IEEE 802.11 scanning algorithms: cross-layer experiments

German Castignani, Nicolas Montavont, Andres Emilio Arcia Moret, Mohamed Rabie Oularbi, Sébastien Houcke

Collection des rapports de recherche
de Télécom Bretagne

RR-2011-05-RSM

IEEE 802.11 scanning algorithms: cross-layer experiments

German Castignani (Télécom Bretagne)
Nicolas Montavont (Télécom Bretagne)
Andrés Arcia-Moret (Université de Los Andes, Vénézuela)
Mohamed Oularbi (Télécom Bretagne, Lab-Sticc)
Sébastien Houcke (Télécom Bretagne, Lab-Sticc)

IEEE 802.11 Scanning Algorithms: Cross-Layer Experiments

German Castignani[1], Nicolas Montavont[1], Andrés Arcia-Moret[2], Mohamed Oularbi[1], and Sébastien Houcke[1]

[1]Institut TELECOM / TELECOM Bretagne, Université Européenne de Bretagne, Brest, France, email:{firstname.lastname}@telecom-bretagne.eu
[2]Universidad de Los Andes, Computer Sciences Department, Mérida, Venezuela, email: amoret@ula.ve

September 20, 2011

Abstract

In wireless communication, the discovery of the surrounding access points is crucial to provide seamless connectivity to mobile users. In IEEE 802.11, this discovery process is performed by passive and active scanning functions. For both, timers are usually constant (within the 802.11 driver) and their configuration affects the two main scanning performance metrics, i.e., scanning latency and scanning failure. In order to manage this trade-off, we propose some adaptive scanning algorithms that are aware of the access point's signal power and congestion level using cross-layer information. As shown by experimentation in a real testbed, this allows a faster topology discovery while mostly reducing the failure rate.

Contents

List of Figures

List of Tables

1 Introduction

Since its first standardization, IEEE 802.11[1] has seen a significant growth, going through a boom in recent years. With more than a decade of evolution, 802.11 has come to be the leading technology for wireless local area networks. This is primarily because it grants users the possibility to connect to the Internet with an acceptable performance at a very low cost. In a urban environment, thousands of APs can be found in small areas, giving a high-density deployment. Moreover, due to the exploitation of 802.11 in any kind of computing device (e.g., notebooks, tablets and smartphones), users have become increasingly mobile, while running on-line applications. Since users aim to be always best connected [2] in mobility scenarios, they must change their point of attachment to the network while moving, causing a *handover*.

The handover is the transitions between two APs belonging to the same access router (Layer 2 handover, referring to the TCP/IP stack) or to different ones (Layer 3 handover). A layer 2 handover consists of three phases. First, a mobile station (MS) may start a *scanning* process to look for APs. After finding a candidate AP, the MS performs *reauthentication* and *reassociation* with the selected AP. On the other hand, a Layer 3 handover requires in addition to redirect flows to the new location of the MS (i.e., the new AP). This can be managed by protocols like MobileIPv4 [3] and MobileIPv6 [4]. In the rest of this document we will only focus on Layer 2 handovers.

Focusing on the scanning phase, it has been proven that in an open-system deployment (i.e., no encryption or authentication), the scanning phase is the most time consuming phase within an 802.11 handover [5]. As data frames cannot be exchanged during the scanning phase, applications are affected, then its duration must be minimized. The standard Active Scanning procedure [1] specifies that an MS must broadcast a Probe Request management frame on each channel and wait for Probe Responses during *MinChannelTime*. If the MS does not receive a Probe Response management frame (containing the information of an AP) within *MinChannelTime*, it must switch to the next channel and restart the procedure. If at least one Probe Response frame was received before the expiration of *MinChannelTime*, the MS must wait for a longer time, called *MaxChannelTime*. This should allow the MS to capture all the responses from APs on the same channel. One of the main limitations of this procedure is related to the values of both *MinChannelTime* and *MaxChannelTime*, which are not specified in the standard. In our previous works [6] [7] [8] [9] [10], we have studied timers values dynamics and their impact on the scanning performance. It can be appreciated that low values for the timers help to reduce the scanning latency (i.e., the duration of the scanning procedure), but they could increase the scanning failure (i.e., the impossibility to discover any AP after scanning all channels) in some scenarios.

In this report, we propose a cross-layer adaptive scanning approach to find the most suitable timer to probe each channel. We use physical-layer information such as the channel load and the power measured on each channel to improve the scanning performance. Both physical-layer parameters provide the MAC layer with a preliminary knowledge of wireless deployment, allowing the MS to set a timer that fits every particular channel condition. We implement different adaptive approaches in the *ath5k* 802.11 open-source driver and evaluated its performance in different scenarios. We focus on the study of the most salient scanning metrics: the scanning latency, the scanning failure, the percentage of discovered APs and the time to discover the first AP (since a common algorithm is to stop the scanning as soon as an AP is found). Then, we identify and study the trade-off between these performance metrics.

The remainder of this report is organized as follows. In Section 2, we introduce some related work on layer 2 handover optimizations. In Section 3, we present a set of preliminary experiments aiming to show the relation between the channel load and power and the probe

response delay. Section 4 details the physical-layer estimation of the channel load. Then, in Section 5, we discuss the cross-layer adaptive scanning approach. In Section 6, we present the experimental testbed, its implementation details and a comparison of different cross-layer adaptive approaches against standard fixed-timers scanning algorithm. Finally, in Section 7, we conclude the report and give the perspectives for the future work.

2 Related Work

There has been a fair amount of research done in layer 2 handover optimization, particularly in 802.11 AP discovery. Most part of the proposed optimizations focus on reducing the duration of the discovery phase (i.e., the scanning latency). One of the first contributions on analyzing 802.11 scanning was proposed by Velayos et al. [11]. They analyze the relationship between scanning timers and latency, focusing on the probe response delay for different traffic load and number of stations. By means of simulations they provide a concrete value for *MinChannelTime* ($1ms$) and *MaxChannelTime* ($10ms$). However, empirical studies show that those values might not be enough for an MS to receive Probe Responses from an AP. In [10], we show that a minimum waiting time (i.e., MinChannelTime) of $10ms$ is needed to receive the first probe response the 97% of the time. More focused on wireless cards comparison, authors in [5] and [12] show how some wireless manufacturers implement different scanning strategies and timers, resulting in different scanning performance. Specifically in [5], authors show that Probe Responses can be delayed up to $40ms$, depending on the deployment scenario.

Regarding concrete scanning strategies, Xu et al. [13] propose periodic scanning, like [14], [15] and [16], which is based on dividing the complete scanning phase in different sub-phases, which are triggered before the handover. So the scanning is performed before the disconnection to the current AP. Particularly in [13], authors investigate the relationship between the first response delay of an AP and its signal level (Received Signal Strength Indicator, RSSI), showing that APs having better RSSI statistically respond faster. Under this hypothesis, they propose D-Scan, which uses low timers to guarantee the reception of probe responses coming from good APs. This implies that the quality of an AP considers only its signal level and not its traffic load, which also conditions the performance of the AP. Then, the correlation between the RSSI and the AP response time is not always guaranteed, since a low-RSSI AP may reply faster than a high-RSSI AP with a high traffic load. Due to traffic load, the time to process the Probe Request and generate the Probe Response in the AP side could become non-negligible.

