POWER MANAGEMENT OF ACCESS POINTS

A Thesis by

Minhaj Uddin

Bachelor of Engineering, Osmania University, India, 2008

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in fulfillment of
the requirements for the degree of
Master of Science

December 2010

POWER MANAGAMENT OF ACCESS POINTS

The following faculty members has examined the final copy of this thesis for form and content, and recommends that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Electrical Engineering.

____

Vinod Namboodiri, Committee Chair

____

Brian Driessen, Committee Member

____

Yanwu Ding, Committee Member

DEDICATION

To my parents and my elder brother, for their prayers and wishes,
which gave me strength and support

ACKNOWLEDGEMENTS

I would like to thank many people who showed their interest in and supported my work. First, I thank my advisor, Dr. Vinod Namboodiri, for his concern, support, help, and advice with every part of my thesis. I also thank Nagaraja Thanthry for his support and guidance. I express gratitude to all my friends, especially Khubaib, for their kind help and support throughout the course of my thesis work. Finally, my heartfelt appreciation goes out to my parents and brothers for their encouragement, prayers, and good wishes, which helped me to write this thesis and obtain my Master of Science degree.

ABSTRACT

The evolution of hand-held devices and portable machines started the increase in demand for reliable, secure, and faster means of communication from wireless LAN infrastructures. This has lead to an enormous increase in the size of networks, which in turn has increased the cost and power consumption of access points and switches in the network infrastructure. While managing demand, networks become underutilized with less traffic demands and over utilized with more traffic demands.

This research tends to discuss the best designs for maximum efficiency in terms of power and cost while retaining the best user experience. Designs were mainly evaluated based on the benefits of a centralized structure and a non-centralized structure. Furthermore, in order to understand the differences and make centralized structure more efficient in terms of power and cost, a new algorithm was designed with convex hull approach and was also implemented in the emulator kind of network testbed generating outstanding results and evaluations. The algorithm proposed in this thesis and the resulting design, if implemented in the future, could mean 60 percent additional savings in power consumption and costs.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF ABBREVIATIONS

AP	Access Point
GHZ	Gigahertz
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
MHZ	Megahertz
POE	Power over Ethernet
WLAN	Wireless Local Area Network

CHAPTER 1

INTRODUCTION

1.1 Motivation

The Internet has advanced drastically and changed dynamically in two ways. First, Internet users have increased tremendously, surpassing the billion mark, according to world statistics. Second, to accommodate this tremendous increase in Internet users, a number of devices for providing and establishing Internet connections have also increased. The basic goal for every organization is deploying these wireless networks to establish seamless connectivity while their reducing cost and power consumption.

In order to provide users with an Internet connection, a wireless local area network (WLAN) infrastructure is necessary and consists mainly of the following devices: access points (APs), switches, and routers. Power consumption of these devices is considerable when they are deployed in inconsiderable numbers at large organizations. Minimizing power consumption and cost are crucial components in the deployment of wireless networks. In order to understand these two factors in designing wireless networks, they were calculated for a certain location to show why their minimization is crucial.

1.2 Present Problems

The major area of concern revolves around the power consumption and costs incurred in designing large networks. In this thesis, an example from an organization in which the wireless infrastructure is not centrally controlled and deployed in large numbers was used to explain some practical details. A campus building from Wichita State University that usually provides a seamless WiFi connection throughout the day was chosen for this example. It was assumed that this building consists of 100 access points on average for providing wireless coverage and that there are five switches providing power to these APs. Assuming that all 100 access points are awake and ready for processing packets, the total power consumed by the APs and switches throughout the year was calculated, with the following results: The average power consumption of an access point when measured with a wattmeter was 10 watts. Table 1 shows the calculated power consumption and cost incurred for managing 100 APs in a wireless network.

TABLE 1

POWER CONSUMPTION AND COST TO MANAGE 100 ACCESS POINTS IN A WIRELESS NETWORK

Observations (Access Points)	**Results**
Power consumption of access point per hour	0.01 kWh
Power consumption of access point per day	0.24 kWh
Power consumption of access point per month	7.20 kWh
Power consumption of access point per year	87.60 kWh
Power consumption of 100 access points per year	8,760.00 kWh
Total cost to manage power of 100 access points per year at 15 cents/kilowatt hour (kWh)	$1,314.00

According to the Electricity Board of the United States of America [13], on the average, 15 cents is charged for 1 kWh. The total cost for 100 access points per year is $1,314.00. At least five switches are needed to provide power over the Ethernet (POE) [18] to the 100 APs. On the average, a Cisco switch [17, 18] requires 100 watts of power to be turned on, which is in addition to the above power it provides to the access points. The total cost incurred and power consumption to manage the switches for providing power to the APs is shown in Table 2. Thus, the total cost for maintaining a 100-access-points site for one year (including switches) is $1,971.

TABLE 2

POWER CONSUMPTION AND COST TO MANAGE FIVE SWITCHES IN A WIRELESS NETWORK

Observations (Switches)	**Results**
Power consumption of switch per hour	0.1 kWh
Power consumption of switch per day	2.4 kWh
Power consumption of switch per month	72.0 kWh
Power consumption of switch per year	876.0 kWh
Power consumption of five switches per year	4,380.0 kWh
Total cost to manage power of five switches per year at 15 cents/kWh	$ 657.00

Tables 1 and 2 clearly show that the operational cost for maintaining these wireless networks is very expensive and would become a massive number for a very large network. The centralized structure was studied in detail and a scheme was proposed, which if implemented in wireless networks would decrease the power consumption and cost considerably without sacrificing efficiency.

Another problem related to existing wireless infrastructures is the deployment and management of access points and switches. Each AP requires proper configuration and network provisioning before its deployment in a wireless network. There are also issues with defining and managing DHCP pools, IP addressing, VLANs, and security. Compatible configuration of switches to which these access points would be connected is also required. It is difficult to imagine how much time it would take to configure these devices in large organizations that have hundreds of APs.

To solve the above problems, an algorithm that can be implemented in a centralized structure was proposed, and an attempt was made to emulate the behavior of the algorithm and centralized structure in a simple network testbed. The emulation with proposed algorithm demonstrated some very beneficial results relative to power consumption and costs incurred in the network infrastructures, including drastic improvements.

1.3 Future Communication and Revolution

The future of communication will be completely wireless and mobile in nature. Mobile phones have already outnumbered fixed phones. All laptops are now equipped with wireless cards for connection to wireless networks, and always being connected to the network has become a necessity rather than a luxury nowadays. Just as mobile technologies have their fixed network infrastructure and foundation built on cellular concepts, there has been a strong need for a fixed wireless network infrastructure. Questions arise about the best infrastructure to deploy in wireless networks, best way of controlling network devices, and best way to manage the network without sacrificing seamless Internet connectivity to all users.

The time has come for wireless networks to revolutionize the infrastructure and follow the path of cellular communications. A distributed network infrastructure has been in place for many years fulfilling small business needs of wired and wireless connections to a certain extent But now, the number of Internet users and the demand for seamless Internet connectivity is increasing rapidly, and this has encouraged organizations to maximize their wireless local area networks. The expansion of WLANs, and thus power consumption and costs reaching their high peaks, made everyone think about the best architecture for wireless networks to be deployed. Moreover, managing and configuring all network devices consisting of access points, switches, and routers were becoming difficult with the expansion of networks.

In this paper, critical aspect of the centralized structure and also benefits that provide positive performance and overcoming most problems faced in existing distributed networks have been studied. It has been suggested that a centralized structure is a good infrastructure for a wireless network, but it has never been shown why this is beneficial. The information in this paper demonstrates the benefits of a centralized structure by identifying three key dimensions of a wireless network: architecture, management, and power consumption. They will be illustrated with the proposed algorithm and emulation.

The dominant architecture of today's network, which is distributed, is good only for small networks. Large networks require an architecture where data is continuously carried over longer distances, and intelligence is required for switching based on traffic and usage. A centralized structure is very intelligent and is able to fulfill all the requirements of today's emerging networks. It is creative in handling data and providing seamless connectivity, while maintaining criticality with respect to power and cost constraints, which are the most important design requirements in all infrastructures.

Management is also a very important constraint and is always discussed in architecture design. Managing all network devices efficiently is a challenging task in distributed networks. Every single device must be managed individually and monitored for maintaining prolonged uptime and data handling efficiently. Configurations of these devices also become a difficult task when they increase rapidly in the network. Lastly, deployment of these devices in large numbers requires considerable effort and work. Taking into consideration these requirements, it becomes a challenge for any organization to deploy, configure, and manage network devices. Centralizing the network eases the management and configuration of the network devices to a great extent, as the major configuration is only required on a central device.

Power consumption is another most important aspect that must be considered, as it is interconnected with costs. As power consumption increases, overall cost increases. Expansion of networks means that devices must be multiplied according to demand, and this requires more power consumption. If access points are added to the network, then simultaneously, switches are also added to provide power to these newly added APs. As a result, power consumption increases, which again increases the cost of the network infrastructure. Additional requirements in today’s networks also demand the “always-on” nature of access points, which keeps increased power consumption at a constant rate. Centralization brings intelligence to the network and switches APs back and forth between sleep mode and active mode, depending on the usability and amount of traffic handled by them. In this way, large amounts of power are conserved, and cost is greatly reduced.

Keeping in mind the above aspects and current problems, this paper has worked toward a central design by introducing intelligence in the controller while maintaining the efficiency of all critical aspects necessary for proper functioning of the wireless network. An algorithm has been designed to enable the controller to switch the access points on and off depending on user traffic and hence reduce the power consumption and costs incurred in managing those access points.

This paper is organized as follows: The first part outlines the theoretical perspective on the design requirements of wireless networks. Next, the benefits of a centralized structure in wireless networks are discussed. Then, an algorithm for running in a centralized structure is proposed. Finally, the centralized structure running the algorithm with a few laptops and switches is emulated, demonstrating the results between a centralized structure running a algorithm and a distributed structure. The paper concludes by showing the results of a centralized structure and the proposed algorithm in terms of power consumption, deployment, and management, and why it should be implemented in current wireless networks.

CHAPTER 2

NETWORK DESIGN

2.1 Design Requirements of Wireless Networks and Centralization Benefits

The popularity and rapid widespread use of wireless networks is undeniable and observable. The following play an important role in the popularity of wireless networks:

- Flexibility: Internet users are able to connect almost anywhere in the network with complete ease and comfort. Users are able to access all resources, such as servers, printers, and other systems, regardless of their location within wireless reach.
- Management: Much effort is required in managing wired networks, but managing wireless networks is easy.
- Minimal Hardware: Wired networks must use many interim devices for connecting all users to the network. With wireless networks, hardware installations and costs have been minimized to a great extent.

While designing a wireless local area network, it is important to ensure that its performance and its users are not impacted. In order to avoid such an impact and to maintain the critical requirement of energy savings, a list of design requirements to be implemented when designing and deploying wireless networks follows:

- Security: In wireless networks, security is the most important aspect and also a cause of concern in the transmission medium. A centralized structure brings many security mechanisms and encryption in the data flow in a wireless medium. A common feature of a centralized structure is the ability to support many service set identifiers (SSIDs) simultaneously from the same access point. Each SSID can also have its own authentication and encryption settings based on the capabilities of the clients and their service needs. Hence, a centralized infrastructure in wireless networks ensures considerable security.

- Throughput and Coverage: *Throughput* can be loosely defined as successful transmission of data over a transmission medium. This is also a major concern in wireless networks, since there needs to be a fair share of bandwidth among all users connecting to the network and proper data handling. A centralized structure knows the placement of all access points and signal strengths during all times of communication. It also has a holistic map of the wireless network in order to switch on and off APs according to traffic requirements. That is, if a centralized structure observes a reduced amount of signal strength or a reduction in performance, it will control these factors by appropriately powering the redundant APs. *Coverage* of the network is also a vital requirement for wireless networks. A centralized structure having a holistic map of the network eliminates dead spots and provides complete coverage of the network to all users.
- Redundancy: In the design of a network, designers recommend the availability of redundant access points for extra coverage. Reasons for this can be anything related to heavy traffic loads or when a device suddenly stops functioning. Redundancy depends on the specific type of organization and other requirements related to it. A centralized structure ensures creativity and maintains all the functionality of APs, while providing redundancy and efficiency simultaneously to all users in the network. A centralized structure will also help in calculating the number of access points to be deployed as redundant.
- Responsiveness and Immediate Availability: Whenever a user tries to connect to a network, s/he should be able to immediately observe the available SSID network. The

network elements should immediately respond to the user connecting to the network. A centralized structure maintains knowledge of all clients and devices in the network in order to provide an immediate response. This structure also ensures that some access points are always powered in keys areas for connectivity.

- Latency: Transmission delay and low throughput rates cause one form of latency. Applications must allow plenty of time for packets to arrive at their destination. A centralized structure resolves the delays and ensures that all users are accessing the network efficiently.

CHAPTER 3

LITERATURE SURVEY

3.1 Algorithms

This paper reviews the similarity of a number of other wireless networking systems, including experimental projects, research papers, and organizational projects. A summary of previously related work is discussed below.

Previous work in the area of wireless local area networks relative to reducing power consumption efficiently has been done at various levels, including construction of the algorithm, design of the infrastructure, etc. Reducing power consumption can be achieved on a larger scale by introducing intelligence into the network infrastructure at various levels by employing different kinds of algorithms. One research paper developed a green-clustering algorithm [1], which is very close to the work done here and shares most of its common goals, in particular the goal of reducing power consumption in a wireless network. The green-clustering algorithm was developed to enable the central controller in a WLAN to make certain decisions to power on and off portions of it based on certain pre-defined criteria like deployment, location of access points, and locally derived information. Hence, this algorithm was designed with respect to the centralized network infrastructure and its devices.

In the green-clustering algorithm [1], clusters of APs were positioned using a calculated metric as the distance. They were placed in close proximity to each other and deployed in large organizations. The basic idea was that if access points are close enough in the cluster, a single one would fulfill the needs of users, even those in the vicinity of other APs in the same cluster. This clustering concept was designed with respect to large organizations with high-density

WLANs because access points are placed very close to each other to provide overlapping coverage and high bandwidth.

If access points are placed at a specific calculated distance, then they can form clusters and would be able to support each other. The AP providing access for all users in the particular cluster is termed the ―cluster lead.” This access point provides a gateway for all users to that particular cluster and also helps other APs in that cluster to switch off. Figure 1 shows the coverage regions of access points and the cluster coverage region.

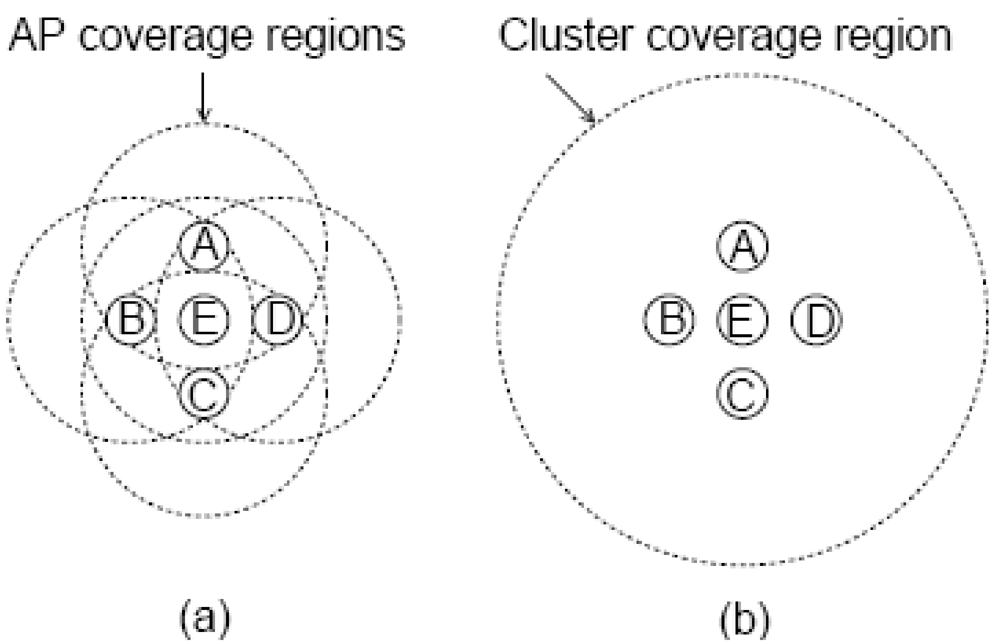

Figure 1: Cluster information [1]

If access points have a coverage region of 40 meters, then each AP will be placed a distance of 30 meters from each other. If these access points are placed in a certain area and at a certain angle to the above distance, then they will form a cluster. Any of the APs in Figure 1 can become a cluster lead and provide coverage access to the composite area. For example, if access point E is the cluster lead, then it can provide coverage to the composite areas of A and C. Hence, in the green-clustering algorithm [1], Euclidean distances and locally derived information are necessary for proper placement of APs and for this algorithm to work.

