# A study of the discovery process in 802.11 networks

German Castignani, Andres Emilio Arcia Moret, Nicolas Montavont

## To cite this version:

German Castignani, Andres Emilio Arcia Moret, Nicolas Montavont. A study of the discovery process in 802.11 networks. ACM SIGMOBILE Mobile Computing and Communications Review, 2011, 15 (1), pp.25-36. 10.1145/1978622.1978626 . hal-00609309

HAL Id: hal-00609309
https://hal.science/hal-00609309

Submitted on 18 Jul 2011

# A study of the discovery process in 802.11 networks

**German Castignani**[a, b] *german@castignani.com.ar*

**Andrés Arcia**[c] *amoret@ula.ve*

**Nicolas Montavont**[a, b] *nicolas@montavont.net*

[a] Institut TELECOM, TELECOM Bretagne, RSM department, Rennes, France
[b]Université Européenne de Bretagne, Rennes, France
[c]Universidad de Los Andes, Mérida, Venezuela

*Today wireless communications are a synonym of mobility and resource sharing. These characteristics, proper of both infrastructure and ad-hoc networks, heavily relies on a general resource discovery process. The discovery process, being an unavoidable procedure, has to be fast and reliable to mitigate the effect of network disruptions. In this article, by means of simulations and a real testbed, our contribution is twofold. First we assess the discovery process focusing on the values of IEEE 802.11 timers: MinChannelTime and MaxChannelTime. Then, varying these timers, we propose and evaluate an adaptive discovery strategy from which we obtain notable improvements over a fixed timers strategy.*

## I. Introduction

Nowadays, 802.11 wireless networks appear as the most popular access network since the demand for mobile accesses continuously increases. Moreover, modern portable computing devices such as PDAs and Cell Phones, which represent an important quantity of the Internet devices, embed WiFi chipsets. In this context, users can run applications and services over the Internet by accessing different networks depending on his/her location. The access to an 802.11 network can be achieved in two different modes, depending on the nature of the point of attachment. A mobile station (MS) can form spontaneous networks (ad-hoc mode) or it can get connected to an access point (AP) which is directly connected to a backbone (infrastructure mode). In both modes, *mobility* appears as the key benefit of 802.11, providing the users the possibility to move inside and between cells.

When moving out of the range of its current point of attachment (i.e., between cells), an MS should quickly discover and attach to a new point of attachment to reconnect to the network. This process is known as a handover. In infrastructure mode, it consists in finding a new AP. In ad-hoc mode, an MS may additionally need to discover new services, and eventually update routing states if multi-hops protocols are used. The wide usage of 802.11 networks implies that an MS may deal with a wide variety of deployment scenarios. These scenarios consist of heterogeneous wireless devices deployments (including APs), managed by several ISP and characterized by overlapping frequencies, different traffic load and high interference. These conditions cannot be anticipated by the moving MS and so the need for an appropriate scanning algorithm.

Independently of the access mode (ad-hoc or infrastructure), the scanning phase can be regarded as critical. When an MS starts up or moves around, it needs to discover its environment: radio frequencies, neighbor point of attachment (MS or AP), and available services. This process must be reliable, efficient and fast. In this article, we present experiments to assess the discovery process in 802.11 networks, and more specifically, we are interested in studying how long an MS has to wait before receiving a response from a point of attachment. We believe that these experiments and conclusions can be applied to other technologies, or extended to other discovery systems not only for 802.11 (e.g., Kozat and Tassiulas [1]).

Within the 802.11 scanning phase, an MS uses management frames called *Probe Request* to actively scan a channel and discover point of attachments operating on it. Nevertheless, in the infrastructure mode, an MS should start a discovery process each time it switches AP — known as Layer 2 handover — to join a new Basic Service Set (BSS). In the ad-hoc mode, an MS will start a discovery process to form an Independent Basic Service Set (IBSS) with its direct neighbors. Each time an MS moves, it needs to discover again its environment and join new service sets.

### I.A. The IEEE 802.11 Discovery Process

As shown in Fig. 1, an MS probes channels by broadcasting *Probe Requests* and waiting for *Probe Responses* from available points of attachment. The IEEE 802.11 standard [2] defines two timers,

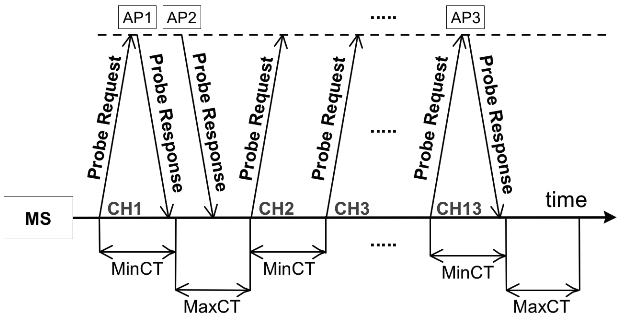

Figure 1: Standard Active Scanning

namely *MinChannelTime* (*MinCT*) and *MaxChannelTime* (*MaxCT*), that determine the time an MS needs to wait on a channel after having sent a Probe Request. Once a *Probe Request* is sent, *MinCT* defines the maximum time to wait for the first *Probe Response*. If a Probe Response is not received within *MinCT* (i.e., *MinCT* expires), the MS considers that the channel is empty, and starts the process in a different channel. Otherwise, if a Probe Response is received within *MinCT*, then the MS waits *up to MaxCT* for further Probe Responses that may be sent by other point of attachment operating within the same channel.

We have chosen to characterize the discovery process by two salient metrics: the *full scanning failure* and the *full scanning latency*. A *full scanning failure* is defined as the impossibility to discover any of the MSs or APs within all the available scanned channels. On the other hand, the *full scanning latency* corresponds to the time spent during the discovery process, i.e., to scan all available channels one after the other in whatever order.

In Eq. (1), we represent the full scanning latency ($L$) as a function of *MinCT* and *MaxCT*, and the probability of finding activity on a particular channel (c), referred as $P(c)$. Then, $N_{ch}$ refers to the total number of available channels [1]:

$$L = \sum_{c=1}^{N_{ch}} (1-P(c)) \cdot MinCT + P(c) \cdot (MinCT + MaxCT) \tag{1}$$

Since the arrival to the cell, an MS cannot be directly attached to any BSS or IBSS (i.e., the MS can not exchange data frames during the discovery process), making the discovery process mandatory. As explained in [3] and [4], concrete measurements of the handover latency in infrastructure mode demonstrates that the discovery phase takes about 90% for the total handover latency. Moreover, authors in [4] show the different approaches among manufacturers to implement different discovery techniques and settings for *MinCT* and *MaxCT*, and how variable is the AP's response time (i.e., from few milliseconds up to $40ms$).

