ADAPTIVE DISCOVERY MECHANISM FOR WIRELESS ENVIRONMENTS

by

German Castignani

An Engineering Thesis submitted to:

Facultad de Ingeniería
Universidad de Buenos Aires

Fulfillment of the requirement of the degree of

Ingeniero en Informática

Departamento de Computación

Facultad de Ingeniería - Universidad de Buenos Aires

August 2009

Co-directed thesis between the Universidad de Buenos Aires and TELECOM BRETAGNE within the framework of an internshiph agreement:

Dr. Jose Ignacio Alvarez-Hamelin, Universidad de Buenos Aires, Argentina
Dr. Nicolas Montavont, TELECOM BRETAGNE, Cesson-Sevigne, France

ABSTRACT

ADAPTIVE DISCOVERY MECHANISM FOR WIRELESS ENVIRONMENTS

German Castignani

FACULTAD DE INGENIERIA - UNIVERSIDAD DE BUENOS AIRES

Departamento de Computación

The mobility necessity in the 802.11 wireless environment requires handover processes to be executed, so a mobile station must detect and associate to a new access point while moving. As network deployment conditions occur in a heterogeneous manner, a general Discovery Algorithm is required to satisfy all possible scenarios. A new line of attack to perform discovery processes is defined in this document, applying an adaptive comportment.

DEDICATION

To my parents.

ACKNOWLEDGMENTS

Initially, I want to express gratitude to Dr. Nicolas Montavont, who always suggested and encouraged me in the daily work. In addition, a special appreciation has to be made since Dr. Montavont accepted to write and submit a paper regarding to the main purpose of this Thesis.

Then, I would like to recognize Dr. Jose Ignacio Álvarez-Hamelin for his support from my University in Argentina.

As this Thesis was developed under an Internship Program in France, I would like to acknowledge EGIDE (Centre Français pour l'Accueil et les Échanges Internationaux) for the financial support during the internship period.

An important thank has to be made to all the community of the *grande école* TELECOM Bretagne. Its recognized researchers, administrative authorities and PhD students always offered their help during my stage.

Finally I do not want to forget very important people walking with me in the same direction, Alejandro Paltrinieri and Hernan Danielan.

Contents

List of Figures

Chapter 1

Introduction

Nowadays, 802.11 networks appear as the most widely implemented wireless access in the market, while a vast number of devices embed WiFi chipsets. These situation produces both benefits and constraints. The main benefit is directly referred to compatibility and portability, so users can employ their devices in different networks, accessing to diverse type of services and the Internet. Mobility is also an additional benefit, but inside the 802.11 environment, it implies several limitations. The fact that 802.11 networks are so widely deployed generates a significant medium saturation, and even more, because of the coexistence of different devices in the 2.4 GHz spectrum high levels of interference are also frequent. Access points using this technology are usually cost reduced, so then, several operators coexist in the same geographic location, producing overlapped coverage areas.

A mobile user will take profit of this overlapped coverage areas so as to move in the direction of other neighbor access point while it is still associated to its current point of attachment. The process of moving to other point of attachment is referred as a ***handover***.

These processes deal with time related constraints, so their duration should be minimized so as to avoid non desirable effects on the client side, due to disconnections in the application layer. The 802.11 standard specification has established two different mechanisms to obtain information about the next access point to associate. Both approaches have a number of limitations that affect the effectiveness of the handover process, so it is possible that no access point is found and a disconnection occurs. This situation takes place due to deployment's heterogeneity presented in the first paragraph.

This research proposes a new approach so as to increase the successfulness of the handover process while reducing its duration. An ***Adaptive Discovery Algorithm*** is presented so as to achieve this goal. An adaptive mechanism, that dynamically decides how long to wait for access points' responses, should produce successful handover

processes independently of the network deployment heterogeneity.

This document is organized as follows. An introduction to 802.11 wireless concepts and mobility, a detail on management operations together with a deep state-of-the-art analysis in the field of handover optimizations is offered in chapter 2. Then, the new scanning technique is proposed in chapter 3. Subsequently, in order to validate the designed algorithm, simulations scenarios and a real testbed are proposed in chapter 4. Finally, conclusions and a description of future work are stated in chapter 5.

Additionally two publications have been proposed to the community based on this research. These papers can be found as appendices.

Chapter 2

The 802.11 Context

Understanding the 802.11 wireless world is the first step to attack so as to deepen in the introduced problematic. An overview into 802.11 Protocol Suite will help to be aware of the context and smoothly focus on proposed optimizations.

Medium Access Control (MAC) first and Physical (PHY) Layers next, will be explained with an appropriate level of detail. Furthermore the state-of-the-art of those layers will be taken into consideration.

Some notions about Wireless Networks Deployment will be put forward in order to help to understand the philosophy of the proposed enhancements.

Concepts about Management Operations and Mobility in 802.11 networks will be expounded in order to give way to the Adaptive Discovery Mechanism definition.

2.1 The ABC of 802.11 Wireless Networks

802.11 Wireless Network Standard [1] is one of the component of the IEEE 802 family. This protocol family defines the specification for local area networks (LAN) technologies. In the last years, the collection of devices implementing the 802.11 standard in any of its available physical options, has undergone a deep growth, since consumers increasingly demands mobile applications. In order to be familiar with this set of protocols, Fig. 2.1 describes the different specifications providing a relation with the OSI Model.

All defined networks in the 802 specification contain both Medium Access Control (MAC) and Physical (PHY) components. As it can be appreciated, the 802 family includes the specifications for the popular 802.3 Carrier Sense Multiple Access with

[1] Inspired from [2] on page 13

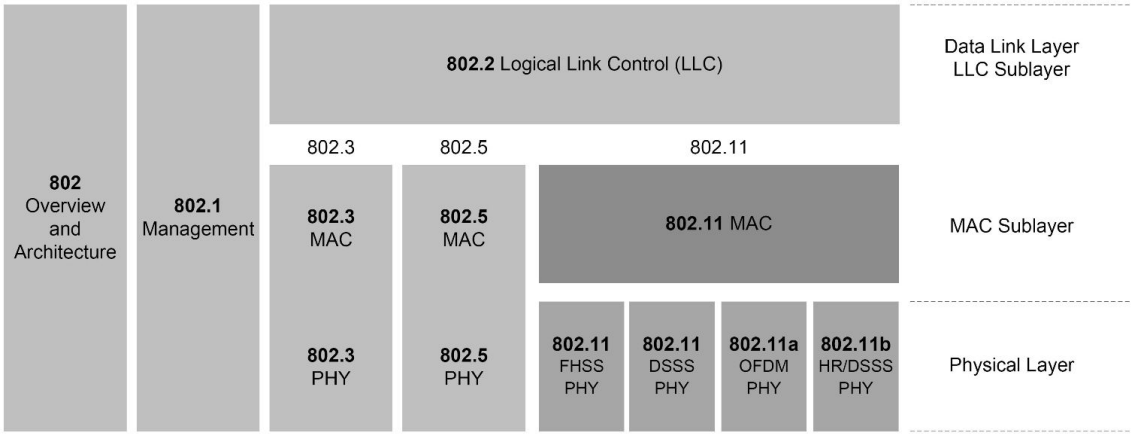


Figure 2.1 The IEEE 802 Protocols family[1]

Collision Detection (CSMA/CD) LAN, related to and often erroneously called Ethernet. Furthermore, 802.5 describes Token Ring networks.

In the case of 802.11 wireless networks, it usually employs 802.2 Logical Link Control (LLC) encapsulation as Data Link Layer, like above described protocols. Also, it defines a single MAC layer that works with several specified PHY layers. It has to be considered that due to the complexity of the wireless environment, 802.11 MAC layer includes numerous features not commonly included in others 802 MAC layers.

2.1.1 Components

With the aim of introducing the main physical components in an 802.11 wireless atmosphere, Fig. 2.2 shows the interaction between the Distribution System, Access Points and Mobile Stations within the Wireless Medium.

Distribution System The Distribution System acts as a logical component of 802.11, it is used to forward frames to the correspondent destination in the case where more than one access point is connected to form a larger coverage area. It usually receives the name of *Backbone Network*, and it can be implemented using a Bridging Engine and a Distribution Medium. Typically it is carried out by using ethernet components, using both copper or optic fiber wires within a switched architecture.

Access Point Access Points deal with the challenge of converting the frames captured from the wireless medium so they can be delivered to the wired world. This feature is executed by a bridging function. Other than bridging, access points can provide other important features like encryption and security.

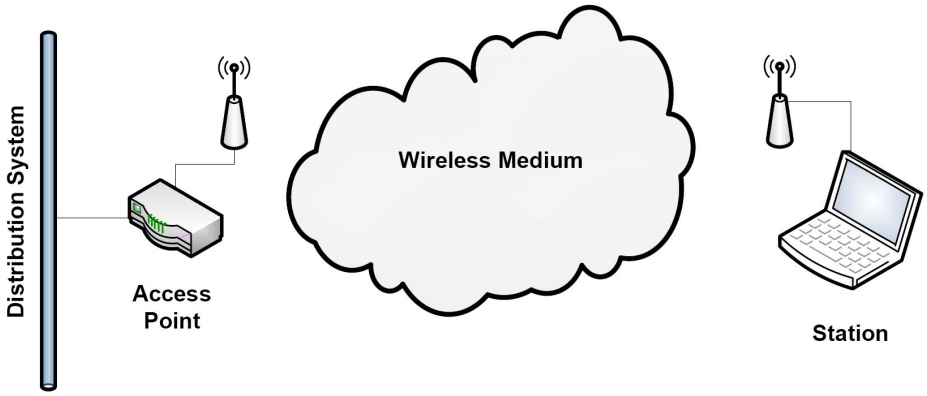


Figure 2.2 IEEE 802.11 Components

Wireless Medium The Wireless Medium takes action in frame delivery operations between stations. The 802.11 specification defines several physical layers that supports the single MAC layer. In the beginnings, two radio frequency (RF) physical layers and one infrared layer were specified.

Station Not only popular laptops should be considered as stations, but any computing device attached to a Wireless Network Interface Card (WNIC) as well. For instance, handhelds, tablet PCs, mobile phones or simple desktop devices could act as a station in the wireless context.

2.1.2 Network Natures

The main building block of an 802.11 network is defined by the *Basic Service Set (BSS)*, consisting in a group of stations communicating each other. Thus, the area where the communication takes place is identified as the *Basic Service Area*, which is conditioned by the wireless medium characteristics. Two different approaches to communicate between stations in the same BSS can be applied. Fig. 2.3 illustrates both architectures.

Ad hoc Mode

This mode is also referred as *independent BSS* (IBSS). Stations communicate directly with each other while they are within a common coverage range. The *ad hoc* designation is related with the fact that this kind of networks are designed for specific purposes and usually for a short period of time. Commonly, when using an *ad hoc* architecture, one of the nodes has access to the Internet, so the other nodes in the network without a direct Internet connection may use the former node's connection so as to reach the public network.

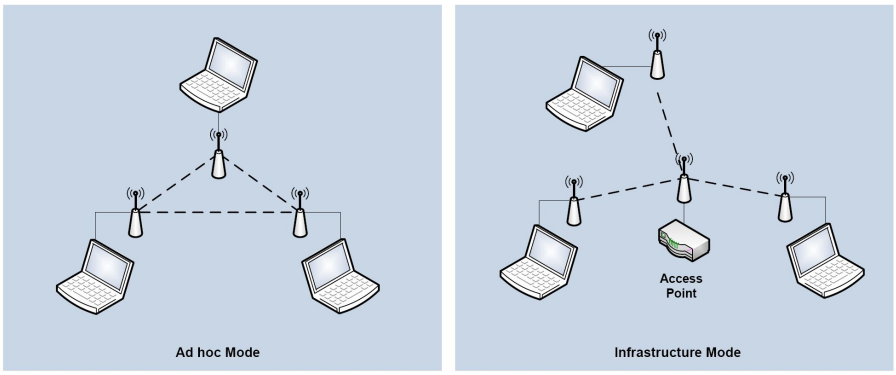


Figure 2.3 Ad hoc and Infrastructure Basic Service Sets

Infrastructure Mode

In this case, access points are used to perform communications between stations in the same or different BSSs through the Distribution System. This kind of architecture provides two main advantages:

- BSSs are defined by the distance from the access point. Differing from an ad hoc approach, stations within the access point range in an infrastructure BSS can communicate each other independently from the separation between them.
- Access points may allow a station to enter in *Power Saving Mode*, so the former can buffer frames targeted to the latter. As mobile stations are usually powered with batteries, this situation allows them to turn on the wireless interface only when a transmission to other station must be achieved.

Similar to a traditional wired ethernet, a station should *associate* with an access point in order to start a communication process. In the 802.11 environment, a station always initiates the association phase performing a request with only one access point, thus, the latter may grant or deny the association. The standard does not define a certain limit for the number of stations connected to one access point, but as relative throughput[2] on a wireless network is correlated with the number of stations, network operators should decide a precise limit for each implementation.

Extended Service Area

So as to achieve a larger coverage area, several BSSs may be chained together using a backbone network and conceiving an *Extended Service Set* (ESS). The main feature offered by an ESS is that a station can send and receive frames from any other station, even though these stations belong to different BSSs. Fig. 2.4 indicates a typical ESS

[2]Throughput concepts are elucidated in Section 2.4.2

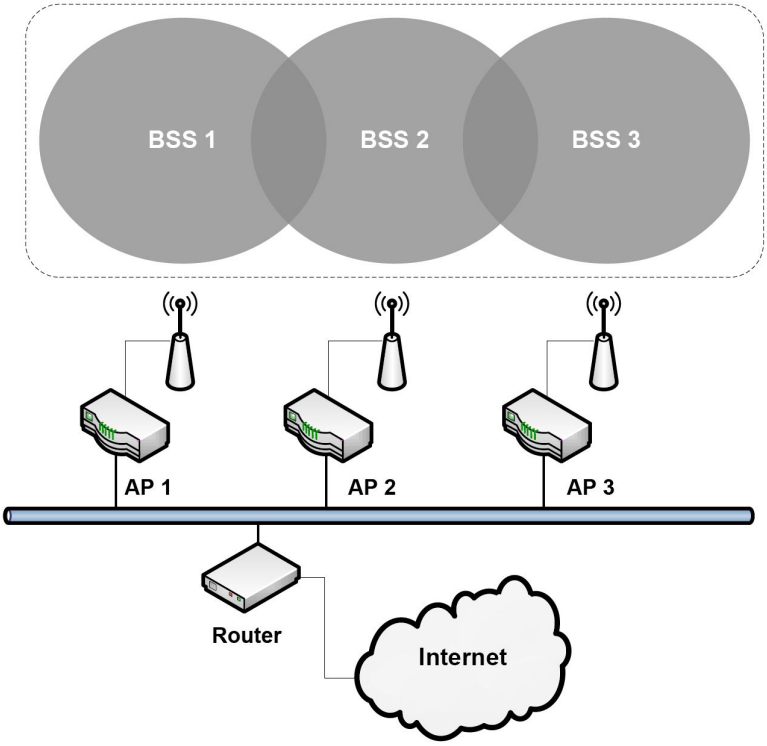


Figure 2.4 Extended Service Set Area (ESS)

deployment scenario. Access points are always responsible to locate the station and deliver frames to the final destination.

2.2 Medium Access Control Layer

As it was introduced above, the 802.11 specification provides a unique MAC Layer supporting all available physical layers. The complete 802.11 MAC Layer set of functions grants user data transmission within the air in a controlled manner. As the wireless environment appears so different compared with the wired medium, 802.11 MAC becomes the essential factor of the whole 802.11 specification.

802.11 MAC Layer was designed focused on several challenges to be taken into consideration. This context is always related with the fact that receiver nodes could not be able to receive frames because of limitations in the air medium. These limitations are caused both by the low RF link quality and the hidden node problem. By the former, it must be assumed that RF transmissions are subject to noise and interference, thus in the case of 802.11 where several physical layers work under unlicensed ISM bands, this limitation takes place in a relevant way. For that reason,

unlike many others link layer protocols, the 802.11 MAC Layer incorporates a *positive acknowledgment mechanism*. This procedure establishes that all transmitted frames must be acknowledged by the receiver, assuring that the medium contention is locked out during an atomic operation.

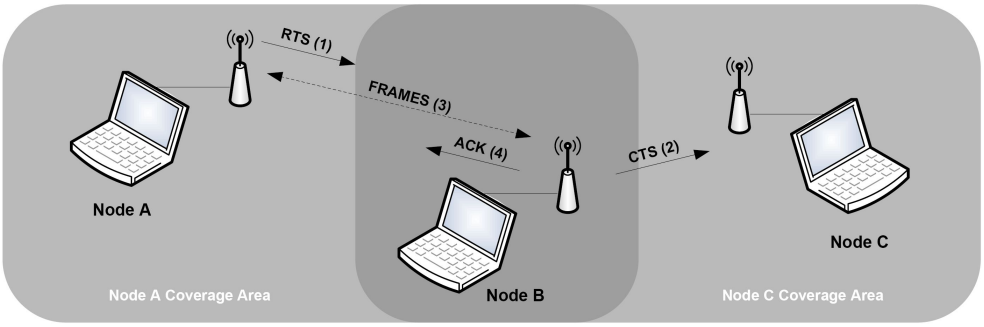


Figure 2.5 Solving the hidden node problem with a RTS/CTS procedure

The latter limitation is associated with the characteristics of the wireless network coverage areas. Fuzzy coverage boundaries characterizes 802.11 networks, and in some scenarios as illustrated in Fig. 2.5, simultaneous transmissions could be initiated, so frame collision occurs and no errors will be reported. In the illustrated case, Node A transmits a frame to Node B. Then Node C can start sending information to Node B because the former is not able to distinguish the simultaneity problem. For that reason, Node C is *hidden* for Node A. So as to prevent these scenarios, the MAC Layer provides a RTS/CTS mechanism. This mechanism allows the sender to request to send (RTS) so it must wait for a clear to send signal (CTS) from the receiver in order to initiate the transmission without being preoccupied about hidden nodes interferences. As this solution adds significant latency, consuming network capacity, the RTS/CTS mechanism could be activated occasionally, depending on each implementation. For instance, RTS/CTS exchanges could be useful in crowded areas with multiple overlapped networks, where the existence of hidden nodes cannot be predicted at all.

2.2.1 CSMA/CA

As like to 802.3 ethernet networks, 802.11 MAC provides a carrier sense multiple access (CSMA) scheme to control the access to the air medium. As collisions waste the transmission capacity, contrarily to detecting them as in the wired approach, a *collision avoidance* method is applied in the 802.11 MAC Layer. In CSMA, the access control is achieved using two *Coordination Functions*, while the carrier sensing becomes a responsibility of the *Network Allocation Vector*.

Access Control The carrier sense multiple access with collision avoidance mechanism (CSMA/CA) is provided by the *distributed coordination function* (DCF) and it is used in most part of communication processes, when nodes have to content for the medium in order to communicate each other. Another defined function in the 802.11 MAC is the *point coordination function* (PCF) that allows a station to send frames avoiding medium contention.

- **DCF** - The DCF behavior provides a contention-based service acting likewise the traditional ethernet tactic, first the radio link must be checked so it has to be clear before sending. The main feature included in the DCF is the *backoff* function detailed in this section.
- **PCF** - The main purpose of the PCF is to provide a contention-free service. This feature is included in the access point, so it only works in an infrastructure architecture, as described in Section 2.1.2. In this case, frames are transmitted after a shorter interval of time

Carrier Sense It is well known that the carrier sense mechanism is implemented by lower physical layers in the wired environment. However, in the wireless world, developing carrier sense hardware introduces high manufacturing costs related to the fact that transceivers can transmit and receive simultaneously only if they introduce expensive components. In addition, physical layers could not solve the hidden node problem, that is worked out by the MAC Layer functions.
As suggested above, 802.11 MAC Layer introduces the usage of the *Network Allocation Vector* (NAV) under the concept of the *Virtual Carrier-Sensing* procedure. Its structure helps to reserve the medium for a certain period of time. To achieve this, it is mandatory for all frames to include the *Duration* field, that indicates the transmission time for the frame, guaranteeing that the medium will be considered busy for this time. Each station listens to wireless frames and reads the *Duration* field to set its NAV. Then stations count down using the value indicated on the NAV as a timer, when NAV reaches zero, the medium is considered idle again. Consequently, stations on the same physical channel appropriately defer the access, based on the information contained on the NAV.

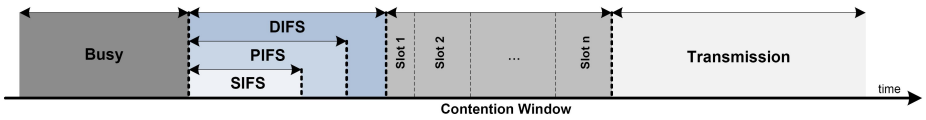


Figure 2.6 Different time spacing between frames: *SIFS*, *PIFS* and *DIFS*[3]

[3]Inspired from http://www.cisco.com/warp/public/cc/pd/witc/ao1200ap/prodlit/qosdg$_a$4.*jpg*

Frame Timing The access control mechanism defines different times that a node could wait before sending a frame. Fig. 2.6 characterizes three different delays, offering distinct priority levels for the medium contention. Different priorities can be applied to different kind of traffic. Other than differentiating time magnitudes, Fig. 2.6 helps to understand the time intervals a node must wait after deciding to consider the medium as idle and initiate the transmission.

- **SIFS** - The Short Interframe Space (SIFS) provides the highest priority for the medium content. RTS/CTS frames and positive acknowledgments (ACK frames) use it to access the medium. Even though each physical layer implements specific values for these times, the most common 802.11 physical layers assign $10\mu s$ as the SIFS delay value.
- **PIFS** - Abbreviation of PCF Interframe Space, it is used in contention-free operations by the PCF function.
- **DIFS** - The DCF Interframe Space is the time used for contention based services. Stations always have access to the medium after waiting an interval no longer than DIFS. Usual values for DIFS are located near $50\mu s$.
- **EIFS** - The Extended Interframe Space is a specific delay used when there is an error in the transmission. Its value is not fixed, so it takes different concrete values.

A special consideration has to be made regarding the above explained issue. This research focuses on the optimization of time related variables, so further enhances could be performed changing the behavior frames wait for the medium to become idle. As it will be presented in Section 3.3.2 these concepts are applied in order to optimize the scanning process, explained in Section 2.5.2.

The backoff function

The most important part of transmitted frames in an 802.11 environment must content for the medium before being transmitted, thus the DCF is applied. DCF allows multiple stations to interact without a central control, distributing the decision making in order to access the medium, avoiding frame collision.
Before trying to send a frame, a station checks if the medium is idle. If the medium is busy, the deferring process is started and an orderly exponential backoff is calculated, so as to prevent collisions. The basic rules applied while attempting to send a frame are described in Fig. 2.7. Moreover the different backoff values are illustrated.

Initially, if the medium has been idle for at least DIFS, previous sent frame is checked for errors. If no errors were reported, the medium must be free for at minimum DIFS. Else, a period equal to EIFS must be waited. Otherwise, if the medium is busy, the station must wait for the channel to become idle performing the *access*

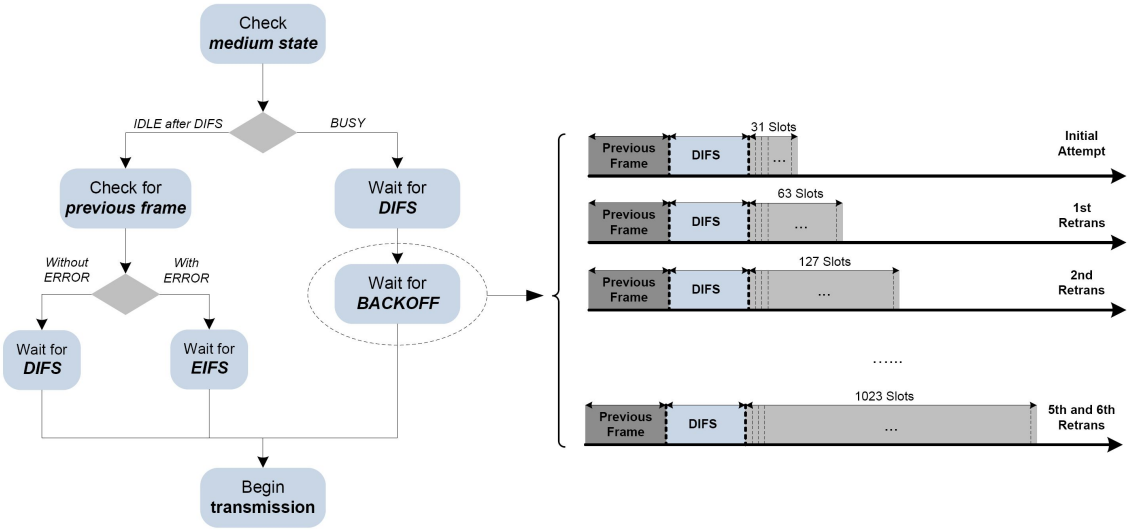


Figure 2.7 Contention-based access using the DCF

deferral, so it waits first for DIFS and then the exponential backoff is calculated. The period of time resulting from the backoff procedure is called *contention window* or *backoff window*, and it is obtained selecting a random number of fixed time *slots*. Moreover, the number of available *slots* are function of the number of retransmissions for that particular frame, referred as n in equation 2.1 as shown in Fig. 2.7.

$$slots = 2^{5+n} - 1 \tag{2.1}$$

Above presented behavior corresponds to the general rule to be applied while sending a frame, but, additional rules may be applied depending on each circumstance. For instance, ACK, CTS and fragments in a fragmented sequence can be transmitted after waiting for SIFS, incrementing their priority.

2.2.2 Enhancements to the 802.11 MAC Layer

The official IEEE 802.11 Working Group has been continuously working with the aim of improving the standard MAC Layer introduced above. Internet size grows as equal as its requirements, so inside the wireless world and after the first 802.11 standard was published, some amendments for the MAC Layer have been presented. Security and Quality of Service (QoS) issues have been the main target researches of the official team.

In the field of security, the 802.11i amendment was published in 2004 and finally incorporated into the 802.11-2007 standard version. Concepts about 802.11i will be

introduced in Section 2.4.4.

On the other hand, another improvement was introduced in 2005 by the 802.11e amendment. Therefore, some new procedures were defined in order to support local area network (LAN) applications with Quality of Service (QoS) requirements so as to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. These defined procedures concern transport of voice, audio, and video over IEEE 802.11 wireless LANs. Basically, apart from the traditional DCF and PCF defined in section 2.2.1, a new Hybrid Coordination Function (HCF) was defined providing two methods for channel accessing: HCF Controlled Channel Access (HCCA) and Enhanced Distributed Channel Access (EDCA). Both methods work using Traffic Classes in order to identify different data traffic priorities.

2.3 Physical Layer

2.3.1 Radio Frequency

At the outset, Radio Frequency (RF) concept should be defined as a frequency or rate of oscillation within the range of about 3 Hz to 300 GHz. When referring to Radio Communications, RF is used so as to confine an information-carrying signal to a narrow frequency band with as much power as possible into the signal. Naturally, inside the air medium several frequencies concurrently operate, generating possible noises in transmitted signals, so it must be ensured that their power are much greater than the offered noise.

Therefore as well as managing the physical RF limitations described above, legislative and regulatory matters should be taken in consideration. Legal authorities controlled by governmental commissions, such as the case of the FCC (Federal Communication Commission) in the United States, the European Radiocommunication Office (ERO) or the European Telecommunications Standards Institute (ETSI) inside the European Union, deal with the challenge of avoiding interference in the RF spectrum while allocating frequency usage in a fairly approach granting *Licenses* to transmitters. Licenses can restrict the frequencies and used transmission power, as well as the coverage area.

Furthermore, several frequency bands have been reserved for unlicensed use. With that aim, the FCC defined certain bands for *industrial, scientific and medical* (ISM) equipment. Table 2.1 indicates the unlicensed frequency bands inside the United States.

Table 2.1 Unlicensed Frequencies

Frequency	Description
2.400-2.483,5 GHz	ISM Band (max 4W EIRP[1])
902-928 MHz	ISM Band (Used by GSM in most countries)
5.800-5.925 GHz	ISM Band
5.15-5.25 GHz	UNII[2] max. 200mW EIRP
5.25-5.35 GHz	UNII max. 1W EIRP
5.725-5.825 GHz	UNII max. 4W EIRP

[1] Equivalent Isotropically Radiated Power (EIRP) is the amount of power that a theoretical isotropic antenna would need to emit to produce the peak power density observed in the direction of maximum antenna gain.
[2] Unlicensed - National Information Infrastructure

Radio Frequency Propagation

Extremely far from traditional fixed networks the air medium introduces several difficult-to-manage limitations while considering the propagation of data signals. The propagation of radio waves in 802.11 applications is characterized by several factors:

- Signal power is **diminished** by geometric spreading of the wavefront, commonly known as free space loss. This factor acts as a function of the distance from the transmitter.
- Signal power is **attenuated** as the wave passes through solid objects such as trees, walls, window and the floors of buildings
- The signal is **scattered** and can interfere with itself if there are objects in the beam of the transmit antenna even if these objects are not on the direct path between the transmitter and the receiver.

These factors help to introduce the *Multipath Fading* problem. It is based on the theory that waves are added by superposition, so in a specific point in the space inside the coverage area, multiple waves converge, resulting in the simply sum of any component wave. Fig. 2.8 illustrates a typical case where RF energy is radiated in every direction like an omnidirectional antenna usually applied in 802.11 devices. Furthermore Fig. 2.9 shows the worst multipath interference scenario where the sum of both received wave components (Signal A and Signal B) gives a net wave (Sum Signal) with a reduced amplitude.

2.3.2 Spread Spectrum

So as to establish secured communications and increase the resistance to interference, data signals are not directly transmitted as RF waves. The Spread Spectrum concept

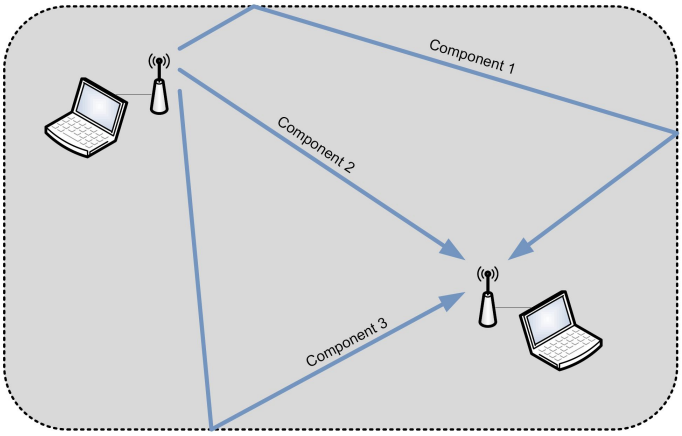


Figure 2.8 Multiple component waves converging in the receiver

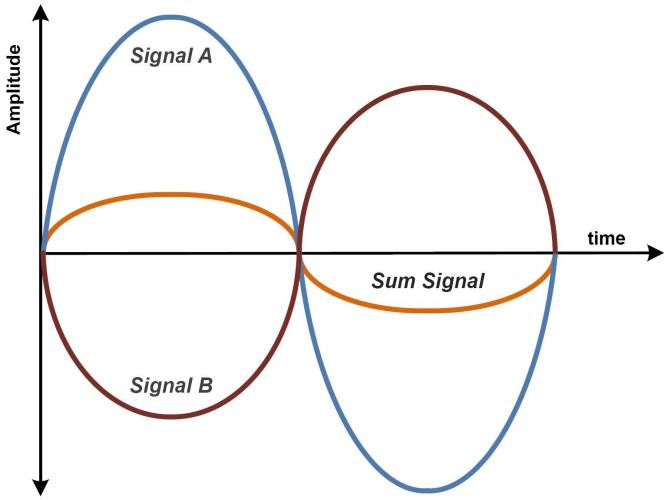


Figure 2.9 Unpleasant *multipath fading* scenario

appears as a set of mathematical techniques employed in the field of radio communications with the aim of diffusing signal power over a large range of frequencies. Thus the energy generated in a particular bandwidth is deliberately *spread* in the frequency domain, resulting in a signal with a wider bandwidth. Therefore, two inferences have to be made about the use of Spread Spectrum techniques:

- The bandwidth of the transmitted signal is much larger than the bandwidth of the original one
- The bandwidth of the transmitted signal is established by a mathematical function known by the receptor and totally independent of the message content

Spread spectrum techniques are used for a variety of reasons, including the establishment of secure communications, increasing resistance to natural interference and noise[4]. Despite of that, spread spectrum devices can interfere with any other communication systems, as well as between them. So as to minimize interference problems, the transmission power is limited by regulations.

Classification of the Spread Spectrum Inside the 802.11 ambiance different spread spectrum techniques were employed in order to define the physical layers.

- *Frequency Hopping (FH or FHSS)* - Frequency-hopping systems jump from one frequency to another in a random pattern, transmitting a short burst at each sub channel. In 1997 802.11 FH PHY was published based on this technique.
- *Direct Sequence (DS or DSSS)* - Direct sequence systems spread the power out over a wider frequency band using mathematical coding functions. Centered in this technique two different physical layers were designed. The DS PHY was published in 1997 and the the HR/DSSS (802.11b) in 1999.
- *Orthogonal Frequency Division Multiplexing (OFDM)* - This approach divides an available channel into several sub channels and encodes a portion of the signal across each sub channel in parallel. The OFDM PHY also called 802.11a was defined in 1999.

In addition to above mentioned physical layers, an infrared light physical layer was also introduced in the first specification of 802.11. Concepts about physical layer implementation will be introduced in Section 2.3.4.

[4]A special division has to be made between interference and noise in the field of telecommunications, so *noise* signals wish for disrupting communications (or prevent listening to broadcasts) whereas the term *interference* is used to describe unintentional forms of disruption

2.3.3 Physical Layer Architecture

802.11 Physical Layer is divided in two different sublayers, as shown in Fig. 2.10. The Physical Layer Convergence Procedure (PLCP) and the Physical Medium Dependant (PMD) sublayers were defined so as to work separately, the former closely to the MAC Layer and the latter narrowly to the Physical link.

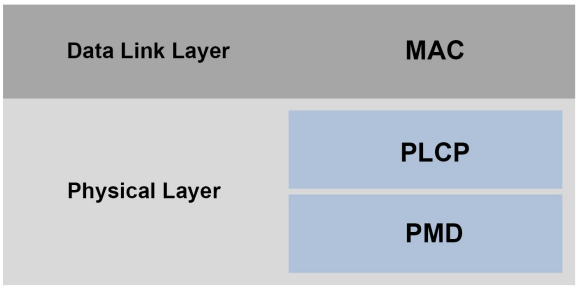


Figure 2.10 The logical architecture of the 802.11 Physical Layer

The MAC layer communicates with the Physical Layer Convergence Protocol (PLCP) sublayer via primitives or fundamental instructions through a service access point (SAP). When the MAC layer instructs it to do so, the PLCP prepares MAC protocol data units (MPDUs) for transmission. The PLCP minimizes the dependence of the MAC layer on the PMD sublayer by mapping MPDUs into a frame format suitable for transmission by the PMD. The PLCP also delivers incoming frames from the wireless medium to the MAC layer. The PLCP appends a PHY-specific preamble and header fields to the MPDU that contain information needed by the Physical layer transmitters and receivers. The 802.11 standard refers to this composite frame (the MPDU with an additional PLCP preamble and header) as a PLCP protocol data unit (PPDU). The MPDU is also called the PLCP Service Data Unit (PSDU), and is typically referred to as such when referencing physical layer operations. The frame structure of a PPDU provides for asynchronous transfer of PSDUs between stations. As a result, the receiving station's Physical layer must synchronize its circuitry to each individual incoming frame.

Under the direction of the PLCP, the PMD sublayer provides transmission and reception of Physical layer data units between two stations via the wireless medium. To provide this service, the PMD directly interfaces with the wireless medium and provides modulation and demodulation of the frame transmissions. The PLCP and PMD sublayers communicate via primitives, through a SAP, to govern the transmission and reception functions.

2.3.4 Available Physical Layers

This section introduces the main concepts about most common available 802.11 Physical Layers. Below described specifications use the 2.4 GHz frequency band excepting the 802.11a design, that exploit the 5 GHz band.

802.11 FH PHY

Frequency Hopping (FH) approach is based on changing the transmission frequency in a predetermined pseudo-random pattern. Frequency band is divided in a predefined number of slots, so the transmitter hops to another frequency during each fixed time interval, as indicated in the previous generated sequence. It has to be appreciated that both transmitter and receiver must be synchronized so as to the latter could listen the former frequency and successfully receive sent frames. Thus, each frequency slot is used for a small amount of time, called *dwell time*. In order to maximize the throughput, hopping sequences are allocated in an orthogonal way, so they do not overlap at all. This behavior is illustrated in Fig. 2.11. While generating the frequency slots, the 2.4 GHz spectrum is divided into narrow 1MHz channels, beginning in 2.400 GHz up to 2.495 GHz, so 95 different channels could be defined. Different regulations impose certain rules about the allowed channels and the hop sequences, for instance, the FCC allowed channels 2 to 79 (2.402 - 2.479 GHz) and defined a hop set size of 26 slots, so each predefined pseudo random sequence consents to hop into 26 different channels.

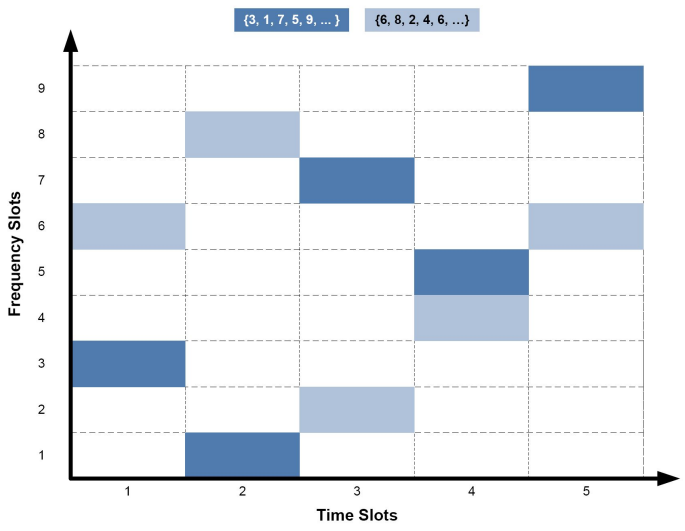


Figure 2.11 Two orthogonal hopping sequences

The effect of the interference in the FH approach is reduced because the transmitter use only a tiny portion of the frequency band to transmit. With almost 80

available channels to transmit during each hop, interference on one channel reduces the raw bit rate of the medium by approximately 1.25%.

The application of this spread spectrum technique in the 802.11 specification generated two raw data rates of 1 and 2 Mbps.

802.11 DS

Direct Sequence (DS) spread spectrum technique acted as the starting point for further adaptations of the 802.11 PHY. The first 802.11 DH PHY defined in 1997 reached the same throughput than the FH approach, but it opened the door to attain greater data rate capacities. Despite of that, DS method fall in a higher power consumption, affecting mobile equipment based on battery power supply.