Further simulations were done with a custom simulator for replaying the association of users with access points. If users are associated with the ―A" cluster, then any of the cluster ―A" access points will provide coverage to the users. If a certain number of users exceeds the limit of a certain access point of the cluster, then additional surrogate APs will get switched on. Hence, it was proposed that considerable savings would be incurred, since there would be no need for additional APs being powered on all the time, as most of them could be switched off, and even the switches associated with them could be powered off.

Computations [1] have shown that if n out of total N access points can be powered off during an hour, then the percentage of power saved over the always-on APs will be (n/N X 100), thus leading to potentially enormous additional savings due to switch reconfigurations. If a 32-port switch consumes 320 W of power, then it is equivalent to the power consumed by 32 access points. Therefore if s switches out of S are powered during an hour, then the percentage of energy savings is $(n+(s\times32)/N+(S\times32) \times 100)$ percent. Overall, 30 to 50 percent savings was proposed in the green-clustering algorithm. This research was able to propose an algorithm that might help the controller make smart decisions for switching the access points on and off. Although the goals in that work are similar to those in this thesis, there are major additions and differences, which will be discussed in the forthcoming sections.

Depending on the network infrastructure and certain specified requirements, wireless networks can be operated or configured in two modes: infrastructure and ad-hoc. Infrastructure modes are basically a means of connecting the wireless network to the wired network, representing itself as a central point of reference for communications in the network. In a wireless network, the central point becomes the access point through which all users in a particular network gain access to Internet resources. The ad-hoc mode does not require any central device for communication. Clients or computers can simply create their own network and start communicating with each other. This mode is primarily used inside a common local area network (LAN) for local communication purposes.

A mixed mode, which is the combined result of the infrastructure mode and the ad-hoc mode [2], was introduced to maximize efficiency and minimize burden on the network. Under a common access point, users would be capable of switching between the ad-hoc mode and the infrastructure mode, depending on the traffic conditions in a cell. This switching of modes [2] would be transparent to users, beneficial in terms of network resources, and have the following strengths: better utilization of the network bandwidth depending on the type of communication, i.e., whether an intra-cell communication or exclusive Internet traffic; and greatly improved throughput of the network, since traffic would be divided evenly according to the requirements.

With the infrastructure mode, [3] some issues with the channel-selection mechanism in larger networks can occur. If access points are placed in common channels, then there is the possibility of sharing the same channel with other neighbor networks, which would cause interference. If access points are in a common domain and are observing interference and collisions, they should be able to change the channels appropriately. A distributed channel-selection algorithm [3] checks the quality of the channels of the AP and senses the need for a change in channel number. If there are many collisions or much interference, then the channels tend to change to a more appropriate channel based on previous feedback by the self-learning mechanism. Information required for this kind of algorithm is feedback on the access points for the interference in a particular channel, which is commonly provided by the 802.11 WLAN protocol.

Similar work for reducing power consumption was also approached by algorithms used on location estimation systems [4, 5], where users are located based on the wireless signals employing a wireless device with minimum computational resources and power requirements. Using a combination of data-mining techniques and analysis of real wireless data by probabilistic location estimation and the multiple-decisions tree-based approach, some very good results can be found. These results would help to identify the most utilized access points in the network, which in turn is an easy way to predict client locations. These decisions would allow a minimum of APs to be used, thus reducing the power consumption and wake-up time of the client. This particular algorithm is called the clustering and decision-tree-based method (CaDet). Here, the accuracy of location estimation is related to sampling time and energy consumption.

The above algorithm was further referenced by adding small hardware [5] to the clients, and numerous samples of the wireless signals were collected and tabulated, depending on the area. Hence, certain prediction models and sampling models were studied on the basis of tabulated results to find the location of users in the network.

Relative to the above work, the research in this thesis introduces an algorithm that defines a central controller that intelligently switches on and off access points depending on certain communicated messages that carry all of the important network parameters. The emphasis here is on a central device and access points or location from where all devices are controlled with maximum efficiency, thus resulting in overall reduction in power consumption and costs of the network.

3.2 Network Design

Research papers on designing a WLAN network also provide insight into deploying such networks in large organizations. One research paper provided insight into Carnegie Mellon University's wireless network to show the infrastructure that is usually needed.

Figure 2 shows how a wireless network is connected to a wired network. Each access point is connected to a hub. The hub is connected to the distribution switch, which is interconnected with the central switch. The central switch connection to the router in the central network is the connection link to the wired backbone network of the organization. It can also be called a link for connecting the wireless network to the wired network. This infrastructure [6] is very efficient, as it allows the wireless network to operate independently of the wired network, and both of them can be disconnected if necessary, in turn providing the flexibility required in networks.

One of the major challenges in designing wireless networks is complete coverage of the organization or the area and proper means of handling the traffic load during busy hours. The major components, which are always studied prior to the design of a network, are proper access point locations and proper assignment of radio frequencies to those access points. When deploying APs, a broad application should not be used instead of a detailed study on measurement techniques and careful propagation of radio waves across the complete organization to avoid any black holes in the network. The typical range of coverage of access points varies with respect to the environment. It can go up to 200 m in open environments and 20–50 m in office environments. Network performance can also be an issue in high-density networks that have a large number of APs, as interference can occur between access points.

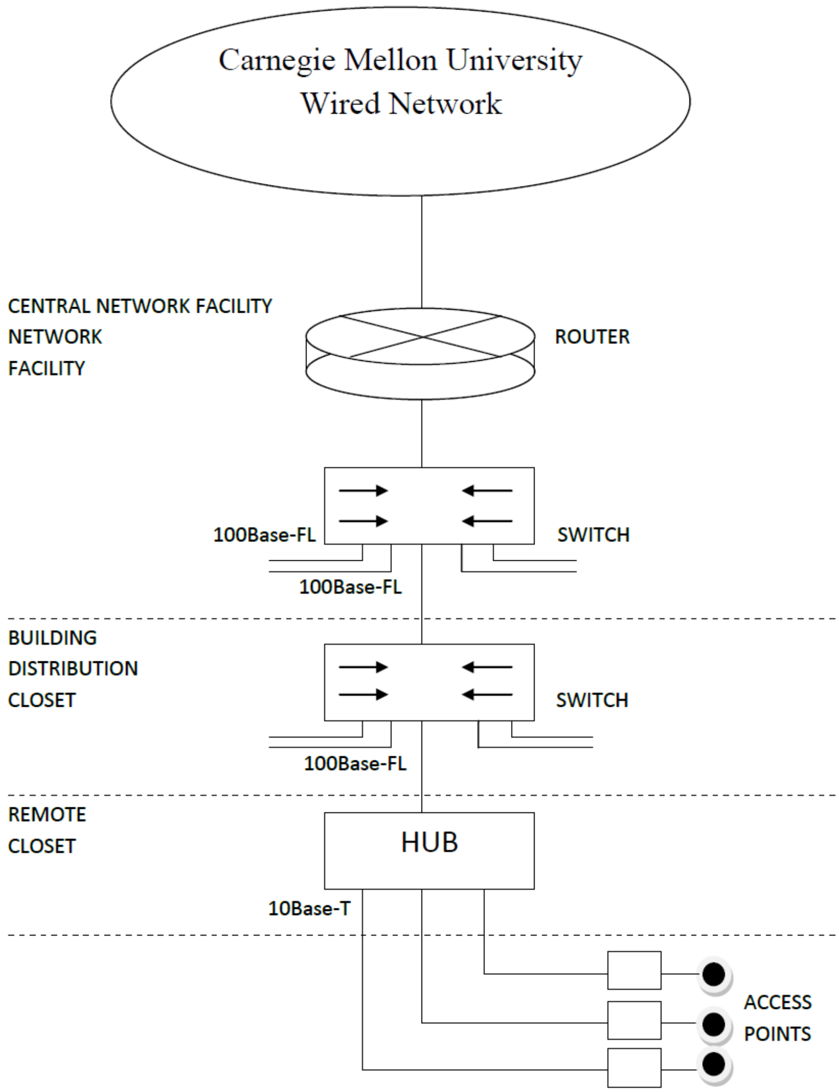

Figure 2: Network infrastructure at Carnegie Mellon University [6]

Two design layout techniques [6] that are useful in high-density situations are increasing the receiver threshold settings and multiple radio channels. If coverage is the main criteria in the network, then minimum threshold settings should be applied, and if capacity is the concern, then maximum receiver threshold settings should be active to reduce the coverage area. The use of multiple radio channels allows for the use of multiple access points to provide coverage in the same space due to the very high density. For example, three APs can operate in three different channels without any interference or spectral overlap and, at the same time, provide the coverage and access.

In one study [6], it was found that there are 11 SSS radio channels available in a 2.400–2.4835 GHz bandwidth used in North America. Of these channels, three have minimal spectral overlap: 1, 6, and 11. Hence, the use of multiple radio channels and receiver threshold settings for coverage and capacity can be very powerful tools in designing the most efficient networks. Figure 3 displays the normal positions of access points in a common network.

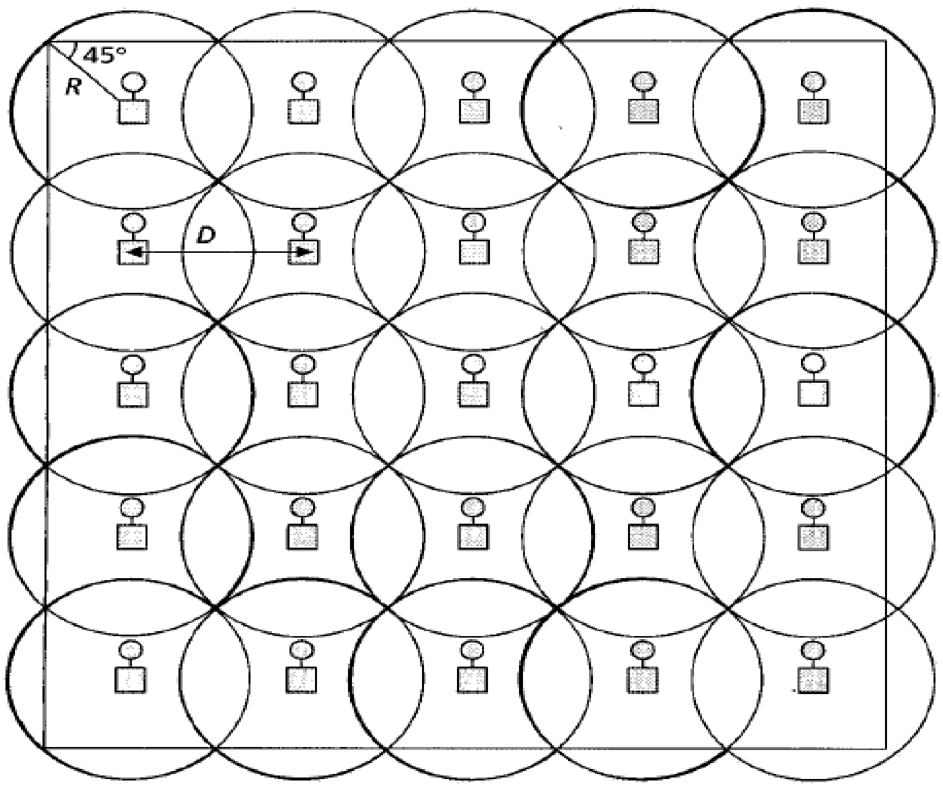

Figure 3: Placement of access points [6]

This paper demonstrated well the design procedure and steps that need to be taken when deploying a wireless network. This design procedure is generally always the same:

- Select access points.
- Test and redesign—adjust AP locations based on signal strength design measurements.

- Create a coverage map.
- Assign frequencies to access points.
- Document all access point locations, and assign a final set of all frequencies and signal strengths.

Selection of AP locations [6] is very iterative; all coverage areas are studied in detail to ensure that proper adjustment with respect to spectral overlapping is refined completely. Typically, the goal is to eliminate any coverage gaps and black holes, and to provide complete access for everyone throughout the network. Frequencies and radio channels are allotted to these access points properly and refined so that there is no channel overlap. Finally, complete coverage maps are drawn with the above specific parameters, like AP locations, radio channels, and signal strengths.

The design of large-scale WLANs [6] should be done in such a manner that complete coverage of the target space is ensured and that there is enough capacity to handle network traffic even during busy periods. The design should also consider proper location of access points, frequency assignments, and radio channels properly configured to avoid any co-channel coverage overlap.

Another paper [6] worked mainly on designing a network and locating access points but did not mention anything about power consumption of devices in the network. Power consumption, as well as cost, is a major criterion that should be studied and considered when designing these networks. Both can be reduced if the planning is efficient.

3.3 Site Survey or Propagation Modeling

The wireless LAN and wireless communication has become flexible after the introduction of wireless networks throughout the world [7]. Using electromagnetic waves, WLANs transmit and receive data over the air medium, thus eliminating the need for a wired infrastructure without affecting any connectivity issues for Internet users. Almost all WLAN systems are specified in the IEEE 802.11 standard, which is divided into two main layers: medium access control and physical. These layers provide independent functional components to the standard and also provide different radio transmission standards through which data can be propagated along the system. WLAN systems usually work on one of the below standards:

- 802.11b (802.11HR) - DSSS at 2.4 GHz with 1, 2, 5.5, and 11 Mbps data rates.
- 802.11g - OFDM at 2.4 GHz with 1, 2, 5.5, 11, and 22 Mbps data rates.
- 802.11a - OFDM in 5 GHz band with 6, 12, 18, 24, 36, 48, and 54 Mbps data rates.
- HiperLAN2 - OFDM in 5.15–5.35 GHz and 5.725–5.825 GHz bands, similar to the 802.11aPHY (with varying convolution codes).

Again, the indoor signal strength [7] differs from the outdoor signal strength due to multiple factors, such as diffraction, scattering, reflections, and refractions, which affect the signal strength to a great extent, as well as radio signal distortions, which are due to the environment and surrounding devices. Access points can be deployed easily in small networks depending on coverage signals specified by vendors. But in large networks, it is very difficult to deploy them. A more accurate procedure or method is required to ensure proper coverage and for handling all network functions like signal strengths, security, interference, and throughput. Presently there are two approaches [7]: the first is based on site surveys with many measurement analyses and prediction studies; the second is based on software planning using propagation models. Hence, this paper will discuss the issues related to these approaches and the advantages and disadvantages of using them to deploy wireless networks efficiently.

A site survey is usually done with a software-enabled device to calculate, on the basis of coverage, throughput and signal strengths [7]. The main goal of the site survey is to deploy the access points based on the coverage area. To successfully employ this site survey concept, it can be temporarily placed in a specific area, and certain tests can be taken to measure the strengths and coverage. Afterwards the quality of service (QoS) and coverage are measured and studied extensively. Based on the results from these survey tests, and considering both the location and the traffic density, proper positioning and configuration of access points can be undertaken. Future changes in the deployment of APs and other devices will again depend on fresh surveys, which can be undertaken numerous times according to need.

