Spring 2016

THE STUDY OF DOMOTICS FOR GREEN SERVER ROOM INFRASTRUCTURE

Eric Michael Hutchins
University of New Hampshire, Durham

Recommended Citation
Hutchins, Eric Michael, "THE STUDY OF DOMOTICS FOR GREEN SERVER ROOM INFRASTRUCTURE" (2016). *Master's Theses and Capstones*. 855.
https://scholars.unh.edu/thesis/855

THE STUDY OF DOMOTICS FOR GREEN SERVER ROOM INFRASTRUCTURE

BY

ERIC HUTCHINS
Business B.A., UNH Manchester, 2012

THESIS

Submitted to the University of New Hampshire
In Partial Fulfillment of
The Requirements for the Degree of

Master of Science
In
Information Technology

May, 2016

This thesis has been examined and approved in partial fulfillment of the requirements for the degree of Master of Science in Information Technology by:

Thesis Director, Dr. Michael Jonas, Assistant Professor of Computer Science

Dr. Karen Jin, Assistant Professor of Computer Science

Dr. Jeremiah Johnson, Assistant Professor of Mathematics

On May 9th, 2016

Original approval signatures are on file with the University of New Hampshire Graduate School.

DEDICATION

I dedicate this thesis to my mother, Maureen Hutchins, my father, Michael Hutchins, and my brother, Jason Hutchins, of whom I could not have gotten this far without.

ACKNOWLEDGEMENTS

The journey of writing a thesis was an incredible learning experience. It challenged me to work with a variety of different systems I was unfamiliar with and to get them all working in unison. An initially limited knowledge of Linux caused this to be more difficult than it had to be, but much was learned and documented throughout the process. The network issues, although simple in retrospect, were also valuable lessons learned when setting up the testing environment. There was also a significant amount of learning when it came to domotics, not to mention energy consumption and waste.

It is my hope that this work continues forward, and can perhaps even be expanded to the UNH-Durham campus. I would like to thank all of the faculty and staff of the University of New Hampshire at Manchester for their continued support and encouragement. I offer my sincere appreciation to my committee for the learning experiences they have provided. In particular I would like to thank Dr. Michael Jonas for playing a critical role in my development and learning which culminates in this thesis.

TABLE OF CONTENTS

ABSTRACT

THE STUDY OF DOMOTICS FOR GREEN SERVER ROOM INFRASTRUCTURE

By

Eric Hutchins

University of New Hampshire, May, 2016

Server rooms everywhere face a continuous problem: energy costs. These costs come from running the servers more often than necessary, including college settings where students' use of servers for projects can be sparse and lacks continuity. The University of New Hampshire at Manchester (UNH-M) has such a server room with these problems. domotics stood out from the rest when researching possible solutions. domotics is the study of applications of information technology to create intelligent home environments and can be used to conserve energy. In this work we thoroughly examine common domotics solutions to determine the best one to apply to a server room setting to save energy.

We demonstrate a solution that is both cost-conscious as well as simple to implement. This is done by developing a prototype tool based on the X10 protocol that allowed for easy and efficient power management for a series of servers. The result is a significant reduction in energy waste for the UNH-M server room over the course of a year with a relatively simple installation process. These estimated results are also compared to the overall energy analysis of the academic building it is housed in. This analysis focused on the monthly usage of energy, a survey of faculty habits, and finally the impact of the solar panels on the building.

CHAPTER 1

INTRODUCTION

As much as half of the energy needed to power servers is wasted due to poor operational practices and inefficient hardware, particularly in small server rooms and server closets. [1] In order to understand the solution, it is best to separate the issue into several distinct factors. The first is the most obvious: the continuously running servers add more to the electric bill even if they are not being utilized. Once students have finished their work the servers remain on. Often students work remotely and are unaware if other students are currently using the server. If they shut it down it could interrupt their work, or stop other students from working since there is no way to turn the servers back on.

The second and more overlooked cost is that air conditioning units (A/C) can roughly double the energy expenditure of a server room. In 2005, direct power consumption by servers was 2.6 billion kW. Including cooling costs, the overall power consumption rises to about 5 billion kW. [2] By mitigating the amount of time a server is on, we also directly lower the amount of time the A/C is in use. These first two factors are the costlier ones, though there is a third that we will be focusing on as well.

The third and less obvious factor to be considered is what is known as ‘Standby Power’, also known as ‘Vampire Power’. When devices are plugged in, even if they are powered down, they still draw electricity. It is estimated that 5-10% of electricity is wasted due to standby power in the United States. [3] A server room in standby may not be much initially; yet over the course of an entire year, or multiple years, the energy wasted will start to add up. This impact is further increased by the increasing number of servers and machines present.

Over the past forty years many strides have been made in the field of domotics. domotics is the study of applications of information technology to create intelligent home environments. Intelligent devices are capable of many things, but of particular significance they can be turned off and on remotely or even automatically depending on programming. This type of technology is typically focused on home environments for lamps and heaters; however, it is not restricted to those devices. It can also be put to use in server rooms for computers and other devices. Most importantly, domotics technology is meant to be a low-cost solution to energy management. When compared to enterprise solutions that can range in the tens of thousands of dollars, it was clear which option would be more reasonable to pursue.

The goal of this thesis and project is to study power usage at University of New Hampshire at Manchester (UNH-M), specifically focusing on a server room, and to create a solution that limits waste as much as possible. The main tool to be examined and implemented for this solution is domotics. By using this technology as an intermediary between power sources and their intended end points significant energy can be saved. The specific ways of doing so, reducing unnecessary up-time, requiring less A/C to cool systems, and reducing standby power, will be discussed in more detail later in this work. It is also important to note that the aim of this project is to make it easier for students and staff to access and manage the UNH-M servers regardless of their location. These efforts will lower the wasteful power consumption of UNH-M. Before looking into that it is important to first understand how the college operates today.

UNH-M consumes power in similar ways to every college and school in the nation. It is fortunate to have a robust computer department with a room full of servers that allow students to log in and conduct their studies. One assignment they commonly work on is a speech recognition application. This requires the students to run “trains”, which are essentially training algorithms that build speech recognition models to enable the system to recognize new speech. These trains can take many hours to complete. If a server that is needed is off, then the work cannot be done. Currently the solution to this problem requires someone to physically enter the server room, locate the correct server, and turn it on. There are rules in place for being in the building after hours, so any work that needs to be done after a certain time simply cannot be done if the server is off past that time. The current solution is to simply keep the servers running so that they remain available to students. This is not an environmentally friendly solution to the problem, nor is it an easy to manage system.

The rest of this thesis will discuss several different topics ending with a solution to the above problem. A history of different domotics, from some of the first created to the current popular versions, will be an essential baseline. A deeper discussion on how energy is consumed is necessary for a thesis on saving energy. Linked with it will be more details on how energy is commonly wasted in obvious and not so obvious ways framed from the point of view of a server room. The project portion of this thesis will focus on finding the best solution to energy waste and the problems unique to a college campus-based server room. Prior to that, domotics should be explained in more detail.

CHAPTER 2

HISTORY OF DOMOTICS

Domotics is not something new. Since the 1970’s people have been trying to control their homes in a variety of ways, with one of those intentions being to reduce their electric bills. Part of this work will concern itself with comparing and contrasting different domotics solutions in regards to their application in a server room environment. These differing technologies have a great many uses that will not be covered here. The chosen solution may not be the best solution for other tasks; however, it will be the most efficient, cost-effective, and reliable solution for the test environment. [4]

There are many different types of domotics available today. Some use existing wiring in houses to communicate, and others use modern wireless communication (Wi-Fi). The end result is the ability to control one’s home, or office, remotely and to utilize energy more efficiently. The list of domotics choices that will be evaluated for this paper are: X-10, UPB, Zig-Bee, Z-Wave, and WeMo. It should be noted here that X-10 has already been chosen as the most reasonable technology to use for a number of reasons, and will also be the benchmark that the others will be held against. The following sections will cover a brief history of each technology, its main functionality, advantages, and disadvantages. The project in which X-10 was used will be explained in more detail later on.

2.1 X-10 Protocol

“X-10 was developed in 1975 by Pico Electronics of Glenrothes, Scotland, in order to allow remote control of home devices and appliances. It was the first general purpose domotic network technology and remains the most widely available.” [5]

X-10 functions by sending signals over an existing powerline structure in a home or office environment. [6] This paper will not be going into complete detail as to how this protocol functions, however it will discuss enough details to allow for a proper comparison to other protocols and standards. These signals are short Radio Frequency (RF) bursts that hold information that existing modules in the system pick up. As seen in Figure 2.1, the transmissions are synchronized to be at or within 200 milliseconds of the zero crossing point. Modules plugged into electrical outlets are able to detect these bursts over the normal electrical noise, and take appropriate action.

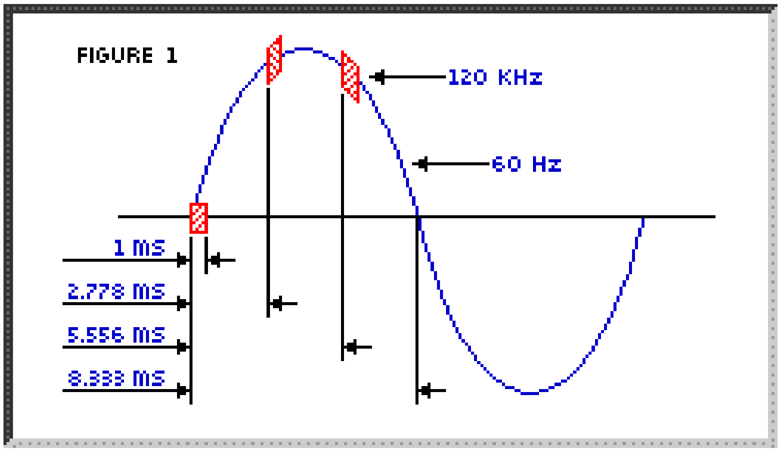
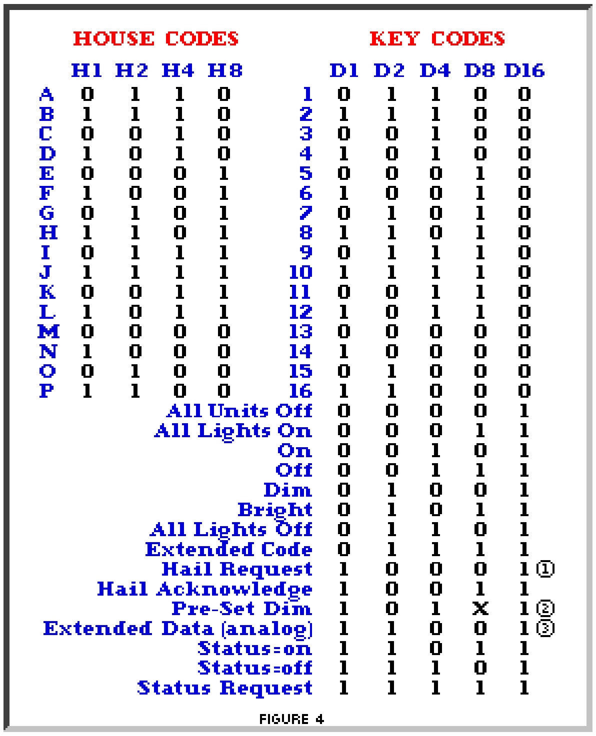


Figure 2.1 - Zero Crossing Points

The action that each module takes is not simply on or off, but is instead a combination of binary 1’s and 0’s. A ‘one’ is indicated by a 1 millisecond burst of 120kHz at the zero crossing point mentioned above. A ‘zero’ is indicated by the lack of the ‘one’ burst. Each set of bursts is repeated a total of three times to coincide with the three zero crossing points that occur within a three phase distribution system. The summary of all of this is, each time an X-10 device wants to send a signal it is actually sending the signal three times, and is sending a series of 1’s and 0’s.

