

Access Point Discovery in 802.11 Networks

Andrés Arcia-Moret*, Laudin Molina*, Nicolas Montavont[†], German Castignani[‡] and Alberto Blanc[†]

*Networking and Distributed Systems Group, University of Los Andes, {andres.arcia, laudin}@ula.ve

[†]Institut Mines Telecom/Telecom Bretagne, Irista, {name.lastname}@telecom-bretagne.eu

[‡]University of Luxembourg / SnT, german.castignani@uni.lu

Abstract—This paper analyzes the scanning process in IEEE 802.11 networks in an urban setting characterized by a high Access Point (AP) density. Most of these APs belong to a community network, known as a collection of APs announcing the same network name (Service Set Identifier or SSID). The owner of an AP can optionally configure the community network of his/her AP, resulting in an irregular topology for each community network as there is no central planning authority. We investigate the relationship between the time spent in each channel while scanning for available APs and the number of AP actually detected. In particular we show that, in order to discover all available APs at a given location, we need to combine the results of multiple scans. Based on this result we argue that the efficiency of the scanning process could be greatly improved by using a database shared by all the users of a community network, containing the available APs at different locations.

I. INTRODUCTION

Given the popularity and omnipresence of low-cost 802.11 networks operating in the free-licensed 2.4GHz band, it is common to find massive installations of IEEE 802.11 Access Points (AP) deployed by independent home users and organizations in a decentralized manner. This behaviour creates spontaneous deployments with unpredictable discovery patterns, *i.e.*, devices operating at different locations, having different performance in terms of hardware and software, and having WiFi specific settings [1]. Several authors have proposed different ways to share these APs among a certain subset of the users, a notion often described by the Community Network (CN) term [2]. Some of these solutions are already deployed, for instance, all the users of an Internet Service Provider (ISP) can access the APs of any other customer of the same ISP. Another solution is for users to join a community of other users (including those in other countries and using other ISPs) allowing mutual access to their APs (e.g., such is the case of FON [3]).

In any case, before accessing the network, a mobile station (MS) must search for available networks and assess their characteristics during the so-called scanning process. The IEEE 802.11 standard [4] describes two types of scanning, called passive and active. In the passive scanning an MS hops (and listens) over each channel, waiting for beacon frames that are periodically sent by APs. During the active scanning, an MS actively broadcasts management frames called Probe Requests over the different channels in a particular order, waiting for APs response frames (unicast Probe Responses) in the same channel. Note that an MS may also receive beacon frame while waiting for Probe Responses. The Probe Request - Probe Response exchange follows the Distributed Coordination Function (DCF), which controls the medium access by assigning a random backoff to each of the contending station. Active and

passive scanning are also confronted to the random variations in the wireless channel (signal degradation, fading, interference and noise). All this can lead to an unpredictable scanning behaviour, especially when the number of APs increases.

Modern devices hide the implementation of the scanning algorithms. To the best of our knowledge, PCs running Linux are the best alternative as a research platform since most of the scanning implementation code is open. Until recently, most of the IEEE 802.11 Medium Access Control were implemented within the hardware of the wireless card. Recent versions of the Linux kernel, instead, use a software implementation of MAC so that it is possible to modify several of the scanning parameters, by generating a customized `mac80211.ko` kernel module. We have implemented a mechanism that can accurately measure the response delay and that can also modify the Channel Time parameter (see [5] for more details).

In this paper, using the above mentioned implementation, we conducted a **measurement campaign** in a dense network scenario in the city of Rennes (France). The contribution of this paper is twofold. First, it shows the **dynamics of the scanning** in a crowded scenario. Counter-intuitively, as it seems to be generally understood (based on existing implementations), **we show that the scanning process may not be as efficient as supposed when using long values for the timers**. Nor is the opposite true, *i.e.*, short timers report only a partial view of the network. Second, we analyze the effect of the aggregated scanning on the AP discovery to finally come up with a discussion of a collaborative system where a centralized data base could gather several MS scanning results to build an accurate view of the network.

The remainder of the paper is organized as follows. In Section II we present previous work on the IEEE 802.11 network discovery process from two different perspectives, from the client and from the AP. In Section III we describe the methodology used for collecting the data and compare Linux and Android client platforms. In Section IV we highlight the main parameters that impact the AP discovery. In Section V we study the effects of repeated scanings and discuss about the need for a central system for assisted scanning. Finally, in Section VI we conclude the paper.

II. RELATED WORK

A. Scanning from the MS point of view

Several papers have analyzed the scanning process from the point of view of the MS and have identified it as the largest contribution to the handover or the association delay.

Some proposals consider a discontinued scanning from the MS in order be ahead of the handover event [6]. Authors

consider the MS having enough time to perform the handover in a make-before-break fashion. Other improvements on this technique consider dynamically adapting the timers so that the user perceives no disconnection [7]. Additionally, interruption phases rely on network buffering of information so as to minimize the impact of scanning induced cut-off of data transfers.

