Intelligent Network Discovery for Next Generation Community Wireless Networks

Andrés Arcia-Moret*, Antonio Araujo †, José Aguilar‡, Laudin Molina§, Arjuna Sathiaseelan*
*Computer Laboratory, University of Cambridge, Cambridge, UK
†Centro Nacional de Desarrollo e Investigación en Tecnologías Libres, Mérida, Venezuela
‡Universidad de Los Andes, Mérida, Venezuela
§Institut Telecom / Telecom Bretagne, Université Européene de Bretagne, Rennes, France
Email: {andres.arcia, arjuna.sathiaseelan}@cl.cam.ac.uk, aaraujo@cenditel.gob.ve, aguilar@ula.ve, laudin@telecom-bretagne.eu

***Abstract*—Nowadays, IEEE 802.11 networks are the most popular option to have wireless access to the Internet and a promising technology to tackle the digital divide that accounts for 2/3 of the world population. However, the popularity of these networks have raised a complex discovery and connection process, i.e., any device has to pass through an expensive scanning process of available Access Points in crowded and chaotic deployments. The scanning process can be modelled by a set of metrics exposing a trade-off between latency and the discovery rate when searching for appropriate Wi-Fi connection. Consequently, in order to improve the connection process, we use a multi-objective optimisation approach for generating optimal scanning sequences. We propose a framework to assist the network discovery within a Community Network, and we have adapted a Cultural Algorithm as an intelligent component for calculating optimal scanning sequences. Results show that we can derive optimal scanning sequences better than standard approaches for scanning in chaotic network deployments.**

I. Introduction

Today, IEEE 802.11 networks are the first option to have ubiquitous and low-cost wireless access to the Internet. We count on millions of Wi-Fi devices in many different new and challenging contexts: Community Networks [1], Sensor Networks [2], Internet of Things, and personal computers in Home Networks. In all of these scenarios, there has been a dramatic increase on the number of devices requiring to connect to Wi-Fi Access Points (AP). In order to connect to the wireless access network, a device must scan its surrounding and find an appropriate AP. However, the scanning process embedded in mobile and desktop devices does not follow a standard pattern or design principle.

In Community Networks, people spontaneously deploy APs. Nomadic users are able to access thousand of community APs belonging to the same direct provider [3], and also have access to virtual network operators (VNOs) such as FON[1]. In both cases a user has to pass through a scanning process, being the most expensive sub-process within the discovery of a wireless topology. In this respect, the discovery process is becoming iterative and time consuming in Wi-Fi dense deployments. Recently, we have observed in [4] that in a regular discovery process, the device has to scan multiple times to discover available APs in a densely-covered urban area. World-wide trends show that this is likely to be a regular case in the near future[2].

The current suboptimal behaviour of the scanning algorithm is present in the vast majority of devices, significantly consuming energy and impacting network performance [5]. The scanning traffic is becoming a potential problem lowering the available bandwidth and frequently interrupting regular transmissions. This is mostly because of the increasing number of devices using Wi-Fi and the congestion induced by non-adapted scanning process [4]–[6].

On the other hand, a Mobile Station (MS) may also perform a regular scanning by testing a partial or a complete set of channels. Approaches for partial scanning show that small bursts of scans allow uninterrupted execution of applications and look forward to perform seamless handovers. Whereas regular complete-set approaches for modern devices (iOS or Android based) perform sequential scanning without any adaptation on the sequence nor the duration of the timers. Considering that the later is most common scanning strategy [7], and that the nature of the scanning process represents about 80% of the handover time [8], an efficient scanning will not only represent an improvement on aggregated control traffic reduction for public Wi-Fi access [5], but also – if specific scanning sequences are given to specific group of users (§ III)– performing load-balancing in systems like VPuN [1] or Meraki[3].

A. Scanning process in 802.11 networks

The scanning is the first (and most time consuming) sub-process for a client willing to attach to an IEEE 802.11 network, in which a client interface looks for available APs for later associate to them [9]. Although the ultimate (and common) goal of a scanning is to find all available APs to which the station might be able to join, it is very costly in terms of aggregated number of beacons and energy at the client. To discover all APs, i.e., the topology within a dense area, the device should properly be adjusted with a pair of timers, namely *MinChannelTime* and *MaxChannelTime*, referred as *MinCT* and *MaxCT* respectively from now on [10], [11].

IEEE 802.11 standard [9] defines two kinds of scanning

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

[2] https://wigle.net/

[3] https://meraki.cisco.com/

ISBN 978-3-901882-80-7 © 2016 IFIP

procedures: active and passive scanning[4]. As shown in Fig. 1, during an *Active Scanning* the client starts a timer called *MinCT* to wait for responses on a single channel, then it broadcasts a Probe Request frame waiting for Probe Responses. If it receives no Probe Responses before *MinCT*, then it passes onto the next channel of the scanning list. Otherwise, if at least one Probe Response is received, it will trigger an extended timer called *MaxCT*. After it expires, all Probe Responses are processed before jumping onto the next channel.

