Cross-layer techniques for Wireless Local Area Networks

Maria Eugenia Berezin

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

María Eugenia BEREZIN

Thèse dirigée par **Andrzej DUDA**
et codirigée par **Franck ROUSSEAU**

préparée au sein de l' **UMR 5217 - LIG - Laboratoire d'Informatique de Grenoble**
dans l'**École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (EDMSTII)**

Cross-layer techniques for Wireless Local Area Networks

Thèse soutenue publiquement le **19/12/2013**,
devant le jury composé de :

M. Vivien QUÉMA
Professeur, Grenoble INP – Ensimag, Président
M. Marcelo DIAS DE AMORIM
Directeur de Recherche CNRS, Université Pierre et Marie Curie, Rapporteur
M. Thomas NOËL
Professeur, Université de Strasbourg, Rapporteur
M. Alexandre GUITTON
Maître de Conférences, Université Blaise Pascal, Examinateur
M. Andrzej DUDA
Professeur, Grenoble INP – Ensimag, Directeur de thèse
M. Franck ROUSSEAU
Maître de Conférences, Grenoble INP – Ensimag, Co-Directeur de thèse

Abstract

In this dissertation, we examine important aspects of infrastructure IEEE 802.11 Wireless Local Area Networks (WLANs) and identify issues that can affect their performance. Reviewing the state of the art, we observe that numerous research efforts have proposed diverse solutions with several limitations that impede their deployment in existing WLANs. Moreover, users have ever-increasing expectations of availability, reliability, instantaneous response and security from their wireless connections.

Motivated by these challenges, we design and implement novel but practical solutions that address open issues affecting the performance of IEEE 802.11 WLANs. We adopt an Access Point (AP)-based approach, which does not require any modification in the clients. We focus on the following aspects of WLANs: client mobility, channel management, and quality of service, and explore three different scenarios for the most common deployments: an enterprise, a city (urban area), and a personal residence (home). To provide a common basis for practical implementation of new 802.11 solutions, we present a *Smart AP* model, inspired by self-management techniques.

The main contributions of this thesis are:

1. We develop a seamless mobility solution for Voice over IP (VoIP) services in Enterprise WLANs, called *Multichannel Virtual Access Points* (mVAP), which requires no client modifications and is compatible with current devices. We implement and evaluate mVAP using commodity 802.11 hardware, and achieve transparent mobility without interruption or degradation of ongoing communications.

2. We investigate the feasibility of harnessing the existing WiFi coverage in urban areas for mobile Internet access, through trace-based simulations using real data collected by mobile phones. The results show that the WiFi coverage is large and the connectivity it offers can be effectively exploited. We identify open issues for the actual deployment of such a citywide WiFi network and the applications that could benefit from it.

3. We propose an adaptive traffic-aware channel selection mechanism for Home WLANs, that uses the time-varying traffic load for interference estimation. We implement this solution using commodity 802.11 hardware and experimentally evaluate it: the network performance is drastically improved by constantly picking the channel with the least interference.

Résumé

Dans cette thèse, nous examinons les aspects essentiels des réseaux locaux sans fil IEEE 802.11 (réseaux WiFi) en mode infrastructure, et identifions les problèmes qui peuvent affecter leurs performances. Après avoir étudié l'état de l'art, nous constatons que de nombreux efforts de recherche ont proposé des solutions diverses mais présentant des limitations qui empêchent leur déploiement dans les réseaux locaux sans fil existants. En outre, les utilisateurs de ces réseaux ont des attentes toujours croissantes de disponibilité, de fiabilité, de réponse instantanée et de sécurité de la part de leurs connexions sans fil.

Motivés par ces défis, nous concevons et mettons en œuvre des solutions nouvelles et concrètes aux problèmes ouverts liés à la performance des réseaux locaux sans fil IEEE 802.11. Nous adoptons une approche centrée sur le point d'accès (Access Point), qui n'introduit pas de modifications côté client. Nous nous concentrons sur les aspects suivants des réseaux locaux sans fil : la mobilité des clients, la gestion des canaux, et la qualité de service, et nous explorons trois différents scénarios pour les déploiements les plus répandus : une entreprise, une ville (zone urbaine), et une résidence personnelle (maison ou appartement). Afin de fournir une base commune pour la mise en œuvre pratique de nouvelles solutions 802.11, nous introduisons un modèle de point d'accès intelligent, inspiré des techniques d'auto-gestion.

Les contributions principales de cette thèse sont les suivantes :

1. Nous développons une solution de mobilité transparente pour la Voix sur IP (VoIP) dans les réseaux sans fil d'entreprise, appelée Multichannel Virtual Access Point (mVAP), qui n'introduit aucune modification côté client et reste compatible avec les appareils actuels. Nous mettons en œuvre et évaluons mVAP en utilisant du matériel 802.11 standard, et accomplissons une mobilité transparente sans interruption ni dégradation des communications en cours.

2. Nous étudions la possibilité d'exploiter la couverture WiFi existante dans les zones urbaines pour obtenir un accès mobile à Internet, grâce à des simulations réalisées à partir de données réelles collectées par des téléphones portables. Les résultats montrent que cette couverture WiFi est étendue et que la connectivité offerte peut être efficacement utilisée. Nous identifions des questions ouvertes concernant le déploiement effectif d'un tel réseau WiFi à l'échelle d'une ville, et les applications qui pourraient en bénéficier.

3. Nous proposons un mécanisme dynamique de sélection de canal pour les réseaux locaux sans fil domestiques (maisons et appartements), qui utilise la charge de trafic variable dans le temps pour l'estimation d'interférences. Nous mettons en œuvre cette solution en utilisant du matériel 802.11 standard, et nous l'évaluons expérimentalement : les performances d'un tel réseau sont considérablement améliorées en choisissant le canal qui présente le moins d'interférences.

Remerciements

Je souhaite avant tout remercier Andrzej Duda de m'avoir donné l'opportunité de faire ma thèse dans un excellent laboratoire de recherche scientifique, dans une équipe disposant d'importantes ressources (très précieuses pour le travail expérimental), et aussi la possibilité de participer aux conférences et d'y présenter mes travaux, ce qui m'a permis d'enrichir encore davantage cette expérience.

Je remercie également Franck Rousseau de m'avoir fait découvrir le monde de la recherche, pour toutes nos discussions qui ont contribué à l'avancement de mon travail, et aussi pour son esprit toujours positif et ses qualités humaines et relationnelles.

Mes remerciements vont aussi aux membres du jury de ma thèse pour leurs commentaires et leurs questions, qui ont fait de la soutenance un très bon exercice de réflexion et d'échange d'idées.

Durant ces quelques années à Grenoble, j'ai eu la joie de rencontrer des personnes très sympathiques avec lesquelles j'ai passé de très bonnes soirées, des sorties à la montagne, des fêtes costumées et des discussions autour d'une bière, et avec lesquelles j'ai pu découvrir de nouvelles cultures et des points de vue différents sur la vie, le monde académique et la recherche scientifique. Je pense en particulier à (par ordre chronologique) Olivier, Fabrice, Noha, Yan, Benoît, Mohammad, Amal, Eryk, Sadaf, Vincent, Nazim, Asif, Martin, Christelle, Carina, Reinaldo, Maciej, Bogdan, Isabel, Ana, Martin K., Sofia, Didine, Étienne, Giorgio, Ghalem, Chi-Anh, Fabien et Mustafa. Merci beaucoup pour votre amitié !

Enfin, je voudrais remercier toute ma famille et mes amis (d'un côté de l'océan Atlantique comme de l'autre) qui m'ont toujours encouragée et ont affectueusement fait l'effort de comprendre ce que je faisais durant toutes ces années (merci à Sam pour les relectures et le pot !). Et merci à M d'être parti à l'aventure et d'avoir surfé avec moi ;). Le nouveau monde nous attend !

Cette thèse est dédiée à Pablito et Berni.

Maru.

19/12/2013

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

ACK	acknowledgment
AIFS	arbitration interframe space
AIFSN	arbitration interframe space number
AP	access point
ARP	Address Resolution Protocol
BSS	basic service set
BSSID	basic service set identifier
CAPWAP	Control and Provisioning of Wireless Access Points
CBR	constant bit rate
CCA	clear channel assessment
CCDF	complementary cumulative distribution function
CDF	cumulative distribution function
CFP	contention-free period
CP	contention period
CRC	cyclic redundancy code
CS	carrier sense
CSA	channel switch announcement
CSMA/CA	carrier sense multiple access with collision avoidance
CTS	clear to send
CW	contention window
DA	destination address
DCF	distributed coordination function
DFS	dynamic frequency selection
DIFS	distributed (coordination function) interframe space
DNS	Domain Name System
DS	distribution system
DSL	digital subscriber line
DSS	distribution system service
DSSS	direct sequence spread spectrum
DTLS	Datagram Transport Layer Security
EAP	Extensible Authentication Protocol (IETF RFC 3748-2004)

EDCA	enhanced distributed channel access
EIFS	extended interframe space
ESS	extended service set
ESSID	extended service set identifier
FCS	frame check sequence
FHSS	frequency-hopping spread spectrum
GPS	Global Positioning System
HC	hybrid coordinator
HCCA	HCF controlled channel access
HCF	hybrid coordination function
HR/DSSS	High Rate direct sequence spread spectrum using the long preamble and header
IAPP	Inter-Access Point Protocol
IBSS	independent basic service set
IEEE	Institute of Electrical and Electronics Engineers
IEEE 802.1X	IEEE Standard for Port-based Network Access Control
IF	interference factor
IFS	interframe space
IP	Internet Protocol
IR	infrared
ISP	Internet service provider
LAN	local area network
LCCS	Least Congested Channel Search
MAC	medium access control
MIMO	multiple input, multiple output
mVAP	multichannel virtual access points
NAT	Network Address Translation
NG	neighbor graph
NIC	network interface controller
NP	nondeterministic polynomial time
OFDM	orthogonal frequency division multiplexing
OSI	Open Systems Interconnection (ISO/IEC 7498-1:1994)
PACMAP	packet manipulation framework
PC	point coordinator
PCF	point coordination function
PHY	physical layer
PIFS	point (coordination function) interframe space
PLCP	physical layer convergence procedure
PMD	physical medium dependent

QoS	quality of service
QAP	QoS access point
QSTA	QoS station
RA	receiver address or receiving station address
RADIUS	remote authentication dial-in user service (IETF RFC 2865-2000)
RF	radio frequency
RSS	received signal strength
RSSI	receive signal strength indicator
RTP	Real-time Transport Protocol
RTS	request to send
SA	source address
SIFS	short interframe space
SNMP	Simple Network Management Protocol
SSID	service set identifier
STA	station
TA	transmitter address or transmitting station address
TC	traffic category
TCP	Transmission Control Protocol
TPC	transmit power control
UDP	User Datagram Protocol
USB	Universal Serial Bus
VAP	virtual access point
WEP	wired equivalent privacy
WLAN	wireless local area network
WMM	WiFi Multimedia
WNM	wireless network management
WPA	WiFi Protected Access
WPA2	WiFi Protected Access II

Chapter 1

INTRODUCTION

Contents

1.1 Motivation

With the widespread of mobile devices such as smartphones and tablets, wireless technology is rapidly becoming the method of choice for network access. One of these wireless technologies, IEEE 802.11, commonly known as WiFi, is widely adopted in indoor environments such as enterprises, organizations, and homes. It is also a fast and inexpensive alternative to cellular networks for Internet access. Nowadays, WiFi is present virtually everywhere, as the most diverse electronic devices, from mobile phones to home appliances, come with integrated WiFi chipsets. Users can connect effortlessly, because WiFi provides an easy setup and configuration.

Since its introduction, more than fifteen years ago, the uses and capabilities of IEEE 802.11 Wireless Local Area Networks (WLANs) have evolved significantly:

- Originally, WiFi was seen as a replacement for wired links. Portable computers connected wirelessly, but from fixed locations. Lately, mobile devices, principally handsets that consume both voice and Internet services, demand on-the-go connectivity.
- First data bit rates ranged from 1 to 2 Mb/s, but WiFi can now achieve high speeds of up to 1.3 Gb/s.

- Initially designed for best-effort traffic, WiFi later incorporated quality of service mechanisms to support multimedia applications with tight constraints.
- The original WiFi security algorithm suffered several flaws and was finally deprecated. Robust authentication and traffic encryption algorithms were subsequently introduced.

Indeed, users have ever-increasing expectations of availability, reliability, instantaneous response and security from their wireless connections. Rich-media applications such as audio and video streaming, seamless mobile connectivity, and cellular data offload are some of the emerging challenges that WiFi networks need to tackle, and more are to come.

Previous studies have proposed many approaches to improve different aspects of network performance. However, few of these research proposals contain practical solutions that can be deployed in existing WLANs. Furthermore, each type of deployment, from the hotspot in a café to a large university network, shows distinct characteristics and a single solution cannot fit all possible scenarios.

Motivated by these challenges, we focus on the important aspects of WLANs that are client mobility, channel management, and quality of service, and identify opportunities for optimization and innovation. Since the Access Point (AP) is the point of attachment to an 802.11 WLAN, the network performance of clients depends mainly on their connection with the AP. Additionally, clients are now so diverse that a global driver modification became an almost impossible task. Therefore, we decided to adopt an AP-based approach to develop new 802.11 solutions that are backward compatible and can be deployed in existing WLANs.

1.2 Thesis Statement

The aim of this thesis is to design and implement novel but practical solutions that address issues affecting the performance of *single-hop infrastructure* IEEE 802.11 WLANs. Infrastructure networks use central entities called Access Points (APs), which are in charge of maintaining connections with clients and managing traffic. In a single-hop infrastructure network, the APs do not route packets to or from other APs, but act essentially as bridges between a wireless and a wired network.

The proposed solutions follow an AP-based approach. They do not require any modification in the clients and do not introduce changes in the WiFi protocol, allowing interoperability with existing WiFi devices. These mechanisms are implemented in high-level software, not in the AP's firmware (embedded in the hardware), and evaluated on commodity 802.11 hardware.

To provide a common basis for practical implementation of new 802.11 solutions, we present a *Smart AP* model, inspired by self-management techniques. By means of adaptive algorithms, the AP will react to changes in the wireless medium and network, adjusting its configuration to improve the WLAN's performance. The AP will make decentralized decisions based on local measurements and cross-layer information.

We explore three different scenarios of WLAN deployments: an enterprise, a city (urban area), and a personal residence (home). These scenarios were selected in order to present a comprehensive work that explores the most common WLAN deployments, each one with distinct and complementary characteristics to exploit.

- **Enterprise WLANs:** These managed networks are composed of numerous APs, which all belong to a same administrative domain. APs are carefully deployed and configured, to avoid interference and maximize capacity and coverage. They are interconnected through a wired network, and a central controller usually manages the wireless network information. Client devices such as laptops, tablets and mobile phones are generally homogeneous. Other large-scale networks of this type include university campuses, hospitals, and other public institutions.

- **Citywide wireless network:** Not a proper network yet, but a collection of APs distributed throughout a city (residential gateways, commercial hotspots, and municipal networks, for example). The dense deployment of these APs create large WiFi coverage zones in urban areas. The APs usually offer Internet access, but not all of them provide free and unrestricted connections. They do not communicate with each other and have different configurations, such as channel and security parameters. Clients are highly mobile, mostly data-hungry handsets with limited Internet subscriptions provided by cellular services.

- **Home WLANs:** These residential networks are characterized by a single AP and a broadband Internet connection. Owners have little or no administration skills, so APs are set up with default configurations. These networks reveal a dense and unplanned deployment, with overlapping coverage areas among neighboring WLANs, commonly causing interference. Clients are diverse and consume different types of traffic, such as web, file sharing, audio and video streaming, and gaming.

We now proceed to enumerate the main contributions of this dissertation:

1. We develop a seamless mobility solution for Voice over IP (VoIP) services in Enterprise WLANs. Roaming in WiFi networks interrupts established communications and higher-layer sessions, which can be fatal to delay-sensitive applications such as VoIP. We design an AP-based mobility management technique called *Multichannel Virtual Access Points* (mVAP), which requires no client modifications and is compatible with current devices. We implement and evaluate mVAP using commodity 802.11 hardware, and achieve transparent mobility without interruption or degradation of ongoing communications.

2. We investigate the feasibility of exploiting the WiFi coverage in urban areas for mobile Internet access. We analyze the data collected by mobile phones to identify the distribution and properties of APs already deployed in a city. Through trace-based

simulations we evaluate the WiFi coverage and the characteristics of the connectivity of mobile users, using different mobility speeds and AP settings. We discuss the open issues for a real deployment of a citywide WiFi network and the applications that could benefit from it.

3. We propose a traffic-aware channel selection mechanism for Home WLANs. The extensive deployment of WiFi technology in urban areas has caused an overcrowding of the license-free frequency bands, leading to high interference among neighboring WLANs. We design a channel assignment technique that uses the time-varying traffic load for interference estimation. We implement this solution using commodity 802.11 hardware and experimentally evaluate it: the network performance is drastically improved by constantly picking the channel with the least interference.

1.3 Dissertation Outline

This thesis is structured as follows: in Chapter 2, we introduce the IEEE 802.11 standard and briefly describe the basic terms and concepts that are relevant for this dissertation. In Chapter 3, we focus on important aspects of WLAN deployments: client mobility, channel selection, and quality of service. We study these aspects in detail, identify the issues that can affect the WLAN's performance, and review previous research works. Chapter 4 describes our *Smart AP* model, which will serve as a guide for the contributions of this thesis. In Chapter 5, we present our seamless mobility solution for VoIP services in Enterprise WLANs. In Chapter 6, we investigate the characteristics of WiFi coverage in urban areas for a citywide mobile Internet access. In Chapter 7, we develop an adaptive traffic-aware channel selection mechanism for Home WLANs. Finally, we conclude this dissertation and outline future research directions in Chapter 8.

Chapter 2

IEEE 802.11 OVERVIEW

Contents

2.1 Introduction

The IEEE 802.11 standard [35] defines the Physical (PHY) layer and Medium Access Control (MAC) sublayer of the OSI model (as depicted in Figure 2.1) for Wireless Local Area Networks (WLANs). It was created in 1997, but had many subsequent amendments, adding new functionalities, for instance new modulation methods, and reinforcing aspects such as security and quality of service.

In this chapter, we present a general description of the IEEE 802.11 standard and introduce the concepts and terminology used within this dissertation.

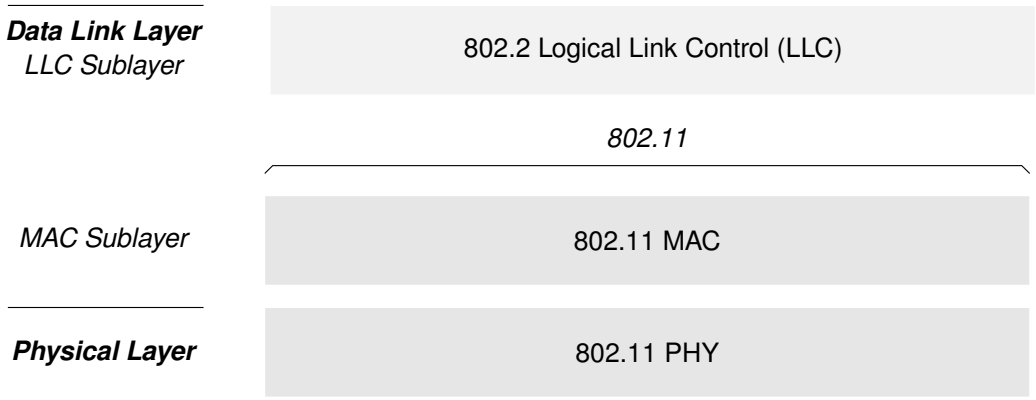


Figure 2.1: IEEE 802.11: Physical and MAC layers.

2.2 Architecture

A WLAN is composed of two or more devices that communicate through radio waves. The devices can move around a coverage area while still being connected to the network.

2.2.1 Components

A WLAN can consist of the following components:

- **Station (STA):** a device with an 802.11 wireless network interface, that acts as a client of the network. Stations are generally battery-powered and easily transportable; for example, laptops, smartphones, tablets, and game consoles.
- **Access Point (AP):** a special wireless node with 802.11 capabilities, that acts as a central transmitter and receiver of the stations' signals. APs often operate as bridges between wireless and wired networks.
- **Distribution System (DS):** a logical component that interconnects APs of a same WLAN. A DS is commonly an Ethernet wired network.

2.2.2 Types of Networks

The Basic Service Set (BSS) is the basic building block of an 802.11 WLAN. It is a group of stations (and optionally one AP) that communicate with each other. A *basic service area* defines the coverage area where these stations are fully connected.

BSSs can be classified as:

- **Independent network (IBSS):** consists of two or more stations which can only communicate directly with the other stations in range. These networks are generally deployed for a short period of time and without previous planning, and are therefore also referred to as *ad hoc networks*.

- **Infrastructure network:** uses a central point (the AP) for all communications. When a station wants to communicate with another station, the first station sends a frame to the AP, which in turn forwards this frame to the second station. In order to be members of this network, the stations need to be in the coverage area of the AP. This type of BSS is the subject of our study.

A BSS is identified by its Basic Service Set Identification (BSSID): in an Infrastructure BSS, it is the 48-bit MAC address of the wireless network interface of the AP (in an IBSS, the BSSID is a random MAC address). This value is unique for each BSS.

An Extended Service Set (ESS) is a set of multiple BSSs, connected by a backbone network (the DS). The BSSs may overlap, in order to create a large continuous coverage area, as shown in Figure 2.2. Thus, stations can move freely around the ESS while maintaining connectivity. The DS provides seamless integration of the BSSs, logically interconnecting them at the MAC layer. Consequently, all stations within a same ESS are able to communicate with each other. All BSSs in an ESS have the same Extended Service Set Identifier (ESSID), a human-readable network name for the users. Examples of such ESS networks are: university campuses, convention centers, airports, and enterprises.

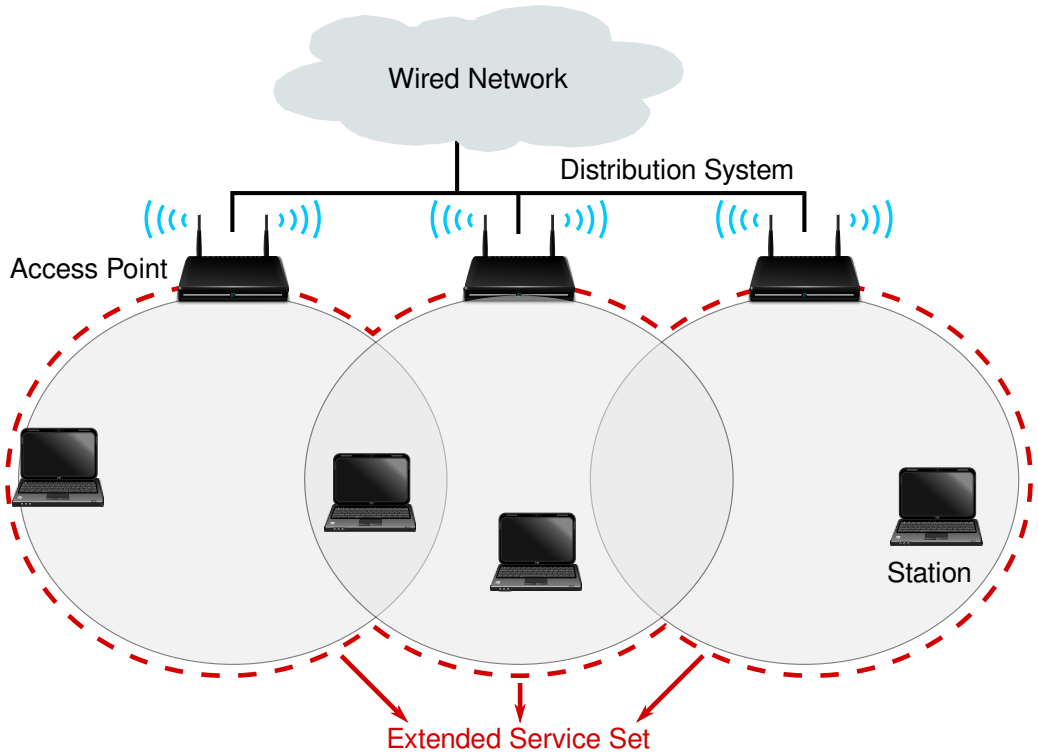


Figure 2.2: Extended Service Set architecture.

2.3 Physical Layer

The 802.11 PHY layer is the interface between the MAC sublayer and the wireless medium. It is responsible for frame transmission and reception tasks. The PHY layer prepares the MAC frames for transmission, transforms them into signals, and controls the wireless medium for transmission and reception opportunities. IEEE 802.11 is designed to support wireless communication in license-free spectrum.

Originally, in 1997, three PHY layers were standardized in the initial publication of 802.11: FHSS (Frequency hopping spread spectrum), DSSS (Direct-sequence spread spectrum), and IR (Infrared). Only DSSS is still present in today's devices: FHSS is not used anymore for WLANs, and IR never had a commercial product implementation.

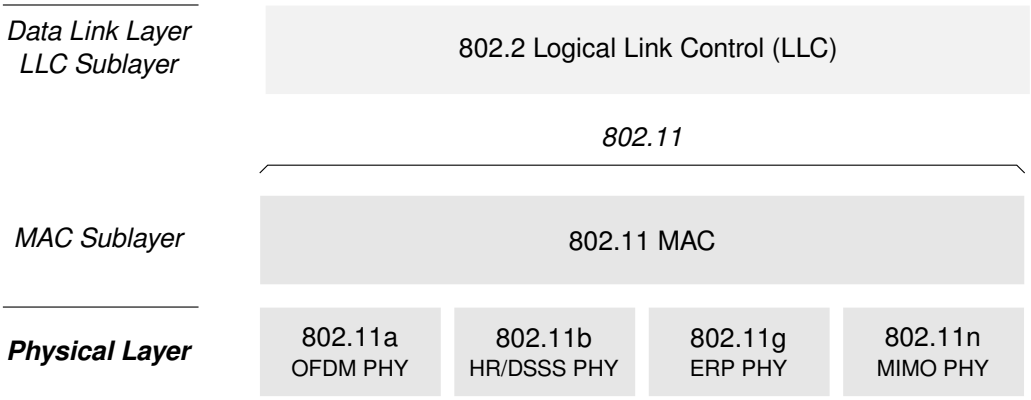


Figure 2.3: IEEE 802.11: Physical layers.

As shown in Figure 2.3, the following physical layers are currently supported by 802.11:

- **802.11a**, introduced in 1999, uses the OFDM (orthogonal frequency division multiplexing) technology. It operates in the 5 GHz band and provides data rates from 6 Mb/s to 54 Mb/s.

- **802.11b**, also ratified in 1999, uses the HR/DSSS (High-rate direct-sequence spread spectrum) modulation technique, which extends DSSS. It operates in the 2.4 GHz band and offers data rates of 1, 2, 5.5, and 11 Mb/s. The 2.4 GHz band has the disadvantage of being more crowded (for instance, by microwave ovens and cordless phones) than the 5 GHz band. However, 802.11b signals can reach farther distances, as they are less readily absorbed by walls and other obstacles.

- **802.11g**, introduced in 2003, utilizes the same OFDM technology as 802.11a, providing high data rates (up to 54 Mb/s), but operates in the 2.4 GHz band. 802.11g is fully backward compatible with 802.11b, allowing the co-existence of devices that use these technologies in a same WLAN. Both 802.11b and 802.11g are the most common technologies found nowadays in wireless devices.

- Lastly, **802.11n** implements the MIMO (multiple-input and multiple-output) technology, significantly increasing data throughput. It also operates in the 2.4 GHz band. The use of multiple antennas and wider channel bandwidth (40 MHz instead of 20 MHz, as in 802.11g) provides high data rates of up to 600 Mb/s. 802.11n was ratified in the year 2009.

  New physical layers are constantly emerging, such as 802.11ac which operates in the 5 GHz band and outperforms 802.11n by using a much wider channel bandwidth (up to 160 MHz) and more MIMO spatial streams.

Internally, the 802.11 PHY layer consists of two sublayers:

- the **Physical Layer Convergence Procedure** (PLCP) sublayer receives the MAC frames and prepares them for radio transmission. It appends a PHY-specific preamble and the PLCP header to the frame, in order to synchronize the transmitter with the receiver. Each modulation technique implements a different PLCP.
- the **Physical Medium Dependent** (PMD) sublayer is responsible for transmitting the frame into the air, using the antenna. It transforms the bits of information received from the PLCP into RF signals using the modulation technique of choice.

The PHY layer also supervises the state of the wireless medium, by means of a Carrier Sense/Clear Channel Assessment (CS/CCA). This procedure detects the beginning of a network signal that can be received (CS) and determines whether the medium is idle, before transmitting a frame (CCA).

2.4 MAC Sublayer

The 802.11 MAC sublayer controls the delivery of frames into the air and provides the principal framing operations. It has the advantage of being interoperable with every PHY layer presented above, and uses the 802.2 Logical Link Control encapsulation to interact with an Ethernet wired network.

Access to the shared wireless medium is regulated by a coordination function. 802.11 defines three coordination functions, as shown in Figure 2.4, each providing different services:

- **Distributed Coordination Function (DCF):** This is the fundamental MAC protocol of IEEE 802.11 and can be found in every WiFi product. DCF defines the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) contention access mechanism, in which several nodes can independently contend for the medium in order to transmit a frame, without the presence of a central controller.

  In DCF, a wireless node first checks if the medium is idle before transmitting, using a carrier sensing technique. Wireless NICs are half-duplex: they can only transmit or receive at a time. Collisions cannot be detected while transmitting, thus, in order to avoid them, DCF applies an exponential random backoff.

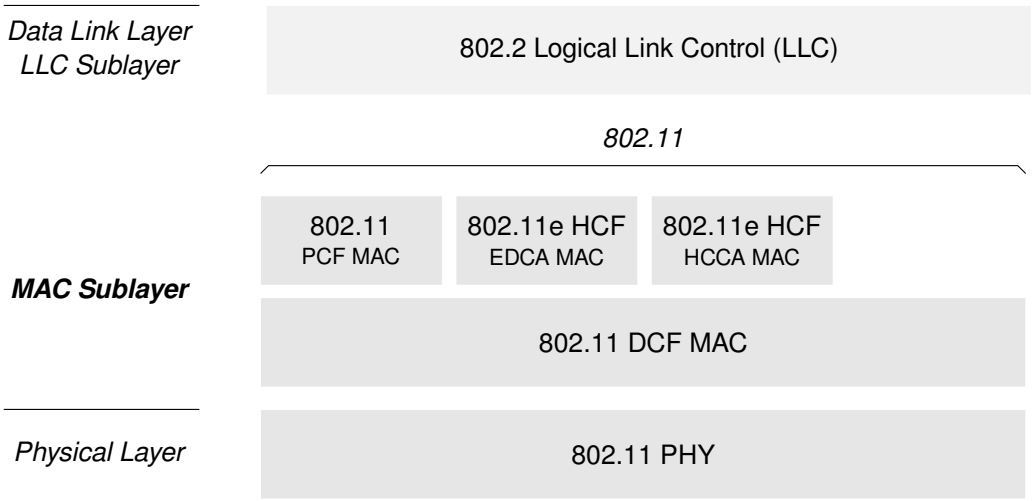


Figure 2.4: IEEE 802.11: MAC protocols.

- **Point Coordination Function (PCF):** It provides contention-free services, by means of a Point Coordinator (PC) that restricts access to the medium. Associated stations can transmit frames only when the PC allows them. The PC resides in the AP, so PCF is limited to Infrastructure networks.

  PCF interoperates with stations implementing only DCF, alternating contention-free periods with standard DCF-based service. PCF is optional in the IEEE 802.11 specification, and very few hardware devices implement it.

- **Hybrid Coordination Function (HCF):** Part of the 802.11e amendment, HCF combines and enhances both contention-based and contention-free access methods, enabling prioritized and parameterized Quality of Service (QoS) access to the wireless medium. It is compatible with both DCF and PCF.

  HCF proposes two different methods of channel access: Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA). EDCA is similar to DCF, since it uses the CSMA/CA access mechanism. HCCA works like PCF and its contention-free periods. EDCA is implemented in existing hardware, but very few wireless products implement the optional HCCA.

2.5 Frame Transmission

We now detail the frame transmission in 802.11 using the DCF technique, as we will focus on this MAC protocol in our dissertation. When a station has a frame in its transmission queue, it first checks whether the medium is idle:

1. On the one hand, if the medium is idle for longer than a DCF Interframe Space (DIFS) interval, the transmission begins immediately.

2. On the other hand, if the medium is busy, the station defers its access and waits for the channel to be idle without interruption for a period of time equal to DIFS. Then, the station prepares for an exponential backoff procedure, in order to minimize the probability of collision with other stations that are also waiting for transmission.

   After the elapsed DIFS, the aforementioned backoff procedure begins and lasts for a period of time called *contention window*. This period is divided into slots of equal time. The station picks a random slot and waits for it, decrementing its backoff time as long as the medium is idle. When the contention timer expires, the station starts the transmission if the medium is idle. If not, the station restarts the deferral process.

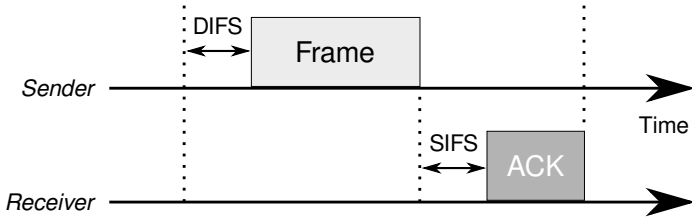


Figure 2.5: Frame acknowledgement.

Because the wireless medium is unreliable, 802.11 includes a positive acknowledgement of the received frame. Only unicast frames are acknowledged. As an example, in Figure 2.5, a first station sends a frame to another station:

1. After receiving this frame, the second station waits for a Short Interframe Space (SIFS) period and sends back an Acknowledgement frame. The first station receives it, and the transmission of the original frame ends successfully.

