DIAGNOSING WI-FI ASSOCIATION ISSUES
AND OPTIMIZING NETWORK
PERFORMANCE

By

SUMA SUBBARAO

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

MAY 2013

ACKNOWLEDGEMENTS

It is always a pleasure to thank those people without whom it would have been impossible to complete my thesis. First and foremost, I would like to express my sincere gratitude to my supervising professor Dr. Yonghe Liu for being a great support and guide, throughout my research work. His valuable innovative thinking and timely advises have been a constant motivation to complete my thesis work.

I would also like to thank Dr. Ioannis Schizas and Dr. Alan Davis for their valuable support and interest in my research work. I sincerely appreciate them taking time to serve on my thesis committee and providing me with feedbacks without which my thesis work would not have been complete.

Finally, I would like to thank my parents, family and friends who have been on my side throughout my masters, helping me with all my endeavors at University of Texas at Arlington.

April 11, 2013

ABSTRACT

DIAGNOSING WI-FI ASSOCIATION ISSUES
AND OPTIMIZING NETWORK
PERFORMANCE

Suma Subbarao, M.S

The University of Texas at Arlington, 2013

Supervising Professor: Dr. Yonghe Liu

Extensive deployment of Wi-Fi networks at homes and public places has triggered research interest in several aspects of Wi-Fi technology like deployment strategies and optimization. People use Wi-Fi or 802.11 networking, to connect their computers at home or office for instant internet access and it is a common scenario to have faced association and network congestion problems. Finding appropriate solution to these issues is a challenging task. In this thesis, an attempt has been made to solve Wi-Fi related concerns through a novel approach.

Diagnosing problems occurring during the association process with the network and getting better throughput from the Wi-Fi connection are the most common Wi-Fi related concerns for a common user. Several diagnostics tools and network performance monitors have been developed in this direction. An efficient diagnosing and optimizing solution has been discussed in this thesis to improve user's experience with Wi-Fi networks.

Choosing a good channel for the transmitting access point is important in a crowded network with numerous access points occupying the Wi-Fi frequency band. Also, in any large work area with distributed wireless routers, the devices have to connect to an access point which can deliver better throughput. The channel assessment and AP selection algorithm discussed in this thesis aims at addressing this concern. A novel approach to predict channel occupancy using beacon shift analytical model, calculate the maximum achievable throughput and compare the performance of a channel or an access point on the channel is proposed. The performance of the algorithm with respect to channel selection has been experimented. The diagnostics framework has been implemented on Android and the channel assessment algorithm has been tested for accuracy and can be integrated into the software.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

AES: Advanced Encryption Standard

AP: Access Point

API: Application Programming interface

BSS: Basic Service Set

CCMP: CCM mode protocol

CSMA/CA: Carrier Sense Multiple Access/ Collision Avoidance

EAP: Extensible Authentication Protocol

MIB: Management Information Block

NIC: Network Interface Card

RADIUS: Remote Authentication Dial In User Service

RSN: Robust Security Network

RSSI: Received Signal Strength Indication

SINR: Signal to interference and noise ratio

SSID: Service Set Identifier

STA: Client workstation

TKIP: Temporal Key Integrity Protocol

WPA: Wi-Fi Protected Access

CHAPTER 1

INTRODUCTION

1.1 Motivation

Wi-Fi networks have become popular for several reasons. Simplicity in planning and deployment and inexpensive infrastructure are a few important reasons. People use Wi-Fi at home or office for instant internet access. Most of the modern equipments like laptops, Smartphone, tablets and gaming consoles have Wi-Fi capability. Mobile Internet, mobile applications and services are majorly used in day to day life for being constantly connected to the network. As long as the equipments have wireless adapters, devices can use one router to connect to the Internet. This connection is convenient, virtually invisible and fairly reliable.

Home networks mostly use wireless routers or access points with basic authentication schemes. The coverage and capacity provided by these routers is reasonable. They are inexpensive and easy to install. Enterprise deployments usually employ more expensive equipment in order to get higher coverage and security for the network.

Wi-Fi technology has been growing at an extensive pace. New specifications and enhancements have been added to improve performance, satisfy the need for higher data rates and quality of service. With these enhancements, the complexity has increased multi-fold due to which it is difficult for the common users to understand the nuances of the technology.

The increased complexity with respect to Wi-Fi network deployment, association and optimization has motivated several research proposals in the engineering community. Efforts have been made to provide software and hardware based solutions to improve Wi-Fi performance. This thesis is trying to provide one such solution to diagnose association and performance issues faced by the user.

Association is a process of establishing connection with the wireless router or access point through either open or shared authentication mechanism. The client device and access points have to exchange encryption keys in order to make the data transaction secure. Users have to enter proper credentials to successfully authenticate and associate. Home networks employ simple security schemes like Wired Equivalent Privacy (WEP) or Wi-Fi Protected Access (WPA) with passphrase. In enterprise networks, higher security schemes are employed for improved security. Configuring a Wi-Fi profile for an enterprise network involves configuring client and server certificates, specifying the correct EAP method and Phase2 authentication method. Different schemes have different configuration requirements, which make it more complicated for the common user. The association process may be disrupted due to issues pertaining to driver initialization or a hardware glitch.

Table 1.1 An overview of authentication and encryption schemes used in different Wi-Fi security protocols

	Encryption	Authentication
WEP	Pre-shared key	Open Passphrase – 40 or 104 bits

WPA Personal	TKIP	Passphrase
		ASCII(8 – 63 characters)
		HEX(64 characters)

WPA2 Personal	AES-CCMP(default)
	TKIP
		PSK
		ASCII(8 – 63 characters)
		HEX(64 characters)

WPA Enterprise	TKIP	802.1x/EAP
WPA2 Enterprise	AES- CCMP(default) TKIP	802.1x/EAP

There are many factors which determine network performance like quality of service, throughput and delay. There are several specific parameters which can help tune the Wi-Fi network for required performance. It is difficult to choose these parameters for any given scenario. Also, factors like channel assignment and selecting an access point which can deliver better throughput under given conditions affect the user's network experience. In Wi-Fi, there are a limited number channels available in each frequency band. Access Points (AP) in a home environment are surrounded by APs transmitting on the same channel. There will be spectral overlapping between the APs due to a limited number of channels. This causes both co-channel and inter-channel interference. An identifier called Service Set Identifier is assigned to each AP. In any enterprise network, multiple APs with the same Service Set Identifier (SSID) are installed to cover the work area. In such cases, it is necessary to choose APs according to the available capacity.

The concerns explained previously are the motivating factors behind this thesis work. Through this thesis, an approach to design and implement user-friendly solution to address these concerns is explored.

Android platform was chosen to develop the idea of providing the software solution to diagnose association issues and optimize Wi-Fi networks. The diagnostics solution can be extended to other operating systems satisfying required conditions and the channel assessment algorithm can be integrated into the tool further. Android is a popular operating system for tablets and mobile devices. Android is an open source operating system. It is inexpensive and easy to find devices and tools for experimentation. The Android Wi-Fi stack is a modified version of the Linux Wi-Fi stack. The Java based application framework is well designed and user friendly. It is easy to develop a user interface, and to use the Wi-Fi application programming interface (API).

1.2 Related Work

Several Wi-Fi diagnostics and network monitoring solutions are available, which use proprietary implementation to provide assistance to the user. This section discusses a few popular designs and algorithms targeted for solving Wi-Fi related concerns of interest for this thesis. The common association diagnostics tools on operating systems like Windows help diagnose issues like password errors, Wi-Fi card and backend errors. There are a few applications for Android devices as well which analyze Wi-Fi related issues but they are mostly related to monitoring connection loss and reporting connect/disconnect events. All these tools use APIs exposed by the underlying Wi-Fi framework to collect events and failure codes. Only surface level debugging is possible due to limitations in the API framework.

The diagnostics framework design is based on analyzing system logs to detect issues. System logs like windows event logs or Linux system logs are an important part of the respective operating system management. They contain information generated by different applications and processes. [11], [12] are some research work based on using system or network logs to diagnose issues. Steven Melvin [11] proposes a technique for endpoint identification in the internet cloud using network logs. A similar approach is followed in which the system logs for Wi-Fi events and messages are analyzed.

In order to achieve good network performance, analysis of channel performance, interference modeling, Wi-Fi deployment strategies are important. With dense Wi-Fi deployments, algorithms to manage network load and performance become important to keep up with the demands. These strategies include dynamic channel assignment, AP load balancing, Centralized network monitoring.

Papanikos and Logothetis [8] have studied approaches for channel assessment from the AP and the client device. In their work, they assume some parameters like the number of stations connected to neighboring APs and mean received signal strength indicator (RSSI) to be included in the beacons or probe response which can be used by the stations (STA) to choose AP.

Dynamic channel assignment algorithms such as in [15] include parameters like AP traffic load, Noise, interference and signal strength. The algorithm presented here is closely related to the techniques in [18]. Murty et al [18] have proposed a method to design high performance Wi-Fi networks in which the channel condition is measured using the ‘free air time’ parameter. They use a technique called probe-gap to measure channel occupancy. An idea to use beacons to measure the channel occupancy instead of transmitting probes to determine channel load is proposed in this work. Ferguson and Fonseca [17] use timestamps from Internet protocol (IP) packets to deduce router statistics to calculate bounds for rate of UDP traffic.

Similarly, an argument, that the beacon shift can provide useful information regarding the channel quality, is made.

Kim et al [35] have implemented android application for choosing candidate access points based on signal to noise ratio as the criteria for ranking each AP. The algorithm presented here considers several factors for choosing access points based on throughput prediction. It is not currently supported on the android software presented here but can be integrated similar to the method implemented in [35].

1.3 Contributions

The goal of this thesis work is to address association and optimization issues in Wi-Fi networks with a novel approach. In this direction, a solution to effectively diagnose and optimize Wi-Fi performance through channel selection is proposed. To this end, the software based debugging and optimizing algorithm is convenient and accurate. This approach is implementable on any Wi-Fi capable laptop or mobile device without any modification to the existing infrastructure. The software design presented in this thesis work is targeted for android devices and implements log based analysis of the association issues to accurately detect common Wi-Fi association issues unlike other API based approaches. However, the diagnostics framework can be extended to other operating systems satisfying certain requirements. The software solution also provides assistance to the user in choosing parameters to tune a Wi-Fi network.

In order to provide assistance in choosing a good channel for the home routers, an algorithm is proposed, to select a better channel, using a novel concept of beacon shift and throughput prediction. The channel assessment theory and AP selection algorithm is not currently implemented in the version of the software solution demonstrated but can be extended to do so. The beacon shift phenomenon, when an AP transmits beacons in an environment with interference has been studied. Experiments were conducted to study this behavior with different loads on the AP. A mathematical model is used to show that beacon shift and other parameters can be used to calculate the probability of a channel being busy. Using the channel occupancy parameter, a technique to determine achievable throughput for a connection has been demonstrated. The experimental results show that the algorithm performs well in predicting throughput and helps in choosing a good channel or an AP in a large work environment.

The proposed access point selection method includes the effect of several factors like channel occupancy, number of access points on the channel, signal to interference ratio, modulation and coding schemes to accurately determine the achievable throughput if the client associates with it. The algorithm is simple and all the parameters considered are available with a client device unlike other approaches mentioned in section 1.2 and hence, with suitable modifications, can be implemented on the client side without changes to the infrastructure. It can be integrated with the android software demonstrated here and will be useful for common users to associate with a good AP or determine a good channel for their wireless routers using the client android devices.

The following chapters demonstrate the above concept in detail. In chapter 2, the design approach and implementation details of the diagnostics tool including a brief description of Android Wi-Fi stack and the libraries used is explained. Chapter 3 deals with the best channel assessment algorithm and AP selection method. The analytical model, throughput calculation technique and experimental method are presented in this chapter. This thesis is concluded by discussing results and future work.

CHAPTER 2

DIAGNOSTICS TOOL

2.1 Application design

As explained earlier, the software solution for android aims at providing assistance to the user in diagnosing association issues and optimize Wi-Fi network performance. The current version of the software demonstrated in this thesis implements the log analysis based diagnosis of association issues into which the optimization algorithms explained in chapter 3 can be integrated. It is a Java based application installable on any android device. This application is targeted for rooted Android devices. Android rooting is the process of allowing users of tablets or mobile devices to gain privileged control (known as "root access") within Android system. It is similar to accessing administrative permissions on Linux. The application is compatible with both mobile devices and tablets. The number of users switching for rooted Android devices has increased. Features like enabling Wi-Fi hotspot or attaching USB memory sticks to tablets are possible with rooting. There are one-click rooting software available and hence it is not difficult to root an Android device. Rooted devices can be easily reverted to factory condition.

The diagnostics framework is designed to investigate possible problems with the Wi-Fi connection and provide feasible solution, if exists, based on our understanding of the issue. It may not always be straight forward to detect and fix all the problems but common user errors fall into solvable category if an exact cause is known.

2.2 Diagnostics framework

In order to debug issues, it is required to collect data regarding events and procedures during the connection process. The Wi-Fi APIs exposed by the Android framework provide only surface level information regarding these events. The APIs do not always return a valid reason code for all the disconnect events. The logging framework provides sufficient information to debug issues with an ongoing connection.