An efficient scanning approach should rely on appropriate and independent pair of timers per channel. However, as suggested by Fig. 1, a scanning function should not underestimate the time spent on a given channel (e.g., as in *ch11*) nor overestimate it (e.g., as in *ch2*) to discover an appropriate number of APs. As we discuss in this article, *the efficient estimation of timers depends on the knowledge of the network topology, hence the need for an assisted scanning.*

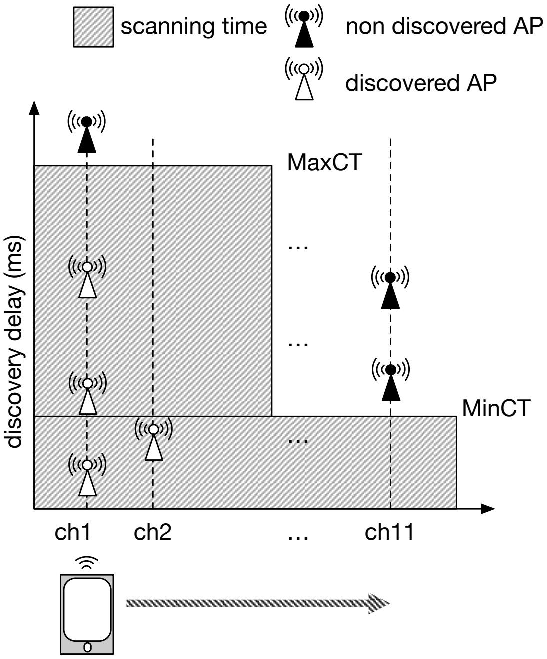

Figure 1. Active Scanning in IEEE 802.11 Networks

There is a high number of possibilities for timer configuration. There are 22 timers in total, given 11 channels and 2 timers per channel. Adapting these timers could depend not only on the topology, but also on the application requirements [12]. However, from a general perspective, an appropriate discovery process depends on the scale and the distribution of APs within the wireless network. As the complexity of finding an appropriate scanning sequence increases, we propose the use of a computational intelligent technique for improving the scanning process.

B. Using computational intelligence for assisting Wi-Fi scanning

In this work we propose a Topology Manager that hosts an intelligent scanning process, and uses a computational intelligence (CI) technique to calculate efficient scanning sequences that improve the Wi-Fi discovery process. As suggested in [1], [4], [13], in future dense (e.g., community) Wi-Fi deployments, these central entities or *topology managers* can help wireless clients in the decision process for efficiently connecting to a dense Wi-Fi network, moreover they could also serve for coarse content or service indication by means of the 802.11u access network query protocol.

In the CI field, there are several multi-objectives optimisation techniques that mimic the nature evolutionary principles that are used to execute search and optimisation procedures. Some of these techniques are based on theories proposed by sociologist to model cultural evolution. In our particular case, we adapted and implemented a Cultural Algorithm (CA) in order to deal with multi-objective optimisation problems [14]. Our version of the CA is based upon evolutionary programming, Pareto ranking and elitism (i.e., individuals finely filtered) to accomplish a more efficient search for optimal scanning sequences.

In this work we present a novel approach to find optimal scanning configurations using an assisted scanning approach [15]. As the scanning problem can be stated as a multi-objective optimisation problem [12], we use CA to produce topology-adapted scanning sequences. We discuss a new trade-off (see Section IV) balancing the discovery rate (in number of discovered APs per time unit) and total latency for the discovery process. Furthermore, we contextualise the use of an intelligent algorithm within a Topology Manager (Section III).

Our results are **based on real measurements and evaluated through an emulation model** [4] described in § V. Upon this, the CA generates a set of sequences along the Pareto front that can be used depending on the user needs. We compared with scanning sequences that have been reversed engineered from various devices and we show advantages of our approach.

The remainder of this paper is organized as follows. In Section II we present the related work on the different scanning approaches and discusses techniques for scanning optimization. In Section III we describe the Topology Manager for assisting the network discovery process. In Section IV we present the design and implementation of the Cultural Algorithm. In Section V we discuss the results of the emulations. Finally, in Section VI we conclude the paper and give the perspective for the future work.

II. Related Work

The scanning process has been assessed and optimised considering two main perspectives. Firstly, some authors present different grouping strategies to empirically find an optimal sequence. These findings are supported with controlled experimental testbeds. Secondly, other authors discuss methods for calculating adaptive sequences considering an unknown number or high number of APs in the deployment, such as those found in Community Networks.

[4] In *passive scanning*, the station just listens in each channel an amount of time defined by *MaxChannelTime* parameter. The interested reader can refer to [11] for further details on the discovery process.

The optimization of the scanning sequences has been systematically addressed by means of empirical approaches, i.e., by modifying the timers according to a controlled topology. The first to propose a modification to timers were Velayos and Karlson [16], who proposed fixed timers based on the theoretical best and estimated worse case for sending and receiving probes. Other authors have also followed this proposal increasing the *MinCT* arguing that even for minimal congestion on channels with few APs, the theoretical value is inappropriate. Another approach extensively worked by Castignani et al. [17] consisted in adapting the timers depending on the load of the network, i.e., efficiently adapting timers while performing the scanning process. Authors have also studied the impact of cross-layer information for setting higher timers where pre-sensed energy thresholds were detected [18].