Each signal sent is a four or five-bit code that correlates to a predetermined pattern. One portion of the message applies to what is called a ‘House Code’ which is made up of a number and a letter and designates the object that should receive the message. There are 16 letters and 16 numbers, allowing for a total of 256 addresses. The other part of the message is a command such as ‘Turn on’, ‘Turn off’, ‘Dim’, etc., as can be seen below.



	HOUSE CODES					KEY CODES					
	H1	H2	H4	H8		D1	D2	D4	D8	D16	
A	0	1	1	0	1	0	1	1	0	0	
B	1	1	1	0	2	1	1	1	0	0	
C	0	0	1	0	3	0	0	1	0	0	
D	1	0	1	0	4	1	0	1	0	0	
E	0	0	0	1	5	0	0	0	1	0	
F	1	0	0	1	6	1	0	0	1	0	
G	0	1	0	1	7	0	1	0	1	0	
H	1	1	0	1	8	1	1	0	1	0	
I	0	1	1	1	9	0	1	1	1	0	
J	1	1	1	1	10	1	1	1	1	0	
K	0	0	1	1	11	0	0	1	1	0	
L	1	0	1	1	12	1	0	1	1	0	
M	0	0	0	0	13	0	0	0	0	0	
N	1	0	0	0	14	1	0	0	0	0	
O	0	1	0	0	15	0	1	0	0	0	
P	1	1	0	0	16	1	1	0	0	0	
					All Units Off	0	0	0	0	1	
					All Lights On	0	0	0	1	1	
					On	0	0	1	0	1	
					Off	0	0	1	1	1	
					Dim	0	1	0	0	1	
					Bright	0	1	0	1	1	
					All Lights Off	0	1	1	0	1	
					Extended Code	0	1	1	1	1	
					Hail Request	1	0	0	0	1	①
					Hail Acknowledge	1	0	0	1	1	
					Pre-Set Dim	1	0	1	X	1	②
					Extended Data (analog)	1	1	0	0	1	③
					Status=on	1	1	0	1	1	
					Status=off	1	1	1	0	1	
					Status Request	1	1	1	1	1	

FIGURE 4

Figure 2.2 - X-10 Signal Codes

2.1.1 Advantages

There are several key advantages to using X-10 as opposed to other domotics tools and protocols. The first is that overall X-10 is far cheaper than competing systems, with prices hovering around $20 for wall modules and $30 for controllers. The alternatives to X-10 start typically around $40 and rise sharply depending on the technology. This can mean a significant difference in costs, if the number of units needed is high.

X-10 modules are easy to install and operate. If users only have a few modules it can be a simple task to get them working as intended. With a special remote it is possible to quickly turn off and on different devices around a home or office environment. However, doing more than that can quickly become complicated, especially if automation is required. The more complex a scenario a user tries to create with X-10 the harder it is to implement. This issue in particular will be discussed in more detail in the disadvantages section. [7]

2.1.2 Disadvantages

Despite the cost advantage and general ease of installation, X-10 has some disadvantages. The first major issue is that it is not secure. Anyone with a transmitter can send X-10 signals to activate any X-10 modules, as long as they send the right house code. There is nothing stopping outside attacks, nor even stopping accidental activations by a neighbor who happens to use the same settings. The attacker will need to use the right codes in order to do this, but that is merely an issue of trial and error. A determined individual could eventually find the right information, if he/she was not able to locate the physical devices and see what their settings were.

Depending on a particular electrical infrastructure there can be a significant amount of noise on the line. [8] The amount of appliances that can generate noise or false signals is rather large. Common appliances include: electric razors, hair dryers, vacuum cleaners, PC monitors, compact fluorescent lamps and halogen lamps, laser printers, baby monitors, and more. X-10 does not have any safeguards against this, which means not every environment is suitable for X-10 deployment.

Users may find X-10’s range to cause problems when setting up the technology. The longer the distance an X-10 signal has to travel the weaker it becomes. This is especially true if it has to travel to a breaker panel at the other end of a home or office, cross over to the opposite phase, and then go all the way back to reach a device that is plugged into the phase opposite the transmitter. The larger the building and the more X-10 devices are involved, the worse this issue will be. Researchers have found that homes over 2500 square feet often have these problems, whereas those under 2000 square feet do not. There are many factors to consider when attempting to use X-10. It is necessary to evaluate the surroundings thoroughly to see if anything will interfere with the signal.

The wiring in a building can become more or less detrimental to X-10 depending on its size and complexity. A small home may not have many issues; however, a large commercial building can. It is possible for one outlet to be on a completely different circuit than one across the room, which can cause the pathway for the signal to be too far or be lost in noise by the time it loops around and gets over to the destination outlet. Even small office rooms can be problematic, as experienced personally by the author when testing devices. This is a rather large disadvantage due to the potential problems endured during setup that other technologies do not have to deal with.

This domotics protocol is also simple. This can be seen as an advantage in some ways, but overall it is a disadvantage against competitors. Newer technologies feature better UI, smart phone integration, etc. X-10 is an old technology that has not changed much and is rather bare-bones as far as features and compatibilities. For low-tech individuals X-10 can seem daunting and less friendly, even if it may still fulfill all the functions that they require. Other alternatives may also simply feel user-friendly, with newer and more interesting designs, compared to the old looking style that most X-10 units still have. For those more technically skilled who understands the problem they need to solve X-10 may be the better solution.

An issue relating to the simplicity of the protocol is that it is a one way only system. This means that when a user issues a command, that command is sent along the wiring of the building and hopefully reaches its destination. Regardless of whether or not it reaches that destination, the user will not know unless they are looking for a reaction or watching the X-10 device. Every other domotics technology in this paper has the ability for two-way communications. This has the potential to cause issues when trying to remotely trigger devices, especially if the user is nowhere near the location. All they can do is look for some kind of reaction if they have another device monitoring the situation.

Finally, X-10 does not have a large network capacity. In total, under normal circumstances, X-10 can only support up to 256 units at a time. All of the alternatives that will be discussed later in this paper have significantly more support if not theoretical infinite support depending on the technology. This means that if a user wants to create a large, complex automation system X-10 will likely not be the answer. That said, 256 is not exactly a small number, and could fulfill most small to mid-sized projects. If a user was going to do something involving thousands of modules, they may want to choose a newer technology.

Despite the variety of problems and challenges, for small applications and projects X-10 can work just as well as any other technology. For medium to large sized projects it would be recommended to choose another technology. However, for learning about domotics and testing out small devices, X-10 can be quite useful. X-10 is an old technology, and the basis for many others, yet it is still rather popular for home automation. As long as a user understands the hurdles, X-10 can be a significantly cheaper and yet still fully functional tool to fit their needs.

2.2 Universal Powerline Bus (UPB)

Universal Powerline Bus (UPB) was originally developed in 1998 by PCS Powerline Systems, based on the original X-10 standard. This means that UPB uses power lines, as the name suggests, in order to transmit signals throughout an area in a similar manner to X-10. [9] X-10 uses a fixed carrier frequency to transmit its data, as discussed above. UPB on the other hand, uses Pulse Position Modulation (PPM), that utilizes timed pulses instead of a carrier frequency. [10] The figure below depicts the timed pulses in relation to the electrical cycles and the zero crossing point. Of note is how all of the pulses occur at the very end in what is called the ‘UPB Frame’, whereas with X-10 the transmissions are spread throughout the half-cycle. [11]

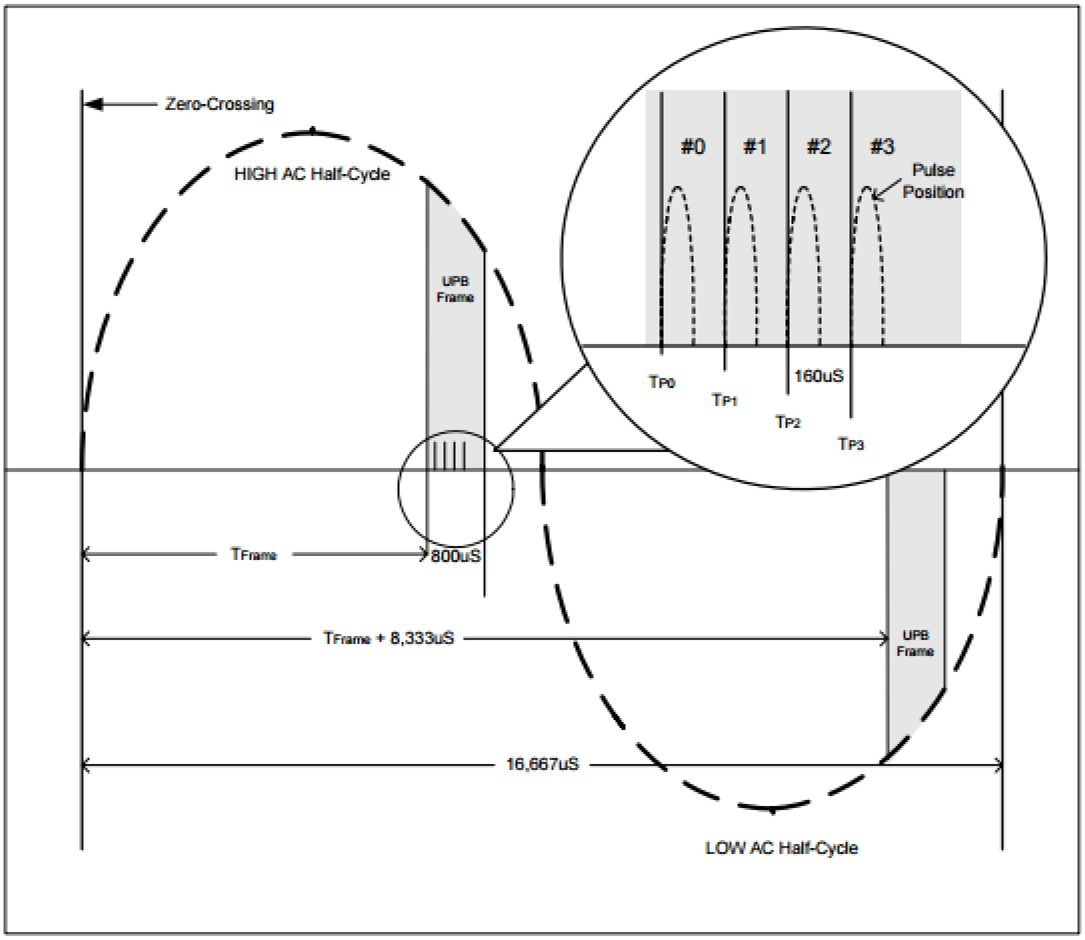


Figure 2.3 - UPB Frame in Half Cycle

2.2.1 UPB versus X-10

Overall, UPB is better than X-10. By not relying on the carrier frequency, as X-10 does, UPB has fewer issues to deal with. UPB is not as susceptible to attenuation or noise due to the timed nature of PPM. To further widen the gap, X-10 uses at maximum 4 volts, whereas UPB uses up to 40 volts. This increase in voltage signifies a stronger, more powerful signal that travels farther and faster with less attenuation. It is possible to send signals just over a mile with a favorable environment. One of the disadvantages of

X-10 was getting the signal to reach the intended destination, which makes this rather important for UPB to overcome.