In this article, we propose a set of experiments both by simulation and a real testbed, on the discovery process and focus on evaluating the impact of *MinCT* and *MaxCT* on the *full scanning latency* and the *full scanning failure*. In particular, we propose two strategies to set the values for *MinCT* and *MaxCT*. The first one bases on using fixed timers while the second one is to dynamically adapt *MinCT* and *MaxCT* from one channel to another during the scanning process. We have observed that fixed timers strategies are implemented in existent open-source 802.11 drivers, like MadWiFi[2] and ath5k[3].

The purpose of adapting the values of *MinCT* and *MaxCT* is not to determine the best values that fit a particular deployment, since we assume an unknown and unpredictable deployment and so the topology on every discovery process is rather unknown. Thus we aim at finding a trade-off between a minimal full scanning latency and a minimal full scanning failure. Recall that when decreasing the latency we increase the failure and vice-versa. The principle is thus to lower *MinCT* and *MaxCT* values when a point of attachment has already been discovered, and on the opposite, to use higher values when no point of attachment has been found. Moreover we will see that the sequence in which the channels are scanned have an impact when *MinCT* and *MaxCT* are adaptable.

The remaining of the article is organized as follows. In Section II we survey the related work and present a taxonomy of the different techniques to improve the discovery process. In Section III we introduce two different strategies to set the timers during a scanning process. In Section IV we evaluate the performance of both strategies by simulation and a real testbed. Finally, in Section V we conclude the article.

## II. Related Work

Most of the related work done for the 802.11 discovery process concerns the optimization of the Layer 2 handover, when an MS roams from one AP to another. In this section, we present the main strategies to reduce the *full scanning latency*.

[1] Recall that this number depends on regional regulations

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

[3] http://wireless.kernel.org/

### II.A. Selective Scanning and Caching

One simple way to reduce the *full scanning latency* is to only scan a subset of channels, instead of probing each of them. In [5], authors suggest the utilization of a *channel binary mask* to decide which channels to scan. This mask is updated after each handover. During the first handover, the mask is initialized with ”1” for all channels, meaning that all channels are scanned. During a handover, the MS builds a new mask for the next handover, containing a value of ”1” for the non overlapping channels $(1, 6$ and $11)$ and for those where a probe response was received. The mask contains ”0” for channels that may not have activity (i.e., no Probe Response was received on the previous scanning). Upon the next handover, the channel on which the MS’s AP was operating is turned to ”0” in the mask, since authors consider that a neighboring AP operating on the same channel is not probable. This consideration contradicts the statement presented in [6], where a neighboring AP on the same channel is considered highly probable. Then, channels marked as ”1” are scanned, if no probe responses are received on those channels, the mask values are logically inverted and the MS continues probing these new channels. If the scanning process is still unsuccessful, a standard full scanning process is executed all over again. In addition, the authors propose to use a *Caching* method where neighbor APs are stored during MS operation. This table will allow the MS to directly probe a neighbor AP when it returns to an AP that has already been visited. Selective Scanning reduces the *full scanning latency* in an average of $43\%$. Applying the Caching mechanism, the handover latency is reduced to reauthentication and reassociation delays, reaching a 97% of reduction. Regardless of these results, since they do not require modifications on the AP side, it has to be counted that the neighbor APs cache has to be carefully maintained. Erroneous information in the cache such as unavailable APs, leads to *full scanning failure*. On the other hand, as both the cache and the binary mask are incrementally built, the first handover will apply the standard technique, resulting in higher latencies.

### II.B. Reducing the time spent on each channel

One of the paradigms within the handover optimization has been focusing on reducing the value of the scanning timers. Several works based on simulations proposed different values for *MinCT* and *MaxCT* timers. Velayos and Karlsson [7] focus on fixing the best values for both timers presenting theoretical considerations and simulation results. For *MinCT*, authors establish the concrete value for the maximum time an AP needs to answer a probe request, considering that both the AP and the channel being probed are idle. If propagation time and probe response generation time are neglected, then the 802.11 MAC Distributed Coordination Function (DFC) establishes that the maximum response time has the form of equation 2, reaching $670\mu s$ (approximated to $1TU^4$). So *MinCT* should allow the station to wait first for DIFS (DCF Interframe Space) and then for the backoff (with maximum value for the contention window during the first transmission attempt, $aCWmin$).

$$\begin{aligned} MinChannelTime &= DIFS + (aCWmin \cdot aSlotTime) \\ &= 50\mu s + (31slot \cdot 20\frac{\mu s}{slot}) \\ &= 670\mu s \\ &\cong 1TU \end{aligned} \tag{2}$$

Authors analyse the probe response delay depending on traffic load and the number of stations on each channel. They conclude that *MaxCT* is not bounded as long as the number of stations can increase. They recommend to set *MaxCT* to avoid responses from overloaded APs. They fixed a value of $10TU$ based on the hypothesis that ten MS associated with the same AP is an adequate number in order to achieve good throughput. However, the fact of providing fixed timers does not guarantee a successful discovery process. Authors introduced several considerations regarding the number of stations operating on each channel and data traffic conditions. These fixed values could effectively work for some scenarios, but in other cases unnecessary delays may be introduced or even worse, the scanning may fail to find any candidate AP, resulting in a link layer disconnection.

### II.C. Interleaved scanning sub-phases

The 802.11 standard active scanning algorithm implicitly defines that the handover process should be performed after detecting weak signal from the current AP. The *Smooth Handover* [8] and the *Periodic Scanning* [9] methods are based on splitting the discovery phase into multiple sub-phases. The objective of this division is to allow an MS to alternate between data packet exchange and the scanning process. An MS periodically performs anticipated short discovery phases so it can look for candidate APs

[4] One Time Unit (TU) is equal to $1024\mu s$

before reaching a disconnection. During the anticipated scanning, the station builds a list of target APs maintaining some basic information (MAC address, operating channel and Service Set Identifier (SSID)). Authors of [8] propose to scan a group of channels in each sub-phase, while in [9] only one channel is scanned during *MinCT*. Each sub-phase is triggered depending on the Received Signal Strength Indicator (RSSI).

