A Machine Learning Approach for IEEE 802.11 Channel Allocation

Olivier Jeunen*, Patrick Bosch†, Michiel Van Herwegen‡, Karel Van Doorselaer‡, Nick Godman‡, and Steven Latré†
*University of Antwerp, Adrem Data Lab, Department of Mathematics and Computer Science, Belgium
†University of Antwerp - imec, IDLab, Department of Mathematics and Computer Science, Belgium
‡Technicolor, Belgium

***Abstract*—Today's communication is mainly done over wireless networks, with IEEE 802.11 (Wi-Fi) at the forefront. There are billions of devices and millions of access points (APs), but only very few non-overlapping channels. As a result, the performance of Wi-Fi devices is severely degraded, because perfect channel allocation - with every AP alone in its channel - is close to impossible. Even in situations where all networks are under centralised control, existing approaches quickly tend to be either unscalable or suboptimal. By focusing on a subset of problems, identifying Wireless Local Area Networks (WLANs) that severely interfere with each other, performance can be improved even in such a complex situation. We tackle this problem through machine learning and coin it Bad Neighbour Detection (BND). Based on this output alongside monitoring data about the networks' activity, we then propose a channel allocation that optimises performance and as a side effect, stabilises networks that we do not control. We evaluate our approach in a field trial and show that we significantly improve the experience for users, eliminating virtually all interference-related issues.**

I. Introduction

Wireless devices, and especially Wi-Fi devices, are ubiquitous in our society. This leads to a high amount of interference and congestion as the medium is shared by all devices. With it comes performance degradation and an overall decreasing user experience. The root cause of problems can be categorised into *medium access* problems on transmitter side, (i.e. the medium is occupied by other Wi-Fi or non-Wi-Fi devices) and *frame delivery* problems affecting the link capacity [1]. The latter can be caused by radio path issues such as high path loss as well as by interference/collisions at receiver side which will both trigger the Wi-Fi link to step back and use slower but more robust physical layer rates. This reduced link capacity results in a less efficient usage of the medium, contributing indirectly to congestion of the medium. Problems can even be caused by probe requests, which have been shown to cause a significant decrease in throughput in crowded wireless areas [2].

In recent years, along with the rise of Internet of Things (IoT), wireless network usage has continued to grow at astonishing rates. In densely populated areas, it is not uncommon for a Wi-Fi access point to share the medium with dozens of neighbouring networks. Biswas et al. substantiate that most of the interference Wi-Fi networks endure, originates from neighbouring Wi-Fi networks operating on the same or partially overlapping frequency channels [3].

This suggests that congestion (sharing Wi-Fi) and interference problems (related to medium access as well as frame delivery) can in many cases be tackled by moving the AP to a less occupied channel. However, in densely populated areas, finding such a "clean" channel is a non-trivial task.

Throughout this work we use passive monitoring data on AP-level concerning network utilisation, scan lists and such, obtained through the Wi-Fi Doctor™ platform. From this gathered data, aggregate metrics are calculated that quantify the performance loss in a given channel due to interference and/or congestion. Using these metrics properly allows to find the best channel to use in more or less stable environments. On a higher level of aggregation, metrics that reflect the user experience are calculated from the radio and station related statistics of the APs based on the detection of Wi-Fi bottlenecks - i.e. when little to no bandwidth is available beyond the bandwidth consumed by applications using the Wi-Fi link. These metrics are substantiated throughout Sections III and IV.

The rise of Software-Defined Networking (SDN) entails a growing interest towards managed Wi-Fi networks, where parameters are centrally configured. These environments, however, ask for more advanced resource management algorithms in turn. One of the main challenges remains to define an optimal channel allocation in order to minimise the amount of interference and congestion for all APs. The amount of data collected by the APs and the recent advances in machine learning provide a basis for this problem. Therefore, we propose a novel scheme considering airtime overlap, as well as a method called Bad Neighbour Detection (BND) that identifies devices that are interfering with each other and proposes different channels for them in the following channel allocation algorithm. As this might cause new problems with other devices and Wi-Fi usage patterns can change over time, this process is repeated iteratively. Later on we show that our method succeeds in improving user experience for all involved devices over time.

Our contributions are therefore threefold. First, we provide a novel method to identify interfering devices in a highly complex wireless environment. Second, we present our novel iterative algorithm to perform channel allocation with the interfering devices in mind, based on minimising the overlap in airtime usage amongst neighbouring networks. Third, we eval-

978-3-903176-14-0 © 2018 IFIP

uate our algorithm in a field trial with a European Internet Service Provider (ISP) and show a clear decrease in interference issues that degrade Quality of Experience (QoE). Due to this real-world evaluation, room for further experimentation with extensions that include Tx power adjustment and the 5 GHz band into the algorithm was limited. However unfortunate, this opens up interesting directions for future research.

The remainder of this paper is structured as follows. First, we explore related work in Section II. Afterwards, we present our novel architecture including BND in Section III. Then we evaluate the architecture in Section IV, followed by a discussion in Section V. We conclude in Section VI.

