WI-FI SENSING ALGORITHMS UTILIZING ZIGBEE RF RECIEVER FOR USE IN EMERGENCY COMMUNICATIONS MESH
WI-FI SENSING ALGORITHMS UTILIZING ZIGBEE RF RECEIVER FOR USE IN EMERGENCY COMMUNICATIONS MESH

An Undergraduate Honors College Thesis

in the

Department of Computer Science and Computer Engineering
College of Engineering
University of Arkansas
Fayetteville, AR

By

Alexander H Nelson

May 2012
University of Arkansas
ABSTRACT

This thesis introduces the idea of a low-power Wi-Fi sensing wake-up controller for an emergency communications mesh network, progressively developing a prototype system which could be used in a live environment. Wireless network protocols are reviewed, as well as a limited view of cluster analysis, in order to introduce relevant concepts crucial to understanding this thesis. Algorithms for system implementation are developed, and pseudocode, designed to be configurable and platform independent, is given for each. Design goals for the system are identified with potential approaches are defined in order to optimize for each. An example hardware configuration is given, in conjunction with analysis of benefits and drawbacks of several design options. Finally, the prototype is tested according to design goals in order to prove its feasibility. The results demonstrate that the prototype meets the proposed design goals.
This thesis is approved for recommendation to the Graduate Council.

Thesis Director:

[Signature]
Dr. Pat Parkerson

Thesis Committee:

[Signature]
Dr. Nilanjan Banerjee

[Signature]
Dr. David Andrews
THESIS DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this thesis when needed for research and/or scholarship.

Agreed ____________________________

Alexander H Nelson

Refused ____________________________
ACKNOWLEDGEMENTS

I would like to thank Dr. Pat Parkerson and Dr. Nilanjan Banerjee for their continued support throughout this thesis. They provided me the opportunity, the outlet, and the resources to produce this system. Their encouragement when I was able to achieve goals quickly, and patience when it seemed nothing was going right, helped me to be able to create a project that I am proud of and that has the potential to be implemented in a live-fire system. I would like to thank Dr. Parkerson and Dr. Banerjee, as well as Dr. David Andrews for agreeing to critique and polish this thesis. I would like to thank the Honors College for the financial support I received to work on this thesis.

I would like to thank James Morris for his help getting me integrated into the lab environment. That first step was a big one, and without James it would have been a much rougher experience.

I would also like to thank my family, who have supported me throughout my education, and who are constantly proud of my accomplishments. Without my brother setting a bar for me to reach for, and my parents pushing me to that level, I would not be the student I have become. To my friends, I thank you all for your support of varying kinds during this process. Some of you gave emotional support, some gave intellectual support, and still others gave me something better to do when I could not conceive of working another minute. I also would like to thank Randi Collins for being patient during my long hours at the lab hoping to finally see positive results, and for being there for me whatever the outcome.

To all who have given me support during this venture, I sincerely thank you. It has been an honor to live up to the support given.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................................................ iv

1. Introduction .................................................................................................................................. 1
   1.1 Problem .................................................................................................................................... 1
   1.2 Thesis Statement ..................................................................................................................... 3
   1.3 Approach ............................................................................................................................... 3
   1.4 Organization of this Thesis ..................................................................................................... 4

2. Background ................................................................................................................................... 6
   2.1 Key Concepts .......................................................................................................................... 6
      2.1.1 802.11 (Wi-Fi) and 802.15.4 (ZigBee) Communications Protocols ....................... 6
      2.1.2 A priori Clustering Algorithms ...................................................................................... 8
   2.2 Wi-Fi interprotocol Cooperation and ZiFi ......................................................................... 9
      2.2.1 Esense ............................................................................................................................ 10
      2.2.2 ZiFi .................................................................................................................................. 10

3. Architecture .................................................................................................................................. 13
   3.1 High Level Design - Hardware ............................................................................................ 13
   3.2 High Level Design – Software .............................................................................................. 14
   3.3 Design .................................................................................................................................... 15
      3.3.1 Initial Experimentation ................................................................................................... 16
      3.3.2 Packet Identification Algorithm .................................................................................... 19
      3.3.3 Packet Classification Algorithm .................................................................................... 20
   3.4 Implementation ....................................................................................................................... 22
4. Methodology, Results and Analysis ................................................................. 26

4.1 Methodology ................................................................................................. 26

4.2 Results ........................................................................................................... 27

4.3 Analysis ......................................................................................................... 28

5. Conclusions ..................................................................................................... 30

5.1 Summary ....................................................................................................... 30

5.2 Contributions and Potential Impact ......................................................... 30

5.3 Future Work .................................................................................................. 31

6. References ...................................................................................................... 33
LIST OF FIGURES