There has been considerable amount of work on the concept of building tools which use system or network logs to automate detecting issues and failures. The design of this tool is also based on the same logic of using the Android system logging framework, extracting information about the Wi-Fi events and deducing the cause of errors or issues during the connection process. In order to completely analyze the issues, an understanding of the both the logging framework and the Android Wi-Fi stack are necessary.

2.3 Android Wi-Fi stack

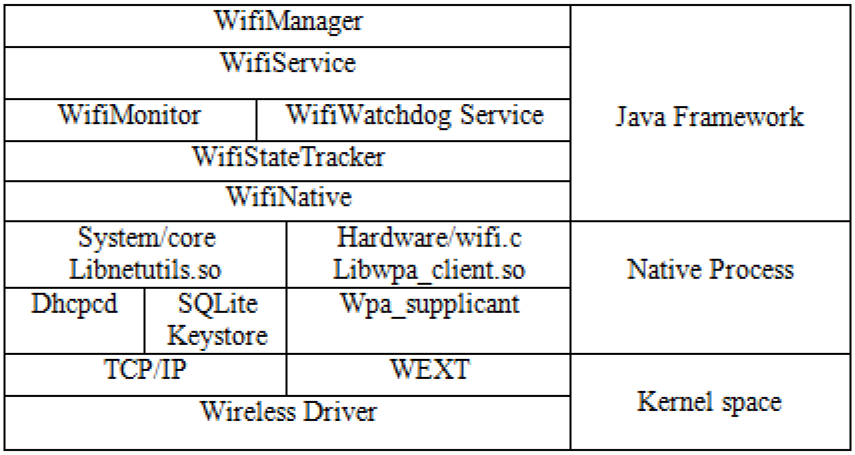


Figure 2.1 - Android Wi-Fi framework

Android Wi-Fi framework consists of Linux kernel and supplicant on top of which is the Android API framework that gives the ability to write Wi-Fi based applications.

2.3.1 Supplicant

Enterprise Wi-Fi networks use an advanced scheme called IEEE 802.1x authentication. Supplicant is the main IEEE 802.1X security protocol component. It implements key negotiation with a Wi-Fi Protected Access (WPA) Authenticator or the RADIUS server. The RADIUS stands for Remote Authentication Dial In User Service. It is a network protocol which provides centralized authentication and authorization management for devices to connect to a network. The supplicant on the client device controls roaming and IEEE 802.11 authentication/association with the RADIUS server. Supplicant, termed the Wpa_supplicant, runs in the background and acts as the backend component controlling the wireless connection. There are two programs called wpa_cli and wpa_gui which allow access to some of the features like increasing the debug log level and setting few MIB fields (management information block fields).

2.3.2 Wi-Fi APIs

The Wi-Fi APIs or application programming interface provides a means by which applications can communicate with the lower-level wireless stack that provides Wi-Fi network access. Information from the device supplicant is available through these APIs, including the connected network's link speed, IP address, negotiation state, and information about other networks are available through Scan APIs. Some other API features include the ability to scan, add, save, terminate and initiate Wi-Fi connections. The API framework is in Java and some of the main classes are WiFiConfiguration, WiFiManager and WifiInfo. Applications are implemented using the Android software development kit (SDK) which provides API libraries and tools to build, debug and run. These APIs can be used to obtain information like SSID, signal strength and channel about the scanned access points.

2.3.3 Native libraries

The native libraries are mostly hardware abstraction libraries which include libraries like libnetutils.so and libwpa_client.so. They are written in c/c++ and can be directly used by certain types of applications which are implemented using the Android NDK (native development kit).

2.3.4 *Wireless extension*

Wireless extension or WEXT are the extensions added to the Linux kernel. They are the wireless APIs which facilitate accessing wireless drivers beneath in a uniform way irrespective of different hardware implementations. Wireless extensions have 3 complimentary parts. The first part is the user interface, a set of tools to manipulate those extensions. The second part is a modification of the Linux kernel to support and define the extensions. The third part is the hardware interface and is implemented in each network driver itself to map the extensions to the actual hardware manipulations. However, Wireless extension is becoming obsolete with cfg80211, the new Linux wireless configuration API. nl80211 is used to configure a cfg80211 device.

2.3.5 *Wi-Fi drivers*

These drivers are hardware specific and given by the third party chipset vendors. These drivers implement the medium access control layer and physical layer algorithms.

2.4 Android logging framework

The Android logging system provides a mechanism for collecting and viewing system debug output. Logs from various applications and portions of the system are collected in a series of buffers, which can be viewed and filtered by the ‘Logcat’ commands. The framework keeps multiple circular buffers for log messages, and not all of the log messages are sent to the default circular buffer. There are three types of buffers:

- Radio — Buffer contains radio/telephony related messages.
- Events — Buffer contains events-related messages.
- Main — Main log buffer.

Commands can be used to read either radio, events or the main buffer. Though the ‘radio’ buffer is the one which contains all the wireless related information, it was noticed that it actually contains only the events/states like Idle, authenticating or connecting. The ‘main’ buffer contains more detailed information and hence the main buffer is used to obtain information in the implemented design.

2.5 Software design

The diagnostics framework in the software solution is built on top of the Android logging framework explained in the previous section. There are two external binaries that are required by the application. They are wpa_cli and iwconfig. These Linux binaries can be used for configuring the wpa_supplicant and the Wi-Fi card respectively. They are command based and expose features like set/get parameters, increase verbosity and depth of logs. These are Linux binaries that were built for Android using the Android build environment and makefile for the application.

Wpa_cli is a text-based frontend for interacting with wpa_supplicant. It is used to query current status, change configuration, get some system information and set values like log level, connect to profile. wpa_cli can show the current authentication status, selected security mode, dot11 and dot1x MIBs, etc. Iwconfig is a binary which can be used to set parameters for the network interface which are specific to the wireless operation. For example: frequency, bit rates, transmits power etc. It can also be used to get Wi-Fi connection parameters like signal strength and link quality.

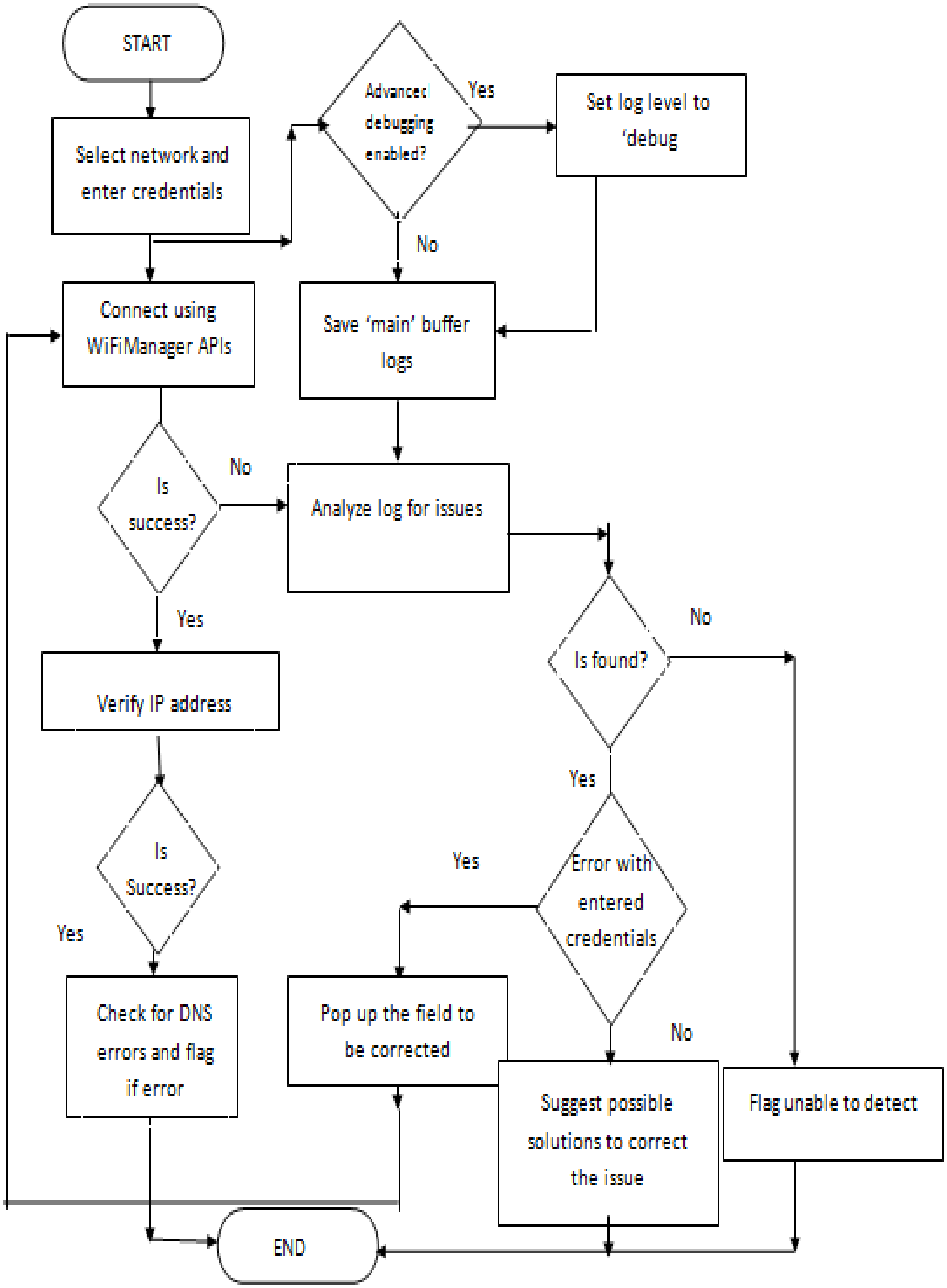

START
Select network and enter credentials
Advanced debugging enabled?
Yes
Set log level to 'debug
No
Connect using WiFiManager APIs
Save 'main' buffer logs
Is success?
No
Analyze log for issues
Yes
Verify IP address
Is found?
No
Yes
Is Success?
Yes
Error with entered credentials
Yes
No
Check for DNS errors and flag if error
Pop up the field to be corrected
Suggest possible solutions to correct the issue
Flag unable to detect
END

Figure 2.2 - Flow chart for the diagnostics framework

Figure 2.2 explains the flow of operation for the diagnostics framework. The application calls the logging framework to collect the data once the user creates or selects a Wi-Fi profile to connect to. With the connection happening in the foreground, the background threads keep track of the system logs and detect issues based on our analysis of those messages in the logs.

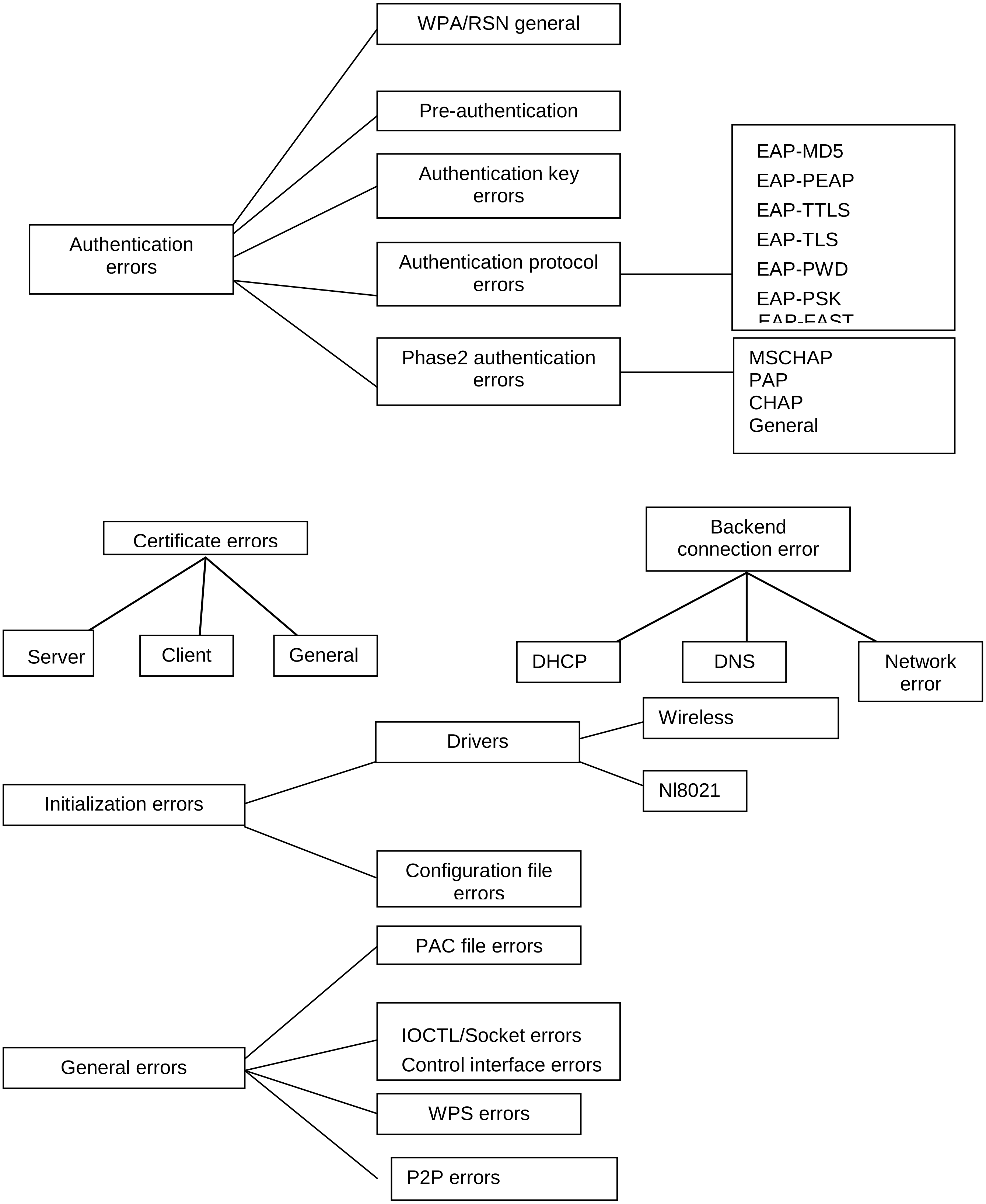
Based on the severity of the events, the log messages from the Android Wi-Fi stack will be marked as ‘Excessive’, ‘Dump’, ‘Debug’, ‘Info’, ‘Warning’ and ‘Error’. Excessive and Dump categories are not of interest since they are general logs helping to trace each and every step in the Wi-Fi state machine. By default, log level will be set to ‘Info’ and hence any logged event which belongs to information, warning or error category will be reported in Logcat. This is sufficient to debug major errors like password or certificate issues. However, there are other issues like configuration file errors which can be detected by increasing the log level severity. We use WPA_CLI binary to set the debugging level to ‘Debug’ using “wpa_cli log_level debug” command.

