Acknowledgement

We would like to thank Dr. Luc Vandeurzen for making this thesis possible and introducing the subject. He supported us and gave useful insight into the studied matter. His indispensable personal guidance has proved to be an asset for this thesis.

Moreover, we would like to thank our families and friends who supported us throughout the making of this thesis. They gave moral support and gave us the possibility of using their infrastructure for our testing purposes.

We wish to express our warmest thanks to Tom and Jelle who were there during the whole process. The pleasant and sometimes hilarious interactions lead to some creative solutions.

We are also grateful to the makers of the open source programs we used to test the HomePlug devices and typeset this text. In particular we like to mention LaTeX and *The GIMP*. Without these tools no such result would be possible.

The financial support of the GroupT engineering school is gratefully acknowledged. It gave us the possibility to purchase and test several HomePlug devices.

Contents

List of Figures

List of Tables

Summary

HomePlug is a new technology that enables high-speed networking over the domestic powerlines. It allows home owners to set up a home-network without the need of rewiring the whole house with UTP cables. This thesis answers the question whether this new HomePlug technology is able to serve as a backbone for a modern multimedia home-network.

The HomePlug Alliance has published several specifications that enable devices to use the powerlines to access the home-network. A first standard, the HomePlug 1.0 standard, allows communication at 14 Mbps at the physical layer. An evolution of this standard, the HomePlug AV standard, allows communication at 200 Mbps at the physical layer. Devices from manufacturers that comply with these standards are interoperable with each other. The existance of a standardising body is important for consumers. Along with these two standards, the HomePlug Alliance has published two other standards (HomePlug CC and HomePlug BPL), but these are not studied here. Intellon has specified an extention of the HomePlug 1.0 standard which enables speeds up to 85 Mbps at the physical layer. Devices that are based on this chipset are used in this thesis.

The technology is based on OFDM (*Orthogonal frequency-division multiplexing*). With this technique the frequency spectrum is divided into several narrowband channels. This way one gets many flat frequency responses and equalisation of the whole spectrum isn't needed anymore. Both DBPSK and DQPSK can be used to modulate the channels. DQPSK is more efficient since it can convey four data-bits per symbol, while DBPSK can only trasmit two data-bits. To ensure data integrity and error correction the data is coded using FEC (*Forward Error Correction*). Another enhancement for communication over powerlines is the ability to (de)select the used narrowband channels. This way, the HomePlug devices can avoid heavily impaired frequency's. At the MAC layer HomePlug uses CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Because detecting a collision is difficult and would result in slow communication, *Collision Detection* (as used with ethernet) wouldn't be efficient. Collision avoidance, on the other hand, requires more intelligence from the HomePlug devices but results is faster communication.

Since the HomePlug signals can be detected 200 meters from the sending device, security is a big issue. Neighbouring houses connected to the same transformer would be able to eavesdrop on the HomePlug communication. To solve this problem, HomePlug enables devices to use encryption for communication. However, this requires the assignment of a shared password to each HomePlug device in the installation process.

QoS is handled by dividing traffic into four categories. Network traffic that requires low latency and jitter is assigned Priority 3, while other traffic is assigned other priorities according to the needed QoS.

In order to understand how HomePlug signals traverse powerlines, a closer look was taken to the physical part of powerlines in general. The generation and distribution of electricity was studied. Since HomePlug signals are rather susceptible to noise, a study was carried out. Many household devices were tested on their potential harm to these signals. Only three of them attracted attention: the mobile phone charger, the drill and the electrical heater.

HomePlug has no problems crossing phases whatsoever. The neighbours, when in close proximity, are theoretically able to intercept private traffic although that should not form a problem because of the unique link that is established between two endpoints and because of the encryption.

To protect HomePlug signals from noise or protect private and partial traffic from the environment, a filtering topic was discussed. Tests were done in order to obtain full-duplex filtering. A good unambiguous solution was not found but we were able to suppress the HomePlug signal to 50% of its original level. It became clear that also powerstrips (the more expensive ones with transient voltage suppressors) were good filters for the HomePlug signals. They should however be avoided when using HomePlug devices.

When evaluating HomePlug in a company environment it became clear that this technology should best be used in smaller, specific areas for special purposes rather than using it to cover a whole company. The best use – as the name of the technology already implies – is within a home environment.

To see how HomePlug networks perform while multicasting a configuration was set up to measure multicast performance. This exposed a problem that could become very important in a home-network. Since HomePlug devices switch to a more redundant but slower mode (ROBO-mode) the devices only deliver a multicast throughput of 580 kbps, only a fraction of the throughput while unicasting. The Devolo dLAN Highspeed adapters use another approach. Here the multicast frames are unicasted to each multicast-subscriber separately. This means that more subscribers will lower the available throughput. The Devolo AV adapters also switch to a less performant mode that allows multicast communication at 820 kbps. Broadcast frames are handeled the same way, but since no huge amounts of data have to be broadcasted, no problems arise here.

The way HomePlug networks can coexist with eachother is another important factor. It is observed that the available throughput isn't divided honestly, and that a HomePlug network that already has lower throughput available, suffers even more when another HomePlug network is present on the same powerlines. Coexistance is possible, but not optimal. The same holds when HomePlug 1.0 and AV networks are combined. The two networks can coexist, but the available throughput will drop to 50% or even 30% of the original throughput.

To get to know the capabilities and limits of HomePlug networks, several tests were performed in a real-life situation. Jitter, throughput and latency were measured. It was observed that a whole house can easily be covered by a HomePlug network. Moreover, the QoS is sufficient to allow demanding applications (such as VoIP and streaming video) to use this network as a backbone. 80% of all outlet pairs allowed a throughput of minimum 15 Mbps when the Devolo dLAN Highspeed adapters (adapters with the turbo extention) were used.

To obtain a clear overall picture of the HomePlug technology and to investigate whether HomePlug can be used as a reliable backbone, user scenarios were made. Many popular, modern applications were addressed and taken into account when making these use cases. It became clear that HomePlug encounters few problems when working with these applications and protocols. The most important bottleneck for HomePlug Turbo (85Mbps) is its bandwidth but this is improved thanks to the HomePlug AV standard (200Mbps).

The range of a HomePlug network can be extended. This way, a range of 400 meters, instead of 200 meters can be covered by a HomePlug network. Two methods were tested. A HomePlug range extender based on a linux device and one HomePlug adapter and a range extender making use of two HomePlug adapters connected by means of a simple UTP cable. Both solutions perform equally well, but have their advantages and disadvantages. Although the range of the network can be extended using these solutions, throughput will be only a fraction of the throughput obtained in better conditions.

To use HomePlug as a real backbone of the home-network, the powergrid can be divided into several segments. This would increase the performance of the HomePlug network. The way the powergrid can be segmented is similar to the way ethernet networks are segmented using a switch. Since the HomePlug signals don't propagate throughout the whole powergrid, beter usage of the home-network is expected because the collision domain is reduced.

1 Introduction

Home networking is evolving at a quick pace. In today's modern society where new technologies arise and ubiquitous (broadband) communication is needed, networking plays an important role. Our focus here lies on in-home networking. Many technologies exist but ethernet and wireless technologies are the most popular ones. However there is a good competitor making his way to the public: PowerLine Communication (PLC). PowerLine Communications technology allows the usage of electrical power supply networks for communication purposes and, today, also broadband communication services. The main idea behind PLC is the reduction in operational costs and expenditure for realization of new telecommunication networks.

HomePlug vs PLC

PLC technology is nothing new and is used for example by electricity companies for remote measuring purposes, control tasks and for internal communications of electrical utilities. PLC is a more general term for technologies that use powerlines to communicate. PLC technologies have been around a long time. *X10* for example is a PLC technology that is used to automate houses and has been on the market for a long time. *CAN-bus* is a system to communicate over the powergrid in vehicles. This technology allows devices to extract energy from the grid, and to communicate with other devices over the same grid. This is common to all PLC technologies. They allow devices to communicate over the same wires that provide them with power. Other examples of PLC technologies are *LonsWorks* and *CEBus*. These technologies, however, are unable to provide broadband communication over powerlines. Generally, we can divide PLC systems into two groups: narrowband PLC allowing communication services with relatively low data rates (up to 100 kbps) and ensuring realization of various automation and control applications as well as a few voice channels, and broadband PLC systems (BPL) allowing data rates beyond 2Mbps and, accordingly, realization of a number of typical telecommunication services in parallel, such as telephony and internet access. BPL in low-voltage supply networks (coming into our homes) seems to be a cost-effective solution for "last mile communications networks, the so-called PLC access networks. Due to the many activities concerned with the developement of PLC technology in the access area, higher data rates can be established. To obtain higher rates, PLC systems are supposed to use the frequency range up to 30Mhz, which is also used by various radio services. Unfortunately, a PLC network acts as an antenna producing electromagnetic radiation in its environment and disturbs other services operating in the same frequency range. Therefore, the regulatory bodies specify very strong limits regarding the electromagnetic emission from the PLC networks, with a consequence that PLC networks have to operate with a limited signal power. This causes a reduction of network distances and data rates and increases sensitivity to disturbances. These, and other, problems regarding PLC technology will also be discussed in this thesis.

The terms HomePlug and PLC are no synonyms. HomePlug is a standard that defines how broadband communication over powerlines should be implemented. The latest HomePlug standard, HomePlug AV, promises speeds up to 200 Mbps at the physical layer. This is enough for most in-home applications today, although we must keep in mind that there will always be an increasing demand for bandwidth and QoS.

Some manufacturers have created their own solution for broadband communication over powerlines. They don't comply with the HomePlug standard, but offer a comparable throughput.

The HomePlug Alliance

The HomePlug Alliance was founded in 2000. It is supported by numerous companies like Comcast, Earthlinke, GE, Intel, Linksys, Cisco, Motorola, Radio Shack, Samsung and Sony. In June 2001 the alliance published the HomePlug 1.0 standard that must prevent a morbid growth of incompatible protocols. HomePlug has become the leading standard for broadband powerline communications.

Products that comply with the HomePlug standard are recognized by the HomePlug logo

Figure 1.1: HomePlug logo

(see figure 1.1). Devices that comply with the standard have no problem to communicate with each other. This is very importand for consumers. They are not forced to stick to one manufacturer of HomePlug devices. When a user wants to expand his/her HomePlug network, he/she can buy every device that complies with the HomePlug standard.

So far, the HomePlug Alliance has published four standards for powerline communication [1]:

- HomePlug 1.0,
- HomePlug AV,
- HomePlug CC and
- HomePlug BPL.

HomePlug 1.0 devices deliver a maximal throughput of 14 Mbps. The original goal of this first HomePlug standard was to offer broadband access to devices that are hard to reach. It was designed for internet applications like browsing, e-mail, ftp, ... Although it can be used to set up a network with more than two devices, it was meant to be used in a point-to-point configuration.

The HomePlug AV standard is an evolution of the 1.0 standard and promises a througput up to 200 Mbps. It addresses the higher requirements for multimedia communication. More attention is paid to QoS issues, so it is usable for transmitting HDTV (high definition television) and VoIP (Voice over IP) data. These applications demand more bandwidth and a better QoS. These enhancements make the AV standard more suitable as a true backbone for the home-network.

HomePlug CC stands for HomePlug Command and Control. It is a slow and cheap technology. It allows communication with simple in-home devices like lamps and alarms. Communication with such devices doesn't demand much bandwidth. It allows communication that isn't multimedia related.

HomePlug BPL (broadband over powerline) is a workgroup that studies the possibility to allow broadband internet access through the powerlines that connect houses to the main electrical grid. BPL could become a serious competitor for xDSL and cable internet access. Right now there are some BPL setups in America and Europe, but this technology is still in an early stage. To convey communication over the public electrical grid additional problems arise, however, this is beyond the scope of our study.

This study will focus on the HomePlug 1.0 and HomePlug AV standard. Other HomePlug standards (CC and BPL) and proprietary solutions won't be studied.

The title explained

The title of our thesis project is actually a question that we will ask ourselves throughout this thesis and on which this thesis is based: could we use this existing communication technology as a reliable backbone for the existing and future home network? The purpose of this thesis is to solve this question and to provide an anthology of structured tests and their conclusions.

One should take careful notice of the words 'Home-network' and 'Backbone'. However, a short section will be dedicated to it, the meaning of this text is not to investigate the HomePlug technology's behaviour in a large company environment but will be limited to in-home networking only. Also it is important to know that HomePlug will be studied from a backbone point of view. We are not to describe this technology as an extension or total solution for replacement of existing network technologies. The focus lies on the backbone or support qualities of HomePlug in order to function as a communication technology where other technologies can relie on. So actually it is the other way around, no longer we take the popular ethernet as backbone but now we take ethernet and other technologies (e.g wireless) as extensions to the basic HomePlug network.

2 HomePlug technology in a nutshell

To understand how a HomePlug network works, one needs to get familiar with the underlying technology. The problems that arise when using powerlines as a communication medium, along with their solutions, will be described. This chapter describes the low-level concepts used at the physical layer. In the following chapters we will climb to the higher layers of the OSI network stack, starting here at the bottom.

2.1 The technology

The powerlines are made to transport energy in the form of a 50Hz sinusoid. The requirements for such a network differ greatly from the requirements to set up a reliable broadband data communication network. The HomePlug Alliance has set up standards to create such a network over this medium. The techniques used are both efficient and economical feasible.

2.1.1 Location on the OSI network stack

A HomePlug device, for example the *powerline-to-ethernet bridge*, is what we call a layer two device. It only operates on the two bottom layers of the OSI network stack. This means that the HomePlug device can't *see* IP addresses, but only works with the MAC addresses. Another device that also works on layer two of the OSI network stack is a switch, which is only concerned about MAC addresses, and doesn't care about IP or any layer above it. A router is an example of a layer three device. This device makes decisions on the third layer: where must a packet with destination x.x.x.x go? The devices we use are the so called powerline-to-ethernet bridges.

2.1.2 Problems with PLC

The four main problems with powerline communication are the varying impedance, the noise, privacy and interference with amateur radio. The varying impedance will be explained first.

Varying impedance

Powerlines are a bad medium for communication. One of the biggest difficulties is the frequency dependent impedance. This impedance, that is already frequency dependant, also varies in time. The transfer function (both amplitude and fase) is irregular. This is caused by the almost random joining of different metals for the powerlines. A communication medium should be properly terminated to avoid reflections, but in the case of powerlines there is no such termination. Some parts of the frequency spectrum are impaired for communication, or even not usable because the attenuation of the signals in this region is so large that signals can't be separated from the noise.

The characteristics of the powerlines can vary quickly with time. Devices are plugged in or are being switched on and off. Devices like switching power supplies and electric motors can cause a fast varying impedance of the electrical grid.

This irregular impedance could be neutralized by a complicated equalisation of the used frequency spectrum, but the hardware for such an equalisation would be too complex, and thus too expensive. The HomePlug 1.0 standard solves this problem with Ortogonal Frequency-Division Multiplexing (OFDM) which will be explained later.

Noise

An other common source of problems is the noise present on powerlines. This noise can be generated, for example, by a halogen lamp, a switching power supply or a dimmer. This hostile environment for communication signals can cause many bit errors, resulting in a large bit error rate (BER) at the lowest layer of the protocol. These errors can be detected or rectified by using techniques like Forward Error Correction (FEC), interleaving, fault detection techniques and Automatic Repeat Request (ARQ). By using these techniques, the communication channel is reliable for higher layers in the OSI network stack.

Privacy

Households are being provided with power through a transformer that makes the connection between the medium voltage grid to the low voltage grid. Signals originating from HomePlug devices can't *cross over* to the medium voltage grid. However, all households that get their energy through the same transformer can *see* these signals. In an apartment block this is an even bigger problem. Distance is often a natural barrier that blocks these signals, but in the case where two households are located close enough, privacy can become an important issue. An article in Linux Magazine [3] describes how data could be exchanged with HomePlug devices located two houses down the street. That's why the HomePlug 1.0 standard has dedicated much attention to the security of HomePlug networks.

The topology of the powerlines also has consequences for the performance of a HomePlug network. Nearby HomePlug networks (in neighbouring houses or apartments for example) that are being provided with electricity by the same transformer, must share the available throughput. In this regard Europeans are put at a disadvantage compared with the Americans, where the number of connections to the same transformer is much lower than in Europe.

The same problem arises when two or more networks are being set up in the same household. The bandwidth will be shared by all HomePlug devices. Other proprietary PLC systems operating (partly) in the same frequency region will also cause the networks to perform poor.

Interference with amateur radio

Because HomePlug uses frequency's that are also used by amateur radio, it must comply with some rules so these two technologies can coexist, without interfering with each other. To do so, the HomePlug standard suggests notching at some frequency's to reduce the power injection at these frequency's.

2.1.3 Solutions

Orthogonal frequency-division multiplexing (OFDM)

OFDM is a technique that is being used in the IEEE 802.11a and 802.11g standards (WiFi), the xDSL technology and terrestrial wireless distribution of television signals. With this technique the frequency spectrum is being divided into several narrowband channels. The division of the spectrum is done by dividing the range of 0Hz to 25Hz into 128 evenly spaced bands. These bands overlap partly to easily translate the signals from the time domain to the frequency domain. This conversion is done by means of a fast fourier transform (FFT).

84 bands lie within the frequency range 4,49 MHz to 20,7 MHz, this is the working range of HomePlug 1.0. Furthermore it is possible to unselect some of these bands to avoid interference with the 40, 30, 20 and 17 meter bands of amateur radio, as explained before. Avoiding subbands is called *tone masking*. This leaves 76 subbands for HomePlug communication.

Each of these 76 subcarrier can now be modulated and forms a smallband channel. By dividing the broad frequency dependent spectrum into several subchannels, a large number of (quasi) frequency independent frequency spectra are created. This way one gets many flat frequency responses and equalisation of the whole spectrum isn't needed anymore.

One OFDM symbol is a combination of all the information conveyed in all the subchannels. An OFDM symbol is generated by applying an inverse FFT (IFFT). The frequency domain points consist of the set of complex symbols that modulate each subcarrier. This way the frequency domain data is converted into a time domain waveform. One OFDM symbol has a fixed duration of 8,4 microseconds (5,12 microseconds of raw OFDM symbol and 3,28 microseconds for the cyclic prefix).

To avoid Inter Symbol Interference (ISI) a cyclic prefix is added in front of the symbol, which is basically a copy of the last few microseconds of the OFDM symbol. This prefix absorbs the ISI that results from the fact that the delay presented by the channel is not constant with frequency [4].

Signal modulation

The subchannels can be modulated using Differential Binary Phase Shift Keying (DBPSK) or Differential Quadrature Phase Shift Keying (DQPSK). Phase Shift Keying (PSK) is a well known technique for communication systems. With DBPSK one raw bit can be conveyed per OFDM symbol, using DQPSK two raw bit can be conveyed per OFDM symbol. Two communicating HomePlug devices can choose the best modulation technique and apply this technique to all the subchannels. DQPSK can send data twice as fast but is more sensitive to noise. When a bad connection exists between the two devices, they can use DBPSK instead to reduce bit errors, avoiding too many frames being retransmitted.

Signal coding

Bits sent over powerlines must be coded with alot of overhead to avoid too many bit errors in this noisy environment. The data is encoded using Reed-Solomon/Convolutional concatenated code. These Forward Error Correction (FEC) methods make communication over the noisy powerlines more effective. Rates of $\frac{1}{2}$ and $\frac{3}{4}$ are available. Over noisy lines a more robust FEC rate wil be used ($\frac{1}{2}$), while in more favorable conditions a rate of $\frac{3}{4}$ wil be used. $\frac{1}{2}$ means that only one data bit per two raw bits is sent. At a FEC rate of $\frac{3}{4}$ three data bits per four raw bits will be sent.

Tone map

By doing a *channel estimation* two communicating HomePlug devices can decide to avoid some subcarriers. If all subcarriers would be used all the time and there is at least one impaired subchannel, then all subcarriers would have to be modulated and coded using very robust modulation and coding schemes. Fortunately, using channel estimation, impaired subchannels can be deselected, allowing the devices to communicate very efficient. The tone map specifies which subcarriers are used and wich are avoided.

Channel estimation is done at least every 30 seconds. A device can request a channel estimation. As a result, the devices gets a response that indicates which of the 84 available tones are used (every tone can be selected or deselected separately). In addition, information about the modulation and coding technique is supplied in the reply. A HomePlug device can only store 15 different *Tone Map Indexes* (TMI) which refer to the used tone map and the used coding and modulation technique. A device can thus only communicate efficiently with 15 other devices. Other devices must communicate using ROBO mode (see later in this section). A HomePlug network with more than 15 devices, and where every devices comunicates with every device would not perform efficient.

Channel adaptation, the whole picture

Using the previously discussed techniques, HomePlug allows devices to send data very efficiently. Because powerline channels tend to vary over time and place, getting the most out of the channel is of utmost importance. The three described techniques are used together to achieve high bitrates. The adaptation has three degrees of freedom:

- using a tone map to deselect impaired subcarriers,
- selection of the modulation scheme (DBPSK or DQPSK) and
- selection of a FEC code rate ($\frac{1}{2}$ or $\frac{3}{4}$).

Four combinations of coding and modulation are used in the HomePlug 1.0 specification. Three of these are: DQPSK $\frac{3}{4}$, DQPSK $\frac{1}{2}$ and DBPSK $\frac{1}{2}$. A fourth coding/modulation pair is used for ROBO mode, which is discussed next.

Using these techniques, 139 different bitrates can be achieved, ranging from 1 Mbps up to 14,1 Mbps.

ROBO mode

The ROBO mode is a very robust scheme, where all subcarriers are used. It also uses DBPSK at a FEC rate of $\frac{1}{2}$, but repeats every bit four times to ensure reception of the frame, even over heavily noise-impaired powerlines. It is the most redundant mode and is most suitable for harsh environments. It is used for initiation (prior knowledge about the used subcarriers, coding and modulation sheme), multicasting and broadcasting frames, and conditions where the powerline channel is so impaired that ROBO mode is the most efficient scheme for communication. Notice that in ROBO mode, only a very slow bitrate can be achieved and won't be optimal in many cases.