Another approach for scanning consists in scanning just selected channels based on previous experience of the MS [8]. For example, there may be a classification based on previous signal strength experienced by the MS. Caching information about the surrounding APs is also used by the MS and an association of neighboring APs. This approach although suggested by Shin et al. [8], it has been presented as a system using neighbour graphs in [9]. By using neighbor graphs the MS can compute the number of channels to scan and the timer values by exploring its directly connected nodes in the graph. The information on the graph can even be used for unicast Probe Requests as in [10] thereby causing less congestion.

As it has been pointed out by Raghavendra et al. [11], the crowded nature of existing IEEE 802.11 deployments produces several problems such as intermittent connectivity, low throughput and high loss. In particular, high loss in crowded topologies forces MS to trigger unnecessary handovers (including scanning and re-associations), even in absence of mobility. Authors point out the need for adaptive handovers [12] to mitigate this performance degradation in crowded topologies.

In [13] the authors have presented an extensive analysis of the scanning process in a controlled environment. Their results are threefold. First, they have identified the trade-offs between several metrics defining the scanning process: the full discovery latency, the failure rate and the discovery rate. Second, they have shown several proposals for improving the scanning process. Third, authors propose a simple adaptation method for ramping up from low timers to high timer values in order to increase the discovery rate and minimize the failure rate, compared to standard strategies.

Recently, in [14] authors have derived mathematical expressions for quantifying the discovery rate, failure rate and the full scanning latency. Moreover, using genetic algorithms over a large set of collected scanning traces, they have found an optimal channel sequence and corresponding timers for achieving high discovery rates and minimum failure rates in a Community Network scenario. Different from previous work, proposed timers adapt better in a mobile scenario considering thousands of APs.

B. Scanning from centralized point of view

As part of a current discussion at the Global Internet Access for All IETF group [2], by using state-of-the-art open technologies such as openflow [15], there is a possibility to implement centralized systems capable of managing several overlapping APs covering extended areas¹. Such centralized systems can even decide which AP may handle a particular client, for example, depending on the traffic profile. Sathiseelan et al. [16] proposed sharing home routers creating

¹This is the case of PAWS, conceived to crowd-share Internet access in a neighbourhood [16].

a parallel wireless networks (to existing infrastructure) that use LEDBAT transport protocol and DTN over WiFi. These parallel networks (as other CNs) considerably increase the number of active SSIDs in a region, therefore making the discovery of the topology a difficult task [14].

In [17] authors propose using a master-client architecture to centrally manage a group of APs in a region. The 802.11 Probe Requests are handled to the central entity which in turn commands an appropriate AP for serving certain connection request. This centralized decision both offers transparency to clients (as if they were in a regular 802.11 network) and alleviate the task of programmers by offering a *Light Virtual AP*. Although the scanning process may take longer, there is a better control on the quality of the offered point of attachment for an MS.

Virtual Public Network [18] (VPuN) is a similar proposal, for which there is a central entity called the Virtual Network Operator, capable of remotely managing APs serving the VPuN clients. In this approach authors propose simple ways of updating the VPuN scheduling of the APs through the introduction of a interpreter and easy-to-write rules which specify the sharing-time for public clients.

III. DISCOVERY EVALUATION METHODOLOGY

A. Methodology and tools

To collect the network deployment information we have designed and implemented a platform that is able to continuously scan for IEEE 802.11 networks while collecting beacons and Probe Response frames, together with the following network information: Service Set Identifier (SSID), Basic Service Set Identifier (BSSID), security mode, operating channel and signal strength. The main components of this platform are:

- Laptop: Hewlett-Packard, model nc-2400, operating system Debian GNU/Linux Jessie, with the modified `mac80211.ko` (see [5] for details on the configuration).
- NIC: Intel Corporation, model PRO/Wireless 3945ABG-Golan.
- iw tool, version 3.13 [19]
- Mobile: Smartphone Galaxy S3 (GT-I9300) running Android OS 4.3.1.

We used this platform to collect AP information in several locations in the downtown area of Rennes, France. Measurements were obtained on city streets and squares surrounded by mid-size buildings (4 to 10 floors).

During the tests we evaluated different Scanning Timers. The full campaign consisted in 900 scan trials executed at each location, 100 scan per Channel Time value. As depicted in Fig. 1, we triggered sequential scanning, separated by 2 s, accounting for approximately 45 minutes between the first and last trial.

When starting a scanning campaign, the topology is first unknown and progressively obtained through sequential scanings. As observed in our experiments, the more we scan the higher the number of APs discovered. Fig. 2 shows an example

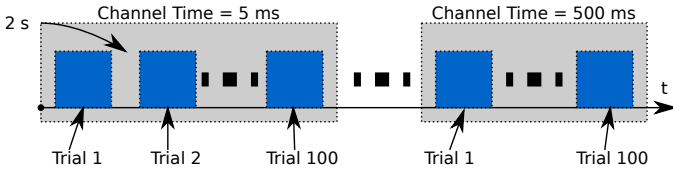


Fig. 1: Test Methodology

of the discovery dynamics. In scan 0, two APs are detected and constitutes the known topology at this time. In scan 1, an extra AP is discovered which decreases the discovery ratio that was observed in scan 0. In scan 2, only one AP already known is discovered. We call AP frequency the number of scanings in which an AP was discovered. In the example of Fig. 2, AP4 was detected in 66% of the times.