Figure 1: Comparison of Wi-Fi and ZigBee channel layout .......................................................... 7
Figure 2: Example J48 tree demonstrating likelihood of actions based on training data ............ 9
Figure 3: Hardware Specification for Wake-Up controller ............................................................. 13
Figure 4: High Level Algorithm Specification ............................................................................. 14
Figure 5: Faraday cage used in experimentation .......................................................................... 17
Figure 6: Comparison of Wi-Fi and Bluetooth RSSI energy signatures ................................. 18
Figure 7: Wi-Fi and Bluetooth waveforms visualized by linear difference and wave size .... 21
Figure 8: J48 tree generated by Weka Classifier .......................................................................... 22
Figure 9: Histogram displaying the frequency of the response time of the algorithm .......... 28
1. INTRODUCTION

1.1 Problem

A stable form of communication has moved from the realm of luxury to being expected, even necessary for modern living. This has become exaggerated in recent years as most Americans have access to an internet enabled mobile phone within a few feet of their person at all times. Physical road maps have even become superfluous as many phones have direct access to GPS navigation. Mobile phones can often communicate regardless of cellular connectivity, as many of these devices can connect to any Wi-Fi hotspot and use a VOIP application to speak with any other internet connected user. The robustness of the mobile phone as a communication device is indisputable, and their presence bordering on ubiquity lends support to smartphones being a viable candidate for any communication in any scenario. However, in the event where a region is left without power for an extended period of time, mobile phones, as well as any other traditional forms of communication, are left crippled, if not useless. This scenario, however, is a common occurrence during a natural disaster.

Natural disasters, such as floods, snowstorms, tornados, and hurricanes cause thousands of deaths around the globe each year. While many of these deaths occur as a direct result of the event itself, a much larger portion occurs after the event, an anomaly which could be mitigated by a proper post-emergency communication infrastructure [1]. For example, hurricane Katrina made landfall in the fall of 2005, and in a matter of hours had left a majority of the gulf coast without power or communication, a situation which would not be remedied for several months. More recently, the Tohuku earthquake and resulting tsunami in 2011 left almost 4.4 million households without power for several days [2]. State of the art emergency communication generally occurs over satellite phone which provides constant access to a reliable system but
suffers from the fact that the systems are costly, and only a small minority of people has access. Most emergency scenarios require the distribution of information such as alerts, warnings, and the location of shelters or sources of clean water; a task which is monumental for just the users of satellite phones.

Therefore, the need is readily apparent for the development of a reliable alternative emergency communications infrastructure capable of handling such requests to a large group of clients. The production of any end-user system for distribution is infeasible, as the cost for such a large deployment would be insurmountable, and the distribution and training for such a system would create overhead that would render the system useless for the cause identified. Therefore, the system should either be a multiuser system at centralized locations, or immediately integrate with current end-user systems with little or no hassle on the users’ part. As stated above, the ubiquity of communications by mobile phones as well as the predefined communication protocols, combined with the fact that users’ require no additional training, make mobile phones a prime target for the second option. An infrastructure which takes advantage of these preexisting conditions would maximize the resources at hand while adding no additional cost. These factors led to the decision to utilize a distributed mesh network which interfaces directly with end-user mobile phones. As the communication protocol to these devices would most certainly be Wi-Fi, this infrastructure should also be able to interface with off-the-shelf laptops.

Unfortunately, the power consumption of a standard Wi-Fi radio (“1.65W, 1.4W, 1.15W, and .045W in transmit, receive, idle and sleep states respectively”) is more than could be hoped to be scavenged by current solar panels (~10mW/cm² outdoors at noon) unless the transceiver maintains a sleep state for long periods of time [3][4]. However, network discoverability is impossible while the wireless access point is in sleep mode, which would render the system
useless. An optimal solution then would be to keep the Wi-Fi radio in sleep mode until a user wishes to connect to the network, at which point the Wi-Fi radio is brought out of its sleep mode to establish a connection with the user.

1.2 Thesis Statement

In conjunction with PIs Banerjee and Parkerson to develop a robust, solar powered emergency communications mesh network, this system has been developed in order to provide access to a standard Wi-Fi radio without exhausting the limited power available from the provided solar cells. To achieve this, a lower power radio called ZigBee which communicates on the same frequency band (albeit a different interface) will be utilized to determine if a client is seeking Wi-Fi presence in a given area.

1.3 Approach

The proposed system seeks to solve the issue of power management for the emergency mesh network. The major drain it seeks to replace is the power consumption of the Wi-Fi radio during periods of inactivity. By placing the Wi-Fi radio into a sleep mode, that drain is reduced by a factor of 30. However, in this sleep mode, the access point cannot allow users to connect, nor sense if a user is attempting to connect. The system solves this divide by using a wake-up controller composed of a low-power PIC microcontroller and a ZigBee RF transceiver to determine if a user is seeking wireless access. The ZigBee radio consumes considerably less power than the Wi-Fi radio; maxing out at 76mW during a transmission [5]. ZigBee and Wi-Fi, however, maintain entirely different physical layers despite transmitting data in the same frequency. Therefore, analysis must be performed on the received transmissions in order to determine if its source is a user seeking Wi-Fi access.
The signal analysis will be performed by utilizing a J48 tree; an implementation of a priori cluster analysis which classifies data points based on statistical properties. The signal must be classified and its origin verified so that the system does not unnecessarily wake the Wi-Fi radio and waste scavenged energy. During active Wi-Fi communication, the ZigBee radio can be put to sleep, during which it consumes only 6.6µW [5].

