World Academy of Science, Engineering and Technology
International Journal of Information and Communication Engineering
Vol:17, No:1, 2023

Low Overhead Dynamic Channel Selection with Cluster-Based Spatial-Temporal Station Reporting in Wireless Networks

Zeyad Abdelmageid, Xianbin Wang

Abstract—Choosing the operational channel for a WLAN access point (AP) in WLAN networks has been a static channel assignment process initiated by the user during the deployment process of the AP, which fails to cope with the dynamic conditions of the assigned channel at the station side afterwards. However, the dramatically growing number of Wi-Fi APs and stations operating in the unlicensed band has led to dynamic, distributed and often severe interference. This highlights the urgent need for the AP to dynamically select the best overall channel of operation for the basic service set (BSS) by considering the distributed and changing channel conditions at all stations. Consequently, dynamic channel selection algorithms which consider feedback from the station side have been developed. Despite the significant performance improvement, existing channel selection algorithms suffer from very high feedback overhead. Feedback latency from the STAs, due the high overhead, can cause the eventually selected channel to no longer be optimal for operation due to the dynamic sharing nature of the unlicensed band. This has inspired us to develop our own dynamic channel selection algorithm with reduced overhead through the proposed low-overhead, cluster-based station reporting mechanism. The main idea behind the cluster-based station reporting is the observation that STAs which are very close to each other tend to have very similar channel conditions. Instead of requesting each STA to report on every candidate channel while causing high overhead, the AP divides STAs into clusters then assigns each STA in each cluster one channel to report feedback on. With proper design of the cluster based reporting, the AP does not lose any information about the channel conditions at the station side while reducing feedback overhead. The simulation results show equal performance and at times better performance with a fraction of the overhead. We believe that this algorithm has great potential in designing future dynamic channel selection algorithms with low overhead.

Keywords—Channel assignment, Wi-Fi networks, clustering, DBSCAN, overhead.

I. Introduction

DURING the last decades, wireless networks have consistently played a critical role in driving the evolution of many technologies. In addition to cellular networks, one of the most significant wireless technologies is Wi-Fi, which operates in unlicensed bands. Nowadays, Wi-Fi, defined by a family of 802.11 standards, is the most popular wireless technology used for data transmission, carrying more than half of user traffic today. With the different standards over the years, the data rates supported by Wi-Fi has increased from 2 Mbps in IEEE 802.11-1997 to almost 10 Gbps in the latest 802.11ax [1]. This significant performance improvement could be traced back to faster modulation, wider channels, and the implementation of Multiple Input Multiple Output technologies (MIMO).

One of the dominant impairments in WLAN networks is interference, because all WLAN devices share the same unlicensed band as well as with other technologies such as Bluetooth, ZigBee, and LTE-U. Furthermore, due to the increase in the number of WLAN networks, co-channel interference has become a huge challenge in Wi-Fi systems. Mitigating such interference could be extremely difficult due to spatially distributed interference, time-varying transmission activities and uncooperative nature among co-existing APs and STAs.

One of the main factors which can dramatically impact the performance of a Wi-Fi network is the channel of operation assigned to the AP, which is then used by the AP to communicate with STAs. The assignment process of the channel of operation is very important since assigning a channel with high interference to an AP will directly reduce the performance and achievable data rate for each STA. In existing channel assignment procedures, the channel is usually assigned to the AP by the user during the initialization process. In addition to lack of user friendliness, this procedure is fundamentally a static channel assignment, which does not consider the dynamic interference and distributed channel conditions of WLAN networks. In other words, this initial channel assignment remains the same unless the user chooses to change it. With dynamic channel use by neighbouring APs and STAs, this means that the initial channel assignment could be no longer optimal due to the time-varying and spatial distributed nature of co-channel interference in the unlicensed band.

In addition, the AP does not request channel conditions feedback from STAs before channel assignment. Without collecting feedback from the STAs in the current BSS, the APs also can not take into account the channel conditions at all STAs. This means that the AP is highly likely to be assigned to operate in a channel where the STAs are experiencing severe co-channel interference while the AP is completely unaware of the poor service the user is experiencing. As a result, measurement experiments conducted by Akella et al. [2] show that most of the deployed APs transmit on the same channel in the ISM band (channel 6) and only about 14 % of APs use the remaining two non-overlapping channels. This means that a lot of APs with overlapping coverage are not configured properly to operate on different channels to avoid interference, which will result in a poor user experience. This is a significant

Zeyad Abdelmageid is with Western University, Canada (e-mail: zhussei@uwo.ca).

issue since the main goal of the AP is to provide the best user experience possible.

The suboptimal performance of current channel assignment and utilization practice have inspired studies for dynamic channel selection algorithms which give the AP the ability to reconfigure itself overtime depending on the channel conditions. However, the APs involved in these studies usually make channel selection directly based the observed channel conditions by the APs themselves without considering the STAs' feedback. Very few of these algorithms give the AP the ability to request feedback from STAs to address the distributed channel condition and improve the overall performance of the Wi-Fi network. Authors in [3] demonstrated performance improvement using an algorithm called CB-AIIM (named after the initials of the authors) when STAs' feedback on channel conditions was taken into account by the AP when selecting a channel for operation. The improvement in performance was expected since the feedback from STAs allows the AP to have more accurate understanding of real-time channel conditions of all STAs. Despite its great performance in [3], one main issue with the algorithm is the high overhead associated with the feedback mechanism. Due to the very dynamic nature of STA channel conditions, the channel selection at AP has to be time-responsive with low latency to ensure the appropriateness of chosen channel. When the AP takes a long time to collect feedback from STAs, the channel conditions collected could already have become completely outdated. In addition, CB-AIIM lacks an adaptive post-channel sustainability procedure to keep the AP updated about possible changes in the select channel's conditions due to the dynamic nature of the wireless environment. These challenges have inspired us to develop a dynamic channel selection algorithm with reduced overhead through a new cluster-based station reporting mechanism (we will refer to it as CB-DCS from now on) where:

1) AP implements a clustering algorithm among STAs, utilizing the spatial-temporal nature of interference to group close-by STAs with similar channel environment conditions together. (Goal 1)
2) AP divides the workload of reporting on all candidate channels among STAs in each cluster. This way, the AP can reduce the feedback overhead while preserving the quality of feedback received, due to the fact that STAs in the same cluster have very similar environments and interference conditions. (Goal 2)
3) The AP has the ability to adapt to the dynamic channel conditions and change channel of operation if necessary. (Goal 3)

The paper will be divided into the following sections: Section II provides a brief literature review; Section III discusses the novelty and technical challenges; Section IV discusses the system model; Section V showcases our proposed algorithm design; Section VI analyzes the algorithm, its strengths and weaknesses; Section VII shows simulation results and comparisons between CB-DCS and CB-AIIM; Section VIII is our conclusion and summarizes the contribution of our work. We believe that our algorithm achieves similar performance to CB-AIIM while reducing the overhead and STA workload compared to CB-AIIM.