Quality of service is a far-reaching concept. ITU-T Recommendation E.800 [5] defines QoS as ―the collective effect of service performance, which determines the degree of satisfaction of a user of the service." It can be defined in such a way that there are certain parameters associated with it and certain mechanisms that evolve in the performance of the network based on the signals and coverage.

Network parameters like throughput, delay, jitter, bandwidth, and packet loss are always taken into account when designing networks. Throughput is the amount of data that can be transferred from one place to another, and delay is when transmitting or receiving a packet is late. Jitter can be the variation in the sequence numbers of packets sent and received, and is related to out-of-sequence packets. Bandwidth can be simply defined as a pipeline through which the data can be transferred, whereby the greater the bandwidth, the better the performance. These are essentially network parameters, which are calculated and noted when doing site surveys [7].

Usually, software provided by specific vendors can be installed in computers to calculate the above-mentioned parameters and to check connectivity. Also, specific instruments are available for calculating which signal strengths and other parameters can best be utilized. Some specific parameters related to signal-strength and access-point measurements, such as packet error rate, multipath, and received signal strength (RSSI), are always taken into account [7]. Some of the problems encountered when implementing techniques like the site survey were discussed in this paper and explained in detail. The authors carried out an experiment with two laptops, one with a directed helix antenna and the other with a built-in antenna. Signal strengths and signal-to-noise ratios were obtained and processed via a standard measuring tool. From the results, they discovered that the surrounding environment and number of people had a large impact on signal strength and coverage. Also taken into account were diversified measurements, such as closing the doors and measuring the variations, as well as opening the doors and measuring the variations. The results were quite impressive, showing how varied the signal strength and coverage of signals can become.

The fluctuations shown in Figure 4 are the results on the received signal strength when doors are opened. Similarly, considerable fluctuation is observed while rotating the laptop and with sporadic movement of people, both of which cannot be avoided at times. These differences in signal strength are usually undiscovered and unstudied, but they can lead to enormous network potential. Later on, measurements were taken while people moved, and again there was a tremendous variation in the signal strengths.

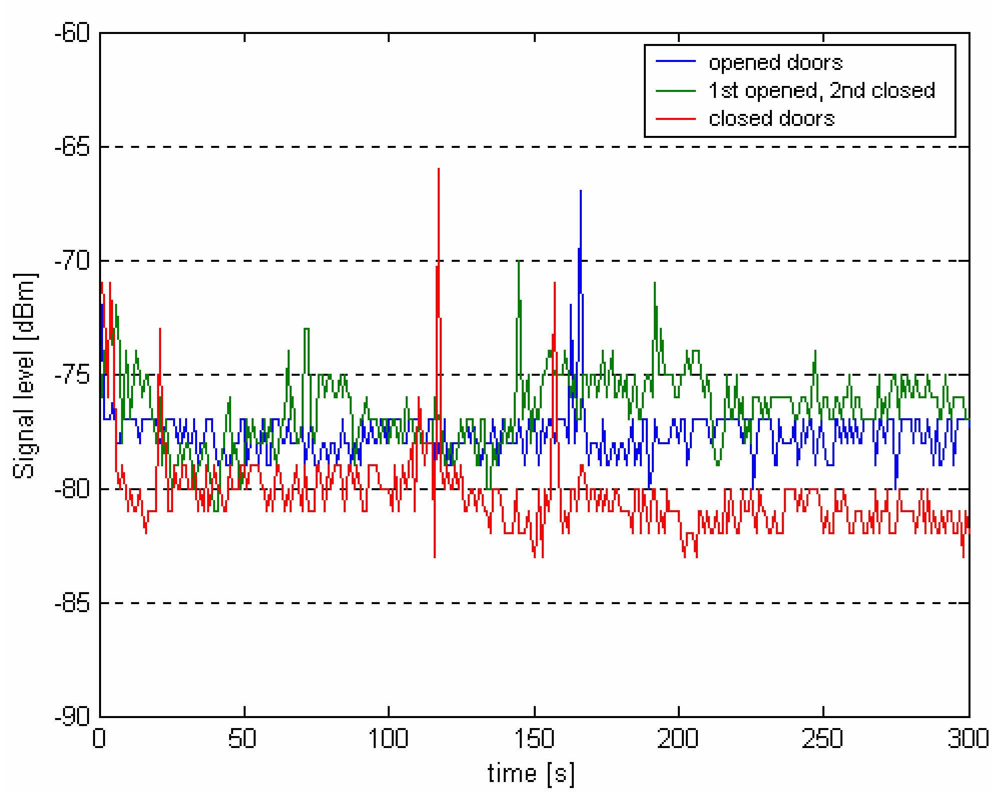

Figure 4: Influence of opening doors on received-signal strength [7]

A comparison of variations in received-signal strengths with people movement to received-signal strengths without people movement showed considerable change in the strengths. Figure 5 shows these differences in some detail, and the results are quite interesting. These graphs simply show the measurements of a simple experiment. However, the results will become more complex and complicated when these factors are measured in a real-time environment consisting of large distractions, a large number of people, electronic devices, and typical surrounding environments.

All of the above issues [7] need to be accounted for when doing an indoor site survey. Many measurements must take place and different techniques must be implemented to average the measurement results with respect to space, measurement antennas being deployed, and environmental conditions. More time and study are involved in these surveys in order to read a pattern for signal strengths and account for all variations caused by surrounding environmental factors and hindrances.

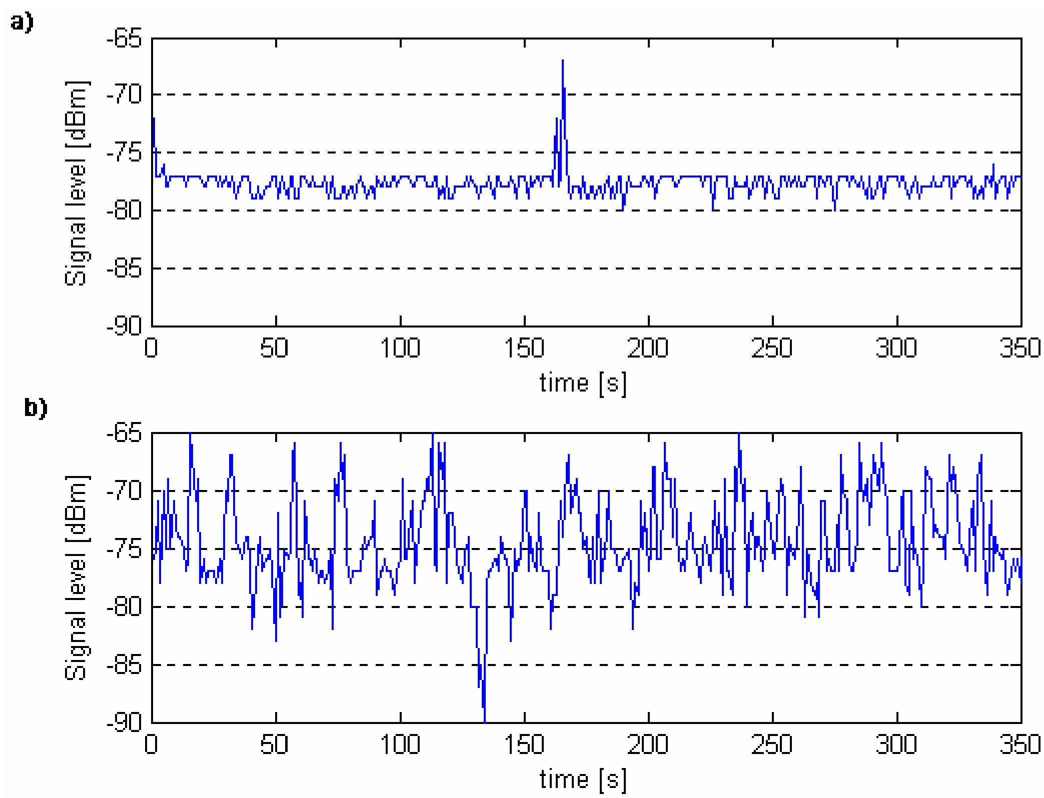

Figure 5: Signal strength received: a) with disruption and b) without disruption of moving people [7]

Propagation modeling [7], another method adopted for deploying the wireless network, makes use of software and certain simulations to test all configurations and design. This method is also cost-effective compared to the site survey because multiple measurements and empirical decisions can be eliminated. This software runs on specific models and depends on computation time, input data, and other requirements that are specific to each network. These models are again divided in to two types: deterministic and empirical.

Deterministic or semi-deterministic models [7] are primarily based on electromagnetic wave propagation theory. Some models, like ray tracing or ray launching, are based on geometrical optics, whereby radio wave propagation can be viewed as optical rays. It can be seen that diffraction and the wave guiding effect of the corridor are considered. In this model, all space-time properties, such as time delay, angle of departure, and angle of arrival, need to be considered for estimating the actual coverage area of the network. Propagation loss and accuracy are the common parameters necessary for common planning. But due to these extra-sophisticated parameters in propagation models, it becomes very time consuming and difficult to manage all of the measurement parameters. Hence, for these reasons, this model is not commonly used. Figure 6 shows the coverage prediction by this model.

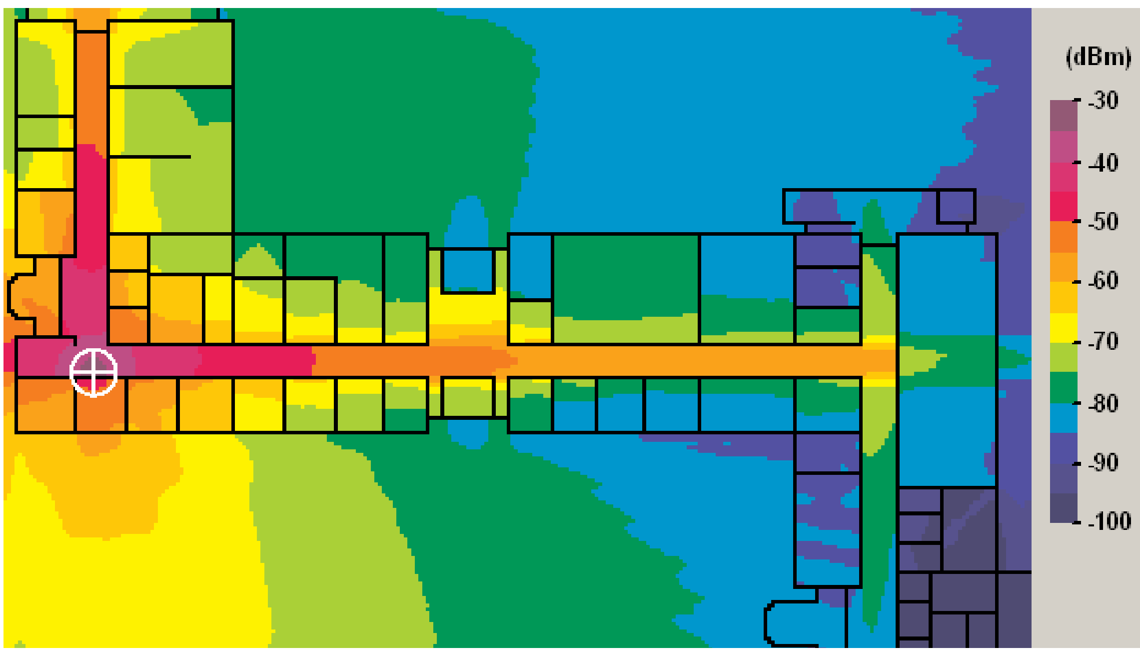

Figure 6: Coverage prediction by deterministic model [7]

Empirical and semi-empirical models [7] are primarily based on statistically processed measurements, which are very fast compared to deterministic models. One of the major examples is the one-slope model and multiwall model (MWM). Since these models are based on statistic-processed measurements, all prediction results are derived from the immediate close-knit expression, which consists of all parameters and provides results quickly.

Total path loss LTOT (dB) can be expressed as

$$L = L(P) + \chi \, TOT \qquad (1)$$

where L(P) (dB) is the average loss based on the position *P* only, and χ (dB) is random fading with a zero-mean statistical distribution.

The one-slope model [3] is a good way of calculating the signal of the location without having knowledge of the actual location layout. The path loss in dB is simply a function of the distance between the transmitter and the receiver antennas:

$$L\,(d) = L_o + 10n \log(d) \qquad (2)$$

where L_o (dB) is the reference loss value for the distance of 1 m, n is the power decay factor (path loss exponent) defining slope, and d (m) is the distance. L_o and n are empirical parameters for a given environment, which fully control the prediction. As an example, Table 3 shows a few values taken from various references. As indicated, the value of the power decay factor n is highly dependent on the type of structure or building, so it has a major influence on the resulting determination of the signal-level coverage. A typical example of the coverage prediction is shown in the Figure 7, which was calculated using different locations.

The values of the power decay factor n vary depending on the type of building and indoor environment. Hence, the value of n can change depending on location and can be adjusted well, according to the requirements. This model basically provides a very good tool when there is no information on the indoor environment and if a fast draft is needed for calculating the signal strengths around any location.

f (GHz)	L_0 (dB)	n (-)	comment
1.8	33.3	4.0	office [13]
1.8	37.5	2.0	open space [13]
1.8	39.2	1.4	corridor [13]
1.9	38.0	3.5	office building [14]
1.9	38.0	2.0	passage [15]
1.9	38.0	1.3	corridor [15]
2.45	40.2	4.2	office building [9]
2.45	40.2	1.2	corridor [9]
2.45	40.0	3.5	office building [16]
2.5	40.0	3.7	office building [17]
5.0	46.4	3.5	office building [18]
5.25	46.8	4.6	office building [17]

Table 3: Empirical parameters of one slope model [7]

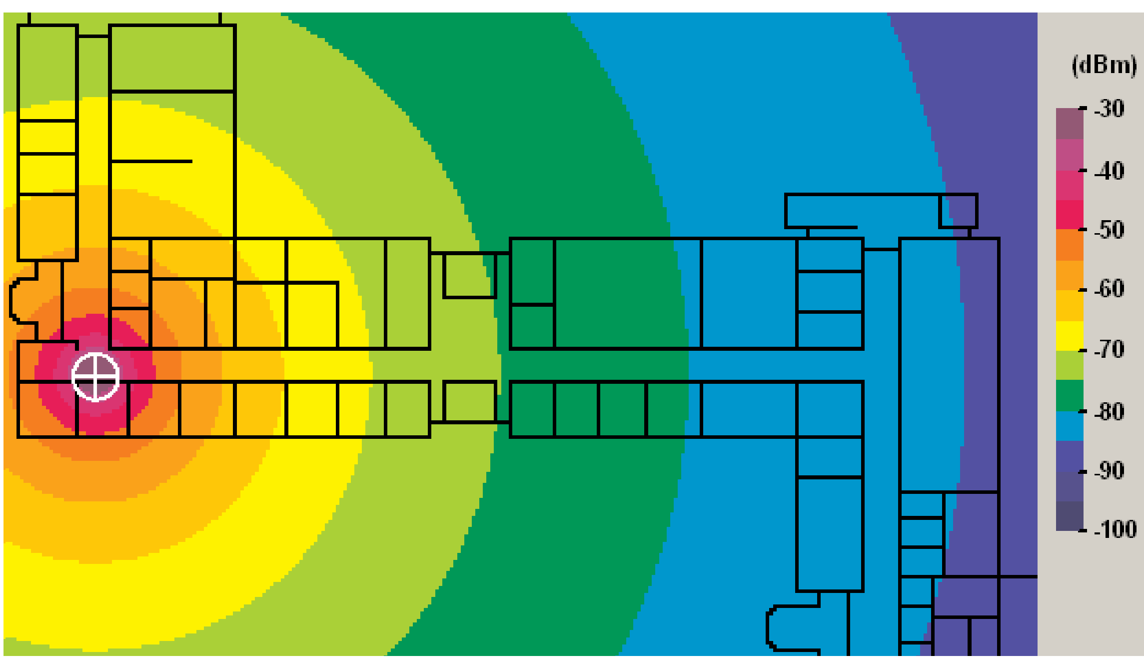

Figure 7: Coverage prediction by one-slope model [7]

A semi empirical multiwall model provides more accuracy and deep insight when compared with the one slope model. However, the results are occasionally site-specific, as some input of the floor plan description is required for better results.