2. However, if after sending the original frame, the first station does not receive an Acknowledgement frame, it considers the original frame to be lost, and starts its retransmission.

   If after several retransmissions, the first station still does not receive an Acknowledgement frame, it finally drops the original frame.

As described above, DCF (as well as other MAC protocols) uses *Interframe Spaces* to coordinate the access to the transmission medium. Interframe spaces are time intervals between frames, and their duration depends on the PHY layer type. Apart from DIFS and SIFS, other interframe spaces exist, such as EIFS (Extended Interframe Space), used after a transmission error, and PIFS (PCF Interframe Space), used by PCF.

2.6 Frame Format

The 802.11 MAC frame format consists of a set of fields that are present in a fixed order in all frames. The format is depicted in Figure 2.6. The first three fields of the MAC header (Frame Control, Duration/ID, and Address 1) and the FCS are present in all frames, constituting the minimal frame format. The remaining fields (Address 2, Address 3, Sequence Control, Address 4, QoS Control, and Frame Body) are only present in certain frame types. Unlike Ethernet frames, 802.11 MAC frames do not include a preamble. As previously described in Section 2.3, the preamble is part of the PHY layer.

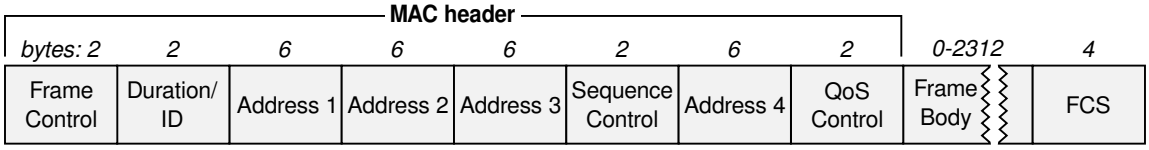


Figure 2.6: General 802.11 MAC frame format.

A frame includes three basic components:

- **MAC header:** contains the frame control, duration, address, and sequence control information, and, in QoS data frames, the QoS control information.
- **Frame Body:** is a variable length field, which contains information specific to the frame type. Its length can vary from 0 to 2,312 bytes.
- **Frame Check Sequence (FCS):** contains a 4-byte CRC (cyclic redundancy check). It is used to check the integrity of the frames. A sender calculates the FCS of the frame, over all the MAC header fields and the Frame Body field, before transmission. The receiver recalculates the FCS and compares it to the one specified by the sender. If they match, the frame was received correctly and the receiver sends back an Acknowledgement frame.

The MAC header is composed of the following fields:

- **Frame Control:** is 2 bytes in length and consists of the following subfields, as illustrated in Figure 2.7:
  - **Protocol Version:** is 2 bits in length and, for this standard, has a value of 0. Different values represent different revisions of the standard, and are not compatible with each other.
  - **Type:** is 2 bits in length and represents the type of the frame: control, data, or management. We detail these frame types in Section 2.6.1.
  - **Subtype:** is 4 bits in length and, together with the Type subfield, unequivocally represents the function of the frame. Each frame type has several subtypes.

Figure 2.7: Frame Control field.

- **To DS, From DS:** are both 1 bit in length, and combined together indicate whether the frame is destined for the DS.
  - *To DS=0, From DS=0:* management and control frames, as well as data frames from an IBSS.
  - *To DS=1, From DS=0:* data frames destined to the DS, in an Infrastructure network.
  - *To DS=0, From DS=1:* data frames destined to a station, in an Infrastructure network.
  - *To DS=1, From DS=1:* data frames using the fourth address field.
- **More Fragments:** is 1 bit in length. Frames can be sent in several fragments; this value is set to 1 if more fragments are to follow.
- **Retry:** is 1 bit in length and is set to 1 if the received data or management frame is a retransmission of an earlier frame.
- **Power Management:** is 1 bit in length and indicates the power management mode of a station. This value is always set to 0 in the AP's frames. A value of 1 indicates that the station will be in powersave mode, and a value of 0 indicates that the station will be active.
- **More Data:** is 1 bit in length. This value is set to 1 if the AP has at least one frame available for a station that is in powersave mode.
- **Protected Frame:** is 1 bit in length. This value is set to 1 if the frame is protected by a link-layer security protocol, such as WEP.
- **Order:** is 1 bit in length. A value of 1 indicates that frames are transmitted in a strict order. This value is set to 0 by default.

- **Duration/ID:** is 2 bytes in length. This field has several uses: when bit 15 is 0, this field is the duration (in microseconds) that the medium is expected to be busy for the current transmission; when bit 15 is 1, this field is used in PCF operation, and represents the Contention-Free Periods or the Association ID.
- **Address fields:** are 6 bytes in length and contain a 48-bit MAC address each. A MAC address has one of the following types: an individual (unicast) address, if the least significant bit of the first byte of the address is set to 0; a multicast address, if the least significant bit of the first byte of the address is set to 1; or a broadcast address, if all bits are set to 1.

As mentioned before, a MAC frame can contain from one to four address fields, depending on the frame's type. An address field can take one of the following values:

  - the **Basic Service Set Identification** (BSSID), the AP's MAC address (in an Infrastructure network) which uniquely identifies the BSS;
  - the **Source Address** (SA), the unicast source of the transmission;
  - the **Destination Address** (DA), the final recipient of the frame;
  - the **Transmitting STA Address** (TA), the station that transmitted the frame onto the wireless medium (only used in wireless bridging);
  - the **Receiving STA Address** (RA), the intended immediate recipient station on the wireless medium (e.g., the receiver of an Acknowledgement frame).

- **Sequence Control:** is 2 bytes in length. It contains two subfields: the Sequence Number, which is assigned to every frame in order to eliminate duplicates and identify message order; and the Fragment Number, which is set to 0 in the first or only fragment of a frame, and is incremented by 1 in each successive fragment. This field is not present in control frames.

- **QoS Control:** is 2 bytes in length and is only present in QoS operation. It identifies the traffic category of the frame and other QoS-related information.

2.6.1 Frame Types

The 802.11 MAC sublayer uses three types of frames:

- **Management frames** support the 802.11 services; they mainly enable stations to establish and terminate their associations with a BSS. We describe their use in Section 2.7.

- **Data frames** carry higher-layer packets in their Frame Body field. Different subtypes of data frames exist, depending on whether contention-based, contention-free, QoS or non-QoS services are used.

- **Control frames** support the delivery of management and data frames. The following are common control frame subtypes:

  - Acknowledgement (ACK): This frame is used to send a positive acknowledgement of a frame transmission, as described in Section 2.5.
  - Ready To Send (RTS) / Clear To Send (CTS): These two frames announce the delivery of a data frame and reserve the channel for the transmission.

2.7 Management Operations

We now detail some of the management operations supported by the 802.11 specification. In particular, we focus on the establishment and termination of a station's link-layer connection to a BSS. Figure 2.8 illustrates the three different states of a station in an Infrastructure network:

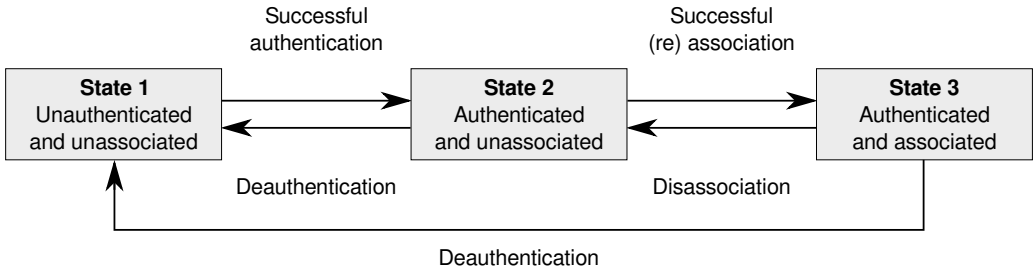


Figure 2.8: Overall states of an 802.11 station.

1. Initially, the station is not member of any BSS: it is unauthenticated and unassociated. The station discovers the available BSSs through a *scanning* operation. Then, the station chooses to *authenticate* with a certain BSS.

2. If the authentication is successful, the station is now authenticated and unassociated. Then, the station sends an *association* message to the BSS's AP.

3. If the association is successful, the station joins the BSS and is now authenticated and associated. Once in this state, the station starts processing the BSS's *beaconing* and is allowed to exchange data frames with its current AP.

The connection between the station and the BSS can always be modified or terminated, through a *deauthentication* or *disassociation* message originated by the station or its current AP. The station then returns to a previous state.

All 802.11 management frames share the structure depicted in Figure 2.9: the frame format is independent of the frame subtype. Only the first three Address fields are used: the first is the destination address DA, the second is the source address SA, and the third is the BSSID address.

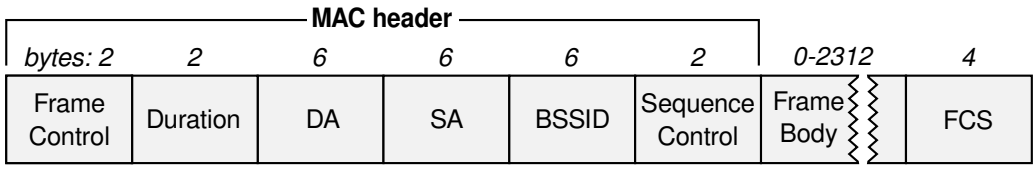


Figure 2.9: Generic management frame.

The Frame Body field contains information encoded with two different types of elements: fixed-length fields, simply referred to as *fields*, and variable-length fields, called *information*

elements. Information elements have a generic format consisting of a 1-byte Element ID field, a 1-byte length field, and a variable-length element-specific Information field, of *length* bytes. Each element has a unique ID, so new elements can be easily defined in newer versions of 802.11.

2.7.1 Beaconing

Beacon frames announce the presence of a BSS. In an Infrastructure network, the AP is responsible for periodically transmitting these frames. Information about the BSS is included in the beacon frames, such as the current operating channel (frequency), the beacon interval, the network's capabilities, the SSID, and the supported bit rates.

The area in which beacon frames can be correctly received determines the basic service area. Stations use the AP's beacons to confirm they are close enough to the AP to maintain communication. Stations also use beacons to identify networks and to synchronize their clocks.

2.7.2 Scanning

Before being able to join a BSS, stations must discover the existing networks in the area and identify the compatible ones. This task is accomplished through a scanning procedure, which builds a list of the available BSSs within range and their description. These results assist the process of AP selection.

The standard defines two scanning approaches: Passive Scanning and Active Scanning. In **Passive Scanning**, the station listens to each channel during a period of time and waits for *Beacon* frames to extract information about the existing BSSs. Passive scanning saves battery power because it does not require any transmission.

Active Scanning involves the broadcasting of *Probe Request* frames and the processing of *Probe Response* frames. Stations in Active Scanning mode use the following procedure, for each channel:

1. Move to the channel and wait for an incoming frame during a `ProbeDelay` interval. If this timer expires, and no incoming frame was detected, the channel is considered empty: just move to the next channel. Otherwise, an incoming frame is detected before the timer expires: the channel is in use.

2. Send a *Probe Request* frame to the broadcast destination address.

3. Wait for a `MinChannelTime` interval:

   - If the medium was idle during that time, there are no available networks in this channel. Move to the next channel.
   - If the medium was busy, wait for a `MaxChannelTime` interval and process any received *Probe Response* frames.

`ProbeDelay`, `MinChannelTime`, and `MaxChannelTime` are scanning timers and their values can vary, depending on the driver.

A *Probe Request* frame contains two fields in its Frame Body: the SSID and the bit rates supported by the station. If the station scans for a specific network, the SSID field takes the value of that network's SSID. On the other hand, if the station is willing to join any network, the SSID field uses an empty SSID and the BSSID Address field uses the broadcast address.

APs are responsible for generating *Probe Response* frames whenever they hear a *Probe Request* frame that is searching for their own SSID or using a broadcast SSID. If the station supports the bit rates required by the network, *Probe Response* frames are transmitted individually to the station. The fields in a *Probe Response* frame are similar to the ones present in *Beacon* frames.

2.7.3 Authentication

This is the first step to join a network: a station must authenticate to an AP before being able to proceed with the association operation. The station sends an *Authentication* frame to the AP, indicating the desired type of authentication. The AP sends back another *Authentication* frame to the station, with a specific response to its authentication request. As this authentication process may require several frame exchanges, the *Authentication* frames include a sequence counter field in their Frame Body. If the authentication process is successful, the station is then authenticated but still unassociated.

There are two types of authentication:

- **Open System authentication**: This is a null authentication algorithm. The stations do not provide any credential, which means that anyone can authenticate. This mechanism uses only two messages: the first message (station to AP) requests authentication, and the second message (AP to station) returns the authentication result. If the result is "Successful", the station is authenticated to the AP.

- **Shared Key authentication**: Stations need a shared secret key in order to authenticate to the AP. This mechanism uses the Wired Equivalent Privacy (WEP) security algorithm and consists of a four-step challenge-response handshake. WEP is also used for encrypting data frames; however, this algorithm is now deprecated.

2.7.4 Association

Once the station is authenticated to an AP, it is able to join the BSS by sending an *Association Request* frame to the AP. When the AP receives this frame, it first verifies that the parameters included in the frame match those of the network, and then transmits an *Association Response* frame to the station, with the result of the operation. If an *Association Response* frame is received with a status value of "Successful", the station is

authenticated and associated with the AP. But if the frame has another status value, the station is not associated with the AP, and the status value indicates the reason for the failed association attempt: the station is then authenticated but unassociated.

At any given time, the station is associated with no more than one AP. When the association process is complete, the station is able to fully use the DS (via the AP) for communication. The association is always initiated by the station.

2.7.5 Reassociation

Mobile stations may need to reassociate with the network, if they are moving around the service area of an ESS and leave the coverage of their current AP. The station requests a change in its association, by transmitting a *Reassociation Request* frame to a new AP from the same ESS. At this point, the station must already be authenticated to this new AP: this operation can happen immediately before the reassociation or even earlier, before the reassociation is actually needed.

When the AP receives the *Reassociation Request* frame, it transmits a *Reassociation Response* frame to the station. This mechanism is similar to the association frame exchange. If the status value of the *Reassociation Response* frame is "Successful", the station is now associated with the new AP: it is authenticated and associated. However, if the status value is not "Successful", the station is authenticated but unassociated. The status value in the frame indicates the reason for the failure to reassociate.

The *Reassociation Request* and *Reassociation Response* frames have the same format as the *Association Request* and *Association Response* frames, respectively. The *Reassociation Request* also includes the address of the mobile station's current AP. This information makes it possible to transfer the station's association context from the current AP to the new AP. However, this mechanism is undefined, as it is not part of the standard.

2.7.6 Deauthentication and Disassociation

Deauthentication and *Disassociation* frames are used to terminate an authentication or an association relationship, respectively. Both frames can be originated either by the station or the AP. They include a single fixed-length field in their Frame Body, the Reason Code, which specifies the reason why the deauthentication or disassociation procedure was initiated.

Upon transmission or reception of a *Disassociation* frame, the station is authenticated but unassociated. Upon transmission or reception of a *Deauthentication* frame, the station is unauthenticated and unassociated, returning to the initial "State 1" in Figure 2.8.

2.8 Security

WLANs are inherently more difficult to secure than wired LANs: due to the broadcast nature of the wireless medium, any transmission can be overheard by any device within range. The IEEE 802.11 standard proposes two types of security mechanisms: WEP and 802.11i.

WEP (Wired Equivalent Privacy) was the first security algorithm included in the IEEE 802.11 standard. It provides confidentiality by encrypting the Frame Body field of data frames (upper-layer information) with a shared key. It also protects the WLAN from unauthorized access, as detailed in Section 2.7.3. WEP has been eventually deprecated because of numerous flaws and failure to provide security services.

The 802.11i amendment was introduced later as a replacement for WEP. It is also known as WPA2 (WiFi Protected Access II) and specifies new authentication and encryption protocols. A station using the WPA2 authentication first uses the Open System authentication (described in Section 2.7.3) with the AP. WPA2 provides two security modes: Personal and Enterprise. The WPA2 Personal mode is designed for home and small office networks and uses pre-shared keys. The WPA2 Enterprise mode is designed for larger enterprise networks. Secured authentication is provided by IEEE 802.1X services.

2.9 Summary

We presented a description of the basic characteristics of the IEEE 802.11 standard, which defines the MAC sublayer and the PHY layer for wireless connectivity in WLANs. The terms and concepts introduced in this chapter are those used within our dissertation.

We particularly observed that not all of the functionalities included in the 802.11 specification are implemented in hardware (for example, PCF and HCCA), and that the values of some parameters depend on the vendors' implementations (for example, the scanning timers `ProbeDelay`, `MinChannelTime`, and `MaxChannelTime`), which leaves room for innovation.

Chapter 3

PERFORMANCE LIMITATIONS IN WLANS

Contents

3.1 Introduction

When deploying a WLAN, there are several factors to take into consideration in order to establish the coverage area, choose a proper configuration, and determine the services to offer. In this dissertation, we will particularly focus on the following three aspects:

- *client mobility*, to support transition from one BSS to another;
- *channel assignment*, to increase network capacity and avoid interference from other APs, other WLANs, or external sources;
- and *quality of service*, to guarantee minimum traffic requirements.

However, these aspects present limitations that can affect the performance of the WLAN. For example, when moving between coverage areas, clients may suffer interruptions that

negatively impact upper-layer communication protocols. 802.11 does not specify how to approach this problem; only vendor-proprietary methods provide seamless and fast roaming. Furthermore, different uses for WLANs emerge, such as streaming applications, introducing a new set of challenges.

The three aforementioned aspects of WLAN deployment create the opportunity for optimization and innovation. In this chapter, we first describe them in detail, then examine their performance limitations, and finally review the related research and discuss open issues.

3.2 Mobility

Mobility is one of the most important advantages of wireless networks. Stations can move freely within the coverage area of an AP, without disrupting their network connections. Moreover, the proliferation of small handheld devices, such as smartphones and tablets, has allowed connectivity on-the-go, so stations can transmit and receive frames while roaming.

The mobility of a station within 802.11 WLANs can be classified into the following three transition types:

- **No-transition:** the station is static or moves within the basic service area of its current AP.
- **BSS-transition:** the station moves from one BSS to another, both part of the same ESS. IEEE 802.11 enables MAC layer mobility through the DS. Cooperation between APs is required and can be achieved by means of an Inter-Access Point Protocol (IAPP). During a BSS transition, the old and new APs exchange information about the mobile station's association state (e.g. security context). If both APs belong to the same IP subnet, the station can keep its IP address and, consequently, station mobility is transparent to upper layers.
- **ESS-transition:** the station moves from a BSS in one ESS, to a BSS in another, different ESS. The 802.11 standard cannot guarantee maintenance of upper-layer connections, so the station is likely to suffer from service disruption. Network-layer mobility needs to be supported by special protocols, such as Mobile IP [82] or Proxy Mobile IPv6 [39].

3.2.1 Handoff Procedure

When a station leaves the basic service area of its current AP, it starts a process called *handoff*, which occurs at the Link layer. The handoff is entirely a client-driven process. The station is responsible for detecting that the connection with its current AP degrades, and for establishing an association with a new AP. Deploying APs with overlapping coverage areas enables continuous connectivity for mobile stations.

As depicted in Figure 3.1, there are three distinct phases in the handoff procedure:

- **Discovery:** The station searches for potential APs to associate with. It initiates an Active Scanning (described in Section 2.7.2), by sending *Probe Request* frames in each available channel, and processing the received *Probe Response* frames from the APs. The station may also obtain the APs' information through Passive Scanning. The station completes the scanning and chooses the new AP.
- **Authentication**: The station requests an authentication to the new AP. The authentication can also be performed in advance, as a *preauthentication*.
- **Reassociation**: After a successful authentication, the station proceeds to associate with the new AP. In a BSS transition, the reassociation service transfers, through the DS, the current association context from the old AP to the new one.

  In an ESS transition, the station exchanges *Association Request* and *Association Response* frames with the new AP.

Handoff Delay

The total handoff delay can be obtained as the sum of the Discovery, Authentication and Reassociation phases delays.

- When the station executes an Active Scanning, it spends, in each channel, either a *MinChannelTime* interval, if the station does not receive any *Probe Response* frame, or a *MaxChannelTime* interval otherwise, as described in Section 2.7.2. Therefore, the Discovery phase delay $T_{Discovery}$ is the sum of the time spent scanning each channel. If $P(c)$ is the probability of finding at least one AP in channel c, $T_{Discovery}$ is calculated as follows:

$$T_{Discovery} = \sum_{c=1}^{n} \Big(MinChannelTime * (1 - P(c)) + MaxChannelTime * P(c) \Big)$$

- The Authentication delay depends on the security offered and required by the network. In its simplest form, the Open System authentication consists of a two-frame exchange. More complex security mechanisms, such as 802.1X, can cause longer delays.
- The Reassociation phase consists of a two-frame exchange, the *Reassociation Request* and the *Reassociation Response*, so this delay is considered constant.

Using wireless hardware from different manufacturers, experimental measurements showed that the handoff can take up to 2 seconds [72, 104]. During this interval, the station stops current traffic activity in order to perform the handoff. This process may disrupt ongoing connections, decreasing throughput and degrading the quality of communications. Specifically, the handoff delay is too long to be tolerated by real-time applications. For example, Voice over IP (VoIP) requires a one-way end-to-end delay not greater than 150 ms for good voice quality [102].

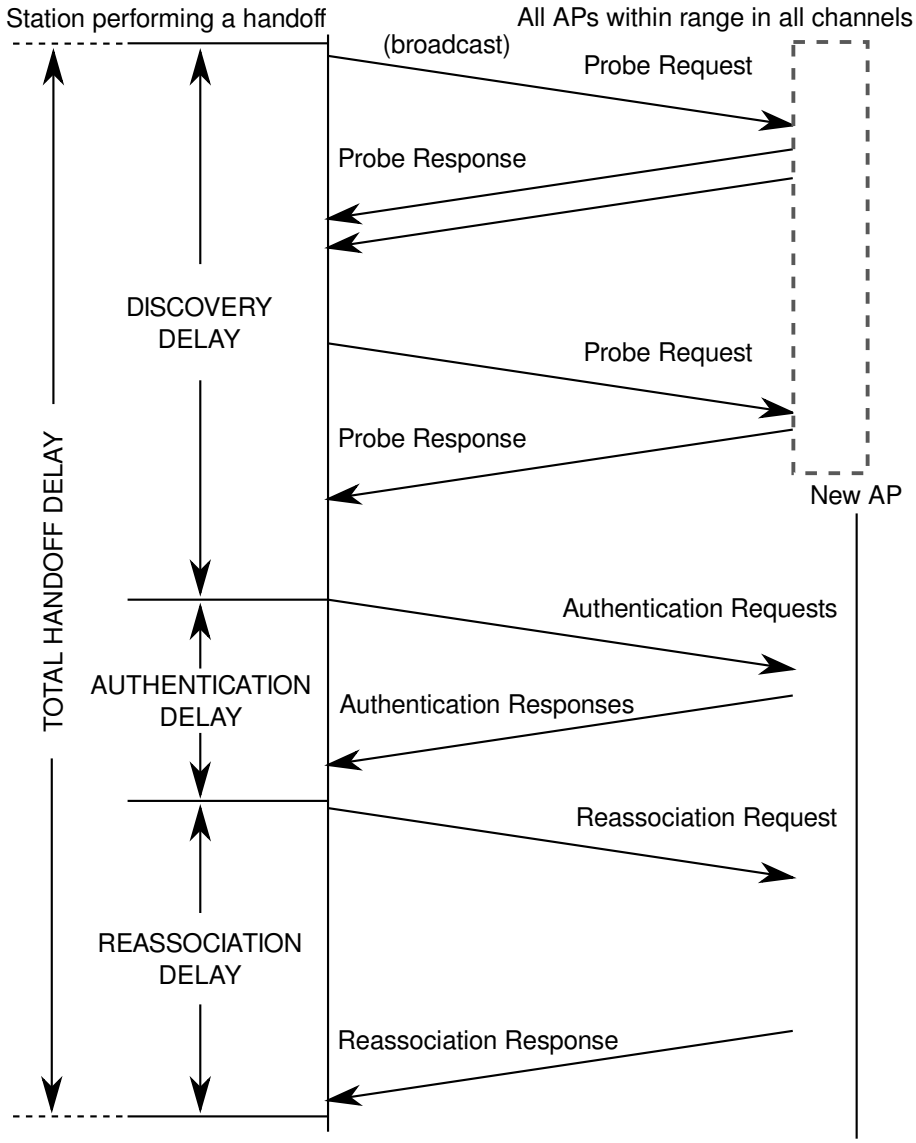


Figure 3.1: The different phases of the handoff process.

3.2.2 IEEE Standardization Efforts

The IEEE released two amendments related to the handoff process: IEEE 802.11f proposes a common protocol to enable communication among APs of different vendors, and IEEE 802.11r enables fast BSS transitions for mobile stations, while providing secure and seamless handoffs.

Inter-AP communications were not standardized in the original IEEE 802.11 specification. In fact, only proprietary methods enabled the transfer of the station's association session across APs, without disrupting link-layer connectivity. IEEE 802.11f was an attempt to introduce AP-to-AP communication and allow mobility within a single ESS, facilitating interoperability among APs of multiple vendors. However, this amendment was finally withdrawn.

Enhanced security mechanisms (such as 802.1X authentication) and multimedia support added more parameters negotiation when reassociating with a new AP, making the handoff delay longer. The IEEE 802.11r amendment specifies the Fast BSS Transition in an ESS, and provides a faster and secure roaming protocol, preventing connection loss and stream disruption. IEEE 802.11r requires both client and AP support. Enterprise-oriented products, such as Cisco, Intel and Apple, are now supporting this functionality.

3.2.3 Related Work

Minimizing the handoff delay has been extensively studied in the literature. Indeed, researchers have proposed a wide array of fast handoff schemes. These research efforts can be divided into the following approaches, described in detail below:

- reducing the discovery delay,
- reducing the authentication delay,
- improving the handoff detection mechanism,
- and providing infrastructure-controlled handoffs.

Reducing the Discovery Delay

Previous research has shown that the discovery is the most time-consuming phase of the handoff and can take more than 80% of the total handoff delay [72, 104]. Many strategies have been proposed to reduce the AP scanning time; they can be classified under three categories: *selective scan*, *proactive scan*, and *eavesdropping* (or passive scan).

The full-scanning technique consists in probing all the channels and waiting for a maximum fixed amount of time for the APs' responses (*MinChannelTime* or *MaxChannelTime*). This technique wastes resources, as some channels can be empty, or the APs in a channel may send back their *Probe Response* frames long before the *MaxChannelTime* elapses. Therefore, when a station performs a handoff within a single ESS, it can limit the scanning to a subset of all channels, observed during previous probings.

Using a selective-scanning technique, Park et al. [80] proposed the use of Neighbor Graphs (NG). This data structure is an undirected graph, in which each edge represents a mobility path between APs. This graph is generated by the APs, using *Reassociation Request* frames and IAPP Move-Notify messages (which notify a station's reassociation to its old AP). Client stations each receive a list of channels to scan, obtained from the NG and based on their current AP association.

Similarly, Shin et al. [93] proposed the use of NG to capture the handoff relationships among APs, but from the perspective of the mobile station. As a result, each station generates and maintains its own NG. Knowing in advance the operating channels of neighbor APs, this technique reduces the total number of channels to scan, as well as the

total time spent in each channel, as the station moves from one probing channel to another as soon as all APs in a channel have replied.

A first proactive scan technique alternates channel probing with transmission periods [61], providing a smooth handoff. The objective is to avoid scanning all the channels one after another and thus interrupting current communications during that time. Channels are divided into groups, so the station scans the whole set of channels in multiple stages, while still being able to exchange some traffic in between. As this scan technique takes longer than a standard one, the station triggers the discovery phase earlier, using a higher threshold. This value is adjusted by an adaptive algorithm, in order to reduce the frequency of channel scanning.

In the same way, Proactive Scan [109] is a fast handoff scheme that splits the scanning into shorter probing intervals, each followed by normal transmission activity. Each scanning interval is short enough to not interrupt current communications. But the station may lose frames when switching channels: therefore, the station announces its powersave mode to the AP, just before performing the scan. Moreover, the discovery phase is carried out in advance, even before triggering the actual handoff, thus reducing the total delay.

Active scanning creates a considerable overhead by probing all channels. SyncScan [84] is a passive scanning technique that continuously collects information about nearby APs. This technique requires all stations to be synchronized with the beacons broadcasted on each channel. Thus, stations can perform a passive scan of the APs by switching channels at the exact moment a beacon arrives. Although this technique is backward compatible with the 802.11 standard and only requires trivial modifications, the synchronization of the APs and the stations is a very complex task.

Passive scanning also has its limitations, as the station has to listen for the beacons in each channel, which takes a long time. D-Scan [100] proposes eavesdropping in dense 802.11 deployments, but using all the different frames transmitted by the APs, such as beacons, probe responses, and data frames. The station periodically measures the link quality of its current AP and, depending on this value, performs a background pre-scanning or initiates a handoff.

Reducing the Authentication Delay

Research works dedicated to reducing the authentication delay have primarily focused on proactive authentication mechanisms. Many of these were proposed long before the introduction of the 802.11r amendment, which is currently the standardized solution for fast handoffs. We describe the few examples from the literature that propose the most noticeable techniques: distribution of the station's context information (such as security and QoS parameters), pre-registration, and forward-and-buffer mechanisms.

The Proactive Caching algorithm [73] uses the NG data structure to reflect the reassociation relationships among APs. The NG information can be obtained in a distributed or centralized manner. This proactive technique forwards the station's context to potential

new APs, which are one hop ahead of the current AP, based on the information provided by the NG. When the station reassociates with one of these APs, its network settings have already been transferred to the new AP, thus reducing the handoff delay.

However, propagating the station's context to all the neighbor APs can result in high overhead. A selective neighbor caching (SNC) scheme [78] reduces the number of transfers by sending the station's context to only a selected group of APs.

Another solution [49] also uses the concept of NGs to reduce the authentication delay and prevent frame loss during the handoff. A central NG server maintains the NG structure and communicates with the client stations. The information exchanged (about channels and APs) allows a station to perform a selective scan. The authors also extend the IAPP scheme, providing a pre-registration of the station with the candidate AP, and a Forward-and-Buffer mechanism to transfer to this new AP the frames still held by the old AP.

Improving the Handoff Detection Mechanism

The handoff process is triggered by reactive mechanisms. The station starts a handoff only when the connection with its AP degrades considerably. Thus, the station has already experienced poor performance during some time, negatively impacting the quality of current communications. Various events can start the discovery phase:

(i) the station does not receive a number of consecutive beacons,

(ii) the station suffers several consecutive frame losses (i.e., it does not receive any acknowledgment for recent frame transmissions),

(iii) the current AP's Received Signal Strength Indicator (RSSI) is below a certain threshold.

Velayos and Karlsson [104] propose a link-layer detection algorithm, based on frame losses. In their technique, a station starts the handoff discovery phase immediately after the transmission of a frame and two subsequent retransmissions. The authors claim that three consecutive frame transmission failures are probably not caused by collisions. Similarly, in the case of a station not transmitting while moving, the handoff can be triggered after the station misses three consecutive beacons and, during that period, does not receive any traffic from the AP.

Mhatre and Papagiannaki propose proactive smart triggers for improved user performance [68]. A station continuously monitors the signal of nearby APs and executes one of the following triggering algorithms:

- The *Hysteresis* algorithm: triggers the handoff if the RSSI of an AP instantaneously exceeds the current AP's RSSI, plus a hysteresis factor.

- The *Trend* algorithm: triggers the handoff as the Hysteresis algorithm does, but uses trending information (instead of instantaneous values) in order to reduce excessive transitions between APs.

- The *Least Squares Estimator (LSE)* algorithm: predicts the future value of the current and nearby APs' RSSIs, and triggers the handoff if the predicted value for a nearby AP is higher than the predicted value for the current AP.

Infrastructure-Controlled Handoffs

All the previous schemes focus on the optimization of the client behavior, clearly because the handoff is entirely a client phenomenon. There is no protocol to help the APs induce or influence the handoff decisions, such as when to start the discovery phase, or which AP to attach to.

An infrastructure-controlled handoff is not a novel idea, as it is already used in cellular wireless networks [101]. However, this concept has not been sufficiently studied in 802.11 WLANs. With this technique, the station does not perform certain phases of the handoff process at all, as the network is in charge of, for example, choosing candidate APs and triggering the reassociation.

Moreover, new mechanisms for the client-based handoff require modifications of the client logic. Some of these involve a lot of processing and are therefore power-consuming, a clear disadvantage for mobile devices that rely essentially on batteries. Due to the proliferation of different manufacturers and platforms for mobile devices, updates of the clients' drivers became an extremely difficult task. On the contrary, introducing modifications in the APs is now feasible thanks to open-source drivers and other frameworks.

We firmly believe that an infrastructure-controlled handoff is interesting and should be explored in detail, so we propose such a new mobility solution, offering fast handoffs, in Chapter 5.