II. Literature Review

Due to its importance for overall network performance, channel selection in Wi-Fi has been studied extensively by the research community but most of them ignore the channel conditions at the STAs and only consider the channel conditions observed by the APs. The authors in [3] provided a literature survey on the articles published on heuristic techniques for channel selection algorithms in 802.11 networks.

In [4], the authors propose an iterative procedure to assign channels to APs (in an un-coordinated, multiple AP environment) ordering them depending on the number of APs that can interfere with each other. In [5], researchers summarized the problem of channel assignment as minimization of the interferences, which is a highly complex problem. Due to the high complexity of the problem, authors proposed MICA (minimum interferences for channel allocation) as an approximated, heuristic algorithm.

In [6], the authors proposed a Network-controlled channel allocation scheme called AIIM (named after the initials of the authors) to improve the performance of the AP that has the lowest performance without compromising the global performance and iterating the procedure. Authors in [1] took the AIIM as the approach to be the basis of their algorithm the CB-AIIM (Cluster Based AIIM), which introduced clustering into the algorithm where an AP and all its associated STAs are considered as a cluster and the AP selects the minimum out of all SIR values reported back by each station to be the SIR of the channel being considered. This process is repeated over each channel and then the channel with the highest SIR/Utility (normalized SIR) is selected. After this entire process is repeated for one AP, the process continues for the rest of the APs and the utility of this entire channel assignment is computed called S. Afterwards, APs with utility lower than 1 are selected to try to assign them other channels which would yield better utility but would not degrade S. This process is repeated R times. This algorithm, however, is suitable for Wi-Fi networks with a central coordinator.

In [7] a heuristic channel assignment algorithm for uncoordinated WLANs, CACAO, was proposed which considers the STA side. In the that algorithm, the AP auto-configures their channels depending on their local traffic information; such feedback was obtained using the 802.11k standard for radio resource management, which defines a series of measurement requests and statistical reports between an AP and its clients. The AP would periodically query each associated client to collect reports on traffic statistics for each channel. This allows the AP to dynamically reconfigure itself and choose the best channel for operation. As we can see, most of these algorithms are assuming that the sources of interference are other Wi-Fi devices. In addition, the overhead in some of these algorithms which consider STA side like [6] and [1] involves high STA workload and large overhead which increases with size of associated STAs. In addition, these algorithms do not consider the dynamic temporal nature of the Wi-Fi environment. In other words, the algorithms do not consider that the STAs are mobile and thus do not clearly discuss how often the channel assignment process needs to be re-done in order to make sure the AP receives up-to-date feedback from the STAs.

As we can see in Fig. 2, CB-AIIM outperforms the KCKC algorithm in [8], an algorithm which chooses channel with least interference based on Wi-Fi control frames, HZNA, an algorithm which attempts to minimize using the same channel in neighbouring APs while trying to avoid total number of channels used, in [9] and AIIM in [6]. Consequently, we will be comparing our algorithm to CB-AIIM.

III. Novelty and Challenges

In this section we highlight the novelty and technical challenges that we faced during the creation of the proposed algorithm. The novelty of the proposed channel selection algorithm can be summarized as:

1) The proposed channel selection algorithm gives the AP the ability to dynamically choose a channel for Wi-Fi network operation based on its changing operational environment over time.
2) In addition to the above channel selection process, a low overhead, cluster-based STA feedback mechanism is implemented in order to reduce the feedback overhead without sacrificing the quality of information received at the AP by utilizing the spatial-dependence of the interference.
3) Lastly, we propose a cluster-based, low-overhead channel sustainability procedure to ensure that the AP is aware when channel conditions change in order to adapt efficiently.

The use of the proposed clustering algorithm will be explained in more details further in the paper. With the overall goal in mind, we faced some technical challenges summarized as:

1) Choosing a suitable clustering algorithm to support the fast, low-overhead channel selection without losing the channel information
2) Fine-tuning the parameters of the clustering algorithm such that STAs belonging to the same clusters are guaranteed to have very similar channel conditions
3) Ensuring that the AP requests feedback from the STAs such that the AP has all the information that it needs to select a channel while keeping overhead at a minimum
4) Establishing a mechanism by which the STAs can report channel estimation information back to the AP which would be backwards compatible with previous versions of 802.11
5) Designing a low overhead post-channel selection feedback procedure in order to keep the AP updated on the dynamic channel conditions and to adapt if necessary

IV. System Model

A. Architecture

The Wi-Fi operating mode we are considering in this report is the infrastructure mode, which involves two different types of Wi-Fi devices i.e., an APs and multiple STAs in one BSS. In addition, the infrastructure mode allows communication between all the STAs and their associated AP. However, communication among STAs is forbidden.

TABLE I
Positions of Node in (x,y)

Node Label	Node Position
AP	(0,0)
1	(1,0)
2	(1,0.5)
3	(0.5,0.5)
4	(-1,0)
5	(-1,0.4)
6	(-0.75, 0.25)

In this paper, we focus on the 2.4 GHz unlicensed band. The AP can choose any of these channels in supporting its operation. Among all available channels in this band, only three channels, i.e. 1, 6, and 11, are non-overlapping. We are only considering the channel selection among these three non-overlapping channels, as that approach is widely accepted. As a result, if two neighboring Wi-Fi networks are operating in different channels, they would not interfere with each other.

B. Channel Model

The Wi-Fi channel was modelled using the “networkx” Python library. “networkX” is a powerful Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Each node has four main features: Label, Position, Edge and Weights. Each node in this graph is given as:

1) A label either as an “AP” or a number if that node is a STA
2) A position in the (x,y) space (we assumed all nodes are the same height)
3) An edge which represents connection between nodes
4) The weight of the edges nodes which represent the received signal strength (RSS) at the nodes

When two nodes, A (an interferer) and B (a STA device), are operating over the same channel, an edge with a weight representing the interference power at node B caused by node A will be present. We can see in Fig. 1 an example where the nodes have the labels and positions in Table I.