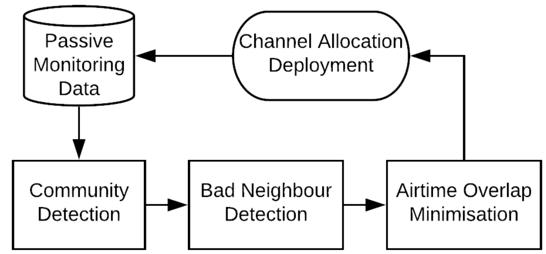

Fig. 1. The full iterative channel allocation process

II. Related Work

Multiple centralised iterative schemes for network-controlled channel allocation have been proposed in recent years. Lim et al. propose a centralised scheme to allocate channels and consider the unmanaged neighbouring APs [4]. Their solution estimates the channel utilisation based on the received beacon signals. Assigning clients to channels that are just good enough to maximise the number of clients per AP and leaving other channels to APs that need more throughput, Kasasbeh et al. propose an algorithm that takes out-of-system interference, random AP deployment, and unmanaged APs into account [5]. Taking channel monitoring results and data traffic from APs into account, Abeysekera et al. propose a centralised algorithm that computes the quasi-optimal frequency channel of each AP [6]. Balbi et al. use a centralised algorithm to organise APs in a way to minimise interference between networks [7]. Not only the networks managed by the controller are taken into account, but unmanaged networks as well. Baid and Raychaudhuri examine the correlation between low cost residential APs and APs managed by an ISP [8]. They conclude that central control is beneficial even in the case of unmanaged APs and that for a percentage increase of managed APs, the throughput can be improved.

But not only centralised algorithms were proposed, decentralised were as well. While traditionally, AP placement and channel selection are performed separately, Zhao et al. propose an algorithm that optimises both at once and therefore improves performance compared to other solutions [9]. Dynamic channel assignment based on channel segregation, improves the co-channel interference compared to traditional channel assignments when the environment changes [10]. Gazis et al. present a self-organizing system that utilises communication between APs [11]. The algorithm converges fast and reaches a stable solution.

However, all these works rely on results obtained either through simulations or relatively small test-beds. Seyedebrahimi et al. propose an algorithm for jointly adjusting the transmission power levels and optimising the channel assignment of APs. Taking into account the flow's required qualities it shows a performance improvement of up to 25 % [12]. Static traffic needs are assumed throughout the experimental results, with a focus on optimising the throughput throughout the network. We argue that static traffic needs are unrealistic for real-world environments, as different applications will require different amounts of bandwidth. As a consequence, we will not focus on overall throughput optimisation throughout the rest of this work, but rather on user-experience based metrics w.r.t. Wi-Fi usage.

Other recent work has shown that, especially in densely populated areas, a need rises for channel allocation schemes that collaboratively take information from other APs into account in order to minimise inter-cell interference [13]. Our iterative approach achieves this by matching collected data about network usage and measured interference in order to identify disruptive neighbouring networks, and consecutively taking this information into account when computing a global channel allocation. Moreover, our method provides a scalable channel assignment procedure that only uses data provided by the AP, which makes it widely deployable with minimal end-user intrusion.

III. Architecture and Algorithms

In what follows we provide an overview of the architecture of our approach, its different components and how they fit together. Figure 1 visualises the full iterative process, from passive monitoring to deployment. For simplicity and without loss of generality, we focus on the classical case of Wi-Fi networks in the 2.4 GHz spectrum throughout the rest of this paper. Note however that our approach can easily be generalised to include the 5 GHz band, provided some additional measures are taken w.r.t. varying channel widths.

We set up the controlled APs to periodically scan the medium to detect neighbouring networks, generating scan lists that contain information from the detected beacons: Service Set Identifier (SSID), measured Received Signal Strength Indicator (RSSI), et cetera. Let $G = (V, E)$ be an undirected graph, where V denotes the set of vertices and E the set of edges. Let every AP i be a vertex v_i and let the set of neighbouring APs derived from a scan list performed by AP i be denoted as $S(v_i)$. To incorporate possible changes in the environment over time, we define $S_u(v_i)$ as the union of all observed sets $S(v_i)$.

$$S_u(v_i) = \cup_{\forall S(v_i)} S(v_i)$$

An edge exists from v_i to every other AP v_j observed in its scan lists, i.e.:

$$\exists e \in E : v_i \mapsto v_j, \forall v_j \in S_U(v_i).$$

This graph model of the environment could, for example, be extended by weighting edges with the mean measured RSSI values; we decided not to further explore this option. It is non-trivial to draw a priori conclusions from RSSI measurements alone: intuitively one would avoid to share the channel with a neighbour with a strong RSSI. But a case could be made that higher RSSI values would lead to less disturbing networks, since they are less likely to produce interfering Wi-Fi packets as chances are high that all Wi-Fi nodes in the network will see each other, allowing the listen-before-talk IEEE 802.11 Medium Access Control (MAC) protocol to work properly. It would be far worse to introduce hidden nodes than an AP we can efficiently share the medium with. We provide and discuss results w.r.t. the correlation between RSSI and interfering networks in Section IV-D.

A. Community Detection and Dynamic Programming

As large-scale metropolitan areas are prone to generate densely interconnected graphs, any optimal computation involving them quickly becomes unscalable. We therefore look into community detection techniques, which divide graphs into logical components.

Once we have obtained these community structures in the network graph, we use this information to break down the global channel allocation problem into a local allocation per such community. Later on, additional constraints are added to minimise channel overlap between neighbouring nodes that belong to different communities when combining the local allocations to form the global solution. The methodology of solving a complex problem through its often simpler sub-problems is widely known as Dynamic Programming.

By detecting communities and solving the channel allocation per community in a first iteration, we also take the limitations of our model into account: by looking at communities rather than individual links, we incorporate the most probable candidate APs to generate (hidden node) interference into the set of neighbouring APs for every AP.