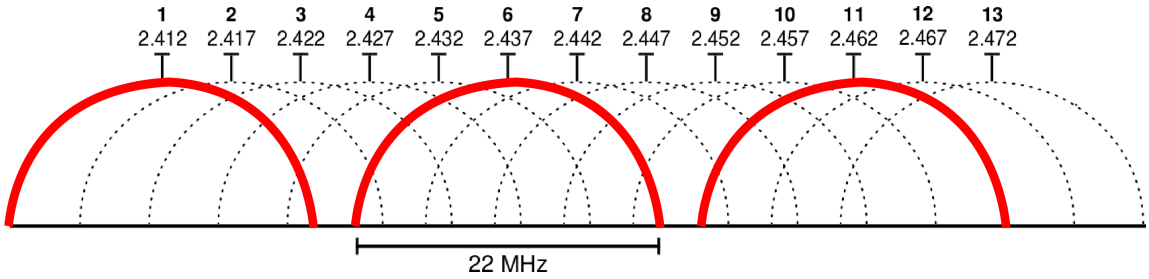
3.3 Channel Assignment

802.11 devices operate in the license-free 2.4 GHz and 5 GHz frequency bands. The 2.4 GHz band, used by 802.11b/g/n, is divided into thirteen channels (eleven for North America, fourteen for Japan). Thus, channel 1 operates in the 2.412 GHz frequency, channel 2 in the 2.417 GHz frequency, etc. The available channels in the 5 GHz band, used by 802.11a, may vary depending on the regulations of the different countries.

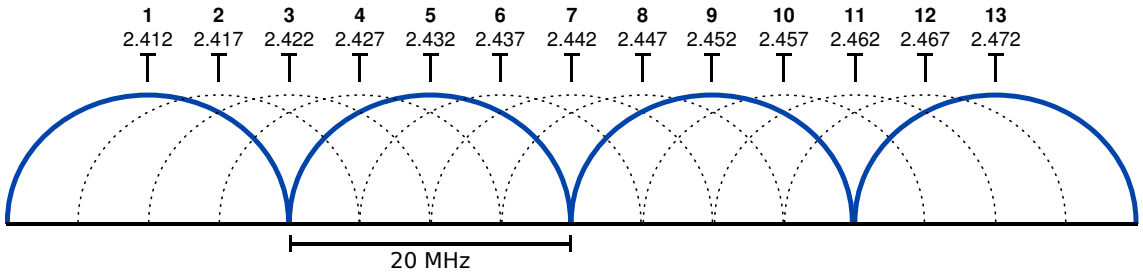
802.11 signals are transmitted in a channel that is specified by its center frequency and spread over its channel width. As a result, part of the signal can be heard in adjacent channels. For example, the 802.11b DSSS modulation uses a channel width of 22 MHz, as shown in Figure 3.2a. As channels are spaced 5 MHz apart, a separation of at least five channels is needed in order to completely avoid interference. Neighboring 802.11 devices in

overlapping channels are a source of interference and contention. Consequently, the overall throughput and performance decrease.

Orthogonal channels are non-overlapping channels that are separated by a minimum distance in order to avoid interference. In 802.11b, there are three orthogonal channels: 1, 6, and 11. The 802.11g OFDM modulation uses a channel width of 20 MHz; therefore, there are four orthogonal channels: 1, 5, 9, and 13, as depicted in Figure 3.2b. However, because 802.11g APs also support 802.11b stations and their DSS modulation, 802.11g WLANs are usually configured with the three 802.11b orthogonal channels. 802.11a has twelve non-overlapping channels, but some of these are for indoor use only, as their frequencies are also used by satellites and radars.



(a) 802.11b, with a 22 MHz channel width.



(b) 802.11g, with a 20 MHz channel width.

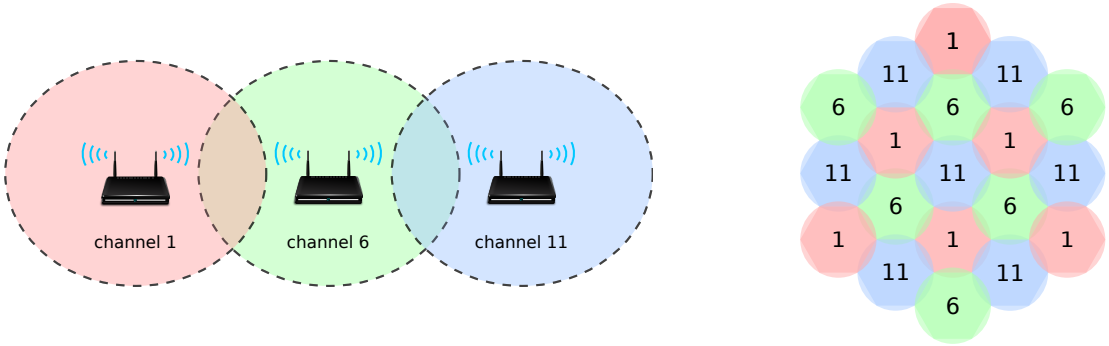
Figure 3.2: Orthogonal channels in the 2.4 GHz band.

An AP uses a determined channel for wireless communications, published in *Beacon* and *Probe Response* frames. All the stations associated with a particular AP communicate in the channel advertised by this AP.

When extending WLANs, two co-located APs allow continuous mobility, but may create (i) *co-channel interference*, if both APs operate in the same channel, or (ii) *adjacent interference*, if both APs operate in adjacent overlapping channels. Hence, the use of non-overlapping channels avoids interference and allows to take full advantage of the network capacity.

In a simple topology, as depicted in Figure 3.3a, channel assignment consists in choosing a different orthogonal channel for each AP. Figure 3.3b shows the hexagonal cell tiling

model, inspired from cellular wireless networks. It allows maximum capacity by having both uninterrupted coverage and maximum channel separation in large deployments. Each AP is represented by a hexagonal cell, and all its neighbors use the other two orthogonal channels. However, wireless coverage does not have a regular shape, so this channel allocation model is not always adequate for WLAN deployments. In fact, the three orthogonal channels provided by 802.11b/g are not sufficient for large three-dimensional topologies, where WiFi signals can pass through walls and floors.



(a) Simple topology. **(b)** Hexagonal cell tiling model.

Figure 3.3: Orthogonal channel assignments.

3.3.1 IEEE Standardization Efforts

The IEEE released the following amendments relevant to channel assignment mechanisms: spectrum management services in 802.11h, and radio resource management in 802.11k and 802.11v.

Spectrum and Transmit Power Management Extensions

Wireless devices operating in the 5 GHz band must employ Dynamic Frequency Selection (DFS) and Transmit Power Control (TPC) capabilities to avoid interference with satellites and radars. The DFS and TPC services are introduced by the 802.11h amendment.

DFS enables the AP to detect radar operation and migrate the BSS to another channel if a radar is found. The AP alone chooses the new channel and the moment to switch; it notifies the associated stations of the channel switch by means of a Channel Switch Announcement (CSA) element in its Beacon frames. The AP maintains the associations with the stations after the channel switch.

The CSA element announces the new channel *number* and *when* the BSS will be switching channels. The format of the CSA element is shown in Figure 3.4. As an information element, the first field is the Element ID, and the second is the length field, whose value is set to 3 bytes. The three specific CSA fields are:

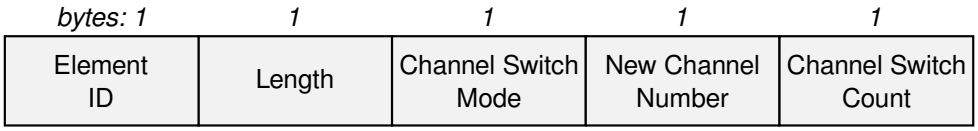


Figure 3.4: CSA element format.

- **Channel Switch Mode**: indicates transmission restrictions before the channel switch. If the value is set to 1, the stations must stop transmitting frames until the channel switch. If the value is set to 0, the stations have no transmission restrictions.
- **New Channel Number**: the new channel number of the BSS, after the switch.
- **Channel Switch Count**: the number of beacons the AP will send before the channel switch. This value decreases in each consecutive beacon. If the value is 0, the AP moves to the new channel immediately after sending the frame, as depicted in Figure 3.5.

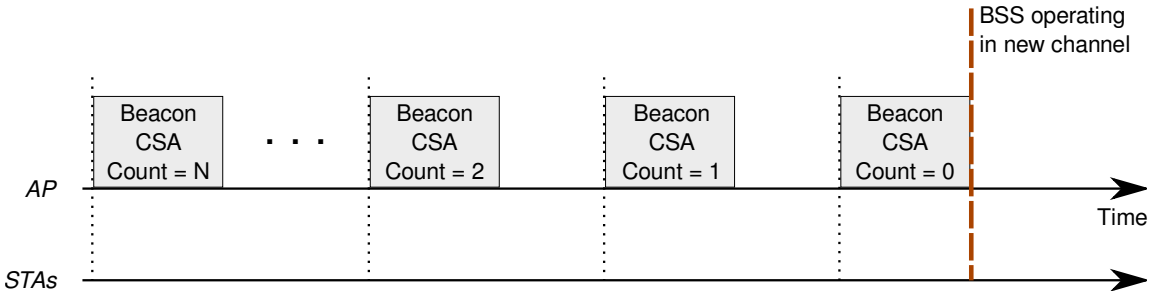


Figure 3.5: CSA example: the AP sends beacons with the CSA element, decrementing the CSA Count field in each beacon. When CSA Count reaches zero, the BSS (AP and associated stations) moves to the new channel.

TPC, the other capability introduced by the 802.11h amendment, automatically adjusts and adapts the transmit power level of radios, according to changes in the RF environment of the AP, in order to reduce interference with satellites and radars.

Radio Resource Management

The 802.11k amendment enables APs and client stations to perform measurements and collect statistics about their radio environment, with the objective of assisting the WLAN management and improving its performance. 802.11k proposes standard measurement mechanisms that do not depend on a specific vendor. Upper layers can take advantage of the information obtained in order to make efficient decisions, for instance, in channel selection, load balancing and roaming, through noise histograms, channel loads and neighbor reports.

The 802.11v amendment introduces Wireless Network Management (WNM) to the 802.11 MAC and PHY layers. It extends the mechanisms of 802.11k, by enabling APs and

client stations to exchange information about the network and radio conditions, such as interference and diagnostic reports, power management, and location services.

802.11k requires both AP and client support. Commercial products now implement this functionality, in a joint effort with 802.11r to provide fast and seamless handoffs. On the other hand, 802.11v is still not supported by wireless devices.

3.3.2 Related Work

Channel assignment has received great attention from the research community. Many different approaches have been proposed for either managed or uncoordinated WLAN deployments, in order to maximize the overall network throughput. Essentially, channel assignment techniques can be divided into two broad categories: *centralized*, in which a central entity decides the channel attribution for each AP in a WLAN, and *distributed*, in which each AP chooses its own operating channel, generally based on local measurements.

Since radio spectrum is a scarce resource, network administrators need to manage it efficiently. When planning a wireless network, administrators first conduct a Radio Frequency (RF) site survey, using a spectrum analyzer. With the information obtained, they determine the number of APs to deploy and their locations, in order to provide maximum coverage. Channel assignment is the next step, and depending on the size of the wireless network, this task may be performed manually [33].

Centralized Channel Assignment Schemes

Several authors proposed centralized offline algorithms to solve the channel allocation problem in large managed WLANs. All these proposals use only the three non-overlapping channels 1, 6, and 11. Wertz et al. [107] divide the floor plan into pixels and give priorities to each pixel, based on traffic requirements. Channels are assigned following a greedy heuristic, using this priority map. Hills [47] also elects the channels of the APs with higher traffic demands first, and then concentrates on the remaining APs.

Knowing the traffic demands and locations beforehand, Lee et al. [59] optimize both AP placement and channel selection at the same time, using Integer Linear Programming (ILP). Evaluating all the possibilities to obtain the optimal solution with a brute-force approach can take a considerable amount of time. Thus, Ling and Yeung [63] propose a heuristic called patching algorithm for the joint problem of AP location and channel assignment. This algorithm gradually places the APs, meeting the traffic demands, and determines the channel distribution at each step.

Channel allocation can also be modeled as a graph coloring problem: the APs are the vertices of the graph, and each edge represents a potential interference between two APs with overlapping coverage. Coloring consists in assigning a color to each node, so that adjacent vertices do not have the same color and the number of utilized colors is minimal. However, obtaining an optimal solution is NP-hard. In a series of research

papers [65, 87, 88], the authors use the graph coloring model and implement a heuristic called DSATUR [16]. Each AP first constructs its neighbor list and sends this information to a central server, which is responsible for building the complete graph and performing the channel allocation. The channel assignment is dynamically generated (as opposed to previous offline algorithms [41, 28, 13, 29] which are static and do not take into consideration the changing nature of channel conditions): hence, this algorithm is reapplied periodically or whenever new topology information is available.

Moving from an *AP-centric* approach, in which interference is measured only at the APs, to a *client-centric* approach, in which clients perform the interference detection, Mishra et al. [70] propose a centralized algorithm called CFAssign-RaC (ConFlict set color Assignment using Randomized Compaction). This technique addresses both problems of channel assignment and load balancing with a graph that captures different types of conflicts: APs reachable from the client's location (potential APs for load balancing), and APs and client stations within one-hop range of the client or its current AP (potential interferers). To collect the graph information, stations periodically conduct a *Neighbor Report*, as specified in 802.11k, in which all APs found nearby are listed. The algorithm performs channel assignment in order to minimize the interference conflicts among the APs. Channel re-use is applied whenever two adjacent APs do not have any station in their overlapping region. This technique is regularly executed and can also be dynamically triggered.

Distributed Channel Assignment Schemes

Distributed channel assignment schemes are appropriate for uncoordinated environments, but managed WLANs can also implement this type of approach. Least Congested Channel Search (LCCS) [4] is a simple and common technique provided by commercial APs. As the AP starts up, it listens to every channel, waiting for Beacon frames broadcasted by neighbor APs. These beacons include the number of associated clients in the BSS. Based on this information, the AP chooses the least crowded channel. However, this technique requires that all APs be from the same vendor, because it uses particular proprietary fields.

Similarly, Yu et al. [112] develop a dynamic, distributed channel assignment mechanism. Each AP estimates the number of active stations in its current channel. The AP then monitors the other channels and switches to the one with the least channel utilization. APs execute this heuristic independently, without inter-AP communication.

An overlapping-channel allocation solution can in some cases achieve better performance than a solution that uses only the three orthogonal channels [69]. The algorithms Hminmax and Hsum use a weighted variant of the graph coloring problem, where the weight of an edge represents the number of clients associated with the two vertex-APs. Both algorithms exploit local information, obtained through 802.11k *Neighbor Reports*, and make decisions in a distributed way. In Hminmax, each AP executes the algorithm independently and

periodically. On the other hand, the Hsum algorithm requires inter-AP cooperation in order to exchange the interference metrics.

Akl and Arepally [10] propose a dynamic channel selection algorithm that also uses overlapping channels. Each AP executes the algorithm and chooses the channel with the least interference, estimated as a function of the overlapping channel interference, the AP's transmit power, and the path loss between interfering APs. The authors propose two variants of their algorithm when there are several channels to choose from: Pick-Rand, in which the AP randomly picks a channel, and Pick-First, in which the AP picks the first channel of the ordered channel list.

Using a different interference metric, Kauffmann et al. [53] propose a distributed algorithm that minimizes the total interference in uncoordinated WLANs. Channel selection is performed using a Gibbs sampler, which takes into account the downlink traffic. For this purpose, APs and stations measure the local interference and the APs' transmission delays, a functionality supported by 802.11k. Channel switch is done regularly, when a timer expires.

Finally, the algorithm MAXchop [74] is a distributed channel assignment algorithm based on channel-hopping, which requires minimal coordination among APs and is compatible with the current 802.11 standard. Each AP maintains a channel-hopping sequence (a list of channels in which it will operate) and hops through this sequence over time. All APs need to be synchronized, since they should all switch channels at the same time. Stations measure the interference from neighbor APs, and this information is used to build the hopping sequences. This technique provides long-term fairness among the APs.

3.4 Quality of Service

With the spread of multimedia applications, such as videoconferencing and online gaming, 802.11 WLANs struggle to provide the expected quality of service. Furthermore, new physical layers with higher data bit rates do not suffice to guarantee services such as bandwidth, delay, jitter and packet loss, to these applications. The 802.11 MAC sublayer is at the origin of the problem, because it is responsible for controlling channel access.

DCF, the fundamental 802.11 MAC coordination function, was originally designed for best-effort services. Performance evaluation results in [23, 25] show that DCF experiences important throughput degradation and high latency during heavy load conditions. Thus, various research efforts have been devoted to providing some QoS enhancements: differentiated services at the network layer [46], station-based schemes [1, 51, 95, 103, 105], and queue-based schemes [2, 66, 90].

Moreover, the DCF protocol offers equal probability of medium access to every associated station, causing a performance anomaly in BSSs with multi-rate stations [44]. Indeed, low bit rate stations limit the overall throughput of the cell, penalizing stations with higher bit rate by degrading their throughput. Different approaches to providing fairness among

stations have been proposed [27, 45, 94, 99, 110], but many of these novel techniques require client-side modifications, introduce changes in the firmware of the wireless cards, or depend on new MAC protocols that are incompatible with the current 802.11 MAC. Unfortunately, their limitations prevent these techniques from being implemented in commodity hardware and their subsequent deployment in ordinary WLANs.

3.4.1 802.11e: QoS Enhancements

Accordingly, the 802.11e amendment aimed at providing enhancements in the MAC sublayer, introducing QoS features and support for real-time multimedia applications in WLANs, while maintaining full backward compatibility. A station using QoS is called *QSTA* and an AP using QoS is called *QAP*. A QAP may allow associations with non-QoS stations.

802.11e defines a new MAC protocol: the hybrid coordination function (HCF), which was briefly described in Section 2.4. HCF combines and enhances both DCF and PCF, enabling prioritized and parameterized QoS access to the wireless medium for QSTAs, while providing best-effort services for non-QSTAs. HCF supports two mechanisms: a contention-based channel access method, called Enhanced Distributed Channel Access (EDCA), and a polling-based channel access method for contention-free operation, called HCF-Controlled Channel Access (HCCA).

HCCA provides parameterized QoS access and may be used by application flows that require delay or bandwidth guarantees, such as video and voice. Transmissions are regulated by a central entity called hybrid coordinator (HC), collocated with the QAP. QSTAs request transmission slots and negotiate QoS requirements with the HC. HCCA alternates contention-free periods (CFP) with contention periods (CP), and uses DCF exclusively during the CPs. Since HCCA support is optional, it has not been implemented in commercial wireless products.

EDCA provides prioritized QoS using a CSMA/CA access mechanism. Traffic differentiation is achieved through the use of four Traffic Categories (TCs): Background (lowest), Best Effort, Video, Voice (highest). MAC frames with higher TC priority are given precedence over frames with lower priority. The TC of a given flow is specified by the application layer, in the ToS field of the IP packets.

Each TC has its own transmit queue and corresponding parameters: *CWmin* and *CWmax*, the contention window limits used by the backoff algorithm, and the *Arbitration Interframe Space (AIFS)*, the minimum idle time before transmission of a frame. The value of AIFS is the sum of a SIFS interval and the duration of the number of slots indicated by the AIFS Number (AIFSN) parameter; the minimum value of AIFS is the same as DIFS ($2 \cdot Slot\ time + SIFS$). All these parameters are announced in the Beacon, Probe Response, and (Re)Association frames, sent by the QAP. When two or more frames of different TC queues try to simultaneously access the wireless medium, an *internal collision* occurs. The

queue with the highest priority proceeds with the channel access, and the lower-priority queues start the backoff algorithm, as if a classical collision had occurred.

In spite of the higher channel access probability for time-sensitive applications, such as video and voice, EDCA cannot provide throughput guarantees: in heavy load conditions, the performance of these applications degrades significantly. Worse yet, flows that come from different stations but belong to the same TC still compete for access to the wireless medium.

EDCA is implemented in commercial hardware; actually, WiFi Multimedia (WMM)-certified APs must support EDCA. Also, EDCA is compatible with DCF, since non-QSTAs use DCF to exchange frames with the QAP.

3.4.2 Related Work

We examine three distinct approaches that propose priority mechanisms in order to guarantee traffic requirements. Contrary to the majority of QoS-related research proposals, which were mainly studied through analytical and simulation methods, these approaches have been experimentally evaluated and effectively implemented in current 802.11 hardware.

Han et al. [42] present the Channel Access Throttling (CAT) scheme, an extension of EDCA. CAT provides access-priority groups (similar to traffic categories) that stations can dynamically join. As a result, some stations have a higher priority than others when transmitting, independently of their flows' traffic category. CAT is compatible with EDCA and can be implemented in the MadWifi driver; however, it requires both AP and client support.

The prioritization of a specific station's traffic can be achieved with SHAPE [22], a Smart Home Access Point Environment that supports multimedia services. SHAPE implicitly reserves the channel for a certain station by sending it unsolicited CTS frames, forcing the rest of the stations to defer their transmissions. This mechanism provides the required bandwidth and significantly reduces the delay jitter, using the basic DCF (not 802.11e). SHAPE is implemented in the AP, so existing client stations can benefit from this technique without any modification.

Although WiFox [40] does not yet support QoS, it offers an interesting approach to solving traffic asymmetry in large audience environments. WLAN traffic is mostly downlink, but the AP merely gets the same share of channel access as every contending station, causing congestion at the AP. WiFox uses basic 802.11e concepts to provide adaptive prioritization of the AP's channel access over competing stations. Specifically, WiFox assigns high priority values to the AP's channel access parameters during heavy load conditions at the AP. This mechanism only introduces modifications in the AP and was deployed in off-the-shelf commercial hardware.

3.5 Summary

In this chapter, we described three aspects of WLANs that must be taken into account during their design and deployment: client mobility, channel assignment, and quality of service. These aspects suffer from performance limitations that have been largely discussed in the literature: the research works proposed diverse solutions that improve the WLAN performance and support new services. Indeed, new functionalities have been added to the original 802.11 standard through subsequent amendments.

However, despite the enhanced capabilities of the 802.11 standard, WLANs face new challenges introduced by novel applications and uses, such as multimedia streaming and location-based services. Moreover, we observed that many solutions proposed by the research community introduce changes in the client behavior or modify the MAC protocol, making them incompatible with legacy clients. Also, static or offline techniques do not reflect the changing dynamics of the wireless medium and network properties. These factors impede the deployment of the previously described proposals in current WLANs.

In the following chapters, we first exploit the observations made in this chapter to build a common basis for practical implementation of new 802.11 solutions. Then, we identify open challenges in different WLAN scenarios and present the main contributions of this thesis: novel mechanisms that can be effectively deployed in current WLANs.

Chapter 4

SMART ACCESS POINTS

Contents

4.1 Introduction

We present our *Smart AP* model, for developing practical AP-based solutions using self-management techniques in existing WLANs. As discussed in the previous chapter, there are several issues that affect the IEEE 802.11 WLAN performance. These problems have been much discussed in recent literature and numerous approaches to improve the WLAN operation exist. However, many of these techniques have limitations that prevent their deployment in current WLANs, since they propose manual or static optimization mechanisms, cannot be practically implemented in today's wireless cards, or require modifications in the IEEE 802.11 standard, neglecting interoperability with existing WiFi devices.

The objective of our thesis is to develop solutions that improve the performance of WLANs, and most importantly, to deploy these solutions in existing WLANs. We focus on approaches that can be implemented in off-the-shelf hardware, do not require client-side modifications, are backward compatible with legacy wireless devices, and manage the

available resources in a dynamic way. Therefore, in this chapter, we introduce our *Smart AP* model, to provide a common foundation upon which to build our solutions.

In Section 4.2, we first outline the limitations of existing schemes, which motivate us to propose the *Smart AP* model. Then, we describe the design goals of our model in Section 4.3, and its architecture in Section 4.4. Finally, we summarize the chapter in Section 4.5.

4.2 Motivation

In recent years, WLAN performance issues have been largely discussed in the literature. These works are accompanied by techniques and algorithms to enhance the WLAN operation, and thus deliver increased capacity, higher throughput, fairness among clients, and quality of service, to wireless traffic.

However, many of these research efforts suffer from several limitations that prevent their practical and correct deployment in existing WLANs:

- **Low-level modifications:** Several techniques, such as new MAC protocols and collision or interference measurement methods [34, 45, 98], configure low-level parameters and change functionalities in the Physical and MAC layers. Thus, they require access to the firmware of the wireless card, which in many cases is proprietary and not available for modification [36].

- **Client support:** In the past, the introduction of changes in the infrastructure was not possible, due to proprietary source code restrictions. Clients were fairly homogeneous, and client code development was relatively easy. Consequently, numerous approaches proposed changes in the client behavior, for example AP selection and association techniques. But nowadays, clients are remarkably diverse: laptops, smartphones, tablets, and a variety of vendors and manufacturers such as Nokia, BlackBerry, and Apple. Hence, supporting all these different platforms is rather impractical.

- **802.11 modifications:** Introducing fundamental changes to the IEEE 802.11 standard can provide new functionalities or improve the network performance [45]. However, these novel methods prevent interoperability with legacy (or even current) WiFi devices. Additionally, vendors take time to introduce new features in their hardware, so modifications in the standard reduce their impact.

- **Unsuitable centralized model:** In most centralized wireless networks such as Enterprise WLANs, there are two primary components: a central controller, which is in charge of the management and configuration activities, and the APs, which are essentially the end-points of the network [21]. APs in these networks act as mere MAC layer bridges, forwarding frames between the wireless clients and the wired network.

In recent years, uncoordinated WLANs gained importance because of the increasing number of hotspots and residential wireless networks. However, performance optimizations designed for centralized WLANs are inappropriate, as uncoordinated WLANs have distinct characteristics (for instance, the lack of a central authority).

- **Manual or static configurations:** Usually, several WiFi parameters such as the operating channel are assigned in a manual way, which can be time-consuming. Moreover, wireless channel conditions are known to vary over time, as are other network characteristics (the number of clients, the type and amount of traffic). However, static configurations of WiFi devices assume a stable environment and do not adapt to changes, failing to provide better performance.

4.3 Smart AP Model

The limitations outlined in the previous section motivated us to conceive a framework for developing new optimization techniques for existing WLANs. With this in mind, we propose our *Smart AP* model, inspired by Autonomic Computing [55].

The AP is the central entity of a WLAN, responsible for establishing and maintaining connections with client stations, and assigning the available resources to cope with their demands. Therefore, AP-based approaches can benefit from this leading role and improve the network performance.

Self-management techniques can give autonomy and "intelligence" to the AP. Essentially, decision-making should be distributed and decentralized. Based on the autonomic control loop depicted in Figure 4.1, the AP should: continually monitor the wireless environment and measure its current performance; detect problems and repair them; and automatically reconfigure and adapt to changes, in order to optimize the WLAN performance.

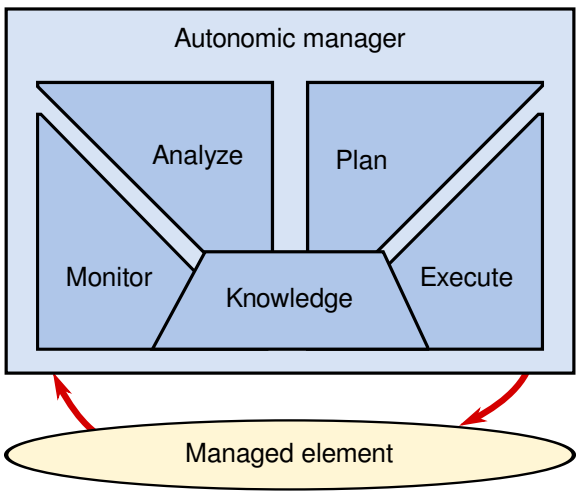


Figure 4.1: Autonomic control loop.

Our *Smart AP* model follows the design guidelines detailed below:

- **Use of commodity hardware:** Commercial off-the-shelf WiFi hardware is inexpensive (as opposed to Software-Defined Radio) and usually runs open-source software. Lately, most of the 802.11 protocol is implemented in the driver software, leaving only low-level and real-time operations (for example, the DCF) implemented in the wireless card's firmware. Development of new functionalities in software is more flexible and permits rapid rollout.

- **No client-side modifications:** Nowadays, infrastructure-based solutions are easier to implement, since many commercial AP products use OpenWrt [77], a common Linux platform for residential gateways and WiFi routers. Vendors and ISPs provide periodic firmware updates, so new features can be made readily available.

- **Backward compatibility:** By creating extensions and adding new functionalities compatible with the 802.11 standard, novel techniques can be rapidly deployed and transparently used with regular (and legacy) WiFi devices.

- **Self-management:** APs make decentralized and distributed decisions based on their own measurements and on client feedback. Techniques falling into this category are more suitable for unmanaged environments.

- **Dynamic optimization:** Adaptive reconfiguration reflects the wireless medium dynamics, as well as the different types of clients and traffic characteristics. By means of online algorithms, the AP reacts to changes, adjusts parameters, and, accordingly, improves the WLAN's performance and maintains flexible objectives.

- **Cross-layer techniques:** Layered protocols are not optimal for wireless networks, because they were designed for wired networks [97]. Cross-layer design enables interaction and collaboration between layers. Thus, by combining information from the Physical and MAC layers, we can obtain performance improvements.

4.4 Architecture

We present the basic architecture for the implementation of our *Smart AP* model, depicted in Figure 4.2, which consists of the following components:

- **Wireless NICs:** We use commodity 802.11 wireless cards (with omni-directional antennas) based on Atheros chipsets, which are widely adopted by the research community. These cards have open-source drivers that permit direct modification of their code.

  We use two wireless cards: the *AP interface*, dedicated to the AP functionality, and the *monitor interface*, to monitor activity. Attaching an extra wireless card to an AP

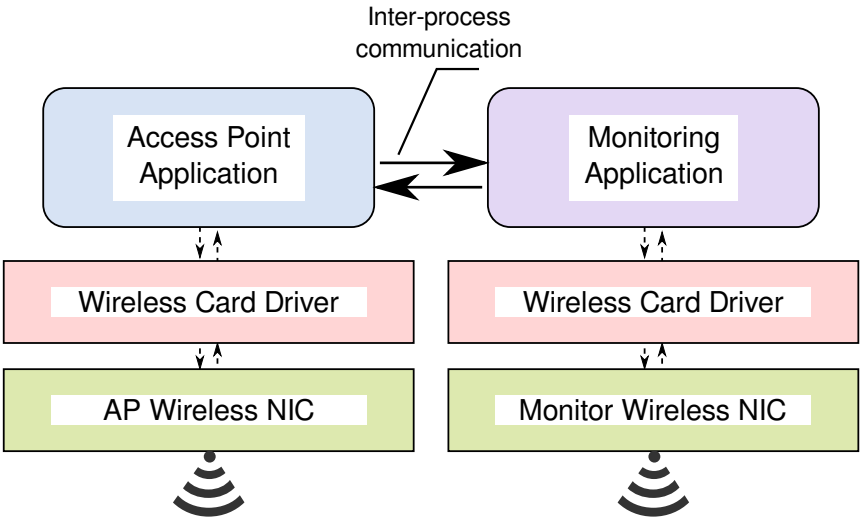


Figure 4.2: *Smart AP* architecture.

is easily done, since many wireless router models have USB ports, and wireless cards with USB adapters are quite cheap. Moreover, OpenWrt [77], the widely adopted operating system for WiFi APs, supports a large number of wireless NIC drivers.

- **Wireless Card Driver:** This driver lives in kernel space and links each wireless card to the AP and Monitoring applications, respectively. The driver's main function is the correct transmission and reception of frames. For our experiments, we used the MadWifi driver and its successors ath5k (for 802.11a/bg-enabled wireless NICs) and ath9k (for 802.11n-enabled wireless NICs) [108]. These drivers have the advantage of implementing correctly the complete 802.11 standard and most of the subsequent amendments.

- **Applications:** We implement our solutions as applications that run in user space. This choice alleviates the dependence on driver and kernel versions.

  The AP and Monitoring applications can exchange information through **inter-process communication**.

  - **Access Point Application:** This application is in charge of the AP management functionality. As an example, hostapd [48] is an 802.11 AP daemon service commonly used on Linux platforms, and supports various authentication modes (IEEE 802.1X, WPA, WPA2, EAP, and RADIUS).

  - **Monitoring Application:** This application, by setting its wireless card to monitor mode, can sequentially hop through all channels or listen to a specific one, and capture all available frames, not only those actually meant for the AP. Thanks to this application and the monitor wireless NIC, the AP can follow the activity of any channel, while serving the clients in its own channel at the same

time, with its AP wireless NIC. It can also obtain local measurements such as channel usage, and keep records of the stations that enter or leave its coverage area.

4.5 Summary

In this chapter, we presented our *Smart AP* model, based on self-management techniques, for creating practical AP-based solutions to improve WLAN performance. Our efforts are directed toward novel approaches that use commodity hardware, introduce changes only to the infrastructure (and not to the client behavior), are fully compatible with existing 802.11 devices, and dynamically manage and optimize the available resources, by exploiting cross-layer information.

The design and architecture of the *Smart AP* described in this chapter will serve as a guide for the three contributions of this thesis, which follow in the next chapters.

Chapter 5

TRANSPARENT MOBILITY FOR ENTERPRISE-CLASS VOIP SERVICES

Contents

5.1 Introduction

We now present the first contribution of this thesis: a transparent mobility mechanism for Voice over IP (VoIP) services in Enterprise WLANs. This type of WLANs is characterized by the deployment of several APs that all belong to a same administrative domain and therefore offer a wide-area coverage. Within such a WLAN, client stations can move freely,

but because of the short-range nature of the APs, they usually need to reassociate with different APs in order to continue their communications. When changing APs, a client station starts a process known as *handoff* that can take up to 2 seconds, too long a delay for real-time applications such as VoIP. Using the concept of Virtual Access Points (VAP), we have developed a mobility solution called Multichannel Virtual Access Points (mVAP) that provides seamless handoffs without performance degradation of applications with tight delay constraints.

In Section 5.2, we first introduce the problem of mobility in WLANs when running real-time applications such as VoIP, and discuss the related work in Section 5.3. Next, we present our mVAP solution and its design in Section 5.4, describe its implementation and our PACMAP framework in Section 5.5, and evaluate its performance in Section 5.6. Last, we point out possible future work in Section 5.7, and summarize the chapter in Section 5.8.

The results presented in this chapter were published in the proceedings of IEEE 73rd Vehicular Technology Conference (VTC Spring), in May 2011 [14].