As we will describe in the rest of the paper, we have observed that the total number of APs may be fully discovered only after many scans (see Fig. 8c). Besides, two consecutive scanning results may have different set of APs.

B. Dependency on the platform

In order to draw better conclusions about the AP discovery, we made sure that observations were not platform dependent. While the scanning algorithm is different from one terminal (i.e., network card and its driver) to another [20], we compare the scanning results obtained with the Linux platform (see above) against an Android platform. For the Android client, we have used an application called Wi2Me [21] which periodically triggers a scanning and logs their results in a database.

Contrary to the Linux platform, we leaved untouched the default scanning behaviour in Android. We observed that the timer is around $250ms$ per channel, corresponding to a full scanning latency of 3.5 seconds, for the 13 channels. Once we got the scanning results, we launched a new scan after an inactivity timer of 3 seconds. Thus, a new scanning is launched approximately every 7 seconds. We have noticed that the Android scanning behaviour is more aggressive than the Linux one: in Android, several Probe Requests (4 on average) are sent in a channel, while in Linux a single Probe Request is sent on each channel. The aggressive sending of Probe Requests increases the chances for APs to receive them. Furthermore, this behaviour supports one of our claims in this paper: *for dense topologies, a single scanning is most likely to discover only part of the available APs*.

Fig. 3a and 3b show how many times each AP has been discovered after 100 scanings. The APs that were discovered in all scanings are illustrated in black. We can see in both figures that most of APs are intermittently discovered: only 12% and 20% of the available APs are discovered in all scanings for Android and Linux respectively. The total number of discovered APs is different between the two platforms: Linux is able to discover a total of 75 APs, while Android discovers 53 APs. We believe that this may be due to differences in hardware configuration (such as antenna, chipset, etc.) [20].

In Fig. 4a and 4b we can observe an approximate difference of 5 dBm favouring the received power on Linux versus Android. While we can observe that there is a poor correlation

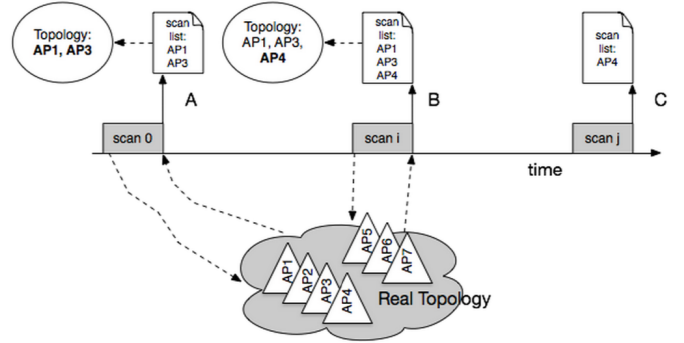


Fig. 2: Sequential Scannings Impact on the Discovery Rate

between the power of an AP and the number of times it is discovered, some APs have relative good power and still are discovered a few times, and some other APs have a bad power, and are often discovered. This is further discussed in Section IV-D. Fig. 4c finally shows the CDF of the number of discovered APs per scanning. For each platform, the X-axis is normalized with the total number of discovered APs. We can see that in Linux, the number of APs is higher, and that there is more APs per scanning (in median almost 90% of the APs while around 80% in Android). But these distributions are close. Also note that both systems almost reach the total number of APs in one scanning (for few scanings), which seems to indicate that the memory size may not be an issue. The Android curve is less exponential than the Linux one, meaning that there are more scanings with fewer APs in Android. This may be due to the platform performance regarding the radio reception. Globally, we can also say that the number of APs per scanning has a small variance.

IV. ANALYSIS OF SCANNING CHARACTERISTICS

In this section, we present a detailed analysis of the AP discovery and highlight the different parameters that impact the discovery. All the results presented in this section corresponds to the ones obtained with the Linux platform.

A. Timer impact

Fig. 5a, 5b and 5c show all discovered APs and its respective appearance frequency, when using timers of $5ms$, $20ms$, and $100ms$ respectively. In the abscise we represent the APs by a unique identifier and ordered by its appearance frequency. To isolate the effect of active scanning, in these figures we consider only Probe Responses (i.e., we leave out passive Beacons). As expected, we can observe that the higher the timer, the higher the total number of discovered APs. After 100 scanings with a $5ms$ timer, the client discovered 47 APs, with $20ms$ timer, 55 APs, and 67 APs with $100ms$ timer. As discussed in next section, we observe that when increasing the timer above $100ms$, there is no more gain on topology knowledge derived from Probe Responses. However, for the first time we observed, as in Fig. 5c, that 3 APs appeared in 100% of the scanings. In Fig. 5a, we can see that the most discovered AP has only been seen 72% of the (scanning) times.

