UNIVERSITÉ
DE GENÈVE

Département : Informatique

Détection de canaux cachés dans le MAC 802.11 via des métriques issues de la théorie de l'information

Bachelor en Informatique

Année : 2024–2025

Superviseurs

Prof. Eduardo Solana
Alexandre-Quentin Berger

Étudiant

Mustafa al-Nuaimi

Contents

Introduction

Abstract

Les réseaux locaux sans fil appartenant principalement à la famille de normes IEEE 802.11 sont devenus omniprésents dans notre quotidien numérique. De nos domiciles à nos lieux de travail, ils offrent une connectivité flexible et pratique à une multitude d'appareils. Cette prolifération s'accompagne cependant à une exposition de plus en plus importante aux menaces de sécurité. La nature même du médium sans fil, partagé et ouvert, rend ces réseaux intrinsèquement plus vulnérables que leurs homologues filaires. En effet, alors que les réseaux filaires bénéficient d'une protection physique relative par le câblage, les ondes radio des réseaux 802.11 se propagent au-delà des frontières physiques des bâtiments, les exposant à des écoutes et des interférences non autorisées. Assurer la confidentialité, l'intégrité et la disponibilité des informations transitant sur ces réseaux est donc une préoccupation majeure, non seulement pour les utilisateurs individuels mais aussi pour les organisations qui s'appuient sur ces infrastructures pour leurs opérations critiques. Parmi les menaces sophistiquées qui pèsent sur la sécurité des communications, les canaux cachés (souvent désignés par le terme anglais *covert channels*) représentent un défi particulier.

Les Canaux cachés

Définition et Analogie du Gardien

Un canal caché, ou *covert channel*, est défini par le National Institute of Standards and Technology (NIST) comme "un canal intra-système non intentionnel ou non autorisé qui permet à deux entités coopérantes de transférer des informations d'une manière qui viole la politique de sécurité du système, mais sans excéder les autorisations d'accès de ces entités".

Essentiellement, ils exploitent des aspects du système (ressources partagées, protocoles de communication) qui ne sont pas normalement considérés comme des voies de communication pour transmettre secrètement des données. La création d'un canal caché demande généralement une connaissance approfondie du système ou du protocole ciblé afin d'identifier les mécanismes détournables.

L'analogie des prisonniers Alice et Bob illustre les canaux cachés. Pour communiquer secrètement un plan d'évasion malgré l'inspection de leur courrier par le gardien Walter, ils modifient subtilement des aspects de leurs lettres d'apparence anodine.

Par exemple, ils peuvent convenir d'un codage binaire basé sur l'espacement vertical de leur écriture : un double interligne entre deux phrases pourrait encoder un bit '1', tandis qu'un simple interligne encoderait un bit '0'. Walter, lisant le contenu pour y déceler des messages suspects, pourrait facilement ignorer cette variation mineure de mise en page, l'attribuant à une habitude d'écriture. Il est l'entité de surveillance, Alice et Bob les communicants. Le canal caché est cette exploitation des propriétés du système autorisé.

L'objectif fondamental est de masquer l'existence même du canal. Le trafic doit se fondre dans la normalité pour que le canal lui-même ne soit pas détecté comme une anomalie, avant même de considérer le chiffrement du message secret.

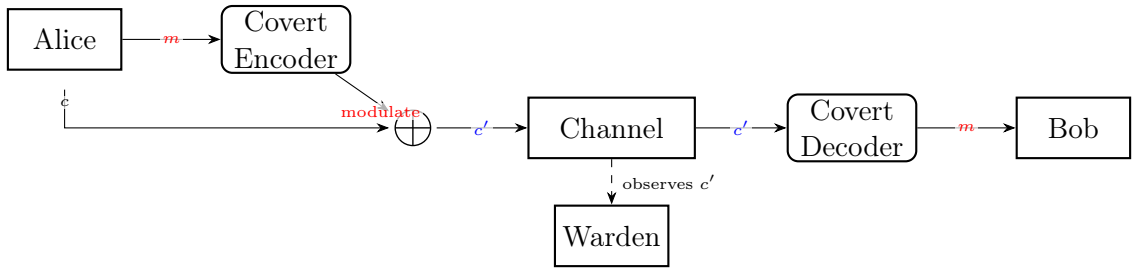

Figure 1: Modélisation du canal caché via l'analogie des prisonniers. Alice envoie un message secret (m) en modulant une communication autorisée (c, la lettre) pour créer un message modifié (c'). Walter, le gardien, observe c' mais ne détecte que la communication normale. Bob peut ensuite décoder c' pour récupérer le message secret m.

Taxonomie des Canaux cachés

Les canaux cachés sont généralement classifiés en fonction de la manière dont l'information secrète est encodée et transmise par l'entité émettrice, et comment elle est perçue et décodée par l'entité réceptrice. La taxonomie la plus courante et la plus fondamentale, notamment définie dans le TCSEC, distingue principalement deux types de canaux : les canaux de stockage et les canaux temporels.

- **Canaux de Stockage (*Storage Channels*)** : Ces canaux transmettent de l'information en modifiant un attribut d'un objet ou d'un emplacement de stockage (physique ou logique) qui est accessible, à la fois par l'émetteur et le récepteur du message secret. L'émetteur écrit des données dans cet emplacement partagé, et le récepteur lit ces données pour récupérer l'information. Dans le contexte des réseaux, cela implique souvent de manipuler des champs spécifiques au sein des en-têtes des protocoles de communication. Par exemple :
  - Utiliser des champs réservés ou inutilisés dans les en-têtes de paquets.
  - Cacher des données dans des noms de fichiers, des tailles de fichiers, des permissions de fichiers, ou des attributs étendus de fichiers sur un serveur de fichiers partagé accessible par les deux entités.

- **Canaux Temporels (*Timing Channels*)** : Contrairement aux canaux de stockage qui modifient directement le contenu ou l'état d'une ressource, les canaux temporels encodent l'information en modulant le moment où des événements se produisent ou la durée entre des événements. L'émetteur manipule le timing de ses actions, et le récepteur observe ces variations temporelles pour décoder le message secret. Ces canaux sont souvent considérés comme plus difficiles à détecter que les canaux de stockage car ils ne modifient pas nécessairement le contenu des paquets ou des données échangées.
  Exemples de canaux temporels :
  - Moduler les délais inter-paquets : un délai court entre deux paquets successifs peut représenter un bit '0', tandis qu'un délai long peut représenter un bit '1'.
  - Modifier l'ordre d'arrivée de paquets qui pourraient autrement arriver dans un ordre différent ou non spécifié.

  Ces canaux sont particulièrement pertinents pour notre étude de la couche MAC 802.11, car les mécanismes d'accès au médium comme le DCF (Distributed Coordination Function) sont fondamentalement basés sur des temporisations et des comportements aléatoires qui peuvent être subtilement biaisés.

Limitations et détection

Limitations des Canaux cachés

Malgré leur potentiel de menace et leur sophistication conceptuelle, les canaux cachés ne sont pas sans faiblesses. Ils présentent plusieurs limitations qui restreignent leur efficacité pratique et surtout, offrent des pistes pour leur détection :

- **Faible Bande Passante** : C'est sans doute la limitation la plus significative et la plus fréquemment citée. Pour rester furtifs et se diluer dans le trafic normal ou les opérations légitimes du système, les canaux cachés ne peuvent généralement transmettre que de faibles volumes de données par unité de temps.
- **Sensibilité au Bruit et aux Perturbations** : Les canaux cachés, en particulier les canaux temporels qui reposent sur des modulations subtiles du timing des événements, sont souvent très sensibles au "bruit" inhérent aux systèmes physiques et réseaux informatiques. Ce bruit peut prendre la forme de latence variable, de retransmissions de paquets dues à des erreurs, d'interférences radio dans les réseaux sans fil, de variations de charge du système, ou d'actions imprévisibles d'autres utilisateurs ou processus. Ces facteurs peuvent facilement corrompre l'information transmise secrètement, introduisant des erreurs de bits.
- **Difficulté de Mise en oeuvre et Synchronisation** : La création d'un canal caché fiable et robuste demande une expertise technique avancée et une connaissance approfondie du protocole ou du système exploité. L'émetteur et le récepteur doivent se mettre d'accord sur un protocole de communication secret, incluant l'encodage des bits, la synchronisation, et la gestion des erreurs. La synchronisation entre l'émetteur et le récepteur du canal secret peut être particulièrement complexe à établir et à maintenir, surtout dans des environnements réseau dynamiques et distribués où les conditions changent constamment.

Méthodes de détection

Deux grandes approches dominent la détection des canaux cachés :

- **Détection basée sur les signatures :** Cette méthode repose sur l'identification de motifs spécifiques connus dans le trafic réseau. Par exemple, un canal caché pourrait exploiter le champ `IP ID` ou le champ `TTL` dans les paquets IP pour transmettre de l'information en modifiant ces valeurs selon un schéma prédéfini. Si ce schéma est connu à l'avance, un système de détection à signatures peut surveiller ces champs et déclencher une alerte en cas de correspondance. Cette approche est peu coûteuse en calcul et efficace contre des canaux déjà documentés. En revanche, elle devient inefficace face à des techniques nouvelles ou adaptatives qui modifient dynamiquement leur comportement.
- **Détection par anomalie :** Cette approche modélise le comportement normal du réseau (trafic habituel, distribution des intervalles temporels, etc.), puis signale toute déviation statistiquement significative. Par exemple, on peut mesurer les caractéristiques du trafic 802.11 en l'absence de canal caché, puis surveiller des variations inhabituelles dans les délais ou les champs de trame. Des techniques statistiques ou d'apprentissage automatique peuvent être employées pour construire ce profil de référence. Le principal inconvénient est le nombre potentiellement élevé de faux positifs (un trafic normal inhabituel peut déclencher une alarme) et le coût de mise en place de ces modèles de référence.

Cas Pratiques et Scénarios d'Attaque

Malgré leurs faiblesses et les difficultés de leur détection, les canaux cachés ne sont pas de simples curiosités théoriques. Ils représentent une facette avancée de la stéganographie réseau, ouvrant la porte à des scénarios concrets capables de contourner des mesures de sécurité établies. Traditionnellement, leur potentiel est perçu à travers le prisme de la menace : exfiltration discrète de données sensibles, communication de commande et de contrôle pour des logiciels malveillants, ou encore coordination furtive d'attaques en contournant les pare-feux et les systèmes de détection d'intrusion (IDS).

Cependant, il est crucial de noter que leur nature furtive n'est pas intrinsèquement malveillante et peut, paradoxalement, être mise au service de la défense. En tant que voie de communication secondaire et non évidente, un canal caché peut en effet constituer une couche de sécurité additionnelle dans une stratégie de défense en profondeur. Par exemple, il peut être employé pour l'échange sécurisé d'informations d'authentification critiques, comme des clés de session, des jetons à usage unique ou des mots de passe, en dehors des canaux de communication principaux qui pourraient être sous surveillance.
Dès lors, comprendre le potentiel d'application de ces canaux dans leur totalité est essentiel, non seulement pour évaluer la menace qu'ils représentent et développer des contre-mesures adaptées, mais aussi pour envisager leur utilisation dans des mécanismes de sécurité innovants.

La section suivante présente un bref exemple afin de concrétiser ces concepts.

Exfiltration Discrète de Données Sensibles

Imaginons un poste de travail compromis par un logiciel malveillant au sein d'un réseau d'entreprise. Ce réseau est protégé par des pare-feux, des systèmes de détection d'intrusion (IDS/IPS) ect. L'objectif du malware est d'exfiltrer de petites quantités de données hautement sensibles (par exemple, des clés de chiffrement, des fragments de mots de passe, des listes de cibles internes).

Pour se faire, le malware utilise l'interface Wi-Fi du poste compromis, non pas pour se connecter au réseau d'entreprise (ce qui serait surveillé), mais pour établir un canal de communication MAC direct et furtif. Il émet des trames 802.11 spécialement forgées, qui peuvent apparaître comme du trafic de gestion périodique (la trame de balise, par exemple) ou de découverte anodin émanant d'un appareil non associé.

Un dispositif récepteur, contrôlé par l'attaquant et situé à portée Wi-Fi (par exemple, un ordinateur portable dans un véhicule à proximité ou dans un bâtiment adjacent), est en mode écoute. Ce récepteur est programmé pour identifier et capturer ces trames spécifiques, en filtrant potentiellement sur l'adresse MAC émettrice ou sur des signatures uniques du canal caché lui-même. Une fois capturées, les trames sont décodées pour reconstituer les données exfiltrées. L'avantage de cette approche est qu'elle opère "sous le radar" des systèmes de sécurité traditionnels qui analysent principalement les flux TCP/IP et les connexions établies.