A first logical step when detecting communities is to look at individual connected components instead of the super-graph as a whole [14].

a) Label Propagation Algorithm: The Label Propagation Algorithm (LPA) is a stochastic algorithm with linear runtime complexity. Initially, unique random labels are assigned to every node. These labels are then consecutively propagated throughout the network, according to the labels occurring with the highest frequency among a node's neighbours [15].

b) Girvan-Newman: The Girvan-Newman algorithm iteratively removes the edge with the highest betweenness centrality (as defined by Freeman [16]) from the network [17]. As it is considerably more computationally expensive than LPA, but generally yields intuitive and deterministic results, we apply it where LPA falls short.

B. Minimising Airtime Overlap

Define $\mathbb{V}$ as the set of all access points v_i we want to allocate channels to and $\mathbb{K}$ as the set of non-overlapping channels to allocate. Assume we have $|\mathbb{V}| = n$ access points and $|\mathbb{K}| = m$ channels. We define the inter-AP distance matrix $\mathbf{D}$ as

$$\mathbf{D} = (d_{ij}) \in \mathbb{R}^{n \times n}$$

$$d_{ij} = dst(v_i, v_j), \forall v_i, v_j \in \mathbb{V}$$

$$dst : \mathbb{V} \times \mathbb{V} \mapsto \mathbb{R}^+$$

where dst is a distance function. Note that this is a conceptual distance, and not related to physical distance or RSSI values.

Furthermore, the channel allocation matrix $\mathbf{C}$ is defined as

$$\mathbf{C} = (c_{ik}) \in \mathbb{R}^{n \times m}$$

where

$$c_{ik} = \begin{cases} 1, & \text{if AP } i \text{ is assigned channel } k \\ 0, & \text{if AP } i \text{ is } \textbf{not} \text{ assigned channel } k. \end{cases}$$

Assuming a fixed distance function for now, our objective function is formalised as follows:

Minimise

$$F(\mathbf{D}, \mathbf{C}) = \sum_{k \in \mathbb{K}} \sum_{i \in \mathbb{V}} \sum_{j \in \mathbb{V}} \frac{1}{1 + \mathbf{D}_{ij}} \mathbf{C}_{ik} \mathbf{C}_{jk} \tag{1}$$

subject to

$$\sum_{k \in \mathbb{K}} C_{ik} = 1, \forall v_i \in \mathbb{V}. \tag{2}$$

The rationale behind Equation 1 is that APs with the largest distance between them should be assigned to the same channel, as they are less likely to interfere with each other than APs with a smaller distance between them. The constraints in Equation 2 are added to ensure every AP gets assigned exactly one channel, thus avoiding the trivial solution where all $\mathbf{C}_{ik}$ are set to zero.

1) Activity-based distance function: We propose an activity-based distance function to be used when constructing the matrix $\mathbf{D}$. Wi-Fi activity in a WLAN is indicated by the airtime usage (transmitting or receiving) in the radio statistics of the APs. For a week of reference data, we discretise airtime usage over this week into 672 bins representing a 15 minute window:

$$\forall v \in \mathbb{V} : u_v \in \mathbb{R}^{672}.$$

Next, we use the constraint dynamic time warping technique ($cdtw$) proposed by Sakoe and Chiba [18] to compute the distance between two such time series. The constraint is set to 3, representing a one-hour window.

$$cdtw(u_i, u_j) : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^+$$

$$\forall v_i, v_j \in \mathbb{V} : dst(v_i, v_j) = cdtw(u_i, u_j)$$

With this approach we effectively minimise airtime overlap between medium sharing APs: a heavy user during evening hours could perfectly well share the medium with a heavy user during office hours, because of the minimality in airtime usage overlap.

We introduce a learning method that takes the weighted average of historical and new data, that is:

$$u_{i+1} = \alpha . u_i + (1 - \alpha) u_{\text{new}}.$$

Where u_i is the usage pattern up to last week, u_{new} is the usage pattern during the last week, and u_{i+1} is the eventual updated pattern. $0 \leq \alpha \leq 1$ is the learning rate, specifying whether there should be an emphasis on either new or historical data.

C. Linear Programming

Linear programming is a commonly known method to compute the global optimum for a linear objective function of a mathematical model represented by linear relationships, subject to linear equality or inequality constraints. It was first proposed by Luenberger and has been a very active field of research, with various extensions such as quadratic programming, mixed integer programming, et cetera [19]. The specific subcategory we use is binary integer programming, i.e. the special case where all variables are required to be either 0 or 1.

The mathematical model and objective function defined in the previous section consist of linear relationships and linear equality constraints. The objective function, however, is quadratic. Various linearisation techniques to tackle this issue are proposed by Adams and Sherali [20].

For each pair of binary variables x, y that appear in quadratic relationships, define $z = x.y$, subject to

$$z \leq x$$

$$z \leq y$$

$$z \geq x + y - 1.$$

The first two constraints ensure the newly defined variable is 0 if any of the original variables is, whereas the latter makes sure z will be set to 1 when both x and y are as well.

D. Bad Neighbour Detection

As we have collected data regarding the networks' airtime usage as well as other measurements on the medium, we can exploit this to detect and analyse performance issues. We define BND as the problem of identifying pairs of neighbouring networks where at least one network is disruptive in terms of available bandwidth for the other network. In this way we detect a suffering neighbour along with a bad neighbour (the relationship is often symmetrical, meaning that the suffering neighbour is in itself also a bad neighbour and vice versa).