5.2 Problem Statement

Over the past years, IEEE 802.11 WLANs have become the preferred solution to extend wired networks, thanks to their rapid deployment and easy configuration. These characteristics, in addition to low-cost hardware, have caused an increasing growth of WLANs. Wireless networking also brings the advantage of mobility, allowing clients to roam freely.

We consider 802.11 infrastructure networks in which APs convey traffic between associated clients and the wired part of the network. Examples of such networks are university campuses, convention centers, airports, and corporation intranets. Because APs have a limited range, coverage can be extended to a larger area by deploying multiple APs (for example, one AP in every office in the case of an enterprise WiFi network), thus resulting in a densely deployed network. APs are interconnected through a Distribution System (DS), generally a wired network, to enable inter-AP communications.

As detailed in Chapter 3, a station can join the wireless network by associating with an AP. When a station moves away from this AP, its signal falls off. If it drops below a certain threshold, the station starts searching for a new AP to associate with, initiating the MAC layer *handoff* process, until the new association takes place. The handoff delay takes a significant amount of time: up to 2 seconds, as measured in [72, 104]. During this time, the station neither receives nor sends data packets, which may interrupt current connections. Consequently, the handoff delay is too long for real-time applications like VoIP, which recommend a one-way end-to-end delay not greater than 150 ms for good voice quality [102].

Several solutions have been proposed to improve different handoff phases: discovery of new APs, re-authentication, and reassociation. Most of these solutions modify the client

behavior, because the client is the one in charge of its association and handoff process when moving. But as handheld VoIP WiFi devices become increasingly popular, such solutions are not practical due to, for example, proprietary source code restrictions, as discussed in Chapter 4. The modification of the AP behavior, instead of the client's, is therefore an interesting alternative way to improve different aspects of wireless client mobility.

In this light, Grunenberger et al. proposed the concept of Virtual Access Points [37], in which the VAP manages the client mobility. However, a serious drawback of this approach is that all APs need to operate in the same channel. Indeed, APs in managed environments like Enterprise WLANs usually operate in different channels, in order to avoid interference and increase network capacity.

Taking all the previous problems into consideration, we designed a complete mobility solution that provides seamless handoffs in multichannel wireless networks, can be deployed in off-the-shelf hardware, and requires no modifications on the client side.

Thus, the contributions of this chapter are as follows:

- Proposal of our new solution, *Multichannel Virtual Access Points* (mVAP), for seamless and efficient handoffs.
- Implementation of mVAP in a real environment, using a brand new version of our PACMAP framework, and running on top of the MadWifi [64] driver.
- Experimental evaluation of the mVAP performance with different VoIP codecs (8, 16, and 64 Kb/s).

5.3 Related Work

Several authors proposed fast-handoff schemes to reduce the handoff delay, as described in Section 3.2. They fall into the following main categories: (1) reducing AP scanning (probe) time by using different strategies of channel scanning, such as proactive scan [109], selective scan [61], eavesdropping [84, 100], and (2) reducing the authentication and reassociation time, for example by proactive distribution of authentication information [73]. Nevertheless, all these schemes focus on modifying the client behavior, clearly because the client is the device that controls the handoff process.

Network-based mobility management is another interesting approach, since the AP can negotiate the client's reassociation beforehand and therefore reduce the handoff delay. Indeed, some solutions in the literature develop this aspect: HaND [24] is a handoff technique in which APs exchange information about moving clients, and the current AP itself triggers the handoff procedure. The disadvantage of this technique is that it also introduces modifications in the client behavior.

On the other hand, OmniVoice [7], a mobile voice solution for small-scale organizations, does not require any client-side modification. With a single-channel WLAN design, this solution uses a central controller for managing interferences and scheduling transmissions.

Similarly, the Virtual Access Points (VAP) [38] technique mentioned before does not modify the client stations. Providing one VAP for each client, APs control and handle the stations' association session. The clients themselves believe they are always connected to the same VAP, and therefore avoid the handoffs altogether when moving.

Although both solutions provide seamless handoffs by means of network-based mobility and both can be used by regular 802.11 clients, their use is restricted to single-channel WLANs, an uncommon and highly inefficient configuration.

5.4 Multichannel Virtual Access Points

5.4.1 Overview

We reuse the idea behind VAP in order to develop Multichannel Virtual Access Points (mVAP), a complete solution for seamless handoffs in Enterprise WLANs where APs listen to different channels. A WLAN that deploys mVAP actually broadcasts two different network names:

- the "Voice" WLAN (composed of VAPs) in which client stations benefit from our new handoff scheme and can switch from one AP to another without performance degradation of their real-time applications;
- the "Default" WLAN (composed of regular APs) that handles the rest of the traffic.

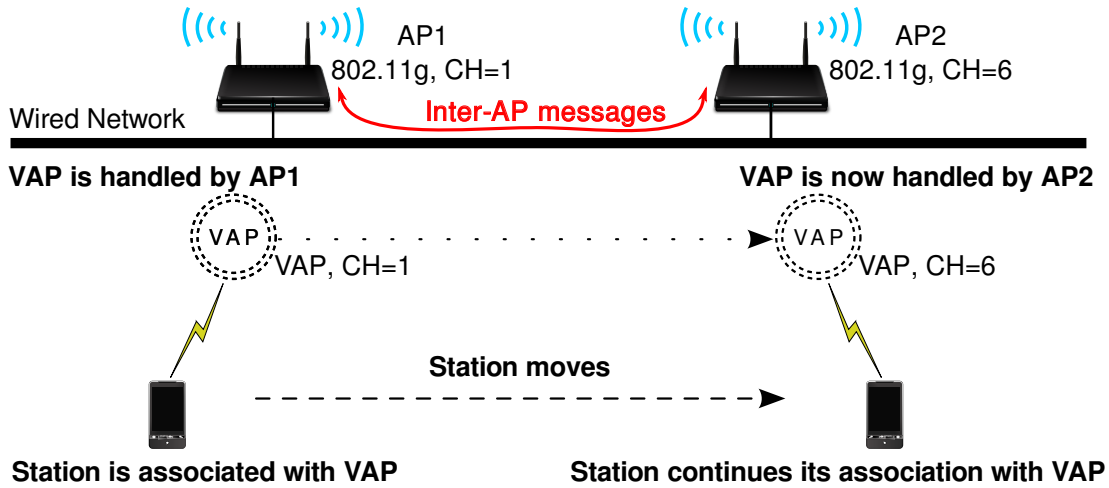


Figure 5.1: Mobility management with mVAP.

Mobility management in mVAP consists in the following procedure, as shown in Figure 5.1. When a station wishes to join the Voice WLAN, the first AP in charge of the station's connection creates a dedicated VAP for this particular station. Therefore, each station associates with its own unique VAP and maintains connectivity by the continuous reception of beacons. When the station moves, APs communicate with each other in order to move the association state to the new AP, which now handles the VAP for the station.

Therefore, the station avoids starting a handoff when moving, because it has the impression of being always connected to the same VAP, thanks to the cooperation between APs to create this transparent operation.

The APs that implement mVAP adhere to the *Smart AP* model, introduced in Chapter 4. Specifically, they can service their own clients (from both the Voice and Default networks) via their *AP interface*, while monitoring roaming clients at the same time via their *monitor interface*, as depicted in Figure 5.2. Additionally, as APs operate in different channels, they can cooperate through the DS, commonly using an Ethernet interface in order to exchange Inter-AP messages.

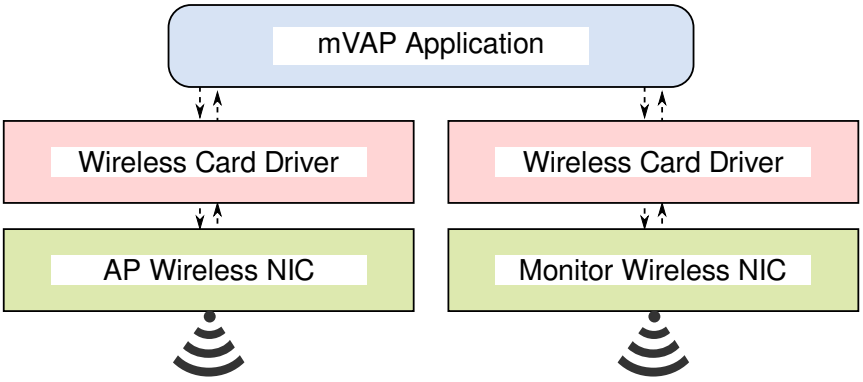


Figure 5.2: Architecture of mVAP.

The other key element in the mVAP solution is the Channel Switch Announcement (CSA): when a client station moves and needs to associate with another AP and switch channels, it receives the necessary information by means of a CSA in the VAP beacon, as described in Section 3.3.1. Finally, if no new AP can be found to handle the VAP of a moving client station, the client falls back to the standard handoff mechanism, also making this solution compatible with legacy 802.11 devices that do not support the CSA.

5.4.2 Protocol

The following steps form the basis of the mVAP protocol, depicted in Figure 5.3:

1. A client station (STA) is connected to the Voice WLAN and is associated with AP_i in channel i. STA starts moving and AP_i detects that the signal of STA drops below a threshold *Threshold*.

2. AP_i sends a **Scan Request** message to its neighbor APs ($AP_{j \neq i}$) through the DS.

3. All $AP_{j \neq i}$ that receive the **Scan Request** message switch their *monitor interface* to channel i and listen to STA packets for a short period of time.

4. If AP_j successfully listens to STA packets, it sends a **Scan Response** message to AP_i through the DS.

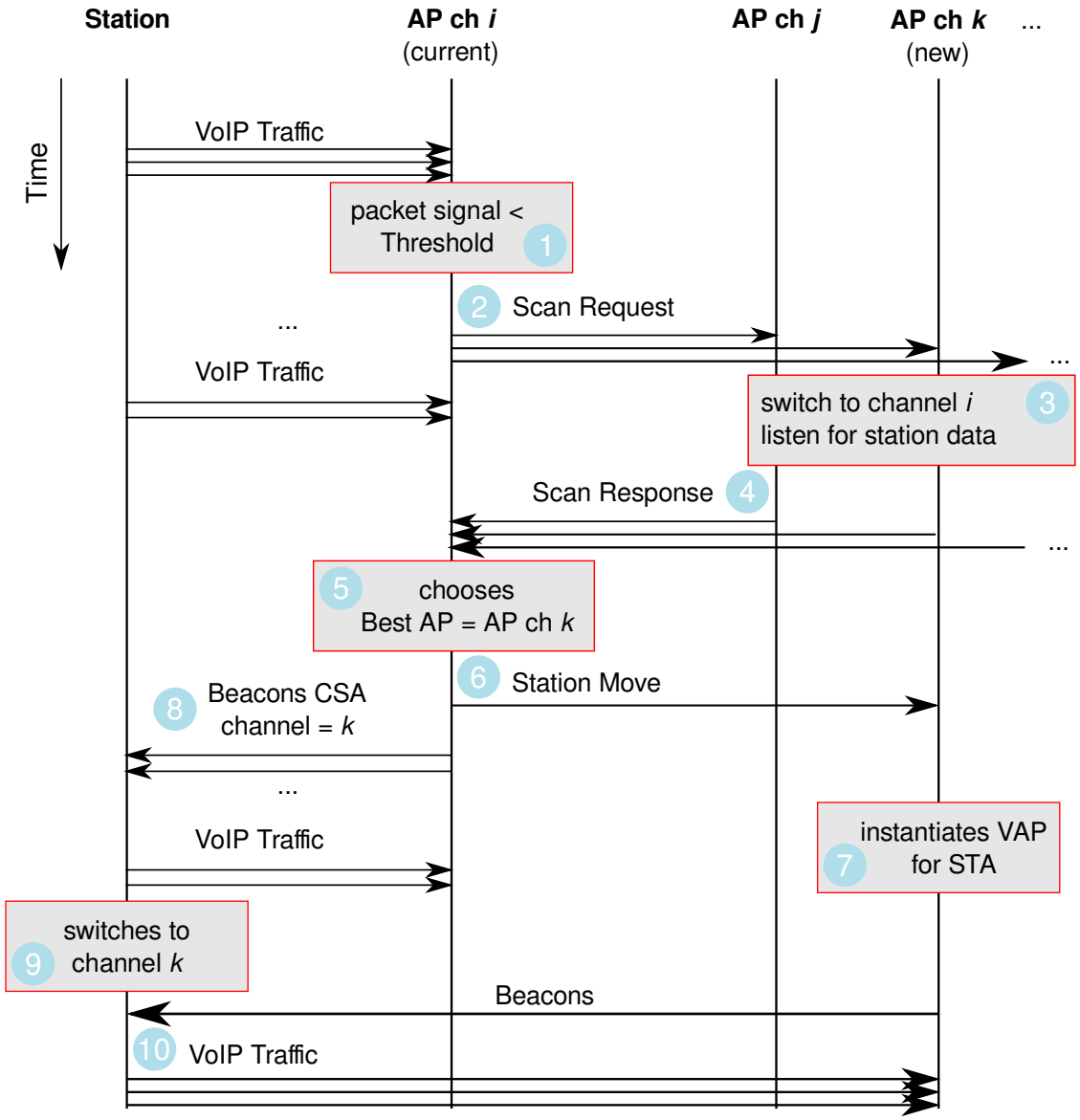


Figure 5.3: mVAP protocol.

5. AP_i collects the **Scan Response** messages and chooses the AP with the best signal (if better than its own).

6. AP_i sends a **Station Move** message to the chosen AP_k through the DS. AP_k's *AP interface* is servicing its clients on channel k.

7. AP_k receives the **Station Move** message. Thus, it instantiates the VAP for STA and starts sending the corresponding beacons (in channel k).

8. AP_i sends beacons (in channel i) to the STA, with the CSA element set in order to force the STA to switch to channel k.

9. STA receives the beacons with the CSA element and switches to channel k.

10. STA has successfully moved from AP_i in channel i to AP_k in channel k, without losing connectivity. From the STA perspective, it is still connected to the same VAP.

5.4.3 Protocol Details

In this section, we discuss some of the finer aspects of mVAP.

Uniqueness of each VAP

Each client of the Voice WLAN is associated with its own unique VAP. Indeed, a client station maintains its connection with an AP as long as it successfully receives beacons from this AP's BSSID, which means that a VAP has to keep its BSSID when moving from one AP to another. Furthermore, since we do not want this AP and channel switch to affect other clients, the VAP's BSSID needs to be unique and dedicated to a particular client station.

A VAP's uniqueness can be achieved by basing its BSSID on its client's MAC address and flipping the first bit, for example:

Client MAC address		VAP BSSID
`00:11:22:33:44:55`	→	`80:11:22:33:44:55`
`aa:bb:cc:dd:ee:ff`	→	`2a:bb:cc:dd:ee:ff`

Channel Switch Announcement

A VAP uses the CSA element, as described in Section 3.3.1, to advertise that it is switching to a new channel. Since each station associates with its own VAP and receives dedicated beacons, the channel switch only applies to the station that effectively changes APs. The MadWifi driver for example implements the CSA element, so stations can use this solution without client-side modifications.

The CSA element is sent in the beacons only when the new AP is chosen (step 8 in Figure 5.3). In our implementation, the old AP sends three consecutive beacons, with an interval of 100 ms between them, decrementing the value of the Channel Switch Count in each beacon. When this value reaches zero, 300 ms after the first CSA announcement, the station switches to the new AP and channel, and the old AP deletes the station from its associated client list and stops sending VAP beacons for the station.

Inter-AP Messages

Inter-AP communications (steps 2 to 7 in Figure 5.3) take place over the DS, as an Ethernet wired network commonly interconnects APs in current deployments. Messages are exchanged over reliable TCP connections established between APs, and contain the following information, also depicted in Figure 5.4:

- **Scan Request**: Station's MAC address, Station's IP address, BSSID, Station channel (where the APs' *monitor interfaces* will listen for Station's traffic).
- **Scan Response**: Station's MAC address, Station's IP address, Station Received Signal Strength Indicator (the signal strength of Station's frames that reached the AP), new AP channel (the channel in which the responding AP's *AP interface* operates).
- **Station Move**: Station's MAC address, Station's IP address (because the Station will keep its IP address when changing APs, the new AP needs this information to update the network's bridging tables via gratuitous ARP messages), CSA count, Beacon interval (these last two elements give the new AP a hint about when to expect the Station's channel switch).

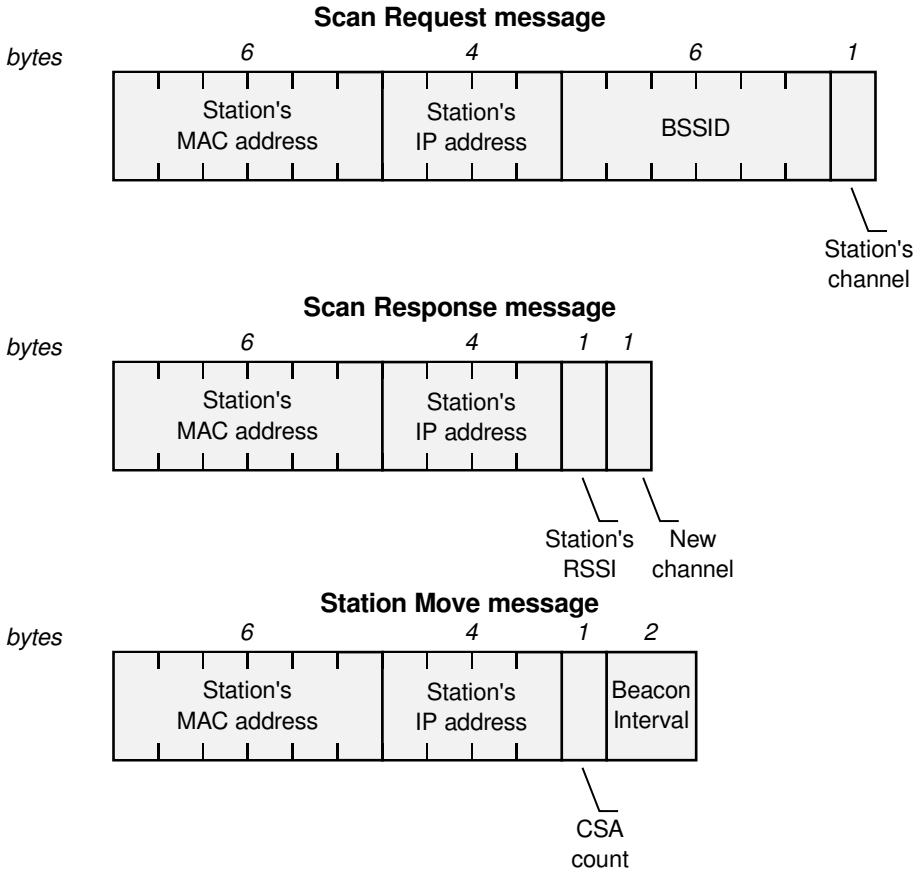


Figure 5.4: Inter-AP messages for the mVAP protocol.

Use of 2 Radios

When a client station moves away from its current AP, the AP starts the process of searching for a new AP among its neighbors. Therefore, the neighbor APs need to listen for packets from the moving station in order to obtain its RSSI. Since the station itself is transmitting in a channel different from the neighbor APs' channels, these APs need

to switch channels in order to listen for the station's packets. As a result, if using only one wireless interface, the APs leave their own channels and clients unattended. If their clients send packets, they will be lost. Consequently, in accord with our *Smart AP* model, we use two wireless radios, the *AP interface* for the AP functionality (for both Voice and Default WLANs), and the *monitor interface* for monitoring the roaming VoIP clients in other channels.

5.5 PACMAP

In order to implement our mVAP solution, we developed a tool called PACMAP: the PACket MAniPulation framework. PACMAP is a framework for controlling and manipulating 802.11 frames. It is a user-space frame monitor and injector that allows for fast prototyping of modifications and customizations of the IEEE 802.11 MAC protocol (management and data functions) and upper-layer networking protocols.

Initially, PACMAP was conceived for prototyping with the Python scripting language, but due to poor performance measures during the evaluation of the implementation of VAP [38], it was rewritten entirely for this thesis, as a C library with the objective of code reusability and extensibility. Consequently, this new version is generic and can be used to test other ideas, for example, new handoff or AP selection techniques. Additionally, as the framework runs in user-space, code development is easier than in kernel space.

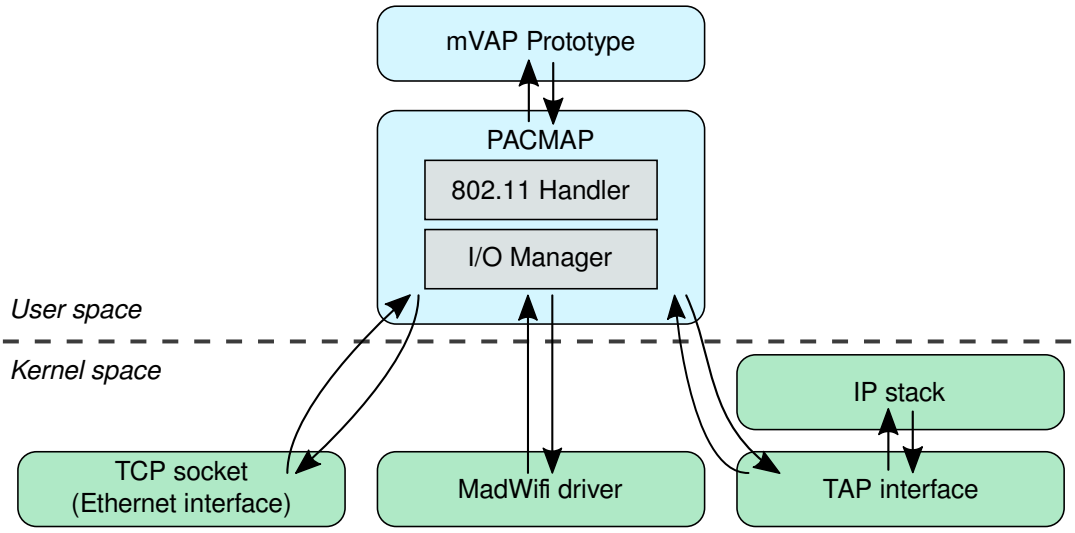


Figure 5.5: mVAP implementation using PACMAP.

Figure 5.5 shows a high-level description of how PACMAP is used to implement our mVAP prototype. A lower-level description of PACMAP's internal architecture and implementation will be presented in Sections 5.5.1 and 5.5.2, respectively.

5.5.1 Architecture

The PACMAP core is composed of the I/O manager and the 802.11 protocol handler, as shown in Figure 5.6. Incoming packets handled by PACMAP may come from the wireless or the TAP interface [106]. These packets are processed by the I/O manager and sent to the 802.11 protocol handler, which passes them to the appropriate prototype callback function.

The callback functions process these incoming packets, and might forward them or create new wireless frames. These outgoing packets are then passed to the PACMAP core, which in turn delivers them to the appropriate interface. For example, beacon frames are generated by the AP prototype code, then transmitted to PACMAP, and finally injected into the wireless card.

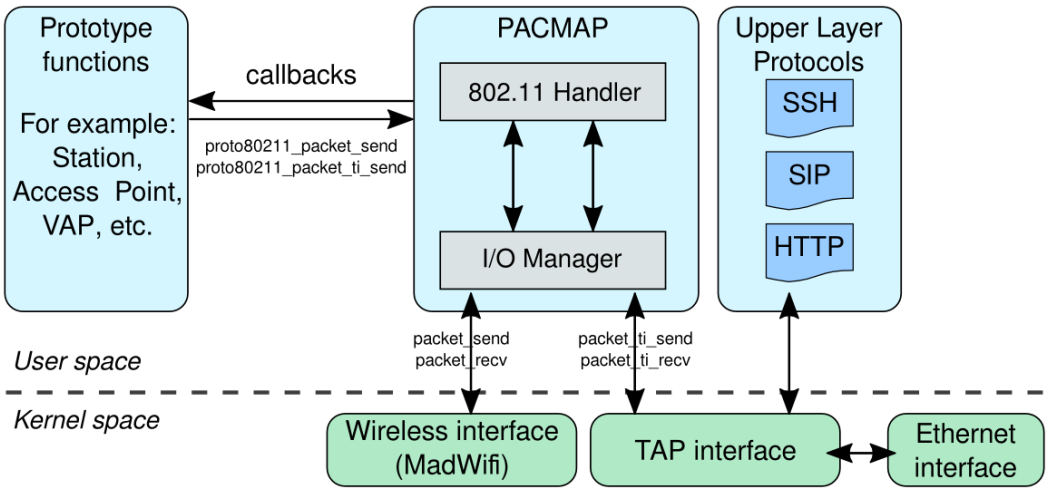


Figure 5.6: Low-level description of PACMAP.

I/O Manager

The I/O manager is an abstraction layer to the wireless and TAP device handling. It writes to and reads from the devices, configures their settings, manages sockets, and handles periodic functions. Wireless frame injection and I/O management is based on Aircrack-ng [8] code, a well-known tool usually used to recover 802.11 encryption keys.

802.11 Protocol Handler

The 802.11 protocol handler receives packets from the I/O manager, classifies them according to their type (for example, an authentication management frame), and executes the appropriate prototype callback function. It also sends the outgoing packets to the I/O manager, and can forge 802.11 header frames if needed.

5.5.2 Implementation

Event Handler

We have designed this new version of PACMAP as an event-driven C library. We use the libevent API [62] to execute a callback function when one of the following events occurs:

- A packet is received from the wireless interface.
- A packet is received from the TAP interface.
- A periodic function is activated, for example, to send beacons.
- A function timeouts, for example, when scanning a channel (*MinChannelTime*, *MaxChannelTime*).
- An Inter-AP message is received, since the AP is listening on a specific TCP port of the Ethernet interface.

Wireless Interface

We use Atheros-based wireless cards with the MadWifi driver. In monitor mode, the card listens to all packets and does not filter them. At the same time, MadWifi makes it possible to inject packets and send them over the wireless medium.

The wireless card firmware manages the 802.11 control frames such as ACK and RTS/CTS, as well as channel access methods, so the details of frame delivery and reception in the wireless medium are beyond the reach of PACMAP.

TAP Interface

A TAP interface [106] emulates an Ethernet device and handles Ethernet frames. PACMAP uses a TAP interface to inject the incoming wireless data packets into the kernel network stack for further processing.

Similarly, the packets sent by the kernel to the TAP interface, from upper-layer applications or the Ethernet interface, are received and processed by PACMAP.

Cross-layer Information

PACMAP can take advantage of cross-layer information, which is very useful in wireless networking. One important example is the signal strength that is measured by an AP or an associated client station, and that is used to trigger a handoff.

The MadWifi driver supports the radiotap header [83] and provides physical layer information such as channel frequency, bit rate, and RSSI. Thanks to the design of PACMAP, the use of cross-layer techniques between upper-layer networking protocols is also possible.

5.5.3 Prototyping

The simplicity of PACMAP lies in the abstraction layer provided by its 802.11 protocol handler, which interacts with the I/O manager, leaves out complex hardware details, and allows one to concentrate on the main functionality when implementing a prototype.

In order to modify the default behavior of MadWifi's monitor mode, a prototype consists of several callback functions executed by the 802.11 protocol handler upon receiving a packet. Examples of these functions are: processing an authorization request, handling a disassociation message, or selecting a new AP. It is also possible to create periodic functions such as beaconing, or timeout functions such as AP scanning.

PACMAP can use a configuration file, allowing different settings for a same prototype. Examples of configuration options are: the number of neighbor APs and their MAC and IP addresses, channel, beacon interval, and RSSI threshold.

PACMAP is available for download at `http://pacmap.ligforge.imag.fr/`

5.6 Evaluation

In this section, we first describe the setup for the experimental evaluation of our mVAP solution, and next present its performance results.

5.6.1 Methodology

We use two laptops acting as APs (AP_1 and AP_2), one laptop acting as a wireless mobile station M, and one desktop computer D connected to the wired network, as shown in Figure 5.7. All computers run Ubuntu 9.10. AP_1 and AP_2 have two wireless cards: one for the AP functionality, a D-Link DWL-AG660 wireless card running the MadWifi driver; and another one for the monitor functionality, an Intel PRO/Wireless 3945ABG card running the ipw2200 driver. The mobile station M has one D-Link DWL-AG660 wireless card and runs the MadWifi driver. The version of the MadWifi driver is 0.9.4. The APs run PACMAP with the mVAP implementation, and both of them use 802.11g: AP_1 listens to channel 1 and AP_2 listens to channel 6. The mobile station uses the standard MadWifi driver configured in station mode.

During the experiment, station M associates with AP_1 (listening to channel 1) and then moves toward AP_2 (listening to channel 6) which becomes its new de facto AP. Finally, station M returns to AP_1.

For performance evaluation, we use a Constant Bit Rate (CBR) UDP stream to mimic the VoIP traffic generated by different voice codecs, shown in Table 5.1. Packets are sent from station M to computer D. We generate the UDP traffic with the tool *iperf* [50], and final UDP payload size contains the voice payload size of the codec, plus 12 bytes of the Real-time Transport Protocol (RTP) header.

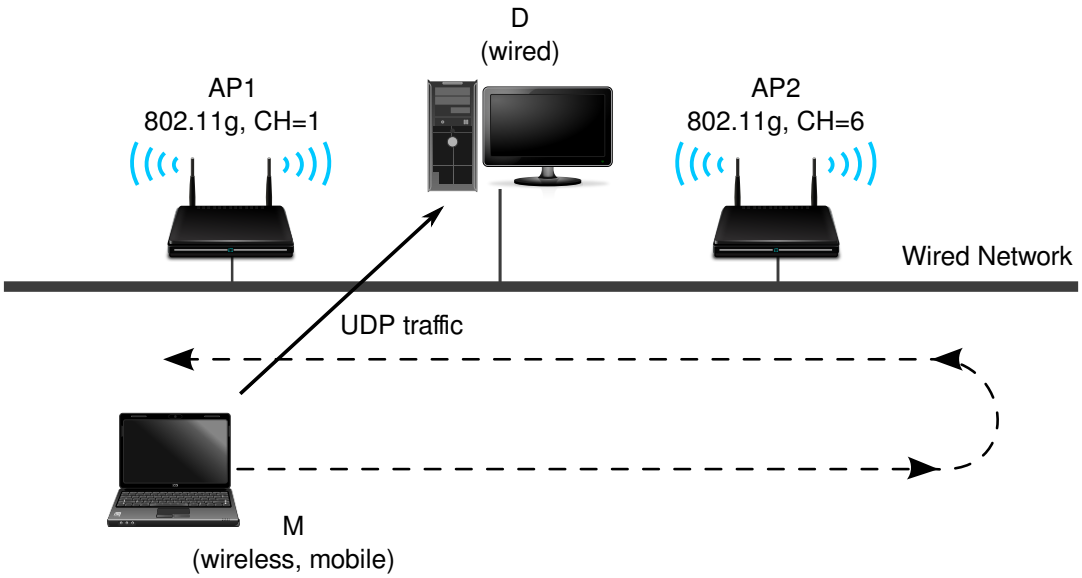


Figure 5.7: Experimental setup.

Codec	Bit Rate	Voice Payload Size	Interval
G.729	8 Kb/s	20 Bytes	20 ms
G.728	16 Kb/s	60 Bytes	30 ms
G.711	64 Kb/s	160 Bytes	20 ms

Table 5.1: Voice codecs used in the evaluation.

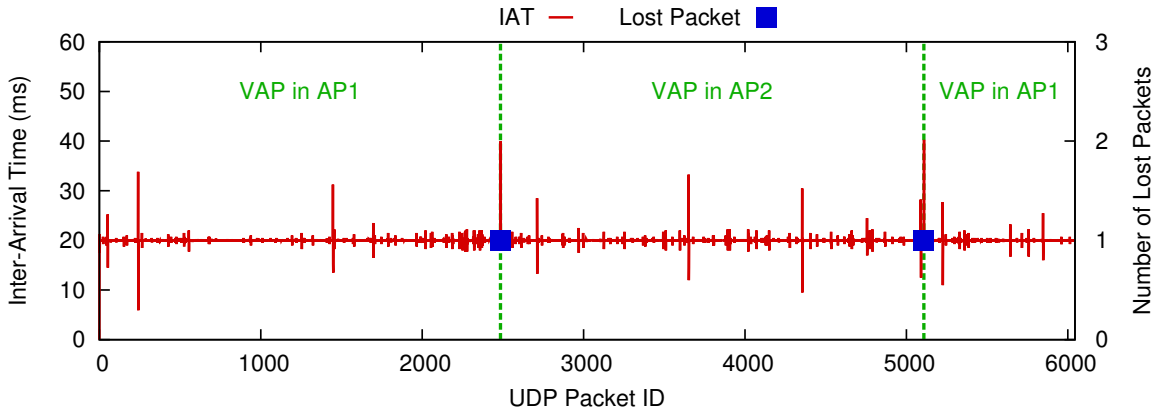
5.6.2 Results

We measure the packet Inter-Arrival Time (IAT), defined as the difference between the arrival times (AT) of the n$^{\text{th}}$ packet and the (n-1)$^{\text{th}}$ packet:

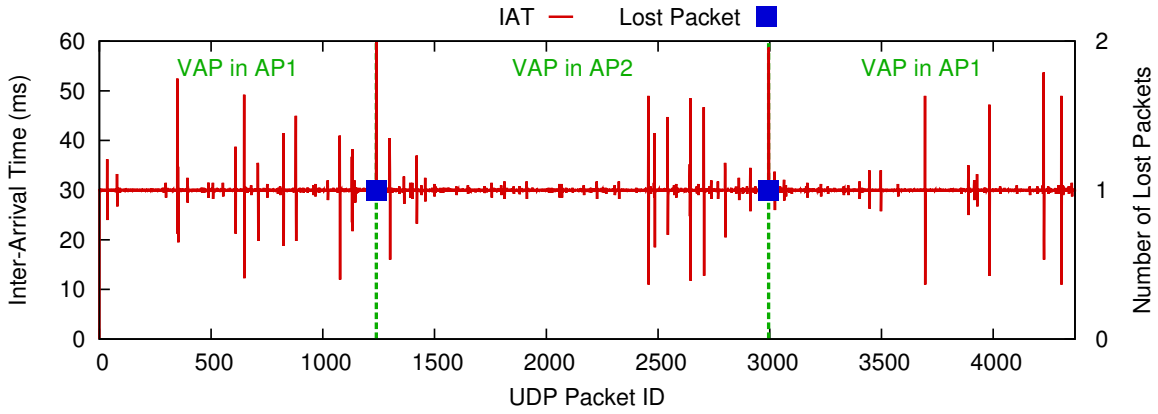
$$IAT(n) = AT(n) - AT(n-1) \tag{5.1}$$

In Figure 5.8, we observe that in each experiment the IAT between packets is never greater than 60 ms: the mVAP solution stays well below the maximum delay of 150 ms required for quality VoIP communication. Moreover, there is no disruption in the communication and no packet loss when the station moves from one AP to the other, except one single packet because of ARP tables updates.