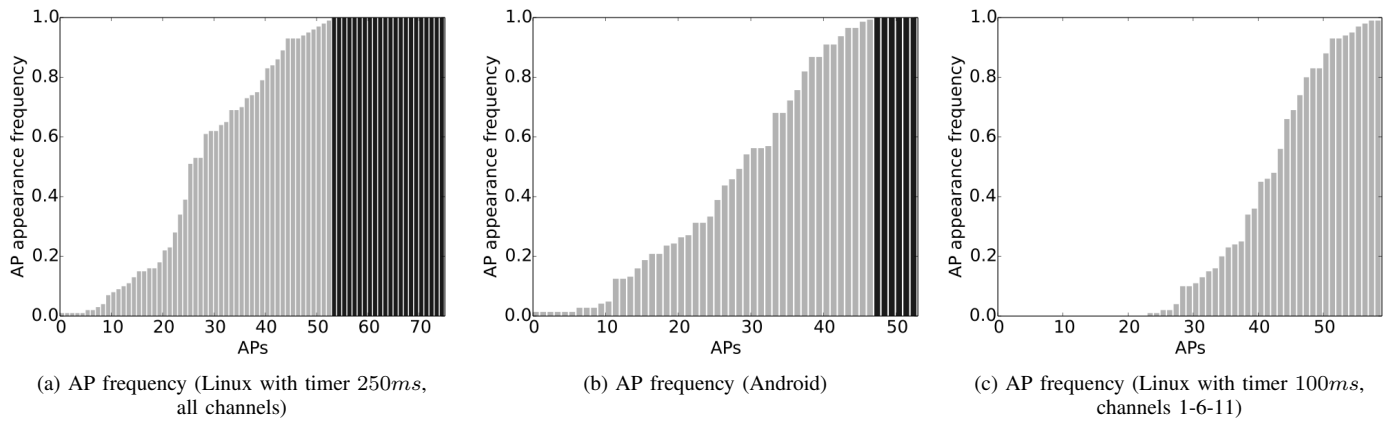


Fig. 3: AP Appearance Frequency - a comparison between Android and Linux. These figures show the proportion of scanings in which an AP has been seen. In the x-axis, we show each individual AP classified from the least detected to the most detected. In the y-axis, we present the proportion of scanning in which a given AP has been seen. For example, the AP indexed 50 in Fig. (a) has been seen in 97% of the scanings. The APs in black have been seen 100% of times.

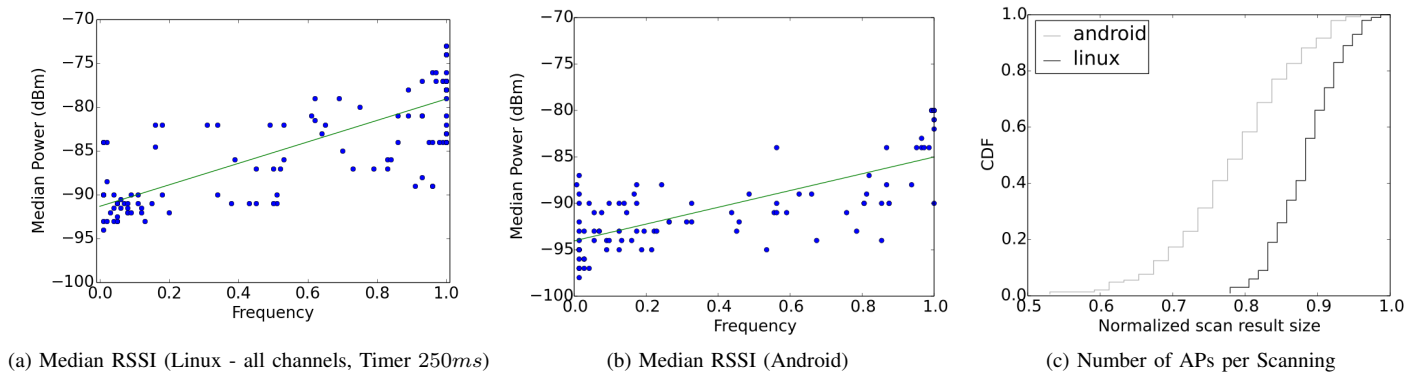


Fig. 4: RSSI and Number of APs per Scanning. Fig (a) and (b) show one RSSI value for each detected AP. When a given AP has been discovered several times (in multiple scanings), we calculated the median of all observed RSSI. In Fig (c) we show the CDF of the number of detected APs in each scanning. We normalized the number of APs with the total number of detected APs after all scanings. For example, if in one scanning the MS discovered 25 APs out of 30 total available APs, we account that this scanning instance had 83% of APs.