As we can see in Fig. 1, the weight labeled on each edge represents the RSS at each STA from the associated AP. The edge-weights between the AP and any of its associated nodes is represented as:

$$I_{AP->\ node} = \ P_t + G_t + G_r - \ L \ - \ P_{Loss} \qquad (1)$$

where P_t is the transmission power, G_r and G_t are the receiver and transmitter antenna gains, L is the propagation losses due to obstacles and, assuming that all the antennas are placed 1.5 m above the floor, P_{Loss} represents the signal loss due to distance which has been formulated in [10] for the 2.4 GHz band and is formulated as:

$$P_{Loss} = 40 + 20log\left(d\right) \qquad (2)$$

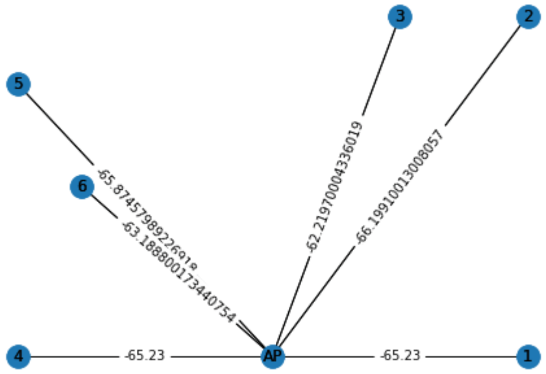

Fig. 1 Networkx Graph of Network

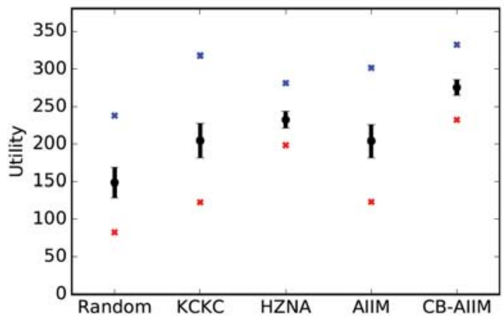

Fig. 2 Performance analysis of channel selection algorithms

where d is the distance between the two nodes in meters. On the other hand, the weight of the edges between an interfering node “m” and any other STA node “n” is defined as:

$$I_{m->\ n} = I * (P_t + G_t + G_r - \ L \ - \ P_{Loss}) \qquad (3)$$

where I is zero if m and n do not operate on the same channel. Keep in mind that STAs which belong to the same AP do not interfere with each other as the AP coordinates the transmissions. In unlicensed communications, the dominant impairment is the co-channel interference while the background and thermal noise can be neglected. Consequently, communication performance at Wi-Fi AP and STAs is determined by the ratio between the desired signal and the combined co-channel interference from neighboring nodes. To determine the severity of the co-channel interference, the signal to interference ratio (SIR) for station “n” could be expressed using (4):

$$SIR_n = \frac{S_{AP\ ->\ n}}{\sum_k \ I_{m\ ->\ n}} \qquad (4)$$

where $S_{AP->n}$ is the desired signal received from the AP at node ”n”.

V. Proposed Algorithm Design - CB-DCS

A. *CB-DCS Algorithm Assumptions*

Now we can finally present our cluster-based channel selection algorithm for uncoordinated networks which we will refer to as CB-DCS. Before we do so, we must list some assumptions:

1) The AP is operating in the 2.4 GHz band and is only considering channels 1, 6, and 11 for operation
2) APs are capable of operating on multiple channels simultaneously and supports UL MU-MIMO
3) The AP can locate STAs in space
4) The number of channels that the AP will be considering for operation is: x
5) The AP and STAs are all in the same z plane (they have the same height from the ground)

B. *Clustering Algorithm*

Firstly, we present the clustering algorithm which will be essential to our low overhead STA reporting algorithm. Considering the spatial nature of interference, a very important observation to note is that STAs in close proximity of each other tend to have very similar channel conditions; therefore, the main goal of the clustering algorithm is to make sure that STAs in the same cluster have very similar channel conditions. Grouping STAs with similar channel conditions will help reduce feedback overhead dramatically. For example, let’s consider three STAs which are very close to each other and, therefore, have very similar channel conditions. If the AP wants to know the conditions of channels 1, 6, and 11 in the approximate location of the three STAs, each STA would have to scan all three channels and report back to the AP. On other hand, if the AP groups these three STAs into a single cluster, each STA could just report on one of the channels as the conditions will be very similar to the those of the other two STAs in the same cluster. This way the AP knows the channel conditions at all the STAs in the cluster while only requiring each STA to report on one channel.

One of the most popular density based clustering algorithm is DBSCAN [11], which is available in the python library ‘sklearn’. The algorithm takes in two parameters: ϵ (the radius) and ”min samples” (the minimum number of points needed to define a core point). We selected this clustering algorithm as it has the ability to form clusters with arbitrary shapes, without a limitation on the cluster sizes and can handle outliers/noise in a satisfactory manner. The steps of the algorithm are as follows:

The steps of the algorithm are as follows:

1) A random point is chosen
2) A circle of radius ϵ is drawn around the chosen point as the center
3) If the number of sample points in the circle is at least equal to ‘min_samples’, the point is a core point
4) If there are no other points in the circle, the point is an outlier
5) This process is repeated across all points in the dataspace
6) Clusters are then formed based on the core, border, and outliers

A Python simulation using the DBSCAN algorithm was carried out and the results are shown below. The value of epsilon was set to 2 and ‘min_samples’ to 2. Fig. 4 shows the dataset plotted in space and the result of the DBSCAN

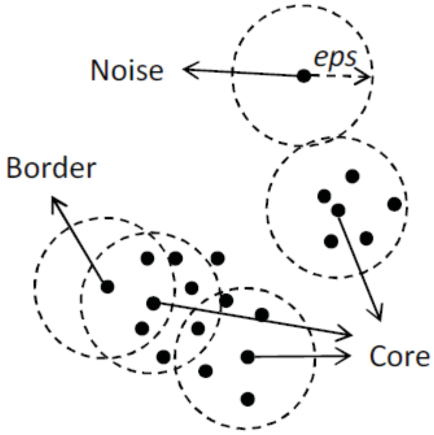

Fig. 3 DBSCAN algorithm overview

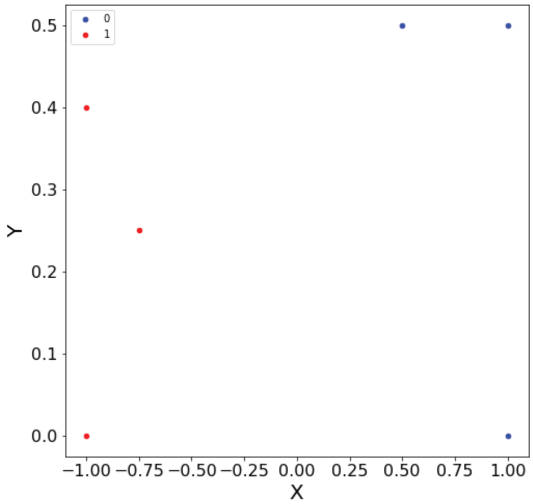

Fig. 4 STAs after being divided into clusters; each color/index corresponds to a different cluster

algorithm, where nodes with the same label belong to the same cluster and nodes with a label '-1' are outliers which do not belong to any cluster.