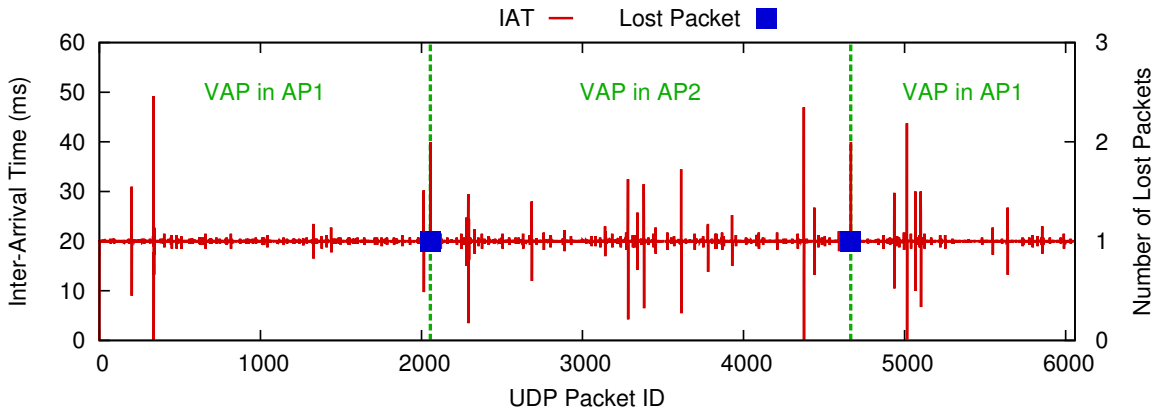
In Figure 5.9, we present the empirical Cumulative Distributive Function of the IAT values for each codec. Most of the values are clustered around the expected intervals of 20 ms, 30 ms and 20 ms for codecs G.729, G.728 and G.711, respectively. The mean, 90$^{\text{th}}$ percentile, and standard deviation of the IAT distributions are shown in Table 5.2.



(a) G.729 (8 Kb/s).



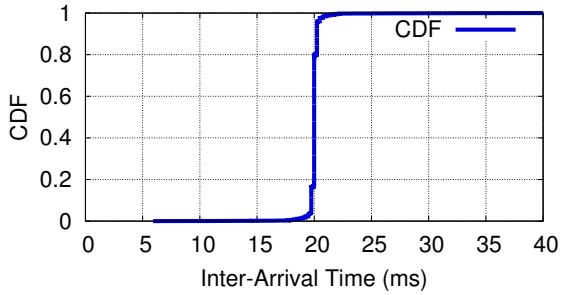
(b) G.728 (16 Kb/s).



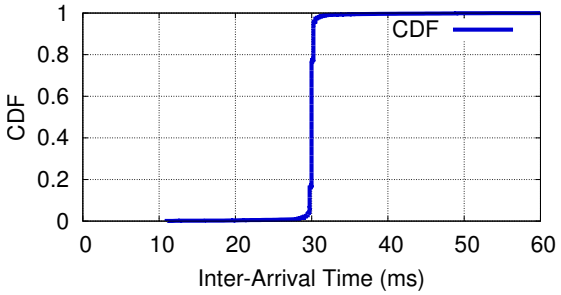
(c) G.711 (64 Kb/s).

Figure 5.8: Inter-Arrival Time (IAT) between UDP packets, for each codec.

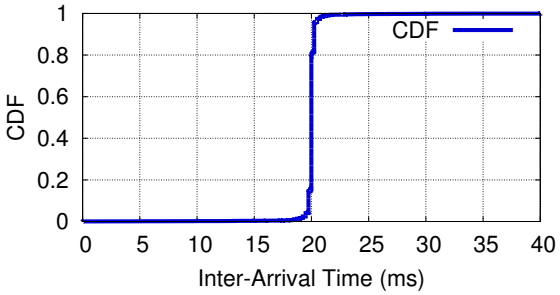
From these results, we conclude that mVAP handles the transition from one AP to another without disrupting the current communications and offers exceptional handoff performance. Furthermore, we confirm that our mVAP solution can be implemented in off-the-shelf hardware and can be deployed without any modification on the client side.



(a) G.729 (8 Kb/s).



(b) G.728 (16 Kb/s).



(c) G.711 (64 Kb/s).

Figure 5.9: Empirical CDF of the IAT values, for each codec.

Codec	Mean	90^{th} Percentile	Std. Deviation
G.729	20.01 ms	20.24 ms	0.72 ms
G.728	30.01 ms	30.23 ms	1.63 ms
G.711	20.01 ms	20.24 ms	1.16 ms

Table 5.2: Descriptors of the IAT distributions, for each codec.

5.7 Future Work

When we described the mVAP protocol in Sections 5.4.2 and 5.4.3, we intentionally omitted some details that are currently implemented in a very straightforward way. However, they constitute an interesting area open for enhancements and future work.

Metrics

In our mVAP prototype, the metrics to trigger a Scan Request (step 1 in Figure 5.3) and to choose a new AP (steps 3 to 5) are simply the instantaneous RSSI of the client station. The conditions of the wireless medium are highly variable, so using just one single value as a trigger might not accurately represent the movement of the station. Historical information and long-term trends, as suggested in [68], and other values such as AP loads, can improve these decisions.

List of Neighbor APs

In order to send the Scan Request messages (step 2), each AP needs the list of its neighbor APs. This list is manually provided by an administrator, which can be a laborious task in a WLAN with many APs. But this list could also be created by a neighbor discovery algorithm: during a preliminary learning phase, the APs monitor all the channels and use the information found in management frames (such as beacons and probe responses) to build their neighbor lists.

Inter-AP Communication

In steps 2 to 7, APs could use the CAPWAP protocol [21] and exchange messages through a Datagram Transport Layer Security (DTLS) tunnel. This secure connection between APs would also allow a client's security context to be transferred, when passing its VAP from one AP to another.

VAP Beacons

- When many VoIP clients are active simultaneously, the transmissions of VAP beacons increase, reducing the available bandwidth. In order to limit the number of beacons, APs could create VAPs as a service, only when they detect VoIP phones (for example, by distinguishing such a phone's MAC address from normal devices such as laptops). Another option is to decrease the beacon frequency, by using a larger beacon interval, which is 1024 μs $\approx$ 100 ms by default.

- The beacon interval is advertised in the beacon frame. We observed in the MadWifi driver's code that client stations update the value of this interval upon receiving a beacon. In order to speed up the channel switch when a station moves toward a new AP, the beacons with the CSA element could be sent with a smaller beacon interval (e.g. 50 ms).

Quality of Service

Although not implemented in this mVAP version, we could provide higher priority to VAP frames in the Voice network, over common WLAN frames in the Default network, by using two separate queues, depicted in Figure 5.10. Each queue has different channel access parameters, such as AIFS, CWmin, and CWmax.

The mechanism for channel access is similar to 802.11e EDCA (described in Section 3.4.1): two frames ready to be sent, one in each queue, obtain the same time slot when executing the contention algorithm. The Voice frame is allowed to be transmitted, while the Default frame starts a backoff, as if a collision had occurred. This mechanism is completely transparent to the client stations.

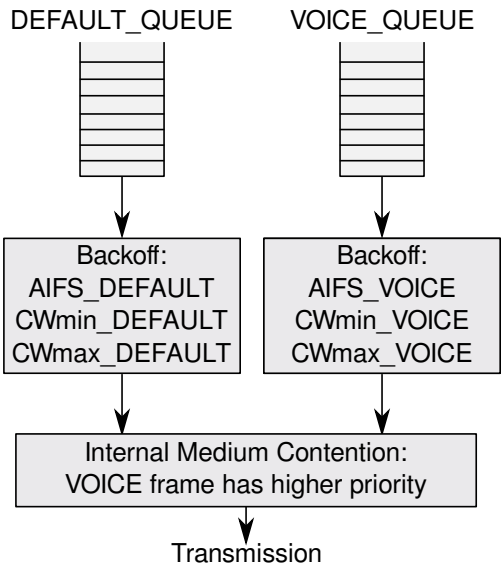


Figure 5.10: Two transmission queues: the Voice queue has higher priority than the Default queue.

5.8 Summary

We presented the first contribution of this thesis: Multichannel Virtual Access Points, a transparent mobility solution for VoIP services in Enterprise WLANs. With this solution, clients can move from one AP to another without disrupting their current communications. The advantages of mVAP include very low latency of handoffs, required for multimedia applications such as VoIP, and no client-side modifications at all, in accordance with our *Smart AP* model.

We also presented PACMAP, our flexible framework for developing IEEE 802.11 prototypes, in user-space, using inexpensive hardware. The new design of PACMAP allows for easy experiments with MAC layer (and above) prototypes for 802.11 wireless networks, and was used to implement our mVAP solution.

We evaluated this implementation with three different types of voice codecs, and the results show that the delay between packets does not vary when moving from one AP to another, and current communications are not disrupted.

The promising results of this network-based mobility technique encouraged us to analyze the deployment of such a solution in uncoordinated environments, very different from the Enterprise WLANs we just worked with. Therefore, in the next chapter, we will focus on the characteristics of mobile connectivity in a wireless network composed of the numerous but heterogeneous APs present in an urban area.

Chapter 6

CITYWIDE MOBILE INTERNET ACCESS

Contents

6.1 Introduction

In this chapter, we present the second contribution of our thesis: we investigate the feasibility of exploiting the WiFi coverage in urban areas for mobile Internet access, and the type of applications that can benefit from this Internet access provided by already deployed APs.

Nowadays, most smartphones and other mobile handsets are WiFi-enabled. Moreover, mobile Internet data traffic is expected to grow significantly in the next few years. WiFi is an interesting alternative to cellular networks, as it is a widespread wireless technology, provides high data rates, and has a low deployment cost.

Working with the data collected by smartphones across a city, we analyze the characteristics of WiFi coverage and connectivity for mobile users, using various AP settings and mobility speeds. These results allow us to study different applications that could be supported and the challenges faced to create a citywide WiFi network, given the current infrastructure composed of residential WiFi APs and hotspots.

The results presented in this chapter were published in the proceedings of the first workshop on Urban networking (UrbaNe '12), in December 2012 [15].

6.2 Motivation

With the proliferation of mobile Internet-enabled devices, Internet connection is expected by users anywhere and anytime. Indeed, many forecasts predict an exponential growth of mobile data traffic [79]. Cellular broadband networks are therefore facing the problems of traffic congestion and network capacity. But improvements to these wireless networks are expensive, and new technologies such as 4G still have issues with their service performance.

WiFi networks are a viable solution to reduce the use of 3G networks, offering high bandwidth and a cheap infrastructure [58]. In fact, mobile operators have already started to use WiFi for data offloading. Thus, dense deployment of WiFi APs could ensure Internet connectivity, falling back to 3G where service cannot be offered.

WiFi APs can be found almost everywhere nowadays: municipal wireless networks, cafés, hotels, airports, and private environments at home or work. However, only a fraction of all these APs are open for association to any user. In some other cases, such as residential APs, their owners are members of "community networks" and share their Internet bandwidth with other members (e.g., FON [30], FreeWifi [31], CableWifi [20]).

WiFi APs in a city are generally unmanaged with a default configuration. Moreover, they are deployed indoors and in an unplanned manner, which can cause poor reception, interference between neighbor APs, and result in low throughput. They connect to the Internet over a broadband access, such as DSL, which provides high data rates.

In this chapter, we consider the use of already deployed WiFi APs to provide urban mobile Internet access. If the coverage is sufficiently dense, users moving with different speeds may profit from continuous WiFi connectivity. The questions are whether such an architecture is feasible, on what parameters its performance depends, and what applications can benefit from its services. We investigate these issues by simulating connectivity of mobile users in a city, based on detailed traces provided by the Nokia Mobile Data Challenge [67].

Our contributions can be summarized as:

- We analyze the data collected by smartphone users during more than one year, mainly in the Swiss city of Lausanne, and provided by the Nokia Mobile Data Challenge [67]. We first identify the distribution of the WiFi APs already deployed in the city, and properties such as channel assignment, link quality, and authentication mode. We also study the patterns of the users' paths to derive a simple mobility model.

- We evaluate the WiFi coverage and the characteristics of the connectivity of mobile users in the central urban area of Lausanne. We simulate various scenarios using different AP ranges, security and handoff parameters, but also user speeds and mobility patterns.

- Finally, we discuss the open issues for a real deployment of such a citywide WiFi network, given the results above. We explore the applications that can benefit from Internet access provided by WiFi-based connectivity, such as sensor-data collection and location-based services.

6.3 Related Work

Several works have considered the deployment and usage of WiFi APs in cities. We present an overview of the different aspects of the subject:

AP Characterization

The first study examined statistics of over 5 million APs collected through wardriving. It characterized the default settings (e.g. SSID), location, and density of APs found in several metropolitan cities [52]. Similarly, other works quantified the impact of interference on end-client performance [9], analyzed the coverage and duration of connectivity, and measured the performance of TCP uploads [19].

Large-scale WiFi Networks

Inspired by the popularity of WiFi technology, municipalities, non-profit and private entities have deployed free WiFi hotspots in cities. Examples of such urban WiFi networks include MIT Roofnet, Google WiFi Network, Madison MadMesh. Despite the efforts, these networks suffer mainly from packet loss [6] and coverage holes [11]; they need more investment in infrastructure, and are far from having a good performance.

WiFi Offloading

Prior works have studied mobile data offloading as an alternative to cellular network upgrades, as WiFi hotspots are cheaper and easier to deploy. Balasubramanian et al. studied the availability of WiFi connectivity and its performance, and proposed a predictor of offload capability [12]. Lee et al. analyzed the traces of 100 mobile phones during two weeks and a half to predict the duration of data transfers based on mobility patterns [58].

6.4 Data Analysis

In this section, we describe the data used later in our simulations. We characterize essential AP properties and detail our corrections to the APs' positions. Finally, we present a mobility pattern generated from the GPS traces of the mobile users.

6.4.1 Data Description

We obtained the data from the Mobile Data Challenge organized by Nokia [67]. It was collected by the smartphones of 38 participants during more than one year. We used the GPS, WLAN, and WLAN location datasets [57]. The GPS traces contain the GPS information of the mobile phones. Because of energy constraints, GPS coordinates were sampled with a period of 10 seconds, mostly during outdoor movements.

The WLAN traces contain the APs discovered during the WiFi scans performed by the phones, and the WLAN location traces contain the GPS coordinates of some of the APs; these traces were sampled with a period between 60 and 900 seconds. Sensitive WLAN fields, such as the MAC address and the SSID, were anonymized to ensure the privacy of the participants.

6.4.2 Characterization of the APs

From the WLAN traces, we extracted the APs operating in infrastructure mode and found nearly 127,000 unique APs distributed across Switzerland. The following information is present in the dataset: timestamp, anonymized MAC address and SSID, signal strength, channel, encryption type, operational mode.

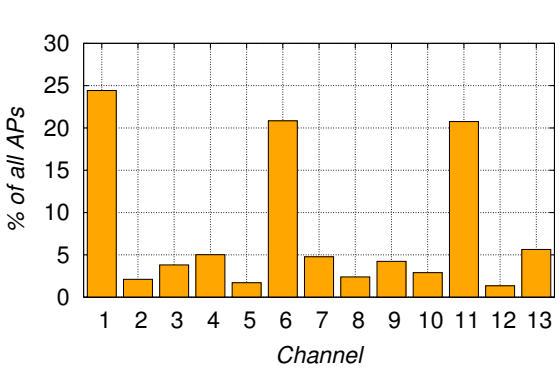
When inspecting the traces, we discovered that a few APs changed their settings over time, such as channel number and security type. Consequently, for the following plots, we considered all the different values of channel number and security type that an AP might have had during the data collection.

We analyzed the following characteristics:

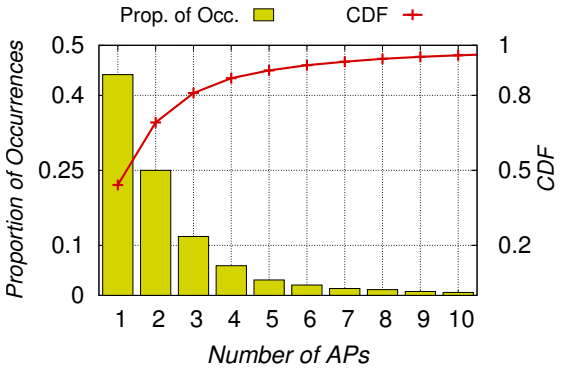
Channels

The smartphones performed their WiFi scans in the 2.4 GHz band, so the APs they discovered listen to channels between 1 and 13. In Figure 6.1a, we observe that 65% of the APs operate in channels 1, 6, and 11; indeed, the use of orthogonal channels decreases the interference among neighboring APs.

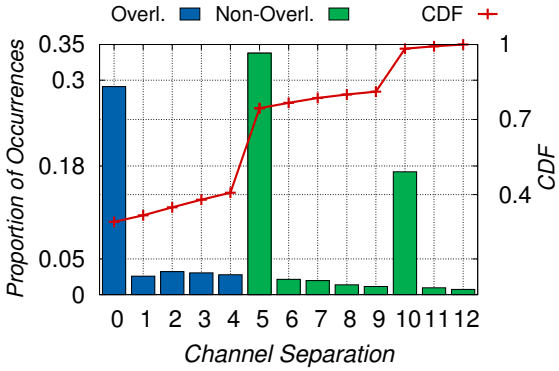
We examined the APs' channel assignment in detail. Figure 6.1b represents the number of APs detected per scan: 45% of the scans found only 1 AP, and nearly 42% found between 2 and 4 APs. When analyzing the scans that detected 2 or more APs, we calculated the channel separation between all pairs of APs found during the scan.



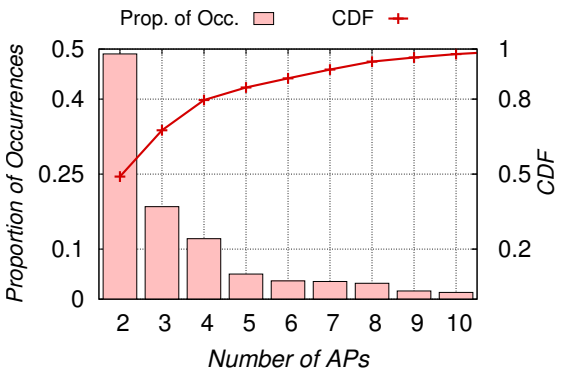
(a) Channel distribution.



(b) Number of APs detected per scan.



(c) Channel separation between the APs detected per scan.



(d) Number of co-channel APs detected per scan.

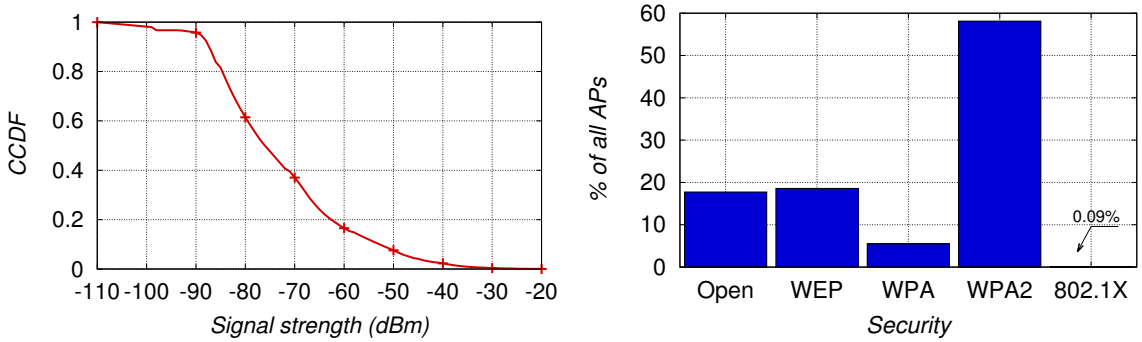
Figure 6.1: APs' channel distribution and channel separation.

Figure 6.1c depicts the resulting channel separation distribution: on the one hand, 40% of the APs have a channel separation of less than 5 channels, which is not sufficient to guarantee a wireless medium free of interference. Worse yet, 30% of the APs use the same channel, creating important co-channel interference. On the other hand, 60% of the APs have enough channel separation (5 or more), and 35% and 17% of the APs have a channel separation of 5 and 10, respectively, which reflects the use of orthogonal channels (channels 1-6 and 6-11, and channels 1-11).

Figure 6.1d portrays the number of APs that share a same channel, as detected by a single scanning operation. Nearly 50% of the scans found at least 2 APs operating in the same channel, and 40% found 3 or more. These results show that channel interference is a serious problem in such dense urban deployments.

Link Quality

(Figure 6.2a) 70% of the records in the WLAN traces correspond to APs that were detected with a signal stronger than -85 dBm, providing a data rate of at least 11 Mb/s.



(a) Signal strength CCDF. **(b)** Security distribution.

Figure 6.2: AP characteristics: (a) Link quality and (b) Authentication.

Actually, nearly 40% of the records indicate a signal stronger than -70 dBm, a common value for data rates of 54 Mb/s[1].

Authentication

(Figure 6.2b) Less than 20% of the APs are open for association (they do not require any authentication). Open APs already share their bandwidth and provide free Internet access to guests. Community networks, such as FreeWifi, appear as "Open" in the scan list but require users to authenticate through a web portal.

6.4.3 Geographic Distribution of the APs

The WLAN location traces contain the coordinates of nearly 2,500 APs. The following information is present in the dataset: timestamp, anonymized MAC address, longitude, latitude. The position of each AP appears several times in the traces, with coordinates that sometimes differ by a few meters. Therefore, we calculated their arithmetic mean in order to obtain a definitive position.

The coordinates of the APs are expressed in decimal degrees (for example, 45.2002 °, 5.7222 °) and while most of them have between 7 and 11 decimal places, some of them appear truncated, with less than 3 decimal places. We removed these truncated coordinates from the dataset, because they differ from the other coordinates by more than 1 km. The result of applying this heuristic can be seen in Figure 6.3: on the left map, many APs were erroneously positioned on an artificial straight line.

Finally, from the coordinates of the APs contained in the WLAN location traces (only 2,500 APs, less than 2% of all the APs discovered by the phones), we were able to extrapolate the coordinates of nearly 33,500 APs (26% of all the APs). We assumed that all the APs detected by a scan are located within a radius of 50 m from the phone that performed the scan (a common coverage range for indoor APs). Thus, we generated new coordinates (for

[1]Measured experimentally in `http://www.tp-link.com/en/products/details/?model=TL-WN721N#spec`

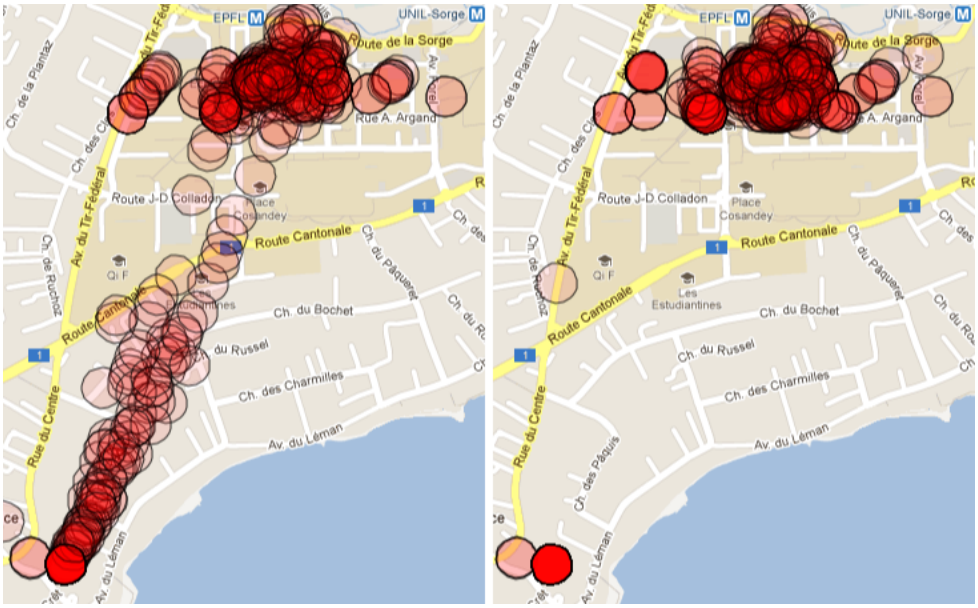


Figure 6.3: AP distribution: with (left) and without (right) the truncated coordinates.

the APs whose position was unknown) by appending five random digits to the first three digits of the original coordinates (of a nearby AP whose position is known), for example:

- original coordinates: [45.2002 °, 5.7222 °]
- new coordinates: [45.200***40053*** °, 5.722***20417*** °]
- distance: 22.3 m.

6.4.4 Mobility Model

The GPS traces contain the coordinates of the users' outdoor whereabouts. The following information is present in the dataset: timestamp, geolocation (altitude, longitude, latitude), speed, heading. Based on this information, we wanted to create a mobility model more realistic than the Random Walk model: we used a Markov chain [111] to describe, for a moving user, the probabilities of transition from one direction to another.

We therefore processed all the heading fields that appear in the GPS dataset. We first transformed these fields (angles in degrees) into eight different directions: north, north-east, east, south-east, south, south-west, west, north-west. We then used a chain of three directions to obtain fine-grained movements, for example:

$$(previous, current, next) = (south, south, east)$$

Finally, we computed the transition probabilities with the following formula:

$$\Pr(next|previous, current) = \frac{\sum \Pr(previous, current, next)}{\sum \Pr(previous, current)}$$

In more than 75% of the cases, the new direction of a user (*next*) corresponds to his two previous directions (*current* and *previous*). Figure 6.4 illustrates how this model (on the right) generates paths with longer straight segments than the Random Walk model (on the left).

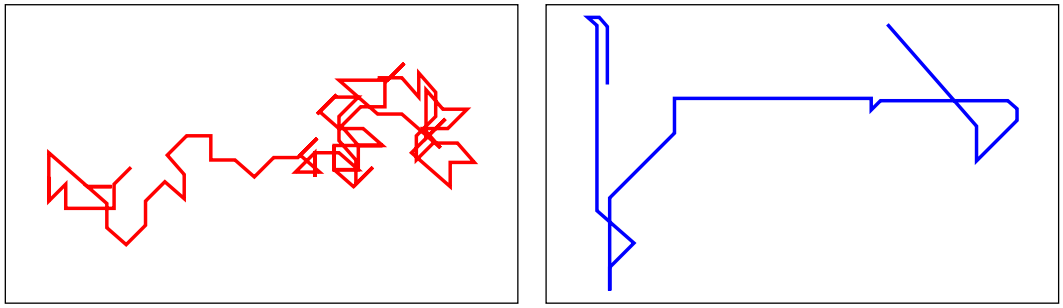


Figure 6.4: Paths generated with the Random Walk model (left) and the transition probabilities from the GPS traces (right).

6.5 Trace-based Evaluation

To study the feasibility of citywide WiFi networks, we simulated the connectivity of mobile urban users with the traces provided by the Nokia Mobile Data Challenge. We built a model of Lausanne, the city in the traces where most of the APs are located. We ran various simulations to analyze and understand the characteristics of the WiFi coverage and connectivity provided by the APs already deployed throughout the city.

6.5.1 Simulation Scenario

We focused on the central area of Lausanne, where the AP distribution is densest. We developed a simple simulator that moves users along random paths modeled with the Markov chain described above. The simulator computes the position of a mobile user with a granularity of 1 meter, and at each step checks if a handoff (or an association, if the user's device has lost WiFi connectivity) has to be performed. Therefore, it computes the distance between the device and its current AP and uses the Log-Distance Path Loss model [86] to obtain the current AP's signal strength. If needed, the simulator performs a handoff and switches from the current AP to another. It takes the following parameters into account:

AP Ranges

We assumed that all APs have omni-directional antennas, the prevalent and cheapest model encountered nowadays. We used three different coverage ranges for the APs: 20 m, 50 m, and 100 m. The two former values are common ranges for indoor APs, and the latter is a common range for outdoor APs. As 50 m is the most common range [9], we performed the simulations that involve different user speeds with this value.

AP Associations

As an ideal situation, and unless otherwise stated, we assumed that every AP in the city is available for association (we call this scenario "All APs"). But we also ran simulations using only the "Open APs" (those not protected by any security mechanism) from the traces (nearly 17% of all the APs).

Handoff Durations

A handoff is the process in which a client device searches for a new AP, authenticates and associates with this AP, and in this particular case (because every AP is independent and belongs to a different network) finally obtains an IP address. During the handoff, the client cannot send nor receive traffic, which has an important impact on the delay and packet loss that applications may have to tolerate.

We simplified the handoff process by modeling it as a fixed delay. We assumed the following values for the handoff duration: 0 s, 150 ms, 1 s, 2 s, and 5 s. The first value corresponds to an instantaneous handoff, as in the case of an efficient network-supported mobility management. The next value, 150 ms, is the maximum acceptable latency in end-to-end communications for VoIP applications. The latter values are commonly measured handoff durations.

Handoff Strategy

When a device is not associated with any AP, it chooses the AP with the strongest signal. When its current AP's signal strength (SS1) drops below a fixed threshold, the device attempts to associate with another AP whose signal strength (SS2) is greater than SS1. We used a -90 dBm threshold, a common limit for the lowest data rate (1 Mb/s).

User Speeds

We wanted to represent the different speeds of a mobile user, and therefore selected the following values: walking (1 m/s), by bicycle (5 m/s $\cong$ 18 km/h), by bus (11 m/s $\cong$ 40 km/h), and by car (20 m/s $\cong$ 70 km/h).

User Paths

We generated 10,000 different user paths with our Markov-chain mobility model. Each path is 3,600 meters long and made of 50-meter segments. We applied the transition probabilities after each segment, in order to change the user's direction. Finally, we centered the paths on the downtown area of Lausanne.

6.5.2 Results

We define the following performance indicators:

- **Temporal Coverage:** Considering a user's path as a whole, this is the total amount of time during which the user moves in WiFi-covered areas [58].

- **Connection Duration:** A continuous portion of a user's path during which the user is associated with the same AP. The connection period stops when the user leaves this AP's coverage area.

- **Disconnection Duration:** A continuous portion of a user's path during which the user is out of the range of any AP. The disconnection period stops when the user is able to associate with an AP.

- **Internet Access Session:** A continuous portion of a user's path during which the user is able to associate with one or more APs, and is able to send and receive Internet traffic (i.e., he is not performing a handoff), until a disconnection occurs because of the loss of WiFi coverage. As mentioned before, the user's IP address might change (if he moves from one AP to another) during such an Internet access session.

First, in Figure 6.5, we compute the mean temporal coverage for different AP ranges, when using "All APs" or only the "Open APs". In the "All APs" scenario, the most common (indoor) AP range of 50 m averaged 70% of temporal coverage. With a 20-m range (indoor APs surrounded by obstacles that decrease their range) we obtained less than 55% of temporal coverage, and with a 100-m range (outdoor APs) we obtained more than 80% of temporal coverage. These two values (20 m and 100 m) are the minimum and maximum AP ranges, so 55% and 80% represent the minimum and maximum temporal coverage when all the APs are available for association.

In the "Open APs" scenario, the results are lower but quite similar; the decrease in temporal coverage is only about 10%, which means that the WiFi coverage is already dense even when only 17% of all the APs are open for association.

In Figure 6.6, we plot the Complementary Cumulative Distribution Function (CCDF) of the temporal coverage when using "All APs", in order to compare the WiFi coverage of our 10,000 simulated mobile users, for each AP range. As expected, the longer the range, the better the coverage.

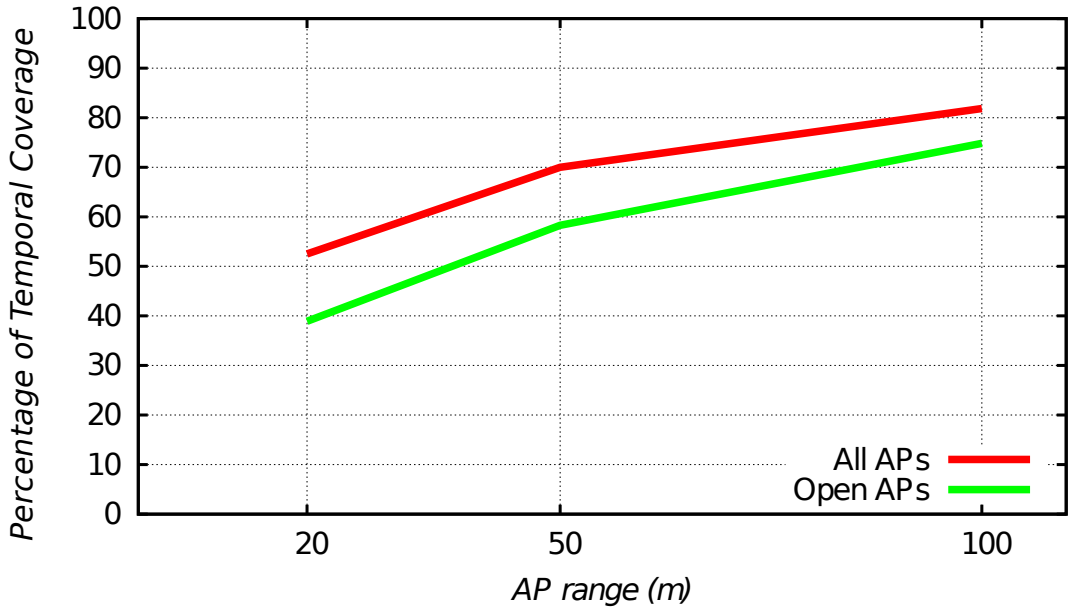


Figure 6.5: Mean temporal coverage (percentage of a complete user path), for different AP ranges, when using "All APs" or only "Open APs".