The other main advantage is a significantly larger address range. As mentioned before X-10 can typically only have 256 objects. UPB is capable of hosting 250 houses and 250 units, coming to a total of 64,000 addresses. Not only does this allow users to have a larger amount of units, as you can have multiple ‘houses’ in a single area, but you also have more room to avoid neighbors if they are also using UPB. This allows UPB to be used for much larger projects than X-10.

Another significant advantage for UPB is that the hardware and software used allows for two-way communication. [12] This means that it is possible to send status requests to nodes in order to see if they are currently on or off. Allowing for two-way communication helps create a friendly user experience, especially for setup purposes.

The main disadvantage UBP has when compared to X-10 is in price. X-10 averages around $20 per device, whereas UPB devices are typically around $80 per device. This gap can increase even further depending on what is ordered as there are more complex modules in UPB then X-10. Unless the project in question has a sizable budget, it may not make financial sense to use UPB over X-10.

One final note on UPB is that it is not a secure technology, in the same way that X-10 is not a secure technology. [13] The only minor advantage is that UPB can accommodate more houses and units, which makes it harder to guess what combination of settings someone is using. However, that is hardly safe or secure, and so neither technology should be used if security is at all important to the user.

2.3 ZigBee

ZigBee will be the first technology examined that no longer functions via the existing power lines within a home or office. Instead ZigBee operates on a standard 2.4GHz band, which also happens to be the common Wi-Fi band in households and offices. [14] Each module is capable of extending signals similar to repeaters, allowing for a larger area to be covered without losing signal quality. Overall this increases reliability in comparison to the power line structure, and there are other added advantages to this technology as well that will be examined in more detail.

Each ZigBee module acts as a repeater. Specifically, each device in a network is able to repeat a signal up to roughly 70 meters or 230 feet. [15] Ranges far greater can be accomplished by using the nodes as a mesh network to communicate throughout an area. Large houses or offices are able to benefit greatly from this increased range over X-10 and UPB with far less susceptibility to attenuation. The standard restrictions and issues of the 2.4GHz band still apply. For example, users need to be conscious of solid or metal objects blocking or degrading the signal, as well as if there are too many devices using the 2.4GHz band in a small area. The 2.4GHz band also allows for more data to be passed back and forth.

The maximum data rate for using a 2.4GHz band is 250 kbps. For contrast, X-10 is only capable of using 20 bits per second. UPB is only capable of 240 bits, significantly faster yet still only able to perform basic tasks. [16] The slow transmission rates of X-10 and UPB limit the capabilities of what they can accomplish. For instance, ZigBee is capable of having constant two-way communication with its modules. This allows it to send out signals and receive confirmations, as well as sending out multiple complicated requests that will move throughout an entire network at a significantly faster rate.

While X-10's primary abilities are simple, ZigBee is capable of much more. With a higher speed and larger data packet size, ZigBee is able to have more complicated logic implemented into the system. One such example is having a temperature sensor trigger an event that causes the heat to turn on or off, or a time of day to change the thermostat automatically. This programmable logic allows for advanced configuration of home automation that X-10 is not capable of without outside assistance.

In terms of potential nodes in an area, ZigBee is closer to UPB than X-10. ZigBee is theoretically capable of having up to 65,000 assigned nodes, just over UPB's limit of 64,000. This puts ZigBee above X-10 for larger networking needs, just as it does for UPB. However, networks for ZigBee have only been tested up to 1,000 nodes, a fraction of its theoretical limit. [17] This is not to say that it cannot support more than 1,000 nodes, but instead that there is simply no evidence of a network going beyond that limit. As both UPB and ZigBee are essentially at the same amount of nodes it would not make sense to rate one above the other except in very specific circumstances. Even still, large offices/buildings completely wired up with the technology may encounter issues. ZigBee may offer many advantages over its predecessors.

The last way ZigBee differs from X-10 is in its security. This paper will not be going too far in depth in regards to the security measures of ZigBee, or others. However, there are a few important things to note about ZigBee. ZigBee's security is built upon the 802.15.4 protocol, which uses AES with a 128-bit encryption key. [18] This algorithm is used to encrypt the information, but also used to validate the data being sent. This means that, at its core, ZigBee should be secure. Unfortunately, there is significant evidence that it is not.

The issue with ZigBee’s security is that it is only truly useful for stopping accidental signals. Those who are intent on malicious behavior have not had much of an issue getting into the system. Attackers that are able to physically interact with a device can easily obtain the security key from it. Keys are typically written on all of the devices in a ZigBee network, which allows an attacker to obtain the key as it moves from flash memory to RAM during power up. Once an attacker has the key they then have full access to the network. Even without physical access it is not difficult to mimic a network node and receive data packets that can be decrypted and analyzed to obtain the key. Based on this information ZigBee technically has security, but in reality it essentially doesn’t. This is only a brief overview of the security vulnerabilities of ZigBee, as there are more cases and more ways to break its security. ZigBee’s security is only helpful for avoiding accidental signals. The evidence for ZigBee being secure from malicious attacks is sparse. [19]

Another problem with ZigBee is that its devices are manufactured by multiple different companies. Normally this would not be an issue. However, they are all able to interpret the 802.15.4 protocol in their own way and do. [20] This allows manufacturers to create their own lines of ZigBee products with slightly different internal designs. The problem is that if a consumer purchases two ZigBee products, one from manufacturer A and one from manufacturer B, there is a strong chance they will not be able to work in the same network together. This can be frustrating and confusing to consumers. In contrast, all X-10 products work with each other and all UPB products work with each other, making them much simpler and easier to implement.

Surprisingly, compared to the other domotics listed in this document, ZigBee was the hardest to find general pricing for. As the protocol is open for anyone to use and develop for, there are many different manufacturers. Many of these manufacturers, such as Develco [21], require customers to contact them for pricing for even the smallest units they make. Overall it appears that ZigBee costs on average three times more than X-10. There are nuances to each manufacturer's products that also require a consumer to do further research to compare options, as they are almost guaranteed not to function the same.

Based on price, complexity, and consumer confusion, ZigBee is not an ideal choice for small or first time projects. It is hard for most to justify such a high price when rivaling technologies are either cheaper or easier to implement. The one benefit that could make a consumer want to purchase ZigBee is the use of Wi-Fi and all of the advantages that come with it. Even with that the chance of ZigBee products not working together is another black mark against the technology. This forces a consumer to rely on a single manufacturer that may or may not offer all of what they need, and then be dependent on them from there on. ZigBee does not seem to be the right choice for most applications, as there are many better choices that will be examined below.

2.4 Z-Wave

Z-Wave is a domotics technology that was created and designed based on ZigBee. [22] For this section Z-Wave will be compared with ZigBee over X-10, as most of the same key differences that exist between ZigBee and X-10 also exist between Z-

Wave and X-10. One of the main differentiators between these two wireless technologies is how they transmit their data.

This domotics technology also creates less of a strain on modern homes and offices with its communications. Instead of operating on the 2.4GHz band, of which many common devices already do, it operates at 908.42MHz. [23] The fact that Z-Wave operates on a different band than typical devices means that there will be less interference. If there are too many devices in one area transmitting on the same band, it can cause issues and a loss of information. The drawback of using this frequency is a significantly lower range.

The range of Z-Wave is significant according to their official website. Disregarding obstacles and assuming fair conditions, the range between Z-Wave products is approximately 120 feet or 40 meters. [24] When the conditions are poor or there are obstacles between devices that cause the range to be reduced, Z-Wave is capable of using mesh networking. This mesh networking allows signals to ‘hop’ between Z-Wave products to extend the range of the network and allow the signal to arrive at its intended destination. Z-Wave supports at maximum 4 hops, which can ultimately extend the range in fair conditions up to 600 feet or 200 meters.

While Z-Wave is capable of reaching up to 200 meters, ZigBee’s technology does not have any official limit to the amount of ‘hops’ that can be achieved. ZigBee’s 2.4GHz band is capable of reaching up to 100 meters per hop, allowing for a much farther reaching network. For the sake of argument, if ZigBee is also assumed to be locked at four hops it would have an upper limit of about 400 meters or roughly 1312 feet. Realistically both reach a significant distance that should satisfy most small projects. Unfortunately, having a short range is not the only network related issue that Z-Wave has.

Z-Wave has been reported to have node limit issues within its networks. The technology is supposed to be able to support up to 232 nodes in any given instance. There have been user reports that networks start to see problems when it grows to a size larger than 30-40 nodes. This indicates that Z-Wave may have an unstable network structure, or that its devices are simply not communicating properly once the network reaches a certain amount of devices. 30-40 nodes do not seem to be that many, which makes Z-Wave fall far short of ZigBee in this area.

While ZigBee devices may not always speak with one another, Z-Wave has a much better reputation. ZigBee has the current problem of being unable to interoperate with all ZigBee devices which can be frustrating for consumers. Z-Wave does not have that problem according to Michael Wolf, the founder and chief analyst of NextMarket Insights, which is a strategy and research firm in Edmonds, Washington. Z-Wave has "…better interoperability, historically, than ZigBee. All Z-Wave devices generally, without exception, work with other Z-Wave devices." [25] From the perspective of interoperability Z-Wave is clearly superior.

The last advantage Z-Wave has is in cost. A typical wall module for ZigBee costs around $60 as opposed to Z-Wave which costs around $40. Due to wall modules being a major factor in price because of the quantity a user will likely need, Z-Wave is the winner on this metric. The prices on other pieces such as remotes or wall switches can be double to quadruple the price of Z-Wave products. This difference in price can be prohibitive to a tight budget.

Much like ZigBee, Z-Wave offers security that is not actually secure. It is fully possible to use packet sniffing when initializing devices and inject packets. In this way it is simple to obtain the encryption key and have full access to the network. Some devices do not even use the encryption correctly as stated here, “The one device they found that was using it, implemented the encryption incorrectly - the key exchange was done in the clear so an attacker could intercept the keys and decrypt all of the communication.” [26] This technology is used all over the world from hotels to cruise ships, yet it is as insecure as having no security at all. It is surprising that such an insecure technology could be so widely implemented, and that these issues have not been addressed and dealt with. Regardless of this, it means that no domotics technology mentioned so far has been secure.

It is unclear which of the two more modern domotics technologies are better; ZigBee or Z-Wave. ZigBee has significant range and far less network related issues. Z-Wave has no interoperability problems with its products and a minor advantage in pricing. Both technologies fail on network security. It is up to the consumer to weigh these options and determine which technology may offer them better use based on what they are trying to accomplish.

2.5 WeMo

WeMo is designed and manufactured solely by the company Belkin. [27] Both Z-Wave and WeMo have official smartphone apps, however WeMo focuses further on the user experience by integrating WeMo with devices to make them ‘smart’. This smart integration effort can be seen in coffee makers and crock pots, among a small list of other common appliances. The technology also supports voice control via integration with an Amazon Echo, which shows their interest in newer technologies and methods of controlling home automation.

According to the manufacturer, a WeMo network is capable of supporting as many devices in a network as your network can support. [28] This is rather vague terminology, yet it does make sense. By using Wi-Fi and assuming IPv4 addressing the only limit on how many devices can communicate in one area is the point in which interference between devices prevents reliable communication. It seems that the setup of the network environment is more restrictive than the technology itself, which puts WeMo on the high end of the domotics spectrum for network limits.