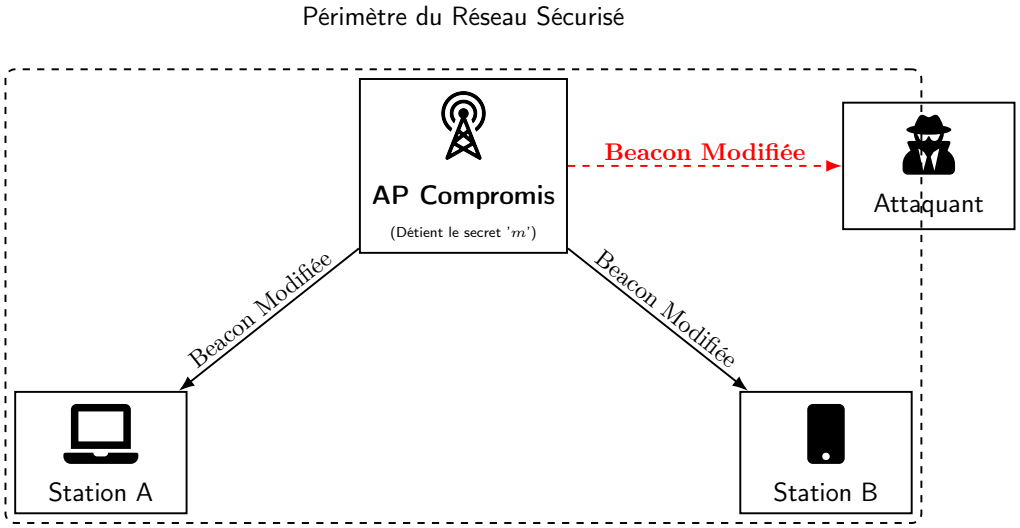

Figure 2: Schéma d'exfiltration de données via un canal caché dans les balises 802.11. Un point d'accès (AP) compromis, qui détient une information secrète 'm', la dissimule dans les headers de ses trames de beacon périodiques par exemple. Les stations légitimes (A et B) reçoivent ces beacons comme un trafic normal. Un attaquant à portée radio, cependant, capture ces beacons et en extrait le secret, contournant ainsi les sécurités traditionnelles.

La Couche MAC 802.11 et la Fonction DCF

L'analyse des canaux cachés au niveau MAC, qui est au cœur de cette étude, exige une connaissance précise de ses mécanismes. Ce chapitre établit cette base en expliquant la couche MAC (Medium Access Control) du standard IEEE 802.11, un composant fondamental qui gère la communication entre les stations et le médium sans fil partagé. Nous y présenterons ses rôles principaux, les types de trames qu'elle utilise et son mécanisme d'accès (la fonction DCF).

Rôle de la Couche MAC IEEE 802.11

La sous-couche MAC 802.11 assure la liaison entre les couches réseau supérieures et la couche physique sans fil. Pour transformer le médium radio en un canal de communication fiable, elle exécute plusieurs fonctions critiques :

- **Contrôle d'accès au médium :** Elle arbitre l'accès au canal partagé via le protocole CSMA/CA pour minimiser les collisions de transmission.
- **Formatage et Adressage :** Elle encapsule les données des couches supérieures dans des trames MAC (MPDU) en y ajoutant les en-têtes nécessaires.
- **Fiabilité et Intégrité :** Elle assure la fiabilité par un mécanisme d'accusé de réception (ACK) pour chaque trame transmise, et garantit l'intégrité via une séquence de contrôle (FCS / CRC). Elle peut également fragmenter les données volumineuses.

Dans le cadre de la détection de canaux cachés au niveau MAC, cette variété de mécanismes, chaque champ de trame, chaque état ou option du protocole devient un point d'injection potentiel pour un message secret. Plus la couche est riche, plus elle offre de possibilités pour y glisser discrètement des informations et compliquer leur repérage.

Types de Trames MAC 802.11

La norme 802.11 base sa communication autour de trois catégories principales de trames MAC, chacune conçue pour une fonction spécifique :

- **Trames de gestion :** Gèrent l'établissement et la maintenance des connexions. Elles incluent les (*Beacons*) annonçant un réseau, les requêtes de sonde (*Probe Requests*) pour la découverte active, et les trames d'association ou d'authentification.
- **Trames de contrôle :** Ce sont des trames courtes, comme les accusés de réception (*ACK*) qui confirment la livraison, ou les *PS-Poll* pour la gestion de l'énergie.
- **Trames de données :** Assurent le transport de la charge utile. Les standards récents leur adjoignent des mécanismes de Qualité de Service (QoS).

Structure d'une trame

Toute trame MAC IEEE 802.11 comprend trois grandes parties :

- **En-tête** : champ *Frame Control*, champ *Duration/ID*, adresses (1–4) et champ *Sequence Control*.
- **Corps de trame** : données utiles (absent sur la plupart des trames de contrôle).
- **FCS** : code CRC32 de détection d'erreurs.

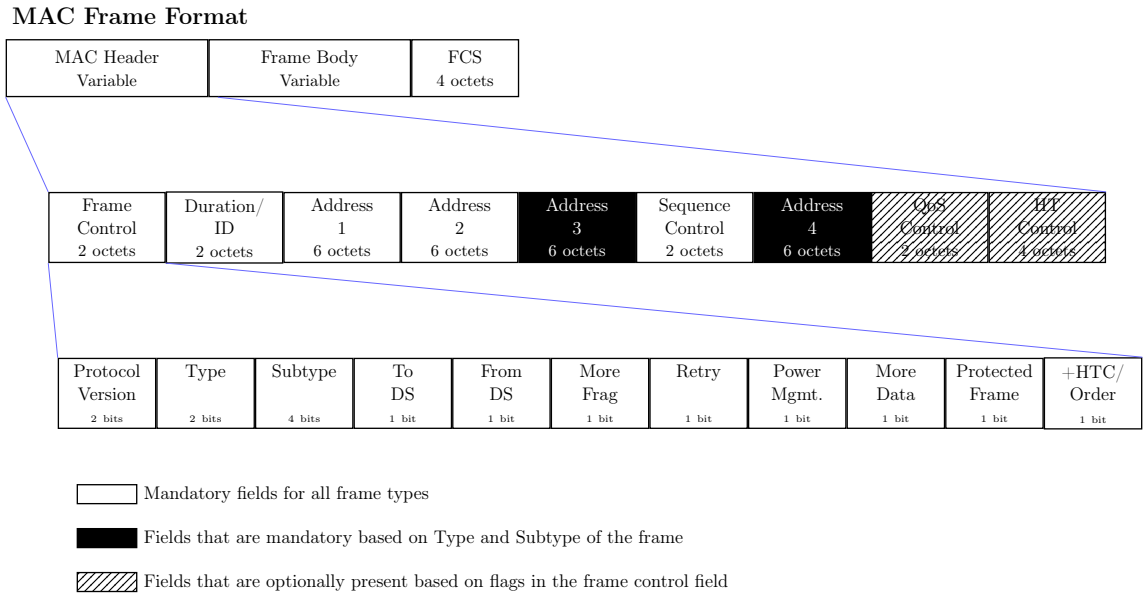

Figure 3: Structure détaillée d'une trame MAC 802.11.

La Fonction de Coordination Distribuée (DCF)

La DCF est le mécanisme d'accès fondamental du standard IEEE 802.11, basé sur le protocole CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).

Son principe est simple : *Listen Before Talk*.

Plutôt que de détecter les collisions comme en Ethernet (CSMA/CD), une approche difficile en radio, le standard cherche justement à les éviter. Pour cela, DCF combine deux éléments clés : des temporisations précises, les Intervalles Inter-Trames (IFS), et un système d'accusés de réception (ACK) systématique. Toute trame unicast doit être explicitement acquittée par le destinataire. L'absence d'ACK dans le délai imparti est interprétée comme une collision ou une perte, déclenchant une retransmission.

Intervalles Inter-Trames (IFS)

Les IFS sont des temporisations critiques qui priorisent l'accès au médium. Pour notre analyse (en ignorant le mécanisme RTS/CTS), deux sont essentiels :

- **DIFS (DCF Interframe Space)** : C'est l'intervalle initial qu'une station doit observer sur un canal inactif avant de pouvoir *initier* une transmission et commencer sa procédure de contention (backoff).
- **SIFS (Short Interframe Space)** : Beaucoup plus court, il est réservé aux réponses de haute priorité qui ne doivent subir aucune attente, comme l'envoi d'un ACK immédiatement après la réception d'une trame.

Vecteur d'Allocation Réseau (NAV).

Le **NAV (Network Allocation Vector)** est un mécanisme de détection de porteuse virtuelle. Il s'agit d'un chronomètre que chaque station met à jour en lisant le champ 'Durée' des trames qu'elle surprend sur le canal. Tant que ce chronomètre n'est pas expiré, la station considère le médium comme occupé et diffère toute nouvelle transmission, ce qui aide à prévenir les collisions.

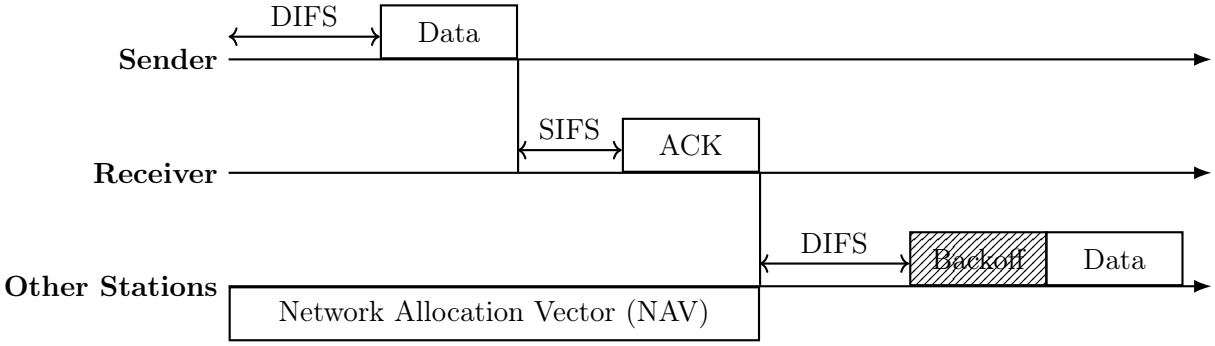

Figure 4: Chronogramme du mécanisme d'accès de base en DCF (CSMA/CA).

Mécanisme de Backoff Exponentiel

Comme illustré à la page précédente, une fois l'intervalle DIFS écoulé, plusieurs stations peuvent tenter d'émettre simultanément. Pour résoudre cette contention, DCF emploie un mécanisme de backoff aléatoire.
Le processus est régit par deux paramètres clés : la Fenêtre de Contention (CW), qui définit la plage de l'attente aléatoire, et le Slot Time, l'unité de temps discrète durant laquelle le canal est évalué.
L'algorithme ajuste dynamiquement la CW : elle est initialisée à une valeur minimale (CW_{min}), doublée à chaque échec de transmission (collision présumée) jusqu'à un maximum (CW_{max}), et réinitialisée à CW_{min} après chaque succès. La procédure est décrite dans l'Algorithme 1.

Algorithm 1 IEEE 802.11 DCF Backoff Procedure

Input: A frame to be transmitted.
Output: Transmission success or failure.

$CW \leftarrow CW_{\min}$
$retries \leftarrow 0$
while $retries <$ max_retries **do**
    Wait until the channel is idle for a DIFS period.
    $backoff\_counter \leftarrow$ random_integer$(0, CW)$
    while $backoff\_counter > 0$ **do**
        if the channel is detected busy **then**
            Wait until the channel is idle for a DIFS period again.
        Wait for one *Slot Time*.
        $backoff\_counter \leftarrow backoff\_counter - 1$
    Transmit the frame.
    Wait for an ACK.
    if an ACK is received **then**
        return Success
    else
        ▷ Collision is presumed
        $CW \leftarrow \min(2 \times (CW + 1) - 1, CW_{\max})$
        $retries \leftarrow retries + 1$
return Failure (max retries reached)

Le caractère de ce mécanisme est sa force, mais aussi sa vulnérabilité. Un attaquant peut manipuler ce timing pour créer un canal temporel caché. Au lieu de choisir une valeur de backoff purement aléatoire, il peut la moduler pour encoder des bits d'information. Pour un observateur externe, ce trafic peut sembler légitime car les délais de transmission restent variables, mais un récepteur complice peut décoder le message secret en analysant ces micro-variations temporelles.

Problématique

Après avoir examiné la littérature sur les canaux cachés, et après avoir détaillé les mécanismes fondamentaux de la couche MAC 802.11 et le fonctionnement de la fonction DCF, il devient apparent que ces mécanismes protocolaires peuvent être détournés de leur usage nominal. Cette potentialité expose une vulnérabilité d'autant plus critique qu'elle échappe à la surveillance traditionnelle. En effet, bien que les réseaux locaux sans fil IEEE 802.11 soient omniprésents en entreprise et dans les infrastructures critiques, les outils de sécurité en place ciblent essentiellement les attaques de couche supérieure. La menace se situe donc à un niveau plus profond du protocole, là où les défenses habituelles sont souvent aveugles.

Comment détecter et quantifier la présence de canaux cachés au niveau MAC à l'aide de métriques information-théoriques ?

Pour aborder cette problématique, notre démarche se concentrera sur l'étude approfondie de un à deux canaux cachés particulièrement représentatifs, choisis pour leur pertinence fondamentale à une analyse par la théorie de l'information. Ce travail se divisera en deux parties principales. Dans un premier temps, nous développerons une analyse mathématique de ces canaux et du mécanisme qu'ils exploitent afin de caractériser formellement leur fonctionnement et leur empreinte statistique. Dans un second temps, une phase de simulation/test viendra valider les hypothèse théoriques établis et quantifier la performance des métriques de détection proposées dans un environnement contrôlé.

État de l'Art

Canaux cachés dans les réseaux IEEE 802.11

Nous allons décrire ci-dessous trois canaux cachés de la couche MAC 802.11, que nous avons sélectionnés pour leur intérêt analytique. Ces techniques sont en effet particulièrement adaptées à une analyse s'appuyant sur la théorie de l'information. Il s'agit dans les trois cas de canaux temporels (timing channels), qui encodent l'information en modulant les délais du protocole. Le fonctionnement de chaque méthode est ensuite détaillé dans sa propre sous-section.