The basic idea of an MWM is illustrated in Figure 8. The path loss between a transmitter and receiver L_{MW} is given by

$$L_{MW} = L_{FSL}(d) + \sum_{i=1}^{N} k_{wi} L_{wi} + k_f L_f$$

where L_{fsl} (dB) is the amount of free-space loss for the distance d (m) between transmitter and receiver antennas. With this model, several parameters are introduced, such as number of levels in the building, types of walls, and number of walls. The addition of this input brings about good accuracy and coverage results, which are closer to the real measurements. These models [7] are based on well-calculated expressions, which take into account all necessary parameters of the specific site or building so that coverage area and signal strengths can be estimated well.

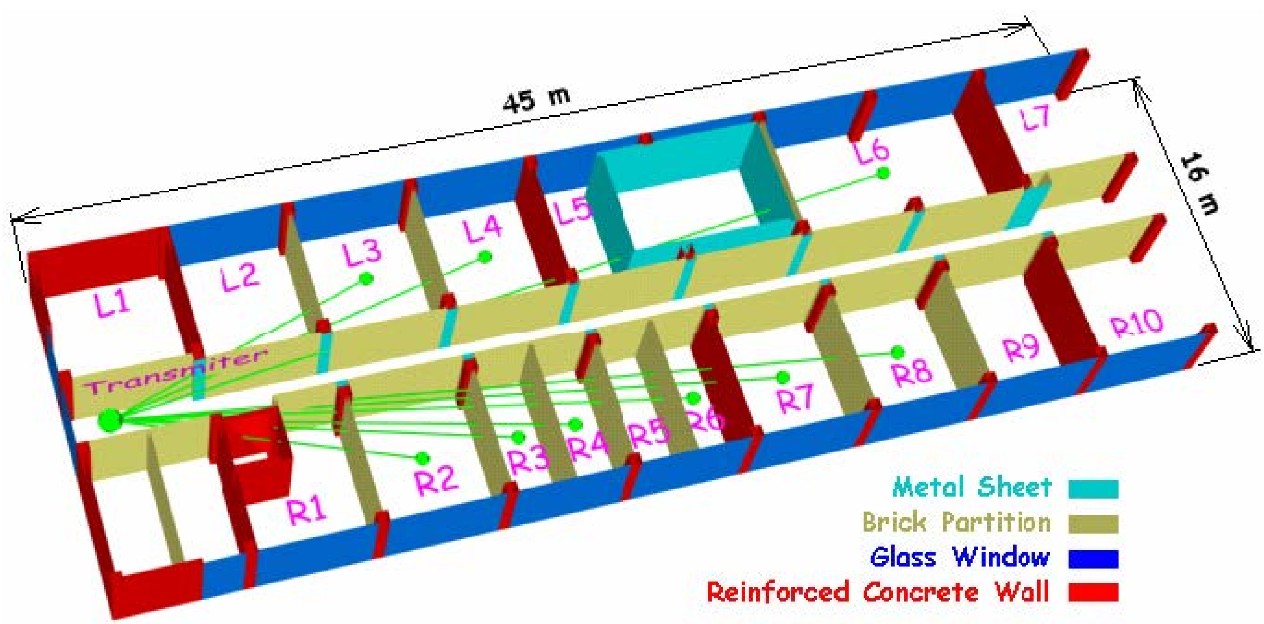

Figure 8: Multiwall model geometry

Most WLANs are deployed manually without any forethought in software planning. The initial design is usually done by a site survey after deploying a set of access points and then physically checking the signal strengths and coverage region. However, these models can be very useful when an estimate of the hardware and its appropriate performance review in the pre-build network are needed. Again, there are certain advantages and disadvantages to these specific models. The advantages are that it enables fast and flexible trials of the network performance without having to deploy the actual hardware, and very minimum input is required to derive the output. One of the major disadvantages is that it requires a skilled person to run the model appropriately; otherwise, there might be large differences in the results. This particular paper introduced methods for designing networks in a very detailed manner, including advantages and disadvantages. The network in this thesis was designed in a manner similar to that of the site survey method but with a power energy constraint as the main concern.

3.4 Constraint-Based Wireless Infrastructure

Other research [8] proposed and introduced a design idea in infrastructure-based networks that would be capable of supporting a data rate demand (data rate density) in a given area. Here, the design methodology was to determine the number of access points, frequency channels, power levels, and placement of access points that would satisfy a set of constraints introduced in the beginning of the network design process. The set of constraints [8] included data rate density, radio propagation, and received-signal strength intensity. Unlike optimization techniques that are available for enhancing network designs, the authors of this research proposed a design and algorithm based on a constraint-satisfaction problem.

The algorithm represented [8] a service region as a discrete space of grid size 1 m by 1 m. The grid points represented candidate locations for installing access points and specifying locations that required radio signal coverage. Their design considered two cases: the first allowed APs to be located at any grid intersection, and the second restricted AP locations to a more narrowly defined feasible space. This design approach is shown in Figure 9. The design algorithm was structured in two main parts: the first involved determining the access points necessary for a given service and initializing their configuration, and the second [8] implemented a solution algorithm that manages the constraints, such as power levels, frequency channels, radio propagation areas, and received-signal strengths, so that all requirements are fulfilled in the designed network. Again, two sets of input components were defined in this algorithm: the first being the network traffic demands and physical location or structure where the algorithm was being adopted, and the second being a set of inputs incorporating the path-loss models that would approximate the radio propagation in a given physical area.

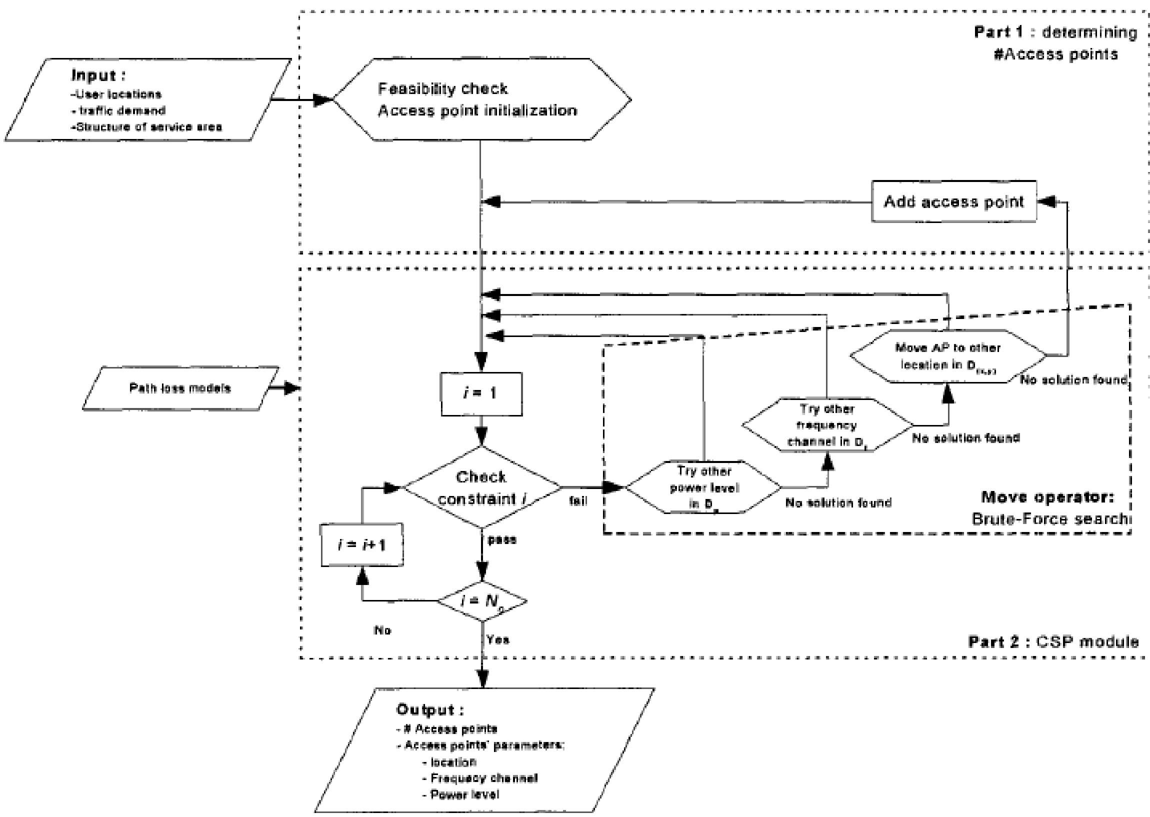

Figure 9: Constraint-based design algorithm [8]

With the above inputs, this algorithm [8] tries to maintain all constraints and calculate all parameters in such a way that all specific requirements are met to provide the right amount of bandwidth and signals to the Internet users. This algorithm was based on providing the data rates to the user according to their needs. If a larger consumption of data rate is required, then it allocates the parameters accordingly to meet the demands. If the requirements are small, then it ensures that only those demands are fulfilled, dividing equally all the necessary constraints and parameters in the wireless network. Hence, this algorithm [8] placed more emphasis on the data-rate demands of the users and accordingly allocated all the proper channels and all the parameters to ensure that the network functioned properly.

The capacity-driven approach [8] brought by this algorithm can be very useful because it depends on the data capacity demand by users rather than allotting bandwidth to each user, which could be a waste of total available bandwidth and might hamper those who need it more. This algorithm was explained in a very generic manner. One process was dedicated to the input, which comprised all constraints for providing necessary radio signal coverage and data rate according to the demand. The other process consisted of a brute force search technique, which appears after obtaining the demand of how many access points and how much bandwidth are required. This approach exhaustively searches for available APs with their individual power levels and frequency channels in that location and allots them properly in the right directions. This is accomplished with the assistance of strong processors, which in turn consume a large amount of computational power.

The above algorithm [8] and design approach introduced a new concept in WLAN design for considering capacity-driven constraints. It also made a strong statement by not reducing the hardware but rather using available hardware intelligently so that all the demand for data is met and the power consumption is reduced. However, the authors of this paper were not able to make this algorithm work in large-area networks, where searching for available locations of devices would take considerable time in providing service, which in turn would affect network efficiency. The concept was well intentioned and was able to explain a new way of looking at the design of networks by giving priority to some constraints and parameters, which could enhance the working of networks.

Considering the above designs and different methods for designing a network, a network in this thesis was designed based on real-time data analysis with access points, switches, and a wireless controller. The controller has all the information of the APs and also the amount of users connected to those APs. The methods of designing a network were not analyzed, but rather a central infrastructure design with an algorithm to minimize the power consumption and costs in the overall infrastructure was proposed. It was possible to reduce the overall power consumption and costs by simulating the basic design and algorithm in the testbed by 60 percent, which is a tremendous savings for any large organization.

CHAPTER 4

PROPOSED ALGORITHM

This section explains a proposed algorithm that might be running in a centralized structure and that adheres to the design requirements discussed in Chapter 3. The objective of this algorithm is to simply explore the magnitude of energy savings that can be achieved in its adoption and how it works in a centralization structure. This algorithm does accomplish the objective, and later on, details and how the algorithm is implemented in the prototype are explained.

4.1 Convex Hull Algorithm

This algorithm was designed for placing access points strategically in specific locations and which would be inherent to all above design parameters based on concepts of Computational geometry. We have chosen computational geometry design because it is concerned with the study of algorithms involving shapes, designs and curves [9]. Its most widely recognized use, however, is to describe the subfield of algorithm theory that involves the design and analysis of efficient algorithms for problems involving geometric input and output.

One of the fundamental structures in Computational geometry is called Convex Hull. Convexity is a very important geometric property. A geometric set is convex if for every two points in the set, the line segment joining them is also in the set. Now Convex Hull can be defined intuitively by surrounding a collection of points with a rubber band and letting the rubber band snap tightly around the points. Figure 10 shows a set of points and its convex hull design which depicts it as a specific area [9].

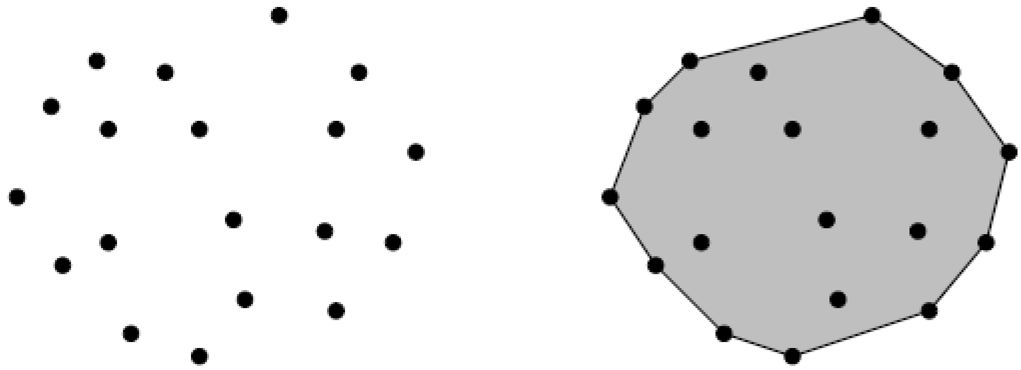

Fig 10: A point set and its convex hull [10]

There are number of reasons for choosing convex hull as our design structure as it is the simplest shape approximations for a set of points and can be used to approximate all kinds of shapes and structures [9]. Convex hull also fulfills all our design aspects which are mainly efficient coverage and limited usage of access points in the area at the same time. There has been a lot of ongoing study in algorithms for designing convex hull depending upon different shapes and different applications but we are only concerned with the global geometric design of convex hull and not in algorithms.

4.2 Convex Hull in Centralized Structure

Proper placement of access points is one of the best practices that should be adhered to in order to unleash the full performance potential of the Wireless Network. Earlier existing networks, access points are distributed mainly throughout interior spaces, providing coverage to the surrounding areas [10]. This method had the potential of diminishing the signal strength efficiency in the border areas. In our Convex Hull design, we have placed the access points on the perimeters of the area and corners which then have a potential to complement interior access points [10]. These border access points play a vital role in ensuring good location fidelity within the areas they encircle.

The access points that form the perimeter and corners of the location can be thought of as outlining the convex hull or set of possible device locations where the best potential for high accuracy and precision exists [10]. Revisiting the definition, the convex hull of a set S of points, denoted hull(S), can be regarded as the smallest polygon P for which each point of S is located either on the boundary or within the interior of P. Figure 11 illustrates the concept of convex hull with interior area and blacks circles representing access point locations.