B. Probe Responses versus Beacons for Topology Discovery

A client discovers an AP either by receiving a Probe Response or a Beacon. Fig. 6 presents the frequency of received Beacons and Probe Responses for the longest measured timer of 500ms, for channels 1, 6 and 11. We clearly see that up to 100ms, the terminal is subject to receiving Probe Responses in response to its Probe Request. After 100ms, very few or no Probe Responses were received, while the number of received Beacons is high and relatively constant up to the expiration of the timer. Thus Beacons significantly contribute to the AP discovery when using large timers. Even when Beacon reception is decreased during the reception of Probe Responses, large timers (greater than 100ms) allow discovering a larger part of the topology trough passive scanning techniques (e.g., see Fig. 6).

C. Channel overlapping

It is well known that 802.11 b/g available channels are, by definition, overlapping at worst up to four adjacent channels.

However, channels 1, 6 and 11 are non-overlapping, for which several studies show that more than 75% of the APs are deployed on them [22]. Based on this observation, studies such as [14] or [8] propose to reduce the scanings to those (crowded) channels to reduce the scanning latency, while still obtaining a fair discovery ratio or having good chances to find APs. Despite the benefits of only using non-overlapping channels, Fig. 3c shows that the scanning performance is lower when scanning only those channels. We have noticed that when performing a scanning for all channels (like in Fig. 5c or 3a) the terminal discovers more APs, and a higher number of APs are present in all scanning trials. This is because, when scanning all channels, the client probes more often each channel due to the overlapping nature of channels. For example, scanning channel 2 probes the overlapping channel 1, therefore increasing the total number of APs found and coming from channel 1.

In Fig. 7, we present the ratio of APs that were discovered on overlapping channels. For example, when the client was in

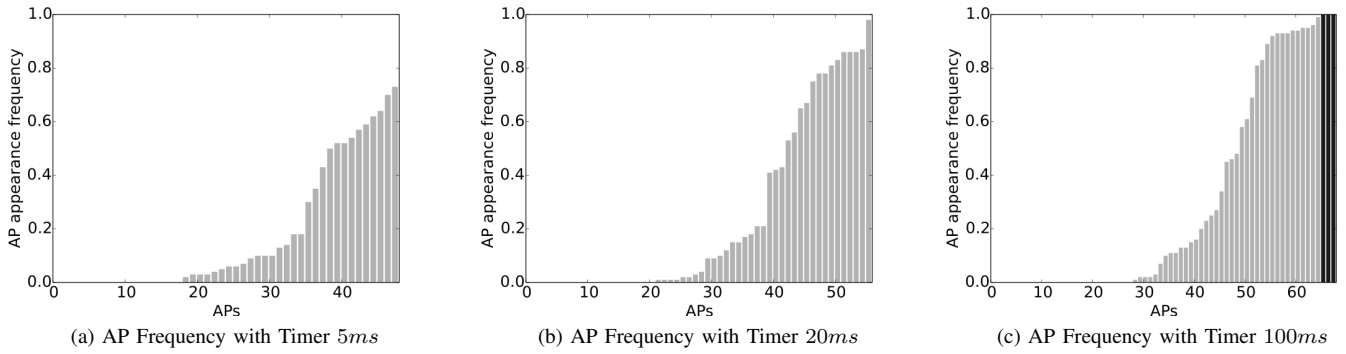


Fig. 5: AP Appearance Frequency depending on the Scanning Timers (Linux, scanning all channels)

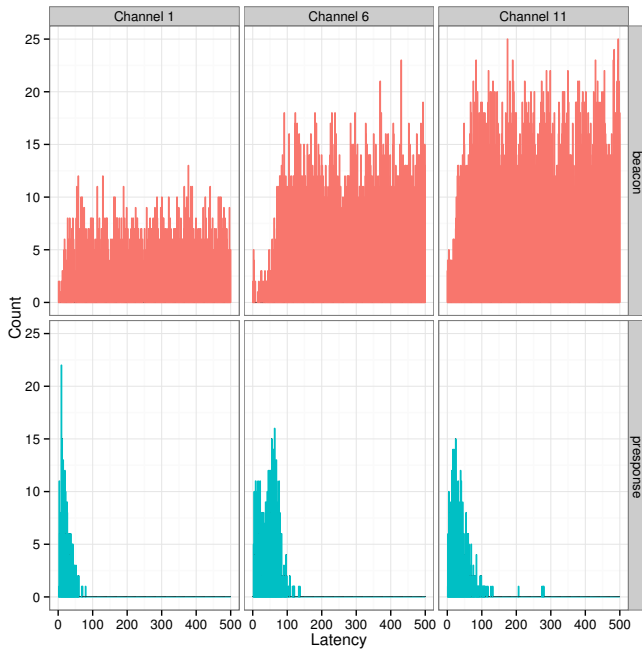


Fig. 6: Probe Responses and Beacons Reception Time (Timer 500ms)

channel 2, it discovered 40% of the APs operating on channel 1, 90% of the APs operating on channel 3 and 40% of the APs operating on channel 4. Note that some channels were empty, and channels 1, 6 and 11 are more populated than the others. Channels that are directly adjacent to crowded channels (channels 2, 5, 7, 10 and 12) usually discovery around 50% of the APs on channels 1, 6 and 11. Observe also that, in a given channel, an AP can be discovered even when operating in up to 3 distant channels.

