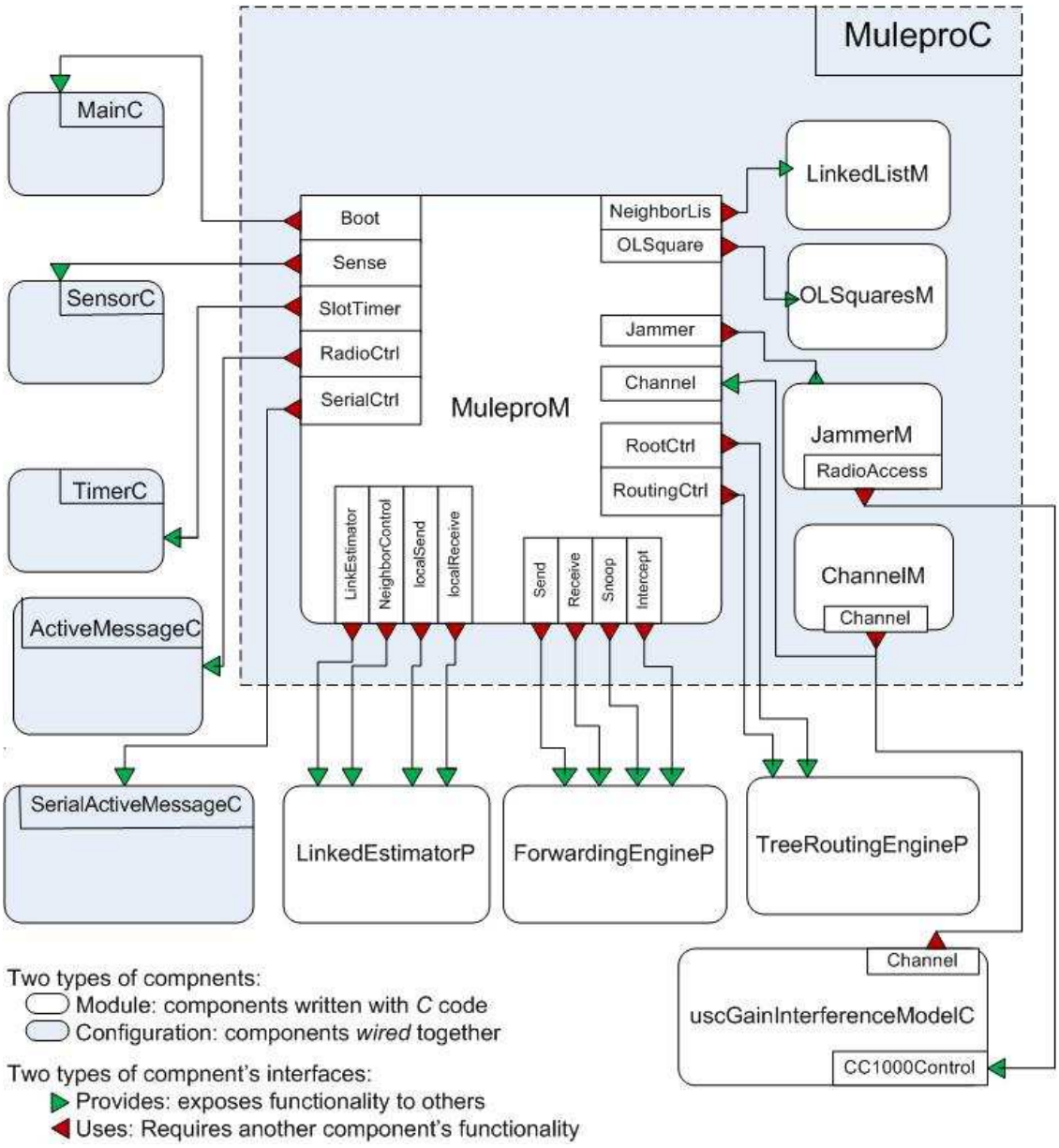


Figure 5.5: Component diagram of MULEPRO's implementation in nesC.

possible, namely, *configurations* and *modules*. A configuration component is implemented by wiring different modules together while a module is implemented with C code. In the MULEPRO's component figure, `MuleproM` represents MULEPROS's main module component which interacts with the other MULEPRO's module components, namely, `MuleproM`, `LinkedListM`, `OLSquaresM`, and `JammerM`. `MuleproM` also interacts with other external components such as `MainC`, `SensorC`, `TimerC`, and so on. These external components are generic components which are included with the TinyOS distribution [20].

Component-oriented programming supports constructing software systems by composing independent components into a software architecture. A typical architecture includes a set of components, connections between the components, and constraints on how components interact. In MULEPRO's component diagram, the interactions between the different modules are shown as part of the system's configuration component `MuleproC`. The different modules are connected through common *interfaces*. A component either *provides* or *uses* an interface. For a component to be able to invoke a functionality of another component, it needs to *wire* its "uses" interface to the other component's "provides" interface. Thus, interfaces define interactions between the different components. An interface can sometimes be *multiply wired*, such as the case in the *channel* interface which is provided by the `ChannelM` component. In this case, `ChannelM` is providing service for to the MULEPRO application and also to the underlying radio component, thereby providing channel coordination between the application and the underlying radio. In fact, the `ChannelM` component plays an important role in extending the default single-channel environment provided by TOSSIM. The following table 5.2 lists the main components used by the MULEPRO program along with their expected functionality.

5.4 Summary

This chapter presented MULEPRO, a fully distributed network based protocol designed to rapidly exfiltrate data from an attacked region using multiple communication channels simultaneously. The basic idea is to assign scheduling symbols to the different nodes based on

Table 5.2: The MULEPRO software:major components and their functions.

Component	**Type**	**Category**	**Role**
MuleproC	configuration	user	provides the MULEPRO application.
MainC	configuration	system	provides the systems initialization interface.
SensorsC	configuration	system	provides sensor's readings.
LinkEstimatorP	module	system	provides link quality estimate to and from a neighbor, also implements a beaconing protocol to keep track of neighbors and therefore incorporates attack detection at neighboring nodes (i.e. determining a *boundary* status).
TreeRoutingEngineP	module	system	building and maintaining the spanning tree from each node to the sink.
ForwardingEngineP	module	system	provides routing layer functionality; uses the tree built by the TreeRoutingEngineP component to route packets back to the sink; capable of incorporating attack detection.
JammerM	module	user	models a jammer; this component is meant to replace the packet-level radio components which implements the CSMA algorithm in the TinyOS radio stack.
ChannelM	module	user	provides channel switching capability.

their vertex color to produce conflict-free transmission schedules between sensor nodes, up to 2-hop distance in order to address the terminal hidden problem. The protocol is designed to operate with any number of channels. Software structure of the MULEPRO protocol was also presented using the component-oriented approach implemented by TOSSIM. Experimental evaluations of MULEPRO is presented in the next chapter.

Chapter 6: Simulation and Performance Evaluation

This chapter presents the performance analysis of the data exfiltration framework methods and techniques presented in the previous chapters. We evaluate their performance via extensive simulation and experiments done in TOSSIM, the standard network simulator for the TinyOS environment. TOSSIM simulates the TinyOS sensor nodes by abstracting their hardware components and modeling them in software instead. An advantage for using the TOSSIM simulator is that TinyOS code written for real hardware platforms can be run in TOSSIM with little change, and vice versa [42]. We use the synthetic network modeling software provided by TOSSIM to generate random networks. We present simulation results to demonstrate the performance of the data exfiltration defense under a variety of attack configurations, where we compare MULEPRO's performance with other existing approaches through simulation and experiments.

We start this chapter by describing our simulation model in section 6.1. We then describe our simulation methodology in section 6.2. Section 6.3 presents our experiments and shows the results. We then provide a discussion of the results in section 6.4.

6.1 Simulation Communication Model

Rather than providing a specific radio model, TOSSIM provides a few low level primitives that can express a wide range of radios and behaviors [19]. The default TOSSIM radio model is single-channeled and gain-strength based, which considers signal strength of transmission and propagation and includes a clear channel assessment (CCA) functionality. This section provides an overview of TOSSIM's radio communication models. In addition, we explain our methodology in extending TOSSIM's default single-channel communication environment to allow for multi-channel communication.

6.1.1 The Radio Communication

TOSSIM allows for specifying a network topology in terms of propagation gain values, where the network is modeled as a directed graph with each vertex representing a node and edges representing connections between nodes. There are two radio models in TOSSIM: simple radio model and lossy radio model. The simple radio model assumes that all nodes are in a single cell and can hear each other perfectly, hence assumes no bit errors are possible in transmissions. The use of the propagation modeling leads to a lossy radio model. In the lossy model, each edges has an associated bit error rate (also known as a propagation gain value) that models the probability that a bit can be corrupted if sent along that link. Such loss modeling is derived from empirical measurements of a certain platform (such as MICA or MICA2) and a file that contains the bit error rates of all possible links of a network, called a loss model, is passed to TOSSIM as an input.

The simulation and experiment code performed in this chapter was based on the version included in the TinyOS 2.x distribution included in [20]. In [20], a user defined simulation environment is decided through a scripting interface and includes the following parameters:

- **size**: network size defined by the number of nodes in the network.
- **connectivity**: represented by the propagation gain values between the different network nodes in dBms.
- **interference**: represented by the noise levels, which includes both the floor (ambient) noise and its standard deviation in dBms.

The following parameters are predefined by TOSSIM:

- **receiver's sensitivity**: 4.0 dBm.
- **CCA threshold**: -95.0 dBm.
- **local noise**: set to a random value from the uniform distribution of noise, i.e. in the range [floor noise - deviation, floor noise + deviation].

- **signal strength**: calculated as the signal strength of ambient noise plus the signal strength of all other transmissions.

- **transmission power**: the actual power when transmitting the bits on the radio is set equal to the RSSI value plus the gain level between the sender and the receiver. However, since the RSSI readings are hardware dependent (there are hardware variations, uneven propagation, receiver/transmitter collaboration issues, etc.), therefore, current version of TOSSIM does not have any support for RSSI and its value is ignored.

In addition to the radio propagation model, TOSSIM simulates radio frequency noise, both as an interference from other nodes as well as environmental noise. It utilizes the Closest Pattern Matching (CPM) algorithm, which takes a noise trace as an input and generates a statistical model from it. This statistical model captures bursts of interference and other correlated phenomena which greatly improves the quality of the radio frequency simulation [99]. To configure CPM, noise traces taken from Meyer Library at Stanford University (found in [20]) were used for our simulation purposes. Figure 6.1 shows the simulation's communication model as presented in TOSSIM's original documentation [99].

