Load inference in Wi-Fi networks : models and experimentations

Nour El Houda Bouzouita

Numéro National de Thèse : **2022LYSEN003**

THESE DE DOCTORAT DE L'UNIVERSITE DE LYON
opérée par
l'Ecole Normale Supérieure de Lyon

Ecole Doctorale N° 512
École Doctorale en Informatique et Mathématiques de Lyon

Spécialité de doctorat : Informatique

Soutenue publiquement le 25/01/2022, par :
Nour El Houda Bouzouita

Inférence de la charge dans les réseaux Wi-Fi: modèles et expérimentations

---

Devant le jury composé de :

Noël, Thomas	Professeur des universités	Université de Strasbourg	Rapporteur
Boukhatem, Lila	Maître de Conférences	Université Paris-Saclay	Rapporteure
Boussetta, Khaled	Maître de Conférences	Université Paris 13	Examinateur
Vèque, Véronique	Professeur des universités	Université Paris-Saclay	Examinatrice
Busson, Anthony	Professeur des universités	ENS de Lyon	Directeur de thèse
Rivano, Hervé	Professeur des universités	INRIA	Co-encadrant de thèse

Abstract

As one of the cornerstones for delivering flexibility and ease of deployment, Wireless Local Area Networks (WLANs), especially IEEE 802.11, have been widely deployed in a variety of situations: homes, corporate or campus networks, public areas, which has led to the explosion of wireless data usage and the colossal rise of access points, smartphones, and various mobile devices. In such dense environments, a device seeking connectivity must choose among multiple available Wi-Fi networks that are within its radio range. However, the procedure for selecting an access point is still a striking concern and a critical ongoing challenge, especially in public areas (e.g., train stations, airports, malls, etc.), since it is based on simple criteria that do not relate to the quality of service that the device will experience. In particular, the network load is not taken into account even though it is a key parameter for the quality of service and experience. In this dissertation, we study the possibility/capacity for an unmodified vanilla device, especially a smartphone, to estimate the load of a network from local measurements in the user space with no interventions from the access points nor root permissions. The network load can be expressed in many ways. In this work, we consider the Busy Time Fraction (BTF), defined as the fraction of time the wireless medium is sensed busy due to successful or unsuccessful transmissions.

In this regard, we propose relatively simple and versatile analytical Markovian models specific to the application of BTF estimation in the presence of the IEEE 802.11 frame aggregation scheme introduced in recent 802.11 amendments. We model and simulate different scenarios in which a device induces the UpLink (UL) or the DownLink (DL) mean aggregation levels, in the user space, of an aggregated deterministic probe traffic competing with the traffic present in the network that can aggregate or not its frames.

We then propose a novel and practical method called Frame Aggregation based Method (FAM). It leverages the frame aggregation mechanism to estimate the network load through its BTF and characterize the network traffic type. FAM combines an active probing technique to measure the actual packet aggregation level and analytical Markov models that provide the expected rate as a function of the volume and nature of the traffic on the network.

The performance evaluation of the proposed Markovian models and the method has been established with the aid of the ns-3 network simulator and experimental test-beds under several scenarios. Results have shown that our method FAM is able to infer the network load with a granularity based on six levels of network loads for the considered scenarios.

Keywords: IEEE 802.11 networks, aggregated MAC protocol data unit (A-MPDU), Performance evaluation, Network load, Markov chain.

Résumé

La technologie Wi-Fi est devenue un réel besoin puisqu'elle répond de manière fiable et rapide à la demande exponentielle des services de données sans fil. Les réseaux Wi-Fi se sont densifiés ces dernières années et ils ont été largement déployés dans de nombreuses situations: réseaux d'entreprises ou de campus, espaces publics, etc. Dans de tels environnements denses, un terminal cherchant à se connecter doit choisir parmi plusieurs réseaux Wi-Fi disponibles. Cependant, la procédure de sélection d'un point d'accès (AP) reste une préoccupation majeure, en particulier dans les espaces publics (gare, aéroport, etc.), car elle est basée sur des critères simples qui prennent en compte que la qualité du lien d'un utilisateur et négligent les informations prenant en compte les autres. En particulier, la charge du réseau n'est pas prise en compte alors qu'elle est un paramètre clé pour la qualité du service. Dans cette thèse, nous abordons comment un appareil non modifié, en particulier un smartphone, pourrait estimer la charge d'un réseau via le temps d'occupation du canal (BTF) dans l'espace utilisateur sans aucune intervention des APs. Le BTF est défini comme étant la fraction de temps pendant laquelle le support sans fil est considéré comme occupé en raison de transmissions réussies ou non.

À cet égard, nous proposons des modèles analytiques, basés sur des chaines de Markov relativement simples et polyvalents, spécifiques à l'application de l'estimation de BTF en présence du mécanisme d'agrégation de trames IEEE 802.11 introduit dans les récents standards (802.11n, ac, ax). Nous modélisons et simulons différents scénarios dans lesquels un terminal induit le taux d'agrégation moyen de la liaison montante (UpLink) ou descendante (DownLink) d'un trafic de sondage déterministe agrégé en concurrence avec le trafic présent dans le réseau qui peut agréger ou non ses trames dans l'espace utilisateur.

Nous proposons ensuite une nouvelle méthode nommée FAM (Frame Aggregation based Method). Elle exploite le mécanisme d'agrégation de trames pour inférer le BTF et caractériser le type de trafic réseau qui peut agréger ou non ses trames. Cette méthode combine une technique de sondage actif pour mesurer le taux réel d'agrégation de trames et des modèles analytiques basés sur des chaines de Markov qui fournissent le taux d'agrégation théorique en fonction du niveau de charge et de la nature du trafic dans le réseau. Les modèles et la méthode sont confrontés à des simulations effectuées sur le simulateur réseau ns-3 et à des expérimentations réelles. Les résultats ont montré que notre méthode FAM est capable de déduire la charge du réseau avec une granularité basée sur différents niveaux de charges du réseau pour les scénarios considérés.

Mots clés : Réseaux IEEE 802.11, Agrégation de trames A-MPDU, Évaluation des performances, charge du réseau, Chaîne de Markov.

Résumé Long

Les réseaux Wi-Fi se sont densifiés ces dernières années ce qui mène à la forte augmentation des points d'accès (APs). Dans un environnement qui propose plusieurs réseaux Wi-Fi, un client cherche toujours à se connecter au réseau Wi-Fi ou point d'accès (AP) (lorsque le réseau est constitué de plusieurs AP) capable de lui offrir le meilleur service à ce moment-là. Le critère de sélection conventionnelle est la puissance du signal en réception. Néanmoins, cette approche peut amener à préférer un point d'accès qui ne donnera pas forcément les meilleures performances. Par exemple, si il y a plusieurs APs avec différentes puissances de signal en réception, en se basant sur ce critère, tous les appareils peuvent sélectionner le même AP. Les autres APs ne sont pas choisis alors qu'ils peuvent proposer de meilleures performances (moins chargés par exemple).

Pour palier à ce problème, aujourd'hui les entreprises et les universités déploient un ensemble d'APs correspondant au même réseau Wi-Fi logique (identifié par un identifiant commun : le Service Set ID). L'ensemble des APs d'un même réseau Wi-Fi est géré par un contrôleur. Celui-ci collecte des informations provenant des APs (nombre de clients associés, canaux, bande passante disponible, etc.) et applique des algorithmes centralisés permettant d'améliorer les performances (répartition de charges, allocation des canaux optimaux, la sécurité, etc.).

Toutefois, la densification des réseaux Wi-Fi est aussi liée au fait que des réseaux Wi-Fi différents sont déployés dans des mêmes zones. Il est possible d'avoir à portée radio les box free (auquel les clients free peuvent se connecter), le réseau de la ville, le réseau de la gare SNCF, le réseau Wi-Fi du café auquel on est installé, etc. Un client a donc potentiellement le choix entre des réseaux Wi-Fi différents. Mais ces réseaux étant différents, ils ne sont pas contrôlés par un serveur central et le client ne peut pas savoir quel est le réseau pouvant lui offrir le meilleur service. Par conséquent, la procédure de sélection d'un AP reste une préoccupation majeure, en particulier dans les espaces publics puisqu'elle repose sur des critères simples qui prennent en compte que la qualité du lien d'un utilisateur et négligent les informations prenant en compte les autres. En particulier, la charge du réseau n'est pas prise en compte alors qu'elle est un paramètre clé de la qualité de service.

Parallèlement à l'omniprésence du Wi-Fi, la dernière génération de terminaux mobiles tels que les smartphones, les smartwatches et les tablettes sont rapidement devenues omniprésentes dans notre quotidien. Le succès des équipements électroniques portables par le grand public a amené à l'émergence de *mobile crowd-sensing*. Le mobile crowd-sensing est un nouveau paradigme dont le principe est d'utiliser ces équipements portables, dotés d'une multitude de capteurs intégrés, pour développer des plates-formes de mesures (mesures capteurs). Cette approche offre un grand nombre de possibilités dans la mesure où l'on profite des équipements omniprésents pour mesurer un phénomène donné sans avoir à déployer une infrastructure où un ensemble de capteurs spécifiques. Les applications visées et existantes sont multiples : du monitoring de quantités environnementales aux conditions de trafic sur les routes. Une question théorique sous-jacente au crowd-sensing qui est au cœur de cette thèse concerne la capacité à exploiter la masse et la diversité des mesures pour inférer une information que chaque objet ne peut pas mesurer individuellement. Par exemple, dans une application type « waze », chaque smartphone mesure sa vitesse de déplacement instantanée. Le système est ensuite en capacité d'inférer une estimation globale des temps de trajet (voire une prédiction).

L'objectif de cette thèse consiste à étudier la possibilité de fournir les bases techniques d'une future application de mobile crowd-sensing. Cette application permettrait d'améliorer le processus de la sélection d'un point d'accès dans les espaces publics de la manière suivante. Nous concevons une application de crowd-sensing dans laquelle les appareils participants mesurent et partagent la charge de leurs réseaux environnants. Ils construiraient un savoir collectif de sorte que lorsqu'un appareil arrive dans une zone et veut se connecter à un réseau Wi-Fi, il pourrait choisir le point d'accès le moins chargé pour essayer de se connecter. Pour atteindre cet objectif, il est nécessaire qu'un appareil non modifié puisse estimer la charge d'un réseau sans exigence sur les points d'accès. Cette fonctionnalité permettrait donc d'optimiser/de coordonner au travers

d'une association intelligente des réseaux Wi-Fi différents, d'améliorer la qualité de service utilisateur, et inciter les utilisateurs à participer aux applications collaboratives crowd-sensing.

Il est à noter que, dans cette thèse, nous nous intéressons au choix du type d'informations à collecter et le déploiement de mécanismes pour les mesurer. En particulier, nous étudions la possibilité/capacité d'un terminal mobile, généralement un smartphone, à estimer la charge du réseau Wi-Fi à partir de mesures dans l'espace utilisateur. La charge du réseau peut être exprimée de plusieurs façons. Dans ce travail, nous considérons le temps d'occupation du cannal (BTF), définie comme la fraction de temps pendant laquelle le médium sans fil est considéré comme occupé en raison de transmissions réussies ou non. Il s'agit de transmissions simultanées, constituant la charge de l'AP, ainsi que l'interférence inter-réseau.

Contributions de la thèse

Dans cette thèse, tout en relevant les défis mentionnés ci-dessus, nous apportons les contributions suivantes:

La première contribution est la proposition de deux modèles analytiques basés sur des chaînes de Markov spécifiques à l'application de l'estimation de la charge du réseau par un terminal mobile en utilisant l'agrégation de trames IEEE 802.11. Dans la première génération de modèles, nous avons supposé que la connexion entre l'AP et le serveur était idéale. Nous l'avons donc modélisée comme si le serveur était implémenté sur l'AP. Ces modèles permettent d'estimer les niveaux d'agrégation moyens de la liaison montante (UpLink (UL)) d'un trafic de sondage déterministe agrégé en concurrence avec le trafic du réseau actuel qui peut agréger ou non ses trames. Nous avons démontré par des simulations et des expérimentations que le débit qu'une station peut obtenir dépend du fait que le trafic concurrent utilise ou non l'agrégation de trames. Par conséquent, le premier modèle considère un trafic concurrent agrégé, tandis que le second modèle considère un trafic concurrent non agrégé. Les résultats numériques obtenus avec le simulateur de réseau ns-3 et une expérimentation ont montré l'efficacité de cette solution.

La deuxième contribution est de dériver la deuxième génération de modèles analytiques qui considèrent un scénario différent, plus réaliste et plus pratique. Ils considèrent que le serveur est intégré à un second dispositif sans fil appartenant à l'utilisateur et connecté au même point d'accès par le biais du même réseau Wi-Fi. Ils évaluent les niveaux d'agrégation de trames du trafic de sondage de la liaison descendante (DownLink (DL)) entre deux stations connectées au même AP. Deux chaînes de Markov sont proposées pour la génération de ce modèle. Le premier modèle est basé sur un trafic concurrent agrégé tandis que le deuxième est basé sur un trafic concurrent non agrégé. Tous les modèles proposés ont révélé que l'agrégation de trames incorpore un ensemble riche de propriétés qui peuvent être utilisées pour discerner la charge du réseau.

La troisième contribution consiste à proposer une nouvelle méthode nommée FAM (Frame Aggregation based Method). Elle exploite le mécanisme d'agrégation de trames, introduit depuis le standard IEEE 802.11n, pour estimer le temps d'occupation du canal (BTF) et donc la charge actuelle. Vu que le débit qu'un terminal peut obtenir dépend du fait que le trafic concurrent utilise ou non l'agrégation de trames, FAM estime non seulement le BTF mais aussi la nature du trafic. La précision de FAM est évaluée par des simulations effectuées sur le simulateur réseau ns-3 et des expérimentations. A partir des résultats, nous pouvons obtenir les informations techniques suivantes sur la faisabilité d'une plateforme de mobile crowd-sensing pour l'estimation de la charge du réseau.

- La précision du FAM est suffisante pour une classification de la charge en quelques niveaux.
- Lorsque la majorité du trafic concurrent n'utilise pas de mécanisme d'agrégation de trames, il n'est possible d'identifier que deux classes de la charge.

Les modèles et la méthode sont confrontés à des simulations effectuées sur le simulateur réseau ns-3 et à des expérimentations réelles. Les résultats ont montré que notre méthode FAM est capable de déduire la charge du réseau avec une granularité basée sur différents niveaux de charges du réseau pour les scénarios considérés.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof. Anthony Busson and Prof. Hervé Rivano, for giving me the opportunity to make this Ph.D. thesis. Thank you for your invaluable guidance, regular encouragement, constant support, inspiration, and thorough evaluations throughout the course of this thesis. It was a pleasure working with you.

I am deeply grateful to my beloved husband for his unconditional love, patience and for always encouraging me and believing in me. This work could not have been achieved without the continued support of my parents, sisters, brother, my parents-in-law, as well as my friends and lab peers, any words I could offer would only fail to grasp the full extent of your help.

Last but not least, I would like to warmly thank Thomas Noël and Lila Boukhatem for accepting to be Reporters of my thesis, as well as Khaled Boussetta and Véronique Vèque for accepting to be examiners of my thesis and for taking part as jury members. To all of you, thank you.

Nour El Houda Bouzouita

Contents

List of Figures

List of Tables

Introduction 1

1.1 General context

Since its apparition in the late 1980s, the advent of the Internet brought in several new revolutions. The beginning of the 20th century saw a revolution in the field of telecommunications with the advent of so called *radio network technologies* such as cellular (e.g., 3G, 4G) and wireless fidelity (Wi-Fi), among others. Unsurprisingly, Internet users have vigorously embraced these new ways of networking since they allow a cordless experience and quickly became the most favored manner for delivering several advantages over wired networks such as mobility, flexibility, ease of deployment, as well as reducing deployment costs.

" We are all now connected by the Internet, like neurons in a giant brain. "

– Stephen Hawking

In this work, we focus on IEEE 802.11 technology (commonly known as Wi-Fi). Nearly all indoor and outdoor environments such as companies, universities, enterprises, and homes use Wi-Fi to satisfy users' connectivity needs. Such proliferation has led to the explosion of wireless data usage and the colossal rise of access points (APs), smartphones, and various mobile devices, and these sharp increases are projected to continue in the foreseeable future. According to Cisco [1] , Internet users represented 3.9 billion in 2018, and they will exceed 5.3 billion by 2023, while the number of Wi-Fi hotspots will grow four-fold from 169 million hotspots in 2018 to nearly 628 million public Wi-Fi hotspots by 2023.

[1]: Cisco (2020), *Cisco Annual Internet Report (2018–2023) White Paper*

Such massive utilization has resulted in higher expectations from Wi-Fi users (efficiency, throughput, etc.), in particular, with the parallel enhancement of cellular networks and the advent of the fifth generation (5G) cellular technology [2] that lies in providing very high data rates (typically of Gbps order) and offering one of the highest performances with extremely low latency. To meet the increasing and continuous demand of Wi-Fi consumers, this technology has undergone a myriad of remarkable enhancements for both Physical Layer (PHY) and MAC Medium Access Control Layer (MAC), resulting in several generations of IEEE 802.11 standard amendments and the deployment of even more APs. In the dawning of the last new IEEE 802.11ax amendment, Wi-Fi networks are settled for the relentlessly booming of mobile devices that must share the spectral resources. Given the narrow spectrum for wireless channels, optimizing and evaluating the Wi-Fi performance in an efficient way is paramount important. That is what we aim to address in this dissertation in order to improve the users' perceived Quality of Service (QoS).

[2]: Agiwal et al. (2016), 'Next Generation 5G Wireless Networks: A Comprehensive Survey'

To join a Wi-Fi network, a wireless STAtion (STA) has to associate with an Access Point (AP). In some environments such as companies and universities, the choice of the access point is controlled: the objective is to join a specific network that is composed of a set of APs corresponding to the same logical Wi-Fi network (Extended Service Set (ESS)) and identified by a common Service Set IDentifier (SSID). This ESS is managed and coordinated by a centralized WLAN controller. This controller

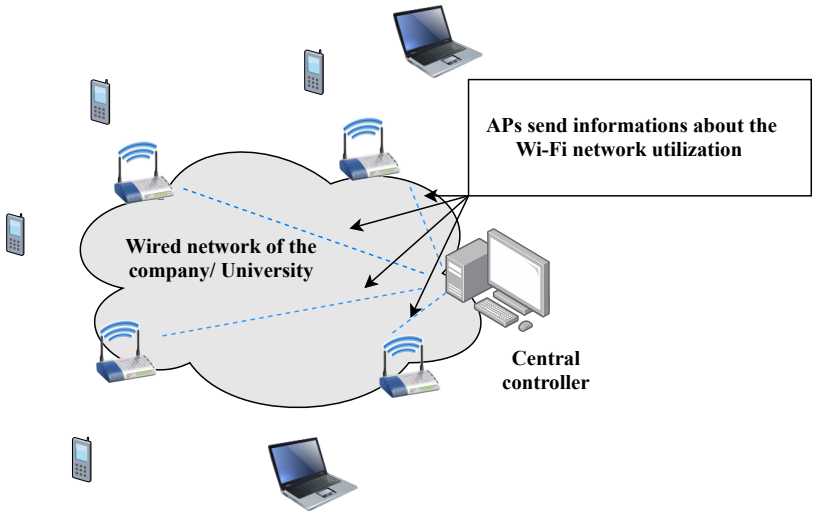


Figure 1.1: Centralized management

can implement performance optimization algorithms such as channel allocation and load balancing [3, 4] to eke out as much performance as possible, improve the network functionality, and guide the device to a chosen AP as shown in Figure 1.1. In this figure, the four APs offer the same Wi-Fi network to clients. These APs send data about the network utilization (number of associated clients, bandwidth, the channel used, etc.) to the controller. Based on collected information, the controller configures the APs. Such an approach aims to ensure efficient coverage of a Wi-Fi network and offers the STA the possibility to have at least an AP in its vicinity and judiciously associate with the optimal AP that guarantees the QoS it will experience. Furthermore, some works propose to manage networks trough software-defined networking (SDN) paradigm [5] .

[3]: Sood et al. (2015), 'Dynamic access point association using Software Defined Networking'
[4]: Amer et al. (2018), 'Considering Frame Aggregation in Association Optimization for High Throughput Wi-Fi Networks'

[5]: Yang et al. (2015), 'Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey'

1.2 Thesis statement

The spatial densification of Wi-Fi networks is also related to the fact that many different Wi-Fi networks can be deployed in the same area, especially in public areas. In such areas, there may be the choice between several access points, potentially belonging to different networks, without any coordination between them. If we take the example of a train station, a Wi-Fi network with a couple of APs may have been deployed by the train's company to offer free Wi-Fi service. Shops, bars, and restaurants may have deployed their own Wi-Fi networks, each composed of one or several independent APs. These public networks are managed by different entities and there is no common policy or rational management to help the devices select the Wi-Fi network or AP that offers the best performance. It is then up to the terminal to choose the access point it will associate with. This heterogeneity in the wireless networks is, therefore, an increasing problem that breaks down coordination between wireless terminals and APs and leads to a bad distribution of the load in the network, irrational management of the resources, and a penalty of the overall network performance.

Along with the ubiquity of heterogeneous Wi-Fi networks, the latest generation of mobile devices such as smartphones, smartwatches, and

tablets have quickly become necessary in our daily life. By taking advantage of their ubiquitous nature and rapid evolution over recent years, mobile computing devices with several embedded sensors (e.g., cameras, accelerometers, etc.) have led to the emergence of Mobile Crowd Sensing. This paradigm is defined as "individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest" [6] . This approach offers a large number of possibilities where we take advantage of ubiquitous equipment to measure a given phenomenon without having to deploy an infrastructure or a set of specific sensors. The targeted applications and devices' existing systems are numerous: from monitoring environmental quantities to traffic conditions on the roads. For example, in a typical application "Waze", each smartphone measures its instantaneous speed of movement. These applications can be grouped into two categories: personal and community sensing based on the kind of phenomena being monitored. The personal sensing applications aim to monitor phenomena pertaining to an individual (e.g., walking, running,...). On the other hand, community sensing aims to supervise large-scale phenomena that cannot be performed by an object/person individually (e.g., air pollution level and traffic congestion monitoring). Based on the type of involvement from users, the community can be broadly classified into participatory sensing and opportunistic sensing. Participatory crowd sensing requires the active involvement of individuals in contributing sensing information (e.g., taking a picture) related to any large-scale processes of common interest whereas opportunistic crowd sensing requires a minimal user involvement where the data is sensed and shared automatically without user intervention.

[6]: Ganti et al. (2011), 'Mobile crowdsensing: current state and future challenges'

Our long-term objective is to design a crowd-sensing application. This approach would improve the access point selection process in public areas in the following way. We design a crowd-sensing application in which participant devices measure and share the load of their surrounding networks. They would build a collective knowledge so that when a device arrives in some area and wants to connect to a Wi-Fi network, it could choose the less loaded AP to try and attach. To reach this goal, it is necessary that an unmodified device can estimate the load of a network with no requirements on the access points. It is worth noting that, in this dissertation, we are interested in choosing the type of information to collect and deploying mechanisms to measure them. In particular, we study the possibility/capacity for a vanilla Wi-Fi client, typically a smartphone, to infer the Wi-Fi network load from local measurements in the user space.

1.3 Contributions

In this thesis, while tackling the above-mentioned challenges, we make the following contributions.

The **first contribution** is the proposition of two analytical models based on Markov chains specific to the application of the network load estimation by a device under the IEEE 802.11 frame aggregation scheme. In this first generation of models, we assumed that the connection between the AP and the server is ideal. We, therefore, modeled it as if the server is

implemented on the AP. These models allow estimating the UpLink (UL) mean aggregation levels, in the user space, of an aggregated deterministic probe traffic competing with current network traffic that can aggregate or not its frames. We demonstrated through simulations and a test-bed experiment that the throughput that a joining device could get depends on whether the competing traffic uses frame aggregation or not. Therefore, the first model considers aggregated concurrent traffic, whereas the second model captures non-aggregated cross traffic. The numerical results obtained with the network simulator ns-3 and a test-bed experiment have shown the effectiveness of this solution.

The **second contribution** is to derive the second generation of the analytical models that consider a different scenario which is a more realistic and practical one. They consider that the server is embedded on a second wireless device owned by the user and connected to the same AP through the same Wi-Fi network. They appraise the frame aggregation levels of the DownLink (DL) probe traffic between two stations connected to the same AP. Two Markov chains are proposed for this model generation. The first model is based on aggregated cross traffic while the second is based on non aggregated cross traffic. All the proposed models revealed that the frame aggregation scheme embodies a rich set of Wi-Fi link properties that can be used to discern the network load.

The **third contribution** is to propose a novel method Frame Aggregation based Method (FAM). It leverages the frame aggregation mechanism, introduced since the IEEE 802.11n amendment, to estimate the channel Busy Time Fraction (BTF), thus the current load. As the throughput that a joining device could get depends on whether the competing traffic uses frame aggregation or not, FAM estimates not only the BTF but also the nature of the traffic. FAM accuracy is evaluated through extensive discrete-event simulations and test-bed experiments. From the results, we can get the following engineering insights on the feasibility of a crowd-sensing platform for network load estimation.

- The accuracy of FAM is enough for the classification of the load in a few levels.
- When the majority of the competing traffic does not use frame aggregation, it is only possible to identify two load classes.
- The number of devices composing the DownLink competing traffic which is coming from the same AP does not influence the prediction.

1.4 Thesis outline

This manuscript is organized in seven chapters.

Chapter 1 presents the general context, as well as the motivation and contributions of this thesis.

Chapter 2 gives a survey on the IEEE 802.11 standard evolution from the first IEEE 802.11 release (1999), published more than two decades ago, up to the bleeding edge, IEEE 802.11ax (2020). It gives a description of the most MAC and PHY layers mechanisms while highlighting their evolution over the years.

Chapter 3 reviews some of the existing approaches in the field of network performance evaluation that one can use to evaluate these networks, with a focus on the IEEE 802.11 standard. We establish a taxonomy of these works according to several criteria. This chapter brings out the motivation of our contributions, discusses several open problems, and contextualizes the novelty of our work.

Chapter 4 gives a detailed description of the system technical implementation. It first exposes the various network load measurement metrics. A special focus has been laid on the metric of interest related to our work. It then exhibits the main challenges faced while estimating the network load in a vanilla device and provides the procedure of detecting the frame aggregation level at the user space. This chapter also presents the network simulation environment and the choice of the programming languages used throughout this thesis coupled with detailed explanations of the reasoning behind such choices. In the interest of clarity, we also include a detailed general description of the system that we consider in our modeling approaches that should help the reader to better understand the rest of this manuscript.

Chapter 5 presents a thorough description of our analytical Markovian models by exposing the system model, generic assumptions, notation, and mathematical formulation used in the different models. We then evaluate the accuracy of the models against a set of ns-3 simulations that consider different scenarios and a test-bed experiment.

Chapter 6 exposes the third contribution of this thesis which is the method FAM that helps a vanilla device to choose the less loaded AP in a given area by relying on the measurement of the actual frame aggregation level and the expected rate returned by the analytical models as a function of the volume and nature of the traffic on the network. The proposed method is evaluated through ns-3 simulations, a test-bed experiment, and a real-world trace-driven simulation, and a detailed analysis of the results is given.

Finally, Chapter 7 draws conclusions and discusses potential enhancements and drawbacks of our solutions.

1.5 Publications

The contributions of this thesis have been published or submitted in several peer-reviewed national and international conferences and international journals.

International Conferences

- Nour El Houda Bouzouita, Anthony Busson, Hervé Rivano, *Analytical Study of Frame Aggregation Level to Infer IEEE 802.11 Network Load*. 2020 International Wireless Communications and Mobile Computing (IWCMC), 2020, pp. 952–957
- Nour El Houda Bouzouita, Anthony Busson, Hervé Rivano, *Exploiting Frame Aggregation to Enhance Access Point Selection*. Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), 2021.

National Conferences

- Nour El Houda Bouzouita, Anthony Busson, Hervé Rivano. *"Etude du niveau d'agrégation des trames dans les réseaux IEEE 802.11 pour l'évaluation du niveau de charge"*. CORES 2020, Sep 2020, Lyon, France.

Journal paper

- [Under major revision] Nour El Houda Bouzouita, Anthony Busson, Hervé Rivano. *FAM: A Frame Aggregation Based Method to Infer the Load Level in IEEE 802.11 networks*. Computer Communications.

IEEE 802.11 Networks 2

This chapter goes into detail about the general context of this thesis. The first part of this chapter (Section 2.1) presents the basic architectures and mechanisms of the IEEE 802.11 standard. Section 2.2 brings a synthesis of the evolution of the IEEE 802.11 made over the years, from the original 802.11 standard published in 1997 to the IEEE 802.11ax amendment, while highlighting the main improvements brought generation after generation. We then present in Section 2.3 some new IEEE 802.11 features, mainly those related to this thesis. In Section 2.4, we present data rates and modulation and coding scheme. Section 2.5 exposes the channel access modes in Wi-Fi networks. At last, in Section 2.6, several challenges were mentioned to point out the relevance of the association process when it comes to choosing the best AP in public areas.

2.1 IEEE 802.11 overview

The IEEE 802.11 standards, known as Wi-Fi, are focused on the two lowest layers of the Open Systems Interconnection (OSI) model: the MAC Medium Access Control Layer (MAC), determining how to access the medium and send data, and the Physical Layer (PHY), dictating the details of transmission and reception.

The design of a wireless network under the IEEE 802.11 specifications relies on a specified architecture composed of multiple physical components as follows.

" I feel the need...the need for speed! "

– Maverick and Goose, Top Gun

Access Point (AP): an AP is a specific Wi-Fi node that performs the wireless-to-wired bridging function. In addition, it performs a number of other roles, such as the management of transmissions between devices that belong to the same Wi-Fi network.

STAtion (STA): a STA is a computing device with a wireless network interface such as a smartphone, laptop, tablet, etc. These STAs can be either mobile or fixed.

Wireless medium: a wireless medium is used to transport transmissions from a node to another node. Multiple PHY layers were developed to support the 802.11 MAC layer.

Distribution System: in order to ensure the connection of the AP to the network, each AP is associated with a Distribution System (DS). The DS is a logical component that forwards frames to their destination. The IEEE 802.11 standard does not specify any specified technology for the distribution system. The Ethernet is the commonly used technology.

The design of any WLAN relies on the Basic Service Set (BSS). The BSS is the set composed of a number of nodes in a specific way to communicate

together. Each BSS has a unique 48-bit identifier referred to as Basic Service Set IDentifier (BSSID).

BSSs can be of different kinds: Infrastructure BSS (known as infrastructure mode), Independant Basic Service Set (IBSS) (known as Ad-Hoc mode) or the Mesh Basic Service Set (MBSS) (Known as mesh mode). Figure 2.1 illustrates an example of the three modes. Infrastructure mode is distinguished by the utilization of the AP. In order to set up a network service, A STA must associate with an AP. All the communications must be centralized by taking two hops relayed through the AP. In Ad-Hoc mode, STAs communicate directly with each other. In mesh mode, nodes are comprised of mesh clients, mesh routers, and gateways, where each node acts as a host and a router, forwarding traffic on behalf of other nodes that might not be within the direct radio range of their destinations.

The 802.11 standard allows a group of BSSs to form an Extended Service Set (ESS) to create coverage in large-sized networks. An ESS is formed by linking BSSs together through the distribution system. It permits stations within the same ESS to communicate with each other, even though they belong to different Basic Service Area (BSA).

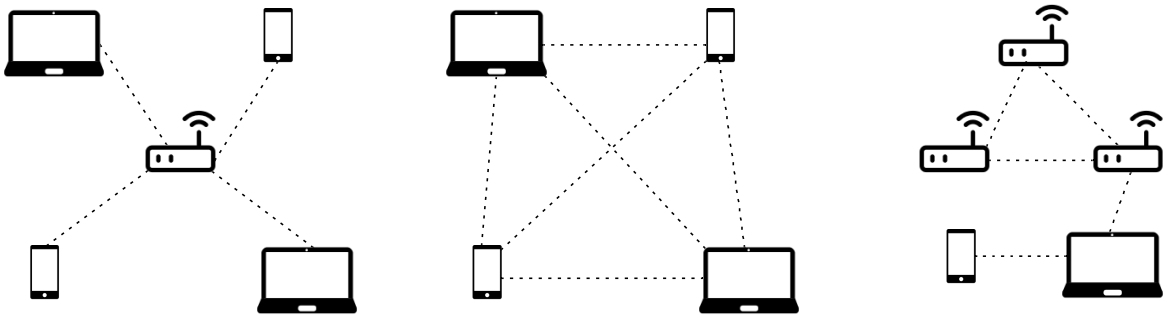


Figure 2.1: Comparison of infrastructure (left), ad-hoc (middle) and mesh (right) modes in WLANs.

2.2 IEEE 802.11 evolution

[7]: IEEE (1997), 'IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications'

The first IEEE 802.11 standard was initially published in 1997 [7] by the Working Group 11 of IEEE 802 LAN/MAN Standards Committee, providing the data rate of 1 or 2 Mbps and operating in the unlicensed 2.4 GHz radio frequency band. This band is reserved for the Industrial, Scientific and Medical (ISM) band. The PHY layer of this version uses either frequency hopping spread spectrum (FHSS), infrared (IR) or Direct Sequence Spread Spectrum (DSSS).

Since then, the IEEE 802.11 working group quickly began working on faster radio layers and developing new techniques implemented in different 802.11 standard amendments. In 1999, they standardized and published both 802.11b and 802.11a, operating respectively in the 2.4 GHz band and 5 GHz band, which is taken from the Unlicensed National Information Infrastructure (U-NII). The complementary code keying (CCK) mode was introduced in 802.11b to support up to 11 Mbps, whereas Orthogonal

Frequency-Division Multiplexing (OFDM) was applied for 802.11a, which pushes the original 2 Mbps data rate up to 54 Mbps.

On the other hand, there was a demand for higher data rates than 11b systems in the 2.4 GHz band. In 2003, 802.11g [8] was released, which is a 2.4 GHz standard supporting the data rate up to 54 Mbps by bringing the OFDM modulation technique of 11a.

[8]: (2003), 'IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band'

With the stunning growth of the usage of Internet browsing and multimedia services, the demand for higher data rates has never stopped growing in WLAN. In 2009, the IEEE working group made a significant step forward by publishing the 802.11n amendment, called Wi-Fi 4, or High Throughput (HT), reaching a theoretical maximum data rate of 600 Mbps. This throughput enhancement was made possible by introducing a bunch of new features to both PHY and MAC layers. These include:

- **PHY Enhancements:** 802.11n introduced the Multiple-Input Multiple-Output (MIMO) technology, which allows the utilization of multiple antennas at both the transmitter and receiver sides to send up to four spatial streams simultaneously for every single user (Single user MIMO). It also supports deployment of wider channels, with a width of 40 MHz, which is twice larger than those used in the previous 802.11a/b/g and uses higher 5/6 coding rates;
- **MAC Enhancements:** for enhancing efficiency, the main approaches in 802.11n are frame aggregation and block acknowledgments (detailed in Section 2.3). In addition, 802.11n introduces a new Reduced Inter-Frame Space (RIFS) of 2 μs that can be exploited instead of the 10 or 16 μs Short Inter-Frame Space (SIFS).