2.1.4 Security and privacy

HomePlug networks in nearby housholds can be separated by means of cryptography. This way many different *logical networks* can coexist on the shared medium. A logical network groups all devices that belong together. Typically, users will assign their devices to a logical network to differenciate with devices of a nearby household. Two devices belonging to different logical networks aren't able to communicate, they are separated logically although they could be located in each others vicinity. The encryption is based on the Data Encryption Standard (DES). This is a well known encryption standard that offers a good level of security.

Each HomePlug device has a table of Encryption Key Select (EKS) values that serve as an index for the actual encryption key. Each device that wants to send a frame encrypts the payload data with an encryption key from the table. In the frame control field (a field in the HomePlug MAC frame) the associated EKS is added so each receiving device knows which encryption key must be used to decrypt this frame. Stations that have the same encryption key (called the Network Encryption Key, NEK) associated with the same EKS from a logical network. Stations that don't belong to this logical network can use other NEK's in combination with other EKS's.

The HomePlug standard describes a way to derive a NEK from an ASCII password that is easily remembered. This technique is also used with wireless WEP encryption. It uses MD5 as the underlying hash algorithm.

HomePlug devices are preloaded with EKS 0x01 and NEK 'HomePlug' by default. This allows out-of-the-box communication, but clearly makes all security efforts useless. It is strongly recommended to select another password at installation.

To distribute the EKS/NEK pairs over the HomePlug network, and thus to configure logical networks, a Default Encryption Key (DEK) is used. This is a unique password that is *burned* into all HomePlug devices. The DEK is indicated on all HomePlug devices and ensures secure distribution of the keys over the HomePlug network. To program a

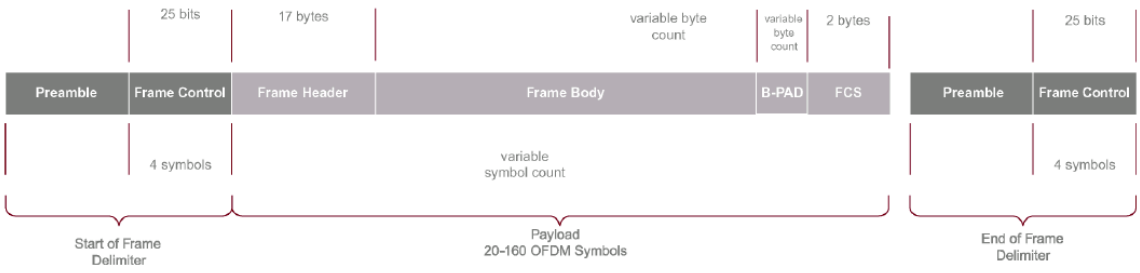


Figure 2.1: Long frame format

HomePlug device so it belongs to a certain logical network, you need to know it's DEK.

There is another *natural* level of security in HomePlug networks. Each link in a HomePlug network is unique, and is characterized by the used modulation technique, FEC rate and tone map (to indicate which subcarriers are used). An intruder trying to monitor a HomePlug conversation will have great problems because the channel from the sender to the receiver and the intruder will be different. The communication will be optimized for the receiver, resulting in a set of usable subcarriers for communication between the two devices, while the intruder will have problems to monitor the symbols on all the used subcarriers, because at least some of those subcarriers will be heavily impaired or attenuated for this intruder.

2.2 Media Access Control (MAC)

The MAC protocol that HomePlug uses is developped keeping HDTV and VoIP data transmission in mind. These applications require low latency and low jitter. A modern protocol can deliver a good Quality of Service (QoS). The used MAC protocol is a variation on the CSMA/CA protocol. CSMA stands for *Carrier Sense Multiple Access*. It means that nodes on the same network access the same medium. Ethernet is another example of a CSMA network, all nodes have physical access to the wires. This implies that collisions can occur when two stations try to send information at the same time.

Two types of HomePlug MAC frames exist: long frames (figure 2.1) and short frames (figure 2.2). Long frames consist of three parts: a Start Of Frame (SOF), the payload data and a End Of Frame (EOF). Both SOF and EOF begin with a *preamble* that is easily

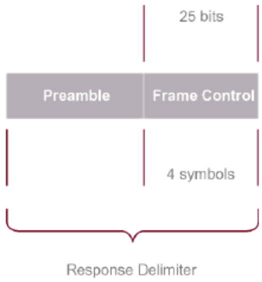


Figure 2.2: Short frame format

detected by all HomePlug devices on the same network. Short frames are being used with *Stop-and-wait Automatic Repeat Request* (ARQ).

2.2.1 Collision avoidance

With HomePlug collisions are avoided (hence the name CA, which stands for collision avoidance). Ethernet uses a slightly different scheme: CSMA/CD (collision detection). Collision detection isn't an option for HomePlug because of the imperfect channel conditions. Collissions can't be detected easily. The only way to detect such a collion is the absence of an ACK frame, a condition that can only be detected after a relatively long period. Collision detection would perform poor for HomePlug. Instead collision avoidance is used.

Collisions can be avoided by keeping timers that tell the station when the medium is busy and when it is available. If a station wants to get access to the medium, it first listens to the preambles that are being send over the powerlines. When a preamble is detected the frame control field that follows is inspected. When it detects an EOF delimiter or a response delimiter the device knows the medium is almost available. If a SOF is detected, the frame control field carries information about the length of the frame being sent. With this information the device can calculate when the medium will be available again.

Every (unicast) frame that is sent by a HomePlug device must be acknowledged by a short frame that indicates the way the frame was received. ACK indicates successful receipt of the frame. NACK indicates the frame contained errors that couldn't be corrected using the redundant information of the FEC codes. FAIL indicates the frame has been received

Figure 2.3: Priority resolution slots and contention period

but the device has insufficient resources to process the frame. Both NACK and FAIL cause the original sender to send the frame again. Multicast frames can also be acknowledged. In this situation only one receiving device sends ARQ frames back to the sending device.

2.2.2 Priorities and QoS

HomePlug Frames can have several priorities. These are the four priorities defined by the HomePlug standard:

- Priority 3: low latency streams (less than 10 msec) like VoIP,
- Priority 2: low latency streams (less than 100 msec) like HDTV,
- Priority 1: bulk transfer like FTP, and
- Priority 0: best effort traffic.

After a frame has been sent, there is a priority resolution interval consisting of two 35,84 microseconds priority resolution slots. In the first priority resolution slot devices with priority 2 and priority 3 frames send out a signal. Devices with lower priority traffic pick up these signals and defer sending their frames. In the second slot the devices with the highest priority traffic send out a signal (either priority 1 if no signal was received in the first slot, or priority 3 if a signal was received). After these slots only the devices with the highest priority traffic may contend for the medium. The priority resolution slots are depicted in figure 2.3.

The four priorities of the HomePlug MAC map to the eight priorities used in the IEEE 802.1Q VLAN standard. This is a standard that uses extra bits in the ethernet frame

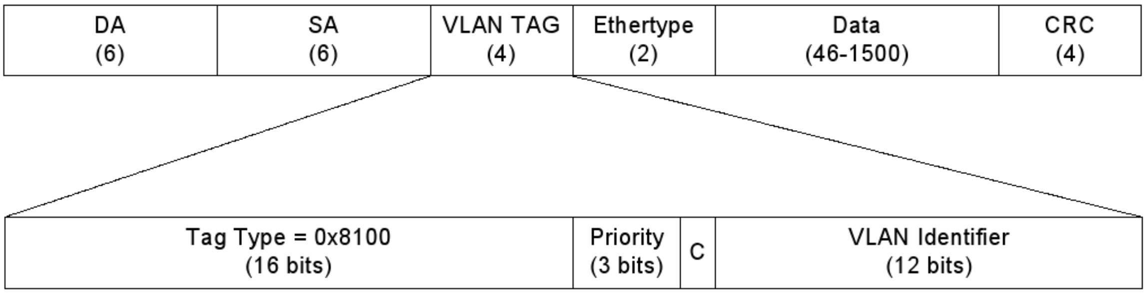


Figure 2.4: VLAN tag structure

header to signal priorities. Using these VLAN tags it is possible to prioritize the frames on the HomePlug network. In figure 2.4 the VLAN tag structure is depicted. The mapping from VLAN tags to HomePlug priority tags is done as follows:

- VLAN priority 7 and 6: HomePlug priority 3,
- VLAN priority 5 and 4: HomePlug priority 2,
- VLAN priority 3 and 2: HomePlug priority 1 and
- VLAN priority 1 and 0: HomePlug priority 0.

We observed that this mapping is only a recommendation. In the datasheet of the INT51X1 chip [2], an Intellon chip for HomePlug 1.0 communication, we saw that this mapping can be changed. To optimize the network efficiency all traffic must be tagged with the propper VLAN tag.

Like ethernet, HomePlug has a segmentation and reassembly mechanism. The maximum size of a HomePlug frame is 160 OFDM symbols. The maximum amount of payload bytes that can be carried in one HomePlug frame depends on the chosen modulation, coding and tone map. This means that no *optimal* frame size can be chosen, as is the case with ethernet. Higher network layers can't determine what the best packet size is, because it isn't a fixed amount of bytes, but changes all the time. With ethernet as underlying technology a *maximum transfer unit* (MTU) of 1500 bytes is chosen to optimize network usage. Because ethernet frames can't contain more than 1500 bytes, IP packets will never get fragmented when travelling over a pure ethernet network. Because the maximum payload a HomePlug frame can carry is variable and changes in time, no such optimization is possible with a HomePlug network.

When a complete ethernet frame, that can be up to 1518 bytes long (header included), doesn't fit into one HomePlug frame, fragmentation occurs. Like fragmentation on IP level in the OSI stack, several bytes in the HomePlug frame header indicate the frames are segments, and in which order they should be reassembled. Segmented HomePlug frames belonging to the same ethernet frame can be sent in one burst. This means that all the segments are sent one after the other, without going through the trouble of contending for the medium. This greatly improves performance. On the other hand, all segments still have to indicate their priority. This means that traffic with greater priority can still interrupt this burst of segments, which improves the QoS of the whole network. Traffic with the highest priority can only burst two segments at the time. This ensures low jitter for priority 3 traffic.

2.2.3 HomePlug MAC frame structure

As stated before, two types of HomePlug frames exist: long frames and short frames. Long frames consist of a Start Of Frame delimiter (SOF), the payload and an End Of Frame delimiter (EOF). They encapsulate the payload data. Short frames are used in the ARQ process and only consist of a response delimiter.

All delimiters begin with a preamble that is followed by a frame control information field. The preamble is a very robust signal that can be detected, even in very noisy conditions. The frame control information field is coded in a very robust way, using Turbo Product Coding. The frame control information field can be detected, even if the signal is below the noise-floor. The frame control information fields carries information about the type of frame that is being sent, the frame length (so other devices that listen can calculate when the medium will be free for contention), a tone map index and the priority of the frame being sent (only in an EOF delimiter). In case of a response (short) frame the type of response is also encoded in the frame control information field (either ACK, NACK or FAIL).

The frame header, which are the 17 first bytes of the payload in a long frame, contain information about the sender, receiver and segmentation. A Frame Check Sequence (FCS) is appended to the payload to detect bit-errors. A wrong FCS results in a NACK response frame from the receiver, so the sender can retransmit the corrupted frame. A B-PAD field is added to ensure the payload fits into a multiple of OFDM symbols.

Most information about the HomePlug 1.0 technology was found in [4], [5] and [6]. More information can be found in these articles, or in the original HomePlug 1.0 standard specifications.

2.3 Other HomePlug standards

The technology that was described earlier in this chapter applies to the HomePlug 1.0 standard. This is a standard that is used in commercial devices that are on the market today. In this section we will discuss two other technologies. These technologies are evolutions of the HomePlug 1.0 standard to allow a higher throughput and a better QoS. We will focus on the differences with the HomePlug 1.0 standard and how improvements were made. First 85 Mbps devices, HomePlug 1.0 devices with a turbo mode, are discussed. Later the HomePlug AV standard is discussed.

2.3.1 HomePlug 1.0 Turbo

HomePlug 1.0 devices with a turbo extension can handle speeds up to 85 Mbps. These are the devices we used in our tests. It is an improvent of the HomePlug 1.0 standard, but isn't a standard defined by the HomePlug Alliance. We could use the *dLan Ethernet Highspeed 85* adapters of Devolo and the *PowerLan 200 Turbo* adapters of Topcom. Both adapters are based on the Intellon *INT5500* chipset.

The throughput improvement is a result from the improved modulation techniques that are used. It uses 256 QAM for the fastest communication. Using this modulation, one symbol can code eight raw information bits. HomePlug 1.0 with turbo uses the same spectrum range as does the original HomePlug 1.0 standard: from 4 Mhz to 21 Mhz.

These devices are backwards compatible with the HomePlug 1.0 standard, which is important if you have other HomePlug 1.0 adapters that have to communicate with these adapters. Communication between a HomePlug 1.0 device and a HomePlug 1.0 device with turbo extension has a maximal throughput of 14 Mbps.

2.3.2 HomePlug AV

HomePlug AV supports speeds up to 200 Mbps at the physical layer and 150 Mbps at the MAC layer [7]. It's a standard defined by the HomePlug Alliance. Some products that comply with this standard are on the market today. It addresses QoS issues to support streaming media better than the HomePlug 1.0 standard. It also has a better security management.

HomePlug AV uses a broader spectrum, from 2 to 28 MHz, instead of the 4 to 21 MHz spectrum HomePlug 1.0 uses. The underlying technology remains OFDM, but the subcarriers are placed much closer to each other. The spectrum is now divided into 917 usable subcarriers, instead of 84 that are used with HomePlug 1.0. This means that the spectrum can be used much more efficient since the tone map is more precise. Unusable frequencies can be notched out more accurately.

Modulation of the subcarriers is also altered in the AV specification. Modulation densities vary from BFSK (one bit per subcarrier per OFDM symbol) to 1024 QAM (10 bits per subcarrier per OFDM symbol). Moreover, the modulation technique can be specified per subcarrier. Again, more efficiency can be obtained.

A major improvement has been made on the MAC layer. Media access now can be time based or contention based. Time based media access (also called TDMA, Time Division Multiple Access) improves QoS because it can assure stations can access the media for a specific period of time. A station can thus obtain a certain bandwidth. This is different with CSMA as used with HomePlug 1.0 because all devices with traffic of the highest priority contend for the medium altogether. HomePlug AV works with beacon periods. In such a period there is a CSMA period where the medium can be used in a way that was also used with HomePlug 1.0. Another period is reserved for TDMA access. The period stations get access to the medium is organised by the Central Coordinator (CCo). Every station has to request for a piece of the total bandwidth. If the CCo approves this request the station has contention free access to the medium during a certain period, every beacon period. Stations still can obtain access to the medium through contention, but only in the CSMA portion of the beacon period.

All stations in the same HomePlug AV network are synchronized with each other. This synchronization is based on the zero-crossing of the AC on the powerlines. This implies that HomePlug AV devices must operate on AC powerlines. This is not strictly necessary with HomePlug 1.0 because no clocks have to be synchonized. HomePlug 1.0 networks could be build on DC powerlines or dead powerlines without any voltage applied to it. Note that the device itself would still need power, power that now is taken from the same powerlines as where the network is build upon.

This synchronized clock is useful for devices that need this timing, like surround sound speakers. Such speakers have tight timing constraints. These devices can access timing information. This timing information can also be used to *smooth* (de-jitter) a stream of frames. This can be very useful for VoIP applications where frames have to arrive at a constant rate. De-jittering is then done by the network instead of using a de-jitter buffer in the application.

Security is enhanced in this standard. Instead of 56-bit DES encryption, 128-bit AES encryption is used. The way to enable encryption is similar as described in section 2.1.4.

HomePlug AV enables *coexistence* and *interoperability* with HomePlug 1.0 devices. Interoperability is an option, but coexistence is required by the standard. Coexistence means that in a mixed environment of AV and 1.0 adapters both devices will be able to send data at an acceptable rate. Interopability means that AV devices can communicate with 1.0 devices. HomePlug AV devices on the market today (like the AVdesk from Devolo) only have the ability to coexist with HomePlug 1.0 devices, interoperability is not possible. On the devolo website a bridge is suggested to interconnect 1.0 and AV networks. The bridge consists of one HomePlug 1.0 device and one HomePlug AV device, both connected to the powerlines and connected with each other by means of a simple ethernet cable. However, this is a very expensive solution.

3 HomePlug getting physical

With our knowledge of the HomePlug technology from the previous chapter we can now focus on how our signals will propagate through the domestic electricity grid and the problems it may encounter while traversing it. We will also investigate how HomePlug is suited for its use in a company environment instead of a stable home environment. As the title of this chapter already implies, this is where we dive into the physical layer of the network stack so no attention will be paid to applications or transport protocols used.

3.1 The electricity grid

The distribution of electricity to our homes and offices is often taken for granted. Normally there is no shame in doing so but for this thesis it becomes rather important. So it may prove useful if the basics and the topology of electricity distribution are understood.

3.1.1 Power Plant

The start of the electricity's journey is at the power plant where water, coal, natural gas or nuclear fuel is used to produce steam. This steam is then used to make a turbine spin which in turn generates electricity. Due to the long distances this electricity must travel, step-up transformers are used in a power plant to increase the voltage and facilitate the flow of electricity over long distance high-voltage lines with a minimal loss.

3.1.2 Substation

When we reach a group of customers (e.g. city), the high-voltage line is coupled to a transformer in a facility referred to as a substation. This substation reduces the voltage via a step-down transformer and places it on medium-voltage lines. These lines can run overhead or underground.

3.1.3 Low-voltage lines

As the power lines reach an area where homes or offices are in close proximity, the voltage on the medium-voltage lines is further reduced by smaller transformers. These transformers are either mounted on utility poles or set as rectangular metal enclosures on concrete pads. Either type of transformer reduces the voltage to 220 volts which will run to the customers on low-voltage lines. From these transformers a cluster of customers is served.

3.1.4 Service Panel

The last step of our journey takes place at the homes and offices that are being served. As already explained, two types of services exist: overhead service and underground service. Whatever service is used, the power line will be routed to an electrical meter which measures how much electricity is used to make up the monthly bill. From this meter, wires are routed into a service panel, which distributes the electricity throughout the home or office. Older service panels consist of fuses that must be physically replaced when they blow. Nowadays the service panel is a breaker panel with breakers that trip when to much current is flowing. Power remains off until the breaker is physically reset.

3.2 Classification and quantification of noise

In section 2.1.2, we already saw that there's lots of noise present on powerlines coming from various sources. Although solutions were proposed in that section, a noisy medium is always a problem and will affect higher-layer throughput and subsequently the QoS. Therefore this section will take a closer look at which typical and commonly used household devices will pose a potential problem when using powerlines as a communication medium. And if they do form a problem, we will try to quantify the damage. There is also random noise generated from atmospheric conditions, nearby power lines interference, discharge at locations where there are faulty connectors or dirty insulators, but these noise sources are less important in comparison to regular household appliances.

3.2.1 Iperf

Iperf is a tool that is being used in many tests we performed. It's a tool that can be used to measure many parameters of a network connection. In our tests we used version *2.0.2 (03 May 2005)*. Iperf is a command-line tool and can be started using several options. To test a network connection between two computers, two instances of the tool have to be started. One computer has to start the iperf tool with the *-s* option while the other computer has to start it with the *-c* option. The *-s* option indicates that iperf has to listen for an incoming connection from an iperf client. The *-c* option indicates that iperf has to run as a client that connects to a remote iperf server. The minimal options to test a network connection are:

```
iperf -s (server)
iperf -c 192.168.0.1 (client)
```

In this case the computer running the server must have 192.168.0.1 as IP address. By default a TCP connection is set up between server and client. The option *-u* instructs server and client to initiate a UDP connection.

The tool tests the throughput by sending as many packets as possible over a network connection. The packets flow from client to server. By sending packets during a specified amount of time (default 10 seconds) both client and server can calculate how much throughput was available throughout this period by dividing the amount of data sent by the time it took to send this data.

When iperf is used to send UDP data the server can report *jitter* and *packet loss*. Other options can instruct iperf to measure throughput in both directions (*-r*) and to use another TCP window size (*-w 512k*). Other options of the iperf tool will be explained when needed.

3.2.2 Test description

We created a point-to-point HomePlug network with two Turbo adapters. The goal is to connect the household devices to the same electrical network and measure their effect on the TCP throughput between the two workstations. We first measure the throughput (both up- and downlink) with the iperf tool without any device attached in the vicinity of the network. This way we can compare the results easily with a 'normal and stable' home situation (no devices attached). One must notice that although no devices are present in the network, the noise level will vary in time. As a consequence we must keep this inaccuracy in mind. We measured the TCP throughput with the following devices attached: a laptop power supply, an IP router, a USB hard-disk, a mobile phone charger, a HiFi installation, a mixer, hedge shears, a drill, an electrical heater, a hairdryer, a vacuum cleaner, a computer, a desk lamp and a guitar amplifier. The throughput is measured in two directions with iperf. We use the command *iperf -s* and *iperf -c <ip> -f m -i 5 -t 20 -r*. Note that we measure the TCP throughput and not the UDP throughput.

3.2.3 Test setup

Two laptops are used to measure the throughput. The laptops are working on battery power so they are not plugged into the net. The two laptops make a point-to-point network connection through two Devolo Ethernet-to-Powerline bridges (dLAN Ethernet Highspeed 85). Both laptops run Windows XP and measure throughput with the iperf tool, as stated before. A powerstrip is used to connect the first laptop to the powerlines. A second outlet in this powerstrip is used to connect a second powerstrip. At this second powerstrip the second laptop is connected to the powerlines. Both powerstrips have electrical wires of approximately two meters. The two powerline adapters are thus separated by aproximately two meters of electrical wiring. The devices creating the noise and changing the line-impedance are connected to the first powerstrip. Upstream, in this context, means a stream from laptop 1 to laptop 2, downstream from laptop 2 to laptop 1. The results are presented below.