Yoo et al. [17] take into account the user's QoS requirements (delay and loss ratio) during the handover. They propose to adapt the time an MS will take to scan a channel and the time interval between two scanning sub-phases, according to a desired loss rate and delay.

Like in any other periodic scanning technique, dividing the scanning phase and triggering sub-phases before the handover cannot ensure that the discovered APs may be available when the handover decision is taken. This limitation becomes even more important if the MS speed increases, and the environment changes faster than in low mobility scenarios. In any case, these optimization techniques always require to fix a value for the timer.

3 Preliminary Experiments

Because Probe Requests are not acknowledged, an MS has to wait for the timer on every channel, even on those where no APs have been deployed. Since our goal is to explore the most suitable timer for each channel, we focus on studying the effects of data traffic flows on the First probe Response Delay (FRD). We first isolate an AP and use an MS as a scanner. This

MS is responsible of sending Probe Requests and process Probe Responses. Then, in order to find out the relationship between the FRD and the load on the cell, we compute the FRD for seven different pre-defined channel loads. Table 1 presents the main results of this preliminary experiment, considering different amounts of traffic, from background (i.e., only management frames circulate on the channel) up to 12.5 Mbps using 802.11g APs and iPerf traffic injector. We can appreciate that both the empirical mean ($\bar{E}[FRD]$) and standard deviation ($\bar{\sigma}(FRD)$) of the FRD tends to increase as the traffic load increases. One relevant observation is illustrated in Fig. 1, which shows the cumulative distribution function of FRD ($P[FRD < t]$). The effect of loading the cell produces a large dispersion of the FRD. In the case of background traffic, almost no FRD dispersion is observed. However, when high traffic load is injected (12.5 Mbps in Fig. 1), the FRD tends to follow a displaced exponential distribution. Based on this observation we propose in Section 5 an adaptive approach that considers the load on the cell to adjust the scanning timers.

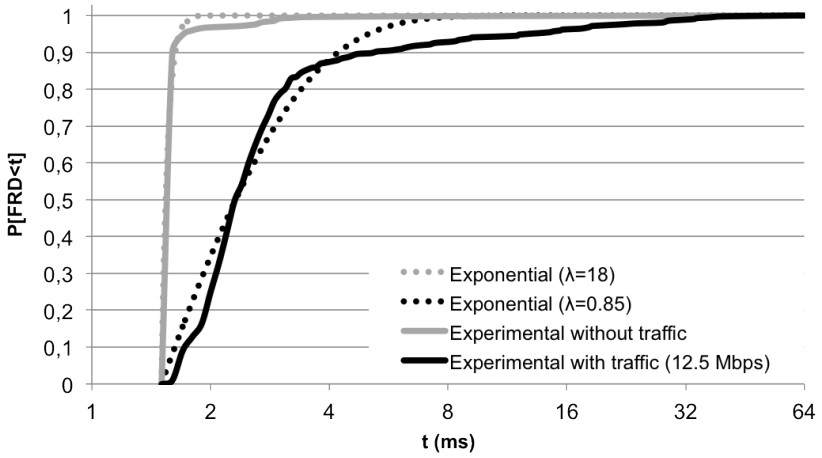

Figure 1: First Probe Response Delay (FRD) CDF

Table 1: Preliminary Observation

Flow (Mbps)	Load (%)	$\widehat{E}[FRD]$ (ms)	$\widehat{\sigma}(FRD)$ (ms)
Background	1.52	1.83	2.12
1	5.62	1.79	1.19
2	9.68	1.84	1.03
4	20.05	1.81	0.58
10	51.97	2.07	0.62
11.5	73.11	3.9	5.7
12.5	74.49	3.58	4.87

4 Physical-layer channel load estimation

4.1 Introduction

The cross-layer adaptive scanning algorithms proposed in this report uses physical-layer information. In our case, we focus on two physical parameters: the *channel load* and the measured *power* on each channel. Within the experimentation, we evaluate the performance of the scanning algorithms in two steps. First, after deploying the scenario (APs and MSs for traffic

generation) we passively measure the channel load and the power. Second, we perform scanning using our algorithms which consider both physical layer measurements. Regarding the power, we simply measure it from the captured signal o,n the same channel during a given time window. On the other hand, the channel load is estimated by calculating the ratio between the signal-plus-noise and the noise-only samples. The estimation mechanism is presented in Section 4.2. This physical-layer information could be estimated by a wireless network card. In the particular case of this experimentation, as it will be explained in Section 6, this physical-layer information is obtained using a dedicated device, a Universal Software Radio Peripheral [1] (USRP2), that allows sensing and processing 802.11-based physical signals.

4.2 Estimation Mechanism

When an AP is active, between two consecutive frames, we have different inter frame spacing (IFS) intervals which guarantee different type of priority. At the receiver side, the observed signal is a succession of frames of noise samples corresponding to the IFS intervals or idle periods and of signal plus noise samples corresponding to data frames.

For clarity reason, we first assume that we have only one data frame in the observation window. Let $\mathbf{y} = [y(1), \ldots, y(N_s)]$ be a set of N_s observations on a given WiFi Channel, such that

$$\begin{cases} y(m) = w(m) & 1 \leq m \leq m_1 - 1 \\ y_i(m) = \sum_{l=0}^{L-1} h(l)x(m - m_1 - l) + w(m) & m_1 \leq m \leq m_2 \\ y(m) = w(m) & m_2 + 1 \leq m \leq N_s \end{cases} \quad (1)$$

where the x for $j = 1, \ldots, M$ is the data transmitted signal, $h(l)$ is the channel response from source signal to the receiver's antenna, L is the order of the channel h. $w(m)$ is a complex additive white Gaussian noise with zero mean and variance σ_w^2. The variance σ_w^2 is assumed to be known or at least estimated. In practice, the noise power is captured by the USRP device. We observe a given channel when no traffic and no access point is active in that channel. In this case, no data signal is present and the only signal observed is due to thermal noise and background noise. Thus the noise power is equal to the variance of the observed samples.

The vector $\mathbf{y}$ can be divided into three parts : noise , signal plus noise and noise. Starting from the set of observation $\mathbf{y}$, we would like to find which samples correspond to noise and which ones correspond to signal plus noise. The used approach relies on the following : since the samples are supposed to be independent in the noise areas and correlated in the signal plus noise area due to the channel effect and their OFDM structure, we propose to use a likelihood function that provides an information about the independence of the processed sample.

