



**HAL**  
open science

# Économie d'énergie sur la couche MAC pour les appareils Wi-Fi

Esther Guérin

► **To cite this version:**

Esther Guérin. Économie d'énergie sur la couche MAC pour les appareils Wi-Fi. Informatique [cs].  
Université Claude Bernard - Lyon I, 2024. Français. NNT : 2024LYO10242 . tel-04846311

**HAL Id: tel-04846311**

**<https://theses.hal.science/tel-04846311v1>**

Submitted on 18 Dec 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# THESE de DOCTORAT DE L'UNIVERSITE CLAUDE BERNARD LYON 1

**Ecole Doctorale N° 512  
Informatique et Mathématiques de Lyon (InfoMaths)**

**Discipline :**  
Informatique

Soutenue publiquement le 28/11/2024, par :  
**Esther Guérin**

---

## Économie d'énergie sur la couche MAC pour les appareils Wi-Fi

---

Devant le jury composé de :

|                          |   |                       |
|--------------------------|---|-----------------------|
| JAFFRES-RUNSER, Katia    | Professeure des Universités à l'Institut National Polytechnique de Toulouse | Rapporteuse           |
| THÉOLEYRE, Fabrice       | Directeur de recherche au CNRS  | Rapporteur            |
| DUPONT, Florent          | Professeur des Universités à l'Université Claude Bernard Lyon 1             | Examineur             |
| ROUSSEAU, Franck         | Maître de conférence à Grenoble INP - Ensimag                               | Examineur             |
| TEXIER, Géraldine        | Professeure des Universités à l'Institut Mines-Télécom Atlantique           | Présidente            |
| GUÉRIN LASSOUS, Isabelle | Professeure des Universités à l'Université Claude Bernard Lyon 1            | Directrice de thèse   |
| BEGIN, Thomas            | Professeur des Universités à l'Université Claude Bernard Lyon 1             | Co-directeur de thèse |



# RÉSUMÉ

Au fil des années, le Wi-Fi est devenu la technologie utilisée par défaut dans les réseaux locaux sans fil, le rendant incontournable dans les foyers comme dans les entreprises. Le Wi-Fi représente ainsi une consommation énergétique non négligeable à l'échelle du pays, d'autant plus qu'elle ne cesse de croître. Pour atteindre la neutralité carbone d'ici 2050, il est donc indispensable d'optimiser les dépenses énergétiques liées aux réseaux Wi-Fi.

Les standards IEEE 802.11 successifs, qui définissent le protocole Wi-Fi, ont proposé divers mécanismes d'économie d'énergie au long des années, et ce depuis sa création en 1999. Cependant, ces mécanismes sont rarement mis en œuvre par les appareils que nous utilisons au quotidien. Cela peut s'expliquer - entre autres - par le manque de clarté dans la description des mécanismes par les standards IEEE 802.11, ainsi que par l'absence d'indication quant à la configuration desdits mécanismes.

Dans cette thèse, nous identifions et décrivons les mécanismes d'économie d'énergie introduits par les différents standards IEEE 802.11. À partir de ces descriptions, nous classons et abstrayons les mécanismes d'économie d'énergie selon la méthode qu'ils utilisent. Plus précisément, les mécanismes d'économie d'énergie reposent tous sur des restrictions de trafic réseau : restreindre le trafic descendant d'une station (STA) permet à celle-ci d'éteindre temporairement sa radio pour économiser de l'énergie ; restreindre le trafic montant réduit la contention dans le réseau Wi-Fi. Cependant, restreindre un trafic présente un risque de perte de débit et d'augmentation du délai en cas de mauvaise configuration.

L'évaluation des mécanismes d'économie d'énergie nécessite donc de quantifier leur impact exact à la fois sur les performances réseau et sur la consommation énergétique des STAs. À notre connaissance, aucune étude ne compare l'ensemble des mécanismes d'économie d'énergie de manière équitable sur ces deux critères. Trois méthodes d'évaluation existent : une évaluation par modèle analytique, par expérimentation matérielle, ou par simulation. Un modèle analytique présente clairement les influences des différents facteurs sur les métriques, mais impose un certain niveau d'abstraction et est difficile à appliquer sur des flux hétérogènes. L'expérimentation matérielle donne des résultats holistiques et factuels, mais qui dépendent largement des appareils utilisés : elle restreint donc les scénarios explorables. Enfin, l'évaluation par simulation présente un compromis intéressant entre les deux premières méthodes : d'une part, elle autorise un niveau d'abstraction arbitraire, d'autre part elle ne limite pas les scénarios. Pour notre étude, nous avons utilisé les trois méthodes d'évaluation, avec une attention particulière pour l'évaluation par simulation et par modèle analytique.

Pour ce faire, nous avons construit et développé notre propre simulateur, nommé WE3S, spécialement dédié à l'étude des mécanismes d'économie d'énergie. Moins complexe que des simulateurs tels que ns-3, il nous permet de comparer les différents mécanismes d'économie d'énergie de manière équitable en se concentrant sur la manière dont ils restreignent le ou les trafics. Avec la compréhension que nous a apporté WE3S, nous avons pu construire un modèle analytique représentant les impacts de l'économie d'énergie sur le débit, le délai sur les trames et sur la consommation d'énergie des STAs.

Ces évaluations nous permettent ensuite d'identifier les facteurs qui mènent aux pertes de débit, et ceux qui favorisent l'économie d'énergie. Forts de cette connaissance, nous proposons une solution adaptative ainsi qu'une heuristique permettant de configurer les mécanismes d'économie d'énergie en fonction des trafics de la STA et des conditions du réseau de façon à économiser autant d'énergie que possible

sans perte de débit. En testant notre solution sur le simulateur WE<sub>3</sub>S, nous montrons que nos solutions permettent d'économiser jusqu'à 80 % d'énergie par rapport à un fonctionnement classique, à savoir sans mécanisme d'économie d'énergie.

## ABSTRACT

Over the years, Wi-Fi has become the default technology used in wireless local area networks, making it essential in homes and businesses alike. As a result, Wi-Fi represents a non-negligible energy consumption on the scale of a country, especially as it continues to grow. Therefore, in order to achieve carbon neutrality by 2050, Wi-Fi must take its share and reduce its energy consumption.

Ever since its inception in 1999, the different IEEE 802.11 standards, which define the Wi-Fi protocol, have proposed various energy-saving mechanisms over the years. However, these mechanisms are rarely implemented by the devices we use in our daily life. This can be explained - among other things - by the lack of clarity in the description of the mechanisms by the IEEE 802.11 standards, as well as by the absence of any indication as to how to configure the proposed mechanisms.

In this thesis, we identify and describe the energy-saving mechanisms introduced by the various IEEE 802.11 standards. Based on this description, we classify and abstract the energy-saving mechanisms according to the method they use. Specifically, energy-saving mechanisms are all based on network traffic restrictions: restricting the downstream traffic of a station (STA) allows it to temporarily switch off its radio to save energy whereas restricting upstream traffic reduces contention in the Wi-Fi network. However, restricting traffic carries the risk of reducing throughput and increasing delay if incorrectly configured.

Evaluating energy-saving mechanisms therefore requires quantifying their exact impact on both network performance and STA energy consumption. To the best of our knowledge, no study compares all energy-saving mechanisms equally on these two criteria.

Three evaluation methods exist: evaluation by analytical model, by hardware experimentation, or by simulation. An analytical model clearly presents the influences of the various factors on the metrics, but imposes a certain level of abstraction and is difficult to apply to heterogeneous flows. Hardware experimentation provides holistic and factual results, but these depend largely on the devices used: it therefore restricts the scenarios that can be explored. Lastly, the evaluation by simulation offers an interesting compromise between the first two methods: on the one hand, it allows an arbitrary level of abstraction, and on the other, it does not limit the scenarios that can be explored. For our study, we used all three methods, with a particular focus for the evaluation by simulation and by analytical model.

To this end, we have built and developed our own simulator, named WE<sub>3</sub>S, specifically dedicated to the study of energy-saving mechanisms. Less complex than simulators such as ns-3, it allows us to fairly compare different energy-saving mechanism, focusing on how they restrict traffic. With the understanding gained from WE<sub>3</sub>S, we were able to build an analytical model representing the impacts of energy saving on throughput, frame delay and energy consumption.

With this evaluation, we then identify the main factors that lead to throughput losses, and other factors that promote energy savings. With this knowledge, we propose an adaptive solution and a heuristic for configuring energy-saving mechanisms according to STA traffic and network conditions, so as to save as much energy as possible without losing throughput. By testing our solution on the WE<sub>3</sub>S simulator, we show that our solutions can save up to 80 % energy compared with conventional operation, *i.e.* without energy-saving mechanisms.

## REMERCIEMENTS

Cette thèse est le fruit de trois années de labeur, qui n'a été possible qu'au travers le concours de nombreuses personnes. Je souhaite exprimer ici ma gratitude à toutes celles et ceux qui ont contribué d'une manière ou d'une autre à ce travail.

Je tiens tout d'abord à remercier mes directeurs de thèse, Thomas et Isabelle, pour m'avoir guidée et sans cesse poussée à m'améliorer. Grâce à vous et à vos efforts, ma thèse s'est déroulée dans les meilleures conditions possibles. Ces trois années ont été riches d'enseignements, et il me tarde de continuer à les mettre en pratique.

Les membres de mon jury ont également toute ma gratitude pour avoir accepté d'évaluer ce travail : merci à Katia Jaffres-Runser et Fabrice Théoleyre pour le temps passé à rapporter cette thèse, et merci à Florent Dupont, Franck Rousseau et Géraldine Texier pour l'avoir examinée. J'en profite pour remercier Andrzej Duda, pour ses retours lors de mes comités de suivi.

Merci également à tous les membres du projet FACTO : Anne-Cécile, François, Jean-Philippe, Laurent, Martin, Meysam, Patrick, Rediet, Sandrine, Stéphane et Sylvie. J'ai beaucoup apprécié tous nos échanges lors de nos réunions de projet, qui n'ont jamais manqué de discussions.

Je n'oublie évidemment pas toutes les personnes que j'ai rencontrées au LIP. À mon éternelle complice : Meriem. À tous mes collègues, au détour d'un couloir ou autour d'un café : Théophile (et ses canards), Johann (et ses plantes vertes), Youssouph, Anthony, Augustin, Daniel, Francesco, Francescomaria, Gabriel, Joël, Loïc C. et Loïc D., Pascale, Russ, Théotime. Aux anciens, qui sont partis vers de nouveaux horizons : Anthony, Calvin, Jules, Mathieu, Samir. Aux MALIP pour leur efficacité et leur bonne humeur : Chiraz, Chrystelle, Élise, Laure, Leslie, Marie B. et Marie N. Merci à tous pour tous les bons moments partagés ensemble et pour toute la motivation que vous avez su m'apporter dans les moments difficiles. Je n'oublierai pas de sitôt l'ambiance que j'ai trouvée dans notre laboratoire.

Enfin, j'aimerais remercier ma famille qui m'a encouragée de loin, comme elle a pu. Papa, Maman, merci pour votre soutien indéfectible alors même que je ne rentrais à la maison que pour m'enfermer et continuer de travailler. Sans vous, je n'aurais jamais mené ce travail à bien.

À tous, et à d'autres que je suis sûre d'oublier, merci du fond du cœur.

# TABLE DES MATIÈRES

|       |  |    |
|-------|--|----|
| 1     | Introduction   | 1  |
| 1.1   | Motivations  | 1  |
| 1.2   | Organisation du manuscrit                            | 2  |
| 2     | État de l'art  | 5  |
| 2.1   | Description des standards                            | 5  |
| 2.2   | Consommation d'énergie des appareils Wi-Fi           | 6  |
| 2.3   | Étude des mécanismes d'économie d'énergie            | 7  |
| 2.3.1 | Standard "legacy"                                    | 7  |
| 2.3.2 | Amendement 802.11e                                   | 8  |
| 2.3.3 | Amendement 802.11ah                                  | 8  |
| 2.3.4 | Amendement 802.11ax                                  | 9  |
| 2.3.5 | Nos contributions                                    | 9  |
| 2.4   | Économie d'énergie hors standard                     | 9  |
| 3     | L'économie d'énergie en Wi-Fi                        | 11 |
| 3.1   | Le protocole Wi-Fi                                   | 11 |
| 3.1.1 | Les communications radio                             | 11 |
| 3.1.2 | Les bases du protocole Wi-Fi                         | 12 |
| 3.1.3 | Les améliorations du protocole Wi-Fi                 | 13 |
| 3.1.4 | La consommation d'énergie dans le Wi-Fi              | 15 |
| 3.2   | Stratégies d'économie d'énergie                      | 16 |
| 3.2.1 | Quel type de trafic restreindre?                     | 16 |
| 3.2.2 | Comment restreindre un trafic?                       | 16 |
| 3.2.3 | Nomenclature   | 18 |
| 3.3   | Mécanismes d'économie d'énergie                      | 18 |
| 3.3.1 | Standard "Legacy" (1999)                             | 20 |
| 3.3.2 | Amendement 802.11e (2005)                            | 21 |
| 3.3.3 | Amendement 802.11ah "HaLow" (2016)                   | 23 |
| 3.3.4 | Amendement 802.11ax (2021)                           | 27 |
| 4     | Méthodes d'évaluation                                | 31 |
| 4.1   | Simulation avec WE3S                                 | 31 |
| 4.1.1 | Présentation   | 32 |
| 4.1.2 | Validation du partage du médium avec ns-3            | 33 |
| 4.2   | Modèle analytique                                    | 34 |
| 4.2.1 | Méthode "slot"                                       | 35 |
| 4.2.2 | Méthode "prompt"                                     | 42 |
| 4.3   | Comparaison entre WE3S et le modèle analytique       | 45 |
| 4.3.1 | Débit  | 45 |
| 4.3.2 | Délai  | 46 |
| 4.3.3 | Consommation d'énergie                               | 47 |
| 4.4   | Évaluation expérimentale                             | 47 |
| 4.4.1 | Plateforme expérimentale                             | 47 |
| 4.4.2 | Obstacles  | 48 |
| 4.4.3 | Résultats  | 49 |
| 4.5   | Conclusion   | 49 |
| 5     | Évaluation de performances et guide de configuration | 51 |
| 5.1   | Restriction du trafic descendant                     | 51 |
| 5.1.1 | Discussions  | 51 |
| 5.1.2 | Étude des paramètres                                 | 57 |
| 5.2   | Restriction du trafic montant                        | 61 |
| 5.2.1 | Restriction par l'AP                                 | 62 |
| 5.2.2 | Décalage du trafic par la STA                        | 64 |

|       |  |    |
|-------|--|----|
| 5.3   | Configuration automatique de la restriction des trafics  | 69 |
| 5.3.1 | Description de la solution                               | 69 |
| 5.3.2 | Étude des performances de la solution                    | 70 |
| 5.3.3 | Discussions  | 75 |
| 5.4   | Conclusion   | 75 |
| 6     | Conclusion   | 77 |
| 6.1   | Résumé de thèse  | 77 |
| 6.2   | Travaux futurs   | 77 |
| 6.2.1 | Élargir les stratégies d'économie d'énergie              | 77 |
| 6.2.2 | Rapprocher les métriques de performance de l'utilisateur | 78 |
| 6.2.3 | Réduire les délais additionnels                          | 78 |
|       | Bibliographie  | 81 |



## 1.1 MOTIVATIONS

Le numérique fut une révolution qui s'est infiltrée dans tous les aspects de notre vie quotidienne. Encore aujourd'hui, son ubiquité se manifeste dans la question environnementale : les fronts pour réduire son impact sur notre planète sont nombreux, et pressants. En début de vie, la fabrication des appareils électroniques demande un volume d'eau colossal, notamment au vu des quantités à produire. Au cours de leur vie, ces appareils engloutissent de l'énergie pour fonctionner ; en particulier quand ils proposent des services en datacenters, véritables gouffres d'électricité. Et en fin de leur vie, ces appareils produisent beaucoup de déchets qui ne sont pas ou peu valorisés. La neutralité carbone ne peut être atteinte qu'en abordant les fronts un par un, solution après solution, et en faisant compter chaque amélioration.

Dans cette thèse, nous nous intéressons à la consommation d'énergie des réseaux Wi-Fi. En effet, le Wi-Fi représente aujourd'hui le protocole de communication sans fil le plus utilisé dans les réseaux locaux, domestiques et professionnels. La Wi-Fi Alliance, en charge de la certification des appareils Wi-Fi, estime en 2024 leur nombre en utilisation à plus de 21 milliards (*Site de la Wi-Fi Alliance s. d.*). Ainsi, toute réduction de la consommation d'énergie, même mineure, a des répercussions majeures si elle peut être appliquée à tous les appareils Wi-Fi.

Le standard IEEE 802.11, à l'origine du protocole de communication qu'est le Wi-Fi, le définit sur les deux couches réseaux les plus basses : sur la couche physique, notée PHY, qui définit entre autres comment les signaux radio doivent être modulés, et sur la couche "liaison de données", comprenant dans le cas du Wi-Fi le *Medium Access Control* (MAC), qui détermine comment le médium radio est partagé entre les appareils. Les deux couches abritent chacune des mécanismes dédiés à l'économie d'énergie.

Nous avons fait le choix de nous concentrer sur les mécanismes d'économie d'énergie de la couche MAC qui reposent sur la restriction d'un ou plusieurs trafics des stations (STAs), c'est-à-dire qu'ils interdisent ces trafics sur certaines périodes de temps. Ces mécanismes ont la particularité de proposer un compromis : en échange d'un sacrifice sur les performances réseaux des STAs, ils peuvent réduire la consommation d'énergie du WLAN, soit en réduisant la contention, soit en permettant aux STAs d'hiberner (c'est-à-dire d'éteindre temporairement leur carte Wi-Fi). Ce choix a été guidé par le projet ANR FACTO, qui a financé cette thèse. En effet, l'un de ses objectifs est de promouvoir le Wi-Fi comme technologie pour les appareils dits "Internet of Things" (IoT), et l'une de leurs exigences est de pouvoir hiberner sur de longues durées pour sauvegarder au maximum leur énergie. Les mécanismes d'économie d'énergie restreignant le trafic sont les seuls mécanismes à pouvoir permettre cela, ce sont donc ceux que nous avons étudiés.

Le standard IEEE 802.11 définit un nombre significatif de mécanismes d'économie d'énergie, mais dans les faits, aucun ne semble être utilisé de manière généralisée. Pour poser ce constat, nous nous basons sur les programmes de certification de la Wi-Fi Alliance, un consortium d'entreprises comprenant entre autres Apple, Cisco ou Qualcomm, qui décide des fonctionnalités à intégrer pour être certifié Wi-Fi. D'une part, peu de mécanismes d'économie d'énergie sont implémentés sur les cartes Wi-Fi. Parmi les mécanismes d'économie d'énergie que nous étudions (et détaillons) dans cette thèse, la Wi-Fi Alliance propose à la certification seulement deux mécanismes pour les STAs : le "legacy Power Save Mode" (legacy PSM), introduit en 1999 dans le programme de certification "WMM-Power Save" et le "Target Wake

Time” (TWT), introduit en 2021 dans le programme de certification du Wi-Fi 6. Il est difficile d’expliquer pourquoi les autres mécanismes ont été ignorés par la Wi-Fi Alliance et de fait, par les constructeurs. L’une des raisons pourrait être le manque d’étude comparative entre les différents mécanismes, ce qui aurait permis de mettre en valeur les gains potentiels des mécanismes ignorés. Ceci couplé au succès mitigé du legacy PSM a dû amener les constructeurs à se concentrer sur d’autres aspects plus prometteurs du Wi-Fi pendant près de vingt ans. Il est cependant à noter que la tendance semble changer. Les derniers programmes de certification de la Wi-Fi Alliance portent une attention toute particulière sur TWT et ses variantes, ce qui dénote un certain intérêt pour des mécanismes d’économie d’énergie de la part des constructeurs.

D’autre part, les deux mécanismes qui sont effectivement implémentés sur des cartes Wi-Fi ne sont, pour l’instant, que rarement activés. Dans le cas du “legacy PSM”, ses failles sont nombreuses et profondes : en plus de réduire drastiquement les performances réseaux des STAs (I.-G. Lee et al., 2020), il présente une faille de sécurité ouvrant sur un potentiel déni de service (SOUILAH2014). Il est donc assez naturel de ne pas utiliser ce mécanisme bien qu’il soit disponible. En ce qui concerne le mécanisme TWT, il faut d’abord noter qu’il est nettement plus récent – il a été introduit en 2021 – et que par conséquent, seuls les appareils Wi-Fi les plus récents peuvent l’implémenter. Il est donc plus que probable que son utilisation évolue dans les années à venir. Néanmoins, à l’heure actuelle, TWT semble contre-performer sur les produits commercialisés : il réduit les débits atteints par les STAs sans diminuer leur consommation d’énergie (Zhao et al., 2023). Nous imputons ces résultats à une mauvaise configuration de TWT qui empêche les STAs d’hiberner. Dans tous les cas, TWT est un mécanisme bien plus versatile que le legacy PSM, mais cela implique aussi qu’il a beaucoup plus de paramètres à configurer ; et le standard IEEE 802.11 ne donne aucune indication à cet effet.

Pour résumer, les entraves à l’utilisation effective des mécanismes d’économie d’énergie sont les suivantes :

- ❖ l’absence de comparaison entre les mécanismes d’économie d’énergie,
- ❖ le manque de guide de configuration,
- ❖ l’étude de leurs impacts.

Pendant ma thèse, j’ai donc essayé de combler ces manques pour offrir une vision plus globale des mécanismes d’économie d’énergie dans le protocole Wi-Fi.

## 1.2 ORGANISATION DU MANUSCRIT

Cette thèse propose un guide pour choisir et utiliser efficacement les mécanismes d’économie d’énergie de la couche MAC du Wi-Fi qui reposent sur la restriction d’un ou de plusieurs trafics. L’organisation du manuscrit est la suivante :

- ❖ Le chapitre 2 présente l’état de l’art de la recherche sur l’économie d’énergie.
- ❖ Le chapitre 3 décrit les possibilités d’économie d’énergie offertes par le protocole Wi-Fi que nous considérons dans notre étude. Cela comprend un descriptif des mécanismes d’économie d’énergie proposés dans les différents amendements du standard IEEE 802.11, ainsi qu’une modélisation desdits mécanismes pour simplifier notre étude en ne gardant que les rouages essentiels. Nous appelons ces modélisations les stratégies d’économie d’énergie ; nous en avons identifiés sept.
- ❖ Le chapitre 4 présente les méthodes que nous avons employées pour évaluer les stratégies d’économie d’énergie. La première méthode est un simulateur, nommé WE<sub>3</sub>S, que nous avons construit spécialement pour l’étude des stratégies d’économie d’énergie. La seconde méthode est un modèle analytique que nous avons dérivé de nos observations sur WE<sub>3</sub>S. La dernière méthode

est une évaluation expérimentale, qui a mis en lumière les obstacles à l'implémentation des stratégies d'économie d'énergie sur un appareil Wi-Fi.

- ❖ Le chapitre 5 analyse les performances obtenues tour à tour par les stratégies restreignant le trafic descendant et les stratégies restreignant le trafic montant. Nous extrayons de cette analyse un guide d'utilisation et de configuration des stratégies d'économie d'énergie que nous complétons avec une solution automatique d'économie d'énergie.
- ❖ Enfin, le chapitre 6 conclut ce manuscrit.



Dans ce chapitre, nous posons l'état de l'art de l'économie d'énergie dans le Wi-Fi. L'organisation de ce chapitre est la suivante. Nous commençons par regarder les informations générales relatives à l'économie d'énergie : en section 2.1, nous regardons les standards IEEE 802.11 qui définissent le Wi-Fi, ainsi que les études qui les vulgarisent, tandis qu'en section 2.2, nous détaillons les études menées sur la consommation des appareils Wi-Fi. Ensuite, les sections 2.3 et 2.4 s'intéressent aux études menées sur les méthodes permettant d'économiser de l'énergie, respectivement avec des mécanismes du standard IEEE 802.11 ou avec des solutions hors standard.

## 2.1 DESCRIPTION DES STANDARDS

Le Wi-Fi est un protocole de communication radio pour les réseaux locaux sans fil. Le protocole Wi-Fi est défini par la Wi-Fi Alliance en se basant sur les standards IEEE 802.11 (*Institute of Electrical and Electronics Engineers*). Les standards IEEE 802.11 offrent un éventail de fonctionnalités – que les appareils sont libres d'implémenter ou non – en s'assurant que chaque fonctionnalité soit interopérable avec le reste du standard IEEE 802.11. Plusieurs standards IEEE 802.11 ont vu le jour au fil des années. En effet, l'IEEE publie régulièrement un nouveau standard 802.11 qui supplante les précédents. À noter tout de même que tous les standards IEEE 802.11 sont rétro-compatibles et que par conséquent, la publication d'un nouveau standard ne rend pas obsolètes les appareils suivant une des anciennes versions. À l'heure actuelle, le dernier standard en date est le standard IEEE 802.11-2020, corrigé en 2022. ("[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 - Corrigendum 1 – Correct IEEE 802.11ay Assignment of Protected Announce Support bit](#)" 2022).

Le standard IEEE 802.11 évolue par le biais des TG (*Task Group*), des groupes de travail qui se focalisent sur l'ajout d'une ou plusieurs fonctionnalités. Pour intégrer la nouvelle fonctionnalité, les TGs créent des amendements au standard : ce sont des documents présentant le différentiel entre le dernier standard publié et ce même standard intégrant la ou les nouvelles fonctionnalités. Les amendements sont souvent désignés par une ou deux lettres minuscules (par exemple : 802.11e ou 802.11ax). Lorsqu'un nombre suffisant d'amendements est publié, ils sont regroupés dans une nouvelle version du standard IEEE 802.11.

La Wi-Fi Alliance, un large consortium d'entreprises comprenant entre autres Apple, Cisco ou Qualcomm, sélectionne un sous-ensemble des fonctionnalités proposées par le standard IEEE 802.11 et de ses amendements pour créer un programme de certification. Les fabricants de cartes Wi-Fi peuvent ensuite payer la Wi-Fi Alliance pour soumettre leur produit à un ou plusieurs programmes de certification. Lorsqu'un produit passe tous les tests d'un programme de certification, il gagne le droit d'arborer le macaron "Wi-Fi certified" (en figure 1), et est listé sur le site de la Wi-Fi Alliance (cf. [Site de la Wi-Fi Alliance s. d.](#)).

Notre étude s'appuie sur plusieurs standards et amendements IEEE 802.11, à savoir : le standard dit "legacy" ([IEEE WG 802.11, 1999](#)), l'amendement 802.11e ([IEEE WG 802.11e, 2005](#)), l'amendement 802.11ah ([IEEE WG 802.11ah, 2016](#)) et l'amendement 802.11ax ([IEEE WG 802.11ax, 2021](#)). Ces documents décrivent, entre autres, les



FIGURE 1 – Macaron “Wi-Fi certified” délivré par la Wi-Fi Alliance.

mécanismes d'économie d'énergie que nous présentons de manière moins aride et plus synthétique en chapitre 3 et dans notre article [Guérin, Begin et Lassous, 2023](#).

D'autres études font le travail de vulgarisation des standards IEEE 802.11, mais chacune avec une perspective différente. On peut par exemple citer [Tian, Santi et al., 2021](#) qui décrit l'amendement 802.11ah, [Khorov et al., 2018](#) et [Mozaffariahrar et al., 2022](#) l'amendement 802.11ax, ou [Deng et al., 2020](#) l'amendement 802.11be. À notre connaissance, notre étude est la seule qui se concentre exclusivement sur l'économie d'énergie dans les standards IEEE 802.11, en offrant une vue d'ensemble au travers du temps.

Dans tous les cas, ces études sont très utiles pour appréhender les standards IEEE 802.11 qui restent sinon obscurs à moins d'y consacrer une quantité de temps non négligeable. Cependant, ces études n'ont généralement pas vocation à aller au-delà du standard, elles ne donnent donc pas d'indications précises sur quand et comment utiliser les différents mécanismes du standard. Par exemple, elles décrivent les paramètres des mécanismes, mais ne précisent généralement pas la valeur à lui donner.

## 2.2 CONSOMMATION D'ÉNERGIE DES APPAREILS WI-FI

Un sujet crucial à notre étude est l'étude de la consommation d'énergie des appareils Wi-Fi. Nous détaillons dans cette section les études que nous avons trouvées sur le sujet.

Pour la consommation des STAs, il faut distinguer le Wi-Fi “classique” du Wi-Fi HaLow, issu de l'amendement 802.11ah et dédié à l'IoT. Pour les STAs du Wi-Fi “classique”, les études ne sont pas nombreuses à notre connaissance. Parmi les plus récentes, nous comptons [Halperin et al., 2010](#) qui détaille clairement un certain nombre de mécanismes de la couche PHY, comme la largeur du canal radio, le nombre d'antennes utilisées ou la puissance d'émission, et les relie avec la consommation d'énergie qu'ils occasionnent, le tout par évaluation expérimentale. On peut également noter [Venkateswaran, Tai, Ben-Yehzekel et al., 2022](#) qui conduit aussi une évaluation expérimentale et en dérive un modèle analytique qu'il ajoute au simulateur ns-3. Enfin, [Serrano et al., 2014](#) introduit la notion de crossfactor. Quand un paquet est envoyé ou reçu, il doit traverser la pile réseau, chaque couche de la pile réseau nécessitant des calculs et donc de l'énergie. Par conséquent, chaque paquet représente un coût énergétique que les auteurs baptisent crossfactor. Si les études restent toujours valables sur leurs principes, il est probable que les ordres de grandeur des valeurs avancés ne soient plus d'actualité, vu l'âge des études. En particulier, d'après nos informations, la consommation d'énergie d'une carte Wi-Fi inactive (c'est-à-dire, allumée mais ne recevant ni n'émettant de signal radio), a largement diminué, et se rapprocherait maintenant plus de la consommation d'énergie de la carte Wi-Fi en hibernation (c'est-à-dire, éteinte à part l'horloge).

Sur le Wi-Fi HaLow, les études sont plus nombreuses et plus récentes aussi, car la contrainte sur la consommation d'énergie est plus forte pour les appareils IoT. Nous pouvons citer par exemple [Maudet et al., 2023](#) et [Maudet et al., 2024](#), où les auteurs mènent et analysent la consommation d'énergie d'appareils IoT utilisant

le Wi-Fi HaLow, par évaluation expérimentale. [Morin et al., 2017](#) modélise à plus haut niveau la consommation d'énergie des appareils IoT, selon la technologie de communication sans fil utilisée, comprenant entre autres le Wi-Fi "classique" et le Wi-Fi HaLow. Leur étude permet ensuite de comparer ces technologies entre elles en termes de durée de vie de la batterie des appareils.

Enfin, d'autres études s'intéressent à la consommation d'énergie des APs. [Silva et al., 2019a](#) propose un état de l'art très complet de la consommation d'énergie des APs allant de l'amendement 802.11a à l'amendement 802.11ac. Les auteurs complètent ensuite leur étude dans [Silva et al., 2019b](#) en proposant une solution multi-APs : ils proposent d'éteindre certains APs peu utilisés (par exemple pendant la nuit) tout en maintenant la couverture Wi-Fi complète. L'étude de [Dembele et al., 2023](#), un peu plus récente, développe un modèle analytique pour la consommation d'énergie des APs, tiré d'une évaluation expérimentale. Enfin, [Ozkaya et al., 2023](#) cherche à réduire la consommation des APs en provoquant des temps de silence pendant lesquels l'AP peut hiberner, d'une manière très proche de celle que nous appliquons pour les STAs dans notre étude. Cependant, le standard IEEE 802.11 n'autorise pas les APs à hiberner, contrairement aux STAs, car ils doivent rester à la disposition des STAs entrantes qui sont impossibles à prévoir. Les auteurs testent leur solution avec des simulations ns-3.

## 2.3 ÉTUDE DES MÉCANISMES D'ÉCONOMIE D'ÉNERGIE

Dans cette section, nous présentons les recherches qui ont été menées sur les différents mécanismes d'économie d'énergie que nous avons identifiés. Comme ces mécanismes ont été introduits dans des standards et/ou amendements IEEE 802.11 différents, ils ont été examinés avec des sensibilités différentes. Pour faciliter la lecture, nous avons donc rassemblés les mécanismes d'économie d'énergie selon l'amendement IEEE 802.11 qui les a introduits, en indiquant au début de chaque section la sensibilité de l'époque.

### 2.3.1 Standard "legacy"

Lorsque le standard "legacy" a été introduit en 1999, beaucoup d'études de l'époque ont étudié la couche MAC dans son intégralité (par exemple [Woesner et al., 1998](#)), plutôt que de se concentrer sur un de ses mécanismes en particulier. Le standard "legacy" a introduit le legacy PSM (*Power Save Mode*), un mécanisme d'économie d'énergie qui a malgré tout bénéficié de quelques études décrites ci-dessous.

Nous pouvons citer [Anastasi et al., 2004b](#) et [Anastasi et al., 2004a](#), qui ont développé un modèle analytique pour prédire le débit et la consommation d'énergie avec le legacy PSM, puis ont validé ce modèle avec des simulations. Les auteurs développent ensuite dans [Anastasi et al., 2008](#) un nouveau mécanisme d'économie d'énergie qui englobe plusieurs couches de la pile réseau.

En parallèle, [Wu et S.-C. Chen, 2011](#) et [He et Yuan, 2009](#) identifient chacun certaines limitations du legacy PSM, et développent un nouveau mécanisme d'économie d'énergie en réponse. [Wu et S.-C. Chen, 2011](#) étudient le débit, le délai et la consommation d'énergie, avec des simulations Qualnet. [He et Yuan, 2009](#) considèrent le délai et la consommation d'énergie dans leur étude, qu'ils mènent principalement avec le simulateur ns-2. Ces limitations comprennent par exemple le fait que les trames descendantes soient récupérées une par une, sans possibilité de les agréger, ou bien le fait que le legacy PSM crée un pic de contention en forçant les STAs à essayer de récupérer leurs trames descendantes en attente au même moment. Il est à noter que les limitations découvertes par les auteurs ont depuis été comblées par différents mécanismes.

Enfin, les études plus récentes de [Souilah et al., 2014](#) et de [I.-G. Lee et al., 2020](#) prouvent que le legacy PSM présente des failles de sécurité qui restent ouvertes à ce jour. En effet, les trames utilisées par le legacy PSM ne sont pas chiffrées, et rendent donc les STAs vulnérables aux attaques par déni de service.

### 2.3.2 Amendement 802.11e

L'amendement 802.11e a vu l'arrivée de la QoS (*Quality of Service*) dans le Wi-Fi. En particulier, l'amendement 802.11e définit différentes classes d'accès pour le trafic téléphonique et vidéo. Cela transparait dans les scénarios qui incluent généralement au moins l'une de ces catégories d'accès. En ce qui concerne les mécanismes d'économie d'énergie, l'amendement 802.11e en a introduit deux : S-APSD (*Scheduled Automatic Power Save Delivery*) et U-APSD (*Unscheduled Power Save Delivery*).

Dans [Pérez-Costa et Camps-Mur, 2006](#) et [Pérez-Costa, Camps-Mur et Vidal, 2007](#), les auteurs comparent U-APSD avec le legacy PSM sur le débit, le délai et la consommation d'énergie avec des simulations sur le simulateur OPNET, et proposent une solution de configuration pour le mécanisme U-APSD. Dans [Pérez-Costa et Camps-Mur, 2010](#), les mêmes auteurs mènent une étude plus complète en incluant également le mécanisme S-APSD et en proposant des heuristiques permettant de configurer S-APSD et U-APSD.

À la suite de ces études, [T.-H. Lee et Hsieh, 2012](#) améliorent une des heuristiques de [Pérez-Costa et Camps-Mur, 2010](#) pour configurer S-APSD en proposant un modèle analytique et en le validant par simulation.

Enfin, [Camps-Mur et al., 2012](#) analyse et compare par simulations OPNET les performances de U-APSD avec le legacy PSM, sur le débit, le délai et la consommation d'énergie au regard de l'agrégation de trames, apparue dans le standard IEEE avec l'amendement 802.11n.

### 2.3.3 Amendement 802.11ah

L'amendement 802.11ah, que la Wi-Fi Alliance a baptisé "Wi-Fi HaLow", est dédié à l'intégration des appareils IoT dans les réseaux Wi-Fi. L'une des particularités des appareils IoT est d'être généralement nombreux, mais avec des débits faibles. L'amendement 802.11ah introduit principalement le mécanisme d'économie d'énergie RAW (*Restricted Access Window*), mais aussi le mécanisme TWT (*Target Wake Time*).

[Dong et al., 2016](#) cherchent à résoudre le problème des nœuds cachés (c'est-à-dire, quand des STAs d'un même WLAN sont trop éloignées l'une de l'autre pour s'entendre, et risquent donc d'interférer entre elles au niveau de l'AP). Les auteurs rassemblent les STAs d'une même zone géographique dans des mêmes groupes RAW, pour qu'ainsi les STAs susceptibles d'entrer en collision puissent s'entendre. Les auteurs évaluent ensuite leur solution au niveau du débit, en utilisant un modèle analytique et en le validant par simulation MATLAB.

En parallèle, [Kureev et al., 2017](#) s'attaquent à un autre problème, qui est la cohabitation des STAs demandant un faible débit avec des STAs demandant un débit élevé. Les auteurs proposent un modèle analytique pour aider à la configuration du mécanisme RAW et évaluant à la fois le débit et la consommation d'énergie des STAs.

Enfin, [Santi et al., 2019](#) évaluent et comparent les mécanismes du legacy PSM couplé à RAW, avec le mécanisme TWT, du point de vue des débits, des délais sur les trames et de la consommation d'énergie. Pour ce faire, ils utilisent un modèle analytique qu'ils valident avec des simulations ns-3.

