3-14-2011

Integrated Home Server

Santiago John Rose
Santa Clara University

Integrated Home Server

By

SANTIAGO JOHN ROSE

MASTER'S THESIS
(M.S. Computer Engineering)

Submitted in partial Fulfillment of the Requirements

For the Degree of Master of Science

in Computer Engineering

in the School of Engineering at

Santa Clara University, 2011

Santa Clara, California
UNITED STATES OF AMERICA

DEDICATED

To my parents (Amma and Appa) for planting faith in my heart,
brothers (Panneer and Xavier) for strengthening and supporting me,
niece (Printha), nephew (Rakesh), sister-in-law (Anitha) for their prayers,
and to the Society of Jesus for what I am today.

ACKNOWLEDGEMENTS

It is beautifully said, “An attitude of gratitude is a Beatitude.” Beatitude comes from the Latin word *beatus*, meaning “blessed” or “happy.” Yes, I am blessed with so many people who helped me directly and indirectly during my journey at SCU, and therefore I am happy for what they have been to me.

First of all, I thank God for having created me, calling me to live a Jesuit way of life in the Society of Jesus and giving me this wonderful opportunity to study at the Engineering School of Santa Clara University (SCU) located in Silicon Valley. I consider it both as a blessing and privilege. Secondly, I thank God for my parents whose prayers nurture me spiritually to live a joyful life. **The Jesuit Communities of Santa Clara University and Xavier Institute of Engineering, Mumbai, India,** have assisted me to make use of this opportunity and to do this course to the best of my ability by providing me all the logistics I required. I am deeply grateful to the Jesuits of both institutions.

You might wonder why I have chosen to do this project in Linux OS. Ever since I landed at Xavier Institute of Engineering (Aug. 2008) after teaching for a few years at the Jesuit High Schools in Gujarat, India, I have been fascinated by the ‘open source Linux operating system’ simply because of its philosophy. The philosophy of open source system, I found, fitted well with the philosophy of human life itself – “give back to the society what we have freely received from the society.” I just fell in love with this philosophy of living and sharing. This is the reason for my choosing Linux OS.

While I was in India, with the help of Prof. Kashinath M Kulkarni and Dr. Glenn Baptista I prepared a DVD consisting of all the engineering softwares incorporated into Fedora Core 7. Something similar I wanted to do it here at SCU. This idea led

me to explore an OS that is viable and serves all the needs of a home or office. And that is how the "Integrated Home Server" (IHS) was born.

IHS is basically built on Dr. Glenn's 'Gateman Lifestyle Server Security System' which is carved from CentOS version 5.3. CEntOS stands for **C**ommunity **ENT**erprise **O**perating **S**ystem. I became interested in it simply because it is community supported. This project, therefore, would not have been possible without Dr. Glenn who has been my mentor, guide and teacher.

When I presented this idea to my academic advisor at SCU, Dr. Weijia Shang, she immediately gave me a green signal to work on it during summer of 2010. Her instant "yes" encouraged me and I felt supported. She went further to make the research laboratory available for my work and periodically assisted me with new and creative ideas.

Needless to say, I have been receiving constant support and encouragement from my Jesuit Companions at SCU and Prof. Kashinath Kulkarni. I am very much indebted to their creative suggestions, concern, and thoughtfulness.

I am grateful to Lantz Johnson, the manager of Design Center, for having uploaded the entire OS on SCUengr website and made it available for the students to use, experiment and play with it.

Finally, I thank the Dean of School of Engineering, Dr. Godfrey Mungal and Associate Dean, Dr. Aleksandar Zecevic, for their encouragement and support. Gratitude, as they say, is the memory of the heart. Truly, my heart is filled with gratitude. And this memory will remain with me till this mortal body of mine lingers on this earth.

INTEGRATED HOME SERVER

Santiago John Rose
Department of Computer Engineering
Santa Clara University
Santa Clara, California, 2011.
Advisor: Dr. Weijia Shang

ABSTRACT

Since the advent of the microprocessor in the 1970s, the market for consumer electronics has exploded with new devices changing the way we live and do business. Today, mobile phones, cameras, PCs, iPads, mp3 players, network media players, security systems, automation and IT systems, all have common functionality and there is an increasing need for unification of access to all these devices around a common server based architecture to unlock the benefits of smart integration and to simplify access for the end user.

IHS project is designed to provide to its business and home owners a unified network for all IT and electronic systems within a home or an office. This system integrates security, surveillance, access and attendance, home automation, audio and video players, File Server, Email Server, SMS Server (Texting), HTTP Proxy Server, DHCP Server, a caching DNS Server, Web Server and an internet gateway with an automatic virus scanner.

In fact, it is a comprehensive system that completely governs a place wherever it is installed and provides integrated remotely accessible infrastructure for a Home or Business. Access to all home and business systems is available from any computer on the LAN, the internet and mobile phone.

IHS is built around the Gateman Lifestyle Server which uses the robust Enterprise Linux Kernel CEntOS 5 and is written in Java. It can be accessed from Windows, MAC, Linux machines and i-phones as well as from any device that has a Java script enabled web browser. The device driver architecture allows additional electronic hardware to be incorporated making it relevant and extendable well into the future.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

One of the questions that haunts not only Americans but every human being on earth is, "How safe am I or is my home?" It is not only because of what happened on 9/11. We witness the world becoming increasingly unsafe due to wars, religious intolerance, climate change and other human rights violations errupting out of selfish and vested interests. More recently the intellectuals predicted that the next world war would not be fought on the ground or in air, rather it will be a cyberwar.

To make us more secure, the latest technologies for mobiles, TVs, Cameras and other gadgets are coming into the market aiming at the Home Server / Owner. But what has an OS's (Operating System's) role in serving the purpose of a home or an office? Sadly, for the past four decades OS has remained PC-centric. Until now we do not have an OS that integrates all these lifestyle products. Today we do everything through i-phones – browsing, texting, receiving and sending emails, bank transactions, flight boarding passes, barcode checking of the tickets, etc,. i-phones / i-pads may soon replace PCs with more memory and speed. But again can my mobile talk to my PC at home or can I start the sprinkler to water my garden from my office? No.

It is not a secret that Microsoft Home Server is just a file server and is not able to protect a home or an office. A radical new approach to a solid kernel-based OS safe from network attacks is required to satisfy the needs of the home owner today. People are tired of things being provided in bits and pieces, e.g., file server, proxy server, attendance, surveillance, mail server, gateway, etc. They want an Integrated Home or Business Server that could take care of everything at all times, so that home or business owners get a sound sleep and peace of mind, and the entire home / office be at owner's finger tips.

When we look for "Home Server"[1] in Google, we see what people expect a home server to be.

[1] http://en.wikipedia.org/wiki/Windows_Home_Server

The expectations are that it should be headless, media serving, web server, web proxy, email server, DHCP and DNS server; have a centralized storage system, security and surveillence, and do home automation. Do we have an OS that can meet all the above expectations? No.

This project, IHS, provides a solution. IHS is an adhesive kind of product that puts all the fragmented systems together and integrates the systems into one unit. It demonstrates how it can act like "gateman" to our homes and business offices.

What can IHS do? Let me answer in the form of questions. Would you like to take a look at the children or pets at home while you were at work? Are you concerned about the safety of your home when you are on vacation in Hawaii or India or Seychelles? Would you like an SMS alert when something of interest happened at home or when someone enters your home when you are away? Do you want an automatic report along with the pictures to be sent to your email address, when someone enters your home? Do you wonder if you had left the lights on or the TV or home heater and wanted to turn them off when you are at your office? Would you like to have access to all your photographs, home videos, music, and other files when you visit a friend or are on vacation overseas? Would you like to take a look at your emails? You are away for the holidays and you want to sprinkle water your garden and turn on the lights automatically so that your home looks lived in. What would you do? For all the above, "an Integrated Home Network" will be an ideal answer to the above questions.

IHS uses UPB (Universal Powerline Bus) to control remotely all the electrical appliances at home. Why UPB? In a UPB system the control signals are sent on the home's main power cable so that we do not need to do any special wiring. Basically, IHS communicates with UPB devices (like a pump, oven, lights and all that operate in 110 V or 230 V) via a serial powerline interface module (PIM). This device is plugged into the required phase of your 110V or 230V line and the serial interface is connected to your PC using a Serial cable.

A general expectation of a Home / business Server and its characteristics are enunciated in the first chapter. In the second chapter, I have dealt specifically with a home network taking the

different computers and OSs into consideration, and how IHS could comprehensively implement them. The existing home security and home automation systems available in the market and the superiority of IHS implementation are articulated in the third and fourth chapters. The fifth chapter deals with Home Gateway in general and its different components working in IHS. The physical layer is discussed in the last chapter and a proposal for the future research in ZigBee is mentioned in the conclusion.

Some have asked me how and why I named the project IHS. As I started this project, I kept thinking hard about what name would be appropriate. Nothing seemed accurate as to what the project actually signified. Discouragement overshawdowed my will for the new search. At that time I was reminded of Shakespeare's famous quote from *Romeo and Juliet*: "What's in a name? That which we call a rose by any other name would smell as sweet." Rose being my last name, I was a bit consoled. But my search for an appropreate name continued. One day Dr. Glenn suggested, "What about an '**I**ntegrated **H**ome **S**erver'?" Immediately, I clung to it both for the name itself and its acronym. I felt this to be the exact name for the project. And the acronymn, IHS, because I am a Jesuit, stood close to my heart. IHS is the first three letters of Jesus in Greek (IHΣ - *iota*-*eta*-*sigma*). St. Ignatius of Loyola, the founder of the Society of Jesus, chose IHS emblem to signify what he called the Least Society.

"IHS" has also a backronym and sometimes interpreted as meaning *Iesus Hominum Salvator* ("Jesus, Savior of (wo)men" in Latin). Usually, there is a cross over the horizontal line of 'H' symbolizing that God blesses all humans as H stands for Humans. In our case, blessings are on Home, God's favorite dwelling place. May this Integrated Home Server be like the True IHS blessing and protecting humans and homes across the world. This is my sincere wish to all those who would be using this Integrated Home Server carved from the Open Source.

CHAPTER 1

1. Essentials of a Home Server

What does a home server do at home? What should be its functions? What should its daily or routine jobs at home be? What are the expectations of a family when they want to install a home server? These are the questions that will be dealt in this chapter. Let us take one by one.

1.1. Common Storage System (CSS)

A family at home shares most of the things that are at home. So also files common to the family. A home server must act like a common storage system that is centralized. The advantage of having centralized storage is that each family member can access remotely and if they are writing a book, say family history, each one can attribute their part in the same file whenever they get time. This common or centralized storage system could include their personal vis-a-vis common files, photographs, music files, video files, etc. And to a great extent it gives a guarantee for the security. It offers a flexibility for access permissions, say, the children are not allowed to view some video films. Permissions could be granted or denied depending upon the nature and confidentiality of the matter.

CSS also paves the way for easy printer sharing as it is networked to all the individual PCs. If it is equipped with samba suite, it can provide domain control, custom login scripts and personalized application preferences. This avoids having multiple accounts on each computer in the home. CSS is expected to have two important functions among others: backup and disaster recovery. It should have HTTP based interface so that one can access his or her files from anywhere in the world.

1.2. Voice-Video-Data Service

We have facing a world that is multimedia centered. Multimedia includes voice-video-data files. People want a remote access to media content stored at home server. A home server, therefore, must be able to handle voice-video-data files and to serve them to other devices

used at home. How wonderful it would be, if it has all the family videos, music, photos and history which could be accessible from any part of the world! This is what a home server is meant to be. iTunes users, for example, can access their media files from anywhere at home. The recent 4G i-phones would be able to play remotely from any corner of the globe.

Hewlett-Packard (HP) had come out with PacketVideo. For Linux users, there is Linux MCE[2]. which allows other devices to boot off a hard drive image on the server, allowing them to become appliances such as set-top boxes. There are many open source projects like VideoLAN, SlimServer, etc fully integrated for a seamless home theater/automation/telephony experience. The recent phenomenon is to connect a Network Media Player[3] (NMP) to the server and play it on TV or home theater.

1.3. Remotely Accesscible

Today we are talking about 'virtual office'. The giant software companies like Google, Microsoft, Yahoo, Apple, etc, have allowed their employees to work from coffeteria, home, garden and wherever they feel comfortable and productive. One can even work while travelling on a plane. With this new outlook towards modern work-culture, a home server should be accessible remotely as it is also a centralized storage system. All that it requires a static IP which is very easy to aquire from an ISP. IPv6 that has the size of 128 bits would make the IP addresses available abundantly.

Any web browser should be able to connect to the home server via internet. GUI[4] as well as command-based access are also possible. Telnet, SSH[5], etc., helps to troubleshoot or change

[2] LinuxMCE (Linux Media Center Edition) is a free and open source software platform with a 10-foot user interface designed to allow a computer to act as a home theater PC (HTPC) for the living-room TV, personal video recorder, and home automation system. It allows control of everything in the home, from lighting and climate to surveillance cameras and home security. It also includes a full-featured VoIP-compatible phone system with support for video conferencing.

[3] This is used to stream media across a network, sometimes directly to a hi-fi or television

[4] A *GUI* (Graphical User Interface) offers graphical icons, and visual indicators, as opposed to text-based interfaces, typed command labels or text navigation to fully represent the information and actions available to a user.

[5] **SSH** is a network protocol that allows data to be exchanged using a secure channel between two networked

configuratoin in the server, whereas VNC and Webmin offer GUI access.

1.4. Serving Web Pages

A web server is an intermediate server between client request and server resource. A home server should also act like a web server that serves web pages to all the family member when requested. A family web page can be set up in the same server and it could supply images, documents, java scripts, etc. It will have the facility to upload and download media files. There is no additional software required for a home server to be a web server. All that a client needs is a web browser which every OS comes with it.

Today Apache web server takes about 54% of the market while Microsoft's IIS (Internet Information Services) is only 24%. It should be a kernal-mode web server as it is directly linked to hardware (memory, buffer, CPU, network adapters, etc). Virtual hosting (serving many web sites using one IP-address), large file support, bandwidth throttling (limit the speed of responses in order to not saturate the network and to be able to serve more clients) and server-side scripting (generating dynamic web pages) are some of the salient features of web server.

1.5. Acting as Web Proxy

A proxy that focuses on WWW (World Wide Web) traffic is called a "web proxy". Public proxies are rather slow and is cumbersome to configure for each home user, and therefore we need a home server to be a web proxy, not to rely upon public proxies. The chief aim of a web proxy is to serve as a web cache[6].

As for as home is concerned, web proxy is a must as the family may like to block some sites from their children. Blacklisted sites are to protect the young minds being polluted by the

---

devices

[6] Web caching is the caching of web documents (e.g., HTML pages, images) to reduce bandwidth usage, server load, and perceived lag. A web cache stores copies of documents passing through it; subsequent requests may be satisfied from the cache if certain conditions are met. It should not be confused with web archive, a site that

unwanted materials displayed on the web. Web proxy can be configured in such a way that it blocks websites on the local network from being viewed if it is set up as a transparent proxy. It can deny access to URLs (Uniform Resource Locator) specified in a blacklist, thus providing content filtering. Some web proxies reformat web pages for a specific purpose or audience, such as for cell phones and PDAs (Personal Digital Assistants).

1.6. Electronic Mail Service

There are many good advantages, if a home server can also function as electronic mail (e-mail) server. First of all it increases security as the emails reside in home server instead of off-site server. Home affairs or business secrets are intact within a home or office, not exposed to network attacks. A family can have a domain name for handling all the emails of the members. It is definitely faster than one having it outside. Since it is on the home network, email handling of sending and receiving bigger files will be faster. Another advantage of having email server incorporated into home server is that each member can have a bigger mail boxes for storage.

1.7. Domotics

Domotics is also called as Home Automation. Our work culture is changing day by day. So also our eating habits and life style. We do not have time to cook our own food. Either we go to restaurant or buy some ready made cooked frozen food from malls. Small and big restaurants offer fast food for the busy people. Gone are the days when women were at home cooking and doing all household works. Specially in the developed nations women are equal to men both in competence and professionalism. Every work that we do is being converted into $s. Human labour is becoming more and more expensive. Solution, therefore, to manage a home is to install a home automation system. To control home automation, we need a home server that can control and operate all household items and electrical / electronic devices. It can include centralized control of lighting, heating, airconditioning, ventilation, etc. It reduces man power, increases comfort and provides high security. As far as aged and physically and

---

keeps old versions of web pages.

mentally challenged people are concerned, home automation is a real blessing and provides increased quality of life.

1.8. Surveillence and Security

Security is a major concern all over the world. After 9/11 we feel that we no longer are safe as before. The mighty and most powerful nation is attacked by terrorists. “For if they (terrorists) do these things when the wood is green, what will happen when it is dry?” is the feeling of most developing economies. Therefore we need a home server not only for home automaion but to keep us safe and to guard us like watchman by 24x7. It should be able to handle surveillence cameras and do recording and reporting according to the configuration set by the home owners. There are CCTV (Closed Circuit Television) DVR (Digital Video Recorder), USB web cameras and Network Cameras are in place for surveillence.

Besides surveillance, access to the home is to be controlled by the home server. It can be done either by RFID card reader or Mag Stripe card reader. The door lock can be coonected to one of the swipe card uinit systems and no intruders would be able to succeed. And the home server should be configured in such a way that when someone breaks open the door, it should send an SMS text message to all the home owners as per the configuration. It also needs to send a snap shot of the photo of the unwanted intruder while he / she enters the gate depending upon where the camera is located.

1.9. Scheduling Family Activities

A home server can act like a secretary to the family members. Lists of programs a family is scheduled for a month or a year are to be maintained in a home server. It could act like a bulletin board and maintain family calendar regarding the important events of the family. It can also remind the members the important task of the day and display a to-do lists when the computer boots up or in the form of an email message. This way all the members are family are united in minds and hearts.

1.10. Game Server

Entertainment industry is one of the booming ones in the world. Among them, online gaming is the most attractive ones for the youths. It fascinates them specially when they have a high speed internet connectivity. For adults, too, it is one of the relaxing activities to wile away their free time. A home server can act like a game server and it can support online gaming. There are sigle player and multiplayer games are availabe in the market. A home server, hence, can be a wonderful game server for the family.

1.11. SMS (Short Message Service) Server

Mobile / Cell phones are most common even in the developing countries. SMS text messaging is the most widely used data application in the world, with 2.4 billion active users, or 74% of all mobile phone subscribers[7]. This saves the end user's time. Like mobile phones, a home server is capable of sending SMSs both locally and internationally. It can also send bulk SMSs to all the members of the family at one shot. It is economical when it comes to sending international SMSs.

1.12. Virus Scanner

Virus is one of the biggest nuicence for the internet users. Deadly virus attacks in the past had destroyed vital information of big companies. Home computers should not be vulnerable to such attacks and need to be protected from virus and network attacks. Virus scanning, therefore, has to be a part of home server's daily or routine activity. The virus database needs to be updated daily while the virus scan engine is to be done routinely.

1.13. Linux – the Open Source Operating System

Now is the time to ask a question: "If we want a home server to do all the above services for us (from 1.1 to 1.12), then who can do these jobs?" The answer is simple, "Only an OS can do

[7] http://en.wikipedia.org/wiki/SMS

it". The next question is, "What kind of OS that we need to choose? Windows? Mac? Unix? Linux?" The answer depends upon various factors both internal and external. Feasibility and finance also play a role in decision making. We want it cheaper and at the same time to be viable and secure. One of the best options is to choose an OS that is free and open source. For IHS, I have chosen the same – CentOS which is basically a Linux OS. It is not only free but community supported. It can do everything that we discussed above and much more. It can be a file server, email server, proxy server, web server, game server, media server, SMS server, DNS server, DHCP server, virus scanner, gateway, etc,. All in one. It is also capable of handling network cameras, access and attendance, and do surveillance for save living, besides doing a complete home automation. Isn't it awesome?

CHAPTER 2

2. Computer Network for Home

While constructing computer network for home, first and foremost we need to be aware of different computers with various operating systems being used by the family members. One may use an apple PC with MAC OS, another may use open source like Linux, and third one may use Windows ranging from Windows 98 to Windows 7, fourth one may use SUN Micro systems, and the rest may just use i-pods or i-phones or PDAs. Different OSs or application softwares create different files with the extensions pertaining to them. Home Network, therefore, should not conflict with the operating systems and other application software, and at the same time should server the purpose and files. Broadly, we may think of the following points:

- how many computers we want to network
- whether we have any mobile computers at home
- the operating system(s) running on our home computers
- the type of Internet service we have at home
- what we plan to use your home network for
- where networked computers drive will be located inside the home

The second aspect we need to be concerned about is ‘centralized storage’. Each individual has both personal and family files. But all the family files are scattered all over. Say for example, a file is to be accessed only by two people or family photos are to be accessed by all. So there is a need for a storage system from where all the family members can access their own and family files as well. Each file can be assigned with Read/Write permissions. NAS should be serving its files to mobile phones, desktop computers, laptops, media players both in home network and remotely through internet.

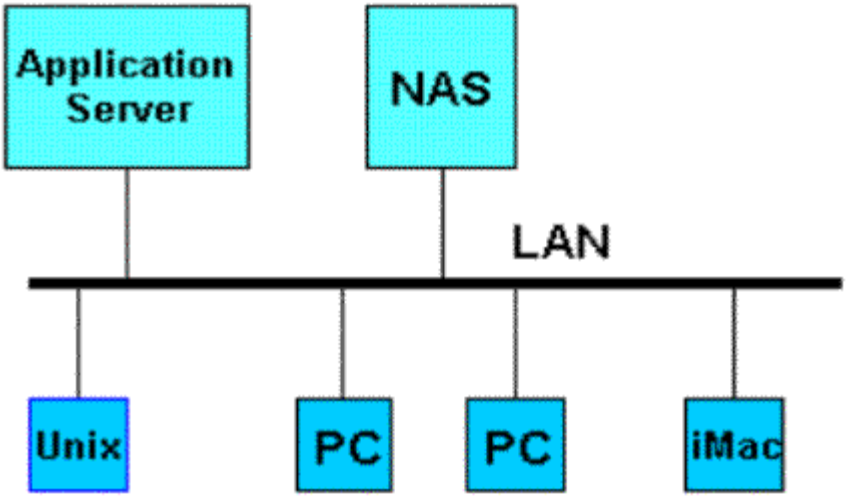


Fig. 2.1. Structural Diagram for NAS.

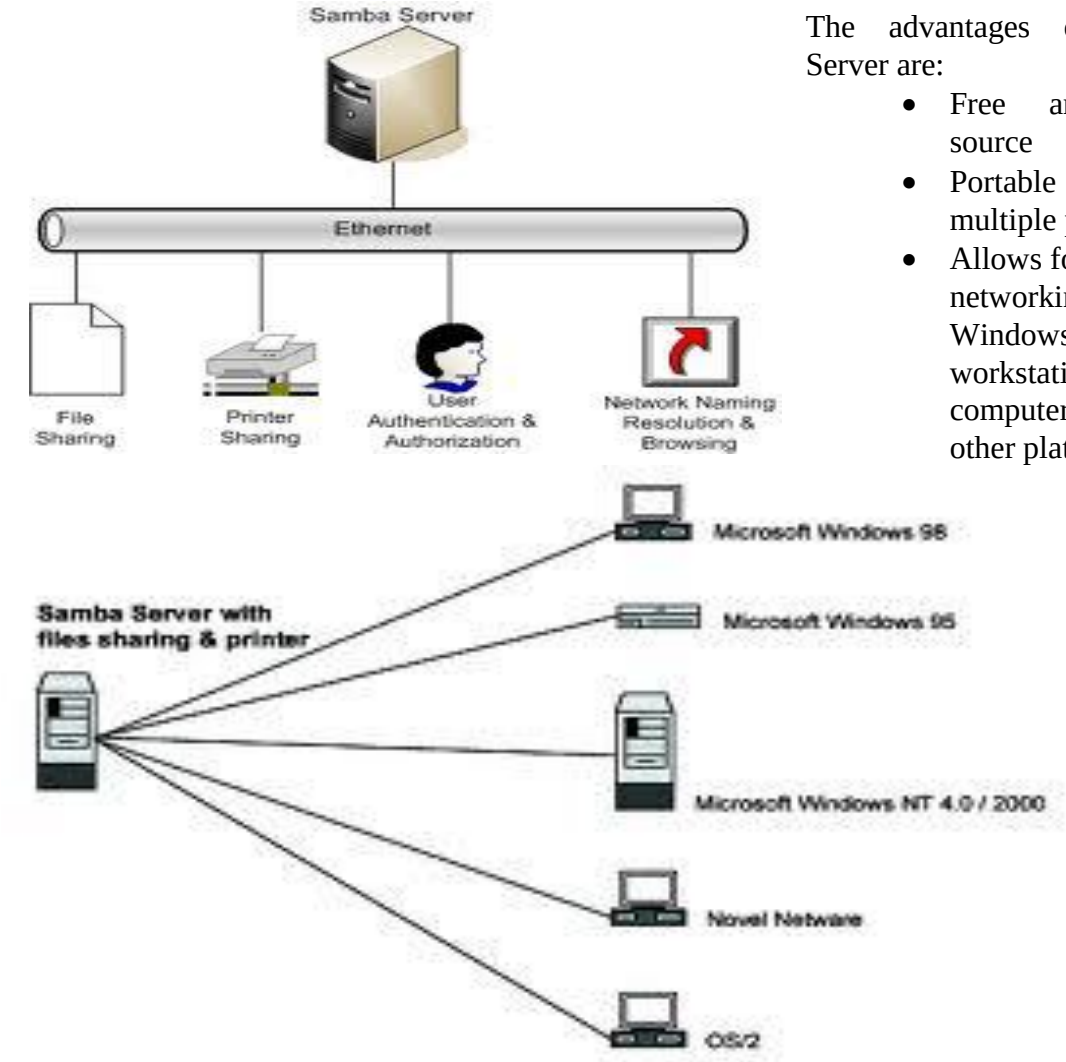
Some of the advantages of NAS are:

- Users running different types of machines (PC, Apple iMac, SUN etc.) and running different types of operating systems (Windows, Unix, Mac OS, SUN, etc.) can share files.
- Centralized storage, which makes it easier and cheaper to maintain, backup, and administer (comparable to DAS (Direct Attached Storage). Incidentally, centralized storage is more expensive than local disks on byte cost basis, but users have to do tasks such as backups and restores on their own.
- Separates purchase of storage from the purchase of application servers.
- Fast response times for users since NAS are on LANs, close to the users, as opposed to being on a backbone SAN (Storage Area Networks), marginally faster than DAS, but slower than a local disk.

Since NAS stores all the files – voice, video and data – there is a need for a network media player for a family to play them on TV or show the family photos when visitors come. Besides being accessing these files from networks, they should be also accessible remotely through web browser. Now let us analyze how file sharing, media serving, network games are done with the help of NAS and media players. Finally we shall see how these applications are implemented in IHS smoothly.

2.1. File Sharing

The most important aspect of file sharing is that it must be Samba[8] enabled. Samba server provides file sharing for all the OSs, provides print services, authentication and authorization, network naming resolution & browsing. Samba is a powerful and popular server today as it can cater to all the devices with different OSs.



The advantages of Samba Server are:

- Free and open-source
- Portable across multiple platforms
- Allows for easier networking of Windows workstations to computers running other platforms

Fig. 2.2. Structural Diagram for Samba Server

[8] Samba is an open source free software that can be run on a platform other than Microsoft Windows, for example, UNIX, Linux, IBM System 390, OpenVMS, and other operating systems. Samba uses the TCP/IP protocol that is installed on the host server.

2.2. Media Serving

We do not want to duplicate files in all the home computers. NAS solves this problem and can work as media servers, too. Gone are the days when people used to pile up movie CDs and DVDs on the racks. Nowadays all the movies are sold as softcopies. All that we need is a network media player which is inexpensive to play them. It can have access to PC media content or the streaming / downloading of audio, video and still image directly from the internet or NAS drive that can be played on our TV. Here is a sample as to how it looks.

As we know audio and video content are easily available via internet. Network Media Player that can be connected to either TV or HDTV or home theater system helps us integrate internet with NAS that has all the media files.

Fig. 2.3. Network Media Player

2.3. Network Games

One of the most enjoyable things we can entertain ourselves at home with our / other family members is to play connected games. It could be Lan games or online games. As there is already a NAS drive in the home network, it makes much easier.

There are different types in Network Games. Single player PC games run only on one computer whereas multiplayer LAN (Local Area Network) Games or online (internet) games can run in several computers. But each computer must run its own copy of the game. Online games allow us to play live across a high-speed internet connection. We require at least 1 Mbps broad band connection.

PC multi-player games typically work across any wired or wireless home network, but

experienced gamers may prefer to use wired Ethernet connections for local network gaming as it could offer steady speed LAN connections. Besides, wired Ethernet connections give reliable network. Today all modern game consoles also contain built-in Ethernet support for connecting to each other and to the internet. With a console we can also use wireless game adapters that convert its Ethernet connector to a Wi-Fi link suitable for connecting to wireless home routers. Both PC and console games benefit from having a fast internet connection when used online.

Fig. 2.4. Modern Network Game Graphics Screen

2.4. Implementation in IHS

We have so far seen three important aspects in home networks: first it should have a centralized storage which should be accessible both from locally (LAN) and remotely (via internet or web browser). Secondly, the centralized storage should be a Samba Server that can support all the OSs and network media players. Lastly, it should have a quota system as to how much space each user could use with read / write permissions. All these are well taken care of by ‘Gateman Lifestyle OS’.

1. Gateman’s Centralized storage:

- File access from Windows, MAC and Linux machines
- Files accessible remotely via web browser

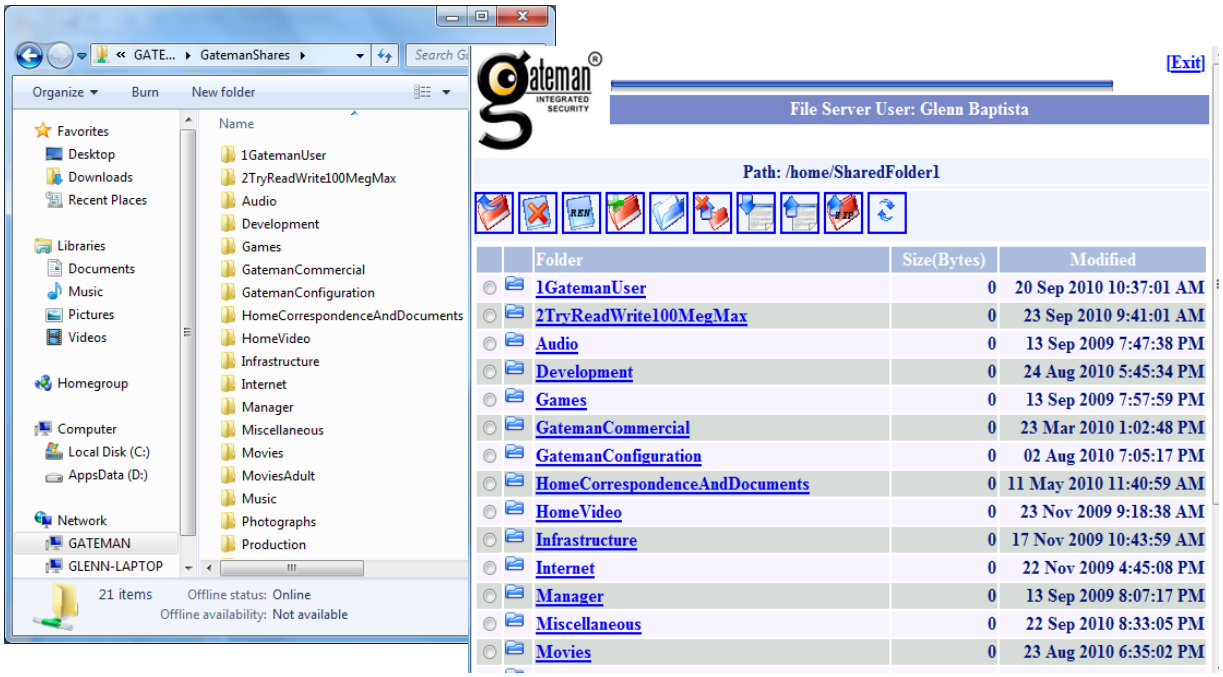


Fig. 2.5. IHS File Server Screen Shot.