Table 2.1: Comparison of 802.11 standards.

standard	Speed	Frequency band
802.11	Up to 2 Mbps	2.4GHz
802.11a	Up to 54 Mbps	5GHz
802.11b	Up to 11 Mbps	2.4GHz
802.11g	Up to 54 Mbps	2.4GHz
802.11n	Up to 600 Mbps	2.4 GHz 5 GHz
802.11ac	Up to 1 Gbps	5GHz
802.11ax	Up to 10 Gbps	2.4 GHz 5 GHz

After the completion of 802.11n in 2009, an evolution from 802.11n was released, 802.11ac, called Wi-Fi 5. This new amendment, initiated by the Very High Throughput (VHT) study group and referred to as VHT standard, maintains the specifications of 802.11n while adopting some remarkable improvements. It introduced more advanced MIMO techniques. Unlike 802.11n that uses MIMO only to increase the number of data streams sent to a single client, 802.11ac deployed a DownLink (DL) Multiple User MIMO (MU-MIMO) that allows an AP to send data to multiple clients at the same time. 802.11ac also added new wider channel sizes: 80 MHz, 160 MHz, and “80+80 MHz", introduced the 256-QAM modulation that supports eight bits per symbol period and specifies up to eight spatial streams. Rather than operating all across the unlicensed spectrum bands allocated to WLANs, 802.11ac is restricted only to the 5 GHz frequency band.

Finally, the 802.11ax, which is both referred to as High Efficiency (HE) and Wi-Fi 6, was introduced by the IEEE 802.11 ax Task Group (TGax), mainly dealing with the performance of WLANs in massively crowded scenarios with a high density of user stations and APs. Similar to the previous amendments, 802.11ax develops a new set of PHY layer specifications. First, it has introduced the Orthogonal Frequency-Division Multiple Access (OFDMA), widely deployed in cellular networks, and has adopted it for both DownLink (DL) and UpLink (UL) transmissions. This technique relies on the same approach as OFDM but offers more resilience and enhancements by assembling adjacent sub-carriers (tones) into a Resource Unit (RU). A sender can thus choose the best RU for each particular receiver. Conversely to earlier standards, OFDMA allows allocating a

channel to multiple users in the same time slot, thereby improving the overall network efficiency. Moreover, unlike 802.11ac, which uses only the DownLink MU-MIMO, 802.11ax extends the MU communication by adopting the UpLink MU-MIMO which is based on sending multiple spatial streams to multiple stations. Additionally, 802.11ax has introduced the 1024-QAM modulation and has quadrupled the duration of the OFDM symbols used for the PHY payload up to 12.8μs. When it comes to the MAC layer, multiple improvements have been proposed. One of the most key features is the improvement of the Spatial Reuse (SR) operations such as enhancement of the PHY Clear Channel Assessment (CCA), BSS coloring, and Interference management, thereby enabling better management of the available resources.

In order to embrace the accumulated changes, a revision was applied to the original 802.11-1997 standard known as a roll-up. The roll-up of the approved amendments is carried out only once every several years since the modification of the entire 802.11 standard is a delicate mission. So far, the standard has been revised in 2007, 2012, 2016, and 2020 with the 802.11-2020 [9] being the most recent roll-up available.

[9]: IEEE (2021), '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'

In the next section, we will goes into detail about the new MAC features, introduced since IEEE 802.11n, since our contributions, discussed in the following chapters, are based on their behavior.

2.3 New IEEE 802.11 features overview

The recently rewritten IEEE 802.11 standard [9] and the most recent amendments, IEEE 802.11n, IEEE 802.11ac, and IEEE 802.11ax, made significant network efficiency and channel utilization gains by introducing new PHY and MAC layers features. The improvement of the MAC layer, since 802.11n, relies mainly on two new mechanisms, Frame Aggregation and Block acknowledgments (BlockACK). In the following, we briefly describe these two enhancements. We then give a brief description of how the AP queue handles packets under the frame aggregation, being of special interest to Chapter 5.

Frame aggregation

" We must, indeed, all hang together, or most assuredly we shall all hang separately. "

– Benjamin Franklin

Before adapting the HT enhancement in 802.11n, each source node spends a notable amount of time trying to gain access to the wireless shared medium instead of sending data. As a result, 802.11 MAC is generally reckoned to be 50% efficient. A simple solution to effectively spread the cost of gaining access to the radio resource is to send frames that carry several higher-layer packets (e.g., IP packets) known as frame aggregation. There are two types of methods that perform this mechanism: the Aggregate MAC Protocol Data Unit (A-MPDU) and the Aggregate MAC Service Data Unit (A-MSDU). These two schemes differ by wherein the network stack they apply the frame aggregation.

- A-MSDU: This scheme is performed before the MAC header encapsulation process. Its concept is to allow multiple Service Data Units (SDUs) to be grouped together. As shown in Figure 2.2, each A-MSDU sub-frame consists of a sub-frame header, an MSDU, and

padding bytes. The resulting A-MSDU, a common MAC header and Frame Check Sequence (FCS) are concatenated in a single MPDU with a PHY header to be sent to the same sink node. Consequently, the corruption of any sub-frame during transmission causes the corruption of the whole aggregated frame. The benefit of using A-MSDU is reducing the overhead. However, in noisy environments, the cost of the retries may be bigger than the gain of the aggregated frames.

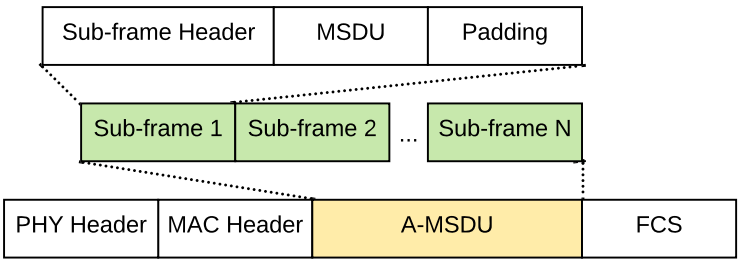


Figure 2.2: A-MSDU frame aggregation

- A-MPDU: This method is performed after the MAC encapsulation. Its principle is to assemble several MAC Protocol Data Units (MPDUs) sub-frames into a single PHY protocol data unit (PPDU) frame with a common PHY header and send it to the same receiver. Each A-MPDU, as illustrated in Figure 2.3, starts with an MPDU delimiter followed by the MPDU, consisting of its own MAC header, MAC payload, and FCS, and ends with padding bytes. Consequently, the corruption of any A-MPDU sub-frame does not cause the corruption of the whole A-MPDU, and only the corrupted MPDUs must be transmitted again. In environments that are prone to error with lots of retries, A-MPDU aggregation permits the receiver to acknowledge the MPDUs individually and definitely results in higher throughput than when using A-MSDU. Note that the maximum number of frames assembled within the same A-MPDU depends on many factors.

  - The block ACK frame can acknowledge a maximum of 64 frames (the 802.11ax amendment increased it to 256).
  - The size of an A-MPDU is limited and depends on the Wi-Fi card vendor's implementation.
  - Any of the following three conditions occur: 1) the number of bytes reaches the maximum; 2) the estimated transmission duration of the A-MPDU reaches the maximum; or 3) the number of frames within the same A-MPDU reaches the maximum.

The frame aggregation level is therefore dependent on the sender buffer state at the moment it accesses the medium.

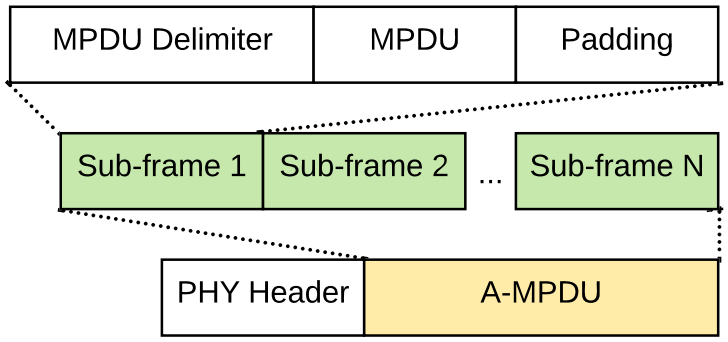


Figure 2.3: A-MPDU frame aggregation

To our knowledge, A-MSDU aggregation is rarely implemented in practice. In this thesis, we focus on A-MPDU aggregation because it is mandatory in the standard and implemented by default in recent Wi-Fi cards.

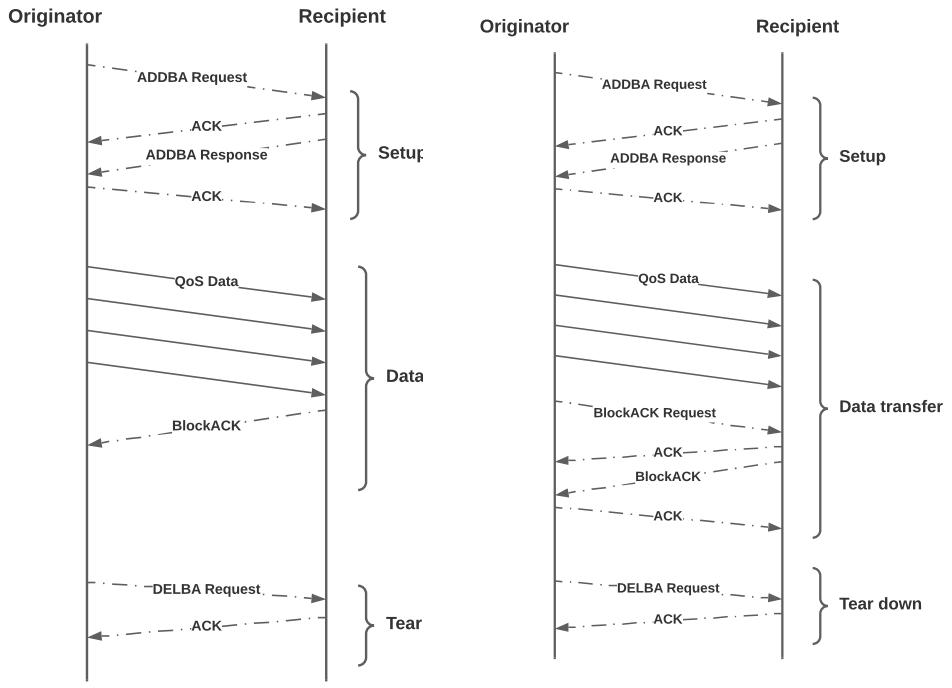
Block acknowledgment

Rather than sending an individual positive acknowledgment following each data frame, the 802.11e QoS extensions introduce Block acknowledgments (BlockACK) technique, allowing the receiver to transmit a burst of frames and have them all acknowledged at once by a single BlockACK frame. This mechanism was carried over to 802.11n, 802.11ac, and 802.11ax since it is well-suited for use with frame aggregation. By setting up a window, this scheme allows a receiver to selectively acknowledge all the successfully received frames within this latter and request the retransmission of just the lost frames by using the bitmap field. In this bitmap, each bit indicates the reception status (failure:bit=0/success:bit=1) of a subframe that has that offset from the starting sequence number. For example, if the starting sequence number is 200, then the first bit acknowledges sequence number 200, the second bit acknowledges sequence number 201, and so on.

Conversely to the normal ACK, which is automatically expected after a frame transmission, the BlockACK has to be negotiated through a BlockACK session. Two types of BlockACK were defined. The first one is the Immediate BlockACK (Figure 2.4a) which enables a receiver to acknowledge frames right away from their reception by implicitly requesting the BlockACK within the Data frame. This latter is used for applications with strong latency requirements. The second one is the Delayed BlockACK (Figure 2.4b) that allows a receiver to transmit the BlockACK later. It is rather intended for applications without strong latency constraints.

Both immediate BlockACK and delayed BlockACK sessions are composed of three main phases. Each session begins with the setup phase that consists of an exchange of ADD Block ACK (ADDBA) request and ADD Block ACK (ADDBA) response. The second phase consists in sending the Data blocks from the originator to the recipient. Once all data frames are correctly received, and the final BlockACK has been completed, the originator sends the DELBA request frame to its recipient. The recipient

of the DELBA frame shall release all resources allocated for the BlockACK transfer.



(a) Immediate BlockACK

(b) Delayed BlockACK

Figure 2.4: BlockACK

AP queuing system under frame aggregation

Before the introduction of frame aggregation scheme and for a given Enhanced Distributed Channel Access (EDCA) class (an extension of the Distributed Coordination Function scheme that includes Quality of Service traffic prioritization for data packets.), the AP purses a predefined per-packet scheduling method (e.g, First In First Out (FIFO)). It is important to note that building a queuing algorithm is up to product designers that decide how to manage a transmit queue.

With frame aggregation, the aggregated transmissions act as a batch scheduler which alters the timing characteristics of received packets [10, 11] . Figure 2.5 illustrates the following example. It consists of an AP queue with two interleaved sequences of packets sent to two different clients, A and B, on a Wi-Fi link. Ai defines the i^{th} packet targeted to client A while Bi is the i^{th} packet destined to client B. In the case depicted in Figure 2.5, the packet at the head of the AP queue is addressed to A. All A's packets (within the limit of the maximum number of aggregated sub-frames) will be aggregated and sent out as an A-MPDU. Then, an aggregated frame with B's packets is sent. In this thesis, we consider this approach in order to build the third Markovian model (detailed in Section 5.3).

[10]: Zhu et al. (2020), 'A Frame-Aggregation-Based Approach for Link Congestion Prediction in WiFi Video Streaming'
[11]: Song et al. (2017), 'Leveraging Frame Aggregation for Estimating WiFi Available Bandwidth'

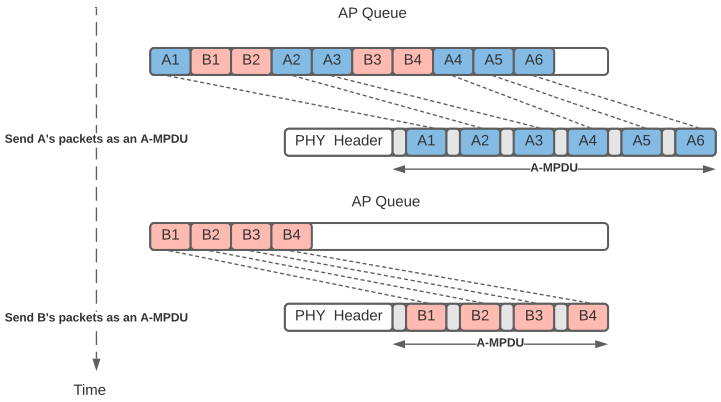


Figure 2.5: An Example of scheduling under frame aggregation.

2.4 Data rates and modulation and coding scheme

Each IEEE 802.11 standard defines a list of available data rates. The first 802.11-1997 standard has only two available data rates, 1 Mbps and 2 Mbps, while the modern standards starting from the 802.11n provide a list of data rates ranging from low to high. These latter are achieved by introducing the Modulation and Coding Scheme (MCS) that index the modulation type (e.g., BPSK, QPSK, 1024 QAM...) and the coding rate (such as 1/2, 2/3, 3/4...) couples. The physical transmission rate is thus a combination of the following parameters: the MCS index, the number of spatial streams, the channel width, and the guard interval length.

2.5 Channel access modes

To mediate the access to the shared wireless medium in IEEE 802.11 networks, coordination functions are used. Like Ethernet, Wi-Fi employs the Carrier Sense, Multiple Access (CSMA) scheme, operating in the MAC Layer in which carrier sensing is used before each transmission to circumvent simultaneous access (collisions) on the wireless medium. Nevertheless, rather than Carrier Sense, Multiple Access with Collision Detection (CSMA/CD) used by Ethernet that detects collisions as they happen on the medium, 802.11 relies on collision avoidance CSMA/CA. The wireless source node senses the radio link before transmitting and waits for it to be available since it cannot listen and transmit at the same time.

Clear Channel Assessment

In order to determine whether the wireless medium is currently in use before attempting to transmit, each 802.11 device performs a Clear Channel Assessment (CCA). The CCA includes two modes: carrier sensing and energy detection. The carrier sense mode is based on measuring the received signal strength and comparing it to a predefined threshold at which the CCA scheme detects a transmission. Any signal that is

greater than this threshold results in detecting a busy channel. For the energy detection mode, each device measures the energy level on the medium and compares it to a given threshold. Note that the thresholds for the two modes are predefined in the standard. For robustness, 802.11 also uses the virtual carrier sensing function, called Network Allocation Vector (NAV) which is a nonzero value included in most frames and indicates the number of microseconds that a device must wait for before attempting to transmit.

Inter-Frame Space

To handle the channel access for the STAs, the Distributed Coordination Function (DCF) relies on several timers called Inter-Frame Space (IFS) which is a period of inactivity between two successive frames. 802.11 defines four types of IFS to build up a set of system priority levels between different types of frames.

- Distributed Inter-Frame Space (DIFS): this Inter-Frame Space is used to separate atomic exchanges.
- Short Inter-Frame Space (SIFS): is the shortest of the four Inter-Frame spaces. It is used to separate frames within the same transmission (e.g., between the data frame and its ACK or BlockACK, or between fragments of the same frame.
- PCF Inter-Frame Space (PIFS): is used during the Point Coordination Function (PCF) mode in which any station is free to transmit if the medium is idle for the duration of one PIFS.
- Extended Inter-Frame Space (EIFS): is the longest of the four Inter-Frame spaces. It is used when an error occurs in transmission.

Note that 802.11n devices may use a new IFS called Reduced Inter-Frame Space (RIFS). This latter does not define a new priority level. Its only goal is to be used in place of the SIFS to increase the efficiency since it is shorter.

Distributed coordination function

IEEE 802.11 devices use the basic mechanism of Distributed Coordination Function (DCF) to gain access to the channel. An example of the basic of the DCF atomic transmission is depicted in Figure 2.6.

Before transmitting, each node has to make sure that the resource is not in use by listening to the medium for the period DIFS. Next, the node prepares the exponential contention window called backoff by picking a random slot and waiting for it. The station that picks the lowest number wins the medium and can transmit once its backoff counter reaches zero, while the backoffs of other stations are frozen and resumed when the medium becomes idle again. The backoff duration is calculated as the product of an integer value randomly generated in the interval [0, CW] and the slot time T_{slot}, where CW is the current contention window size. The mean duration of the backoff period is given as:

$$T_{backoff} = \frac{CW_{min} \times T_{slot}}{2} \tag{2.1}$$

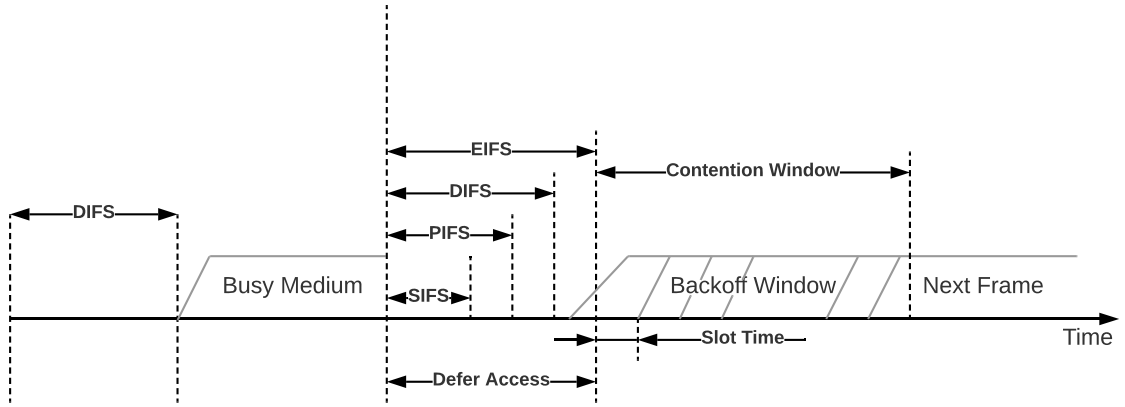


Figure 2.6: Basic channel access in DCF mode.

After the backoff duration has elapsed, the STA can then transmit for a duration referred to as Transmission Opportunity (TXOP). In 802.11 networks, the positive ACK or the blockACK are the only proof of successful transmissions. Each unicast data frame must be acknowledged by the receiver, or the frame is assumed to be lost and will be retransmitted. These types of frames are sent after the SIFS period. Since the ACK/BlockACK frame presents one of the management frames, this latter is sent using a data rate belonging to the basic rate set, which defines the list of the mandatory data rates that must be supported by any station wishing to make a happy union with the network. In case of non-received frames, the DCF, stations must wait for the EIFS period.

In order to enhance transmission efficiency under some circumstances, such as the hidden node problem that usually occurs in ad-hoc networks, the DCF may use the Request to Send, Clear to Send (RTS/CTS) clearing mechanism to further reduce the possibility of collisions. The node that gains the medium starts by first sending the RTS frame to explore the availability for exchanging data. If the sink node accepts the communication, it sends a CTS frame. Note that, in our work, we do not use the RTS/CTS mechanism.

Other access modes

Because contention-based channel access modes necessarily lead to collisions, the 802.11 standard defines other access modes. The PCF mode, built on top of the DCF, is used if contention-free services are required. This function is performed by special stations called Point coordinators, employed to guarantee that the medium is supplied without contention.

To enhance the QoS requirements, the IEEE 802.11e has put a notable effort and has introduced the Hybrid Coordination Function (HCF). This mode provides two channel access methods. The first one is the Enhanced Distributed Channel Access (EDCA) which is an enhanced version of the contention-based DCF that includes traffic prioritization based on differentiated Access Category (AC) that distinguish from the highest to the lowest priority: voice (AC_VO), video (AC_VI), best effort (AC_BE) and background (AC_BK). The second one is the Hybrid

Controlled Channel Access (HCCA) which is a contention-free technique that provides algorithms to schedule transmissions as a function of specific QoS demands such as, the bandwidth and the packet loss ratio.

The AP uses either DCF access mode or PCF access mode. In this thesis, we focus on the DCF which is the mode used in practice on the products.

2.6 Which AP to choose?

The widespread deployment of 802.11 networks means that a wireless STA is usually in the vicinity of several AP with which to affiliate. To join a network from these infrastructure-based networks, a user terminal has to associate with one AP. Networks deployed in public areas, such as airports, train stations, are usually independent BSSs managed by different entities and there is neither common policy or rational management nor coordination between them, to help the devices selecting the Wi-Fi network or AP that offers the best performance. It is then up to the terminal to choose the access point it will associate with.

Despite its importance, the choice made by the device's operating system is often based on simple criteria considering that the device is alone in its vicinity. Conceptually, usual metrics focus blindly on the Received Signal Strength Indicator (RSSI) which is the de-facto approach on most devices, if not only on a static list established from the user's connection history. However, these metrics do not convey information regarding other parameters that affect user performance. In particular, they do not consider the number of already associated STAs per AP nor the STA's data rate. Indeed, user terminals with low data rates occupy the channel longer than the terminals using high data rates, thereby significantly penalizing high data rate STAs. Therefore, the association between APs and STA must be based on other metrics that relate to the quality of service and experience that the device will experience in order to enhance the overall wireless network performance. A survey on the Wi-Fi performance evaluation metrics and tools that helps choose the best AP in terms of performance and availability is presented in the next chapter.

State of the Art for Wi-Fi Networks Performance Evaluation

3

With the increasing presence of WLANs and their densification, any reader familiar with the MAC and PHY layers improvements of the modern IEEE 802.11 standards can understand that properly building a functional Wi-Fi is far from a simple task. If not, Chapter 2 introduces the reader to the Wi-Fi standards and their evolution over the years. The performance evaluation of such networks has been intensely studied in the literature since their first introduction in the late 1990s to offer relevant information and methods to help its configuration and improvement. As the IEEE 802.11 standards have quickly evolved over the years, so did the works that modeled, simulated, and analyzed them. The various challenges of Wi-Fi performance evaluation can generally be grouped into two classes of work: those that relate to the performance methodology itself and those that relate to picking the relevant performance metric or combination of metrics depending on the desired goal. In this chapter, we focus on methods that focus on choosing the most appropriate metric. In this regard, we are interested in four categories of work that evaluate the performance of WLANs: bandwidth estimation, crowd-sensing of wireless networks, Analytical models, and Wi-Fi AP selection.

This chapter gives a detailed literature review regarding these topics. We first review the preliminary concepts and explore the existing works in the general Fields of active and passive bandwidth estimation while classifying them into sub-categories based on the specific metrics used. For the crowd-sensing and analytical models, the existing works are classified into different application scenarios and purposes. Finally, we discuss the main differences between the existing works and highlight the need for novel methods and approaches for the case of network load estimation for unmodified handheld devices.

3.1 Bandwidth estimation for performance evaluation of Wi-Fi networks

As a crucial metric for the QoS and Quality of Experience (QoE), the bandwidth estimation in computer networks has become a key parameter for measuring the overall network performance. Based on the metrics used, existing bandwidth estimation tools primarily estimate one or more of three related properties: capacity, available bandwidth, and bulk transfer capacity (achievable throughput). In Table 3.1, we provide a taxonomy of bandwidth estimation tools according to the bandwidth metric they try to discern.

Table 3.1: Bandwidth estimation related works summary.

Available bandwidth	Active	Packet Rate Model (PRM)	[12], [13], [14], [15], [11][1]
		Probe Gap Model (PGM)	[16], [17], [18], [19][1], [20][1]
	Passive	[21], [22][1], [23][1]	
capacity			[24], [25], [26], [27], [28], [29], [30]
Achievable throughput			[31], [32], [33], [34], [35], [36]

[1] Wi-Fi.

Capacity

The capacity presents the maximum transmission rate a device can achieve on a link. In the following, we distinguish between the data link capacity (Layer 2 (L2)) and the Internet Protocol (IP) link capacity (Layer 3 (L3)). In literature, the term link capacity generally refers to the measurement carried out at the application layer which is usually less than the data link transmission rate due to the layer 2 encapsulation's overhead and framing. Let us assume that the maximum transmission rate at layer 2 is C_{L2} and its header is H_{L2}. The layer 3 capacity, denoted C_{L3} when the payload size is P is given by:

$$C_{L3} = C_{L2} \times \frac{P}{P + H_{L2}} \tag{3.1}$$

When applying this definition to an end-to-end network path, composed of a sequence of hops, this capacity is generally determined by the minimum link capacity along the path, called *the narrow link*, expressed as:

$$C = min\ Ci, i = 1, ..., N \tag{3.2}$$

Where Ci represents the capacity of the ith hop and N denotes the number of hops in the network path.

It is worth keeping in mind that some technologies do not operate with a constant transmission rate, such as IEEE 802.11 WLANs that send their frames using different transmission rates depending on the bit error rate of the wireless medium. In these cases, the capacity is computed for given time periods in which the capacity remains constant.

Several capacity measurement techniques have been proposed. These works can be divided into two groups as follows.

Per-Hop capacity estimation techniques Per-Hop capacity estimation techniques aim at inferring the capacity of each hop in the network path. The end-to-end capacity is thereby the minimum of all hop measures.

In 1997, Jacobson designed Pathchar [30] , a tool aiming at estimating the characteristics of individual links along an Internet path. Since then, multiple techniques have been proposed to measure the per-hop capacity of the network path, such as clink [27] , pchar [24] , and [28] . These methods are based on a common measurement technique named Variable Packet Size (VPS). Bellovin [37] and Jacobson [30] were the first to propose this scheme. It sends probe packets from the source while varying packet size and the Time-To-Live (TTL) field of the IP header. The latter is used to force probing packets to expire at a specific hop. The router at that hop

[30]: Jacobson (1997), *Pathchar: A tool to infer characteristics of Internet paths*

[27]: Downey (1999), 'Using Pathchar to Estimate Internet Link Characteristics'
[24]: Mah (2000), 'Pchar: A tool for measuring internet path characteristics'
[28]: Lai et al. (2000), 'Measuring Link Bandwidths Using a Deterministic Model of Packet Delay'

[37]: Bellovin (1992), 'A Best-Case Network Performance Model'

discards these packets, thereby sending back an Internet Control Message Protocol (ICMP) time-exceeded error messages to the source. The source then uses these messages to measure the Round Trip Time (RTT) metric that in turn will be used to estimate the latency and bandwidth of each hop in the path. However, relying on such messages from routers limits the accuracy and the applicability of these tools.

End-to-End Capacity Estimation Tools To extend the capacity estimation from per-hop to end-to-end, several tools attempt to infer the capacity of the *narrow link* along an end-to-end path also referred to as *bottleneck bandwidth*. Most of them use the packet pair or packet train technique (presented below). Bprobe [25] utilizes the packet pair dispersion technique to discern the capacity along a given path by leveraging the ICMP protocol messages. In order to improve its accuracy, this tool processes variable-sized probing packets using union and intersection filtering to produce the final capacity estimate. The authors of [29] proposed Nettimer which is a capacity estimation packet pair tool to passively measure the bottleneck bandwidth of a path in real-time. It uses a statistical technique called *Kernel density estimation* that processes the packet pair measurements. Based on the distribution of these measurements, it then identifies the dominant mode. It is worth noting that nettimer is a method that performs both per-hop capacity estimation by using the packet tailgating technique [28] and end-to-end capacity estimation using the packet pair technique [29]. Pathrate [26] gathers many packet pair measurements using several probing packet sizes. It then Analyzes the distribution of the resulting measurements that reveals all local modes, one of which typically relates to the capacity of the path.

[25]: Carter et al. (1996), *Dynamic Server Selection Using Bandwidth Probing in Wide-Area Networks*

[29]: Lai et al. (1999), 'Measuring bandwidth'

[26]: Dovrolis et al. (2001), 'What do packet dispersion techniques measure?'

TCP throughput and Bulk transfer capacity

One of the common bandwidth-related metrics in TCP/IP networks is the Transmission Control Protocol (TCP) throughput of a TCP connection. TCP throughput metric is a crucial factor that is of great interest to end users since it remains the most widely used protocol till today (carries more than 80% of the internet traffic) [38] . The Bulk Transfer Capacity (BTC) represents the TCP achievable throughput, i.e., the maximum throughput obtained by a TCP connection. TReno [34] and cap [35] are considered as the pioneering works that help measuring the BTC.

Today, the throughput test tools such as, ttcp [32], netperf [33], iperf [31] have widely been embraced for TCP performance measurement. In addition, many popular Internet speed tests applications, such as [36, 39] offer suitable measurements via the web. As part of the measurement process, these tools examine the performance of an underlying end-to-end network path by performing an active TCP upload from a client to a server and an active TCP download to estimate the UL and DL TCP throughput respectively.

While throughput tests provide valuable insights into network state, they suffer from high intrusiveness (generated traffic) and dependence on transport and application protocol. Unfortunately, the expected throughput of a TCP connection cannot be easily forecast by an end-user as it depends on multiple factors, such as the transfer size, the number of

[38]: Murray et al. (2017), 'An analysis of changing enterprise network traffic characteristics'

[34]: Mathis (1999), *TReno Bulk Transfer Capacity*

[35]: Allman (2001), 'Measuring End-to-End Bulk Transfer Capacity'

[36]: (2021), *Ookla Speedtest*

[39]: (2021), *AT & T Internet Speed Test*

competing TCP connections, and the type of the cross traffic (UDP or TCP).

Available bandwidth

In order to avoid the intrusiveness and high-cost time of achievable throughput tests, several works focus on estimating the available bandwidth to infer the network status. The end-to-end available bandwidth of a network path, i.e., the residual capacity that is left over by other traffic, is determined by its *tight link*, which is the link that has the minimal available bandwidth, during a certain time period. The available bandwidth of a link depends on the underlying technology, its transmission parameters and medium, and the traffic load at that link.

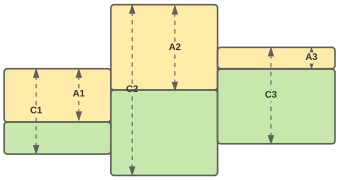


Figure 3.1: A three-hop network path

At any time instant, the network link is either transmitting using a given capacity or idle. So its available bandwidth is given by:

$$Ai = Ci \times (1 - Ui) \tag{3.3}$$

where Ai defines the available bandwidth of the hop i during a specific time period, Ci presents the ith hop's capacity, and Ui is the average utilization.

Figure 3.1 illustrates an example of an end-to-end network path with three hops, where each hop is modeled by a pipe. The width of each pipe corresponds to the capacity of the corresponding link. The green area of each pipe depicts the utilized part of that link's capacity, while the yellow area depicts the available capacity. The minimum link capacity, $C1$ in this example, determines the end-to-end capacity, whereas the minimum available bandwidth $A3$ determines the end-to-end available bandwidth. This example nicely highlights the fact that the narrow link of this path is not the same as the tight link.

In order to conduct Available bandwidth estimation in IP networks, there are primarily two types of measurements: passive and active. The former non-intrusively monitors and analyses the current real traffic while the latter involves extra traffic during the estimation process.

[40]: Nayak et al. (2019), 'Virtual Speed Test: an AP Tool for Passive Analysis of Wireless LANs'
[21]: Zhu et al. (2020), 'A Frame-Aggregation-Based Approach for Link Congestion Prediction in WiFi Video Streaming'

Passive measurement approaches [21, 40] evaluate the available bandwidth based on the non-intrusive monitoring of network traffic without emitting any traffic. These tools capture and analyze live network traffic by relying on specific hardware, e.g., chipsets that can switch to monitor mode and dedicated capture software. Some techniques even need to be run on routers or require direct access to the router or any node in the network path. Additional challenges include the capturing speed used during the measurement, data storage capabilities, and processing energy for analyzing data traces. On the contrary, active methods often need a lighter set-up, are easier to deploy and run on regular devices and grant a flexible design of the probe packets. The counterpart is that they strain the network by injecting probe traffic, thus perturbing the measure itself and laying additional burdens over the resource in terms of intrusiveness. In this dissertation, we focus on active measurements since it is non-trivial to collect the network performance information from passive measurements.

Multiple active measurement techniques have been developed, in the field of estimation of available bandwidth and busy time, in wired and wireless networks.

Available bandwidth estimation in wired networks

Most of the available bandwidth estimation techniques can be grouped into two classes.

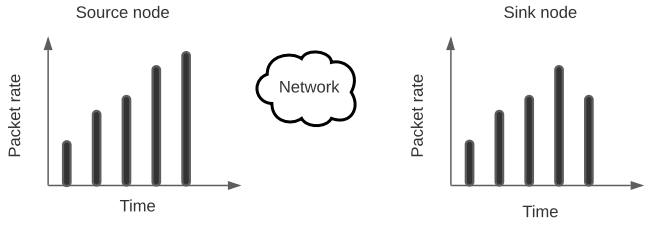


Figure 3.2: Packet Rate Model

The Packet Rate Model (PRM) This model (Figure 3.2) is based on the concept of self-induced congestion. It calculates the end-to-end available bandwidth by monitoring the transmitted and received packet rates and detecting the queuing delays. For doing so, a sequence of small probe packets is sent at different rates from a source to a sink node. If the probe packet rate exceeds the actual available bandwidth, then probe packets will be queued up at a router, and the packet rate at the reception will be therefore lower than at emission. Thus, the available bandwidth can be discerned by detecting the turning rate at which queuing delays start to occur.

A significant number of PRM techniques have been developed, such as TOPP [12] , pathload [13] , pathChirp [14] and DietTOPP [15] . These tools differ according to the probing rate adjustment and in their receiver-side analysis approaches. Pathload uses Constant Bit Rate (CBR) streams and adjusts the probe rate during each round based on a binary search method. TOPP relies on a linearly growing rate. As a typical PRM technique, DietTOPP deploys the TOPP algorithm with a simplified search method. Compared to Pathload, PathChirp reduces probe traffic overhead by using a sequence of exponentially spaced probe packets of the same size, denoted chirps. Within the same chirp, several rates can be probed, thereby improving accuracy.

