AGH UNIVERSITY OF KRAKOW

FIELD OF SCIENCE: Engineering and Technology

SCIENTIFIC DISCIPLINE: Information and Communication Technology

DOCTORAL THESIS

Steganographic methods for hidden data transmission in IEEE 802.11 wireless local area networks

Author: Geovani Teca
First Supervisor: Marek Natkaniec, PhD, DSc
Assistant Supervisor: Janusz Gozdecki, PhD

Completed in:
AGH University of Krakow
Faculty of Computer Science, Electronics and Telecommunications
Institute of Telecommunications

Krakow, 2025

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUKI: Inżynieria i technologia

DYSCYPLINA NAUKOWA: Informatyka Techniczna i Telekomunikacja

ROZPRAWA DOKTORSKA

Steganograficzne metody ukrytej transmisji danych w lokalnych sieciach bezprzewodowych standardu IEEE 802.11

Autor: Geovani Teca
Promotor: dr hab. inż. Marek Natkaniec, prof. AGH
Promotor pomocniczy: dr inż. Janusz Gozdecki

Praca wykonana w:
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Informatyki, Elektroniki i Telekomunikacji
Instytut Telekomunikacji

Kraków, 2025

Acknowledgment

First of all, I want to express my gratitude to Almighty God, Jehovah, for His grace that has allowed me to be where I am today. His hands have opened doors and guided me through life, and above all, for His love shown in Christ, who died for me (John 3:16).

My grandparents in Victória Luís Manuel (in memory) and Jaime Pedro Neto, for taking care of me, supporting, and advising me in life. I will always be grateful for all you have done for me. To my parents (Jeremias and Catarina), for raising me and making every effort so that I could have enough — my mother for making sure I received a good education, and my father for the opportunities, such as a university scholarship.

To my wife and children, for the support and understanding of the busyness and the time away from regular family activities that research consumed.

To my supervisor, Professor Natkaniec, who guided me through the ideas, encouraged me to do good research and raise the bar, and patiently reviewed the papers, helping during submissions and especially with this thesis. I deeply appreciate the support and guidance throughout the journey.

All friends and family who motivated me, asked for the status of the research, and shared a word of encouragement with me, thank you.

This page is intentionally blank.

Abstract

The term "steganography" originates from the Greek words "steganós" (covered) and "graphia" (writing or description), meaning "covered writing." Historically, it was used to conceal messages on battlefields. Still, steganography has evolved into a sophisticated technique that provides an additional layer of security in communication within modern digital and networked environments, complementing cryptography. In the context of wireless networks, particularly IEEE 802.11, steganography is implemented through covert channels. These channels enable the hidden embedding of data within legitimate network traffic, without alerting observers to the presence of secret information. Such covert channels exploit various protocol features, including headers, timing patterns, and frame structures, allowing the transmission of secret messages without raising suspicion or disrupting normal communication. This research offers a comprehensive analysis of the state-of-the-art in IEEE 802.11-based covert channels, reviewing approximately 59 published methods and proposing a taxonomy that classifies them based on shared characteristics. Building on this foundation, the thesis introduces four novel covert channels: two are storage-based, one is timing-based, and one represents a hybrid approach. These methods aim to enhance stealth, improve throughput, and ensure compatibility with standard IEEE 802.11 operations. In addition to discussing metrics such as throughput, efficiency, delay, jitter, transparency, and resistance to steganalysis, the thesis guides the real-life application of such covert channels in IEEE 802.11 networks, including modern amendments. As a future research directions, we highlight the urgent need for the development of robust countermeasures that can detect, neutralize, or eliminate covert communication in wireless networks. Furthermore, we emphasize the potential of hybrid covert channels to increase resilience and survivability, particularly in adversarial environments where detection mechanisms are in place.

This page is intentionally blank.

Streszczenie

Termin „steganografia” pochodzi od greckich słów „steganós” (ukryty) i „graphia” (pisanie lub opis), co oznacza „ukryte pismo”. Historycznie była wykorzystywana do ukrywania wiadomości na polach bitew, jednak steganografia rozwinęła się w zaawansowaną technikę, która stanowi dodatkową warstwę bezpieczeństwa w komunikacji w nowoczesnych środowiskach cyfrowych i sieciowych, uzupełniając kryptografię. W kontekście sieci bezprzewodowych, a w szczególności standardu IEEE 802.11, steganografia realizowana jest poprzez kanały ukryte (covert channels). Kanały te umożliwiają osadzanie ukrytych danych w legalnym ruchu sieciowym, nie wzbudzając podejrzeń o obecność informacji tajnych. Wykorzystują one różne cechy protokołu, takie jak nagłówki, wzorce czasowe czy struktury ramek, pozwalając na przesyłanie tajnych wiadomości bez zakłócania normalnej komunikacji. Niniejsza praca stanowi kompleksową analizę aktualnego stanu wiedzy na temat kanałów ukrytych w standardzie IEEE 802.11, obejmującą przegląd około 59 opublikowanych metod oraz propozycję taksonomii, która klasyfikuje je na podstawie wspólnych cech. W oparciu o tę bazę, rozprawa przedstawia cztery nowe kanały ukryte: dwa oparte na przechowywaniu, jeden oparty na zależnościach czasowych i jeden reprezentujący podejście hybrydowe. Celem zaproponowanych metod jest zwiększenie odporności na wykrycie ukrytych danych, poprawa przepustowości oraz zapewnienie zgodności z procedurami standardu IEEE 802.11. Oprócz omówienia metryk takich jak przepustowość, efektywność, opóźnienie, jitter, transparentność i odporność na stegoanalizę, rozprawa zawiera również wskazówki dotyczące praktycznego zastosowania takich kanałów ukrytych w sieciach standardu IEEE 802.11, w tym również w kontekście nowoczesnych rozszerzeń standardu. Jako kierunek przyszłych badań podkreślono pilną potrzebę opracowania skutecznych mechanizmów przeciwdziałania, które będą w stanie wykrywać, neutralizować lub eliminować ukrytą komunikację w sieciach bezprzewodowych. Zwrócono również uwagę na potencjał kanałów hybrydowych, które mogą zwiększać odporność i przeżywalność kanałów ukrytych, szczególnie w środowiskach wrogich, gdzie stosowane są mechanizmy detekcji.

This page is intentionally blank.

Contents

List of Figures

List of Tables

1 Introduction

1.1 802.11 networks evolution

IEEE 802.11 networks, commonly known as Wi-Fi networks[1], refer to Wireless Local Area Networks (WLANs) that operate according to the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standard [1]. These networks can be found in various settings, including homes, offices, campuses, and enterprises. While each deployment may be distinct, they all share a common foundation defined by the 802.11 standard. The IEEE 802.11 standard is not a single protocol, but rather a collection of specifications that includes multiple amendments such as 802.11a, 802.11b, 802.11e, 802.11g, 802.11i, 802.11n, 802.11ac, and 802.11ax, among others. Each of these variants introduces enhancements designed to achieve specific performance goals, adapt to different environments, or address various usage demands. For instance, they may improve throughput, enhance security, or support Quality of Service (QoS). The evolution of this standard reflects the increasing and diverse demand for high-performance wireless connectivity. As Wi-Fi continues to play a central role in modern communication, it supports a wide range of applications, from personal devices to enterprise and industrial settings.

Wi-Fi has become a leading technology, driven by the rapid growth of smart devices, offering a reliable, efficient, and cost-effective way to connect to the Internet [2]. It eliminates the need for complicated cable installations and extensive maintenance of network devices, making it an ideal choice for both residential and commercial environments. The flexibility of Wi-Fi and its ease of deployment enable user mobility, freeing individuals from physical cables and spatial constraints. This has enabled seamless connectivity in various settings, including homes, public hotspots such as shopping malls, bus stops, hospitals, and schools. As a result, Wi-Fi enhances productivity, supports economic and social activities, and contributes to educational development.

These networks were established to address the limitations inherent in traditional wired Local Area Networks (LANs) [3], which often presented significant challenges in terms of administration and maintenance, particularly in environments where mobility is essential. Initially designed to provide a flexible and accessible solution for internet connectivity, especially for mobile devices, Wi-Fi has undergone substantial evolution. It now plays a critical role beyond basic internet accessibility for computers and mobile phones, facilitating a diverse range of applications, including Internet of Things (IoT) systems [4], for smart home devices [5], and Machine-to-Machine (M2M) communications [6].

1.2 Exposure of data transmission in 802.11 networks

Wi-Fi operates by transmitting data through electromagnetic waves between a source and a destination. However, any device within the frequency range can detect these transmissions, even if it is not a participant in the communication [7]. This creates a significant security concern: since any device can potentially notice ongoing transmissions, it may attract unwanted attention to the communication process itself. This exposure

[1] In this thesis, the terms IEEE 802.11 networks, 802.11 networks and Wi-Fi networks will be used interchangeably.

elevates the risk of passive monitoring, traffic analysis, and active interference, ultimately threatening the confidentiality, integrity, and security of the transmitted data. One of the most significant risks associated with transmissions being detected by third parties is the potential for eavesdropping. Since transmissions are not restricted to physical cables, a malicious third party, often referred to as an attacker or *evil twin* [8], can easily intercept and monitor network traffic. This risk is particularly heightened in public Wi-Fi environments, where the identity of the network administrator may be unknown or unverifiable. As a result, the likelihood of such attacks increases considerably. An attacker could use *packet sniffing* tools to capture sensitive information [9], including personal data, passwords, and business communications, as they travel across the network.

Additionally, attacks such as *Man-in-the-Middle (MitM)* [10] are becoming increasingly common in wireless networks. In this type of attack, an attacker intercepts and potentially alters the communication between two legitimate parties without their knowledge or consent. This can result in the theft of sensitive data or the injection of malicious content into the communication, all while remaining undetected. Another concerning attack in 802.11 networks is *deauthentication* [11]. In this scenario, an attacker sends a deauthentication frame to disconnect a legitimate device from the network. This tactic can disrupt services or force users to reconnect to a rogue access point controlled by the attacker.

The exposure of Wi-Fi network transmissions makes them more vulnerable to *Denial-of-Service (DoS)* attacks [12]. Attackers can flood the network with malicious traffic or interfere with legitimate transmissions, thereby overwhelming the network's resources and rendering it inoperable. Broadcasting wireless signals can unintentionally reveal the presence of a network, thus increasing the risk of transmission detection. Consequently, anyone within range can inject disruptive signals to interfere with the ongoing transmission. Even if the data being transmitted is encrypted, the fact that a transmission is happening can attract the unwanted attention of a third party. *Traffic analysis* [13] becomes a possible strategy for these attackers; even if they cannot directly read the transmitted data, they can still discern important information about the communication, such as timing patterns, frequency, and volume of traffic. This reconnaissance can pave the way for further attacks.

1.3 Concealing data transmission in 802.11 networks

1.3.1 Network steganography

The term *steganography* comes from the Greek words *steganós*, meaning covered, and *graphia*, meaning writing or description [14]. Therefore, steganography translates to covered writing. One of the earliest recorded techniques of steganography dates back to ancient Greece in the 5th century Before Common Era (BCE), where the Greek tyrant Histiaeus devised a method to send a secret message to his son-in-law, who was serving in the Persian army. Histiaeus shaved the head of a servant, tattooed a message on his scalp, and allowed the hair to grow back, effectively concealing the message. Once the servant arrived at his destination, his hair was shaved, revealing the hidden message [15], [16]. Similarly, in ancient Rome, around the 1st century BCE, invisible ink was employed as a method of secret writing. Roman officials would use ink that was invisible under normal conditions but could be made visible through heat or other chemicals, providing a simple yet effective method for sending covert messages [15], [16].

Steganography, an ancient practice, has been adapted for modern networks to address the challenge of secure data transmission. This technique enables the discrete transmission of secret data in plain sight. Network steganography involves embedding hidden information within seemingly harmless media, such as images, audio files, or text, and sending it over a network in a manner that remains undetected by unauthorized entities. The primary goal of steganography is not to hide the transmission itself but to conceal the existence of data within the transmission. This differs from cryptography [17], which focuses on encoding messages to restrict access to specific recipients through methods such as shared secret keys. In contrast, steganography enables the transmission of secret data even if the regular communication is exposed or detected. The hidden information is embedded in such a way that its presence goes unnoticed, known only to the participants.

1.3.2 Covert channel definition

Network steganography is a technique used in computer networks to conceal a message within another explicit message, where the explicit message serves as a courier or envelope. A *covert channel* refers to the pathway through which this concealed information is transmitted. In computer networks, a covert channel exploits protocol features or procedures (either standardized or left open to interpretation). For instance, when a hidden message is embedded within unused or reserved fields in the packet header, that header field is the covert channel, while the embedding of the secret message represents a form of steganographic communication.

Covert channels are generally classified into three categories:

- Storage Covert Channels (SCC): These channels embed hidden messages within specific fields of a covert data carrier, typically a Protocol Data Unit (PDU), or a message associated with a particular network protocol.
- Timing Covert Channels (TCC): These channels exploit the timing patterns of message transmissions to convey hidden information. For example, variations in delay times between packets, changes in frame generation intervals, or random transmission intervals can be utilized to transmit concealed messages.
- Hybrid Covert Channels (HCC): These channels combine both storage and timing methods to transmit hidden messages simultaneously, leveraging both techniques for more robust and diverse covert communication.

1.4 Goals and thesis statement

This dissertation proposes a new taxonomic classification of covert channels in IEEE 802.11 networks by conducting an in-depth survey of existing implementations. It includes a systematic analysis of the underlying concepts, implementation techniques, and evaluation methods, and presents both existing and potential countermeasures. Since 2003, the field of 802.11 network steganography has steadily grown, with research publications regularly introducing new covert channel techniques. However, the available knowledge remains scattered across individual studies. To address this, we have collected and analyzed all known IEEE 802.11 covert channel implementations published by the research community. Based on this data, we present a structured classification, a conceptual map designed to guide newcomers in understanding the state of the art. This

classification not only highlights past contributions but also identifies areas that require further exploration and improvement, serving as a roadmap for future research.

After establishing the foundational knowledge, this dissertation also aims to contribute to the field of 802.11 network steganography by presenting four novel steganographic methods designed to conceal the presence of information during explicit transmissions and successfully deliver the secret message to the destination. Each method exhibits unique characteristics and is evaluated using a set of key metrics. In all cases, the analysis includes throughput, which measures how much covert data can be transferred in a single transmission and cumulatively over time, efficiency, which expresses the proportion of covert data successfully delivered under varying network conditions, transparency, which reflects the degree to which the covert channel operates without disrupting normal traffic, and resistance to steganalysis, which indicates how difficult it is for an adversary to detect the presence of the covert communication. Covertness, which evaluates how well covert traffic blends into regular network behavior, is not systematically addressed for all four methods; however, it is highlighted where applicable. Similarly, delay and jitter are considered only for methods where these metrics are relevant. In addition to quantitative evaluation, the practical applications of each covert channel are also discussed, illustrating potential use cases and the conditions under which they may be most effective. Notably, the proposed methods incorporate fundamental aspects of networking protocols, such as message retransmission mechanisms to enhance reliability and flow control to reduce congestion; they also leverage features introduced in recent IEEE 802.11 amendments. We believe that this dissertation provides a valuable contribution to the field of 802.11 network steganography and will help advance ongoing research in this domain.

The fulfillment of the objectives of the dissertation, along with the supporting results, provides the foundation for formulating and substantiating the following thesis statements:

- It is possible to create a covert communication mechanism that allows stations to transmit secret data while disconnected.
- It is possible to create a high-throughput, high-efficiency covert channel that camouflages secret data within an 802.11 management frame.
- It is possible to create a highly transparent covert channel to embed secret messages in a manner that mimics standard network behavior.
- It is possible to create an efficient mechanism that combines multiple covert channels to increase throughput and enhance resistance to steganalysis in Wi-Fi networks.

1.5 The structure of the thesis

The structure of this thesis is organized as follows:

In Chapter 2, we present the fundamental concepts required for understanding covert channel analysis. This chapter introduces the core concepts commonly used in covert channel implementations. A solid understanding of these foundational concepts is crucial, as they provide the basis and context for a clear understanding of the subsequent chapters and methods presented later in the dissertation.

Chapter 3 provides a comprehensive review of the existing literature on covert channels in 802.11 networks. It presents an overview of current techniques, covering their

design approaches, evaluation methodologies, and proposed countermeasures. This chapter serves as a roadmap for the research community, providing insight into the current state-of-the-art in 802.11 network steganography. It also highlights potential gaps and unexplored areas that may benefit from further investigation and development.

Chapter 4 introduces the first proposed covert channel. It describes the design, implementation, and evaluation metrics of a covert communication mechanism that enables stations to transmit data while remaining disconnected from the network. This covert channel presents an interesting paradox: it demonstrates that a station can avoid explicit network connections, avoiding authentication and association, yet still covertly transmit secret messages.

Chapter 5 introduces a second proposed covert channel, presenting a novel steganographic method that incorporates networking concepts such as reliability and sliding window techniques to enhance covert communication. In this chapter, we demonstrate how the adoption of both concepts significantly improves throughput and efficiency, especially in densely populated environments. This improvement is achieved by effectively mitigating the impact of interference caused by external traffic, minimizing its effects to a negligible level.

Chapter 6 introduces the third covert channel implementation, which demonstrates the opportunity of transmitting covert messages in a way that closely mimics the normal behavior of stations in an 802.11 network. This approach achieves high transparency and strong resistance to steganalysis, allowing it to operate virtually undetected. The chapter also analyzes various factors that influence the performance of such covert channels, providing insight into their practical limitations and opportunities for optimization.

In Chapter 7, we present the fourth and final covert channel. This chapter introduces a novel approach that marks a significant shift in the design and implementation of covert channels by demonstrating the possibility of combining two or more covert techniques. This multi-layered strategy increases overall throughput and distributes the covert data across different vectors, enhancing resistance to steganalysis. It represents one of the few methods in the literature that explores such a hybrid model.

Chapter 8 presents the conclusion of this dissertation. It highlights the most significant findings and contributions made to the field of 802.11 network steganography. The chapter also outlines potential directions for future research based on the results and gaps identified throughout the work. Finally, it summarizes the individual contributions of the author in the development and completion of the dissertation.

2 Fundamental 802.11 concepts for covert channel analysis

2.1 Physical (PHY) layer

2.1.1 Physical PDU frame structure

Understanding the Physical PDU (PPDU) format is crucial for effectively designing and implementing covert channels. In the context of non-HT (non-high-throughput) networks supporting Orthogonal Frequency-Division Multiplexing (OFDM) [18], the PPDU format significantly defines how data is transmitted over the air.

The structure of the 802.11 PPDU format for non-HT networks supporting OFDM, as presented in Figure 2.1 [19], consists of the following fields:

- Preamble: The PPDU begins with preamble fields that carry essential information on the transmission vector format. These preamble fields help synchronize the receiver with the transmitter and establish communication parameters.
  - Short Training (L-STF) and Long Training field (L-LTF): The L-STF and L-LTF together form a sequence of 12 OFDM symbols within the 802.11 Physical Layer frame. These fields play a crucial role in helping the receiver recognize the start of an 802.11 frame, thereby facilitating the synchronization of timers and frame reception processes.
  - SIGNAL (L-SIG): The L-SIG field within the 802.11 Physical Layer frame describes essential parameters such as data rate and frame length in bytes. Receivers utilize the information provided in the L-SIG field to calculate the precise time duration required for the transmission of the frame.
- Data: The data field carries the user payload and higher-layer headers. This payload may include data from upper layers such as MAC fields, Cyclic Redundancy Check (CRC), and any additional information required for data integrity and error checking.
  - Service: The service field is added at the beginning of the higher-level protocol data before transmission to initialize the data scrambler.
  - Physical Layer Convergence Protocol (PLCP) Service Data Unit (PSDU): The PSDU is a variable-length field that encapsulates a frame from the 802.11 MAC layer, encompassing the essential information for wireless communication.
  - Tail: Six bits of 0 are used to reset the state of the convolutional encoder.
  - Pad: This field carries no meaningful user data or significance for frame processing. It is included to ensure that the data field contains multiple OFDM symbols. These additional bits ensure that the total bit count aligns precisely with the symbol requirements of the transmission medium. Because the pad has no impact on the actual transmission, it is sometimes used for covert channels, as it is generally ignored during data transmission.

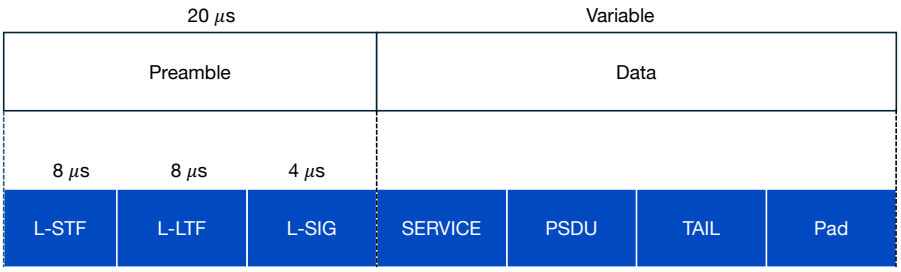


Figure 2.1: 802.11 non-HT PPDU frame structure

2.1.2 Modulation schemes

The 802.11 specification primarily utilizes OFDM as a radio carrier technique. OFDM is a multiplexing method widely employed in wireless communication systems to transmit data over radio frequencies. It subdivides the available frequency spectrum into multiple orthogonal subcarriers, facilitating efficient data transmission with reduced interference and enhanced spectral efficiency. The orthogonality of these subcarriers ensures that they do not interfere with each other despite occupying the same frequency range. By utilizing orthogonal subcarriers, OFDM can simultaneously transmit multiple parallel data streams, improving overall data throughput. Each OFDM subcarrier carries a modulated symbol representing a segment of the overall data. Modulation encodes information by altering one or more signal parameters, such as phase, amplitude, or frequency. Various digital modulation schemes [20], including Quadrature Amplitude Modulation (QAM), Phase Shift Keying (PSK), and Frequency Shift Keying (FSK), are used to encode digital data into symbols transmitted over the subcarriers. These modulation schemes map digital data into symbols represented on a constellation diagram. This graphical depiction illustrates the relationship between the amplitude and phase of each symbol, as shown in Figure 2.2. The in-phase (I) component corresponds to the part of the signal aligned with the reference carrier, while the quadrature (Q) component is orthogonal to it, being shifted by 90°. Each point corresponds to a unique symbol, defined by a particular combination of amplitude and phase, known as a constellation point. Figure 2.2 illustrates Quadrature Phase Shift Keying (QPSK), a type of digital modulation in which each subcarrier symbol is modulated by changing the phase of the carrier signal to one of four possible values, which are depicted as points on the constellation diagram.

The transmitted symbols are susceptible to noise and imperfections from the transmitter, the channel, and the receiver. These factors can disrupt the positions of symbols in the constellation diagram by introducing random disturbances to the signal's amplitude and phase. Such disturbances cause the symbols to deviate from their ideal locations, leading to errors during demodulation. Higher noise levels exacerbate these deviations, increasing the risk of symbol misinterpretation and degrading overall signal quality. Noise and imperfections throughout the entire data transmission process contribute to what is known as a noisy channel, resulting in the presence of distorted symbols or dirty constellations. Many covert channels in the PHY layer exploit these distortions and the constellation diagram to conceal secret information. They disguise the secret data as part of the imperfections, typically ignored or undetected during the signal processing phase.

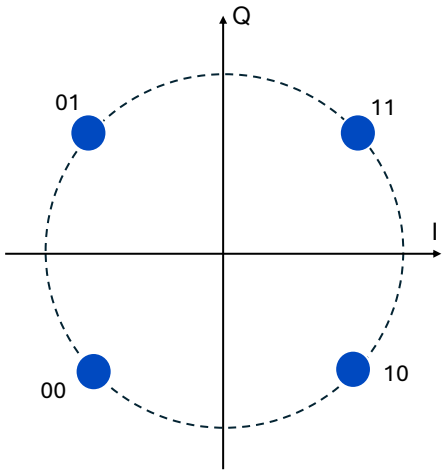


Figure 2.2: An I/Q diagram, also known as a constellation diagram, with QPSK, which uses four phase states (0°, 90°, 180°, and 270°)

2.1.3 Modulation and coding scheme (MCS)

In IEEE 802.11 networks, the MCS defines the data transmission characteristics of a wireless link [21]. Each MCS index corresponds to a specific combination of parameters, including the modulation type (e.g., BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM), coding rate (e.g., 1/2, 3/4, 5/6), and the number of spatial streams. These parameters determine the efficiency, reliability, and throughput of wireless transmission. The MCS index directly influences the physical data rate, which can be calculated using the Equation 2.1 [22]:

$$\text{Data Rate} = \frac{N_{\text{SD}} \cdot N_{\text{BPSCS}} \cdot R \cdot N_{\text{SS}}}{T_{\text{DFT}} + T_{\text{GI}}} \tag{2.1}$$

Where:

- N_{SD} – Number of data subcarriers (depends on channel bandwidth)
- N_{BPSCS} – Number of coded bits per subcarrier per stream (based on modulation)
- R – Coding rate (fraction of useful bits)
- N_{SS} – Number of spatial streams
- T_{DFT} – OFDM symbol duration
- T_{GI} – Guard interval duration

The values of N_{BPSCS}, R, and N_{SS} are defined by the selected MCS index, which directly influences the data rate. Higher MCS indices typically involve higher-order modulations and coding rates, resulting in increased throughput, but also requiring better channel conditions to maintain error resilience.

2.1.4 Multiple input multiple output (MIMO)

MIMO technology is a transmission technique that utilizes multiple antennas at both the transmitter and receiver. MIMO enables the simultaneous transmission of multiple

independent data streams over the same frequency channel. This technique, known as spatial multiplexing, significantly enhances both throughput and reliability without requiring additional spectral resources. Beyond throughput gains, MIMO also improves link robustness through beamforming. This method focuses transmission energy in the direction of intended receivers, improving signal quality, reducing interference, and extending coverage, especially beneficial in complex or obstructed environments [23].

Mathematically, a MIMO system is modeled as in Equation 2.2:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{2.2}$$

where $\mathbf{y}$ is the received signal vector, $\mathbf{x}$ is the transmitted signal vector, $\mathbf{H}$ is the channel matrix that contains the fading coefficients between the transmit and receive antennas, representing how much of the transmitted signal passes through each possible path between the transmit and receive antennas. The higher the value of a coefficient, the stronger the signal transmitted along that path, and $\mathbf{n}$ is the noise vector.

The Gain expressed in the Equation 2.3 quantifies the maximum spatial multiplexing achievable in a MIMO system:

$$\text{Gain} = \min(N_t, N_r) \tag{2.3}$$

where N_t and N_r are the number of transmit and receive antennas, respectively.

2.2 Medium access control (MAC) layer

2.2.1 MAC PDU frame structure

The MAC layer in 802.11 plays a crucial role in managing how devices share and access the wireless medium. It is responsible for coordinating transmission opportunities among multiple users to minimize collisions and ensure efficient use of the channel. Each 802.11 MAC PDU (MPDU) frame consists of at least the following elements: header, frame body, and Frame Check Sequence (FCS), as illustrated in Figure 2.3 [24]. A realistic MPDU is depicted in Figure 2.4 (QoS data frame) with its respective values. Researchers explore aspects that are not addressed or left open for interpretation, leveraging this ambiguity to create covert channels. For example, custom header fields, reserved bits, or fields with flexible usage can be manipulated to embed secret messages without disrupting normal network operations.

The header begins with the *frame control* field, which, upon expansion, displays its subfields, which are the following (Figure 2.4):

- Version: According to the 802.11 standard, the default and only supported version value of the protocol is 00. The values of 01, 10, and 11 are reserved for future versions.

- Type: Specify the type of WLAN frame, distinguishing between control frame (for acknowledgment of the successful delivery of data frames, managing channel access and providing the reliability of the MAC layer), data frame (to carry data from higher network layers) and management frames (enable nodes to connect or disconnect from the network and handle network discovery, authentication, and association processes).

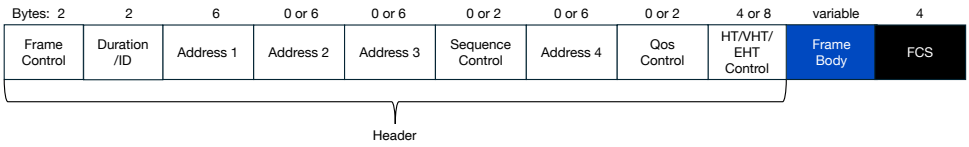


Figure 2.3: 802.11 MPDU frame structure

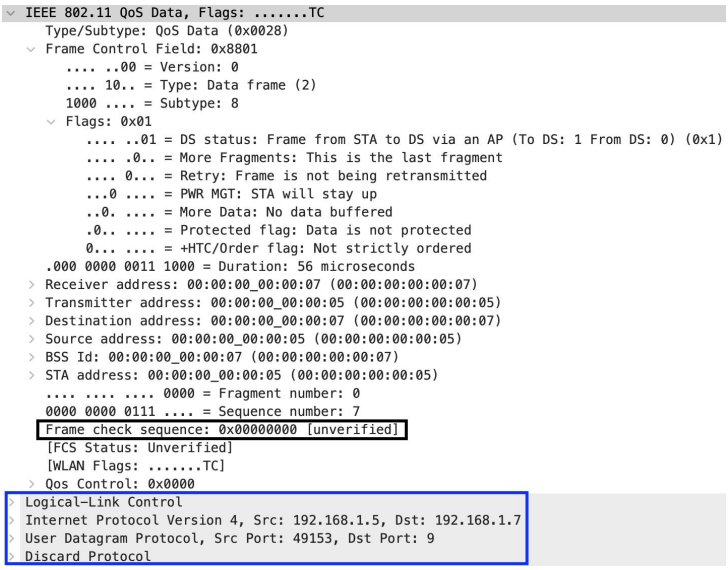


Figure 2.4: 802.11 MPDU frame captured from Wireshark

- Subtype: Working with the type field to identify the specific frame function. For example, a frame can have a management type, and its subtype can indicate whether it is a beacon (network advertisement), a probe request (network discovery), or a probe response.

- To DS and From DS: The To DS and From DS indicate the direction of the data frames relative to the Distribution System (DS), which is the infrastructure used to interconnect the Access Point (AP) with other networks. When a frame is sent to the DS (toward an AP), the To DS field is set. In contrast, when a frame comes from the DS (from an AP), the From DS field is set. The control and management frames always have both fields set to zero. Both fields are zero in Independent Basic Service Set (IBSS) topology (communication through a central device typically an AP) or adhoc networks (client to client communication without an AP).

- More Fragments: Set when a packet is divided into multiple frames for transmission, indicating that all frames except the last one in a packet will have this bit set.

- Retry: Used when frames require retransmission, this bit is set to one when a frame is being retransmitted, aiding in eliminating duplicate frames.

- Power Management: Indicates the power management state of the sender after completing a frame exchange. Access points manage connections and never set the power saver bit.

- More Data: Used in buffering frames received in a distributed system, the More

Data bit assists access points in supporting stations in power saver mode. It indicates the availability of at least one frame and addresses all connected stations.

- Protected: When set to 1, this bit indicates that the frame is protected using the standard 802.11 security mechanisms (e.g., WPA2, CCMP) [25].
- Order: This field is set when utilized to ensure that frames and fragments are sent and processed in strict order. The received frames must be handled in the exact transmission order when set.

After frame control, the *Duration/ID* field indicates the amount of time the client station (STA) will occupy the channel for frame transmission. Alternatively, in Power Save Poll (PS-Poll) frames, it contains the STA's Association Identifier (AID) and is used by the STA to request buffered data from the AP that was held during its sleep period.

The *Address fields* (Addresses 1-4) are the MAC addresses of the frame source and destination, and usage depends on the values of the To DS and From DS sub-fields. Address 1 is typically the receiver (or destination) address, the address 2 is the transmitter (or source) address, the address 3 is often the Basic Service Set ID (BSSID), typically the AP address or the final destination address, and the address 4 is used in Wireless Distribution System (WDS) frames to carry an additional address, such as the original sender.

The *Sequence Control* manages packet sequencing in fragmented transmissions, and the first 4 bits indicate the fragment number (to identify a fragment within the MPDU) and the remaining 12 bits the frame sequence number (to manage packet sequencing, particularly in fragmented packets).

The *QoS Control* field is included in frames when QoS functionality is enabled, as specified in the 802.11e amendment [26]. This field carries information used to differentiate traffic priorities and manage Quality of Service for time-sensitive applications. Additionally, enhanced transmission technologies introduce further control fields into frames when their respective features are active, which is the case for a High Throughput (HT) control when 802.11n is in use, a Very High Throughput (VHT) in 802.11ac networks, the High Efficiency/Extremely High Throughput (HE/EHT) in 802.11ax and the upcoming 802.11be.

The *Frame Body* carries higher-level protocol data. Unlike Management and Control frames, Data frames include a Frame Body containing the data payload, and the *FCS* field consists of a 32-bit CRC error detection sequence to guarantee frame integrity [27].

2.2.2 802.11 connection establishment

Unlike wired networks, where the plugging of the Ethernet cable typically grants immediate network access, 802.11 wireless networks require the STA to exchange information with the AP before gaining full access[1]. As described in Figure 2.5, the process unfolds in three stages. Initially, the STA is in an unauthenticated and unassociated state (State 1). Here, the STA initiates network discovery through passive or active scanning. *Passive scanning* involves listening to beacon frames broadcast by APs announcing the network's

[1]802.11 supports two different network setups: The Basic Service Set (BSS) mode, in which an AP is responsible for creating, announcing, and managing client associations, and the Independent Basic Service Set (IBSS) mode, where there is no AP and each station communicates directly with others. This thesis considers only the BSS scenario.

presence. The beacon frame is sent periodically and advertises network information, including the Service Set Identifier (SSID), commonly referred to as the network name, beacon interval, timestamp, capability information, and supported rates. In contrast, *active scanning* involves the STA sending a probe request frame to search for available networks and awaiting a response (Figure 2.5 right side).

Upon receiving the probe request, the AP checks if the information provided aligns with the requirements of the network. If so, the AP responds with a probe response detailing the network information. Following network discovery, the STA sends an authentication request that provides its credentials. If the credentials match, the AP sends an authentication response indicating success, transitioning the STA to an authenticated but unassociated state (State 2). To acquire full access to a network, the STA issues an association request. Once the request is accepted, the AP responds with an association response, assigning the STA an association ID, and the STA transitions to an authenticated and associated state (State 3), which grants the STA complete access to the network.

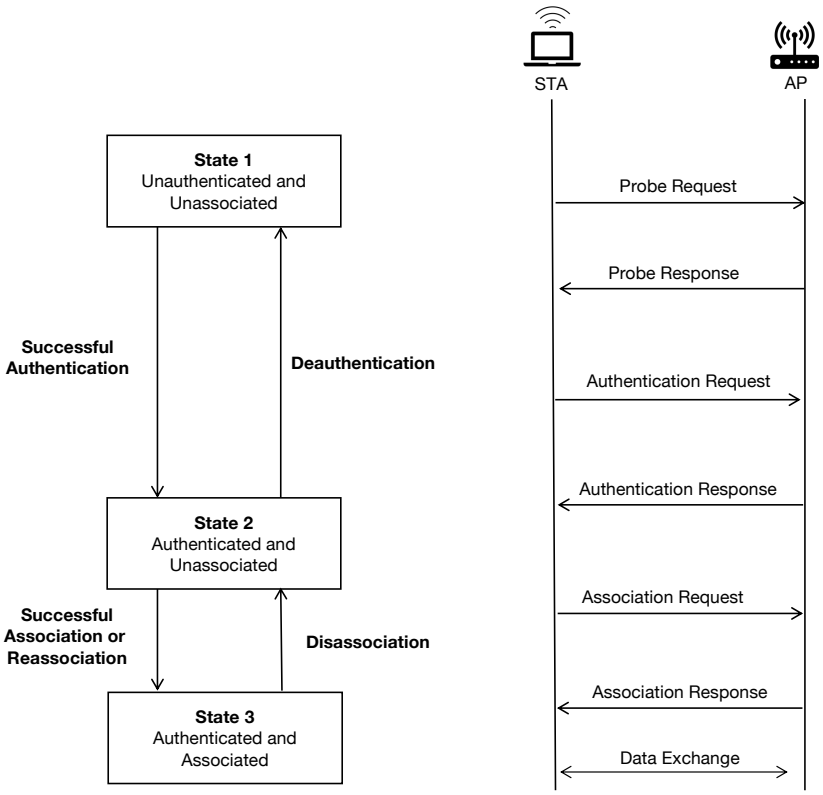


Figure 2.5: 802.11 connection establishment. State machine (left) and frame exchange (right)

2.2.3 MAC address randomization

When the STA performs the active scanning, it does so in two ways: direct scanning, when the STA specifies the SSID of the target network in the probe request; or indirect scanning, when the SSID field is left empty (wildcard SSID), signaling that the STA is searching for any available network in the area. In both cases, the STA includes its

globally unique MAC address in each probe request it transmits, as presented in Figure 2.6.

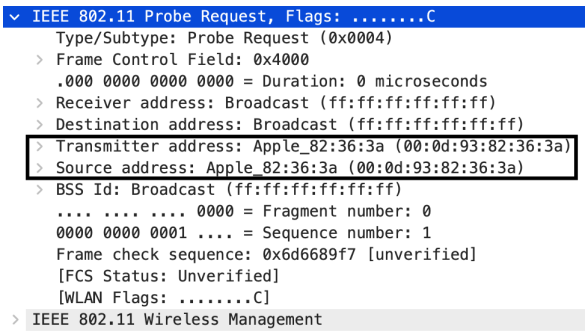


Figure 2.6: Captured probe request frame showing the transmitter's globally unique MAC address

Every IEEE 802.11-compliant device is assigned a distinctive MAC address, which serves as a layer 2 identifier. This address acts as the unique source identifier for any frame transmitted over a wireless medium, ensuring that frames can be correctly routed by receiving devices. A standard MAC address consists of 48 bits (6 bytes), and its global uniqueness is maintained through a hierarchical allocation system managed by the IEEE Registration Authority (IEEE RA). Specifically, the first 24 bits (3 bytes) of a MAC address, known as the Organizationally Unique Identifier (OUI), are assigned to hardware vendors. Each vendor then generates a unique 24-bit identifier to distinguish individual Network Interface Controllers (NICs). The combination of the OUI and the vendor-assigned identifier forms the full 48-bit MAC address, uniquely assigned to each device, as illustrated in Figure 2.7. Also, as per Figure 2.7, the MAC address structure has the following semantics: In the first octet, the Least Significant Bit (LSB) designates whether the address is unicast or multicast. Additionally, the second LSB indicates whether the address is universal or locally administered.

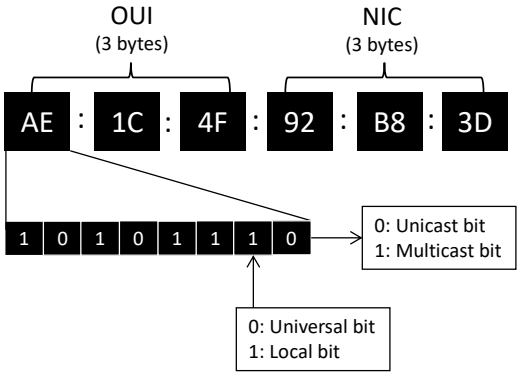


Figure 2.7: MAC address structure with OUI, NIC, and control bits interpretation

By default, most devices supporting Wi-Fi connectivity are configured to perform active scanning. This behavior typically persists, unless explicitly disabled by the user through system settings. Active scanning introduces privacy concerns, as it involves transmitting broadcast frames that contain the unique identifier of the device, the MAC address. Since the MAC address in the probe request frames is sent in plaintext, any

nearby device capable of monitoring wireless traffic can intercept and record this information. Over time, repeated broadcast of a consistent MAC address allows observers to track the movements or presence of a device, potentially linking it to a specific user or behavior pattern. This opens the door to unauthorized surveillance, user profiling, and targeted advertising by third parties, including public Wi-Fi providers, marketers, or even malicious entities. The real-life examples are the research conducted that aimed to demonstrate how much information is exposed and can be retrieved through the probe requests and the MAC addresses it advertises.

The research [28], highlights the potential for long-term device surveillance using persistent MAC address identifiers. The first method, known as the beacon replay attack, involves an adversary gathering SSIDs from specific geographic areas, such as residential neighborhoods, and later broadcasting spoofed beacon frames containing those SSIDs in different environments, like offices or public venues. This deceptive broadcast induces nearby devices to respond with probe request frames, thus disclosing their MAC addresses. To highlight critical privacy concerns related to the probing behavior of devices enabled by Wi-Fi and the exposure of their unique identifiers, the study [29] many devices transmit probe requests at a notably high frequency, with the transmission rate increasing in proportion to the number of known SSIDs stored on the device. In some cases, devices were found to emit around 55 probe requests per hour, considerably increasing the risk of identifier exposure. This persistent broadcast of unique MAC addresses enables malicious actors to collect and analyze data, thus constructing detailed user profiles and tracking individuals across various locations with impressive accuracy; such surveillance is usually carried out without the knowledge or consent of the individuals involved, as also emphasized in studies [30]–[32]. Another area where MAC addresses are being used is to facilitate geolocation by linking them to known physical locations through methods such as Wi-Fi scanning and database lookups, where monitoring tools record the advertised MAC addresses and cross-reference them with a location. Mobile operating systems contribute by collecting MAC addresses, signal strength, and Global Positioning System (GPS) coordinates, creating extensive databases accessible for location estimation. Geolocation Application Programming Interfaces (APIs), such as Google, use MAC addresses to estimate device locations, and integrating Wi-Fi data with GPS information on devices such as buses allows the mapping of user movements [33]–[35].

Exposure to a user's MAC address can also pose physical security risks by enabling location tracking and surveillance. When a device broadcasts its MAC address, especially on public Wi-Fi networks, malicious actors or tracking systems can monitor and record the user's physical movements and routines [36]. If the MAC address is linked to an individual, this information could be used to stalk, identify their home or workplace, or plan targeted physical threats. Although a MAC address alone is not typically personally identifiable, in combination with other data, it can compromise a user's physical safety, necessitating mitigation strategies to protect individual privacy and safety.

MAC address randomization is a privacy and security preserving technique that helps protect users by periodically generating temporary and disposable MAC addresses in place of the device's permanent, globally unique identifier. This approach mitigates the risk of device tracking and user profiling by decoupling network identifiers from physical hardware. As shown in Figure 2.7, randomized MAC addresses are typically identified by setting the second LSB of the first byte to indicate a locally administered address.

Major Operating System (OS) vendors began introducing MAC randomization around 2014 - 2016. Apple Inc. added support in iOS 8 (2014) [37], Linux incorporated it from

kernel 3.18 (2014) [38], Android followed with version 6.0 Marshmallow (2015) [39], and Microsoft introduced it in Windows 10 (2016) [40]. Since then, the implementations have become more robust. Microsoft enables MAC randomization by default in Windows 11, allowing users to choose between using a single randomized MAC address for all networks or a unique one per SSID. In iOS, since version 14, randomization is also active by default, with each SSID associated with a persistent randomized MAC address, which only changes if the network is forgotten and rejoined. Android 10 and later follow a similar pattern, maintaining a consistent random MAC per SSID unless the network is forgotten. On Linux, many distributions now support per SSID random MAC addresses, generating new ones when reconnecting to forgotten networks. MAC address randomization techniques continue to evolve as privacy concerns and tracking methods become more sophisticated. Since their initial implementation, major operating system vendors have progressively refined their approaches to enhance user anonymity and resist tracking mechanisms. At the time of writing, each vendor may have updated its randomization logic to reflect these advances and the implementations that may vary between OS versions and device models [41], [42].

MAC address randomization is inconsistently implemented across devices. Some devices randomize only the last 24 bits while using a fixed OUI, whereas others randomize the entire 48-bit address [43]. As of the time of writing this thesis, no official standard for MAC randomization exists; however, several best practices have been proposed [44].

2.2.4 Supported rates in 802.11 scanning

A standard probe request frame in IEEE 802.11 is structured as any regular MPDU presented in Figure 2.3. In the header, the frame type is a management frame, with a subtype probe request header, followed by a body composed of several Information Elements (IEs) that reveal the station's capabilities, and the FCS as presented in Figure 2.8. The probe request is transmitted from the STA to all the stations within the frequency range, targeting any AP. It is an active search from the STA side to find the list of available networks to join.

For instance, Figure 2.9 shows a probe request captured using Wireshark. In the frame body, the SSID field typically carries a wildcard value to indicate that the station is open to discovering any network. Conversely, if the SSID contains a specific value, the station is conducting a targeted search for that particular network. The following field is the *supported and extended supported rates* field, which enumerates the data rates that the station can employ for communication with an access point. Each entry in this field is an 8-bit value, where the first 7 bits indicate the actual data rate, and the Most Significant Bit (MSB) signals whether that rate is mandatory (basic rate) or optional. If a station supports more than eight data rates, it can supplement this list with extended supported rates.

2.2.5 Channel access mechanisms in 802.11

IEEE 802.11 defines three primary channel access mechanisms: The Distributed Coordination Function (DCF) is the fundamental contention-based access method, with a randomized backoff procedure to minimize collisions [45], [46]. The Point Coordination Function (PCF)[47], which provides a contention-free mechanism by having access points poll stations for transmissions and is rarely used in commercial systems. The Enhanced

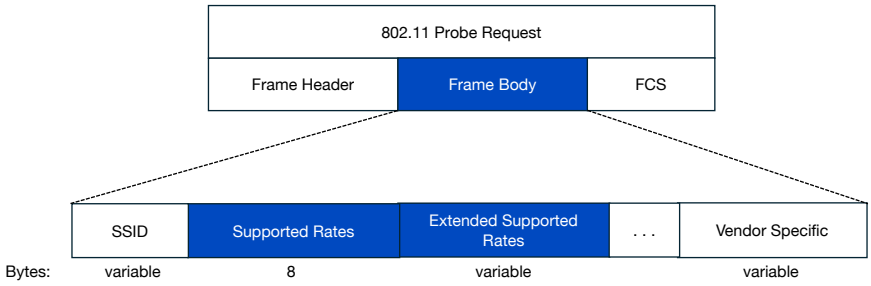


Figure 2.8: 802.11 probe request frame structure

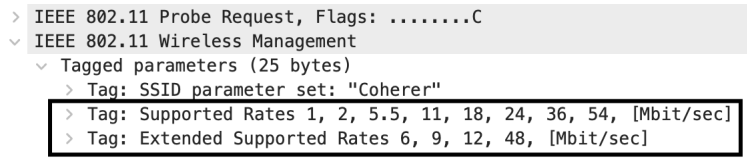


Figure 2.9: 802.11 probe request body captured from Wireshark, with listed supported rates and extended supported rates

Distributed Channel Access (EDCA) [48], introduced in 802.11e, EDCA builds on DCF by adding QoS features that prioritize different types of traffic. This allows for improved performance for latency-sensitive applications. In this chapter, we focus on DCF due to its widespread use and its significance in covert channel implementations.

In any shared communication medium, the risk of data collisions arising from multiple devices attempting to transmit simultaneously can lead to degraded performance, information loss, and reduced overall efficiency. To maintain order and reliability in such environments, collision avoidance or mitigation mechanisms are essential. In wired networks such as Ethernet, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is employed [49]. This method allows devices to detect collisions during transmission and respond by stopping and retrying after a random delay. However, this approach is not viable in wireless networks, where it is often impossible for a device to detect a collision while transmitting, due to limitations such as signal fading and the hidden node problem.

To address these challenges, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used in 802.11 networks [50]. Rather than detecting collisions after they occur, CSMA/CA aims to prevent them from happening. The design of CSMA/CA makes it particularly well-suited for wireless communication, where devices operate in a shared and often unpredictable medium. Its collision avoidance approach respects the constraints of radio-based systems, where simultaneous transmission and listening are typically infeasible. By minimizing the chances of interference before transmission occurs, CSMA/CA helps ensure more stable and efficient communication in wireless networks.

The principles of CSMA/CA form the foundation of the DCF, which is the default channel access method defined in the IEEE 802.11 standard. DCF implements CSMA/CA in practice, enabling wireless devices to compete for access to the shared medium in a decentralized manner, without the need for a central device acting as a coordinator. The IEEE 802.11 standard employs DCF as its fundamental access mechanism, allowing multiple devices to contend for use of the wireless medium in an asynchronous manner. DCF implements the CSMA/CA technique through two forms of carrier sensing: physical and virtual. Physical carrier sensing involves measuring the energy level on the channel to determine if it is currently in use, while virtual carrier sensing uses the Network Allocation

Vector (NAV), which provides an estimate of how long the medium will remain occupied based on observed transmissions.

The DCF procedure, illustrated in Figure 2.10, operates as follows. When a station has a frame to transmit, it first performs carrier sensing. If the medium is sensed idle, the station waits for a DCF Inter-Frame Space (DIFS). At this point, it either resumes a previously paused backoff counter or, if no valid counter is available, draws a new random backoff from the range $[0, CW]$, where CW is the current contention window.

The contention window begins at a minimum value, $CW_{\min}$, and doubles exponentially after each failed transmission attempt, up to a maximum value $CW_{\max}$. This update is defined in Equation 2.4:

$$CW = \min\left(2 \cdot CW + 1,\ CW_{\max}\right) \tag{2.4}$$

This ensures that the backoff space grows but never exceeds the maximum. After each successful transmission, for example, when an acknowledgment (ACK) is received, the contention window is reset to $CW_{\min}$.

During the backoff period, the station decrements its counter by one slot time whenever the channel is sensed idle. If the channel becomes busy, the countdown is paused and resumes only once the medium has been idle for a DIFS period. When the backoff counter reaches zero, the station transmits its frame. If no ACK is received, the transmission is considered failed: the retry counter is incremented, the contention window is increased according to Equation 2.4, and a new backoff is drawn. If the retry counter exceeds its limit, the frame is dropped, and the contention window is reset. By combining random backoff, carrier sensing, and inter-frame spacing rules, the DCF reduces the likelihood of collisions while maintaining fairness among all contending stations [46].

Despite the effectiveness of the DCF mechanism, it is still subject to two well-known limitations: the *exposed and hidden node* problems [51]. The exposed node problem occurs when a device unnecessarily holds back its transmission after detecting activity on the channel, mistakenly assuming that its transmission would cause interference. This often results in an underutilization of the available bandwidth. For example, a station that overhears a nearby transmission may assume the entire channel is occupied, even if its intended recipient is not within range of that ongoing communication and would not be affected. In contrast, the hidden node problem occurs when two stations attempt to communicate with the same access point but are outside each other's transmission range. Since they cannot detect each other's transmissions, they may attempt to send data simultaneously, resulting in collisions at the access point. This scenario is particularly problematic in wireless environments, where the inability to detect all nearby transmitters can result in significant performance degradation due to repeated collisions and retransmissions.

To mitigate hidden and exposed node problems, the 802.11 standard introduces the Request-to-Send (RTS) and Clear-to-Send (CTS) mechanisms, as illustrated in Figure 2.11. When a station has data to transmit, it first sends an RTS frame to its intended recipient. If the receiver is available, it replies with a CTS frame. Both RTS and CTS frames include a duration field that reserves the channel for the entire exchange, including Short Inter-Frame Space (SIFS), data, and ACK. Stations that overhear the RTS update their NAV, while those that only hear the CTS also defer their transmissions for the remaining duration. Once the current transmission ends, stations with pending frames resume contention through the backoff procedure.

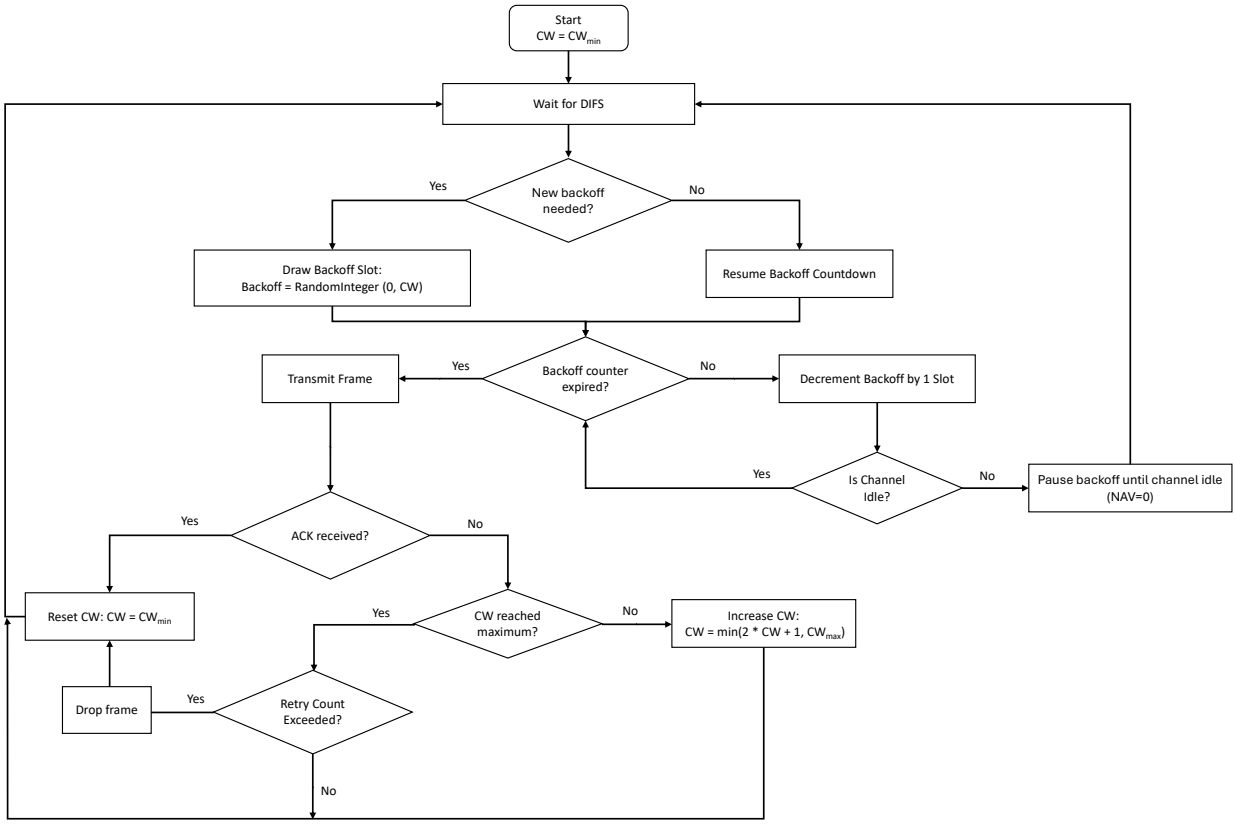


Figure 2.10: 802.11 DCF with random backoff procedure

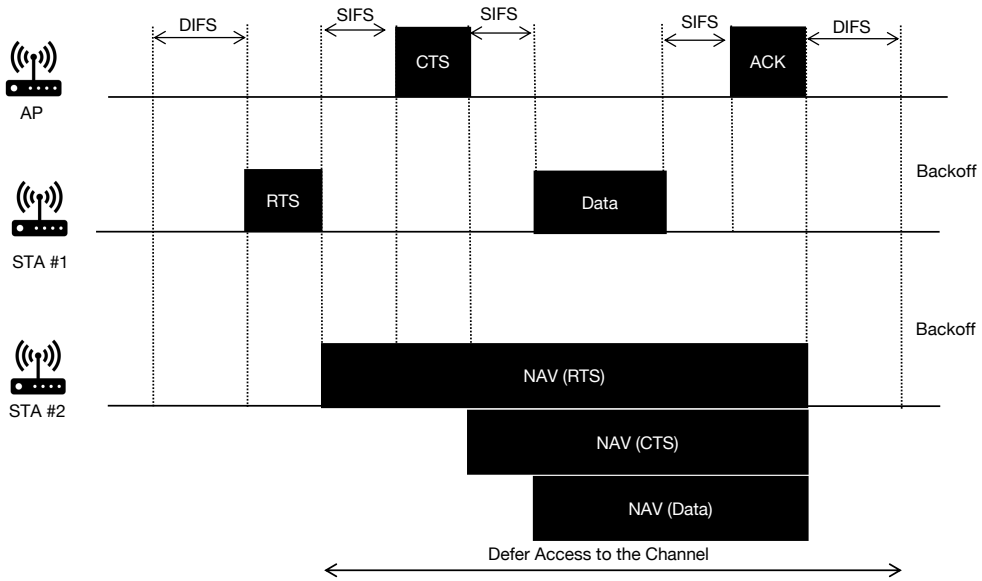


Figure 2.11: Transmission of a data frame with RTS/CTS mechanism

2.2.6 QoS in 802.11 networks

Traditional IEEE 802.11 wireless networks were originally designed to operate on a best-effort delivery model, where all data frames, regardless of type or application, were treated with the same transmission priority. Although this model was sufficient for early applications, such as web browsing and file transfers, it is inadequate for modern latency-sensitive

services, including Voice over IP (VoIP), video conferencing, and real-time gaming. In such scenarios, treating all traffic equally results in increased contention, which can lead to excessive delay, jitter, and packet loss, particularly when the network is congested.