Our approach works on aggregated measurement data of airtime usage and Relative Channel Interference (RCI). RCI quantifies the amount of performance that is lost due to interference relative to the theoretical maximum performance of a Wi-Fi link. It is defined as the relative performance improvement that would be obtained in case all interference and/or congestion could be removed. This metric - which is used as an input to the channel recommendation provided by Wi-Fi Doctor™- is calculated based on the radio and station related statistics reported by the AP for all active Wi-Fi links [21]. The main line of reasoning is that we try to explain the interference a network experiences by a weighted sum of medium usage of its bad neighbours, along with some bias. Medium usage by any or all Wi-Fi links associated to this radio, either by the radio transmitting or receiving frames from its associated stations, is measured over 2 second intervals and later aggregated over 10 seconds. RCI is calculated over the same 10 s interval. After aligning the measurement intervals between APs, the data is transposed to match a row of measured RCI in a certain radio to medium usage of its neighbours. Table I shows the structure of the final preprocessed data. In this toy example, it might very well be that radios A and B are bad neighbours: their RCI is perfectly correlated with each others medium usage (plus some bias term). Naturally, an APs' own medium usage is not taken into account when computing bad neighbours for said AP.

TABLE I
BND DATA EXAMPLE

Radio	Interval	Channel	RCI	Airtime usage A	Airtime usage B
A	00:00:00 - 00:00:10	1	0.7	0.2	0.6
A	00:00:10 - 00:00:20	1	0.5	0.2	0.4
A	00:00:20 - 00:00:30	1	0.3	0.2	0.2
A	00:00:30 - 00:00:40	1	0.8	0.2	0.7
B	00:00:00 - 00:00:10	1	0.2	0.2	0.6
B	00:00:10 - 00:00:20	1	0.2	0.2	0.4
B	00:00:20 - 00:00:30	1	0.2	0.2	0.2
B	00:00:30 - 00:00:40	1	0.2	0.2	0.7

1) Feature Selection: As we target densely populated areas with densely interconnected network graphs as a consequence, Table I grows exponentially. Our approach to solving this regression problem should thus be able to handle the high-dimensional nature of the data both for scalability and preventing over-fitting. With this criteria in mind, we use Least Absolute Shrinkage and Selection Operator (LASSO) regression [22] with the RCI as dependent variable to predict, based on the neighbours' medium usage as independent variables. We include a bias term and force non-negative coefficients: we do not want to find inverse correlations between medium usage in network A and interference in network B. For our model selection procedure, we perform LASSO through least angle regression with the Bayes Information Criterion as metric to evaluate lambdas [23]. All medium usage inputs that do not have a 0 coefficient in the result, are retained after feature selection. The other independent variables are dropped for both the training and testing procedures.

2) Neighbour Importance: After this feature selection procedure promoting model sparsity, we perform Ordinary Least-Squares (OLS) regression instead of directly evaluating on the LASSO model [24]. This two-step approach yields two-fold benefits: it allows for non-penalised coefficients whilst still delivering a sparse solution, as well as full control over which independent is left out (which we exploit when evaluating candidate bad neighbours).

The performance of the OLS model is measured by calculating the R^2 metric of the trained model when predicting for the test set. Intuitively, it represents the proportion of the variance in RCI that can be explained by the model, through the neighbours' medium usage.

Next, for every possible bad neighbour, a new OLS model is trained on the train set, omitting that neighbours' medium usage. The R^2 metric is calculated for the test set with every new model.

The final *bad neighbours score* of every radio is the R^2-score of the full OLS model minus the R^2-score of the OLS model that excluded the medium usage of said radio. This in effect gives an estimate on how much of the variance is *at least* due to that specific radio.

The higher the score, the worse the neighbour is. Low scores are unlikely to represent actual bad neighbours, high scores are most likely to be main causes of interference. From experimentation, a hard cut-off on this score can prevent most false positives, if needed.

E. Iterative Updates

In what follows we combine the methods outlined above to formalise our final channel allocation algorithm.

1) We analyse the networks' scan lists and generate a graph representing the network topology. In the case of differing results (e.g. A's Basic Service Set Identifier (BSSID) is sometimes present in network B's scan lists, and sometimes not) we decide to use a conservative approach: if a BSSID is seen in a scan list at least once, an edge between both APs is drawn.
2) We perform community detection. Depending on the network topology, different techniques are preferential. For smaller networks with lots of connected components, the Girvan-Newman algorithm performs very well. For larger networks that are more interconnected, the use of LPA proves to be more scalable. If attainable, a good goal is to aim for communities no larger than 25 APs, with a diameter of no more than 4 hops. This first limit is to keep the linear programming algorithm's optimality computations feasible, whereas the latter rests on the fact that when the number of hops between two APs increases, in the limit, the probability of generating hidden nodes will decrease.
3) For every detected community, we compute the optimal channel allocation with respect to the activity-based distance function formulated in Equation 1. We generate a mapping for every AP to one of three channels.
4) We merge these individual results for every community together: where edges between different communities are present, we calculate the optimal permutation of community A's mapping compared to community B. Formally: community A has a mapping for every AP to one of three channels $\{x, y, z\}$, whereas B's APs are mapped to $\{x', y', z'\}$. For every possible permutation of A's channels, we compute the objective function for the inter-community edges. For the optimal permutation, channels are then mapped onto each other.
5) We now have a mapping from AP to $\{x, y, z\}$ for the whole network. Similarly, we compute an objective function for every permutation of the available channels $\{1, 6, 11\}$. We minimise the average number of external BSSIDs every AP sees, based on the scan lists. The rationale behind this is an attempt to minimise external influences on our network.
6) We recompute distances between APs based on their updated usage patterns on a weekly iterative basis. Results from the BND technique outlined above are included into the model as hard constraints: pairs of bad neighbours should not be allowed to be allocated the same channel.