1.4 Organization of this Thesis

The rest of the thesis is laid out as follows:

Chapter 2 covers background subjects that pertain to this thesis so that a general understanding of the methods utilized can be obtained. A literature review is included for interprotocol cooperation in order to understand the current progress of similar sensing projects, and how this thesis fits into the current progression.

Chapter 3 describes the architecture of the system in terms of both the hardware and software. Example hardware is described from a high-level view, describing the benefits and drawbacks of several design options, allowing for a completely user defined hardware interface as the sensing algorithm is designed to be platform independent. The architecture of the software is discussed in high-level, including a diagram of the final system, and in low level, including discussion of techniques used to develop the algorithms and pseudocode for prototype implementations of each. Chapter 3 also describes the initial experimentation used in determining the makeup of the prototype which is developed throughout the chapter.

Chapter 4 covers the testing procedures used in verifying the correct functionality of the developed prototype. Design criteria discussed in Chapter 3 are reviewed, and analysis is provided describing how the system met each with suggestions for possible future
improvements. Numerical analysis is provided describing the speed and accuracy of the system, detailing its feasibility as a functional wake-up unit.

Chapter 5 provides conclusions derived from the body of work presented, detailing its potential contributions and impact for both its intended purpose other possible realms. Improvements and additions to the current system are discussed in addition to a glimpse of possible future projects describing several possible uses for a similar system tweaked for specific design goals.
2. BACKGROUND

2.1 Key Concepts

The reader must understand wireless communications protocols 802.11 and 802.15.4 and limited aspects of a priori clustering algorithms in order to foster a thorough understand this thesis. A short review of each is given below:

2.1.1 802.11 (Wi-Fi) and 802.15.4 (ZigBee) Communications Protocols

802.11 wireless communication protocol, more commonly referred to as Wi-Fi, is the standards which set forth the method in which data is transmitted between wireless radio transceivers. Wi-Fi communication exists in the 2.4 GHz frequency band utilizing special blends of frequency modulation called phase-shift keying in order to encode the binary representations of the data being sent. Wi-Fi “channels” are 22 MHz frequency bands in the spectrum encapsulating a single stream of communication. There are 14 channels in the 2.4 GHz range, the peaks of which are situated every 5 MHz [6]. Most wireless access points (WAPs) communicate on one of four channels 1, 6, 11, and 14 which represent disparate non-overlapping channels, thus reducing noise between competing channels. Channel 14 is in licensed airspace in most countries, and therefore reserved for restricted applications.

Data is transmitted in “frames” of different types, each containing a subset of properties agreed upon as the current standard. Frames may be encrypted if security protocols such as WEP or WPA are enabled, which obfuscates the frame for all but the intended devices so that frames may not be read properly by any other listening client.
802.15.4 wireless communication protocol is the unlicensed airspace which allows the user to define the upper layers of the protocol standards. ZigBee is one of several specific standards created within this protocol, and operates within several frequencies, primarily the 2.4 GHz range. ZigBee radio devices have the advantage of shifting from sleep mode to active mode in a short period of time (~15ms), allowing the transceiver to be both responsive and power efficient [7]. ZigBee, when used as a means of communication will attempt to transmit data on channels that correspond to the least amount of Wi-Fi interference by utilizing ZigBee channels 15, 16, 21, 22, and 27 (if not in licensed airspace) which fall between Wi-Fi channels 1, 6, 11, and 14 respectively as seen in Figure 1 [8]. These channels provide optimum signal strength and noise reduction for unlicensed airspace in the 2.4 GHz range. ZigBee channels occur every 5 MHz and each channel covers approximately 2 MHz.

The physical layers of the two protocols are distinct, and consequently one cannot read the contents of the other. However, the basis of this thesis hinges on the fact that these two communication protocols overlap in frequency, and therefore have the ability to sample the same air space.

Figure 1: Comparison of Wi-Fi and ZigBee channel layout
2.1.2 A priori Clustering Algorithms

Cluster analysis is any statistical model which groups statistically alike data points into categories [9]. These categories can then be modeled in order to provide analysis and assumptions about future data points. A priori clustering is the set of clustering algorithms which uses predetermined nominal categories and training data in order to improve the statistical clusters based on the statistical features gathered about the objects. The purpose of such algorithms is to be able to provide adaptive computer systems which can make decisions based upon a high statistical likelihood without any further input from a user. The statistical data to be modeled against can be user defined, or a large amount of statistical data can be sifted through to provide maximum accuracy.

For instance, a clustering algorithm could be used to group similar people into categories so that, given a few data points concerning an individual, proper advertisements may be delivered to that individual, constituting an adaptive version of market research.