2. Stores music, video and photographs centrally

- NAS box style access to Network Media Servers
- Media also accessible via Laptop and Mobile Phone

Fig. 2.6. IHS Serving Media Files

3. Remote Access via i-phones.

Fig. 2.7. IHS accessed through Mobile Phones.

It can easily be configured through templates which guide the user to set up the pictures within a predetermined form with minimum effort. To select the appropriate icon the user may browse the required directories and select the suitable icon that is also shown for the user's convenience in the 'File Chooser' window. Thus **IHS serves both files and purpose.**

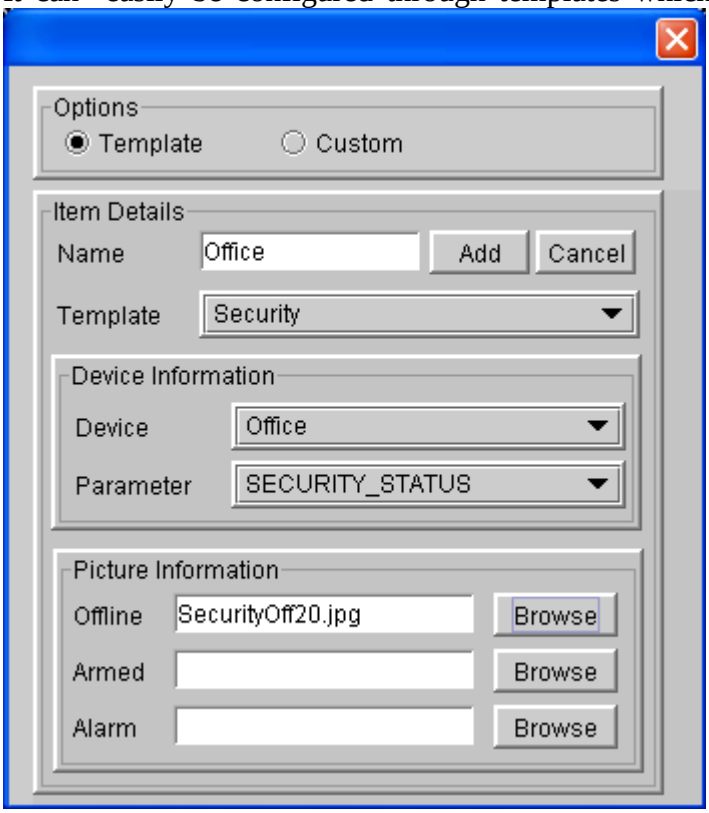


Fig. 2.8. Item Template Screen.

CHAPTER 3

3. Home Security System

What are the present home security systems are in place today? I can say in full confidence that there is no integrated system for a home. They are all segmented into bits and pieces. The total cost depends on number of modules one chooses as there are different plans available in the market. Most systems installed at home inform the base center in case of fire or burglary. It is called 'back to base monitoring system'. The other is surveillance for which CCTV or IP cameras are used. Let us analyze both with pros and cons.

3.1. Back to Base Monitored Security System

The back to base monitoring system is nothing but a centralized monitoring system. For ADT Security Company, its center is the boss and is the base. So all the security systems installed at homes will communicate to the center in case of burglary or fire. It is also possible that these systems alert police at the same time. These are called burglary monitoring services and fire monitory services and again these services cost on monthly basis depending upon the number of hours we ask for. If a family leaves abroad for vacation, it may require 24x7 services which would cost dear. It is not one time investment that you buy it and install it once and for all. Each month the family has to pay to the company that installs and monitors it.

In America one of the most famous companies that installs 'Back to Base Monitoring' Home Security System is ADT Security Company. When an intruder gets into a house, it automatically calls the police or informs ADT monitoring center. Mostly motion sensors are used for detection of movements. So also when the house on fire, heat sensors and smoke sensors connected to the system trigger the system to communicate to the base system.

3.2. Surveillance

For surveillance DVR systems are used for recording. They use CCTV cameras which are basically connected through coaxial cable. Some use IP cameras instead of CCTV. Let us see in detail.

3.2.1. DVR Systems

Sometimes the DVR (Digital Video Recorder) systems are also called PVR (Personal Video Recorder) systems as they record video in a digital format to a computer hard disk. Nowadays there are provisions to store the recorded data to USB memory stick or memory cards (MMC – Multi-Media Card, SD – Secure Digital Cards) and other mass storage devices. It includes set-top boxes, portable media players, camcorders for recording, memory cards for storing and software to coordinate all the devices. They use coaxial cables for connectivity. Hence the recording is analog not digital. DVR systems use CCTV cameras which cannot be rotated automatically. Manual adjustment is required to monitor the desired place. Again these DVR systems are no way connected to other electronic or electrical devices at home or home computers. It is an isolated system for surveillance.

3.2.2. IP Camera Systems

IP Camera Systems do exactly the same as that of DVR. The only difference is that IP cameras use CAT 5 or 6 or 7 (Category 5 or 6 or 7), whereas DVRs coaxial cable. Besides they use IP addresses to communicate to the base or router. It is more modern that CCTV IP cameras that are available in the market today can be rotated in different angles. There is no need of manually focusing on the place we want to watch over.

3.2.3. Drawbacks of the Present Home Security Systems

There are many drawbacks in the existing systems.

- It makes dependent on the security companies.
- It is not a comprehensive or an integrated system to serve a home or an office. It is not integrated to file server or mail server or access and attendance system.

- There are many false alarm calls. The customers are fined for each false alarm.
- CCTV is using co-axial cable for connectivity.
- Each month the family has to pay for the service. So it is a liability.
- It is an isolated system that does a small job and that is all.
- If the people at the ADT monitoring center are not alert, the house will be at the mercy of burglars or fire.
- It is an old and outdated system in this modern world. They cannot be operated or controlled by mobile or i-phones or i-pads.

3.3. Streaming

After having discussed the existing security systems in the market, we shall need to analyse the type of streaming used in those systems. Streaming refers to the delivery method of the medium rather than to the medium itself. Live streaming means taking the video and broadcasting it live over the TV or Monitor or Internet.

Let us fall back to 1990s. In those days watching videos and listening to music online was a torture. WWW was humorously defined as World Wide Wait. It was like stop-look-go kind of traffic. It was because the computers were slow and also the internet connection. There were no broad band connections with 2 Gbps as we have now. It was just 56 kbps or less. There used to be lots of noise on the telephone line. Because of the slow speed, most of the time we see 'buffering' on the screen rather than watching videos or listening music. Now the situation is different[9].

According to Bridge Ratings, 57 million people listen to Internet radio every week. In 2006, people watched more than a million streaming videos a day on YouTube. The same year, television network ABC started streaming its most popular TV shows over the Web. People who missed an episode of shows like "Lost" or "Grey's Anatomy" could catch up on the entire

[9] http://computer.howstuffworks.com/internet/basics/streaming-video-and-audio.htm

thing online -- legally and for free[10]. Isn't it amazing?

Nowadays the entire episodes of television shows streamed directly to the computers. In streaming video and audio, the traveling information is a stream of data from a server. The most important thing in streaming is that we need to take a look at how to make good streaming media files. It all depends on the protocols that we use.

3.3.1. RTP

The Real-time Transport Protocol (RTP) is one of the protocols that is used for streaming. It defines a standardized packet format for delivering audio and video over the Internet. It is used extensively in communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications and web-based push to talk features. For these it carries media streams controlled by H.323, MGCP (Media Gateway Control Protocol), Megaco, SCCP[11] (Skinny Call Control Protocol), or SIP (Session Initiation Protocol) signaling protocols, making it one of the technical foundations of the Voice over IP industry.

RTP is usually used in conjunction with the RTP Control Protocol (RTCP). While RTP carries the media streams (e.g., audio and video) or out-of-band events signaling (DTMF in separate payload type), RTCP is used to monitor transmission statistics and quality of service (QoS) information. When both protocols are used in conjunction, RTP is usually originated and received on even port numbers, whereas RTCP uses the next higher odd port number.

The RTP specification describes two sub-protocols:

1. The data transfer protocol, which deals with the transfer of real-time multimedia data. Information provided by this protocol include timestamps (for synchronization), sequence numbers (for packet loss detection) and the payload format which indicates the encoded format of the data.
2. The Real Time Control Protocol (RTCP) is used to specify Quality of Service (QoS)

[10] ibid.

[11] A VoIP (Voice over Internet Protocol) terminal control protocol defined by Cisco Systems, Inc.

feedback and synchronization between the media streams. The bandwidth of RTCP traffic compared to RTP is small, typically around 5%.

3.3.2. RTSP (Real Time Streaming Protocol)

It is one of the famous protocols for streaming. It is used to establish and control media sessions between end points. Clients of media servers issue VCR-like commands, such as record, forward, reverse, play and pause, to facilitate real-time control of playback of media files from the server. The transmission of streaming data itself is not a task of the RTSP protocol. Most RTSP servers use the Real-time Transport Protocol (RTP) for media stream delivery, however some vendors implement proprietary transport protocols. The RTSP server from RealNetworks, for example, also features RealNetworks' proprietary RDT[12] stream transport.

3.3.3. HTTP (Hyper Text Transfer Protocol)

HTTP is a request-response protocol standard for client-server computing. In HTTP, a web browser, for example, acts as a client, while an application running on a computer hosting the web site acts as a server. The client submits HTTP requests to the responding server by sending messages to it. The server, which stores content (or resources) such as HTML files and images, or generates such content on the fly, sends messages back to the client in response. In between the client and server there may be several intermediaries, such as proxies, web caches or gateways.

The difference between RTSP and HTTP is that RTSP adds new requests. While HTTP is stateless, RTSP is a stateful protocol. A session identifier is used to keep track of sessions when needed; thus, no permanent TCP connection is required. RTSP messages are sent from client to server, although some exceptions exist where the server will send to the client. Some typical HTTP requests, like the OPTIONS request, are also available. The default transport layer port number is 554, but one is not forced to use only this port number.

[12] **Real Data Transport** (**RDT**) is a proprietary transport protocol for the actual audio/video data, developed by RealNetworks in the 1990s.

Now having said about different protocols, one has to choose an appropriate protocol for streaming for a specific application. So that streaming is smooth and no 'buffering' appears on the screen.

3.4. File Formats and Compression

In this section we shall deal with the file formats and compression issues. One thing that we need to in mind while choosing file formats is that they are intrincially connected to the resolution, quality and file size. After having chosen the file formats and compressing the files, we need to transport them.

3.4.1. File Format

A file format is essentially a specific way that information is encorded for storage. We know computer operates on binary system – 0s and 1s. The storage system, therefore, would store in bits or bytes. Formats depend upon the kind of information (important or not so important or absolutely important) we want to store and the purpose for using them. Video files are based on frame rate, i.e., the number of still pictures per unit time. If the number of frames are more, the resolution will be good and so also the flow.

The Present Home Security Systems store audio and video formats. For example, Ping files store bitmapped images using lossless data compression, while ogg[13] format can act as a container for many different types of multimedia, including any combination of audio and / or video, with or without text and metadata.[14] On the other hand a text file can contain any stream of characters encolded as ASCII (American Standard Code for Information

[13] **Ogg** is a free, open standard container format maintained by the Xiph.Org Foundation. The creators of the Ogg format state that it is unrestricted by software patents and is designed to provide for efficient streaming and manipulation of high quality digital multimedia.

[14] Metadata is defined as data providing information about one or more other pieces of data, such as means of creation of the data, purpose of the data, time and date of creation, creator or author of data, placement on a network (electronic form) where the data was created, what standards used, etc,.

Interchange) or unicode. In the file fomat we have two types: uncompresed type and complressed type. Both have advantages and disadvantages. Let us study them.

3.4.2. Compressed Format

As the security systems use both audio and video recordings, we shall deal with both in compressed format. There are two kinds of compressed format: lossless compression and lossy compression. MPEG, WV (wave pack), etc., will come under lossless, whereas MP3, WMA (Windows Media Audio), AAC, ATRAC, etc., are considered to be lossy compression.

A lossless compressed format requires much more processing time than an uncompressed one, but is more efficient in space usage. Different systems use different formats depending upon the volumes of storage media and its value.

For video compression, varieties of methods have been used as they contain spatial and temporal redundancy, making uncompressed video streams extremely inefficient. Intraframe compression and interframe compression technolgoy are used to reduce spatial redundancy and temporal redundancy respectively. They are popular technicques for image compression. MPEG-4 is most common today.

3.4.3. Uncompressed Format

The uncompressed audio formats are CDA, WAV, AIFF, AU, raw header-less PCM, etc. These formats are meant to be more reliable than compressed. The AIFF format is based on the IFF format whereas WAV on RIFF format. BWF (Broadcast Wave Format) is mostly used in the television stations and film industries, not in the security systems at home or an office. The drawback of uncompressed formats is that they encode both sound and silence with the same number of bits per unit of time. A common uncompressed video format is RGB with 5 bits for the Red and Blue components and 6 bits for the Green component, for a total of 16 bits per pixel.

3.4.4. Transmission Format

Now comes the transmission type. Types of transmission line could include wires (ethernet cables), coaxial cables, dielectric slabs, striplines, optical fibers, electric power lines, and waveguides, radio frequency, infrared, blue tooth, etc,. Unfortunately the existing home security systems mostly use coaxial cables for communication. They do have light contoling system, but it is not wireless. There are some systems using UPB (Universal Powerline Bus) and again they are not an integrated one, rather isolated one.

3.4.5. What is preferred?

What is prefered? Today compressed formats are quite reliable and nearly perfect. MP3 or MP4 for audio and MPEG-4 for video are preferrable. As far as transmission is concerned, people do not like wires going here and there in the house. Wireless is prefered. Communication between security system and the devices controlled should use radio frequency waves or infrared. Back to base monotoring system is not a good system. It has created lots of confustion by sending false alarms, besides it makes people dependent on the company. Security system should be able to govern a home wirelessly and dependent on family members, not on outsiders.

3.4.6. Implementation in IHS

- Being a network centric server Gateman utilizes IP cameras to provide the user with surveillance capabilities. It has a driver layer through which it talks with various cameras. Gateman utilizes two techniques for surveillance;
- JPEG (Joint Photographic Experts Group) pictures allow clear sharp pictures and the frame rate can be adapted to suit the network bandwidth available
- Streaming video allows continuous motion and allows storage of audio and video

The simultaneous implementation of both these technologies allows users to be able to review video footage on fast or slow internet connections as well as from their mobile phones. Gateman stores JPEG pictures as snapshots with configurable frame storage rates. Triggers for storage are varied and user configurable and consist of time based, motion in region of

interest, as well as triggers based on IO including door or motion sensors, card swipe and an advanced combination of multiple triggers. For detection of motion within the Region of Interest, Gateman uses the 0^{th} and 1^{st} order Fourier coefficients and allows the user to select the threshold above which to trigger. This allows detection of edges as opposed to bitmap comparison allowing more realistic motion detection.

Gateman catalogs pictures within a database, enabling recall of pictures within an instant. This allows the user to review hours of video in just a few seconds and hence identify segments that he wishes to review in greater detail quickly.

In the streaming area Gateman is able to integrate Microsoft ASF[15] streams as well as RTP streams. It stores these streams as one minute MP4 or ASF files which can be downloaded for review by the user. This allows the user to review even video footage even on a slow internet connection.

3.5. Integration of Devices with IHS

Implementation of security and surveillance system for home or office implies the following tasks: selection of cameras and programming them, integration of an existing alarm system or a new one, and integrating to an email server and SMS server.

3.5.1. Selection of Cameras

There are a variety of network cameras available in the market using different technologies for compression and streaming. We need to write drivers to read picture and streaming data from the camera.

3.5.1.1. Foscam

This camera is very popular, has a variety of low cost indoor, outdoor and night vision models

[15] Advanced Systems Format (formerly Advanced Streaming Format, Active Streaming Format) is Microsoft's proprietary digital audio/digital video container format, especially meant for streaming media. ASF is part of the Windows Media framework.

and can be bought from hundreds of sources on the internet from USD 50/- onwards. It implements jpeg compression for picture data and passes audio and video data in an ASF stream. We have utilized this camera in our live demo.

3.5.1.2. Axis

Axis was one of the first manufacturers to implement network cameras and they have implemented more sophisticated compression technologies such as MPEG4 and H264. Therefore the network bandwidth and storage requirements for audio and video are reduced. They also have better image sensors and audio clarity. Axis cameras are available starting from USD 300/- onwards. They are available in a variety of models including indoor, outdoor, night vision, Pan / Tilt / Zoom models with a variety of housings.

3.5.1.3. A-soni

This manufacturer provides MPEG4 and H264 compression technologies at an economical price. A-soni cameras are available in a variety of models including indoor, outdoor, night vision, Pan / Tilt / Zoom models with a variety of housings.

3.5.2. Configuration of Foscam Cameras

Home Server integrates the IP Foscam Cameras that come with wired or wireless models. Some have night vision capabilities with long range infrared LEDs and waterproof. Configuration part is trivial as we shall see below.

The IP Foscam cameras have to be configured in such a way that we exploit the full potentialities of them. While configuring we need to take care of picture parameters, especially VGA picture size, the mode of operation (50Hz for 240Volt systems, 60Hz for 110Volt), wireless LAN settings and assigning IP addresses, etc.

1. Every camera comes with the driver CD that is window based and it is easy to install. A screen similar to the one below pop's up and lists the Foscam cameras available on the LAN. Double click to configure it.

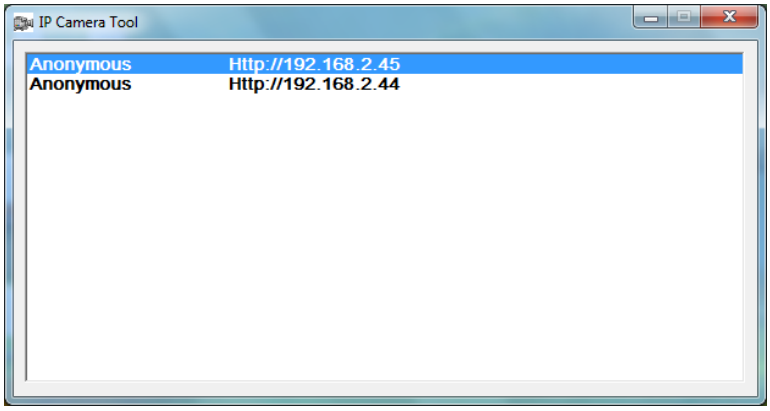


Fig. 3.1. Foscam Camera Configuration

2. Depending upon the browser that window uses, select the button appropriately. The username normally is '**admin**' and no password for **admin**.
3. Now select the 'Live Video' button from the LHS panel. Here we setup the resolution, tick on flip if camera is inverted, frequency based on indoor or outdoor.
4. From the Device Management screen select 'Basic Network Settings'

   Put the IP address, and other network parameters.
5. If camera is wireless, enter the settings appropriate to your wireless router.
6. Click the 'Users Settings' to add an admin password. Do not forget to enter the same password in the server configuration for this camera. That's it.

3.5.2.1. Advantage of Choosing Foscam FI8908 Model

1. It supports streaming; hence we can hear, record and replay full motion video streams with audio in .asf format
2. It has a digital input, so it could be integrated into all sensors such as door sensors, smoke sensors, motion sensors, gas sensors, glass break sensors etc.
3. It has a relay output that can be used to switch 12 volt devices up to 1 Amp or energize an external relay to control bigger loads and also to integrate with a remote key switch option on the alarm panel.

4. To use a sensor with FI8908, we must make sure that the sensor contacts are closed when it is in alarm state.
5. If we use closed door sensors, there will be no alarm when door is shut, and alarm gets triggered the moment the door is open.
6. In this project I have used closed door sensors. So I get an SMS and email with the pictures when the door is open. No message during safe state.
7. For smoke sensors, the sensor must be OPEN when there is no smoke, and closed for smoke.

Thus we need to keep in mind the type of sensors that we use and active state while configuring.

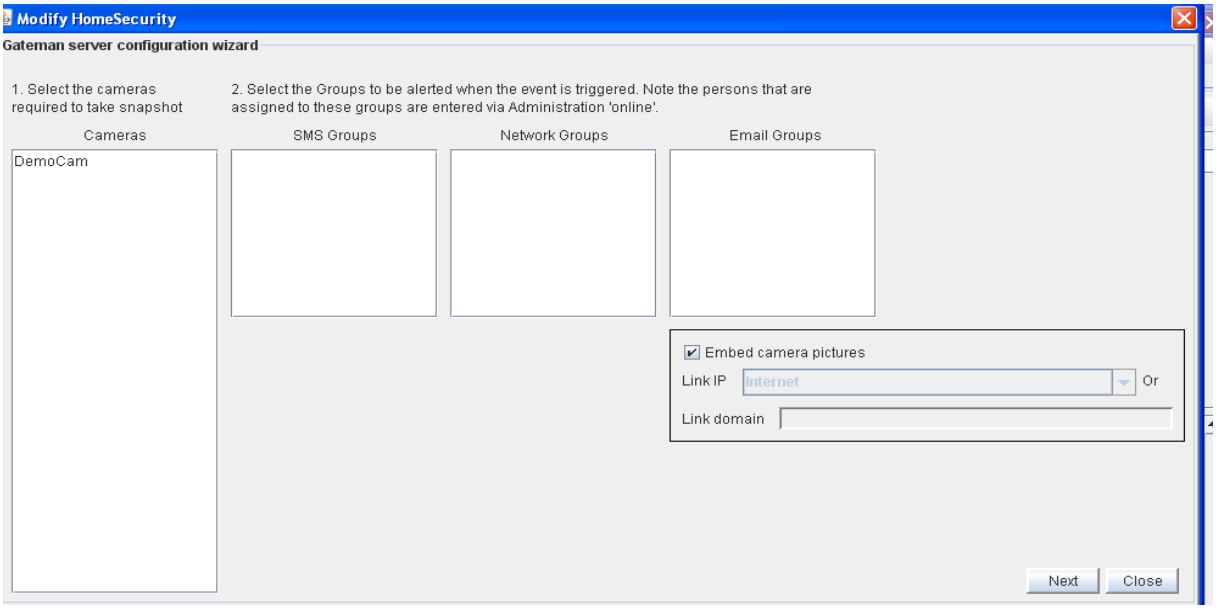
3.5.3. Security Alarm Integration

Most houses in the developed countries have equipped with some sort of security system like auto-dialing a number or back to base monitoring system. IHS can be integrated into it easily. By doing that we are not compromising the existing one. All that we need to do is to feed the output status of the alarm into IHS. That is all. The rest is IHS's job. It will text to all the family members, send an email with video footage, turn the lights on and even alert the fire and police department. There are two ways of doing it: First, take the spare 'alarm output relay' to an input sensor monitored by IHS. If there is only one alarm output, integrate a dual pole relay and feed one set of contacts to the siren or strobe, and the other to the desired sensor monitored by IHS.

The second way is get an optional output for the security system. Use these outputs as inputs to the input devices connected with IHS which can be configured to trigger additional alerts, start recording of cameras, etc,. If the security system provides a remote key-switch option, IHS can be used to arm and disarm the security system. Apart from these, if one provides with the protocol and documentation details of any security system under the Sun, IHS will be able to communicate through its own device drivers. In this project we have connected the input sensor to ICM (Input Control Model). The same is available in Foscam F18908, Axis 207,

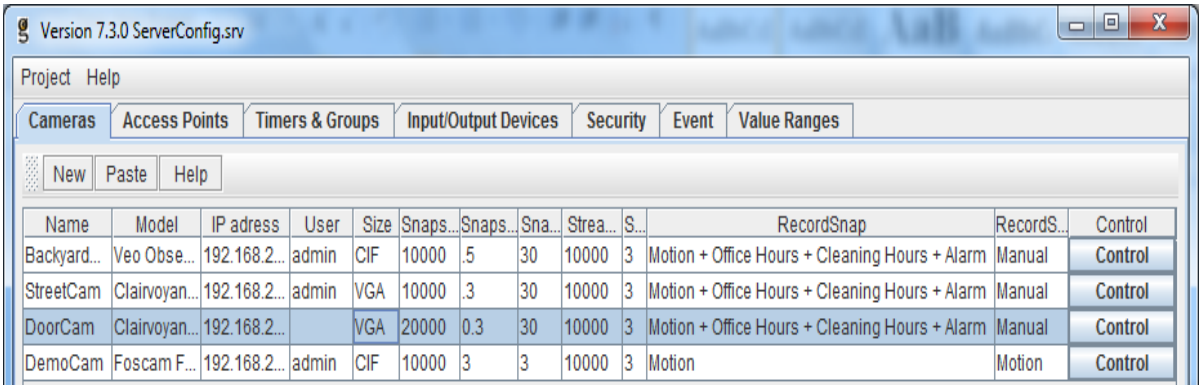
etc., which are high end models. We can use these input sensors to get the status of the existing security system.

As far as server configuration is concerned, first we need to create a SMS group, e-mail group and security network group as required within the 'timers & groups' tab to be notified of the

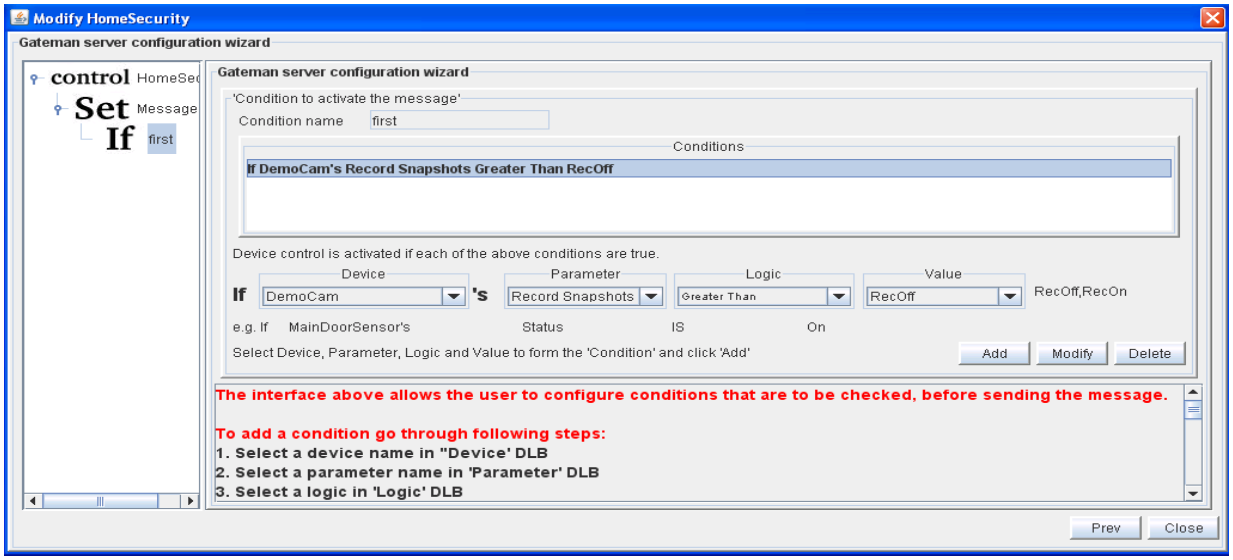


even. A screen would appear like this.

Secondly, we have to configure the cameras to record based on security alarm input. Lastly, we need to setup an event trigger within 'events' tab to notify via SMS or email. To change



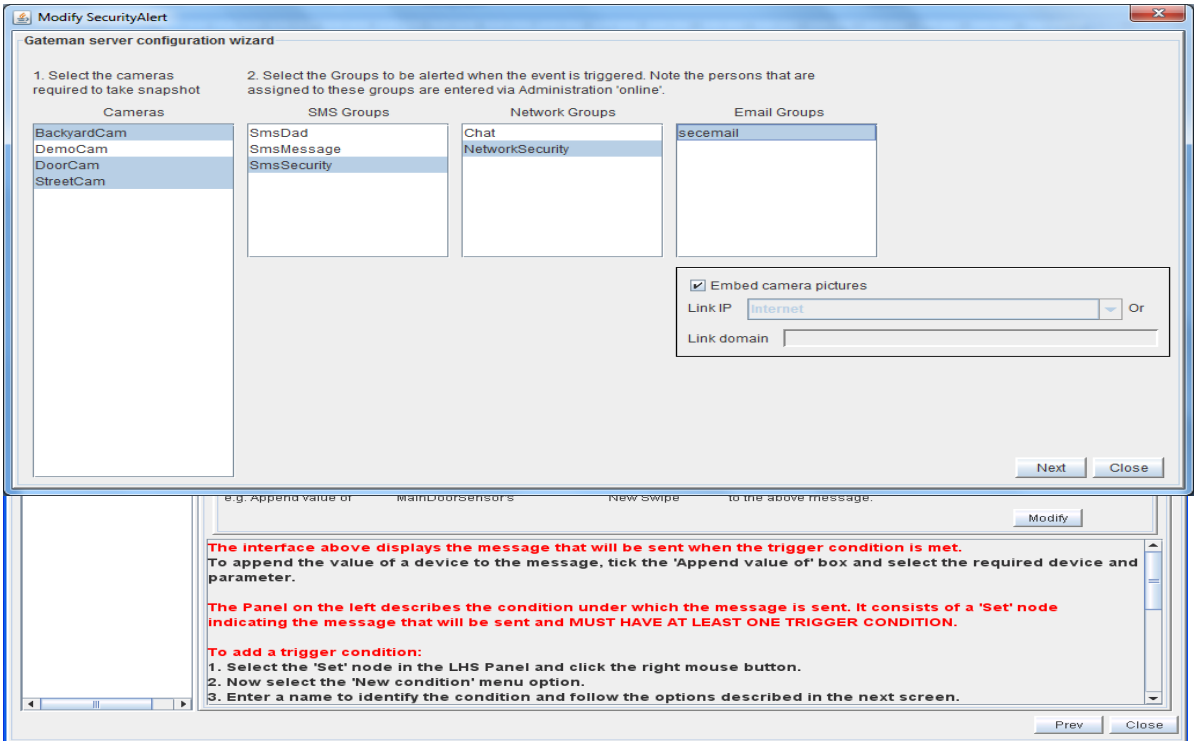
'camera configuration' a screen like this will appear.