6.1.2 Multi-channel Communication

In our simulation model, we extend TOSSIM's default single-channel environment to facilitate multiple channel communication. In our multi-channel environment, we simulate a multi-channel single-transceiver radio model. That is, a node can either send or receive during any single time-slot but cannot perform both operations simultaneously. In addition, a node can only send and receive messages on the channel it is currently assigned on. We follow the same approach suggested by Huang in [59], and we define a new component called `ChannelM` to coordinate the multiple channel use and access which defines the following interface:

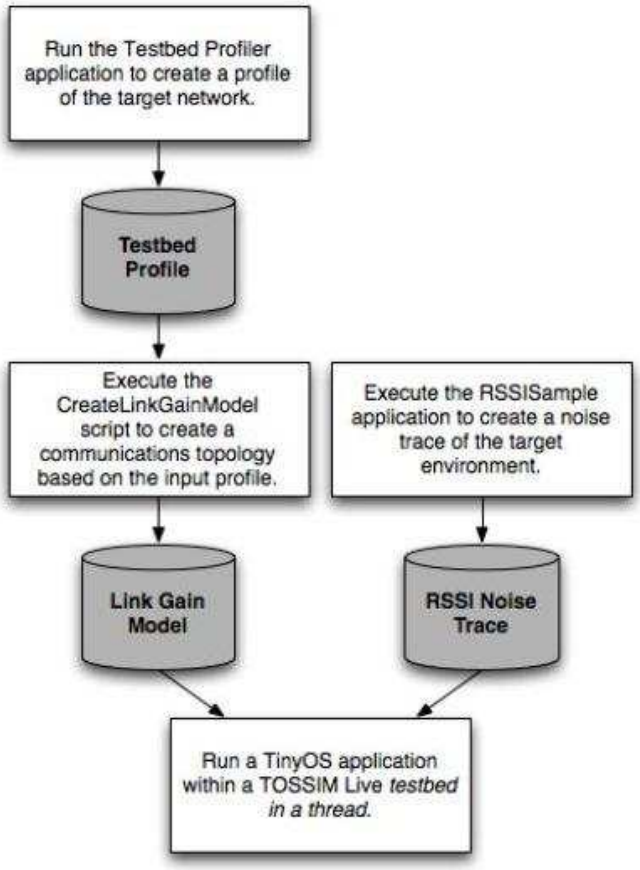
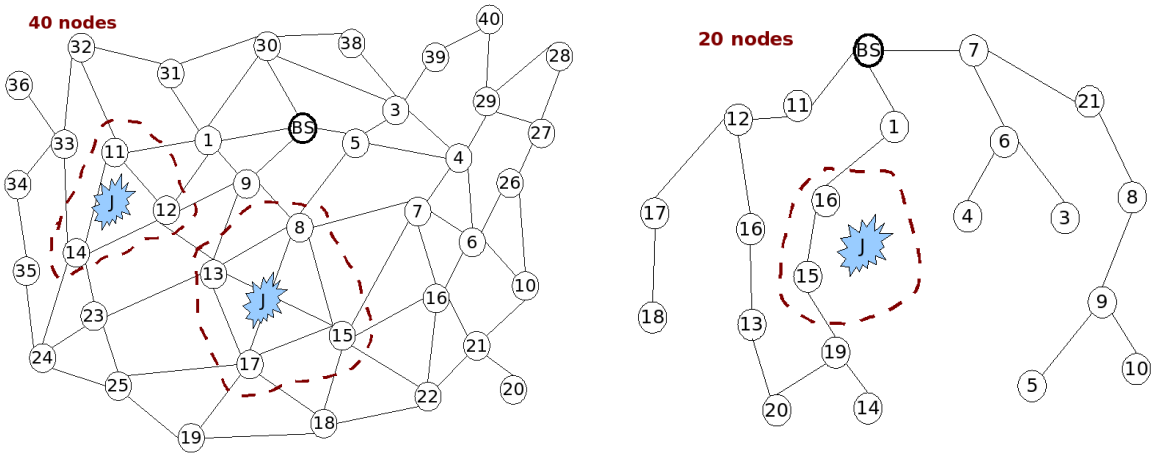


Figure 6.1: TOSSIM's radio communication model: combining radio propagation gain values and frequency noise traces (source: [99]).

```
interface Channel {
   /**
   * set the current communication channel for the mote
   */
   command error_t setCurrent(uint8_t ch);

   /**
   * returns the current communication channel a mote is listening on.
   */
   command uint8_t getCurrent();

   /**
   *a node may also report it self as being jammed here; i.e) when the
   *utility of the communication channel drops below a certain threshold
   */
   event void jammed();
   }
```



(a) multiple jammers in a 40-nodes network with a dense topology.

(b) a single jammer in a 20-nodes network with a sparse topology.

Figure 6.2: Sample simulation scenarios.

6.2 Simulation Methodology

We assume a multi-hop network environment where N nodes are uniformly distributed and randomly placed. We considered several networks of different sizes with dense topologies as shown in figure 6.2. A network is considered to have a sparse topology if the maximum nodal degree of the network does not exceed 3 neighbors, and is considered dense other wise. There is a single sink node (abbreviated BS for base station) and one or more jammer nodes. The number of attacked nodes in each experiment is set to be $20 - 25\%$ of the total network nodes, and 1-3 hops diameters. To maintain network connectivity, we ran the exfiltration protocols on top of a globally optimized topology reformation strategy. This guarantees the reconstruction of the underlying spanning tree in case a parent is no longer reachable. In conducting all experiments, we assume that there will always be at least one channel by which an inner attacked node will be able to exfiltrate its data on, and also at least one channel by which an outer attacked node will be able to communicate with a boundary neighbor. We also assume that network connectivity is always maintained and that an attack will never result in a partitioned network.

A time-slot represents the default sensing period and is set to 40 msec. The exfiltration

time-frame length is set to 200 msec and this value also represents the sampling rate required by the network (which result in the packet generation rate of 1 packet every 200 msec). The network topology is assumed static and the radio range of all nodes are assumed to be the same, R_{tx}. Nodes are placed according to a uniform random distribution such that an attacker node is always placed more than R_{tx} distant away from the sink. Such a setting is needed to guarantee that an attack will not have any direct effect on the sink.

6.2.1 Attack Modeling and Detection

To simulate attacks, we created a special component in TOSSIM called *JammerM*. This component is intended to override the default TinyOS packet-level radio component that executes the CSMA algorithm. If a node is defined as a jammer, it will have the functionality of being able to have direct access to the radio thereby keeping the channel busy (by the continuous submission of a disrupting signal on the targeted channel) for the duration of the attack. The jammer component is also capable of attacking multiple channels at the same time.

To simulate attacks, a jammer node is activated after the sensor network is completely deployed and neighbor discovery between the different nodes is done. Therefore, a node executing *JammerM* remains quiet when it initiates and appears invisible to neighboring nodes during the discovery and the vertex-coloring phase. It then starts performing attacks on the different channels at different times according to a predetermined pseudo-random sequence. Jammer nodes are placed in the same region targeting different channels, or in different regions targeting the same or different channels. We considered 2 types of jammers: a *fixed jammer* that jams the same channel(s) and a *hopping jammer* that hops between channels.

Attack detection, in general, can be performed both at the physical layer and the data link layer. At the physical layer, signal strength measurements are usually used. For instance, two nodes i and j are able to communicate if the value of of $RSSI_{(i,j)}$ is greater than some threshold value $RSSI_{min}$ [72]. Since RSSI has no support in TOSSIM, the data

link layer detection approach was used in our simulation model and we specifically used the PDR [16] metric. This metric monitors the ratio of *successfully* received packets by the total number of received packets. Thus, it allows for the detecting of an attack at the receiver side, by using CRC check results of all incoming packets.

6.2.2 Dual Mode at the Link Layer

The underlying MAC paradigm in TOSSIM is based on a CSMA scheme. In our simulation model, the MAC layer behaves normally in a CSMA mode, however, when an attack is present, the MAC layer switches into a TDMA mode to perform the multiple channel communication. Figure 3.2 showed the design of the data link layer in the data exfiltration framework, in which RBMS provides the underlying mechanism at the link layer. The main objective of the RMBS approach is to act as a deterministic time division channel access to guarantee collision-free, concurrent transmissions out of an attacked region. Therefore, RMBS works on top of a TDMA scheme. We therefore perform a global time-slotting mechanism during the network setup phase to establish time synchronization to provide for RMBS at the link layer, when needed. In other words, attacked nodes will be operating in a TDMA mode all throughout the attack period. A boundary node is capable of communicating with both attacked and other normal network nodes and hence will be switching back and forth between CSMA and TDMA mode at the MAC layer.

6.2.3 Performance Metrics

To evaluate the performance of our suggested protocols, we use the same metrics that were derived from MULEPRO's design objectives in Section 5.1. Table 6.1 lists those metrics and their calculation method.

6.3 Experiments

We now present simulation results of the MULEPRO protocols in various network and attack scenarios. MULEPRO is a fully distributed network based protocol designed to

Table 6.1: Metrics used in the protocols' evaluation.

Metric	**Calculation Method**
MDR	MDR is calculated by dividing the total number of packets received at the base station by the total number of packets successfully transmitted from all nodes.
Overhead	Overhead is captured by the number of control packets exchanged between nodes.
Reaction Time	Time to react is equal to the number of time-slots needed before attacked nodes regained connectivity to the base station.

rapidly exfiltrate data from an attacked region using multiple communication channels simultaneously. The protocol is designed to operate with any number of channels and has two variants, namely, MULEPRO-1 and MULEPRO-k. MULEPRO-1 uses RMBS at the link layer and operates under the L-VC mode and hence it is effective against single-hop attacks. MULEPRO-k provides a defense against multi-hop attacks and uses RMBS under the MuVC-S/R mode. We note that all of the simulated attacks extended to 1-3 hop areas and we therefore use MULEPRO-1 and MULEPRO-k appropriately. We also simulated a different defense approach that is based on the **channel surfing** protocol (which involves finding a new communication channel for the network) as presented in [67], as a point of reference. The network scenarios and parameters considered in all simulation experiments are identical when evaluating the different protocols. We considered configurations with the number of available channels Γ (besides the *common channel*) is equal to 3, 5, and 7 (denoted as MULEPRO-k/Γ).

We ran experiments for 10,20,30,40, and 50 nodes, with each experiment lasted for 6000-10,000 seconds. Each experiment was run at least 20 times with different random number seeds. 95% confidence intervals were obtained for the average values reported and are shown in the figures. We introduced jammer nodes at different regions of the network

using uniform distribution, and jamming effects 20-25% of network nodes.