TABLE II
BASIC PROPERTIES OF THE NETWORK GRAPH

# Nodes	# Edges	Avg. degree	Min. degree	Max. degree
60	128	4	1	8

IV. Evaluation

We experimentally validated our approach during a real-life field trial with a large European ISP over the course of several weeks. The ISP operates several Multiple Dwelling Units (MDUs) (large apartment buildings where a significant amount of residential units occur in vicinity of each other). Such MDUs often have an exclusive arrangement with their ISP, ensuring a maximally controllable and minimally contaminated environment. Updates to the channel assignment were deployed on a weekly basis.

Our results are evaluated on four different aspects, broadly relating with the steps lined out in Section III: first we analyse the environment and its degree of controllability, then we provide results of our community detection approach; afterwards we assess the evolution in regards of QoE, and finally we study the correlation between RSSI values (obtained through periodic scan lists) and the occurrence of validated bad neighbours. Furthermore, we discuss the impact of our approach on uncontrolled neighbouring networks.

A. Environment

An elementary quantitative breakdown of the MDU used for the field trial can be found in Table II. When modelling the network graph, we chose to only take Wi-Fi Doctor™ enabled devices into account, as uncontrolled devices are not explicitly considered during the optimisation procedure of our channel allocation algorithm. However, we exploit information about the occurrence of neighbouring uncontrolled networks when mapping the abstract channels from the algorithm onto physical channels.

Table III presents an overview of the number of respectively known and unknown APs seen by Wi-Fi Doctor™ enabled

TABLE III
NUMBER OF (UN)KNOWN NEIGHBOURING APS

	Min.	Max.	Avg.
Known	1	8	4.27
Unknown	0	5	1.99
All	2	13	6.25

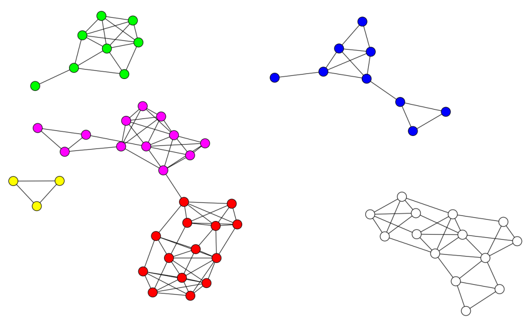

Fig. 2. A visualisation of community detection results on the network graph of the experimental environment, every colour denotes a different community

TABLE IV
QUANTITATIVE ANALYSIS OF THE COMMUNITY DETECTION RESULTS SHOWN IN FIGURE 2

Colour	c_{id}	$\|\mathbb{V}_c\|$	deg_{intra}	deg_{inter}	d
■ (magenta)	0	12	4.50	0.08	3
■ (red)	1	14	5.67	0.07	3
■ (green)	2	8	4.00	0.00	3
■ (blue)	3	9	2.89	0.00	4
	4	14	4.14	0.00	4
■ (yellow)	5	3	2.00	0.00	1
	Mean	10	4.23	0.03	3

devices. With on average nearly 70 % of the environment being controlled devices, it proved to be a perfect environment for our experiments.

B. Community Detection

Figure 2 visualises the result of the community detection procedure laid out in Section III on the MDU used in the field trial. A quantitative analysis of the generated communities is presented in Table IV. We define the intra-community degree as the number of outgoing edges to vertices within the same community, whereas we define the inter-community degree as the number of outgoing edges to vertices in other communities. The diameter of every community is shown as d, this is the longest shortest path between any two nodes belonging to the same community.

As the network graph clearly consists of multiple connected components, performing community detection on this graph is reduced to a rather trivial case. To further validate our approach, we show community detection results on another network consisting of 196 APs in one single connected component in Figure 3. As the controlled part of the environment

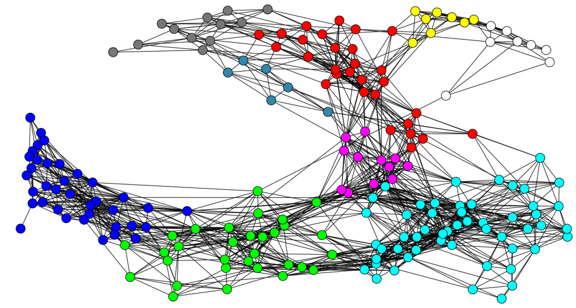

Fig. 3. Community detection results on the network graph of a second environment

from which the graph is constructed was only 15 %, we used a different, more decentralised methodology to tackle interference issues.

C. Quality of Experience

1) Metrics: Wi-Fi Doctor™ provides a metric called Wi-Fi Experience Index (WFEI) to indicate the end user Wi-Fi QoE of a given network over a given longer period of time - typically one week. In order to calculate WFEI, an intermediate metric called Wuxi is calculated for each 10 s interval of each active Wi-Fi link in the WLAN. This metric quantifies the estimated chance that the observed Wi-Fi conditions result in a problem that is noticeable to the end user. It is based on the calculation of the available bandwidth and the observed data rate consumed by the application(s) using the Wi-Fi link in order to identify Wi-Fi bottlenecks [1, 25]. WFEI is obtained through the summation of Wuxi for all Wi-Fi links in the WLAN over the longer time period into SWuxi, and performing a logarithmic transformation function that has been calibrated to cover all cases observed in large scale field trials involving thousands of end user WLANs. WFEI 100 indicates a near-perfect WLAN, whereas WFEI 0 indicates WLANs where the bandwidth available to applications is fully bottlenecked during several hours by the reduced Wi-Fi bandwidth.