Now comes the event configuration:

A screen like below will appear to modify alarm office node.

A trigger event can be configured using any number of conditions. In our project, for example,



IHS has been configured to take a snapshot whenever a door is open or record video when a

momvement occurs. IHS notifies the group members when the even occurs. Similar setup can be done for SMS server.

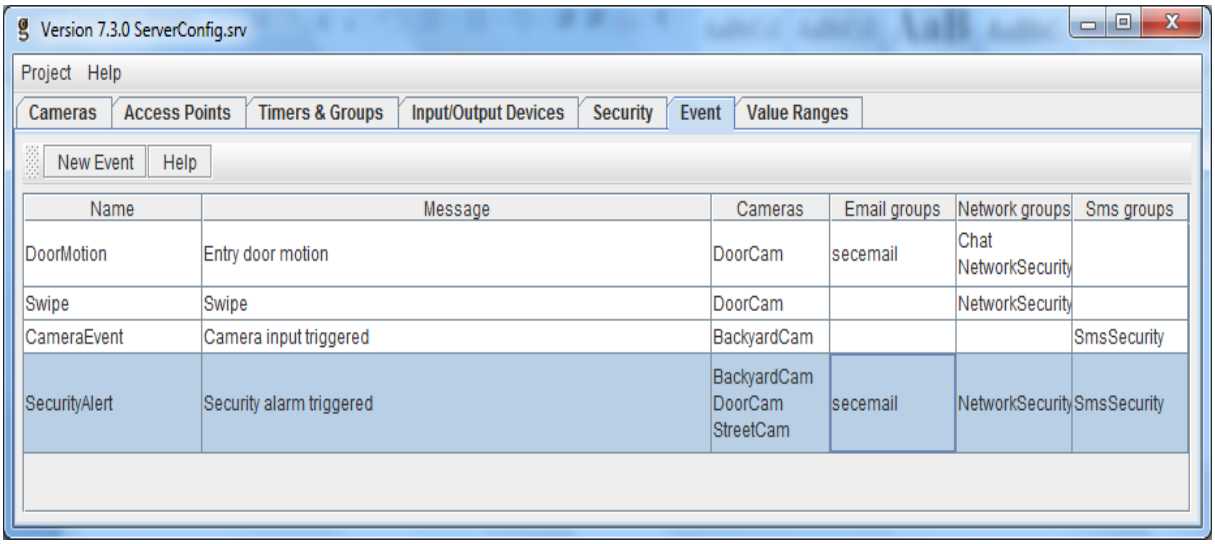


Fig. 3.3. Server Configuration Wizard: 1, 2, 3, 4, 5, 6.

3.5.4. SMS Server

First I purchased SMS credits from bulk sms from http://usa.bulksms.com/. This URL guides us to register and also SMS coverage locations.. According to me, it is one of the best options to choose from as it covers most of the countries in the world.

Basically this is going to be the server that is used by IHS to send text messages. It is, therefore, subscribed to the SMS gateway API. Bulk SMS is prescribed as provider ID 2. When we subscribe Bulk SMS with an account name, it gives us a password. So we need to enter as 2 (provider id), account name (johnrose), password (********), gateman (IHS). The only limitation is that it does not provide the sender identity feature. One has to understand from the message who the sender is or the message can contain the sender's name.

CHAPTER 4

4. Home Automation Systems

At this rate, we might like to define what is 'home automation' before we enter into different home automation systems. **Home automation** which is also called **domotics** designates an emerging practice of increased automation of household appliances and features in residential dwellings, particularly through electronic means that allow for things impracticable. This would include the automatic or semi-automatic control of lighting, doors and windows, Heating, Ventilation and Air Conditioning, and security and surveillance systems.

The techniques employed in home automation include those in building automation as well as the control of home entertainment systems, houseplant watering (sprinklers that we use for the lawn), pet feeding, changing the ambiance "scenes" for different events (such as dinners or parties), and the use of domestic robots.

Nowadays wireless systems are commonly installed because people do not like wires going here and there. These communicate via radio signals or infrared signals with a central controller.

Now having defined what a home automation is, let us glance at IHS that we are talking about. It is not that IHS is creating a completely a new system that is not available in the market. There are already systems available, but in bits and pieces. In every system, there is something missing. It cannot call it a complete integrated system that could integrate all the devices. Does a system in the market have a facility that can work as a file server, proxy server, mail server, access, attendance, security, surveillance, reporting, etc,? IHS integrates everything. IHS is not a home automation system, but an integrated system. Let us take some of the most popular ones one by one.

4.1. Existing Systems in the Market

When we type on Google for “Security Systems” and search for the links, we get dozens and dozens of systems available from market. Unfortunately, almost all of them are built or meant for windows OS. In this chapter we are taking some important ones that are most common and are somewhat integrated to other devices and appliances. I have taken 7 sample ones and analyzed them.

4.1.1. PowerHome2

PowerHome2 which supports X-10, Insteon and UPB (Universal Powerline Bus) protocols is an extremely powerful and flexible Windows based software package that will give us full control over our home’s lighting, appliances, infrared, audio and video devices when used in conjunction with the appropriate control hardware.

With PowerHome2’s programmable interface, this control can be achieved via keyboard, mouse, touch screen, web, e-mail, X-10, Insteon, UPB, IR, RS-232, voice recognition, socket communications, Windows Messaging, and even our internet enabled cell phone.

Demerits: It is Windows based, not Linux / kernel based. It cannot work as a file server or proxy server and besides, RFID cannot be connected to it for attendance system / security.

4.1.2. HAI Software

HAI software is for home automation and it is based on Windows Home Server. It allows user to monitor and control HAI control systems (not any systems that we use at home) via any device with a web browser. It is enhanced for Vista. No UBP protocol is used so wires running here and there connecting devices to the server is unavoidable.

Demerits: It is windows based, it cannot store files and not a web server or SMS server. We cannot call it as home server in a strict sense.

4.1.3. HS2 Home Automation

It is basically a software known in the market as HomeSeer HS2. It is being advertized as an advanced home automation that is designed to integrate the major systems of any home. It helps us control and monitor lighting, appliances, security, HVAC, telephone, home theater, etc., from one point. It has an amazing options to control the devices, namely through touch screens, wireless remotes, in-wall push-button controllers, by voice (mic and telephone) or MS Media Center. As it contains a built-in webserver, one can control the devices remotely through iphones, PDAs, PacketPCs and Laptops

It uses Universal Powerline Bus (UPB) technologies for total lighting and appliance control. It has an advantage that we can connect the existing security system to it. Since it is web-enabled, it alerts us by sending text message to mobile phones, emails and trigger any number of events within the home. Another additional feature is that it uses thermostat to contol HVAC[16] system. It works like an email server, so we can create message boxes for the whole family.

Demerits of HS2 is that its reporting facilities are minimal. It is neither a file server nor a proxy nor a DNS nor DHCP server. Besides it is vulnarable to cyber attacks as it is built for windows platform. It is costly for even a middle class family.

4.1.4. Control4

A Control4 system starts with a central controller that communicates with and controls other devices using a combination of Ethernet, RF (Radio Frequency), IR, RS-232 and ZigBee (IEEE 802.15.4) mesh networking technology. Control4 home automation claims that it can control everything with one remote and no new wiring required. With just one remote, we can automate and contol almost everything in our house, including home theater, music, security system, lights and thermostats. It does not provide file server, proxy server, mail server, and it does not talk about how the existing serveillance and security systems can be incorporated

[16] **HVAC** (Heating, Ventilating, and Air Conditioning) refers to technology of indoor or automotive environmental comfort.

into it. Poor documentation is provided for those who want to implement it at homes. Its price is exorbitant as it starts from $1,495 per system[17].

4.1.5. X10 Home Automation

Nobody can deny that X10 technology has been dominating the market for the past 2 decades, though it is considered to be an old system today. It is an international and open industry standard for communication among electronic and electrical devices used for home automation. X10 used to be almost equivalent to Domotics as it swept the market and pioneered in surveillance and security. It primarily uses powerline wiring for signaling and control. Signals involve radio frequency bursts. It is popular because it is inexpensive. Of late X10 software is using Debian open source system for configuration and control. There is a Debian package called Bottlerocket that will let us use a Firecracker to send X10 commands to modules from a Linux shell prompt.

The main demerit of X10 is that it is a one way protocol; so when we send an online command, there is no acknowledgement whether the command reached the device or not. Secondly, if someone turns on the device manually, we do not know about it. With one way communication, one is not sure of what is happening on the other side. Besides, there are no built-in noice filters. Some X10 controllers may not work well or at all with low power devices, say below 50 watts, or devices like fluorescent bulbs that do not present resistive loads. Besides, X10 protocol is slow and lack support for encryption and can only address 256 devices[18]. As I have said earlier, it is an old technology and it is not a server centric OS that can guard and serve a home.

4.1.6. Smart Home

Smarthome distributes SmartLabs' proprietary lighting design products, including a full line of lighting and appliance automation products enabled with INSTEON technology. INSTEON

[17] http://www.hometheatermag.com/accessories/206control4/
[18] http://en.wikipedia.org/wiki/X10_(industry_standard)

is a powerful, wireless home-control networking technology that simply, affordably and reliably integrates systems in the home for improved comfort, safety, convenience and value. It uses radio-frequency signals and a dual-mesh system that significantly increases signal reliability. We can say that INSTEON is one of the most reliable home area networking (HAN) technology. Like other technology, INSTEON can also control Heaters, Pumps, Appliances and Lightings. It is famous because it can communicate over a dual-mesh (using both RF and existing electrical wires in the home powerline.

Demierits of 'Smart Home' automation is that is guards a home but does not serve the purpose of a family. First of all it is not webbased and it cannot talk to the mobile phones. It is costly even for a middle class families. As the name stands, it is not 'smart' at all in terms of reporting to the home owners as to what is happening at home.

4.1.7. Gateman Home Server[19]

Gateman Home Server is basically built on CentOS, an open source Operating System. It is a server-centric and it can, therefore, work as an integrated system that could control all the modern gadgets and lifestyle products. Some of the salient features of 'Gateman' are;

- It is headless, so it can be contacted through web browser. Complete configuration and administration can be done through i-phones / i-pads and laptops.
- It is a file server like NAS (Network Attached Storage) and remotely accessible. It provides SAMBA[20] to access file and print services from MAC, Unix, Windows and Linux.
- It is integrated to the Network Media Player and serves media (voice, video, data) files.
- It is web server and web proxy which use squid and squid guard. Its extensive reporting at various levels and timings to the home or business owner is awesome.
- It is also a virus scanner (clam AV), uses spam assassin to filter Spams, DNS server, black lists unwanted sites and also a DHCP server.
- It can use UPB protocol, Schneider or Clipsal CBus or Allen Bradley PLCs or Modnet Protocols or Tibbo 1005 Ethernet I/O.
- This server centric OS can alert via text message, view live footage and replay stored

[19] www.mygateman.org

[20] Since 1992, Samba has provided secure, stable and fast file and print services for all clients using the SMB/CIFS protocol, such as all versions of DOS and Windows, OS/2, Linux and many others. Samba is an important component to seamlessly integrate Linux/Unix Servers and Desktops into Active Directory environments using the winbind daemon.

video by integrating a wide variety of network cameras.

- It can be said that 'Gateman' is most modern as it supports Twitters and talks to 4G i-phones / i-pads. It has a built in backup and restore mechanisms for disaster recovery.

4.2. Implementation of Domotics in IHS

In this project I have used the (UBP) Universal Powerline Bus to communicate with the devices like pump, lights and door sensor. This is because it is easy to integrate within the existing home as it can communicate on the existing power line and no special wiring was necessary.

4.2.1. Universal Powerline Bus

A. What is UPB?

Universal Powerline Bus (UPB) was originally developed by PCS Powerline Systems of Northridge, California and released in 1999. **UPB** is basically a protocol for communication among devices used for home automation. It uses power line wiring for signaling and control. Based on the X10[21] concept, UPB is considered to be one of the best home automation options on the market, due to its affordability, simplicity and transmission speed. In fact, UBP has an improved transmission rate and higher reliability. While X10 without specialty firewalls has a reported reliability of 70-80%, UPB reportedly has a reliability of more than 99%.

It is a highly reliable powerline wiring solution to control our home automation devices. As with all powerline protocols, the technology requires no new wiring, as it's designed to use our existing power wiring framework to signal messages. In the research lab, I have used the

[21] X10 is an international and open industry standard for communication among electronic devices used for home automation, also known as domotics. It primarily uses power line wiring for signaling and control, where the signals involve brief radio frequency bursts representing digital information. A wireless radio based protocol transport is also defined.

UPB for IHS with the existing powerline and wiring. In this respect, UPB is different from other wireless, radio frequency systems like Insteon[22] and Z-Wave[23].

B. How does UPB Communicate?

The UPB communication method consists of a series of precisely timed electrical pulses (called UPB Pulses) that are superimposed on top of the normal AC power waveform (sine wave). Receiving UPB devices can easily detect and analyze these UPB Pulses and pull out the encoded digital information from them.

UPB Pulses are generated by charging a capacitor to a high voltage and then discharging that capacitor's voltage into the powerline at a precise time. This quick discharging of the capacitor creates a large "spike" (or pulse) on the powerline that is easily detectable by receiving UPB devices wired large distances away on the same powerline.

C. UPB Protocol

While transmitting, one UPB Pulse is generated each half-cycle of the 60Hz AC electrical power cycle. The generation of each UPB Pulse is precisely timed to occur in one of four predefined positions in the half-cycle of the AC powerline. The position of each UPB Pulse determines its value as either 0 or 1 or 2 or 3. This method of encoding data as a relative position of a pulse is a well-known and used method in digital communications known as Pulse Position Modulation (PPM). Since each UPB Pulse can encode two bits of digital information and there are 120 AC half-cycles per second (at 60Hz), UPB communication has

[22] **INSTEON** is a system for connecting lighting switches and loads without extra wiring, similar to the X10 standard, designed specifically to address the inherent limitations in the X10 standard but also to incorporate backward compatibility with X10. Insteon is designed to enable simple devices - such as light switches - to be networked together using the powerline and/or radio frequency (RF).
All Insteon devices are peers, meaning each device can transmit, receive, and repeat any message of the Insteon protocol, without requiring a master controller or routing software. The system is a dual-band mesh topology employing AC-power lines and a radio-frequency (RF) protocol to communicate with devices. It is a home automation networking technology designed by SmartLabs, Inc.

[23] **Z-Wave** is a proprietary wireless communications protocol designed for home automation, specifically to remote control applications in residential and light commercial environments. The technology uses a low-power RF radio embedded or retrofitted into home electronics devices and systems, such as lighting, home access control, entertainment systems and household appliances.

a raw speed of 240 bits per second. Although this speed isn't fast enough for doing high bandwidth applications it is perfectly adequate for doing command and control communication.

UPB Pulses are transmitted in a special region toward the end of the AC half-cycle known as the UPB Frame. This region was selected due to its relatively low noise characteristics and for other attributes that make it an optimum position for powerline communications. UPB Frames are synchronized to the low-to-high transition of the AC waveform (known as the AC zero-crossing point) such that one Frame starts T/Frame microseconds after the zero crossing and the other Frame starts 8,333 microseconds (one half-cycle at 60Hz) after the first one.

D. Limitations of UPB

The UPB system has overcome many of the signal and reliability problems that plague X10. However, because it remains a 100% based powerline carrier technology, it can be affected by various powerline devices, and it remains susceptible to problems encountered in the noisy powerline environment. This is what exactly happened when I was working in the research laboratory (Room Number 240 of Electrical Department Building) which is closely located to high-end electrical motors. I was not getting signals (green) to configure the PIM device, rather only noise (red). So I had to take it to Sobrato Resident Hall where I reside and assigned device IDs for each device and Network ID for the network.

4.2.2. Selection of Powerline Devices

IHS communicates with UPB devices through PIM (Powerline Interface Module). PIM is plugged into 110 volt and its serial interface is connected to IHS. In the market there are number of UPB devices available, namely, dimmable output units wherein we can control the brightness of light or speed of speed of fan, etc.; On/Off output units which have got only two status: 0 and 1; Input Sensors such as smoke sensors, door sensors, motion sensors, gas sensors, etc.; remote switches which are used for both dimmable and On/Off output units.

There are some pluggable modules available by USA manufacturers for USA power system of

110 Volt. This avoids the hazels of wiring.

1. Lamp Models which are dimmable.

Fig. 4.1. PIM Device

2. Relay output Modules which are used to provide ON / OFF controls for Lights, pumps, fans, computers, TVs, Projectors, Home Theaters, etc.

Fig. 4.2. OCM Device

3. Input Modules which are meant to be used to interface sensors like gas sensors, light sensors, motions sensors, smoke sensors, etc.
4. Remote Switches which allow remote control of UPB devices.

Fig. 4.3. Remote Control for UPB Devices.

4.2.3. Configuration of the Powerline Devices

Configuration of powerline devices may not at first sight look trivial. But there are enough user guides provided by the manufacturers. The URL link - http://pulseworx.com – gives an extensive explanation of how these devices can be setup. In this section we shall see how PIM modules, OCM modules and ICM modules are configured.

4.2.3.1. Powerline Interface Module (PIM)

PIM is a plug-in module designed to interface a host computer device to the powerline communications of the UPB[24]. PIM directly communicates to a host computer device via a RS-232 serial cable running at 4800 bps. It can be used to both transmit information onto the UPB and receive information from the UPB. To understand how PIM works, we need to know the UPB communication method, the UPB System Model and the UPB Message Structure.

[24] UPB (Universal Powerline Bus) communication is a method is to communicate command and control information on a standard 60 Hz AC (Alternating Current) electrical powerline.

A. Modes of Operation

PIM has two modes of operation: Pulse Mode and Message Mode. Pulse Mode gives more detailed information as to what is happening on the powerline whereas Message Mode is more suited to users who only want to deal with valid UPB messages. Here we follow the pulse mode.

Fig. 4.4. PIM with Serial Port

The general philosophy of the pulse mode is that the PIM not only informs the host of the value (0 or 1 or 2 or 3) of the UPB pulse, but it also informs the host of the relative strength of that UPB pulse.

B. Setting up the Modes of Operation

As Pulse Mode is being used in IHS, it can be enabled by writing #1 of setup register 0x70 to a logical 0. The beauty of PIM is that once it is set to a particular mode of operation it will stay in that mode even if power is removed. That is to say that the setup registers are non-volatile.

While receiving UPB messages, the PIM sends “UPB pulse reports” to inform the host whenever it has detected a valid UPB pulse. The host can use these reports to build them into a received UPB message and take action accordingly. It is up to the host device to validate the checksum and other fields of the received UPB communication packet. Whereas while transmitting UPB messages the PIM will send “UPB Pulse Reports” back to the host to indicate each UPB pulse that was transmitted and also to inform the host of the presence or absence of an ACK Pulse at the end.

C. PIM-To-Host Responses

PIM-To-Host Responses are serial messages that travel from the PIM to the Host to inform the Host of the status of the previous Host-To-PIM command.

PIM-To-Host Response	Syntax	Description
PIM Accept	‘P”A’<cr>	Used to inform the host that the PIM accepted the previous command.
PIM Busy	‘P”B’<cr>	Used to inform the Host that the PIM rejected the previous command because it was busy.
PIM Error	‘P”E’<cr>	Used to inform the Host that the PIM rejected the previous command because it found an error in it.
PIM register report	‘P”A’<cr>	Used to inform the Host of the contents of the PIM Registers previously requested by a “Read PIM Registers” command. RR = Starting Register Number. VV = Value(s)
ACK response	‘P”R’RRVV<cr>	Used to inform the Host that the UPB transmission is complete and an ACK pulse was received.
NAK pulse	‘P”N’<cr>	Used to inform the Host that the UPB transmission is complete and an ACK pulse was not received.

Table 4.1. Host-To-PIM Commands

4.2.3.2. Output Control Module (OCM)

The Output Control Module, is a high quality plug-in home automation isolated relay contact controller capable of transmitting and receiving digital commands and status over the existing

powerlines. The OCM provides two independent channels of general purpose isolated relay contact outputs to control motors, lights, or to produce contact closures for other control devices. Simply plug the OCM into any standard wall outlet and wire the device to the controlled relay. The OCM is designed in a white enclosure with two completely isolated Class 2 contacts – channel 1 and channel 2 - rated at 8 Amps each.

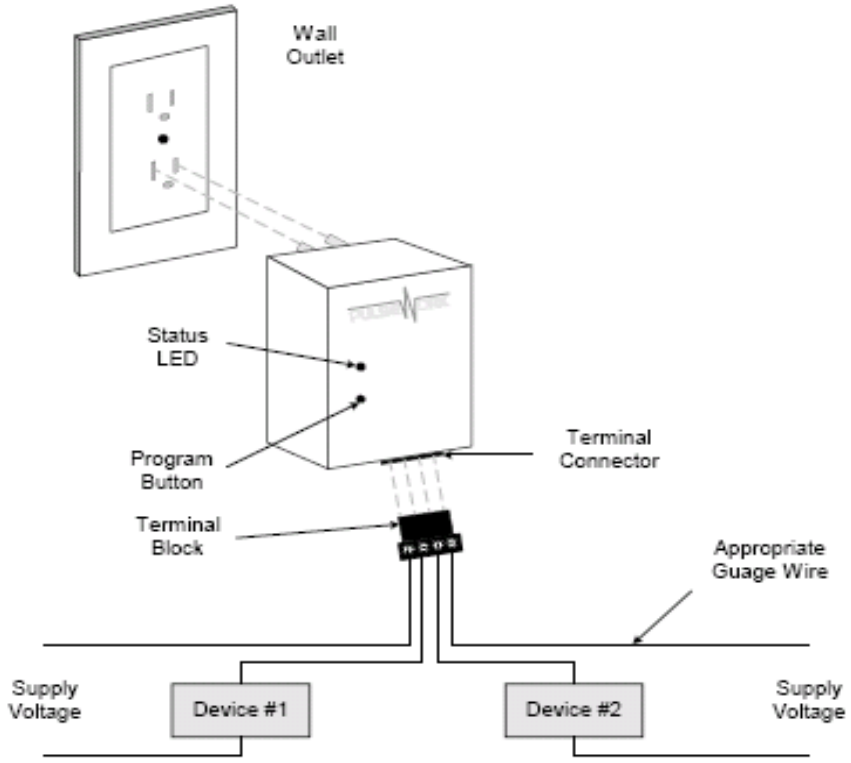


Fig. 4.5. Structural Diagram for Output Control Model (OCM)

A. Usage in IHS:

The OCM is perfect for controlling valves, gates, garage doors, gas fireplaces, pumps, a large high voltage contactor, and anything that is controlled by a relay. In this project OCM is used to control a small submersible pump and 60 W bulb.

Fig: 4.6. Fountain with submercible pump

B. Configuration

Once the OCM is installed it can be configured either manually or with the **UPStart** Setup Software. Manual configuration can be used to add our OCM device into a UPB network, link it to controller buttons and change preset relay states. Although the factory default operation of the OCM is useful in many situations, it is highly recommended that our device be configured with the UPStart Setup Software so that we can take advantage of its many configurable features. PCS (Powerline Control Systems) has developed a Powerline Interface Module (PIM) and free software (UPStart) to help us configure all of our Pulse Works Lighting System devices.

C. SETUP Mode

When configuring a UPB system, it will be necessary to place the OCM in SETUP mode. To do this, we have to press the Program Button **five** times rapidly. The Status LED will continuously blink Blue when the device is in SETUP mode. To exit SETUP mode, press the Program Button **twice** or wait five minutes for it to time out.

D. Operation

The Output Control Module (OCM) operates according to commands sent by UPB controllers. The OCM can accept powerline commands from any UPB-compatible transmitter such as PulseWorx Keypad Controllers, Timed Event Controllers, Wall Switches, Interface Modules, Approved Third-Party Controllers and touch screens. Each channel of the OCM can be a member of up to 16 scenes with the capability to store a pre-set relay state (OPEN or CLOSED) for each scene.

E. AUTO-OFF Timers

New for Generation 2 is the ability to set a maximum on time. If we forget to shut the load off

ourselves then the output module will do it for us automatically. The OCM can be configured to inform the rest of the network when it has automatically turned the load off. This allows the OCM to remotely control other devices or update feedback indicators such as LEDs.

F. TEST Mode

A manual test feature allows the relay to be CLOSED and OPENED locally. To enable the TEST mode we have to **press and hold the Program Button** on the OCM **for at least three seconds**. The Status LED will blink Magenta to indicate TEST mode. Relay #1 can now be CLOSED and OPENED by **single-tapping** the Program Button. Lamp #2 can be CLOSED and OPENED by **double-tapping** the Program Button. Press and hold the Program Button again for at least three seconds to exit from TEST mode.

G. Factory Default Settings

To restore the following default settings place the OCM into SETUP mode and then press the Program Button **ten** times rapidly. The Status LED will blink red to indicate that factory defaults have been restored. Then we have to press the Program Button **twice** more to stop the blinking.

4.2.3.3. Input Control Module (ICM)

The ICM Input Control Module is a high quality plug-in home automation controller that is capable of transmitting UPB digital commands over the existing power wiring. The ICM transmits preconfigured commands to turn ON and OFF (or blink) other UPB devices whenever a contact closure is made or a low voltage is present on any of its two separate inputs. The ICM can be freely located anywhere a wall outlet exists throughout the home. Only limited additional wiring is required and no radio frequency signals are used for communication.

A. Features:

- Senses on two separate contact closure inputs.
- Senses dry-contact switch closures or low voltage inputs.
- Command lights to flash when an alarm is tripped, a pressure mat is stood upon, a doorbell rings, etc.
- Ability to remotely enable/disable the input sensing mechanism.

B. Typical Uses:

The ICM is perfect for interfacing off-the-shelf motion sensors, light sensors, occupancy sensors, pressure mats, RF remote controls, pushbuttons, etc. to the PulseWorx Lighting Control System to trigger actions such as:

• Flash a light for the hearing impaired

• Turn off a vacuum or appliance

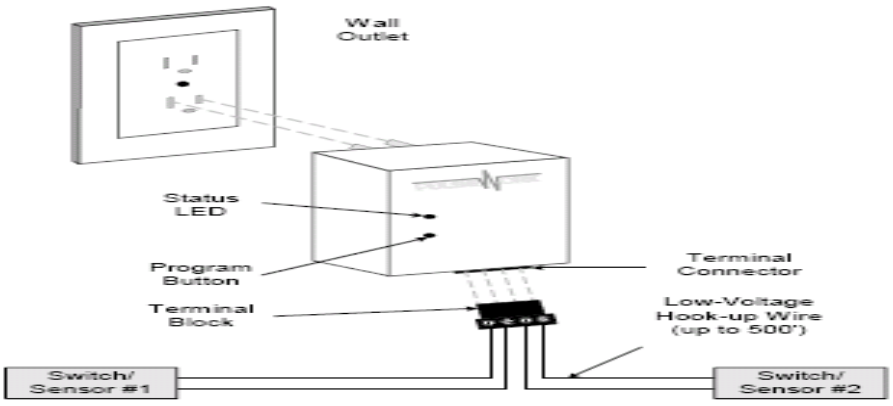
• Automatically turn on porch lights

• Automatically turn on security cameras

C. Input Sensing Modes

Each of the ICM's two input channels (channel 1 and 2) can be configured for one of two different input sensing modes: contact closure or low voltage. The two jumpers under the Terminal Connector should be positioned to select the desired mode.

Fig. 4.7. Structural Diagram for Input Control Model (ICM)



D. Installation

The ICM is designed for indoor use. To install the ICM module:

1. We have to locate any free grounded wall outlet throughout the home. Plug the ICM into the wall outlet (see illustration).
2. Optionally, the ICM can be secured to the wall outlet by screwing the wall plate center screw through the ICM's mounting tab.
3. Remove the supplied Terminal Block from the ICM's Terminal Connector (see illustration).
4. Connect two wires from the first Switch (or Sensor) to the Terminal Block. Terminate the two wires to blocks 1 and 2 using the clamping screws provided.
5. Connect two wires from the second Switch (or Sensor) to the Terminal Block. Terminate the two wires to blocks 3 and 4 using the clamping screws provided.
6. Plug the Terminal Block into the Terminal Connector (see illustration).

E. Configuration

Similar to OCM, PCS has developed a Powerline Interface Module (PIM) and free software (UPStart) to help us configure all of our PulseWorx Lighting / electric System devices.

4.2.3.4. UPStart Configuration

Follow the steps described below to configure the ICM:

Step 1: Add the ICM to the given UPB Network

Add the ICM to the UPB Network by selecting the **Device Add** menu item in the UPStart Setup Software. UPStart will find the ICM and allow us to name / rename it.

Step 2: Start ICM Configuration

Double-click the ICM icon to begin configuration. Select the **Transmit Components** tab to begin configuring what to control.

Step 3: Specify the Link(s) to Control

On the **Transmit Components** tab, select one link to be controlled for each channel.

Step 4: Specify the Sensing Mode

On the **Transmit Components** tab, press the Mode button and select a Sense Mode that we desire.

Step 5: Specify a Suspend Link (optional)

A sense channel can be configured to be enabled (active) or disabled (suspended) by a UPB Link command received by the ICM. To configure a sense channel to be controlled by a Link, select the **Receive Components** tab and specify the Link to control the sensing channel. The channel will be suspended when a DEACTIVATE (or GOTO 0%) command is received for the specified link and activated when an ACTIVATE or (GOTO > 0%) command is received.

Step 6: Program the ICM

Once we have specified the desired configuration, press the Program Device button to program the information into the ICM.

SETUP Mode

When configuring a UPB system, it will be necessary to place the ICM into SETUP mode. To do this, we have to press the Program Button **five** times rapidly. The Status LED will continuously blink Blue when the device is in SETUP mode. To exit SETUP mode, press the Program Button **twice** or wait five minutes for it to time out.

OPERATION

Once it is installed and configured, the ICM will operate on the stored configuration settings without further user intervention. When the doorbell switch is pressed the doorbell should ring and the ICM should transmit the configured UPB command onto the powerline. In IHS, I have connected a door sensor which communicates with the ‘Gateman’ whether the research lab door is open or closed.

Setting up UPB Devices:

To set up UPB devices, we use UPStart software. It is windows based application software that gives us the ability to easily and test our UPB devices. Using UPStart, we can unlock the hidden potential inside of our UPB devices to design a custom lighting and control system that

is tailored to our needs and desires.

Not only are we able to configure our lighting, pumps and other other control systems but, with UPStart's powerful test capabilities, we can also test our powerline for noise, measure communication signal strengths, and functionally test our UPB devices. Good signals are in green and noise in red or orange.