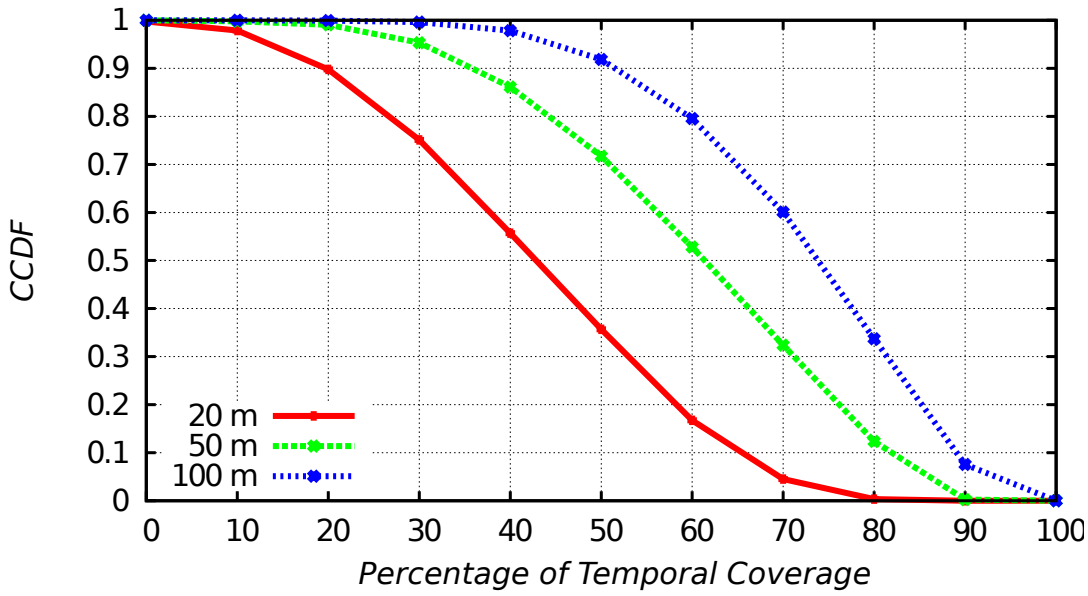


Figure 6.6: Complementary CDF of the temporal coverage (percentage of a complete user path), for different AP ranges, when using "All APs".

Next, we ran our simulations with an AP range of 50 m and different user speeds. In Figure 6.7, we show the mean duration of a connection with the same AP, when assuming a handoff delay of 0 s. If the user is walking, the mean duration of a single connection is 130 s (more than 2 minutes). However, faster speeds have shorter connections of less than 30 s.

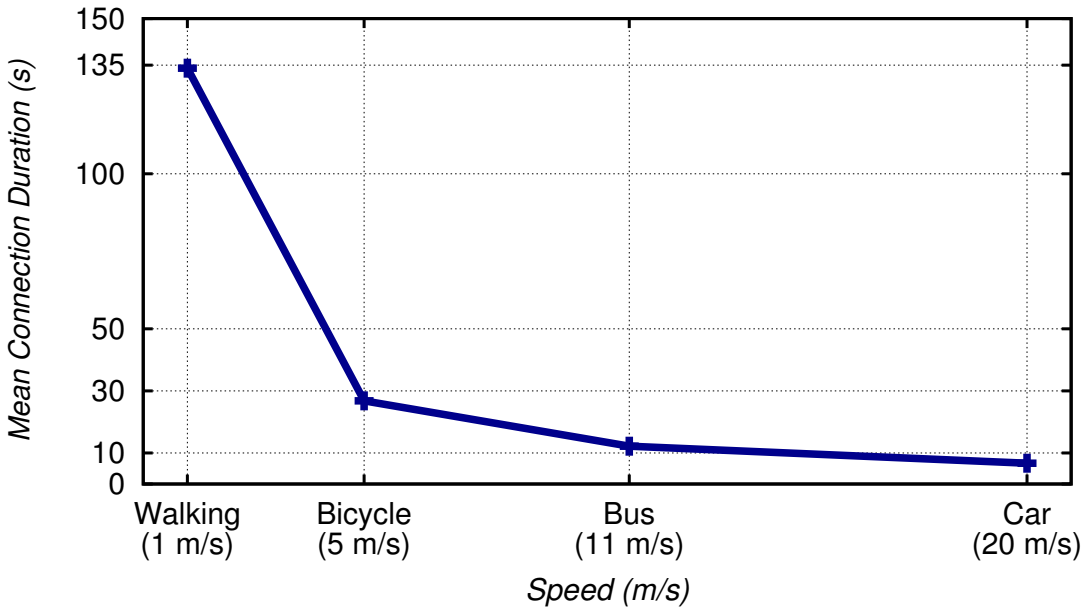


Figure 6.7: Mean connection duration with the same AP, for different user speeds.

In Figure 6.8, we present the mean duration of an Internet access session, for different handoff delays and user speeds. A walking user can experience an average session of 572 s (more than 9 minutes). At bicycle speed, the average session lasts 114 s (nearly 2 minutes), and this duration decreases to 52 s by bus, and 28 s by car.

We observe that if the user is walking, the duration of the handoff does not have as much impact as if traveling by car. Because the handoff duration does not depend on the user's speed, the short sessions by car are dramatically impacted by this delay: the average Internet access session decreases from 28 s to 8 s when the handoff duration increases. In contrast, the average session of a walking user merely decreases from 572 s to 551 s.

In Figure 6.9, we present the total Internet access session, during a complete user's path around the city. Again, a longer handoff delay has almost no impact on the total Internet access session of a walking user; but by car, the duration of the total Internet access session decreases dramatically from 70% to 20%. As mentioned before, longer handoff delays reduce the periods of useful connectivity.

Finally, Figure 6.10 and Figure 6.11 present the overall disconnection results. The first figure shows the mean disconnection duration. On the one hand, a walking user suffers long periods without WiFi connectivity (almost 3 minutes). On the other hand, faster users experience shorter disconnections.

The second figure shows the Complementary CDF of the duration of all the disconnections that occurred during our 10,000 simulations, for each user speed. For walking users, 40% of the disconnections lasted more than 100 s (nearly 2 minutes).

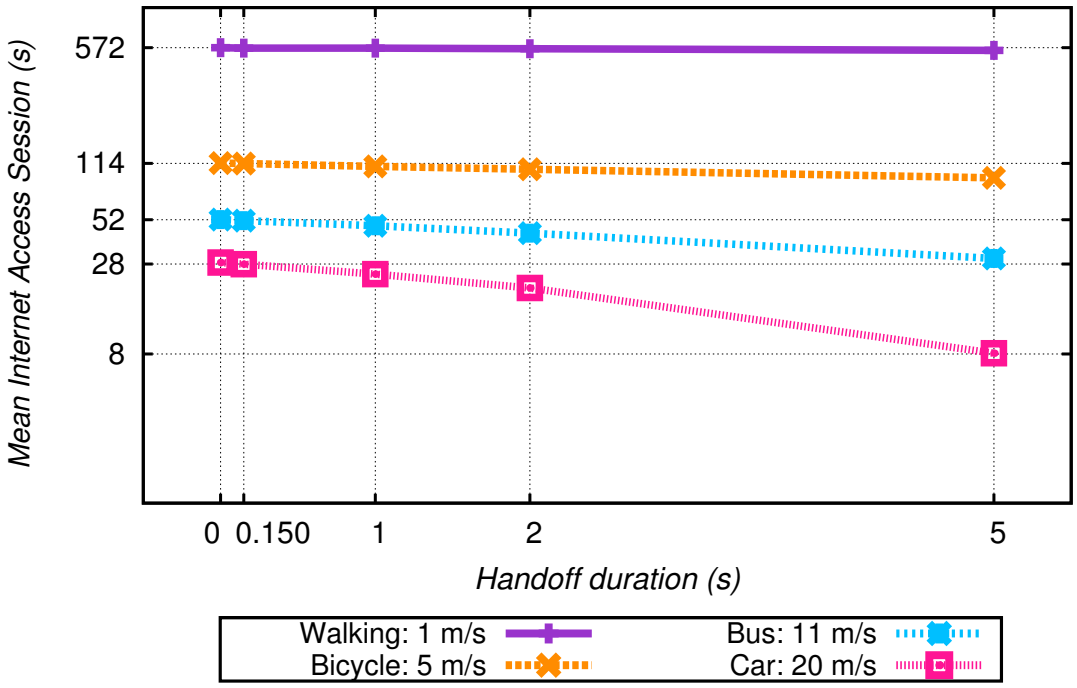


Figure 6.8: Mean Internet access session duration (logarithmic scale), for different handoff durations and user speeds.

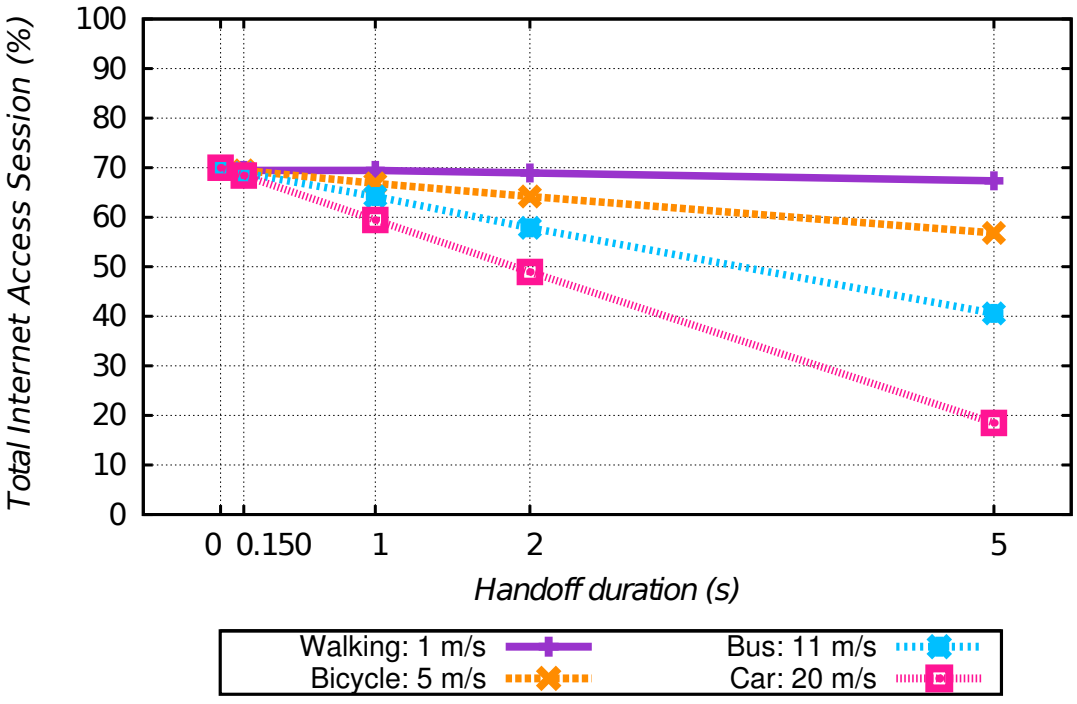


Figure 6.9: Total Internet access session (expressed as the mean percentage of a complete user path), for different handoff durations and user speeds.

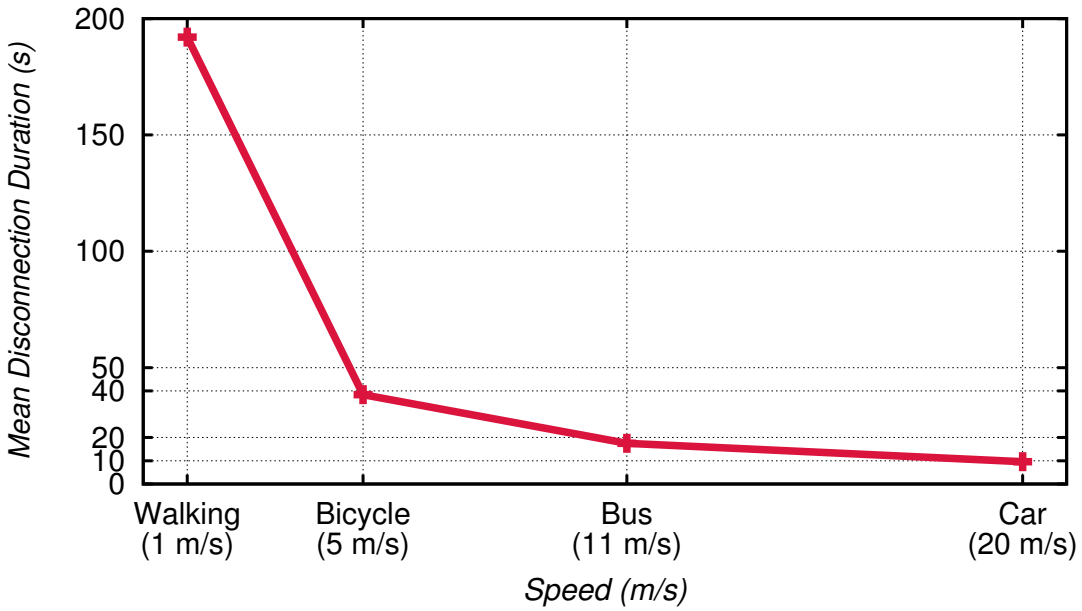


Figure 6.10: Mean disconnection duration, for different user speeds.

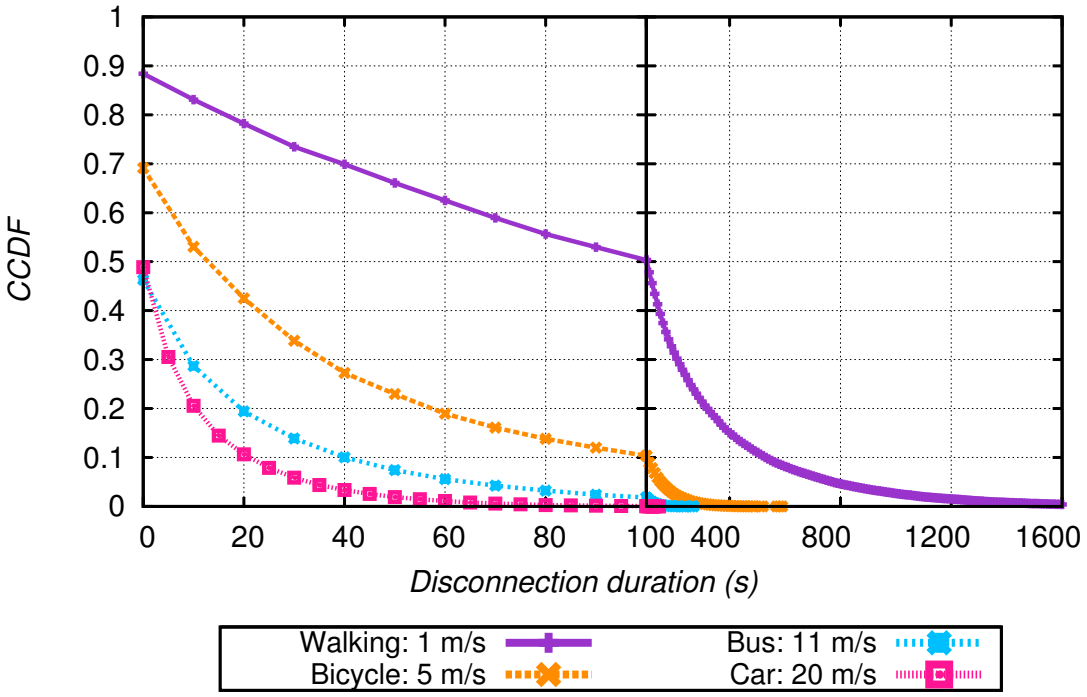


Figure 6.11: Complementary CDF of the disconnection duration, for different user speeds.

6.6 Discussion

The previous results, obtained from trace-based simulations, confirm our initial conjecture: WiFi APs in urban areas are so widely deployed that we could provide a seamless mobile Internet access based on this dense WiFi coverage. Indeed, a walking user can expect to be Internet-connected 70% of the time via such a citywide network. Unplanned placement

of APs may lead to the existence of areas that lack coverage, but short disconnections may be tolerated if the duration of the connections is sufficient.

The applications that could be supported in a citywide WiFi network, without any modification of the 802.11 standard, are delay-tolerant applications, that can either wait between connections or alternate between a 3G and a WiFi link [12, 58]. Short, but frequent connectivity can be used for opportunistic sensing. For instance, traffic and road conditions can be shared by mobile phones and disseminated to vehicles [76]. Thanks to the large coverage, location-based services can use the numerous APs for positioning, even indoors, without using a GPS.

When moving among APs, users should only experience very short handoff delays. Different strategies can be implemented to decrease this value: we can use different handoff and association algorithms (for example, multiple simultaneous associations [96]) or manage mobility in the network, which results in instantaneous handoffs [14]. With the handoff technique proposed in the previous chapter, the APs are aware of nearby clients by scanning all the available channels with their *monitor interface*, thus anticipating the handoff and reducing the delay.

Currently, there is no homogeneous method of authentication, the selection of APs has to be done manually, and mobility (session transfer) is inexistent. Nevertheless, efforts in this direction are under way: the Hotspot 2.0 and the Next Generation Hotspot initiatives, based on the IEEE 802.11u amendment. These specifications allow mobile devices to log into WiFi networks in a seamless way. Certainly, a uniform authentication and connection mechanism among Internet providers would help these mobile devices to profit from the extensive WiFi coverage and maintain their current communications.

6.7 Summary

The questions we answered in this chapter are: Can we use the already deployed WiFi APs for a citywide mobile Internet access? What are the characteristics of these WiFi connections? What type of applications can benefit from this wireless technology? And finally, what are the challenges to face if we want to take advantage of this infrastructure?

We studied the data originated from the activity of smartphones around the Swiss city of Lausanne. We analyzed the geographic distribution of the city's APs and their characteristics. We ran several simulations, varying mobile user speeds and AP properties such as range, association, and handoff duration. We measured the WiFi coverage, Internet access session, connection and disconnection durations.

The results showed that the existing WiFi coverage is large and the connectivity it offers can be exploited. We proposed several applications that could be used in such a citywide wireless network, and the challenges that should be tackled in order to run these applications in an effective way.

This study has also highlighted the fact that channel assignment is not a trivial matter in these dense urban deployments, as detailed in Section 6.4.2 ("Characterization of the APs", subsection "Channels") and depicted in Figure 6.1. This problem motivated us to investigate the performance and optimization of WLANs on a smaller scale ("Apartment" or "Home" WLANs), in the next chapter.

Chapter 7

TRAFFIC-AWARE CHANNEL SELECTION FOR HOME WLANS

Contents

7.1 Introduction

Home WLANs are increasingly popular, as more electronic devices have the ability to use the 802.11 technology. Because of the dense deployment of these WLANs in residential zones, neighboring networks have to share the wireless spectrum. This creates interference and reduces the available channel capacity. Careful channel selection is therefore required in order to provide a better performance. Moreover, we need a new adaptive technique that takes into account the characteristics specific to Home WLANs: they are uncoordinated networks, they generally do not have an administrator, and their traffic demands vary

through the hours and days. Indeed, home networks typically generate more traffic in the evenings and on weekends, while office networks are more active during the week days.

In this chapter, we present the third and last contribution of our thesis: an adaptive traffic-aware channel selection algorithm for Home WLANs. The chapter is organized as follows: we first explain the problem of interference and frequency allocation for Home WLANs in Section 7.2. We review the existing solutions for uncoordinated and traffic-aware channel assignment in Section 7.3. Then, in Section 7.4, we detail the design and development of our traffic-aware channel selection mechanism. In Section 7.5, we describe the experimental evaluations of our technique, and present the results we obtained, which show an improvement of the overall performance. Finally, we summarize the outcomes of this chapter in Section 7.6.

7.2 Problem Statement

In the last few years, high-speed Internet Service Providers have equipped their clients with a broadband router (also known as residential gateway). Commonly, a router comes with a DNS cache, and functionalities such as NAT, routing, and firewall. It offers the possibility of sharing the Internet connection, and other resources such as printers and file servers, among several devices connected to the router through a wire. Eventually, router manufacturers included the WiFi technology in their products, giving their customers the opportunity to deploy a wireless network at home: the router is the AP, and the wireless clients are devices such as notebooks and game consoles, which associate with the AP.

Residential WiFi networks are referred to as *chaotic deployments* [9] and have two important characteristics: they are *unplanned*, because the AP is deployed without careful placement or channel assignment, and they are *unmanaged*, because their owners have little or no administration skills. In unplanned WLANs, the placement of the AP is made spontaneously and might depend on its proximity to telephone or wall sockets. In contrast, larger WLANs in enterprises or university campuses benefit from a planned deployment, where network administrators use WiFi planning tools in order to maximize coverage and minimize interference by choosing an optimal frequency allocation. Unmanaged WLANs do not have an administrator in charge of the configuration and troubleshooting of the network; as a result, these APs are generally set up with the default settings from their manufacturers.

In urban areas, there exists a dense deployment of Home WLANs, mainly originated from the growing popularity of WiFi technology and the simplicity of installing and running these networks. Although APs have a short range (50 m for indoor deployments [9]) and obstructions such as walls reduce their signal reach, adjacent WLANs are not isolated from each other. Interference from neighboring WLANs can negatively impact the performance of a wireless network. Therefore, it is important to choose an operating frequency for the AP that minimizes this interference.

Indeed, Broustis et al. [17] state that there are three aspects of WLANs that can improve the overall network capacity in dense deployments affected by interference: 1) intelligent channel assignment, 2) load-balancing of user associations across APs, and 3) adaptive power-control. For Home WLANs, the only option is channel assignment: with only one AP, it is not possible to load-balance clients, and reducing the AP's transmission power would decrease its coverage.

In the research literature, we find many approaches that solve the problem of channel assignment in wireless networks such as cellular networks, mesh networks, and managed WLANs. However, as Home WLANs differ significantly from the previously studied wireless networks, these solutions do not address the problem in a complete way.

A basic channel assignment technique that avoids interference altogether is the use of orthogonal channels. As detailed in Chapter 3, 802.11 operates in the 2.4 GHz and 5 GHz bands. These bands have several channels each, offering three and twelve orthogonal channels, respectively. We will focus on the 2.4 GHz band, since it is the most common for commercial wireless hardware. But because of the large number of WLANs sharing the same wireless spectrum, the three non-overlapping channels (1, 6, and 11) of the 2.4 GHz band do not suffice to guarantee a coverage free from interference, as several neighboring APs end up choosing the same channel, as discussed in Section 6.4.

The selection of partially overlapping channels can solve the co-channel interference problem: a few solutions [71, 75] show their advantage over channel assignments that use only orthogonal frequencies. Nevertheless, these algorithms are traffic-agnostic and non-adaptive, since they do not consider changes in the environment, where APs switch channels, new APs can appear, and the traffic loads of WLANs fluctuate, creating more or less interference.

Many channel assignment solutions have been designed for centralized networks, such as Enterprise WLANs. But Home WLANs are uncoordinated networks: they do not have a centralized controller and they do not communicate or exchange information with each other. Therefore, we need a decentralized solution for channel selection, where each AP makes a decision based on local measurements.

Hence, we design and develop a channel selection algorithm that is based on our *Smart AP* model and runs on off-the-shelf hardware. Changes are introduced only in the software (not in the firmware) of the AP, whereas clients do not require any modification at all. Using the second WiFi card in monitor mode, the AP scans all the available channels in the spectrum in order to calculate the interference from neighboring WLANs. Therefore, the AP can detect better channels (those with more capacity) that could boost the performance of its WLAN. Having a dedicated WiFi card for monitoring avoids the disruption of ongoing communications with associated clients, and allows for longer eavesdropping intervals in each channel. Also, the monitoring process and the channel switch decision are executed online, in a transparent and passive way, with no need to introduce traffic for interference measurements.

7.3 Related Work

In Chapter 3, we presented a detailed description of previously published work on the channel assignment problem. In our case, we focus on distributed solutions for decentralized networks. Despite the fact that centralized channel selection can deliver an optimal configuration, it requires a global vision of the network [47, 59, 63, 65, 107]. Home WLANs are uncoordinated networks and the information about neighbors can be obtained only from local measurements.

Most of the channel selection mechanisms for uncoordinated networks that can be found in the literature have a series of properties that do not allow their implementation in today's Home WLANs. Requirements such as communication among APs [3, 10, 69] and AP synchronization [74] cannot yet be fulfilled in residential WLANs. Interference measured by clients gives a better and more complete perspective of the network dynamics [69], but so far this feature is not readily available in the clients' drivers.

As detailed in Section 3.3, the Least Congested Channel Search (LCCS) [4] is a simple channel selection algorithm that selects the best channel using information about the neighboring APs. The AP scans all the channels to determine the most lightly loaded channel, based on the number of associated clients, as published by the beacons of other APs. This metric is too simplistic, as the actual channel traffic load is not necessarily correlated with the number of associated client stations. Worse yet, this algorithm is static, only executed at the AP's startup or when the parameters of the radio are modified. A channel can be declared empty just because at that time a neighbor AP was inactive or did not have any associated clients. Other examples of similar metrics include the total number of active clients [69] and the RSS of transmitted frames [54].

Several dynamic traffic-aware channel assignment mechanisms can be found in the research literature. Although designed for centralized networks, Rozner et al. address the problem by employing historical SNMP samples provided by the APs [91]. In multi-channel wireless mesh networks, nodes periodically exchange information about channel utilization [85]. However, in residential wireless networks, each AP is independent of the others, and there is no protocol that allows communication among APs through the wireless medium (nor through the Internet).

In a first approach, channel selection is done by measuring the downlink traffic seen by the AP [53, 60]. An improved and very recent technique, published at the same time our own solution was developed, computes the airtime consumed by the Home WLAN and other neighboring WLANs as part of the interference metric [43]. The authors consider the combined problem of frequency and channel width selection. Out-of-band measurements are carried out periodically with the same 802.11 card used for AP functionality. Consequently, these micro-sensing intervals slightly reduce the throughput of ongoing traffic exchanges.

The use of a second wireless card dedicated to monitoring is useful for two reasons: first, current traffic transmissions with the AP are not interrupted by measurements of other channels, and second, constant scanning provides a more complete and precise view of the wireless spectrum. Monitoring the 802.11 spectrum can also be practical for different reasons, such as network management [5] and rogue AP detection [92]. The use of sampling techniques [26], hopping through every channel and processing captured frames, seems to be sufficient to accurately estimate the state of the channels, and will be explored in the following sections.

7.4 Traffic-Aware Channel Selection

We propose a new channel assignment technique for uncoordinated chaotic WLANs, using a traffic-aware metric. The aim of our channel selection mechanism is to find the channel with the least interference, in order to improve the network performance. The interference in a channel can be estimated using the traffic captured in that channel. Our technique makes use of a dedicated wireless interface for channel monitoring. By periodically sampling each channel, we can compute the amount of time a channel is occupied by transmissions that belong to neighboring WLANs, and use the results to evaluate the potential effects of a channel switch on our Home WLAN.

We design and implement our channel assignment technique following the *Smart AP* model, described in Chapter 4. Unless explicitly stated, all the operations of the channel selection algorithm are performed by the *monitor interface*, whereas the channel switch operation itself is carried out by the *AP interface*.

In this section, we first introduce the design of our technique: we explain the metric we use to measure the channel interference, and we detail the algorithm that samples all the channels, builds the metrics, and makes the decision of switching to a better channel. Lastly, we describe the software implementation of our algorithm and the architecture of the tool that enables the AP to execute our traffic-aware channel selection mechanism.

7.4.1 Channel Interference Estimation Metric

A first, naïve approach

We need a metric to measure the amount of interference in a channel, in order to determine the best channel for our Home WLAN. But since we do not have one monitor interface available for each of the possible channels, we cannot obtain a complete or perfect view of every channel. Therefore, we adopt a sampling approach: we divide the time into intervals, and during each interval, we set our *monitor interface* to a determined frequency and listen to that specific channel for transmissions. Accordingly, we now detail how to compute the first, naïve version of our channel interference metric.

- First, we calculate the airtime (in microseconds) of each frame captured in the channel we are monitoring. This value mainly depends on the frame's transmission duration, which is calculated using the length (in bits) of the frame, $Length_f$, and the bit rate used in the transmission, $Bitrate_f$, plus other constants:

$$Airtime_f = IFS_f + Preamble_f + \frac{Length_f}{Bitrate_f} \tag{7.1}$$

  where IFS_f is an inter-frame space, which depends on the frame's type, and $Preamble_f$ is the transmission duration of the PLCP (Physical Layer Convergence Protocol) preamble. We take the inter-frame space interval into account for the frame's airtime value because, although no real transmission occurs, we consider that it must be reserved in order to continue successfully with the frame's transmission.

  The value of IFS_f depends on the frame's type and the modulation used (DSSS for 802.11b, OFDM for 802.11g, etc). ACK and CTS frames, Data frames with the More Fragments bit set (in the Frame Control field of the MAC header), Data frames part of an RTS/CTS exchange, among others, are all preceded by the $SIFS$. The rest of the frames, such as Management frames, RTS frames, and initial Data frames, are preceded by the $DIFS$.

- Then, in Equation 7.2, and for each channel α (ch_α), we calculate $\lambda_W(ch_\alpha)$: the fraction of time that channel α is occupied by ongoing transmissions, over a window of time W. We extrapolate this value to the entire window W, from the intervals of W that were effectively monitoring channel α.

$$\lambda_W(ch_\alpha) = \left(\sum_{\substack{i \in W \\ i \in ch_\alpha}} \sum_{f \in i} Airtime_f \right) * \left(\frac{Duration_W}{\sum_{\substack{i \in W \\ i \in ch_\alpha}} Duration_i} \right) \tag{7.2}$$

  i represents a sampling interval of W where channel α was being monitored; f represents a frame captured during the interval i; $Airtime_f$ (in microseconds) is calculated using Equation 7.1; $Duration_W$ is the length of the window W (in seconds); and $Duration_i$ is the length of the interval i (in microseconds).

- Finally, our first and naïve channel interference metric $\mathcal{M}_W(ch_\alpha)$ is simply:

$$\mathcal{M}_W(ch_\alpha) = \lambda_W(ch_\alpha) \tag{7.3}$$

To verify this approach through experimental evaluation, we performed a series of tests using two WLANs ($WLAN_1$ and $WLAN_2$), each with one AP and one client station.

We use *iperf* to generate traffic and our own channel monitoring tool to calculate the channel interference metric. In the first series of experiments, $WLAN_1$ (AP_1 and STA_1) is operating in channel 1. STA_1 sends a UDP traffic to AP_1 at a 20 Mb/s data rate during 120 seconds, and AP_2 monitors channels 1 and 2.

Figure 7.1 presents the normalized metrics for channels 1 and 2. Channel 1 clearly shows the load of the traffic between STA_1 and AP_1, but channel 2 indicates almost no interference, although it should definitely suffer from the traffic in its neighbor channel 1. We observe that this naïve channel interference metric, calculated using only the frames captured in that channel, unequivocally fails to reflect the interference from adjacent channels.