In order to distinguish between the different causes of QoE problems, interference or radio path issues, the weight of these causes is calculated for each sample by comparing how much bandwidth could be gained by removing each of the problem causes separately. For example, if a Wi-Fi link only has 10 Mbps available because the medium is occupied for 90 % of the time and the signal strength is such that the link speed is reduced to half the theoretical maximum, a gain of 90 Mbps and 10 Mbps could be obtained by removing interference and radio path issues respectively, leading to a 90 % and 10 % for interference and radio path "weight".

2) Results: Figure 4 shows the evolution of the trend in WFEI for the APs involved in the field trial, throughout the field trial. We provide the mean and minimum WFEI: this gives a global overview as well as the worst case. The mean WFEI shows an upward trend throughout the field trial, but as it already starts quite high this is hard to notice. For the worst

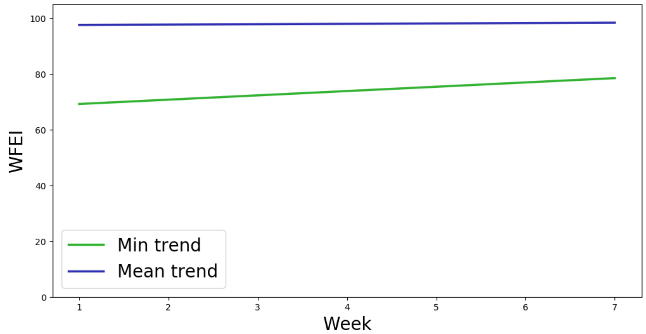

Fig. 4. Wi-Fi Experience Index trend throughout the field trial

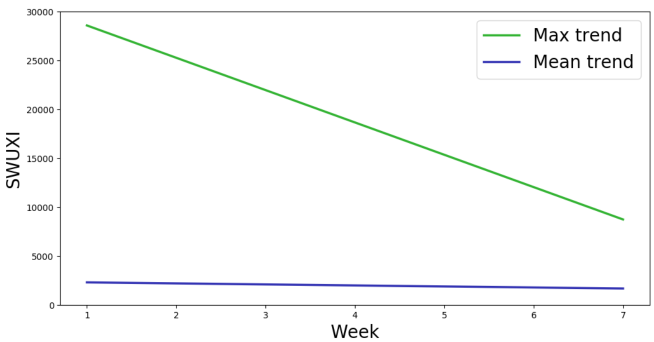

Fig. 5. SWuxi trend throughout the field trial

performers however, we can clearly see the WFEI rising from $\sim 70\%$ to $\sim 80\%$.

Note that this graph alone gives the total picture, resulting in a skewed overview: WFEI is not only dependent on problems caused by interference, but radio path issues are included as well. On top of that, the metric's inherent non-linearity makes it unintuitive to compare results. Lastly, artefacts of the environment can come at play here: the environment could very well contain devices with issues that cannot be solved by channel allocation.

To mitigate the above-mentioned issues we include the SWuxi trends in Figure 5. As SWuxi is inversely correlated with WFEI, we show the maximum instead of the minimum trend. Figure 6 shows the trend in SWuxi after we filtered out issues that were not caused by interference-related problems using the interference weight factor. As a consequence, what is left focuses purely on those QoE degrading moments that can be solved through intelligent channel allocation. Here, we clearly see the impact of our algorithm, decreasing the worst performers' throttled link time by a factor of 3.

Finally, Figure 7 visualises the WFEI computed from the SWuxi shown in Figure 6, thus providing a clearer, less skewed and more relevant overview than Figure 4. It is clear to see that we mitigated virtually all interference-related issues, by raising the minimum WFEI from $\sim 80\%$ to $\sim 100\%$ over the course of only six iterations.