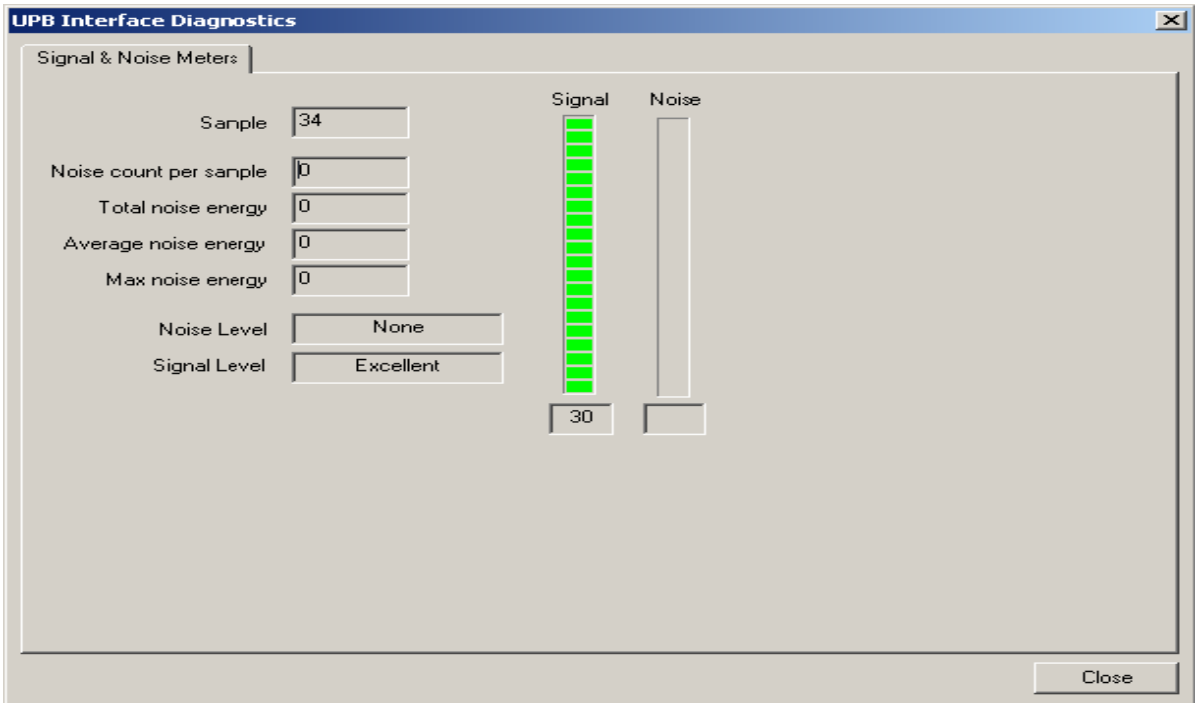


Fig. 4.8. UPB Interface Diagnostics.

All this gives us good confidence that once we are finished installing and configuring our UPB lighting and control system, we never have to worry about coming back later to fix or 'tweak' the system. Basically, UPStart is designed to interface to the powerline through a special device called a Powerline Interface Module (PIM). The PIM plugs into any standard electrical wall outlet and connects to our PC or laptop computer via either a serial or USB cable.

Connection PIM to PC

Step 1: In this project PIM is connected to the serial port of the PC. We need to specify on which com port it is connected.

Step 2: Test the PIM Communication path.

If it is successful, the window will appear as following.

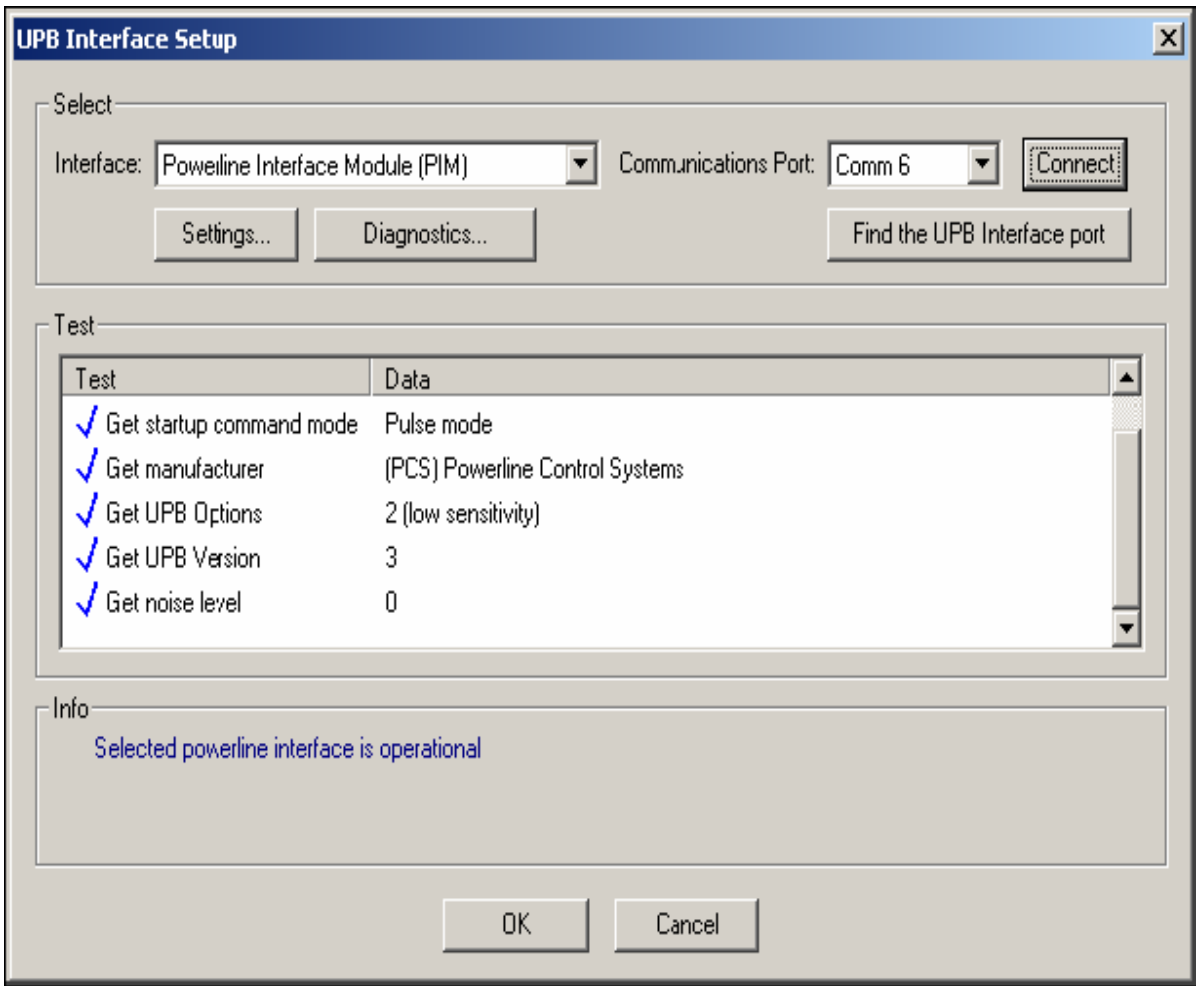


Fig. 4.9. UPB Interface Setup

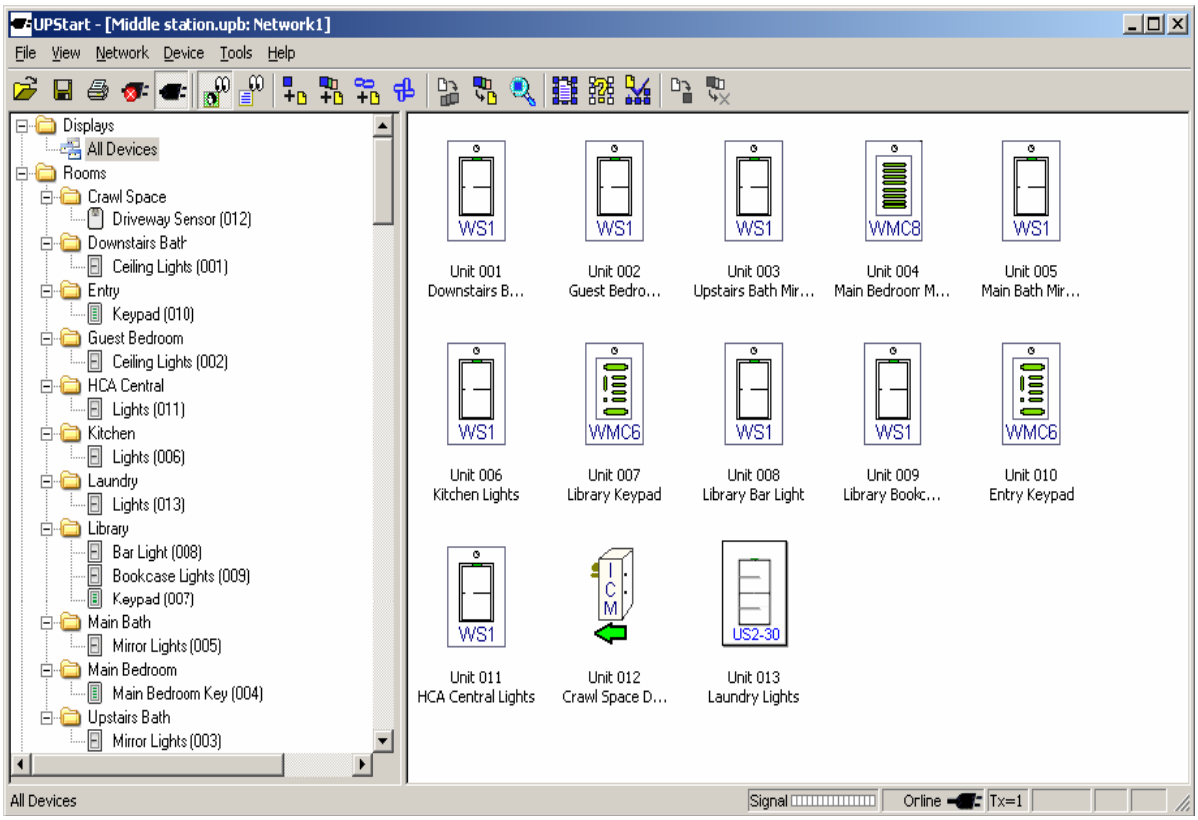


Fig. 4.10. Configuration for PIM (Powerline Interface Module) Unit.

Once we install UPStart software, we can start it by double clicking on the icon. We get the following window after configuration of PIM.

Design Pane on the left side shows us each room along with the devices placed, whereas Display Pane on the right side is the main window showing the network devices as well as icons for links.

When the PIM is connected to a PC and if there are good signals indicated by green color, UPStart checks the ID portion of the device memory to make sure it is working with the expected device. This displays as:

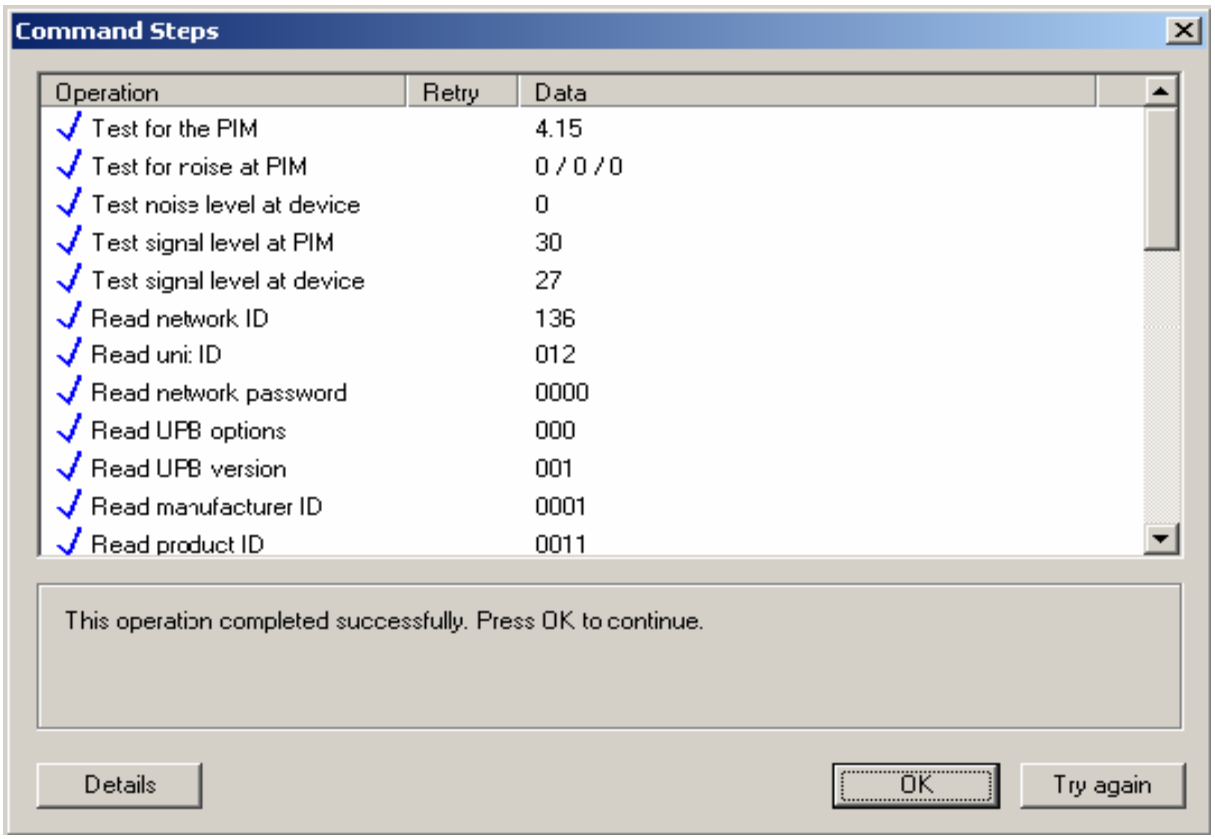


Fig. 4.11. Testing PIM

Then the operation continues with reading or writing device memory. UPStart reads it back after data is written just to make sure that data is written into the device memory. The following window appears once it is done successfully.

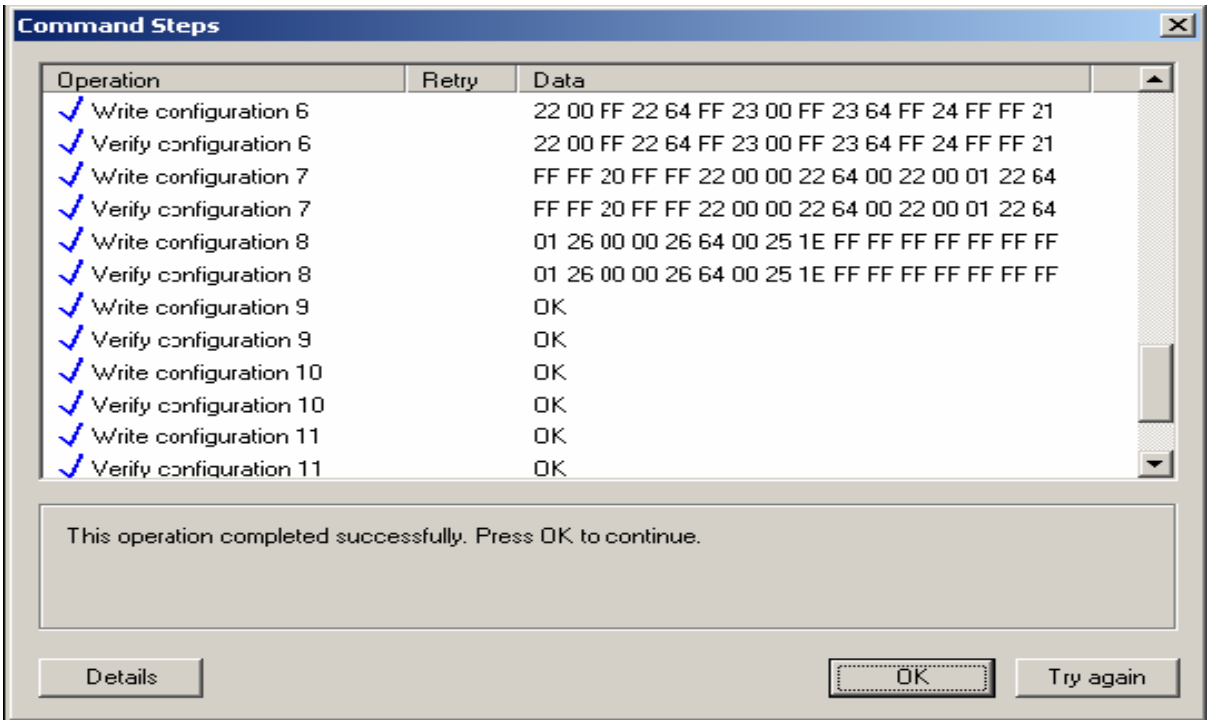


***Fig. 4.12. UPStart checks the ID portion of the* Device Memory**

Creating a UPB Network:

To create a UPB network, we need to give an ID and name for the network. The network ID is a unique number that can be given ranging from 1 to 250. This ID is stored in each of the UPB device. Similarly, network name that consists of 16 characters is also stored in the UPB devices.

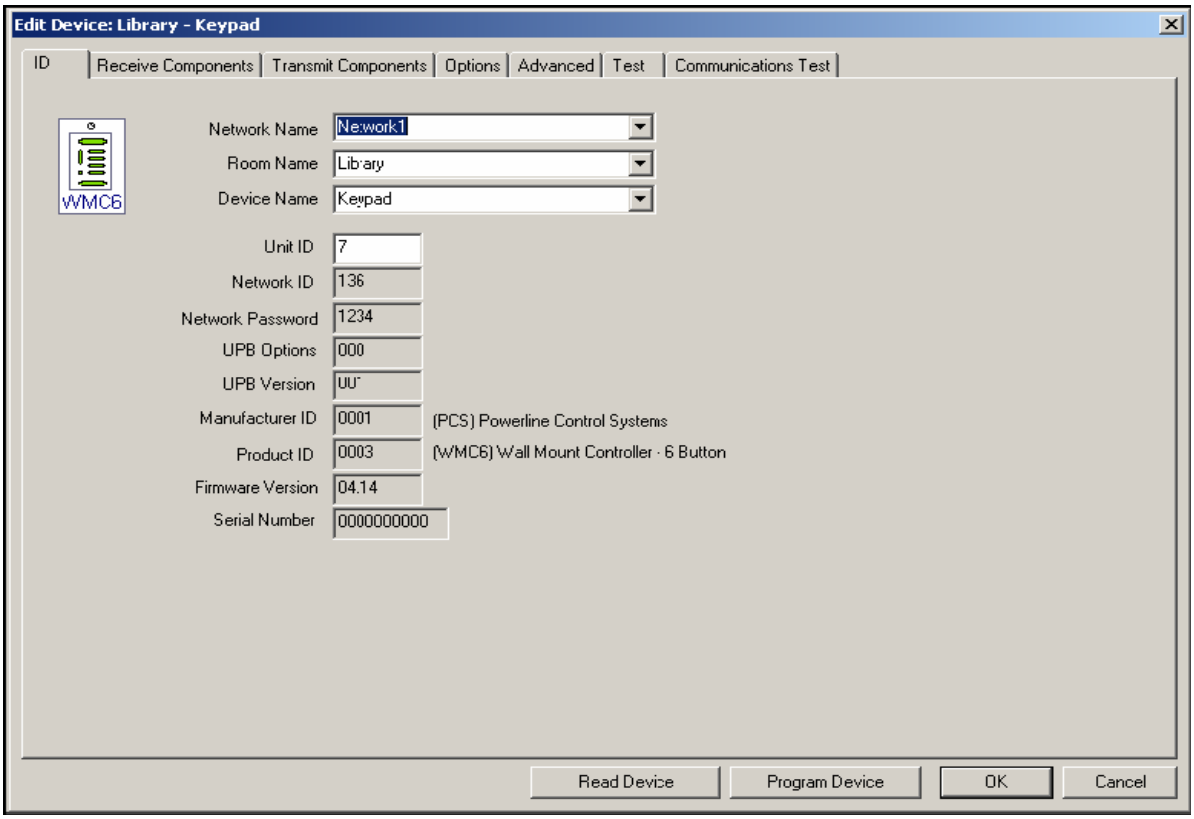


Fig. 4.13. Configuring an UPB Network

Once the device is located the network parameters are read from it and using this information UPStart recognizes all the devices as shown below.

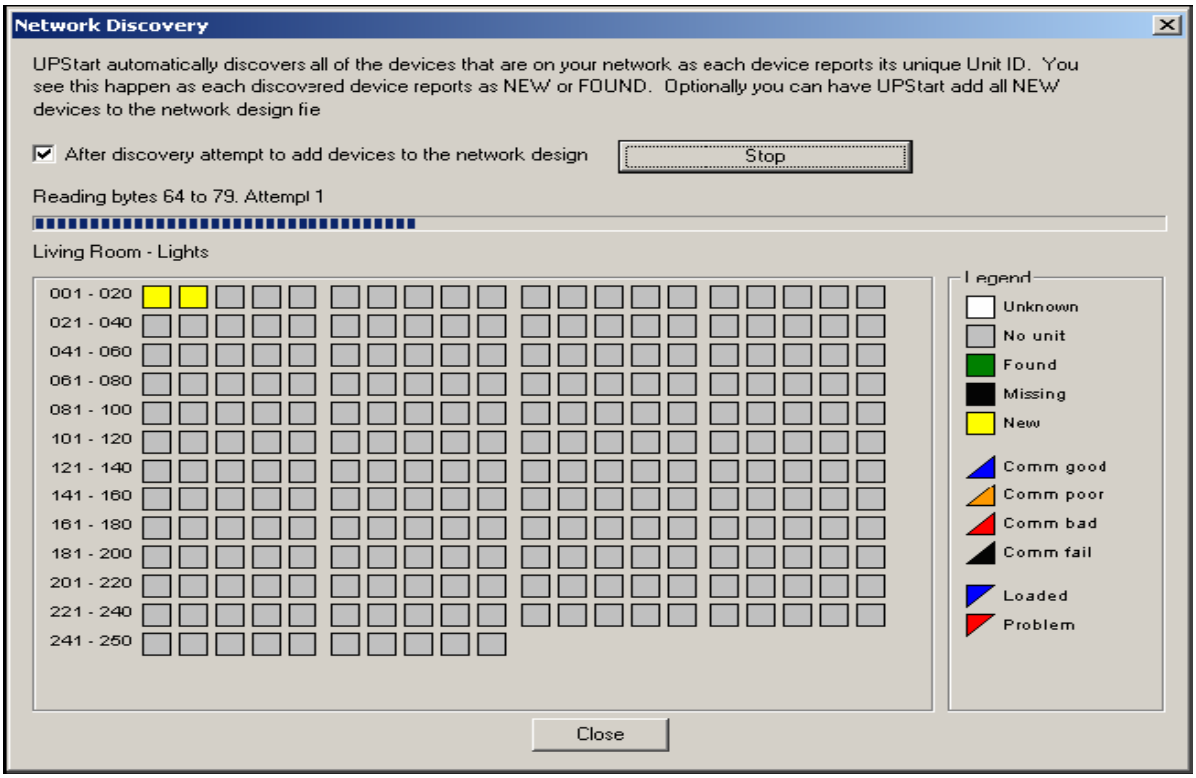


Fig. 4.14. UPStart Recognizes the Devices Configured.

4.2.3.5. PIM Configuration and UPB Imlementation

In this chapter it is described how PC talks to PIM through serial cable and how UPB has been implemented in IHS. Basically IHS communicates only to PIM and as a result PIM dictates terms to ICM and OCM as to what action to be taken based on the instructions received from IHS. Brays Terminal software is used for configuration. The implementation of UPB protocol by IHS will be based on the following specifications.

a. IHS uses message reports, not pulse.
b. Within the Driver parameters, Network ID is configured.
c. The default number of repeats is 0. It can range from 0 to 3.
d. The default SourceID is 255, but is also configurable.
e. IHS uses direct addresses as link addresses are meant for multiple devices.
f. The Control Word will be 0rrl llll 0100 0000. Bit 15=0 means that direct addressing is used. rr=Retransmits and retries will be controlled by the retries within Gateman driver; hence rr ==00. llll is a 5 bit packet length (6 to 24). Bits 7 to 0 must be set as shown.
g. Each device will have a Direct Address and optional channel number. 0 acts on all the channels. Channel numbers are used for multi channel devices.

h. IHS uses the GOTO command to write a value. The Rate will be set to 0. E.g. 0x22 LL 0x00 CC where LL is the desired level between 0 and 100 and CC is the channel number.

i. IHS uses the REPORT STATE command to read the value of a device. The response of the Report State command is an ACK message followed by an unsolicited response of the form **0x86 VV <VV>…** UPTO 16 MORE VALUES one for each channel.

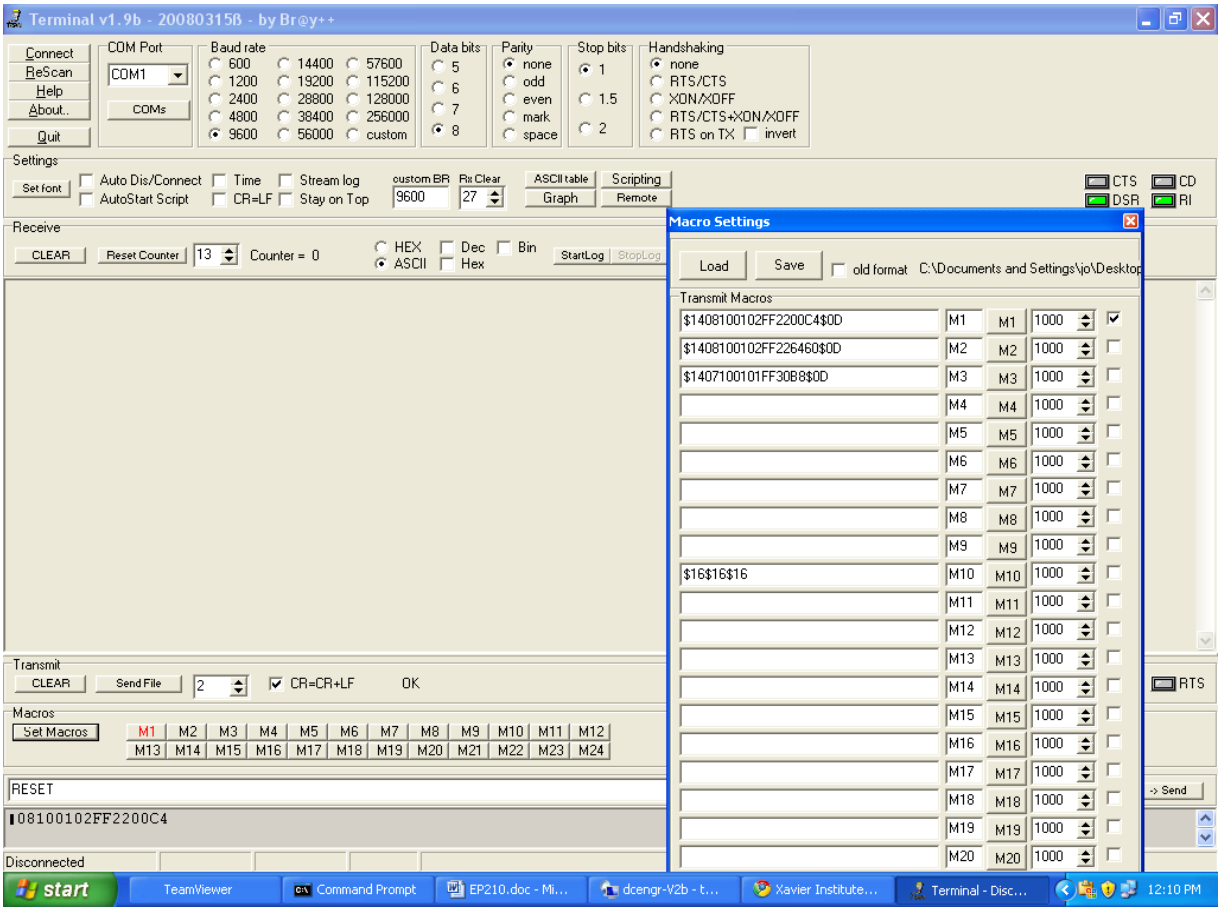


Fig. 4.15. PIM Configuration using Bray's Software

As shown in the above figure (using Bray's software), the DTR line must be green by the host when the driver starts. If not green, no communication is taking place. On the other hand, the PIM module must not have the 'Enable PIM disconnection watch' checked. The switch units must be programmed to report their status on switch press. To get a switch unit into 'Setup Mode' one must press the rocker switch 5 times in quick succession. It will then start blinking its blue light. The device is then configurable via the PIM.

The message format from Gateman to the PIM is of the form <Ctrl-T>UU….KK<CR>. Each byte within the packet between the <CTRL-T> and <CR> is sent as an ASCII HEX character.

The message includes a single byte checksum as 2 ASCII chars. Every UPB Communication Packet must end with a 1-byte field called the Packet Checksum. The Packet Checksum is used to verify the integrity of the received packet.

Channel - 0

08100102FF2200C4
08100102FF220ABA
08100102FF2214B0
08100102FF221EA6
08100102FF22289C
08100102FF223292
08100102FF223C88
08100102FF22467E
08100102FF225074
08100102FF225A6A
08100102FF226460

channel - 1

0A100102FF220AFF01B8
0A100102FF2214FF01AE
0A100102FF221EFF01A4
0A100102FF2228FF019A
0A100102FF2232FF0190
0A100102FF223CFF0186
0A100102FF2246FF017C
0A100102FF2250FF0172
0A100102FF225AFF0168
0A100102FF2264FF015E

How do we interpret the above codes? The packet checksum is computed as follows:

Sum all of the bytes of the Packet Header and UPB Message fields together STARTING FROM THE CONTROL WORD AND EXCLUDING THE HEADER CHAR (CTRL T). Then take the 2's complement of the sum and truncate the result to 8-bits. The message bytes are as follows:

- 2 byte control word 000l llll 0001 0000 ; llllll is the **number of bytes – this is counted from itself upto the checksum but not including the trailing CR**. 0001 => reply using ack pulse. 0810 FOR GOTO / 0710 FOR REPORT STATE
- Network id = Driver parameter
- Destination id = Device addressed
- Source id = PIM id – Driver parameter – default 255

- Message Data Id – 0x22 = GOTO; 0x30 = REPORT STATE
- Message Data Arguments – 0 to 18 bytes = LL (level) for GOTO
- Checksum

Description	**Write destina. 100 cmd**	**Write to channel**	**Write destin. 0 cmd**	**Report State**	**Report state multi-channel**	**Switch val update**
Header (1 char)	0x14	0x14	0x14	0x14	0x14	
Control (no of bytes)	08	0A	08	07	07	
Word (Ack pulse)	10	10	10	10	10	
Network Id	01	01	01	01	01	
Destination Id	01	01	01	01	01	
Source Id	FF	FF	FF	FF	FF	
Message Data Id	22	22	22	30	30	
Value	64	64	00	-	-	
Rate	-	FF	-			
Channel	-	NN	-			
Checksum-from ctrl word	61	XX	C5	B8	B8	
End of packet	0x0D	0x0D	0x0D	0x0D	0x0D	
Packet Accept	PA	PA	PA	PA	PA	

Packet Acknowledged	PK	PK	PK	PK	PK	
Packet				P	P	P
Unsolicited				U	U	U
Control (no of bytes)				08	08	08
Word (Ack pulse)				00	00	40
Network Id				01	01	01
Dest Id				FF	FF	00
Source Id				01	01	01
Message Data Id				86	86	86
Value				64	64	00
Value 2 (Multi-channel)				-	00	-
Value 3 (Multi-channel)				-	64	-
Checksum (from ctrl word)				0D	xx	6C
End of packet				0x0D	0x0D	0x0D

Table 4.2. Guidelines to Interpret the Codes

Driver Implementation:

a. On startup the driver will loop through all un-initialized points and issue the 'Report State' (0x30) command for each point. Multi-channel devices will report all devices in a single 'Device State Report' (0x86) message.
b. The driver will implement writes using the GOTO (0x22) command. This command will be used without any optional parameters for devices with a single channel (channel number = 0). For devices with multiple channels the rate will be 0xff and the channel number will be specified.
c. When a switch is pressed, the device will notify the driver using the 'Device State Report' (0x22)

"APPENDIX A" has the java program that was used to configure the PIM device.