Table 3.1: Effect of various noise sources on TCP throughput

	up (Mbps)	down (Mbps)
No device	29,5	17,8
Mixer (300W) off	28,7	17,7
Mixer (300W) on	27,4	16,3
Laptop powersupply	29,1	15,2
Router + adapter (7W)	29,3	17,9
External HD	27,9	16,2
Moblile phone charger	20,7	13,3
HiFi installation (standby)	28,5	17,2
HiFi installation (on)	27,7	17,3
Hedge shears (600W)	28,1	17,1
Drill (550W)	24,1	11,6
Electrical heater (2000W)	15,8	12,4
Hairdryer	28,3	17,2
Vacuum cleaner	29,2	17,5
Computer (250W) (on)	27	16,7
Computer (250W) (standby)	27,1	17,4
Guitar amplifier (480W)	28,1	17,4
Desk lamp (11W)	29,6	17,7

3.2.4 Conclusion

As we can see in table 3.1 and also in figure 3.1, the HomePlug network presents a highly asymmetrical throughput, even with this simple topology. This behaviour will be explained later in section 4.2. Three devices degrade the performance notably: the mobile phone charger, the drill and the electrical heater. The other devices only create a minor change in throughput. The three devices that impair the network performance both change up- and downstream throughput. We know from section 2.3.2 that HomePlug AV adapters operate in approximately the same frequency region as the Turbo adapters. So we could expect that also AV adapters will suffer from these devices although no such tests were performed.

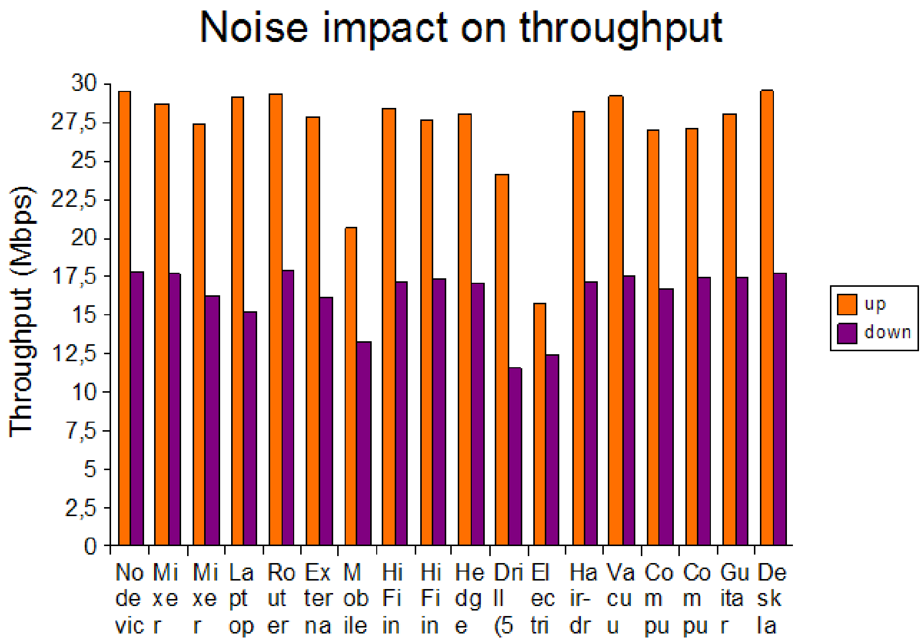


Figure 3.1: Noise impact on throughput

3.3 The phase issue

As we discussed the electricity grid in section 3.1 of this chapter, we saw that electrical energy is produced at a power plant by a turbine. The turbine has a rotating shaft with a magnet attached to it. As this magnet rotates within a stationary conducting ring, it generates electricity. It is however relatively simple and cost effective to include three seperate coils on a single shaft instead of a single coil. These three coils produce three signals equal in magnitude but 120° out of phase. Most home users do not have three phases at all and if they do they only need it for large appliances like a cooking range, an electric boiler or an electric motor. It is then split out at the main distribution board.

Now the question arises: will this phase issue be a problem with the HomePlug network? Will HomePlug signals traverse to lines on different phases? We created an easy HomePlug network with only two adapters with each one on a voltage line with a different phase and we measured the TCP/UDP throughput (if possible) and compared it with the throughput if both outlets are on the same phase (other outlet is located in same room so the distance can be considered equal).

The throughput when on different phases was nearly the same when on the same phase! So three phase systems/lines do not hinder the HomePlug traffic in a significant way. This cross-phase barrier is broken because of the high frequency spectrum of the HomePlug signals. Powerline technologies before the HomePlug 1.0 standard, which operated at much lower frequencies, were hindered greatly due to the signal isolation that occurred between different phases. But thanks to the high frequency range of current standards, three-phase sytems do not pose a problem [8].

3.4 Neighbours: possible intruders?

Privacy of the communication link is obviously very important and becomes even more important when the medium is not limited to your own house or office as is the case with HomePlug. Three crucial remarks regarding privacy: Firstly, in 2.1.2 we discussed the privacy problem. Many efforts have been made to make HomePlug as safe as possible: DES-encryption, unique link between sender and receiver because of the tone map and let us not forget the limited range the signals can travel. The most important reason this range is rather short is because of the very noisy communication medium. If a longer range is needed, one must boost the signal too hard to overcome the signal attenuation to stay above the noise level.

Secondly, as also explained in that section, a step-down distribution transformer which converts medium voltage to low voltage provides a cluster of customers with electricity. In Europe the number of served customers by one distribution transformer is bigger than in the United States. This has the annoying consequence that in theory different households can 'see each other's traffic and that every customer on the same distribution transformer must share the available bandwith.

Thirdly, a network encryption key (NEK) is used to set up a logical network. With these remarks kept in mind one can already feel that curious neighbours don't stand much of a chance of intruding someone's privacy [3].

3.5 Isolation of HomePlug signals

Suppose a company would like to give a demonstration using the HomePlug technology on an exhibition. It would be nice if that company could isolate their demo traffic from the other companies so that their business is assured to stay their business. On the other hand it may be useful if that same company could isolate itself from noise and other possible HomePlug traffic present on that exhibition that may interfere with their demonstration. This bidirectional filtering using HomePlug Turbo adapters will be addressed in this paragraph.

3.5.1 Noise filtering

Let us first delve into the noise filtering topic. In this chapter we have already seen that various devices cause a relapse in throughput. So the goal of this subsection is to stop noise from entering our isolated HomePlug network. Various netfilters (e.g. Figure 3.2) exist to suppress high frequency noise (differential as well as common mode noise). Basicly a netfilter's task is twofold: it can be used to filter high frequency noise in order to protect hifi-systems but can also protect the electricity grid from noise coming from such hifi-systems. That is why we put a Schaffner netfilter to the test. This netfilter has a choke coil to act as a load to suppress high frequency noise, two X grade capacitors to ensure that the common mode currents on both hot wires are equal and two Y grade capacitors to ensure a good and safe path for the common mode noise to the safety ground. The noise source used for this test was the mobile phone charger from section 3.2 where we saw it had a serious impact on HomePlug's throughput. The objective is to see whether the netfilter is capable of withholding the noise generated by this charger.
After measuring the throughput between 2 PC's with the iperf tool one time with and one time without the filter, it shows that the filter had no affect whatsoever. So maybe this is because the noise of the charger isn't in the stopband frequency region. Therefore netfilters are no reliable noise filters when working with HomePlug.

Figure 3.2: Example of a Schaffner netfilter

3.5.2 HomePlug filtering

Now our focus will be on the filtering of HomePlug signals itself.
Powerstrips with transient voltage suppressors are known to be good suppressors of HomePlug signals. But can we turn this disadvantage into an advantage for isolation purposes? We tested this with a Premium-Line (BF30000 EMI/RFI filter from Brennenstuhl) powerstrip. Again a little HomePlug network with two WindowsXP laptops and iperf as our tool was set up. One (the client) is one time plugged in in the powerstrip (so with the filter) and one time in an ordinary home outlet (without the filter). The other one (the server) is on a fixed location on an ordinary home outlet. The following results (table 3.2) are obtained from iperf after invoking the *iperf -s* command on the server and the *iperf -c <ip> -f m -i 5 -t 20 -r* command on the client machine.

The powerstrip does block our signals but not entirely as the results show us. Especially the upstream suffers from a drawback in throughput. The connection could still be established although the speed as such is not usable for most applications. As a conclusion one may say that such powerstrips are rather good isolaters of the HomePlug signals but only for one-way traffic. So it seems that powerstrips in households do more harm to the communication than providing a good full-duplex shield for private traffic.

Table 3.2: Powerstrip as HomePlug filter

	up – client to server (Mbps)	down – server to client (Mbps)
TCP throughput with filter	0,53	2,00
TCP throughput without filter	9,38	2,21

The Schaffner netfilter from section 3.5.1 was found to be unsuccessful for noise filtering purposes but what about filtering HomePlug signals? When using the netfilter for this purpose, it showed a remarkable decrease in throughput of 50% but this filter cannot block the signals entirely.

It is unassailable that filtering noise when dealing with HomePlug or filtering the HomePlug signals itself is no easy task. Although we managed to obtain remarkable decreases in throughput, the connection could still be made (and noise still found his way through). So blocking the signal (or noise) completely has not succeeded.

3.6 HomePlug in a company

Our thesis objective is to evaluate the HomePlug technology in a typical home environment which is characterized by the words: stable, low noise level, short distance links. This is certainly true if we compare the medium in a home with the one in a company.

To investigate HomePlug in a company environment, a few tests were done in the GroupT Engineering School building in Leuven. This building is divided into several modules, fourtheen to be exact, with module 1 being the lowest and module 14 being the highest module. Each module has it's own breaker panel to distribute electricity to the module on which the panel is located. Every panel is connected to the central service panel located in the basement of the building. So if one HomePlug adapter wants to communicate with an other one that is not on the same module, the traffic must pass the central service panel. This causes the point-to-point distance to increase drastically! This increase can not be ignored in this case as the turbo HomePlug adapters ensured a working distance of approximately 200 meters.

Table 3.3: HomePlug measurements in GroepT

	RTT (ms)	up (Mbps)	down (Mbps)
Module 10 (same phase)	2,272	30,90	33,10
Module 10 (different phase)	2,237	28,40	29,70
Module 14 (same phase)	2,214	29,20	27,50
Module 14 (different phase)	2,329	27,70	24,60
Module 9 – 10	3,686	5,52	5,02
Module 12 – 14	5,143	3,28	3,22
Module 13 – 14	4,651	1,96	3,17

Table 3.3 shows the results of the tests. A first conclusion is that the link is encouraging when both adapters are plugged in on the same module but the throughput decreases heavily - as expected from previous remark- when used between different modules.
Secondly there is a slight decrease in throughput when the adapters are using a different phase but this is negligible.

These conclusions tell us that a company environment can not be covered entirely with HomePlug technology as a reliable backbone of the network. The noise level is too unstable and the distance between possible adapters is often too large to ensure a good QoS. It can however provide a good alternative to existing technologies when used in smaller specific areas like the modules in this building.

4 HomePlug in home-networks

In previous chapters we discussed how the HomePlug network works at the bottom layers of the OSI network stack. These layers are very specific to HomePlug and are specified by the HomePlug Alliance.

Although HomePlug *stops* at the second layer, it is still interesting how a such a network performs, as seen from the layers above. In this chapter we present the results from tests that were done to get an idea of how a HomePlug network performs on the network layer. In chapter 5 we will look at the impact on applications of an underlying HomePlug network.

4.1 Varying throughput at each layer

When we first measured the throughput of a HomePlug network we were very disappointed. Although the box mentioned a throughput up to 85 Mbps (we used HomePlug 1.0 adapters with a *turbo extension*), we only measured a throughput of maximal 30 Mbps when the adapters were locate right next to each other! This is only 35% of the promised throughput. When both adapters were located in different rooms, speed dropped to about 15 Mbps which is only 17% of the advertised throughput.

Although the HomePlug 1.0 standard promises a throughput of 14 Mbps, we only can use a fraction at the higher levels of the OSI stack. The *14.000.000 bits per second* that theoretically can be sent, are bits at the lowest layer, the physical layer. These bits include all the overhead bits that are needed at the physical layer. Only 8,2 Mbps (8.200.000 bits per second) can be used by the MAC layer, in an ideal situation. This means that about 41% of the bits sent over the HomePlug network are meant for management at the lowest layer. This huge loss is mainly caused by the fact that powerline communication requires much redundancy to avoid a high *Bit Error Rate* (BER) that otherwise would become a problem due to the noise on the powerlines.

The differences between physical and MAC layer bitrates are shown in the following table:

Table 4.1: Differences between PHY and MAC throughput

	HomePlug 1.0	10 Mbps Ethernet	IEEE 802.11b
Physical Layer (Mbps)	14	10	11
MAC Layer (Mbps)	8,2	9,8	7,48

The discrepancy between physical and MAC layer is much less with ethernet because no such redundancy is needed to communicate reliably over this medium. For the wireless technology IEEE 802.11b (WiFi) the difference is again bigger because noise is also an issue for this technology. Thus more redundancy and management is needed.

We must remember, especially with HomePlug, that only a fraction of the advertised throughput is available for the highest layers. The same is true with the newer HomePlug AV standard. This standard promises a throughput of 200 Mbps, but at the network layer, no more than 100 Mbps is to be expected. Tests in the following section will show us how much throughput is left at the network layer in a typical home environment.

4.2 Measuring QoS parameters in a home situation

To get more feeling with the HomePlug technology, and to get to know the capabilities and limits, we performed several tests. All these tests were performed under non-ideal conditions, since we are more interested in how the HomePlug technology performs in these circumstances, rather than in an ideal situation.

A first and very interesting test was a measurement of some QoS parameters throughout a typical home. The test was inspired by an article in c't Magazine [9]. In this article they measured the performance of some HomePlug (and WiFi) adapters throughout a house. In this article only the throuhput was measured, while we are also interested in other QoS parameters. We measured throughput, round-trip-time and jitter.

4.2.1 Description

The test was conducted in a typical, fairly new house with modern electrical wiring. The house and the electrical wiring is about 10 years old. All electrical oulets are connected to the same phase. Although we can't possibly know exactly how the electrical wiring is applied, and what the distances are between outlets, we have a rough idea since we have access to a detailed plan of the wiring. Outlets wired to the same circuit are indicated, and distances can be estimated based on this plan.

We could have chosen to measure all the paths (pairs of two outlets) in the home, but with more than fourty electrical outlets, this would require more than 1.600 measurements, in both ways. Instead we chose a central point in the home at the second floor (indicated with the logo of a computer on figure 4.2 and 4.3). At the second floor we measured almost all the other outlets. At the other floors (street level and basement), we measured at some random, and well scattered outlets. The pc's were connected to the HomePlug network using the Devolo dLan Higshpeed 85 adapters (adapters with the turbo extension).The results from this test can give us an idea how the throughput, round-trip-time (RTT) and jitter varies throughout the house.

4.2.2 Tools used

Measuring throughput with iperf

To measure the available throughput we used *iperf.* At the server side (the fixed location) we ran it with the following options:

```
iperf -s -w128k -i 5 -f m
```

The *-w128k* option tells iperf to maintain a TCP window of 128 KByte. We do this explicitly because we are running the server and the client on different operating systems. By defining the TCP window size explicitly, we avoid having two different window sizes at the client and the server, and we get better results (especially when measuring in both directions).

The *-i 5* options makes iperf report to the screen every 5 seconds, which is just convenient for us, to follow the results. The *-f m* option tells iperf to output the measurements in the *Mbps* format. Notice that 1 Mbps, here, equals 1.000.000 bits per second, and not 1.024^2 as one could expect.

At the client, we had to make iperf run and connect to the constantly running iperf-server. Iperf was started with this command:

```
iperf -c 10.0.0.1 -r -w128k -t 30 -i 5 -f m
```

The three options *-w128k*, *-i 5* and *-f m* have the same meaning as described before. The *-c* option, followed by the IP address of the listening server, makes the client connect to the server to perform the test. The *-r* option makes iperf do the test in both directions. This way, we get measurements from the maximal bandwith from server to client, and from client to server. The *-t 30* option makes iperf run each test (both up- and downstream) for a period of 30 seconds.

Measuring jitter with iperf

To measure the jitter, we also used iperf. When iperf measures the throughput of a UDP connection, it reports the amount of jitter (expressed in ms). Server and client were started with these commands respectively:

```
iperf -s -u -i 5 -f m
iperf -c 10.0.0.1 -u -r -t 30 -i 5 -b 10K -f m
```

The *-u* option makes iperf send UDP packets instead of TCP packets. This is the only way that iperf can report jitter. The *-b 10K* option makes the client send UDP packets at a speed of 10 *KBytes* per second. We chose this bandwidth in the vicinity of a conventional VoIP telephone call, since jitter is of major importance for this application. Later we noticed that a typical VoIP telephone call requires about 200 kbps, which is more than the used 10 KBytes per second (= 80 kbps). But we still think the measurements are useful.

Measuring round-trip-time with ping

We measured the *round-trip-time* (RTT) to get an idea of the actual delay frames, and thus packets, experience when travelling from server to client. Since an easy and ready-made solution to measure the delay doesn't exist, we decided to measure the RTT instead. The popular and easy to use tool *ping* gives the necessary information. The implementation of ping under Windows can't measure RTT's accurately if they fall under 1 ms. Luckily, the implementation of ping under Linux can. The RTT was measured at every location of the client. The server invoked the command:

```
ping 10.0.0.2 -c 10
```

This makes ping run ten times in a row (one ping packet every second). After ten seconds, a summary is displayed. An average, a minimum, a maximum and a mean deviation is calculated, based upon the information of every ping packet.

4.2.3 Results

All the measurements are shown in table 4.2. Outlets indicated with the same letter are connected to the same circuit. Note that *C1* and *C1'*, for example, refer to outlets on two different circuits. A summary of all the parameters are shown in table 4.3, 4.4 and 4.5.

4.2.4 Conclusion

We see that the whole house can be covered by HomePlug. The adapter at the fixed position at the second floor, can communicate with an adapter located in the basement. Throughputs vary from 7,9 Mbps to 32,7 Mbps. The lowest speed was measured in the basement, which was expected, because this is a location located far from the fixed adapter. The highest throughput was measured in the hallway. This outlet is located at the other side of the wall of the other outlet, where the first adapter is connected. Physically, the adapters are separated by only a couple of centimeters, which explains the excellent througput in this situation.

Table 4.2: Results from the QoS tests

	Throughput		Round-trip-time			Jitter	
outlet	up (Mbps)	down (Mbps)	min (ms)	avg (ms)	max (ms)	up (ms)	down (ms)
Room 1							
E4	28,5	22,7	2,172	2,550	5,312	0,002	0,331
D1	25,7	23,5	2,216	2,701	6,118	0,007	0,047
Room 2							
A3	27,3	24,9	2,177	2,546	5,315	0,001	0,042
B1	29,4	29,2	2,179	2,576	5,398	0,003	0,131
E7	29,9	28,8	2,128	2,783	7,280	0,001	0,086
Room 3							
A1	27,4	17,9	2,210	2,598	5,336	0,001	0,060
A2	24,6	19,6	2,062	2,532	5,319	0,005	0,102
A4	26,4	25,1	2,172	2,949	6,575	0,001	0,048
C1	22,0	20,8	2,127	2,704	5,320	0,002	0,103
Bathroom							
Y2	25,4	20,9	2,185	2,655	5,247	0,001	0,071
Y3	29,2	26,5	2,168	2,568	5,388	0,008	0,094
Shower							
C1'	15,5	12,4	2,240	2,878	5,378	3,686	0,371
C2'	18,6	19,3	2,174	3,172	5,796	0,001	0,058
Room 4							
D2'	22,0	19,6	2,167	2,540	5,321	0,002	0,280
Room 5							
E2'	21,2	15,2	2,061	2,529	5,387	0,002	0,040
O2	15,0	15,4	2,235	3,602	8,186	0,003	0,449
Room 6							
E7'	30,6	32,6	2,173	2,567	5,315	0,002	0,044
Hallway							
E1	24,8	32,7	2,183	2,839	7,355	0,002	0,057
Dining Room							
Q7	12,7	13,6	2,252	10,150	38,540	1,779	0,655
S1	16,9	16,6	2,206	2,840	5,690	0,003	0,335
U4	16,1	17,6	2,202	2,707	5,317	2,775	0,077
Television							
H1	26,2	21,2	2,163	2,929	7,228	0,001	0,221
Q2	11,6	16,5	2,280	5,998	9,944	7,402	0,843
Main Hall							
Q1	19,7	18,3	2,725	3,719	6,500	0,002	0,573
Garage							
M3	18,2	14,9	2,173	3,524	13,497	0,001	2,433
Storage							
A1'	19,1	16,5	2,228	2,673	5,424	0,001	0,223
S8	19,0	13,3	2,242	3,118	5,324	0,003	0,640
Kitchen							
U6	16,7	12,4	2,206	3,054	9,618	0,314	0,046
Basement							
M2	19,8	16,4	2,177	2,660	5,388	1,893	0,080
S3	15,0	13,5	2,204	2,651	5,374	1,896	0,421
S4	11,1	12,0	2,180	2,710	6,015	1,096	0,369
U1	11,9	7,9	2,238	2,711	5,829	0,002	0,036

Table 4.3: Summary of the measured throughputs

	up (Mbps)	down (Mbps)
Average	21,2	19,3
Max	30,6	32,7
Min	11,1	7,9

Table 4.4: Summary of the measured round-trip-times

	min (ms)	avg (ms)	max (ms)
Average	2,203	3,148	7,345
Max	2,725	10,150	38,540
Min	2,061	2,529	5,247

Table 4.5: Summary of the measured jitter

	up (ms)	down (ms)
Average	0,653	0,293
Max	7,402	2,433
Min	0,001	0,036

In general, throughputs at the second floor are higher than speeds at the other floors. The average speed at the second floor is 23,6 Mbps, while the average speed to the first floor is about 16,9 Mbps. Outlets located in the basement only allow an average speed of 13,5 Mbps. It is obvious that the outlets located at the second floor, the same floor as the fixed HomePlug adapter, can reach higher speeds due to the minimal distance between the two outlets. Outlets at lower floors don't allow such fast throughput because of the increasing distance between the two HomePlug devices. Going from the second floor to the floor below results in a throughput loss of about 6,8 Mbps on average. Two floors lower, the throughput decreases an additional 3,4 Mbps. Thus, going from the second floor to the basement results in a throughput loss of 10,2 Mbps.

The overall average may be somewhat misleading, because we measured throughputs at almost all outlets at the second floor, and only measured some outlets at the lower floors. Measurements at the second floor will influence the average more than the measurements of the first floor and basement. Hence, the overall average will be too high.

We learn that it is more efficient to use the HomePlug adapters on the same floor. But it is still possible to use them to network the whole house.