6.3.1 Metric 1: Message Delivery Ratio (MDR)

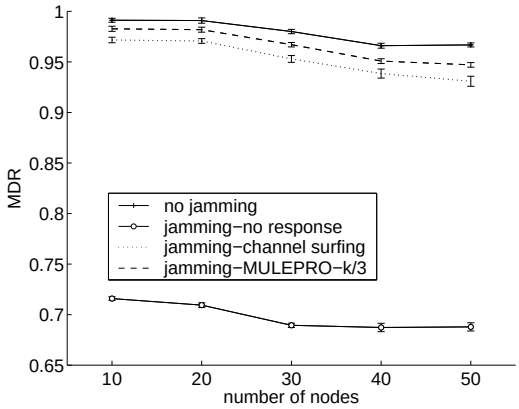
In our first set of experiments, we investigate the impact of various approaches, including MULEPRO, on the overall message delivery ratio. The results presented here show MDR for the entire system, not just the attack region. The purpose of this experiment is to show the overall impact of the attack represented by the delivery ration at the base station. The experiments examined two different attack scenarios, single-channel jamming and multi-channel jamming.

6.3.1.1 Single Channel Attacks.

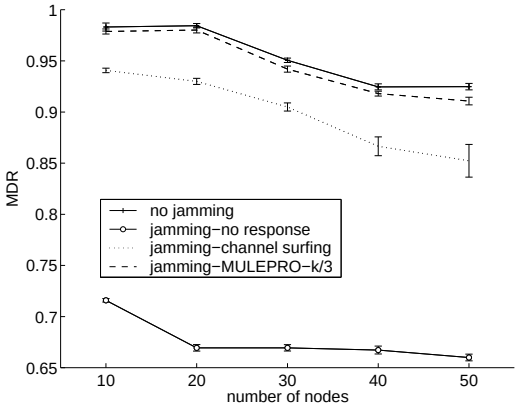
In these experiments, we looked at the jammer targeting a single channel at a time in a multi-hop area, in the same or different regions. In figure 6.3, we report on the performance results for the MULEPRO-$k/3$ protocol and the channel surfing approach and compare them with the case when the network did not experience any jamming (as represented by the "no jamming" line) and also when the network experienced the same exact attack but without any reaction to the attack (represented by the "jamming-no response" line in the figures).

As expected, the MDR values are lower when the jammer is hopping since the jammed region will be experiencing even more packet loss due to occasional loss of communication. We also see that MULEPRO-$k/3$ significantly improves MDR in all four scenarios. In addition, while the use of channel surfing results in lower MDR values in the case of dense networks (this is because of the flooding of the *switch-channel* message throughout the whole network), the use of the MULEPRO-k in denser networks results in even higher MDRs. This is because denser networks will maximize multiple channel use. Further, as node density increases, the probability of having a non-jammed neighbor also increases.

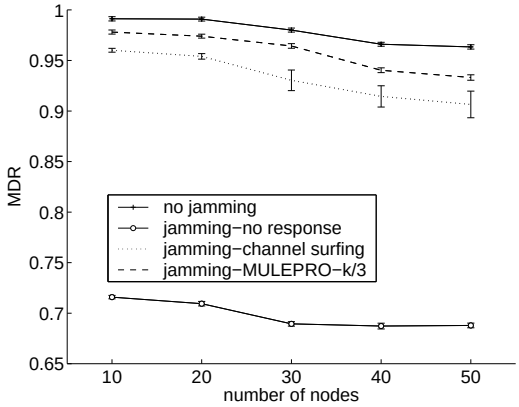
While the shown diagrams present results for the MULEPRO-$k/3$ protocol, the MDR values for the MULEPRO-$k/5$ and MULEPRO-$k/7$ have resulted in even higher MDR



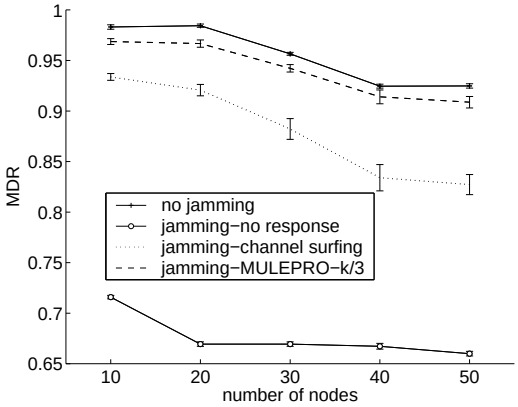
(a) MDR at the BS for the case of a fixed jammer - sparse network.



(b) MDR at the BS for the case of a fixed jammer - dense network.



(c) MDR at the BS for the case of a hopping jammer - sparse network.



(d) MDR at the BS for the case of a hopping jammer - dense network.

Figure 6.3: MDR experiments results.

values (not shown on the above diagrams for clarity). In fact, the difference between the MULEPRO-7 results and the case where the network did not experience any jamming was *statically insignificant*, which lead us to conclude that *MULEPRO-k/7 leads to complete resilience against jamming* in this case.

6.3.1.2 Multiple Channel Attacks.

In this set of experiments, we studied the effect of multiple jammers affecting two or more channels (or a single jammer attacking multiple channels), in the same or different regions. We note that the channel surfing approach is not applicable in this case since it does not

Table 6.2: MDR results for the case of multiple jammers affecting multiple single-hop regions.

γ	1	2	3	4	5	6
no jamming	92%	92%	92%	92%	92%	92%
no response	66%	66%	66%	66%	66%	66%
channel surfing	84%	–	–	–	–	–
MULEPRO-1/3	91%	88%	85%	–	–	–
MULEPRO-1/5	92%	91%	90%	87%	84%	–
MULEPRO-1/7	92%	92%	92%	91%	88%	86%

provide resilience against multiple jammers.

Single-hop Regions. In our first experiment under this scenario, we model a hopping jammer attacking multiple channels in a single-hop area. The number of nodes in this experiment was fixed to 40 nodes and the number of available channels Γ is set to 3,5,7. We vary the number of jammed channels γ from 1 up to Γ-1. The different jammers were set to start at different times, attacking different channels in different regions. The results of the MDR for the dense network topology are stated in table 6.2. The results indicate that a delivery ratio of 84% is acheivable *even* with only 1 channel available. We note here how the results when only 1 channel is available for the MULEPRO-1 protocol are lower bounded by the channel surfing protocol indicating the fact that both protocols are operating on one single channel in this case.

Multi-hop Regions. To see the effect of multiple jammers in the case of multi-hop attacks we perform the following experiment. We use different network sizes and we set the Γ value to 3,5,7. Again, the multiple jammers affected different regions of the network and incorporated some hopping strategy. We fix the number of jammed channels γ to 3 to note the effect of the multiple jammers on the different network sizes. The results for the dense networks are depicted in figure 6.4. Again, notice how the channel surfing approach is not applicable in this case. In addition, MULEPRO-k with higher number of available channels resulted in higher MDRs at the base station. In fact, for the case of MULEPRO-k/3, there seem to be a degradation in the protocol performance, this is due to the fact that the

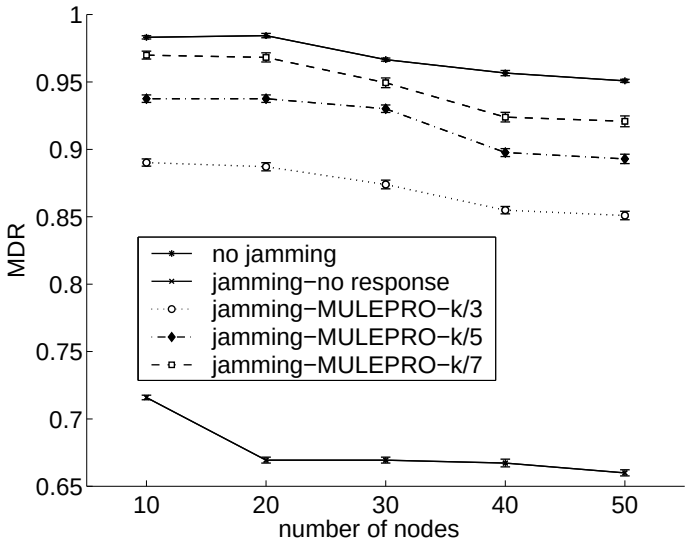


Figure 6.4: MDR results for the case of multiple jammers affecting a multi-hop region.

protocol in this case is working on a single channel only. to note the effect of the multiple jammers on the different network sizes. The results for the dense networks are depicted in figure 6.4. Again, notice how the channel surfing approach is not applicable in this case. In addition, MULEPRO-k with higher number of available channels resulted in higher MDRs at the base station. In fact, for the case of MULEPRO-$k/3$, there seem to be a degradation in the protocol performance, this is due to the fact that the protocol in this case is working on a single channel only.

Localized Effects. While the previous experiments showed a more global view of the exfiltration approach (number of packets received at the base station), in this experiment, we study the effect of data exfiltration on the jammed region by itself. We use the total number of exfiltrated packets as a mean to compare performance of the different protocols. The results are shown in figures 6.5. It is clear from the figure that increasing the number of concurrent channels for the task of data exfiltration will result in much higher throughput out of the jammed region.

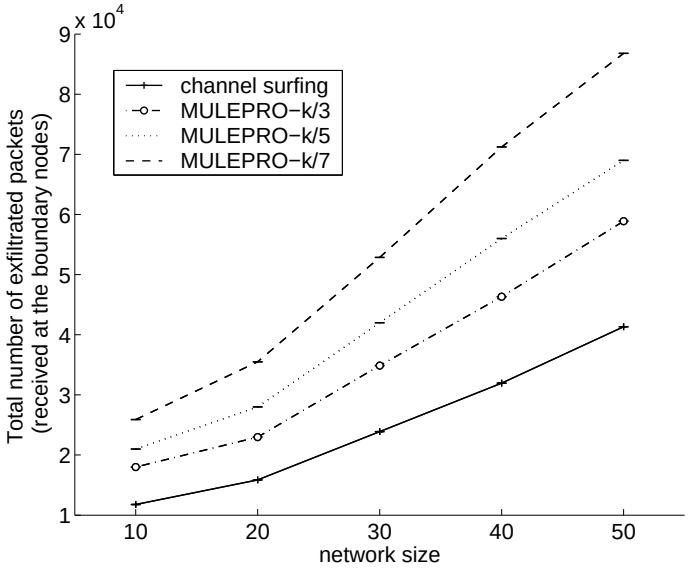


Figure 6.5: Total number of exfiltrated packets received by boundary nodes.

6.3.2 Metric 2: Protocol Overhead

The communication overhead for the case of channel surfing increases with the size of the network since it requires the use of flooding messages to initiate a channel change across the entire network. MULEPRO does *not* add communication overhead after an attack. The overhead of control messages sent are plotted in figure 6.6 for both dense and sparse networks with different topologies. As expected, the overhead for the case of channel surfing is much higher, especially in the case of dense networks, due to flooding. The overhead in the channel surfing case is also increasing with the increased network size. MULEPRO-k, however, remains independent of the network size and network topology since the scheduling algorithm executed by MULEPRO-k is distributed and needs minimal exchange of information between nodes. It is interesting to note also that the results shown here are for the MULEPRO-k/Γ since the results in this case are independent of the number of the available channels.