Authors of [8] evaluated the performance of the Smooth Handover in a real testbed and showed that the data packet loss is strongly reduced. In [9], network simulations on six different scenarios are proposed. The handover delay and the packet loss rate are reduced only in some scenarios, depending on the characteristics of the APs deployment. In other scenarios, the station continuously probes and switches between channels which contributes to a higher battery consumption, and some additional delays.

In addition, these techniques require that there must be *enough overlapping area* between neighboring APs, limiting the deployment scenarios where these techniques may be applied. If only small overlapping areas exist, there will not be enough time to distribute the scanning process during the MS movement. The need for overlapping area between neighboring APs strongly constrains the network deployment and requires to deploy more APs in a given area.

### II.D. Synchronized passive scanning

Unlike common handover optimizations focusing on active scanning, the SyncScan [10] method is based on the standard passive scanning approach where an MS simply waits for periodic beacons on the current channel. The passive scanning latency is related to the number of channels and the *BeaconPeriod* timer, commonly set to $100ms$. Then, passive scanning latencies usually exceed one second. SyncScan synchronizes the MS at the same time the APs beacons are received on each channel, so the MS switches to a channel when a beacon is about to arrive. Using SyncScan the MS has up-to-date information about APs. Since the MS does not probe the channels, the handover latency should be reduced to authentication and association delay.

This new approach may eliminate the scanning delay, but some difficulties should be analyzed. The fact that the MS must switch to a channel when a beacon is *about* to arrive, adds a complex time synchronization management between MSs and all deployed APs. Clock accuracy becomes critical in this approach because even a minor deviation in time synchronization becomes non-negligible preventing an MS from discovering neighbor APs. Authors propose the usage of Network Time Protocol (NTP) that maintains time within $10ms$ accuracy over the Internet, achieving precisions of $200\mu s$ or better in local area networks under ideal conditions. Under these considerations, we believe that SyncScan implementation is limited to very homogeneous deployments (e.g. enterprise or campus deployments), where a central administrator can manage the channel allocation and synchronization between APs for the beacon sending. Synchronizing APs in a fully heterogeneous environment (e.g. hotspots or community deployments from multiple operators around a city) for the implementation of SyncScan seems impractical. In all the cases, the SyncScan procedure is performed regularly, producing several unavailable periods for data packets transmissions, so packet loss may be observed while exploring other channels.

## III. Timers Setting Strategies

Although optimization techniques have been designed for the discovery process, there is still a lack of work in the determination of the most adequate values defining the time to wait on each channel. For every fast handover approach, an MS still needs to scan channels one after the other to discover APs. In smooth handover [8] or in periodic scanning [9], the discovery phase is split into several independent subphases that are separated by certain time period (during which the MS may still exchange data packets). During each of these sub-phases, the MS scans the channels one by one, just as in a continuous scanning phase. In the selective scanning [5], the order in which channels are scanned is determined by a binary mask built from previous scanning phases. In all cases, for each channel, APs also need to be probed and thus the time to spend on each channel needs to be defined. In any other method, such as the synchronized passive scanning [10], there is still a probability that no AP is found through the optimized method. In fact, the optimization proposed in [10] cannot be applied in an opportunistic scenario, since synchronisation between possible neighbours is not achievable. In case the optimized method fails, we need a fall-back mechanism to discover AP, i.e., we need to scan the channels one by one because all other alternatives failed.

To determine the time needed for an MS to wait for a Probe Response on each channel, we study the impact of *MinCT* and *MaxCT* on the discovery process, introduced in Section I.A. We define in this section two strategies to set the values for these timers. The first method is the most intuitive and consists in scanning channels one after the other with fixed values for both *MinCT* and *MaxCT*. The other method, referred as adaptive timers scanning, consists in varying the values of *MinCT* and *MaxCT* channel by channel, according to the AP(s) that was (were) already found. By adjusting these values, we expect to provide a better success rate for the AP discovery (i.e., low *full scanning failure*) than using a fixed timers strategy, while maintaining a low *full scanning latency*.

### III.A. Fixed Timers Scanning

This first strategy consists in fixing pre-defined values for both *MinCT* and *MaxCT*, which determine the time an MS will wait on a channel for AP's responses. Low values will provide low *full scanning latency*, but will increase the risk of missing an AP because the MS is not waiting long enough to get a response. While theoretically an MS should expect a response before $1ms$ (see Section II.B), experimental results presented in Section IV.C suggest that the response from an AP varies from $1ms$ to $40ms$. Considering the empirical analysis proposed by Mishra et al. [4], and our experience, we decided to evaluate the following timers: $< 10ms, 20ms >$ and $< 25ms, 50ms >$ for $< MinCT, MaxCT >$. These sets of timers try to represent two limit cases for the *full scanning latency* and *full scanning failure* trade-off. The first set prioritizes the *full scanning latency* while the second try to avoid high levels of *full scanning failure*. In both sets, we fix the double of the value of *MinCT* for its correspondent *MaxCT*. This ratio was not varied during simulations and experimentations. Different values for *MaxCT* for the same *MinCT* may increase the number of discovery APs in some scenarios, but it does not affect the trade-off under study.

As stated in Section I, fixed-timers strategies are commonly implemented in open-source drivers. In the case of MadWiFi, two timers are implemented: *mindwell* and *maxdwell*. Once a Probe Request is sent on each channel, the MS waits until maxdwell ($200ms$) for a Probe Response or Beacon. If at least one Probe Response or Beacon is received, and if mindwell ($20ms$) elapsed, the station immediately switches to the next channel in the sequence. Then, the *full scanning latency* could vary between $260ms$ and $2.6s$ (without considering the channel switching delay), depending on the number of channels with APs deployed. In the case of ath5k, an MS first waits for $100ms$ on the current channel and then it sequentially scans using two timers. It first waits for the *Probe_Delay* timer ($28ms$) and then it sends a Probe Request and waits for *Channel_Time* ($28ms$) for the Probe Responses. In this case, the *full scanning latency* is close to one second. Both implementations differ from the 802.11 standard (see Section I.A). Moreover, authors of [4] present an analysis of the scanning phase for different 802.11 network interface cards. A very large variation of the *full scanning latency* is observed depending on the AP and MS technology. They found that fixed timers strategies differ from one MS to another, since they implement alternatively one or two timers, using different values. The maximum average variation found in this works was between $58.74ms$ and $394.27ms$ for the same AP configuration using different cards in the MS. As we can see, these handover latencies are too high for real time applications.