There are other similar approaches such as grouping channels to scan each group in a time-shifted fashion. This allows the alternation between the sending of probes and data traffic to produce minimal impact on continuous transfers. Montavont et al. [19] use this approach using two fixed and independent periods. Liao and Cao [20] use a smooth scanning technique that introduces variable scanning periods. The main purpose of this approach is to minimize the handover time. Nah et al. [21] use the same principle of alternate scan-bursts to improve the user experience.

Other approaches intend to minimize the scanning time by reducing the scanning sequence. Shin et al. [22] propose the scanning in non overlapping channels 1, 6, and 11. However, they do not address the adaptation to the topology. Eriksson et al. [23] propose to progressively calculate the probability of certain AP operating in a channel, and then calculate the scanning sequence.

On the other hand, Castignani et al. [11] have identified three metrics from which a scanning process can be modelled: the elapsed time for scanning the whole set of channels (namely *full scanning latency*), the *failure rate* as the probability of finding zero APs after completing the full scanning, and the *discovery rate* as the fraction of discovered APs over the total number of available APs which, in turn, could also be interpreted as the total number of discovered APs after a single full scanning. Note that authors rely on a complete knowledge of the (high scale) topology in order to calculate the discovery rate.

In Montavont et al. [12] we have used genetic algorithms (GA) considering the above mentioned metrics to obtain efficient scanning sequences. Sequences generated by GA are used to minimize the full scanning latency reducing the impact of discovery on applications and, at the same time, maximize the number of discovered APs (e.g., to have more alternatives for connection). Differently, in this work, we focus on a discovery rate (APs per time unit) for obtaining better fast scanning sequences favouring real-time applications and fast connection procedures.

Finally, some work has been done on central entities for assisting wireless networks. Alternatives such as Meraki, Behop project [24], VPuN [1] or PAWS [13] offer centralised services to control wireless network deployments but focus on traffic profiling and do not offer details about the discovery process.

III. A Topology Manager for IEEE 802.11 networks

As in an Information Centric Networking architecture [25], a Topology Manager (TM) conveniently hosted by the wireless service provider, could opportunistically assist mobile users to better discover and control a crowded wireless network topology. It could also help mobile users to determine link quality and best possible connection allowing improved access to the network. As we have observed, within a single scanning, a subset of the available APs are discovered, and usually, a client does not have time to scan multiple times [4]. However, with an intelligent TM, several community users could share their partial vision of the topology so that up-to-date and customized efficient sequences are computed by the TM, thus giving a global and more precise view of the access network. The main advantage of this approach is that the TM is able to generate efficient scanning sequences, that allow clients saving time during the expensive discovery process.

In order to update the vision of the topology on the TM, we rely upon two roles for participants. Firstly, new users would act as *feeders*, as they push their vision of the topology from time to time with a simple posts of their partial vision of the network (e.g., through regular non optimized scannings) on the TM. This update could be pushed using the IEEE 802.11u amendment also known as Access Network Query Protocol (ANQP). ANQP allows clients to query or to pass information to the TM behind designated APs and before authentication. Moreover, a client could use specific messages for finding out about a specific VNO whose network is accessible through a designated AP. Secondly, regular clients (RC) correspond to those feeders that have already contributed enough to the vision of the topology and that the TM has promoted to RC. Hence, RCs looking for a candidate AP could retrieve efficient sequences with special queries to the TM who, in turn, asynchronously interacts with an intelligent algorithm as the one proposed in § IV. Specific use-cases in which a RC benefits from this scheme, correspond to handovers or when needing to real-time applications. As shown in Fig. 2, a simple round trip between the RC and the TM, or even by retrieving the appropriate scanning sequence through an alternative interface (e.g., GSM or 3G), could save several hundred of milliseconds relying on the increased efficiency of the scanning.

A. Interaction with the Community Topology Manager

Fig. 2 illustrates the workings of the assisted scanning. A *feeder* (F) arriving to the network contributes with a raw scanning of the topology, from which the TM benefits with every new entry. The TM is able to store information about geo-tagged fingerprints created from previous feeders. Eventually, the TM will have enough information about the inter-response times of the APs on the topology under control. Meanwhile, this information is being stored at the *topology model* and being used by the intelligent algorithm to compute *smart sequences*.