The basic approach of the DS technique is to smear the RF energy over a wide band in a carefully controlled way. The original signal is processed by a *spreader*, which mathematically transforms the signal in order to achieve a narrowband input and flatten the amplitude across a wide frequency band. So then receivers monitor a wide frequency band looking for changes. The spreading process is inverted in the receiver side by a *correlator*, as shown in Fig. 2.12.

Noise implications do not effortlessly affect direct sequences transmissions. Noise tends to take the form of narrow pulses, so the correlator splits up noise across the band and the correlated signal is successfully recovered. For this reason, direct sequence modulated signals are more resistant to interference than frequency hopping ones.

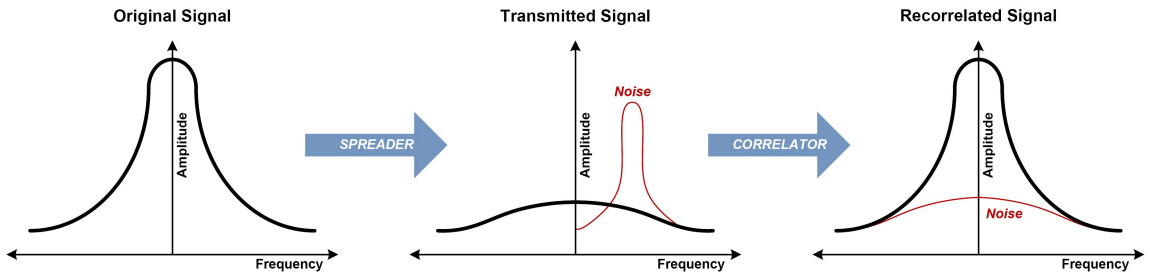


Figure 2.12 Direct Sequence Technique and the Noise Spreading[5]

Frequency band is divided in channels. These channels are much more longer than those for the FH PHY. The standard defined fourteen 5 MHz long channels, with channel 1 centered in 2.412 GHz. Regulations limit the number of available channels, for instance, inside the United States channels 1 to 11 can be operated.

[5]Inspired from [2] on page 253

In Section 2.4 channel concepts will be deepened referring to network deployment conditions.

802.11b HR/DSSS

In face of the introduction of the first 802.11 wireless approach, there was a consensus about low data rates proposed by FH and DS technique. Higher data rates were needed in order to successfully introduce the wireless philosophy as a well-founded option besides 802.3 wired networks.

To accomplish that, the IEEE Standard Board published in 1999 the specification for 802.11b networks, based on a direct sequence modulation. It has reached higher data rates up to 11 Mbps, on account of that it received the denomination of *High Rate Direct Sequence* technique (HR/DR).

Higher data rates are accomplished using an alternate encoding method rather than the baker code keying applied in 1997 specification. Complementary Code Keying (CCK) encoding function works using sophisticated mathematical transforms that use 8-bit sequences to encode 4 or 8 bits per code word.

802.11a

802.11a expanded the set of the 802.11 Physical Layers to the 5 GHz frequency band, an unlicensed range, providing more spectrum space than the 2.4 GHz bands and less overloading. It uses a 52-subcarrier orthogonal frequency-division multiplexing (OFDM) with a maximum raw data rate of 54 Mbps, but achieving realistic values closer to 20 Mbps. Despite of these greater data rates, the fact of using a higher frequency band introduces several problems, for instance signals are absorbed more readily by walls and other solid objects in their path, so network deployment requires more access point to be installed in order to obtain similar results than using 802.11b devices in the same scenario.

OFDM technique uses orthogonal sub-carriers to carry data. Data is divided into several parallel data streams or channels, one for each sub-carrier, separated by 0.3125 MHz. Each sub-carrier is modulated with a conventional modulation scheme BPSK, QPSK, 16-QAM or 64-QAM.

Basically, OFDM takes a coded signal for each subchannel and uses the inverse fast Fourier transform (IFFT) creating a composite waveform from the strength of each subchannel. Receivers on the other side, apply the FFR extracting the amplitude of each component subcarrier.

It could be considered that 802.11a standard was not introduced in the proper time within the 802.11 history. 802.11a was not widely adopted by consumers because of less-expensive 802.11b devices were already extensively adopted. Furthermore, 5 GHz operating components are much more expensive than those compatible with the 2.4 GHz spectrum. Initial products introduced in the market had operational problems. Then, a second generation of products was presented, but is was too late. However, 802.11a established the bases for the 802.11g standard, described in the following section.

802.11g

The 802.11g is actually the most widely adopted 802.11 wireless technology by consumers. It bases its operation in the 2.4 GHz frequency band and uses the OFDM spread spectrum technique. It operates at a maximum raw data rate of 54 Mbps giving about 19 Mbps of net throughput.

802.11n

802.11n amendment came to the light in 2004 and it bases on the utilization of *multiple input multiple output* (MIMO) technology and wider frequency channels. MIMO involves the use of multiple antennas at both the transmitter and receiver to improve communication performance. Moreover, instead of using the traditional 20 MHz channel bandwidth, 802.11n introduces the employment of a 40 MHz wide channel. The combination of MIMO and wider channel bandwidth fall in cost-effective approaches for increasing the physical transfer rate.

Transmitters and receivers use precoding and postcoding techniques, respectively, to achieve the capacity of a MIMO link. Precoding includes spatial beamforming and spatial coding, where spatial beamforming improves the received signal quality at the decoding stage. Spatial coding can increase data throughput via spatial multiplexing and increases range by exploiting the spatial diversity, through techniques such as Alamouti coding.

802.11n promises theoretical data rates of 300 Mbps, but far from that, speeds degrade quickly to about 45 Mbps in a 100 to 200 m range.

The compatibility problem between the new proposed scheme and previous widely deployed networks forced to take the decision of applying the 2.4 GHz frequency band in 802.11n specification. Initially, it was considered to use the 5 GHz band in order to maximize the utilization of this new technique. Furthermore, 802.11n devices could not avoid all the congestion in the 2.4 GHz band and operate full time with double-wide 40 MHz channels due to legacy devices currently operating in the same radio frequency.

Nowadays the IEEE Working Group has just approved the fourth draft of the specification with an acceptance of the 88% of the whole commission and 349 comments were resolved from the last version.

802.11y

The existing overload of the 2.4 GHz frequency due not only to 802.11b an 802.11g devices but to cordless phones and other wireless devices as well, forced the networking community to explore new unused frequency spectrum.
In 2005 the FCC started to analyse the possibility of opening the 3.650 to 3.700 GHz band for terrestrial wireless broadband operations. In may 2007 the FCC made this spectrum available for using in a broad range of new products and services, including high-speed, wireless local area networks and broadband Internet access operating equipment using contention-based protocols. Moreover, non-exclusive Nationwide 25 to 50 MHz bandwidth licenses are being conferred in the United States.
Furthermore rules allow operating at much higher power than above described physical layers, up to 20W EIRP. The combination of higher power limits and enhancements to be made into the MAC, will allow operating at distances of 5 km or more.
Contrarily to the main previous 802.11 applications, 802.11y focus on providing solutions to different target markets, like back haul for governmental wireless networks, industrial automation and control, campus and enterprise networking, last mile Wireless Broadband Access and public safety and security networks, among others.
At the moment, 802.11y Draft 10 was submitted to the IEEE-SA Standards Board Review Committee (RevCom).

2.4 Network Deployment

Deploying a 802.11 based Wireless Networks is far from being a simple task. Unlike wired ethernet networks, in a 802.11 deployment people can not directly locate stations and connect it to a backbone. Mobility requirements to be introduced in section 2.6 demand to focus on several issues; positioning access points, configuring the operating channels and securing the network are the most relevant points to weigh up. Network Deployment concepts and in particular channel allocation topics are awfully relevant for the purpose of this document. Thus following sections help to introduce 802.11 physical concepts with regard to channel allocation and operation. Analyzed theory corresponds to a *Direct Sequence Channel Layout*, since simulation results evaluated in Section 4.2 match with an 802.11b deployment.

2.4.1 Network Topology

Typically, a wireless network extends the usage of a wireline network. Thus, several access points are connected into a backbone network corresponding to a single IP subnet so a moving station has not to change its IP address. This considerations are deepen in Section 2.6.3, where Mobile IP concepts are introduced. Fig. 2.13 illustrates a common deployment scenario.

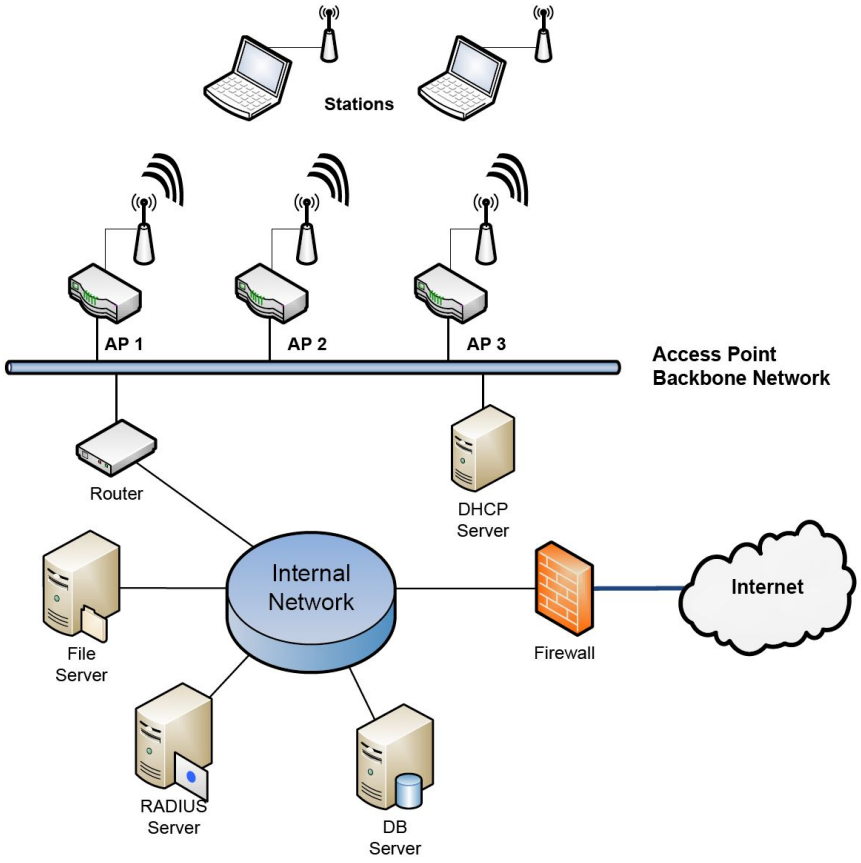


Figure 2.13 A common deployment scenario for 802.11 based networks

2.4.2 Throughput and Coverage Tradeoff

Both available and proposed physical layers focus on the *maximum throughput* and *maximum coverage* tradeoff. In this context, throughput must be considered as the average rate of successfully delivered messages over a wireless channel. With regard to *throughput*, there are four important concepts related to the maximum value, and since they will be used in this document, a briefly description is presented bellow:

- **Maximum Theoretical Throughput** - This concept is directly related to the theoretical channel capacity under ideal conditions.
- **Maximum Achievable Throughput** - Analogously to the Maximum Theoretical Throughput, the Maximum Achievable Throughput is a theoretical value. It introduces the idea of a data packet communication context and considers that control frames, hardware limitations and protocol issues affect the channel capacity.
- **Peak Measured Throughput** - This value correspond to a real measurement on a specific channel in a relative short period of time. For that reason is also called as instantaneous throughput.
- **Maximum Sustained Throughput** - This last concept considers stable measurement obtained in a relative long period and it is the most accurate indicator of the communication performance.

Coverage problem was briefly introduced in Section 2.2, when the hidden node limitation was detailed. In addition, channel throughput falls down while coming close to the coverage boundaries. In order to manage this issue, different types of antennas with several radiation patterns can be set in 802.11 devices.

2.4.3 Channel Layout

As it was defined in Section 2.3.4 different regulations allow a specific number of operating channels. Each one of them spread most part of its energy in a 22 MHz wide band, as shown in fig. 2.14. As the channel separation is 5 MHz wide, adjacent channels are interfered each others. For that reason components outside the 22 MHz band are filtered first to 30 dB bellow the power at the channel center frequency and the then to 50 dB.

It is recommended to prevent adjacent channels interference by operating only in those channels that are separated more than 22 MHz. Therefore, a separation of at least five channels prevents the undesired interference. These channels are identified as the *non overlapping channels* and their operation is illustrated in Fig. 2.15. In 802.11 deployments they match with channels 1, 6 ans 11.

The real main advantage of a non overlapping frequency deployment is extremely related with throughput. Maximum sustained throughput is reached when adjacent access points operate in non overlapping channels because interference is avoided. A typical network deployment in order to achieve this is known as the *Hexagonal Pattern Deployment*, as shown in fig. 2.16. Furthermore, access point boundaries should be carefully defined in order to prevent shared coverages areas on overlapping channels.

[6]Inspired from http://forskningsnett.uninett.no/wlan/pictures/channelselect$_d$sss.gif

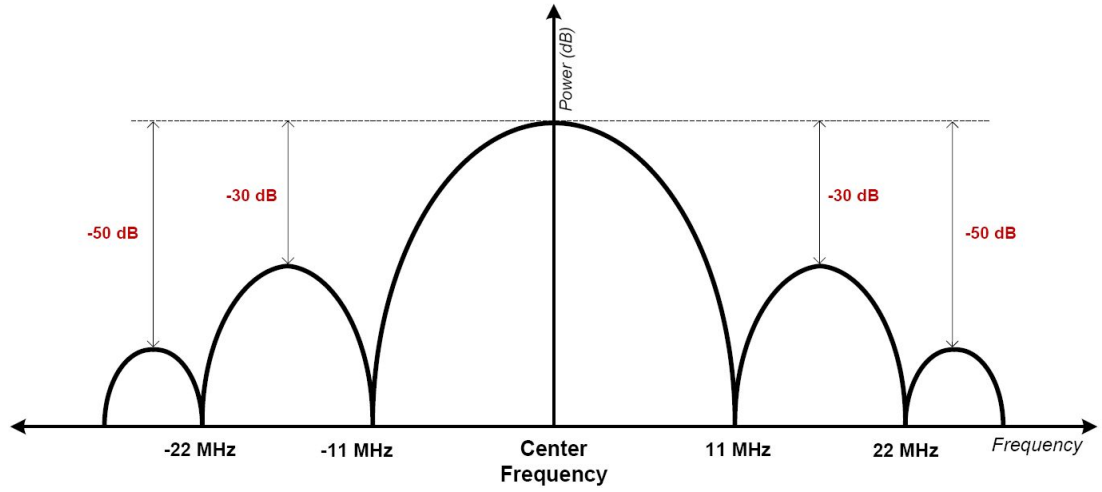


Figure 2.14 Energy spread in a single channel

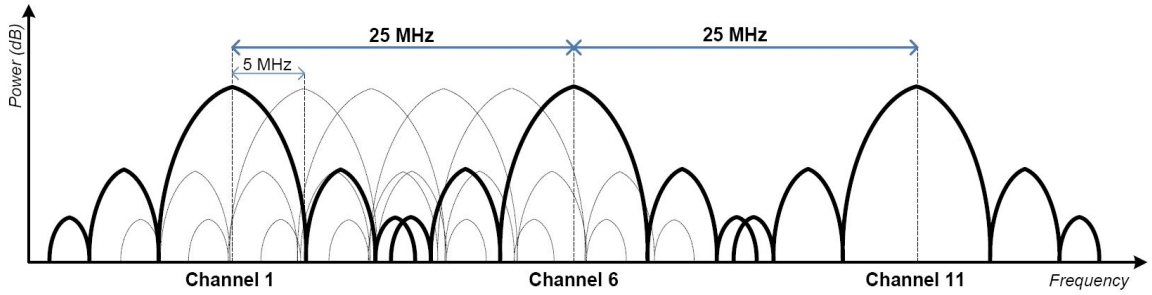


Figure 2.15 The *non overlapping channels* for a Direct-Sequence Layout[6]

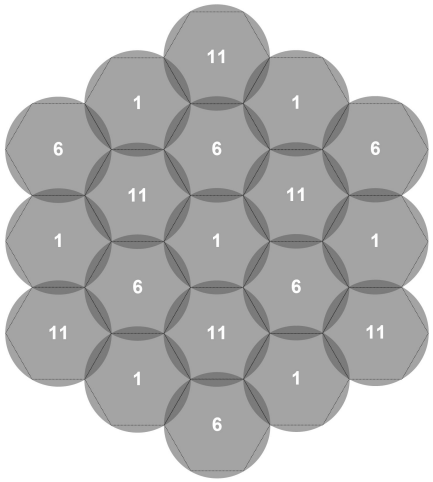


Figure 2.16 Hexagonal Deployment Pattern

The most important limitation in the channel allocation field is that only three non overlapping channels are provided, which is not enough for a two-dimensional deployment, where four frequency-independent channels are needed. Moreover in a three-dimensional deployment, like offices in a big building, overlapping may occur between different floors.

Understanding the channel allocation and operation is one of the first steps to analyze the mobility requirements and handover process that will be introduced in section 2.6.

2.4.4 Security Issues

As security features are not extremely related with the main purpose of this document, only the most important security concepts will be introduced without falling in a deep analysis.

Communications within the wireless medium are usually vulnerable to security problems. Data and control signals passing through the air are not only available to receiver devices, every wireless network interface configured on the same channel the signal is being sent may be able to capture the information contained on it.

Common security violations are due to device theft, denial of service, malicious hackers, malicious code, theft of service, and industrial and foreign espionage. Therefore, security objectives concern ensuring confidentiality, integrity and authenticity in the whole wireless deployment.

In 1999 the IEEE published a set of standardized security services, provided by the Wired Equivalent Privacy protocol (WEP) so as to protect link-level data between clients and access point. WEP uses stream cipher RC64 for confidentiality and the CRC-32 checksum for integrity issues. Therefore, in order to achieve authentication, WEP applies both Open System and Shared Key approaches.

As soon as the IEEE standardized WEP protocol for the 802.11 environment, several problem were encountered and the algorithm was declared as deprecated. Thus as a result of 802.11i Task Force work, WPA and WPA2, also called RSN (Robust Security Network), appeared in the market.

It has to be considered that a differentiation between lower and upper layers security protocols is necessary. The incorporation of IP security (IPSec) or Transport Layer Security (TLS) protocols help to reinforce the protection of the whole network. However due to the fact that lower layers send and receive control packets to perform management operations like those to be detailed in Section 2.5, 802.11 protocol suite must maintain and fortify security subjects itself.

2.5 Management Operations

The complexity of the wireless medium requires a set of management operations in order to increase 802.11 networks reliability. Predominantly, the fact of the station being in continuous movement introduces several requirements so as to look for access points to authenticate and associate, to conserve the mobile station power and timing synchronization issues among nodes.

The 802.11 standard established a Management Architecture in order to successfully achieve these operations. Both MAC and PHY Layers deal with useful centralized information contained in a database structure termed *Management Information Base* (MIB). The MIB consists in a hierarchical data base of entries addressed by objects identifiers expresses in Abstract Syntax Notation (ASN.1). These entries contain information about four main different domains, as indicated bellow:

- **Station Management** - It refers to global configuration parameters related to scanning, authentication and association processes and WEP keys management among others.
- **MAC Layer Management** - It contains configuration parameters about the MAC, like the MAC address and fragmentation parameters.
- **Physical Layer Management** - Comprising Physical Layer information for each spread spectrum technique as defined in Section 2.3.4 like the hop time for FH, the current frequency for OFDM or the current channel for the DSSS approach.
- **Resources** - Containing information related to available resources.

2.5.1 Management Frames

All available management operations founds on frame's exchanging over the wireless medium. The 802.11 standard defines a set of *Management Frames* in order to perform these services.

- **Beacon** - Access points periodically send *beacon* frames in order to advise stations about their presence in the BSS area. It contains all the required information to associate to it.
- **Probe Request** - A station may require to connect to an access point, so the *Probe Request* frame allows it to ask for beacons instead of waiting for them. It contains the supported data rates, so only compatible access points may be able to respond.
- **Probe Response** - They act as the Probe Request answer, containing essential information so as to connect to the replier access point.

- **Authentication Request** - Station must validate its identity before associating with an access point. The main important fields of this frame are the specific authentication algorithm, the sequence number, a status code and a challenge text.
- **Authentication Response** - As different authentication algorithm may be applied, the *Authentication Response* could be divided in a set of different frames, as indicated in 2.5.3. If the authentication is not successfully performed a *Deauthentication* frame is sent containing the reasons.
- **Association Request** - In order to finally join the access point, the station request to associate by sending an *Association Request*
- **Association Response** - The access point uses this frame in order to inform the station about the association identifier so as to start data frames communication. If the association is not successfully performed a *Deassociation* frame is sent containing the reasons.

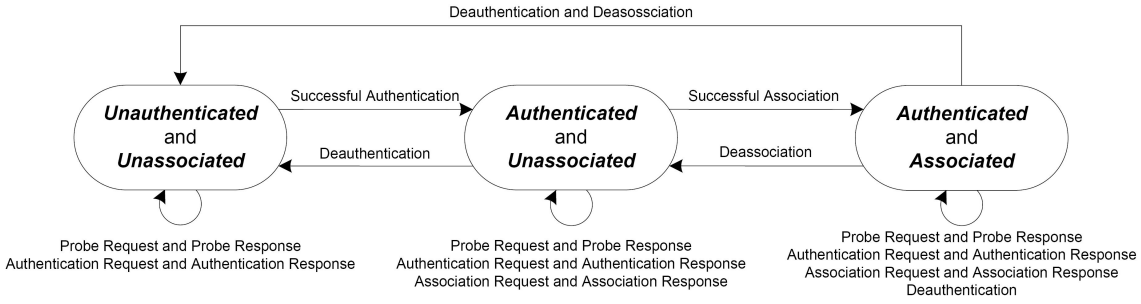


Figure 2.17 Station-AccessPoint relationship and Management Frames

Management frames exchanging helps to deal with different states within the relationship between stations and access points in an infrastructure network. These stages are presented in fig. 2.17.

2.5.2 Scanning

The scanning process refers to the fact that a station must look for available access points in the coverage area so as to establish a connection and start data communication. Both *scanning* and *discovery* terms will be used to describe this process. The 802.11 standard defines a default scanning process using a set of parameters and information contained in the MIB. These variables help to define the discovery context in order to successfully find an access point.

Firstly, the nature of the network must be considered, *BSSType* variable describes if the BSS is an infrastructure, ad hoc or both. Then, a specific BSSID (Basic Service Set Identifier) or SSID (Service Set Identifier) containing the desired network name to connect allow filtering the discovery process to that particular network, in this case, the station usually try to find any access point in a range, so they are set as broadcast addresses.

As different access point may be operating in different frequency bands inside the 2.4 GHz spectrum, the number of channels to be scanned could be set as well, in order to perform *Selective Scanning* or *Full Scanning* processes.
Moreover, as it will be further detailed, discovery process mainly consumes time, so time related variables are defined so as to decide how many time is to be spent in each channel. These parameters are listed above.

- **ProbeDelay** - Describes the time that a station spends before probing a channel[7].
- **MinChannelTime** - It is the minimum amount of time to spend when probing a channel.
- **MaxChannelTime** - It is the maximum amount of time to spend when probing a channel.

Finally, the standard defines two singular scanning approaches, acting with dissimilar behaviors in order to accomplish the main goal of finding an access point. Both Passive and Active scanning are to be introduced.

Passive Scanning

The standard defines that an operating access point must periodically send a *beacon* frame, as detailed in Section 2.5.1. Thus, a station should change its operating frequency in order to receive and process beacon frames. The interval an access point waits to send a periodical beacon is defined in the MIB as *BeaconPeriod*, inside the Station Management domain. Fig. 2.18 illustrates the Passive Scanning approach.

It has to be considered that *BeaconPeriod* timer should be carefully set. High values may allow stations to have access points information soon, but on the other hand, as the station must change its channel frequency, data transmission is temporally interrupted and the average throughput falls down sooner. Usual values for *BeaconPeriod* in the 802.11 environment is 100 ms.

[7]To **Probe** a channel refers to the fact of sending Probe Request management frames so as to obtain information from access points

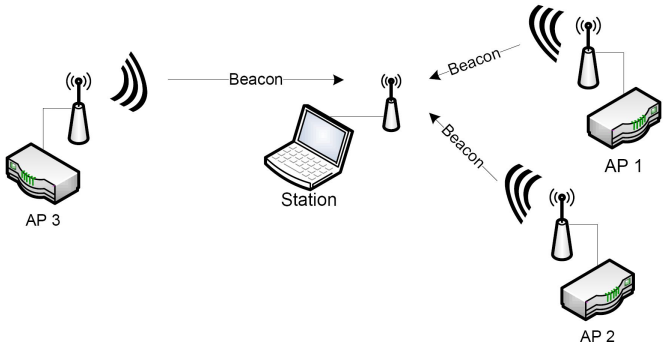


Figure 2.18 Passive Scanning - The station simply waits for *beacons*

Active Scanning

The Active Scanning approach allows a station to look for an access point to associate in a more proactive way than that defined in the passive method. Probe Request management frames are broadcasted in order to receive immediate response from the access point, contained into the Probe Response management frames. The standard specifically defines a basic algorithm to be performed by stations while discovering access points, as defined in Algorithm 1 and illustrated in fig. 2.19.

Algorithm 1: Active Scanning Algorithm

```
0.1  forall channel do
0.2      Tune the station in channel;
0.3      Wait for activity or ProbeDelay expiration;
0.4      if ProbeDelay expired then
0.5          channel is empty, go to next channel;
0.6      end
0.7      if activity detected then
0.8          station gets access to medium;
0.9          station broadcasts a ProbeRequest;
0.10         station waits for MinChannelTime;
0.11         if no activity detected then
0.12             go to next channel;
0.13         end
0.14     end
0.15     if activity detected or ProbeResponse received then
0.16         station waits for MaxChannelTime;
0.17         repeat
0.18             process all ProbeResponse;
0.19         until MaxChannelTime expires ;
0.20         go to next channel;
0.21     end
0.22 end
```

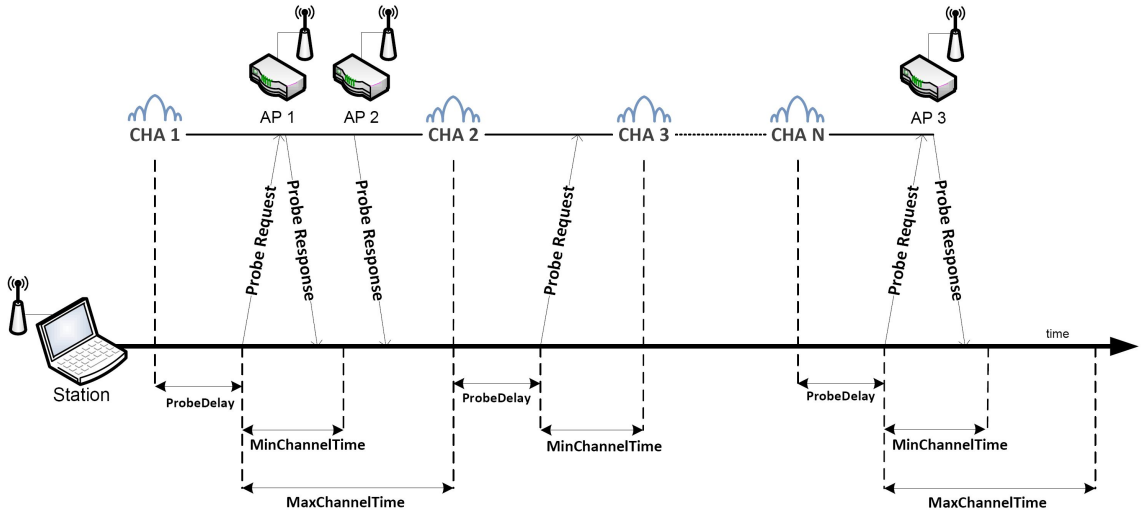


Figure 2.19 Active Scanning - *Probe Requests* and *Probe Responses*

After scanning all available channels, a scan report is generated containing information about all candidate access points to associate with, for instance, physical and timing parameters and supported data rates by each access point.
The optimizations proposed in this document focus on the Active Scanning approach, thus section 2.6 will take back and deepen these concepts to propose the solution.

2.5.3 Authentication

Subsequently, when a candidate access point is available to connect with, the authentication process is performed. Authentication is a security matter and it allows the access point to accept or deny a particular station to associate with it. As described in Section 2.4.4, the standard provides both Open System and Shared Key authentication schemes. The former provides an identity independent authentication, thus an *Authentication Request* frame sent by the station is replied by a *Authentication Response* by the access point with no further identity verifications. In contrast, the Shared Key method uses a WEP key (or further standardized keys as WPA and WPA2). In this scheme, the station sends an *Authentication Request* so the access point send a second frame containing a *challenge text* field. After being received by the station, it constructs a new frame encrypted by the key. The access point finally receive the frame and decrypts it using its own key. If integrity is verified, the access point confirms the approval to associate sending a last *Authentication Response* with a successful status code.

2.5.4 Association

Association process guarantees the station full access to the distribution system. The main purpose of this process is to explicitly identify the station by a unique identifier on the access point side, so the latter will be able to buffer frames targeted to the former while it is in a Power Saving Mode. The station sends an *Association Request* management frame to the access point, then if accepted by the access point, it informs the station about the Association ID by sending an *Association Response*. After this, data traffic transmission can start immediately, so frames directed to the station being received from the ethernet interface in the access point are bridged to the wireless interface and sent to the station.

2.6 Mobility Requirements

Initially, with the introduction of the first laptops in the market in the 80's, the *mobility* concept was not such a requirement. As an instance, the first laptop developed in 1981 weighted more than 4 kg and batteries were an optional feature. It is more than obvious that *mobility* was not a main market necessity like as *portability*. By portability it should be understood the action of placing the hardware in different

geographic positions so as to work in diverse offices or physical contexts. Moreover, in the early 80's wireless networks were not widely adopted, so laptop developers did not focus on network connection matters.

During the 90's several researches and further technological improvements were accomplished in the field of battery power supply and portable computer-based devices. Then, with the introduction of the first 802.11 set of protocols in 1997, laptops started to implement wireless network interfaces both in built-in or external mode. Wireless networks were widely and fast developed after the presentation of 802.11b and further 802.11g protocols, providing a relative low-cost bridge to existing wired networks and the Internet.

Additionally, laptop sales growth up rapidly, obtaining the first position against desktop based computers in 2005 while reducing the gap *Price vs. Capacity* in comparison with the latter ones. While some years ago users had to pay more than USD 1.000 for a basic featured laptop, today they should disburse no more than USD 400. Nowadays, the sales rate for laptop computers is doubling the desktop sales rate and this difference should be maintained and increased during the following years.

Furthermore, in the last years the *convergence* concept between different mobile devices has been taken in consideration. Not only laptops focus on *mobility*, but handhelds and smart phones as well. Several devices are providing different wireless connection natures, those provided by 802.11 and cellular based interfaces as well (for instance: 3G[8]), among others.

Therefore, the market has been migrating to mobile devices not only because of price reductions and economical convenience, but as a consequence of changes in the condition they perform their tasks as well. Customer necessities require *mobility*, they should be provided all around the world and with no delays, so companies are obligated to adapt their actions focusing on *mobility*. In the last years *mobility* comes up as a necessity and no more as a simply feature.

2.6.1 The Handover Process

The concept of *mobility* is defined by a mobile device user *moving* around a defined area without breaking the connection between the hardware and a network.

As it was introduced, *mobility* is a necessity, but in order to be successfully performed, it has to consider the widest frontiers as possible. To achieve this, some

[8]Third generation of mobile phone standards and technology, providing high data rate connections depending on the implemented protocol

possible solutions may be applied. The first one is performed by increasing access points and mobile stations power, which concludes with a higher consumption of resources and becomes a vicious circle while the mobility concept is being limited itself because the user will not be able to move for a long period of time due to out-of-battery constraints.

Another solution is to deal with the *roaming* process, that allows users to move on a wide area, covered by multiple access points. This process requires *Handover* mechanisms based on concrete algorithms, so as to manage the migration from an old access point to a new candidate one, focusing on minimizing the disconnection time of the mobile station and avoiding non desired effects in the upper-layers such as application and services. Fig. 2.20 illustrates this scenario, where a mobile station passes across three different Basic Set Services while being connected to the same network.

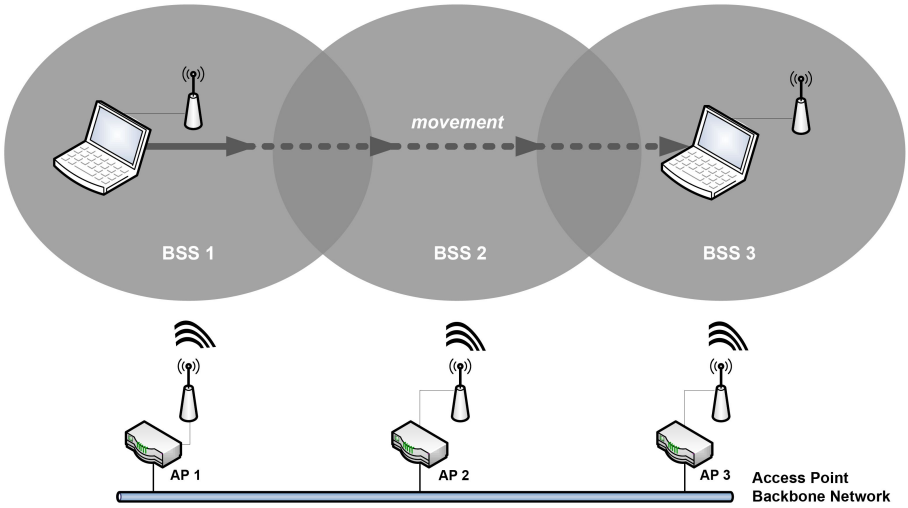


Figure 2.20 Mobility Scenario

This situation is strictly related to the introduction given in Section 2.4, the roaming decision depends on the network deployment. Basically, successful transitions between access points are related with the overlapping areas between them. Greater overlapping areas increase the success rate, while adding interference if operating channels are different from the non overlapping.

The *Handover* process should be defined as the set of actions with the aim of looking for, authenticating and associating to an access point because of a possible or concrete disconnection from a previous access point. Handovers may be classified in two different categories:

- **Horizontal Handover** - This class of handovers involves mobility between the same network interface technology or simply the same network type. For instance, the mobility situation described in fig. 2.20 should be considered a *Horizontal Handover* if all the access points employ the same technology, as an example 802.11b.

- **Vertical Handover** - On the other hand, *Vertical Handovers* concern different network interfaces technologies within the mobile node. As an instance, a mobile node could associate with a cellular link (as a 3G network) when leaving a 802.11 coverage area. Vertical Handovers implement decision algorithms so as to connect to the best technology available in any particular case.

This document will focus on the first handover class because the particular case of a 802.11 handover approach is being studied. Therefore, within a *Horizontal Handover* another important division regarding to the network layer it affects has to be made. Layer 2 and Layer 3 handovers will be detailed in the following section.

2.6.2 Layer 2 Handovers

Also referred as a *Link Layer Handover*, a Layer 2 handover essentially consists in a Basic Service Set transition, like as illustrated in figure 2.20. Supposing that network nodes implement IPv4 or IPv6 network protocol, in a Layer 2 handover, all BSSs should be configured with the same IP subnet mask, so as to avoid changes in the IP addresses inside the nodes. Applying these constraints, no effects in the upper layers should be suffered, because the handover process is managed by the MAC layer. However, depending on the applied strategy and the context for the handover, some disconnection times could appear, so all layers are affected. This concepts will be developed in the following sections.

Layer 2 Handover Phases A Layer 2 Handover is performed using the Management Operations described in section 2.5. As described in fig.2.21, a mobile station starts moving from the old access point to a new candidate access point (AP1 to AP2). While moving (T1), the mobile station detects low signal strength from data packets coming from the old access point, so it starts performing the discovery process as defined by the scanning algorithm [9] (T2). After finishing the scanning process, the mobile station should take the decision to *reauthenticate* between all candidates detected access points. The *reauthentication* process differs from the *authentication* operation described in Section 2.5.3. The former corresponds to a situation where the mobile station has not a previous access point, while the latter is related with the fact of leaving an old access point when attempting to connect to a new one. Thus, after performing reauthentication, the mobile station attempts to *reassociate* (T3) with the

[9]For the purpose of this work, an Active Scanning approach must be considered when talking about the standard Discovery Process

new access point in order to start data packet communications. During the reassociation phase, the new access point must verify previous association with the old access point exchanging some frames, if satisfied, it will decide to allow the mobile station to associate, depending on some variables as the number of attached mobile stations and other traffic conditions. If it grants the association, the association identifier is sent to the mobile station. Before attempting to send traffic using the new access point, the mobile station will receive packets buffered in its old access point. These packets are obtained by sending a *Handover Request* from the new to the old access point, which delivery them adding a *Handover Response*. Both Handover Request and Response are Inter Access Point Protocol (IAPP) frames. IAPP is introduced in section 2.7.3.
In summary, we have identified three differentiated stages: **Discovery**, **Reauthentication** and **Reassociation** Phase.

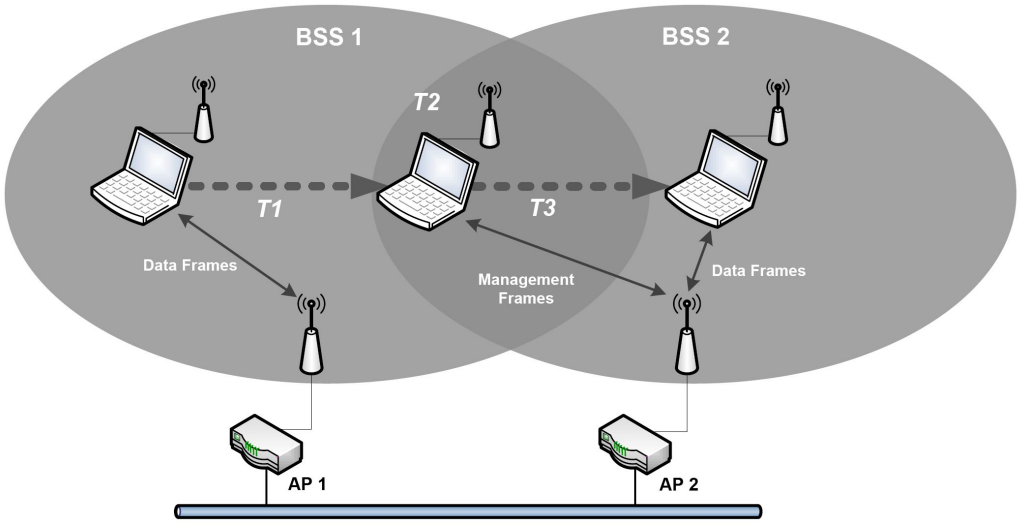


Figure 2.21 Layer 2 Handover

Layer 2 Handover Latency Within a handover process, *time* is the most relevant variable to be taken in consideration. Depending on the applied strategy to perform the handover, different delays will occur. The concept of Handover Latency is related to the period of time between the first *Probe Request* sent in the first scanned channel and the reception of the *Association Response* management frame by the mobile station.