Now that we have outlined the clustering algorithm of choice, DBSCAN algorithm, we have to specify which parameters will be considered in the clustering process.

The first parameter being considered is the location of STAs in space. This will ensure that STAs in close proximity of each other are grouped into one cluster as they will have similar channel conditions (SIR value) due to the spatial-temporal nature of interference. We can see the result of running the DBSCAN algorithm (shown in Fig. 4) on the nodes (shown in Fig. 1). We can see that each cluster index is represented by a color and '-1' indicates a node that does not belong to a cluster.

The second parameter we considered in the clustering process is the SIR for a specific channel at the STAs. The AP requests the SIR measurement across the same channel/RU from all STAs. The AP then applies the DBSCAN algorithm on the SIR values reported back from all STAs. This can be seen in Fig. 5.

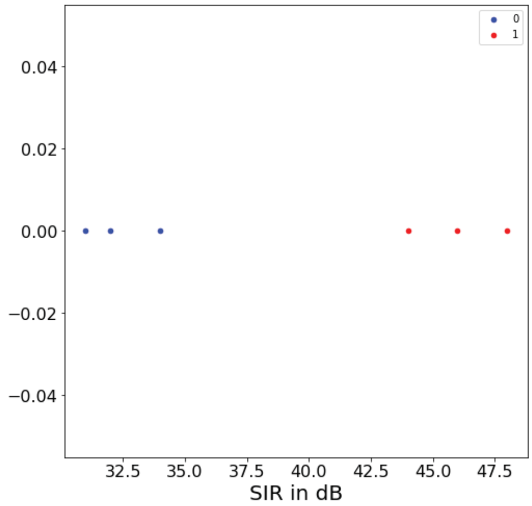

Fig. 5 SIR clustering

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Extended Range	False
Channel Bandwidth	20 MHz
APEP Length	1000 bytes
MCS	0
Number of Spatial Streams	1
Number of Transmit Antennas	1
Sampling Frequency	20 MHz
Carrier Frequency	2.4 GHz
Large Scale Fading Effect	Pathloss and Shadowing
Number of Receive Antennas	1
Number of Penetrated Walls	Variable
Number of Penetrated Floors	Vairable
Distance between Transmitter and Receiver	Variable

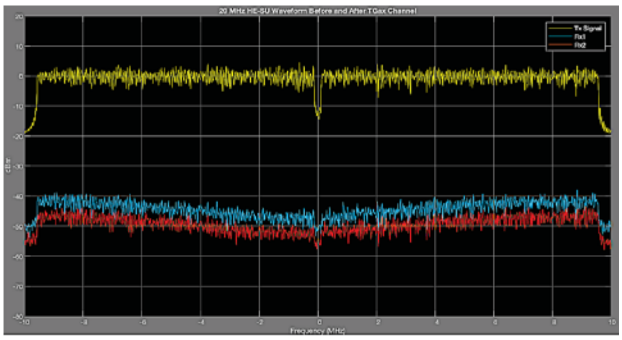

Fig. 6 rx2 (blue): NumPenWalls = 1, NumPenFloors = 1, TxRx2 distance = 2 m

The purpose behind including a second parameter is that sometimes, despite two STAs being close to each other, they can have different channel conditions. This could be due to a barrier between these STAs like a wall or a piece of furniture, which the AP has no way to identify. Simulations were run on MATLAB in order to investigate the nature of an indoors Wi-Fi channel. The WLAN toolbox was used. A HE-SU packet was filtered through a 802.11ax multipath fading channel. The simulation parameters are summarized in Table II.

As shown in Fig. 7, despite both receivers being only 1

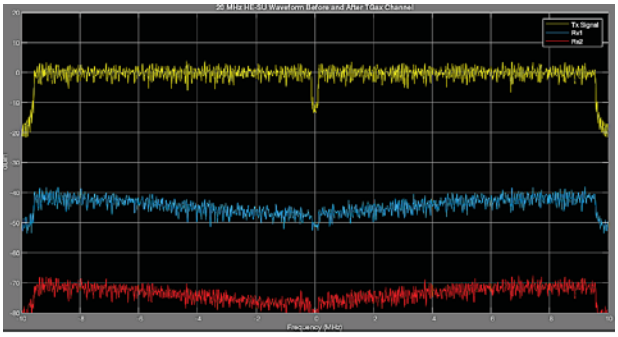

Fig. 7 rx2(blue): NumPenWalls = 1, NumPenFloors = 0, TxRx2 distance = 1 m

meter apart, the wall and floor in between caused a reduction of up to 30 dB at rx2, compared to the case in Fig. 6. This could work the other way around as well. For example, rx2 could be experiencing interference from an interferer but rx1 would not detect this interference due to the attenuation caused by the wall and floor. Now if the clustering is done only based on the spatial location, then the AP would group rx1 and rx2 in the same cluster because the AP has no way to tell if there is a wall between these two receivers. The fact that the AP would cluster rx1 and rx2 together means that the AP would assume that they both have similar conditions, which is not the case. For this reason, we add another parameter in the clustering algorithm: SIR. The AP will start by assigning all channels in the same spatial cluster to report back the SIR of a common channel. Now the AP runs the DB-SCAN algorithm again on the reported back SIR values resulting in a set of SIR-based clusters. As a result, STAs A and B belong to the same cluster if and only if:

1) STA A and STA B belong to the same spatial cluster
2) STA A and STA B belong to the same SIR-based cluster

After the AP clusters the STAs based on their spatial location and their channel conditions, the STAs in the same clusters will have very similar channel conditions. This lays the foundation for our algorithm's ability to achieve 'Goal 1': Reducing feedback overhead.