A *feeder* is expected to do regular raw scannings during first attempts to join the network. Also contributing to the TM vision will allow the *feeder* to be promoted to RC, in order to be able to obtain efficient scanning sequences. Otherwise,

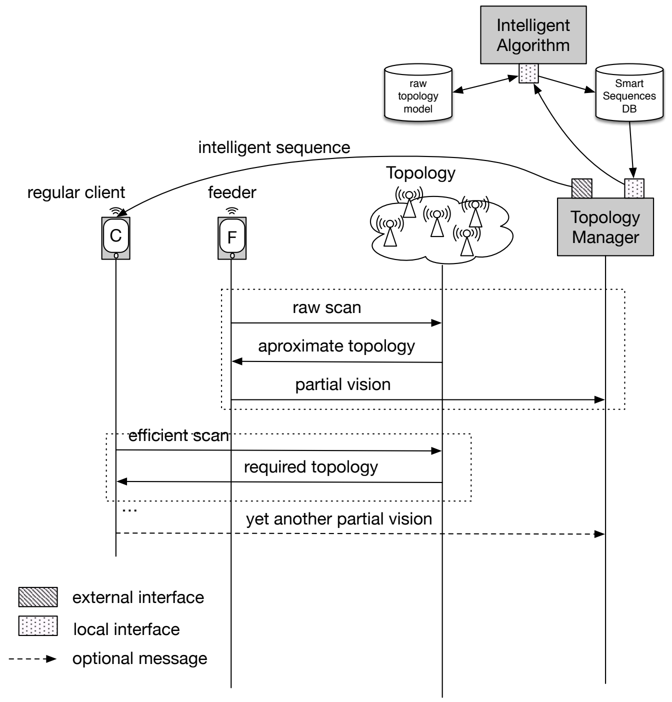

Figure 2. Architecture for the Next Generation Wireless

a non-feeder is be considered as non-cooperative client and would not profit from the TM estimations. In future subsequent scannings (either raw or efficient) a client is able to know whether it has a new fingerprint of its surrounding topology (e.g., if it is in a different location), and then it could send new fingerprints to the TM to improve the topology model. Finally, a regular client is likely to be interested in requesting to the TM a special scanning sequence based on the requirements of the applications it is running, e.g., a client imposes latency restrictions because it is running a VoIP application.

IV. An Intelligent Module for Topology Management

As shown in Fig. 2, the TM uses an intelligent algorithm to manage an efficient network discovery. The TM asynchronously and conveniently invokes the calculation of smart scanning sequences through the CA. These sequences are systematically stored into the so-called *smart sequences DB* (or elite DB in the CA context) so that the TM can have immediate access for efficient sequences whenever requested by a client.

The CA is a dual inheritance system with evolution considering two main components for calculating approximate solutions for a multi-objective problem [14]. The method relies on an initial population **P** and further refinements directions called beliefs. These components impact the creation of new generations of better individuals. The CA algorithm also allows directed mutation through which we can produce better intermediate individuals that bring better convergence to non-dominated individuals.

Fig. 3 is a macro representation of the CA. The approach relies on two big processes, the operations transforming the

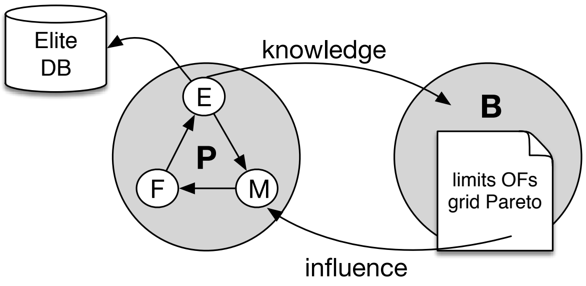

Figure 3. A general perspective of the Cultural Algorithm

population **P**, and the adjustment of the belief space **B**. An initial population **P** passes through several cycles of (M)utation, then (F)iltering of best performing individuals and finally an (E)xtraction process of the non-dominated individuals going into the elite-DB. These non-dominated individuals contribute to the knowledge, within the belief space **B**, about the limits of the studied variables and the exploration of a broader Pareto front. Consequently **B** influence subsequent (M)utation processes for future generations.

Algorithm 1: The Cultural Algorithm

```
   Data: wireless topology model
   Result: adapted (individuals) scanning sequences
1  Initialise with P random individuals;
2  Compute $OF_1$ and $OF_2$ for every individual;
3  Compute Super_Individual;
4  Initialise belief space;
5  while $i \leq Total\_Generations$ do
6      Mutate $P$ individuals into $2P$ population ($P_{new}$);
7      Filter $P_{best}$ individuals through tournaments on $P_{new}$;
8      Add non-dominated individuals ($P_{nd}$) of $P_{best}$ into elite-DB;
9      Update grid using $P_{nd}$;
10     Update phenotypical component every $M < Total\_Generations$;
11     Update Super_Individual;
12 end
```

Table 1 presents the pseudo-code of the CA. In line 1 we define the initial population **P** to be mutated and eventually will end-up in smart scanning sequences. The initial number of individuals is set according to [14], however based on our experimental results with the model, we have observed that we could successfully reduce the number of total iterations due to the convenient tuning of the initial population. Then in line 2 it is evaluated the performance of every individual of **P** by computing the Objective Function OF_1 and OF_2 for each of them. In line 3 we calculate a super individual which is going to influence specific number of genes during the mutation process. Then in line 4 we set up the belief space **B**, i.e., we determine the minimal and maximal values for OF_1 and OF_2, and we create a grid for bounding individuals so that grouped solutions do influence the initial population during the mutation process. The belief space **B** makes an impact through the adaptation of the limits of the ranges for each objective function. We have modified the Influencing process by incorporating a new mutation operator to guide the population to have equal chances of exploring the Pareto optimal front.