Covert DCF

La technique *Covert DCF* utilise un canal de synchronisation caché (timing channel) fondé sur le mécanisme de **backoff aléatoire** de la fonction DCF d'IEEE 802.11. Covert DCF consiste à détourner ce délai aléatoire pour y encoder des données secrètes. L'idée est que le caractère aléatoire normal du backoff serve de couverture, rendant les modifications discrètes aux yeux d'un observateur.
Concrètement, l'émetteur et le récepteur établissent à l'avance un **codebook** qui associe des séquences de bits à des valeurs spécifiques de délai de backoff. Par exemple, on peut décider que chaque combinaison de 3 bits sera mappée vers un temps d'attente précis avant émission (par ex. 000_2 correspond à un backoff de k_1 slots, 001_2 à k_2 slots, etc.). Lors de la transmission, la station émettrice, au lieu de choisir un backoff aléatoire, sélectionne la valeur prédéfinie dans le codebook correspondant aux bits qu'elle souhaite envoyer. Ainsi, chaque trame transmise est précédée d'un délai dont la durée encode une portion du message secret. Le récepteur, positionné à l'écoute du médium, mesure l'intervalle de temps écoulé entre le moment où le canal devient libre (fin de DIFS, ou fin de la trame précédente et le DIFS qui suit) et l'émission effective de chaque trame par l'émetteur. En utilisant le codebook partagé, il peut alors décoder la séquence binaire transmise en traduisant chaque durée de backoff observée en bits correspondants. La taille du codebook et la différence de temps entre les symboles influencent directement le débit du canal mais aussi sa détectabilité ; des codebooks plus grands ou des différences de temps plus marquées pourraient s'écarter davantage des distributions de backoff typiques. Ce type de canal souligne la difficulté de valider le caractère véritablement aléatoire des mécanismes de temporisation dans les protocoles distribués.

StegoBackoff

StegoBackoff est une méthode qui repose également sur la manipulation du backoff DCF, mais de façon plus simple et potentiellement plus furtive. Ici, chaque bit secret est encodé dans la parité (pair ou impair) du nombre de slots de backoff utilisés avant une transmission. Normalement, la station choisit un nombre aléatoire de slots dans l'intervalle $[0, CW - 1]$ après le DIFS. Avec StegoBackoff, l'émetteur s'assure que ce nombre soit *pair* pour signifier un bit « 0 », ou *impair* pour signifier un bit « 1 ». Pour ce faire, après avoir tiré une valeur de backoff aléatoire B_{rand}, la station l'ajuste si nécessaire afin d'obtenir la parité souhaitée correspondant au bit à transmettre. Si l'ajustement conduit à une valeur hors de l'intervalle $[0, CW - 1]$ (par exemple, $B_{rand} = 0$ et on doit le rendre impair, ou $B_{rand} = CW - 1$ et on doit le rendre pair), la station peut choisir de re-tirer une valeur, ou d'encoder le bit en privilégiant le non-dépassement des bornes, c'est-à-dire en soustrayant le backoff de une unité.

Cette astuce subtile permet d'insérer un bit par trame transmise par le canal caché, sans modifier aucun champ explicite des paquets émis, ce qui est difficilement discernable du point de vue du protocole puisqu'on reste dans les limites autorisées. Le récepteur, de son côté, écoute le médium et calcule, pour chaque trame reçue de l'émetteur, le nombre de slots de temporisation écoulés depuis la fin du DIFS. Il détermine alors simplement si ce nombre est pair ou impair afin de reconstituer le bit secret. Ainsi, StegoBackoff réalise un canal timing en s'intégrant dans la randomisation inhérente du backoff : la variabilité naturelle du délai d'accès au canal sert de masque, et le schéma pair/impair reste statistiquement discret. Cette approche offre un canal timing potentiellement robuste et transparent, avec un débit d'environ 1 bit par trame pertinente, tout en étant difficile à distinguer d'un comportement normal car l'ajustement perturbe très peu la distribution statistique attendue des backoffs. La simplicité de l'encodage par parité est sa force en termes de furtivité par rapport à Covert DCF qui impose des timings plus spécifiques.

StegoFrameOrder (SFO)

La méthode *StegoFrameOrder* exploite l'ordre d'émission des stations dans le réseau pour créer un canal caché au niveau MAC. Contrairement aux techniques précédentes qui modifient des champs ou des timings d'une seule station, SFO nécessite la collaboration d'au moins deux stations au sein du même réseau Wi-Fi. L'idée centrale est qu'en observant quelle station émet en premier un paquet après un événement de synchronisation commun (typiquement, la balise/beacon périodique envoyée par le point d'accès), on peut encoder un bit d'information. Les stations coopérantes synchronisent leur comportement sur les trames Beacon (envoyées typiquement toutes les 100 ms par l'AP), afin de définir des intervalles temporels dans lesquels un bit sera transmis.
Supposons deux stations A et B qui participent au canal caché : à chaque réception d'une Beacon, elles savent qu'une fenêtre pour un bit caché s'ouvre. Pour transmettre un bit `0`, la station A va délibérément initier la première transmission après la Beacon, tandis que la station B retarde sa propre émission. Inversement, pour transmettre un bit `1`, c'est la station B qui émettra en premier. L'ordre est obtenu en manipulant délibérément le timing d'accès au médium (par exemple, en modifiant le backoff ou en mettant temporairement la trame en tampon). Le champ Timestamp des Beacons peut être utilisé pour une synchronisation plus fine des messages, par exemple via un mécanisme de counter rollover pour indiquer le début d'un message.
Deux modes de fonctionnement principaux existent :

- **Mode « deux informées » (*both-informed*) :** Les deux (ou N) stations connaissent le message secret et coordonnent activement leurs transmissions. Ce mode est plus simple à implémenter mais requiert que l'information secrète soit présente sur plusieurs terminaux.
- **Mode « une informée » (*one-informed*) :** Une seule station connaît le message et module son timing par rapport au trafic (supposé régulier et prévisible) d'une autre station non complice. Ce mode est plus robuste mais sa fiabilité peut être moindre car elle dépend de la prédictibilité du trafic de la station non informée.

Le destinataire du message observe l'ordre des premières trames émises par les stations participantes après chaque Beacon pour reconstituer le message. Avec N stations collaboratrices, la capacité peut atteindre $\log_2(N!)$ bits par intervalle de synchronisation. Cette méthode ne requiert aucune modification du contenu des trames elles-mêmes, ce qui la rend difficile à détecter par des mécanismes classiques. Ce type de canal est conceptuellement plus difficile à détecter car il faut analyser le comportement coordonné de multiples entités.

Approches Générales de Détection

Après avoir détaillé des exemples de canaux cachés au niveau de la couche MAC 802.11, il est essentiel de se pencher sur les méthodologies générales développées pour leur détection.
Comme évoqué précédemment, les approches se divisent principalement en deux catégories : la détection par signature et la détection par anomalie.

Du Signature à l'Anomalie

La détection basée sur les signatures, bien que simple et efficace pour les menaces connues, présente une faiblesse fondamentale : elle est réactive. Elle repose sur une base de données de motifs ou de séquences de bits caractéristiques d'un canal caché identifié au préalable. Par conséquent, cette méthode est incapable de détecter des canaux cachés nouveaux, dont le mécanisme ne correspond à aucune signature existante.
Pour surmonter ce problème, la recherche s'est orientée vers la détection par anomalie. Ce paradigme adopte une démarche proactive : plutôt que de chercher des motifs malveillants connus, il vise à construire un modèle statistique du comportement normal du système ou du réseau. Toute déviation statistiquement significative par rapport à ce modèle de référence (baseline) est alors signalée comme une anomalie potentielle, et donc un possible canal caché. L'efficacité de cette approche repose sur l'hypothèse fondamentale qu'un canal caché, pour transmettre de l'information, doit nécessairement moduler une ressource ou un paramètre du système, altérant ainsi ses propriétés statistiques d'une manière mesurable. Les sections suivantes se concentrent exclusivement sur ce paradigme, en explorant les métriques, notamment issues de la théorie de l'information, qui permettent de quantifier ces altérations.

Détection par la Théorie de l'Information : Quantifier l'Anormalité

La théorie de l'information, initiée par Claude Shannon, fournit un cadre mathématique rigoureux pour quantifier l'information, l'incertitude et le bruit. Son application à la détection de canaux cachés permet de passer d'une simple observation binaire à une mesure concrète de la distance entre le comportement observé et le comportement attendu.

L'Entropie de Shannon

Le concept le plus fondamental de la théorie de l'information est l'entropie. Pour une variable aléatoire discrète X pouvant prendre des valeurs x issues d'un alphabet $\mathcal{X}$ avec une distribution de probabilité $p(x)$, l'entropie de Shannon $H(X)$ est définie comme :

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

Cette valeur mesure l'incertitude moyenne associée à la variable X, ou la quantité moyenne d'information (en bits) obtenue en observant un tirage de X.
Dans le contexte de la détection de canaux cachés, cette métrique est utilisée comme un (*shape test*). L'idée est de modéliser une caractéristique observable du trafic (par exemple, la valeur du compteur de backoff, la taille d'un paquet) comme une variable aléatoire X. On calcule l'entropie de cette variable sur un flux de trafic légitime pour établir une valeur de référence. Ensuite, on surveille l'entropie du trafic en temps réel. Un canal caché qui, par exemple, utiliserait préférentiellement un petit sous-ensemble de valeurs de backoff possibles pour encoder des données, réduirait la diversité des symboles observés. Cela se traduirait par une distribution de probabilité plus biaisé et donc une entropie plus faible que la normale, déclenchant une alerte.
Cependant, un attaquant sophistiqué peut déjouer un simple test d'entropie. Il peut concevoir son canal de manière à préserver la distribution statistique de premier ordre du trafic légitime. Par exemple, un canal temporel comme le TRCTC (Time-Replay Covert Timing Channel) enregistre des délais inter-paquets légitimes et les rejoue dans un ordre aléatoire pour transmettre des informations. La distribution globale des délais (l'histogramme) reste similaire à celle du trafic normal, rendant l'entropie de premier ordre inefficace pour la détection. Cette limitation met en évidence la nécessité d'analyser non seulement *quelles* valeurs apparaissent, mais aussi *dans quel ordre* elles le font.

L'Entropie Conditionnelle Corrigée (CCE) comme Test de Régularité

Pour analyser les dépendances séquentielles, on se tourne vers l'entropie conditionnelle. L'entropie conditionnelle d'ordre m, notée $H(X_n|X_{n-1}, \ldots, X_{n-m})$, mesure l'incertitude restante sur le prochain symbole X_n connaissant les m symboles précédents . Elle quantifie la prédictibilité d'une séquence.
Cette métrique est la base du *test de régularité* (*regularity test*). Son pouvoir réside dans sa capacité à détecter deux types d'anomalies opposées :

- **Une régularité excessive :** Un canal caché simple qui utilise un motif répétitif créera un trafic très prévisible. L'entropie conditionnelle sera donc anormalement basse.
- **Un caractère aléatoire excessif :** À l'inverse, un canal comme TRCTC, en rejouant des délais légitimes dans un ordre aléatoire, détruit les corrélations temporelles naturelles du trafic. Le trafic résultant, bien que sa distribution de premier ordre soit correcte, est "trop aléatoire". Son entropie conditionnelle sera anormalement élevée, proche de l'entropie de premier ordre, signalant une absence de corrélation là où il devrait y en avoir .

L'un des défis pratiques est que l'estimation de l'entropie conditionnelle sur des échantillons de données finis (comme une fenêtre de quelques centaines de paquets) est notoirement biaisée . Pour pallier ce problème, Gianvecchio et Wang ont introduit l'*Entropie Conditionnelle Corrigée* (Corrected Conditional Entropy, CCE) . La CCE ajuste l'estimation de l'entropie avec un terme correctif qui dépend de la taille de l'échantillon N et du nombre de sous-séquences uniques U observées. Cette correction rend la métrique plus robuste pour l'analyse de flux réseau réels, ce qui en fait un outil de détection puissant.

La Divergence de Kullback-Leibler (KL) : Mesurer la Distance Distributionnelle

Une autre approche consiste à mesurer directement à quel point la distribution du trafic observé s'écarte de la distribution normale de référence. La divergence de Kullback-Leibler (KL), ou entropie relative, est l'outil de choix pour cette tâche. Pour deux distributions de probabilités discrètes P (trafic observé) et Q (trafic de référence), la divergence KL est définie par :

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log_2 \left(\frac{P(x)}{Q(x)} \right)$$

Cette métrique quantifie le nombre de bits d'information supplémentaires nécessaires en moyenne pour coder des échantillons de P en utilisant un code optimisé pour Q . En d'autres termes, elle mesure "l'inefficacité" ou la "surprise" d'utiliser le modèle Q alors que la réalité est P.
Dans notre contexte, $D_{KL}(P||Q) \geq 0$, et l'égalité n'a lieu que si $P = Q$. Un canal caché parfaitement furtif chercherait à générer un trafic P tel que $D_{KL}(P||Q)$ soit aussi proche de zéro que possible . Toute modulation, même subtile, qui altère la distribution des probabilités du trafic légitime se traduira par une valeur de $D_{KL} > 0$, qui peut servir de base à un détecteur.

Cette métrique met en lumière un compromis fondamental et inévitable pour l'attaquant : celui entre la capacité du canal caché et sa furtivité. Pour augmenter la capacité, l'attaquant doit utiliser des modulations plus agressives, ce qui crée une distribution P de plus en plus distincte de la distribution de référence Q. Techniquement, cela augmente la valeur de D_{KL}, rendant le canal plus facile à détecter. Un système de détection basé sur la divergence KL vise donc à rendre ce compromis le plus défavorable possible pour l'attaquant, le forcant soit à un débit quasi nul, soit à une détection quasi certaine.