### 2.3.4 Amendement 802.11ax

L'amendement 802.11ax a introduit l'OFDMA, une technique de transmission radio qui permet à plusieurs STAs de communiquer simultanément. L'effort des études menées sur l'amendement 802.11ax se concentre donc généralement sur l'amélioration de l'utilisation du médium radio.

Nous pouvons par exemple citer [Q. Chen, Liang et al., 2019](#), [Q. Chen et Zhu, 2020](#) et [Nurchis et Bellalta, 2019](#), qui cherchent à optimiser le nombre de transmissions parallèles dans un WLAN grâce à l'OFDMA et à TWT. Ils évaluent leur solution avec des métriques comme le débit à saturation ou le taux d'utilisation du médium, mais négligent entièrement la consommation d'énergie qui ne fait pas partie de leurs objectifs.

Malgré tout, le mécanisme TWT a aussi été exploré pour économiser l'énergie des STAs. Pour [Yang et al., 2021](#), le mécanisme TWT revient en quelques sortes à superposer du TDMA (*Time Division Multiple Access*) sur l'accès au médium par CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) du Wi-Fi. Les auteurs proposent alors d'explorer les différents algorithmes classiques pour définir et allouer les créneaux TDMA aux STAs. Ils testent leurs solutions du point de vue du débit, de l'équité et de la consommation d'énergie, en utilisant des simulations ns-3.

Du fait des longues périodes d'hibernation qu'il autorise, le mécanisme TWT peut provoquer des dérives d'horloge au niveau des STAs. Pour résoudre ce problème, [Stepanova et al., 2020](#) explorent la combinaison du mécanisme TWT avec une "wake up radio" (c'est-à-dire une radio secondaire, à faible consommation d'énergie, qui à un certain signal réveille la radio Wi-Fi principale).

L'une des particularités du mécanisme TWT est qu'il est implémenté et certifié sur des appareils Wi-Fi commercialisés. Cela a permis des évaluation des performances du mécanisme TWT sur des appareils réels, comme l'ont fait [Liu et Choi, 2023](#) ou [Zhao et al., 2023](#).

### 2.3.5 Nos contributions

Toutes ces études s'attachent chacune à quelques mécanismes d'économie d'énergie, mais aucune n'en offre une vue d'ensemble. De plus, les études qui cherchent à configurer intelligemment un mécanisme d'économie d'énergie le font en se basant sur des scénarios spécifiques, et par conséquent avec un objectif bien précis en tête.

Notre étude se positionne à un plus haut niveau pour mieux appréhender les différents impacts des mécanismes d'économie d'énergie dans leur globalité, avec un degré de détails suffisant pour que nos résultats soient génériques et puissent généralement s'appliquer, quels que soient le scénario et le mécanisme d'économie d'énergie considérés. Ainsi, notre étude est la seule à comparer l'ensemble des mécanismes d'économie d'énergie.

## 2.4 ÉCONOMIE D'ÉNERGIE HORS STANDARD

Quelques études proposent des mécanismes d'économie d'énergie en dehors des standards IEEE. Nous les présentons dans cette section.

[Jang et al., 2011](#) développe un mécanisme appelé SNOOZE, compatible avec les standards. En utilisant SNOOZE, les STAs sont capables de moduler la durée de leur hibernation en prédisant l'arrivée des prochaines trames descendantes. La prédiction en elle-même est simple à mettre en œuvre : il s'agit de mesurer le dernier intervalle inter-trames du trafic descendant de la STA, et de supposer que le prochain intervalle inter-trame aura la même valeur. De plus, les auteurs essaient d'optimiser le nombre d'antennes utilisées par les STAs, suivant le constat que les antennes sont

un des facteurs qui peuvent augmenter significativement la consommation d'énergie d'une STA. Les auteurs testent SNOOZE sur des appareils Wi-Fi réels, avec des résultats intéressants en termes de débit, de délai et de consommation d'énergie. Cependant, contrairement à notre étude, SNOOZE ne repose sur aucun mécanisme d'économie d'énergie du standard, bien qu'il soit compatible avec celui-ci (c'est-à-dire que SNOOZE n'entrave pas le bon fonctionnement d'un réseau Wi-Fi). En d'autres termes, si une STA veut utiliser SNOOZE, il faut que l'AP ait été modifié au préalable pour le supporter.

De manière assez similaire, [Sheth et al., 2021](#) utilise du machine learning pour prédire quand arrivent les prochaines trames descendantes d'une STA. Du fait du coût calculatoire d'un modèle de machine learning, les auteurs déplacent la prédiction de trames au niveau de l'AP qui doit ensuite indiquer aux STAs la durée de leur hibernation, avec une trame non standardisée. Les auteurs mesurent à la fois la consommation d'énergie et le délai sur les trames, avec des appareils Wi-Fi réels. Malheureusement, les auteurs ont choisi de comparer leur solution avec le legacy PSM, qui est reconnu pour ses limitations (cf. section 2.3.1), plutôt qu'avec des mécanismes d'économie d'énergie plus récents comme U-APSD.

Enfin, [Saeed et Kolberg, 2018](#) proposent une solution permettant de classifier les applications en utilisation sur un smartphone, et d'adapter le mécanisme d'économie d'énergie et sa configuration en fonction de la classe détectée. S'il est vrai que les mécanismes d'économie d'énergie choisis par les auteurs ne sont pas les plus pertinents, en particulier le legacy PSM dont les failles ont été prouvées, nous remarquons qu'ils proposent une estimation du délai tolérable par l'utilisateur, ce qui est une donnée précieuse dans le cadre de l'économie d'énergie.

# 3

## L'ÉCONOMIE D'ÉNERGIE EN WI-FI

### 3.1 LE PROTOCOLE WI-FI

Le Wi-Fi est un protocole de communication radio pour les réseaux locaux sans fil. Il impose certaines règles aux appareils pour assurer des communications compréhensibles et un partage équitable des ressources. Le protocole Wi-Fi concerne deux couches réseau : la couche *physique* (PHY), qui décrit entre autres comment moduler un signal radio pour que deux appareils Wi-Fi puissent se comprendre, et la couche liaison de données avec le MAC (*Medium Access Control*), qui organise les temps de parole des appareils. Cette section présente le fonctionnement global du Wi-Fi. Compte tenu du nombre immense de règles et fonctionnalités proposées par le protocole Wi-Fi, nous restreignons cette description aux aspects du Wi-Fi nécessaires pour comprendre cette thèse. Pour une description plus exhaustive du fonctionnement, nous invitons le lecteur à lire les livres de [Gast, 2013](#).

#### 3.1.1 Les communications radio

Le médium radio est un support de communication, au même titre que les fils électriques ou optiques. Un appareil source peut moduler un signal sur les ondes radio, qui peut ensuite être démodulé par l'appareil destination : les deux appareils communiquent. La manière de moduler les ondes radio a énormément évolué au fil des années, permettant sans cesse d'envoyer plus de données en moins de temps. Cependant, l'une des particularités du médium radio reste son manque de fiabilité. Toute interférence, de courte ou longue durée, peut rendre inintelligible un signal radio. Notamment, si deux appareils proches émettent un signal radio simultanément sur la même bande de fréquence, ils interfèrent entre eux. On ne peut alors plus comprendre ni l'un ni l'autre (ou alors, difficilement). Les protocoles de communication radio doivent donc mettre en place des mécanismes pour éviter ces problèmes, et plus généralement pour assurer la bonne transmission des signaux radio.

Dans le cas du Wi-Fi, les signaux ne sont pas continus comme pour la radio FM, par exemple. Les appareils Wi-Fi envoient ponctuellement des messages, et restent silencieux entre ces messages. On appelle *trames* les messages que s'échangent les appareils Wi-Fi. Pour assurer la bonne transmission d'une trame, l'appareil destination doit retourner un accusé de réception (ACK) à l'appareil source pour chaque trame correctement reçue. Lorsque l'appareil source reçoit un ACK, il est sûr que sa trame a été correctement reçue. Si en revanche il ne reçoit pas d'ACK, l'appareil source considère que la transmission de cette trame a été un échec et il réessayera plus tard de la retransmettre.

L'autre particularité du Wi-Fi est que la bande de fréquence utilisée est décidée à l'avance. Par exemple, si deux appareils Wi-Fi d'un même réseau (et donc sur la même bande de fréquence) essaient d'envoyer simultanément une trame vers un même appareil destination, ils vont interférer entre eux (à moins qu'ils n'utilisent de l'OFDMA – pour *Orthogonal Frequency-Division Multiple Access*, ce que nous présentons en section 3.1.3). Dans ce cas, au moins une des deux trames n'est pas reçue correctement (et dans le pire cas, aucune des deux trames n'est reçue correctement). On parle alors de *collision*. Comme les trames qui ne sont pas reçues correctement doivent être retransmises plus tard, les collisions sont synonymes de perte de per-

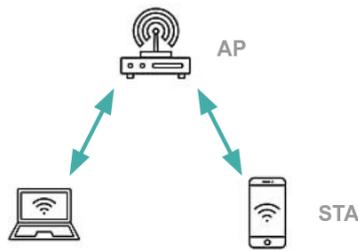


FIGURE 2 – Schéma d'un réseau Wi-Fi en infrastructure.

formances. Éviter les collisions est donc important pour un réseau efficace : c'est le rôle de la couche MAC.

### 3.1.2 Les bases du protocole Wi-Fi

#### *Le mode infrastructure*

Dans cette thèse, nous nous intéressons exclusivement aux réseaux Wi-Fi en *mode infrastructure*, car ils représentent l'écrasante majorité de ceux utilisés de par le monde. Les réseaux Wi-Fi en mode infrastructure sont composés d'un point d'accès, nommé AP pour *Access Point*, et d'une ou plusieurs stations, abrégées en STAs, qui lui sont associées. Dans un réseau Wi-Fi domestique, l'AP est la box internet tandis que les STAs sont les appareils connectés, comme par exemple les ordinateurs ou les smartphones.

Les réseaux Wi-Fi en mode infrastructure ont une topologie en étoile avec l'AP au centre, comme schématisé en figure 2. Autrement dit, les STAs ne sont autorisées à communiquer qu'avec l'AP ; deux STAs n'ont pas le droit de communiquer directement l'une avec l'autre (même si elles le peuvent physiquement). On distingue alors deux types de transmission : les transmissions montantes, qui vont d'une STA vers l'AP, et les transmissions descendantes, qui vont de l'AP vers une ou plusieurs STAs.

Par la suite, nous appelons WLAN (*Wireless Local Area Network*) les réseaux Wi-Fi en mode infrastructure.

Dans les WLANs, l'AP diffuse régulièrement des trames particulières, appelées *beacons*, pour annoncer le réseau à de potentielles futures STAs. Ce sont par exemple les beacons qui annoncent le nom du WLAN aux appareils, ce qui apparaît quand l'utilisateur cherche à se connecter à un réseau Wi-Fi. Ces beacons servent aussi à transmettre des informations aux STAs déjà associées à l'AP. Il est donc demandé aux STAs d'écouter régulièrement les beacons pour prendre connaissance de ces informations.

#### CSMA/CA

Le protocole Wi-Fi propose l'algorithme DCF (*Distributed Coordination Function*) pour distribuer le temps de parole aux appareils Wi-Fi (AP et STAs). L'algorithme DCF s'appuie sur la technique dite du CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Il poursuit un double objectif : d'une part, éviter les collisions, d'autre part, garder la distribution des temps de parole souple (pour s'adapter aux trafics fluctuants des STAs, et aux départs et arrivées de STAs).

Le principe du CSMA/CA est le suivant. Lorsqu'un appareil a une trame à émettre, il commence par écouter le médium radio. Si le médium est libre, l'appareil peut commencer immédiatement sa transmission. Sinon, il commence par attendre que le médium se libère. Ensuite, il attend encore une durée fixe appelée AIFS (*Arbitration Inter-Frame Space*), puis une durée aléatoire déterminée par le *backoff*. Cette durée est décidée en tirant le *backoff*, une valeur aléatoire dans la *fenêtre*

de contention, puis en multipliant cette valeur par la durée d'un créneau de backoff (de l'ordre de quelques microsecondes). Si un autre signal radio occupe le médium alors que l'appareil n'a pas fini de décrémenter son backoff, celui-ci suspend le décompte le temps que le médium se libère de nouveau.

En utilisant le CSMA/CA, deux appareils Wi-Fi peuvent rentrer en collision s'ils ont tiré la même valeur de backoff. La probabilité de collisions augmente avec le nombre de STAs qui essaient d'accéder au médium après une transmission donnée. On dit que la contention est élevée lorsqu'un nombre significatif de STAs sont en compétition pour le médium via le CSMA/CA, ce qui entraîne un risque élevé de collision.

Pour limiter la contention, la fenêtre de contention n'est pas de taille fixe. Sa borne inférieure est toujours à zéro, mais sa borne supérieure varie dans le temps en fonction des transmissions de l'appareil, en suivant un algorithme appelé "Binary Exponential Backoff". Lorsque l'appareil a essayé de transmettre une trame après avoir attendu son backoff, il sait si la transmission a été un succès ou non grâce à l'éventuel ACK que lui renvoie l'appareil destination. S'il ne reçoit pas d'ACK, il va doubler la borne supérieure de sa fenêtre de contention. Ainsi, la prochaine fois qu'il essaiera de transmettre, il tirera sa valeur de backoff sur une fenêtre de contention plus large. Globalement, l'algorithme du Binary Exponential Backoff permet de disperser les transmissions des appareils lorsque la contention est élevée. Si au contraire, l'appareil source reçoit un ACK, il réinitialise sa fenêtre de contention.

En plus du CSMA/CA, le protocole Wi-Fi met en place la possibilité de réserver le médium pour une certaine durée : le NAV (*Network Allocation Vector*). Elle est particulièrement utile dans le cas où toutes les STAs ne s'entendent pas entre elles. En effet, il arrive que des STAs soient situées suffisamment loin pour qu'elles ne puissent pas savoir quand l'autre est en train de transmettre à l'AP, et peuvent donc entrer en collision sans s'en apercevoir. Les appareils peuvent annoncer en amont qu'ils vont recevoir une transmission, et indiquent sa durée attendue dans le champ "duration" de l'en-tête MAC. Les autres appareils peuvent alors mettre à jour leur NAV, qui joue le rôle de chronomètre, et ne s'autorisent à émettre que lorsque celui-ci atteint zéro. Le NAV permet aux appareils d'annoncer en amont qu'il va recevoir une transmission, et que par conséquent les autres appareils autour de lui doivent se taire sous peine de collision.

Dans le cadre de l'économie d'énergie, certaines STA n'écoutent pas en permanence le médium radio, notamment lorsqu'elles hibernent. Ainsi, une STA sortant d'hibernation fait face à deux cas de figure. Soit elle détecte que le médium radio est occupé, auquel cas elle peut accéder au médium via le CSMA/CA sans autre problème. Soit elle détecte libre le médium radio. Alors elle ne peut pas savoir si le médium est réellement libre, ou si un autre appareil qu'elle n'entend pas a réservé le médium avec un NAV pendant son hibernation. Pour éviter les collisions, la STA doit alors attendre un *probe delay*, de l'ordre de plusieurs millisecondes, qui assure que si une transmission était en cours, elle est désormais terminée.

### 3.1.3 Les améliorations du protocole Wi-Fi

L'algorithme DCF atteint relativement bien ses objectifs, mais il introduit des surcoûts temporels non négligeables qui détériorent les performances globales des WLANs. Pour modérer ces détériorations, le protocole Wi-Fi a introduit un certain nombre de techniques. Nous présentons ici trois d'entre elles : l'agrégation de trames, les TXOPs, et l'OFDMA.

#### *L'agrégation de trames*

L'agrégation de trames consiste à envoyer plusieurs trames dans une seule transmission. Dans cette thèse, nous considérons une version simplifiée de l'agrégation de trames.

Chaque trame a normalement un en-tête PHY et un en-tête MAC. Ces en-têtes contiennent des informations utiles liées à la transmission (par exemple l'appareil destination). Lorsque plusieurs trames sont agrégées, elles n'ont plus qu'un en-tête PHY commun à toutes les trames (mais elles conservent leur en-tête MAC personnel). Après une transmission comportant plusieurs trames agrégées, l'appareil destination acquitte ses trames avec un ACK agrégé, où une bitmap indique quelles trames de la transmission ont été reçues avec succès. Nous considérons qu'au maximum, huit trames peuvent être agrégées ensemble.

Le gain apporté par l'agrégation de trames est double. D'un côté, on économise des en-têtes PHY, qui peuvent durer longtemps. De l'autre côté, on économise des accès au médium, puisque plusieurs trames sont envoyées dans une même transmission, qui n'a besoin que d'un accès au médium.

### Les TXOPs

En plus de l'agrégation de trames, le protocole Wi-Fi met en place des TXOPs (*Transmission Opportunities*). Les TXOPs donnent un accès privilégié au médium sous certaines conditions. Quand un appareil gagne l'accès au médium via l'algorithme DCF, il a le droit de le conserver pendant plusieurs transmissions consécutives (chaque transmission pouvant transporter plusieurs trames agrégées). Le seul impératif est que toutes les transmissions d'une même TXOP aient le même appareil destination, ce qui n'est réellement une limitation que pour les transmissions descendantes, du fait de l'architecture en étoile des WLANs. Le standard IEEE 802.11 ne fixe pas de valeur pour la durée maximale des TXOPs. Dans la suite, nous la considérons égale à 3 ms.

Pour conserver l'accès au médium entre deux transmissions consécutives d'une même TXOP, l'appareil attend seulement un SIFS (*Short Inter-Frame Space*) au lieu d'un AIFS avec backoff. Comme les SIFS sont plus courts que les AIFS, l'appareil évite la contention.

Voici un exemple pour donner un aperçu du surcoût protocolaire. Si une STA a un débit physique de 100 Mbps, elle peut atteindre un débit utile de l'ordre de 70 Mbps en utilisant l'agrégation de trames et les TXOPs, là où elle obtiendrait 30 Mbps sans ces améliorations.

### L'OFDMA

L'OFDMA (*Orthogonal Frequency-Division Multiple Access*) est une technique de transmission introduite récemment (en 2021, par l'amendement 802.11ax) qui permet d'envoyer plusieurs trames simultanément sur le médium, sous les conditions décrites ci-après. Lors d'une transmission OFDMA, le médium radio est découpé temporellement et fréquentiellement en blocs nommés RU (*Ressource Unit*). Chaque RU est allouée à une unique STA, mais deux RUs d'une même transmission OFDMA peuvent être allouées à des STAs différentes. Pour les transmissions OFDMA montantes, il existe également des RUs laissées libres, c'est-à-dire qu'elles ne sont allouées à aucune STA. Ces RUs sont dites en accès aléatoire, ou RA en anglais (*Random Access*). N'importe quelle STA peut essayer d'utiliser ces RUs en appliquant une technique proche de ALOHA.

Que la transmission OFDMA soit montante ou descendante, c'est à l'AP que revient la décision de l'allocation des RUs aux STAs. Pour les transmissions OFDMA descendantes, l'AP sait exactement combien de trames sont en attente pour chaque STA. Il peut donc allouer les RUs de manière optimale. Le cas des transmissions OFDMA montantes est plus compliqué car l'AP n'a pas de vision sur les buffers des STAs, et ne sait donc pas de combien de RUs chacune a besoin. Les STAs doivent donc lui signifier leur demande en RUs via des trames spécifiques, et l'AP peut allouer les RUs le plus justement possible avec ces informations.

Comme le protocole Wi-Fi impose la rétro-compatibilité à ses nouvelles fonctionnalités, l'OFDMA doit pouvoir cohabiter avec du CSMA/CA classique. Ainsi, l'AP

accède au médium avec du CSMA/CA puis initie une transmission OFDMA avec une *OFDMA trigger frame*. Cette trame annonce une transmission OFDMA imminente, et indique l'allocation des RUs aux STAs ainsi que les informations techniques pour que les STAs puissent les utiliser.

Parmi les mécanismes d'économie d'énergie que nous considérons dans notre étude, seuls deux reposent explicitement sur l'OFDMA : le *trigger-enabled TWT* (section 3.3.4) et le *Broadcast TWT* (section 3.3.4). Les autres mécanismes d'économie d'énergie que nous considérons reposent sur du CSMA/CA classique.

### 3.1.4 La consommation d'énergie dans le Wi-Fi

Le dernier aspect qui nous intéresse n'est pas lié à proprement parler au standard. Il s'agit des facteurs qui déterminent la consommation d'énergie d'une carte Wi-Fi. Là encore, c'est un problème trop vaste pour pouvoir en faire le tour ici. Nous nous sommes limités à trois facteurs, présentés ci-après, et dont nous détaillons les valeurs retenues pour notre étude en table 1.

Le premier facteur que nous considérons est l'état dans lequel se trouve la carte Wi-Fi, auquel nous associons une consommation d'énergie (Halperin et al., 2010). Nous considérons les états suivants :

- ❖ réception : la carte Wi-Fi est en train de recevoir une transmission,
- ❖ transmission : la carte Wi-Fi envoie une transmission,
- ❖ inaction (*idle* en anglais) : la carte Wi-Fi écoute le médium radio mais ne détecte aucune transmission,
- ❖ hibernation (*doze* en anglais) : la carte Wi-Fi est éteinte.

Le second facteur est le coût énergétique pour entrer ou sortir d'hibernation (Venkateswaran, Tai, Ben-Yehezkel et al., 2022). En effet, quand un appareil éteint sa carte Wi-Fi, elle continue de consommer pendant quelques millisecondes une faible quantité d'énergie. De même, lorsque la carte Wi-Fi est en train d'être rallumée, elle consomme un petit peu. Comme ce sont des coûts statiques, nous les avons converti en une quantité d'énergie que la carte Wi-Fi consomme chaque fois qu'elle entre ou sort d'hibernation. Ce facteur détermine également la durée minimale d'hibernation pour les appareils. Pour notre étude, nous avons utilisé la valeur de 5 ms pour la durée minimale d'hibernation, en accord avec l'étude de Venkateswaran, Tai, Ben-Yehezkel et al., 2022.

Le troisième facteur est le *crossfactor*, introduit par Serrano et al., 2014. Lorsqu'un appareil reçoit ou envoie une trame, il doit la traiter au niveau du processeur pour chaque couche de la pile réseau, comme TCP/IP. Nous avons choisi de ne pas prendre en compte le *crossfactor* dans notre étude. La première raison est que nous avons certains doutes sur la valeur avancée par l'article. En effet, cette étude a finalement été beaucoup citée, mais assez peu reproduite par la communauté. Deuxièmement, la consommation d'énergie induite par le *crossfactor* ne dépend que du volume de données envoyé et reçu au niveau applicatif, et qui doit donc traverser la pile réseau. Comme la couche applicative est hors-sujet dans cette thèse, nous avons préféré le retirer.

| Facteur de consommation d'énergie        | Consommation d'énergie |
|--|------------------------|
| État réception                           | 0.94 W                 |
| État transmission                        | 1.28 W                 |
| État inaction                            | 0.82 W                 |
| État hibernation                         | 0.1 W                  |
| Coût pour entrer et sortir d'hibernation | 242 $\mu$ J            |

TABLE 1 – Valeurs choisies pour notre étude de la consommation d'énergie associée à chaque facteur de consommation.

## 3.2 STRATÉGIES D'ÉCONOMIE D'ÉNERGIE

Les standards IEEE 802.11 ont introduit un certain nombre de mécanismes d'économie d'énergie. Dans notre article [Guérin, Begin, Guérin Lassous et al., 2023](#), nous les avons modélisés par des *Stratégies d'économie d'énergie*, qui représentent les grandes familles de mécanismes d'économie d'énergie en reprenant leur traits généraux tout en écartant les détails d'implémentation.

L'idée maîtresse des stratégies d'économie d'énergie est de restreindre certains trafics. Elles découpent le temps en deux types de période : les périodes de service, pendant lesquelles le trafic restreint est autorisé, et les périodes de restriction, pendant lesquelles le trafic restreint est interdit. Les stratégies d'économie d'énergie se différencient selon le type de trafic restreint (section 3.2.1), et la manière de le restreindre (section 3.2.2). Nous décrivons ci-après comment choisir quel type de trafic restreindre et nous décrivons les deux manières de restreindre un trafic.

### 3.2.1 Quel type de trafic restreindre ?

Une STA a deux types de trafic : le trafic descendant et le trafic montant. Une stratégie d'économie d'énergie peut donc restreindre soit le trafic descendant, soit le trafic montant, soit les deux.

Restreindre le trafic descendant bénéficie uniquement à la STA. En effet, en restreignant son trafic descendant, la STA impose à l'AP des périodes de restriction pendant lesquelles la STA est certaine de ne pas recevoir de trames. Elle peut donc entrer en hibernation (c'est-à-dire éteindre sa carte Wi-Fi) sans manquer une seule trame.

À l'opposé, restreindre le trafic montant bénéficie au WLAN dans sa totalité, lorsque les restrictions sont correctement appliquées. En effet, en restreignant le trafic montant de toutes les STAs et en leur assignant des périodes de service différentes, le WLAN peut grandement réduire le nombre de collisions. Dans ce cas, l'objectif est autant d'améliorer l'utilisation des ressources du médium que de réduire la consommation d'énergie du WLAN en évitant des retransmissions.

### 3.2.2 Comment restreindre un trafic ?

Nous considérons maintenant un trafic quelconque, allant d'un appareil source vers un appareil destination. Il existe deux méthodes pour le restreindre : la méthode "slot" et la méthode "prompt". Elles définissent les périodes de service (pendant lesquelles l'appareil source a le droit d'envoyer des trames à l'appareil destination) et les périodes de restriction (pendant lesquelles l'appareil source a interdiction d'envoyer des trames à l'appareil destination). Il est à noter que les autorisations et interdictions ne concernent que le trafic considéré : l'appareil destination a le droit d'envoyer des trames à l'appareil source pendant les périodes de restriction.

#### *La méthode "slot"*

La méthode "slot" est implicite : elle ne repose sur aucune signalisation (sauf lors de sa mise en place). L'appareil source (qui doit restreindre son trafic) et l'appareil destination se mettent d'accord au préalable sur une série de périodes de service, et donc sur une série de périodes de restriction par soustraction. Pour le dire simplement, l'appareil source et l'appareil destination posent une série de rendez-vous. Concrètement, la série de périodes de service est définie par la durée des périodes de service, notée  $D_{\text{slot}}$ , et l'intervalle entre la fin d'une période de service et le début de la suivante, notée  $I_{\text{slot}}$ . À noter que  $I_{\text{slot}}$  correspond à la durée des périodes de restriction. Pendant les périodes de service, l'appareil source peut envoyer normalement son trafic vers l'appareil destination sans règle supplémentaire. La figure 3 illustre la méthode "slot" avec ses paramètres.

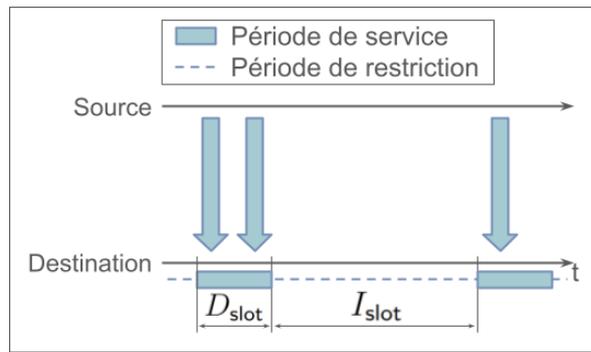


FIGURE 3 – Schéma d'une restriction de trafic utilisant la méthode "slot".

### La méthode "prompt"

La méthode "prompt" est explicite : elle repose entièrement sur la signalisation. L'appareil destination décide du début d'une période de service en envoyant à l'appareil source une trame spéciale – que nous baptisons prompt –, puis l'appareil source indique la fin de la période de service (après avoir éventuellement envoyé une ou plusieurs trames) en utilisant deux flags : le flag "EOSP" et le flag "More Data" (ici, un flag est un bit dans l'en-tête MAC des trames). La figure 4 schématise le fonctionnement de la méthode "prompt".

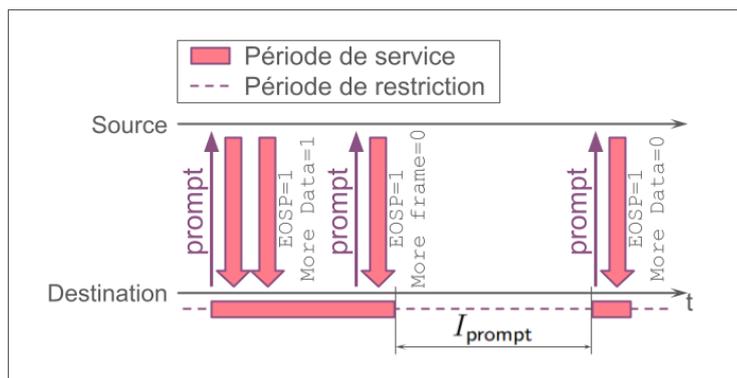


FIGURE 4 – Schéma d'une restriction de trafic utilisant la méthode "prompt".

En pratique, les flags EOSP et More Data servent à gérer les cas où l'appareil source a besoin de plusieurs TXOPs pour envoyer toutes les trames en attente à l'appareil destination. Lorsque l'appareil destination envoie un prompt, il déclenche une TXOP pour que l'appareil source lui envoie ses trames en attente. Le flag EOSP sert à indiquer que la fin de la TXOP, soit parce que l'appareil source n'a plus de trame en attente, soit parce que la TXOP a atteint sa durée maximale. Le flag More Data sert ensuite à faire la distinction entre ces deux cas : s'il est à 1, l'appareil source a encore des trames en attente pour l'appareil destination. Ainsi, lorsque l'appareil destination reçoit une trame dont le flag EOSP et le flag More Data sont à 1 (ce qui est le cas sur la première transmission de la figure 4), il doit accéder de nouveau au médium pour envoyer un autre prompt à l'appareil source. La table 2 détaille les combinaisons possibles des flags EOSP et More Data, et les actions à prendre les cas échéants.

Par rapport à la méthode "slot", la méthode "prompt" est plus souple et peut être dynamique, mais elle le paie avec le surcoût que représente la signalisation dont elle a besoin.

| Flags                     | Signification  | Action   |
|---------------------------|--|--|
| EOSP = 1<br>More Data = 1 | La TXOP de l'appareil source s'achève car elle a atteint sa durée maximale, mais l'appareil source a encore des trames en attente pour l'appareil destination. | L'appareil destination envoie de nouveau un prompt après avoir accédé au médium. |
| EOSP = 1<br>More Data = 0 | La TXOP de l'appareil source s'achève car il n'a plus de trames à envoyer à l'appareil destination.  | L'appareil destination entre dans une période de restriction.                    |
| EOSP = 0<br>More Data = 1 | La TXOP de l'appareil source n'est pas terminée et il a d'autres trames à envoyer à l'appareil destination.  | L'appareil destination attend la prochaine transmission de l'appareil source.    |
| EOSP = 0<br>More Data = 0 | Combinaison impossible   | N/A  |

TABLE 2 – Combinaisons possibles des flags EOSP et More Data, et les actions associées.

### 3.2.3 Nomenclature

Une stratégie d'économie d'énergie combine un type de trafic restreint avec la méthode pour le restreindre. Pour désigner une stratégie d'économie d'énergie, nous indiquons donc d'abord le type de trafic restreint (DL pour le trafic descendant, et UL pour le trafic montant), puis la méthode utilisée. Ainsi, la stratégie "DL prompt" restreint le trafic descendant avec la méthode "prompt". Certains mécanismes d'économie d'énergie restreignent à la fois le trafic descendant et le trafic montant. Pour représenter ces mécanismes, il faut alors combiner une stratégie qui restreint le trafic descendant avec une stratégie qui restreint le trafic montant. Par exemple, la stratégie "DL slot + UL prompt" restreint le trafic descendant avec la méthode "slot" et le trafic montant avec la méthode "prompt". L'AP doit donc envoyer des prompts à la STA pour que celle-ci puisse envoyer ses trames à l'AP, mais l'AP n'est pas autorisé à envoyer de trames en dehors des périodes de service du DL slot, ce qui comprend les prompts. Ainsi, tout le trafic de la STA est de fait regroupé autour des périodes de service du DL slot. La figure 5 illustre les différentes stratégies d'économie d'énergie à notre disposition.

Dans notre étude de performances, nous considérons principalement les stratégies qui ne restreignent qu'un seul type de trafic.

## 3.3 MÉCANISMES D'ÉCONOMIE D'ÉNERGIE

Le standard IEEE 802.11 définit deux modes de fonctionnement pour les STAs : le mode actif (*Active mode*) ou le mode économie d'énergie (*Power-Saving mode* ou *PS mode*). Une STA en mode actif n'a pas le droit d'hiberner car l'AP la considère prête à recevoir des trames descendantes à tout moment. Au contraire, une STA en mode économie d'énergie a le droit de régulièrement hiberner pour réduire sa consommation d'énergie. Cependant, pour passer en mode économie d'énergie, une STA doit annoncer à l'AP quel mécanisme d'économie d'énergie elle utilise. En effet, le mécanisme d'économie d'énergie régit quand la STA a le droit d'hiberner et quand l'AP a le droit d'envoyer des trames descendantes à la STA. En annonçant

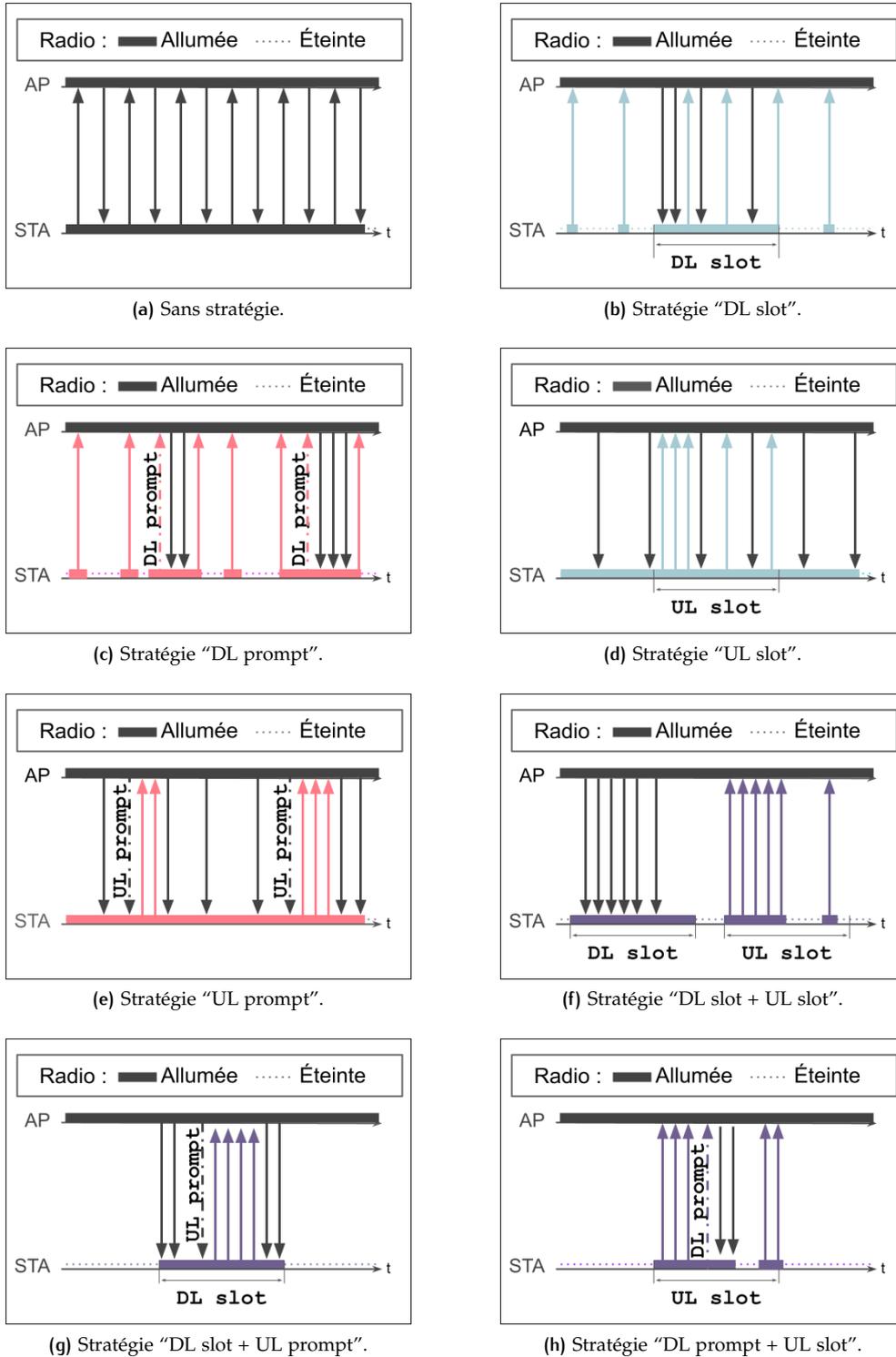


FIGURE 5 – Schéma des stratégies d'économie d'énergie.

son mécanisme d'économie d'énergie, la STA est sûre qu'elle ne manquera aucune de ses trames descendantes. À noter que l'AP n'a jamais le droit d'hiberner du fait de son rôle central.

Nous détaillons dans les sections qui suivent les mécanismes d'économie d'énergie que nous avons identifiés. Il est à noter que nous avons réutilisé certains termes des mécanismes d'économie d'énergie dans nos définitions des stratégies d'économie d'énergie, mais ces termes ne représentent pas rigoureusement la même chose pour les stratégies d'économie d'énergie et pour les mécanismes d'économie d'énergie.

