Power Optimization of Wifi Routers

Winfred Darko

Adviser: Kyle Jamieson

Abstract

In order to reduce the amount of energy wasted by wireless networks with idle access points, I implemented an approach to dynamically change the transmit power used by APs. This approach is an extension of the resource on demand (RoD) strategy SEAR, which is a set of approaches to reduce the amount of power used by APs in WLANs. However, the approach was designed to be applicable for access points that provide very little information about its current status, and the only real-time information it exploits is the number of users connected to an AP. In experiments, this implementation reduces the power used by APs that are not used for a sustained amount of time, but fails to provide reliable service, and so further work is required to understand how to ensure better service in future approaches.

1. Introduction

Throughout many campuses, corporate offices, libraries, and homes, one can easily find an array of wifi routers and access points when looking up or near a computer. While access to wifi in buildings was once seen as a luxury, the presence of wifi has proliferated in many locations. As wifi becomes more of an expectation than a luxury in many spaces, enterprise wireless local area networks, or WLANs, are using an increasing amount of devices to support this proliferation. Dell'Oro Group, a source for market information about the networking and telecommunications industries, estimates that by 2020, nearly 1 billion WLAN network devices will be installed for use [4].

In addition to routers and access points that provide internet service to homes and individual people, there are networks of enterprise WLANs on university campuses and corporate offices that contribute to this high number of network devices. In these spaces, large numbers of network devices such as access points and routers are placed close to each other, in order to provide quality service when supporting requests for high bandwidth and running delay-sensitive applications [7]. One such example is the enterprise WLAN installed within a four floor building of the Intel Corporation, in which 125 APs have been deployed about five meters from each other in a single floor, in a 80 meters x 38 meters area [7]. Another such example is the Dartmouth campus, consisting of 500 APs spread across 188 buildings in a 4 km x 5 km area [7]. When the APs are always at full power, this high density of APs means that the coverage of routers and access points can overlap each other.

While this high density of access points may be necessary to provide service for situations such as conferences or hackathons, where many wifi stations (STAs), or clients, are connecting to the network, most of the time these access points may be left unused. When multiple access points provide coverage to an area, but the clients connected to the WLAN need only one access point to maintain the needed bandwidth and quality of service, more power is consumed than necessary.

With multiple APs providing redundant coverage to a space where only one AP is needed, the additional APs continue to consume power without actually needing to be used. Such additional APs can be extremely common. For example, it was found that more than 70 percent of the APs in Intel's building were idle for more than 60 contiguous minutes [7]. On the Dartmouth campus, it was observed that more than 50 percent of the WLAN APs were idle for more than 60 minutes, with some of these APs idle for more than a full day [7]. Especially buildings that are generally left unused on certain times of the day, weekends, and holidays, a WLAN can be consuming enough energy to accommodate hundreds of clients when it actually supports just a handful of devices connected to the network.

With 100 APs consuming more than 70,000 MW of energy per year (8.76 MWh per year), and nearly 1 billion devices estimated to be installed by 2020, the amount of energy consumed by APs in WLANs can be significant [7, 4]. Therefore, it is important to reduce the amount of excess energy consumed by these APs.

The goal of this project is to design an approach that can decrease the amount of excess power used by wireless networks, particularly in rooms that are left unused for significant amounts of time. Because of the lack of knowledge of universally accessible router state information in real time, the algorithm is expected to run with as little information used in real time as possible.

Existing work towards this goal has already established a set of expectations for what a proper reduction of excess power should look like. Networks that reduce energy wastage while fulfilling these standards are called resource-on-demand (RoD). Furthermore, an RoD strategy, called SEAR, that implements these expectations, has been established. This strategy leaves space for input concerning changing transmit power dynamically, or as an AP runs, to allow for some APs to be powered off without affecting the quality of internet service.