The framework has a main activity which helps in configuring a Wi-Fi profile and connecting to the network. The logging activity happens in the background with the logger module saving the logs collected during the connection. The error analysis is made using the following procedure.

The application checks for major Wi-Fi configuration issues that the user could make while creating a Wi-Fi profile. These are the most common issues like wrong credentials or certificates entered into the Wi-Fi profile and hence we prioritize these errors. An analysis of the Wi-Fi stack, protocols and features has been made to categorize all the issues based on the log messages. This categorization helps to identify the root cause for the issues. The classification is shown in Figure 2.3. It is easy to present to the user the category of error than the actual technical error. For example, if an error occurs during obtaining private key for the certificates, it is easy to explain to the user that the error is related to the included certificate rather than explaining the technical details of the error.

This type of classification not only helps the common user, it helps developers who want to develop or add features to the Android Wi-Fi stack. Classification is mainly based on what stage of the Wi-Fi connection process the error occurs. The problem with the connection may occur during driver or stack initialization, the authentication process or after the connection with the AP is successful but backend connection is not. Backend connection issues may be DHCP errors, errors with the DNS settings on the AP or problem with the internet service provider connection. Wrong DHCP or DNS settings on the AP are quite common. Along with information from the system log, we check if the device has a valid IP and if the device is able to ping any external server.

The advanced debugging feature to increase the log level severity and debug issues is used if the user chooses to obtain advanced level information or dig deeper in into the events. This technique is not used as a default method since it increases the size of the logs enormously. Using the ‘debug’ level by default has a lot of computation overhead.



WPA/RSN general
Pre-authentication
Authentication key errors
Authentication errors
Authentication protocol errors
Phase2 authentication errors
EAP-MD5
EAP-PEAP
EAP-TTLS
EAP-TLS
EAP-PWD
EAP-PSK
EAP-FAST
MSCHAP
PAP
CHAP
General
Certificate errors
Server
Client
General
Backend connection error
DHCP
DNS
Network error
Drivers
Wireless
Nl8021
Initialization errors
Configuration file errors
PAC file errors
General errors
IOCTL/Socket errors
Control interface errors
WPS errors
P2P errors

Figure 2.3 - Classification of issues on the Android Wi-Fi stack

Some Wi-Fi issues may not be logged as errors. Instead, a direct association reject or disconnect message is reported by the supplicant and these messages usually carry the reason code for failure if the access point reports the reason code. The reason codes sent along with the disassociation or de-authentication response and returned to the wpa_supplicant can be used to identify issues like capability or information element mismatch. IEEE 802.11 specification specifies standard reason codes (Table 2) to be sent by the access points or routers in their association or authentication reject messages. It is not mandatory for APs to implement these reason codes. But we make use of this for our debugging if the AP reports a reason code.

Table 2.1 Association response codes according to the IEEE 802.11 specification

Status	Meaning
0	Success
1	Unspecified failure
2-9	Reserved
10	Cannot support all requested capabilities in the Capability Information field
11	Reassociation denied due to inability to confirm that association exists
12	Association denied due to reason outside the scope of this standard
13	Responding STA does not support the specified authentication algorithm
14	Received an Authentication frame with authentication transaction sequence number out of expected sequence
15	Authentication rejected because of challenge failure
16	Authentication rejected due to timeout waiting for next frame in sequence
17	Association denied because AP is unable to handle additional associated STAs
18	Association denied due to requesting STA not supporting all of the data rates

Table 2.1 -- *Continued..*

19	Association denied due to requesting STA not supporting the short preamble option
20	Association denied due to requesting STA not supporting the PBCC modulation option
21	Association denied due to requesting STA not supporting the Channel Agility option
22	Association request rejected because Spectrum Management capability is required
23	Association request rejected because the information in the Power Capability element is unacceptable
24	Association request rejected because the information in the Supported Channels element is unacceptable
25	Association denied due to requesting STA not supporting the Short Slot Time option
26	Association denied due to requesting STA not supporting the DSSS-OFDM option
27-31	Reserved
32	Unspecified, QoS-related failure
33	Association denied because QoS AP has insufficient bandwidth to handle another QoS STA
34	Association denied due to excessive frame loss rates and/or poor conditions on current operating channel
35	Association (with QoS BSS) denied because the requesting STA does not support the QoS facility
36	Reserved
37	The request has been declined
38	The request has not been successful as one or more parameters have invalid values
39	The TS has not been created because the request cannot be honored; however, a suggested TSPEC is provided so that the initiating STA may attempt to set another TS with the suggested changes to the TSPEC
40	Invalid information element, i.e., an information element defined in this standard for which the content does not meet the specifications in Clause 7
41	Invalid group cipher
42	Invalid pairwise cipher
43	Invalid AKMP
44	Unsupported RSN information element version

Table 2.1 -- *Continued..*

45	Invalid RSN information element capabilities
46	Cipher suite rejected because of security policy
47	The TS has not been created; however, the HC may be capable of creating a TS, in response to a request, after the time indicated in the TS Delay element
48	Direct link is not allowed in the BSS by policy
49	The Destination STA is not present within this BSS
50	The Destination STA is not a QoS STA
51	Association denied because the ListenInterval is too large

After the error detection is made, if the issue pertains to one of the fields in the Wi-Fi profile, the applications indicates the field to be corrected. This makes it easier for the user to recognize the field and try to reconnect. If the problem is related to any of the settings on the AP or device side, methods are suggested to correct it. For example, if it is an association timeout, there can be multiple reasons for the issue. Small association timeout value or MAC filter may be the reason. Some driver issues may be fixed by just toggling the state of the Wi-Fi interface.

2.6 Tunable Wi-Fi parameters

In order to provide assistance to the user to choose parameters to optimize Wi-Fi performance for a given requirement, the software solution has been extended to suggest appropriate parameters to tune the Wi-Fi network. There are several Wi-Fi parameters available on the AP or STA which can be tuned for optimum Wi-Fi performance. It is important to understand each parameter in order to categorize them according to the requirement. These parameters have been widely studied. This is an additional feature in the application and the software simply suggests these parameters and information related to them based on the category chosen by the user. The following sections briefly describe the classification of these tunable parameters.

2.6.1 Optimizing performance for multi-user environment

When more users are connected to the same wireless router, simultaneous transmissions cause collision and data loss. This reduces throughput of the Wi-Fi network. Users will experience more latency or delay. The best possible solution for this issue is to move some users to a different router or different band if some of the devices support the transmission frequency. However, configuring a network with multiple routers and single internet connection may not be straightforward. Using multiple routers is an expensive proposition.

There are few parameters exposed by most Wi-Fi network interface cards (NIC) which can actually help mitigate the effect of collision in such environment. Fragmentation threshold indicates whether the packet is to be fragmented when the size of the packet is above a threshold. The probability of smaller packets undergoing collision is less compared to larger packets. Request to send or RTS threshold indicates if the device has to do a request to send/clear to send handshake, usually termed as Request to send/Clear to send (RTS/CTS) mechanism, for packets which are above RTS threashold.RTS/CTS protection mechanism is used usually by the devices to sense if the medium is clear to send packets to avoid collision. These parameters can be set on the wireless router interface as well fo downlink transmission.

The user can set the retry or retransmission limit to increase the number of MAC retransmissions the NIC should make. Having higher retry limit, the user can ensure his packets gets transmitted successfully. All these parameters add extra overhead on the network but are useful in high user density scenarios. Tolani et al [2] have explored the optimum values for maximizing throughput and delay performance for fragmentation and RTS thresholds

The 802.11 standard specifies two types of preambles in the physical layer. The access point can be configured to use long preamble in high collision environment since the long preamble transfer time is much longer than short preamble and the receiver gets longer time to synchronize using the sync bits in the preamble. Short preamble can be used only if the environment is noise free and if there are no 802.11b devices. Using short preamble improves throughput but reduces synchronization time available for the receiver and hence is not preferable in a noisy environment.

2.6.2 Optimizing performance for Video/multimedia streaming

Video is a very demanding application that immediately exposes any weaknesses in the network. When delay, packet loss, and jitter enter visible thresholds, the usefulness of video quickly drops to zero. WMM (Wi-Fi multimedia) is a feature included as part of IEEE 802.11e standard which defines quality of service implementation. Though the technical specification does not specify exact implementation of the WMM feature, the quality of service(QoS) requirements specify that video packets be treated differently with respect to the Medium access control (MAC) scheduling, data rate and retransmission. Wi-Fi alliance specification and studies in [7] indicate prioritization of video over data.

Some routers also provide the option to define quality of service priority for different internet applications. This QoS setup includes setting priorities to different types of activities like video applications like YouTube or Netflix, gaming, file sharing.

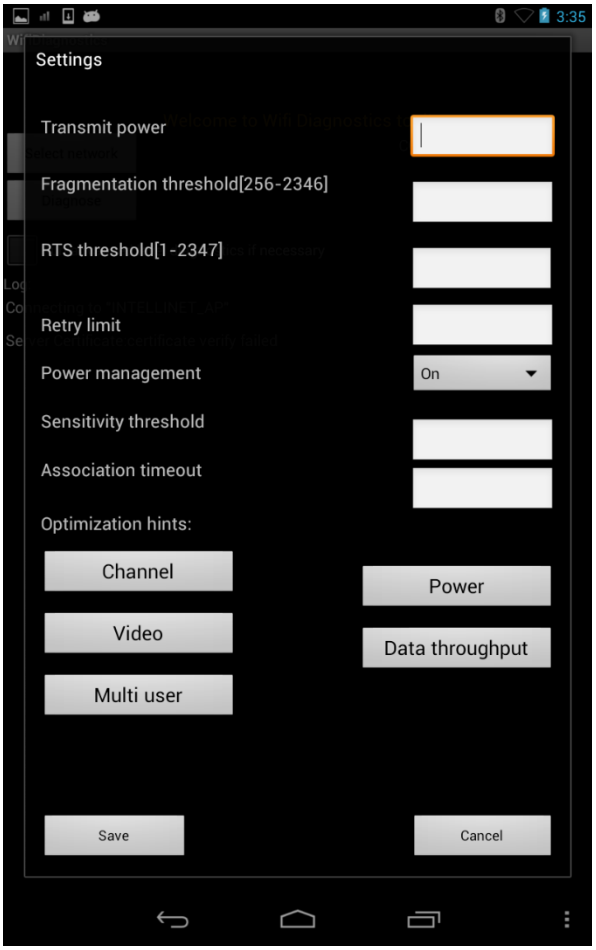
2.6.3 Optimizing performance for better data throughput

There are some parameters like short preamble and guard interval in 802.11n which can help get better throughput by reducing overhead. Short preamble can be used to reduce the number of bits as part of the sync field in the 802.11 preamble. However, this feature is not backward compatible with 802.11b devices. Also, placing the router in the right place is important to get maximum performance. Different 802.11 technologies behave differently according to their frequency range and receiver sensitivity. Walls, reflective surfaces and other obstructions reduce the throughput of the network drastically. The 2.4GHz band is also susceptible to interference from microwave and cordless phones. The 5GHz band has lesser coverage compared to the 2.4GHz band for the same transmit power.