4.2.4. EP210 (Magnetic Stripe Card Reader) Configuration

EP210 is small, low cost, general purpose data collection peripheral devices that communicates with a software application program over a TCP/IP network. It connected to the router via 10/100BaseT network interface and to the PC via serial cable.

Fig. 4.16. Magnetic Strip Card Reader with RJ-45 and Serial Port

When EP210 is connected to the network it sends data to and receives commands from a user program. Reader input is decoded and forwarded to the application program. Likewise, the application program can send data to or receive data from the serial port. The digital I/O features allow the software to control the relay outputs and query the counter inputs.

Network Configuration

The Network Configuration parameters must be set BEFORE the device is installed on the network. Once the network parameters are set properly they can be modified via software over the network. Either a custom program or a Telnet client program can be used to connect to the EP210/WEP210 and change the parameters. In this project I have used telnet with the help of patty software. If the primary TCP/IP port is set for VTC (Virtual Terminal Command) emulation mode (MODE=1) it can also be used to modify the parameter. In addition, the configuration parameters can be modified through the serial port using VTC commands.

The **"SHOW"** command will display the network parameters and their current values. Any parameter can be changed using a VTC command and the new value. A VTC command consists of the parameter name, followed by the "=" character, followed by the new value, and terminated by a <CR> [<LF>] character sequence. For example, to change the NETMASK parameters, use the command:

NETMASK=255.255.255.0<CR><LF>

The following table summarizes the Network Configuration parameters and the VTC command names used to change them. Also the factory default value and a description of each parameter are provided. Some of the useful VTC commands are: BELL, CLOSE, CONFIGURATION, CONNECTED, COUNTER, DISPLAY, ERROR, INPUTS, KEY, NETRETRY, RELAY, RESET, SAVE.

Name	Default Value	Description
MYIP	192.168.168.50	Defines the unique network address of this device. This value must be set BEFORE installing the device on a network. The default value may NOT be appropriate and a new address should be obtained from the network administrator.
NETMASK	255.255.255.0	Defines the network mask for the IP address. This value must be set BEFORE installing the device on a network. The default value may NOT be appropriate and a new value should be obtained from the network administrator.
GATEWAY	0.0.0.0	Can be set to the address of a router or gateway if the network extends to multiple segments. This value must be

		set BEFORE installing the device on a network. The default value may NOT be appropriate and a new address should be obtained from the network administrator.
TCPPORT	1070	Defines the primary TCP/IP port number used for this device. This value must be set BEFORE installing the device on a network. In most cases the default value will be acceptable. However, it may NOT be appropriate and a new port number should be obtained from the network administrator.
SERVER	0.0.0.0	Defines a TCP/IP address for a server application. This value must be set BEFORE installing the device on a network. Normally, this value is set to 0.0.0.0 which causes the decoder to operate as a "server" and connects with a host computer running a "client" application. If this value is set to any other IP address the decoder will operate as a "client" and will attempt to automatically connect to a host computer "server" application.
AUXPORT	9600,0,8,1,1	Defines the communications format for the serial port. The parameter string consists of five (5) integer value fields and has the following format: Baud, parity, data bits, stop bits, xoff Where: Baud = baud rates in the range of 110–57600 Parity = 0 (none), 1 (even), 2 (odd) Data bits = 7 or 8 Stop bits = 1 or 2 Xoff = xon/xoff protocol (0=disable, 1=enable)
MODE	1	Defines terminal emulation mode. The three (3) valid emulation modes are as follows: 1 = VTC Mode (default); 2 = ANSI Mode 3 = ANSI Mode.
MYMAC	0050C2163007	This command will return a 12 character hexadecimal string representing the Ethernet hardware address. It is read only and can not be used to change the hardware address.

Table 4.3. Some Important VTC Commands

Function	EP210 Msg	Remark
Make Connection	CONNECTED	When connection established
User swipe	KEY=12345	Magnetic card or bar code swipe
PROX Swipe	AUX=23456	Input on Serial link
ReadInputs	INPUTS=0,1,0,0	Value of the 4 inputs (2 used)
Control Relay	RELAY=1,1 RELAY=1,2 RELAY=1,0	Turn Relay1 on Turn Relay1 on for 2 seconds Turn Relay1 off
Beep	BELL	Beeps the Unit

Watchdog	WATCHDOG=10	0=>disable; 10=>receives commands at least for 10 secs.
Invalid Command	ERROR	When an invalid Command received.

Table 4.4. Some Important Functions to Configure EP210

Refer: **APPENDIX B** for the driver program that accepts both KEY and AUX commands.

Here is the screen shot of PuTTY.

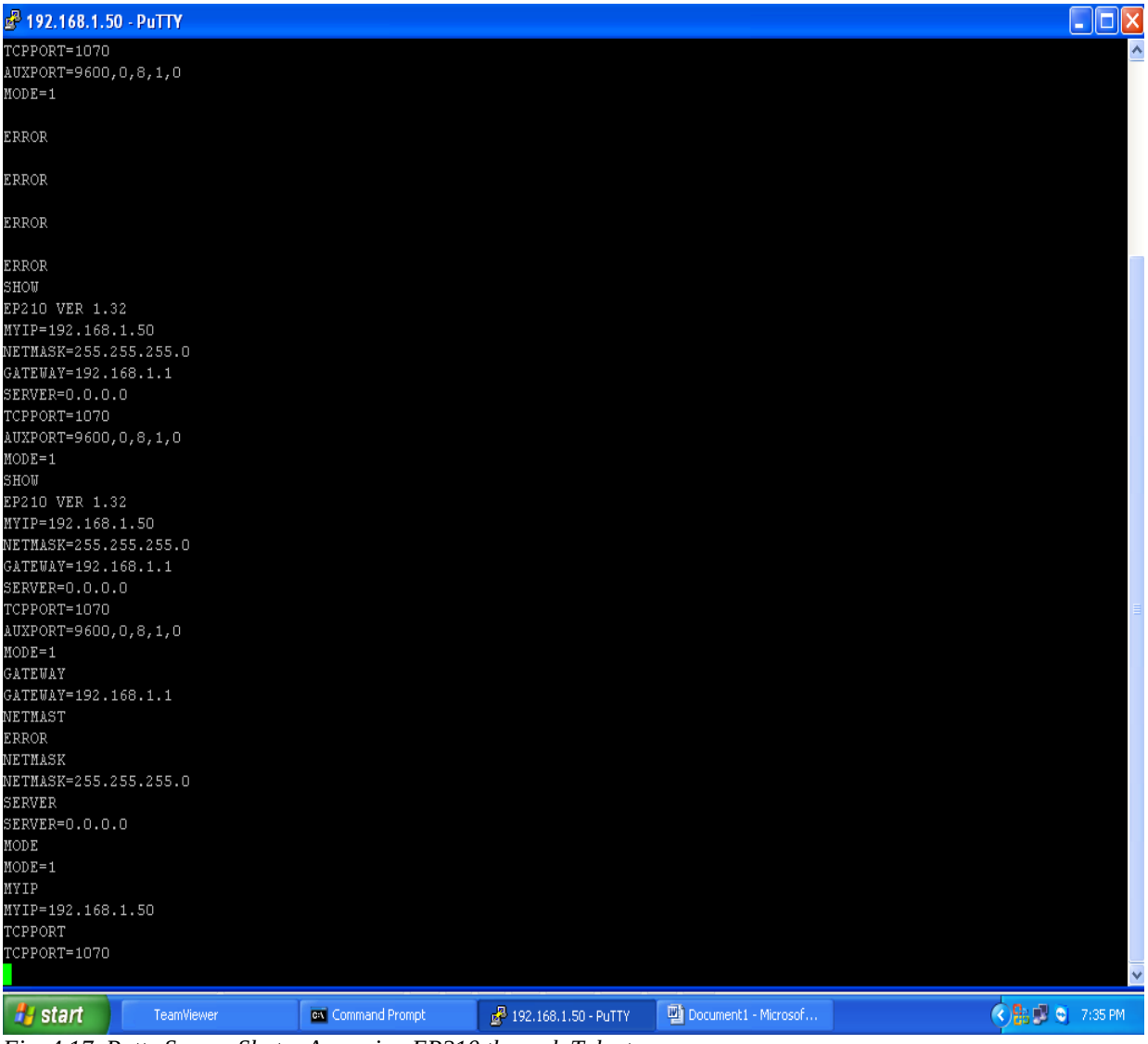


Fig. 4.17. Putty Screen Shot – Accessing EP210 through Telnet

CHAPTER 5

5. The Home Gateway[25]

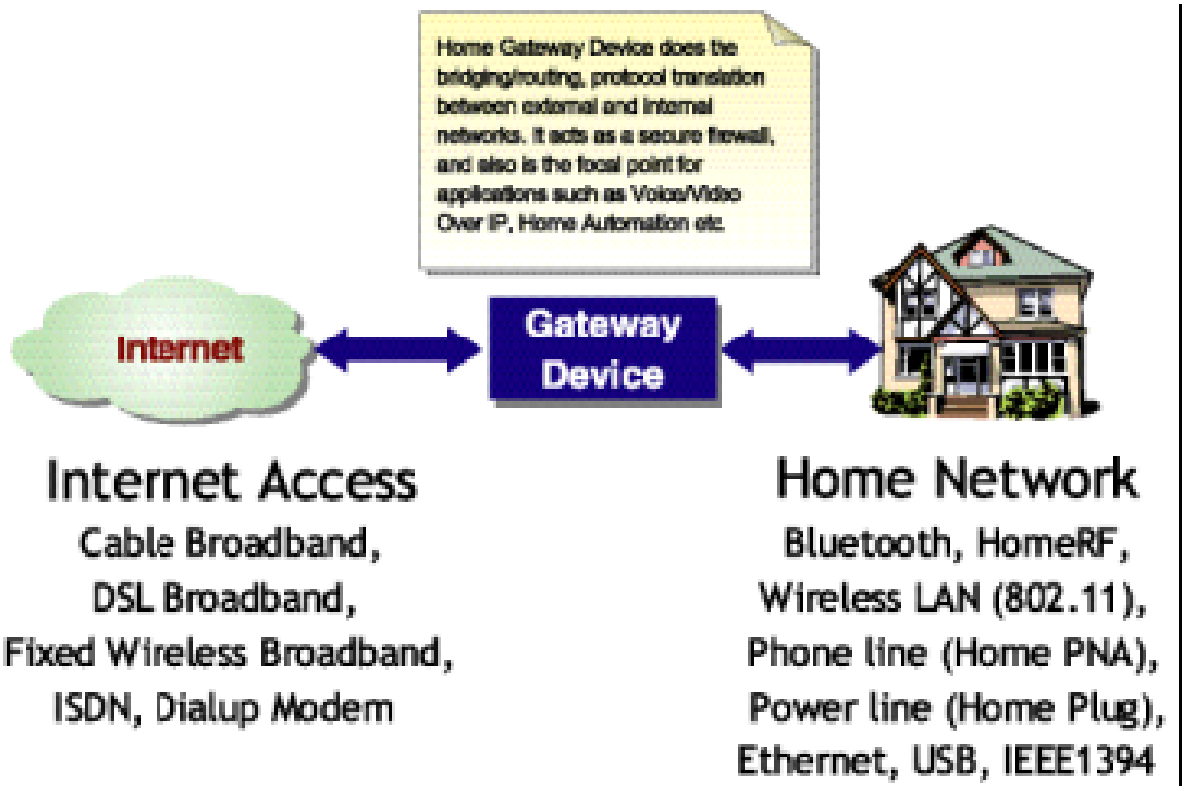


Fig. 5.1. Structural Diagram for Internet Gateway.

Gateway, by definition, joins two networks together. Rightly, Home Gateway device does the bridging / routing, protocol and address translation between external broadband network and the internal home networks. It acts as a secure firewall, and also is the focal point for applications such as voice/video over IP (VoIP), home automation etc. It allows the home users to access their home networks and to control various devices from a remote location through internet. The benefits of Home Gateway are:

- Internet connection sharing and simultaneous Internet access. In-home file/print sharing
- In-home Wireless LAN connectivity. VPN connectivity to work place

[25]http://www.broadcastpapers.com/whitepapers/WiproHomeGateway.pdf?CFID=19547332&CFTOKEN=5cbb0b8311f7a0ee-A7E626F6-9560-86F2-B035477CD9266066

- Firewall security, parental protection.
- Broadband telephony (Voice over IP). IP Video on Demand
- In-home audio & video streaming. Remote health monitoring
- Security surveillance. Home automation & home control. Remote meter reading

5.1. Email Server for Home

A mail server (also known as a *mail transfer agent* or MTA, a *mail transport agent*, a *mail router* or an *Internet mailer*) is an application that receives incoming e-mail from local users (people within the same domain) and remote senders, and forwards outgoing e-mail for delivery. For example, Microsoft Exchange is a mail server. Email server also stores email data prior to delivery. It is necessary for home because it can filter spams, scan for viruses, delivers emails speedily, does not depend upon the public email servers to get home emails.

5.1.1. SMTP

SMTP stands for Simple Main Transfer Protocol. Email Server uses SMTP to send and receive mail messages whereas user-level client mail applications typically use only SMTP for sending messages to a mail server for relaying. In short the SMTP works like a post assistant handling the sending of emails from an email client to an email server and receives outgoing mail messages from users and routes them to the mail recipients they are intended for.

5.1.2. POP

POP stands for Post Office Protocol. It is one of the technologies that allows email sent from anywhere in the world to arrive in your inbox. When a person sends an email to our address, it is transmitted over the internet, and eventually lands on our mail server. In order for our personal computer to get that mail it must follow a certain protocol. POP allows our computer to talk to the email server and then download all the messages each time we connect.

Advantages: It downloads all our messages onto a local computer. This way we can view our messages without having a live internet connection. Another advantage of POP is for email servers with limited storage space. For example, a small company may have only one small email server for its entire staff. If each person periodically downloads all of his or her email to

a local computer, it saves the company from having to invest in more storage space.

Disadvantages: POP is useless for mobile user. Since messages are permanently downloaded, the user will only be able to access the messages from one PC. It does not leave the messages on the server. The other alternative is to use IMAP.

Difference between POP and SMTP: POP is a protocol for storage of emails whereas SMTP is for sending and receiving. POP is like a mailbox or post office box. It is the location the mail is delivered to and where it stays until the recipient is ready to read it. Outgoing mail can also be put in the mail box. SMTP would be like letter carrier or mailman.

5.1.3. HTTP

HTTP stands of Hyper Text Transfer Protocol. It is the protocol for transferring hypertext documents that makes the World Wide Web possible. In other words, it defines how messages are formatted and transmitted, and what actions web servers and browsers should take in response to various commands. When we enter, for example, www.xavierengg.com in the browser, this actually sends an HTTP command to the web server directing it to fetch and transmit the requested web page. It is a stateless protocol because each command is executed independently without any knowledge of the commands that came before it. The shortcoming of HTTP is being addressed in a number of new technologies, including ActiveX, Java, JavaScript and cookies.[26]

5.1.4. IMAP

IMAP stands of Internet Message Access Protocol. It is an email service that helps accessing the mailbox from different computers and locations. It allows the user to download only the message headers and decide if they want to actually download the rest of the message and the attachments. With IMAP, mail is kept on the mail server, so it is very convenient for PDAs, i-phones, i-pads, etc., to access the mails. Besides, it provides folders for the user to store emails and attachments on the server so that they can retrieve those stored messages when they log into the server from different computers. This new feature provides a powerful

[26] http://www.webopedia.com/TERM/H/HTTP.html

benefit for the user to create folders for received and sent messages to be retrieved from any computer. Thus IMAP allows us to read our emails from any location and any device with IMAP support.

Advantages of IMAP over POP[27]:

- Freedom for user to download attachments at will
- Robust folders for storing received and send messages
- Provision for determining message structure without downloading entire message.
- Selective fetching of individual MIME (Multipurpose Internet Main Extensions) body parts.
- Server-based searching and selection to minimize data transfer.
- Ability to append messages to a remote folder.
- Ability to set standard and user-defined message status flags.
- Support for simultaneous update and update discovery in shared folders. New mail notification.
- Ability to manipulate remote folders other than INBOX. Remote folder management (list/create/delete/rename).
- Support for folder hierarchies. Suitable for accessing non-email data; e.g., NetNews, documents.
- In IMAP, when a client program performs any operation on a mailbox, the server will automatically include in its response notification of any new messages that have arrived since the last notification.
- IMAP's ability to manipulate remote folders other than INBOX is fundamental to online and disconnected operation. This means being able to save messages from one folder to a different one, being able to access archived messages subsequently, and allowing for multiple incoming message folders.

5.1.5. Implementations in IHS

Implementation of IHS is that it:

- provides HTML browser based email client for laptop and i-phones / i-pads.
- provides built-in ClamAV based virus scanner
- provides built-in Spam Assassin support
- provides Configurable filters, DNSBL servers, blacklists and whitelists
- integrates with email clients via IMAP
- receives external mail via SMTP or POP

[27] http://www.cyberindian.com/web-hosting/article.php?article_id=88

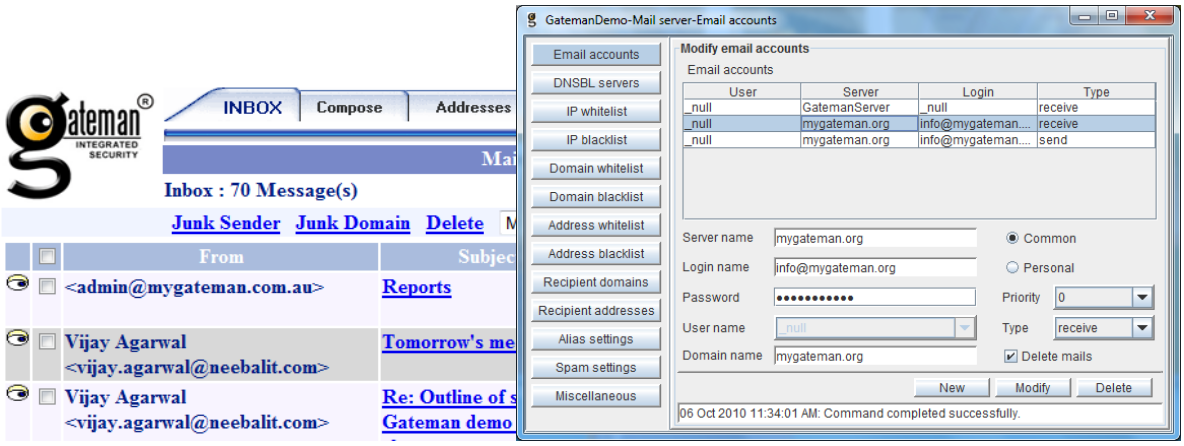


Fig. 5.2. Configuration of Email Server in IHS

5.2. Web server for Home

Do we need a web server for home? Yes, we do need it, because it delivers web pages when we ask for them through internet or www. It is not necessary to have a separate computer as web server, rather it could be incorporated into other servers like email server or proxy server or file server or DHCP server. There are two web servers which are most common: Jetty and Apache.

Fig. 5.3. Schematic Diagram for Web Server

Apache is one of the most popular Web Servers (HTTP Server) in the world. It was originally designed for Unix servers, but later has been ported to Windows and other **N**etwork **O**perating **S**ystems (NOS). The name Apache derives from 'patchy' that the Apache developers used to describe early versions of their software. It provides a full range of Web server features, including GGI, SSL, and virtual domains. It also supports plug-in modules for extensibility. Besides, it is reliable, FREE, and relatively easy to configure.

Jetty is free and open source project under Apache 2.0 license. Jetty Web Server is a 100% Java-based HTTP Server and servlet container (application server). This means that we do not need to configure and run a separate web server (like Apache) in order to use java, servlets and JSPs to generate dynamic content. Jetty is a fully featured web server for static and dynamic content. Unlike separate server / container solutions, Jetty server and web application run in the same process, without interconnection overheads and complications. Jetty deployment focuses on creating a simple, efficient, embeddable and pluggable web server. Jetty's small size makes it suitable for providing web services in an embedded Java application. IHS uses Jetty web server.

5.3. Web Proxy

A proxy server is basically a server that acts as an intermediary between a workstation users and the Internet so that the enterprise can ensure security, administrative control, distribution of bandwidth and caching service. A proxy server is associated with or part of a gateway server that separates the enterprise network from the outside network and a firewall server that protects the enterprise network from outside intrusion. For home users Web Proxy is most essential for the users to go outside the home network and talk to other computers or access web pages.

An advantage of a proxy server is that its cache can serve all users. If one or more Internet sites are frequently requested, these are likely to be in the proxy's cache, which will improve user response time. In fact, there are special servers called cache servers. A proxy can also do logging. The functions of proxy, firewall, and caching can be in separate server programs or combined in a single package. Different server programs can be in different computers. For example, a proxy server may in the same machine with a firewall server or it may be on a separate server and forward requests through the firewall.[28]

[28] http://whatis.techtarget.com/definition/0,,sid9_gci212840,00.html

5.3.1. Distribution of Bandwidth

One of the significant contributions of Web Proxy is the distribution of bandwidth to the users. Some may work on graphics and heavy download and others may just work on emails. So the distribution of bandwidth and balancing the traffic are necessary. Here is an example how WinGate software does the bandwidth distribution.

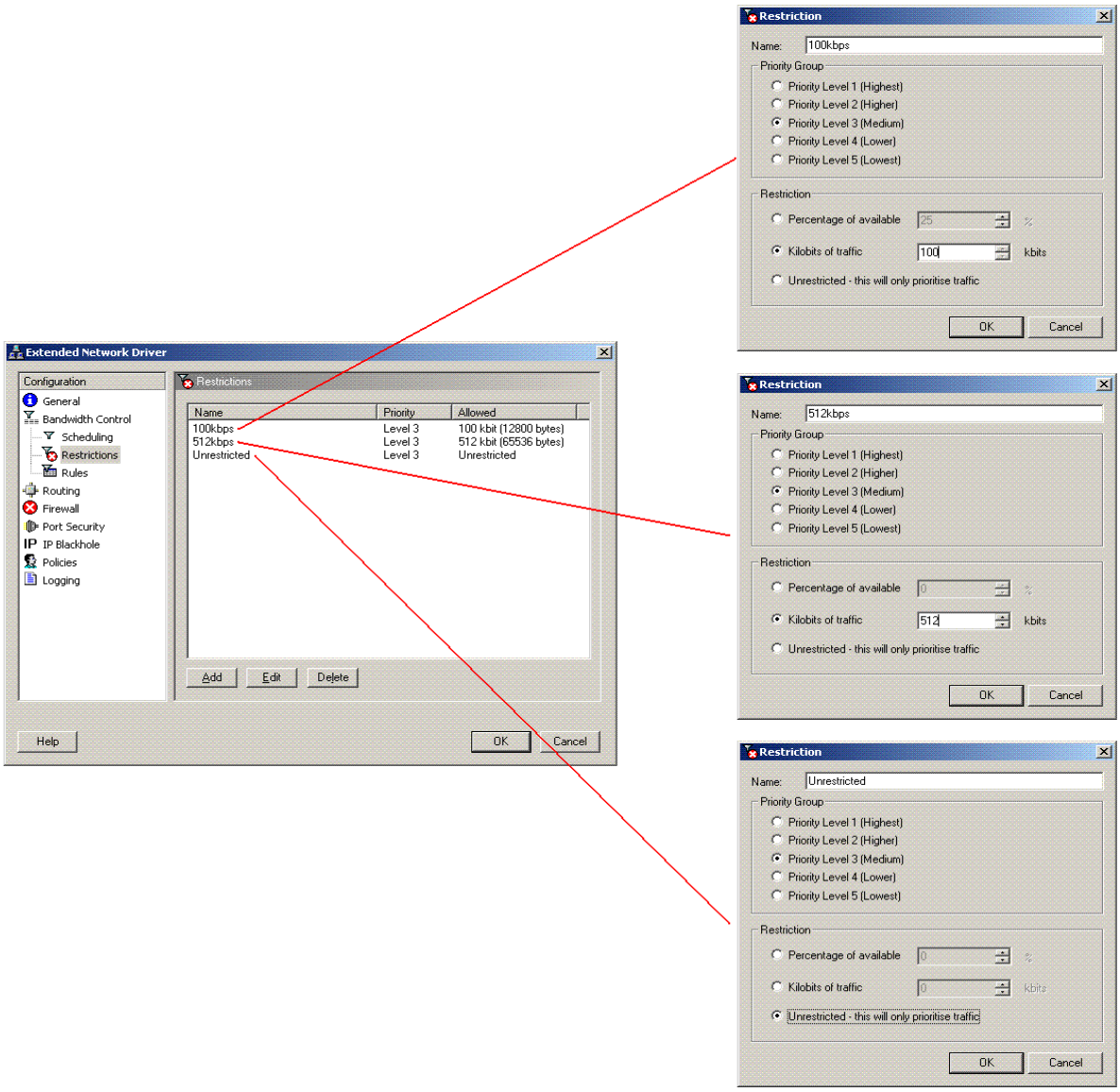


Fig. 5.4. Web Proxy Configuration in IHS

The drawback of WinGate is that it does not do dynamic bandwidth restriction adjustments. Microsoft has come out with "Bandwidth Splitter" which is a program extension for Microsoft ISA Server. It can limit the internet connection bandwidth used by individual users and hosts, as well as groups of users and hosts (traffic shaping, throttling), can setup quotas for the maximum allowable internet traffic use for a period of time and finally it can also monitor of all users and their connections through ISA server.

Drawback of MS-bandwidth splitter is that it does not control low-level, non TCP/UDP based protocol.[29]

5.3.2. Caching Web Pages

The proxy caching technique can significantly reduce server's load. Web caching is one of the major advantages of web proxy. It makes web browsing faster. For example when many people are searching for a "Lifestyle OS" on the internet, they would be served at once as the pages are cached in web proxy when the first one searched it. How does it work? A proxy server receives a request for an Internet service (such as a Web page request) from a user. If it passes filtering requirements, the proxy server, assuming it is also a cache server, looks in its local cache of previously downloaded Web pages. If it finds the page, it returns it to the user without needing to forward the request to the Internet. If the page is not in the cache, the proxy server, acting as a client on behalf of the user, uses one of its own IP addresses to request the page from the server out on the Internet. When the page is returned, the proxy server relates it to the original request and forwards it on to the user. To the user, the proxy server is invisible; all internet requests and returned responses appear to be directly with the addressed internet server.

5.3.3. Being Aware of Children's Activities

The third salient feature of proxy server is that the parents are in control of what their children are browsing on the internet. They can keep a track of browsed websites. If the parents feel that some websites are not useful and waste of time or the children should be restricted to non-

[29] http://www.bsplitter.ir/Bandwidth%20Splitter.pdf

pornographic sites, they can block and blacklist them. Thus, the parents know what the children are doing when they are on the internet. Secondly, it is important for the parents to know how much time their children spend on the internet and on what sites they spend most of their time. This is the power of web proxy. Isn't it amazing?

5.3.4. Implementation in IHS

As we know IHS is carved from CentOS, it uses open source program Squid Proxy Server and Squid Guard for direct implementation. Squid proxy server allows us to go to the internet and does caching of the web pages. It gives an idea of where people are going and gives an extensive bandwidth management tool to control the bandwidths. Let us say for example, the parents decide that the children can browse only between 6 pm and 8 pm. It can be set. One of the amazing things of IHS is its reporting. It can report the Children's browsing patterns and sites visited. In an office this feature allows monitoring usage of the internet facility for non-official purposes.

The configuration is simple in IHS because it has a GUI. So an end user can easily do it. Secondly, IHS uses 'Squidguard' which does blocking, blacklisting and automatic updating of its latest viruses through internet. It is important to note that the Squid proxy server will go along with Squid guard.

A. To Configure the Allowed and Denied Sites

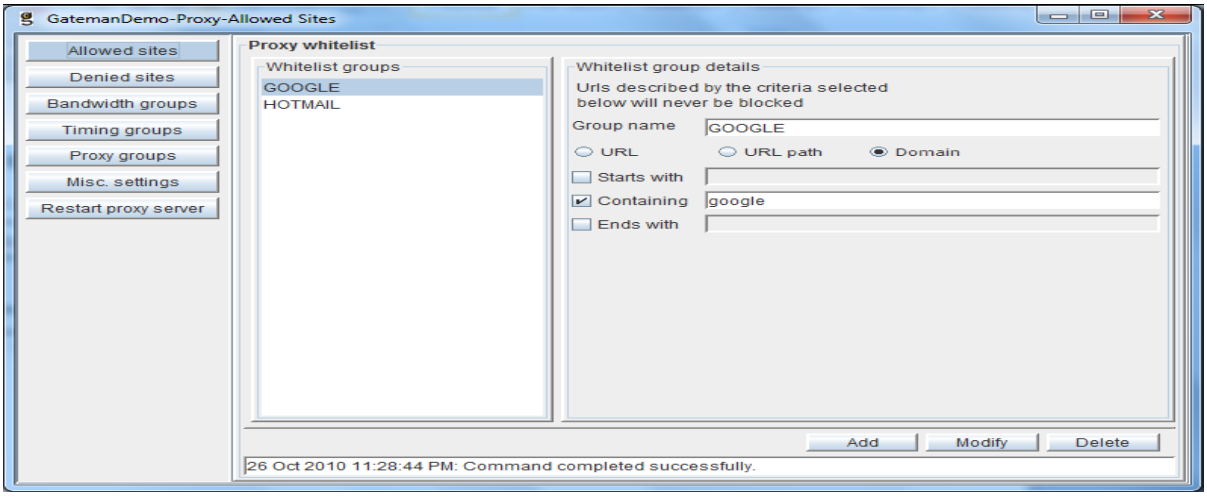


Fig. 5.5. Web Proxy Allowed Sites

As shown above, the allowed sites are included for access. Similarly, the denied sites can be set up. If we wish to block someone downloading .mp3 files, we just need to add it against group name as shown below.