While there are existing algorithms that control the transmit power to manage sets of routers, they required information that was not accessible in real time from the device we used. I have created a new algorithm which only uses the number of users connected to an access point once being run, and implemented this algorithm using the ESP8266 wifi development board. While tests have suggested that energy can be saved by controlling the transmit power of the routers, doing so would not abide by RoD standards. In following sections, this paper will

1. Define WLANs and their connection to transmit power
2. Summarize the implementation of SEAR, a set of resource-on-demand approaches to reduce excess power output of WLANs
3. Discuss existing approaches to dynamically change transmit power
4. Introduce an algorithm that may work within and without SEAR to change the transmit power of the router and save energy
5. Highlight details about the implementation of this algorithm on the ESP8266
6. Explain results of this implementation
7. Conclude with insights and suggest future work in this area

2. Problem Background and Related Work

2.1 High density WLANS

A wireless local area network (WLAN) is a network of devices that are connected to each other wirelessly, in a local area such as a room, building, or campus. The networks can be comprised of stations (STAs), otherwise known as clients, that can connect wirelessly to other devices, access points, that allow for such wireless devices to connect to a wired network, and switches and routers. Access points assist in connecting clients that are within a certain range of proximity to it, covering areas of a certain radius away from the AP. Here, high density WLANs are defined as WLANs with multiple access points that overlap in their area of coverage.

Something that affects the range of the area of coverage of an AP is the transmit power. When an access point sends data to another device, the transmit power determines the strength of the signal the router produces when it is transmitting. The transmit power is directly linked with the resulting signal strength, and therefore with the radius of coverage of an access point.

2.2 ROD Approaches to reduce power usage

Resource On-Demand (RoD) WLANs are centrally-managed WLANs that reduce the wastage of energy in high-density WLANs by "strategically powering on and off WLAN resources [7]." This means powering on and off APs, switches, and routers in the network such that the amount of power used scales with the amount of power needed to maintain coverage, and that redundant coverage is reduced without hurting the connection of clients to the network.

There are three requirements for a WLAN to be a RoD WLAN. First, that the WLAN ensures as much coverage as it would if all devices in the network were always on [7]. A RoD WLAN may not create coverage holes in which clients cannot connect to the network. Second, that the performance of the wireless network, as seen in packet loss or reduction in the sustainable transmission rate, is the same for RoD and non-RoD WLANs [7]. Lastly, that frequent client disconnects from the WLANs are avoided [7]. Powering on and off WLAN APs can force clients' connections to APs to change, which may negatively impact performance, so this should not occur frequently.

2.3 Existing SEAR approach

Using these requirements, an RoD strategy called Survey, Evaluate, Adapt, and Repeat (SEAR) was developed. It is comprised of four parts: Green clustering, user demand estimation, topology management, and user association management [7].

Green clustering is an algorithm that organizes sets of APs close to each other into subsets, or neighborhoods. The APs in any given neighborhood, or green cluster, are close enough so that it is possible for one AP to have the same area of coverage as all of the APs in its neighborhood.

Within each neighborhood, SEAR estimates the amount of user demand, in order to determine the amount of devices needed to provide sufficient coverage within the neighborhood, by using channel utilization [7].

At regular intervals, SEAR manages the topology of the WLAN by powering on or off clusters when the channel utilization value exceeds or drops below a pre-configured threshold, and configuring the channels of the APs within the cluster so that they don't overlap [7].

When powering on or off an AP, SEAR distributes clients and loads within the APs of a cluster. It manages user associations with clients by "reducing excessive roaming of users between APs," in which it proactively switches clients between APs to balance the load detected in a cluster so that there is better performance [7].

In managing the topology, the description of SEAR mentions that it is also possible to control the transmit power to decrease energy use [7]. However, an implementation of an algorithm for transmit power control for RoD WLANs is not discussed in the description.

3. Existing Approaches to Dynamic Power Change

TcP is a duo of algorithms that use the list of neighbors of an AP to calculate the ideal transmit power for that access point to “maximize the coverage and minimize its co-channel interference” [3]. Version 1, TcPv1, uses the received signal strength (RSSI) of a neighboring AP in addition to other variables to calculate the ideal transmit power [3]. Version 2, TcPv2, uses the radio frequency (RF) distance between it and a neighbor in order to calculate the ideal transmit power at a certain point [3]. It is unlikely that algorithms that depend on the details of their neighbors will consistently work for RoD clusters. For example, in the case where only one AP is turned on, there would be no list of active APs for both TcPv1 and TcPv2 to work with.