6.3.3 Metric 3: Reaction Time

In the channel surfing approach, the whole network has to switch to a new channel before nodes are reconnected again. In MULEPRO only jammed and boundary nodes switch their

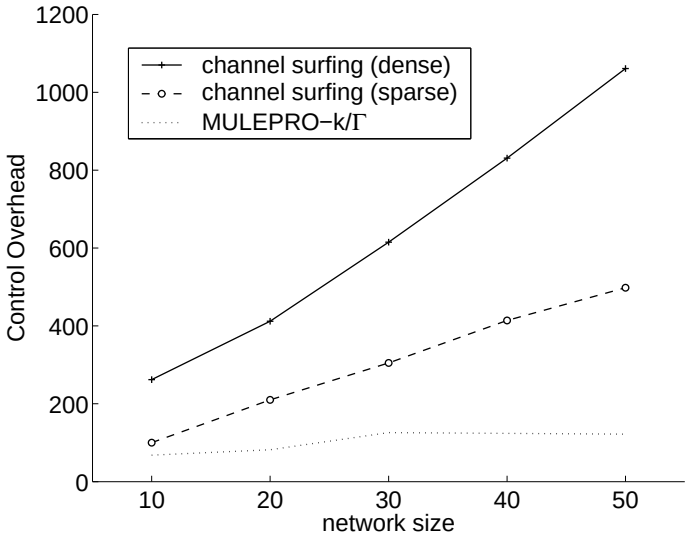


Figure 6.6: Control overhead increases in dense topologies for the channel surfing approach but remains about the same for MULEPRO.

channels and the rest of the network remains unaffected. Figure 6.7 represents the time series of the number of time-slots needed before jammed nodes regained their connectivity to the base station.

The plot shows that the jammed nodes in the MULEPRO case, when compared to the channel surfing approach, resumed their normal packet delivery performance in fewer time-slots and all other nodes of the network were not effected much by the jammer. Note that for both protocols this time includes the waiting time needed by a node prior to deciding whether it is a jammed or a boundary node. Nodes inside the jammed region can detect jamming much faster than boundary nodes.

In all experiments we ran, the time it takes a jammed node to detect jamming was almost constant and independent of the network size, topology, or the jammer type (about 50 time-slots). However, a jammed node's connection to the rest of the network will remain disrupted until their non-jammed neighbor(s) (boundary nodes) learn about the attack. It takes much longer for a boundary node to detect jamming since the sensor networks in general will be experiencing a lot of topological changes and therefore neighbor probing must allow for the possibility of such temporarily changes before declaring a neighbor as being missing (or jammed). This time seems to also be constant in all the different experiments

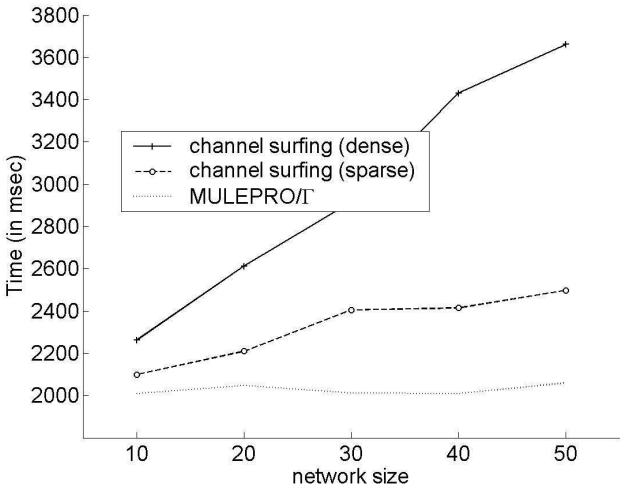


Figure 6.7: Time it takes to regain connectivity increases with dense topologies for the channel surfing approach but is lower bounded by the time needed for a boundary node to detect jamming in the MULEPRO protocol.

(roughly at 2000 time-slots). It is after this time that nodes inside of the jammed region would be able to react to the attack and regain connectivity to the network.

Therefore, in figure 6.7, it is clear how the different protocols were lower bounded by this constant time. In addition, the MULEPRO-k protocol did not need much more time to react. In the case of the channel surfing protocol, more time was incurred to react due to the time required for the *switch-channel* announcement to travel through the entire network. The flooding effect is even more brutal in the case of dense networks.

6.4 Discussion

Experimental evaluations of MULEPRO shows that high MDR values can be obtained even when multiple attackers are present. With increased number of available channels, simulation results showed that MULEPRO can be completely resilient to jamming attacks. Another major advantage of our approach is the reduction or elimination of control messages. Therefore, attacked nodes will still be able to react quickly even when the communication channel is not available. We conclude that MULEPRO is an effective countermeasure to jamming attacks for all situations that do not result in network partition. In the next

chapter, MULEPRO is applied into a data-driven model under more realistic conditions to evaluate the performance of the data exfiltration approach when used as a fault tolerance mechanism.

Chapter 7: Data Exfiltration for Critical Infrastructure Protection

This chapter presents an application of the exfiltration algorithms for the purpose of critical infrastructure protection. We extend the simulation model presented in the previous chapter and present the design of a data-driven simulation of the MULEPRO protocol. The underlying assumption is the placement of sensor technology on pre-existing distribution networks. The scenario we consider is a sensor network, specifically deployed for the purpose of monitoring some critical infrastructure, faced with some form of disturbance that may alter communication links in some areas. The main objective is to examine the role of the data exfiltration framework presented in this thesis when used as a fault tolerance mechanism in the WSNs' domain. The expectation is that the incorporation of the data exfiltration framework in the application's design will lead to increased reliability.

We start by discussing the role of WSNs in the protection of critical infrastructure in section 7.1 and we specifically consider the role of data exfiltration when used as a fault tolerance mechanism in such network. Section 7.2 introduces the application of the exfiltration algorithms for the purpose of critical infrastructure protection. We use the EPANET simulator to build a realistic data model, we then apply the data model to benchmark distribution networks. Performance evaluation is presented in section 7.3 and we conclude this chapter in Section 7.4.

7.1 The role of WSNs in Critical Infrastructure Protection (CIP)

Critical infrastructure are those systems that are essential for the functioning of a society. Examples include electricity generation, agriculture, transportation, telecommunication,

and oil and water supply systems. WSNs allow for data gathering and monitoring to be deeply embedded in the physical environment. Therefore, WSNs provide a great tool for monitoring critical infrastructure. The intelligent distributed capabilities and the ability to operate under severe conditions are some of the characteristics which make the application of WSNs technology to be a very attractive tool for the purpose of monitoring infrastructure systems. Vital events and data are automatically collected and fully integrated for real-time monitoring and long-term observation purposes. Critical infrastructure protection (CIP) is the study of protective measures that aim to increase the resilience of critical infrastructure to failure as a result of disaster, vandalism, sabotage, natural causes, or unforeseen faults [100]. The research community is recognizing the potential role of WSNs in the field of critical infrastructure protection [31, 35–39].

The deployment of a WSN for the purpose of monitoring a critical infrastructure is also characterized by unique requirements for communications performance, including timing and critical data delivery. Unfortunately, unattended functioning of WSNs makes them susceptible to a wide array of disturbances. The communication links, for instance, are especially vulnerable to disruptions [38]. Furthermore, the interconnected nature of these systems means that single, isolated disturbances can cascade through with potentially disastrous consequences[100]. Therefore, it is essential to have resilient and robust infrastructure protection systems that could deal with unexpected disturbance situations.

The application scenario in this chapter considers the use of WSNs as a CIP tool. Since real-time data delivery and responses are critical in many scenarios, we investigate the role of concurrent data exfiltration in adding a layer of *fault tolerance* capabilities to such networks. The rational behind this study is that, for the widespread adoption of sensor technology, it is essential to ensure that sensor networks are robust in the event of abnormal behavior such as a network intrusion or failure of components or nodes.

7.1.1 Data exfiltration for Fault Tolerance

The utilization of fault tolerance in the design process of WSNs applications is an affective approach for achieving more efficient sensor networking systems [101, 102]. Fault tolerance is a system property that enables the computer network to continue operating properly in the event of the failure of some of the network's components. The design of dependable fault-tolerant networks and systems plays a significant role in the successful and reliable implementation of wireless ad hoc networks in general. A fault-tolerant design enables a system continuous operation, possibly at a reduced level (also known as graceful degradation), rather than failing completely, when some part of the system fails [102]. Since sensor networks are inherently fault-prone [101] and since on-site maintenance is impractical, fault tolerance becomes even more vital for delivering efficient data sensing applications. Therefore, we extend the rational of the data exfiltration framework presented in this dissertation, and we investigate its role as a fault tolerance mechanism for protecting a drinking water distribution system.

7.2 Application of Exfiltration Algorithms for CIP

In this chapter, the application scenario is intended to show MULEPRO's effectiveness in protecting a critical infrastructure where we specifically consider a drinking water distribution system. The purpose is twofold: 1) to examine the exfiltration algorithms' role as a fault resilience mechanism for protecting CI applications, and 2) to validate the algorithms' performance under a more realistic data model and sensing application using real world settings. The underlying assumption is the placement of sensor technology on pre-existing water distribution networks (WDNs).

7.2.1 Sensors' Placement on Water Distribution Networks (WDNs)

Water supply systems are designed to satisfy the water requirements of domestic, commercial, industrial, and firefighting purposes. The importance of this evaluation is that water

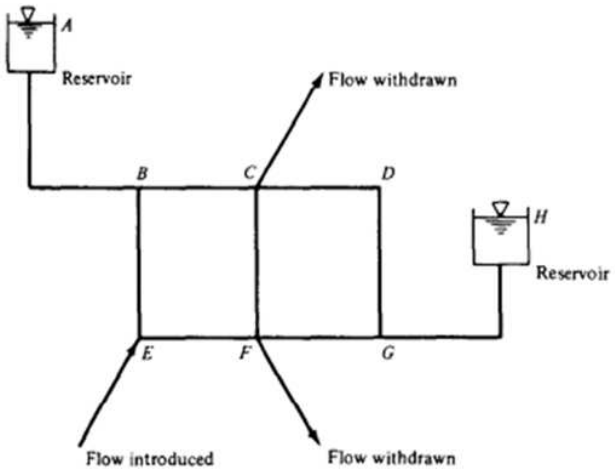


Figure 7.1: A model of a water distribution network.

distribution networks are everywhere and considered a critical infrastructure for all societies. Threat management in drinking water distribution systems involves, among other factors, real-time characterization of any contaminant source where near real-time responses are critical. In addition, disruptions in water flow can be due to natural disasters, temperature fluctuations in the environment, or simply due to aging of the pipes, and such disruptions often take time to identify and report. Providing up to date information to such municipal organizations as well as the ability to locate and characterize disturbances in the supply network itself is extremely beneficial.