### III.B. Adaptive Timers Scanning

The other possible strategy is to adapt, or dynamically change the values for *MinCT* and *MaxCT* during the scanning process based on the discovered resource. This new approach allows an MS to spend less time on channels once candidate APs have been already found whereas the fixed timers scanning would spend the same time on each channel. The main goal is to reduce the timers channel by channel while APs are discovered, because the impact of missing an AP will be less important as if no AP were found. On the contrary, timers may be increased if no AP has been found, in order to increase the chances of finding an AP on the next channel(s). In this article, we do not investigate a proper adaptation function, however we rather focus on the understanding of the discovery process using either fixed timers or variable timers. We present the adaptation function used in our experiment in Section IV, but other adaptation function could be implemented.

The selection of the sequence of channels to scan becomes important if we consider timers adaptation, because timers are adapted according to the activity on each channel. The sooner an AP is found, the faster the timers will be decreased, and thus, it is important to scan first channels on which AP(s) may be operating. In 802.11 networks, only three non-overlapping channels exist. As stated in [11] and [12], a proper deployment typically uses only these channels. Moreover, Eriksson, et al. [13] and Gerla et al. [14] presented two different works in which the channel occupancy distribution is calculated in a real environment. In [13], the authors propose an optimal scanning strategy that gives more priority to channels 1, 6 and 11, since they found that 83% of APs are assigned to those channels. On the other hand, experimenting over a different deployment, the authors of [14] states that 77,98% of APs are deployed in the non-overlapping channels. Our own experiments on a city-wide WiFi deployment results in 78,21% of occupancy in non-overlapped APs. Then, it could be assumed that prioritizing those channels, as stated in [5] and [13], candidate APs may be discovered sooner. We suggest to randomize the channel switching sequence in two different subsequences. The first subsequence randomly switches between the non-overlapping channels. Then, the rest of the channels are also randomly considered. If an AP with relative good signal level is discovered in channels 1, 6 and/or 11, the adaptive system will set lower timers for the next channels to scan. In all cases, timers are adapted between pre-established bounds (defined by experimentation on Section IV.C.2). This strategy differs from that proposed in [13], where channels are scanned at a rate proportional to the channel occupancy, with the final goal of minimizing the time to find a channel with an AP. In our case, we consider scanning all the channels in the sequence, but giving more priority to the non-overlapping channels, i.e., increasing the probability that a high timer will be used on those channels.

# IV. Performance Evaluation

## IV.A. Strategies Implementation

To evaluate both timers setting strategies, we have conducted simulations and experimentations under a real testbed. The simulation aims to evaluate the scanning process in different AP deployments, considering different time an AP employs to respond to a request. The main goal of the simulations is to roughly show the expected behavior of the different algorithms in terms of the *full scanning latency* vs *full scanning failure* trade-off, giving a global view of the problematic. Since the implementation of the simulator is mainly based on computing the expected *full scanning latency* and *full scanning failure* for different probe responses delay distribution, we decided not to implement or modify an existent network simulator, but to develop the algorithms in a simple light weight simulator in the language C. Within these algorithms, a probe response delay computation and an exponential back-off for retransmissions have been implemented.

The testbed is based on the deployment of 802.11 MSs and APs, allocated in different channels and under different traffic conditions, as detailed in Section IV.C.1. The scenarios experimented in this testbed are a subset of the simulation space in order to focus on some (limited) representative scenarios. These experiments also show that an adaptive strategy can be implemented in an open-source driver and allow measuring the handover performance with real devices.

On the one hand, the fixed timers strategy is implemented as described in the previous section with the timers $< 10ms, 20ms >$ and $< 25ms, 50ms >$, which give a good overview of the performance of using fixed timers. On the other hand, for the adaptive strategy, we need to define how the timers evolve from one channel to another. We decided to increase the timers when no AP is found, and to decrease the timers when an AP is found. The decrease of the timers are proportional to the quality of the discovered AP, which is computed from the signal level and the number of APs sharing the same channel. Note that the definition of a proper metric for AP quality is out of the scope of this paper, so other algorithms might be used. In the following paragraph, we give the details of our implementation.

As illustrated in Fig. 2, the MS starts scanning using half the maximum bounds for both timers. We considered this strategy as an approximation to balance the trade-off between the *full scanning latency* and the *full scanning failure*. Observe that, if an MS started scanning using the maximum bounds of the timers, we would end up with a high *full scanning latency*. Otherwise, if an MS started scanning with the lower bounds of the timers, we would fall in a high *full scanning failure*. For a channel on which at least one AP has been discovered, the MS calculates the greatest quality of all discovered APs on that channel ($Q$) and the number of APs that have replied on that channel ($N$). $N$ is obtained by simply counting the number of probe responses received from different APs. Regarding $Q$, the *Received Signal Strength Indicator* (RSSI) parameter, included on each received probe response, is considered. Both $Q$ and $N$ are combined in order to establish the criteria that will be used to rank discovered APs, and decide whether the timers to be used on the next channel ($T_{n+1}$) can be reduced or increased. In Fig. 2, $T_{n+1}$ represents the tuple $< MinCT, MaxCT >$, since both timers are simultaneously adapted. Then, $T_{n+1}$ is calculated considering a *decision making* parameter ($R$) calculated for each channel by using $Q$ and $N$.

$$R = \frac{Q}{N} \tag{3}$$

This simple relation allows adapting timers differently, depending on the environment that have been discovered on the previous channels. Two different

APs having the same signal strength and operating in separated channels may be considered in a different way. Populated channels will not be weighted as well as those with a lower number of APs. This choice comes from the observation that in a wireless environment, we mainly consider *weak signal* and *collisions* as the issues limiting link performance. A packet transmitted on an 802.11 link may be lost because of a weak signal or a collision, but discerning the real cause is quite difficult. We focus on results obtained in [15], in which a set of testbeds were implemented so as to independently analyze the effect of a weak signal (due to a low RSSI between the MS and the AP) and collision (due to several MSs and APs operating on the same channel). Authors have empirically showed that for the same packet-loss rate, the BER (Bit Error Rate) in a weak signal scenario (low RSSI, without collisions) was less than 12% against a 50% for a collision scenario (without weak signal effects). We can infer that the effect of multiple APs sharing the channel (producing collisions) is less desirable than a low RSSI scenario.