Approches par Apprentissage Automatique (Machine Learning)

Les approches par apprentissage automatique représentent une évolution naturelle des méthodes statistiques. Plutôt que de s'appuyer sur une seule métrique pour prendre une décision, les modèles de ML peuvent apprendre automatiquement à partir de données pour classifier le trafic comme "légitime" ou "caché" en se basant sur une multitude de caractéristiques.

La performance de tout modèle d'apprentissage supervisé dépend de manière critique de la qualité des caractéristiques extraites des données pour être fournies en entrée du classifieur. L'apprentissage automatique ne remplace pas la théorie de l'information ; au contraire, il la met en pratique. Les métriques décrites précédemment (entropie, CCE, divergence KL) constituent d'excellentes caractéristiques, car elles sont conçues pour être sensibles aux anomalies statistiques que l'on cherche à détecter.

Un vecteur de caractéristiques robuste pour un flux réseau pourrait ainsi combiner :

- Des statistiques de base : moyenne, variance, minimum, maximum des délais inter-paquets, des tailles de paquets, etc.
- Des métriques information-théoriques : scores d'entropie et de CCE calculés sur des fenêtres de trafic glissantes.
- Des caractéristiques protocolaires : fréquence d'utilisation de certains drapeaux, valeurs dans des champs spécifiques

Une fois le vecteur de caractéristiques défini, il est utilisé pour entraîner un classifieur. Les modèles couramment employés dans la littérature incluent les k-plus proches voisins (k-NN), les machines à vecteurs de support (SVM) et les réseaux de neurones profonds (DNN). L'entraînement se fait sur un vaste jeu de données étiquetées, contenant de nombreux exemples de trafic normal ainsi que du trafic généré par divers outils de canaux cachés connus.

Analyse par la Théorie de l'Information

Pourquoi étudier DCF ?

Bien que le panorama des canaux cachés au niveau de la couche MAC IEEE 802.11 soit vaste et englobe une diversité de techniques de stockage et temporelles, ce travail ciblera ses efforts spécifiquement sur les canaux qui exploitent la procédure de backoff de la Fonction de Coordination Distribuée (DCF). Des canaux tels que le *Covert DCF* et le *StegoBackoff*, qui fonctionnent en manipulant subtilement les intervalles de backoff aléatoires, ont été choisis comme nos principales études de cas.

Cette focalisation est motivée par plusieurs facteurs critiques. Premièrement, le mécanisme de backoff du DCF est un composant fondamental des réseaux 802.11 basés sur la contention. Sa dynamique temporelle régit l'accès au canal pour l'ensemble du trafic de données standard, ce qui rend toute vulnérabilité en son sein particulièrement pertinente et impactante.

Deuxièmement, et c'est le point le plus important pour notre approche analytique, le comportement légitime du processus de backoff du DCF est exceptionnellement propice à une analyse information-théorique. Contrairement à d'autres sources de variabilité temporelle plus chaotiques dans les réseaux, le caractère aléatoire du DCF est structuré et bien compris. Il peut être décrit formellement avec un haut degré de précision par des cadres mathématiques établis, notamment le modèle de Bianchi. Cette capacité à modéliser formellement le processus légitime est primordiale, car elle fournit une distribution de référence quantitative et robuste pour les choix de backoff légitimes. Cette référence transforme le problème de la détection d'un exercice purement heuristique ou basé sur des signatures en une analyse quantitative et rigoureuse. Nous ne sommes plus limités à constater qu'un schéma de trafic semble étrange; nous pouvons mesurer avec précision la divergence statistique entre un comportement observé et la vérité terrain théorique.

Par conséquent, les canaux basés sur le DCF constituent une étude de cas idéale pour l'application d'un cadre information-théorique rigoureux. Cette approche quantitative nous permet de dépasser les simples heuristiques et de modéliser formellement le conflit entre l'émetteur secret et le gardien comme un problème de distinctibilité statistique. Nous pouvons mesurer avec précision l'empreinte laissée par les manipulations d'un émetteur secret, en analysant non seulement la distribution globale des valeurs de backoff choisies, mais aussi les motifs structurels plus subtils cachés au sein de la séquence de ces choix. Cela permet une exploration mathématique approfondie du compromis fondamental auquel un attaquant est confronté : toute tentative d'augmenter la capacité d'un canal crée inévitablement une déviation statistique plus significative et donc plus facilement détectable. Les méthodologies développées ici fournissent un modèle puissant pour l'analyse d'une vaste classe de canaux temporels qui fonctionnent en altérant un processus aléatoire connu.

Implications Pratiques pour la Détection

Dans les réseaux IEEE 802.11 en conditions réelles, le décompte du backoff DCF de chaque station est géré au sein de firmwares propriétaires. Il peut être interrompu (gelé) et repris par la couche physique (PHY) sans émettre d'indices temporels explicites sur le médium, seul l'événement de transmission final est visible. Ainsi, un observateur passif du monde réel (par exemple, un point d'accès ou un moniteur passif) peut seulement enregistrer l'*intervalle d'inactivité inter-trames* après la période obligatoire DIFS. Cet intervalle observé fusionne un nombre inconnu de slots de backoff, ainsi que les pauses induites par le gel du compteur dues aux autres stations en compétition. Sans une visibilité directe sur les temporisateurs de slots, il faut recourir à l'estimation statistique des valeurs de backoff cachées, ce qui complique grandement la conception d'un détecteur dans des conditions de congestion réalistes.

En revanche, dans un simulateur à événements discrets tel que SimPy ou ns-3, les slots de backoff et les opérations de gel/dégel sont modélisés comme des événements logiciels explicites. Le compteur de slots de chaque station réside dans la mémoire de l'hôte, et chaque gel est planifié et visible par tout objet de surveillance. Étant donné que notre trafic simulé est configuré pour être fortement congestionné, avec toutes les stations luttant sans relâche pour l'accès au canal, la capacité d'observer chaque cycle de gel est garantie. Cela nous permet de supposer une connaissance complète et précise des durées de backoff de chaque station. Dans ce scénario idéalisé (promiscuous mode), nous pouvons calculer directement les métriques information-théoriques sur les distributions de slots réelles sans introduire d'erreur d'inférence.

Il est important de noter que même si les temporisateurs de backoff individuels sont pas toujours possible à calculer, une manipulation systématique du backoff d'une station se manifestera par une fréquence d'accès au canal anormalement élevée ou par des intervalles d'inactivité inhabituellement courts entre les transmissions consécutives de cette même station. Cela fournit un canal auxiliaire détectable malgré le manque de visibilité directe sur le temporisateur.

Le Modèle de Référence : Comportement Légitime du Backoff

Afin d'analyser quantitativement les canaux cachés temporels qui manipulent la procédure de backoff du DCF, il est essentiel d'établir d'abord un modèle formel et précis du comportement légitime des stations. Ce modèle de référence fournit la vérité terrain théorique par rapport à laquelle nous pouvons mesurer les déviations statistiques introduites par un émetteur secret. À cette fin, nous employons le modèle bien établi de Bianchi, un cadre analytique puissant pour le DCF de la norme IEEE 802.11.

Le Modèle de Bianchi : Hypothèses et Formalisme

Le modèle de Bianchi fournit une description mathématique du débit en régime permanent d'un réseau sans fil fonctionnant sous des hypothèses spécifiques et importantes. L'hypothèse principale est celle d'un **réseau homogène et saturé**.

- **Saturation :** Cela implique que les n stations dans le domaine de collision ont toujours une trame prête à être transmise. Ceci représente un scénario de contention assez commun, où le canal est sous une contrainte maximale. Cette hypothèse est non seulement cruciale pour la tractabilité analytique, mais c'est aussi une configuration couramment utilisée dans les simulateurs de réseau comme ns-3 pour créer un environnement stable et reproductible afin d'analyser la performance et le comportement des protocoles MAC sous forte charge. De plus, cette condition de saturation est également essentielle au bon fonctionnement du canal *Covert DCF*, car elle garantit que le récepteur soit exposé à un flux de trames continu, lui permettant ainsi de mesurer avec précision les délais de backoff modulés afin de décoder l'information secrète.
- **Homogénéité :** Cela signifie que les n stations sont statistiquement identiques, elles utilisent les mêmes paramètres DCF, elles partagent les mêmes caractéristiques physiques et se comportent selon les mêmes règles. Cela permet à l'analyse de se concentrer sur le comportement d'une seule station représentative.

Sous ces hypothèses, le modèle est défini par trois paramètres système : le nombre de stations en compétition (n), le stade de backoff maximal (m), et la taille initiale de la fenêtre de contention (W_0). Il résout la probabilité de transmission (τ) et la probabilité de collision (p) d'une station en régime permanent en trouvant la solution numérique unique $(\tau^\star, p^\star)$ au système suivant d'équations couplées non linéaires :

$$p = 1 - (1-\tau)^{n-1} \tag{1}$$

$$\tau = \frac{2(1-2p)}{(1-2p)(W_0+1) + pW_0(1-(2p)^m)} \tag{2}$$

Une fois la probabilité de collision en régime permanent $p^\star$ connue, nous pouvons trouver la probabilité, $\mathbb{P}_k$, qu'une station opère à un stade de backoff donné k (ayant subi k collisions consécutives précédentes pour son paquet actuel) :

$$\mathbb{P}_k = \frac{(p^\star)^k(1-p^\star)}{1-(p^\star)^{m+1}}, \quad \text{pour } k \in \{0,1,\ldots,m\} \tag{3}$$

À chaque stade k, la taille de la fenêtre de contention est $W_k = 2^k W_0$, et une station sélectionne son compteur de backoff b uniformément dans l'intervalle d'entiers $[0, W_k - 1]$.

La PMF Légitime du Backoff $P_0(b)$: Formulation et Propriétés

En moyennant le processus de sélection uniforme sur tous les stades de backoff possibles, pondérés par leurs probabilités respectives $\mathbb{P}_k$, nous dérivons la fonction de masse de probabilité (pmf) globale pour la valeur initiale du compteur de backoff b sélectionnée par une station légitime. Cette distribution de référence, notée $P_0(b)$, est donnée par :

$$P_0(b) = \sum_{k=0}^{m} \left(\mathbb{P}_k \cdot \frac{\mathbf{1}_{\{0 \le b \le W_k - 1\}}}{W_k} \right), \quad \text{pour } b \ge 0 \tag{4}$$

où $\mathbf{1}_{\{\cdot\}}$ est la fonction indicatrice. Cette pmf est la signature statistique des choix de backoff légitimes.

Une propriété importante de cette distribution, qui découle directement de sa formulation, est que $P_0(b)$ est une fonction en escalier (constante par morceaux). En outre, cette structure distinctive engendre deux caractéristiques interdépendantes qui définissent la signature du trafic légitime :

- **Uniformité Locale par Construction :** Pour toute valeur de backoff b à l'intérieur du premier segment de la fenêtre de contention, $0 \le b < W_0$, la fonction indicatrice dans l'Équation 4 est égale à 1 pour tous les stades $k = 0, \ldots, m$. Ainsi, $P_0(b)$ est constant sur toute cette plage :

  $$P_0(b) = C_0 = \sum_{k=0}^{m} \frac{\mathbb{P}_k}{W_k} \quad \text{pour } b \in [0, W_0 - 1]$$

  De même, pour le segment suivant, $W_0 \le b < W_1$, $P_0(b)$ est constant à une nouvelle valeur inférieure $C_1 = \sum_{k=1}^{m} \frac{\mathbb{P}_k}{W_k}$, car ces backoffs ne peuvent plus être choisis au stade 0.

- **Asymétrie Globale (Skewness) :** En général, $P_0(b)$ est plat sur chaque intervalle $[W_{j-1}, W_j - 1]$, et comme $\mathbb{P}_k > 0$, la hauteur de ces paliers diminue strictement à chaque limite de fenêtre, c'est-à-dire $C_0 > C_1 > C_2 > \ldots$. Cette structure crée une asymétrie globale vers les petites valeurs de backoff, car la majeure partie de la masse de probabilité est concentrée dans les premiers paliers, les plus élevés. Cette asymétrie étagée inhérente est la signature clé du comportement légitime.

Canaux Cachés Basés sur le DCF et Leurs Empreintes Statistiques

Ayant formellement défini la distribution de référence du backoff légitime, $P_0(b)$, nous analysons maintenant les canaux cachés spécifiques qui manipulent ce processus. Parmi ceux étudiés dans l'état de l'art, *Covert DCF* et *StegoBackoff* sont des sujets d'analyse idéaux car ils exploitent directement la procédure de backoff du DCF. Cette section rappellera brièvement leurs mécanismes et, plus important encore, discutera de leurs vulnérabilités statistiques inhérentes et des compromis de conception.

Covert DCF

Le canal *Covert DCF* opère en mappant des blocs d'un message secret (par exemple, de longueur L bits) à des valeurs de backoff (b) spécifiques et absolues, tirées d'un codebook prédéfini, $\mathcal{B}_{code}$. Au lieu de choisir un backoff de manière aléatoire comme le ferait une station légitime, l'émetteur choisit de manière déterministe $b = \mathcal{E}(msg)$ pour représenter le bloc de message secret msg.

La vulnérabilité centrale d'une implémentation naïve de *Covert DCF* réside dans le fait que les propriétés statistiques du message source peuvent fuiter directement dans la distribution des backoffs choisis. Si un mappage simple et direct est utilisé avec un texte source non uniforme (comme du texte ASCII non compressé), la fréquence des valeurs de backoff choisies reflétera la fréquence des blocs du message source. Un gardien pourrait détecter cette distribution asymétrique, car elle ne correspondrait pas à la distribution légitime attendue $P_0(b)$, qui est constante par morceaux.