A water distribution network consists of links and pipes of varying diameters joined together by distribution *junctions*, figure 7.1 depicts a sample WDN. Water can enter or leave the network at these junctions; we assume that wireless sensors are placed on the junction nodes. Such placement architecture has been proposed in the literature [34, 36]. The different sensor nodes will be forming an ad hoc wireless sensor network on top of the water distribution network. Each sensor node will be performing some sensing specific task and will periodically send back data to a communication base station.

7.3 Performance Analysis

In our simulation model of MULEPRO (presented in the previous chapter), we used the default topology modeling software provided by TOSSIM to generate random *synthetic* networks. We also used TOSSIM's default *synthetic* sensing component to provide sensors' readings to the MULEPRO's application. Our simulation approach in this chapter is based on *realistic* network modeling where we use freely available benchmark topologies for the water industry found in [103]. Furthermore, we adopt a *data-driven* modeling approach to replace TOSSIM's default sensing component with a more realistic sensor capable of providing accurate readings for each sensor node.

As described in [103], two UK research groups are collaborating to develop standards and benchmark models based on real water distribution networks. Figure 7.2 shows a sample benchmark network found in [103]. The network supplies water to more than 76,000 inhabitants. The whole zone consists of smaller interconnected sub-systems called demand management areas (DMAs). For this research, we focus on the demand management areas (DMAs) models along with their initial demands and conditions, with some minor modifications from its original form found in [103].

7.3.1 Simulation Methodology

We specifically considered four different water topological scenarios from the benchmark models ranging from 11 to 80 nodes per a demand management area (DMA). Table 7.1 shows the simulated topology characteristics. The sensor placements are done in the following way: using the benchmark water distribution topology, we place a sensor at each junction location in the network. As apparent from the table, the density in some scenarios is very sparse and the distance between two neighboring sensing nodes may exceed the transmission range of a sensor node. To guarantee network connectivity, we assume that repeater nodes are also embedded along a pipe connecting any two junction nodes. Data packets are transferred between the two junction nodes via repeater nodes.

Each sensor node is sensing *hydraulic* properties of the water. Hydraulics is a topic of

Figure 7.2: A model of a water supply system that supplies water to more than 76,000 inhabitants and consists of many smaller networks based on demand level.

Table 7.1: Simulated topology characteristics
(models used with some minor modifications from their original forms found in [103])

Model	**Junctions**	**Demand Area**	**Density**
SIM_2 Model	11	2000 m x 1800 m	3.06 nodes/km^2
SIM_1 Model	13	48 m x 88 m	3077.68 nodes/km^2
SIMPL_1 Model	60	2000 m x 2700 m	111.12 nodes/km^2
OPT_1 Model	80	15,200 m x 10700 m	2.03 nodes/km^2

science and engineering dealing with the mechanical properties of liquids such as pipe flow [104]. To get more accurate and realistic hydraulic contents, we use EPANET [21], a high fidelity data simulator for water distribution systems. EPANET is designed to simulate hydraulic flow in pipe networks. An EPANET simulation predicts the dynamic of hydraulic and water quality behavior within a distribution network operating over an extended period of time. The EPANET software allows the user to define simulation duration, as well as intervals at which EPANET is to sample the networks hydraulic sources, pipes, and junctions for data. One of the most powerful features of EPANET is that it allows complete customization of demand at every node in a network, i.e. custom demand curves can be defined and ascribed to all demand points. These curves allow diversity across a network

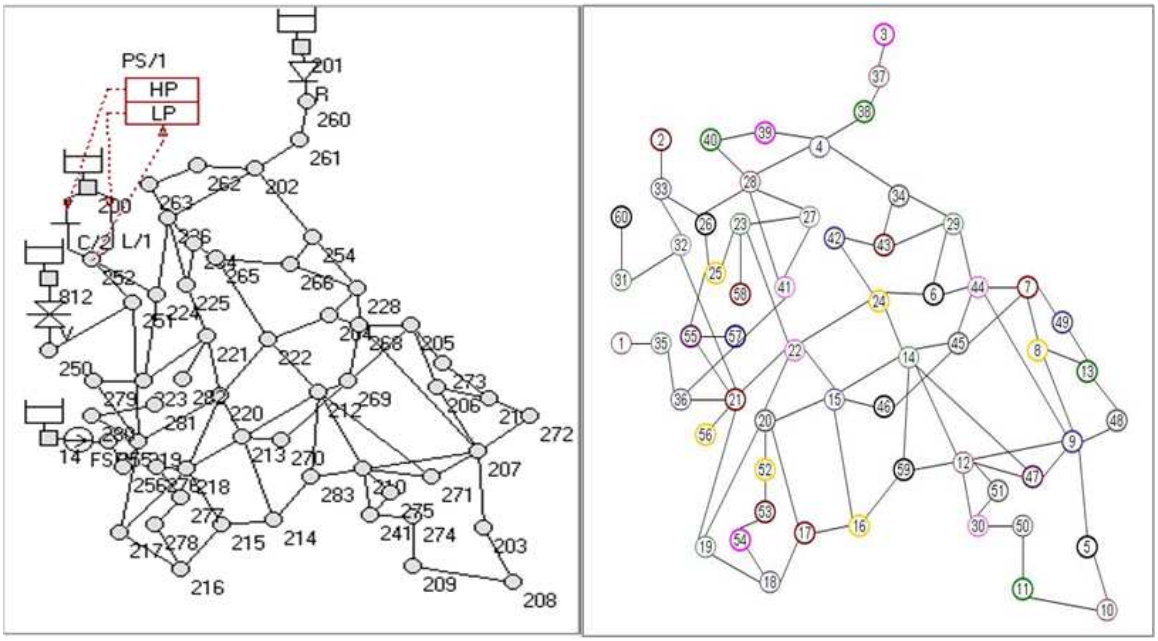


Figure 7.3: Model SIMPL_1 is the benchmark water distribution network with 60 nodes.

creating a more realistic scenario for analysis. A plain text report containing all the hydraulic readings of all nodes (at each time-step in the simulation) is generated at the end of the simulation run. In our data-driven approach, EPANET provides the simulation engine that generates the hydraulic data model. In other words, we use EPANET simulation results to generate an input component to be used by the TOSSIM simulator as part of our simulation architecture in this chapter. We now explain our approach in building the data-driven simulation model used in this chapter.

As a preprocessing step, the benchmark distribution networks are first converted into a sensor network topology in a pre-step performed in TOSSIM. Two nodes sharing a pipe are considered one-hop neighbors. If a pipe length between two junction nodes extends the transmission range of a sensor node, a repeater node is placed on the pipe such that the repeater is within the transmission range of the sensor node on the junction. Repeaters are added along a pipe as needed. A sample input network showing the network before and after the pre-processing step (conversion to a sensor network and the vertex-coloring step) is depicted in figure 7.3. A single sink node is located at node 28.

In addition to using a real topology as input to our simulation model, we design and develop a new TinyOS component to replace the default sensor component provided by TOSSIM. The new component is intended to act as a sensor device in a water quality surveillance WSN that periodically provides the sensor network application with sensor readings. The different parameters, such as sampling rate, data aggregates, network rate, etc., were assigned with respect to real-world scheduling provided by the benchmark models in [103]. Table 7.2 lists some of the parameters used when performing the EPANET simulations. For instance, the Hazen-Williams equations, which are the formulae used when determining friction head loss in a pipe system, are used in all the different EPANET simulations.

Table 7.2: Parameters used in EPANET simulations.

Benchmark Model	**Pipe Model Type**	**Start Time (hr:min)**	**End Time (hr:min**	**Time Step** (min)
SIM_2 Model	HAZEN-WILLIAMS	03:00	19:00	60
SIM_1 Model	HAZEN-WILLIAMS	09:00	33:00	180
SIMPL_1 Model	HAZEN-WILLIAMS	00:00	24:00	60
OPT_1 Model	HAZEN-WILLIAMS	00:00	24:00	30

Figure 7.4 shows a framework for the synthesis of a sensor node component based on data-driven modeling. Time-series hydraulic data for the benchmark WDNs are generated using the EPANET simulator. The hydraulic data model is then used for building the new sensor component *HydraulicReaderC*. To incorporate the data-driven modeling component, we replace the *SensorC* component in MULEPRO's component-architecture diagram with the new component *HydraulicReaderC*, as shown in figure 7.5.

7.3.2 Experimental Results

In this section, we present simulation results of the MULEPRO protocol to validate its effectiveness in protecting a water distribution system consisting of the DMAs described above (table 7.1). The purpose of the sensor network application is to monitor the readings