Then the MS takes into account the value of $R$ calculated on the channel and reduces both *MinCT* and *MaxCT* using the same proportional factor ($f(R)$). $f(R)$ is implemented in away that for higher values of $R$, timers are more strongly reduced. In contrast, as shown in Fig. 2, if no AP is discovered, no $R$ is calculated on the correspondent channel and then timers are increased to half the difference between the last successful timers (those from whom at least one response was received from an AP) and the timers used on the previous channel (those from whom no response was received from any AP). Increasing timers using this approach avoids overshooting, since timers are smoothly augmented. Recall that the purpose of this article is not to provide the best adaptive strategy, but evaluate the impact of timers on the *full scanning latency and failure*. Other adaptive strategy could be proposed and additional or alternative parameters could be taken into account to estimate the quality of the discovered APs.

## IV.B. Simulation results

We describe an evaluation done by simulation of both fixed and adaptive timers strategies in 25 different scenarios. For each scenario, there is either 0 or at maximum 1 AP per channel. This is to simplify the simulation, since we are interested only if the channel has activity or not. In all cases, the quality (RSSI) of each AP is randomly generated. The 25 scenarios are the following:

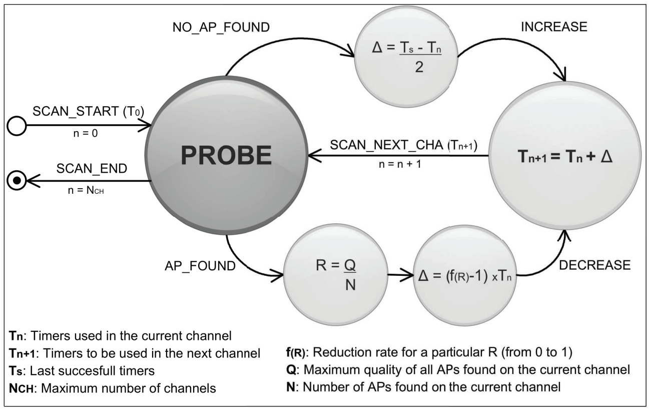
NO_AP_FOUND
Δ = Ts - Tn / 2
INCREASE
SCAN_START (T0)
n = 0
PROBE
SCAN_NEXT_CHA (Tn+1)
n = n + 1
Tn+1 = Tn + Δ
SCAN_END
n = NCH
AP_FOUND
R = Q / N
Δ = (f(R)-1) xTn
DECREASE
Tn: Timers used in the current channel
Tn+1: Timers to be used in the next channel
Ts: Last succesfull timers
NCH: Maximum number of channels
f(R): Reduction rate for a particular R (from 0 to 1)
Q: Maximum quality of all APs found on the current channel
N: Number of APs found on the current channel

Figure 2: Adaptive Strategy Implementation

- **12 optimistic scenarios**, where the APs are deployed in the first scanned channels. The first optimistic scenario corresponds to the case where there is only one AP in the first scanned channel, and then no more APs are deployed on the other channels. Following, the second, third and next optimistic scenarios corresponds to two, three and more subsequent occupied channels (with an AP operating) and then the rest of the channels are empty (no available AP).
- **An ideal scenario** where 13 APs are deployed one by one in the 13 available channels.
- **12 pessimistic scenarios**, where APs are deployed in the last scanned channels. The first pessimistic scenario corresponds to the case where all channels are empty (no AP), except the last scanned channel where one AP is operating. Then, more APs are deployed on the last scanned channels, up to the case where only the first scanned channel in the sequence has no AP.

We identified both optimistic and pessimistic channel sequences since the adaptive strategy depends on when APs are discovered in the sequence of scanned channels. In order to evaluate the impact of different probe response delays, for each of these 25 scenarios we performed 10 different simulations. We consider $P$ from 10% to 100% of probes responses received before $10ms$, with a step of 10 points. Then, we generate uniform random probe response delays between 0 and $10ms$ with probability $P$ and values greater than $10ms$ with probability $1 - P$. We chose a uniform random law because Mishra et al. [4] suggests that the time of response from an AP follows a uniform law. For each simulation, a hundred thousand scanning experiences is performed in order to obtain

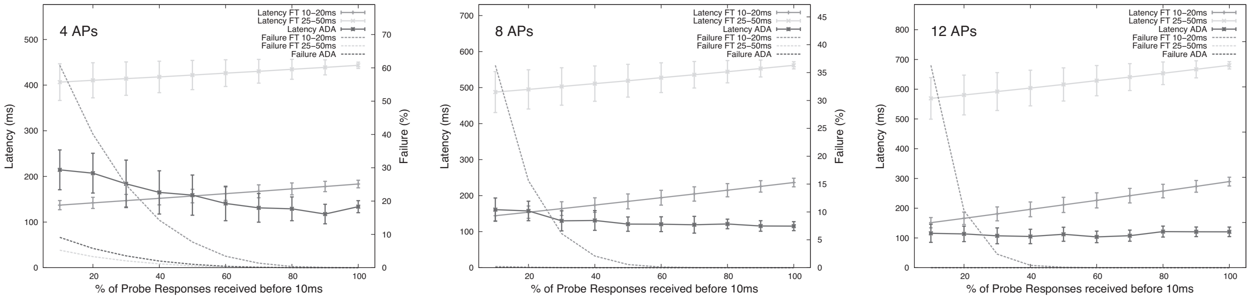

Figure 3: Simulation results for 4, 8 and 12 APs using the optimistic sequences

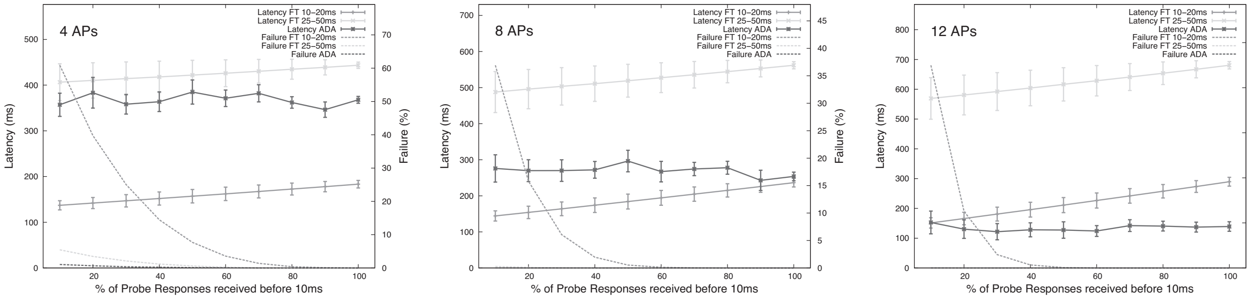

Figure 4: Simulation results for 4, 8 and 12 APs using the pessimistic sequences

averaged results.