To overcome these limitations, the IEEE introduced EDCA (802.11e amendment), which extends the DCF, adding QoS support [48]. These enhancements allow traffic to be classified and prioritized based on its sensitivity to delay and reliability requirements. QoS mechanisms are now an integral part of the 802.11 standard and are essential to support various types of traffic that coexist and compete for access to the shared wireless medium. The QoS in 802.11 networks is implemented through traffic classification into predefined access categories (ACs): Voice (AC_VO), Video (AC_VI), Best Effort (AC_BE), and Background (AC_BK). Higher-priority traffic (e.g., voice) gains faster access to the medium compared to lower-priority traffic (e.g., background), as a result of lower CW and Arbitrary Inter Frame Space (AIFS) values. An additional QoS feature introduced with 802.11e is the Transmission Opportunity (TXOP), which defines a time interval during which a station can transmit multiple frames without needing to contend for the channel again. This mechanism enhances channel efficiency, ensuring that high-priority traffic can maintain consistent transmission with reduced delays and overhead.

To support QoS at the frame level, 802.11e introduces a field in the MAC header known as the QoS Control field, as presented in Figure 2.3. This field carries QoS-related information that informs how the frame should be handled by the receiving and transmitting stations. The QoS Control field is 16 bits in length and is composed of the following subfields, also demonstrated in Figure 2.12:

- Traffic Identifier (TID): Specifies the traffic priority or AC.
- End of Service Period (EOSP): Indicates the end of a service period in power-saving modes.
- ACK Policy: Defines the acknowledgment method (e.g., normal ACK, no ACK, or block ACK).
- Payload Type: Indicates whether the MAC payload is an MSDU or an A-MSDU.
- TXOP Duration Requested or Queue Size (STA), and AP Buffer Status or Power Save Info (for AP).

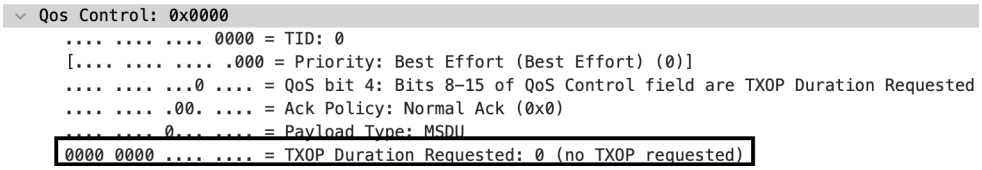


Figure 2.12: 802.11 QoS control field from a STA, with highlight to the TXOP Duration Requested subfield

In the QoS control, crucial to this thesis, it is essential to emphasize the *TXOP Duration Requested* subfield. The STA uses the TXOP Duration Requested to indicate the amount of time it requires for its next TXOP, associated with a specific TID. This value informs the AP about the duration that the STA expects to occupy the medium

during its subsequent transmission. A zero value in this field means that the station does not request a TXOP for the indicated TID within the current service period. Conversely, a nonzero value represents an explicit request for a specific duration of channel access. These values are not cumulative; any new TXOP duration value replaces any previously indicated request for the same TID. If the transmission queue for that TID becomes empty, the STA may use a zero value to cancel a previously requested TXOP. It is important to note that the AP is not obligated to grant the requested full duration. It may assign a shorter TXOP depending on network conditions, policy, or fairness in scheduling.

2.2.7 Frame aggregation

In IEEE 802.11 networks, each data frame incurs considerable overhead due to the inclusion of a MAC header, PHY preamble, interframe spaces (such as SIFS and DIFS), ACKs, and other control frames. When transmitting small payloads, this per frame overhead becomes significant, resulting in lower throughput as more airtime is consumed by protocol overhead and control signaling rather than useful data. Additionally, an increase in frame transmission frequency leads to higher contention and collisions, which in turn trigger more backoff periods, causing greater transmission delays, increased jitter, reduced fairness among stations, and higher energy consumption, especially for battery-powered devices that must remain active for frequent and short transmissions.

A resolution to this inefficiency is frame aggregation, which involves combining multiple small frames into a single transmission unit [52]. Frame aggregation was introduced in the IEEE 802.11n standard [53], [54], and was further enhanced in 802.11ac to help achieve high data rates, including gigabit-level throughput [55]. This technique enables the transmission of multiple MAC Service Data Units (MSDUs) or MPDUs in a single aggregated frame, thereby significantly reducing protocol overhead and improving channel efficiency. To support this aggregation mechanism, a more efficient acknowledgment strategy was introduced: the Block Acknowledgment (Block ACK or BA). Unlike traditional ACK, which responds to each frame individually, Block ACK allows a receiver to acknowledge multiple data frames in a single control frame. This is initiated by a Block ACK Request (BAR) sent by the transmitter, which prompts the receiver to respond with a Block ACK frame indicating the reception status (received or missing) of each frame in the aggregated transmission. This selective acknowledgment mechanism reduces the overhead of the control frame and supports the reliable delivery of aggregated frames, thus enhancing overall network performance. The 802.11 standard defines two types of aggregation: Aggregated MSDU (A-MSDU) and Aggregated MPDU (A-MPDU), as well as a hybrid approach that combines both.

The MSDU is the service data unit delivered from the upper layers to the MAC layer, typically containing a network-layer packet, for example, an Internet Protocol (IP) datagram. It represents the payload to be transmitted before being encapsulated into a complete MAC frame. To improve transmission efficiency, especially in scenarios involving numerous small MSDUs, the 802.11 standard introduces Aggregated A-MSDU as one form of frame aggregation. *A-MSDU* allows multiple MSDUs, destined for the same receiver, to be combined into a single MPDU as presented in the Figure 2.13. This aggregation reduces the per packet overhead by sharing a single MAC header and minimizing the number of individual frame transmissions, thus decreasing processing load and airtime consumption. Each MSDU within an A-MSDU is encapsulated with

its subframe header consisting of three fields: the Destination Address (DA), the Source Address (SA), and the length of the MSDU. To ensure proper alignment, padding is added so that each subframe is a multiple of 4 bytes. However, the limitation of A-MSDU is that the entire aggregation is encapsulated into a single MPDU, which contains only one FCS at the MPDU level. As a result, if any individual MSDU within the A-MSDU is corrupted, the integrity check will fail for the entire MPDU. This means that even if only one subframe is affected by transmission errors, the entire A-MSDU and thus the entire MPDU must be retransmitted, rather than just the corrupted MSDU.

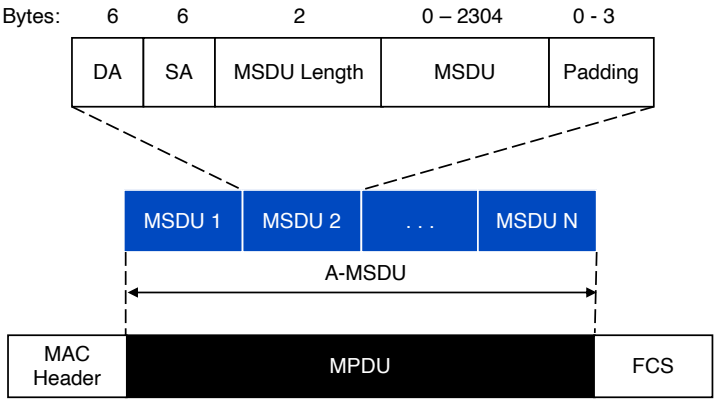


Figure 2.13: Aggregation of MSDUs into an A-MSDU and encapsulation into an MPDU

The MAC layer constructs the MPDU for transmission over the wireless medium. It encapsulates a single MSDU, or in the case of A-MSDU, an aggregated group of MSDUs. The MPDU is the complete frame that is handed off to the Physical (PHY) layer and is subject to standard 802.11 channel access, framing, and retransmission procedures. To further enhance efficiency, the 802.11 introduces *A-MPDU*, a technique that allows multiple MPDUs to be transmitted in a single PHY-layer transmission. Each MPDU within the A-MPDU retains its MAC header and FCS, enabling independent error detection and selective retransmission at the MPDU level as depicted in Figure 2.14. A delimiter precedes each MPDU and may be followed by padding, and unlike A-MSDU, where the whole aggregate is retransmitted if any part is corrupted, A-MPDU uses Block ACKs to selectively retransmit only failed MPDUs, reducing overhead and improving throughput.

In addition to supporting A-MSDU and A-MPDU as separate aggregation methods, the 802.11 standard allows their combined use in a multi-level aggregation scheme. In this approach, A-MSDUs are encapsulated within MPDUs, which are then aggregated into a single A-MPDU, as illustrated in Figure 2.15. The complete A-MPDU consists of multiple such MPDUs transmitted as a single aggregated frame. This layered aggregation offers greater flexibility and more efficient use of the wireless medium, especially in high-throughput environments. In the hybrid structure, each MPDU in the A-MPDU can carry one A-MSDU, which itself contains multiple MSDUs. This method combines the reduced overhead of A-MSDU with the selective retransmission capability of A-MPDU, enhancing overall performance.

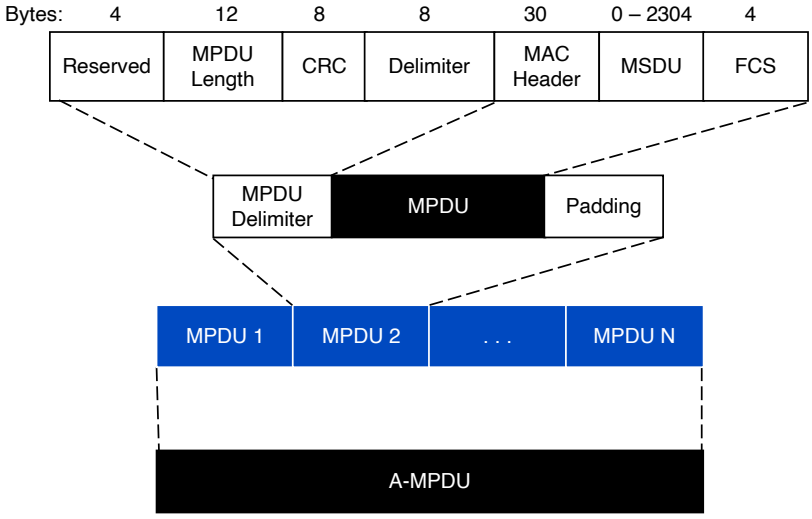


Figure 2.14: Format of A-MPDU aggregation

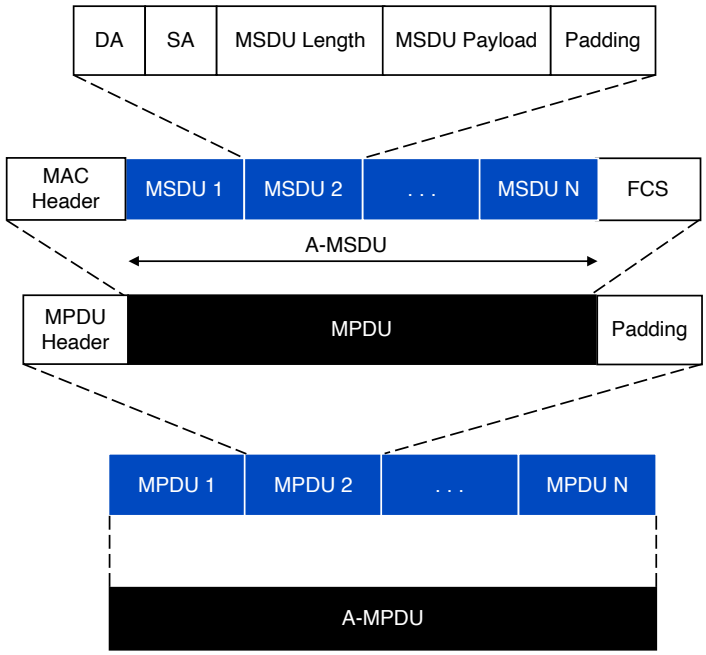


Figure 2.15: Multi-level frame aggregation in IEEE 802.11, the A-MSDUs within MPDUs aggregated into an A-MPDU.

3 Evaluation of covert channel techniques in IEEE 802.11 Networks

This chapter presents a comprehensive analysis of the estimated 51 covert channels for 802.11 networks. Each covert channel is individually examined and categorized according to a novel taxonomic classification dedicated to the 802.11 networks, as presented in Figure 3.1. The classification is organized along two main levels: the layer in which the channel operates, MAC or PHY, and the method used for encoding the hidden data, distinguishing between storage, timing, or hybrid techniques. In addition, the taxonomy identifies the specific field, mechanism, or behavior that is altered or exploited to enable covert transmission, such as custom fields, packet timing, backoff procedures, or physical signal characteristics.

Unlike general classifications, this scheme is dedicated exclusively to 802.11 covert channels, highlighting not only the structural layer and encoding type, but also where and how the covert channels are implemented. This thesis also analyzes their chronological distribution, prevalence across protocol layers, encoding strategies, and validation methods, distinguishing between theoretical proposals and practical validation methods. Practical implementations range from software-based simulations to real hardware deployments.

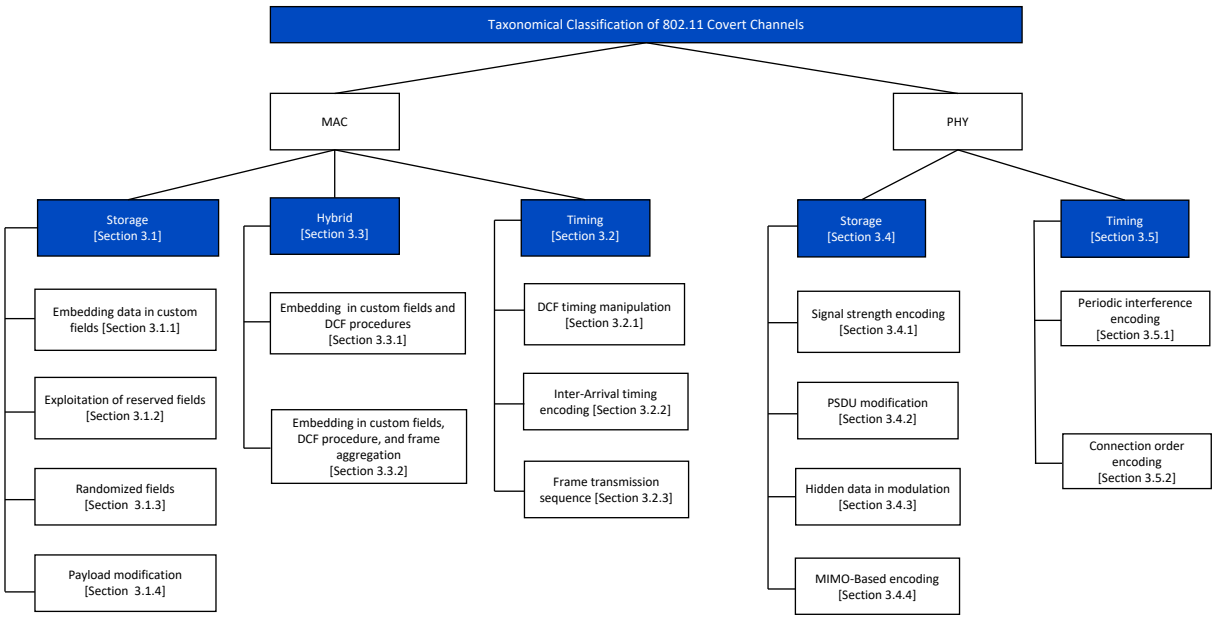


Figure 3.1: Proposed taxonomic classification of covert channels in 802.11 networks

3.1 802.11 MAC storage covert channels

3.1.1 Embedding data in custom fields

According to available information, the Hidden Communication System for Corrupted Networks (HICCUPS) [56] is believed to be the first covert channel designed for 802.11 networks. The author of the system identified three Hidden Data Channels (HDCs).

HDC1 utilizes cipher initialization vectors for data hiding within the Wired Equivalent Privacy (WEP) protocol, employing Rivest Cipher 4 (RC4) initialization vectors. HDC2 leverages MAC network addresses for data hiding. HDC3 is based on integrity mechanism values such as frame checksums using CRC-32. In a shared Wi-Fi medium, the stations that participate in the covert channel are grouped into a hidden group. The covert channel operates in two modes: Basic Mode, which utilizes low-bandwidth channels (HDC1 and HDC2) to exchange control messages among the stations in the hidden group. Corrupted Frame Mode - in this mode, frames with intentionally corrupted checksums (HDC3) are exchanged among stations within the hidden group. The sender intentionally introduces incorrect checksum values known and processed only by the stations in the hidden group. Unaware of this mechanism, stations outside this group discard frames with incorrect checksums, assuming that they are corrupted according to the 802.11 standards.

The Stego PR [57] involves covertly embedding hidden information within probe request frames, specifically within the SSID field. Typically, this field contains a list of network names to which the station attempts to connect. In this method, the sender inserts the probed network names into the SSID field and transmits them in plaintext hexadecimal format. The hidden information is encoded in the last 4 LSB of the plaintext network names. For instance, to encode the hidden message LIGHT, the station should populate the SSID field with the network names AP-FALL, AP-WIFI, BIG-BANG, SPOT-HIGH, and WIFI-GUEST. It is essential to note that the sequence of each network name is taken into consideration.

Among the covert channels proposed in [58], one utilizes the 16-bit Sequence Control (SC) field. The SC comprises two subfields: sequence number (12 bits) and fragment number (4 bits). To avoid suspicion, the method uses the first 8 bits of the sequence number for covert messaging, while the remaining 4 bits are used for frame sequencing. For example, to encode the character B (ASCII code 66), the sequence number field would appear as $01000010****$, with the asterisks representing the 4 bits reserved for the frame sequence number. The transmission of the covert message through the sequence control field occurs in three phases. Firstly, a predefined sequence signals the start of the covert transmission. Secondly, the sender transmits a frame to indicate the length of the expected message. Lastly, the source sends frames that contain the hidden message within the sequence control field. For instance, to send "Hello every body!" 19 frames are used: one frame signals the start, one conveys the length, and 17 contains the message.

A covert channel leveraging the 802.11b MAC rate switching protocol [59] to transmit hidden authentication messages to APs is proposed in [60]. This approach prevents attackers from detecting user authentication activities and protects user credentials from malicious software attacks. The core concept behind this covert channel involves mapping message bits to corresponding data rates. The establishment of the covert channel using the rate-switching protocol requires the following steps: 1) Initialization - each station calculates a One-Time Password (OTP) using the initial key and a nonce, a unique value for each authentication session. 2) One-Time password generation - the OTP is generated using a cryptographic hash function on the previous key concatenated with the nonce. This prevents repeated transmission of the same password and enhances security against replay attacks. 3) Watermark encoding - the station encodes the OTP as a sequence of available data rates, referred to as a watermark. For example, in 802.11b, four data rates (1 Mbps, 2 Mbps, 5.5 Mbps, 11 Mbps) represent bit combinations (00, 01, 10, 11). 4) Authentication frequency - the administrator determines the authentication frequency, which in turn influences the security level. More frequent watermarks enhance

security but increase network overhead. The system uses a pseudo-random cycle length to determine when watermarks are sent, making it difficult for adversaries to predict. 5) Synchronization and decoding - the MAC sequence identifiers synchronize the client and access point. The access point stores received data rates in a watermark buffer and authenticates the client by comparing the decoded watermark with the expected OTP. The designer of the underlined covert channel presented an extended version with two main application scenarios [61], specifically for covert authentication and a Wi-Fi botnet, providing performance and covert channel accuracy for both scenarios.

In their research, Teca et al. introduce an innovative covert communication mechanism that encodes data within probe request frames by targeting the supported rate and extended supported rate fields of the 802.11 protocol [62]. An STA uses the supported rate field in a probe request frame to advertise its compatible data rates for potential network connections. Each data rate value is 8 bits long, with the last bit, known as the MSB, indicating whether the rate is mandatory or optional. If the STA supports more than eight data rates, it utilizes the extended supported rates field to include additional data rates. The essence of the research lies in developing a covert channel that allows STAs to transmit secret data to AP without undergoing authentication or association procedures. This covert channel exploits the control over the MSB within the supported rates field. For each data rate value, the STA sets the MSB to either zero or one, depending on the message it intends to transmit.

Blanco et al. have presented a unique method in [63] for establishing a covert communication channel by exploiting the Windows native Wi-Fi API. This API allows userland processes to manipulate specific fields within Wi-Fi management frames. The covert sender listens to and processes beacon and probe response frames to gather information, such as a BSS list. The covert message is encoded within the SSID field and custom Information Elements of the probe request frames. Additionally, the BSSID field and the first few bytes of the SSID field can include a signature and flags to identify and manage covert communication. Subsequent Probe Requests carry the covert message data, segmented into the SSID field and custom Information Elements.

A high-throughput covert channel for 802.11ad Directional Multi-Gigabit (DMG) networks has been proposed in [64], leveraging Forward Error Correction (FEC) [65]. This covert channel concept emerged following the amendment of the IEEE 802.11-2016 standard [66], which introduced additional MCS indices to the single carrier PHY. The covert channel intentionally selects a lower MCS index than the channel conditions would typically support. For example, if the channel supports a 7/8-rate code (i.e., MCS 9.1), the covert channel selects a 13/16-rate code (i.e., MCS 9) instead. This intentional selection leaves the first 48 parity bits available for hidden data.

Belhamra et al. [67] introduce a steganographic scheme for wireless multicast communications using the MAC-independent Opportunistic Routing & Encoding (MORE) protocol, which leverages Random Linear Network Coding (RLNC). The covert channel utilizes the transfer matrix M_T of the MORE protocol, exploiting its random nature to hide secret messages during each transmission phase. The sender encodes the hidden data within the coefficients of the linear combinations used to generate coded packets. The sender divides the secret message M into blocks and arranges them into a vector S_{γ^2} suitable for embedding. It then creates two triangular matrices, L (lower triangular) and U (upper triangular), using the blocks of the secret message. The product $M_T = LU$ is used as the transfer matrix for the current batch of packets. This matrix generates random linear combinations and encodes the secret message within the coefficients of

the linear combination. The receiver of the covert data is the subsequent hop, and it performs a LU decomposition on the transfer matrix M_T to retrieve the hidden message. The sender extracts the original secret message by concatenating the parts retrieved from the L and U matrices.

3.1.2 Exploitation of reserved fields

The covert channel proposed in [68] utilizes the Protocol Version (PV) field of the MAC header in CTS and ACK frames. The PV field consists of 2 bits and is typically set to 00. The covert channel leverages the remaining combinations $\{01, 10, 11\}$ to transmit hidden messages. The transmission begins with five frames where the PV field is set to 01. This sequence signals the start of the covert message transmission. The actual message is encoded in binary, where a binary 0 is represented by 10, and a binary 1 is represented by 11 in the PV field. The transmission concludes with another sequence of five frames with the PV field set to 01, indicating the end of the covert message. To illustrate the process, consider sending the message A as an 8-bit ASCII character (binary 01000001). The PV field of the transmitted frames would be set as $\{01, 01, 01, 01, 01\}$ (start message delimiter), then $\{10, 11, 10, 10, 10, 10, 10, 11\}$ (secret message), and finally $\{01, 01, 01, 01, 01\}$ (end message delimiter).

In the research [69], the authors propose a covert channel based on the 802.11e specification, which starts upon association or reassociation. The process involves using the QoS capability field, which the QoS Station (QSTA) sends during these requests. This field includes the QoS, CF-Pollable, and CF-Poll fields, each 2 bits long. These three fields combine to form 8 possible combinations, with the last three combinations $\{101, 110, 111\}$ are reserved by the 802.11 standard. The covert communication process is as follows: 1) Initiation - the covert communication begins when a QSTA sends an Association Request frame with the QoS, CF-Pollable, and CF-Poll fields set to 101. The sender requests to be placed on the pollable list by sending another Association Request with the QoS, CF-Pollable, and CF-Poll fields set to 010. This signals the start of covert communication to the intended receiver. 2) Data Transmission - once the communication is established, secret data is transmitted using the QoS Control fields, specifically through the TXOP and TID subfields within regular data frames. 3) Termination - the covert communication ends when a Reassociation Request frame with the CF-Pollable and CF-Poll fields set to 110 is sent, signaling the end of the covert channel.

3.1.3 Randomized fields

Teca et al. have developed a new covert communication channel based on the MAC address randomization scheme [70]. In regular Wi-Fi operations, devices continuously scan for available networks by sending out probe requests that include their unique MAC addresses, posing a threat to user privacy. To address this issue, modern Wi-Fi devices now use random MAC address generation to enhance user anonymity. This new covert channel leverages the disposable nature of these random MAC addresses to transmit secret messages covertly. Although these addresses may seem disposable and random to regular users, they can convey encoded secret messages among participants. The authors outlined three distinct operational scenarios: one without retransmission, another with retransmission, and a third that integrates an adapted Sliding Window Protocol (SWP) [71].

The second proposal in [58] introduces a covert channel based on the Initialization Vector (IV). The IV field is a three-byte random value used by the RC4 encryption algorithm in WEP-secured networks. This covert channel operates similarly to the first proposal, which uses the Sequence Number field. However, it becomes more efficient when WEP is enabled, as the payload is embedded within the IV field. The communication process is established as follows: 1) Start Signal - a predefined sequence of digits is transmitted to signal the start of the covert communication. This helps the receiver identify the beginning of the hidden message. 2) Message length - the IV field in the first frame carries the length of the hidden message. This informs the receiver about the number of subsequent frames to expect for reconstructing the message. 3) Data transmission - each subsequent frame's IV field carries three characters (each character has the size of one byte) of the hidden message, making this method more efficient than the one using the sequence control field, which only carries one byte per frame.

3.1.4 Payload modification

A covert channel based on frame modification is proposed in [72]. The primary concept involves replacing the contents of specific fields in the MAC header (such as padding bits or reserved fields) with values that encode covert messages. Implementing this covert channel involves three key operations: 1) Channel/AP selection - the sender uses a configuration file that outlines the criteria for selecting a WLAN channel and AP. One such criterion is the presence of frames with the retry bit flag set to 1 (the covert channel targets retransmitted packets). This helps ensure that the selected channel meets the necessary conditions for covert communication. 2) Frame filtering - frames must meet the conditions specified in the configuration file to be considered candidates for the covert channel. The sender filters out frames that do not match these criteria, ensuring that only suitable frames are used for embedding the hidden messages. 3) Embedding the hidden message - during this phase, the content of the MAC header is replaced with a new header containing the hidden message. The fields to be replaced are specified in the configuration file and shared between the transmitter and the receiver to maintain synchronization and ensure correct message extraction.

3.2 802.11 MAC timing covert channels

3.2.1 DCF timing manipulation

Covert DCF, as presented in [73], exploits the randomness of backoff by intelligently selecting specific CW values that correspond to predefined symbols from a codebook. These symbols represent bits or bits of strings that form the covert message. Creating this covert channel involves several steps: 1) Share a codebook - both the sender and receiver share a codebook, essentially a map associating characters with bit strings. Each bit string is associated with a specific backoff value. 2) Message creation - the sender STA creates a message it intends to send, ensuring that every character in the message corresponds to those listed in the codebook to ensure correct detection. 3) Traffic-Fitting symbols insertion - the original message is incorporated with traffic fitting symbols, each having the same number of bits, to create the data transmitted through the WLAN channel. 4) Backoff sequence generation - instead of randomly selecting a backoff value as standard practice dictates, the sender manipulates the backoff interval according to

the message it intends to send using the codebook. 5) Backoff detection - the receiver monitors the backoff values selected by the sender and employs a detector to reverse the process, retrieving the original message content.

To enhance the data transmission rate of the covert channel presented in [73], a code-based timing approach is introduced in [74]. This approach leverages the random backoff intervals of the DCF protocol combined with the Exploiting Modification Direction (EMD) method used in Joint Photographic Experts Group (JPEG) steganography. The primary goal is to create a covert channel that achieves high data rates and security by embedding hidden messages within the timing patterns of packet transmissions. The transmission involves the following steps: 1) Acquiring the distribution of free time intervals - the hidden sender and receiver monitor the free time intervals in the wireless channel to estimate their distribution. This involves sensing the channel to determine periods free from transmissions. For instance, assume the observed intervals are $\{5, 10, 15, 20\}$ μs. 2) Symbol encoding - the sender uses the acquired values representing the free time interval distribution and maps each symbol in the hidden message to a specific sum of free time intervals. For example, the symbol 0001 is mapped to a backoff of 25 μs. The message is encoded by setting the backoff value to a length n μs, where n is the sum of the selected free time intervals between two consecutive data transmissions. For example, the sender can achieve this 25 μs backoff by choosing a combination of the observed free time intervals: Sends the first frame and backs off for 10 μs, sends the next frame and waits for 15 μs, and finally, send the last frame.

In a paper [75], the authors propose a timing covert channel based on scenarios similar to the prisoner's problem [76]. In this setup, a warden oversees covert communication attempts between a sender and a receiver. The concept involves manipulating the backoff intervals between packet transmissions to embed hidden messages within the timing patterns of these transmissions, effectively creating a secret communication channel to bypass the warden. The covert channel works as follows: 1) Codebook agreement - the sender and the receiver share a codebook that maps symbols to specific backoff values. Symbol transmission - the sender encodes a message by choosing backoff values corresponding to the symbols in the message. For example, to send the secret message CAB, the sender uses the backoff periods associated with each character of the message as $\{C, 0.3 \text{ ms}\}$, $\{A, 0.1 \text{ ms}\}$, $\{B, 0.7 \text{ ms}\}$ sequentially.

Based on the DCF mechanism [77], the ternary covert channel operates by having both the sender and receiver monitor the channel to collect free time intervals for statistical analysis. This analysis determines the distribution of free time intervals, which is then utilized to conceal a covert message. The time distribution is divided into three subsets, indicating that the message is ternary, with trits 0, 1, and 2 being transmitted, each carrying $\log_2 3$ information. The process of sending a covert message involves the following steps: 1) Acquire free time interval distribution - both the sender and receiver sense the channel using signal power in the physical layer and NAV in the MAC layer to acquire the time interval distribution of the channel H. Each divides the time intervals into three subsets: for the sender, H_S^0, H_S^1, and H_S^2 represent the time distributions for sending trits 0, 1, and 2, respectively. In contrast, for the receiver, H_R^0, H_R^1, and H_R^2 represent the time distribution of decoding trits 0, 1 and 2, respectively, with each set consisting of n elements. 2) Send secret message - the sender selects the time interval based on the trit it intends to send. For example, if the sender wants to transmit trit 0, it selects the backoff as the sum of the free time intervals belonging to set H_S^0 and set H_R^0 to allow the receiver to decipher the hidden message properly. This relationship between sender and

receiver is expressed in Equation 3.1:

$$\sum_{i=0}^{n} H_S^i \in H_R^i \tag{3.1}$$

If, for instance, the receiver intends to send trit 0, and the time interval 25 belongs to H_R^0, the sender waits for 25 free time slots between two consecutive data transmissions. This involves waiting for 15 free time slots after the first transmission and 10 free time slots after the second transmission.

A recent study proposes a new covert communication method for Smart Grid (SG) environments using 802.11ax technology [78]. Smart Meters (SMs) are vital components of SG infrastructures, providing crucial data to the SG network's data processing core. However, traditional Wi-Fi-enabled SMs face challenges in securely transmitting sensitive data over insecure networks and authenticating incoming data from other SMs. The proposed covert channel aims to address these weaknesses, offering a solution for secure authentication and confidential data transmission. This covert communication framework enhances communication between SMs and data collectors within specific SG networks. The covert channel utilizes the 802.11 standard's backoff procedure by encoding secret messages within the parity of backoff slots. Even backoff slots represent binary 0, while odd slots signify binary 1. Operating at a bandwidth of one bit per frame, any node within the network, whether a station or an AP, can act as a recipient.

The spyware, described in [79], is designed to secretly transfer chip data from the host platform through a covert channel. The spyware's components are integrated into the host's circuit to exploit the DCF using the backoff mechanism of the CSMA/CA protocol. The spyware is not always active; external or internal signals trigger it. The covert channel induces specific inter-frame delays between consecutive frame transmissions to encode secret messages. The sender adjusts the timing of these transmissions based on the secret message, creating distinguishable timing patterns. For example, different intervals might represent different bits or groups of bits. The receiver, aware of the spyware's presence and the encoding scheme, measures the time intervals between frames and decodes the hidden messages using the pre-agreed encoding and decoding schemes.

3.2.2 Inter-arrival timing encoding

The article [80] proposes a covert channel that modulates management frames' interarrival times, specifically beacon frames or probe request frames, to encode data. The authors present two methods for encoding the message: the value of the interarrival time and the difference between successive interarrival times. Modulation of interarrival time - the covert channel embeds information by varying the time intervals between successive beacon frames or probe request frames. Specific intervals (in milliseconds) are mapped to specific bit sequences. For example, a 5 ms transmission interval might be mapped to the bit sequence 000, while an interval of 10 ms might be mapped to 001. Transitions between time intervals - this method encodes data by altering the time between frame n and frame $n + 1$. Different transitions between two time intervals can represent different bit sequences. For instance, a transition from 5 ms to 10 ms could be mapped to 001, while transitions from 10 ms, to 15 ms could be mapped to bit sequences to 000.

In [81], the authors propose two covert channels that operate on periodic timing by leveraging the beacon frame, precisely the 64-bit timestamp field. The first covert channel modifies the LSB of the timestamp field. Before sending the beacon frame, the AP fills up

to 59 bits with the timestamp value, reserving the last 4 bits for the covert message used in STA authentication. The receiving AP then extracts these last four bits from each beacon frame's timestamp field and concatenates them to form the authentication string. The second covert channel exploits the periodicity of the beacon frame. Assuming beacon frames are generated every 100 ms, with each subsequent frame's timestamp increasing by 100 ms, the second covert channel subtly alters the timestamp of the next beacon frame. The receiver compares the received timestamp to the previous one. For instance, if the current timestamp is less than the previous, it decodes bits 10; otherwise, it decodes 00. Ultimately, the receiver constructs the authentication sequence based on these decoded bits.

In the study described in [82], researchers introduced a covert communication method using the beacon frame and beacon interval. In this proposed covert channel, secret messages are encoded by intentionally introducing a time delay, denoted as Δ (in microseconds), in the beacon interval generation. This delay is either added (to represent bit 1) or subtracted (to indicate bit 0) from the standard beacon interval of 102.4 ms, effectively embedding the covert information.

The proposed algorithm [83] improves the performance and robustness of [82] by addressing packet loss and performance. The method leverages the Interpacket Delay (IPD) between transmitted packets in 802.11 networks. It brings an improvement of 0.46 bps, and enhanced robustness of the covert channel is achieved by extending its capability to recover from consecutive packet losses. For instance, if T is the IPD and α represents a delay. Then T_+ and T_- signify $T + \alpha$ and $T - \alpha$ respectively. The proposed algorithm for transmitting covert bits operates as follows:

- If the first bit is 0, the covert sender transmits a packet with T_-.
- If the first bit is 1, the sender transmits a packet with T_+.
- If the next bit is the same as the previous bit, the sender transmits the packet with an IPD of T.
- If the next bit differs from the previous bit, the sender initiates a bit change process.

The second covert channel described in [72] conceals information by duplicating frames. To encode binary values, the sender duplicates packets and uses the Retry bit in the 802.11 MAC header. The Retry bit indicates whether a packet is being retransmitted. For example, a duplicated packet with the Retry bit set could represent a binary 1, while a regular packet (without duplication) could represent a binary 0. The transmission process consists of three main steps: 1) Initialization - The communication begins by synchronizing the sender and receiver using a specific packet sequence that includes the AP's address. This step is crucial for the receiver to interpret the timing patterns accurately. 2) Data encoding - the actual data is encoded in the timing intervals between packet transmissions and the state of the Retry bit. The sender embeds binary data within the normal traffic flow by carefully controlling these intervals and the Retry bit. 3) Error correction application - error correction codes are applied to the transmitted packets to ensure the receiver can correct any errors introduced during transmission.

CHAOS [84] is a timing-based covert channel technique designed for Wi-Fi networks, which embeds hidden information in beacon frames. It exploits two core mechanisms: manipulation of the Timing Synchronization Function (TSF) and controlled reordering of beacon transmissions. Typically, each beacon contains a 64-bit TSF timestamp that

increments predictably due to the periodic nature of beacon broadcasts. CHAOS encodes covert bits by introducing subtle and controlled deviations into these timestamp values, carefully shaped using a Gaussian distribution to imitate natural clock drift and avoid detection. In environments with multiple APs, where beacon frames from different BSSIDs may arrive concurrently, CHAOS additionally encodes data by intentionally altering the order in which these beacons are transmitted. The receiver, operating in monitor mode, passively captures beacon frames and decodes the covert message by analyzing the relative timing of TSF values and the order of beacon arrivals. These two encoding methods operate in tandem to increase bandwidth while preserving high covertness. In terms of performance, the TSF-based encoding alone achieves approximately 300 bps, order-based encoding reaches around 180 bps, and the combined CHAOS mode yields up to 520 bps, making it suitable for low-rate broadcast communication in dense or public Wi-Fi environments.

3.2.3 Frame transmission sequence

Sawicki et al. introduced a method [85] that adds a new rule for how participant stations access the medium, along with the backoff time. This method involves at least two stations and one access point. Data is hidden within the transmission order of the stations, and for this to function correctly, the stations must synchronize with each other using the information provided in beacon frames generated by the AP. The covert channels transmit information in the following manner:

- When transmitting bit 0, STA A sends the first frame after the Beacon frame, before any transmissions from STA B. STA B delays its transmissions until STA A completes its frame transmission.

- When transmitting bit 1, STA B sends the first frame after the Beacon frame before any transmissions from STA A. STA A delays its transmissions until STA B completes its frame transmission.

3.3 802.11 Hybrid covert channels

3.3.1 Embedding in custom fields, and DCF procedure

Natkaniec et al. created a new covert channel in [86] that operates at the MAC layer, combining timing and storage methods to enhance previous work [78]. In particular, it is the first covert channel to support QoS. This method encodes secret messages using the DCF mechanism (timing) and the duration/ID field in the MAC header (storage). The covert channel encodes binary data based on the sender's backoff time within the DCF mechanism. Specifically, an even number of slots since the last transmission represents a 0, while an odd number represents a 1. Furthermore, the secret message is hidden in the duration/ID field of the MAC frame using the three least significant bits (bits 12-14, with the 15^{th} bit always set to 0. This approach allows the duration value to be adjusted by up to 7 μs from the norm. The minimal alteration in the duration value does not affect the overall network performance, as it remains shorter than the SIFS.

3.3.2 Embedding in custom field, DCF procedure and frame aggregation

The research in [87] introduces StegoEDCA, a multilayered high-throughput covert channel framework adapted to smart grid networks, which combines four complementary covert channels to balance covertness and bandwidth. The covert channels are as follows: StegoTXOP, which embeds up to 3 bits by varying the number of aggregated frames transmitted within a TXOP period. Each number of MPDUs within the A-MPDU is mapped to a particular bit sequence, and after encoding, it is placed in the MPDU Duration/ID field. The StegoBackoff, which encodes 1 bit using the parity of the DCF backoff slot, and StegoDuration, which hides 1 bit in the 13th bit of the Duration/ID field.

3.4 802.11 PHY storage covert channels

3.4.1 Signal strength encoding

A covert channel has been developed for secure military communication over 802.11 networks, as detailed in the study [88]. This alternative mechanism allows two stations within a BSS to switch to a new channel if the main channel is disrupted or when transmitting critical data. The covert channel facilitates direct communication between nodes A and B, bypassing AP and rendering it undetectable to potential threats. The process begins with Node A requesting direct transmission parameters to Node B (frame #1) through the AP. This frame contains the MAC addresses for Node A (source address), Node B (destination address), and the BSS address (AP address). Node B, within the same range as Node A, receives two frames: Frame #1 from Node A and frame #2 forwarded by the AP. If the signal strength of frame #1 exceeds a predetermined threshold, Node B responds with frame #3. Node A receives two frames from Node B: frame #3 directly and frame #4 through the AP. Node A determines the Received Signal Strength Indicator (RSSI) from frame #3, confirming that both nodes are within the same frequency range and have adequate signal quality. Once this direct communication link is established, the nodes operate in ad-hoc mode. Additional robustness is achieved by implementing scrambling, and the Variably Modified Permutation Composition (VCMP) encryption algorithm [89]. The process is further complicated by scrambling, which consists of modifying the sequence of fragments by assigning different sequence numbers to them. As a result, an unintended receiver cannot process the fragments or decode the message properly since they are out of order. Only the covert receiver knows the correct order to reconstruct the message accurately.

Shadow Wi-Fi [90] is a covert channel that integrates Software-Defined Radio (SDR) capabilities into smartphones. Researchers have modified the firmware of Broadcom Wi-Fi chips to enable them to transmit raw In-phase and Quadrature (IQ) samples. This modification allows the Wi-Fi chips to operate in ways not typically accessible through standard firmware, creating a new method for covert communication. The process unfolds as follows. 1) Prefiltering frames - outgoing Wi-Fi frames are pre-filtered to embed secret information. This involves subtly modifying the phase of specific subcarriers within the frames to mimic the natural fading effects of the wireless channel, making the covert channel difficult to detect. 2) Channel State Information (CSI) extraction - CSI is extracted per frame, capturing detailed information about the wireless channel between the transmitter and receiver. This information includes the phase and amplitude of each sub-

carrier, which is crucial for decoding the embedded covert information at the receiver's end. The CSI reflects the modifications made by the prefiltering process at the transmitter. 3) Covert Communication process - the transmitter embeds covert symbols by altering the phases of specific subcarriers in the Wi-Fi frames. These phase changes are designed to be subtle and do not interfere with the frames' normal communication function. At the receiver end, the CSI decodes the embedded covert symbols. The receiver analyzes the phase information from the CSI to identify and extract the covert data embedded in the frames.

AIR-FI [91] is a covert channel to extract information from Air-Gapped networks [92]. Unlike conventional methods, AIR-FI does not rely on Wi-Fi hardware within the air-gapped systems. Instead, it utilizes Double Data Rate Synchronous Dynamic Random-Access Memory (DDR SDRAM) buses to induce electromagnetic emissions in the 2.4 GHz Wi-Fi bands, and nearby Wi-Fi-capable devices can receive and decode modulated data, subsequently transmitting it to the attacker via the Internet. The attack model encompasses three primary stages: 1) Infecting the Air-Gapped network - attackers infiltrate the air-gapped network using Advanced Persistent Threats (APTs) through various vectors, including supply chain attacks, USB drives, social engineering, or insider threats. 2) Infecting Wi-Fi devices - the attacker needs to infect Wi-Fi-capable devices near the air-gapped network. These devices could include smartphones belonging to visitors or employees, desktop and laptop computers equipped with wireless networking, or IoT devices featuring Wi-Fi transceivers. Wi-Fi hardware and software vulnerabilities, as well as network protocols, are exploited for this purpose. 3) Data exfiltration - the attacker gathers desired data from compromised computers, such as documents or encryption keys, and initiates the AIR-FI covert channel. Data is encoded and transmitted via electromagnetic emissions from DDR SDRAM buses to the Wi-Fi band at 2.4 GHz.

3.4.2 PSDU modification

As proposed in [93], the Camouflaged Subcarriers technique utilizes the differences in subcarrier allocation between the 802.11a/g and 802.11n standards to encode secret data. While 802.11a/g uses 52 subcarriers (48 for data and 4 for pilots), 802.11n increases this to 56 subcarriers (52 for data and 4 for pilots), thus providing four additional subcarriers. These extra subcarriers are used to embed covert data within 802.11a/g signals. By replacing the 802.11a/g LTF with the 802.11n HT-LTF, proper timing synchronization and channel estimation for the covert subcarriers are achieved. This method ensures that the covert transmission appears as valid 802.11n traffic, undetectable in spectral analysis by standard 802.11 receivers, as they cannot determine the exact number of subcarriers used. The STF allows frame detection and content decoding, making the covert data embedding seamless and difficult to detect.

3.4.3 Hidden data in digital modulation

The research outlined in [94] introduces an innovative approach to establishing covert communication within the MIMO-OFDM system in 802.11, where the signal comprises 52 subcarriers, with subcarrier 0 as the frequency center. Typically, subcarriers numbered -1 to -26 and $+1$ to $+26$ are used for data transmission, while subcarriers -27 to -32 and $+27$ to $+32$ remain unused, known as zero padding subcarriers. The authors propose leveraging these zero-padding subcarriers by dynamically altering their positions using

a pseudo-random generator. This method effectively disguises the secret data carriers, making them appear at pseudo-random locations. The sender and receiver share the key for the pseudo-random generator, allowing the receiver to decode the transmitted information accurately despite its disguised placement.

The WiPad covert channel, described in [95], secretly encodes data within the padding of frames at the physical layer. This method involves inserting hidden data into the padding of OFDM symbols used in the 802.11 a/g standards. Padding is usually added to ensure that the frame meets the minimum size requirement or to make the frame size a multiple of a specific value, such as an even number. The standard practice is to set the padding bits to zero. The authors propose manipulating these padding bit values to create a covert communication channel. This technique can embed up to 27 octets in each OFDM symbol, allowing for up to 210 bits per frame for hidden communication.

In the paper [96], the authors introduce a covert channel utilizing the high throughput IEEE 802.11n specification. This covert channel conceals a message within the cyclic prefix of the OFDM symbols. In the 802.11n standard, a Guard Interval (GI) is used to prevent Inter-Symbol Interference (ISI), implemented as a cyclic prefix that avoids overlap by copying the ending part of an OFDM symbol and placing it before the start of the next symbol. This ensures no overlap between symbols, effectively minimizing ISI. However, the receiver's antenna typically ignores the content within the cyclic prefix, presenting a unique opportunity to establish a hidden communication channel. The proposed steganographic method modifies the cyclic prefixes of selected OFDM symbols, embedding secret messages without affecting the regular operation of the wireless network. To synchronize the hidden data transmission and reception, the transmitter and receiver utilize identical Pseudo-Random Number Generators (PRNG) with a shared secret key, ensuring secure communication.

Similar to the approach described in [96], a covert channel can be established by replacing the cyclic prefix in a method outlined in [93]. This method involves embedding hidden data within the cyclic prefix of OFDM symbols in WiFi frames to compensate for ISI and multipath fading. The covert data is encoded within the cyclic prefix and inserted into the OFDM symbols during transmission. This replacement can apply to either half or the entire cyclic prefix, depending on the desired balance between covert channel throughput and detectability. The covert data is encoded into symbols that replace the cyclic prefix, with careful consideration given to minimizing the impact on the overall signal integrity. The receiver then extracts the covert data by isolating the cyclic prefix from the received OFDM symbols and decoding the embedded covert information.

Chew et al. have developed a covert channel utilizing imperfect cancellation [97]. This method exploits the unused Data Carrier (DC) bin of the OFDM signal. The DC bin is located at the center of the frequency spectrum, with a zero frequency offset from the carrier frequency, typically carrying no actual data. It serves as a reference point for demodulation and synchronization. The proposed technique conceals a covert message within this bin while ensuring that the transmitted signal complies with the spectral mask specified by the 802.11 standard. The hidden signal can be extracted at the receiver end by canceling the unwanted OFDM signal components.

Dutta et al. have introduced a covert channel using PSK modulation, as detailed in [98]. This covert channel transmits hidden messages disguised as symbols that imitate the distorted symbols in the constellation diagram. These distortions typically result from hardware imperfections and the inherent additional noise in wireless transmissions. These covert symbols appear indistinguishable from regular distorted symbols caused by noise

to an observer. The implementation strategy involves encoding data in BPSK or QPSK at a low rate. A symbol is chosen, and noisy symbols are strategically placed around it, with each noisy symbol phase-shifted from the others. Standard receivers decode the message at the symbol rate indicated in the frame, treating the dirty symbol as noise. However, for covert receivers, a pre-configured function filters out carriers containing covert symbols.

In [99] Grzesiak et al. present a covert channel based on a dirty constellation with Phase drift. In wireless communication, transmitted symbols exhibit dirty constellations due to their passage through the radio channel and imperfections caused by both transmitting and receiving devices, resulting in phase and amplitude distortions. This effect manifests as phase and amplitude distortions in received signal constellations. The covert information is encoded within phase changes of selected signal harmonics. The m^{th} harmonic chosen for steganography is expressed according to [100], as presented in Equation 3.2:

$$y_k(t) = A_m \exp(j(2\pi f_m t + \phi_x + \Delta X_m)) \tag{3.2}$$

Here, A_m and f_m represent amplitude and frequency, respectively, ϕ_x denotes a random initial phase that remains constant during the transmission of covert symbols, and ΔX_m represents the phase drift step for the proposed dirty constellation. Compared to [98], this proposed mechanism introduces a phase drift in the noisy symbols representing the covert information.

Cao et al. introduced a new Wireless Covert Channel with Constellation Shaping Modulation (WCC-CSM) in [101]. By adding constellation-shaping modulation, the WCC-CSM improves the system's efficiency presented in [98]. In the system described in [98], there is a chance that the covert constellation is recognizable within all subcarriers transmitting the covert signal, even when only a subset of the subcarriers are indeed transmitting covert signals. This recognition happens only after mapping at the receiver's end. With WCC-CSM, the process proceeds as follows: First, the original message bits, denoted as m_c, are modulated using QPSK to produce the output cover signal s_c. Then, the secret message m_s undergoes modulation with constellation error, resulting in artificial noise s_s. Both s_c and s_s are combined to form the covert signal (s_{ct}). The sender selects symbols to add artificial noise, ensuring covert symbols only appear in selected subcarriers. The signal is transmitted over the wireless channel after additional processes, such as Inverse Fast Fourier Transform (IFFT) and adding a cyclic prefix.

The authors in [102] utilized Pseudo-Noise Amplitude Shift Keying (PN-ASK) to hide data in plain sight. In PN-ASK, concealed symbols are positioned at specific locations within an M-ary Phase Shift Keying (MPSK) constellation diagram. The distance of each hidden symbol from the symbol representing explicit information is depicted by its radius length. The principle is that all symbols containing explicit information are displayed in an M-PSK constellation diagram at a fixed distance from the center. Conversely, symbols containing the hidden data are exhibited in the constellation diagram with amplitude variation to the explicit data symbol. In mathematical terms, the hidden mapping assigns a specific amplitude variation, d, to each hidden symbol, thereby altering its position within the constellation diagram. For a given element indexed as i, the hidden symbol is determined by the Equation 3.3:

$$k(i) = 1 - (i - 1) \cdot d \tag{3.3}$$

Where d is the amplitude variation of the hidden symbol.

Grzesiak et al. introduced a Covert Channel Based on Quasi-Orthogonal Coding to allow a low-energy covert signal to avoid detection in the presence of Spectral Correlation-based Interference (SCI) [103]. They proposed that a covert channel consists of two signals – the covert signal and the cover signal, where the latter acts as an envelope for the former. The covert signal must have significantly lower power than the cover signal to minimize the probability of detection. However, channel estimation errors and the impact of Successive Interference Cancellation (SIC) can lead to a high likelihood of detecting a low-power covert signal after SIC processing. To overcome this challenge, the authors put forward a new covert channel in which the characteristics of the covert signal are retained but transmitted in a quasi-orthogonal manner. The Equation 3.4 expresses the quasi-orthogonality:

$$\left| \int_{-\infty}^{\infty} x_1(\rho) \cdot x_2^*(\rho - t)\, d\rho \right| < \varepsilon \tag{3.4}$$

Where $\varepsilon << 1$ for any t, and x_1, x_2, are the cover and covert signal normalized to a unit of energy. The cover signal is extracted at the receiver using Frequency Shift Keying (FSK) modulation. This extraction process entails a meticulous comparison of the IQ samples of the covert signal against the cover signal over time.

Several covert communication channels are proposed in the paper [93]. One of these channels utilizes the Short Training Field with Phase Shift Keying (STF-PSK). In this method, symbols are embedded in the STF and remain unchanged post-detection, forming the basis of the covert communication channel. Within the STF, symbols are modulated using Binary Phase Shift Keying (BPSK), with each symbol spaced 45° apart. Additional phase shifts ($\Delta\phi$) are introduced to each STF symbol without compromising the PHY functionality, encoding the covert message within these phase shifts and corresponding the number of shifts to the number of bits to be encoded. This approach offers flexibility by allowing one phase shift to represent a group of bits through an agreement between the covert transmitter and receiver. Moreover, introducing multiple phase shifts enhances the channel's bandwidth.

Another method proposed in [93] is Carrier Frequency Offset (CFO) Frequency Shift Keying (CFO FSK). In 802.11 systems, obstacles during signal propagation can lead to reduced signal strength at the receiving end. 802.11 transmitters use amplifiers to send signals at the frequency f_{tx} to address this. However, the message signal is received at a slightly different frequency (f_{rx}), known as the Carrier Frequency Offset (CFO). A CFO generator is introduced at the receiver to counteract the Doppler effect and establish a covert channel. The covert transmitter assigns a group of bits to two frequencies, ± CFO, each with a symbol duration of 4 microseconds. These frequencies are multiplied to generate f_{tx}, shifting symbols by ± CFO in the frequency domain. Upon reception, the receiver estimates the phase shift of the message symbols, separates the CFO signal from the message signal using a low-pass filter, and extracts the encoded bits using a hard decision decoder.

3.4.4 MIMO-based encoding

In a paper [104], the authors introduce a wireless covert channel for MIMO systems. This covert channel exploits a noise signal in the wireless channel. The covert message, a copy of the noise signal, is combined with the explicit message and transmitted through

the wireless channel. The MIMO system allows the covert receiver to transmit multiple messages through the wireless channel simultaneously.

CovertMIMO [105] leverages the MIMO beamforming technique used in WiFi networks to create a covert transmission scheme. The central idea is to introduce slight modifications to the normal uplink transmission process, allowing covert information to be encoded. The covert communication process is as follows: 1) Control channel operations - the client (covert transmitter) modifies the CSI matrix during the feedback process. The modified CSI matrix, denoted as $H_a = H \cdot \Phi$, where Φ is an invertible matrix, appears normal to the AP but introduces a controlled deviation that encodes covert information. 2) Data channel operations - during data transmission, the client sends a modified signal $X_a = \Psi \cdot X_0$ instead of the normal signal X_0. Here, Ψ is another invertible matrix that slightly alters the transmitted signal to encode covert data. 3) Signal recovery - at the AP, the received signal Y_a is processed using the modified CSI H_a to recover the transmitted data. The deviation introduced by Φ and Ψ allows the covert information to be extracted while the overt message remains unchanged.