After the initialization process, subsequent refinement of the population start at line 5. The refinement starts with a mutation process (line 6) to obtain $2P$ individuals, for then filtering of the best individuals (P) through a tournament-like procedure (line 7). Afterwards, it follows the calculation of the non-dominated individuals (line 8) that help on the re-adjustment of the grid. Finally, the iteration process for refining the current individuals consist on making a new generation.

After a predefined number of generation, resulting individuals are the input to decision maker that actually adapts further the scanning sequence to the application needs.

A. Optimisation Model

The problem of optimizing the performance parameters of the scanning in IEEE 802.11 networks is modelled as follows:

$$\max \sum_{i=1}^{11} Nmin_{C_i}/MinCT_{C_i} + Nmax_{C_i}/MaxCT_{C_i} \quad (1)$$

$$\min L = \sum_{i=1}^{n} (MinCT_{C_i} + p_i \cdot MaxCT_{C_i}) \forall C_i \in [1, 11] \quad (2)$$

We have built a model based on the inter Probe Response time based on an extensive measurement campaign [4]. After analysis of several thousand of scanning traces, we imposed the following restrictions on the timers: $5 \leq MinCT \leq 15$, $3 \leq MaxCT \leq 90$.

$Nmin_{C_i}$ and $Nmax_{C_i}$ correspond to the number of APs discovered in channel i, with $MinCT$ and $MaxCT$ respectively. Hence, Eq. 1 corresponds to an index on the total discovery rate. The interval values for *MinCT* and *MaxCT* are expressed in milliseconds, and were obtained considering percentiles 20 and 80 respectively [4], [19]. L corresponds to the aggregated latency, expressed in milliseconds (ms). Within the latency, p_i correspond to the probability of having at least one response within any channel. This probability is obtained per channel, considering that (1) every channel has different effect on overlapping APs and (2) the time between Probe Responses varies according to the total number of APs living in a single channel. C_i represents the i channel from a total of 11 channels in the 2.4 GHz band for the crowded AP deployment [4]. Finally, our min/max model specified in Eq. 1 and 2, were used by the CA as objective functions.

Structure of the population. The population of a CA is conceived as individuals that represent candidate solutions and their characteristics are translated into an OF. In order to perform the evolutionary search we propose an individual with the following structure: $< X_1, ..., X_{11} >< OF_1, OF_2 >$. In which each subgroup X_i is conformed as $< C_i, Min_i, Max_i, AP_i >$ with $i = 1, ..., 11$, where: C_i is the channel numbered i, $MinCT_i$ is *Min Channel Time* (ms) for channel i, $MaxCT_i$ is *Max Channel Time* (ms) for channel i, AP_i is the number of discovered access points in channel i, OF_1 is the value that maximizes the discovery rate of APs, and OF_2 is the value that minimizes the latency. An individual will have a total of 44 parameters and two values of objective functions (value of discovery of APs and latency value).

Mutation. The main advantage our version of the CA consists in directing the mutation process. The mutation is performed to the parameters x_i $(i = 1, ..., 44)$ mapping into every timer of the P individual of the population. The Gaussian mutation is used to obtain the mutated value of a variable x_i of an individual as follows: $x_i' = x_i + N(m, \sigma)$. $N(m, \sigma)$ is a random variable with a normal distribution with mean m and standard deviation σ. In our particular case, we considered $m = 0$. Observe that this directed mutation allows confined and gradual changes to individuals.

The guided mutation. As we have previously described, the execution of the process of Mutation is based on the normative knowledge within the belief space **B** (see Fig. 3). For better results we have further defined executing a guided mutation (GM) with probability $P_{GM} = 0.3$. The goal of a randomly applied GM is to guide new individuals to regions in the solution space that have been little evaluated, and do exercise influence on younger generations. The GM finds cells of the grid within (B) with the smallest number of individuals but in which there are at least two non-dominated individuals.

The GM process uses a model individual built from the best genes of P in order to modify similar genes of new individuals on the knowledge base, stored in the grid of the belief space.

The tournament selection. This process is carried out by considering the population with size $2P$. Each individual will face C confrontations, chosen at random from the main population. There are two basic tournament rules. Firstly, if an individual dominates the other, then the non-dominated individual is the winner. Secondly, if they are not comparable or their values of the objective functions are equal, then: (a) Being both within the grid of the belief space, then wins the one within a cell less populated (according to the counter of the cells), (b) otherwise, if one falls off the grid, then this one wins (in order to increase the space of exploration).

Upon completion of the tournament, the **P** individuals with a greater number of victories are selected to be promoted to the next generation.

Insertion of individuals to the elite DB. The elite DB contains unique non-dominated individuals. To add an individual to the elite DB, if the candidate individual (CI) is dominated by any individual within the DB, then the CI is discarded. Otherwise, it replaces the set of individuals it dominates leaving other (pairs) non-dominated individuals in the DB. Hence, all individuals living within the DB corresponds to non-dominated among themselves, i.e., are spread along the Pareto front.