Let $\mathbf{Y}(u)$ denotes the following set of observations :

$$\mathbf{Y}(u) = [y(u), \ldots, y(N_s)] \qquad 1 \leq u < N_s \quad (2)$$

And let us define f_Y the joint probability density function of $\mathbf{Y}(u)$. If $\mathbf{Y}(u)$ is composed of only noise samples

$$f_Y(\mathbf{Y}(u)) = \prod_{m=u}^{N_s} f_w(y(m)), \quad (3)$$

where f_w is the probability density function of a complex Normal law centered and variance σ_w^2, given by

$$f_w(x) = \frac{1}{\pi\sigma_w^2} e^{-|x|^2/\sigma_w^2}, \quad (4)$$

[1] http://www.ettus.com/

The log-likelihood that the vector $\mathbf{Y}(u)$ is formed of $(N_s - u)$ noise independent samples is expressed as

$$\begin{aligned} \mathcal{J}(u) &= \log\left[\prod_{m=u}^{N_s} f_w(y(m))\right] \\ &= -(N_s - u)\log(\pi\sigma_w^2) - \frac{1}{\sigma_w^2}\sum_{m=u}^{N_s}|y(m)|^2 \end{aligned} \tag{5}$$

As u varies in the interval $[1, m_1)$, the number of noise samples composing $\mathbf{Y}(u)$ decreases and so does $\mathcal{J}(u)$ until it reaches a minimum bound at m_1 (see Fig 2). However, for u varying from

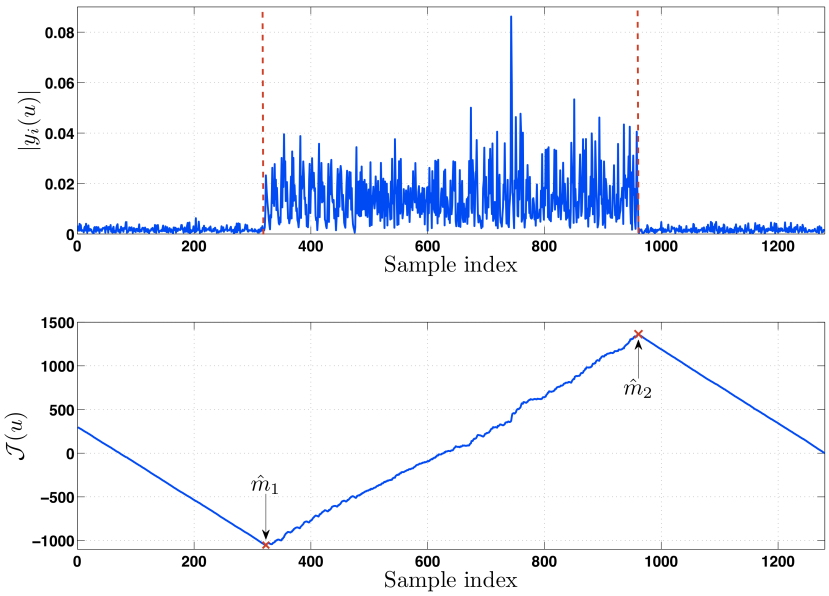

Figure 2: Example with one frame and corresponding criterion behaviour.

m_1 to m_2 the number of signal plus noise samples decreases, therefore the ratio noise samples over signal plus noise samples increases and by the way $\mathcal{J}(u)$ increases. It reaches its maximum value if and only if $\mathbf{Y}(u)$ contains only noise samples, i.e when $u = m_2$.

Finally for $m_2 < u < N_s$, $\mathcal{J}(u)$ decreases again for the same reason than the one explained for $1 < u < m_1$.

The assumption to have only one frame in the observation window is too restrictive. Based on the behavior of $\mathcal{J}(u)$, we can clearly see in Fig 2 that the slope of $\mathcal{J}(u)$ is positive when u corresponds to the index of a signal plus noise sample and negative when u corresponds to the index of a noise sample. Therefore, we can take advantage of the gradient of $\mathcal{J}(u)$ to distinguish the nature of the observed samples. Introducing the function $\Phi(u)$ such that

$$\Phi(u) = \frac{1}{2}\left[\text{sign}\{\nabla(\mathcal{J}(u))\} + 1\right] \tag{6}$$

Here we denote by ∇ the gradient of $\mathcal{J}(u)$ and sign{.} denotes the sign operator. According to this, $\Phi(u)$ equals 1 when signal plus noise samples are present and zero when it is only noise,

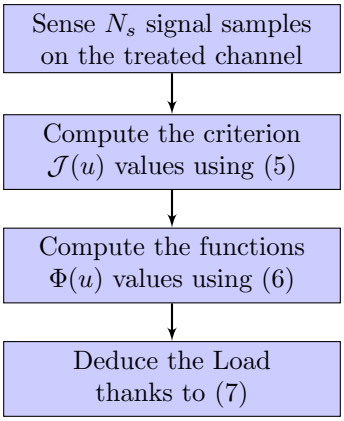

Figure 3: Proposed algorithm phases

and the channel occupancy rate is estimated by :

$$\widehat{C_{or}} = \frac{1}{N_s} \sum_{u=1}^{N_s} \Phi(u) \tag{7}$$

The whole algorithm is summerized in Figure 3.

5 Cross-Layer Adaptive Timers Setting

We propose to use the channel load and the power to set a proper timer on each channel while scanning. We illustrate in Fig. 4 the basic behavior of an Adaptive timers setting approach versus a fixed one. Within an adaptive strategy, timers are independently set for each channel (T1, T2,..., T13), depending on physical-layer information (channel load and power). Moreover, the channel sequence is no more a frequency ordered sequence (CH1, CH2..., CH13) but it follows a predefined order (CH#1, CH#2 ..., CH#13), being CH#n the n^{th} channel to scan (i.e., which means a concrete frequency). We first present the theoretical basis of such a mechanism. Then, we detail the algorithms design and implementation.

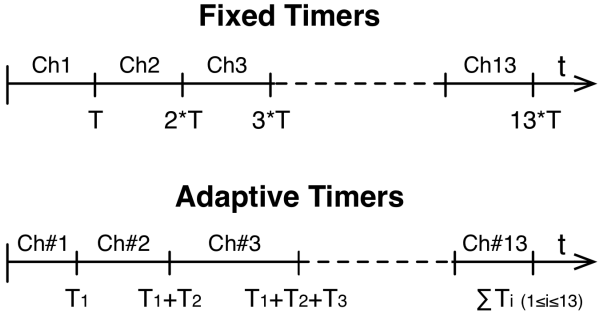

Figure 4: Fixed versus adaptive timers setting strategy

5.1 Theoretical Analysis

We aim to obtain an expression to generate the waiting time on each channel. In Fig. 1 the empirical FRD distribution is approximated using a random variable that follows a displaced

exponential distribution. We focus on a single-timer approach that differs from the standard two-timers approach, (i.e. *MinChannelTime* and *MaxChannelTime*) since we want to analyze the main effects of an adaptive behavior in terms of the failure and latency trade-off, without considering maximizing the number of discovered APs. Being T the FRD, then $T \sim a + exp(\lambda)$. Remark that for each traffic load, we get an exponential law with a different parameter λ. The value of a is defined as the minimal observed time for a response to arrive. We aim to find an expression that represents the amount of waiting time on each channel (t_{min}) that allows receiving a probe response with a given probability ($P[T \leq t_{min}] > p$). We can use the probability density function (pdf, in Eq. 8) of the displaced exponential variable T to calculate probabilities, then $P[T \leq t_{min}]$ can be expressed as shown in Eq. (9).