When discussing the communication between devices, it raises the question of how well WeMo products interoperate. Fortunately, this is not an issue as Belkin is the sole provider of WeMo products. This means there is no chance of incompatibility issues that one would face with other domotics choices such as ZigBee. Being completely in charge allows them to send out updates through their apps, which also can be pushed onto WeMo devices to update their software and firmware. [29]

WeMo comes the closest so far when it comes to the issue of security. Prior to January 2014 there were a number of serious vulnerabilities affecting WeMo. One such vulnerability involved XML injection attacks. Fortunately, Belkin did eventually issue firmware updates that added SSL encryption, added password protection, and other measures to fix all known issues. Despite all vulnerabilities being fixed, there is still a significant oversight in the way their software works.

According to one reviewer of the technology, “Our only complaint remains the lack of privacy, such that anyone on your home Wi-Fi network has access to those local devices. They need to add some sort of user name & password security layer to their hardware.” [30] Essentially it means that if a user has access to the Wi-Fi network that WeMo is on, and downloads the app, he or she is then able to see and change all settings as if they were the administrator. On one hand the network has not been known since 2014 to have security vulnerabilities, but on the other their design makes it so an attacker just needs to obtain a Wi-Fi password. For all intents and purposes this is only marginally better than other technologies until Belkin updates their software with a simple login name and password system.

WeMo offers relatively simple setup and installation of its devices. [31] It is actually closest to X-10 in this regard, as Belkin sells standalone modules that plug into wall sockets instead of having to replace the wall socket all together. These devices are then set up and configured via the wireless app. This approach also means that users who are not allowed to perform any kind of home customization, such as apartment renters, will be able to take advantage of this technology. Z-Wave also offers standalone modules, but much like ZigBee they seem to focus on replacing wall outlets and other components rather than focusing on plug and play setup.

WeMo is an attractive option for domotics users. The ease of use for setup, configuration, and daily use at home or remotely offers more for consumers than previous examples. Financially, WeMo is somewhere in the middle as is not as cheap as X-10 but not as expensive as ZigBee. It also offers a strong network and an extended appliance user experience that others do not. Its security may not be ideal, though no other choice so far has any substantial security either. The plug-in modules are easily one of its most significant advantages. For non-technical users, or users who want a quick and easy experience, this could be the right choice.

2.6 Addressable Power Strip

When researching various domotics it was surprising to see that there was a missing product in each technology examined. This was discovered when attempting to set up multiple X-10 devices on a single power strip as it was difficult to get the devices to fit together without bumping or blocking one another. Some domotics such as Z-Wave offer a cord instead of a block, but the crowded situation was still the same. Every power outlet needed its own module, creating a rather large cluster of devices that seemed completely unnecessary. The ideal solution would be to have a device that acts similar to a power strip.

Larger power strips with wide gaps between outlets came to mind first as a potential solution. After all it was necessary to ensure enough space for each module, or else lose one or two outlets for each device plugged in. However, what if the power strip didn’t need any domotics modules, and was instead a series of switches by itself? It would be far more space and equipment efficient, requiring far less materials in the end. Individual modules are fine if there is only one device in that area that needs to be turned off remotely, but that appears to be rarely the case.

The ideal smart power strip would have all of the following features. The most important feature would be that each outlet could be controlled individually. Perhaps it could be manual as a secondary function, but the most important aspect would be to do so remotely. This ties in with the ability to access, read, and modify the states of each outlet via a smartphone or a computer. Being able to access it anywhere in the network would be a good start, but the true goal would be to access it from anywhere a user had an internet connection. Other metrics such as how much power was currently being used, or even the ability to constantly log that information into a spreadsheet for download and later review would make it even more appealing.

All of the research done on domotics came up with nothing on a device of this type. There are models that do one or two of the features listed above, but not all of them causing it to be next to useless for the problem at hand. Some smart strips are helpful against standby power drain, switching off the outlet based on usage and other metrics. Others could be remotely turned off, yet it would turn off the entire power strip and not individual outlets. There was not one smart power strip that had everything necessary to fill this gap, and none of the ones on the market appeared to have any kind of wireless or wired connectivity. Yet outside the world of domotics there was an answer, though it was found rather late into the period of research that this paper was done.

Devices called Addressable Power Strips have the core concepts that are needed to make a server room run efficiently. Taking one such model found on Amazon, the “NP-05B IP Managed Remote Power Switch/Strip “, it has individually switchable AC outlets and can be accessed remotely via web, telnet, or RS-232 port. This offers the main support for any small server room. Typically, these devices cost several hundred dollars or more. Despite the usefulness of the product there are definitely features lacking that server admins would want to know about.

In 2013, there was almost a device that had everything a person could want out of their addressable power strip. There is a Kickstarter project, that was successful, by the name of “Smart Power Strip”. [32] It had everything an addressable switch had, including smartphone support that allowed users to understand what was off or on at any given time, and remotely manage their home in a user friendly way. The only thing it could have benefited from would be gathering information and providing useful metrics such as energy saved, current energy usage, and other interesting statistics on general usage by individual outlet. Still, this would have been the near perfect product that could easily have been improved upon over time.

Upon researching the rather generic name of the device there was nothing on any online store matching what was offered on the Kickstarter. The only website linked from the campaign page was barely functional, with many links broken or leading to missing content. There are also multiple negative comments about the project claiming to have never received the device. Although this is only speculation, this could indicate that there were manufacturing problems as it certainly was not an issue of interest or support based on the success of the Kickstarter.

Despite its failure, the Kickstarter for an inexpensive “Smart Power Strip” is exactly what is needed and the success of its backing shows that people are interested in such a device. This means that there is room in the market to create another such device, and also room to do better than predecessors. Currently there is no public information on why exactly this product never worked out, yet from the view of an outsider there should not have been a reason why it failed. Having done extensive research on the topic, the author believes that there is a definite gap in the marketplace that no other product has fulfilled. This represents an opportunity for anyone with the resources and knowledge to take advantage of this market niche.

CHAPTER 3

ENERGY CONSUMPTION

Going forward it is important to set a baseline for how this paper will be discussing energy. Due to energy bills referring to all energy consumption as kWh (Kilowatt-Hour), all of the energy consumption referenced will be converted into kWh regardless of its original designation. This will allow for an even comparison field to better see the differences between appliances and devices. A significant aspect to start with is how that conversion works.

Common appliances and devices will be listed in watts, and will need to be converted to kWh. There are several things to keep in mind when going through with this process. The first step is determining how many watt hours a day each device is using. One way of doing this is to simply find the wattage label and multiply it by the number of hours it is used on a daily basis. For the purposes of this paper each device will be monitored with a device as well, to ensure accuracy. This step produces the watt-hours per day for that device, but not kWh.

The next step is to convert the watt-hours by day into kWh. This step is easier as it is simply dividing the watt-hours by 1000. [33] If XYZ device has a labeled wattage of 100 and is being used 3 hours a day, then its watt-hours are 300. Converting it to kWh produces the result of .03 kWh every day. This is the process that everything will go through, when applicable, in order to keep things simple for the remainder of this project. Without context however, .03 kWh means little. It is important to see how wattage is compared between devices.

The following is a list of common devices in households with their respective wattage ranges:

Common Device Wattage	
Coffee maker	900-1200 watts
Microwave	750-1100 watts
Toaster	800-1400 watts
Dishwasher	1200-2400 watts
Washer	350-500 watts
Dryer	1800-5000 watts
Iron	100-1800 watts
Space heater (40gal)	4500-5500 watts
Hair dryer	1200-1875 watts
Laptop	50 watts
Computer monitor	150 watts
Computer tower	120 watts
Television 19"-36"	65-133 watts
Television 53"-61"	170 watts

Table 3.1 – Common Device Wattage

As can be seen above, some appliances consume far more energy than others. What this table does not specify is how long each device is typically used each day. Even something with a large wattage consumption may not actually use that much if it is only used infrequently or for very brief periods of time. For example, assuming the Hair Dryer above uses the highest range of 1875 watts and is used for 7 hours on average in a week, then the result is 13 kWh. Commonly, refrigerators run consume between 300-780 watts. Assuming the lowest amount of 300, multiplied by 24 hours for 7 days, results in 50.4 kWh. This example shows that even if an appliance has a large watt consumption, it may not be as significant as it seems. More importantly, it is the constant running of lower wattage devices that can consume the most and by extension waste the most.

The next step in understanding what these numbers mean is to determine the price per kWh. Electricity costs can vary significantly in different cities in the same state. Location is not the only aspect that determines price. Whether the location in question is considered Commercial, Residential, or Industrial has a significant impact on the price. Industrial typically has the lowest cost, then Commercial, followed lastly by Residential. Note that individual locations can have differing prices still, and it is best to look at the electric bill for that building to be sure. The following is a table summarizing the differences in price between them within Manchester, New Hampshire. Note that this information may not be up to date as prices can fluctuate:

New Hampshire (Manchester) Electricity Prices per kWh		
Commercial	**Residential**	**Industrial**
9.56 cents	16.29 cents	6.11 cents

Table 3.2 - Manchester Electricity Prices [34]

Using the previous examples, this would mean that the hair dryer would cost approximately $2.12 and the refrigerator would cost approximately $8.21 per week. It may not seem to be that much, yet when the numbers are extrapolated to an entire year the hair drier costs roughly $101.76 and $394.08 respectively (Multiplied by 4 weeks, and 12 months of use). It is simple to see how over even longer periods of time the costs can become quite high. However, these common appliances mentioned, and especially the refrigerator, can be necessities that cannot be turned off for any period of time. Stepping away from the costs of waste, there is another factor to consider.

Beyond the financial impact there is also an environmental one. Energy has to be generated somehow, and although clean energy is making progress most places are still heavily reliant on traditional methods. According to the Environmental and Energy Study Institute (EESI), fossil fuels continue to be the primary energy source for the world. These fossil fuels, which include coal, oil, and natural gas, are burned to generate electricity and in doing so are responsible for 79% of U.S. greenhouse gas emissions in 2010. [35] Being able to reduce the amount of energy wasted, and therefore less energy needing to be generated from the burning of fossil fuels, can only create positive effects on the environment.

3.1 Ways to Waste Energy

In order to cut down on these negative environmental and financial impacts it is important to cut down on waste, and this means achieving a better understanding of how energy is being wasted. The following three sections provide further details on the major ways energy can be wasted.

3.1.1 Poor Energy Practices

Equipment running when it doesn’t need to be is a large source of waste. This is not the same as standby power, as will be discussed below, but instead simply a device turned on that isn’t currently doing anything useful. This could be anything from a television left on while going to the kitchen to get something to eat, to leaving a computer on all day while not at home. However, these are minor issues compared to what data centers are responsible for.

A report on data center energy efficiency from the Natural Resources Defense Council (NRDC) demonstrates a significant amount of wasted energy. [36] The report, compiled based on numbers for 2013, showed that data centers were using approximately 91 billion kWh, and by 2020 would use 139 billion kWh worth of power. The problem is not that these data centers consume massive amounts of energy, but that large portions of that energy could be saved. According to that report it would be possible to achieve up to an 80% reduction in energy usage were centers to use all best practices. That 80% offers almost $8 billion in savings for businesses nationwide.

Unfortunately, it is not always simple to make the kind of changes that would be required. To step into this problem briefly, there are many stakeholders involved that are not interested despite the potential for savings of not only energy but money as well. IT managers are concerned about potential downtime due to any implementation process. This perception of risk has slowed down efforts towards energy efficiency. This problem is the main focus of the project portion of this thesis, as will be discussed later on. Poorly managing energy consumption is the most obvious issue concerning energy waste, but it is not the only one.