This algorithm used several instances of a priori clustering in order to determine a model of high statistical significance which would provide a responsive yet highly accurate formula for predicting the existence of a user seeking Wi-Fi access. The specific algorithm chosen for this application is the J48 decision tree implemented in Weka, a data mining tool. Decision trees are described by Bresfelean as the following:

“Decision trees models are commonly used in data mining to examine the data and induce the tree and its rules that will be used to make predictions. The true purpose of the decision trees is to classify the data into distinct groups or branches that generate the strongest separation in the values of the dependent variable” [10]
A J48 decision tree is implemented using a set of training data points, and discriminating between them based on their given attributes. The algorithm uses the following structure in order to create the most distinctive tree. First, a decision is made on which attribute creates the most distinct split between the data. Then, for each created node, if all data points are not within the same category, a subtree is created which best divides the remaining data. Otherwise, the branch is terminated and the node is assigned to the category describing the data. This algorithm continues until there is no more ambiguity or no more attributes can be used in order to further separate the data. The latter case then assigns all leftover nodes to the category which best describes the data. Figure 2 gives an example of how a J48 tree can be used to determine future actions based on a known training set using data taken from the 2009 NFL season [11].

![Figure 2: Example J48 tree demonstrating likelihood of actions based on training data](image)

2.2 Wi-Fi interprotocol Cooperation and ZiFi

Several projects are currently seeking to increase the cooperation between distinct communication protocols in order to increase productivity in some manner. Projects like Esense,
S-WOW and Wake on WLAN promote cooperation by direct communication between protocols, while projects like ZiFi, Wake-On-Wireless, and Turducken promote a hands-off listening approach to provide information about networks to extend the lifetime of battery-powered clients [12][13][14][15][16][17].

2.2.1 Esense

Esense proposes a mode of communication between separate physical layers of communication protocols by creating a limited alphabet from the energy profile determined from the RSSI (received signal strength indicator) values scanned during a given frame. Using this method, the writers prove that a reliable paradigm of communication can be created. “Our results show that we could potentially create an alphabet size as high as 100; such a large alphabet size promises efficient Esense communication” [12]. The drawbacks of such a project is that significant changes have to be created to end-devices using each protocol in order to accept such a language, and is therefore currently feasible only in specialized applications.

2.2.2 ZiFi

ZiFi is an NSF funded project which aims to utilize a ZigBee wake-up controller for mobile-phone or laptop users in order to save battery life while constantly scanning for Wi-Fi access points. In essence, ZiFi represents the exact opposite goal from what is proposed in this paper. The driving force behind ZiFi is the constant drain that a Wi-Fi radio enacts on a mobile device during its active scanning period. In order to reduce this cost, a secondary low-power ZigBee radio is used to scan the frequency in order to determine the presence of Wi-Fi WAPs in the vicinity, and to wake up the Wi-Fi radio on the client device (typically a mobile phone or laptop). The process by which ZiFi achieves its goal is detailed here:
“To capture Wi-Fi interference signatures, ZiFi utilizes the received signal strength (RSS) indicator available on ZigBee-compliant radios. However, we observed that the statistics of Wi-Fi RSS samples, such as power magnitude, time duration, and inter-arrival gap, exhibit surprising resemblance with those of other RF sources, and hence provide little hint about the existence of Wi-Fi. Motivated by this observation, ZiFi is designed to search for 802.11 beacon frames in RSS samples. Periodic beacon broadcasting is mandatory in Wi-Fi infrastructure networks and hence provides a reliable means to indicate Wi-Fi coverage.” [15]

Beacon broadcasting is the method which promotes Wi-Fi WAP discovery, wherein WAPs will release a periodic frame of a short duration containing a succinct homogeneous set of data. The energy signature created by beacon frames is easily spotted given a controlled environment. However, ZiFi operates in a lively RF environment, and as a result the algorithm used must be more robust to noise. ZiFi has the advantage that it may take longer for calculations as users are generally in a WAP’s range for an extended period of time, allowing the algorithm to use more data and complex arithmetic operations. The false negative rate can also be higher than for this application, as a false negative in ZiFi represents a user missing the opportunity to utilize a Wi-Fi network where a false negative in the mesh network represents a potential survivor missing access to critical emergency information.

The algorithm developed utilizes a special variation of folding to determine Wi-Fi activity with an error rate of below 4.8%. The prototypes developed require additional hardware to be attached to the mobile phones or laptops, which increases power consumption and adds
bulky external equipment. However, the goal is to extend this idea to a dedicated piece of hardware inside machines as a functional low-power wake up controller provided the transceiver can be integrated with current technologies.
3. ARCHITECTURE

3.1 High Level Design - Hardware

The design of the hardware is dependent upon two factors: whether or not the ZigBee module will also be used for communication, and whether or not the Wi-Fi sensing module will be implemented as a canned-up piece of hardware acting specifically as a wake-up controller, or as software on the client device.