### 3.3.1 Standard "Legacy" (1999)

#### PS-Poll

Le mécanisme PS-Poll (*Power-Save Poll*) suit la stratégie "DL prompt", il est décrit en section 11.2.1.1 de [IEEE WG 802.11, 1999](#). Lorsque la STA souhaite récupérer ses éventuelles trames descendantes en attente dans le buffer de l'AP, elle envoie une trame dédiée, nommée PS-Poll. Lorsque l'AP reçoit un PS-Poll, soit il y a effectivement des trames descendantes en attente pour la STA, soit il n'y en a pas. Dans le premier cas, l'AP envoie une unique trame descendante en retour au PS-Poll (pour laquelle la STA renverra ensuite un ACK). Si l'AP a encore d'autres trames descendantes en attente pour cette STA, il le fera savoir en mettant le flag More Data de l'en-tête MAC à 1. Dans le second cas, l'AP retourne un simple ACK au PS-Poll. À noter que le mécanisme PS-Poll a été introduit dans le standard IEEE 802.11 avant l'agrégation de trames et avant les TXOPs, c'est la raison pour laquelle les STAs doivent aller chercher les trames en attente une par une, chaque fois avec un PS-Poll.

Les STAs sont relativement libres de décider quand envoyer un PS-Poll. La seule contrainte est le *Listen Interval*, dont la valeur est choisie par la STA : elle doit envoyer au moins un PS-Poll par Listen Interval. Typiquement, un Listen Interval dure quelques centaines de millisecondes. Cette restriction vient du fait que l'AP est obligé de garder en mémoire les trames descendantes en attendant que la STA envoie un PS-Poll. Pour éviter d'encombrer la mémoire de l'AP, la STA doit donc régulièrement récupérer ses trames descendantes en attente.

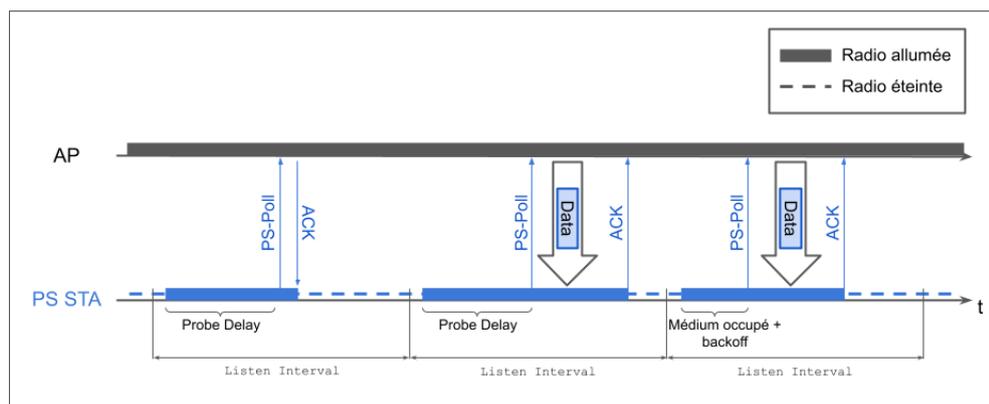


FIGURE 6 – Exemple d'échange de trames avec le mécanisme PS-Poll.

La figure 6 présente un exemple d'échanges de trames lorsque le mécanisme PS-Poll est utilisé. Lors du premier échange, le médium est libre lorsque la STA se réveille. Elle peut donc envoyer un PS-Poll après avoir attendu le "probe delay" (cf. section 3.1.2). Cependant, l'AP n'a pas de trame en attente pour la STA, il lui renvoie donc un ACK. La STA peut alors retourner en hibernation. Lors du deuxième échange, le médium est toujours libre et la STA envoie un PS-Poll après le "Probe delay". Cette fois-ci, l'AP a une trame en attente pour la STA, il lui envoie donc.

Comme il n'en a qu'une, il met à 0 le flag More Data. La STA retourne un ACK après avoir reçu la trame, puis elle peut retourner en hibernation puisqu'elle n'a plus de trame en attente. Lors du troisième échange, le médium est occupé quand la STA se réveille. Elle attend donc que le médium se libère, puis envoie un PS-Poll après avoir attendu un backoff. Le reste se déroule comme lors du deuxième échange. Dans la suite, nous ne représentons plus le "probe delay" sur les figures.

### TIM

Le mécanisme TIM (*Traffic Indication Map*) a été introduit pour servir de complément au mécanisme PS-Poll, il est décrit en section 11.2.1.3 de [IEEE WG 802.11, 1999](#). Il ne restreint aucun trafic par lui-même, et ne correspond donc à aucune stratégie d'économie d'énergie. La combinaison de TIM et de PS-Poll est souvent appelée *Legacy PS (Power-Save) mode*.

Une TIM est une bitmap de taille égale au nombre de STAs associées que l'AP insère dans ses beacons. Chaque bit est associé à une STA, et sa valeur indique si l'AP a des trames en attente. Une STA peut donc se réveiller pour écouter le beacon, et si le bit lui correspondant est à 0, elle peut retourner en hibernation immédiatement et ainsi, éviter d'envoyer un PS-Poll inutile. Si au contraire, son bit est à 1, cela signifie que l'AP a des trames pour elle, et qu'elle doit donc envoyer un PS-Poll.

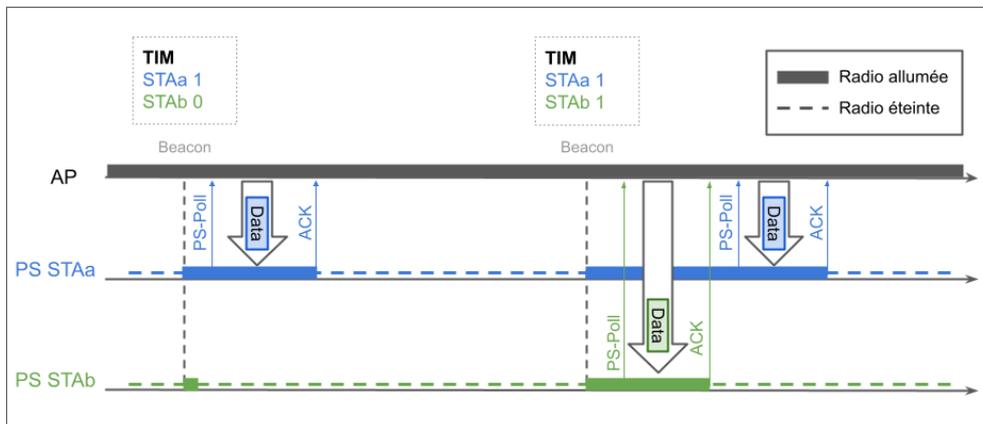


FIGURE 7 – Exemple d'utilisation des bitmaps TIM.

La figure 7 montre un exemple d'utilisation de TIM avec PS-Poll, en considérant un Listen Interval égal à un intervalle inter-beacon. Les STAs a et b se réveillent pour écouter le premier beacon de l'AP. La bitmap TIM de ce beacon indique que seule la STAa a des trames descendantes en attente dans le buffer de l'AP. La STAb retourne donc en hibernation pendant que la STAa accède au médium via le CSMA/CA et envoie un PS-Poll et récupère ses trames en attente. Les STAs se réveillent de nouveau pour écouter le deuxième beacon de l'AP. Cette fois-ci, elles ont toutes deux des trames descendantes en attente. Elles rentrent donc en contention pour accéder au médium après le beacon. La STAb remporte la contention et envoie en premier son PS-Poll, ce à quoi l'AP répond avec sa trame en attente. Puis la STAa peut envoyer à son tour un PS-Poll et récupérer sa trame en attente.

### 3.3.2 Amendement 802.11e (2005)

L'amendement 802.11e a introduit la notion de QoS (*Quality of Service*) dans le standard. Les mécanismes d'économie d'énergie introduits dans ce même amendement sont donc conçus pour fonctionner avec les catégories d'accès définies pour la QoS : chaque catégorie d'accès peut avoir son propre mécanisme d'économie d'énergie, indépendamment des autres catégories d'accès. Pour simplifier notre ex-

plication, nous considérons ci-après que les mécanismes sont appliqués à toutes les catégories d'accès, sans distinction.

### U-APSD

U-APSD (*Unscheduled Automatic Power Save Delivery*) est un mécanisme utilisant la stratégie "DL prompt", il est décrit en section 11.2.1.4 de [IEEE WG 802.11e, 2005](#). Par rapport au mécanisme PS-Poll, qui utilise aussi la stratégie "DL prompt", le mécanisme U-APSD apporte deux améliorations. Premièrement, le mécanisme U-APSD utilise une alternative à la trame PS-Poll, qui se nomme *APSD trigger frame*. La trame PS-Poll est une trame de management qui a été conçue dans l'unique but de signaler à l'AP le début d'une période de service. Avec le mécanisme U-APSD, l'AP considère n'importe quelle trame montante de la STA comme *APSD trigger frame*. Si la STA n'a pas de trame montante à envoyer mais souhaite tout de même récupérer ses trames descendantes en attente, elle peut envoyer un NDP (*Null Data Packet*). Ainsi, l'APSD trigger frame désigne plus un évènement qu'une trame en particulier.

Deuxièmement, le mécanisme U-APSD utilise les TXOPs : quand l'AP reçoit une *APSD trigger frame*, il accède à une TXOP pour envoyer les trames descendantes en attente. Ainsi, la STA n'a pas à envoyer une *APSD trigger frame* pour chaque trame descendante en attente, comme c'est le cas avec le mécanisme PS-Poll. L'AP désigne la fin de la période de service avec le flag "EOSP".

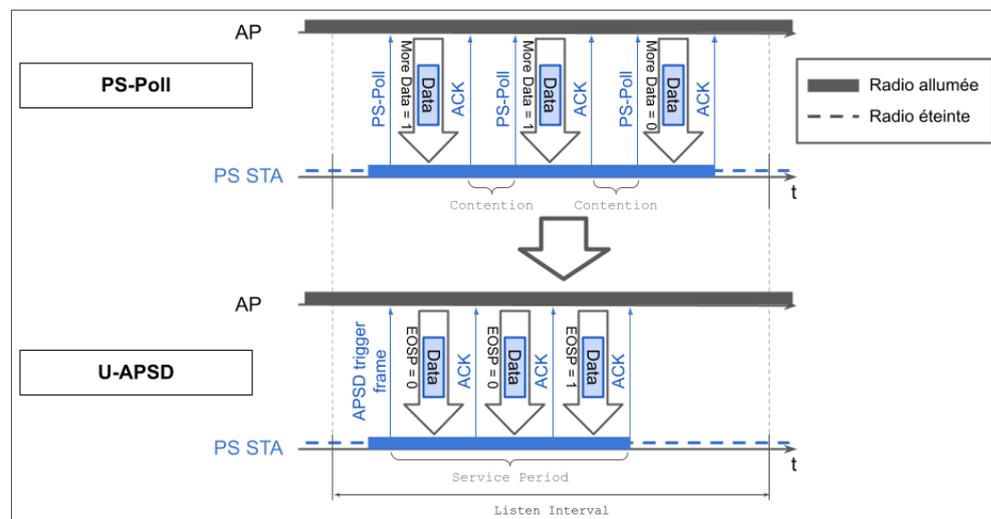


FIGURE 8 – Comparaison entre le mécanisme U-APSD et le mécanisme PS-Poll.

La figure 8 illustre le deuxième avantage du mécanisme U-APSD face au PS-Poll. Avec le mécanisme PS-Poll, la STA doit envoyer trois PS-Poll, chaque fois en gagnant l'accès au médium face aux autres appareils potentiellement en compétition. Avec le mécanisme U-APSD, la STA n'a besoin d'accéder qu'une seule fois au médium pour récupérer toutes ses trames descendantes en attente.

Par la suite, nous utiliserons la formulation "envoyer un prompt" pour indiquer qu'une STA envoie soit une trame PS-Poll, soit une *APSD trigger frame* pour récupérer ses trames descendantes en attente.

### S-APSD

Le mécanisme S-APSD (*Scheduled Automatic Power Save Delivery*) utilise la stratégie "DL slot", décrit en section 11.2.1.4 de [IEEE WG 802.11e, 2005](#). Avec le mécanisme S-APSD, les périodes de service commencent à l'initiative de l'AP. Il transmet à la STA l'intervalle de temps entre les début de deux périodes de service consécutives, ainsi que le début de la première période de service. Ensuite, l'AP peut

ponctuellement changer le début des périodes de service avec une trame dédiée, appelée "Schedule frame". La fin des périodes de service n'est pas statique comme c'est le cas pour la stratégie "DL slot". Elle est entièrement déterminée par l'AP, qui l'indique à la STA avec le flag EOSP de ses trames.

Les paramètres de S-APSD sont négociés entre la STA et l'AP : la STA indique à l'AP l'éventail des valeurs qui lui conviennent, et l'AP choisit dans cet éventail la valeur finale.

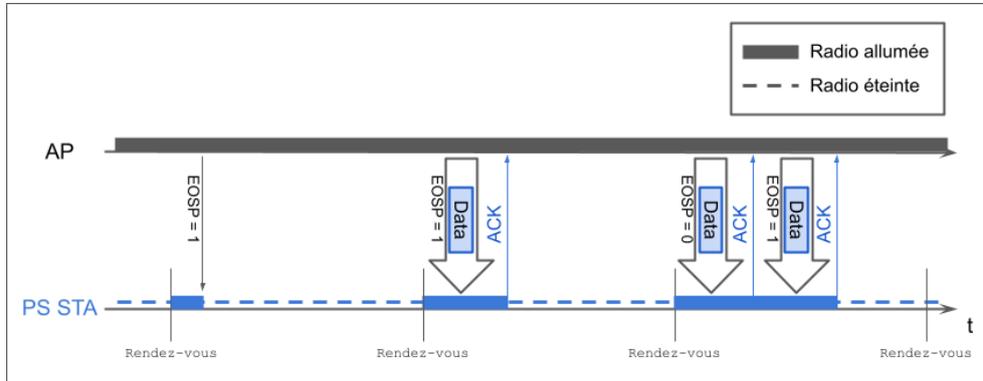


FIGURE 9 – Exemple d'échange de trames avec le mécanisme S-APSD en mode implicite.

La figure 9 représente le processus du mécanisme S-APSD. La STA se réveille à intervalle régulier. Dans le premier cas, l'AP n'a pas de trame en attente pour la STA, et envoie donc une trame vide avec le flag "EOSP" levé pour que la STA puisse se rendormir rapidement. Dans le deuxième cas, l'AP envoie une transmission avec le flag "EOSP" levé, la STA retourne donc en hibernation après avoir reçu la transmission. Enfin, dans le troisième cas, l'AP envoie une première transmission avec le flag "EOSP" baissé. La STA retourne donc un ACK et attend la prochaine transmission. L'AP renvoie une deuxième transmission avec le flag "EOSP" cette fois-ci levé, ce qui permet à la STA de retourner en hibernation après la transmission.

### 3.3.3 Amendement 802.11ah "HaLow" (2016)

L'amendement 802.11ah occupe une place particulière dans le standard. En effet, il opère sur une bande de fréquence différente de celle du Wi-Fi "classique", et est à destination des appareils dits IoT (*Internet of Things*). Ce terme regroupe toutes sortes d'appareils qui ont un trafic léger, et qui ont besoin de conserver au maximum leur énergie. Un exemple-type d'appareil IoT est un capteur sur batterie qui doit régulièrement envoyer la valeur qu'il a mesurée. Comme l'économie d'énergie est une contrainte majeure pour les appareils IoT, l'amendement 802.11ah y a apporté une attention particulière.

#### La segmentation TIM

La segmentation TIM étend le mécanisme TIM introduit par le standard "Legacy", en autorisant la bitmap TIM à être découpée en plusieurs morceaux qui sont diffusés séparément. L'objectif est de pouvoir gérer un plus grand nombre de STAs sans que la bitmap TIM devienne trop grande (et ne prenne trop de temps à être diffusée dans les beacons). Ce mécanisme est décrit en section 10.47 de [IEEE WG 802.11ah, 2016](#)

Pour découper la bitmap TIM, l'amendement 802.11ah commence par introduire la notion de *beacon court* : ce sont des beacons qui ne transportent qu'une partie des informations d'un beacon traditionnel. Un AP de l'amendement 802.11ah est chargé de diffuser à la fois les beacons traditionnels et les beacons courts. L'intervalle inter-beacon traditionnel est divisé en intervalles de temps égaux, et à chaque

sous division, l'AP diffuse un beacon court. Autrement dit, l'AP diffuse un certain nombre de beacons courts entre chaque beacon traditionnel.

En parallèle, les STAs sont divisées en groupes. Chaque short beacon est associé à un de ces groupes de STAs, et transporte la bitmap TIM qui le concerne. Entre chaque beacon traditionnel, l'AP diffuse un beacon court pour chaque groupe de STAs. Quant aux beacons traditionnels, ils transportent un autre type de bitmap TIM appelée DTIM (*Delivery Traffic Indication Map*). Ses bits ne correspondent plus à des STAs en particulier, mais aux groupes de STAs définis précédemment. S'il est à 1, alors au moins une STA du groupe a une ou plusieurs trames descendantes en attente. Si au contraire, il est à 0, alors aucune STA du groupe n'a de trame descendante en attente.

Pour diviser les STAs en groupe, l'AP utilise le mécanisme du *Page slicing*. Les groupes contiennent des STAs dont les AIDs (*Association ID*) sont consécutifs. Les groupes sont annoncés dans les beacons traditionnels, et sont valables pour l'intervalle inter-beacon subséquent, ce qui permet de faire varier les groupes d'un intervalle inter-beacon à un autre. L'AP peut également changer les AIDs des STAs en les réassociant, si besoin. Cela lui permet de regrouper certaines STAs plus facilement.

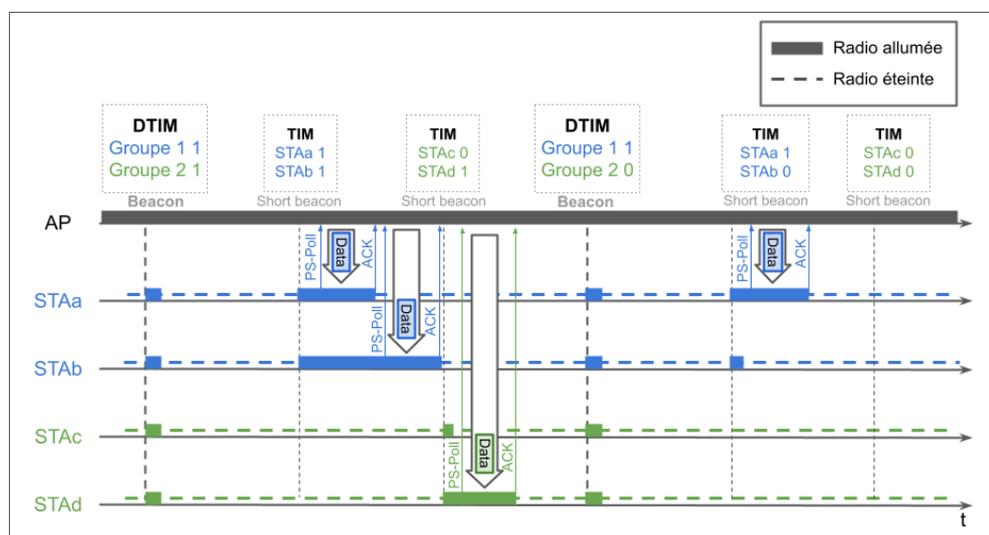


FIGURE 10 – Exemple de bitmaps TIM segmentées.

La figure 10 représente la segmentation TIM. Dans cet exemple, on considère deux groupes de STAs : les STAs a et b appartiennent au groupe 1, et les STAs c et d au groupe 2. Comme il y a deux groupes, il y a aussi deux beacons courts entre les beacons traditionnels. Le premier beacon traditionnel annonce dans sa bitmap DTIM que les deux groupes ont des trames descendantes en attente : les STAs du groupe 1 vont donc se réveiller pour le premier beacon court, et les STAs du groupe 2 pour le second beacon court. Le premier beacon court transporte la bitmap TIM du groupe 1, et annonce que les deux STAs ont des trames descendantes en attente. Les STAs a et b vont donc entrer en contention pour envoyer un PS-Poll à l'AP. Dans le second beacon court, la bitmap TIM, qui concerne le groupe 2, indique que seule la STA d a des trames descendantes en attente. La STA c retourne donc immédiatement en hibernation pendant que la STA d envoie un PS-Poll à l'AP. Dans le second beacon traditionnel, la bitmap DTIM indique que seul le groupe 1 a des trames descendantes en attente, pas le groupe 2. Les STAs du groupe 2 ne se réveillent donc pas pour leur beacon court, puisqu'elles savent qu'aucune d'entre elles n'a de trame descendante en attente.

## RAW

En complément de la segmentation TIM, l'amendement 802.11ah propose le mécanisme RAW (*Restricted Access Window*) qui utilise la stratégie "UL slot", et décrit en section 10.22.5 de [IEEE WG 802.11ah, 2016](#).

RAW impose aux STAs des périodes de silence pour réduire la contention dans le WLAN. En pratique, l'AP définit des périodes de temps, appelées *créneaux RAW*, pendant lesquelles un seul groupe de STAs est autorisé à faire des transmissions ; les STAs qui n'appartiennent pas au dit groupe ont interdiction de transmettre quoi que ce soit. L'AP peut également définir d'autres périodes de temps où toutes les STAs, quel que soit le groupe auquel elles appartiennent, peuvent émettre en utilisant le CSMA/CA classique. D'une certaine manière, RAW peut être vu comme du TDMA (*Time Division Multiple Access*) superposé à du CSMA/CA.

RAW repose sur le page slicing (cf. section 3.3.3) pour définir les groupes de STAs. Cela signifie que s'il est utilisé avec la segmentation TIM, les deux mécanismes utiliseront les mêmes groupes de STAs. Ensuite, RAW divise de nouveau les STAs des groupes en sous-groupes. L'AP annonce dans les beacons traditionnels les groupes de STAs, puis le nombre de sous-groupes pour chaque groupe. Les STAs d'un groupe sont ensuite distribuées dans les sous-groupes en round robin. Les intervalles entre les beacons courts sont ensuite divisés également par le nombre de sous-groupes.

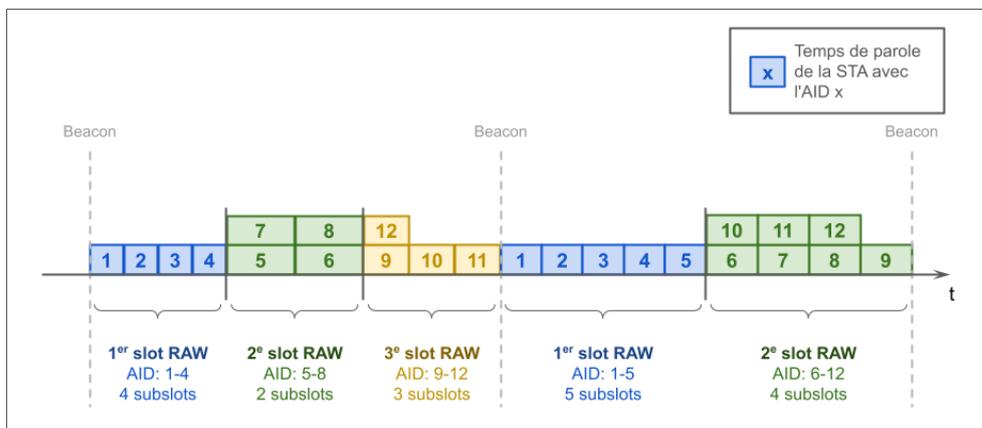


FIGURE 11 – Exemple d'une division de STAs en utilisant RAW.

La figure 11 montre une manière de diviser 12 STAs en groupes et sous-groupes avec RAW. Pendant le premier intervalle inter-beacon, l'AP divise les STAs en trois groupes, chaque groupe ayant quatre STAs. Le premier groupe est ensuite divisé en quatre sous-groupes, chaque STA a donc un créneau RAW dédié. Le second groupe est divisé en deux sous-groupes, il y a donc deux STAs par créneau RAW. Le troisième groupe est divisé en trois sous-groupes, les STAs avec l'AID 9 et 12 doivent donc partager un créneau RAW, tandis que les STAs avec l'AID 10 et 11 ont leur créneau RAW dédié. Pendant le second intervalle inter-beacon, l'AP change les divisions : il y a seulement deux groupes, avec respectivement cinq et quatre sous-groupes.

Un des paramètres de RAW est le CSB (*Cross Slot Boundary*). Il autorise ou non les STAs à déborder dans le créneau RAW qui suit le leur. Si le champ CSB, contenu dans le beacon, est à 0, les STAs ne doivent pas commencer une transmission si elle ne se terminera pas avant la fin de leur créneau RAW. Avec cette configuration, les STAs ne sont pas sujettes au probe delay (cf. section 3.1.2) pendant leur créneau RAW, elles peuvent accéder au médium avec du CSMA/CA classique dès le début de leur créneau RAW. En effet, elles ont la garantie que les autres STAs sont toutes silencieuses. Sinon, lorsque le champ CSB est à 1, les STAs qui se réveillent pour leur créneau RAW doivent respecter le probe delay avant d'essayer d'émettre.

Comme le niveau de contention est différent à l'intérieur et à l'extérieur des créneaux RAW, l'amendement 802.11ah introduit un nouveau backoff spécialement pour les créneaux RAW : le *backoff RAW*. Une STA utilise le backoff RAW à l'intérieur de ses créneaux RAW, et le backoff traditionnel pendant les périodes sans restriction. Le backoff RAW et sa fenêtre de contention associée sont réinitialisés au début de chaque créneau RAW. Quant au backoff régulier, il est suspendu pendant les créneaux RAW (que la STA soit concernée ou non). Ainsi, le temps des créneaux RAW n'est pas perdu à cause d'un backoff particulièrement long.

À noter que l'amendement 802.11ah prévoit la combinaison de la segmentation TIM avec RAW, et avec PS-Poll. Ainsi, les STAs savent efficacement si elles ont des trames descendantes en attente, et peuvent envoyer un PS-Poll pour les récupérer en subissant le moins de contention possible grâce à RAW. Cette combinaison utilise la stratégie "DL prompt + UL slot", avec information sur les trames descendantes en attente.

### *TWT individuel*

Le mécanisme TWT (*Target Wake Time*) individuel utilise la stratégie "DL slot + UL slot" si la STA est en mode économie d'énergie, ou seulement "UL slot" si la STA est en mode actif. L'idée du mécanisme TWT individuel est de réduire la contention tout en permettant aux STAs d'économiser de l'énergie si elles le souhaitent. Il est décrit en section 10.43 de [IEEE WG 802.11ah, 2016](#), puis a été ensuite détaillé en sections 26.8.2 et 26.8.3 de [IEEE WG 802.11ax, 2021](#)

Avec le mécanisme TWT individuel, le début des périodes de service peut être définie selon deux modes : le mode implicite et le mode explicite. En mode implicite, l'intervalle de temps entre les débuts de deux périodes de service consécutives est défini à l'avance, après négociation entre l'AP et la STA. En mode explicite, l'AP doit, à chaque période de service, informer la STA du début de la prochaine période de service. Les périodes de service se terminent par l'un des deux événements suivants, selon celui qui arrive en premier : (i) l'envoi d'une trame avec le flag EOSP à 1 de l'AP vers la STA, ou (ii) l'écoulement de la durée définie par le paramètre "MinimumTWTWakeDurationTime" depuis le début de la période de service.

Le mécanisme TWT individuel innove sur plusieurs aspects, notamment en prenant en compte les besoins de la STA. En premier lieu, l'AP et la STA peuvent négocier les paramètres du TWT individuel selon leurs objectifs respectifs. Par exemple, une STA va plutôt chercher à optimiser son économie d'énergie tandis que l'AP va chercher à éviter les collisions et à distribuer le temps équitablement. La négociation se déroule avec un ensemble de trames spécialement conçues pour. Une STA peut avoir jusqu'à huit TWT individuels en parallèle avec l'AP, chacun indépendant des autres et avec sa propre configuration. L'AP comme la STA peuvent rompre unilatéralement un TWT individuel à tout moment, s'il ne leur convient plus.

Le mécanisme TWT individuel ajoute un paramètre appelé *flow type*. Si ce paramètre est à 0, l'AP ne peut pas envoyer de trames descendantes pendant la période de service tant que la STA ne lui a rien envoyé d'abord. Cela permet à la STA de ne pas se réveiller à toutes les périodes de service. Si au contraire, sa valeur est 1, l'AP peut envoyer des trames descendantes dès le début des périodes de service.

La figure 12 montre les différentes façons d'utiliser le mécanisme TWT individuel. Nous considérons trois STAs qui ont négocié au préalable un TWT individuel. La fréquence des périodes de service est différente pour chaque STA. La STA<sub>a</sub> est en mode économie d'énergie et a négocié un TWT individuel avec le paramètre "flow type" à 1. La STA<sub>b</sub> est aussi en mode économie d'énergie, mais son paramètre "flow type" est à 0. Enfin, la STA<sub>c</sub> est en mode actif, et n'a donc pas besoin du paramètre "flow type" puisque son trafic descendant n'est pas restreint.

Avec son paramètre "flow type" à 1, la STA<sub>a</sub> est obligée de se réveiller à chacune de ses périodes de service, au cas où l'AP aurait des trames descendantes en attente pour elle. Elle rentre en hibernation entre ses périodes de service. Lors de

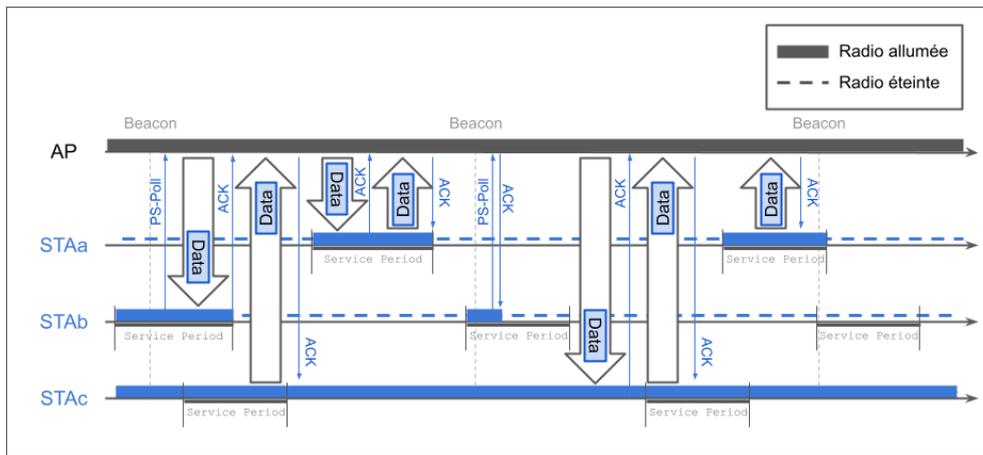


FIGURE 12 – Exemple d'échanges de trames lorsque des STAs souscrivent à un TWT individuel.

sa première période de service, l'AP a une trame descendante pour la STA en attente, il l'envoie dès le début de la période de service. La STAA peut ensuite envoyer ses trames montantes en attente. Lors de sa deuxième période de service, l'AP n'a aucune trame descendante à envoyer. La STAA attend donc le probe delay avant d'envoyer ses trames montantes en attente.

La STAB a le paramètre "flow type" à 0, elle a donc le droit de sauter certaines de ses périodes de service, mais elle doit envoyer un prompt si elle veut récupérer ses trames descendantes en attente. C'est ce qu'elle fait lors de sa première et sa deuxième périodes de service. Lors de sa première période de service, l'AP répond avec une trame descendante. Lors de sa seconde période de service, il répond avec un ACK. Enfin, pour sa troisième période de service, la STAB décide de ne pas se réveiller, et n'envoie donc pas de prompt à l'AP.

Comme la STAC est en mode actif, l'AP lui envoie ses trames descendantes quand il veut. En revanche, la STAC doit attendre ses périodes de service pour envoyer ses trames montantes.

### 3.3.4 Amendement 802.11ax (2021)

L'amendement 802.11ax a introduit l'OFDMA dans le standard. Par conséquent, il a aussi adapté le mécanisme TWT pour qu'il puisse tirer parti au mieux de l'OFDMA. Comme expliqué en section 3.1.3, l'OFDMA permet d'avoir plusieurs transmissions en parallèle. Dans le cas présent, le but est d'éviter que les STAs occupent tout le canal pour envoyer de petites trames, comme par exemple des PS-Polls. Cela a donné naissance à deux nouvelles formes de TWT : le *trigger-enabled TWT* et le *broadcast TWT*.

#### *Trigger-enabled TWT*

Le mécanisme Trigger-enabled TWT suit globalement la stratégie "DL prompt + UL slot", et est décrit en section 26.8.2 de [IEEE WG 802.11ax, 2021](#).

Le mécanisme trigger-enabled TWT repose sur des périodes de service définies au préalable, comme pour les TWTs individuels. Durant les périodes de service d'une STA, l'AP va initier au moins une transmission OFDMA montante dont certaines RUs sont réservées pour la STA. Elle peut les utiliser soit pour envoyer des trames de données montantes, soit pour envoyer un prompt à l'AP. Les STAs utilisant le mécanisme trigger-enabled TWT n'ont plus le droit d'envoyer des trames en dehors des transmissions OFDMA. En retour, l'AP est obligé de leur fournir des RUs pendant chacune de leur période de service. Pour l'AP, l'idée est donc de regrouper

plusieurs STAs sur une même période de service, et de prévoir une transmission OFDMA montante avec des RUs pour chacune de ces STAs.

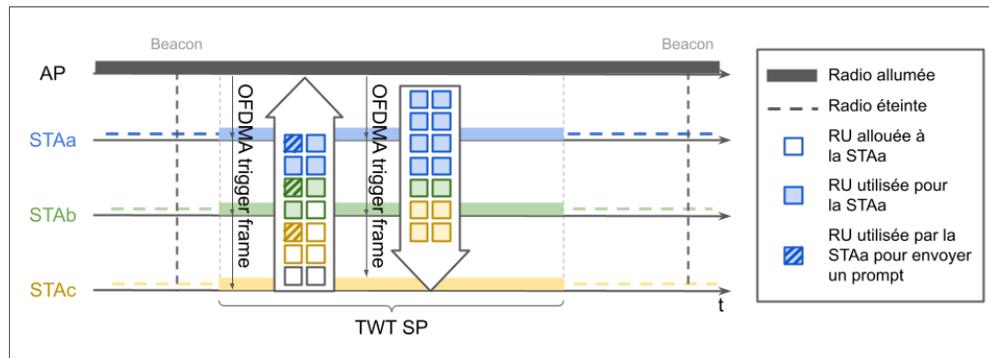


FIGURE 13 – Exemple d'échange de trames durant une SP de trigger-enabled TWT. Les couleurs de la légende s'adaptent en fonction de la STA.

La figure 13 illustre comment mutualiser trois STAs qui utilisent le mécanisme trigger-enabled TWT. Les trois STAs (STAa, STAb et STAc) ont accepté un trigger-enabled TWT, et leurs périodes de service se superposent. Elles ont respectivement 3, 2 et 0 trames montantes à envoyer, et 8, 2 et 4 trames descendantes à recevoir. Pendant leur période de service commune, l'AP initie une transmission OFDMA montante avec quatre RUs allouées à chaque STA. Les STAs utilisent cette transmission OFDMA montante pour envoyer leurs éventuelles trames montantes en attente, et pour envoyer un prompt à l'AP. À noter que la STAb et la STAc n'utilisent pas toutes leurs RUs car elles n'en ont pas besoin. L'AP répond avec une transmission OFDMA descendante, qui contient les trames descendantes en attente pour la STAa, la STAb et la STAc. Comme l'AP sait combien de trames il doit envoyer à chaque STA, il peut cette fois allouer exactement le nombre de RUs nécessaire à chaque STA.

### Broadcast TWT

Le mécanisme broadcast TWT suit globalement la stratégie "DL prompt + UL slot", et est décrit en section 26.8.3 de [IEEE WG 802.11ax, 2021](#).

Le mécanisme broadcast TWT utilise l'OFDMA, comme le mécanisme trigger-enabled TWT. Cependant, au lieu d'allouer des RUs aux STAs en mode économie d'énergie, il utilise l'accès aléatoire.

L'AP annonce les périodes de service du broadcast TWT dans ses beacons, dans un champ appelé *TWT element*. En particulier, il indique quand ces périodes de service commencent, et combien de temps au minimum elles dureront. Pendant chaque période de service, l'AP doit prévoir au moins une transmission OFDMA avec des RUs en accès aléatoire. Pour les STAs qui utilisent le broadcast TWT, elles doivent se réveiller d'abord lors des beacons pour savoir quand ont lieu les périodes de service. Puis elles doivent se réveiller pendant l'une de ces périodes de service, et envoyer un prompt en utilisant une RU en accès aléatoire. Elles sont en effet obligées d'envoyer un prompt pour recevoir leurs trames descendantes en attente, car elles ne sont pas tenues de se réveiller à chaque période de service (et donc l'AP ne sait pas exactement quand elles sont éveillées).