3.1.2 Heat Generation

Server rooms and data centers consume a significant amount of electricity. As mentioned previously, in 2013 it is estimated that they consumed around 91 billion kWh of electricity, which is roughly equivalent to 34 coal-fired power plants. [37] Part of this energy is used by the machines in order to complete operations and produce outputs one would expect. The other part of this energy is used to mitigate the consumption of energy in the first place. Server rooms and data centers generate a significant amount of heat in return for their work.

It is estimated that between 30% and 50% of energy costs in data centers come from cooling costs. [38] The more advanced the server room, and the more efficient, the smaller this cost becomes. This means an inefficient setup could cost as much in cooling costs as it does for running the servers to begin with. It also shows that with the right planning cooling costs can be quite low. Unfortunately, that is a topic so large that it would require its own separate paper to go through. For now, it is best to understand that cooling costs are significant and cannot be ignored. It also follows that the less energy equipment is using, regardless of cooling efficiency, the less energy for cooling will be required.

3.1.3 Standby Power

As mentioned in the Introduction, Standby Power, also known as ‘Vampire Power’ or ‘Energy Vampire’, is a small but important aspect of power management to consider. According to the Lawrence Berkeley National Laboratory standby power can account for 5-10% of residential electricity use. Essentially every device that is constantly plugged into an outlet is using some amount of energy. To first clarify, when discussing constantly powered appliances such as refrigerators or other devices that are always running for a reason, such as to keep food from spoiling, are not what are being referenced. Standby power is only about devices that are plugged in yet are in their ‘off’ state. Devices such as televisions, computers, anything with an external power supply or that has an LED can be using standby power.

Dealing with standby power can be challenging, especially because it does not actually account for much waste. Constantly unplugging and plugging in devices can be hazardous due to the chance of electrocution due to frayed wires or plugs. Even using power strips, though they do offer more devices to be turned off at once, can be difficult if it is behind furniture or an entertainment system. Finally, it can simply be bothersome to the user to constantly shut off and turn on the power source for their appliances. Users typically want the easiest route, and if the amount of effort they put forth does not feel significant enough, due to the small amount of energy actually saved, they will be less likely to go through with it. There is a way to solve all of these concerns in a simple, elegant way, but first it would be better to discuss a real world scenario and the main focus of this thesis and project.

3.2 UNH Energy Analysis Introduction

There are multiple computer labs, classrooms, a cafeteria, and offices, that all consume electricity within the UNH-M building. To understand how much energy is being consumed by the computer department’s server room, it is important to compare those numbers with that of the entire building. There is also a set of solar panels on the building that provides supplemental energy that is worth taking into consideration. The following sections are broken out into how much energy the building consumes on a monthly basis, an anonymous survey conducted to get an idea of how teachers and office workers deal with power management, and detailed information on the power consumption of the server room.

3.2.1 UNH-M Power Consumption and Generation

UNH-M consumes power just like any other campus. The following chart shows the amount of energy consumed during the months of April, May, and June, in kWh:[1]

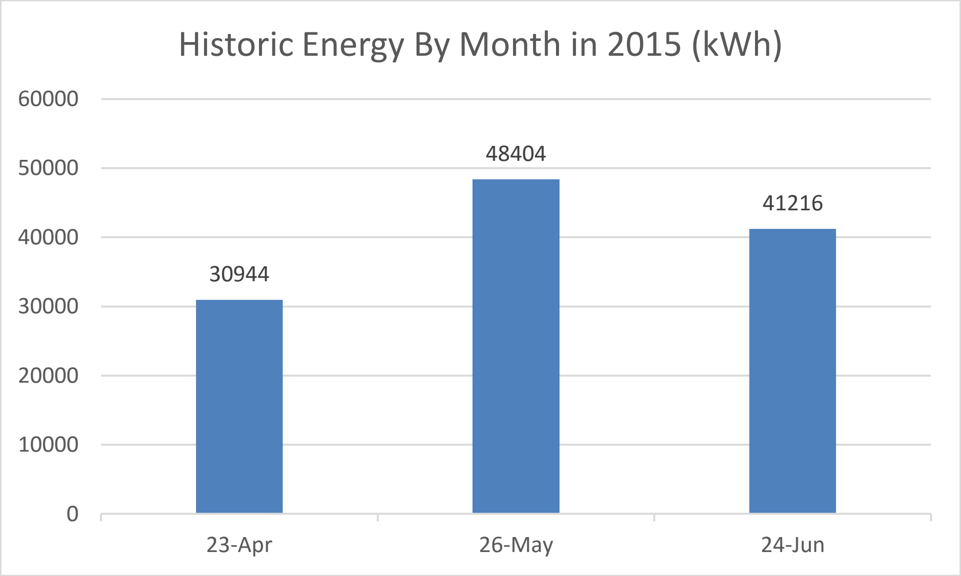


Figure 3.1 – Historic Energy By Month in 2015 (kWh)

According to the report on solar panel usage, during the time period between September 10th, 2013 and September 4th, 2014, the solar panel system generated

---

[1] Robert Beecher, the Facilities Maintenance Manager, was able to provide some information on the building's energy consumption.

88,426 kWh.[2] When compared to what is consumed each month, the amount generated is only capable of covering the costs of two months in the year. Depending on which months are looked at, it may only be able to completely cover just one.

3.2.2 Faculty & Staff Power Management Survey

UNH-M has a number of different teachers and staff as demonstrated by the following table found on the UNH-M website:

Faculty & Staff Snapshot (Undergraduate)	
No. Full-time Faculty	40
No. Part-time Faculty	60
No. Graduate School Faculty	44
No. of Full-time Staff	60

Table 3.3 – Faculty & Staff Snapshot (Undergraduate) [39]

As part of an inquiry into how UNH-M consumes power, an informal survey was conducted to see how the average teacher/faculty member manages their devices. The only people surveyed were those in offices. Eleven people total were surveyed from three different floors. There were no restrictions as to who was surveyed, only that they

---

[2] Donald Clark, Director of Property Management and Development for 1850 Associates Limited Partnership, and owner of the UNH Pandora building, was able to supply information about the solar panels on the building.

had an office and would take the survey. The following is a summary of the habits of these individuals:

Anonymous Survey Results
50% of people turn their devices off when leaving for class
63% of people turn their devices off when going home
27% of people turn off their devices when going to meetings
27% of people turn of their devices when going to meals

Table 3.4 – Anonymous Survey Results

Classes are typically once a week at three hours or twice a week at ninety minutes equating the same amount of time in the end. UNH-M residential faculty load is 3 Fall and 3 Spring classes. Courses typically run about 15 weeks depending on holidays and spring break. To keep things simple and conservative due to the lack of a fully complete and in-depth survey, the rest of the calculations will only consider the full time residential faculty. If out of the 40 full time faculty fifty percent of them did not turn of their devices when leaving for class, as suggested by the survey, this means that each semester there is approximately 5,040 hours of wasted energy (3 courses * 15 weeks * 3 hours per class = 135 hours * 20 teachers = 2,700 hours * 2 semesters = 5,400 total wasted hours). Of course this only assumes 20 individuals, when there are still part time faculty, graduate school faculty, and general staff members all with their own schedules and habits.

Using the same math, further assumptions can be made about the rest of the results of the survey. Meetings and meals can vary widely for the individual. Some people only eat lunch at their desk, or take a quick fifteen-minute lunch. Others can take up to an hour. Meetings can also take between fifteen minutes and an hour, and some people have multiple meetings a day. To keep numbers conservative, and simple, it is fair to have each meal and meeting be 30 minutes each. If only 27% shut down for these activities, it means the others waste 4,380 hours between meals and meetings (1 hour per day * 5 days * 15 weeks = 75 hours * 2 semesters = 150 hours * 29.2 teachers = 4380 wasted hours).

Finally, if only 63% of people turn their devices off when going home, they are actually wasting far more than those neglecting to do so when leaving for class, meals, and meetings combined. Normal hours of operation for UNH-M are from 8:30 am to 9:00 pm. Being conservative and assuming that the devices are used properly during school hours for this example, there are still 46,842 wasted hours over the course of a school year (14.8 teachers * 5 days * 11.5 night hours = 851 hours + (48 weekend hours * 14.8 teachers) = 1,561.4 * 15 weeks = 23,421 hours per semester * 2 semesters = 46,842 wasted hours). Class, meetings, and meals only account for 9,128 hours, which means there is 4.79 times more energy lost by faculty not turning off their devices when leaving for class. Combined, the total result of waste is 56,622 hours.

It is possible to break down per person, based on their individual habits and the devices they have in their office, approximately how much energy each of them wastes during a school year. After all it is not just the amount of hours they waste but the wattage multiplied by the hours. To find an exact number goes beyond the scope of this thesis. The following table shows an average of what each person surveyed had in their office:

Average Devices Per Office (%)	
Desktops	0.73
Laptops	1.00
Printers	0.64
Monitors	0.82

Table 3.5 – Average Devices Per Office (%)

On average everyone had at least one laptop in their office. The majority of people also had a desktop computer, a printer, and a monitor. As mentioned, without going too deeply into personal habits, it will be more than enough for the purposes of this paper to assume everyone has a laptop and to then extrapolate the kWh from the total wasted hours above. Of course, laptops are not running at full power when someone is not actively using them.

Laptops are capable of sleep modes and typically turn of their displays after a certain amount of time. Sleep mode typically uses only about 1 watt based on the tests done to personal laptops with a wattage meter. However, these sleep settings can be changed by the user, so it is difficult to be accurate without going to each laptop and checking the settings, and then comparing them to the user's habits. Instead it will be assumed that the sleep mode will begin 30 minutes after each user has stopped using it. The number of times per week a laptop is woken up will need to be considered, though that amount changes depending on the activity.

This also makes it necessary to determine which hours will be allocated to ‘sleep’ minutes at 1 watt, and which will be ‘active’ minutes before sleep mode starts at 50 watts. Earlier, it was decided that each meal and meeting would be 30 minutes. As the sleep settings are also assumed to be 30 minutes, this means that the entire 4,088 hours calculated will be used towards the 50-watt total, and not the sleep mode total. The time for going home will be simple enough, coming to 1,110 hours (5 times a week * 30 weeks * 30 minutes * 14.8 teachers = 66,600 hours / 60 = 1,110 total hours). Subtracting this amount from before, it comes to 45,732 hours for ‘sleep’ hours.

Classes are a bit more difficult to calculate as they are either once or twice a week, which comes up with an average of 1.5 times for each class. Each teacher has at least 3 classes, bringing the total up to 4.5 times per week for each person. This slightly modifies the calculation, and lowers the total amount to 63 hours (4.5 times per week * 30 weeks * 30 minutes = 4,050 minutes / 60 minutes = 67.5 hours), which is then multiplied by 20 teachers for a total of 1,350 hours.

The following tables show the differences between ‘sleep’ mode and ‘active’ mode hours:

Total Hours		
Activity	**Sleep Hours**	**Active Hours**
Class	4,050.00	1,350.00
Meal/Meeting	0.00	4,380.00
Home	45,732.00	1,110.00
Total Hours	49,782.00	6,840.00

Table 3.6 – Total Hours

The financial cost for UNH-M is $0.105600. The table below displays the final costs of the estimated waste:

Total kWh and Cost		
Mode	**kWh**	**Financial Cost**
Sleep (1 Watt)	49.782	$5.26
Active (50 Watts)	342	$36.12
Total	391.782	$41.37

Table 3.7 – Total kWh and Cost

The final results from the survey and estimated numbers were surprisingly low. It would be interesting to see more specifics on energy waste per person and per room, but that would be another thesis in and of itself. These numbers offer a good comparison point to the waste of the server room. Even as rough estimates, they pale in comparison to the expected usage of the server room.