$$f(t, \lambda) = \begin{cases} \lambda e^{-\lambda(t-a)} & t \geq a \\ 0 & t < a \end{cases} \tag{8}$$

Focusing on the side $t \geq a$, then we aim to find $T = t_{min}$ that satisfies $P[T \geq t_{min}] > p$. Then we have Equation 9.

$$\begin{aligned} P[T \leq t_{min}] &= \int_a^{t_{min}} \lambda e^{-\lambda(t-a)} dt \\ &= \lambda \int_a^{t_{min}} e^{-\lambda(t-a)} dt \\ &= -\lambda \frac{e^{\lambda(a-t)}}{\lambda} \Bigg]_a^{t_{min}} \\ &= 1 - e^{\lambda(a-t_{min})} \end{aligned} \tag{9}$$

Then, t_{min} can be expressed in terms of λ and p.

$$\begin{aligned} P[T \geq t_{min}] &> p \\ 1 - e^{\lambda(a-t_{min})} &> p \\ e^{\lambda(a-t_{min})} &< 1 - p \\ \lambda(a - t_{min}) &< ln(1-p) \\ t_{min} &> a - \frac{ln(1-p)}{\lambda} \end{aligned} \tag{10}$$

Note that Eq. (10) is an expression for t_{min} that depends on the parameter of the distribution (λ, which varies with the traffic load), the minimum observed FRD and the probability p, which represents the confidence interval (a grade of precision for the calculated t_{min}). Then, giving that the variance of an exponential random variable is well known ($\sigma^2 = 1/\lambda^2$), we can estimate the parameter of the distribution (λ) by using the empirical standard deviation ($\bar{\sigma}$) obtained in the preliminary experimentation (Table 1). We may finally express in Eq. (11) an estimator for t_{min} considering $\widehat{\lambda} = 1/\bar{\sigma}$.

$$\widehat{t}_{min} > a - \bar{\sigma} ln(1-p) \tag{11}$$

This two-variables function gives values for t_{min} that are used on each channel depending on the standard deviation (σ) and the precision (p). Fig. 5 illustrates the behavior of this function, the x-axis represents the standard deviation, the y-axis represents the precision and the z-axis

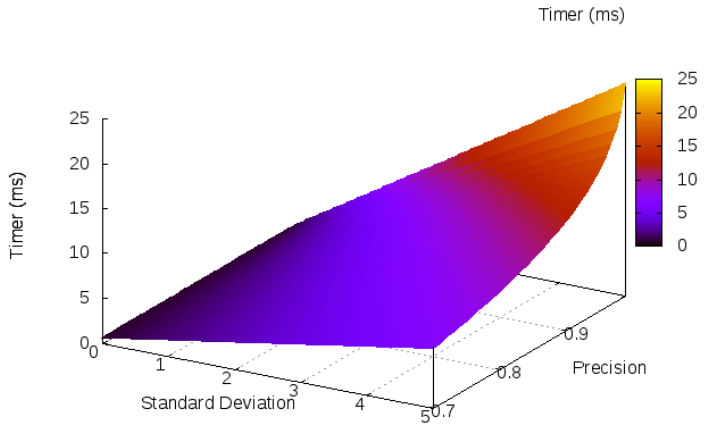

Figure 5: Timer values for different σ and p

gives the timer value (t_{min}). If the value of σ increases (i.e., the traffic load increases) the value of t_{min} linearly increases for a same value of p. Then, when increasing the confidence interval p, for a fixed value of σ, the value of t_{min} increases exponentially.

In summary, we have a simple mechanism that produces timer values using two variables. We may use the value of σ depending on the channel load. Then the value of p should be adjusted considering other parameters, as the channel power, the priority of the channel, etc.

5.2 Algorithm Design and Implementation

5.2.1 Parameters conditioning the scanning process

Before introducing our algorithms, we identify and analyze the parameters influencing the scanning process in a real implementation. From the MS side, we may consider the following:

- **Channel Sequence:** Before triggering the scanning process, the MS may establish an ordering of channels to scan. We could consider that if channels with activity (i.e., those who have at least one deployed AP) are scanned first, we have a higher probability of finding an AP soon, which may produce a reduced latency. Common implementation in open-source drivers like ath5k[2] and madWiFi[3] switch channels in an ordered sequence from channel 1 to channel 13.

- **Timer Value:** The amount of time that an MS waits for Probe Responses on a channel after sending a Probe Request. Ath5k and madWiFi open-source drivers use different fixed values, varying between $20ms$ and $200ms$. This results in a scanning latency that can be greater than one second.

Then, from the network deployment side, the parameters conditioning the scanning process are the following:

- **Number of APs:** In a scenario where very few APs are deployed, the probability of missing an AP is higher than in a high-density scenario.

[2]http://linuxwireless.org/
[3]http://madwifi-project.org/

- **Traffic Load:** The presence of traffic load in an AP may produce extra delays in the generation of the Probe Response, which may cause the expiration of the timer.
- **Quality of the channel link:** The current condition of the radio channel between the MS and the discovered APs.

5.2.2 Implementation

Based on Eq. (11), we implement the timer setting strategy in the ath5k open-source driver as follows:

```
Timer = FRD_min + Deviation * Precision
```

Where the `FRD_min` component is the absolute minimum observed FRD ($0.75ms$ from our measurements), `Deviation` is the empirical standard deviation of the FRD (Table 1) and `Precision` is the calculation of the term $-ln(1-p)$ for different values of p. Using different `Precision` values, we could increase or decrease the timer value for each channel. `Precision` values are indicated in Table 2 and have been set after several experimental trials, in order to find the best ones in terms of the scanning performance. In general, we decided to use higher values of p, i.e., higher `Precision` for the first channels to scan, since we have observed (as illustrated in Fig. 6), that the power level generally indicates the presence of APs in the channel. For that reason, we give a higher precision value (i.e., a longer timer) to channels with relative high power in order to avoid missing a response due to a scarce timer. Channels with relative low power will use shorter timers (due to a smaller precision value), since an AP is less likely to be found. Different algorithms have been envisaged, the difference between them is related to how they consider the measured channel power to calculate the channel sequence and the number of probe requests sent on each channel among others. During the experimentation, we first implemented the Conservative Algorithm. Then we made some modifications and optimizations in order to arrive to the Local Maximum Precision Algorithm (LMPA) which outperforms all other algorithms.