Li et al. proposed a novel system for covert data transmission utilizing antenna coding[106], designed for systems equipped with multiple R_X and T_X antennas. at the center of the system is the concept of a codebook, which must be present at both the sender and receiver end. This codebook maps covert bits to antenna code blocks, where each set of bits corresponds to a specific antenna scheme. For instance, consider a scenario where N_T represents the number of transmit antennas and N_S denotes the number of selected antennas designated for transmitting secret data. In this setup, the sender possesses a set of bits 0101. These bits are mapped to the corresponding scheme in the antenna code block, generating an antenna array. In this array, elements marked as 0 signify antennas transmitting public information, while elements marked as 1 represent antennas transmitting covert data. Following this mapping process, the covert data is modulated using Quadrature Amplitude Modulation (4-QAM), generating four constellation points that represent public information and four additional points that represent secret data. The receiver, equipped with algorithms outlined by the authors, can detect these secret data points, enabling covert communication channels.

3.5 802.11 PHY timing covert channels

3.5.1 Periodic interference encoding

Shah et al. have introduced a new covert channel facilitated by external interference [107]. This covert channel intermittently utilizes an external wireless jammer to disrupt the wireless medium, resulting in intervals of network activity during these disruptions. Access to the wireless network is optional for this method to function. The main concept involves establishing a covert channel through external interference. This is achieved by utilizing a jamming signal within the same frequency band as the 802.11 signal frequency, with the jammer strategically positioned between the sender and receiver to induce interference and disrupt frames during transmission. The wireless jammer transmits covert messages in this setup while the receiver is an eavesdropper. Both parties use a shared encoding and decoding scheme facilitated by a real-time clock, eliminating the need for synchronization. In this context, a jam cycle combines a wait time (a constant interval) and the interference duration. Encoding a single bit involves multiple jam cycles, which are achieved by adjusting the duration of the interference time. For example, when the

wireless jammer wants to transmit a bit zero, it can use a wait time of 0.5 seconds across three jam cycles, with a $\{0.1, 0.2, 0.3\}$ pattern. Thus, the wireless jammer communicates the hidden message through patterns such as:

- Interfering with the transmission for 0.1s followed by a 5s wait.
- Interfering with the transmission for 0.2s followed by a 5s wait.
- Interfering with the transmission for 0.3s followed by a 5s wait.

3.5.2 Connection order encoding

Two covert channels are described in the paper [108]. Both covert channels use the same concept: the covert sender and receiver select specific WiFi clients (C_1, C_2, ..., C_n) for covert communication. They establish an encoding system where each client C_i is linked to a secret symbol, S_i. To send the symbol S_i, the covert sender prompts client C_i to reconnect to the network. The covert receiver monitors the network for these reconnections. When it detects a reconnection from client C_i, it interprets it as representing the symbol S_i. The first method involves a deauthentication attack, where the sender forces a specific client to deauthenticate and then reconnect to the AP. The receiver monitors the station to identify the reconnection and decodes the secret message based on the symbol assigned to the reconnected station. The second covert channel method involves the sender directly controlling clients through a wired connection. The sender uses a command wire to trigger client reconnections and a status wire to monitor their connection status. The status wire continuously reports whether the client is connected (high voltage) or disconnected (low voltage), while the command wire, generally at low voltage, sends a high-voltage pulse to initiate reconnection.

3.6 Analysis of covert channel techniques

3.6.1 Temporal and structural distribution of covert channels

An analysis of the collected data provides insight into the current state of 802.11 covert channels. As shown in Figure 3.2, these covert channels first appeared in 2003. After a five-year period of inactivity, a notable increase occurred in 2008, with three publications indicating a growing interest among researchers. This trend continued with fluctuations in the following years, although the numbers remained relatively modest. Between 2010 and 2017, there was a significant increase in research on covert channels. The researchers explored various aspects of covert communication, including protocols, detection methods, and countermeasures. In 2018, Wi-Fi covert channels reached their peak, with the highest number of publications to date. From 2019 to 2025, at least one article has been consistently published each year. This suggests that the topic is gaining attention in the network research community as a primary method for concealing data in Wi-Fi networks.

When looking at the distribution of covert channels as depicted in Figure 3.3, we found that those operating at the MAC layer make up a more significant portion, accounting for 57% of the total. PHY covert channels represent a slightly lower percentage at 43%. MAC covert channels offer a more straightforward method for transmitting data secretly. They utilize features of network protocols, such as the 802.11 DCF and frame structure. The random backoff mechanisms and fields within the 802.11 frame structure

facilitate easier implementation. MAC covert channels are especially straightforward to implement using network simulation tools for 802.11 networks. On the other hand, although PHY has various features and modulation schemes allowing for a broader range of covert communication methods, we found PHY covert channels to be more complex and pose significant challenges. They require a proper understanding of signal processing, modulation schemes, and the noisy communication channel.

In covert channel classification, storage-based channels are the most common, making up 71% of the total distribution, as shown in Figure 3.4. Storage channels are used more frequently because certain aspects of the specification are left open. These aspects enable flexibility in using random, custom, or mandatory values without compromising the network's functionality. These values can include frame number, segment number, supported rates, and other relevant details. This flexibility provides a wide range of options for implementing storage channels. However, timing-based covert channels, although representing only 25% of the pool, are still an important area of research. These channels rely on timing variations in transmissions, such as altering the transmission frequency of beacon and probe request frames, to transmit information secretly between connected devices. Although they make up a smaller percentage, timing channels present challenges in detection and prevention because they depend on subtle timing patterns. Furthermore, hybrid covert channels, which combine aspects of both storage and timing-based channels, represent a small share of 4% in our collected data, highlighting the growth of innovative approaches being explored by the research community.

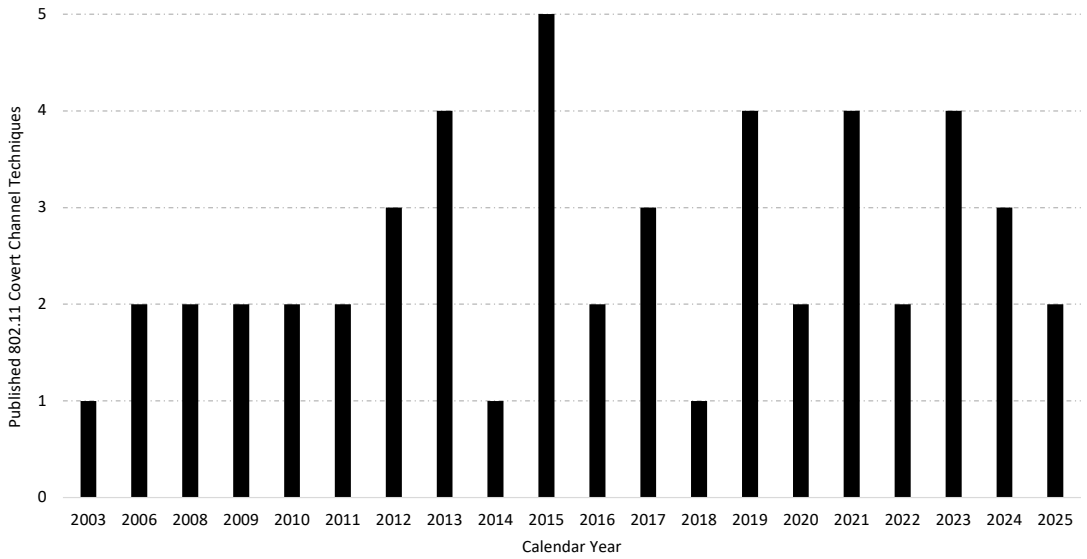


Figure 3.2: 802.11 covert channels related publication over the years

3.6.2 Implementation techniques and tools

Creating covert channels often involves the use of specialized hardware, enabling researchers and practitioners to implement and test their theories in real-world scenarios. The hardware tool includes the Wireless Universal Serial Bus (USB) WiFi Adapter, which is commercially available [109], [110], and supported by accessible drivers that are typically available as open-source projects in public code repositories [111], [112]. These

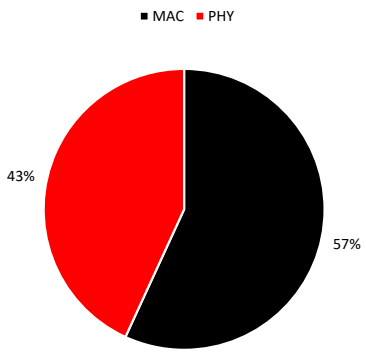


Figure 3.3: Distribution of MAC and PHY covert channels

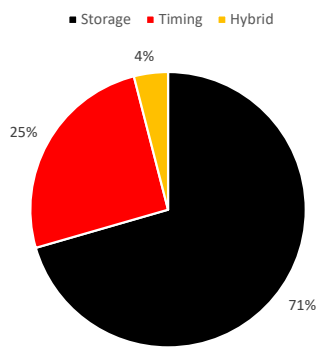


Figure 3.4: Distribution of storage, timing, and hybrid covert channels

drivers can be easily modified and integrated into the Linux kernel as modules, facilitating customization and experimentation.

We notice the use of SDR ADALM-PLUTO [113], a portable Software-Defined Radio (SDR) ideal for experimentation with Radio Frequency (RF) signals. The Universal Software Radio Peripheral (USRP), a high-performance SDR, is employed for transmitting and receiving wireless signals. The Virtex-V Field-Programmable Gate Array (FPGA) is utilized for custom hardware functions, including signal processing. Additionally, the Xilinx Zynq Board, equipped with the OpenWIFI driver, integrates a programmable logic fabric and an ARM-based processor, making it suitable for complex wireless communication projects. Both Virtex and Zynq are product lines developed initially by Xilinx, and then acquired by Advanced Micro Devices, Inc., known as AMD, in 2022. We also observed that the Raspberry Pi [114], a versatile single-board computer, is frequently used for prototyping and small-scale projects.

In addition to hardware, various software tools are essential for simulating and analyzing covert channels. Network simulators such as the NS-3 Network Simulator [115], an open-source, discrete-event network simulator designed primarily for research and educational use, and the Optimized Network Engineering Tool (OPNET) Simulator [116], a comprehensive network simulation software suite used to plan, design, and manage networks. Programming frameworks like MATLAB and Simulink are utilized for signal processing, modeling, and simulation, whereas LabVIEW, a visual programming environment, is employed for hardware interfacing and signal processing tasks. Programming languages such as C++ are used to develop custom frameworks and algorithms. With its readability and extensive library support, Python frameworks are often employed for network packet manipulation and covert channel implementation. We also observed one use case of the Microsoft Windows Native Wi-Fi API, which allows the manipulation of Wi-Fi management frames on Windows platforms. Software implementations are easy and fast to set up and reproduce, offering greater scalability in terms of resources and customization. The hardware allows us to observe the effects of simulated conditions in real-life scenarios, such as noise, signal degradation, and other factors that affect signal quality.

Furthermore, theoretical propositions for covert channels often involve mathematical or probabilistic models, such as Markov models, that describe their behavior and performance, for example [56], [67], [81]. Some ideas were theoretically proposed and later implemented using hardware or software tools, for example [57].

Figure 3.5 illustrates the number of experiments, simulations, and theoretical proposals conducted. The analysis of the collected dataset shows a balanced distribution among theoretical models, simulations, and experimental implementations. While most covert channels are implemented using network simulators, a significant portion is also implemented in hardware devices. This approach tests the feasibility of solutions in real-world environments, demonstrating how these channels can be executed with readily available hardware and software. On the other hand, network simulation is the most common choice, as simulations provide practical insights into the behavior of covert channels under various network conditions. This is achieved by utilizing custom-built frameworks and the availability of established simulators. Theoretical models, in contrast, offer a foundational understanding of covert channel mechanisms and their potential impacts. Notably, at least one article in the studies did not specify the implementation method for the covert channel [101].

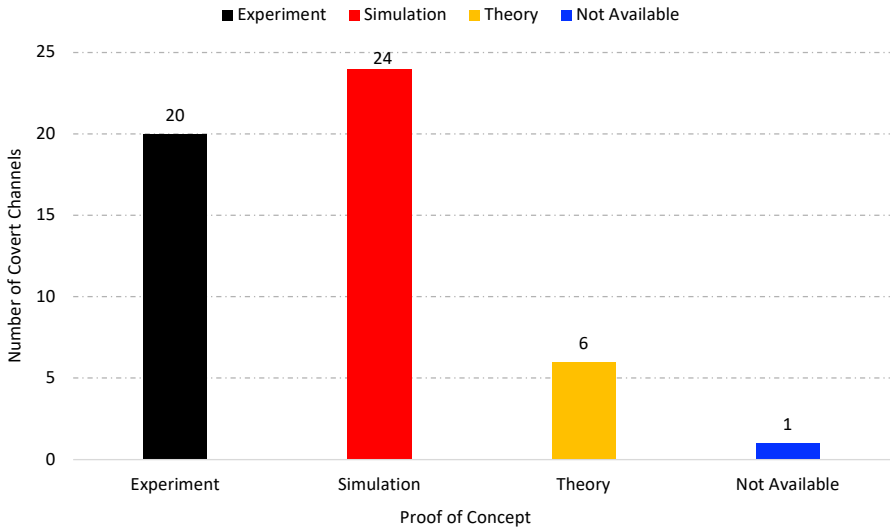


Figure 3.5: Overview of tools used in the covert channel implementation process

3.6.3 Bandwidth and performance considerations

The data analysis revealed diverse data transfer rates or available bandwidth achieved by various covert channels, ranging from very low rates, such as 0.78 bps [72], to high rates, like 640 Mbps [67]. This wide range highlights the versatility of covert channels, which can be tailored for different levels of stealth and data throughput. Many covert channels operate at relatively low bandwidths to maintain stealth and minimize detection, with typical values in the order of bps. These lower rates suggest that covert channels prioritize blending in with normal network traffic over high-speed data transfer. On the other hand, some channels achieve significantly higher data rates, indicating their use in scenarios where higher data rates are essential and the risk of detection is mitigated by other means. Patterns in bandwidth usage reveal that higher bandwidth channels typically involve sophisticated hardware, such as USRP or FPGA devices. In contrast, lower bandwidth channels may employ simpler methods, such as manipulating packet timings or embedding data in less obvious fields of standard network protocols. Our analysis highlights the trade-offs between stealth, transparency, and performance in covert

channel design, underscoring the importance of careful consideration before implementing these channels.

3.7 Countermeasures against covert channels

In their work [117], Zhao et al. introduce a method for detecting covert channels in 802.11 WLANs that utilize covert timing channels to transmit secret data by leveraging different data rates. Their proposed Statistical Hypothesis Testing Method (SHTM) aims to discern covert channels from standard data rate patterns. The available data rates in a mixed wireless environment range from $\{1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54\}$ Mbps, and covert channels exploit specific data rates to encode secret messages, resulting in a unique subset of these rates. The objective is to differentiate between covert channel patterns and normal data rate distributions through hypothesis testing. This involves selecting samples to analyze the characteristics of the data rate distribution within the network. The method characterizes the probability distribution of transmission data rates to capture rate diversity in WLANs based on measurements taken over a specific timeframe. By examining these distributions, the authors aim to identify potential covert timing channels or unstable wireless channel conditions, achieving a 100% detection rate with no false positives in their simulations.

The article [118] introduces a novel approach to detecting timing covert channels using a probability model. This method is based on the observation that covert timing channels exhibit different distributions of inter-packet delays compared to legitimate channels. The approach involves sampling 10% of inter-arrival delays and calculating the variance between these samples. Assuming that covert channels use two distinct inter-arrival times to transmit confidential information (t_0 for bit 0 and t_1 for bit 1), the authors calculate the probability distribution function and variance of these delays. Their analysis shows that the variance of inter-packet delays in covert channels is significantly higher than that of legitimate channels, specifically twice as large. This significant difference in variance allows for the easy differentiation between covert and legitimate channels. By leveraging this variance discrepancy, the probability-model-based approach effectively identifies covert timing channels, enhancing detection accuracy and robustness in diverse network traffic conditions.

AL-Khulaidi et al. conducted a comprehensive survey on covert channel detection in their work referenced in [119]. Their study provides a systematic taxonomy for categorizing covert channel generation and detection methods, distinguishing between classic (rely heavily on statistical analysis, signature-based detection, and protocol behavior examination) and modern approaches (leverage the power of machine learning, deep learning, and advanced data mining techniques to provide more adaptive and accurate detection capabilities). Each category is meticulously outlined, accompanied by a detailed summary presented in a tabular format. Among the classic methods discussed, the authors prominently featured [120] and [121], which serve as effective countermeasures specifically designed to detect timing-based covert channels in Wi-Fi networks.

Elsadig et al. conducted an extensive survey on existing techniques for detecting network covert channels [122]. The authors present methods to mitigate (reducing the covert channel's capacity without affecting system usability) or eliminate (rendering it impossible to use) specific covert channels. Building upon their survey findings, the authors proposed the Network Covert Channel Triangle (NCCT), which encompasses

three elements. The first is the Rapid Development of Network Technology, which refers to keeping aware of recent developments in network fields such as IoT and cloud computing. Switching Techniques involve altering the covert message carrier and switching between protocols or fields to introduce variation and enhance the security of covert messages. Micro-protocol refers to embedding protocol features into the covert channel to bolster security.

Chourib et al. in [123] present a comprehensive investigation into covert networks and their detection using Machine Learning (ML) algorithms. Specifically, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), and Deep Neural Networks (DNN) were employed for this purpose. The study highlights two primary shortcomings of existing network surveillance systems: The reliance on manual rule insertion for identifying network traffic ambiguities and the limitation of detecting only known threats. A Network Anomaly Detection System (NADS) was proposed to address these limitations. NADS identifies abnormal network traffic patterns that deviate from expected normal behavior. It can adopt various approaches, including statistical, classification, clustering, or information theory-based methods. The proposed approach comprises three key steps: 1) generating datasets containing network traffic, 2) training and extracting features from the data, and 3) testing the models using diverse tools. A dataset was created from nine standard covert channel tools, and the covert channels were classified into patterns. The k-NN model demonstrated the highest precision rate at 98% detection of a given covert channel with a low false positive rate of 1%.

Caviglione in [124], presents a comprehensive examination of countermeasures against network covert channels. Rather than introducing a novel mechanism or conducting a survey, the article offers an in-depth review of strategies to combat hidden communications within networks. The author discusses the essential steps required to bolster security in the presence of covert channels, encompassing detection, limitation, and elimination measures. Special attention is paid to methodologies aimed at preventing covert communications. Key considerations highlighted include the necessity for a deep understanding of covert channel functionality, awareness of irregularities indicative of their presence, and the ability to localize and identify the carrier of secret data. The challenges associated with detecting timing, storage, Internet Protocol version 6 (IPv6), Voice over IP (VoIP), Domain Name System (DNS), and Hypertext Transfer Protocol (HTTP) covert channels are also addressed. Moreover, the author emphasizes the importance of eliminating covert channels in specific protocols, such as HTTP and IPv6, and the significance of preventive measures that entail considering covert channels from the early design stages.

The literature review indicated that, in general, the implementation and detection of Transmission Control Protocol/Internet Protocol (TCP/IP) covert channels have received more attention than 802.11. This disparity highlights the need for more focused research on wireless protocols. To enhance network security, further research is needed to develop targeted countermeasures that effectively mitigate covert channel threats in wireless networks. During the research, only two countermeasures designed for Wi-Fi timing covert channels with excellent accuracy metrics were proposed. This development marks a significant step towards creating more dedicated countermeasures for 802.11 networks.

4 Covert channel StegoRates

4.1 StegoRates operation

StegoRates leverages the flexibility of the *Supported Rates* field by manipulating the MSB to indicate whether a specific data rate is mandatory or optional (see Section 2.2.4 for a discussion about supported rates). In this approach, the STA encodes its secret message by setting the MSB of each data rate value to 0 or 1, depending on the bit it wants to transmit, ensuring a *bandwidth of at least 8 bits per frame*. Figure 4.1 illustrates an example in which up to 12 bits are encoded using the supported rates and the extended supported rates field. The sender announces the set of data rates it supports and uses the MSB of each rate field to distinguish between basic (mandatory) rates and optional rates. A data rate with the MSB set to 1 is marked as a basic rate with (B), whereas a rate with the MSB set to 0 is treated as optional, denoted without a (B). To construct the covert message, the MSB of each listed supported rate is extracted. For regular observers, the probe request is sent by a station that supports basic rates (such as 1, 2, 5.5, and 11 Mbps), while all other rates, including those in the extended supported rates element, are optional. However, within the covert channel scenario, the MSB values are deliberately altered to encode a secret message 111100000000.

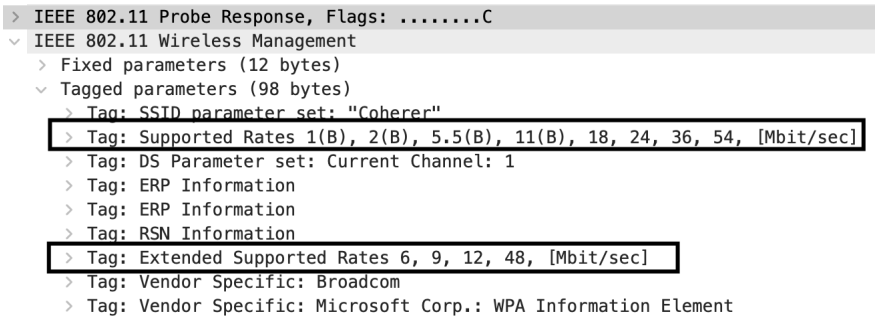


Figure 4.1: Example of encoding a covert message 111100000000 through StegoRates

To provide a clearer understanding of how the covert channel operates, we present the pseudocode for the encoding and decoding processes in Algorithms 1 and 2, respectively. To initiate and terminate covert communication, both the STA and the AP agree on two distinct 8-bit secrets: one to signal the beginning of the covert channel and another to indicate its termination. Furthermore, they share a CRC polynomial for encoding. The sender uses this agreed-upon polynomial to compute the CRC for the 8-bit start or termination secret. The resulting 8-bit encoded value is then mapped onto the supported rates field of a probe request. Each bit is inserted into the MSB of the corresponding data rate value in the supported rates field.

Once the STA sends a probe request with the initial secret and receives a corresponding probe response indicating that the AP is ready for covert communication, the STA transmits the first covert message. After each covert transmission, two periodic events are scheduled: one to trigger the subsequent message transmission and the other to verify the receipt of a probe response. If no response is detected, the STA immediately retransmits the probe request. Since the STA does not perform network association, it can ignore the incoming probe responses and use them as acknowledgments.

To close the covert channel, the STA repeats the initiation procedure, but this time using the secret that indicates termination of the covert communication, and the AP decodes the message accordingly. In general, upon receiving a probe request, the AP extracts the MSB from each data rate value to form an 8-bit sequence, computes the CRC, and uses the result to determine whether the secret indicates the opening, transmission of data, or termination of the covert channel.

Algorithm 1 StegoRates pseudocode for the transmitter

```
Input: secret — 8-bit value for signaling or covert data
Input: crc — shared CRC polynomial for encoding covert data
procedure TransmitCovertData(secret, crc)
    if secret = startsecret then
        channelActive ← true
        rates ← EncodeSupportedRates(secret, crc)
        SendProbeRequest(rates)
        t ← now + interval
        Schedule(TransmitCovertData, secret, crc, t)
        ackReceived ← false
        t_ack ← now + acktimeout
        Schedule(CheckAck, secret, crc, t_ack)
    else if secret = endsecret then
        channelActive ← false
    end if
    if channelActive = true and secret ≠ endsecret then
        rates ← EncodeSupportedRates(secret, crc)
        SendProbeRequest(rates)
        t ← now + interval
        Schedule(TransmitCovertData, secret, crc, t)
        ackReceived ← false
        t_ack ← now + acktimeout
        Schedule(CheckAck, secret, crc, t_ack)
    end if
end procedure
procedure CheckAck(secret, crc)
    if ackReceived = false then
        TransmitCovertData(secret, crc)
    end if
end procedure
```

4.2 StegoRates properties and deployment scenarios

When it comes to resisting steganalysis, the covert channel utilizes the flexibility of the IEEE 802.11 standard, which does not impose strict rules regarding mandatory and optional data rates. With the introduction of new amendments that allow for higher data rates while still supporting lower rates for backward compatibility, the covert channel's use of the supported rates field appears harmless. It embeds its secret message in the

Algorithm 2 StegoRates pseudocode for the receiver

```
Input: rates — the set of data rate in the probe request
Variable: isCovertActive ← false
procedure ProcessProbeRequest(rates)
    secret ← ExtractMSBsecret(rates)
    decoded ← DecodeCRC(secret, crc)
    if decoded = startsecret then
        isCovertActive ← true
    else if decoded = endsecret then
        isCovertActive ← false
    else if isCovertActive = true then
        StoreCovertDatadecoded
    end if
end procedure
```

MSB of each data rate, a modification that is difficult to detect due to the lack of a defined pattern. The covert channel may only raise suspicion if there is an abnormal increase in the frequency of probe requests with different supported rates coming from a single source, as correlation techniques can identify a device that repeatedly sends probe requests without ever establishing a connection. However, an additional layer of security is provided by a CRC code. Even if an observer suspects that covert communication is taking place, it remains extremely challenging to decode the hidden message without knowing the specific encoding scheme and the CRC seed (polynomial generator). A key feature of StegoRates is its variable covert bandwidth, enabled by the extended supported rates field, which can be as long as 255 bytes. This flexibility enables the sender to adjust the number of covert bits according to the desired covert data rate.

In terms of transparency, the covert channel is designed to operate with minimal impact on normal network functionality by transmitting probe requests at controlled intervals. However, if probe requests and responses are generated too frequently, this may slightly increase overhead by consuming additional bandwidth on the wireless medium. Furthermore, any mismatch between the station's rates and those expected by the AP does not affect the overall functionality of the network. This is because the covert STA intentionally avoids associating with the AP, which reduces the risk of exposure or targeted attacks while transmitting covert data.

The practical use of covert channels is crucial in untrusted environments, where verifying the true identity of the network administrator poses a challenge [125]. It can be difficult to identify who is managing a specific Wi-Fi network, which helps prevent attacks such as rogue APs [126], [127], deauthentication attacks [128], and traffic analysis [129]. Despite this uncertainty, two parties may still wish to exchange messages securely without exposing the data through regular data transmissions.

4.3 Simulation scenarios and metrics

4.3.1 Environment and scenarios

The simulation of the wireless network was carried out using NS-3 [115]. NS-3 is an open-source simulator that supports various network models implemented in both C++ and

Python. It provides the necessary APIs to create a STA and an AP, along with the WLAN protocol stack. To meet the requirements of the covert channel, we made modifications to the simulator's source code. We utilized the API provided by the simulator. Specifically, we used `AllSupportedRates::AddSupportedRate(uint64_t bs)` to append each data rate. The initial, termination, and covert messages were then inserted into the MSB using `AllSupportedRates::SetBasicRate(uint64_t bs)`. The simulation parameters are presented in Table 4.1. The simulation was carried out under two different scenarios. In the first scenario, the covert messages are transmitted without retransmission in the event of a probe response timeout. In the second scenario, the retransmission mechanism was deployed to enhance the throughput of the covert channel.

Simulations were repeated multiple times to ensure reliability, and the average values for each metric were computed. In all figures, the error margin for each simulation point, within a 95% confidence interval, did not exceed ±5%.

Table 4.1: StegoRates simulation parameters

Parameter	**Value and Unit**
802.11 standard	802.11ac
Channel width	20 [MHz]
Wi-Fi channel model	Yet Another Network Simulator
TX and Rx antennas per node	1
Active probing	true
Beacon interval	100 [ms]
Probe request interval	10 [ms]
Transport protocol	UDP
Frame size	1000 [bytes]

4.3.2 Metrics

Establishing metrics for the covert communication channel is crucial for assessing its performance under various conditions and identifying areas for optimization. We have selected throughput, frame efficiency, delay, and jitter as key performance indicators.

- *Throughput*: The rate of successfully transmitted probe requests, denoted as N_{PR} each carrying B secret bits during the simulation time, measured in bits per second (bps), as shown in Equation 4.1:

$$\text{Throughput} = \frac{N_{\text{PRequests}} \times B}{T_{\text{simulation}}} \quad [\text{bps}] \tag{4.1}$$

- *Transmission Efficiency*: This metric expresses the reliability of the covert channel. It is defined as the percentage of probe requests that result in a Probe Response, representing successful covert exchange, as shown in the Equation 4.2:

$$\text{Transmission efficiency} = \frac{N_{\text{PResponses}}}{N_{\text{PRequests}}} \times 100 \quad [\%] \tag{4.2}$$

- *Delay*: This parameter refers to the average round-trip time (RTT) between the transmission of a probe request and the reception of the corresponding Probe Response. It reflects how long the covert sender waits for an acknowledgment as per Equation 4.3:

$$\text{Delay} = \frac{1}{T_{\text{simulation}}} \cdot \left(\frac{1}{N} \sum_{i=1}^{N} \left(t_{\text{PResponse}_i} - t_{\text{PRequest}_i} \right) \right) \quad [\text{ms}] \tag{4.3}$$

- *Jitter*: Represents the variation in delay across consecutive probe requests transmissions. It provides a measure of the temporal stability of the channel and is calculated as per Equation 4.4:

$$\text{Jitter} = \frac{1}{T_{\text{simulation}}} \cdot \left(\frac{1}{N-1} \sum_{i=2}^{N} \left| \text{Delay}_i - \text{Delay}_{i-1} \right| \right) \quad [\text{ms}] \tag{4.4}$$

4.4 Performance evaluation

4.4.1 Periodic transmission without retransmission

The first scenario involves the covert station sending periodic probe requests without retransmissions. In this case, there is no timeout for probe responses; the sender simply triggers periodic probe requests. Additionally, a counter is implemented to track the number of probe requests received by the AP for metrics purposes.

As illustrated in Figure 4.2, the covert channel reaches its peak throughput of 800 bps in scenarios without external interference, utilizing a transmission interval of 10 ms. By examining the relationship between throughput and transmission interval, several observations can be made. The first notable observation is that as additional STAs join the network and begin generating traffic at a slower rate, the throughput decreases for the same transmission interval. This reduction occurs because the covert STA must contend with other STAs to access the wireless medium. Additionally, the AP's resources are divided among multiple STAs, further impacting performance. The second observation indicates that throughput in all experiments is inversely proportional to the transmission interval. As the transmission interval increases, throughput decreases. Even in isolation, we observe a significant drop in throughput during the first 10 to 20 seconds (half of the initial value). After this initial drop, the decline becomes less pronounced, suggesting that beyond a certain point (e.g., 50 ms), further increases in the transmission interval have a diminishing effect on the covert channel, resulting in a slower decline in throughput.

When examining the transmission efficiency, as demonstrated in Figure 4.3, the covert channel achieves optimal performance in the absence of external interference, with no frame loss and a frame efficiency of 100%. However, introducing 10 additional stations results in a 40% decrease from the initial efficiency value. As more stations are added, the efficiency continues to decline, but the drop becomes less evident compared to the reduction in throughput. In particular, when there are 20 stations, adding more stations causes the frame efficiency to decrease just minimally. This suggests that, at this point, enough interference is generated (approaching saturation) so that adding more stations does not significantly impact efficiency. It is important to note that frame efficiency is directly influenced by the number of external stations rather than the transmission interval.

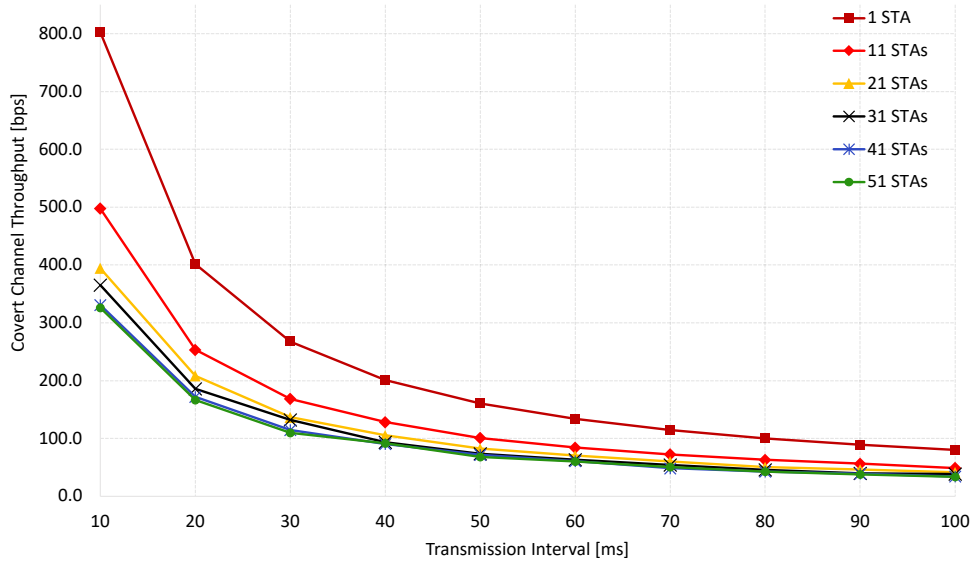


Figure 4.2: Impact of transmission interval and the number of stations on covert channel throughput

The number of frames successfully reaching the destination decreases due to increased contention, which is caused by collisions and a larger queue at the receiver that must process and respond to incoming frames, leading to a decline in network performance.

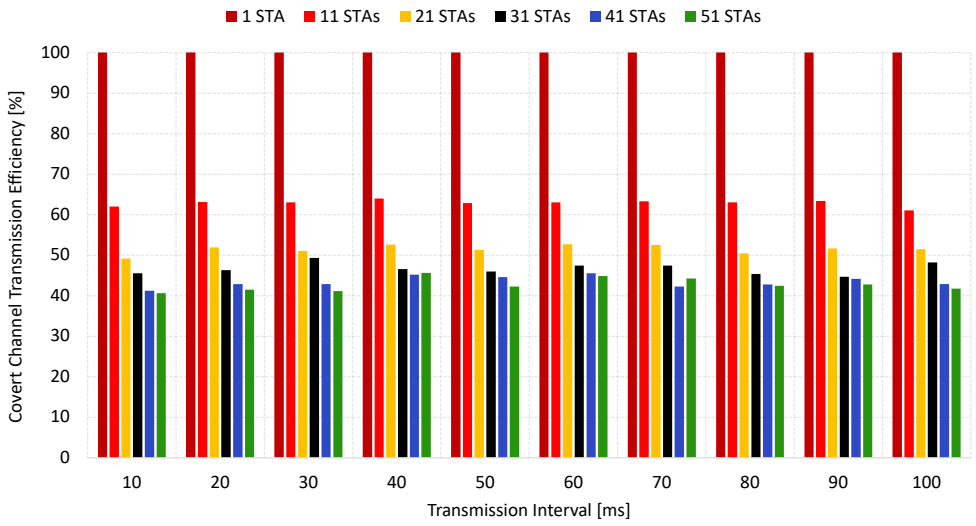


Figure 4.3: Impact of transmission interval and the number of stations on covert channel efficiency

Figure 4.4 illustrates the impact of the transmission interval and the number of external stations on the delay in the covert channel. The findings indicate that the transmission interval has a direct impact on the response time, while the number of contending stations has a lesser effect. Notably, in isolation and with ten additional stations, the delay remains almost constant. This suggests that the presence of external traffic does not significantly affect the covert channel, regardless of the transmission interval. However, the delay begins to respond to the number of contending stations starting at about 20 ms;

it increases but then remains constant as the number of additional stations rises from 20 to 50. This increase in delay can be attributed to higher contention and longer idle times between transmissions. Although the actual transmission and acknowledgment processes may be quick, the overall elapsed time between successive messages contributes to the increase in the average delay metric.

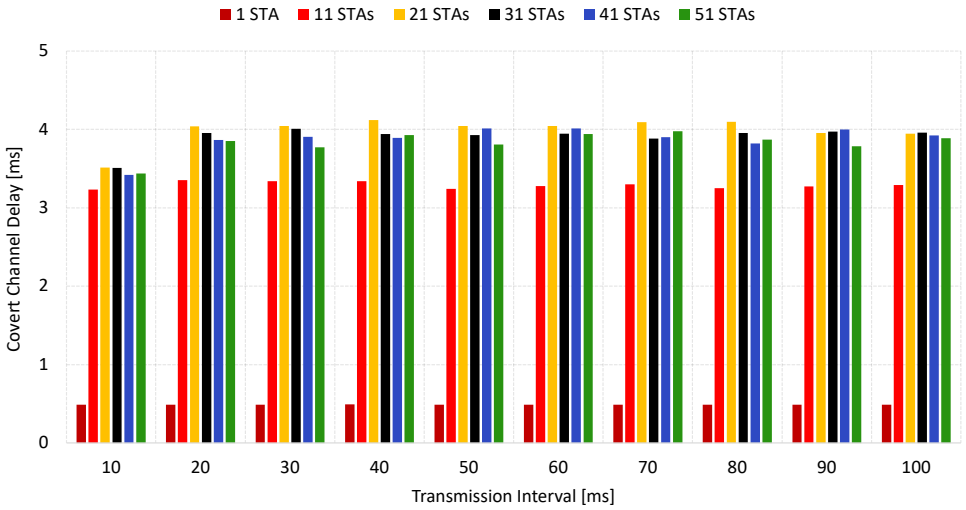


Figure 4.4: Impact of transmission interval and the number of stations on covert channel delay

Figure 4.5 illustrates how transmission intervals and the number of external stations affect the jitter of covert channels. As expected, the jitter pattern closely follows the delay behavior shown in Figure 4.4. In scenarios with a single station, the jitter is at its lowest. However, as more stations are introduced, there is a noticeable increase in jitter. This increase highlights the immediate impact of medium contention and queuing delays due to external traffic. Additionally, once the transmission interval reaches approximately 20 ms, the jitter stabilizes across all scenarios. This consistency supports the conclusions from the delay analysis: longer transmission intervals and a higher number of contending stations lead to longer waiting times for consecutive transmissions and increased variations between frames.

4.4.2 Periodic transmission with retransmission

This scenario involves enabling retransmission (limited to one attempt) when a probe response timeout is triggered. The performance of the throughput is presented in Figure 4.6. As observed in Figure 4.3, in the isolated scenario where no frame loss occurs, the retransmission mechanism has no visible effect; therefore, the throughput remains unchanged. However, once external interference is introduced, starting with the addition of 10 stations, a noticeable gain is observed. The throughput increases from approximately 490 (no retransmission) to 690 bps (with retransmission), representing a gain of roughly 200 bps.

Although the benefit of retransmission becomes less pronounced with higher station densities due to increased contention and interference, it consistently contributes to throughput improvements in all cases. Retransmission increases the probability of

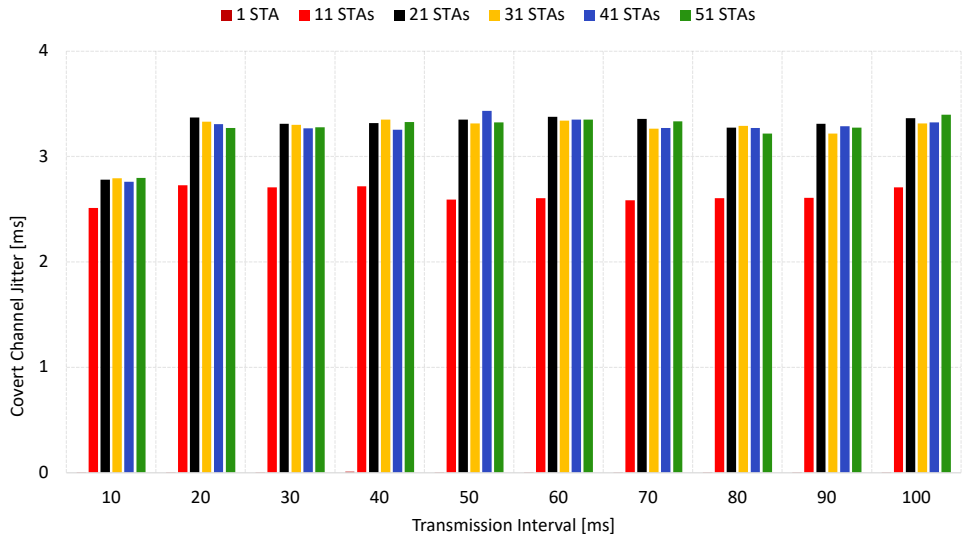


Figure 4.5: Impact of transmission interval and the number of stations on covert channel jitter

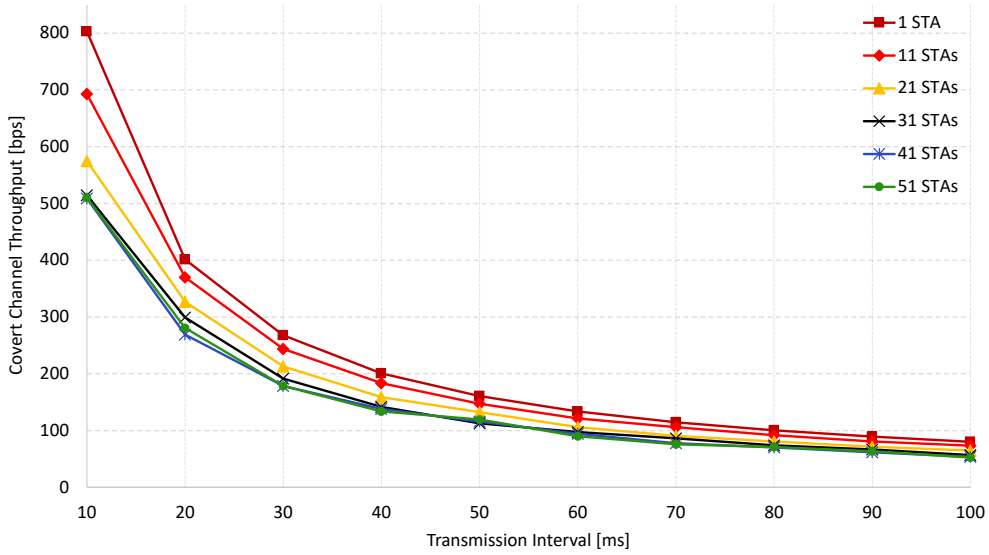


Figure 4.6: Impact of frame retransmission on covert channel throughput

successful frame delivery, which is also reflected in the increased frame efficiency in Figure 4.7. For example, when 10 external stations are present, the frame efficiency improves by about 24% compared to the scenario without retransmission, aligning with the observed throughput gain. Generally, the observed trend remains the same: Efficiency is impacted by the number of external stations, not by the transmission interval.

However, this improvement in reliability introduces a trade-off. As shown in Figures 4.8 and 4.9, jitter and delay increase. On average, the delay increases by approximately 2 ms and the jitter by around 1 ms. Retransmissions increase both delay and jitter because when a frame is not delivered at time t_1, the subsequent retransmission at time t_2 is still associated with the same frame. As a result, the total delay includes the time from the initial transmission attempt through all retransmissions until successful delivery. This

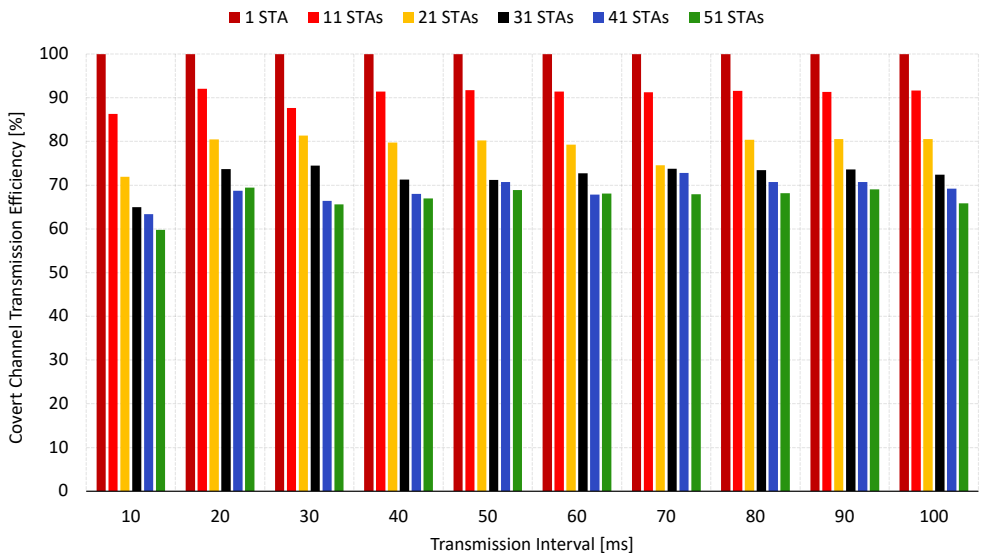


Figure 4.7: Impact of frame retransmission on covert channel efficiency

extended transmission period increases the average delay and introduces variability in inter-frame timing, thereby increasing jitter as well. Despite these increases, the additional reliability, which yields a throughput gain of nearly 200 bps, demonstrates that enabling retransmission is a practical strategy for enhancing covert channel performance in the presence of external traffic.

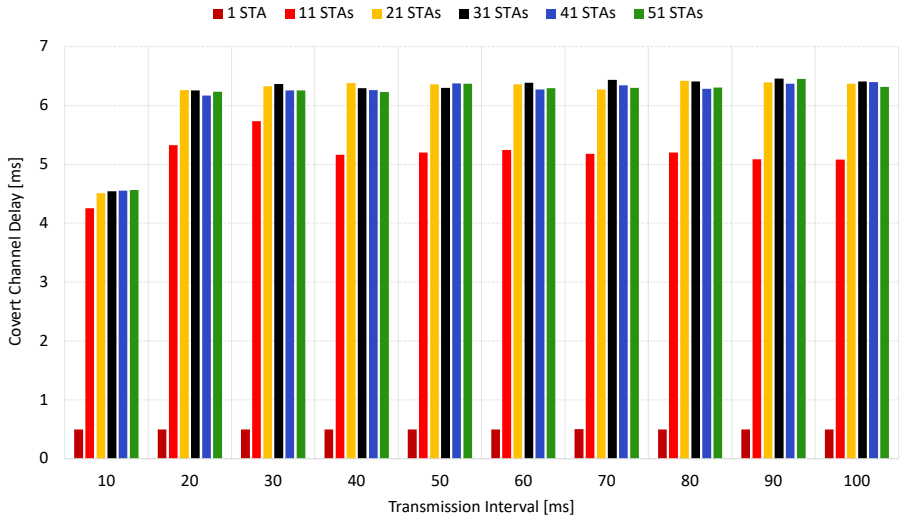


Figure 4.8: Impact of frame retransmission on covert channel delay

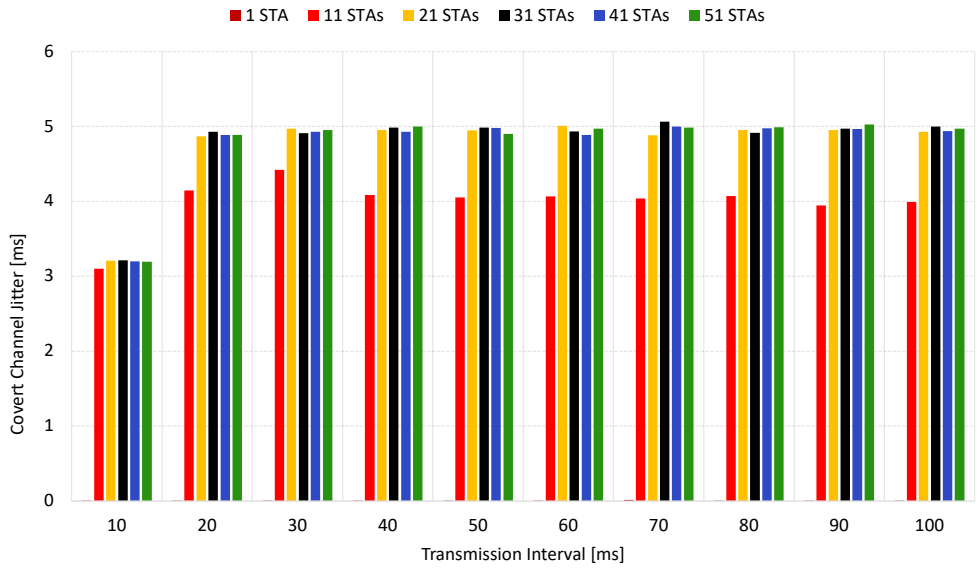


Figure 4.9: Impact of frame retransmission on covert channel jitter

4.5 Discussion of results

StegoRates is a covert channel that enables stations to exchange hidden data even while disconnected, achieving a maximum throughput of approximately 800 kbps. The analysis of the StegoRates covert channel reveals that two main parameters have a direct impact on its performance: the transmission interval and the number of stations sharing the wireless medium with the covert station. Based on the experimental results, the following conclusions can be drawn.

- *Transmission interval and throughput*: Shorter transmission intervals result in higher throughput. This is because stations generate traffic more frequently, increasing the rate at which covert bits are transmitted per second.
- *Transmission interval, delay, and jitter*: The transmission interval does not significantly influence throughput beyond a certain threshold. However, it has a clear impact on delay and jitter. Longer transmission intervals result in more stable delay and jitter characteristics, as a low volume of traffic is generated, leading to reduced contention and shorter transmission and receiver queues.
- *Impact of number of stations*: The number of competing stations has a more pronounced effect on overall network performance than the transmission interval. An increased number of stations leads to lower throughput and efficiency, while delay and jitter increase considerably.
- *Retransmission*: Enabling frame retransmission introduces reliability to the covert channel, thereby increasing both throughput and efficiency. This improvement comes at the cost of a moderate increase in delay (approximately 2 ms) and jitter (approximately 1 ms), which is an acceptable trade-off given the throughput gains.

The Tables 4.2 summarize the performance improvements observed when retransmission is enabled. Even in densely populated environments with numerous competing

stations, retransmission yields noticeable throughput gains (throughput and efficiency), indicated by the preceding plus sign. Even in very dense environments with 40 and 50 additional stations, the covert channel achieves throughput gains of 178 bps and 184 bps, respectively. These values are significant given the covert nature of the communication, demonstrating the channel's ability to improve its performance under high contention. The delay and jitter also increase; therefore, they are preceded by the plus sign as well.

It is important to note that these results are based solely on the use of the supported rates field. Additional improvements are expected with the inclusion of extended supported rates, which further increase the covert throughput capacity, resulting in higher covert throughput.

Table 4.2: Delta between retransmission and no retransmission scenario for throughput, delay, and jitter across station counts and transmission intervals

Stations	Tx Interval [ms]	Δ Throughput [bps]	Δ Delay [ms]	Δ Jitter [ms]
11 STAs	10	+195.09	+1.04	+0.58
	20	+116.32	+1.98	+1.45
	30	+75.44	+2.01	+1.39
	40	+55.25	+1.77	+1.33
	50	+46.48	+1.95	+1.46
	60	+37.23	+1.95	+1.43
	70	+33.20	+1.97	+1.42
	80	+28.91	+2.04	+1.49
	90	+24.35	+1.66	+1.29
	100	+24.32	+1.83	+1.38
21 STAs	10	+180.27	+0.99	+0.44
	20	+118.72	+2.21	+1.43
	30	+76.27	+2.35	+1.64
	40	+53.73	+2.34	+1.62
	50	+49.97	+2.40	+1.61
	60	+35.44	+2.43	+1.76
	70	+30.32	+2.14	+1.46
	80	+30.46	+2.25	+1.66
	90	+25.36	+2.43	+1.65
	100	+23.25	+2.28	+1.47
31 STAs	10	+149.52	+1.04	+0.41
	20	+113.55	+2.33	+1.12
	30	+59.89	+2.32	+1.60
	40	+48.48	+2.32	+1.63
	50	+39.31	+2.45	+1.63
	60	+34.32	+2.42	+1.59
	70	+32.24	+2.60	+1.73
	80	+28.77	+2.46	+1.60
	90	+26.99	+2.44	+1.67
	100	+17.97	+2.54	+1.70
41 STAs	10	+178.19	+1.12	+0.43
	20	+97.17	+2.27	+1.54
	30	+63.71	+2.36	+1.68
	40	+47.25	+2.33	+1.68
	50	+44.51	+2.43	+1.62
	60	+33.15	+2.33	+1.56
	70	+29.39	+2.40	+1.76
	80	+27.76	+2.39	+1.83
	90	+22.61	+2.40	+1.67
	100	+19.09	+2.46	+1.70
51 STAs	10	+184.13	+1.15	+0.40
	20	+114.13	+2.38	+1.62
	30	+68.48	+2.48	+1.64
	40	+41.82	+2.26	+1.67
	50	+51.92	+2.65	+1.55
	60	+30.32	+2.35	+1.63
	70	+25.09	+2.37	+1.64
	80	+28.59	+2.44	+1.78
	90	+25.73	+2.82	+1.90
	100	+19.36	+2.43	+1.48

5 Covert channel StegoMAC

5.1 StegoMAC operation

StegoMAC explores the ambiguity surrounding MAC address randomization, as discussed in Section 2.2.3, due to the absence of a standardized method for randomizing addresses and divergent implementations by vendors [130]. This concept involves transmitting *secret messages disguised as random MAC addresses*. In this method, the secret message is embedded in the Source Address (SA) field of the Probe Request (PR) frame, resulting in a *bandwidth of up to 48 bits per frame*. These random addresses are disposable for network observers and carry no significance other than their role in channel scanning and concealing the device's identity. This technique leverages the widespread adoption of MAC address randomization in modern Wi-Fi networks.

In the covert channel design, the AP must differentiate between PR frames generated by the covert station and those from regular stations. To avoid the predictability of sequential numbering in the Sequence Number (SN) field, a practice aligned with MAC address randomization recommendations, the SN does not follow the normal frame sequence. Instead, a 12-bit seed is constructed by combining a 4-bit random value with eight trailing zeros (e.g., 1011 00000000). This seed is then processed using a pre-agreed CRC-8 polynomial, producing a 12-bit output. The resulting value is placed directly into the SN field of the PR, as illustrated in Figure 5.1.

Upon receiving the frame, the AP performs the same CRC-8 division on the SN value. If the remainder resulting is zero, the frame is accepted as originating from the covert station; otherwise, it is considered a standard PR. This mechanism provides a lightweight authenticity check without disrupting normal network operations. In fact, altering the SN field is a common practice in MAC address randomization strategies, aiming to prevent persistent device tracking [43].

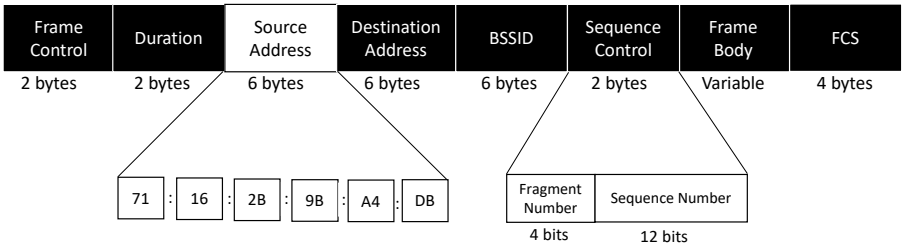


Figure 5.1: The structure of a covert message using MAC address randomization

5.1.1 Periodic transmission

The first variant presented in Figure 5.2 illustrates the workflow of the covert channel that periodically transmits covert messages, and the Algorithm 3 complements it as the practical implementation. The PRs are sent periodically, in the traditional way. In this setup, both the STA and AP share two specific randomized MAC addresses, one to signal the beginning of covert communication and another to indicate its end, using the SN field to provide verification. The covert exchange begins when the STA transmits a PR frame using the designated MAC address in the source address field to indicate initiation. Following this, the STA sends a sequence of PR frames at fixed intervals (e.g., every 10

milliseconds), designed as a *transmission interval*, where each SA field encodes a hidden message. To ensure reliability, the channel supports retransmissions; for example, if a PR frame (such as Probe Request #3) is not acknowledged, the STA retransmits it using the same randomized MAC address. Once all messages have been transmitted, the STA concludes the covert session by sending a PR frame with the termination MAC address. After this point, any subsequent PR frames are interpreted by the AP as standard network discovery traffic.