![Diagram of Hardware Specification for Wake-Up controller]

Figure 3: Hardware Specification for Wake-Up controller

Figure 3 shows an example of how the hardware could be configured. This setup assumes that the microcontroller used for sensing Wi-Fi activity is separate from the application, and allows for interrupt behavior to be modeled using the Interrupt/Ack ports, or could simply act as a constantly updating Wake value by ignoring the Ack port. The enable bit allows for the entire wake-up controller to be asleep while the main system is in use, while the wake-up function would allow the opposite behavior, allowing the controller to wake up the main system upon Wi-Fi discovery. This hardware specification also keeps the ZigBee Transceiver separate so that the main system may also communicate with it using the SPI interface, or let it sleep when unused as the wake-up time for the ZigBee module is approximately 15ms.
3.2 High Level Design – Software

The software, then, is dependent upon how the hardware is specified, and whether or not the controller should act in an interrupt manner, or as a constantly updating value. The basic discovery algorithm is platform independent, and requires only the RSSI stream sent from the ZigBee transceiver. We will assume for this instance that the sensing module is a separate entity, and that it has full control of the RF Transceiver. The behavior of the output bits is dependent upon the platform, therefore, for this instance, we will assume a one bit output that is active high once Wi-Fi activity is determined.

![High Level Algorithm Specification](image)

**Figure 4: High Level Algorithm Specification**

The two main functions within this algorithm are the packet identifier method and the packet classifier method. These two methods take as input the RSSI data stream from the ZigBee
module, transforming it into a recognizable data structure which is then classified according to
the a priori clustering algorithm defined by the user, and finally a determination is made on
whether the packet is a Wi-Fi packet, a Bluetooth packet, a ZigBee packet, or noise. If the packet
is determined to be a Wi-Fi packet, an optional redundancy check can be used to increase
accuracy, but at the cost of speed and possibly causing a false negative result. This optional
redundancy check can be defined by the user based on the parameters that are trying to be
optimized.

3.3 Design

As stated above, ZigBee communication generally occurs on a specific subset of channels
in order to avoid the peaks created by Wi-Fi interference. However, if one were to attempt to
sense active Wi-Fi access points or users searching for Wi-Fi access, one would instead listen on
ZigBee channels correlated most closely with the peaks of Wi-Fi channels 1, 6, and 11. ZigBee
channels 12, 18, and 25 therefore would need to be scanned periodically to check for wireless
activity. A lack of activity on these channels indicates that there is likely no Wi-Fi presence in
the area. However, activity on these channels does not necessitate that there is an active WAP or
user searching for Wi-Fi access. Other sources of noise (including microwave radiation,
Bluetooth, and other ZigBee components) can emit waves in this frequency causing a false
positive during which it would be a waste of scavenged energy to turn on the Wi-Fi antenna.
Therefore, cluster analysis was chosen to be leveraged to determine within a relative accuracy
whether a given signal belongs to Wi-Fi or some other signal or noise in the same frequency
band.
3.3.1 Initial Experimentation

In order to process signal at a constant rate, a method native to the ZigBee module was utilized to calculate the signal strength throughout reception of a packet, and to store every signal sequentially on some external media so that a contiguous waveform could be generated. The ZigBee module must be tweaked to accept all packets, regardless of its ability to decode them, so that it could calculate signal strength of packets external to the ZigBee standard. In order to determine what waveforms corresponded to different sources of noise, controlled experiments were devised to block for all signals except the current source being analyzed. An unfortunate side-effect of the ubiquity of Wi-Fi Hotspots was the absolute inability to block for noise in an inhabited setting.

To meet this constraint, a faraday cage was created that could house the experiment in a signal free environment. The cage is a 5’ x 5’ X 7’ structure that is wrapped entirely in brass wire mesh in a continuous conductive sheet so that no signal in the spectrum could penetrate its surface. In order to increase its portability, each face is disconnected from the whole, but held together by hinge clamps. The electrical continuity is preserved by wrapping the brass mesh to the inside of each contact point, and filling the void with a copper braid which expands to fill the width. In trial, with two layers of brash mesh, the faraday cage was found to completely attenuate all but the strongest signals in the 2.4 GHz range, and attenuate even a WAP in direct line of sight to almost unrecognizable noise. Removing this WAP allowed for complete radio silence in the testing environment.
While it is industry standard that WAPs will emit a beacon frame periodically in order to signal its availability, and that these frames are mostly identical in their energy signature, the same fact was not initially known for clients searching for wireless access. It seemed logical that a client could simply passively listen for beacon frames which correspond to available access points and then offer the user the ability to connect if desired. This scenario would have rendered this algorithm unfeasible because it is unlikely that the average end-user would be able to deploy a WAP in an emergency scenario where such a system is needed for communication.

Therefore, experiments were then developed to send a constant Wi-Fi scanning signal in order to emulate a user searching for an active WAP. Several preexisting Android Apps were utilized during this process, including WiFiScanner by PinApps and Wi-Fi Analyzer by Kevin Yuan [18]. Originally, Wireshark was utilized to capture all packets to determine if there was any client structure similar to the beacon frame. Initial scans did not pick up any activity, but
allowing Wireshark to collect packets in promiscuous mode showed a repetitive packet of short
duration was emitted by each of the client devices used in testing.