DTcP is an algorithm that does not rely on lists of other APs. Instead, it relies on real-time information about the state of the access points being changed, such as the channel occupancy, RSSI, and Modulation and Coding Scheme (MCS) index values [5]. While these variables are accessible for some networks, getting information such as the RSSI and channel occupancy is not available for some wifi devices, such as the ESP8266, the device used for experimentation. Therefore, an approach using even less information about the access point is needed to create a more general transmit control algorithm that can work for the ESP8266.

4. Limitations

In this paper, I propose an implementation for an algorithm that controls the transmit power of APs in RoD WLANs. In order for the algorithm to meet expectations, it must abide by all requirements of RoD WLANs while allowing for a decrease in the amount of power used by a green cluster. Additionally, while many APs are used in high density WLANs, there are also non-high density WLANs that may benefit from a new approach. Creating an approach that is also implementable for non-high density WLANs may allow this implementation to reduce power usage in networks, such as homes, that may not have clusters of routers.

Given the potential variety of access points by structure, permissions granted by the creators of the device, accessible features, and more, and given the variety of client devices that may connect to an AP, the information openly accessible to an implementation sphere consisting of APs, STAs, or both, is unknown. In order to help increase the applicability of the algorithm to various kinds of networks, the algorithm would ideally be as general as possible, so that as many devices as possible are able to use them.

In order to simulate such an environment, where a limited number of details about the accessible parameters of the access point are known, using an AP and/or STA combination with simply basic functionalities is the way in which I enforce the creation of an approach that is applicable to many WLAN networks.

For this reason, I used the ESP8266, a device that functions as an AP, with extremely limited information concerning the status of its use available. In fact, the only information available from the device itself about its use is the number of clients connected to it, and the transmit power level.

5. Approach

In an AP neighborhood that does not need extra broadband, energy would be saved by turning off all routers in that cluster except for one, and increasing the transmit power of that one router to increase its service coverage to the entire area typically covered by the neighborhood. If a given neighborhood is expected to cover a certain distance dist_coverage, the transmit power of a singular AP must be sufficient enough to provide service to devices that are of radius dist_coverage away, to allow for the remaining routers in that cluster to be shut off without creating holes in service. Having holes in the coverage area would be a detriment in the case where a stationary client needs to connect to an AP, but is out of range of the network.

Because only the number of STAs attempting to connect to the AP is accessible, information such as the RSSI, channel occupancy rate, and actual distance of client from access point is not able to be used in this approach.

Initially, an approach of simply increasing the transmit power as the number of STAs connected increased was considered. However, since a given network is comprised of neighborhoods in an RoD implementation, each neighborhood needs to cover a certain distance for the entire network to have coverage. If the minimum transmit power of an AP already allows for the neighborhood to cover this distance, and surrounding areas are already covered by surrounding neighborhoods, then increasing the transmit power above this minimum would be redundant.

To mitigate this inability to change the transmit power from the minimum without causing redundancy or holes in coverage, I took advantage of the information that there are periods in which APs are completely unused. When unused, the transmit power of the AP has no effect on any of the devices connected to it, as there are no devices connected to it. Outside of the scope of being able to connect to potential STAs in the coverage area, the transmit power can effectively be 0.

This knowledge is the inspiration for the approach used. While the transit power level must be constant when users are connected to it, the transmit power level can be 0 when users are not connected to it. In order for users to connect from the point that the power level is 0, the power level can spike to Tx_min after a certain amount of time wait_time for a certain amount of time checking_interval, and check whether users are attempting to connect. If no users are connected within that time interval, then the transmit power returns to 0. Otherwise, if a client is attempting to connect, then the transmit power remains at Tx_min until the power changes

6. Implementation

The algorithm created is listed here:

1. Determine Tx_min from dist_coverage, so that the access point's transmit power is sufficient enough to provide quality coverage to the area of the green cluster
2. Determine wait_time
3. Set transmit power of AP to Tx_min
4. After wait_time has passed, check if STAs are connected or connecting to the AP for a time period of checking_interval.
5. If the AP has 0 clients, set transmit power to 0.
6. If the AP has at least 1 client, set transmit power to Tx_min.
7. Return to 3.