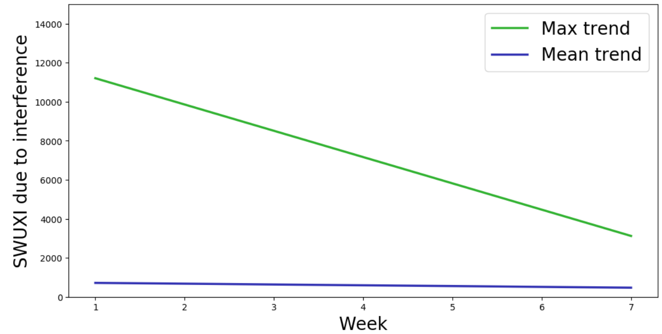

Fig. 6. SWuxi trend due to interference

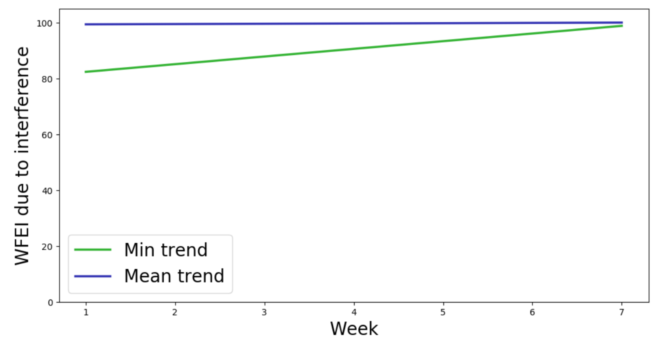

Fig. 7. Wi-Fi Experience Index trend due to interference

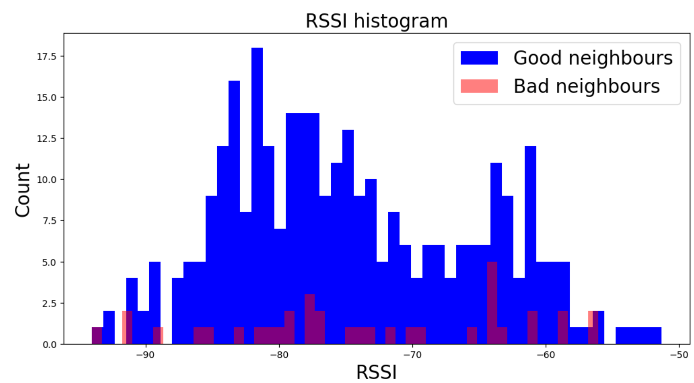

Fig. 8. RSSI distribution for pairs of "good" and "bad" neighbours throughout the field trial

D. Bad Neighbours

Figure 8 shows the distribution in average RSSI measurements for all pairs of neighbouring APs throughout the field trial, based on their scan lists. Identified bad neighbour couples are shown by red bars, whereas blue bars represent those couples whose airtime did not have a significant impact on each other's measured RCI.

E. Uncontrolled Neighbours

Figure 9 shows the evolution of the channels uncontrolled neighbouring networks operate on over time. The first week in the graph is a reference week during which we only pas-

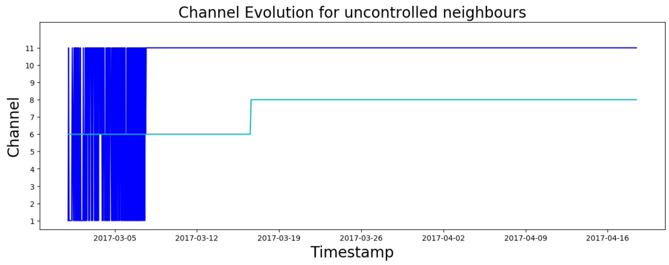

Fig. 9. Evolution of channel choice for uncontrolled neighbours

sively monitored data to gather initial data for the community detection, airtime-minimisation and BND procedures.

V. Discussion

A good community detection should minimise the inter-community degree to intra-community degree ratio whilst sub-dividing the graph into meaningful and manageable clusters. This ensures that the most probable interfering Wi-Fi networks are taken into account together, whereas those that are entirely unrelated are optimised independently of each other. Figure 2 and Table IV clearly show both trends: intra-community edges were much more prevalent than inter-community edges, indicating that the communities accurately represent the network structure as a whole. The network environment depicted in Figure 2 was rather structured and less chaotic, so we used the Girvan-Newman algorithm. However, Figure 3 shows LPA results on a more complex environment, validating the effectiveness of our approach. Defining the optimal number of communities is less formal and more empirical: we managed to reduce the problem size to a point where the exact optimality of our linear programming approach and its computations for every community became feasible and practical. Although it has been shown that binary integer programming is NP-complete [26], our approach managed to find a solution within seconds on commodity hardware. After only a few iterations, the BND algorithm described in Section III-D no longer identified pairs of neighbouring Wi-Fi networks that were in bad conditions due to each other's presence, which also indicates we effectively selected those networks that should share a channel and prevented new pairs from arising. There were still correlations between airtime usage in one network and interference in the other, but no loss in QoE as a consequence of it. This confirms the phenomenon presented by Biswas et al.: the mere presence of a network on a channel does not predict channel utilization [3]. Analogously, more than once, the linear programming method identified the same interfering pairs as the BND technique and assigned them to different channels, demonstrating how these different techniques worked in a complementary fashion. We fixed the learning rate α for the airtime-vectors to 0.5 for the first few weeks, and later on decreased it to 0.1 when we gained confidence in the identified patterns. Optimising this parameter might very well result in a performance gain, but we did not have the resources to explore this further due to the real-world field trial we were bound to.

The RSSI distribution in Figure 8 confirms our reservations about using RSSI. Interfering APs can not be uniquely identified using RSSI measurements alone as bad neighbour couples seem to appear evenly spread over the range of measured values, which justifies the use and advantage of our methodology.

When studying the channel evolution for uncontrolled neighbouring networks as shown in Figure 9, it is clear to see that the environment is extremely chaotic: controlled and uncontrolled neighbouring networks constantly switch channels in an attempt to bypass interference and saturation, but fail to find a static equilibrium. As a side effect of our approach, we can clearly see that after we deployed our more static channel allocation scheme, uncontrolled neighbours were also able to transition into a more harmonious environment.

VI. Conclusion

In this paper, we proposed a novel method for performing channel allocation in IEEE 802.11 networks, based on passive monitoring data obtained from said networks. Our method is data-driven, based on graph analysis, linear programming and regression. We introduced a novel objective function to minimise the overlap in medium usage between neighbouring APs, and a novel methodology to identify and validate pairs of disruptive neighbouring networks that impact each other's QoE, which we have coined BND. These techniques cooperate in a complementary fashion: identifying networks that should or should not be allowed to share frequency channels. Our approach learns from new data over time, making it flexible and dynamic. We presented results from a real-life field trial, and have shown that our iterative technique leads to a clear improvement of QoE and a decrease in QoE loss due to Wi-Fi interference, for the average as well as the worst performers. After only 6 iterations, we were able to solve virtually all interference-related issues. On top of this, as a side effect, we have stabilised the uncontrolled neighbouring networks in the environment.