In order to visualize the energy signature of these discovered waveforms, each packet
was collected by the ZigBee module and sent by a serial RS232 interface to a computer which
cataloged the data. Each trial included 4,000,000 data points being collected in sequence, which
included the time and the signal strength of each scan. This experiment was repeated for
Bluetooth to gain access to a similar result set and the signals were compared. These experiments
were reproduced using several end-devices, including mobile-phones from several different
manufacturers, and recent market laptops with Bluetooth and WiFi connectivity.

Figure 6: Comparison of Wi-Fi and Bluetooth RSSI energy signatures
Wi-Fi signal in scanning mode is a regular periodic waveform with a slim profile where Bluetooth is a much longer, blockier waveform as shown in Figure 6. At this point, it became apparent that the statistical properties of individual waveforms of different protocols were likely to provide an easy schism which could be clustered against in order to provide an interface for simple protocol determination of future packets.

3.3.2 Packet Identification Algorithm

To perform the clustering algorithms, it is necessary to have some amount of statistical data about the waveforms in order to classify them. The statistical features originally determined to be clustered on was the standard deviation of a wave, the wavelength, and the time between waveforms. In order to determine distinct waveforms from a continuous signal stream, a method was created to check the RSSI throughout transmission of a packet and end capturing the waveform when the RSSI had remained below 75% of the maximum value found within the wave for an extended period of time. This method separated waveforms with a decent accuracy in a controlled environment, but a number of 0 values read during each wave caused the algorithm to be flaky, often dividing waveforms into pieces, and significantly reduced the accuracy of the statistical data. Therefore, as each packet was scavenged, the 0 values were stripped, leaving only significant data points. As the 0 values were randomly distributed over a multitude of values, stripping them did not cause any negative effect on packet size or standard deviation, but rather smoothed the waveforms, allowing for better analysis.

Additionally, in attempting to create distinct values for wavelength and for time between graphs, it became difficult to determine precisely when to start and end each waveform. A sudden noise spike could create a couple data points and trigger the start of a waveform erroneously. Similarly, determining when a waveform has run its course and contained no more
data was imprecise, and could result in lost accuracy. Therefore, it was determined to combine
the wavelength and dead time statistics as each wave was periodic in nature, and adding the dead
time into the wavelength standardizes each waveform with short tail periods before and after
each wave. While not optimal, combining the two statistics does not dramatically reduce the
responsiveness of the algorithm, but depending on the strength and timing of interference, may
skew the results of the first Wi-Fi frame in a sequence, causing it to be recognized as noise.

This was performed on the gathered training data for both Bluetooth and Wi-Fi, and the
results were then used in the development of the Packet Classification Algorithm. The
identification algorithm is also used in the Wi-Fi wake-up module in a modified capacity using a
linear difference statistic as opposed to the standard deviation for reduced calculation time and
reduced memory requirement. Analysis of the training set showed that the linear difference
increased accuracy as detailed in the results section.

### 3.3.3 Packet Classification Algorithm

Once a packet has been successfully parsed from the energy stream, its statistical
properties must be determined so that it can be classified according to the cluster analysis
algorithm. Breakdown of the waveforms showed that Wi-Fi waveforms were generally less than
300 data points in size (corresponding to approximately 0.5ms) and had a standard deviation of
RSSI between 74 and 80 (when using the byte representation of the signal strength, which can be
found on pages 95-96 on the datasheet for the ZigBee module [5]). Bluetooth, however, had
wave sizes generally between 600 and 1500 data points (~1ms – 2.5ms) and an RSSI standard
deviation between 54 and 78. This split has some amount of overlap in standard deviation, but a
clear cut division in wave size. A combination of the two provided about a 95% accurate split in
the training set using a J48 decision tree using the Weka data mining tool (figures 7 and 8).
Figure 7: Wi-Fi and Bluetooth waveforms visualized by linear difference and wave size

However, standard deviation requires storing a large amount of data for each wave, and thus, a linear difference \(((\sum(|SignalStrength \,−\, PreviousStrength|))/(WaveSize \,−\, 1))\) was used to accommodate for the limited stack sizes provided by microchip embedded systems. This result provided an improved split at an extremely reduced time and memory necessary. For the training data, Wi-Fi maintained a linear difference under 26, while Bluetooth was consistently above 30 with a median closer to 65. The J48 tree for the training set using a linear difference modifier produced a 99% accurate split.
3.4 Implementation

In order to create the system described above, a prototype was created using a low power PIC microcontroller with a built in SPI interface and several general I/O pins. The ZigBee transceiver used was the MRF24J40, which provides a detailed datasheet for ease of use purposes. The MiWi ZigBee stack was leveraged using the MCC18 compiler available through Microchip in order to easily interface with the ZigBee transceiver [19]. Utilizing the C compiler also allowed for quicker code creation in a testing environment as the algorithms were not established until late in the experimentation phase.