This algorithm was implemented using the ESP8266 NodeMCU LUA CP2102 ESP-12E Internet WiFi Development Board, using the Arduino IDE. This device was used as an AP, while phone devices were used as clients connecting to the STA. The code used for implementation is inspired by code about the ESP8266 Arduino IDE library, articles of the devices' wifi modes, and an article about visualizing data in the Arduino IDE [1, 6, 8, 9].

In order to measure the received signal strength, or RSSI, of the devices from the ESP8266, we used a phone app called WiFi Visualizer.[1] Doing so required a connection to the AP, which would not occur when the transmit power would be set to 0. In an attempt to keep the phone measuring signal strength connected to the router, the minimum transmit power of the router was set to 5 dBm, instead of 0 dBm. In order to simulate the use of a variety devices in a WLAN, the devices connected to the AP were an Apple iPhone, a Sony Xperia X, and an Asus X00RD.

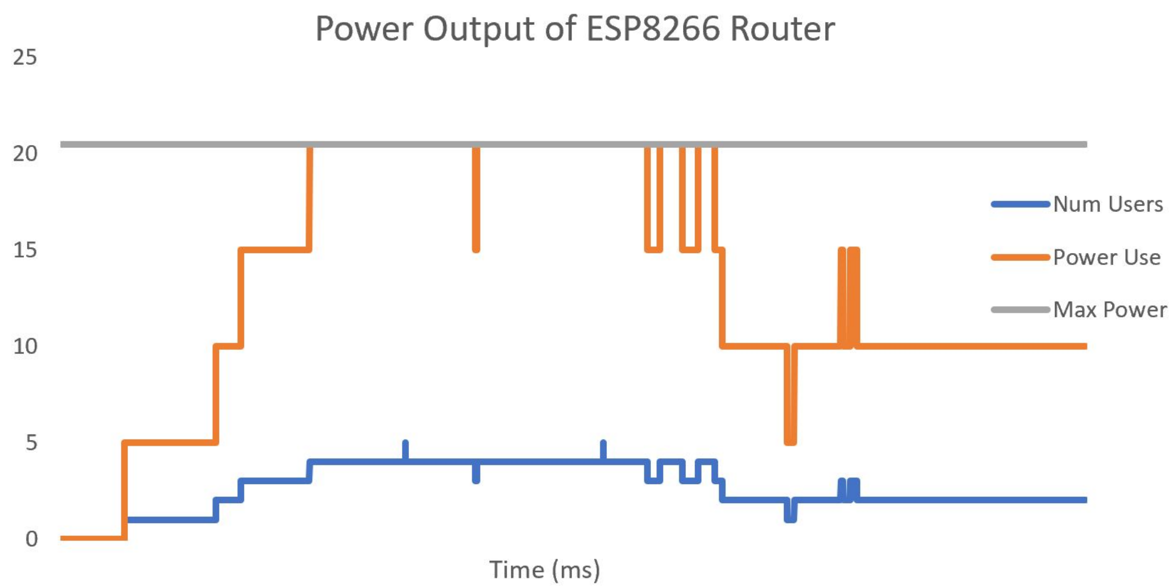
7. Evaluation

For the initial approach of increasing the transmit power based on the particular number of connected clients, as opposed to the existence of connected clients, a run measuring the transmit power in this process occurred before the approach was changed. Results of this initial

---

[1] ITO Akihiro, "Wifi Visualizer App," found in *Google Play Store*, https://play.google.com/store/apps/details?id=net.i.akihiro.simplewifianalyzer&hl=en_US

approach can be seen here:



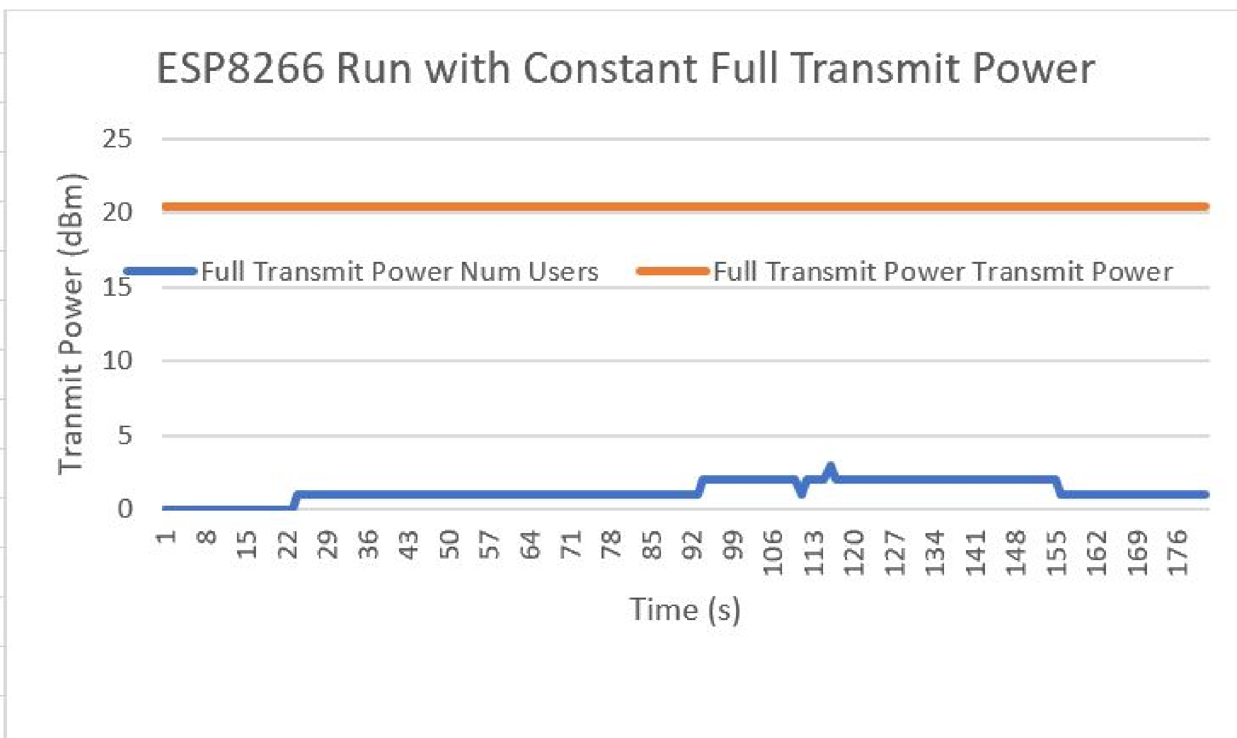
As shown in the graph, the transmit power output correlated directly with the amount of users. While the concept of an immediate, direct correlation between the amount of power used by an access point without any need for overhead is ideal, this cannot be achieved by changing the transmit power, as coverage of the entire neighborhood area is necessary.

The final experiment is a run-through of the coding implementation of the algorithm created. To configure Tx_min, I remotely measured the RSSI from the router upon connecting at different distances from it. This way, the transmit power required to provide service of sufficient quality to the neighborhood's area of coverage would be determined prior to running the algorithm. As a result, the desired quality of service is left to the implementer of the algorithm. This responsibility allows for the transmit power implemented to be tailored to the particular network in place.