In a first graph (figure 4.1) we calculate how much percent of all measured throughputs allows a minimal throughput. Because 7,9 Mbps was the lowest throughput we measured, all outlets allow a minimal throughput of 7,9 Mbps. Only one outlet allowed a throughput of 32,7 Mbps. This outlet represents 1,56% of all outlets. Both upstream and downstream measurements were used separately to construct the graph.

On the graph we indicated the 80% and 90% thresholds because they are important for consumers who want to know how much throughput is available in most of the cases. Using the graph, one can see that 90% of all outlet-pairs allow speeds of minimum 12,5 Mbps. If we lower the threshold we see that about 15 Mbps is reached in 80% of all cases. These numbers are meaningful, because they predict better the typical speed that can be obtained in a typical house. We can say that there is about 90% chance that HomePlug devices connected to two random outlets will achieve a throughput of minimum 12,5 Mbps. This is a better way to describe speeds that can be obtained with this technology. From a economical point of view, it is better to advertise the maximal speeds, but these speeds will never be obtained in a real-life situation.

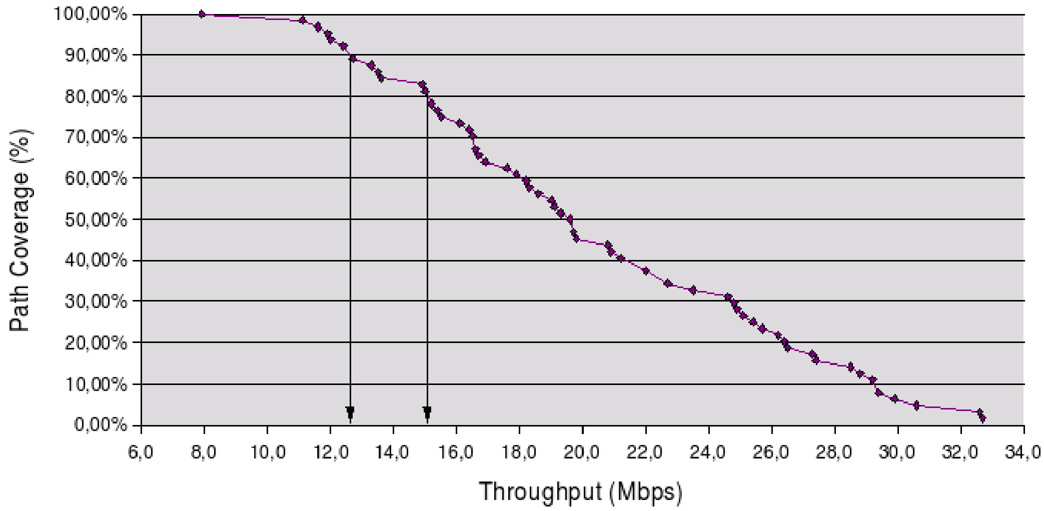
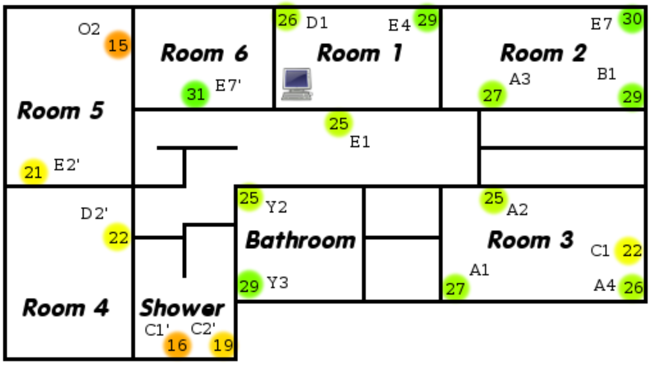


Figure 4.1: Coverage plot

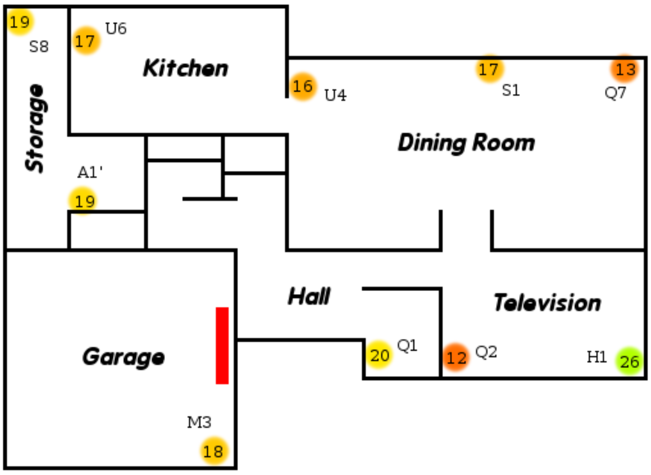
A visual representation of the measured throughputs is useful to see how the throughput decreases in function of the rising distance between the two outlets. In figures 4.2 and 4.3 such a visual representation is given. The legend used to color the dots according to their throughput is given in figure 4.4. To construct these figures we used the measurements presented earlier. Over an interval from 5 to 35 Mbps a gradient was drawn. Every bitrate between 5 and 35 Mbps now corresponds to a color on this gradient. Dots that are green represent outlets that allow a good throughput to the server, while red dots represent outlets which don't allow very high speeds. Both upstream and downstream throughputs are drawn separately.

Again, we clearly see that outlets located in the vicinity of the other outlet allow much higher throughputs. At ground level only one outlet allows a throughput of more than 20 Mbps in both directions, which makes it a good candidate to connect other HomePlug devices to that have to communicate with HomePlug devices at the second floor.

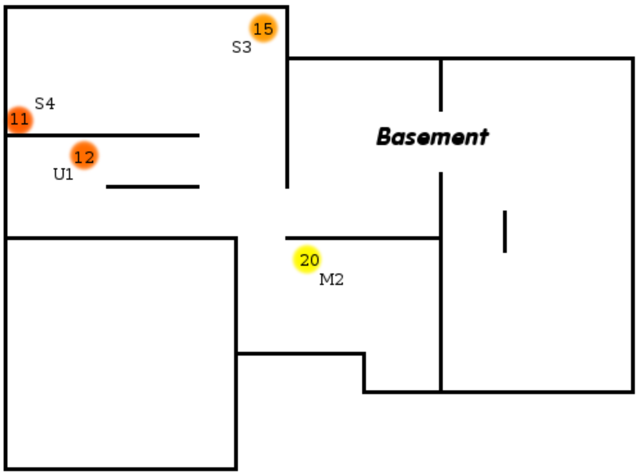
Another observation that we made was that the throughput can be very *asymmetric*. The maximal difference between up- and downstream throughput is about 9,5 Mbps. The average difference between up- and downstream is 3,1 Mbps. This was something we observed in most experiments we conducted. One of the reasons is that both devices experience another impedance of the electrical powerlines. A device sending data in one direction can only use some subcarriers (specified by a tone map), because some parts of



(a) Throughput at the second floor

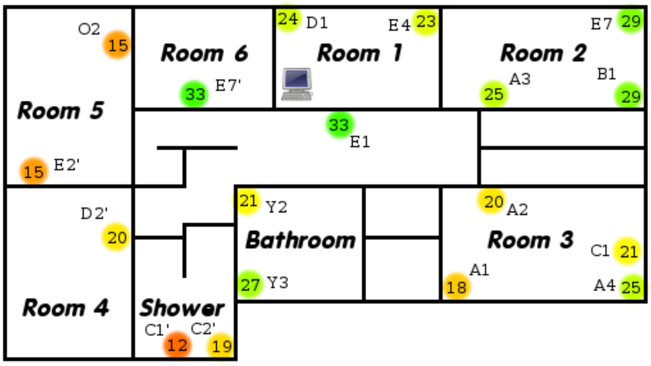


(b) Throughput at ground level

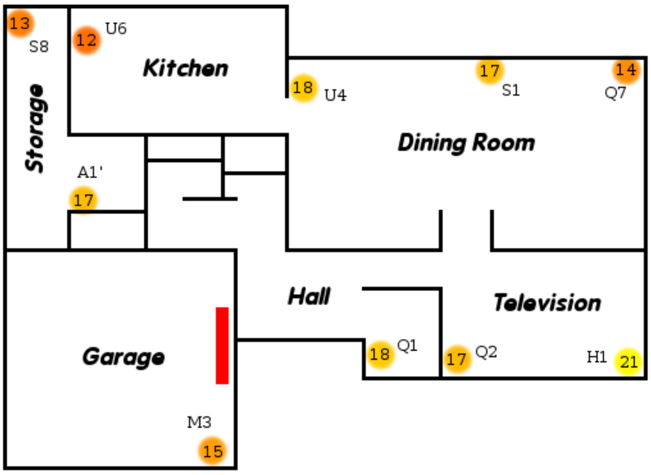


(c) Throughput in the basement

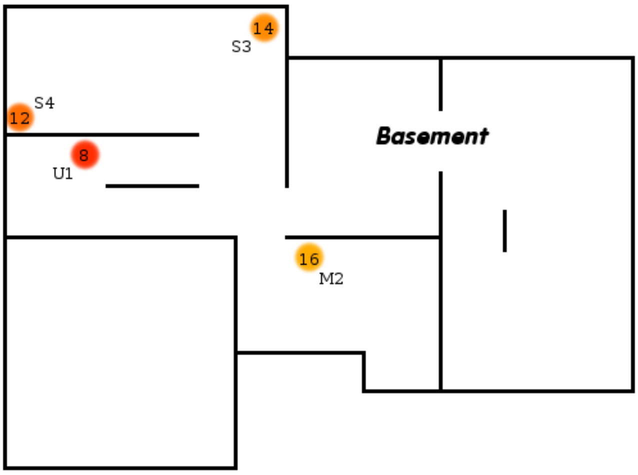
Figure 4.2: Measured throughputs in a typical home. Datastream goes from every outlet to the server.



(a) Throughput at the second floor



(b) Throughput at ground level



(c) Throughput in the basement

Figure 4.3: Measured throughputs in a typical home. Datastream goes from the server to every outlet.

Figure 4.4: Throughput legend

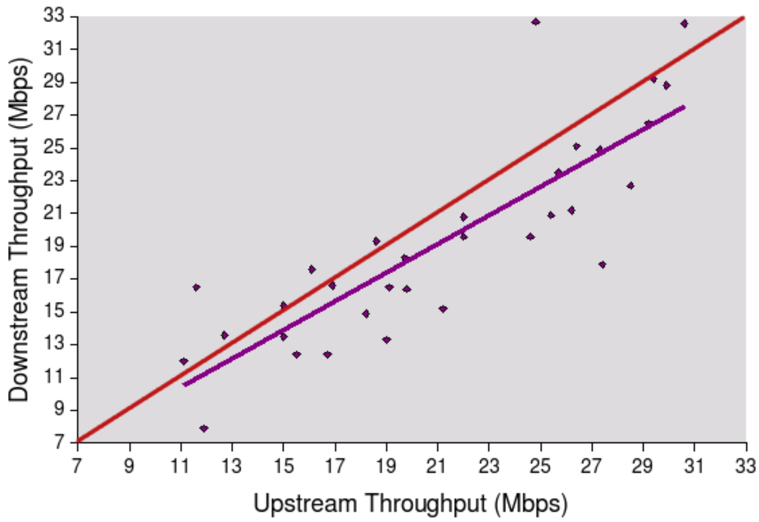


Figure 4.5: Correlation between up- and downstream throughput

the spectrum introduce too much attenuation, while another device uses other subcarriers (another tonen map) which results in another throughput.

The amount of asymmetry is visualized in figure 4.5. In this figure the correlation between up- and downstream throughput is drawn. We see an obvious correlation between up- and downstream throughput. The brown line represents the ideal case where up- and downstream throughput are exactly the same. More measurements lie on the right side of this line, which indicates that the overal average upstream throughput is slightly higher than the downstream throughput. The purple line is a linear regression of the measurements. Dots that lie further from the brown line represent measurements that had a big difference between up- and downstream throughput.

Round-trip-times are an estimation for the actual delay packets experience when sent over the network. When we assume the time it takes to reply on a ping packet is almost zero

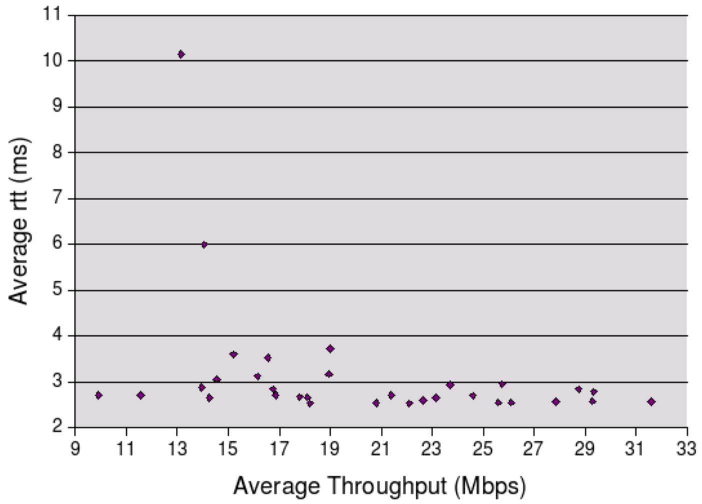


Figure 4.6: Correlation between throughput and round-trip-time

when it *enters* a network interface card, we can calculate the delay in one direction as half the round-trip-time. This is only an estimation, and in reality the round-trip-time will be asymmetric. The average round-trim-time of all measurements is 3,1 milliseconds. A maximal round-trip-time (of the averages based on the 10 consecutive round-trip-times) of 10,1 milliseconds was measured, and a minimum of 2,5 milliseconds. This maximal measurements was rather an exception, all other measurements were located more closely to the average of 3,1 milliseconds.

It could be expected that the average round-trip-time, and thus the delay, is an indication for the quality of the link. If we measure more delay, the quality of the link is bad, and less throughput is to be expected. However, this assumption is wrong, as can be seen on figure 4.6. This figure shows the correlation between the average throughput (an average of up- and downstream throughput) and the average measured round-trip-time. We see that the round-trip-time is fairly constant, with two exceptions. It doesn't decrease with increasing throughputs. Based on the round-trip-time no estimation can be done for the quality (in terms of throughput) of the HomePlug link.

Jitter was also measured throughout the house. Both up- and downstream jitter measurements differ greatly. In the upstream direction it is often only one *microsecond*. Notice that there is no other traffic on the network, so the adapters don't have to contend for the medium. An average jitter of 0,7 ms in the upstream direction and 0,3 ms in the down-

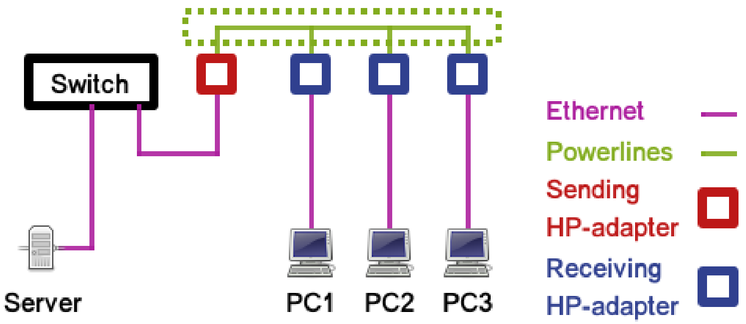


Figure 4.7: Multicast streaming – setup

stream direction is observed. This would be very good for applications like VoIP where such amounts of jitter are no problem.

4.3 Multicast streaming

An important aspect of network streaming is multicast streaming. This section explores this interesting topic for the Homeplug 1.0 standard with a turbo extension and explains how it is implemented in our test devices.

4.3.1 Multicasting with Devolo adapters

For the first test, we took a Devolo adapter as the server/transmitter. The four clients/receivers consist of three Powerline subscribers to the multicast group 239.255.255.2 and one subscriber that is connected to the switch. This last client was added to the multicast group for control purposes. Transmission and reception of the multimedia streams was done with *vlc* (Video Lan Client). To analyse the multicast stream wireshark was used. This tool could give us useful information about the bitrate and packet loss. These parameters are measured at the network layer. The transport protocol for multicast streams is UDP, which means that packets are not acknowledged by the receiver.

RTP is the actual format in which the multimedia is sent over the network. Video and audio information is embedded in the RTP packets, together with timing information for synchronized playback. The multimedia samples we used were found on the internet. The first one is a High Definition (HD) fragment at 19 Mbps. This bandwidth would be used to transmit High Definition content. A 1,7 Mbps DivX Movie is used to simulate less demanding traffic. In this test we also used a music fragment in the common mp3 format.

Multicasting a 19 Mbps HDTV fragment

We started with the initial setup as depicted in figure 4.7. A 19 Mbps HDTV fragment was multicasted to the three clients connected with 85 Mbps HomePlug adapters and the one connected directly to the switch. Wireshark was used to analyse all the streams that arrived at the clients. The results are given in table 4.6.

Table 4.6: Multicasting a HDTV fragment with a Devolo adapter, three listeners

Client	Bitrate	Packet loss
PC1	6,716 Mbps	66,45%
PC2	7,173 Mbps	63,67%
PC3	7,096 Mbps	64,20%
Sum	20,985 Mbps	

Notice that playback was almost impossible. A packet loss this high is unacceptable for smooth playback. We noticed that the sum, 21 Mbps, was the maximum bitrate that HomePlug can carry over standard powerlines. One HomePlug client was removed and the streams were measured again. The results are given in table 4.7.

Table 4.7: Multicasting a HDTV fragment with a Devolo adapter, two listeners

Client	Bitrate	Packet loss
PC1	9,073 Mbps	47,31%
PC2	10,285 Mbps	48,07%
Sum	19,358 Mbps	

Again, packet loss was too high for smooth playback. The sum of both received streams was almost the same as in the previous case. This lead to the theory that multicast packets were actually sent to each multicast subscriber individually, a sort of *simulated* multicast. This would explain why three (and even two) listeners to the multicast transmission experience such packet loss. Three listeners would require a bandwidth of 37 Mbps (3 x 19 Mbps), a throughput that these HomePlug devices can't deliver. If this was true, one listener would be able to receive the multicast stream without much loss. We disconnected PC2 and did the test again. The result is shown in table 4.8.

Table 4.8: Multicasting a HDTV fragment with a Devolo adapter, one listener

Client	Bitrate	Packet loss
PC1	17,069 Mbps	0,17%
Sum	17,069 Mbps	

Like we expected, no huge packet loss was experienced by the multicast listener. The multimedia stream now only had to be forwarded to one subscriber.

Multicasting a 1,7 Mbps DivX movie

If the assumption about the simulated multicast would be true, multicasting a 1,7 Mbps DivX movie to three clients wouldn't cause any problem. This would only require 5,1 Mbps (3 x 1,7 Mbps), a throughput that is easily delivered by the HomePlug network. Analysing the three RTP streams gave the results shown in table 4.9.

Table 4.9: Multicasting a DivX fragment with a Devolo adapter, three listeners

Client	Bitrate	Packet loss
PC1	1,750Mbps	0%
PC2	1,749 Mbps	0%
PC3	1,749 Mbps	0%
Sum	5,248 Mbps	

All clients received all packets, no packet loss was experienced. This result was expected.

4.3.2 Multicasting with Topcom adapters

For the second test, we took a Topcom adapter as the server/transmitter (the *red* square in figure 4.7). We expected the same results as with the Devolo adapters, but other behaviour was observed.

Multicasting a 1,7 Mbps DivX movie

The same test with the 1,7 Mbps DivX movie was performed, using a Topcom adapter as transmitter. The results are shown in table 4.10.

Table 4.10: Multicasting a DivX fragment with a Topcom adapter

Client	Bitrate	Packet loss
PC1	0,094 Mbps	53%
PC2	0,111 Mbps	93%
PC3	0,089 Mbps	95%
Sum	0,294 Mbps	

When the same movie was multicasted using a Topcom adapter the movie couldn't be played back smooth by the three clients (no playback at all was possible). A high packet loss was observed. This would mean that Topcom uses another method to send multicast packets over the HomePlug network. Here, the sending adapter switches to the ROBO mode (see page 10). Only a small throughput can be delivered with this technique.

Multicasting a 256 Kbps mp3 song

If the Topcom adapters switch to ROBO mode, sending an mp3 over the HomePlug network wouldn't be a problem. The mp3 was coded at a 256 Kbps bit-rate. The number of multicast subscribers is irrelevant in this case. See table 4.11 for the results.

Streaming the mp3 didn't cause a problem to any of the clients.

Table 4.11: Multicasting an mp3 with a Topcom adapter

Client	Bitrate	Packet loss
PC1	0,257 Mbps	0%
PC2	0,257 Mbps	0%
PC3	0,257 Mbps	0%
Sum	0,771 Mbps	

Using iperf to measure multicast throughput

To strengthen our assumption that the Topcom adapters switch to the ROBO mode to deliver multicast packets, we used iperf. Iperf has the option to transmit packets to a multicast address. The servers have to be instructed to listen to the multicast address, while the client can send packets to this multicast address. Notice that many iperf servers can be started at different stations connected to the same network, while only one iperf client may send to this address. The server is started with this command: *iperf -s -u -B 239.255.255.2*. The client is started with *iperf -c 239.255.255.2 -u -t 15 -i 5*. The *-B* option makes the server bind to address 239.255.255.2. It's obvious that UDP must be used, since multicast packets can't be acknowledged like with TCP.

We measured a throughput of 580 Kbps. The client was moved to several outlets, but everywhere the same throughput was measured. This is expected, because when the Topcom adapters use ROBO mode for multicast operations, only a fixed small throughput is available, no matter how far (within the 200m range) the adapters are separated from each other.

4.3.3 Multicasting with Devolo AVdesk adapters

Since we only had two HomePlug AV devices to our disposition another setup must be used to test the multicast behaviour for these adapters. A simple network between two computers was set up. One pc acted as a server, while the other pc acted as a subscriber to the multicast group (239.255.255.2). Because the HomePlug AV standard is an improvement of the older 1.0 standard we expected better multicast capabilities from these devices. We tested with both vlc and iperf.