Conservative Algorithm A first approach was implemented using the timer calculation mechanism of Eq. 11 with a `FRD_min` equal to 1.69, since it has been the longest minimal First Response Time in the preliminary experiments. Then, the `Deviation` term was implemented considering an extrapolation of the empirical relation between the channel load and the standard deviation of the FRD (Table 1). In this case, we use channel load values as an input, separated in steps of 20%. The channel sequence was calculated by simply ordering the channels by power (decreasingly), i.e., different values of p (i.e., different `Precision` values) are used for different position in the channel sequence (the first channels have greater p). Only one Probe Request is sent on the channel under this approach.

Aggressive Algorithm This approach is similar to the conservative algorithm but using `FRD_min` equal to 0.75, since while performing the experimentation for the conservative algorithm, we have observed this lower floor. In this case, up to two Probe Requests are sent if the calculated timer for the channel is greater or equal to $10ms$.

Simple Precision Algorithm (SPA) Because of edge conditions of the if-clause, the granularity of 20% on channel load produced overshooting/undershooting of timers values. For that reason we decided to implement a 5% step if-clause, i.e., we estimate the value of the deviation term for values of load +/- 2.5%. This produces smoother transitions between deviation values,

that is used in Eq. 11. Also two Probe Requests are sent in this mechanism and the assignation of p values remains unchanged.

Local Maximum Precision Algorithm ($LMPA$) The Local Maximum Precision Algorithm, $LMPA$, takes into account the problem of channel overlapping. As shown in Fig. 6, even if only channels 1, 6 and 11 have an AP deployed, the power measured in their neighboring channels, is still high. This is due to the frequency spectrum usage in IEEE 802.11, which is based on 22MHz-wide channels, separated only by 5MHz, causing the well-known channel-overlapping problem. To avoid this situation, we propose to use a high precision value for the local maximums in terms of measured power. Note that the power levels of channels 1, 6 and 11 are quite different among them. This can be explained by the fact that different AP brands and models have been used, which may yield in different radio modules, antennas, etc. Also two Probe Requests are used in this algorithm.

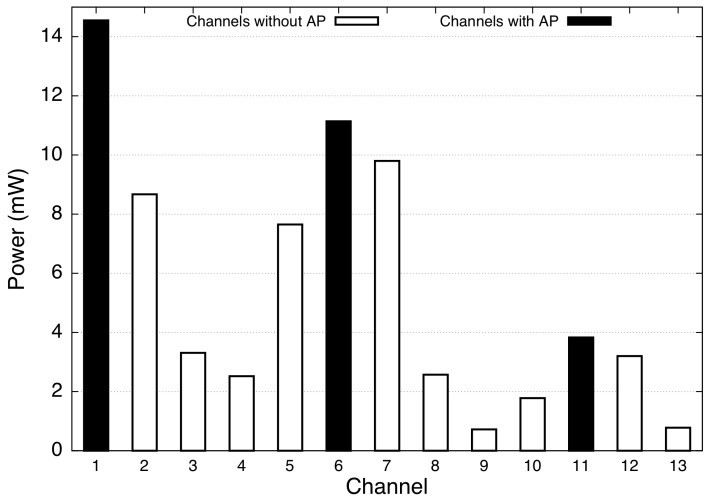

Figure 6: Measured Power on Scenario 2

Table 2: Precision Values

p	`Precision` $(-ln(1-p)$	No of Channels to scan
0.95	2.996	3 (or all local maximum)
0.85	1.897	3
0.80	1.609	3
0.75	1.386	4

Fig. 7 illustrates the main behavior of the adaptive algorithms. The MS first computes the channel sequence depending on the approach (Conservative, Aggressive, SPA or $LMPA$) by considering the power measured on the channels (Ch_Power_List). For each channel to scan, the MS calculates a timer ($T[i]$) based on the channel load ($Ch\_Load[i]$) and the precision (obtained after computing the channel sequence using Ch_Power_List). The scanning process finishes when the number of scanned channels reaches MAX_{CH} (whose value depends on local regulations).

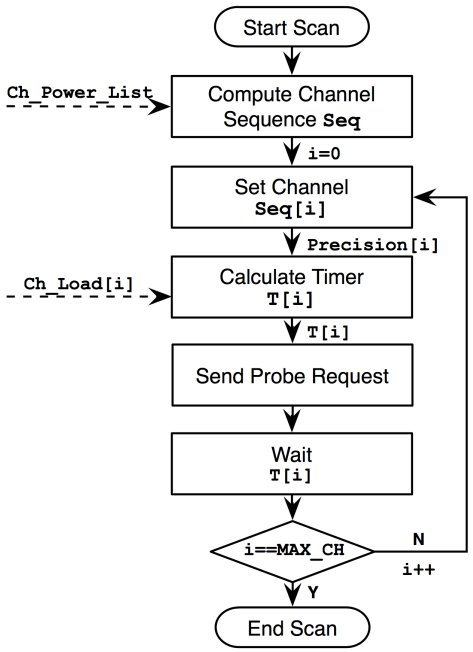

Figure 7: Adaptive Algorithm Approach

6 Performance Evaluation

6.1 Testbed

To evaluate the performance of the proposed adaptive approaches, we have deployed a testbed using real 802.11b/g devices. An MS implementing a Netgear WG511T card based on an Atheros chipset is used for scanning. We used a modified version of the ath5k driver that allows controlling all the parameters of the active scanning function. Up to five Linksys WRT54GL APs were deployed in different channels (see Section 6.3) in an isolated environment without interferences from any other wireless device. Traffic load was generated using jPerf[4]on a set of DELL Latitude laptops and ASUS netbooks. Since the Atheros card using ath5k driver does not allow physical-layer measurements in the MAC layer, we use a dedicated device, a Universal Software Radio Peripheral [5] (USRP2), that allows sensing and processing 802.11-based signals. Measurements from this device are statically set on the ath5k driver before scanning a particular scenario. Note that these measurements could also be performed before triggering the handover, in periodic sub-phases. However, with the recent standardization of the 802.11k [18] protocol for Radio Resource Measurement (not yet implemented in 802.11 drivers), physical-layer information can be directly requested by the MAC layer to an AP or an MS by simply using Channel Load Request/Report and Link Measurement Request/Report messages.

6.2 Metrics

The evaluation process focuses on some performance metrics characterizing a scanning algorithm, as listed bellow.

- **Latency:** the duration in milliseconds of the whole scanning process, from the scanning triggering to its completion (after scanning the last channel in the sequence).

[4]http://sourceforge.net/projects/jperf/
[5]http://www.ettus.com/

- **Discovery Rate:** the percentage of discovered APs in the scenario.
- **Failure Rate:** the percentage of scanning instances in which no AP was discovered after scanning the whole sequence of channels.
- **First Discovery Time:** the time in milliseconds needed by an MS to discover the first AP. This metric becomes important in the case an MS decide to stop scanning after finding the first AP in order to minimize disruptions on the upper layers.

These metrics define a trade-off, since using a fixed timer for scanning all channels cannot simultaneously keep a reduced latency, provide a high discovery rate, reduce the failure rate and spend a low time to discover the first AP.