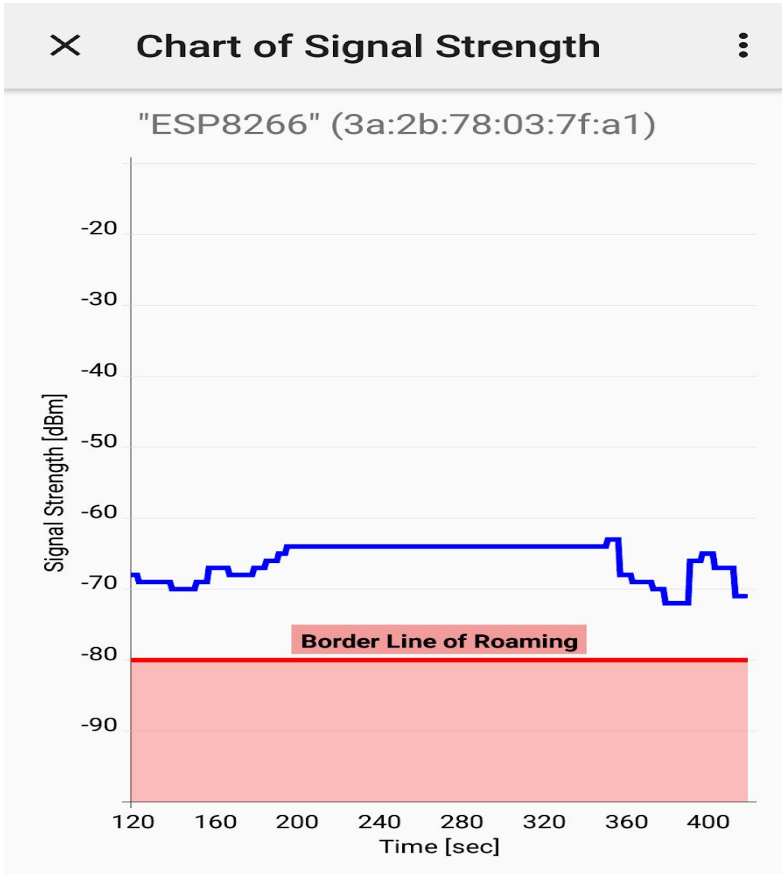
As Tx_min depends on the quality expectations of the implementer, and the quality expectations of this experiment is that the quality remains at par with the implementation of the experiment at full power, instead of a specific signal strength, the dist_coverage is arbitrarily chosen to be 10 feet. Therefore, the received signal strength 10 feet away away from the access point will be recorded as the device runs. This access point will be run with the transmit power constantly at full power, and with the transmit power using the created algorithm where transmit power is variable. In this particular experiment, wait_time is set to 5 seconds, and checking_interval is set to 1 second. In future tests, these two variables may be changed to fit the user's needs.

For each run, the number of clients, real-time transit power, and received signal strength were recorded. While the number of clients and the real time transmit power were recorded from the ESP8266 itself, the real time RSSI was recorded remotely, from a phone using the the Wifi Visualizer mobile application. Results are shown in graphs 1 through 4.

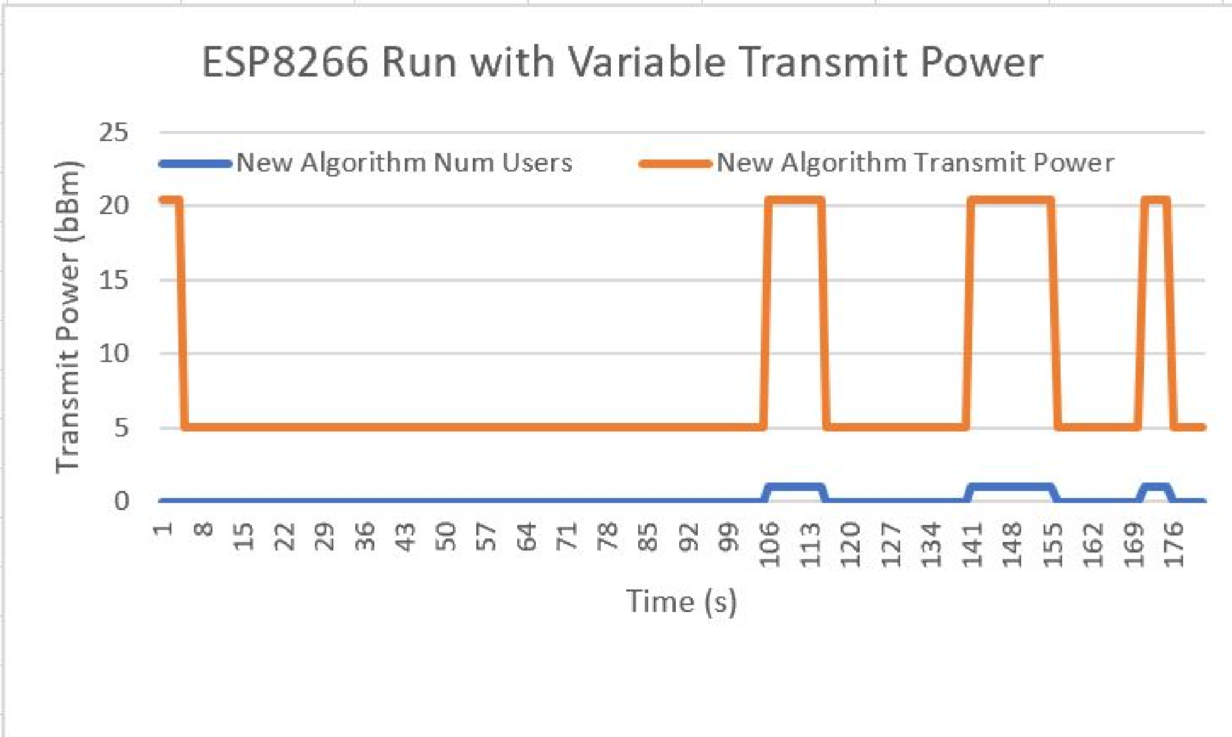
Graph 1: Transmit Power of ESP8266 AP Run with Full Transmit Power