C. STA Feedback Mechanism

One of the main aspects in need of discussion is the mechanism by which the STAs report the SIR to the AP. Since the main aspect of the CB-DCS is the STA reporting back to the AP, we need to very clearly specify how this process will be carried out. In 802.11ax, MU-MIMO was introduced to the standard and is a key technology responsible for increasing the capacity of the network. This procedure, however, requires the AP to know the channel conditions at the STA. Therefore, a channel sounding procedure was developed. Since the channel conditions change frequently, the AP needs to update its knowledge about the STAs channel conditions periodically. The sounding procedure starts by the AP sending a reference signal to the STAs and then getting explicit channel feedback from the STAs. The procedure starts as follows [12]:

1) AP sends out a null data packet announcement (NDPA) to notify STAs about the following reference signal

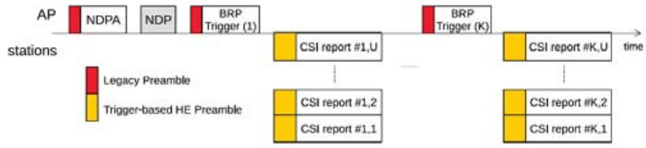

Fig. 8 Explicit Channel Sounding

2) After a short inter-frame space, AP sends a NDP frame, which is used by the STAs to assess the channel.
3) AP sends a beamforming report trigger to the STAs to let the them know that they need to transmit back a beamforming report to the AP
4) STAs report back to the AP the beamforming reports (BFRs) either sequentially or in parallel using OFDMA

The BFR/CSI report contains the average SNR per spatial stream. The MU-MIMO in 802.11ax allows AP to serve a maximum of 4 STA per RU [13]. The frame format of the BRP can be seen in Fig. 9, where the 'RU Allocation' in the 'User Info' field is the RU which the AP is requesting feedback on. The whole sounding process is shown in Fig. 8. Due to the fact that this procedure is already implemented in 802.11ax, we decided that this is the mechanism that the AP will use to request and receive feedback from STAs.

D. STA Channel Assignment

In order for the AP to take advantage of the clustering algorithm, the AP must have the capability to intelligently assign STAs channels to scan such that the overall feedback received by the AP is comprehensive and does not lack important information of the AP to know before selecting a channel for operation, while keeping the overhead to a minimum. Considering one cluster at a time, the AP divides the work among the STAs in such a way that provides the AP with feedback on all the candidate channels. The AP assigns the STAs the channels to report on. The number of channels assigned to each STA depends on the cluster size. If the cluster size is less than the number of the channels the AP is considering, then the AP would have to assign some STAs more than one channel to scan in order to have a complete understanding of channels conditions. In this case, the priority of assigning multiple channels to scan is for STAs with the highest capabilities. This information is known to the AP due to the association frames sent by STAs when joining the network. This ensures that devices with limited resources, such as IoT devices, are not burdened with additional workload. In the case where all the STAs have the same capabilities, then the AP would assign the extra channels to STAs randomly. For example, if the cluster consists of just two STAs of equal capabilities, then the AP would randomly assign one of the STAs two channels to scan while the other STA would just be assigned the remaining channel to scan. On the other hand, when the cluster size is bigger than or equal to the number of channels the AP is considering, the AP assigns each STA a single channel to scan. If the cluster size is equal to the number of channels, then the AP assigns each STA a channel to report on. On the other hand, the AP would assign multiple

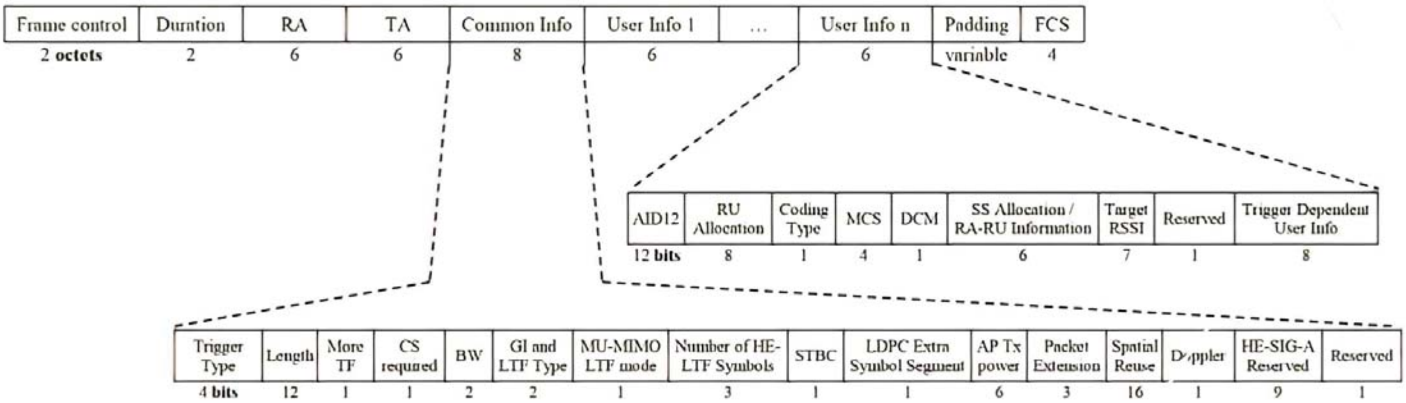

Fig. 9 BRP Frame Format

STAs the same channel to scan then calculate the mean of the results in further calculation in (5). For example, if the cluster is of size three, then AP would assign STA 1 to report on channel 1, STA 2 to report on channel 6 and STA 3 to report on channel 11. If the cluster size is bigger than three then STA 4 would be assigned to scan channel 1, STA 5 would be assigned to scan channel 6, etc...

E. Data Processing and Channel Selection

After the AP receives feedback from all clusters, the AP would have a SIR value corresponding to each channel from each cluster. The way the AP processes this data is crucial. A weakness in algorithms proposed in recent literature is sensitivity to outliers; therefore, we aim to make our algorithm less sensitive to outliers. As a result, for each channel, the final SIR value is a weighted average of the SIR values reported back by different clusters where the weights correspond to the size of clusters. The weighted average gives more weight to denser clusters and less weight to less dense clusters, as can be seen in (5). This makes the algorithm less sensitive to outliers, unlike the CB-AIIM algorithm. Now the AP selects the channel with the highest average SIR.

$$SIR_x = \frac{\sum_N NumSTA_n \bullet SIR_n}{Total\ Number\ of\ STAs\ in\ Network} \qquad (5)$$

where $NumSTA_n$ is the total number of STAs in cluster n, SIR_n is the SIR for channel x in cluster n (reported back by the STA assigned to channel x) and SIR_x is the total SIR for channel x.

F. Post Channel Selection Sustainability

After the AP selects the channel of operation, it is very crucial for the AP to be able to adapt to the dynamic nature of the wireless environment and to update any outdated information the AP has about the channel conditions. In the sounding procedure implemented in MU-MIMO in the 802.11ax standard, the AP repeats the procedure periodically every 50-200 ms depending on the channel conditions. Despite the fact that the AP stays updated on the latest channel conditions, this process could be extremely inefficient in low-mobility networks. This inspired us to design our very own procedure to keep the AP aware of the dyamic channel conditions and, therefore, give it the ability to adapt consequently. In each cluster, the AP assigns the STA with the highest capability to be the cluster head. This cluster head monitors the channel conditions and triggers the AP to re-select a channel of operation if and only if the channel conditions changes over a period of time. This way unnecessary reporting when the channel conditions have not changed is avoided, reducing overall overhead and workload.