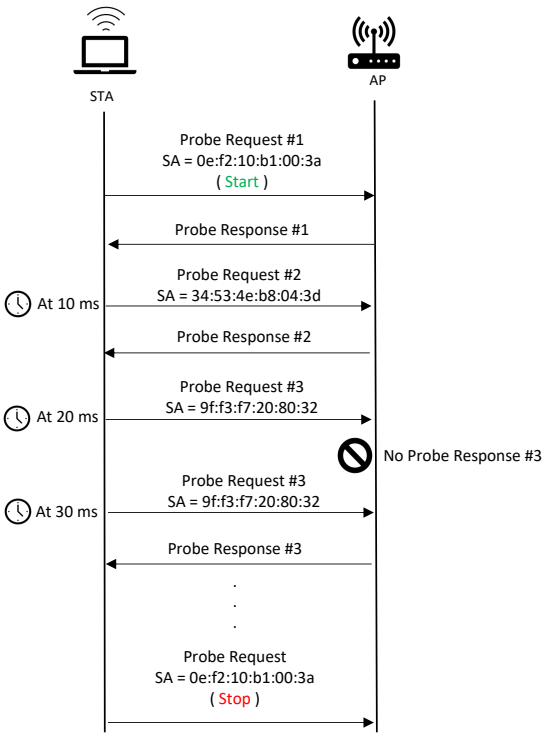


Figure 5.2: Diagram of StegoMAC operation using periodic transmission

5.1.2 Transmission using sliding window algorithm

The second variant of the covert channel implements the sliding-window protocol (SWP) to improve reliability and efficiency. The SWP is based on the adaptation of the modified Selective Repeat (SR) ARQ (Automatic Repeat Request) protocol [131]. Its operation is illustrated in Figure 5.3, and its practical implementation is presented in Algorithm 4. In this approach, the *window size* defines the number of PR frames the station STA is allowed to send consecutively without waiting for acknowledgments (probe responses). Each PR frame internally is assigned a unique sequence number and assigned a timeout for receiving the corresponding probe response. The process for initiating and terminating the covert channel remains consistent with the first variant.

At the start of communication, each transmitted covert frame reduces the available window size by one and activates a timeout for the expected probe response (Probe Request #1 and #2). Upon successful reception of a response, the window size is incremented by one, sliding the window forward (Probe Response #1). If the timeout expires without receiving a probe response, the STA marks the frame as lost and schedules it for retransmission, and this retransmission does not reduce the window size (Probe Request #2). If the window size reaches zero, the STA temporarily holds the transmission

Algorithm 3 StegoMAC pseudocode for the transmitter using periodic transmission

```
Input: msgId
Input: macAddr
procedure StartCovertTransmission(msgId, macAddr)
    macMsgId[msgId] ← macAddr
    msgIdAcked[msgId] ← false
    SendProbeRequest(macAddr)
    t ← now + transmissionInterval
    Schedule(StartCovertTransmission, msgId, macAddr, t)
    t_ack ← now + acktimeout
    Schedule(CheckAck, msgId, , t_ack)
end procedure
procedure CheckAck(msgId)
    for msgId in macMsgId.keys() do
        if msgIdAcked[msgId] = false then
            macAddr ← macMsgId[msgId]
            StartCovertTransmission(msgId, macAddr)
        end if
    end for
end procedure
```

and resumes only once acknowledgments are received and the window size increases. This protocol ensures that the STA transmits covert messages only when acknowledgments are being received, avoiding unnecessary retransmissions and conserving airtime. Unlike the first variant, which sends frames at fixed intervals regardless of acknowledgment status, the SWP allows continuous and acknowledgment-driven transmission, optimizing the use of available time while maintaining reliability.

5.2 StegoMAC properties and deployment scenarios

The presence of covert channels is challenging to detect due to the widespread use of MAC address randomization and the lack of standardization. Sending probe request frames with randomized source MAC addresses is generally considered a normal and benign activity. Since these random MAC addresses are temporary and change between scans, repeated transmissions from the same device are unlikely to raise suspicion, even if the sequence number patterns change. As long as the transmission rate remains within typical behavioral patterns, covert messages can blend in effectively as simple, disposable MAC addresses, even in the face of traffic monitoring and statistical analysis. Furthermore, the information is not embedded in plain text characters. Instead, each octet of the MAC address is mapped to a set of characters known only by the sender and receiver, creating a sort of private dictionary. As a result, what is broadcasted in the probe request does not directly resemble the original content, making the covert channel even more challenging to interpret, even if it is detected.

The covert channel operates with a high degree of transparency, as it utilizes probe requests, which are the only frame type broadcated to initiate the scanning process, and do not interfere with normal network functionality. These frames are typically transmitted before any authentication or association takes place, allowing the covert channel

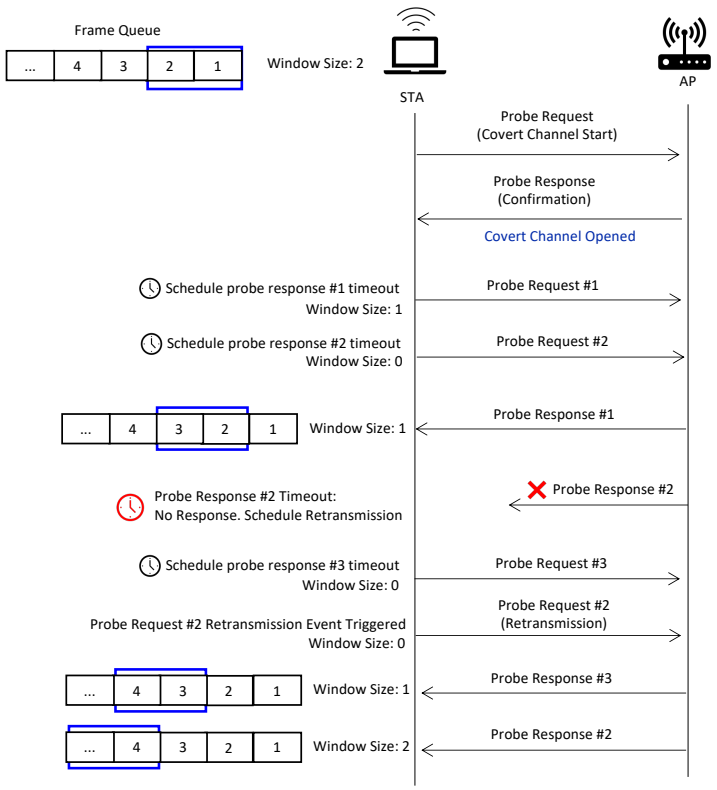


Figure 5.3: Diagram of StegoMAC operation using sliding window protocol

to remain outside the scope of most access point controls. However, the channel may consume additional bandwidth if the frequency of probe requests exceeds typical rates or if the station remains in state 1 (unauthenticated and unassociated) for extended periods while continuously transmitting high volumes of probe requests. In such cases, the abnormal traffic pattern could lead to increased airtime usage and raise suspicion through statistical analysis.

StegoMAC can be deployed independently or in conjunction with StegoRates to encode covert messages during the scanning phase, eliminating the need for network association or explicit data transmission. This allows the STA to remain passive and reduces its exposure to potential attackers. Additionally, MAC address randomization can serve as a lightweight authentication mechanism in untrusted environments, enabling the establishment of trust or ensuring confidentiality between the covert station and the AP.

5.3 Simulation scenarios and metrics

5.3.1 Environment and scenarios

The simulation environment was based on the NS-3, a widely used open-source platform for modeling and analyzing networking protocols and systems. NS-3, which was customized to meet our specific requirements. These modifications included generating randomized MAC addresses, implementing event scheduling for transmissions, managing acknowledgment checks, and other related functionalities.

The main simulation parameters are summarized in Table 5.1. To implement the covert channel, we utilized several NS-3 API components. The `Simulator::Schedule()`

Algorithm 4 Pseudocode of StegoMAC operation using sliding window protocol

```
Input: msgId
Input: macAddr
Variable: windowSize
Variable: macMsgId <msgId, macAddr>
Variable: msgIdAcked <msgId, bool>
procedure StartCovertTransmission(msgId, macAddr, isRetransmittion)
    while windowSize > 0 do
        macMsgId[msgId] ← macAddr
        msgIdAcked[msgId] ← false
        SendProbeRequest(macAddr)
        t_ack ← now + acktimeout
        Schedule(CheckAck, msgId, , t_ack)
        if isRetransmittion = false then
            windowSize ← windowSize −1
        end if
    end while
end procedure
procedure CheckAck(msgId)
    for msgId in macMsgId.keys() do
        if msgIdAcked[msgId] = false then
            macAddr ← macMsgId[msgId]
            StartCovertTransmission(msgId, macAddr, true)
        else
            windowSize ← windowSize +1
        end if
    end for
end procedure
procedure OnAckReceived(msgId)
    msgIdAcked[msgId] ← true
    windowSize ← windowSize +1
    StartCovertTransmission(msgId, macAddr, false)
end procedure
```

function was employed to manage the timing of both probe request transmissions and acknowledgment checks. The `Mac48Address` class was used for all operations related to MAC address handling, including generating randomized addresses, assigning them, and retrieving their values. The actual modification of the source MAC address occurs within the `StaWifiMac::SendProbeRequest(const Mac48Address& srcMacAddress)` method, where the covert message is embedded into the probe request frame header.

Simulations were repeated multiple times to ensure reliability, and the average values for each metric were computed. In all figures, the error margin for each simulation point, within a 95% confidence interval, did not exceed ±5%.

Table 5.1: StegoMAC simulation parameters

Parameters	**Value and Unit**
IEEE standard	802.11ac
Frequency band	5 [GHz]
Channel width	20 [MHz]
Channel number	36
Number of Tx and Rx antennas	1
Beacon interval	100 [ms]
Mobility model	Constant mobility
Probe request interval	10 [ms]
MCS index	VHT9
Guard interval	800 [ns]

5.3.2 Metrics

Evaluating the performance of any communication channel, whether standard or covert, requires a well-defined set of metrics. These performance indicators help quantify channel quality, identify factors that influence behavior, and guide optimization strategies to mitigate performance degradation. For the proposed covert channel, we adopt the conventional performance metrics commonly used in traditional network evaluation: throughput, delay, jitter, and channel efficiency.

- *Throughput*: Measures the rate at which the STA can transmit secret data through the covert channel. Reflects the number of probe request frames successfully acknowledged by the access point AP, multiplied by the number of covert bits encoded per frame. Since the covert payload in our case is the MAC address itself, each frame carries 48 bits of covert data and is defined in Equation 5.1 expressed in bps:

$$\text{Throughput} = \frac{N_{\text{PRequests}} \times 48}{T_{\text{simulation}}} \quad [\text{bps}] \tag{5.1}$$

- *Efficiency*: Defined in Equation 5.2, measures the proportion of successfully received probe request frames relative to those transmitted by the STA. Expressed as a percentage, this metric captures the reliability of covert communication by reflecting the actual success rate of frame delivery.

$$\text{Transmission efficiency} = \frac{N_{\text{PResponses}}}{N_{\text{PRequests}}} \times 100 \quad [\%] \tag{5.2}$$

- *Delay*: As shown in Equation 5.3, it represents the total time elapsed between the transmission of a PR frame and the reception of its corresponding probe response. It encompasses all latency sources within the transmission path, including propagation delay, queuing at the access point AP, processing time, and transmission delay:

$$\text{Delay} = \frac{1}{T_{\text{simulation}}} \cdot \left(\frac{1}{N} \sum_{i=1}^{N} \left(t_{\text{PResponse}_i} - t_{\text{PRequest}_i} \right) \right) \quad [\text{ms}] \tag{5.3}$$

- *Jitter*: As in Equation 5.4, the jitter refers to the variability in frame delivery delay, measuring how consistently packets arrive over time. It is computed using the formula in Equation 7.12, where N denotes the total number of frames:

$$\text{Jitter} = \frac{1}{T_{\text{simulation}}} \cdot \left(\frac{1}{N-1} \sum_{i=2}^{N} \left| \text{Delay}_i - \text{Delay}_{i-1} \right| \right) \quad [\text{ms}] \tag{5.4}$$

5.4 Performance evaluation

5.4.1 Periodic transmission

Figure 5.4 presents the results of the covert channel throughput analysis. Two main factors influence throughput performance: the Transmission Interval (TI) and the number of regular STAs operating concurrently on the same network as the covert STA. As the TI increases, a consistent decrease in throughput is observed, regardless of network density. For example, in a scenario with no competing traffic, the maximum observed throughput is approximately 4.8 kbps. However, when 10 additional stations join the network, the throughput drops to nearly 3 kbps, representing almost a 50% decrease. As more stations join, throughput continues to decline.

This behavior can be attributed to increased frame collisions and channel access contention, which limit the covert STA's ability to transmit effectively as the network density increases. It is also important to note that throughput drops significantly with the addition of 20 additional stations, but the decline becomes less pronounced beyond that point. This indicates that with more stations, the probability of successful transmission decreases at almost the same rate, a trend that is accentuated by increasing the TI.

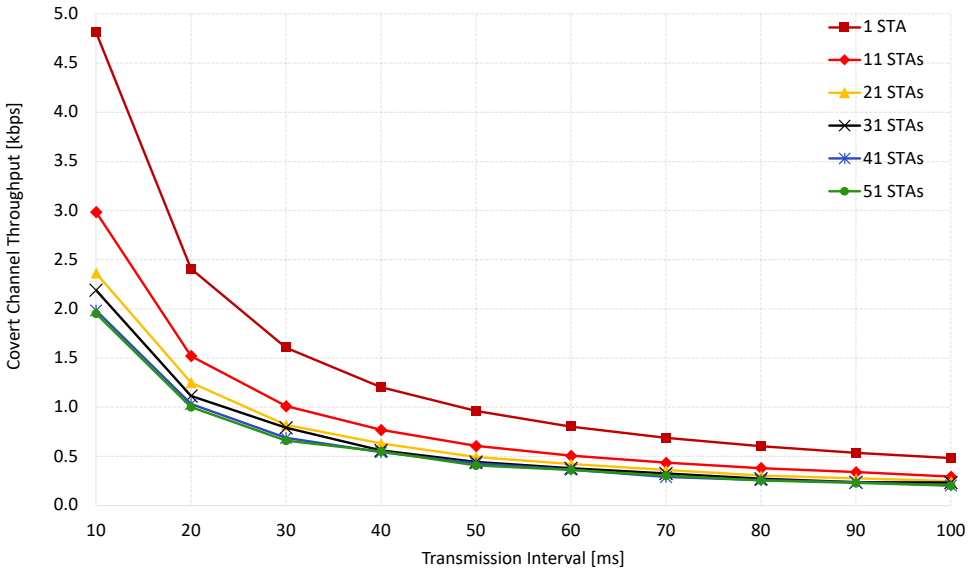


Figure 5.4: Covert channel throughput as a function of transmission interval in the scenario with retransmission disabled

Focusing on the transmission efficiency of the covert channel, as shown in Figure 5.5, the TI has minimal influence on the efficiency. Instead, the number of STA connections

to the network plays a more crucial role in how efficiently probe requests are delivered. This indicates that transmission efficiency is less dependent on the frequency of frame transmissions and is more influenced by the level of external traffic present during transmission.

In scenarios without competing traffic, no frame losses were observed, resulting in an initial transmission efficiency of 100%. However, when only ten regular STAs were introduced, there was a substantial decrease in efficiency, reducing it to approximately 60%. With 20 STAs active, the efficiency further decreased to below 50%. Beyond 20 additional stations, only minimal decreases in efficiency were observed, suggesting that the channel experiences high contention from 20 stations. Consequently, increasing the number of stations further has a minor impact on the current situation.

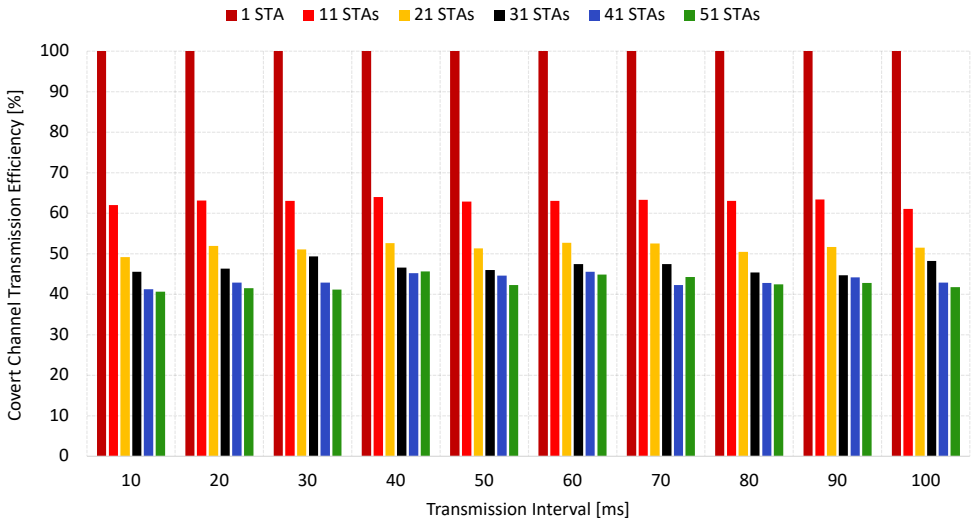


Figure 5.5: Covert channel efficiency as a function of transmission interval in scenario with retransmission disabled

Figure 5.6 illustrates the delay behavior of the covert channel. In the absence of external traffic, the delay remains stable at approximately 0.5 ms, regardless of the TI. However, when additional STAs are introduced into the network, a noticeable increase in delay is observed. For instance, with 10 additional stations, we see a consistent delay with varying TIs. Starting from a TI of 20 ms and beyond, there is a slight uptick in the delay. This indicates that both the TI and the presence of competing STAs contribute to the overall delay. As network congestion increases and the AP frame queue grows, frames are transmitted more slowly, resulting in longer response times.

The jitter analysis is presented in Figure 5.7, which shows variability in delay across different scenarios. In an isolated environment, jitter levels are minimal, indicating consistent transmission conditions. However, when external traffic is present, jitter increases due to fluctuating queuing and contention delays. With a TI of 10 ms, the jitter remains below 3 ms. However, as the network experiences congestion, such as with a 20 ms TI, jitter also rises above 3 ms, yet it remains below the overall delay. These findings demonstrate that increased traffic load and longer TIs lead to both higher delay and greater jitter.

An additional experiment was conducted to evaluate the effect of the covert channel on overall network throughput. The primary goal was to verify that the covert channel

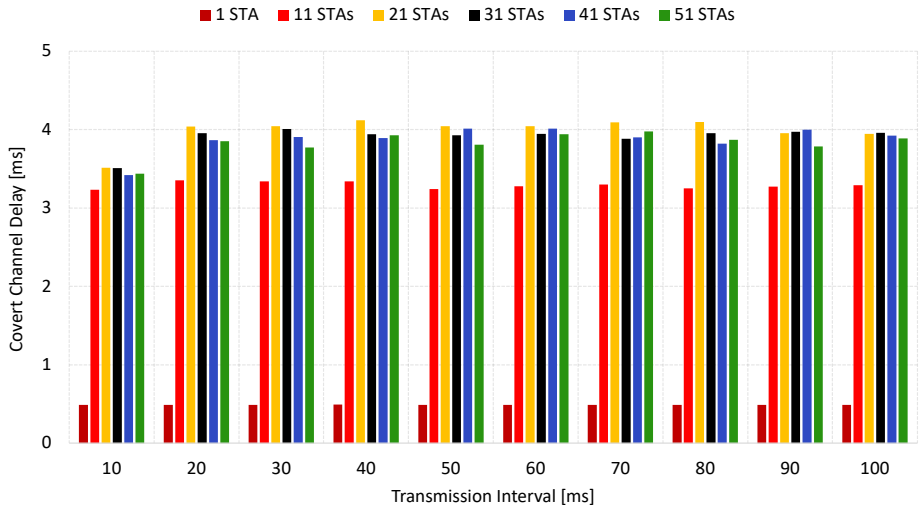


Figure 5.6: Covert channel delay as a function of transmission interval in scenario with retransmission disabled

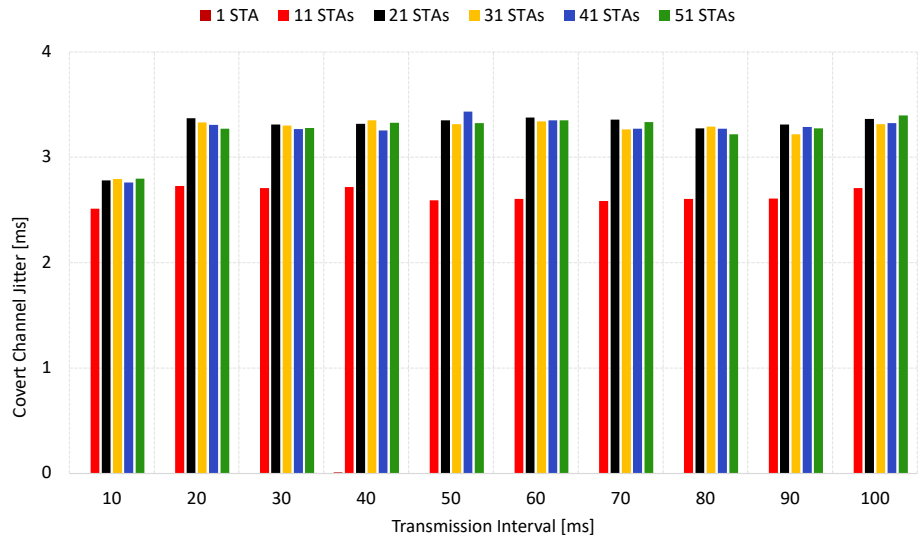


Figure 5.7: Covert channel jitter as a function of transmission interval in scenario with retransmission disabled

could operate without consuming excessive network resources or negatively impacting the performance experienced by the regular stations. The test used the same network topology as previous experiments but introduced multiple regular stations generating background traffic at saturation. Meanwhile, the covert STA transmitted PR frames at fixed intervals of 10 milliseconds, representing the most aggressive use of the covert channel.

The experiment was performed under varying network densities, ranging from 1 to 50 regular STAs. As expected, increasing the number of active STAs under saturation resulted in more frequent frame collisions and a reduction in overall network efficiency. For each configuration, we first measured the saturation throughput without the covert channel and then repeated the test with the covert STA active. This approach allowed for a direct comparison of network performance with and without the covert channel's

presence. The results, shown in Figure 5.8, help assess whether the covert channel can coexist with regular traffic while maintaining acceptable levels of network performance.

These results suggest that covert channels are more disruptive in lightly loaded networks, especially when the covert STA transmits using shorter TIs, such as below 10 ms. Consequently, these findings provide insight into how to deploy the covert channel and integrate its traffic into the overall network. In densely populated environments, larger TIs can be utilized since their impact is minimal. For average scenarios (e.g., up to 10 STA), longer TIs should be considered to avoid consuming regular channel bandwidth and to blend the traffic into a regular scanning activity pattern.

The most significant impact on regular network performance was observed in the scenario with only one regular STA and one covert STA, where the throughput of the regular station dropped by nearly 6 Mbps. This result suggests that the presence of the covert STA is more pronounced in less populated networks, which is a crucial factor to consider when attempting to conceal covert transmissions. However, as the number of regular STAs increased to 10, the negative effect of the covert STA diminished, resulting in a reduced impact of approximately 3 Mbps. In more densely populated networks, the presence of the covert STA caused only a minor reduction in throughput, with losses remaining below 1 Mbps.

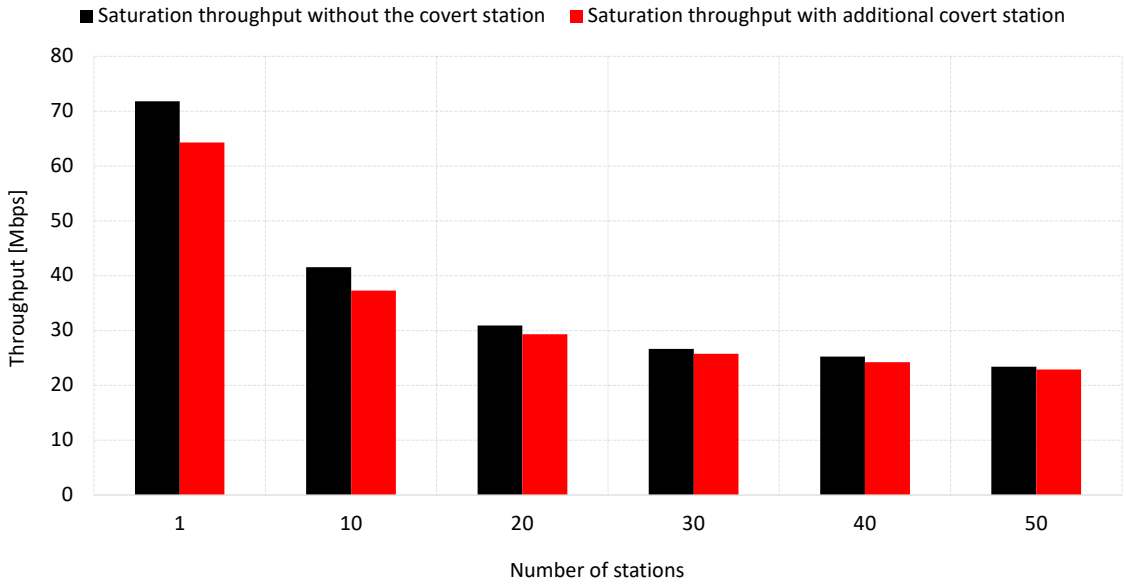


Figure 5.8: Impact of covert station activity with retransmissions disabled on the regular network performance

5.4.2 Periodic transmission with retransmission

In the second experimental scenario, enabling retransmission for the covert STA increases the likelihood of successful frame delivery, especially under higher traffic conditions. As shown in Figure 5.9, this effect is not immediately noticeable in the isolated environment, where frame loss is minimal, similar to the behavior observed in the initial scenario. However, as the number of regular STAs rises, the advantage of retransmission becomes more evident, helping to mitigate the impact of increased contention.

When comparing the two extremes of the TI, the 10 ms as the shortest and 100 ms as the longest, the overall trend of reduced throughput with longer TI remains consistent. However, in contrast to the previous scenario, the inclusion of retransmission yields substantial improvements in throughput as network density increases. For instance, at the 10 ms interval, adding 10 STAs resulted in a throughput gain of 1.169 kbps, while adding 20 STAs led to an increase of approximately 1 kbps. In more congested scenarios involving 30, 40, and 50 STAs, additional gains of roughly 700 bps, 1 kbps, and 900 bps were observed, respectively. Even at the longest TI of 100 ms, retransmission consistently contributed to performance improvements, yielding gains exceeding 100 bps across all network densities tested. These findings highlight the effectiveness of retransmission in enhancing covert channel reliability under load, regardless of transmission frequency.

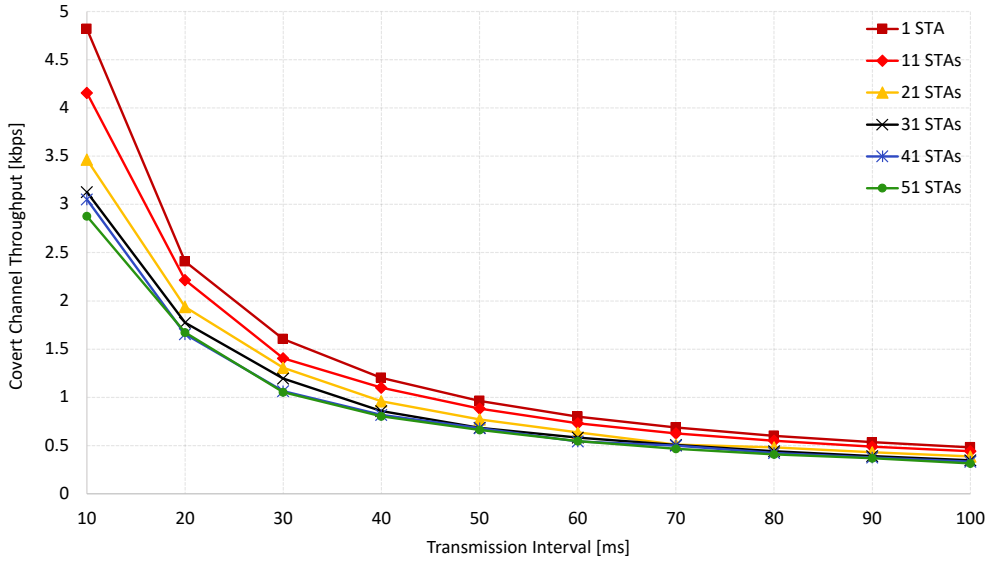


Figure 5.9: Covert channel throughput as a function of transmission interval in a scenario with retransmission enabled

Enabling retransmission significantly improves the performance of the covert channel in terms of transmission efficiency. As demonstrated in Figure 5.10, notably, efficiency is more strongly influenced by the number of active regular STAs in the network than by the TI values. Compared to the initial scenario, the inclusion of retransmission resulted in substantial efficiency gains. For example, with the addition of 10 additional STAs, the efficiency increased from approximately 60% to more than 80%. With 20 STAs, the efficiency improved by 22%. In scenarios with even higher densities, 30, 40, and 50 STAs, the efficiency gains were 19%, 22%, and 18%, respectively. Unlike the first scenario, where significant frame loss was observed in dense networks, the second scenario demonstrated that frame loss remained below 50% across all configurations.

Although retransmission improves both throughput and efficiency, it does introduce some performance trade-offs, particularly in terms of delay. As illustrated in Figure 5.11, retransmissions led to a noticeable increase in delay, on average up to 2.5 ms higher than the delay recorded in the first scenario. This can be attributed to the additional network load generated by repeated transmissions, especially under high-traffic conditions where collisions are more frequent. The delay increase is a natural consequence of multiple attempts being required to successfully deliver frames in a congested medium.

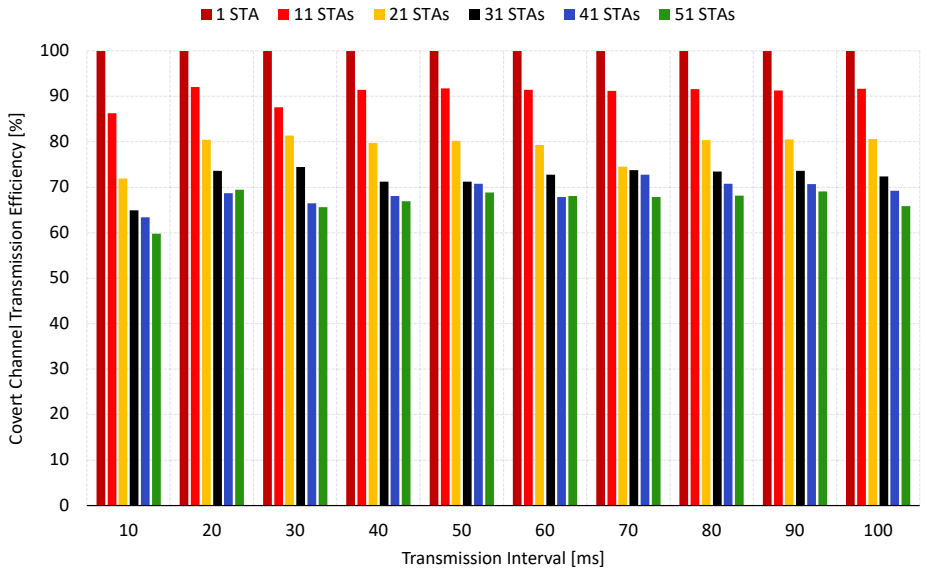


Figure 5.10: Covert channel efficiency as a function of transmission interval in a scenario with retransmission enabled

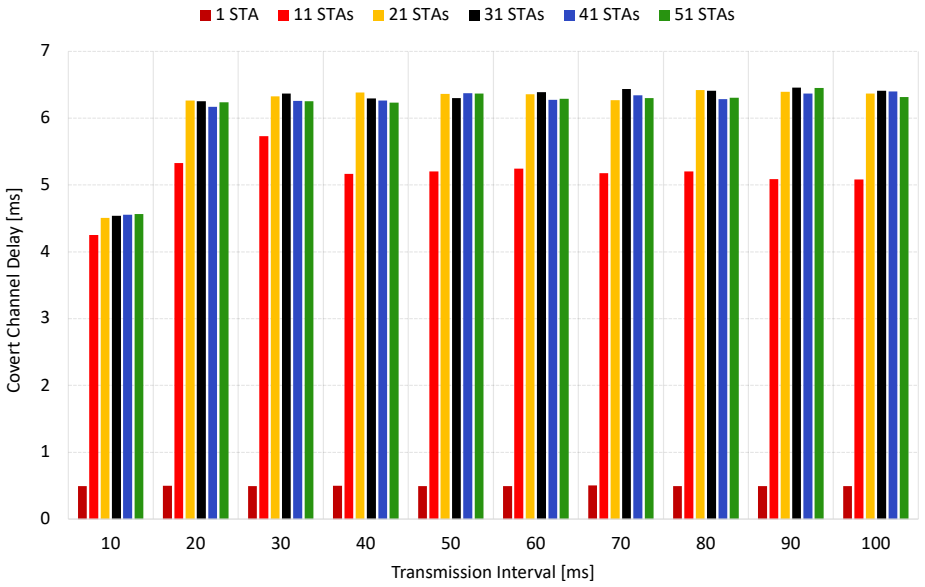


Figure 5.11: Covert channel delay as a function of transmission interval in a scenario with retransmission enabled

Additionally, retransmission contributes to increased jitter, as shown in Figure 5.12. In scenarios with background traffic, enabling retransmission resulted in an average jitter rise of approximately 1.5 ms. Despite these increases in delay and jitter, the overall performance benefits, most notably improved throughput and reliability, make retransmission a valuable strategy for enhancing covert channel communication in dense network environments.

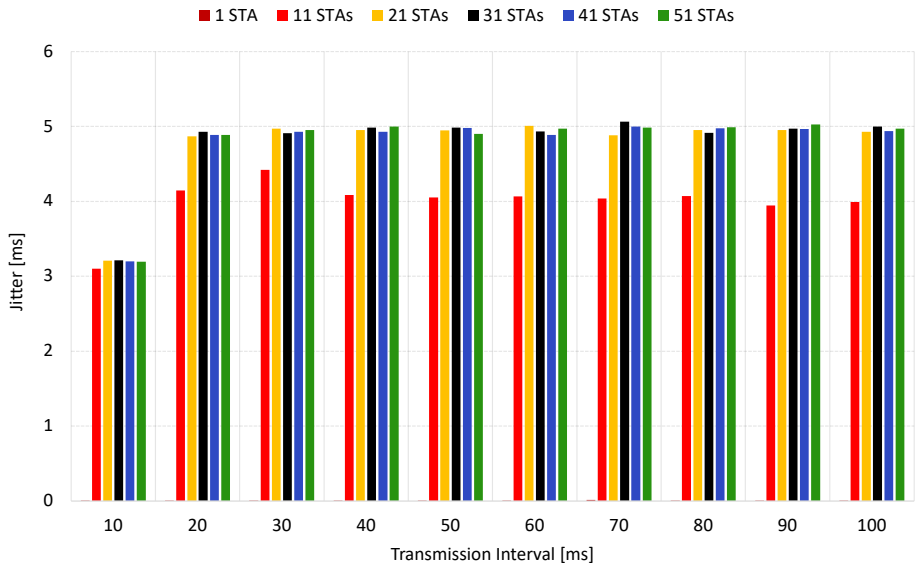


Figure 5.12: Covert channel jitter as a function of transmission interval in a scenario with retransmission enabled

5.4.3 Transmission using sliding window protocol

This time, the covert STA changes its transmission strategy. Instead of sending a message and waiting for an acknowledgment before sending the next one, it utilizes a window size that tracks the number of messages that can be sent without waiting for an acknowledgment. However, each probe request is still associated with a probe response timeout. In the absence of a corresponding probe response, the probe request is retransmitted.

To further evaluate the covert channel's performance, we explored the impact of varying the window size, ranging from 1 to 100000. The test conditions were aligned with the parameters described in Table 5.1. The covert station transmitted a sequence of messages with a 10 ms time interval until the window size reached zero, and the results are presented in Figure 5.13. In the absence of external traffic, throughput remained stable across all window sizes, a predictable result consistent with observations from previous scenarios, which demonstrated that, in isolation, frame efficiency remains at 100%. As a result, the transmission window size never drops to zero, ensuring uninterrupted transmission at a constant transmission interval.

The operation of the mechanism becomes evident when additional stations join the network. It requires a very large window size to fully leverage the sliding window protocol, specifically a window size of 1000 for 11 to 30 stations and 10000 for 40 and 50 stations, to achieve higher throughput compared to periodic transmission with retransmission.

When comparing these setups for periodic transmission, we can observe that the number of competing stations has a significant impact on the covert throughput. For instance, with 10 stations and 10 ms, none of the setups achieved a throughput exceeding 4.5 kbps. Conversely, when using a larger window size in a heavily populated environment, all setups, from 10 to 50 stations, achieved a covert channel throughput above 4 kbps (specifically, with 10 to 20 stations achieving above 4.5 kbps). While the impact of competing stations is noticeable, it is not as severe as with periodic transmission. Generally, using a larger window size allows all setups to achieve values above 4 kbps, and the impact of the competing station is mitigated.

Using a small transmission window size offers limited benefits, especially in densely contested environments, where the window can quickly become full. Sending messages at shorter transmission intervals exacerbates the issue: the mechanism frequently freezes as the window size reaches zero more regularly, resulting in repeated retransmissions. This significantly disrupts the communication flow. In contrast, a larger window size mitigates this problem by allowing continuous transmission over a longer period. Provides sufficient time for responses to arrive without stalling the process. As illustrated in Figure 5.13, a larger window size (i.e, from 10000) enables near-continuous transmission, resulting in a stable and consistent covert channel throughput. The impact of contention is still visible, but less pronounced than when using periodic transmission, even with retransmission.

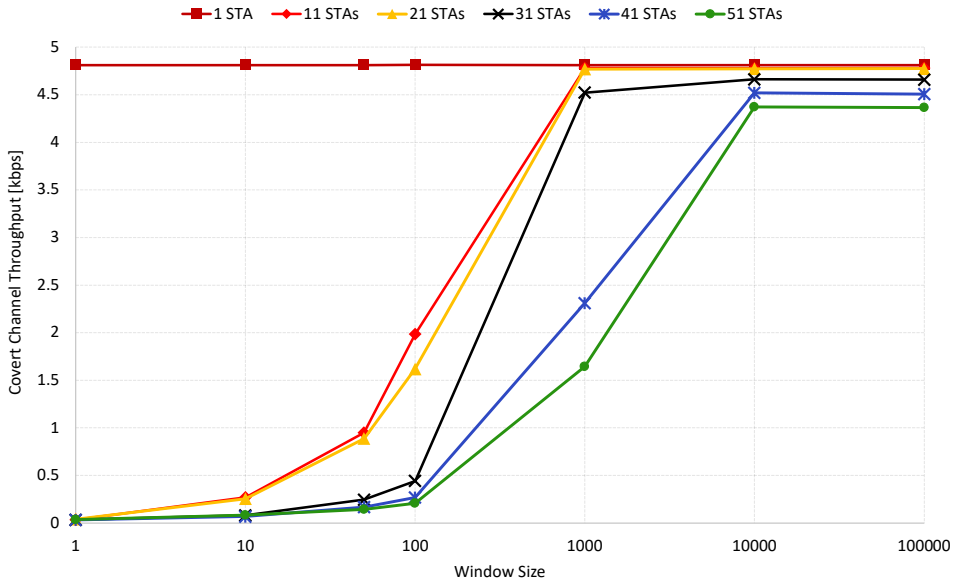


Figure 5.13: Covert channel throughput after the adoption of SWP

The improvement is further reflected in the results shown in Figure 5.14, which highlight the superior frame efficiency achieved through SWP. Compared to the retransmission-based scenario shown in Figure 5.10, it becomes clear that higher station densities typically lead to increased frame loss, even when retransmission is used. In those earlier configurations, the efficiency values rarely exceeded 90%, and such performance was only observed under isolated conditions. By contrast, SWP consistently maintained frame efficiency at or above 90% across all tested cases. In highly congested settings, specifically with 40 and 50 STAs, efficiency improvements of 30% and 31% were recorded, respectively, compared to the retransmission-only approach. These results underscore the effectiveness of SWP in maintaining high transmission reliability and efficiency, even in challenging network environments.

The adoption of the SWP caused a noticeable change in frame behavior within the covert channel, as illustrated in Figures 5.15 and 5.16, which detail the delay and jitter results, respectively. A clear relationship was observed between window size and both metrics: smaller windows consistently produced lower delay and jitter, whereas larger window sizes introduced greater variability and latency. That is attributed to the fact that the larger the window size, the higher the number of frames, which contributes to the number of frames queued at both the sender and receiver sides.

When these findings are compared with those of Figures 5.11 and 5.12, particularly

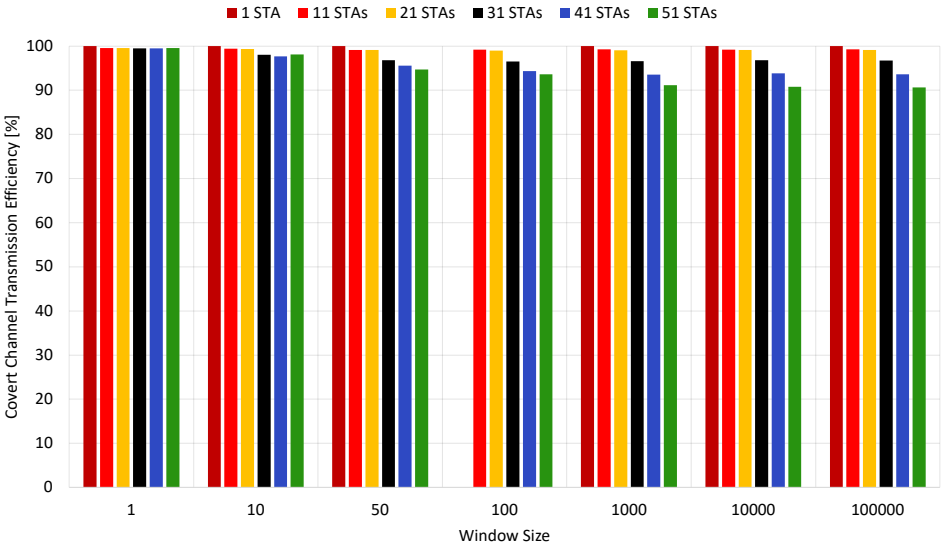


Figure 5.14: Covert channel efficiency after the adoption of SWP

under external traffic conditions with 10 to 50 additional stations, it becomes apparent that SWP effectively limited delay spikes. In fact, delays remained below 1 ms across all test cases. In more congested scenarios involving 30 or more STAs, SWP contributed to a notable performance boost, with an average delay reduction of 4.2 ms and an improvement in jitter of 3.8 ms compared to the retransmission-only approach.

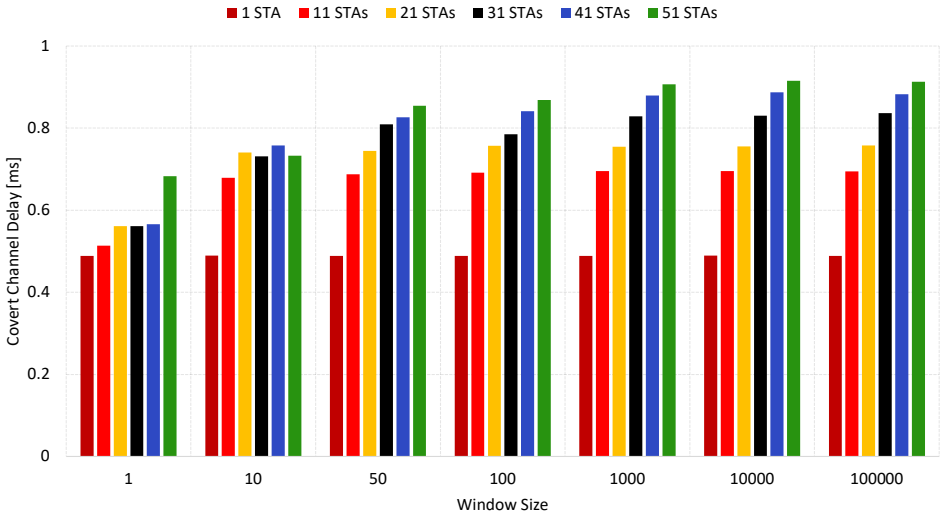


Figure 5.15: Covert channel delay after the adoption of SWP

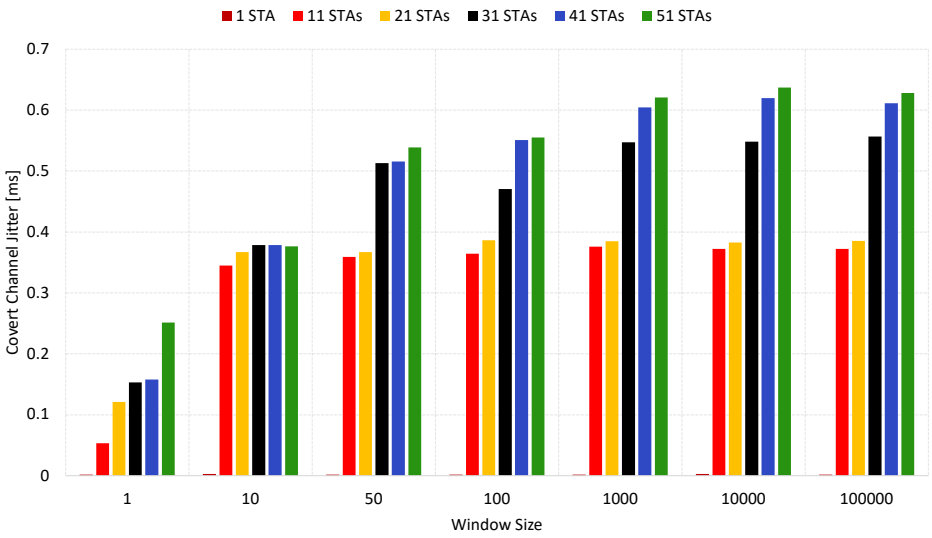


Figure 5.16: Covert channel jitter after the adoption of SWP

5.5 Discussion of results

StegoMAC establishes a high-throughput, high-efficiency covert channel by embedding hidden data within probe request frames, achieving up to 4.8 kbps throughput with over 90% efficiency, even in dense environments with around 50 stations. The analysis considered three transmission strategies for covert messaging: periodic transmission without retransmission, periodic transmission with retransmission, and a sliding window strategy. Each approach yielded distinct performance metrics, leading to the following conclusions:

- *Scenario without retransmission:* Covert messages are sent periodically using randomized MAC addresses without any acknowledgment or retransmission mechanism. As the number of stations increases, both throughput and efficiency decrease due to higher contention and frame loss. This method introduces minimal additional delay and jitter, but suffers from lower throughput and reliability in dense networks.

- *Scenario with retransmission enabled:* Retransmissions are triggered when a frame is not acknowledged, increasing the chance of successful covert message delivery. Notable gains in throughput (maximum throughput of around 1 kbps) and frame efficiency (maximum efficiency of around 20%) are observed across all tested station densities. However, extra probe request frames introduce overhead, resulting in a longer delay and jitter.

- *Scenario using Sliding Window Protocol:* Covert messages are transmitted in bursts, up to a predefined window size, without waiting for acknowledgment after each frame. The window slides forward as acknowledgments are received, allowing efficient and continuous transmission. SWP significantly improves throughput (gain of approximately 1.5 kbps) and efficiency (minimum 90%) while also reducing delay (maximum 0.91 ms) and jitter (maximum 0.62) compared to periodic transmission with retransmission, mainly in scenarios with a high number of competing stations.

StegoMAC supports multiple transmission strategies adaptable to different network conditions. All configurations were evaluated under a uniform transmission interval of 10 ms for consistent comparison. Figure 5.2 summarizes peak throughput, efficiency, delay, and the mechanisms used in each case.

Table 5.2: Comparison of the three transmission strategies employed in the StegoMAC, based on the highest achieved metrics

Stations	**Scenario**	**Throughput [kbps]**	**Delay [ms]**	**Jitter [ms]**	**Efficiency**
1 STA	1 - w/o retrans.	4.8	0.4	0.002	100
	2 - with retrans.	4.8	0.4	0.002	100
	3 - with SWP	4.8	0.4	0.002	100
11 STAs	1 - w/o retrans	2.98	3.23	2.51	62
	2 - with retrans.	4.15	4.25	3.09	86.27
	3 - with SWP	4.78	0.69	0.37	99.29
21 STAs	1 - w/o retrans	2.3	3.5	2.7	49
	2 - with retrans.	3.46	4.5	3.2	71.93
	3 - with SWP	4.77	0.75	0.38	99.11
31 STAs	1 - w/o retrans	2.19	3.51	2.79	45
	2 - with retrans.	3.12	4.54	3.21	64.96
	3 - with SWP	4.66	0.83	0.54	96.8
41 STAs	1 - w/o retrans	1.98	3.42	2.76	41.18
	2 - with retrans.	3.05	4.55	3.2	63.39
	3 - with SWP	4.52	0.88	0.61	93.86
51 STAs	1 - w/o retrans	1.95	3.4	2.7	41
	2 - with retrans.	2.87	4.56	3.19	59.79
	3 - with SWP	4.37	0.91	0.62	90.68

6 Covert channel StegoBackoff

6.1 StegoBackoff operation

Based on the DCF mechanism (see Section 2.2.5 for a discussion about the DCF mechanism), the StegoBackoff encodes a secret message using the *parity of the backoff slot*: an even-numbered slot represents a binary 0, while an odd-numbered slot represents a binary 1. This method yields a *bandwidth of 1 bit per frame*. Since encoding takes place within the standard channel access behavior, any device within range, whether a station or an access point, can passively observe the transmissions and act as a receiver.

The logic used by the covert sender is detailed in Algorithms 5. The sender retrieves a secret bit from its buffer and then selects a random backoff value. After this selection, it conditionally adds an extra time slot, if necessary, to ensure that the parity of the backoff aligns with the transmitted bit.

On the receiving side, as described in Algorithm 6, the covert receiver continuously monitors the channel. It measures the time elapsed between the end of the DIFS period and the moment the time frame is received. The receiver subtracts the DIFS duration from this interval to calculate the number of backoff slots that the sender deferred. Based on the parity of that value, the receiver determines the corresponding bit.

Algorithm 5 Pseudocode for the sender in StegoBackoff

```
Input: bit — covert bit to transmit
Variable: cw — contention window size
procedure DifsExpiredEvent(bit)
    backoff = Random(0, cw)
    if bit == 0 && backoff % 2 == 1 then
        backoff = backoff + 1
    else if bit == 1 && backoff % 2 == 0 then
        backoff = backoff + 1
        UpdateLatestBackoff(backoff)
        StartBackoffCountdown(backoff)
    end if
end procedure
```

To demonstrate the practical operation of the proposed covert channel, consider the scenario depicted in Figure 6.1. In this example, a covert sender maintains a queue of secret bits to transmit, specifically a sequence of two bits: the first is *1*, and the second is *0*. Following the DIFS period, the covert sender draws a random backoff value of 5 slots and begins its countdown. During the same time, a regular station also attempts to access the channel and selects a smaller backoff value of 3 slots, thus winning the competition. As a result, the covert sender pauses its countdown and waits until the ongoing transmission concludes. Once the regular station completes its transmission and a new DIFS period passes, the covert sender resumes its countdown with 2 remaining slots, while the regular station draws a 10-slot backoff. Upon reaching zero, the sender deliberately waits for one extra slot to ensure that the total number of slots elapsed since DIFS is odd, corresponding to the transmission of *bit 1*. After sending the first covert bit, the covert sender draws a new backoff value of 4 slots. At the same time,

Algorithm 6 Pseudocode for the receiver in StegoBackoff

Input: *tslot* — duration of one time slot in microsends
Input: t_{DIFS} — time when DIFS ended
Input: t_{frame} — time frame is detected on the channel
procedure RECEIVEDATAFRAME(t_{DIFS}, t_{frame}, *tslot*)
  $t_{\text{elapsed}} \leftarrow t_{\text{frame}} - t_{\text{DIFS}}$
  `nslots` $\leftarrow \left\lfloor \frac{t_{\text{elapsed}}}{\texttt{tslot}} \right\rfloor$
  if `nslots` mod $2 = 0$ **then**
    `secretBit` $\leftarrow 0$
  else
    `secretBit` $\leftarrow 1$
  end if
end procedure

the regular station is still counting down from its initial backoff value (10 slots), now resumed at 7. In this case, the covert sender wins the contention and transmits its frame without additional delay, as the total number of slots since DIFS is even, aligning with the intention of transmitting *bit 0*.

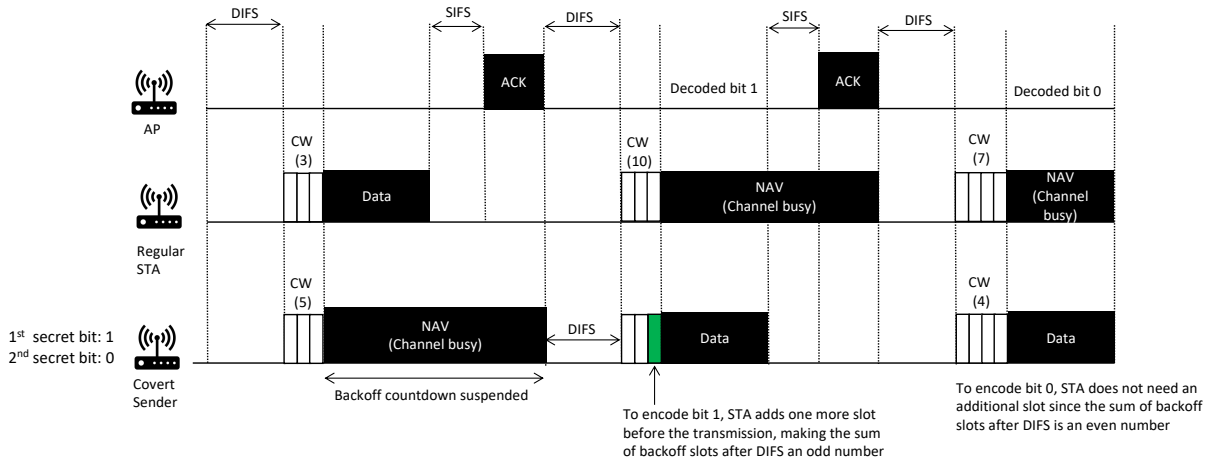


Figure 6.1: Example of StegoBackoff operation for encoding the secret bit sequence 10

6.2 StegoBackoff properties and deployment scenarios

The strength of StegoBackoff is based on its resistance to steganalysis. Unlike methods that modify reserved fields or introduce irregular timing patterns, which can raise suspicions during traffic analysis, this covert channel utilizes the inherently random nature of the DCF backoff. Consequently, the transmission patterns remain indistinguishable from the legitimate contention behavior. For external observers or monitoring systems, each backoff period appears as a regular part of the channel contention process. Even if there are slight alterations, the backoff values are not arbitrarily chosen; instead, they are drawn randomly according to the protocol, thereby maintaining the inherent stochastic nature of the mechanism. The adjustment is applied only when necessary, when the

randomly selected slot does not match the parity of the bit being encoded, making this manipulation virtually indistinguishable from standard behavior.

The covert channel proposed in this work achieves a high degree of transparency by utilizing the backoff mechanism, an integral part of the 802.11 standard channel access procedure. This technique involves the optional addition of a single backoff slot to ensure that the parity of the selected backoff value aligns with the bit being transmitted. This minor adjustment does not significantly affect network performance, as it neither interferes with the sender's behavior nor impacts other stations competing for the medium.

In terms of practical applications, this mechanism can be used for authentication, allowing devices to verify each other's identities without exposing sensitive data. This approach helps mitigate threats such as MAC address spoofing, packet replay attacks, and rogue access points seeking to impersonate legitimate devices. Additionally, in untrusted or monitored environments, the covert channel provides a secure means of transmitting confidential information. Since covert messages blend in with standard network behavior, they remain invisible to passive observers, ensuring privacy and confidentiality even under surveillance. This makes the proposed channel a valuable tool for enhancing both authentication and data protection in wireless deployments.