3.2.3 Computing Server Room Expected Consumption

The Computing Technology program's server room at UNH-M may not have as many devices as all of the teachers in the building, but the servers do use far more energy per device. The following is a table summarizing the different server types, how many there are of each type, and what the wattage is per model:

Server Summary		
Model	**Quantity**	**Wattage**
Dell PowerEdge 1750	3	320W [40]
Dell PowerEdge 1950	6	670W [41]
Dell PowerEdge 2650	1	500W [42]
Dell PowerEdge 2900	1	930W [43]
Dell PowerEdge 2950	1	750W [44]

Table 3.8 – Server Summary

These servers currently run constantly. Unlike typical laptops, these servers do not have sleep or low power modes that start after a certain period of time. It is simply the nature of the server to constantly run so that someone can access it if they need to. As mentioned, there is currently no reasonable way of telling when a student will need a particular server, and this results in having all of them on all the time. Unfortunately, it is not feasible for a completely in-depth study on how much each server uses over the course of even a single semester due to time and equipment constraints. All that can be done is an estimation based on current habits. Currently the servers are running day and night as they are worked on, which may be bad for the environment and the school’s finances, but it does make it easier to estimate what that means. By using the same amount of hours wasted over nights and weekends, it is possible to come up with the following estimates:

Server Hours and kWh Summary		
Model	**Hours**	**kWh**
Dell PowerEdge 1750	9,495	3,038.4
Dell PowerEdge 1950	18,990	12,723.3
Dell PowerEdge 2650	3,165	1,582.5
Dell PowerEdge 2900	3,165	2,943.45
Dell PowerEdge 2950	3,165	2,373.75

Table 3.9 – Server Hours and kWh Summary

The final result of these servers is a rather significant 22,661.4 kWh over the time period of two semesters, which equates to roughly 3,021.52 kWh per month. A reminder that these numbers are not completely accurate, but are decent estimates. Not every server is left on all the time, meaning less energy would be used, however it is also true that the servers that are on during business hours are certainly not used during that entire period either. Most likely the numbers above are slightly lower than they should be. From personal experience it is probably a little low, as students are not always using the servers during the early hours of the school day.

What isn’t shown here is how much electricity is drawn by the air conditioning unit that is dedicated to cooling this server room. As mentioned previously, cooling costs for server rooms can be as much as 30%-50% of energy costs. Without a simple way to measure the unit, it is best to estimate based on those national averages. This means that the unit may be expending between 6,798.42 and 11,330.7 kWh. The total cost for the entire server room would then be at maximum 33,992.1 kWh over the course of the year, or $3,589.57.

The solution to this problem can be achieved with domotics. Using some ingenuity and comparatively cheap technology it is possible to avoid a large majority of the waste that is currently happening within the server room. One issue not addressed, standby power, would even be taken care of to avoid those smaller yet constant power drains. The next section details the solution and also why that particular solution was chosen.

CHAPTER 4

X-10 Controlled Server Room

This project is a combination of hardware and software designed to make the most out of domotics technologies, specifically for a server room type setting. There were a number of goals laid out in the beginning that seemed to be within reach. The primary goal was to save as much energy as possible in the easiest way possible. This meant the ability to turn off servers when they were no longer in use without physically being near the servers to do so. Access to the server room is limited, especially after hours, making remote access an essential part of the project. The secondary goal was to make the work system on an automated timer, where the server would automatically shut itself down when not in use. Though there are clearly other areas of waste which were discovered through this research, this project's focus is on saving energy in the server room and leaves open the opportunity for further study and progress in the future. Before going into any specifics on how the project works, it would be best to describe the server room environment first.

As discussed in the previous section, there are a number of different servers on a single rack. Additionally, there are several desktop computers and monitors, which will be ignored for the purposes of this project. If the project were successful, it would be possible to re-visit expanding it to other objects in the room and beyond. These servers host student projects among other things, and are accessed mainly during the day and less frequently at night. There is only one monitor shared between them all, using front facing ports to plug in. As these are all set up on one server rack, it made things easier as far as implementation and organization. Working with them proved to be difficult, as it was not clear when someone was actually using one.

The project itself appears simple at first; open a program, choose a server, and connect to that server. Where it becomes more complicated is in the details. Each server has a device between its power supply and an outlet that can be disconnected or connected remotely via an X-10 transceiver. The servers are also set up to have Wake-On-Lan (WOL) capabilities.

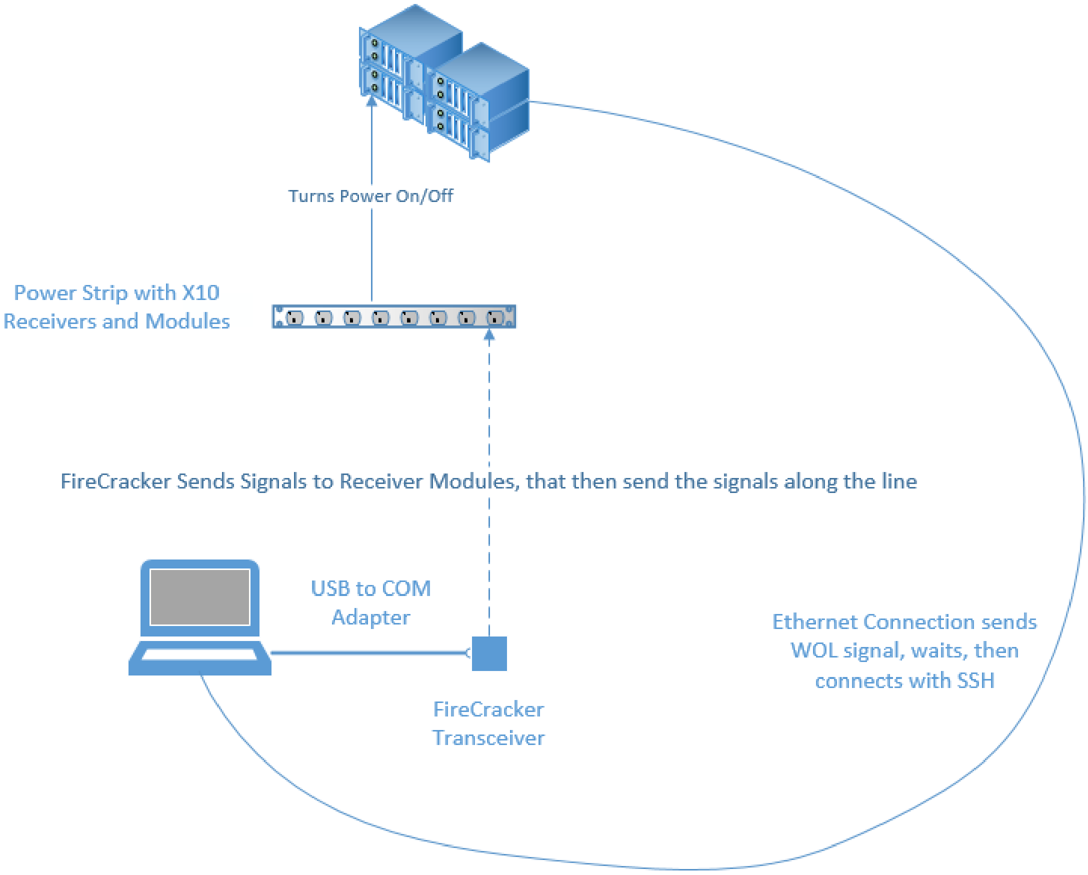


Figure 4.1 – Testing Environment

Our software will check to see if a server is available, and if not, attempt to connect its power supply and then send a WOL signal to get it started, all before finally attempting to connect via SSH after a small amount of time to allow the server to boot. Without getting too far ahead, it would be better to explain the testing environment used and go into more details about each step.

Creating and setting up a testing environment was not a simple task, and consisted of two major phases. The first phase was to take an old laptop and install Red Hat Enterprise Linux (RHEL) to match the environment of the servers. From here it would be configured to work with a USB-to-COM adapter for use with the FireCracker module. The FireCracker is an X-10 transceiver, capable of sending X-10 signals wirelessly to X-10 receivers. This is the most important device, as without it nothing else would have been possible. It was also one of the most difficult portions of the project, as RHEL had no idea what the COM adapter was and refused to send signals through it. Fortunately, once this was solved there was useful open-source code available called 'HeyU', which is designed for sending signals through X-10 devices; the FireCracker being one them.

The second half of the project consisted of successfully sending signals through the Linux laptop to an existing server and producing the desired results. As the servers had recently been upgraded to Dell PowerEdge 1950's, it was possible to borrow one of the older server blades, a Dell PowerEdge 1750. This server was running exactly the same setup as the newer models just with older hardware, making it perfect to test on. By putting an X-10 module in an outlet and connecting the server's power cord into it, it allowed the FireCracker to send signals to disconnect the power to the server without physically removing the plug from the wall. There were a few network issues to work through initially after removing it from its intended environment, but it was eventually fixed and could be tested on. Those issues consisted mainly of fixing DHCP settings to allow the server to recognize and work with the new environment, as well as adjusting the Ethernet adapter settings. The second part of this phase required light coding in Python in order to create simple logic for getting into the server.

The coding logic was simple enough, and is easily scalable. The system is broken down into two main scenarios. [3] The first scenario simply allows the application on the test laptop to SSH[4] into the test server by inputting the name of the server. The server details are kept in the same file. This scenario assumes that the server is already up and running, and simply accepts the SSH session request.

The second scenario is slightly more complex as it attempts to SSH into a specified system but fails. If the system is not responding, i.e. powered off, then the program will send out the X-10 signal to allow power to X-10 module assigned to that server. The program will then send a Wake-On-Lan (WOL) signal to the server to power it up. There is a waiting period after the WOL signal is sent to make sure the server has enough time to boot up. Once the wait is over the application will attempt to SSH once again, and should succeed if the X-10 and WOL signals were sent properly.

As the application works off of a master list file, it should be simple to scale this up as long as the X-10 signal can reach its destination. The only limitation to this design is the X-10 module itself. The master file is simply a lookup list, and every server or device should have an Internet connection making the WOL a non-issue. The fact that

---

[3] See Appendix A and Appendix B for code details.
[4] SSH, or Secure Shell, is a secure way of logging into servers remotely.

X-10 has issues over longer distances and especially in complex wiring environments begs the question as to why X-10 was chosen for this project instead of a more reliable, easier to implement domotics technology.

4.1 Why X-10

X-10 was chosen for several different reasons. The first is that it fit the proposed solution perfectly. The range of the protocol was enough to fulfill the project’s needs, the device limit was not an issue, and security was not a concern. X-10 was also an option of convenience. Several X-10 modules were readily available and a FireCracker transceiver was on hand so it only made sense to begin working with them at least as a baseline. This point of this project was not to create a commercial grade solution but instead to create a functioning prototype for a proof of concept. domotics are already capable of turning things off and on, at their core, yet what was needed was outside of the scope of what all domotics offered for out of the box features. Most importantly X-10 looked promising for the server room problem.

X-10 also proved to be viable for such a small server room. One of the largest concerns when testing with X-10 was the attenuation of the signal and the wiring of the various environments that it was tested in. This was fortunately not an issue in practice and the signals were sent and received properly without issue. Even if there had been an issue, there was backup plan to use a large power strip and have the signals sent to the receiver at the far end. This would have forced the signals to pass through each device to then take the appropriate response.