Therefore, the whole handover latency could be also divided in three different latencies, related with the handover phases. First, the *scanning latency* is must be described as the period of time used to discover all candidates access point. It starts

when the first probe request is sent on the first channel and it finishes after the last channel is probed. The scanning latency is extremely correlated with the activity on each channel. Both *MinChannelTime* and *MaxChannelTime* define the whole delay, as shown in equation 2.2. L refers to the scanning latency, c refers to the channel number, function $P(c)$ refers to the probability of finding an access point on each channel.

$$L = \sum_{c=1}^{MAX\_CH} (1 - P(c)) \cdot MinChannelTime + P(c) \cdot MaxChannelTime \qquad (2.2)$$

Thus L is calculated as the sum of delays on each scanned channel. For channels with activity detected the mobile station will wait for $MaxChannelTime$, otherwise it will just wait for $MinChannelTime$ expiration.

Both reauthentication and reassociation delays, there are not time control variables. Thus, delays are only related with management frame transmissions during each phase.

As explained in [3] and [13], concrete measurements of the handover latency applying the Standard Active Scanning Algorithm (Algorithm 1) demonstrate that during the 90% of the total handover latency, the mobile station performs the Discovery Phase. Then, if optimizations focus on this particular stage, the handover cost could be reduced. Moreover [13] shows how different hardware manufacturers implement different techniques and variables values, so then handover latencies greatly variate. Common values for L are generally located above a hundred of milliseconds.

There is a significative difference between the *Handover Latency* and the *Disconnection Time*. During the Handover Latency, the station could send and receive both Management and Data frames, so it cannot be considered as disconnected. The Disconnection Time is only related with a portion of the Handover Latency where the mobile station is no more attached to an access point, so any kind of connectivity is lost.

2.6.3 Layer 3 Handovers

During a *Layer 3 Handover* process a mobile station moves to a new coverage area where a change of access router subsequent to a change of access point will take place. Thus, the handover also produces effects on the upper layers, because, in this case is mandatory that the mobile station changes its own IP address in order to establish a connection with the new access point. Applications and services using any transport protocol, like User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) will be interrupted until the new IP address is configured for the new

link. Fig. 2.22 illustrates a typical Layer 3 Handover scenario. A mobile station moves from AP1 to AP2. Both AP1 and AP2 implement IPv4, the former is attached to the subnetwork 192.168.1.XXX, and the latter to 192.168.2.XXX. Thus, when associating with AP2 a new IP address must be configured in the mobile station (192.168.2.2).

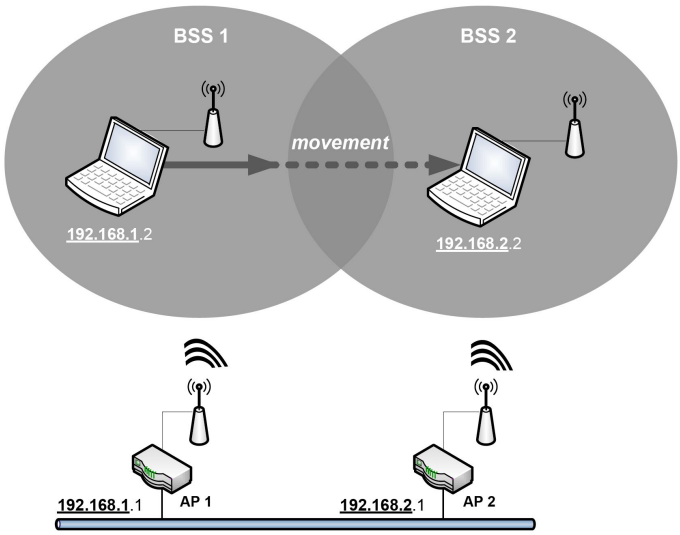


Figure 2.22 Layer 3 Handover

As it was described, the 802.11 standard defines a unique MAC layer and different PHY layers, giving a set of Management Operations to deal with the MAC layer mobility, but it does not implement any mechanism to handle mobility into the IP Layer. For that reason, concepts about IPv6 Protocol and further Mobile IP will be introduced so as to describe how to manage handover processes concerning IP address modification.

IPv6 Network Protocol

When talking about data frames delivery over a packet switched internetwork, the Internet Protocol (IP) emerges as the most widely deployed protocol while acting as the prevailing network layer protocol on the Internet. However, since its introduction in 1981 and considering the expansion of the Internet, an imminent exhaustion of the *Address Space* and the *Global Routing Table Growth* come into view as the most important challenges to solve in the Internet domain.

Considering the address space problem, one of the main factors producing the shortage is related with the efficiency of address assignment (H factor) with real values between 14% and 26%, which indicates the big defficiency in the assigment. Moreover it is estimated that IPv4 address space will be fulfilled between 2009 and

2011 depending on the Internet growing speed. Before considering the implementation of a new IP protocol some mitigation factors were introduced, reclaiming assigned Class A Addresses and increasing assignation costs. Far from solving the main problem, these measures only increase the shortage deadline.

Therefore, Network Address Translation protocol (NAT) has been widely implemented so as to map multiple internal IP addresses to a single external, because of the address shortage. Thus, inside an organization network, the utilization of NAT acts as a single point of failure.

With the introduction of IPv6 in 1998 [6] a great number of features were presented in order to solve IPv4 deficiencies and provide an enhanced tool so as to manage the mobility aspect. Some of them are itemized bellow:

- **Larger Addresses** - 128 bits addresses were defined, so NAT could be avoided and end-to-end capabilities would be restored.
- **More levels of addressing hierarchy** - Providing better aggregation of routes, easier allocation of addresses and scalability of the routing table.
- **Multiple addresses on an interface** - Enabling multiple uses, virtual hosting and multihoming.
- **Fixed address architecture** - Decreasing network management costs.
- **Neighbor Discovery** - Providing an efficient use of the link
- **Auto configuration of nodes** - Based on advertisements about link addresses sent by the router. Link-layer (MAC) address is part of the IPv6 address, enabling fast and reliable configuration of nodes and easy renumbering.
- **Address conflicts on links are solved** - Embedded MAC address guarantees uniqueness of the IPv6 address on the link.
- **Multicast address scoping** - Multicast is easy-to-manage since the scope of the channel is inside the address.
- **Simpler and efficient IPv6 header** - Routers process packets faster, improving forwarding performances.
- **Extension headers** - Extra information could be added incrementally in an IPv6 packet without impact.
- **Mandatory IP Security** - IP Security (IPsec) is mandatory so it makes all nodes in a position to secure their traffic.

- **Labeling flows for QoS** - Making more efficient Quality of Service (QoS) processing.
- **Private but unique address space** - Making easier to connect private networks together.

One of the most important enhancements is related with the Neighbor Discovery Process [4] and Stateless Address Auto Configuration [5]. The former allows nodes to discover each other's presence, to determine each other's link-layer addresses, to find routers and to maintain reachability information about the paths to active neighbors. Features included in the latter allow nodes to configure addresses themselves as soon as they are connected to the link; first, they concatenate the prefix of the network with the host part of the address that embeds node link-layer address.

Under the scope of this thesis, a detailed implementation of IPv6 Neighbor Discovery and Stateless Address Auto Configuration mechanisms under a Wireless Network Software Simulator was performed. This implementation acted as the first step to get immersed within the simulator context (presented in Section 4.1.3) and to provide a tool to emulate communications on a IPv6-based network.

Mobile IP

Up to this section, concepts about mobility and IPv6 were independently introduced. However, it should be considered that as most part of actual deployed networks implements the Internet Protocol (IPv4 or IPv6), the mobility concept is narrowly associated with IP-based networks.
For that reason, in the IP context, *IP Mobility* should be defined as the case where a node changes its IP address while moving. Therefore the networking community introduced in 1996 the first approach to IP Mobility Support [7] and several modifications have been successively done after that in order to arrive to the last mobility support protocols for both IPv4 [8] and IPv6 [9] network layer implementations, designed to sustain IP connections while IP addresses change.
When a mobile node moves, several problems appear. Packets will be not forwarded to the new destination and they will be dropped. In the meantime, packets sent to the older address will be lost. Then connections between mobile node's applications will be terminated. Mobile IP allows a node to be reachable, offering them a stable address.
Basically, the main Mobile IP purpose is for a mobile node to be known by its peers with a permanent address and so an agent is mandated to forward the traffic to the current location. In order to define an IP mobility scenario, some concepts should be introduced:

- **Mobile Node** - A node that can change its point of attachment from one link to another.

- **Home Address** - A permanent IP address for the mobile node.
- **Home Agent** - A router on the home network.
- **Care-of-address (CoA)** - Represents the current location of the mobile node
- **Correspondent Node** - A node that communicates or corresponds with the Mobile Node
- **Home Link** - A particular link where the home address prefix is assigned
- **Foreign Link** - Any different link from the home link where the mobile node moves to.

Thence, the basic Mobile IP process is described by the following sequence and illustrated in fig. 2.23.

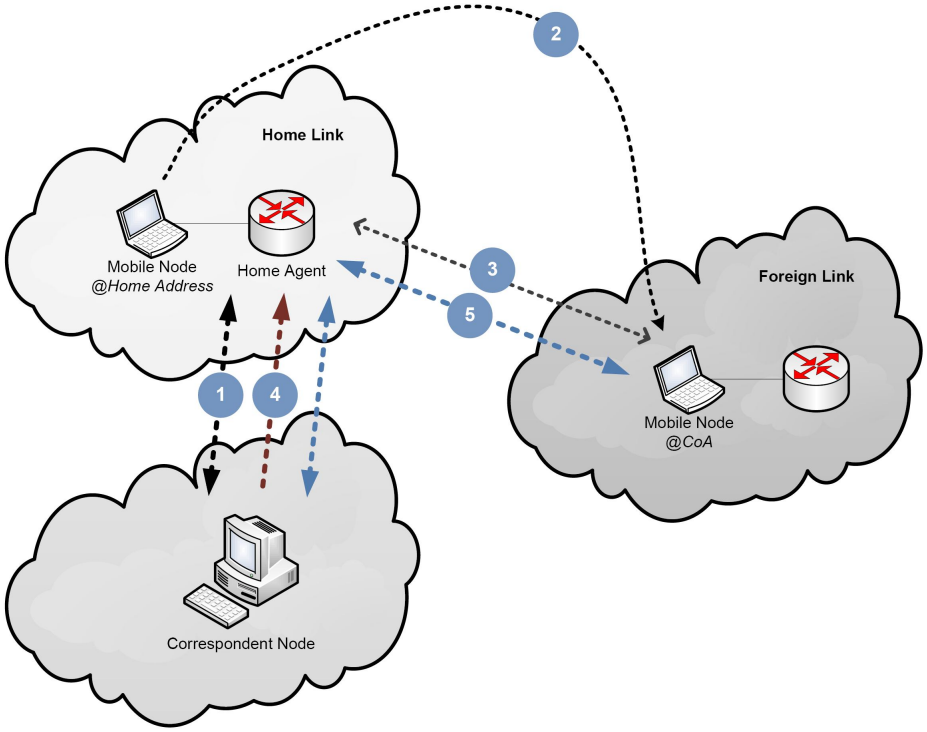


Figure 2.23 Mobile IP basic behavior

1. While the mobile node is located in its home network, it uses its permanent address (Home Address). If a Correspondent Node sends a datagram to the mobile node, it sends it to the home address. The mobile node normally receives the packets from the correspondent node.

2. The mobile node moves to another network, acquiring a temporary IP address (CoA).

3. This new CoA is registered by the mobile node in the home agent.

4. The correspondent node sends more packets to the mobile station, using the same permanent address.

5. Home agent intercepts and forwards the packets to the mobile node. When the mobile node sends a packet to the correspondent node, it uses the home address as a source and then the home agent directly sends the packets to the correspondent node.

In this case, a correspondent node does not have any knowledge about mobile node mobility, so the home agent takes care about communication between them. However another approach known as *Route Optimization* allows the correspondent node to directly send packets using mobile node's CoA, thus the home agent is not considered in this case.

Mobile IP Handover In a wireless development using IP as network layer a node moving out from its current access point coverage area will initiate a handover process. As described in Section 2.6.2, the mobile node should start scanning for access points to associate with, but in this case, all candidate access points have a different subnet mask configured and variations in the mobile node's IP address will take place.

Two different lines of attack may be described when performing a network layer handover. In a *Break-Before-Make* handover, a mobile node disconnects from its current link and then reconnects to the new link, so synchronization between node's transceiver and access point's transceiver is lost. On the other hand, during a *Make-Before-Break* handover, a mobile node establishes the new link while maintaining the old one.

So as to support Mobile IP handovers, [9] introduces a set of frames, as *Binding Update* and *Binding Acknowledgement* and a set of conceptual data structures as the *Binding Cache*, the *Binding Update List* and the *Home Agents List*, that in addition to other IPv6 frames helps to successfully perform a layer 3 handover.

Mobile IP Handover Latency In addition to layer 2 handover latencies, during a layer 3 handover the mobile station must inform all correspondent nodes and the home agent about such change in its CoA. Thus several neighbor discovery frames, as Router Solicitations and Advertisements are delivered. As described in [10], the most important part of this latency is related to the *Duplicate Address Detection* process

(DAD) in the new link. As the duplicate address probability is such a small value[10], an *optimistic DAD* is proposed as a possible optimization, so latency should be reduced by assuming that a mobile node can use its new address while performing DAD.

A layer 3 handover should totalize a latency from 60 ms to 250 ms [11], so it impacts on TCP and UDP traffic. Application and transport layers will view this delay as an indication of a disruption on the network availability.

2.7 Fast Handover Approaches

2.7.1 Overview

After introducing the layer 2 and layer 3 handover concepts in the 802.11 ambiance, layer 2 handover optimization processes will be presented.

Further research focused on reducing the duration of the layer 2 handover duration, with the scanning process as the focal point.

A brief introduction to the Fast Handover concept and then particular implemented mechanisms will be evaluated.

2.7.2 What is a Fast Handover?

In the field of networking the *optimization* concept is usually referred to modifying defined standard operations to make them work more efficiently using fewer resources. In a wireless approach like 802.11, the set of operations to optimize are related with the *coverage and throughput tradeoff* (Section 2.4.2). Thus, coverage is a fact of *power* resources and throughput deals with *time* resource. Optimizing the usage of *time* resource means that in a communication process, the number of management frames relative to data frames should be reduced to the minimum as possible. On the other hand, as *power* is also limited, if a large coverage area is needed, several access points are positioned considering overlapping areas so as to provide *mobility*.

Joining these two concepts in the mobility domain, mobile stations should initiate handover processes while changing access points *coverage* areas, so reduced *time* delays are required so as to avoid effects in upper layers.

Under this point of view, the *Fast Handover* concept appears with the aim of optimizing the whole handover process. Thus, several approaches were defined so as

[10] In [10] a probability of 10^{-12} is considered

to reduce the *time* used by Management Operations during a handover.

Basically, actual studies focus on the reduction of the scanning or authentication latencies applying different possible strategies. These strategies could be divided as described in fig. 2.24. Moreover, some fast handover proposals suggest to combine different strategies from each domain.

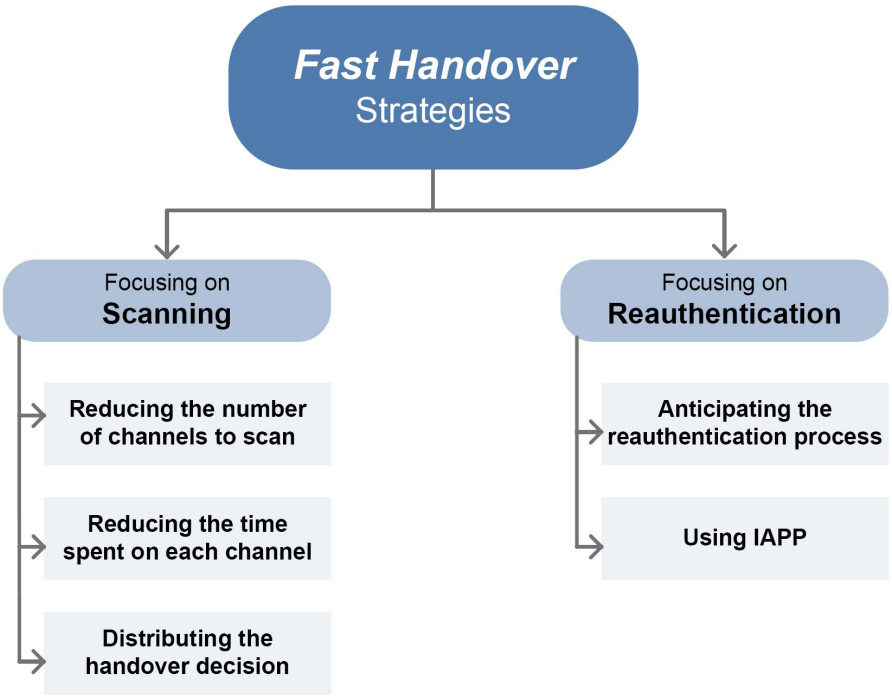


Figure 2.24 Different lines of attack to reduce the handover latency

As several particular studies have been presented applying these strategies, they are being presented and analyzed in the following sections, complying with the conceptual division of figure 2.24.

2.7.3 Fast Layer 2 Handover Mechanisms

Reducing the number of channels to scan

As it was detailed in Section 2.5.2 the 802.11 standard defines an Active Scanning Algorithm. Its behavior is also know as *Full Scanning*, because all allowed channels are probed for activity, so as to find a candidate access point. One of the most trivial ways to reduce the scanning process delay is to avoid probing some allowed channels,

introducing some constraints and managing some data structures containing useful information about the presence of access points on each channel.

Selective Scanning and Caching In [14], authors introduce an alternative technique to the traditional active scanning method. It is based on the utilization of a *channel binary mask* so as to take the decision about which channels to scan. Thus, only some preselected channels will be scanned for access points to associate with, contributing to the scanning latency reduction.

As illustrated in fig. 2.25, a mobile station starts moving (T1) and goes out from its current access point coverage area, operating in channel 8. During the first scanning process (T2) a common full scanning is performed, so all the channels are probed. This can be appreciated inside the binary mask, because is set as "1" in all its positions. While scanning, the mobile station starts building a new channel binary mask, containing a value of "1" for channels were a probe response was received (channels 2, 4 and 1) and a value of "0" if no access points answered to the request. Additionally, the mask is turned on for the non overlapping channels (See Section 2.4.3), because they are considered as highly probable used. Then the station should authenticate and associate to the best candidate access point, so once associated, the current channel is turned of from the mask, since is low probable to find a neighboring access point on the same channel.
Once the mask is finally built and a second handover is up to occur (T3), the Selective Scanning approach defines the following steps to discovery candidate access points:

1. Probe channels which binary mask reference is set as "1".
2. If some access point is detected, then compute the new mask and reassociate.
3. Otherwise, if no probe responses were received on those channels, logically invert the mask values and continue probing these new channels.
4. If some access point is detected in these channels, compute the new mask and reassociate.
5. Elsewise, execute a standard full scanning process again.

Furthermore authors defined an extra technique to be combined with the Selective Scanning approach. The *Caching* method consents the mobile station to built and maintain a cache table containing the link-layer (MAC) address of best adjacent access points to associate. This table is keyed by the current access point as shown in fig. 2.26. It has to be considered that the number of MAC address stored for each access point should be limited to a fixed amount.

The combined mechanism works as follow. When the mobile station associate with an access point, the MAC address of the latter is included in the Cache as a

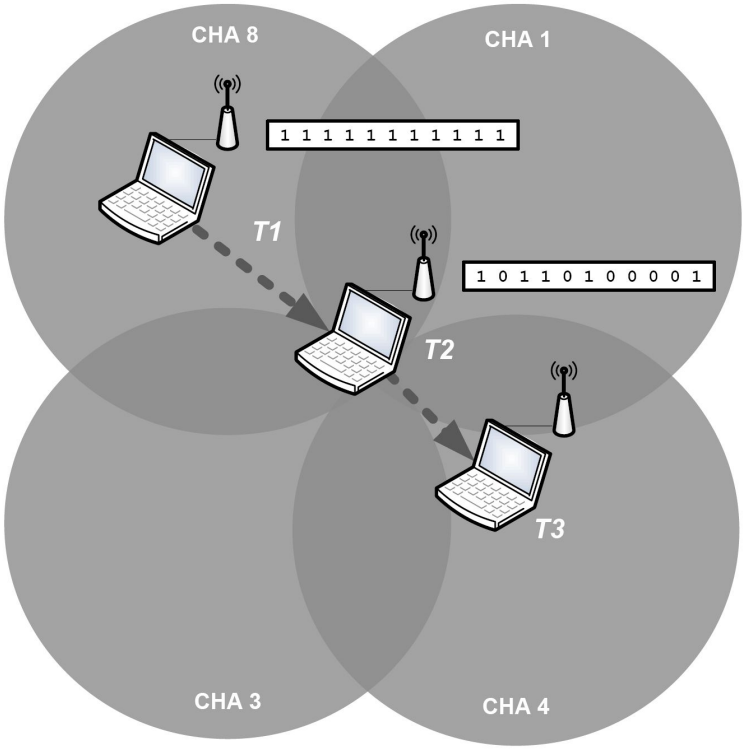


Figure 2.25 Selective Scanning Scenario

No	KEY	1st BEST AP	2nd BEST AP
1	MAC1 (Ch 1)	MAC2 (Ch 2)	MAC3 (Ch 3)
2	MAC2 (Ch 2)	MAC1 (Ch 1)	MAC4 (Ch 1)
...	...	...	...

Figure 2.26 Cache structure

key. Then, when the mobile station needs to handoff, it first checks for entries in the cache; if there is no entry for that key (cache miss) the station immediately starts a selective scanning. After it is finished, the best two access points (best two detected signal strength) are pushed into the cache using current access point MAC address as key. Otherwise, if cache entries are found for that key, the mobile station should instantly request to associate the first best new access point. If association fails, then the second best entry is requested. Finally if this second association attempt fails, selective scanning technique is applied.

Thus, the proposed cache structure is built and conserved by mobile stations. Its entries are incrementally inserted, handover after handover, so the mobile station has a static view of the deployed network while moving.

Experiments suggested in [14] show excellent results both referring to handover latency reduction and packet loss rate. As an example, the simple application of Selective Scanning reduces in an average of 43% the total handover latency. Applying the Caching mechanism, the handover latency is only related with the authentication and association delay, so the reduction averagely arrives to a 97%. Moreover packet loss is reduced a 41% and 74% applying Selective Scaning and the Caching combined solution respectively.

Regardless of these interesting results and considering the non-invasive tactic used by both techniques, since they do not require modifications in the access point side, it has to be counted that the access point cache has to be carefully maintained. Erroneous values in the cache should unsuccessfully end the handover process. Otherwise, as both the cache and the binary mask are incrementally built, initial handovers will apply the standard technique, falling in highest latencies. Moreover, both structures contain static information about access point so possible modifications in the network layout or sudden out-of-service access points are not considered.

Moreover, as access points' information is incrementally obtained, selective scanning and caching method give best results after the mobile station performs several and continous handovers. If the mobile station disrupts its connection and starts operating in other different network deployment, more than a few handovers should take place before this technique starts providing low scanning latencies.

Selective Scanning based on Neighbor Graphs Continuing with the same line of thought, the solution proposed in [15] reduces the number of channels to scan using a *Neighbor Graph* structure.

A *Neighbor Graph* is an undirected graph structure, containing *vertexes* representing access points in a particular area and *edges* symbolizing each mobility path

between them. Fig. 2.27 illustrates a typical indoor-outdoor access point deployment. Blocks represent mobility obstacles, so a mobile station moving in that area should handover like the pattern shown in the neighbor graph. For instance, someone leaving the room where $AP3$ is located, will always associate with $AP2$ because it has a single mobility path.

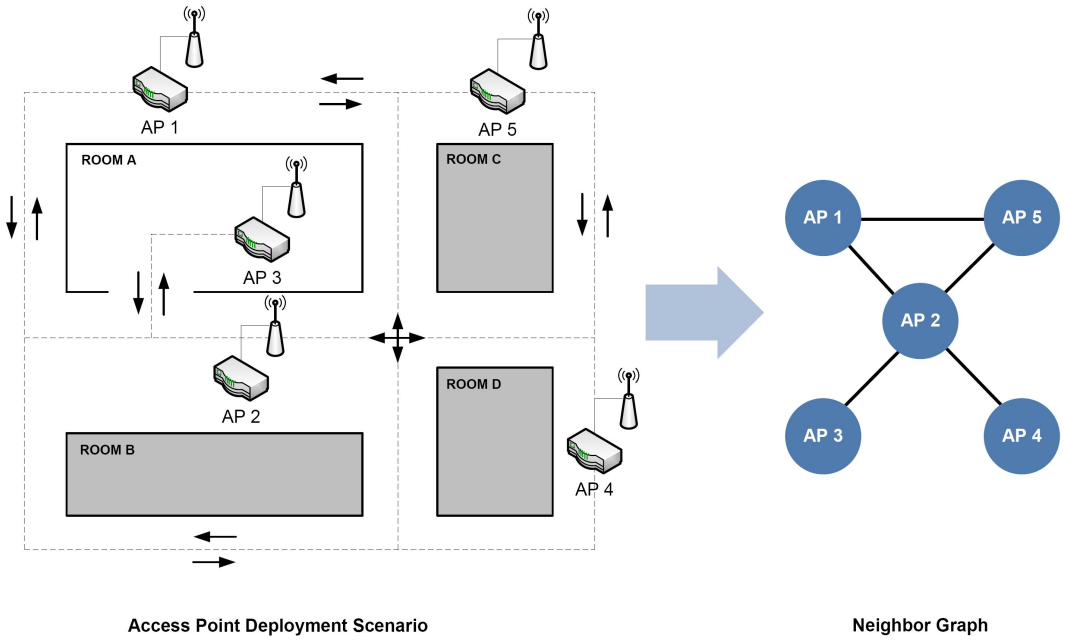


Figure 2.27 A Neighbor Graph example[11]

The neighbor graph structure is maintained by each access point connected to the distribution system. Thus, in order to build the neighbor graph two different approaches exist. First, each access point could use information contained on *Reassociation Requests* received by mobile stations, that include the Basic Service Set Identifier (BSSID) of previous access point (See Section 2.6.2). On the other hand, access points may build the neighbor graph using *Move-Notify* messages provided by the Inter Access Point Protocol (IAPP).

Therefore, when a mobile station needs to start a handover, it unicast *Probe Requests* exclusively to the neighboring access points. Thus, there is no necessity to wait for *MinChannelTime* or *MaxChannelTime* to receive a *Probe Response*. This technique denotes a completely different philosophy than that proposed in the 802.11 standard.

[11]Extracted from [15]

Experimental results show a notable latency reduction, which variates with the number of neighbor access point to unicast Probe Requests. On the other hand, as this solution is based in implementations taking place inside the access points, presently developed 802.11 networks should upgrade its firmwares so as to put into practice this new method. This is such an extremely important constraint that counteracts its employment in the currently widely deployed 802.11 wireless ambiance.

Global Positioning System Assistance The main goal of a handover process is to associate to an access point having a relative good signal. Usually, signal quality is related to the distance between the mobile station and the access point. The Global Positioning System (GPS) enables a GPS receiver to determine its location, speed, direction and time.

Authors of [18] designed and implemented a handover algorithm assisted by GPS information. The main purpose of this approach is to obtain real time information about mobile station position and to maintain a list of access point keyed by latitude and longitude location, operating channel, SSID and IPv6 prefix. The latter prefix is maintained in order to provide also assistance to network layer handover (see section 2.6.3)

Mobile stations are equipped with a GPS receiver that periodically sends a location update message (LU) to a GPS Server containing its coordinates. So mobile station's position is estimated every second with an accuracy of ten meters. In order to identify mobile station's movement, the distance between the previous position and the actual one is calculated using the Haversine formula, that considers a spherical earth, ignoring ellipsoidal effects[12].

When the last distance calculation provides a value greater than one meter, movement is notified to the GPS server sending a LU message. The GPS server then checks the distance between the mobile station and its current access point. Thus this result is compared with a fixed distance threshold so as to decide to initiate the handover. When the calculated value is lower than the threshold the closest access point is chosen from the list. Authors set the threshold as the 50% of the maximum access point range.

A Probe Request is sent to the selected access point, waiting for a probe response so as to then initiate the authentication process. *MinChannelTime* and *MaxChannelTime* are no longer used in this approach. The discovery latency is reduced to the minimum.

[12]Using Haversine formula good results can be obtained in relative short distances, as typical 802.11 network coverage areas

The integration of GPS technologies for the handoff assistance contributes to achieve a low cost handover. However, as GPS signal is only detected in open environments, wireless LAN indoor applications are out of the scope of this method. Moreover, an accuracy of ten meters appears as such an important error source, while common coverage ranges for 802.11 networks arrive to a hundred meters. This condition should affect the best access point determination concluding in bad handover decision.

Reducing the time spent on each channel

One of the paradigms within the handover optimization has been focusing on reducing the time of control time variables of the handover process. Thus, several works based on simulations and real testbeds have proposed different values for *MinChannelTime* and *MaxChannelTime*, introducing some special considerations and defining new techniques.
The main contribution in this field has been introduced by Velayos and Karlsson in [19]. The authors divide the handover process in three different phases: detection, discovery and execution. It can be appreciated that they differ from those phases described in section 2.6.2, since they refer to a prior context which is not taken in consideration by the standard. This previous phase is related to the handover triggering, in other word, when to decide to initiate a discovery phase to obtain candidate access points.

Thus, there is a relationship between lacks in radio connectivity and the number of failed frame transmissions. Failures might be produced by several causes: collisions, radio signal fading or simply because the mobile station is getting closer to its current access point coverage boundary. Mobile stations first assume collision and retransmit the frames. If transmission fails again, then radio fading is assumed and the link is probed by sending probe requests. Only after several unanswered requests, the station declares the out of range status and starts the search phase. Then authors focus on the number of collisions using a cumulative distribution function in order to decide the scanning initialization. Fig. 2.28 presents an analysis of the number of consecutive collisions per transmitted frame under different network loading conditions (five, ten and twenty stations operating on the same channel) without station mobility or handover risks. It can be appreciated that three consecutive collisions is such an unusual situation, so after the third successive collision there should be no necessity to probe the channel and the scanning phase could be initiated.

Thenceforward the authors present some considerations to optimize the discovery phase. They focus on fixing the best values for *MinChannelTime* and *MaxChannelTime* presenting theoretical considerations and simulation results. In the case of

[13]Extracted from [19]

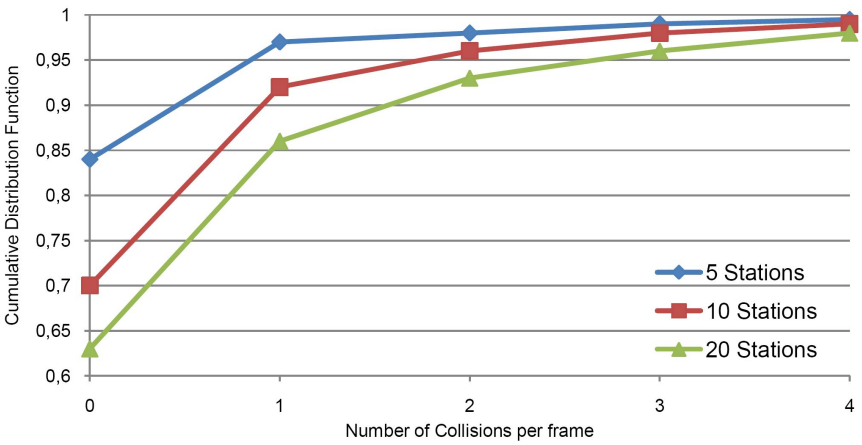


Figure 2.28 Consecutive collisions per transmitted frame in different contexts[13]

MinChannelTime, authors establish the concrete value for the maximum time an access point should need to answer, considering that both the access point and channel were idle.

$$\begin{aligned} MinChannelTime &= DIFS + (aCWmin \cdot aSlotTime) \\ &= 50\mu s + (31slot \cdot 20\frac{\mu s}{slot}) \\ &= 670\mu s \\ &\cong 1TU \end{aligned} \tag{2.3}$$

If propagation time and probe response generation time are neglected, the 802.11 DCF establishes that the maximum response time has the form of equation 2.3 presented in [19]. Inside this equation, $DIFS$ refers to the DCF Inter-Frame Spacing, $aCWmin$ refers to the minimum size of the backoff contention window and $aSlotTime$ refers to the duration of the time slot in micro seconds. So the value of *MinChannelTime* should consider the worst case in order to receive a probe response from any access point, that is, with maximum value for $aCWmin$. Thus, as *MinChannelTime* is expressed in TU[14] authors propose to establish $1TU$ as a concrete value.
Hence, when analyzing the *MaxChannelTime* value, some simulations were run and results show that the transmission time of a probe response depends on the offered load and the number of stations on each channel (see Fig. 2.29). Moreover they conclude that *MaxChannelTime* is not bounded as long as the number of stations can

[14]When referring to time related variables, TU (Time Unit) indicates a period of time equal to 1024 μs (microseconds)

increase. Author recommend then to set a value for *MaxChannelTime* that would avoid responses from overloaded access points. They fixed a value of $10TU$ based on a maximum value of ten stations associated with the same access point, each one waiting for $1TU$.

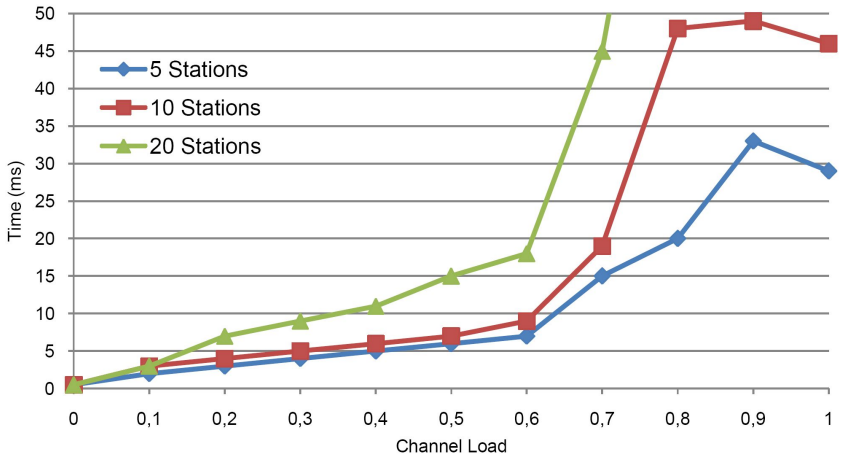


Figure 2.29 Time to wait for a Probe Response considering different loads[15]

As a conclusion, this work has introduced concrete values for scanning control variables based on theoretical considerations and simulations. It has to be appreciated that the 802.11 standard does not provide any approximation for those values. However, the fact of providing a fixed and static definition for those variables does not guarantee a successful discovery process. Authors introduced several constraints and consideration regarding the number of stations operating on each channel (network deployment) and data traffic conditions. Thus, these fixed values could effectively work for some scenarios, but in other cases unnecessary delays should be introduced or oppositely, the scanning process could not find any candidate access point.

Distributing the handover decision

The 802.11 standard active scanning algorithm implicitly defines that the handover process should be performed after loosing the connection with the current access point. Thus, several studies proposed to divide and distribute the scanning operation so as to avoid undesired effects on sensible real time applications.

Smooth Handover The *smooth handover* method [12] is based on the division of the discovery phase into multiple subphases with the aim of reducing packet delay and jitter by allowing the mobile stations to send a receive data packets during the

[15]Extracted from [19]

period of time between each subphase.

Each subphase is built by making *groups* of channels, so instead of sequentially scanning all allowed channels, the mobile station breaks the scanning process and switches to the normal transmission mode. Then, the following group is scanned, so after all groups are scanned the mobile station should have useful information to handover.

As it can be appreciated, the total handover latency should be the same than in the standard approach, but upper layers will not be affected during the process. The key concept of this approach is to start the discovery phase earlier, as soon as the mobile station realizes that it is entering into an overlapping area and a possible disconnection with its actual access point is close to take place.

Therefore, in order to start discovering access points earlier, the smooth method allows the mobile station to modify the RSSI (Receive Signal Strength Indicator) threshold to a higher value, so as to previously trigger the discovery phase. On the other hand, the 802.11 active scanning algorithm simply launch the discovery process when the RSSI threshold fall to a fixed low value. Thus, the basic behavior of the smooth algorithm dynamically changes the RSSI threshold value between a defined minimum and maximum. If when scanning a channel, a good signal quality access point is found, the threshold is increased in a fixed rate, decreasing the probability of relaunching the discovery soon. Otherwise, if no access point is discovered in the group, the threshold is decreased in a fixed rate, making more probable to start the scanning on the next group soon.Then, the decision to associate depends on the relative signal quality of the candidate access point and the current connection quality. In order to prevent the *ping-pong effect*, in which the mobile station associates with a candidate access point but suddenly disassociates because of a slight change in the mobile station movement, the association decision is taken under the condition expressed in equation 2.4 (where $RSSI_{candAP}$ refers to the received signal strength of the candidate access point and $RSSI_{oldAP}$ refers to the received signal strength of the old access point) and illustrated in fig. 2.30.

The inclusion of D in the condition implies that a mobile station only associate with the candidate access point when its signal strength is greater than D respect the old access point signal strength indicator. In fig. 2.30, a mobile station moves out from its old access point coverage range, so when it arrives to position 1, it starts scanning a group of channels. Then the threshold could be increased or decreased depending on the information received from discovered access points. In this case, a new candidate access point with relative good RSSI is found, so the mobile station will be able to associate to it after passing across position 2.

$$RSSI_{candAP} - RSSI_{oldAP} > D \tag{2.4}$$

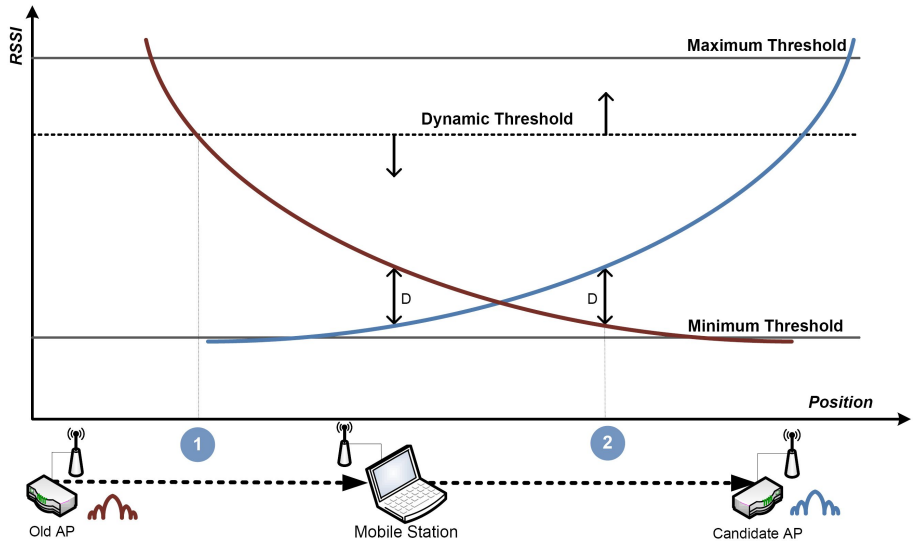


Figure 2.30 A smooth handover scenario

Authors of [12] evaluated the performances of the smooth approach implementing a testbed consisting in four access point and the mobile station. As a result, an important reduction of the packet loss rate is obtained. Common values for packet loss using the standard full scanning algorithm were close to 50 packets, while applying the smooth handover approach, only 6 packets were considered as lost.