La figure 14 montre un exemple de période de service avec le mécanisme broadcast TWT. Nous considérons quatre STAs qui ont toutes accepté d'utiliser le mécanisme broadcast TWT. Les STAs a, b et d se réveillent à chacune des périodes de service, tandis que la STAc en saute de temps en temps. Durant la première période de service, les quatre STAs se réveillent. Aucune d'entre elles n'a de trame montante à envoyer, mais elles essaient toutes d'envoyer un prompt à l'AP. La STAc et la STAd parviennent à envoyer leur prompt, mais pas les STAs a et b, car elles ont essayé

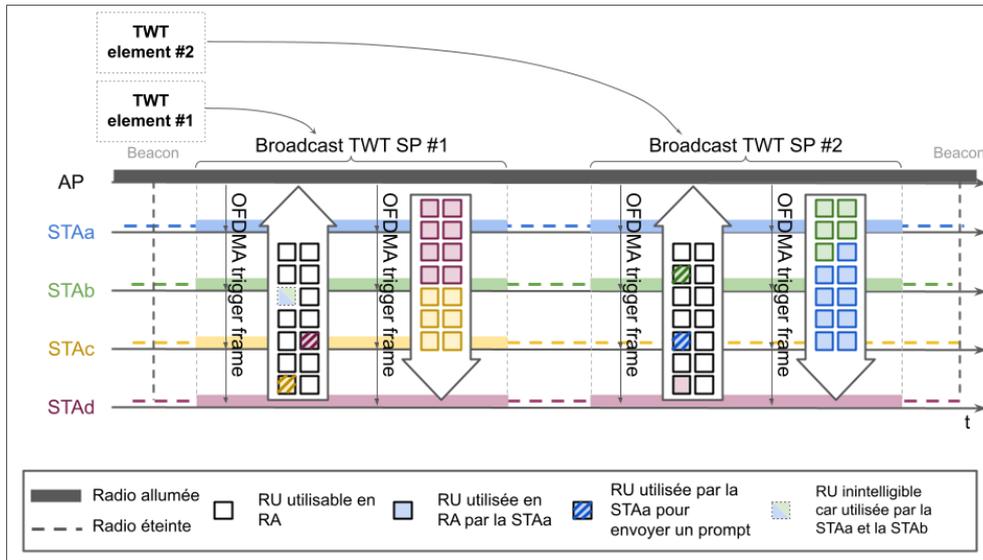


FIGURE 14 – Exemple de transmissions OFDMA durant une SP de broadcast TWT. RA signifie ici “Random Access”.

d’accéder à la même RU. L’AP répond alors avec une transmission OFDMA descendante avec des trames uniquement pour la STAc et la STAd. Durant la deuxième période de service, seules les STAs a, b et d se réveillent. Les STAs a et b réessaient d’envoyer un prompt à l’AP, tandis que la STA d souhaite envoyer une trame montante. Comme chaque STA choisit une RU différente de celle des autres, aucune collision n’arrive pendant la transmission. L’AP répond donc avec une transmission OFDMA descendante qui contient les trames pour les STAs a et b.

### OPS

Le mécanisme OPS (*Opportunistic Power Save*) est en fait une combinaison entre le mécanisme broadcast TWT et le mécanisme TIM.

Au début de chaque période de service du mécanisme broadcast TWT, l’AP envoie une bitmap TIM qui autorise ou interdit les STAs à hiberner avant la fin de la période de service. Dans les faits, la bitmap TIM n’indique alors plus la présence ou non de trame descendante en attente dans le buffer de l’AP, mais plutôt les intentions de celui-ci. Si l’AP prévoit d’allouer des RUs dans une transmission OFDMA montante à une STA, ou s’il a des trames en attente pour elle, il peut ainsi l’empêcher d’hiberner. Dans le premier cas, cela évite à l’AP de gâcher des RUs d’une transmission OFDMA montante. Dans le deuxième cas, ça évite que la STA manque une partie de ses trames descendantes.

L’avantage du mécanisme OPS par rapport au broadcast TWT est l’absence de prompt. Comme l’AP est certain que les STAs qu’il a désignés dans sa bitmap TIM ne retourneront pas en hibernation avant la fin de la période de service, il peut envoyer ses trames descendantes sans avoir reçu de prompt auparavant.



# 4

## MÉTHODES D'ÉVALUATION

Nous souhaitons pouvoir configurer les stratégies d'économie d'énergie et les comparer entre elles. Pour ce faire, nous avons besoin d'évaluer leurs performances dans différentes situations, avec différentes configurations. Les questions qui s'en suivent sont alors les suivantes : quelles métriques considérer et comment les quantifier ?

Comme les mécanismes d'économie d'énergie reposent sur la restriction d'au moins un trafic, ils posent un compromis : d'un côté, ils risquent de dégrader les performances réseaux, de l'autre, ils peuvent réduire la consommation d'énergie des appareils. Il est donc nécessaire de prendre en compte les deux faces du compromis dans l'étude des mécanismes d'économie d'énergie.

Nous considérons trois métriques de performance pour évaluer les stratégies d'économie d'énergie. Les deux premières sont des performances réseaux : il s'agit du taux de perte moyen et du délai moyen sur les trames. Ces métriques sont calculées séparément pour chaque trafic. Le taux de perte moyen est défini comme le rapport entre le nombre de paquets qui ont été perdus et le nombre de paquets qui sont arrivés à l'entrée du buffer de l'appareil source. Le délai d'une trame est calculé comme la différence de temps entre le moment où la trame arrive dans le buffer de l'appareil source et le moment où l'appareil source reçoit avec succès l'ACK Wi-Fi de l'appareil destination. Nous ajoutons une dernière métrique qui est la consommation d'énergie moyenne de chaque STA.

Quant à la méthode d'évaluation, il y a trois méthodes possibles comme l'explique clairement [Heidelberger et Lavenberg, 1984](#). La première est l'évaluation expérimentale. Elle a l'avantage d'être factuelle et complète, mais elle nécessite beaucoup de travail et d'équipements pour avoir des résultats. Elle limite donc énormément les scénarios explorables. La seconde est l'évaluation analytique, qui repose sur un modèle mathématique. Cette méthode met clairement en lumière les impacts de chaque paramètre sur les métriques observées. Cependant, un modèle analytique repose sur des hypothèses simplificatrices que nous ne pouvons pas nécessairement choisir. La dernière méthode est l'évaluation par simulation, ce qui correspond à une sorte de compromis entre les deux méthodes précédentes. Avec des simulations, on peut explorer un panel de scénarios beaucoup plus large qu'avec l'évaluation expérimentale. De plus, le degré d'abstraction peut effectivement être arbitraire, dans la mesure où l'on sait ce que prend en compte le simulateur, et ce qu'il ne prend pas en compte.

Pour notre étude, nous avons commencé par de la simulation, car c'est la méthode la plus accessible et elle permet de se forger des intuitions. Au fur et à mesure que notre compréhension des stratégies grandissait, nous avons construit un modèle analytique pour les représenter, et pour en tirer un guide de configuration. En parallèle, nous nous sommes intéressés à de l'évaluation expérimentale avec une carte Wi-Fi programmable.

### 4.1 SIMULATION AVEC WE3S

Pour mener notre évaluation par simulation, nous avons développé notre propre simulateur dédié à l'étude des stratégies d'économie d'énergie, que nous avons baptisé WE3S (*Wireless Energy Saving Strategies Simulator*).

Les simulateurs généralistes comme ns-3 implémentent beaucoup de mécanismes du Wi-Fi et proposent une image assez complète des performances que l'on peut ob-

tenir. Cependant, ce niveau de détails et de précision, qui est généralement un avantage, ne jouait pas en notre faveur au moment où nous commençons l'étude. En effet, nous voulions alors dégager les impacts des mécanismes d'économie d'énergie sur les performances, et prendre trop de facteurs en compte d'un coup risquait de les noyer.

De plus, l'effort de précision dans ns-3 est généralisé, mais ne se situe pas forcément là où nous en avons besoin. La preuve en est que ns-3 n'inclut nativement aucun mécanisme d'économie d'énergie. Des patches peuvent permettre de simuler certains mécanismes d'économie d'énergie dans ns-3, mais ils sont codés par des auteurs différents, et donc avec des degrés de détails différents. Par exemple, il existe un module présenté par Venkateswaran, Tai, Ben-Yehezkel et al., 2022 qui implémente le legacy PSM et un modèle d'énergie dans ns-3. Nous pouvons citer les deux études Tian, Deronne et al., 2016 et Tian, Šljivo et al., 2018 qui ajoutent l'amendement 802.11ah dans ns-3. Enfin, les auteurs de Venkateswaran, Tai, Garnayak et al., 2024 étendent le module 802.11ax de ns-3 pour y inclure TWT. Cela nous posait problème car nous souhaitions aussi pouvoir comparer des mécanismes entre eux.

Pour toutes ces raisons, nous avons opté pour un simulateur "maison", aussi simple que possible. Il nous permet de savoir exactement quels facteurs sont implémentés (et lesquels ne le sont pas), ce qui nous permet ensuite de dégager plus facilement les relations entre les facteurs et les performances. De la même manière, nous avons pu étudier les stratégies d'économie d'énergie au lieu des mécanismes, ce qui a simplifié l'étude.

#### 4.1.1 Présentation

WE3S est un simulateur à événements discrets codé en Python 3.10 qui est dédié à l'étude des stratégies d'économie d'énergie en Wi-Fi. Son code est disponible à l'adresse suivante : <https://github.com/EstGue/WE3S/> WE3S implémente les principaux mécanismes de la couche MAC du Wi-Fi, à savoir le CSMA/CA, ce qui inclut le backoff et le Binary Exponential Backoff. WE3S implémente également l'agrégation de trames et les TXOPs, présentés respectivement en sections 3.1.3 et 3.1.3. Par défaut, les appareils peuvent agréger jusqu'à huit trames, et les TXOPs ont une durée maximale de 3 ms (approximativement la valeur proposée par le standard pour la classe d'accès vidéo). Ces deux paramètres sont configurables dans WE3S. La couche PHY est largement simplifiée : les STAs utilisent un débit de données physique fixe pour envoyer ou recevoir des trames, et au lieu d'un modèle de propagation, les trames sont simplement sujettes à des erreurs aléatoires à la réception. De plus, nous négligeons les effets de capture : deux appareils qui tirent le même backoff seront toujours en collision, et devront tout deux réessayer plus tard d'envoyer leurs trames. WE3S n'implémente rien au-dessus de la couche MAC, les trames sont simplement générées indépendamment au niveau de l'AP et des STAs.

Les scénarios peuvent être variés dans WE3S. Les WLANs simulés n'ont forcément qu'un AP, mais peuvent inclure autant de STAs que souhaité. Chaque STA est définie selon les paramètres suivants : la capacité de son buffer, son débit de données physique et sa charge de travail descendante et montante. Son trafic peut être généré selon les modèles suivants : CBR (*Constant Bit Rate*), Poisson, distribution hyperexponentielle, ou avec une trace pré-enregistrée à partir de la sortie de la commande `tcpdump`. Additionnellement, les STAs peuvent implémenter les stratégies d'économie d'énergie telles qu'elles sont décrites dans la section 3.2 du chapitre 3. Cela peut être une stratégie qui restreint le trafic descendant, une stratégie qui restreint le trafic montant, ou une combinaison des deux.

Chaque simulation produit un registre d'événements, qui peut ensuite être exploité pour calculer des métriques sur la durée de la simulation. En particulier, ces métriques comprennent le débit descendant et le débit montant de chaque STA, leur taux de perte associé, et le délai moyen sur les trames. La consommation d'énergie moyenne des STAs est aussi incluse. Elle est calculée selon les facteurs discutés en

section 3.1.4 du chapitre 3 : les états de la carte Wi-Fi, les transitions entre hibernation et éveil, et le crossfactor. Chaque facteur peut être librement paramétré. Nous avons utilisé les valeurs suivantes, en nous inspirant des études de Halperin et al., 2010 et Venkateswaran, Tai, Ben-Yehzekel et al., 2022 :

- ❖ État “transmission” : 1.28 W
- ❖ État “réception” : 0.94 W
- ❖ État “inactif” : 0.82 W
- ❖ État “hibernation” : 0.1 W
- ❖ Transition d’hibernation à éveil : 130  $\mu$ J
- ❖ Transition d’éveil à hibernation : 112  $\mu$ J
- ❖ Crossfactor : 0 J

#### 4.1.2 Validation du partage du médium avec ns-3

WE3S étant un simulateur que nous avons développé et codé par nous-mêmes, nous devons nous assurer que ses simulations donnent des résultats fiables. Pour valider le partage du médium, nous comparons les résultats de WE3S avec ceux du simulateur ns-3, qui est généralement reconnu comme fiable par la communauté. Nous ne pouvons cependant pas utiliser ns-3 pour valider les stratégies d’économie d’énergie car elles n’y sont pas implémentées (et n’ont pas vocation à l’être vu que nos stratégies d’économie d’énergie sont des modélisations haut niveau des mécanismes d’économie d’énergie existants). Nous nous appuyons alors sur notre modèle analytique, décrit en section 4.2, pour vérifier que les simulations de WE3S sont cohérentes avec ses résultats. Nous comparons les résultats obtenus avec WE3S avec ceux prédits par notre modèle analytique en section 4.3.

Dans ns-3, nous plaçons quatre STAs sur un cercle d’un mètre autour d’un AP. Chaque STA utilise un débit de données physique fixe : le MCS HT15 avec un *Guard Interval* court sur un canal de 20 MHz, ce qui donne un débit physique de 162.5 Mbps. L’agrégation de trames est autorisée. Les STAs ont une charge de travail égale pour leur trafic montant, et n’ont pas de trafic descendant. En effet, la distribution des trafics descendants est géré à l’intérieur de l’AP, ce qui n’est pas l’aspect que nous souhaitons valider ici. Ces charges de travail sont générées par des flux UDP à débit constant, avec des paquets de 1440 octets. Nous simulons le même scénario avec WE3S en mettant à quatre le nombre de STAs, leur débit de données physique à 162.5 Mbps, et en générant leur trafic montant par un flux CBR.

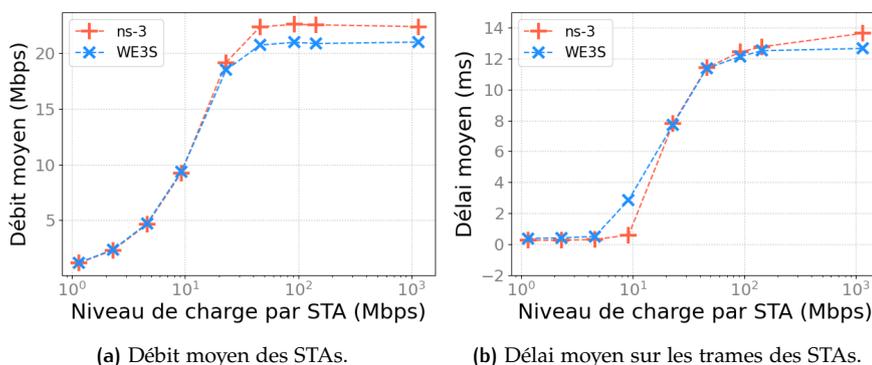


FIGURE 15 – Comparaison entre les métriques de performances obtenues avec ns-3 et celles obtenues avec WE3S.

La figure 15 compare les métriques obtenues par ns-3 (en rouge) et WE3S (en bleu) pour différentes charges de travail montante : la figure 15a présente les débits

montants obtenus par les STAs, et la figure 15b présente le délai moyen subi par les trames montantes.

Pour des charges de travail inférieures à 23.04 Mbps, ns-3 et WE3S obtiennent des performances quasiment identiques. Ils s'accordent aussi sur le moment où la saturation se met en place : les STAs commencent à perdre des paquets quand leur charge de travail dépasse 23.04 Mbps. Pour les charges de travail où les STAs sont en saturation, ns-3 obtient des débits légèrement supérieurs à ceux de WE3S, avec une différence allant jusqu'à 2 Mbps, soit 9 %. Cette différence est due à l'effet de capture : quand deux STAs tirent le même backoff, une des deux transmissions peut être parfois correctement reçue dans ns-3, là où elles sont systématiquement perdues pour WE3S. Au contraire, les délais sont un peu plus longs avec ns-3 qu'avec WE3S, avec une différence de 1 ms, soit 7 %. Nous expliquons cet écart par le fait que ns-3 implémente plusieurs niveaux de buffer là où WE3S n'en a qu'un. Dans l'ensemble, la figure 15 démontre la capacité de WE3S à délivrer des estimations justes des débits et des délais pour différents niveaux de charge.

## 4.2 MODÈLE ANALYTIQUE

Avec notre compréhension acquise via WE3S, nous avons construit des modèles analytiques permettant de calculer les métriques de performances d'un trafic restreint soit par la méthode "slot", soit par la méthode "prompt". Nous pouvons en dériver les performances acquises avec les différentes stratégies d'économie d'énergie (par exemple, nous pouvons calculer le débit descendant, le délai sur les trames descendantes et la consommation d'énergie d'une STA qui utilise la stratégie "DL slot" ou "DL prompt"). Nos modèles analytiques nous permettront d'abord de valider notre simulateur WE3S en section 4.3, puis dans le chapitre 5.1, de configurer efficacement les stratégies d'économie d'énergie en procurant un éclairage sur les liens entre configuration et métriques de performance.

Pour les hypothèses, nous supposons que le trafic est lisse dans le temps : les trames sont générées uniformément dans le temps. Cette hypothèse nous sert principalement pour calculer le délai sur les trames, puisque le cœur du calcul réside dans la différence entre le moment où une trame est générée et le moment où elle est envoyée.

Nos modèles prennent plusieurs paramètres en entrée, quoique nous avons limité leur nombre pour éviter des modèles trop tentaculaires, difficiles à utiliser. Ils sont au nombre de trois. Les deux premiers paramètres représentent les performances du trafic lorsque celui-ci n'est pas restreint par une méthode "slot" ou "prompt" : le débit maximum qu'il peut atteindre, noté  $X_{\emptyset}$ , et le délai sur ses trames, noté  $\tau_{\emptyset}$ . Ces deux paramètres intègrent l'impact du partage du médium avec des appareils concurrents, ainsi que les diverses pertes de trame dues à des collisions ou à des erreurs de transmission. Ces variables ne sont généralement pas connues en avance, mais des modèles peuvent les prédire (par exemple Bianchi, 2000, Wang et al., 2015). Pour notre étude, nous mesurons au besoin ces variables dans WE3S, pour éviter d'implémenter un nouveau modèle alors que WE3S est prêt à l'emploi. Nous ajoutons un troisième paramètre d'entrée, notée  $t_{\text{med}}$ , qui est le premier temps d'accès au médium au début d'une période de service. Ce paramètre permet de modéliser les mécanismes comme le probe delay qui augmentent significativement le temps d'accès au médium.

En sortie, nos modèles donnent les métriques de performance que nous avons considérées dans le reste de notre étude, à savoir le débit atteint, le délai moyen sur les trames, et la consommation d'énergie moyenne. La table 3 résume les différentes notations utilisées dans le modèle.

### 4.2.1 Méthode "slot"

Nous nous intéressons à un trafic, de niveau de charge  $\lambda$  et dont les trames sont de taille  $l_{\text{frame}}$ , restreint par la méthode "slot" avec  $D_{\text{slot}}$  comme durée des périodes de service et  $I_{\text{slot}}$  comme durée des périodes de restriction. Par ailleurs, l'appareil source (l'AP ou une STA, selon le trafic considéré) a un buffer de capacité  $l_{\text{buffer}}$ .

#### Débit

Avant de calculer le débit atteint par le trafic, nous nous intéressons aux causes de perte de paquets (ce qui nous sert dans le chapitre 5.1 pour configurer les stratégies d'économie d'énergie). Nous en avons identifiés deux.

La première cause de pertes de paquet est le mauvais dimensionnement du lien par rapport à la charge de travail qu'il doit supporter. Si l'on note  $X_{\text{dim}}$  le débit maximal que peut supporter le lien, alors on a :

$$\lambda > X_{\text{dim}} \Rightarrow \text{Pertes de paquet} \quad (1)$$

Si le trafic n'était pas restreint, on aurait  $X_{\text{dim}} = X_{\emptyset}$ . Dans le cas présent, le trafic est restreint par la méthode "slot", et il ne peut utiliser le lien que pendant  $D_{\text{slot}}$  avant d'attendre  $I_{\text{slot}}$  pour recommencer. Ainsi, on peut exprimer  $X_{\text{dim}}$  de la manière suivante :

$$X_{\text{dim}} = \frac{D_{\text{slot}}}{D_{\text{slot}} + I_{\text{slot}}} \times X_{\emptyset} \quad (2)$$

La seconde cause de pertes de paquet, indépendante de la première, est la mauvaise configuration de la durée des périodes de restriction. En effet, l'appareil source doit garder les paquets générés pendant les périodes de restriction dans son buffer, qui a une capacité limitée. Quand les périodes de restriction durent trop longtemps, le nombre de paquets générés dépasse la capacité du buffer de l'appareil source, et celui-ci est obligé de jeter des paquets. Si l'on note  $X_{\text{conf}}$  le débit maximal soutenable (c'est-à-dire n'impliquant pas de pertes de paquet) pendant les périodes de restriction, alors on a :

$$\lambda > X_{\text{conf}} \Rightarrow \text{Pertes de paquet} \quad (3)$$

Pour calculer  $X_{\text{conf}}$ , nous divisons la quantité maximale de données que peut contenir le buffer de l'appareil source ( $l_{\text{buffer}} \times l_{\text{frame}}$ ) par la durée de la période de restriction, à laquelle on ajoute le temps d'accès au médium ( $I_{\text{slot}} + t_{\text{med}}$ ). En effet, l'appareil source ne peut commencer à vider son buffer qu'après avoir accédé une première fois au médium. Ainsi, on a :

$$X_{\text{conf}} = \frac{l_{\text{buffer}} \times l_{\text{frame}}}{I_{\text{slot}} + t_{\text{med}}} \quad (4)$$

Ce sont les seules causes de pertes de paquet que nous considérons dans notre modèle. En particulier, nous négligeons les pertes liées à des retransmissions trop nombreuses, car leur probabilité est faible tant que le médium radio n'est pas trop bruité. Nous pouvons alors écrire l'équivalence suivante :

$$\text{Pertes de paquet} \Leftrightarrow \lambda > \min(X_{\text{dim}}, X_{\text{conf}}) \quad (5)$$

À présent, intéressons-nous au calcul du débit atteint  $X$ . Pour ce faire, nous calculons d'abord le nombre de trames qui sont correctement reçues sur une période de service et une période de restriction cumulées. Nous distinguons les trames selon qu'elles aient été générées pendant la période de service ou pendant la période de restriction. Pendant la période de restriction, le nombre de trames générées qui, à terme, seront correctement reçues, est le suivant :

$$n_{\text{rest}} = \min\left(l_{\text{buffer}}, \frac{\lambda}{l_{\text{frame}}} \times I_{\text{slot}}\right) \quad (6)$$

Pendant la période de service, nous supposons dans un premier temps que toutes les trames qui sont générées seront à terme correctement reçues par l'appareil destination (le cas inverse signifie que l'appareil source sature son lien) :

$$n_{\text{serv}} = \frac{\lambda}{l_{\text{frame}}} \times D_{\text{slot}} \quad (7)$$

Nous pouvons ensuite passer au calcul du débit moyen. Pour traiter le cas où l'appareil source est en saturation, nous limitons le débit atteint par  $X_{\text{dim}}$ . Pour le reste, il s'agit de diviser la quantité de données correctement reçues par le temps sur lequel elle est envoyée :

$$\begin{aligned} X &= \min \left[ X_{\text{dim}}, \frac{(n_{\text{rest}} + n_{\text{serv}}) \times l_{\text{frame}}}{D_{\text{slot}} + I_{\text{slot}}} \right] \\ &= \min \left[ X_{\text{dim}}, \frac{\min(l_{\text{buffer}} \times l_{\text{frame}}, \lambda \times I_{\text{slot}}) + \lambda \times D_{\text{slot}}}{D_{\text{slot}} + I_{\text{slot}}} \right] \\ &= \min \left[ X_{\text{dim}}, \min \left( \frac{l_{\text{buffer}} \times l_{\text{frame}} + \lambda \times D_{\text{slot}}}{D_{\text{slot}} + I_{\text{slot}}}, \frac{\lambda \times I_{\text{slot}} + \lambda \times D_{\text{slot}}}{D_{\text{slot}} + I_{\text{slot}}} \right) \right] \\ &= \min \left[ X_{\text{dim}}, \frac{l_{\text{buffer}} \times l_{\text{frame}} + \lambda \times D_{\text{slot}}}{D_{\text{slot}} + I_{\text{slot}}}, \lambda \right] \end{aligned} \quad (8)$$

### Délai

Pour calculer le délai, nous commençons par traiter le cas où l'appareil source est en saturation même si son trafic n'était pas restreint, ce qui est équivalent à  $\lambda > X_{\emptyset}$ . Nous utilisons la loi de Little pour calculer le délai, en considérant que (i) le buffer de l'appareil source est constamment rempli, et que (ii) le trafic s'écoule au débit maximal qu'il peut atteindre, soit  $X_{\text{dim}}$ .

$$\lambda \geq X_{\emptyset} \Rightarrow \tau = \frac{l_{\text{buffer}} \times l_{\text{frame}}}{X_{\text{dim}}} \quad (9)$$

Supposons à présent que  $\lambda < X_{\emptyset}$ . Pendant les périodes de service, l'appareil source envoie donc les trames à l'appareil destination plus vite qu'elles ne sont générées. Dans ce cas, nous pouvons observer deux régimes distincts pendant les périodes de service : un régime dit "par cycle" où l'appareil source est en train d'envoyer les trames accumulées pendant les périodes de restriction, et le régime non restreint, où le trafic s'écoule comme s'il n'était pas restreint. Nous reviendrons sur ces régimes en détails plus tard. Avant cela, nous pouvons d'ores et déjà exprimer le délai de la façon suivante :

$$\tau = \frac{n_{\text{cycle}} \times \tau_{\text{cycle}} + n_{\text{free}} \times \tau_{\emptyset}}{n_{\text{cycle}} + n_{\text{free}}} \quad (10)$$

où  $n_{\text{cycle}}$  est le nombre de trames envoyées pendant le régime par cycle,  $\tau_{\text{cycle}}$  le délai moyen subi par les trames envoyées pendant le régime par cycle et  $n_{\text{free}}$  le nombre de trames envoyées pendant le régime non restreint. Nous cherchons à présent à expliciter les termes de cette formule, en commençant par ceux concernant le régime par cycle.

Le régime par cycle se met en place au début des périodes de service. L'appareil source a accumulé pendant la période de restriction un certain nombre de trames pour l'appareil destination. Envoyer toutes ces trames lui prend un certain temps. Pendant ce temps, d'autres trames sont arrivées et se sont accumulées dans le buffer de l'appareil source, dans une quantité moindre qu'au cycle précédent. L'appareil source doit donc encore envoyer cette accumulation de trames à l'appareil destination, ce qui lui prend du temps, et ainsi de suite. La figure 16 illustre ce régime par cycle. L'appareil source a accumulé quatre trames pendant la période de restriction. Le premier cycle couvre la période pendant laquelle l'appareil source envoie les

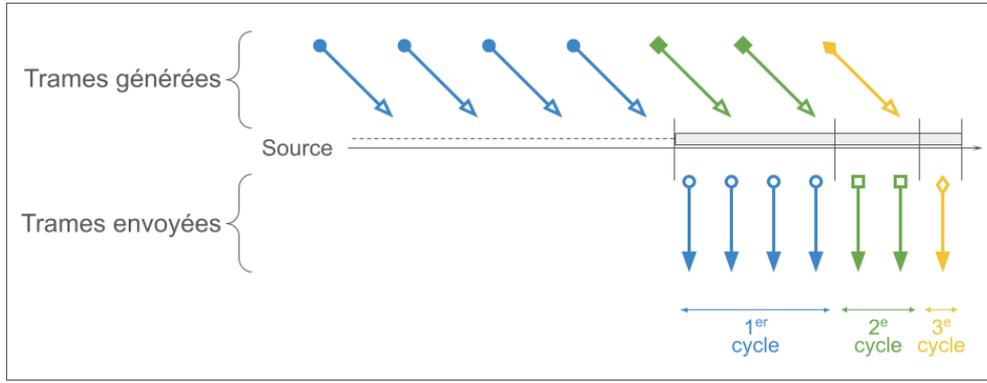


FIGURE 16 – Description du régime par cycle à l'issue d'une période de restriction.

quatre trames, générées pendant la période de restriction. Toujours pendant le premier cycle, deux autres trames ont été générées, l'appareil source les envoie donc pendant le second cycle. Une seule trame a été générée pendant le second cycle, et l'AP l'envoie pendant le troisième cycle, qui clôture le régime par cycle.

Commençons par définir deux fonctions : la fonction  $n_{frames}$ , qui calcule le nombre de trames générées pendant une période de temps donnée, et la fonction  $t_{serv}$ , qui calcule le temps nécessaire pour envoyer une quantité de trames donnée. On a :

$$\begin{cases} \forall t \in \mathbb{R}_+ & n_{frames}(t) = \min \left( l_{buffer}, \frac{\lambda}{l_{frame}} \times t \right) \\ \forall x \in \mathbb{R}_+ & t_{serv}(x) = \frac{l_{frame}}{X_{\emptyset}} \times x \end{cases} \quad (11)$$

À présent, nous modélisons le régime par cycle avec deux suites :

- ❖ La suite  $(n_k)_{k \in \mathbb{N}} \in \mathbb{R}^{\mathbb{N}}$  qui indique le nombre de trames à récupérer à chaque cycle.
- ❖ La suite  $(d_k)_{k \in \mathbb{N}} \in \mathbb{R}^{\mathbb{N}}$  qui indique le délai des trames à récupérer à chaque cycle.

La suite  $(n_k)_{k \in \mathbb{N}}$  est simple à définir par récursion, en utilisant la fonction  $n_{frames}$  :

$$\begin{cases} n_0 = n_{frames}(I_{slot}) \\ \forall k \in \mathbb{N} & n_{k+1} = n_{frames}(t_{serv}(n_k)) \end{cases} \quad (12)$$

La suite  $(d_k)_{k \in \mathbb{N}}$  est plus difficile à définir précisément. Dans le cas général, nous l'approximons par la durée du cycle concerné. Le cas particulier de  $d_0$  représente le délai subi par les trames générées pendant la période de restriction précédente, qui doivent donc attendre la prochaine période de service en plus de la durée de leur cycle. Comme par hypothèse, les trames sont générées uniformément dans le temps, elles sont censées attendre en moyenne la moitié de la durée de la période de restriction avant la prochaine période de service, soit  $\frac{1}{2} \times I_{slot}$ . Cependant, si des paquets sont perdus parce que le buffer de l'appareil source est plein, alors les trames ne sont plus générées uniformément dans le temps puisque l'appareil source jette les paquets les plus récents, et ils ne comptent plus dans le calcul du délai moyen. Les trames sont alors générées uniformément seulement sur le segment de temps où il n'y a pas de pertes de paquet. La figure 17 schématise le calcul de  $d_0$  dans le cas où il y a des pertes de paquet dues à une mauvaise configuration de la durée de la période de restriction. Au final, nous définissons la suite  $(d_k)_{k \in \mathbb{N}}$  comme suit :

$$\begin{cases} d_0 = I_{\text{slot}} + t_{\text{med}} - \frac{1}{2} \times \min \left( I_{\text{slot}}, \frac{l_{\text{buffer}} \times l_{\text{frame}}}{\lambda} \right) \\ \forall k \in \mathbb{N}^* \quad d_k = t_{\text{serv}}(n_k) \end{cases} \quad (13)$$

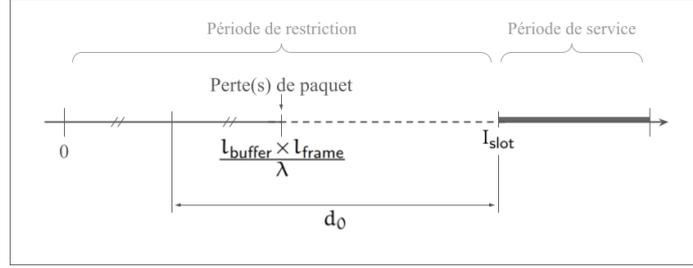


FIGURE 17 – Délai moyen des trames arrivant pendant une période de restriction dans le cas où il y a des pertes de paquet.

Nous pouvons à présent calculer le délai moyen des trames traitées pendant le régime par cycle, noté  $\tau_{\text{cycle}}$  : il suffit d'associer le délai des trames à leur nombre. Si l'on note  $n_{\text{cycle}}$  le nombre total de trames traitées pendant le régime par cycle, alors on a :

$$\tau_{\text{cycle}} = \frac{1}{n_{\text{cycle}}} \times \sum_{k=0}^{\infty} (n_k \times d_k) \quad (14)$$

Pour donner une forme close à cette expression, nous commençons par nous intéresser de plus près à la suite  $(n_k)_{k \in \mathbb{N}}$ . Soit  $k \in \mathbb{N}$ . On a :

$$\begin{aligned} n_{k+1} &= n_{\text{frames}}(t_{\text{serv}}(n_k)) \\ &= \min \left( l_{\text{buffer}}, \frac{\lambda}{l_{\text{frame}}} \times t_{\text{serv}}(n_k) \right) = \min \left( l_{\text{buffer}}, \frac{\lambda}{X_{\emptyset}} \times n_k \right) \end{aligned} \quad (15)$$

Or, nous avons supposé que  $\lambda < X_{\emptyset}$ . Comme  $n_0 \leq l_{\text{buffer}}$  par définition, on peut montrer par récursion que :

$$\forall k \in \mathbb{N} \quad \frac{\lambda}{X_{\emptyset}} \times n_k < l_{\text{buffer}} \quad (16)$$

Par conséquent, nous avons :

$$\forall k \in \mathbb{N} \quad n_{k+1} = \frac{\lambda}{X_{\emptyset}} \times n_k \quad (17)$$

La suite  $(n_k)_{k \in \mathbb{N}}$  est donc une suite géométrique de raison  $\lambda/X_{\emptyset}$ . Ses termes peuvent donc s'écrire ainsi :

$$\forall k \in \mathbb{N}^* \quad n_k = n_0 \times \left( \frac{\lambda}{X_{\emptyset}} \right)^k \quad (18)$$

Et nous pouvons calculer sa somme comme suit :

$$\begin{aligned} n_{\text{cycle}} &= \sum_{k=0}^{\infty} n_k \\ &= \frac{n_0}{1 - (\lambda/X_{\emptyset})} \\ &= \frac{n_0 \times X_{\emptyset}}{X_{\emptyset} - \lambda} \end{aligned} \quad (19)$$

Passons maintenant à la suite  $(n_k \times d_k)_{k \in \mathbb{N}}$ . Soit  $k \in \mathbb{N}^*$ . Alors :

$$\begin{aligned}
 n_k \times d_k &= n_0 \times \left( \frac{\lambda}{X_\emptyset} \right)^k \times t_{\text{serv}}(n_k) \\
 &= n_0 \times \left( \frac{\lambda}{X_\emptyset} \right)^k \times \frac{l_{\text{frame}}}{X_\emptyset} \times n_k \\
 &= n_0 \times \left( \frac{\lambda}{X_\emptyset} \right)^k \times \frac{l_{\text{frame}}}{X_\emptyset} \times n_0 \times \left( \frac{\lambda}{X_\emptyset} \right)^k \\
 &= n_0^2 \times \frac{l_{\text{frame}}}{X_\emptyset} \times \left( \frac{\lambda^2}{X_\emptyset^2} \right)^k
 \end{aligned} \tag{20}$$

La suite  $(n_k \times d_k)_{k \in \mathbb{N}}$  est donc une suite géométrique à partir de son terme de rang 1, de raison  $\lambda^2/X_\emptyset^2$ . Ainsi, nous pouvons calculer sa somme comme suit :

$$\begin{aligned}
 \sum_{k=0}^{\infty} [n_k \times d_k] &= n_0 \times d_0 + \sum_{k=1}^{\infty} [n_k \times d_k] \\
 &= n_0 \times d_0 + \sum_{k=1}^{\infty} \left[ \frac{n_0^2 \times l_{\text{frame}}}{X_\emptyset} \times \left( \frac{\lambda^2}{X_\emptyset^2} \right)^k \right] \\
 &= n_0 \times d_0 + \frac{n_0^2 \times l_{\text{frame}}}{X_\emptyset} \times \frac{\lambda^2}{X_\emptyset^2 \times (1 - \lambda^2/X_\emptyset^2)} \\
 &= n_0 \times d_0 + \frac{n_0^2 \times l_{\text{frame}} \times \lambda^2}{X_\emptyset \times (X_\emptyset^2 - \lambda^2)}
 \end{aligned} \tag{21}$$

On peut alors reprendre le calcul de  $\tau_{\text{cycle}}$  :

$$\begin{aligned}
 \tau_{\text{cycle}} &= \frac{1}{n_{\text{cycle}}} \times \sum_{k=0}^{\infty} [n_k \times d_k] \\
 &= \frac{X_\emptyset - \lambda}{n_0 \times X_\emptyset} \times n_0 \times d_0 + \frac{X_\emptyset - \lambda}{n_0 \times X_\emptyset} \times \frac{n_0^2 \times l_{\text{frame}} \times \lambda^2}{X_\emptyset \times (X_\emptyset^2 - \lambda^2)} \\
 &= \frac{(X_\emptyset - \lambda) \times d_0}{X_\emptyset} + \frac{n_0 \times l_{\text{frame}} \times \lambda^2}{X_\emptyset^2 \times (X_\emptyset + \lambda)}
 \end{aligned} \tag{22}$$

Nous passons maintenant au régime non restreint, qui arrive après que le régime par cycle se soit achevé. Le régime non restreint est le même régime que celui du trafic lorsqu'il n'est pas restreint, par conséquent ses trames subissent en moyenne un délai égal à  $\tau_\emptyset$ .

La question est de savoir combien de trames sont concernées par le régime non restreint, valeur que l'on note  $n_{\text{free}}$ . Pour cela, il faut connaître le temps que dure le régime non restreint, et donc le temps que dure le régime par cycle (puisque le régime non restreint occupe le reste du temps des périodes de service). Nous notons  $t_{\text{free}}$  la durée du régime non restreint, et  $t_{\text{cycle}}$  la durée du régime par cycle.