D. On the quality of APs

The evaluation of the quality of APs is a complex issue, especially during the scanning phase. Aspects such as the variability of radio channels, the signal strength, the number of operating stations and the environment deeply impact the user quality of experience. When performing an active or passive

Ch. $i + 3$	0.1		0.2		0.1								
Ch. $i + 2$	0.6	0.4	0.4		0.3				0.7				
Ch. $i + 1$	0.7	0.9	0.8		0.6	1.0			0.8				
$i =$	1	2	3	4	5	6	7	8	9	10	11	12	13
Ch. $i - 1$		0.4	0.4		0.6	1.0				0.5			0.5
Ch. $i - 2$					0.6					0.2			0.5
Ch. $i - 3$					0.3								

Fig. 7: Percentage of APs detected on a different channel. Shaded values indicate the channel where the MS did the scanning. Values above (resp. below) the shaded row give the percentage of AP detected in upper (resp. lower) channels. For example, while scanning in channel 1, the MS discovered 70% of the APs operating in channel 2.

scanning, the client is able to measure the RSSI for each received frame from the APs. Fig. 4a and 4b show the median RSSI for each received frame from all discovered APs. Fig. 8a shows each RSSI samples from all discovered APs when scanning with timer 100ms. We can see that there is a general tendency which indicates that the stronger the signal, the higher the number of discovered APs. However, this observation is mitigated, as we have also seen that the signal strength varies from one sample to another, and we can see APs with good power signal that respond only a few times. Thus, there is no straightforward guarantee that all discovered APs (with good signal strength) will be able to provide data connection for the client. Interested reader can refer to [23] to see an in depth discussion of the subject.

Fig. 8b and Table I show the number of frames per AP per scanning. While an AP may not be discovered in all scanning attempts as explained up to now, we can also see that the client may have received multiple responses from the same AP in one scanning. On average, there are 3.73 frames from the same AP in one scanning with a maximum of 29 frames. The reason why multiple beacons are received from the same AP is that they are sent periodically. Regarding Probe Responses, there are retransmissions, either because an AP is configured to send multiple Probe Responses to answer a Probe Request,

TABLE I: Number of frames (Probe Resp. or Beacon) per AP per scanning

		10ms		50ms		250ms	
		Mean	Max	Mean	Max	Mean	Max
All (Probe Resp. + Beacon)	Min	0	0	0.13	3	1.23	11
	Mean	0.69	5.99	1.53	9.2	3.73	20.04
	Max	3.04	13	0.63	14	7.67	29
Beacons	Min	0	0	0.06	3	1.01	10
	Mean	0.36	4.12	0.87	7.54	3.11	15.95
	Max	1.89	9	2.84	12	6.33	25
Probe Resp.	Min	0	0	0.03	1	0.04	1
	Mean	0.34	2.18	0.66	3.71	0.62	5
	Max	1.81	9	2.54	12	2.19	15

or because the ACK from the client is not received by the APs, which causes retransmissions. The number of received frames may give a hint about the link quality with the AP. This metric combined with other metrics such as the RSSI and Probe Response delay may lead the client to better choose which APs to select in case of a handover.

V. UNDERSTANDING TOPOLOGY DISCOVERY

Fig. 8c shows the dynamics of the independent cumulative discovery rate for a series of 100 trials for every timer when accounting Probe Responses only. Observe that none of the plots reaches the discovery of 100% of the topology since we account the 100% as the whole set of discovered APs after the 800 trials. That is, even for the highest timer, there are some APs (up to 15%) that were not detected after 100 trials, but that have been detected while using other values for the Channel Time, suggesting that further scanning will increase the topology. Nevertheless, this could also be due to new APs appearing during short periods of time (e.g., tethering with smart-phones or public transportation APs).

Note that, after every new scanning, the discovered topology may increase, thus accounting only the differences for increasing the knowledge of the topology. See for example, as indicated by region A within Fig. 8c, how we gain a 4% of the topology after just a couple of scanning in the middle of the cumulative scanning process. This effect was found on trials 44 and 45, when detected a total of 7 and 6 new APs respectively, reporting 3 new APs during trial numbered 45. We observe an even more dramatic result for region B, the 20ms curve around trial numbered 60, in which we gain 10.6% of the topology after just 4 scanning when more than half of the discovery process has taken place.