Despite of these good results, some relevant constraints have to be considered when deciding to implement the smooth behavior. The first is related with the station's movement, so it can just move at a modest speed. On the other hand, a really strong constraint is introduced, so there must be *enough overlapping area* between neighboring access points, limiting the deployment scenarios where this technique may be applied. If small overlapping areas exist, there will not be enough time to distribute the scanning process while moving. Moreover, the situation becomes worst if the mobile station increases its speed, so it will not be able to associate, remaining permanently disconnected, affecting upper layers.

SyncScan Unlike common handover optimizations, that have focused on active scanning, the SyncScan [16] method bases on the standard passive scanning approach. Authors consider that the active scanning behavior produces a considerable overhead because all allowed channels must be probed for potential access points. Therefore, great delays are produced because of channel switching, so the station must resynchronize and start demodulating packets on the new frequency.

On the other hand, delays related with standard passive scanning are far from

being small. Typically, a passive scanning latency (refereed as *ScanDelay*) has the value given in 2.5, where *NumChannels* refers to the number of channels to be scanned and *BeaconPeriod* refers to the period of time between two beacons sent by the same access point. As described in section 2.5.2, a common value for the *BeaconPeriod* parameter in the Management Information Base is 100 ms, so scanning latencies usually exceed one second.

$$ScanDelay = NumChannels.BeaconPeriod \tag{2.5}$$

The aim of SyncScan is to synchronize clients with the time of beacon broadcast on each channel, so station switches channel exactly when a beacon is about to arrive. Thus, access points on the same channel will send their beacons at time t, and then in the following channels beacons are sent delaying time by d (on time $t + d$), with d being incremented when channel is switched. Thence, all access points on the same channel synchronize beacon sending and a mobile station receives the information successfully because it has switched channel just before the beacon is sent.

As it can be appreciated in fig. 2.31, access points send beacons in a synchronized manner, so the mobile station switches channel for a fixed amount of time, waiting for the beacon. In the same figure, the influence several delay (D) intervals inside the transmission (T) periods is shown.

As mobile stations have real-time information about neighboring access points because they have been receiving beacons regularly, the handover latency is reduced to authentication and association delay. The mobile station can always select the best access point to associate, since it has signal strength values contained inside access point's beacons. Equation 2.6 details the SyncScan latency (refereed as *SyncScanDelay*) for each channel; $SwitchTime$ is the period of time that station needs to change channel (authors estimate 19 ms by experimentation) and $WaitTime$ is the interval of time the station waits for beacons.

$$SyncScanDelay = 2.SwitchTime + WaitTime \tag{2.6}$$

This new approach may eliminate the scanning delay inside the handover latency, but some difficulties should be analyzed:

- **Time Synchronization** - The fact that the station must switch channel *just* when a beacon is *about* to arrive, add a complex time synchronization managing between stations and all deployed access points. Clock accuracy becomes critical in this approach. Minimum deflections may produce bad effects, while the station will never discover access points to handoff. Authors propose the usage of Network Time Protocol (NTP) that maintains time to within $10ms$ (1/100 s) accuracy over the Internet, achieving precisions of $200\mu s$ (1/5000 s) or better in local area networks under ideal conditions.

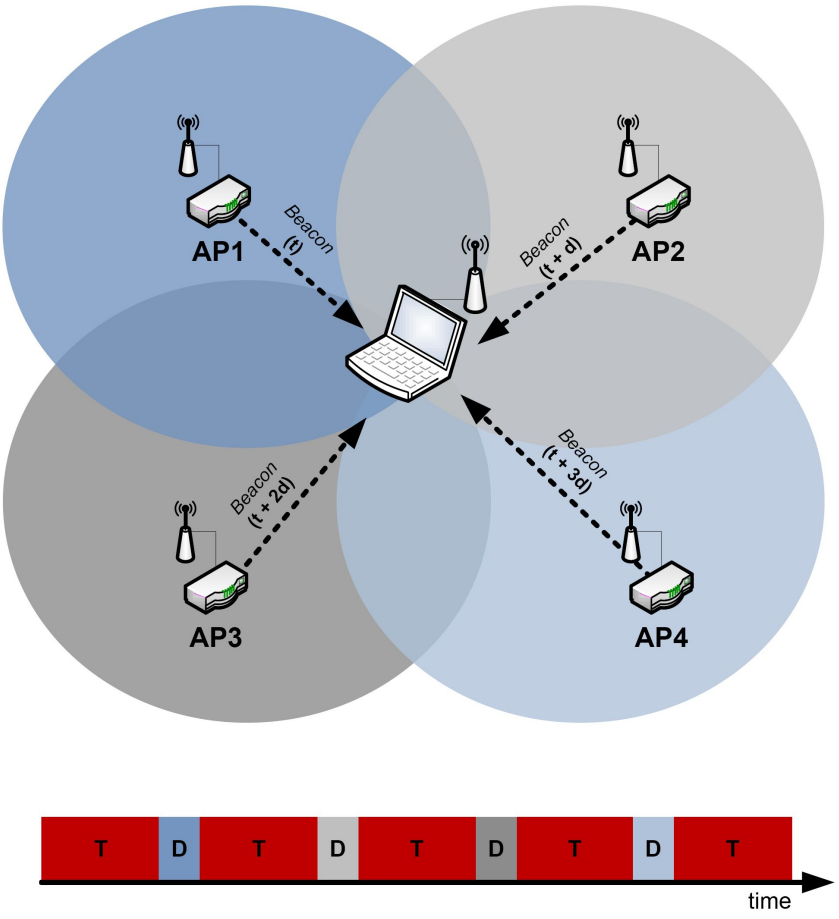


Figure 2.31 SyncScan: A synchronized passive scanning

- **Network Operation** - Related with the above mentioned limitation, synchronizing times between access points under the control of the same network operator is a complex but not unachievable work. The problem appears in a common multiple operators scenario, where the SyncScan implementation appears impracticable.
- **Overheading** - SyncScan procedure is performed regularly, producing several unavailable periods for data packets transmissions, so some of them may be missed while exploring other channels.
- **Collisions** - When multiple access points have to be discovered on the same channel, the station should wait for all beacons to be received. As all access points on the same channel will send beacons synchronously there is a high probability of collision between them, so the station must wait for the medium contention as detailed in section 2.2.1, increasing the unavailability for data packet transmissions.

Periodic Scanning Analogously to the SyncScan tactic, the Periodic Scanning method presented in [17] deals with the distribution of the scanning latency while the station is still connected to its current access point. But in this case, despite of waiting for passive beacons, the station should act in a proactive way.

With this aim, a mobile station periodically performs a short discovery phase, each time on a different channel, during a period of time equal to MinChannelTime, so it can discover candidate access points before handover. During the anticipated scanning, the station builds a list of target access points. The information maintained is basically the MAC address of the access point, the operating channel and the Service Set Identifier (SSID).

The decision to enter in a short discovery phase depends on the Received Signal Strength Indication (RSSI) of its current access point and the number of targeted access point in the list. In the case where the signal strength is good enough (-50dBm to -75dBm), the station chooses a random number between 1 and 2 seconds and build a timer. Upon timeout, the station switches to probe mode, changes channel and starts sending Probe Requests. When the signal strength drops below -75dBm, and the station has not yet found a target access point, it sets the next scan between 200 ms and 300 ms. This shorter period of anticipated scanning is necessary to accelerate the discovery of a new access point while a handover might occur soon. If the station discovers at least one access point during an anticipated scanning, it sets again the default interval for the next scanning, so overhead is minimized.

When it is time to handover, the station consults its list of target access points. If the list is empty and does not contain any access point, a standard active scanning

process is initiated. Otherwise, the station selects the last found access point. It switches to the appropriate channel and tries to authenticate. If the target access point is still in range and accepts to authenticate the station should associate and the handover finishes. If the Authentication Request is not acknowledged, it means either that there was a collision, or that the access point is not in range anymore because of station's mobility. Thus, after three not acknowledges authentication requests the station should try the next access point in the list, or begin a standard handover if the list is empty.

Authors of [17] presented an experiment based on network simulations in six different scenarios. Good results are limited for no more than a few scenarios, depending on the characteristics of the access points deployment. Then this solution cannot be applied as a general one. On the other hand, the station continuously probing and switching between channels contribute to a higher power consumption, a non desirable situation in a battery operating environment, which has energy constraints.

Pre-Authenticating using IAPP

As the 802.11 standard does not specify possible communication processes between access points within the same extended service set (ESS), an optional Inter Access Point Protocol extension (IEEE 802.11F) was defined so as to provide information exchanging between access points from different manufacturers.

IAPP deals with security, so it verifies association uniqueness between a mobile station and an access point. Moreover it manages mobile station's security context exchanging between access points in a secure mode.

After IAPP was ratified and published in 2003, some handover optimizations focused on it in order to reduce authentication and association latency. In [20], authors propose a fast handover solution using IAPP. Thus this solution bases on reducing the scanning latency making use of a Neighbor Graph structure (as defined in Section 2.7.3) but merging it with other techniques based on IAPP in order to reduce reauthentication and reassociation delays.

The mobile station should periodically check the received signal of its current access point. If it is less than a defined threshold, the station should enter in a *Selective Scanning* and *Pre-Registration* state. Thus, in order to prevent packet loss, a station should require its current access point to buffer its incoming packets by entering in a *Power Saving Mode*. All neighbor channels contained in the neighbor graph will be probed for a time equal to *MinChannelTime*. After scanning all neighbors, the station continues receiving data packets. Without further delays the station will send a *IAPP Pre-Registration-indication* packet to the selected access point. Then the candidate access point should verify current access point BSSID exchanging frames

between a Remote Authentication Dial-In User Service (RADIUS) server. If identity was successfully verified, the candidate access point exchanges a *security block* with the current one. Then, the former sends an encrypted *IAPP Pre-Registration-request* to the latter, so as to require context information of the mobile node. This information is transmitted together with an *IAPP Pre-Registration-response*. After sending an *IAPP Pre-Registration-confirm*, the candidate access point will be able to directly accept an association request from the mobile station.

Thus, when the mobile station moves to a new location where the received signal becomes lower than the threshold, the handover process will be triggered. An *IAPP Handoff-notify* will inform the current access point about the identifier of the new access point. The association is broken and reassociation must be performed. Data packets sent to the mobile station are still delivered to the old access point, so it must buffer and then redirect them to the new access point using *IAPP Data-forwarding* frames. After *Reassociation Request and Response* are successfully exchanged, buffered data is transmitted to the mobile station and data communication process starts.

Instantly, the mobile station obtains information about neighbors for this new access point, in order to be ready for the next handover.

All in all, the introduction of IAPP features helps to prevent packets loss. The combination with a Neighbor Graph Selective Scanning additionally improves the handover latency. Despite of that, as any new protocol it has to be implemented. In this case, it contemplates modification both in the station and access point sides. Thus, it is a not an easily implementable solution.

Discussion

Several handover optimization tactics have been presented in this section. As a strength, all proposed solutions truly contribute to the handover latency reduction. However a division has to be made between them. Some of them focus on optimizing the handover mechanism in a *low invasive* approach, so only minimum modifications are required in the mobile station side. Contrariwise, others far-reaching handover tactics require implementations both inside the mobile station and the infrastructure network drivers. The latter should be referred as *high invasive* methods.

In all current *low invasive* approaches, like those described in 2.7.3, several constraints are introduced in the field of network deployment. As 802.11 networks have been widely deployed, it is particularly common to find scenarios were multiple access points allocated in different channels and managed by different operators coexist. Thus, *low invasive* solutions could not provide an open and widely applicable answer to the handover optimization due to these limitations. In the case of Selective Scan-

ning approaches, it is being assumed that some channels have no activity, so they should not be scanned for access points. In addition, a Smooth handover approach, considers enough overlapping areas between two neighboring cells, for instance, an indoor wireless LAN environment. Then it also forces the mobile station to just move in a modest speed, limiting the mobility feature. Thus stations moving at higher speeds will not be able to adapt the threshold variable, since the signal strength will fall down very fast.

On the other hand, the fact of reducing handover time control variables, *MinChannelTime* and *MaxChannelTime*, to the low static values could fall in handover process failures, since the station may not be able to discover an access point in a particular channel because of traffic and collision matters.

High invasive techniques are difficult to implement in actual deployed networks. However, they could be gradually applied into new wireless components.

Chapter 3

Adaptive Discovery Mechanism

We have presented an analysis of different proposed scanning optimizations. Our objective is to analyze and evaluate 802.11 scanning algorithm so as to propose an easy-to-deploy scanning algorithm that reduces the number of cases where no AP is found due to bad deployment conditions while maintaining a reasonable scanning latency, reducing the disconnection time. An Adaptive Scanning Algorithm is presented so as to dynamically decide how long to wait for AP responses on each channel depending on the deployed scenario.

3.1 A Real-World Case Study

In order to introduce concepts related with Adaptive Systems, a real-life situation should be presented as an analogy.

In an urban environment, where vehicles mobility is a real difficult-to-manage matter, an Adaptive Urban Traffic Control System (AUTCS) appears as a promising solution.

The main goal is to optimize the traffic flow, minimizing delays and queues. Under this approach a centralized system simply adapts traffic control variables, in this case, red light times for main streets and avenues in the city. Several video or traditional loop detectors are strategically positioned in the city in order to capture real scenario descriptors about traffic conditions, as the number of vehicles flowing in a specific area or the average speed of a particular way. A centralized adaptive system takes this useful information and evaluates the scenario state so then an Adaptive Algorithm immediately generates an optimized output containing the new red light time for a particular road, based on preexisting rules and policies.

For instance, high values of traffic flow generally produce shorter red light times. In this case, the scenario descriptors are the real traffic conditions of the street to

optimize and its neighboring roads. As a result, an Adaptive Traffic Control System drastically reduces delays and stops in the vehicle flow. As a limitation, one of the main problems about an Adaptive Traffic Control scheme is related to the data collection of decision making variables, because physical street installations are difficult to maintain.

Under this considerations, an analogy could be presented between Traffic Control and Discovery Processes. In the wireless ambiance, a *scenario* is defined by all the components of the infrastructure network, like access points, mobile stations and the distribution system. Then, control variables for the discovery process are *MinChannelTime* and *MaxChannelTime*, like red and green light times in the traffic control approach. Scenario descriptors are a set of parameters that precisely defines link conditions. As it was analyzed in the first chapter, the *Received Signal Strength Indicator* (RSSI), the number of operating access point and the channel layout are useful parameters so as to describe the discovery scenario, as traffic flow measurements describe the traffic control scenario. On the other hand, inside the adaptive system, some policies are to be defined so as to implement the adaptive algorithm that will produce optimized values for both *MinChannelTime* and *MaxChannelTime* control variables.

3.2 Introduction to Adaptive Systems

First, it is useful to define the concept of the word ***Adaptive***. Adaptive means *having the ability or tendency to adapt to different situations*[1]. The word *adaptive* could be used in several contexts, as medicine, optics, economics, management, traffic and transportation as we have shown in the example. In the field of informatics, computer sciences and networking, *adaptive* usually nominates a particular class of *system*. Thus, an ***Adaptive System*** should be defined as *a physical system that is capable of self-adapting in response to changing environments*[2].

Adaptive Systems appear as an alternative for traditional systems, that are unable to adjust to novel environmental scenarios. They have the ability to act with a different behavior depending on each environment state. Their core is based on a particular algorithm, also referred as *Adaptive Algorithm* containing the policies and primitives for the decision making.

Fig. 3.1 illustrates the different components inside an Adaptive System. Basically, a scenario is characterized by a set of descriptors that act as an *input* for the adaptive system. The *scenario descriptors* take values due to measurements inside the scenario,

[1]From the Collins English Dictionary
[2]From the International Society for Complexity, Information, and Design Encyclopedia

which is managed by some *control variables*. These descriptors allow the adaptive system to determine the *state* of the scenario in a particular moment. Thus, an adaptive algorithm uses some information and policies contained in a a *database* so as to produce an *output* that gives new optimized values for the control variables. This process becomes a cycle and the scenario state should be frequently evaluated by the adaptive system through the scenario descriptors, and a new output is generated for this new state.

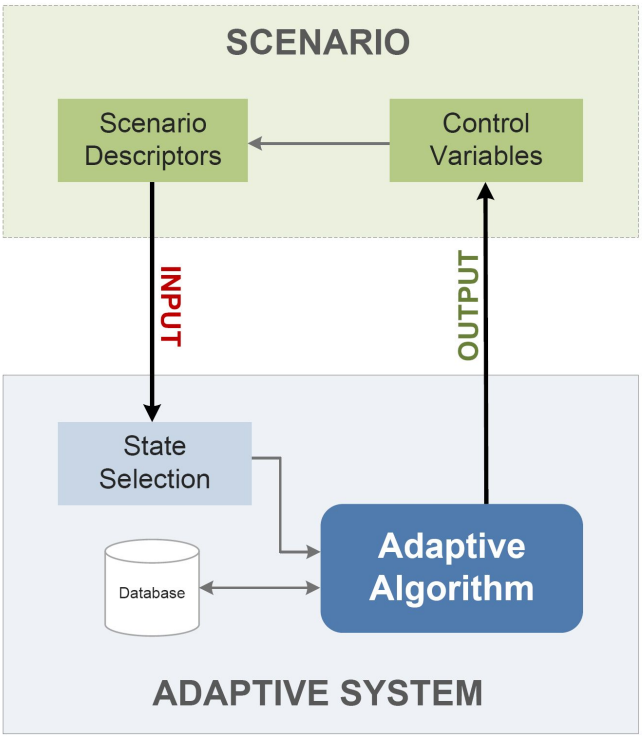


Figure 3.1 An Adaptive System Architecture

3.2.1 Scenario Descriptors

In order to define a general and low invasive discovery mechanism, simple and standard available information should be used as descriptors for the scenario condition. In this section we propose some variables that will act as scenario descriptors in the adaptive behavior.

Receive Signal Strength Indicator During the scanning process, the mobile station should receive Probe Responses from candidate access points. Once received, mobile stations have information about the received signal level from each access

point. These notified levels correspond to radio frequency propagation measurements, as seen in section 2.3.1. Therefore, the 802.11 standard defined the Receive Signal Strength Indicator (RSSI) as an optional parameter to implement in wireless network interface cards that has a value of 0 through $RSSI_{Max}$. This parameter is a measure by the PHY sublayer of the energy observed at the antenna so as to determine the quality of the communication link. As $RSSI_{Max}$ in the Probe Response frame has a value of 256, the RSSI parameter corresponds to a one byte integer value. Despite of that, manufacturers only implement a reduced scale, typically starting from 0 up to 31, 60 or 100.

As shown in [21], RSSI is not a traditional unit for energy measurement. In 802.11 and others RF based technologies, milliwatts (mW) and decibel milliwatts (dBm)[3] are used for energy magnitudes. Specifically in 802.11 network interface cards, which have a rated output of 100mW, power is transmitted at roughly 20dBm (100mW) and received all the way to 96dBm (2,51 e-10 mW).
Thus, a conversion technique is needed in order to obtain a precise RSSI value. As the relation is far from being linear, a practical approach must be considered depending on the RSSI scale selected by the manufacturer. As an example, CISCO uses a 0 to 100 RSSI scale based on a lower level of -113dBm and an upper one close to -10dBm.

In the proposed mechanism, as the analysis is based on simulation processes, a particular conversion technique will be presented so as to emulate the RSSI parameter.

Number of Discovered Access Points Inside a discovery process, several frequency-separated channels are scanned for access points. As it was stated in section 2.4.3, network interfaces operating in the same frequency band produce interferences, causing a negative impact in the achievable throughput. An adaptive behavior should weigh up a scenario were a lower number of access point are working in the same channel.

Fig. 3.2 illustrates a particular case where a mobile station arbitrarily moves to an area covered by three different access points. Blue colored access points, *AP 1* and *AP 2* operate in the same channel. Oppositely, *AP 3* operates in a different channel. Thus, in an equivalent RSSI condition, *AP 3* should be selected to associate so as to prevent greater interferences. It must be considered that only if these access points operate in non overlapping channels, interference would be completely avoided, otherwise, it is only reduced.

This descriptor is very accessible to obtain during the scanning process, because it is simply implemented as a counter of received probe responses from different access points.

[3] dBm is a 10-based logarithm measurement of signal strength

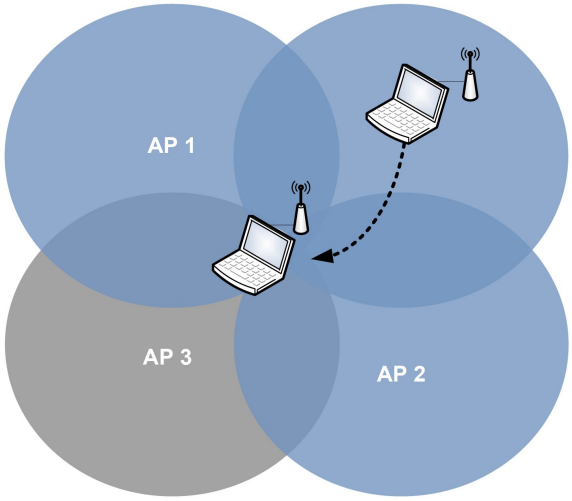


Figure 3.2 Multiple overlapping access points

3.2.2 Control Variables

Taking information from Scenario Descriptors, the Adaptive Algorithm to be implemented should optimize a set of Control Variables that defines the scanning behavior. As defined in the standard active scanning procedure, both *MinChannelTime* and *MaxChannelTime* play the main role when trying to reduce the scanning latency.

The new adaptive behavior needs to control and dynamically change the values of both *MinChannelTime* and *MaxChannelTime* so as to adapt them to the scenario's requirements. For that reason, a careful analysis of both values and possible limits will be defined in this section.

MinChannelTime

MinChannelTime, was defined as the minimum time to spend on each channel while scanning, thus it is the maximum time for an access point to respond to a Probe Request. Moreover it was stated that because of the complexity of the wireless medium one frame should take different amounts of time in order to reach its destination, depending on the actual network condition. In a scenario characterized by a high level of interference, low RSSI, high levels of congestion, or any combination of them, a frame should take more time to arrive than in an optimum scenario due to collisions and further necessary retransmissions.

Under this consideration, if a mobile station waits just for a reduced or optimum static period of time (as stated in [19]) it may not receive any Probe Response in those problematic scenarios.

For that reason, the inclusion of a dynamical comportment for *MinChannelTime* value will allow the mobile station to wait different periods of time so as to maximize the number of discovered access points independently of the network condition.

This variation on the value should be achieved within certain limits, so as to prevent undesirable high or low values. These limits are presented in the following section.

MinChannelTime: Lower Limit A lower limit for *MinChannelTime* is equivalent to a value to be applied only under optimistic network conditions. For instance, these conditions should be applied when a mobile station is probing a channel where only one access point is present. In this case, there is a very low probability of collision, while a low congestion level is being considered. Thus, equation 3.1 presents the different delays to be incurred, refereed as $MinLowerLimit$. Inside this equation, $DIFS$ refers to the DCF Inter-Frame Spacing, $backoff_{min}$ refers to the minimum attainable backoff delay, and $T_{TXProbeResponse}$ the delay for the Probe Response transmission. As detailed in 2.2.1, CSMA/CA access control approach establishes that a node trying to send the first frame should wait for *DIFS* expiration. Then the mobile station should wait for the backoff to expire. In this case, it may be considered that the value for $aCWmin$ (the backoff window size) should be equal to the result given by replacing n as 1 in equation 2.1. This is the minimum value for the congestion window. In addition, the delay for the transmission of the *Probe Response* frame from the access point to the mobile station must be considered, so it can be assured that finally, the mobile station could obtain the required information for that candidate access point. In this last case, the obtained value for the parameter was calculated by simulation.

$$\begin{aligned} MinLowerLimit &= DIFS + backoff_{min} + T_{TXProbeResponse} \\ &= DIFS + (aCWmin \cdot aSlotTime) + T_{TXProbeResponse} \\ &= 50\mu s + (31slot \cdot 20\mu s/slot) + 104.27\mu s \\ &= 774.27\mu s \\ &\cong 800\mu s = 0.8TU \end{aligned} \tag{3.1}$$

As a conclusion, a mobile station should wait as least for *MinLowerLimit* before switching to the next channel. So as to validate these theoretical results, a simulation process has been performed using *SimulX Network Wireless Simulator* (introduced on section 4.1.3). It considers an scenario using only one station. A hundred simulations have been carried out for each evaluated scenario. As illustrated in fig. 3.3, results

are organized into an histogram, considering the possible delay to receive a probe response and relative frequencies.

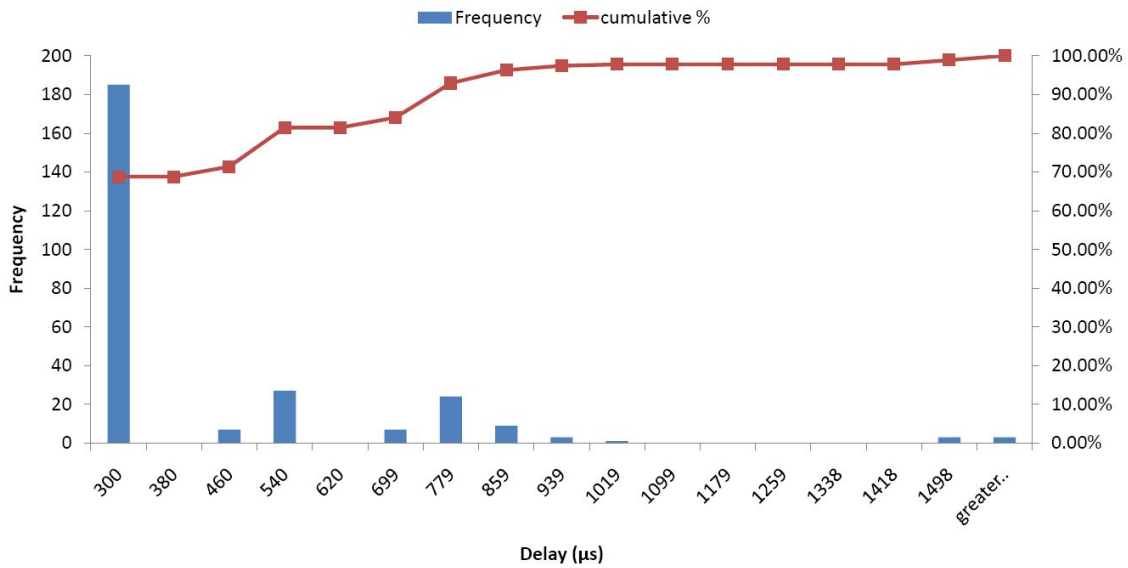


Figure 3.3 MinLowerLimit: Simulation Histogram

In most than 93% of the cases, the delay is less than the proposed *MinLowerLimit*. Thus, the calculated threshold is enough to avoid unsuccessful handovers.

MinChannelTime: Upper Limit Thinking about a worse scenario, frame collisions due to several mobile stations and access points operating in the same channel may occur. Furthermore, when more than one access point offer their services within the same frequency band, then they will concurrently send a *Probe Response* management frame to the requesting station, which it will conclude in a collision, and retransmissions must be performed.

It has to be considered that if the mobile station only receives one *Probe Response* from any of all operating access point, it will be enough to trigger *MaxChannelTime* timer, so further responses will be received during this period.

From this point of view, after collisioning, the second attempt will be initiated by the access points, so the transmission will be randomly deferred considering a longer value for the backoff contention window size. As a conservative perspective, the station should receive at least one Probe Response after the first retransmission. Equation 3.2 shows the obtained upper limit. Then, the second attempt will be initiated by the access points, so the transmission will be randomly deferred considering

a longer value for the backoff contention window. As a conservative perspective, the station should receive at least one Probe Response after the first retransmission. Equation 3.2 shows the obtained upper limit.

$$\begin{aligned} MinUpperLimit &= DIFS + backoff_{min} + 2 \cdot T_{TXProbeResponse} \\ &= DIFS + (aCWrtr_1 \cdot aSlotTime) + 2 \cdot T_{TXProbeResponse} \\ &= 50\mu s + (63slot \cdot 20\mu s/slot) + 208.54\mu s \\ &= 1518.54\mu s \end{aligned} \tag{3.2}$$

In the worst case, a second retransmission could be necessary, so the contention window size must be incremented to the next possible value, as detailed in equation 3.3.

$$\begin{aligned} MinUpperLimit &= DIFS + backoff_{min} + 3 \cdot T_{TXProbeResponse} \\ &= DIFS + (aCWrtr_2 \cdot aSlotTime) + 3 \cdot T_{TXProbeResponse} \\ &= 50\mu s + (127slot \cdot 20\mu s/slot) + 312.81\mu s \\ &= 2902.81\mu s \end{aligned} \tag{3.3}$$

It is necessary to define the upper threshold considering an enough value, so as to avoid extra delays, but also to prevent the possibility of no receiving response. As the previous case, some simulations were performed in order to appreciate the behavior of this amount of time inside a complex scenario. The histogram presented in fig. 3.4 was obtained by simulation as in the last case, but considering eight mobile stations. It shows a linear cumulative comportment, with still high occurrences for the greater values.
Those greater values are positioned around $1800\mu s$, as seen in fig. 3.4. Thus $MinUpperLimit$ should take this value as shown in 3.4.

$$\begin{aligned} MinUpperLimit &= 1843.20\mu s \\ &= 1.8TU \end{aligned} \tag{3.4}$$

MaxChannelTime

The second discovery control variable to be detailed is *MaxChannelTime*. In this case the mobile node, after receiving a first Probe Response in a certain channel before *MinChannelTime* expiration, should set a timer equal to *MaxChannelTime* and receive all received Probe Responses during this new interval.

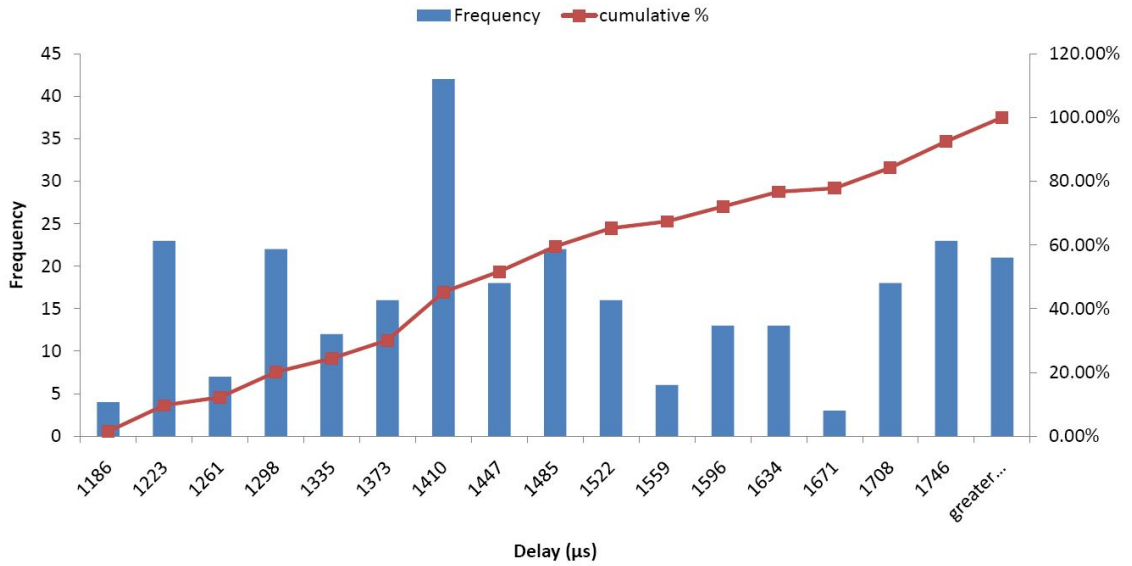


Figure 3.4 MinUpperLimit: Simulation Histogram

The introduction of *MaxChannelTime* allows receiving information from different access points operating in the same channel. As the case of *MinChannelTime*, the standard does not provide any certain value, thus protocol implementations have defined different values.
Waiting for a longer period allows each access point to compete for the medium and send a *Probe Response*. It has to be considered that there are some difficulties to determine the threshold values for this timer, mainly because of the number of access points to be detected on the same channel and traffic conditions. The first constraint is basically related with the deployment policy.

MaxChannelTime: Lower Limit In order to define the *MaxLowerLimit* it should be considered that at least, an interval equal to *MinLowerLimit* is necessary. Analogously to that case, it could be considered that network condition is optimistic. Thus, *MaxChannelTime* can be adapted to the minimum as necessary, according to 3.5.

$$\begin{aligned} MaxLowerLimit &= 774.27\mu s \\ &\cong 0.8TU \end{aligned} \tag{3.5}$$

MaxChannelTime: Upper Limit On the other hand, *MaxUpperLimit* should allow receiving all possible responses, and not only the first one. A lower timer could exclude a possible good candidate access point that was not able to successfully send

its response because of medium conditions. However, higher values produce longer latencies, contradicting the fast handover aim. Therefore, it should not be a good decision to wait for a long time in order to receive several responses for different access points, because that is an indication of an overcrowded channel, and interference levels will be definitely significant.

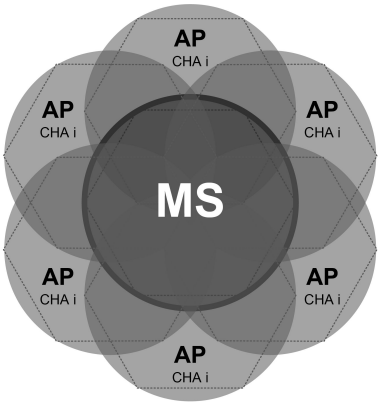


Figure 3.5 Six access points operating on the same channel

Considering an hexagonal access point deployment like that introduced in 2.4, it can be stated that six different access points to be detected on the same channel (as illustrated in fig. 3.5) could work as an appropriate limit. Initially, it can be set a maximum limit of six times *MinUpperLimit*, as shown in 3.6. Simulations were run using the *SimulX* platform, and considering the proposed hexagonal pattern in order to validate this statement.

As illustrated in histogram 3.6, in more than the 97% of the cases the station discovered all access point in less than $10TU$. So, a concrete value for *MaxUpperLimit* is presented in 3.6

$$\begin{aligned} MaxLowerLimit &= 10240\mu s \\ &\cong 10TU \end{aligned} \tag{3.6}$$

3.2.3 Channel Switching Policy

During an *adaptive full scanning* procedure, the channel sequence could produce some effect in the discovery latency as well as *MinChannelTime* and *MaxChannelTime* values. The adaptive process will be performed step-by-step while switching channels, so

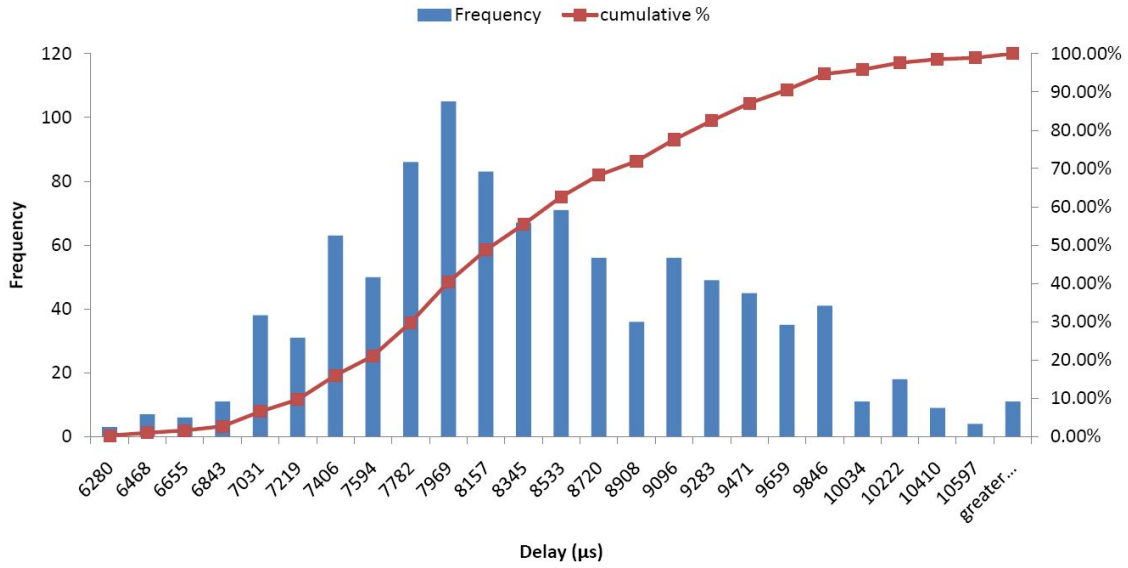


Figure 3.6 MaxUpperLimit: Simulation Histogram

if a method to control the order is imposed, successful handovers should be achieved faster.

In the context of 802.11 networks, three non-overlapping channels are offered. Considering North American specifications, channels *1*, *6* and *11* do not overlap. As stated in [22], a proper deployment typically uses only these channels.

Thus, it could be assumed that prioritizing those channels, the probability of discovering candidate access point faster may be increased. However, experiments proposed in [24], show that in more than 75% of the cases, Wireless Network Interface Cards (WNICS) send *Probe Request* to channels 1, 7 and 9 with higher significance. So there is not any correlation between the non-overlapping channels and its priority in the scanning process in the common active discovery approaches implemented by manufacturers.

For our purpose, it has been suggested to randomize the sequence in two different phases. The first subsequence is randomly created between the non-overlapping channels. Subsequently, the rest of the channels are taken into consideration. Therefore, if channels 1, 6 and 11 are ranked first and access points with relative good signal level answered mobile station's request, it should not be necessary to wait for longer times in the following channels. Fig. 3.7 illustrates this random mechanism for the case of a FCC 11 channels configuration.

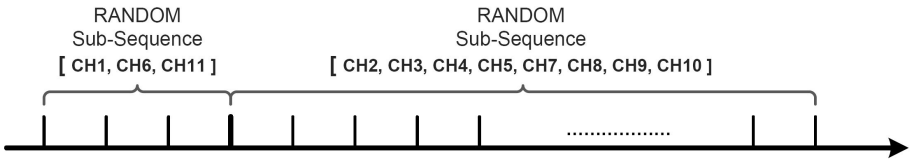


Figure 3.7 Random channel sub-sequences

3.2.4 Adaptive Discovery Algorithm

Concepts about the scenario definition have just been introduced. Thus, an intensive analysis of the adaptation process should be presented. First, following the illustration 3.1, before making any adaptation on the *control variables*, the adaptive system should define the correspondent *state* for the scenario after obtaining *scenario descriptors* values.