2.6.4 Optimizing battery performance

Power management is important to get a better battery performance. All the devices which are connected to Wi-Fi will be constantly scanning and receiving management frames even though they are not doing any data transaction. The Delivery traffic indication message (DTIM) is used to deliver multicast or broadcast messages to all the stations. This can be set to a higher value on the AP if the user is not using any multicast applications so that the device wakes up less frequently. Transmit power on the AP as well as the device can be varied according to the environment according to acceptable transmit and receive quality. Default transmit power on the devices can be sometimes very high. Typical transmission power on laptops and tablets is 12-15dBm (32mw).

The above classification is included as an additional feature to help users identify parameters for different scenarios as shown in Figure 2.4. Another important factor affecting the network performance is the APs channel configuration. It is important to choose a Wi-Fi channel with less interference and less number of APs for better throughput. The study regarding best channel selection and algorithm to choose AP for association is described in chapter 3.



Transmit power

Fragmentation threshold[256-2346]

RTS threshold[1-2347]

Retry limit

Power management: On

Sensitivity threshold

Association timeout

Optimization hints:

Figure 2.4 – The settings menu; allows users to set parameters on the device as well as choose scenarios to get hints regarding the tunable parameters.

2.7 Experiments and results

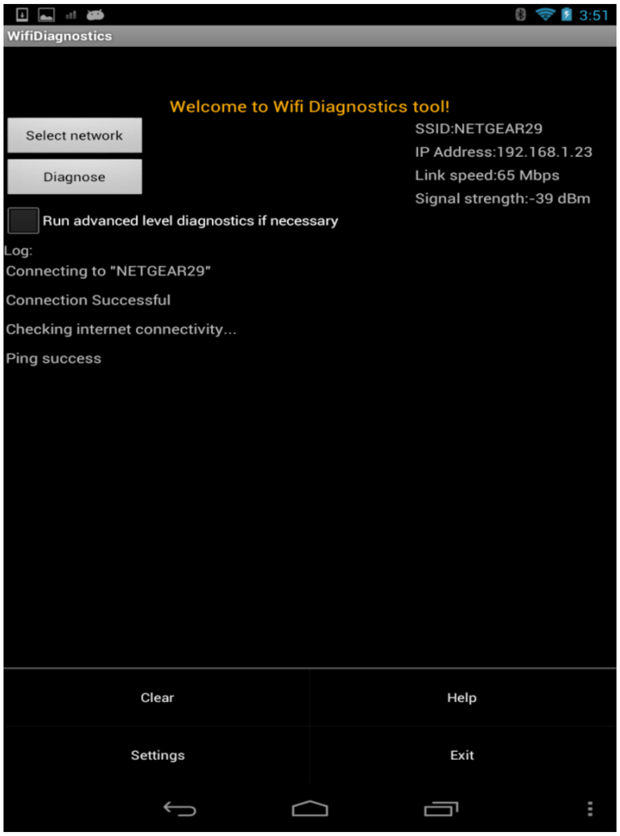
The diagnostics framework was tested on the ‘Nexus 7’ Android device and an 802.11n access point capable of doing the passphrase and the 802.1x authentication with internal and external radius server. An external radius server software ‘TekRadius’ is configured on a Windows laptop connected to the AP via Ethernet. The external radius server is configured for 802.1x authentication with PEAP MSCHAPV2 and self signed certificates. The application was tested for issues like wrong credentials, invalid authentication, encryption methods and certificates to determine if event logs were accurately captured and errors were correctly detected according to the analysis. The experiments also covered scenarios for detecting back end issues like DHCP and DNS error. Issues related to Wi-Fi driver or socket errors are very rare and hence are difficult to simulate. But the software is programmed to detect these errors based the analysis and classification of logs.

Backend issues like invalid IP configuration will be reported in logcat by the DHCP state machine. These messages are also considered for the analysis. However, ping commands are used to detect backend errors like DNS and gateway accessibility issues. Ping packets or ICMP packets in general are usually blocked in large enterprise networks like universities. Hence the application cannot detect these issues with packet filters.

As explained earlier, the debug messages have been internally classified into different categories in the supplicant and other modules according to severity of the errors and issues. The major association issues have been handled as high priority errors. The common user errors can hence be accurately identified. Upon identification, the application is designed to ask the user to correct the particular field in the profile. These errors include wrong authentication methods used in the profile, invalid certificates, invalid identity and password credentials and invalid private key for certificates.

The following are some example scenarios handled by the software.

Experiment 1: The AP and RADIUS server are configured. The SSID is selected through the menu and a successful association and IP configuration is verified by entering correct credentials and certificates.



Welcome to Wifi Diagnostics tool!

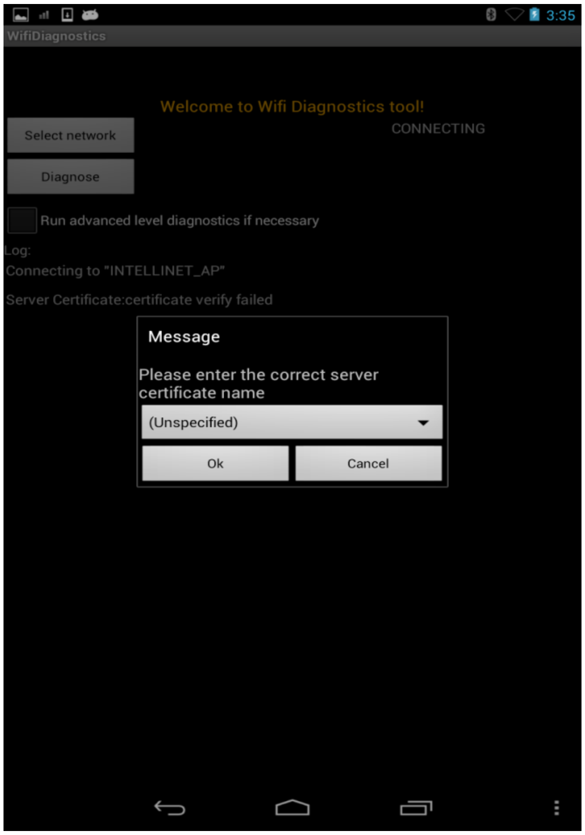
SSID:NETGEAR29
IP Address:192.168.1.23
Link speed:65 Mbps
Signal strength:-39 dBm

□ Run advanced level diagnostics if necessary

```
Log:
Connecting to "NETGEAR29"
Connection Successful
Checking internet connectivity...
Ping success
```

Figure 2.5 – Successful connection; SSID, link speed and signal strength is refreshed periodically

Experiment 2: The RADIUS server is configured to use a self signed server certificate ‘TEST’. The AP SSID is selected from ‘Select Network’ menu. The profile is configured with EAP method ‘PEAP’, phase2 method ‘MSCHAPV2’ and server certificate THAWTE from the certificate store. A ‘Thawte’ certificate is installed on the device for this purpose. From the log analysis, the application flags ‘Server certificate verify failed’.



Welcome to Wifi Diagnostics tool!

CONNECTING

□ Run advanced level diagnostics if necessary

```
Log:
Connecting to "INTELLINET_AP"
Server Certificate:certificate verify failed
```

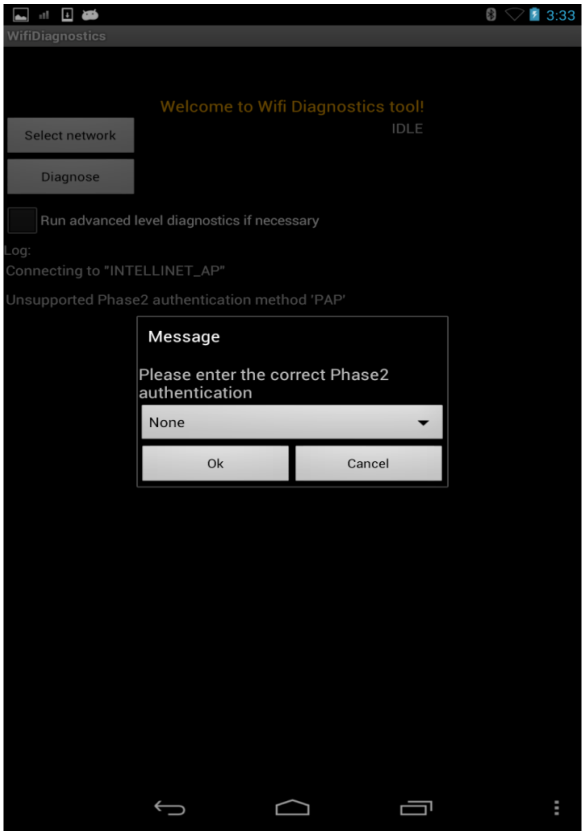
Message

Please enter the correct server certificate name

(Unspecified)

Figure 2.6 – Server certificate verification failure

Experiment 3: The phase2 authentication is used by many EAP methods like PEAP and TTLS. The default phase2 authentication used by TekRadius is MSCHAPV2. The above experiment is repeated with phase2 method ‘CHAP’ (Challenge handshake authentication protocol) from the user interface, the logs indicate that the phase2 method is not supported on the other end. The application returns ‘Unsupported phase2 EAP method’ and allows the user to choose a different method for the field. It is not possible to indentify the correct method from the logs (Figure 2.7). It is easier for the user to choose the appropriate field which is erroneous than filling all the fields again.



Welcome to Wifi Diagnostics tool!

IDLE

□ Run advanced level diagnostics if necessary

```
Log:
Connecting to "INTELLINET_AP"
Unsupported Phase2 authentication method 'PAP'
```

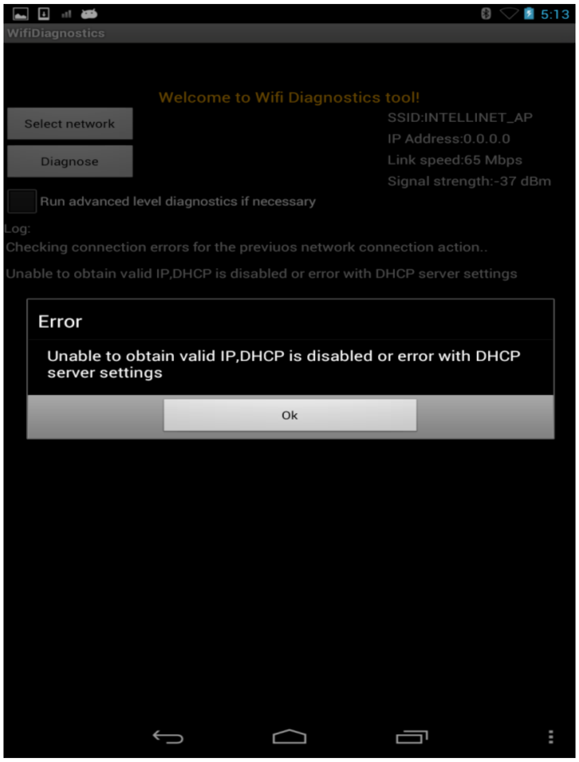
Message

Please enter the correct Phase2 authentication

None

Figure 2.7 – Phase2 authentication error scenario; the user will have to set proper phase2 authentication method for the application to try reconnecting to the AP.

Experiments 4: The DHCP server setting on the AP is disabled. The association with the AP will be successful but the DHCP state machine will flag ‘IP configuration failed’. In another scenario, the DHCP server setting on the AP is enabled and the IP range is set to an invalid value. Though the device is associated with the AP and IP address is assigned, ping failure can be detected to verify the IP address.



Welcome to Wifi Diagnostics tool!

SSID:INTELLINET_AP
IP Address:0.0.0.0
Link speed:65 Mbps
Signal strength:-37 dBm

☐ Run advanced level diagnostics if necessary

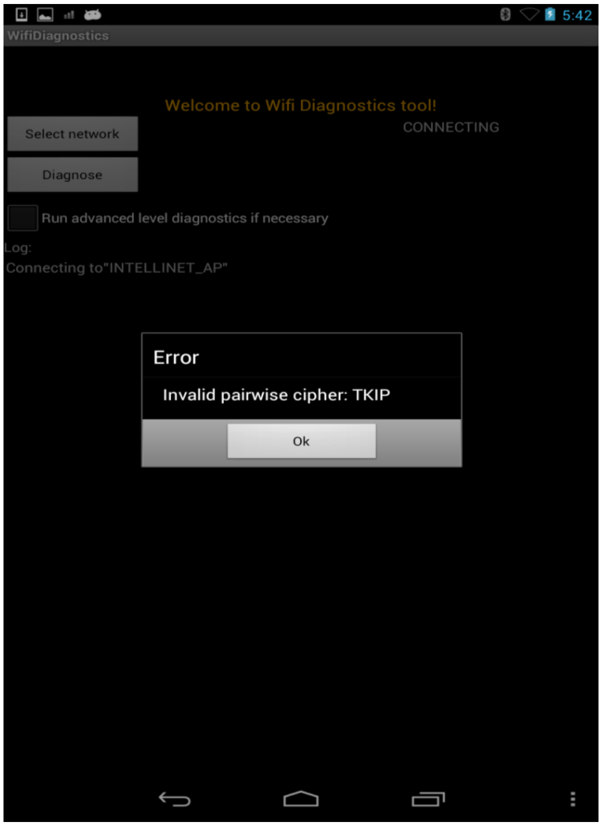
```
Log:
Checking connection errors for the previuos network connection action..
Unable to obtain valid IP,DHCP is disabled or error with DHCP server settings
```

Error

Unable to obtain valid IP,DHCP is disabled or error with DHCP server settings

Figure 2.8 – DHCP error when DHCP server setting not enabled on the AP

Experiment 5: The reason code mapping has been tested for scenarios in which the disassociation happens due to mismatch in the AP and the client capabilities. For example, if the client device uses TKIP encryption scheme and the AP is configured to advanced encryption system (AES) for pair wise ciphering, 'Invalid pairwise cipher' will be decoded from the reason code 42 associated with the disassociation message.