V. Results

The emulation model. The cultural algorithm was implemented using C++ programming Language in an Intel Core i5 computer. Source code is available in a GitHub repository at https://github.com/antonioaraujob/omocac. The parameters used for simulations are shown in Table I.

Table I. PARAMETER VALUES FOR SIMULATIONS

Parameter	Value
Initial population (P) size	20
Max. generation number	200
Standard deviation for Gaussian mutation MinCT	1
Standard deviation for Gaussian mutation MaxCT	3
Grid subinterval number	10
Update frequency for phenotypical normative part	$M = 5$
Tournaments by individual	10
Directed mutation probability	0.3

The Wi-Fi deployment is emulated, based upon results derived from an extensive measurement campaign in Rennes, France [4]. We war-walked 2.5 km of an urban area densely covered by Wi-Fi. We have registered regular AP characteristics such as SSID, BSSID, RSSI and, we have managed to carefully and orderly trace the time-difference between all probe responses. Later, we computed a CDF for inter-probe-response time for each individual Wi-Fi channel (i.e., from 1 to 11). We further modelled an empty channel out of the probability of receiving no Probe Response before *MinCT*, otherwise our model is able to emulate the discovery of recorded APs based on the empirical probability distribution for inter-probe-response time.

Computing smart sequences. After few hundreds of generations, we got channel sequences with optimized values of scanning parameters (*MinCT*, *MaxCT*) as it is shown in Table II. In this table, each channel is represented as follows: $\mathbf{Ch}^{l}_{<m,M>\rightarrow AP}$ where *Ch* is a channel within the sequence (between Ch1 and Ch11), *l* is the latency obtained as *MinCT+MaxCT* for which previous values get accumulated, *m* is *MinCT*, *M* is *MaxCT*, *AP* is the number of APs discovered in the channel. Note also that the order of channels is different for every sequence. This is because the CA has adapted channel speed according to the response of the system on every different channel, hence finding the maximum rate of useful APs per time unit in every channel. Finally observe that the optimal sequence improves the scanning time in 65% compared to the reference sequence. This reference sequence was obtained by reverse engineering the scanning sequence of an iPhone 4 using iOS 7. The sequence consist of an increasing scanning on all channels from 1 to 11 using a fixed timer of 37 ms on each channel [7].

Table II show scanning sequences sorted by average number of APs, found per channel, in descending order calculated after the TM has computed enough sequences. We can see that the number of APs per channel for the reference sequence (in the last row) is systematically higher than an optimal sequence (on the rest of the rows). This higher appearance is because more (hidden) APs appear if we wait longer. The main reason for an AP showing late is because the Probe Responses from those APs get through after several retransmissions. However, given that we are restricting the latency, we are interested on those APs that appear at the beginning of the waiting time-window (i.e., *MinCT*). Note that these are the average number of APs appearing after 30 independent scannings. The smart sequences correspond to non-dominated individuals calculated by the Cultural Algorithm.

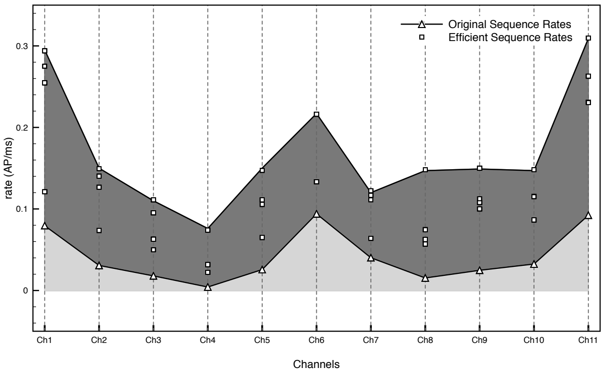

Figure 4. Comparison rates for the Cultural Algorithm smart sequences

Fig. 4 show the comparison of various smart sequences derived from the CA. Originally, the algorithm was fed with the reference sequence (on the last row in Table II), and afterwards the algorithm derived a variety of sequences whose discovery rates are shown within the dark grey region. Note that all sequences have considerably higher discovery rate compared to the reference. Moreover, delay improvements range from 30% to 70% respect to the reference.

Different configurations of channel sequences can be used for applications that support specific values of latency, e.g., elastic applications could consider channel sequences with longer timers. Obtaining sequences for these restrictions is easier since the application becomes delay tolerant and we have observed that in crowded environments APs keep sending probe responses within 100 ms span. Hence this is an upper bound that makes scanning sequences independent of the deployment.

VI. CONCLUSION AND FUTURE WORK

In this work we have shown that the scanning process can be significantly improved in Community Network (Wi-Fi) deployments by using computational intelligence techniques such as Cultural Algorithms. We have successfully characterized the trade-off for a better discovery process by using the discovery rate (APs per time unit) versus the total latency for scanning. Having efficient sequences produce direct energy savings, because resources are dedicated to channels that worth interaction with an appropriate waiting time.