Un émetteur secret sophistiqué doit par conséquent découpler les statistiques brutes de la source du choix final du backoff. Cela conduit à un compromis de conception fondamental entre le débit et la furtivité, qui se manifeste dans la manière dont le codebook est construit.

StegoBackoff

Le canal *StegoBackoff* est intrinsèquement furtif car il imite plus fidèlement le comportement légitime. Il encode un bit par trame en s'assurant que la valeur de backoff choisie, b, possède une parité spécifique (par exemple, pair pour '0', impair pour '1'). Ceci est réalisé en tirant d'abord un backoff aléatoire légitime, b_{rand}, puis en appliquant un ajustement minimal et conditionnel (par ex., $b \leftarrow b_{rand} + 1$) uniquement si la parité est incorrecte.

Puisque la valeur b choisie est presque toujours b_{rand} ou $b_{rand} + 1$, la distribution globale des valeurs de backoff reste extrêmement proche de $P_0(b)$, ce qui la rend très résistante aux tests qui analysent la forme complète de la distribution des backoffs. Sa vulnérabilité principale, cependant, réside dans les statistiques de la séquence de parités. Le processus légitime a une probabilité naturelle de produire un backoff pair, dérivée du modèle de Bianchi $P_0(b)$. Si le flux de bits secrets transmis est biaisé (c'est-à-dire, $\Pr(\text{bit secret} = 0) \neq 0.5$), la séquence de backoffs choisie par l'émetteur *StegoBackoff* présentera une distribution de parités différente. Par exemple, si l'émetteur transmet une longue chaîne de zéros, il forcera constamment ses valeurs de backoff à être paires. Un gardien pourrait détecter que cette station produit un ratio anormal de backoffs pairs par rapport aux impairs, en comparaison avec le ratio de référence attendu dérivé du modèle de Bianchi pour les conditions actuelles du réseau. Il s'agit d'un test statistique de premier ordre sur un alphabet binaire simple $\{pair, impair\}$.

Analyse Information-Théorique de Premier Ordre

Ayant établi le modèle de référence $P_0(b)$ pour la sélection légitime du backoff, nous construisons maintenant le cadre formel pour sa détection. Cette section explore les métriques information-théoriques de premier ordre, qui analysent la distribution de probabilité marginale des choix de backoff tout en ignorant les corrélations temporelles. Cette approche cadre le problème de la détection comme une comparaison statistique directe entre deux distributions :

- La **distribution légitime**, $P_0(b)$, telle que dérivée du modèle de Bianchi.
- La **distribution secrète**, notée $Q(b)$, qui représente le comportement statistique global d'une station opérant le canal caché.

Analyse par l'Entropie de Shannon

La propriété information-théorique la plus fondamentale d'une distribution est son Entropie de Shannon, qui quantifie son incertitude ou son contenu informationnel moyen. Une méthode de détection de premier ordre simple consiste donc à comparer l'entropie de la distribution secrète observée, $H(Q)$, à l'entropie de la référence légitime, $H(P_0)$:

$$H(P_0) = -\sum_b P_0(b)\ \log_2 P_0(b) \tag{5}$$

$$H(Q) = -\sum_b Q(b)\ \log_2 Q(b) \tag{6}$$

Un canal caché qui contraint de manière significative ses choix à un ensemble de valeurs de backoff plus restreint ou plus prévisible produirait probablement une distribution avec moins d'aléa, résultant en une chute d'entropie détectable où $H(Q) < H(P_0)$. Cette chute d'entropie, quantifiée par la différence,

$$\Delta H = H(P_0) - H(Q) \tag{7}$$

peut alors servir de métrique de détection directe. Cependant ΔH n'enregistre que l'amplitude globale de l'incertitude ; deux distributions aux formes profondément distinctes peuvent partager la même entropie. Autrement dit, un canal tel que *StegoBackoff*, qui se contente de moduler systématiquement la parité sans changer la densité globale, laisse souvent $H(Q) \simeq H(P_0)$ et passe sous le radar. En pratique, ΔH constitue donc un signal de pré-alarmes, si l'entropie chute nettement, la présence d'un canal caché est plausible ; si elle reste stable, aucune conclusion ne peut être tirée, il faut alors se tourner vers des divergences sensibles à la forme complète de la distribution.

Analyse par la Divergence de Kullback-Leibler (KL)

L'*entropie relative*, ou divergence de Kullback-Leibler (KL), quantifie directement la manière dont une distribution de probabilité diverge par rapport à une autre. Pour nos distributions de backoff, elle est définie comme suit :

$$D_{\mathrm{KL}}(P_0 \parallel Q) := \sum_b P_0(b) \log_2 \frac{P_0(b)}{Q(b)} \tag{8}$$

Cette quantité mesure la "surprise" informationnelle moyenne (en bits) lorsque la distribution secrète Q est supposée à la place de la distribution réelle P_0. Bien que $D_{\mathrm{KL}} = 0$ si et seulement si $Q = P_0$ et qu'elle augmente à mesure que Q s'écarte de P_0, la divergence de KL présente deux inconvénients majeurs dans un contexte de détection empirique :

- *Asymétrie :* $D_{\mathrm{KL}}(P_0 \parallel Q) \neq D_{\mathrm{KL}}(Q \parallel P_0)$. Cette mesure n'est pas symétrique, ce qui signifie que la valeur de la divergence dépend de la distribution choisie comme référence. Dans un scénario de détection, cela est problématique car nous recherchons une notion unique et bien définie de "distance" entre P_0 et Q, plutôt que deux valeurs différentes. Par exemple, une distribution secrète $Q(b)$ pourrait produire un faible $D_{\mathrm{KL}}(Q \parallel P_0)$ même si elle sous-représente systématiquement certaines valeurs de backoff auxquelles P_0 attribue une probabilité non nulle. Une telle divergence se manifesterait plutôt par un $D_{\mathrm{KL}}(P_0 \parallel Q)$ élevé. Cela illustre que l'ampleur de la divergence de KL peut varier considérablement selon la direction, ce qui complique son utilisation directe comme métrique de détection.

- *Singularités aux probabilités nulles :* $D_{\mathrm{KL}}(P_0 \parallel Q)$ devient infinie (non définie) s'il existe une valeur de backoff b telle que $P_0(b) > 0$ mais $Q(b) = 0$. En d'autres termes, si la stratégie secrète évite entièrement une valeur de backoff particulière que le processus légitime utilise avec une probabilité non nulle, l'entropie relative diverge formellement. Même dans des cas moins extrêmes, des événements simplement très improbables sous Q peuvent gonfler D_{KL} de manière disproportionnée. Cette sensibilité est problématique pour la détection de canaux subtils : avec un nombre fini d'observations, un résultat qui n'apparaîtrait pas par hasard dans l'échantillon secret (donnant $Q(b) = 0$ dans la distribution empirique) impliquerait fallacieusement une divergence infinie par rapport à P_0.

Analyse par la Divergence de Jensen-Shannon (JS)

Pour pallier ces problèmes, nous adoptons la *divergence de Jensen-Shannon (JS)* comme notre principale métrique de détection de premier ordre. La divergence JS est définie comme une variante symétrisée et lissée de la divergence KL :

$$D_{\mathrm{JS}}(P_0 \parallel Q) := \frac{1}{2} D_{\mathrm{KL}}(P_0 \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M) \tag{9}$$

Dans cette définition, $M = \frac{1}{2}(P_0 + Q)$ est la distribution moyenne des deux distributions.
Par construction, $D_{\mathrm{JS}}(P_0 \parallel Q)$ possède trois propriétés clés qui la rendent bien adaptée à la comparaison de distributions empiriques.
Premièrement, elle est *symétrique*, ce qui signifie que $D_{\mathrm{JS}}(P_0 \parallel Q) = D_{\mathrm{JS}}(Q \parallel P_0)$.

Deuxièmement, elle est *toujours finie*, car la distribution mixte M attribue une probabilité non nulle à tout résultat ayant une probabilité non nulle sous P_0 ou Q, évitant ainsi tout terme KL infini.

Troisièmement, elle est *bornée* : en utilisant le logarithme en base 2, $0 \leq D_{\mathrm{JS}}(P_0 \parallel Q) \leq 1$ bit. La valeur 0 n'est atteinte que si $P_0 = Q$, et 1 bit est la divergence maximale, obtenue lorsque P_0 and Q ont des supports disjoints (aucun chevauchement dans les valeurs de backoff choisies). Cette échelle bornée fournit une mesure intuitive de la différence et garantit que les écarts modérés ne sont pas éclipsés par des singularités.

Une autre perspective met davantage en lumière la pertinence de D_{JS} pour notre problème de détection. En fait, $D_{\mathrm{JS}}(P_0 \parallel Q)$ est égale à l'information mutuelle entre un backoff observé (sois une variable aléatoire B) et un indicateur binaire Z spécifiant si cette observation provenait de la distribution légitime ou de la distribution secrète (en supposant que les deux cas sont *a priori* équiprobables). Formellement :

$$D_{JS}(P_0 \parallel Q) = I(Z; B) = H(M) - \frac{1}{2} H(P_0) - \frac{1}{2} H(Q) \tag{10}$$

En d'autres termes, D_{JS} quantifie combien de bits d'information une seule observation de backoff fournit sur la présence du canal caché. Ainsi, une valeur de D_{JS} plus grande correspond à une déviation plus détectable (chaque backoff observé apporte une preuve significative d'un comportement anormal), tandis qu'une très petite valeur de D_{JS} indique que chaque observation ne contient presque aucune information distinctive, signifiant un canal hautement furtif.

Pourquoi aller au-delà des métriques de premier ordre?

Les analyses menées jusqu'ici comparent la *distribution marginale* des valeurs de back-off mesurées, notée $Q(b)$, au modèle légitime $P_0(b)$ dérivé de Bianchi. Bien que assez puissantes, ces métriques ne disent rien de l'*ordre* dans lequel les valeurs apparaissent : deux stations peuvent partager exactement la même histogramme $Q(b)$ et pourtant véhiculer des quantités d'information très différentes dès lors qu'elles réorganisent la séquence de back-off pour encoder, par exemple, du texte encodé en ASCII. Pour révéler cette portion cachée du signal, les régularités ou, au contraire, l'excès d'aléa introduits par l'encodage,il faut mesurer la dépendance temporelle au sein du processus $\mathbf{B} = (B_1, B_2, \dots)$. Les métriques dites d'*ordre supérieur* s'attaquent précisément à cette tâche : en évaluant combien d'incertitude subsiste sur le prochain back-off *après* avoir observé les m précédents, elles capturent les motifs séquentiels qu'un simple histogramme laisse échapper et fournissent ainsi un test bien plus puissant contre les canaux qui conservent la forme marginale tout en modifiant la dynamique interne.

Cadre formel

Soit la suite de back-off d'une station $\mathbf{B} = (B_1, B_2, \dots)$, $B_i \in \mathbb{N}$. On note H_m l'entropie conditionnelle d'ordre m :

$$H_m \;=\; H\big(B_n \mid B_{n-1}, \dots, B_{n-m}\big) \;=\; -\sum_{b_0^m} P(b_0^m)\log_2 \frac{P(b_0^m)}{P(b_0^{m-1})}, \tag{11}$$

où b_0^m abrège $(b_{n-m}, \dots, b_n)$.

Pour rendre ce concept plus concret, le diagramme suivant illustre comment un canal cachant du texte engendre des dépendances détectables.

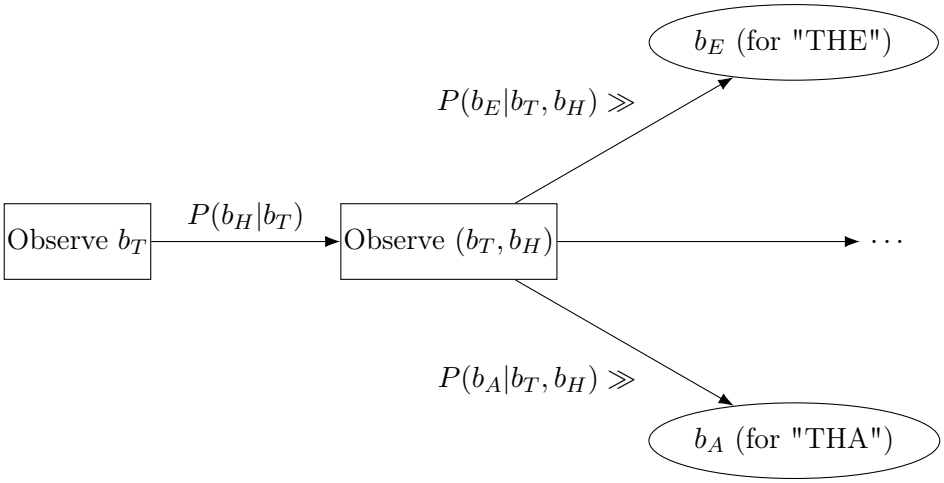

Figure 5: Illustration "caricatural" d'une chaîne de Markov d'ordre 2 pour la détection d'un canal caché encodant un caractère par backoff. Après avoir observé la séquence de back-off pour « TH » (historique (b_T, b_H)), la distribution de probabilité pour le back-off suivant se réduit à quelques issues très probables, comme b_E (pour « THE ») ou b_A (pour « THA »). Cette forte prédictibilité se traduit par une entropie conditionnelle très faible, $H(B_t|B_{t-1} = b_H, B_{t-2} = b_T)$, ce qui constitue une anomalie statistique détectable.