6.3 Scenarios

We have considered five different scenarios, using different channel allocations and traffic load. Within these five scenarios we aim to define representative scenarios and deployments. The scenarios are as follows:

- **Scenario 1:** Three APs deployed in channels 1-6-11. This could be an example of an enterprise or campus non-overlapping deployment. Uplink and downlink traffic from 5 to 15 Mbps is injected in all channels.
- **Scenario 2:** Five APs deployed in channels 1-6-11. One AP in channel 1, and 2 APs in channels 6 and 11. This is another non-overlapping scenario. Uplink and downlink traffic from 5 to 15 Mbps is injected in all channels.
- **Scenario 3:** Five APs deployed in channels 3-4-8-9-13. This could be an example of an heterogeneous city-wide deployment. Uplink and downlink traffic from 1 to 15 Mbps is injected in all channels.
- **Scenario 4:** Two APs deployed in channels 1-11. This is another non-overlapping scenario (separation of 9 channels). Uplink and downlink traffic from 5 to 10 Mbps is injected in both channels.
- **Scenario 5:** One AP deployed in channel 6. This is a common non populated scenario. Downlink traffic of 20 Mbps is injected in the channel.

6.4 Results

6.4.1 General Results

We have evaluated seven scanning approaches, three fixed strategies using a single fixed timer (FX $2ms$, FX $5ms$ and FX $10ms$) and four adaptive approaches (Conservative, Aggressive, SPA and $LMPA$), in the five different scenarios described above. In the following figures, the complete set of experimental results are presented for each scenario. Metric values are highlighted in different colours, in order to give a comparison against the fixed strategies. The average and standard deviation (σ) for the metrics are presented.

We mainly focus on the analysis of the two last and adaptive approaches having the best performance (SPA and $LMPA$) against the standard approaches, using fixed timers (FX). For the fixed timers approaches, the channel sequence is always randomly calculated on each scanning trial. This is to avoid that a particular channel sequence penalizes or benefits some scenario. For the two adaptive approaches, we observe that timers vary from $2ms$ to $18ms$,

SCE1	Latency (ms)	σ	Discovery AP	σ	Failure (%)	First Discovery (ms)	σ	E[FRD]	σ(FRD)	min(FRD)	max(FRD)
FX 2ms	75,74	4,8	0,5	0,6	56,8	34,2	24,19	1,86	0,8	0,77	6,04
FX 5ms	109,52	3,3	1,59	0,8	8	45,33	29,39	2,17	0,8	0,76	8,8
FX 10ms	172,62	3,4	1,62	1	15,4	63,15	46,22	2,58	1,54	0,77	9,45
Conservative	**204,3**	4,1	**1,64**	0,8	**8,6**	**42,86**	59,9	2,82	2,4	0,85	13,63
Aggressive	**193,53**	5	**1,49**	0,7	**4,6**	**18,6**	30,37	4,5	1	0,77	17,54
SPA	**169,82**	2,9	**2,45**	0,6	**0,6**	**7,95**	4,63	2,05	1,55	1,55	15,73
LMPA	**146,14**	3	**2,14**	0,7	**1**	**12,65**	7,98	2,82	2,46	1,56	14,88

Figure 8: Complete Results for Scenario 1

SCE2	Latency (ms)	σ	Discovery AP	σ	Failure (%)	First Discovery (ms)	σ	E[FRD]	σ(FRD)	min(FRD)	max(FRD)
FX 2ms	76,29	5,3	0,8	0,8	42,8	34,1	23,8	1,83	0,74	0,76	6,79
FX 5ms	109,68	3,3	2,21	1,2	7,2	38,65	26,9	2,16	0,79	0,75	9,67
FX 10ms	172,85	3,4	2,8	1,2	3,2	57,23	43,8	2,59	1,68	0,76	10,83
Conservative	**199,89**	3,9	**2,65**	1,2	**5,2**	**20,44**	32,3	3,04	2,41	0,8	16,93
Aggressive	**188,63**	4,5	**2,8**	1	**1,2**	**10,55**	11,8	4,08	3,47	0,88	17,68
SPA	**154,2**	17	**1,78**	0,9	**4,6**	**12,98**	11,9	5,94	4,42	0,87	19,42
LMPA	**131,85**	4,6	**2,87**	1	**1**	**9,66**	7,56	3,75	3,75	0,76	20,16

Figure 9: Complete Results for Scenario 2

depending on the traffic load and the measured power. For the adaptive algorithms, the standard deviation for all the performance metrics is always comparable to fixed timers strategies. Focusing on the scanning latency, Fig. 13 shows that the adaptive approaches give a latency less than or equal to the FX $10ms$ strategy, being always between $85ms$ and $176ms$. Regarding the failure (see Fig. 14), the adaptive strategies always give reduced rates, between 0.2% and 16%. Only in scenario 5 $LMPA$ failure is slightly higher than the FX $10ms$ approach. Fig. 15 shows that the adaptive approaches keep a high discovery rate (up to 84%) even for scenarios where the latency is lower than the fixed timers strategies. Finally, focusing on the first discovery time, Fig. 16 shows that both SPA and $LMPA$ always discover the first AP sooner than all of the fixed timer approach. The first discovery time for the adaptive approaches varies between $6.35ms$ and $26.87ms$, but for fixed timer approaches it can reach up to $86.14ms$ in average.

6.4.2 Comparative Results

In order to provide a comparative view for the metric results, we define a simple score function (Eq. 12), where D is the discovery rate, L is the latency, F is the failure and FD defines the first discovery time. All metrics are equally considered.

$$S_i = 1 - \frac{D_i}{max(D_i)} + \frac{L_i}{max(L_i)} + \frac{F_i}{max(F_i)} + \frac{FD_i}{max(FD_i)} \tag{12}$$

7 Conclusions

For each approach (i) a score is assigned. The approach managing better the trade-off is the one that tends to minimize the score function (S_i). Fig. 17, illustrate the score functions on each scenario. This figure gives a global view of each approach and also illustrates how a particular scenario conditions the scanning performance. The adaptive approaches minimize the score in every scenario, since both SPA and $LMPA$ curves are closer to the origin. Moreover, $LMPA$

SCE3	Latency (ms)	σ	Discovery AP	σ	Failure (%)	First Discovery (ms)	σ	E[FRD]	σ(FRD)	min(FRD)	max(FRD)
FX 2ms	76,85	6,2	0,47	0,7	60,6	35,4	23,4	2,21	1,42	0,76	13,19
FX 5ms	110,23	4,3	1,47	1	19,2	43,85	28,5	2,52	1,05	0,77	9,54
FX 10ms	172,81	3,5	2,08	1,1	7,2	58,23	41,5	3,33	2,07	0,77	9,66
Conservative	**216,6**	3,6	**2,6**	1,2	**3,2**	**24,09**	30,9	3,88	3,01	0,77	19,55
Aggressive	**204,86**	4	**3,17**	1,1	**1,2**	**18,85**	19,2	5,92	4,39	0,78	19,19
SPA	**157,46**	4	**2,22**	1,1	**6,4**	**26,87**	17,4	4,73	3,5	0,77	17,27
LMPA	**176,78**	5,3	**1,78**	1	**9,6**	**16,32**	17,8	4,62	3,75	0,62	18,04