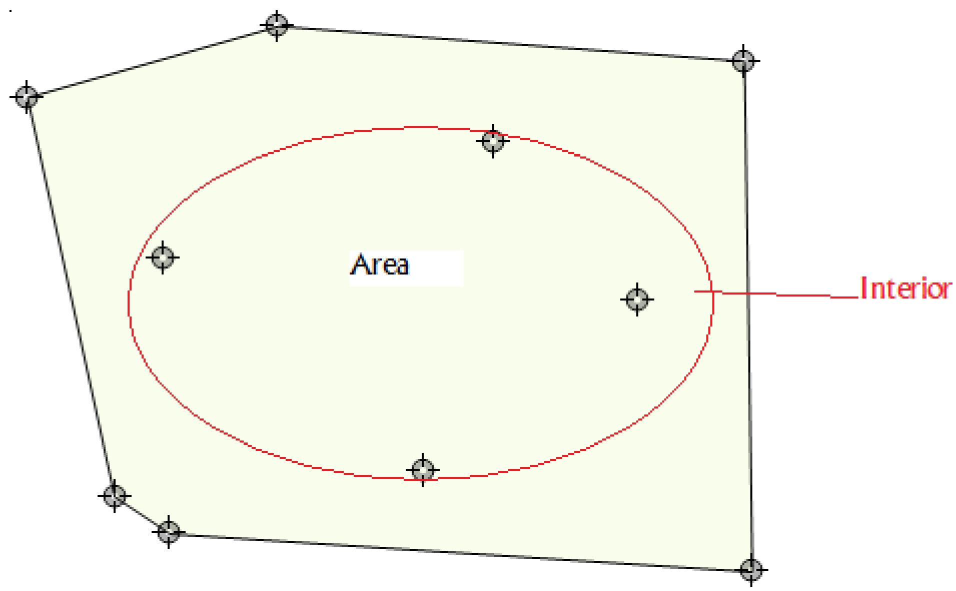

Fig 11: Convex Hull of Access Points

Energy Savings: Previously proposed distributed approaches for energy-efficient WLANs rely on locally derived information. For instance, a distributed approach called a wake-on WLAN powers an access point on or off if a sensor attached to it detects the presence or absence of users in its vicinity. As a result, more than a necessary number of access points may power themselves on when only a single AP may be sufficient.

The power-saver algorithm enables the central controller in a WLAN to make smart decisions to power portions of it on and off. These smart decisions ensure that all the requirements of an energy-efficient WLAN design are met while maximizing energy savings.

The power-saver algorithm is like a policy-based algorithm, which maintains up-to-date knowledge of those access points being deployed in a given area and makes a smart decision to switch on the required APs and switch off the redundant or unnecessary APs, thus not sacrificing the efficiency of any design requirements discussed above.

WLAN Coverage: In this algorithm, a close-knit array of access points is designed in convex hull manner and each section is called an area. These APs were placed in areas at a specific metric distance. Each access point was placed a specific distance from each other, and each area was located at a specific distance. All areas were interconnected with each other, and all access points were placed correctly so that all users in their vicinity received full access. Assuming that access points can cover a 50 m area, then one access point, Access Point A, was placed in Area 1, and another access point, Access Point B, was placed at the end of 40 m. Access Point B acted as a redundant AP for Area 1 and as an active AP for Area 2 to which it belongs. All access points were placed respectively in each area, and all areas were interconnected to maintain complete connectivity in the network.

4.3 How the Convex Hull Design Might Work in the Centralized Structure

Assume that interior access point in Area 1 is in sleep mode after large span of time scanning to check if there were any available users in that location. If an Internet user comes to a border location of Area 1, where border access point is residing, then it gets connected to it. This AP determines the density of the users, if they overpass a particular limit it senses that the signal strength is weakening, and then it communicates with the central controller to switch on the interior access point which is in the vicinity of border access point and an actual resident of interior area. When interior access point receives the message from the controller, then it activates itself. When interior access point is active, it scans the area and finds that particular user and helps that user to switch service to itself. After the switching of access points, interior access point communicates with the controller to inform border access point that it has acquired the user that was previously connected to it. The controller keeps knowledge of all access points and interconnected areas, since they are connected to it either directly or indirectly. Figure 12 illustrates the location of border and interior access points and the process of interchanging processes.

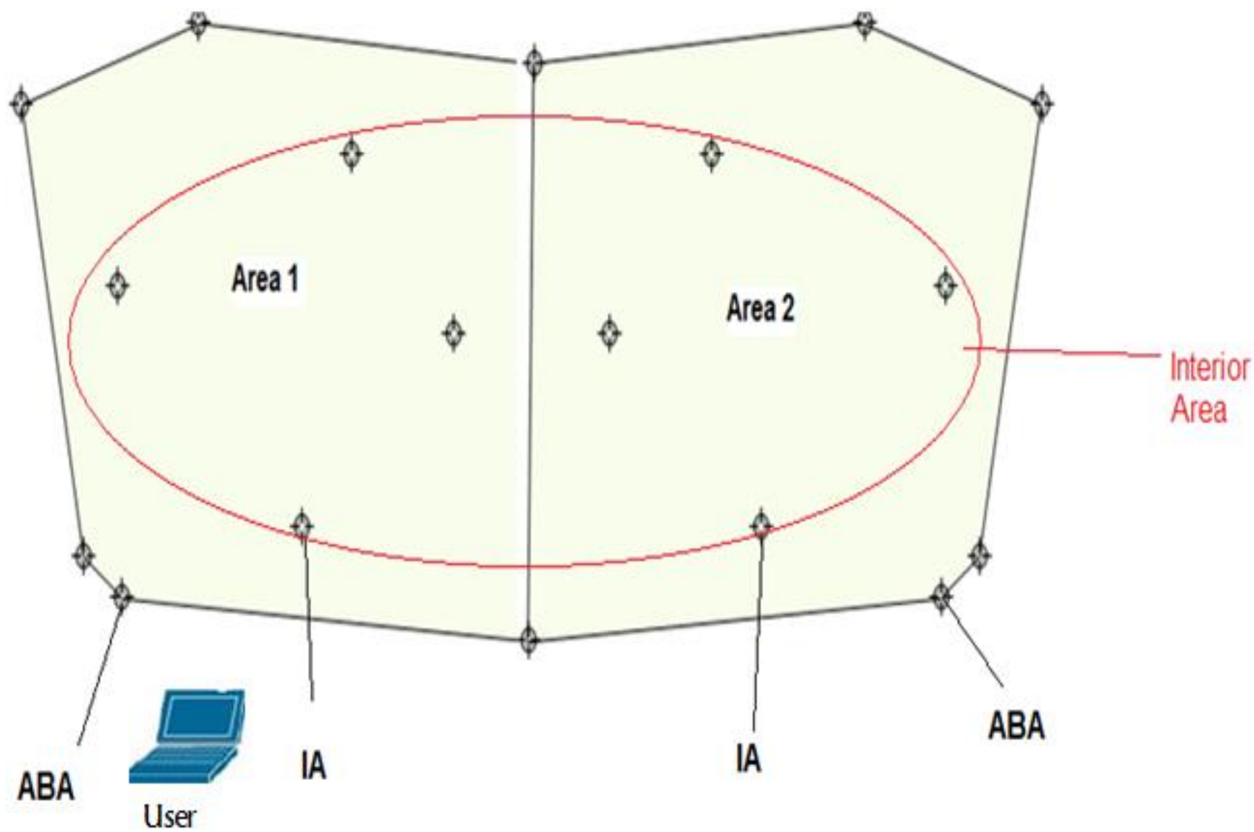

Fig 12: Illustration of border and interior access points in communication with user

The major emphasis is always on the border access points which are the first point of contact for the users and they are also responsible for maintaining the connectivity throughout the times. These border access points also have full potential in handing of users to other area's border access points when the user is on the move.

4.4 Energy Saving Potential by Convex Hull Design

The restriction on border access points of keeping themselves active consistently throughout gives opportunity to the interior access points to get switched off or get in to sleep mode. For every area of 10 access points, there is potential of energy savings for at least 4 access points which is a complete saving of half energy consumed in a network. Interior access points become active only when they are required due to excessive load or busy hours at the organizations, otherwise they are always energy saving sources in the network. Now, we can imagine the idle times of the access points at night's and non busy hours, this can lead to an immense potential of huge savings in terms of energy as well as money. Additional energy can also be saved by placing the border access accurately by implementing proper methods to measure signal strengths and identifying black holes, which then brings a potential to even switch off border access points.

Now, a discussion about the communication that was involved earlier between the controller and the access points follows. The communication message might, and should, include the following headers to communicate effectively:

- Area ID
- Distance
- MAC Address
- Sleep Period
- Users Connected
- IP Address, if involved
- Beacon Interval

Area ID is the identification of an area to which an access point is placed. This ID will be useful for informing the controller and the APs for switching on a sleeping access point or activating a redundant access point. Distance is a mandatory part of the communication message, since all access points are placed at all locations with respect to distances. Every AP is placed a certain distance from another AP, and each area will be placed a certain distance when connecting to another area. MAC address is the identification address of an access point, which should be forwarded in each communication message such that every controller knows exactly where to forward the message. The sleep period is identified when an access point informs the controller about its sleep time, how much time it will sleep, and when it will wake up. Users that are connected inform the controller about the frequency of users connected to each access point. The IP address is identified for each access point in the messages.

4.5 Additional Energy Savings with This Algorithm

The controller is the central position of contact in this network. In addition to controlling the access points, it can also power off the switches, which will considerably affect overall power consumption. This can only be achieved in locations where all switches and access points are connected together in one particular room or where all devices are placed under a close-knit group.

The convex hull algorithm works on achieving most power consumption by bringing intelligence into the controller so that it can power on and off access points based on user frequency. It also tries to conserve power by reducing the usage of access points and switches at certain locations.

This research proposed a new design methodology of placing the access point’s area wise by applying convex hull structure and examined the potential of energy savings through the mechanism in a centralized structure without sacrificing the network connection efficiency. The communication pattern and algorithm imposes all critical parameters in the network leading to a power-efficient network.

Furthermore Chapter 5 will demonstrate the emulation of communication pattern which is carried out in switching on and off access points in the centralized structure through the Wireless Controller. Chapter 6 has evaluated all the savings in terms of energy and cost.

CHAPTER 5

EMULATION

Instead of using any network simulator, the centralized behavior of a network infrastructure was emulated experimentally in a basic testbed. Figure 10 shows this centralized structure. Typically, centralized structures consist of hundreds of access points, switches, and controllers, as shown here. In such a controlled centralized structure, minimal tasks are handled by the access points, and the majority of tasks are handled by the centralized controller. The controller is connected to the access points either directly or indirectly. Communication between the APs and the controller is based on the algorithm mentioned in the previous section and helps the centralized structure to achieve the reduced power consumption by bringing the desired intelligence into the network.

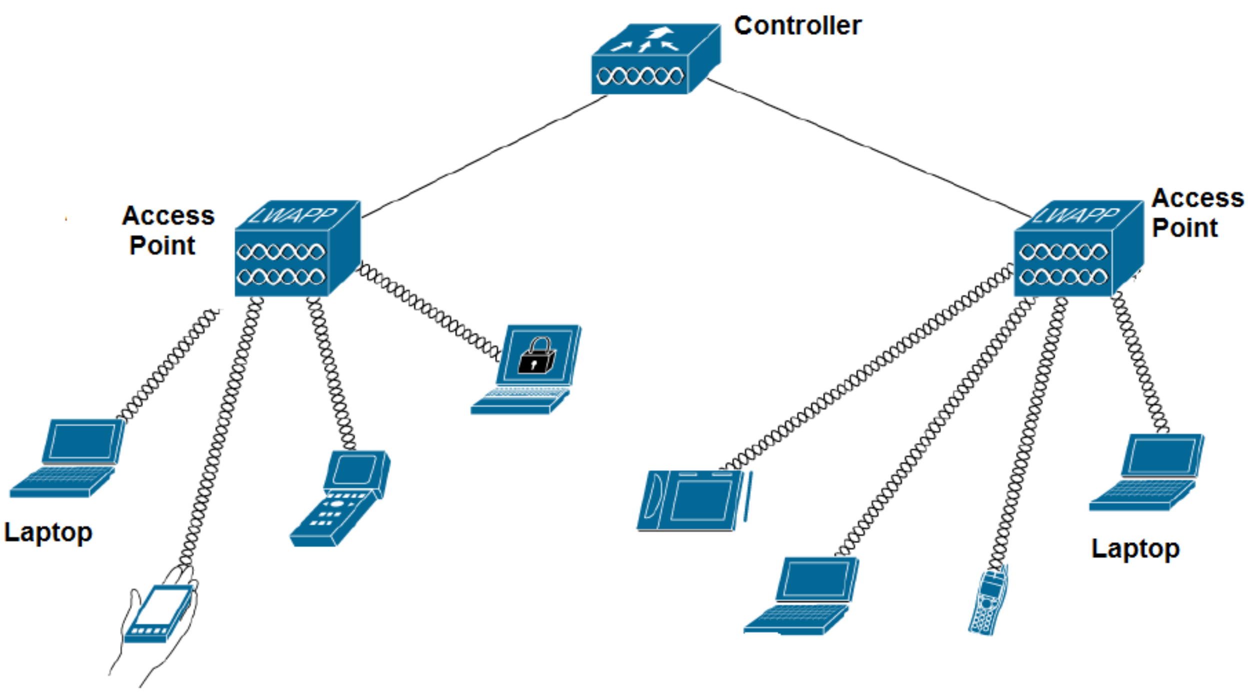

Figure 10: Centralized testbed structure [17]

In order to experimentally emulate the communication and the artificial intelligence carried out between the controller and the access points, the following devices were used:

- Laptops: Laptops were used in place of access points, since it was difficult to program access points into performing appropriately. Laptops can be programmed and forced to act like access points. Hence, the term access point is used in place of laptop throughout this paper.
- Switches: Switches were used to connect access points.

Figure 11 shows the basic experimental setup consisting of two access points, a controller, and a switch. The controller and access points were connected to the switch through an Ethernet cable.

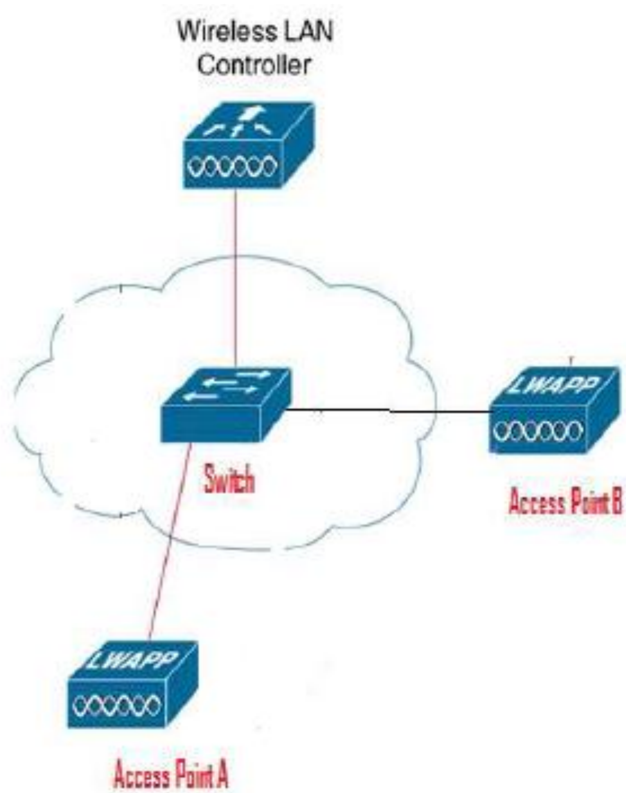

Figure 11: Emulation diagram

First, the establishment of communication between the controller and the access points and which resources were utilized to power on and sleep the access points according to user needs will be explained. A wake-on LAN program was used to wake an access point, and a socket program was used for sleeping. Both programs are explained in detailed in the following sections.

5.1 Wake-On-LAN

A wake-on WLAN program was used in the emulation to power on access points if they were in sleep mode. A Wake-on-LAN [15] is a combination of both software and hardware technology, which allows a computer to be powered on or awakened from standby or hibernation by sending specially coded network packets called “magic packets.” The wake-on WLAN works in the following manner.

- The target computer is in standby, hibernating, or shutdown, using power reserved for the network card.
- The network card listens for a specific packet, or magic packet.
- The magic packet is broadcast on the broadcast address for that particular subnet (or an entire LAN, although this requires special hardware and/or configuration).
- The listening computer receives this packet, checks it for correct information, and boots the computer if the magic packet is valid and contains the network card’s MAC address.