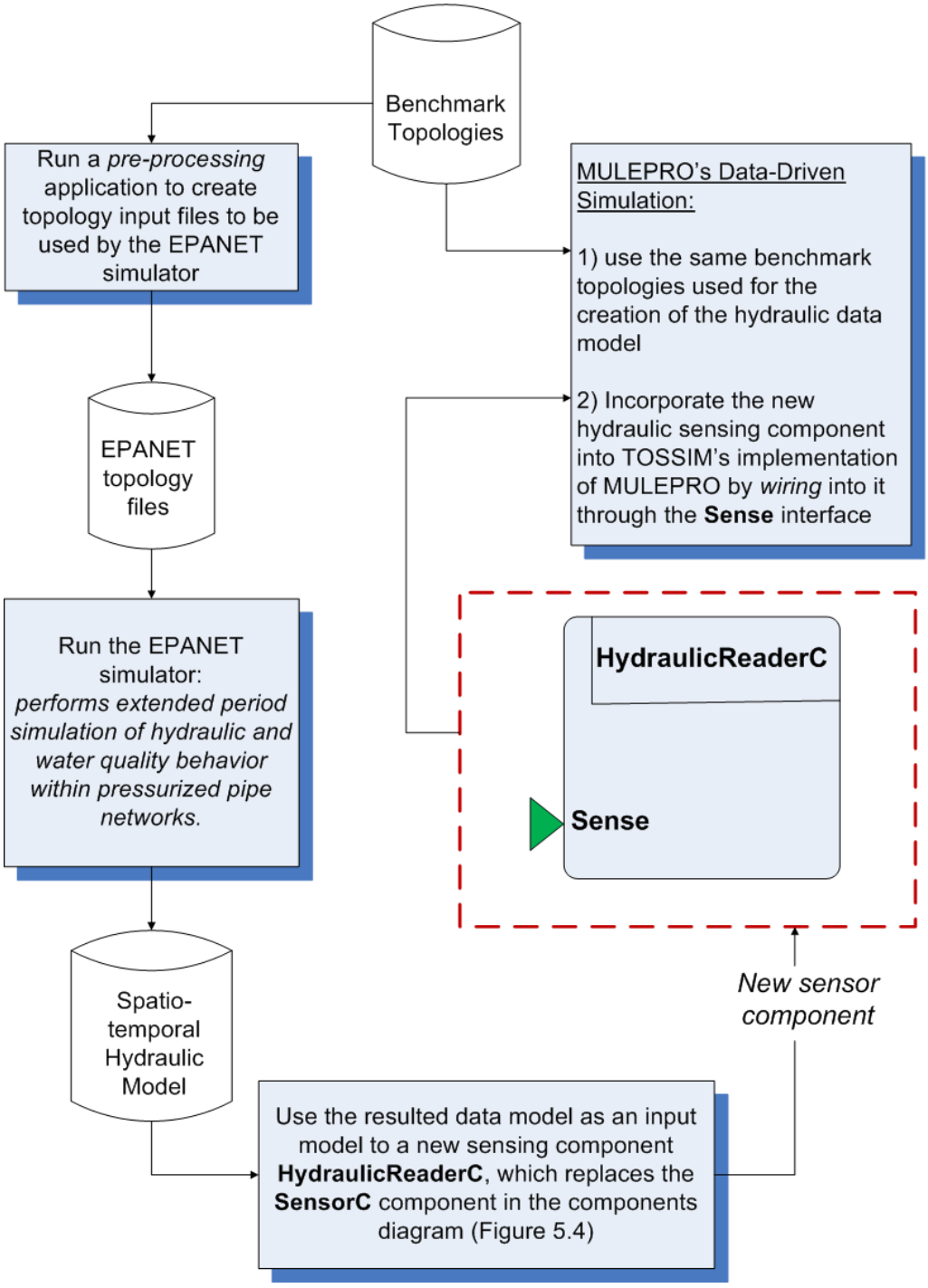


Figure 7.4: Concept of realistic data-driven modeling: combining real benchmark topologies and physically-based data models. The EPANET simulator is used to build the hydraulic data models that are applied into the TOSSIM's simulations of the different benchmark water distribution networks.

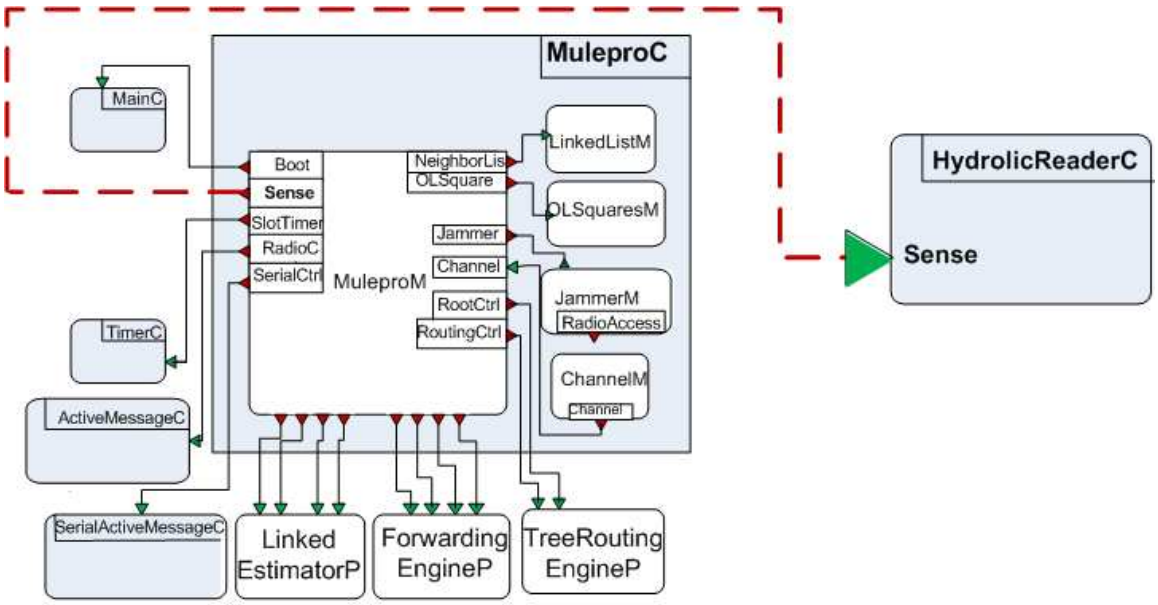


Figure 7.5: Data-driven Simulation of MULEPRO. The *wiring* of the new sensor component that captures *physically-based* hydraulic data generated by the EPANET simulation.

of the different sensors. The frequency of data reporting of the sensors is between 11 and 14 seconds. Our main objective is to use the hydraulic time-series with simulated communication link *faults* to validate the performance of the exfiltration algorithms. A fault will affect data delivery and we evaluate the role the data exfiltration approach in minimizing such affect. We do that by monitoring data delivery rates at the base station node. Thus, the metric that we use to validate the simulation model in this set of experiments is the message delivery ratio (MDR).

Experiment 1: A fault affecting a single communication channel in a multi-hop region. The results for a network fault that alters a portion of the frequency on which the network is communicating are shown the table 7.3. The presence of the fault (simulated as a single unintentional hopping interference attack) alters communication within a multi-hop region, affecting only a *single* channel at a time. Such an assumption may not be very practical. However, we consider the single channel assumption for the purpose of comparing against the channel surfing approach (which assumes a single channel is blocked

Table 7.3: MDR values for the benchmark water distribution networks, showing the case of a single channel fault extending to a multi-hop area, number of extra channels (Γ) is set to 3.

number of nodes	11	13	60	80
no fault	99%	99%	99%	99%
fault/no response	71%	71%	69%	70%
fault/channel surfing	98%	99%	96%	94%
fault/exfiltrate data (MULEPRO-k/3)	99%	99%	98%	98%

at a time). We note here that the channel surfing approach is also applicable as a defense in this case; however, it does come with a much higher overhead (as was shown in the performance analysis chapter, chapter 6).

Experiment 2: A fault affecting multiple communication channels in a multi-hop region. We also ran experiments to simulate more sophisticated communication link faults (simulated as intentional interference attacks). In those experiments, *multiple* attackers affecting *multiple-hop* areas were applied to the different benchmark topologies. The results are shown in the Figure 7.6 and 95% confidence intervals are also shown along the bars. The figure displays MDR values for the following three cases: when the network is operating under normal conditions without the presence of sophisticated attackers, when data exfiltration is used as a defense, and when no defense is implemented by the affected nodes. As can be seen, MULEPRO manages to exfiltrate approximately the same amount of data as conditions when no attackers were present, whereas an unresponded-to interference attack degrades network performance by approximately 30%. In fact, for the 60 and 80 nodes networks, MULEPRO's exfiltration results are statistically insignificant from the case where the network did not experience any faults, as represented by the confidence intervals on the bar figure.

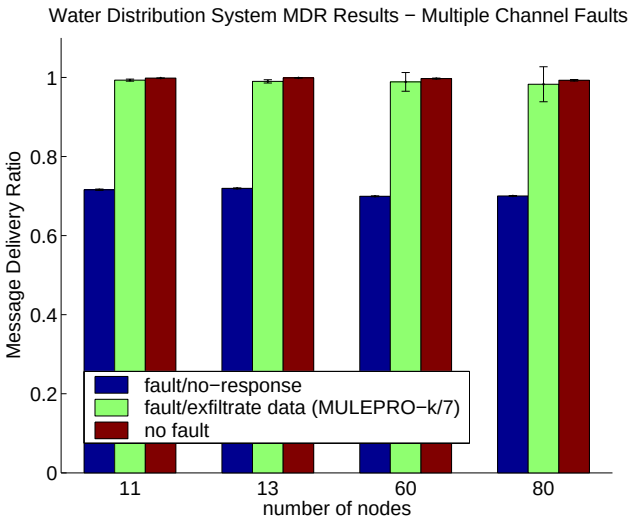


Figure 7.6: MDR values for the water distribution system, showing the case of *multiple* simultaneous attacks effecting multi-hop areas of the network, Γ is set to 7.

7.4 Summary

We presented a practical application of our data exfiltration framework and presented its role in protecting a critical infrastructure. We used benchmark water distribution networks along with realistic data models, and we discussed the deployment of sensor technology on pre-existing networks. The findings lead to the conclusion that the framework presented in this dissertation is capable of supporting many applications that the real world may require and, most importantly, the incorporation of the data exfiltration framework in the applications design will lead to increased reliability and robustness in WSNs.

Chapter 8: Conclusions and Future Directions

This Chapter summarizes our work along with important conclusions and outlines some of the possible areas for future research.

8.1 Summary and Conclusions

Given the current hardware technology, wireless networks remain vulnerable to denial of service attacks against the communication channel either at the link or physical layers. Solutions that tackle the problem can be integrated via the construction of efficient systems. In this dissertation, we presented a multi-channel defense against communication denial-of-service attacks in wireless networks. We adopted an architectural approach for the purpose of multi-channel data exfiltration as an automatic response to such attacks. We developed a framework that aims at having minimal active security procedures at the link layer and only activates the security defense when an attack is actually present. Concurrent data exfiltration is also incorporated in the wireless sensor communication stack for developing more efficient sensor networking architectures, protocols and systems. In general, by addressing the radio interference issue in the design of sensor networking, an overall enhancement of the wireless sensor communication stack is achieved.

We focused on the development of this framework, particularly on the design and development of the two reactive mechanisms it is based on: reactive multi-channel broadcast scheduling (RMBS) and the multi-channel data exfiltration protocol (MULEPRO). The main objective of RMBS scheduling is to act as a deterministic time division channel access to guarantee collision-free, concurrent transmissions out of an attacked region. Furthermore, it provides the underlying approach to performing the distributed channel assignments for the multi-channel exfiltration framework. Four modes of operations are designed ranging

from management-free scheduling to precise coordinated scheduling. A distinctive feature of RMBS is that its performance is robust to topology changes and varying channel conditions; in the worst case, its performance always guarantees a minimum exfiltration throughput out of an attacked region. We also designed and implemented MULEPRO, a fully distributed network based protocol designed to rapidly exfiltrate data from a region undergoing intentional or non-intentional interference. MULEPRO constitutes the higher layer of the exfiltration stack in the presented framework and implements RMBS at the link layer. It allows network nodes to perform in a normal state when no attack is underway and facilitates the multi-channel defense only when a communication DoS attack is present. Hence, it incorporates a hybrid MAC at the link layer that allows for the switching between CSMA mode of operation (during normal state) to a time-slotted mode operating under RMBS (when an attack is detected). In addition, MULEPRO has two protocol variants, one to act as a response against single-hop attacks and another that provides resilience against attacks affecting multi-hop regions.