6.3 Simulation scenarios and metrics

6.3.1 Environment and scenarios

The covert channel was developed and evaluated using the ns-3 network simulator, a discrete-event simulation tool widely used to study various types of networks. This open-source framework, implemented in C++ and Python, provides comprehensive support for modeling 802.11 networks. The selection of ns-3 was motivated by its robust support for the 802.11ax amendment, its active developer community, and the platform's general maturity and extensibility. Its modular design enables efficient abstraction of the underlying MAC and PHY layer mechanisms, allowing realistic experimentation in controlled environments. The implementation leverages the existing NS-3 API, with specific custom modifications introduced to support the covert channel functionality. In particular, the backoff mechanism was adapted to dynamically reflect the covert message content. This was achieved by modifying the `ChannelAccessManager::UpdateBackoff()` method, which updates the backoff slot before transmission. Internally, this method invokes `Txop::UpdateBackoffSlotsNow(uint32_t nSlots, Time backoffUpdateBound, uint8_t linkId)` to update the number of backoff slots, ensuring the backoff value aligns with the intended covert message encoding.

The main simulation parameters used in the study are summarized in Table 6.1. Furthermore, in all experiments, the statistical margin of error per data point, calculated at a 95% confidence interval, remained within ±5%.

6.3.2 Metrics

During the performance evaluation of StegoBackoff across different setups, we define and calculate the following metrics:

- *Throughput*: Measures the rate at which the STA can transmit secret data through the covert channel. Since the bandwidth is 1 bit encoded per frame, the covert

Table 6.1: StegoBackoff simulation parameters

Parameter	**Value and Unit**
IEEE standard	802.11ax
Transport protocol	UDP and TCP
Payload size	1024 [bytes]
Frequency band	5 [GHz]
Channel width	20 [MHz]
Guard interval	800 [ns]
Time slot	9 [µs]
SIFS	16 [µs]
DIFS	34 [µs]
MCS index	11
Mobility model	Constant mobility
RTS/CTS	Enabled and disabled
Number of Tx and Rx antennas	1

throughput is defined as the number of data frames that the AP has successfully received during the simulation time, expressed in Equation 6.1:

$$\text{Throughput} = \frac{F_{\text{RX}}}{T_{\text{simulation}}} \quad [\text{bps}] \tag{6.1}$$

- *Efficiency*: The definition in Equation 6.2 measures the proportion of data frames that are received successfully by the AP compared to those transmitted by the STA. This metric, expressed as a percentage, indicates the actual delivery rate of the covert channel, which also reflects the performance of the regular channel.

$$\text{Efficiency} = \frac{F_{\text{RX}}}{F_{\text{TX}}} \times 100 \quad [\%] \tag{6.2}$$

- *Delay*: Equation 6.3 represents the average RTT per data frame. Measures the time interval between the transmission of a data frame and the reception of its corresponding ACK. This metric includes both the transmission and acknowledgment delays and reflects the responsiveness of the channel:

$$\text{Delay} = \frac{\sum_{i=1}^{N} (t_{\text{ACK}i} - t_{\text{TX}i})}{N \cdot T_{\text{simulation}}} \quad [\text{ms}] \tag{6.3}$$

- *Jitter*: As in Equation 6.4, the jitter measures the variability of frame delay over time. It is computed as the average absolute difference between the delays of consecutive data frames, providing insight into the channel stability or instability.

$$\text{Jitter} = \frac{1}{(N-1) \cdot T_{\text{simulation}}} \sum_{i=2}^{N} \left| \text{Delay}_i - \text{Delay}_{i-1} \right| \quad [\text{ms}] \tag{6.4}$$

6.4 Performance evaluation

6.4.1 Isolated covert station

Initially, we designed a scenario consisting solely of a covert STA and an AP, without competing traffic from other STAs. This setup was intended to assess the expected behavior of the covert channel in the presence of external interference. We repeated this procedure using different frame sizes, ranging from 20 bytes to 2000 bytes, to analyze how variations in payload size affect the throughput of the covert channel under increasing load conditions (from 1 Mbps to 200 Mbps). The simulation results for this isolated setup are presented in Figure 6.2.

The results obtained demonstrate that the maximum achievable throughput of the covert channel is approximately 4.7 kbps. As the data frame size increases, the covert throughput decreases. This behavior is attributed to the fact that larger payloads require more time to transmit, occupying more airtime, and thereby reducing the number of frames that can be sent per second. Additionally, the offered load influences the throughput as it determines the rate at which data is generated. However, this effect is constrained by the size of the frame. For small frames (e.g., 20 bytes), saturation is reached at a low offered load (around 1 Mbps), and the throughput remains nearly constant thereafter.

In contrast, larger frames require a higher offered load to achieve saturation. For example, with 2000-byte frames, saturation is observed starting from 60 Mbps. These results highlight two key aspects of StegoBackoff: smaller frames yield higher covert throughput due to their shorter transmission times, and saturation thresholds vary with frame size, thereby affecting the maximum throughput.

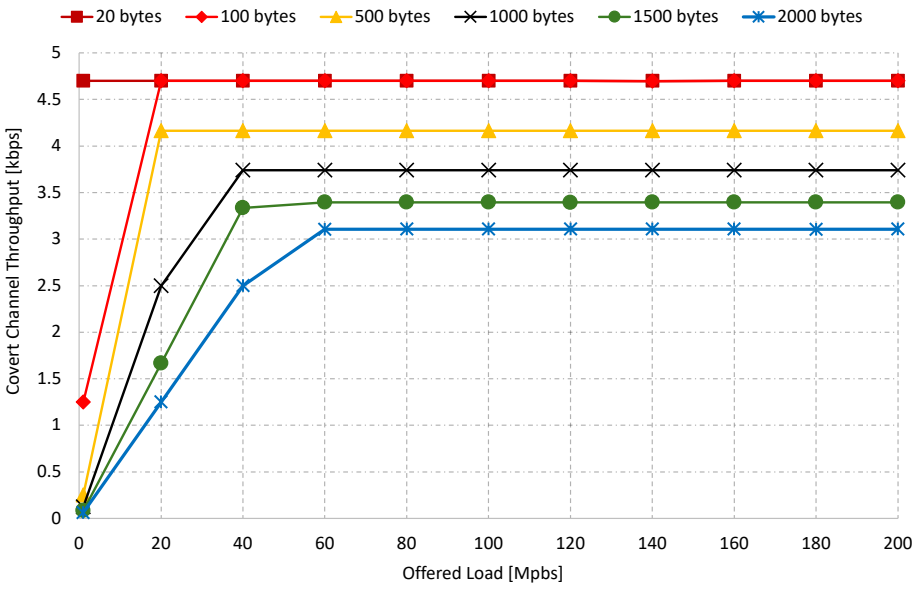


Figure 6.2: Covert channel throughput as a function of offered load for varying payload sizes

In Figure 6.3, we present channel efficiency, which represents the amount of data generated according to the offered load. With a combination of a low offered load (e.g., 1 Mbps) and a small payload (e.g., 20 bytes), the data generation rate is higher compared

to heavier payloads; in this case, the efficiency is approximately 70%. This behavior is because the massive volume of very small frames introduces additional overhead, overwhelming the AP's processing buffer. This introduces delays due to the number of frames to process and acknowledge, which increases the probability of frame loss due to packet expiration in the queue as the queue becomes larger. Conversely, higher payload sizes at the same rate of 1 Mbps result in fewer frames being generated per second. This lower frame generation rate translates into reduced traffic volume, which in turn decreases the AP's receiving buffer queue and the collision probability. As a result, the covert channel exhibits nearly 100% efficiency.

In general, the observed behavior indicates that, regardless of frame size, as the offered load increases, frame efficiency decreases, with a more pronounced effect observed with smaller frame sizes. This behavior is attributed to the fact that a high data frame generation on the sender side overwhelms the resources of the receiving side. Although an unreliable transport protocol like UDP is used, the 802.11 MAC layer implements a reliable mechanism that forces the AP to process and acknowledge every received frame.

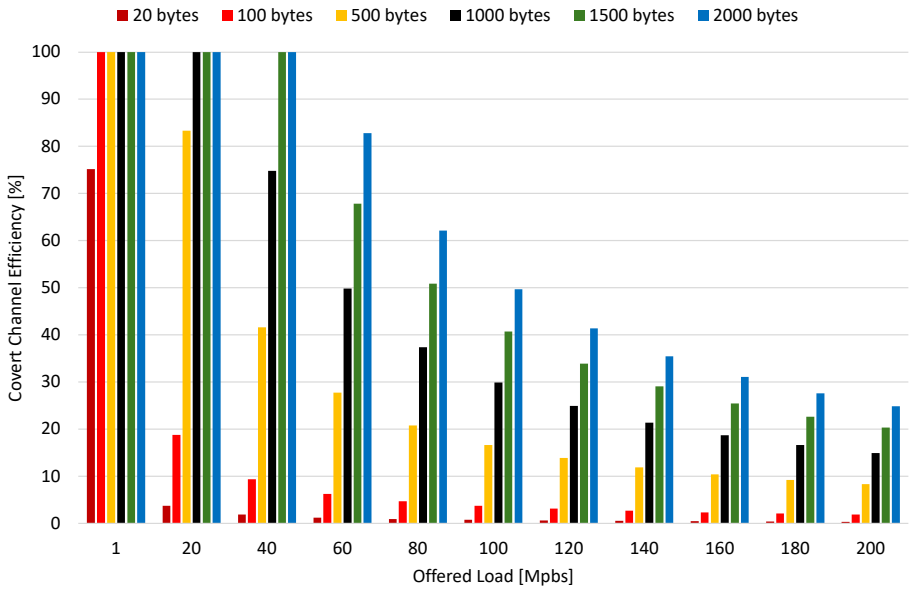


Figure 6.3: Covert channel efficiency as a function of offered load for varying payload sizes

6.4.2 Impact of increasing station density

We decided to increase the number of competing stations on the network, initially deploying both UDP and TCP traffic to compare their performance when StegoBackoff is used without RTS/CTS, and then in the same scenario with RTS/CTS enabled. As shown in Figure 6.4, the throughput of the covert channel varies under different traffic conditions, highlighting distinct behaviors between UDP and TCP transmissions.

At first, with only one additional station, UDP consistently achieves higher throughput than TCP. This advantage comes from UDP's connectionless, best-effort delivery model, which avoids the overhead of acknowledgments, congestion control, and retransmissions that TCP requires. However, when the RTS/CTS mechanism is enabled, the throughput decreases for both protocols. The effect is more pronounced for UDP, where

the throughput drops from about 2 kbps to 1.2 kbps. This reduction occurs in UDP because once additional control overhead is introduced by RTS/CTS, the extra exchange directly reduces the throughput. TCP, by contrast, maintains more stable throughput since its reliability mechanisms (acknowledgments, retransmissions, congestion control) make it less sensitive to the added coordination overhead of RTS/CTS.

When five additional stations are introduced, the impact of channel contention becomes visible. UDP continues to outperform TCP, and this trend persists throughout the experiment as the number of competing stations increases to 10, 15, and 20. Interestingly, the influence of RTS/CTS has a more positive impact once multiple stations are active. With only a single competing station, RTS/CTS introduces noticeable overhead, reducing throughput. However, as the number of stations increases, channel access is more evenly divided, and the efficiency of RTS/CTS in reducing collisions begins to outweigh its overhead. These results show better performance when RTS/CTS is enabled compared to when it is only one competing (less throughput difference when comparing the enabled and disabled RTS/CTS). Beyond 10 stations, the throughput steadily decreases but eventually converges to nearly uniform levels across scenarios, reflecting the onset of network saturation where additional stations contribute little to further throughput differentiation.

The results indicate that the RTS/CTS mechanism has a more significant favorable effect on covert channels using the TCP protocol compared to those using UDP for a small number of stations. As station density increases, RTS/CTS proves to be similarly beneficial for both protocols, since the relative overhead for UDP decreases due to the higher frequency of collisions, which makes the coordination mechanism more effective in preventing collisions or holding transmissions.

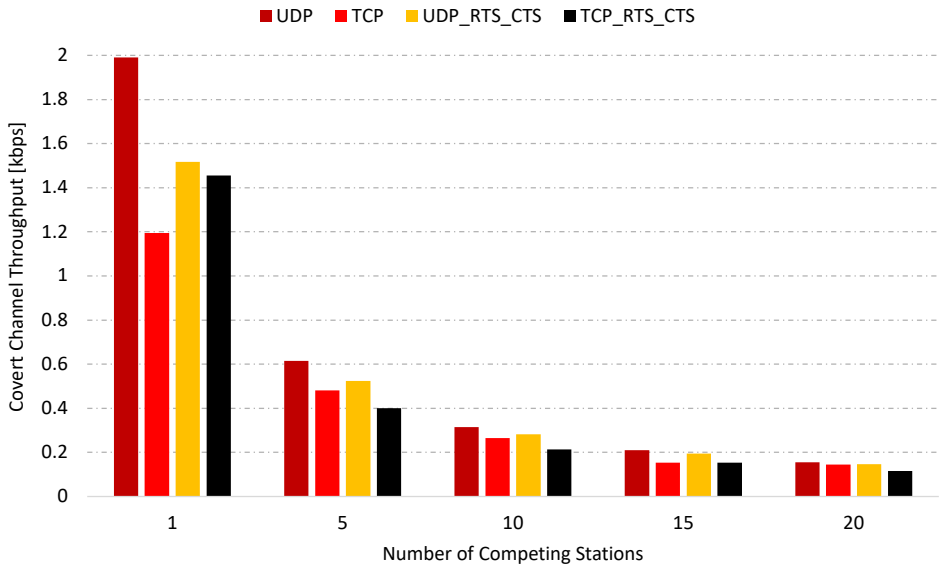


Figure 6.4: Comparing throughput of covert communication using UDP and TCP traffic, both with and without RTS/CTS, under different station densities

Figure 6.5 illustrates the efficiency of the covert channel, highlighting the volume of transmitted frames and the channel behavior under increasing load conditions. Notably, the results indicate that higher throughput does not always correspond to greater efficiency. In scenarios involving TCP traffic, we observed a higher ratio of transmitted

frames to successfully received frames compared to those using UDP. TCP consistently achieved a frame efficiency of nearly 99.9%.

TCP achieves nearly 99% frame performance because it incorporates mechanisms for flow control and congestion avoidance, which dynamically adjust the sending rate based on feedback from the receiver and overall network conditions. This ensures a more balanced transmission process, frames are acknowledged and retransmitted when necessary, providing reliable delivery even under challenging network conditions, ensuring a successful frame delivery.

In contrast, UDP lacks these control mechanisms and transmits data at a constant rate without regard for congestion or the receiver's capacity. Although this can result in higher raw throughput under ideal conditions, as observed in Figure 6.4, it also leads to significant frame loss when the network experiences interference or contention. Consequently, UDP's frame performance remains below 20% in these scenarios, reflecting its vulnerability to collisions and packet drops in a shared channel environment.

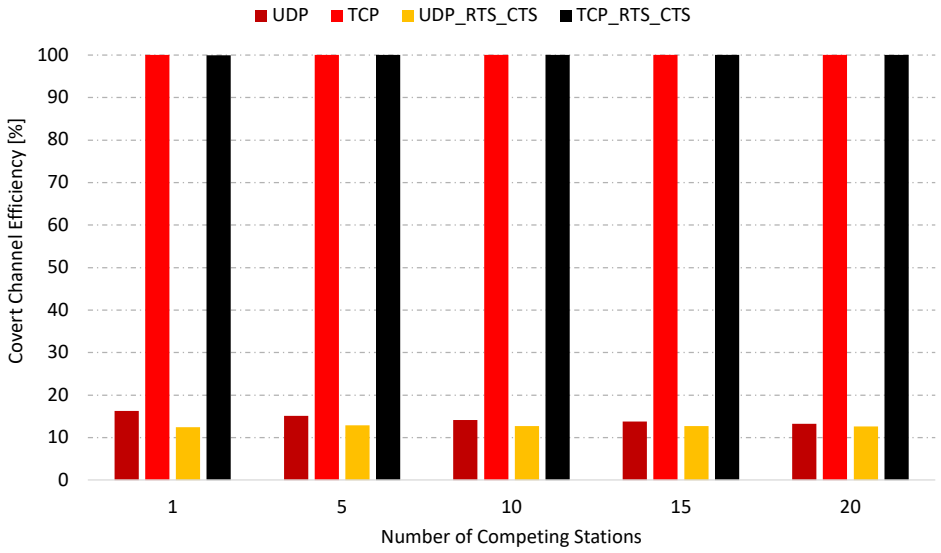


Figure 6.5: Comparing efficiency of covert communication using UDP and TCP traffic, both with and without RTS/CTS, under different station densities

Figure 6.6 illustrates the delay in the covert channel as a function of the number of competing stations, under various conditions. The UDP exhibits a non-linear trend as the number of competing stations increases. With a single additional station, delay is moderate (approximately 2024 ms) due to minimal contention. As the number of stations rises to 5 and 10, delay increases sharply (3679 - 3649 ms) because the shared channel becomes congested, causing packets to wait longer in the MAC layer and increasing the likelihood of collisions. Interestingly, when 15 and 20 stations are active, the delay decreases to 1155 ms and 620 ms, respectively. This occurs because the channel is heavily saturated, and many UDP frames are dropped quickly rather than queued, so the measured delay of successfully received frames is reduced. In contrast, TCP exhibits consistently low delay across all station densities, which, when compared to the UDP is almost unnoticeable in the plot. Its congestion control and flow management mechanisms prevent the channel from being overwhelmed, maintaining stable transmission times even as network load increases.

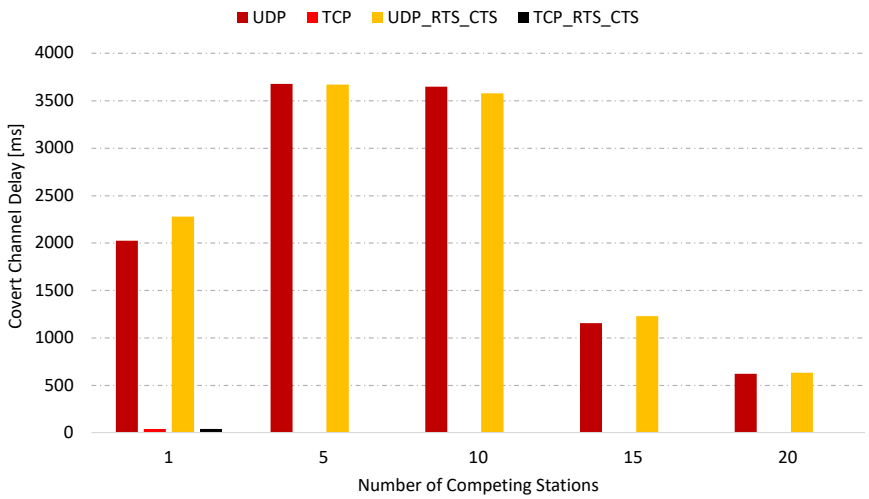


Figure 6.6: Comparing delay of covert communication using UDP and TCP traffic, both with and without RTS/CTS, under different station densities

Figure 6.7 illustrates the jitter observed in the covert channel, defined as the variation in delay between successive covert message transmissions. For UDP without RTS/CTS, jitter increases steadily as the number of stations grows, starting from values around 332 ms with one competing station and reaching 5845 ms for 20 stations. This occurs because UDP is connectionless and does not implement congestion control, so increased station numbers lead to more collisions and variable packet delays, resulting in higher jitter. When using TCP without RTS/CTS, jitter starts at 970 ms for one station and rises slightly to 1180 ms for five stations, but then decreases to 729 ms for 20 stations. The TCP congestion control and reliable delivery mechanisms smooth packet transmission, and as more stations compete for the medium, TCP adapts its transmission rate, resulting in a more uniform packet spacing and a lower jitter under heavy load. Introducing RTS/CTS for UDP slightly increases jitter at higher station counts, reaching 6483 ms for 20 stations, due to the added frame exchange overhead, which introduces additional variability in packet arrival times. For TCP with RTS/CTS, jitter remains relatively stable across all station counts because TCP flow control dominates, and the RTS/CTS overhead has only minor effects.

6.4.3 Increasing the offered load of the regular stations

In this scenario, we progressively increase the network density and the offered load while the covert station maintains a constant transmission rate at the saturation level. The purpose of this experiment is to assess how the increase in background traffic (using the UDP transport protocol), introduced by the addition of new stations, affects the performance of the covert channel.

As shown in Figure 6.8, the simulation examines the impact of adding 1, 5, 10, 15, and 20 regular stations to the network. When external traffic is minimal, at just 1 Mbps, the covert channel demonstrates stable performance at approximately 3.6 kbps, regardless of the number of contending stations. This stability occurs because the traffic generated by these stations is insufficient to interfere with the covert station at saturation. However, as the offered load from these external stations increases, the effects of contention become

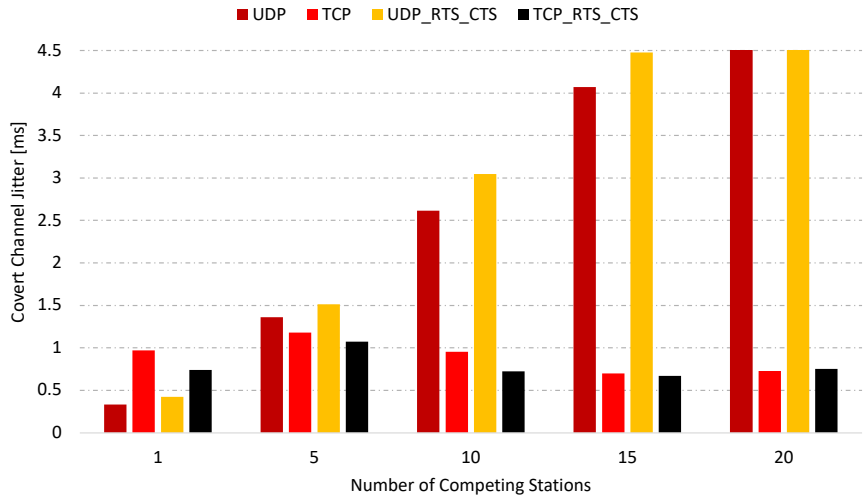


Figure 6.7: Comparing jitter of covert communication using UDP and TCP traffic, both with and without RTS/CTS, under different station densities

more evident, leading to a noticeable drop in covert channel throughput. This decline is directly attributed to more frequent competition for channel access. The minimum point to reach saturation varies based on the combination of the number of stations and the offered load. For example, one extra station requires an offered load of 20 Mbps, whereas having up to 20 stations necessitates 40 Mbps to establish the minimum value that the covert channel can achieve, which drops to 153 bps with 20 stations.

Generally, a higher number of stations also requires a higher traffic load to effectively restrict the covert channel.

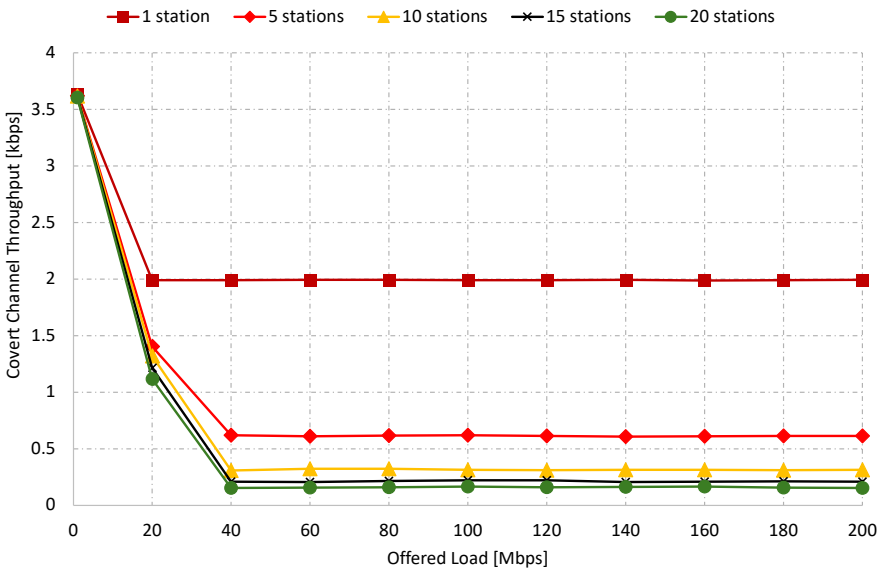


Figure 6.8: Impact of increasing offered load of the competing stations on covert channel throughput

The efficiency of the covert channel, measured as the percentage of frames delivered successfully, is highly sensitive to increasing network load, as illustrated in Figure 6.9.

This figure shows the ratio of frames received by the AP to those transmitted, revealing a significant decrease in efficiency as the volume of traffic increases. At a low offered load of 1 Mbps, the channel efficiency remains constant, which aligns with the findings in Figure 6.8, where the throughput also stabilizes at this level of traffic. However, a closer examination reveals that only approximately 25% of the transmitted frames are successfully received by the AP, indicating that three-quarters of the frames are lost, likely due to buffer overflows, contention, or medium access delays. At 20 Mbps, the covert channel reaches its saturation point in the presence of a single external station, as also evidenced in the throughput behavior. For higher numbers of competing stations, this saturation occurs at around 40 Mbps, after which the efficiency level remains constant. Nonetheless, the overall trend remains consistent: the higher the number of competing stations, the lower the covert channel efficiency. This decline can be attributed to increased collisions, longer queueing delays at the AP, and overall contention for medium access, all of which reduce the likelihood of successful frame delivery.

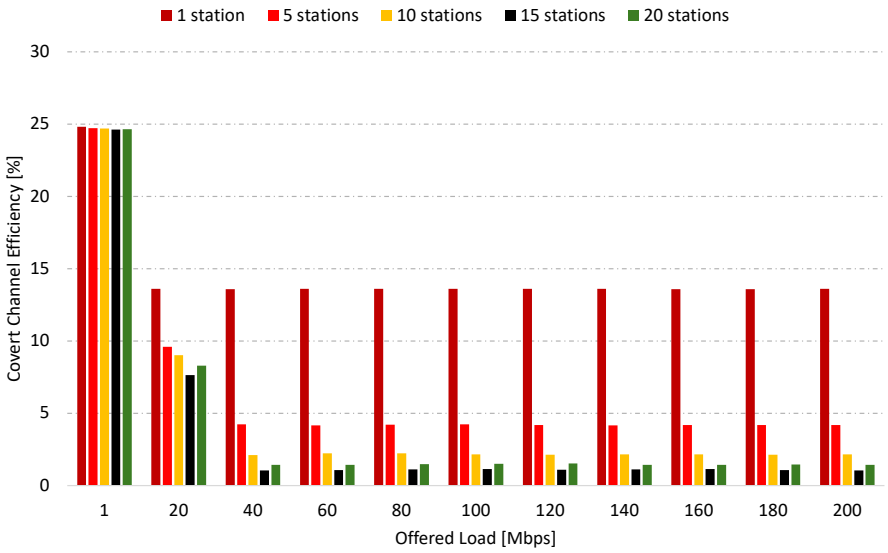


Figure 6.9: Impact of increasing offered load of the competing stations on covert channel efficiency

Figure 6.10 illustrates the behavior of covert channel delay as a function of offered load and the number of competing stations. The observed trend shows an equal delay at 1 Mbps, where the offered load is insufficient to cause significant frame variations. This results in constant throughput, and the efficiency is reflected in the delay. At 20 Mbps, the delay stabilizes at around 1500 ms when there is one competing station, as this is the saturation point for the covert channel throughput. Similarly, at 40 Mbps, the delay also stabilizes, indicating saturation points against two to five competing stations.

The Figure 6.11 depicts the delay variation. The jitter remains very low, below 1 ms at 1 Mbps, and even at 20 Mbps, with up to five competing stations, the jitter does not exceed 1.5 ms at any offered load. The saturation point occurs at 40 Mbps (with 5 to 20 stations), where the jitter stabilizes across the offered load.

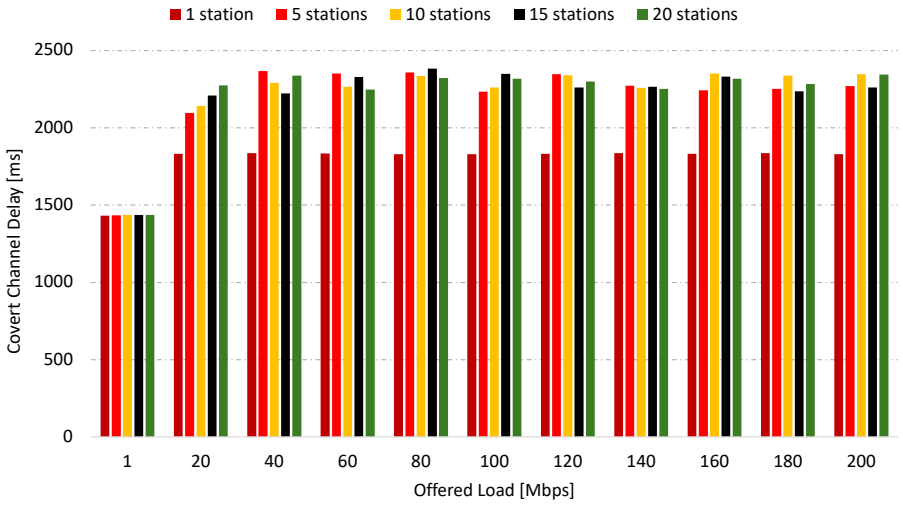


Figure 6.10: Impact of increasing offered load of the competing stations on covert channel delay

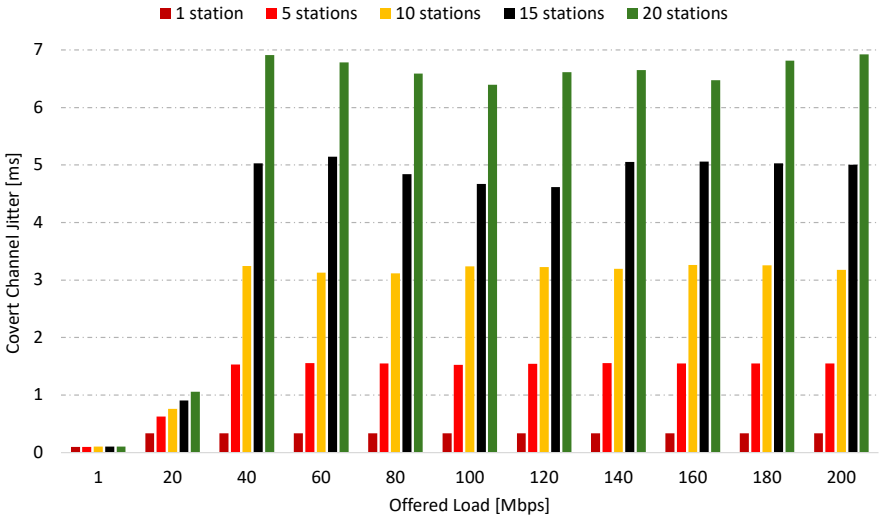


Figure 6.11: Impact of increasing offered load of the competing stations on covert channel jitter

6.4.4 Increasing offered load of the covert STA

In this experiment, we examined the behavior of the covert channel as the traffic volume of the covert station increased, while a competing station maintained a constant offered load at the saturation level (using the UDP transport protocol). We gradually increased the offered load of the covert station from 1 to 200 Mbps, progressively stressing the network until saturation was achieved.

Figure 6.12 presents the covert channel throughput as a function of the offered load for varying numbers of competing stations. The maximum observed throughput reaches approximately 2 kbps in the scenario with only one external station, consistent with the results shown in Figure 6.8. As the number of competing stations increases, the throughput drops significantly. With five stations, the throughput falls below 1 kbps, and beyond ten stations, the differences between configurations become marginal. This indicates a

saturation point where the covert channel throughput stabilizes due to increased contention and limited opportunities for covert transmissions.

Across all configurations, the throughput quickly reaches its peak at an offered load of approximately 20 Mbps. Beyond this threshold, additional load does not lead to further throughput gains, suggesting that 20 Mbps represents the saturation point of the covert channel in the presence of external interference. This value can be considered the optimal load offered for this scenario, as increasing it further does not have a performance benefit.

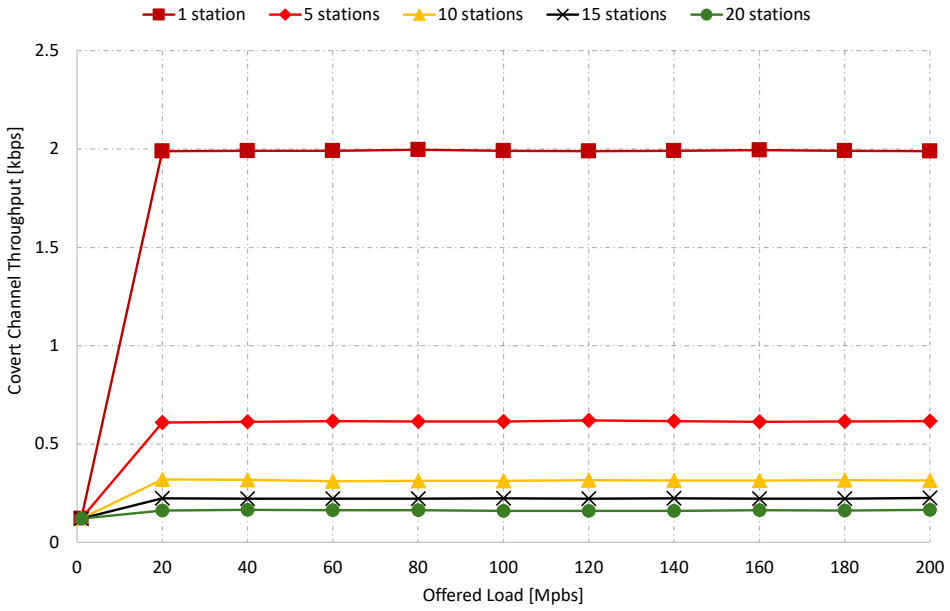


Figure 6.12: Impact of increasing offered load of the covert stations on covert channel throughput

Figure 6.13 illustrates the efficiency of the covert channel. At a low offered load of 1 Mbps, the network experiences minimal contention, resulting in an efficiency close to 100% across all station counts. This outcome is expected because the low transmission rate does not saturate the channel.

However, as the offered load increases, a significant decrease in efficiency is observed. At 20 Mbps, which is the identified saturation point, additional traffic does not enhance throughput, and the AP receives frames at a constant rate. Meanwhile, the transmission rate continues to increase, resulting in a growing disparity between the number of transmitted and received frames. This imbalance causes efficiency to drop sharply.

The decline in efficiency is even more pronounced with a higher number of competing stations. Regardless of the station count, the efficiency consistently falls below 10% on high offered loads. This indicates that the covert channel becomes overwhelmed by the high volume of traffic generated, resulting in increased contention, collisions, and frame losses.

Figure 6.14 illustrates how the delay in the covert channel varies with the offered load and the number of competing stations. A clear trend emerges: as the offered load increases from 1 Mbps to 20 Mbps, the delay rises in all scenarios, indicating the onset of channel saturation. During this phase, contention becomes significant, resulting in increased queuing delays due to collisions and backoff procedures.

After reaching the saturation point, the delay begins to decrease as the offered load

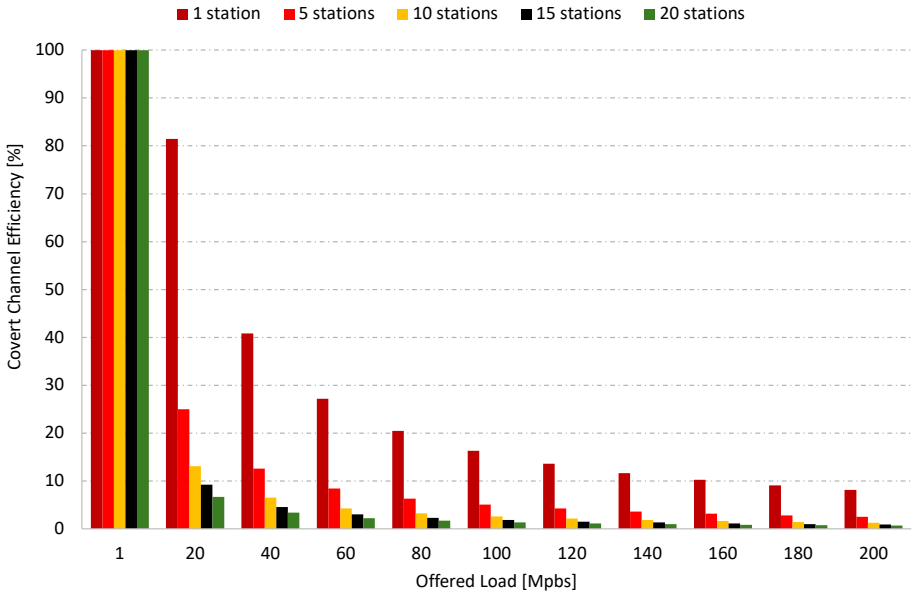


Figure 6.13: Impact of increasing offered load of the covert stations on covert channel efficiency

continues to rise. This behavior occurs because, once the buffers at the sender or access point become full, more frames are dropped due to expiration, resulting in fewer frames being transmitted and a reduction in experienced delay over the channel. From 120 Mbps onward, the delay stabilizes across all station configurations, suggesting that the channel has reached a point where additional offered load does not significantly increase covert message delay.

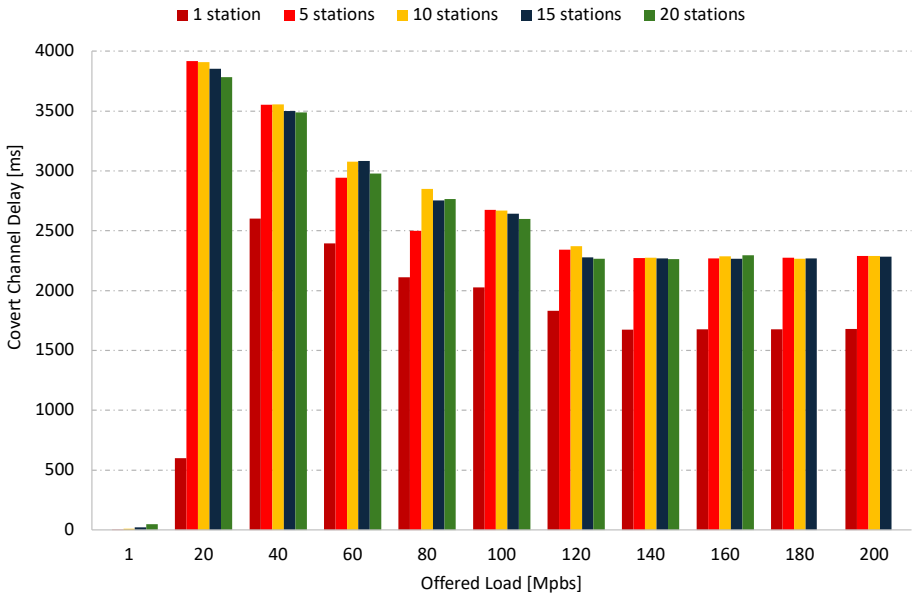


Figure 6.14: Impact of increasing offered load of the covert stations on covert channel delay

Figure 6.15 demonstrates the jitter experienced by the covert channel under varying

offered loads and different numbers of competing stations. At a low load of 1 Mbps, the jitter is already significant; for 15 and 20 stations, it exceeds 5 ms and 7 ms, respectively. This highlights the early impact of channel contention. As the offered load increases, the jitter either stabilizes or rises slightly, particularly for 15 and 20 stations. This suggests that, in high contention scenarios, channel access becomes more unpredictable, resulting in irregular delays between consecutive covert transmissions.

In contrast, with fewer competing stations (1 or 5), the jitter remains relatively low (below 2 ms) and increases only moderately with the load. This indicates more consistent timing for covert messages and less contention. The findings imply that jitter is primarily influenced by the number of competing stations rather than the offered load alone. High levels of contention lead to variable backoff and access delays, which further amplify jitter, even when throughput or delay appears stable or saturated.

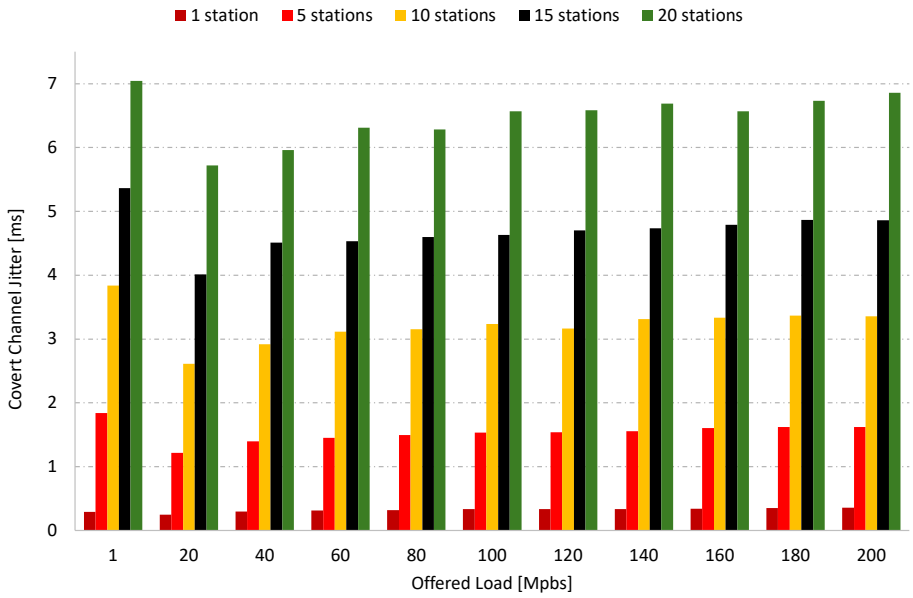


Figure 6.15: Impact of increasing offered load of the covert stations on covert channel jitter

To quantify the amount of covert data that can be embedded relative to the regular channel capacity, we calculate the embedding ratio. This ratio is defined as the percentage of covert throughput compared to regular throughput, both measured in bps, as shown in Equation 6.5. This metric offers insight into the proportion of covert information that can be transmitted over the available bandwidth of the regular channel.

$$\text{Embedding Ratio} = \left(\frac{\text{Covert throughput}}{\text{Regular throughput}} \right) \times 100[\%] \tag{6.5}$$

Based on the computed values presented in Figure 6.16, we observe that despite increasing contention, the embedded ratio remains nearly constant at approximately 0.01188%. This indicates that the covert channel scales proportionally with the available legitimate bandwidth, which is one bit per data frame.

Although the offered load ranges from 1 Mbps to 200 Mbps, the throughput reaches saturation around 20 Mbps across all scenarios. This means that the same amount of data is effectively delivered beyond this point. This saturation behavior explains why

both the covert and regular throughputs flatten out, leading to a stable ratio between them.

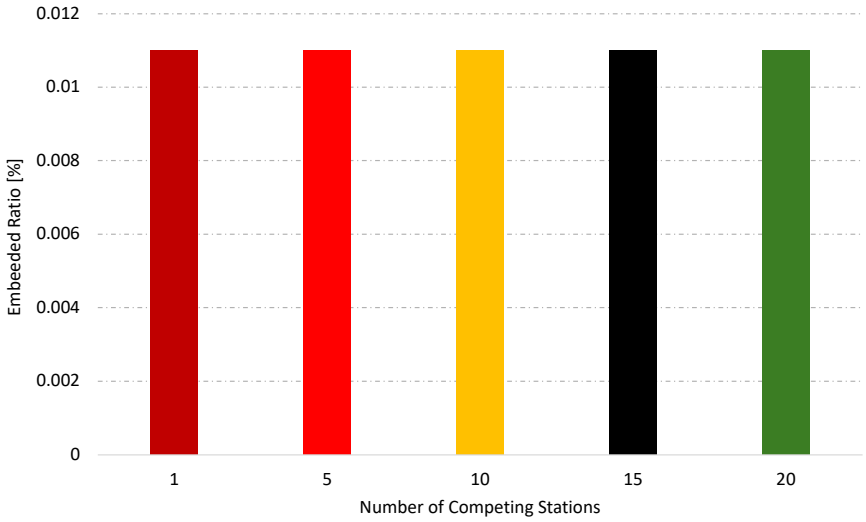


Figure 6.16: Embedded ratio as a function of the number of competing stations

6.4.5 Impact of changing MCS index

In this experimental scenario, we adjusted the MCS index of the covert station while maintaining a constant presence of 10 regular stations that were generating traffic at saturation. The objective of this experiment is to observe how variations in the MCS index affect the covert channel.

Figure 6.17 illustrates how the transport protocol and the MCS index impact the throughput of the covert channel. At lower MCS values (e.g., 1–3), the available physical data rate is limited, which directly constrains the maximum throughput that can be achieved. The lower modulation schemes used at these MCS levels also introduce higher latency and a greater probability of frame errors, which for TCP translates into frequent retransmissions and very low throughput. UDP performs better under the same conditions because it continuously sends packets without waiting for acknowledgments, but its throughput is still limited by the modest data rate at the physical layer. As the MCS index increases, higher-order modulation and coding schemes are employed, improving spectral efficiency and reducing the relative impact of errors. Both TCP and UDP therefore achieve higher throughput, with TCP gradually closing the gap to UDP as retransmissions become less frequent.

Figure 6.18 compares the efficiency of covert channels under TCP and UDP traffic across different MCS indices. TCP consistently achieves over 90% efficiency due to its robust delivery guarantees, making it highly effective for covert transmissions when successful delivery is essential. Conversely, UDP exhibits lower efficiency, especially at lower MCS levels, as it lacks reliability mechanisms to recover lost or dropped frames. While UDP's efficiency improves slightly at higher MCS levels (due to better channel conditions), it remains significantly below that of TCP.

In Figure 6.19, we analyze the delay behavior of covert channels across different MCS indices. For UDP, the delay remains consistently high, fluctuating around 800 ms regardless of the MCS level. Although increasing the MCS slightly reduces the delay due

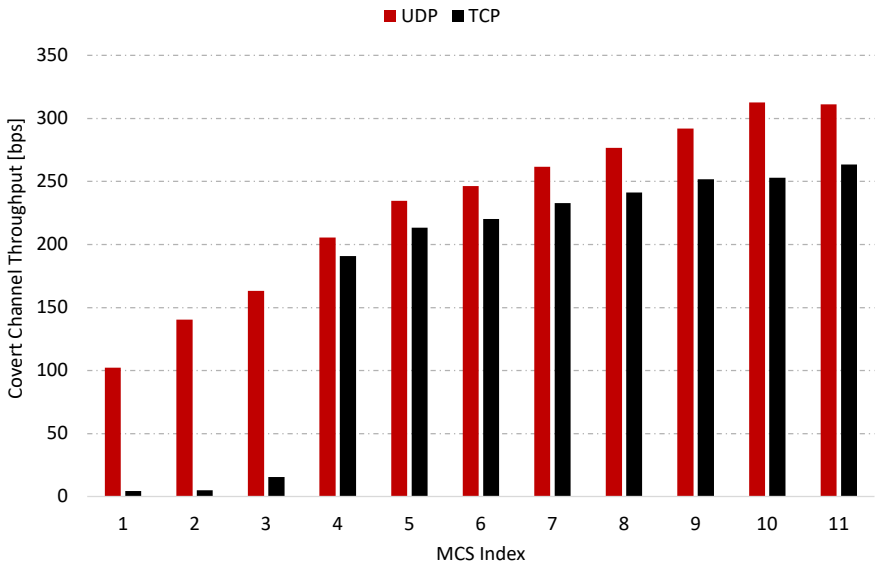


Figure 6.17: Covert channel throughput variation across different MCS indices under UDP and TCP traffic

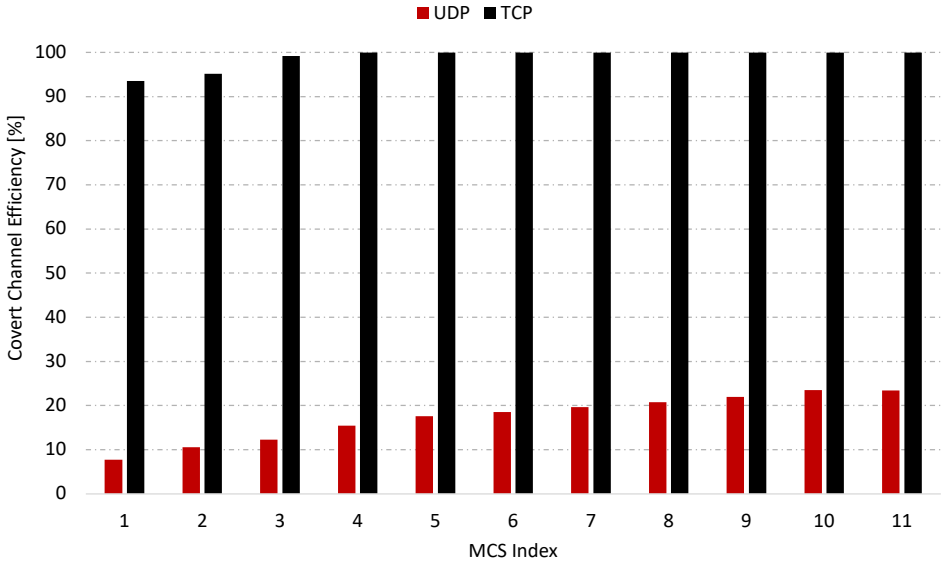


Figure 6.18: Covert channel efficiency variation across different MCS indices under UDP and TCP traffic

to improved PHY parameters that enable faster transmission, UDP lacks flow control mechanisms, limiting its adaptability.

In contrast, TCP delay stays below 100 ms and demonstrates a strong correlation with increasing MCS. Notably, starting from MCS 4, where the throughput begins to rise, TCP delay stabilizes at very low values. This behavior reflects the effectiveness of TCP's flow control and acknowledgment mechanisms, which allow it to quickly adapt and maintain low latency under higher data rates, a feature that UDP lacks.

Regarding covert channel jitter, as illustrated in Figure 6.20 , this metric consistently shows a downward trend as the MCS index increases, indicating reduced variability in

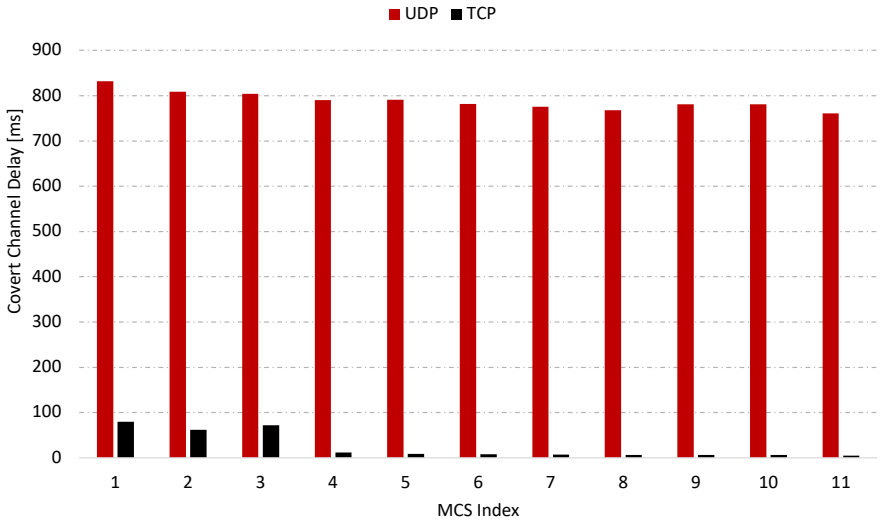


Figure 6.19: Covert channel delay variation across different MCS indices under UDP traffic

inter-frame transmission intervals. Initially, UDP jitter is high (from MCS 1 to 3) but gradually decreases due to faster and more regular frame delivery enabled by higher modulation rates, although the absolute delay remains high. Conversely, TCP jitter is consistently lower than that of UDP and stabilizes quickly. From MCS 4 onward, TCP jitter remains below 1 ms, demonstrating TCP's adaptability to channel conditions, which aligns with the behavior observed in delay from MCS 4.

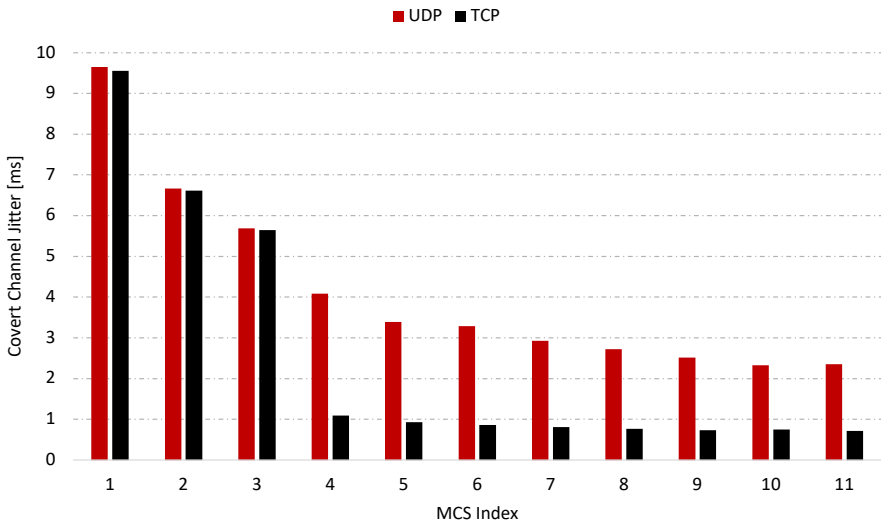


Figure 6.20: Covert channel jitter variation across different MCS indices under TCP traffic

6.5 Discussion of results

StegoBackoff establishes a highly transparent covert channel by encoding a hidden bit in the backoff slot before each data frame transmission. This design makes the channel imperceptible, as the slot adjustment is conditional and indistinguishable from normal contention. StegoBackoff embeds one covert bit per data frame, achieving a maximum throughput of 4.7 kbps. In our performance evaluation, we analyzed the channel under varying frame sizes, traffic loads, external interference, and MCS levels. The key findings are summarized as follows:

- Smaller frame sizes yield higher covert throughput, while larger payloads decrease throughput due to increased transmission time.
- UDP traffic generates a higher volume but is less efficient than TCP traffic, resulting in fewer successful frame deliveries at the AP side. In contrast, TCP provides more reliable delivery, albeit at slightly reduced data rates. The choice of protocol depends on whether throughput or reliability is prioritized.
- An increase in the number of competing stations and their respective offered loads leads to a reduction in covert throughput. However, the number of stations has a more pronounced negative impact than load intensity alone.
- Increased traffic from covert stations raises the probability and frequency of transmissions, which helps counterbalance the effects of contention and improves covert throughput.
- The RTS/CTS mechanism significantly enhances throughput in less contended environments, particularly for TCP traffic. Higher MCS indices also contribute to faster data rates and reduced airtime per transmission.
- Overall, the performance of the covert channel resembles that of a regular network, since covert data are encoded within standard data frame transmissions.

7 Covert channel StegoHybrid

7.1 StegoHybrid operation

This chapter introduces a novel steganographic approach: a covert channel that leverages frame aggregation (A-MSDU and A-MPDU) to hide secret messages using three distinct vectors. Each vector is considered an independent subchannel, and each subchannel is discussed in a dedicated section, which explains both the underlying concept and the operational mechanism of the respective covert channel. Subsequently, the metrics and results are presented, first considering each component individually and then examining their overall combination, to analyze the contribution of each subchannel to the achieved results.