[12]: Melander et al. (2000), 'A new end-to-end probing and analysis method for estimating bandwidth bottlenecks'
[13]: Jain et al. (2002), 'Pathload: A Measurement Tool for End-to-End Available Bandwidth'
[14]: Ribeiro et al. (2003), 'PathChirp: Efficient Available Bandwidth Estimation for Network Paths'
[15]: Johnsson et al. (2006), 'An Analysis of Active End-to-end Bandwidth Measurements in Wireless Networks'

The Probe Gap Model (PGM) This model (Figure 3.3) measures the available bandwidth by inferring the intensity of the cross traffic at the bottleneck. PGM techniques typically send a batch of probes into the network path at a single rate. These packets are sent with a defined packet size L and inter-packet gap referred to as input gap δ_{in}. While traversing through the network path, the probe packets compete with the cross traffic and reach the receiver with a new inter-packet gap denoted δ_{out}. These techniques then rely on the dispersion in time between two successive probes at the receiver side $\Delta = \delta_{out} - \delta_{in}$ to estimate the cross traffic. The dispersion is positive, i.e., $\delta_{out} > \delta_{in}$, when the cross traffic

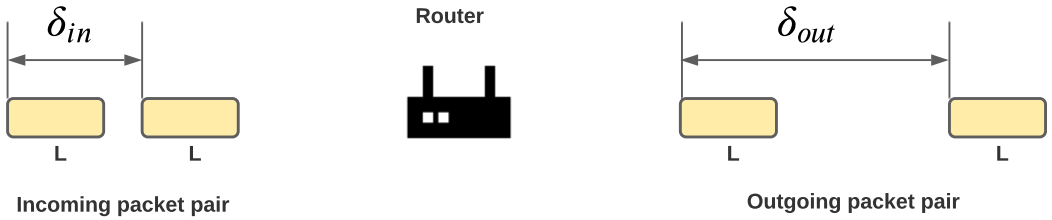


Figure 3.3: Packet pair dispersion

packets get inserted between the probe traffic packets. The dispersion is null if the path is empty of any cross traffic (which is far from realistic in practice). Finally, the negative dispersion is treated as infeasible and thus discarded by available bandwidth techniques. The PGM approaches, such as Spruce [16] and Initial Gap Increase/Packet Transmission Rate (IGI/PTR) [18] send batches of probe packets and measure the inter-arrival time of the consecutive packets. Since the contention with the cross traffic induces queuing delays, the packets will be dispersed in time. This dispersion increases with the load, thus making it possible to infer the available bandwidth. Delphi [17] probes the path with a series of chirp trains. Based on the transmitted and the received inter-packet delays relationship, it can estimate the load induced by the cross traffic.

[16]: Strauss et al. (2003), 'A Measurement Study of Available Bandwidth Estimation Tools'

[18]: Hu et al. (2003), 'Evaluation and characterization of available bandwidth probing techniques'

[17]: Ribeiro et al. (2000), 'Multifractal cross-traffic estimation'

Most of the aforesaid techniques have been designed for classic wired networks. The rise of wireless networks and complex network infrastructures have fostered the development of available bandwidth estimation approaches, e.g., for 3G/4G networks [41] , cloud networks [42] , ad-hoc networks [43, 44] , or SDN networks [45] . The advent of these new tools has motivated some studies, such as [46] to propose an updated summary of the metrics, characteristics, and techniques related to the measurement of the available bandwidth.

[41]: Aceto et al. (2018), 'Available Bandwidth vs. Achievable Throughput Measurements in 4G Mobile Networks'

[42]: Ha et al. (2019), 'A Novel Timestamping Mechanism for Clouds and Its Application on Available Bandwidth Estimation'

[43]: Castellanos et al. (2019), 'Available Bandwidth Estimation for Adaptive Video Streaming in Mobile Ad Hoc'

[44]: Zhao et al. (2009), 'Accurate available bandwidth estimation in IEEE 802.11-based ad hoc networks'

[45]: Megyesi et al. (2017), 'Challenges and solution for measuring available bandwidth in software defined networks'

[46]: Salcedo et al. (2018), 'Available bandwidth estimation tools: Metrics, approach and performance'

The methods developed for wired networks fail when applied to Wi-Fi, in particular, because of the CSMA/CA MAC method that implies complex waiting times, e.g., random backoff, DIFS, etc. The mechanisms used to cope with the fluctuating wireless channel conditions, e.g., interference, fading, and bit error rates need also specific attention: dynamic rate adaptation, Automatic Repeat Request (ARQ).

Available bandwidth estimation in IEEE 802.11 networks

As this dissertation primarily focuses on bandwidth characterization on IEEE 802.11 networks, in this section, we present an overview of the corresponding available bandwidth estimation tools.

Available bandwidth estimation in IEEE 802.11 networks Exact [47] was one of the earliest works that leveraged the MAC layer overhead while estimating the available bandwidth in Wi-Fi networks. By supposing that this overhead is constant for every single packet, Exact varies the packet size to generate various transmission times. It then infers the available

[47]: Shah et al. (2003), *Available Bandwidth Estimation in IEEE 802.11-based Wireless Networks*

bandwidth by removing the constant cost from the MAC layer. As an application case study, EXACT uses the available bandwidth obtained in dynamic bandwidth management for single-hop mobile ad hoc networks and an explicit rate-based flow-control scheme for multi-hop mobile ad hoc networks.

In order to handle the different link behavior on wireless networks, the authors of [48] proposed ProbeGap that considers a more realistic wireless link model for bandwidth characterization that addresses the problems caused by non-First In First Out (FIFO) scheduling and the case where the link supports multiple distinct rates. ProbeGap probes for the idle periods called *gaps* in the link by gathering One Way Delay (OWD) samples and then multiplying by the capacity to obtain an estimation for the available bandwidth.

[48]: Lakshminarayanan et al. (2004), 'Bandwidth Estimation in Broadband Access Networks'

In [49] , the authors proposed IdleGap, which requires running on a real-time system. It infers the idle time fraction, defined as the fraction of time in which the channel is idle and in turn estimates the available bandwidth, based on low layer information, such as the Network Allocation Vector (NAV). Such information is generally not available at the upper layer making this technique difficult to implement in practice. WBest [19] is a PGM tool consisting of two steps. In the first step, it sends pairs of probe packets and infers the effective capacity of the network, defined as the maximum capability of the wireless network to deliver network layer traffic [50] . It then transmits a probe packet train at the estimated rate and infers the available bandwidth based on the dispersion rate.

[49]: Lee et al. (2006), 'Bandwidth Estimation in Wireless Lans for Multimedia Streaming Services'

[19]: Li et al. (2008), 'WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks'

[50]: Li et al. (2006), 'Packet Dispersion in IEEE 802.11 Wireless Networks'

All cited approaches were proposed before the implementation of the frame aggregation scheme by the recent IEEE 802.11 standards, making them ill-suited for deployment. Unfortunately, when using the frame aggregation scheme, the assumption of FIFO per-packet based scheduling does not hold as the aggregated transmissions act as a batch scheduling which dramatically alters the timing properties of received packets. Indeed at the application level, packets aggregated within the same frame are received at the same time. These packets thus appear to have negligible packet gaps between each other. PGM techniques cannot process the negligible gap between packets that were aggregated in the same frame. On the other hand, PRM techniques assume that when passing through a congested link, the packet rate at the reception should be typically less than the probe rate at the emission. Nevertheless, this approach breaks under the frame aggregation. PRM schemes may therefore distort and overestimate or underestimate the received rate.

Aggregation aware available bandwidth estimation Recent works have adapted PGM and PRM techniques to take into account frame aggregation. In [20] , the authors proposed WBest+ that deploys the WBest algorithm with a modified packet rate calculation method by considering aggregated frames as a unique jumbo frame. The available bandwidth estimation is performed based on the time between aggregated frames instead of the time between probe packets. Another solution is proposed by L. Song and A. Striegel [11] , who have proposed Aggregation Intensity based WiFi Characterization (AIWC), aiming at estimating the frame aggregation level at the reception side to capture link congestion and deduce the available bandwidth. To capture the aggregation, AIWC and

[20]: Farshad et al. (2014), 'On the impact of 802.11n frame aggregation on end-to-end available bandwidth estimation'

[11]: Song et al. (2017), 'Leveraging Frame Aggregation for Estimating WiFi Available Bandwidth'

WBest+ rely on a threshold-based method: if the gap time between two consecutive received packets is less than a given threshold, the corresponding packets are deemed to be aggregated. The network scenario used in these papers consists of a server with a wired Internet connection that sends the probing traffic along the network path from the server to the client with a last-hop wireless connection that is assumed to be the bottleneck. The scenario considered in this thesis (as we will see in Chapter 4) is different as the probe traffic is sent from the client to the server. In addition, while the authors of AIWC [11] assume that the first packet that arrives in the AP queue will be held for a pre-defined delay time to wait for more traffic, the authors of [51] suppose that there is no waiting time for forming an A-MPDU frame, and the maximal delay can be set to 1 s to form an A-MSDU. Therefore, the parameterization of these methods is empirical. A fundamental yet challenging task is to perform a formal study of the aggregation behavior as a function of the traffic to settle these types of techniques and prove their accuracy.

[51]: Skordoulis et al. (2008), 'IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs'

3.2 Crowd-sensing for performance evaluation of Wi-Fi networks

Thanks to their ubiquitous nature and built-in sensors, crowd-sensing applications have become an excellent approach to crowd-sense the Wi-Fi performance to actively estimate the end-to-end network performance. The idea of exploiting smartphones to monitor the wireless networks and/or spectrum has been a hot topic in the literature due to the technology advance and the remarkable benefits.

Spectrum Sensing and Monitoring In [52] , the authors proposed an approach that uses smartphones to capture and map heterogeneous networks and devices in home networks. The system periodically carries out measurements and utilizes them to capture new devices, discern the impact of one device on another, etc. The works presented in [53–55] are three crowd-sensing-based Radio Frequency (RF) spectrum monitoring tools using smartphones. In order to perform spectrum measurements, these techniques do not use only smartphones but also an external hardware: a software defined radio in [53] and a frequency translator in [54]. In [55], the authors designed a spectrum discovery tool, where the cloud collects the spectrum sensing information from many smartphones and infers location-specific spectrum availability based on data fusion.

[52]: Nychis et al. (2014), 'Using Your Smartphone to Detect and Map Heterogeneous Networks and Devices in the Home'

[53]: Nika et al. (2014), 'Towards Commoditized Real-Time Spectrum Monitoring'
[54]: Zhang et al. (2015), 'A Wireless Spectrum Analyzer in Your Pocket'
[55]: Lin et al. (2020), 'Crowdsensing for Spectrum Discovery: A Waze-Inspired Design via Smartphone Sensing'

Wireless Measurements The authors of [56] proposed MCNet, a tool based on active smartphone measurements to estimate the user-perceived performance in enterprise wireless networks. Another work [57] based on channel scans was also proposed. It is based on passive measurements while our work investigates the effectiveness of active ones. The authors in [58] proposed a measurement crowd-sensing study in the city of Edinburgh to characterize urban Wi-Fi. It revealed several problems in Wi-Fi deployments in public spaces. However, none of these works involves the network load while crowd-sensing the network performance.

[56]: Rosen et al. (2014), 'MCNet: Crowdsourcing wireless performance measurements through the eyes of mobile devices'

[57]: Shi et al. (2016), 'A walk on the client side: Monitoring enterprise Wifi networks using smartphone channel scans'
[58]: Farshad et al. (2014), 'Urban WiFi characterization via mobile crowdsensing'

3.3 Analytical models for performance evaluation of Wi-Fi networks

At the other end of the literature, there are the modeling approaches that analyze the performance of the MAC Frame aggregation techniques based on analytical models. Hajlaoui et al. proposed a Discrete-Time Markov Chain (DTMC) to model the functioning of the aggregation mechanism and block acknowledgment under the assumption of a binary symmetric channel [59] . Then, they presented an analytical model to evaluate IEEE 802.11n saturation throughput based on the proposed DTMC. A property of this model is that it considers saturated networks. Because the saturation assumption can be deemed too restrictive in some cases, in [60] , Kim et al. proposed an enhanced DTMC to evaluate the impact of A-MPDU and A-MSDU frame aggregation mechanisms on the throughput under unsaturated conditions. However, the objectives of these works differ from ours since in this dissertation we explore how to profit from the frame aggregation in order to infer the network load for vanilla devices. Moreover, the scenarios considered in these papers are not suitable for network load estimation as they require different traffic patterns.

[59]: Hajlaoui et al. (2018), 'An accurate two dimensional Markov chain model for IEEE 802.11n DCF'

[60]: Kim et al. (2008), 'Effect of Frame Aggregation on the Throughput Performance of IEEE 802.11n'

3.4 Wi-Fi access point selection for performance evaluation of Wi-Fi networks

Since their important role in improving Wi-Fi performance, several works of Access Point selection, also known as user association, have been proposed in the literature based on several network metrics.

The authors in [22] proposed the new metric Estimated aVailable bAndwidth (EVA) that passively computes the average transmission time per data unit based. It then uses this metric to choose the AP that provides the maximum achievable throughput among scanned APs. By leveraging the broadcasting nature of WiFi networks, SmartAssoc [23] captures packet transmission rate information by monitoring the network from the client-side. A common property of these methods is that they are based on the passive measurements approach, which is not applicable in our work (See Chapter 4). In [61] , the authors proposed a new method AMPDU-based ap LoAd Mechanism (ALAM), that exploits the characteristics of aggregated frames to infer the expected throughput of a link for the purpose of AP selection. This method acknowledges the throughput information to mobile users via beacon frames. However, this thesis aims at inferring the network load without modifications neither on APs nor on STAs.

[22]: Lee et al. (2008), 'Available Bandwidth-Based Association in IEEE 802.11 Wireless LANs'

[23]: Xu et al. (2013), 'SmartAssoc: Decentralized Access Point Selection Algorithm to Improve Throughput'

[61]: Song et al. (2017), 'Leveraging frame aggregation to improve access point selection'

3.5 Summary

In Table 3.2, we summarize the main characteristics of a selection of the Wi-Fi performance evaluation related works. We highlight the case where the proposed tools take into account the considered criteria.

- We first notice that most of the existing works fall in the category of available bandwidth estimation, and only a few of them tackle the problem of network load estimation in Wi-Fi networks.
- As shown in Table 3.2, only a few works take into consideration the frame aggregation mechanism while optimizing the Wi-Fi performance. Unfortunately, all these works break with this scheme. Consequently, these approaches became irrelevant for our context.
- To the best of our knowledge, these existing works have so far never been applied to the available bandwidth/throughput estimation on vanilla handheld devices, especially smartphones. In fact, in some works [11, 19, 20], the probe traffic is sent from an application server with a wired Internet connection along the network path to a client with a last-hop wireless connection. In addition, the bottleneck constraint is relaxed by assuming that the last Wi-Fi hop is the bottleneck. Nevertheless, this is not the case for many networks, in particular those with a low capacity broadband Internet connection.

Table 3.2: Related work taxonomy with several criteria taken into account

Paper	Metric	Frame aggregation	Type of measurements	Applicable to smartphones
WBest [19]	Available bandwidth	-	Active	-
WBest+ [20]	Available bandwidth	✓	Active	-
AIWC [11]	Available bandwidth	✓	Active	-
Idlegap [49]	Available bandwidth	-	Passive	-
EXACT	Available bandwidth	-	active	-
ALAM [61]	Throughput	✓	Active	-
EVA [22]	Available bandwidth	-	Passive	-
SmartAssoc [23]	Available bandwidth	-	Passive	-

3.6 Discussion and conclusion

The question of interest that we pose in this dissertation is: how could a device choose an access point to associate with based on the expected network performances? To that aim, choosing the appropriate performance evaluation metric is a crucial decision. Not only does the choice affect the kind of expected outcome, but it often characterizes the complexity of the adopted approach. This choice can thereby be a make or break factor.

A STA would much rather pick the best AP according to its expected throughput. Unfortunately, determining the average throughput cannot be easily forecast by a device. Indeed, this value is a function of the AP conditions. The available bandwidth as measured by a given STA depends on many factors, including the antenna and channel gains that are approximately captured by the Received Signal Strength Indicator (RSSI), but also many other hardware or software parameters that condition the modulation and coding schemes it can use. Moreover, the available bandwidth estimation does not necessarily reflect what a STA may actually get as it is difficult to interpret it to accurately indicate the user experience. The network load and the characteristics of the competing traffic are also crucial. In this regard, this thesis considers the estimation of the network load. We propose conceptually simple analytical Markovian models specific to the application of BTF estimation in the presence

of the IEEE 802.11 frame aggregation scheme. We model and simulate scenarios in which a vanilla device infers the mean aggregation level of an aggregated deterministic probe traffic competing with a cross traffic that can aggregate or not its frames. The analysis results are then delivered to our scheme FAM to characterize the channel load and the cross traffic type.

Our aim in the following chapter is to give an overview of the various network load measurement metrics, exhibit the main challenges faced while estimating this metric in a vanilla device, and provide the procedure of detecting the frame aggregation level at the user space. This chapter will also present the network simulation environment and the choice of the programming languages used throughout this thesis, coupled with detailed explanations of the reasoning behind such choices. In addition, we will describe the system that we consider in our modeling approaches.

Implementation System Overview

4

The years of this thesis were essentially focused on answering the following question: how an unmodified vanilla device, in particular a smartphone, could estimate the load of a network in the user space with no requirements from the access points and without root permissions? To answer this question, we intrinsically need to answer other questions: 1) which metric should we take into account to measure the network load? 2) what information can be collected from an Android smartphone and how much could we learn about surrounding Wi-Fi networks by performing channel scans? 3) can this information provide new engineering insights on the feasibility of a network load estimation method for vanilla handheld devices? These are the questions that we set out to answer in this chapter. We first give a description of the network load measurement used metrics. We then detail the various challenges that we encountered with smartphones while estimating the network load of the wireless channel that we believe highlight the complexity of the task. Note that we are only interested in smartphones that use Android as an operating system since it is more open than iOS or Windows. We then introduce the technique used to estimate the level of frame aggregation at the application layer and show the importance of choosing the right network simulator tool when evaluating the Wi-Fi performance. Finally, we dedicate the last part of this chapter to the network load measurement system considered in our contributions (the modeling approaches and the scheme FAM). We provide a high-level description of the corresponding system coupled with explanations of the reasoning behind our choices.

4.1 Technical implementation issues

Before plunging into the details of our system, this section presents multiple key backgrounds by giving an overview of the network load measurement and its related metrics and discusses the challenges faced when adapting this type of measurement in Android smartphones.

Network load measurement

A variety of metrics are used for measuring the Wi-Fi network load. These metrics, which are chosen according to the desired aim, share the same goal that consists of unloading the most loaded APs by balancing the network load between them. In the following, we express the load of an AP in three ways:

- **Load as the number of stations in the BSS**: this approach is meaningful only if we assume that each station has the same data rate, traffic pattern, and, thereby, the same bandwidth requirements. Unfortunately, in practice, this approach is not applicable since the relation between the number of stations per AP and the network

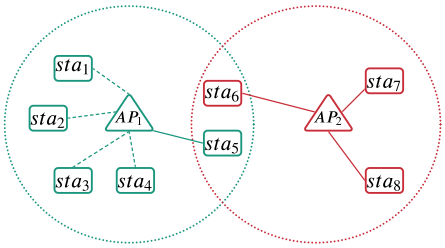


Figure 4.1: Example topology

load is nonlinear. Unsurprisingly, those stations might be actively transmitting or in sleeping mode. Even among active stations, their demands in terms of bandwidth may differ from one station to another. In Figure 4.1, we have two APs, AP_1 and AP_2. The AP_1 has five stations associated with it where STA_5 is actively transmitting (the active transmissions are represented through the solid lines) while STA_1, STA_2, STA_3, and STA_4 are inactive (represented through short dashed lines). On the other hand, the AP_2 has three active associated stations and handles a network load higher than AP_1's load. Based on the number of stations approach, a new STA seeking connection will be automatically associated with AP_2, hence further exacerbating the issue.

- **Load as the number of frames**: this approach is based on the number of frames that an AP can successfully handle per unit of time. The number of frames does not give the real load of an AP due to several reasons:

  1. the packets can have different sizes. Indeed sending ten packets of 1500 bytes is not the same as sending ten packets of 64 bytes;
  2. there are different MCS indexes (different data rates) per STA and, therefore, even the same traffic can have a very different impact depending on the used data rate (which affects the transmission time);
  3. taking all these parameters (MCS indexes for each station, number of frames per STA, number of retransmissions, the use of frame aggregation or not, the activity of the neighboring APs, etc.) into account is a difficult problem. We rather need a single metric that is easy to be forecast and that reflects the real WLAN load.

- **Load as the channel busy time fraction**: this load is defined as the fraction of time the wireless medium is sensed busy due to successful or unsuccessful transmissions. It captures concurrent transmissions and summarizes when the channel is above the CCA, constituting the AP load, as well as inter-network interference. This metric thereby measures the real load, conditions the throughput, and is independent of the transmissions' conditions of the device.

Since metrics such as the number of frames and the number of STAs per AP do not provide an accurate indication of the real AP's load, this work measures the network load (named BTF hereafter) through its BTF or the channel utilization as it seems more appropriate. When relying on some specific hardware, this quantity can be easily obtained.

- Atheros (Qualcomm) wireless chipsets offer direct access to the BTF. The Qualcomm Atheros 802.11ac and 802.11n chipsets support the *spectral scan* mode that can passively measure the radio activity on a channel.
- From a protocol point of view, the BTF may be included in IEEE 802.11k measurement reports [62] . It should be noted that this value is only available to the APs implementing this amendment, and it is not provided to the end-user device.
- It is also possible, under some hardware conditions, for a computer to switch its Wi-Fi card in monitor mode and obtain the BTF.

[62]: IEEE (2008), 'IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs'

Unfortunately, it requires privileged access, and not all drivers are offering this functionality.

In summary, the critical technical obstacle here is that all the aforementioned techniques cannot be applied to vanilla non-rooted Android devices with traditional operating systems. In the next section, we detail Wi-Fi information that one should expect to get from an Android-based smartphone. We then discuss the challenges faced under such a platform.

Challenges of Wi-Fi performance measurement on Android smartphones

Built on Linux, Android embeds various sensors, processors, and memories and provides several interfaces or Application Programming Interface (API) for sending packets, recording timestamps, spontaneously performing channel scans by listening passively to devices' Wi-Fi broadcasts, and collecting the network performance information. In the following, we summarize information that a non-rooted Android smartphone can learn about surrounding Wi-Fi connections using the *android.net.wifi* API when it performs a simple channel scan.

- **Service Set IDentifier (SSID)**: the name of the Wi-Fi network.
- **Encryption type**: describes authentication, key management, and encryption schemes supported by the AP.
- **Frequency**: the Wi-Fi network operates mostly in two frequency bands: the 2.4GHz or the 5GHz band.
- **Received Signal Strength Indicator (RSSI)**: it measures the link quality between a STA and an AP. This metric does not take into account the number of already attached STAs per AP, neither the traffic load on the APs.
- **Link speed**: the data rate used by the device.

During the one minute it takes to read this paragraph, over three billion Android-based devices worldwide will naturally perform Wi-Fi channel scans that record the information of nearby Wi-Fi APs listed above. While considered as a client-side network measurement tool, this approach tends to be poor and useless since it does not provide the potential for network load characterization. In order to get more Wi-Fi network information under the Android platform, the smartphone could switch its low-power Wi-Fi adapter in monitor mode and use a dedicated packet capture utility. However, this action requires root permission, which needs risky and warranty breaking manipulations. Besides, not all smartphones chipsets can support monitor mode. Therefore, we need to develop a new efficient BTF measurement technique for non-rooted smartphones.

4.2 Our methodology

The goal of this thesis is to address earlier issues by exploring new alternatives in order to evaluate the possibility for a vanilla Wi-Fi client, typically a smartphone, to infer the Wi-Fi network load from local

measurements in the user space. To circumvent the aforesaid limitations, we consider active probing of the network, which has been widely used in the literature to estimate network loads (cf. Chapter 3). In this thesis, we propose relatively simple and versatile analytical Markovian models specific to the application of BTF estimation in the presence of the IEEE 802.11 frame aggregation scheme. We explore how the usage of such a scheme can be exploited to properly discern not only the BTF of an AP but also to accurately convey the type of traffic. In this regard, we model and simulate scenarios in which a vanilla device induces the mean aggregation level of an aggregated deterministic probe traffic competing with cross traffic that can aggregate or not its frames in the user space. The analysis results are then delivered to our scheme FAM to characterize the channel load and the cross traffic type. In the following, we first detail how the proposed method estimates the ground truth of the BTF which serves as a reference for the proposal. We then discuss the choice of the programming language used to implement the probing sender application for Android devices, and we present how the method detects the frame aggregation levels at the user application level. We then discuss the choice of the network simulator environment.

Measuring the ground truth value of the BTF

Serving as a reference for our scheme FAM, we here give the computation details of the ground truth BTF. For doing so, we used a computer (Sniffer) with a specific Wireless Network Interface Controller (WNIC) that supports the *survey dump* feature of the *iw* command which is a Linux utility that shows the survey information of all the available channels including the *channel busy time* and *channel active time*. We thereby measure the ground truth value of the BTF as follows.

$$BTF = \frac{channel\ busy\ time}{channel\ active\ time}$$

It is worth noting that not all drivers support this feature.

Choosing the right programming language in Android

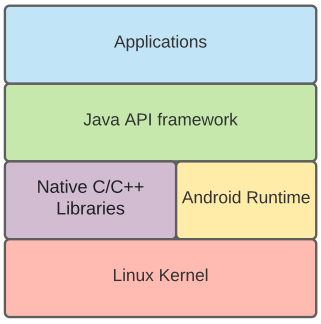


Figure 4.2: Android software stack

As this dissertation primarily focuses on the network load estimation on Android smartphones, in this section we briefly describe the Android platform. We then highlight the benefit of using native applications to build sensitive-time network measurement applications.

Google Android is an open-source software stack that includes the Linux operating system, middleware, and applications. Although built on Linux, the platform Android differs from other Linux distributions by putting a Java interpreter and runtime environment called Android RunTime (ART). The diagram in Figure 4.2 shows the major components of this platform.

For developing Android applications, Google provides two kits:

- Android Software Development Kit (SDK): released in December 2007, the SDK includes the necessary tools and libraries to develop and run Android applications using Java or Kotlin.

- Android Native Development Kit (NDK): released in June 2009, the NDK allows developers to embed native codes written in C/C++ and assembly language in Android applications.

Choosing the right kit and the adequate programming language for time-critical applications requires careful consideration. While most of the Android applications are willfully Java programs, C/C++ programming is possible too thanks to the NDK. In order to call the functions in the native library, the Java Native Interface (JNI) is used. Since Java is comparatively slow and it is not a good programming language for handling network measurement, many works propose to use C/JNI applications. The authors of [63] compared the performance between native C and Java applications under the same tasks, and proved that native C applications can be up to 30 times faster than running Java in Dalvik Virtual Machine (DVM) (DVM is the runtime system used on Android devices running versions below 5.0 and has since been replaced with ART). The papers [64, 65] studied the delay overhead in 802.11 networks and recommended using native C implementation in order to mitigate the user level overhead. In this thesis, we therefore delegate the sending of the probe traffic task from Java to native libraries written using the C programming language.

[63]: Batyuk et al. (2009), 'Developing and Benchmarking Native Linux Applications on Android'

[64]: Li et al. (2015), 'On the accuracy of smartphone-based mobile network measurement'
[65]: Li et al. (2018), 'Toward Accurate Network Delay Measurement on Android Phones'

Detecting the frame aggregation levels at the application level

In order to get the aggregation level ground truth in the experiments conducted throughout this thesis, we used the wireless packet capturing method for its easy deployment and low cost. This latter is a simple technique that passively monitors the wireless network traffic by listening to the WNIC of the device. We used a computer configured into monitor mode to capture the aggregated frames by using the dedicated capture software *Wireshark*. The aggregation level is then computed according to the A-MPDU reference number in the radiotap header (additional information added by the wireless adapter or its driver). We also disable the security options (e.g., WPA, WEP) to simplify the decoding of the frames.

We note that since the final targeted application cannot use this approach, our technique will rely on a threshold-based method as a workaround. This latter computes the inter-arrival time between two packets at the application layer. If the gap time between two consecutive received packets is less than a given threshold, the corresponding packets are deemed to be aggregated. The rationale behind this approach is that if probe packets are aggregated, they are sent over the Wi-Fi link as part of the aggregated frame and not as individual probe packets. At the reception, these probes are decapsulated from a frame, thereby making it challenging to classify which probes belonged to which frame. Based on the fact that aggregated probe packets tend to have a small interval between them, we can reconstruct aggregated frames and count the number of probe packets of each frame. It should be borne in mind that this technique is used by much of the prior literature [11, 20] . To assess the effectiveness of this approach, we conducted the following experiment.

[20]: Farshad et al. (2014), 'On the impact of 802.11n frame aggregation on end-to-end available bandwidth estimation'
[11]: Song et al. (2017), 'Leveraging Frame Aggregation for Estimating WiFi Available Bandwidth'

Experimental setup

To verify this approach, we conducted the following experiment. Our experimental environment (Figure 4.3) was set up as follows. The probing traffic sender application, executed on an Android smartphone, was written using C (by using the Android NDK) and Kotlin programming languages. It sends a set of UDP probe packets to the server. The server's source code, which receives the probes, records their reception timestamps, and infers the corresponding aggregation level, was written in C and executed on a laptop running Ubuntu 18.04. The client, as well as the server, were connected to an 802.11 Linksys LAPAC1750 AP. The sniffer was configured in monitor mode and running the packet capture utility *Wireshark* to capture the wireless frames forwarded by the AP to the server. We ran this experiment at noon at a residence with all the real-world wireless interference.

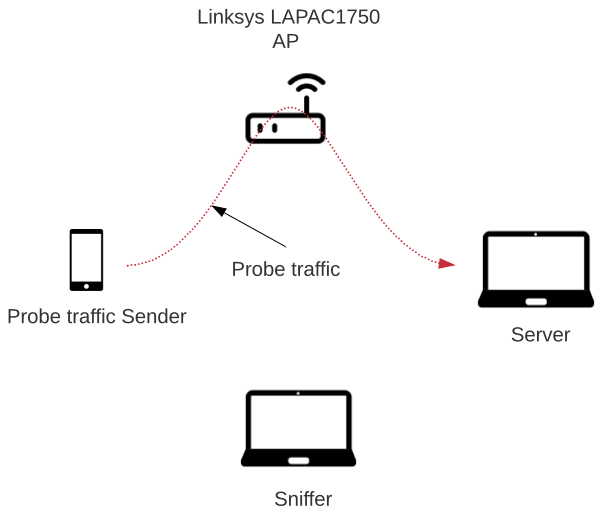
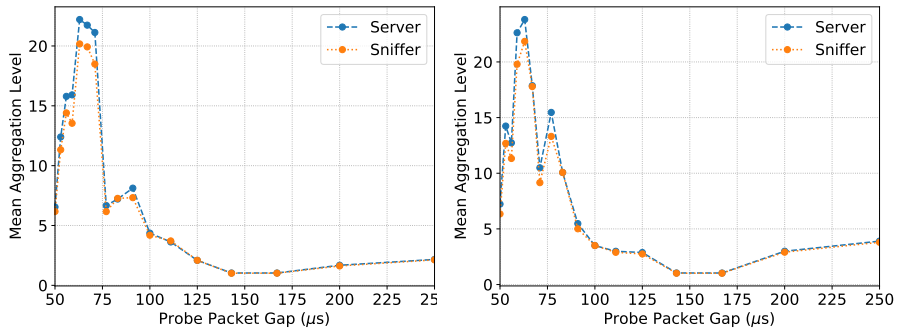


Figure 4.3: Experimental setup

Numerical results

In Figure 4.4, we compare the mean aggregation levels computed by the server with those captured by the sniffer. Several experiments have been performed with equivalent results, the figure shows two of them. We tested several thresholds. For these experiments, a value of 250 μs gives inferred aggregation levels that match well with the ones observed by the sniffer. It is noteworthy that this threshold probably depends on the system used.



(a) Experiment 1

(b) Experiment 2

Figure 4.4: The mean aggregation levels captured by the server versus the sniffer

Choosing the network simulator

Having already described the procedure to detect the frame aggregation level at the application layer and how to measure the ground truth BTF in practice, we now set the IEEE 802.11 network simulation environment used throughout this thesis.

In the domain of networks, multiple simulation tools, free or not, have been developed in order to mimic the behavior of networks by re-implementing sophisticated network stacks, from the PHY layer up to the application layer. The simulation tool is practical since one can design any network scenario close to real-life while freely adjusting the parameters. In this regard, choosing the right simulator tool for evaluating the performance of our approach is a crucial task that needs careful consideration.

We resort to using discrete-event simulators that are considered capable of rendering results close to real-life scenarios. The ns-2 discrete event simulator [66] had been the network simulation tool of choice for academic research in networks until the development of ns-3 [67] discrete event simulator in July 2006. Built using C++ and Python, ns-3 quickly took the place of ns-2 and has become one of the leading network simulators. Compared to its predecessor ns-2, it includes multiple additional features allowing to model additional wireless technologies.

[66]: (2010), *The Network Simulator ns-2, https://www.isi.edu/nsnam/ns/*

[67]: (2021), *The Network Simulator ns-3, https://www.nsnam.org/*

In this thesis, we opted for the ns-3. This choice is driven by the fact that the latter is the de facto standard for the simulation of Wi-Fi networks. Indeed, ns-3 is an open-source network simulator that implements all the basic schemes of the IEEE 802.11 standard and the DCF method. Since its first release in June 2008, this tool actively kept on developing by adding all newer standard amendments and their features, e.g., the 802.11ax standard amendment including HE MCS indexes. In this manuscript, we used ns-3.30 since it was the latest version at the time of setting up our simulations. The simulation under ns-3 relies on several existing modules. Users can configure the network either by using those modules or by expanding them to cover non-supported features if needed. Focusing on the Wi-Fi module, both the MAC and the PHY are modeled, up to the 802.11ax amendment, including all the modes (ad-hoc, infrastructure, and mesh modes).

4.3 System architecture

Now that we have a way to compute the aggregation level at the user application level and a way to compute the ground truth BTF, in this section, we describe the system architecture used in our contributions (the analytical models and the scheme FAM).

The system we consider is a general WLAN that uses the IEEE 802.11 Distributed Coordination Function (DCF). Figure 4.5 depicts the overall architecture. We distinguish two different entities: an AP and user STAs that are assumed to be covered by the AP and can be either user stations or servers.

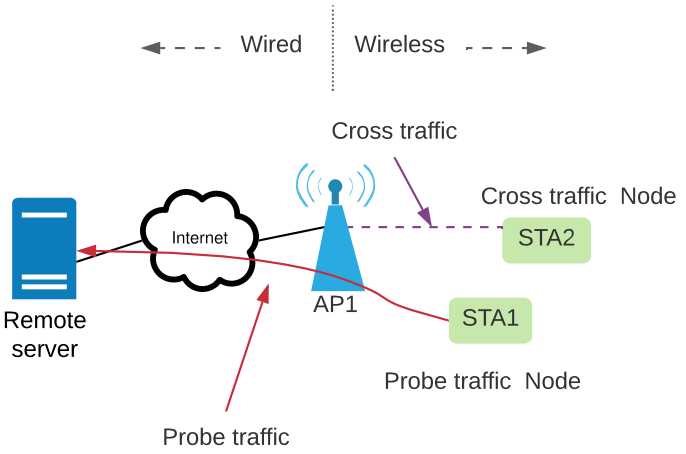


Figure 4.5: Architecture scenario 1

The user device is denoted as the probe traffic node. It associates with $AP1$ and sends probe traffic to the server. We present three hypotheses on the server's position inside the network's topology.