Avec les calculs que nous avons déjà menés pour le régime par cycle, le calcul de  $t_{\text{cycle}}$  est assez direct. Il suffit de calculer le temps de service de l'ensemble des trames traitées pendant le régime par cycle :

$$\begin{aligned}
 t_{\text{cycle}} &= t_{\text{serv}}(n_{\text{cycle}}) \\
 &= \frac{l_{\text{frame}}}{X_\emptyset} \times \frac{n_0 \times X_\emptyset}{X_\emptyset - \lambda} \\
 &= \frac{l_{\text{frame}} \times n_0}{X_\emptyset - \lambda}
 \end{aligned} \tag{23}$$

Nous faisons alors face à plusieurs possibilités. Supposons que  $t_{\text{cycle}}$  est supérieur à  $D_{\text{slot}}$ . Alors, cela signifie que le lien est sous-dimensionné par rapport au niveau

de charge du trafic. Dans ce cas, nous utilisons la loi de Little pour calculer le délai moyen des trames, comme nous l'avons fait dans le cas où  $\lambda \geq X_{\emptyset}$  :

$$t_{\text{cycle}} > D_{\text{slot}} \Rightarrow \tau = \frac{l_{\text{buffer}} \times l_{\text{frame}}}{X_{\text{dim}}} \quad (24)$$

Supposons qu'au contraire  $t_{\text{cycle}}$  est inférieur à  $D_{\text{slot}}$ . Alors, le régime non restreint existe et nous pouvons calculer sa durée comme la différence entre  $D_{\text{slot}}$  et  $t_{\text{cycle}}$  :

$$\begin{aligned} t_{\text{free}} &= D_{\text{slot}} - t_{\text{cycle}} \\ &= D_{\text{slot}} - \frac{l_{\text{frame}} \times n_0}{X_{\emptyset} - \lambda} \end{aligned} \quad (25)$$

Enfin, nous pouvons calculer le nombre de trames qui sont générées pendant le régime non restreint comme suit :

$$\begin{aligned} n_{\text{free}} &= \frac{t_{\text{free}} \times \lambda}{l_{\text{frame}}} \\ &= \frac{D_{\text{slot}} \times \lambda}{l_{\text{frame}}} - \frac{n_0 \times \lambda}{X_{\emptyset} - \lambda} \end{aligned} \quad (26)$$

### Récapitulatif du calcul du délai moyen

Nous proposons ici un bref récapitulatif des formules pour calculer le délai moyen.

Si  $\lambda \geq X_{\emptyset}$ , alors :

$$\tau = \frac{l_{\text{buffer}} \times l_{\text{frame}}}{X_{\text{dim}}}$$

Supposons maintenant que  $\lambda < X_{\emptyset}$ . Soit  $t_{\text{cycle}} \in \mathbb{R}_+$  tel que :

$$t_{\text{cycle}} = \frac{l_{\text{frame}}}{X_{\emptyset} - \lambda} \times \min \left( l_{\text{buffer}}, \frac{\lambda}{l_{\text{frame}}} \times I_{\text{slot}} \right)$$

Si  $t_{\text{cycle}} > D_{\text{slot}}$ , alors :

$$\tau = \frac{l_{\text{buffer}} \times l_{\text{frame}}}{X_{\text{dim}}}$$

Sinon,  $t_{\text{cycle}} \leq D_{\text{slot}}$ , et nous obtenons :

$$\tau = \frac{n_{\text{cycle}} \times \tau_{\text{cycle}} + n_{\text{free}} \times \tau_{\emptyset}}{n_{\text{cycle}} + n_{\text{free}}}$$

$$\text{avec } \begin{cases} n_{\text{cycle}} = \frac{X_{\emptyset}}{X_{\emptyset} - \lambda} \times \min \left( l_{\text{buffer}}, \frac{\lambda}{l_{\text{frame}}} \times I_{\text{slot}} \right) \\ \tau_{\text{cycle}} = \frac{(X_{\emptyset} - \lambda) \times d_0}{X_{\emptyset}} + \frac{l_{\text{frame}} \times \lambda^2}{X_{\emptyset}^2 \times (X_{\emptyset} + \lambda)} \times \min \left( l_{\text{buffer}}, \frac{\lambda}{l_{\text{frame}}} \times I_{\text{slot}} \right) \\ n_{\text{free}} = \frac{D_{\text{slot}} \times \lambda}{l_{\text{frame}}} - \frac{\lambda}{X_{\emptyset} - \lambda} \times \min \left( l_{\text{buffer}}, \frac{\lambda}{l_{\text{frame}}} \times I_{\text{slot}} \right) \\ d_0 = I_{\text{slot}} + t_{\text{med}} - \frac{1}{2} \times \min \left( I_{\text{slot}}, \frac{l_{\text{buffer}} \times l_{\text{frame}}}{\lambda} \right) \end{cases}$$

### Consommation d'énergie

Dans le cas particulier de la consommation d'énergie, nous sommes obligés de définir le type de trafic restreint par la méthode "slot". En effet, restreindre le trafic montant n'a pas du tout les mêmes conséquences sur la consommation d'énergie

de la STA que restreindre son trafic descendant. Si le trafic restreint est le trafic montant de la STA, alors cette STA reste toujours éveillée pour pouvoir recevoir les trames envoyées par l'AP, qui, elles, ne sont pas restreintes.

Nous supposons donc pour cette section que la STA restreint son trafic descendant, et que son trafic montant est laissé libre. Nous faisons également la distinction entre  $\lambda_{DL}$  comme étant le niveau de charge du trafic descendant, et  $\lambda_{UL}$  le niveau de charge du trafic montant. Nous cherchons à modéliser la consommation d'énergie de cette STA, et en particulier l'économie apportée par l'hibernation de la STA.

Nous commençons par calculer le pourcentage de temps passé en hibernation, noté  $T_{doze}$ , et la fréquence des passages entre hibernation et éveil, noté  $F_{doze}$ . La STA est autorisée à hiberner uniquement durant les périodes de restriction, qui durent  $I_{slot}$ . Elle n'hiberne cependant pas la totalité des périodes de restriction, car elle envoie son trafic montant quand des trames arrivent dans son buffer. Par conséquent, le temps que la STA peut passer en hibernation s'écrit :

$$T_{doze} = \frac{I_{slot}}{D_{slot} + I_{slot}} \times \left[ 1 - \min \left( 1, \frac{\lambda_{UL}}{X_{\emptyset}} \right) \right] \quad (27)$$

Pour la fréquence de passage entre hibernation et éveil, notée  $F_{doze}$ , nous devons compter combien de fois la STA doit se réveiller. Pour le trafic descendant, la STA doit se réveiller uniquement pour ses périodes de service, ce qui arrive une fois par  $D_{slot} + I_{slot}$ . Pour le trafic montant, la STA doit se réveiller chaque fois qu'une trame montante est générée pendant ses périodes de restriction du trafic descendant. Au total, nous définissons  $F_{doze}$  comme suit :

$$\begin{aligned} F_{doze} &= F_{DL} + F_{UL} \\ F_{DL} &= \frac{1}{D_{slot} + I_{slot}} \\ \text{avec} \\ F_{UL} &= \frac{I_{slot} \times \lambda_{UL}}{l_{frame} \times (D_{slot} + I_{slot})} \end{aligned} \quad (28)$$

De même, pour la fréquence des passages entre hibernation et éveil, nous prenons en compte à la fois les périodes de restriction et de service, et le trafic montant à envoyer.

$$F_{doze} = \frac{l_{frame} + \lambda_{UL} \times I_{slot}}{l_{frame} \times (D_{slot} + I_{slot})} \quad (29)$$

Nous pouvons ensuite donner une estimation de la consommation d'énergie finale de la STA, notée  $P$ . Commençons par déterminer si la fréquence de passage entre hibernation et éveil permet à la STA d'hiberner. En effet, entrer et revenir d'hibernation demande un peu de temps à la STA, que l'on note  $t_{switch}$ . Si la fréquence des passages entre hibernation et éveil est trop importante, la STA n'a pas le temps d'hiberner, et alors :

$$F_{doze}^{-1} \leq t_{switch} \quad \Rightarrow \quad P = p_{awake} \quad (30)$$

Sinon, nous lions le temps passé en hibernation ( $T_{doze}$ ) à la consommation en hibernation ( $p_{doze}$ ), le temps passé en éveil ( $1 - T_{doze}$ ) à la consommation en éveil ( $p_{awake}$ ), et nous ajoutons le coût énergétique dû aux passages entre éveil et hibernation ( $F_{doze} \times e_{switch}$ ). Au total, cela donne :

$$F_{doze}^{-1} > t_{switch} \quad \Rightarrow \quad P = T_{doze} \times p_{doze} + (1 - T_{doze}) \times p_{awake} + F_{doze} \times e_{switch} \quad (31)$$

Le dernier cas à traiter serait le cas où le trafic descendant et le trafic montant sont restreints. Si les périodes de service du trafic descendant sont exactement identiques à celles du trafic montant, alors le modèle précédent peut être facilement adapté. Sinon, le modèle devient beaucoup plus complexe car il nécessite entre autres de calculer les intersection entre les deux séquences de périodes de service. Comme indiqué en chapitre 3, nous considérons dans notre étude seulement les cas où un seul des trafics est restreint. Nous ne détaillerons donc pas plus ce cas.

### 4.2.2 Méthode "prompt"

Nous considérons maintenant que le trafic précédent est restreint par la méthode "prompt". Comme la méthode "slot", la méthode "prompt" produit un régime par cycle à l'issue des périodes de restriction, le temps que l'appareil source envoie toutes les trames qu'il a accumulées. Mais contrairement à la méthode "slot", le régime par cycle définit les périodes de service de l'appareil destination, car celles-ci ne se terminent que lorsque l'appareil source n'a plus aucune trame en attente pour l'appareil destination. La durée des périodes de service n'est donc pas limitée. Nous commençons donc par modéliser le régime par cycle pour avoir accès à la durée des périodes de service.

Commençons par calculer le débit maximal atteignable par le trafic lorsque celui-ci est restreint par la méthode "prompt". Il est atteint lorsque l'appareil destination renvoie un prompt aussi vite que possible après que l'appareil source ait terminé sa TXOP précédente. Concrètement, le débit est quasiment égal à celui atteignable lorsque le trafic n'est pas restreint, le seul surcoût étant le temps que passe l'appareil destination à envoyer ses trames prompt avant chaque TXOP, que l'on note  $t_{\text{prompt}}$ . Par conséquent, nous pouvons calculer le débit maximal atteignable sur lien restreint comme suit :

$$X_{\text{dim}} = \left(1 - \frac{t_{\text{prompt}} \times X_{\emptyset}}{n_{\text{TXOP}} \times n_{\text{agg}} \times l_{\text{frame}}}\right) \times X_{\emptyset} \quad (32)$$

$t_{\text{prompt}} \times X_{\emptyset}$  correspond à la quantité de données qui ne peut pas être reçue par l'appareil destination car le médium est occupé par l'envoi d'un prompt par l'appareil destination. On le rapporte à la quantité de données maximale qui peut être envoyée sur une TXOP, à savoir  $n_{\text{TXOP}} \times n_{\text{agg}} \times l_{\text{frame}}$ .

De nouveau, nous définissons les fonctions  $n_{\text{frames}}$  et  $t_{\text{serv}}$ , qui calculent respectivement le nombre de trames générées sur une période de temps donnée et le temps nécessaire pour envoyer une quantité de trames donnée.  $n_{\text{frame}}$  est exactement identique à celle utilisée pour la méthode "slot" (équation 11), tandis que  $t_{\text{serv}}$  est adaptée pour indiquer que les trames sont écoulées au débit  $X_{\text{dim}}$  et non  $X_{\emptyset}$  (car le débit maximal atteignable est modifié par les trames prompt) :

$$\begin{cases} \forall t \in \mathbb{R}_+ & n_{\text{frames}}(t) = \min\left(l_{\text{buffer}}, \frac{\lambda}{l_{\text{frame}}} \times t\right) \\ \forall x \in \mathbb{R}_+ & t_{\text{serv}}(x) = \frac{l_{\text{frame}}}{X_{\text{dim}}} \times x \end{cases} \quad (33)$$

De même, nous redéfinissons la suite  $(n_k)_{k \in \mathbb{N}}$  représentant le nombre de trames à récupérer à chaque cycle. Pour la méthode "prompt", sa définition est la suivante :

$$\begin{cases} n_0 = n_{\text{frames}}(I_{\text{prompt}}) \\ \forall k \in \mathbb{N} & n_{k+1} = n_{\text{frames}}(t_{\text{serv}}(n_k)) \end{cases} \quad (34)$$

En supposant que  $\lambda < X_{\text{dim}}$ , avec des calculs similaires à ceux utilisés pour la méthode "slot", nous pouvons exprimer les termes de la suite  $(n_k)_{k \in \mathbb{N}}$  de la manière suivante :

$$\begin{cases} n_0 = n_{\text{frames}}(I_{\text{prompt}}) \\ \forall k \in \mathbb{N}^* & n_k = n_0 \times \left(\frac{\lambda}{X_{\text{dim}}}\right)^k \end{cases} \quad (35)$$

Et ainsi, nous calculons sa série, qui représente la quantité de trames envoyées pendant une période de service et que nous notons  $n_{\text{cycle}}$ , ainsi que la durée nécessaire pour envoyer ces trames, ce qui correspond à la durée des périodes de services, notée  $t_{\text{cycle}}$ .

$$\begin{cases} n_{\text{cycle}} = \frac{n_0 \times X_{\text{dim}}}{X_{\text{dim}} - \lambda} \\ t_{\text{cycle}} = t_{\text{serv}}(n_{\text{cycle}}) = \frac{l_{\text{frame}} \times n_0}{X_{\text{dim}} - \lambda} \end{cases} \quad (36)$$

### Débit

Avant de calculer le débit atteint par le trafic, nous cherchons à exprimer les causes de pertes de paquet (ce qui nous est utile plus tard pour configurer les stratégies d'économie d'énergie). La première cause de perte de paquet est le sous-dimensionnement du lien par rapport à la charge de travail du trafic : les trames arrivent plus vite dans le buffer de l'appareil source qu'il ne peut les envoyer à l'appareil destination. Nous exprimons cette première cause de la manière suivante :

$$\lambda > X_{\text{dim}} \Rightarrow \text{Pertes de paquet} \quad (37)$$

La seconde cause est la mauvaise calibration de la durée de la période de restriction (égale à  $I_{\text{prompt}}$ ). En effet, l'appareil source doit stocker dans son buffer (qui peut contenir  $l_{\text{buffer}}$  trames au maximum) toutes les trames pour l'appareil destination pendant les périodes de restriction. Si la période de restriction dure trop longtemps par rapport à la charge de travail du trafic, alors l'appareil source est obligé de jeter des trames. Nous exprimons alors le débit maximal soutenable pendant les périodes de restriction de la manière suivante :

$$X_{\text{conf}} = \frac{l_{\text{buffer}} \times l_{\text{frame}}}{I_{\text{prompt}} + t_{\text{med}}} \quad (38)$$

Et nous obtenons alors l'implication suivante :

$$\lambda > X_{\text{conf}} \Rightarrow \text{Pertes de paquet} \quad (39)$$

Nous négligeons les autres causes de pertes de paquet, notamment celles liées à de trop nombreuses retransmissions. Nous avons donc l'équivalence suivante :

$$\text{Pertes de paquet} \Leftrightarrow \lambda > \min(X_{\text{dim}}, X_{\text{conf}}) \quad (40)$$

Nous pouvons maintenant calculer le débit atteint par le trafic, noté  $X$ . Nous commençons par traiter le cas où le lien physique est saturé par la charge de travail, c'est-à-dire quand  $\lambda > X_{\text{dim}}$ . Dans ce cas, le trafic occupe la totalité du lien physique :

$$\lambda \geq X_{\text{dim}} \Rightarrow X = X_{\text{dim}} \quad (41)$$

Supposons maintenant que  $\lambda \leq X_{\text{dim}}$ . Alors, nous nous appuyons sur la suite  $(n_k)_{k \in \mathbb{N}}$  pour calculer le débit atteint. Le débit atteint par le trafic est égal à la quantité de données envoyées ( $n_{\text{cycle}} \times l_{\text{frame}}$ ) divisée par le temps pour les envoyer ( $t_{\text{cycle}} + I_{\text{prompt}}$ ) :

$$\begin{aligned} X &= \frac{n_{\text{cycle}} \times l_{\text{frame}}}{t_{\text{cycle}} + I_{\text{prompt}}} \\ &= \frac{n_0 \times X_{\text{dim}} \times l_{\text{frame}}}{(X_{\text{dim}} - \lambda) \times \left( \frac{l_{\text{frame}} \times n_0}{X_{\text{dim}} - \lambda} + I_{\text{prompt}} \right)} \\ &= \frac{n_0 \times X_{\text{dim}} \times l_{\text{frame}}}{l_{\text{frame}} \times n_0 + (X_{\text{dim}} - \lambda) \times I_{\text{prompt}}} \end{aligned} \quad (42)$$

### Délai

Nous commençons par traiter le cas en saturation, c'est-à-dire quand  $\lambda \geq X_{\text{dim}}$ . Nous utilisons alors la loi de Little en considérant que le buffer de l'appareil source est constamment rempli à son maximum. Nous avons donc :

$$\lambda \geq X_{\text{dim}} \Rightarrow \tau = \frac{l_{\text{buffer}} \times l_{\text{frame}}}{X_{\text{dim}}} \quad (43)$$

Nous supposons à présent que  $\lambda < X_{\text{dim}}$ . Pour calculer le délai sur les trames, nous réutilisons la suite  $(n_k)_{k \in \mathbb{N}}$ , et définissons la suite  $(d_k)_{k \in \mathbb{N}}$  qui représente le délai des trames générées à chaque cycle. Nous utilisons une définition très proche de celle utilisée pour la méthode "slot" :

$$\begin{cases} d_0 = I_{\text{prompt}} + t_{\text{med}} - \frac{1}{2} \times \min \left( I_{\text{prompt}}, \frac{l_{\text{frame}}}{\lambda} \times l_{\text{buffer}} \right) \\ \forall k \in \mathbb{N}^* \quad d_k = t_{\text{serv}}(n_k) \end{cases} \quad (44)$$

Le calcul du délai moyen est alors le suivant :

$$\begin{aligned} \tau &= \frac{1}{n_{\text{cycle}}} \times \sum_{k=0}^{\infty} (n_k \times d_k) \\ &= \frac{n_0 \times d_0}{n_{\text{cycle}}} + \frac{1}{n_{\text{cycle}}} \times \sum_{k=1}^{\infty} (n_k \times d_k) \end{aligned} \quad (45)$$

Pour simplifier, commençons par calculer la somme. Soit  $k \in \mathbb{N}^*$ , alors :

$$\begin{aligned} n_k \times d_k &= n_0 \times \left( \frac{\lambda}{X_{\text{dim}}} \right)^k \times \frac{l_{\text{frame}}}{X_{\text{dim}}} \times n_0 \times \left( \frac{\lambda}{X_{\text{dim}}} \right)^k \\ &= \frac{n_0^2 \times l_{\text{frame}}}{X_{\text{dim}}} \times \left( \frac{\lambda^2}{X_{\text{dim}}^2} \right)^k \end{aligned} \quad (46)$$

Et donc :

$$\begin{aligned} \sum_{k=1}^{\infty} (n_k \times d_k) &= \frac{n_0^2 \times l_{\text{frame}}}{X_{\text{dim}}} \times \frac{\lambda^2}{X_{\text{dim}}^2 \times (1 - (\lambda^2/X_{\text{dim}}^2))} \\ &= \frac{n_0^2 \times l_{\text{frame}} \times \lambda^2}{X_{\text{dim}} \times (X_{\text{dim}}^2 - \lambda^2)} \end{aligned} \quad (47)$$

Enfin, nous avons donc :

$$\begin{aligned} \tau &= \frac{n_0 \times d_0}{n_{\text{cycle}}} + \frac{X_{\text{dim}} - \lambda}{n_0 \times X_{\text{dim}}} \times \frac{n_0^2 \times l_{\text{frame}} \times \lambda^2}{X_{\text{dim}} \times (X_{\text{dim}}^2 - \lambda^2)} \\ &= \frac{n_0 \times d_0}{n_{\text{cycle}}} + \frac{n_0 \times l_{\text{frame}} \times \lambda^2}{X_{\text{dim}}^2 \times (X_{\text{dim}} + \lambda)} \end{aligned} \quad (48)$$

### Consommation d'énergie

Comme pour le modèle de la méthode "slot", nous supposons pour la consommation d'énergie que le trafic restreint par la méthode "prompt" est le trafic descendant. Nous pouvons donc estimer la consommation d'énergie de la STA en calculant le temps passé en hibernation. À l'opposé, le trafic montant de la STA est supposé être libre, et la STA envoie les trames montantes dès qu'elles arrivent dans son buffer. Nous faisons donc la distinction entre  $\lambda_{\text{DL}}$  comme la charge du trafic descendant, et  $\lambda_{\text{UL}}$  la charge du trafic montant.

Nous traitons en premier le cas où le lien est saturé, c'est-à-dire quand  $\lambda_{\text{DL}}$  est supérieur à  $X_{\text{dim}}$ . Alors, la STA a toujours des trames en attente au niveau de l'AP,

et est donc constamment en train d'envoyer des trames prompts. Elle ne peut donc pas entrer en hibernation :

$$\lambda_{DL} \geq X_{dim} \Rightarrow T_{doze} = 0 \quad (49)$$

Supposons maintenant que  $\lambda_{DL} < X_{dim}$ . On sait que la STA a le droit d'hiberner durant les périodes de restriction, qui durent  $I_{prompt}$ , mais elle est obligée de rester éveillée pendant les périodes de service, qui durent  $t_{cycle}$ . Cependant, la STA n'hiberne pas pendant la totalité des périodes de restriction. En effet, dès qu'une trame montante arrive dans son buffer, la STA se réveille pour envoyer cette trame, avant de retourner en hibernation. Les trames montantes sont envoyées avec un débit physique non restreint, soit  $X_{\emptyset}$ , et leur trafic représente une charge de travail notée  $\lambda_{UL}$ . Par conséquent, envoyer le trafic montant occupe un pourcentage de temps égal à  $\min(1, \lambda_{UL}/X_{\emptyset})$ . De plus, les trames montantes arrivent dans le buffer de la STA à la fréquence de  $\lambda_{UL}/l_{frame}$ , donc il y en a  $I_{prompt} \times \lambda_{UL}/l_{frame}$  qui arrivent pendant la période de restriction. Au total, nous pouvons donc calculer le pourcentage de temps que la STA passe en hibernation, ainsi que la fréquence à laquelle elle passe d'hibernation et éveil et inversement :

$$\lambda < X_{dim} \Rightarrow \begin{cases} T_{doze} = \frac{I_{prompt}}{I_{prompt} + t_{cycle}} \times \left[ 1 - \min\left(1, \frac{\lambda_{UL}}{X_{\emptyset}}\right) \right] \\ F_{doze} = \frac{1}{I_{prompt} + t_{cycle}} + \frac{I_{prompt} \times \lambda_{UL}}{l_{frame} \times (I_{prompt} + t_{cycle})} \end{cases} \quad (50)$$

Ensuite, nous pouvons estimer la consommation d'énergie finale de la STA, notée  $P$ . En premier lieu, nous regardons si la fréquence de passage entre hibernation et éveil ( $F_{doze}$ ) permet à la STA d'hiberner. C'est le cas si  $F_{doze}^{-1} > t_{switch}$ . Dans ce cas, nous associons le temps passé en hibernation  $T_{doze}$  à la consommation en hibernation  $p_{doze}$ , le temps passé en éveil ( $1 - T_{doze}$ ) à la consommation en éveil  $p_{awake}$ , et nous ajoutons le coût des passages entre hibernation et éveil. Sinon, la STA reste constamment éveillée, et sa consommation est alors égale à  $p_{awake}$ .

Pour récapituler, la consommation de la STA est donc la suivante :

$$\begin{cases} \lambda \geq X_{dim} \Rightarrow P = p_{awake} \\ \lambda < X_{dim} \text{ et } F_{doze}^{-1} < t_{switch} \Rightarrow P = p_{awake} \\ \lambda < X_{dim} \text{ et } F_{doze}^{-1} \geq t_{switch} \Rightarrow \\ P = T_{doze} \times p_{doze} + (1 - T_{doze}) \times p_{awake} + F_{doze} \times e_{switch} \end{cases} \quad (51)$$

## 4.3 COMPARAISON ENTRE WE3S ET LE MODÈLE ANALYTIQUE

Dans cette section, nous vérifions que les prédictions de notre modèle analytique concordent avec les résultats obtenus par WE3S. Nous garantissons ainsi la fiabilité de WE3S pour simuler les stratégies d'économie d'énergie, car nous prouvons que nous pouvons justifier ses résultats. Nous effectuons cette vérification pour le débit (section 4.3.1), le délai (section 4.3.2) et la consommation d'énergie (section 4.3.3).

### 4.3.1 Débit

Dans WE3S, nous considérons un scénario où quatre STAs sont associées à un AP. Une STA dite "observée" utilise soit la stratégie "DL slot", soit la stratégie "DL prompt" avec une durée de périodes de restriction ( $I_{slot}$  ou  $I_{prompt}$ ) égale à 90 ms, et pour la stratégie "DL slot", une durée de périodes de service ( $D_{slot}$ ) de 20 ms. Cette

STA n'a que du trafic descendant, dont l'intensité varie entre 1 et 100 Mbps. Les autres STAs génèrent du trafic de fond, sans restriction. Les spécifications précises du scénario sont les suivantes :

- ❖ Taille des trames : 1250 octets
- ❖ Débit physique : 100 Mbps
- ❖ Niveau de charge de la STA observée : entre 1 Mbps et 100 Mbps
- ❖ Niveau de charge total du trafic de fond : 6 Mbps

Le modèle analytique prend trois paramètres en entrée. Nous donnons la valeur de 71 Mbps à  $X_{\emptyset}$ , le débit maximal atteignable lorsque le trafic n'est pas restreint ; 270  $\mu$ s pour  $\tau_{\emptyset}$ , le délai sur les trames du trafic lorsque celui-ci n'est pas restreint ; et 106  $\mu$ s pour  $t_{med}$ , le premier temps d'accès au médium au début d'une période de service. Nous avons obtenus ces valeurs avec WE3S (à noter que cet aspect de WE3S a été validé avec ns-3, puisque le trafic n'est pas restreint).

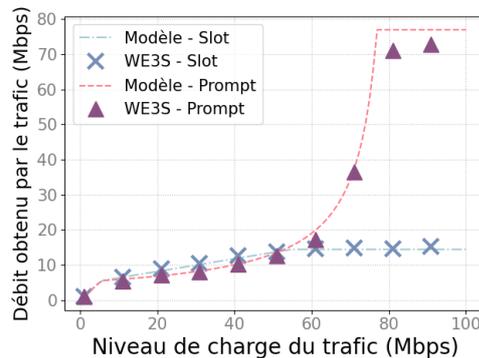


FIGURE 18 – Comparaison du débit descendant obtenu par la STA observée selon les résultats de WE3S et selon la prédiction de notre modèle analytique.

La figure 18 compare les résultats obtenus avec WE3S avec les prédictions des modèles analytiques. Les modèles analytiques semblent correctement capturer les variations des métriques en fonction du niveau de charge de la STA observée.

#### 4.3.2 Délai

Nous utilisons le même scénario pour valider le délai que pour valider le débit. La figure 19 montre de nouveau que les modèles analytiques reflètent correctement les résultats obtenus avec WE3S.

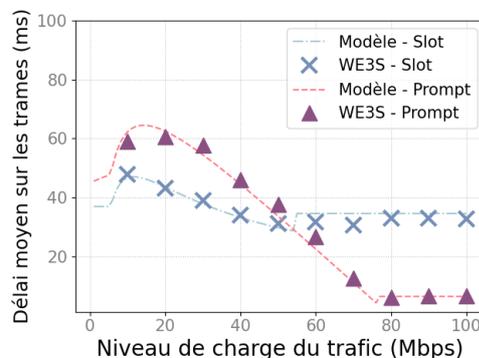


FIGURE 19 – Comparaison des délais sur les trames du trafic descendant de la STA observée tels que mesurés sur WE3S par rapport aux prédictions du modèle analytique.

### 4.3.3 Consommation d'énergie

Pour valider la consommation d'énergie, nous modifions la STA observée pour qu'elle ait une charge de travail fixe à 1 Mbps pour le trafic descendant et pour le trafic montant, puis nous faisons varier la durée de sa période de restriction entre 1 et 50 ms.

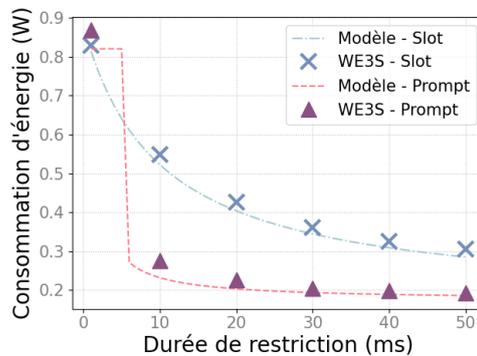


FIGURE 20 – Comparaison de la consommation d'énergie de la STA observée selon WE3S et selon le modèle analytique.

La figure 20 présente les résultats obtenus avec WE3S par rapport aux prédictions du modèle analytique. Comme pour les autres métriques de performances, les modèles analytiques sont relativement proches des résultats obtenus avec WE3S.

## 4.4 ÉVALUATION EXPÉRIMENTALE

Pour compléter notre étude, nous avons jugé souhaitable de nous confronter à de vrais appareils Wi-Fi, pour voir les éventuelles différences par rapport à WE3S et notre modèle analytique. Cette partie a été confiée à Tom Ogier pour son stage de Master 2. L'objectif initial était de mettre en œuvre un ou plusieurs mécanismes d'économie d'énergie sur un appareil Wi-Fi puis de mesurer leurs impacts sur les métriques de performances. Nous aurions pu ainsi comparer les résultats des simulations WE3S, du modèle analytique et de l'expérimental. Bien que nous n'ayons pas atteint cet objectif, nous avons su dessiner une partie des obstacles à l'évaluation expérimentale des mécanismes d'économie d'énergie, que nous rapportons dans la présente section.

### 4.4.1 Plateforme expérimentale

Notre plateforme expérimentale, dont la photographie est présentée en figure 21, comporte un AP et une STA qui lui est associée. Les appareils utilisés sont des *FiPys*. Par ailleurs, nous utilisons des ampèremètres *Power Profiler Kit II* pour mesurer la consommation d'énergie de la STA et de l'AP.

Les *FiPys* sont des modules de développement IoT ou M2M (*Machine to Machine*) conçus par la société PyCom. En particulier, ils intègrent beaucoup de protocoles de communication sans fil : le Wi-Fi, le Bluetooth, le LoRa, le Sigfox et le LTE-M, qui peuvent chacun être activé ou désactivé à la volée. Dans notre cas, la fonctionnalité qui nous intéressait le plus est le fait que les *FiPys* soient programmables par code Python. Cette fonctionnalité nous laissait entrevoir la possibilité d'implémenter et de modifier des mécanismes d'économie d'énergie, voire même nos propres solutions. Depuis, cette possibilité s'est avéré être un mirage, au moins pour ce qui concerne l'économie d'énergie par hibernation.

Le Power Profiler Kit II permet de mesurer les courants électriques avec une précision pouvant aller jusqu'à 100 nA, là où les FiPy opèrent à l'échelle des dizaines de milli-Ampère. Il permet également de récupérer l'évolution du courant électrique sur une période de temps donnée avec une fréquence d'échantillonnage de 100 ksp/s. Le Power Profiler Kit II offre un voltage de 5 V pour les appareils mesurés.



FIGURE 21 – Photographie de la plateforme expérimentale.

#### 4.4.2 Obstacles

Pour mener cette étude expérimentale, le cœur de la difficulté réside dans l'API Python des FiPy. Toutes les fonctionnalités de l'API sont extrêmement faciles d'accès à partir du programme Python qui contrôle la FiPy. Le problème est que l'API des FiPy ne recouvre pas toutes les fonctionnalités de la FiPy. Ces fonctionnalités manquantes sont gérées au niveau du pilote logiciel des FiPy, un code binaire qui est autrement plus complexe à accéder que l'API, et de fait inenvisageable au vu du temps imparti.

En ce qui concerne l'économie d'énergie, l'API des FiPys ne propose qu'un booléen permettant de l'activer ou non. Le pilote logiciel se charge ensuite de choisir le mécanisme d'économie d'énergie et de le configurer – après investigations, il s'avère que les FiPys utilisent le legacy PSM. En particulier, le pilote logiciel gère le passage en hibernation de la carte Wi-Fi, ce qui est un élément clef des mécanismes d'économie d'énergie restreignant le trafic descendant. Nous ne pouvons donc pas espérer implémenter correctement ces mécanismes par nous-mêmes.

Le second obstacle vient de la gestion des trames entrantes et sortantes. Il y a deux manières de les gérer : soit avec le mode promiscuous, soit avec des sockets en mode normal. Lorsque l'appareil est en mode promiscuous, il récupère toutes les trames qui passent sur le médium radio, qu'elles lui soient destinées ou non. Pour gérer les trames entrantes et sortantes, il faut donc pouvoir forger les trames de niveau MAC et filtrer les trames reçues selon leur adresse MAC. En soi, ce n'est pas un problème, mais l'implémentation du pilote logiciel de la FiPy n'est pas robuste en mode promiscuous, et a tendance à très vite mettre les FiPys en erreur. L'autre manière, avec les sockets en mode normal, bénéficie d'une implémentation plus robuste, mais force une gestion des paquets à plus haut niveau. En particulier, les appels système aux fonctions des sockets sont bloquants. Lorsqu'un FiPy cherche à récupérer un paquet reçu, il peut rester bloqué sur l'appel système tant que l'appareil n'en reçoit aucune. Il faut donc jouer sur la durée maximale de blocage pour essayer de minimiser le biais introduit par l'utilisation des sockets.

### 4.4.3 Résultats

Malgré les obstacles mentionnés précédemment, nous avons réussi à implémenter une version dégradée d'un TWT individuel. Nous avons contourné le fait que la STA ne puisse pas passer en hibernation en faisant en sorte qu'à la place, elle chronomètre le temps qu'elle aurait dû passer en hibernation (ce qui est une technique déjà utilisée par [Jang et al., 2011](#)). Pour gérer les trames entrantes et sortantes, nous avons utilisé une socket UDP, avec tous les biais que cela implique, notamment les temps de blocage dus aux appels système. Cependant, à cause du temps imparti, nous n'avons pas pu mener une étude comparative avec les résultats obtenus par WE3S.

## 4.5 CONCLUSION

En conclusion, nous avons exploré les trois méthodes d'évaluation. Les simulations WE3S et notre modèle analytique représentent tout deux la même vision simplifiée de l'économie d'énergie dans le Wi-Fi, à savoir celle décrite par les stratégies d'économie d'énergie. L'évaluation expérimentale représente quant à elle la réalité, c'est-à-dire les mécanismes d'économie d'énergie. Ses résultats sont donc censés différer légèrement par rapport à ceux de WE3S ou de notre modèle analytique. Malheureusement, du fait de la fermeture du code des appareils Wi-Fi testés, nous n'avons pas été capables de mesurer cette différence. Dans la suite de notre étude, nous nous appuyons donc uniquement sur WE3S et sur notre modèle analytique.

| Symbole   | Signification   | Unité            |
|---|---|------------------|
| <i>Paramètres du scénario</i>                     |   |                  |
| $X_{\emptyset}$                                   | Débit maximal atteignable sur le lien non restreint                       | Mbps             |
| $\tau_{\emptyset}$                                | Délai moyen sur les trames du trafic lorsque celui-ci n'est pas restreint | ms               |
| $t_{\text{med}}$                                  | Temps d'accès au médium à l'issue d'une période de restriction            | ms               |
| $\lambda$   | Niveau de charge du trafic considéré                                      | Mbps             |
| $\lambda_{\text{DL/UL}}$                          | Niveau de charge du trafic descendant (DL) ou montant (UL)                | Mbps             |
| $l_{\text{frame}}$                                | Taille moyenne des trames du trafic                                       | bits             |
| $l_{\text{buffer}}$                               | Capacité du buffer de l'appareil source                                   | Nombre de trames |
| $p_{\text{awake}}$                                | Consommation d'énergie d'une STA éveillée                                 | W                |
| $p_{\text{doze}}$                                 | Consommation d'énergie d'une STA en hibernation                           | W                |
| $e_{\text{switch}}$                               | Coût énergétique pour entrer et sortir d'hibernation                      | J                |
| $t_{\text{switch}}$                               | Temps nécessaire pour entrer et sortir d'hibernation                      | s                |
| $t_{\text{prompt}}$                               | Temps nécessaire pour envoyer un prompt                                   | s                |
| $n_{\text{TXOP}}$                                 | Nombre de trames pouvant être envoyées dans une TXOP                      | Nombre de trames |
| $n_{\text{agg}}$                                  | Nombre de trames pouvant être agrégées en une transmission                | Nombre de trames |
| <i>Configuration de la méthode de restriction</i> |   |                  |
| $I_{\text{slot}}$                                 | Durée des périodes de restriction avec la méthode "slot"                  | ms               |
| $D_{\text{slot}}$                                 | Durée des périodes de service avec la méthode "slot"                      | ms               |
| $I_{\text{prompt}}$                               | Durée des périodes de restriction avec la méthode "prompt"                | ms               |
| <i>Métriques de performances en résultat</i>      |   |                  |
| $X$   | Débit atteint par le trafic   | Mbps             |
| $X_{\text{dim}}$                                  | Débit maximal atteignable sur le lien restreint                           | Mbps             |
| $X_{\text{conf}}$                                 | Débit maximal soutenable pendant les périodes de restriction              | Mbps             |
| $\tau$  | Délai moyen sur les trames du trafic lorsque celui-ci est restreint       | ms               |
| $T_{\text{doze}}$                                 | Pourcentage de temps pendant lequel la STA est en hibernation             | %                |
| $F_{\text{doze}}$                                 | Fréquence à laquelle la STA passe d'hibernation à éveil                   | $s^{-1}$         |
| $P$   | Consommation d'énergie de la STA  | W                |

TABLE 3 – Principales notations utilisées dans ce chapitre.