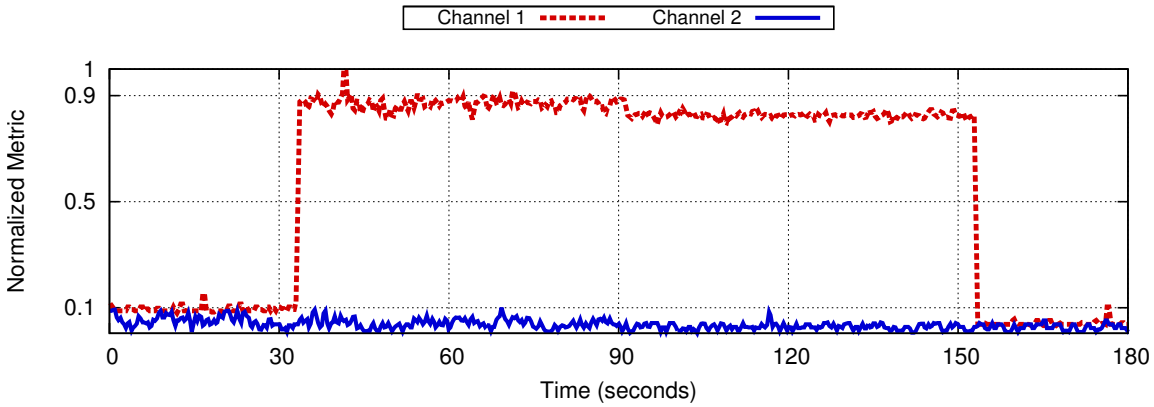
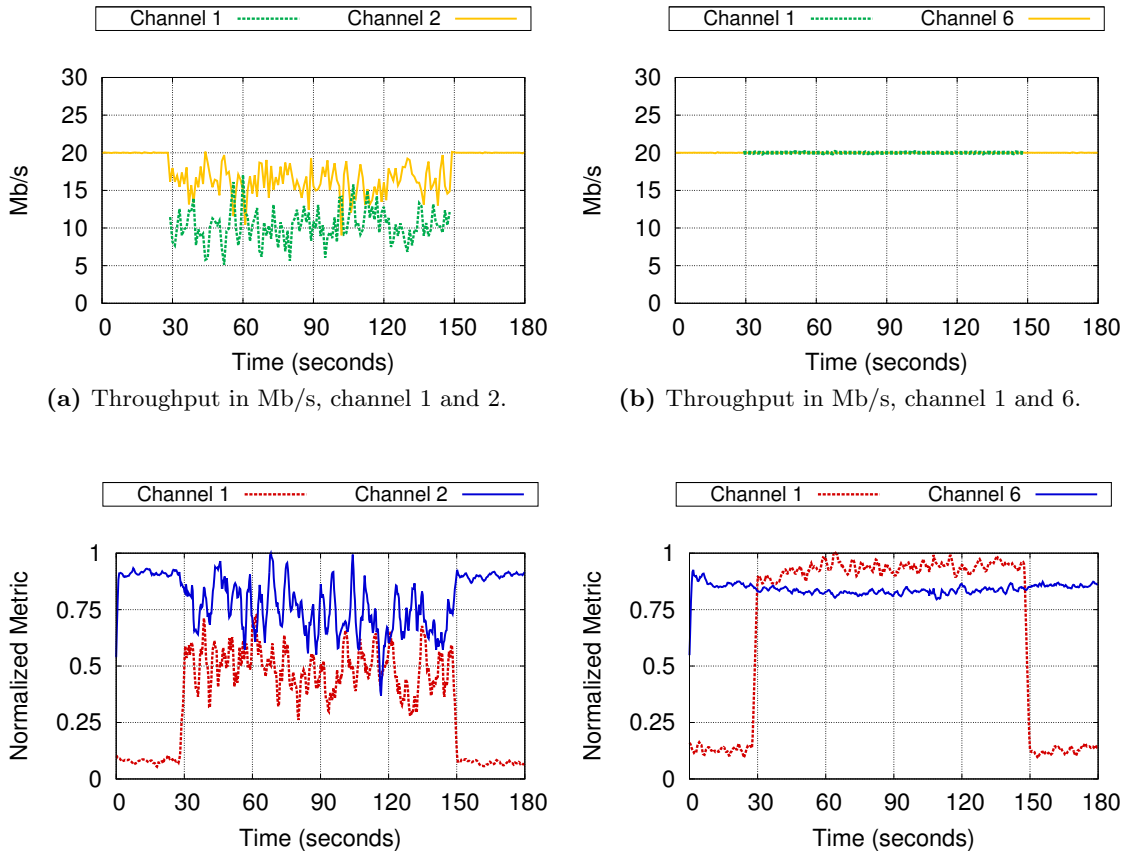


Figure 7.1: Normalized interference metric (naïve approach), channel 1 and 2.

In the second series of experiments, we wanted to explicitly demonstrate the effect of interference between two simultaneous transmissions, comparing the use of overlapping channels with the use of orthogonal channels, and the resultant metrics. In both evaluations, the two WLANs transmit traffic at the same time: STA_2 starts sending a UDP traffic to AP_2 at a 20 Mb/s data rate during 180 seconds, and after 30 seconds, STA_1 sends a UDP traffic to AP_1 at a 20 Mb/s data rate during 120 seconds. AP_1 and STA_1 are in channel 1, while AP_2 and STA_2 are in channel 2 for the first evaluation, and in channel 6 for the the second evaluation.

Figure 7.2 presents the results of the experiments: on the left, the evaluation using overlapping channels 1 and 2, and on the right, the evaluation using orthogonal channels 1 and 6. Figure 7.2a, on the top left, shows the throughput of both WLANs (in channel 1 and 2, respectively). We observe that when STA_1 in channel 1 starts transmitting, the throughput of STA_2 varies erratically from 20 Mb/s to 15 Mb/s, and returns to a stable 20 Mb/s only when STA_1 stops transmitting. Figure 7.2b, on the top right, shows the throughput of both WLANs (in channel 1 and 6, respectively). It is clear that the throughput of STA_2 does not change during the transmission of STA_1. Naturally, the use of orthogonal channels eliminates (or at least reduces drastically) the effect of interference on simultaneous transmissions, whereas the use of overlapping channels distinctly suffers from it.



(a) Throughput in Mb/s, channel 1 and 2.

(b) Throughput in Mb/s, channel 1 and 6.

(c) Normalized interference metric, channel 1 and 2.

(d) Normalized interference metric, channel 1 and 6.

Figure 7.2: Interference effect on simultaneous transmissions, when channels overlap (left) and when channels are orthogonal (right).

We now study the normalized metrics obtained through the monitoring of both channels. Figure 7.2c, on the bottom left, accompanies the throughput results presented above. Between seconds 30 and 150, both metrics follow the traffic load of their own channel. During the first and last 30 seconds of the evaluation, when STA_1 is not transmitting, the metric of channel 2 shows an extremely busy channel, whereas the metric of channel 1 indicates an almost empty channel, although the interference from neighbor channel 2 should definitely appear, since STA_2 is actively transmitting at that time. For comparison, in Figure 7.2d on the bottom right, we see that the metric of channel 6 is stable during the 180 seconds of the experiment, and the metric of channel 1 accurately shows no interference during the first and last 30 seconds, and the rest of the time reflects the traffic load of its own channel.

We conclude that computing a channel interference metric in this naïve way, using only the frames captured in that channel, shows the traffic dynamics of that channel, but omits the interference created by adjacent channels. Indeed, many of the interfering frames from active adjacent channels cannot be successfully decoded by our wireless card, and

are therefore not taken into account during the metric calculations. Therefore, we propose a refined approach that effectively includes the overlapping channel interference into the metric, in order to reflect the actual use of the wireless spectrum.

A better, refined approach

As explained in Section 3.3, channels in the 2.4 GHz band are spaced at 5 MHz intervals, and each channel has a width of 22 MHz (when using 802.11b, or 20 MHz when using 802.11g), clearly overlapping. Therefore, frames that are transmitted in a certain channel can also be overheard (at least partially) in adjacent channels. The amount of interference between two channels, designated as *Interference Factor* in the literature [74], can be calculated, as detailed in [18], using the formulas detailed below.

The power spectrum snx of an unfiltered modulated signal x is given by:

$$snx(x) = \begin{cases} \left|\frac{\sin(2\cdot\pi\cdot x)}{2\cdot\pi\cdot x}\right| & \text{if } x \neq 0 \\ 1 & \text{otherwise} \end{cases} \tag{7.4}$$

Both transmitter and receiver use Intermediate Frequency filtering; the following function $filt$ applies to the SAWTEK 855653 filter [81], and may vary for other chipset models or filter configurations:

$$filt(x) = \frac{1}{1 + (2.6 \cdot x)^6} \tag{7.5}$$

From Equation 7.4 and Equation 7.5, the filtered channel *overlap* between channels α and β, over the band of interest x, is:

$$overlap(\alpha, \beta, x) = \big(filt(ch(\alpha, x)) \cdot snx(ch(\alpha, x))\big) \cdot \big(filt(ch(\beta, x)) \cdot snx(ch(\beta, x))\big) \tag{7.6}$$

$ch(n, f)$, the channel number n and frequency f conversion factor, is given by:

$$ch(n, f) = \frac{f - (2412 + 5 \cdot (n - 1))}{bw} \tag{7.7}$$

where bw is the null-to-null channel bandwidth in MHz, a constant equal to 22 MHz for HR/DSSS signals. As an example, when the frequency f is equal to channel n's center frequency, $ch(n, f)$ is equal to 0.

Finally, the Interference Factor $\mathcal{IF}$ between channels α and β, of channel separation $\Delta = |\alpha - \beta|$, is:

$$\begin{gathered} \mathcal{IF}(\Delta) = \frac{1}{\mathcal{S}_0} \cdot \int_{2200}^{2700} overlap(\alpha, \beta, x)\ \mathrm{d}x \\ \Longleftrightarrow \\ \mathcal{IF}(\Delta) = \frac{1}{\mathcal{S}_0} \cdot \int_{2200}^{2700} overlap(1, 1 + \Delta, x)\ \mathrm{d}x \end{gathered} \tag{7.8}$$

where $\mathcal{S}_0$ is a scaling value that should result in the maximum Interference Factor of 1 when the channels overlap completely, i.e. $\alpha = \beta \iff \Delta = 0$:

$$\mathcal{IF}(0) = \frac{1}{\mathcal{S}_0} \cdot \int_{2200}^{2700} overlap(1,1,x)\ \mathrm{d}x = 1$$
$$\iff \tag{7.9}$$
$$\mathcal{S}_0 = \int_{2200}^{2700} overlap(1,1,x)\ \mathrm{d}x = 9.2655$$

Accordingly, we now detail how to compute our improved channel interference metric, using the Interference Factor $\mathcal{IF}$ described above.

- First, we calculate the airtime $Airtime_f$ (in microseconds) of each frame captured **and originated** in the channel we are monitoring, as detailed in Equation 7.1.

- Then, in Equation 7.10, and for each channel α (ch_α), we calculate $\lambda_W(ch_\alpha)$: the fraction of time that channel α is occupied by ongoing transmissions, actually originated in that channel, over a window of time W. We extrapolate this value to the entire window W, from the intervals of W that were effectively monitoring channel α.

$$\lambda_W(ch_\alpha) = \left(\sum_{\substack{i \in W \\ i \in ch_\alpha}} \sum_{\substack{f \in i \\ f \in ch_\alpha \\ f \notin WLAN_\gamma}} Airtime_f \right) * \left(\frac{Duration_W}{\sum_{\substack{i \in W \\ i \in ch_\alpha}} Duration_i} \right) \tag{7.10}$$

  i represents a sampling interval of W where channel α was being monitored; f represents a frame captured during the interval i, that was originated in channel α, but does not belong to our own $WLAN_\gamma$; $Airtime_f$ is calculated using Equation 7.1; $Duration_W$ is the length of the window W; and $Duration_i$ is the length of the interval i.

  We would like to mention again that in order to calculate $\lambda_W(ch_\gamma)$ for our $WLAN_\gamma$ (operating in channel γ), we do not take into account the airtime of the frames belonging to our own $WLAN_\gamma$ (frames sent by our AP or associated stations) since we want to determine the amount of interference from neighbor WLANs.

- Finally, our new and improved channel interference metric $\mathcal{M}_W(ch_\alpha)$ is the total interference our WLAN would suffer in channel α from all the neighbor WLANs that operate in every possible channel of the spectrum, over a window of time W. It is calculated as the sum of the interference from all the available channels:

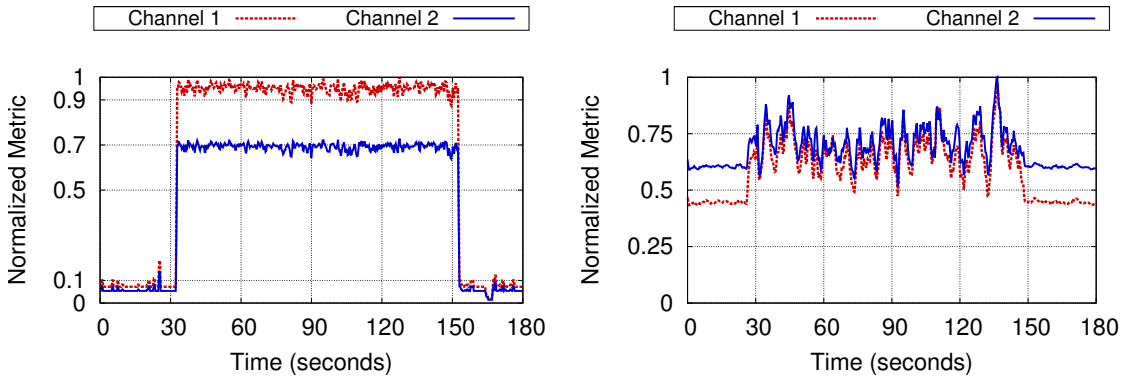
$$\mathcal{M}_W(ch_\alpha) = \sum_{ch_\beta \in CH} \left(\lambda_W(ch_\beta) \cdot \mathcal{IF}(|ch_\alpha - ch_\beta|) \right) \tag{7.11}$$

where channel β is in the set of all available channels CH (e.g., channels 1 to 11 in the 2.4 GHz band); $\lambda_W(ch_\beta)$ is obtained from Equation 7.10; and $\mathcal{IF}(|ch_\alpha - ch_\beta|)$ denotes the *Interference Factor* between channels α and β, obtained from Equation 7.8.

As a last remark, $\mathcal{IF}$ should normally be applied to the signal power of each transmitted frame, but the wireless NICs installed in Home APs do not possess the calibration needed to provide a correct estimation of the received signal power. Therefore, in our metric, we use $\mathcal{IF}$ directly with the busy time fraction of a channel, $\lambda_W(ch_\beta)$, in order to approximate the interference perceived by each channel.

To validate our new metric and the use of $\mathcal{IF}$, we repeat the previous experiments and confirm that this improved metric reflects the interference created by traffic from overlapping channels, as depicted in Figure 7.3.

On the left, in Figure 7.3a, STA_1 (in channel 1) sends a UDP traffic to AP_1 at a 20 Mb/s data rate between seconds 30 and 150, while AP_2 monitors channels 1 and 2. On the right, in Figure 7.3b, STA_2 (in channel 2), sends a UDP traffic to AP_2 at a 20 Mb/s data rate during the whole evaluation, and at around 30 seconds, STA_1 (in channel 1) starts sending a UDP traffic to AP_1 at a 20 Mb/s data rate, and stops at around 150 seconds. We observe that in both figures the channel metrics reflect not only the load of the channel itself, but also the interference from the adjacent channel.



(a) One transmission. **(b)** Simultaneous transmissions.

Figure 7.3: Normalized interference metric, channel 1 and 2.

7.4.2 Channel Selection Algorithm

In this section, we detail the algorithm that makes the decision of switching channels. We use our channel interference metric to choose the new channel, but there are a few other conditions that must be met in order to perform a channel switch of the entire Home WLAN. These considerations include the stability of the interference level in the new channel, the characteristics of the traffic exchanged in our own WLAN, the time elapsed since the last

channel switch operation, and a minimum threshold of interference to avoid ping-pong effects (continuously changing from one channel to another). Algorithm 7.1 depicts the pseudocode of the channel selection mechanism.

- For each channel in the list of all channels available for monitoring, the AP starts listening to the corresponding frequency during an interval of time i, using the *monitor interface*, and processes every captured frame. After that interval, the AP switches to the next channel in the list, and repeats the operation. We call *monitoring cycle* the period of time that the AP spends scanning once every channel of the channel list (line 1). When the AP completes a monitoring cycle, it computes the interference metric $\mathcal{M}_W(ch_\alpha)$ for each channel (line 2) and starts the channel selection mechanism.

- On the one hand, if a channel switch timer has not yet been set up (line 3), the AP checks whether switching to a new channel will improve the performance of the WLAN (line 4). If this is true, we save the best channel as the candidate channel (line 5) and set up a random timer that will effectively perform the channel switch when expired (line 6). The AP then continues with the channel monitoring.

- On the other hand, if a channel switch timer has already been set up, the AP first checks whether the candidate channel stills holds its position as the best channel (or near-best) to switch the WLAN to (line 7). If this is the case, and the timer has expired (line 8), the timer is turned off (line 9) and the AP proceeds to perform the channel switch (line 10). However, if a channel switch to the candidate channel is not a favorable option, the timer is stopped (line 11) and the pending channel switch is canceled. Finally, in both outcomes, the AP continues with the channel monitoring.

We now clarify some details about this algorithm. The main goal is to choose a good channel with a stable metric, so when we find a channel that meets the conditions for a channel switch, we set a timer for a certain period of time (line 6). During this period, we verify that the candidate channel does not show abrupt changes in the interference metric, and stays an optimal solution to improve the WLAN performance (line 7). Additionally, as other APs of neighboring WLANs can also implement this algorithm, we choose a random duration for the timer in order to minimize the probability that two or more neighboring WLANs switch to the same channel at the same time, which would result in degraded performance.

The conditions required to start and continue with the channel switch operation are evaluated in the function `MaySwitchTo` (line 12). We first determine if the WLAN itself, independently of the candidate channel, is allowed to perform a channel switch. We test the following conditions: (i) the WLAN has stayed in its channel at least a minimum amount of time, preventing rapid-fire channel switches that would lead to a ping-pong behavior (line 18); (ii) the interference of the current channel is greater than a minimum threshold,

Algorithm 7.1: Channel Selection Algorithm

```
Program ChannelMonitoring ():
    [ ··· ]
1   while ChannelSampling () do
2       UpdateChannelsMetric ();
        /* Verify if there is a pending channel switch operation */
3       if CurrentChannel.Timer == None then
            /* No candidate channel yet */
4           if MaySwitchTo (BestChannel) then
                /* Best channel meets conditions for a channel switch */
5               CurrentChannel.Candidate = BestChannel;
6               CurrentChannel.Timer = RandomTime();
        else
            /* Confirm current channel switch operation */
7           if MaySwitchTo (CurrentChannel.Candidate) then
8               if CurrentChannel.Timer has Expired then
                    /* Perform channel switch */
9                   CurrentChannel.Timer = None;
10                  SwitchTo (CurrentChannel.Candidate);
                else
                    /* Channel switch operation canceled */
11                  CurrentChannel.Timer = None;
    [ ··· ]

/* Verify if NewChannel meets the conditions for a channel switch */
12 Function bool MaySwitchTo(NewChannel):
13      if not MaySwitch () then
            return FALSE;
14      if BestChannel WAY_BETTER_THAN NewChannel then
            return FALSE;
15      if CurrentChannel WAY_BETTER_THAN NewChannel then
            return FALSE;
16      return TRUE;

/* Verify if WLAN is allowed to perform a channel switch,
   independently of NewChannel */
17 Function bool MaySwitch():
18      if SwitchedTooRecently () then
            return FALSE;
19      if CurrentChannel GOOD_ENOUGH then
            return FALSE;
20      if HaveRealTimeOrCriticalTraffic () then
            return FALSE;
21      return TRUE;
```

avoiding unnecessary channel switches (line 19); and (iii) the current traffic is not critical or real-time, e.g. emergency calls, preventing the disruption of such sensitive communications (line 20).

If the three previous conditions are met, the AP can qualify for a channel switch. Subsequently, we evaluate two more conditions, related to the candidate channel: it should still be the best or near-best option for a channel switch compared to (i) other channels (line 14) and (ii) the current channel (line 15), avoiding a switch to a channel whose level of interference has increased since we first chose it as the candidate channel, or if there is another channel that can provide a better performance to our WLAN than the candidate channel.

Frame Filtering

We would like to dedicate a few lines to explaining how we decide whether a frame was originally transmitted in the currently monitored channel, in order to calculate $\lambda_W(ch_\alpha)$. The AP scans all available frequencies and processes every captured frame. Frames transmitted in one channel can be received in overlapping channels. Therefore, we need to determine the channel in which the frame was originally transmitted.

First, we create a dataset of the neighboring BSSs using their beacon frames, which contain their BSSID MAC address (the same as the AP's) and their operating channel. The rest of the management frames, control and (especially) data frames, help us discover their clients' MAC addresses.

After that, when a frame is captured, we first verify whether its BSSID MAC address exists in the dataset. If found, we extract the operating channel of the BSS from the dataset. Finally, if we confirm that the frame was indeed originated in the channel we are currently monitoring, we calculate the frame's airtime and use it in our metric.

7.4.3 Implementation

Our channel selection algorithm was developed as a C program, called *channelmon*. As can be seen in Figure 7.4, the *Smart AP* model provides us with two wireless NICs: one card for AP-related functions, and one card for monitoring. We use the hostapd [48] daemon as an IEEE 802.11 AP. *Channelmon* is also running on the AP as a daemon. Having a dedicated WiFi card for the monitoring task prevents disruption of ongoing communications, and enables longer and more frequent scanning intervals in each channel.

The *channelmon* application continuously monitors all available channels, and processes the frames captured by the *monitor* interface. When the algorithm finds a better channel for the AP to operate in, the *channelmon* application informs the hostapd daemon of the channel switch, indicating the new channel number for the AP.

The implementation of our algorithm does not introduce any modification on the client side, as mandated by the requirements of the *Smart AP* model. To capture the

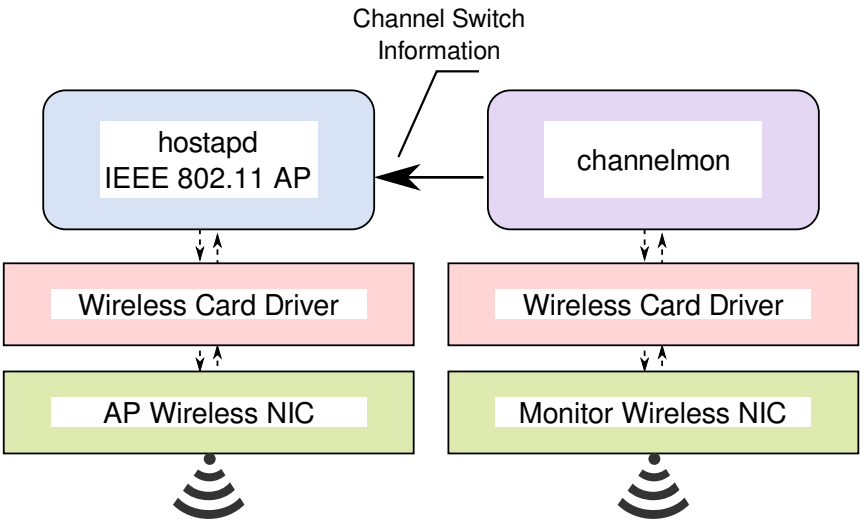


Figure 7.4: Architecture of the Channel Selection mechanism.

whole network dynamics, it would also be possible to include the interference measured by client stations in our channel interference metric, as defined in the 802.11k and 802.11v amendments, presented in Section 3.3.1.

In order to calculate the airtime of each captured frame, as described in Equation 7.1, we obtain several parameters from the radiotap headers [83], such as the preamble type (long or short), the data rate, the channel type (e.g., 802.11b, 802.11g only, 802.11g with 802.11b support), and the channel frequency.

The following values are used for the inter-frame spaces [35]:

	802.11b	**802.11b/g**	**802.11g**	**802.11a**
SIFS	10 μs	10 μs	10 μs	16 μs
Slot time	20 μs	20 μs	9 μs	9 μs
DIFS	50 μs	50 μs	28 μs	34 μs

Table 7.1: IFS values for the different 802.11 technologies.

where 802.11b/g means 802.11g with 802.11b support. $DIFS$ is calculated as $(2 \cdot Slot\ time + SIFS)$.

And for the preamble transmission durations:

Long preamble	**Short preamble**
192 μs	96 μs

Table 7.2: Preamble transmission durations.

In order to determine the amount of channel overlap, we use the normalized values of the *Interference Factor* $\mathcal{IF}(\Delta)$ described in Equation 7.8 and calculated in [18]:

Channel Separation	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7 to 10**
Interference Factor	1	0.7272	0.2714	0.0375	0.0054	0.0008	0.0002	0

Table 7.3: Interference Factor $\mathcal{IF}(\Delta)$.

7.5 Evaluation

In this section, we present the evaluation of our channel assignment technique on a small testbed that corresponds to a typical Home WLAN setup. The objective of this evaluation is to provide a proof-of-concept for our mechanism, and evidence that our technique can be effectively implemented in off-the-shelf hardware. The results show the improvement of the network performance when applying our channel selection algorithm.

7.5.1 Platform and Experiments Description

The experiments were carried out in an urban residential area, where a large number of Home WLANs can be found in the 2.4 GHz band. We decided to include the background interference generated mostly by the APs' beacons, since this reflects a common scenario in Home WLANs. However, we performed the experiments at night, to avoid severe interference from unrelated traffic.

For our platform, we used laptops for the APs and the client stations. They all run Ubuntu 12.04. The APs have two cards: one for the AP functionality, a D-Link DWL-AG660 wireless card running the ath5k driver; and another one for the monitor functionality, a TP-Link TL-WN821N wireless card running the ath9k driver. Each station has one D-Link DWL-AG660 wireless card and runs the ath5k driver.

Our testbed consists of three WLANs, operating in the 2.4 GHz band. Each WLAN is composed of one AP and one station. $WLAN_3$ has an AP which runs the *channelmon* application, whereas $WLAN_1$ and $WLAN_2$ act as neighboring WLANs. $WLAN_1$ and $WLAN_2$ operate in orthogonal channels. $WLAN_3$ starts in the same channel as $WLAN_1$ and switches channels along the experiment, choosing the most suitable channel as designated by the channel selection algorithm. The *channelmon* application was executed with the following parameters: the window size W is 150 s (a value large enough to get a clear picture of every channel's state, but small enough to not miss any opportunity for channel switching and performance improvement), the monitoring interval i for each channel is 100 ms, and the list of channels to scan goes from 1 to 11. To generate the wireless traffic

in the WLANs, we used the tool *iperf*. In each WLAN, the AP sends a downlink traffic to the station. We evaluated two different scenarios, one using TCP traffic and one using UDP traffic.

7.5.2 Interference Factor Adjustment

In the first scenario, we performed the evaluation using TCP traffic. $WLAN_1$ operates in channel 11 and $WLAN_2$ in channel 6. $WLAN_3$ starts operating in channel 11. AP_1 and AP_2 send TCP traffic without interruption to STA_1 and STA_2, respectively. On average, $WLAN_1$ has a throughput of 14.2 Mb/s and $WLAN_2$ has a throughput of 16.8 Mb/s. After 170 seconds, AP_3 sends a TCP traffic to STA_3 during 120 seconds, with a throughput of 9.17 Mb/s on average. After the transmission, at around 314 seconds, AP_3 switches to channel 3, as designated by the *channelmon* application. AP_3 then resends a TCP traffic to STA_3 during 120 seconds. This time, $WLAN_3$ has a throughput of 9.96 Mb/s, which shows a slight improvement of 8.6%. During (and because of) AP_3's first and second transmissions, the throughputs of $WLAN_1$ and $WLAN_2$ decrease to 7.22 Mb/s and 9.04 Mb/s, respectively.

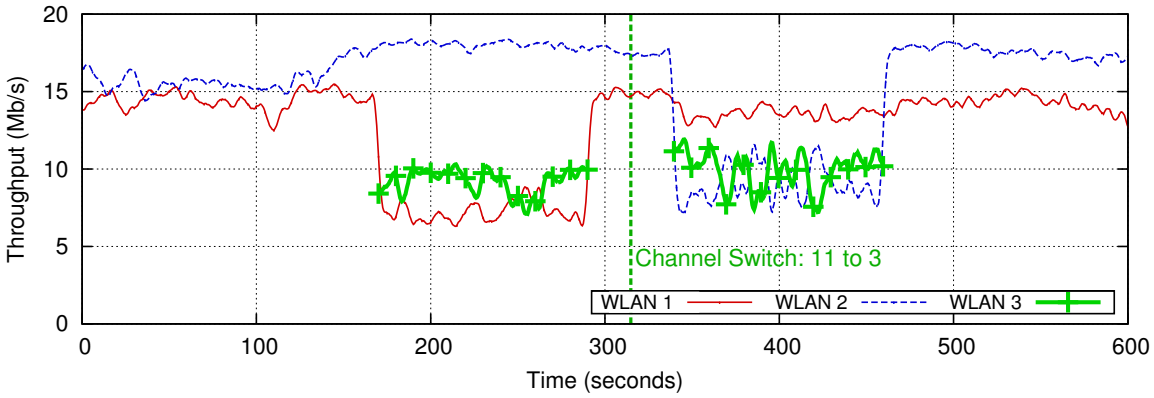


Figure 7.5: TCP traffic evaluation: $WLAN_3$ switches from channel 11 to channel 3.

Despite the improvement of $WLAN_3$'s network performance, we know that choosing channel 1 instead of channel 3 would have led to a better performance yet, because although the metrics of channel 1 and channel 3 are very close, as depicted in Figure 7.6, channel 1 is farther away (than channel 3) from the interference created in channel 6. We infer that the theoretical values of the Interference Factor are too optimistic, since the metrics obtained do not perfectly reflect the actual interference. As an example of one source of error, the theoretical calculation of $\mathcal{IF}(\Delta)$ included Equation 7.5 for an Intermediate Frequency filter, which may vary depending on the chipset model and filter configuration.

In order to validate our intuition about the error of the $\mathcal{IF}$ values, we apply a fractional power K between 0 and 1 to these values, to increase them and consequently amplify the interference from adjacent channels. As a first approach, we chose $K = 0.5$ (square root),

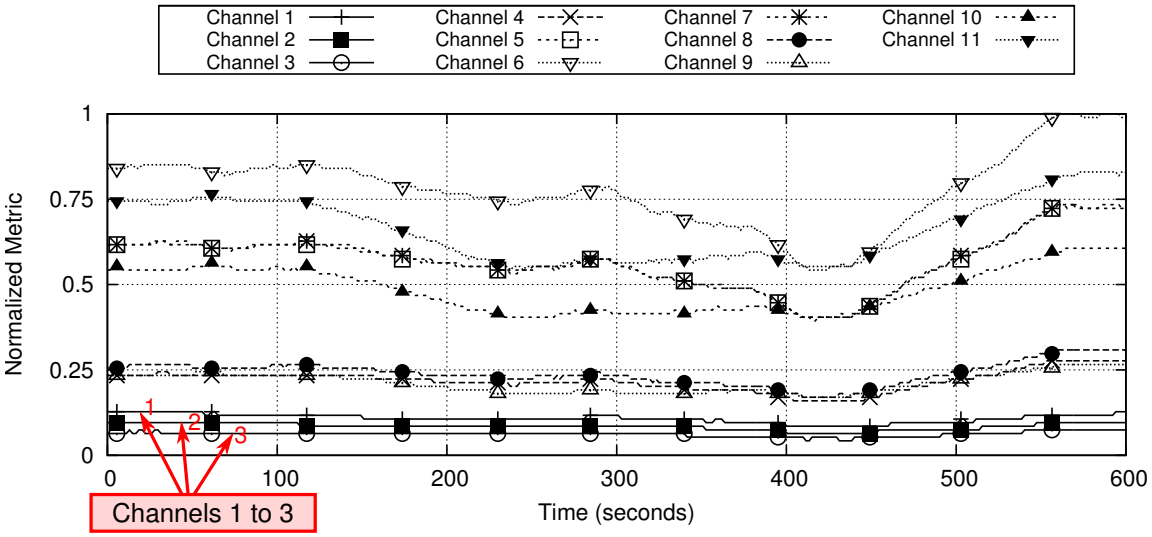


Figure 7.6: TCP traffic evaluation: interference metric for channels 1 to 11.

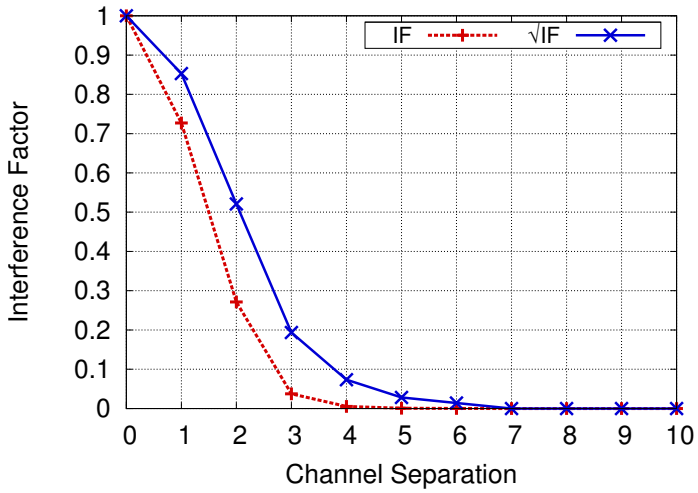


Figure 7.7: Interference Factor values (theoretical vs. adjusted).

a common function that follows closely the original function but generates higher values. As an illustration, in Figure 7.7 we compare the theoretical Interference Factor values $\mathcal{IF}(\Delta)$ (detailed in Table 7.3) to the adjusted values $\sqrt{\mathcal{IF}(\Delta)}$.

7.5.3 Results

We repeated the previous experiments with the adjusted Interference Factor values. $WLAN_3$ starts operating in channel 11, but now the *channelmon* application effectively requests AP_3 to switch to channel 1, the channel with the least interference, as shown in Figure 7.8. Figure 7.9 depicts the throughput of the three WLANs, and $WLAN_3$ shows an important performance improvement of 122%, from 8.30 Mb/s to 18.50 Mb/s. The decrease in the metrics, at around 300 seconds, is due to limitations in multi-radio wireless

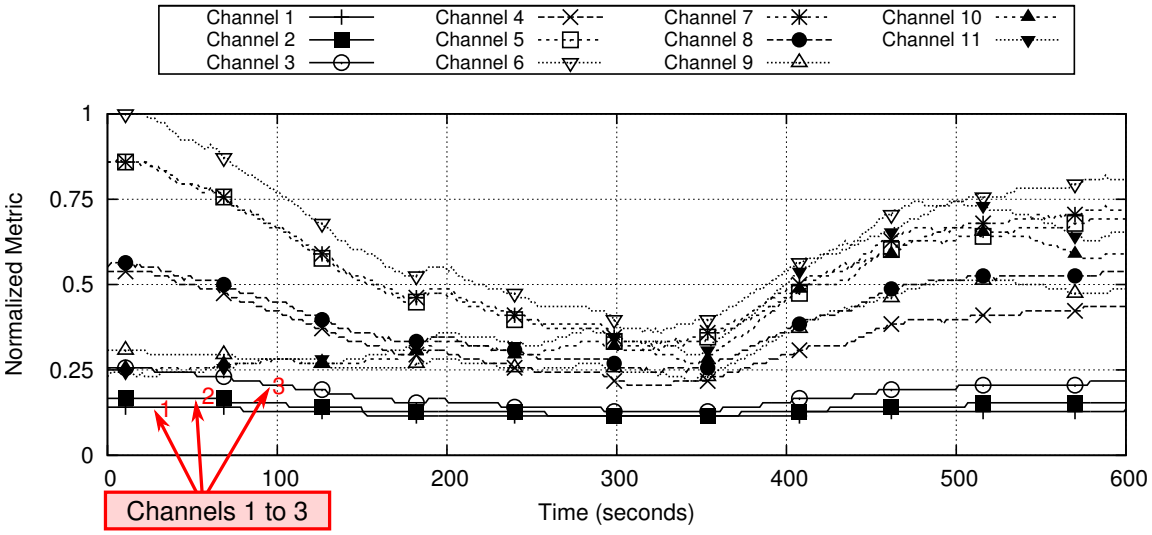


Figure 7.8: TCP traffic evaluation: interference metrics for channels 1 to 11, using $\sqrt{\mathcal{IF}(\Delta)}$.