We have also presented the design of a Topology Manager (TM) and discussed the interaction with wireless users. We have proposed two separate functionalities for the TM: the wireless topology model abstraction and the channel sequence calculation engine. For the topology model we require feeders to send their partial vision of the topology so that the TM can incrementally build the model. Eventually, the calculation engine computes efficient sequences through the Cultural Algorithm so that client's queries could be asynchronously attended.

We are working on the implementation of a distributed TM that allows separating the network into cells for obtaining better scanning sequences on smaller and more cohesive portions of the network. We have observed that the model behind the TM could be more precise if we include areas with the same characteristics, for example, a commercial zone in downtown can be on its own managed by a separate TM building a different network model.

Table II. CHANNEL SEQUENCES ORDERED BY DESCENDING NUMBER OF DISCOVERED APS.

Channel scan sequence ordered by total discovered APs											OF_1 $(AP/ms$	OF_2 (*ms*)
$\mathbf{11}^{12}$ <7,5>1.633 0.26	$\mathbf{1}^{25}$ <10,3>1.16 0.12	$\mathbf{2}^{45}$ <15,5>0.83 0.13	$\mathbf{6}^{59}$ <11,3>0.73 0.22	$\mathbf{10}^{72}$ <7,6>0.533 0.09	$\mathbf{9}^{81}$ <6,3>0.5 0.15	$\mathbf{3}^{91}$ <7,3>0.5 0.10	$\mathbf{8}^{104}$ <8,5>0.5 0.06	$\mathbf{7}^{115}$ <8,3>0.366 0.12	$\mathbf{5}^{124}$ <6,3>0.3 0.11	$\mathbf{4}^{134}$ <7,3>0.066 0.03	1.39	134
$\mathbf{11}^{24}$ <15,9>3.53 0.31	$\mathbf{1}^{41}$ <9,8>2.033 0.29	$\mathbf{6}^{62}$ <14,7>1.96 0.13	$\mathbf{9}^{73}$ <8,3>0.766 0.10	$\mathbf{2}^{87}$ <8,6>0.6 0.07	$\mathbf{5}^{96}$ <6,3>0.533 0.11	$\mathbf{8}^{109}$ <7,6>0.433 0.07	$\mathbf{10}^{121}$ <9,3>0.4 0.15	$\mathbf{7}^{131}$ <6,4>0.333 0.06	$\mathbf{3}^{140}$ <6,3>0.133 0.11	$\mathbf{4}^{152}$ <6,6>0.066 0.02	1.43	152
$\mathbf{11}^{25}$ <15,10>3.3 0.24	$\mathbf{6}^{43}$ <15,3>1.9 0.14	$\mathbf{1}^{60}$ <9,8>1.733 0.21	$\mathbf{5}^{73}$ <10,3>1.06 0.17	$\mathbf{9}^{86}$ <10,3>0.66 0.09	$\mathbf{10}^{99}$ <10,3>0.63 0.12	$\mathbf{3}^{116}$ <11,6>0.56 0.13	$\mathbf{7}^{132}$ <10,6>0.53 0.12	$\mathbf{8}^{144}$ <6,6>0.433 0.09	$\mathbf{2}^{154}$ <7,3>0.233 0.13	$\mathbf{4}^{164}$ <7,3>0.066 0.05	1.49	164
$\mathbf{11}^{14}$ <10,4>1.76 0.22	$\mathbf{6}^{31}$ <10,7>1.6 0.19	$\mathbf{1}^{44}$ <10,3>1.56 0.30	$\mathbf{5}^{64}$ <12,8>0.86 0.08	$\mathbf{7}^{79}$ <11,4>0.83 0.14	$\mathbf{2}^{93}$ <11,3>0.76 0.09	$\mathbf{10}^{114}$ <15,6>0.7 0.06	$\mathbf{8}^{132}$ <13,5>0.63 0.06	$\mathbf{9}^{152}$ <15,5>0.6 0.10	$\mathbf{4}^{166}$ <11,3>0.26 0.06	$\mathbf{3}^{184}$ <15,3>0.26 0.22	1.52	184
$\mathbf{11}^{39}$ <39,0>3.63 0.11	$\mathbf{1}^{78}$ <39,0>3.26 0.06	$\mathbf{6}^{117}$ <39,0>3.16 0.08	$\mathbf{5}^{156}$ <39,0>1.4 0.03	$\mathbf{7}^{195}$ <39,0>1.16 0.04	$\mathbf{10}^{234}$ <39,0>1.16 0.02	$\mathbf{2}^{273}$ <39,0>1.16 0.03	$\mathbf{8}^{312}$ <39,0>0.56 0.01	$\mathbf{3}^{351}$ <39,0>0.53 0.02	$\mathbf{9}^{390}$ <39,0>0.5 0.02	$\mathbf{4}^{429}$ <39,0>0.2 0.01	0.43	429

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's (EU) Horizon 2020 research and innovation programme under grant agreement No. 644663. Action full title: architectuRe for an Internet For Everybody, Action Acronym: RIFE.