For space reasons, we only present the results for 4, 8 and 12 deployed APs for the optimistic and pessimistic channel sequences in Fig. 3 and 4. These figures show the *full scanning latency* on the left ordinate and the *full scanning failure* percentage on the right ordinate according to the probability of receiving a Probe Response before a given percentage in abscissa. Focusing first on Fig. 3 (optimistic scenarios), we can appreciate that the fixed timers strategy (using both sets of timers) tends to increase the *full scanning latency* (crosses and minus signs with error bars) when the number of probe responses received before $10ms$ increases. This is due to the effect of *MaxCT*, since if activity is detected on more channels the MS waits more time on each one.

On the other hand, the adaptive strategy tends to decrease the *full scanning latency* (stars with error bars) when the number of probe responses received before $10ms$ increases. This occurs because the adaptive strategy reduces the timers for the following channels to scan if activity is detected on the current scanned channel. Then if more activity is detected (due to more probe responses received on different channels) the adaptive strategy is able to reduce even more the timers. We can appreciate that there is an intersection between the *full scanning latency* curves for the fixed timers strategy using $< 10ms, 20ms >$ and the adaptive one in all the optimistic sequences scenarios. In the zone where *full scanning latency* of the adaptive strategy is higher than the fixed timers strategy (from the intersection point to the left), the fixed timers strategy performs worse in terms of *full scanning failure*, reaching very high levels in comparison to the adaptive strategy.

The *full scanning failure* for the fixed timers strategy using $< 10ms, 20ms >$ is always high for a low percentage of probe responses received before $10ms$ (between 20% and 60% for the scenario with 4 APs). On the other hand, the adaptive strategy tends to maintain the *full scanning failure* as low as possible while reaching a *full scanning latency* comparable to the fixed timers strategy using $< 10ms, 20ms >$. We can also appreciate that the higher the number of deployed APs, the better *full scanning latency* for the adaptive strategy. For 12 APs, the adaptive strategy is always better (in terms of *full scanning latency and failure* than the fixed timers strategy.

As shown in Fig. 4 (pessimistic scenarios), the adaptive strategy performs worse in terms of *full scanning latency* than in the optimistic sequences scenarios because the timers are increased during the first scanned channels as no AP is discovered. However, it always maintains *full scanning failure*, under 1%. When 4 or 8 APs are deployed, the *full scanning latency* for the fixed timers strategy is always below the

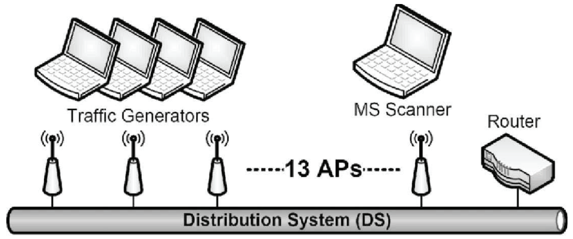

Figure 5: Testbed configuration

adaptive one (for 4 APs, we even observe an important difference, where the fixed timers strategy requires $150ms$ in average, and the adaptive strategy requires 380ms). However, for a low percentage of Probe Response received before $10ms$ (less than 50%), the fixed timers strategy shows high full scanning failure (between 10% and 60%).

In summary, these simulations highlight that the discovery process is sensible to *MinCT* and *MaxCT* variations, the delay of the first response received from an AP, the APs deployments and the sequence in which the channels are scanned when *MinCT* and *MaxCT* are dynamically adapted. The literature often evaluates the efficiency of the discovery process by measuring the *full scanning latency*, but we can see that the discover process has also an important impact on the *full scanning failure*.

## IV.C. Experimentation

### *IV.C.1. Testbed*

A real testbed was implemented using up to thirteen APs from different providers and seven MSs for traffic generation (see Fig. 5). All the equipments in the testbed implemented 802,11b as physical layer. The MS uses an Atheros based D-LINK DWL-AG660. Both scanning strategies defined in Section III were implemented inside the *MadWiFi* driver (Version 0.9.4). Up to 64 different network scenarios were evaluated using eight different channel allocations, two different traffic conditions and four configurations for *MinCT* and *MaxCT* timers. Traffic was generated using *D-ITG* (Distributed Internet Traffic Generator), producing, in all configurations, a load of 8 Mbit/s between one sender and one receiver, which leads to overloaded cells. With regard to the channel allocation, the following configurations were evaluated.

- **Configuration 1**: Thirteen APs allocated one by one on channels 1 to 13 (one AP per channel).

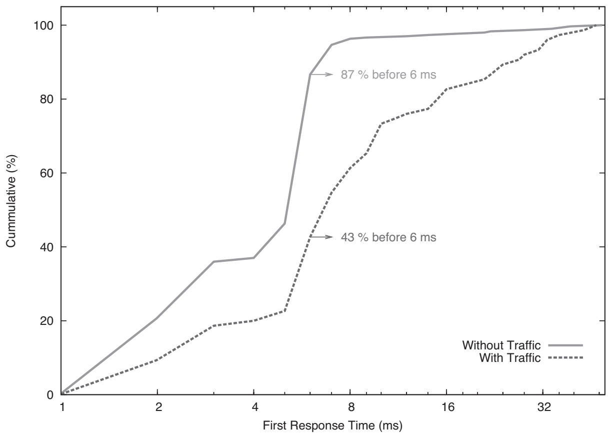

Figure 6: First Probe Responses Delay

- **Configuration 2**: Thirteen APs all allocated on channel 11 (13 APs on the same channel).
- **Configuration 3**: Three APs allocated one by one on channels 1-6 -11 (one AP per channel).
- **Configuration 4**: Twelve APs allocated four by four on channels 1-6-11 (4 APs per channel).

We have selected these configurations from all the possible cases since they are a valid representation of real deployments (as shown in [14] and [13]) and include favourable cases (for the discovery process) as well as challenging environments. In each of our experiments, we measure the discovery process over a hundred of full scanning processes, i.e., where all channels are scanned by the MS one by one.