Mémoire et Entropie conditionnelle des backoffs légitimes

Le protocole DCF n'est pas sans mémoire: la valeur de backoff choisie avant la tentative t dépend implicitement du *stade* k_t (nombre de collisions déjà subies) qui, lui-même, dépend du sort de la trame précédente. Le graphe de Markov de la figure 6 résume ces dépendances : après chaque collision on avance vers k+1, après chaque réussite on revient à k=0.

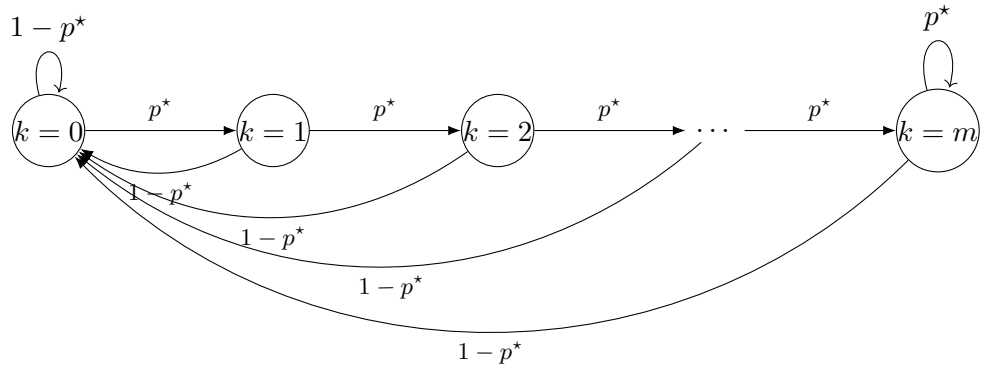

Figure 6: Chaîne de Markov des stades de backoff (modèle de Bianchi).

Cependant, en tant qu'observateur, nous nous interessons et avons typiquement que accès aux backoffs suivis de transmissions *réussie* appelons-les $B_1, B_2, \ldots$, alors chaque succès réinitialise le protocole en $k = 0$. Le tirage B_n provient donc toujours du même ensemble uniforme $\{0, \ldots, W_0\}$, *indépendamment* des collisions qui ont précédé les succès antérieurs. Formellement, pour tout n et tout antécédent de longueur r:

$$P\big(B_n = b \mid B_{n-1} = b_{n-1}, \ldots, B_{n-r} = b_{n-r}\big) \;=\; P_0(b),$$

d'où directement

$$H\big(B_n \mid B_{n-1}, B_{n-2}, \ldots\big) \;=\; H(B_n) \;=\; H_0.$$

Autrement dit, la chaîne locale est mémorielle, mais la suite global des backoffs suivie de succès est i.i.d, une propriété qui servira de référence pour les tests d'ordre supérieur. Tout canal caché efficace devra rompre cette indépendance, ce qui fera apparaître une entropie conditionnelle strictement inférieure ou upérieure à H_0 selon qu'il contraint ou qu'il sur-aléatorise la séquence. Ici pour quantifier la détection, nous calculerions typiquement :

$$H_0 - H_m$$

où

$$H_m \;=\; H\big(B_n \mid B_{n-1}, \ldots, B_{n-m}\big) \;=\; -\sum_{b_0^m} Q(b_0^m) \log_2 \frac{Q(b_0^m)}{Q(b_0^{m-1})},$$

Généralisation aux Séquences : Divergences KL et Jensen-Shannon d'Ordre Supérieur

Une approche alternative consiste à généraliser les divergences KL et JS pour comparer directement les distributions de probabilité jointes des séquences (ou n-grammes). L'idée est de comparer la distribution de probabilité jointe de ces séquences dans la trace capturée, notée $Q^{(n)}$, avec la distribution de référence attendue, $P_0^{(n)}$.
Considérons une séquence de n backoffs, $\mathbf{b} = (b_1, \ldots, b_n)$, où chaque b_i appartient à l'alphabet des valeurs possibles $\mathcal{S} = \{0, 1, \ldots, W_{\max}\}$. Conformément au modèle de Bianchi, une station légitime génère des backoffs de succès de manière indépendante. Sa distribution de probabilité jointe se factorise donc simplement en un produit des probabilités marginales :

$$P_0^{(n)}(\mathbf{b}) = \prod_{k=1}^{n} P_0(b_k), \quad \text{où } P_0 \text{ est la loi de référence (ex: uniforme).} \tag{1}$$

À l'inverse, la distribution jointe réelle, $Q^{(n)}$, qui peut être influencée par un canal caché, capture les dépendances temporelles. En appliquant la règle de la chaîne des probabilités, nous l'exprimons comme :

$$Q^{(n)}(\mathbf{b}) = \prod_{i=1}^{n} Q(b_i | b_1, \ldots, b_{i-1}). \tag{2}$$

La divergence de Kullback-Leibler d'ordre n, notée $D_{\mathrm{KL}}^{(n)}$, mesure l'information "supplémentaire" requise pour décrire les séquences observées $Q^{(n)}$ en utilisant le modèle indépendant $P_0^{(n)}$ comme base. Elle est définie par :

$$D_{\mathrm{KL}}^{(n)} = \sum_{\mathbf{b} \in \mathcal{S}^n} Q^{(n)}(\mathbf{b}) \log_2 \frac{Q^{(n)}(\mathbf{b})}{P_0^{(n)}(\mathbf{b})}. \tag{3}$$

En substituant les expressions de $P_0^{(n)}$ et $Q^{(n)}$, cette formule révèle sa signification profonde. Elle se décompose en une somme des divergences conditionnelles à chaque étape, pondérée par la probabilité d'observer le contexte (l'historique) :

$$D_{\mathrm{KL}}^{(n)} = \sum_{\mathbf{b}} Q^{(n)}(\mathbf{b}) \sum_{k=1}^{n} \log_2 \frac{Q(b_k \mid b_1, \ldots, b_{k-1})}{P_0(b_k)}. \tag{4}$$

Cette équation montre que $D_{\mathrm{KL}}^{(n)}$ agrège systématiquement les déviations par rapport à l'indépendance à tous les ordres jusqu'à n. Pour obtenir une mesure symétrique et bornée, on utilise la divergence de Jensen-Shannon, qui est une distance métrique (via sa racine carrée). Elle est construite à partir de D_{KL} en utilisant une distribution moyenne $M^{(n)} = \frac{1}{2}(Q^{(n)} + P_0^{(n)})$:

$$D_{\mathrm{JS}}^{(n)} = \frac{1}{2} D_{\mathrm{KL}}(Q^{(n)} \| M^{(n)}) + \frac{1}{2} D_{\mathrm{KL}}(P_0^{(n)} \| M^{(n)}). \tag{5}$$

D'un point de vue pratique, ces métriques sont estimées à partir d'une trace de trafic de longueur T. La première étape consiste à construire une approximation empirique de $Q^{(n)}$ en comptant les occurrences de chaque n-gramme $\mathbf{b}$:

$$N(\mathbf{b}) = \big|\{t \leq T - n + 1 : (B_t, \ldots, B_{t+n-1}) = \mathbf{b}\}\big|,$$

ce qui permet d'estimer la probabilité jointe par la fréquence relative : $\hat{Q}^{(n)}(\mathbf{b}) = \frac{N(\mathbf{b})}{T-n+1}$. Cette distribution empirique $\hat{Q}^{(n)}$ remplace alors $Q^{(n)}$ dans les équations (3) et (5) pour le calcul.

Limite Pratique des Métriques d'Ordre Supérieur

Si le formalisme des n-grammes est puissant, sa mise en oeuvre possède un obstacle fondamental connu sous le nom de « malédiction de la dimensionnalité ». Ce problème se manifeste par une explosion combinatoire des états possibles, qui rend l'estimation statistique à la fois coûteuse en mémoire et statistiquement instable.

Le coeur du problème réside dans l'augmentation exponentielle du nombre de séquences à analyser. Le nombre d'états à suivre est $|\mathcal{S}|^n$, où $|\mathcal{S}|$ est la taille de l'alphabet des backoffs.
Dans un scénario réaliste avec $W_{\max} = 1023$, l'alphabet $|\mathcal{S}|$ est de 1024. Par conséquent, une analyse des paires de backoffs ($n = 2$) requiert déjà de suivre plus d'un million de compteurs (1024^2). Pour les triplets ($n = 3$), ce chiffre explose à plus d'un milliard (1024^3) ect. Le stockage de ces compteurs devient rapidement prohibitif pour des équipements embarqués comme les points d'accès.

Cependant, la contrainte la plus critique n'est pas la mémoire, mais la rareté des données (*data sparsity*). Pour estimer de manière fiable la distribution de probabilité d'un milliard de triplets, il faudrait un volume de données de plusieurs ordres de grandeur supérieur. Une trace, même conséquente, d'un million de backoffs (10^6) serait répartie sur un milliard d'issues possibles. La quasi-totalité des compteurs $N(\mathbf{b})$ serait nulle, rendant le calcul des divergences instable (à cause du terme $\log(0)$) et dominé par le bruit d'échantillonnage. L'estimateur est alors dit « limité par la variance » (*variance-limited*) : nous manquons de données bien avant de manquer de puissance de calcul pour les traiter.

Pour contourner ce mur, plusieurs stratégies d'atténuation existent. La première consiste à réduire la taille de l'alphabet $|\mathcal{S}|$, par exemple en regroupant les valeurs de backoff (quantification) ou en appliquant un opérateur modulo. Une autre approche est la régularisation statistique, qui utilise des techniques comme le lissage de Laplace ou des modèles plus avancés pour estimer les probabilités d'événements rares sans les énumérer explicitement. Enfin, la voie la plus pragmatique, adoptée dans ce travail, est de se limiter à un ordre faible ($n = 1$ ou $n = 2$), où les exigences en mémoire et en données restent compatibles avec une analyse en temps réel sur des équipements réseau standards.

Simulations et Analyse des résultats

Pour nos expérimentations, nous emploierons systématiquement le même message secret. Ce dernier est un extrait de la nouvelle d'Arthur Conan Doyle, *The Adventure of the Speckled Band*. Le texte est converti en une séquence binaire en utilisant la table de correspondance ASCII.

De plus, nous avons opté pour l'enregistrement exhaustif de tous les tirages de backoff. Cette approche suppose un *warden* idéal, placé en mode promiscuous, capable de capturer chaque trame de données émise par la station cible et d'écouter le `ACK` qui suit. En connaissant précisément les durées fixes de DIFS, SIFS et du préambule, ce warden peut mesurer l'intervalle de temps écoulé entre la fin d'une trame et le début de la suivante, en déduire la durée de backoff réelle, puis la convertir en nombre de slots.

Ce modèle, bien qu'utile pour obtenir la « vérité terrain » nécessaire au développement et à la validation de détecteurs, reflète l'un des principaux biais des simulateurs: ils accordent souvent au warden une visibilité interne irréaliste (lecture directe des compteurs de backoff). En pratique, un warden humain ou logiciel doit *inférer* ces valeurs, ce qui augmente la variance de ses estimations et peut rendre un canal covert plus discret.

Description du Simulateur

Pour nos expériences, nous avons sélectionné *DCF-SimPy*, un simulateur de l'accès au canal IEEE 802.11 développé par Paweł Topór. Écrit en Python, ce simulateur s'appuie sur la bibliothèque **SimPy**, un framework puissant pour la simulation à événements discrets. L'approche à événements discrets est particulièrement bien adaptée à notre cas d'étude; au lieu de simuler chaque microseconde, le simulateur saute directement d'un événement pertinent à un autre (ex: fin d'un backoff, réception d'un ACK), ce qui le rend très efficace.

Le simulateur modélise chaque station du réseau comme un processus indépendant qui suit rigoureusement les règles de la fonction DCF, incluant la procédure de backoff exponentiel et la détection de collision. Un avantage notable de cette architecture est qu'elle représente de manière fidèle la nature décentralisée et concurrentielle de l'accès au médium, où chaque station prend ses décisions de manière autonome en se basant sur l'état perçu du canal.
Un aspect fondamental du simulateur *DCF-SimPy* est qu'il implémente, par défaut, un modèle de trafic saturé. Cela signifie que chaque station est supposée avoir systématiquement une trame prête à être transmise dès que la transmission précédente est achevée.

Ce choix de modélisation est crucial car il correspond précisément à l'hypothèse centrale du modèle de Bianchi que nous avons détaillé dans notre section théorique. En assurant que la simulation opère sous les mêmes conditions de saturation que le modèle analytique, nous garantissons une comparaison juste et directe. La simulation devient ainsi un banc d'essai empirique permettant de vérifier si le comportement agrégé des stations, suivant les règles du protocole, converge bien vers les probabilités de collision (p) et de transmission (τ) prédites par la théorie.

Validation par Simulation

Pour valider notre cadre théorique et obtenir une distribution de référence empirique du trafic légitime, nous avons eu recours à une simulation détaillée du mécanisme DCF. Cette démarche permet de confronter les prédictions du modèle de Bianchi à un comportement émergent issu d'une implémentation fidèle des règles du protocole.
Nous avons mené une simulation pour un réseau de $\mathbf{n = 10}$ stations opérant en conditions de saturation, avec les paramètres standards du DCF ($W_0 = 16$, $m = 7$). L'objectif était d'obtenir la distribution empirique des valeurs de backoff initiales et de la comparer à la distribution théorique $P_0(b)$ dérivée de Bianchi.