Therefore, the main purpose is to obtain the best candidate access point from all discovered. While scanning a channel, one or more access points can be discovered with different values of $RSSI$. The desired context is that where a single access point with relative good signal strength is found rather than multiples access points sharing the same channel. For that purpose, and so as to obtain the *decision making* parameter (R) acting as an input for the *adaptive algorithm*, a single function is defined containing both *control variables*. This function is indicated in equation 3.7, where $RSSI$ is the Receive Signal Strength Indicator of the responder access point and $APsChannel_i$ is the number of discovered access points on channel i.

$$R = \frac{RSSI}{APsChannel_i} \tag{3.7}$$

This simple relation allows balancing the decision of selecting the best access point in a fairness approach. Under this consideration, two different access points having the same signal strength and operating in separated channels will be ranked in a different way. Overcrowded channels will not be well weighted as those with lower number of access points.
After calculating R value for each discovered access point, the adaptive algorithm maintains a *global maximum* of R, that it will be labeled as R_G. This global value corresponds to the best ranked access point and acts as a reference to the best candidate access point to associate after the full scanning procedure is concluded. Furthermore, for each channel, a *local maximum* value for R between all the discovered access points is calculated, obtaining R_L. Using the latter, the adaptive process will decide the reduction of *MinChannelTime* and *MaxChannelTime* for the next channel to be scanned.

The scanning process initiates using *MinUpperLimit* and *MaxUpperLimit* val-

ues for *MinChannelTime* and *MaxChannelTime* respectively. Thus, the main structure of the adaptive algorithm uses a comparison mechanism based on a fixed rate. Thus, if R_L is compared between one of the defined ranges, $MinChannelTime$ and $MaxChannelTime$ will be reduced to a fraction of their present values, defined by F_R. Otherwise, if no access point is found in the channel, $MinChannelTime$ and $MaxChannelTime$ are increased to a new average value between the actual and the previous calculated, due to the application of F_I factor. This factor is calculated so as to obtain new $MinChannelTime$ and $MaxChannelTime$ values increased a 50% of the different between actual values and those previously calculated. For instance if the actual value is $800\mu s$ and the previous was $1000\mu s$, the new value is $900\mu s$.

The main behavior of the adaptive process is illustrated in fig. 3.8. Control variables are adapted until the last channel is scanned. Then, the mobile station should initiate the reauthentication process with the candidate access point referenced by R_G. Moreover, typical reduction factors (F_R) are presented in figure 3.9.

Different deployments generate different scanning conditions. In some cases, as a heterogeneous channel disposition, a mobile station should wait the same time for all channels using the standard algorithm. The Adaptive Mechanism allows to distinguish between finest and poorest scanning conditions, so it can manage the risk of discovering less access points in the following channels by changing the values of $MinChannelTime$ and $MaxChannelTime$. In order to deepen this concept, some practical examples will be introduced in the following section.

Adaptive Discovery Algorithm: A practical approach The proposed mechanism [25] [26] was designed with the aim of providing a fast handover solution for a wide-spectrum of network deployments. This section will introduce some examples in order to demonstrate the algorithm's operation. Deeper analysis and scenarios will be presented in section 4.2, so as to evaluate results and make further conclusions.

Let's consider the scenario presented in figure 3.10, in which a mobile station moves from its current access point coverage area to a new location, where three different access points operates as follows:

- **AP1:** Operating in channel **3**. RSSI sensed by the mobile station is **80%**
- **AP2:** Operating in channel **3**. RSSI sensed by the mobile station is **70%**
- **AP3:** Operating in channel **11**. RSSI sensed by the mobile station is **60%**

When the scanning process initiates, AP3 should be discovered first, due to channel sequence policies stated in section 3.2.3. R_L is calculated and then as it was the first discovered access point, R_G will reference to it. Then, $MinChannelTime$ and $MaxChannelTime$ are adapted and the next channel is scanned. In this case

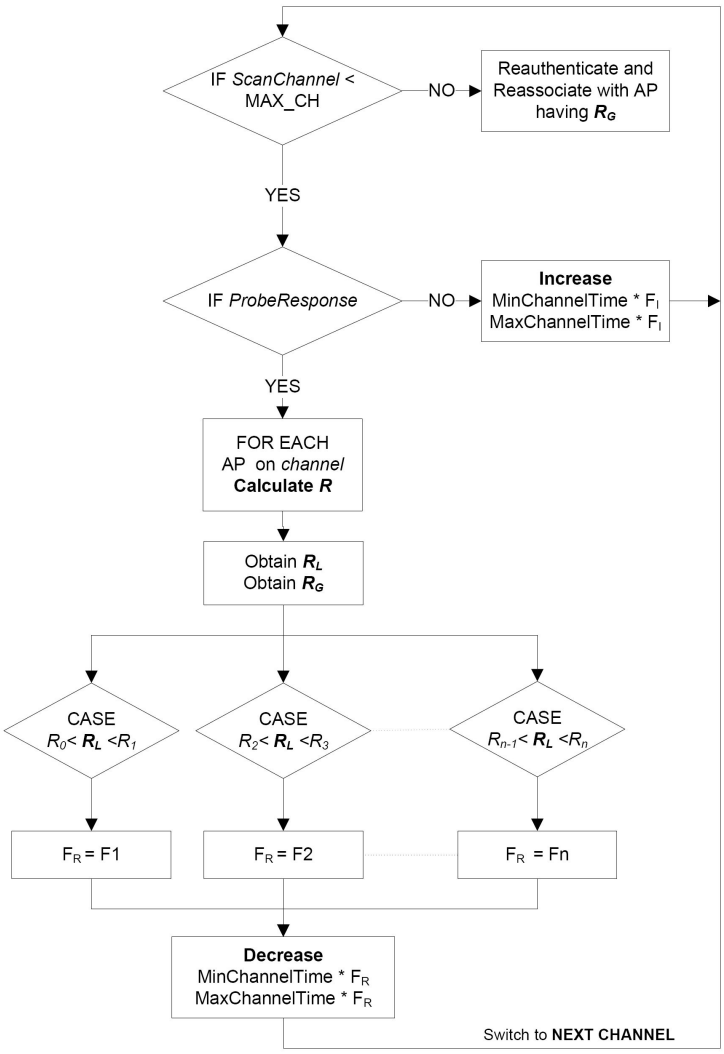


Figure 3.8 Adaptive Discovery Algorithm: A block diagram

R1		R2		R3		R4	
FROM	TO	FROM	TO	FROM	TO	FROM	TO
0%	20%	20%	40%	40%	60%	60%	100%
FR	**0,6**	FR	**0,5**	FR	**0,4**	FR	**0,3**

Figure 3.9 Typical values of F_R for each defined range of R_L

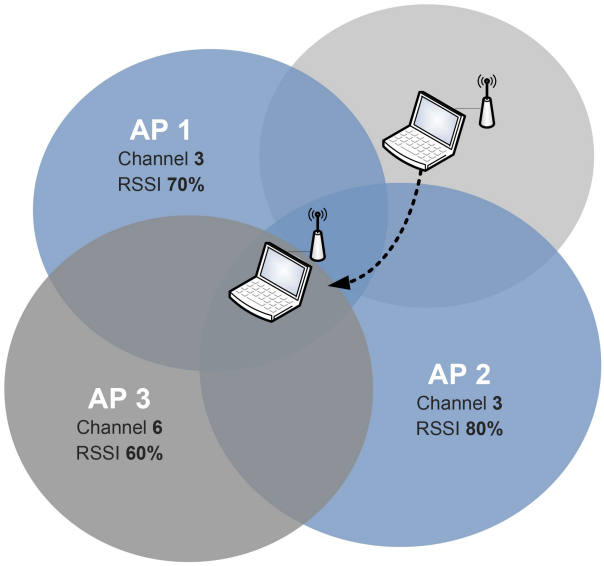


Figure 3.10 An example scenario

no access point will be discovered so $MinChannelTime$ and $MaxChannelTime$ will be increased by F_I. The station should continue looking for access point in other channels until it switches to channel 3 and both AP1 and AP2 should be discovered. Before switching to channel 3, $MinChannelTime$ and $MaxChannelTime$ were being increased by F_I. Then, as two access point are discovered in channel 3, R_L is obtained dividing the maximum RSSI between AP1 and AP2 by two. Then if the last R_G is lower than actual R_L, so the former will become equal to the latter. If channel 3 is not the last channel to scan, the following channel is scanned with lower values for $MinChannelTime$ and $MaxChannelTime$ depending on the range obtained by R_L. Then, $MinChannelTime$ and $MaxChannelTime$ are increased step by step until the last channel is scanned and the reauthentication is initiated.

Figure 3.11 shows the evolution of control variables for this particular case, assuming a particular channel sequence from all possible random sequences.

It has to be appreciated that as the channel sequence is randomly assigned, different latencies should be obtained for the same scenario using the same adaptive algorithm. However, it will always select the best discovered access point. Moreover, when AP3 is discovered, as the calculated R_L is relatively low, control *MinChannelTime* is adapted up to *MinLowerLimit*.

Channel	Discovered AP	R_L	R_G	F_R	F_I	MinChannelTime [TU]	MaxChannelTime [TU]	Latency [TU]
1	0	0%	0%	-	-	1,800	10,000	1,800
6	1	60%	60%	0,4	-	1,800	10,000	10,000
11	0	0%	60%	-	1,625	0,800	4,000	0,800
8	0	0%	60%	-	1,192	1,300	6,500	1,300
7	0	0%	60%	-	1,081	1,550	7,750	1,550
3	2	40%	60%	0,5	-	1,675	8,375	8,375
9	0	0%	60%	-	1,500	0,838	4,188	0,838
10	0	0%	60%	-	1,167	1,256	6,281	1,256
4	0	0%	60%	-	1,071	1,466	7,328	1,466
5	0	0%	60%	-	1,033	1,570	7,852	1,570
2	0	0%	60%	-	1,016	1,623	8,113	1,623
						Total Scanning Latency		30,577

Figure 3.11 Possible results for the example case

3.3 Handover Optimizations

As well as the adaptive solution could achieve a fast handover process, other techniques may be applied in order to complement and optimize the whole solution. Two different solutions are being presented in this section: the elimination of the *ProbeDelay* timer on the discovery phase and a modification in the way the access point compete for the medium while sending a Probe Response.

Both mechanisms could be introduced in the *Adaptive Discovery Mechanism* so as to evaluate its results via simulation processes in the following chapter.

3.3.1 Avoiding the *ProbeDelay* Timer

The 802.11 standard discovery mechanism establishes that before probing a channel, a mobile station should sense it for a period of time equivalent to *ProbeDelay* or until an indication of an incoming frame arrives. The application of this timer avoids an empty channel blocking the entire scanning procedure, so the station will not indefinitely wait for incoming frames.

The 802.11-2007 standard [1] states in section 10.3.2 a detailed description for the control variables taking part of the scanning phase. However, the standard defines it as an integer value acting as the delay (in microseconds) to be used prior to transmitting a Probe frame during active scanning, but the valid range of this parameter does not appear available. On the other hand, it also states that for the case of *MinChannelTime*, its valid range is any integer equal or greater than *ProbeDelay*.

From an objective point of view, *ProbeDelay* timer acts as a passive *MinChannelTime* timer, while it does not act in a proactive way. During this period the mobile station only waits for incoming frames. Moreover, going in the direction of a general application algorithm for heterogeneous channel deployments, forcing the station to wait for this extra time contributes to increase the total handover latency.

Finally, the proposed adaptive solution will act in a proactive way, directly probing the channel while waiting for *MinChannelTime*.

3.3.2 Fast Medium Access for *Probe Responses*

When the MAC Layer and the CSMA/CA strategy were analyzed in section 2.2.1, it was stated that a node trying to transmit a frame may content for the medium using a Distributed Coordination Function (DCF) or request for a higher transmission priority while using the Point Coordination Function (PCF).

In the case of the standard active scanning approach, management frames uses the DCF, so any request or response must wait for DIFS before competing for the medium. It is also clear that if reducing the handover latency is the main goal, some management frames could change their behavior while acting in a contention-free approach.

For instance, modifying the way the access points compete for the medium while sending a *Probe Response* could help to reduce the scanning time. In the particular case of the *Probe Response*, the standard determines that it should be sent after waiting for DIFS, which has a typical value of $50\mu s$. Let's think in a deployment scenario in which more than one access point operate in the same channel. Then, as they will answer to the same *Probe Request* it is highly probable to have a first frame collision after waiting for DIFS, so then the backoff function is called. It can be easily appreciated that the risk of not finding an access point in a certain channel before $MinChannelTime$ expires is quite high. Thus, if the Short Inter-Frame Space (SIFS) is applied, this highly probable collision would be managed sooner, so the backoff function is called earlier and finally the risk of falling in a *MinChannelTime* expiration before the mobile station receives the *Probe Response* is decreased. Moreover, if SIFS is used, the mobile station can keep the control on the channel soon, avoiding that another mobile station starts a transmission on the same channel.

The inclusion of a contention-free service within the scanning procedure becomes a difficulty while it requires firmware modifications inside the wireless interface cards. However, as the presented solution is evaluated using simulations, this enhancement will be introduced in the proposed implementation.

Chapter 4

Simulation and Experimentation

A new optimized discovery algorithm was presented in chapter 3. At this point, the proposed solution should be carefully evaluated so as to validate its results.

An analysis of available network simulators is presented in section 4.1. Then, the configuration and results for the proposed simulation process is presented in 4.2. Finally, testbed experiments are presented in 4.3.

4.1 Wireless Network Simulators

4.1.1 Simulation Requirement

In the networking field, an incredible growing phase has been carrying out since common users are able to access the Internet wherever they are located. This growth has been performed with the modification and the introduction of new protocols so as to fulfill several requirements. Considering the wireless ambiance, the *mobility* requirement has initiated numerous research processes in order to solve limitations related to this field. On the other hand, with the introduction of innovative communication technologies, not only regarding the 802.11 standard but to some cellular network interfaces as well, users commonly may want to make use of a set of heterogeneous technologies in a particular device. This last matter is referred as *multihoming*.

Thus, while trying to introduce new protocols and techniques so as to improve existent features, *mobility* and *multihoming* introduce some limitations when these methods have to be tested. First, thinking about building real deployment scenarios becomes impracticable due to high costs. Moreover it is impossible to consider only one scenario to be tested, because of *mobility*. As users may move within a coverage area, real testing should involve such a great amount of time in order to cover all possibilities. In addition, when testing interactions between different communication technologies, it is highly possible that concrete devices are not still offered in the

market forcing researches to build ad-hoc equipment.

Network Simulation processes help to avoid these limitations while providing a set of tools and techniques in order to analyse and evaluate different behaviors of real-world network deployments operating for a defined period of time. Simulation is based in two main concepts:

- **Simulation Model** - A representation of the real-world deployment and its parameters
- **Simulation Process** - Execution of a particular model that offers results indicating the system behavior

Thus, a network simulation model will contain communication devices, interfaces and links, including particular parameters for their configuration. In the case of a network simulation process, it should be classified as *dynamic*, *aleatory* and *discrete*. It is *dynamic* because it represents a system behavior considering evolution on the time. Then, it is *aleatory* because the communication process itself may involve aleatory variables such as traffic conditions and random timers. Finally, and not least, network simulation processes are *discrete*, because they are represented as a chronological sequence of *events*; each *event* occurs during a period of time and sets a change of state in the system, scheduling a new *event*.

Therefore, there are some particular components that define a *discrete-event network simulator*. Concrete implementations to be introduced in 4.1.2 base their operations on them. This components are itemized below.

- **Entities:** - An entity may be a communicating node, a link, a protocol or any other element that is related with the system state. Usually they are hierarchically organized, for instance, protocols implemented by a particular node in a layered approach. *Entities* generate and handle simulation events.
- **Activities:** - It should be defined as the action the entity applies to run a particular simulation, while creating events. Three types of activities in simulation could be stated: *delays*, *queues* and *logic*. The *delay* is the deferral of an entity for a definite constant or random interval of time. For instance, a node that has to wait for a timer to expire is considered as a delayed entity. Then, entities wait for an unspecified period of time when they are placed in a *queue*. Common examples of this activity are represented by*packet buffers*. *Logic* activities allow the entity to affect the state of the system through the manipulation of state variables.
- **Events:** - An *event* is a notable occurrence at a particular point in time which causes changes in the state of the system. An entity interacts with activity to

create events. For instance, when an entity starts a delay, an event is scheduled in the calendar to occur.

- **Calendar:** - The calendar is a list of events that are scheduled to occur in the future. In every simulation there is only one calendar of future events and it is generally time ordered (the first scheduled is the first launched).
- **Resources:** - They are any element that the entity needs to perform its operations as traffic intersections and links; They usually involve with utilization rates and costs.
- **Global variables:** - A variable that is available to the entire model at all times. They usually help to schedule events and show results whenever is necessary.
- **System state and global variables:** - The key indicator of a system state in simulation is the current time of simulation. This variable is updated every time an entity takes an event from the calendar. The system state is the set of values all system's variables take at a given point of simulations time.
- **Random number generator:** - It generates numbers used in sampling random distributions, and statistical law generation. For instance, randoms timers related to the CSMA/CA *backoff* function.
- **Statistics collectors:** - They are the aim of any simulation process, because their values should act as the input for further optimization processes. These collectors, record information about state of resources, values of global variables, or certain performance statistics based on attributes of the entity.

Thence, a network simulation process requires modeling a particular network deployment, which may varies on its complexity. Only after the model has been set, simulation processes could be launched. With this aim, the networking community has been working in the development of several simulation tools offering diverse entities in order to cover wide-spectrum network scenarios.

4.1.2 Network Simulation Platforms

In the field of networking, powerful simulation tools are offered in the market so as to model, simulate and analyze different kinds of networks. The most popular implementations are being described in this section in order to finally introduce the solution that has been taken into account to evaluate the adaptive discovery mechanism: *SimulX Wireless Network Simulator*.

With this aim, NS platform [27], OMNet++ [28], OPNET [29] and finally SimulX [30] are detailed below.

NS-2 and NS-3

Features NS-2 and further NS-3 [27] belong to one of the most popular open source Network Simulator solution. The NS platform is a discrete-event network simulator oriented toward network research and education, with a special focus on Internet based systems. The NS-3 project is designing a follow-on successor to the popular NS-2 simulator. Far from NS-2, that supported OTcl[1] scripts, in NS-3 simulation scripts are written in C++, with support for extensions that allow them to be written in Python[2].

Nowadays, the third version is still under development, so most part of simulation processes involve the utilization of NS-2 version. NS-2 source code is a combination of both C++ and OTcl. This is supposed to offer a compromise between performance and ease of use. Being a discrete event simulator, the core of simulation of NS-2 is composed of three main C++ classes, namely the class *Event*, the class *Handler* modeling entities that generate and consume events, and finally the class *Scheduler* which is in charge of scheduling and dispatching events. The simulation is configured, controlled and operated through the use of the interfaces provided by the class *Simulator*.

NS-2 topology model consists of the interconnection of network elements created through the stand alone OTcl classes *node* and *link*. The classic node structure is composed of two important NsObjects the *Classifier* and the *Agent* which enable network and transport protocol simulation. The classic node structure does not model low layer protocols but as the need for wireless modeling become strong, NS-2 now proposes a new mobile node structure that represents the OSI model. The class *MobileNode* extends the basic capability of the *node* class by adding functionalities of a wireless and mobile node such as a link layer (LL), and a MAC layer implementing the specifications of the IEEE 802.11 standard. Furthermore, a physical network interface is used by the mobile node to access the wireless channel. To interconnect Mobile Nodes, NS-2 defines a new kind of link which is *WirelessChannel*. It is supplied with three radio propagation models, *shadowing* model, *free-space* model and *Two Ray Ground* model.

NS-2 has a large number of protocol models, mostly centered on TCP/IP. It is well-suited for packet switched networks and wireless ad-hoc networks, and is used mostly for small scale simulations of queuing and routing algorithms, transport protocols, congestion control, and some multicast related works.

[1]Object Oriented Tool Command Language. It is a scripting language created by John Ousterhout based on a very simple and consistent syntax

[2]Python is a general-purpose and very high-level programming language that emphasizes programmer productivity and code readability

Limitations From the short description of its architecture given above, it is quite clear that implementing a new protocol in NS-2 is not a straightforward process, since it involves adding C++ code for the protocols functionality, as well as updating key OTcl configuration files. In addition, the learning curve for NS-2 is steep and debugging is hard due to the dual C++/OTcl nature of the simulator. Moreover, provided documentation does not help from this point of view. Indeed, it is often limited and out of date with the current release of the simulator. As a net result, NS-2 is not so easy to use in the perspective of contributing new models, protocols, and studying different scenarios at different levels of detail. Furthermore, although the project started since 1989, the simulator is not stable. This is due to the incorporation of contributions from different sources in addition to the continuous changes in the trends of network community requirements and the submersion of new technologies.

An important limitation is referred to the inexistence of *multihoming* features. Moreover, it is not possible to set multiple 802.11 interfaces, because of the channel modeling, a single node could not be operating on two different channels. As stated in [30], NS-2 platform is not suggested for the wireless environment.

OMNet++

Features The second open source simulation tool to analyze is OMNet++ [28] (Objective Modular Network Testbed in C++) is a discrete event simulation environment running on Linux and Windows operative systems and based on C++. It targets to academic and non-profit use, providing a GUI support. Moreover it provides parallel execution (MPI) and several component add-on libraries.

OMNeT++ is component-based, totally modular and open architecture tool. A module is a C++ object, having well specified interface and state, and implementing a specific functionality. There are two main types of modules in OMNeT++, *SimpleModules* and *CompoundModules*. Unlike NS, there are no predefined network devices in OMNeT++. In fact, OMNeT++ models a system, as a particular network, by imbricating hierarchical modules. This allows the user to reflect the logical structure of the actual system in the model structure. Modules communicate via messages passing either directly to their destination modules or along a predefined path, through gates and connections. Modules at the lowest level of the module hierarchy are to be provided by the user, and they contain the algorithms in the model.

Nowadays OMNet++ provides some models in the communication ambiance, TCP, UDP, IP, PPP for the TCP/IP suite and Ethernet, 802.11, FDDI, Token Ring and Peer-to-peer for the network layers.

Limitations Nevertheless, attempts to integrate all these separated contributions usually fail because of their mutual incompatibility. Being highly modular and well structured is a big advantage for OMNeT++ when it comes to implementing new protocols. Extensibility can even be confused with usability since a user is required to define its own modules and classes in C++ language. Unfortunately, the problem of incompatibility between modules (most of the time because they are developed separately) remain a major issue. On the other hand, according to the well-planned conception of its kernel of simulation and to the youth and modernity of its architecture, it is reasonable to expect a good scalability of OMNeT++. Besides, for a good CPU resource management, simple modules appear to run in parallel during simulation execution, since they are implemented as co-routines. In addition, a consistent documentation is delivered with OMNeT++ package. However the incompatibility between crucial modules and the difficulty to run simulation make the OMNeT++ package no suitable for next generation IP simulation.

OPNET

Features OPNET (Optimized Network Engineering Tools) [29]is a commercial discrete event network simulation platform designed by *OPNET Technologies, Inc* widely used in the market in order to provide solutions for governmental and military projects, being also used in the research ambiance as well. It provides the user with direct access to the code and an useful GUI. Several models are available in OPNET, as Ethernet and 802.11. Moreover, it provides predefined devices as routers, modems, switches, access points and servers among others.

Programming in OPNET includes defining protocol packet format, defining the state transition machine for processes running the protocol, defining process modules and transceiver modules for each device node, and finally defining the network model by connecting the device nodes together using user-defined link models.

Limitations As stated in [31], in which the performance of wireless multihop systems with different wireless channel conditions is evaluated under OPNET, the standard routing procedure of the standard wireless library acts as a relevant limitation, while only *static routing* is possible. As a result, the mobility study and possible mobility adaptation techniques left out from the current simulation research. Moreover, as the OPNET project is not an open source solution, it is hard to introduce updates containing new protocols and features to the current version that could become available for the entire networking community.

4.1.3 SimulX Wireless Network Simulator

Overview

As it was presented above, current network simulators provide several tools, but in all cases some limitations appear involving wireless network features, as mobility and multihoming. These limitations are related with the wired origins of those simulators, that did not focus on the wireless constraints to be solved in the physical and access control layers.

In 2006, the contribution of a working group from Louis PASTEUR University, and members of Networks and Protocols team in the LSiiT (Laboratoire des Sciences de l'Images, de l'Informatique et de la Télédétection) in Strasbourg, France, presented the first version of SimulX [30], a C++ Wireless Network Simulator, which is especially designed to simulate IEEE 802.11 networks and IPv6 mobility. It proposes a full implementation of the IEEE 802.11 standard as well as other available link layer protocols.

This simulator was conceived considering the following motivations:

- Research and educational purposes
- Ease of development, simplicity of using, adding and modifying a protocol implementation in a simulation framework
- Open-source
- Modular simulation providing extensibility and re-usability
- Logged into easily readable files inputs and outputs

Nowadays, the second version of SimulX is being developed in TELECOM Bretagne, France, containing additional modules regarding to mobility and network security as well as the incorporation of WiMax physical and access control layers.

In the following sections, the internal architecture of the simulator will be presented.

SimulX Architecture

A briefed Class Diagram of the actual SimulX version is available on its current documentation and it is shown in Figure 4.1. As it can be appreciated, three top level classes help to define the entire architecture of the framework, the *Event* class, the *Handler* class and the *Scheduler* class. Following sections introduce the interaction between all architecture components.

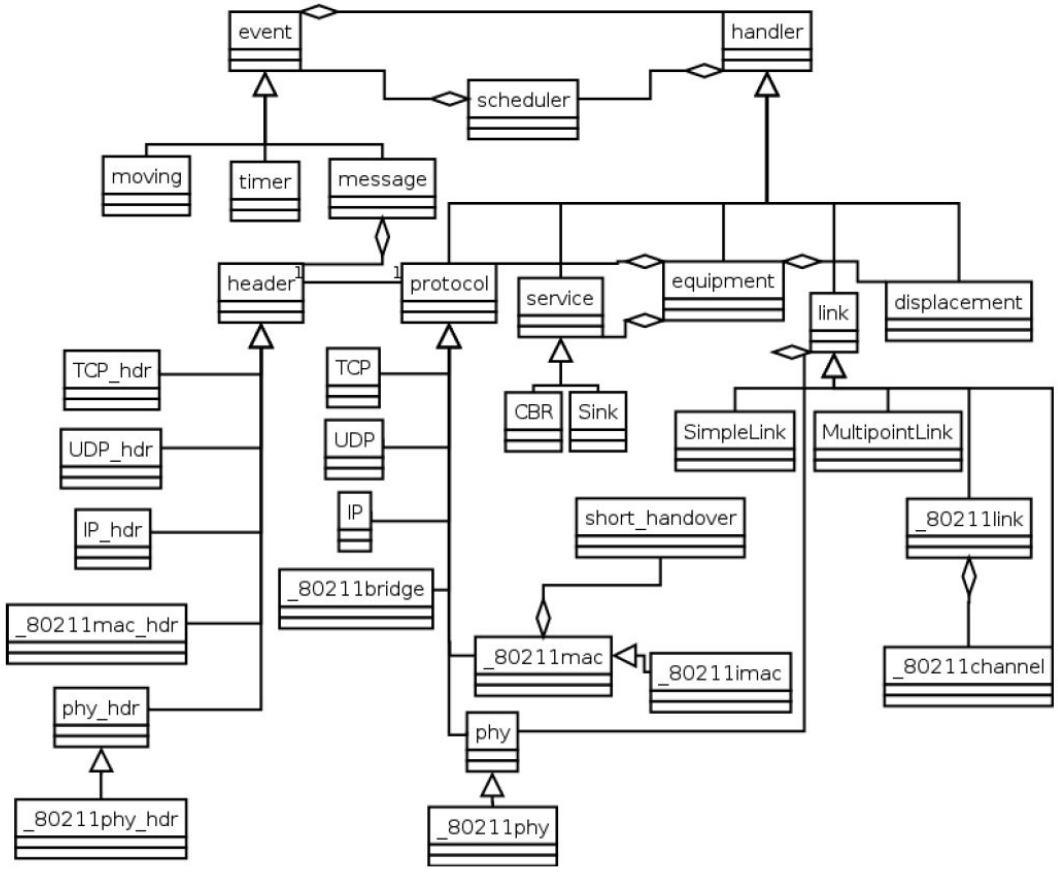


Figure 4.1 SimulX Architecture

The *Event* Class Its instances represent a *condition* that occurs at a given simulated time which causes changes in the state of the system. It acts as the parent class for some relevant classes on the framework, as the particular case of the class *Message*, which is the basic communication entity between layers. Events may be: *scheduled*, *reported*, *canceled* or *sent* directly to another handler. All manipulations on events are made via methods of handlers and the scheduler. Events are generated by entities interacting with their activities, and exchanged between them via the scheduler. The most important attributes of an Event object are:

- **uid** - Unique identifier in the whole simulation instance that allows tracking the event instance and eventually canceling or reporting it, even after it has been scheduled.
- **time** - It represents the date of execution of the event.
- **type** - Referring to a class that inherits from Event. This attribute is used when cast is needed, it means when a specific processing is to be made according the type of the event.
- **h** - It is a reference to the entity in charge to *handle* the event when its date is reached by the simulation time. Handling an event may lead to the generation of other events. An instance of event may be handled by many handlers at different times. Every handler that receives it, performs some activities, and then changes some attributes. The present handler sets *h* field of the event to the next handler address, and sends it. Commonly, this is the action performed by a *Message* event. Actually, the same message is usually handled successively by entities in the simulation, as different protocol layers.

Furthermore, three main classes specialize the *Event* class. The class *Message* is the most frequent unit of exchange between the simulation entities. *Message* is the representative class of data units exchanged between nodes. Derived from the *Message* class the different *Headers* can be added so as to build the complete communication packet. Then, the class *timer* helps to postpone activities for a definite or randomly generated period of time. Finally, *moving* events are generated by moving nodes, and consumed in centralized manner by the wireless channel.

The *Handler* Class All events created in a simulation instance is strongly associated with an object of the class *Handler*, which is responsible for performing the specific task related with the dispatched event, present in the *handle()* function. It is the parent class of all entities that are likely to consume an event, in fact, all classes inheriting from *Handler* are supposed to treat events. For instance, network layers are represented by a set of objects of the class *Protocol*, that inherits from *Handler* and perform the logical algorithm when receiving and sending packets during the simulation. Therefore, all available links in the framework are modeled by another

Handler inheriting class, *Link*, which acts also as a generalization of the specific physical specifications.

The *Scheduler* Class The *Scheduler* class is a Singleton-based class that acts as the the main entity in the simulator, scheduling and dispatching events in a centralized way. The scheduler is the entity that holds the most important global variable in a simulation: *the simulated time* represented by *simul_time* field. It is also the only object that can advance time and provide to other objects about the current simulation time using its *get_simul_time* function. The calendar is represented by a list of events (sched_list) that are scheduled to occur in the future. This events are dispatched considering the attribute *time* of the Event class.

Whenever an object generates an event, it calls the schedulers method *schedule*. This function inserts it in the *sched_list* based on its time field. Finally, this function returns the *uid* of the inserted event to the entity that asked to schedule it. The scheduler *runs* by selecting the earliest event, advancing the simulation time to event time, calling the function *dispatch* before removing the dispatched event from the sched list, and returning to execute the next event. The *dispatch* function insures the consumption of the event by its destination object, its *Handler*. As there is only one event in execution at any given time, simultaneous events are executed on a first scheduled-first dispatched manner.

SimulX Topology

In addition to previous defined concepts, which delineate the main architecture, *SimulX* also provides a set of classes that help to build a complete *Network Topology* inside a particular simulation scenario. *Equipments* will contain a stack of available *Protocols* (TCP, UDP, IP, MAC or PHY, among others) and they will be interconnected using a wired or wireless *Link*. Finally, *Services* are configured inside the equipment so as to establish data packet communication between them.

The *Equipment* Class This *Handler* inherited class acts as a simulation device containing *protocols* and *services* in two independent stack structures. It can represent a simple node, an access point, a router or any other networking device. Because of its flexible and modular internal organization, any kind of equipment can be set up. As the equipment belongs to a specific simulation scenario, it defines a set of *coordinates* so as to precisely locate it and keep track of its movements. It is mandatory to implement at least one physical layer inside the equipment, in order to be able to connect to a link. However, upper layer protocols can be defined as user requires. The member function *send* allows the equipment to start a unicast or broadcast com-

munication.

A common wireless ad hoc scenario is presented in figure 4.2. In this case, *Equipment A* acts as a sender wireless station, so both 802.11-Physical and 802.11-MAC layers protocols are implemented. Moreover, UDP takes care about transport and a CBR (Constant Bit Rate) service is defined above. Finally the equipment is connected to a wireless link. *Equipment B* has the same internal structure, but it implements a different service (SINK) that it will consume the packets sent by *Equipment A*.

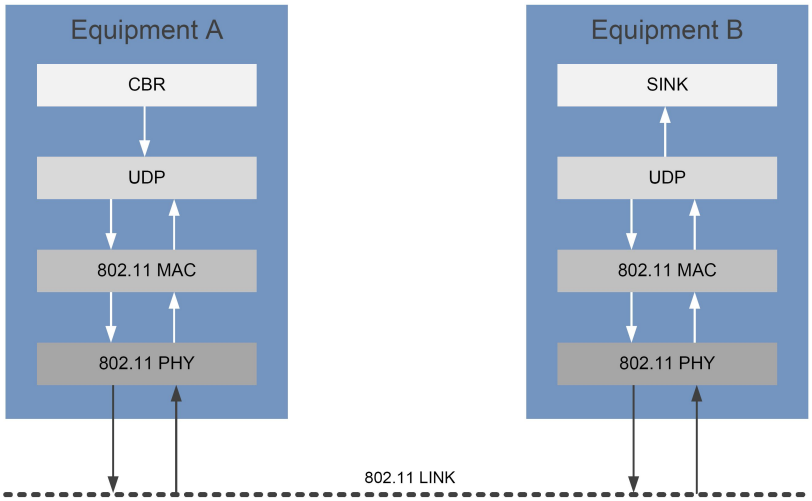


Figure 4.2 Equipments in SimulX

The *Protocol* Class So as to provide modularity to the simulator topology, *SimulX* implements the *Protocol* class acting as the parent class of all present and future protocols to be implemented, so then they can be *attached* to an equipment. The parent *Protocol* class manages the reception of SDU[3] or PDU[4] from applications and upper or lower layers, encapsulation and decapsulation of messages, forwarding to upper or lower protocols, applications or links.

Furthermore, a protocol has both up and down target vectors that refers to all upper and lower layers, which can be one or more protocols or services. In the case of physical layers, the down target vector should contain a reference to a link.

As *Protocol* is a *Handler* inherited class, it has the ability to manage *events*, in general *messages* and *timers*. It makes use of two functions in order to manage them:

[3]Service Data Unit: a set of data that is sent by a user of the services of a given layer

[4]Protocol Data Unit: a unit of data that is specified in a protocol of a given layer and that consists of protocol-control information of the given layer and possibly user data of that layer

recv_from_down and *recv_from_up*. In the case a message is received by an upper layer, it makes some operations and generally sets a *header* so as to send the message to a lower layer using the *find_down_target* function that sets a new protocol or service that will handle the message. On the other hand, while receiving a message from down, the *header* is removed and an upper target is searched by using the *find_up_target* function that looks into the next header type.

Specific functionalities of each protocol are implemented by any of the subclasses inheriting the *Protocol* class, representing specific protocols functionalities. As an example, *SimulX* currently implements TCP, UDP, IP and several MAC and PHY protocols among others.

The *Link* Class One of the main purposes of SimulX is to provide a simulation platform for heterogeneous technologies. So, a parent *Link* class is defined providing common features. It contains a vector that refers to all attached physical interfaces of equipments and an integer value to describe the link delay.

As the case of Equipment and Protocol, a *Link* is a handler, so messages are passed from the physical layer to the link, using the scheduler. The link simply looks for the destination interface within its attached interfaces. Thus, the link schedules the message to be received by the destination node after propagation delay.

Inheriting from the *Link* class, *Simplelink* allows to connect two equipments. When it receives a message from the sender, it transmits the packet to the receiver, after delaying it using its propagation delay. The class *Multipointlink* models an Ethernet link. When an attached node transmits a message on this link, the link delivers a copy of the message to each attached equipment. Finally, the *802.11Link* class models a wireless link, that will receive packets from 802.11 Physical layer.

The *Service* Class *SimulX* implements the parent *Service* class acting completely independent from the *Protocol* class. Usually, a *Service* model an application acting as sender or receiver of data packets. Typically, a *CBR* (Constant Bit Rate) service generates a flow of data packets by using the *gen* function. Then a *SINK* service, is implemented inside an equipment in order to *consume* those generated packets.

A service is always implemented at the top of the equipment stack, so it does not owns upper protocols or services. It is identified by a *srv_id*, acting as the application port number.

SimulX Scenarios

Current version of *SimulX* requires the scenarios to be defined inside the *main* C++ function. As an example, a common data packet communication simulation is described below.

First, *output* log text files are configured. *SimulX* allows managing simulation output into different files, using different filters. For instance, a general output may contain the complete output and then, other complementary files might include only specific protocol output as MAC or UDP, among others.

Then, the *Scheduler* object is created giving the dimensions of the scenario as a parameter. After that, first the *links* and then the *equipments* are instantiated. As parameters, the link should include its own characteristic delay. Equipments' locations inside the scenario are also configured. Thence, all layers for defined equipments are declared, including transport, network, link-layer and physical protocols among others. Afterward, both CBR and SINK services are declared, so as to describe the characteristics of the data packet communication. Protocols and services are linked using the *addDownTarget* and *addUpTarget* functions. As an example, a MAC layer protocol object may use *addDownTarget* function to connect to a physical layer and its *addUpTarget* function to target a network or transport layer.

All declared protocols and services are attached to each equipment using the *addProtocol* or *addService* functions. Finally, the first simulation events are declared using the equipment's *send* function and the simulation is started using the function *run* specified by the *Scheduler*.

4.1.4 Managing Mobility on SimulX

Overview

Classes defined in section 4.1.3 helps to define a network scenario containing equipments implementing protocols and services and being attached to one or more links, in order to perform a communication process. In order to provide a tool for evaluating mobility simulations in a wireless environment, the *Displacement* class has been defined inheriting from the *Handler* class.

Basically a *Displacement* instance contains origin and destination coordinates and a list of places that defines a unique path for a particular *equipment*. The movement is initiated when *simul_time* reaches the value indicated in the *tStart* attribute of the *Displacement* object. Thus, the equipment starts moving at a constant speed, that is also set as an attribute. Thus, after the *Displacement* is configured, it is added to an equipment, that will schedule an *Event* of type *START_MOVE* at time described

in the *tStart* field, using the public function *move*.

After the simulation is started, by calling the scheduler's*run* function, Events start being dispatched. When the *START_MOVE* event is launched, the mobile equipment start moving. it has to be observed that as the movement is a quality of an equipment, not only mobile stations but router mobility can be simulated as well, providing a powerful tool to evaluate the behavior of mobile networks.

Handover Implementation

In order to analyze the handover implementation under *SimulX*, an infrastructure wireless scenario must be considered from all possible designed scenarios. Thus, an equipment can simultaneously move and perform data packets communication with other equipments until it leaves the current access point coverage area. The out-of-range notification is implemented by the *MovementDetection* class. A *MovementDetection* object is configured and attached to the mobile equipment as the same case as the *Displacement* object. Each time the equipment moves, an Event is scheduled in order to verify if it is still inside the coverage area. The *handle* function of the *MovementDetection* object checks the distance between the mobile equipment and the current access point and decides to launch the handover or to continue associated with the same access point.