Welcome to Wifi Diagnostics tool!

CONNECTING

□ Run advanced level diagnostics if necessary

Log:
Connecting to"INTELLINET_AP"

Error

Invalid pairwise cipher: TKIP

Figure 2.9 – Capability mismatch decoded through the reason code mapping; pairwise cipher set to TKIP instead of AES.

From the above experimental results, it is clear that the diagnostics tool can debug common Wi-Fi issues according to the analyzed log messages. The extensive logging framework, provided by the android operating system, can be used to debug and solve issues in different parts of the system. The framework is designed to capture events and messages from every part of the system from kernel to applications. Wi-Fi stack is found to be a well-documented framework providing enough insight into its implementation details and issues. The wpa_supplicant and wireless extension has been studied in this thesis work to understand the protocols and design of the state machine and classify Wi-Fi related issues.

There were some cases in which intermittent authentication timeouts were seen before the connection succeeded. If the AP takes longer time to respond and a timeout occurs, the network interface card (NIC) tries to re-authenticate. This behavior was not seen with all the access points. The network interface allows user to set association timeouts. However, authentication timeouts are not configurable. The timeout value in the authentication state machine is set to 70 sec for 802.1x and 30 sec for passphrase. This should be sufficient for a normal authentication. Hence, longer authentication duration must be due to some issue on the AP side.

Some of the error messages are more generic and cannot be pointed to any specific issue. For example: Association reject will be returned if the association timeout value on the device is very low or if there are any MAC filters on the AP side. If the AP does not return specific error to the device, it is difficult to categorize such issues. Multiple suggestions are given based on the understanding and analysis of the underlying problems.

The reason code mapping has also been found to be useful in analyzing problems. Reason code mapping specifically helps in identifying capability mismatch between AP and the client device. Since it is not mandatory for the access points to implement reason codes, all access points may not send the reason codes for all the failures. Information from the AP in the form of reason codes will be very useful on the device side to identify errors. Hence, the access point firmware should implement reason codes.

It is also observed that the android system exposes only a few of the authentication schemes to the user, though the Wi-Fi stack supports several other schemes like CISCO proprietary lightweight extensible authentication protocol (LEAP) and EAP-FAST with protected credential access. Some of these schemes are exposed by the API framework and others are not. If these security features can be supported completely on android, more robust security schemes can be used in high security enterprise networks.

CHAPTER 3

BEST CHANNEL and AP SELECTION METHOD

3.1 Introduction

As explained earlier, one of the important parameters for better Wi-Fi performance is a good channel. Wi-Fi utilizes two frequency bands, 2.4Gigahertz and 5 Gigahertz with 13 and 24 channels in each band respectively. The actual number of channels available for operation among these is country specific. These days Wi-Fi networks have become very common and hence we will find numerous APs in offices or residential areas. APs experience both co-channel and neighboring channel interference due to other transmitting APs. This type of interference can severely affect AP performance and throughput and hence it is necessary to configure the AP on a good channel so that the effect of interference is small.

Another scenario in which AP selection becomes important is an an office or enterprise environment where multiple APs are configured with same SSID to cover the whole area. On the currently available Wi-Fi capable devices, the network interface card (NIC) chooses the AP with a higher RSSI for association. It is assumed that the AP with maximum RSSI is the best AP. This behavior inappropriate for the following reasons:

Firstly, the STA NIC does not consider the quality of the channel while connecting. Usually APs with common SSIDs are spread across different channels during deployment to optimize performance and reduce interference. Hence connecting to an AP operating on a different channel may be better than connecting on a channel with high interference.

Secondly, if all the STA choose the same access point, the channel loading and interference will increase. This might result in connection loss. APs initiate load balancing mechanisms to balance the load. Due to load balancing, the connected clients will have to switch to other APs on the same channel or to a different channel.

Selecting the best AP during the connection will balance the traffic load and performance at an earlier stage and hence AP load balancing would not be necessary. Load balancing after the connection has issues like wrong handovers, ping-pong handovers. The client devices may lose connection if there is a delay in the handover. However, the major issue with selecting a better AP during association is to determine relative rating of each AP. The information available with the client device NIC is very limited, to accurately choose APs.

The goal of this algorithm is to perform channel or AP selection from the client device. If information required by the algorithm should be available at the client device to predict channel and AP performance, the algorithm can be accommodated into the software solution explained previously. In this direction, a study regarding the beacon shift phenomenon has been done. An analytical model for measuring channel occupancy is used and a technique to determine achievable throughput has been proposed.

The algorithm is closely related to the work in [18] in which a dense AP deployment strategy has been proposed. They use an approach called ‘Probe gap analysis’ to determine channel free time. In this method, a series of UDP blasts or probes are sent from AP to the client device to determine the delay in channel access time [18, Section 4.1.1 – 4.1.3]. The channel access delay is used to calculate the channel free time which will determine the achieved throughput given the data rate for the AP. A bucketized rate map approach is used by Murty et al [18, Section 6.1] to calculate the achievable throughput. In the current work, a novel concept of beacon shift technique has been studied and a proposal that the beacon shift mechanism can used for channel occupancy analysis has been put forth. The analytical model accommodates beacon shift, number of APs transmitting and the duty cycle to derive the channel occupancy probability. The rate selection method specified in the IEEE 802.11 standards for each technology is considered to further calculate the achievable throughput for the connection.

During AP selection, this algorithm can be run by the client device to rank the surrounding access points and choose a better candidate. It can also run the algorithm periodically to determine if the currently associated access point is performing well or the device has a better choice among the other access points. The proposed solution can also help users choose better channel in the frequency band used by the associated AP to get better throughput.

3.2 Beacon shift

Beacons are the management frames broadcasted at regular intervals. They carry information about the SSID, signal strength, capabilities like data rates, modulation schemes supported, power save period (DTIM bit) and other AP information. In most of the routers, Beacon intervals can be varied and is usually set to 100 milliseconds. Though beacon interval is constant and AP cannot transmit any packet until the medium is clear i.e. The AP has to follow the CSMA/CA procedure. If the air medium is already occupied with more packets, the time at which beacons are broadcasted shifts. This shift can be calculated by taking the actual timestamp on the beacon and comparing it with the beacon interval. The time slot between two consecutive beacons of an AP will be the sum of the time slot for AP's data (downlink or uplink), back off time (when the AP senses channel to be busy) and the time slot for which the channel is idle.

Time between two consecutive beacons = $\mathrm{T_{data}} + \mathrm{T_{back\ off}} + \mathrm{T_{idle}}$

The main factors determining the actual interval between two consecutive beacons are the fraction of time spent on data transmission at the time instant at which beacon is to be ideally transmitted and the back-off period.

Some experiments were conducted to study the beacon shift phenomenon and the factors affecting beacon shift. Figure 4.1 and Figure 4.2 show the beacon shift when the AP has no incoming data and when there is data on the uplink. Beacon shift observed in the case of 'no data' can be attributed to interference due to transmissions from other APs. But the beacon shift observed in the case uplink data indicates the effect of more loads on the channel. However, it is difficult to characterize beacon shifts completely based on shifts seen for different traffic loads.

Beacon shift on the APs transmitting on the same channel will be similar if the APs are close by. Load or interference on the channel affects all the APs. If one AP is transmitting, the other APs will back off from transmitting. This was verified using an experimental setup consisting of 3 APs configured on the same channel on the frequency band with data traffic on one of the APs. The interference due to other factors remains the same on each of the APs. It was observed that the beacons from each of these APs are shifted to the same extent. If any AP is switched to a different non-overlapping channel, then the observed beacon shift will not be the same. However, if the distance between the APs is considerable, then the beacon shift indicates the channel characteristics of the environment surrounding each AP.

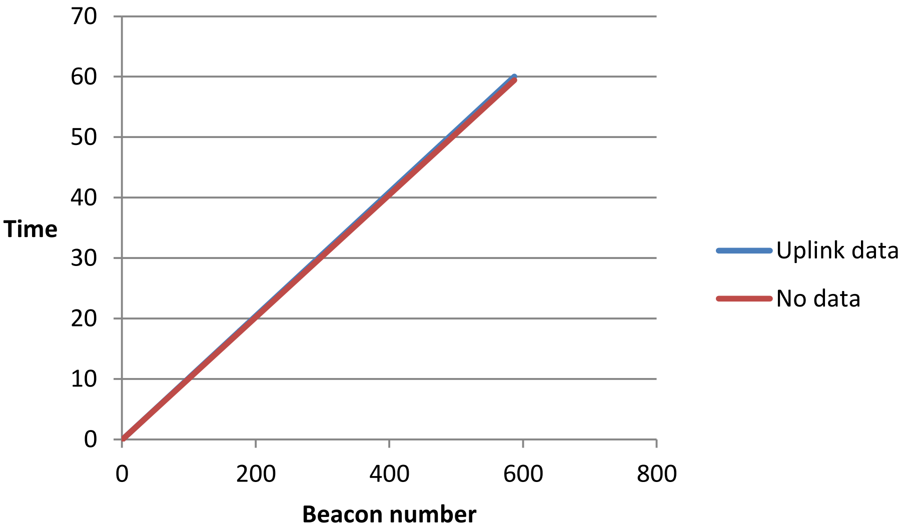


Figure 3.1 - Beacons versus absolute time when the beacon is received (in seconds) for two scenarios – No data load and large file upload on the AP

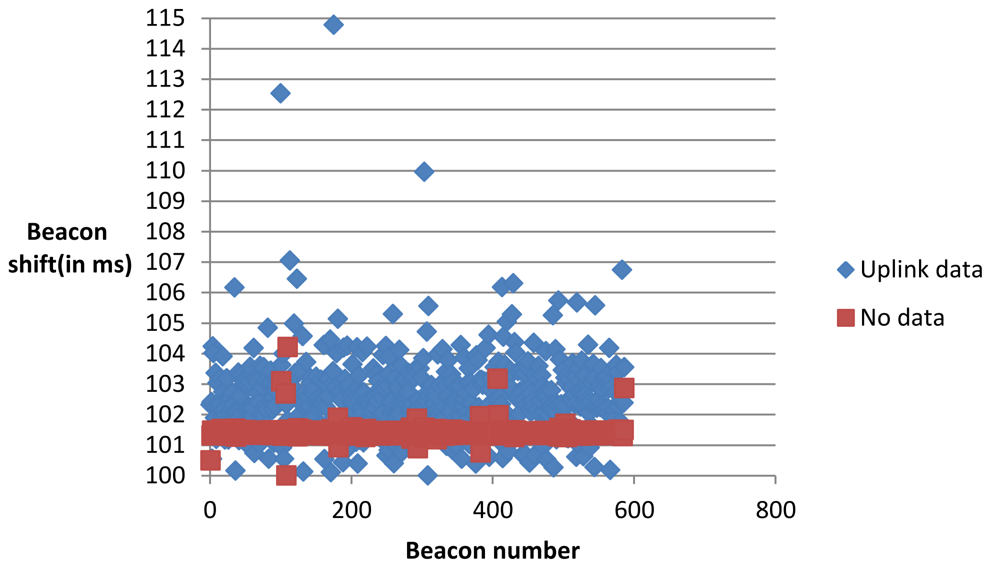


Figure 3.2 - Beacon number versus beacon interval shift in milliseconds no data load and large file upload on the AP

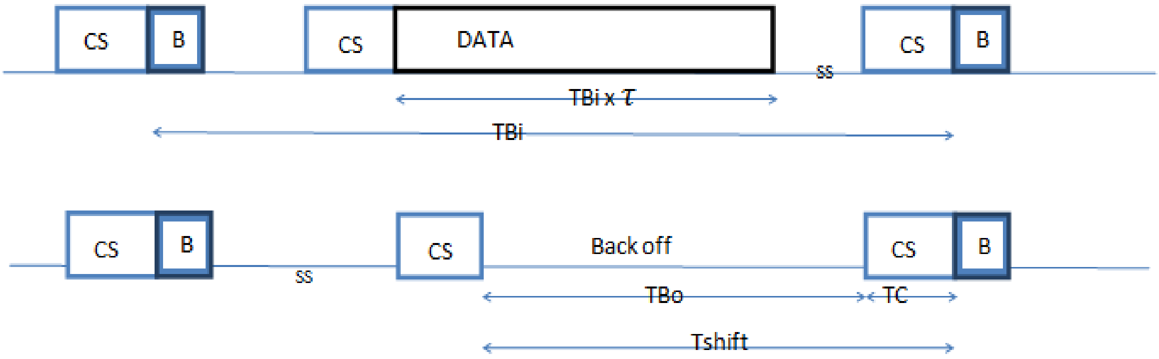
The experimental results have shown that the average beacon shift can be as high as 3-4 milliseconds with instantaneous shifts of 10-13 milliseconds for a beacon interval of 100 milliseconds. Instantaneous shifts can be considered as jitters due to instantaneous traffic. Since load pattern depends on several factors like delay in the AP, TCP flow control and slow internet traffic, jitters are seen in the beacon shift especially during downloads.