Comme l'illustre la Figure 7, les résultats démontrent une correspondance quasi parfaite entre le modèle et la pratique. La distribution empirique issue de la simulation épouse fidèlement la forme caractéristique en escalier, décroissante, de la distribution théorique de Bianchi. Cette forte corrélation valide simultanément la justesse de notre implémentation du modèle théorique et la fidélité du simulateur aux mécanismes fondamentaux du DCF. Nous disposons ainsi d'une baseline légitime, robuste et validée, contre laquelle nous pourrons évaluer l'impact des canaux cachés.

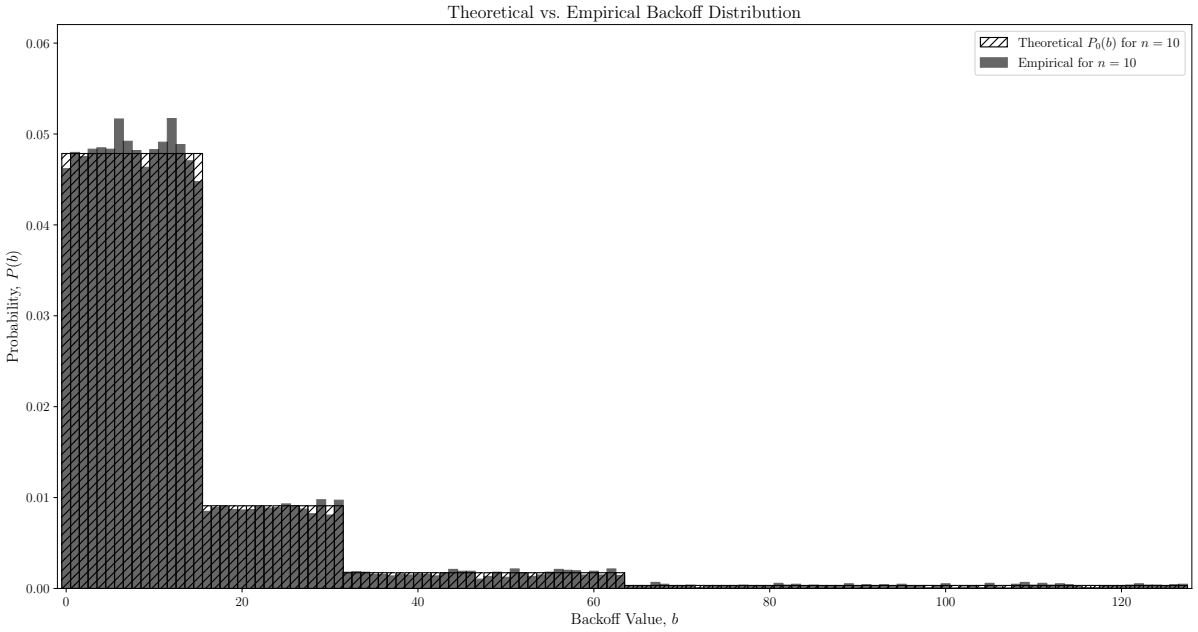

Figure 7: Comparaison entre la distribution de backoff théorique $P_0(b)$ (dérivée du modèle de Bianchi) et la distribution empirique obtenue via la simulation pour $n = 10$ stations. L'alignement étroit valide la fidélité de la simulation au modèle théorique.

Validation expérimentale et discussion

Détection locale de la phase "covert"

Dans notre scénario de test, un petit nombre de stations DCF coexistent sur le même canal. La station covert transmet d'abord 2000 trames selon le mécanisme DCF standard, rendant son trafic indiscernable des autres flux légitimes. À la trame numéro 2001 débutent plus de 2000 bits de message secret, encodés via le schéma *StegoBackoff* : chaque backoff tiré aléatoirement est ajusté d'au plus un slot pour respecter la parité du bit à transmettre. Les autres stations poursuivent, elles, leurs tirages de backoff selon la loi légitime $P_0(b)$, de sorte que la seule différence statistique se manifeste dans la distribution de parités et les incréments minimaux de backoff.
Nous concentrons notre étude sur *StegoBackoff*, dont le niveau de furtivité dépasse largement celui de *Covert DCF*. En effet, tout test suffisamment sensible pour révéler les ajustements discrets de StegoBackoff sera a fortiori capable de détecter la variante plus grossière de Covert DCF. Ainsi, en nous assurant d'une méthodologie robuste pour le cas le plus exigeant (StegoBackoff), nous couvrons automatiquement le cas de Covert DCF sans devoir redéfinir nos métriques.

Analyse globale des divergences

Comparer l'intégralité du trafic clandestin à la distribution de référence n'est pas informatif : les phases « normales » qui précèdent et suivent la séquence secrète diluent l'écart statistique. En effet, sur l'ensemble des backoffs émis, nous obtenons pour la station StegoBackoff :

Métrique	StegoBackoff vs P_0	Station normale vs P_0
$D_{\mathrm{KL}}(P_0 \parallel Q)$	0.074669	0.065695
$D_{\mathrm{JS}}(P_0 \parallel Q)$	0.006021	0.004803
$H(P_0)$	4.884168	4.884168
$H(Q)$	4.818182	4.811493
$\Delta H = H(P_0) - H(Q)$	0.065987	0.072676

Table 1: Comparaison des métriques d'information sur l'ensemble du trafic pour StegoBackoff et une station normale.

Ces valeurs globales, de l'ordre de quelques centièmes de bit, soulignent la faiblesse de la divergence moyenne lorsque la phase covert ne constitue qu'une fraction négligeable du trafic total. De plus, ce simple calcul ne renseigne pas sur la variation temporelle de ces métriques ni sur leur sensibilité à la longueur de la séquence covert ; il reste difficile de prévoir comment ces écarts se comportent en fonction du volume de données observé. C'est précisément pour surmonter cette incertitude que nous adoptons l'analyse par fenêtres glissantes, présentée dans la Section suivante.

Pourquoi des fenêtres glissantes ?

Le trafic d'une station IEEE 802.11 n'est pas stationnaire : ce dernier peut rester longtemps légitime, puis activer un canal furtif pour une durée limitée, et enfin revenir à un comportement normal.
Comparer une distribution globale Q_{global} à la référence P_0 reviendrait donc à *moyenner* sur des régimes hétérogènes ; l'écart dû au canal furtif serait dilué par les périodes légitimes (Tableau 1).
Pour capturer ces déviations locales, nous adoptons l'analyse par *fenêtres glissantes* : à chaque pas S (ici $S = 100$ observations), on calcule, sur une fenêtre de taille W, la distribution empirique

$$\hat{Q}_t(b) \;=\; \frac{1}{W} \sum_{i=t}^{t+W-1} \mathbf{1}\{b_i = b\},$$

puis les métriques de premier ordre :

$$\Delta H_t, \quad D_{\mathrm{KL}}\big(P_0 \,\|\, \hat{Q}_t\big), \quad D_{\mathrm{JS}}\big(P_0 \,\|\, \hat{Q}_t\big).$$

Le choix de W gouverne alors un compromis *réactivité / stabilité*, que nous explorons empiriquement.

Comparaison des différentes métriques issues de la théorie de l'information

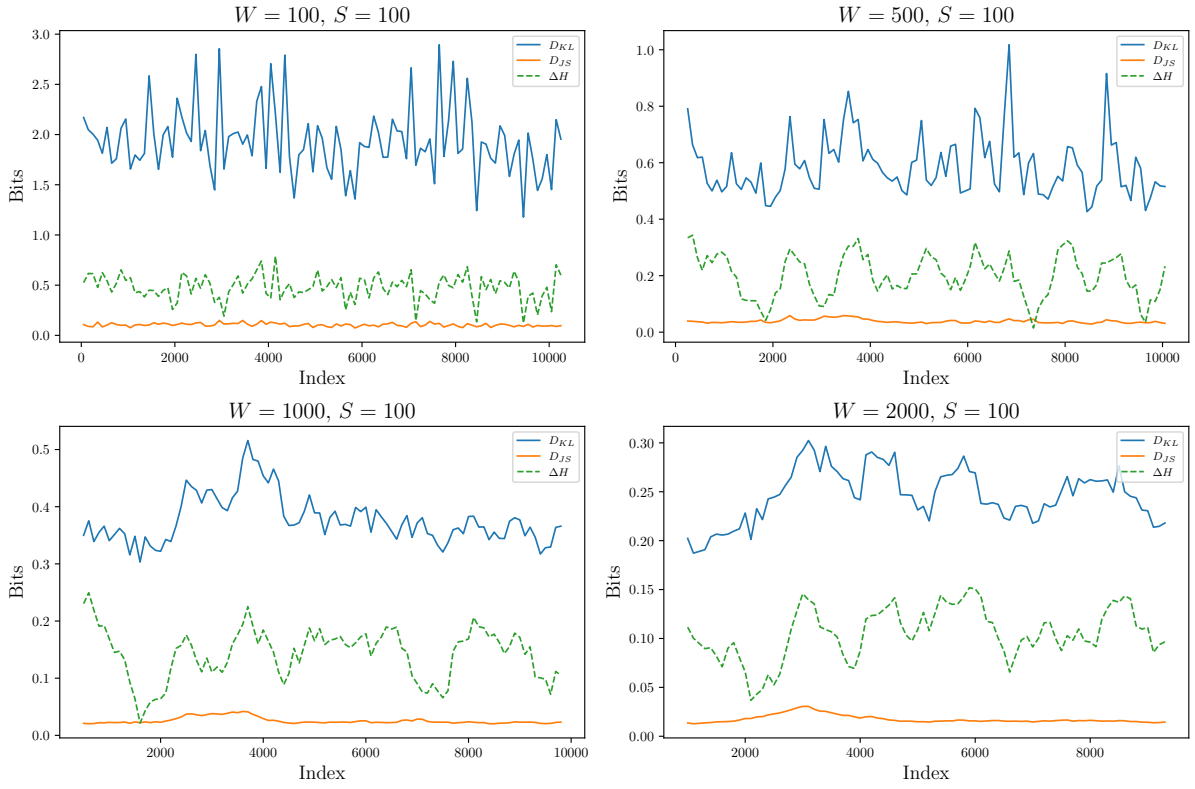

Figure 8: Évolution temporelle des métriques ΔH_t (vert), D_{KL} (bleu) et D_{JS} (orange) calculées sur la station StegoBackoff par rapport à la distribution de Bianchi $P_0(b)$ pour quatre tailles de fenêtre glissante $W \in \{100, 500, 1000, 2000\}$.

Fenêtres courtes ($W = 100$ et $W = 500$). Lorsque l'on observe seulement 100 ou 500 backoffs par segment, la variance d'échantillonnage domine entièrement le signal. La divergence de Kullback–Leibler explose en une série de pics irréguliers, même en l'absence de transmission secrète, ce qui interdit de fixer un seuil de détection fiable(Idem pour ΔH_t). La divergence de Jensen–Shannon, bien que bornée, reste plongée dans le bruit de fond et ne parvient qu'à esquisser un léger soulèvement comparable aux fluctuations naturelles du protocole.Ces résultats confirment que des fenêtres trop courtes génèrent autant de fausses alarmes qu'elles ne détectent de véritables anomalies.

Fenêtre intermédiaire ($W = 1000$). Avec mille backoffs par segment, le compromis *réactivité / stabilité* devient optimal. Hors phase covert, D_{KL} se stabilise autour d'une ligne de base silencieuse, et fait surgir un deux pic parfaitement après l'activation de StegoBackoff, autorisant le choix d'un seuil de détection clair. La divergence de Jensen–Shannon dessine un plateau presque rectangulaire, marquant précisément le début et la fin de l'encodage. L'entropie ΔH_t montre une légère pente pendant la phase secrète, mais ses fluctuations ultérieures montrent que ce critère reste avant tout un indicateur de pré-alerte, nécessitant un appui sur les divergences pour confirmer la présence du canal furtif. parasite.

Fenêtre longue ($W = 2000$). Sur cette fenêtre étendue, D_{JS} fournit une détection exemplaire : en effet, la courbe reste parfaitement plate hors phase furtive et délimite clairement la période d'attaque par un plateau lisse et marqué. En revanche, D_{KL} bénéficie d'un bruit de fond très réduit, mais ses fluctuations après le pic principal compliquent l'identification de la fin de la phase covert. De même, ΔH_t présente un pic visible suivi de fluctuations persistantes, rendant difficile la détection précise du retour à la valeur d'entropie de base. Ainsi, pour $W = 2000$, D_{JS} s'impose comme métrique de choix, tandis que D_{KL} et ΔH_t pâtissent du lissage temporel et deviennent plus difficiles à interpréter.

Comparaison station normale vs StegoBackoff

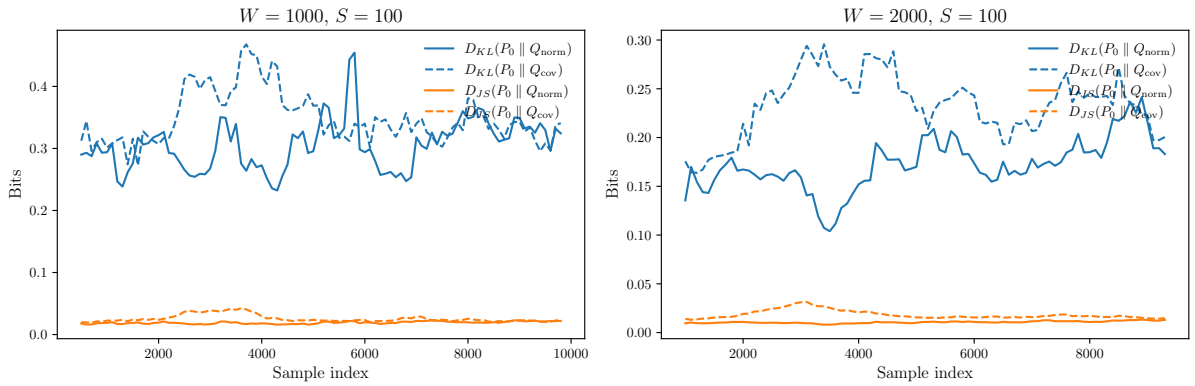

Figure 9: Superposition des divergences D_{KL} (bleu) et D_{JS} (orange) pour la distribution de la station normale (trait plein) rapport à $P_0(b)$ et la station furtive StegoBackoff (trait pointillé), avec les deux fenêtres les plus informatif : $W = 1000$ à gauche et $W = 2000$ à droite.