The algorithms described above were implemented using C syntax for the following pseudo code:
3.4.1 Packet Identification Pseudocode

This algorithm runs through the entire RSSI stream, determining where to distinguish individual waveforms. Input variables are: the RSSI stream, which may be read in the algorithm in real time directly from the ZigBee module in a live system, tailSize, which represents the number of packets read below the threshold before terminating the current waveform, and Thresh, a decimal value between 0 and 1 which adjusts the threshold to different levels of sensitivity. For the prototype, tailSize was set to 30 while Thresh was maintained at 0.75. The output is two integer arrays which give the start and end packets of each successive waveform.

```
PacketIdentify(int[] RSSI, int tailSize, float Thresh, int[] &Start, int[] &End){
    int i = 0;
    int start = 0;
    for each(RSSI){
        int Max = 0;
        int count = 0;
        int belowThreshCount = 0;
        while(belowThreshCount< tailSize){
            if(RSSI[count] > Max) Max = RSSI[count];
            if(RSSI[count] < (Thresh)Max)
                belowThreshCount++;
            else
                belowThreshCount=0;
            count++;
        }
        Start[i]=start;
        End[i]=start+count;
        Start += count;
        Start++;
    }
}
```
3.4.2 Packet Classification Pseudocode

This algorithm determines whether or not a given frame is a Wi-Fi frame. The input variables are: the RSSI stream for the frame, determined by the Packet Identification algorithm, the linear difference threshold, an integer value determined by the J48 cluster analysis, and the packet size threshold, also determined by the J48 tree. These threshold values are left as input variables instead of constants because the packet size is dependent upon the clock frequencies of the microcontroller and the ZigBee transceiver, while the linear difference is dependent upon the unit used in determining the RSSI. While the methods utilized in this paper were implemented using the byte representation of the RSSI determined by the ZigBee transceiver so that redundant conversions would be avoided, the actual unit for RSSI is dBm (decibels referenced to one milliwatt). The output value is simply a Boolean value which determines whether or not the received frame belongs to a Wi-Fi radio.

Bottom threshold values could be used to guard against noise and therefore decrease the false positive rate. However, in experimentation, noise packets were rarely below the Wi-Fi threshold values, but a partial Wi-Fi frame could easily be below the packet size threshold. Rather than analysis at this point to reduce false positives, a redundancy check was used to determine whether an individual was searching for wireless access. This was implemented as a state machine, with a positive value only being produced with three consecutive Wi-Fi frames or four Wi-Fi frames out of five consecutive frames.

```c
PacketClassify(int[] PacketRSSI, int linDiffThresh, int packetSizeThresh, bool &isWiFi){
    isWiFi=false;
    int packetSize = 1;
    int linDiff = 0;
```
for(int i=1; i<RSSI.size(); i++){
    packetSize++;
    linDiff += (RSSI[packetSize]-RSSI[packetSize-1]);
}
linDiff /= (packetSize-1);

if(linDiff < linDiffThresh)
    if(packetSize < packetSizeThresh)
        isWiFi=true;
}
4. METHODOLOGY, RESULTS AND ANALYSIS

The prototype developed through experimentation had several specific design goals which were necessary to meet. First, the system needed to be responsive. The window of opportunity to connect to a user may be relatively small, and the start-up time for the Wi-Fi radio interface represents a majority of the overhead. Second, the system needed to be low power. This was achieved by use of the ZigBee radio interface and a low-power microcontroller, and keeping the Wi-Fi interface asleep as often as possible. Lastly, the system needed to limit false negatives, as the main goal of the mesh network is to distribute emergency information to users. If a user cannot discover the network, then the system fails in its goal. This was achieved by accepting frames with lower precision, but maintaining accuracy by a redundancy check. This functionality was chosen due to the periodic nature of the Wi-Fi waveform, allowing the system to take a slight hit on speed in order to maintain accuracy while avoiding false negative responses. The system above was developed with these parameters in mind.

4.1 Methodology

In order to test the system described, two testing procedures were developed:

For the first test, a program was created for the microcontroller which would display an LED when the system detected Wi-Fi, display a separate LED when the system detected Bluetooth, and display no LEDs when the system detected indeterminate noise or radio silence. This test was developed in order to demonstrate whether the system could still detect Wi-Fi in the presence of other sources of interference.

The second test was a specifically numerical test in order to determine the speed of the algorithm in practice. In order to achieve this, the detection algorithm sent a signal through the...
serial interface upon receipt of the first non-zero data point, and then a second signal was sent upon determination of Wi-Fi activity. A program was developed in Processing which would receive these signals and timestamp the receipt of each as a comma separated pair so that the difference could be calculated [20]. The trial was run in a controlled environment with one end-user device constantly scanning for wireless access. This was repeated for 1000 iterations of the algorithm in order to maximize the accuracy of the statistical data.

4.2 Results

The first test proved the accuracy and robustness of the system in a noisy environment, as it visually proved that the system could determine the source of radio interference within a few seconds with several different end devices.