Due to *SimulX* modularity and flexibility, all *Handover* management operations, including scanning, reauthentication and reassociation, have been implemented in two autonomous classes. The class *HandoverClient* its implemented by the mobile stations while the class *HandoverServer* is implemented in the access point side.

The HandoverClient Class The *HandoverClient* class is a specialization of the parent *Protocol* class and it is implemented inside equipments acting as mobile station. It is directly targeted to the 802.11-MAC layer as shown in figure 4.3; so it can directly interact with it by sending Probe, Reauthentication and Reassociation Requests. Like all other *Protocol* inherited classes, it implements a *handle* function so as to manage events containing received responses messages and handover related timers expiration (*MinChannelTime* and *MaxChannelTime*). Because the *HandoverClient* only has a down target, *the rcv_from_down* function is called when response messages are received, in order to successfully process them and perform the handover.

Internally, it manages a map structure containing a reference to the candidate access points that have answered to previously sent Probe Requests.

Section 4.1.5 will deepen into the *HandoverClient* implementation so as to clearly define the operation of the Adaptive Discovery Mechanism.

The HandoverServer Class On the other hand, *HandoverServer* objects are implemented in the access point side. They are also attached to the 802.11-MAC layer so as to receive request messages and process them to give a response.

In fig. 4.3 the *HandoverServer* object is located inside the equipment. Moreover, due to access point behavior of this equipment, it also implement the 802.11-BRIDGE Protocol, in order to provide connectivity with the wired world.

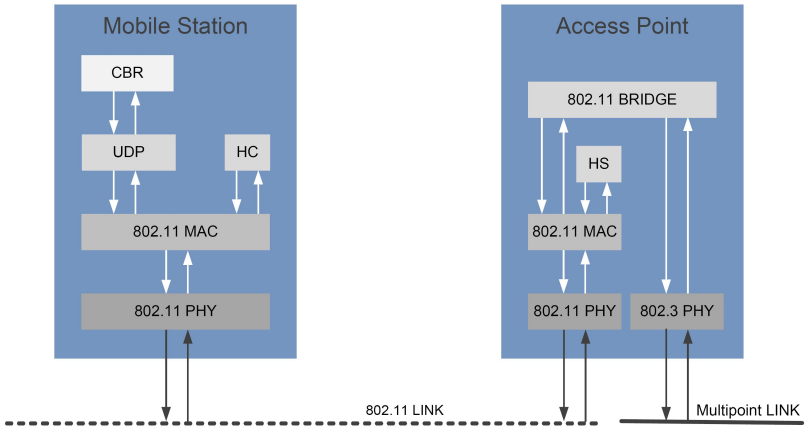


Figure 4.3 Equipment configuration to provide Handover features

4.1.5 Adaptive Discovery Algorithm Upgrade

In order to upgrade the standard handover previously developed under SimulX, two new classes have been designed. *AdpHandoverClient* and *AdpHandoverServer* are based on the standard features but several modification have been introduced in the functions concerning the discovery process.

First, some few variables were added to the class specification in order to provide a reference to both *Scenario Descriptors* and *Control Variables* as detailed below:

- **Timers** - Both *MinChannelTime* and *MaxChannelTime* were implemented as a constant values inside the class specification. With the introduction of the adaptive behavior, they have been defined as integers class attributes. Moreover, because the adaptive approach bases on previous values for the timer, auxiliar attributes were defined so as to maintain a reference to those values and then perform the new calculation for the next timers.

- R_G - The global maximum value for the RSSI scenario descriptor is maintained as a class attribute.
- **Discovered AP** - This scenario descriptor becomes a new class parameter
- **Scanned Channels** - The adaptive approach performs a full scanning process, so a reference to the number of channels scanned should be maintained

Thence, as the adaptive mechanism bases on a *random* channel sequence police, a new procedure in order to decide the next-to-scan channel have been implemented, so the previous linear channel sequence (from 1 to MAX_CHA) was replaced. This function manages two different list of values, one containing the non overlapping channels and the other the rest of them. Values are randomly taken from the lists following the policies stated in section 3.2.3.

Adaptive Algorithm Core

Finally the adaptive mechanism core has been embedded inside the *changeChannel* function. This decision was taken because the adaptation process takes place when the station switch to the next channel. Thus, before the channel is physically switched, if a Probe Response has been received, *MinChannelTime* and *MaxChannelTime* are adapted to lower values.

As it was presented in section 3.2.4, the R variable acts as the decision maker for the adaptation process. Inside the simulator, the theoretical formula presented in equation 3.7 is implemented with available parameters in the simulator environment, as stated in equation 4.1. Inside this equation, $distance_{AP-MS}$ refers to the distance between the access point and the mobile station, $APsChannel_i$ refers to the number of access point in channel i and $Range_{AP}$ and $Range_{MS}$ refers to the range of both the access point and the mobile station respectively.

$$R = \frac{1 - \frac{distance_{AP-MS}}{min(Range_{AP}, Range_{MS})}}{APsChannel_i} \tag{4.1}$$

Thus, as the simulator does not implement the RSSI concept or any indicator referring to signal strength, concretes values for R are obtained using the real distance between the mobile station and the access point, their coverage ranges and the number of discovered access points on each channel. The domains of $distance_{AP-MS}$ and $min(Range_{AP}, Range_{MS})$ functions are indicated in expressions 4.2 and 4.3. So the quotient between them will be within the domain stated in expression 4.4. The subtraction between 1 and the quotient between those variables helps to relativize the numerator of expression 4.1 to the same domain stated in expression 4.4, so for

identical values of $distance_{AP-MS}$ and $APsChannel_i$, greater values of R are obtained when the $min(Range_{AP}, Range_{MS})$ value is superior.

$$Dom\,\{distance_{AP-MS}\} : \{x \in \Re / 0 \leq x \leq \infty\} \tag{4.2}$$

$$Dom\,\{min(Range_{AP}, Range_{MS})\} : \{x \in \Re / distance_{AP-MS} \leq x \leq \infty\} \tag{4.3}$$

$$Dom\left\{\frac{distance_{AP-MS}}{min(Range_{AP}, Range_{MS})}\right\} : \{x \in \Re / 0 \leq x \leq 1\} \tag{4.4}$$

As it can be appreciated, the minimum between the ranges is used as a divisor in order to prevent the case where one of the wireless network interfaces has a reduced coverage area, so it is very close to the necessary range to establish a communication. Hence, if the mobile station continues moving it will be more unprovable that another handover should be performed soon.

As an example, table 4.1 shows possible values for R considering different distances and ranges (expressed in meters) as well as number of access points.

Table 4.1 Example values for R

Case	$Distance_{AP-MS}$	$Range_{AP}$	$Range_{MS}$	$APsChannel_i$	$\boldsymbol{R}$
1	80	150	180	2	**0.233**
2	100	180	200	1	**0.444**
3	50	120	100	4	**0.125**
4	30	100	110	1	**0.700**

As stated in section 3.2.4, after R is calculated, R_G and R_L are obtained. Then, R_L is compared with a fixed range scale, using *if-else* structures, so finally new values are obtained for both *MinChannelTime* and *MaxChannelTime*.

Above presented calculations are performed when one or more access points answer to the mobile station request. Then, if no response is received and there are more channels to scan, the values are incremented as stated in section 3.2.4.

4.2 The Simulation Process

4.2.1 Overview

As it has been asserted several times in this document, in order to suggest the adaptive discovery mechanism as a general applicable solution, simulation process must totally cover the most common and heterogeneous network deployment scenarios.

Once all required scenarios are defined and simulation process are run, obtained results should be critically evaluated, making comparison against the standard active scanning approach, using fixed parameters.

Proposed scenarios will be evaluated using *SimulX* output files. With this aim, two independent files applying different filtering rules are defined in the simulation main function, as introduced in section 4.1.3. The first will present achieved results for each simulation in a single text line. As four hundred simulations will be run for each approach, this output file will contain eight hundred lines displaying the following information:

- **Timestamp** indicating the specific simulation time when the handover process finishes.
- **Duration** in microseconds of the handover process.
- **Interface Identifier** of the selected and further associated access point.
- **Number of Discovered Access Points** after performing the scanning process.
- **RSSI** of the selected access point.

Then, using a second output file, *MinChannelTime* and *MaxChannelTime* values are recorded for each scanned channel, so as to evaluate the profile of the adaptation process.

It has to be considered that in all cases, four hundred simulations are launched for each approach under the same conditions. A single format will be used to present results in order to simplify comparisons between scenarios. These tables presenting the simulation results for each scenario contains both results or the standard approach (Active Scanning AS) and the adaptive discovery mechanism (ADM). Results are highlighted in green and red help to identify desirable and undesirable situations while scanning, for example, high latencies or low success rates will be highlighted in red.

Simulation results will be independently introduced on this chapter, so as to present a general discussion in section 4.2.3 and further raise conclusions and perspectives for future work on the next chapter.

4.2.2 Proposed Simulation Scenarios and Results

SCE1: Random Scenario

Description In order to have a first view of the adaptive behavior, a set of random scenarios was designed. In this case, a mobile station leaves the coverage area of its current access point, and initiate a discovery process in a scenario where seven different access point are located. These seven access point changes their location, range and operating channel every time a new simulation is initiated. A set of four hundred simulations for each approach will be launched, so as to evaluate an heterogeneous spectrum.

The necessity of configuring a totally random scenario is based on the behavior of the adaptive algorithm, while RSSI calculations use distances, positions and ranges of mobile stations and access points.

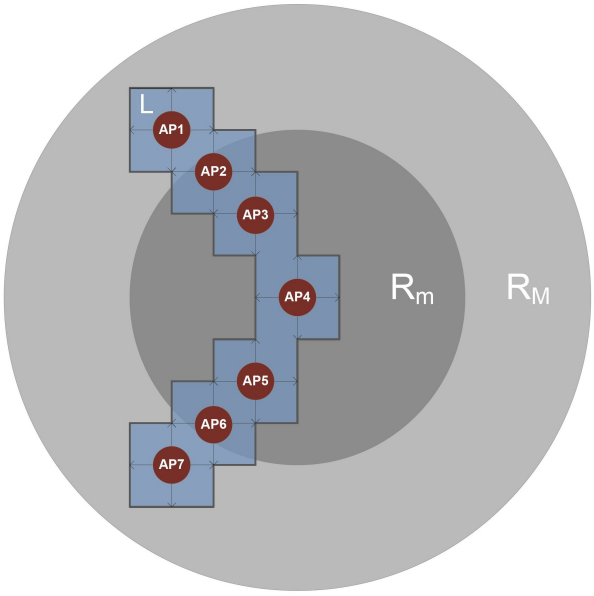


Figure 4.4 A set of random scenarios

Fig. 4.4 shows the proposed scenario. R_m and R_M delineate the minimum and maximum ranges employed individually by the access points ($AP1$ to $AP7$). Then, the blue colored area labeled as L defines the potential location area of the access

points. Each access point is able to move within the limits indicated by gray arrows. Simulations will be run intercalated, so an adaptive discovery simulation will be preceded by an active scanning approach, using the same channel allocation, ranges and access point physical location for both approaches.

Moreover, fig. 4.5 illustrate one of the possible random cases to be evaluated, in which the mobile station, labeled as MS moves from its current access point area and initiates the scanning phase in a point where signal from all the access points is received.

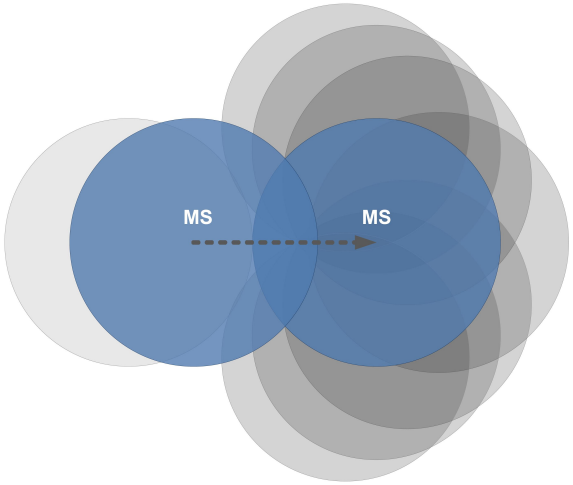


Figure 4.5 A potential random scenario

For the active scanning algorithm both *MinChannelTime* and *MaxChannelTime* are set considering values referred in [19], that were detailed in section 2.7.3. $1TU$ and $10TU$ are respectively assigned. On the other hand, for the adaptive behavior, values established in section 3.2.2 are applied. Thus, $1.8TU$ and $10TU$ are set as the initial maximum thresholds.

Results Figure 4.6 presents the simulation results and figures 4.7 and 4.8 illustrate the behavior of the adaptation process for both *MinChannelTime* and *MaxChannelTime*. In this case, the adaptive discovery latency (labeled ADM) is lower than the active scanning one (labeled AS). In both fig. 4.7 and fig. 4.8, we can see the reduction of both timers while the scanning goes on. There is not failure (no access point discovered after scanning 14 channels) in both approaches.

	SCE1: RANDOM SCENARIO						
	LATENCY					DISCOVERY RATE	RSSI
	DURATION (µs)	VARIATION (%)	FAILED HDV	SUCCESS RATE (%)	σ	VALUE (%)	VALUE (%)
AS	64353,63	-38,32	0	100	10651,70	86,54	48,87
ADM	39695,50		0	100	6353,05	85,18	48,78

Figure 4.6 SCE1 - Simulation Results

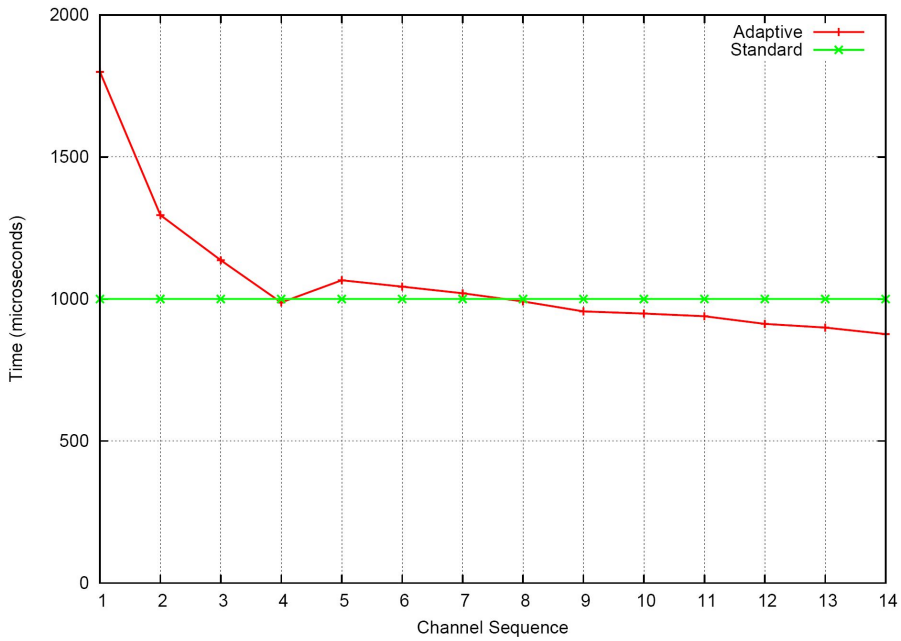


Figure 4.7 SCE1 - Adaptive MinChannelTime Profile

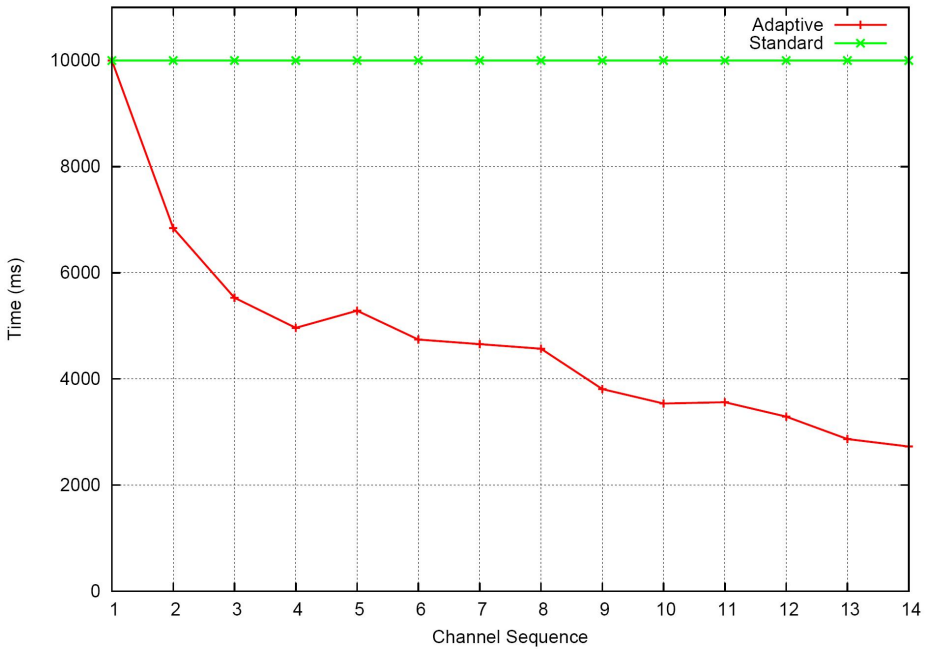


Figure 4.8 SCE1 - Adaptive MaxChannelTime Profile

SCE2: Non Overlapping Hexagonal Pattern

Description As a second step, a non overlapping hexagonal deployment is designed and illustrated in 4.9. Channels 1, 6 and 11 are assigned so as to minimize the interference between them. A mobile station leaves the coverage area of $AP1$ so as to start a discovery process. Applying the same rules than in the random case, an adaptive approach is performed after a standard active scanning. Timer are also configured as in the random simulation. This proposed scenario helps to evaluate the discovery process under an ideal deployment condition, in which only one operator assures the non overlapping coverage so as to maximize the throughput.

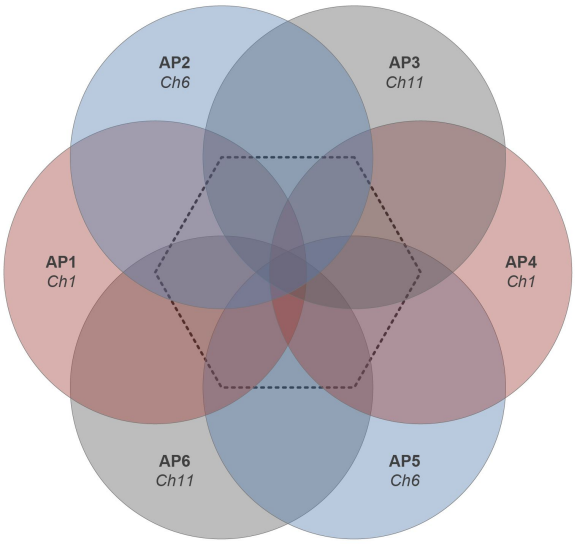


Figure 4.9 Non overlapping hexagonal simulation scenario

Results Figure 4.10 presents the simulation results and figures 4.11 and 4.12 illustrate the behavior of the adaptation process for both *MinChannelTime* and *MaxChannelTime*. In this case, the adaptive discovery latency is still lower than the active scanning one, but only a $9,11\%$. In both fig. 4.11 and fig. 4.12, we appreciate that access points are discovered in the first three scanned channels (1, 6 or 11), and so timers are reduces. Then, no access point is discovered and timers are smoothly increased. There is not failure in both approaches.

	SCE2: NON OVERLAPPING HEXAGONAL PATTERN						
	LATENCY					DISCOVERY RATE	RSSI
	DURATION (µs)	VARIATION (%)	FAILED HDV	SUCCESS RATE	σ	VALUE (%)	VALUE (%)
AS	36107,94	-9,11	0	100	7195,93	64,90	7,60
ADM	32818,07		0	100	1952,66	67,80	7,60

Figure 4.10 SCE2 - Simulation Results

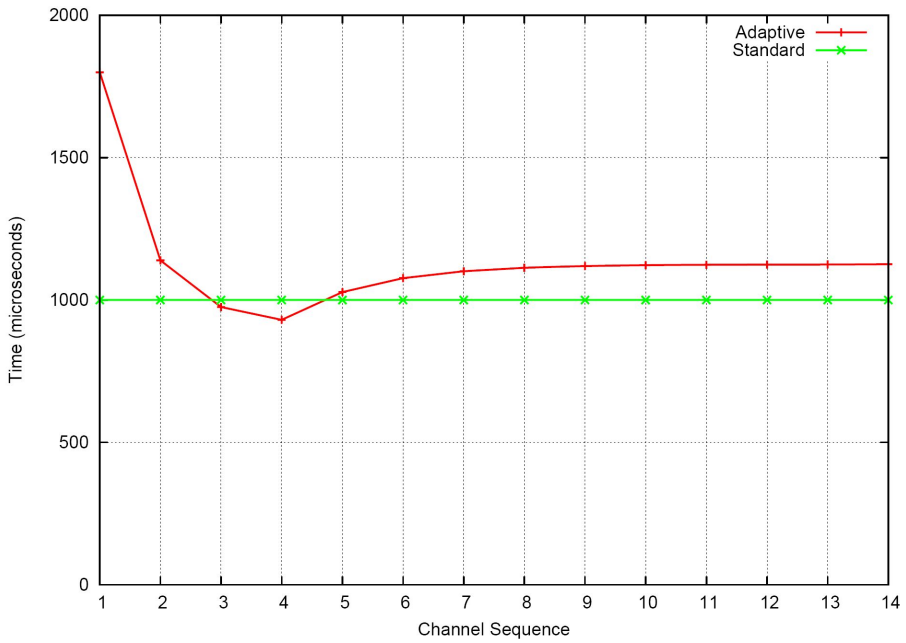


Figure 4.11 SCE2 - Adaptive MinChannelTime Profile

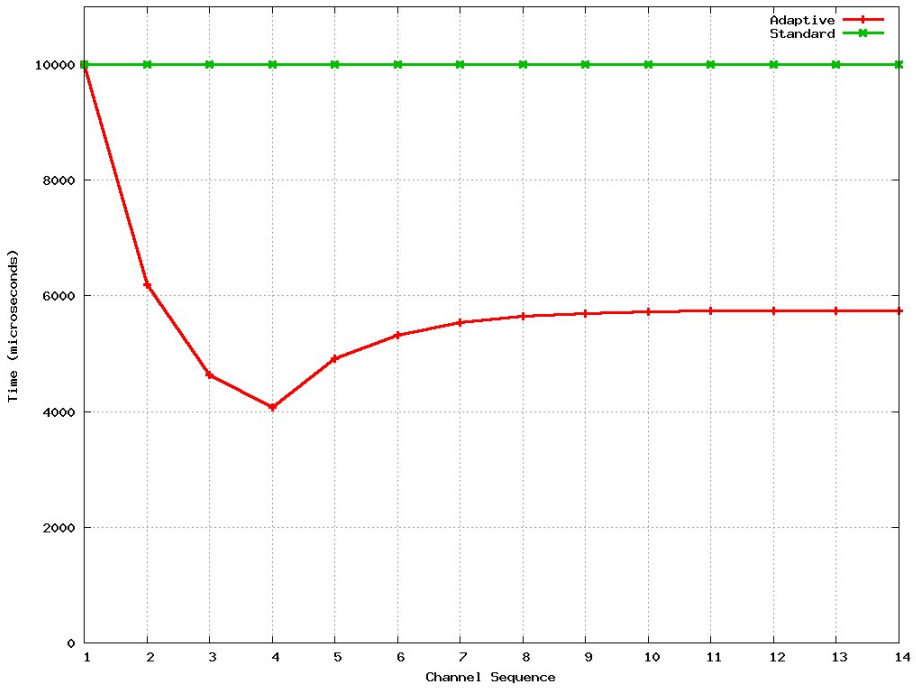


Figure 4.12 SCE2 - Adaptive MaxChannelTime Profile

SCE3: Overlapping Hexagonal Pattern

Description Following the same line of attack, the same access point configuration presented in SCE2 is applied using a complete channel overlapping. All deployed access points are randomly arranged in channel R (as illustrated in fig 4.13). So the same channel R is used to simulate both active and adaptive scanning one after the other. As an hypothesis, a high interference level and further frame collision is expected. Results are evaluated in section 4.2.

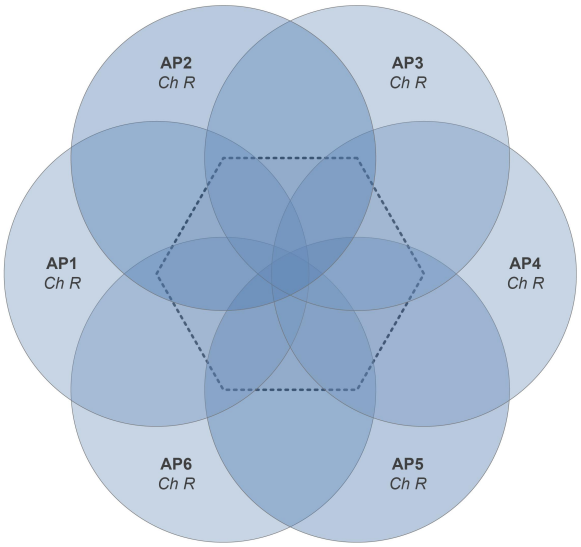


Figure 4.13 Overlapping hexagonal simulation scenario

Results Figure 4.14 presents the simulation results and figures 4.15 and 4.16 illustrate the behavior of the adaptation process for both *MinChannelTime* and *MaxChannelTime*. In this case, the adaptive discovery latency is higher than the active scanning one. In both fig. 4.15 and fig. 4.16, we appreciate timers reduction, but as the access points are set in the same channel, the adaptive system cannot reduce timers as much as on SCE2. It can also be appreciated that active scanning fails in $63,5\%$ of the cases, while active scanning only fails in $13,75\%$ of the cases.

	SCE3: OVERLAPPING HEXAGONAL PATTERN						
	LATENCY					DISCOVERY RATE	RSSI
	DURATION (µs)	VARIATION (%)	FAILED HDV	SUCCESS RATE	σ	VALUE (%)	VALUE (%)
AS	25659,53	36,27	254	36,5	442,26	65,07	1,73
ADM	34965,29		55	86,25	555,03	63,60	1,78

Figure 4.14 SCE3 - Simulation Results

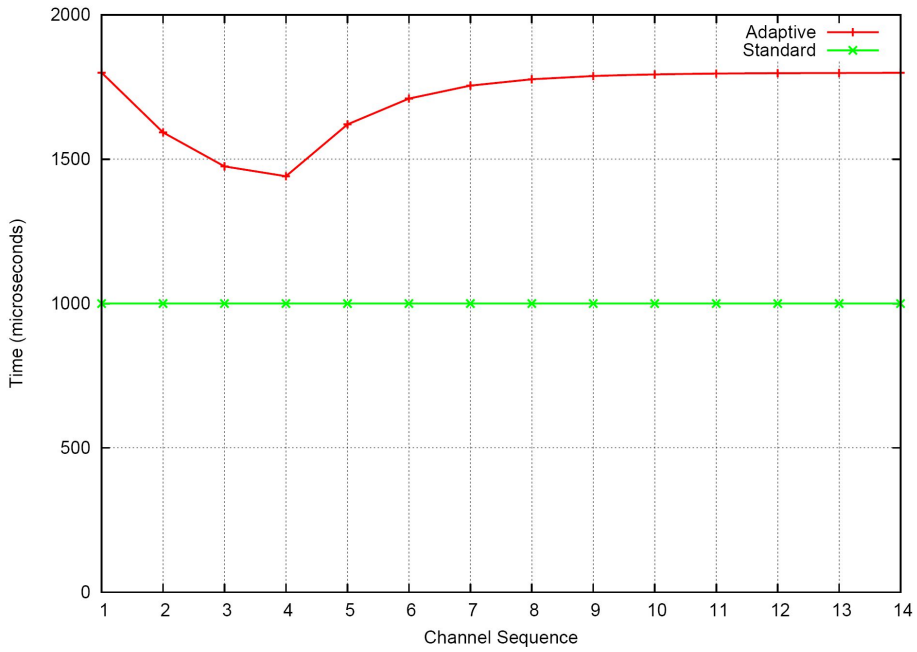


Figure 4.15 SCE3 - Adaptive MinChannelTime Profile

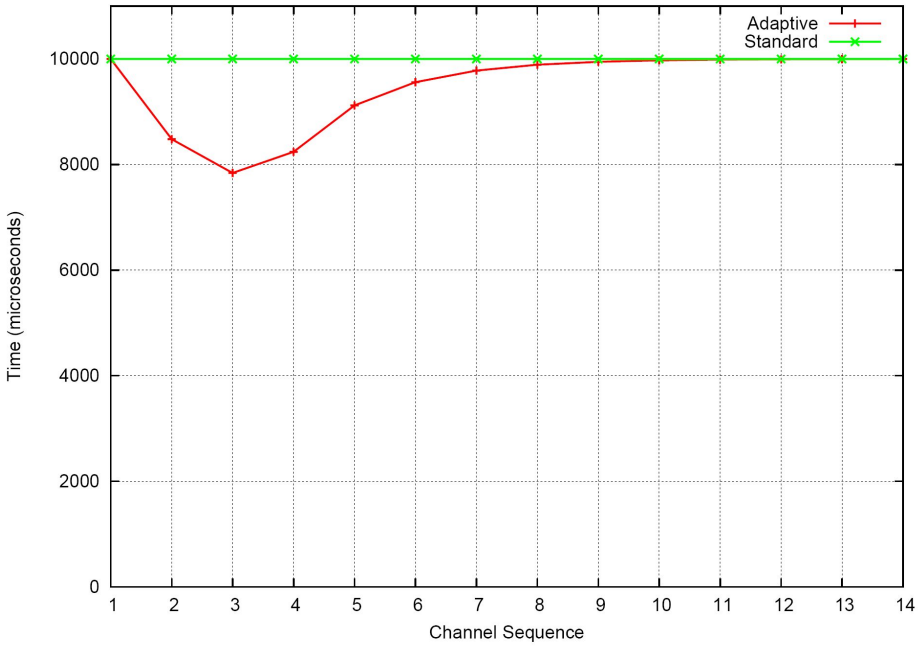


Figure 4.16 SCE3 - Adaptive MaxChannelTime Profile

SCE4: Overcrowded Scenario

Description Following the scenario described in figure 4.4, a special case is configured, so several simulations will be run using fixed access point locations and ranges. Moreover, channel allocation is randomized as the case of SCE3, and simulations are run alternatively.

Results Figure 4.17 presents the simulation results and figures 4.18 and 4.19 illustrate the behavior of the adaptation process for both *MinChannelTime* and *MaxChannelTime*. In this case, we highlight a 100% of failure while using active scanning, if several access points are set in the scenario, active scanning does not allow the mobile station to receive the first probe response. We see that the adaptive discovery mechanism uses higher timers on this scenario.

	SCE4: OVERCROWDED SCENARIO						
	LATENCY					DISCOVERY RATE	RSSI
	DURATION (µs)	VARIATION (%)	FAILED HDV	SUCCESS RATE	σ	VALUE (%)	VALUE (%)
AS	N/A	N/A	400	0	N/A	N/A	N/A
ADM	35952,76		0	100	430,06	77,57	11,42

Figure 4.17 SCE4 - Simulation Results

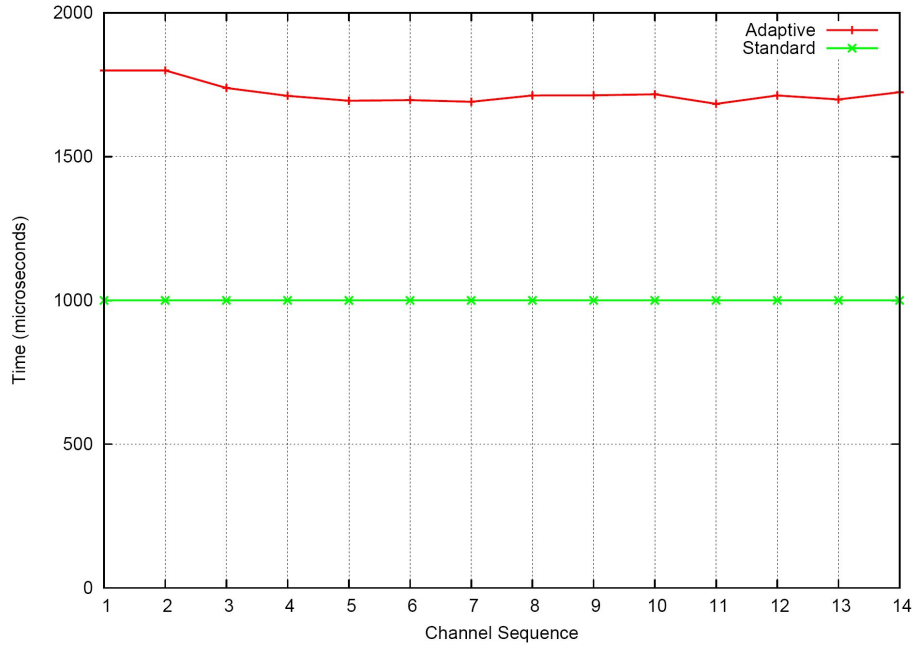


Figure 4.18 SCE4 - Adaptive MinChannelTime Profile

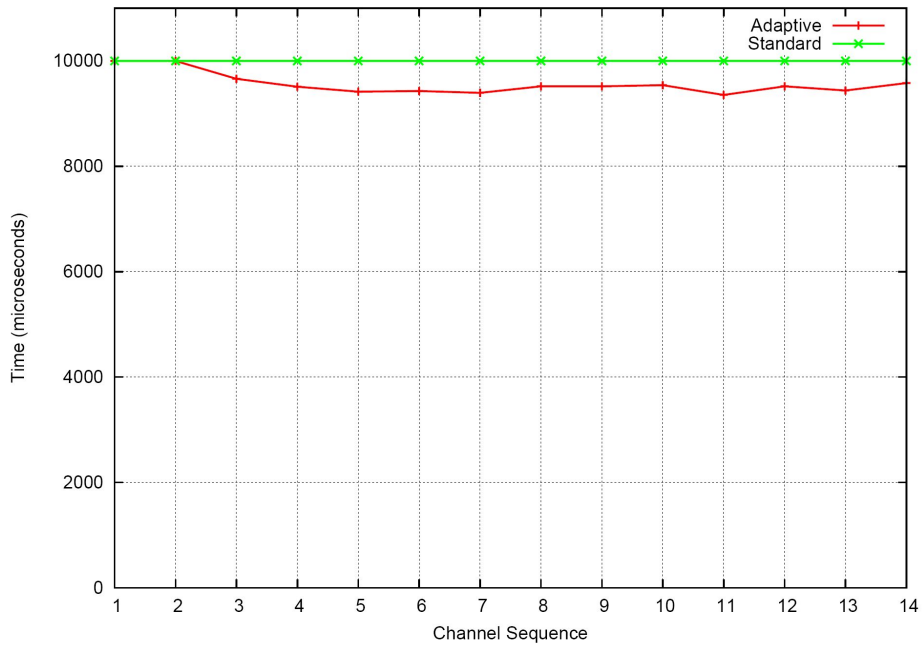


Figure 4.19 SCE4 - Adaptive MaxChannelTime Profile

4.2.3 Discussion

Results attained in the first proposed scenario, in which a random behavior is evaluated, show that average latency reductions are obtained reaching a $38,32\%$ (see fig. 4.6) compared with the standard active scanning approach. Moreover, when analysing the standard deviation (σ) for both cases, an homogeneus comportment is appreciated. Thus, as σ is often used as a measure of risk, in this particular case, low values of σ for the adaptive algorithm assure that the risk of finding high scanning latency values is quite reduced. Depending on the particular case within all possible random instances, not always all present access points in the scenario are discovered. Moreover, when analyzing the discovery rate, both approaches discover almost the same number of access points on each performed scanning process. A rate equal to $85,18\%$ (see fig. 4.6) for the adaptive mechanism indicates that in average, six different access points are discovered on each simulation instance. On the other hand, it is important to emphasize that only in six cases over eh hundred simulations, the adaptive behavior associate with a different access point than the standard discovery algorithm, obtaining almost the same average signal strenght.

It is interesting to analyse the obtained profiles of *MinChannelTime* and *MaxChannelTime* (figs. 4.7, 4.8, 4.11, 4.12, 4.15, 4.16, 4.18 and 4.19), since it can be appreciated the evolution of the adaptive process during the full channel sequence. In this particular case, both values are continuously decreasing with small rising peaks when no access point is discovered in that channel. Strong reductions for both variables are obtained before the fourth channel is scanned. This is because of the completely random channel allocation and the two phases scanning sequence. In order to make a comparison between the standard approach, in which fixed values are considered, a mobile station applying the adaptive behavior uses as an average value for *MinChannelTime*, $1093\mu s$ and $4740\mu s$ for *MaxChannelTime*.

Because of the significance of the hexagonal deployment in 802.11 wireless networks, some results have to be evaluated under this scenario condition. In the first case, considering that a single operator is responsible to manage the channel allocation of all available 802.11 network in the area, a non overlapping deployment is considered so interference is reduced to the minimum expressions. Despite of those optimal conditions, the adaptive approach continues giving better latency values for the scanning process, so a reduction of $9,11\%$ (see fig. 4.10) is observed. In this case, better values are also obtained for σ in the adaptive approach. Signal strenght of the selected access point is equal in both cases, but in the proposed mechanism a superior discovery rate is obtained.

The benefits of using a scanning sequence that weights the non overlapping channels helps to reduce the latency. After channels 1, 6 and 11 are scanned no more access points should be discovered, so *MinChannelTime* and *MaxChannelTime* profiles will

have an asymptotic comportment to the lower values that those variables had taken when an access point was discovered. In this case, the average *MinChannelTime* used is close to $1100\mu s$ and $MaxChannelTime$ is decreased to an average of $5764\mu s$.

Based on the same network deployment and considering the case in which there is no a unified network management, an scenario were hexagonal positioned access points are not well configured may appear. As it can be appreciated, using the standard active scanning algorithm, the handover is not successfully performed in more than 63% ((see fig. 4.14) of the cases. This situation is produced due to the high level of colisions, that forbid the mobile station to discover a candidate access point in a time equal to the optimal proposed, $1024\mu s$. The adaptive behavior presents a lower failure rate, close to 13%. However, when using the adaptive mechanism, successful handovers are in average lower to those performed using the standard approach.

The analysis of *MinChannelTime* and *MaxChannelTime* evolution during the scanning process show that these timers are initially reduced but then they are increased again to the initial value. s it can be appreciated, this situation will take place whenever the network deployment corresponds to a single-channel allocation.

Finally an overcrowded scenario is studied ((see fig. 4.17). In this case, frame collision becomes a major factor. This situation can be recognized due to the nonexistence of successful handovers using the scanning algorithm. A value of $1024\mu s$ for *MinChannelTime* is not enough so as to deal with multiple *Probe Response* retransmissions by access points. On the other hand, the proposed behavior could adapt scanning timers in order to efficaciously perform handovers in all cases. However, with the aim of reducing the scanning time, high timer values are used, producing greater latencies.