Sur le graphe de gauche ($W = 1000$), on constate que la station normale oscille fréquemment, sa courbe de D_{KL} présente un pic important hors phase covert($W = 6000$), traduisant la sensibilité de cette métrique aux fluctuations aléatoires. En revanche, D_{JS} pour la station normale reste strictement nul, ce qui souligne sa robustesse face aux variations statistiques naturelles. Du côté de la station furtive, les courbes pointillées s'élèvent rapidement dès le début de l'encodage secret : D_{KL} affiche un premier pic marqué avant de redescendre, tandis que D_{JS} forme un bombement rectangulaire précisément circonscrit à la plage d'attaque. Cette comparaison montre qu'un détecteur fondé sur un simple seuil de D_{KL} risquerait des fausses alertes, alors que D_{JS} offre un signal clair et sans fluctuations parasites pour isoler la période covert.

A droite ($W = 2000$), l'écart entre normal et furtif reste évident, mais la montée se fait de façon plus progressive : D_{KL} s'étale sur une durée plus longue avec une amplitude légèrement réduite, tandis que D_{JS} dessine une forme quasi-pointue autour des $W = 3000$, donc parfaitement circonscrite à la fenêtre d'attaque et exempte de fluctuations hors phase. Ce lissage confirme que D_{JS} reste le critère de détection le plus fiable pour délimiter la période covert, tandis que D_{KL} apporte une confirmation plus fine dès que sa variance d'estimation devient raisonnable.

Détection temporelle par divergence KL d'Ordre Supérieur

Comme indiqué précédemment, nous conservons l'ordre $n = 1$:
la conjonction de deux backoffs successifs suffit à trahir *StegoBackoff* tout en évitant l'explosion combinatoire et la variance massive qui apparaissent pour $n \geq 2$.
L'algorithme applique la même fenêtre glissante que les métriques d'ordre 0 ; mais, pour chaque fenêtre de W backoffs, il calcule la divergence de Kullback–Leibler $\mathrm{KL}_\ell\big(Q_\ell \| P_0 \otimes P_0\big)$ non pas pour un seul décalage, mais pour tous les retards $\ell = 1, \ldots, L$. Le vecteur $(\mathrm{KL}_1, \ldots, \mathrm{KL}_L)$ est alors condensé en un score unique K_t :
si ce vecteur change peu par rapport à la fenêtre précédente (variation quadratique $< \tau$), on retient son *minimum* ; ainsi les fluctuations numériques normales sont aplanies ; sinon on retient son *maximum*, ce qui accentue toute rupture introduite par un canal caché.

Algorithm 2 Score *KL-lag* adaptatif, ordre 1

Require: trace $(B_t)_{t=1}^{T}$, fenêtre W, pas S, retards L, seuil de stabilité τ

prev $\leftarrow \varnothing$
for $s \leftarrow 1$ **to** $T - W$ **step** S **do**
  $\mathbf{K} \leftarrow []$
  for $\ell \leftarrow 1$ **to** L **do**
    $Q_\ell \leftarrow \mathsf{Freq}\big((B_{t-\ell}, B_t) = (i, j)\big)$ sur $[s, s + W)$
    $K_\ell \leftarrow D_{\mathrm{KL}}\big(Q_\ell \,\|\, P_0(i)P_0(j)\big)$
    append K_ℓ to $\mathbf{K}$
  $\sigma^2 \leftarrow \sum_\ell (K_\ell - \mathit{prev}_\ell)^2$
  $K_t \leftarrow \begin{cases} \max \mathbf{K}, & \mathit{prev} = \varnothing \text{ ou } \sigma^2 \geq \tau \\ \min \mathbf{K}, & \sigma^2 < \tau \end{cases}$
  prev $\leftarrow \mathbf{K}$
  émettre K_t

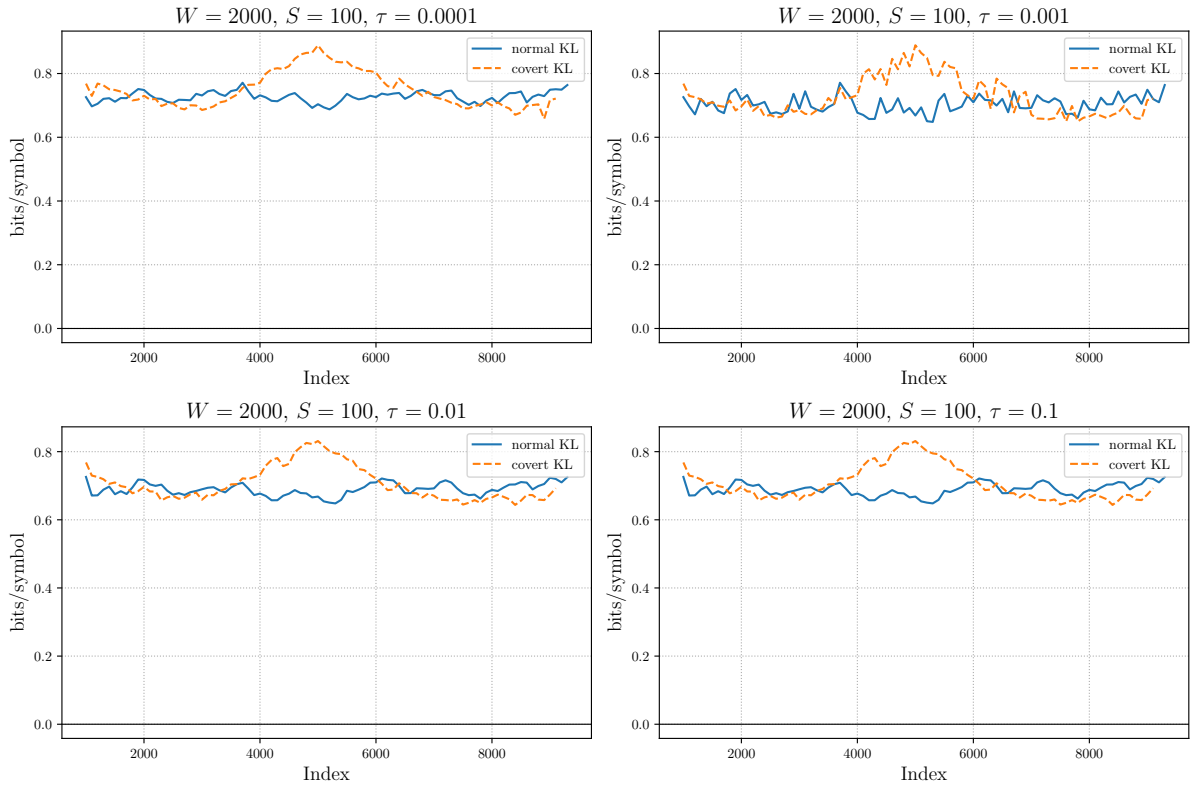

Figure 10: Superposition de la divergence KL multi-retards pour la distribution de la station normale rapport à $P_0(b)$ et la station furtive StegoBackoff pour 4 différents threshold

Le résultat montre clairement que quel que soit le seuil τ retenu (10^{-4}, 10^{-3}, 10^{-2} ou 10^{-1} dans les quatre graphes), la courbe orange de la station **covert** présente une bosse bien marquée entre les indices de 4000 à 6000, alors que la courbe bleue de la station normale reste collée à sa ligne de fond.
Quand τ est très petit (10^{-4}) le filtre « min quand stable » s'active souvent ; la partie légitime est donc plus plate, tandis que la portion StegoBackoff, qui brise la stabilité, passe immédiatement en mode « max » et s'envole autour de 0.9 bit. En augmentant τ on laisse revenir un peu de granularité, mais la bosse covert reste partout au-dessus de la normale et conserve 0.10 à presque 0.2 bit d'avance.
Autrement dit, l'exploitation de la dépendance temporelle (ordre 1, balayage de retards) donne à la divergence KL un pouvoir de séparation que la version « naïve » (n = 0, simple comparaison des distributions marginales) n'avait pas: la divergence KL de la station normale par rapport au modèle de Bianchi reste assez stable et constante, tandis que le canal caché fait surgir un « spike » clair quel que soit le réglage raisonnable du paramètre τ.

Conclusion Générale et Perspectives

Ce projet s'est focalisé sur la détection de canaux cachés temporels au sein du protocole IEEE 802.11, une menace d'autant plus dangereuse qu'elle exploite les mécanismes fondamentaux et aléatoires de l'accès au médium. En se focalisant sur la procédure de *backoff* de la fonction de coordination distribuée (DCF), nous avons cherché à quantifier la présence d'une communication covert, non pas par des signatures préétablies, mais par une analyse rigoureuse de ses empreintes statistiques. Cette conclusion synthétise les apports de notre travail, en mettant en évidence les limites à notre cadre expérimental, et trace les pistes d'une recherche future, plus proche encore des complexités du monde réel.

Si les résultats obtenus sont convaincants, il est important de reconnaître qu'ils ont été acquis dans un environnement de simulation qui, par nécessité, simplifie la réalité. J'aurais aimé pouvoir confronter notre méthodologie à un environnement plus authentique, où les hypothèses que nous avons posées ne tiennent pas nécessairement.
Premièrement, notre analyse s'est déroulée sous une hypothèse de trafic constamment saturé, condition idéale pour la validité du modèle de Bianchi mais pas toujours représentative des réseaux réels. Dans un tel contexte, la *baseline* de normalité n'est plus statique mais devient une cible mouvante, ce qui exigerait un détecteur capable d'adapter son modèle de référence en temps réel.
Deuxièmement, notre gardien bénéficiait d'un accès direct et parfait aux compteurs de *backoff* internes de chaque station, par supposition qu'il était en mode promiscuous. Dans un déploiement réel, un gardien est souvent passif, et doit donc inférer ces valeurs à partir des intervalles de silence observés sur le canal, un signal possiblement bruité par les interférences et les gels de compteur.
Enfin, notre simulation opérait dans un vide physique, sans les contraintes du bruit radio, des collisions dues aux nœuds cachés ou des pertes de paquets qui caractérisent les déploiements sans fil. Ces phénomènes du monde réel constituent une source de bruit qui corrompt non seulement la communication légitime, mais aussi le canal caché lui-même, tout en compliquant la tâche du détecteur.

Bibliography

[1] National Institute of Standards and Technology (NIST), "Covert Channel," *NIST Computer Security Resource Center Glossary*. [En ligne]. Disponible: https://csrc.nist.gov/glossary/term/covert_channel.

[2] NordVPN, "What is a covert channel?" *NordVPN Cybersecurity Glossary*. [En ligne]. Disponible: https://nordvpn.com/cybersecurity/glossary/covert-channel/.

[3] IEEE Std 802.11™-2020 (Revision of IEEE Std 802.11-2016), "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, 2020. [En ligne]. Disponible: https://ieeexplore.ieee.org/document/9363693

[4] R. Holloway and R. Beyah, "Covert DCF: A DCF-Based Covert Timing Channel in 802.11 Networks," in *Proc. 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, Valencia, Spain, 2011, pp. 570–579, doi: 10.1109/MASS.2011.60. [En ligne]. Disponible: https://ieeexplore.ieee.org/document/6076655

[5] Segura-Garcia, J., et al., "A Covert Channel Using the IEEE 802.11 Backoff Procedure for Enhanced Data Protection in Smart Grids," *Energies*, vol. 17, no. 3, p. 716, 2024..[En ligne]. Disponible: https://www.mdpi.com/1996-1073/17/3/716.

[6] W. Mazurczyk, K. Szczypiorski, and S. Wendzel, "StegoFrameOrder—MAC Layer Covert Network Channel for Wireless IEEE 802.11 Networks," *Sensors*, vol. 21, no. 18, p. 6268, Sep. 2021. [En ligne]. Disponible: https://www.mdpi.com/1424-8220/21/18/6268.

[7] S. Gianvecchio and H. Wang, "Detecting Covert Timing Channels: An Entropy-based Approach," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 547-561, Jul.-Aug. 2011. [En ligne]. Disponible: https://ieeexplore.ieee.org/document/5590253.

[8] P. Nowakowski, P. Żórawski, K. Cabaj, and W. Mazurczyk, "Detecting Network Covert Channels using Machine Learning, Data Mining and Hierarchical Organisation of Frequent Sets," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 12, no. 1, pp. 20–43, Mar. 2021. [En ligne]. Disponible: https://doi.org/10.22667/JOWUA.2021.03.31.020

[9] 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, Mar. 2000. [En ligne]. Disponible: https://ieeexplore.ieee.org/document/840210.

[10] P. Topor, "DCF-Simpy: An IEEE 802.11 DCF simulation in Python with SimPy," *GitHub repository*, 2022. [En ligne]. Disponible: https://github.com/ToporPawel/DCF-Simpy.