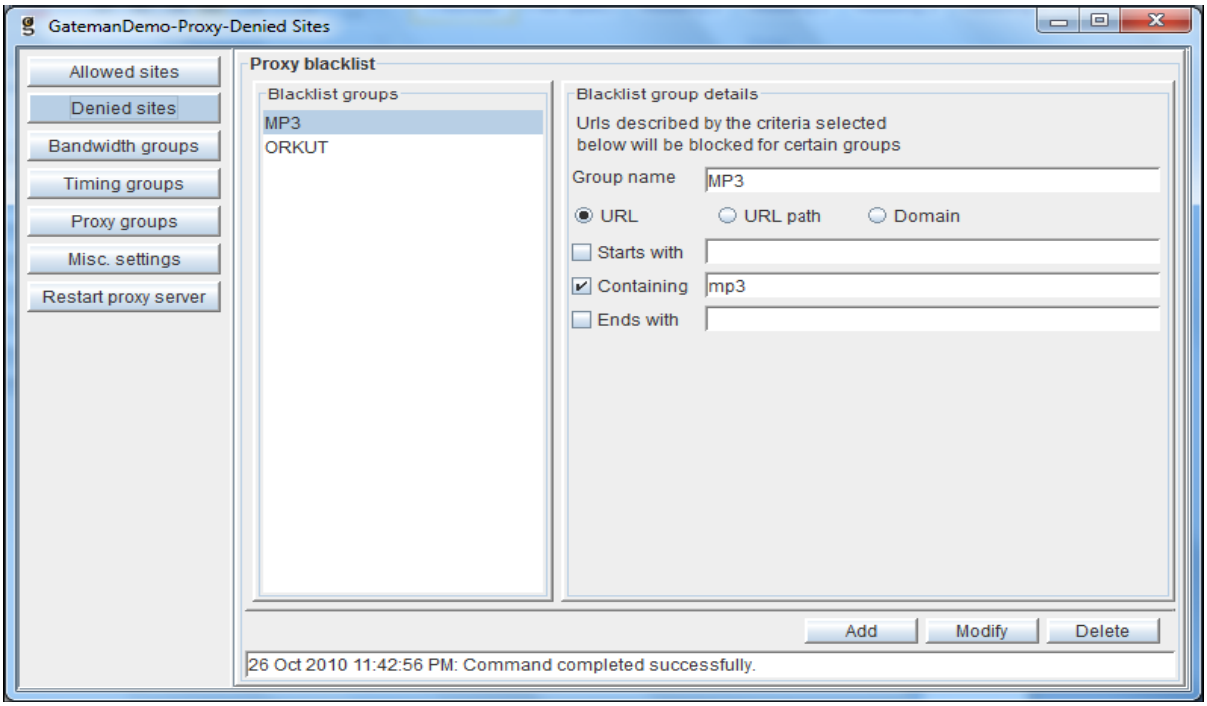


Fig. 5.6. Web Proxy Denied Sites

B. To Configure Bandwidth Groups

Distribution of available bandwidth to the home / office users is necessary for managing home / office network. For this we need to create a bandwidth group for which a user group must be assigned. As shown below, for each user the maximum bandwidth can be allotted and the maximum size of the file that can be downloaded.

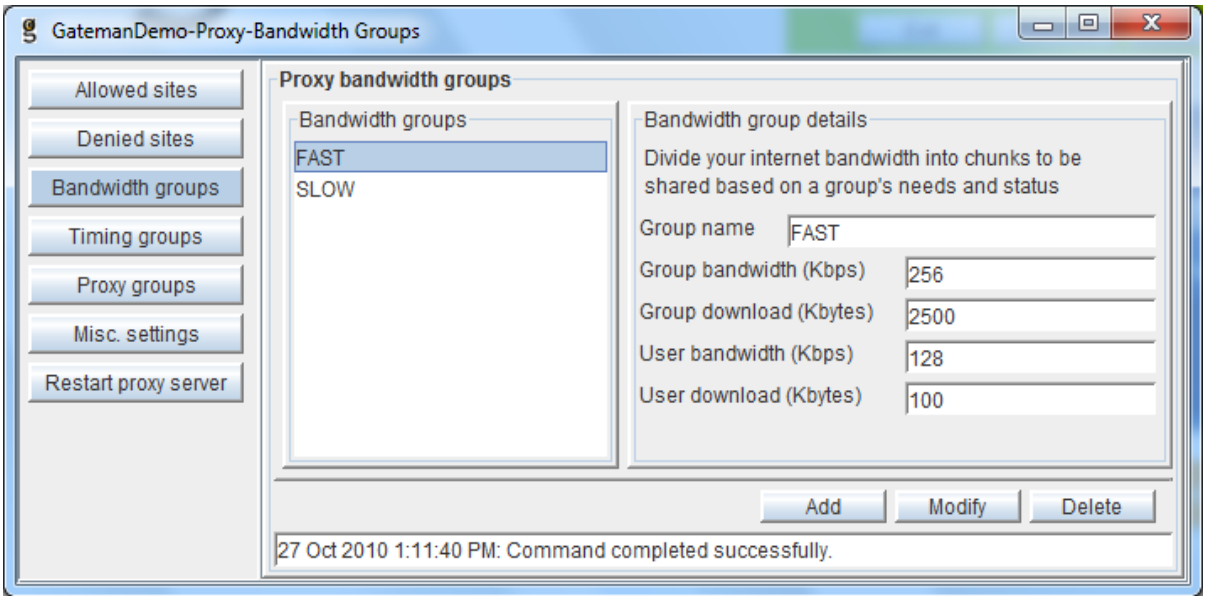


Fig. 5.7. Asssigning Bandwidth to Groups.

C. To Configure the Timing Groups

This is really useful for parents who want to setup timings wherein they could allow their children to browse and play on the internet. They want, for example, to allow from 7 pm till 9 pm on weekdays and from 8 am till 9 pm on Sundays. Similarly, it can be done in an office where the employees are allowed to browse during lunch / coffee break.

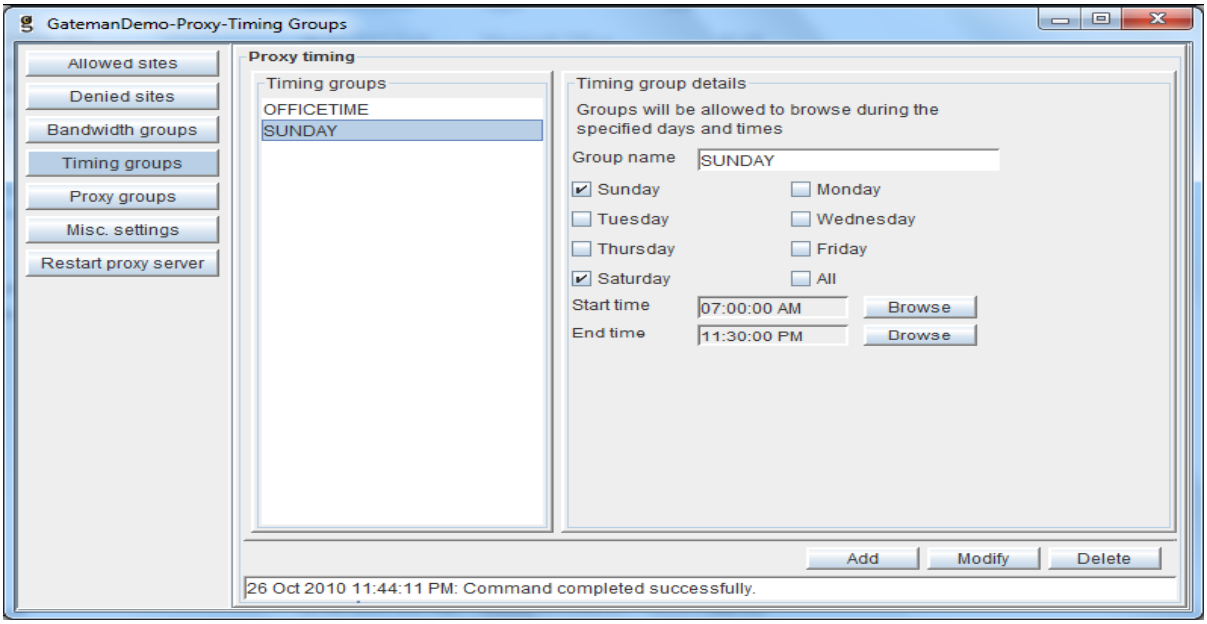


Fig. 5.8. Assigning Timings to Groups

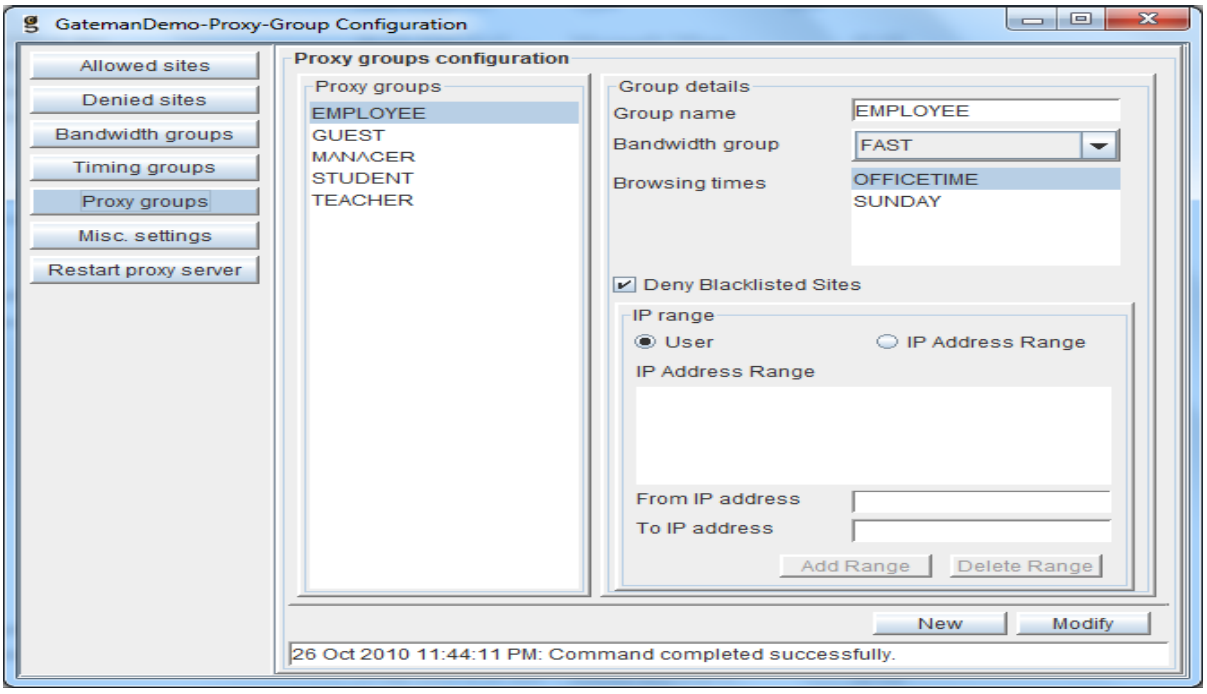


Fig. 5.9. Proxy Group Configuration

Configuring squidGuard is very trivial. The default location for the block lists is `/usr/local/squidGuard/db/`. Each category is located in a different directory. Currently, the categories available are ads, aggressive, audio-video, drugs, gambling, hacking, porn, violence, and warez. The configure file is `/etc/squid/squidGuard.conf`. Here is a sample configuration file:[30]

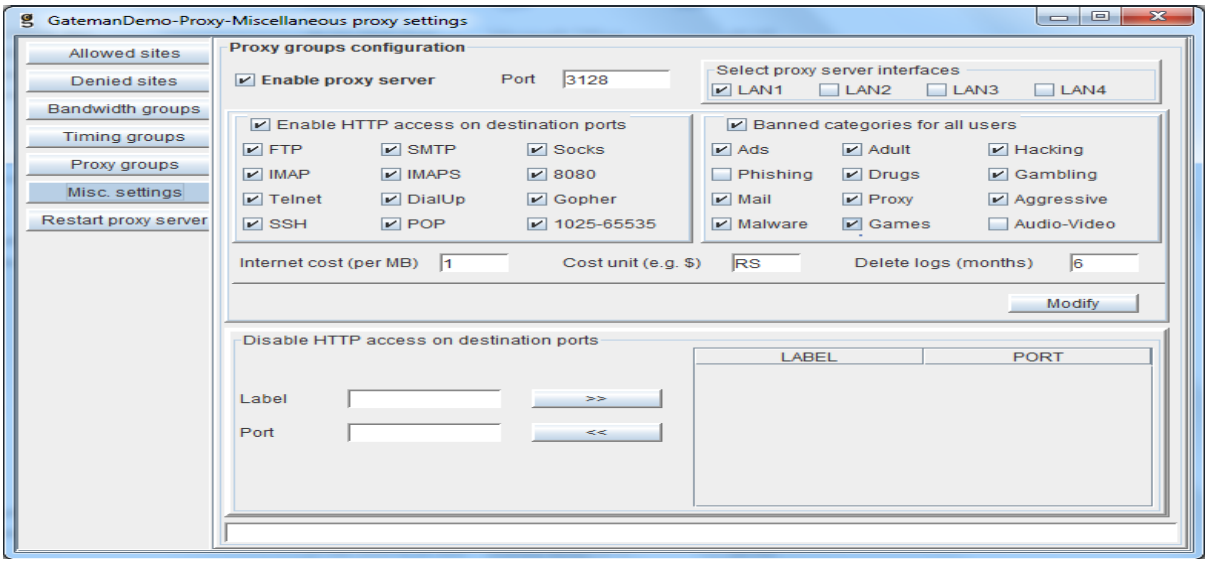


Fig. 5.10. Miscellaneous Proxy Settings

[30] http://squidguard.mesd.k12.or.us/

```
dbhome /usr/local/squidGuard/db
logdir /usr/local/squidGuard/log

dest gambling{
        log             gambling
        domainlist      gambling/domains
        urllist         gambling/urls
}

dest warez{
        log             warez
        domainlist      warez/domains
        urllist         warez/urls
}

acl {
          default {
                  pass !gambling !warez all
                  redirect 302:http://www.google.com
          }
}
```

dbhome defines where the block list databases are located
logdir defines where to log blocked requests
dest defines a category
acl defines the access control lists.

5.4. Guarding the Home Network against Intrusion

Fig: 5.11. IHS with two ethernet cards.

Network security is a primary concern for any internet user. Internet threats are becoming more widespread, sophisticated and persistent every day. Some countries are talking about the next world war as cyber war. Keeping today's trend in mind, we need to stop people from outside break into home network. Most homes use ADSL router which is the first barrier to stop people to come inside from outside. But we know it is not difficult to break into ADSL and get into the

server. It is advised, therefore, to have two network cards in IHS server. As shown in the figure, ISP's line comes to one Ethernet card and the other connected to the router

5.4.1. IP Tables and Shorewall

One of the means of providing additional protection is to have a firewall. It is easy to use LINUX IPtables package on an existing server. A Linux server can be converted into a firewall while simultaneously being our website's mail server, web server, web proxy, DHCP server and DNS server.

The Shorewall Firewall, popularly known as "Shorewall" is a gateway / firewall high-level configuration tool for GNU/Linux / netfilter. We describe our firewall / gateway requirements using entries in a set of configuration files. Shorewall reads those configuration files and with the help of the iptables, iptables-restore, IP and TC utilities. Shorewall also configures Netfilter and the Linux (in our case CentOS) networking subsystem to match our requirements. Shorewall can be used on a dedicated firewall system, a multi-function gateway/router/server or on a standalone GNU / Linux system, but it does not use Netfilter's ipchains compatibility mode and can thus take advantage of Netfilter's connection state tracking capabilities.

It is important to remember that Shorewall is not a daemon. Once Shorewall has configured the Linux networking subsystem, its job is complete and there is no "Shorewall process" left running in our system. Though Shorewall may not be the easiest to use of the available iptables configuration tools, but it is definitely the most flexible and powerful. Shorewall consists of four packages: **Shorewall**, **Shorewall6, Shorewall-lite and Shorewall6-lite**[31].

5.4.2. PSAD – Port Scan Attack Detector

PSAD is an awesome security tool for Linux and it analysis what iptables are stopping. It is basically a collection of three system daemons that are designed to work with the Linux iptables firewalling code to detect port scans and other suspicious traffic. A typical

[31] http://www.shorewall.net/

deployment is to run PSAD on the iptables firewall where it has the fastest access to log data.

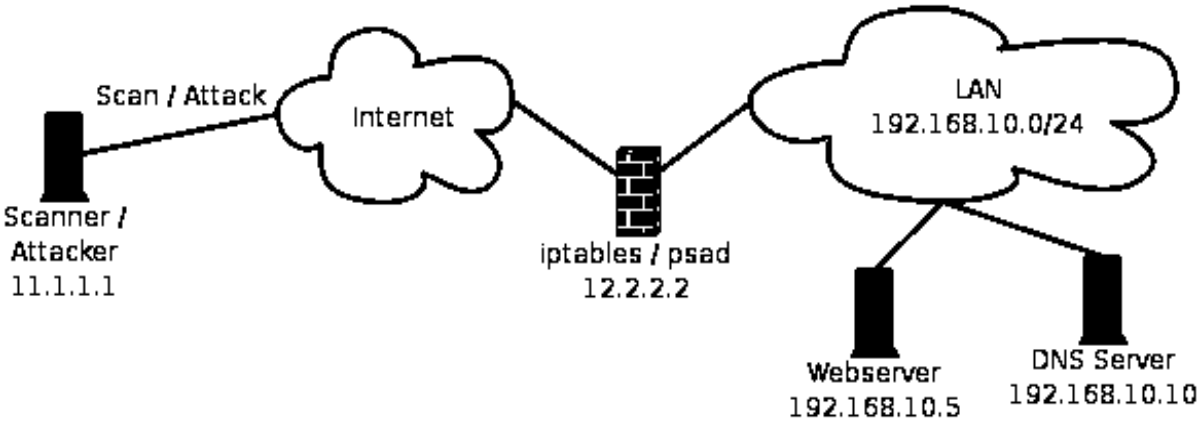


Fig. 5.12. Schematic Diagram for PSAD

It features a set of highly configurable danger thresholds, verbose alert messages, email alerting, DShield reporting, and automatic blocking of offending IP addresses. It also incorporates many of the packet signatures included in Snort to detect various kinds of suspicious scans.[32] PSAD is developed around there main principles:

- Good network security starts with a properly configured firewall.
- A significant amount of intrusion detection data can be gleaned from firewalls logs.
- Suspicious traffic should not be detected at the expense of trying to also block such traffic.

5.5. Other network requirements

5.5.1. DHCP Server

The Dynamic Host Configuration Protocol (DHCP) server is the one that is responsible for assigning unique IP address to the computers on a network. DHCP allows a computer to be configured automatically, eliminating the need for intervention by a network administrator to assign an IP address for each computer. It avoids two computers from accidentally being configured with the same IP address as no two computers (actually, no two network cards [even if two are in one computer] can have the same IP address on a network at the same time. To that end, DHCP servers will take a request from a computer that has just been added (or is renewing) to the network and assign it a unique IP address (i.e. 192.168.55.23) that is

[32] http://freshmeat.net/projects/psad/

available. These assignments typically only last for a limited time (an hour to a week usually) and so we are never gauraunteed that the IP address for a particular computer will remain the same when using a DHCP.

There are two versions of DHCP: One for IPv4 which is very common and another for IPv6. As the public IP addresses are running short, IPv6 is slowly replacing IPv4. IHS is capable of supplying both the IP formats.

5.5.2. DNS Server

We need a Domain Name System (DNS) to translate domain names to IP (Internet Protocol) addresses which are unique to each computer. Without DNS, the internet would shut down immediately. When we use the Web, we use a domain name to do it. For example, the URL http://www.john.mygateman.org contains the domain name mygateman.org. So does the e-mail address "john.mygateman.org."

It is easy to remember "john.mygateman.org" instead of an IP address, "129.210.19.43". But the machines use names called IP addresses to refer to one another. So whenever we use the domain name, we use the internet's domain name servers (DNS) to translate the human-readable domain name into the machine-readable IP address. DNS system, therefore, is a database. It is unique because no other database on the earth has millions of people changing it every day.

DNS Server and Home Networking: Computers on our home network locate a DNS server through the Internet connection setup properties. ISP gives us the public IP addresses (for example 129.210.19.43) of primary and backup DNS servers. We can find the current IP addresses of our DNS server configuration through various methods.

- By typing ifconfig (in Linux) and ipconfig (windows, DOS prompt)
- By seeing into the configuration of the home network router
- On the TCP/IP connection properties screens in control panel or Network Administration

CHAPTER 6

6. Physical Layer Considerations

The Physical Layer is the first and lowest layer in the seven-layer OSI model[33] of computer networking. It consists of the basic hardware transmission technologies of a network. It is a fundamental layer underlying the logical data structures of the higher level functions in a network. Due to the plethora of available hardware technologies with widely varying characteristics, this is perhaps the most complex layer in the OSI architecture. In short, the physical layer deals with transporting bits between two machines.

Computer networks for the home and small business can be built using either wired or wireless technology. Wired Ethernet has been the traditional choice in homes, but Wi-Fi wireless technologies are gaining ground fast. Both wired and wireless can claim advantages over the other; both represent viable options for home and other local area networks (LANs). It can also be done ethernet over power, powerline carrier, ZigBee, etc. Let us analyse one by one.

6.1. Wired Ethernet

When Ethernet was developed, it used a fat coaxial cable with taps clamped on at prescribed intervals. Today the most common type of Ethernet wiring is unshielded twisted pair (UTP) copper cable consisting of 4 pairs of wire terminated with 8 conductor jacks similar to those used for telephone wiring. This has dramatically reduced the cost of implementing a LAN.

[33] The seven layers of the OSI Basic Reference Model are: The **Physical Layer** that describes the physical properties of the various communications media, as well as the electrical properties and interpretation of the exchanged signals, the **Data Link Layer** that describes the logical organization of data bits transmitted on a particular medium, the **Network Layer** that describes how a series of exchanges over various data links can deliver data between any two nodes in a network, the **Transport Layer** that describes the quality and nature of the data delivery, the **Session Layer** that describes the organization of data sequences larger than the packets handled by lower layers, the **Presentation Layer** that describes the syntax of data being transferred and finally the **Application Layer** that describes how real work actually gets done. Ex: this layer would implement file system operations.

Ethernet IEEE 802.3 is the most common local network technology used today. It is based on CSMA/CD (Carrier Sense Multiple Access With Collision Detection) scheme[34].

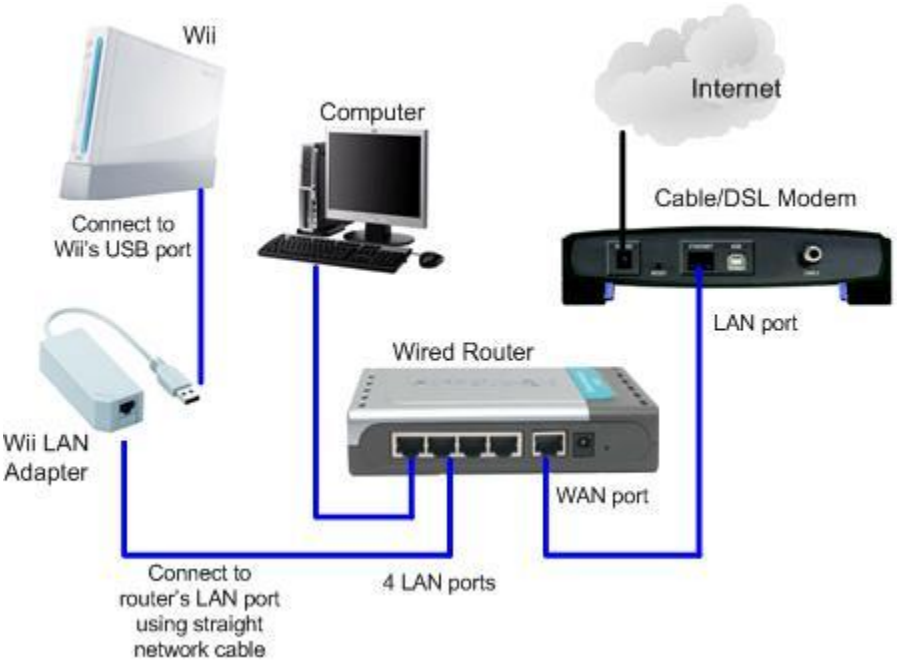


Fig. 5.13. Schematic Diagram for Wired Ethernet

CAT5 and CAT6 cables are common in wired ethernet. Typically, wired networks offer greater security and more protection from data hacking. In addition, they usually run faster, allowing users to move large files around more quickly. It gives a stable and reliable network. The disadvantages of using wired network are that the cables are installed inside the walls of a home or office. Installation of these cables can be expensive specially if there are many computers. A wired network can also hinder mibility as the laptop users will not be able to move their computers from one room to another within a home.

6.2. Wireless Ethernet

(Fig. 5.14. Wireless Router)

Today the slogan goes as "throw out those CAT5 and CAT6 wired Ethernet," since speed and power of wireless technology is rather good. It is being reported that 802.11n wireless technology will start eroding the wired Ethernet market within the next two to three years. This is because the number of laptop users are growing, the

[34] http://www.dslreports.com/faq/2731

enterprise uses more mobile applications, fast ethernet throughput is good enough, the enterprise deploys voice over internet protocol (VoIP),), the risk of deliberate denial of service attack is low to moderate and ethernet cable installation is difficult as human labour is expensive.

IEEE 802.11 is a set of standards carrying out wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands. The most popular are those defined by the 802.11b and 802.11g protocols. The data transfer rate ranges from 2 Mbits to 54 Mbits. And its transmit power is measured in dBm (power ratio in decibals.

6.3. Ethernet Over Power

Power over Ethernet (PoE) is a technology for wired Ethernet LANs that allows the electrical current, necessary for the operation of each device, to be carried by the data cables rather than by power cords. This minimizes the number of wires that must be strung in order to install the network. The result is lower cost, less downtime, easier maintenance, and greater installation flexibility than with traditional wiring.

(Fig. 5.15. Unit for PoE)

For PoE to work, the electrical power must go through the data cable at the power-supply end, and come out at the device end, in such a way that the electric power is kept separate from the data signal so that neither interferes with the other. The current enters the cable via injector. If the device at the other end of the cable is PoE compatible, that device will function properly without modification. If the device is not PoE compatible, then picker or tap must be installed to remove the current from the cable. The picked-off current is routed to the power jack. Normally PoE systems emply fault protection which shuts off the

power supply if excessive current or a short circuit is detected.[35]

6.4. Power line Carrier (UPB)

(Fig. 5.16. UPB Devices with Null Modem Cable)

UPB is a protocol for communication among devices. It uses powerline wiring for signaling and control. Its communication method consists of a series of precisely timed electrical pulses (UPB pulses) that are superimposed on top of the normal AC power waveform (sine wave). One UPB pulse is generated each half-cycle of the 60 Hz AC electrical power cycle. The position of each UPB pulse determines its value as either 0, 1, 2 or 3. It follows the Pulse Position Modulation (PPM). Its reliablity is about 99%. Its drawback is that it is easily affected by various devices particularly heavy electrical motors. When the signals are disturbed, UPB devices do not function as programmed.

6.5. ZigBee

ZigBee is upcoming new technology. It will soon dominate the market for devices requiring monitoring and control. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, and secure networking.

[35] http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci846792,00.html

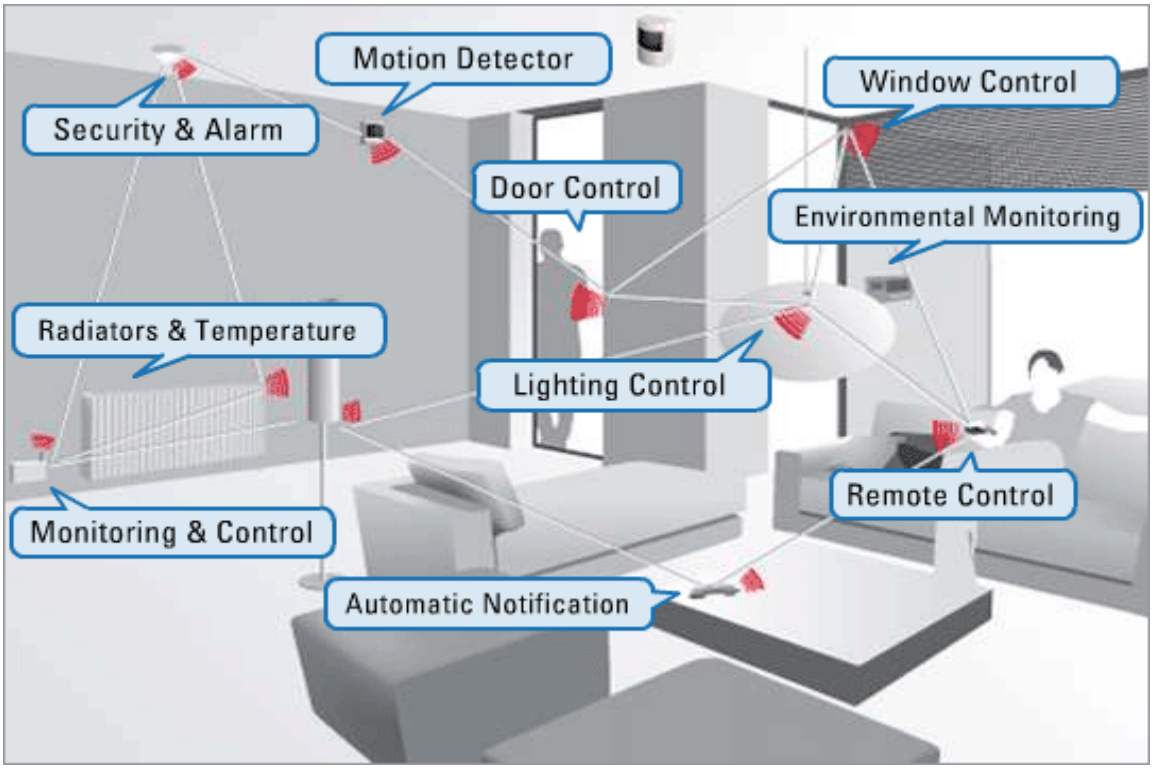


Fig. 5.17. Schematic Diagram for ZigBee.

ZigBee is a set of specs which are built around IEEE 802.15.4 protocol. It provides routing and multi-hop functions to the packet-based radio protocol. The ZigBee stack is shown below[36].