4.3 Experimentation

Different simulation's scenarios were designed and executed using the SimulX Wireless Network Simulator, obtaining interesting results. In order to validate them in a real 802.11 environment, the proposed Adaptive Mechanism was implemented using the *MADWiFi* 802.11 open-source Kernel Device Driver. Afterward, different wireless scenarios were configured using real equipment and the adaptive behavior was evaluated. We present empirical results based on experimentation with the aim of evaluating the proposed scanning approach, but we will not analyze the gap between simulation and experimentation results.

Section 4.3.1 presents an overview of the MADWiFii driver and the adaptive algorithm implementation. Then section 4.3.2 details the evaluation experience, using 802.11 devices [26].

4.3.1 MADWiFi 802.11 Driver

The *Multiband Atheros Driver for WiFi (MADWiFi)* project[5] appears as the most popular open-source kernel device driver for 802.11 WLAN. MADWiFi provides support for users using *Atheros* based equipment. Despite the driver is open source itself, it depends on the proprietary Hardware Abstraction Layer (HAL) that is available only in binary code.

In 2004, Sam Leffler, former developper of the FreeBSD driver for Atheros, decided to port it to the Linux platform. After a year, Mr. Leffler left Atheros and the MADWiFi project became a colaborative project, involving several developers all around the world.

Up to now, the MADWiFi project team has released several versions of the driver, maintaining it updated to the 802.11 amendments and new functionalities. The current MADWiFi version is v0.94 and includes the 802.11n draft specifications.

One of the most interesting functionalities of MADWiFi is the inclusion of a Debugging system, that allows a developer to capture and visualize real-time driver parameters on the console. This functionality allows obtaining information about different processes independently, since it implements a multi-flag mechanism. Different flags allows identifying scanning, association, authentication and packet handling information among others.

MADWiFi Scanning Algorithm

Inside the MADWiFi core, the mobile station's active scanning implementation is included in the *ieee80211_scan* and *ieee80211_scan_sta* source code and header files, since it is implemented in C programming language.

After a deep analysis of the current MADWiFi scanning implementation, it has been appreciated that it does not rigorously follow the specification established in the 802.11 standard [1]. The main functions regarding the MADWiFi active scanning process are included in *ieee80211_check_scan*, *ieee80211_start_scan*, *scan_next* and *ieee80211_add_scan*.

Following the same ideas than those established in the standard, two time control variables *ss_mindwell* and *ss_maxdwell* manage the scanning procedure and so the time to spend on each channel so as to wait for Probe Responses. But then, there is not a direct correlation between MADWiFi timers and both standards *MinChannelTime* and *MaxChannelTime*.

[5]Official website: http://madwifi-project.org

A mobile station performing active scanning acts as follows. It sends a Probe Request on a particular channel and then it waits until *ss_maxdwell* expiration for a Probe Response or Beacon. Current MADWiFi implementation defines 200ms as a concrete value for *ss_maxdwell* and 20ms for *ss_mindwell*. Then, if information from an access point is received before *ss_maxdwell* expiration, and if *ss_mindwell* elapsed, the station immediately switches to the next channel in the sequence. So a mobile station waits for *ss_maxdwell* in channels without activity. Then *ss_mindwell* acts as the minimum time to wait for responses on channels where access points are operating.

Since the proposed adaptive scanning implementation is based on modifications inside the 802.11 standard active scanning specification, and because experimental results should clearly contrast the latter versus the former, both approaches were to be implemented under the MADWiFi platform.

Adaptive Scanning Implementation

In order to include the adaptive scanning under MADWiFi, the channel switching sequence was modified. MADWiFi uses the following fixed 802.11 b/g channel sequence: [1 - 6 - 11 - 7 - 13 - 2 - 3 - 4 - 5 - 8 - 9 - 10 - 12]. Then a randomized channel sequence, like that proposed in section 3.2.3, was implemented.

Afterward, the behavior of both *ss_mindwell* and *ss_maxdwell* was modified so as to force them to act like *MinChannelTime* and *MaxChannelTime* standard timers. Lower and upper timers' thresholds like those introduced in 3.2.2 are statically set in the code.

The main functions controlling the scanning sequence were amended considering that each time before channel is switched, probe responses received on the current channel are analyzed so as to decide the adaptation rate to be applied to the timers for the next channel.

Finally, inside the MADWiFi implementation, timers are set using the *mod_timer* kernel function. Since the adaptive implementation reduces timers's values down to the lowest as possible, a modification in the kernel timeslice time *HZ* was performed so as to allow to set timers down to 1ms. The HZ parameter was fixed in 1000 Hz, so a complete kernel compilation was carried out.

For testing purposes, the background scanning function, that maintains information of responder access points in the scanning table, was avoided. Doing this, the independence of each scanning process to evaluate was assured.

Output Generation

As it was introduced above, the MADWiFi debugging tool was used so as to obtain scanning outputs. For this purpose, only the *scan* flag was activated and, inside the adaptive scanning implementation, useful information to be shown was referenced using this flag.

Then, a set of scripts were implemented so as to accomplish different purposes. The scanning process was performed simply asking the wireless network card to associate to an access point using a particular SSID. This instruction was performed using the *iwconfig* command.

Output files were generated using the *dmesg* linux command for the kernel output, and saving it as a text file. All output generated by this command shows an absolute kernel timestamp, that allow as to perform time related calculatios. Then, graphics, mathematical calculations and other results were obtained using ad-hoc scripts developed in *Perl*.

4.3.2 Testbed Configuration

The proposed testbed was designed considering the architecture illustrated in fig 4.20. Up to thirteen access points were configured using a single SSID and deployed under the same IP subnet. On the other hand, four mobile stations were set so as to generate traffic using the Distributed Internet Traffic Generator (D-ITG) [32].

A single mobile station using an Atheros based DLINK DWL-AG660 was used as scanner. Up to thirty different network scenarios were evaluated using different channel allocations, traffic conditions and *MinChannelTime* and *MaxChannelTime* standard timers.

With regard to the channel allocation, the aim was to evaluate both commonly implemented and particular scenarios. In all cases, the influence of traffic load was taken into consideration. Table 4.2 describes different access points' configuration deployed in the testbed.

In each of our experiments, the discovery latency was measured over a hundred of full scanning processes, i.e., where all channels are scanned by the mobile station one by one. Outputs have been obtained using the MADWiFi's *debugging* utility and analyzed using *Perl* scripts and *gnuplot* [33] for graphical considerations.

The experiment was divided in two phases. In a first stage, all scenarios were evaluated using high *MinChannelTime* and *MaxChannelTime* standard timers in order to

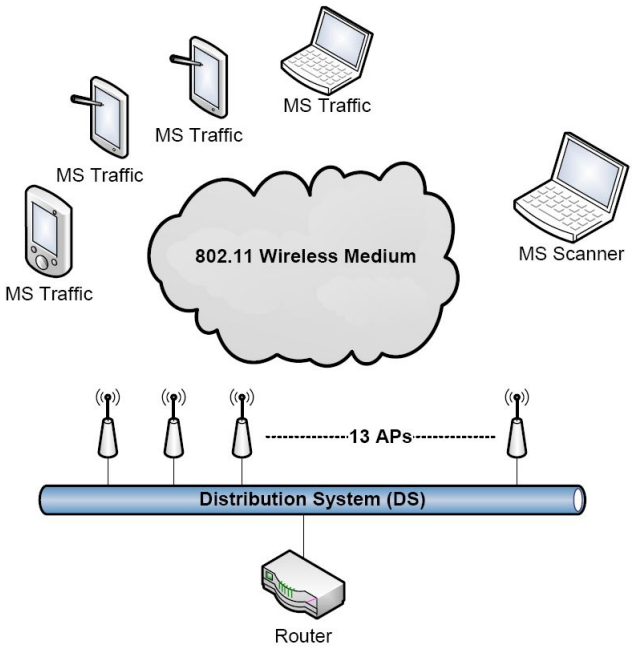


Figure 4.20 Testbed Architecture

establish reasonable values for both upper and lower limits. During the second phase, the adaptive scanning was evaluated in all scenarios against the standard algorithm using fixed timers.

The following sections will detail both phases and results will be presented and discussed.

Table 4.2 Scenarios' Configuration

Conf. 1	13 APs allocated one by one on channels 1 to 13 (one AP per channel)
Conf. 2	13 APs all allocated on channel 11 (13 APs on the same channel)
Conf. 3	3 APs allocated one by one on channels 1, 6 and 11 (one AP per channel)
Conf .4	12 APs allocated four by four on channels 1, 6 and 11 (4 APs per channel)

Threshold Determination

In the first part of the experience, the aim is to determine upper and lower thresholds for *MinChannelTime* and *MaxChannelTime* between which the adaptive algorithm will work. For this purpose, the delay of the *first* and *further* received probe responses was measured on each channel for each particular access point configuration, with and

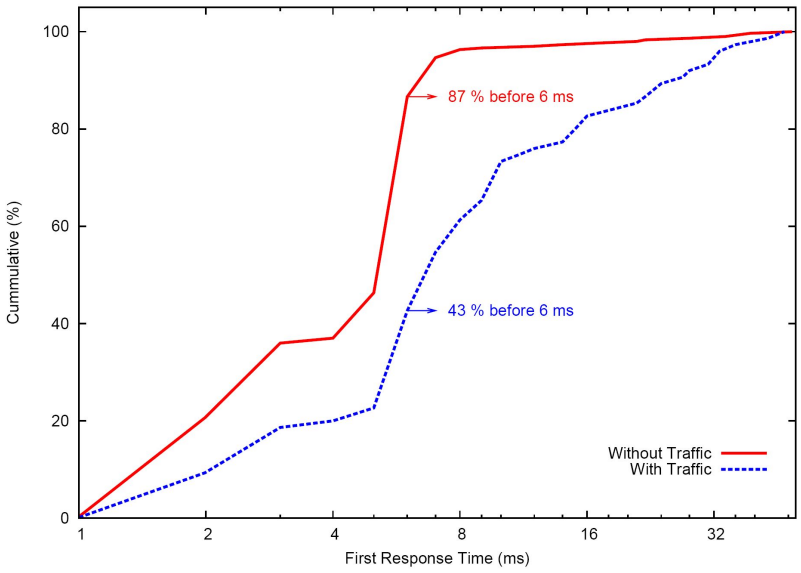


Figure 4.21 First probe response's delay in configuration 3

without traffic.

It has to be appreciated that a channel is considered as active only if a probe response from an access point is received before *MinChannelTime* expiration. Then, the adaptive algorithm should prevent the situation where no access point is discovered after scanning all channels, while giving a controlled latency. Then, further probe responses are those that arrive after the first and until the last one before *MaxChannelTime* expiration.

The mobile station acting as scanner was configurer with high timers, i.e. 50ms for *MinChannelTime* and 200ms for *MaxChannelTime*, so as to have enough time to discover all active access points. After evaluating all deployed scenarios several histograms were built, as illustrated in fig. 4.21, so lowest and highest thresholds could be identified. In this case, fig. 4.21 represents the delay of the reception of first probe responses in configuration 3 where a total of three access points were operating in channels 1, 6 and 11. The influence of traffic load falls in higher delays of the first probe response. While in the 87% of the cases first probe responses are received before 6ms in a scenario without traffic load, during a loaded scenario only the 43% of the cases first probe responses are received within the same period of time (6ms).

All obtained histograms were evaluated, and so thresholds have been stablished as indicated in table 4.3. *MinLower* and *MinUpper* represent minimum and maximum thresholds for *MinChannelTime*, while *MaxLower* and *MaxUpper* describe those for *MaxChannelTime*.

Table 4.3 Thresholds for *MinLower*, *MinUpper*, *MaxLower* and *MaxUpper*

Threshold	Value	P.Resp. Received	Conf.	Traf.
MinLower	6 ms	87%	3	No
MinUpper	34 ms	96%	1	Yes
MaxLower	8 ms	50%	4	No
MaxUpper	48 ms	87%	2	Yes

The scenario concerned in fig. 4.21, where only three access points in non-overlapping channels are deployed, is considered as an ideal configuration where interferences are minimized and thus helps to determine the minimum limits for *MinChannelTime*. Analysing the accumulated percentage function of first probe responses received over all the trials without traffic, it can be appreciated that 87% of the first probe responses were received before 6 ms. Thus we decide to set *MinLower* at 6 ms as stated in table 4.3. It can be allowed this relative low percentage (8 ms could also be taken, where 96% of the probe responses were received) because it can be afforded to risk few unsuccessful discoveries using the adaptive algorithm when this minimum value is used. This minimum value (6ms) would only be picked once access points in other channels have already been discovered.

With the same aim and considering configuration 4, without traffic, *MaxLower* is set at 8 ms where 50% of following probe responses from other access points were already received. Then, *MaxChannelTime* can be adapted down to a low limit that covers less cases than *MinChannelTime* (only a 50%), since the situation of not discovering all access points is not as risky as not discovering the first access point, in which the channel will be declared empty.

On the other hand, upper thresholds have been obtained evaluating further probe response's delay for undesirable scenarios, like configurations 1 and 2. For the case of *MinUpper* a value of 34ms was set since 96% of further probe responses were received in configuration 1. Then *MaxUpper* was set at 48ms, because 87% of further probe responses were received under configuration 2.

4.3.3 Results

During the second part of the experience, the adaptive system was tested using thresholds defined in table 4.3 and the standard scanning was evaluated considering three different sets of timers, [10 - 20ms], [25 - 50ms] and finally [50 - 200ms] for *MinChannelTime* and *MaxChannelTime* respectively. Table 4.4 shows the results organized by scenario, where the *full-discovery rate* indicates in how many scanning processes all available access point were discovered. The *failed scanning* values describe in how many cases no access point is found during the full discovery phase (scanning of all

Table 4.4 Comparative results

Scenario			Full-Discovery Rate (%)				Failed Scanning (%)				Average Scanning Latency (ms)				
AP conf.	No APs	Traffic	STD			ADP	STD			ADP	STD			ADP	
			10-20	25-50	50-200	Dyn.	10-20	25-50	50-200	Dyn.	10-20	25-50	50-200	Dyn.	σ
1	13	No	65%	87%	93%	49%	0	0	0	0	275	708	2567	256	11%
1	13	Yes	24%	69%	82%	40%	2%	0	0	0	317	636	2378	248	13%
2	13	No	75%	92%	94%	96%	2%	2%	0	0	152	360	807	423	3%
2	13	Yes	54%	88%	98%	83%	29%	3%	0	2%	159	363	814	434	5%
3	3	No	92%	94%	99%	94%	0	0	0	0	117	414	1119	190	11%
3	3	Yes	38%	51%	61%	81%	52%	20%	13%	0	227	403	1025	210	18%
4	12	No	98%	98%	100%	95%	0	0	0	0	179	419	1121	390	3%
4	12	Yes	39%	60%	87%	84%	13%	1%	0	0	239	450	1110	378	13%

channels). Finally the average *scanning latency* describes the time to scan all channels, including the standard deviation for the adaptive system latencies, which shows a controlled dispersion of the obtained latencies.

Throughout this work, it was repetitively stated that the standard scanning and further optimized mechanisms fall in a narrow network deployment dependence. Experience results prove the high number of unsuccessful scanning when using a standard active scanning approach in some common network scenarios. The adaptive system only has 2% of failure in a single scenario (configuration 2 and loaded cells) and keeps low scanning latencies. A detailed analysis of results presented in table 4.4 is presented in the following sections.

Impact of Traffic

Fig. 4.21 illustrates configuration 3, where probe responses are notably delayed when traffic is injected. While before 6 ms the 87% of the probe responses are received in non loaded scenario, only the 43% is received when traffic is introduced. As shown in table 4.4, in the case of configuration 3 with traffic, in several scannings a probe response is not received before 25 ms, causing 20% of scanning failure. Even using a *MinChannelTime* equal to 50 ms the failure rate arrives to 13%. The effect of traffic also produces a diminuition in the average number of discovered access points in all evaluated scenarios. The adaptive system helps to reduce the effects of traffic, since no scanning process fails except in one scenario, where only 2% of failure was observed.

Impact of Number of Access Points

In configurations 3 and 4 where only non-overlapping channels were used, less scanning failures were observed when there are four access points operating on the same channel (configuration 4). When there is a single access point per channel (configuration 3), higher failure rates are attained in all evaluated timers for the standard algorithm. This may be due to the operation of the backoff algorithm of the MAC protocol, since there are more chances to pick a small random number when there

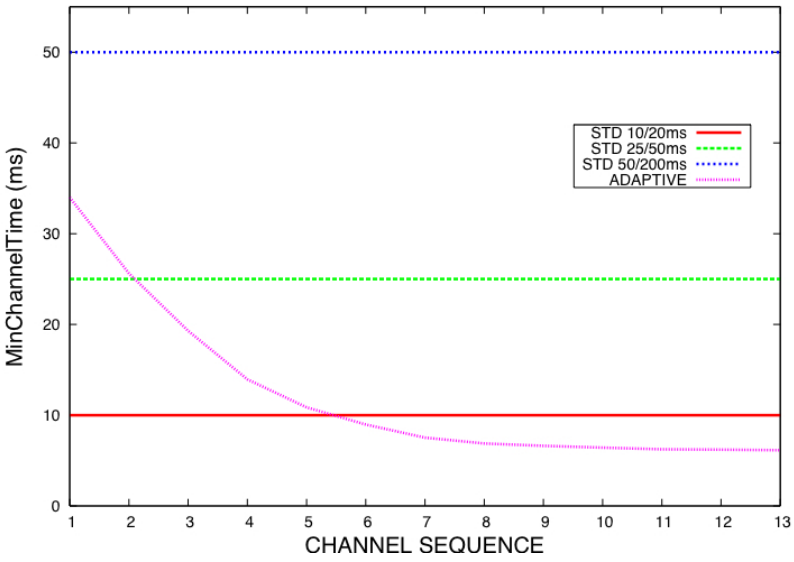


Figure 4.22 MinChannelTime values for configuration 1 with traffic

are more answering access points.

Latency Analysis

Scanning latencies are related with values taken by both *MinChannelTime* and *MaxChannelTime* during the discovery phase. Fig. 4.22 shows different average values taken by *MinChannelTime* in the case of configuration 1, with traffic load. In the adaptive system it can be appreciated that *MinChannelTime* is initially set to *MinUpper* and it gradually decreases down to *MinLower*. On the other hand all other evaluated algorithms implement constant values for *MinChannelTime*.

Fig. 4.23 shows latency values for all configurations with traffic including the failure rate for each case. Even if the standard scanning using fixed timers may give good latency results in some scenarios, our adaptive system provides lower or equivalent scanning latencies from 190 ms to 434 ms. The standard scanning algorithm configured with [10 - 20 ms] gives significant better latencies in access point configuration 2 around 150 ms against 420 ms for the adaptive system. But in this case failure rate reaches 29%, while the adaptive system only produces a 2% of failure. Moreover, in configuration 3 with traffic, the adaptive system gives the best scanning latency without any scanning failure, while all other evaluated algorithms reach high levels of failure, up to 52%.

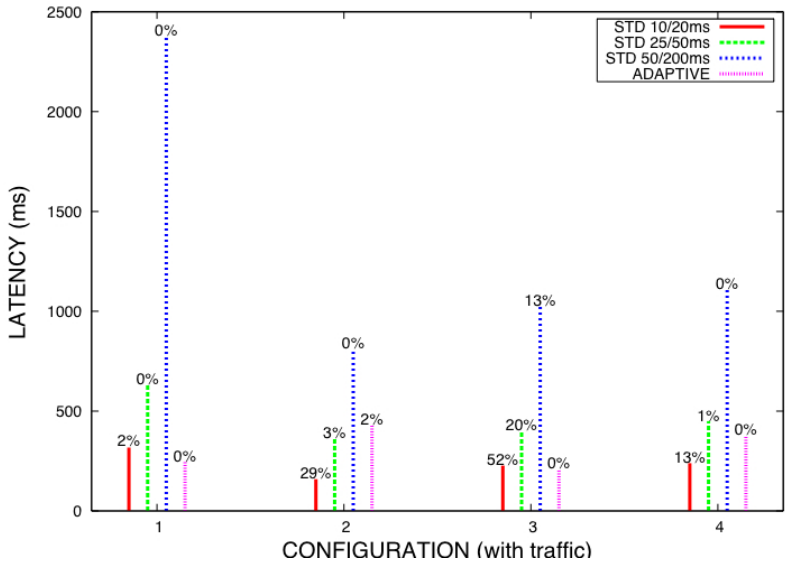


Figure 4.23 Latency values for different configurations and timer values considering traffic

4.3.4 Discussion

The implementation of a real testbed appears as a reliable tool to contrast simulation results. As it was discussed in section 4.1.1, simulations platforms are widely used in the networking ambiance, because for some deployment scenarios there is not feasibility to implement a real deployment for testing purpose. In the domain of study proposed by this work, the testbed was relatively reachable, since simple available access points and mobile stations were used to build it.

As the adaptive algorithm dynamically set scanning timers between predefined thresholds, *MinChannelTime* is able to take values between 6 ms and 34 ms and *MaxChannelTime* between 8 ms and 48 ms. The proposed adaptive system scans first using high values for both timers, and then decreases these values when some access points were already found. This helps to perform effective handovers, independently of the particular network deployment. It was showed that in almost all presented scenarios that, the adaptive system offers a better percentage of discovered access points, minimizes the number of full scanning failures (at maximum 2%), and keeps a low and controlled scanning latency (between 190 ms to 434 ms). This demonstrates the importance and the efficiency of using an adaptive system adapting to all possible scenarios, instead of defining an static algorithm which only fits some access point deployment configurations.

The real testbed allowed evaluating the influence of traffic on probe response's delay. It has to be stated that previous simulation results did not focus on the effect

of traffic in the scanning procedure, so its real influence was discovered implemented the testbed.

Chapter 5

Conclusion and Perspectives

Throughout this document, it was repetitively stated that proposed scanning standard mechanisms fall in a narrow network deployment dependence. Simulations presented in section 4.2 and experimental results analyzed in section 4.3.2 clearly prove the high number of unsuccessful handovers when using an active scanning approach in some common network scenarios.

The main purpose of a standard scanning algorithm should guarantee a wide level of application considering all possible network scenarios. Moreover, because the scanning feature is embedded in wireless network interfaces, its implementation has to be unique. It is impossible to implement a set of different algorithms applying different techniques so as to optimize every possible network deployment.

The general application of the adaptive behavior in the scanning process helps to perform effective handovers, independently of the particular network deployment. A successful handover appears more important than a time reduced handover. A temporal interruption is always preferred than a permanent disconnection while looking with the eyes of an application user.

This independence is extremely related with the channel configuration in wireless networks. As it was proposed in several optimization techniques, a selective scanning approach not only reduces the scanning latency, but it conditions the successfulness of the handover process as well. An heterogeneous channel allocation must be always supposed.

As important as the general application of the adaptive behavior is the low invasive deployment of this technique, because only modifications on the client side are required. Access points should continue answering to *Probe Request* using *Probe Response* management frames. However, an optional independent optimization has been introduced in section 3.3.2 requiring implementations in the access point side.

The proposed adaptive behavior is based on a simple linear factor reduction. F_I and F_R are employed so as to adapt *MinChannelTime* and *MaxChannelTime*. Because of this condition, two considerations should be stated. First, an accurate sensibility analysis of these factors may be performed so as to enhance the adaptation process. On the other hand, deepen the adaptive systems theory, concepts about Neural Networks and Learning Algorithms should be introduced in order to improve the optimization.

Moreover, considering that different applications could tolerate different handover latencies interrupting their operations , a set of adaptive scanning strategies based on cross-layer information could be defined so as to use different lines of attack while scanning. This could allow a mobile station to take the risk of falling in a scanning failure while looking for faster scanning (for instance, a real time application over a 802.11 link) or, on the other hand, look for the best access point to associate with, without weighting so much the scanning latency.

Regarding proposed handover optimizations stated in section 3.3, they are considered as theoretical optimizations, and they have been introduced on the simulation for all cases, without analyzing the effect of not implementing them. They have not been introduced on the experimentation due to limitations on driver's implementation.

Simulations and the real testbed are based on an horizontal handover scenario. As the employed simulator does not implement several wireless links inside its core, a real test bed should be designed and evaluated using different network access technologies (as 3G, WiMax or other available wireless links) in order to consider the application of the adaptive solution in vertical handover scenarios.

This work gave me the opportunity to participate in an European Summer School and in an International Workshop. In the first case, a seven pages paper (annexed on appendix A) was submitted, accepted and presented in the *EUNICE 2008 14th Open European Summer School*. This conference was held at Brest, France in September 2008. In the second case, a two pages abstract (annexed on appendix B)was submitted in the student workshop organized by *IEEE INFOCOM 2009 Conference on Computer Communications*. The abstract was accepted by the scientific committee and I had the opportunity to present a poster in the workshop, held at Rio de Janeiro, Brazil. This allows me to share my work with other colleagues from all around the world.

Appendix A

EUNICE 2008 14th Open European Summer School

Adaptive Discovery Mechanism for Wireless Environments

German CASTIGNANI
IT / TELECOM Bretagne, France
UBA - Facultad de Ingeniería, Argentina
german.castignani@telecom-bretagne.eu

Nicolas MONTAVONT
IT / TELECOM Bretagne, France
nicolas.montavont@telecom-bretagne.eu

***Abstract*—The mobility necessity in the 802.11 wireless environment requires handover processes to be executed, so a mobile station must detect and associate to a new access point while moving. As network deployment conditions occur in a heterogeneous manner, a general Discovery Algorithm is required to satisfy all possible scenarios. This paper introduces a new line of attack to perform discovery processes applying an adaptive comportment.**

I. Introduction

In recent, the implementation of wireless technologies for computer-based communication systems has undergone a profund stage of growth. As a result, network deployments using the 802.11 standard, commercially known as Wi-Fi, have led the number of implementations in companies, industries and government offices, without regard to their size. One of the reasons that supports this adoption is the relatively low cost associated with equipments and installations using this technology. The standard provides a single MAC (Medium Access Control) layer employed by a set of different Physical layers, that differs between them in the frequency usage and modulation issues, providing different data rates and coverage areas. Therefore different classes of wireless devices have appeared in the market in recent years in order to satisfy several user necessities. Wide applications and dynamic operations of these devices requires great mobility, and this feature is not provided by common wired-based networks.

Mobility is a necessity, but in order to be successfully performed, the widest possible frontiers must be considered. To achieve this, some possible solutions may be applied. The first one is performed by increasing Access Points (APs) and Mobile Stations (MSs) Power, which results in a higher consumption of resources and becomes a vicious circle while the mobility concept is being limited itself because the user will not be able to move for a long period of time due to out-of-battery constraints. Another solution is to deal with the Roaming process, that allows users to move on a wide area, covered by multiple APs. This process requires Handover mechanisms based on concrete algorithms, so as to manage the migration from an old AP to a new candidate AP, focusing on minimizing the disconnection time of the MS and avoiding non-desired effects in the upper-layers.

One possible classification for Handover processes is related to the network layer it concerns. From that point of view we can refer to a Layer 2 or Layer 3 Handover. This article introduces a new and innovative *Adaptive Discovery Mechanism* in order to minimize the negative impacts on services and applications running under a Wireless environment during Layer 2 Handover Processes, satisfying mobility necessities for all possible network deployment. The proposed mechanism is achieved thanks to the introduction of a Value Adaptation Process for some handover-related variables of the scanning process.

This document is organized as follows: first, section II presents a brief overview of the IEEE 802.11 standard handover process and the associated handover latency. Section III provides an overview of current handover optimizations. Section IV focuses on the scanning variables definition so as to introduce the proposed adaptive solution in section V. Finally simulation results and conclusions are presented in sections VI and VII.

II. The Handover Process

A. Introduction

As stated earlier, MSs users are able to move within a wide area while carrying out the communication process. In the case a user moves from one BSS (Basic Service Set) to another BSS a Layer 2 Handover Process will be initiated. During this process, some management frames and context information are exchanged between APs and MS. For that reason, there will be a period of time, technically called *Handover Latency*, during which the station might not be able to send and receive traffic. The standard behavior of a Handover process defined in the 802.11 specification is divided into three differentiated stages: Discovery, Reauthentication and Reassociation.

The standard defines Active and Passive Discovery Processes in [1]. In a Passive Scanning approach, an MS waits for beacons periodically broadcast from nearby APs,

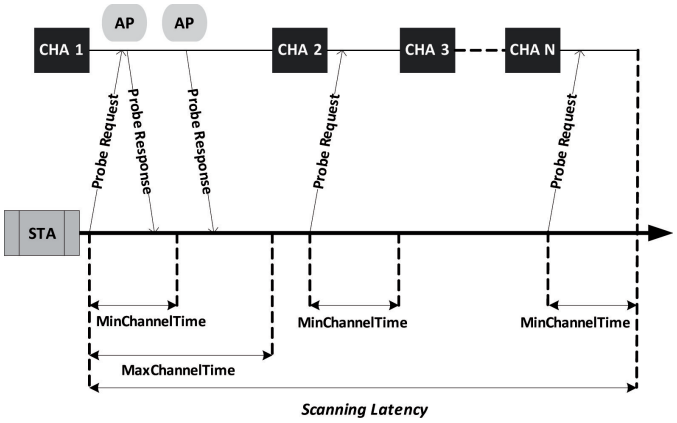


Fig. 1. Standard Scanning Process

so the MS can associate with the AP with the strongest beacon signal. In general, the beacon period is set to 100ms, so high latencies are reached while listening to all the channels. In an Active Scanning approach, an MS tunes to a channel and sends a *Probe Request* after detecting channel activity before *Probe Delay* timer expires. If no *Probe Response* is received from a candidate AP after waiting for *MinChannelTime*, the next channel is probed. Else, the station waits for *MaxChannelTime* to expire and processes all the *Probe Response* received.

As shown, both variables $MinChannelTime$ and $MaxChannelTime$ play a relevant role in the scanning process. It has to be contemplated that the above presented description belongs to a *Full Scanning* behavior, in which all allowed channels are taken into consideration. The standard scanning approach is outlined in fig. 1

B. The Handover Latency

Throughout the scanning process we deal with a relatively long period of time of the whole handover duration. We will refer to the former as the Scanning Time S, and the latter as the Handover Latency L, which considers the authentication and association delays. Thus, the Scanning Time is greater than delays produced by authentication and reassociation processes.

In a *Full Scanning* approach, we consider equation 1, where ct is the number of channels where traffic is found and ce is the number of empty channels. Thus, T_{ct} and T_{ce} represent the time needed to scan operating and empty channels respectively. In this case it is being considered that an MS directly waits for $MinChannelTime$, removing the *Probe Delay* timer, considered as a passive component of the active scanning.

$$S = ct.T_{ct} + ce.T_{ce} \quad (1)$$

Going deeper into this consideration, we can assume that through a Discovery Process, T_{ct} involves a period of time equal to $MaxChannelTime$, while T_{ce} comprises only an interval of time equal to $MinChannelTime$.

Thus, we can analyze the influence of times described in this section on the Standard 802.11 Handover approach. We can mention three different amounts of time that should be considered as wasted or non-productive time. Wherever an MS discovers activity but no *Probe Response* is received, an amount of time equal to $MinChannelTime$ is wasted; besides, an MS scanning an empty channel will consume an amount of time equal to *Probe Delay* or $MinChannelTime$. One of the most important amounts of wasted time is related to the interval between the last *Probe Response* received in a channel and the $MaxChannelTime$ expiration.

III. Previous Studies

Several previous related works, such as [2] and [3], demonstrate that the contribution of the first handover stage, where the scanning process is performed, is around 90% of the total Handover Latency, so we can affirm that optimizing the behavior of the scanning phase variables could produce important improvements in the handover effects related to packets delay and loss.

In order to reduce the scanning time, previous studies introduced some new concepts in order to propose fast handoff mechanisms. Most authors have focused on the most obvious possible tactic to optimize the Scanning Process, that is to define an appropriate order for channel scanning, known as a *Selective Scanning* technique.

In [4] a Fast Handoff algorithm is proposed to reduce the handover latency. This solution is based on the *Selective Scanning* mechanism using a binary mask. The channels must be scanned following the information contained on the mask, which is turned on for the channels from which *Probe Responses* were received previously. Additionally, the mask is turned on for non-overlapping channels 1, 6 and 11. If there is no success finding an AP to associate using the mask, it is inverted and the previous turned off channels are scanned. In addition to the *Selective Scanning* Mechanism, the Caching technique is presented in [5]. This mechanism uses IAPP (Inter Access Point Protocol) messages in order to build and maintain a cache with MAC Addresses of the adjacent APs.

In [4] the authors also present the utilization of a Neighbor Graph in a *Selective Scanning* context. This Neighbor Graph contains information about channels to be scanned. Both Caching and Neighbor Graph Selective Scanning approaches are very attractive, but as the implementation of the data structures has to be maintained by the APs, deep and complex modifications must be implemented. In addition, some possible problems could occur when the possible APs

to handover are deployed by different operators.

The solution proposed in [6] uses a time synchronization algorithm, hence all the APs connected to the Distributed System schedule different and synchronized Beacon sending times, so stations can passively scan by switching channels exactly when a Beacon is about to arrive. This should reduce the handoff latency to the authentication and association delays. In this case, there is difficulty in managing the time, because the synchronization process requires an optimal clock precision. Authors propose the implementation of the Network Time Protocol (NTP) to solve clock synchronization.

We can see that all the handoff optimization schemes presented above require changes in both the MS and AP sides, thus the deployment of these mechanisms in already existing networks is far from an easy task.

In [7] and [8] the main goal is to reduce the Handover Latency only to the reauthentication and reassociation delay, distributing the scanning phase during data communication period between the MS and the current AP. The authors of [7] focus on breaking the duration of the Discovery Phase in numerous sub-phases, so it can be executed in a smooth way. This scheme is based on an algorithm that dynamically changes the value of the threshold triggering the Discovery Phase. The RSSI (Received Signal Strength Indication) of current and candidate AP to handoff is used to adapt the threshold. The scanning is initiated earlier, and the threshold starts going down until an AP with relative good signal is found.

The scheme presented in [8] defines a Periodic Scanning Mechanism, in which MS periodically performs a Discovery Phase (each time on a different channel, during MinChannelTime).

Using these techniques, Data Frame communications may be performed during sub-phase intermediate times. In the case presented in [7], despite effectively reducing handoff latency, some strong constraints are introduced. The first one refers to a limited Network Deployment that considers enough overlapping area between two neighboring cells, for instance, an Indoor Wireless LAN environment. The second one is related to Mobility, considering that the MS node just moves in a modest speed. We can see that stations moving at higher speeds will not be able to adapt the threshold variable, while the signal strength will fall down very fast. Therefore, Periodic Scanning gives good results (in general less than a 50ms latency) but generates extra traffic consuming the MS power.

IV. Definition of Scanning Variables

As we have presented in section II and III, the standard handover process and further studies consider fixed values of $MinChannelTime$ and $MaxChannelTime$. Furthermore, the standard does not explicitly indicate certain values for those variables. As mentioned in [9] Wireless Network Interfaces (NICs) manufacturers implement proprietary scanning algorithms with different values for $MinChannelTime$, $MaxChannelTime$ and the channel on which the first *Probe Request* is sent.

In our approach we will focus on the adaptation of $MinChannelTime$ and $MaxChannelTime$ that have the major influence on the scanning latency. For that reason, as a first step, we must set the domain for these variables. On the other hand we will present a mechanism to decide which channel sequence to be implemented while scanning.

A. $MinChannelTime$

In the case of $MinChannelTime$, we define it as the minimum time to spend on scanning each channel, while at the same time it is the maximum time for an AP to respond to a *Probe Request*. If we thoroughly analyze the times, we can assume equation 2. As an ideal worst case scenario, the station should wait for $DIFS$ (Distributed Inter-Frame Space) and then choose $acWmin$ slots to backoff. We can valuate the transmission delay for sending a *Probe Response*, T_{TX} as $104.27\mu s$, based on simulation measurements.

$$\begin{aligned} MinChannelTime &= DIFS + backoff_{min} + T_{TX} \\ &= DIFS + (acW_{min}.aSlotT) + T_{TX} \\ &= 50\mu s + (31.20\mu s) + 104.27\mu s \\ &= 774.27\mu s \end{aligned} \quad (2)$$

Because we need to establish a maximum threshold for $MinChannelTime$, we performed some scanning simulations in order to evaluate the sensibility of this variable and establish the value. We found that the upper threshold fixed in $2048\mu s$ is an adequate value. Thus, the upper and lower thresholds are shown in equation 3,

$$774.27\mu s \leq MinChannelTime \leq 2048\mu s \quad (3)$$

B. MaxChannelTime

In the case of $MaxChannelTime$, we can define it as the maximum time to spend on each channel while scanning, allowing each AP to compete for the medium and send a *Probe Response*. There are some difficulties in determining the threshold values for $MaxChannelTime$ mainly because of the number of AP to be detected on the same channel and traffic conditions. The first constraint is basically one related to the Deployment Policy. As the aim of this mechanism is to develop a general purpose behavior, we will consider, as the worst case, a deployment scenario as shown in fig. 2, where all neighboring APs are configured in the same channel in an hexagonal coverage design. In this consideration, the maximum of APs to be discovered on the same channel is six. Based on the determination of $MaxChannelTime$ presented in [10], where it is considered a value of $10.240\mu s$ for a reasonable number of ten wireless interfaces operating on the same channel, then, as our approach considers only six APs have to be detected, the maximum threshold for $MaxChannelTime$ will be set

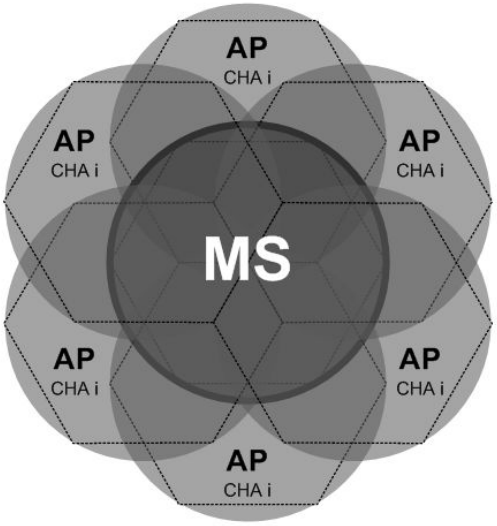


Fig. 2. Hexagonal Deployment on the same channel

up in $6144\mu s$. The sensibility of this value was also studied towards accomplishing simulations processes.

As an MS must wait for $MaxChannelTime$ after receiving a first *Probe Response*, we will let the lower threshold as the minimum $MinChannelTime$ possible value, so it can be adapted to the minimum as necessary, according to equation 4.

$$774.27\mu s \leq MaxChannelTime \leq 6144\mu s \quad (4)$$

C. Channel Sequence

Not only the above-mentioned variables impact on the scanning time. While executing a Full Scanning procedure, the chosen channel sequence could have an effect on the latency as well. In the context of 802.11 networks, three non-overlapping channels are offered. Considering North American specifications, channels 1, 6 and 11 do not overlap. So we can assume that by prioritizing those channels, the probability of discovering candidate AP faster may be increased. Experiments proposed in [9] show that in more than 75% of the cases, Wireless Network Interface Cards send *Probe Request* to channels 1, 7 and 9 with higher significance. So there is not any correlation between the non-overlapping channels and its priority in the scan sequence in the common scanning algorithms implemented by manufacturers. For our purposes, we suggest randomizing the sequence in two phases: first between the non-overlapping channels and subsequently among the others.