Multicasting a 19 Mbps High Definition movie resulted in 97% packet loss. *Unicasting* the same movie fragment in the same situation gave no problems. Even a 3 Mbps Standard Definition movie couldn't be multicated and played back smooth, because 70% of all packets were lost. Multicasting a 500 kbps DivX movie gave no problems.

Iperf can give a more precise indication of the maximal throughput while multicasting. For this test we used these commands:

```
iperf -c 239.255.255.2 -u -t 5 -b 850k (client)
iperf -s -u -B 239.255.255.2 (server)
```

The speed at which the client sends the UDP packets is specified with the option *-b 850k*. This value was varied and it was observed that a maximal throughput of 820 kbps was available.

4.3.4 Conclusion

With these tests we can already notice that multicasting is implemented different in both HomePlug 1.0 Turbo adapters. Devolo adapters simulate multicasting by unicasting frames to every subscriber while Topcom adapters provide "real" multicasting. This has some radical effects on the performance while multicasting. Topcom switches to the so-called ROBO mode where he transmits at a maximal constant rate of ± 580 kbps no matter how many listeners. Devolo's speed on the other hand decreases as the number of subscribers to that specific multicast address rises. So Devolo utilises the full bandwith capacity and divides it under the receivers while Topcom uses a garanteed but fixed and very low rate for its transmission.

The Devolo dLAN 200 AVdesk adapters also switch to a robust mode, like the Topcom adapters, to ensure reception of the frames at all outlets. The maximal multicast throughput here is ± 820 kbps. Although these adapters are designed to deliver a high QoS, they don't allow high bandwidth (multimedia) streams to be multicasted throughout the house. This is a major disadvantage.

When only a moderate number of HomePlug 1.0 clients subscribe to a specific multicast stream, the Devolo dLAN Highspeed 85 adapters can be used. High definition content can't be multicasted to more than one client. A simple unicast would give the same result, so the advantages of multicasting are completely undone. If many HomePlug 1.0 subscribers have to be serviced, only using a small bandwidth, it's better to use the Topcom adapters. If for example many clients want to listen to an audio stream (i.e. digital radio) the throughput that is available in ROBO mode is sufficient. The number of subscribers is unimportant in this case.

Multicasting will always be a problem for (broadband) powerline communication. Communication is optimized between two nodes. If one packet has to be sent to several nodes, no such optimization can be used. Switching to a more robust and redundant mode is an option. Emulating multicast is another. Ethernet has the advantage that all nodes on the same segment *see* all packets, so multicast packets are actually the same as unicast packets. Since multicasting will be used in modern home networks, this can get very important. Using HomePlug as underlying backbone of the in-home network will have it's implications on multicast communication.

4.4 Broadcast

Broadcasting is a slight variation on multicasting. If a packet has to reach all the clients of a network it's more efficient to broadcast the packet than to send a copy to each client separately. Moreover, this would require the sending station to know the addresses of all the clients on his network. On the *Network Layer* two kinds of broadcast techniques exist. An IP address exists of two parts: a network-id portion and a host-id portion. If a packet is sent to the network-id, followed by all 1's, it is sent to all the participants of that particular network. This is called a *Directed Broadcast*. The associated IP address is called the Broadcast Address. An example of a broadcast address is 192.168.0.255 (with subnetmask 255.255.255.0). Another type of broadcast is the *Limited Broadcast*. The associated IP address consists of thirty-two 1's (255.255.255.255). A packet sent to this address is delivered to all participats of the network, only, in this case, the sender of such a packet has to belong to the same network as the receivers.

The way frames are broadcasted over the network is dependant on the underlying network technology. As explained before, ethernet has no problems to broadcast a packet over the network. The *all 1's* MAC address (ff:ff:ff:ff:ff:ff) is reserved for broadcast communication at the *Data Link Layer*. HomePlug on the other hand has no efficient way to forward frames to all participants of the network.

Luckily, most applications don't require much broadcast packets to be sent. One of the most important uses of limited broadcast is DHCP (Dynamic Host Configuration Protocol). This protocol gives clients the possibility to configure their network interface at start-up. Since such an interface has no information about the DHCP server at boot time, it broadcasts a DHCP request over the subnet. In some cases the reply of the DHCP server is broadcasted back. Since DHCP is so fundamental to set up a network, broadcast is a feature that the underlying technology must support. In homes where many new devices are connected regularly DHCP must be available to configure them dynamically. However, this protocol doesn't requires much bandwidth.

Care must be taken with broadcast packets since all connected devices must accept these packets and interpret them. Broadcasting all packets would require much overhead of the connected devices, since all packets would have to be inspected and interpreted by all stations.

DHCP was no problem for any of the HomePlug adapters we could test. All the HomePlug devices forwarded the DHCP requests correct, and an IP address could be obtained easily. As long as no great bandwidth is required to broadcast packets, no problems are to be expected. Protocols like DHCP work reliably over a HomePlug network.

4.5 Multiple concurrent streams

To understand how HomePlug works in a more home-like environment, we performed a four-node test. In this test four identical pc's were connected using four adapters. We want to know what the behaviour is when several pc's are sharing the same medium for network connectivity, in this case the electrical wiring. In a real-life situation, many applications will demand bandwidth from the network. When HomePlug is used as a backbone for the home-network, it must be able to transport different streams of information (video, voice,

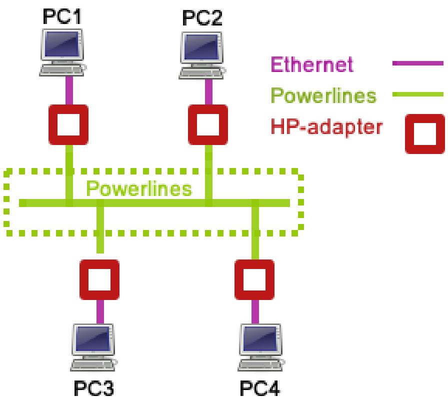


Figure 4.8: Four-node test – setup

audio, ...) simultaneously. The network should know how to handle this, and divide the total bandwidth among all demanding applications.

4.5.1 Description

To simulate this situation (a simple situation though, but easy to analyse) we use iperf to measure the throughput simultaneously between the pairs. PC1 and PC2 form the first pair, PC3 and PC4 form the second pair. All pc's are connected to the HomePlug network with dLan Ethernet Highspeed 85 adapters (adapters with the turbo extension). The set-up requires four pc's and was conducted in the pc laboratory at school. The electrical wiring is *not* representative for a typical home (noise from many running computers, impedance from all the connected appliances, ...). This is not a problem though, because we are interested in the relative throughputs.

The tests were conducted both with TCP and UDP packets. For TCP we used the following commands to measure the throughput:

```
iperf -s -i 5 (server)
iperf -c 10.0.0.1 -t 60 -i 5 (client)
```

For UDP we used these commands:

```
iperf -s -u -i 5 (server)
```

```
iperf -c 10.0.0.1 -u -t 60 -i 5 -b 20m (client)
```

When a higher bandwidth than 20 Mbps was achievable, we instructed iperf to send at a higher rate, so iperf wasn't the bottleneck. Notice that both UDP and TCP tests run for one minute. This way we got a good average and made sure that starting iperf at both clients at exactly the same time wasn't critical. Errors made at the first tenths of a second, and last tenths of a second, because iperf couldn't be started exactly at the same time at both clients, can be neglected.

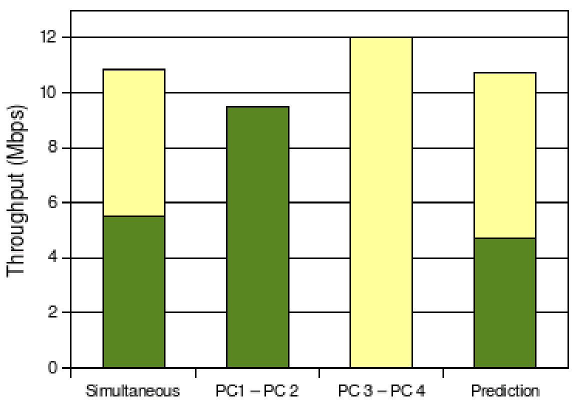
The test was first performed with four Devolo adapters (dLan Ethernet Highspeed 85). The same test was repeated with two Devolo adapters and two Topcom adapters (PowerLan 200 Turbo), both forming one pair. This was done to simulate a real-life setting where different adapters could be used to form one network. Both adapters in this test support the HomePlug 1.0 standard and have a turbo mode that improves performace up to 85 Mbps (at the physical layer).

We decided to perform all the tests when the adapters of the pairs are in a different logical network, and once again when they are in the same logical network. Thus first the security ID's of the two pairs of adapters were set differently from each other. Later, all the adapters were provided with the same security ID, so all the pc's were in the same logical network and could see each other.

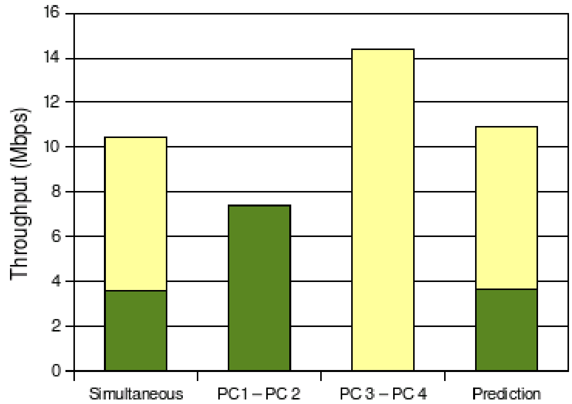
4.5.2 Results

The results are represented in figure 4.9 and 4.10. The first bin represents the total throughput when two stations are sending data simultaneously. The green part of this bin is the throughput available between PC1 and PC2, the yellow part is the throughput available between PC3 and PC4. The second and the third bin represent the throughput measured when the other pair isn't sending any data. This throughput can be obtained if the two adapters would be the only adapters on the same powerline.

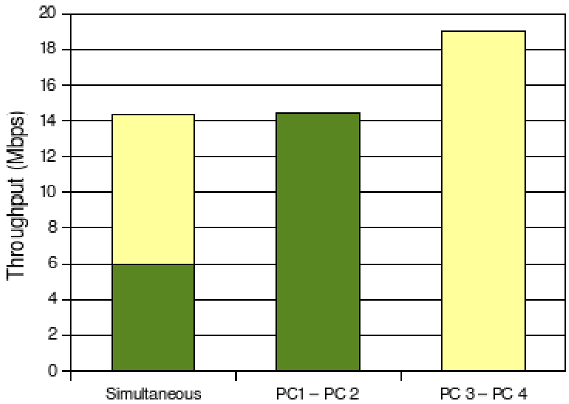
Figure 4.9 represents the measurements from the tests with adapters in the same logical network, while 4.10 represents the measurements from the tests with adapters located in two different logical networks.



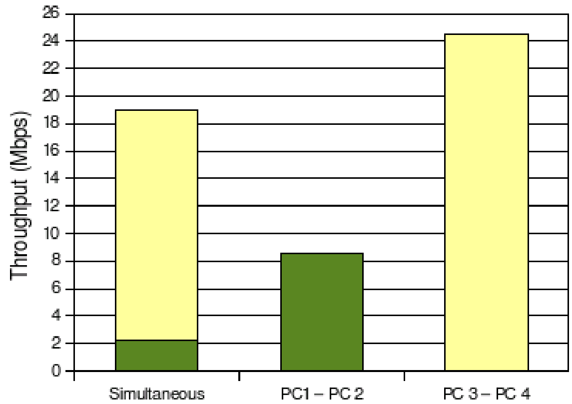
(a) TCP, same logical network, Devolo to Devolo



(b) TCP, same logical network, Devolo to Topcom

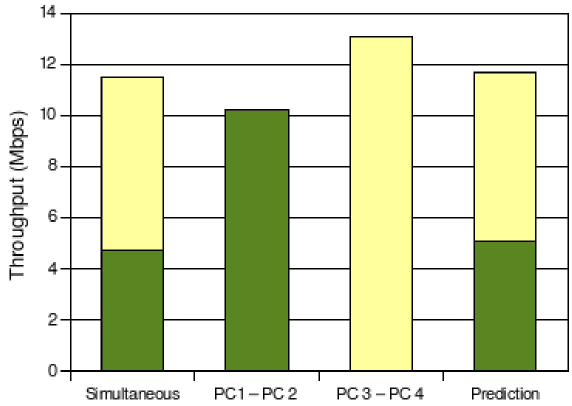


(c) UDP, same logical network, Devolo to Devolo

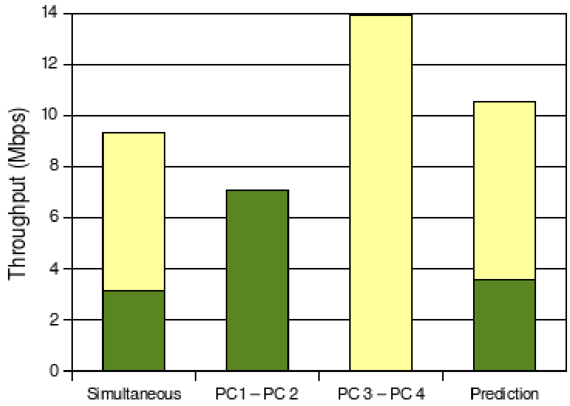


(d) UDP, same logical network, Devolo to Topcom

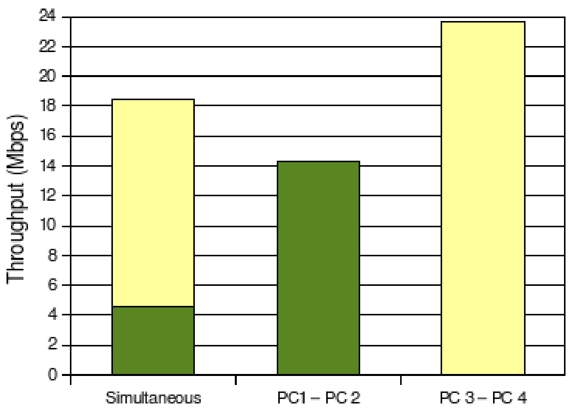
Figure 4.9: Throughput with four nodes on the same logical network



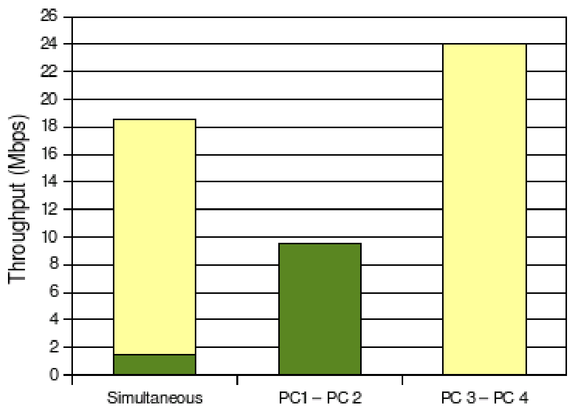
(a) TCP, different logical network, Devolo to Devolo



(b) TCP, different logical network, Devolo to Topcom



(c) UDP, different logical network, Devolo to Devolo



(d) UDP, different logical network, Devolo to Topcom

Figure 4.10: Throughput with four nodes on a different logical network

4.5.3 Conclusion

It is observed that the connection between PC3 and PC4 always offers a higher throughput than the connection between PC1 and PC2. This is a result of this specific setup. The impedance and noise are less favorable for PC1 and PC2. In all tests the resulting total throughput, when the two networks are sending data, is less than the throughput between PC3 and PC4. This means that no additional gain in throughput can be obtaind when using two different HomePlug networks, instead of just one.

Another observation that can be made is that the throughput isn't divided honestly. The capacity of the link between PC1 and PC2 is always less than the capacity between PC3 and PC4 when both pairs are sending data simultaneously. The most extreme measurement is seen in figure 4.10(d) where PC1 and PC2 only get about 1 Mbps, while PC3 and PC4 get more than 18 Mbps. The link that already has less bandwidth to it's disposition suffers even more when it has to share the medium with other HomePlug devices.

The differences between adapters in the same logical network and adapters in different logical networks is minimal. No additional gain or loss of throughput is observed. Thus, dividing a HomePlug network in different logical networks does not alter the performance. Adapters in two different logical networks still see the preambles and frame control fields of the adapters in a different logical network. All the adapters have to content in the same contention period and signal their priority in the same priority resolution interval.

A prediction of the simultaneous throughputs can be made based on the throughput measured between the pairs when sending separately. This prediction only holds for TCP connections though. The cumulated throughput when sending simultaneous is aproximately the average of the throughput between the pairs. The proportions of the individual streams in this total throughput is almost the same as relation between the individual throughput (when the other pairs is silent) and the sum of the two individual throughputs. This prediction is shown as the fourth bin in figure 4.9(a), 4.9(b), 4.10(a) and 4.10(b). The error made by the prediction is never higher than 1 Mbps for the individual proportions.

4.5.4 HomePlug AV and 1.0 concurrent streams

To see how HomePlug AV and HomePlug 1.0 (turbo) networks influence each other we made the same setup as shown in figure 4.8. Here PC1 and PC2 are connected to the powerlines using HomePlug 1.0 turbo adapters (Devolo dLAN Highspeed 85) while PC3 and PC4 are connected using HomePlug AV adapters (Devolo dLAN 200 AVdesk). These were te commands used to measure the throughput:

```
iperf -s -w 512k (PC1)
iperf -c 192.168.0.1 -w 512k -f m -i 5 -t 60 -r (PC2)
iperf -s -w 512k (PC3)
iperf -c 192.168.0.3 -w 512k -f m -i 5 -t 60 -r (PC4)
```

First, only PC1 and PC2 were used to measure the throughput, then PC3 and PC4 were used. After these measurements both networks were measured together. The results are shown in table 4.12.

Table 4.12: HomePlug AV and 1.0 concurrent streams

	1.0 network		AV network	
	up (Mbps)	down (Mbps)	up (Mbps)	down (Mbps)
PC1 – PC2	16,0	15,6	-	-
PC3 – PC4	-	-	41,7	42,8
Simultaneous	7,28	7,29	13,6	12,1

From these measurements we can see that there is a serious drop in throughput when both networks are occupied simultaneous. This is expected because both PLC technologies work in frequency ranges that overlap. When both networks are occupied, the HomePlug 1.0 throughput drops to 47% of the original available throughput. The HomePlug AV network experiences an average drop of 69%! The impact of a neighbouring HomePlug 1.0 (turbo) network on a HomePlug AV network is very big.

Replacing the two AV adapters with Devolo dLAN Highspeed 85 adapters resulted in the measurements shown in table 4.13.

Table 4.13: HomePlug 1.0 concurrent streams

	1.0 network #1		1.0 network #2	
	up (Mbps)	down (Mbps)	up (Mbps)	down (Mbps)
PC1 – PC2	22,2	15,1	-	-
PC3 – PC4	-	-	20,3	23,7
Simultaneous	10,0	8,16	9,21	10,1

These measurements were done to see how a HomePlug 1.0 network interferes with an other HomePlug 1.0 network. It is basically the same test as described in section 4.5. Here we see that network #1 experiences a throughput loss of 51%, and network #2 experiences a loss of 56% in the same situation. This is far less than the impact of a HomePlug 1.0 network on a HomePlug AV network.

4.6 Compatibility between HomePlug Turbo devices

Since we had different adapters from different manufacturers to our disposition, we were interested in how well they cooperate with each other. Since the turbo extension is not a standard of the HomePlug Alliance, it could well be possible that these devices can only communicate at the 14 Mbps speed that HomePlug 1.0 provides. The setup was easy. A network was set up between two computers, using the HomePlug adapters. Since we had two kinds of adapters (Devolo dLan Ethernet Highspeed 85 and Topcom PowerLan 200 Turbo) four possible combinations were possible. The maximum TCP throughput was measured with all combinations. Again, the iperf tool was used with the following commands:

```
iperf -s -i 5 (server)
iperf -c 10.0.0.1 -r -t 15 -i 5 (client)
```

The results are given in table 4.14.

Table 4.14: Thourghput between Devolo and Topcom adapters

PC1	PC2	up (Mbps)	down (Mbps)
Devolo	Devolo	25,3	16,6
Topcom	Topcom	26,0	15,0
Devolo	Topcom	26,8	14,9
Topcom	Devolo	24,4	13,9

4.6.1 Conclusion

Both adapters seem to perform equally well. Upstream (from PC1 to PC2) the average throughput is 25,6 Mbps, while downstream (from PC2 to PC1) the average throughput is 15,1 Mbps. It seems logical that both adapters are compatible, because they are both based on the same Intellon chipset (INT5500).

Compatibility between HomePlug 1.0 adapters of different manufacturers is mandatory. Also, adapters that comply with the HomePlug AV standard have to be able to communicate with each other. After all, thats the whole point of the HomePlug (1.0 and AV) standard.

4.7 Comparing throughput of HomePlug 1.0 turbo and AV devices

Since we also had two Devolo dLAN 200 AVdesk adapters to our disposition it's interesting to know how much throughput is available using these adapters. It's even more interesting to compare this throughput with the throughput available using the Devolo dLAN Highspeed 85 adapters. Thoughput was measured with the iperf tool using these commands:

```
iperf -c 192.168.0.1 -t 30 -r -i 5 -w 512k (client)
iperf -s -t 30 -r -i 5 -w 512k (server)
```

Notice that the option *-w 512k* specifies a TCP window size of 512 Kbyte. This window size is important because it indicates how much TCP packets can be sent to the receiver without receiving an acknowledgement. This improves throughput greatly because the sender hasn't got to wait to send the next packet until it receives an acknowledgement. Not specifying the *-w 512k* option reduces the throughput of the AV adapters significatly, to about 25 Mbps!

Both AV and 1.0 turbo adapters were placed at the same outlets. With the HomePlug AV adapters a throughput of *57 Mbps* was obtained while with the 1.0 turbo adapters a throughput of *21 Mbps* was obtained at the same outlet pair. This means that the throughput can be improved with a factor 2,7 in this situation when using Devolo dLAN 200 AVdesk adapters instead of dLAN Highspeed 85 adapters.

5 Home-network applications

Previous chapters already explained the HomePlug technology and the physical layer and it's problems. Also extensive throughput tests were carried out in a modern house to have a better overview and understanding of the QoS of HomePlug in such an environment.
Now we arrive at the highest layer of the OSI stack: the application layer. With all the knowledge from previous chapters, it is interesting and useful to look one step further and think about what the needs are of todays families. To obtain this high-level overview, a number of user scenarios for different family situations were made. The goal of this chapter, therefore, is to see how the discussed HomePlug technology suffices to the needs of modern families taking the most recent and popular applications into account. Let us first start with the discussion of some popular applications/protocols before we take a look at the complete picture with the user scenarios.