An analytical model is required in order to completely model the beacon shift phenomenon and quantify the available information. We use the same analytical model used in beacon interval shifting scheme proposed in [19]. Events such as data transmissions and back-off occurring during the beacon interval can be characterized as a Poisson process. The number of events in any interval, t, is the Poisson distribution with mean λt. Hence, the probability of k event occurrences in the interval t is given by,

$$P(k,t) = \frac{(\lambda\,.\,t)^{k}}{k!}\,.\,e^{-\lambda.t} - - - - - - - - -(2)$$

Where, λ is the average number of event occurrences during the given interval.

The period between two beacons is a sum of idle and active period. When an AP has to transmit a beacon, the delay in transmission is the amount of time the channel is occupied. This duration, for which the channel remains busy between two consecutive beacons, Tbusy is shown.



$$T_{Busy} = T_{BI}.\tau + T_{beaconshift}$$

Figure 3.3 - Possible channel busy period

The probability that the channel is idle in this period, Pidle, is given by,

$$P_{Idle} = P(0, T_{Busy}) = e^{-\lambda T_{Busy}} - - - - - (3)$$

Since there are no events occurring in this interval, k =0.Hence, the probability of a channel being busy in the active period is,

$$P_{Busy} = 1 - P(0, T_{Busy}) = 1 - e^{-\lambda T_{Busy}} - - - - - (4)$$

Since active periods cannot overlap due to carrier sensing and back-off, [19, eq (2)] holds good. The average rate of occurrence of an active period in a beacon interval can be written as:

$$\lambda = \frac{1}{T_{BI}} + \frac{1}{T_{BI} - T_{BI}.\tau} + \cdots + \frac{1}{T_{BI} - (n-1).T_{BI}.\tau} = \sum_{i=1}^{n} \frac{1}{T_{BI} - (i-1).T_{BI}.\tau} \quad ------(5)$$

The parameter τ is defined as the duty cycle or the channel holding time of each AP on the channel. Hence $T_{BI}.\tau$ is the actual data transmission period available for an AP in the beacon interval.

The channel busy probability increases as the back-off period increases as can be seen from Fig 3.4. $T_{Beaconshift}$ is given by the difference in ideal and observed beacon interval

$$T_{Busy} = T_{BI}.\tau + T_{beaconshift} ------(6)$$

And

$$T_{Beaconshift} = T_{ObservedBI} - T_{idealBI} ----- (7)$$

Beacon shift period consists of a fraction of data transmission time if the AP is sending or receiving data and the rest is carrier sensing and back-off period. AP does an exponential random back-off till it senses the medium to be clear to send the beacon. T_{Busy} increases as the $T_{Beaconshift}$ increases. Hence, we are trying to determine the probability of the channel being busy to the observed busy period. Following is the plot of probability of the channel being busy versus beacon shift period.

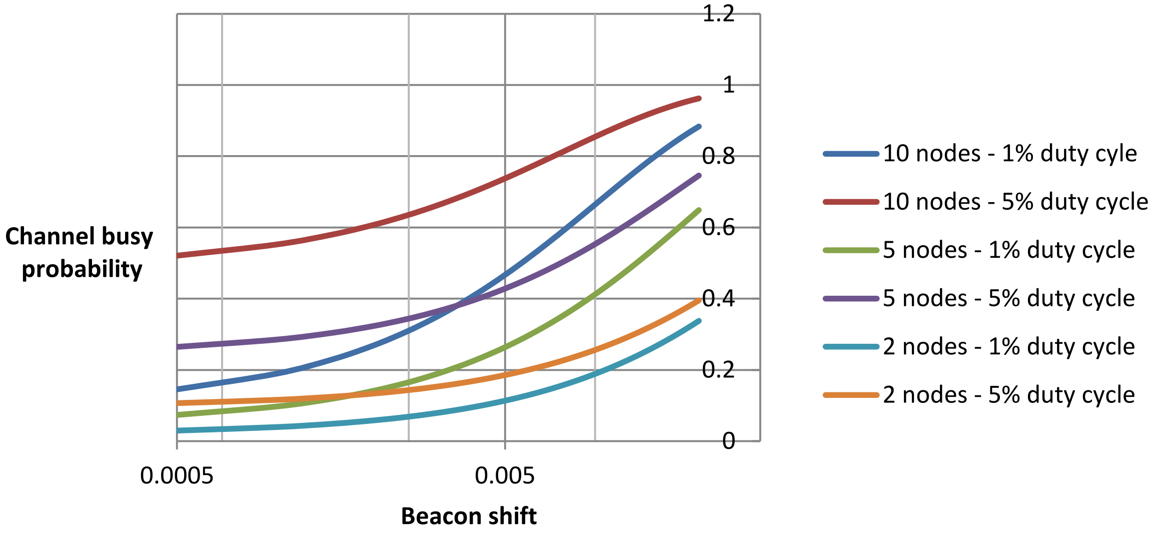


Figure 3.4 - Channel busy probability versus beacon shift for 2, 5, and 10 nodes

The probability $\mathrm{P}_{\mathrm{Busy}}$ is a measure of channel occupancy for the beacon interval. The average channel occupancy over a period of time can be calculated using the above technique which will determine the free air time for the channel. The free air time can be used to calculate the maximum achievable throughput.

3.3 Throughput measurement

The theoretical data rate or throughput for an access point and device communication is a measure of network performance under ideal conditions in which it is assumed that the link between the AP and the device does not have any delay or interference. It is also assumed that the channel availability is 100% for the communication. However, the obtained throughput for a connected client reduces as the channel availability reduces. From the experimental results in [18] it is clear that the maximum obtained throughput is a product of channel free time and theoretical throughput an access point can deliver.

$$\begin{aligned} &Maximum\ achievable\ throughput \\ &\qquad = channel\ free\ probability \times Maximum\ theoretical\ throughput - - - (8) \end{aligned}$$

The data rate for a connection or link between AP and an STA depends on the physical layer technology used and other parameters like RSSI, signal to noise ratio, number of spatial streams used if the AP and the client device support Multiple input multiple output antenna technology (MIMO). IEEE 802.11 has a set of standards which define certain physical layer characteristics and protocols supporting different data rates and performance enhancements. Main extensions to the basic 802.11 technology are 802.11a, 802.11b, 802.11g and 802.11n. 802.11n achieves higher data rates and better performance than the previous standards and is currently the most widely used Wi-Fi standard. Table 3.1 lists some of the major differences.

It is possible to calculate theoretical data rate the AP can deliver based on the standard the AP supports and the parameters that are considered for choosing the data rate and throughput. Since 802.11n is currently the most widely deployed Wi-Fi technology, the experiments in this work were performed considering 802.11n standard. The other standards employ similar techniques for data rate selection. The physical data rate is chosen based on the physical layer modulation schemes, the coding rate used and other parameters pertaining to the specific 802.11 technology.

Table 3.1 - Major differences between 802.11a/b/g/n standards

	802.11a	802.11b	802.11g	802.11n
Frequency	5GHz	2.4GHz	2.4GHz	2.4GHz/5GHz
Spread spectrum technology	OFDM	DSSS	OFDM	OFDM
Modulation	BPSK, QPSK, 16QAM, 64QAM	DBPSK, DQPSK	DBPSK, DQPSK	BPSK, QPSK, 16QAM, 64QAM
Data rates	Upto 54Mbps	1, 2, 5.5 and 11Mbps	Upto 54Mbps	Upto 300 Mbps
Other features	Low interference from Microwave, cordless phone. Current deployment density is low			MIMO, variable modulation and coding schemes, channel bonding

In order to verify the explained analytical model and estimate of achievable throughput, some experiments were performed using 802.11n capable access points and devices. The theoretical data rate supported by the AP as specified in the IEEE 802.11n standard takes into account the following factors; The signal to noise ratio to determine the modulation and coding scheme to be used by the AP and the device, number of spatial streams used for is multiple antenna technology (MIMO) is used, the guard interval used for orthogonal frequency division multiplexing (OFDM). Orthogonal frequency-division multiplexing (OFDM) is a method of digital modulation in which a signal is split into several narrowband channels. Guard intervals are used in OFDM to avoid overlapping of the consecutive bits/symbols. The IEEE standard specifies the mapping between these parameters. The modulation and coding scheme index table (Table 3.2) indicates the mapping between different 802.11n parameters and the data rate. The standard also specifies the minimum signal to noise requirement for using any particular modulation scheme to be used (Table 3.3). This is based on the fact that lower the SNR, the link will not be able to support higher data rate and maintain less bit or packet error rate. The SNR requirement has been specified for a bit error rate of 1%. In a multiple access point environment, the transmissions from other APs can be considered as noise or interference for the AP associated with the client device. Hence, signal to inference and noise ratio (SINR) instead of signal to noise ratio (SNR) will be considered for data rate selection. SINR or signal to interference ratio for a link (i, j) is given by,

$$\text{SINR} = {}^{\text{RSSI of the associated AP}}\!/_{\eta + \sum \text{RSSI of other APs}} ------(9)$$

The SINR for the link between the access point and the client can be measured using the above method, considering the Wi-Fi spectral mask into account for co-channel and inter-channel interference power. The spectral mask indicates the influence of transmit power of an AP transmitting with center frequency fc on other neighboring channels.

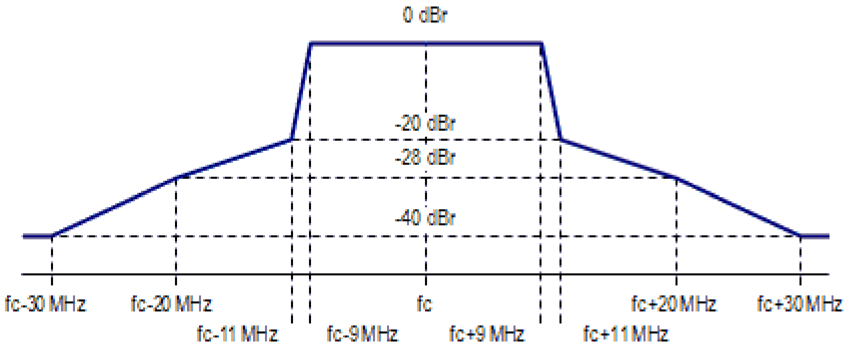


Figure 3.5 - Wi-Fi spectral mask

Considerable amount of timing overhead is incurred for each frame transmission in a wireless network. Since a wireless transmission is more susceptible to collision, packet corruption and security risk, the 802.11 standard specifies several measures to avoid data loss and improve robustness. The standard specifies fixed timing parameters like short inter-frame space (SIFS), slot timing and distributed inter-frame space (DIFS) for each frame transmission. The transmitter has to sense the medium for a period of DIFS to determine if the channel is busy, before deciding to transmit. After transmission, it has to wait for a period of SIFS to receive acknowledgement from the receiver. Other overhead factors include Wi-Fi headers, synchronization bits or preambles and upper layer protocol overheads. This overhead reduces the throughput even further given the physical data rate.