Figure 4.5 presents a scenario similar to classical "Speed Test" applications where the server is located outside the local networks of the Wi-Fi APs. However, the characteristics of the connection within the Internet between the AP and the server are too complex to capture and model. We, therefore, introduce the scenario in Figure 4.6 in which we assume that the connection between the AP and the server is ideal. We model it as if the server were implemented on the AP.

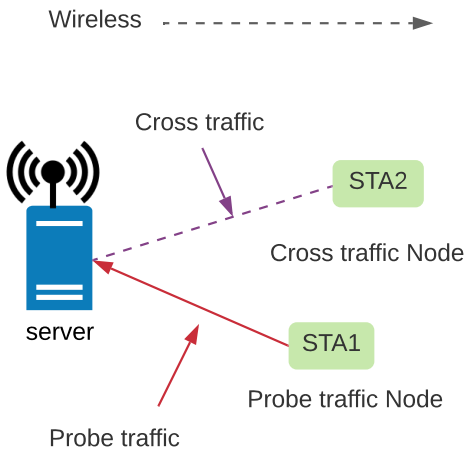


Figure 4.6: Architecture scenario 2

The scenario of Figure 4.7 is a more realistic and practical one. We consider that the server is embedded on a second wireless device owned by the user and connected to the same AP through the same Wi-Fi network. The task of deploying the first scenario proves to be very challenging and complicated. In this dissertation, we therefore model, simulate, and implement the second and third scenarios.

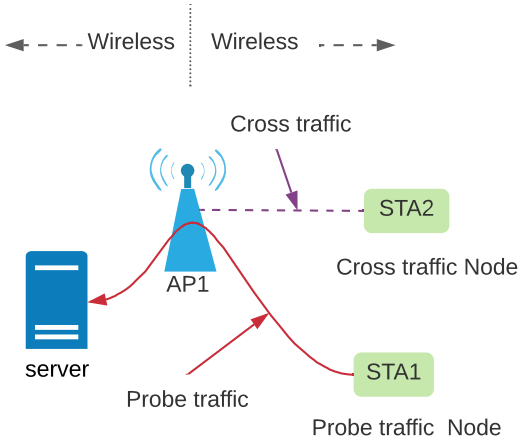


Figure 4.7: Architecture scenario 3

The probe traffic node and the server run an application that aims to infer the two following information.

- The load of the channel: this load can be generated by the traffic from $AP1$ or other APs/stations (STAs) using the same channel. The load is defined here as the BTF.
- The type of traffic: do other nodes aggregate their frames or not?

The traffic carried by the network is denoted the *cross traffic*. We consider that the DownLink (DL) cross traffic is the predominant compared to

[68]: Gupta et al. (2012), 'WiFox: Scaling WiFi Performance for Large Audience Environments'

UpLink (UL) traffic. The amount of UL flow is considered negligible, or at least not significant, as opposed to DL traffic [68] .

When the probing application runs, it generates supplementary traffic denoted the *probe traffic*. It is UpLink traffic from the probe node to the AP. In the wireless server scenario, it is also DownLink.

To perform the BTF estimation and determine the type of concurrent traffic, the probe traffic node sends a sequence of small probe packets to the server with an increasing inter-packet arrival using the UDP. The probing node is assumed to aggregate frames and thus uses a recent IEEE 802.11 amendment (since IEEE 802.11n). Upon receiving the probe packets, the server analyzes them with the algorithms given in Chapter 6 and deduces the BTF and the presence of aggregation. This estimation relies on an analytical modeling approach described in Chapter 5. Based on the scenario used, we propose two models. The first model, called *Ideal server*, simulates the simplified case where the connection between the AP and the server is ideal and therefore the server is modeled as if it were implemented on the AP or close to the AP. The second model is called *Wireless server* and captures the case where the server is embedded on a device associated with the AP. The frame aggregation level is computed using the corresponding model in each case.

4.4 Conclusion

This chapter has dealt with the technical implementation system coupled with a presentation of the main obstacles and challenges encountered while measuring the network load through its BTF for vanilla hand-held devices, especially Android-based devices. We exposed the system architecture scenarios adopted throughout this thesis and presented the solutions to overcome the different faced issues. In doing so, we first showed the importance of choosing the right performance metric when optimizing the network performance, we identified the pitfalls of BTF measurement in practice, and we provided the information that an Android smartphone can learn from a channel scan on a Wi-Fi network. We then presented our methodology by presenting the method used to measure the ground truth of the BTF, introducing the network simulation tool ns-3, and highlighting that is possible to infer the frame aggregation level at the device's user application level by proposing a simple feasible application. At last, we presented the system architecture used by our proposed analytical models as well as the method FAM.

In brief, this chapter sets out the technical implementation details. Based on several analytical models, the system provides the BTF estimation of the Wi-Fi network for an Android-based device. In the next chapter, we will describe the proposed Markovian models, their resolution, and their numerical validation.

Analytical Study of Frame Aggregation Level

5

Having already described the overall architecture system and the scenarios considered throughout this thesis in the previous chapter, we now try to solve the following question: how can we develop an analytical model that helps infer the frame aggregation behavior as a function of the competing traffic in WLANs? To that aim, the main objective of this chapter is to propose analytical models and validate them. In order to strengthen the study, analysis of performance based on several simulations and an experimental test-bed are proposed.

This chapter is organized as follows. The system model, common assumptions, and parameters are presented in Section 5.1. Section 5.2 details the model *Ideal server* and Section 5.3 details the mathematical formulation of the model *Wireless server*. Validation and comparison of the proposed models are then exposed in Section 5.4 by exposing the advantages and drawbacks of each solution. Conclusions are drawn in Section 5.5. Finally, some perspectives are exposed in Section 5.6.

5.1 System model

In this section, we describe the system considered for the proposed models coupled with a presentation of the common assumptions as well as the system notations.

We consider a general IEEE 802.11 infrastructure WLAN based on the DCF access method composed of two different entities: an AP and user STAs that are assumed to be covered by the AP and can be either user stations or servers. Each station or AP has its own physical transmission rate (MCS index).

The two proposed analytical models are Discrete-Time Markov Chain (DTMC). They evaluate the aggregation levels of the probe traffic for a given cross traffic load in congested and non-congested networks. While the first model (*Ideal server*) estimates the probe aggregation level of the UL traffic sent by the probe node, the second model (*Wireless server*) appraises the aggregation level of the DL traffic forwarded by the AP to the server.

As the probe frame aggregation level depends on the nature of cross traffic (aggregated or non-aggregated traffic), each model relies on two Markov chains. The first chain considers that the cross traffic uses the frame aggregation scheme, whereas the second is based on non-aggregated cross traffic. Table 5.1 summarizes the principal notations used in the models and provides a listing of the used IEEE 802.11 parameters.

Our prediction models rely on the following assumptions:

Table 5.1: Principal notations.

Parameter (unit)	Definition
T_{DIFS} (μs)	Distributed Inter Frame Space duration
T_{SIFS} (μs)	Short Inter Frame Space duration
T_{PHY} (μs)	Preamble and physical header duration
FCS ($Bytes$)	Frame Check Sequence
$T_{BlockACK}$ (μs)	Required time to send the block acknowledgment
T_{ACK} (μs)	Required time to send the acknowledgment
$T_{backoff}$ (μs)	Average backoff time
T_{slot} (μs)	Slot time
CW_{min}	Minimum size of the contention window
d_c (μs)	Inter-arrival time of cross traffic packets
d_p (μs)	Inter-arrival time of probe traffic packets
$AMPDU_{AP}$	Maximum A-MPDU size for the AP
$AMPDU_P$	Maximum A-MPDU size for the probe traffic node
$\mathbb{1}_{condition}$	Indicator function that equals to 1 if $condition$ is true and 0 otherwise
Ideal server model	
$P_{(i,j)(l,m)}$	Transition probability from state (l, m) to state (i, j)
$f(k)$ (μs)	Required time to send k probe traffic aggregated sub-frames
$g(k)$ (μs)	Required time to send k cross traffic aggregated sub-frames
h (μs)	Required time to send a single cross traffic frame
Wireless server model	
$P_{(i,j,k,u)(l,m,q,v)}$	Transition probability from state (i, j, k, u) to state (l, m, q, v)
$T_{AP}(k)$ (μs)	Required time to send k DL probe aggregated sub-frames by the AP
$T_{SP}(k)$ (μs)	Required time to send k UL probe aggregated sub-frames by the probe traffic node
$T_{AC}(k)$ (μs)	Required time to send k DL cross traffic aggregated sub-frames by the AP
T_{AC} (μs)	Required time to send a single DL cross traffic frame by the AP

- **probe traffic**: it is a Constant Bit Rate (CBR) traffic, sent at regular interval d_p. Frame aggregation scheme is always enabled for this flow;
- **cross traffic**: the cross traffic is modeled by a CBR source sending packets at regular interval d_c and managed by a unique queue. The cross traffic can be either aggregated or not. This flow is coming from the distribution system (a wired network connected to the AP, which is not represented in Figure 4.6 and Figure 4.7 (Chapter 4)) and sent to the cross traffic node;
- **buffers**: the probe traffic node, the cross traffic node, and the AP buffers are assumed to have a finite size. More precisely, we assume that when an aggregated frame is sent for a given destination, the corresponding buffer becomes empty for this destination.

The impact of these assumptions is evaluated through simulations and test-bed experiments and discussed in Section 5.4.

In building the four Markov chains, we first present the set of possible states and transitions and compute the transitions probabilities for each chain. We then detail the calculation of the stationary probabilities and the mean aggregation levels of the probe traffic for all the proposed models.

5.2 Ideal server model

Let us first consider the *Ideal server* model that models the scenario of Figure 4.6. There are three nodes: an AP, and two user stations. The first sends the probing packets to the server located on the AP. The second receives the cross traffic forwarded by the AP. As mentioned earlier, we propose two Markov chains for each model. The first model is referred to as *Ideal server model based on aggregated cross traffic* and the second one as *Ideal server based on non-aggregated cross traffic*. This scenario permits us to evaluate the aggregation level of the UL probe traffic sent by the probe traffic node.

Ideal server model based on aggregated cross traffic

In this section, we describe the first Markov chain of the *Ideal server* model where the frame aggregation mechanism is enabled for the cross traffic.

We consider the Markov chain defined as the couple $(X_n, Y_n)_{n\geq 0}$. The process X_n describes the number of aggregated sub-frames contained in the n^{th} transmitted probe frame, while the process Y_n represents the number of packets at the cross traffic buffer at the moment of the n^{th} probe frame transmission. The set of all possible states is $\{0, ..., AMPDU_P\}$ for the X_n process and $\{0, ..., AMPDU_{AP}\}$ for the Y_n process.

The transition probabilities are fully determined by the time between two consecutive probe traffic transmissions. As both probe and cross traffics are deterministic, this time sets the number of packets that arrived in the two buffers between two transmissions and thus the number of frames that will be sent in the aggregated frame. Consequently, we analyze the events that may occur between two probe traffic transmissions. Figure 5.1 shows an example of the possible events between two probe traffic transmissions. Let assume that the current state of the Markov chain at step n is (l, m), i.e. $(X_n = l, Y_n = m)$. First, the probe traffic frame is sent. The transmission duration is denoted $f(l)$. Note that if $l > 1$, it is an A-MPDU frame that contains l aggregated packets. It is important to note that $f(l)$ counts the time to access the medium (composed of the DIFS, and the mean backoff estimated as $\frac{CWmin \cdot T_{slot}}{2}$), the physical header (PHY Overhead), the MAC header, the payload, the FCS, the SIFS, and the ACK or BlockACK.

We get:

$$\begin{aligned} f(l) = {} & T_{DIFS} + T_{backoff} + T_{PHY} + T_{SIFS} + T_{Block\ ACK} \\ & + \frac{1}{frequency} \cdot T_{Block\ ACK\ Request} \\ & + \frac{(MPDUDelimiter + MacHeader + Payload + FCS) \times 8 \times l}{Physical\ transmission\ rate} \end{aligned} \tag{5.1}$$

where $T_{Block\ ACK\ Request}$ means that a response is requested upon transmission of a frame whose sequence number is distant at least by a given threshold multiplied by the transmit window size from the starting sequence number of the transmit window. We compute the frequency

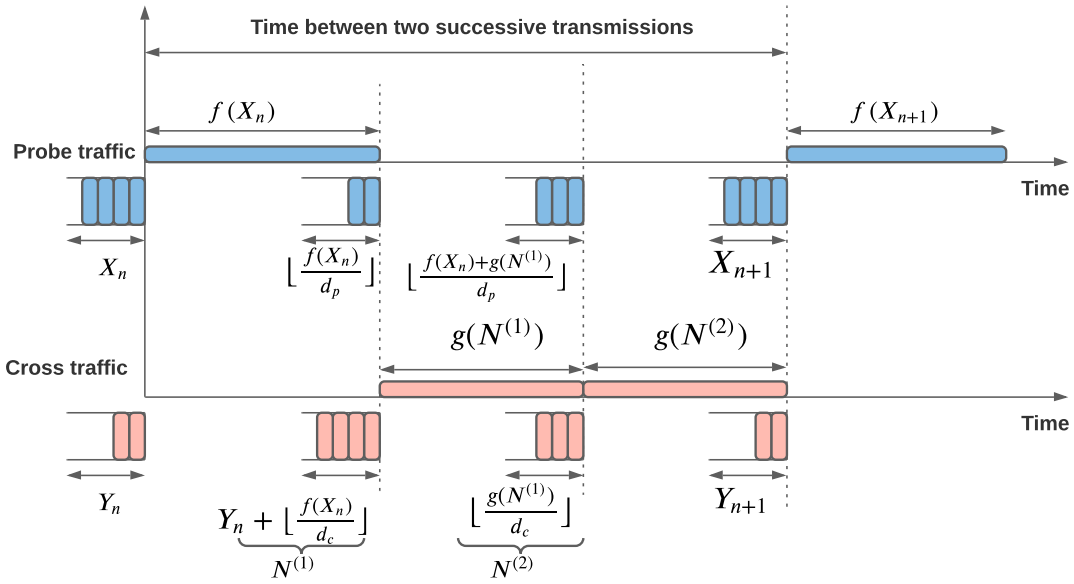


Figure 5.1: Possible events between two successive probe traffic transmissions for the *Ideal server* model based on aggregated cross traffic. At the step n, the n^{th} probe frame is transmitted. It contains X_n sub-frames. Its duration is $f(X_n)$. The competing station accesses the medium to transmit the previous data at its buffer (Y_n) plus the packets that arrived during the period $f(X_n)$. The amount of time to send those packets is given by $g\left(\left\lfloor Y_n + \frac{f(X_n)}{dc} \right\rfloor\right)$. In this case, between the transmission of the X_n and the X_{n+1} frames, the cross traffic succeeds to access the medium two successive times.

of sending of these frames and we add its duration to the frame duration. Note that $T_{Block\ ACK}$ and $T_{Block\ ACK\ Request}$ also count their physical header.

During this transmission, the number of received packets that arrived in the two buffers can be approximated by $\lfloor \frac{f(l)}{d_p} \rfloor$ and $\lfloor \frac{f(l)}{d_c} \rfloor$ for probe and cross traffic respectively (we round down these values to the nearest integers). At the end of this transmission, the probe traffic buffer contains $\lfloor \frac{f(l)}{d_p} \rfloor$ packets, and the AP buffer contains $N^{(1)} = m + \lfloor \frac{f(l)}{d_c} \rfloor$ cross traffic packets.

Before the next probe transmission, several successive transmissions of cross traffic may occur. Let $N^{(k)}$ be the number of packets in the cross traffic buffer at the time when the cross traffic tries to access the medium for the k^{th} time. $N^{(1)}$ has already been computed and corresponds to the buffer size at the end of the probe traffic transmission. If it succeeds to access the medium (assuming that $N^{(1)} > 0$), a frame or aggregated frame composed of $N^{(1)}$ packets is sent. During this transmission, $N^{(2)}$ packets arrived in the cross traffic buffer with:

$$N^{(2)} = \left\lfloor \frac{g(N^{(1)})}{d_c} \right\rfloor \tag{5.2}$$

The function $g(x)$ is the duration of the transmission of a frame ($x = 1$) or an aggregated frame (with $x > 1$ sub-frames). The only difference

with $f(.)$ is the physical transmission rate and the packet size that can be different from the probe traffic.

More generally, for $k > 1$, we get:

$$N^{(k)} = \left\lfloor \frac{g(N^{(k-1)})}{d_c} \right\rfloor \tag{5.3}$$

Now, we compute the probability that k cross traffic frames are sent successively. It is denoted $\mathbb{P}\Big(Q(l,m) = k\Big)$ where m and l denote the buffer states as in the previous equations and $Q(l,m)$ is the number of successive times that the cross traffic accesses to the medium. $k = 0$ means that the cross traffic does not access to the medium between two successive probe traffic transmissions. It can be due to an empty buffer or because the probe traffic wins access to the medium. We denote $p(k)$ the probability for the cross traffic to access the medium at least k successive times given that probe and cross traffics have non-empty buffers. This probability depends on the contention window and k. We get:

$$\mathbb{P}\Big(Q(l,m) = 0\Big) = \mathbb{1}_{m+\frac{f(l)}{dc}<1} + (1 - p(1)) \cdot \mathbb{1}_{m+\frac{f(l)}{dc}\geq 1} \tag{5.4}$$

For $k > 0$, we get,

$$\mathbb{P}\Big(Q(l,m) = k\Big) = \prod_{q=1}^{k} \mathbb{1}_{N^{(q)}\geq 1} \cdot \Big(p(k)\mathbb{1}_{N^{(k+1)}=0} + (p(k) - p(k+1))\mathbb{1}_{N^{(k+1)}>0}\Big) \tag{5.5}$$

In this equation, the product corresponds to the probability that the cross traffic has a non-empty buffer during each of the k successive transmissions. The term in brackets describes the probability that the cross traffic does not access the medium after its k^{th} transmission, either because it loses access to the medium at the $k+1$ accesses when competing with the probe traffic (where the term $(p(k) - p(k+1))$ is a factor) or because of an empty buffer. In the numerical results section, we take $p(k) = \frac{1}{2^k}$, but a more complex formula can be considered instead.

To obtain the transition probabilities, we condition by the number of cross traffic accesses and their transmission times. As the probe traffic is CBR, the number of frames in the probe traffic buffer is directly deduced from this time. For $i \geq 2$ we obtain:

$$P_{(i,j)(l,m)} = \sum_{k=0}^{\infty} \mathbb{P}\Big(Q(l,m) = k\Big) \cdot \mathbb{1}_{d_p \cdot i \leq f(l) + \sum_{q=1}^{k} g(N^{(q)}) < d_p \cdot (i+1)} \cdot \mathbb{1}_{N^{(k+1)}=j} \tag{5.6}$$

For $i = 1$ we get:

$$P_{(i,j)(l,m)} = \sum_{k=0}^{\infty} \mathbb{P}\Big(Q(l,m) = k\Big) \cdot \mathbb{1}_{0 \leq f(l) + \sum_{q=1}^{k} g(N^{(q)}) < 2d_p} \cdot \mathbb{1}_{N^{(k+1)}=j} \tag{5.7}$$

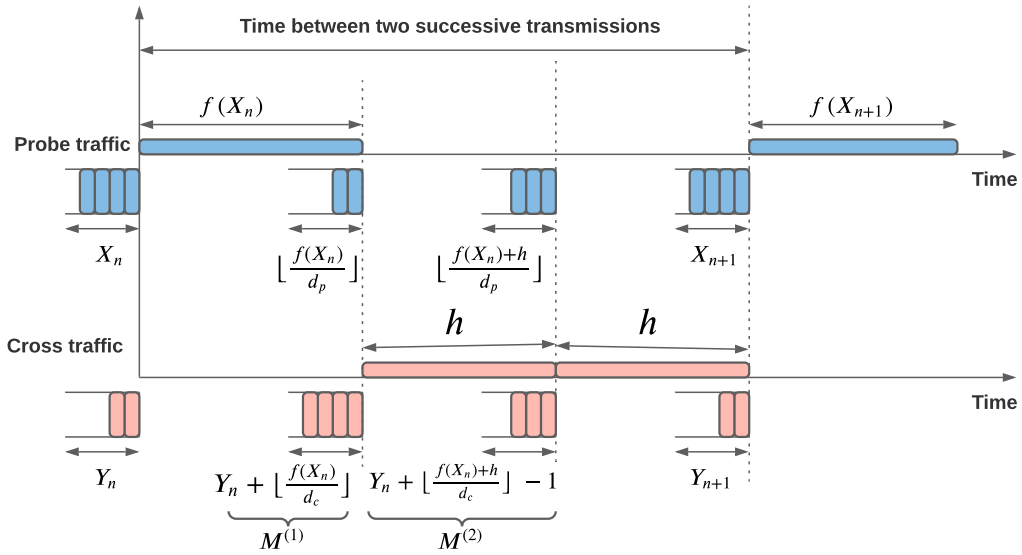


Figure 5.2: Example timeline of possible events between two successive probe transmissions for the *Ideal server* model based on non-aggregated cross traffic.

Ideal server model based on non-aggregated cross traffic

We now describe the second Markov chain of the *Ideal server* model where the frame aggregation mechanism is disabled for the cross traffic.

The principle of this second model is the same: we condition by the number of successive cross traffic transmissions except that each transmission consists only of a single frame. For the sake of clarity, we recourse to Figure 5.2 to describe the execution steps of the proposed model. It shows an example of possible events between two successive probe traffic transmissions when the aggregated probe traffic and the non-aggregated cross traffic are competing for the channel resource.

In this example, at time 0, we start from a state (l, m), i.e., $(X_n = l, Y_n = m)$ $((4, 2)$ in Figure 5.2). By construction, a transition begins by a probe transmission. It sends the l frames currently in its buffer, aggregating them in a unique A-MPDU (4 in our example) using a specific data rate. Its transmission duration is also $f(l)$.

During this transmission, the probe buffer receives $\lfloor \frac{f(l)}{d_p} \rfloor$ packets and the cross traffic buffer receives $\lfloor \frac{f(l)}{d_c} \rfloor$. Consequently, at the end of this transmission, the probe and cross buffers will contain respectively, $\lfloor \frac{f(l)}{d_p} \rfloor$ and $m + \lfloor \frac{f(l)}{d_c} \rfloor$ packets (2 and 4 in our example).

Before the next probe frame transmission, an arbitrary number k of successive transmissions of cross traffic can occur. In our example, the cross traffic gains two successive medium accesses. We denote $M^{(k)}$ the number of frames in the cross traffic buffer before its k^{th} transmission.

$M^{(1)}$ is equal to $m + \lfloor \frac{f(l)}{d_c} \rfloor$. As cross traffic is not aggregated, a single frame among the $M^{(1)}$ is sent. We denote this frame duration by h. It

differs from $g(.)$ as the local properties (transmission rates, packet sizes) and the protocol is different (IEEE 802.11g, for instance) for the cross traffic. For the non-aggregated cross traffic, the transmission duration is expressed as:

$$h = T_{DIFS} + T_{backoff} + T_{PHY} + T_{SIFS} + T_{ACK} + \frac{(MacHeader + Payload + FCS) \times 8}{Physical\ transmission\ rate} \tag{5.8}$$

At the end of the first cross traffic transmission, the cross buffer contains $M^{(2)}$ frames which can be calculated by $M^{(1)} - 1 + \lfloor \frac{h}{d_c} \rfloor$. Generalizing for $k > 1$, we get:

$$M^{(k)} = M^{(k-1)} - 1 + \left\lfloor \frac{h}{d_c} \right\rfloor \tag{5.9}$$

Similarly to our previous Markov chain, we define $Q(l, m)$ the random variable that describes the number of consecutive times that the cross traffic gains access to the medium. Once again, we denote by $p(k)$ the probability for the cross traffic to access the medium at least k successive times given that probe and cross traffics have non-empty buffers.

We distinguish the case where $Q(m, l) = 0$:

$$\mathbb{P}\Big(Q(l, m) = 0\Big) = \mathbb{1}_{m + \frac{f(l)}{dc} < 1} + (1 - p(1)) \cdot \mathbb{1}_{m + \frac{f(l)}{dc} \geq 1} \tag{5.10}$$

The first term corresponds to the case where the cross traffic buffer is empty, and the second term where the cross node loses access to the medium at its first attempt.

For $k > 0$, we get,

$$\mathbb{P}\Big(Q(l, m) = k\Big) = \prod_{q=1}^{k} \mathbb{1}_{M^{(q)} \geq 1} \cdot \Big(p(k)\mathbb{1}_{M^{(k+1)}=0} + (p(k) - p(k+1))\mathbb{1}_{M^{(k+1)}>0}\Big) \tag{5.11}$$

Analogously to the previous Markov chain, we take $p(k) = \frac{1}{2^k}$ in the numerical results section.

We are now able to calculate the transition probabilities based on the number of cross traffic accesses and their transmission times. As the probe traffic is deterministic, the number of frames at the probe traffic buffer is thereby fully determined from this time.

For $i \geq 2$, we get:

$$P_{(i,j)(l,m)} = \sum_{k=0}^{\infty} \mathbb{P}\Big(Q(l, m) = k\Big) \cdot \mathbb{1}_{d_p \cdot i \leq f(l) + \sum_{q=1}^{k} h < d_p \cdot (i+1)} \cdot \mathbb{1}_{M^{(k+1)}=j} \tag{5.12}$$

For $i = 1$, we get:

$$P_{(i,j)(l,m)} = \sum_{k=0}^{\infty} \mathbb{P}\Big(Q(l, m) = k\Big) \cdot \mathbb{1}_{0 \leq f(l) + \sum_{q=1}^{k} h < 2d_p} \cdot \mathbb{1}_{M^{(k+1)}=j} \tag{5.13}$$

5.3 Wireless server model

We now move to the description of the second generation of models named *Wireless server* models. In the *Wireless server* model, the server is embedded on an additional wireless device associated with the AP, as depicted in Figure 4.7. In this section, we extend the previous model to deal with real-world implementation constraints. The paradigm shifts from the evaluation of the aggregation levels of the UL probe traffic to the aggregation levels of the DL probe traffic forwarded by the AP to the server. Two Markov chains are proposed for this model.

Wireless server model based on aggregated cross traffic

In the following, we present the *Wireless server* model based on aggregated cross traffic.

We consider the Markov chain defined as $(X_n, Y_n, Z_n, S_n)_{n\geq 0}$. The process X_n describes the number of probe sub-frames contained in the AP queue at the moment of the n^{th} frame transmission departure. The possible states are $\{0, .., AMPDU_{AP}\}$. The process Y_n defines the number of cross traffic sub-frames in the AP queue. The possible states are $\{0, .., AMPDU_{AP}\}$. Let Z_n be the number of packets at the probe traffic node. It takes its values in the set $\{0, .., AMPDU_P\}$. Finally, S_n describes the n^{th} transmission. It takes three possible values $\{APC, APP, SP\}$. APP (Access Point Probe) is a DL transmission of Probe traffic from the AP. APC (Access Point Cross) denotes a cross traffic transmission from the AP to the cross server, while SP (Station Probe) corresponds to a UL transmission from the probe traffic node.

Transition probabilities

Having defined the set of possible states for each process, we shall now derive the transition probabilities. The transition probabilities are denoted $P_{(i,j,k,u)(l,m,q,v)}$, and represents the probability to go from state $(X_n, Y_n, Z_n, S_n) = (i, j, k, u)$ to state $(X_{n+1}, Y_{n+1}, Z_{n+1}, S_{n+1}) = (l, m, q, v)$.

The transition probabilities depend on the time between two successive transmissions. As both UL probe and UL cross traffics are deterministic, this time sets the number of packets that arrived in the AP buffer and the probing node buffer between two transmissions and thus the number of frames that will be sent in the next aggregated frame. We, therefore, analyze the events that may occur between two successive transmissions.

The next stage of our modeling approach is to decide when a transition from one state to another is allowed and compute its probability. Note that impossible transitions have zero probability and that the associated transition probability is computed by assuming that all concerned nodes are equally likely to access the channel. For ease of illustration, we categorize the state transitions into the following three classes.

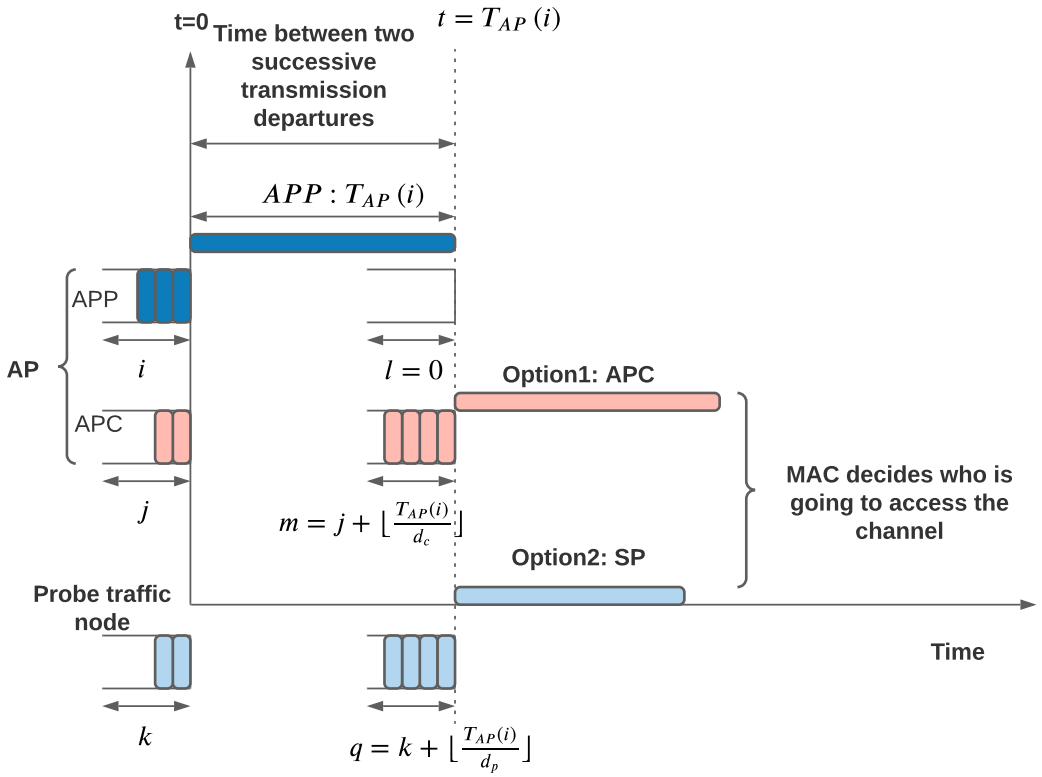


Figure 5.3: Possible events between two successive transmissions for the *Wireless server* model based on aggregated cross traffic when the current transmission is APP.

Class I: Transition from state APP For the sake of clarity, we recourse to Figure 5.3 to describe the possible transition probabilities and their computations. It illustrates an example of possible events between two successive transmissions when the current transmission is APP (an aggregated frame containing probe packets is transmitted by the AP). In this example, at time $t = 0$, we start from state (i, j, k, APP) (i.e., (3,2,2,APP) in Figure 5.3). The AP sends the i corresponding probe frames currently in its buffer ($i = 3$ in this example), aggregating them in a single A-MPDU using its current MCS index. Its transmission duration is denoted $T_{AP}(i)$ (the same formula used to compute $f(l)$ for the two previous Markov chains). Note that the AP has a single buffer that contains at once the probe and the cross traffics, but in order to clarify the explanation of the model, we distinguish between them in Figure 5.3.

During this transmission, the probe traffic node buffer receives $\lfloor \frac{T_{AP}(i)}{d_p} \rfloor$ packets and the AP buffer receives $\lfloor \frac{T_{AP}(i)}{d_c} \rfloor$ cross traffic packets. As a result, at the end of the APP state, the probe traffic node buffer and the AP will contain respectively $k + \lfloor \frac{T_{AP}(i)}{d_p} \rfloor$ of probe packets and $j + \lfloor \frac{T_{AP}(i)}{d_c} \rfloor$ of cross traffic packets ((4,4) in the Figure 5.3).

At the end of this transmission, the buffer of the AP does not have probing frames to send ($X_{n+1} = 0$ almost surely), and another APP transmission is impossible ($S_{n+1} \neq APP$ almost surely). So from this state, only two transitions are allowed: to APC or SP with $X_{n+1} = l = 0$. It can occur only

if $Y_{n+1} = m > 0$ and $Z_{n+1} = q > 0$ respectively.

We derive the non-null transition probabilities as follows. If the AP gains access to the channel, it will send the cross traffic currently in its buffer. For $m > 0$, the next transmission will be APC with probability:

$$P_{(i,j,k,APP)(0,m,q,APC)} = P\Big(S_{n+1} = APC | S_n = APP, X_{n+1} = l = 0, Y_{n+1} = m > 0, Z_{n+1} = q\Big) \cdot \mathbb{1}_{k+\lfloor \frac{T_{AP}(i)}{d_p} \rfloor = q} \cdot \mathbb{1}_{j+\lfloor \frac{T_{AP}(i)}{d_c} \rfloor = m}$$

where $\mathbb{1}_{condition}$ is the indicator function that equals to 1 if $condition$ is true and 0 otherwise. $P\Big(S_{n+1} = APC | S_n = APP, X_{n+1} = l = 0, Y_{n+1} = m > 0, Z_{n+1} = q\Big)$ denotes the probability that the event APC will occur given that the event APP has already occurred. In the interest of brevity, we postpone the computation of such probabilities to the Appendix on Page 95.

Now, if the probe traffic node gains access, the next event will be SP and the transition probability from APP to SP for $q > 0$ is given by:

$$P_{(i,j,k,APP)(l,m,q,SP)} = P\Big(S_{n+1} = SP | S_n = APP, X_{n+1} = l = 0, Y_{n+1} = m, Z_{n+1} = q > 0\Big) \cdot \mathbb{1}_{k+\lfloor \frac{T_{AP}(i)}{d_p} \rfloor = q} \cdot \mathbb{1}_{j+\lfloor \frac{T_{AP}(i)}{d_c} \rfloor = m}$$

Class II: Transition from state APC Once again, when presenting the transition probabilities of this class, we resort to Figure 5.4. It depicts a timeline of feasible events between two successive transmission departures when we start from the state APC.

The AP sends the j cross traffic frames currently in its buffer as an A-MPDU using the transmission rate associated to the Cross traffic node. Its transmission duration is denoted $T_{AC}(j)$. During the period $T_{AC}(j)$, the probe traffic node buffer receives $\lfloor \frac{T_{AC}(j)}{d_p} \rfloor$ packets and the AP buffer receives $\lfloor \frac{T_{AC}(j)}{d_c} \rfloor$ cross traffic packets. At the end of the APC state, the probe traffic node buffer and the AP will thus contain respectively, $k + \lfloor \frac{T_{AC}(j)}{d_p} \rfloor$ of probe traffic packets and $\lfloor \frac{T_{AC}(j)}{d_c} \rfloor$ of cross traffic packets.

Conversely to the previous class of transitions where only two possible transitions are allowed from the state APP, there are here three possible transitions under some conditions. S_{n+1} can be APP, APC, or SP.

First, we suppose that there will be another APC, the transition from APC to APC is deemed possible if and only if $X_n = i = 0$ and $Y_n = l = 0$: it is impossible to have two successive APC transmissions if $i > 0$ or $l > 0$ due to the AP queuing system detailed in Section 2.3 (Figure 2.5).