Figure 10: Complete Results for Scenario 3

SCE4	Latency (ms)	σ	Discovery AP	σ	Failure (%)	First Discovery (ms)	σ	E[FRD]	σ(FRD)	min(FRD)	max(FRD)
FX 2ms	75,19	3,8	0,33	0,5	69,8	32,34	22,7	1,99	0,686	0,8	5,49
FX 5ms	109,44	3,1	0,99	0,7	25,2	51,34	30,1	2,21	0,84	0,76	12,12
FX 10ms	172,44	3,1	1,15	0,7	19,6	75,9	49,3	2,38	1,3	0,77	9,67
Conservative	**146,8**	3,5	**1,53**	0,6	**3,6**	**10,53**	12,7	2,01	0,41	1,67	5,07
Aggressive	**134,81**	3,1	**1,52**	0,6	**3,4**	**9,27**	8,51	2,85	2,81	1,67	16,86
SPA	**171,81**	3,1	**1,33**	0,6	**5,6**	**13,82**	11,1	5,68	4,32	1,49	16,39
LMPA	**123,5**	17	**1,77**	0,4	**0,2**	**7**	6,01	2,25	1,41	1,69	17,33

Figure 11: Complete Results for Scenario 4

gives better scores than SPA in scenarios 2 and 4, since it allows setting longer timers only on channels where an AP is deployed. Regarding fixed timers approaches, FX $5ms$ behaves better than the rest of the fixed strategies. The adaptive approaches are capable to behave differently for each particular scenario, by taking into account its specific constraints in terms of interference and traffic load. They help to keep a reduced latency and failure rate while also importantly reducing the time to discover the first AP, and giving high discovery rates. We underline that only one timer (MinChannelTime) is used in the five evaluated approaches. Although this is not the case of the standard specification, we focus on the most critical timer and we expect that a second timer (MaxChannelTime) may only improve the discovery rate in a fixed or adaptive approach, by equally increasing latency in both cases.

In this work, we have studied a cross-layer mechanism to improve the IEEE 802.11 scanning process in the context of Layer 2 handover. We have observed the impact of cross-layer information on different performance metrics. So, we have shown that scanning timers can be successfully adapted using samples of signal power and congestion levels.

We have also defined a set of metrics that illustrate the trade-off to be managed while scanning. Since an optimal scanning timer value (i.e., the one to wait for probe responses) depends on each particular AP deployment, we proposed and evaluated two different adaptive algorithms which use physical-layer information. These adaptive approaches ensure that each timer matches the characteristics of each channel (i.e., signal level and traffic load).

Our future work will focus on integrating the physical-layer measurements on the wireless card. For this, we will deploy the adaptive algorithms on different available 802.11 drivers. Moreover, we pretend to use the recent standardized 802.11k protocol, that explicitly enables a mobile station to request for physical-layer characteristics.

Another hint on cross-layer optimization for IEEE 802.11 scanning could be to use transport layer information (like TCP) in order to adapt the scanning timers for different application needs.

SCE5	Latency (ms)	σ	Discovery AP	σ	Failure (%)	First Discovery (ms)	σ	E[FRD]	σ(FRD)	min(FRD)	max(FRD)
FX 2ms	75,47	4,2	0,31	0,5	68,8	32,55	24	1,83	0,72	0,78	4,19
FX 5ms	109,44	3,4	0,89	0,3	11	57,41	31,4	2,01	0,44	0,67	3,94
FX 10ms	172,26	3,1	0,88	0,3	12,4	86,14	50	2,03	0,41	0,83	3,46
Conservative	**106,9**	3,2	**0,9**	0,3	**10**	**6,43**	1,98	1,96	0,32	1,71	3,84
Aggressive	**96,57**	17	**0,99**	0,1	**0,6**	**7,57**	3,54	3,13	3,23	1,71	16,57
SPA	**85,71**	17	**0,87**	0,3	**12,6**	**6,35**	1,55	1,99	0,34	1,71	3,44
LMPA	**85,95**	17	**0,84**	0,4	**16,4**	**17,12**	2,09	1,91	0,25	1,11	2,68

Figure 12: Complete Results for Scenario 5

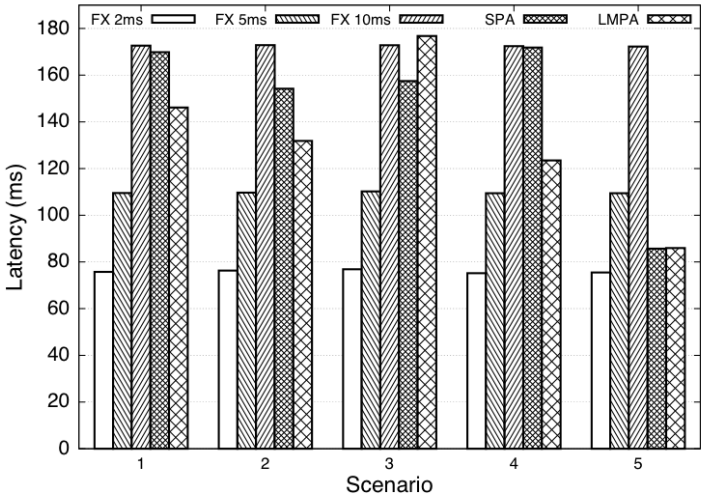

Figure 13: Latency for each scenario

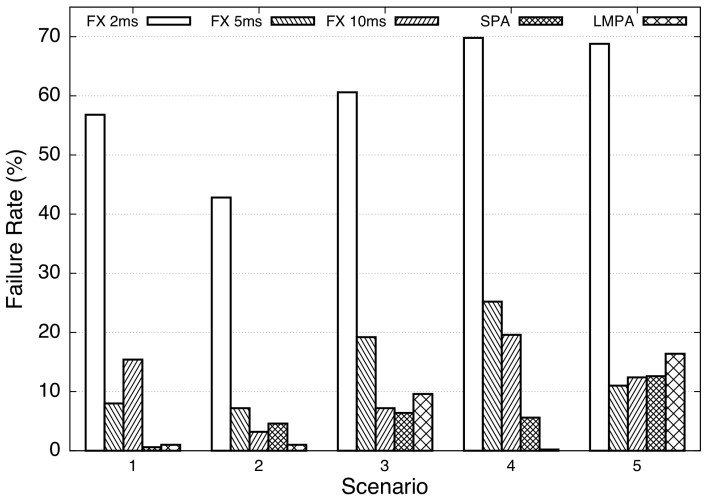

Figure 14: Failure Rate for each scenario

Acknowledgements

The authors would like to thank Saïd Hadin for his help and support during the implementation of the testbed.