In addition to the dynamic nature of the wireless environment, the AP also has to account for the possible mobile nature of the STAs. The AP periodically estimates the location of the STAs and runs the DBSCAN algorithm on their locations and as soon as they detect that there is a new cluster forming or the presence of changes to the current clusters, the AP must perform the channel selection algorithm all over again. The periodicity at which the AP estimates the location of the STAs depends on the nature of the network. Networks with low mobility would not need very frequent estimations by the AP; therefore, the period will not be specified and is a parameter that can be adjusted.

One important aspect to consider, however, is the fact the SIR values estimated by cluster heads or the estimated locations of the associated STAs might be affected by noise due to environmental factors. In order to eliminate the effect of noise, we use the Hampel filter for denoising to get more accurate data samples [12], [13]. The basic idea of the Hampel filter is a moving, weighted average. The Hampel filter was mainly developed in order to remove outliers and singular data points from datasets and it does so effectively. The filter judges a point as either singular or correct. When the point is judged as singular, the data point is replaced by the median. The judgement equations of the filter are as follows:

$$\begin{cases} |x-m| > \text{t*std} & singular \\ |x-m| \leq \text{t*std} & correct \end{cases}$$

After the Hampel filter is applied and the singular points are eliminated, the data set becomes more reliable. Once a cluster-head notices an upward or downward trend in the data, then the AP assumes that channel conditions have changed and that the channel of operation might not be the optimal selection anymore and triggers the AP to re-select a channel; therefore, the AP runs the algorithms again. On the other hand, if there is no trend in the SIR values, then no triggering action is taken by the cluster-head.

Algorithm 1 CB-DCS Pseudocode

```
1:  AP sends frames to STA to estimate common channel
2:  AP divides STAs into n cluster based on Spatial location
    and SIR reported back in step 1
3:  for <each cluster n> do
4:      AP assigns clusterhead to STA with highest capabilites
    or randomly if not applicable
5:      if size(n) ≥ x then
6:          for (each STA m in Cluster n) do
7:              if m >x and mod(m,x) = 0: then
8:                  m ← x^th channel
9:              else if m >x and mod(m,x) ≠ 0: then
10:                 STAm ← mod(m,x)^th channel
11:             else if m <x: then
12:                 STA m ← m^th channel
13:             end if
14:         end for
15:     else if size(n)<x then
16:         for <each STA m in Cluster n> do
17:             if STA m is a high capability STA then
18:                 assign the STA the m^th channel
19:                 assign the STA the excess channels to scan
20:             else if STA m is not a high capability STA then
21:                 assign the STA the m^th channel
22:             end if
23:         end for
24:         if Still some channels are unassigned then
25:             randomly assign STAs the excess channels
26:         end if
27:     end if
28: end for
29: AP requests feedback from clusterheads and estimates
    position of all STAs
30: while feedback and positions do not change do
31:     Nothing
32: end while
33: if feedback or positions change then
34:     Go back to Step 1
```

VI. Evaluation

Now we will evaluate how the CB-DCS algorithm meets the goals we set in Section I and its strengths and weaknesses.

A. Algorithm Analysis

The first goal is AP implements a clustering algorithm among STAs, utilizing the spatial-temporal nature of

Algorithm 2 post-channel selection sustainability Algorithm

```
Cluster-head periodically measures SIR of the selected
channel by the AP
AP periodically measure positions of STAs
Hampel filter is applied on measured data points
while channel conditions and positions do not change do
    Nothing
end while
if channel conditions change then
    cluster-head triggers AP to re-select channel
else if STA position changes then
    AP re-selects channel
```

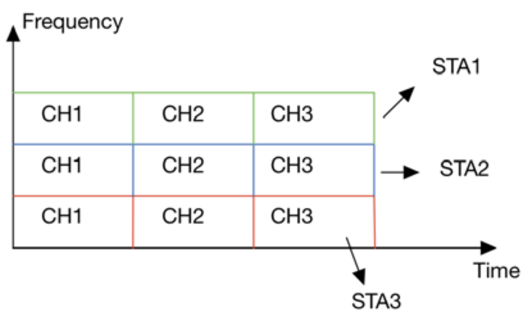

Fig. 10 Feedback without clustering

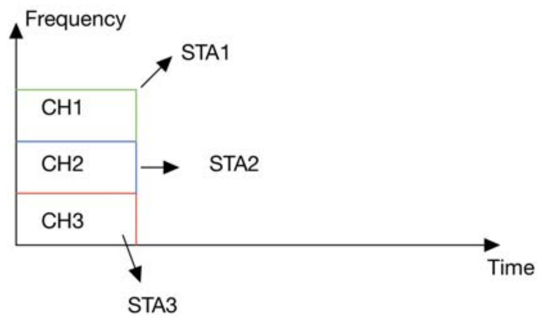

Fig. 11 Feedback with clustering

interference to group STAs with similar channel environment conditions together. Let's refer to steps 1-4 of 'Algorithm 1': CB-CDS Pseudocode'. The AP begins by dividing the STAs into clusters based on the chosen parameters: spatial coordinates and SIR at the STA for a specific channel. The result of these steps is clusters where STAs in each cluster have almost identical channel conditions, which is essential for STA feedback overhead reduction. Afterwards, the AP takes advantage of this in steps 3-28 in order to reduce the feedback overhead. The AP just assigns each STA in the cluster channel/s to scan and collects the feedback from all the STAs in the cluster to have complete knowledge about the channel conditions across all the candidate channels in the vicinity of the cluster. This process can be seen in Figs. 10 and 11, where STAs 1, 2, and 3 belong to the same cluster. We can see the dramatic decrease in overhead in Fig. 11 compared to Fig. 1 after applying clustering and more importantly, the AP does not sacrifice any information.

The second goal we set is AP divides the workload of reporting on all candidate channels among STAs in each cluster. This way, the AP can reduce the workload per STA and feedback overhead dramatically while preserving the quality of feedback received, due to the fact that STAs in the same cluster have very similar environments and interference conditions. This way, the AP can reduce the feedback overhead dramatically while preserving the quality of feedback received, due to the fact that STAs in the same cluster have very similar environments and interference conditions. The algorithm makes sure that the AP receives feedback from clusters on the set of all candidate channels being considered. This can be seen in Fig. 11. The AP assigns each STA in the cluster one channel to report on. This way, the AP will have information about all three channels despite each STA reporting back on just one channel. This is due to the intelligent division of workload on the STAs. Steps 3-28 in 'Algorithm 1' show the details of how the workload is divided among STAs in a cluster.