To accurately measure the obtained data rate or the throughput, it is necessary to consider this overhead factor to get accurate prediction. Table 3.4 lists the parameters to measure the throughput from the data rate for 802.11n. These parameters vary for each 802.11 technology and are specified by the standard. The maximum throughput is given by,

$$Datarate \times \frac{T_{data}}{(T_{data} + T_{overhead)}} ------(10)$$

$$T_{overhead} \;=\; 3 \times T_{SIFS} + T_{DIFS} + \left(\frac{CW}{2}\right) \times T_{slot} + T_{PHYheader} + \; T_{MACheader} + T_{\frac{TCP}{IP}} \;--- (11)$$

where, T_{SIFS}= Short inter-frame space

T_{DIFS} = Distributed inter-frame space

CW = contention window size

T_{slot} = Slot width

$T_{PHYheader}$ = Overhead due to Wi-Fi PHY header

$T_{MACheader}$ = Overhead due to Wi-Fi MAC header

$T_{TCP/IP}$ = Overhead due to TCP, UDP and IP headers

Table 3.2 – Modulation and coding index for 802.11n

MCS Index	Number of spatial streams	Modulation scheme	Coding rate	Data Rate (in Mbps)		Data Rate (in Mbps)	
				(GI = 800ns)		(GI = 400ns)	
				20MHz	40MHz	20MHz	40MHz
0	1	BPSK	½	6.5	13.5	7.2	15
1	1	QPSK	½	13	27	14.4	30
2	1	QPSK	¾	19.5	40.5	21.7	45
3	1	16-QAM	½	26	54	28.9	60
4	1	16-QAM	¾	39	81	43.3	90
5	1	64-QAM	2/3	52	108	57.8	120
6	1	64-QAM	¾	58.5	121.5	65	135
7	1	64-QAM	5/6	65	135	72.2	150
8	2	BPSK	½	13	27	14.4	30
9	2	QPSK	½	26	54	28.9	60
10	2	QPSK	¾	39	81	43.3	90
11	2	16-QAM	½	52	108	57.8	120
12	2	16-QAM	¾	78	162	86.7	180
13	2	64-QAM	2/3	104	216	115.6	240
14	2	64-QAM	¾	117	243	130.3	270
15	2	64-QAM	5/6	130	270	144.4	300

Table 3.3 - Minimum SNR requirement for 802.11n and required bit error rate of 1%

SNR	MCS
2	BPSK1/2
6	QPSK1/2
5	BPSK3/4
10	QPSK3/4
12	16QAM1/2
15	16QAM3/4
18	64QAM2/3
20	64QAM3/4
24	64QAM5/6

Table 3.4 - Parameters for calculating protocol overhead

T_{SIFS}	16μs
T_{slot}	9μs
T_{DIFS}	34μs
CW	15
PHY header	48 bits
PHY preamble	72 bits
MAC header	224 bits
TCP header	160 bits
UDP	64 bits
IP	160 bits
Acknowledgement	112 bits
Frame body	2312 bytes
WPA overhead	54 bytes

Table 3.5: Calculated throughput for physical data rates according to equation 10

Physical data rate for GI =400ns	Theoretical throughput
7.2	7.111602656
14.4	14.05897007
21.7	20.94424383
28.9	27.5786705
43.3	40.40817065
57.8	52.77480241
65	58.71477546
72.2	64.5291712

3.4 Experiments

Two sets of experiments were conducted to study the accuracy of the throughput prediction algorithm. These experiments were conducted in a residential area, where, channels 1, 6 and 11 are usually crowded with large number of access points.

3.4.1 Throughput versus SINR

The experimental setup consists of two 802.11n capable laptops with Windows operating system and Intel wireless N-1030 card. The laptops are associated with Linksys E1000 access point. The access point interface allows channel, transmit power and guard interval configuration. The experiments are conducted for 2.4GHz frequency band. The guard interval on the AP is set to 400 nanoseconds and the client laptops support only one spatial stream. The laptops are placed approximately 3m from the access point. Some software tools are required to scan access points, measure SINR, capture beacons and measure throughput. Laptop 1 is used to measure SINR. Laptop 2 is used to capture beacons while measuring SINR. Both the laptops are required for measuring actual throughput.

In order to measure the throughput for different values of SINR, the transmit power of the access point is varied with a fixed transmit channel. The corresponding SINR is measured every time. We measure the SINR using a Wi-Fi tool called ‘inSSIDer’. The software calculates the SINR based on the RSSI values of the access points returned by the Wi-Fi card on the laptop. Beacons are captured on Laptop 2 during the SINR measurement for a period of 25 seconds. Packet capturing tool called ‘Network Monitor’ is used for this purpose. The tool captures all the Wi-Fi packets seen by the interface and can be filtered to get beacon frames and record the timestamp. Averaging of channel busy period for 25 seconds is long enough to determine channel condition accurately. The number of APs on the channel was also recorded from the scan results of the inSSIDder tool.

A throughput measurement tool called QCheck is used to measure the actual throughput for the connection. In order to measure throughput accurately, Laptop 2 is connected via Gigabit Ethernet to the router (acts as a server) and Laptop 1 acts as the wireless client. Laptop 2 is on Ethernet to make sure that the server end has no throughput bottleneck. Hence the measured throughput will be accurate for the Wi-Fi connection. Both the laptops should run QCheck to establish a client-server TCP connection. Twenty continuous throughput readings are taken to measure the average throughput. The above procedure is repeated for different values of transmit power and SINR to plot the throughput variation with respect to SINR. The calculations are made as follows.

The data rate for the connection can be mapped from Table 4.2 and 4.3 according to the SINR, guard interval and spatial streams used. $T_{overhead}$ and T_{data} can be calculated from Table 4.4 for TCP(Maximum transmission unit of 1500 bytes) since the throughput measurement tool uses TCP connection to measure throughput. Minimum contention period is assumed between each data frame. The above calculation gives an approximation of the packet overhead as shown in Table 4.5. Substituting the values of data rate and overhead in equation 10, the maximum throughput can be calculated. The timestamp from each beacon is obtained using which the beacon shift from the ideal value of 100 milliseconds is calculated. Substituting the values of beacon shift, number of APs recorded and assumed duty cycle of 5% in equation

4, P_{busy} and averaging it over the total time interval, channel free probability is calculated for each SINR reading. The beacon shift is determined for each measurement of SINR for accuracy. The beacon shift variation for the channel was found to be minimal for each SINR reading with an average channel busy probability of 45.58%. The calculated and measured throughput versus SINR is as shown in Figure 4.5.

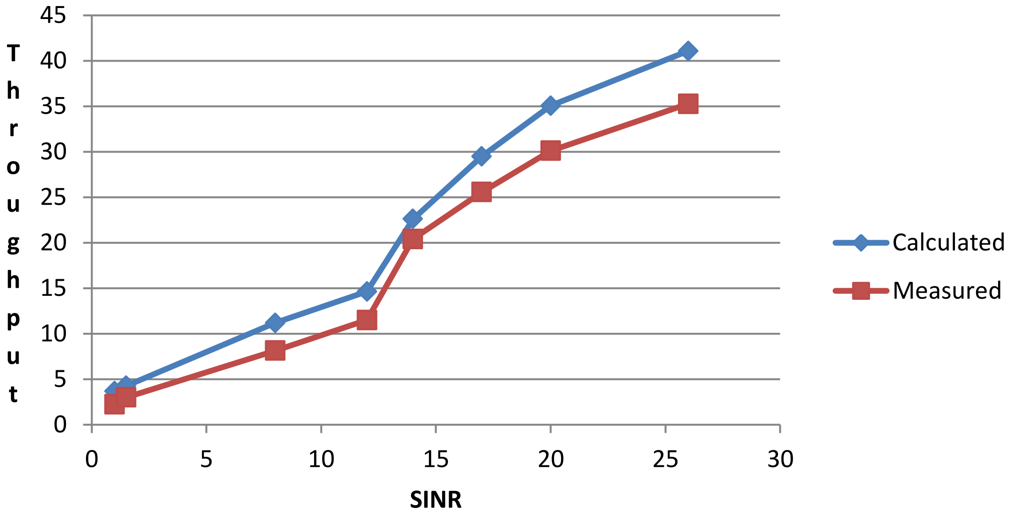


Figure 3.6 - Comparison of calculated and measured throughput versus SINR

In another experiment for throughput versus SINR, a third laptop is used to associate with the AP and transfer data. In this experiment, Laptop 1 is used to measure SINR and Laptop 2 will simultaneously capture beacons. The throughput is measured using both laptops with one connected to Ethernet and another on Wi-Fi as explained previously. This is repeated with laptop 3 transferring large amount of data. This is to test the effect of data load on the measured and calculated throughput.

Though small amount of data do not affect the throughput for Laptop 1 which is measuring throughput using QCheck, an operation like large file transfer will generate sufficient load on the AP and the channel to a good extent. Laptop 3 is used to generate this load on the AP and the beacon shift is measured simultaneously. The throughput measured by laptop 1 reduces. Beacon shift increases with the load. In Figure 3.7, calculated and measured throughputs for without load and with load scenarios are plotted.

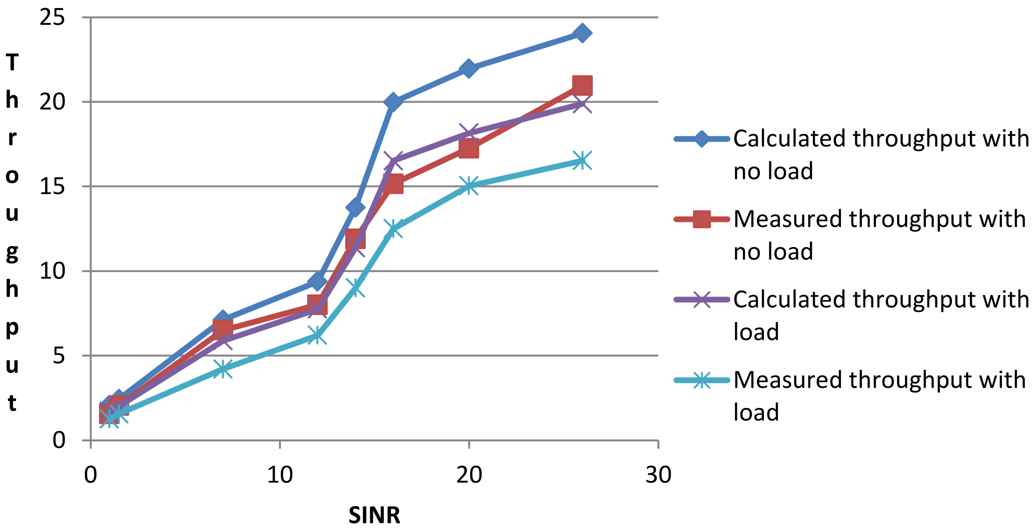


Figure 3.7 - Comparison of calculated and measured throughput versus SINR with and without data load on AP.

3.4.2 Throughput versus channel

This experiment is intended to verify that the channel assessment is possible using the proposed algorithm. The AP is configured to transmit on different channels. To calculate the achievable throughput, the measurements required are: SINR for the channel, number of access points transmitting on the channel, beacon shift measurement for each channel. The experimental setup explained above can be used for this experiment as well.

The experimental setup consists of two 802.11n capable laptops with Windows operating system and Intel wireless N-1030 card connected to Linksys E1000 access point. The guard interval of 400 nanoseconds and short preamble is used. The client laptops support only one spatial stream. The access point is configured to transmit on channel 1 and transmission power is set to maximum.

The SINR is measured on laptop 1 and beacons are captured simultaneously on the other laptop. The number of transmitting APs is recorded. The number of APs on each channel will be the total number of APs contributing to co-channel and inter-channel interference. The interference from the inter-channel access points on the AP vary according to the spectral leakage in Wi-Fi. The access points whose relative RSSI on any channel is below a certain level will not contribute significantly to the SINR. The RSSI threshold considered in our calculation is -90dBm. The number of APs influencing a channel can thus be calculated by scanning for APs on the same and neighboring channels and then filtering. For example: If there are 10 APs on channel 1, based on the RSSI of each AP and reduction in the transmit power, number of APs influencing channel 2, 3 and 4 vary. The duty cycle of 5% is assumed for the high traffic density environment. The throughput measurement is made as explained in the earlier section. The channel busy probability varies for each channel according to the measured beacon shift and number of APs influencing each channel. The maximum achievable throughput is calculated from equations 4, 10 and 11. The experiment is repeated for all the 11 channels. The following figure shows the comparison of measured throughput and throughput calculated from the above procedure.

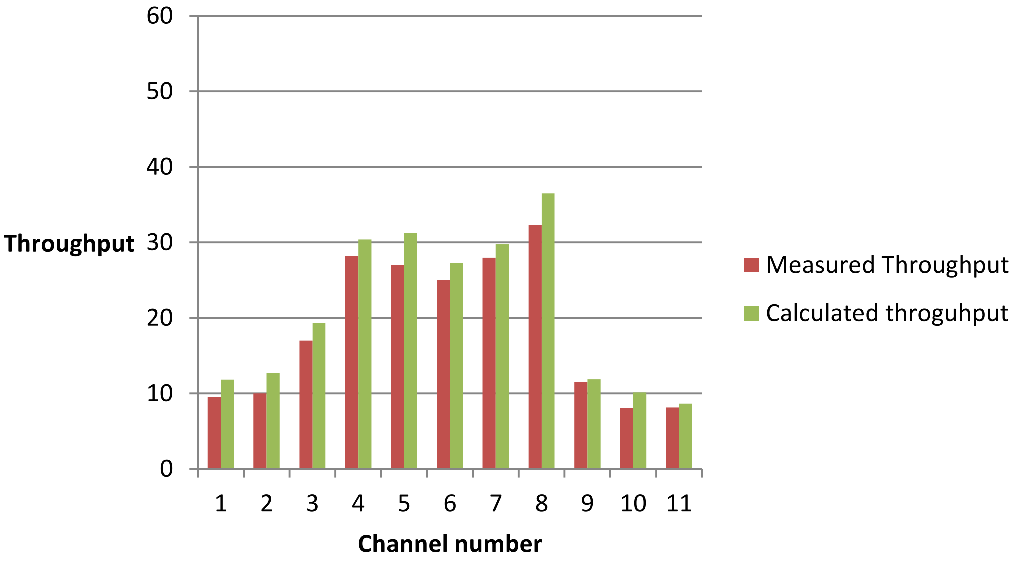


Figure 3.8 - Comparison of calculated and measured throughput for channels 1 to 11 in 2.4GHz range

3.5 Results

From the above experiments, it can be seen that algorithm to find a better channel performs well in calculating the achievable throughput. The analytical model used to characterize beacon shift takes into account number of access points on influencing the channel and the duty cycle. The results of comparison between the predicted achievable throughput and measured throughput show that the probability of channel occupancy can be used to measure the channel free period quite accurately. The channel occupancy model predicts the percentage of throughput reduction. The throughput measurement algorithm can calculate the achievable throughput and through the above experiments, it is evident that the results match the observed throughput for the AP on each channel.