7.1.1 First subchannel

The first subchannel, which embeds the covert message, utilizes the Duration/ID field of each MPDU header (Figure 2.3), which is part of the A-MPDU. The first proposal to use the Duration/ID field is presented in the work [86], where the 16-bit duration field is modified by replacing the last three least LSBs with the secret message, which is sent in plain text. Another proposition is described in work [87], which introduces a covert channel based on EDCA. In this case, the authors take the 2-bit QoS class (AC_BE, AC_BK, AC_VI, and AC_VO), XOR it with a 2-bit secret message, and insert the resulting encoded value into the last two LSBs of the duration field.

The logic of this covert channel is outlined in Algorithms 7 and 8, which describe the operations performed at the sender and receiver sides, respectively. The encoding process embeds a 3-bit secret message by replacing the last three LSBs of the duration field. To enhance the covertness of the communication, the sender applies a simple transformation to these bits: the three LSBs are either rotated to the left, rotated to the right, or left unchanged. The decision to rotate the bits depends on the backoff counter, specifically on the remainder of its value modulo three. If the result is zero, the bits are rotated one position to the right; if it is one, they are rotated one position to the left; and if it is two, the bits remain unchanged. The decoding process reverses these operations to recover the original message. The three LSBs of the duration field are extracted and grouped into a temporary variable. Based on the same backoff counter value, the inverse transformation is applied: for a remainder of zero, the bits are rotated left, for a remainder of one, they are rotated right, and for a remainder of two, no modification is needed. The resulting bits are then separated and reassembled into the original 3-bit secret message, allowing the receiver to reconstruct the hidden information.

7.1.2 Second subchannel

The second subchannel leverages the TXOP Duration Requested field, as detailed in Chapter 2.2.6 and illustrated in Figure 2.12. In the context of frame aggregation, the sender embeds the covert message into the 8-bit TXOP Duration Requested field of each MPDU, except the final one. The last MPDU carries a TXOP value of zero to indicate the cancellation of previous requests and signal an apparent change in the queue state. The whole encoding and decoding process is presented in Algorithms 9 and 10, respectively: the encoding algorithm inserts an 8-bit covert message into the TXOP

Algorithm 7 Encoding a 3-bit message in the Duration/ID field

Input: m — 3-bit secret message to encode
Input: d — Duration/ID field in MAC header
Input: $r \leftarrow$ backoff mod 3
procedure EncodeBitInDuration(m, d, r)
  $d[13] \leftarrow m[0]$
  $d[14] \leftarrow m[1]$
  $d[15] \leftarrow m[2]$
  $d_{\text{LSB}} \leftarrow (d[15] \ll 2) \mid (d[14] \ll 1) \mid d[13]$
  if $r = 0$ **then**
    $d'_{\text{LSB}} \leftarrow (d_{\text{LSB}} \gg 1) \mid ((d_{\text{LSB}} \,\&\, 1) \ll 2)$ ▷ Right shift
  else if $r = 1$ **then**
    $d'_{\text{LSB}} \leftarrow ((d_{\text{LSB}} \ll 1) \;\&\, 0b111) \mid (d_{\text{LSB}} \gg 2)$ ▷ Left shift
  else
    $d'_{\text{LSB}} \leftarrow d_{\text{LSB}}$
  end if
  $d[13] \leftarrow d'_{\text{LSB}} \,\&\, 0b001$
  $d[14] \leftarrow (d'_{\text{LSB}} \gg 1) \,\&\, 0b001$
  $d[15] \leftarrow (d'_{\text{LSB}} \gg 2) \,\&\, 0b001$
end procedure

Algorithm 8 Decoding a 3-bit message from the Duration/ID field

Input: m — 3-bit secret message to encode
Input: d — Duration/ID field in MAC header
Input: $r \leftarrow$ backoff mod 3
procedure DecodeBitFromDuration(m, d, r)
  $d'_{\text{LSB}} \leftarrow (d[15] \ll 2) \mid (d[14] \ll 1) \mid d[13]$
  if $r = 0$ **then**
    $d_{\text{LSB}} \leftarrow ((d'_{\text{LSB}} \ll 1) \,\&\, 0b111) \mid (d'_{\text{LSB}} \gg 2)$ ▷ Reverse right shift
  else if $r = 1$ **then**
    $d_{\text{LSB}} \leftarrow (d'_{\text{LSB}} \gg 1) \mid ((d'_{\text{LSB}} \,\&\, 1) \ll 2)$ ▷ Reverse left shift
  else
    $d_{\text{LSB}} \leftarrow d'_{\text{LSB}}$
  end if
  $m[0] \leftarrow d_{\text{LSB}} \,\&, 0b001$
  $m[1] \leftarrow (d_{\text{LSB}} \gg 1) \,\&\, 0b001$
  $m[2] \leftarrow (d_{\text{LSB}} \gg 2) \,\&\, 0b001$
  return m
end procedure

Duration Requested field of each MPDU within an A-MPDU. The procedure iterates over the entire A-MPDU, whose length is explicitly determined at the beginning to ensure consistent interpretation by both the sender and the receiver. For every MPDU except the last one, the covert message is combined with the last eight bits of the sender and receiver MAC addresses using the XOR operation, and the result is written into the TXOP Duration Requested field. The last MPDU in each A-MPDU is reserved for resetting purposes, and its TXOP Duration Requested field is therefore set to zero. An

exception occurs when the A-MPDU consists of only a single MPDU that itself contains multiple A-MSDUs. In this case, the reset is not applied and the covert message is still embedded. This design ensures that covert communication is possible not only when A-MPDU aggregation is employed, but also when aggregation is limited to A-MSDU within a single MPDU.

The decoding algorithm performs the inverse operation at the receiver side. The length of the A-MPDU is again determined to ensure correct interpretation. For every MPDU except the last one, or for the single-MPDU case, the covert message is recovered by XORing the value of the TXOP Duration Requested field with the last eight bits of the sender and receiver MAC addresses. When the A-MPDU contains more than one MPDU, the last one is ignored, since it is reserved as a reset.

Algorithm 9 Encoding covert data in the TXOP Duration Requested field across an A-MPDU

Input: $AMPDU$ — sequence of MPDUs
Input: msg — 8-bit covert message
Input: ms — last 8 bits of sender's MAC address
Input: md — last 8 bits of receiver's MAC address
procedure EncodeTxop($AMPDU, msg, ms, md$)
  $L \leftarrow$ length of $AMPDU$
  for $i \leftarrow 1$ to L **do**
    if $L = 1$ **or** $i < L$ **then**
      $AMPDU[i].txopd \leftarrow msg \oplus ms \oplus md$
    else
      $AMPDU[i].txopd \leftarrow 0$
    end if
  end for
  return $AMPDU$
end procedure

7.1.3 Third subchannel

The third strategy for hiding covert messages during wireless transmission involves the use of frame aggregation schemes. In this approach, the sender leverages multilevel aggregation by combining both A-MSDU and A-MPDU structures. A specific A-MSDU-to-A-MPDU configuration represents a particular bit sequence, which is the content of the covert message.

Given the frame size F, the maximum allowed A-MSDU size as S_{AMSDU}, and the total number of MSDUs within A-MSDU as L_{AMSDU}. Also, considering the maximum allowed A-MPDU size as S_{AMPDU}, and the total number of MPDUs within A-MPDU as L_{MPDU}. The sender can construct a covert message by mapping a specific bit sequence to a valid aggregation layout (M, N), where:

- M is the number of MSDUs per A-MSDU (i.e., how many frames are grouped into one A-MSDU)
- N is the number of MPDUs per A-MPDU (i.e., how many MPDUs are grouped into the A-MPDU), where a single MPDU contains one A-MSDU.

Algorithm 10 Decoding covert data from the TXOP Duration Requested field across an A-MPDU

```
Input: AMPDU — sequence of MPDUs
Input: ms — last 8 bits of sender's MAC address
Input: md — last 8 bits of receiver's MAC address
procedure DecodeTxop(AMPDU, ms, md)
    L ← length of AMPDU
    msg ← [ ]
    for i ← 1 to L do
        if L = 1 or i < L then
            m ← AMPDU[i].txopd ⊕ ms ⊕ md
            append m to msg
        else
            stop decoding
        end if
    end for
    return msg
end procedure
```

The goal is to create an (M, N) aggregation layout to encode a secret message while respecting the constraints regarding the aggregation of A-MSDU and A-MPDU frames in the Equations 7.1.

$$\mathcal{S} = \{(M, N) \,|\, 2 \leq M \leq M_{\max},\ 1 \leq N \leq N_{\max}(M)\} \tag{7.1}$$

The set $\mathcal{S}$ defines all valid combinations of (M, N) that can be used to encode covert information without violating the constraints. The $M_{\max}$ is the maximum number of MSDUs that can be compacted into a single MPDU (A-MSDU aggregation level), limited by the size of the A-MSDU and the length of the MSDU allowed by 802.11 standard, and $M_{\max}$ is presented by Equation 7.2. The $N_{\max}$ is the maximum number of MPDUs that can be included in an A-MPDU, given a specific M, limited by the total A-MPDU and MPDU size allowed by 802.11 standard, and $N_{\max}$ is expressed in the Equation 7.3.

$$M_{\max} = \min\left(\left\lfloor \frac{S_{AMSDU}}{F} \right\rfloor, L_{MSDU}\right) \tag{7.2}$$

$$N_{\max}(M) = \min\left(\left\lfloor \frac{S_{AMPDU}}{M \cdot F} \right\rfloor, L_{MPDU}\right) \tag{7.3}$$

Once $M_{\max}$ is computed, the sender can flexibly select a value $M \in [2, M_{\max}]$ to determine how many MSDUs are grouped into each A-MSDU. For example, considering the case where $M_{\max} = 4$, meaning that up to 4 MSDUs can be transmitted. One valid aggregation layout is to place all four MSDUs into a single A-MSDU and encapsulate it within a single MPDU. This results in the aggregation configuration $(M = 4, N = 1)$, where a single MPDU carries one A-MSDU, which in turn contains 4 MSDUs. Alternatively, the sender may choose to split the 4 MSDUs into two A-MSDUs, each composed of 2 MSDUs. Then, these two A-MSDUs are each encapsulated in separate MPDUs, which produces the aggregation layout $(M = 2, N = 2)$, where two MPDUs are used, and each carries an A-MSDU with two MSDUs. Such a strategy allows the creation of

multiple valid (M, N) combinations that the sender can choose from to encode a secret message. For instance, if $M_{\max} = 4$, the sender has at least two alternatives about how to aggregate the frames, it could be for example using the layout (4,1) to encode secret bit 0 or use the layout (2,2) to encode secret bit 1. The covert channel capacity denoted by B, is defined as the number of bits that can be encoded per one A-MPDU transmission, which is calculated according to Equation 7.4:

$$B = \lfloor \log_2(k) \rfloor \, [bits] \tag{7.4}$$

where $k = |\mathcal{S}|$ the length of set $\mathcal{S}$ (Equation 7.1), representing the total number of unique and valid combinations of aggregation layout (M, N) available in $\mathcal{S}$ under the given constraints. Each layout in $\mathcal{S}$ can be mapped to a unique bit pattern of length B. That means that for each A-MPDU transmission, it is possible to encode up to B bits of covert information by selecting an appropriate (M, N) combination from the set. This enables a one-to-one correspondence between aggregation layouts and the covert message represented by a particular bit sequence of length B.

To illustrate how this encoding scheme operates in practice, consider the following configuration (for simplicity, F denotes the frame size, which includes the headers, padding, and the frame payload):

- MSDU size: $F = 512$ bytes
- Maximum A-MSDU size: $S_{AMSDU} = 2048$ bytes
- Maximum A-MPDU size: $S_{AMPDU} = 2048$ bytes

From these values, we can compute the maximum number of MSDUs per A-MSDU as per Equation 7.2:

$$M_{\max} = \left\lfloor \frac{2048}{512} \right\rfloor = 4 \quad \Rightarrow \quad M \in \{2, 3, 4\}$$

Next, we compute $N_{\max}(M)$, the number of MPDUs that can be created depending on how the A-MSDUs are aggregated as per Equation 7.3:

$$\begin{aligned}
M = 2 \Rightarrow N_{\max}(2) &= \left\lfloor \frac{2048}{2 \cdot 512} \right\rfloor = 2 \quad \Rightarrow N \in \{1, 2\} \\
M = 3 \Rightarrow N_{\max}(3) &= \left\lfloor \frac{2048}{3 \cdot 512} \right\rfloor = 1 \quad \Rightarrow N \in \{1\} \\
M = 4 \Rightarrow N_{\max}(4) &= \left\lfloor \frac{2048}{4 \cdot 512} \right\rfloor = 1 \quad \Rightarrow N \in \{1\}
\end{aligned}$$

This leads to the construction of the valid aggregation layout set $\mathcal{S}$ as per Equation 7.1:

$$\mathcal{S} = \{(2, 1), (2, 2), (3, 1), (4, 1)\}$$

Each pair $(M, N) \in \mathcal{S}$ represents a valid combination of A-MSDU and A-MPDU structures under each frame size constraint. This means the sender has four distinct aggregation strategies to encode covert information. The possible interpretations of each layout are:

- $(2, 1)$: A single MPDU carrying one A-MSDU composed of 2 MSDUs.
- $(2, 2)$: An A-MPDU with 2 MPDUs, each containing an A-MSDU of 2 MSDUs.
- $(3, 1)$: A single MPDU carrying one A-MSDU of 3 MSDUs. The remaining 1 MSDU (assuming a total of 4) would need to be transmitted in a separate aggregation.
- $(4, 1)$: A single MPDU carrying one A-MSDU composed of all 4 MSDUs.

The total number of available layouts is $k = 4$, and the maximum number of bits that can be encoded per A-MPDU transmission, according to Equation 7.4, is $B = \lfloor \log_2(4) \rfloor = 2$ bits. Each layout in $\mathcal{S}$ can now be mapped to a unique 2-bit symbol as presented in Table 7.1.

Table 7.1: Illustrative example showing how the sender encodes secret messages by employing an (M, N) aggregation layout, resulting in a unique combination of B bits

Position (i)	**M**	**N**	**Secret Message**
0	2	1	00
1	2	2	01
2	3	1	10
3	4	1	11

When the total number of valid aggregation layouts $|\mathcal{S}| = k$ exceeds the number of required combinations for encoding a B-bit covert message, the remaining aggregation layouts are excluded from the encoding process (this combination is permitted for regular transmission but conveys no covert information). For example, if $|\mathcal{S}| = 6$ and $B = 2$, then only 2^B layouts are needed for encoding (4 valid combinations). The remaining two layouts do not convey any covert messages. The sender and receiver can agree on which specific aggregation layouts to use and which to ignore. The constraint of the useful layout of the aggregation within the set $\mathcal{S}$ is expressed in Equation 7.5:

$$2^B \leq k = |\mathcal{S}| \tag{7.5}$$

This uncertainty complicates steganalysis, as an external observer cannot easily distinguish between legitimate variations introduced by the sender and those that are part of the covert encoding. The entire process that the sender uses to encode the covert channel is implemented using Algorithm 11, and the decoding scheme is in Algorithm 12.

7.2 StegoHybrid properties and deployment scenarios

In general, most of the covert channels presented in Chapter 3 involve implementations where secret messages are embedded into a single feature, location, or mechanism. This approach introduces a significant vulnerability: once the covert channel is detected, it can be easily disrupted or disabled, leading to a total loss of hidden communication. The novelty brought by StegoHybrid is a more sophisticated approach that distributes the covert message across three distinct protocol features or dimensions, thereby enhancing both resilience and resistance to steganalysis. If one covert channel is detected or blocked, the remaining channels can continue to operate independently. This strategy introduces

Algorithm 11 Covert Encoding via Aggregation Layout

Input: F — Frame size in bytes
Input: S_{AMSDU} — Maximum A-MSDU size
Input: S_{AMPDU} — Maximum A-MPDU size
Input: L_{MSDU} — MSDU limit per A-MSDU (protocol)
Input: L_{MPDU} — MPDU limit per A-MPDU (protocol)
Input: m — Covert message that corresponds to a (M, N) aggregation layout

procedure EncodeUsingAggr($F,\ S_{\text{AMSDU}},\ S_{\text{AMPDU}},\ L_{\text{MSDU}},\ L_{\text{MPDU}},\ m$)
  $M_{\max} \leftarrow \min\left(\left\lfloor \frac{S_{\text{AMSDU}}}{F} \right\rfloor,\ L_{\text{MSDU}}\right)$
  $\mathcal{S} \leftarrow \emptyset$
  for $M = 2$ to $M_{\max}$ **do**
    $N_{\max} \leftarrow \min\left(\left\lfloor \frac{S_{\text{AMPDU}}}{M \cdot F} \right\rfloor,\ L_{\text{MPDU}}\right)$
    for $N = 1$ to $N_{\max}$ **do**
      Append (M, N) to $\mathcal{S}$
    end for
  end for
  $k \leftarrow |\mathcal{S}|$
  $B \leftarrow \lfloor \log_2(k) \rfloor$
  $\text{Map}[i] = \mathcal{S}[i], \quad 0 \le i < 2^B$
  $(M, N) \leftarrow \texttt{Map.find}(m)$
  return (M, N)
end procedure

uncertainty for the adversary regarding the location of the remaining covert channels, complicating analysis and countermeasures.

In the StegoHybrid scheme, the three covert encoding mechanisms operate in a defined order as presented in the Figure 7.1. Because the scheme relies on A-MPDU, the first step is to embed covert bits into each MPDU individually. For every MPDU that forms part of the A-MPDU, the Duration/ID field is modified to carry a 3-bit fragment of the covert message according to Algorithm 7. At the same time, in each MPDU, the TXOP Duration Requested field is populated following Algorithm 9. This means that each MPDU contributes two separate opportunities for embedding hidden information, one in the MAC header and one in the QoS control field. Once the MPDU-level encoding is complete, the next step is to establish the aggregation layout of the A-MPDU. The aggregation structure, defined by the number of A-MSDUs per MPDU and the number of MPDUs per A-MPDU, is selected based on the mapping procedure in Algorithm 11. In this way, an additional portion of the covert message is carried by the choice of the (M, N) aggregation layout. On the receiver side, the decoding proceeds in the reverse direction: first, the aggregation layout is interpreted to recover its portion of the hidden message per Algorithm 12, then the TXOP Duration Requested fields are decoded (Algorithm 10), and finally the Duration/ID fields are read (Algorithm 8).

Regarding resistance to stegoanalysis and transparency, each subchannel within StegoHybrid has distinct characteristics:

- The resistance of the duration field encoding to steganalysis stems from its reliance on the inherent randomness of the backoff procedure, combined with the modulo-based transformation, neither of which is known by external parties. At the same

Algorithm 12 Covert Decoding via Aggregation Layout

Input: F — Frame size in bytes
Input: S_{AMSDU} — Maximum A-MSDU size
Input: S_{AMPDU} — Maximum A-MPDU size
Input: L_{MSDU} — MSDU limit per A-MSDU (protocol)
Input: L_{MPDU} — MPDU limit per A-MPDU (protocol)
Input: (M, N) — Observed aggregation layout

procedure DECODEUSINGAGGR($F,\ S_{\text{AMSDU}},\ S_{\text{AMPDU}},\ L_{\text{MSDU}},\ L_{\text{MPDU}},\ (M,N)$)
  $M_{\max} \leftarrow \min\left(\left\lfloor \frac{S_{\text{AMSDU}}}{F} \right\rfloor,\ L_{\text{MSDU}}\right)$
  $\mathcal{S} \leftarrow \emptyset$
  for $M_i = 2$ to $M_{\max}$ **do**
    $N_{\max} \leftarrow \min\left(\left\lfloor \frac{S_{\text{AMPDU}}}{M_i \cdot F} \right\rfloor,\ L_{\text{MPDU}}\right)$
    for $N_i = 1$ to $N_{\max}$ **do**
      Append (M_i, N_i) to $\mathcal{S}$
    end for
  end for
  $k \leftarrow |\mathcal{S}|$
  $B \leftarrow \lfloor \log_2(k) \rfloor$
  $\text{Map}[i] = \mathcal{S}[i], \quad 0 \le i < 2^B$
  $m \leftarrow$ `Map.find`$((M,N))$
  return m
end procedure

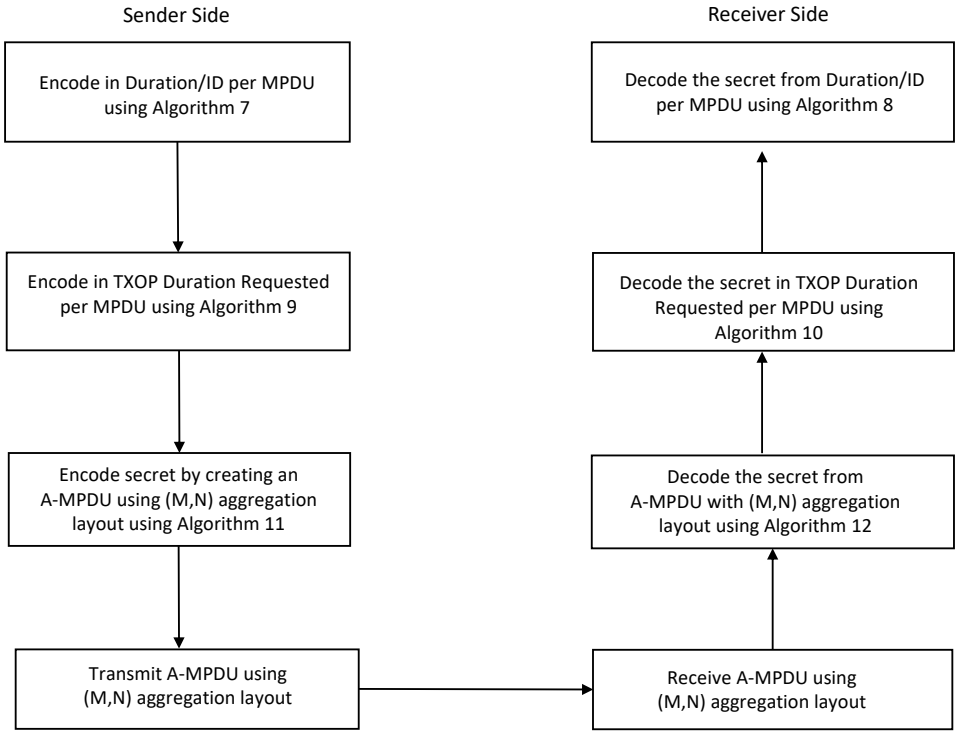


Figure 7.1: Encoding and decoding order of the StegoHybrid covert channel

time, modifying the duration field has a negligible impact on normal network op-

eration. Prior work [86] has demonstrated that the deviation introduced by this encoding is approximately 7 μs, a difference imperceptible under practical network conditions. Consequently, the covert channel maintains transparency, introducing no noticeable delays or disruptions.

- The covert message carried in the TXOP Duration Requested field is not transmitted in plain text. Instead, it is derived from the last 8 bits of the sender and receiver MAC addresses. While these addresses are observable in data frames, the exact portion used for encoding remains unknown to an external observer. To further complicate analysis, the covert message is XORed with these values, making it more difficult for an adversary to reconstruct the hidden data without knowledge of both the selected MAC address bits and the applied operation. Moreover, to avoid any adverse impact on normal network performance, the final TXOP Duration Requested field within each A-MPDU is deliberately reset, thereby canceling any pending requests, ensuring that the covert communication remains transparent to regular network operation.

- Frame aggregation encoding aligns well with modern IEEE 802.11 enhancements. It is inherently stealthy, as the IEEE 802.11 standard does not mandate strict rules on how frames must be aggregated. The choice of aggregation layout can legitimately vary depending on factors such as channel conditions, station capabilities, and traffic patterns. For a covert capacity of B bits, the number of possible aggregation patterns reaches 2^B, which allows the covert message to blend into normal transmission behavior. This subchannel is highly transparent and difficult to detect, as frame aggregation is widely used in practical deployments to achieve throughput and efficiency gains.

The covert channel is practically applicable in deployments that support IEEE 802.11 QoS and frame aggregation. Moreover, advanced ML methods can accurately detect anomalies in frame fields, allowing the covert channel to persist in such environments. Even if one or two fields indicate the presence of covert channels, the frame aggregation layout has a high probability of survival in case of detection, because it differs from both duration fields that store the secret message in the frame header. Frame aggregation can legitimately change according to network conditions.

7.3 Simulation scenarios and metrics

7.3.1 Environment and scenarios

The simulations were conducted using the ns-3 network simulator [115], which provides comprehensive support for IEEE 802.11 networks, from basic functionalities to advanced features such as QoS support, frame aggregation, and network protocol stack. To implement covert channels based on the Duration/ID field in the MAC header and the TXOP Duration Requested field in QoS Data frames, the native methods from `wifi-mac-header.h` were called. The `SetDuration(Time duration)` to set the value of the `Duration/ID` field in the MAC header. `SetQosTxopLimit(uint8_t txop)` to sets the `TXOP Duration Requested` field in QoS Control in the MAC header. The covert channel based on frame aggregation was realized through custom modifications to the MAC

layer. Additional counters and instrumentation were implemented to measure relevant performance metrics and validate covert transmission behavior during the simulations.

The primary objective of the simulations is to analyze various parameters that may impact the performance of the covert channel. The selected parameters to investigate include frame size, the number of competing stations, as well as the covert transmitter's wireless medium, in addition to the covert transmitter, the maximum A-MSDU and A-MPDU size. Additionally, a simulation scenario is included where frame aggregation is disabled. This configuration enables the evaluation of the covert channel in environments without aggregation, thereby decoupling the channel's performance from the frame aggregation technique.

The fundamental simulation parameters used consistently throughout the experiments are summarized in Table 7.2. Unless explicitly stated otherwise, any parameter not mentioned as varying in a specific scenario should be assumed to take the default value specified in the table.

Simulations were repeated multiple times to ensure reliability, and the average values for each metric were computed. In all figures, the error margin for each simulation point, within a 95% confidence interval, did not exceed ±5%.

Table 7.2: StegoHybrid simulation parameters

Parameter	**Value**
Frame Size	512 [bytes]
Offered Load	Saturation
Transport Protocol	UDP
IEEE Standard	802.11ax
Frequency Band	5 [GHz]
Channel Width	20 [MHz]
Number of Tx and Rx Antennas	1
Mobility Model	Constant Mobility
MCS Index	11
Guard Interval	800 [ns]
RTS/CTS	Disabled
Block ACK	Enabled
Max A-MSDU Size	11398 [bytes]
Max A-MPDU Size	65535 [bytes]
Propagation and Loss Model	Log-Distance Path Loss
Distance (AP to STA)	1 [m]

7.3.2 Metrics

To measure the amount of data that can pass through the covert channel during a specific period, the metric total throughput is determined by combining the throughput from the individual contributions of each covert channel as per Equation 7.6:

$$\text{Throughput}_{\text{total}} = \text{Throughput}_{\text{duration}} + \text{Throughput}_{\text{txop}} + \text{Throughput}_{\text{agg}} \quad [\text{bps}] \quad (7.6)$$

- *Throughput from Duration/ID*: Three bits are embedded per A-MPDU via the Duration/ID field, considering that all the MPDUs within the A-MPDU must have the same duration. The throughput is calculated in Equation 7.7:

$$\text{Throughput}_{\text{duration}} = \frac{AMPDU_{\text{RX}} \cdot B_{\text{duration}}}{T_{\text{simulation}}} \quad [\text{bps}] \tag{7.7}$$

  where:

  - $AMPDU_{\text{RX}}$ is the number of A-MPDU frames received.
  - B_{duration} is the number of bits embedded per Duration/ID field.
  - $T_{\text{simulation}}$ is the simulation time in seconds.

- *Throughput via TXOP Duration Requested*: Each MPDU, except the last one of the A-MPDU, carries a covert message. If an A-MPDU contains only one MPDU, then that frame is counted. The throughput is given by the Equation 7.8:

$$\text{throughput}_{\text{txop}} = \frac{(MPDU_{\text{RX}} - R_{\text{ignored}}) \cdot B_{\text{txop}}}{T_{\text{simulation}}} \quad [\text{bps}] \tag{7.8}$$

  where:

  - $MPDU_{\text{RX}}$ is the total number of MPDUs within the received A-MPDU.
  - R_{ignored} is the number of last MPDUs excluded from each A-MPDU as per Equation 7.9.

$$R_{\text{ignore}} = \begin{cases} 1 & \text{if } N_{\text{MPDU}} > 1 \\ 0 & \text{if } N_{\text{MPDU}} = 1 \end{cases} \tag{7.9}$$

    where N_{MPDU} is the number of MPDUs in a given A-MPDU transmission.
  - B_{txop} is the number of bits embedded per MPDU via the TXOP Duration Requested.
  - $T_{\text{simulation}}$ is the simulation time in seconds.

- *Throughput in aggregation layout*: The aggregation layout encodes a secret of B-bits per A-MPDU. The covert throughput from the aggregation layout encoding is computed according to Equation 7.10:

$$\text{Throughput}_{\text{agg}} = \frac{AMPDU_{\text{RX}} \cdot B}{T_{\text{simulation}}} \quad [\text{bps}] \tag{7.10}$$

  where:

  - $AMPDU_{\text{RX}}$ is the number of A-MPDU frames received.
  - B is the channel bandwidth according to Equation 7.4.
  - $T_{\text{simulation}}$ is the simulation time in seconds.

Apart from the throughput, performance metrics such as delay, jitter, covert channel efficiency, and utilization ratio are defined respectively and calculated as:

- *Delay*: As shown in Equation 7.11, this metric represents the average time elapsed from the transmission of an A-MPDU to the reception of the corresponding Block ACK. It captures the total delivery time of each aggregated transmission and is normalized over the total simulation time:

$$\text{Delay} = \frac{\text{Tx A-MPDU Time} + \text{Rx BACK Time}}{T_{\text{Simulation}}} \quad [\text{ms}] \tag{7.11}$$

- *Jitter*: This metric quantifies the variation in delay across successive A-MPDU transmissions, and is computed using Equation 7.12, where N is the total number of received A-MPDU:

$$\text{Jitter} = \frac{1}{N-1} \sum_{i=2}^{N} |Delay_i - Delay_{i-1}| \quad [\text{ms}] \tag{7.12}$$

- *Efficiency*: As defined in Equation 7.13, this metric measures the reliability of the covert communication channel. It is the ratio of successfully acknowledged A-MPDU frames (i.e., those that received a Block ACK) to the total number of transmitted A-MPDUs, expressed as a percentage:

$$\text{Efficiency} = \frac{\text{Rx A-MPDUs}}{\text{Tx A-MPDUs}} \times 100 \quad [\%] \tag{7.13}$$

- *Covert Channel Utilization Ratio (CCUR)*: This generalized metric expresses how much a covert metric (e.g., throughput, delay, jitter) occupies or relates to the corresponding metric of the regular channel as per Equation 7.14. It allows evaluating the relative footprint of the covert channel in terms of resource usage or visibility.

$$\text{CCUR} = \frac{M_{\text{covert}}}{M_{\text{nominal}}} \times 100 \quad [\%] \tag{7.14}$$

  Where:

  - M_{covert} is the value of a given covert metric (e.g., covert throughput, covert delay).
  - M_{nominal} is the corresponding value for the regular channel (e.g., nominal data rate, delay).

7.4 Performance evaluation

To evaluate the proposed covert channel, it is important to note that all experiments were carried out under network saturation conditions and utilized the maximum allowable values for A-MSDU and A-MPDU, as detailed in Table 7.2. The purpose of this evaluation is to estimate the maximum achievable capacity of the covert channel through aggregation-based encoding. Figure 7.2 displays the maximum number of MSDUs per A-MSDU and the maximum number of MPDUs per A-MPDU for each tested frame size. These values were obtained according to the aggregation constraints outlined in Equations 7.1, 7.2, and 7.3. For instance, when the frame size is 256 bytes, each MPDU can carry

a maximum of an A-MSDU containing 37 MSDUs, and the A-MPDU can encompass up to 5 such MPDUs. This results in an aggregation layout of (37, 5), which means that a single A-MPDU transmission includes five MPDUs, each encapsulating 37 MSDUs.

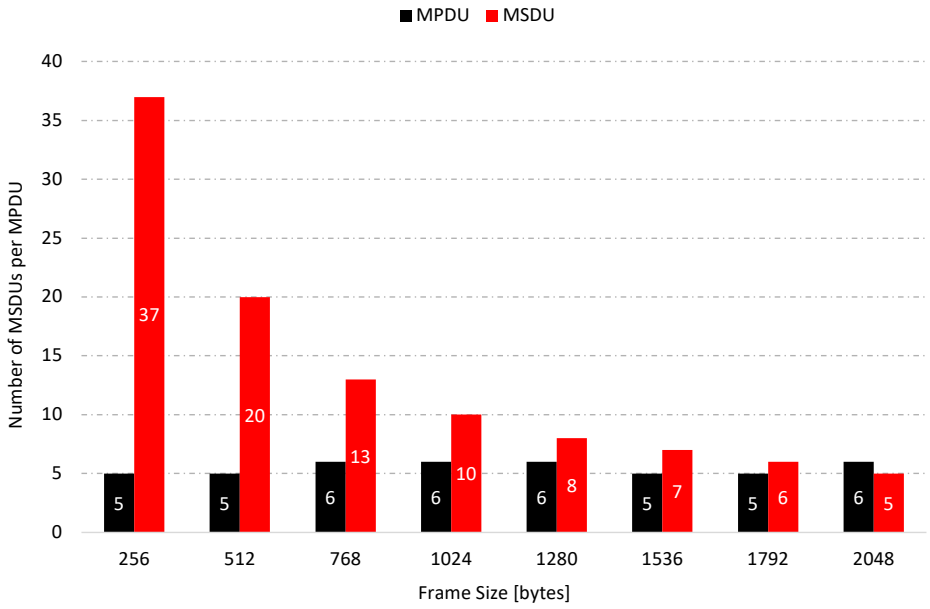


Figure 7.2: Maximum number of MPDUs and MSDUs per A-MPDU transmission for different frame sizes

Table 7.3 is created to calculate the maximum covert channel capacity based on the maximum values of the MSDU and MPDU, considering the frame size and the layout of each aggregation layout. From the table, we can conclude that any aggregation layout derived from the set $\mathcal{S}$, consisting of k elements, results in the transmission of a fixed-length covert message of B bits per A-MPDU. That implies that whether the sender alternates between different aggregation layouts or consistently uses the same one, the bandwidth of the covert channel remains constant, and such a strategy adds flexibility, allowing the sender to vary encoding strategies without affecting the overall channel capacity.

However, an important consideration must be addressed: the number of MPDUs within each A-MPDU directly influences the amount of covert data that can be embedded using the TXOP Duration Requested field. A higher number of MPDUs enables greater covert throughput from this field. In contrast, the number of bits that can be encoded via the Duration/ID field is limited by frame efficiency, as the duration value is shared between all MPDUs in a single A-MPDU transmission.

7.4.1 Impact of frame size

In the first scenario, we assess the covert station's capability without external traffic to understand its behavior, and then evaluate its performance when external traffic is present. In Figure 7.3, we start by analyzing the covert channel based on aggregation. As frame size increases, the throughput of this covert subchannel decreases, ranging from approximately 1.8 kbps to 1 kbps. This decline is primarily due to a reduction in the number of MSDUs per A-MSDU as the frame size grows. Larger frame sizes require fewer

Table 7.3: Covert channel capacity per frame size

F [bytes]	$\mathbf{M}_{\max}$	$\mathbf{N}_{\max}$	$\boldsymbol{k} = (M_{\max} - 2 + 1) \times N_{\max}$	$\boldsymbol{B} = \lfloor \log_2(k) \rfloor$
256	37	5	185	7
512	20	5	100	6
768	13	6	78	6
1024	10	6	60	5
1280	8	6	48	5
1536	7	5	35	5
1792	6	5	30	4
2048	5	5	25	4

Legend: F – The MSDU length; $M_{\max}$ – Maximum number of MSDUs per A-MSDU; $N_{\max}$ – Maximum number of MPDUs per A-MPDU; k – Total aggregation layout combinations (the size of the set $\mathcal{S}$); B – Covert bits encodable per A-MPDU

MSDUs to reach the maximum A-MSDU size, resulting in a decrease in the overall number of aggregated frames. While the number of MPDUs per A-MPDU remains relatively constant, with only minor variations. The throughput is logarithmically related to the product of the number of MSDUs and MPDUs. Consequently, having fewer MSDUs results in a lower product, which reduces the covert channel capacity.

Next, we analyze the covert channel embedded in the Duration/ID field. This subchannel demonstrates stable throughput, averaging approximately 776 bps across various frame sizes. This stability can be attributed to consistent channel transmission, regardless of the MSDU-to-MPDU ratio. The duration relies solely on A-MPDU transmission, and when examined in isolation, there is not much variation observed.

A higher throughput covert channel is the one based on the TXOP Duration Request, due to the 8 secret bits embedded linearly over the covert channel. A stable trend is observed, with an average throughput of approximately 8 kbps for 5 MPDUs and 10 kbps for 6 MPDUs. This increase in throughput is attributed to the growth in MPDUs, specifically, from 768 to 1280 bytes, and later to 2048 bytes, as the secret message is concealed within the MPDU.

The overall throughput, which represents the sum of the throughput in the three subchannels, closely follows the trend of the TXOP Duration Request. This component contributes the largest share of covert bits per A-MPDU transmission. Analyzing the behavior of the covert station in isolation reveals that the TXOP Duration Request is a dominant factor in the channel's behavior.

In terms of frame efficiency, the observed behavior is expected. Under isolated conditions, where there is no contention for channel access, collisions do not occur. Furthermore, we consistently observe a frame efficiency of 99.9%, regardless of frame size, as shown in Figure 7.4. This also indicates the consistent throughput of the covert channel shown in Figure 7.3. The ratio behaves as a linear function of the number of encoded bits in the A-MPDU, and regardless of the different frame sizes, the A-MPDU is consistently delivered with the same efficiency.

Analyzing Figure 7.5, we observe that the delay remains stable, averaging around 3.6 ms, indicating that the channel is in good condition, as the time required to transmit the A-MPDUs and the response experience similar conditions. The stable delay observed

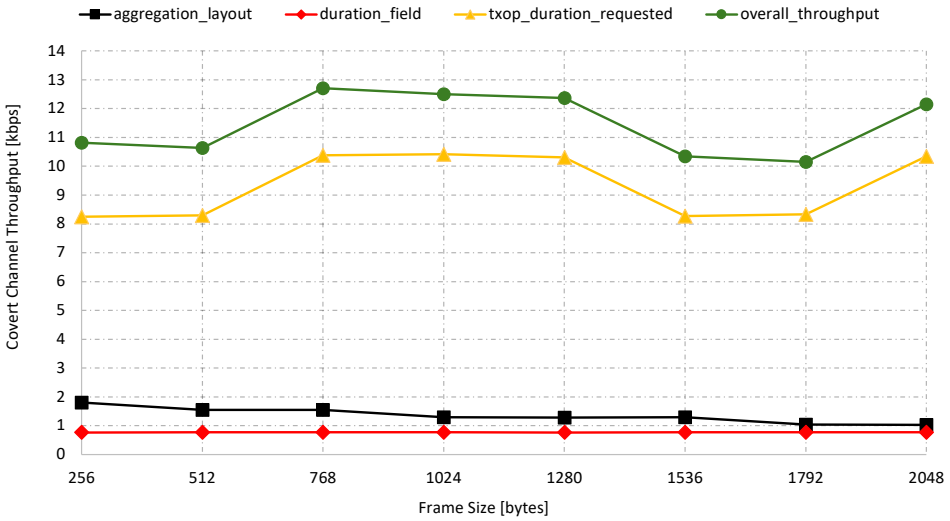


Figure 7.3: Covert channel throughput components versus frame size

across different frame sizes demonstrates consistently high frame efficiency, which remains around 99%. This consistency indicates that frames are delivered with equal effectiveness, regardless of their size, which suggests stable channel conditions in the absence of competition.

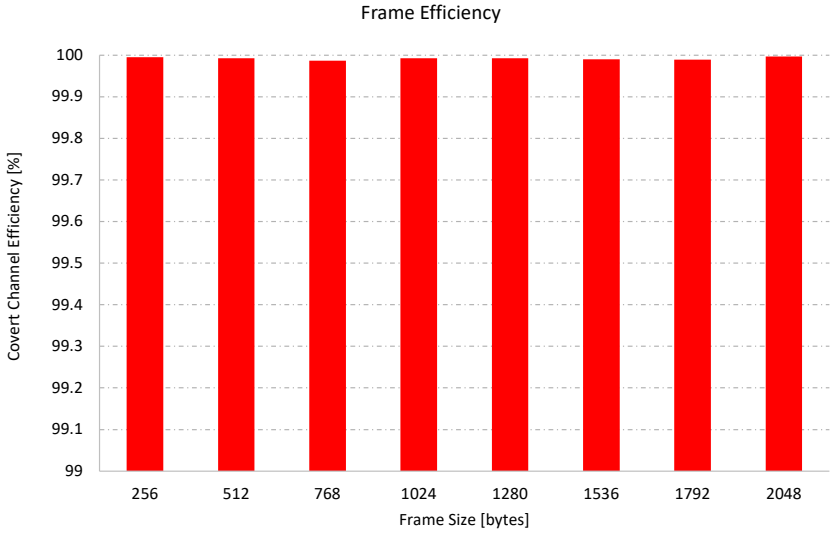


Figure 7.4: Covert channel efficiency components versus frame size

To evaluate the transparency of the covert channel, that is, how much of the resources of the regular channel it consumes, we begin by analyzing Figure 7.6, which compares the throughput of the regular channel with that of the covert channel. As frame size increases, regular throughput also increases, approaching saturation (100% channel utilization). This is expected since larger frame sizes allow more user data to be transmitted per frame, maximizing channel efficiency. However, the covert channel follows a similar trend already observed in previous throughput evaluations (Figure 7.3): frame sizes that allow aggregation with up to six MPDUs result in the highest covert throughput. Despite this increase, the figure clearly demonstrates the high transparency of the covert channel.

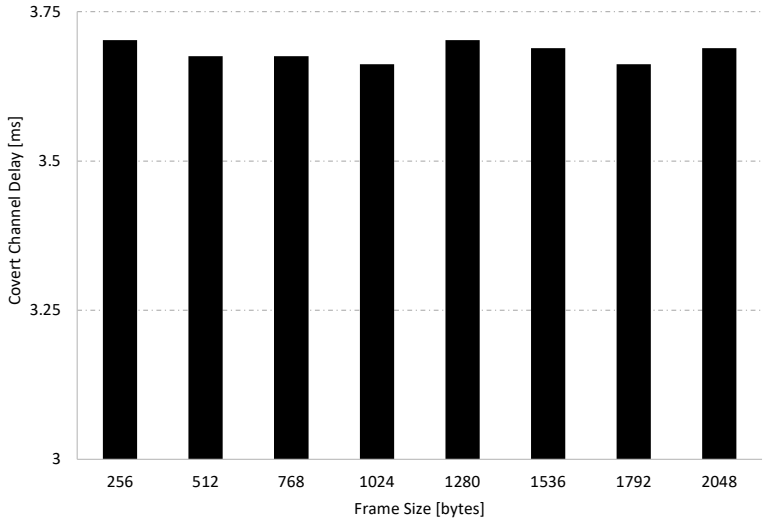


Figure 7.5: Covert channel delay components versus frame size

Specifically, the covert channel never consumes more than 0.01% of the total channel capacity. For example, even at its peak covert throughput of approximately 12 kbps (which is relatively high for covert channels), it represents almost only 0.01% of the maximum observed regular channel throughput. This highlights the covert channel's ability to operate with minimal impact on the primary communication channel, making it highly transparent.

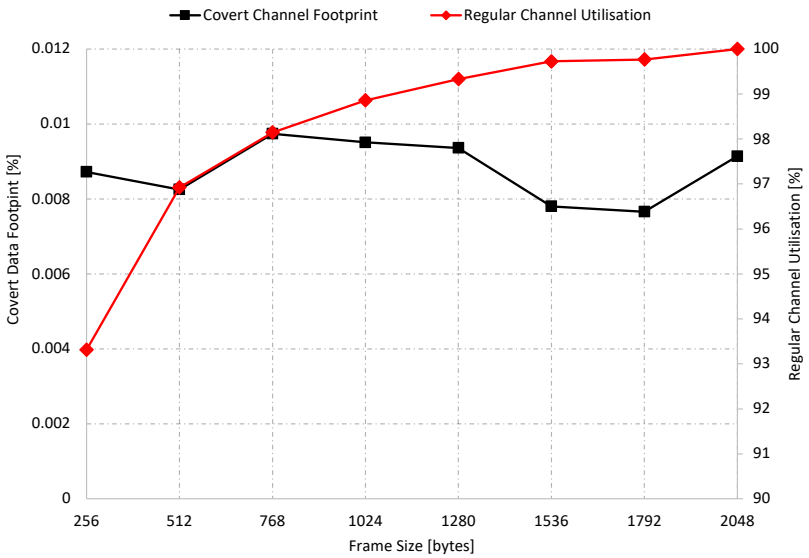


Figure 7.6: Covert channel throughput footprint versus frame size

An impressive aspect shown in Figure 7.7 is that the covert channel delay footprint across A-MPDUs remains consistent. At the same time, in the regular channel, increasing the frame size increases delay, as the lengthy frame requires more time to be transmitted and also extends the queue on the receiver side. It is essential to note that when calculating the delay for the regular channel, all frames are considered, regardless of whether they are aggregated or not, based on the queue size and current channel conditions. In

contrast, the covert channel delay measurement only accounts for frames deliberately aggregated by the covert station to convey secret information.

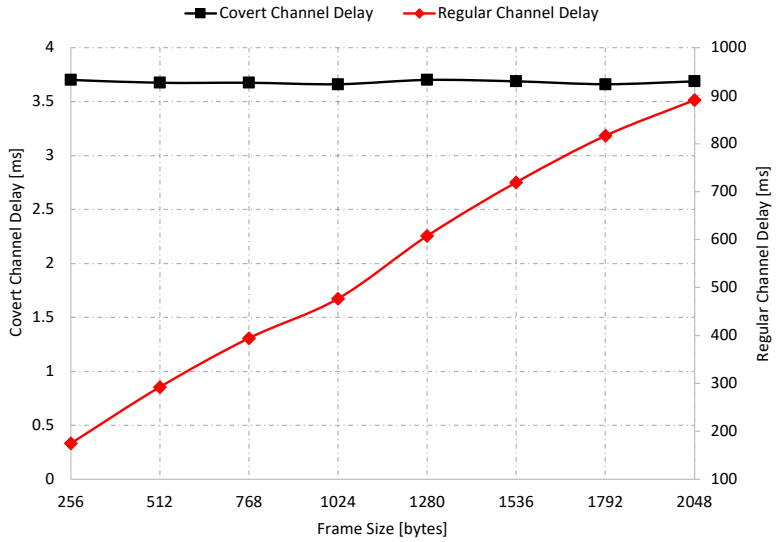


Figure 7.7: Covert channel delay footprint versus frame size

7.4.2 Impact of competing stations and frame size

In this section, we evaluate how external traffic affects the performance of the covert channel while keeping the frame size fixed at 512 bytes. As presented in Figure 7.8, the experiment progressively increases the number of competing stations on the channel, beginning with one external station and incrementally rising to 5, 10, 15, and finally to 20. Initially, when only one additional station is introduced, the overall covert channel throughput decreases by almost half compared to the isolated scenario, dropping to approximately 5 kbps. As more stations join the network, throughput decreases across all covered subchannels. The aggregation layout and Duration/ID subchannels, in particular, begin to converge in throughput as early as five competing stations, stabilizing at around 200 and 100 bps, respectively, and eventually dropping to approximately 44 and 22 bps for each. The drop in throughput as more stations join the network can be explained by increased contention for the medium. The Duration/ID field reflects this congestion: as more stations attempt to transmit, the channel becomes busier, leading to fewer A-MPDUs being successfully sent. Additionally, the efficiency of the aggregation layout decreases under contention, as it becomes harder to transmit multiple MPDUs within an A-MPDU without collisions or delays. The TXOP Duration Requested subchannel follows the same trend, having to its advantage the number of bits encoded per A-MPDU.

Frame efficiency is illustrated in Figure 7.9. As the number of competing stations increases, the efficiency of A-MPDU transmissions declines. This decrease results from a higher likelihood of collisions and increased queuing delays, which raise the probability of frame drops at both the sender and receiver sides. The plot reflects this effect through a reduced ratio of successfully received to transmitted A-MPDUs. However, even under heavy contention with 20 competing stations, the covert channel efficiency remains

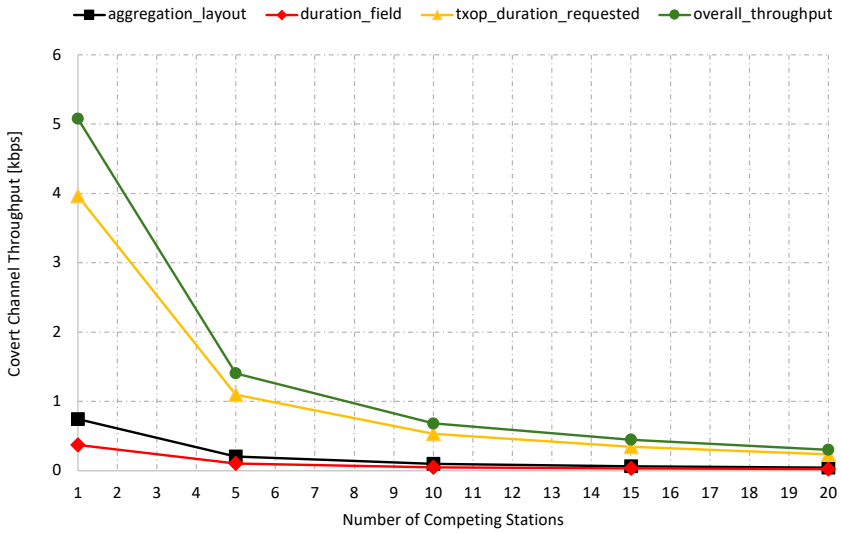


Figure 7.8: Impact of channel contention on covert channel throughput components

above 50%, indicating that more than half of the transmitted A-MPDUs still reach their destination successfully.

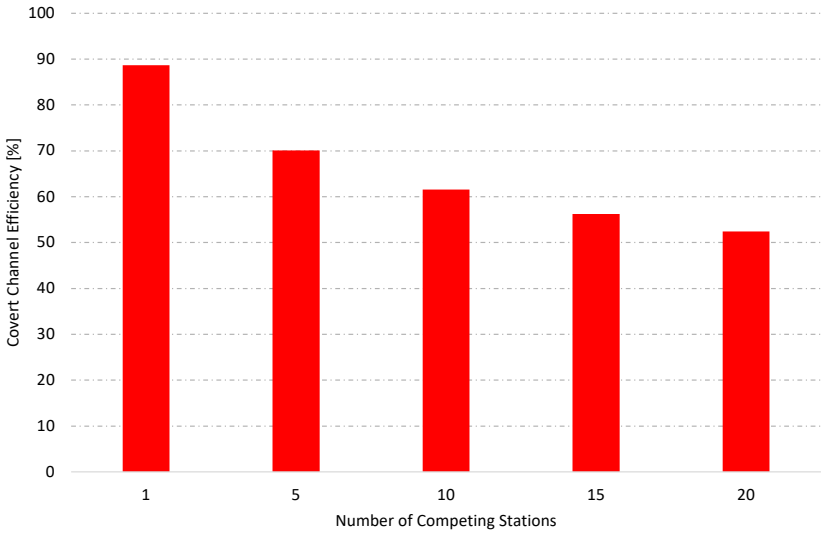


Figure 7.9: Impact of channel contention on covert channel efficiency

The delay shown in Figure 7.10 slightly increases by half a millisecond with the addition of one more station and continues to rise as more stations are added, reaching up to 40 milliseconds. This increase in delay is due to several factors, such as the queue size and the time frames spend waiting in the queue to be transmitted or processed after being received. This occurs because the access point's resources are now shared among multiple stations, as well as the wireless channel.

Figure 7.11 illustrates the jitter of covert channels as a function of the number of competing stations. As more stations join the network, the jitter steadily increases, surpassing 1.4 ms when 20 stations are present. This increase reflects the growing variability in transmission delays caused by channel contention, queuing delays, and shared access to the medium. With more stations vying for access, frames experience longer wait times

in queues before being transmitted or processed upon receipt, resulting in greater timing fluctuations between successive transmissions. This trend is consistent with the observed delay patterns, where increased contention leads to higher average delays, which subsequently affect the consistency of frame delivery intervals and contribute to higher jitter.

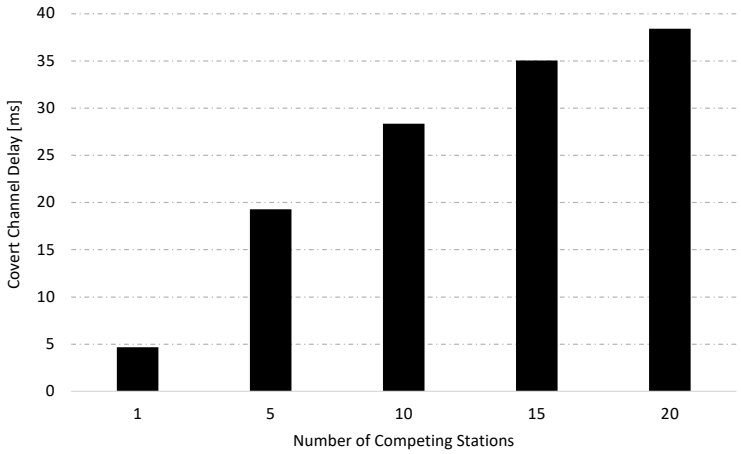


Figure 7.10: Impact of channel contention on covert channel delay

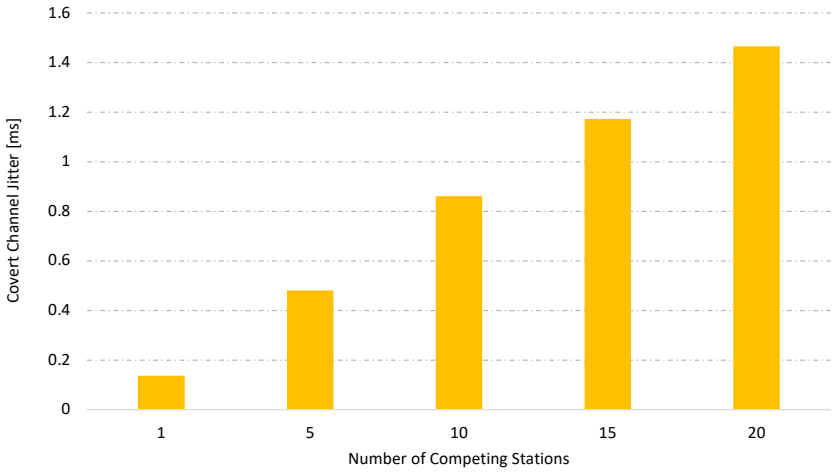


Figure 7.11: Impact of channel contention on covert channel jitter

The resource utilization of covert channels in terms of throughput is illustrated in Figure 7.12. The performance of regular covert channels is significantly affected by the increasing number of stations. The channel utilization drops to about 50% once a second station joins; as more stations access the medium, it further declines to below 10% of the available bandwidth when there are more than five stations. The utilization of the covert channel is also impacted; fewer stations allow for a more observable footprint of the covert channel compared to the available bandwidth. The maximum consumption of the covert channel ranges from approximately 0.00825% (5 kbps of covert data with one station present) to 0.0081% (303 bps with 20 stations present).

The delay footprint, as shown in Figure 7.13, reflects the behavior of the covert channel in relation to the regular network delay. While the regular channel experiences

increasing delays as more stations join the network, due to longer queues and more frequent contention, the covert channel exhibits a similar trend but on a much smaller scale. This is because the covert channel leverages A-MPDU frames under saturation conditions, mirroring the transmission behavior of the regular traffic.