Lastly, the third goal we set is: The AP has the ability to adapt the dynamic channel conditions and change channel of operation if necessary. This goal is important to meet due to the mobile, dynamic nature of WLAN environments. Keeping the AP updated on channel conditions across all STAs while keeping overhead low is a challenge which was overcome by: 'Algorithm 2', which selects a cluster-head in each cluster to trigger the AP to re-select a channel of operation if and only if the current channel of choice is not suitable anymore. In addition to channel conditions, the position of the STAs can change overtime due to the mobile nature of STAs. As a result the AP periodically estimates the positions of the STAs in the network and is triggered to re-select the channel of operation if the positoins of the STAs change enough to form new cluster or change current clusters. An issue which could arise is noisy data points which would trigger the AP to take action when in reality it is not necessary, so in order to eliminate the noise from the data measured by the cluster-heads or the AP, a Hampler filter is applied which can effectively eliminate singular data points. As a result, the resulting dataset is more reliable and can be utilized by the cluster-heads and AP to make decision on whether the current channel is still a suitable selection or if it needs to alert the AP.

B. Algorithm Strengths and Weaknesses

Lastly, let's consider which scenarios are the most suitable for the CB-DCS algorithm. The CB-DCS algorithm excels when the STAs in the network are dense and can form clusters of size at least greater than the number of channels the AP is considering. For example, if the AP is considering selecting one of three channels to operate in, the smallest cluster would have to be at least of size 3. This way, the AP would receive at least one feedback on each channel from that cluster. Otherwise, the AP would need to assign a STA more than one channel in order to have complete understanding of the channel conditions. This will eventually increase the feedback overhead but there would still be an overhead reduction compared to a cluster-free algorithm. In the worst case scenario where all STAs are single clusters

TABLE III
STAs Locations in (x,y)

STA ID	Coordinates
1	(1,0)
2	(1,0.5)
3	(1.5,0)
4	(-2,0)
5	(-2,0.75)
6	(-2.5, 0.5)
7	(-3, -3)
8	(-3.5,-3)
9	(-4,-3.5)

TABLE IV
Interferers Info

Interferer ID	Coordinates	Operational Channel
1	(2,-7)	1
2	(6,-3)	6

(clusters are all of size one), the feedback overhead would be equal to a cluster-free case.

One more thing to consider is scalability issues which might arise when applying this algorithm in the 5 and 6 GHz bands. In these bands, the number of non-overlapping channels is significantly higher than the 2.4 GHz band; therefore, if the AP considers every single channel and requests feedback on all these channels, the overhead feedback would increase dramatically. Even though this scalability issue is a concern, it is not an alarming issue as the co-channel interference in the 5 and 6 GHz bands is not as severe as it is in the 2.4 GHz band. As a result, less channels in the 5 and 6 GHz band will suffer from co-channel interference compared to channels in 2.4 GHz. Thus, if the AP operate in the 5 and 6 GHz bands, it does not have to consider all channels available for operation and it can instead consider a random subset of the set of all available channels. This will reduce the scaling issue with our algorithm while not compromising performance. On the other hand, one of the algorithm's strengths is its insensitivity to outliers. This can be seen clearly in (5), where the total SIR of a channel is calculated as a weighted average where the weights are proportional to the size of the cluster. This way, denser clusters will have a greater influence on the AP's channel selection more than a less dense cluster. This guarantees that the algorithm will select the channel which will improve the overall performance of the network the most.

VII. Simulations

We assume that in the network, the AP is only considering channels 1, 6 and 11. The STAs and AP have the coordinates in Table III.

Before we compare show simulation results, we will introduce three equations which calculate values related to overhead reduction, which is denoted by ζ.

$$\zeta_m = \frac{1}{x}(x - c_m) \times 100\% \tag{6}$$

$$\bar{\zeta} = \frac{\sum_m \frac{1}{x}(x - c_m) \times 100\%}{M} \tag{7}$$

TABLE V
FEEDBACK AT AP WITHOUT CLUSTERING (CB-AIIM)

STA ID	Channel 1	Channel 6	Channel 11
1	36.99	35.31	50.00
2	36.61	34.74	49.03
3	33.40	31.14	46.47
4	32.10	32.61	43.97
5	32.21	32.22	43.41
6	30.71	31.14	41.87
7	23.58	26.53	37.45
8	23.38	26.28	36.73
9	22.32	25.50	35.49
Average SIR per Channel	30.14	30.62	42.71

TABLE VI
FEEDBACK AT AP WITH CCLUSTERING

Cluster ID	Channel 1	Channel 6	Channel 11
1	36.99	34.74	46.48
2	32.10	32.33	41.87
3	23.58	26.28	35.49
Average SIR Per Channel	30.89	31.12	41.28

$$\zeta_{min} \leq \zeta_m \leq \zeta_{max} \quad (8)$$

where m is STA index, c_m is the number of channels assigned for STA m to report on, M is the total number of STAs in the network and x is the number of channels the AP is considering for operation: Firstly, (6) which calculates the overhead reduction of STA m; secondly, (7) which calculates the average overhead reduction across all STAs; lastly, (8) which calculates the range of overhead reduction in the network ζ_{min} and ζ_{max} can be calculated using (6) where STA m is replaced by STAs assigned the minimum and maximum number of channels respectively.

Now we will compare our algorithm to a cluster free algorithm like the steps outlined in Section IV subsection 'CB-AIIM'. In this algorithm each STA scans and reports all three channels. We added two sources of interference which can be seen in Table IV. We ran both algorithms to see which channel would they choose to operate in, how they would rank the channels and then compare the results. Firstly, we must specify that the channel with the best conditions is channel 11 since the interferers are not operating on that channel.

When running the algorithm without clustering (CB-AIIM), the AP gets the feedback information seen in Table V. The AP has the SIR value in dB for each STA across every channel. Looking at the data in Table V, we can see that channel 11 has the highest average SIR, followed by channel 6 then lastly channel 1. On the other hand, when we ran our algorithm, we can see the feedback in Table VI. The table shows the SIR values in dB reported back by each cluster for each channel. The channel with the highest SIR factor is channel 11 then channel 6 then channel 1, which is the same exact result as the first algorithm but with 66.67 % reduction in the overhead. Lastly, we tested the accuracy of the SIR values reported back by our low-overhead algorithm by comparing them to the SIR values reported back when every STA in the BSS reports back to the AP (these SIRs are denoted by 'actual' in (9)). We can see the equation for the percentage error in (9). The results are summarized in Table VII. The highest error recorded is less than 4%.

$$Error = \frac{|\mathrm{SIR}_{actual} - \mathrm{SIR}_{cluster-based}|}{\mathrm{SIR}_{actual}} * 100 \quad (9)$$

In addition, we simulated our algorithm over two different network scenarios: Scenario I and Scenario II, where scenario I is a low STA-density network and scenario II is a high-density network. The purpose of this simulation is to show that our algorithm performs better in denser networks, which is evident in table VIII. The biggest error reported in scenario I is 2.59% while the biggest error in scenario II is 1.56%. In addition, the average error for scenario I is 1.63% while the average error for scenario II is 0.87% which is half of the error in scenario I. These results show the great performance of our algorithm with a $\overline{\zeta}$ of 66.7% while maintaining very accurate results.