Magic Packet Technology

Magic packet technology was basically introduced to put a computer in an extremely low power state and still have it be manageable by a network system administrator. It is a feature designed to be incorporated into an Ethernet controller. This feature contains three modes:

- Magic packets enable.

- Magic packets frame detection.
- Magic packets disable.

Once the computer goes into sleep mode, the LAN controller in the network interface controller (NIC) is put into the magic packet mode. Here it scans all incoming frames addressed to the node for specific data sequences, which indicate to the controller that this is a magic packet frame.

The LAN controller in the NIC card scans for a particular data sequence in the Ethernet frames it is receiving [11]. If it sees any sequence of six bytes of all one's (FF FF FF FF FF) followed by 16 duplications of a targeted computer's MAC address (address of the machine to be awakened), then it simply wakes up that computer. This particular data sequence can be located anywhere in the standard Ethernet frame. The destination address in this frame would be a BROADCAST address for that subnet. The device also accepts the MULTICAST frame, as long as the 16 duplications of the MAC address match the address of the machine to be awakened.

If the 48-bit MAC address of the particular node is 01:22:33:44:55:66, then the LAN controller would be scanning for the following data sequence (assuming an Ethernet frame):

DESTINATION SOURCE MISC FF

FF FF FF FF FF 01 22 33 44 55 66 01 22 33 44

55 66 01 22 33 44 55 66 01 22 33 44 55 66 01

22 33 44 55 66 01 22 33 44 55 66 01 22 33 44

55 66 01 22 33 44 55 66 01 22 33 44 55 66 01

22 33 44 55 66 01 22 33 44 55 66 01 22 33 44

55 66 01 22 33 44 55 66 01 22 33 44 55 66 01

22 33 44 55 66 01 22 33 44 55 66 MISC CRC

The magic packet can go beyond a router by using UDP port 0, 7, or 9 to send packets, so if the router is configured to forward packets that are coming from these ports, they can go beyond the router. Another way to send a magic packet beyond a router is with DIRECTED BROADCAST. Magic packets are sent via the data link layer, which is not very secure, can be abused to a certain extent, but can be prevented by firewalls. Certain NIC cards have a special feature called ―Secure On," which allows users to store a hexadecimal password of six bytes. The NIC card will only wake the device if the password and MAC address are correct [17].

The wake-on WLAN concept in the experimental testbed was used to wake access points, with the help of a controller emulating the centralized structure behavior. A special code was used to run on all access points so that it could wake other access points with the help of a controller. Waking other access points during sleep mode was successful with the help of other access points by propagating the request through the controller. In this way, it is assumed that the communications pass through the access points and the controller. The emulation steps will be described in detail in an upcoming section.

5.2 Sleeping an Access Point

In order to sleep an access point, a socket program was used. This would again be propagated through the controller to other access points and then sleep it whenever it is idle.

5.3 Emulation Steps of Algorithm and Testbed

The idea behind the emulation was to understand the communications that might be occurring between the controller and the access points. Usually, information that is exchanged between access points and controllers is known, but the logic and the algorithm for establishing this communication is not known.

The basic communication modules involved in a centralized structure are the sleeping of access points by the controller and the waking of access points, depending on network traffic, which was explained previously. The complete communication pattern will be explained in multiple sections. Figure 12 is an exact replica of the emulation testbed, which was set up in the laboratory for emulating the complete communication pattern of the access points and controller.

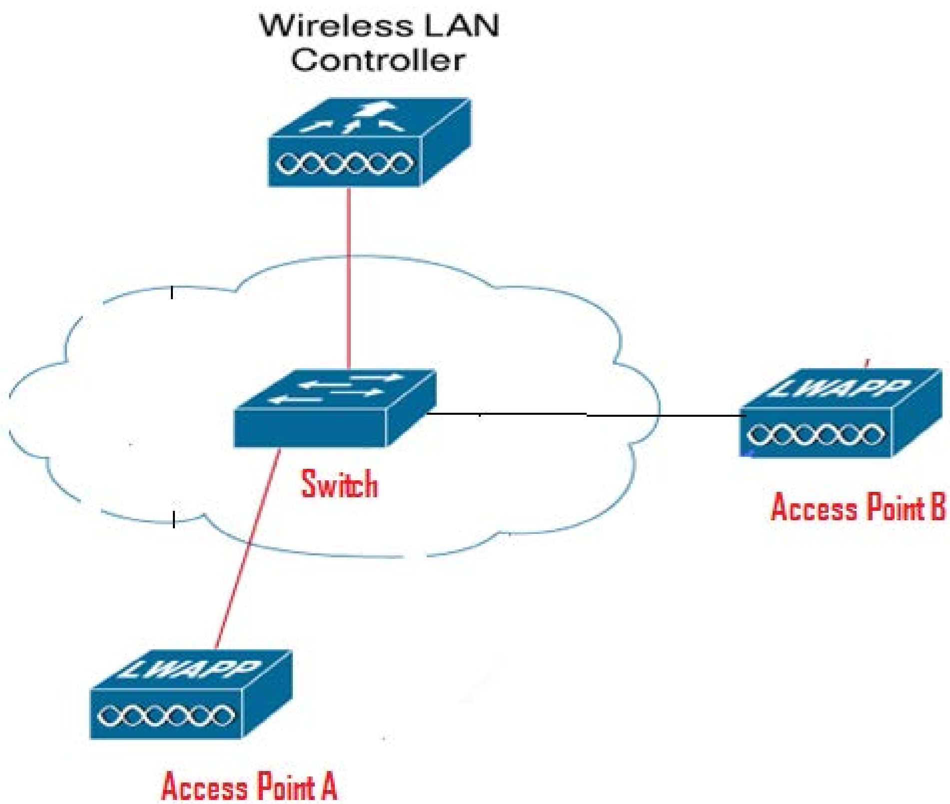

Figure 12: Initial testbed

Figure 13 explains the sleep scenario in which the controller sleeps/shuts down an internal access point on the basis of clients accessing that AP. The preceding phases explain the emulation about waking the access point if users try to associate with the sleeping access point.

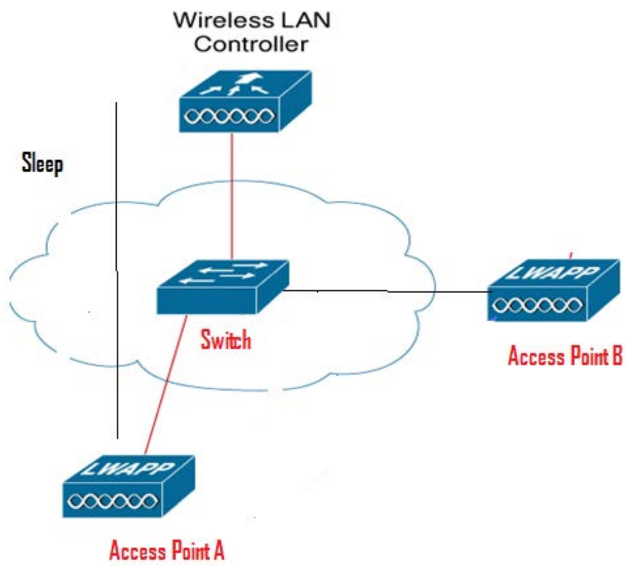

Figure 13: Phase 1. Controller sleeping the access point

The communication pattern was emulated using four laptops, since access points and a controller cannot be used directly in this setup. This is because access points and a controller come as a standard and cannot be programmed. In this testbed, one laptop represents a wireless LAN controller, two laptops act as an access point, and one laptop acts as a client accessing the access point. The complete experiment was simulated in six phases in order to understand the communication pattern that might be evolving in the standard centralized structures.

The controller and access points were all connected via a switch through the Ethernet cables in the tested. This typical setup in which the switch is providing power over the Ethernet is widely available and used in a centralized structure where the controller and access points are hooked to a common power source and connected directly or indirectly with each other for sharing common updates. Even here, the controller is responsible for managing access points according to the needs of Internet users in the network. For this experiment, a four-port Ethernet switch was used, and four Dell laptop computers were interconnected with each other.

5.3.1 Phase 1

In Phase 1, it was assumed that there were no clients in the vicinity of Internal Access Point A, and it was put to sleep by sending a sleep signal by the controller. This was done by the socket programming in C language. The intention behind sleeping the access point was mainly to reduce the power consumption and to emphasize the technology in wireless networks, which is very productive in nature.

This technology is already evident in a Cisco centralized structure, which was introduced due to the nature of all networks where all access points need not be powered on all the time in order to have radial coverage of the network. A prototype is being built to study the kind of technology that might be employed when such concepts come into play. As a starting point to this work, it will include a very basic idea for sleeping the access point by executing a program from another system so that another access point can be put to sleep. The programming, which is a basic implementation of socket programming, will be explained in a later section.

5.3.2 Phase 2

As shown in Figure 14, while Access Point A is sleeping, whenever a client user comes into the vicinity of access points, it will search for those access points that can be reached with strong signals, and if it does not find any, then it will search for an alternate access point to which it can connect. This is the usual way users connect with access points in the network, and it has a very negligible delay, if any. As can be seen in Figure 14, the user gets connected to Border Access Point B, which is done with low signals due to its exceeded limit. As soon as the client is connected to Border Access Point B, an exchange of information takes place between Access Point B and the controller about all of its configured parameters including power levels, signal strengths, and connected users. The controller and access points share all necessary information to keep the database up to date at all times.

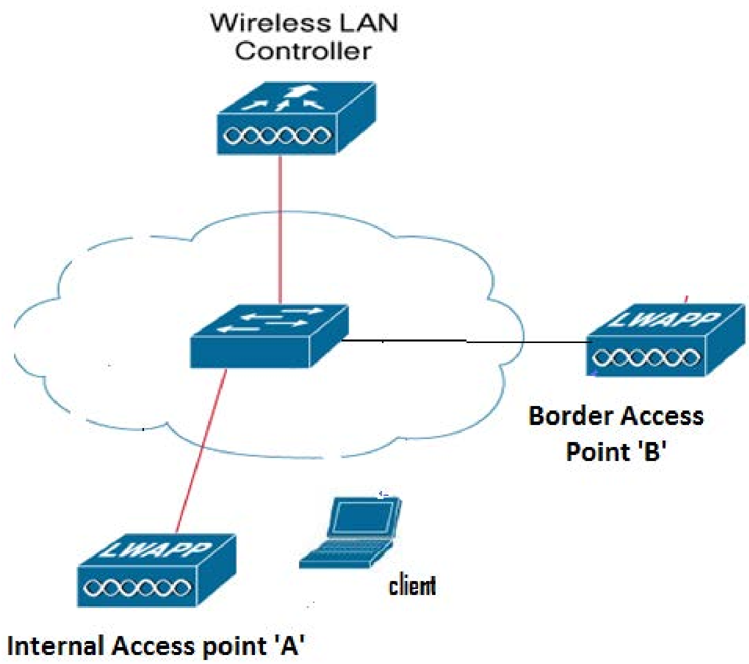

Figure 14: Phase 2. Internal Access Point A sleeping

5.3.3 Phase 3

The updated database is very vital for the controller to have all the time and keeps exchanging all the time for the proper funchioning of the network. The controller also provides all the necessary information to access points in order to make them switch from on and off according to the requirement. Figure 15 shows the communication pattern of updating databases between clients and controller. These concepts are necessary for the proper functioning of current centralized structures. From this testbed and the prototype implementation, it is believed that Border Access Point B sends information to those access points that might be in the vicinity of its area, indicating that it needs help in switching the load.

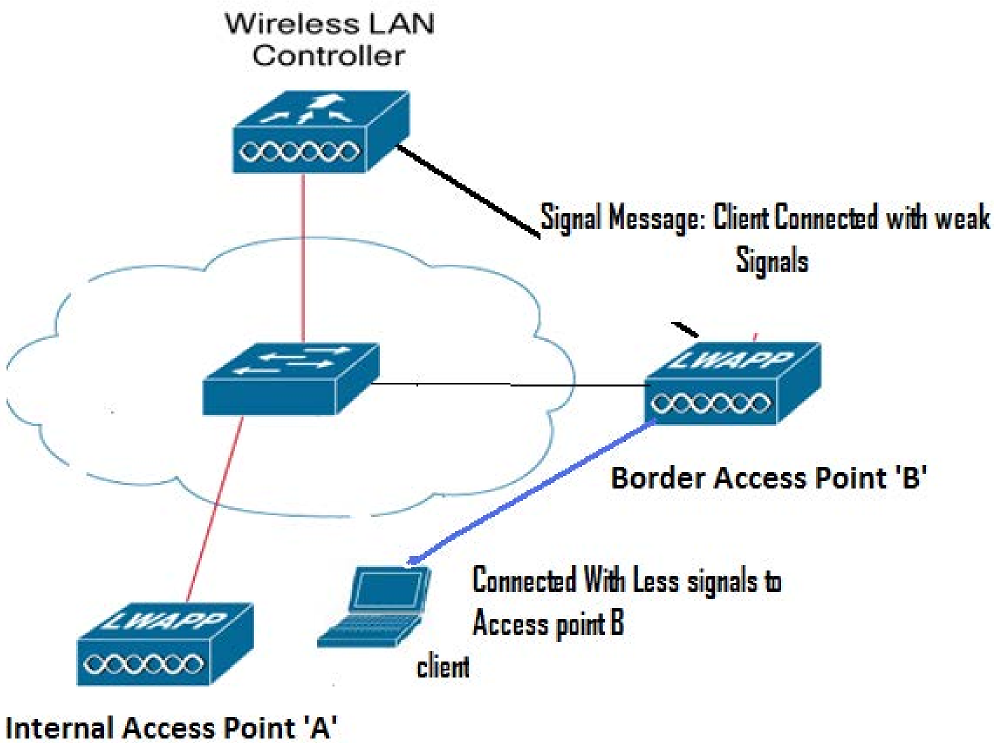

Figure 15: Phase 3: Client in vicinity of sleeping Internal Access Point A but getting connected to Access Point B

5.3.4 Phase 4

Internal Access Point A responds since it is the closest AP in that area, but it is also possible to call a redundant access point location. When Border Access Point B sends information about the load, power levels, and signal strengths through the controller to Access Point A, the controller triggers the wake-on WLAN signal to Internal Access Point A. When Internal Access Point A receives the wake-on signal, it becomes active with a refreshed interval having full power levels and a strong signal. Figure 16 shows this communication graphically.

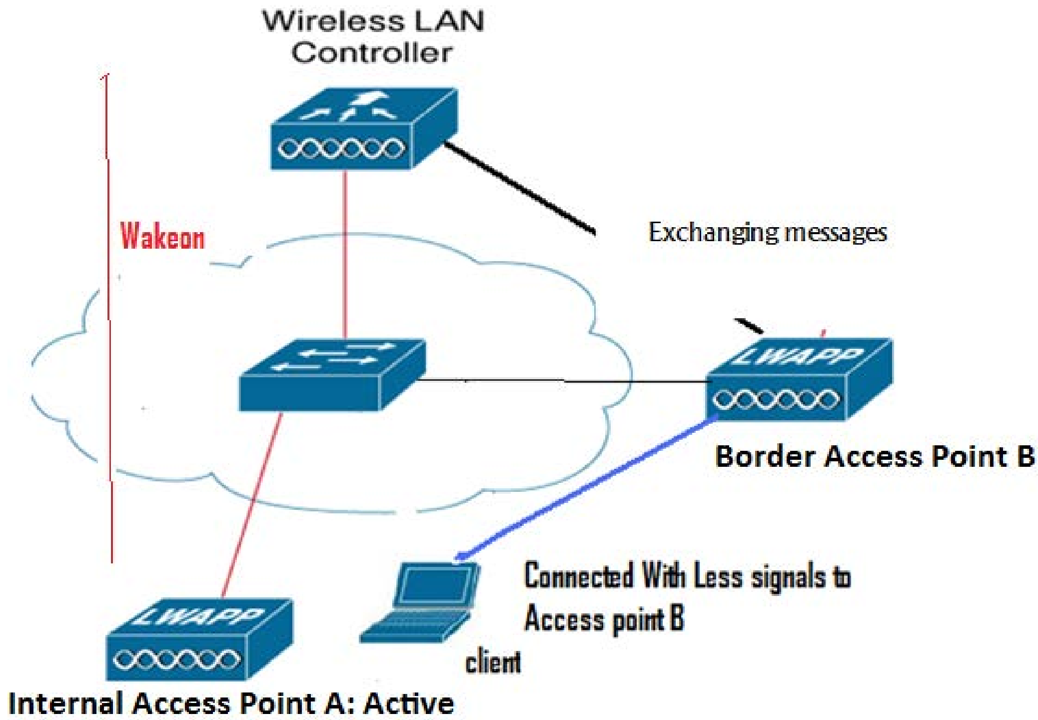

Figure 16: Phase 4. Now-active Access Point A after receiving wake-on WLAN signal from controller

5.3.5 Phase 5

When Internal Access Point A is active, it begins propagating signal strengths, updates all parameters again to the controller, and gets to know other APs through the controller. In the meantime, the client computer is still connected to Border Access Point B but comes under the vibe of Internal Access Point A. Phase 5 is shown in Figure 17. This is the usual way a laptop connects with access points in current networks, informing them about all available existing networks and probable signal strengths associated with those networks.