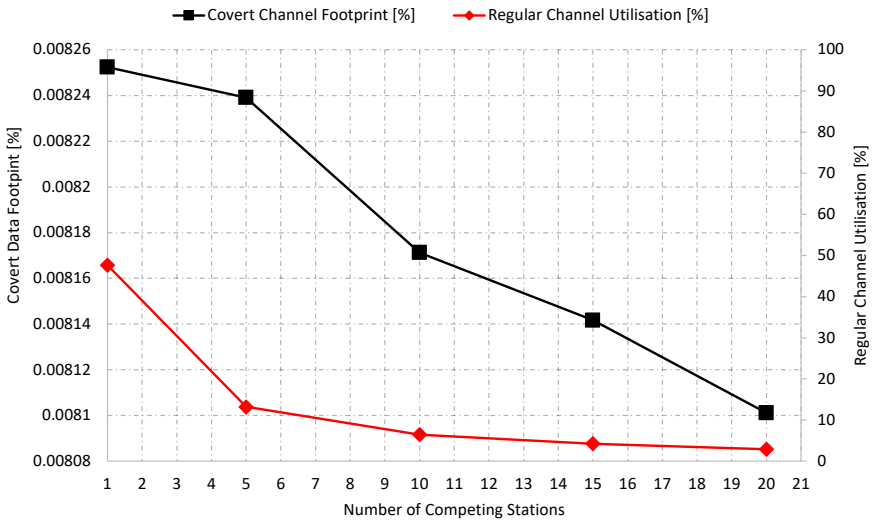


Figure 7.12: Impact of channel contention on covert channel throughput footprint

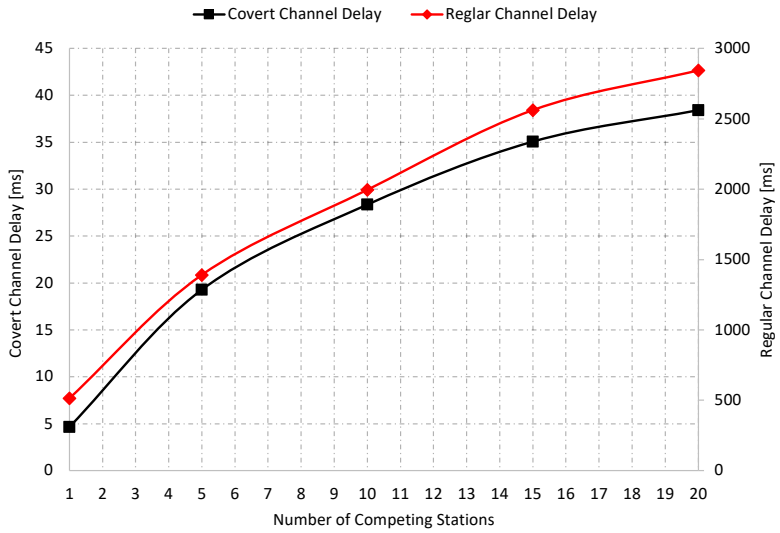


Figure 7.13: Impact of channel contention on covert channel delay footprint

To gain a comprehensive understanding of covert channel behavior, we evaluate the combined effects of frame size and the number of competing stations. Figures 7.14a (2D view) and 7.14b (3D surface plot) illustrate how these two factors influence covert throughput. This dual-parameter analysis allows us to identify the optimal conditions under which the covert channel can achieve higher performance. The results show that larger frame sizes (particularly those between 768 and 1280 bytes, and to some extent 2048 bytes) enable higher throughput when the channel is lightly loaded (i.e., with 1 to 5 external stations). This is because more MPDUs are generated, maximizing the amount

of covert data encoded per A-MPDU transmission. However, as the number of competing stations increases, the throughput steadily declines across all frame sizes. When there are more than 10 external stations, the advantage of larger frames is significantly reduced. Between 15 and 20 stations, the throughput remains constant and becomes almost independent of frame size, indicating that channel contention becomes the primary limiting factor, canceling any benefits from aggregation.

In Figures 7.15a (2D view) and 7.15b (3D surface plot), we illustrate how the efficiency of the covert channel is significantly influenced by the number of competing stations rather than by the frame size. Regardless of the frame size used, the efficiency reflects the channel's capability to transmit frames from the sender to the receiver successfully. As the number of stations increases from one to five, we observe a substantial decline in efficiency, approximately 20% for one, and then also from one to five additional stations. Beyond five stations, efficiency continues to decrease, but never drops below 50%. Notably, from around 15 competing stations onward, the efficiency stabilizes. This indicates a saturation point where adding more stations results in only minimal further reductions in efficiency compared to the initial decrease.

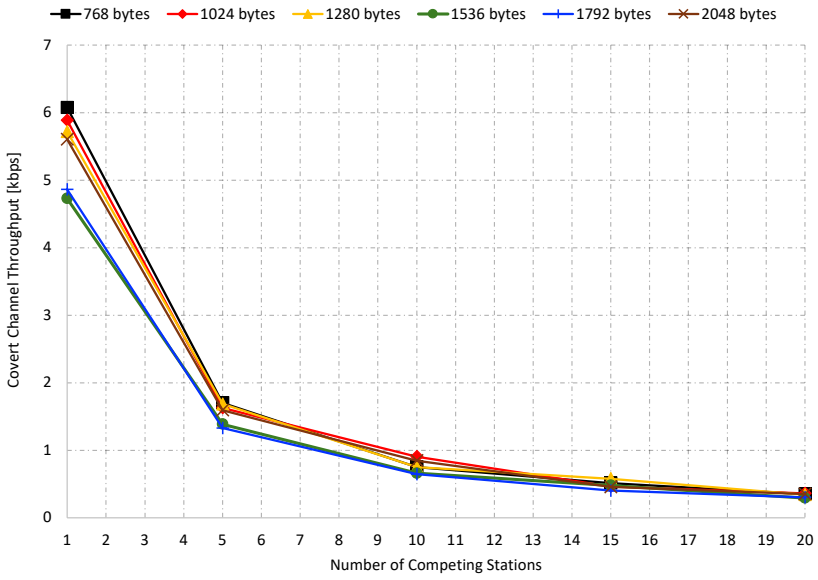
Figure 7.16 shows how covert channel delay varies with the number of competing stations and frame sizes. As in the efficiency results, the delay increases significantly with the number of stations, regardless of frame size. With a single station, the delay remains low across all frame sizes (around 5 ms), indicating minimal contention. However, as more stations are added, the delay grows sharply, particularly between 5 and 15 stations, and continues to increase beyond 15 stations. This trend is consistent with the efficiency behavior, where higher contention reduces performance and leads to longer delays. Moreover, smaller frames (e.g., 768 and 1024 bytes) suffer from higher delays, especially under heavy contention, likely because they require more transmission time to deliver the same amount of data. Conversely, larger frames (e.g., 1792 and 2048 bytes) generally achieve lower delays, as fewer medium accesses are needed.

Figure 7.17 confirms this effect in terms of jitter. Similar to delay, jitter increases with the number of competing stations, especially for smaller frames (e.g., 768 and 1024 bytes), reflecting greater inconsistency in transmission timing. Larger frames (e.g., 2048 bytes) tend to exhibit lower jitter even under heavy load, providing more stable timing for covert message delivery in congested scenarios.

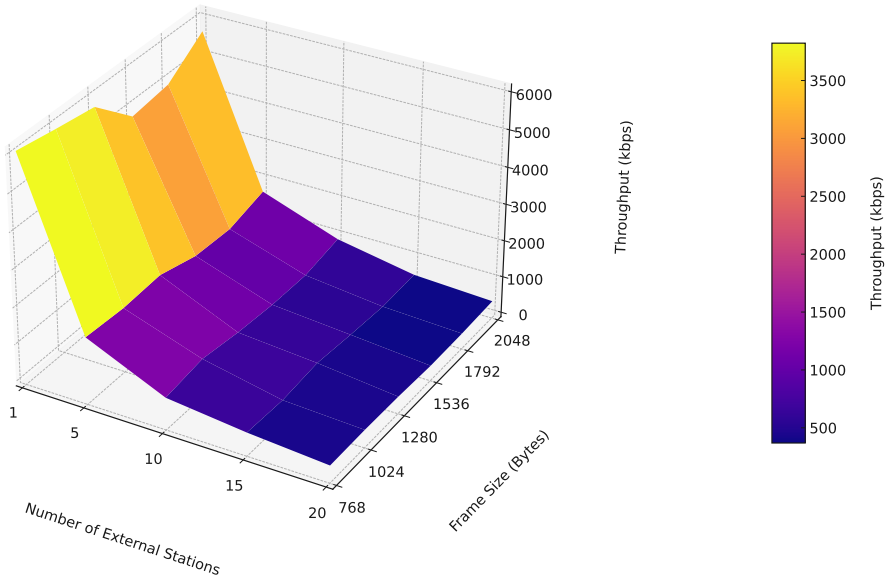
7.4.3 Impact of the maximum A-MSDU and A-MPDU

In this section, we analyze how the behavior of the covert channel is affected by changes in the maximum A-MSDU and A-MPDU sizes, while keeping the frame size fixed at 512 bytes. Our goal is to understand how these parameters directly influence the layout of the aggregation and, consequently, impact the performance of the covert channel. In the Figure 7.18a, we examine the impact of increasing the A-MSDU size. As the A-MSDU size increases from 1712 bytes to 11398 bytes, the number of MSDUs per MPDU decreases from 38 to just 5. This reduction occurs because larger A-MSDUs occupy more space within an MPDU, thereby reducing the number that can be packed together. At the same time, the number of MPDUs per A-MPDU increases. This indicates that fewer large A-MSDUs per MPDU allow for more MPDUs to be included in the overall A-MPDU frame. This trade-off shows that while larger A-MSDUs reduce the granularity of the subframes, they may enable a higher number of aggregate units at the MPDU level.

Conversely, the second plot evaluates the effect of increasing the A-MPDU size in



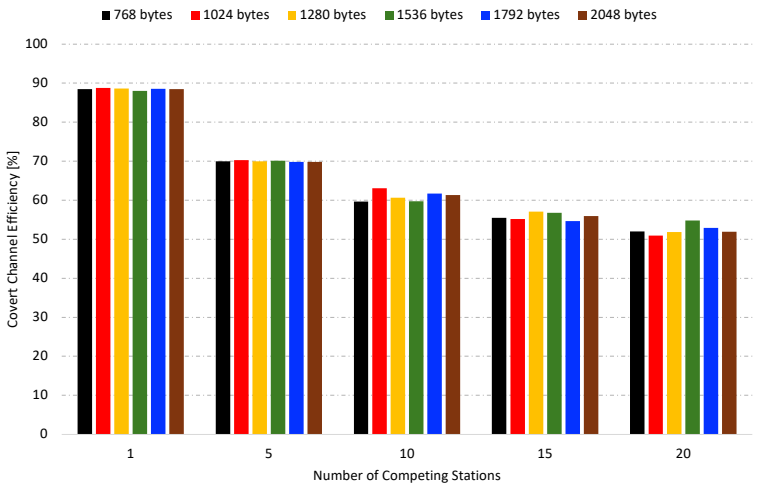
(a) Covert throughput vs. number of competing stations and frame size (2D view)



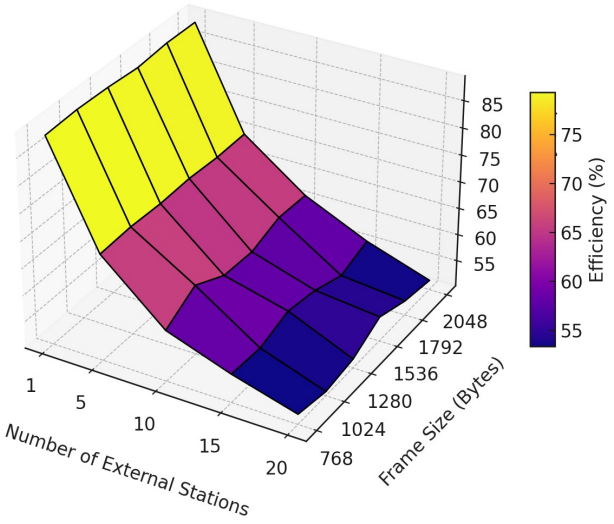
(b) Covert throughput vs. number of competing stations and frame size (3D surface plot)

Figure 7.14: Combined analysis of covert throughput influenced by frame size and network contention

Figure 7.18b. Here, the number of MSDUs per MPDU remains constant at 20, reflecting a fixed A-MSDU configuration. However, the number of MPDUs per A-MPDU grows as the A-MPDU size increases, rising from 1 MPDU at 10000 bytes to 5 MPDUs at the



(a) Covert throughput vs. number of competing stations and frame size (2D view)



(b) Covert throughput vs. number of competing stations and frame size (3D surface plot)

Figure 7.15: Combined analysis of covert throughput influenced by frame size and network contention

maximum size of 65535 bytes. This scaling enables more data to be transmitted in a single transmission opportunity, thereby improving overall throughput. This improvement results from the increased throughput during the TXOP Duration Requested and the combination of MSDUs to MPDUs.

In Figure 7.19a, we analyze the behavior of the covert channel in isolation. For the aggregation-based covert channel, we observe that as the maximum A-MSDU size increases, the number of MSDUs per MPDU decreases; however, the number of MPDUs per A-MPDU increases. This creates a balance: when one value decreases, the other

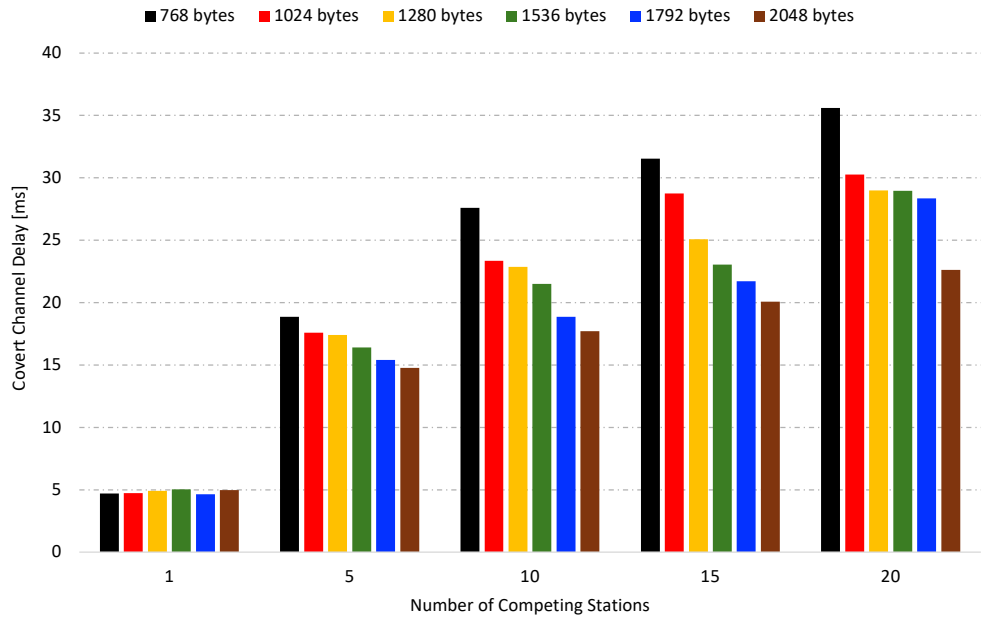


Figure 7.16: Impact of channel contention and frame size on covert channel delay

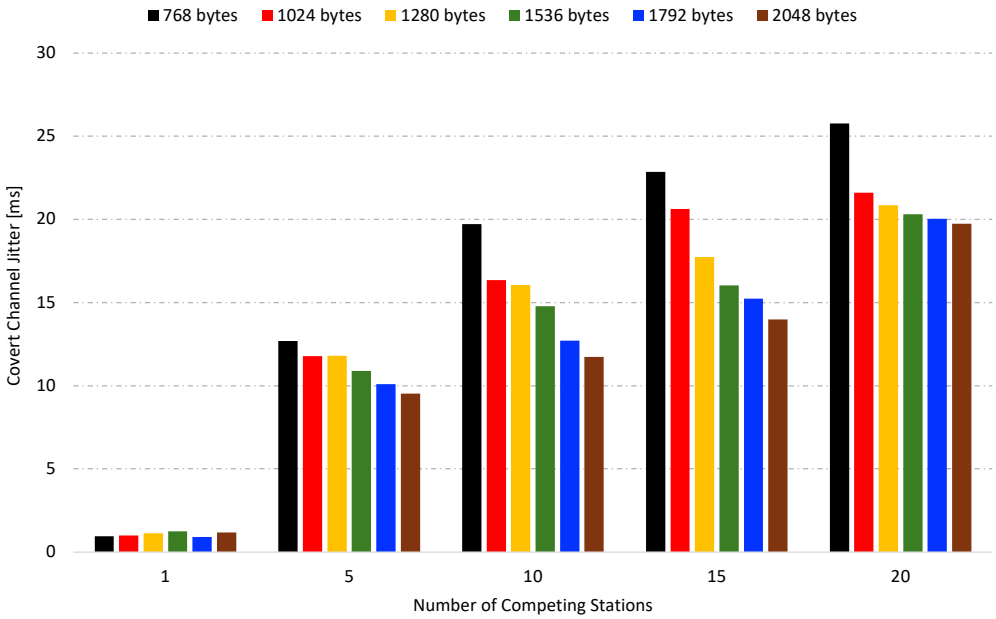
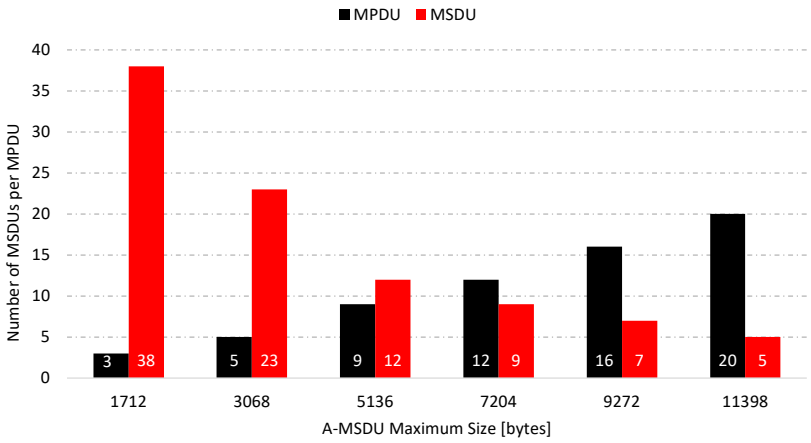
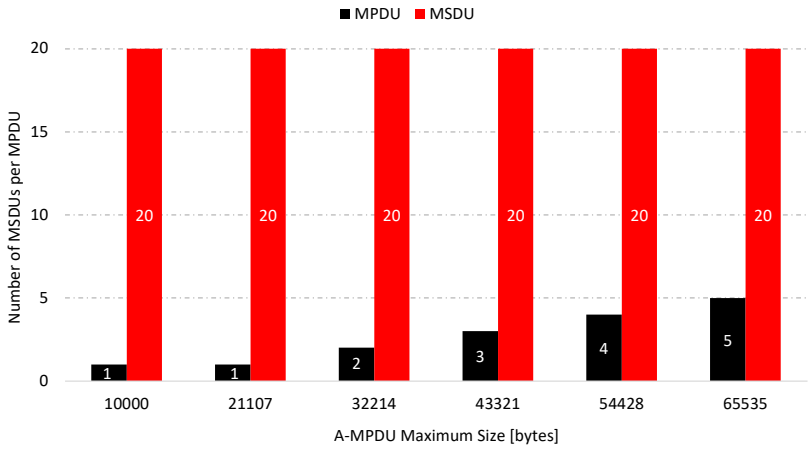


Figure 7.17: Impact of channel contention and frame size on covert channel jitter

increases proportionally. As demonstrated in Table 7.2, the parameter k, which represents the product of these two values (MSDUs $\times$ MPDUs), remains within a consistent range. This variation explains the behavior of the subchannel using the aggregation layout to encode a secret message. Since throughput is calculated on a logarithmic scale, the resulting values remain within the same scale, ultimately achieving a maximum capacity of 6 bits per A-MPDU transmission. The covert channel that uses the Duration/ID field has nearly constant throughput. However, this reflects the efficiency of the covert channel when considered in isolation, as it embeds the secret message within each A-MPDU transmission. As demonstrated in Figure 7.21a, the frame efficiency remains constant



(a) Number of MSDUs per MPDU vs. A-MSDU maximum size



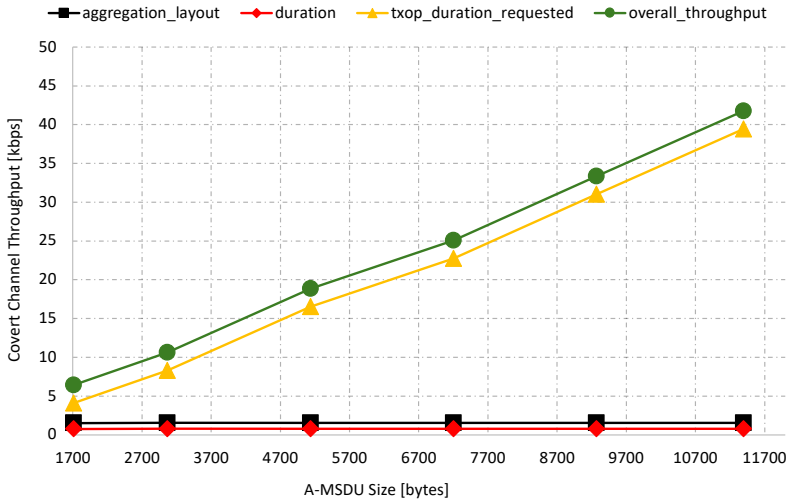
(b) number of MSDUs per MPDU vs. A-MPDU maximum size

Figure 7.18: Influence of maximum A-MSDU and A-MPDU sizes on aggregation layout

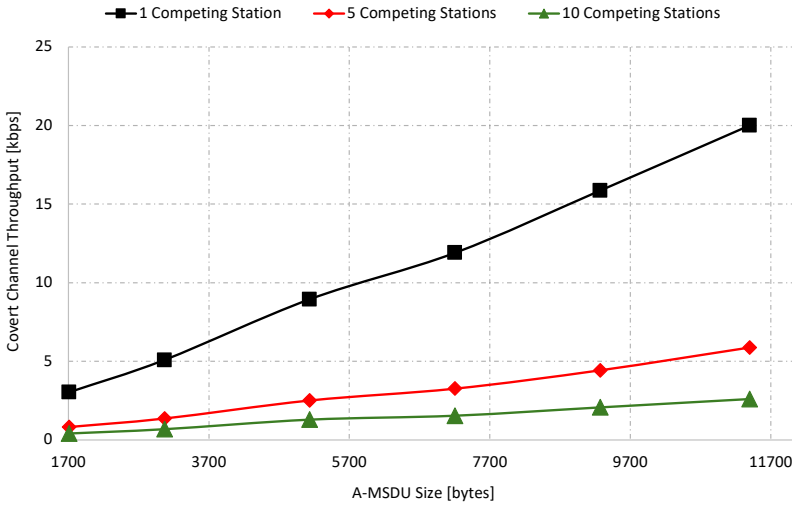
regardless of the frame size. Among covert channels, the one that benefits the most from an increase in A-MSDU size is the one that utilizes the TXOP Duration Requested field. This advantage arises because it depends on the number of MPDUs generated. As the maximum size of A-MSDU increases, more MPDUs are required to accommodate the same number of MSDUs, resulting in more MPDUs carrying secret data.

Figure 7.19b illustrates the overall behavior of the covert channels under external traffic load. As anticipated, increased channel access contention (e.g., with 10 stations) results in a reduction in covert throughput. However, even during average load conditions, employing larger A-MSDU sizes helps sustain higher overall covert throughput, particularly for the TXOP-based covert channel.

In Figure 7.20, we analyze the impact of the maximum A-MPDU size. When changing the maximum A-MPDU size in isolation, as shown in Figure 7.20a, for the Duration/ID-based covert channel, which encodes information per A-MPDU regardless of its internal aggregation structure, the throughput decreases as the A-MPDU size increases. This behavior occurs because smaller MPDUs are typically generated and transmitted more frequently, resulting in a higher rate of A-MPDU transmissions. These smaller frames are ready for transmission sooner and incur lower queuing delays, allowing the transmitter to



(a) Impact of A-MSDU size on covert throughput in isolation

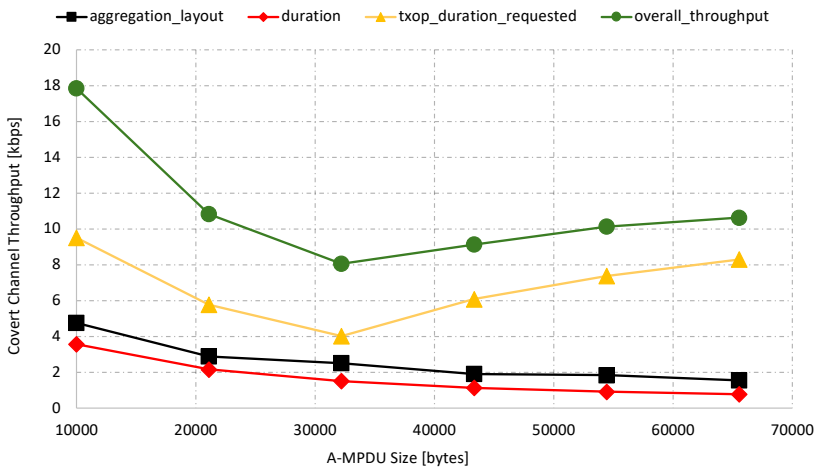


(b) Impact of A-MSDU size on covert throughput with channel contention

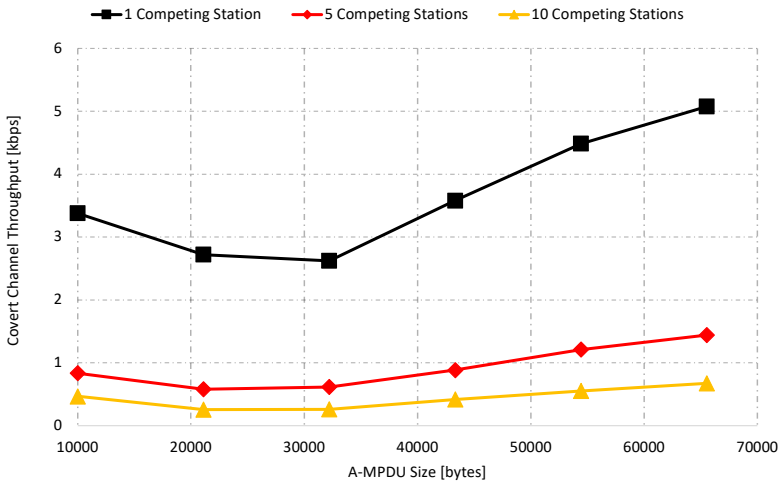
Figure 7.19: Combined analysis of the impact of A-MSDU size on covert throughput in isolation and with channel contention

quickly fill and send them. In contrast, larger A-MPDUs require more time to aggregate sufficient data before transmission, which reduces the overall number of transmissions. Although they are more efficient per transmission, the lower frequency of large A-MPDUs results in reduced throughput. For the aggregation-based covert channel, although the number of bits per A-MPDU increases (since the logarithm of the product of MSDUs x MPDUs increases), we observe a similar limitation: larger A-MPDUs are transmitted less frequently. While increasing the number of MPDUs can theoretically increase the encoded value, the total number of A-MPDUs carrying these aggregations decreases. This results in lower actual throughput, since the covert channel depends on how often these aggregated frames can be sent. The TXOP Duration Requested covert channel behaves slightly differently. Although it also suffers from a reduced A-MPDU frequency when the A-MPDU size increases, it benefits from the fact that larger A-MPDUs contain

more MPDUs, where the covert data is encoded. In the first few A-MPDU sizes (from 10.000 to 32.214 bytes), its throughput trend is similar to that of the Duration/ID sub-channel due to the reduced frequency of A-MPDU transmissions. However, due to the linear encoding model used in the TXOP-based channel (which assigns a covert value per MPDU), it continues to carry more covert bits per transmission as the MPDU count grows. Additionally, the TXOP-based channel design assumes that at least two MPDUs per A-MPDU are needed to avoid wasting encoding capacity (Equation 7.8). While larger A-MPDUs reduce transmission frequency, the increased number of MPDUs per A-MPDU enhances covert capacity per transmission. Figure 7.20b illustrates the impact of channel contention on the overall covert channel throughput. While the overall behavior of the covert channel remains unchanged, its throughput decreases as more stations join the network



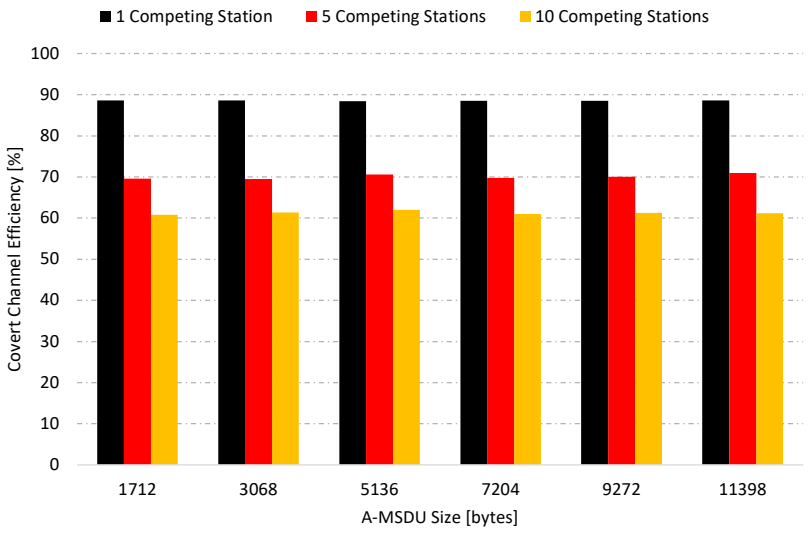
(a) Impact of A-MPDU size on covert throughput in isolation



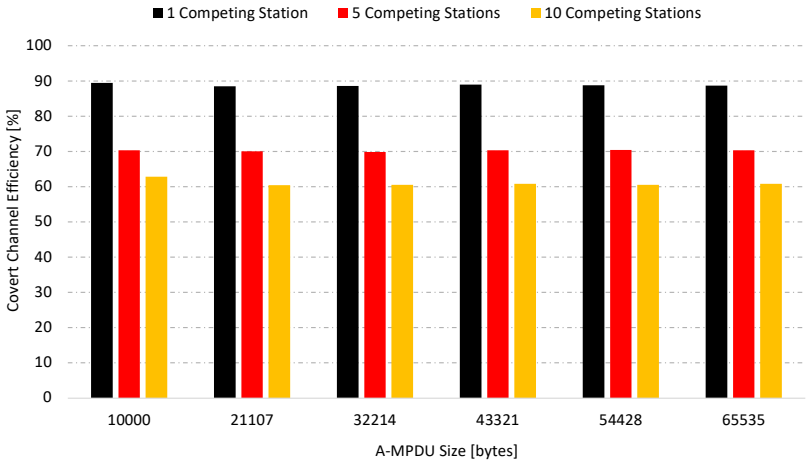
(b) Impact of A-MPDU size on covert throughput with channel contention

Figure 7.20: Combined analysis of the impact of A-MPDU size on covert throughput in isolation and with channel contention

Figures 7.21a and 7.21b illustrate the efficiency of the covert channel as the maximum sizes of A-MSDU and A-MPDU are varied under different levels of network contention (1, 5, and 10 competing stations). For both A-MSDU and A-MPDU size adjustments, a consistent trend emerges: the number of competing stations has a significant impact on efficiency. As contention increases, the efficiency of the covert channel decreases. With only one competing station, the efficiency remains high (around 85 - 88%), whereas with 10 stations, it drops to about 60 - 65%. The efficiency remains nearly constant across all A-MSDU sizes, suggesting that increasing A-MSDU size has little to no effect on the covert channel's ability to transmit successfully. A similar trend is observed with A-MPDU sizes; changes in A-MPDU size do not significantly impact covert channel efficiency across any level of contention.



(a) Covert channel efficiency vs. maximum A-MSDU size with channel contention



(b) Covert channel efficiency vs. maximum A-MPDU size with channel contention

Figure 7.21: Combined analysis of covert efficiency for maximum A-MSDU and A-MPDU size with channel contention.

Figures 7.22a and 7.22b illustrate the covert channel delay as a function of maximum A-MSDU and A-MPDU sizes under varying numbers of competing stations. The covert channel delay remains relatively stable across all A-MSDU sizes. However, it is significantly influenced by the number of competing stations: with one station, the delay is approximately 20 ms. With five stations, the delay remains similar, and for larger A-MSDU sizes, it is slightly lower. However, adding ten more stations noticeably increases the delay to around 30 ms. The A-MSDU size does not significantly affect the delay, indicating that frame size has minimal impact on timing in this context. In contrast to A-MSDU, the delay of the covert channel increases as the A-MPDU size rises in the presence of contention. The delay steadily grows with larger frame sizes, as more MPDUs create larger queues and take longer to complete a transmission compared to shorter ones.

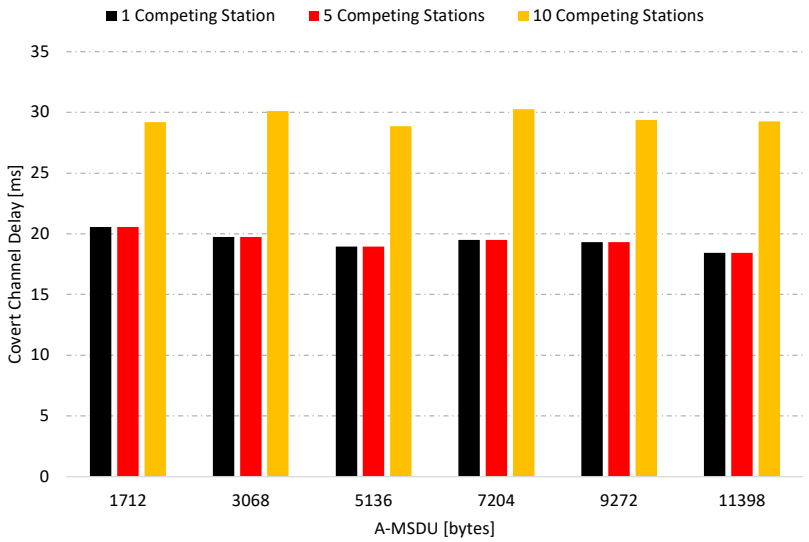
In Figures 7.23, we analyze the behavior of covert channel jitter as the maximum A-MSDU and A-MPDU sizes vary under channel contention. The results indicate that network contention is the primary factor influencing jitter, which also corresponds to the delays observed in previous studies. In the A-MSDU jitter plot shown in Figure 7.23a, the number of competing stations has a more significant impact on jitter than the maximum A-MSDU size. This suggests that once contention is introduced, the main contributor to jitter is the competition for access to the medium rather than the A-MSDU size. In contrast, the A-MPDU jitter plot in Figure 7.23b shows a combined effect of both contention and aggregation. This is mirrored by the delay observed in the same figure, indicating that jitter is a consequence of the delays between consecutive frames. This implies that larger A-MPDU sizes lead to higher transmission delays and variability, likely due to longer transmission times and queuing delays during channel contention.

7.4.4 Covert channel without aggregation

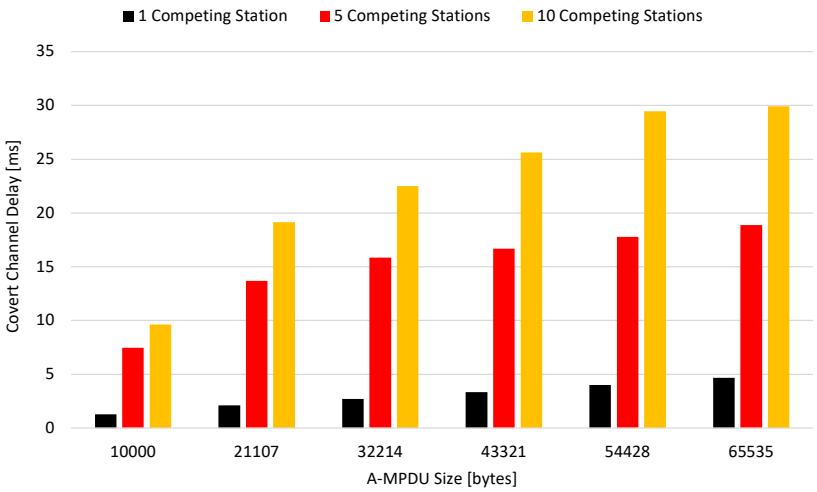
Among the three covert channel components, frame aggregation proved to be a pivotal factor influencing the overall covert channel performance. The number of MPDUs was strongly dependent on the configured maximum A-MSDU and A-MPDU sizes, which in turn affected the bandwidth available for the TXOP Duration Requested covert channel. Additionally, the use of larger A-MPDUs impacted the Duration/ID-based covert channel, since it operates at the A-MPDU level. These dependencies reveal that frame aggregation conditions the performance of the covert channel.

In this section, we analyze the potential deployment of covert channels independent of frame aggregation, focusing instead on the two duration-related fields, Duration/ID and TXOP Duration Requested, which together offer a combined covert bandwidth of 11 bits per MPDU. As illustrated in Figure 7.24, the throughput of the secret message within the Duration Field, TXOP Duration Requested, and overall throughput shows a declining trend as the frame size increases from 256 bytes to 2048 bytes. This observation confirms that smaller frame sizes yield higher covert throughput, as they enable more frequent transmissions and allow for a greater number of MPDUs (MAC Protocol Data Units) per unit of time. The throughput in the Duration/ID category ranges from 13 to 9 kbps, the TXOP Duration ranges from 35 to 24 kbps, and the overall throughput varies from 48 to 34 kbps. This indicates that as the covert throughput increases, the throughput decline becomes more pronounced. At first glance, we can observe that without frame aggregation, the maximum achievable throughput increases by approximately 35 kbps.

The efficiency of the covert channels, as illustrated in Figure 7.25, demonstrates that larger frames have a higher probability of being successfully delivered. Although shorter



(a) Covert channel delay vs. maximum A-MSDU size with channel contention



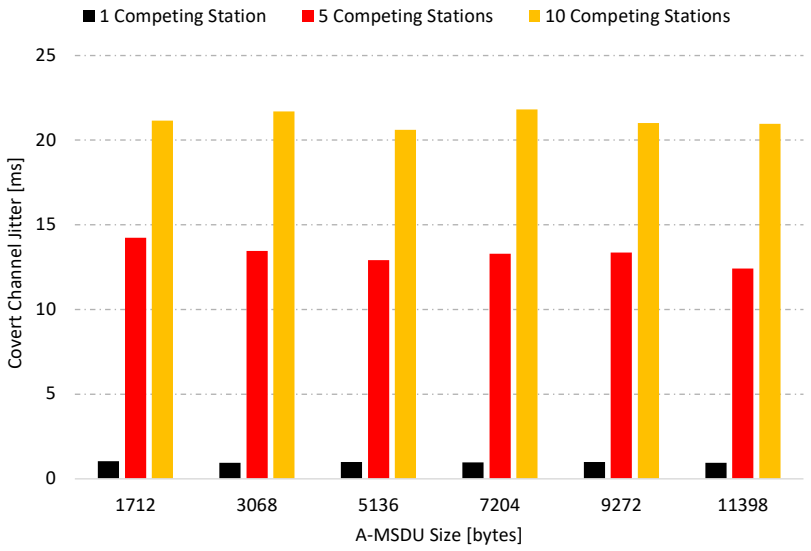
(b) Covert channel delay vs. maximum A-MPDU size with channel contention

Figure 7.22: Combined analysis of covert delay for maximum A-MSDU and A-MPDU size with channel contention

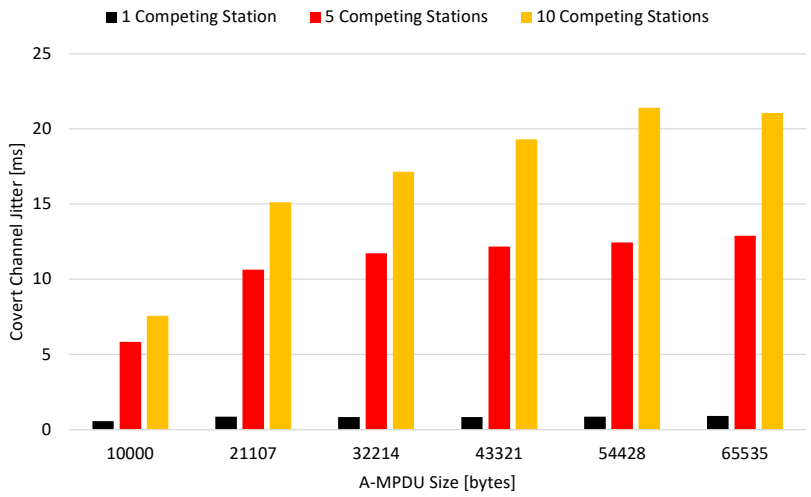
frames may initially increase the amount of generated data, they also result in a greater number of frames being queued for transmission or acknowledgment. This increases the likelihood of frame drops due to an excess of frames, which can cause frames to remain in the queue for longer and, consequently, expire. This scenario can occur at both ends of the queue.

As a result, we observe a linear increase in efficiency as the frame size also increases. However, as the frame size continues to grow, the channel efficiency does not exceed 50%.

Figures 7.26 and 7.27 illustrate the behavior of covert channel delay and jitter as the frame size increases from 256 to 2048 bytes. In Figure 7.26, we observe a nearly imperceptible increase in delay up to a frame size of 1024 bytes. Beyond this point, there



(a) Covert channel delay vs. maximum A-MSDU size with channel contention



(b) Covert channel jitter vs. maximum A-MPDU size with channel contention

Figure 7.23: Combined analysis of covert jitter for maximum A-MSDU and A-MPDU size with channel contention

is a noticeable increase of about 200 ms in delay, indicating that covert channels handle delays differently for frames below 1024 bytes, while they become more reactive to delays at larger sizes. This behavior occurs because larger frames require more time for both transmission and acknowledgment. Additionally, the number of frames transmitted per second decreases as the payload size increases, leading to a longer average time between successive covert transmissions.

Figure 7.27 similarly demonstrates an increase in jitter with larger frame sizes, reflecting the trend observed in delay. Jitter starts at approximately 0.08 ms for 256-byte frames and gradually increases to over 0.1 ms at 2048 bytes. This observation aligns with the findings presented in Figure 7.26, indicating that larger frame sizes not only increase

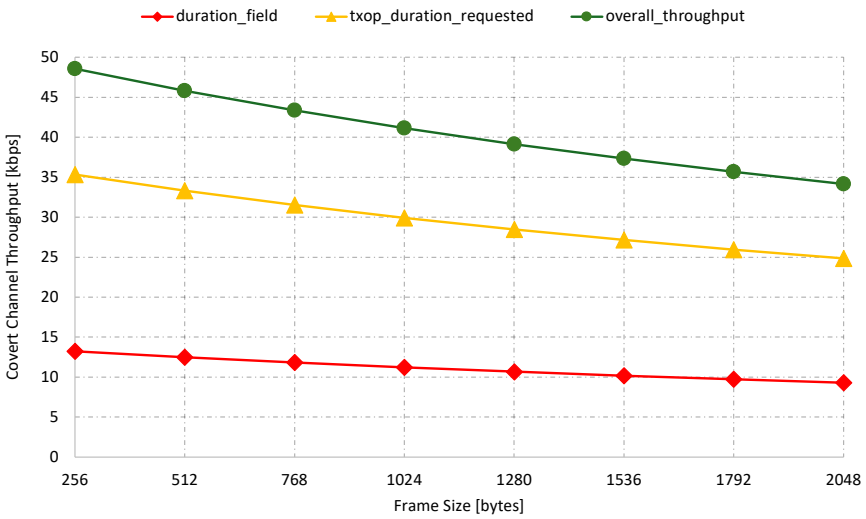


Figure 7.24: Effect of frame size on covert throughput with aggregation disabled

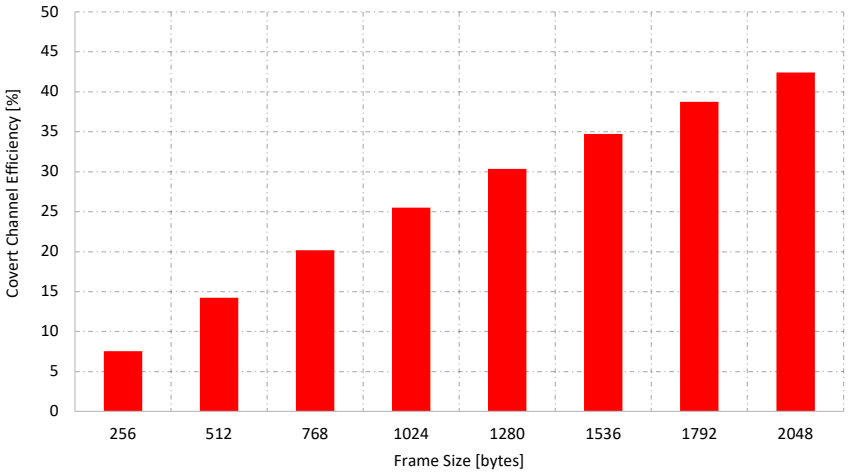


Figure 7.25: Effect of frame size on covert channel efficiency with aggregation disabled

the average delay but also introduce greater variability in transmission timing.

Figure 7.28 illustrates the covert channel throughput as the number of competing stations increases. The results indicate a decline in throughput across all channels, reflecting the overall trend in throughput. With only one competing station, the throughput reduces to about half that of the isolated scenario. As more stations are added, the throughput of all covert channels decreases significantly. This decline is attributed to increased contention for channel access, resulting in frequent collisions, a higher probability of longer backoff periods, and fewer transmission opportunities. Consequently, this results in fewer frames being transmitted per unit of time, directly reducing the number of covert symbols that can be sent. The results demonstrate that covert throughput is highly sensitive to medium contention, and all covert channels experience a substantial reduction in throughput as the number of stations rises, stabilizing at very low levels beyond 10 to 15 stations.

Figure 7.29 presents a two-dimensional analysis of covert channel throughput, showing

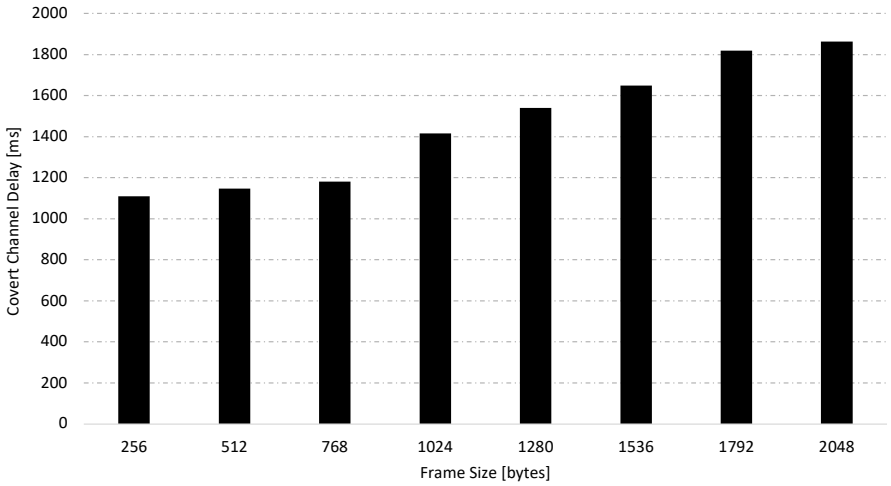


Figure 7.26: Effect of frame size on covert channel delay with aggregation disabled

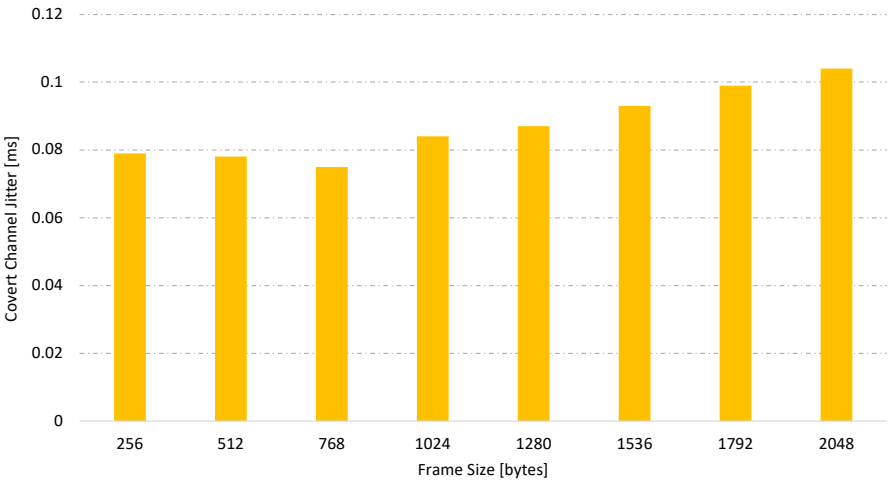


Figure 7.27: Effect of frame size on covert channel jitter with aggregation disabled

the relationship between frame size and the number of competing stations. The results reveal several important observations. Throughput consistently decreases as the number of competing stations increases, regardless of the frame size, although the difference becomes minimal over time. The decline is more pronounced when using larger frame sizes, as they require more airtime for transmission, resulting in fewer frames being sent compared to smaller sizes. However, a key finding is that once there are 10 or more stations, the impact of frame size begins to diminish, as the differences among various frame sizes become less pronounced.

Figure 7.30 illustrates the efficiency of the covert channel. Although the covert channel generates a large number of frames to encode the secret message, the results show an apparent decrease in efficiency as the number of competing stations increases. When a single station operates at a frame efficiency below 50%, adding just one more station drastically reduces the efficiency to below 16%. As contention increases, with 5 to 20 stations competing, the efficiency continues to decline, reaching values close to 12%. This decrease can be attributed to the rise in collisions and the longer backoff durations that occur due to heightened competition for access to the transmission channel. The

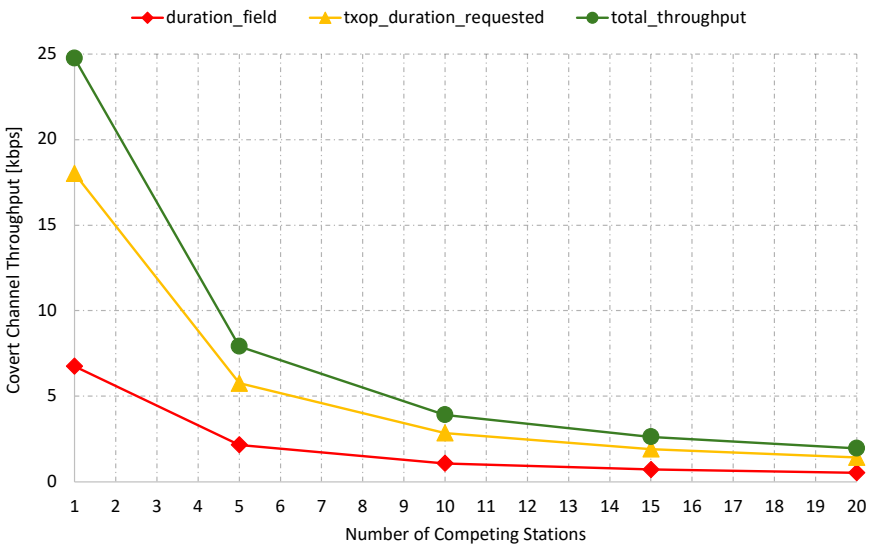


Figure 7.28: Impact of channel contention on covert channel throughput with aggregation disabled

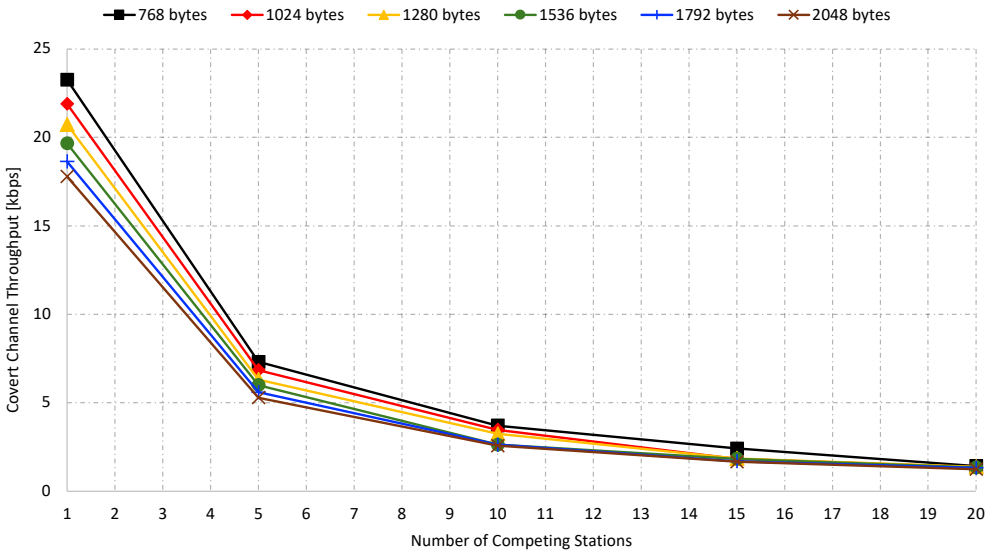


Figure 7.29: Impact of channel contention and frame size on covert channel throughput with aggregation disabled

results also indicate that while generating small frames (e.g., 512 bytes) leads to higher throughput, it comes at a cost.

As illustrated in Figure 7.31, the delay in the covert channel increases as the number of competing stations rises. This increase in delay is primarily due to heightened contention for access to the wireless medium. With more stations vying for channel access, each one experiences longer waiting times before it can transmit. This situation arises from factors such as channel sensing, backoff procedures, and a higher frequency of collisions. Consequently, the accumulated waiting time results in longer delays between the moment a covert message is ready to be sent and the actual time of transmission, with backoff intervals and queuing effects being the most significant contributors.

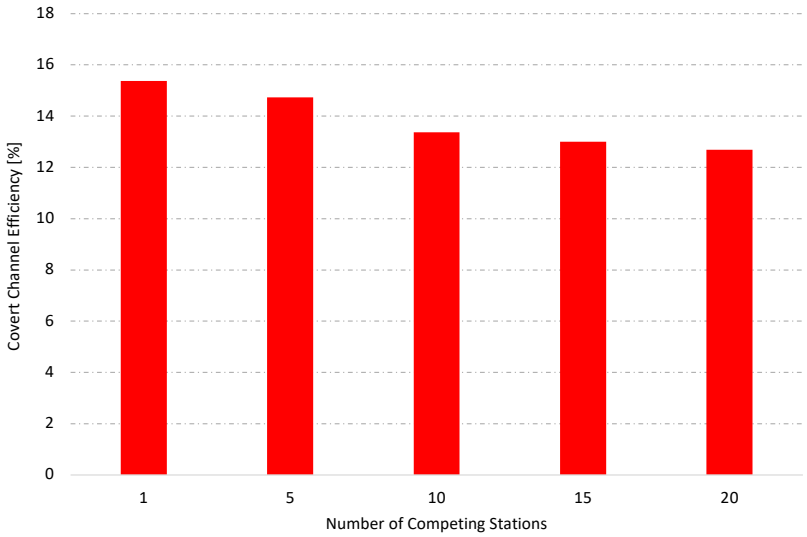


Figure 7.30: Impact of channel contention on covert channel efficiency with aggregation disabled

Additionally, Figure 7.32 depicts the corresponding jitter in the covert channel, which also increases as more stations compete for access. Jitter gauges the variability in transmission timing; as the channel becomes more congested, not only do transmissions face delays, but the intervals between them also become increasingly erratic. This irregularity is a direct result of contention; as competition for channel access grows, the delay can vary significantly from one covert message to the next, leading to a natural increase in jitter alongside the delays.