A simulation-based evaluation of the data exfiltration methods and protocols has demonstrated that significant performance improvement in data delivery is always possible, even if each node has only one single channel available for communication. Continuous data reception at the sink node (or the base station) is guaranteed as long as an attacker does not affect *all* of the communication channel continuously. Furthermore, the limited communication bandwidth is greatly improved with the use of the multiple channels. By simulating different types of jamming attacks, simulation results showed that MULEPRO can be completely resilient to such attacks when the number of available channels is large. Another major advantage of our approach is the reduction or elimination of control messages via the use of Latin square scheduling. As a result, attacked nodes will still be able to react quickly even when the communication channel is not available. We concluded that the data exfiltration protocols and methods are an effective countermeasure against communication DoS attacks for all situations that do not result in network partition.

To investigate the applicability of the data exfiltration defense in more realistic scenarios

and to further evaluate its performance, we further address data exfiltration as a reactive approach for protecting critical infrastructure. The scenario we considered is when a WSN, specifically deployed for the purpose of monitoring some critical infrastructure, is faced with accidental or intentional interference that may alter communication in some area. The main objective was to examine the role of the data exfiltration framework presented in this thesis when used as a fault tolerance mechanism in the WSNs' domain. We used benchmark water distribution networks along with realistic, hydraulic data models. The findings lead to the conclusion that the data exfiltration framework presented in this dissertation is, in fact, capable of supporting many applications that the real world may require, and most importantly, the incorporation of the exfiltration framework in the applications design will lead to increased reliability and robustness in WSNs.

8.2 Future Work

There are some open issues related to this dissertation that may be addressed in future research.

8.2.1 Protocol Issues

The following are possible extensions to the exfiltration protocols presented in this thesis. These extensions are aimed at improving the practicality of broadcast scheduling and fit smoothly into the distributed framework that we have described.

Adaptive Scheduling. It can be expected in a large network that some nodes may expectedly or unexpectedly join or leave the network. Managing the joining and leaving of nodes to ensure smooth operation of the reactive protocols is therefore an important issue. The RMBS scheduling techniques do not consider possible topological changes. In the current design, time-slot assignments are performed based on the vertex-colors of the competing nodes. The vertex-coloring algorithm is run only once during the network initialization phase. Such an approach places a limit on the *adaptivity* of the exfiltration protocols.

When new nodes are added to the network, the total number of vertex-colors in the system (k) may change. A new k value must then be propagated to the entire system. This could incur high cost for adapting to a small change in the network topology. In addition, the change in the k value may also necessitate the generation of a new MOLS family. Thus, further research is needed to extend the design of the exfiltration protocols to make them more efficient in the face of possible topological changes.

Local Time Frame. Even though current channel assignment techniques in the exfiltration protocols makes some attempt to utilize unused slots within a broadcast schedule, many slots remain unused within an exfiltration time-frame. When the network's density is uniform, a global schedule would create a smaller number of empty slots. But if the network contains many sparse areas with only a few dense areas, then some local framing (i.e. local schedules) would be more preferable. The minimization of the time-frame length is expected to result in higher throughput and also minimal delay.

Dynamic MOLS generation. In the current implementation, the generation of the MOLS family is incorporated within the MULEPRO protocol itself. Separating the task of the MOLS creation and maintenance from the actual operation of the data exfiltration protocol will lead to a more robust exfiltration architecture as was discussed in subsection 5.2.3. Such an extension also conforms with the optimal MOLS construction (subsection 4.4.4.1), which shows that the throughput bounds are decreasing when the p values increases (equation 5.1). The use of the suggested MOLS management protocol will provide MULEPRO with the robustness needed so that the determination of the order of the MOLS family (i.e. p value) is optimized.

8.2.2 Optimization of the Formal Model

As part of our research, we formulated the RCAP problem model that represents the problem of concurrent data exfiltration as a variant of the channel assignment problem, which is known to be NP-hard. We also defined a formal solution based on the branch and bound

method, resulting in an efficient and unique scheduling design that satisfies the RCAP constraints and requirements. The revised branch and bound algorithm in our work is based on the use of Latin square assignments as a heuristic methods for the branching procedure. While the branch and bound approach has been shown to be effective in reducing the complexity of the channel assignment problem [91], formal evaluation of the use of the Latin square assignments heuristic remains an open issue.

Bibliography

Bibliography

[1] A.D.Wood and J. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[2] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, "Adaptive security for multi-layer ad-hoc networks," *Special Issue of Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 533–547, 2002.

[3] J. M. Mccune, E. Shi, A. Perrig, and M. K. Reiter, "Detection of denial-of-message attacks on sensor network broadcasts," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy (SP)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 64–78.

[4] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*. Anchorage, Alaska, USA: IEEE, 2003, pp. 113–127.

[5] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, San Antonio, TX, USA, January 27-31 2002.

[6] B. Potter, "Wireless securityś future," *IEEE Security and Privacy*, vol. 1, no. 4, pp. 68–72, 2003.

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

[8] V. D. Gligor, "A note on the denial-of-service problem," in *Proceedings of the 1983 IEEE Symposium on Security and Privacy (SP)*. Washington, DC, USA: IEEE Computer Society, 1983, pp. 139–149.

[9] "Crossbow Radio Communication," http://www.xbow.com/Technology/RadioCommunication.aspx.

[10] P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom)*. New York, NY, USA: ACM, 2005, pp. 43–57.

[11] K. Römer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, 2004.

[12] D. C. Schleher, *Electronic Warfare in the Information Age.* Norwood, MA: Artech House, Inc., 1999.

[13] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM Sigmobile Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, 2002.

[14] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: defenses against wireless denial of service," in *Proceedings of the 3rd ACM workshop on Wireless security (WiSe).* New York, NY, USA: ACM, 2004, pp. 80–89.

[15] Y. W. Law, L. van Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols," in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN).* New York, NY, USA: ACM, 2005, pp. 76–88.

[16] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc).* New York, NY, USA: ACM, 2005, pp. 46–57.

[17] D. Litchfield, *The Oracle Hacker's Handbook: Hacking and Defending Oracle.* New York, NY, USA: John Wiley & Sons, Inc., 2007.

[18] P. A. Fishwick, *Simulation Model Design and Execution: Building Digital Worlds.* Upper Saddle River, NJ, USA: Prentice Hall PTR, 1995.

[19] P. Levis, "Tossim: A simulator for tinyos networks," 2003. [Online]. Available: http://www.cs.berkeley.edu/ pal/pubs/nido.pdf

[20] "TinyOS HTTP Site," http://www.tinyos.net/.

[21] "EPANET HTTP Site," http://www.epa.gov/nrmrl/wswrd/dw/epanet.html.

[22] "IEEE 802.11. The Working Group Setting the Standards for Wireless Local Area Networks (WLANs)," http://ieee802.org/11.

[23] "IEEE 802.15. The Working Group Setting the Standards for Wireless Personal Area Networks (WPANs)," http://ieee802.org/15.

[24] "XBOW MICA Mote Specifications," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA.pdf.

[25] "XBOW MICA2 Mote Specifications," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

[26] "XBOW MICAZ Mote Specifications," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf.

[27] "XBOW TELOS Mote Specifications," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[28] "ZigBee Alliance," http://www.zigbee.org/en/index.asp.

[29] L. Bao and J. Garcia-Luna-Aceves, "Chapter 4: Distributed channel access scheduling for ad hoc networks," in *Handbook of Algorithms for Wireless Networking and Mobile Computing*, A. Boukerche, Ed. Chapman & Hall/CRC, 2006.

[30] K. Greene, "A wireless-sensor network to report pollution and traffic comes to cambridge, ma," 2007. [Online]. Available: http://www.technologyreview.com/Infotech/18533/

[31] M. Hatler, D. Gurganious, and C. Chi, "Wireless sensor networks for oil gas 2008," 2008. [Online]. Available: http://www.onworld.com/wsn/oilgas.htm

[32] S. of Engineering and H. U. Applied Sciences, "Codeblue: Wireless sensor networks for medical care," 2006. [Online]. Available: http://www.eecs.harvard.edu/ mdw/proj/codeblue/

[33] K. Lorincz, D. J. Malan, T. R. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: Challenges and opportunities," *IEEE Computer Society*, vol. 3, no. 4, pp. 16–23, 2004.

[34] J. Berry, L. Fleischer, W. Hart, and C. Phillips, "Sensor placement in municipal water networks," in *Journal of Water Resources Planning and Management*, vol. 131, no. 3, 2005, pp. 237–243.

[35] A.D.Wood and J. Stankovic, "The role of wireless sensor networks in the area of critical information infrastructure protection," in *Information Security Tech. Report*, vol. 12, no. 1. Oxford, UK, UK: Elsevier Advanced Technology Publications, 2007, pp. 24–31.

[36] K. Mahinthakumar, G. von Laszewski, S. R. Ranjithan, D. Brill, J. Uber, K. Harrison, S. Sreepathi, and E. M. Zechman, "An adaptive cyberinfrastructure for threat management in urban water distribution systems," in *International Conference on Computational Science*, ser. Lecture Notes in Computer Science. Springer, 2006, pp. 401–408.

[37] "Critical information infrastructures security based on internet working sensors," http://www.isac.uma.es/CRISIS/.

[38] M. Albano, S. Chessa, and R. D. Pietro, "Information assurance in critical infrastructures via wireless sensor networks," *Information Assurance and Security, International Symposium on*, vol. 0, pp. 305–310, 2008.

[39] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, "PIPENET a wireless sensor network for pipeline monitoring," in *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks.* New York, NY, USA: ACM, 2007, pp. 264–273.

[40] B. Ames, "Cambridge to host wireless sensor network," 2007. [Online]. Available: http://www.washingtonpost.com/wp-dyn/content/article/2007/04/09/AR2007040900012.html

[41] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler, "The emergence of networking abstractions and techniques in tinyos," in *The 1st Conference on Symposium on Networked Systems Design and Implementation (NSDI)*, San Fransisco, CA, USA, March 29-31 2004.

[42] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *The 1st ACM conference On Embedded Networked Sensor Systems (SynSys)*, Los Angeles, CA, USA, November 5-7 2003.

[43] "Environmental Protection Agency - Risk Management Research Site," http://www.epa.gov/ORD/NRMRL/.

[44] W.-H. Tam and Y.-C. Tseng, "Joint multi-channel link layer and multi-path routing design for wireless mesh networks," in *The 26th Annual IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, May 6-12 2007.