Access Point B stays connected with the controller while the client becomes completely associated with Access Point A. All information is again shared with the controller so it can maintain an updated connection table and all updated parameters from all APs, keeping the data current at all times for managing AP connectivity and switching off or on the access points appropriately according to the requirement.

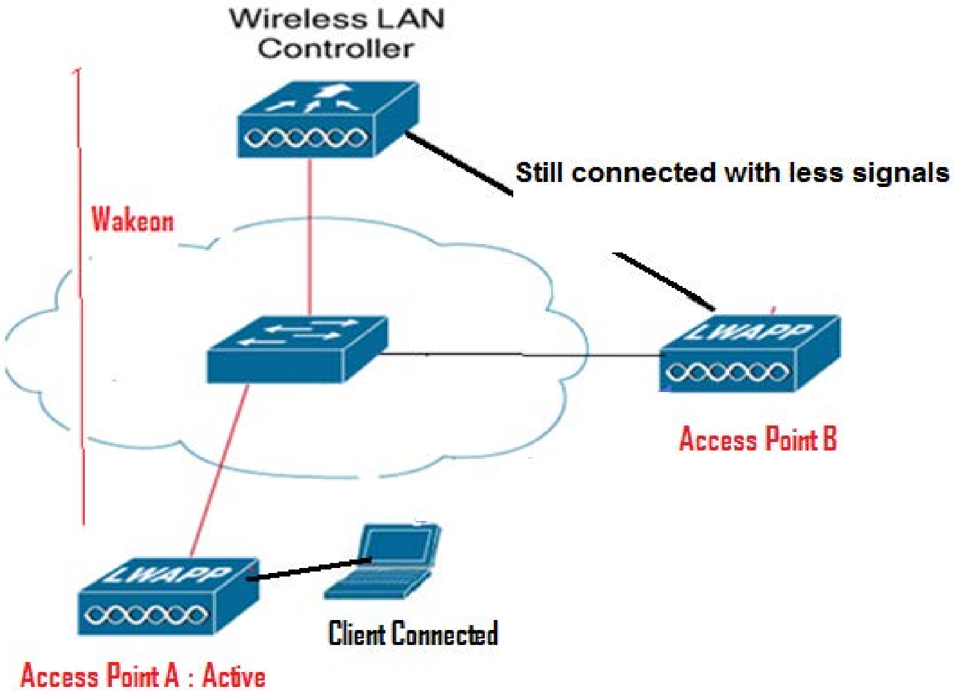

Figure 17: Phase 5. Client gets connected to Access Point A

5.3.6 Phase 6

After some time, the controller checks the Access Point B state table to verify if there are any additional clients associated with this AP exceeding its limit. If there are clients, then the controller will switch on other access points appropriately. In Figure 18, it is assumed that there are no clients connected to the access point, and the controller will send the sleep signal to Access Point B in order for it to settle down in a sleep state.

The client now changes the connection from Access Point B to Access Point A after observing stronger signal strengths. Again, all information is propagated periodically between the controller and other access points in that certain area. The aim here is always to have the highest bandwidth efficiency and have all clients connected with the strongest signals.

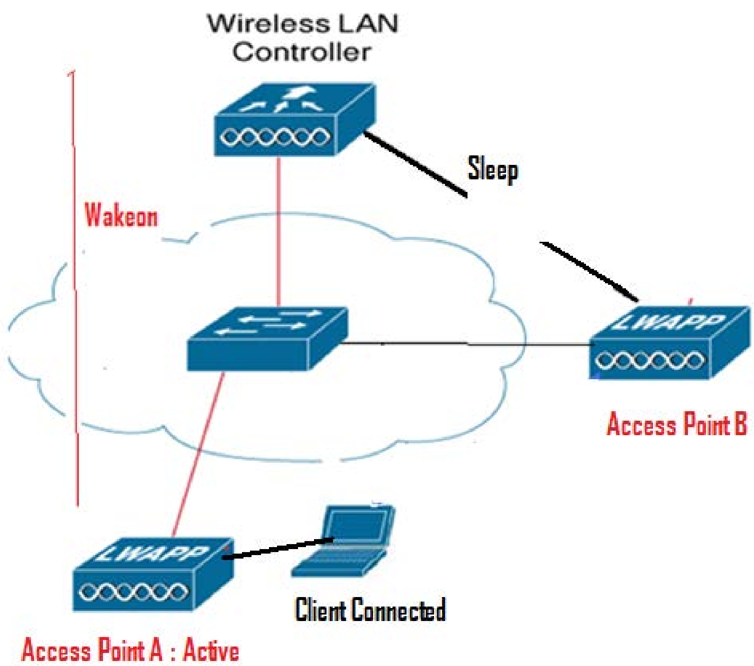

Figure 18: Phase 6. Sleeping Access Point B when no clients are connected

Phases 1 to 6 are basic implementations of a working centralized structure, where the controller intelligently controls the funchioning of access points while delivering efficient radio-signal coverage and reduced power consumption throughout the network. This small prototype testbed simulating a centralized structure was built as a starting point for others to continue this work. Not all of the funtionalities were added to this prototype, but the major ones, like switching on and switching off the access points through the controller, were added. An algorithm was proposed, and the possibilities that might be running in the existing centralized structure by which the controller makes creative decisions with access points and switching between them, according to the requirement, were studied.

CHAPTER 6

EVALUATION

6.1 RESULTS

This section presents results of the previously explained experiments, evaluating them in different scales. Not all of the functionalities and features in the centralized structure were able to be built, but the basic idea of reducing power consumption and intelligently switching access points on or off according to the client requirement was successfully implemented. Presented first are the results of a non-centralized network infrastructure where all access points are active in a certain location. To demonstrate the results better, a location that has a dense deployment of access points was selected to show the major differences in results.

Considering a location consisting of 100 to 200 access points, which is a common installation location in major corporations and universities for evaluating on the basis of cost and power consumption, every access point and switch requires a minimum amount of electricity to function properly. Efficiency can be defined as the measure of power relative to the amount of work accomplished. When devices use more power to perform the same amount of work, they are considered to be less efficient, and low efficiency will lead to increased costs in building the network infrastructure:

Cost incurred = energy consumed per hour * cost of power for
every kilowatt (kWh) * number of hours access point is running

Assuming that the median California rate is 15 cents per kilowatt hour (kWh), then

Cost of one access point running 24 hours:

= 0.01 kWh * 0.15 cents/kWh * 24 hours

= $0.036

Cost of one access point running 24 hours for one year:

= 0.036 * 365

= $13.14

Total cost for 200 access points in the network:

= 200 * 13.14

= $2,628

Total energy consumption of 200 access points in the network:

= 200 * 0.01 kWh * 24 * 365

= 17,520 kWh

Table 4 shows the final energy consumption and cost of access points in a non-centralized structure.

TABLE 4

ENERGY CONSUMPTION AND COST OF ACCESS POINTS
IN NON-CENTRALIZED STRUCTURE

Energy per access point	0.01 kWh
Energy per access point per day	0.24 kWh
Energy per access point per month	7.20 kWh
Energy for 200 access points per year	17,520.00 kWh
Cost per access point per day	$0.036
Cost per access point per month	$1.08
Cost per access point per year	$13.14
Cost for 200 access points per year	$2,628.00

To accommodate 200 access points, five switches with 48 ports are required, and each switch consumes 100 W on average. The energy consumption and cost of the five switches is as follows:

Average energy consumption of five switches per hour:

= 0.1 kWh

Average energy consumption of five switches per day:

= 0.1 * 5 * 24 hours

= 12 kWh

Average energy consumption of five switches per month:

= 12 kWh * 30 days

= 360 kWh

Total energy consumption of five switches per year:

= 12 kWh * 365 days

= 4,380 kWh

Cost of energy consumption of five switches per year at 15 cents/kWh/year:

= $.15 * 4,380 kWh

= $657.00

Table 5 shows the final calculated values for the energy consumption and cost of switches in a non-centralized structure.

TABLE 5

ENERGY CONSUMPTION AND COST OF SWITCHES
IN NON-CENTRALIZED STRUCTURE

Energy per switch	0.1 kWh
Energy per switch per day	2.4 kWh
Energy per switch per month	72.0 kWh
Energy for five switches per year	4,380.0 kWh
Cost per switch per day	$0.36
Cost per switch per month	$0.45
Cost per one switch per year	$131.4
Cost for five switches per year	$657.00

Total energy consumption when both switches and access points are combined:

= 4,380 kWh + 17,520 kWh = 21,900 kWh

Total cost incurred in managing the power of switches and access points:

= $657 + $2,628 = $ 3285

The above results and calculations were based on the assumption of a non-centralized structure where all switches and access points are actively consuming energy and running all the time.

The second set of results are from a centralized structure where all access points are intelligently switching on and off on the basis of algorithm. Table 6 shows all observations of a centralized structure with access points running algorithm.

TABLE 6

ENERGY CONSUMPTION AND COST OF ACCESS POINTS IN CENTRALIZED STRUCTURE

Energy per access point	0.01 kWh
Energy per access point per day	0.24 kWh
Energy per access point per month	7.2 kWh
Energy for 200 access points per year	8,760.0 kWh
Cost per access point per day	$0.036
Cost per access point per month	$1.08
Cost per access point per year	$13.14
Cost for 200 access points per year	$1,314.00

When a centralized structure with the proposed algorithm was deployed, the energy and cost consumption of access points were almost half when compared to the non-centralized structure. However, the consumption for switches in both infrastructures was the same because these devices need to be powered on all the time in order to provide energy to the active access points. Values for the energy consumption and the cost of switches in a centralized structure are provided in Table 7.

TABLE 7

ENERGY CONSUMPTION AND COST OF SWITCHES
IN CENTRALIZED STRUCTURE

Energy per switch	0.1 kWh
Energy per switch per day	2.4 kWh
Energy per switch per month	72.0 kWh
Energy of five switches per year	4,380.0 kWh
Cost per switch per day	$0.36
Cost per switch per month	$0.45
Cost per switch per year	$131.40
Cost for five switches per year	$657.00

As shown in Tables 6 and 7, it is clearly evident that deployment of a centralized structure with algorithm makes a considerable difference in terms of cost and energy consumption. If these results are observed in larger organizations, the difference would be more evident and would be the best solution for deployment. Figure 19 shows the differences in power and costs between a centralized structure and non-centralized structure.

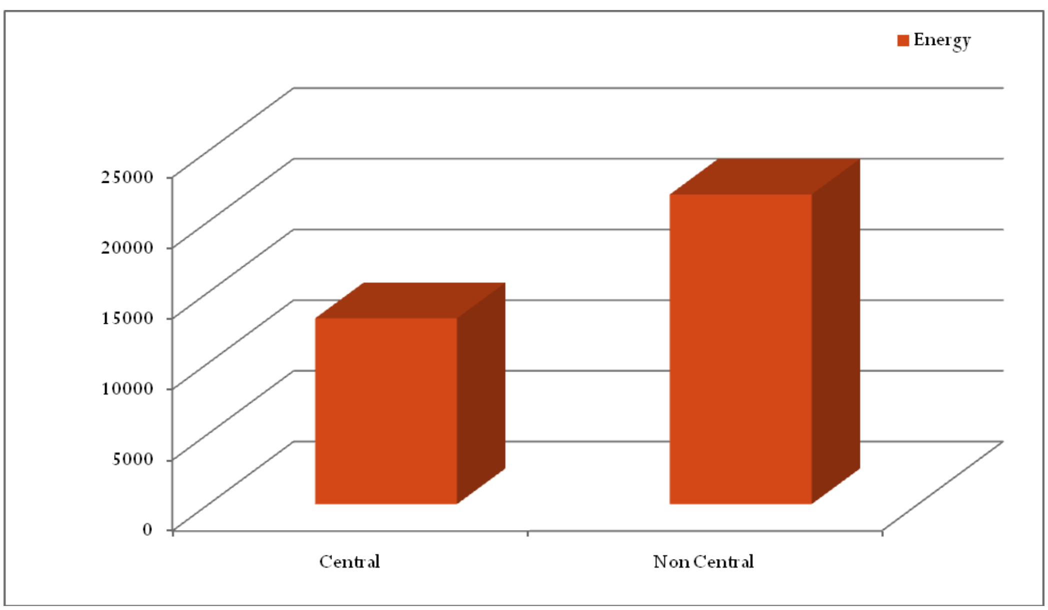

Figure 19: Energy consumption in centralized structure and non-centralized structure

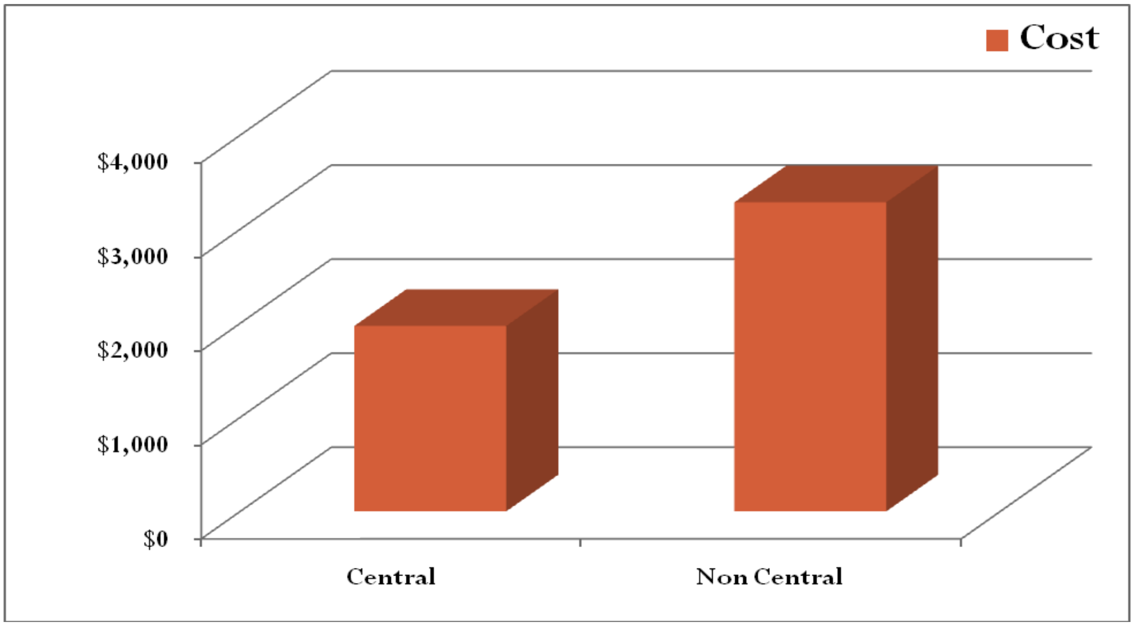

Figure 20: Cost comparison in centralized structure and non-centralized structure

CHAPTER 7

DISCUSSION

7.1 Future Work

This research, is a study of which algorithm could be more beneficial in existing centralized network infrastructures. Initially, focus was on the working of the centralized structure, and the algorithm was built accordingly. This algorithm is a preliminary step to understanding and studying the perceptions that could be present in the designing of the centralized structure. The proposed algorithm only had elements of the central working of the centralized structure concerning the intelligent on and off switching of the access points according to the network requirement. Further research could also be done on designing the algorithm based on convex hull approach more accurately and with a combination of all parameters that are critical for the network. While designing the algorithm, standard algorithms available in the field of computer networks could be studied in depth, and a finer standard could be developed. In this thesis, the access points were manually switched with the help of a controller. Further research could be in done in automatically switching access points according to the requirement. Also, signal strengths could be calculated by distancing the controller and access points and also creating a threshold level of signal strengths according to which the access points would get properly switched.