The transition probability from APC to APC with $Y_{n+1} = m > 0$ is:

$$P_{(0,j,k,APC)(0,m,q,APC)} = P\Big(S_{n+1} = APC | S_n = APC, X_n = X_{n+1} = 0, Y_{n+1} = m > 0, Z_{n+1} = q\Big) \cdot \mathbb{1}_{\lfloor \frac{T_{AC}(j)}{d_c} \rfloor = m} \cdot \mathbb{1}_{k+\lfloor \frac{T_{AC}(j)}{d_p} \rfloor = q}$$

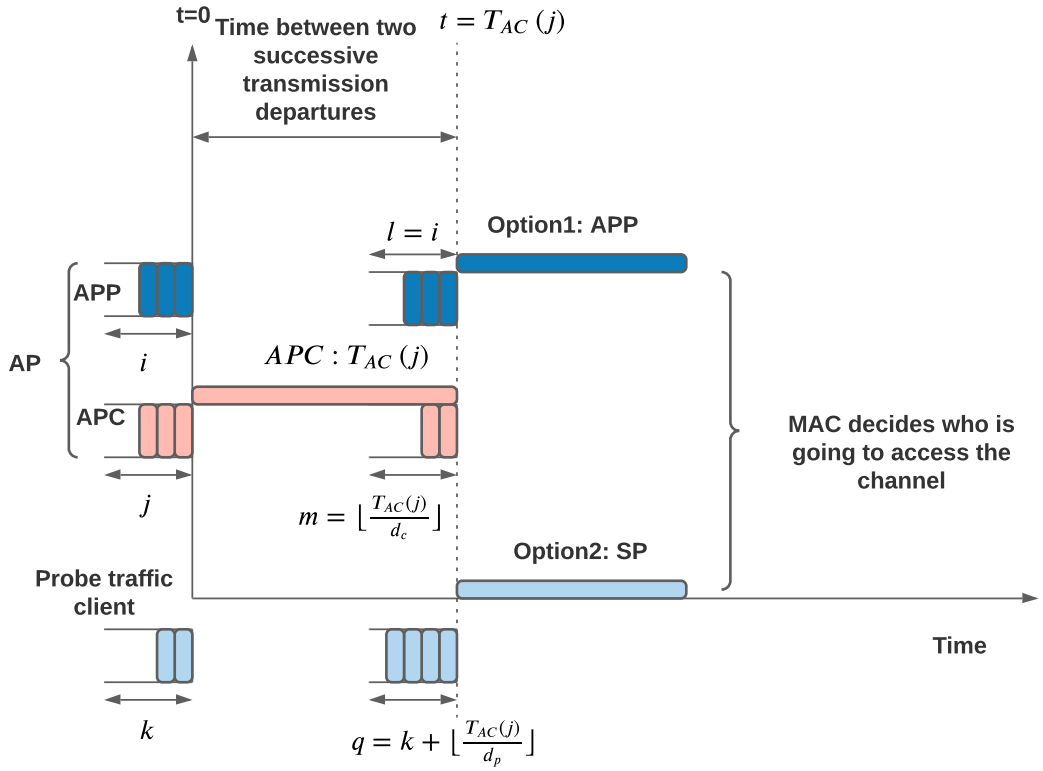


Figure 5.4: Possible events between two successive transmissions for the *Wireless server* model based on aggregated cross traffic when the current transmission is APC.

Now if we suppose that the AP gains the medium access to transmit the DL probe flow, the transition probability from APC to APP is possible only if $X_n = i = X_{n+1} = l$ since the AP cannot receive probe frames in its buffer during the transmission of the cross traffic and $X_{n+1} = l > 0$. The corresponding probability is given by:

$$P_{(i,j,k,APC)(l,m,q,APP)} = P\Big(S_{n+1} = APP | S_n = APC, X_{n+1} = X_n = l > 0, \\ Y_{n+1} = m, Z_{n+1} = q\Big) \cdot \mathbb{1}_{\lfloor \frac{T_{AC}(j)}{d_c} \rfloor = m} \cdot \mathbb{1}_{k + \lfloor \frac{T_{AC}(j)}{dp} \rfloor = q}$$

If we assume that the probe traffic client gains the medium access, the transition from APC to SP is also allowed only if $X_{n+1} = l = X_n = i$ and $Z_{n+1} = q > 0$. We have:

$$P_{(i,j,k,APC)(l,m,q,SP)} = P\Big(S_{n+1} = SP | S_n = APC, X_{n+1} = X_n = l, \\ Y_{n+1} = m, Z_{n+1} = q > 0,\Big) \cdot \mathbb{1}_{\lfloor \frac{T_{AC}(j)}{d_c} \rfloor = m} \cdot \mathbb{1}_{k + \lfloor \frac{T_{AC}(j)}{dp} \rfloor = q}$$

Class III: Transition from state SP In order to derive the transition probabilities of the last class, we apply the same principle. For this

class, we, therefore, do not describe the details of the derived transition probabilities.

From the SP state, we have three possible transitions APP, APC, or SP depending on the competition for the channel resource. First, if the next transition is APP with $X_{n+1} = l > 0$, the transition probability is given by:

$$P_{(i,j,k,SP)(l,m,q,APP)} = P\Big(S_{n+1} = APP | S_n = SP, X_{n+1} = l > 0, Y_{n+1} = m, Z_{n+1} = q\Big) \cdot \mathbb{1}_{l=i+k} \cdot \mathbb{1}_{j+\lfloor \frac{T_{SP}(k)}{d_c} \rfloor = m} \cdot \mathbb{1}_{\lfloor \frac{T_{SP}(k)}{d_p} \rfloor = q}$$

Second, if the next transition is APC with $Y_{n+1} = m > 0$, the transition probability is defined as:

$$P_{(i,j,k,SP)(l,m,q,APC)} = P\Big(S_{n+1} = APC | S_n = SP, X_{n+1} = l, Y_{n+1} = m > 0, Z_{n+1} = q\Big) \cdot \mathbb{1}_{l=i+k} \cdot \mathbb{1}_{j+\lfloor \frac{T_{SP}(k)}{d_c} \rfloor = m} \cdot \mathbb{1}_{\lfloor \frac{T_{SP}(k)}{d_p} \rfloor = q}$$

Finally, if the next transition is SP with $Z_{n+1} = q > 0$, the transition probability is thus formulated as:

$$P_{(i,j,k,SP)(l,m,q,SP)} = P\Big(S_{n+1} = SP | S_n = SP, X_{n+1} = l, Y_{n+1} = m, Z_{n+1} = q > 0\Big) \cdot \mathbb{1}_{l=i+k} \cdot \mathbb{1}_{j+\lfloor \frac{T_{SP}(k)}{d_c} \rfloor = m} \cdot \mathbb{1}_{\lfloor \frac{T_{SP}(k)}{d_p} \rfloor = q}$$

Wireless server model based on non-aggregated cross traffic

Let us now derive the second Markov chain of the *Wireless server* model where the frame aggregation scheme is disabled for the cross traffic.

Figure 5.5 breaks down the network topology simulated by this model that consists of two co-located IEEE 802.11n and IEEE 802.11g WLANs operating on the 2.4GHZ band. The 802.11n WLAN is composed of an AP and two nodes. A probe traffic node to send the UL probe traffic, and a server node to receive the forwarded DL traffic. The 802.11g WLAN is composed of an AP and a node that receives the non-aggregated cross traffic.

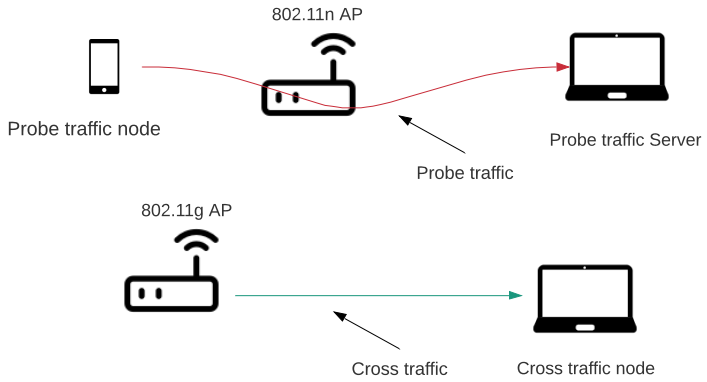


Figure 5.5: Scenario modeled by the *Wireless server* based on non-aggregated cross traffic.

The rationale behind this model is the same as the model described in the previous section except that each cross traffic transmission consists only of a single frame. Like the aggregated version of the *Wireless server* model, we divide the state transitions into the following three classes for the non-aggregated version as follows.

Class I: Transition from APP We first consider the transition probabilities when the current transmission is APP. Since during this transmission, the 802.11n AP sends an aggregated probe frame to the probe traffic server, the transitions probabilities from state APP are the same. We note that the computation of the conditional probabilities related to these three classes are given in Appendix B on Page 97.

If the 802.11g AP gains access to the channel, the transition probability from state APP to state APC with $Y_{n+1} = m > 0$ is:

$$P_{(i,j,k,APP)(0,m,q,APC)} = P\Big(S_{n+1} = APC | S_n = APP, X_{n+1} = l = 0, \\ Y_{n+1} = m > 0, Z_{n+1} = q\Big) \cdot \mathbb{1}_{k+\lfloor \frac{T_{AP}(i)}{d_p} \rfloor = q} \cdot \mathbb{1}_{j+\lfloor \frac{T_{AP}(i)}{d_c} \rfloor = m}$$

Now if the probe traffic node gains the competition for the channel resource, the transition from APP to SP with $Z_{n+1} = q > 0$ is given by:

$$P_{(i,j,k,APP)(l,m,q,SP)} = P\Big(S_{n+1} = SP | S_n = APP, X_{n+1} = l = 0, \\ Y_{n+1} = m, Z_{n+1} = q > 0\Big) \cdot \mathbb{1}_{k+\lfloor \frac{T_{AP}(i)}{d_p} \rfloor = q} \cdot \mathbb{1}_{j+\lfloor \frac{T_{AP}(i)}{d_c} \rfloor = m}$$

Class II: Transition from APC

We second consider the transition probabilities when the current transmission is APC. When presenting the probabilities of this class, we resort to Figure 5.6. It exposes a set of possible events between two successive transmission departures when we start from the state APC.

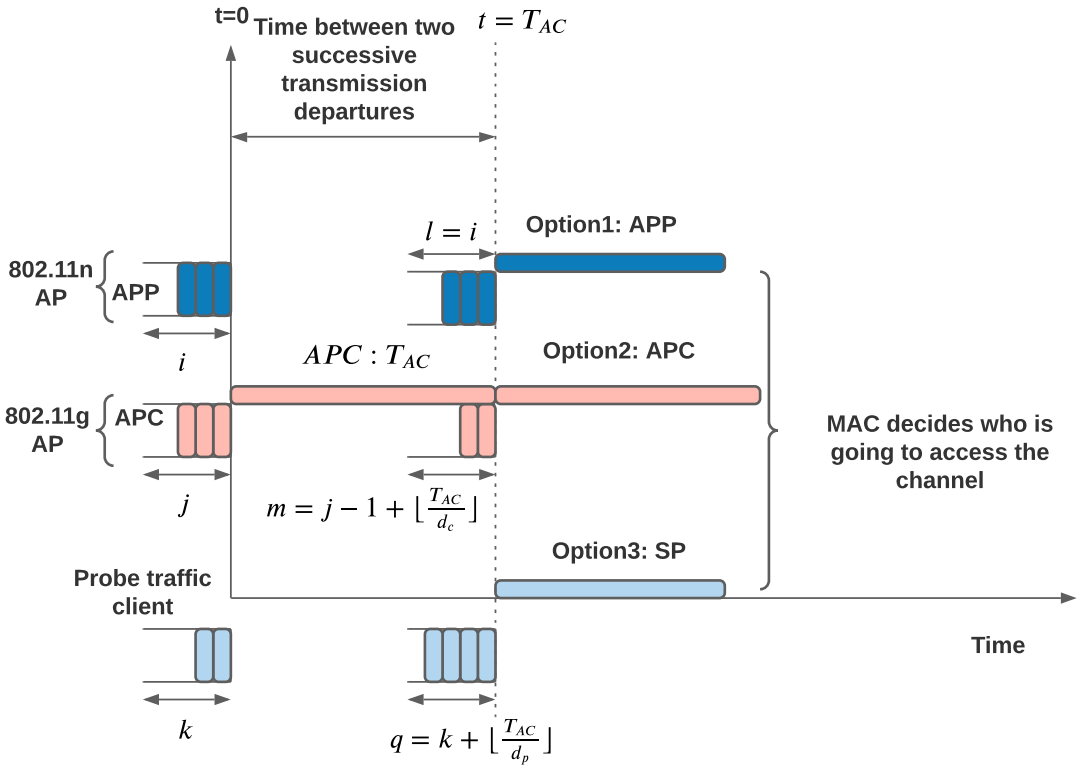


Figure 5.6: Possible events between two successive transmissions for the *Wireless server* model based on non-aggregated cross traffic when the current transmission is APC.

The 802.11g AP sends a single frame from the j cross traffic frames currently in its buffer. Its transmission duration is denoted T_{AC}. During this period, the probe traffic node receives $\lfloor \frac{T_{AC}}{d_p} \rfloor$ of probe packets, and the 802.11g AP receives $\lfloor \frac{T_{AC}}{d_c} \rfloor$ of cross traffic.

Here, there are three possible transitions. The process S_{n+1} can be APP, APC, or SP. It should be noted that all the following transitions are deemed possible if $X_n = i = X_{n+1} = l$ since the 802.11n AP cannot receive probe frames in its buffer during the transmission of the cross traffic by the 802.11g AP.

If we assume that there will be another APC, the corresponding probability with $Y_{n+1} = m > 0$ is defined as follows:

$$P_{(i,j,k,APC)(l,m,q,APC)} = P\Big(S_{n+1} = APC | S_n = APC, X_{n+1} = X_n = i = l, Y_{n+1} = m > 0, Z_{n+1} = q\Big) \cdot \mathbb{1}_{j-1+\lfloor \frac{T_{AC}}{d_c} \rfloor = m} \cdot \mathbb{1}_{k+\lfloor \frac{T_{AC}}{d_p} \rfloor = q}$$

Now if we expect that there will be APP, the transition probability from

APC to APP with $X_{n+1} = l > 0$ is given by:

$$P_{(i,j,k,APC)(l,m,q,APP)} = P\Big(S_{n+1} = APP | S_n = APC, X_{n+1} = X_n = i = l = l > 0, Y_{n+1} = m, Z_{n+1} = q\Big) \cdot \mathbb{1}_{j-1+\lfloor \frac{T_{AC}}{d_c} \rfloor = m} \cdot \mathbb{1}_{k+\lfloor \frac{T_{AC}}{d_p} \rfloor = q}$$

Finally, we establish the transition probabilities from APC to SP. This transition is allowed if $Z_{n+1} = q > 0$ and given by:

$$P_{(i,j,k,APC)(l,m,q,SP)} = P\Big(S_{n+1} = SP | S_n = APC, X_{n+1} = X_n = i = l, Y_{n+1} = m, Z_{n+1} = q > 0\Big) \cdot \mathbb{1}_{j-1+\lfloor \frac{T_{AC}}{d_c} \rfloor = m} \cdot \mathbb{1}_{k+\lfloor \frac{T_{AC}}{d_p} \rfloor = q}$$

Class III: Transition from SP

We now derive the transition probabilities of the last class of this model. From the state SP, there are also three possible transitions APP, APC, or SP depending on the competition for the channel resource.

First, if the next transition is APP with $X_{n+1} = l > 0$, the transition probability is given by:

$$P_{(i,j,k,SP)(l,m,q,APP)} = P\Big(S_{n+1} = APP | S_n = SP, X_{n+1} = l > 0, Y_{n+1} = m, Z_{n+1} = q\Big) \cdot \mathbb{1}_{l=i+k} \cdot \mathbb{1}_{j+\lfloor \frac{T_{SP}(k)}{d_c} \rfloor = m} \cdot \mathbb{1}_{\lfloor \frac{T_{SP}(k)}{d_p} \rfloor = q}$$

Second, if the next transition is APC with $Y_{n+1} = m > 0$, the transition probability is formulated as:

$$P_{(i,j,k,SP)(l,m,q,APC)} = P\Big(S_{n+1} = APC | S_n = SP, X_{n+1} = l, Y_{n+1} = m > 0 Z_{n+1} = q\Big) \cdot \mathbb{1}_{l=i+k} \cdot \mathbb{1}_{j+\lfloor \frac{T_{SP}(k)}{d_c} \rfloor = m} \cdot \mathbb{1}_{\lfloor \frac{T_{SP}(k)}{d_p} \rfloor = q}$$

Finally, if the next transition is SP with $Z_{n+1} = q > 0$, the transition probability is thus formulated as:

$$P_{(i,j,k,SP)(l,m,q,SP)} = P\Big(S_{n+1} = SP | S_n = SP, X_{n+1} = l, Y_{n+1} = m, Z_{n+1} = q > 0\Big) \cdot \mathbb{1}_{l=i+k} \cdot \mathbb{1}_{j+\lfloor \frac{T_{SP}(k)}{d_c} \rfloor = m} \cdot \mathbb{1}_{\lfloor \frac{T_{SP}(k)}{d_p} \rfloor = q}$$

Stationary probabilities for all the models

So far, we have evaluated all the transition probabilities for the four models. The last step consists in deriving the stationary probability and computing the frame aggregation levels. Let us recall that a Markov chain is irreducible if and only if every state can be reached by any other state through one or several transitions. Since our Markov chain is

irreducible and has a finite number of states, it exists a unique stationary distribution [69] . We solve this Markov chain through a numerical method to compute the stationary probabilities. The vector containing the corresponding values is denoted π. The mean probe traffic aggregation levels for the first (*Ideal server*) and second (*Wireless server*) models denoted respectively $MeanAgg_{Ideal\ server}$ and $MeanAgg_{Wireless\ server}$ are then given by:

$$MeanAgg_{Ideal\ server} = \sum_{n=1}^{AMPDU_P} n \cdot \pi_n$$

$$MeanAgg_{Wireless\ server} = \sum_{n=1}^{AMPDU_{AP}} n \cdot \pi_n$$

[69]: Ross (2010), *Introduction to Probability Models*

5.4 Numerical results

In this section, we start by evaluating the accuracy of the proposed models in predicting the frame aggregation levels of the probe traffic. We do so under several scenarios with different network parameters, such as the topology and its size, different IEEE 802.11 amendments, and different traffic patterns. In this regard, we compare the aggregation levels given by the models with those delivered by the discrete-event network simulator 3 (ns-3 version 3.30). We also compare the models' aggregation levels with the measurements made during a test-bed experiment in order to get realistic scenarios that capture the complexity of the whole network stack and definitely give a convenient behavior preview. Then, we study the difference between the proposed Markovian models. For simplicity reasons, we assume for all the scenarios that all nodes have a random but fixed position during the whole simulation duration. We compute the four models for six cross traffic loads/BTF levels: 0, 0.125, 0.25, 0.375, 0.5, and 0.625 ranging from low to high levels of BTF. The maximum numbers of aggregated sub-frames $AMPDU_{AP}$ and $AMPDU_P$ were set to 36. The parameters used in the simulations and models are summarized in Table 5.2

Table 5.2: The DCF parameters for IEEE 802.11n/g standard amendments in 2.4GHz band.

Parameter	value
CWmin	15
T_{slot} (μs)	9 or 20
T_{DIFS} (μs)	28 or 50
T_{SIFS} μs	10
FCS (Bytes)	4
Packet Size (Bytes)	1024
Channel width	20MHz

Validation of the Ideal server model based on aggregated cross traffic

We start by examining the accuracy of the *Ideal Server* model using the Markov chain where frame aggregation mechanism is enabled for both probe and cross traffics under the IEEE 802.11n amendment [70] . We

[70]: Bouzouita et al. (2020), 'Analytical study of frame aggregation level to infer IEEE 802.11 network load'

compare the model outcomes to ns-3 simulations and a controlled lab experiment.

Simulation - Two STAs

In this simulation, the network topology illustrated in Figure 5.7, is composed of an AP, and two user STAs: a station that sends the probe traffic and a second one for receiving the cross traffic from the AP. The two stations are connected to the AP and satisfy IEEE 802.11n specifications. The channel is assumed error-free, and all stations operate with the same physical data rate corresponding to the HT-MCS 15 (144.4 Mbps).

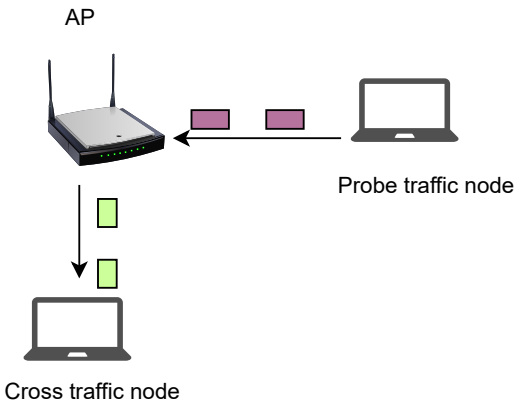


Figure 5.7: ns-3 simulation with two STA for the *Ideal server model* based on aggregated cross traffic.

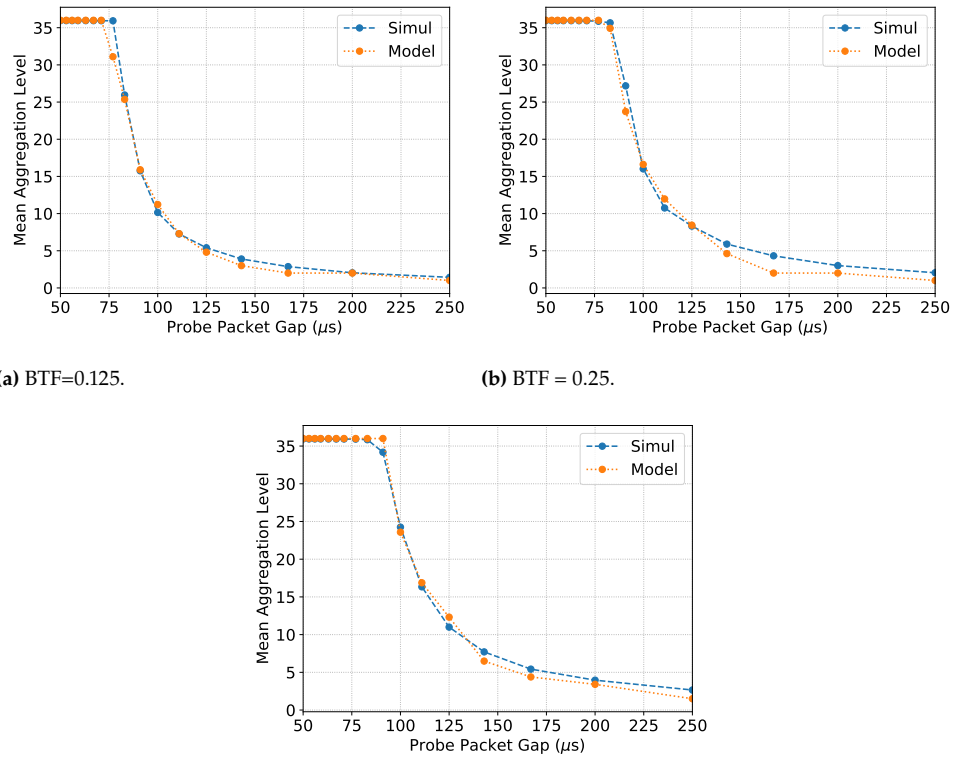
Figure 5.8 shows the mean aggregation level for the probe traffic as a function of the probe packet gap for the analytical model and simulations for the three BTFs: 0.125, 0.25, and 0.375. In order to generate different probing rates, we increase gradually the probe packet gap from 50 μs to 250 μs, and we fix the packet size to 1024 Bytes.

According to these results, it appears that the model follows closely the pattern of the ns-3 simulations for all the three levels of network load with a relative error typically less than 10%. We note that we obtain the same results for the other three cases of BTF.

The obtained curves can be divided into two zones. We observe a first zone where the aggregation level is at its maximum. It corresponds to a very congested state where the probe traffic buffer is always full and exceeds the maximum number of frames that can be aggregated. When the probe packet interval increases, the aggregation decreases and follows a curve close to a hyperbola explained by the fact that the number of generated packets per second is the inverse of the probe packet interval.

Simulation - Five STAs

In our Markov chain-based models, the cross traffic is sent by a single queue. In practice, an AP will have several associated stations most of the time. So this simulated scenario involves a more complex topology that models cross traffic sent to four concurrent STAs to properly quantify the impact of the number of stations to which the AP sends the DL traffic. The



(a) BTF=0.125.

(b) BTF = 0.25.

(c) BTF = 0.375.

Figure 5.8: Mean aggregation levels of *Ideal server* model based on aggregated cross traffic versus ns-3 simulations - Two STAs.

network topology is hence composed of an AP and five nodes, illustrated in Figure 5.9.

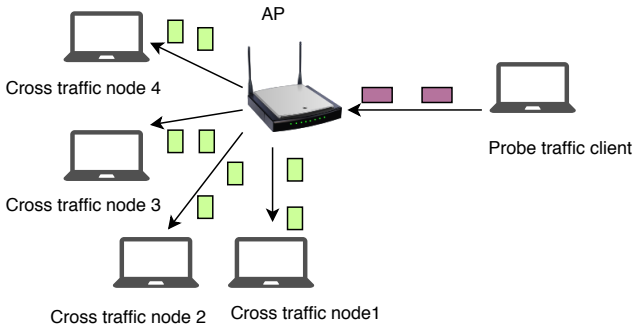
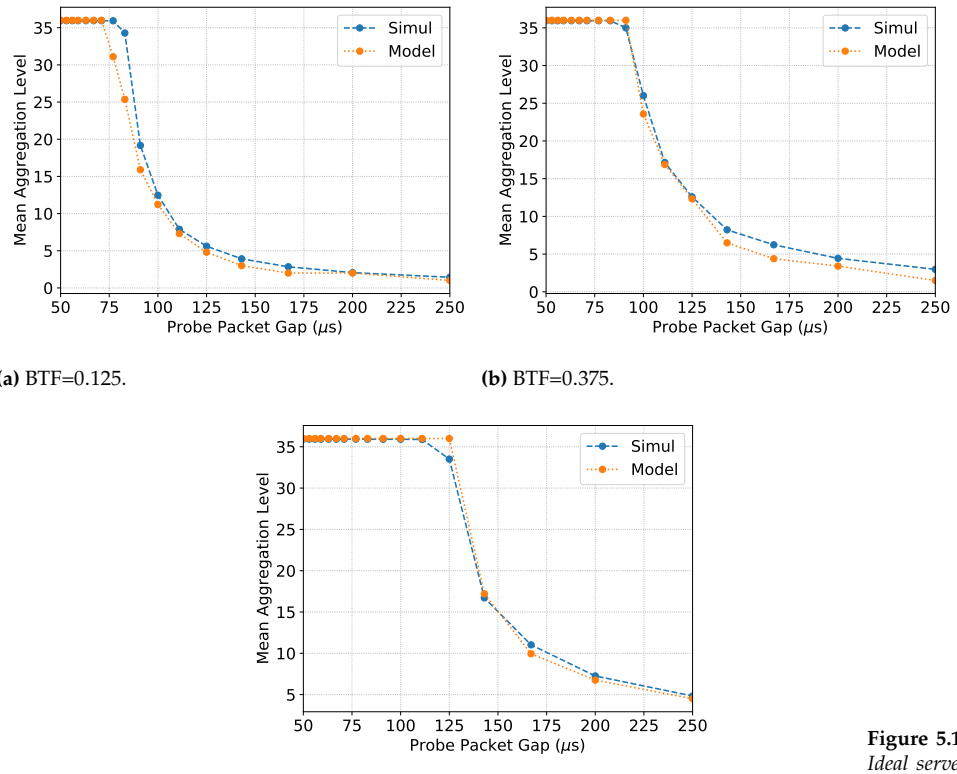


Figure 5.9: ns-3 simulation with five STAs for the *Ideal server* model based on aggregated cross traffic.

In Figure 5.10, we compare the mean aggregation levels obtained with ns-3 with the model based on aggregated cross traffic for this scenario. The corresponding results show that the mean aggregation levels derived from simulations are always close to our model, and all the curves show similar patterns. We choose to present the results for the BTFs 0.125, 0.375, and 0.625, bearing in mind that the three other cases show the same accuracy. Based on these results, we show that the model might be relevant even for several stations composing the DL competing traffic coming from the same AP.



(a) BTF=0.125.

(b) BTF=0.375.

(c) BTF=0.625.

Figure 5.10: Mean aggregation levels of *Ideal server* model based on aggregated cross traffic versus ns-3 simulations - Five STAs.

Experimental validation

Simulations are good for capturing bounds and trends of analytical models, but not as good for evaluating the performance one should expect in practice. Consequently, the analytical results were compared and validated using an experimental test-bed. This experiment (illustrated in Figure 5.11) was conducted in a controlled lab environment (a helical room) where there is no interference or channel fading [71] . The general setting is as follows: we used a Linux laptop to execute the probe traffic sender application. Another computer, configured as an IEEE 802.11n AP, was used as a server for the probe and client for the cross traffic. The physical transmission rate was set to 144.4 Mbps (i.e., HT-MCS 15 in 802.11n). An Android phone was deployed, acting as the cross traffic receiver application. Also, we configured a computer (Sniffer) with a specific WNIC that supports the *survey dump* feature of the *iw* command which is a Linux utility that shows the survey information of all the available channels including the *channel busy time* and *channel active time*, thereby we measure the ground truth value of the BTF as detailed in Chapter 4. This computer is also configured in monitor mode to capture the aggregated frames by using the dedicated capture software *Wireshark*. The aggregation level is computed according to the A-MPDU reference number in the radiotap header (additional information added by the wireless adapter or its driver).

[71]: Massouri et al. (2014), 'CorteXlab: An open FPGA-based facility for testing SDR cognitive radio networks in a reproducible environment'

All the devices hardware configurations versions are detailed in Table 5.3. All the nodes operate on channel 1 in the 2.4GHz band with 20MHz bandwidth.

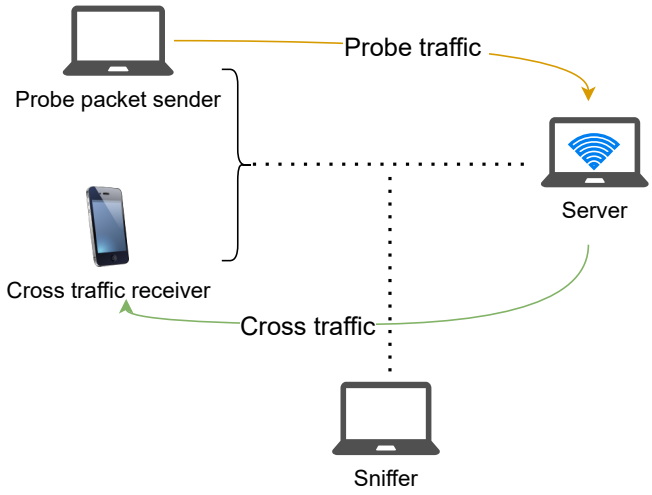


Figure 5.11: Experimental test-bed.

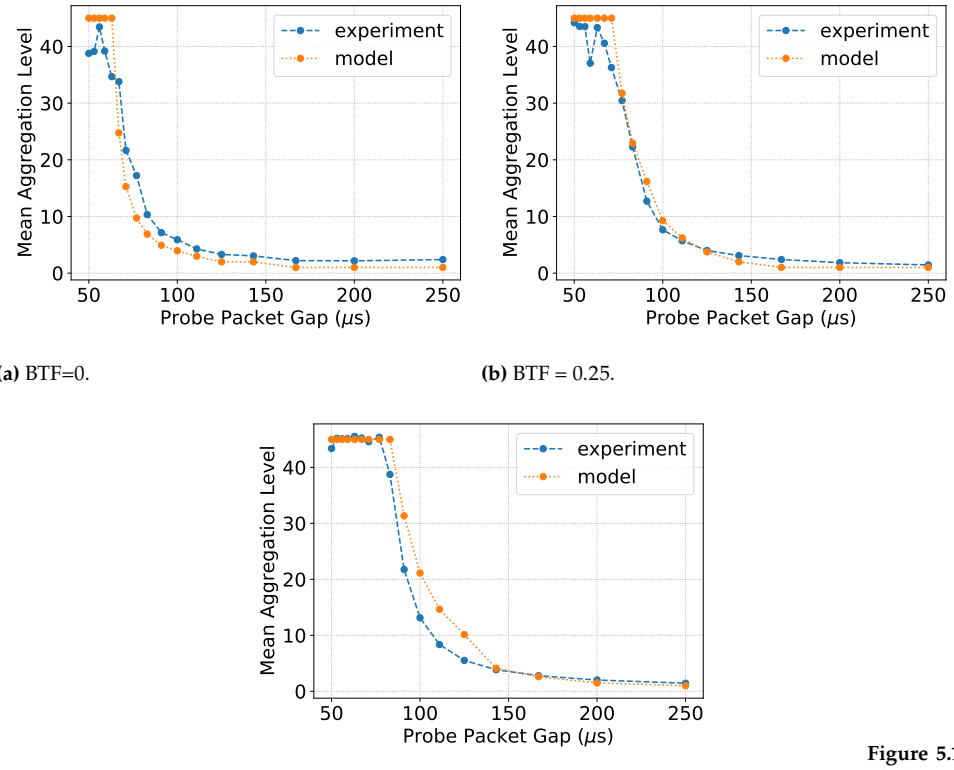
Table 5.3: Mobile phone and laptops used in the experiment.

Role	Model	WNIC
Packet Receiver and Access Point	HP EliteBook 840 G1	Intel Dual-Band Wireless-N 7260
Packet Sender (computer)	HP EliteBook 840 G1	Intel Dual-Band Wireless-N 7260
Packet Sender (Phone)	Pixel 3 XL	
Sniffer	Dell Inspiron 7559	Intel Dual Band Wireless-AC 3165 M.2 Card

Figure 5.12 shows the mean aggregation level for the probe traffic as a function of the probe traffic packet gap for the analytical model and experiments. In these experiments, frame aggregation is enabled for cross traffic. Experiments are therefore compared to the *Ideal server* model based on aggregated cross traffic outcomes. We show only the results of 0, 0.25, and 0.5 BTFs since the other cases present the same accuracy.

We observe that our model is able to capture with reasonable accuracy the experimental mean aggregation level of the probe traffic for all the levels of the cross traffic. Not surprisingly, for a very high level of probe traffic (small probe packet gaps), the mean aggregation level reaches the maximum. Then, it decreases until reaching 1. The slight discrepancy in predicting the precise frame aggregation level can be explained by the fact that our model does not take into account the Wi-Fi's dynamic rate adaptation algorithms. For Wi-Fi networks, the rate adaptation algorithm is the process of choosing suitable transmission parameters to cope with the fluctuating wireless channel conditions in order to maintain the QoS. However, despite the slight discrepancy for some probe packet gaps, the results returned by the model are in good agreement with those provided by the experiments.

In summary, this experimentation allows us to test our model under a more realistic PHY and MAC layers. Overall, these results demonstrate that despite the complexity brought by the network stack layers (beacon frames, congestion, ARQ, random backoff, etc.), our approach captures the mean aggregation level with a reasonable level of precision in the considered scenarios.



(a) BTF=0.

(b) BTF = 0.25.

(c) BTF = 0.5.

Figure 5.12: Mean aggregation levels of *Ideal server* model based on aggregated cross traffic versus experiments.