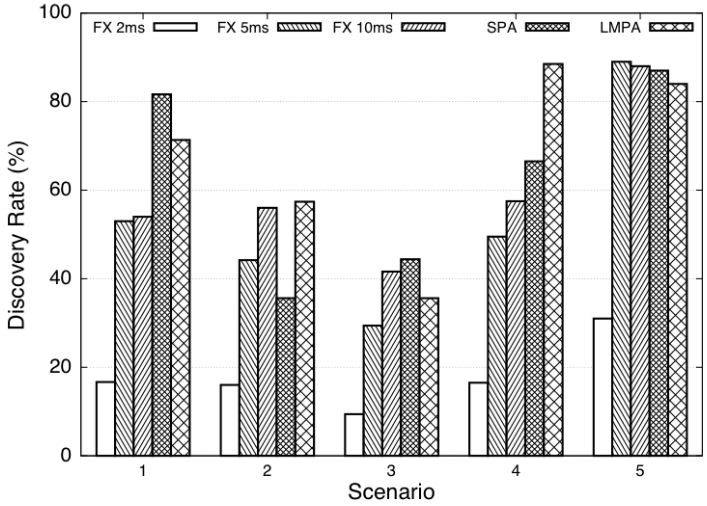

Figure 15: Discovery Rate for each scenario

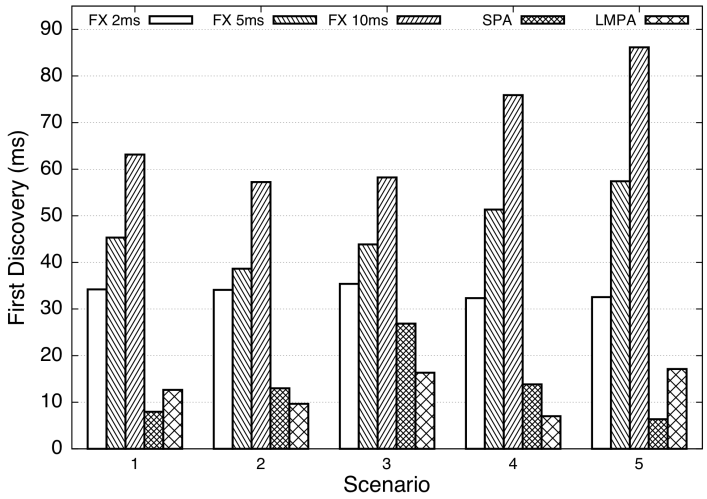

Figure 16: First Discovered AP Time for each scenario

References

[1] "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 Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, jun. 2007.

[2] E. Gustafsson and A. Jonsson, "Always best connected," *Wireless Communications, IEEE*, vol. 10, no. 1, pp. 49 – 55, 2003.

[3] C. Perkins, P. Calhoun, and J. Bharatia, "Mobile IPv4 Challenge/Response Extensions (Revised)." RFC 4721 (Proposed Standard), Jan. 2007.

[4] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6." RFC 3775 (Proposed Standard), June 2004.

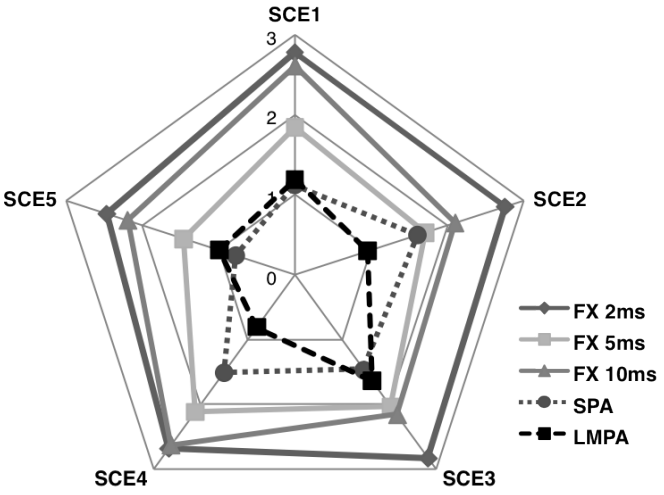

Figure 17: Score Function

[5] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, 2003.

[6] G. Castignani and N. Montavont, "Adaptive Discovery Mechanism for Wireless Environments," in *14th Eunice Open European Summer School*, 2008.

[7] G. Castignani and N. Montavont, "Adaptive system for 802.11 scanning," in *INFOCOM Workshops 2009, IEEE*, pp. 1 –2, 2009.

[8] G. Castignani, A. Arcia-Moret, and N. Montavont, "An evaluation of the resource discovery process in IEEE 802.11 networks," in *MobiOpp '10: Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, ACM, 2010.

[9] G. Castignani, A. E. A. Moret, and N. Montavont, "Analysis and evaluation of wifi scanning strategies," in *IV Cibelec 2010 : 4to congreso iberoamericano de estudiantes de ingeniera electrica electrica*, 2010.

[10] G. Castignani, A. Arcia-Moret, and N. Montavont, "A Study of the Discovery Process on 802.11 Networks," *SIGMOBILE Mob. Comput. Commun. Rev (MC2R)*, vol. to appear in, 2011.

[11] H. Velayos and G. Karlsson, "Techniques to reduce the IEEE 802.11b handoff time," in *Communications, 2004 IEEE International Conference on*, vol. 7, jun. 2004.

[12] 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*, mar. 2007.

[13] J. Teng, C. Xu, W. Jia, and D. Xuan, "D-scan: Enabling fast and smooth handoffs in ap-dense 802.11 wireless networks," in *INFOCOM 2009, IEEE*, pp. 2616 –2620, april 2009.

[14] Y. Liao and L. Cao, "Practical schemes for smooth MAC layer handoff in 802.11 wireless networks," in *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, 2006.

[15] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive Scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, may. 2007.

[16] J. Montavont, N. Montavont, and T. Noel, "Enhanced schemes for L2 handover in IEEE 802.11 networks and their evaluations," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 3, sep. 2005.

[17] S.-J. Yoo, N. Golmie, and H. Xu, "QoS-aware channel scanning for IEEE 802.11 Wireless LAN," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, sep. 2008.

[18] "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 Amendment 1: Radio Resource Measurement of Wireless Lans," *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)*, pp. c1 –222, jun. 2008.

www.telecom-bretagne.eu

Campus de Brest
Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
Tél. : + 33 (0)2 29 00 11 11
Fax : + 33 (0)2 29 00 10 00

Campus de Rennes
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
France
Tél. : + 33 (0)2 99 12 70 00
Fax : + 33 (0)2 99 12 70 19

Campus de Toulouse
10, avenue Edouard Belin
BP 44004
31028 Toulouse Cedex 04
France
Tél. : +33 (0)5 61 33 83 65
Fax : +33 (0)5 61 33 83 75

Imprimé à Télécom Bretagne
Dépôt légal : août 2011
ISSN : 1255-2275

TELECOM
Bretagne