[36] http://www.rabbit.com/documentation/docs/manuals/ZigBee/Introduction/index.htm

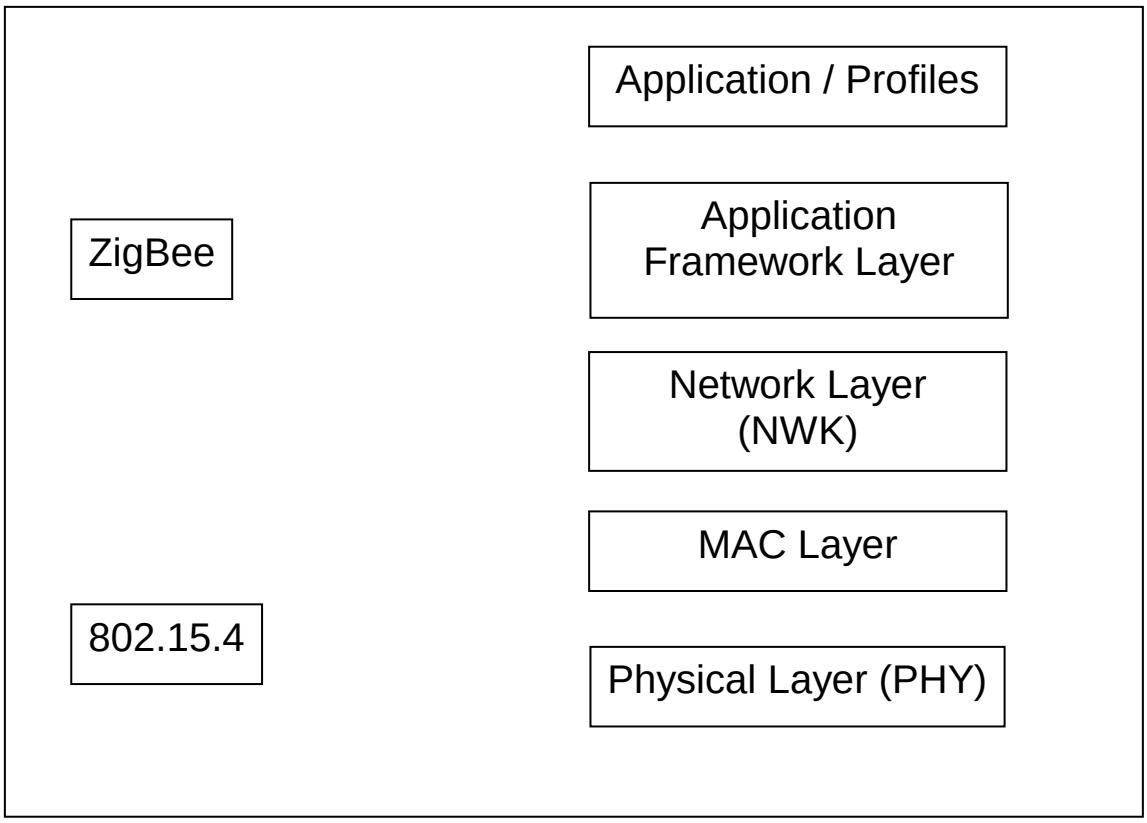


Fig. 5.18. The ZigBee Stack

For the ZigBee devices, the maximum operation range which is specified is 250 feet that is substantially further when compared to what is used for the devices using Bluetooth. The disadvantage of ZigBee is that since it is using low power devices, security is a major concern.

The chief advantages of ZigBee are:

- ZigBee provides a cost breakthrough that means wireless technology can be more widely deployed in wireless control and monitoring applications
- ZigBee power requirements are so low that devices can have extremely long battery life and use smaller "coin cell" type batteries.
- A ZigBee node can wake up, check in, send data, and shut down in less than 30 ms
- ZigBee is an open standard that is supported by a large number of vendors[37].

[37] http://advantage-dev.com/ZigBee-development-white-paper-2/?gclid=CNKr05io6KQCFQYHbAodHX2x2A

6.6. Media Evaluation

ZigBee seems to be ideal, but still not come out successfully in the market. If it comes we will have only ONE remote control for all the electronic devices and appliances. Next to ZigBee is UPB which avoids wiring and it is an ideal protocol for home automation, but it cannot be connected to a line where heavy elctrical devices are connected. It disturbs the signals and the home automation will not function properly. Third is wireless network. People are in transit constantly. Laptops and i-phones / i-pads are dominating the market as some cities in USA give wireless network for the entire city. New York is a case in point. Security is a major concern in wireless network. Lastly, wired network is stable and most reliable, but it is expensive in laying cables.

6.7. What is Prefered for home system?

If we are building a new home, we lay CAT6 cables. We may not use it regularly but can be kept as an alternative to wireless. If someone wants to do video streaming or play 3D games on the net and wireless is slow, then ethernet connections can be used. We need to install one good wireless (802.11 g/n) router, preferably at least 300 feet range, is to be installed. To control the appliances and home automation, UPB can be setup.

If it is an old home, we need to retrofit wireless (802.11 g/n) set up can be made available for the entire home users and also UPB could control the home automation. In future, we will have to setup ZigBee instead of UPB.

CONCLUSION AND FUTURE WORK

There are many things that we want to do, explore and experiment things with the existing technology to innovate a new one so that our life becomes easier and smoother. Our life is shorter and we do not just want to spend our time in switching ON and switching OFF the heater and lights. We do not want to do monotonous or routine things as a futile exercise. We want to be creative and productive. Plus we want to have a full control of our office or home, even though we may not be physically present. IHS serves this purpose.

The future work of this project would be that it uses ZigBee protocol instead of UPB. The only drawback of ZigBee is security. Since it uses radio signals, it is easy to tap and interrupt. The IHS enabled by ZigBee would look like this[38]:

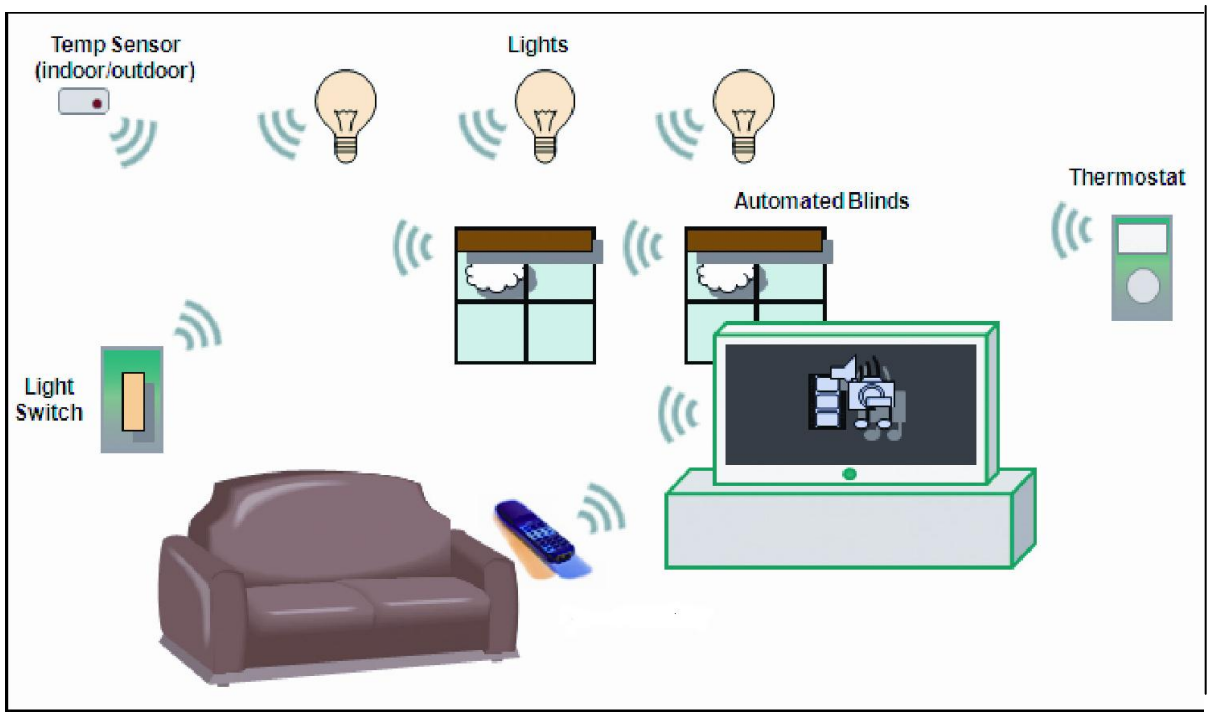


Fig. 5.19. Future Technology in IHS

In this figure, all the lights and switches are controlled wirelessly and blinds and thermostat as well. A single press of a button on the remote control could dim the lights, lower the blinds in order to watch a movie. Similarly, another press could switch ON the air conditioner or

heating and put them on working mode. If no motion is detected at home, all lights could go OFF. To configure the ZigBee network, we may use the TV set or a PC or an i-phone. A WiFi router could provide internet access to ZigBee networks as well.

Why we need to switch over from UPB to ZigBee is that when there are no signals or when there is a noice in the powerline due to heavy electrical motors connected to the same powerline as it has happened to me in the research lab, it cannot communicate with the device. UPB requires noice-free powerlines to operate. The second drawback is that it cannot do multi-hobbing and it cannot also communicate between two buildings. When there are changes in phases, it requires phase coupling device. Lastly, UBP is not a low power (energy sustainability) protocol. The above drawback could be overcome by ZigBee protocol. ZigBee can do multi-hobbing and it does not require phase coupling device if a home has two phases or three phases. Definitely it is more reliable than UPB and it can go between buildings and also between phases. No noice will affect its performance. Security is a major concern for ZigBee.

[38] http://www.zigbee.org/Products/DownloadZigBeeTechnicalDocuments.aspx

APPENDIX A

```
package UPBTesting;
import java.util.*;
import java.io.*;
import javax.comm.*;

/**
 *
 * @author  glenn
 */
public class UpbTest extends javax.swing.JFrame implements Runnable,
SerialPortEventListener {

    private int repeats = 0;
    private byte repeatReq = 0;
    private byte networkId = 0x01;
    private byte pimId = (byte)0xff;
    private String strSerialPort = "COM1";
    private int readIndex = 0;
    private int cmd = 0;
    public static final String KEY_RESULT = "RESULT";
    public static final String KEY_VALUES = "VALUES";

    private byte[] readBuffer = new byte[1000];

    /** Reference to the serialPort to which the UPB controller is connected
     */
    private SerialPort serialPort;
```

```
    /** input stream associated with the serialPort
     */
    private InputStream serialInputStream;

    /** output stream associated with the serialPort
     */
    private OutputStream serialOutputStream;

    private Thread listenerThread;

    private boolean runFlag = false;

    /** Creates new form PLCTest */
    public UpbTest() {
        initComponents();
        repeats = repeats & 0x03;
        repeatReq = (byte)((repeats << 5) & 0x60);
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();
        tfValue = new javax.swing.JTextField();
```

```
jLabel13 = new javax.swing.JLabel();
btWrite = new javax.swing.JButton();
stopListenerButton = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
tranTextArea = new javax.swing.JTextArea();
tfDevice = new javax.swing.JTextField();
jLabel16 = new javax.swing.JLabel();
tfChannel = new javax.swing.JTextField();
modeComboBox = new javax.swing.JComboBox();
btClear = new javax.swing.JButton();
btRead = new javax.swing.JButton();
jLabel17 = new javax.swing.JLabel();
tfNetwork = new javax.swing.JTextField();

addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.WindowEvent evt) {
        exitForm(evt);
    }
});
getContentPane().setLayout(null);

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Channel");
getContentPane().add(jLabel1);
jLabel1.setBounds(30, 160, 136, 20);

jLabel12.setText("Value");
getContentPane().add(jLabel12);
jLabel12.setBounds(80, 210, 60, 20);
getContentPane().add(tfValue);
tfValue.setBounds(30, 230, 134, 25);
```

```
jLabel13.setText("Com");
getContentPane().add(jLabel13);
jLabel13.setBounds(30, 30, 40, 20);

btWrite.setText("Send");
btWrite.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btWriteActionPerformed(evt);
    }
});
getContentPane().add(btWrite);
btWrite.setBounds(30, 340, 138, 28);

stopListenerButton.setText("Exit");
stopListenerButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        stopListenerButtonActionPerformed(evt);
    }
});
getContentPane().add(stopListenerButton);
stopListenerButton.setBounds(30, 380, 138, 28);

tranTextArea.setForeground(new java.awt.Color(0, 0, 204));
tranTextArea.setLineWrap(true);
tranTextArea.setWrapStyleWord(true);
tranTextArea.setMinimumSize(new java.awt.Dimension(700, 15));
tranTextArea.setPreferredSize(new java.awt.Dimension(700, 15));
jScrollPane1.setViewportView(tranTextArea);

getContentPane().add(jScrollPane1);
```

```
        jScrollPane1.setBounds(182, 32, 703, 400);

        tfDevice.setText("1");
        getContentPane().add(tfDevice);
        tfDevice.setBounds(30, 130, 138, 25);

        jLabel16.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel16.setText("Device");
        getContentPane().add(jLabel16);
        jLabel16.setBounds(50, 110, 96, 20);

        tfChannel.setText("0");
        tfChannel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                tfChannelActionPerformed(evt);
            }
        });
        getContentPane().add(tfChannel);
        tfChannel.setBounds(30, 180, 138, 25);

        modeComboBox.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "COM1", "COM2" }));
        getContentPane().add(modeComboBox);
        modeComboBox.setBounds(70, 30, 90, 25);

        btClear.setText("Clear");
        btClear.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btClearActionPerformed(evt);
            }
        });
```

```
        getContentPane().add(btClear);
        btClear.setBounds(30, 260, 138, 28);

        btRead.setText("Request");
        btRead.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btReadActionPerformed(evt);
            }
        });
        getContentPane().add(btRead);
        btRead.setBounds(30, 300, 138, 28);

        jLabel17.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel17.setText("Network");
        getContentPane().add(jLabel17);
        jLabel17.setBounds(50, 60, 96, 20);

        tfNetwork.setText("1");
        getContentPane().add(tfNetwork);
        tfNetwork.setBounds(30, 80, 138, 25);

        setBounds(0, 0, 752, 475);
    }// </editor-fold>//GEN-END:initComponents

    private void btWriteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_btWriteActionPerformed
        // Add your handling code here:
        // Send Serial packet
        cmd = 1; // Write value to device
}//GEN-LAST:event_btWriteActionPerformed
```

```
    private void stopListenerButtonActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_stopListenerButtonActionPerformed
        // Add your handling code here:
        try {
            stopThread();
            System.exit(0);
        }
        catch(Exception e) {
        }
    }//GEN-LAST:event_stopListenerButtonActionPerformed

    /** Exit the Application */
    private void exitForm(java.awt.event.WindowEvent evt) {//GEN-
FIRST:event_exitForm
        try {
            stopThread();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        System.exit(0);
    }//GEN-LAST:event_exitForm

private void tfChannelActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_tfChannelActionPerformed
// TODO add your handling code here:
}//GEN-LAST:event_tfChannelActionPerformed

private void btClearActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_btClearActionPerformed
// TODO add your handling code here:
```

```
    tranTextArea.setText("");
}//GEN-LAST:event_btClearActionPerformed

private void btReadActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_btReadActionPerformed
// TODO add your handling code here:
    cmd = 2; // Read device value
}//GEN-LAST:event_btReadActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        UpbTest t = new UpbTest();
        try {
            t.openAndInitializePort();
            t.runFlag = true;
            t.listenerThread = new Thread(t);
            t.listenerThread.start();
        }
        catch (Exception e){
            e.printStackTrace();
        }
        t.show();
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton btClear;
    private javax.swing.JButton btRead;
    private javax.swing.JButton btWrite;
    private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JComboBox modeComboBox;
private javax.swing.JButton stopListenerButton;
private javax.swing.JTextField tfChannel;
private javax.swing.JTextField tfDevice;
private javax.swing.JTextField tfNetwork;
private javax.swing.JTextField tfValue;
private javax.swing.JTextArea tranTextArea;
// End of variables declaration//GEN-END:variables

public void run() {
   while(runFlag) {
      try {
         Thread.sleep(1000);
         readSerialPort();
         if(cmd == 1) { // Write value to device
            int devId = Integer.parseInt(this.tfDevice.getText());
            int channel = Integer.parseInt(this.tfChannel.getText());
            int value = Integer.parseInt(this.tfValue.getText());
            try {
               if(channel == 0) { // Write to device
                  boolean result = writeDeviceValue(devId, value);
                  if(!result) {
                     // Comms error on device
                     // todo
                  }
               }
```

```
        else { // Write to channel
            boolean result = writeChannelValue(devId, channel, value);
            if(!result) {
                // Comms error on device
                // todo
            }

        }
    }
    catch (Exception e) {
        e.printStackTrace();
    // Comms error with PIM.  Driver down
    // Try to return initialise and on reconnect re-read all IO values
    // todo
    }
    cmd = 0;
}

if(cmd == 2) { // Read value from device
    int devId = Integer.parseInt(this.tfDevice.getText());
    try {
        boolean result = requestDeviceValue(devId);
        if(!result) {
            // Comms error on device
            // todo
        }
    }
    catch (Exception e) {
        e.printStackTrace();
        // Comms error with PIM.  Driver down
        // Try to return initialise and on reconnect re-read all IO values
```

```
                // todo
            }
            cmd = 0;
        }
    }
    catch(Exception e1) {
        e1.printStackTrace();
    }
  }
}

private void updateNetworkId() {
  try {
    networkId = (byte)Integer.parseInt(this.tfNetwork.getText().trim());
  }
  catch (Exception e){}
}

/**
 * Requests the value from a upb device.
 * @param devId - The id of the device
 * @return
 */
boolean requestDeviceValue(int devId) throws Exception{
  updateNetworkId();
  byte[] reg = new byte[6];
  reg[0] = (byte)(0x07 | repeatReq);
  reg[1] = 0x10;
  reg[2] = networkId;
  reg[3] = (byte)devId;
  reg[4] = pimId;
```

```
        reg[5] = 0x30; // for request
        Hashtable valueSet = getValuesFromPim(reg, devId);
        updateValues(valueSet);
        if(valueSet.containsKey(new Integer(devId))) {
            // Value received else comms error with device - todo
            System.out.println((Vector)valueSet.get(new Integer(devId)));
            return true;
        }
        return false;
    }

    /**
     * Accepts a message to be sent.  Adds the Header (Ctrl T), Checksum and CR
     * First reads the port to clear any pending messages
     * Writes the command
     * Waits 500msec for the response
     * Reads the response and returns all Device values in a hashtable
     * Returns the valuesets received in a hashtable as follows
     * KEY = the device id as an Integer.  If the value >= 256 it is a link id
     * VALUE = Vector of Integer values received
     * Waits for an acknowledgement (PA + PK) else does retries.
     * @param message a byte array with the message excluding the Start header
(Ctrl T)
     * and excluding the checksum and CR character.
     * @param devId The device whose values are sought
     * @return A hashtable with
     * KEY = the device id as an Integer.  If the value >= 256 it is a link id
     * VALUE = Vector of Integer values received
     * The hashtable may be empty
     * @throws java.lang.Exception if it does not receive a PA indicating that it
     * cannot communicate with the base unit (PIM)
```

```
 */
Hashtable getValuesFromPim(byte[] message, int deviceId) throws Exception {
    boolean commsFailure = true;
    int len = message.length;
    Hashtable valueSet = new Hashtable();
    // Find the checksum
    byte chkSum = getChecksum(message, len);
    byte[] msg = new byte[len+1];
    System.arraycopy(message, 0, msg, 0, len);
    msg[len] = chkSum;

    // Convert the new message to an ASCII string
    String strMsg = "\u0014" + toAsciiHexString(msg, msg.length) +"\r";

    boolean received = false;
    int retries = 0;
    int result = 0;
    while(!received && (retries < 3)) {
        if(retries > 0) {
            // Wait a bit more for the response
            Thread.sleep(200);
        }
        // Flush anything in the port
        Hashtable returnHash = readSerialPort();
        System.out.println(returnHash);
        if(returnHash.containsKey(KEY_VALUES)) {
            valueSet = (Hashtable)returnHash.get(KEY_VALUES);
            commsFailure = false;
        }
        if(valueSet.containsKey(new Integer(deviceId))) {
            received = true;
```

```
            commsFailure = false;
        }
        else {
            serialOutputStream.write(strMsg.getBytes());
            tranTextArea.append(strMsg+"\n");

            // Write to the port
            System.out.println(strMsg);
            Thread.sleep(500);
            returnHash = readSerialPort();
            // Check if any values were read
            if(returnHash.containsKey(KEY_VALUES)) {
                commsFailure = false;
                Hashtable h = (Hashtable)returnHash.get(KEY_VALUES);
                Enumeration deviceIds = h.keys();
                while(deviceIds.hasMoreElements()) {
                    Integer devId = (Integer)deviceIds.nextElement();
                    valueSet.put(devId, h.get(devId));
                }
            }
            if(valueSet.containsKey(new Integer(deviceId))) {
                commsFailure = false;
                received = true;
            }
            else {
                retries++;
            }
        }
        // Check the Ack/Nck status
        result = ((Integer)returnHash.get(KEY_RESULT)).intValue();
    }
```

```
        if(commsFailure && (result == 2))  {
            throw new Exception("Communication failure");
        }
        return valueSet;
    }

    /**
     * Writes a value to a upb device.  To write to a channel please use
writeChannelValue
     * @param devId - The id of the device
     * @param value - the value to be written - between 0 and 100
     * @return
     */
    boolean writeDeviceValue(int devId, int value) throws Exception{
        updateNetworkId();
        byte[] reg = new byte[7];
        reg[0] = (byte)(0x08 | repeatReq);
        reg[1] = 0x10;
        reg[2] = networkId;
        reg[3] = (byte)devId;
        reg[4] = pimId;
        reg[5] = 0x22; // for write
        reg[6] = (byte)value;
        boolean result = writeToPim(reg);
        return result;
    }

    /**
     * Writes a value to a upb device channel.
     * @param devId - The id of the device
     * @param channelNo - The id of the channel
```

```
     * @param value - the value to be written - between 0 and 100
     * @return
     */
    boolean writeChannelValue(int devId, int channelNo, int value) throws Exception{
        updateNetworkId();
        byte[] reg = new byte[9];
        reg[0] = (byte)(0x0A | repeatReq);
        reg[1] = 0x10;
        reg[2] = networkId;
        reg[3] = (byte)devId;
        reg[4] = pimId;
        reg[5] = 0x22; // for write
        reg[6] = (byte)value;
        reg[7] = (byte)0xFF;
        reg[8] = (byte)channelNo;
        boolean result = writeToPim(reg);
        return result;
    }

    /**
     * Accepts a message to be sent.  Adds the Header (Ctrl T), Checksum and CR
     * Waits for an acknowledgement (PA + PK) else does retries.
     * If successful it returns true else returns false
     * @param message a byte array with the message excluding the Start header
(Ctrl T)
     * and excluding the checksum and CR character.
     * @return true if successful, false on failure after retries
     * @throws java.lang.Exception if it does not receive a PA indicating that it
     * cannot communicate with the base unit (PIM)
     */
    boolean writeToPim(byte[] message) throws Exception {
```

```
Hashtable valueSet = new Hashtable();
int len = message.length;
// Find the checksum
byte chkSum = getChecksum(message, len);
byte[] msg = new byte[len+1];
System.arraycopy(message, 0, msg, 0, len);
msg[len] = chkSum;

// Convert the new message to an ASCII string
String strMsg = "\u0014" + toAsciiHexString(msg, msg.length) +"\r";
tranTextArea.append(strMsg+"\n");
// Flush anything in the port
Hashtable returnHash = readSerialPort();
if(returnHash.containsKey(KEY_VALUES)) {
   valueSet = (Hashtable)returnHash.get(KEY_VALUES);
}
System.out.println(returnHash);

boolean sent = false;
int retries = 0;
int result = 0;
while(!sent && (retries < 3)) {
   // Write to the port
   System.out.println(strMsg);
   serialOutputStream.write(strMsg.getBytes());
   Thread.sleep(500);
   returnHash = readSerialPort();
   tranTextArea.append("ResponseToWrite: "+returnHash+"\n");
   System.out.println("result: "+returnHash);
   // Check if any values were read
   if(returnHash.containsKey(KEY_VALUES)) {
```

```
            Hashtable h = (Hashtable)returnHash.get(KEY_VALUES);
            Enumeration deviceIds = h.keys();
            while(deviceIds.hasMoreElements()) {
               Integer devId = (Integer)deviceIds.nextElement();
               valueSet.put(devId, h.get(devId));
            }
         }
         result = ((Integer)returnHash.get(KEY_RESULT)).intValue();
         if(result == 0) {
            sent = true;
         }
         else {
            retries++;
         }
      }
      if(result == 2) {
         throw new Exception("Communication failure");
      }
      updateValues(valueSet);
      return sent;
   }

   /**
    * Will update the value objects whose values were read.  todo
    * @param valueSet
    */
   void updateValues(Hashtable valueSet) {
      // todo
   }

   /**
```

```
     * Computes the UPB checksum as the 2's complement of all bytes
     * @param b
     * @param size
     * @return
     */
    byte getChecksum(byte[] b, int size) {
        int sum = 0;
        for(int i=0; i<size; i++) {
            int uByte = 0x0ff & b[i];
            sum = sum + uByte;
        }
        sum = ~sum;
        sum = sum & 0x0ff;
        sum++;
        byte bb = (byte)(sum & 0x0ff);
        return bb;
    }

    /** This function is called whenever some event occurs on the serial port
     *  In this case  we are only interested in the event that data is received
     */
    public void serialEvent(javax.comm.SerialPortEvent serialPortEvent) {
        try{
            switch (serialPortEvent.getEventType()) {
                case SerialPortEvent.DATA_AVAILABLE:
                    int numBytes = 0;
                    try{
                        while(serialInputStream.available() > 0){
                            numBytes = serialInputStream.read(readBuffer, readIndex, 1000-
readIndex);
```

```
                    byte[] b = new byte[numBytes];
                    System.arraycopy(readBuffer, readIndex, b, 0, numBytes);
                    String response = new String(b);
                    System.out.println("DATA_AVAILABLE: "+ response+"
************");

                    readIndex = numBytes + readIndex;
                  }
                }
                catch(Exception e1){}
                break;
              default:
                break;
            }
        }
        catch(Exception e){
          e.printStackTrace();
        }
    }

    /**
     *
     * NOTE IT IS THE DUTY OF THE CALLING FUNCTION TO WAIT TILL THE
REPLY IS EXPECTED
     * ELSE IT MAY RECEIVE KEY_RESULT = 2
     * @return Hashtable with keys KEY_RESULT and KEY_VALUES.  KEY_RESULT
will hold
     * an Integer 0 if PA and PK received; 1 if Device did not reply;
     * 2 if PIM did not reply or if exception.
     * KEY_VALUES will hold a hashtable of any device values read in the following
form
```

```
     * The key will be the device id and the value will be a Vector of Integer values
     * returned by the device.  If the device has only a single channel then the
     * vector has a single value, else for a multichannel device, it returns many
     * Integer values.  The device id will be between 1 and 255 for a device and for
     * a Link id 256 will be added to the value.  todo Link is not yet implemented.
     */
    private Hashtable readSerialPort() {
        Hashtable returnHash = new Hashtable();
        Hashtable valueHash = new Hashtable();
        int result = 2;
        // Wait for PA followed by PK
        try {
            // Parse string into individual messages;
            // Each message is terminated by a 0x0d char
            while(serialInputStream.available() > 0){
                int numBytes = serialInputStream.read(readBuffer, readIndex, 1000-
readIndex);
                if(numBytes > 0) {
                    byte[] b = new byte[numBytes];
                    System.arraycopy(readBuffer, readIndex, b, 0, numBytes);
                    String response = new String(b);
                    System.out.println("READ DATA: "+ response+" ************");

                    readIndex = numBytes + readIndex;
                }
            }
            if(readIndex == 0) {
                returnHash.put(KEY_RESULT, new Integer(result));
                return returnHash;
            }
```

```
int startIndex = 0;
byte[] thisMsg = new byte[100];
int thisMsgIndex = 0;
while(startIndex < readIndex) {
   thisMsg[thisMsgIndex] = readBuffer[startIndex];
   if(thisMsg[thisMsgIndex] == 13) {
      // Found CR character - Message Complete
      byte[] b = new byte[thisMsgIndex];
      System.arraycopy(thisMsg, 0, b, 0, thisMsgIndex);
      String response = new String(b);
      tranTextArea.append("Recd: "+response+"\n");
      if(response.equals("PA")) {
         // Message accepted by PIM
         result = 1;
         tranTextArea.append("Accept\n");
      }
      else if(response.equals("PK")) {
         // Message accepted by Device
         result = 0;
         tranTextArea.append("Ack\n");
      }
      else if(response.equals("PE")) {
         // PIM rejected message
         // This should not happen - will return timeout
         result = 1;
         tranTextArea.append("Error\n");
      }
      else if(response.equals("PB")) {
         // PIM is busy - will return timeout
         result = 1;
         tranTextArea.append("Busy\n");
```

```
        }
        else if(response.equals("PN")) {
            // No acknowledgement received from device - return timeout
            result = 1;
            tranTextArea.append("NAK\n");
        }
        else if(response.startsWith("PU")){
            // Translate message received
            tranTextArea.append("Unsolicited Msg\n");
            extractDeviceValues(response, valueHash);
        }
        else {
            tranTextArea.append("Unknown response\n");
            // We seem to get a message from the switch directly via PIM
            // followed by the same message with a PU at the front from
            // the PIM.  The direct message is of the form
            // Dest id (00) Source id(xx) msgDataId (86) val (00) chksum

        }

        // Reset thisMsg pointer
        thisMsgIndex = -1;
    }
    thisMsgIndex++;
    startIndex++;
}
if(thisMsgIndex > 0) {
    // Move over any unread data into buffer
    System.arraycopy(thisMsg, 0, readBuffer, 0, thisMsgIndex);
    readIndex = thisMsgIndex;
    System.out.println("Balance buffer size: "+readIndex);
```

```
        }
        else {
            readIndex = 0;
        }
    }
    catch (Exception e){
    }
    returnHash.put(KEY_RESULT, new Integer(result));
    if(valueHash.size() > 0) {
        returnHash.put(KEY_VALUES, valueHash);
    }
    return returnHash;
}

/**
 *
 * @param stringVal - Must be a string with 2 characters
 * Takes the first 2 characters of this string and transforms
 * it into a single byte.  The chars of stringVal must be any HEX character
 * @return a byte value
 */
byte getByte(String stringVal) {
    if(stringVal.length() != 2) {
        return 0;
    }
    String str = stringVal.toUpperCase();
    int result = toInt(str.charAt(1));
    result = result + (16 * toInt(str.charAt(0)));
    return (byte)result;
}
```

```
int toInt(char c) {
    int result = 0;
    switch(c) {
        case '0':
            result = 0;
            break;
        case '1':
            result = 1;
            break;
        case '2':
            result = 2;
            break;
        case '3':
            result = 3;
            break;
        case '4':
            result = 4;
            break;
        case '5':
            result = 5;
            break;
        case '6':
            result = 6;
            break;
        case '7':
            result = 7;
            break;
        case '8':
            result = 8;
            break;
        case '9':
```

```
            result = 9;
            break;
        case 'A':
            result = 10;
            break;
        case 'B':
            result = 11;
            break;
        case 'C':
            result = 12;
            break;
        case 'D':
            result = 13;
            break;
        case 'E':
            result = 14;
            break;
        case 'F':
            result = 15;
            break;
        default :
            result = 0;
    }
    return result;
}

/**
 * Expects a PU command with the Message Data Id of 86
 * @param response
 * @param valueHash
 */
```

```
void extractDeviceValues(String response, Hashtable valueHash) {
   if(response.length() < 17) {
      return;
   }
   if(!(response.substring(0, 2).equals("PU"))) {
      return;
   }
   if(!(response.substring(12, 14).equals("86"))) {
      return;
   }
   Vector v = new Vector();
   String str = response.substring(2, 4);
   int noOfBytes = getByte(str) & 0x1f;
   int noOfRegisters = noOfBytes - 7;
   str = response.substring(10, 12);
   int devId = getByte(str);
   for(int i=0; i< noOfRegisters; i++) {
      str = response.substring(14+(i*2), 16+(i*2));
      int val = getByte(str);
      v.add(new Integer(val));
   }
   valueHash.put(new Integer(devId), v);
   tranTextArea.append("ValueSet="+valueHash+"\n");
}

public void stopThread() {
   runFlag = false;
}

String toAsciiHexString(byte[] byteArr, int noOfBytes) {
   String str = new String();
```

```
    for(int i=0; i<noOfBytes; i++) {
        byte j = byteArr[i];
        int unsign;
        if(j<0)
            unsign = j+256;
        else
            unsign = j;
        int highNibble = unsign/16;
        int lowNibble = unsign%16;
        str = str + hexChar(highNibble);
        str = str + hexChar(lowNibble);
        //str = str + " ";
    }
    return str;
}

char hexChar(int i) {
    char ch;

    switch(i) {
        case 0:
            ch = '0';
            break;
        case 1:
            ch = '1';
            break;
        case 2:
            ch = '2';
            break;
        case 3:
            ch = '3';
```

```
    break;
case 4:
    ch = '4';
    break;
case 5:
    ch = '5';
    break;
case 6:
    ch = '6';
    break;
case 7:
    ch = '7';
    break;
case 8:
    ch = '8';
    break;
case 9:
    ch = '9';
    break;
case 10:
    ch = 'A';
    break;
case 11:
    ch = 'B';
    break;
case 12:
    ch = 'C';
    break;
case 13:
    ch = 'D';
    break;
```

```
        case 14:
            ch = 'E';
            break;
        case 15:
            ch = 'F';
            break;
        default:
            ch = 'x';
    }
    return ch;
}

private void openAndInitializePort() throws Exception {
    CommPortIdentifier    portId;
    Enumeration           portList;

    portList = CommPortIdentifier.getPortIdentifiers();
    boolean portFound = false;

    while (portList.hasMoreElements()) {
        portId = (CommPortIdentifier) portList.nextElement();
        if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL) {
            if (portId.getName().equals(strSerialPort)) {
                System.out.println("Found port: "+strSerialPort);
                portFound = true;
                //Open the port and cast it to serialport
                try {
                    serialPort = (SerialPort) portId.open(strSerialPort,2000);
                } catch (Exception e) {
                    //setCommStatus(ValueObject.COMMUNICATION_FAILURE);
                    System.out.println(portId.getCurrentOwner());
```

```
                    throw e;
                }
                        //get inputstream and outputstream from the port
                serialInputStream = serialPort.getInputStream();
                serialOutputStream = serialPort.getOutputStream();

                //add serialporteventlistener to get and process events
                serialPort.addEventListener(this);
                //show interest in obtaining events
                serialPort.notifyOnDataAvailable(true);
                //set baud rate and other parameters of serialport
                serialPort.setSerialPortParams(4800, SerialPort.DATABITS_8,
                        SerialPort.STOPBITS_1,
                        SerialPort.PARITY_NONE);
                //set no flow control
                serialPort.setFlowControlMode(SerialPort.FLOWCONTROL_NONE);
                serialPort.setDTR(true);
            }
          }
        }
        if (!portFound) {
          /*
          setCommStatus(ValueObject.COMMUNICATION_FAILURE);
          throw new Exception("Could not find port: " + strSerialPort +
            " Driver: " + name + " stopping");
           * */
        }
      }
}
```