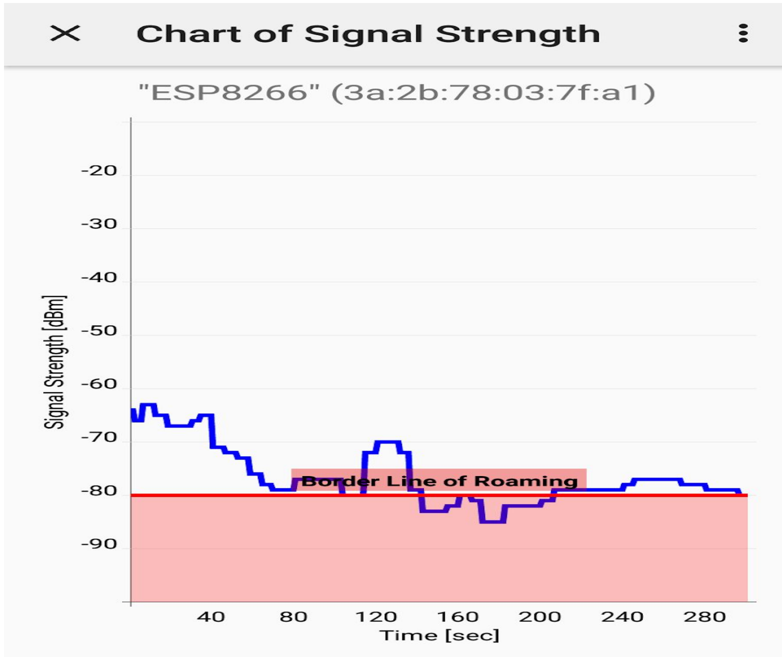
Graph 2: Received Signal Strength of Full Transmit Power Run from 10 feet Away



Graph 3: Transmit Power of ESP8266 AP Run with Transmit Power Set By New Algorithm



Graph 4: Received Signal Strength of Variable Transmit Power Run from 10 feet



While initially, the transmit power output of both runs were similar, the power output of the variable transmit power run quickly changed, as expected. Upon noticing the RSSIs of the runs, however, we can observe that the signal strength varied greatly. Initially, the signal strength received by the variable transmit power run was even higher than the received signal strength of the constant transmit power run, but the received signal strength of the variable transmit power run soon diminished while the signal strength of the constant transmit power run remained more consistent. Something to note is that even at times when the transmit power is the same for both runs, the RSSIs of the first and second approaches at those times were still different.

During experimentation, the device recording the received signal strength continuously lost connection to the AP when running under the created algorithm, so the received signal strength of the new algorithm could not even be measured for as long as the set of the new transmit power. Something unexpected about this is that at the same time that this phone (the Sony Xperia X) was disconnected, another phone (the Asus X00RD) was able to connect to it. A potential explanation for this is detailed in the summary.