In general, we observe that the discovery ratio after the trial 100 reflects the timer length: shorter timers allow discovering less APs and vice-versa. However, we observe the case for 50ms and 100ms in which counter-intuitively, the former discovered more topology than the later. We believe that this may be because some APs appear only during short periods of time, or because of the hidden terminal problem in which many stations are negatively interfering with each other, therefore increasing the delay of the response or the probe losses. Yet another reason could be that the intrinsic diversity in the implementation of the 802.11 devices [20]. As reported in other papers, DIFS and SIFS may be different, RTS/CTS

mechanism may be activated just in some of the APs and so on.

A. On the need for a Central Database

Given the results presented in this paper, we argue that an open data system could opportunistically assist mobile users to better discover and control the Community Network topology and determine their link quality. As we saw, within a single scanning, only a subset of the available APs are discovered, and usually a client does not have the time to scan multiple times. However, with a centralized database system where users could share their information about their vision of the topology, offline servers could compute up-to-date map of the topology, proposing a global and more precise view of the AP deployment. Users would act as both data producer and consumer: user would push their vision of the topology from time to time, with a simple post on central servers. In situation of handover or simply when looking for a candidate AP, users could retrieve this information about the topology by sending a query to the central server. Information maintained by the central server can be consolidated by each new user committing data, and thus providing users with accurate information about the topology. A simple round trip between the server and the client could save several hundred of milliseconds of scanning.

VI. CONCLUSIONS

In this paper we have analysed the dynamics of the topology discovery for crowded 802.11 Community Networks. We have observed that for different client devices, when the number of APs is high, a single scanning procedure is not enough for discovering the available topology. Moreover, we have presented the differences for topology recognition between active and passive scanning. Although the former contribute to informing the highest portion of the topology, the later may be conveniently used for complementing the knowledge of the topology. We have also observed that Active Scanning can take advantage of overlapping channels by increasing the total number of reported APs per scanning.

Finally, obtained results suggest that the information of the available topology can be stored in a central data base, and be shared in turn with other users for minimizing its discovery time. This observation is subject of on-going work. At this time, we are developing and testing a central system and related protocols that help in storing a crowded topology and conveniently informing topology-agnostic clients.

ACKNOWLEDGMENT

The authors would like to thank Tanguy Kerdoncuff and Nicolas Kuhn for their assistance and valuable comments on this paper. This project was partially funded by CDCHA of University of Los Andes under the code I-1369-13-02-B.

REFERENCES

- [1] A. D. Stefano, G. Terrazzino, L. Scalia, I. Tinnirello, G. Bianchi, and C. Giaconia, "On the anomalous behavior of IEEE 802.11 commercial cards," in *The IFIP Fifth Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC)*, Lipari, Italy, June 2006.