5.1 UPnP

A first popular and rapidly growing technology is UPnP (universal plug and play). It should better be called: network plug and play because that exactly describes what it does. It provides the ability to plug and play network devices. This should be the technology to interconnect almost every device that would want to communicate with other devices. It works with ethernet but the interoperability with HomePlug until today is rather vague.
To evaluate UPnP with HomePlug technology, tests were done with some popular UPnP tools: the Intel tools (media renderer, media server and media controller), On2Share (media renderer and media server), TwonkyVision (media server) and the built-in WindowsXP UPnP service. A media server shares/streams media-data content to UPnP-clients on the network, a media controller can browse the content from auto-detected media servers on the network and a media renderer can render the content.

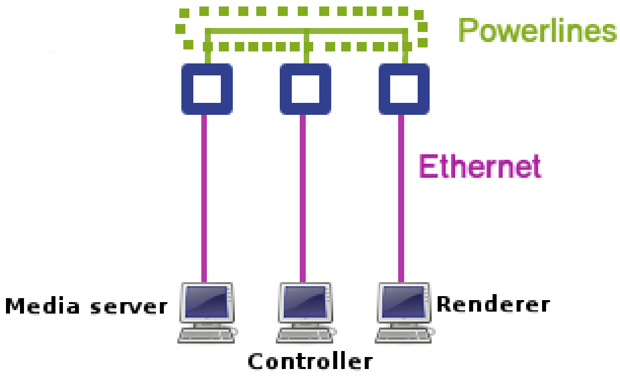


Figure 5.1: UPnP with Topcom adapters – setup

The tools from Intel were installed on three different workstations (see Figure 5.1) to provide each station with a unique function: renderer, controller or media server. The workstations were connected to the network with Topcom adapters. The renderer didn't display images or movies (only the sound). With On2Share as media renderer and media server and the Intel tool as a control point, images were shown and movies were played. Although not every movie worked. It seems that some tools (the renderer tools) can't process movies (or files in general) larger than a fixed size. We noticed that in this setup a video of 170Mb was too large and a video of 70Mb worked perfect. When the file limit is reached for that particular tool, zero window advertisements are being sent to the media server and therefore the streaming stops. The server probes the renderer but the streaming will never be re-initiated or continued and consequently the video will never be played. It should be noted however that this is also the case with ethernet! So let us not blame the HomePlug technology too fast.
When Devolo adapters are used to connect a device to the network, UPnP doesn't work at all. To investigate this strange behaviour, a network packet capture by ethereal/wireshark was carried out. While looking at the capture it became clear: Devolo adapters do not send or receive the SSDP packets used to discover the offered services on the network. This is because these packets are sent to a multicast address and as seen before in section 4.3 Devolo turbo adapters have some issues regarding multicasting.
Topcom adapters, on the other hand, have no problems with these packets and consequently can be used for UPnP purposes.

To conclude, the problem with UPnP and HomePlug lies mostly in the unpredictability of the tools currently available: large files (like movies) can not be streamed and played with current UPnP applications (as also the case with ethernet). This is because most tools do not seem to support real streaming (like RTP streaming) but instead they do HTTP-streaming. Anyhow, UPnP and Devolo Turbo adapters are no good match. The AV desk adapters from Devolo, on the contrary, will not form a problem since multicasting occurs in a ROBO mode just like Topcom adapters. A lot of work and research still has to be done to guarantee flawless UPnP operation with HomePlug.

5.2 VoIP

Normal PSTN telephony or the POTS (plain old telephone service) is meeting it's worst enemy: Voice over IP (VoIP). With their low prices as their biggest advantage, VoIP service providers are emerging at a rapid pace. Although POTS have their reliability, customer's trust and some calling features to count on, they can not stop the VoIP train from moving. VoIP telephony is used more and more and is reaching the common home-user. Because of this significance in today's world, VoIP will be tested with the HomePlug technology in this section.

5.2.1 Introduction

Right now there are two different networks, a data communications network and a partial analog telephone network, that both need to be managed and maintained. The underlying technology used in these two networks differ greatly, but the Internet has evolved and is now able to deliver voice-services. Using the Internet for voice communication has the big advantage that only one network, the data network, needs to be managed and maintained, and thus also paid for. We already see companies migrating their analog telephone systems to digital systems.

Unlike traditional telephone networks, which establish a real-time connection between the two calling parties, a VoIP call conveys the voice data over an IP network. As we know, IP packets can get delayed, corrupted or even dropped. To get a pleasant telephone conversation the underlying IP connection of a VoIP call must be stable and deliver a high

QoS. Telephone companies are offering VoIP services. Programs like Skype are gaining interest because it makes telephone calls cheaper or even free when the telephone call is between two computers. By purchasing a special bundle you can call with a program like Skype to a normal telephone number, at rates much lower than the traditional rates using two analog phones. The downside of Skype is that it is proprietary, and it uses proprietary protocols. Luckily other protocols have emerged and are not closed for the public. Two important protocols for VoIP are H.323 and SIP (Session Initiation Protocol). SIP is the most supported protocol, with several hardware implementations available today.

To make a (SIP) VoIP call you can use:

- a softphone,
- a hardware (SIP) VoIP telephone,
- an analog telephone, connected to the IP network using an ATA (analog telephone adapter).

A softphone is by far the easiest solution for testing. A softphone is a software program that runs on an ordinary computer, and uses the network card to connect to the IP network. X-lite, Ekiga and Yate are examples of softphones.
The other two solutions both require new hardware. SIP telephones just need to be connected to the home-network (e.g. using an ethernet interface or WiFi). They replace the analog telephones. The ATA solution is cheaper, because the old analog telephone doesn't need to be replaced. Instead, the ATA makes it possible to connect the analog phone to the digital (local area) network. This way the end users don't even notice a change when migrating from analog to VoIP.
As stated before, a high QoS is needed to setup a pleasant conversation. Because a telephone call is interactive, delays must be kept below a maximum. Also jitter must be avoided, because aditional jitter, requires a large de-jitter buffer, adding to the overal delay. The total delay must not exeed 150 ms. VoIP also requires a constant bandwidth. This is usually not a problem. Depending on the codec used, a VoIP conversation doesn't consume much of the total available throughput. For example, a VoIP call compressed with the G.711 codec requires a bandwidth of 172kbps.

Table 5.1: VoIP conversation on LAN between two softphones using Yate

Client location	RTT		Throughput	
	Client 1	Client 2	Client 1	Client 2
Basement (U1)	2ms	2ms	8,18Mbps	7,8Mbps
Kitchen (S7)	2ms	2ms	14,1Mbps	17,1Mbps
Living room (Q6)	2ms	8ms - 151ms	7,35Mbps	3,97Mbps
Main Hall (Q1)	2ms	2ms	9,82Mbps	17,0Mbps
Hallway (E1)	2ms	2ms	8,36Mbps	12,9Mbps
Room upstairs (O2)	2ms	2ms	8,7Mbps	11,4Mbps

5.2.2 VoIP within the home

We first tried to setup a VoIP conversation within the home (the same house as in section 4.2). We did this to get an idea of how this technology works, without having to worry about the influence of the Internet between the two calling parties.
To be able to setup a SIP VoIP conversation we need a SIP server where both phones can register themselves. For this purpose we used the Yate (yet another telephony engine) SIP server. By changing the configuration file (/conf.d/regfile.conf) we added some acounts. Entries in this file should be in the form: [username] password=secret
We used the Yate softphone to make a call in the house. One desktop PC was configured to run the Yate server and was located in a bedroom (room 1) on the first floor, while two laptops used the Yate softphone to make the call. The server was connected to the network by means of an ordinary ethernet connection to a switch. This switch was connected to the HomePlug network using a Devolo ethernet-to-HomePlug adapter. The two laptops were connected to the network using the same type of adapters.
One laptop stayed fixed in the home (located in the basement, outlet S2), while the other laptop was located at six different locations throughout the house. We measured the round-trip-time to the server with the ping command from both laptops. We also measured the TCP throughput between the two clients to get an idea of the quality of the connection between the two clients. These measurements are shown in table 5.1.

Notice the round-trip-times in the living room which ranged from 8ms to 151ms. The available bandwidth from the client at this location to the fixed client is also noticable smaller. Next to this outlet a power adapter was plugged in. This might have caused the bad results. Measurements at the outlet in the main hall, which is connected to the same electrical branch, didn't suffer from large delays.

Communication between the clients went smooth, except at the outlet in the living room. Using Wireshark to analyse the RTP stream, we noticed that all packets were sent from the clients to the server, and then forwarded to the other client. So all packets go through the server. Using the SIP protocol this shouldn't be the case because after the call-setup all communication should be between the clients directly and should not go via the server. This could be the result of a bad configuration of the SIP server. As told in the previous section, jitter and delay are very important QoS parameters when dealing with VoIP. But because we are still physically within the same house (also with an in-home SIP server), we don't have to connect with the Internet. Therefore the jitter peaked never above 10ms (an average of 5ms) and the delay was always way below the maximum of 150ms. These parameters, however, will need a more thorough investigation when dealing with conversations over the Internet.

5.2.3 VoIP over the Internet

Now, VoIP conversations will be made and investigated between users across the Internet. First the additional delay that HomePlug adapters introduce, will be tested. Secondly the jitter differences between HomePlug and ethernet will be addressed. Thirdly a SIP call is going to be set-up and analysed between two VoIP softphones.

Influence on delay

To measure the influence that HomePlug adapters have on delay with VoIP (but of course also other applications), we compared HomePlug's Round Trip Times (RTT) to a certain server with the ones from ethernet to that same server.
The tool used for this purpose is VisualRoute. This program includes integrated traceroute, ping tests, reverse DNS and Whois lookups, and displays the actual route of connections and IP address locations on a global map. The feature we used is the visual route tracing

Hop	%Loss	IP Address	Node Name	Location	Tzone	ms	Network
0		192.168.1.2	Terminator	*			(private use)
1		192.168.1.1	-	...		11	(private use)
2		195.95.30.13	-	(Belgium)	01:00:00	11	Scarlet
3		194119228225	gig0-1.exc-ar01.ias.scarlet.be	(Belgium)	01:00:00	11	Scarlet
4		194.119.226.13	ser2-0.bel-pe06.mpl.scarlet.be	(Belgium)	01:00:00	11	Scarlet
5		64.209.92.37	g2-2.ar1.BRU1.gblx.net	Brussels, Belgium	01:00:00	12	Global Crossing GBLX-11A
6		67.17.105.6	ge4-1-10G.ar2.SJC2.gblx.net	San Jose, CA, USA	-08:00:00	168	Global Crossing GBLX-13
7		64.209.88.126	Asia-Netcom-Corporation.ge-7-1-0.409.ar1.SJC2.gblx.net	San Jose, CA, USA	-08:00:00	166	Global Crossing GBLX-11A
8		202.147.0.57	po14-1-0.cr1.nrt1.asianetcom.net	...		281	Asia Netcom Corporation
9		202.147.0.170	po5-2.cr1.hkg3.asianetcom.net	...		334	Asia Netcom Corporation
10		202.147.16.206	po0-0.gw2.hkg3.asianetcom.net	...		330	Asia Netcom Corporation
11		203192137174	CNI-0001.gw2.hkg3.asianetcom.net	(Hong Kong)	08:00:00	336	Asia Netcom HKG HUB
12		159.226.1.133	-	Beijing, China	08:00:00	342	CHINA SCIENCE AND TECHNOLOGY NETWORK
13		159226254105	8,2	Beijing, China	08:00:00	343	CHINA SCIENCE AND TECHNOLOGY NETWORK
14		159.226.254.86	8,22	Beijing, China	08:00:00	343	CHINA SCIENCE AND TECHNOLOGY NETWORK
15		192.168.46.254	-	...		374	(private use)
16		192.168.254.5	-	...		354	(private use)
17	10	159.226.202.4	-	Beijing, China	08:00:00	343	CHINA SCIENCE AND TECHNOLOGY NETWORK
18	10	159.226.202.44	www.cnnic.cn	Beijing, China	08:00:00	343	CHINA SCIENCE AND TECHNOLOGY NETWORK
Roundtrip time to www.cnnic.cn, average = 343ms, min = 343ms, max = 347ms -- 8-mrt-2007 13:38:58							

Figure 5.2: VisualRoute – different hops and average RTT to www.cnnic.cn measured with HomePlug

of network packets to their destination.
The servers that we tried to reach, are located across the globe. First we try this with a HomePlug adapter that is connected to the Internet (through a router). Afterwards we study the same route but then with a standard ethernet connection (through the same router). The distance estimate between the PC with ethernet and the router amounts to 10 meters and the one between the HomePlug adapter and the router amounts to approximately 20 meters. This difference however may be neglected in the conclusions.
As an example, snapshots from VisualRoute are shown (Figure 5.2 and 5.3) when tracing the route to a chinese website: www.cnnic.cn

From table 5.2 one can already notice the minor differences in RTT between the 2 technologies. The average difference is approximately 1,5ms. So HomePlug does introduce additional delay but one should keep in mind that these measurements are round trip times which measure the time to go to the server and back! The real delay is only half of it! This and also the fact that 1,5ms is just a small portion of the total RTT to a server make us conclude that, although HomePlug adapters create a little amount of delay smaller than 1ms, it may be neglected when evaluating a standard route across the Internet.

Hop	%Loss	IP Address	Node Name	Location	Tzone	ms	Network
0		192.168.1.6	tiffany	*			(private use)
1		192.168.1.1	-	...		8	(private use)
2		195.95.30.13	-	...		10	Scarlet
3		194119228225	gig0-1.exc-ar01.ias.scarlet.be	...		9	Scarlet
4		194.119.226.13	ser2-0.bel-pe06.mpl.scarlet.be	...		9	Scarlet
5		64.209.92.37	g2-2.ar1.BRU1.gblx.net	Brussels, Belgium	01:00:00	10	Global Crossing GBLX-11A
6		67.17.105.6	ge4-1-10G.ar2.SJC2.gblx.net	San Jose, CA, USA	-08:00:00	166	Global Crossing GBLX-13
7		64.209.88.126	Asia-Netcom-Corporation.ge-7-1-0.409.ar1.SJC2.gblx.net	San Jose, CA, USA	-08:00:00	164	Global Crossing GBLX-11A
8		202.147.0.57	po14-1-0.cr1.nrt1.asianetcom.net	...		280	Asia Netcom Corporation
9		202.147.0.170	po5-2.cr1.hkg3.asianetcom.net	...		335	Asia Netcom Corporation
10		202.147.16.206	po0-0.gw2.hkg3.asianetcom.net	...		330	Asia Netcom Corporation
11		203192137174	CNI-0001.gw2.hkg3.asianetcom.net	...		334	Asia Netcom HKG HUB
12		159.226.1.133	-	...		342	CHINA SCIENCE AND TECHNOLOGY NETWORK
13		159226254105	8,2	...		367	CHINA SCIENCE AND TECHNOLOGY NETWORK
14		159.226.254.86	8,22	...		341	CHINA SCIENCE AND TECHNOLOGY NETWORK
15		192.168.46.254	-	...		381	(private use)
16		192.168.254.5	-	...		358	(private use)
17		159.226.202.4	-	...		341	CHINA SCIENCE AND TECHNOLOGY NETWORK
18		159.226.202.44	www.cnnic.cn	...		342	CHINA SCIENCE AND TECHNOLOGY NETWORK
Roundtrip time to www.cnnic.cn, average = 342ms, min = 341ms, max = 347ms -- 8-mrt-2007 4:30:01							

Figure 5.3: VisualRoute – different hops and average RTT to www.cnnic.cn measured with Ethernet

Table 5.2: VisualRoute – measuring RTT across the globe

		With HomePlug		Without HomePlug	
		RTT local	RTT to server	RTT local	RTT to server
Leuven	www.madoka.be	2	15	0,7	13
Netherlands	www.speedtest.nl	2	21	0,7	19
America	www.google.com	2	19	0,7	16
Asia	www.cnnic.cn	2	343	0,7	342
Australia	www.aph.gov.au	2	339	0,7	338
France	www.ekiga.net	2	25	0,7	23

Influence on jitter

A second important QoS parameter with VoIP is jitter. To see if there is any difference between HomePlug and ethernet regarding jitter and VoIP applications, a SIP call was made with a softphone and an Ekiga.net account. Ekiga is an open source VoIP and video conferencing application for gnome and windows. It supports SIP as well as H.323 protocols and is fully interoperable with any other SIP compliant application.
The call was made to an echo test number of ekiga (sip:500@ekiga.net). This feature makes it possible to evaluate the call because it echos whatever you said immediately back to you. Like the previous section, jitter will be measured by making this call with a HomePlug adapter and one time without. The additional delay of 2ms can be neglected as proved before.

The conversations were captured with ethereal/wireshark in order to analyse the jitter that is present in the streams. When doing this, the following pictures were made.

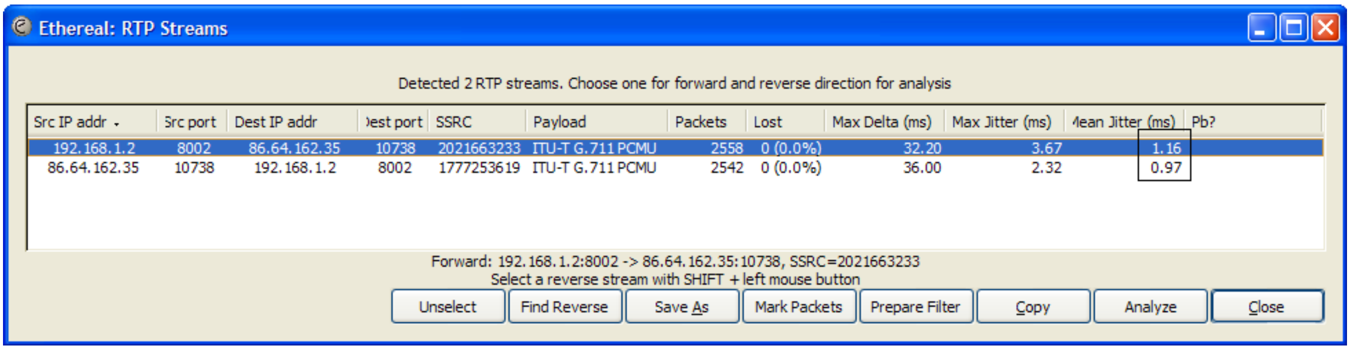


Figure 5.4: Ethereal – Jitter analysis of VoIP call to echo test server with Ethernet

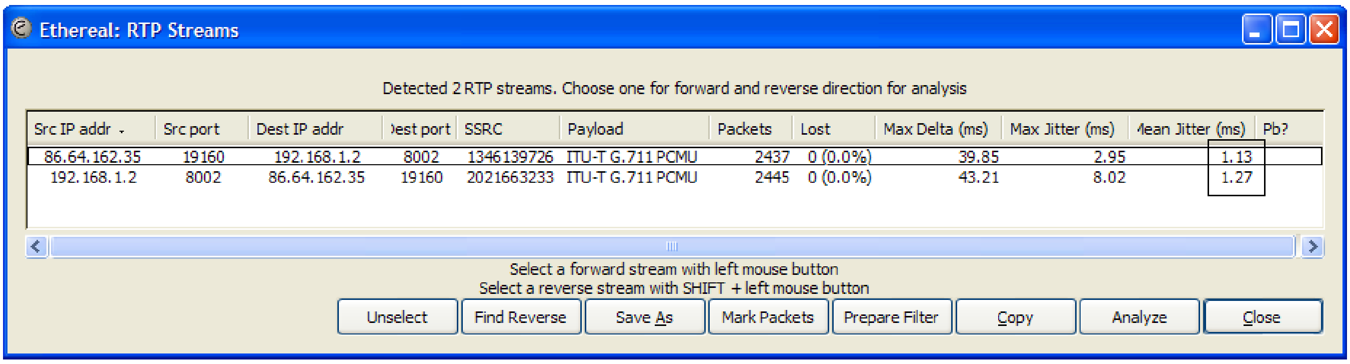


Figure 5.5: Ethereal – Jitter analysis of VoIP call to echo test server with HomePlug

From figures 5.4 and 5.5 you can see that two streams are present: one from the softphone (192.168.1.2) to the server (86.64.162.35) and one from the server to the softphone. The mean jitter is really low in both cases and the difference is negligible although the jitter peaks a little bit more with HomePlug compared to ethernet.

Calling between softphones

Until now only VoIP calls between a softphone and a test server were carried out. Now let's turn our attention to normal VoIP conversations with two VoIP softphone users.

The test we are about to describe involves two users with the same ISP (scarlet), located in the same geographical region and with both the VoIPCheap softphone (two voipcheap accounts, so the registration went through a voipcheap server).
The voice quality was very good and the initialisation of the call had no problems whatsoever. But when looking at the captured streams with ethereal/wireshark (see Figure 5.6), we noticed that there was only UDP traffic via the server, so no direct traffic between both users. This is rather strange because the SIP protocol describes that only the initialisation should go via the server and once the connection is set up, all communication should occur directly between both users. That is why we assume that VoIPCheap uses some proprietary dirty tricks to get round the firewall and NAT (network address translation). The Operating System's firewall and the NAT system can form a big problem when using VoIP to make WAN calls. Besides proprietary protocol solutions (like also skype uses), other protocols such as STUN (Simple Traversal of UDP through NAT) or ICE (Interactive Connectivity Establishment) are used to overcome these problems. Anyway this VoIP call went smoothly keeping the previous remark in mind.