Simulation - 802.11ax traffic

With the advent of new IEEE 802.11 standards such as 802.11ax [72] , the performance of WLANs has changed. The 802.11ax amendment offers HE MCS indexes that provide higher data rates than any of its predecessors. For the sake of completeness, we ran our model for another scenario based on this recent standard. In this scenario, we study the model's accuracy when we change the underlying Wi-Fi network from 802.11n to 802.11ax in the 2.4 GHz band for both the probe and the cross traffics. We used the current implementation of the IEEE 802.11ax of ns-3. All the nodes (AP and STA) operate at two data rates, 286.8$Mbps$ (HE-MCS 11, spatial streams=2, 20MHz with 1024QAM, 0.8 Guard Interval (GI)) and 2402$Mbps$ (HE-MCS 11, spatial streams=2, 160MHz with 1024QAM, 0.8 GI). The network topology is composed of an AP and two user STAs. A probe traffic node that sends the probe traffic, and another node that receives the aggregated cross traffic from the AP.

[72]: IEEE (2019), 'Draft 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 Amendment Enhancements for High Efficiency WLAN'

Figure 5.13 shows that the mean aggregation levels obtained with simulations for BTF=0.25 perfectly fit the ones from the model for both the low and the high data rates. We note that the model shows the same accuracy for the other cases of BTF.

Validation of Ideal server model based on non-aggregated cross traffic

We now examine the proposed approach's accuracy when the frame aggregation mechanism is disabled for the cross traffic. The considered

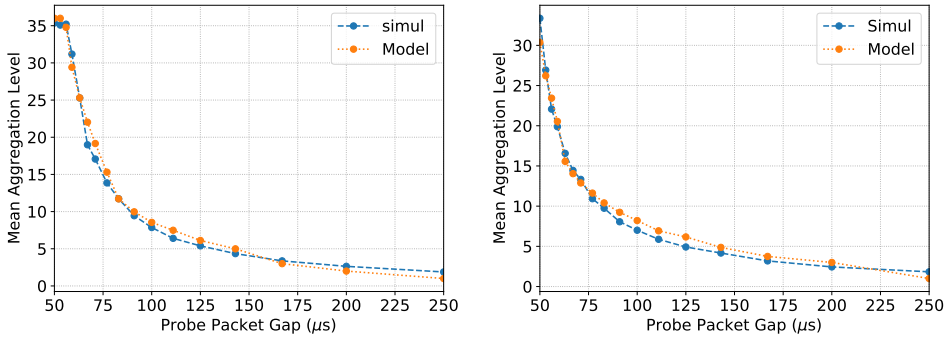


Figure 5.13: Mean aggregation levels of *Ideal server* model based on aggeragted cross traffic versus ns-3 simulations - 802.11ax traffic.

(a) BTF = 0.25, Data rate = 286.8 Mbps.

(b) BTF = 0.25, Data rate = 2404 Mbps.

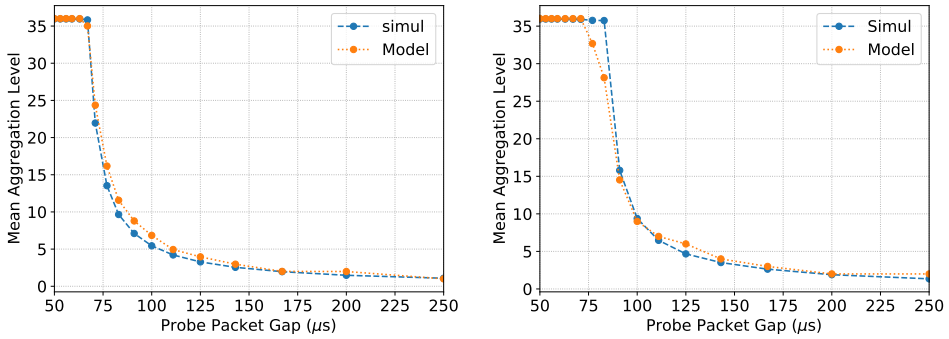


Figure 5.14: Mean aggregation levels of *Ideal server* model based on non-aggregated cross traffic versus ns-3 simulations.

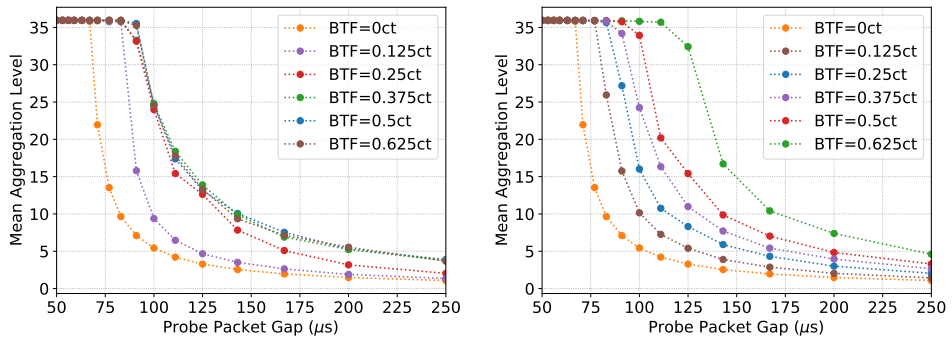
(a) BTF = 0.

(b) BTF = 0.125.

scenario consists of two co-located IEEE 802.11n and IEEE 802.11g WLANs on the 2.4 GHz band. The 802.11n WLAN is composed of an AP and a node that sends the probe traffic, while the 802.11g is composed of an AP and a node that receives the non-aggregated cross traffic. The data rate was set to 144.4 Mbps for the probe traffic and 54 Mbps for the cross traffic.

We compare the aggregation given by the model to the values obtained by the ns-3 simulations in Figure 5.14a and Figure 5.14b. It appears that the model is able to estimate the mean aggregation level with reasonable accuracy for the two BTFs. The difference that can be observed is due to the fact that beacons are not taken into account in the model, that is why it underestimates the aggregation level for some probe packet gaps. We note that our models take into account the time to send block ACK requests while computing the frame transmission duration.

Figure 5.15a and Figure 5.15b show the probe mean aggregation levels for each of the six BTFs when the frame aggregation scheme is disabled or enabled for the cross traffic respectively. Based on Figure 5.15a, we can see that varying the level of cross traffic barely affects the probe aggregation levels when the $BTF > 0.25$. In this case, each cross traffic frame is sent independently, with short transmission times. Consequently, the probe traffic receives less packets to aggregate between two consecutive medium accesses. Also, cross traffic reaches saturation faster (as it sends less frames on average). As soon as it has always a frame to send, its access time does not depend on its buffer state. Consequently, the aggregation level of the probe traffic becomes insensitive to the level of congestion



(a) Simulated Mean Aggregation levels for all BTFs when the cross traffic does not aggregates its frames.

(b) Simulated Mean Aggregation levels for all BTFs when the cross traffic aggregates its frames.

Figure 5.15: Mean aggregation levels versus BTFs, cross traffic aggregates or not its traffic.

of the cross traffic. On the contrary (Figure 5.15b), when cross traffic aggregates its frames, the state of its buffer has a deeper impact on probe traffic aggregation since the cross traffic buffer state determines the transmission duration. These results nicely highlight the fact that varying the level of the aggregated network load significantly affects the probe aggregation levels. Consequently, the results are sufficiently separated to be used to infer the load level (Chapter 6).

Validation of the Wireless server model based on aggregated cross traffic

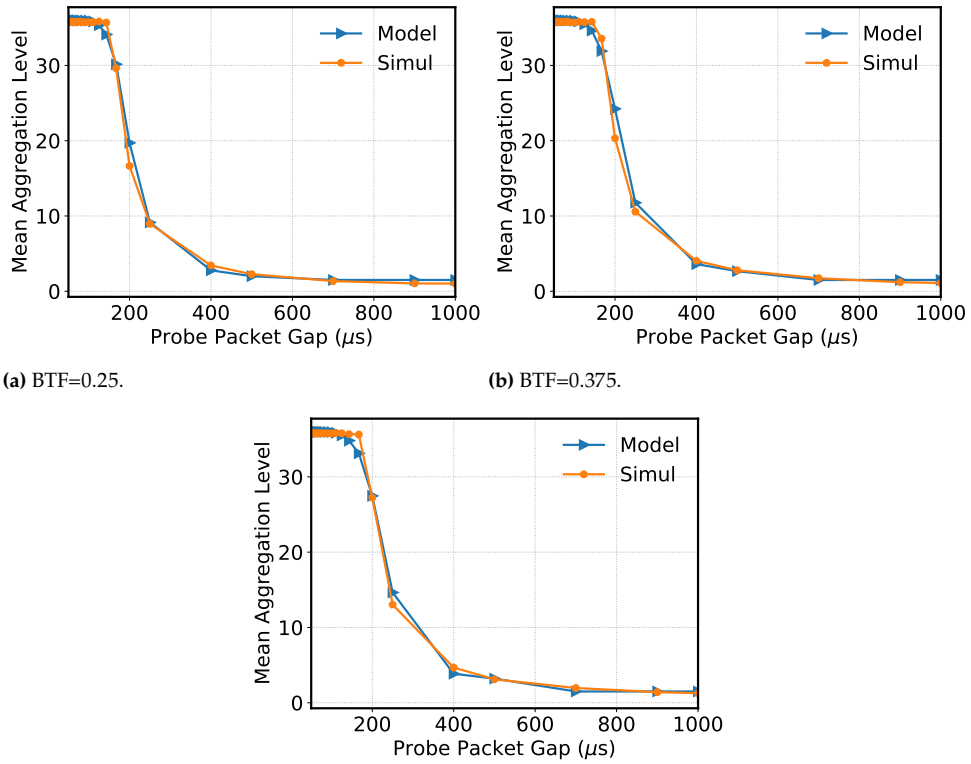
We now examine the accuracy of the *Wireless server* model using the Markov chain where frame aggregation scheme is enabled for both probe and cross traffics under the 802.11 standard amendment.

Simulation - Same MCS

We start by examining the performance details of this model when the same MCS index (HT-MCS 15 with a physical transmission rate of 144.4 Mbps) is used for the probe traffic node and the AP. The maximum numbers of aggregated sub-frames $AMPDU_{AP}$ and $AMPDU_P$ are set to 36. Figure 5.16 shows the mean aggregation level for the probe traffic as a function of the probe packet gap for the model and simulations. We let the probe packet gap gradually varies from 50μs to 1000μs. According to these results, it appears that the model performs well since it follows closely the pattern of the ns-3 simulations for the three levels of the network loads (0.25, 0.375, and 0.5). It is also the case for the three other BTFs (0, 0.125, and 0.625) that are not shown here. We can observe that the aggregation levels depend on the loads of the network. Indeed, when the BTF increases, the probe traffic has to wait longer, and more packets are received between two successive probe transmissions.

Simulation - Different MCS

We now evaluate the performance details of the *Wireless server* model when different MCS indexes are used for the probe traffic node and the



(a) BTF=0.25.

(b) BTF=0.375.

(c) BTF=0.5.

Figure 5.16: Mean aggregation levels of *Wireless server* model based on aggregated cross traffic versus ns-3 simulations - Same MCS.

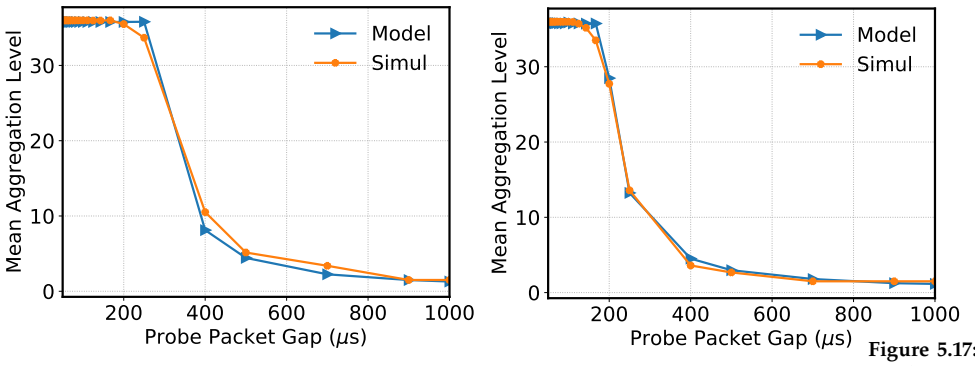
AP.

Figure 5.17 shows the corresponding results. Figure 5.17a plots the results when we use the MCS 15 (144.4 Mbps) for the AP to send the DL probe traffic and the cross traffic and the MCS 11 (57.8 Mbps) for the probing node. Figure 5.17b provides the corresponding results when we use MCS 15 for the AP and MCS 13 (115.6 Mbps) for the probe traffic node. The results show that considering different MCS indexes for the AP and the probe traffic node does not impact the accuracy of our approach.

Simulation - Exponential On/Off cross traffic

To provide a broader overview of the accuracy reached by the proposed modeling approach, this scenario simulates another traffic pattern for the cross traffic which is exponential On/Off traffic, reflecting some of the kinds of cross traffic that would occur in practice. In ns-3, we use exponentially distributed On/Off periods thanks to the ns-3 OnOff Application class. This latter mainly relies on an *OffTime* and *OnTime* duration attributes that represent respectively the duration during which the data transfer is switched off and the duration of the continuous data transfer.

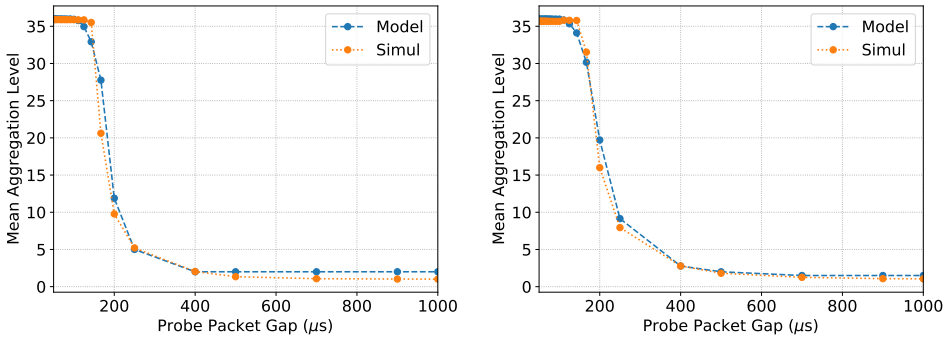
Figure 5.18 shows the mean aggregation levels for the BTFs 0.125, 0.25 and 0.375. These results show that the estimations made by our model fit those delivered by ns-3 for the exponential On/Off aggregated cross traffic. We conclude that the proposed models still perform well under exponential On/Off traffic pattern.



(a) BTF= 0.375, MCS 11 for the probe traffic node.

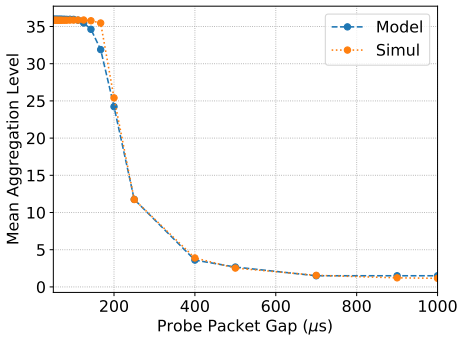
(b) BTF= 0.375, MCS 13 for the probe traffic node.

Figure 5.17: Mean aggregation levels of *Wireless server* model based on aggregated cross traffic versus ns-3 simulations - Different MCS.



(a) BTF=0.125

(b) BTF=0.25.



(c) BTF=0.375.

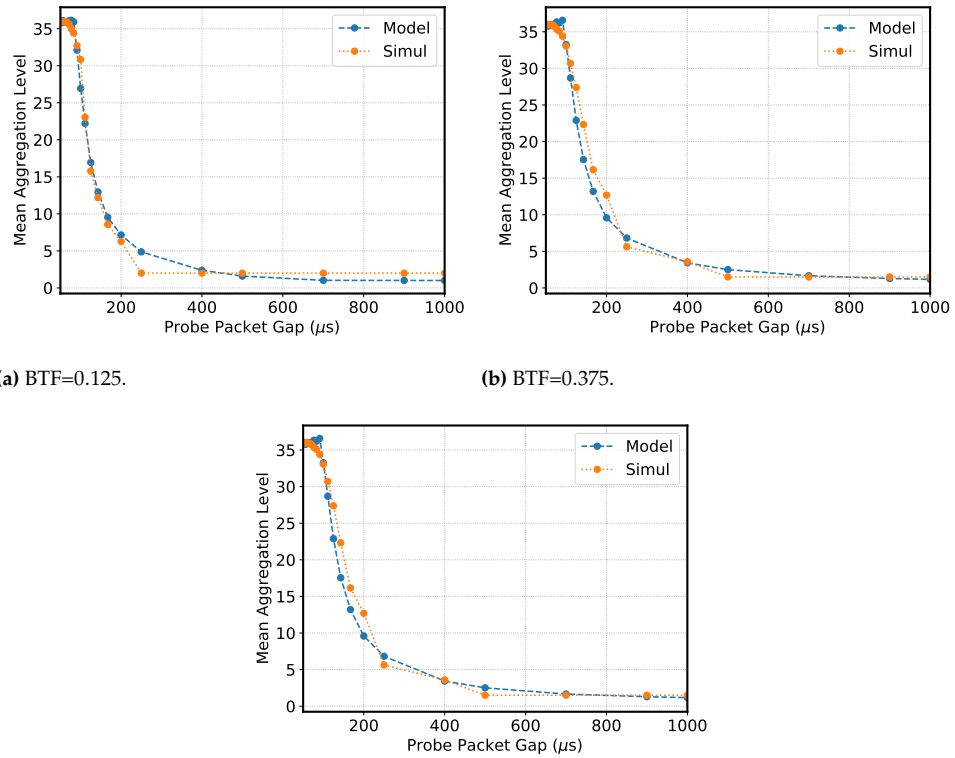
Figure 5.18: Mean aggregation levels of *Wireless server* model based on aggregated cross traffic versus ns-3 simulations - Exponential On/Off cross traffic.

Simulation - 802.11ax traffic

Here, we run the model *wireless server* when both the probe and cross traffics use the 802.11ax standard. All the nodes use the data rate=286.8Mbps. It is noticeable from Figure 5.19 that the mean aggregation levels obtained with simulations fit the ones from the model.

Validation of the Wireless server model based on non-aggregated cross traffic

We now examine the accuracy of the *wireless server* model based on non-aggregated cross traffic under several scenarios.



(a) BTF=0.125.

(b) BTF=0.375.

(c) BTF=0.5.

Figure 5.19: Mean aggregation levels of *Wireless server* model based on aggregated cross traffic versus ns-3 simulations - 802.11ax traffic.

Simulation - Three STAs

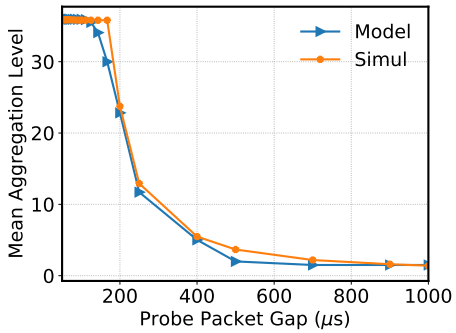
We consider the same scenario described in Figure 5.5. We compare the aggregation levels given by this model to the ns3 simulations' outcomes. The results are plotted in Figure 5.20 for the three levels of BTF: 0.375, 0.5 and 0.625.

Based on this figure, we observe that the model is able to reproduce the DL probe traffic aggregation behavior for the three BTFs with a satisfying degree of precision.

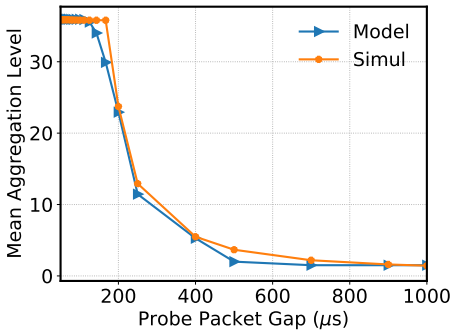
Simulation - Six STAs

We now consider the case where we increase the number of competing cross traffic nodes from one to four. The results shown in Figure 5.21 reveal that the number of stations composing the DL competing traffic coming from the same AP does not influence the prediction for the model wireless server.

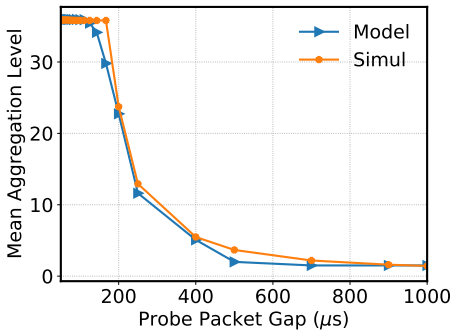
Overall, these analytical solutions are found to be accurate delivering estimates in good agreement with the simulations results for the considered scenarios. Note that in order to investigate the robustness of the *Wireless server* models, we explored the same scenarios used to validate the accuracy of the *Ideal server* models. It was our experience that the *Wireless server* models show the same accuracy.



(a) BTF=0.375.

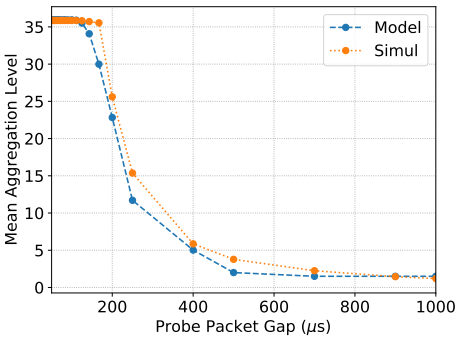


(b) BTF=0.5.

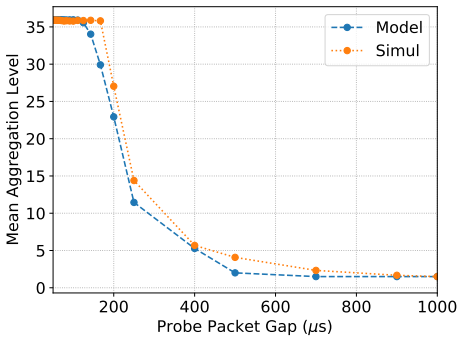


(c) BTF=0.625.

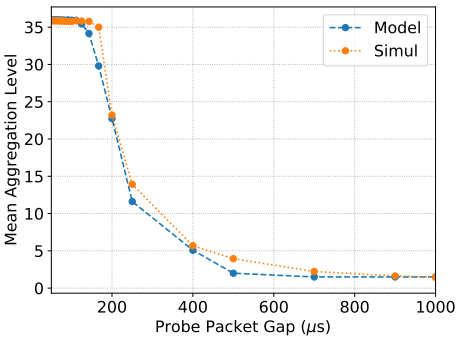
Figure 5.20: Mean aggregation levels of *Wireless server* model based on non-aggregated cross traffic versus ns-3 simulations - Three STAs.



(a) BTF=0.375.

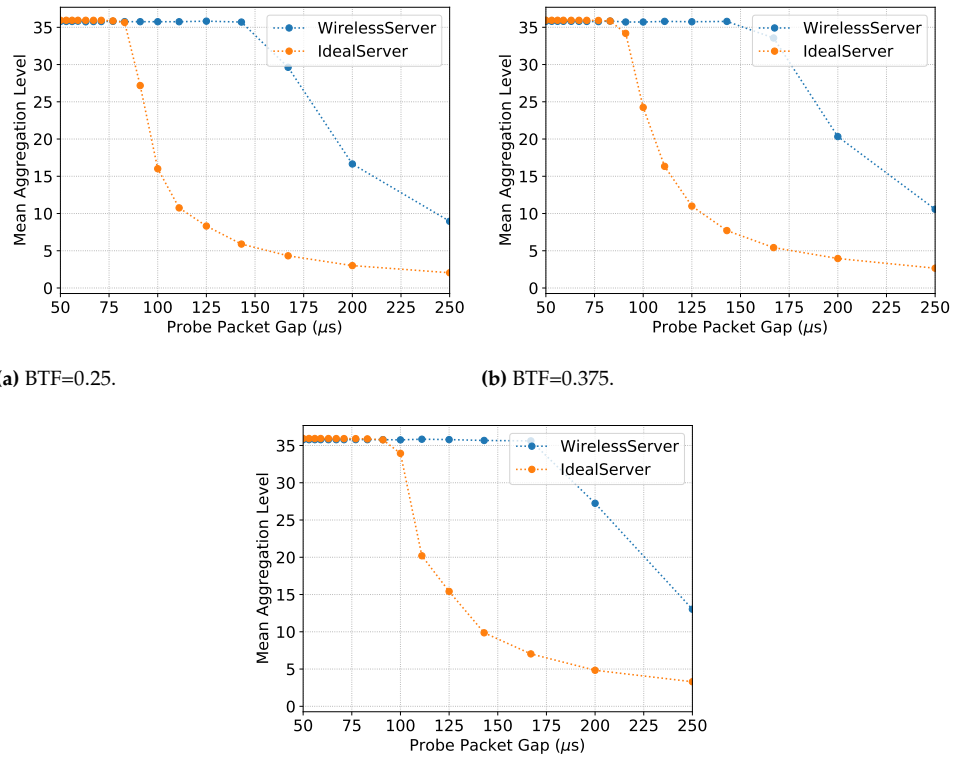


(b) BTF=0.5.



(c) BTF=0.625.

Figure 5.21: Mean aggregation levels of *Wireless server* model based on non-aggregated cross traffic versus ns-3 simulations - Six STAs.



(a) BTF=0.25.

(b) BTF=0.375.

(c) BTF=0.5.

Figure 5.22: Mean aggregation levels: *Ideal server* model versus *Wireless server* model.

Comparison between the models

We now compare the two proposed models according to the aggregation levels. Figure 5.22 provides the mean probe aggregation levels when frame aggregation is enabled for the cross traffic for the two models. It can be seen that the UL aggregation levels obtained by the *Ideal server* model are consistently lower than the DL aggregation levels returned by the *Wireless server* model. Indeed, the behavior of the aggregation levels is more complex since the DL probe traffic depends on the competition for medium access with the UL probe traffic and the cross traffic. In particular, when the cross traffic increases, the DL probe traffic has to wait longer. More packets may hence accumulate in the queue of the AP and be aggregated when the next DL probe transmission occurs.

5.5 Conclusion

In this chapter, the overall goal was to propose new analytical Markov-based models that discern the aggregation levels of an aggregated deterministic probe traffic competing with the current network traffic that can aggregate or not its frames. To that aim, we first proposed the model named *Ideal server*. We have assumed that the connection between the AP and the server is ideal. We modeled it as if the server were implemented on the AP or close to the AP. This latter evaluates the aggregation levels of the UL probe traffic. The second configuration considers that the server is embedded on a second wireless device, owned by the user and connected to the same AP. This model called

Wireless server evaluates the aggregation levels of the DL traffic. Despite the findings of the ideal server model, one has to acknowledge that the *wireless server* model accommodates several improvements since it emulates a more realistic scenario.

We made a thorough evaluation of the performance of the proposed analytical modeling approaches through simulations and a test-bed experiment. In all considered scenarios, the comparison of the frame aggregation levels of the probe traffic computed thanks to the analytical models and the one measured through ns-3 simulations, and the experiment yielded similar performance. These results have shown that the models allow an accurate estimation and nicely highlight the fact that varying the level of the network load significantly affects the probe aggregation levels for the proposed models. Consequently, these theoretical findings are sufficiently separated to be used to infer the load level and the type of the network. This correlation will therefore be addressed in the next chapter.

5.6 Perspectives

The next step would consist in studying the feasibility of implementing such Markovian models when the server is located outside the local networks of the Wi-Fi APs. These future works would consider more complicated network topology and discuss the impact of the placement of the server, in particular, when the bottleneck is located somewhere on the Internet between the AP and the server in order to establish a convenient performance evaluation of different configurations. The comparison would be based on analyzing the aggregation levels of the probe traffic offered by each configuration under different scenarios that would differ in the number of user stations, their data rates, the number of hops constituting the network path, etc.

Our models are validated under the assumption that all the network traffic is DL, i.e., downloaded from the AP to the STA such as P2P file download application. It is worth keeping in mind that Internet traffic is asymmetric since user STAs download much more data than they upload [68] . Additional thoughts would, therefore, be dedicated to considering the UL traffic in our modeling approach. Even if the DL is predominant, the UL traffic can impact the results. Future work would consider more scenarios that discuss the impact of such traffic on the accuracy of the models.

[68]: Gupta et al. (2012), 'WiFox: Scaling WiFi Performance for Large Audience Environments'

An additional perspective will investigate the extension of the models to incorporate the use of a more sophisticated function $p(k)$ that depends on the contention window, the number of competing STAs and k. Overall, all these kinds of improvements will allow us to modify our models in order to accommodate the new findings.

FAM: A Frame Aggregation Based Method to Infer the Load Level in IEEE 802.11 Networks

6

The previous chapter proposed analytical Markov-based models to estimate the theoretical aggregation levels of the probe traffic and compiled a series of observations indicating that we can rely on this estimation to infer the network load for modern Wi-Fi networks. Specifically, we find a strong correlation between the aggregation levels of the probe traffic and the network load. This realization actually gives a stepping stone to designing a system able to infer the latter from the former. In this chapter, we hence introduce a novel network load inference method called Frame Aggregation based Method (FAM). It exploits the rich information embedded in frame aggregation to infer both the BTF of an AP and the type of traffic. Along with an active probing approach and Markovian models, the designed BTF estimation approach conquered the problem of estimating the BTF for vanilla devices, especially Android-based devices. The designed proof of concept system is carefully evaluated under various ns-3 scenarios, a test-bed experiment, and a real-world trace-driven simulation.

This chapter is organized as follows. In Section 6.1, we give a detailed overview of the method FAM by exposing all the involved algorithms. A thorough performance evaluation is then presented in Section 6.2. Finally, a short discussion on the strengths and limitations of the approach and some conclusions are given in Section 6.3.

6.1 FAM overview

In this section, we describe our scheme, FAM, that allows the node conducting the measurement to estimate the BTF of the wireless channel and infer the type of traffic. Note that no changes are required, neither to the device nor to the AP. The designed system aims at achieving this estimation for an unmodified mobile device. The foundation of FAM is mainly built on three steps as follows.

- Measurements of the mean aggregation levels for different probe traffic flows
- Detection of the use of frame aggregation in the cross traffic
- Estimation of the wireless channel load

We unfold the details of the proposed approach by exposing the algorithmic representation of each stride. Each step relies on one or many algorithms. We note that all the algorithms detailed below are implemented on the server except the client procedure (Algorithm 1), which is implemented on the probe traffic node (as shown in Figure 6.1).

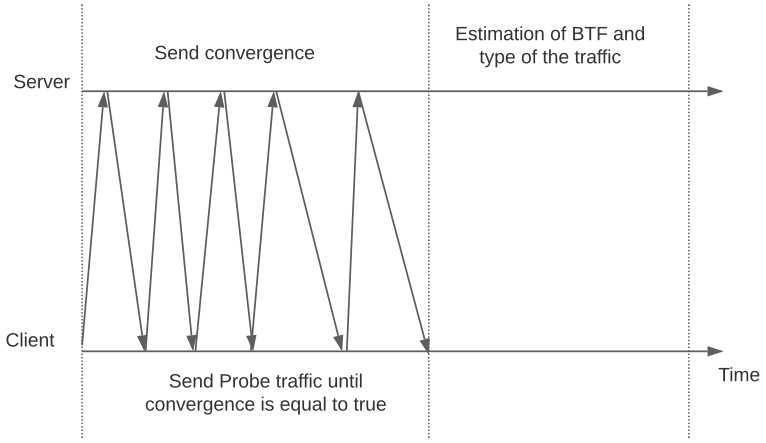


Figure 6.1: FAM work flow

Mean aggregation level computation

Initially, we start with the measurement of the aggregation levels of the probe traffic. This process is described in algorithms 1 and 2. Algorithm 1 runs on the probe traffic node. It starts by sending successive batches of probe packets. For each batch, the gap interval between packets, d_p, is increased from $d_{p_{min}}$ to $d_{p_{max}}$. In order to evaluate the mean aggregation levels accurately while keeping the batch size as small as possible, the number of packets in a batch, denoted np, is not fixed. This number is determined using the Central Limit Theorem (CLT) detailed in the server procedure (Algorithm 2). Upon receipt of this batch, the server responds with a packet that indicates to the probe traffic node whether to stop sending packets. It is the *convergence* variable in the algorithm.

Algorithm 1 Client Procedure

Input: $AMPDU$: Maximum A-MPDU Length, np : size of a batch of probes

1: $d_{p_{min}} = \frac{f(AMPDU)}{AMPDU}$
2: $d_p = d_{p_{min}}$
3: **repeat**
4: **repeat**
5: client.send (d_p, np) ▷ Send np packets with interval d_p
6: client.receive(convergence)
7: **until** convergence is True
8: client.receive($meanAgg$)
9: $d_p = d_p$+increment
10: **until** $meanAgg > 2$
11: client.send(0, 1) ▷ Send 1 packet to inform that the campaign is finished.

On the other side, The Algorithm 2 runs on the server. It receives and processes the incoming probe packets from the probe traffic node. During a given batch, the mean aggregation is computed on the fly at the reception of probe packets. We use the CLT to evaluate the accuracy of the mean aggregation level. When the error is lower than the expected

error E, the server sends the *convergence* notification to the probe traffic node (that stops its batch). The use of the CLT allows the application to appropriately measure the mean aggregation levels with a minimal amount of time and bandwidth cost.

Algorithm 2 Server Procedure

Input: E :acceptable standard error of the mean, Z : Value of the distribution function

MeanAgg=∅, D=∅
while campaign in progress for this client **do**
  $n = 0$, convergence = False, $meanAgg = 0$
  while !convergence **do**
    server.receive(1) ▷ Reception of the first packet that contains the parameters (d_p, np)
    server.receive($np - 1$) and store agg_{i+n}, $\forall i \in \{1, np\}$
    $n = n + np$
    $meanAgg = \frac{\sum_{j=1}^{n} agg_j}{n}$
    $S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (agg_i - meanAgg)^2$
    $convergence = \left(n \geq \frac{Z^2 \times S^2}{E^2}\right)$
    server.send(convergence)
  end while
  Add $meanAgg$ to vector MeanAgg
  Add d_p to vector D
  server.send($meanAgg$)
end while

Output: (MeanAgg, D) ▷ Returns two vectors: the "MeanAgg" and "D"

Cross traffic nature

Having already computed the mean aggregation levels of the probe traffic, we now proceed with the cross-traffic type estimation. Our intuition and reasoning to assess the cross traffic nature are based on the idea that if the cross traffic does not aggregate its frames, its channel access time becomes constant in average when the channel load increases. This time is defined here as the time the cross traffic uses the channel between two successive probe traffic accesses. It corresponds to $h + h$ in Figure 5.2 (cf. Chapter 5). The cross traffic access time depends only on the successive number of times it accesses the channel and the time to transmit a single frame. When the two buffers (cross and probe) are non-empty, it becomes independent of the number of frames/packets in these buffers and thus on the probe and traffic loads. On the contrary, when the cross traffic aggregates its frames, this time increases with the load as A-MPDU contains more aggregated frames.