### *IV.C.2. Bounds Determination for MinCT and MaxCT*

In the first part of the experience, we aim at determining the bounds for *MinCT* and *MaxCT* for our adaptive strategy, i.e., the intervals in which *MinCT* and *MaxCT* will vary (defined as *MinLower*, *MinUpper*, *MaxLower* and *MaxUpper*). For this purpose, we measure the *delay of the first and further received probe responses* on each channel for each particular AP deployment configuration, with and without traffic. Further probe responses are those that arrive after the first until the last probe response. We configured the MS with $< 50ms, 200ms >$ for $< MinCT, MaxCT >$ in order to allocate enough time for the discovery of all operating APs (we were not interested in the *full scanning latency*, but in collecting the delay of each Probe Response).

Table 1 gives the main conclusion for bounds determination. We have considered optimistic and

Table 1: Bounds for *MinCT* and *MaxCT*

| Bound | Value | % of Probe 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 |

pessimistic scenarios to define the upper and lower bounds. We also considered different percentages of received probe responses depending on the considered timer (e.g., MinUpper should represent a high value of probe response received because this bound will be used when no AP has been found). Fig. 6 shows a cumulative distribution function of the delays of the first probe response (in response to the broadcasts probe request) under a three non-overlapping channel configuration with and without traffic (configuration 3). This scenario is considered as an ideal AP configuration where interferences are minimized and thus help to determine the minimum limits for *MinCT*. If we observe the accumulated percentage function of first probe responses received over all the trials without traffic, we can see that 87% of the first Probe Responses were received before $6ms$. Thus we decide to configure *MinLower* at $6ms$ as stated in Table 1. We can allow this relative low percentage - we could have taken $8ms$ where 96% of the Probe Responses were received - because we can afford to risk few unsuccessful discoveries in our adaptive strategy when this minimum value is used. Note that in the considered adaptive strategy, this minimum value is only used when APs have been discovered previously (See Section IV.A).

With the same aim and considering configuration 4, without traffic, *MaxLower* is set at $8ms$ where 50% of following Probe Responses from other APs were already received. We let *MaxCT* to be adapted to a low limit that covers less cases than *MinCT* (only a 50%), since the situation of not discovering more APs is not as risky as not discovering the first AP, in which the channel will be declared empty.

On the other hand, the upper bounds *MinUpper* and *MaxUpper* are determined using results obtained on the other scenarios (which are highly affected by interference) including traffic, like configuration 1 and 2. Cumulative functions are not illustrated due to space reasons, but we decided to pick $34ms$ for *MinUpper* (96% of further Probe Responses received in configuration 1) and $48ms$ for *MaxUpper* (87% of further Probe Responses received in configuration 2).

### *IV.C.3. General Results*

During the second part of the experience, the adaptive strategy (ADA) was tested using bounds defined in Table 1 and the fixed timers strategy was evaluated considering three different sets of timers, $< 10ms, 20ms >$, $< 25ms, 50ms >$ and finally $< 50ms, 200ms >$ for *MinCT* and *MaxCT* respectively. Table 2 shows the results organized by scenario, where the *full-discovery rate* indicates in how many scanning processes all available AP were discovered. The *failed scanning* values describe the *full scanning failure*. Finally the average *full scanning latency*, including the standard deviation ($\sigma$) for the adaptive strategy, shows a controlled dispersion of the obtained latencies.

The main observation of these experiments confirm those of the simulations: the discovery process performance highly depends on the deployment scenarios and a high *full scanning failure* may be observed for the fixed timers strategy in some common network scenarios. The adaptive strategy only has 2% of *full scanning failure* in a single scenario (configuration 2 and loaded cells) and keeps a low *full scanning latency*. A detailed analysis of results of Table 2 is presented in the following paragraphs.

**Impact of Traffic Load** - Fig. 6 illustrates configuration 3, where probe responses are notably delayed when traffic is injected. While before $6ms$ 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 2, in the case of configuration 3 with traffic, for several scannings a probe response is not received before $25ms$, causing 20% of *full scanning failure*. Even using a *MinCT* equal to $50ms$ the *full scanning failure* arrives to 13%. The effect of traffic also produces a decrease in the average number of discovered APs in all evaluated scenarios. The adaptive strategy helps to reduce the effects of traffic, since no scanning process fails except in one scenario, where we observe only 2% of *full scanning failure*.

**Theory vs Experimentation** - Our experimental results do not match theoretical considerations and simulation presented in [7]. In this work, a value of $1ms$ for *MinCT* is considered enough to wait for the first probe response before switching channel (see Section II.B). On the other hand, our experience shows that *MinCT* needs to be greater than $10ms$ to receive the 97% of first probe responses in an ideal three non-overlapping channel scenario without

Table 2: Comparative results

| **Scenario** | | | | **Full Discovery Rate (%)** | | | | **Full Scanning Failure (%)** | | | | **Full Scanning Latency (ms)** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AP conf. | Channels | Number of AP | Traffic | Fixed Timers | | | ADA | Fixed Timers | | | ADA | Fixed Timers | | | ADA | |
| | | | | 10-20 | 25-50 | 50-200 | | 10-20 | 25-50 | 50-200 | | 10-20 | 25-50 | 50-200 | | $\sigma$ |
| 1 | 1 to 13 | 13 | No | 65% | 87% | 93% | 49% | 0 | 0 | 0 | 0 | 275 | 708 | 2567 | 256 | 11% |
| 1 | 1 to 13 | 13 | Yes | 24% | 69% | 82% | 40% | 2% | 0 | 0 | 0 | 317 | 636 | 2378 | 248 | 13% |
| 2 | 11 | 13 | No | 75% | 92% | 94% | 96% | 2% | 2% | 0 | 0 | 152 | 360 | 807 | 423 | 3% |
| 2 | 11 | 13 | Yes | 54% | 88% | 98% | 83% | 29% | 3% | 0 | 2% | 159 | 363 | 814 | 434 | 5% |
| 3 | 1-6-11 | 3 | No | 92% | 94% | 99% | 94% | 0 | 0 | 0 | 0 | 117 | 414 | 1119 | 190 | 11% |
| 3 | 1-6-11 | 3 | Yes | 38% | 51% | 61% | 81% | 52% | 20% | 13% | 0 | 227 | 403 | 1025 | 210 | 18% |
| 4 | 1-6-11 | 12 | No | 98% | 98% | 100% | 95% | 0 | 0 | 0 | 0 | 179 | 419 | 1121 | 390 | 3% |
| 4 | 1-6-11 | 12 | Yes | 39% | 60% | 87% | 84% | 13% | 1% | 0 | 0 | 239 | 450 | 1110 | 378 | 13% |

traffic. Moreover, as shown in Fig. 6 earliest first probe responses only appear after $2ms$ in the same configuration. Regarding *MaxCT* authors of [7] state that it is unbounded and claims for a *MaxCT* equal to $10ms$ to be enough. We have shown during the bound determination that this value could not be sufficient for some scenarios. This gap between results proposed in [7] and our experimentation (that are close to those presented in Mishra et al.[4]) are still under research. It may be explained by additional delays neglected in the literature, such as channel switching, congestion condition and queueing architecture for management frames.