In order to obtain a real SIP call with the proper initialisation, we tried to place a call between the same users but with other softphones (Express Talk and Xlite) and via an other SIP server (Ekiga.net). A few problems were encountered however. When capturing the conversation (see Figure 5.7), it became clear that the SIP protocol worked like it is supposed to be and the call was succesfully set up. But once the initialisation was done, there was only one-way traffic between the users. Both users saw traffic from them to the other party but no traffic coming back. This problem is caused by the previously described issue about firewalls and NAT boxes. When the right ports were opened or

No.	Time	Source	Destination	Protocol	Info
33	17:04:26.024540	192.168.1.2	194.221.62.194	UDP	Source port: 49473 Destination port: 11113
34	17:04:26.024983	192.168.1.2	194.221.62.194	UDP	Source port: 49473 Destination port: 11113
35	17:04:26.028509	192.168.1.2	194.120.0.19	UDP	Source port: 49473 Destination port: 11937
36	17:04:26.059456	194.221.62.194	192.168.1.2	UDP	Source port: 11113 Destination port: 49473
37	17:04:26.087611	194.120.0.19	192.168.1.2	UDP	Source port: 11937 Destination port: 49473
38	17:04:26.088021	192.168.1.2	194.120.0.19	UDP	Source port: 49473 Destination port: 11937
39	17:04:26.147784	194.120.0.19	192.168.1.2	UDP	Source port: 11937 Destination port: 49473
40	17:04:26.148178	192.168.1.2	194.120.0.19	UDP	Source port: 49473 Destination port: 11937
41	17:04:26.263574	194.120.0.19	192.168.1.2	UDP	Source port: 11937 Destination port: 49473
42	17:04:26.263984	192.168.1.2	194.120.0.19	UDP	Source port: 49473 Destination port: 11937
43	17:04:26.269591	192.168.1.2	194.120.0.19	UDP	Source port: 49473 Destination port: 11937
44	17:04:26.511632	194.120.0.19	192.168.1.2	UDP	Source port: 11937 Destination port: 49473
45	17:04:26.512034	192.168.1.2	194.120.0.19	UDP	Source port: 49473 Destination port: 11937
46	17:04:26.519584	192.168.1.2	194.120.0.19	UDP	Source port: 49473 Destination port: 11937
47	17:04:26.763821	194.120.0.19	192.168.1.2	UDP	Source port: 11937 Destination port: 49473
48	17:04:26.764234	192.168.1.2	194.120.0.19	UDP	Source port: 49473 Destination port: 11937

Figure 5.6: Ethereal – Capture of a VoIP call between two VoIPCheap softphone users

when we placed ourselves in the DMZ (Demilitarized zone), those problems were solved and consequently the call went fine. It should be noticed however that the VoIP call with VoIPCheap went notably better than the call just described. The connection was more stable and the perceived quality was better.

No.	Time	Source	Destination	Protocol	Info
1	17:20:37.311790	192.168.1.200	86.64.162.35	UDP	Source port: 5072 Destination port: 5060
2	17:20:39.005640	192.168.1.3	192.168.1.255	BROWSE	Domain/Workgroup Announcement WERKGROEP, NT Workstation, Domain Enum
3	17:20:48.372285	192.168.1.200	193.194.136.137	STUN	Message: Binding Request
4	17:20:48.389380	193.194.136.137	192.168.1.200	STUN	Message: Binding Response
5	17:20:48.390663	192.168.1.200	86.64.162.35	SIP/SD	Request: INVITE sip:filip.ballon@ekiga.net, with session description
6	17:20:48.440049	86.64.162.35	192.168.1.200	SIP	Status: 100 trying -- your call is important to us
7	17:20:48.582151	86.64.162.35	192.168.1.200	SIP	Status: 180 Ringing
8	17:20:52.283411	62.235.152.144	192.168.1.200	UDP	Source port: 39607 Destination port: 8003
9	17:20:52.323698	62.235.152.144	192.168.1.200	UDP	Source port: 39606 Destination port: 8002
10	17:20:52.341197	62.235.152.144	192.168.1.200	UDP	Source port: 39606 Destination port: 8002
11	17:20:52.361696	62.235.152.144	192.168.1.200	UDP	Source port: 39606 Destination port: 8002
12	17:20:52.381681	62.235.152.144	192.168.1.200	UDP	Source port: 39606 Destination port: 8002
13	17:20:52.407710	62.235.152.144	192.168.1.200	UDP	Source port: 39606 Destination port: 8002
14	17:20:52.420143	86.64.162.35	192.168.1.200	SIP/SD	Status: 200 OK, with session description
15	17:20:52.421374	192.168.1.200	86.64.162.35	SIP	Request: ACK sip:filip.ballon@62.235.152.144:40606;rinstance=6726231995c5dd03
16	17:20:52.421948	62.235.152.144	192.168.1.200	RTP	Payload type=ITU-T G.711 PCMU, SSRC=390383492, Seq=6325, Time=840400
17	17:20:52.441633	62.235.152.144	192.168.1.200	RTP	Payload type=ITU-T G.711 PCMU, SSRC=390383492, Seq=6326, Time=840560
18	17:20:52.461690	62.235.152.144	192.168.1.200	RTP	Payload type=ITU-T G.711 PCMU, SSRC=390383492, Seq=6327, Time=840720
19	17:20:52.481555	62.235.152.144	192.168.1.200	RTP	Payload type=ITU-T G.711 PCMU, SSRC=390383492, Seq=6328, Time=840880
20	17:20:52.488952	192.168.1.200	62.235.152.144	RTP	Payload type=ITU-T G.711 PCMU, SSRC=2021663233, Seq=53709, Time=22369128, Mark
21	17:20:52.501691	62.235.152.144	192.168.1.200	RTP	Payload type=ITU-T G.711 PCMU, SSRC=390383492, Seq=6329, Time=841040
22	17:20:52.510390	192.168.1.200	62.235.152.144	RTP	Payload type=ITU-T G.711 PCMU, SSRC=2021663233, Seq=53710, Time=22369288

Figure 5.7: Ethereal – Capture of a VoIP call between Express Talk and XLite

As a final test, we used the MyVoipSpeed server program. This application very accurately measures the quality and performance of your Internet connection for Voice over IP (VoIP) usage by simulating a real VoIP session across the Internet. It measures both the QoS and the MOS score. The MOS score (mean opinion score) is frequently used besides the QoS parameters when dealing with VoIP. This score provides a numerical indication of the perceived quality of received media after compression and/or transmission. The MOS is expressed as a single number in the range 1 to 5, where 1 is lowest perceived quality, and 5 is the highest perceived quality.

One user installed the server program and another user can initiate the speed test by surfing to the program's server web interface. Notice that this test was carried out between the same two users as above. The following summary, generated by the program, describes the MOS score.

```
Summary
-------
Download speed: 316712 bps
Upload speed: 204240 bps
Quality of service: 97 %
Download test type: socket
Upload test type: socket
Maximum download pause: 60 ms
Average download pause: 36 ms
Minimum round trip time to server: 61 ms
Average round trip time to server: 71 ms
Jitter: you --> server: 3.6 ms
Jitter: server --> you: off
Packet loss: you --> server: 0.0 %
Packet loss: server --> you: off
Packet discards: 0.0 %
Packets out of order: 0.0 %
Number of supported VoIP lines: 3
Estimated MOS score: 4.0
```

As a conclusion one might say that VoIP does not form any additional problems when used with the HomePlug technology. On the contrary, VoIP works just as well as with ethernet as you can see from the 4.0 MOS score and the 97% QoS score!

5.3 Application anthology

The goal of this section is to provide an overview of other popular applications accompanied with a short explanation to describe their interoperability with HomePlug. For these applications no big problems should be expected but they are mentioned because they contribute to a better overall picture.

Peer-to-peer

P2P or peer-to-peer applications are very popular because of their filesharing capabilities. A pure peer-to-peer network does not have the notion of clients or servers, but only equal peer nodes that simultaneously function as both clients and servers to the other nodes on the network. The big advantage is that the load is spread among all the nodes and that there is no single point of failure. Napster, Kazaa, LimeWire are some popular filesharing programs that use this network architecture. These programs were tested and were found perfectly operational when used with HomePlug.

BitTorrent

The bit torrent (BT) protocol is a special and popular P2P communication protocol for file-sharing. With a BitTorrent client, which is any program which implements the BitTorrent protocol, you are capable of preparing, requesting, and transmitting any type of computer file over a network, using the protocol. A peer is any computer running an instance of such a client. Though both ultimately transfer files over a network, a BitTorrent download differs from a classic full-file HTTP request in several fundamental ways: BitTorrent makes many small P2P requests over different TCP sockets, while web-browsers typically make a single HTTP GET request over a single TCP socket.
BitTorrent downloads in a random or "rarest-first approach that ensures high availability, while HTTP downloads in a contiguous manner. By using the Torrent client it became clear that HomePlug works fine with this BT protocol.

Web browsing

The Hypertext Transfer protocol (better known as HTTP) is a method used to transfer information on the world wide web. Its original purpose was to provide a way to publish and retrieve HTML pages. Since HTTP uses simple TCP connections (on the well-known port 80), no problems were encountered when surfing the WWW or when HTTP-streaming from a website (e.g youtube).

Internet Radio

Internet radio or e-radio is an audio broadcasting service transmitted via the Internet. Audio is streamed to the listeners in a continuous way on which they have no control, much like traditional broadcast media. It should not be confused with podcasting which involves downloading of the media and therefore also copyright issues. Nor does e-radio suggest on-demand file serving. Mostly it uses the popular MP3 codec or Ogg Vorbis. The stream is sent over the network in TCP or UDP packets. Just as with the previous applications, HomePlug works perfectly together with this one.

File Transfer

The File Transfer Protocol or FTP is the most popular protocol to transfer data from one computer to another over the Internet, or any network for that matter. This protocol runs over the TCP protocol exclusively. From the successful transmissions of data using programs like GuildFTP Server and the windows built-in ftp client, one may conclude that also FTP applications are not hindered by HomePlug.

Online gaming

Online gaming has emerged as one of the fastest-growing applications for home networking. No longer is the game limited to how many controllers can be attached to a PC or game console or to the size of the display. Now, gamers can play against others in the house over the home network or can play against others from around the globe. That's why it may be interesting to also review this trend. The most important QoS parameter for this application is delay (called lag in the gaming world). That's because gaming is fast real-time action and to see were a player was a second ago is not interesting. A very famous game called Counter-Strike was tested with HomePlug for its behaviour while gaming. Thanks to an in-game tool the specifics (down/up rate, loss, choke and response time) could be monitored. They showed no notable differences when gaming with HomePlug or with ethernet.

Figure 5.8: Legend for user scenarios

5.4 User Scenarios

As promised before, this chapter will be ended by a high-level overview given by two user scenarios to obtain a more real-life feeling with the HomePlug technology. Actually this section adds all previously discussed topics together: multiple concurrent streams, multicasting, using adapters with different standards at the same time, various applications, . . . This section will actually test the backbone qualities of the HomePlug technology. Throughout the use cases the legend from Figure 5.8 will be utilised. The scenarios are described using the same house as in section 4.2. The ground floor and the first floor are shown in figure 5.9 and figure 5.10 respectively.

5.4.1 Use case 1

In this first scenario we have a modern family with a teenager and two adolescents with an ADSL Internet connection. The multimedia server is located in Room 3 and provides multimedia to the network users. Below the action of every family member is described.

- Child: Playing online games on his PC in his bedroom. (Room 5)
- Adolescent 1: Surfing the Internet and streaming music from the multimedia server. (Room 4)

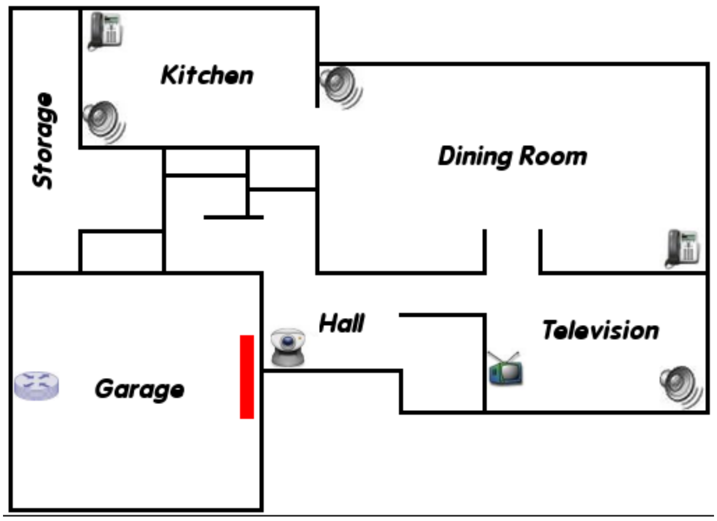


Figure 5.9: Home environment – downstairs

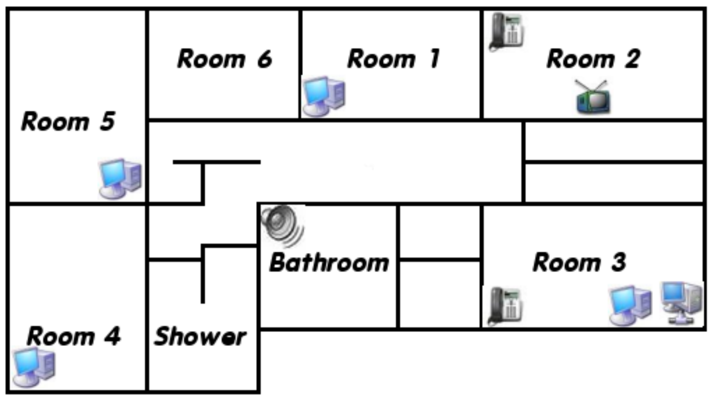


Figure 5.10: Home environment – upstairs

- Adolescent 2: Surfing the Internet and streaming music from an Internet radio station. (Room 1)
- Father: Watching digital television. (Television)
- Mother: Calling with a VoIP hardphone to one of her friends. (Kitchen)

An important aspect to evaluate this scenario is bandwidth. Here are the bandwidth requirements (approximately) for the given applications:

- Online gaming: 100Kbps
- Streaming audio from multimedia server: 192Kbps
- Surfing: 100Kbps
- Streaming audio from an Internet radio: 128Kbps
- Watching digital television: 15-20Mbps
- VoIP conversation (using G.711 codec): 174Kbps

It is obvious that watching digital television requires the most bandwidth of all applications. Digital television can be received through cable or DSL. In this case it is watched through a DSL provider. Digital television implicates that a set-top box (STB or better know as digibox), which decodes the digital signal to analog so it can be displayed on televisions without a digital tuner, is connected to the external source of signal through cable, ethernet, telephone line, even BPL. When using DSL, a telephone line is connected to a modem/router which on his turn must be connected to the STB. In our use case, of course, this connection is established using HomePlug.
For this application the HomePlug AV adapter, with its maximum 'real 60Mbps TCP throughput, from Devolo comes in handy. There is a drawback however as seen in section 4.5 dealing with multiple concurrent streams. Whenever another stream is set up, suppose by two HomePlug turbo adapters, that stream will suffer enormously from the television stream. That's because the average throughput for the turbo adapters is approximately 20Mbps on TCP level which is approximately the required bandwith for digital television.

This is however a worst case scenario, because when the bandwith required for the television stream is less than 17Mbps, there is still more or less 3Mbps bandwith available for the other streams which require in this case only ±1Mbps. Seen the bad interoperability between AV and turbo adapters, it's not a good and efficient idea to use these two kinds of adapters together and should better be used seperately. The best but most expensive solution is to provide your whole network with AV adapters. When only turbo adapters are used or a mix of AV and turbo adapters this scenario can be on the edge of being operational. Also distance is an important factor. When distances become too long, the TCP throughput may not even reach 20Mbps.
When the (digital) television is watched through cable, this activity should not be taken into account because then the signal is provided by cable instead of a telephony line. Only the control signals (e.g. switching channels, interactive television) go to the DSL router (by means of HomePlug) which only require minimal bandwidth. When this is the case, no problems are encountered whatsoever when dealing with this specific use case because of the low bandwith requirements of the other applications.

One can notice the speakers throughout the house. They can be used to play audio coming from various sources. With HomePlug's audio technology one can stream audio through the power cabling. That means you do not have to have your audio source next to your speakers. You can listen to songs from a PC that is located downstairs, in the bathroom for example. However this technology was not studied here.

To conclude it can be said that when using only HomePlug turbo adapters, much of the succeeding of this use case relies on the way digital television is provided and also the distances between the endpoints of every connection that is made. For efficient use, the router, that is connected to the Internet, should be best located near the service panel. Then the distances between every home outlet (on different circuits) and the router, when a certain outlet wants to access the Internet, are more or less the same. To further increase efficiency, one should connect all endpoints of a specific connection (point-to-point, multicast, ...) to the same circuit so minimal distances could be realized between those endpoints.

5.4.2 Use case 2

A second scenario could be a group of friends, assume young adults, who live together. Depending on the ages of the inhabitants, the applications used will of course change accordingly. The same house is used as before with exactly the same devices as in figure 5.9 and figure 5.10. This scenario is started by giving the actions of every inhabitant.

- Friend 1: Giving a party and streaming a High Definition (HD) movie from the multimedia server. (Television)
- Friend 2: Watching the IP surveillance camera from outside the house.
- Friend 3: Surfing the Internet while calling with his VoIP softphone on his PC. (Room 5)
- Friend 4: Listening to her favorite music from the multimedia server while downloading using a BT client. (Room 4)

Here are the bandwidth requirements (approximately) for the given applications:

- Streaming HD content from multimedia server: 19-20Mbps
- Watching the camera (using MPEG-4 unicast): 1Mbps
- Surfing: 100Kbps
- VoIP conversation (using G.711 codec): 174Kbps
- Streaming audio from multimedia server: 192Kbps
- Downloading using BitTorrent: 1Mbps

Also in this scenario there is one big consumer namely the HD content stream. In Figure 5.9 only a TV is present in the television room where the stream is being sent to. But one may assume for example that a PC is connected with his TV-out (option on computer video card) to the television so the stream is being sent to that PC.
Most professional IP cameras support multiple viewing formats (e.g. Motion JPEG or MPEG-4). When using Motion JPEG, which uses standard JPEG still images in the

video stream, images are displayed and updated at a rate sufficient to create a stream that shows constantly updated motion. The Motion JPEG stream uses considerable amounts of bandwidth, but also provides excellent image quality and access to each and every individual image contained in the stream. A more bandwidth-saving technique is MPEG-4. This is a video compression standard that makes good use of bandwidth, and which can provide high-quality video streams at less than 1 Mbps. That's why MPEG-4 is chosen in this particular use case. One should also notice the camera is being consulted from outside the LAN so firewall settings must be altered in order to pass the security precautions. Most cameras support various applications to access the unicast (or multicast) streams.
The BitTorrent user has a download speed of 1Mbps (which equals 125KByte/sec). This may of course increase (and decrease) in time but we assume it is limited by a download manager.

As in the previous use case there is one spoilsport being the HD stream. When only AV adapters are used, no problems will arise in this user scenario. But when Turbo adapters are used this scenario will probably fail because the total required bandwidth exceeds the average of 20Mbps. When short distances are encountered however between crucial endpoints (like the two endpoints of the HD stream), it may still be operational. Also here it is imperative and wise that the endpoints of a certain connection are connected to the same circuit to ensure efficient use of the communication medium. When we replace the HD movie by a DivX movie (a popular video codec using lossy MPEG-4 Part 2 compression) this scenario has a good chance of succeeding because this stream only requires more or less 2Mbps instead of the tenfold.

5.4.3 Conclusion

It is obvious from these use cases that there are some applications that are deadly for an operational network with HomePlug as a backbone. Those application are streaming High Definition content and receiving digital television entirely through a telephony line. When these modern technologies are being asked for, one is advised to use only HomePlug AV adapters because of their increased bandwidth and their favourable features. If no HD is used and digital television is provided by means of another method, the HomePlug Turbo adapters will normally do just fine if you keep the multiple streams to a certain minimum.

The decision whether to use HomePlug as a backbone for your home-network should not be based on this two user scenarios alone. There are many vague and unpredictable factors like distance, noise, electrical wiring, family situation, ... that should be taken into account when making this decision. Therefore it is rather difficult to give an unambiguous solution for every family or house situation. One should evaluate this technology using the acquired knowledge from this thesis and of course from your personal experience with HomePlug certified products.

6 Taking HomePlug to the next level

HomePlug is a network technology that can be used today. Many HomePlug products are on the market and people are getting more and more interested in this technology that upgrades the residential powerlines to a communication network. In this chapter we describe two interesting evolutions of the HomePlug technology that can extend the usability of HomePlug networks and increase performance. In the first section, utilities to extend the range of a HomePlug network are presented. In section 6.2 we describe how a HomePlug switch could be build to increase the performance of a HomePlug network in a home situation.

6.1 Extending the range of a HomePlug network

Normally, a HomePlug network can't communicate over powerlines longer than 200 meter. The HomePlug AV standard can communicate over powerlines up to 300 meter long. Still, in large buildings, longer powerlines can be encountered.

6.1.1 Linux IP Repeater

An article called "Simple Linux IP Repeaters to Extend HomePlug Range" [10] caught our attention. It presents an interesting device that can extend the range of a HomePlug network. The repeater only needs one HomePlug connection and can double the range of a HomePlug network. It is build on a linux machine.

First, packet forwarding must be enabled on the linux machine. This makes the linux machine act like a router. Incoming IP packets are inspected and forwarded if a route is

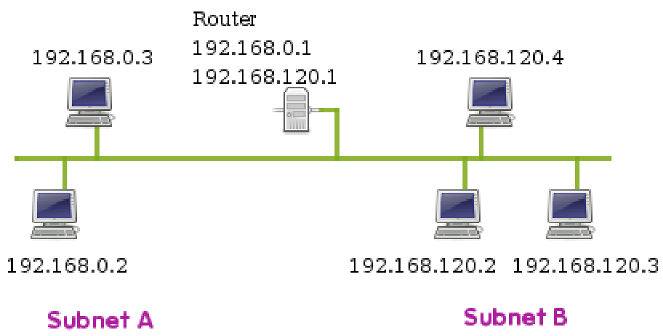


Figure 6.1: Linux range extender

found. To do so we used the command *sudo sysctl -w net.ipv4.ip_forward=1*.

Sysctl is a utility to edit and display kernel parameters at runtime. This command has immediate effect, no reboot is needed. The repeater makes use of another ability in linux, the ability to assign an alias to a network interface. Normally all interfaces are accessed using names like *eth0*, *eth1*, ... with each name corresponding to a network interface card. However, with names like *eth0:0* we can assign an alias to a network interface. Under linux this can be done with the following command: *sudo ifconfig eth0:0 192.168.0.60*. The network interface now listens to all aliases defined for this interface. Devices putting an ARP request on the network for 192.168.0.60 now get an answer, the MAC address of the repeater. IP packets sent to this address will be processed by the repeater and relayed as explained before.