# 5

## ÉVALUATION DE PERFORMANCES ET GUIDE DE CONFIGURATION

Dans ce chapitre, nous évaluons les différentes stratégies d'économie d'énergie. Les stratégies poursuivent des objectifs différents selon qu'elles restreignent le trafic descendant ou le trafic montant. Les stratégies restreignant le trafic descendant permettent aux STAs d'hiberner, et donc d'économiser de l'énergie, en échange d'une dégradation de leurs performances réseaux. Nous étudions le compromis que ces stratégies proposent en section 5.1. Les stratégies restreignant le trafic montant peuvent avoir deux approches. Lorsque le WLAN est saturé, l'AP peut restreindre le trafic montant de toutes les STAs pour leur attribuer des temps de parole différents, et ainsi éviter les collisions et les retransmissions qui représentent une perte sèche d'énergie. Alternativement, une STA qui restreint déjà son trafic descendant peut choisir de restreindre également son trafic montant pour le regrouper avec son trafic descendant, et ainsi rendre son hibernation moins fragmentée. Nous discutons de ces deux approches en section 5.2. Enfin, la section 5.3 propose une solution d'économie d'énergie qui s'adapte automatiquement au trafic.

### 5.1 RESTRICTION DU TRAFIC DESCENDANT

Dans cette section, nous nous intéressons aux stratégies d'économie d'énergie qui restreignent le trafic descendant, avec l'intention de déterminer les règles générales de leur configuration. Pour simplifier nos discussions, nous négligeons pour l'instant le trafic montant des STAs. De plus, nous supposons que les STAs n'ont accès à aucune information concernant la présence ou non de trames en attente dans le buffer de l'AP. Autrement dit, nous ne considérons pas les mécanismes tels que TIM. En effet, les bitmaps TIM sont insérées dans les beacons, qui sont diffusés toutes les 100 ms. Or nous souhaitons considérer des STAs qui récupèrent leurs trames descendantes en attente plus souvent qu'une fois toutes les 100 ms, ce qui est impossible si elles doivent attendre la bitmap TIM. Nous supposons donc que les STAs cherchent à récupérer leurs trames en attente au niveau de l'AP, sans être sûres s'il y en a.

#### 5.1.1 Discussions

Dans la section 4.2 du chapitre 4, nous avons établi un modèle analytique pour décrire les impacts de la configuration des stratégies d'économie d'énergie sur les métriques de performances. Pour rappel, notre modèle analytique repose sur l'hypothèse d'un trafic lisse. Il prend en paramètres d'entrée le débit maximal atteignable sans restriction  $X_{\emptyset}$ , le délai sur les trames lorsque le trafic n'est pas restreint, et le premier temps d'accès au médium au début d'une période de service. En sortie, il donne les impacts des stratégies d'économie d'énergie sur le débit obtenu par le trafic restreint, le délai sur ses trames, et la consommation d'énergie de la STA.

Notre objectif est maintenant de visualiser ces impacts pour en tirer des règles de configuration pour les stratégies d'économie d'énergie (qui seront ensuite précisées en section 5.1.2).

#### *Débit : introduction*

Nous commençons par nous intéresser au débit obtenu par le trafic descendant. Nous considérons que les stratégies d'économie d'énergie ne doivent être la cause

d'aucune perte de paquet car cela serait dommageable pour la STA ; nous regardons donc de plus près ce qui les cause.

Comme établi dans le modèle analytique, il y a deux causes indépendantes de pertes de paquet. La première cause est le **sous-dimensionnement du lien** par rapport à la charge de travail du trafic restreint. Sa principale influence est le débit physique du lien. La seconde cause de pertes de paquet est la **configuration de la durée des périodes de restriction**. Ces influences sont la durée des périodes de restriction, mais aussi la taille du buffer de l'AP. Dans la suite, nous appellerons "durée de restriction" la durée des périodes de restriction. Ces deux causes de pertes de paquet imposent chacune une limite sur le niveau de charge, au-delà de laquelle la STA perdra des paquets. Nous comparons ces deux limites dans différentes situations pour déterminer quelle cause de pertes de paquet limite les performances de la STA.

### **Débit : méthodologie**

Cette section présente la méthodologie que nous avons suivie pour comparer les deux limites sur le débit. Intéressons-nous à la première cause de pertes de paquet, le sous-dimensionnement du lien. Elle impose une limite dure sur le débit de la STA : elle ne peut pas recevoir à un débit supérieur à cette limite, quel que soit le niveau de charge du trafic. Cela correspond au débit que la STA reçoit lorsque son trafic surcharge largement le lien. Nous désignons cette limite comme le débit maximal atteignable par la STA, et nous le notons  $X_{\emptyset}$  (en accord avec la notation utilisée dans notre modèle analytique).

Passons maintenant à la seconde cause de pertes de paquet, la mauvaise configuration de la durée des périodes de restriction. La limite qu'elle définit indique le niveau de charge à partir duquel la STA subira des pertes de paquet. La STA peut cependant obtenir un débit supérieur à cette limite (mais en subissant des pertes de paquet). Nous appelons cette limite le débit maximal soutenable, que nous notons  $X_{\text{conf}}$  (également en accord avec la notation utilisée dans notre modèle analytique).

Pour comparer ces deux limites, nous devons prendre en compte ce qui les influence. Pour le débit maximal atteignable,  $X_{\emptyset}$ , la principale influence est le débit physique du lien. En Wi-Fi, les débits physiques peuvent aller jusqu'à environ 10 Gbps pour la modulation la plus rapide de l'amendement 802.11ax. Nous considérons donc le débit maximal atteignable en fonction du débit physique avec un débit physique variant entre 100 Mbps et 10 Gbps. Pour le débit maximal soutenable,  $X_{\text{conf}}$ , il dépend de la durée des périodes de restriction, et de la taille du buffer de l'AP. Nous explorons le débit maximal soutenable avec des durées de périodes de restriction comprises entre 5 ms et 100 ms, avec un buffer de l'AP pouvant contenir 50, 500 ou 5000 trames.

Pour mesurer le débit maximal atteignable  $X_{\emptyset}$ , nous utilisons WE3S car c'est un paramètre d'entrée de notre modèle analytique, et que WE3S a été validé par ns-3 dans le cas de trafics non restreints. Pour mesurer le débit maximal soutenable  $X_{\text{conf}}$ , nous nous appuyons sur notre modèle analytique. Avec la stratégie "DL slot" (équation 4) comme avec la stratégie "DL prompt" (équation 38), les pertes de paquet dues à la mauvaise configuration de la durée des périodes de restriction interviennent pour le même niveau de charge. Si l'on note  $I$  la durée des périodes de restriction (au lieu de  $I_{\text{slot}}$  pour la stratégie "DL slot" ou  $I_{\text{prompt}}$  pour la stratégie "DL prompt"), alors on a :

$$X_{\text{conf}} = \frac{l_{\text{buffer}} \times l_{\text{frame}}}{I + t_{\text{med}}} \quad (52)$$

La figure 22 compare les valeurs de  $X_{\emptyset}$  et  $X_{\text{conf}}$  pour différentes valeurs de capacité du buffer de l'AP, à savoir 50 trames (figure 22a), 500 trames (figure 22b) et 5000 trames (figure 22c).

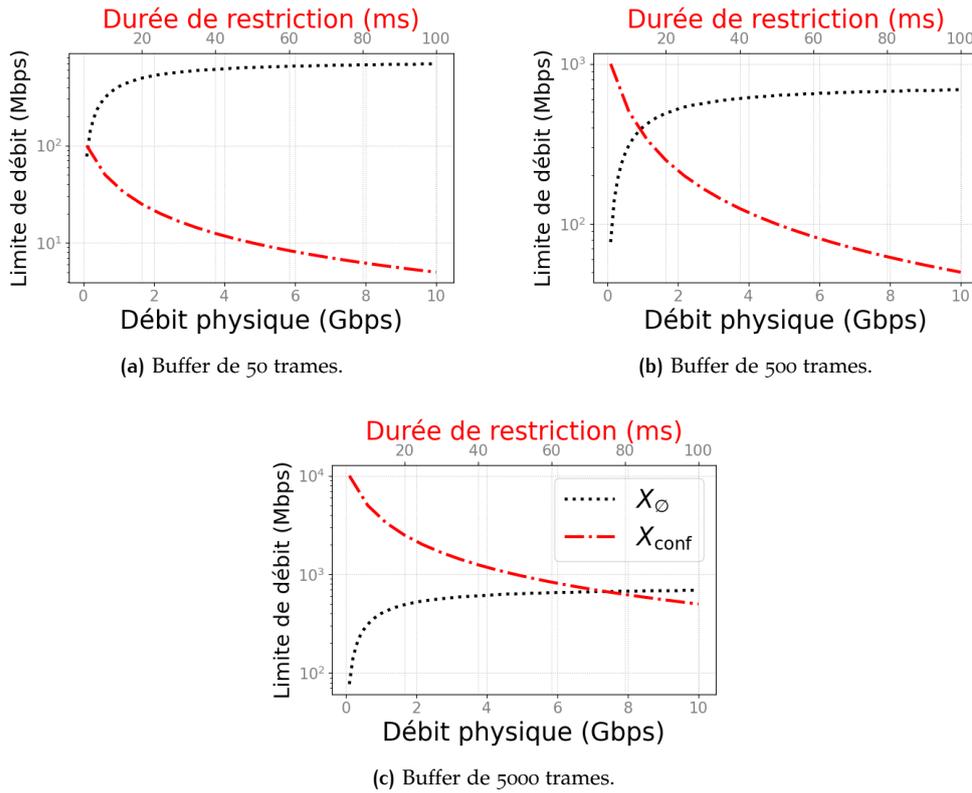


FIGURE 22 – Comparaison du débit maximal atteignable  $X_\emptyset$  (en noir) avec le débit maximal soutenable  $X_{\text{conf}}$  (en rouge).

### Débit : analyse

Le débit maximal atteignable  $X_\emptyset$  augmente avec le débit physique du lien. Il est égal à 77 Mbps pour un débit physique de 100 Mbps, et atteint un plateau à 700 Mbps pour les valeurs de débits physiques les plus hautes. Cela est dû aux surcoûts incompressibles des couches MAC et PHY, qui deviennent les principaux freins aux hauts débits. À noter que cette courbe est strictement identique sur les figures 22a, 22b et 22c (mais l'échelle change).

Le débit maximal soutenable  $X_{\text{conf}}$  diminue lorsque la durée des périodes de restriction augmente. C'est normal puisque plus la STA attend longtemps avant de récupérer ses trames en attente, plus leur nombre augmente (et risque donc de dépasser la capacité du buffer de l'AP). L'évolution du débit maximal soutenable suit une fonction inverse par rapport à la durée de restriction. Le débit maximal soutenable est donc nettement plus élevé pour une durée de période de restriction égale à 5 ms ou 10 ms, que pour des durées plus longues.

Regardons maintenant l'influence de la capacité du buffer de l'AP. En figure 22a, le buffer de l'AP est petit (50 trames). Il ne peut donc pas absorber des débits élevés pendant plusieurs millisecondes sans déborder. De fait, le débit maximal soutenable  $X_{\text{conf}}$  est égal à 100 Mbps pour une durée de restriction de 5 ms, et il descend à 5 Mbps pour une durée de restriction égale à 100 ms. Dans la majorité des cas, le débit maximal atteignable  $X_\emptyset$  est supérieur au débit maximal soutenable  $X_{\text{conf}}$ . La configuration de la durée de restriction est donc cruciale pour éviter les pertes de paquet lorsque le buffer de l'AP est petit.

Au contraire, en figure 22c, le buffer de l'AP est très grand (5000 trames). Alors, le débit maximal soutenable  $X_{\text{conf}}$  est aussi très élevé : entre 500 Mbps pour une durée de restriction de 100 ms, et 10 Gbps pour une durée de restriction de 5 ms. Le débit maximal soutenable  $X_{\text{conf}}$  est donc quasiment tout le temps supérieur au

plafond à 700 Mbps du débit maximal atteignable  $X_{\emptyset}$ . Dans ce cas, la configuration de la durée de restriction importe peu, car le trafic saturera le lien avant de saturer le buffer de l'AP pendant les périodes de restriction.

Enfin, la figure 22b présente le cas où le buffer de l'AP est de taille moyenne (500 trames). Ce cas est moins déterminé que les deux précédents : selon le débit physique du lien, la configuration de la durée de restriction peut avoir une importance ou non.

À notre connaissance, les APs domestiques ont des buffers larges, ce qui nous a également été confirmé par les chercheurs d'Orange dans le cadre du projet ANR FACTO (qui a financé cette thèse). Ainsi, il y a peu de risque de mal configurer la durée de restriction, quel que soit le trafic descendant des STAs. À noter tout de même que le buffer de l'AP est partagé entre toutes les STAs qui lui sont associées. Si plusieurs STAs restreignent leur trafic descendant, elles peuvent occuper une part importante du buffer de l'AP, et le risque de pertes de paquet augmente. Cela peut par exemple être le cas des APs dans les réseaux Wi-Fi d'entreprises ou d'universités.

Il est également à noter que le débit maximal atteignable  $X_{\emptyset}$  que nous avons utilisé dans cette étude peut être atteint seulement lorsque le trafic n'est pas restreint. L'utilisation d'une stratégie d'économie d'énergie restreignant le trafic descendant diminue le débit maximal atteignable, comme le montrent les équations 4 pour la stratégie "DL slot" et 38 pour la stratégie "DL prompt". En particulier, pour la stratégie "DL slot" la configuration de la durée des périodes de service  $D_{\text{slot}}$  influe largement sur le débit maximal atteignable.

### *Délai*

Supposons à présent que la STA ne subit aucune perte de paquet malgré la restriction de son trafic descendant, en accord avec notre objectif défini pour le débit. Désormais, nous pouvons observer l'évolution du délai moyen sur les trames en fonction de la durée des périodes de restriction.

Nous considérons une STA dont le trafic descendant a un débit de 5 Mbps. La STA utilise soit la stratégie "DL slot", soit la stratégie "DL prompt". Dans le cas de la stratégie "DL slot", nous considérons deux configurations : la configuration "Slot 10 %" où les périodes de service occupent 10 % du temps (alors  $D_{\text{slot}} = I_{\text{slot}}/9$ ), et la configuration "Slot 50 %" où les périodes de service occupent la moitié du temps ( $D_{\text{slot}} = I_{\text{slot}}$ ). Nous faisons ensuite varier la durée des périodes de restriction ( $I_{\text{slot}}$  ou  $I_{\text{prompt}}$ ) entre 5 ms et 100 ms. Pour éviter les pertes de paquets, nous supposons que le buffer de l'AP est important, c'est-à-dire fixé à une taille correspondant à 5000 trames.

La figure 23 présente les délai moyen des trames descendantes en fonction de la durée des périodes de restriction. Quelles que soient la stratégie utilisée et sa configuration, le délai moyen des trames augmente de façon linéaire en fonction de la durée de restriction. En effet, les trames générées pendant une période de restriction doivent attendre au moins la fin de celle-ci pour être envoyées. Si la période de restriction dure plus longtemps, alors ces trames subissent un délai plus long.

Pour expliquer les différences de pente entre les stratégies et les configurations, il faut regarder le pourcentage de trames qui sont générées pendant les périodes de service, trames qui subissent un délai faible et quasiment invariant, par rapport à celles générées pendant les périodes de restriction, qui subissent un délai proportionnel à la durée des périodes de restriction. Comme les trames sont supposées être générées de façon uniforme dans le temps, cela revient à regarder le pourcentage de temps occupé par les périodes de service par rapport aux périodes de restriction. Comme les périodes de service occupent un temps plus important avec la stratégie "Slot 50 %", sa pente est plus faible que pour la stratégie "Slot 10 %". La stratégie

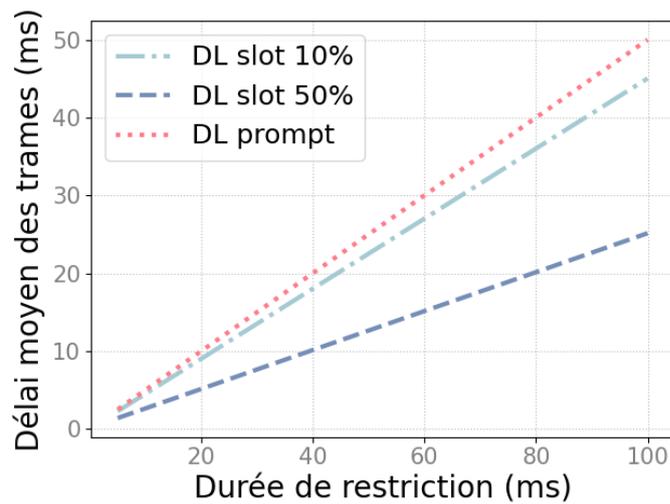


FIGURE 23 – Évolution du délai moyen sur les trames descendantes en fonction de la durée des périodes de restriction.

“DL prompt” ajuste au plus près la durée des périodes de service au nombre de trames en attente, sa pente est donc encore plus élevée que la stratégie “Slot 10 %”.

Dans tous les cas, toute élongation de la durée des périodes de restriction a un impact direct sur le délai moyen des trames descendantes. Du point de vue du délai, il vaut donc mieux garder la durée des périodes de restriction aussi courte que possible.

#### Consommation d'énergie

Intéressons-nous maintenant à la consommation d'énergie de la STA. Nous utilisons le même scénario que celui des délais. La figure 24 présente l'évolution de la consommation d'énergie de la STA en fonction de la durée de ses périodes de restriction.

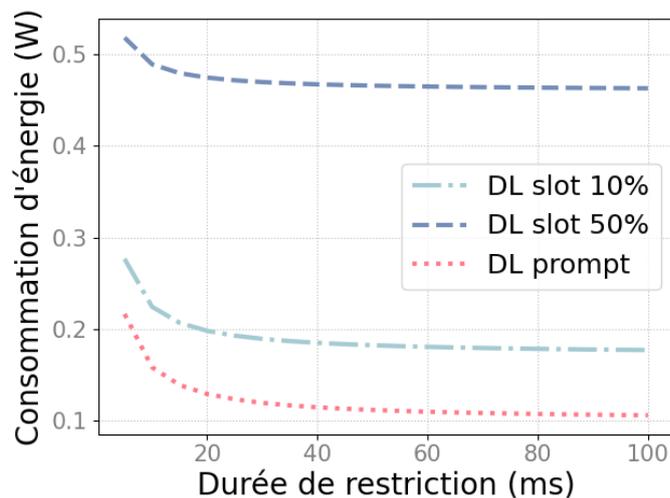


FIGURE 24 – Évolution de la consommation d'énergie de la STA en fonction de la durée des périodes de restriction.

Les courbes présentent toutes la même évolution, celle d'une fonction inverse. Ainsi, les durées de période de restriction les plus courtes produisent les consommations d'énergie les plus élevées, mais dès qu'elles s'allongent un peu, la consommation d'énergie atteint un plateau. Cela s'explique par les changements d'états

entre hibernation et éveil, qui diminuent l'efficacité de l'hibernation. Chaque fois que la STA passe d'hibernation à éveil ou inversement, elle doit payer un coût énergétique. Plus les durées de périodes de restriction sont courtes, plus la STA passe souvent en hibernation, et plus souvent elle paie ce coût énergétique.

La figure 25 représente l'effet d'une hibernation fragmentée par rapport à une hibernation unifiée (c'est-à-dire avec moins d'allers-retours entre hibernation et éveil). Dans les deux cas, la STA passe un temps égal en hibernation. Lorsque la STA a une hibernation fragmentée, elle paie plus souvent le coût énergétique du passage entre hibernation et éveil (représenté en rouge sur la figure), ce qui résulte en une consommation d'énergie moyenne plus élevée.

La stratégie d'économie d'énergie utilisée ajoute ensuite une constante plus ou moins grande aux courbes : la stratégie "DL prompt" mène à la consommation d'énergie la plus basse, suivie de la stratégie "Slot 10 %", puis de la stratégie "Slot 50 %". Cette différence vient du pourcentage de temps passé en hibernation, équivalent au pourcentage de temps passé en périodes de restriction. Comme la stratégie "DL prompt" ajuste la durée des périodes de service au plus près du trafic, elle permet de passer autant de temps que possible en hibernation. La stratégie "Slot 10 %" est un peu moins ajustée et consomme donc un petit peu plus que la stratégie "DL prompt". La stratégie "Slot 50 %" n'est pas du tout ajustée au trafic. Autrement dit, la durée des périodes de restriction est largement supérieure au temps nécessaire pour récupérer les trames accumulées pendant les périodes de restriction. Dans le modèle analytique, cela se caractérise par une valeur de  $D_{\text{slot}}$  largement supérieure à celle de  $t_{\text{cycle}}$ .

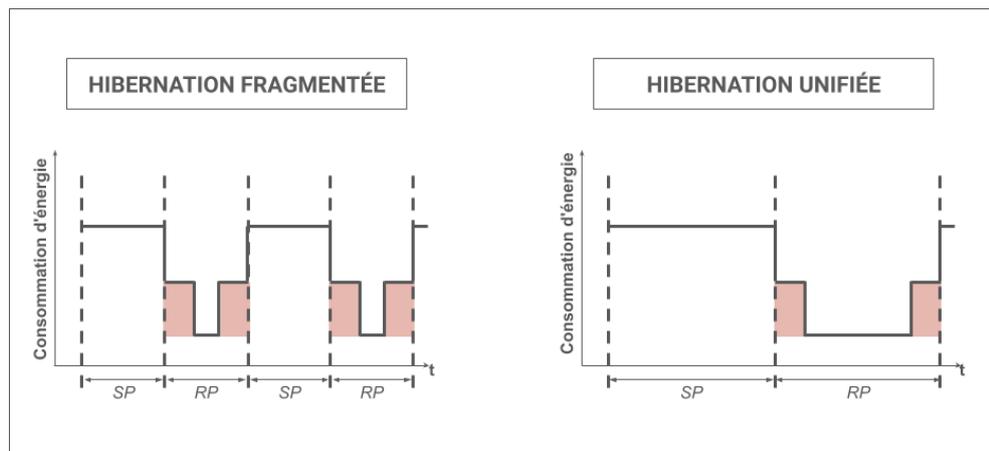


FIGURE 25 – Schéma comparant une hibernation fragmentée à une hibernation unifiée, pour un temps total d'hibernation égal.

SP signifie ici période de service; et RP période de restriction.

Nous en tirons deux conclusions. Premièrement, il faut d'abord chercher à augmenter le temps passé en hibernation car c'est la plus grosse influence sur la consommation d'énergie. La stratégie "DL prompt" le fait par défaut, mais pour la stratégie "DL slot" il faut ajuster la durée des périodes de service au trafic à récupérer, dans la mesure du possible. Deuxièmement, il ne sert à rien d'utiliser des durées de périodes de restriction supérieures à 30 ms, car la consommation d'énergie cesse quasiment de diminuer avec des valeurs plus grandes.

En couplant avec la conclusion liée aux délais, la plage de valeurs intéressantes pour la durée de restriction est entre 5 et 30 ms. Au-delà de 30 ms, on continue de perdre du point de vue du délai sans rien gagner du point de vue de la consommation d'énergie.

### 5.1.2 Étude des paramètres

Dans le chapitre 4, le modèle analytique développé permet de prédire les performances d'une STA qui restreint son trafic. Le but est maintenant d'utiliser ces calculs pour configurer intelligemment les stratégies d'économie d'énergie qui restreignent le trafic descendant.

Nous résumons ici les conclusions obtenues dans la section 5.1.1 et les complétons au besoin avec des formules qui s'appuient sur le modèle analytique présenté dans la section 4.2 du chapitre 4.

#### Guide de configuration

Notre objectif a trois volets. Premièrement, l'utilisation d'une stratégie d'économie d'énergie ne doit pas provoquer de pertes de paquet. Deuxièmement, elle doit permettre de réduire significativement la consommation d'énergie de la STA qui l'utilise. Troisièmement, le délai maximal qu'elle induit doit pouvoir être limité selon les besoins de la STA.

En ce qui concerne le premier volet, l'analyse de la section 5.1.1 nous montre qu'éviter les pertes de paquets est trivial lorsque le buffer de l'AP est de grande taille. Néanmoins, il faut que la configuration de la stratégie "DL slot" respecte les inégalités suivantes :

$$\begin{cases} I_{\text{slot}} \leq \frac{l_{\text{buffer}} \times l_{\text{frame}}}{\lambda_{\text{DL}}} - t_{\text{med}} \\ D_{\text{slot}} \geq \frac{\lambda_{\text{DL}}}{X_{\emptyset} - \lambda_{\text{DL}}} \times I_{\text{slot}} \end{cases} \quad (53)$$

Et pour la stratégie "DL prompt" :

$$I_{\text{prompt}} \leq \frac{l_{\text{buffer}} \times l_{\text{frame}}}{\lambda_{\text{DL}}} - t_{\text{med}} \quad (54)$$

Pour le second volet, nous nous basons exclusivement sur l'analyse de la section 5.1.1. Les durées pertinentes pour les périodes de restriction sont comprises dans la plage de valeurs allant de 5 ms à 30 ms. Par conséquent, la configuration de la stratégie "DL slot" (respectivement de la stratégie "DL prompt") doit respecter l'inégalité suivante :

$$\begin{aligned} I_{\text{slot}} &\in [5 \text{ ms}, 30 \text{ ms}] \quad \text{pour la stratégie "DL slot"} \\ I_{\text{prompt}} &\in [5 \text{ ms}, 30 \text{ ms}] \quad \text{pour la stratégie "DL prompt"} \end{aligned} \quad (55)$$

Enfin, pour le troisième volet, il nous faut définir  $d_{\text{max}}$ , le délai maximal acceptable par la STA. Nous cherchons alors à limiter  $d_0$  qui est le délai moyen maximal subi par les trames, tel que nous l'avons défini dans les équations 13 et 44. Nous pouvons de plus supposer qu'il n'y a pas de perte de paquet pour simplifier les équations. Alors, pour limiter le délai sur les trames à  $d_{\text{max}}$ , il faut que la configuration de la stratégie "DL slot" (respectivement "DL prompt") respecte l'inégalité suivante :

$$\begin{aligned} I_{\text{slot}} &< 2 \times (d_{\text{max}} - t_{\text{med}}) \quad \text{pour la stratégie "DL slot"} \\ I_{\text{prompt}} &< 2 \times (d_{\text{max}} - t_{\text{med}}) \quad \text{pour la stratégie "DL prompt"} \end{aligned} \quad (56)$$

#### Validation

Pour valider nos formules de configuration, nous nous appuyons sur deux scénarios que nous simulons avec WE3S. Le premier scénario teste nos formules de

configuration face à un trafic streaming de forte intensité. Il est donc particulièrement axé sur le respect des débits. Au contraire, le second scénario présente un trafic visio, peu intense mais très sensible au délai. Il nous permet de vérifier que nos formules limitent bien le délai sur les trames en accord avec le délai maximum souhaité.

### Scénario 1 : validation axée sur les débits

Nous considérons un WLAN avec trois STAs : une STA observée et deux STAs concurrentes qui génèrent du trafic de fond. La STA observée a un trafic qui suit une trace de trafic réel, récupéré par tcpdump sur un ordinateur branché en Ethernet. Lors de la mesure, l'ordinateur faisait jouer une vidéo YouTube en très haute qualité (2160p). La trace qui en résulte a un niveau de charge et un coefficient de variation des intervalles inter-trames élevés. La table 4 résume les caractéristiques de la trace de trafic ainsi récupérée. Les STAs concurrentes représentent un trafic de fond total de 8 Mbps généré par processus de Poisson, réparti également entre les STAs et entre trafic montant et descendant. Enfin, nous supposons que les buffers de l'AP et des STAs peuvent contenir 5000 trames. Le reste du scénario est décrit ci-dessous :

- ❖ Taille des trames du trafic de fond : 1250 octets
- ❖ Débit physique de toutes les STAs : 100 Mbps

|                                 | Trafic descendant | Trafic montant |
|---------------------------------|-------------------|----------------|
| Débit moyen                     | 18.7 Mbps         | 18 kbps        |
| Intervalle inter-trames moyen   | 0.5 ms            | 31 ms          |
| CV de l'intervalle inter-trames | 56                | 7.5            |
| Taille moyenne des trames       | 1351 octets       | 70 octets      |

TABLE 4 – Spécifications de la trace “YouTube” utilisée pour le scénario 1. CV signifie ici coefficient de variation.

Passons maintenant à ce que préconisent nos formules de configuration pour ce scénario. Nous supposons pour celui-ci n'avoir aucune contrainte sur les délais des trames (ce qui est généralement le cas avec les services de streaming car ils utilisent la technique du buffering, c'est-à-dire qu'ils envoient leurs paquets largement en avance pour éviter les temps de chargement au milieu des vidéos). Nous n'avons donc que les contraintes sur le débit et sur la consommation d'énergie à respecter. Pour les calculer, nous utilisons la valeur 77 Mbps pour le paramètre  $X_{\emptyset}$ , et la valeur 106  $\mu$ s pour  $t_{med}$ , valeurs que nous avons récupérées grâce à WE3S. D'après nos formules de configuration (équations 53 et 55), la configuration de la stratégie “DL slot” doit respecter les inégalités suivantes :

$$\begin{cases} I_{slot} < \frac{5000 \times 8 \times 1351}{18.7 \times 10^6} - 106 \times 10^{-6} \approx 2.89 \text{ s} \\ D_{slot} \geq \frac{18.7}{77 - 18.7} \times I_{slot} \approx 0.32 \times I_{slot} \\ I_{slot} \in [5 \text{ ms}, 30 \text{ ms}] \end{cases} \quad (57)$$

Et pour la stratégie “DL prompt” (équations 54 et 55) :

$$\begin{cases} I_{prompt} < 2.89 \text{ s} \\ I_{prompt} \in [5 \text{ ms}, 30 \text{ ms}] \end{cases} \quad (58)$$

Cela concorde avec notre analyse à gros grains en section 5.1.1, qui indique que la durée de restriction ne devrait pas avoir beaucoup d'impact sur la perte de paquet, puisque le buffer de l'AP est de grande taille. En bref, nous préconisons une durée

de restriction ( $I_{\text{slot}}$  ou  $I_{\text{prompt}}$ ) égale à 30 ms, et pour la stratégie “DL slot”, une durée des périodes de service égale à au moins 32 % de celle des périodes de restriction.

Pour vérifier la validité de nos formules de configuration, nous prenons un éventail assez large pour la durée de restriction ( $I_{\text{slot}}$  ou  $I_{\text{prompt}}$  selon la stratégie), à savoir entre 5 ms et 100 ms. En parallèle, nous observons pour la stratégie “DL slot” trois valeurs pour le ratio  $D_{\text{slot}}/I_{\text{slot}}$  : 20 %, 33 % et 40 %. Nous indiquons ces différentes configurations avec les notations “DL slot 20 %”, “DL slot 33 %” et “DL slot 40 %”. Pour la comparaison, nous ajoutons également les métriques de performance obtenues par la STA lorsqu’elle n’applique aucune stratégie d’économie d’énergie.

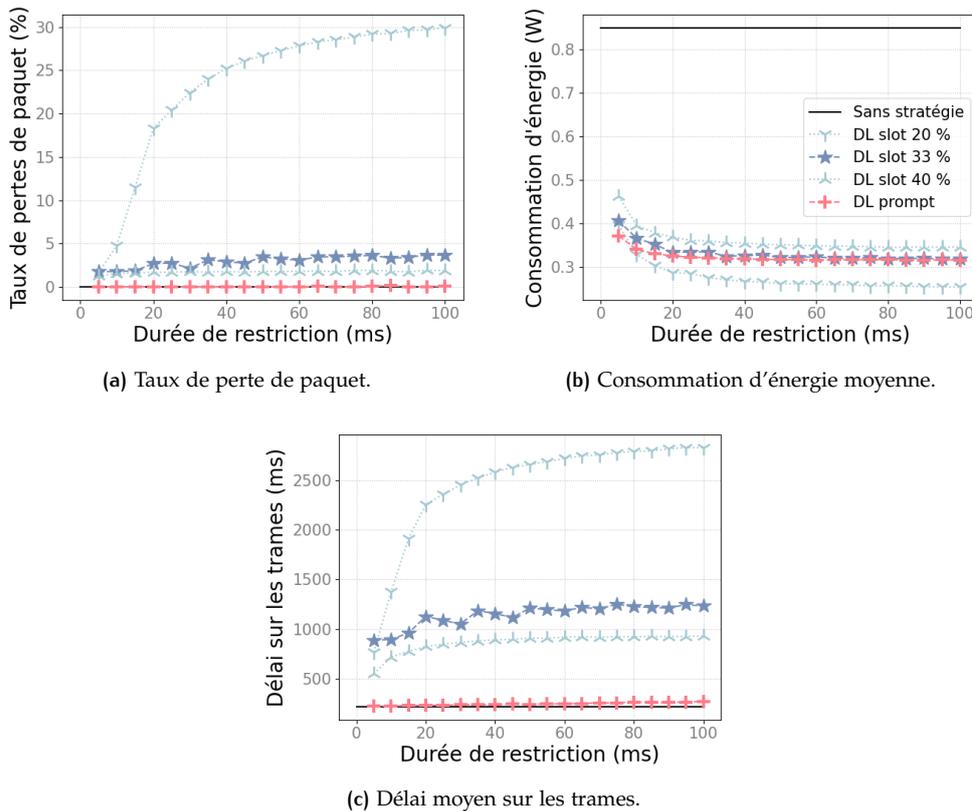


FIGURE 26 – Scénario 1 : métriques de performances obtenues par les différentes configurations des stratégies d’économie d’énergie “DL slot” et “DL prompt”.

La figure 26 présente les métriques de performances obtenues par la STA observée, en fonction de sa stratégie d’économie d’énergie et de sa configuration, pendant une simulation WE3S de 100 s. Au niveau du taux de pertes de paquets, en figure 26a, on observe d’une part que la stratégie “DL prompt” ne subit aucune perte de paquets, quelle que soit la durée des périodes de restriction, ce qui est cohérent avec nos prédictions au vue de la taille du buffer de l’AP. Quant à la stratégie “DL slot”, elle subit des pertes de paquet significatives lorsque ses périodes de service occupent 20 % du temps, mais le taux de pertes de paquet descend en-dessous de 5 % quand les périodes de service occupent 33 % ou 40 %, ce qui est également cohérent avec nos calculs.

Au niveau de la consommation d’énergie, en figure 26b, la stratégie qui consomme le moins est la stratégie “DL slot” à 20 %, mais c’est en partie grâce aux paquets perdus. Les stratégies “DL prompt” et “DL slot” à 33 % consomment un petit peu plus. Enfin la stratégie “DL slot” à 40 % consomme encore un niveau au-dessus. Au niveau de la durée des périodes de restriction, 30 ms est la valeur la plus basse à partir de laquelle la consommation d’énergie ne varie plus significativement.

Enfin, au niveau des délais sur les trames, en figure 26c, la stratégie “DL slot” subit un délai particulièrement important, de l’ordre de la seconde, là où la stratégie “DL prompt” et le cas sans stratégie restent sur des délais de l’ordre de 200 ms. L’ordre de grandeur est totalement différent des délais observés en figure 23 où le trafic descendant était généré par un processus de Poisson. Nous expliquons ces longs délais par la quantité de trafic très importante, et le fait que celui-ci vienne par rafale. La trace utilisée présente des rafales de plusieurs milliers de trames, qui arrivent quasiment toutes au même moment. Sans restriction, cela prend déjà un temps non négligeable pour écouler une telle quantité de données. Cela empire avec la stratégie “DL slot” qui n’adapte pas la durée de ses périodes de service en fonction du trafic : il faut plusieurs périodes de service pour écouler une rafale, ce qui allonge encore les délais.

Au total, si nous prenons en compte à la fois le taux de pertes des paquets et la consommation d’énergie, nous confirmons que la configuration la plus intéressante est la stratégie “DL prompt” avec une durée de périodes de restriction à 30 ms ou la stratégie “DL slot” à 33 % avec une durée de périodes de restriction à 30 ms, parmi le sous-ensemble exploré. La stratégie “DL prompt” reste préférable par rapport à la stratégie “DL slot” au vu des délais que cette dernière induit.

### Scénario 2 : validation axée sur les délais

Pour valider les délais, nous utilisons la même configuration que pour le scénario 1, en changeant le trafic de la STA observée. Cette fois-ci, nous utilisons la trace d’un ordinateur faisant un appel en visio avec l’application Skype. La table 5 résume les caractéristiques de la trace utilisée.

|                                 | Trafic descendant | Trafic montant |
|---------------------------------|-------------------|----------------|
| Débit moyen                     | 172 kbps          | 726 kbps       |
| Intervalle inter-trames moyen   | 13 ms             | 6.4 ms         |
| CV de l’intervalle inter-trames | 1.4               | 1.2            |
| Taille moyenne des trames       | 286 octets        | 590 octets     |

TABLE 5 – Spécifications de la trace “Visio” utilisée pour le scénario 2.  
CV signifie ici coefficient de variation.