We use this observation to detect the nature of the cross traffic. The mean cross traffic access time is denoted T_C. If this time is constant with d_p and with the load, we can express the mean aggregation level for the probe traffic, $meanAgg$, through a fixed point equation. This equation is formulated as:

$$meanAgg = min\left(AMPDU, \frac{f(meanAgg)}{d_p} + \frac{T_C}{d_p}\right) \tag{6.1}$$

where AMPDU is the maximum A-MPDU length (we note that $AMPU = AMPDU_P$ for the *Ideal server* model and $AMPDU = AMPDU_{AP}$ for the *Wireless server* model)

The rationale of this equation is that the mean number of aggregated frames ($meanAgg$ on the left-hand side of the equation) is equal to the number of frames received at the buffer since the last probe transmission. In Figure 5.2, we would get $X_{n+1} = \frac{f(X_n)}{d_p} + \frac{h+h}{d_p}$. Substituting (X_n, X_{n+1}) by the mean aggregation ($X_n = X_{n+1} = meanAgg$) and assuming that the cross traffic access time is constant ($(h + h)$ becomes T_C for this example), we obtain the Equation 6.1.

To verify this conjecture, we plot in Figure 6.2 the mean cross traffic access time, T_C (computed according to equation 6.1), as a function of the probe packet gap. Based on these results, it appears clearly that T_C varies slightly with d_p and with the load when the $BTF > 0.25$.

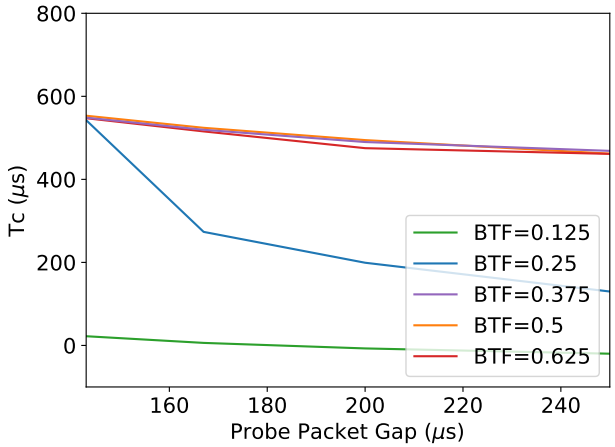


Figure 6.2: T_C versus Probe packet gap for non-aggeragted cross traffic

This approach is described in Algorithm 3. It takes as input the mean aggregation level for the probe traffic and computes T_C, according to equation 6.1, for each probe packet gap d_p. It keeps only values of T_C for which $\frac{f(meanAgg)}{d_p} + \frac{T_C}{d_p}$ is less than $AMPDU$. Then, it returns the percentage of variation between the maximum and minimum of these values. In Algorithm 6, this percentage is compared to a given threshold to determine if T_C can be considered as constant (cross traffic does not aggregate its frames) or not (cross traffic aggregates its frames).

BTF estimation

Having already computed the mean aggregation levels and discussed how to detect the cross traffic nature, we now detail how to estimate the network load of a wireless channel via its BTF.

Two algorithms are proposed to estimate the BTF. They are both based on the comparison between the mean aggregation levels given by our Markov chains and the measures made for different probe traffic batches.

Algorithm 4 computes the mean error as the sum of the difference between theoretical values obtained by the models and the measures. It

Algorithm 3 Percentage Increase Algorithm

Input: MeanAgg=$\left(meanAgg(d_{p_{min}}), ..., meanAgg(d_{p_{max}})\right)$,
D=$(d_{p_{min}}, ..., d_{p_{max}})$
$TC\_VECTOR = \emptyset$
for all $d_p \in D$ **do**
  Compute T_C:
  $T_C(d_p) = d_p * meanAgg(d_p) - f(meanAgg(d_p))$
  if $\frac{f(meanAgg)}{d_p} + \frac{T_C}{d_p} < AMPDU$ **then**
    Add $T_C(d_p)$ to vector TC_VECTOR
  end if
end for
$\%increase \leftarrow \frac{max(TC\_VECTOR) - min(TC\_VECTOR)}{min(TC\_VECTOR)} * 100$
Output: $\%increase$

returns the BTF that minimizes this error. The considered levels of BTF are parameters of the algorithm. It returns a BTF for the model based on aggregated competing traffic (w/A) and a BTF for the model based on non-aggregated cross traffic (wo/A).

Algorithm 4 BTF Error Based Method Algorithm

Input: MeanAgg=$\left(meanAgg(d_{p_{min}}), ..., meanAgg(d_{p_{max}})\right)$,
D=$(d_{p_{min}}, ..., d_{p_{max}})$
for each $case \in \{w/A, wo/A\}$ **do**
  for $bt \in$ levels of BTF **do**

$$Error_{(bt,case)} = \frac{1}{Size(MeanAgg(.))} \sum_{d_p=d_{p_{min}}}^{d_{pmax}} \mid model_{(case,d_p,bt)} - meanAgg(d_p) \mid$$

  end for
Output: $(\text{argmin}_{bt}\, Error_{(bt,w/A)}, \text{argmin}_{bt}\, Error_{(bt,wo/A)})$
end for

Algorithm 5 considers a notion of score to infer the BTF. For each probe packet gap d_p and a given model, the BTF that minimizes the error scores one. For each model, the final BTF will be the one that maximizes this score.

Algorithm 5 BTF Score Based Method Algorithm

Input: MeanAgg=$\left(meanAgg(d_{p_{min}}), ..., meanAgg(d_{p_{max}})\right)$,
D=$(d_{p_{min}}, ..., d_{p_{max}})$
for $d_p \in D$ **do**
  for each $case \in \{w/A, wo/A\}$ **do**
    for $bt \in$ all levels of BTF **do**
      $Error(case, d_p, bt) = \frac{1}{Size(meanAgg(.))} \mid model(case, d_p, bt) - meanAgg(d_p) \mid$
    end for
  end for
  $Score(\text{argmin}_{bt,case}\, Error(case, d_p, bt), case) += 1$
end for
Output: $(\text{argmax}_{bt}\, Score(bt, w/A), \text{argmax}_{bt}\, Score(bt, wo/A))$

Finally, Algorithm 6 gives the final output of our method as follows. First, it evaluates if the BTF is less or equal to 0.25. We have observed that, for such a low load, it is difficult to distinguish precisely the nature of the traffic since the cross traffic tends to have very small mean aggregation levels. In this case, no precision on the aggregation is given. If the BTF is greater than 0.25 at least for one of the two models, it computes the percentage of variation of T_C with Algorithm 3 and compares it to a threshold T. It determines if the cross traffic access time can be considered as constant and consequently if the cross traffic aggregates or not. For both cases, the corresponding BTF is returned with the cross traffic nature.

Having fully described the BTF and cross traffic nature estimation approach, we now provide its summary in a flowchart form in Figure 6.3).

Algorithm 6 BTF Estimation Algorithm

Input: T:Threshold, E: acceptable standard error of the mean

(MeanAgg, D) = Server Procedure(E, Z)
$PI = PercentageIncrease(MeanAgg, D)$
$BTF_{w/A}, BTF_{wo/A}$ = Error Computation Procedure(MeanAgg,D)
$BTF1_{w/A}, BTF1_{wo/A}$ = Score Computation Procedure(MeanAgg,D)
if $(BTF_{w/A} \leq 0.25 \; or \; BTF1_{w/A} \leq 0.25) \; and \; (BTF_{wo/A} \leq 0.25 \; or \; BTF1_{wo/A} \leq 0.25)$ **then**
    return $\{wo/A, BTF \leq 0.25\}$
else if $0 < PI < T$ **then**
    return $\{wo/A, BTF > 0.25\}$
else
    return $\{w/A, BTF_{w/A}\}$
end if

As a summary, in this section, we introduced the algorithmic details of the system FAM. We start by installing a dedicated client application on the probe traffic node, that sends a probing flow to the server. This latter uses the packet reception time to estimate the mean aggregation levels that, in turn, will serve to discern the BTF and convey the network nature. We recall that the computation of the frame aggregation level follows the threshold-based method detailed in Chapter 4. The evaluation of this method is detailed in the following section.

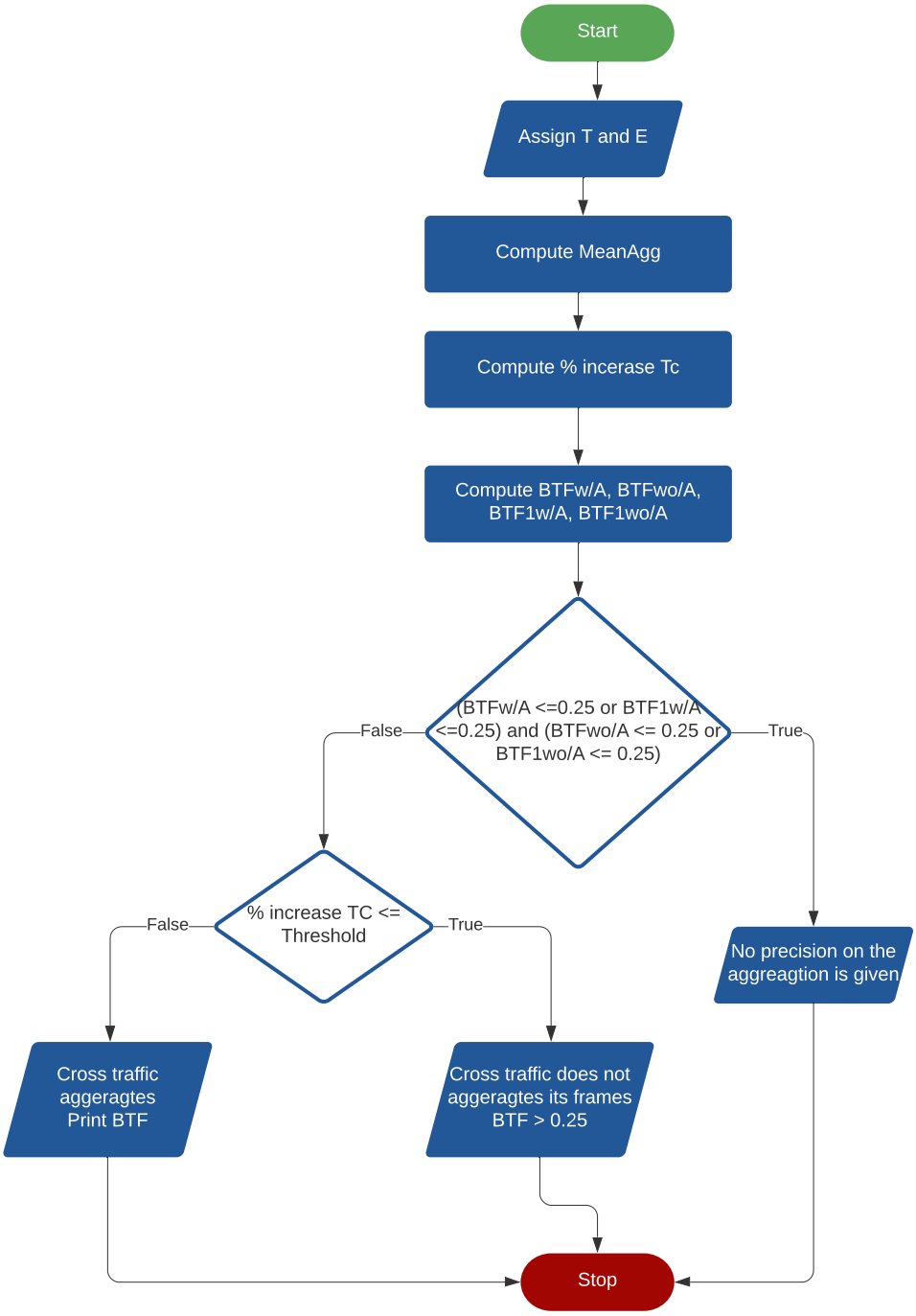


Figure 6.3: FAM flowchart

6.2 Performance evaluation

In this section, we evaluate the accuracy of the proposed method FAM by relying mainly upon the same ns-3 scenarios used in the previous chapter (Chapter 5) that expose different network parameters, different IEEE 802.11 amendments, different traffic patterns, and the same experimental test-bed. Moreover, we validate the method under another ns-3 scenario that covers the case where the network traffic is a mix of aggregated and non-aggregated traffic. We also evaluate the accuracy of our approach with the measurements made during a real-world trace-driven simulation.

The parameters of our method are as follow. Similar to the analytical models, we compute the four Markovian models for six cross traffic loads/BTF levels: 0, 0.125, 0.25, 0.375, 0.5, and 0.625 ranging from low to high levels of BTF. In algorithm 2, the FAM method uses a 95% confidence interval ($Z = 1.96$) and an error $E = 0.05$. The CLT results indicated that the relevant number of packets to send increases proportionally to the probe rate. The number of sent packets ranges between 600 and 3800.

In algorithm 6, the threshold T is chosen according to an empirical method. Our simulations and experiments have shown that a value of $T = 200\%$ offers good performances. For instance, we take different values for the parameter T that depend on the device's transmission rate. These values are known by FAM.

FAM validation for Ideal server model with aggregated cross traffic scenarios

In this section, we assess the effectiveness of FAM when relying on the theoretical aggregation levels returned by the *Ideal server* models when the cross traffic implements or not the frame aggregation feature. As noted earlier, FAM relies on the aggregated and non aggregated versions outcomes of the model *Ideal server* in order to infer the BTF and the exact type of traffic.

Simulation - Two STAs

We now evaluate the capacity/ability of FAM to infer the BTF and cross traffic nature for the scenario depicted in Figure 5.7 under the 802.11n amendment.

Table 6.1 presents the estimated BTF and cross traffic nature for the considered scenario. The *ground truth* column gives the real value of the BTF. In the table, 'S' indicates the BTF and cross traffic nature that were set in the simulations, and 'A' presents the value returned by our algorithm FAM. We note that FAM finds the good result when the two letters ('S'/'A') are in the same box. As FAM does not return the cross traffic nature when the load is lower than 0.25, we set 'S' and 'A' in the two corresponding boxes (aggregated and non-aggregated cross traffic) in this case.

Based on these results, it is noticeable that FAM provides the good BTF for all the cases except for the fifth case: BTF=0.5, where the frame

aggregation scheme is enabled for the cross traffic. This error is only of 0.125 (approximately 10%). In addition, for all the cases, FAM is able to infer that the cross traffic is aggregated (when BTF> 0.25).

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375		S/A				
0.5		A	S			
0.625				S/A		

Table 6.1: BTF and cross traffic nature estimations for ns-3 simulations - Two STAs

Simulation - Five STAs

We here reuse the scenario depicted in Figure 5.9 where the DL cross traffic is sent to four concurrent cross traffic nodes instead of a single node. Table 6.2 lists the estimated BTF and cross traffic nature returned by FAM for this scenario. Similar to previous results, we notice that FAM provides the good BTF for all the cases except for the seventh case: BTF=0.5. For all the cases, it is able to infer that the cross traffic is aggregated (when BTF> 0.25).

It is useful to recall that FAM seeks to infer the BTF based on six levels of loads ranging from 0 to 0.625 with a resolution of 0.125. It tends to classify the APs into three categories according to the network load: low, medium, and high network loads to detect the APs that are less loaded. In the following, we intentionally test the proposal under other values of BTFs. In Table 6.2, we ran the ns-3 simulations with two new network loads: 0.15 and 0.4. For BTF=0.15, the method returns that the BTF is less or equal to 0.25, and the cross traffic is aggregated, which is true. For BTF = 0.4, the method returns that the BTF = 0.375 since it is the nearest value to 0.4. These results, therefore, demonstrate our method's accuracy and its possibility to adapt other network load levels.

We should bear in mind that if an application needs better resolution, we can adapt our method according to this new resolution.

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.15	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375		S/A				
0.4		S/A				
0.5		A	S			
0.625				S/A		

Table 6.2: BTF and cross traffic nature estimations for ns-3 simulations - Five STAs

Experimental validation

We now reveal the ability of FAM to infer the BTF and cross traffic nature when we reuse the experiment illustrated in Figure 5.11. Table 6.3 presents the estimated BTF and cross traffic nature for this experiment. In the table, 'Exp' indicates the BTF and cross traffic nature that were set in the experiments, and 'A' presents the value returned by our algorithm FAM.

Table 6.3: BTF and cross traffic nature estimations for test-bed experiment

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	Exp/A				Exp/A	
0.125	Exp/A				Exp/A	
0.25	Exp/A				Exp/A	
0.375	A	Exp				
0.5		A	Exp			
0.625			A	Exp		

Results show that the method finds good results for the three lowest BTFs and slightly underestimates the BTF for the last three cases (O.375, 0.5, and 0.625). This error is only of 0.125 (approximately 10%), and FAM returns the precise nature of the cross traffic for all the cases. Overall, the experiment results show that the designed method can help capture the BTF and the traffic nature in practice.

Simulation - 802.11ax traffic

In this scenario, we study the methodology's accuracy when we change the underlying Wi-Fi network from 802.11n to 802.11ax in the 2.4 GHz band. As previous, all the nodes (AP and stations) operate at two data rates, 286.8$Mbps$ (corresponding to HE-MCS11 with a bandwidth of 20MHz) and 2402$Mbps$ (corresponding to HE-MCS11 with a bandwidth of 160MHz). The network topology is composed of an AP and two nodes. A probe traffic node sends the probe traffic, and another node receives the aggregated cross traffic from the AP. We run our FAM method while considering an empirical threshold T equal to 100 for transmission rate= 286.8$Mbps$ and T equal to 40 for transmission rate=2402$Mbps$. This threshold is adapted according to the transmission rate.

The results of FAM are shown in Table 6.4. It can be seen that FAM infers reasonably well the BTF values for all the cases except the fifth case BTF=0.5 for the first table where the transmission rate =286.6$Mbps$ and the fourth case BTF=0.375 for the second table where the transmission rate = 2402$Mbps$ (with typically 10% of error). Besides, it discerns the right cross traffic nature for all the cases. These results highlight that our approach is still relevant under the high data rates proposed by the last standard amendment IEEE 802.11ax.

Ground Truth BTF	Estimated BTF and cross traffic nature					
	cross traffic aggregates				cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375		S/A				
0.5		A	S			
0.625				S/A		

(a) Transmission rate = 286.8 Mbps

Ground Truth BTF	Estimated BTF and cross traffic nature					
	cross traffic aggregates				cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375		S	A			
0.5			S/A			
0.625				S/A		

(b) Transmission rate = 2402 Mbps

Table 6.4: BTF and cross traffic nature estimations for ns-3 simulations - 802.11ax traffic

Real-world trace-driven evaluation

Because of the unlicensed nature of the Wi-Fi network, the network load in real-environments is decided not only by the concurrent traffic on the same AP but also by the interference on the same or overlapped channel. In order to validate the accuracy of the *Ideal server* model and the method FAM under real-environment conditions, we conducted a trace-driven ns-3 scenario. For collecting the trace, we ran a sniffer capture using *Wireshark* to record all the traffic transmitted on channel 1 of the largest train station "Part Dieu" in the French city Lyon. We selected this popular station suited in the downtown as many customers used free Wi-Fi while waiting for their trains. With the densely deployed Wi-Fi networks in the downtown area, we can not only capture the traffic inside this particular station but also collect the traffic from several APs nearby. This capture lasted 200 seconds. It is therefore long enough to collect a trace that includes a variety of channel conditions, mobility, multiple APs, multiple user stations, Wi-Fi, and non-Wi-Fi interference. This real-word captured dataset is then injected in ns-3 as the cross traffic. By adjusting a time factor that will be multiplied by the cross traffic inter-arrivals, we obtained four different network loads: 0.11, 0.328, 0.459, and 0.729.

The results of FAM are shown in Table 6.6. We can observe that FAM estimates well the BTF values and the cross traffic nature for the cases 0.11, 0.328 and 0.729 cases and slightly underestimates the BTF for the case 0.459 since the nearest value to 0.459 is 0.5, not 0.375.

Overall, these results illustrate that despite using new traffic loads and real-world traffic, the proposal still deliberates good estimates under the considered scenario.

Table 6.6: BTF and cross traffic nature estimations for ns-3 simulations - Real-world trace-driven simulation

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0.11	S/A				S/A	
0.328		S/A				
0.459		A	S			
0.729				S/A		

FAM validation for Ideal server model with non-aggregated scenarios

We now examine the proposed approach's accuracy when relying on the *Ideal server* model. We consider non-aggregated cross traffic. The considered scenario consists of two co-located IEEE 802.11n and IEEE 802.11g WLANs operating on the 2.4 GHz band. The 802.11n WLAN is composed of an AP and a node that sends the probe traffic, while the 802.11g is composed of an AP and a node that receives the non-aggregated cross traffic.

Table 6.7 lists the estimated BTF and cross traffic nature returned by FAM for each of the BTF levels. For all the cases, we observe that the method performs extremely well and provides the expected level of the channel load and the precise nature of the cross traffic.

Table 6.7: BTF and cross traffic nature estimations for ns-3 simulations - Non-aggregated cross traffic

Ground Truth BTF	Estimated BTF and cross traffic nature					
	cross traffic aggregates				cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375						S/A
0.5						S/A
0.625						S/A

Additional simulation results under the Ideal server model

For the sake of completeness, we also ran the *Ideal server* models and our method FAM using a mix of aggregated and non-aggregated cross traffic scenarios (based on 802.11n and 802.11g cross traffic).

We investigate the BTF of a mixed cross traffic. We present two scenarios. The network topology is depicted in Figure 6.4. We simulate a scenario where the mixed cross traffic is composed of 80% of aggregated traffic and 20% of non-aggregated traffic (Scenario 1) and inversely (Scenario 2).

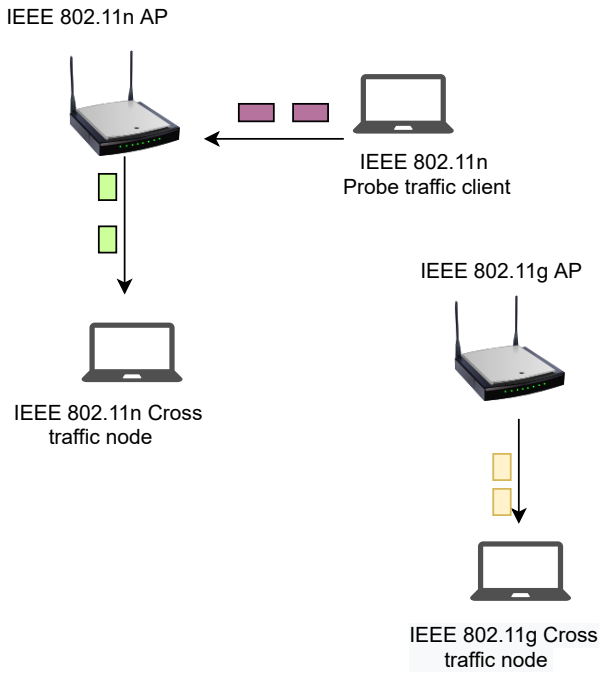


Figure 6.4: ns-3 simulation scenario for the mixed cross traffic

Table 6.8 and Table 6.9 provide the estimations made by scheme FAM. Table 6.8 shows that the method is able to find good results for the three lowest BTFs, 0, 0.125, and 0.25. Since the cross traffic is composed of 80% of aggregated traffic, the method tends to predict reasonably well the nature of the cross traffic (aggregated) for the highest BTFs 0.375, 0.5, and 0.625; however, the predicted values were slightly overestimated.

Table 6.9 reveals that FAM returned the same results as the cases of 100% non-aggregated cross traffic, except for the fourth case: $BTF = 0.375$. Note that we have $0.375 * 0.8 = 0.3$ of non-aggregated traffic in this case. It is close to the case with a BTF of 0.25 of cross traffic with no aggregation for which it has been shown that the access time T_C varies significantly (Figure 6.2).

Ground Truth BTF	Estimated BTF and cross traffic nature					
	cross traffic aggregates				cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375		S	A			
0.5			S	A		
0.625				S/A		

Table 6.8: BTF and cross traffic nature estimations for ns-3 simulations with mixed cross traffic, 80% of aggregated traffic and 20% of non-aggregated traffic

Table 6.9: BTF and cross traffic nature estimations for ns-3 simulations with mixed cross traffic, 20% of aggregated traffic and 80% of non-aggregated traffic

Ground Truth BTF	Estimated BTF and cross traffic nature					
	cross traffic aggregates				cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375			A			S
0.5						S/A
0.625						S/A

Note that in order to investigate the robustness of our approach, we explored two other ns-3 scenarios where the mixed cross traffic is composed of 60% of aggregated traffic and 40% of non-aggregated traffic (first scenario) and inversely (second scenario). The corresponding results are not presented in this manuscript. However, they show that FAM's accuracy is similar to the two previous mixed scenarios.

FAM validation for Wireless server model with aggregated cross traffic scenarios

In this section, we assess the effectiveness of FAM when relying on the theoretical aggregation levels returned by the *wireless server* models when the cross traffic uses or not the frame aggregation feature. We want to evaluate the ability of FAM to infer the BTF and the cross traffic type when we consider aggregated cross traffic scenarios.

Simulation - Same MCS

We now study the case when all the nodes (AP and STAs) use the same MCS index (MCS15 corresponding to 144.4 Mbps). We reuse the three-STAs scenario of Figure 4.7.

Results presented in Table 6.10 show that the method finds the good results for all the levels of BTF.

Table 6.10: BTF and cross traffic nature estimations for ns3 simulations where the cross traffic is aggregated - Same MCS

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375		S/A				
0.5			S/A			
0.625				S/A		

Simulation - Different MCS

Once again, we consider the scenario depicted in Figure 4.7 to study the accuracy of our approach under nodes (AP and STAs) that have different data rates. We use the MCS15 (144.4 Mbps) for the AP to send the DL probe traffic and the cross traffic and the MCS 11 (57.8 Mbps) for the probing node (scenario 1). Then, we use MCS15 for the AP and MCS13 (115.6 Mbps) for the probe traffic node (scenario 2).

Table 6.11 shows the results of scenario 2. We can observe that the method finds the good results for the five BTFs 0, 0.25, 0.375, 0.5, and 0.625 and slightly overestimates the BTF for the case 0.125. This error is only of 0.125 (approximately 10%) and is due to the fact that we neglect some protocol aspects, such as the beacons sent by the AP. We note that we obtain similar results as Table 6.11 for the scenario 2.

Overall, we notice that despite having STAs with significantly different data rates, the method FAM is still able to deliver reasonable predictions for the BTF.

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S	A			S	
0.25	S/A				S/A	
0.375		S/A				
0.5			S/A			
0.625				S/A		

Table 6.11: BTF and cross traffic nature estimations for ns3 simulations where the cross traffic is aggregated - Different MCS

Simulation - Exponential On/Off cross traffic

Instead of taking into account deterministic cross traffic, this scenario considers cross traffic generated using exponentially distributed On/Off periods.

Table 6.12 shows the results returned by FAM for the corresponding scenario. It can be seen that FAM predicts reasonably well the BTF values and the cross traffic nature for the 0, 0.125, 0.25, 0.375 and 0.625 cases and slightly overestimates the BTF for the 0.5 case.

Overall, despite a slight discrepancy in estimating the precise level of BTF in some cases, the results returned by the method are in good agreement with those provided by the simulations.

Table 6.12: BTF and cross traffic nature estimations for ns3 simulations - Exponential On/Off aggregated cross traffic

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375		S/A				
0.5			S	A		
0.625				S/A		

Simulation - 802.11ax traffic

In order to further assess the validity of the approach, in this scenario, we study FAM's accuracy when relying on the *wireless server* model based on 802.11ax aggregated probe and cross traffics in the 2.4 GHz band. All the nodes (AP and stations) operate at HE-MCS11 corresponding to 286.8$Mbps$. The network topology is the same as the scenario depicted in Figure 4.7. We run our FAM method while considering an empirical threshold T equal to 100.

The results of FAM are shown in Table 6.13. It can be seen that FAM predicts reasonably well the BTF values for all the cases except the fifth case: BTF=0.5 with 10% of error. In addition, it predicts well the nature of the traffic for all the cases. We, therefore, can conclude that FAM provides results that match well with the simulations under 802.11ax when relying on theoretical estimates provided by *Wireless server* models.

Table 6.13: BTF and cross traffic nature estimations for ns3 simulations - 802.11ax traffic

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S/A				S/A	
0.25	S/A				S/A	
0.375		S/A				
0.5		A	S			
0.625				S/A		

FAM validation for Wireless server model with non-aggregated cross traffic scenarios

We use this section to validate the accuracy of our method FAM when we consider the *wireless server* models' outcomes based on aggregated and non-aggregated cross traffic. We here consider non-aggregated cross traffic scenarios. Analogously to the previous section, we validate the method by comparing its estimation to the results delivered by the network simulator.

Simulation - Three STAs

We here evaluate FAM's accuracy when we used the scenario described in Section 5.4, where an 802.11n WLAN and an 802.11g WLAN were deployed.

Table 6.14 shows that FAM estimates well the BTF and the type of the traffic for the cases 0, 0.25, 0.3.75, 0.5 and 0.625 and overestimates the BTF for the case BTF=0.125. Unfortunately, the Wi-Fi link may experience several transient effects in practice. Since the analytical model neglects some protocol aspects such as beacons and MAC losses, it underestimates the aggregation level for some probe packet gaps. As a result, FAM overestimates the BTF which is typically a better outcome for most applications than an aggressive underestimate.

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S	A			S	
0.25	S/A				S/A	
0.375						S/A
0.5						S/A
0.625						S/A

Table 6.14: BTF and cross traffic nature estimations for ns3 simulations, non-aggregated cross traffic - Three STAs

Simulation - Six STAs

We now consider the case where we vary the number of cross traffic nodes from one to four.

Table 6.15 shows the corresponding results. The results of four cross traffic STAs show similar tendencies to the results with a single cross traffic STA. Based on these results, we can observe that FAM finds the same results as the previous case. In summary, we find that our method provides relative errors generally smaller than 10%.

Ground Truth BTF	Estimated BTF and cross traffic nature					
	Cross traffic aggregates				Cross traffic does not aggregate	
	≤ 0.25	0.375	0.5	0.625	≤ 0.25	> 0.25
0	S/A				S/A	
0.125	S	A				
0.25	S/A				S/A	
0.375						S/A
0.5						S/A
0.625						S/A

Table 6.15: BTF and cross traffic nature estimations for ns3 simulations for the case where the cross traffic is non-aggregated - Six STAs

6.3 Discussion and conclusions

Our purpose in this chapter, was to propose a novel method, named FAM, that allows a vanilla device, in particular a smartphone, to estimate the network load via its BTF based on the observed frame aggregation level and infers if the current traffic aggregates its frames or not. This method was built by relying on the measurement of the actual frame aggregation levels and the theoretical ones returned by analytical models.

We studied a variety of scenarios to assess the effectiveness of our method by comparing its outcomes with those delivered by the ns-3 simulator, test-bed experiment, and a real-world trace-driven evaluation. We considered several network topologies, different IEEE 802.11 amendments, various levels of the channel load, several traffic patterns as well as scenarios where the cross traffic is a mix of aggregated and non-aggregated traffics.

Overall, from a general viewpoint and for almost all the tested scenarios, we find that FAM estimates with a satisfactory degree of precision the Busy Time Fraction and the traffic nature with at most 10% of errors. Based on six level of loads from 0 to 0.625 with a resolution of 0.125, FAM seeks to classify the APs into three classes according to the network load: low, medium, and high loaded APs. It, therefore, compares APs with each other to detect the APs that are less loaded and provide guidance to hierarchize different WLANs. However, we should note that if an application needs better resolution, FAM can be adapted according to this new resolution. Although the accuracy is not perfect, we believe that FAM is a practical scheme for enabling mobile clients to crowd-sense Wi-Fi performances, providing guidance to hierarchize different WLANs by potentially finding more optimal association schemes for the stations, and inferring the better access point selection. Moreover, we believe that applying the utilization of FAM to the handover decision criterion would enhance the throughput performance of mobile STAs.

In this work, there are several possible improvements that, mainly due to time constraints, we were not able to finish. In order the compute the aggregation level, FAM sends a sequence of probe traffic from the client to the server. While this active probing method undoubtedly results in finding good performance, the number of sent probes for each probe packet gap is far from optimal and the overhead of the proposal was neglected in the evaluation. Indeed, for instance, the method tends to be expensive in terms of bandwidth as we use a large set of packet intervals with fine granularity in order to test it. However, we believe that optimizations are probably possible by significantly reducing the number of sent packets by reducing the number of probing intervals (probing rates). These future extensions would render our approach of greater practical use.

Finally, as we already pointed out in this chapter, in order to predict the cross traffic type, our method relies on an empirical value of the threshold T that is accordingly adapted as a function of the data rate. Although we made sure to constantly evolve our system, this thesis does not provide a methodology for this adaptation. Hence, a future extension would tackle this issue as we believe that there are more optimal methods to adopt the value of this threshold.

7 Conclusion

This chapter wraps up this manuscript by reminding the addressed issues, highlighting the main contributions made thus far, and discussing potentially viable perspectives.

Concluding Remarks

In this dissertation, we considered the task of evaluating the possibility/-capacity for a vanilla Wi-Fi client, typically an Android smartphone, to infer the Wi-Fi network load from local measurements for the purpose of optimal AP selection. This estimation is proceeded in the user space without making modifications to network equipment (APs and STAs). This crucial task has an important impact on the performance of Wi-Fi networks in public areas where the Wi-Fi networks are managed by independent entities.

The contributions of this thesis can be summarized into several main axes:

1. We proposed analytical models based on Markov chains that demonstrate how one could induce the theoretical frame aggregation levels of targeted probe traffic sent from a client to a server across several scenarios.
2. We revealed how the frame aggregation scheme embodies a rich set of Wi-Fi link properties that can be utilized to discern the network load. We illustrated through ns-3 simulations, test-bed experiments, and a real-world trace-driven simulation that frame aggregation levels correlate extremely well with the expected BTF across a variety of challenging scenarios and environments. Moreover, we demonstrated that the throughput that a joining device could get depends on whether the competing traffic uses frame aggregation or not.
3. We proposed a novel method FAM that leverages both the theoretical frame aggregation levels (returned by the analytical models) and the measured ones to estimate not only the BTF but also the nature of the traffic.
4. We demonstrated the robustness of the analytical Markovian models and FAM across varying network topologies, IEEE 802.11 standard amendments, traffic patterns, and levels of the wireless channel load. We showed that this proposal returns results with a satisfactory degree of precision.

Initially, during the first months of the dissertation, after some initial state of the art, our main goal was to evaluate the two Wi-Fi-based available bandwidth estimation tools WBest+ [20] and AIWC [11] in the presence of the frame aggregation scheme. For WBest+, the experiments were conducted using its open-source code.[1] Despite having access to the theoretical way the authors of AIWC managed to evaluate the

1: https://github.com/afarshad/WBestPlus

available bandwidth, we could not get our hands on the real tools they used to evaluate the effectiveness of their method. We, therefore, had to recreate it by following the paper's instructions as best as we could. The two papers highlighted that their approaches should give precise estimations. Unfortunately, we could not reproduce the results promised by the authors of these two papers leading us to believe that studying the analytical behavior of the frame aggregation as a function of the network load has to be proposed to settle these approaches.

Over the course of this thesis, we proposed several simple and versatile analytical Markovian models in accordance with the IEEE 802.11 standard specific to the application of BTF estimation under the IEEE 802.11 frame aggregation scheme. We modeled and simulated scenarios in which a device induces the mean aggregation levels of an aggregated deterministic probe traffic competing with a cross traffic that can aggregate or not its frames in the user space.

In the first analytical solution, we proposed a mathematical model to evaluate and predict the mean aggregation level of the UL probe traffic for given network traffic. In this model called *Ideal server*, we have assumed that the connection between the server and the AP is ideal. We modeled it as if the server were implemented on the AP. Even if this network topology is rare, the results obtained by the model helped us to better understand the relationship between the aggregation level of the probe traffic and the network BTF. For this model, we have proposed two Markov chains based on whether the cross traffic aggregates its frames or not.