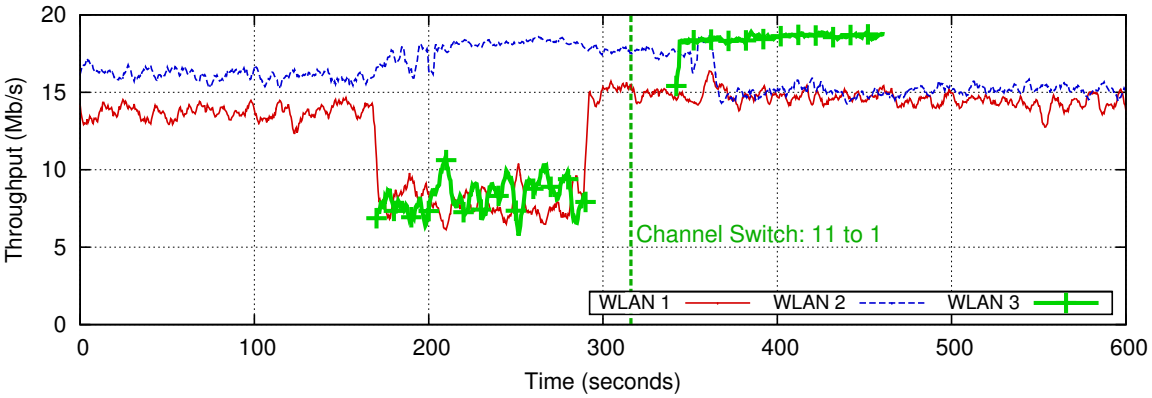


Figure 7.9: TCP traffic evaluation: $WLAN_3$ switches from channel 11 to channel 1, using $\sqrt{\mathcal{IF}(\Delta)}$. The increase in network performance is notably greater than during the previous evaluation.

platforms [89]: when AP_3 sends an intense TCP traffic through its *AP interface*, its *monitor interface* captures less traffic. However, the relative order (vs. absolute order) of the channels sorted by their normalized metric is conserved, which is the critical property of the metric for the Channel Selection algorithm.

For the second scenario, we performed the same experiments, but used UDP traffic. $WLAN_1$ now operates in channel 1 and $WLAN_2$ in channel 6. $WLAN_3$ starts operating in channel 1. AP_1 and AP_2 send a UDP traffic at a 13 Mb/s data rate (e.g. a streaming application), without interruption, to STA_1 and STA_2, respectively. At around 170 seconds, AP_3 sends a UDP traffic to STA_3 at a 13 Mb/s data rate during 120 seconds. At the end of the transmission, AP_3 switches to channel 11, as designated by the *channelmon* application. AP_3 then resends a UDP traffic to STA_3 during 120 seconds.

Figure 7.10 shows the results of this evaluation. When sharing channel 1 with $WLAN_1$, between seconds 170 and 290, $WLAN_3$ has a throughput of 11.7 Mb/s. After switching to channel 11, the UDP traffic of $WLAN_3$ has a stable throughput of 13 Mb/s. Figure 7.11 shows the aggregate throughput of the three WLANs along the experiment. We observe that after the channel switch, the three UDP flows have a stable throughput of 13 Mb/s, making a total throughput of 39 Mb/s.

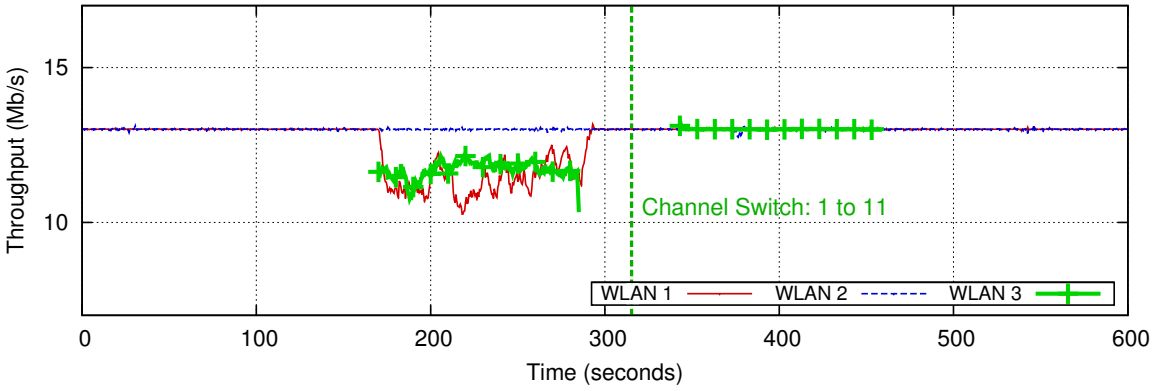


Figure 7.10: UDP traffic evaluation: $WLAN_3$ switches from channel 1 to channel 11, and thus improves its throughput.

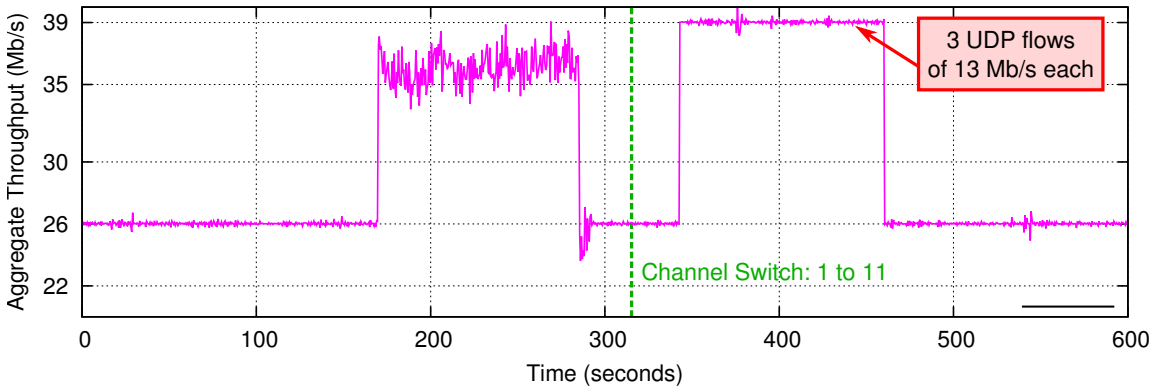


Figure 7.11: UDP traffic evaluation: aggregate throughput of all three WLANs.

With all the results obtained, we conclude that our channel selection mechanism, based on our traffic-aware metric, effectively improves the network performance by choosing the channel with the least interference.

7.6 Summary

In this chapter, we presented a new solution for the channel selection problem, specifically designed for uncoordinated Home WLANs. Our mechanism is adaptive and online, because it takes into account the dynamics of the wireless medium; and it is passive, because it monitors every channel during a fixed amount of time, in order to capture the traffic

characteristics of every channel. We proposed a traffic-aware metric that quantifies the amount of interference a channel suffers from its neighboring channels. We implemented our technique in off-the-shelf hardware, without any modification in the clients, and evaluated two different scenarios using TCP and UDP downlink traffic. Results show that our technique improves the network performance, by choosing the channel with the least interference.

Future work should include: adjustment of parameters, such as the window size W, the monitoring interval i, and more importantly, the Interference Factor $\mathcal{IF}(\Delta)$ (i.e., find the optimal value of K through simulations); performance evaluations that include the interference measured by clients; and further evaluations to determine the scalability and convergence of the algorithm.

Chapter 8

CONCLUSIONS

Contents

The subject of study in this dissertation has been *single-hop infrastructure* IEEE 802.11 WLANs. In particular, we examined the important aspects of a WLAN's deployment and identified issues that can affect its performance. Reviewing the state of the art, we observed that numerous research efforts have proposed diverse solutions with several limitations that impede their use in existing WLANs: they propose manual or static optimization mechanisms, cannot be practically implemented in today's wireless cards, or require modifications in the IEEE 802.11 standard, neglecting interoperability with existing WiFi devices.

Motivated by these challenges, the aim of this dissertation has been to design and implement novel but practical solutions that address open issues affecting the performance of IEEE 802.11 WLANs. We focused on the following aspects of WLANs: client mobility, channel management, and quality of service, and explored three different scenarios for the most common deployments: an enterprise, a city (urban area), and a personal residence (home). For each scenario, we identified interesting opportunities for optimization and innovation and proposed original solutions, which we validated via simulation and experimentation.

In this final chapter, we first summarize the main contributions of our dissertation and then outline future research directions for our work.

8.1 Contributions

To provide a common basis for practical implementation of new 802.11 solutions, we presented a *Smart AP* model, inspired by self-management techniques. We adopted an AP-based approach, which does not require any modification in the clients. Indeed, the proposed mechanisms can benefit from the AP's leading role in a WLAN to improve the network performance of the clients. Furthermore, clients are now so diverse that a global driver modification has become an almost impossible task.

Our primary objective was to develop novel and practical 802.11 solutions that can be deployed in existing WLANs and thus do not introduce changes in the WiFi protocol, allowing interoperability with today's WiFi devices. We utilized adaptive algorithms in order to reflect the wireless medium and network dynamics, by continuous monitoring and measurement of the current performance. Cross-layer design played an important role there, by combining information from the Physical and MAC layers.

We presented the basic architecture for the implementation of our *Smart AP* model, which principally consists of two commodity 802.11 Atheros-based wireless cards, extensively used by the research community. One wireless card is responsible for the AP functionality, while the other card is dedicated to monitoring activity in all channels, allowing the AP to simultaneously serve the clients in its own channel, without interruption. We implemented our solutions as applications that run in user space, which alleviates the dependence on the driver and kernel versions.

The main contributions of this thesis are the following:

1. We presented a transparent mobility solution for VoIP services in Enterprise WLANs, called *Multichannel Virtual Access Points* (mVAP). It provides seamless handoffs with no performance degradation, for applications with tight delay constraints. This network-based mobility technique does not introduce any modification in the client behavior and is compatible with the existing 802.11 protocol.

   We implemented mVAP in commodity hardware, using a brand new version of our PACMAP framework and running on top of the MadWifi driver. We experimentally evaluated the mVAP performance with different VoIP codecs: results showed that mVAP handles the change of AP without disrupting the ongoing communications and offers exceptional handoff performance.

2. We investigated the feasibility of exploiting the WiFi coverage in urban areas for mobile Internet access. First, we characterized the distribution of the WiFi APs already deployed in a city and properties such as channel assignment, link quality, and authentication mode. We also studied the patterns of users' paths in order to derive a simple mobility model.

   We simulated different scenarios, varying mobile user speeds and AP properties such as range, association type, and handoff parameters, and we measured the WiFi coverage,

Internet access session, connection, and disconnection durations. The results showed that the existing WiFi coverage is large and the connectivity it offers can be exploited in new and creative ways.

Finally, we proposed several applications that can benefit from this Internet access provided by already deployed WiFi APs, and discussed the challenges that should be faced in order to run these applications effectively.

3. We proposed an adaptive traffic-aware channel selection mechanism for uncoordinated Home WLANs. Our technique takes into account the dynamics of the wireless medium and makes online channel switch decisions based on local measurements. Channel load estimation is done in a transparent and passive way, by monitoring every channel and without introducing traffic to measure the interference. We proposed a traffic-aware metric and showed that this metric accurately quantifies the amount of interference a channel suffers from its neighboring channels.

   We implemented a proof-of-concept of our technique using commodity WiFi cards and evaluated two scenarios using, respectively, TCP and UDP downlink traffic. The results obtained showed that this channel selection mechanism, based on our traffic-aware metric, effectively improves the network performance by constantly choosing the channel with the least interference.

8.2 Future Work

The work presented in this dissertation has permitted us to identify new opportunities and open challenges related to the deployment and utilization of IEEE 802.11 WLANs. We now divide extensions of our work into two categories: (1) Enterprise and Home WLANs, which are indoor deployments where clients are trusted and allowed access to all available resources, and (2) Citywide networks, which are loose collections of APs distributed throughout a city, where clients are roaming and allowed only partial access to resources, usually broadband Internet connections.

Enterprise and Home WLANs

The development of a complete *Smart AP* framework (based on our model) is critical to enable the automation of WLANs. The *Smart AP* manages the radio and network resources for the subsequent optimization of the WLAN's performance. It follows an autonomic approach, in which the AP continuously monitors the environment and the WLAN's performance, and performs specific actions to adapt to the current scenario and demands. This framework could be implemented in OpenWrt, which already provides a package manager for easy customization and upgrades. Algorithms designed for different scenarios and goals (for instance, our mVAP and channel selection mechanisms) could be

easily "plugged-in" to the framework, providing a common platform for the deployment of research prototypes and facilitating code reuse.

Despite the utilization of fast communication techniques such as MIMO, wider-bandwidth channels, and IEEE 802.11ad Gigabit WiFi, traffic management is still an open challenge in WLANs, as WiFi-enabled devices continue to proliferate. Web traffic (mostly downlink) represents a significant fraction of the total amount of data traffic, and video consumption over the Internet (downlink and uplink) is predicted to increase considerably in the next few years [79]. Data-hungry devices such as smartphones and tablets compete with real-time and rich-media applications for channel access and bandwidth, so APs must implement mechanisms to provide real-time QoS guarantees, and estimate and anticipate traffic demands.

Citywide networks

Users who access the Internet primarily via mobile wireless connections can benefit from inter-technology mobility in urban areas. But there are two important problems that need to be solved in order to improve the performance of such a mobile wireless connectivity:

- support of seamless mobility between heterogeneous wireless technologies, by means of intelligent handover decisions;
- smooth adaptation of multimedia content across the different connections, e.g. dynamic adjustment of video bit rate and resolution.

Moreover, the caching of popular data close to users has recently attracted research attention [56]. In this approach, data is generally stored in the routers along the network path, in order to minimize network traffic and provide faster content delivery. A new interesting proposal for this problem would be to store and cache data in the numerous APs deployed throughout cities. However, their limited storage capacity is one of the challenges to overcome when developing such a solution.

Bibliography

[1] Aad, I., and Castelluccia, C. Differentiation mechanisms for IEEE 802.11. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (2001), vol. 1, IEEE, pp. 209–218.

[2] Aad, I., and Castelluccia, C. Remarks on per-flow differentiation in IEEE 802.11. In *Proc. of European Wireless* (2002), vol. 2002.

[3] Abusubaih, M., Rathke, B., and Wolisz, A. A framework for interference mitigation in multi-BSS 802.11 wireless LANs. In *World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a* (June), pp. 1–11.

[4] Achanta, M. Method and apparatus for least congested channel scan for wireless access points, Oct. 5 2004. US Patent App. 10/959,446.

[5] Adya, A., Bahl, P., Chandra, R., and Qiu, L. Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking* (2004), ACM, pp. 30–44.

[6] Aguayo, D., Bicket, J., Biswas, S., Judd, G., and Morris, R. Link-level measurements from an 802.11b mesh network. In *SIGCOMM* (2004).

[7] Ahmed, N., Keshav, S., and Papagiannaki, K. OmniVoice: a mobile voice solution for small-scale enterprises. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (2011), ACM, p. 5.

[8] Aircrack. `http://www.aircrack-ng.org/`.

[9] Akella, A., Judd, G., Seshan, S., and Steenkiste, P. Self-management in chaotic wireless deployments. In *ACM MobiCom* (2005).

[10] Akl, R., and Arepally, A. Dynamic channel assignment in IEEE 802.11 networks. In *Portable Information Devices, 2007. PORTABLE07. IEEE International Conference on* (2007), IEEE, pp. 1–5.

[11] Arjona, A., and Takala, S. The Google Muni Wifi Network–Can it Compete with Cellular Voice? In *AICT* (2007).

[12] Balasubramanian, A., Mahajan, R., Venkataramani, A., Levine, B. N., and Zahorjan, J. Interactive wifi connectivity for moving vehicles. *SIGCOMM CCR* (2008).

[13] Battiti, R., Bertossi, A., and Cavallaro, D. A randomized saturation degree heuristic for channel assignment in cellular radio networks. *Vehicular Technology, IEEE Transactions on 50*, 2 (2001), 364–374.

[14] Berezin, M. E., Rousseau, F., and Duda, A. Multichannel Virtual Access Points for Seamless Handoffs in IEEE 802.11 Wireless Networks. In *VTC* (2011).

[15] Berezin, M. E., Rousseau, F., and Duda, A. Citywide mobile internet access using dense urban WiFi coverage. In *Proceedings of the first workshop on Urban networking* (2012), UrbaNe '12, ACM, pp. 31–36.

[16] Brélaz, D. New methods to color the vertices of a graph. *Communications of the ACM 22*, 4 (1979), 251–256.

[17] Broustis, I., Papagiannaki, K., Krishnamurthy, S., Faloutsos, M., and Mhatre, V. MDG: measurement-driven guidelines for 802.11 WLAN design. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking* (2007), ACM, pp. 254–265.

[18] Burton, M. Channel overlap calculations for 802.11b networks. *White Paper, Cirond Networks Inc* (2002).

[19] Bychkovsky, V., Hull, B., Miu, A., Balakrishnan, H., and Madden, S. A measurement study of vehicular internet access using in situ Wi-Fi networks. In *ACM MobiCom* (2006).

[20] CableWifi. `http://www.cablewifi.com/`.

[21] Calhoun, P., Montemurro, M., and Stanley, D. Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification. RFC 5415 (Proposed Standard), Mar. 2009.

[22] Carrera, M., Srikantha, P., May, M., and Rosenberg, C. SHAPE: Scheduling in wireless home network. Tech. rep., Technical report, Technicolor and UPMC Paris Universitas and University of Waterloo, 2011.

[23] Chen, K. Medium access control of wireless LANs for mobile computing. *Network, IEEE 8*, 5 (1994), 50–63.

[24] Chen, X., and Qiao, D. HaND: fast handoff with null dwell time for IEEE 802.11 networks. In *Proceedings of the 29th conference on Information communications* (Piscataway, NJ, USA, 2010), INFOCOM'10, IEEE Press, pp. 1604–1612.

[25] Chhaya, H., and Gupta, S. Throughput and fairness properties of asynchronous data transfer methods in the IEEE 802.11 MAC protocol. In *Personal, Indoor and Mobile Radio Communications, 1995. PIMRC'95.'Wireless: Merging onto the Information Superhighway'., Sixth IEEE International Symposium on* (1995), vol. 2, IEEE, pp. 613–617.

[26] Deshpande, U., Henderson, T., and Kotz, D. Channel sampling strategies for monitoring wireless networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on* (2006), IEEE, pp. 1–7.

[27] Dunn, J., Neufeld, M., Sheth, A., Grunwald, D., and Bennett, J. A practical cross-layer mechanism for fairness in 802.11 networks. In *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on* (2004), IEEE, pp. 355–364.

[28] Eisenblätter, A., Grötschel, M., and Koster, A. M. *Frequency planning and ramifications of coloring*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2000.

[29] Even, G., Lotker, Z., Ron, D., and Smorodinsky, S. Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. *SIAM Journal on Computing 33*, 1 (2003), 94–136.

[30] FON. `http://www.fon.com/`.

[31] FreeWifi. `http://www.free.fr/`.

[32] Gast, M. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly Media, 2005.

[33] Geier, J. Assigning 802.11b access point channels, 2004.

[34] Giustiniano, D., Malone, D., Leith, D. J., and Papagiannaki, K. Experimental assessment of 802.11 MAC layer channel estimators. *IEEE Communications Letters 11*, 12 (2007), 961–963.

[35] Group, I. . W. IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE 802.11 LAN Standards 2007* (2007).

[36] Grunenberger, Y., Heusse, M., Rousseau, F., and Duda, A. Experience with an implementation of the Idle Sense wireless access method. In *Proceedings of the 2007 ACM CoNEXT conference* (2007), ACM, p. 24.

[37] Grunenberger, Y., and Rousseau, F. Virtual access points for transparent mobility in wireless LANs. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE* (2010), IEEE, pp. 1–6.

[38] Grunenberger, Y., and Rousseau, F. Virtual Access Points for Transparent Mobility in Wireless LANs. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)* (2010).

[39] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and Patil, B. Proxy Mobile IPv6. RFC 5213 (Proposed Standard), Aug. 2008. Updated by RFC 6543.

[40] Gupta, A., Min, J., and Rhee, I. WiFox: scaling WiFi performance for large audience environments. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies* (2012), ACM, pp. 217–228.

[41] Hale, W. K. Frequency assignment: Theory and applications. *Proceedings of the IEEE 68*, 12 (1980), 1497–1514.

[42] Han, B., Ji, L., Lee, S., Miller, R., and Bhattacharjee, B. Channel access throttling for improving WLAN QoS. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on* (2009), IEEE, pp. 1–9.

[43] Herzen, J., Merz, R., and Thiran, P. Distributed Spectrum Assignment for Home WLANs. In *IEEE Infocom* (2013), IEEE.

[44] Heusse, M., Rousseau, F., Berger-Sabbatel, G., and Duda, A. Performance anomaly of 802.11 b. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies* (2003), vol. 2, IEEE, pp. 836–843.

[45] Heusse, M., Rousseau, F., Guillier, R., and Duda, A. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs. *ACM SIGCOMM Computer Communication Review 35*, 4 (2005), 121–132.

[46] Heusse, M., Starzetz, P., Rousseau, F., Berger-Sabbatel, G., and Duda, A. Bandwidth allocation for DiffServ based quality of service over 802.11. In *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE* (2003), vol. 2, IEEE, pp. 992–997.

[47] Hills, A. Large-scale wireless LAN design. *Communications Magazine, IEEE 39*, 11 (2001), 98–107.

[48] hostapd. `http://hostap.epitest.fi/`.

[49] Huang, P., Tseng, Y., and Tsai, K. A fast handoff mechanism for IEEE 802.11 and IAPP networks. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd* (2006), vol. 2, IEEE, pp. 966–970.

[50] Iperf. `http://iperf.sourceforge.net/`.

[51] Jiunn, D., and Chang, R. A priority scheme for IEEE 802.11 DCF access method. *IEICE Transactions on Communications 82*, 1 (1999), 96–102.

[52] Jones, K., and L, L. What Where Wi: An Analysis of Millions of Wi-Fi Access Points . In *IEEE Portable* (2007).

[53] Kauffmann, B., Baccelli, F., Chaintreau, A., Mhatre, V., Papagiannaki, K., and Diot, C. Measurement-based self organization of interfering 802.11 wireless access networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* (2007), IEEE, pp. 1451–1459.

[54] Kazdaridis, G., Keranidis, S., Fiamegkos, A., Korakis, T., Koutsopoulos, I., and Tassiulas, L. Novel metrics and experimentation insights for dynamic frequency selection in wireless LANs. In *Proceedings of the 6th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization (WiNTECH)* (2011), ACM, pp. 51–58.

[55] Kephart, J. O., and Chess, D. M. The vision of autonomic computing. *Computer 36*, 1 (2003), 41–50.

[56] Koponen, Teemu and Chawla, Mohit and Chun, Byung-Gon and Ermolinskiy, Andrey and Kim, Kye Hyun and Shenker, Scott and Stoica, Ion. A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Computer Communication Review* (2007), vol. 37, ACM, pp. 181–192.

[57] Laurila, J. K., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T.-M.-T., Dousse, O., Eberle, J., and Miettinen, M. The mobile data challenge: Big data for mobile computing research. In *Mobile Data Challenge by Nokia Workshop* (2012), Pervasive.

[58] Lee, K., Rhee, I., Lee, J., Chong, S., and Yi, Y. Mobile data offloading: how much can WiFi deliver? In *Co-NEXT* (2010).

[59] Lee, Y., Kim, K., and Choi, Y. Optimization of AP placement and channel assignment in wireless LANs. In *Local Computer Networks, 2002. Proceedings. LCN 2002. 27th Annual IEEE Conference on* (2002), IEEE, pp. 831–836.

[60] Leung, K., and Kim, B. Frequency assignment for IEEE 802.11 wireless networks. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th* (2003), vol. 3, IEEE, pp. 1422–1426.

[61] Liao, Y., and Gao, L. Practical Schemes for Smooth MAC Layer Handoff in 802.11 Wireless Networks. In *WoWMoM* (2006).

[62] LIBEVENT. http://www.monkey.org/~provos/libevent/.

[63] LING, X., AND YEUNG, K. Joint access point placement and channel assignment for 802.11 wireless LANs. In *Wireless Communications and Networking Conference, 2005 IEEE* (2005), vol. 3, IEEE, pp. 1583–1588.

[64] MADWIFI. http://madwifi-project.org/.

[65] MAHONEN, P., RIIHIJARVI, J., AND PETROVA, M. Automatic channel allocation for small wireless local area networks using graph colouring algorithm approach. In *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on* (2004), vol. 1, IEEE, pp. 536–539.

[66] MANGOLD, S., CHOI, S., MAY, P., KLEIN, O., HIERTZ, G., AND STIBOR, L. IEEE 802.11 e Wireless LAN for Quality of Service. In *Proc. European Wireless* (2002), vol. 2, pp. 32–39.

[67] MDC, N. http://research.nokia.com/mdc.

[68] MHATRE, V., AND PAPAGIANNAKI, K. Using smart triggers for improved user performance in 802.11 wireless networks. In *MobiSys* (2006).

[69] MISHRA, A., BANERJEE, S., AND ARBAUGH, W. Weighted coloring based channel assignment for WLANs. *ACM SIGMOBILE Mobile Computing and Communications Review 9*, 3 (2005), 19–31.

[70] MISHRA, A., BRIK, V., BANERJEE, S., SRINIVASAN, A., AND ARBAUGH, W. A client-driven approach for channel management in wireless LANs. In *IEEE Infocom* (2006), vol. 6.

[71] MISHRA, A., ROZNER, E., BANERJEE, S., AND ARBAUGH, W. Exploiting partially overlapping channels in wireless networks: turning a peril into an advantage. In *IMC* (2005).

[72] MISHRA, A., SHIN, M., AND ARBAUGH, W. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. *SIGCOMM Comput. Commun. Rev. 33*, 2 (2003), 93–102.

[73] MISHRA, A., SHIN, M., AND W., A. Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. In *INFOCOM* (2004).

[74] MISHRA, A., SHRIVASTAVA, V., AGRAWAL, D., BANERJEE, S., AND GANGULY, S. Distributed channel management in uncoordinated wireless environments. In *Proceedings of the 12th annual international conference on Mobile computing and networking* (2006), ACM, pp. 170–181.

[75] Mishra, A., Shrivastava, V., Banerjee, S., and Arbaugh, W. Partially overlapped channels not considered harmful. In *Proceedings of the joint international conference on Measurement and modeling of computer systems* (New York, NY, USA, 2006), SIGMETRICS '06/Performance '06, ACM, pp. 63–74.

[76] Mohan, P., Padmanabhan, V. N., and Ramjee, R. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys* (2008).

[77] OpenWrt. `http://openwrt.org/`.

[78] Pack, S., Jung, H., Kwon, T., and Choi, Y. SNC: a selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks. *ACM SIGMOBILE Mobile Computing and Communications Review 9*, 4 (2005), 39–49.

[79] Paper, C. W. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015, February 2011.

[80] Park, S., Kim, H., Park, C., Kim, J., and Ko, S. Selective channel scanning for fast handoff in wireless LAN using neighbor graph. In *Personal Wireless Communications* (2004), Springer, pp. 629–629.

[81] Pearson, B., and Culibrk, M. Choosing the IF Frequency for the PRISM II 11MBPS Radio Reference Design. Tech. rep., Technical report, Intersil Corporation, http://www. intersil. com/design/prism, 2000.

[82] Perkins, C. IP Mobility Support for IPv4, Revised. RFC 5944 (Proposed Standard), Nov. 2010.

[83] radiotap. `http://www.radiotap.org/`.

[84] Ramani, I., and Savage, S. SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks. In *INFOCOM* (2005).

[85] Raniwala, A., and Chiueh, T.-c. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* (2005), vol. 3, IEEE, pp. 2223–2234.

[86] Rappaport, T. *Wireless Communications: Principles and practice*. Prentice Hall, 2002.

[87] Riihijarvi, J., Petrova, M., and Mahonen, P. Frequency allocation for WLANs using graph colouring techniques. In *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on* (2005), IEEE, pp. 216–222.

[88] Riihijarvi, J., Petrova, M., Mahonen, P., and Barbosa, J. Performance evaluation of automatic channel assignment mechanism for IEEE 802.11 based on graph colouring. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on* (2006), IEEE, pp. 1–5.

[89] Robinson, J., Papagiannaki, K., Diot, C., Guo, X., and Krishnamurthy, L. Experimenting with a multi-radio mesh networking testbed. In *1st workshop on Wireless Network Measurements* (2005).

[90] Romdhani, L., Ni, Q., and Turletti, T. Adaptive EDCF: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE* (2003), vol. 2, IEEE, pp. 1373–1378.

[91] Rozner, E., Mehta, Y., Akella, A., and Qiu, L. Traffic-aware channel assignment in enterprise wireless LANs. In *Network Protocols, 2007. ICNP 2007. IEEE International Conference on* (2007), IEEE, pp. 133–143.

[92] Sheng, Y., Chen, G., Yin, H., Tan, K., Deshpande, U., Vance, B., Kotz, D., Campbell, A., McDonald, C., Henderson, T., et al. MAP: A scalable monitoring system for dependable 802.11 wireless networks. *Wireless Communications, IEEE 15*, 5 (2008), 10–18.

[93] Shin, M., Mishra, A., and Arbaugh, W. A. Improving the latency of 802.11 hand-offs using neighbor graphs. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services* (New York, NY, USA, 2004), MobiSys '04, ACM, pp. 70–83.

[94] Siris, V., and Stamatakis, G. Optimal CWmin selection for achieving proportional fairness in multi-rate 802.11 e WLANs: test-bed implementation and evaluation. In *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization* (2006), ACM, pp. 41–48.

[95] Sobrinho, J., and Krishnakumar, A. Real-time traffic over the IEEE 802.11 medium access control layer. *Bell Labs Technical Journal 1*, 2 (1996), 172–187.

[96] Soroush, H., Gilbert, P., Banerjee, N., Corner, M. D., Levine, B. N., and Cox, L. Spider: improving mobile networking with concurrent wi-fi connections. In *Proceedings of the ACM SIGCOMM 2011 conference* (2011), SIGCOMM.

[97] Srivastava, V., and Motani, M. Cross-layer design: a survey and the road ahead. *Communications Magazine, IEEE 43*, 12 (2005), 112–119.

[98] Sundaresan, K., and Papagiannaki, K. The need for cross-layer information in access point selection algorithms. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (2006), ACM, pp. 257–262.

[99] Tan, G., and Guttag, J. Time-based fairness improves performance in multi-rate WLANs. In *Proceedings of the annual conference on USENIX Annual Technical Conference* (2004), Boston, MA, pp. 23–23.

[100] Teng, J., Xu, C., Jia, W., and Xuan, D. D-Scan: Enabling Fast and Smooth Handoffs in AP-dense 802.11 Wireless Networks. In *INFOCOM* (2009).

[101] Tripathi, N. D., Reed, J. H., and VanLandinoham, H. F. Handoff in cellular systems. *Personal Communications, IEEE 5*, 6 (1998), 26–37.

[102] Union, I. T. International Telephone Connections and International Telephone Circuits. *ITU-TG.114* (2003).

[103] Vaidya, N., Dugar, A., Gupta, S., and Bahl, P. Distributed fair scheduling in a wireless LAN. *Mobile Computing, IEEE Transactions on 4*, 6 (2005), 616–629.

[104] Velayos, H., and Karlsson, G. Techniques to reduce the IEEE 802.11 b handoff time. In *Communications, 2004 IEEE International Conference on* (2004), vol. 7, IEEE, pp. 3844–3848.

[105] Veres, A., Campbell, A., Barry, M., and Sun, L. Supporting service differentiation in wireless packet networks using distributed control. *Selected Areas in Communications, IEEE Journal on 19*, 10 (2001), 2081–2093.

[106] VTun. `http://vtun.sourceforge.net/`.

[107] Wertz, P., Sauter, M., Landstorfer, F., Wolfle, G., and Hoppe, R. Automatic optimization algorithms for the planning of wireless local area networks. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th* (2004), vol. 4, IEEE, pp. 3010–3014.

[108] Wireless, L. `http://wireless.kernel.org/`.

[109] Wu, H., Tan, K., Zhang, Y., and Zhang, Q. Proactive scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN. In *INFOCOM* (2007).

[110] Yoo, S., Choi, J., Hwang, J., and Yoo, C. Eliminating the Performance Anomaly of 802.11 b. *Networking-ICN 2005* (2005), 1055–1062.

[111] Yoon, J., Noble, B. D., and Liu, M. Building realistic mobility models from coarse-grained traces. In *ACM MobiSys* (2006).

[112] Yu, M., Luo, H., and Leung, K. A dynamic radio resource management technique for multiple APs in WLANs. *Wireless Communications, IEEE Transactions on 5*, 7 (2006), 1910–1919.