Regardons ce que préconisent nos formules de configuration dans ce scénario. Admettons que nous souhaitons un délai moyen inférieur à 10 ms, adapté à une visio. Alors, nos formules de configuration indiquent pour la stratégie “DL slot” (équations 53, 55 et 56) :

$$\begin{cases} I_{\text{slot}} < \frac{5000 \times 8 \times 286}{172 \times 10^3} - 106 \times 10^{-6} \approx 66 \text{ s} \\ D_{\text{slot}} \geq \frac{172 \times 10^3}{77 \times 10^6 - 172 \times 10^3} \times I_{\text{slot}} \approx 0.002 \times I_{\text{slot}} \\ I_{\text{slot}} \in [5 \text{ ms}, 30 \text{ ms}] \\ I_{\text{slot}} < 2 \times (10 \times 10^{-3} - 106 \times 10^{-6}) \approx 19.788 \text{ ms} \end{cases} \quad (59)$$

Et pour la stratégie “DL prompt” (équations 54, 55 et 56) :

$$\begin{cases} I_{\text{prompt}} < 66 \text{ s} \\ I_{\text{prompt}} \in [5 \text{ ms}, 30 \text{ ms}] \\ I_{\text{prompt}} < 19.788 \text{ ms} \end{cases} \quad (60)$$

Autrement dit, le niveau de charge est suffisamment faible pour que le trafic soit insignifiant en termes d’occupation des ressources. Nous n’avons donc aucun risque

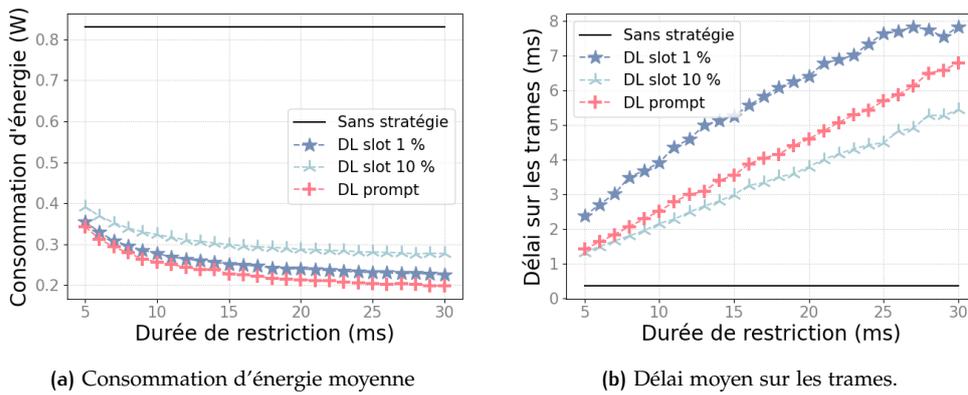


FIGURE 27 – Scénario 2 : métriques de performances obtenues par la STA observée en fonction de la stratégie d'économie d'énergie utilisée et de sa configuration. Aucune de des configurations explorées ne subit de pertes de paquet.

de pertes de paquet, quelle que soit la stratégie d'économie d'énergie utilisée et sa configuration. Nous recommandons donc une durée de restriction ( $I_{\text{slot}}$  ou  $I_{\text{prompt}}$ ) proche de 20 ms, et pour la stratégie "DL slot", un ratio  $D_{\text{slot}}/I_{\text{slot}}$  faible.

Pour vérifier nos préconisations, nous explorons une plage de valeurs assez restreinte pour la durée de restriction, entre 5 ms et 30 ms. Pour la stratégie "DL slot", nous regardons les valeurs de 1 % et 10 % pour le ratio  $D_{\text{slot}}/I_{\text{slot}}$ . Enfin, pour la comparaison, nous ajoutons le cas où la STA observée n'applique aucune stratégie d'économie d'énergie.

La figure 27 présente les métriques de performances obtenues pour différentes valeurs de durée de période de restriction, pour les stratégies "DL slot 1 %", "DL slot 10 %" et "DL prompt", avec en figure 27a la consommation d'énergie moyenne de la STA observée, et en figure 27b le délai moyen constaté sur les trames de la STA observée. Ces résultats ont été obtenus par une simulation WE3S de 100 s. À noter que nous n'incluons pas la figure du taux de pertes de paquet car aucune des configurations explorées ne subit de pertes de paquet.

Toutes les configurations explorées respectent la contrainte d'un délai moyen inférieur à 10 ms. Notre formule de configuration pour les délais est donc plus stricte que nécessaire, puisqu'un certain nombre de solutions qui ne respectent pas l'inégalité de l'équation 56 respectent tout de même la contrainte sur les délais. Cela s'explique par la construction de notre formule de configuration. Nous avons cherché à limiter le délai maximal  $d_0$  que nous rencontrons dans notre modèle analytique, à savoir le délai sur les trames du premier cycle. Or la figure 27b observe le délai moyen des trames. Il est donc normal que le délai moyen soit inférieur à ce délai que nous limitons, et par conséquent, que notre formule de configuration soit stricte. Pour être plus exacts dans notre analyse, nous devrions calculer le pourcentage de trames qui subissent un délai inférieur à  $d_{\text{max}}$ , ce que ne permet pas WE3S en l'état.

## 5.2 RESTRICTION DU TRAFIC MONTANT

Dans cette section, nous nous intéressons à la restriction du trafic montant. Il y a deux approches différentes pour l'utiliser.

La première approche vise à réduire la consommation d'énergie du WLAN en évitant les retransmissions – coûteuses en énergie – dues aux collisions. L'AP utilise alors des mécanismes d'économie d'énergie (RAW par exemple) pour restreindre le trafic montant des STAs qui lui sont associées et ainsi distribuer les ressources plus strictement qu'avec le seul CSMA/CA. Cette approche n'est pertinente que

lorsque le risque de collision est non négligeable dans le WLAN. Nous explorons cette approche dans la section 5.2.1.

Nous explorons une seconde approche qui vise à regrouper les transmissions montantes et descendantes d'une STA sur une même période de temps pour que son hibernation soit moins fragmentée. Cette approche ne peut s'appliquer que si la STA a son trafic descendant déjà restreint. Certains mécanismes d'économie d'énergie adoptent cette approche, par exemple TWT. Elle ne repose cependant pas nécessairement sur un mécanisme du standard IEEE de manière explicite : la STA peut décaler son trafic montant sans le signaler à l'AP. Pour éviter les ambiguïtés, nous parlons alors de décalage de trafic et non de restriction, puisqu'elle n'est pas obligatoire. Cette approche est abordée en section 5.2.2.

### 5.2.1 Restriction par l'AP

Dans cette section, nous supposons que les STAs sont nombreuses dans le WLAN. Leur trafic est important, et dans le cas de leurs trafics descendants, non restreints. Le risque de collisions est donc élevé si leurs trafics montants ne sont pas restreints non plus. La question est alors la suivante : l'AP doit-il restreindre le trafic montant des STAs pour réduire le risque de collisions ? Cela permet ainsi de ne pas gaspiller les ressources radio en évitant les retransmissions, coûteuses pour tous les appareils. Il est à noter que les STAs ne peuvent pas hiberner dans cette situation, puisque nous supposons que leur trafic descendant est laissé libre ; le gain d'énergie ne peut donc se faire que sur le nombre de retransmissions.

#### *Scénario*

Nous considérons un WLAN avec 14 STAs identiques. Nous faisons varier la charge de travail totale du WLAN entre 10 et 80 Mbps, que nous distribuons uniformément entre les trafics descendant et montant des STAs. Pour chacune de ces valeurs, nous observons le débit total atteint par le WLAN, ainsi que la consommation d'énergie moyenne des STAs et de l'AP. Plus précisément, les variables du WLAN sont les suivantes :

- ❖ Nombre de STAs : 14
- ❖ Taille des trames : 1250 octets
- ❖ Débit PHY : 100 Mbps
- ❖ Capacité des buffers de l'AP et des STAs : 500 trames
- ❖ Trafics générés par des processus de Poisson

Nous comparons les cas où les STAs n'utilisent aucune stratégie d'économie d'énergie avec les cas où les STAs utilisent la stratégie "UL slot" ou "UL prompt". Pour configurer les stratégies, nous procédons comme suit. Pour la stratégie "UL slot", nous divisons le temps en 15 créneaux égaux, et nous attribuons chacun des 14 premiers à une STA. Le dernier créneau est laissé libre pour que l'AP puisse aussi envoyer des trames sans risque de collision. Le temps total des 15 créneaux est alors le seul paramètre d'entrée, nous en explorons les résultats avec les valeurs de 30 ms et 120 ms. Pour la stratégie "UL prompt", l'AP envoie des trames prompt aux STAs en *round robin*, à intervalle régulier. Un intervalle entre deux trames prompt pour une même STA est égal soit à 30 ms, soit à 120 ms dans notre expérience.

#### *Résultats*

Nous utilisons WE3S pour simuler le scénario décrits en section 5.2.1 pendant 10 s. La figure 28 présente le taux de pertes de paquet moyenné sur la totalité des trafics, et la consommation d'énergie moyenne des appareils (STAs et AP) en fonction de la charge de travail totale du WLAN.

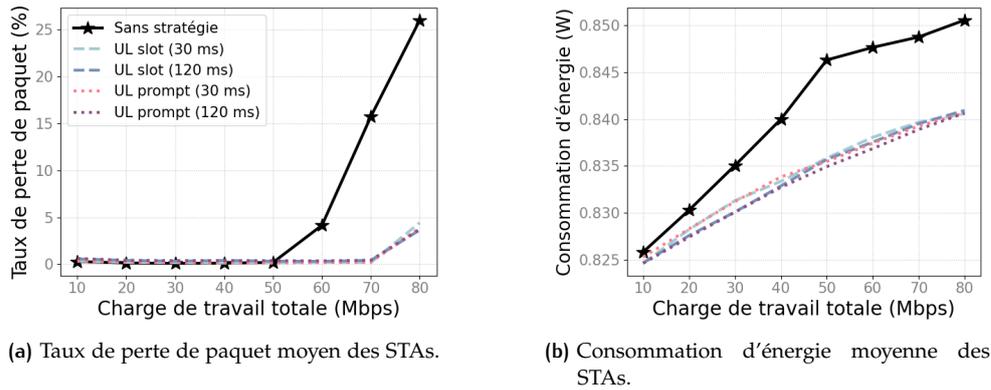


FIGURE 28 – Évaluation de l'impact de la restriction des trafics montants par l'AP.

Commençons par nous intéresser à l'impact de la restriction des trafics montants sur le taux de pertes de paquet des trafics, présenté en figure 28a. Les données importantes sur cette figure sont les niveaux de charge qui marquent le début des pertes de paquet. Sans restriction des trafics montants, ce niveau de charge est égal à 50 Mbps. Avec une restriction, quelle que soit la stratégie utilisée et sa configuration, ce niveau de charge est déplacé à 70 Mbps. Restreindre les trafics montants atteint donc son premier objectif, à savoir de permettre une meilleure utilisation des ressources du médium en évitant les collisions.

Tournons-nous maintenant vers la consommation d'énergie des appareils, en figure 28b. De manière générale, les appareils consomment plus quand leur charge de travail augmente, ce qui est normal puisqu'ils doivent passer plus de temps dans les états réception ou transmission, coûteux en énergie. Les évolutions sont cependant différentes selon que les trafics montants soient restreints ou non. Lorsque les trafics montants ne sont pas restreints, la consommation d'énergie augmente plus vite avec la charge de travail que lorsque les trafics montants sont restreints. Cela est dû aux collisions. Cette tendance est plus limitée sur les charges de travail très élevées car le nombre de paquets perdus (liés aux débordements de buffer) permet aux appareils de passer un peu moins de temps en mode réception ou transmission. Il est cependant important de noter l'ordre de grandeur des consommations d'énergie enregistrées : elles varient de 0.825 W à 0.85 W, ce qui fait une variation de 3 %. Cette variation est négligeable, et ainsi restreindre les trafics montants ne permet pas de réduire significativement la consommation d'énergie des appareils.

On peut aussi noter, sur la figure 28, que les stratégies utilisées et leur configuration ont peu d'impact visible sur les pertes de paquet et sur la consommation d'énergie. Nous l'expliquons par les buffers de grande taille utilisés à la fois par les STAs et par l'AP : ils sont suffisamment grands pour absorber le trafic pendant les périodes de restriction. Les pertes de paquet interviennent quand les ressources allouées deviennent insuffisantes par rapport à la quantité de trafic.

## Conclusion

Les stratégies d'économie d'énergie qui restreignent le trafic montant permettent donc bien de réduire les collisions dans le WLAN, mais l'impact sur la consommation d'énergie des appareils est négligeable. Nous recommandons donc d'utiliser de telles stratégies que dans une perspectives de performances réseaux, et uniquement si le risque de collisions est important. Dans les autres cas, elles font peser des délais supplémentaires sur les trafics montants sans qu'il n'y ait de contreparties intéressantes.

### 5.2.2 Décalage du trafic par la STA

Dans cette section, nous nous intéressons à une STA en particulier dont le trafic descendant est restreint, peu importe la méthode. Cette STA a donc le droit d'hiberner pendant ses périodes de restriction. La question est la suivante : est-il pertinent pour la STA de décaler son propre trafic montant afin de le regrouper avec son trafic descendant ?

Pour rappel, la restriction du trafic descendant fonctionne avec des périodes de service pendant lesquelles le trafic descendant est autorisé, et le reste du temps constitue les périodes de restriction pendant lesquelles le trafic descendant est interdit, ce qui permet à la STA d'hiberner sans manquer de transmission. Si la STA ne décale pas son trafic montant, elle doit se réveiller pendant ses périodes de restriction chaque fois qu'une trame montante est générée. En décalant son trafic montant, la STA n'a à se réveiller qu'une seule fois, et son hibernation serait moins fragmentée. La STA subirait alors moins souvent le coût énergétique du passage entre hibernation et éveil. Le désavantage est que restreindre le trafic montant induirait un délai additionnel sur ses trames montantes, alors que ses trames descendantes subissent déjà le délai de la stratégie d'économie d'énergie. La figure 29 schématise l'effet du décalage du trafic montant.

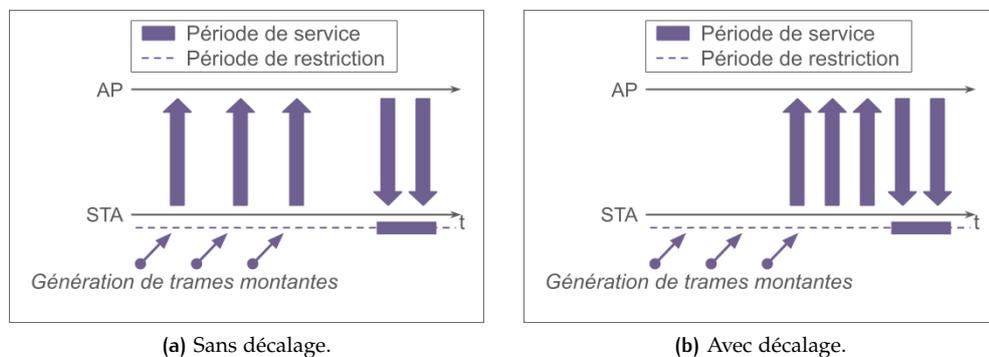


FIGURE 29 – Schémas comparant l'écoulement du trafic montant avec ou sans décalage.

Cette approche existe dans les standards IEEE mais peut également ne pas reposer sur un mécanisme. Si la STA utilise un mécanisme du standard, alors l'AP est au courant, et la restriction du trafic montant se fait de manière moins flexible. Si la STA décide par elle-même de décaler son trafic, l'AP n'en est pas au courant, mais cela ne gêne pas son fonctionnement puisque la STA continue malgré tout d'accéder au médium en utilisant le CSMA/CA.

#### Description de la solution

L'algorithme 1 détaille notre solution de décalage de trafic. Pour une trame montante qui vient d'être générée, il choisit le moment où la STA essaiera de l'envoyer. Si la trame est générée pendant une période de service, la STA essaie immédiatement de l'envoyer via le CSMA/CA. Sinon, quand la trame est générée pendant une période de restriction, elle regarde le temps qu'il reste jusqu'à la prochaine période de service prévue : si ce temps est plus court que le décalage autorisé, la STA attend juste avant la période de service pour envoyer la trame montante, sinon, elle cherche à l'envoyer immédiatement. (Dans tous les cas, elle doit ensuite respecter le CSMA/CA).

Notre solution de décalage de trafic prend en compte deux paramètres de configuration : le `decalage_autorise` et la `temporisation`. Le paramètre `decalage_autorise` indique le sursis maximal que peuvent subir les trames montantes. En d'autres termes, nous n'autorisons la STA à décaler l'émission d'une trame montante seulement si la prochaine période de service est suffisamment proche. Pour

le paramètre de temporisation, la STA n'envoie pas ses trames montantes décalées exactement au début de ses périodes de service mais un petit peu avant, afin de ne pas perturber le trafic descendant. Le paramètre temporisation est la durée qui sépare l'émission des trames montantes décalées du début de la période de service.

En décalant l'émission des trames montantes, la STA évite des passages entre éveil et hibernation, qui représentent chacun un coût énergétique de  $e_{\text{switch}}$ . De plus, cela permet de mieux profiter de l'agrégation de trames et des TXOPs : la STA emmagasine des trames et les envoie toutes en une fois, en mutualisant l'accès au médium et les en-têtes. Par conséquent, leur transmission nécessite moins de temps, et la STA peut passer plus de temps en hibernation.

#### Algorithme 1 – Décalage du trafic montant

---

##### PARAMETRES :

decalage\_autorise  
temporisation

##### ENTREES :

creation\_trame : horodatage de la creation de la trame montante  
consideree  
prochaine\_SP : horodatage du debut prevu de la prochaine periode de  
service

##### SORTIE :

envoi\_trame : horodatage du moment ou la STA commencera a essayer d'  
envoyer la trame montante consideree

##### FONCTION :

**si** creation\_trame est dans une periode de service, **alors** :  
    envoi\_trame = creation\_trame  
**sinon, si** prochaine\_SP - creation\_trame > decalage\_autorise, **alors** :  
    envoi\_trame = creation\_trame  
**sinon** :  
    envoi\_trame = max (creation\_trame, prochaine\_SP - temporisation)  
**Retourner** envoi\_trame

---

### Scénario

Pour tester les effets du décalage du trafic montant, nous utilisons un scénario avec un AP, une STA observée et deux STAs concurrentes. La STA observée a un trafic descendant de 10 Mbps, et un trafic montant dont le débit est égal soit à 10 Mbps pour le premier scénario "chargé", soit à 1 Mbps pour le second scénario "léger". Dans les deux scénarios, les trafics montant et descendant sont générés par des processus de Poisson. Les STAs concurrentes produisent du trafic de fond à hauteur de 2 Mbps, réparti de manière égale entre le trafic montant et descendant et entre STAs.

La STA observée utilise la stratégie "DL prompt" pour restreindre son trafic descendant. Pour ne pas surcharger les courbes, nous ne considérons pas la stratégie "DL slot", mais elles donnent des résultats similaires. Nous considérons des durées de périodes de restriction allant de 5 ms à 100 ms. Comme paramètres du décalage, nous explorons les valeurs 20 ms, 40 ms et 80 ms pour le paramètre decalage\_autorise, et nous mettons le paramètre temporisation à 1 ms. Nous implémentons les scénarios "chargé" et "léger" sur WE3S et les simulons pendant 30 s pour évaluer les métriques de performances obtenues par la STA observée.

## Résultats

La figure 30 présente les résultats obtenus par des simulations WE3S de 30 s implémentant les scénarios décrits précédemment. Les figures 30a, 30c et 30e représentent les résultats du scénario “léger” respectivement en termes de consommation d’énergie, de délai moyen sur les trames montantes et descendantes, et de délai moyen sur les trames montantes seulement. De la même façon, les figures 30b, 30d et 30f sont les résultats du scénario “chargé” respectivement en termes de consommation d’énergie, de délai moyen sur les trames montantes et descendantes, et de délai moyen sur les trames montantes seulement.

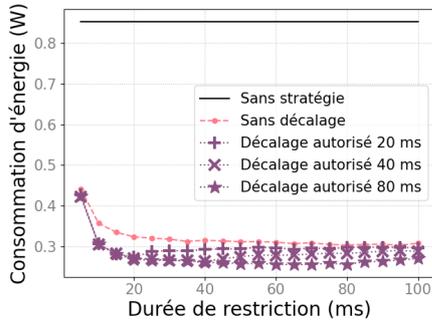
La première chose à noter sur ces courbes, qui n’apparaît pas sur ces figures, est que la STA observée ne perd aucun paquet, et ce quelles que soient son éventuelle stratégie d’économie d’énergie et sa configuration.

De manière générale, décaler le trafic montant a bien l’effet escompté. Une STA qui restreint son trafic descendant et décale son trafic montant a une consommation d’énergie plus basse et ses trames subissent un délai plus élevé en moyenne par rapport à une STA qui ne fait que restreindre son trafic descendant ; et plus encore par rapport à une STA qui n’utilise aucune stratégie d’économie d’énergie. Toujours de manière générale, nous retrouvons également les évolutions déjà observées en section 5.1.1. Lorsque la STA observée restreint son trafic descendant (avec ou sans décalage), sa consommation d’énergie suit une fonction inverse par rapport à la durée de restriction du trafic descendant, et le délai moyen sur ses trames suit une fonction affine avec une ordonnée à l’origine quasiment nulle.

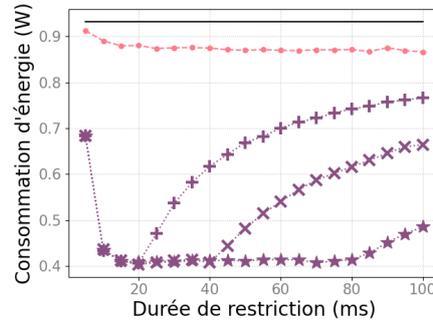
Regardons maintenant plus précisément l’évolution des courbes correspondant aux cas où la STA observée décale son trafic montant. Que ce soit pour la consommation d’énergie ou pour le délai moyen des trames, les courbes présentent toutes un point de rupture plus ou moins visible, au moment où la durée de restriction dépasse le décalage autorisé. Lorsque la durée de restriction est inférieure au décalage autorisé, la consommation d’énergie de la STA observée suit une fonction inverse par rapport à la durée de restriction. Cette fonction inverse a une constante moins élevée que celle obtenue dans le cas où la STA ne décale pas son trafic montant. Le délai moyen sur les trames suit une fonction affine avec une pente plus élevée que lorsque la STA observée ne décale pas son trafic montant. Lorsque la durée de restriction dépasse le décalage autorisé, les courbes quittent les fonctions précédemment décrites pour rejoindre les courbes qui représentent le cas où la STA restreint son trafic descendant sans décaler son trafic montant.

Nous expliquons ce phénomène de la manière suivante. Lorsque la durée de restriction est inférieure au décalage autorisé, la STA peut décaler toutes ses trames montantes. En revanche, lorsque la durée de restriction est supérieure au décalage autorisé, la STA ne peut plus décaler les trames montantes qui sont générées au début des périodes de restriction du trafic descendant, puisqu’elles sont trop éloignées de la prochaine période de service. Alors, si l’on augmente encore la durée de restriction sans changer le décalage autorisé, la proportion des trames montantes qui ne sont pas décalées augmente également, et les performances de la STA observée se rapprochent d’autant de celles observées quand elle ne décale pas son trafic montant du tout. La figure 31 illustre ce phénomène.

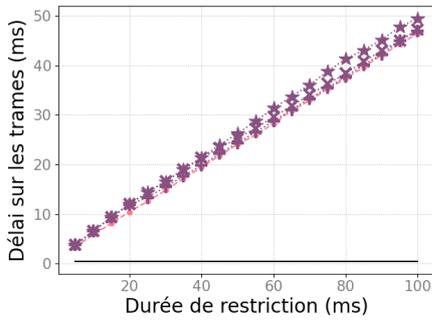
Passons maintenant à la comparaison entre le scénario “léger” et le scénario “chargé”. La différence la plus marquante se situe au niveau de la consommation d’énergie, lorsque la STA restreint son trafic descendant sans décaler son trafic montant. Alors, si son trafic montant est important (ce qui est le cas dans le scénario “chargé”), la STA observée consomme quasiment autant d’énergie que le cas où elle n’utilise aucune stratégie d’économie d’énergie. Au contraire, lorsque le trafic montant de la STA observée est léger (comme dans le scénario “léger”), alors la STA observée parvient à économiser significativement de l’énergie en restreignant son trafic descendant sans avoir besoin de décaler son trafic montant. En comparaison, le niveau de charge du trafic ascendant a beaucoup moins d’influence sur les cas



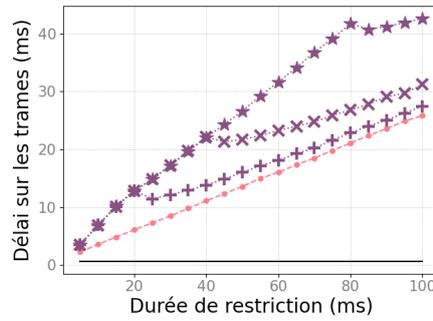
(a) Scénario "léger" : consommation d'énergie de la STA observée.



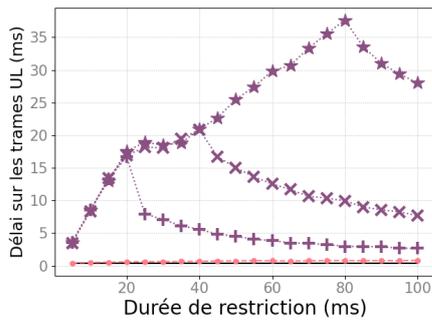
(b) Scénario "chargé" : consommation d'énergie de la STA observée.



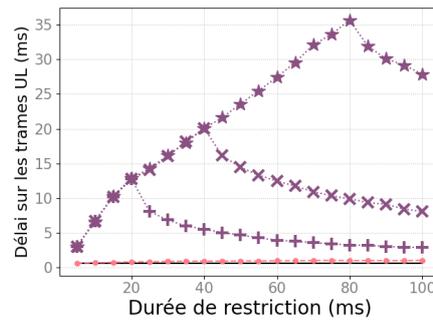
(c) Scénario "léger" : délai moyen sur toutes les trames de la STA observée.



(d) Scénario "chargé" : délai moyen sur toutes les trames de la STA observée.



(e) Scénario "léger" : délai moyen sur les trames montantes de la STA observée.



(f) Scénario "chargé" : délai moyen sur les trames montantes de la STA observée.

FIGURE 30 – Consommation d'énergie et délai moyen sur les trames de la STA observée en fonction de la durée de restriction.

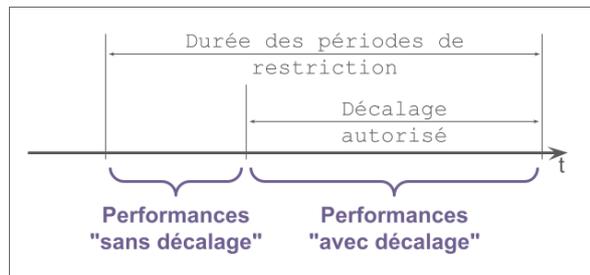


FIGURE 31 – Schéma représentant l'impact de la valeur du décalage autorisé sur les performances.

où la STA observée restreint son trafic descendant et décale son trafic montant, avec un décalage autorisé supérieur à la durée de restriction.

Notre explication est la suivante. Dans le scénario "chargé", le trafic montant de la STA est suffisamment intense pour qu'elle ne puisse pas hiberner entre la génération de deux trames montantes consécutives, et ce, même pendant les périodes de restriction du trafic descendant. La STA observée n'hiberne donc jamais, à moins de décaler son trafic montant. Dans le cas du scénario "léger", les trames montantes sont suffisamment espacées pour que la STA observée puisse hiberner même sans décaler son trafic montant. Le gain apporté par le décalage de trafic est alors plus modeste.

Plus généralement, si la STA ne décale pas son trafic montant, alors la durée de ses hibernations est limitée à la fois par la durée de restriction et par l'intervalle de temps entre la génération de deux trames montantes consécutives. Si elle décale son trafic montant, son hibernation n'est plus limitée que par la durée de restriction, qui est configurable.

### Discussions

Notre solution de décalage du trafic montant, simple à mettre en œuvre, propose un nouveau compromis entre consommation d'énergie et performances réseau en plus de celui déjà apporté par la stratégie restreignant le trafic descendant. Notre solution améliore l'économie d'énergie apportée par la restriction du trafic descendant, sans effet sur le débit, mais ajoute un délai sur les trames du trafic montant qui jusque là étaient laissées libres. Le gain sur l'économie d'énergie devient particulièrement important lorsque le trafic montant de la STA est intense au point d'empêcher la STA d'hiberner, mais il devient plus modeste lorsque le trafic montant est léger. En revanche, l'impact du décalage sur le délai des trames montantes reste identique, quelle que soit la charge de travail du trafic montant.

Nos recommandations sont donc les suivantes : si le trafic montant de la STA est intense (c'est-à-dire avec un intervalle inter-trames inférieur à 5 ms), il faut soit restreindre le trafic descendant avec une stratégie d'économie d'énergie et décaler le trafic montant, soit ne restreindre aucun des trafics de la STA. En effet, si la STA ne restreint que le trafic descendant, alors elle subira le délai additionnel sur les trames descendantes et le risque de pertes de paquet sans économiser d'énergie pour autant. Si le trafic montant de la STA est léger (avec un intervalle inter-trames supérieur à 5 ms), alors restreindre seulement le trafic descendant propose déjà un compromis acceptable. Décaler le trafic montant de la STA propose alors un gain additionnel marginal, au prix d'un délai sur les trames montantes, ce qui peut être envisageable si l'appareil a besoin de consommer aussi peu d'énergie que possible.

## 5.3 CONFIGURATION AUTOMATIQUE DE LA RESTRICTION DES TRAFICS

Pour cette section, nous nous intéressons aux stratégies restreignant le trafic descendant dans les cas où le buffer de l'AP est de petite taille. Les ressources du buffer de l'AP sont donc limitées. Les STAs qui restreignent leur trafic descendant sont celles qui ont le plus besoin de cette ressource, car leurs trames descendantes y sont stockées pendant les périodes de restriction. Plus les périodes de restriction d'une STA sont longues, plus celle-ci utilise les ressources du buffer de l'AP. De la même manière, plus le niveau de charge du trafic descendant de la STA est élevé, plus celle-ci utilise les ressources du buffer de l'AP. Comme nous considérons n'avoir pas la main sur le niveau de charge du trafic descendant de la STA, il faut adapter la durée de restriction utilisée par la STA selon le niveau de charge du trafic descendant de la STA pour qu'elle n'utilise pas trop de ressources du buffer de l'AP, et ainsi éviter les pertes de paquet. Le guide de configuration de la section 5.1.2 nous donne des inégalités pour un trafic supposé lisse, ce qui sous-entend un niveau de charge qui ne varie pas significativement dans le temps. Or ce n'est pas toujours le cas avec du trafic réel. L'éventail des durées de restriction pertinentes varie alors dans le temps, avec le niveau de charge du trafic.

Dans cette section, nous présentons une solution simple à mettre en œuvre qui adapte automatiquement la durée de restriction utilisée par la STA selon le niveau de charge de son trafic descendant. De plus, notre solution présente les avantages d'être transparente pour l'utilisateur et de ne nécessiter aucune connaissance préalable sur le trafic descendant de la STA.

### 5.3.1 Description de la solution

Notre solution repose sur la stratégie "DL prompt" seulement, nous écartons la stratégie "DL slot". Il y a deux raisons à ce choix. La première est que la modification de la durée de restriction à la volée est plus naturelle avec la stratégie "DL prompt" qu'avec la stratégie "DL slot". En effet, les périodes de service avec la stratégie "DL prompt" commencent quand la STA envoie un prompt, ce qui est forcément toujours un peu variable; là où la STA et l'AP doivent s'accorder en amont sur le début des périodes de service avec la stratégie "DL slot". La seconde raison est que la stratégie "DL slot" nécessite aussi de définir la durée des périodes de service, ce que fait automatiquement la stratégie "DL prompt". Pour pouvoir nous concentrer sur le problème lié à la durée de restriction, nous avons décidé de ne considérer que la stratégie "DL prompt".

L'objectif de notre solution est d'utiliser une quantité des ressources du buffer de l'AP quasiment constante sur chaque période de restriction, définie par un paramètre que nous nommons `objectif_trames`. Autrement dit, la STA va choisir la durée de chaque période de restriction de manière à ce que le nombre de trames à récupérer à l'issue de la période de restriction soit égal à `objectif_trames`. Cela permet deux choses : d'une part, la STA ne monopolise pas toutes les ressources du buffer de l'AP pour elle-même (tant que la valeur du paramètre `objectif_trames` est choisie intelligemment). D'autre part, la STA va tout de même chercher à allonger ses durées de restriction quand le niveau de charge de son trafic descendant le permet.

Notre solution permet à une STA qui atteint la fin d'une période de service, de choisir la durée de la période de restriction qui commence. Elle dispose de deux informations pour prendre sa décision : la durée de la période de restriction précédente, et le nombre de trames descendantes qu'elle a récupérées pendant la période de service courante.

Notre solution suit un algorithme de type AIMD (*Additive Increase Multiplicative Decrease*). Le nombre de trames descendantes récupérées pendant la période de

service représente la quantité de ressources du buffer de l'AP utilisées par la STA. Si pendant sa période de service, la STA récupère moins de trames que `objectif_trames`, c'est qu'elle n'utilise pas autant de ressources du buffer de l'AP qu'elle le pourrait, alors elle augmente prudemment la durée de sa prochaine période de restriction par rapport à sa précédente. Si au contraire elle récupère plus de trames, alors la STA utilise trop de ressources du buffer de l'AP, et elle réduit rapidement sa durée de restriction. Nous ajoutons un second palier, que nous appelons `max_trames`. Si la STA récupère plus de trames que `max_trames`, cela veut dire qu'elle utilise largement trop de ressources du buffer de l'AP, alors elle réduit la prochaine durée de restriction à sa valeur minimale. L'algorithme 2 présente plus en détails le fonctionnement de notre solution AIMD.

### 5.3.2 Étude des performances de la solution

Algorithme 2 – Solution AIMD

---

**PARAMETRES :**

`restriction_min` : durée de restriction minimale.  
`restriction_max` : durée de restriction maximale.  
`pas_incrementation` : pas avec lequel la STA incremente sa durée de restriction.  
`objectif_trames` : nombre de trames que la STA doit idéalement récupérer pendant une période de service.  
`max_trames` : nombre de trames maximal que la STA doit récupérer pendant une période de service.

**ENTREES :**

`restriction_precedente` : durée de la précédente période de restriction.  
`retour_trames` : nombre de trames récupérées lors de la période de service qui vient de s'écouler.

**SORTIE :**

`prochaine_restriction` : durée de la prochaine période de restriction

**FONCTION :**

**Si** `retour_trames` < `objectif_trames`, **alors** :  
`prochaine_restriction` = `restriction_precedente` + `pas_incrementation`  
**Sinon, si** `retour_trames` ≥ `objectif_trames` **et** `retour_trames` < `max_trames`, **alors** :  
`prochaine_restriction` = `restriction_precedente` / 2  
**Sinon, si** `retour_trames` ≥ `max_trames`, **alors** :  
`prochaine_restriction` = `restriction_min`  
`prochaine_restriction` = max(`prochaine_restriction`, `restriction_min`)  
`prochaine_restriction` = min(`prochaine_restriction`, `restriction_max`)  
**Retourner** `prochaine_restriction`

---

Pour évaluer l'efficacité de notre solution AIMD, nous la testons dans deux scénarios : d'abord dans un scénario avec du trafic synthétique pour observer ses performances dans un environnement contrôlé, puis dans un second scénario avec traces de trafic réel, pour la tester en conditions aussi réelles que possible. Nous la comparons avec des configurations statiques de la stratégie "DL prompt", pour lesquelles

la valeur de la durées des périodes de restriction est constante dans le temps. Nous explorons les valeurs suivantes pour les durées de restriction des configurations statiques : 5 ms, 10 ms, 15 ms, 20 ms, 25 ms et 30 ms.

Nous utilisons les valeurs suivantes pour les paramètres de notre solution AIMD :

- ❖ `restriction_min` : 5 ms
- ❖ `restriction_max` : 30 ms
- ❖ `pas_incrementation` : 1 ms
- ❖ `objectif_trames` : 32 trames
- ❖ `max_trames` : 50 trames

Nous mettons la valeur de `restriction_min` à 5 ms car avec les valeurs que nous utilisons, c'est le temps minimum pour que la STA puisse entrer en hibernation (cf. section 3.1.4). Pour `restriction_max`, nous nous basons sur les conclusions du chapitre 5.1, indiquant qu'il n'y a pas un grand intérêt à avoir une période de restriction plus longue que 30 ms. Le paramètre `pas_incrementation` est choisi arbitrairement : il faut qu'il soit assez petit pour bien échantillonner la plage des durées de périodes de restriction, sans trop l'être au risque de ralentir notre solution AIMD. Nous mettons `objectif_trames` à 32 trames car c'est le nombre de trames récupérables en une TXOP dans les conditions des scénarios. Enfin, `max_trames` a également été choisi de manière arbitraire : il suffit qu'il soit compris entre 32 (la valeur de `objectif_trames`) et 100 (la capacité du buffer).

Pour être complet, nous ajoutons les résultats de la STA lorsqu'elle implémente notre solution AIMD couplée avec le décalage de son trafic montant, présenté dans la section 5.2.2 du chapitre 5.2. Nous utilisons alors les valeurs suivantes pour les paramètres du décalage :

- ❖ `decalage_autorise` : 30 ms
- ❖ `temporisation` : 1 ms

Ces valeurs nous assurent que le décalage se fait au maximum en interférant peu avec la solution AIMD.