The system described for the *Ideal server* model is limiting as it is based on a hypothesis that can be deemed too restrictive or unrealistic in some scenarios. Nevertheless, this model provided us with solid understandings that help the development of the subsequent model that would not have been possible without it. The second solution proposed the *wireless server* model that relies on more real architecture system. In this model, the server is embedded on an additional wireless device associated with the AP. The paradigm shifts from the evaluation of the aggregation levels of the UL probe traffic to the aggregation levels of the DL probe traffic forwarded by the AP to the server.

Based on the theoretical frame aggregation levels returned by the models, we designed a network load estimation method FAM that relies on theoretical and measured frame aggregation levels in order to discern not only the BTF, but also the type of the traffic. From the findings, we can get the following engineering insights on the feasibility of a crowd-sensing platform for network load estimation.

- The accuracy of FAM is enough for the classification of the load in a few levels.
- When the majority of the competing traffic does not use frame aggregation, it is only possible to identify two load classes.
- The number of devices composing the DownLink competing traffic coming from the same AP does not influence the prediction.

This dissertation has considered the main methods utilized in Wi-Fi performance evaluation which are analytical modeling, experimentation, and simulation. Based on several simplifying assumptions, analytical

models tends to model the existing system with a given degree of fidelity. Simulations, on the other hand, are often put between experiments and models. They simply model a part of the real behavior of the network, but their degree of detail gets close to experimentation. At last, the experimentation has the less interrogated fidelity as it offers real-life insights.

Overall, based on simulations and experiments, we find that FAM estimates with a satisfactory degree of precision the BTF and the traffic nature with at most 10% of errors. We believe that such modeling approaches and the method can be used to reconfigure networks in real-time to balance the load among APs and share the available throughput among STAs.

Perspectives

The contributions of this dissertation can be extended in several directions. In the following, we present the directions that are worthy of exploring in the future.

Crowd-sensing

One of the directions for continuing the contributions presented in this thesis is the extension of the proposed method to be used in a crowd-sensing context. This future work would propose a Waze-inspired (Waze is a GPS-based mobile navigation application that collects traffic data from users and gives real-time traffic information to help the user adapt his route) crowd-sensing AP selection, where the cloud gathers the BTF estimation results from many smartphones and infers location-specific AP availability based on data fusion.

In such a case, collected data would be partial since only the smartphone that integrates the crowd-sensing application would sense the channel and perform the measurements. Given this limited sensing capability, inference of missing data is, therefore, necessary in order to get more accurate sensing results.

Furthermore, the BTF estimation tends to sample the BTF in a small time window. This estimation is generally sensitive to any temporal effect that occurred on the wireless channel. To take advantage of the estimated results, we have to study their applicability. The issue thus involves interpolating, i.e., predicting the estimation of a point between the results, and extrapolating, i.e., inferring future or past results based on given tests.

Association optimization

A logical next step regarding the best AP selection procedure is the optimization of the association of wireless STAs to APs. Indeed, in order to maximize the overall network throughput, for instance, through the load balancing, the proposed method would have to propose a solution to manage and optimize the actions of association and re-association. More precisely, the proposal would aim at diminishing the load of the

most loaded AP in the WLAN. This latter would not provide the same set of APs to all the users because in such a case everyone goes on the same AP. Instead, it would offer each user the more efficient subset of APs. The load sharing problem could integrate geographic aspects to optimize a given objective function which decreases with the distance traveled by users.

Evolution of 802.11

Are our analytical models still relevant under 802.11ac and 802.11ax? The skyrocketing evolution of 802.11 standards quickly renders some Wi-Fi performance evaluation techniques irrelevant or obsolete, pushing us to ask whether our method FAM and analytical models will have a practical use in a decade?

The 802.11ax introduces two new features that change how the nodes access the medium: the BSS coloring and OFDMA. The BSS coloring permits the STAs operating on the same channel, but in different BSSs to access the channel at the same time by using a different color for each BSS. The OFDMA allows neighboring nodes to simultaneously transmit on the same channel using different sub-carriers. While we systematically assessed the effectiveness of our proposal under the standard 802.11ax by using its current implementation on ns-3, we note that the ns-3.30 version used throughout this thesis does not include all 802.11ax features. For now (2021), ns-3.30 does not support the 802.11ac/ax MU-MIMO technique. We, therefore, need to revisit the analytical models and the method FAM under these new features. The BTF characterization in these cases may need intrinsically different perspectives.

Optimization of the handover

Wi-Fi mobility mechanisms are based on the notion of handover: when a STA moves from an AP coverage area to another, it is able to seamlessly connect to a new AP while keeping a good connection quality. Our final crowd-sensing application, based on mapping and measurements, could allow anticipating/knowing points in the network where handovers should be anticipated (within the same ESS or not). Comparison between the default method and an "informed" method could be established under several scenarios.

Appendix

A

Additional probabilities 1

$$\mathbb{P}\Big(S_{n+1} = APC | S_n = APP, X_{n+1} = l = 0, Y_{n+1} = m > 0, Z_{n+1} = q\Big) = \frac{1}{2}\mathbb{1}_{q>0} + \mathbb{1}_{q=0}$$

$$\mathbb{P}\Big(S_{n+1} = SP | S_n = APP, X_{n+1} = l = 0, Y_{n+1} = m, Z_{n+1} = q > 0\Big) = \frac{1}{2}\mathbb{1}_{m>0} + \mathbb{1}_{m=0}$$

$$\mathbb{P}\Big(S_{n+1} = APC | S_n = APC, X_n = X_{n+1} = 0, Y_{n+1} = m > 0, Z_{n+1} = q\Big) = \frac{1}{2}\mathbb{1}_{q>0} + \mathbb{1}_{q=0}$$

$$\mathbb{P}\Big(S_{n+1} = APP | S_n = APC, X_{n+1} = X_n = l > 0, Y_{n+1} = m, Z_{n+1} = q\Big) = \frac{1}{2}\mathbb{1}_{q>0} + \mathbb{1}_{q=0}$$

$$\mathbb{P}\Big(S_{n+1} = SP | S_n = APC, X_{n+1} = X_n = l, Y_{n+1} = m, Z_{n+1} = q > 0, \Big) = \frac{1}{2}\mathbb{1}_{l+m>0} + \mathbb{1}_{l+m=0}$$

$$\mathbb{P}\Big(S_{n+1} = APP | S_n = SP, X_{n+1} = l > 0, Y_{n+1} = l, Z_{n+1} = q\Big) = \frac{1}{2} \cdot \frac{1}{2}\mathbb{1}_{q>0,m>0} + \frac{1}{2}\mathbb{1}_{q>0,m=0} + \frac{1}{2}\mathbb{1}_{q=0,m>0} + \mathbb{1}_{q=0,m=0}$$

$$\mathbb{P}\Big(S_{n+1} = APC | S_n = SP, X_{n+1} = l, Y_{n+1} = m > 0, Z_{n+1} = q, \Big) = \frac{1}{2} \cdot \frac{1}{2}\mathbb{1}_{q>0,l>0} + \frac{1}{2}\mathbb{1}_{q>0,l=0} + \frac{1}{2}\mathbb{1}_{q=0,l>0} + \mathbb{1}_{q=0,l=0}$$

$$\mathbb{P}\Big(S_{n+1} = SP | S_n = SP, X_{n+1} = l, Y_{n+1} = m, Z_{n+1} = q > 0\Big) = \frac{1}{2}\mathbb{1}_{m+l>0} + \mathbb{1}_{m+l=0}$$

B

Additional probabilities 2

$$\mathbb{P}\Big(S_{n+1} = APC | S_n = APP, X_{n+1} = l = 0, Y_{n+1} = m > 0, Z_{n+1} = q\Big) = \frac{1}{2}\mathbb{1}_{q>0} + \mathbb{1}_{q=0}$$

$$\mathbb{P}\Big(S_{n+1} = SP | S_n = APP, X_{n+1} = l = 0, Y_{n+1} = m, Z_{n+1} = q > 0\Big) = \frac{1}{2}\mathbb{1}_{m>0} + \mathbb{1}_{m=0}$$

$$\mathbb{P}\Big(S_{n+1} = APC | S_n = APC, X_n = X_{n+1} = i = l, Y_{n+1} = m > 0, Z_{n+1} = q\Big) = \frac{1}{3}\mathbb{1}_{q>0,l>0} + \frac{1}{2}\mathbb{1}_{q>0,l=0} + \frac{1}{2}\mathbb{1}_{q=0,l>0} + \mathbb{1}_{q=0,l=0}$$

$$\mathbb{P}\Big(S_{n+1} = APP | S_n = APC, X_{n+1} = X_n = i = l > 0, Y_{n+1} = m, Z_{n+1} = q\Big) = \frac{1}{3}\mathbb{1}_{q>0,m>0} + \frac{1}{2}\mathbb{1}_{q>0,m=0} + \frac{1}{2}\mathbb{1}_{q=0,m>0} + \mathbb{1}_{q=0,m=0}$$

$$\mathbb{P}\Big(S_{n+1} = SP | S_n = APC, Z_{n+1} = q > 0, X_{n+1} = X_n = i = l, Y_{n+1} = m, Z_{n+1} = q > 0\Big) = \frac{1}{3}\mathbb{1}_{l>0,m>0} + \frac{1}{2}\mathbb{1}_{l>0,m=0} + \frac{1}{2}\mathbb{1}_{l=0,m>0} + \mathbb{1}_{l=0,m=0}$$

$$\mathbb{P}\Big(S_{n+1} = APP | S_n = SP, X_{n+1} = l > 0, Y_{n+1} = m, Z_{n+1} = q\Big) = \frac{1}{3}\mathbb{1}_{q>0,m>0} + \frac{1}{2}\mathbb{1}_{q>0,m=0} + \frac{1}{2}\mathbb{1}_{q=0,m>0} + \mathbb{1}_{q=0,m=0}$$

$$\mathbb{P}\Big(S_{n+1} = APC | S_n = SP, X_{n+1} = l, Y_{n+1} = m > 0, Z_{n+1} = q\Big) = \frac{1}{3}\mathbb{1}_{q>0,l>0} + \frac{1}{2}\mathbb{1}_{q>0,l=0} + \frac{1}{2}\mathbb{1}_{q=0,l>0} + \mathbb{1}_{q=0,l=0}$$

$$\mathbb{P}\Big(S_{n+1} = SP | S_n = SP, X_{n+1} = l, Y_{n+1} = m, Z_{n+1} = q > 0\Big) = \frac{1}{3}\mathbb{1}_{l>0,m>0} + \frac{1}{2}\mathbb{1}_{l>0,m=0} + \frac{1}{2}\mathbb{1}_{l=0,m>0} + \mathbb{1}_{l=0,m=0}$$

Bibliography

[1] Cisco. *Cisco Annual Internet Report (2018–2023) White Paper*. 2020 (cited on page 1).

[2] M. Agiwal, A. Roy, and N. Saxena. 'Next Generation 5G Wireless Networks: A Comprehensive Survey'. In: *IEEE Communications Surveys Tutorials* 18.3 (2016), pp. 1617–1655. DOI: 10.1109/COMST.2016.2532458 (cited on page 1).

[3] K. Sood, S. Liu, S. Yu, and Y. Xiang. 'Dynamic access point association using Software Defined Networking'. In: *International Telecommunication Networks and Applications Conference (ITNAC)*. 2015, pp. 226–231. DOI: 10.1109/ATNAC.2015.7366817 (cited on page 2).

[4] M. Amer, A. Busson, and I. Guérin Lassous. 'Considering Frame Aggregation in Association Optimization for High Throughput Wi-Fi Networks'. In: *ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PE-WASUN)*. Montreal, QC, Canada, 2018, pp. 55–62. DOI: 10.1145/3243046.3243057 (cited on page 2).

[5] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos. 'Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey'. In: *Mobile Networks and Applications* 20 (2015), pp. 4–18. DOI: 10.1007/s11036-014-0533-8 (cited on page 2).

[6] R. K. Ganti, F. Ye, and H. Lei. 'Mobile crowdsensing: current state and future challenges'. In: *IEEE Communications Magazine* 49.11 (2011), pp. 32–39. DOI: 10.1109/MCOM.2011.6069707 (cited on page 3).

[7] IEEE. 'IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications'. In: *IEEE Std 802.11-1997* (1997), pp. 1–445. DOI: 10.1109/IEEESTD.1997.85951 (cited on page 8).

[8] 'IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band'. In: *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)* (2003), pp. 1–104. DOI: 10.1109/IEEESTD.2003.94282 (cited on page 9).

[9] 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'. In: *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)* (2021), pp. 1–4379. DOI: 10.1109/IEEESTD.2021.9363693 (cited on page 10).

[10] S. Zhu, A. Mohammed, and A. Striegel. 'A Frame-Aggregation-Based Approach for Link Congestion Prediction in WiFi Video Streaming'. In: *International Conference on Computer Communications and Networks (ICCCN)*. 2020, pp. 1–8. DOI: 10.1109/ICCCN49398.2020.9209675 (cited on page 13).

[11] L. Song and A. Striegel. 'Leveraging Frame Aggregation for Estimating WiFi Available Bandwidth'. In: *IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 2017, pp. 1–9. DOI: 10.1109/SAHCN.2017.7964908 (cited on pages 13, 20, 25, 26, 28, 35, 89).

[12] B. Melander, M. Bjorkman, and P. Gunningberg. 'A new end-to-end probing and analysis method for estimating bandwidth bottlenecks'. In: *IEEE Global Telecommunications Conference (Globecom)*. Vol. 1. 2000, pp. 415–420. DOI: 10.1109/GLOCOM.2000.892039 (cited on pages 20, 23).

[13] M. Jain and C. Dovrolis. 'Pathload: A Measurement Tool for End-to-End Available Bandwidth'. In: *Passive and Active Measurement Workshop (PAM)*. Mar. 2002, pp. 14–25 (cited on pages 20, 23).

[14] V. Ribeiro, R. Riedi, J. Navrátil, and L. Cottrell. 'PathChirp: Efficient Available Bandwidth Estimation for Network Paths'. In: *Passive and Active Measurement Workshop (PAM)*. Apr. 2003, pp. 1–11. DOI: 10.2172/813038 (cited on pages 20, 23).

[15] A. Johnsson, M. Bjorkman, and B. Melander. 'An Analysis of Active End-to-end Bandwidth Measurements in Wireless Networks'. In: *IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services*. 2006, pp. 74–81. DOI: 10.1109/E2EMON.2006.1651282 (cited on pages 20, 23).

[16] J. Strauss, D. Katabi, and F. Kaashoek. 'A Measurement Study of Available Bandwidth Estimation Tools'. In: *ACM SIGCOMM Conference on Internet Measurement (IMC)*. Oct. 2003, pp. 39–44. DOI: 10.1145/948205.948211 (cited on pages 20, 24).

[17] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk. 'Multifractal cross-traffic estimation'. In: *ITC Specialist Seminar on IP Traffic Measurement, Modeling, and Management*. 2000, pp. 15–1 (cited on pages 20, 24).

[18] N. Hu and P. Steenkiste. 'Evaluation and characterization of available bandwidth probing techniques'. In: *IEEE Journal on Selected Areas in Communications* 21.6 (2003), pp. 879–894 (cited on pages 20, 24).

[19] M. Li, M. Claypool, and R. Kinicki. 'WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks'. In: *2008 33rd IEEE Conference on Local Computer Networks (LCN)*. 2008, pp. 374–381. DOI: 10.1109/LCN.2008.4664193 (cited on pages 20, 25, 28).

[20] A. Farshad, M. Lee, M. K. Marina, and F. Garcia. 'On the impact of 802.11n frame aggregation on end-to-end available bandwidth estimation'. In: *IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 2014, pp. 108–116. DOI: 10.1109/SAHCN.2014.6990333 (cited on pages 20, 25, 28, 35, 89).

[21] S. Zhu, A. Mohammed, and A. Striegel. 'A Frame-Aggregation-Based Approach for Link Congestion Prediction in WiFi Video Streaming'. In: *International Conference on Computer Communications and Networks (ICCCN)*. 2020, pp. 1–8. DOI: 10.1109/ICCCN49398.2020.9209675 (cited on pages 20, 22).

[22] H. Lee, S. Kim, O. Lee, S. Choi, and S. Lee. 'Available Bandwidth-Based Association in IEEE 802.11 Wireless LANs'. In: *International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*. MSWiM '08. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2008, pp. 132–139. DOI: 10.1145/1454503.1454529 (cited on pages 20, 27, 28).

[23] F. Xu, X. Zhu, C. C. Tan, Q. Li, G. Yan, and J. Wu. 'SmartAssoc: Decentralized Access Point Selection Algorithm to Improve Throughput'. In: *IEEE Transactions on Parallel and Distributed Systems* 24.12 (2013), pp. 2482–2491. DOI: 10.1109/TPDS.2013.10 (cited on pages 20, 27, 28).

[24] B. A. Mah. 'Pchar: A tool for measuring internet path characteristics'. In: (2000) (cited on page 20).

[25] R. Carter and M. Crovella. *Dynamic Server Selection Using Bandwidth Probing in Wide-Area Networks*. Tech. rep. USA, 1996 (cited on pages 20, 21).

[26] C. Dovrolis, P. Ramanathan, and D. Moore. 'What do packet dispersion techniques measure?' In: *IEEE Conference on Computer Communications (INFOCOM)*. Vol. 2. 2001, 905–914 vol.2. DOI: 10.1109/INFCOM.2001.916282 (cited on pages 20, 21).

[27] A. B. Downey. 'Using Pathchar to Estimate Internet Link Characteristics'. In: *ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. SIGMETRICS '99. Atlanta, Georgia, USA: Association for Computing Machinery, 1999, pp. 222–223. DOI: 10.1145/301453.301582 (cited on page 20).

[28] K. Lai and M. Baker. 'Measuring Link Bandwidths Using a Deterministic Model of Packet Delay'. In: *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (ACM SIGCOMM)*. 2000, pp. 283–294 (cited on pages 20, 21).

[29] K. Lai and M. Baker. 'Measuring bandwidth'. In: *IEEE Conference on Computer Communications (INFOCOM)*. Vol. 1. 1999, pp. 235–245. DOI: 10.1109/INFCOM.1999.749288 (cited on pages 20, 21).

[30] V. Jacobson. *Pathchar: A tool to infer characteristics of Internet paths*. 1997. URL: ftp://ftp.ee.lbl.gov/pathchar (cited on page 20).

[31] *Iperf3*. [Online; accessed 18. Feb. 2019]. 2021. URL: https://software.es.net/iperf/ (cited on pages 20, 21).

[32] M. Muuss and T. Slattery. *Using Test TCP (TTCP) to Test Throughput*. 2005. URL: https://www.cisco.com/c/en/us/support/docs/dial-access/%20asynchronous-connections/10340-ttcp.html (cited on pages 20, 21).

[33] Hewlett Packard. *netperf(1): network performance benchmark - Linux man page*. URL: https://linux.die.net/man/1/netperf (cited on pages 20, 21).

[34] M. Mathis. *TReno Bulk Transfer Capacity*. 1999. URL: https://tools.ietf.org/%20html/draft-ietf-ippm-treno-btc-03 (cited on pages 20, 21).

[35] M. Allman. 'Measuring End-to-End Bulk Transfer Capacity'. In: *ACM SIGCOMM Workshop on Internet Measurement*. IMW '01. San Francisco, California, USA: Association for Computing Machinery, 2001, pp. 139–143. DOI: 10.1145/505202.505220 (cited on pages 20, 21).

[36] *Ookla Speedtest*. [Online; accessed 28. Jul. 2021]. 2021. URL: http://www.speedtest.net/ (cited on pages 20, 21).

[37] S. M. Bellovin. 'A Best-Case Network Performance Model'. In: *ATT Research, Tech. Rep* (1992) (cited on page 20).

[38] D. Murray, T. Koziniec, S. Zander, M. Dixon, and P. Koutsakis. 'An analysis of changing enterprise network traffic characteristics'. In: *Asia-Pacific Conference on Communications (APCC)*. 2017, pp. 1–6. DOI: 10.23919/APCC.2017.8303960 (cited on page 21).

[39] *AT & T Internet Speed Test*. [Online; accessed 28. Jul. 2021]. 2021. URL: http://speedtest.att.com/speedtest (cited on page 21).

[40] P. Nayak, S. Pandey, and E. W. Knightly. 'Virtual Speed Test: an AP Tool for Passive Analysis of Wireless LANs'. In: *IEEE Conference on Computer Communications (INFOCOM)*. 2019, pp. 2305–2313. DOI: 10.1109/INFOCOM.2019.873759 (cited on page 22).

[41] G. Aceto, F. Palumbo, V. Persico, and A. Pescape. 'Available Bandwidth vs. Achievable Throughput Measurements in 4G Mobile Networks'. In: *International Conference on Network and Service Management (CNSM)*. 2018, pp. 125–133 (cited on page 24).

[42] P. Ha, E. Zhang, W. Sun, F. Cui, and L. Xu. 'A Novel Timestamping Mechanism for Clouds and Its Application on Available Bandwidth Estimation'. In: *IEEE International Conference on Distributed Computing Systems (ICDCS)*. 2019, pp. 12–22. DOI: 10.1109/ICDCS.2019.00011 (cited on page 24).

[43] W. Eduardo Castellanos, Juan Guerri, and Pau Arce. 'Available Bandwidth Estimation for Adaptive Video Streaming in Mobile Ad Hoc'. In: *International Journal of Wireless Information Networks* 26 (May 2019), pp. 218–229. DOI: 10.1007/s10776-019-00431-0 (cited on page 24).

[44] H. Zhao, E. Garcia-Palacios, J. Wei, and Y. Xi. 'Accurate available bandwidth estimation in IEEE 802.11-based ad hoc networks'. In: *Computer Communications* 32.6 (2009), pp. 1050–1057. DOI: 10.1016/j.comcom.2008.12.031 (cited on page 24).

[45] P. Megyesi, A. Botta, G. Aceto, A. Pescapé, and S. Molnár. 'Challenges and solution for measuring available bandwidth in software defined networks'. In: *Computer Communications* 99 (2017), pp. 48–61. DOI: 10.1016/j.comcom.2016.12.004 (cited on page 24).

[46] D. Salcedo, C. Guerrero, and R. Martínez Aguilar. 'Available bandwidth estimation tools: Metrics, approach and performance'. In: *International Journal of Communication Networks and Information Security* 10 (Jan. 2018), pp. 580–587 (cited on page 24).

[47] S. H. Shah, K. Chen, and K. Nahrstedt. *Available Bandwidth Estimation in IEEE 802.11-based Wireless Networks*. 2003 (cited on page 24).

[48] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye. 'Bandwidth Estimation in Broadband Access Networks'. In: *ACM SIGCOMM Conference on Internet Measurement*. IMC '04. Taormina, Sicily, Italy: Association for Computing Machinery, 2004, pp. 314–321. DOI: 10.1145/1028788.1028832 (cited on page 25).

[49] H. K. Lee, V. Hall, K. H. Yum, K. I. Kim, and E. J. Kim. 'Bandwidth Estimation in Wireless Lans for Multimedia Streaming Services'. In: *IEEE International Conference on Multimedia and Expo (ICME)*. 2006, pp. 1181–1184. DOI: 10.1109/ICME.2006.262747 (cited on pages 25, 28).

[50] M. Li, M. Claypool, and R. Kinicki. 'Packet Dispersion in IEEE 802.11 Wireless Networks'. In: *IEEE Conference on Local Computer Networks (LCN)*. 2006, pp. 721–729. DOI: 10.1109/LCN.2006.322028 (cited on page 25).

[51] D. Skordoulis, Q. Ni, H.-h. Chen, A. P. Stephens, C. Liu, and A. Jamalipour. 'IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs'. In: *IEEE Wireless Communications* 15.1 (2008), pp. 40–47. DOI: 10.1109/MWC.2008.4454703 (cited on page 26).

[52] G. Nychis, S. Seshan, and P. Steenkiste. 'Using Your Smartphone to Detect and Map Heterogeneous Networks and Devices in the Home'. In: *ACM Workshop on Hot Topics in Wireless*. HotWireless '14. Maui, Hawaii, USA: Association for Computing Machinery, 2014, pp. 31–36. DOI: 10.1145/2643614.2643624 (cited on page 26).

[53] A. Nika, Z. Zhang, X. Zhou, B. Y. Zhao, and H. Zheng. 'Towards Commoditized Real-Time Spectrum Monitoring'. In: *ACM Workshop on Hot Topics in Wireless*. HotWireless '14. Maui, Hawaii, USA: Association for Computing Machinery, 2014, pp. 25–30. DOI: 10.1145/2643614.2643615 (cited on page 26).

[54] T. Zhang, A. Patro, N. Leng, and S. Banerjee. 'A Wireless Spectrum Analyzer in Your Pocket'. In: *International Workshop on Mobile Computing Systems and Applications*. HotMobile '15. Santa Fe, New Mexico, USA: Association for Computing Machinery, 2015, pp. 69–74. DOI: 10.1145/2699343.2699353 (cited on page 26).

[55] S. Lin, J. Zhang, and L. Ying. 'Crowdsensing for Spectrum Discovery: A Waze-Inspired Design via Smartphone Sensing'. In: *IEEE/ACM Transactions on Networking* 28.2 (2020), pp. 750–763. DOI: 10.1109/TNET.2020.2976927 (cited on page 26).

[56] S. Rosen, S. Lee, J. Lee, P. Congdon, Z. M. Mao, and K. Burden. 'MCNet: Crowdsourcing wireless performance measurements through the eyes of mobile devices'. In: *IEEE Communications Magazine* 52.10 (2014), pp. 86–91. DOI: 10.1109/MCOM.2014.6917407 (cited on page 26).

[57] J. Shi, L. Meng, A. Striegel, C Qiao, D. Koutsonikolas, and G. Challen. 'A walk on the client side: Monitoring enterprise Wifi networks using smartphone channel scans'. In: *IEEE International Conference on Computer Communications (IEEE INFOCOM)*. 2016, pp. 1–9. DOI: 10.1109/INFOCOM.2016.7524453 (cited on page 26).

[58] A. Farshad, M. K. Marina, and F. Garcia. 'Urban WiFi characterization via mobile crowdsensing'. In: *IEEE Network Operations and Management Symposium (NOMS)*. 2014, pp. 1–9. DOI: 10.1109/NOMS.2014.6838233 (cited on page 26).

[59] N. Hajlaoui, I. Jabri, and M. Ben Jemaa. 'An accurate two dimensional Markov chain model for IEEE 802.11n DCF'. In: *Wireless Networks* 24 (2018), pp. 1019–1031. DOI: 10.1007/s11276-016-1383-z (cited on page 27).

[60] B. S. Kim, H. Y. Hwang, and D. K. Sung. 'Effect of Frame Aggregation on the Throughput Performance of IEEE 802.11n'. In: *IEEE Wireless Communications and Networking Conference (WCNC)*. 2008, pp. 1740–1744. DOI: 10.1109/WCNC.2008.310 (cited on page 27).

[61] L. Song and A. Striegel. 'Leveraging frame aggregation to improve access point selection'. In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2017, pp. 325–330. DOI: 10.1109/INFCOMW.2017.8116397 (cited on pages 27, 28).

[62] IEEE. 'IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs'. In: *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)* (2008), pp. 1–244. DOI: 10.1109/IEEESTD.2008.4544755 (cited on page 32).

[63] L. Batyuk, A.-. Schmidt, H.-G. Schmidt, S. Camtepe, and S. Albayrak. 'Developing and Benchmarking Native Linux Applications on Android'. In: *Mobile Wireless Middleware, Operating Systems, and Applications*. Vol. 7. Apr. 2009, pp. 381–392. DOI: 10.1007/978-3-642-01802-2_28 (cited on page 35).

[64] W. Li, R. K. P. Mok, D. Wu, and R. K. C. Chang. 'On the accuracy of smartphone-based mobile network measurement'. In: *IEEE Conference on Computer Communications (INFOCOM)*. 2015, pp. 370–378. DOI: 10.1109/INFOCOM.2015.7218402 (cited on page 35).

[65] W. Li, D. Wu, R. K. C. Chang, and R. K. P. Mok. 'Toward Accurate Network Delay Measurement on Android Phones'. In: *IEEE Transactions on Mobile Computing* 17.3 (2018), pp. 717–732. DOI: 10.1109/TMC.2017.2737990 (cited on page 35).

[66] *The Network Simulator ns-2, https://www.isi.edu/nsnam/ns/*. 2010 (cited on page 37).

[67] *The Network Simulator ns-3, https://www.nsnam.org/*. 2021 (cited on page 37).

[68] A. Gupta, J. Min, and I. Rhee. 'WiFox: Scaling WiFi Performance for Large Audience Environments'. In: *International Conference on Emerging Networking Experiments and Technologies (CONEXT)*. CoNEXT '12. Nice, France: Association for Computing Machinery, 2012, pp. 217–228. DOI: 10.1145/2413176.2413202 (cited on pages 40, 69).

[69] S. M. Ross. *Introduction to Probability Models*. Tenth. San Diego, CA, USA: Academic Press, 2010 (cited on page 56).

[70] N. E. H. Bouzouita, A. Busson, and H. Rivano. 'Analytical study of frame aggregation level to infer IEEE 802.11 network load'. In: *International Wireless Communications and Mobile Computing (IWCMC)*. 2020, pp. 952–957. DOI: 10.1109/IWCMC48107.2020.9148448 (cited on page 56).

[71] A. Massouri, L. Cardoso, B. Guillon, F. Hutu, G. Villemaud, T. Risset, and J. Gorce. 'CorteXlab: An open FPGA-based facility for testing SDR cognitive radio networks in a reproducible environment'. In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2014, pp. 103–104. DOI: 10.1109/INFCOMW.2014.6849176 (cited on page 59).

[72] IEEE. 'Draft 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 Amendment Enhancements for High Efficiency WLAN'. In: *IEEE P802.11ax/D4.0, February 2019* (2019), pp. 1–746 (cited on page 61).

Special Terms

A
AC Access Category. 16
ACK Acknowledgement. 15, 16, 43
ADDBA ADD Block ACK. 12
AIWC Aggregation Intensity based WiFi Characterization. 25, 26, 89
ALAM AMPDU-based ap LoAd Mechanism. 27
A-MPDU Aggregate MAC Protocol Data Unit. 10–12, 26, 27, 42, 46, 59, 73
A-MSDU Aggregate MAC Service Data Unit. 10–12, 26, 27
AP Access Point. 1–3, 7–10, 17, 19, 27, 28, 31–33, 36, 38, 41, 43, 52, 57, 61, 68, 69, 81, 89–92
API Application Programming Interface. 33
ARQ Automatic Repeat Request. 24, 60
ART Android RunTime. 34, 35

B
BlockACK Block acknowledgments. 10, 12, 13, 15, 16, 43
BSA Basic Service Area. 8
BSS Basic Service Set. 7, 8, 10, 17, 31, 92
BSSID Basic Service Set IDentifier. 8
BTC Bulk Transfer Capacity. 21
BTF Busy Time Fraction. i, x, 4, 28, 32, 34, 37, 38, 40, 56, 71, 74, 75, 78–80, 84–92

C
CBR Constant Bit Rate. 42, 45
CCA Clear Channel Assessment. 14, 32
CCK complementary code keying. 8
CLT Central Limit Theorem. 72, 73, 78
CSMA Carrier Sense, Multiple Access. 14
CSMA/CA Carrier Sense, Multiple Access with Collision Avoidance. 14, 24
CSMA/CD Carrier Sense, Multiple Access with Collision Detection. 14

D
DCF Distributed Coordination Function. xii, 15–17, 37, 41, 56
DIFS Distributed Inter-Frame Space. 15, 24, 43
DL DownLink. i, 4, 9, 21, 39–42, 48, 52, 57, 58, 64, 66, 68, 69, 79, 85, 90
DS Distribution System. 7
DSSS Direct Sequence Spread Spectrum. 8
DTMC Discrete-Time Markov Chain. 27, 41
DVM Dalvik Virtual Machine. 35

E
EDCA Enhanced Distributed Channel Access. 13, 16
EIFS Extended Inter-Frame Space. 15, 16
ESS Extended Service Set. 1, 8, 92
EVA Estimated aVailable bAndwidth. 27

F
FAM Frame Aggregation based Method. i, x, xii, 4, 5, 29, 31, 34, 38, 40, 71, 72, 76, 78–92
FCS Frame Check Sequence. 11, 43, 56
FHSS frequency hopping spread spectrum. 8
FIFO First In First Out. 13, 25

G
GI Guard Interval. 61

H
HCCA Hybrid Controlled Channel Access. 16, 17
HCF Hybrid Coordination Function. 16
HE High Efficiency. 9, 37, 61, 80, 86
HT High Throughput. 9, 10, 57, 59, 63

I
IBSS Independant Basic Service Set. 8
ICMP Internet Control Message Protocol. 21
IFS Inter-Frame Space. 15
IP Internet Protocol. 10, 20–22
ISM Industrial, Scientific and Medical. 8

J
JNI Java Native Interface. 35

M
MAC MAC Medium Access Control Layer. 1, 4, 7, 9–11, 14, 19, 24, 25, 37, 43, 60, 87
MBSS Mesh Basic Service Set. 8
MCS Modulation and Coding Scheme. xii, 14, 32, 37, 41, 49, 57, 59, 61, 63–65, 80, 84–86
MIMO Multiple-Input Multiple-Output. 9
MPDUs MAC Protocol Data Units. 11
MU-MIMO Multiple User MIMO. 9, 10, 92

N
NAV Network Allocation Vector. 15, 25
NDK Native Development Kit. 35, 36

O
OFDM Orthogonal Frequency-Division Multiplexing. 8–10
OFDMA Orthogonal Frequency-Division Multiple Access. 9, 92
OSI Open Systems Interconnection. 7
OWD One Way Delay. 25

P
PCF Point Coordination Function. 15–17
PGM Probe Gap Model. 20, 23–25
PHY Physical Layer. 1, 4, 7–11, 19, 37, 43, 60
PIFS PCF Inter-Frame Space. 15
PRM Packet Rate Model. 20, 23, 25

Q
QAM Quadrature Amplitude Modulation. 9, 10, 14
QoE Quality of Experience. 19
QoS Quality of Service. 1, 2, 12, 16, 17, 19, 60

R
RF Radio Frequency. 26
RIFS Reduced Inter-Frame Space. 9, 15
RSSI Received Signal Strength Indicator. 17, 28, 33
RTS/CTS Request to Send, Clear to Send. 16
RTT Round Trip Time. 21
RU Resource Unit. 9

S
SDK Software Development Kit. 34
SDN software-defined networking. 2, 24
SIFS Short Inter-Frame Space. 9, 15, 16, 43
SR Spatial Reuse. 10
SSID Service Set IDentifier. 1, 33
STA STAtion. xi–xiii, 1, 2, 7, 16, 17, 28, 32, 33, 38, 41, 57–59, 61, 66, 67, 69, 78, 79, 84, 85, 87–89, 91, 92

T
TCP Transmission Control Protocol. 21, 22
TTL Time-To-Live. 20
TXOP Transmission Opportunity. 16

U
UDP User Datagram Protocol. 22, 36, 40
UL UpLink. i, 4, 9, 21, 40–43, 48, 52, 68, 69, 90
U-NII Unlicensed National Information Infrastructure. 8

V
VHT Very High Throughput. 9
VPS Variable Packet Size. 20

W
WEP Wired Equivalent Privacy. 35
WLAN Wireless Local Area Network. 1, 7, 9, 19, 20, 32, 38, 41, 52, 61, 88, 92
WNIC Wireless Network Interface Controller. 34, 35, 59
WPA Wi-Fi Protected Access. 35