This study and proposal is an initial and first study in the area of an existing centralized structure. The research here can serve as an initial exploration point to the algorithm study in the area of centralized and non-centralized structures, because the working of a centralized structure is already known, but how the working initially starts and what kind of algorithms are employed are not known.

7.2 Conclusion

This research proposes an algorithm that might be running in a centralized structure or it can be implemented which has the potential to yield intelligence in switching access points on and off as per requirements, hence reducing cost and power consumption to a large extent. The same mechanism ensures proper coverage of signals and a definite allocation of bandwidth to users. The proposed algorithm was also run on a basic emulation testbed consisting of access points, and the respective parameters were calculated to define the benefits of a centralized structure in terms of power consumption and cost while assuring efficiency and quality of service to network users. While this research evaluated the effect of the proposed algorithm on a simple local network, it needs to be practically implemented and tested in a production-level environment. However, if implemented in the right direction, it could lead to greater artificial intelligence and 60 percent additional savings in power and cost.

REFERENCES

LIST OF REFERENCES

[1] Iannaccone, G., Jardosh, A., Papagiannaki, K., and Vinnakota, B., 2007, ―Towards an Energy-Star WLAN infrastructure,” *Mobile Computing Systems and Applications,* Hotmobile., Eighth IEEE Workshop., pp. 85-90 http://www.computer.org/portal/web/csdl/doi/10.1109/HOTMOBILE.2007.15 [cited 10 April 2009]

[2] Chen, J., Chan, G. H., He, J., and Liew, S., 2003, ―Mixed-mode WLAN: The integration of ad hoc mode with wireless LAN infrastructure,” *Global Telecommunications Conference, GLOBECOM*, Vol. 1, pp. 231-235. [cited 12 July 2009]

[3] Leith, D. J., and Clifford, P., 2006, ―A self-managed distributed channel selection algorithm for WLANs,” *Proceedings of 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Network,* pp. 1-9. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.119.2902 [cited 20 July 2009]

[4] Chen, Y., Chin, X., Yang, Q., and Yin, J., 2006, ―Power-Efficient Access-Point Selection for Indoor Location Estimation,” *IEEE Transactions and Data Engineering,* Vol. 18, No.7, pp. 877-888, DOI: 10.1109/TKDE.2006.112

[5] Smailagic, A., Siewiorek, P, D., and Small, J., 2001, ―Determining user location for context aware computing through the use of a wireless LAN infrastructure,” *IEEE Intelligent Systems,* Vol. 16, pp. 38-46 http://www.ices.cmu.edu/reports/040201.pdf

[6] Hills, A., and Johnson, D, B., 1996, ―A wireless data network infrastructure at Carnegie Mellon University,” *Personal Communications*, IEEE, Vol. 3, No.1, pp. 56-63.

[7] Zvanovec, S., Pechac, P., and Klepal, M., 2003, ―Wireless LAN Networks Design: Site Survey or Propagation Modeling,” Retrieved from: http://www.urel.feec.vutbr.cz/RADIOENG/fulltexts/2003/03_04_42_49.pdf [cited 10 August 2009]

[8] Charnsripinyo, C., Kabara, J., Prommak, C., and Tipper, D., 2002, ―Next Generation Wireless LAN System Design,” *Milcom Proceedings,* Vol. 1, pp. 473-477. http://netlab.cse.yzu.edu.tw/~bpmania/Paper/wlan-net-dgn.pdf [cited 11 September 2009]

[9] Computational Geometry Studies, University of Maryland

[10] Best Practices- Location Aware WLAN Design Considerations. Cisco Systems, http://www.cisco.com [cited 20 October 2009]

[11] Mishra, N., Chebrolu, K., Raman, B., and Pathak, A., 2006, “Wake-on-WLAN,” *Proceedings of the 15th International Conference on World Wide Web,* Edinburg, Scotland, pp. 761-769.

[12] Centralized Wireless Controller Vendors. Aruba, http://www.arubanetworks.com/ [cited 13 October 2009]

[13] Centralized Wireless Controller Vendors. Cisco Systems, http://www.cisco.com/ [cited 14 October 2009]

[14] Centralized Wireless Controller Vendors. Airespace, http://www.airespace.com/ [cited 14 October 2009]

[15] U.S. Energy Information Administration http://www.eia.doe.gov/cneaf/electricity/ [cited 13 October 2009]

[16] Hills, A., and Johnson, B, D., 1996, “A Wireless Data Network Infrastructure at Carnegie Mellon University,” *IEEE Pers. Communication,* Vol.3, pp. 56-63.

[17] “Wake-on-LAN,” http://www.datasynergy.co.uk/products/wakeman/pdfs/Wake-on-LANExplained.pdf [cited 9 September 2009]

[18] “Benefits of Centralization in Wireless LANs,” Cisco Systems. http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps6521/prod_white_paper0900aecd8040f7b2_ns337_Networking_Solutions_White_Paper.html [cited 10 August 2009]

[19] “Ethernet Power Study of Cisco and Competitive Products,” Cisco Systems. http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps7077/white_paper_c11-470808_ps5023_Products_White_Paper.html [cited 5 September 2009]

[20] “Cisco Catalyst 3560 Series Switches Data Sheet,” Cisco Systems http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps5528/product_data_sheet09186a00801f3d7d.html [cited 20 January 2009]

APPENDICES

APPENDIX A

COMPUTER PROGRAM FOR SERVER

```
/*
** server.c -- a stream socket server demo
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>

#define PORT "3490"  // the port users will be connecting to

#define BACKLOG 10       // how many pending connections queue will hold


void sigchld_handler(int s)
{
      while(waitpid(-1, NULL, WNOHANG) > 0);
}

// get sockaddr, IPv4 or IPv6:
void *get_in_addr(struct sockaddr *sa)
{
      if (sa->sa_family == AF_INET) {
            return &(((struct sockaddr_in*)sa)->sin_addr);
      }

      return &(((struct sockaddr_in6*)sa)->sin6_addr);
}

        char buf2[10] = "shut3";
        char buf3[10] = "shut2";

int main(void)
{
      int sockfd, new_fd;  // listen on sock_fd, new connection on new_fd
      struct addrinfo hints, *servinfo, *p;
      struct sockaddr_storage their_addr; // connector's address information
      socklen_t sin_size;
      struct sigaction sa;
      int yes=1;
      char s[INET6_ADDRSTRLEN];
        char r[100];
```

```
int rv;

  FILE * sure;
  sure = fopen("harthick.txt","r");


memset(&hints, 0, sizeof hints);
hints.ai_family = AF_UNSPEC;
hints.ai_socktype = SOCK_STREAM;
hints.ai_flags = AI_PASSIVE; // use my IP

if ((rv = getaddrinfo(NULL, PORT, &hints, &servinfo)) != 0) {
      fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
      return 1;
}

// loop through all the results and bind to the first we can
for(p = servinfo; p != NULL; p = p->ai_next) {
      if ((sockfd = socket(p->ai_family, p->ai_socktype,
                  p->ai_protocol)) == -1) {
            perror("server: socket");
            continue;
      }

      if (setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR, &yes,
                  sizeof(int)) == -1) {
            perror("setsockopt");
            exit(1);
      }

      if (bind(sockfd, p->ai_addr, p->ai_addrlen) == -1) {
            close(sockfd);
            perror("server: bind");
            continue;
      }

      break;
}

if (p == NULL)  {
      fprintf(stderr, "server: failed to bind\n");
      return 2;
}

freeaddrinfo(servinfo); // all done with this structure

if (listen(sockfd, BACKLOG) == -1) {
      perror("listen");
      exit(1);
}

sa.sa_handler = sigchld_handler; // reap all dead processes
sigemptyset(&sa.sa_mask);
sa.sa_flags = SA_RESTART;
if (sigaction(SIGCHLD, &sa, NULL) == -1) {
      perror("sigaction");
```

```
            exit(1);
        }

        printf("server: waiting for connections...\n");

        while(1) {  // main accept() loop
            sin_size = sizeof their_addr;
            new_fd = accept(sockfd, (struct sockaddr *)&their_addr,
&sin_size);
            if (new_fd == -1) {
                perror("accept");
                continue;
            }

            inet_ntop(their_addr.ss_family,
                get_in_addr((struct sockaddr *)&their_addr),
                s, sizeof s);
            printf("server: got connection from %s\n", s);



            if (!fork()) { // this is the child process
                //close(sockfd); // child doesn't need the listener
                    if(s[6]=='3'){
                    recv(new_fd,buf3, 10, 0);
                    //printf("%s",buf3);
                    //printf("%c",s[13]);
                        if(buf3[4]=='3'){
                        send(new_fd, "shut3", 10, 0);
                        }
                        else{}
                    }
                    if(s[6]=='2'){
                    recv(new_fd,buf2, 10, 0);
                    //printf("%s",buf2);
                    //printf("%c",s[13]);
                        if(buf3[4]=='2'){
                        send(new_fd, "shut2", 10, 0);
                        }
                        else{exit(0);}
                    }


                    perror("send");
                //close(new_fd);

                exit(0);
            }

            // close(new_fd);  // parent doesn't need this
        }

        return 0;
}
```

APPENDIX B

COMPUTER PROGRAM FOR CLIENT 1

Client 1:

```
/*
** client.c -- a stream socket client demo
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>

#include <arpa/inet.h>

#define PORT "3490" // the port client will be connecting to

#define MAXDATASIZE 100 // max number of bytes we can get at once

// get sockaddr, IPv4 or IPv6:
void *get_in_addr(struct sockaddr *sa)
{
      if (sa->sa_family == AF_INET) {
            return &(((struct sockaddr_in*)sa)->sin_addr);
      }

      return &(((struct sockaddr_in6*)sa)->sin6_addr);
}

int main(int argc, char *argv[])
{
      int sockfd, numbytes;
      char buf[MAXDATASIZE];
      struct addrinfo hints, *servinfo, *p;
      int rv;
      char s[INET6_ADDRSTRLEN];
        char shutdown[10];
        int a;

      if (argc != 2) {
          fprintf(stderr,"usage: client hostname\n");
          exit(1);
      }

      memset(&hints, 0, sizeof hints);
```

```
hints.ai_family = AF_UNSPEC;
hints.ai_socktype = SOCK_STREAM;

if ((rv = getaddrinfo(argv[1], PORT, &hints, &servinfo)) != 0) {
      fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
      return 1;
}

// loop through all the results and connect to the first we can
for(p = servinfo; p != NULL; p = p->ai_next) {
      if ((sockfd = socket(p->ai_family, p->ai_socktype,
                  p->ai_protocol)) == -1) {
            perror("client: socket");
            continue;
      }

      if (connect(sockfd, p->ai_addr, p->ai_addrlen) == -1) {
            close(sockfd);
            perror("client: connect");
            continue;
      }

      break;
}

if (p == NULL) {
      fprintf(stderr, "client: failed to connect\n");
      return 2;
}

inet_ntop(p->ai_family, get_in_addr((struct sockaddr *)p->ai_addr),
            s, sizeof s);
printf("client: connecting to %s\n", s);

freeaddrinfo(servinfo); // all done with this structure

  printf("To put AP1 in sleep mode please enter one \n");
  scanf("%d", &a);
  if (a == 1)
  {printf("befor send");
  send(sockfd,"shut3",10,0);
  }
  if (a == 0)
  {
  send(sockfd,"shut",10,0);
  }
  printf("befor comparison");
if ((numbytes = recv(sockfd, buf, MAXDATASIZE-1, 0)) == -1) {
    perror("recv");
    exit(1);
}


  //printf("value of a %d", a);

buf[numbytes] = '\0';
```

```
        //printf("client: received print '%s'\n",buf);
        //printf("befor if");

        if( buf[0] == 's' && buf[1] == 'h' && buf[2] == 'u' && buf[3] ==
't'&& buf[4] == '2')
        {
           system("sudo /etc/acpi/sleep.sh force");
        }
        //else { printf("hi");}


      close(sockfd);

      return 0;
}
```

APPENDIX C

COMPUTER PROGRAM FOR CLIENT 2

Client 2:

```
/*
** client.c -- a stream socket client demo
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>

#include <arpa/inet.h>

#define PORT "3490" // the port client will be connecting to

#define MAXDATASIZE 100 // max number of bytes we can get at once

// get sockaddr, IPv4 or IPv6:
void *get_in_addr(struct sockaddr *sa)
{
      if (sa->sa_family == AF_INET) {
            return &(((struct sockaddr_in*)sa)->sin_addr);
      }

      return &(((struct sockaddr_in6*)sa)->sin6_addr);
}

int main(int argc, char *argv[])
{
      int sockfd, numbytes;
      char buf[MAXDATASIZE];
      struct addrinfo hints, *servinfo, *p;
      int rv;
      char s[INET6_ADDRSTRLEN];
        char shutdown[10];
        int a;

      if (argc != 2) {
          fprintf(stderr,"usage: client hostname\n");
          exit(1);
      }

      memset(&hints, 0, sizeof hints);
      hints.ai_family = AF_UNSPEC;
```

```
hints.ai_socktype = SOCK_STREAM;

if ((rv = getaddrinfo(argv[1], PORT, &hints, &servinfo)) != 0) {
      fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
      return 1;
}

// loop through all the results and connect to the first we can
for(p = servinfo; p != NULL; p = p->ai_next) {
      if ((sockfd = socket(p->ai_family, p->ai_socktype,
                  p->ai_protocol)) == -1) {
            perror("client: socket");
            continue;
      }

      if (connect(sockfd, p->ai_addr, p->ai_addrlen) == -1) {
            close(sockfd);
            perror("client: connect");
            continue;
      }

      break;
}

if (p == NULL) {
      fprintf(stderr, "client: failed to connect\n");
      return 2;
}

inet_ntop(p->ai_family, get_in_addr((struct sockaddr *)p->ai_addr),
            s, sizeof s);
printf("client: connecting to %s\n", s);

freeaddrinfo(servinfo); // all done with this structure

  printf("To put AP2 in sleep mode press one \n");
  scanf("%d", &a);
  if (a == 1)
  {printf("befor send");
  send(sockfd,"shut3",10,0);
  }
  if (a == 0)
  {
  send(sockfd,"shut",10,0);
  }
  printf("befor comparison");
if ((numbytes = recv(sockfd, buf, MAXDATASIZE-1, 0)) == -1) {
    perror("recv");
    exit(1);
}


  //printf("value of a %d", a);

buf[numbytes] = '\0';
```

```
        //printf("client: received print '%s'\n",buf);
        //printf("befor if");

        if( buf[0] == 's' && buf[1] == 'h' && buf[2] == 'u' && buf[3] ==
't'&& buf[4] == '3')
        {
           system("sudo /etc/acpi/sleep.sh force");
        }
        //else { printf("hi");}


      close(sockfd);

      return 0;
}
```