8. Summary

8.1 Conclusions

Success for the algorithm would mean that the requirements of RoD would not be hindered by a new implementation. In this experiment, this experiment did not achieve this goal. A successful algorithm, that uses the RoD requirements, would mean that

1. The algorithm ensures the same amount of coverage as the coverage of the cluster of devices always on
2. The algorithm allows for the performance rate of a fully-on cluster to be the same as the performance rate of the cluster using the algorithm
3. Frequent client connections/disconnections are avoided.

Within the RoD framework, increasing the transmit power of one access point in a cluster would increase the area of coverage enough so that remaining access points in that cluster can be turned off when there is little demand for service. In this experiment, these RoD standards were not achieved.

With a singular access point being tested, while the transmit power output was different between the two runs, the received signal strength of the run with the new algorithm was, at most times, significantly lower than that of the used algorithm. Since a singular access point failed to provide the same signal strength in the new algorithm's run, if other APs surrounding it were turned off, there would be holes of coverage. In this experiment, a hole of coverage was observed when the phone measuring the RSSI data disconnected due to lack of service.

With a singular router on, with a lower performance rate, the performance rate of the entire cluster would be lower than that of a fully-on cluster.

As seen by the disconnection of the phone to the AP, frequent disconnections could not be prevented.

8.2 Limitations and Future Work

The reasoning for the failure of this algorithm is not fully realized. An assumption of the algorithm is that since the minimum transmit power of the device necessary to cover a certain distance is determined by the user, the presence of service in an area is dependent only on the Tx_min chosen by the implementer of the algorithm. However, even when the transmit power of the access point in both runs were the same, the resulting received signal strengths were different between the two runs.

One potential explanation is that the ability of certain clients to connect to access points depends largely on the consistency at which the signal persists. From the same distance away, the Asus X00RD was able to connect to the access point in both runs, while the Sony Xperia was only able to remain connected for the run with consistently full transmit power. Future work may look into the process by which clients determine access points to connect to.

At this time, it is still uncertain whether it is possible to create an RoD approach using only the transmit power and the number of users connected to a particular AP to reduce the power usage of clusters of neighborhoods, without creating holes in coverage. For this particular approach, future work may look into changing the values of the parameters used, such as the wait_time and checking_interval, as the length at which the transmit power is at needed capacity may also impact the connection of devices to access points.

9. Honor Code

I pledge my honor that I have not violated the Honor Code in this research

- Winfred Darko

References

1. Sarah Ali, “How to use ESP8266 Wifi Modes: Station and Access Point”, https://onlineshouter.com/use-esp8266-wifi-modes-station-access-point/
2. ITO Akihiro, “Wifi Visualizer App,” found in Google Play Store, https://play.google.com/store/apps/details?id=net.i.akihiro.simplewifianalyzer&hl=en_US
3. Cisco, “Transmit Power Control (TPC) Algorithm”, found in Radio Resource Management White Paper, https://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-1/mobility_express/b_RRM_White_Paper/b_RRM_White_Paper_chapter_0101.html
4. Chris DePuy, “Internet of Things (IoT) and Wireless LAN (WLAN) Go Hand-in-Hand”, Dell’Oro Group, http://www.delloro.com/other/internet-of-things-iot-and-wireless-lan-wlan-go-hand-in-hand
5. C. Gandarillas, C. Martín-Engeños, H. L. Pombo and A. G. Marques, "Dynamic transmit-power control for WiFi access points based on wireless link occupancy," found in 2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, 2014, pp. 1093-1098.
6. Jonathan Pereira, “Arduino Serial Plotter”, https://www.instructables.com/id/Ultimate-Guide-to-Adruino-Serial-Plotter/
7. Jardosh, Pooja & Papagiannaki, Konstantina & Belding, Elizabeth & Almeroth, Kevin & Iannaccone, Gianluca & Vinnakota, Bapi, “Green WLANs: on-demand WLAN infrastructures,” found in MONET
8. “TX Power Setting”, found in ESP8266 Community Forum, https://www.esp8266.com/viewtopic.php?f=32&t=13496
9. ESP8266WiFiGenericClass Class Reference, https://links2004.github.io/Arduino/d0/d52/class_e_s_p8266_wi_fi_generic_class.html#ae94015cbccaa5e13fe2d2ee3756e9047