In addition, we simulated the Hampel filter on a data points which resemble the SIR values reported back by a cluster-head, which included some noise. The filter was successful in eliminating all the singular data points. This can be seen clearly in Fig. 12.

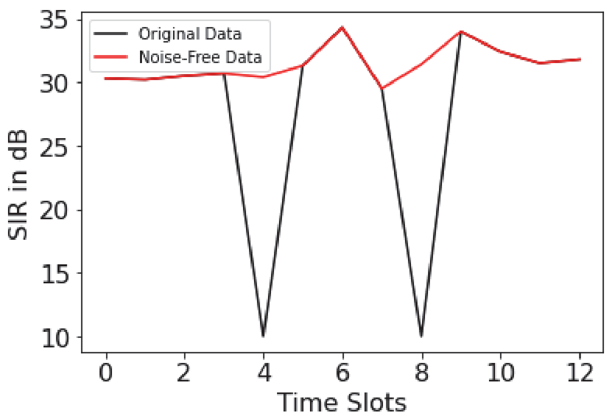

Fig. 12 Hampel Filter

VIII. CONCLUSION

To conclude, the problem of channel selection in Wi-Fi networks is a complex problem which has never been more relevant due to the increasing density of Wi-Fi technology and the growth of the applications of the technology as a

TABLE VII
ACCURACY TEST

Channel	Percentage Error
1	2.48%
6	1.60%
11	3.34%

TABLE VIII
DENSITY TEST

Density	Error on Ch1	Error on Ch6	Error on Ch 11	Avg Error
Scenario I	0.17%	2.59%	2.15%	1.63%
Scenario II	0.66%	0.41%	1.56%	0.87%

whole. In this paper, we propose CB-DCS: a low overhead, dynamic channel selection algorithm which is suitable for Wi-Fi networks without a central coordinator. In addition, it is suitable for Wi-Fi networks experiencing interference from non-WiFi devices. The algorithm implements a cluster-based STA reporting mechanism to utilize the observation that STAs in close proximity to each other tend to experience similar channel conditions. As a result, instead of each STA reporting on all candidate channels, the AP can divide this workload among STAs within the same cluster, which keeps the overhead to a minimum. Finally, we ran simulation results in order to compare our algorithm to already existing channel assignment algorithms and the results showed an overhead reduction of up to 67% with minimum sacrifice of accuracy. Despite the promising results, we believe there is still room for improvement. The future direction of our work revolves around dynamically tuning the parameters of the DB-scan clustering algorithm.

REFERENCES

[1] E.Khorov, I. Levitsky and I. F. Akyildiz, ”Current Status and Directions of IEEE 802.11be, the Future Wi-Fi 7,” in IEEE Access, vol. 8, pp. 88664-88688, 2020, doi: 10.1109/ACCESS.2020.2993448.

[2] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, “Self-Management in Chaotic Wireless Deployments,” Proc. ACM MobiCom, 2005.

[3] Gimenez-Guzman, Jose Crespo-Sen, David Marsá-Maestre, Ivan. (2020). A Cluster-Based Channel Assignment Technique in IEEE 802.11 Networks. Telecom. 1. 228-241. 10.3390/telecom1030016.

[4] Mahonen, P.; Riihijarvi, J.; Petrova, M. Automatic channel allocation for small wireless local area networks using graph colouring algorithm approach. In Proceedings of the 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No. 04TH8754), Barcelona, Spain,5–8 September 2004; Volume 1, pp. 536–539.17]

[5] Cui, Y.; Li, W.; Cheng, X. Partially overlapping channel assignment based on “node orthogonality” for 802.11 wireless networks. In Proceedings of the 2011 Proceedings IEEE INFOCOM, Shanghai, China, 10–15 April 2011; pp. 361–365. [18]

[6] Abeysekera, B.H.S.; Ishihara, K.; Inoue, Y.; Mizoguchi, M. Network-controlled channel allocation scheme for IEEE 802.11 wireless LANs: Experimental and simulation study. In Proceedings of the 2014 IEEE 79th Vehicular Technology Conference (VTC Spring), Seoul, Korea, 18–21 May 2014; pp. 1–5.

[7] Yue, X.; Wong, C.F.; Chan, S.H.G. CACAO: Distributed client-assisted channel assignment optimization for

[8] Kwon, Y.M.; Choi, K.; Kim, M.; Chung, M.Y. Distributed channel selection scheme based on the number of interfering stations in WLAN. Ad Hoc Netw. 2016, 39, 45–55.

[9] Handrizal, M.Z.; Noraziah, A.; Abdalla, A. An improved of channel allocation for wlan using vertex merge algorithm. In Proceedings of the International Conference on Computational Science and Information Management (ICoCSIM), Toba Lake, Indonesia, 3–5 December 2012; Volume 1, pp. 205–213.

[10] Green, D.B.; Obaidat, A. An accurate line of sight propagation performance model for ad-hoc 802.11 wireless LAN (WLAN) devices. In Proceedings of the 2002 IEEE International Conference on Communications, Conference Proceedings (ICC 2002 Cat. No. 02CH37333), New York, NY, USA, 28 April–2 May 2002; Volume 5, pp. 3424–3428.

[11] sWeb.Stanford.Edu, 2021, https://web.stanford.edu/class/cs345a/slides/12-clustering.pdf.

[12] Ji-Xian Zhang, Qiu-Hai Zhong, Ya-Ping Dai and Zheng Liu, ”A new de-noising method based on wavelet transform and transforming Hampel filter,” SICE 2003 Annual Conference (IEEE Cat. No.03TH8734), 2003, pp. 2147-2151 Vol.2.

[13] R. K. Pearson, Y. Neuvo, J. Astola and M. Gabbouj, ”The class of generalized hampel filters,” 2015 23rd European Signal Processing Conference (EUSIPCO), 2015, pp. 2501-2505, doi: 10.1109/EUSIPCO.2015.7362835.