**Impact of the Number of APs** - In configurations 3 and 4 where only non-overlapping channels were used, we observe less *full scanning failure* when there are four APs operating on the same channel (configuration 4). When there is a single AP per channel (configuration 3), a higher *full scanning failure* is attained in all evaluated timers for the fixed timers strategy. This may be due to the backoff timer of the MAC protocol, since there are more chances to pick a small random number when there are more active APs.

**Impact of full scanning latency** - *Full scanning latency* depends on the values of *MinCT* and *MaxCT* during the discovery process. In the adaptive strategy implemented by experimentation *MinCT* is initially set to *MinUpper* and it gradually decreases until *MinLower*. Fig. 7 shows *full scanning latency* values for all configurations with traffic including the *full scanning failure* (in percentage) for each case. Even if fixed timers strategy may give good results in some scenarios, the adaptive strategy provides lower or equivalent *full scanning latency* from $190ms$ to $434ms$. The fixed timers strategy configured with $< 10ms, 20ms >$ gives better latencies in AP configuration 2 around $150ms$ against $420ms$ for the adaptive strategy. But in this case *full scanning failure* reaches $29\%$, while the adaptive strategy only produces a $2\%$. Moreover, in configuration 3 with traffic, the adaptive strategy gives the best *full scanning latency* without any *full scanning failure*, while all other evaluated strategies reach high levels of *full scanning failure*, up to $52\%$.

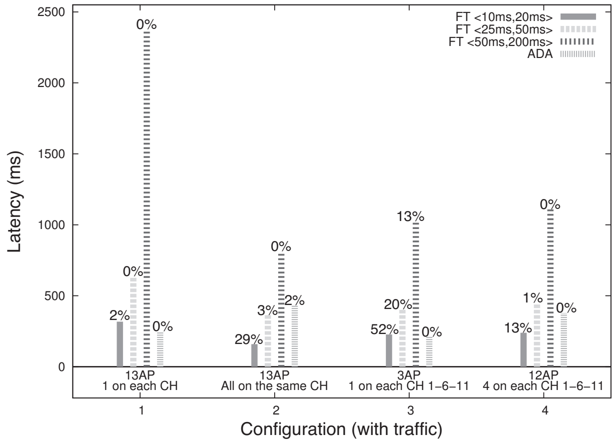

Figure 7: Testbed Results

## V. Conclusions

In this article, we have analyzed and evaluated the discovery process on 802.11 devices. This process has been extensively treated in the literature when it concerned the handover, i.e., when an MS needs to switch from one AP to another. Several optimizations were proposed in the literature that condition the *full scanning latency* and *full scanning failure* trade-off, highlighting the importance of the values of *MinCT* and *MaxCT*. In our first evaluation, we used a light weight C simulator to evaluate the influence of both timers for different probe response delays on the trade-off. We proposed two different strategies for the timers, one with fixed timers, and another one using adaptive timers. We observed that the adaptive strategy gives a better balance between the *full scanning latency* and the *full scanning failure* than the fixed timers strategy. The *full scanning latency* does not overshoot and the *full scanning failure* is always maintained below low limits using the adaptive timers strategy (9% of *full scanning failure* on the optimistic 4AP scenario with only 10% of received probe responses before $10ms$).

The fixed timers strategy shows high *full scanning failure* for long probe response delays. The second evaluation that we proposed in this article is an experimentation over a testbed with different APs and MSs. Results show that in almost all proposed scenarios, the adaptive strategy offers a better percentage of discovered APs, minimizes the number of *full scanning failure* (at maximum 2%), and keeps a low and controlled *full scanning latency* (between $190ms$ to $434ms$). As we have shown, each particular scenario may have different optimal timer values to achieve an optimal trade-off between *full scanning latency* and *full scanning failure*. This demonstrates the importance and the efficiency of using an adaptive strategy, since the user faces heterogeneous scenarios and the discovery process must dynamically adapt to them.

As a future work we plan to further investigate different adaptive functions, scanning policies and candidate AP selection algorithms. A detailed analysis of the adaptive algorithm parameters is currently being performed in order to obtain a unique set of parameters that optimizes the algorithm. As it was proposed in several optimization techniques, a selective scanning approach not only reduces the *full scanning latency*, but it also conditions the success of the handover process. Thus, we could apply an optimized channel switching policy, and interrupt the scanning process before all channels have been scanned.

## References

[1] U. Kozat and L. Tassiulas, *Network Layer Support for Service Discovery in Mobile Ad Hoc Networks*, Proceedings IEEE INFOCOM, San Francisco. April 2003.

[2] 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.

[3] F. Gonzalez, J. Perez and V. Zarate, *HAMS: Layer 2 Accurate Measurement Strategy in WLANs 802.11*, In Proceedings of the 1st IEEE International Workshop on Wireless Network Measurements (WiNMee 2005), 2005

[4] A. Mishra, M. Shin, and W. Arbaugh, *An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process*, SIGCOMM Comput. Commun. Rev. 33, 2, pp 93-102, Apr. 2003.

[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] V. Mhatre and K. Papagiannaki, *Using Smart Triggers for Improved User Performance in 802.11 Wireless Networks*, MobiSys06, Uppsala, Sweden, 2006.

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

[8] 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.

[9] 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.

[10] 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.

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

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

[13] J. Eriksson, H. Balakrishnan, and S. Madden, *Cabernet: vehicular content delivery using WiFi*. In Proceedings of the 14th ACM international Conference on Mobile Computing and Networking (MobiCom '08, San Francisco, USA) September, 2008.

[14] E, Giordano, R. Frank, G. Pau and M. Gerla, *CORNER: a realistic urban propagation model for VANET*, Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on, pp.57-60, 3-5 Feb. 2010

[15] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha and S. Banerjee, *Diagnosing Wireless Packet Losses in 802.11: Separating Collision from Weak Signal*, INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, 2008.