APPENDIX B

```
/*
 * ComputerWiseEP210Driver.java
 *
 * Created on October 22, 2010, 2:14 PM
 *
 * All commands are terminated with a <CR><LF>
 * Gateman          EP210                   Remark
 * Function         Command
 *
 * ReadInputs               INPUTS
 *                  INPUTS=0,1,0,0  Value of the 4 inputs (2 used)
 * Control Relay    RELAY=1,1       Turn Relay1 on
                    RELAY=1,2       Turn Relay1 on for 2 seconds
                    RELAY=1,0       Turn Relay1 off
 *
 * Beep                     BELL
 *
 * MAG Swipe                KEY=12345       Mag card or bar code swipe
 *
 * PROX Swipe               AUX=23456       Input on Serial link
 *
 * Watchdog         WATCHDOG=10     0 => disable 10=> must receive
                                    comms at least 10 secs
 *
 * Connected        CONNECTED               Sent when another conn
                                            established
 *
 * Our driver must implement both KEY and AUX commands as a user swipe.
 *
```

```
 * Default IP = 192.168.168.50
 * Default Port = 1070
 * Default Serial Parms = 9600,N,8,1,1 (Software handshake = on)
 *
 */

package SwipeUnit.Driver;
import java.util.*;
import java.net.*;
import java.io.*;
import NextGen.Access.*;
import NextGen.Base.*;
import NextGen.Utility.*;

/**
 *
 * @author glenn
 */
public class ComputerWiseEp210Driver extends AbstractAccessDriver {

    public static final String CMD_INPUT = "INPUTS";
    public static final String CMD_OUTPUT = "RELAY";
    public static final String CMD_BEEP = "BELL";
    public static final String CMD_SWIPE = "KEY";
    public static final String CMD_SWIPE_AUX = "AUX";
    public static final String CMD_WATCHDOG = "WATCHDOG";
    public static final String CMD_CONNECTED = "CONNECTED";

    boolean ackReceived = false;

    static final int DEFAULT_PORT_NUMBER = 1070;
```

```
static final int DEFAULT_TIMEOUT = 300;
static final int MAX_RETRIES = 3;
static final int TAG_LENGTH = 16;
static final int POLL_TIME = 300;
static int tagRemainder = 0;

InetAddress apIpAddress;
String ipAddressStr;
int apPort;

Socket senderSoc;
OutputStream out;
InputStream in;

String driverIdenStr = "";
String driverIdenWithIpAndPort = "";

/**
 * Will default to a 2 second pulse
 */
int out1Pulse = 2;
/**
 * Set to true when required to beep
 */
boolean beep = false;

/**
 * Will be set to true for -ve logic
 */
boolean out1Negative = true;
boolean in1Negative = false;
```

```
boolean in2Negative = false;

/**
 * Holds the value written by the driver to the EP210
 * Since we cannot read back the value, we assume that what is written is fine
 */
int out1Value = 0;

/**
 * Set to true when the output has been turned on and is waiting to be turned off
 */
boolean out1TurnOff = false;

/**
 * Will hold the time after which the output is to be turned off
 */
NextGenTimestamp out1OffTime;

public static final int DEFAULT_PULSE_TIME = 1;

private NextGenTime loggedErrorHost;

byte readArr[] = new byte[500];

/**
 * Holds swipe data
 */
String cardData = "";

/** Creates a new instance of AccessPointGatemanRFDriver */
public ComputerWiseEp210Driver(Hashtable configHash, AccessPoint
```

```
accessPoint) throws Exception {
        super(configHash,accessPoint);
        driverType = AccessModule.ACCESS_COMPUTERWISE_EP210_DRIVER;
        //In case ipaddress,port number,blocks are not defined in configHash
        //then log the information.
        driverIdenStr = "type="'+driverType+"' ,"+" name="'+name+"'";
        if(!configHash.containsKey(KEY_IP_ADDR))
           throw new Exception(KEY_IP_ADDR+" not found.");
/*       if(!configHash.containsKey(KEY_PORT_NUMBER))
           throw new Exception(KEY_PORT_NUMBER+" not found.");
 */
        if(!configHash.containsKey(KEY_PORT_NUMBER)) {
           configHash.put(KEY_PORT_NUMBER, "1070");
        }

        if(configHash.containsKey(AccessPoint.KEY_OUT1)) {
           // Set pulse time
           Hashtable out1Hash = (Hashtable)configHash.get(AccessPoint.KEY_OUT1);
           if(out1Hash.containsKey(AccessPoint.KEY_PULSE)) {
              Integer pulseTime = (Integer)out1Hash.get(AccessPoint.KEY_PULSE);
              this.out1Pulse = pulseTime.intValue();
           }

           // Set negative logic
           if(out1Hash.containsKey(AccessPoint.TAG_LOGIC)) {
              String logic = (String)out1Hash.get(AccessPoint.TAG_LOGIC);
              // By default this driver already implements negative logic
              if(logic.equals(AccessPoint.LOGIC_NEGATIVE)) {
                 this.out1Negative = false;
              }
              else {
```

```
            this.out1Negative = true;
        }
    }
}

if(configHash.containsKey(AccessPoint.KEY_IN1)) {
    Hashtable in1Hash = (Hashtable)configHash.get(AccessPoint.KEY_IN1);
    // Set negative logic
    if(in1Hash.containsKey(AccessPoint.TAG_LOGIC)) {
        String logic = (String)in1Hash.get(AccessPoint.TAG_LOGIC);
        if(logic.equals(AccessPoint.LOGIC_NEGATIVE)) {
            this.in1Negative = true;
        }
        else {
            this.in1Negative = false;
        }
    }
}

if(configHash.containsKey(AccessPoint.KEY_IN2)) {
    Hashtable in2Hash = (Hashtable)configHash.get(AccessPoint.KEY_IN2);
    // Set negative logic
    if(in2Hash.containsKey(AccessPoint.TAG_LOGIC)) {
        String logic = (String)in2Hash.get(AccessPoint.TAG_LOGIC);
        if(logic.equals(AccessPoint.LOGIC_NEGATIVE)) {
            this.in2Negative = true;
        }
        else {
            this.in2Negative = false;
        }
    }
```

```
        }

        //Get ip address and port number and blocks from configHash and try to set the
        //value of the corresponding attributes of the current object.
        try {
            ipAddressStr = (String)configHash.get(KEY_IP_ADDR);
            ipAddressStr = "129.210.19.109";
            apIpAddress = NextGenUtility.getInetAddress(ipAddressStr);

            String portStr = (String)configHash.get(KEY_PORT_NUMBER);
            apPort = Integer.parseInt(portStr);
            driverIdenWithIpAndPort = driverIdenStr+" "+ipAddressStr+":"+apPort;
            createReadThread();
        }
        catch(Exception e) {
            throw new Exception(e.getMessage()+" for "+driverIdenStr);
        }
    }

    /**
     * @return Integer 0 if there is no user or invalid user else the user id
     * This method must read the string from the AccessPoint reader.  If there is any
data
     * then it must identify the user.  In the event the user is not identified, it must log
     * that there was a failure to identify the user based on the CARD used.
     * This method is required for those drivers that use the run method of the
AbstractAccessDriver
     * class.
     * @throws an Exception in the event there is a communication problem
     */
    Integer getUserAccess() throws Exception {
```

```
      Integer usrId = new Integer(0);
    try {
        ap.amHealthy();
        createSocket();
        String userIdentityString = readUserAtAccessPoint();
        if(userIdentityString != null) {
            //System.out.println("Tag="+userIdentityString);
            usrId = identifyUser(userIdentityString);
        }
        return usrId;
    }
    catch (Exception e){
        //System.out.println("getUserAccess Error");
        //e.printStackTrace();
        throw(e);
    }
}

/**Create a socket if one does not already exist or if there is a comms error */
public void createSocket() throws Exception {
    try {
        if((!ap.getEnabled()) || (Startup.licenseAccess == 0)) {
            try {
                if(senderSoc != null) {
                    senderSoc.close();
                    senderSoc = null;
                }
                if(out != null) {
                    out.close();
                    out = null;
                }
```

```
                if(in != null) {
                    in.close();
                    in = null;
                }
            }
            catch (Exception soc) {
            }
        }
        else {
            if(senderSoc == null) {
                connect();
            }
            else if(driverCommFailure) {
                connect();
            }
            driverCommFailure = false;
        }
    }
    catch(Exception e) {
        if(!driverInitialized) {
            driverInitialized = true;
        }
        driverCommFailure = true;
        //May be rfid unit or switch is down
        //Log exception if 1 hour is over after logging this exception
        if(loggedErrorHost == null) {
            loggedErrorHost = new NextGenTime();

Startup.fl.logError(Startup.ID_MODULE_ACCESS_POINT,driverIdenWithIpAndPort+
            " Unable to connect to EP210 unit.",LOG_ERROR);
        }
```

```
            else {
                NextGenTime currentTime = new NextGenTime();
                if(loggedErrorHost.secondsBetween(currentTime) > 3600L) {

Startup.fl.logError(Startup.ID_MODULE_ACCESS_POINT,driverIdenWithIpAndPort+
                        " Unable to connect to EP210 unit.",LOG_ERROR);
                    loggedErrorHost = new NextGenTime();
                }
            }
            throw e;
        }
    }

    /**
     * Establishes a socket connection and sets the timeouts
     * We should get a CONNECTED response
     * We should clear the read and write buffers
     * Set WATCHDOG=10
     * Shut the output
     */
    private void connect() throws Exception {
        try {
            senderSoc = new Socket(apIpAddress,apPort);
            senderSoc.setSoTimeout(DEFAULT_TIMEOUT);
            out = senderSoc.getOutputStream();
            in = senderSoc.getInputStream();

            // WE SHOULD SET WATCHDOG = 10
            out.write(("WATCHDOG=10\r\n").getBytes());
            Thread.sleep(200);
            // send twice because sometimes there is some garbage 1st time
```

```
            out.write(("WATCHDOG=10\r\n").getBytes());
            // Should receive CONNECTED response - dont care - just clear the buffer
            in.read(readArr);
            //System.out.println("***************connect****************");
            // Shut Output
            int value = adjustLogic(out1Negative, 0);
            // Write the value
            setOutput(value);
            out1Value = 0;
        }
        catch (Exception e){
            //e.printStackTrace();
            throw new Exception(e);
        }
    }

    /**
     * @return String with the userIdentityString (string read from access point specific
to the card
     * or null if there is no pending data.
     * Tye actual read of the data is done in the readValuesFromHardware routine
     * and stored in the variable cardData
     * The Access String is of the form
     * KEY=12345\r\n
     * AUX=12345\r\n
     */
    String readUserAtAccessPoint() throws Exception {
        if(cardData.length() > 0) {
            String result = cardData.replace("\r\n", "");
            cardData = "";
            return result;
```

```
        }
        return null;
    }

    /**
     * This method must write the values that have been set by the system into the
hardware
     * It can throw an exception in the event of a comms failure
     * If the 'PULSE' condition has been enabled, then it must check if it is waiting to
turn
     * an output automatically off after it has been turned on
     * @param hash Hashtable Contains the keys for the parameters to be set
supported by the driver
     */
    void writeValuesToHardware(Hashtable hash) throws Exception{
        // Are and outputs waiting to be set
        // Check OUT1
        if(hash.containsKey(AccessPoint.KEY_OUT1)) {
            String valStr = (String)hash.get(AccessPoint.KEY_OUT1);
            int valueToSet = Integer.parseInt(valStr);
            int value = adjustLogic(out1Negative, valueToSet);
            // Check if the value written was 1 and if PULSE output required set the off
time
            if(out1Pulse > 0) {
                if(valueToSet == 1) {
                    out1TurnOff = true;
                    beep = true;
                    out1OffTime = new NextGenTimestamp();
                    out1OffTime.addSeconds(out1Pulse);
                }
            }
```

```
            // Write the value
            setOutput(value);
            // Assume this value is written
            out1Value = valueToSet;
        }
        // Are any outputs waiting to be reset (pulse)
        NextGenTimestamp now = new NextGenTimestamp();
        // Check OUT1
        if(out1TurnOff && out1OffTime.before(now)) {
            int value = adjustLogic(out1Negative, 0);
            // Write the value
            setOutput(value);
            out1TurnOff = false;
            out1Value = 0;
        }

    }

    /**
     * This method must read the values that have been read by the hardware - other
than the User
     * It can throw an exception in the event of a comms failure
     * @return Hashtable
     * The key must be one of the keys for the parameters read by the driver, other
     * than the user id which is handled separately
     * Typically the keys are KEY_IN1, KEY_IN2, KEY_OUT1
     */
    Hashtable readValuesFromHardware() throws Exception {
        Hashtable hash = new Hashtable();
        int in1=0, in2=0;
```

```
        // Send the Read input values command
        out.write(("INPUTS"+"\r\n").getBytes());
        // Wait a while
        int retries=0;
        int bytesRead=0;
        boolean dataAvailable = false;
        while(!dataAvailable && (retries < 5)) {
           Thread.sleep(100);
           retries++;
           // Read the network data and parse for input values and AUX and KEY
values
           bytesRead = in.read(readArr);
           if(bytesRead == -1) {
              throw new Exception("EP210 connection broken");
           }
           if(bytesRead > 0) {
              dataAvailable = true;
           }
        }
        if(!dataAvailable) {
           throw new Exception("EP210 data not available");
        }
        // Parse the data read
        byte sub[] = new byte[bytesRead];
        System.arraycopy(readArr, 0, sub, 0, bytesRead);
        String data = new String(sub);
        String responseArr[] = data.split("\r\n");
        int ind;
        for(int i=0; i<responseArr.length; i++) {
           String thisStr = responseArr[i];
           if(thisStr.indexOf("INPUTS=") >= 0) {
```

```
            //System.out.println(thisStr);
            // input values found
            ind = thisStr.indexOf("=");
            String val = thisStr.substring(ind+1);
            String input[] = val.split(",");
            in1 = Integer.parseInt(input[0].trim());
            in2 = Integer.parseInt(input[1].trim());
        }
        else if(thisStr.indexOf("KEY=") >= 0) {
            // Swipe card data found
            // If multiple swipe data found we keep only the last
            ind = thisStr.indexOf("=");
            cardData = thisStr.substring(ind+1);
        }
        else if(thisStr.indexOf("AUX=") >= 0) {
            // Swipe card data found
            // If multiple swipe data found we keep only the last
            ind = thisStr.indexOf("=");
            cardData = thisStr.substring(ind+1);
        }
    }

    // Output that was written
    hash.put(AccessPoint.KEY_OUT1, new Integer(out1Value));

    // Adjust input for logic
    int value = adjustLogic(in1Negative, in1);
    hash.put(AccessPoint.KEY_IN1, new Integer(value));

    value = adjustLogic(in1Negative, in2);
    hash.put(AccessPoint.KEY_IN2, new Integer(value));
```

```
        return hash;
    }

    /**
     * Sends OUTPUT=1,value
     * Sets the value of the output
     * @param value 0 or 1
     */
    private void setOutput(int value)  throws Exception {
        out.write(("RELAY=1,"+value+"\r\n").getBytes());
        if(beep) {
            out.write(("BELL"+"\r\n").getBytes());
            beep=false;
        }
    }

}
```

ACRONYMS

ACE	Ancillary Control Equipment
ADSL	Assymetric Digital Subscriber Line
AIFF	Audio Interchange File Format
AMR	Automated Meter Reading
APS	Application Support Sub-layer
ASCII	American Standard Code for Information Interchange
BWF	Broadcast Wave Format
CCA	Clear Channel Assessment
CCTV	Closed Circuit Television
CDMA	Code Division Multiple Access
CDR	Clock and Data Recovery
CEntOS	Community Enterprise Operationg System Version 5.3
CIDR	Classless Inter-Domain Routing
CIE	Control and Indicating Equipment
CLK	Clock
CMOS	Complementary Metal Oxide Semiconductors
CPA	Chosen-Plaintext Attack
CR	Core Routers
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DAS	Direct Attached Storage
DCF	Distributed Co-ordination Function
DHCP	Dynamic Host Configuration Protocol
DID	Destination ID
DMZ	Detection of Motion Zone
DNS	Domain Name System
DSL	Digital Subscriber Line
DSR	Dynamic Source Routing Protocol
DTMF	Dual-Tone Multi-Frequency signaling
DVR	Digital Video Recorder
DTR Line	Data Terminal Ready Line
EAP	Extensible Authentication Protocol
ED	Energy Detection
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESID	Extended Set ID
Gbps	Giga bits per second
GHz	Giga Hertz [10^9 Hertz]
GIS	Geographic Information System
GPRS	General packet Radio Service
GUI	Graphical User Interface
HA	Home Automation

HAN	Home Area Networking
HDLC	High-level Data Link Control
HDSL	High-bit-rate Digital Subscriber Line
HDTV	High-Definition Television
HP	Hewlett-Packard
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HVAC	High Voltage Alternating Current
IAS	Intruder Alarm Systems
IC	Integrated Circuits
ICM	Input Control Module
ICR	Identification and Command Recognition
ID	Identification / Identity / Identifier
IDTs	Inter-Digital Ttransducers
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGMPv2	Internet Group Management Protocol version 2
IHS	Integrated Home Server
IMAP	Internet Message Access Protocol
IMD	Implantable Medical Devices
IMS	IP Multimedia Sub-systems
IN	Intelligent Network
IND-CCA	Indistinguishability - Chosen- Ciphertext Attack.
IND-CPA	Indistinguishability - Chosen-Plaintext Attack.
IP multicast	Internet Protocol Multicast
IP	Internet Protocol
IPS	Internet Protocol Security
IPSec	Internet Protocol Security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IS	Inter Switch
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
IT	Information Technology
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LAP	Link Access Procedures
LAPB	Link Access Protocol Balanced
LBS	Location Based Services
LDO	Low Drop-out Regulator
LinuxMCE	Linux Media Center Edition
LKH	Logical Key Hierarchy
LLC	Logical Link Control

LQI	Link Quality Indication
LWAPP	Lightweight Access Point Protocol
MAC	Multiple Access Channel
MAN	Metropolitan Area Network
Mbps	Mega bits per second
MCGP	Media Gateway Control Protocol
MDA	Message Data Arguments
MDID	Message Data ID
MDT SAFI	Multicast Distribution Trees Sub-address Family Identifier
MEGACO	Media Gateway Control Protocol
MHS	Message Handling Services
MHz	Mega Hertz [10^6 Hertz]
MIBs	Management Information Bases
MID	Message ID or Manufacturer ID
MIME	Multipurpose Internet Mail Extensions
MIMO	Multi-Input Multi-Output
MPEG	Moving Picture Experts Group
MSID	Message Sent ID
NAS	Network Attached Storage
NIC	Network Interface Card
NICr	Network Interface Controller
NID	Network ID
NOS	Network Operating Systems
OCM	Output Control Module
OS	Operating System
PC	Personal Computer
PCS	Powerline Control Systems
PDA	Personal Digital Assistant
PDC	Primary Domain Controller
PIM	Powerline Interface Module
POP	Post Office Protocol
PPM	Pulse Position Modulation
RDT	Real Data Transport
RFID	Radio-Frequency Identification
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SAN	Storage Area Networks
SCCP	Skinny Call Control Protocol
SCU	Santa Clara University
SID	Source ID
SIP	Session Initiation Protocol
SMB / CFIS	Server Message Block / Common Internet File System
SMEs	Small and Medium Enterprises
SMS	Short Message Service

SMTP	Simple Mail Transfer Protocol
SOHO	Small Office / Home Office
SSH	Server SHell
SSN	Secure Service Network
TBD	To Be Determined
TCP / IP	Transfer Control Protocol / Internet Protocol
TCP	Transfer Control Protocol
TEK	Traffic Encryption Key
TLS	Transport Layer Security
TNC	Trusted Network Computing
TPDU	Transport Protocol Data Unit
TS	Translation Servers
TSAP	Transport Service Access Print
UDP	User Datagram Protocol
UID	Unit ID
UPB	Universal Powerline Bus
URL	Uniform Resource Locator
USB	Universal Serial Bus
VCR	Video Cassette Recorder
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding
VTC	Virtual Terminal Command
WAN	Wide Area Network
WAV	Waveform Audio File Format
WCNC	Wireless Communication and Networking Conference
WD	Warning Device
WiCom	Wireless Communications
WID	Wireless Intrusion Detection
WiFi	Wireless Fidelity (IEEE 802.11b wireless Networking)
WiMax	Worldwide Interoperability for Microwave Access (IEEE 802.16 wireless broadband standard)
WIP	Wireless Intrusion Prevention
WLAN	Wireless Local Area Network
WLoc	WLAN Location-based Services Integrated System
WMM	Wi-Fi Multimedia
WMPLS	Wireless Multi Protocol Label Switching
WNIC	Wireless Network Interface Controller
WPA	Wireless Protected Access
WPAN	Wireless Personal Area Network
WWW	World Wide Web
XCAP	Extensible Markup Language Configuration Access Protocol
ZCL	ZigBee Cluster Library

BIBLIOGRAPHY

A. PRIMARY SOURCES

1. Agrawal D. And Zeng, Q. *Introduction to Wireless and Mobile Systems.* Pacific Grove, CA, NJ: Brooks / Cole Thomson Learning, 2008.
2. Bruce Eckel, *Thinking in JAVA*, 3rd edition (Pearson Education, 2009)
3. C. Thomas Wu, *An Introduction to Object-Oriented Programming with JAVA*, 4th edition (Tata McGraw-Hill, 2009).
4. Cay S. Horstmann and Gary Cornell, *Core JAVA 2*, 7th edition (Pearson Education, 2009).
5. Cay S. Horstmann, *Big JAVA*, 2nd edition (John Wiley, 2008)
6. Cheswick, W., Bellovin, S., and Rubin, A. *Firewalls and Internet Security.* Reading, MA: Addition-Wesley, 2009.
7. Comer, D. *Computer Networks.* Upper Saddle River, NJ: Prentice Hall, 2009.
8. Comer, D. *Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture.* Upper Saddle River, NJ: Prentice Hall, 2008.
9. Daug Lowe, Joel Murach, and Andrea Steelman, Murach's *Beginning JAVA 2, JDK 5* (SPD, 2009).
10. David Flanagan, *JAVA in Nutshell* (O'reillly Media, 2009).
11. E. Balagurusamy, *Programming with JAVA: A Primer*, 3rd edition (Tata McGraw-Hill, 2009).
12. Forouzan, B. *Data Communications and Networking.* NY: McGraw-Hill, 2008.
13. Harvey Deitel and Paul Deitel, *JAVA How to Program*, 5th edition (Pearson Education, 2008).
14. Herbert Schildt, *The Complete Reference Java*, 7th edition (Pearson Education, 2009).

15. http://www.centos.org/

16. http://www.linux.com/

17. http://www.linux-books.us/centos_0007.php

18. http://www.mygateman.org/joomla/index.php

19. Jonnni Kanerva, *The Java FAQ* (Pearson Education, 2008).

20. Kumar A., Manjunath, D., and Kuri, J. *Communication Networking.* San Francisco, CA: Morgan, Kaufmans, 2008.

21. Kurose, J. And Ross, K. *Computer Networking.* Reading, MA: Addison-Wesley, 2008.

22. Liedtke, J. *Improving IPC by Kernel Design," Proc. 14th Symp. on Operating Systems Principles,* ACM, pp. 175-188, 1993.

23. Liedtke, J. *On Micro-kernel Construction, Proc. 15th Symp. on Operating Systems Principles.* ACM, pp. 237-250, 1995.

24. McKusick, M.K., and Neville-NeiL, G.V.: *The Design and Implementaion of the FreeBSD Operating System,* Reading, MA: Addison-Wesley, 2004.

25. Negus, Christober. *RedHat Linux Bible 9.* Willey Publishing Inc. Indianapolis, IN, 2009.

26. Patterson, D., and Hennessy, J.: *Computer Organization and Design,* 3rd ed., San Francisco: Morgan Kaufman, 2008.

27. Peterson, L;, and Davie B. *Compuer Networks: A Systems Approach.* San Francisco, CA: Morgan, Kaufmans, 2009.

28. Stallings, W. *Data And Computer Communications.* Upper Saddle River, NJ: Prentice Hall, 2000.

29. Stallings, W. *High Speed Networks.* Upper Saddle River, NJ: Prentice Hall, 2008.

30. Stallings, W. *Wireless Communications and Networks.* Upper Saddle River, NJ: Prentice Hall, 2009.

31. Tanenbaum, A. *Computer Networks.* U Upper Saddle River, NJ: Prentice Hall, 2008.

32. Tanenbaum, A.S., *Modern Operating Systems.* 3rd edition, Upper Saddle River, NJ: Prentice Hall, 2008.

33. www.mygateman.org

B. SECONDARY SOURCES

34. http://www.homeseer.com/automation_basics.htm
35. http://wiki.centos.org/HowTos/Custom_Kernel
36. http://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol
37. http://www.elkproducts.com/pdf/M1XSP_UPB_details.pdf
38. http://www.smarthomeusa.com/Common/UPB/UPBdescription.pdf
39. http://en.wikipedia.org/wiki/Universal_powerline_bus
40. http://en.wikipedia.org/wiki/X10_(industry_standard)
41. http://en.wikipedia.org/wiki/Z-Wave
42. http://en.wikipedia.org/wiki/Insteon
43. http://pulseworx.com/Downloads.htm
44. http://acronyms.thefreedictionary.com/issue+date
45. Williams, J.C., 2000, "Strategic Trends in Device Networking," Harbor Research, jwilliams@harborresearch.com
46. http://en.wikipedia.org/wiki/Christogram
47. http://www.dedoimedo.com/computers/centos.html
48. http://computer.howstuffworks.com/internet/basics/streaming-video-and-audio.htm
49. http://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol
50. http://www.fadfusion.com/selection.php?product_item_number=20161201859
51. http://power-home.info/wiki/index.php?title=PowerHome2
52. http://www.homeseer.com/products/software/hs_software.htm
53. http://www.hometheatermag.com/accessories/206control4/
54. http://courses.engr.illinois.edu/ece445/projects/spring2009/project15_proposal.pdf
55. http://www.silicon-press.com/briefs/brief.nas/

56. http://wiki.answers.com/Q/Samba_server_and_ftp_server_difference
57. http://www.webopedia.com/TERM/H/HTTP.html
58. http://www.cyberindian.com/web-hosting/article.php?article_id=88
59. http://whatis.techtarget.com/definition/0,,sid9_gci212840,00.html
60. http://squidguard.mesd.k12.or.us/
61. http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci846792,00.html
62. http://zigbee.org/Products/DownloadZigBeeTechnicalDocuments.aspx
63. http://www.rabbit.com/documentation/docs/manuals/ZigBee/Introduction/index.htm
64. http://www.sena.com/products/industrial_zigbee/zs10.php