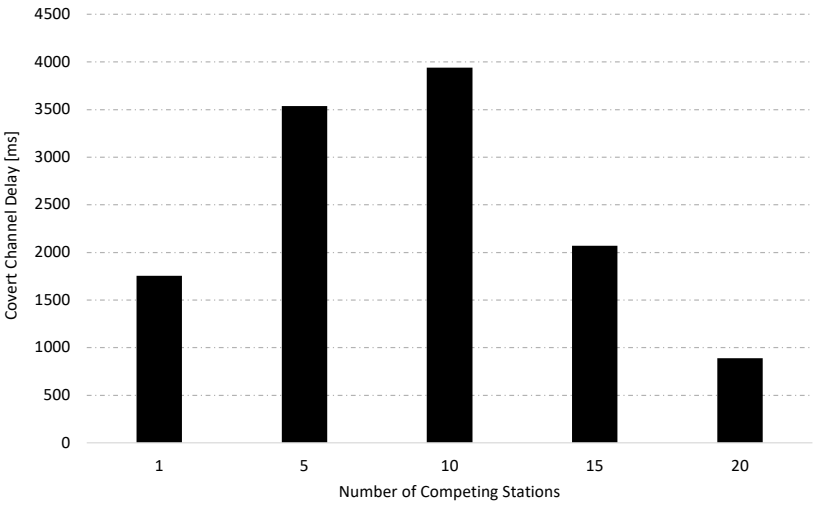


Figure 7.31: Impact of channel contention on covert channel delay with aggregation disabled

As our final measurement, we decided to examine the impact of the offered load from the STA on the covert channel in order to determine the optimal throughput for transmission while facing contention. As shown in Figure 7.33 , we observe that the maximum throughput is about 45 kbps, with a maximum capacity of 25 Mbps. The results indicate that, regardless of contention, 25 Mbps is the optimal throughput value

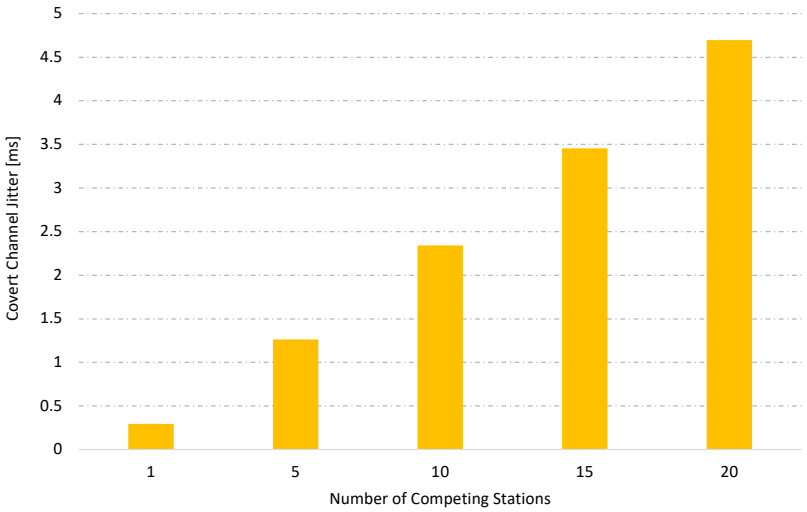


Figure 7.32: Impact of channel contention on covert channel jitter with aggregation disabled

to saturate the network. However, the covert throughput is still affected by contention, achieving rates of 24 kbps, 7.8 kbps, and 4 kbps when facing 1, 5, and 10 stations, respectively. This represents a significant gain compared to similar setups; in the presence of 10 stations, the total throughput registered was only 681 bps.

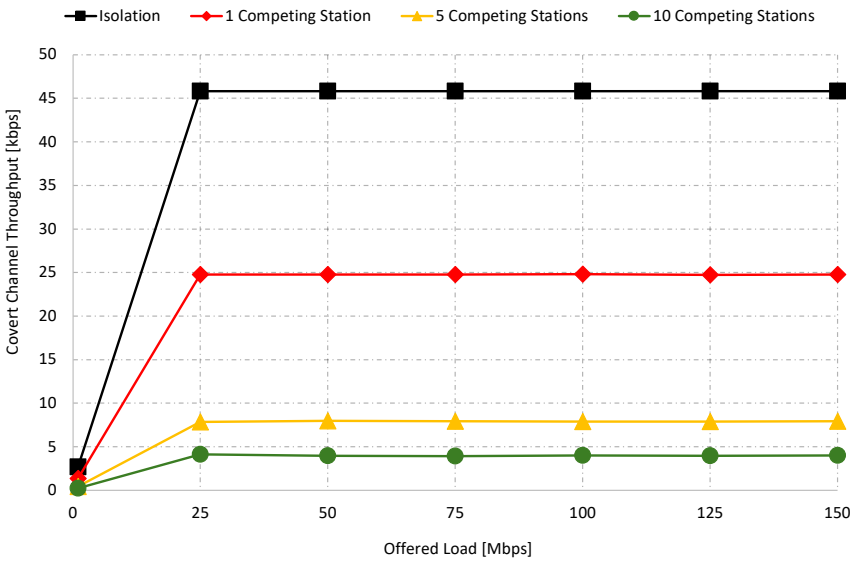


Figure 7.33: Impact of increasing the offered load of the covert STA facing contention over the overall channel throughput with aggregation disabled

7.5 Discussion of results

StegoHybrid is an efficient covert channel mechanism that integrates three storage-based covert channels to increase throughput and improve resistance to steganalysis in Wi-Fi networks. Its novelty lies in distributing the covert embedding across multiple protocol fields and applying a tailored encoding scheme for each channel, thereby enhancing transparency and robustness. Our results demonstrate that up to three covert channels can be combined within a single A-MPDU transmission to encode secret messages using different strategies: the aggregation layout, the Duration/ID field in the MAC header, and the TXOP Duration Requested field in QoS Data frames. From this analysis, we conclude that:

- The bandwidth of the covert channel based on the aggregation layout is proportional, on a logarithmic scale, to the product of the number of MSDUs and MPDUs per A-MPDU. The capacity of the Duration/ID-based channel is limited to three bits per A-MPDU. In contrast, the TXOP Duration Requested channel offers a higher capacity of 8 bits per MPDU within an A-MPDU, except for the last MPDU, which is usually reserved for resetting the request.

- Frame size significantly impacts performance, particularly benefiting the TXOP Duration Requested. Specific frame sizes can result in a higher number of MPDUs per A-MPDU, thereby providing more opportunities to encode information. On the other hand, the Duration/ID-based channel is less affected by frame size because it operates at the A-MPDU level. Regarding the aggregation layout channel, while larger frames can decrease the frequency of transmissions, they may also result in a lower product of MSDUs and MPDUs, thereby limiting capacity.

- The number of competing stations is a more critical factor in degrading the covert channel than the frame size itself. As more stations compete for the medium, the likelihood of collisions increases, along with transmission delays. This results in a decline in frame efficiency, ultimately reducing throughput and increasing both delay and jitter.

- Increasing the maximum size of the A-MSDU, while keeping the size of the A-MPDU fixed, is a more effective strategy for improving the throughput of the covert channel. This configuration benefits both the aggregation layout and the TXOP Duration Requested channels by allowing for more MPDUs to accommodate additional MSDUs. Furthermore, larger A-MSDU sizes lead to more stable frame efficiency, as well as reduced delay and jitter, compared to merely increasing the maximum A-MPDU size.

- A covert channel that operates without frame aggregation achieves higher throughput than one that uses frame aggregation, even when competing stations and larger frame sizes are present. However, this performance advantage has some trade-offs: it results in lower frame efficiency and increased delay (although with shorter jitter in isolation). These drawbacks arise from the necessity of transmitting a greater number of individual MPDUs to maintain high throughput, resulting in increased overhead and variability in frame transmission.

Table 7.4 summarizes the simulation results, reporting the minimum and maximum values observed for the key performance metrics: throughput, efficiency, delay, and jitter. When specific values are not listed, the default configuration from Table 7.2 is assumed. In addition, Tables 7.4b and 7.4a compare scenarios with and without frame aggregation, showing a clear improvement in covert channel throughput when the aggregation covert channel is disabled.

Table 7.4: Summary of covert channel performance metrics: (a) with frame aggregation and (b) without frame aggregation.

(a) With frame aggregation

Parameter	**Metric**	**Minimum**	**Maximum**
Frame size	Throughput [kbps]	10.1	12.1
	Efficiency [%]	99.0	100.0
	Delay [ms]	3.6	3.7
Number of stations	Throughput [kbps]	0.3	5.0
	Efficiency [%]	52.0	88.0
	Delay [ms]	4.6	38.4
	Jitter [ms]	0.1	1.4
A-MSDU max. size	Throughput [kbps]	6.4	41.7
	Efficiency [%]	60.8	88.6
	Delay [ms]	18.4	30.2
	Jitter [ms]	0.9	21.8
A-MPDU max. size	Throughput [kbps]	8.0	17.8
	Efficiency [%]	60.4	89.4
	Delay [ms]	1.2	21.4
	Jitter [ms]	0.5	21.4

(b) Without frame aggregation

Parameter	**Metric**	**Minimum**	**Maximum**
Frame size	Throughput [kbps]	34.1	48.5
	Efficiency [%]	7.5	42.4
	Delay [ms]	1110	1863
	Jitter [ms]	0.75	0.104
Number of stations	Throughput [kbps]	24.7	0.531
	Efficiency [%]	12.6	15.3
	Delay [ms]	891	3939
	Jitter [ms]	0.294	4.6

8 Conclusions

8.1 Thesis contributions

This thesis represents a comprehensive and original contribution to the field of steganography in IEEE 802.11 networks. The key accomplishments and contributions are as follows:

- Introduces a well-structured taxonomic classification of existing covert channels (Figure 3.1), organizing them into distinct categories based on shared characteristics and mechanisms. This classification encompasses the most recent covert channel techniques to date and provides a clear and systematic overview of the 802.11 covert communication landscape. It serves as a valuable reference for understanding the historical evolution, current state of the art, and potential future directions in this field, including considerations of countermeasures.
- The thesis proposes four novel covert channel techniques, each designed to exploit specific aspects of the IEEE 802.11 protocols:
  - A covert channel that leverages the optional MSB in the Supported Rates element. This method is particularly suited for untrusted environments, achieving a channel capacity of 8 bits per frame by encoding information in a way that appears as a legitimate station capability advertisement.
  - A MAC address-based covert channel that exploits disposable randomized MAC addresses. With a capacity of up to 48 bits, this method offers high covertness by blending covert messages with legitimate probe requests commonly used during network scanning.
  - A timing-based covert channel that encodes information using the apparent randomness of the DCF backoff mechanism. This channel transmits 1 bit per frame and remains highly covert due to its alignment with the normal timing behavior of 802.11 stations.
  - A hybrid covert channel that combines three components: Duration/ID, TXOP Duration Requested, and frame aggregation layout. This approach is versatile and adaptable, functioning across both QoS and non-QoS scenarios, with or without aggregation. Each component contributes to the overall throughput and enhances the stealth and flexibility of the channel.

This thesis aims to highlight its contributions to the field of IEEE 802.11 network steganography by focusing on the improvements and innovations achieved, particularly in terms of data rate, transparency, and resistance to steganalysis. To ensure a fair comparison, we first evaluate StegoRates and StegoMAC against existing covert channels that utilize similar message transmission strategies, specifically those that also embed covert data in MAC header fields. In Table 8.1, we provide an overview of the covert channels, including their strategies, transparency, resistance to steganalysis, and the bandwidth or throughput reported by the authors. StegoRates offers a maximum capacity of 255 bytes, corresponding to the theoretical size of the Extended Supported Rates field. StegoMAC encodes 48 bits and offers high transparency and strong resistance to steganalysis, aligning with the standard recommendation for MAC address randomization, a widely adopted

practice for enhancing security. Both covert channels outperform their counterparts in terms of bandwidth while seamlessly blending into modern standard-compliant behavior. They avoid using reserved or deprecated fields, thereby reducing the risk of detection. Unlike the method [63], which remains in a conceptual stage, StegoRates and StegoMAC have been fully implemented and evaluated.

Table 8.1: Existing storage covert channels for comparative analysis

Ref.	**Year**	**Overview**	**Throughput**
StegoRates	2023	Conceals information by encoding it into optional or mandatory supported rates. This approach does not interfere with regular network operations, as supported rates are a standard component of the Wi-Fi scanning process. However, the covert channel may raise suspicion if subsequent Probe Requests are transmitted too frequently or exhibit constant alterations	up to 255 bits/frame
StegoMAC	2023	encodes secret data using disposable MAC addresses. This method offers high transparency, as MAC address randomization is a widely recommended and commonly adopted practice. It is also challenging to detect, as randomized MAC addresses are often treated as temporary identifiers used solely for network scanning, even when generated at a high frequency.	48 bits/frame
[57]	2017	Encodes data in the last 4 LSBs of SSIDs in Probe Requests, offering high transparency by leveraging standard active scanning behavior. Still, it is moderately detectable as frequent or simultaneous searches for multiple or unfamiliar SSIDs, especially those not advertised in the vicinity, can be flagged as anomalies.	at least 4 bits/frame
[69]	2014	This covert channel embeds data in the QoS capability and QoS Control fields of the 802.11e protocol, maintaining moderate transparency by leveraging standard traffic structures, but shows limited resistance to steganalysis due to its reliance on reserved field values and unusually frequent association/reassociation requests.	8 bits/frame
[68]	2012	This covert channel encodes data by altering the 2-bit Protocol Version field in CTS and ACK frames, offering limited transparency due to the use of a reserved field, and low resistance to steganalysis as deviations from the standard PV value (00) are easily detectable through frame inspection.	2 bits/frame
[63]	2011	Encodes data in the SSID and Information Elements of Probe Requests via the Windows WiFi API, maintaining high transparency by mimicking standard network scanning behavior, though it offers moderate resistance to steganalysis, as detection is feasible by analyzing anomalous probe frequency and SSID patterns.	38 bytes/frame
[58]	2008	Embed covert messages within the last 4 bits of the 802.11 header sequence number field. This approach preserves the core functionality of the communication protocol by avoiding modifications to critical frame components. While it ensures minimal disruption to standard traffic, it may interfere with network diagnostics. The channel offers limited resistance to steganalysis, as irregularities in the predictable sequence number pattern can raise suspicion.	4 bits/frame
[58]	2008	Encodes hidden data in the Initialization Vector (IV) field of WEP-encrypted frames, achieving high transparency by leveraging standard random IV generation but offering limited resistance to steganalysis due to WEP's inherent vulnerabilities.	3 bytes/frame

Furthermore, Table 8.2 presents a comparative analysis of timing-based covert channels, particularly those that use DCF for comparison with StegoBackoff. Among these, StegoBackoff achieves a maximum throughput of 4.8 kbps using IEEE 802.11ax with a high MCS index. This performance could be further improved by incorporating features such as MIMO and additional spatial streams. StegoBackoff demonstrates the highest throughput among its peers. Additionally, unlike other methods, StegoBackoff does not always modify the backoff value. When it does, it alters it by only a single slot, minimiz-

ing its impact on normal network behavior and improving its transparency and resistance to detection.

Table 8.2: Existing timing, DCF-based covert channels for comparative analysis

Ref.	Year	Overview	Throughput
[86]	2024	StegoDCF - Encodes covert data in the parity of the backoff value and the last three bits of the duration/ID field in MAC frames. The manipulation causes a negligible duration shift (max 7 µs), which remains below the SIFS, preserving network timing. Both timing fields are altered within normal ranges, offering high transparency and resistance to detection.	144 kbps
StegoBackoff	2024	A covert communication channel designed for smart grid environments, encoding data via the parity of random backoff values in the IEEE 802.11 DCF procedure. Operates transparently by aligning with normal backoff variability and introducing additional slots only when necessary to avoid performance impact. Detection is challenging as it selectively alters backoff values while maintaining randomness, allowing covert symbols to blend in with standard traffic.	4.8 kbps
[77]	2016	Uses free time intervals between packets to encode ternary symbols. The sender and receiver estimate the distribution of these intervals and classify them into three subsets. The method adapts to varying network conditions and is difficult to detect due to its strong resemblance to regular DCF timing.	1800 bps
[74]	2015	Combines random backoff intervals of DCF and empirical modeling (EMD) to construct a timing-based covert channel. Hidden data is encoded in the timing of packet transmissions, with backoff values matching observed traffic distributions. This approach maintains a high data rate and stealth by blending seamlessly with legitimate traffic patterns.	2135 bps
[75]	2013	Exploits DCF's inherent timing variability to encode covert messages within manipulated backoff intervals. It operates with high transparency, mimicking legitimate station behavior and maintaining consistent network performance. Covert values are selected to blend with natural traffic patterns, making the channel indistinguishable from regular activity and hard to detect using standard tools.	2.28 kbps
[73]	2011	Covert DCF - Hides covert messages in the random backoff intervals of the DCF by manipulating CW values. While it allows discrete data transmission, it can introduce unfairness—stations with lower CW values may dominate the channel. Detection is challenging due to the natural randomness of backoff values, further obscured by using an empirical distribution and mixing covert and normal symbols.	1800 bps

Although compared to the earlier hybrid implementation in [87], which achieves a throughput of approximately 248 kbps, StegoHybrid achieves around 45 kbps. However, StegoHybrid offers the advantage of using a variable aggregation layout, making it difficult for observers to determine which aggregation configuration is in use, as it forms part of the covert communication. Additionally, StegoHybrid leverages TXOP, introducing an extra layer of complexity. These features make it a promising candidate for environments that require high-security covert channels.

8.2 Future research directions

This research has identified a significant gap between the increasing sophistication of covert communication techniques and the *lack of effective countermeasures*. Most existing mitigation strategies focus on wired LANs, IoT environments, or alternative wireless technologies and do not adequately address covert channels in current Wi-Fi networks. This highlights an urgent need for the development of targeted detection and mitigation

mechanisms for 802.11-based covert communication. With advances in machine learning, mathematical modeling, and network behavior analysis, there is considerable potential to design intelligent countermeasures that can identify anomalies indicative of covert activity. By detecting spatial, temporal, and behavioral irregularities, these countermeasures could flag the use of covert channels, whether for benign or malicious purposes. The dual-use nature of covert channels further underscores the importance of maintaining defensive capabilities to prevent their abuse in hostile situations.

A notable observation is that many existing covert channels exhibit a structurally uniform nature, either based purely on storage or timing. Such channels are inherently more vulnerable to detection due to their repetitive and predictable characteristics. With the advancement of detection mechanisms, future designs of covert channels must adapt accordingly. A promising avenue for development is the *creation of hybrid covert channels* that integrate multiple encoding techniques (such as timing, storage, and protocol manipulation) to obfuscate the embedded message and minimize the risk of complete detection.

Specifically, for the proposed channels StegoRates and StegoMAC, which utilize manipulation of probe request frames, further enhancements are necessary to improve their stealthiness. A crucial future direction involves analyzing *real-world traffic patterns of probe requests*, including generation timing, Information Elements (IEs), and MAC address randomization, to emulate them more accurately. By adapting covert transmissions to mimic legitimate behavior, these channels can become less distinguishable from regular background traffic, thereby enhancing their resilience to detection techniques.

8.3 Author's publications

The covert channels presented in the thesis are part of a set of publications by the author and his supervisor:

- Teca, G., & Natkaniec, M. (2023). An IEEE 802.11 MAC Layer Covert Channel Based On Supported Rates. International Journal of Electronics and Telecommunications, 69(2), 293–299.
- Teca, G.; Natkaniec, M. A Novel Covert Channel for IEEE 802.11 Networks Utilizing MAC Address Randomization. Appl. Sci. 2023, 13, 8000.
- Teca, G.; Natkaniec, M. StegoBackoff: Creating a Covert Channel in Smart Grids Using the Backoff Procedure of IEEE 802.11 Networks. Energies 2024, 17, 716.

References

[1] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std 802.11™-2024, Published: April 28, 2025, IEEE, Apr. 2025.

[2] Cisco, *Cisco annual internet report (2018–2023): White paper*, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, [Accessed: 2025-09-24], Mar. 2020.

[3] D. Clark, K. Pogran, and D. Reed, "An introduction to local area networks", *Proceedings of the IEEE*, vol. 66, pp. 1497–1517, Dec. 1978. DOI: 10.1109/PROC.1978.11152.

[4] D.-R. Berte, "Defining the IoT", *Proceedings of the International Conference on Business Excellence*, vol. 12, pp. 118–128, May 2018. DOI: 10.2478/picbe-2018-0013.

[5] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A Review of Smart Homes—Past, Present, and Future", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012. DOI: 10.1109/TSMCC.2012.2189204.

[6] J. Latvakoski, A. Iivari, P. Vitic, *et al.*, "A Survey on M2M Service Networks", *Computers*, vol. 3, no. 4, pp. 130–173, 2014, ISSN: 2073-431X.

[7] M. Islam and S. Jin, "An overview research on wireless communication network", *Networks*, vol. 5, no. 1, pp. 19–28, 2019.

[8] R. Muthalagu and S. Sanjay, "Evil Twin Attack Mitigation Techniques in 802.11 Networks", *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021. DOI: 10.14569/IJACSA.2021.0120605. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2021.0120605.

[9] P. Anu and S. Vimala, "A Survey on Sniffing Attacks on Computer Networks", in *2017 International Conference on Intelligent Computing and Control (I2C2)*, Jun. 2017, pp. 1–5. DOI: 10.1109/I2C2.2017.8321914.

[10] M. Vondráček, J. Pluskal, and O. Ryšavý, "Wifimitm: Automated man-in-the-middle attack against wi-fi networks",

[11] A. Arora, *Preventing wireless deauthentication attacks over 802.11 Networks*, 2018. arXiv: 1901.07301 [cs.CR]. [Online]. Available: https://arxiv.org/abs/1901.07301.

[12] J. Bellardo and S. Savage, "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions", in *12th USENIX Security Symposium (USENIX Security '03)*, Washington, D.C.: USENIX Association, 2003. [Online]. Available: https://www.usenix.org/conference/12th-usenix-security-symposium/80211-denial-service-attacks-real-vulnerabilities-and.

[13] T. Ghaleb, "Wireless/Website Traffic Analysis & Fingerprinting: A Survey of Attacking Techniques and Countermeasures", in *2015 International Conference on Cloud Computing (ICCC)*, Apr. 2015. DOI: 10.1109/CLOUDCOMP.2015.7149665.

[14] Wiktionary contributors, *Steganography*, https://en.wiktionary.org/wiki/steganography, Accessed: 2025-09-24, 2025.

[15] D. Kahn, *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. New York: Macmillan, 1967, pp. 57, 120.

[16] F. L. Bauer, *Decrypted Secrets: Methods and Maxims of Cryptology*. Berlin: Springer, 2002, p. 10.

[17] A.-C. Onwutalobi, "Overview of Cryptography", *SSRN Electronic Journal*, Jan. 2011. DOI: 10.2139/ssrn.2741776.

[18] T. Hwang, C. Yang, G. Wu, S. Li, and G. Y. Li, "OFDM and its wireless applications: A survey", *IEEE transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1673–1694, 2008.

[19] MathWorks, *Non-HT PPDU Structure*, https://www.mathworks.com/help/wlan/gs/non-ht-ppdu-structure.html, Accessed: 2025-09-24.

[20] J. Oetting, "A Comparison of Modulation Techniques for Digital Radio", *IEEE Transactions on Communications*, vol. 27, no. 12, pp. 1752–1762, 1979. DOI: 10.1109/TCOM.1979.1094370.

[21] D. E. Capano and C. Vavra, "Understanding modulation and coding schemes: Industrial wireless tutorials: Modulation and coding schemes (MCS) are used to determine the data rate of a wireless connection using high-throughput orthogonal frequency division multiplexing (HT-OFDM)", *Control Engineering*, vol. 61, no. 12, pp. 26–27, 2014.

[22] mcsindex.com, *MCS Index Table, Modulation and Coding Scheme Index 11n, 11ac, and 11ax*, Available at: https://mcsindex.com. [Accessed: 4 July 2025].

[23] B. Zhang, Y. Wang, W. Wang, and Y. Tian, "On the downlink throughput capacity of hybrid wireless networks with mimo", *IEEE Access*, vol. 5, pp. 26 086–26 091, 2017. DOI: 10.1109/ACCESS.2017.2777527.

[24] W. Soyinka, "Wireless Network Administration: A Beginner's Guide", in McGraw-Hill, 2010, pp. 82–88.

[25] W. Soyinka, "Wireless Network Administration: A Beginner's Guide", in McGraw-Hill, 2010, pp. 152–175.

[26] *IEEE Standard for Information Technology–Local and Metropolitan Area Networks–Specific Requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE Std 802.11e-2005, Amendment to IEEE Std 802.11-1999 (Reaff 2003), IEEE, 2005, pp. 1–212.

[27] W. Jepsen, *Cyclic Redundancy Checks and Error Detection*, 2022. arXiv: 2205.11344 [cs.NI]. [Online]. Available: https://arxiv.org/abs/2205.11344.

[28] M. Cunche, "I know your mac address: Targeted tracking of individual using wi-fi", *Journal of Computer Virology and Hacking Techniques*, vol. 10, pp. 219–227, Nov. 2013. DOI: 10.1007/s11416-013-0196-1.

[29] J. Freudiger, "How talkative is your mobile device? an experimental study of Wi-Fi probe requests", in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec '15*, New York, NY, USA: ACM, 2015. DOI: 10.1145/2766498.2766517.

[30] M. Cunche, M. A. Kaafar, and R. Boreli, "Linking wireless devices using information contained in Wi-Fi probe requests", *Pervasive and Mobile Computing*, vol. 11, pp. 56–69, 2014. DOI: 10.1016/j.pmcj.2013.02.003.

[31] L. Schauer, "Wi-fi tracking threatens users' privacy in fingerprinting techniques", in *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation* (Intelligent Data-Centric Systems), J. Conesa, Á. Pérez-Navarro, J. Torres-Sospedra, and R. Montoliu, Eds., Intelligent Data-Centric Systems. Academic Press, 2019, pp. 21–43. DOI: 10.1016/B978-0-12-815905-3.00002-3.

[32] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa, "Signals from the crowd: Uncovering social relationships through smartphone probes", in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ACM, 2013, pp. 265–276.

[33] L. Fabre, C. Bayart, P. Bonnel, and N. Mony, "The potential of wi-fi data to estimate bus passenger mobility", *Technological Forecasting and Social Change*, vol. 192, p. 122 509, 2023. DOI: https://doi.org/10.1016/j.techfore.2023.122509.

[34] I. Moser, C. McCarthy, P. Jayaraman, *et al.*, "A methodology for empirically evaluating passenger counting technologies in public transport", in *Australasian Transport Research Forum 2019 Proceedings*, Australasian Transport Research Forum, 2019.

[35] A. Hidayat, S. Terabe, and H. Yaginuma, "Mapping of mac address with moving wifi scanner", *International Journal of Artificial Intelligence*, vol. 1, pp. 34–40, 2017.

[36] K.-J. Djervbrant and A. Häggström, *A Study on Fingerprinting of Locally Assigned MAC-Addresses*, 2019.

[37] Apple Inc., *Use private Wi-Fi addresses on Apple devices*, https://support.apple.com/en-in/102509, Accessed: 2025-09-29.

[38] E. Grumbach, *iwlwifi: mvm: support random MAC address for scanning*, Git commit, Commit hash: effd05a. Available at: https://github.com/torvalds/linux/commit/effd05ac479b80641835f9126bbe93146686c2b8 [Accessed: 2025-09-24], 2014.

[39] *"Android 6.0. (Marshmallow)". Android Developers*, Available online: https://developer.android.com/about/versions/marshmallow/android-6.0-changes, [Accessed: 2025-09-24].

[40] C. Huitema, *Experience with MAC address randomization on Windows 10*, Available online: https://www.ietf.org/proceedings/93/slides/slides-93-intarea-5.pdf, Accessed: 2025-09-24, 2015.

[41] M. Cunche and C. Matte, "On Wi-Fi tracking and the pitfalls of MAC address randomization", in *National Internet of Things Day. New challenges of the Internet of Things: Human-Computer Interaction and Human Factors*, Sep. 2016.

[42] I. Vasilevski, D. Blazhevski, V. Pachovski, and I. Stojmenovska, "Five years later: How effective is the mac randomization in practice? the no-at-all attack", in *ICT Innovations 2019. Big Data Processing and Mining*, S. Gievska and G. Madjarov, Eds., Cham: Springer International Publishing, 2019, pp. 52–64, ISBN: 978-3-030-33110-8.

[43] J. Martin, T. Mayberry, C. Donahue, *et al.*, "A Study of MAC Address Randomization in Mobile Devices and When it Fails", *Proceedings on Privacy Enhancing Technologies*, vol. 4, pp. 268–286, 2017.

[44] E. Fenske, D. Brown, J. Martin, T. Mayberry, P. Ryan, and E. Rye, "Three Years Later: A Study of MAC Address Randomization In Mobile Devices And When It Succeeds", *Proceedings on Privacy Enhancing Technologies*, vol. 2021, pp. 164–181, Jul. 2021.

[45] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000. DOI: 10.1109/49.840210.

[46] M. Natkaniec and A. R. Pach, "An analysis of the backoff mechanism used in IEEE 802.11 networks", in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2000, pp. 444–449. DOI: 10.1109/ISCC.2000.860664.

[47] S. Rasheed, K. Masnoon, N. Thanthry, and R. Pendse, "PCF vs DCF: a performance comparison, year=2004", in *Thirty-Sixth Southeastern Symposium on System Theory, 2004. Proceedings of the*, pp. 215–219. DOI: 10.1109/SSST.2004.1295651.

[48] H. Wu, X. Wang, Q. Zhang, and X. Shen, "IEEE 802.11e Enhanced Distributed Channel Access (EDCA) Throughput Analysis", in *2006 IEEE International Conference on Communications*, vol. 1, 2006, pp. 223–228. DOI: 10.1109/ICC.2006.254731.

[49] Y.-C. Liu and G. Wise, "Performance of a CSMA/CD Protocol for Local Area Networks", *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 6, pp. 948–955, 1987. DOI: 10.1109/JSAC.1987.1146621.

[50] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans", in *Proceedings of PIMRC '96 - 7th International Symposium on Personal, Indoor, and Mobile Communications*, vol. 2, 1996, 392–396 vol.2. DOI: 10.1109/PIMRC.1996.567423.

[51] L. Wang, K. Wu, and M. Hamdi, "Combating hidden and exposed terminal problems in wireless networks", *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 4204–4213, 2012.

[52] Y. Kim, S. Choi, K. Jang, and H. Hwang, "Throughput enhancement of IEEE 802.11 WLAN via frame aggregation", in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, vol. 4, 2004, 3030–3034 Vol. 4. DOI: 10.1109/VETECF.2004.1400617.

[53] J. Kolap, S. Krishnan, and N. Shaha, "Frame aggregation mechanism for high-throughput 802.11 n wlans", *International Journal of Wireless & Mobile Networks*, vol. 4, no. 3, p. 141, 2012.

[54] B. S. Kim, H. Y. Hwang, and D. K. Sung, "Effect of Frame Aggregation on the Throughput Performance of IEEE 802.11n", in *2008 IEEE Wireless Communications and Networking Conference*, 2008, pp. 1740–1744. DOI: 10.1109/WCNC.2008.310.

[55] M. Yazid, L. Bouallouche-Medjkoune, and D. Aı"ssani, "Performance Study of Frame Aggregation Mechanisms in the New Generation WiFi.", in *VECoS*, 2016, pp. 85–92.

[56] K. Szczypiorski, "HICCUPS: Hidden Communication System for Corrupted Networks", in *The Tenth International Multi-Conference on Advanced Computer Systems*, Informa, 2003.

[57] K. Sawicki and Z. M. Piotrowski, "Two-Way Complex Steganographic System for Authentication and Authorization in IEEE 802.11 Wireless Networks", *Elektronika - Konstrukcje, Technologie, Zastosowania*, vol. 58, pp. 22–26, 2017. DOI: 10.15199/13.2017.1.4. [Online]. Available: https://doi.org/10.15199/13.2017.1.4.

[58] L. Frikha, Z. Trabelsi, and W. El-Hajj, "Implementation of a Covert Channel in the 802.11 Header", *2008 International Wireless Communications and Mobile Computing Conference*, pp. 594–599, 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:17094400.

[59] Y. Inoue, K. Saitoh, T. Sakata, M. Morikura, and H. Matsue, "A Study on the Rate Switching Algorithm for IEEE 802.11 Wireless LANs", *IEEJ Transactions on Electronics, Information and Systems*, vol. 124, no. 1, pp. 33–40, 2004. DOI: 10.1541/ieejeiss.124.33.

[60] T. E. Calhoun, R. Newman, and R. Beyah, "Authentication in 802.11 LANs Using a Covert Side Channel", in *2009 IEEE International Conference on Communications*, 2009, pp. 1–6. DOI: 10.1109/ICC.2009.5198769.

[61] T. E. Calhoun Jr, X. Cao, Y. Li, and R. Beyah, "An 802.11 MAC layer covert channel", *Wireless Communications and Mobile Computing*, vol. 12, no. 5, pp. 393–405, 2012. DOI: https://doi.org/10.1002/wcm.969. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcm.969. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcm.969.

[62] G. Teca and M. Natkaniec, "An IEEE 802.11 MAC Layer Covert Channel Based On Supported Rates", *International Journal of Electronics and Telecommunications*, vol. vol. 69, no. No 2, pp. 293–299, 2023. DOI: 10.24425/ijet.2023.144364. [Online]. Available: http://journals.pan.pl/Content/127374/PDF/13_4054_Teca_sk.pdf.

[63] A. Blanco and E. Gutesman, *Abusing the Windows WiFi native API to create a Covert Channel*, https://www.coresecurity.com/sites/default/files/private-files/publications/2016/05/corelabs-hacklu2011-paperCovertChannel.pdf, Apr. 13, 2011.

[64] P. M. B. Harley, M. Tummala, and J. C. McEachen, "High-Throughput Covert Channels in Adaptive Rate Wireless Communication Systems", in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, 2019, pp. 1–7. DOI: 10.23919/ELINFOCOM.2019.8706484.

[65] L. Vicisano, M. J. Handley, J. Gemmell, J. Crowcroft, L. Rizzo, and M. Luby, *Forward Error Correction (FEC) Building Block*, RFC 3452, Dec. 2002. DOI: 10.17487/RFC3452. [Online]. Available: https://www.rfc-editor.org/info/rfc3452.

[66] IEEE, "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, 2021. DOI: 10.1109/IEEESTD.2021.9363693.

[67] M. A. Belhamra and E. M. Souidi, "A Steganogaphic Scheme for MAC-Independent Opportunistic Routing and Encoding (MORE) Protocol", in *International Conference on E-Business and Telecommunication Networks*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:52117476.

[68] R. Goncalves, M. Tummala, and J. Mceachen, "Analysis of a mac layer covert channel in 802.11 networks", *International Journal on Advances in Telecommunications*, vol. 5, no. 3 and 4, 2012.

[69] H. Zhao, "Covert channels in 802.11e wireless networks", in *2014 Wireless Telecommunications Symposium*, 2014, pp. 1–5. DOI: 10.1109/WTS.2014.6834991.

[70] G. Teca and M. Natkaniec, "A Novel Covert Channel for IEEE 802.11 Networks Utilizing MAC Address Randomization", *Applied Sciences*, vol. 13, no. 14, 2023, ISSN: 2076-3417. DOI: 10.3390/app13148000. [Online]. Available: https://www.mdpi.com/2076-3417/13/14/8000.

[71] J. Brunekreef, "Sliding Window Protocols", in *Algebraic Specification of Communication Protocols* (Cambridge Tracts in Theoretical Computer Science), S. Mauw and G. J. Veltink, Eds., Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1993, pp. 71–112. DOI: 10.1017/CBO9780511721625.005.

[72] C. Kraetzer, J. Dittmann, and R. Merkel, "WLAN steganography revisited", in *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, I. Delp Edward J., P. W. Wong, J. Dittmann, and N. D. Memon, Eds., ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6819, Feb. 2008, 681903, p. 681 903. DOI: 10.1117/12.764557.

[73] R. Holloway and R. Beyah, "Covert DCF: A DCF-Based Covert Timing Channel in 802.11 Networks", in *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011, pp. 570–579. DOI: 10.1109/MASS.2011.60.

[74] F. Tahmasbi, N. Moghim, and M. Mahdavi, "Code-based timing Covert channel in IEEE 802.11", in *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2015, pp. 12–17. DOI: 10.1109/ICCKE.2015.7365854.

[75] F. Tahmasbi, N. Moghim, and M. Mahdavi, "Adaptive ternary timing covert channel in IEEE 802.11", *Security and Communication Networks*, vol. 9, no. 16, pp. 3388–3400, 2016. DOI: https://doi.org/10.1002/sec.1545. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.1545. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1545.

[76] G. J. Simmons, "The prisoners' problem and the subliminal channel", in *Advances in Cryptology: Proceedings of Crypto 83*, Springer, 1984, pp. 51–67.

[77] F. Tahmasbi, N. Moghim, and M. Mahdavi, "Adaptive ternary timing covert channel in IEEE 802.11", *Security and Communication Networks*, vol. 9, no. 16, pp. 3388–3400, 2016. DOI: https://doi.org/10.1002/sec.1545. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.1545. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1545.

[78] G. Teca and M. Natkaniec, "StegoBackoff: Creating a Covert Channel in Smart Grids Using the Backoff Procedure of IEEE 802.11 Networks", *Energies*, vol. 17, no. 3, 2024, ISSN: 1996-1073. DOI: 10.3390/en17030716. [Online]. Available: https://www.mdpi.com/1996-1073/17/3/716.

[79] N. Kiyavash, F. Koushanfar, T. P. Coleman, and M. Rodrigues, "A Timing Channel Spyware for the CSMA/CA Protocol", *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 477–487, 2013. DOI: 10.1109/TIFS.2013.2238930.

[80] T. O. Walker and K. D. Fairbanks, "An off-the-shelf, low detectability, low data rate, timing-based covert channel for ieee 802.11 wireless networks", in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2017, pp. 835–840. DOI: 10.1109/CCNC.2017.7983242.

[81] K. Sawicki and Z. Piotrowski, "The proposal of ieee 802.11 network access point authentication mechanism using a covert channel", in *2012 19th International Conference on Microwaves, Radar & Wireless Communications*, vol. 2, 2012, pp. 656–659. DOI: 10.1109/MIKON.2012.6233587.

[82] H. Seong, I. Kim, Y. Jeon, M.-K. Oh, S. Lee, and D. Choi, "Practical Covert Wireless Unidirectional Communication in IEEE 802.11 Environment", *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 1499–1516, 2023. DOI: 10.1109/JIOT.2022.3204987.

[83] S. Son, D. Kwon, S. Lee, Y. Jeon, and Y. Park, "A Robust Covert Channel With Self-Bit Recovery for IEEE 802.11 Networks", *IEEE Internet of Things Journal*, pp. 1–1, 2024. DOI: 10.1109/JIOT.2024.3398579.

[84] A. Smith and B. Jones, "CHAOS: Exploiting Station Time Synchronization in 802.11 Networks", in *Network and Distributed System Security Symposium (NDSS)*, Also known as "CHAOS: Covert Channel via TSF in Wi-Fi Beacon Frames", 2025.

[85] K. Sawicki, G. Bieszczad, and Z. Piotrowski, "StegoFrameOrder—MAC Layer Covert Network Channel for Wireless IEEE 802.11 Networks", *Sensors*, vol. 21, no. 18, 2021, ISSN: 1424-8220. DOI: 10.3390/s21186268. [Online]. Available: https://www.mdpi.com/1424-8220/21/18/6268.

[86] M. Natkaniec and J. Dyrcz, "StegoDCF: A New Covert Channel for Smart Grids Utilizing the Channel Access Procedure in Wi-Fi Networks", *Energies*, vol. 17, no. 9, 2024, ISSN: 1996-1073. DOI: 10.3390/en17092021. [Online]. Available: https://www.mdpi.com/1996-1073/17/9/2021.

[87] M. Natkaniec and P. Kępowicz, "StegoEDCA: An Efficient Covert Channel for Smart Grids Based on IEEE 802.11e Standard", *Energies*, vol. 18, no. 2, 2025, ISSN: 1996-1073. [Online]. Available: https://www.mdpi.com/1996-1073/18/2/330.

[88] P. Zbigniew, S. Krzysztof, B. Mariusz, and G. Piotr, "New Hidden and Secure Data Transmission Method Proposal for Military IEEE 802.11 Networks", in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2010, pp. 179–183. DOI: 10.1109/IIHMSP.2010.52.

[89] B. Zoltak, "VMPC one-way function and stream cipher", in *Fast Software Encryption: 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004. Revised Papers 11*, Springer, 2004, pp. 210–225.

[90] M. Schulz, J. Link, F. Gringoli, and M. Hollick, "Shadow Wi-Fi: Teaching Smartphones to Transmit Raw Signals and to Extract Channel State Information to Implement Practical Covert Channels over Wi-Fi", in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '18, Munich, Germany: Association for Computing Machinery, 2018, pp. 256–268, ISBN: 9781450357203. DOI: 10.1145/3210240.3210333. [Online]. Available: https://doi.org/10.1145/3210240.3210333.

[91] M. Guri, "AIR-FI: Generating Covert Wi-Fi Signals from Air-Gapped Computers", *ArXiv*, vol. abs/2012.06884, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229152889.

[92] M. Aslaner, "Air-gapped networks: the myth and the reality", *Network Security*, vol. 2022, no. 2, 2022.

[93] J. Classen, M. Schulz, and M. Hollick, "Practical covert channels for WiFi systems", in *2015 IEEE Conference on Communications and Network Security (CNS)*, 2015, pp. 209–217. DOI: 10.1109/CNS.2015.7346830.

[94] R. P. Hudhajanto, I. G. P. Astawa, and A. Sudarsono, "Covert Communication in MIMO-OFDM System Using Pseudo Random Location of Fake Subcarriers", *EMITTER International Journal of Engineering Technology*, vol. 4, no. 1, pp. 150–163, Jun. 2016. DOI: 10.24003/emitter.v4i1.58. [Online]. Available: https://emitter2.pens.ac.id/ojs/index.php/emitter/article/view/58.

[95] K. Szczypiorski and W. Mazurczyk, "Hiding Data in OFDM Symbols of IEEE 802.11 Networks", in *2010 International Conference on Multimedia Information Networking and Security*, 2010, pp. 835–840. DOI: 10.1109/MINES.2010.177.

[96] S. Grabski and K. Szczypiorski, "Steganography in OFDM Symbols of Fast IEEE 802.11n Networks", in *2013 IEEE Security and Privacy Workshops*, 2013, pp. 158–164. DOI: 10.1109/SPW.2013.20.

[97] D. Chew, C. Nguyen, S. Berhanu, C. Baumgart, and A. B. Cooper, "Covert Communications through Imperfect Cancellation", in *Proceedings of the 2022 ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec '22, Santa Barbara, CA, USA: Association for Computing Machinery, 2022, pp. 63–68, ISBN: 9781450393553. DOI: 10.1145/3531536.3532959. [Online]. Available: https://doi.org/10.1145/3531536.3532959.

[98] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "Secret Agent Radio: Covert Communication through Dirty Constellations", in *Information Hiding*, M. Kirchner and D. Ghosal, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–175, ISBN: 978-3-642-36373-3.

[99] K. Grzesiak, Z. Piotrowski, and J. M. Kelner, "A Wireless Covert Channel Based on Dirty Constellation with Phase Drift", *Electronics*, vol. 10, no. 6, 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10060647. [Online]. Available: https://www.mdpi.com/2079-9292/10/6/647.

[100] Z. Piotrowski, "Drift Correction Modulation scheme for digital signal processing", *Mathematical and Computer Modelling*, vol. 57, no. 11, pp. 2660–2670, 2013, Information System Security and Performance Modeling and Simulation for Future Mobile Networks, ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2011.09.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717711005644.

[101] P. Cao, W. Liu, G. Liu, X. Ji, J. Zhai, and Y. Dai, "A wireless covert channel based on constellation shaping modulation", *Security and Communication Networks*, vol. 2018, pp. 1–15, 2018.

[102] S. D'oro, F. Restuccia, and T. Melodia, "Hiding Data in Plain Sight: Undetectable Wireless Communications Through Pseudo-Noise Asymmetric Shift Keying", *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1585–1593, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:86719265.

[103] K. Grzesiak, Z. Piotrowski, and J. M. Kelner, "Covert Channel Based on Quasi-Orthogonal Coding", *Electronics*, vol. 12, no. 10, 2023, ISSN: 2079-9292. DOI: 10.3390/electronics12102249. [Online]. Available: https://www.mdpi.com/2079-9292/12/10/2249.

[104] P. Cao, W. Liu, G. Liu, J. Zhai, X. Ji, and Y. Dai, "A Novel Wireless Covert Channel for MIMO System", in *Artificial Intelligence and Security*, X. Sun, J. Wang, and E. Bertino, Eds., Singapore: Springer Singapore, 2020, pp. 351–362, ISBN: 978-981-15-8101-4.

[105] X. Wang, Y. Liu, X. Lu, S. Lv, Z. Shi, and L. Sun, "A survey of covert: A covert uplink transmission scheme for MIMO systems", in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7996863.

[106] W. Li, J. Liao, Y. Qian, X. Zhou, and Y. Lin, "A wireless covert communication system: Antenna coding and achievable rate analysis", in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 438–443. DOI: 10.1109/ICC45855.2022.9838322.

[107] G. Shah and M. Blaze, "Covert Channels through External Interference", in *Workshop on Offensive Technologies*, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:11943669.

[108] S. Zillien and S. Wendzel, "Reconnection-Based Covert Channels in Wireless Networks", in *ICT Systems Security and Privacy Protection*, A. Jøsang, L. Futcher, and J. Hagen, Eds., Cham: Springer International Publishing, 2021, pp. 118–133, ISBN: 978-3-030-78120-0. DOI: 10.1007/978-3-030-78120-0_9.

[109] TP-Link Technologies Co., Ltd., *TP-Link USB Wi-Fi Adapters*, https://www.tp-link.com/us/home-networking/usb-adapter/, Accessed: 2025-09-24.

[110] ASUS, *ASUS Wireless Adapters – WiFi 7 Series*, https://www.asus.com/networking-iot-servers/wifi-7/all-series/filter?Category=Wireless-Adapters, Accessed: 2025-09-24.

[111] morrownr, *USB WiFi Adapter Information and Linux Driver Guide*, https://github.com/morrownr/USB-WiFi/blob/main/README.md, Accessed: 2025-09-24, 2021.

[112] morrownr, *8821au-20210708: Driver for Realtek 8821AU USB WiFi Adapters*, https://github.com/morrownr/8821au-20210708, Accessed: 2025-09-24, 2021.

[113] A. Devices, *ADALM-PLUTO: Evaluation Board Overview*, Accessed: 2025-09-24. [Online]. Available: https://wiki.analog.com/university/tools/pluto.

[114] Raspberry Pi Foundation, *Raspberry Pi Official Website*, https://www.raspberrypi.com/, Accessed: 2025-09-24], 2025.

[115] ns-3 Project, *ns-3 Network Simulator*, Accessed: 2025-09-24. [Online]. Available: https://www.nsnam.org/.

[116] O. Projects, *OPNET Network Simulator*, Accessed: 2025-09-24. [Online]. Available: https://opnetprojects.com/opnet-network-simulator/.

[117] H. Zhao and M. Chen, "Wlan covert timing channel detection", in *2015 Wireless Telecommunications Symposium (WTS)*, 2015, pp. 1–5. DOI: 10.1109/WTS.2015.7117246.

[118] P. Yang, H. Zhao, and Z. Bao, "A probability-model-based approach to detect covert timing channel", in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 1043–1047. DOI: 10.1109/ICInfA.2015.7279440.

[119] N. A. AL-Khulaidi, A. T. Zahary, M. A. Hazaa, and A. A. Nasser, "Covert Channel Detection and Generation Techniques: A Survey", in *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, 2023, pp. 01–09. DOI: 10.1109/eSmarTA59349.2023.10293582.

[120] P. Yang, H. Zhao, and Z. Bao, "A probability-model-based approach to detect covert timing channel", in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 1043–1047. DOI: 10.1109/ICInfA.2015.7279440.

[121] H. Zhao and M. Chen, "Wlan covert timing channel detection", in *2015 Wireless Telecommunications Symposium (WTS)*, 2015, pp. 1–5. DOI: 10.1109/WTS.2015.7117246.

[122] M. A. Elsadig and Y. A. Fadlalla, "Network protocol covert channels: Countermeasures techniques", in *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*, 2017, pp. 1–9. DOI: 10.1109/IEEEGCC.2017.8447997.

[123] M. Chourib, "Detecting selected network covert channels using machine learning", in *2019 International Conference on High Performance Computing & Simulation (HPCS)*, 2019, pp. 582–588. DOI: 10.1109/HPCS48598.2019.9188115.

[124] L. Caviglione, "Trends and challenges in network covert channels countermeasures", *Applied Sciences*, vol. 11, no. 4, 2021, ISSN: 2076-3417. DOI: 10.3390/app11041641. [Online]. Available: https://www.mdpi.com/2076-3417/11/4/1641.

[125] Y. Song, C. Yang, and G. Gu, "Who is Peeping at Your Passwords at Starbucks? — To Catch an Evil Twin Access Point", in *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, 2010, pp. 323–332. DOI: 10.1109/DSN.2010.5544302. [Online]. Available: http://doi.org/10.1109/DSN.2010.5544302.

[126] B. Alotaibi and K. Elleithy, "Rogue access point detection: Taxonomy, challenges, and future directions", *Wireless Personal Communications*, vol. 90, pp. 5021–5028, Oct. 2016. DOI: 10.1007/s11277-016-3390-x.

[127] S. Shetty, M. Song, and L. Ma, "Rogue Access Point Detection by Analyzing Network Traffic Characteristics", in *MILCOM 2007 - IEEE Military Communications Conference*, 2007, pp. 1–7. DOI: 10.1109/MILCOM.2007.4455018. [Online]. Available: http://doi.org/10.1109/MILCOM.2007.4455018.

[128] K. Yogi and Ernastuti, "Analysis of Deauthentication Attack on IEEE 802.11 Connectivity Based on IoT Technology Using External Penetration Test", *Communication and Information Technology (CommIT)*, vol. 14, no. 1, pp. 45–51, 2020.

[129] P. B and J. Nagamalai, "A Review on Various Sniffing Attacks and Its Mitigation Techniques", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 3, pp. 1117–1125, Dec. 2018. DOI: 10.11591/ijeecs.v12.i3.pp1117-1125. [Online]. Available: http://doi.org/10.11591/ijeecs.v12.i3.pp1117-1125.

[130] M. Andersdotter, "Ongoing developments in IEEE 802.11 WLAN standardization: A study group on randomized and changing MAC addresses", in *Proceedings of the Hot Topics in Privacy Enhancing Technologies (HotPETS 2019)*, Stockholm, Sweden, Jul. 2019. [Online]. Available: https://petsymposium.org/2019/files/hotpets/andersdotter-wlan.pdf.

[131] A. B. Bada, "Automatic repeat request (Arq) protocols", *Int J Eng Sci (IJES)*, vol. 6, pp. 64–66, 2017.

List of Abbreviations

A-MPDU	Aggregated MAC Protocol Data Unit
A-MSDU	Aggregate MAC service data units
ACK	Acknowledgment
AID	Association ID
AP	Access Point
API	Application Programming Interface
APT	Advanced Persistent Threats
ARM	Advanced RISC Machine
ARQ	Automatic Repeat Request
BACK	Block Acknowledgment
BCE	Before Common Era
BSK	Binary Shift Keying
BSS	Basic Service Set
BSSID	Basic Service Set ID
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CFO	Carrier Frequency Offset
CRC	Cyclic Redundancy Check
CSI	Channel State Information
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear To Send
DC	Data Carrier
DCF	Distributed Coordination Function
DDR SDRAM	Synchronous Dynamic Random-Access Memory
DIFS	Distributed Inter Frame Space
DMG	Directional Multi-Gigabit
DNN	Deep Neural Networks

DNS	Domain Name System
DoS	Denial-of-Service
EDCA	Enhanced Distributed Channel Access
EHT	Extremely High Throughput
EMD	Exploiting Modification Direction
EOSP	End of Service Period
FCS	Frame Check Sequence
FEC	Forward Error Correction
FPGA	Field-Programmable Gate Array
FSK	Frequency Shift Keying
GI	Guard Interval
GPS	Global Positioning System
HCC	Hybrid Covert Channel
HDCs	Hidden Data Channel
HE	High Efficiency
HICCUPS	Hidden Communication System for Corrupted Networks
HT	High Throughput
HTTP	HyperText Transfer Protocol
IBSS	Independent Basic Service Set ID
IEEE	Institute of Electrical and Electronics Engineers
IEEE RA	IEEE Registration Authority
IEs	Information Elements
IFFT	Inverse Fast Fourier Transform
IoT	Internet of Things
IPD	Interpacket Delay
IPv6	Internet Protocol Version 6
IQ	In-phase and Quadrature
ISI	Inter-Symbol Interference
IV	Initialization Vector

JPEG	Joint Photographic Experts Group
k-NN	k-nearest Neighbors
LANs	Local Area Networks
LSB	Least Significant Bit
LTF	Long Training Field
M2M	Machine-to-machine
MAC	MAC Address
MATLAB	Matrix Laboratory
MCS	Modulation and Coding Scheme
MIC	Message Integrity Code
MIMO	Multiple Input Multiple Output
MitM	Man-in-the-middle
ML	Machine Learning
MORE	MAC-independent Opportunistic Routing & Encoding
MPDU	MAC Protocol Data Unit
MPSK	M-ary Phase Shift Keying
MSB	Most Significant Bit
MSDU	MAC service data units
NADS	Anomaly Detection System
NAV	Network Allocation Vector
NCCT	Network Covert Channel Triangle
NIC	Network Interface Controller
OFDM	Orthogonal Frequency-Division Multiplexing
OPNET	Optimized Network Engineering Tool
OS	Operating System
OTP	One-Time Password
OUI	Organizationally Unique Identifier
PCF	Point Coordination Function
PDU	Protocol Data Unit

PHY	Physical Layer
PLCP	Physical Layer Convergence Protocol
PN-ASK	Pseudo-Noise Amplitude Shift Keying
PPDU	PLCP Protocol Data Unit
PR	Probe Request
PRNG	Pseudo-Random Number Generator
PSDU	PLCP Service Data Unit
PSK	Phase Shift Keying
PV	Protocol Version
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
QSTA	QoS Station
RAP	Rogue Access Point
RC	Rivest Cipher
RF	Radio Frequency
RLNC	Random Linear Network Coding
RSSI	Received Signal Strength Indicator
RTT	Round-Trip Time
SA	Source Address
SC	Sequence Control
SCC	Storage Covert Channels
SCI	Correlation-based Interference
SDR	Software-defined radio
SG	Smart Grid
SHTM	Statistical Hypothesis Testing Method
SIC	Successive Interference Cancellation
SIFS	Short InterFrame Space
SMs	Smart Meters

SN	Sequence Number
SR	Selective Repeat
SSID	Service Set Identifier
STF	Short Training Field
STF-PSK	Phase Shift Keying
SVM	Support Vector Machine
SWP	Sliding Window Protocol
TCC	Timing Covert Channels
TCP/IP	Transmission Control Protocol / Internet Protocol
TI	Transmission Interval
TID	Traffic Identifier
TSF	Timing Synchronization Function
TXOP	Transmission Opportunity
USB	Universal Serial Bus
USRP	The Universal Software Radio Peripheral
VHT	Very High Throughput
VoIP	Voice over IP
WCC-CSM	Wireless Covert Channel with Constellation Shaping Modulation
WDS	Wireless Distribution System
WEP	Wired Equivalent Privacy
WLANs	Wireless Local Area Networks
WPA	Wi-Fi Protected Access