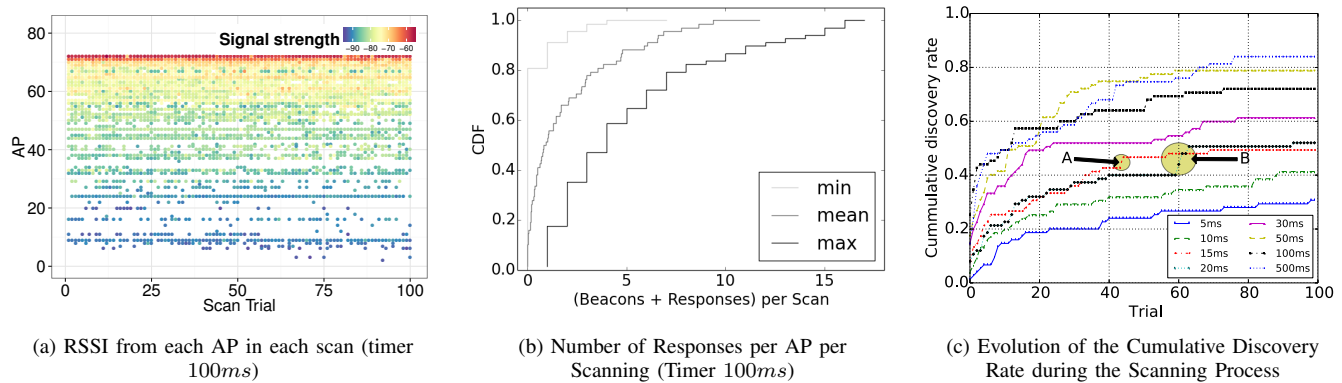


Fig. 8: RSSI, Number of Response per AP per Scanning and Discovery Rate

- [2] J. Saldana, A. Arcia-Moret, B. Braem, L. Navarro, E. Pietrosemoli, C. Rey-Moreno, A. Sathiseelan, and M. Zennaro, "Community Networks. Definition and Taxonomy." draft-manyfolks-gaia-community-networks-00, June 2014. [Online]. Available: <http://tools.ietf.org/html/draft-manyfolks-gaia-community-networks-00>
- [3] "Your global wifi network. <https://corp.fon.com/>" [Online]. Available: <https://corp.fon.com/>
- [4] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," <http://standards.ieee.org/about/get/802/802.11.html>, IEEE, 2007.
- [5] A. Arcia-Moret, L. Molina, G. Castignani, and N. Montavont, "Characterizing spontaneous IEEE 802.11 network deployments," in *Network Games, Control and Optimization (NetGCoP), 2012 6th International Conference on*, Nov 2012, pp. 1–8.
- [6] Y. Liao and L. Gao, "Practical schemes for smooth MAC layer handoff in 802.11 wireless networks," in *Proc. 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks*, ser. WOWMOM '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 181–190. [Online]. Available: <http://dx.doi.org/10.1109/WOWMOM.2006.90>
- [7] J.-W. Nah, S.-M. Chun, S. Wang, and J.-T. Park, "Adaptive handover method with application-awareness for multimedia streaming service in wireless LAN," in *Proc. 23rd international conference on Information Networking*, ser. ICOIN09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1–7. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1699392.1699393>
- [8] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs," in *Proc. second international workshop on Mobility management & wireless access protocols*, ser. MobiWac '04. New York, NY, USA: ACM, 2004, pp. 19–26. [Online]. Available: <http://doi.acm.org/10.1145/1023783.1023788>
- [9] M. Shin, A. Mishra, and W. A. Arbaugh, "Improving the latency of 802.11 hand-offs using neighbor graphs," in *Proc. 2nd international conference on Mobile systems, applications, and services*, ser. MobiSys '04. New York, NY, USA: ACM, 2004, pp. 70–83.
- [10] S.-H. Park, H.-S. Kim, C.-S. Park, J.-W. Kim, and S.-J. Ko, "Selective channel scanning for fast handoff in wireless LAN using neighbor graph," in *Proc. Personal Wireless Communications*, ser. Lecture Notes in Computer Science, I. Niemegeers and S. de Groot, Eds. Springer Berlin / Heidelberg, 2004, vol. 3260, pp. 629–629. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30199-8_16
- [11] R. Raghavendra, E. M. Belding, K. Papagiannaki, and K. C. Almeroth, "Understanding handoffs in large IEEE 802.11 wireless networks," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 333–338. [Online]. Available: <http://doi.acm.org/10.1145/1298306.1298353>
- [12] V. Mhatre and K. Papagiannaki, "Using smart triggers for improved user performance in 802.11 wireless networks," in *Proc. 4th international conference on Mobile systems, applications and services*, ser. MobiSys '06. New York, NY, USA: ACM, 2006, pp. 246–259.
- [13] G. Castignani, A. Arcia, and N. Montavont, "A study of the discovery process in 802.11 networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 15, pp. 25–36, Mar. 2011. [Online]. Available: <http://doi.acm.org/10.1145/1978622.1978626>
- [14] N. Montavont, A. Arcia-Moret, and G. Castignani, "On the selection of scanning parameters in IEEE 802.11 networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, 2013, pp. 2137–2141.
- [15] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: <http://doi.acm.org/10.1145/1355734.1355746>
- [16] A. Sathiseelan, J. Crowcroft, M. Goulden, C. Greiffenhagen, R. Mortier, G. Fairhurst, and D. McAuley, "PAWS: Public Access WiFi Service," Digital Economy All Hands Meeting, 2012.
- [17] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise w lans with odin," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 115–120. [Online]. Available: <http://doi.acm.org/10.1145/2342441.2342465>
- [18] A. Sathiseelan, C. Rotsos, S. C. S., D. Trossen, P. Papadimitriou, and J. Crowcroft, "Virtual Public Networks," in *2nd IEEE European Workshop on Software Defined Networking (EWSDN)*, Berlin, 2013, pp. 1–6.
- [19] "Linux wireless developer documentation." [Online]. Available: <https://www.kernel.org/pub/software/network/iwl/>
- [20] A. Di Stefano, A. Scaglione, G. Terrazzino, I. Tinnirello, V. Ammirata, L. Scalia, G. Bianchi, and C. Giaconia, "On the fidelity of IEEE 802.11 commercial cards," in *Proc. First International Conference on Wireless Internet*, ser. WICON '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 10–17.
- [21] G. Castignani, A. M. Lampropoulos, A. Blanc, and N. Montavont, "Wi2Me: A Mobile Sensing Platform for Wireless Heterogeneous Networks," in *ICDCS 2012: IEEE International Workshop on Sensing, Networking, and Computing with Smartphones*, 2012.
- [22] G. Castignani, A. Blanc, A. Lampropoulos, and N. Montavont, "Urban 802.11 community networks for mobile users: Current deployments and perspectives," in *Proc. Mobile Networks and Applications*, Aug. 2012.
- [23] P. Fuxjager, D. Valerio, and F. Ricciato, "The myth of non-overlapping channels: interference measurements in IEEE 802.11," in *Proc. Fourth Annual Conference on Demand Network Systems and Services (WONS'07)*, 2007, pp. 1–8.