REFERENCES

[1] A. Sathiaseelan, C. Rotsos, C. Sriram, D. Trossen, P. Papadimitriou, and J. Crowcroft, "Virtual public networks," in *Software Defined Networks (EWSDN), 2013 Second European Workshop on*, Oct 2013, pp. 1–6.

[2] T. Serbulent, S. Murat, M. Wei, K. Abtin, and B. Robert, "Wi-Fi Enabled Sensors for Internet of Things: A Practical Approach," *Topics in Consumer Communications and Networking*, pp. 1–10, May 2012.

[3] G. Castignani, L. Loiseau, and N. Montavont, "An evaluation of ieee 802.11 community networks deployments," in *Information Networking (ICOIN), 2011 International Conference on*, Jan 2011, pp. 498–503.

[4] A. Arcia-Moret, L. Molina, N. Montavont, G. Castignani, and A. Blanc, "Access Point Discovery in 802.11 Networks," in *IEEE Wireless Days*, 2014.

[5] X. Hu, L. Song, D. V. Bruggen, and A. Striegel, "Is there wifi yet? how aggressive wifi probe requests deteriorate energy and throughput," *CoRR*, vol. abs/1502.01222, 2015. [Online]. Available: http://arxiv.org/abs/1502.01222

[6] N. Brouwers, M. Zuniga, and K. Langendoen, "Incremental wi-fi scanning for energy-efficient localization," in *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*. IEEE, 2014, pp. 156–162.

[7] A. Araujo and A. Arcia-Moret, "Identifying raw scanning sequences on IEEE 802.11 networks," *Revista Venezolana de Computación*, vol. 1, no. 1, 2014.

[8] A. Mishra, M. Shin, and W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," *SIGCOMM Computer Communication Review*, vol. 2, no. 33, pp. 93–102, 2003.

[9] "IEEE Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard, 2012.

[10] K. H. R. Price, "Performance Comparison of Handover Mechanisms in Wireless LAN Network," Institue of Computer Science, Würzburg Univ, Würzburg, Technical Report 339, December 2004.

[11] G. Castignani, A. Arcia, and N. Montavont, "A study of the discovery process in 802.11 networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 15, no. 1, pp. 25–36, Mar. 2011. [Online]. Available: http://doi.acm.org/10.1145/1978622.1978626

[12] N. Montavont, A. Arcia-Moret, and G. Castignani, "On the selection of scanning parameters in ieee 802.11 networks," in *IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, London, United Kingdom, 2013.

[13] A. Sathiaseelan, J. Crowcroft, M. Goulden, C. Greiffenhagen, R. Mortier, G. Fairhurst, and D. McAuley, "Public access wifi service (paws)," in *Digital Economy All Hands Meeting, Aberdeen*, 2012.

[14] C. Coello and R. Landa, "Evolutionary Multiobjective Optimization Using a Cultural Algorithm," in *IEEE Swarm Intelligence Symposium*, 2003.

[15] A. Arcia-Moret, A. Sathiaseelan, A. Araujo, J. Aguilar, and L. Molina, "Assisted network discovery for next generation wireless networks," in *The 13th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, USA, 9-12 January 2016.

[16] H. Velayos and G. Karlsson, "Techniques to reduce the ieee 802.11b handoff time," in *Communications, 2004 IEEE International Conference on*, vol. 7, June 2004, pp. 3844–3848 Vol.7.

[17] G. Castignani, A. Arcia-Moret, and N. Montavont, "An evaluation of the resource discovery process in ieee 802.11 networks," in *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, ser. MobiOpp '10. New York, NY, USA: ACM, 2010, pp. 147–150. [Online]. Available: http://doi.acm.org/10.1145/1755743.1755770

[18] G. Castignani, N. Montavont, A. Arcia-Moret, M. Oularbi, and S. Houcke, "Cross-layer adaptive scanning algorithms for ieee 802.11 networks," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, March 2011, pp. 327–332.

[19] J. Montavont, N. Montavont, and T. Noel, "Enhanced schemes for l2 handover in ieee 802.11 networks and their evaluations," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 3, Sept 2005, pp. 1429–1434.

[20] Y. Liao and L. Cao, "Practical schemes for smooth mac layer handoff in 802.11 wireless networks," in *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, 2006, pp. 10 pp.–190.

[21] 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 *Information Networking, 2009. ICOIN 2009. International Conference on*, Jan 2009, pp. 1–7.

[22] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing mac layer handoff latency in ieee 802.11 wireless lans," in *Proceedings of the Second International Workshop on Mobility Management &Amp; 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

[23] 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*, ser. MobiCom '08. New York, NY, USA: ACM, 2008, pp. 199–210. [Online]. Available: http://doi.acm.org/10.1145/1409944.1409968

[24] Y. Yiakoumis, M. Bansal, A. Covington, J. van Reijendam, S. Katti, and N. McKeown, "Behop: A testbed for dense wifi networks," in *Proceedings of the 9th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, ser. WiNTECH '14. New York, NY, USA: ACM, 2014, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/2643230.2643233

[25] D. Trossen and G. Parisis, "Designing and realizing an information-centric internet," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 60–67, July 2012.