X-10 provided a more than suitable way to test out ideas as research continued alongside it. The entire project was too small, physically, for X-10 to fail. The technology was old, but that only created additional challenges to overcome rather than becoming a real problem. It is definitely still a viable domotics solution, as long as the user knows its limitations and is willing to work with them.

CHAPTER 5

SUMMARY

Throughout this thesis we compared and contrasted various technologies in the field of domotics. X-10 is an older, simpler technology that uses existing electrical wiring in order to send messages to devices. UPB is a more expensive but improved version of X-10, with greater range and less problems overall. ZigBee was the first technology examined that started using wireless signals to send messages, yet it is manufactured by many different companies, causing there to be issues when trying to pair up devices between them. Z-Wave uses a different radio frequency to avoid clutter in the common 2.4 GHz frequency. Unfortunately, Z-Wave has a shorter range because of its lower frequency, not to mention other network issues that allow for very few devices to successfully work together in a single network. WeMo was the only modern technology without any significant faults, other than that of security. Every domotics listed here has either no security or has largely flawed security in one way or another.

The most ideal solution would have been an intelligent power supply strip that provides many features of the domotics examined with none of the drawbacks. There was significant work done by a Kickstarter campaign to fill this exact niche, but nothing ever materialized. There are other intelligent power supplies that exist today though they are not as useful or as appealing. There may have been one that would have worked for this project, yet by the time they were discovered it was too late and a solution had already been designed with X-10 in mind.

As mentioned, the solution we chose to overcome these problems was X-10. We decided on X-10, as it was both cost effective and worked for our needs. Although it lacks smart phone support, wireless signals, and can suffer from signal degradation over long distances, it filled the role well. It was simple to install, capable of sending signals via a computer through the FireCracker, does not suffer from networking issues in small spaces, and is cost effective when needing to purchase multiple devices.

As we began to look into the problem we also did a power study on the building and the computing server room. We not only looked at power consumption but also cost of energy and what sources were used. We studied both the overall usage and focused on the server room's usage, specifically the Capstone server rack. The results showed that individual faculty behavior does not contribute too much to wasting energy, at least with personal laptops. More significantly, the servers demonstrated a costly waste per school year.

The solution that X-10 provides is close to that of the testing environment mentioned earlier. Students will be able to use any outside computer to log into the primary server, which resides in the server room, and which will then send out X-10 signals when requested to turn on the other servers specified by the user. After the power is connected, a power-on signal will be sent to the server. Once the server is finished booting up, the software will connect the user to the server via SSH and they will be able to do their work. This has not yet been implemented, but it is simple to see how it can be done with low cost and effort on the part of UNH-M. Figure 5.1 shows what that may look like:

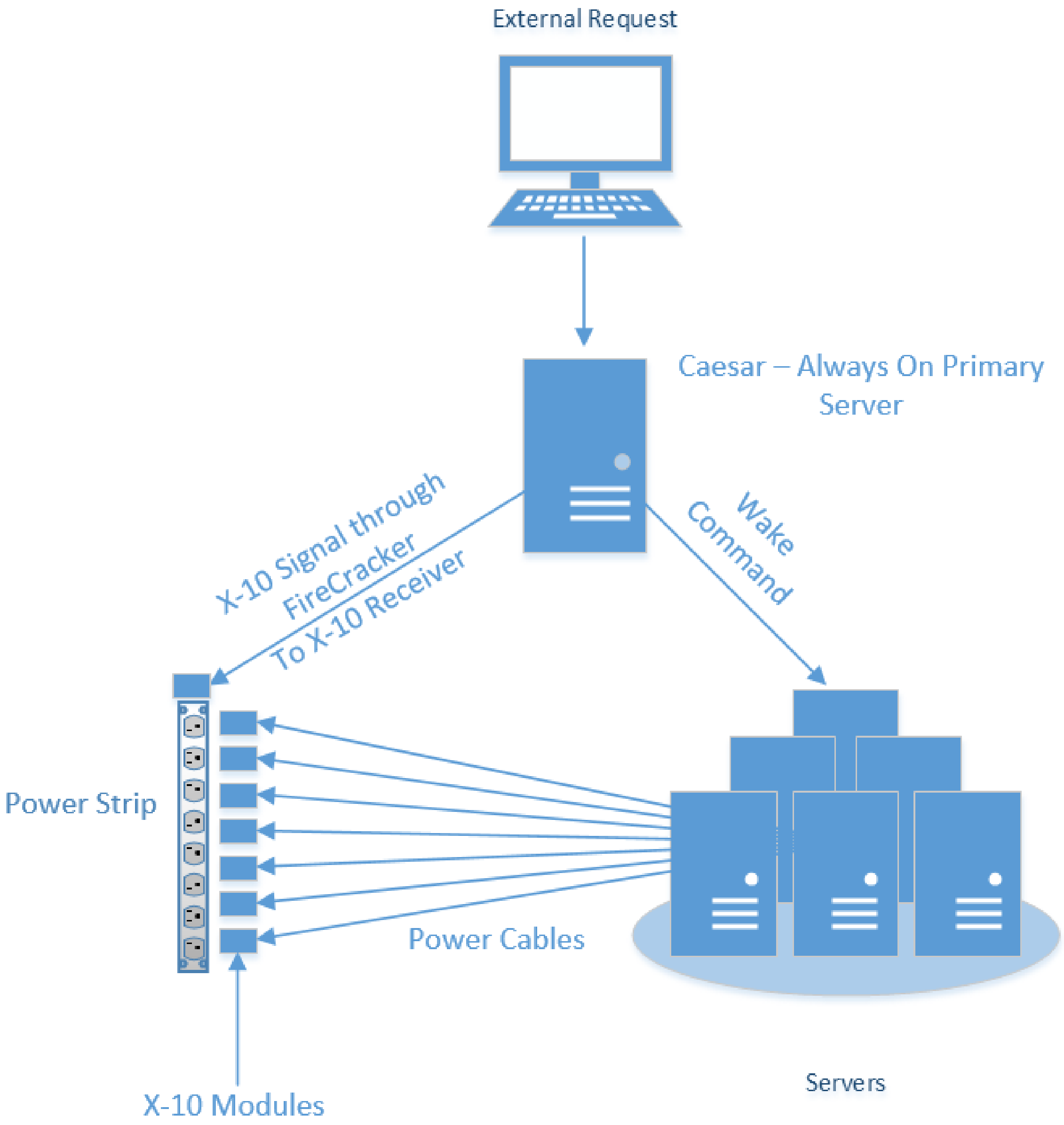


Figure 5.1 – Computing Department Server Room Setup

5.1 Conclusion

There is a significant amount of waste within the UNH-M Pandora building. A small portion of this waste is simply due to poor energy practices with the staff. If users could power down their equipment, or at the very least turn them to sleep mode, when not in use it could reduce power consumption significantly based on these estimates.

It was surprising that the server room was costing over $3,500 every school year by itself. Moreover, that this was not counting the summer, during of which the server habits were not examined. This suggests that the true waste is larger than what is demonstrated here. With the X-10 solution the server room should save nearly all of that wasted energy. With the servers only being on when they need to be, and having users shut them down afterwards, there will be far less energy consumed. In addition to the direct energy costs of the servers decreasing, the indirect cooling costs will decrease as well.

There is a great opportunity for future students to take advantage of this groundwork and move forward with it. There are many different ways this setup can be expanded and added to. There can be direct upgrades, such as creating an effective scheduling program. There is also room to do a much more detailed analysis of server and office power consumption. In conclusion, the X-10 solution created should save the UNH-M Pandora building significant energy and money not only in the short term, but for many years to come.

BIBLIOGRAPHY

1. "Saving Energy in Server Rooms and Closets." *Saving Energy, Energy Efficiency in Server Rooms*. N.p., 11 Apr. 2012. Web. 29 Mar. 2016. <http://www.nrdc.org/energy/saving-energy-in-server-rooms.asp>.
2. Koomey, J.G. "ESTIMATING TOTAL POWER CONSUMPTION BY SERVERS IN THE U.S. AND THE WORLD." (n.d.): n. pag. 15 Feb. 2007. Web. 29 Mar. 2016. <http://www-sop.inria.fr/mascotte/Contrats/DIMAGREEN/wiki/uploads/Main/svrpwrusecompletefinal.pdf>.
3. "FREQUENTLY ASKED QUESTIONS (FAQS)." *Standby Power*. N.p., n.d. Web. 29 Mar. 2016. <http://standby.lbl.gov/faq.html>.
4. Brooks, L. "X10 Home Automation: Upgrading and Alternatives." *Digital Landing X10 Home Automation Upgrading and Alternatives Comments*. N.p., 27 Mar. 2014. Web. 29 Mar. 2016. <http://www.digitallanding.com/x10-home-automation-upgrading/>.
5. "X10 (industry Standard)." *Wikipedia*. Wikimedia Foundation, 9 Mar. 26. Web. 29 Mar. 2016. <https://en.wikipedia.org/wiki/X10_(industry_standard)>.
6. "How X10 Works." *How X10 Works*. Smart Home USA, n.d. Web. 29 Mar. 2016. <http://www.smarthomeusa.com/how-x10-works/#theory>.
7. MATCH Technology Tutorial - Advantages/Disadvantages Of X10." *MATCH Technology Tutorial - Advantages/Disadvantages Of X10*. MATCH - Home Care Technologies, n.d. Web. 29 Mar. 2016. <http://www.cs.stir.ac.uk/~kjt/research/match/resources/tutorial/HomeCare_Networks/X10/X10_6.html>.

8. Custom Solutions, Inc. "X10 Debugging Tips." *X10 DEBUGGING TIPS* (n.d.): 4-5. Web. 29 Mar. 2016. <http://www.csi3.com/app_x10.pdf>.
9. Ganesh T.S. "Home Automation and the 'Internet of Things'" *Anand Tech*. N.p., 4 Oct. 2012. Web. 29 Mar. 2016. <http://www.anandtech.com/show/6354/home-automation-and-the-internet-of-things/2>.
10. "FAQ: Why Is UPB from Simply Automated Significantly Better than X10?" *Simply Automated Incorporated*. N.p., 18 July 2012. Web. 29 Mar. 2016. <http://simply-automated.com/blog/2012/07/faq-why-is-upb-significantly-better-than-x10/>.
11. PulseWorx. "UPB Technology Description." *UPB Technology Description* (n.d.): 4-5. 16 Feb. 2005. Web. 29 Mar. 2016. <http://pcslighting.com/resources/PulseWorx/Installation/Documentation/UPBDescriptionv1.4.pdf>.
12. "About UPB Technology." *Simply Automated Incorporated*. N.p., 2016. Web. 29 Mar. 2016. <http://www.simply-automated.com/UPB_Technology.php>.
13. "Universal Powerline Bus." *Wikipedia*. Wikimedia Foundation, 8 Mar. 2016. Web. 29 Mar. 2016. <https://en.wikipedia.org/wiki/Universal_powerline_bus>.
14. Fuchsia Research Corporation. "Home Automation White Paper / X-10 vs. Insteon vs. Z-Wave vs. RadioRA 2." *Simply Automated Incorporated*. N.p., 10 Oct. 2014. Web. 29 Mar. 2016. <http://simply-automated.com/blog/2014/10/home-automation-white-paper/>.
15. Poole, I. "ZigBee Technology Tutorial." *Radio-Electronics*. N.p., n.d. Web. 29 Mar. 2016. <http://www.radio-electronics.com/info/wireless/zigbee/zigbee.php>.
16. "What Is UPB-Universal Powerline Bus Basics in IoT." *RF Wireless World*. N.p., n.d. Web. 29 Mar. 2016. <http://www.rfwireless-world.com/Terminology/UPB-basics.html>.