The second test demonstrates the responsiveness of the system as a function of the speed. The following information was collected from the 1000 data points (all values are given in milliseconds):

<table>
<thead>
<tr>
<th>Maximum Value</th>
<th>21172</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Value</td>
<td>172</td>
</tr>
<tr>
<td>Mean</td>
<td>1124.824</td>
</tr>
<tr>
<td>10% Trimmed Mean</td>
<td>1009.044</td>
</tr>
<tr>
<td>First Quartile</td>
<td>984</td>
</tr>
<tr>
<td>Median</td>
<td>1015</td>
</tr>
<tr>
<td>Third Quartile</td>
<td>1032</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>38.203</td>
</tr>
</tbody>
</table>
The following is a histogram of the data, showing the distribution of the response time in milliseconds.

![Histogram: Response Time](image)

**Figure 9: Histogram displaying the frequency of the response time of the algorithm**

### 4.3 Analysis

The data given above suggests that the algorithm will determine a user seeking Wi-Fi activity in approximately 1 second. Out of the 1000 trials, only 19 took longer than 2 seconds to determine access, and only 12 trials took more than 3.5 seconds. If one assumes that all 12 of those trials miss the set-up time required, resulting in a false negative, then the false negative rate in a controlled environment is 1.2 percent. This in combination of a median response time of 1 second meets the design goals provided.
These results demonstrate that the prototype system would act as a functional wake-up controller for the emergency mesh network. Further work is needed in order to properly obtain a false positive rate; however, the makeup of the radio space in an emergency scenario is unknown, and any testing environment would be speculative.
5. CONCLUSIONS

5.1 Summary

This thesis introduces the idea of a Wi-Fi sensing wake-up controller, and develops a prototype for the system from proposed design specifications. 802.11 and 802.15.4 protocols are reviewed, as well as a priori cluster analysis in order to properly demonstrate the methods utilized for determining Wi-Fi activity. A prototype is then developed exploring the hardware and software options for the system. The hardware is discussed in a high level format, with analysis on several design options for system implementation. The software for the system is developed from a high level specification, and a sequential analysis of the creation of sensing algorithms is discussed and pseudocode is provided for implementations of each algorithm. The prototype system is then tested and analysis on the system is provided as a function of design goals. The results demonstrate that the prototype system meets the design goals, while analysis is given on possible improvements to the system. The entire implementation is provided in a format that promotes user configuration because the system is application specific, and tweaks to the system are necessary in order to provide a configurable scheme for multiple applications.

5.2 Contributions and Potential Impact

The major contribution of this system is as a power saving Wi-Fi sensing option for a proposed solar-powered emergency communications mesh network. This system is an integral part of that network, as a mesh node would quickly drain any scavenged energy using a standard Wi-Fi radio. Therefore, the potential impact of this thesis is developing a prototype which provides an interface allowing an emergency network to remain alive for a longer period of time,
granting the ability to deliver critical information to more survivors; information that could save human lives.

The concepts developed within also serve as a proof of concept for possible social applications to be built on top of the mesh network provided. This will require some tweaking of the waveform selection methods, possibly selection of more statistical features, in order to grant maximum accuracy in a high noise environment.

The prototype provided is but one specific implementation of the concept which was explored. As this specific use of a ZigBee transceiver as a wake-up controller for a WAP is application specific, the usefulness of the system developed outside of its intended purpose is limited. However, the ability to sense a frequency and determine the cause of signal can be extended for several different interfaces. Using an approach similar to ZiFi, an integrated ZigBee module in a client device could act as a wake-up controller upon sensing Wi-Fi access or a client attempting to gain Bluetooth connectivity, therefore acting as a dual purpose battery-saving option. Combining this purpose with the ability to then communicate directly with SmartGrid devices within the home could be enough driving force to see a similar product implemented in consumer devices [21].

5.3 Future Work

The primary branch of continued work on this project will be integrating it with the prototype mesh network to be deployed later this year. The hardware specification for the network has shifted since the beginning of this project, and therefore changes will need to be made to the wake-up controller in order to provide a shared interface to the ZigBee module, and to properly interface with the new microcontroller. After installation of the prototype, statistics
can be gathered about the system’s behavior in a live environment, which may detail changes needed to be made to the parameters of the algorithms in order maintain a more sensitive or less sensitive sensing interface.

Current progress is being made on a system which utilizes a solar-powered MSP430 and low power GPS receiver to create a method for generating a geographical map of Wi-Fi activity. This will be implemented using the methods given above, optimized for beacon frames of WAPs, and creating comma-separated pairs giving the confidence interval that Wi-Fi access is present and the geographical location. Using the Google Maps API, one could then provide a visual interface which details the Wi-Fi availability of a given location. This particular interface would be good for a business or university setting wishing to distribute a map of wireless access to its user base.

Further work is needed to explore other implementations of similar sensing algorithms for other protocols, and their feasibility as an interface for an end-user device. One possible implementation is the device described above which senses for Wi-Fi access or Bluetooth Clients, acting as a dual-purpose wake up controller. Allowing the ZigBee transceiver to interface with SmartGrid appliances for home-automation applications would further increase the usability of such a system. A prototype system that interfaces with current Smartphone showing the increased battery life and demonstrating connectivity with a SmartGrid application would be a step forward in producing consumer level electronics with a built in ZigBee interface.
6. REFERENCES