We have shown that merely using RSSI measurements is insufficient to accurately identify pairs of disruptive neighbouring networks.

Further extensions that are worth studying include the use of partially overlapping adjacent channels, broadening our solution space from the classical 3 non-overlapping channels in the 2.4 GHz band. A coupling factor could be included in Equation 1 to indicate non-existent, full or partial coupling between frequency channels. For this, a thorough quantitative analysis of interference effects on Orthogonal Frequency-Division Multiplexing (OFDM) systems would need to be conducted.

Acknowledgement

Patrick Bosch is funded by FWO, a fund for fundamental scientific research, and the Flemish Government.

References

[1] D. Neves da Hora, K. Van Doorselaer, K. Van Oost, R. Teixeira, and C. Diot, “Passive Wi-Fi Link Capacity Estimation on Commodity Access Points,” in *Traffic Monitoring and Analysis Workshop (TMA) 2016*, 2016.

[2] X. Hu, L. Song, D. Van Bruggen, and A. Striegel, “Is There WiFi Yet? How Aggressive WiFi Probe Requests Deteriorate Energy and Throughput,” *Proc. of the 2015 ACM Conference on Internet Measurement Conference*, pp. 317–323, 2015.

[3] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-E, A. Bhartia, and D. Aguayo, “Large-scale measurements of wireless network behavior,” in *Proc. of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM ’15. ACM, 2015, pp. 153–165.

[4] T. H. Lim, W. S. Jeon, and D. G. Jeong, “Centralized channel allocation scheme in densely deployed 802.11 wireless lans,” in *2016 18th International Conference on Advanced Communication Technology (ICACT)*, 2016, pp. 249–253.

[5] H. Kasasbeh, F. Wang, L. Cao, and R. Viswanathan, “Generous throughput oriented channel assignment for infra-structured wifi networks,” in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[6] B. A. H. S. Abeysekera, K. Ishihara, Y. Inoue, and M. Mizoguchi, “Network-controlled channel allocation scheme for ieee 802.11 wireless lans: Experimental and simulation study,” in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, 2014, pp. 1–5.

[7] H. Balbi, N. Fernandes, F. Souza, R. Carrano, C. Albuquerque, D. Muchaluat-Saade, and L. Magalhaes, “Centralized channel allocation algorithm for IEEE 802.11 networks,” in *2012 Global Information Infrastructure and Networking Symposium (GIIS)*, 2012, pp. 1–7.

[8] A. Baid and D. Raychaudhuri, “Understanding channel selection dynamics in dense Wi-Fi networks,” *IEEE Communications Magazine*, vol. 53, no. 1, pp. 110–117, 2015.

[9] W. Zhao, Z. Fadlullah, H. Nishiyama, N. Kato, and K. Hamaguchi, “On joint optimal placement of access points and partially overlapping channel assignment for wireless networks,” in *2014 IEEE Global Communications Conference*, 2014, pp. 4922–4927.

[10] Y. Matsumura, S. Kumagai, T. Obara, T. Yamamoto, and F. Adachi, “Channel segregation based dynamic channel assignment for WLAN,” in *2012 IEEE International Conference on Communication Systems (ICCS)*, no. 1, 2012, pp. 463–467.

[11] V. Gazis, K. Sasloglou, N. Frangiadakis, P. Kikiras, A. Merentitis, K. Mathioudakis, and G. Mazarakis, “Cooperative communication in channel assignment strategies for IEEE 802.11k WLAN systems,” in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013, pp. 1924–1929.

[12] M. Seyedebrahimi, F. Bouhafs, A. Raschella, M. Mackay, and Q. Shi, “Fine-grained radio resource management to control interference in dense wi-fi networks,” in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[13] F. den Hartog, A. Raschella, F. Bouhafs, P. Kempker, B. Boltjes, and M. Seyedebrahimi, “A pathway to solving the wi-fi tragedy of the commons in apartment blocks,” in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, 2017, pp. 1–6.

[14] P. Erdos and A. Renyi, “On random graphs i,” *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.

[15] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical review E*, vol. 76, no. 3, p. 036106, 2007.

[16] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, pp. 35–41, 1977.

[17] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proc. of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[18] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[19] D. G. Luenberger, *Introduction to linear and nonlinear programming*, 1973, vol. 28.

[20] W. P. Adams and H. D. Sherali, “Linearization strategies for a class of zero-one mixed integer programming problems,” *Operations Research*, vol. 38, no. 2, pp. 217–226, 1990.

[21] K. Van Oost and K. Van Doorselaer, “Method for testing a wireless link of a wi-fi node, and circuit performing the method,” Patent US 20 160 226 740 A1, 04 08, 2016. [Online]. Available: http://www.freepatentsonline.com/y2016/0226740.html

[22] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[23] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, “Least angle regression,” *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[24] A. Belloni, V. Chernozhukov *et al.*, “Least squares after model selection in high-dimensional sparse models,” *Bernoulli*, vol. 19, no. 2, pp. 521–547, 2013.

[25] K. Van Doorselaer, K. Van Oost, and N. Godman, “Method for evaluating a wireless link, respective device, computer program and storage medium,” Patent US 20 180 077 591 A1, 03 15, 2018. [Online]. Available: http://www.freepatentsonline.com/y2018/0077591.html

[26] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of computer computations*, 1972, pp. 85–103.