The data rate calculation is based on the physical layer characteristics of the 802.11 technology used by the access point and the device. The above experiments were conducted on 802.11n access point and laptops. Figure 3.6 and 3.7 show the plot for calculated and measured throughput versus signal to interference ratio measured. The access points and the client devices negotiate the data rates for the communication based on the signal strength seen by the client devices and the noise in the environment. Hence it is evident that the variation in measured throughout with varying SINR follows the data rate variation.

The SINR versus throughput experiment has been conducted for two scenarios. Figure 3.7 is a plot of throughput versus SINR for with load and without data load conditions on the access point. The data load on the AP will affect the throughput of other clients and hence the predicted and measured values for a loaded AP are less than the values for a non-loaded AP. If the load on the AP is not considerable, then the throughput reduction is minimal. However, if the AP is loaded heavily with a large data transfer operation, the throughput measured by the other client reduces. The beacon shift varies and the analytical model predicts higher channel busy probability due to data load. The plot shows the reduction in the calculated and measured throughput with data load on the AP.

The channel versus throughput experiment was performed to determine throughput for the AP on each channel. From Figure 3.8, it is clear that channel 8 performed better than other channels during the measurement. Though channels 1, 6, and 11 are theoretically non-overlapping channels, due to dense AP deployments on these channels, the intermediate channels may give better performance. It helps to choose a good channel for the AP based on the predicted channel throughput. Users can explore placing the access points on these channels which are usually less crowded.

The channel assessment algorithm can be implemented by the client devices to rank the access points based on the predicted throughput and choose a good candidate. Kim et al [35] have implemented android application for choosing candidate access points based on signal to noise ratio as the criteria for ranking each AP. Similarly, the above algorithm can be integrated into the android software presented in chapter 2. The algorithm is more accurate in terms of predicting throughput for the access points than simply ranking the access points according to the SINR. The application in [35], SINR is calculated through Wi-Fi APIs provided by the android Wi-Fi framework considering the spectral mask.

Though the above algorithm has been tested for Windows system, it can be implemented on android where in the device can independently measure the beacon shift, number of access points, SINR through theoretical calculations using the spectral mask, corresponding data rate and throughput to choose the AP. The Wi-Fi drivers on android system should allow monitor or promiscuous mode to capture packets or beacons using a library called ‘tcpdump’. The currently available default Wi-Fi setting interface which allows user to configure Wi-Fi profile on the device should use the algorithm to filter access points according to the predicted throughput. The channel ranking according to the algorithm can also be provided to the user. This algorithm can be executed periodically to determine if the client can switch to a different connection on either the same channel or a different channel to get better performance. Users can be notified to switch the access point to a different channel if the current setup is not performing well.

CHAPTER 4

CONCLUSION AND FUTURE WORK

As part of this thesis work, an attempt has been made to address complexities involved in debugging Wi-Fi association issues and optimization of networks through a novel approach. Through the diagnostics framework, it is demonstrated that system log analysis is a useful method for identifying and debugging issues and on android system, which has a well developed logging framework, the Wi-Fi diagnostics will help common users identify and correct association issues with the Wi-Fi network. Currently, the tool is designed to monitor only the association process. It can be extended to monitor the connection during the active period for reporting any intermittent issues or connection losses. Using the classification and analysis carried out in this work, the diagnostics tools can be designed for other operating systems, if the Wi-Fi stack uses the same supplicant. The same WPA supplicant is used in other Linux operating systems.

Tuning the applicable Wi-Fi parameters is important to maintain network performance and optimize. The possibility of developing automated software to tune parameters on the client device can be explored. Self-Configuration of parameters like fragmentation threshold, sensitivity and power management scheme based on the performance of the 802.11 MAC and PHY in a particular environment will help improve Wi-Fi performance without user intervention.

A technique to identify a better channel through an algorithm for measuring achievable throughput from an AP transmitting on a given channel has been proposed and experimented in this work. The drift in the beacon timing when an AP transmits beacons in an environment with interference has been studied. Experiments were conducted to study this behavior with different loads on the AP. A mathematical model is used to show that beacon shifts can be used to calculate the probability of a channel being busy. The algorithm proposed to find a better channel or better access point can predict the achievable throughput. Through the experiments, it is shown that the predictions are accurate. The algorithm can be integrated with the diagnostics software if Wi-Fi drivers allow monitor mode on the devices to capture beacons and measure other parameters. The performance of the algorithm can be compared with other algorithms explained earlier. Improving the accuracy of the prediction can be further explored to accommodate factors like number of devices already associated with the access point and predicting data load patterns on the AP. Integration of this idea into other load balancing mechanisms and AP selection methods can also be explored.

REFERENCES

[1] Ramakrishna Gummadi, David Wetherall, Ben Greenstein, Srinivasan Seshan, “Understanding and Mitigating the Impact of RF Interference on 802.11 Networks” in Proceedings of SIGCOMM conference on Applications, technologies, architectures, and protocols for computer communications, 2007.

[2] Manoj Tolani, Rajan Mishra, “PERFORMANCE ANALYSIS OF WLAN BY COMPARATIVE STUDY OF VARIOUS ATTRIBUTES USING OPNET SIMULATOR” , International Journal of Scientific & Engineering Research, Volume 3, Issue 6, June-2012.

[3] IEEE 802.11n-2009—Amendment 5: Enhancements for Higher Throughput. IEEE-SA. 29 October 2009. doi:10.1109/IEEESTD.2009.5307322

[4] Bjoern Dusza, Christoph Ide and Christian Wietfeld, “Interference Aware Throughput measurements for Mobile WiMAX over Vehicular Radio Channels”, in the *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC) Workshop*, 2012 © IEEE

[5] Hetal Jasani, Yu Cai, “Performance Evaluation of Wireless Networks”, in the *Proceedings of IEEE Southeastcon*, Huntsville, Alabama, 2008.

[6] Android development guide [Online]. Available: http://developer.android.com/guide/components/index.html

[7] Wi-Fi CERTIFIED™ for WMM™ - Support for Multimedia Applications with Quality of Service in Wi-Fi® Networks, © 2004 Wi-Fi Alliance.

[8] Ioannis Papanikos and Michael Logothetis, “A Study on Dynamic Load Balance for IEEE 802.11b Wireless LAN”, in the *Proceedings of International conference on advances in communication and control*, 2011.

[9] Michael Buettner, Gary V. Yee, Eric Anderson, Richard Han, “X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks”, in the *Proceedings of the International conference on Embedded networked sensor systems*, 2006 ©Association for Computing Machinery

[10] Li-Hsing Yen, Tse-Tsung Yeh, and Kuang-Hui Chi, “Load Balancing in IEEE 802.11 Networks”, in the *Proceedings of the IEEE conference on internet computing*, 2009 © IEEE

[11] Steven Melvin, “Endpoint Identification Using System Logs”, in the *Proceedings of the 22nd ACM Symposium on Operating System Principles (SOSP)*, Big Sky, Montana, October 2009.

[12] Wanchun Li and Ian Gorton, “Analyzing Web Logs to Detect User-Visible Failures”, in the *Proceedings of the workshop on Managing systems via log analysis and machine learning techniques*, 2010

[13] Onn Haran and Bnei Dror, “RELATIVE VEHICULAR POSITIONING USING VEHICULAR COMMUNICATIONS” U.S. Patent 2011/0112766 A1, May 12 2011.

[14] Shun-Yong Huang, “CONCURRENT CONTROL METHOD FOR A COMMUNICATION DEVICE EMBEDDED WITH WI-FI DIRECT”, U.S Patent 2013/0044739, Feb 21 2013.

[15] Radio resource management under unified wireless networks – [Online] http://www.cisco.com/en/US/tech/tk722/tk809/technologies_tech_note09186a008072c759.shtml

[16] Bob O’Hara and Al Petrick, “IEEE 802.11 Handbook – A Designer’s Companion”, IEEE Press, 1999.

[17] Andrew D. Ferguson and Rodrigo Fonseca, “Inferring Router Statistics with IP Timestamps”, in the *Proceedings of ACM Conference on emerging Networking Experiments and Technologies Student Workshop*, 2010 © ACM

[18] Rohan Murty, Jitendra Padhye, Ranveer Chandra, Alec Wolman, Brian Zill, “Designing High Performance Enterprise Wi-Fi Networks”, in the *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, 2008.

[19] Seungku Kim, Seokhwan Kim, Jin-Woo Kim and Doo-Seop Eom, “A Beacon Interval Shifting Scheme for Interference Mitigation in Body Area Networks”, in the *Proceedings of Sensors*, 2012

[20] Wireless meshing and 802.11n, white paper, 2008 © Ruckus wireless

[21] Markus Tauber, Saleem N. Bhatti, “Low RSSI in WLANs: Impact on Application-Level Performance”, Unpublished

[22] Yoon Hyun Kim and Jin Young Kim, “Performance Comparison of 802.11n System with WLAN Channel Model”, in the *Proceedings of the International symposium on Communications and Information Technology*, 2009 © IEEE.

[23] Qiang Ni, Tianji, Theirry,Yang, Adrian, Changwen, “Design and Analysis of 802.11n MAC protocol” , IEEE workshop, June 2008 © IEEE

[24] Srikanth S, Selvam T, “Performance Study of IEEE 802.11n WLANs”, Communication

Systems and Networks workshop, 2009.

[25] Mehmet Yavuz and David W. Paranchych , “Adaptive Rate Control in High Data Rate

Wireless Networks”, Wireless communication and networking, 2003, © IEEE.

[26] Zaharoula Harpantidou and Michael Paterakis, “Random Multiple Access of Broadcast Channels With Pareto Distributed Inter-arrival Times”, PhD dissertation, Department of Electronics and Communication, Technical Univeristy of Crete, Greece.

[27] P. Raptis, V. Vitsas, K. Paparrizos, P. Chatzimisios, A. C. Boucouvalas, P.Adamidis, "Packet Delay Modeling of IEEE 802.11 Wireless LANs", Mobile Networks and Applications Journal, Volume 14 Issue 6, December 2009.

[28] Bong Jun Choi and Xuemin (Sherman) Shen, "Adaptive Exponential Beacon Period Protocol for Power Saving in Delay Tolerant Networks", PhD Dissertation, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada.

[29] Su HU, Gang WU, Vue XIAO, Shaoqian LI, "Iterative Channel Estimation for Short Preamble Based OFDM/OQAM System", in the *Proceedings of the conference on Communication circuits and Systems*, 2009 © IEEE

[30] Fatemah R Ezaei, "A Comprehensive Analysis of Physical Layer", PhD Dissertation, Department of Electronics and Comupter Engineering, University of Nebraska, 2010.

[31] Thomas Scherer and Thomas Engel, "Bandwidth Overhead in WiFi Mesh Networks for Providing Fair Internet Access", in the *Proceedings of the ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks*, 2006 © ACM

[32] Jangeun Jun, Pushkin Peddabachagari, Mihail Sichitiu, Theoretical Maximum Throughput of IEEE 802.11 and its Applications, *Proceedings of the Second IEEE International Symposium on Network Computing and Applications*, 2003 © IEEE.

[33] A. Banchs and X. Perez, "Providing throughput guarantees in IEEE 802.11 wireless LAN", in the Proceedings of IEEE Wireless Communications and Networking Conference, 2002 © IEEE.

[34] Jiansong Zhang Kun Tan Jun Zhao Haitao Wu Yongguang Zhang, "A Practical SNR-Guided Rate Adaptation ", in the *Proceedings of IEEE conference on Computer Communications*, 2003 © IEEE.

[35] Hyun Soon Kim, Eugene Kim, Hwangnam Kim, “QoE-driven Wi-Fi Selection Mechanism for Next Generation Smartphones”, in the *Proceedings of IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT)*, 2012 © IEEE

[36] Stephane Lohier1, Yacine Ghamri Doudane2, and Guy Pujolle3, “Cross-Layer Loss Differentiation Algorithms to improve TCP Performance in WLANs”, in the *Proceedings of the international conference on Personal Wireless Communications*, 2006

BIOGRAPHICAL INFORMATION

Suma Subbarao graduated from National Institute of Engineering, affiliated to Vishweshwaraiah Technological University, Belgaum, India in 2009. She worked as a Software engineer at Motorola Solutions till July 2011. She started her graduate studies in Electrical Engineering at The University of Texas at Arlington in fall 2011. Her research involves Wi-Fi technology and optimization strategies for wireless LANs.