### *Scénario "synthétique"*

Nous commençons par explorer le potentiel de notre solution AIMD dans un environnement contrôlé. Nous utilisons donc du trafic synthétique, généré par des processus de Poisson.

Notre scénario consiste en un WLAN avec un AP et 3 STAs associées, dont une STA observée. La STA observée a un trafic descendant dont l'intensité varie dans le temps. Plus précisément, nous découpons le temps en segment de 10 s. Sur chaque segment, le trafic descendant de la STA est généré par un processus de Poisson avec une intensité différente de celles utilisées pour les autres segments. La table 6 décrit le trafic descendant de la STA observée avec son niveau de charge, son intervalle inter-trames, et l'intervalle inter-prompt qu'est censé viser notre solution AIMD (calculé en multipliant l'intervalle inter-trames par 32, la valeur du paramètre `objectif_trames`). Le trafic montant de la STA reste constant sur l'ensemble des segments, à hauteur de 1 Mbps. La STA observée implémente soit aucune stratégie d'économie d'énergie, soit la stratégie "DL prompt" avec une configuration de  $I_{\text{prompt}}$  statique, soit cette même stratégie mais configurée avec notre solution AIMD, avec puis sans décalage.

Les STAs concurrentes produisent du trafic de fond à hauteur de 8 Mbps réparti également à la fois entre STAs et entre trafic montant et descendant. Elles n'implémentent aucune stratégie d'économie d'énergie.

Le reste des paramètres du scénario est décrit ci-dessous :

- ❖ Nombre total de STAs : 3
- ❖ Taille des trames : 1250 octets

| Segment            | Niveau de charge | Intervalle inter-trames | $I_{\text{prompt}}$ visé |
|--------------------|------------------|-------------------------|--------------------------|
| 1 (de 0 s à 10 s)  | 100 Mbps         | 100 $\mu$ s             | 5 ms                     |
| 2 (de 10 s à 20 s) | 5 Mbps           | 2 ms                    | 30 ms                    |
| 3 (de 20 s à 30 s) | 50 Mbps          | 200 $\mu$ s             | 6.4 ms                   |
| 4 (de 30 s à 40 s) | 20 Mbps          | 500 $\mu$ s             | 16 ms                    |

TABLE 6 – Description du trafic descendant de la STA observée.

- ❖ Débit physique : 1 Gbps
- ❖ Capacité des buffers de l’AP et de la STA : 100 trames

Pour illustrer le fonctionnement de notre solution AIMD, la figure 32 montre l’évolution de  $I_{\text{prompt}}$  dans le temps, en fonction du trafic descendant de la STA observée. Dans le premier segment (de 0 à 10 s), le trafic descendant est très intense, aussi notre solution AIMD reste sur la valeur minimale, à savoir 5 ms. Lors du second segment (de 10 à 20 s), la situation est inversée : le trafic descendant est très léger, et notre solution AIMD progresse rapidement pour atteindre et rester sur la valeur maximale, à savoir 30 ms. Sur les deux derniers segments (de 20 à 40 s), notre solution AIMD se stabilise autour de la valeur qu’elle vise, en variant plus ou moins significativement autour.

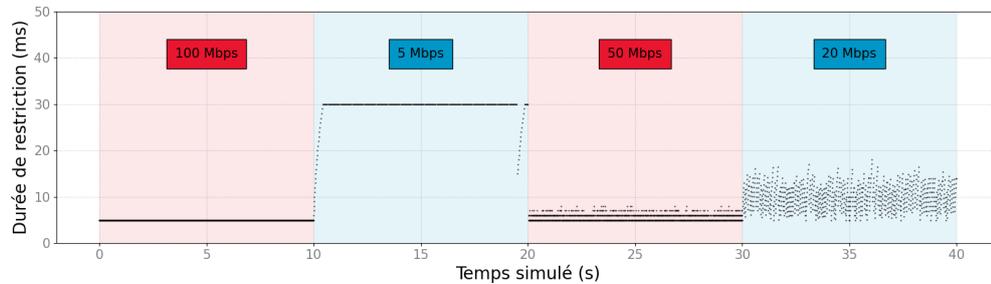


FIGURE 32 – Visualisation de l’algorithme de la solution AIMD.

La figure 33 compare le taux de pertes de paquets et la consommation d’énergie de la STA observée selon la stratégie d’économie d’énergie utilisée et sa configuration. Pour les configurations statiques, la valeur de  $I_{\text{prompt}}$  utilisée est indiquée juste au-dessus de chaque point, en millisecondes.

Lorsqu’elle n’utilise pas de stratégie d’économie d’énergie, la STA observée ne subit aucune perte de paquet, mais sa consommation d’énergie s’élève à 85 W.

Regardons maintenant lorsqu’elle utilise la stratégie “DL prompt”. Les configurations statiques pour lesquelles  $I_{\text{prompt}}$  est supérieur à 5 ms subissent toutes des pertes de paquet, à partir de 2 % pour la configuration à 10 ms. Seule la configuration statique avec laquelle  $I_{\text{prompt}}$  vaut 5 ms, et notre solution AIMD parviennent à éviter les pertes de paquets. En ce qui concerne la consommation d’énergie, la STA observée consomme 0.23 W avec notre solution AIMD, contre 0.26 W avec la configuration statique où  $I_{\text{prompt}}$  vaut 5 ms. Notre solution AIMD obtient donc la consommation d’énergie minimale parmi les configurations qui évitent les pertes de paquet. Les autres configurations statiques, celles qui subissent des pertes de paquet, ont une consommation d’énergie encore inférieure, comprise entre 0.16 W et 0.22 W. Ce dernier point s’explique justement par les pertes de paquet : la STA passe moins de temps dans l’état réception car elle a moins de trames descendantes à récupérer, et peut donc retourner plus vite en hibernation pour économiser de l’énergie.

Comme notre objectif premier est d’éviter les pertes de paquets tout en minimisant la consommation d’énergie, notre solution AIMD est la plus satisfaisante des solutions explorées dans ce scénario.

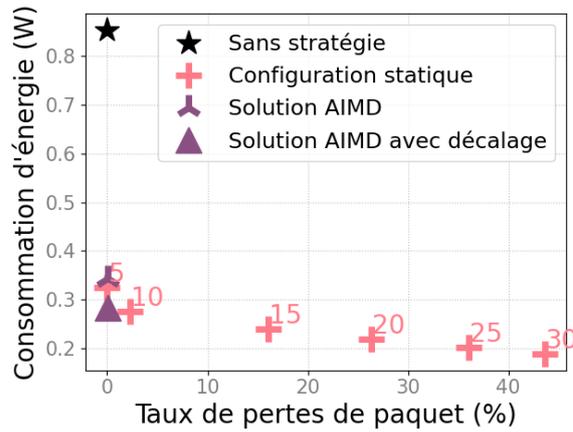


FIGURE 33 – Performances en termes de taux de pertes de paquet et de consommation d'énergie obtenues par la solution AIMD comparées à celles obtenues par des configurations statiques. Les chiffres au-dessus des configurations statiques indiquent la valeur de  $I_{\text{prompt}}$  utilisée (en ms).

### Scénario "réel"

Le scénario "synthétique" utilise du trafic généré par des processus de Poisson, ce qui ne représente pas forcément tous les trafics réels. En effet, les processus de Poisson génèrent un trafic très lisse (leur coefficient de variation est égal à 1), là où certains trafics réels sont sujets à des rafales de paquets, ce qui se traduit par des coefficients de variation plus grands. Pour rendre plus complète l'étude de notre solution AIMD, nous la testons face à des traces de trafic réel.

Nous considérons un WLAN avec quatre STAs en tout : deux STAs observées dont nous mesurons les métriques de performance, et deux STAs concurrentes qui génèrent du trafic de fond.

|   | Trace "Visio" |            | Trace "Streaming" |           |
|---|---------------|------------|-------------------|-----------|
|   | DL            | UL         | DL                | UL        |
| Débit moyen   | 172 kbps      | 726 kbps   | 129 kbps          | 1.5 kbps  |
| Intervalle inter-trames moyen                         | 13.2 ms       | 6.5 ms     | 76.7 ms           | 493 ms    |
| Coefficient de variation de l'intervalle inter-trames | 1.4           | 1.2        | 12.6              | 5.1       |
| Taille moyenne des trames                             | 286 octets    | 590 octets | 1240 octets       | 92 octets |

TABLE 7 – Caractéristiques des traces de trafic utilisées.

Les STAs observées ont un trafic qui suit des traces de trafic préalablement récupérées, de la même manière que les traces utilisées en section 5.1.2. Nous utilisons deux types de trace : une qui représente un appel en visio (que l'on appellera "Visio"), que l'on a déjà utilisé en section 5.1.2, et une autre trace qui représente du streaming de vidéo (que l'on appellera "Streaming"), qui est cette fois différente de celle utilisée en section 5.1.2. Les traces "Visio" ont été récupérées pendant un appel avec l'application Skype, et les traces "Streaming" pendant le visionnage d'une vidéo YouTube avec une qualité aussi basse que possible, c'est-à-dire à 144p. Les deux STAs observées ont une trace différente mais équivalentes, qui ont été récupérées l'une à la suite de l'autre. Les caractéristiques des traces sont données (ou rappelées) en table 7. Les STAs observées implémentent soit aucune stratégie d'économie

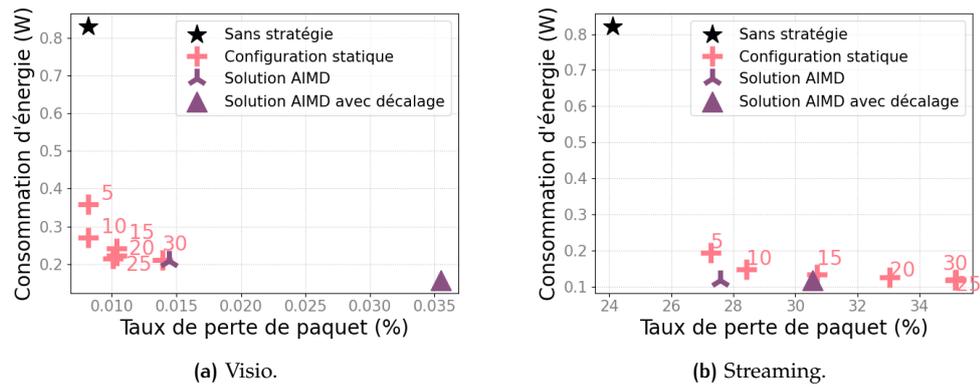


FIGURE 34 – Performances en termes de pertes de paquet et de consommation d'énergie de notre solution AIMD, comparée à des configurations statiques de la stratégie "DL prompt".

d'énergie, soit la stratégie "DL prompt" avec une configuration statique, soit la stratégie "DL prompt" configurée avec notre solution AIMD, avec ou sans décalage.

Les STAs concurrentes produisent du trafic de fond via un processus de Poisson pour un trafic total de 10 Mbps réparti également entre les STAs et le trafic montant et le trafic descendant. Ces STAs ne restreignent aucun de leur trafic.

La figure 34 compare les performances obtenues par notre solution AIMD (avec ou sans décalage) face à celles obtenues avec des configurations statiques de la stratégie "DL prompt", dans des simulations WE3S de 100 s.

Avec du trafic "Visio" (en figure 34a), toutes les configurations explorées parviennent à soutenir le trafic descendant sans perte de paquet significative. En effet, le taux maximal de perte de paquet est de 0.035 %, ce qui est largement négligeable pour cette application. Au niveau de la consommation d'énergie, notre solution AIMD obtient un résultat similaire à celui de la configuration statique à 30 ms, à savoir 0.2 W. C'est la consommation d'énergie minimale parmi toutes les configurations statiques. En activant le décalage du trafic montant, la STA peut encore gagner 0.05 W pour une consommation d'énergie finale de 0.15 W. Notre solution AIMD atteint donc bien ses objectifs face au trafic "Visio" : elle ne provoque pas de perte significative de paquets tout en réduisant au maximum la consommation d'énergie de la STA.

Passons maintenant au trafic "streaming", en figure 34b. Les résultats sont ici largement différents : toutes les configurations subissent des pertes de paquet, avec des taux compris entre 27 % et 36 %. Il est important de souligner que même sans stratégie d'économie d'énergie, la STA perd 24 % de ses paquets, ce qui est déjà trop élevé. Nous expliquons ce taux de pertes de paquet élevé par la nature du trafic de la trace "Streaming". Celui-ci arrive majoritairement par violentes rafales de trafic. Dans le scénario présent, le buffer de l'AP est incapable d'absorber ne serait-ce qu'une seule de ces rafales, les pertes de paquet interviennent avant même que l'AP n'ait eu une chance d'envoyer une trame à la STA.

Pour compléter notre étude à propos de ce cas particulier, nous avons rejoué ce scénario avec cette fois-ci un buffer de l'AP pouvant contenir 300 trames, ce qui est suffisant pour absorber une rafale de trafic. Le résultat est présenté en figure 35.

Sans stratégie d'économie d'énergie, les STAs perdent 5 % de leurs paquets, et entre 7 et 12 % si les STAs utilisent une stratégie d'économie d'énergie avec nos configurations explorées. En combinant ces résultats avec ceux de la figure 34b, nous concluons que les stratégies d'économie d'énergie ne sont pas la cause principale de pertes de paquet dans ce scénario. Les rafales de trafic de la trace "Streaming" sont très importantes, mais espacées dans le temps. Ainsi, à partir du moment où l'AP parvient à absorber une rafale de trafic, la STA a suffisamment de temps pour récupérer ses trames descendantes en attente avant la prochaine rafale de trafic,

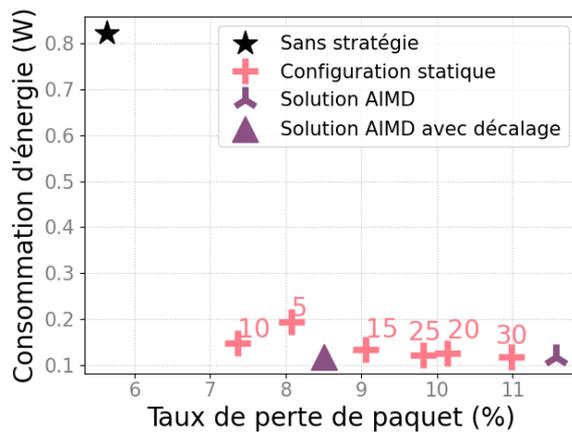


FIGURE 35 – Taux de pertes de paquet et consommation d'énergie des STAs observées face à la trace de trafic "Streaming" et avec un buffer de l'AP de 300 trames.

et ce que son trafic descendant soit restreint ou non. Pour le dire autrement, les rafales de trafic posent un risque de pertes de paquet en amont de toute restriction de trafic, qui ne peut être évité qu'avec un buffer de l'AP plus grand. Une fois cette condition remplie, les configurations statiques obtiennent des résultats équivalents à notre solution AIMD, avec ou sans décalage.

### 5.3.3 Discussions

Le scénario "synthétique" montre notre solution AIMD sous son meilleur jour, ce qui permet de prouver les gains qu'elle peut apporter face à des configurations statiques. Pourtant, nous n'avons pas été capables de reproduire ces conditions avec des traces de trafic réel. Dans le cas de la trace "Visio", le trafic est trop léger pour risquer des pertes de paquet, quelle que soit la durée de restriction. Dans le cas de la trace "Streaming", le trafic est plus lourd, mais provoque des pertes de paquet en amont des éventuelles restrictions de trafic.

Plus généralement, le domaine d'applicabilité de notre solution AIMD paraît relativement limité. En premier lieu, notre solution AIMD n'est pertinente que lorsque le buffer de l'AP est de petite taille, ce qui ne semble pas être la norme pour les appareils réels, d'après nos informations. Ensuite, elle ne fonctionne qu'avec un trafic descendant relativement lisse, qui ne varie que sur des intervalles de temps relativement longs, de l'ordre de la seconde. Les trafics réels ne sont pas tous comme cela.

## 5.4 CONCLUSION

Dans ce chapitre, nous avons exploré les différentes stratégies d'économie d'énergie et leur domaine d'applicabilité. Nous les résumons ci-dessous.

- ❖ L'AP doit restreindre le trafic montant de toutes les STAs seulement si le risque de collisions est particulièrement élevé dans le WLAN. Les stratégies restreignant le trafic montant des STAs s'apparentent alors beaucoup à du TDMA, et leur configuration peut suivre les algorithmes classiques pour le partage du temps.
- ❖ Une STA peut restreindre son trafic descendant pour hiberner si son trafic descendant ne sature pas le lien radio et s'il peut supporter au moins quelques millisecondes de délai additionnel sur ses trames. Dans ce cas, plusieurs autres questions se posent :

- Si le buffer de l'AP est de grande taille (plus de 1000 trames), une bonne valeur pour la durée de restriction est 30 ms, à moduler avec le délai additionnel que peut supporter le trafic descendant.
- Si le buffer de l'AP est de petite ou moyenne taille (moins de 1000 trames), alors il faut peut-être envisager d'adapter la durée de restriction au niveau de charge du trafic descendant de la STA. Il est à rappeler cependant que sur les trafics réels que nous avons testés, aucun ne nécessitait réellement une telle adaptation. Dans tous les cas, la plage de valeurs intéressante pour la durée de restriction se situe entre 5 ms et 30 ms.
- Pour la stratégie "DL slot", il faut prendre garde à ce que les périodes de service soient suffisantes pour écouler le trafic descendant, tout en restant relativement ajustée à celui-ci pour améliorer l'économie d'énergie.
- Quelle que soit la taille du buffer de l'AP, la STA a intérêt à décaler son trafic montant pour le regrouper avec son trafic descendant s'il est trop important. Au niveau de la configuration, il vaut mieux autoriser le décalage jusqu'à 30 ms, mais cette valeur est encore une fois à moduler selon le délai additionnel que peut supporter le trafic montant.

# 6

## CONCLUSION

### 6.1 RÉSUMÉ DE THÈSE

Cette thèse s'intéresse à l'économie d'énergie des STAs dans la couche MAC du Wi-Fi. Malgré la prédominance des questions environnementales, c'est un sujet qui est resté relativement vierge de recherches académiques, la plupart des études existantes s'attachant à résoudre des problèmes très localisés. Dès lors, notre principal objectif fut de dégager des perspectives dans ce sujet, afin de comprendre à la fois les entraves à l'économie d'énergie en Wi-Fi, et les bénéfices que celle-ci peut apporter.

Notre première contribution, présentée en chapitre 3, est une synthèse des mécanismes d'économie d'énergie présents dans les standards IEEE 802.11. À partir de ces mécanismes, nous avons dérivé des abstractions qui modélisent les grandes familles de mécanismes d'économie d'énergie : les stratégies d'économie d'énergie. Plus précisément, les mécanismes d'économie d'énergie reposent sur la restriction soit du trafic montant, soit du trafic descendant, soit des deux, et ces trafics peuvent être restreints par deux méthodes, implicite (slot) ou explicite (prompt). Les stratégies consistent donc à préciser quel type de trafic est restreint, et avec quelle méthode. En passant par les stratégies, nous simplifions notre étude en nous concentrant sur les principaux points communs entre les mécanismes d'économie d'énergie et ainsi en limitant le nombre de facteurs qui influencent les performances finales.

Notre seconde contribution est d'avoir développé les outils permettant d'évaluer les impacts des stratégies d'économie d'énergie sur un ensemble de métriques de performance, ce qui est exposé dans le chapitre 4. Nous avons commencé par concevoir et coder un simulateur Wi-Fi dédié à l'étude des stratégies d'économie d'énergie que nous avons baptisé WE<sub>3</sub>S. Il nous a permis d'obtenir des premiers résultats, qui à leur tour nous ont fait apercevoir les liens entre paramètres, configuration et performances. Nous avons ensuite pu exprimer ces liens dans des modèles analytiques, ce qui met en lumière les influences exactes de la configuration des stratégies d'économie d'énergie sur les métriques de performance.

Enfin, notre dernière contribution est d'avoir tiré de notre étude de performance un guide d'utilisation, à la fois pour les stratégies restreignant le trafic descendant et pour les stratégies restreignant le trafic montant en chapitre 5.

### 6.2 TRAVAUX FUTURS

Nous avons identifié trois axes sur lesquels nous pourrions améliorer notre étude, que nous présentons ci-dessous.

#### 6.2.1 Élargir les stratégies d'économie d'énergie

Les stratégies d'économie d'énergie que nous étudions ne représentent pas tous les aspects des mécanismes d'économie d'énergie. Maintenant que nous maîtrisons bien ces stratégies, nous pourrions les réviser pour y inclure certaines autres fonctionnalités des mécanismes d'économie d'énergie et voir comment ceux-ci influencent les métriques de performance.

Un exemple serait d'intégrer l'OFDMA dans nos stratégies d'économie d'énergie, en suivant l'exemple des mécanismes trigger-enabled TWT ou broadcast TWT. Par rapport au CSMA/CA classique que nous avons considéré, l'OFDMA est censé utiliser les ressources du médium de manière plus efficace, et donc de réduire la consommation d'énergie totale du réseau. Quoique les réponses seront sûrement différentes, les questions restent les mêmes : dans quelles situations l'OFDMA est-il avantageux pour les STAs en mode économie d'énergie, et alors, avec quelles configurations ?

Un autre aspect intéressant serait d'intégrer des fonctionnalités propres aux STAs IoT pour élargir le panel de scénarios. Par exemple, les STAs IoT sont généralement capables de tirer parti des longues périodes d'hibernation – de l'ordre de plusieurs secondes – en entrant en hibernation profonde, c'est-à-dire que l'appareil entier est éteint à l'exception de son horloge.

Dans tous les cas, la marche à suivre est la même : il faut identifier les fonctionnalités intéressantes, les intégrer dans un simulateur et analyser les résultats. La difficulté de cet axe dépend largement des fonctionnalités considérées : implémenter l'OFDMA sera certainement plus difficile que d'implémenter l'hibernation profonde.

### 6.2.2 Rapprocher les métriques de performance de l'utilisateur

Notre étude s'appuie sur des métriques de performance de premier ordre : débit moyen, délai moyen, et consommation d'énergie moyenne. L'avantage de ces métriques de performance est qu'elles sont relativement simples à récupérer d'abord, et à analyser ensuite.

Cependant, ces métriques de performance ne représentent pas toujours correctement l'expérience vécue par l'utilisateur (QoS vs. QoE). Or c'est bel et bien l'expérience utilisateur qui décidera si les mécanismes d'économie d'énergie seront utilisés ou non. En effet, un utilisateur est susceptible d'utiliser un mécanisme d'économie d'énergie s'il ne remarque pas ou peu de différence dans sa navigation internet. Pour voir si le compromis entre sacrifice des performances réseau et économie d'énergie est viable, il vaudrait mieux considérer des métriques de performance comme le temps de chargement au début d'une vidéo en streaming, ou de la durée de vie d'une batterie de smartphone en train de faire défiler un fil d'actualité.

Ces métriques de performance sont intrinsèquement plus difficiles à simuler du fait de la multitude de facteurs qui rentrent en jeu. La première étape serait d'implémenter au moins une stratégie d'économie d'énergie sur une carte Wi-Fi pour pouvoir observer grandeur nature les impacts de l'économie d'énergie sur l'expérience utilisateur. Il est à noter que cette première étape est déjà loin d'être triviale comme nous avons pu le constater avec notre expérience discutée dans la section 4.4 : la configuration des mécanismes d'économie d'énergie existants se fait dans le pilote logiciel des cartes Wi-Fi et non dans l'API ; modifier ladite configuration nécessite donc de plonger dans un code écrit en assembleur.

### 6.2.3 Réduire les délais additionnels

Notre étude nous permet de configurer les stratégies d'économie d'énergie pour éviter les pertes de paquet avec de bonnes garanties, mais pour les délais sur les trames, elle se contente de les limiter à un niveau tolérable pour les applications. Imaginons qu'il soit possible de prédire exactement les arrivées de trames descendantes dans le buffer de l'AP. Dans ce cas, nous pourrions construire une solution avec laquelle l'économie d'énergie n'aurait aucun impact sur les performances réseaux (et donc pour l'utilisateur aussi) : la STA récupérerait ses trames descendantes en attente dès qu'elles arriveraient dans le buffer de l'AP, et n'hibernerait que lorsque son trafic descendant est nul. Il est probablement impossible de prédire

le trafic de manière aussi précise, donc cet axe nécessite de prendre en compte le degré d'incertitude des prédictions.

Cet axe de travail nécessite d'abord de construire un oracle, en sachant qu'il n'a pas accès aux informations du niveau applicatif, seulement celles du niveau MAC. Il faut ensuite trouver une configuration des stratégies d'économie d'énergie qui permette de tirer parti des prédictions de l'oracle tout en prenant en compte son degré d'incertitude. Enfin, la dernière exigence est la frugalité : le code de l'oracle ne doit pas consommer plus d'énergie qu'il ne permet d'en économiser.



## BIBLIOGRAPHIE

- Anastasi, Giuseppe, Marco Conti, Enrico Gregori et Andrea Passarella
- 2004a "A performance study of power-saving policies for Wi-Fi hotspots", *Computer Networks*, 45, 3, p. 295-318.
  - 2004b "Saving energy in Wi-Fi hotspots through 802.11 PSM : an analytical model", in *2nd Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'04)*, p. 227-236.
  - 2008 "802.11 power-saving mode for mobile computing in Wi-Fi hotspots : limitations, enhancements and open issues", *Wireless Networks*, 14, p. 745-768.
- Bianchi, Giuseppe
- 2000 "Performance analysis of the IEEE 802.11 distributed coordination function", *IEEE Journal on selected areas in communications*, 18, 3, p. 535-547.
- Camps-Mur, Daniel, Manil Dev Gomony, Xavier Pérez-Costa et Sebastia Sallent-Ribes
- 2012 "Leveraging 802.11 n frame aggregation to enhance QoS and power consumption in Wi-Fi networks", *Computer Networks*, 56, 12, p. 2896-2911.
- Chen, Qinghua, Guoxi Liang et Zhengqiu Weng
- 2019 "A target wake time based power conservation scheme for maximizing throughput in IEEE 802.11 ax WLANs", in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, p. 217-224.
- Chen, Qinghua et Yi-Hua Zhu
- 2020 "Scheduling channel access based on target wake time mechanism in 802.11 ax WLANs", *IEEE Transactions on Wireless Communications*, 20, 3, p. 1529-1543.
- Dembele, Hamidou, Elio Tohme et Robin Colin
- 2023 "Assessing and Modeling the Energy Consumption of PoE-Powered WiFi Access Point", *IEEE Access*.
- Deng, Cailian, Xuming Fang, Xiao Han, Xianbin Wang, Li Yan, Rong He, Yan Long et Yuchen Guo
- 2020 "IEEE 802.11 be Wi-Fi 7 : New challenges and opportunities", *IEEE Communications Surveys & Tutorials*, 22, 4, p. 2136-2166.
- Dong, Mengxi, Zhanji Wu, Xiang Gao et Huan Zhao
- 2016 "An efficient spatial group restricted access window scheme for IEEE 802.11 ah networks", in *2016 sixth international conference on information science and technology (ICIST)*, IEEE, p. 168-173.
- Gast, Matthew S
- 2013 *802.11 ac : a survival guide : Wi-Fi at gigabit and beyond*, " O'Reilly Media, Inc."
- Guérin, Esther, Thomas Begin, Isabelle Guérin Lassous et Anthony Busson
- 2023 "Performance Analysis of MAC Energy-saving Strategies for WLANs", in *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, p. 119-126.
- Guérin, Esther, Thomas Begin et Isabelle Guérin Lassous
- 2023 "An overview of MAC energy-saving mechanisms in Wi-Fi", *Computer Communications*, 203, p. 129-145.

- Halperin, Daniel, Ben Greenstein, Anmol Sheth et David Wetherall  
 2010 "Demystifying 802.11 n power consumption", in *Proceedings of the 2010 international conference on Power aware computing and systems*, USENIX Association Berkeley, CA, USA, p. 1.
- He, Yong et Ruixi Yuan  
 2009 "A novel scheduled power saving mechanism for 802.11 wireless LANs", *IEEE Transactions on Mobile Computing*, 8, 10, p. 1368-1383.
- Heidelberger et Lavenberg  
 1984 "Computer performance evaluation methodology", *IEEE Transactions on Computers*, 100, 12, p. 1195-1220.
- "IEEE 802.11-saturation throughput analysis"  
 1998 *IEEE communications letters*, 2, 12, p. 318-320.
- "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 - Corrigendum 1 – Correct IEEE 802.11ay Assignment of Protected Announce Support bit"  
 2022 *IEEE Std 802.11-2020/Cor 1-2022 (Corrigendum to IEEE Std 802.11-2020 as amended by IEEE Std 802.11ax-2021, IEEE Std 802.11ay-2021, and IEEE Std 802.11ba-2021)*, DOI : [10.1109/IEEESTD.2022.9999411](https://doi.org/10.1109/IEEESTD.2022.9999411).
- IEEE WG 802.11  
 1999 *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, Standard 802.11, IEEE WG 802.11.
- IEEE WG 802.11ah  
 2016 *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification; Sub 1 GHz License Exempt Operation*, Standard 802.11ah, IEEE WG 802.11ah.
- IEEE WG 802.11ax  
 2021 *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification; Enhancements for High-Efficiency WLAN*, Standard 802.11ax, IEEE WG 802.11ax.
- IEEE WG 802.11e  
 2005 *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification; Medium Access Control (MAC) Quality of Service Enhancements*, Standard 802.11e, IEEE WG 802.11e.
- Jang, Ki-Young, Shuai Hao, Anmol Sheth et Ramesh Govindan  
 2011 "Snooze : Energy management in 802.11 n WLANs", in *Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies*, p. 1-12.
- Khorov, Evgeny, Anton Kiryanov, Andrey Lyakhov et Giuseppe Bianchi  
 2018 "A tutorial on IEEE 802.11 ax high efficiency WLANs", *IEEE Communications Surveys & Tutorials*, 21, 1, p. 197-216.
- Kureev, Aleksey, Dmitry Bankov, Evgeny Khorov et Andrey Lyakhov  
 2017 "Improving efficiency of heterogeneous Wi-Fi networks with joint usage of TIM segmentation and restricted access window", in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, p. 1-5.

- Lee, Il-Gu, Kyungmin Go et Jung Hoon Lee  
 2020 "Battery draining attack and defense against power saving wireless LAN devices", *Sensors*, 20, 7, p. 2043.
- Lee, Tsern-Huei et Jing-Rong Hsieh  
 2012 "Low-complexity class-based scheduling algorithm for scheduled automatic power-save delivery for wireless LANs", *IEEE Transactions on Mobile Computing*, 12, 3, p. 571-580.
- Liu, Ruofeng et Nakjung Choi  
 2023 "A first look at wi-fi 6 in action : Throughput, latency, energy efficiency, and security", *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7, 1, p. 1-25.
- Maudet, Sébastien, Guillaume Andrieux, Romain Chevillon et Jean-François Diouris  
 2023 "Practical evaluation of Wi-Fi HaLow performance", *Internet of Things*, p. 100957.  
 2024 "Evaluation and Analysis of the Wi-Fi HaLow Energy Consumption", *IEEE Internet of Things Journal*.
- Morin, Elodie, Mickael Maman, Roberto Guizzetti et Andrzej Duda  
 2017 "Comparison of the device lifetime in wireless networks for the internet of things", *IEEE Access*, 5, p. 7097-7114.
- Mozaffariahrar, Erfan, Fabrice Theoleyre et Michael Menth  
 2022 "A survey of Wi-Fi 6 : Technologies, advances, and challenges", *Future Internet*, 14, 10, p. 293.
- Nurchis, Maddalena et Boris Bellalta  
 2019 "Target wake time : Scheduled access in IEEE 802.11 ax WLANs", *IEEE Wireless Communications*, 26, 2, p. 142-150.
- Ozkaya, Ozgur, Jetmir Haxhibeqiri, Ingrid Moerman et Jeroen Hoebeke  
 2023 "Utilizing Communication Air-Time to Reduce Energy Consumption of Access Points", in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, p. 867-872.
- Perez-Costa, Xavier et Daniel Camps-Mur  
 2010 "IEEE 802.11 E QoS and power saving features overview and analysis of combined performance [Accepted from Open Call]", *IEEE Wireless Communications*, 17, 4, p. 88-96.
- Pérez-Costa, Xavier et Daniel Camps-Mur  
 2006 "AU-APSD : Adaptive IEEE 802.11 e unscheduled automatic power save delivery", in *2006 IEEE International Conference on Communications*, IEEE, t. 5, p. 2020-2027.
- Pérez-Costa, Xavier, Daniel Camps-Mur et Albert Vidal  
 2007 "On distributed power saving mechanisms of wireless LANs 802.11 e U-APSD vs 802.11 power save mode", *Computer Networks*, 51, 9, p. 2326-2344.
- Saeed, Ahmed et Mario Kolberg  
 2018 "Towards optimizing WLANs power saving : Novel context-aware network traffic classification based on a machine learning approach", *IEEE Access*, 7, p. 3122-3135.
- Santi, Serena, Le Tian, Evgeny Khorov et Jeroen Famaey  
 2019 "Accurate Energy Modeling and Characterization of IEEE 802.11 ah RAW and TWT", *Sensors*, 19, 11, p. 2614.

- Serrano, Pablo, Andres Garcia-Saavedra, Giuseppe Bianchi, Albert Banchs et Arturo Azcorra  
 2014 "Per-frame energy consumption in 802.11 devices and its implication on modeling and design", *IEEE/ACM Transactions on networking*, 23, 4, p. 1243-1256.
- Sheth, Jaykumar, Cyrus Miremadi, Amir Dezfouli et Behnam Dezfouli  
 2021 "EAPS : Edge-assisted predictive sleep scheduling for 802.11 IoT stations", *IEEE Systems Journal*, 16, 1, p. 591-602.
- Silva, Paulo, Nuno T Almeida et Rui Campos  
 2019a "A comprehensive study on enterprise Wi-Fi access points power consumption", *IEEE Access*, 7, p. 96841-96867.  
 2019b "Energy consumption management for dense Wi-Fi networks", in *2019 Wireless Days (WD)*, IEEE, p. 1-8.
- Site de la Wi-Fi Alliance  
 s. d. <https://www.wi-fi.org/>, Accessed : 2024-10-03.
- Site de la Wi-Fi Alliance  
 s. d. <https://www.wi-fi.org/beacon/the-beacon/the-state-of-connectivity-wi-fi-momentum-in-2024>, Accessed : 2024-10-03.
- Souilah, Hocine, Abderrahmane Baadache et Louiza Bouallouche-Medjkoune  
 2014 "Coping with Spoofed PS-Poll Based DoS Attack in IEEE 802.11 Networks.", in *VECoS*, Citeseer, p. 57-62.
- Stepanova, Ekaterina, Dmitry Bankov, Evgeny Khorov et Andrey Lyakhov  
 2020 "On the joint usage of target wake time and 802.11 ba wake-up radio", *IEEE Access*, 8, p. 221061-221076.
- Tian, Le, Sébastien Deronne, Steven Latré et Jeroen Famaey  
 2016 "Implementation and validation of an IEEE 802.11 ah module for ns-3", in *Proceedings of the 2016 Workshop on Ns-3*, p. 49-56.
- Tian, Le, Serena Santi, Amina Seferagić, Julong Lan et Jeroen Famaey  
 2021 "Wi-Fi HaLow for the Internet of Things : An up-to-date survey on IEEE 802.11 ah research", *Journal of Network and Computer Applications*, 182, p. 103036.
- Tian, Le, Amina Šljivo, Serena Santi, Eli De Poorter, Jeroen Hoebeke et Jeroen Famaey  
 2018 "Extension of the IEEE 802.11 ah ns-3 Simulation Module", in *Proceedings of the 2018 Workshop on Ns-3*, p. 53-60.
- Venkateswaran, Shyam Krishnan, Ching-Lun Tai, Yoav Ben-Yehezkel, Yaron Alpert et Raghupathy Sivakumar  
 2022 "IEEE 802.11 PSM in ns-3 : A Novel Implementation and Evaluation under Channel Congestion", in *Proceedings of the 2022 Workshop on ns-3*, p. 33-40.
- Venkateswaran, Shyam Krishnan, Ching-Lun Tai, Roshni Garnayak, Yoav Ben-Yehezkel, Yaron Alpert et Raghupathy Sivakumar  
 2024 "IEEE 802.11 ax Target Wake Time : Design and Performance Analysis in ns-3", in *Proceedings of the 2024 Workshop on ns-3*, p. 10-18.
- Wang, Qi, Katia Jaffrès-Runser, Jean-Luc Scharbarg, Christian Fraboul, Yi Sun, Jun Li et Zhongcheng Li  
 2015 "A thorough analysis of the performance of delay distribution models for IEEE 802.11 DCF", *Ad Hoc Networks*, 24, p. 21-33.

- Woesner, Hagen, J-P Ebert, Morten Schlager et Adam Wolisz  
1998 "Power-saving mechanisms in emerging standards for wireless LANs : The MAC level perspective", *IEEE Personal Communications*, 5, 3, p. 40-48.
- Wu, Shih-Lin et Shu-Chia Chen  
2011 "An energy-efficient MAC protocol with downlink traffic scheduling strategy in IEEE 802.11 infrastructure WLANs", *Journal of Systems and Software*, 84, 6, p. 1022-1031.
- Yang, Changmok, Jinmyeong Lee et Saewoong Bahk  
2021 "Target wake time scheduling strategies for uplink transmission in IEEE 802.11 ax networks", in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, p. 1-6.
- Zhao, Chengqing, Borui Li, Shuai Wang et Tian He  
2023 "The First Measurement Study of Target Wake Time Mechanism in 802.11 ax on COTS Devices", in *ICC 2023-IEEE International Conference on Communications*, IEEE, p. 4695-4700.