To set up the repeater we give it a base IP address: 192.168.0.1. This is achieved with the command *sudo ifconfig eth0 192.168.0.1*. The subnetmask for this subnet is 255.255.255.0. The network is now divided into several subnets. Figure 6.1 shows how the network is divided into two subnets: 192.168.0.0 (A) and 192.168.120.0 (B). The repeater already belongs to the first subnet, now we need to configure it so it belongs to the second subnet as well. We use the command *sudo ifconfig eth0:0 192.168.120.1* to do this. The repeater now listens on the same network interface (eth0) to packets from both subnets and relays them to each other if needed. To make communication possible between all hosts belonging to both subnets, a default gateway must be set on all hosts. Hosts on subnet 192.168.0.0 set their default gateway to the repeater, i.e. 192.168.0.1. Hosts on subnet 192.168.120.0 set their default gateway to 192.168.120.1. On a linux machine, this can be done with the command *route add default gw 192.168.0.1*. This way, packets sent from one subnet to another always go to the repeater, where they are relayed to the correct destination. More subnets can be added.

To automate the configuration of the repeater, one of the autors of the article, Pedro S. Rodríuez Hernández, wrote a program, *hprmanager*, to automate this configuration. We were able to obtain the source code from him by mail. The program is written in C and only runs under linux.

The program is very simple. It places the network card into promiscuous mode to listen to all packets on the network. Because the network card is connected to a HomePlug adapter, it will only capture traffic that is intended for the repeater (packets with destination the MAC address of the repeater). These packets are monitored and their source and destination IP address is compared with a table of known subnets. If it detects an IP address that belongs to a new subnet, it executes the command *ifconfig eth0:x 192.168.y.1*. 'eth0:x' denotes the next free alias and 192.168.y.1 is the first IP address in the newly detected subnet (192.168.y.0 is this subnet). The repeater is always given an IP address in the form 192.168.y.1. Now, new subnets can be added dynamically without needing to configure the repeater manually. A host can assign itself to a new subnet (192.168.y.0) and must set it's default gateway to 192.168.y.1. The repeater now automatically will add an alias for this subnet and name itself 192.168.y.1 in this new subnet.

Using only one HomePlug is very attractive, because it reduces the costs of the repeater, but this solution does require a linux box. Fortunately this repeater can run on an embedded system.

6.1.2 Extending range with two HomePlug adapters

Another solution that can double the HomePlug range was found on the Devolo website. They suggest to use two HomePlug powerline-to-ethernet bridges to build a range extender. A simple ethernet cable connects the two bridges. Some additional measures have to be taken. The two bridges have to be in different logical networks. The bridges of the endpoints must be in the same logical network as one of the bridges that form the range extender. To connect two devices, two logical networks must be created: one logical network connects one endpoint and one bridge of the repeater, the other two bridges belong to the second logical network.

This is a configuration that works fine. It is crucial that both bridges of the range extender don't belong to the same logical network. If they do, a single packet can trigger an infinite loop. A frame comming from one endpoint arrives at both bridges of the repeater. If both repeaters can decode the frame, they will place it on their ethernet interface and both send it to each other. Both bridges will receive this frame on their ethernet interface and put in on the powerlines. Both bridges will receive their original frame again and will continue to send this frame over and over. We experienced this ourselves when we forgot to program the bridges with different logical networks. The first packet arrived, but it triggered an infinite loop that caused the network to be 100% loaded from then on. When both bridges belong to seperate logical networks only one of them will be able to decode the frame and forward it to the other bridge. This bridge will place the frame on the powerline, but the other bridge won't be able to decode it again, only the other endpoint will receive the frame.

6.1.3 Placement of the range extender

The position of the range extender is very important. In an ideal situation it should be located in the middle between both endpoints (we assume two endpoints have to be connected by means of a HomePlug network). Even if it would be possible to locate this exact location (which is practically impossible) there probably wouldn't be an outlet to connect the repeater to. If more then two endpoints have to be connected, an exact center of endpoints doesn't even exist.

A very good position of the range extender is at the distribution box. This is both logically and topologically the center of the residential powerlines. All the branches of the electrical circuit start here and form a star with the distribution box as the center. Locating the extender here ensures that all endpoints probably can see the repeater. When one subnet is located on one branch of the electrical circuit, and another subnet on another circuit, all communication can happen through the range extender.

6.1.4 Comparing both range extenders

Both solutions have their advantages and disadvantages. The *cost* of the extender is of big importance. The embedded solution requires the purchase of a HomePlug adapter and an embedded device that runs linux. The second solution requires two HomePlug adapters. Buying an embedded device will be more expensive than buying a second HomePlug device.

Another way to compare the two solutions is to look at the *ease of installing the extenders*. Here the solution with two adapters has an advantage. Only the different logical networks need to be configured, once at deployment. The linux-solution requires the HomePlug network to be divided into subnets. If a the linux repeater would be sold with all software pre-installed it would be only a matter of plugging in the repeater, but since no out-of-the-box solutions exist, it would require alot of time and effort from the user. A non-experienced user probably wouldn't even be able to successfully setup an embedded device to act as a range extender.

The linux-solution requires a division of the HomePlug network on the network layer. Endpoints have to belong to a certain subnet, and communication between subnets goes through the repeater. The two-adapter solution requires a division of the HomePlug network into different logical networks, which is a division at the data link layer.

Extensibility is anoter issue. If more than two sites have to be connected the linux-solution has to be used. Even with an increasing number of sites (subnets in this case) the cost of the repeater doesn't rise, nor does the complexity of configuration.

Performance is another important criterion. In section 6.1.5 the performance of both range extenders will be compared.

6.1.5 Comparing performance

To compare the two repeaters we performed a test where we measure throughput, latency and jitter. We installed both endpoints in two corners of the same room and positioned the range extender in the middle of this room. This way we could compare the performance with both range extenders. Note that this situation is not ideal because both endpoints could see each other anyway, even without a range extender in place. However, this test demonstrates clearly how the range extenders operate and what their impact is on data transfer. The Devolo dLan Higspeed 85 adapters were used in this test. Throughput and jitter were measured with *iperf*, latency was estimated with the *ping* utility.

Results

These are the results that were measured:

Table 6.1: Comparing range extenders

	No extender	Linux solution	Two-adapter solution
TCP - up (Mbps)	11,30	6,91	7,76
TCP - down (Mbps)	9,13	4,70	5,73
UDP - up (Mbps)	14,20	9,43	9,12
UDP - down (Mbps)	15,70	6,34	6,64
round-trip-time (ms)	2	4	4
Jitter - up (ms)	1,939	3,181	2,494
Jitter - down (ms)	1,999	4,024	2,382

Conclusion

The two extenders decrease the throughput with 30% to 60%. This is expected, because both links (from the extender to both endpoints) will interfere with each other. frames sent from endpoint 1 to the extender must contend for the medium because simultaneously the extender tries to send frames to the second endpoint. This is because both links share the same medium. A similar situation exists as described in section 4.5 where two streams are contending for the medium.

The extender with two adapters performs slightly better than the linux-solution, but these are only little differences. Note that in a real situation where two endpoints that can't barely *see* each other, the bandwidth would increase instead of decrease. When two endpoints can't see each other without an extender, and are able to communicate with an extender in place, then this is obviously an improvement. When both endpoints can barely see each other, an extender can increase the throughput because the number of usable subcarriers increases.

As expected the round-trip-time doubled from 2ms to 4ms in both cases. A repeater thus doubles the latency, if the two endpoints are directly connected to the bridge.

Jitter increases with 0,5 ms if we used the two-adapters as an extender. An increase in jitter of about 2 ms is measured with the linux-solution. But again, the increase is only small.

6.1.6 A case-study

So far, we only tested the range extenders in a situation where both endpoints could communicate with each other anyway. This is not a good situation to use the extenders. That's why we went looking for a situation where this range extenders would be useful. This would have to be a big building, because in a typical home all HomePlug devices at all outlets can be seen by all other devices. We went to the cultural center "De Borre" in Bierbeek. This is quite a big building. We found a sitution where both endpoints couldn't see each other, but when a third device was plugged in, then the endpoints could see the HomePlug adapter plugged in the middle. This was exactly the situation we were looking for.

The device in the middle was actually inserted in the main electrical distribution box, there was access to one outlet. First we tested the linux-box, since this solution only requires one outlet it was easy to set up. Then we measured the throughput to both endpoints. We did this again with iperf and only measured TCP throughput. Then we configured the range extender and measured the throughput between both endpoints.

Results

The results of the measurements are shown in table 6.2.

Table 6.2: Case-study – throughput results

	up (Mbps)	down (Mbps)
Endpoint 1 to Extender	9,43	2,99
Endpoint 2 to Extender	0,29	3,09
Endpoint 1 to Endpoint 2	0,23	0,36

Conclusion

The throughput between the two endpoints is still very disappointing, but we have to keep in mind that without the range extender in place, no communication at all was possible with HomePlug. We see that the bottleneck here is the connection between Endpoint 2 and the extender in the middle. It is obvious that if one of the two links to the extender is very slow, then no matter what the speed is on the other link, the overall speed will be very low. Finding a good location for the extender is crucial but not always possible. With trail and error a better location could be found.

We also wanted to test the other extender with the two adapters. First of all we had a problem to find two outlets because only one outlet was available on the distribution box. We used another outlet in the vicinity of the distribution box and connected both adapters with an ethernet cable. Still we weren't able to connect both endpoints. In this case, only the Linux-box solution was able to connect both endpoints.

6.1.7 A monitor for the Linux-based range extender

The *hprmanager* program that automates the configuration of the range extender works fine. We wrote a program that can monitor how many subnets are being added, and how many clients are on these subnets. This way we can easily get an overview of the network topology. We have to consider that the hprmanager can run on a headless embedded linux-box, so we chose to output the information via a webpage that can be consulted on a browser from another computer in the network.

The first thing we did is alter the original program so that it logs all the subnets it discovers to a file. This discovery is logged in the form "subnet: 192.168.0.0". The original program only kept a list of all discovered subnets, so another adaptation was needed. A list of discovered clients also has to be kept in memory. Every time a new client is seen, this list is updated. Simultaneously a record in the form “node: 192.168.0.2” is added to the log-file. The program is thus altered to log both new subnets and clients it discovers to a log-file.

An example log-file could look like this:

```
subnet: 192.168.0.0
node: 192.168.0.1
node: 192.168.5.55
subnet: 192.168.5.0
node: 192.168.5.85
node: 192.168.7.85
subnet: 192.168.7.0
node: 192.168.0.2
```

A second part of our solution is programmed in PHP. This is a web scripting language, that can be used to make dynamic webpages. A PHP script was made to interpret the log-file, and represent a tree of subnets, and discovered clients on these subnets. An example of the output is shown in figure 6.2.

6.2 A HomePlug switch

In this section another possible evolution of powerline communication will be discussed. In analogy with the ethernet switch, a HomePlug switch is presented. It would enable a higher throughput for all applications using the HomePlug network to communicate with each other. Notice that we present only a theoretical device that is not on the market today.

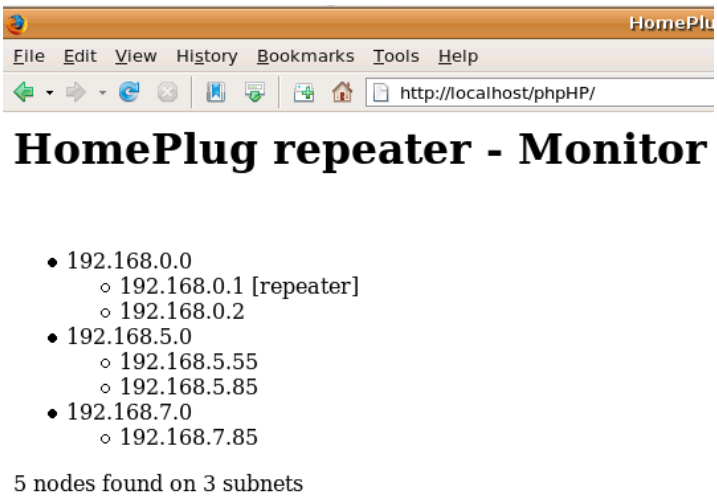


Figure 6.2: Topology monitor

6.2.1 An ethernet switch

An ethernet switch is a device that operates on the second layer, the Data Link Layer, of the OSI model. As networks grow bigger and bigger, they can become very slow due to the high number of collisions that occur on the network segment that is shared by all clients. A switch eliminates the number of collisions by reducing the collision domain to one client. Every device that is connected to a port of the ethernet switch is physically separated from the other devices that are connected to the same switch. Although they belong to the same ethernet segment, no collisions can occur due to the switch. The switch is intelligent enough to deliver a copy of frames it receives to the devices that are meant to receive these frames. An example ethernet switch is shown in figure 6.3. If for example PC1 sends frames to PC4, these frames will only be copied to port 4 of the ethernet switch. Port 2 and 3, and thus PC2 and PC3 won't notice these frames, and thus won't cause collisions if they wich to send at the same time. PC2 can simultaneously send frames to PC3 without causing any collision.

The switch maintains a list of MAC addresses and the physical port at which they are connected. By observing the ethernet frames that pass by it can learn the topology of the network. If no entry is found in the table, it forwards the frame to all the ports. Soon, a reply will follow and the switch will learn at which port the device is connected. This way the capacity of the network can be increased dramatically. The collision domain is reduced

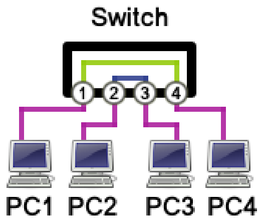


Figure 6.3: Ethernet switch

to one device, while the broadcast domain remains intact. Notice that broadcast-frames (with an all 1's MAC address) are forwarded to all ports.

6.2.2 A HomePlug switch

A similar approach is possible with PLC networks. The network can be divided into smaller segments to reduce the collision domain and to optimize the efficiency of the network. This division is not as simple as with ethernet though. It's impossible to separate all the devices so that they operate in their own collision domain. The original division of the powerlines into several circuits can be used instead. As seen before, an in-home powerline grid exists of several branches that start at the distribution box. If it is possible to separate these branches, a number of groups exist. One group contains all the devices that are plugged in outlets of *circuit A* while, for example, another group contains all the devices that are plugged in outlets of *circuit B*. Although complete separation is impossible, devices can be grouped and the collision domain can be made smaller.

Another problem arises: how can we separate the electrical in-home circuits? If no separation is done, the high frequency signals propagate from circuit A to circuit B and vice versa. Collisions (between members of two different groups) would still occur. That's why a set of (low-pass) filters is needed to remove the HomePlug signals when going from one circuit to another.

One device can be used to interconnect all the circuits so that communication between them becomes possible. This device could operate in the same way as the ethernet switch operates. It can inspect all MAC addresses passing by and update it's table that associates MAC addresses with the ports. The routing could also be done on the third layer, the

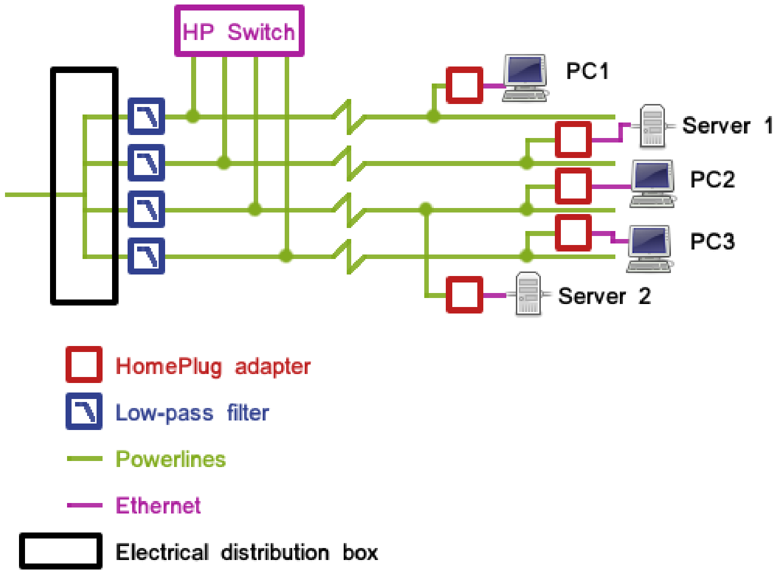


Figure 6.4: HomePlug switch

Network Layer. A router could inspect the packets and decide on which port they have to be forwarded. The circuits would define subnets, with the devices plugged in the outlets belonging to these subnets. The HomePlug switch is shown in figure 6.4.

In this example four circuits separated by the low-pass filters. The HomePlug switch interconnects them. PC1, PC3 and Server 1 have an ideal collision domain: they don't have to share the circuit with other HomePlug devices (other than the switch). PC2 and Server 2 have to share a circuit. Communication between PC2 and Server 2 happens without intervention of the switch. They both are members of the same subnet.

Notice that such a device would also extend the HomePlug network as explained earlier in section 6.1. This would enable two endpoints to be separated 400m from each other. Throughput between these endpoints would not decrease as with the solutions presented in section 6.1 since the two circuits to which the endpoints are connected are separated physically by the filters and no collisions occur with the original information stream.

This solution requires a drastic change of the electrical equipment. Every circuit would need a filter that has to be installed between the distribution box and the rest of the powerlines. Moreover, the HomePlug switch would need access to all the circuits by means of outlets that connect to the circuits *behind* the filters. This would require high installation and equipement costs. The HomePlug switch could be integrated in the electrical distribution box. Installation would become easier an cheaper. In newly build houses this would require no great additional costs.

Upgrading to a newer PLC standard, with such an installation in place, doesn't mean that the whole system would become obsolete. The filters would still do their job, only the switch would require access to the powerlines using the new technique. Upgrading from a HomePlug 1.0 to a HomePlug AV network would only require a new set of adapters to connect the switch to the circuits. Obviously, all the devices in the home would also need new HomePlug AV adapters. The switch wouldn't even have to be reconfigured, because it routes the packets on Network Layer. It sees no difference between HomePlug 1.0, HomePlug AV, WiFi or ethernet networks as the backbone.

In addition, the (broadband) internet connection is often located in the vicinity of the electrical distribution box: in the garage or the basement. The switch would be able to provide internet access to all devices in the network.

Conclusion

In our study to answer the question *can HomePlug technology serve as a backbone for the future home-network* we can make the following conclusions:

The *ubiquity of outlets* in a standard home makes this powerline communication technology suitable for modern domestic applications. Although WiFi technology allows even more flexibility, outlets are found in every room of a standard home. HomePlug offers a bigger and more stable throughput than WiFi throughout the house. Moreover, it can deliver broadband access to the home-network in places where WiFi fails (garage and basement for example).

HomePlug delivers a *QoS* that is sufficient for VoIP, streaming (high definition) video, streaming audio and less demanding applications like ftp and internet browsing. Improvements are possible, because VLAN priority tags are currently unused. Labeling network traffic with appropriate VLAN tags would improve the QoS even more.

The available *throughput* of a HomePlug network is one of the limiting factors. HomePlug 1.0 turbo adapters (with a 85 Mbps throughput at the physical layer) are unable to convey multiple high definition video streams. Less demanding multimedia content, like DivX movies, can be sent over a HomePlug 1.0 turbo network. When multiple high definition streams have to be transmitted over the home-network, HomePlug AV adapters are a better but more expensive solution.

Security is not a problem, as long as the standard Network Encryption Key is replaced by a secret one at installation. No hardware nor software tools exist to eavesdrop on HomePlug networks. Moreover, an attacker must have access to the powerlines. An attack on a wireless network is much easier, since the attacker only has to be in the vicinity of the network.

Multicasting forms a problem for all powerline communication technologies. Since communication is optimized between two nodes on the network, sending a frame to several multicast subscribers requires another strategy. In most cases the frame is sent in a more robust way so all subscribers receive the frame. However, this results in slower communication. Multicasting high definition video content over a HomePlug home-network is not possible. This can become a large problem since more and more applications depend on this multicast ability. For most home networks today, however, this won't pose a problem, since most network traffic relies on unicasting.

The *cost per connection* to the HomePlug network is another disadvantage. Currently, one adapter per connection is needed. Since HomePlug is a relatively unknown technology, the cost per adapter is high. As more HomePlug products are being sold and the HomePlug hardware gets integrated in domestic devices, the price will drop.

HomePlug can become an *integral solution* for home-networks. More integration of HomePlug hardware in domestic devices would make this technology even more appealing. Devices with a connection to the powergrid for energy would require no new wires to have access to the home-network.

Bibliography

[1] *HomePlug Powerline Alliance.* Wikipedia. sept. 2006
<http://en.wikipedia.org/wiki/HomePlug_Powerline_Alliance>

[2] *INT51X1 datasheet.* Intellon Corporation. 2003

[3] Dölle, M., *Power Networking.* In: Linux Magazine, 38 (2004), pp. 36-39

[4] Gardner, S.; Markwalter, B.; Yonge, L., *HomePlug standard brings networking to the home.* CommDesign. dec. 2000
<http://www.comsdesign.com/main/2000/12/0012feat5.htm>

[5] *HomePlug 1.0 Technology white paper.* HomePlug Powerline Alliance. feb. 2005
<http://www.homeplug.org>

[6] Lee M.K.; Newman R.E.; Latchman H.A.; Katar, S.; Yonge L., *HomePlug 1.0 powerline communication LANs – Protocoll description and performance results.* sept. 2002

[7] *HomePlug AV White Paper* HomePlug Powerline Alliance. aug. 2005
<http://www.homeplug.org>

[8] *PLC Performance Testing White Paper* Intellon Corporation. 2006
<http://www.intellon.com/pdfs/whitepaper_PLCPerformanceTesting.pdf>

[9] Ahlers, E., *De draad kwijt.* In: c't Magazine, 11 (2006), pp. 118-123

[10] González-Castaño F.J.; Rodríguez-Hernández P.S.; Gil-Castiñeira F.J.; Rodelgo-Lacruz M.; Valero-Alonso J., *Simple Linux IP Repeaters to Extend HomePlug Range.* Linux Journal. sept. 2005
<http://www.linuxjournal.com/article/8166>