[45] A. Raniwala and T. cker Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *The 24th Annual IEEE Conference on Computer Communications (INFOCOM)*, Miami, March 13-17 2005.

[46] G. Z. J. Stankovic and S. Son, "The crowded spectrum in wireless sensor networks," in *Third Workshop on Embedded Networked Sensors (EmNets)*, Harvard University,Cambridge, Massachusetts, USA, May 30-31 2006.

[47] H.-J. Ju and I. Rubin, "Backbone topology synthesis for multi-radio meshed wireless lans," in *The 25th Annual IEEE Conference on Computer Communications (INFOCOM)*, Barcelona, Catalunya, Spain, April 23-29 2006.

[48] X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks," in *The 26th annual IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, May 6-12 2007.

[49] J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. New York, NY, USA: ACM, 2004, pp. 222–233.

[50] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *Proceedings of the First International Conference on Broadband Networks (BROADNETS)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 344–354.

[51] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, USA, March 13-17 2005.

[52] A. Nasipuri and S. R. Das, "Multichannel CSMA with signal power-based channel selection for multihop wireless networks," in *IEEE Vehicular Technology Conference (VTC)*, Tokyo, Japan, September 24-28 2000.

[53] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," in *The 10th IEEE International Conference on Computer Communications and Networks (IC3N)*, Phoenix, Arizona, USA, October 15–17 2001.

[54] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. abdelzaher, "Mmsn: Multi-frequency media access control for wireless sensor networks," in *The 25th annual IEEE Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, April 23-29 2006.

[55] M. J. Miller and N. H. Vaidya, "A MAC protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 228–242, 2005.

[56] X. Chen, P. Han, Q.-S. He, S. liang Tu, and Z.-L. Chen, "A multi-channel MAC protocol for wireless sensor networks," in *The 9th IEEE International Conference on Computer and Information Technology (CIT)*, Bhubaneswar, India, December 18-21 2006.

[57] H. K. Le, D. Henriksson, and T. Abdelzaher, "A control theory approach to throughput optimization in multi-channel collection sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks (IPSN)*. New York, NY, USA: ACM, 2007, pp. 31–40.

[58] R. Simon, L. Huang, E. Farrugia, and S. Setia, "Using multiple communication channels for efficient data dissemination in wireless sensor networks," in *The 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Washington, DC, USA, November 7-10 2005.

[59] L. Huang, "Reliable bulk data dissemination in sensor networks," Ph.D. dissertation, George Mason University, 2007.

[60] A. G. Chao Gui and P. Mohapatra, "Securing sensor networks using a novel multichannel architecture," in *The 3rd International Conference on Broadband Communications, Networks, and Systems (BROADNETS)*, San Jose, CA, USA, October 1-5 2006.

[61] Y. Wu, J. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *The 27th Conference on Computer Communications (INFOCOM)*, 2008, pp. 1193–1201.

[62] C. F. Laywine and G. L. Mullen, *Discrete Mathematics Using Latin Squares.* Wiley-Interscience, 1998.

[63] J. Denes and A. D. Keedwell, *Latin Squares and Their Applications.* New York, NY, USA: Academic Press, 1974.

[64] J.-H. Ju and V. O. Li, "TDMA scheduling design of multihop packet radio networks based on Latin squares," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1345–1352, 1999.

[65] G. J. Pottie and R. Calderbank, "Channel coding strategies for cellular radio," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 4, pp. 763–770, 1995.

[66] L. Bao, "MALS: multiple access scheduling based on Latin squares," in *Military Communications Conference (MILCOM)*, Monterey, CA, USA, October 31-November 3 2004.

[67] R. Simon and E. Farrugia, "Topology-transparent support for sensor networks," in *Proceedings of 1st European Workshop on Wireless Sensor Networks (EWSN)*, ser. Lecture Notes in Computer Science, vol. 2920. Springer, 2004, pp. 122–137.

[68] C. Zhijun Cai Mi Lu Georghiades, "Topology-transparent time division multiple access broadcast scheduling in multihop packet radio networks," *IEEE Transactions on Vehicular Technology*, vol. 52, pp. 970– 984, 2003.

[69] A. E. Moutia, K. Makki, and N. Pissinou, "Space division multiple access scheme based on uniform Latin squares for wireless sensor networks," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 7, pp. 188–197, 2007.

[70] J. G. Proakis, *Digital Communications*, 4th ed. Boston, MA, USA: McGraw-Hill, 2000.

[71] R. Kahn, S. Gronemeyer, J. Burchfiel, and R. Kunzelman, "Advances in packet radio technology," *Proceedings of the IEEE*, vol. 66, no. 11, pp. 1468–1496, 1978.

[72] W. Xu, K. M. Trappe, and W. Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE Network*, vol. 20, no. 3, 2006.

[73] A. D. Wood, J. A. Stankovic, and S. H. Son, "Jam: A jammed-area mapping service for sensor networks," *24th IEEE Real-time Systems Symposium (RTSS)*, December 3-5 2003.

[74] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *The 2nd ACM Conference On Embedded Networked Sensor Systems (SynSys)*, Baltimore, MD, USA, November 3-5 2004.

[75] T. H. G. Zhou, J. A. Stankovic, and T. F. Abdelzaher, "Rid: Radio interference detection in wireless sensor networks," in *The 24th IEEE Annual Conference on Computer Communications (INFOCOM)*, Miami, FL, USA, March 13-17 2005.

[76] M. Cagalj, S. Capkun, and J. Hubaux, "Wormhole-based anti-jamming techniques in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 100–114, 2007.

[77] W. Xu, W. Trappe, and Y. Zhang, "Anti-jamming timing channels for wireless networks," in *Proceedings of the first ACM conference on Wireless network security.* New York, NY, USA: ACM, 2008, pp. 203–213.

[78] S. Khattab, R. Melhem, and D. Moss, "Modeling of the channel-hopping anti-jamming defense in multi-radio wireless networks," in *The 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, Dublin, Ireland, July 21-25 2008.

[79] S. Khattab, D. Moss, and R. Melhem, "Jamming mitigation in multi-radio wireless networks: Reactive or proactive?" in *The 4th International Conference on Security and Privacy in Communication Networks (SecureComm)*, Istanbul, Turkey, September 22-25 2008.

[80] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing: Defending wireless sensor networks from jamming and interference," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*, 2007.

[81] A. D. Wood, J. Stankovic, , and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks," in *The 4th IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, San Diego, CA, USA, June 18-21 2007.

[82] J. Padhye, R. Draves, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *the 10th ACM International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, USA, September 26 -October 1 2004.

[83] Y. Sun, J. Qian, and J. Wu, "Hybrid energy-aware time synchronization protocol for wsns in coal mine," in *International Conference on Information Acquisition (ICIA)*, Jeju Islan, Korea (South), July 9-11 2007.

[84] S. Yanos, "Energy-aware time synchronization in wireless sensor networks," 2006. [Online]. Available: http://www.cs.berkeley.edu/ pal/pubs/nido.pdf

[85] Y. Sun, J. Qian, and J. Wu, "Energy effective time synchronization in wireless sensor network," in *Proceedings of the The 2007 International Conference Computational Science and its Applications (ICCSA)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 547–553.

[86] G. Noubir and G. Lin, "Low-power dos attacks in data wireless lans and countermeasures," *ACM sigmobile Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 29–30, 2003.

[87] T. Zia and A. Zomaya, "Security issues in wireless sensor networks," in *International Conference on Systems and Networks Communications (ICSNC)*, Tahiti, French Polynesia, October 29 - November 3 2006.

[88] E. L. Lloyd, "Broadcast scheduling for tdma in wireless multihop networks," in *Handbook of Wireless Networks and Mobile Computing*, ser. Wiley Series on Parallel and Distributed Computing. New York, NY, USA: John Wiley & Sons, Inc., 2002, pp. 347–370.

[89] R. Ramaswami and K. Parhi, "Distributed scheduling of broadcasts in a radio network," in *Proceedings of the 8th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2, no. 23, 1989, pp. 497–504.

[90] W. Hale, "Frequency assignment: Theory and applications," *Proceedings of the IEEE*, vol. 68, no. 12, pp. 1497– 1514, 1980.

[91] A. Giortzis and L. Turner, "A mathematical programming approach to the channel assignment problem in radio networks," in *IEEE 46th Vehicular Technology Conference (VTC)*, Atlanta, GA, USA, April 28 - May 1 1996.

[92] L. Giortzis, A.I. Turner, "Application of mathematical programming to the fixed channel assignment problem in mobile radio networks," *The IEEE Communications*, vol. 144, no. 4, 1997.

[93] D.-Y. Chan, C.-Y. Ku, and M.-C. Li, "A method to improve integer linear programming problem with branch-and-bound procedure," *Applied Mathematics and Computation*, vol. 179, no. 2, pp. 484–493, 2006.

[94] J. Gross and J. Yellen, *Graph theory and its applications.* Boca Raton, FL, USA: CRC Press, Inc., 1999.

[95] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?" in *IEEE Global Telecommunications Conference*, Taipei, Taiwan, November 17-21 2002.

[96] S. Krumke, M. Marathe, and S. Ravi, "Models and approximation algorithms for channel assignment in radio networks," *Wireless Networks*, vol. 7, pp. 575–584, 2001.

[97] C. Szyperski, *Component Software: Beyond Object-Oriented Programming.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[98] P. H. Frohlich, "Component-oriented programming languages: Why, what, and how," Ph.D. dissertation, UNIVERSITY OF CALIFORNIA,IRVINE, 2003.

[99] C. S. Metcalf, "Tossim live: Towards a testbed in a thread," Ph.D. dissertation, Colorado School of Mines, 2007.

[100] S. D. Personick and C. A. Patterson, *Critical Information Infrastructure Protection and the Law: An Overview of Key Issues.* Washington, DC, USA: National Academy Press, 2003.

[101] K. Sun, P. Ning, and C. Wang, "Fault-tolerant cluster-wise clock synchronization for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, pp. 177–189, 2005.

[102] D. Jayasimha, "Fault tolerance in multisensor networks," *IEEE Transactions on Reliability*, vol. 45, no. 2, pp. 308–315, 1996.

[103] "Benchmarks for water supply modeling, simulation and optimization," http://www.eng.dmu.ac.uk/ wss/index.html.

[104] P. R. Bhave, *Analysis of Flow in Water Distribution Networks.* Lancaster, PA.: Technomic Publishing, 1991.

Curriculum Vitae

Ghada M Alnifie will return to Saudi Arabia to resume her position at the Institute of Public Administration (IPA) in Riyadh, where she will be working as an assistant professor at the department of Computer Science. Part of her duties at IPA involves serving as a consultant for many Saudi Arabian governmental agencies. Ghada received her Bachelor of Science and Master of Science in Computer Science from George Mason University.