17. Digi International Inc. *FAQs Drop-in Networking* (n.d.): 3. *Digi*. 12 June 2007. Web. 29 Mar. 2016. <http://www.digi.com/pdf/faq_drop_in_networking.pdf>.
18. Gascon, D. "Security in 802.15.4 and ZigBee Networks." *Libelium*. N.p., 28 Apr. 2009. Web. 29 Mar. 2016. <http://www.libelium.com/security-802-15-4-zigbee/>.
19. Bowers, B. "ZigBee Wireless Security: A New Age Penetration Tester's Toolkit." *Cisco*. N.p., 21 Dec. 2015. Web. 29 Mar. 2016. <http://www.ciscopress.com/articles/article.asp?p=1823368&seqNum=4>.
20. Higginbotham, S. "ZigBee and Z-wave Are Out. Broadcom’s New Chips Bet on Bluetooth and Wi-Fi for IoT." *Gigaom*. N.p., 29 May 2013. Web. 29 Mar. 2016. <https://gigaom.com/2013/05/29/zigbee-and-z-wave-are-out-broadcoms-new-chips-bet-on-bluetooth-and-wi-fi-for-iot/>.
21. Develco. "Smart Plug." *Develco Products*. N.p., 3 Apr. 2016. Web. 3 Apr. 2016. <http://www.develcoproducts.com/products/smart-plugs/smart-plug/>.
22. "WHAT IS Z-WAVE?" *Smart Home - Home Automation Superstore*. N.p., n.d. Web. 29 Mar. 2016. <https://www.smarthome.com/sc-what-is-zwave-home-automation>.
23. Frenzel, L. "The Fundamentals Of Short-Range Wireless Technology." *Electronic Design*. N.p., 11 Oct. 2012. Web. 29 Mar. 2016. <http://electronicdesign.com/communications/fundamentals-short-range-wireless-technology>.
24. "FREQUENTLY ASKED QUESTIONS." *Z-Wave*. N.p., n.d. Web. 29 Mar. 2016. <http://www.z-wave.com/faq>.

25. Parrish, K. "ZigBee, Z-Wave, Thread and WeMo: What's the Difference?" *Tom's Guide*. N.p., 14 July 2015. Web. 29 Mar. 2016. <http://www.tomsguide.com/us/smart-home-wireless-network-primer,news-21085.html>.
26. Smith. "Hacking and Attacking Automated Homes." *Network World*. N.p., 25 June 2013. Web. 29 Mar. 2016.
27. "How to Connect Your WEMO Devices to Your Amazon Echo." *Belkin*. N.p., n.d. Web. 29 Mar. 2016. <http://www.belkin.com/us/support-article?articleNum=157351>.
28. "Ensuring That Your Mobile Device Can Control the WEMO Devices in Your Network." *Belkin*. N.p., n.d. Web. 29 Mar. 2016. <http://www.belkin.com/us/support-article?articleNum=8020>.
29. "WEMO and Security." *Belkin*. N.p., n.d. Web. 29 Mar. 2016. <http://www.belkin.com/us/support-article?articleNum=80322>.
30. Comet, A.J. "TESTED: Smart Home Automation – Comparing Apple HomeKit vs Samsung SmartThings Hub vs Wink Connected Home Hub vs INSTEON Hub Pro vs Amazon Echo vs Belkin WeMo." *Ari Jay Comet Blog*. N.p., 16 Nov. 2015. Web. 29 Mar. 2016. <http://arijaycomet.com/2015/11/16/tested-smart-home-automation-comparing-apple-homekit-vs-samsung-smartthings-hub-vs-wink-connected-home-hub-vs-insteon-hub-pro-vs-amazon-echo-vs-belkin-wemo/#belkin-wemo>.
31. "WeMo® Switch." *Belkin*. N.p., n.d. Web. 29 Mar. 2016. <http://www.belkin.com/us/F7C027-Belkin/p/P-F7C027/?initialMaxPrice=199.99&initialMinPrice=15.00&maxPrice=65.32&minPrice=15>.

32. Yiu, R. "Smart Power Strip- Control Your Appliances from Anywhere." *Kickstarter*. N.p., n.d. Web. 29 Mar. 2016. <https://www.kickstarter.com/projects/337922278/smart-power-strip-control-your-appliances-from-any>.
33. "Estimating Electricity Usage." *Energy Savings*. N.p., n.d. Web. 29 Mar. 2016. <https://www.energysavings.com/energy-consumption.html>.
34. "Manchester, NH Electricity Rates." *Electricity Local*. N.p., n.d. Web. 29 Mar. 2016. <http://www.electricitylocal.com/states/new-hampshire/manchester/>.
35. "Fossil Fuels." *Environmental and Energy Study Institute*. N.p., n.d. Web. 29 Mar. 2016. <http://www.eesi.org/topics/fossil-fuels/description>.
36. Thibodeau, P. "Data Centers Are the New Polluters." *Computerworld*. N.p., 26 Aug. 2014. Web. 03 Apr. 2016. <http://www.computerworld.com/article/2598562/data-center/data-centers-are-the-new-polluters.html>.
37. Delforge, P. "America's Data Centers Consuming and Wasting Growing Amounts of Energy." *Energy Efficiency, Data Centers*. N.p., 2 June 2015. Web. 29 Mar. 2016. <http://www.nrdc.org/energy/data-center-efficiency-assessment.asp>.
38. Hruska, J. "IBM Wants to Cool Data Centers with Their Own Waste Heat." *ExtremeTech*. N.p., 28 July 2015. Web. 29 Mar. 2016. <http://www.extremetech.com/extreme/211008-ibm-wants-to-cool-data-centers-with-their-own-waste-heat>.
39. University of New Hampshire at Manchester. "Fast Facts." *University of New Hampshire at Manchester*. N.p., n.d. Web. 03 Apr. 2016. <http://manchester.unh.edu/about/fast-facts>.

40. Dell. "Dell™ PowerEdge™ 1750 Systems User's Guide." (n.d.): n. pag. Web. 3 Apr. 2016. <http://downloads.dell.com/manuals/all-products/esuprt_ser_stor_net/esuprt_poweredge/poweredge-1750_user%27s%20guide4_en-us.pdf>.
41. Dell. "DELL™ POWEREDGE™ 1950 III SERVER." (n.d.): n. pag. Web. 3 Apr. 2016. <http://www.dell.com/downloads/global/products/pedge/en/pe_1950_III_spec_sheet.pdf>.
42. Noesisinc. "Technical Specifications : Dell PowerEdge 2650 Systems User's Guide." *Noesis Consulting Group Inc.* N.p., 19 Dec. 2001. Web. 03 Apr. 2016. <http://www.noesisinc.com/links/DellDocs/PowerEdge2650/UserGuide/5g387aa0.htm>.
43. Dell. "Dell PowerEdge 2900 Server." *Dell PowerEdge 2900 Server* (n.d.): n. pag. *Dell*. Web. 3 Apr. 2016. <https://www.dell.com/downloads/emea/products/pedge/en/PE2900_Spec_Sheet_Quad.pdf>.
44. Dell. "Dell PowerEdge 2950 Server." *Dell PowerEdge 2950 Server* (n.d.): n. pag. *Dell*. Web. 3 Apr. 2016. <http://www.dell.com/downloads/emea/products/pedge/en/PE2950_Spec_Sheet_Quad.pdf>.

APPENDIX A – X10.sh

The following code is the primary software component of the X-10 solution. There is a secondary file located in Appendix B that is necessary for this code to work. All code is written in Bash.

There are some modifications to be made upon installation in a new environment. X-10 modules must be designated properly, as well as IP addresses, MAC addresses, and Ethernet ports. Several examples have been included to help those who would like to adapt this or improve upon it.

```
#!/bin/bash

# Definitions of IP addresses
asterix=192.168.1.1
obelix=192.168.1.2
miraculix=192.168.1.3
traubadix=192.168.1.1
majestix=192.168.1.1
idefix=192.168.1.1
automatix=192.168.1.1
brutus=192.168.1.1
rome=192.168.1.1
test=10.0.0.113
# End of host list


while [ "$host" != "quit" ]; do
echo "Please choose a server to connect to:"

select choice in "Asterix" "Obelix" "Miraculix" "Test Server"; do
case $choice in
      "Asterix")
   echo "Attempting to connect to Asterix..."
   if myping.sh $asterix; then
      #Server is on and responding to pings
       echo "Asterix is alive."
       echo "Do you want to connect? yes/no"
       #The next if statement will trigger the SSH command.
       read input3
       if [ "$input3" = "yes" ]; then
             echo "Now connecting to Asterix…"
             ssh $asterix
       else
            #If it is already on, the user can turn it off.
             echo "Do you want to turn this server off? yes/no"
             read inputoff
             if [ "$inputoff" = "yes" ]; then
                   #Turn off corresponding server module.
```

```
                   heyu foff M1 #Enter correct settings here
             fi #end of inputoff
       fi # end of input3
   else
       #Server is off or unresponsive to ping
       echo "Asterix is offline. Do you want to turn it on? yes/no"
       read input2
       if [ "$input2" = "yes" ]; then
             # This will call ‘heyu’ to turn on the module.
             heyu fon M1
             #Then WOL on the correct MAC and Ethernet port.
             Ether-wake -i eth0 00:0f:1f:03:e1:5b
             echo "Turning on power. Waiting for bootup."
             #Wait ~10 minutes for boot. Increase or decrease per server.
             sleep 300
             # Then it will SSH into the system
             echo "Connecting to Asterix…"
             ssh $asterix
       fi # End of input2
   fi # End of outer if statement for Asterix
;; #End of Asterix Select


      "Obelix")
      #Example
   echo "Obelix"
;;


      "Miraculix")
      #Example
   echo "Miraculix"
;;


   "Test Server")
   echo "Attempting to connect to Test Server..."
   if myping.sh $test; then
       #Server is on and responding to pings
       echo "Test Server is alive."
       echo "Do you want to connect? yes/no"
       #The next if statement will trigger the SSH command.
       read input3
       if [ "$input3" = "yes" ]; then
             echo "Now connecting to Test Server…"
             ssh $test
       else
             #Should trigger the off function for
             #heyu if requested
             echo "Do you want to turn this server off? yes/no"
             read inputoff
             if [ "$inputoff" = "yes" ]; then
                   # Turn off corresponding server module.
                   heyu foff M1 #Whatever the module is
             fi
       fi
   else
```

```
       #Server is off or unresponsive to ping
       echo "Test Server is offline. Do you want to turn it on? yes/no"
       read input2
       if [ "$input2" = "yes" ]; then
             # This will call ‘heyu’ to turn on the module.
             heyu fon M1
             #Then WOL on the correct MAC and Ethernet port.
             Ether-wake -i eth0 00:0f:1f:03:e1:5b
             echo "Turning on power. Waiting for bootup."
             #Wait ~10 minutes for boot. Increase or decrease per server.
             sleep 300
             # Then it will SSH into the system
             echo "Connecting to Test Server…"
             ssh $test
       fi
   fi # End of outer if statement for Test Server
;; #End of Test Server Select statement

esac

done #select statement
done #while statement
```

APPENDIX B – MYPING.SH CODE

This is the secondary file necessary to run the primary software located in Appendix A. This code is referenced by the main file in order to check to see if a server is online or offline.

```
#!/bin/sh
ping -c 1 "$1" >/dev/null
```