V. Adaptive Discovery Mechanism

The newly proposed mechanism is based on the Adaptation of Values for the variables that take part in the scanning process.

In order to introduce concepts related to Adaptive Systems we can make an analogy with a real-life situation, like an Urban Traffic Control System implementation. In an urban environment, where vehicles' mobility is a truly difficult-to-manage matter, an Adaptive Traffic System appears as a promising solution. Under this approach a centralized system simply adapts red light times for main streets and avenues in a city considering unexpected traffic conditions. Several video or traditional loop detectors are strategically positioned in order to capture the real number of vehicles flowing in a specific area. The centralized system takes this useful information and introduces it as an input for an Adaptive Algorithm that immediately generates an output containing the new red light time for a particular road.

High values of traffic flow generally produce lower red light times, depending on the traffic condition of neighboring streets. As a result, an Adaptive Traffic Control radically reduces delays and stops in the vehicle flow. As a limitation, one of the main problems within an Adaptive Traffic Control scheme is related to the data collection of decision-making variables because of the difficulty of maintaining street installations.

Now that we have introduced the concept of adaptive algorithms, our main purpose is to arrive at a reduction in the handover latency, while maximizing the success rate of performed handoff processes. Referring to the Traffic Control analogy, our scanning main objectives can be related to reducing delays and stops in our traffic environment. We can consider the Scanning Process as a particular street to be crossed, and $MinChannelTime$ and $MaxChannelTime$ as the main variables that control the traffic lights. The information we need to adapt these variables is contained in the *Probe Responses* received from APs while performing the scanning procedure, making the analogy with the real traffic flow sent by the controllers located in the intersections. Then the adaptive behavior should consider the deployment scenario, as the traffic system considers the neighboring streets' situation.

In contrast to the Adaptive comportment, several works have discussed how to reduce concrete values of $MinChannelTime$ and $MaxChannelTime$, making different assumptions with the purpose of obtaining low handover latencies. Considering our adaptive traffic system correlation, it is easy to see that trying to fix optimal static red light times for avenues and streets not only is an exceptionally difficult task but it seems almost impossible that a single static time can deal with traffic necessities and unexpected conditions.

Thus, these kinds of solutions fall in a strong dependence with the deploying policies adopted in each case. Different waiting times are required for different scenarios. For instance, in the case where different APs operate on the same channel, collisions will be frequent, and an MS can not detect activity after waiting for $MinChannelTime$. This increases the chance of failure of the whole handover process. To avoid this condition, a dynamic behavior is needed.

A. The Adaptive Discovery Algorithm

1) Decision Making Variables: As we have shown, $MinChannelTime$ and $MaxChannelTime$ will be adapted in their values during the scanning process. For that purpose, we need to define the parameters that will act as the decision makers to tune them. In this case we must use information available while receiving a *Probe Response*, other than the number of Discovered APs in each channel, we will use the $RSSI$ (Received Signal Strength Indication) of each detected AP.

2) Implementation: Our purpose is to obtain the best candidate AP from all the discovered ones. While scanning a channel, we can discover one or more of them with different values of $RSSI$. We desire a context where a single AP with relative good signal strength is found rather than multiple APs on the same channel. For that purpose, we will fuse both decision making variables in a single function, as indicated in equation 5.

$$R = \frac{RSSI}{APsChannel_i} \tag{5}$$

This simple relationship allows us to balance the decision of selecting the best AP in a fair manner. For each scanned channel, the local maximum value for R between all the discovered AP is calculated, obtaining R_L. Also, a global maximum, R_G, is maintained between all the scanned channels. R_L is used to reduce the values of $MinChannelTime$ and $MaxChannelTime$. Instead, R_G is used to reference to the best AP to handover. Therefore, the value of R_L is compared with a fixed range.

Thus, if R_L is between one of the defined ranges, $MinChannelTime$ and $MaxChannelTime$ will be reduced to a fraction of their present values (F_R), always considering the threshold defined in Section IV. Otherwise, if no AP is found in the channel $MinChannelTime$ and $MaxChannelTime$ are increased to a new average value between the previous and the last calculated, (F_I). Typical reduction factor values (F_R) are presented in figure 4. The algorithm behavior is presented in fig. 3.

Different deployments generate different scanning conditions. In some cases, as a heterogeneous channel disposition, the MS should wait the same time for all the channels using the standard algorithm. The Adaptive Mechanism allows to distinguish between finest and poorest scanning conditions, so it can manage the risk of discovering less APs in the following channels by changing the values of $MinChannelTime$ and $MaxChannelTime$.

B. Sending Probe Responses

In addition to the adaptive mechanism and in order to optimize the present approach, other modifications should be included. For instance, modifying the way that the APs send a *Probe Response* to an MS could help to reduce the scanning time. In this case, the standard determines that a *Probe Response* answering a *Probe Request* should be sent after waiting for DIFS, which has a typical value of $50\mu s$. Considering that we are dealing with the risk of not finding an AP in a certain channel before $MinChannelTime$ expires, we should decrease the first delay to a value equal to SIFS (Short Interframe Space), five times lower than DIFS, so the MS can receive information as soon as possible. If more than one AP is operating on that channel and several *Probe Responses* are sent, then there is a high possibility of detecting a collision in the link. If SIFS is used, a possible collision would be managed sooner, making it more probable that an MS could receive the information to handoff on time. Moreover if SIFS is used, the MS can keep the control on the channel earlier, avoiding that another MS starts a transmission.

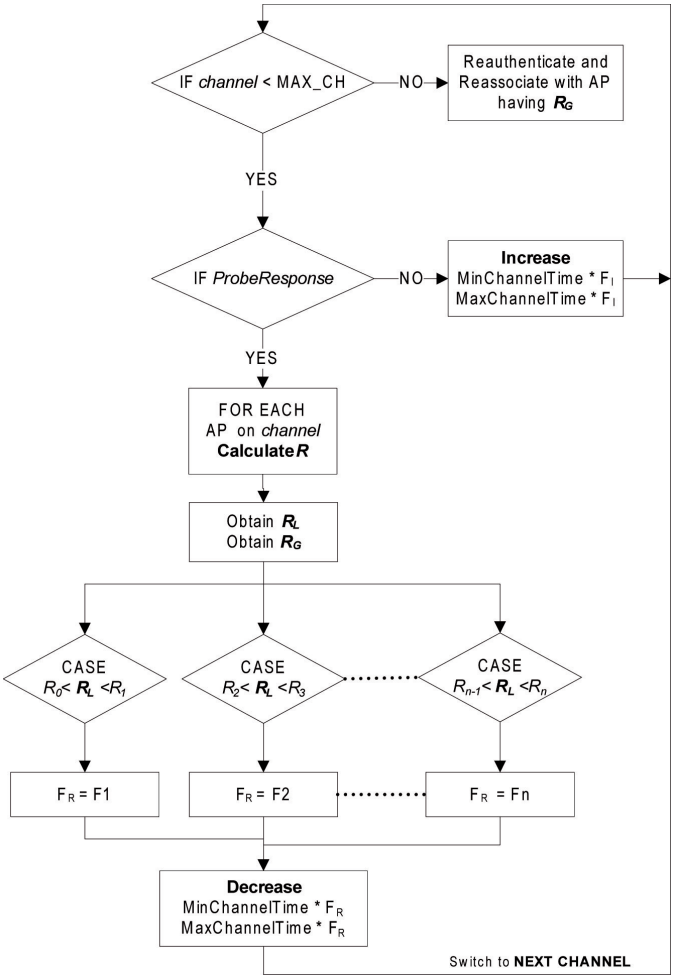


Fig. 3. Adaptive Discovery Algorithm

R1		R2		R3		R4	
FROM	TO	FROM	TO	FROM	TO	FROM	TO
0%	20%	20%	40%	40%	60%	60%	100%
FR	**0,6**	FR	**0,5**	FR	**0,4**	FR	**0,3**

Fig. 4. Typical values of F_R for each defined range of R_L

VI. Simulation Process

In order to appreciate the behavior of our proposed algorithm, a Wireless Network Simulator, *SimulX2* [11], [12], was utilized to design and run simulation scenarios. The complete handover mechanism was implemented, in order that total handover latency could be dimensioned. The main measurements to be obtained are related to the Handover Latency reduction, the number of Failure Handoff Cases, the rate of Discovered APs over total APs in the range and the Relative Signal Quality of the selected AP.

In this particular implementation for the scope of the simulation, the $RSSI$ variable was emulated using the position of the discovered APs, the distance between the MS and AP, and both AP and MS coverage range. Thus, the decision-making variable R has a mathematical form as one referred in equation 6

$$R = 1 - \frac{distance_{AP-MS}}{min(Range_{AP}, Range_{MS})} \quad (6)$$

As it can be seen, the minimum between the ranges is used in order to prevent a situation where one of the wireless network interfaces has a small coverage, so it is very close to the necessary range to establish the connection. Hence, if the MS continues moving it will be less probable that another handover is performed soon.

Initially, the following general scenario is set up. A MN moves from its actual AP in the direction where a group of different candidate APs can be detected. These APs were positioned in an equidistant manner and channel allocation and APs ranges were randomized so a wide range of activity could be experimented. An important number of simulation processes were run using this approach, so an average value of the handover latency reduction was obtained. Our objective was to quantify and evaluate a completely random scenario, as the above presented, so as to take average values for $MinChannelTime$ and $MaxChannelTime$.

After running this first simulation, it could be appreciated that the magnitude of the reduction arrives up to the 46% between an adaptive behavior and the standard algorithm. It has to be considered that both algorithm were set with the same values for $MinChannelTime$ and $MaxChannelTime$, in this case, the maximum threshold defined in section IV. Furthermore, no failures were detected while performing the handoff using both algorithms. Also, in more than 54% of the cases, all AP in range were discovered using the adaptive approach, so in less than half of the total simulations not all APs in range were successfuly discovered. Despite latter situation, it must to be underlined that both the Standard and the Adaptive behavior always successfuly reassociate with a new AP having the same $RSSI$.

Next we focus on the evolution of $MinChannelTime$ and $MaxChannelTime$ during the prior general and completely random scanning process. Figures 5 and 6 show both profiles where the axis represents the number of scanned channels considering the channel sequence described in section IV. It can be appreciated that values of $MinChannelTime$ and $MaxChannelTime$ start going down after some APs are discovered, and then the values are stabilized with some minor increases when no APs are discovered.

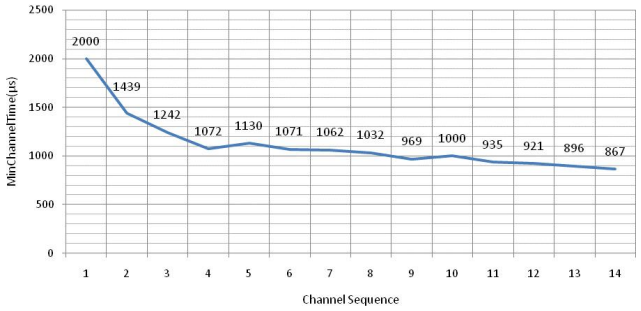


Fig. 5. Evolution of MinChannelTime

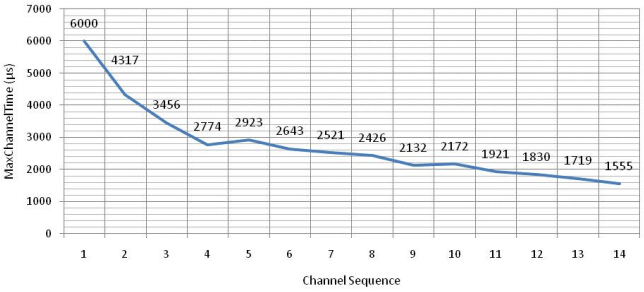


Fig. 6. Evolution of MaxChannelTime

Thus, an average value for $MinChannelTime$ and $MaxChannelTime$ was obtained and used as a parameter for next simulations considering some different scenarios. First the most common non-overlapping deployment (channels 1, 6 and 11) was configured and two different simulations were run. The first used the standard discovery algorithm with the obtained $MinChannelTime$ and $MaxChannelTime$ average values. The second one was processed performing the adaptive approach with the maximum thresholds. Using the standard algorithm the failure rate goes up to 35%, whereas applying the adaptive behavior we achieved a complete success. This allows us to assert that a modification in the deployment could produce unexpected results while trying to force static optimal values for the time control variables. For that reason, the discovery rate of different APs over total APs in range is greater using the adaptive algorithm.

Finally, we configured a new scenario using some APs overlapping on the same channel, as shown in fig.2. After running several simulations, a complete failure of the handoff was achieved while using the standard algorithm with the average values for $MinChannelTime$ and $MaxChannelTime$. However, no failures were encountered using the Adaptive

Discovery Algorithm, which successfully found APs in range on that channel with a modest discovery rate.

VII. Conclusions and Perspectives

A new dynamic behavior for the adaptation of time control variables of the handover process allows us to satisfy both proposed objectives, which are Latency Reduction and Minimum Failure. Better values than traditional handover ones are reached, and independence from network deployment conditions is satisfied, so the MS performs the scanning process successfully for any particular network scenario. A successful handover appears more important than a time reduced handover. A temporal interruption is always preferred than a permanent disconnection while looking with the eyes of an application user

It was also demonstrated that fixed optimal values for $MinChannelTime$ and $MaxChannelTime$ can only optimize predefined and limited scenarios, limiting the application of the handover process and introducing strong constraints that generally result in more than a few modifications not only in the MS side, but also in the network side. The Adaptive Discovery Algorithm only requires modifications on MSs, therefore implementation of the scanning process may not produce extreme difficulties.

The application of the Adaptive behavior in a wireless environment could be merged with other fast handover mechanisms. For instance, an approach similar to that presented in [7] may distribute the scanning process while the MS is still connected to the previous AP, accomplishing the handover process in a smoother way.

The adaptation concept is a general purpose solution for several problems in real life. It could be applied not only in 802.11 networks, but it must be considered as a General Discovery Process behavior in the field of networking as well.

The complexity of the adaptive algorithm could vary depending on the necessities. In this particular case, decision making variables are adapted in a fixed rate. More complex applications and further studies in Discovery Processes should focus on concepts referring to Learning Algorithms and Artificial Neural Networks.

References

[1] IEEE Std. 802.11b, "Wireless LAN Medium Access Control (MAC) and Physical (PHY) specifications: High Speed Physical Layer Extension in the 2.4 GHz Band," IEEE, Tech. Rep., 1999.

[2] F. A. Gonzalez, J. A. Perez, and V. H. Zarate, "HAMS: Layer 2 Accurate Measurement Strategy in WLANs 802.11," IEEE, Tech. Rep., 2005.

[3] P. Huang, Y. Tseng, and K.-C. Tsai, "A Fast Handoff Mechanism for IEEE 802.11 and IAPP Networks," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, 2006.

[4] S. Shin, A. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 Wireless LANs," in *International Conference on Mobile Computing and Networking*, 2004.

[5] S. Park, H. Kim, C. Park, J. Kim, and S. Ko, *Personal Wireless Communications*. Springer Berlin-Heidelberg, 2004, ch. Selective Channel Scanning for Fast Handoff in Wireless LAN Using Neighbor Graph, pp. 194–203.

[6] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005.

[7] Y. Liao and L. Cao, "Practical Schemes for Smooth MAC Layer Handoff in 802.11 Wireless Networks," in *International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2006.

[8] J. Montavont, N. Montavont, and T. Noel, "Enhanced schemes for L2 handover in IEEE 802.11 networks and their evaluations," in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2005.

[9] V. Gupta, R. Beyah, and C. Corbett, "A Characterization of Wireless NIC Active Scanning Algorithms," in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, 2007.

[10] H. Velayos and G. Karlsson, "Techniques to reduce the IEEE 802.11b handoff time," in *IEEE International Conference on Communications*, 2004.

[11] S. Hachana, "Design and Implementation of a Wireless Communication Network Simulator Environment," 2006, Master Thesis Report, Université de la Manouba - École Nationale des Sciences de l'Informatique.

[12] N. Montavont, J. Montavont, and S. Hachana, "Wireless IPv6 Simulator: SimulX," in *10th Communications and Networking Simulation Symposium (CNS07), located with Spring Simulation Multiconference 2007 (SpringSim'07)*, 2007.

Appendix B

IEEE INFOCOM 2009 Conference on Computer Communications

Adaptive System for 802.11 Scanning

German Castignani
Institut TELECOM ; TELECOM Bretagne ; RSM
Université européenne de Bretagne, France
Universidad de Buenos Aires, Argentina
german@castignani.com.ar

Nicolas Montavont
Institut TELECOM ; TELECOM Bretagne ; RSM
Université européenne de Bretagne, France
nicolas@montavont.net

I. Introduction and Motivation

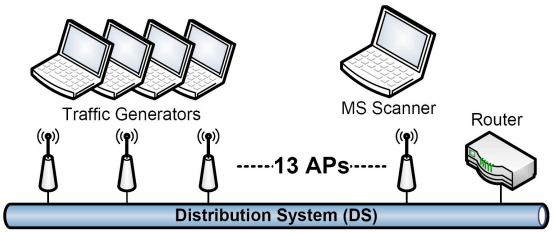
Nowadays, 802.11 networks appear as the most widely implemented wireless access in the market, since a vast number of devices embed WiFi interfaces and access networks are being continuously deployed. In this context, a mobile user may deal with a high variety of scenarios. These scenarios consist on heterogeneous access point (AP) deployments characterized by overlapping frequencies, traffic load and high interference. Moreover, these conditions cannot be anticipated by the mobile user while moving between APs and thus an appropriate scanning algorithm is needed. The change of AP, also known as *handover*, involves scanning for APs, authentication and association. During the handover process, a mobile station (MS) cannot send or receive data, so the handover latency must be as low as possible. Related work particularly focuses on the reduction of the scanning latency, representing 90% of the whole handover process [1]. The 802.11 standard [2] establishes two different scanning algorithms to discover APs. In *Passive Scanning* the MS simply waits for *beacons* from APs and in *Active Scanning*, the MS proactively probes channels for activity by broadcasting *Probe Requests* and waiting for *Probe Responses*. Precisely in active scanning, the MS first sends a Probe Request on the channel and waits for *MinChannelTime*. If a Probe Response is received before *MinChannelTime* expiration, the MS then waits for *MaxChannelTime* so as to discover all possible APs on that channel. If no Probe Response is received before *MinChannelTime* elapses, the MS directly switches to another channel. After scanning all channels, the MS *(re)authenticates* and *(re)associates* with a candidate AP. These timers play a major role in the handover latency. Fixed optimized values for these timers will minimize the handover latency, but may generate failure in discovering APs in some scenarios. For that purpose, we propose an adaptive scanning algorithm that dynamically adjusts and sets both timers.

II. Discussion

MinChannelTime allows discovering the first AP, while *MaxChannelTime* makes it possible to obtain the real number of APs on each channel. As the 802.11 standard [2] does not establish concrete values for these timers, different values can be observed on every single 802.11 card. If a low value is set for *MinChannelTime*, it may be possible that the first AP is not discovered, declaring the channel empty. For that reason, we can state that *MinChannelTime* conditions the scanning successfulness. In the case of *MaxChannelTime*, a low value could not be enough to capture responses from all APs on a particular channel due to collisions and further exponential backoff on each AP. This situation does not allow obtaining the real activity level on each channel. On the other hand, high values for both timers produce high latencies. There is a trade-off between the scanning latency and its successfulness, and both timers influence its balance. Velayos et al. [3] suggest to establish low optimal values for both timers ($1ms$ and $10ms$), to minimize the latency. While low latencies are obtained using fixed timers, APs in channels dealing with high level of traffic or interference may not be able to respond because of high frame transmission delays. Other methods try to optimize the scanning process by selecting only a subset of channels to scan [4] [5] [6]. Liao et al. [7] propose to distribute the scanning phase into interleaved subphases during data communication. These solutions introduce several constraints and can only optimize some particular network deployments, falling in undesirable scanning failure for some other scenarios. A new behavior has to be defined in order to satisfy mobility requirements in a wide spectrum of configurations and deployments.

III. Adaptive Scanning Algorithm

An adaptive system appears as an alternative for the standard active scanning, that is unable to adjust to different environment states. We propose an adaptive variation of both *MinChannelTime* and *MaxChannelTime*, so these timers take different values during a scanning instance, depending on the scenario. Relying on the information received on previous channels, the adaptive algorithm decides to reduce or increase timers. An MS starts scanning using a maximum value for both timers so as to adapt them in subsequent channels between preestablished maximum and minimum thresholds. These thresholds have been defined by the following experimentation. High values for *MinChannelTime* and *MaxChannelTime* ($50ms$ and $200ms$ respectively) have been used over multiple scenarios in order to obtain a high number of responses from different APs. Then, the delay of the first and following responses on each channel has been computed. The results of this experimentation allow us to set thresholds at [6-34ms] for *MinChannelTime* and [8-48ms] for *MaxChannelTime*.
The adaptive algorithm behaves as follows. On each channel



		Latency (ms)	Failure (%)
SC1	STD	227 / 403 / 1025	52 / 20 / 13
	ADP	210	0
SC2	STD	159 / 363 / 814	29 / 3 / 0
	ADP	434	2

Fig. 1. Testbed configuration and main results

with activity, the reduction or increase rate (R) is calculated based on equation 1. The quotient between the RSSI (Received Signal Strength Indication) and the number of APs on the scanned channel is considered, since a low number of APs on the same channel indicates lower interference.

$$R_i = \frac{max(RSSI_i)}{APchannel_i} \quad (1)$$

Depending on the R obtained on channel i, timers are reduced or increased for channel $i+1$. For that reason, the proposed adaptive behavior uses a random channel switching policy. First, it randomly switches between the non-overlapping channels (1, 6 and 11) and then, among all others. We have considered this behavior since channels 1, 6 and 11 are more likely to be used in common AP deployments [5], and so timers could be reduced faster. Finally, after scanning all channels, the MS selects the AP with the greatest RSSI in the channel having the greatest R for the association.

IV. Evaluation and Results

A *testbed* was implemented using up to thirteen APs and four MSs from different manufacturers as illustrated in fig. 1. An Atheros card using a modified *MadWiFi* driver was implemented as a scanner. A large set of configurations has been considered: four AP deployments using eight different conditions of channel allocation, level of interference and traffic load have been evaluated. In all proposed scenarios, up to a hundred scanning instances have been performed. For each configuration both standard scanning using three different values for *MinChannelTime* and *MaxChannelTime* (10-20, 25-50 and 50-200ms) and the proposed adaptive scanning have been evaluated and compared. For space reasons, only major results are being presented in fig. 1. Results for the standard scanning correspond to the three evaluated sets of timers, from the lowest to the highest. In all cases, we observed an important variation of probe responses' delay depending on the activity on each channel. This also produces different scanning latencies and failure rates. In the particular case where only three APs on channels 1, 6 and 11 are deployed (SC1), when traffic is injected, the three evaluated standard algorithms reach high levels of failure (from 13% up to 52%). Contrariwise, the adaptive algorithm gives the lowest latency (210 ms, against 227ms for the best standard scanning scenario) without any scanning failure. Another result was obtained while deploying eleven APs on the same channel (SC2). This is an uncommon extreme scenario that clearly shows the compromise between latency and failure; if *MinChannelTime* and *MaxChannelTime* are reduced, the failure rate immediately increases. Using adaptive scanning, a constant average latency is obtained (434 ms) while reducing the scanning failure (2%). Standard scanning using 10-20ms for timers gives a lower latency (159 ms), but a high failure rate (29%) is also observed. The main benefit of the proposed algorithm is that a single algorithm adjusting timers between fixed thresholds satisfies all scenarios.

V. On-Going and Future Work

An adaptive scanning technique produces effective handovers, independently of any particular deployment. We have shown that the effect of traffic and interference during the scanning process is far from being irrelevant. Different medium conditions require different time to wait for responses from APs, so as to avoid the case where a channel with activity is erroneously declared empty. In all scenarios we experienced, the adaptive algorithm compared with the standard approach offers a better percentage of discovered AP, minimizes the failure rate, and keeps a low scanning latency.

Currently we are working on determining an optimal adaptation rate for both timers in order obtain a single optimized algorithm that fits all scenarios. For that reason, a sensibility analysis of the algorithm's parameters is being performed. In this case the adaptive algorithm is being evaluated allowing the MS to start scanning using not only the maximum thresholds, but lower values as well. We are also investigating an AP selection algorithm. Concepts about potential bandwidth estimation on each channel, like those presented in [8] could help to tackle this issue. How the MS switches channels or when to prematurely stop the scanning instance if an acceptable AP was found will also be studied. As different applications running on a wireless environment tolerate different interruption time, a set of scanning strategies may be embedded in our algorithm based on cross-layer information.

References

[1] Francisco A. Gonzalez, Jesus A. Perez and Victor H. Zarate, *HAMS: Layer 2 Accurate Measurement Strategy in WLANs 802.11*, IEEE Technical Report, .
[2] IEEE Std. 802.11-2007,*IEEE Standard - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE-SA Standards Board, 2007.
[3] H. Velayos and G. Karlsson, *Techniques to reduce the IEEE 802.11b handoff time*, IEEE International Conference on Communications, 2004.
[4] H. Kim, S. Park, C. Park, J. Kim and S. Ko, *Selective Channel Scanning for Fast Handoff in Wireless LAN using Neighbor Graphs*, ITC-CSCC2004, 2004.
[5] S. Shin, A. Singh and H. Schulzrinne, *Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs*, International Conference on Mobile Computing and Networking, Proceedings of the second international workshop on Mobility management and wireless access protocols, 2004.
[6] J. Montavont and T. Noel, *IEEE 802.11 Handovers Assisted by GPS Information*, IEEE WiMob, 2006.
[7] Y. Liao and L. Gao, *Practical Schemes for Smooth MAC Layer Handoff in 802.11 Wireless Networks*, Proceedings of WoWMoM'06, IEEE, 2006.
[8] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose and D. Towsley, *Facilitating Access Point Selection in IEEE 802.11 Wireless Networks*, ACM Internet Measurement Conference, New Orleans, LA, October, 2005

German Castignani
german@castignani.com.ar
Institut TELECOM
TELECOM Bretagne
Universidad de Buenos Aires

Nicolas Montavont
nicolas@montavont.net
Institut TELECOM
TELECOM Bretagne

Andrés Arcia
ae.arcia@telecom-bretagne.eu
Institut TELECOM
TELECOM Bretagne
Universidad de Los Andes

Adaptive Systems for 802.11 Scanning

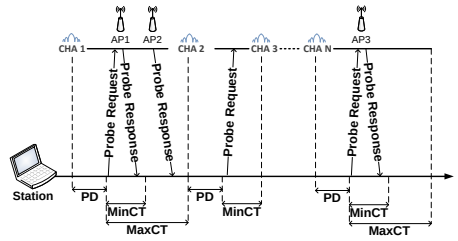
1 802.11 Handover

A mobile user may deal with a high variety of **heterogeneous access point deployments** characterized by overlapping frequencies, traffic load and high interference.

Handover involves *scanning*, *authentication* and *association*. During these processes, a mobile station cannot send or receive data, so **the latency must be minimized**.

Scanning latency represents **90%** of the total handover latency

2 Active Scanning using Fixed Timers



The standard proposes an Active Scanning using ***MinChannelTime*** and ***MaxChannelTime*** timers

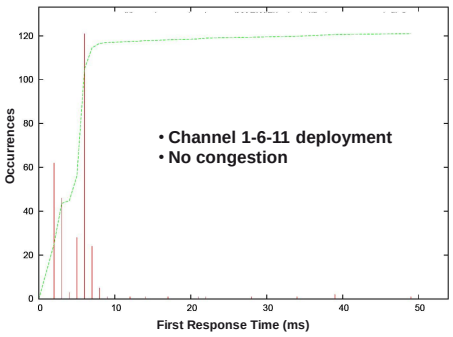
The standard does not provide concrete values for these timers. Velayos et al. [1] suggest setting:

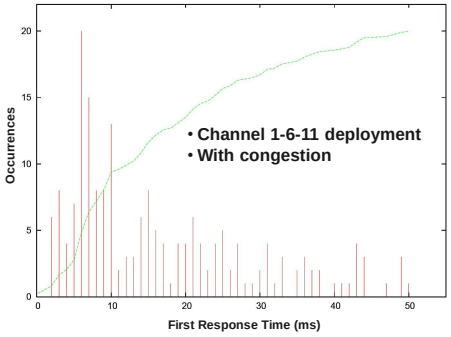
MinChannelTime = 1 ms
MaxChannelTime = 10 ms

What would we expect using these values?

[1] H. Velayos and G. Karlsson, Techniques to reduce the IEEE 802.11b handoff time, IEEE International Conference on Communications, 2004.

3 Probe Response Delay – Latency vs Failure Trade-off



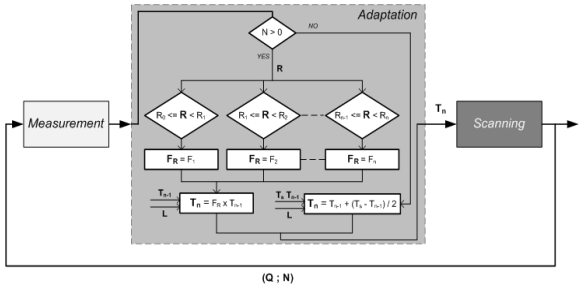


The effect of congestion delays the Probe Responses

Scanning Failure: If the Probe Response delay is greater than *MinChannelTime*, a channel is erroneously declared empty.

Scanning Latency: High values for MinChannelTime and MaxChannelTime provide low failure, but result in a long handover latency

4 Adaptive Scanning



Dynamic adjustment of timers during the scanning to **avoid failure** and keep a **reduced latency**

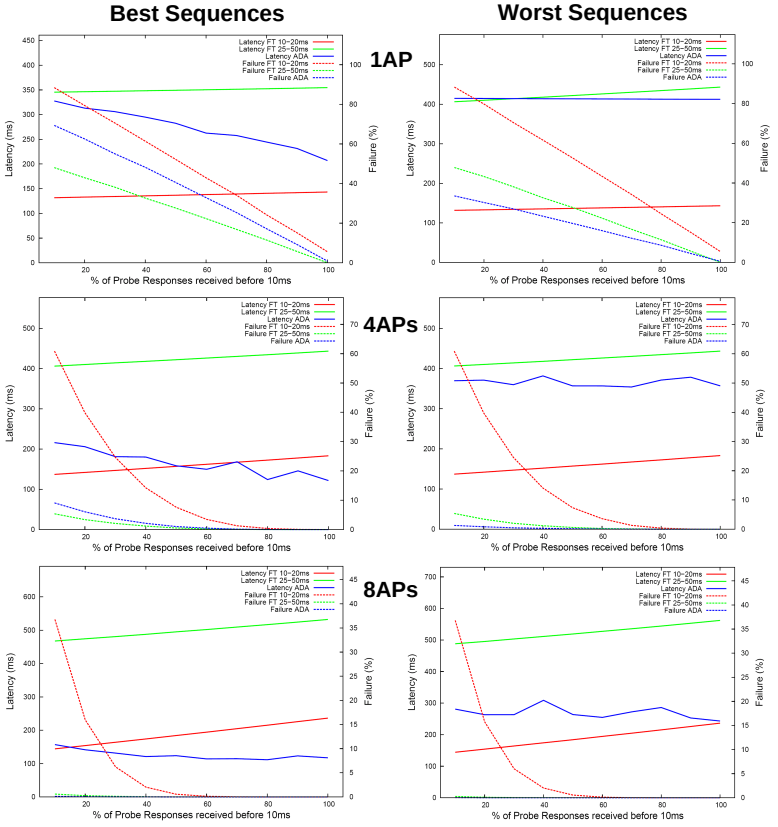
Probe Responses are processed on each channel to decide **which timers to apply** for the next channel

Parameters to decide how to reduce: Signal quality (Q), Number of APs per channel (N), Congestion status

Limits within timers (T) are adapted are defined by experimentation

Two random subsequences to **prioritize non-overlapping channels**

5 Observation by Simulation



- **Fixed Timers** scanning keeps **high failure rates** for **high congestion**
- **Adaptive scanning** reduces the **latency** and the **failure rate** while the number of operating channels augments, independently of the congestion status

6 Experimentation

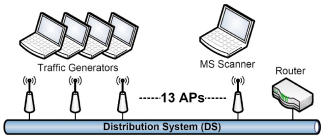
madwifi.org Implementation of both **Fixed Timers** and **Adaptive Scanning**

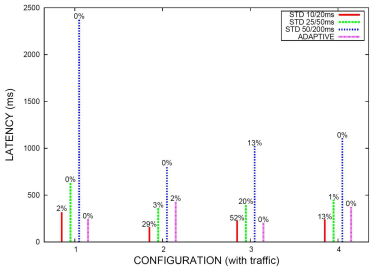
Configuration 1: 13 APs one by one on channels 1 to 13 (one AP per channel)

Configuration 2: 13 APs all on channel 11

Configuration 3: 3 APs one by one on channels 1,6 and 11 (one AP per channel)

Configuration 4: 12 APs by four on channels 1,6 and 11 (four AP per channel)





Fixed Timers scanning reaches high failure rates compared with **Adaptive** scanning in all configurations

Adaptive scanning maintains low latencies

7 Conclusions and On-going Work

- We have observed the **delay of probe responses** in a congested environment, conditioning 802.11 fixed timers scanning
- We have proposed an **Adaptive Scanning** based on a dynamic adjustment of scanning timers
- We have evaluated by **simulation** and **experimentation** while comparing both approaches
- We have shown the influence of the **channel sequences**

- We should evaluate different **adaptation strategies**, trying to find an optimal way to compute the **decision making parameter** and the **adaptation rates**
- We should investigate which other parameters apart from RSSI and number of APs can we consider to compute the adaptation
- We should consider the decision of prematurely stop scanning if APs have been discovered
- Focus on the AP selection policy

Bibliography

[1] IEEE Std. 802.11-2007,*IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks -Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE-SA Standards Board, 2007.

[2] Matthew S. Gast, *802.11 Wireless Networks: The Definitive Guide*, O'Reilly, 1st edition, ISBN: 0596001835.

[3] Francisco A. Gonzalez, Jesus A. Perez and Victor H. Zarate, *HAMS: Layer 2 Accurate Measurement Strategy in WLANs 802.11*, IEEE Technical Report, .

[4] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, *Neighbor Discovery for IP version 6 (IPv6)*, RFC 4861 Standards Track, IEEE Network Working Group, September 2007.

[5] S. Thomson, T. Narten and T. Jinmei, *IPv6 Stateless Address Autoconfiguration*, RFC 4862 Standards Track, IEEE Network Working Group, September 2007.

[6] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460 Standards Track, IEEE Network Working Group, December 1998.

[7] C. Perkins, *IP Mobility Support*, RFC 2002 Standards Track, IEEE Network Working Group, October 1996.

[8] C. Perkins, *IP Mobility Support for IPv4*, RFC 3344 Standards Track, IEEE Network Working Group, August 2002.

[9] D. Johnson, C. Perkins and J. Arkko *Mobility Support in IPv6*, RFC 3775 Standards Track, IEEE Network Working Group, June 2004.

[10] H. Soliman *Mobile IPv6: Mobility in a Wireless Internet*, Addisson-Wesley, ISBN:0201788977, 2004.

[11] S. Yankov and S. Wiethoelter *Handover Blackout Duration of Layer 3 Mobility Management Schemes*, POL4G Project Report, TU Berlin, Germany, May 2006.

[12] Y. Liao and L. Gao, *Practical Schemes for Smooth MAC Layer Handoff in 802.11 Wireless Networks*, Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), IEEE, 2006.

[13] A. Mishra, M. Shin and W. Arbaugh, *An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process*, ACM SIGCOMM Computer Communication Review (Volume 33 - Issue 2), Technical Paper, 2003.

[14] S. Shin, A. Singh and H. Schulzrinne, *Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs*, International Conference on Mobile Computing and Networking, Proceedings of the second international workshop on Mobility management and wireless access protocols, 2004.

[15] H. Kim, S. Park, C. Park, J. Kim and S. Ko, *Selective Channel Scanning for Fast Handoff in Wireless LAN using Neighbor Graphs*, International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2004), 2004.

[16] I. Ramani and S. Savage, *SyncScan: practical fast handoff for 802.11 infrastructure networks*, Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2005.

[17] N. Montavont, J. Montavont and T. Noel, *Enhanced schemes for L2 handover in IEEE 802.11 networks and their evaluations*, IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2005.

[18] J. Montavont and T. Noel, *IEEE 802.11 Handovers Assisted by GPS Information*, IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2006.

[19] H. Velayos and G. Karlsson, *Techniques to reduce the IEEE 802.11b handoff time*, IEEE International IEEE International Conference on Communications, 2004.

[20] P. Huang, Y. Tseng and K. Tsai , *A Fast Handoff Mechanism for IEEE 802.11 and IAPP Networks*, IEEE 63rd Vehicular Technology Conference (VTC), 2006.

[21] WildPackets Professional Services , *Converting Signal Strength Percentage to dBm Values*, Technical Report 20021217-M-WP007, 2002

[22] Cisco Systems, Inc., *Channel Deployment Issues for 2.4-GHz 802.11 WLANs*, Technical Report OL-6270-01, 2004

[23] Cirond Technologies, Inc., *Channel Overlap Calculations for 802.11b Networks*, White Paper, 2002

[24] V. Gupta, R. Beyah and C. Corbett, *A Characterization of Wireless NIC Active Scanning Algorithms*, IEEE Wireless Communications and Networking Conference (WCNC), 2007

[25] G. Castignani and N. Montavont, *Adaptive Discovery Mechanism for Wireless Environments*, EUNICE 14th Open European Summer School, 2008

[26] G. Castignani and N. Montavont, *Adaptive Systems for 802.11 Scanning*, IEEE INFOCOM Conference on Computer Communications, Student Workshop, 2009

[27] NS-3 Project Group, *NS-3 Software Architecture*, Technical Report, 2007

[28] J. García Sánchez, *Introduction to Simulation on OMNET++*, ARCOS Group University Carlos III of Madrid, 2007

[29] Universitat Politècnica de Catalunya, *OPNET Modeler Manual*, Departament d'Enginyeria Telemàtica Secció de l'EPSEVG, 2004

[30] N. Montavont, J. Montavont and S. Hachana, *Wireless IPv6 Simulator: SimulX*, 10th Communications and Networking Simulation Symposium (CNS), located with Spring Simulation Multiconference (SpringSim), 2007

[31] J. Kantorovitch, Z. Shelby and T. Saarinen, *Wireless Adaptation Techniques for Heterogeneous Multihop Networking*, IST Information Society Technologies, IST-2001-37385 6HOP, 2003

[32] Alessio Botta, Alberto Dainotti and Antonio Pescape, *Multi-protocol and multi-platform traffic generation and measurement*, INFOCOM 2007 DEMO Session, May 2007, Anchorage (Alaska, USA).

[33] Thomas Williams, Colin Kelley et al.*gnuplot*, http://www.gnuplot.info